RL11,RLV11 RL01/02 DRIVE TST 1 AH-F118C-MC
CZRLICO FICHE 1 OF 1

IDENTIFICATION

PRODUCT CODE: AC-F119C-MC

PRODUCT NAME: CZRLIC0 RL01/02 DRIVE TEST 1

DATE CREATED: 5-JAN-79

REVISED: 27-FEB-80

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

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH XXDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP+, AND CAN BE CHAINED UNDER XXDP+, ACT AND APT IN ACT MODE (SEE 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, 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 TESTS THE RL01/02 INTERFACE AND BASIC DRIVE LOGIC. GET STATUS WITH RESET, GET STATUS, SEEK, AND READ HEADER ARE THE ONLY COMMANDS EXECUTED IN THE PROGRAM. ONLY SEEKS WITH 0 DIFFERENCE ARE USED SO NO HEAD MOVEMENT IS REQUIRED. A SIGNIFICANT PORTION OF THE PROGRAM REQUIRES MANUAL INTERVENTION. THESE TESTS TEST THE COVER OPEN AND WRITE LOCK STATUS. THE DRIVE MUST BE LOADED AND UNLOADED TO TEST ALL THE CONDITIONS OF HEADS OUT, BRUSH HOME, AND DRIVE STATES. THE PROGRAM CAN BE RUN IN AUTOMATIC MODE IN WHICH CASE ALL TESTS REQUIRING MANUAL INTERVENTION ARE BYPASSED.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

* KW11P CLOCK
* LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLIC0 RL01/02 DRIVE TEST 1

(FORMERLY CZRLCB)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 DISK SUBSYSTEM USER'S GUIDE (EK-RL012-UG-002)
XXDP+/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)

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 OCCURED. 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. YOU NOW HAVE THREE CHOICES:

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):

```
                                        BY
                                        WHOM
                                        ENTERED:
                                        -------

.R CZRLIC                               O
DRS LOADED                              D
DIAG. RUN-TIME SERVICES REV. D APR-79   D
CZRLI-C-0                               D
CZRLI TESTS THE RL01-02 INTERFACE       D
AND BASIC DRIVE LOGIC
UNIT IS RL01, RL02                      D
DR>STA/PASS:1/FLAGS:HOE                 D,O

CHANGE HW (L) ? Y                       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) ? N                       D,O

EXECUTE DRIVE SELECT TESTS (L) N ?      D,O
EXECUTE HEAD ALIGNMENT SUPPORT (L) N ?  D,O
DO MANUAL INTERVENTION TESTS (L) N ? Y  D,O
INPUT ERROR LIMIT (D) 20 ?              D,O


CZRLI 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

CZRLI 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. THE BIC FILFS ARE CREATED BY USING THE SETUP UTILITY PROGRAM WHICH IS USED T PARAMETERIZE THE DIAGNOSTIC PRIOR TO ITS EXECUTION. SETUP PROMPT' THE OPERATOR WITH THE HARDWARE AND SOFTWARE QUESTIONS. THE RESPONS_ TO THESE QUESTIONS ARE USED TO BUILD P-TABLES. THE RESULT OF THE SETUP PROCESS IS A FILE WHICH INCLUDES THE DIAGNOSTIC WITH APPENDED P-TABLES. REFER TO THE XXDP+/SUPERVISOR USER'S MANUAL FOR A COMPLETE DESCRIPTION OF THE SETUP UTILITY.

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 HARDWARE/SOFTWARE SWITCH REGISTERS 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 DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT
4.	AN ERFOR 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 MODE, BYPASSING MANUAL INTERVENTION 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 RESPONCE.

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 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.

EXECUTE DRIVE SELECT TESTS (N)?

IF "YES" TESTS 5 AND 6 ARE EXECUTED IN THE FIRST PASS OF THE PROGRAM. THESE TESTS REQUIRE MANUAL INTERVENTION TO CHANGE ADDRESS PLUGS AND REQUIRE A FULL COMPLEMENT OF ADDRESS PLUGS (0 - 3).

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

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

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

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 RELEVANT DATA.

3.1 ERROR REPORTING

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

```
(1)   PROG NAME   ERR NUM   TEST NUM   SUBTEST NUM   ERR PC
(2)   ROUTINE TRACE SEQ (IN SEQ CALLED)
         (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 SUB TEST 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 AS 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 CONTENTS OF CONTROLLER REGISTERS WHEN THE OPERATION WAS INITIATED.

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

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

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

3.1.1 SPECIFIC OPERATION MESSAGES

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

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

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

READ DATA WITH DATA COMPARE -

IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

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

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

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

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

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

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

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

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

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

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

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS:

RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)
WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTENT 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	(HEADS 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 ERROR 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 LOSING 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.

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

BIT 15 - COMPOSITE ERROR
BIT 14 - DRIVE ERROR
BIT 13 - NON EXISTANT MEMORY ERROR
BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
BIT 11 - HEADER CRC (WITH BIT 10 SET)
- DATA CRC (WITH BIT 10 CLEAR)
BIT 10 - OPERATION INCOMPLETE
BIT 9/8 - DRIVE SELECT (0-3)
BIT 7 - CONTROLLER READY
BIT 6 - INTERRUPT ENABLE
BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
BIT 3-1 - FUNCTION CODE
0 - NOP (PDP-11) MAINT (LSI-11)
1 - WRITE CHECK
2 - GET DRIVE STATUS
3 - SEEK
4 - READ HEADER
5 - WRITE DATA
6 - READ DATA
7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15-7 - 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 COMPLEMENT)

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 BASIC INTERFACE TEST (PART 1)

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

SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

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

CHECK WRITE LOCK STATUS BIT SET. IF NOT:
BAD SWITCH OR WRITE LOCK LOGIC
DRIVE COMMAND SHIFT REGISTER

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

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

CHECK DRIVE ERROR RESET. IF NOT:

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

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

TEST 2 BASIC INTERFACE TEST (PART 2)

REQUEST OPERATOR TO CLOSE COVER AND RESET WRITE LOCK.

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

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

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

IF NEITHER CHANGED, REPORT BOTH FAILURES.

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

TEST 3 HEAD LOADING TEST

(P-CLOCK REQUIRED)

REQUEST OPERATOR TO PRESS LOAD SWITCH.

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

IF LOAD WAS PRESSED:

BAD STATE ROM
BAD LOAD SWITCH OR LOGIC

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

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

AND CHECK IF SPINUP TIMEOUT ERROR SET. IF NOT:

BAD STATE ROM
BAD TIMEOUT DETECTION LOGIC

CHECK THAT STATE GOES TO 2. IF NOT:

BAD STATE ROM

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

BAD BRUSH HOME SWITCH OR LOGIC
BAD BRUSH MOTOR (AC SERVO)

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

BAD AC SERVO
BAD SWITCH OR LATCH

CHECK THAT STATE HAS CHANGED TO 3. IF NOT:

BAD STATE ROM

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

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

CHECK VOLUME CHECK IS SET. IF NOT:

BAD VOLUME CHECK LOGIC

CHECK IF DRIVE ERROR IS SET. IF NOT:

BAD DRIVE ERROR LOGIC OR INTERFACE

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

STATE ROM BAD
SEEK ROM
VEL ROM
GUARD BAND DETECTION

WAIT 15 MS FOR STATE TO CHANGE TO 5.

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

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

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

TEST 4 HEAD UNLOADING TEST

(P-CLOCK REQUIRED)

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

REQUEST OPERATOR TO UNLOAD DRIVE.

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

BAD STATE ROM
BAD SWITCH

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

BAD STATE ROM

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

NO BRAKING
BAD AC SERVO

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

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

TEST 5 DRIVE SELECT TEST

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

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

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

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

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

TEST 6 DRIVE SELECT ERROR TEST

(P-CLOCK REQUIRED)

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

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

SELECT TO A DIFFERENT NUMBER AND BACK AGAIN. DRIVE SELECT ERROR SHOULD SET AGAIN.

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

DRIVE SELECT ERROR DETECTION IS BAD IN THAT DRIVE.

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

TEST 7 INITIAL STATE TEST

(P-CLOCK REQUIRED)

INSTRUCT OPERATOR TO GO THROUGH A LOAD HEADS CYCLE TO INITIALIZE THE TEST.

DO GET STATUS, WAIT FOR INTERRUPT.

IF OPI OCCURS:

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

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

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

DRIVE CYCLE UP DID NOT SELECT HEAD 0

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

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

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

CHECK ALL ERROR BITS ARE 0.

CHECK STATE IS 5. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD

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

TEST 8 INITIAL RESET STATE TEST

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

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

RESET DETECTION, RESET ERROR, OR VOLUME CHECK FLOP BAD
DRIVE COMMAND SHIFT REGISTER BAD

HEAD SELECTED BIT SHOULD STILL BE ZERO. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
HEAD SELECT SHIFT REGISTER NOT LOADING

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

TEST 9 DRIVE READY TEST

(P-CLOCK REQUIRED)

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

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

CHECK DRIVE READY IS RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

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

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

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

VERIFY HEAD SELECT IS ZERO.

TEST 10 SEEK SIGN SWITCH TEST

(P-CLOCK REQUIRED)

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

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

VERIFY DRIVE IS NOT READY

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

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

VERIFY DRIVE ERROR DID NOT SET

VERIFY HEAD SELECT IS ZERO.

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

TEST 11 HEAD ALIGNMENT SUPPORT ROUTINE

THIS TEST IS EXECUTED WHEN HEAD ALIGNMENT SUPPORT IS REQUESTED, AND IN THE FIRST PASS ONLY.

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

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

TEST 12 HEAD SWITCHING TEST

(P-CLOCK REQUIRED)

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

DIFFERENCE COUNTER IS PICKING UP BITS
ASSOCIATED CIRCUITRY IS BAD

VERIFY DRIVE READY RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

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

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

VERIFY DRIVE ERROR DID NOT SET.

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

HEAD SELECT REGISTER BAD
DRIVE COMMAND SHIFT REGISTER BAD

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

TEST 13 READ HEADER TEST (PART 1)

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

DO READ HEADER, WAIT FOR INTERRUPT.

CHECK IF HEADER CRC ERROR SET. IF SET:

READ/WRITE BOARD BAD
READ DATA LINE BAD

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

HEADS ARE SWITCHED (CABLE)
HEAD SELECT LOGIC

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

REPEAT TESTS USING HEAD 1.

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

HEADS ARE MISALIGNED

TEST 14 READ HEADER TEST (PART 2)

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

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

READ.

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

BAD READ/WRITE BOARD
BAD PACK

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

TEST 15 DIFFERENCE OF 1 SEEK TEST (PART 1)

(P-CLOCK REQUIRED)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.
DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT.
DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 4. IF NOT:

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

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

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

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

FAILURE IN READY LATCH OR INTEGRATOR

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

FAILURE IN INTEGRATOR
UNEXPECTED GUARD BAND DETECTED

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

REPEAT TEST USING HEAD 1.

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

TEST 16 DIFFERENCE OF 1 SEEK TEST (PART 2)

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

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

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

COUNT LOGIC BAD
INTEGRATOR FAILED

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

SEEK ROM FAILED

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

REPEAT TEST USING HEAD 1.

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

@

```
     1            000001                  PART1==1
     2                                            .NLIST  CND
     3                                            .ENABLE ABS
     4                                            .ENABLE AMA
     5            002000                          .=2000
     6                                            .MCALL  SVC
     7
     8  002000                                    SVC
     9            000001                          SVCTST=1
    10            000001                          SVCSUB=1
    11            000001                          SVCBGL=1
    12            000000                          SVCINS=0
    13            000000                          SVCTAG=0
    14  002000                                    POINTER BGNSW,BGNSFT,BGNDU
    15
    16  002000                        BGNMOD      MDHEDR
    18  002000                        HEADER      CZRLI,C,0,1,0
(4) 002000      103                               .ASCII /C/
(4) 002001      132                               .ASCII /Z/
(4) 002002      122                               .ASCII /R/
(4) 002003      114                               .ASCII /L/
(4) 002004      111                               .ASCII /I/
(6) 002005      000                               .BYTE   0
(6) 002006      000                               .BYTE   0
(5) 002007      000                               .BYTE   0
(4) 002010      103                               .ASCII  /C/
(4) 002011      060                               .ASCII  /0/
(4) 002012  000000                                .WORD   0
(4) 002014  000001                                .WORD   1
(4) 002016  037574                                .WORD   L$HARD
(4) 002020  037750                                .WORD   L$SOFT
(4) 002022  014164                                .WORD   L$HW
(4) 002024  014202                                .WORD   L$SW
(4) 002026  040162                                .WORD   L$LAST
(4) 002030  000000                                .WORD   0
(4) 002032  000000                                .WORD   0
(4) 002034  000000                                .WORD   0
(4) 002036  000000                                .WORD   0
(4) 002040  014220                                .WORD   L$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  002212                                .WORD   L$DVTYP
(4) 002062  000000                                .WORD   0
(4) 002064  000000                                .WORD   0
(4) 002066  000000                                .WORD   0
(4) 002070  000000                                .WORD   0
(4) 002072  016046                                .WORD   L$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  014266                          .WORD   L$INIT
    (4)  002106  015650                          .WORD   L$CLEAN
    (4)  002110  015322                          .WORD   L$AUTO
    (4)  002112  014260                          .WORD   L$PROT
    (4)  002114  000000                          .WORD   0
    (4)  002116  000000                          .WORD   0
    (4)  002120  000000                          .WORD   0
     23  002122                          ENDMOD
     24
     25  002122                          DESCRIPT        <CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC>
    (3)  002122  055103  046122  020111          .ASCIZ  /CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC/
    (3)  002130  042524  052123  020123
    (3)  002136  044124  020105  046122
    (3)  002144  030460  030055  020062
    (3)  002152  047111  042524  043122
    (3)  002160  041501  020105  047101
    (3)  002166  020104  040502  044523
    (3)  002174  020103  051104  053111
    (3)  002202  020105  047514  044507
    (3)  002210  000103
    (2)                                  .EVEN
     26
     27  002212                          DEVTYP  <RL01,RL02>
    (3)  002212  046122  030460  051054          .ASCIZ  /RL01,RL02/
    (3)  002220  030114  000062
    (2)                                  .EVEN
     28
     29                                  ;COPYRIGHT (C) 1979
     30                                  ;THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
     31                                  ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
     32                                  ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
     33                                  ;SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
     34                                  ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
     35                                  ;FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
     36                                  ;LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
     37                                  ;AT ALL TIMES REMAIN IN DEC.
     38                                  ;
     39                                  ;THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
     40                                  ;WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
     41                                  ;BY DIGITAL EQUIPMENT CORPORATION.
     42                                  ;
     43                                  ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
     44                                  ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     45
```

```
                        .SBTTL  BIT AND OFFSET DEFINITIONS

002224                  BGNMOD  GLBEQAT

002224                  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                 UAM==      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
 52
 53                                 ;       OFFSETS FOR HARDWARE P-TABLE
 54          000000                 CSR     =0                      ;BUS ADDRESS
 55          000002                 VECT    =2                      ;VECTOR ADDRESS
 56          000004                 PRIOR   =4                      ;PRIORITY
 57          000006                 TYPDR   =6                      ;DRIVE TYPE
 58          000010                 DRSB    =10                     ;DRIVE SELECT
 59          000012                 CNT     =12                     ;CONTROLLER TYPE
 60
 61                                 ;       OFFSETS FOR SOFTWARE P-TABLE
 62          000000                 MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
 63          000002                 LOLIM   =2                      ;CYLINDER LOWER LIMIT
 64          000004                 HILIM   =4                      ;CYLINDER HIGH LIMIT
 65          000006                 HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
 66          000010                 ERLIM   =10                     ;ERROR LIMIT
 67          000012                 DCLIM   =12                     ;DATA COMPARE ERROR LIMIT
 68
 69                                 ;       BIT ASSIGNMENTS FOR SOFTWARE P-TABLE SWITCHES
 70          000001                 ALLCYL  =BIT00                  ;USE ALL CYLINDERS
 71          000002                 ALLSEC  =BIT01                  ;USE ALL SECTORS
 72          000004                 DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
 73          000010                 HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
 74          010000                 HEADLM  =BIT12                  ;HEAD LIMIT SPECIFIED FLAG
 75          020000                 HICYL   =BIT13                  ;HI LIMIT SPECIFIED FLAG
 76          040000                 LOCYL   =BIT14                  ;LO LIMIT SPECIFIED
 77          100000                 MITEST  =BIT15                  ;EXECUTE MANUAL INTERVENTION TESTS
 78
 79                                 ;       SUBSYSTEM FUNCTIONS
 80          000102                 CKDATA  =102                    ;WRITE CHECK
 81          000104                 GTSTAT  =104                    ;GET STATUS
 82          000106                 SEEK    =106                    ;SEEK
 83          000110                 RDHEAD  =110                    ;READ HEADER
 84          000112                 WTDATA  =112                    ;WRITE DATA
 85          000114                 RDDATA  =114                    ;READ DATA
 86          000116                 RDNOHR  =116                    ;READ DATA, IGNORE HEADERS
 87          000100                 NOOP    =100                    ;NO OPERATION
 88
 89                                 ;       OPERATION FLAGS
 90          007777                 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
003760                  MQUALS  =OUTINS!INOUTS!FOLWRT!REVSKS!FWDSKS!REVSKO!FWDSKO
                                                         ;MESSAGE QUALIFIER BITS

                        ;       ERROR FLAGS FROM SUBROUTINES
000001                  TOSLOW  =BIT00                   ;OPERATION TOOK TOO LONG
000002                  NOIRPT  =BIT01                   ;NO INTERRUPT FROM OPERATION
000004                  CONHNG  =BIT02                   ;CONTROLLER HUNG
000010                  NOCLR   =BIT03                   ;BAD CONTROLLER CLEAR

000000                  RLCS    =0                       ;CONTROL AND STATUS REGISTER
000002                  RLBA    =2                       ;BUS ADDRESS REGISTER
000004                  RLDA    =4                       ;DISK ADDRESS REGISTER
000006                  RLMP    =6                       ;MULTI-PURPOSE REGISTER

                        ;       REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
000000                  RLCSR   =0                       ;CONTROL AND STATUS REGISTER
100000                  ANYERR  =100000                  ;ANY ERROR BIT
040000                  DRVERR  =40000                   ;DRIVE ERROR BIT
020000                  NXMERR  =20000                   ;NON-EXISTENT MEMORY ERROR
010000                  DLTERR  =10000                   ;DATA LATE ERROR
010000                  HNFERR  =10000                   ;HEADER NOT FOUND ERROR
004000                  DCKERR  =4000                    ;DATA CHECK ERROR
004000                  HCRCERR =4000                    ;HEADER CHECK ERROR
002000                  OPIERR  =2000                    ;OPERATION INCOMPLETE ERROR
001400                  DSMSK   =1400                    ;DRIVE SELECT MASK
000200                  CRDYMSK =200                     ;CONTROLLER READY MASK
000100                  INTEBL  =100                     ;INTERRUPT ENABLE MASK
000060                  BAMSK   ~60                      ;BUS ADDRESS UPPER MASK
000001                  DRDYMSK =1                       ;DRIVE READY MASK
```

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

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

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

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

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

                                ;       REGISTER BIT DEFINITIONS - MP FOR GET STATUS
000007                          STAMSK  =7                      ;STATE MASK
000010                          BHSTAT  =10                     ;BRUSH HOME STATUS
000020                          HOSTAT  =20                     ;HEADS OUT STATUS
000040                          COSTAT  =40                     ;COVER OPEN STATUS
000100                          HSSTAT  =100                    ;HEAD SELECT STATUS
000400                          DSESTAT =400                    ;DRIVE SELECT ERROR STATUS
001000                          VCSTAT  =1000                   ;VOLUME CHECK STATUS
002000                          WGESTAT =2000                   ;WRITE GATE ERROR STATUS
004000                          SPDSTAT =4000                   ;SPIN ERROR STATUS
010000                          STOSTAT =10000                  ;SEEK TIMEOUT ERROR STATUS
020000                          WLSTAT  =20000                  ;WRITE LOCK STATUS
040000                          HCESTAT =40000                  ;HEAD CURRENT ERROR STATUS
100000                          WDESTAT =100000                 ;WRITE DATA ERROR STATUS

002224                          ENDMOD
```

```
                        .SBTTL  MACRO DEFINITIONS

                        ;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MILLISECOND TIME COUNTS.
                        ;THIS TIMING IS PERFORMED BY SOFTWARE USING CPU TIMING AND IS HIGHLY MACHINE
                        ;DEPENDENT.
                        .MACRO  WAITMS  ARG,?WAIT
                                MOV     #ARG,DLYCNT     ;INITIALIZE DELAY COUNTER
                                ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
                                ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
                        WAIT:   DELAY   #250.           ;IMPLEMENT 25-MS TIME DELAY
                                DEC     DLYCNT          ;DECREMENT DELAY COUNT
                                BNE     WAIT            ;BRANCH IF TIME DELAY NOT EXPIRED
                        .ENDM

                        ;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS.
                        ;THIS TIMING IS PERFORMED BY SOFTWARE USING CPU TIMING AND IS HIGHLY MACHINE
                        ;DEPENDENT.
                        .MACRO  WAITUS  ARG
                                DELAY   #ARG            ;IMPLEMENT 100-US TIME DELAY
                        .ENDM

                        ;DELAY EXECUTION OF PROGRAM A SPECIFIED NUMBER OF 100-MICROSECOND TIME COUNTS
                        ;USING A KW11-P PROGRAMMABLE CLOCK.
                        .MACRO  TIMDLY  ARG,?WAIT
                                SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
                                MOV     #ARG,DLYCNT     ;INITIALIZE DELAY COUNT
                                MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
                                                        ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
                                MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
                                                        ;/10 KHZ RATE,START THE CLOCK
                        WAIT:   TST     DLYCNT          ;DELAY COUNT EXPIRED?
                                BNE     WAIT            ;BRANCH IF TIME NOT ELAPSED
                                CLR     @#172540        ;STOP THE CLOCK
                        .ENDM
```

```
                                .SBTTL  GLOBAL DATA AND CONSTANTS

002224                          BGNMOD  GLBDAT

                                ;       TABLE OF OPERATION MESSAGES
002224  000000  OPMSGS: .WORD   0               ;FILLER
002226  005267         .WORD   MWRCHK           ;MESSAGE FOR WRITE CHECK
002230  005313         .WORD   MGTSTA           ;             GET STATUS
002232  005240         .WORD   MSEEK            ;             SEEK
002234  005257         .WORD   MREADH           ;             READ HEADER
002236  005301         .WORD   MWRITE           ;             WRITE DATA
002240  005246         .WORD   MREAD            ;             READ DATA
002242  005376         .WORD   MWRSET           ;             WITH RESET
002244  005325         .WORD   MDATCP           ;             WITH DATA COMPARE
002246  005344         .WORD   MHDRCP           ;             WITH HEADER COMPARE
002250  005443         .WORD   MCYLUP           ;             LOAD HEADS
002252  005432         .WORD   MULOAD           ;             UNLOAD HEADS
002254  005474         .WORD   MINOUT           ;                IN-OUT SEQ
002256  005453         .WORD   MOUTIN           ;                OUT-IN SEQ
002260  005517         .WORD   MFOLWRT          ;                FOLLOWING WRITE
002262  005541         .WORD   MREVSK           ;                REV SEEK
002264  005574         .WORD   MFWDSK           ;                FWD SEEK
002266  005663         .WORD   MRESKO           ;                REV SEEK
002270  005627         .WORD   MFWSKO           ;                FWD SEEK
002272  005717         .WORD   MBADAD           ;                BAD DISK ADD FOR WRITE
002274  005362         .WORD   M40HDR           ;             40 HEADER OPERATION
002276  000000  T.DRIVE: .WORD  0
002300  000000  JJJ:    .WORD   0
002302  000000  HLMTW:  .WORD   0
002304  000000  CLRBYT: .WORD   0
002306  000000  NXTHL:  .WORD   0
002310  000000  GBND:   .WORD   0
002312  000000  CAMSK:  .WORD   0
002314  000000  DIRMSK: .WORD   0
002316  000000  HDCYL:  .WORD   0

                ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
002320  010271  RESTBL: .WORD   MCERR            ;CONTROLLER ERROR
002322  010402         .WORD   MDRERR           ;DRIVE ERROR
002324  010715         .WORD   MNEERR           ;NON-EXISTENT MEMORY ERROR
002326  010667         .WORD   MFLERR           ;HEADER NOT FOUND-DATA LATE
002330  010652         .WORD   MHDERR           ;HEADER OR DATA ERROR
002332  010642         .WORD   MOPERR           ;OPERATION INCOMPLETE
002334  010733         .WORD   MNDRST           ;NO DRIVE STATUS AVAILABLE
002336  000000         .WORD   0
002340  010625         .WORD   MWDERR           ;WRITE DATA ERROR
002342  010607         .WORD   MHCERR           ;HEAD CURRENT ERROR
002344  000000         .WORD   0
002346  010571         .WORD   MSTERR           ;SEEK TIMEOUT ERROR
002350  010536         .WORD   MSPERR           ;SPINDLE ERROR
002352  010554         .WORD   MWGERR           ;WRITE GATE ERROR
002354  000000         .WORD   0
002356  010506         .WORD   MDSERR           ;DRIVE SELECT ERROR
```

```
                                ;       PATTERN TABLE
002360  004762                  PATTBL: .WORD   PAT1
002362  004764                          .WORD   PAT2
002364  005024                          .WORD   PAT3
002366  005064                          .WORD   PAT4
002370  005124                          .WORD   PAT5
002372  005132                          .WORD   PAT6
002374  005172                          .WORD   PAT7
002376  005174                          .WORD   PAT8
002400  005234                          .WORD   PAT9
002402  005236                          .WORD   PAT10


                                ;       SUBROUTINE CALLING STACK
002404  000000                  SUBSTK: .WORD   0               ;STACK IS 12 WORDS LONG
002406  000000                          .WORD   0
002410  000000                          .WORD   0
002412  000000                          .WORD   0
002414  000000                          .WORD   0
002416  000000                          .WORD   0
002420  000000                          .WORD   0
002422  000000                          .WORD   0
002424  000000                          .WORD   0
002426  000000                          .WORD   0

                                ;RL01 TABLE OF CYLINDERS
002430  000002                  T25TBL: .WORD   2               ;TABLE OF DIFFERENCES
002432  000006                          .WORD   6
002434  000011                          .WORD   9.
002436  000014                          .WORD   12.
002440  000021                          .WORD   17.
002442  000026                          .WORD   22.
002444  000033                          .WORD   27.
002446  000042                          .WORD   34.
002450  000051                          .WORD   41.
002452  000200                          .WORD   128.
002454  000377                          .WORD   255.

                                ;RL02 TABLE OF CYLINDERS
002456  000004                  T25TB2: .WORD   4
002460  000014                          .WORD   12.
002462  000022                          .WORD   18.
002464  000030                          .WORD   24.
002466  000042                          .WORD   34.
002470  000054                          .WORD   44.
002472  000066                          .WORD   54.
002474  000104                          .WORD   68.
002476  000122                          .WORD   82.
002500  000400                          .WORD   256.
002502  000777                          .WORD   511.

                                ;       TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS

002504  000020                  T33TBL: .BLKW   16.
002544  000020                  TBT:    .BLKW   16.
```

```
002604      002             CYLTBL: .BYTE   2               ;TABLE OF DEFAULT CYLINDERS
002605      007                     .BYTE   7.
002606      016                     .BYTE   14.
002607      024                     .BYTE   20.
002610      033                     .BYTE   27.
002611      041                     .BYTE   33.
002612      046                     .BYTE   38.
002613      055                     .BYTE   45.
002614      064                     .BYTE   52.
002615      072                     .BYTE   58.
002616      101                     .BYTE   65.
002617      110                     .BYTE   72.
002620      115                     .BYTE   77.
002621      124                     .BYTE   84.
002622      133                     .BYTE   91.
002623      141                     .BYTE   97.
002624      146                     .BYTE   102.
002625      154                     .BYTE   108.
002626      161                     .BYTE   113.
002627      170                     .BYTE   120.
002630      177                     .BYTE   127.
002631      206                     .BYTE   134.
002632      213                     .BYTE   139.
002633      222                     .BYTE   146.
002634      230                     .BYTE   152.
002635      235                     .BYTE   157.
002636      244                     .BYTE   164.
002637      252                     .BYTE   170.
002640      261                     .BYTE   177.
002641      270                     .BYTE   184.
002642      275                     .BYTE   189.
002643      303                     .BYTE   195.
002644      312                     .BYTE   202.
002645      317                     .BYTE   207.
002646      326                     .BYTE   214.
002647      334                     .BYTE   220.
002650      343                     .BYTE   227.
002651      352                     .BYTE   234.
002652      361                     .BYTE   241.
002653      367                     .BYTE   247.
002654      375                     .BYTE   253.
002655      000                     .BYTE   0
002656   000401                     .WORD   257.
002660   000406                     .WORD   262.
002662   000415                     .WORD   269.
002664   000423                     .WORD   275.
002666   000432                     .WORD   282.
002670   000445                     .WORD   293.
002672   000454                     .WORD   300.
002674   000463                     .WORD   307.
002676   000471                     .WORD   313.
002700   000500                     .WORD   320.
002702   000507                     .WORD   327.
002704   000514                     .WORD   332.
```

```
  384  002706  000523                    .WORD   339.
  385  002710  000532                    .WORD   346.
  386  002712  000540                    .WORD   352.
  387  002714  000545                    .WORD   357.
  388  002716  000553                    .WORD   363.
  389  002720  000560                    .WORD   368.
  390  002722  000567                    .WORD   375.
  391  002724  000576                    .WORD   382.
  392  002726  000605                    .WORD   389.
  393  002730  000612                    .WORD   394.
  394  002732  000621                    .WORD   401.
  395  002734  000627                    .WORD   407.
  396  002736  000634                    .WORD   412.
  397  002740  000643                    .WORD   419.
  398  002742  000651                    .WORD   425.
  399  002744  000660                    .WORD   432.
  400  002746  000667                    .WORD   439.
  401  002750  000674                    .WORD   444.
  402  002752  000702                    .WORD   450.
  403  002754  000711                    .WORD   457.
  404  002756  000716                    .WORD   462.
  405  002760  000725                    .WORD   469.
  406  002762  000733                    .WORD   475.
  407  002764  000742                    .WORD   482.
  408  002766  000751                    .WORD   489.
  409  002770  000760                    .WORD   496.
  410  002772  000766                    .WORD   502.
  411  002774  000774                    .WORD   508.
  412  002776  000774                    .WORD   508.
  413  003000  000000                    .WORD   0
  414  003002  000000          SSINDX:   .WORD   0             ;SUBROUTINE STACK INDEX POINTER
  415
  416                           ;         OPERATIONAL FLAGS
  417  003004  000000          OPFLAG:   .WORD   0             ;OPERATION FLAGS
  418  003006  000000          DONE:     .WORD   0             ;OPERATION COMPLETE FLAG
  419  003010  000000          HADONE:   .WORD   0             ;HEAD ALIGNMENT DONE FLAG
  420  003012  000000          ERHEAD:   .WORD   0             ;ADDRESS OF ERROR HEADER
  421  003014  000000          MORECE:   .WORD   0             ;MORE THAN 1 COMPARE ERROR
  422  003016  000000          ERRSWI:   .WORD   0             ;ERROR RETURN SWITCH
  423  003020  000000          BSFLAG:   .WORD   0             ;BAD SECTOR FLAGS
  424  003022  000000          WRTSWI:   .WORD   0             ;WRITE SWITCH
  425  003024  000000          TBLSTR:   .WORD   0             ;TABLE STORAGE
  426
  427  003026  000000          RLBAS:    .WORD   0             ;RL11 BASE ADDRESS
  428  003030  000000          RLVEC:    .WORD   0             ;RL11 VECTOR ADDRESS
  429  003032  000000          RLDRV:    .WORD   0             ;DRIVE NUMBER UNDER TEST
  430
  431  003034  000000          L.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
  432  003036  000000          L.BA:     .WORD   0             ;BEFORE OPERATION
  433  003040  000000          L.DA:     .WORD   0
  434  003042  000000          L.MP:     .WORD   0
  435  003044  000000          T.CS:     .WORD   0             ;CONTROLLER REGISTER STORAGE
  436  003046  000000          T.BA:     .WORD   0             ;   AFTER OPERATION
  437  003050  000000          T.DA:     .WORD   0
  438  003052                    T.MP:
  439  003052  000000          HDWRD1:   .WORD   0             ;HEADER WORD STORAGE
```

```
   440  003054  000000                HDWRD2: .WORD   0
   441  003056  000000                HDWRD3: .WORD   0
   442
   443  003060  000000                T.STAT: .WORD   0               ;DRIVE STATE STORAGE
   444
   445  003062  000000                RESPARM: .WORD  0               ;PARAM BLOCK FOR REASON REPORT
   446  003064  000000                        .WORD   0
   447  003066  000000                        .WORD   0
   448  003070  000000                        .WORD   0
   449  003072  000000                        .WORD   0
   450
   451  003074  000000                DRVCNT: .WORD   0               ;DRIVE COUNT FOR DRIVES UNDER TEST
   452  003076  000000                DIFAUG: .WORD   0               ;DIFFERENCE ARGUMENT FOR SEEK
   453  003100  000000                OLDCYL: .WORD   0               ;OLD CYLINDER
   454  003102  000000                NEWCYL: .WORD   0               ;NEW CYLINDER
   455  003104  000000                CURCYL: .WORD   0               ;CURRENT CYLINDER
   456  003106  000000                DESDIF: .WORD   0               ;DESIRED DIFFERENCE
   457  003110  000000                DESSGN: .WORD   0               ;DESIRED SIGN
   458  003112  000000                DESHD:  .WORD   0               ;DESIRED HEAD
   459  003114  000000                DESSEC: .WORD   0               ;DESIRED SECTOR
   460  003116  000000                TEMP0:  .WORD   0               ;TEMPORARY STORAGE
   461  003120  000000                TEMP1:  .WORD   0               ;TEMPORARY STARAGE
   462  003122  000000                TEMP2:  .WORD   0               ;TEMPORARY STORAGE
   463  003124  000000                TEMP3:  .WORD   0               ;TEMPORARY STORAGE
   464  003126  000000                TEMP4:  .WORD   0               ;TEMPORARY STORAGE
   465  003130  000000                TEMP5:  .WORD   0               ;TEMPORARY STORAGE
   466  003132  000000                TEMP6:  .WORD   0               ;TEMPORARY STORAGE
   467  003134  000000                TEMP7:  .WORD   0               ;TEMPORARY STORAGE
   468  003136  000000                TEMP8:  .WORD   0               ;TEMPORARY STORAGE
   501  003140  000004                ERRVEC: .WORD   4               ;ERROR VECTOR
   502  003142  000000                DLYCNT: .WORD   0               ;DELAY COUNTER USED IN TIMING MACROS
   503  003144  000000                CLKFLG: .WORD   0               ;FLAG INDICATING PRESENCE OF A P-CLOCK
   504  003146  000000                CLKADR: .WORD   0               ;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
   505
   506                                ;       MISCELLANEOUS COUNTERS
   507  003150  000000                PASCNT: .WORD   0               ;PASS COUNTER (LOCAL TO A TEST)
   508  003152  000000                COUNT:  .WORD   0               ;A COUNTER (LOCAL TO A TEST)
   509  003154  000000                ERRPOINT: .WORD 0               ;ERROR POINTER
   510  003156  000100                ERRCNT: .BLKW   64.             ;ERROR COUNTER FOR PROGRAM
   511  003356  000000                PASNUM: .WORD   0               ;PASS NUMBER FOR PROGRAM
   512  003360  000000                PSETNM: .WORD   0               ;COUNTER FOR PARAMETER SET NUMBER IN USE
   513  003362     000                LOCERR: .BYTE   0               ;LOCAL ERROR COUNTER
   514  003363     000                NOERCT: .BYTE   0               ;INHIBIT ERROR COUNTING FLAG
   515  003364  000000                TRPFLG: .WORD   0               ;HARDWARE TRAP FLAG
   516  003366  000000                PWRFLG: .WORD   0               ;POWER FAILURE FLAG
   517
   518                                ;       BAD SECTOR TABLES AND POINTERS
   519  003370  000000                BSFVAL: .WORD   0               ;BAD SECTORS FILES VALID FLAG
   520
   521  003372  000076                SBSFIL: .BLKW   76              ;SOFTWARE BAD SECTOR FILE
   522  003566  000076                FBSFIL: .BLKW   76              ;FACTORY BAD SECTOR FILE
   523
   524  003762  000200                IBUFF:  .BLKW   200             ;INPUT BUFFER
   525  004362  000200                OBUFF:  .BLKW   200             ;OUTPUT BUFFER
   526
   527  004762  000000                PAT1:   .WORD   0               ;PATTERN 1 (ALL ZEROS)
```

```
  528  004764  177772          PAT2:   .WORD   177772
  529  004766  177777                  .WORD   177777
  530  004770  177777                  .WORD   177777
  531  004772  052525                  .WORD   052525
  532  004774  052525                  .WORD   052525
  533  004776  052525                  .WORD   052525
  534  005000  177777                  .WORD   177777
  535  005002  177777                  .WORD   177777
  536  005004  052525                  .WORD   052525
  537  005006  052525                  .WORD   052525
  538  005010  177777                  .WORD   177777
  539  005012  052525                  .WORD   052525
  540  005014  177252                  .WORD   177252
  541  005016  177252                  .WORD   177252
  542  005020  172765                  .WORD   172765
  543  005022  172765                  .WORD   172765
  544
  545  005024  000003          PAT3:   .WORD   000003
  546  005026  000000                  .WORD   000000
  547  005030  000000                  .WORD   000000
  548  005032  177777                  .WORD   177777
  549  005034  177777                  .WORD   177777
  550  005036  177777                  .WORD   177777
  551  005040  000000                  .WORD   000000
  552  005042  000000                  .WORD   000000
  553  005044  177777                  .WORD   177777
  554  005046  177777                  .WORD   177777
  555  005050  000000                  .WORD   000000
  556  005052  177777                  .WORD   177777
  557  005054  000000                  .WORD   000000
  558  005056  177777                  .WORD   177777
  559  005060  000000                  .WORD   000000
  560  005062  177777                  .WORD   177777
  561
  562  005064  025252          PAT4:   .WORD   025252
  563  005066  052525                  .WORD   052525
  564  005070  052525                  .WORD   052525
  565  005072  125252                  .WORD   125252
  566  005074  125252                  .WORD   125252
  567  005076  125252                  .WORD   125252
  568  005100  052525                  .WORD   052525
  569  005102  052525                  .WORD   052525
  570  005104  125252                  .WORD   125252
  571  005106  125252                  .WORD   125252
  572  005110  052525                  .WORD   052525
  573  005112  125252                  .WORD   125252
  574  005114  052525                  .WORD   052525
  575  005116  125252                  .WORD   125252
  576  005120  052525                  .WORD   052525
  577  005122  125252                  .WORD   125252
  578
  579  005124  155555          PAT5:   .WORD   155555
  580  005126  133333                  .WORD   133333
  581  005130  066666                  .WORD   066666
  582
  583  005132  121105          PAT6:   .WORD   121105
```

```
                                                .WORD   150442
                                                .WORD   064221
                                                .WORD   132110
                                                .WORD   055044
                                                .WORD   026442
                                                .WORD   013211
                                                .WORD   105504
                                                .WORD   042642
                                                .WORD   021321
                                                .WORD   110550
                                                .WORD   044264
                                                .WORD   022132
                                                .WORD   011055
                                                .WORD   104426
                                                .WORD   042213

005172  177777                        PAT7:   .WORD   177777

005174  045513                        PAT8:   .WORD   045513
005176  122645                                .WORD   122645
005200  151322                                .WORD   151322
005202  064551                                .WORD   064551
005204  132264                                .WORD   132264
005206  055132                                .WORD   055132
005210  026455                                .WORD   026455
005212  113226                                .WORD   113226
005214  045513                                .WORD   045513
005216  122645                                .WORD   122645
005220  151322                                .WORD   151322
005222  064551                                .WORD   064551
005224  132264                                .WORD   132264
005226  055132                                .WORD   055132
005230  026455                                .WORD   026455
005232  113226                                .WORD   113226

005234  125252                        PAT9:   .WORD   125252

005236  155555                        PAT10:  .WORD   155555

005240                                ENDMOD


                                      .SBTTL  GLOBAL MESSAGES

005240                                BGNMOD  GLBTXT
005240  042523  045505  000040        MSEEK:  .ASCIZ  /SEEK /
005246  042122  042040  052101        MREAD:  .ASCIZ  /RD DATA /
005257     122  020104  042110        MREADH: .ASCIZ  /RD HDR /
005267     127  052122  041440        MWRCHK: .ASCIZ  /WRT CHECK/
005301     127  052122  042040        MWRITE: .ASCIZ  /WRT DATA /
005313     107  052105  051440        MGTSTA: .ASCIZ  /GET STAT /
005325     127  052111  020110        MDATCP: .ASCIZ  /WITH DATA CMP /
005344  044527  044124  044040        MHDRCP: .ASCIZ  /WITH HDR CMP /
005362  047506  020122  030064        M40HDR: .ASCIZ  /FOR 40 HDRS/
005376  044527  044124  051040        MWRSET: .ASCIZ  /WITH RESET /
005412  050117  051105  020072        MOPER:  .ASCIZ  /OPER: /
```

```
643  005421     122  051505  046125  MRSLT:   .ASCIZ  /RESULT:        /
644  005432  047125  042114  042040  MULOAD:  .ASCIZ  /UNLD DRV/
645  005443     114  020104  051104  MCYLUP:  .ASCIZ  /LD DRV /
646  005453     106  046117  030040  MOUTIN:  .ASCIZ  /FOL 0 TO CC SEEK/
647  005474  047506  020114  032462  MINOUT:  .ASCIZ  /FOL 255 TO CC SEEK/
648  005517     106  046117  053440  MFOLWRT: .ASCIZ  /FOL WRT (NO SEEK)/
649  005541     101  045104  041440  MREVSK:  .ASCIZ  /ADJ CYL WRTTN AFTER REV SK/
650  005574  042101  020112  054503  MFWDSK:  .ASCIZ  /ADJ CYL WRTTN AFTER FWD SK/
651  005627     123  020113  053506  MFWSKO:  .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
652  005663     123  020113  042522  MRESKO:  .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
653  005717     117  020116  040502  MBADAD:  .ASCIZ  /ON BAD SEC FILES/
654  005740  040503  047116  052117  MBADSF:  .ASCIZ  /CANNOT GET BAD SEC FILES/
655  005771     102  042101  051440  MFMTER:  .ASCIZ  /BAD SEC FILE FMT ERR/
656  006016  047524  020117  040515  MTMBS:   .ASCIZ  /TOO MANY BAD SEC /
657  006040  052502  020123  042101  BASADD:  .ASCIZ  /BUS ADD=/
658  006051     104  053122  000075  DRVNAM:  .ASCIZ  /DRV=/
659  006056  051104  020126  044504  NOPWR:   .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
660  006116  046122  051503     000  CSNAM:   .ASCIZ  /RLCS/
661  006123     122  041114  000101  BANAM:   .ASCIZ  /RLBA/
662  006130  046122  040504     000  DANAM:   .ASCIZ  /RLDA/
663  006135     122  046514  000120  MPNAM:   .ASCIZ  /RLMP/
664  006142  050117  044440  044516  LAB1:    .ASCIZ  /OP INIT = /
665  006155     117  020120  047504  LAB2:    .ASCIZ  /OP DONE = /
666  006170  047527  042122  000040  MWORD:   .ASCIZ  /WORD /
667  006176  047111  051124  052120  MTOSLOW: .ASCIZ  /INTRPT TOO LATE/
668  006216  047516  042040  053122  MDRRES:  .ASCIZ  /NO DRV RESPONSE/
669  006236  047516  044440  052116  MNOINT:  .ASCIZ  /NO INTRPT ON CMND COMPLETE/
670  006271     103  052116  051114  MCONHNG: .ASCIZ  /CNTLR HUNG /
671  006305     105  051122  042040  MNOCLR:  .ASCIZ  /ERR DID NOT CLR/
672  006325     126  046117  041440  VCNRST:  .ASCIZ  /VOL CHK NOT RSET/
673  006346  047125  050130  052103  UNXERR:  .ASCIZ  /UNXPCTED ERR/
674  006363     040  042524  052123  TSTLAB:  .ASCIZ  / TEST/
676  006371     115  047101  044440  MISTST:  .ASCIZ  /MAN INTERVENT STAT/
677  006414  052123  052101  020105  NSTACHG: .ASCIZ  /STATE CHG/
678  006426  050123  042116  020114  SPDERR:  .ASCIZ  /SPNDL TIMEOUT FAILED TO SET/
679  006462  040506  046111  043040  GSTER1:  .ASCIZ  /FAIL FORCING DRV SEL ERR/
680  006513     111  044516  020124  INITST:  .ASCIZ  /INIT STATE/
681  006526  051104  020126  042523  T05ERR:  .ASCIZ  /DRV SELECT/
682  006541     104  053122  051040  T09ERR:  .ASCIZ  /DRV RDY/
683  006551     123  042505  020113  T10ERR:  .ASCIZ  /SEEK SGN SWITCH/
684  006571     110  020104  053523  T12ERR:  .ASCIZ  /HD SWITCH/
685  006603     122  020104  042110  T13ERR:  .ASCIZ  /RD HDR (P1)/
686  006617     122  020104  042110  T14ERR:  .ASCIZ  /RD HDR (P2)/
687  006633     127  052122  046040  T16ERR:  .ASCIZ  /WRT LCK/
688  006643                          P2T01E:
689  006643     104  043111  020106  P2T02E:  .ASCIZ  /DIFF OF 1 SEEK/
690  006662  051524  020124  020040  NOTST:   .ASCIZ  /TST    CANNOT BE PERFORMED...NO P-CLK/
691  006730  051104  020126  051104  NOCTLR:  .ASCIZ  /DRV DROPPED - NO CNTLR/
692  006757     104  053122  042040  NOTRDY:  .ASCIZ  /DRV DROPPED - NOT RDY/
```

```
007005     110 051504 043040 HDMOVF: .ASCIZ  /HDS FAILED TO MOVE IN 10 TRIES/
007044  054503 020114 047520 CYLPER: .ASCIZ  /CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
007130  042510 042101 040440 HAMES1: .ASCIZ  /HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
007213     124 050131 020105 HAMES2: .ASCIZ  &TYPE "CTL/C" TO GET BACK TO SUPVR CMD MODE AND THEN TYPE "CONT"   &
007317     101 047502 042526 OPR002: .ASCIZ  /ABOVE CONDITIONS MET/
007344  040527 020123 047514 OPR003: .ASCIZ  /WAS LOAD DEPRESSED/
007367     103 045510 042040 OPR1:   .ASCIZ  /CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
007445     103 051514 020105 OPR2:   .ASCIZ  /CLSE COVER & RST WRT LCK /
007477     120 042522 051523 OPR3:   .ASCIZ  /PRESS LOAD /
007513     120 042522 051523 OPR5:   .ASCIZ  /PRESS LOAD & WAIT FOR LOAD LIGHT /
007555     120 042522 051523 OPR6:   .ASCIZ  /PRESS LOAD & WAIT FOR RDY /
007610  042522 047515 042526 OPR7:   .ASCIZ  /REMOVE ADD PLGS EXCPT /
007637     111 051516 052122 OPR8:   .ASCIZ  /INSRT ADD PLG /
007656  047111 040440 046114 OPR9:   .ASCIZ  /IN ALL DRVS /
007673     111 051516 043125 OPR10:  .ASCIZ  /INSUFFICIENT DRVS FOR DRV SEL ERR TST/
007741     122 046120 042503 OPR11:  .ASCIZ  /RPLCE ADD PLGS AS BEFORE/
007772  042522 042523 020124 OPR12:  .ASCIZ  /RESET WRT LCK /
010011     117 020116    000 OPR1A:  .ASCIZ  /ON /
010015     117 020116 051104 OPR1B:  .ASCIZ  /ON DRV /
010025     125 042116 051105 UNDTST: .ASCIZ  /UNDER TEST/
010040  042523 020124 051127 OPR004: .ASCIZ  /SET WRT LCK /
010055     104 043111 020106 DIFWD:  .ASCIZ  /DIFF /
010063     123 047107 000040 SGNWD:  .ASCIZ  /SGN /
010070  042110 000040        HDWD:   .ASCIZ  /HD /
010074  042523 020103    000 SECWD:  .ASCIZ  /SEC /
010101     103 046131 000040 CYLWD:  .ASCIZ  /CYL /
010106  051106 046517 000040 FRMWD:  .ASCIZ  /FROM /
010114  041040 050131 051501 BYPSNM: .ASCIZ  / BYPASSED /
010127     122 052517 044524 SEQMES: .ASCIZ  /ROUTINE TRACE SEQ:/
010152  051104 020126 052123 STAMES: .ASCIZ  /DRV STAT/
010163     102 042101 051440 BSNSTR: .ASCIZ  /BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
010237     124 052117 046101 TCERR:  .ASCIZ  /TOTAL CMP ERRS: /

                             ;              RESULT NAMES
010260  051104 020126 042122 MDRDY:  .ASCIZ  /DRV RDY /
010271     103 047117 020124 MCERR:  .ASCIZ  /CONT ERR /
010303     110 051104 041440 MHCRC:  .ASCIZ  /HDR CRC/
010313     104 052101 020101 MDCRC:  .ASCIZ  /DATA CRC/
010324  042110 020122 047516 MHNF:   .ASCIZ  /HDR NOT FND/
010340  040504 040524 046040 MDLT:   .ASCIZ  /DATA LATE/
010352  042110 020122 047516 MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
010402  051104 020126 051105 MDRERR: .ASCIZ  /DRV ERR /
010413     123 046105 042047 MHSTA:  .ASCIZ  /SEL'D HD /
010425     126 046117 041440 MVOLCK: .ASCIZ  /VOL CHK /
010436  047503 042526 020122 MCOSTA: .ASCIZ  /COVER OPN /
010451     102 052522 044123 MBHSTA: .ASCIZ  /BRUSH HME /
010464  051127 020124 041514 MWLSTA: .ASCIZ  /WRT LCK /
010475     110 051504 047440 MHOSTA: .ASCIZ  /HDS OUT /
010506  051104 020126 042523 MDSERR: .ASCIZ  /DRV SEL ERR /
010523     104 053122 051440 MDRVST: .ASCIZ  /DRV STATE /
010536  050123 047111 052040 MSPERR: .ASCIZ  /SPIN TIMEOUT /
010554  051127 020124 040507 MWGERR: .ASCIZ  /WRT GAT ERR /
010571     123 042505 020113 MSTERR: .ASCIZ  /SEEK TIMEOUT /
010607     110 040505 020104 MHCERR: .ASCIZ  /HEAD CUR ERR /
010625     127 052122 042040 MWDERR: .ASCIZ  /WRT DAT ERR /
010642  050117 026522 047111 MOPERR: .ASCIZ  /OPR-INC/
```

```
788  010652  042110  027522  040504  MHDERR: .ASCIZ  &HDR/DAT ERR &
789  010667     110  051104  047040  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
790  010715     116  054055  046455  MNEERR: .ASCIZ  /N-X-MEM /
791  010726  054503  020114     000  MCYLOC: .ASCIZ  /CYL /
792  010733     103  047101  047516  MNDRST: .ASCIZ  /CANNOT GET DRV STAT/
793  010757     125  045516  020116  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
794  011024  040506  046111  052040  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLEAR/
795  011065     127  052122  040440  MWRTAB: .ASCIZ  /WRT ABORTED/
796  011101     040  053117  051105  MEXERS: .ASCIZ  / OVER ERR LIMIT - UNIT DROPPED /
797  011141     040  051105  047522  MERRS:  .ASCIZ  / ERROR/
798  011150  177607  000377          BELL:   .ASCIZ  <207><377><377>
799
800                                  ;       RESULT SETTINGS
801  011154  051511  000040          RESE3:  .ASCIZ  /IS /
802  011160  051440  020102     000  RESE4:  .ASCIZ  / SB /
803
804                                  ;       RESULT CONDITIONS
805  011165     040  047111  000040  RESE5:  .ASCIZ  / IN /
806  011172  047440  020106     000  RESE6:  .ASCIZ  / OF /
807  011177     123  040524  042524  STATE2: .ASCIZ  /STATE 2/
808  011207     123  040524  042524  STATE3: .ASCIZ  /STATE 3/
809  011217     123  040524  042524  STATE5: .ASCIZ  /STATE 5/
811  011227     123  042505  020113  CDRDY:  .ASCIZ  &SEEK W/O MOTION&
813  011247     061  052123  031440  C10MS:  .ASCIZ  /1ST 3 MS/
814  011260  030065  046460  000123  C500MS: .ASCIZ  /500MS/
815  011266  054503  046103  020105  CCYLUP: .ASCIZ  /CYCLE UP/
816  011277     104  052101  020101  CAFDT:  .ASCIZ  /DATA XFR/
817  011310  020065  042523  000103  C5SEC:  .ASCIZ  /5 SEC/
818
819  011316  047045  052045  047045  FMTOP1: .ASCIZ  /%N%T%N%T%T%06%S%T%01%N/
820  011345     045  022516  022524  FMTOP2: .ASCIZ  /%N%T%01%S1%T%01%N/
821  011367     045  022516  022524  FMTOP3: .ASCIZ  /%N%T%01%S1%T%T%N/
822  011410  052045  052045     000  FMT1:   .ASCIZ  /%T%T/
823  011415     045  022516  022524  FMT1.1: .ASCIZ  /%N%T%T/
824  011424  052045     000          FMT2:   .ASCIZ  /%T/
825  011427     045  000116          FMT3:   .ASCIZ  /%N/
826  011432  047045  052045  052045  FMT4:   .ASCIZ  /%N%T%T%N/
827  011443     045  022516  022524  FMT5:   .ASCIZ  /%N%T%06%S1%T%01/
828  011463     045  022516  030523  FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
829  011525     045  022516  022524  FMT7:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
830  011575     045  022516  022524  FMT8:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
831  011627     045  022516  000124  FMT9:   .ASCIZ  /%N%T/
832  011634  052045  047445  000061  FMT11:  .ASCIZ  /%T%01/
833  011642  052045  047445  000063  FMT12:  .ASCIZ  /%T%03/
834  011650  047045  051445  030461  FMT13:  .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
835  011714  047045  052045  052045  FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
836  011746  047045  051445  030461  FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
837  012002  047045  051445  022465  FMT16:  .ASCIZ  /%N%S5%06/
838  012013     045  030523  022460  FMT17:  .ASCIZ  /%S10%T%N%S11%06%N/
839  012035     045  022516  030523  FMT18:  .ASCIZ  /%N%S15%T%S5%T%S4%T%S5%T%N/
840  012067     045  022524  032123  FMT19:  .ASCIZ  /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
841  012124  052045  051445  022462  FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
842  012154  052045  051445  031061  FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
843  012177     045  022516  030523  FMT22:  .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
844  012233     045  022524  022524  FMT23:  .ASCIZ  /%T%T%T%01%N/
845  012247     045  022516  000124  FMT24:  .ASCIZ  /%N%T/
```

```
 012254  047045  042045  022462  FMT25:  .ASCIZ  /%N%D2%T/
 012264  047045  051445  022461  FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
 012310  047045  052045  042045  FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
 012327     045  022516  022524  FMT28:  .ASCIZ  /%N%T%T%T/
 012340                          ENDMOD


                                 .SBTTL  ERROR MESSAGES

 012340                          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 RESULT NAME
                                 ;               R4 POINTS TO VALUE 1
                                 ;               R5 POINTS TO VALUE 2
                                 ;               RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)

                                 ;       ERR10   COMPARE ERROR REPORT
                                 ;               R3 CONTAINS THE BAD WORD NUMBER
                                 ;               R4 POINTS TO BAD WORD
                                 ;               R5 POINTS TO GOOD WORD
                                 ;               RESULT: WORD (R3) IS (R4) SB (R5)


                                .NLIST  MD,ME
```

```
906
907  012340
908  012340  105737  003363          BGNMSG  ERR1
909  012344  001002                          TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
910  012346  005277  170602                  BNE     1$              ;YES - SKIP
911  012352  010146                          INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
912  012354  004737  023356          1$:     MOV     R1,-(SP)        ;STORE R1
913  012360  012721  000001                  JSR     PC,RPTOP        ;REPORT OPERATION
914  012364  010321                          MOV     #1,(R1)+        ;SET PARAM NUMBER
915  012366  004737  024144                  MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
916  012372  004737  024352                  JSR     PC,RPTRES       ;REPORT RESULTS
917  012376  012601                          JSR     PC,RPTREM       ;REPORT REMAINDER
918  012400  004737  016126                  MOV     (SP)+,R1        ;RESTORE R1
919  012404                                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
(3)  012404                          ENDMSG
(3)  012404                          L10000:
(3)  012404  104423                          TRAP    C$MSG
920
921  012406
922  012406  005277  170542          BGNMSG  ERR2
923  012412  010146                          INC     @ERRPOINT       ;BUMP ERROR COUNT
924  012414  004737  023356                  MOV     R1,-(SP)        ;STORE R1
925  012420  012721  000003                  JSR     PC,RPTOP        ;REPORT OPERATION
926  012424  010321                          MOV     #3,(R1)+        ;SET PARAM NUMBER
927  012426  012721  000001                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
928  012432  005021                          MOV     #1,(R1)+        ;SET IS VALUE
929  012434  004737  024144                  CLR     (R1)+           ;SET SB VALUE
930  012440  004737  024352                  JSR     PC,RPTRES       ;REPORT RESULTS
931  012444  012601                          JSR     PC,RPTREM       ;REPORT REMAINDER
932  012446  004737  016126                  MOV     (SP)+,R1        ;RESTORE R1
933  012452                                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
(3)  012452                          ENDMSG
(3)  012452                          L10001:
(3)  012452  104423                          TRAP    C$MSG
934
935  012454
936  012454  005277  170474          BGNMSG  ERR3
937  012460  010146                          INC     @ERRPOINT       ;BUMP ERROR COUNT
938  012462  004737  023356                  MOV     R1,-(SP)        ;STORE R1
939  012466  012721  000003                  JSR     PC,RPTOP        ;REPORT OPERATION
940  012472  010321                          MOV     #3,(R1)+        ;SET PARAM NUMBER
941  012474  005021                          MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
942  012476  012721  000001                  CLR     (R1)+           ;SET IS VALUE
943  012502  004737  024144                  MOV     #1,(R1)+        ;SET SB VALUE
944  012506  004737  024352                  JSR     PC,RPTRES       ;REPORT RESULTS
945  012512  012601                          JSR     PC,RPTREM       ;REPORT REMAINDER
946  012514  004737  016126                  MOV     (SP)+,R1        ;RESTORE R1
947  012520                                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
(3)  012520                          ENDMSG
(3)  012520                          L10002:
(3)  012520  104423                          TRAP    C$MSG
948
949  012522
950  012522  005277  170426          BGNMSG  ERR4
951  012526  010146                          INC     @ERRPOINT       ;BUMP ERROR COUNT
952  012530  004737  023356                  MOV     R1,-(SP)        ;STORE R1
953  012534  012721  000004                  JSR     PC,RPTOP        ;REPORT OPERATION
954  012540  010321                          MOV     #4,(R1)+        ;SET PARAM NUMBER
955  012542  012721  000001                  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
                                        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
                                        CLR     (R1)+           ;SET IS VALUE
                                        MOV     #1,(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
                                L10004:
                                        TRAP    C$MSG

                                BGNMSG  ERR6
                                        TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
                                        BNE     17$             ;YES - SKIP
                                        INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
                                17$:    MOV     R1,-(SP)        ;STORE R1
                                        MOV     R3,-(SP)        ;STORE R3
                                        MOV     R4,-(SP)        ;STORE R4
                                        MOV     R5,-(SP)        ;STORE R5
                                        JSR     PC,RPTOP        ;REPORT OPERATION
                                        MOV     #3,(R1)+        ;SET PARAM NUMBER
                                        MOV     #1,2(R1)        ;INSERT IS VALUE
                                        CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
                                        MOV     T.CS,R3         ;GET T.CS
                                        BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
                                        CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
                                        BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
                                        MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
                                        MOV     #4,R3
                                        BIS     RLDRV,R3
                                        MOV     R3,RLCS(R2)
                                        WAITUS  #10.            ;WAIT FOR CONTROLLER READY
                                        MOV     ###10.,(PC)+
                                        .WORD   0
                                        MOV     L$DLY,(PC)+
                                        .WORD   0
                                        DEC     -6(PC)
                                        BNE     .-4
                                        DEC     -22(PC)
                                        BNE     .-20
```

```
 1000  012776  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
 1001  013004  001003                          BNE     10$             ;YES - SKIP
 1002  013006  012703  001000          9$:     MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
 1003  013012  000413                          BR      2$              ;IN MESSAGE WORD AND SKIP
 1004  013014  016203  000006          10$:    MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
 1005  013020  010337  003124                  MOV     R3,TEMP3
 1006  013024  113703  003125                  MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
 1007  013030  000402                          BR      13$
 1008  013032  113703  003053          1$:     MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
 1009  013036  042703  177442          13$:    BIC     #177442,R3      ;CLEAR UNUSED BITS
 1010  013042  013704  003044          2$:     MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
 1011  013046  042704  001777                  BIC     #1777,R4        ;CLEAR UNUSED BITS
 1012  013052  050403                          BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
 1013  013054  032703  002000                  BIT     #OPIERR,R3      ;TEST IF OPI SET
 1014  013060  001442                          BEQ     115$            ;NO - SKIP
 1015  013062  032703  010000                  BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
 1016  013066  001026                          BNE     107$            ;YES - SKIP
 1017  013070  032703  004000                  BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
 1018  013074  001020                          BNE     105$            ;YES - SKIP
 1019  013076  012704  010642                  MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
 1020  013102                          100$:   PRINTB  #FMT28,#MRSLT,R4,#MERRS ;REPORT ERROR
  (10) 013102  012746  011141                  MOV     #MERRS,-(SP)
  (9)  013106  010446                          MOV     R4,-(SP)
  (8)  013110  012746  005421                  MOV     #MRSLT,-(SP)
  (7)  013114  012746  012327                  MOV     #FMT28,-(SP)
  (6)  013120  012746  000004                  MOV     #4,-(SP)
  (3)  013124  010600                          MOV     SP,R0
  (4)  013126  104414                          TRAP    C$PNTB
  (4)  013130  062706  000012                  ADD     #12,SP
 1021  013134  000430                          BR      120$            ;SKIP
 1022  013136  012704  010303          105$:   MOV     #MHCRC,R4       ;HDR CRC MESSAGE
 1023  013142  000757                          BR      100$
 1024  013144  032703  004000          107$:   BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
 1025  013150  001003                          BNE     109$            ;YES - SKIP
 1026  013152  012704  010324                  MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
 1027  013156  000751                          BR      100$
 1028  013160  012704  010352          109$:   MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
 1029  013164  000746                          BR      100$            ;SKIP
 1030  013166  032703  004000          115$:   BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
 1031  013172  001403                          BEQ     118$            ;NO - SKIP
 1032  013174  012704  010313                  MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
 1033  013200  000740                          BR      100$            ;SKIP
 1034  013202  032703  010000          118$:   BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
 1035  013206  001403                          BEQ     120$            ;NO - SKIP
 1036  013210  012704  010340                  MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
 1037  013214  000732                          BR      100$            ;SKIP
 1038  013216  012705  100000          120$:   MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
 1039  013222  005004                          CLR     R4              ;CLEAR R4 FOR TABLE COUNT
 1040  013224  030503                  3$:     BIT     R5,R3           ;TEST IF BIT IS SET
 1041  013226  001005                          BNE     6$              ;YES - SKIP TO REPORT
 1042  013230  005724                  4$:     TST     (R4)+           ;ELSE BUMP TABLE POINTER
 1043  013232  000241                          CLC                     ;CLEAR CARRY
 1044  013234  006005                          ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
 1045  013236  001372                          BNE     3$              ;LOOP IF NOT 0
 1046  013240  000405                          BR      7$              ;ELSE REPORT REMAINDER
 1047  013242  016411  002320          6$:     MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
```

```
 1048  013246  004737  024144                 JSR     PC,RPTRES       ;REPORT RESULTS
 1049  013252  000766                         BR      4$              ;GET NEXT BIT
 1050  013254  004737  024352         7$:     JSR     PC,RPTREM       ;REPORT REMAINDER
 1051  013260  005737  003124                 TST     TEMP3           ;TEST IF ANY NEW STATUS
 1052  013264  001414                         BEQ     15$             ;NO - SKIP
 1053  013266                                 PRINTB  #FMT17,#STAMES,TEMP3
  (9)  013266  013746  003124                 MOV     TEMP3,-(SP)
  (8)  013272  012746  010152                 MOV     #STAMES,-(SP)
  (7)  013276  012746  012013                 MOV     #FMT17,-(SP)
  (6)  013302  012746  000003                 MOV     #3,-(SP)
  (3)  013306  010600                         MOV     SP,R0
  (4)  013310  104414                         TRAP    C$PNTB
  (4)  013312  062706  000010                 ADD     #10,SP
 1054  013316  032737  004000  003044  15$:    BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
 1055  013324  001453                         BEQ     25$             ;NO - SKIP
 1056  013326  032737  002000  003044         BIT     #OPIERR,T.CS    ;TEST IF OPI SET
 1057  013334  001047                         BNE     25$             ;YES - SKIP
 1058  013336  005037  003014                 CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
 1059  013342  012701  000200                 MOV     #128.,R1        ;SET COMPARE LENGTH
 1060  013346  012703  000001                 MOV     #1,R3           ;SET WORD COUNT
 1061  013352  012705  004362                 MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
 1062  013356  012704  003762                 MOV     #IBUFF,R4       ;SET TEST WORD POINTER
 1063  013362  021514                 18$:    CMP     (R5),(R4)       ;CHECK WORD
 1064  013364  001427                         BEQ     19$             ;GOOD - SKIP
 1065  013366  023727  003014  000012         CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
 1066  013374  003021                         BGT     20$             ;YES - SKIP
 1067  013376                                 PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
 (13)  013376  011546                         MOV     (R5),-(SP)
 (12)  013400  012746  011160                 MOV     #RESE4,-(SP)
 (11)  013404  011446                         MOV     (R4),-(SP)
 (10)  013406  012746  011154                 MOV     #RESE3,-(SP)
  (9)  013412  010346                         MOV     R3,-(SP)
  (8)  013414  012746  006170                 MOV     #MWORD,-(SP)
  (7)  013420  012746  011746                 MOV     #FMT15,-(SP)
  (6)  013424  012746  000007                 MOV     #7,-(SP)
  (3)  013430  010600                         MOV     SP,R0
  (4)  013432  104414                         TRAP    C$PNTB
  (4)  013434  062706  000020                 ADD     #20,SP
 1068  013440  005237  003014         20$:    INC     MORECE          ;BUMP ERROR COUNTER
 1069  013444  022524                 19$:    CMP     (R5)+,(R4)+     ;BUMP POINTERS
 1070  013446  005203                         INC     R3              ;BUMP COUNTER
 1071  013450  005301                         DEC     R1              ;DEC LENGTH COUNT
 1072  013452  001343                         BNE     18$             ;LOOP IF NOT DONE
 1073  013454  005737  003014         25$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
 1074  013460  001421                         BEQ     27$             ;NO - SKIP
 1075  013462  012701  000200                 MOV     #128.,R1        ;SET COMPARE LENGTH
 1076  013466                                 PRINTB  #FMT27,#TCERR,MORECE,#RESE6,R1
 (11)  013466  010146                         MOV     R1,-(SP)
 (10)  013470  012746  011172                 MOV     #RESE6,-(SP)
  (9)  013474  013746  003014                 MOV     MORECE,-(SP)
  (8)  013500  012746  010237                 MOV     #TCERR,-(SP)
  (7)  013504  012746  012310                 MOV     #FMT27,-(SP)
  (6)  013510  012746  000005                 MOV     #5,-(SP)
  (3)  013514  010600                         MOV     SP,R0
  (4)  013516  104414                         TRAP    C$PNTB
  (4)  013520  062706  000014                 ADD     #14,SP
```

```
 1077  013524  012605                  27$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
 1078  013526  012604                          MOV     (SP)+,R4
 1079  013530  012603                          MOV     (SP)+,R3
 1080  013532  012601                          MOV     (SP)+,R1
 1081  013534  004737  016126                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1082  013540                          ENDMSG
  (3)  013540                          L10005:
  (3)  013540  104423                          TRAP    C$MSG
 1083
 1084  013542                          BGNMSG  ERR7
 1085  013542  005277  167406                  INC     @ERRPOINT       ;BUMP ERROR COUNT
 1086  013546  010146                          MOV     R1,-(SP)        ;STORE R1
 1087  013550  004737  023356                  JSR     PC,RPTOP        ;REPORT OPERATION
 1088  013554  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
 1089  013560  012721  010523                  MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
 1090  013564  013721  003060                  MOV     T.STAT,(R1)+    ;INSERT IS VALUE
 1091  013570  010311                          MOV     R3,(R1)         ;INSERT SB VALUE
 1092  013572  004737  024144                  JSR     PC,RPTRES       ;REPORT RESULTS
 1093  013576  004737  024352                  JSR     PC,RPTREM       ;REPORT REMAINDER
 1094  013602  012601                          MOV     (SP)+,R1        ;RESTORE R1
 1095  013604  004737  016126                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1096  013610                          ENDMSG
  (3)  013610                          L10006:
  (3)  013610  104423                          TRAP    C$MSG
 1097
 1098  013612                          BGNMSG  ERR8
 1099  013612  005277  167336                  INC     @ERRPOINT       ;BUMP ERROR COUNT
 1100  013616  010146                          MOV     R1,-(SP)        ;STORE R1
 1101  013620  010346                          MOV     R3,-(SP)        ;STORE R3
 1102  013622  004737  023356                  JSR     PC,RPTOP        ;REPORT OPERATION
 1103  013626  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
 1104  013632  012721  010726                  MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
 1105  013636  013711  003052                  MOV     HDWRD1,(R1)     ;GET HEADER WORD
 1106  013642  012703  000007                  MOV     #7,R3           ;SET SHIFT COUNT
 1107  013646  000241                  3$:     CLC
 1108  013650  006011                          ROR     (R1)            ;ALIGN CHAR FOR PRINTING
 1109  013652  005303                          DEC     R3              ;  AS IS VALUE
 1110  013654  001374                          BNE     3$
 1111  013656  005721                          TST     (R1)+           ;BUMP PARAM POINTER
 1112  013660  013711  003102                  MOV     NEWCYL,(R1)     ;INSERT SB VALUE
 1113  013664  004737  024144                  JSR     PC,RPTRES       ;REPORT RESULTS
 1114  013670  004737  024352                  JSR     PC,RPTREM       ;REPORT REMAINDER
 1115  013674  012603                          MOV     (SP)+,R3        ;RESTORE R3
 1116  013676  012601                          MOV     (SP)+,R1        ;RESTORE R1
 1117  013700  004737  016126                  JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1118  013704                          ENDMSG
  (3)  013704                          L10007:
  (3)  013704  104423                          TRAP    C$MSG
 1119
 1120  013706                          BGNMSG  ERR9
 1121  013706  005277  167242                  INC     @ERRPOINT       ;BUMP ERROR COUNT
 1122  013712  010146                          MOV     R1,-(SP)        ;STORE R1
 1123  013714  004737  023356                  JSR     PC,RPTOP        ;REPORT OPERATION
 1124  013720  012721  000003                  MOV     #3,(R1)+        ;SET PARAM NUMBER
 1125  013724  010321                          MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
 1126  013726  010421                          MOV     R4,(R1)+        ;SET IS VALUE
```

```
 1127  013730  010521                        MOV     R5,(R1)+        ;SET SB VALUE
 1128  013732  004737  024144                JSR     PC,RPTRES       ;REPORT RESULTS
 1129  013736  004737  024352                JSR     PC,RPTREM       ;REPORT REMAINDER
 1130  013742  012601                        MOV     (SP)+,R1        ;RESTORE R1
 1131  013744  004737  016126                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1132  013750                        ENDMSG
   (3) 013750                        L10010:
   (3) 013750  104423                        TRAP    C$MSG
 1133  013752                        BGNMSG  ERR10
 1134  013752  010146                        MOV     R1,-(SP)        ;STORE R1
 1135  013754  005737  003014                TST     MORECE          ;TEST IF 2ND BAD LINE
 1136  013760  001051                        BNE     3$              ;YES - SKIP
 1137  013762  005277  167166                INC     @ERRPOINT       ;BUMP ERROR COUNT
 1138  013766  004737  023356                JSR     PC,RPTOP        ;REPORT OPERATION
 1139  013772                                PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
  (11) 013772  005046                        CLR     -(SP)
  (11) 013774  153716  003033                BISB    RLDRV+1,(SP)
  (10) 014000  012746  006051                MOV     #DRVNAM,-(SP)
   (9) 014004  013746  003026                MOV     RLBAS,-(SP)
   (8) 014010  012746  006040                MOV     #BASADD,-(SP)
   (7) 014014  012746  011443                MOV     #FMT5,-(SP)
   (6) 014020  012746  000005                MOV     #5,-(SP)
   (3) 014024  010600                        MOV     SP,R0
   (4) 014026  104414                        TRAP    C$PNTB
   (4) 014030  062706  000014                ADD     #14,SP
 1140  014034                                PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
  (14) 014034  011546                        MOV     (R5),-(SP)
  (13) 014036  012746  011160                MOV     #RESE4,-(SP)
  (12) 014042  011446                        MOV     (R4),-(SP)
  (11) 014044  012746  011154                MOV     #RESE3,-(SP)
  (10) 014050  010346                        MOV     R3,-(SP)
   (9) 014052  012746  006170                MOV     #MWORD,-(SP)
   (8) 014056  012746  005421                MOV     #MRSLT,-(SP)
   (7) 014062  012746  011714                MOV     #FMT14,-(SP)
   (6) 014066  012746  000010                MOV     #10,-(SP)
   (3) 014072  010600                        MOV     SP,R0
   (4) 014074  104414                        TRAP    C$PNTB
   (4) 014076  062706  000022                ADD     #22,SP
 1141  014102  000421                        BR      4$
 1142  014104                        3$:     PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
  (13) 014104  011546                        MOV     (R5),-(SP)
  (12) 014106  012746  011160                MOV     #RESE4,-(SP)
  (11) 014112  011446                        MOV     (R4),-(SP)
  (10) 014114  012746  011154                MOV     #RESE3,-(SP)
   (9) 014120  010346                        MOV     R3,-(SP)
   (8) 014122  012746  006170                MOV     #MWORD,-(SP)
   (7) 014126  012746  011746                MOV     #FMT15,-(SP)
   (6) 014132  012746  000007                MOV     #7,-(SP)
   (3) 014136  010600                        MOV     SP,R0
   (4) 014140  104414                        TRAP    C$PNTB
   (4) 014142  062706  000020                ADD     #20,SP
 1143  014146  005237  003014        4$:     INC     MORECE          ;INC COMPARE ERROR COUNT
 1144  014152  012601                        MOV     (SP)+,R1        ;RESTORE R1
 1145  014154  004737  016126                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
 1146  014160                        ENDMSG
   (3) 014160                        L10011:
```

```
  (3)  014160  104423                    TRAP    C$MSG
 1147  014162                  ENDMOD
 1148                                    .EVEN
 1149
 1150  014162                  BGNMOD  HPTCODE
 1151  014162                  BGNHW
  (3)  014162  000006                    .WORD   L10012-L$HW/2
 1152  014164  174400                    .WORD   174400          ;CSR BASE ADDRESS DEFAULT
 1153  014166  000160                    .WORD   160             ;VECTOR DEFAULT
 1154  014170  000240                    .WORD   240             ;PRIORITY DEFAULT
 1155  014172  000001                    .WORD   1               ;TYPE OF DRIVE, RL01=1, RL02=2
 1156  014174  000000                    .WORD   0               ;DRIVE NUMBER DEFAULT
 1157  014176  000001                    .WORD   1               ;RL11 CONTROLLER
 1158  014200                  ENDHW
  (3)  014200                  L10012:
 1159  014200                  ENDMOD
 1160
 1161  014200                  BGNMOD  SPTCODE
 1162  014200                  BGNSW
  (3)  014200  000006                    .WORD   L10013-L$SW/2
 1163  014202  000000          MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
 1164                                                          ;BIT 1 = USE ALL SECTORS
 1165                                                          ;BIT 2 = EXECUTE DRIVE SELECT TEST
 1166                                                          ;BIT 3 = EXECUTE HEAD ALIGNMENT
 1167                                                          ;BIT 12 = HEAD SELECT SUPPLIED FLAG
 1168                                                          ;BIT 13 = HILIMIT SPECIFIED FLAG
 1169                                                          ;BIT 14 = LO LIMIT SPECIFIED FLAG
 1170                                                          ;BIT 15 = DO MANUAL INTERVENTION
 1171  014204  000000          LOLIMW: .WORD   0
 1172  014206  000377          HILIMW: .WORD   255.
 1173  014210  000000          HEADW:  .WORD   0
 1174  014212  000024          ERLIMW: .WORD   20.             ;ERROR LIMIT
 1175  014214  000012          DCLIMW: .WORD   10.             ;COMPARE ERROR LIMIT
 1176  014216                  ENDSW
  (3)  014216                  L10013:
 1177  014216                  ENDMOD
 1178
 1179  014216                  BGNMOD  DSPCODE
 1181  014216                  DISPATCH        16
  (4)  014216  000020                    .WORD   16
  (6)  014220  024636                    .WORD   T1
  (6)  014222  025116                    .WORD   T2
  (6)  014224  025324                    .WORD   T3
  (6)  014226  027614                    .WORD   T4
  (6)  014230  030710                    .WORD   T5
  (6)  014232  031314                    .WORD   T6
  (6)  014234  032470                    .WORD   T7
  (6)  014236  033374                    .WORD   T8
  (6)  014240  033462                    .WORD   T9
  (6)  014242  034130                    .WORD   T10
  (6)  014244  034620                    .WORD   T11
  (6)  014246  035372                    .WORD   T12
  (6)  014250  036056                    .WORD   T13
  (6)  014252  036276                    .WORD   T14
  (6)  014254  036556                    .WORD   T15
  (6)  014256  037300                    .WORD   T16
```

```
 1186  014260                          ENDMOD
 1187
 1188                                  ;LOAD PROTECTION TABLE
 1189  014260                          BGNPROT
 1190  014260  000000                  .WORD     0                   ;P-TABLE OFFSET OF CSR
 1191  014262  177777                  .WORD    -1                   ;NOT A MASS-BUSS DRIVE
 1192  014264  000010                  .WORD    10                   ;P-TABLE OFFSET OF DRIVE
 1193  014266                          ENDPROT
 1194
 1195
 1196                                  .SBTTL  INITIALIZATION CODE
 1197
 1198  014266                          BGNMOD  INITCODE
 1199  014266                          BGNINIT
 1200                                  ;CHECK FOR PRESENCE OF A P-CLOCK
 1201  014266  005037  003144                  CLR     CLKFLG          ;CLEAR CLOCK FLAG
 1202  014272                                  CLOCK   P,CLKADR        ;P-CLOCK?
  (3)  014272  012700  000120                  MOV     #'P,R0
  (3)  014276  104462                          TRAP    C$CLCK
  (3)  014300  010037  003146                  MOV     R0,CLKADR
 1203  014304                                  BNCOMPLETE      1$      ;BRANCH IF NO P-CLOCK
  (2)  014304  103002                          BCC     1$
 1204  014306  005237  003144                  INC     CLKFLG          ;INDICATE PRESENCE OF A P-CLOCK
 1205  014312                          1$:     SETPRI  #340            ;SET PRIORITY TO 7 TO INHIBIT INTERRUPTS
  (3)  014312  012700  000340                  MOV     #340,R0
  (3)  014316  104441                          TRAP    C$SPRI
 1206  014320                                  MANUAL                  ;CHECK IF MANUAL INTERVENTION ALLOWED
  (3)  014320  104450                          TRAP    C$MANI
 1207  014322                                  BCOMPLETE       2$      ;YES - SKIP
  (2)  014322  103403                          BCS     2$
 1208  014324  042737  100014  014202          BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
 1209                                                                  ; INTERVENTION FLAGS
 1210  014332  005037  003002          2$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
 1211  014336                                  READEF  #EF.PWR         ;POWER FAILURE?
  (3)  014336  012700  000034                  MOV     #EF.PWR,R0
  (3)  014342  104447                          TRAP    C$REFG
 1212  014344                                  BNCOMPLETE  4$          ;NO, GO CHECK NEW PASS
  (2)  014344  103005                          BCC     4$
 1213  014346  013737  002012  003366          MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
 1214  014354  000137  014766                  JMP     PWCON           ;GO SERVICE POWER FAIL
 1215                                  ;"START" COMMAND SEQUENCE
 1216  014360                          4$:     READEF  #EF.START       ;CHECK IF START
  (3)  014360  012700  000040                  MOV     #EF.START,R0
  (3)  014364  104447                          TRAP    C$REFG
 1217  014366                                  BNCOMPLETE      RESTART ;NO - SKIP
  (2)  014366  103034                          BCC     RESTART
 1218                                  ;       ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
 1219                                  ;       PASS COUNT, AND ERROR COUNT.
 1220  014370  013737  002012  003074          MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
 1221  014376  005037  003356          RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
 1222  014402  012700  003156                  MOV     #ERRCNT,R0
 1223  014406  012701  000100                  MOV     #64.,R1         ;GET A COUNT
 1224  014412  005020                  1$:     CLR     (R0)+           ;CLEAR ERROR COUNTER STORAGE AREA
 1225  014414  005301                          DEC     R1
 1226  014416  001375                          BNE     1$              ;LOOP TILL ALL CLEARED
 1227  014420  012737  003154  003154          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
```

```
1228  014426  012737  177777  003360          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
1229  014434  012737  177777  003010          MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
1230  014442  032737  040000  014202  LAB:    BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
1231  014450  001002                          BNE     5$              ;YES - SKIP
1232  014452  005037  014204                  CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
1233  014456  000432                  5$:     BR      SETDON
1234  014460                          RESTART:
1235  014460                                  READEF  #EF.RESTART     ;CHECK IF RESTART
 (3)  014460  012700  000037                  MOV     #EF.RESTART,R0
 (3)  014464  104447                          TRAP    C$REFG
1236  014466                                  BCOMPLETE       RSTRT   ;NO - SKIP
 (2)  014466  103743                          BCS     RSTRT
1237                                    ;''CONTINUE'' COMMAND SEQUENCE
1238  014470                           CONTINUE:
1239  014470                                  READEF  #EF.CONTINUE    ;TEST IF CONTINUE
 (3)  014470  012700  000036                  MOV     #EF.CONTINUE,R0
 (3)  014474  104447                          TRAP    C$REFG
1240  014476                                  BCOMPLETE       PWCON
 (2)  014476  103533                          BCS     PWCON
1241                                    ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST
1242  014500                                  READEF  #EF.NEW         ;CHECK IF STARTING NEW PASS
 (3)  014500  012700  000035                  MOV     #EF.NEW,R0
 (3)  014504  104447                          TRAP    C$REFG
1243  014506                                  BCOMPLETE       PASNEW
 (2)  014506  103403                          BCS     PASNEW
1244  014510                          NXTPAS:
1245  014510  005737  003074                  TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
1246  014514  001013                          BNE     SETDON          ;NO - SKIP
1247  014516  005237  003356          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
1248  014522  012737  003154  003154        MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1249  014530  013737  002012  003074        MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
1250  014536  012737  177777  003360        MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
1251  014544  005237  003360          SETDON: INC     PSETNM          ;NEXT SET OF PARAMETERS
1252  014550  005337  003074                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
1253  014554  062737  000002  003154        ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
1254  014562  013700  003360                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
1255  014566  012702  003026                  MOV     #RLBAS,R2       ;GET POINTER TO RL11 BASE ADDRESS
1256  014572                                  GPHARD  R0,R1
 (3)  014572  104442                          TRAP    C$GPHRD
 (3)  014574  010001                          MOV     R0,R1
1257  014576                                  BCOMPLETE       7$      ;SKIP IF GOOD PARAM
 (2)  014576  103406                          BCS     7$
1258  014600  005737  003366                  TST     PWRFLG          ;RECENT POWER FAILURE
1259  014604  001741                          BEQ     NXTPAS  ;NO
1260  014606  005337  003366                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
1261  014612  000736                          BR      NXTPAS
1262                                    ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
1263  014614  012122                  7$:     MOV     (R1)+,(R2)+     ;STORE CSR
1264  014616  012122                          MOV     (R1)+,(R2)+     ;STORE VECTOR
1265  014620  005721                          TST     (R1)+           ;BUMP PAST PRIORITY
1266  014622  012137  002276                  MOV     (R1)+,T.DRIVE   ;STORE DRIVE TYPE
1267  014626  012122                          MOV     (R1)+,(R2)+
1268  014630  022737  000001  002276          CMP     #1,T.DRIVE
1269  014636  001426                          BEQ     65$
1270                                    ;INITIALIZE RL02 PARAMETERS
1271  014640  012737  000776  002306          MOV     #510.,NXTHL
```

```
1272  014646  012737  000777  002302          MOV     #511.,HLMTW
1273  014654  012737  001000  002310          MOV     #512.,GBND
1274  014662  012737  177600  002312          MOV     #177600,CAMSK
1275  014670  012737  177600  002314          MOV     #177600,DIRMSK
1276  014676  012737  177600  002316          MOV     #177600,HDCYL
1277  014704  012737  177000  002304          MOV     #177000,CLRBYT
1278  014712  000425                          BR      PWCON
1279                                  ;INITIALIZE RL01 PARAMETERS
1280  014714  012737  000377  002302  65$:    MOV     #255.,HLMTW
1281  014722  012737  000400  002310          MOV     #256.,GBND
1282  014730  012737  077600  002312          MOV     #77600,CAMSK
1283  014736  012737  077600  002314          MOV     #77600,DIRMSK
1284  014744  012737  077600  002316          MOV     #77600,HDCYL
1285  014752  012737  000376  002306          MOV     #254.,NXTHL
1286  014760  012737  177400  002304          MOV     #177400,CLRBYT
1287
1288  014766  032737  020000  014202  PWCON:  BIT     #HICYL,MISWIW
1289  014774  001003                          BNE     1$
1290  014776  013737  002302  014206          MOV     HLMTW,HILIMW
1291  015004                          1$:     SETVEC  RLVEC,#INTHLR,#340      ;SET UP INTERRUPT VECTOR FOR DRIVE
 (7)  015004  012746  000340                  MOV     #340,-(SP)
 (6)  015010  012746  016052                  MOV     #INTHLR,-(SP)
 (5)  015014  013746  003030                  MOV     RLVEC,-(SP)
 (4)  015020  012746  000003                  MOV     #3,-(SP)
 (3)  015024  104437                          TRAP    C$SVEC
 (2)  015026  062706  000010                  ADD     #10,SP
1292  015032                                  SETPRI  #0              ;SET PRIORITY TO 0 TO ALLOW INTERRUPTS
 (3)  015032  012700  000000                  MOV     #0,R0
 (3)  015036  104441                          TRAP    C$SPRI
1293  015040  013702  003026                  MOV     RLBAS,R2        ;SET RL11 BASE ADDRESS POINTER
1294
1295
1297
1298  015044                                  MANUAL                  ;MANUAL INTERVENTION ALLOWED?
 (3)  015044  104450                          TRAP    C$MANI
1299  015046                                  BNCOMPLETE      4$      ;NO
 (2)  015046  103004                          BCC     4$
1300
1301  015050  005737  003356                  TST     PASNUM          ;YES, CHECK PASS NUMBER
1302  015054  001001                          BNE     4$              ;NOT FIRST PASS, NEED DRIVE UP
1303  015056  000520                          BR      8$              ;FIRST PASS, PROGRAM WILL INSTRUCT USER
1304
1306                                  ;CHECK IF POWER FAILURE WAIT IS NEEDED
1307
1308  015060  005737  003366          4$:     TST     PWRFLG          ;NEEDED?
1309  015064  001515                          BEQ     8$              ;NO, SKIP
1310
1311  015066  013705  003032                  MOV     RLDRV,R5        ;DRIVE SELECT
1312  015072  052705  000200                  BIS     #CRDYMSK,R5     ;SET CRDY
1313  015076  010562  000000                  MOV     R5,RLCS(R2)     ;SELECT DRIVE
1314  015102  012701  000170                  MOV     #120.,R1        ;INITIALIZE WAIT COUNT
1315  015106  032762  000001  000000  9$:     BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET
1316  015114  001101                          BNE     8$              ;YES START TEST
1317
1318                                  .LIST   ME
1319  015116                                  WAITMS  #10.            ;WAIT A SECOND
```

```
    (1)  015116  012737  000012  003142          MOV     ##10.,DLYCNT    ;INITIALIZE DELAY COUNTER
    (1)  015124  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
    (1)  015130  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
    (1)  015134                          64$:    DELAY   #250.           ;IMPLEMENT 25-MS TIME DELAY
    (2)  015134                          M$GNINS <MOV    ##250.,(PC)+>
    (3)  015134  012727  000372                  MOV     ##250.,(PC)+
    (3)                                  .MEXIT
    (2)  015140                          M$GNINS <.WORD  0>
    (3)  015140  000000                          .WORD   0
    (3)                                  .MEXIT
    (2)  015142                          M$GNINS <MOV    L$DLY,(PC)+>
    (3)  015142  013727  002116                  MOV     L$DLY,(PC)+
    (3)                                  .MEXIT
    (2)  015146                          M$GNINS <.WORD  0>
    (3)  015146  000000                          .WORD   0
    (3)                                  .MEXIT
    (2)  015150                          M$GNINS <DEC    -6(PC)>
    (3)  015150  005367  177772                  DEC     -6(PC)
    (3)                                  .MEXIT
    (2)  015154                          M$GNINS <BNE    .-4>
    (3)  015154  001375                          BNE     .-4
    (3)                                  .MEXIT
    (2)  015156                          M$GNINS <DEC    -22(PC)>
    (3)  015156  005367  177756                  DEC     -22(PC)
    (3)                                  .MEXIT
    (2)  015162                          M$GNINS <BNE    .-20>
    (3)  015162  001367                          BNE     .-20
    (3)                                  .MEXIT
    (1)  015164  005337  003142                  DEC     DLYCNT          ;DECREMENT DELAY COUNT
    (1)  015170  001361                          BNE     64$             ;BRANCH IF TIME DELAY NOT EXPIRED
 1320                                    .NLIST  ME
 1321  015172  005301                          DEC     R1              ;SIXTY GONE BY
 1322  015174  001344                          BNE     9$              ;NO
 1323  015176                                  PRINTF  #FMT24,#NOPWR   ;REPORT 'DRV DID NOT REC'R FROM PWR FAIL''
    (8)  015176  012746  006056                  MOV     #NOPWR,-(SP)
    (7)  015202  012746  012247                  MOV     #FMT24,-(SP)
    (6)  015206  012746  000002                  MOV     #2,-(SP)
    (3)  015212  010600                          MOV     SP,R0
    (4)  015214  104417                          TRAP    C$PNTF
    (4)  015216  062706  000006                  ADD     #6,SP
 1324  015222                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT DRIVE UNIBUS
   (11)  015222  005046                          CLR     -(SP)
   (11)  015224  153716  003033                  BISB    RLDRV+1,(SP)
   (10)  015230  012746  006051                  MOV     #DRVNAM,-(SP)
    (9)  015234  013746  003026                  MOV     RLBAS,-(SP)
    (8)  015240  012746  006040                  MOV     #BASADD,-(SP)
    (7)  015244  012746  011443                  MOV     #FMT5,-(SP)
    (6)  015250  012746  000005                  MOV     #5,-(SP)
    (3)  015254  010600                          MOV     SP,R0
    (4)  015256  104417                          TRAP    C$PNTF
    (4)  015260  062706  000014                  ADD     #14,SP
 1325                                                                    ;/ADDRESS AND DRIVE NUMBER
 1326  015264                                  PRINTF  #FMT3           ;NEW LINE
    (7)  015264  012746  011427                  MOV     #FMT3,-(SP)
    (6)  015270  012746  000001                  MOV     #1,-(SP)
    (3)  015274  010600                          MOV     SP,R0
```

```
 (4)  015276  104417                          TRAP    C$PNTF
 (4)  015300  062706  000004                  ADD     #4,SP
1327  015304                                  DODU    PSETNM          ;DO DROP UNIT ON DRIVE
 (3)  015304  013700  003360                  MOV     PSETNM,R0
 (3)  015310  104451                          TRAP    C$DODU
1328  015312                                  DOCLN                   ;INVOKE CLEAN-UP CODE TO RESTORE DRIVE
 (3)  015312  104444                          TRAP    C$DCLN
1329                                                                  ;/TO STATIC STATE
1330  015314  005037  003140                  CLR     ERRVEC          ;CLEAR ERROR VECTOR
1331
1332  015320                          8$:
1333
1334  015320                          ENDINIT
 (3)  015320                          L10015:
 (3)  015320  104411                          TRAP    C$INIT
1335
1336  015522                          ENDMOD
1337
1338
1339                                  .SBTTL  AUTO DROP SECTION
1340
1341                                  ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
1342                                  ;''ADR'' FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
1343                                  ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
1344                                  ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
1345                                  ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
1346                                  ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
1347
1348  015322                          BGNAUTO
1349  015322  005037  003364                  CLR     TRPFLG          ;CLEAR TRAP FLAG
1350  015326                                  SETVEC  ERRVEC,#TRPHAN,#340     ;SET UP TRAP VECTOR TO DETECT
 (7)  015326  012746  000340                  MOV     #340,-(SP)
 (6)  015332  012746  016120                  MOV     #TRPHAN,-(SP)
 (5)  015336  013746  003140                  MOV     ERRVEC,-(SP)
 (4)  015342  012746  000003                  MOV     #3,-(SP)
 (3)  015346  104437                          TRAP    C$SVEC
 (2)  015350  062706  000010                  ADD     #10,SP
1351                                                                  ;/NON-EXISTENT CONTROLLER UNIBUS
1352                                                                  ;/ADDRESS
1353  015354  013702  003026                  MOV     RLBAS,R2        ;GET RL11 BASE ADDRESS
1354  015360  005762  000000                  TST     RLCS(R2)        ;ACCESS DRIVE CONTROLLER UNIBUS ADDRESS
1355  015364  005737  003364                  TST     TRPFLG          ;DID TRAP OCCUR?
1356  015370  001447                          BEQ     1$              ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
1357  015372                                  PRINTF  #FMT24,#NOCTLR  ;ELSE, PRINT MSG. 'DRV DROPPED - NO CNTLR''
 (8)  015372  012746  006730                  MOV     #NOCTLR,-(SP)
 (7)  015376  012746  012247                  MOV     #FMT24,-(SP)
 (6)  015402  012746  000002                  MOV     #2,-(SP)
 (3)  015406  010600                          MOV     SP,R0
 (4)  015410  104417                          TRAP    C$PNTF
 (4)  015412  062706  000006                  ADD     #6,SP
1358  015416                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11)  015416  005046                          CLR     -(SP)
(11)  015420  153716  003033                  BISB    RLDRV+1,(SP)
(10)  015424  012746  006051                  MOV     #DRVNAM,-(SP)
 (9)  015430  013746  003026                  MOV     RLBAS,-(SP)
 (8)  015434  012746  006040                  MOV     #BASADD,-(SP)
```

```
 (7)  015440  012746  011443                MOV     #FMT5,-(SP)
 (6)  015444  012746  000005                MOV     #5,-(SP)
 (3)  015450  010600                        MOV     SP,R0
 (4)  015452  104417                        TRAP    C$PNTF
 (4)  015454  062706  000014                ADD     #14,SP
1359
1360  015460                                PRINTF  #FMT3                   ;PRINT DRIVE INFORMATION
 (7)  015460  012746  011427                MOV     #FMT3,-(SP)
 (6)  015464  012746  000001                MOV     #1,-(SP)
 (3)  015470  010600                        MOV     SP,R0
 (4)  015472  104417                        TRAP    C$PNTF
 (4)  015474  062706  000004                ADD     #4,SP
1361  015500                                DODU    PSETNM                  ;DO DROP UNIT ON DRIVE
 (3)  015500  013700  003360                MOV     PSETNM,R0
 (3)  015504  104451                        TRAP    C$DODU
1362  015506  000460                        BR      2$                      ;BRANCH TO EXIT
1363  015510  013705  003032          1$:   MOV     RLDRV,R5                ;ELSE, GET DRIVE NUMBER
1364  015514  052705  000200                BIS     #CRDYMSK,R5             ;SET CONTROLLER READY
1365  015520  010562  000000                MOV     R5,RLCS(R2)             ;LOAD IN THE DRIVE NUMBER
1366  015524  032762  000001  000000        BIT     #DRDYMSK,RLCS(R2)               ;IS DRIVE READY?
1367  015532  001046                        BNE     2$                      ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
1368  015534                                PRINTF  #FMT24,#NOTRDY  ;PRINT MSG. 'DRV DROPPED - NOT RDY''
 (8)  015534  012746  006757                MOV     #NOTRDY,-(SP)
 (7)  015540  012746  012247                MOV     #FMT24,-(SP)
 (6)  015544  012746  000002                MOV     #2,-(SP)
 (3)  015550  010600                        MOV     SP,R0
 (4)  015552  104417                        TRAP    C$PNTF
 (4)  015554  062706  000006                ADD     #6,SP
1369  015560                                PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 015560  005046                        CLR     -(SP)
 (11) 015562  153716  003033                BISB    RLDRV+1,(SP)
 (10) 015566  012746  006051                MOV     #DRVNAM,-(SP)
 (9)  015572  013746  003026                MOV     RLBAS,-(SP)
 (8)  015576  012746  006040                MOV     #BASADD,-(SP)
 (7)  015602  012746  011443                MOV     #FMT5,-(SP)
 (6)  015606  012746  000005                MOV     #5,-(SP)
 (3)  015612  010600                        MOV     SP,R0
 (4)  015614  104417                        TRAP    C$PNTF
 (4)  015616  062706  000014                ADD     #14,SP
1370
1371  015622                                PRINTF  #FMT3                   ;PRINT DRIVE INFORMATION
 (7)  015622  012746  011427                MOV     #FMT3,-(SP)
 (6)  015626  012746  000001                MOV     #1,-(SP)
 (3)  015632  010600                        MOV     SP,R0
 (4)  015634  104417                        TRAP    C$PNTF
 (4)  015636  062706  000004                ADD     #4,SP
1372  015642                                DODU    PSETNM                  ;DO DROP UNIT ON DRIVE
 (3)  015642  013700  003360                MOV     PSETNM,R0
 (3)  015646  104451                        TRAP    C$DODU
1373  015650                          2$:   CLRVEC  ERRVEC                  ;RELEASE THE ERROR VECTOR
 (3)  015650  013700  003140                MOV     ERRVEC,R0
 (3)  015654  104436                        TRAP    C$CVEC
1374  015656                          ENDAUTO
 (3)  015656                          L10016:
 (3)  015656  104461                        TRAP    C$AUTO
1375
```

```
1376
1377
1378
1379  015660                                  BGNMOD  CLNCODE
1380  015660                                  BGNCLN
1381
1382  015660                                          SETVEC  ERRVEC,#TRPHAN,#340
 (7)  015660  012746  000340                           MOV     #340,-(SP)
 (6)  015664  012746  016120                           MOV     #TRPHAN,-(SP)
 (5)  015670  013746  003140                           MOV     ERRVEC,-(SP)
 (4)  015674  012746  000003                           MOV     #3,-(SP)
 (3)  015700  104437                                   TRAP    C$SVEC
 (2)  015702  062706  000010                           ADD     #10,SP
1383
1384  015706                                          SETPRI #7                       ;SET PRIORITY TO 7
 (3)  015706  012700  000007                           MOV     #7,R0
 (3)  015712  104441                                   TRAP    C$SPRI
1385  015714  032762  000200  000000          2$:     BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
1386  015722  001407                                   BEQ     3$                      ;NO LOOP UNTIL READY
1387  015724  053762  003032  000000                   BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
1388  015732  032762  000001  000000                   BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
1389  015740  001026                                   BNE     5$                      ;NO - SKIP
1390                                          .LIST   ME
1391  015742                                  3$:     WAITMS  #3                      ;WAIT 300 MS
 (1)  015742  012737  000003  003142                   MOV     ##3,DLYCNT              ;INITIALIZE DELAY COUNTER
 (1)  015750  006337  003142                           ASL     DLYCNT                  ;MULTIPLY ARGUMENT BY 2
 (1)  015754  006337  003142                           ASL     DLYCNT                  ;MULTIPLY ARGUMENT BY 2 AGAIN
 (1)  015760                                  64$:    DELAY   #250.                   ;IMPLEMENT 25-MS TIME DELAY
 (2)  015760                                  M$GNINS <MOV    ##250.,(PC)+>
 (3)  015760  012727  000372                           MOV     ##250.,(PC)+
 (3)                                          .MEXIT
 (2)  015764                                  M$GNINS <.WORD  0>
 (3)  015764  000000                                   .WORD   0
 (3)                                          .MEXIT
 (2)  015766                                  M$GNINS <MOV    L$DLY,(PC)+>
 (3)  015766  013727  002116                           MOV     L$DLY,(PC)+
 (3)                                          .MEXIT
 (2)  015772                                  M$GNINS <.WORD  0>
 (3)  015772  000000                                   .WORD   0
 (3)                                          .MEXIT
 (2)  015774                                  M$GNINS <DEC    -6(PC)>
 (3)  015774  005367  177772                           DEC     -6(PC)
 (3)                                          .MEXIT
 (2)  016000                                  M$GNINS <BNE    .-4>
 (3)  016000  001375                                   BNE     .-4
 (3)                                          .MEXIT
 (2)  016002                                  M$GNINS <DEC    -22(PC)>
 (3)  016002  005367  177756                           DEC     -22(PC)
 (3)                                          .MEXIT
 (2)  016006                                  M$GNINS <BNE    .-20>
 (3)  016006  001367                                   BNE     .-20
 (3)                                          .MEXIT
 (1)  016010  005337  003142                           DEC     DLYCNT                  ;DECREMENT DELAY COUNT
 (1)  016014  001361                                   BNE     64$                     ;BRANCH IF TIME DELAY NOT EXPIRED
1392                                          .NLIST  ME
1393  016016                                  5$:     CLRVEC  RLVEC                   ;RELEASE DRIVE VECTOR
```

```
   (3)  016016  013700  003030                    MOV     RLVEC,R0
   (3)  016022  104436                            TRAP    C$CVEC
  1394  016024  005737  003366                    TST     PWRFLG          ;PWR FAIL SET
  1395  016030  001402                            BEQ     7$              ;NO
  1396  016032  005337  003366                    DEC     PWRFLG
  1397  016036                            7$:     CLRVEC  ERRVEC
   (3)  016036  013700  003140                    MOV     ERRVEC,R0
   (3)  016042  104436                            TRAP    C$CVEC
  1398  016044                            ENDCLN
   (3)  016044                            L10017:
   (3)  016044  104412                            TRAP    C$CLEAN
  1399
  1400  016046                            BGNDU
  1401  016046  000240                            NOP
  1402  016050                            ENDDU
   (3)  016050                            L10020:
   (3)  016050  104453                            TRAP    C$DU
  1403
  1404  016052                            ENDMOD
  1405
  1406
  1407
  1408                                    .SBTTL  INTERRUPT SERVICE ROUTINES
  1409
  1410  016052                            BGNSRV  INTHLR
  1411                                    ;INTERRUPT HANDLER FOR DRIVE ABORTS WAIT TIMER AND STORES ALL RL11 REGISTERS
  1412  016052  005037  003142                    CLR     DLYCNT          ;CLEAR UNELAPSED DELAY COUNT
  1413  016056  012237  003044                    MOV     (R2)+,T.CS      ;STORE RL REGISTERS
  1414  016062  012237  003046                    MOV     (R2)+,T.BA
  1415  016066  012237  003050                    MOV     (R2)+,T.DA
  1416  016072  011237  003052                    MOV     (R2),T.MP
  1417  016076  012737  177777  003006            MOV     #-1,DONE        ;SET DONE FLAG
  1418  016104  013702  003026                    MOV     RLBAS,R2        ;RESTORE R2
  1419  016110                            ENDSRV
   (3)  016110                            L10021:
   (2)  016110  000002                            RTI
  1420
  1421                                    ;INTERRUPT SERVICE ROUTINE FOR P-CLOCK DECREMENTS DELAY COUNTER AT 100-MICROSECOND
  1422                                    ;TIME INTERVALS
  1423  016112                            BGNSRV  CLKINT
  1424  016112  005337  003142                    DEC     DLYCNT          ;DECREMENT CLOCK DELAY COUNTER
  1425  016116                            ENDSRV
   (3)  016116                            L10022:
   (2)  016116  000002                            RTI
  1426
  1427                                    ;INTERRUPT SERVICE ROUTINE SETS TRAP FLAG WHEN A NON-EXISTENT UNIBUS ADDRESS IS
  1428                                    ;ACCESSED
  1429  016120                            BGNSRV  TRPHAN
  1430  016120  005237  003364                    INC     TRPFLG          ;INDICATE THAT TRAP OCCURRED
  1431  016124                            ENDSRV
   (3)  016124                            L10023:
   (2)  016124  000002                            RTI
  1432
  1433
```

```
                                        .SBTTL  GLOBAL SUBROUTINES


016126                                  BGNMOD  GLBSUB



                                        ;       ERROR LIMIT CHECKING ROUTINE
                                        ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
016126  027737  165022  014212  CKERLM: CMP     @ERRPOINT,ERLIMW        ;TEST IF ERROR LIMIT EXCEEDED
016134  002453                          BLT     1$                      ;NO - SKIP
016136                                  INLOOP                          ;CHECK IF IN ERROR LOOP
(3) 016136  104420                      TRAP    C$INLP
016140                                  BCOMPLETE       1$              ;YES - SKIP
(2) 016140  103451                      BCS     1$
016142                                  PRINTF  #FMT25,ERLIMW,#MEXERS ;PRINT MSG. 'OVER ERROR LIMIT - UNIT DROPPED''
(9) 016142  012746  011101              MOV     #MEXERS,-(SP)
(8) 016146  013746  014212              MOV     ERLIMW,-(SP)
(7) 016152  012746  012254              MOV     #FMT25,-(SP)
(6) 016156  012746  000003              MOV     #3,-(SP)
(3) 016162  010600                      MOV     SP,R0
(4) 016164  104417                      TRAP    C$PNTF
(4) 016166  062706  000010              ADD     #10,SP
016172                                  PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;PRINT DRIVE INFORMATION
(11) 016172 005046                      CLR     -(SP)
(11) 016174 153716  003033              BISB    RLDRV+1,(SP)
(10) 016200 012746  006051              MOV     #DRVNAM,-(SP)
(9) 016204  013746  003026              MOV     RLBAS,-(SP)
(8) 016210  012746  006040              MOV     #BASADD,-(SP)
(7) 016214  012746  011443              MOV     #FMT5,-(SP)
(6) 016220  012746  000005              MOV     #5,-(SP)
(3) 016224  010600                      MOV     SP,R0
(4) 016226  104417                      TRAP    C$PNTF
(4) 016230  062706  000014              ADD     #14,SP
016234                                  PRINTF  #FMT3
(7) 016234  012746  011427              MOV     #FMT3,-(SP)
(6) 016240  012746  000001              MOV     #1,-(SP)
(3) 016244  010600                      MOV     SP,R0
(4) 016246  104417                      TRAP    C$PNTF
(4) 016250  062706  000004              ADD     #4,SP
016254                                  DODU    PSETNM                  ;DROP DRIVE
(3) 016254  013700  003360              MOV     PSETNM,R0
(3) 016260  104451                      TRAP    C$DODU
016262                                  DOCLN                           ;GO TO CLEAN UP
(3) 016262  104444                      TRAP    C$DCLN
016264  000207              1$:         RTS     PC



                                        ;       READ AND STORE ALL RL11 REGISTERS
016266  016237  000000  003044  READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
016274  016237  000002  003046          MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
016302  016237  000004  003050          MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
016310  016237  000006  003052          MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
016316  000207                          RTS     PC              ;RETURN
```

```
1464
1465
1466                                    ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
1467  016320  011646                    WAITIN: MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
1468  016322  005066  000002                    CLR     2(SP)           ;CLEAR FOR POINTER
1469  016326  032762  000200  000000            BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
1470  016334  001420                            BEQ     4$              ;NO - SKIP TO WAIT
1471  016336  004737  016266                    JSR     PC,READRL       ;READ ALL RL REGS
1472  016342  005737  003006                    TST     DONE            ;TEST IF INTERRUPT OCCURRED
1473  016346  001453                            BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
1474  016350  012766  006176  000002  1$:       MOV     #MTOSLOW,2(SP)  ;ELSE SET TOO SLOW ERROR POINTER
1475  016356  032737  002000  003044            BIT     #OPIERR,T.CS    ;TEST IF OPI SET
1476  016364  001403                            BEQ     2$              ;NO - SKIP
1477  016366  012766  006216  000002            MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
1478  016374  000207                    2$:     RTS     PC              ;RETURN
1479  016376  012737  000001  003142  4$:       MOV     #1,DLYCNT       ;INITIALIZE DELAY COUNT
1480  016404  006337  003142                    ASL     DLYCNT          ;MULTIPLY BY 2
1481  016410  006337  003142                    ASL     DLYCNT          ;MULTIPLY BY 2 AGAIN
1482  016414  012727  000012                    MOV     #10.,(PC)+      ;IMPLEMENT TIME DELAY LOOP
1483  016420  000000                            .WORD   0
1484  016422  013727  002116                    MOV     L$DLY,(PC)+
1485  016426  000000                            .WORD   0
1486  016430  005367  177772                    DEC     -6(PC)
1487  016434  001375                            BNE     .-4
1488  016436  005367  177756                    DEC     -22(PC)
1489  016442  001367                            BNE     .-20
1490  016444  032762  000200  000000            BIT     #CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
1491  016452  001006                            BNE     3$              ;YES - SKIP
1492  016454  004737  016266                    JSR     PC,READRL       ;READ RL REGS
1493  016460  012766  006271  000002            MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
1494  016466  000742                            BR      2$              ;SKIP
1495  016470  005737  003006          3$:       TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
1496  016474  001325                            BNE     1$              ;YES - SKIP TO SET TOO SLOW
1497  016476  004737  016266          5$:       JSR     PC,READRL       ;READ RL REGS
1498  016502  012766  006236  000002            MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
1499  016510  000731                            BR      2$              ;GO TO RETURN
1500
1501
1502
1503                                    ;       OPERATION AND TEST INITIALIZE ROUTINE
1504  016512  005037  003004          TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
1505  016516  105037  003363                    CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
1506  016522  005037  003014                    CLR     MORECE          ;RESET MORE COMPARE ERRORS
1507  016526  000207                            RTS     PC
1508
1509
1510
1511                                    ;       GET STATUS AND GET STATUS WITH RESET ROUTINE
1512  016530  013746  003126          GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
1513  016534  012737  000013  003126            MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
1514  016542  000412                            BR      GSTATG
1515  016544  013746  003126          GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
1516  016550  012737  000003  003126            MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
1517  016556  000404                            BR      GSTATG
1518  016560  013746  003126          GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
1519  016564  005037  003126                    CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
```

```
 1520  016570  010346                GSTATG: MOV     R3,-(SP)          ;STORE R3
 1521  016572  013703  003002                MOV     SSINDX,R3         ;GET SUBROUTINE INDEX
 1522  016576  005723                        TST     (R3)+             ;BUMP IT FOR NEXT ENTRY
 1523  016600  016663  000004  002404        MOV     4(SP),SUBSTK(R3)  ;INSERT THIS CALL
 1524  016606  162763  000004  002404        SUB     #4,SUBSTK(R3)     ;ADJUST IT TO CALLING LOCATION
 1525  016614  010337  003002                MOV     R3,SSINDX         ;STORE IT BACK
 1526  016620  010046                        MOV     R0,-(SP)          ;STORE R0
 1527  016622  010146                        MOV     R1,-(SP)          ;STORE R1
 1528  016624  012737  000002  003016        MOV     #2,ERRSWI         ;SET FOR NO ERROR RETURN
 1529  016632  032737  000010  003126        BIT     #DRSET,TEMP4      ;TEST IF DRIVE RESET
 1530  016640  001523                        BEQ     11$               ;NO - SKIP
 1531  016642  032762  040000  000000        BIT     #DRVERR,RLCS(R2)  ;TEST IF DRIVE ERROR SET
 1532  016650  001426                        BEQ     49$               ;NO - SKIP
 1533                          .LIST   ME
 1534  016652                        WAITMS  #1                ;WAIT FOR DRIVE TO SETTLE
  (1)  016652  012737  000001  003142        MOV     ##1,DLYCNT        ;INITIALIZE DELAY COUNTER
  (1)  016660  006337  003142                ASL     DLYCNT            ;MULTIPLY ARGUMENT BY 2
  (1)  016664  006337  003142                ASL     DLYCNT            ;MULTIPLY ARGUMENT BY 2 AGAIN
  (1)  016670                        64$:    DELAY   #250.             ;IMPLEMENT 25-MS TIME DELAY
  (2)  016670                        M$GNINS <MOV    ##250.,(PC)+>
  (3)  016670  012727  000372                MOV     ##250.,(PC)+
  (3)
  (2)  016674                        .MEXIT
  (3)  016674  000000                M$GNINS <.WORD  0>
  (3)                                        .WORD   0
  (2)  016676                        .MEXIT
  (3)  016676  013727  002116        M$GNINS <MOV    L$DLY,(PC)+>
  (3)                                        MOV     L$DLY,(PC)+
  (2)  016702                        .MEXIT
  (3)  016702  000000                M$GNINS <.WORD  0>
  (3)                                        .WORD   0
  (2)  016704                        .MEXIT
  (3)  016704  005367  177772        M$GNINS <DEC    -6(PC)>
  (3)                                        DEC     -6(PC)
  (2)  016710                        .MEXIT
  (3)  016710  001375                M$GNINS <BNE    .-4>
  (3)                                        BNE     .-4
  (2)  016712                        .MEXIT
  (3)  016712  005367  177756        M$GNINS <DEC    -22(PC)>
  (3)                                        DEC     -22(PC)
  (2)  016716                        .MEXIT
  (3)  016716  001367                M$GNINS <BNE    .-20>
  (3)                                        BNE     .-20
  (1)  016720  005337  003142        .MEXIT
  (1)  016724  001361                        DEC     DLYCNT            ;DECREMENT DELAY COUNT
                                             BNE     64$               ;BRANCH IF TIME DELAY NOT EXPIRED
 1535                          .NLIST  ME
 1536  016726  012701  000030        49$:    MOV     #24.,R1           ;INITIALIZE WAIT COUNTER
 1537  016732  004737  016560        50$:    JSR     PC,GSTAT          ;GET DRIVE STATUS
 1538  016736  017564                        3$
 1539  016740  032737  000001  003044        BIT     #DRDYMSK,T.CS     ;TEST IF DRIVE READY
 1540  016746  001076                        BNE     5$                ;YES - GO DO CLEAR
 1541  016750  032737  000020  003052        BIT     #HOSTAT,T.MP      ;ELSE TEST IF HEADS OUT
 1542  016756  001010                        BNE     51$               ;YES - BYPASS RELOAD WAIT FLAG SETTING
 1543  016760  032737  144000  003052        BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
 1544                                                                   ;THAT CAUSED HEADS TO
 1545                                                                   ;UNLOAD
```

```
1546  016766  001466                          BEQ     5$                ;NO - SKIP
1547  016770  052737  040000  003004          BIS     #RELDWT,OPFLAG    ;ELSE SET WAIT FLAG
1548  016776  000462                          BR      5$                ;SKIP TO CLEAR
1549  017000  032737  040000  003044  51$:    BIT     #DRVERR,T.CS      ;TEST IF DRIVE ERROR NOW
1550  017006  001056                          BNE     5$                ;YES - SKIP TO CLEAR
1551                                  .LIST   ME
1552  017010                                  WAITMS  #1                ;WAIT FOR DRIVE TO GET ERROR, READY, OR HEADS OUT
 (1)  017010  012737  000001  003142          MOV     ##1,DLYCNT        ;INITIALIZE DELAY COUNTER
 (1)  017016  006337  003142                  ASL     DLYCNT            ;MULTIPLY ARGUMENT BY 2
 (1)  017022  006337  003142                  ASL     DLYCNT            ;MULTIPLY ARGUMENT BY 2 AGAIN
 (1)  017026                          65$:    DELAY   #250.             ;IMPLEMENT 25-MS TIME DELAY
 (2)  017026                          M$GNINS <MOV    ##250.,(PC)+>
 (3)  017026  012727  000372                  MOV     ##250.,(PC)+
 (3)                                  .MEXIT
 (2)  017032                          M$GNINS <.WORD  0>
 (3)  017032  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  017034                          M$GNINS <MOV    L$DLY,(PC)+>
 (3)  017034  013727  002116                  MOV     L$DLY,(PC)+
 (3)                                  .MEXIT
 (2)  017040                          M$GNINS <.WORD  0>
 (3)  017040  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  017042                          M$GNINS <DEC    -6(PC)>
 (3)  017042  005367  177772                  DEC     -6(PC)
 (3)                                  .MEXIT
 (2)  017046                          M$GNINS <BNE    .-4>
 (3)  017046  001375                          BNE     .-4
 (3)                                  .MEXIT
 (2)  017050                          M$GNINS <DEC    -22(PC)>
 (3)  017050  005367  177756                  DEC     -22(PC)
 (3)                                  .MEXIT
 (2)  017054                          M$GNINS <BNE    .-20>
 (3)  017054  001367                          BNE     .-20
 (3)                                  .MEXIT
 (1)  017056  005337  003142                  DEC     DLYCNT            ;DECREMENT DELAY COUNT
 (1)  017062  001361                          BNE     65$               ;BRANCH IF TIME DELAY NOT EXPIRED
1553                                  .NLIST  ME
1554  017064  005301                          DEC     R1                ;DEC WAIT COUNTER
1555  017066  001321                          BNE     50$               ;IF NOT DONE, LOOP
1556  017070  012703  010757                  MOV     #MUNDEF,R3        ;MESSAGE FOR UNDEFINED STATE
1557  017074                                  ERRHRD  10001.,,ERR1
 (4)  017074  104456                          TRAP    C$ERHRD
 (5)  017076  023421                          .WORD   10001
 (5)  017100  000000                          .WORD   0
 (5)  017102  012340                          .WORD   ERR1
1558  017104  000137  017560                  JMP     14$               ;EXIT
1559  017110  005737  003126          11$:    TST     TEMP4             ;TEST IF SAVE REGISTERS
1560  017114  001013                          BNE     5$                ;NO SKIP
1561  017116  012701  000004                  MOV     #4,R1             ;SET SAVE COUNT
1562  017122  012703  003044                  MOV     #L.MP+2,R3        ;SET ADDRESS OF FIRST SAVE
1563  017126  014346                  8$:     MOV     -(R3),-(SP)       ;PUT REG ON STACK
1564  017130  005301                          DEC     R1                ;DEC COUNT
1565  017132  001375                          BNE     8$                ;LOOP UNTIL ALL SAVED
1566  017134  012737  000003  003040          MOV     #GETSTAT,L.DA     ;SET FOR GET STATUS
1567  017142  000403                          BR      6$                ;SKIP
```

```
1568  017144  013737  003126  003040  5$:     MOV     TEMP4,L.DA      ;INSERT PRESET FOR STATUS
1569  017152                          6$:
1570  017152  005037  003006                  CLR     DONE            ;CLEAR INTERRUPT FLAG
1571  017156  013737  003032  003034          MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
1572  017164  042737  002000  003034          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1573  017172  052737  000104  003034          BIS     #GTSTAT,L.CS
1574  017200  013762  003040  000004          MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
1575  017206  013762  003034  000000          MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
1576  017214                                  WAITUS  #1              ;WAIT 100 US FOR INTERRUPT
 (3)  017214  012727  000001                  MOV     ###1,(PC)+
 (3)  017220  000000                          .WORD   0
 (3)  017222  013727  002116                  MOV     L$DLY,(PC)+
 (3)  017226  000000                          .WORD   0
 (3)  017230  005367  177772                  DEC     -6(PC)
 (3)  017234  001375                          BNE     .-4
 (3)  017236  005367  177756                  DEC     -22(PC)
 (3)  017242  001367                          BNE     .-20
1577  017244  005737  003006                  TST     DONE            ;CHECK IF INTERRUPT OCCURRED
1578  017250  001534                          BEQ     1$              ;NO - SKIP
1579  017252  013737  003052  003060  4$:     MOV     T.MP,T.STAT     ;STORE MP REGISTER
1580  017260  042737  177770  003060          BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
1581  017266  032737  000010  003040          BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
1582  017274  001533                          BEQ     3$              ;NO - SKIP TO EXIT
1583  017276  032737  040000  003004          BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
1584  017304  001450                          BEQ     12$             ;NO - SKIP
1585  017306  012701  000144                  MOV     #100.,R1        ;INITIALIZE WAIT COUNTER
1586  017312  032762  000001  000000  13$:     BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
1587  017320  001042                          BNE     12$             ;YES - SKIP
1588                                  .LIST   ME
1589  017322                                  WAITMS  #1              ;CALL WAIT
 (1)  017322  012737  000001  003142          MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNTER
 (1)  017330  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
 (1)  017334  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
 (1)  017340                          66$:    DELAY   #250.           ;IMPLEMENT 25-MS TIME DELAY
 (2)  017340                          M$GNINS <MOV    ##250.,(PC)+>
 (3)  017340  012727  000372                  MOV     ##250.,(PC)+
 (3)                                  .MEXIT
 (2)  017344                          M$GNINS <.WORD  0>
 (3)  017344  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  017346                          M$GNINS <MOV    L$DLY,(PC)+>
 (3)  017346  013727  002116                  MOV     L$DLY,(PC)+
 (3)                                  .MEXIT
 (2)  017352                          M$GNINS <.WORD  0>
 (3)  017352  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  017354                          M$GNINS <DEC    -6(PC)>
 (3)  017354  005367  177772                  DEC     -6(PC)
 (3)                                  .MEXIT
 (2)  017360                          M$GNINS <BNE    .-4>
 (3)  017360  001375                          BNE     .-4
 (3)                                  .MEXIT
 (2)  017362                          M$GNINS <DEC    -22(PC)>
 (3)  017362  005367  177756                  DEC     -22(PC)
 (3)                                  .MEXIT
 (2)  017366                          M$GNINS <BNE    .-20>
```

```
   (3)  017366  001367                          BNE     .-20
   (3)                                .MEXIT
   (1)  017370  005337  003142                  DEC     DLYCNT          ;DECREMENT DELAY COUNT
   (1)  017374  001361                          BNE     66$             ;BRANCH IF TIME DELAY NOT EXPIRED
 1590                                 .NLIST  ME
 1591  017376  005301                          DEC     R1              ;DEC COUNT
 1592  017400  001344                          BNE     13$             ;LOOP IF NOT 0
 1593  017402  004737  016560                  JSR     PC,GSTAT        ;GET DRIVE STATUS
 1594  017406  017564                          3$                      ;ERROR RETURN
 1595  017410  012703  011024                  MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
 1596  017414                                   ERRHRD  10003.,,ERR1
   (4)  017414  104456                        TRAP    C$ERHRD
   (5)  017416  023423                        .WORD   10003
   (5)  017420  000000                        .WORD   0
   (5)  017422  012340                        .WORD   ERR1
 1597  017424  000455                          BR      14$             ;GO TO EXIT
 1598  017426                           12$:    WAITUS  #5              ;WAIT
   (3)  017426  012727  000005                MOV     ###5,(PC)+
   (3)  017432  000000                        .WORD   0
   (3)  017434  013727  002116                MOV     L$DLY,(PC)+
   (3)  017440  000000                        .WORD   0
   (3)  017442  005367  177772                DEC     -6(PC)
   (3)  017446  001375                        BNE     .-4
   (3)  017450  005367  177756                DEC     -22(PC)
   (3)  017454  001367                        BNE     .-20
 1599  017456  004737  016560                  JSR     PC,GSTAT        ;GET DRIVE STATUS
 1600  017462  017564                          3$
 1601  017464  032737  100000  003044          BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
 1602  017472  001434                          BEQ     3$              ;NO - SKIP
 1603  017474  032737  001000  003052          BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
 1604  017502  001403                          BEQ     7$              ;YES SKIP
 1605  017504  012703  006325                  MOV     #VCNRST,R3      ;SET REASON POINTER
 1606  017510  000417                          BR      2$              ;EXIT
 1607  017512  032737  040000  003044  7$:     BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
 1608  017520  001405                          BEQ     9$              ;NO - SKIP
 1609  017522                                   ERRHRD  10004.,,ERR6
   (4)  017522  104456                        TRAP    C$ERHRD
   (5)  017524  023424                        .WORD   10004
   (5)  017526  000000                        .WORD   0
   (5)  017530  012642                        .WORD   ERR6
 1610  017532  000412                          BR      14$             ;EXIT
 1611  017534  012703  006346          9$:     MOV     #UNXERR,R3      ;SET REASON POINTER
 1612  017540  000403                          BR      2$              ;EXIT
 1613  017542  004737  016320          1$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 1614  017546  012603                          MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
 1615  017550                           2$:     ERRHRD  10002.,,ERR1
   (4)  017550  104456                        TRAP    C$ERHRD
   (5)  017552  023422                        .WORD   10002
   (5)  017554  000000                        .WORD   0
   (5)  017556  012340                        .WORD   ERR1
 1616  017560  005037  003016          14$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 1617  017564  005737  003126          3$:     TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 1618  017570  001007                          BNE     22$             ;NO - SKIP
 1619  017572  012703  003034                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE
 1620  017576  012701  000004                  MOV     #4,R1           ;SET REGISTER COUNT
 1621  017602  012623                  20$:    MOV     (SP)+,(R3)+     ;RESTORE REG
```

```
1622  017604  005301                          DEC     R1              ;DEC COUNT
1623  017606  001375                          BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
1624  017610  162737  000002  003002  22$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUTINE STACK
1625  017616  012601                          MOV     (SP)+,R1        ;RESTORE R1
1626  017620  012600                          MOV     (SP)+,R0        ;RESTORE R0
1627  017622  012603                          MOV     (SP)+,R3        ;RESTORE R3
1628  017624  012637  003126                  MOV     (SP)+,TEMP4     ;RESTORE TEMP4
1629  017630  005737  003016                  TST     ERRSWI          ;TEST IF ERROR RETURN
1630  017634  001403                          BEQ     99$             ;YES - SKIP
1631  017636  063716  003016                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1632  017642  000207                          RTS     PC
1633  017644  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1634  017650  000207                          RTS     PC
1635
1636
1637
1638                                  ;       GET DRIVE STATE ROUTINE
1639  017652  010346                  GDRSTA: MOV     R3,-(SP)        ;SAVE R3
1640  017654  012701  000004                  MOV     #4,R1           ;INITIALIZE REGISTER SAVE COUNT
1641  017660  012703  003044                  MOV     #L.MP+2,R3      ;INITIALIZE ADDRESS OF FIRST SAVE
1642  017664  014346                  1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER ON STACK
1643  017666  005301                          DEC     R1              ;DECREMENT REGISTER SAVE COUNT
1644  017670  001375                          BNE     1$              ;LOOP UNTIL ALL 4 REGISTERS ARE SAVED
1645  017672  012737  000003  003040          MOV     #GETSTAT,L.DA   ;SET UP DISK ADDRESS REGISTER FOR GET STATUS
1646                                                                  ;/COMMAND
1647  017700  005037  003006                  CLR     DONE            ;CLEAR INTERRUPT FLAG
1648  017704  013737  003032  003034          MOV     RLDRV,L.CS      ;SET UP CONTROL STATUS REGISTER WITH
1649                                                                  ;/DRIVE NUMBER
1650  017712  042737  002000  003034          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVES 4-7 SPECIFIED
1651  017720  052737  000104  003034          BIS     #GTSTAT,L.CS    ;INITIALIZE CONTROL STATUS REGISTER FOR
1652                                                                  ;/GET STATUS COMMAND
1653  017726  013762  003040  000004          MOV     L.DA,RLDA(R2)   ;INITIALIZE DISK ADDRESS REGISTER FOR
1654                                                                  ;/GET STATUS COMMAND
1655  017734  013762  003034  000000          MOV     L.CS,RLCSR(R2)  ;LOAD CONTROL STATUS REGISTER TO EXECUTE
1656                                                                  ;/GET STATUS COMMAND
1657  017742  105762  000C 0          5$:     TSTB    RLCS(R2)        ;WAIT FOR CONTROLLER READY INDICATING
1658  017746  001775                          BEQ     5$              ;/RECEIPT OF GET STATUS COMMAND
1659  017750  005737  003006                  TST     DONE            ;INTERRUPT OCCURRED?
1660  017754  001416                          BEQ     3$              ;BRANCH IF NOT
1661  017756  013737  003052  003060          MOV     T.MP,T.STAT     ;GET CONTENTS OF MULTI-PURPOSE REGISTER
1662  017764  042737  177770  003060          BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE DRIVE BITS
1663  017772  012703  003034                  MOV     #L.CS,R3        ;INITIALIZE POINTER TO RESTORE RL REGISTERS
1664  017776  012701  000004                  MOV     #4,R1           ;INITIALIZE REGISTER SAVE COUNT
1665  020002  012623                  2$:     MOV     (SP)+,(R3)+     ;RESTORE REGISTERS
1666  020004  005301                          DEC     R1              ;DECREMENT REGISTER SAVE COUNT
1667  020006  001375                          BNE     2$              ;LOOP UNTIL ALL 4 REGISTERS ARE RESTORED
1668  020010  000402                          BR      4$
1669  020012  004737  016320          3$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
1670  020016  012603                  4$:     MOV     (SP)+,R3        ;RESTORE R3
1671  020020  000207                          RTS     PC              ;RETURN
1672
1673
1674
1675                                  ;       SEEK ROUTINE
1676  020022  012737  177777  003120  XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
1677  020030  000402                          BR      XSEEK1
```

```
1678  020032  005037  003120          XSEEK:  CLR     TEMP1              ;CLEAR SPECIAL TIMING SEEK FLAG
1679  020036  010346                  XSEEK1: MOV     R3,-(SP)           ;STORE R3
1680  020040  013703  003002                  MOV     SSINDX,R3          ;GET SUBROUTINE INDEX
1681  020044  005723                          TST     (R3)+              ;BUMP IT FOR NEXT ENTRY
1682  020046  016663  000002  002404          MOV     2(SP),SUBSTK(R3)   ;INSERT THIS CALL
1683  020054  162763  000004  002404          SUB     #4,SUBSTK(R3)      ;ADJUST IT TO CALLING LOCATION
1684  020062  010337  003002                  MOV     R3,SSINDX          ;STORE IT BACK
1685  020066  010046                          MOV     R0,-(SP)
1686  020070  010146                          MOV     R1,-(SP)
1687  020072  010546                          MOV     R5,-(SP)           ;STORE REG
1688  020074  012737  000002  003016          MOV     #2,ERRSWI          ;SET FOR NO ERROR RETURN
1689  020102  005037  003076                  CLR     DIFAUG             ;CLEAR DIFFERENCE ARGUMENT (FOR SEEKING
1690                                                                     ;    PAST GUARD BAND)
1691  020106  004737  022662                  JSR     PC,GETPOS          ;GET PRESENT POSITION
1692  020112  020562                          65$
1693  020114  013737  003104  003100          MOV     CURCYL,OLDCYL      ;MOVE CURRENT TO OLD CYLINDER
1694  020122  023737  003102  002302          CMP     NEWCYL,HLMTW       ;TEST IF NEW IS GREATER THAN 255
1695  020130  003427                          BLE     3$                 ;NO - SKIP
1696  020132  163737  002302  003102          SUB     HLMTW,NEWCYL       ;ELSE SUBTRACT 255.
1697  020140  013737  003102  003076          MOV     NEWCYL,DIFAUG      ;STORE DIFFERENCE AS ARGUMENT
1698  020146  013737  002302  003102          MOV     HLMTW,NEWCYL       ;SET NEWCYL AS 255.
1699  020154  022737  000001  002276          CMP     #1,T.DRIVE
1700  020162  001424                          BEQ     6$
1701  020164  162737  000001  003102          SUB     #1,NEWCYL
1702  020172  012737  000001  003110          MOV     #1,DESSGN
1703  020200  012737  000001  003106          MOV     #1,DESDIF
1704  020206  000451                          BR      18$
1705  020210  005737  003102          3$:     TST     NEWCYL             ;TEST IF NEWCYL HAS NEGATIVE VALUE
1706  020214  100007                          BPL     6$                 ;NO - SKIP
1707  020216  005437  003102                  NEG     NEWCYL             ;ELSE MAKE IT POSITIVE
1708  020222  013737  003102  003076          MOV     NEWCYL,DIFAUG      ;AND STORE IT AS ARGUMENT
1709  020230  005037  003102                  CLR     NEWCYL             ;AND SET NEWCYL TO 0
1710  020234  013705  003104          6$:     MOV     CURCYL,R5          ;COMPUTE DIFFERENCE AND NEW CYLINDER
1711  020240  163705  003102                  SUB     NEWCYL,R5          ;SUB NEWCYL FROM CURCYL
1712  020244  100005                          BPL     13$                ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
1713  020246  012737  000001  003110          MOV     #1,DESSGN          ;ELSE SET SIGN FOR FORWARD
1714  020254  005405                          NEG     R5                 ;MAKE DIFFERENCE POSITIVE
1715  020256  000402                          BR      14$                ;SKIP
1716  020260  005037  003110          13$:    CLR     DESSGN             ;SET SIGN FOR REVERSE
1717  020264  010537  003106          14$:    MOV     R5,DESDIF          ;STORE DIFFERENCE
1718  020270  005737  003076                  TST     DIFAUG             ;IS THERE A DIFFERENCE ARGUMENT
1719  020274  001416                          BEQ     18$                ;NO - SKIP
1720  020276  023737  003102  002302          CMP     NEWCYL,HLMTW       ;CHECK IF NEW CYL IS 255.
1721  020304  001007                          BNE     17$                ;NO - SKIP
1722  020306  012737  000001  003110          MOV     #1,DESSGN          ;ELSE FORCE SIGN FOR FORWARD
1723                                                                     ;(INNER GUARD BAND)
1724  020314  022737  000001  002276          CMP     #1,T.DRIVE
1725  020322  001003                          BNE     18$
1726  020324  063737  003076  003106  17$:    ADD     DIFAUG,DESDIF
1727  020332                          18$:
1728  020332  012705  003034                  MOV     #L.CS,R5           ;GET RL REG ADDRESS
1729  020336  012715  000106                  MOV     #SEEK,(R5)         ;SET FOR SEEK
1730  020342  053715  003032                  BIS     RLDRV,(R5)         ;INSERT DRIVE NUMBER
1731  020346  042725  002000                  BIC     #BIT10,(R5)+       ;CLEAR IF DRIVE 4 - 7 SPEC'D
1732  020352  005025                          CLR     (R5)+              ;CLEAR BUS ADDRESS
1733  020354  013715  003106                  MOV     DESDIF,(R5)        ;LOAD DIFFERENCE
```

```
1734  020360  012700  000007          MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
1735  020364  006315              21$:  ASL     (R5)
1736  020366  005300                    DEC     R0
1737  020370  001375                    BNE     21$             ;LOOP UNTIL ALIGNED
1738  020372  005737  003110            TST     DESSGN          ;TEST SIGN
1739  020376  001402                    BEQ     23$             ;SKIP IF 0
1740  020400  052715  000004            BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
1741  020404  005737  003112      23$:  TST     DESHD           ;TEST IF HEAD 0
1742  020410  001402                    BEQ     25$             ;YES - SKIP
1743  020412  052715  000020            BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
1744  020416  052725  000001      25$:  BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
1745  020422  004737  021132            JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
1746  020426  020562                    65$
1747  020430  005037  003006            CLR     DONE            ;CLEAR INTERRUPT FLAG
1748  020434  005737  003120            TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
1749  020440  001050                    BNE     65$             ;YES - SKIP, DO NOT START SEEK
1750  020442  014562  000004            MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
1751  020446  014562  000002            MOV     -(R5),RLBA(R2)
1752  020452  014562  000000            MOV     -(R5),RLCS(R2)  ;PERFORM SEEK OPERATION
1753  020456                      30$:  WAITUS  #1              ;ALLOW TIME FOR RECEIPT OF SEEK COMMAND
 (3)  020456  012727  000001            MOV     ###1,(PC)+
 (3)  020462  000000                    .WORD   0
 (3)  020464  013727  002116            MOV     L$DLY,(PC)+
 (3)  020470  000000                    .WORD   0
 (3)  020472  005367  177772            DEC     -6(PC)
 (3)  020476  001375                    BNE     .-4
 (3)  020500  005367  177756            DEC     -22(PC)
 (3)  020504  001367                    BNE     .-20
1754  020506  005737  003006            TST     DONE            ;TEST IF INTERRUPT DONE
1755  020512  001012                    BNE     32$             ;YES - SKIP
1756  020514  004737  016320            JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
1757  020520  012603                    MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
1758  020522                            ERRHRD  10005.,,ERR1
 (4)  020522  104456                    TRAP    C$ERHRD
 (5)  020524  023425                    .WORD   10005
 (5)  020526  000000                    .WORD   0
 (5)  020530  012340                    .WORD   ERR1
1759  020532  005037  003016            CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1760  020536  000411                    BR      65$
1761  020540  005737  003044      32$:  TST     T.CS            ;TEST IF ANY ERROR
1762  020544  100006                    BPL     65$             ;NO - SKIP
1763  020546                            ERRHRD  10006.,,ERR6
 (4)  020546  104456                    TRAP    C$ERHRD
 (5)  020550  023426                    .WORD   10006
 (5)  020552  000000                    .WORD   0
 (5)  020554  012642                    .WORD   ERR6
1764  020556  005037  003016            CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1765  020562  162737  000002  003002  65$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUTINE STACK
1766  020570  012605                    MOV     (SP)+,R5        ;RESTORE REGISTER
1767  020572  012601                    MOV     (SP)+,R1
1768  020574  012600                    MOV     (SP)+,R0
1769  020576  012603                    MOV     (SP)+,R3        ;RESTORE R3
1770  020600  005737  003016            TST     ERRSWI          ;TEST IF ERROR RETURN
1771  020604  001403                    BEQ     99$             ;YES - SKIP
1772  020606  063716  003016            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1773  020612  000207                    RTS     PC
```

```
1774  020614  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1775  020620  000207                          RTS     PC
1776
1778
1779
1780  020622  010346                  SIMSEK: MOV     R3,-(SP)        ;STORE REGISTERS
1781  020624  013703  003002                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1782  020630  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1783  020632  016663  000002  002404          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1784  020640  162763  000004  002404          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1785  020646  010337  003002                  MOV     R3,SSINDX       ;STORE IT BACK
1786  020652  010046                          MOV     R0,-(SP)
1787  020654  010446                          MOV     R4,-(SP)
1788  020656  012737  000002  003016          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1789  020664  004737  021132                  JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
1790  020670  021074                          65$
1791  020672  012704  003034                  MOV     #L.CS,R4        ;GET POINTER TO L REGS
1792  020676  012714  000106                  MOV     #SEEK,(R4)      ;SET FOR SEEK
1793  020702  053714  003032                  BIS     RLDRV,(R4)      ;INSERT DRIVE NUMBER
1794  020706  042724  002000                  BIC     #BIT10,(R4)+    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
1795  020712  005024                          CLR     (R4)+           ;CLEAR BUS ADDRESS
1796  020714  013714  003106                  MOV     DESDIF,(R4)     ;LOAD DIFFERENCE
1797  020720  012703  000007                  MOV     #7,R3           ;SET COUNT FOR SHIFT TO ALIGN
1798  020724  006314                  3$:     ASL     (R4)            ;ALIGN DIFFERENCE IN DA
1799  020726  005303                          DEC     R3
1800  020730  001375                          BNE     3$
1801  020732  005737  003110                  TST     DESSGN          ;TEST IF SIGN SET
1802  020736  001402                          BEQ     5$              ;NO - SKIP
1803  020740  052714  000004                  BIS     #DIRBIT,(R4)    ;INSERT SIGN
1804  020744  005737  003112          5$:     TST     DESHD           ;TEST IF HEAD 0
1805  020750  001402                          BEQ     7$              ;YES - SKIP
1806  020752  052714  000020                  BIS     #HDSEL,(R4)     ;INSERT HEAD BIT
1807  020756  052724  000001          7$:     BIS     #MBSET0,(R4)+   ;INSERT MARKER BIT
1808  020762  005037  003006                  CLR     DONE            ;CLEAR INTERRUPT FLAG
1809  020766  012701  000012                  MOV     #10.,R1         ;SET WAIT COUNT FOR 800US
1810  020772  014462  000004                  MOV     -(R4),RLDA(R2)  ;LOAD RL REGISTERS
1811  020776  014462  000002                  MOV     -(R4),RLBA(R2)
1812  021002  014462  000000                  MOV     -(R4),RLCS(R2)
1813  021006  005737  003006          10$:    TST     DONE            ;CHECK IF INTERRUPTED
1814  021012  001030                          BNE     65$             ;YES - SKIP
1815  021014  005301                          DEC     R1              ;DEC WAIT COUNT
1816  021016  001415                          BEQ     13$             ;IF 0 - SKIP
1817  021020                                  WAITUS  #1
  (3) 021020  012727  000001                  MOV     ###1,(PC)+
  (3) 021024  000000                          .WORD   0
  (3) 021026  013727  002116                  MOV     L$DLY,(PC)+
  (3) 021032  000000                          .WORD   0
  (3) 021034  005367  177772                  DEC     -6(PC)
  (3) 021040  001375                          BNE     .-4
  (3) 021042  005367  177756                  DEC     -22(PC)
  (3) 021046  001367                          BNE     .-20
1818  021050  000756                          BR      10$             ;GO CHECK DONE
1819  021052  004737  016320          13$:    JSR     PC,WAITIN       ;GO WAIT FOR TIMEOUT
1820  021056  012603                          MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
1821  021060                                  ERRHRD  10011,,ERR1
  (4) 021060  104456                          TRAP    C$ERHRD
```

```
   (5)  021062  023433                          .WORD   10011
   (5)  021064  000000                          .WORD   0
   (5)  021066  012340                          .WORD   ERR1
1822    021070  005037  003016                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
1823    021074                          14$:
1824    021074  162737  000002  003002  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1825    021102  012604                          MOV     (SP)+,R4        ;RESTORE REGS
1826    021104  012600                          MOV     (SP)+,R0
1827    021106  012603                          MOV     (SP)+,R3
1828    021110  005737  003016                  TST     ERRSWI          ;TEST IF ERROR RETURN
1829    021114  001403                          BEQ     99$             ;YES - SKIP
1830    021116  063716  003016                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1831    021122  000207                          RTS     PC
1832    021124  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
1833    021130  000207                          RTS     PC
1835
1911
1912
1913                                    ;       DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
1914                                    ;       500MS FOR READY TO SET.
1915    021132  010346                  RDYCHK: MOV     R3,-(SP)        ;STORE REGS
1916    021134  013703  003002                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
1917    021140  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
1918    021142  016663  000002  002404          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1919    021150  162763  000004  002404          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1920    021156  010337  003002                  MOV     R3,SSINDX       ;STORE IT BACK
1921    021162  010046                          MOV     R0,-(SP)
1922    021164  010146                          MOV     R1,-(SP)
1923    021166  010446                          MOV     R4,-(SP)
1924    021170  012737  000002  003016          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
1925    021176  012701  011610                  MOV     #5000.,R1       ;SET WAIT COUNT
1926    021202  004737  016560          1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
1927    021206  021422                          4$
1928    021210  032737  000001  003044          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1929    021216  001103                          BNE     5$              ;YES - EXIT
1930    021220                                  WAITUS  #1
   (3)  021220  012727  000001                  MOV     ###1,(PC)+
   (3)  021224  000000                          .WORD   0
   (3)  021226  013727  002116                  MOV     L$DLY,(PC)+
   (3)  021232  000000                          .WORD   0
   (3)  021234  005367  177772                  DEC     -6(PC)
   (3)  021240  001375                          BNE     .-4
   (3)  021242  005367  177756                  DEC     -22(PC)
   (3)  021246  001367                          BNE     .-20
1931    021250  005301                          DEC     R1              ;DEC WAIT COUNT
1932    021252  001353                          BNE     1$              ;LOOP IF NOT 0
1933    021254  012703  010260                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
1934    021260  012704  011260                  MOV     #C500MS,R4      ;SET CONDITION MESSAGE POINTER
1935    021264                                  ERRHRD  10010.,,ERR5
   (4)  021264  104456                          TRAP    C$ERHRD
   (5)  021266  023432                          .WORD   10010
   (5)  021270  000000                          .WORD   0
   (5)  021272  012572                          .WORD   ERR5
1936    021274  012701  000030                  MOV     #24.,R1         ;INITIALIZE WAIT COUNT
1937    021300  004737  016560          2$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
1938    021304  021422                          4$
```

```
1939  021306  032737  000001  003044          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
1940  021314  001030                          BNE     3$              ;YES - SKIP
1941                                  .LIST   ME
1942  021316                                  WAITMS  #1              ;WAIT FOR 100MS
 (1)  021316  012737  000001  003142          MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNTER
 (1)  021324  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
 (1)  021330  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
 (1)  021334                          64$:    DELAY   #250.           ;IMPLEMENT 25-MS TIME DELAY
 (2)  021334                          M$GNINS <MOV    ##250.,(PC)+>
 (3)  021334  012727  000372                  MOV     ##250.,(PC)+
 (3)                                  .MEXIT
 (2)  021340                          M$GNINS <.WORD  0>
 (3)  021340  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  021342                          M$GNINS <MOV    L$DLY,(PC)+>
 (3)  021342  013727  002116                  MOV     L$DLY,(PC)+
 (3)                                  .MEXIT
 (2)  021346                          M$GNINS <.WORD  0>
 (3)  021346  000000                          .WORD   0
 (3)                                  .MEXIT
 (2)  021350                          M$GNINS <DEC    -6(PC)>
 (3)  021350  005367  177772                  DEC     -6(PC)
 (3)                                  .MEXIT
 (2)  021354                          M$GNINS <BNE    .-4>
 (3)  021354  001375                          BNE     .-4
 (3)                                  .MEXIT
 (2)  021356                          M$GNINS <DEC    -22(PC)>
 (3)  021356  005367  177756                  DEC     -22(PC)
 (3)                                  .MEXIT
 (2)  021362                          M$GNINS <BNE    .-20>
 (3)  021362  001367                          BNE     .-20
 (3)                                  .MEXIT
 (1)  021364  005337  003142                  DEC     DLYCNT          ;DECREMENT DELAY COUNT
 (1)  021370  001361                          BNE     64$             ;BRANCH IF TIME DELAY NOT EXPIRED
1943                                  .NLIST  ME
1944  021372  005301                          DEC     R1              ;DEC WAIT COUNTER
1945  021374  001341                          BNE     2$              ;LOOP UNTIL TIME DONE
1946  021376  032737  100000  003044  3$:     BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
1947  021404  001406                          BEQ     4$              ;NO - SKIP
1948  021406                                  ERRHRD  10011.,,ERR6    ;REPORT ALL ERRORS
 (4)  021406  104456                          TRAP    C$ERHRD
 (5)  021410  023433                          .WORD   10011
 (5)  021412  000000                          .WORD   0
 (5)  021414  012642                          .WORD   ERR6
1949  021416  005337  003156                  DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
1950  021422  005037  003016          4$:     CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
195   021426  162737  000002  003002  5$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
1952  021434  012604                          MOV     (SP)+,R4        ;RESTORE REGS
1953  021436  012601                          MOV     (SP)+,R1
1954  021440  012600                          MOV     (SP)+,R0
1955  021442  012603                          MOV     (SP)+,R3
1956  021444  005737  003016                  TST     ERRSWI          ;TEST IF ERROR RETURN
1957  021450  001403                          BEQ     99$             ;YES - SKIP
1958  021452  063716  003016                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
1959  021456  000207                          RTS     PC
1960  021460  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
```

```
1961  021464  000207                          RTS     PC
1962
1963                                  ;       CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
1964                                  ;       SELECTED BY SOFTWARE PARAMETER.
1965  021466  005037  003112          CHOSHD: CLR     DESHD           ;CLEAR TO HEAD 0
1966  021472  032737  010000  014202          BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1967  021500  001403                          BEQ     1$              ;NO - SKIP
1968  021502  013737  014210  003112          MOV     HEADW,DESHD     ;INSERT SPECIFIED HEAD
1969  021510  000207                  1$:     RTS     PC
1970
1971
1972
1973                                  ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
1974                                  ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
1975  021512  032737  010000  014202  SWAPHD: BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPECIFIED
1976  021520  001011                          BNE     2$              ;YES - TAKE ABORT EXIT
1977  021522  005737  003112                  TST     DESHD           ;TEST IF HEAD ONE USED
1978  021526  001006                          BNE     2$              ;YES - TAKE ABORT EXIT
1979  021530  012737  000001  003112          MOV     #1,DESHD        ;ELSE SET FOR HEAD ONE
1980  021536  062716  000002                  ADD     #2,(SP)         ;BUMP PAST ABORT RETURN
1981  021542  000207                          RTS     PC              ;RETURN
1982  021544  017616  000000          2$:     MOV     @(SP),(SP)      ;GET ABORT DESTINATION
1983  021550  000207                  3$:     RTS     PC
1984
1985
1986
1987                                  ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
1988  021552  010046                  ONSWAP: MOV     R0,-(SP)        ;STORE R0
1989  021554  013700  003100                  MOV     OLDCYL,R0       ;MOVE OLD TO R0
1990  021560  013737  003102  003100          MOV     NEWCYL,OLDCYL   ;MOVE NEW TO OLD
1991  021566  010037  003102                  MOV     R0,NEWCYL       ;PUT OLD IN NEW
1992  021572  012600                          MOV     (SP)+,R0        ;RESTORE R0
1993  021574  000207                          RTS     PC
1994
2009
2010
2011                                  ;       READ HEADERS ROUTINE.
2012  021576  012737  000001  003126  XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
2013  021604  000402                          BR      XRDHDG          ;GO DO IT
2014  021606  005037  003126          XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AND L. REGS
2015  021612  010346                  XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
2016  021614  013703  003002                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2017  021620  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2018  021622  016663  000002  002404          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2019  021630  162763  000004  002404          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2020  021636  010337  003002                  MOV     R3,SSINDX       ;STORE IT BACK
2021  021642  010046                          MOV     R0,-(SP)
2022  021644  010146                          MOV     R1,-(SP)
2023  021646  010446                          MOV     R4,-(SP)
2024  021650  012737  000002  003016          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
2025  021656  005737  003126                  TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
2026  021662  001007                          BNE     2$              ;NO - SKIP
2027  021664  012703  003044                  MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
2028  021670  012701  000004                  MOV     #4,R1           ;SET COUNT
2029  021674  014346                  1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
2030  021676  005301                          DEC     R1              ;DEC COUNT
```

```
2031  021700  001375                        BNE     1$                      ;LOOP UNTIL ALL ARE SAVED
2032  021702  004737  021132          2$:   JSR     PC,RDYCHK               ;CHECK DRIVE READY
2033  021706  022174                        65$
2034  021710  005037  003006                CLR     DONE                    ;CLEAR INTERRUPT FLAG
2035  021714  012701  003034                MOV     #L.CS,R1                ;GET ADDRESS OF LOAD REGS
2036  021720  013711  003032                MOV     RLDRV,(R1)              ;LOAD DRIVE NUMBER
2037  021724  042711  002000                BIC     #BIT10,(R1)             ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2038  021730  052721  000110                BIS     #RDHEAD,(R1)+           ;INSERT COMMAND
2039  021734  005021                        CLR     (R1)+                   ;CLEAR BA
2040  021736  005021                        CLR     (R1)+                   ;CLEAR DA
2041  021740  014162  000004                MOV     -(R1),RLDA(R2)          ;LOAD RL11 REGS
2042  021744  014162  000002                MOV     -(R1),RLBA(R2)
2043  021750  014162  000000                MOV     -(R1),RLCSR(R2)
2044  021754                          3$:   WAITUS  #10.                    ;WAIT 1 MS FOR INTERRUPT
  (3) 021754  012727  000012                MOV     ###10.,(PC)+
  (3) 021760  000000                        .WORD   0
  (3) 021762  013727  002116                MOV     L$DLY,(PC)+
  (3) 021766  000000                        .WORD   0
  (3) 021770  005367  177772                DEC     -6(PC)
  (3) 021774  001375                        BNE     .-4
  (3) 021776  005367  177756                DEC     -22(PC)
  (3) 022002  001367                        BNE     .-20
2045  022004  005737  003006                TST     DONE                    ;TEST IF INTERRUPT FLAG SET
2046  022010  001460                        BEQ     14$                     ;NO - SKIP
2047  022012  032737  000001  003044  5$:   BIT     #DRDYMSK,T.CS           ;TEST IF DRIVE READY
2048  022020  001035                        BNE     10$                     ;YES - SKIP
2049  022022  012703  010260                MOV     #MDRDY,R3               ;SET NO READY MESSAGE
2050  022026  012704  011277                MOV     #CAFDT,R4               ;CONDITION OF AFTER DATA XFER
2051  022032                                ERRHRD  10017.,,ERR5
  (4) 022032  104456                        TRAP    C$ERHRD
  (5) 022034  023441                        .WORD   10017
  (5) 022036  000000                        .WORD   0
  (5) 022040  012572                        .WORD   ERR5
2052  022042  012701  000030                MOV     #24.,R1                 ;INITIALIZE WAIT COUNT
2053  022046  004737  016560          4$:   JSR     PC,GSTAT                ;GET STATUS
2054  022052  022170                        60$
2055  022054  032737  000001  003044        BIT     #DRDYMSK,T.CS           ;TEST IF DRIVE HAS COME READY
2056  022062  001403                        BEQ     11$                     ;NO - SKIP
2057  022064  005037  003016                CLR     ERRSWI                  ;CLEAR ERROR SWITCH
2058  022070  000411                        BR      10$                     ;SKIP
2059  022072  005301                  11$:  DEC     R1                      ;DEC WAIT COUNT
2060  022074  001364                        BNE     4$                      ;LOOP UNTIL TIME DONE
2061  022076  012704  011310                MOV     #C5SEC,R4               ;SET CONDITION AFTER 5 SECONDS
2062  022102                                ERRHRD  10014.,,ERR5
  (4) 022102  104456                        TRAP    C$ERHRD
  (5) 022104  023436                        .WORD   10014
  (5) 022106  000000                        .WORD   0
  (5) 022110  012572                        .WORD   ERR5
2063  022112  000426                        BR      60$                     ;EXIT
2064  022114  005737  003044          10$:  TST     T.CS                    ;CHECK FOR ANY ERRORS
2065  022120  100005                        BPL     12$                     ;NO - SKIP
2066  022122                                ERRHRD  10016.,,ERR6            ;REPORT ALL ERRORS
  (4) 022122  104456                        TRAP    C$ERHRD
  (5) 022124  023440                        .WORD   10016
  (5) 022126  000000                        .WORD   0
  (5) 022130  012642                        .WORD   ERR6
```

```
2067  022132  000416                        BR      60$
2068  022134  012701  003054        12$:    MOV     #HDWRD2,R1      ;GET POINTER
2069  022140  016221  000006                MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
2070  022144  016221  000006                MOV     RLMP(R2),(R1)+
2071  022150  000411                        BR      65$             ;EXIT
2072  022152  004737  016320        14$:    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
2073  022156  012603                        MOV     (SP)+,R3        ;GET RESULTS
2074  022160                                ERRHRD  10015.,,ERR1    ;REPORT
  (4) 022160  104456                        TRAP    C$ERHRD
  (5) 022162  023437                        .WORD   10015
  (5) 022164  000000                        .WORD   0
  (5) 022166  012340                        .WORD   ERR1
2075  022170  005037  003016        60$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
2076  022174  005737  003126        65$:    TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
2077  022200  001007                        BNE     22$             ;NO - SKIP
2078  022202  012703  003034                MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
2079  022206  012701  000004                MOV     #4,R1           ;SET COUNT
2080  022212  012623                20$:    MOV     (SP)+,(R3)+     ;RESTORE REGISTER
2081  022214  005301                        DEC     R1              ;DEC COUNT
2082  022216  001375                        BNE     20$             ;LOOP UNTIL ALL ARE RESTORED
2083  022220  162737  000002  003002  22$:  SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2084  022226  012604                        MOV     (SP)+,R4        ;RESTORE REGS
2085  022230  012601                        MOV     (SP)+,R1
2086  022232  012600                        MOV     (SP)+,R0
2087  022234  012603                        MOV     (SP)+,R3
2088  022236  005737  003016                TST     ERRSWI          ;TEST IF ERROR RETURN
2089  022242  001403                        BEQ     99$             ;YES - SKIP
2090  022244  063716  003016                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2091  022250  000207                        RTS     PC
2092  022252  017616  000000        99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2093  022256  000207                        RTS     PC
2094
2170
2171
2172                                ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
2173  022260  013705  003052        POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
2174  022264  000402                        BR      POSHD0          ;SKIP
2175  022266  013705  003052        POSHSB: MOV     T.MP,R5         ;START FOR POSITION HD BIT IN MP
2176  022272  010146                POSHD0: MOV     R1,-(SP)        ;STORE R1
2177  022274  042705  177677                BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
2178  022300  012701  000006                MOV     #6,R1           ;SET SHIFT COUNT
2179  022304  006205                1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
2180  022306  005301                        DEC     R1
2181  022310  001375                        BNE     1$
2182  022312  012601                        MOV     (SP)+,R1        ;RESTORE R1
2183  022314  000207                        RTS     PC              ;RETURN
2184
2185
2186
2187                                ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
2188                                ;       FROM THE CALLING ROUTINE IN R1.
2189  022316  010346                RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
2190  022320  013703  003002                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2191  022324  005723                        TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2192  022326  016663  000002  002404        MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2193  022334  162763  000004  002404        SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
```

```
2194  022342  010337  003002                MOV     R3,SSINDX       ;STORE IT BACK
2195  022346  010046                       MOV     R0,-(SP)
2196  022350  010146                       MOV     R1,-(SP)
2197  022352  010446                       MOV     R4,-(SP)
2198  022354  012737  000002  003016       MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
2199  022362  004737  016560        5$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
2200  022366  022616                       10$
2201  022370  032737  000001  003044       BIT     #DRDYMSK,T.CS   ;CHECK IF READY
2202  022376  001111                       BNE     9$              ;YES - SKIP
2203  022400  005301                       DEC     R1              ;DEC WAIT COUNT
2204  022402  001415                       BEQ     7$              ;SKIP IF 0
2205  022404                               WAITUS  #1
  (3) 022404  012727  000001               MOV     ###1,(PC)+
  (3) 022410  000000                       .WORD   0
  (3) 022412  013727  002116               MOV     L$DLY,(PC)+
  (3) 022416  000000                       .WORD   0
  (3) 022420  005367  177772               DEC     -6(PC)
  (3) 022424  001375                       BNE     .-4
  (3) 022426  005367  177756               DEC     -22(PC)
  (3) 022432  001367                       BNE     .-20
2206  022434  000752                       BR      5$
2207  022436  012703  010260        7$:    MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
2208  022442                               ERRHRD  10020.,,ERR3    ;REPORT READY ERROR
  (4) 022442  104456                       TRAP    C$ERHRD
  (5) 022444  023444                       .WORD   10020
  (5) 022446  000000                       .WORD   0
  (5) 022450  012454                       .WORD   ERR3
2209  022452  012701  000030               MOV     #24.,R1         ;INITIALIZE WAIT COUNT
2210  022456  004737  016560        6$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
2211  022462  022616                       10$
2212  022464  032737  000001  003044       BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
2213  022472  001037                       BNE     8$              ;YES - SKIP
2214                               .LIST   ME
2215  022474                               WAITMS  #1              ;WAIT 100 MS
  (1) 022474  012737  000001  003142       MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNTER
  (1) 022502  006337  003142               ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
  (1) 022506  006337  003142               ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
  (1) 022512                       64$:    DELAY   #250.           ;IMPLEMENT 25-MS TIME DELAY
  (2) 022512                       M$GNINS <MOV    ##250.,(PC)+>
  (3) 022512  012727  000372               MOV     ##250.,(PC)+
  (3)                              .MEXIT
  (2) 022516                       M$GNINS <.WORD  0>
  (3) 022516  000000                       .WORD   0
  (3)                              .MEXIT
  (2) 022520                       M$GNINS <MOV    L$DLY,(PC)+>
  (3) 022520  013727  002116               MOV     L$DLY,(PC)+
  (3)                              .MEXIT
  (2) 022524                       M$GNINS <.WORD  0>
  (3) 022524  000000                       .WORD   0
  (3)                              .MEXIT
  (2) 022526                       M$GNINS <DEC    -6(PC)>
  (3) 022526  005367  177772               DEC     -6(PC)
  (3)                              .MEXIT
  (2) 022532                       M$GNINS <BNE    .-4>
  (3) 022532  001375                       BNE     .-4
  (3)                              .MEXIT
```

```
   (2)  022534                          M$GNINS <DEC    -22(PC)>
   (3)  022534  005367  177756                  DEC     -22(PC)
   (3)                               .MEXIT
   (2)  022540                          M$GNINS <BNE    .-20>
   (3)  022540  001367                          BNE     .-20
   (3)                               .MEXIT
   (1)  022542  005337  003142                  DEC     DLYCNT          ;DECREMENT DELAY COUNT
   (1)  022546  001361                          BNE     64$             ;BRANCH IF TIME DELAY NOT EXPIRED
  2216                               .NLIST  ME
  2217  022550  005301                          DEC     R1              ;DEC WAIT COUNT
  2218  022552  001341                          BNE     6$              ;LOOP UNTIL TIME DONE
  2219  022554  012704  011310                  MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
  2220  022560                                  ERRHRD  10021.,,ERR5
   (4)  022560  104456                          TRAP    C$ERHRD
   (5)  022562  023445                          .WORD   10021
   (5)  022564  000000                          .WORD   0
   (5)  022566  012572                          .WORD   ERR5
  2221  022570  000410                          BR      11$             ;EXIT
  2222  022572  032737  100000  003044  8$:     BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
  2223  022600  001406                          BEQ     10$             ;NO - SKIP
  2224  022602                                  ERRHRD  10022.,,ERR6    ;REPORT ALL ERRORS
   (4)  022602  104456                          TRAP    C$ERHRD
   (5)  022604  023446                          .WORD   10022
   (5)  022606  000000                          .WORD   0
   (5)  022610  012642                          .WORD   ERR6
  2225  022612  005337  003156          11$:    DEC     ERRCNT          ;DECREMENT FOR DOUBLE ERROR REPORT
  2226  022616  005037  003016          10$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
  2227  022622  162737  000002  003002  9$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
  2228  022630  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
  2229  022632  012601                          MOV     (SP)+,R1
  2230  022634  012600                          MOV     (SP)+,R0
  2231  022636  012603                          MOV     (SP)+,R3        ;RESTORE R3
  2232  022640  005737  003016                  TST     ERRSWI          ;TEST IF ERROR RETURN
  2233  022644  001403                          BEQ     99$             ;YES - SKIP
  2234  022646  063716  003016                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
  2235  022652  000207                          RTS     PC
  2236  022654  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
  2237  022660  000207                          RTS     PC
  2238
  2239
  2240
  2241                              ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
  2242                              ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
  2243                              ;       NUMBER IN CURCYL.
  2244  022662  010346          GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
  2245  022664  013703  003002          MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
  2246  022670  005723                  TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
  2247  022672  016663  000002  002404  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
  2248  022700  162763  000004  002404  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
  2249  022706  010337  003002          MOV     R3,SSINDX       ;STORE IT BACK
  2250  022712  010046                  MOV     R0,-(SP)
  2251  022714  010546                  MOV     R5,-(SP)
  2252  022716  004737  021606          JSR     PC,XRDHD        ;DO READ HEADER
  2253  022722  022752                  65$
  2254  022724  013703  003052          MOV     HDWRD1,R3       ;GET HEADER WORD
  2255  022730  012705  000007          MOV     #7,R5           ;SET SHIFT COUNT
```

```
2256  022734  006203                  4$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
2257  022736  005305                          DEC     R5
2258  022740  001375                          BNE     4$
2259  022742  042703  177000                  BIC     #177000,R3
2260  022746  010337  003104                  MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
2261  022752  162737  000002  003002  65$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
2262  022760  012605                          MOV     (SP)+,R5        ;RESTORE REGISTERS
2263  022762  012600                          MOV     (SP)+,R0
2264  022764  012603                          MOV     (SP)+,R3
2265  022766  005737  003016                  TST     ERRSWI          ;TEST IF ERROR RETURN
2266  022772  001403                          BEQ     99$             ;YES - SKIP
2267  022774  063716  003016                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
2268  023000  000207                          RTS     PC
2269  023002  017616  000000          99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
2270  023006  000207                          RTS     PC
2271
2300
2301
2302                                  ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
2303                                  ;       IN IBUFF.
2304  023010  010346                  RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
2305  023012  013703  003002                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
2306  023016  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
2307  023020  016663  000002  002404          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
2308  023026  162763  000004  002404          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
2309  023034  010337  003002                  MOV     R3,SSINDX       ;STORE IT BACK
2310  023040  010046                          MOV     R0,-(SP)
2311  023042  010146                          MOV     R1,-(SP)
2312  023044  010446                          MOV     R4,-(SP)
2313  023046  012737  000002  003016          MOV     #2,FRRSWI       ;SET FOR NO ERROR RETURN
2314  023054  012701  000050                  MOV     #40.,R1         ;SET HEADER COUNT
2315  023060  052737  100000  003004          BIS     #HDR40,OPFLAG   ;SET 40 HDR OP FLAG
2316  023066  012703  003762                  MOV     #IBUFF,R3       ;SET POINTER TO STORE HDRS
2317  023072  013704  003026                  MOV     RLBAS,R4        ;GET BASE ADDRESS
2318  023076  062704  000006                  ADD     #RLMP,R4        ;MAKE IT POINT TO MP REG
2319  023102  012737  000010  003034          MOV     #10,L.CS        ;LOAD FOR READ HEADER, NO INTERRUPT
2320  023110  053737  003032  003034          BIS     RLDRV,L.CS      ;INSERT DRIVE NUMBER
2321  023116  042737  002000  003034          BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
2322  023124  005037  003036                  CLR     L.BA            ;CLEAR BA
2323  023130  005037  003040                  CLR     L.DA            ;CLEAR DA
2324  023134  005737  003112                  TST     DESHD           ;TEST IF HEAD 0
2325  023140  001403                          BEQ     3$              ;YES - SKIP
2326  023142  052737  000020  003040          BIS     #HDSEL,L.DA     ;ELSE INSERT HEAD 0
2327  023150  013762  003040  000004  3$:     MOV     L.DA,RLDA(R2)   ;LOAD RLDA REG
2328  023156  013762  003036  000002          MOV     L.BA,RLBA(R2)   ;LOAD RLBA
2329  023164  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
2330  023172  001003                          BNE     6$              ;YES - SKIP
2331  023174  004737  021132                  JSR     PC,RDYCHK       ;ELSE CHECK READY
2332  023200  023316                          65$
2333  023202  013762  003034  000000  6$:     MOV     L.CS,RLCS(R2)   ;LOAD RLCS REG
2334  023210  012700  077777                  MOV     #77777,R0       ;SET COUNT FOR WAIT
2335  023214  032762  000200  000000  7$:     BIT     #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
2336  023222  001016                          BNE     8$              ;YES - SKIP
2337  023224  005300                          DEC     R0              ;DEC COUNT
2338  023226  001372                          BNE     7$              ;SKIP IF NOT YET 0
2339  023230  004737  016266                  JSR     PC,READRL       ;ELSE GET ALL REGISTERS
```

```
 2340  023234  004737  016320            JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
 2341  023240  012603                    MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
 2342  023242                            ERRHRD  10025.,,ERR1
  (4)  023242  104456                    TRAP    C$ERHRD
  (5)  023244  023451                    .WORD   10025
  (5)  023246  000000                    .WORD   0
  (5)  023250  012340                    .WORD   ERR1
 2343  023252  005037  003016            CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 2344  023256  000417                    BR      65$
 2345  023260  005737  003044    8$:     TST     T.CS            ;TEST FOR ANY ERRORS
 2346  023264  100007                    BPL     12$             ;NO - SKIP
 2347  023266                            ERRHRD  10026.,,ERR6
  (4)  023266  104456                    TRAP    C$ERHRD
  (5)  023270  023452                    .WORD   10026
  (5)  023272  000000                    .WORD   0
  (5)  023274  012642                    .WORD   ERR6
 2348  023276  005037  003016            CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
 2349  023302  000405                    BR      65$
 2350  023304  011423    12$:    MOV     (R4),(R3)+      ;STORE HEADER WORDS
 2351  023306  011423                    MOV     (R4),(R3)+
 2352  023310  011423                    MOV     (R4),(R3)+
 2353  023312  005301                    DEC     R1              ;DEC HEADER COUNT
 2354  023314  001332                    BNE     6$
 2355  023316  162737  000002  003002  65$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 2356  023324  012604                    MOV     (SP)+,R4        ;RESTORE REGISTERS
 2357  023326  012601                    MOV     (SP)+,R1
 2358  023330  012600                    MOV     (SP)+,R0
 2359  023332  012603                    MOV     (SP)+,R3
 2360  023334  005737  003016            TST     ERRSWI          ;TEST IF ERROR RETURN
 2361  023340  001403                    BEQ     99$             ;YES - SKIP
 2362  023342  063716  003016            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 2363  023346  000207                    RTS     PC
 2364  023350  017616  000000    99$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 2365  023354  000207                    RTS     PC
 2366
 2367
 2595
 2596                            ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
 2597                            ;       OPERATION BEING PERFORMED PORTION OF ALL
 2598                            ;       ERROR MESSAGES.
 2599  023356  010446            RPTOP:  MOV     R4,-(SP)
 2600  023360  005737  003002            TST     SSINDX          ;TEST SUBROUTINE INDEX 0
 2601  023364  001433                    BEQ     1$              ;SKIP IF 0
 2602  023366  012704  000002            MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
 2603  023372                            PRINTB  #FMT9,#SEQMES   ;PRINT ''SUBROUTINE CALL SEQ''
  (8)  023372  012746  010127            MOV     #SEQMES,-(SP)
  (7)  023376  012746  011627            MOV     #FMT9,-(SP)
  (6)  023402  012746  000002            MOV     #2,-(SP)
  (3)  023406  010600                    MOV     SP,R0
  (4)  023410  104414                    TRAP    C$PNTB
  (4)  023412  062706  000006            ADD     #6,SP
 2604  023416                    3$:     PRINTB  #FMT16,SUBSTK(R4)       ;PRINT CALLING LOCATION
  (8)  023416  016446  002404            MOV     SUBSTK(R4),-(SP)
  (7)  023422  012746  012002            MOV     #FMT16,-(SP)
  (6)  023426  012746  000002            MOV     #2,-(SP)
  (3)  023432  010600                    MOV     SP,R0
```

```
  (4) 023434 104414                      TRAP    C$PNTB
  (4) 023436 062706 000006               ADD     #6,SP
2605 023442 062704 000002                ADD     #2,R4           ;BUMP INDEX
2606 023446 020437 003002                CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
2607 023452 003761                       BLE     3$              ;LOOP IF NOT ALL PRINTED YET
2608 023454                        1$:   PRINTB  #FMT4,ERHEAD,#TSTLAB    ;PRINT ERROR HEADER
  (9) 023454 012746 006363               MOV     #TSTLAB,-(SP)
  (8) 023460 013746 003012               MOV     ERHEAD,-(SP)
  (7) 023464 012746 011432               MOV     #FMT4,-(SP)
  (6) 023470 012746 000003               MOV     #3,-(SP)
  (3) 023474 010600                      MOV     SP,R0
  (4) 023476 104414                      TRAP    C$PNTB
  (4) 023500 062706 000010               ADD     #10,SP
2609 023504 042737 030000 003004         BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
2610 023512 013701 003034                MOV     L.CS,R1         ;GET COMMAND EXECUTED
2611 023516 042701 177741                BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
2612 023522 022701 000006                CMP     #6,R1   ;TEST IF SEEK OPERATION
2613 023526 001003                       BNE     2$              ;NO - SKIP
2614 023530 052737 010000 003004         BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
2615 023536 022701 000012          2$:   CMP     #12,R1          ;TEST IF WRITE
2616 023542 001003                       BNE     20$             ;NO - SKIP
2617 023544 052737 020000 003004         BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
2618 023552 022701 000014          20$:  CMP     #14,R1          ;TEST IF READ
2619 023556 001003                       BNE     22$             ;NO - SKIP
2620 023560 052737 020000 003004         BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
2621 023566                        22$:  PRINTB  #FMT1,#MOPER,OPMSGS(R1) ;PRINT OPERATION
  (9) 023566 016146 002224               MOV     OPMSGS(R1),-(SP)
  (8) 023572 012746 005412               MOV     #MOPER,-(SP)
  (7) 023576 012746 011410               MOV     #FMT1,-(SP)
  (6) 023602 012746 000003               MOV     #3,-(SP)
  (3) 023606 010600                      MOV     SP,R0
  (4) 023610 104414                      TRAP    C$PNTB
  (4) 023612 062706 000010               ADD     #10,SP
2622 023616 020127 000004                CMP     R1,#4           ;CHECK IF GET STATUS
2623 023622 001007                       BNE     4$              ;NO - SKIP
2624 023624 032737 000010 003040         BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
2625 023632 001403                       BEQ     4$              ;NO - SKIP
2626 023634 012701 000016                MOV     #16,R1          ;SET TO PRINT WITH RESET
2627 023640 000436                       BR      9$
2628 023642 032737 007777 003004   4$:   BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
2629 023650 001424                       BEQ     8$              ;NO - SKIP
2630 023652 013704 003004                MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
2631 023656 012701 000020                MOV     #20,R1          ;PRESET THE POINTER
2632 023662 032704 000001          5$:   BIT     #BIT00,R4       ;CHECK THE BIT
2633 023666 001003                       BNE     6$              ;IF SET - SKIP
2634 023670 005721                       TST     (R1)+           ;BUMP POINTLR
2635 023672 006204                       ASR     R4
2636 023674 000772                       BR      5$
2637 023676                        6$:   PRINTB  #FMT2,OPMSGS(R1)
  (8) 023676 016146 002224               MOV     OPMSGS(R1),-(SP)
  (7) 023702 012746 011424               MOV     #FMT2,-(SP)
  (6) 023706 012746 000002               MOV     #2,-(SP)
  (3) 023712 010600                      MOV     SP,R0
  (4) 023714 104414                      TRAP    C$PNTB
  (4) 023716 062706 000006               ADD     #6,SP
2638 023722 032737 100000 003004   8$:   BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
```

```
 2639  023730  001415                          BEQ     10$             ;NO - SKIP
 2640  023732  012701  000050                  MOV     #50,R1          ;ELSE PRINT IT
 2641  023736                          9$:     PRINTB  #FMT2,OPMSGS(R1)
  (8)  023736  016146  002224                  MOV     OPMSGS(R1),-(SP)
  (7)  023742  012746  011424                  MOV     #FMT2,-(SP)
  (6)  023746  012746  000002                  MOV     #2,-(SP)
  (3)  023752  010600                          MOV     SP,R0
  (4)  023754  104414                          TRAP    C$PNTB
  (4)  023756  062706  000006                  ADD     #6,SP
 2642  023762  000434                          BR      15$             ;SKIP
 2643  023764  032737  010000  003004  10$:     BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
 2644  023772  001430                          BEQ     15$             ;NO - SKIP
 2645  023774                                  PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
  (15) 023774  013746  003112                  MOV     DESHD,-(SP)
  (14) 024000  012746  010070                  MOV     #HDWD,-(SP)
  (13) 024004  013746  003110                  MOV     DESSGN,-(SP)
  (12) 024010  012746  010063                  MOV     #SGNWD,-(SP)
  (11) 024014  013746  003106                  MOV     DESDIF,-(SP)
  (10) 024020  012746  010055                  MOV     #DIFWD,-(SP)
   (9) 024024  013746  003100                  MOV     OLDCYL,-(SP)
   (8) 024030  012746  010106                  MOV     #FRMWD,-(SP)
   (7) 024034  012746  011650                  MOV     #FMT13,-(SP)
   (6) 024040  012746  000011                  MOV     #11,-(SP)
   (3) 024044  010600                          MOV     SP,R0
   (4) 024046  104414                          TRAP    C$PNTB
   (4) 024050  062706  000024                  ADD     #24,SP
 2646  024054  032737  020000  003004  15$:     BIT     #RORWOP,OPFLAG  ;TEST IF READ OR WRITE SET
 2647  024062  001424                          BEQ     17$             ;NO - SKIP
 2648  024064                                  PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
  (13) 024064  013746  003114                  MOV     DESSEC,-(SP)
  (12) 024070  012746  010074                  MOV     #SECWD,-(SP)
  (11) 024074  013746  003112                  MOV     DESHD,-(SP)
  (10) 024100  012746  010070                  MOV     #HDWD,-(SP)
   (9) 024104  013746  003104                  MOV     CURCYL,-(SP)
   (8) 024110  012746  010101                  MOV     #CYLWD,-(SP)
   (7) 024114  012746  012177                  MOV     #FMT22,-(SP)
   (6) 024120  012746  000007                  MOV     #7,-(SP)
   (3) 024124  010600                          MOV     SP,R0
   (4) 024126  104414                          TRAP    C$PNTB
   (4) 024130  062706  000020                  ADD     #20,SP
 2649  024134  004737  024606          17$:     JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
 2650  024140  012604                          MOV     (SP)+,R4        ;RESTORE R4
 2651  024142  000207                          RTS     PC
 2652
 2653
 2654
 2655                                  ;       REPORT REASON ROUTINE
 2656                                  ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
 2657  024144  010146                  RPTRES: MOV     R1,-(SP)        ;STORE R1
 2658  024146  010346                          MOV     R3,-(SP)        ;STORE R3
 2659  024150  010446                          MOV     R4,-(SP)        ;STORE R4
 2660  024152  012701  003062                  MOV     #RESPARM,R1     ;GET START OF PARAM
 2661  024156  012103                          MOV     (R1)+,R3        ;GET NUMBER OF PARAM
 2662  024160                                  PRINTB  #FMT1.1,#MRSLT,(R1)     ;PRINT NAME
  (9)  024160  011146                          MOV     (R1),-(SP)
  (8)  024162  012746  005421                  MOV     #MRSLT,-(SP)
```

```
   (7)  024166  012746  011415            MOV     #FMT1.1,-(SP)
   (6)  024172  012746  000003            MOV     #3,-(SP)
   (3)  024176  010600                    MOV     SP,R0
   (4)  024200  104414                    TRAP    C$PNTB
   (4)  024202  062706  000010            ADD     #10,SP
2663    024206  021127  010733            CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
2664    024212  001453                    BEQ     6$              ;YES - SKIP REST OF REPORT
2665    024214  012704  011634            MOV     #FMT11,R4       ;PRESET FOR FORMAT 11
2666    024220  022127  010726            CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
2667    024224  001002                    BNE     3$              ;NO - SKIP
2668    024226  012704  011642            MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
2669    024232  005303            3$:     DEC     R3              ;DEC PARAM COUNT
2670    024234  001442                    BEQ     6$              ;IF 0 - EXIT
2671    024236                            PRINTB  R4,#RESE3,(R1)+ ;REPORT IS VALUE
   (9)  024236  012146                    MOV     (R1)+,-(SP)
   (8)  024240  012746  011154            MOV     #RESE3,-(SP)
   (7)  024244  010446                    MOV     R4,-(SP)
   (6)  024246  012746  000003            MOV     #3,-(SP)
   (3)  024252  010600                    MOV     SP,R0
   (4)  024254  104414                    TRAP    C$PNTB
   (4)  024256  062706  000010            ADD     #10,SP
2672    024262                            PRINTB  R4,#RESE4,(R1)+ ;REPORT SB VALUE
   (9)  024262  012146                    MOV     (R1)+,-(SP)
   (8)  024264  012746  011160            MOV     #RESE4,-(SP)
   (7)  024270  010446                    MOV     R4,-(SP)
   (6)  024272  012746  000003            MOV     #3,-(SP)
   (3)  024276  010600                    MOV     SP,R0
   (4)  024300  104414                    TRAP    C$PNTB
   (4)  024302  062706  000010            ADD     #10,SP
2673    024306  162703  000002            SUB     #2,R3           ;DEC PARAM COUNT
2674    024312  001413                    BEQ     6$              ;IF 0 - EXIT
2675    024314                            PRINTB  #FMT1,#RESE5,(R1)+  ;REPORT CONDITION
   (9)  024314  012146                    MOV     (R1)+,-(SP)
   (8)  024316  012746  011165            MOV     #RESE5,-(SP)
   (7)  024322  012746  011410            MOV     #FMT1,-(SP)
   (6)  024326  012746  000003            MOV     #3,-(SP)
   (3)  024332  010600                    MOV     SP,R0
   (4)  024334  104414                    TRAP    C$PNTB
   (4)  024336  062706  000010            ADD     #10,SP
2676    024342  012604            6$:     MOV     (SP)+,R4        ;RESTORE REGS
2677    024344  012603                    MOV     (SP)+,R3
2678    024346  012601                    MOV     (SP)+,R1
2679    024350  000207                    RTS     PC              ;RETURN
2680
2681
2682
2683                              ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
2684                              ;       AND ALL REGISTER CONTENTS.
2685    024352                    RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (11)  024352  005046                    CLR     -(SP)
  (11)  024354  153716  003033            BISB    RLDRV+1,(SP)
  (10)  024360  012746  006051            MOV     #DRVNAM,-(SP)
   (9)  024364  013746  003026            MOV     RLBAS,-(SP)
   (8)  024370  012746  006040            MOV     #BASADD,-(SP)
   (7)  024374  012746  011443            MOV     #FMT5,-(SP)
   (6)  024400  012746  000005            MOV     #5,-(SP)
```

```
   (3)  024404  010600                        MOV     SP,R0
   (4)  024406  104414                        TRAP    C$PNTB
   (4)  024410  062706  000014                ADD     #14,SP
 2686                                 ;       REPORT RL11 REGISTERS
 2687  024414                                 PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
  (13) 024414  012746  010070                MOV     #HDWD,-(SP)
  (12) 024420  012746  010101                MOV     #CYLWD,-(SP)
  (11) 024424  012746  006135                MOV     #MPNAM,-(SP)
  (10) 024430  012746  006123                MOV     #BANAM,-(SP)
   (9) 024434  012746  006130                MOV     #DANAM,-(SP)
   (8) 024440  012746  006116                MOV     #CSNAM,-(SP)
   (7) 024444  012746  011463                MOV     #FMT6,-(SP)
   (6) 024450  012746  000007                MOV     #7,-(SP)
   (3) 024454  010600                        MOV     SP,R0
   (4) 024456  104414                        TRAP    C$PNTB
   (4) 024460  062706  000020                ADD     #20,SP
 2688  024464                                 PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
  (12) 024464  013746  003042                MOV     L.MP,-(SP)
  (11) 024470  013746  003036                MOV     L.BA,-(SP)
  (10) 024474  013746  003040                MOV     L.DA,-(SP)
   (9) 024500  013746  003034                MOV     L.CS,-(SP)
   (8) 024504  012746  006142                MOV     #LAB1,-(SP)
   (7) 024510  012746  011575                MOV     #FMT8,-(SP)
   (6) 024514  012746  000006                MOV     #6,-(SP)
   (3) 024520  010600                        MOV     SP,R0
   (4) 024522  104414                        TRAP    C$PNTB
   (4) 024524  062706  000016                ADD     #16,SP
 2689  024530                                 PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
  (14) 024530  013746  003112                MOV     DESHD,-(SP)
  (13) 024534  013746  003104                MOV     CURCYL,-(SP)
  (12) 024540  013746  003052                MOV     T.MP,-(SP)
  (11) 024544  013746  003046                MOV     T.BA,-(SP)
  (10) 024550  013746  003050                MOV     T.DA,-(SP)
   (9) 024554  013746  003044                MOV     T.CS,-(SP)
   (8) 024560  012746  006155                MOV     #LAB2,-(SP)
   (7) 024564  012746  011525                MOV     #FMT7,-(SP)
   (6) 024570  012746  000010                MOV     #10,-(SP)
   (3) 024574  010600                        MOV     SP,R0
   (4) 024576  104414                        TRAP    C$PNTB
   (4) 024600  062706  000022                ADD     #22,SP
 2690  024604  000207                        RTS     PC
 2691
 2692
 2693
 2694                                 ;       CLEAR PARAMETER BLOCK FOR REPORTING
 2695  024606  010546                 CLRPARM: MOV    R5,-(SP)        ;STORE R5
 2696  024610  012701  003062                 MOV    #RESPARM,R1     ;GET ADDRESS OF BLOCK
 2697  024614  012705  000005                 MOV    #5,R5           ;SET COUNT
 2698  024620  005021                 2$:     CLR    (R1)+           ;CLEAR WORD
 2699  024622  005305                         DEC    R5              ;DEC COUNT
 2700  024624  001375                         BNE    2$              ;LOOP UNTIL 0
 2701  024626  012701  003062                 MOV    #RESPARM,R1     ;RESET POINTER
 2702  024632  012605                         MOV    (SP)+,R5        ;RESTORE R5
 2703  024634  000207                         RTS    PC
 2704
 2705  024636                         ENDMOD
```

```
2706
2707
2708
2709                            .TITLE  CZRLICO RL01/02 DRIVE TEST 1
2710
2711
2712
2713
2714                            ;DISK STATE FUNCTIONS
2715                            ;
2716                            ;BITS 0-2 OF THE MULTIPURPOSE REGISTER DURING GET STATUS COMMAND DEFINE THE
2717                            ;STATE OF THE DRIVE
2718                            ;
2719                            ;       STATE   0               LOAD STATE
2720                            ;       STATE   1               SPIN UP
2721                            ;       STATE   2               BRUSH CYCLE
2722                            ;       STATE   3               LOAD HEADS
2723                            ;       STATE   4               SEEK
2724                            ;       STATE   5               LOCK ON
2725                            ;       STATE   6               UNLOAD HEADS
2726                            ;       STATE   7               SPIN DOWN
2727
2728
2729
```

```
                        BGNMOD  HRDWTST


                        .SBTTL  *TEST 1         BASIC INTERFACE (PART 1)

                        BGNTST                  ;TEST01
                                                                        T1::
                                TST     PASNUM          ;CHECK IF FIRST PASS
                                BNE     65$             ;EXIT IF NO
                                TST     MISWIW          ;CHECK IF MANUAL INTERVENTION
                                BPL     65$             ;NO - EXIT TEST
                                MOV     #MISTST,ERHEAD  ;LOAD ERR HEADER
                        2$:     PRINTF  #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
                                CLR     -(SP)
                                BISB    RLDRV+1,(SP)
                                MOV     #DRVNAM,-(SP)
                                MOV     RLBAS,-(SP)
                                MOV     #BASADD,-(SP)
                                MOV     #OPR1A,-(SP)
                                MOV     #OPR1,-(SP)
                                MOV     #FMTOP1,-(SP)
                                MOV     #7,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #20,SP
                                CLR     OBUFF           ;CLEAR FOR RESPONSE
                                GMANIL  OPR002,OBUFF,1,NO
                                TRAP    C$GMAN
                                BR      10000$
                                .WORD   OBUFF
                                .WORD   T$CODE
                                .WORD   OPR002
                                .WORD   1
                        10000$:
                                TST     OBUFF           ;TEST RESPONSE YES
                                BEQ     2$              ;YES - SKIP
                        1$:     JSR     PC,TSTINT       ;INITIALIZE TEST
                                JSR     PC,GSTATC       ;GO GET STATUS (NO RESET)
                                65$
                                BIT     #COSTAT,T.MP    ;CHECK IF COVER OPEN SET
                                BNE     7$              ;YES - SKIP
                                MOV     #MCOSTA,R3      ;SET NAME POINTER
                                ERRHRD  101,,ERR3
                                TRAP    C$ERHRD
                                .WORD   101
                                .WORD   0
                                .WORD   ERR3
                        7$:     BIT     #BHSTAT,T.MP    ;TEST IF BRUSHES HOME
                                BNE     9$              ;YES - SKIP
                                MOV     #MBHSTA,R3      ;SET POINTER FOR BRUSH HOME ERROR
                                ERRHRD  102,,ERR3
                                TRAP    C$ERHRD
                                .WORD   102
                                .WORD   0
                                .WORD   ERR3
```

```
 2759  025042  032737  020000  003052  9$:     BIT     #WLSTAT,T.MP    ;TEST IF WRITE LOCK SET
 2760  025050  001006                          BNE     11$             ;YES - SKIP
 2761  025052  012703  010464                  MOV     #MWLSTA,R3      ;SET NAME POINTER
 2762  025056                                  ERRHRD  103.,,ERR3
  (4)  025056  104456                          TRAP    C$ERHRD
  (5)  025060  000147                          .WORD   103
  (5)  025062  000000                          .WORD   0
  (5)  025064  012454                          .WORD   ERR3
 2763  025066  005737  003060          11$:    TST     T.STAT          ;TEST IF STATE ZERO
 2764  025072  001405                          BEQ     15$             ;YES - SKIP
 2765  025074  005003                          CLR     R3              ;SET STATE EXPECTED
 2766  025076                                  ERRHRD  104.,,ERR7
  (4)  025076  104456                          TRAP    C$ERHRD
  (5)  025100  000150                          .WORD   104
  (5)  025102  000000                          .WORD   0
  (5)  025104  013542                          .WORD   ERR7
 2767  025106  004737  016530          15$:    JSR     PC,GSTATR       ;DO DRIVE RESET
 2768  025112  025114                          65$
 2769  025114                          65$:
 2770  025114                          ENDTST
  (3)  025114                          L10024:
  (3)  025114  104401                          TRAP    C$ETST
 2771
 2772
 2773
 2774                                  .SBTTL  *TEST 2         BASIC INTERFACE (PART 2)
 2775
 2776  025116                          BGNTST                  ;TEST 2
  (3)  025116                                                                  T2::
 2777  025116  005737  003356                  TST     PASNUM          ;TEST IF PASS 0
 2778  025122  001077                          BNE     65$             ;NO - SKIP
 2779  025124  005737  014202                  TST     MISWIW          ;TEST IF MANUAL INTERVENTION
 2780  025130  100074                          BPL     65$             ;NO - SKIP
 2781  025132  012737  006371  003012          MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
 2782
 2783  025140                          2$:     PRINTF  #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>  ;REQUEST CLOSE
  (13) 025140  005046                          CLR     -(SP)
  (13) 025142  153716  003033                  BISB    RLDRV+1,(SP)
  (12) 025146  012746  006051                  MOV     #DRVNAM,-(SP)
  (11) 025152  013746  003026                  MOV     RLBAS,-(SP)
  (10) 025156  012746  006040                  MOV     #BASADD,-(SP)
   (9) 025162  012746  010011                  MOV     #OPR1A,-(SP)
   (8) 025166  012746  007445                  MOV     #OPR2,-(SP)
   (7) 025172  012746  011316                  MOV     #FMTOP1,-(SP)
   (6) 025176  012746  000007                  MOV     #7,-(SP)
   (3) 025202  010600                          MOV     SP,R0
   (4) 025204  104417                          TRAP    C$PNTF
   (4) 025206  062706  000020                  ADD     #20,SP
 2784                                                                 ;COVER AND RESET WRITE LOCK
 2785  025212  005037  004362                  CLR     OBUFF           ;CLEAR FOR RESPONSE
 2786  025216                                  GMANIL  OPR002,OBUFF,1,NO
  (3)  025216  104443                          TRAP    C$GMAN
  (3)  025220  000404                          BR      10000$
  (4)  025222  004362                          .WORD   OBUFF
  (5)  025224  000120                          .WORD   T$CODE
  (5)  025226  007317                          .WORD   OPR002
```

```
  (5)  025230  000001                          .WORD   1
  (3)  025232                          10000$:
2787   025232  005737  004362                 TST     OBUFF           ;TEST IF RESPONSE YES
2788   025236  001740                         BEQ     2$              ;NO - SKIP
2789
2790   025240  004737  016512          1$:    JSR     PC,TSTINT       ;INITIALIZE TEST
2791   025244  004737  016530                 JSR     PC,GSTATR       ;GET STATUS WITH RESET
2792   025250  025322                         65$
2793   025252  032737  000040  003052         BIT     #COSTAT,T.MP    ;TEST IF COVER OPEN RESET
2794   025260  001406                         BEQ     9$              ;YES - SKIP
2795   025262  012703  010436                 MOV     #MCOSTA,R3      ;SET NAME MESSAGE POINTER
2796   025266                                 ERRHRD  201.,,ERR2
  (4)  025266  104456                         TRAP    C$ERHRD
  (5)  025270  000311                         .WORD   201
  (5)  025272  000000                         .WORD   0
  (5)  025274  012406                         .WORD   ERR2
2797
2798   025276  032737  020000  003052  9$:    BIT     #WLSTAT,T.MP    ;TEST IF WRITE LOCK RESET
2799   025304  001406                         BEQ     65$             ;YES - SKIP
2800   025306  012703  010464                 MOV     #MWLSTA,R3      ;SET NAME MESSAGE POINTER
2801   025312                                 ERRHRD  202.,,ERR2
  (4)  025312  104456                         TRAP    C$ERHRD
  (5)  025314  000312                         .WORD   202
  (5)  025316  000000                         .WORD   0
  (5)  025320  012406                         .WORD   ERR2
2802   025322                          65$:
2803   025322                          ENDTST
  (3)  025322                          L10025:
  (3)  025322  104401                         TRAP    C$ETST
2804
2805
2806
2807                                   .SBTTL  *TEST 3         HEAD LOADING
2808   025324                          BGNTST                  ;TEST03
  (3)  025324                                                                   T3::
2809   025324  005737  003356                 TST     PASNUM          ;TEST IF PASS 0
2810   025330  001003                         BNE     4$              ;NO - SKIP
2811   025332  005737  014202                 TST     MISWIW          ;TEST IF MANUAL INTERVENTION
2812   025336  100402                         BMI     5$              ;YES - SKIP
2813   025340                          4$:    EXIT    TST
  (3)  025340  104432                         TRAP    C$EXIT
  (3)  025342  002250                         .WORD   L10026-.
2814                                   ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
2815   025344  005737  003144          5$:    TST     CLKFLG          ;P-CLOCK?
2816   025350  001026                         BNE     7$              ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
2817   025352  010246                         MOV     R2,-(SP)        ;SAVE R2
2818   025354  012702  006662                 MOV     #NOTST,R2       ;INITIALIZE POINTER TO TEST MSG.
2819   025360  112762  000060  000004         MOVB    #'0,4(R2)       ;INSERT TEST NUMBER INTO MSG.
2820   025366  112762  000063  000005         MOVB    #'3,5(R2)       ;INSERT TEST NUMBER INTO MSG.
2821   025374                                 PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 3 CANNOT BE PERFORMED...
  (8)  025374  012746  006662                 MOV     #NOTST,-(SP)
  (7)  025400  012746  011627                 MOV     #FMT9,-(SP)
  (6)  025404  012746  000002                 MOV     #2,-(SP)
  (3)  025410  010600                         MOV     SP,R0
  (4)  025412  104417                         TRAP    C$PNTF
  (4)  025414  062706  000006                 ADD     #6,SP
```

```
2822
2823  025420  012602                          MOV     (SP)+,R2        ;/NO P-CLK''
2824  025422  000137  027470                  JMP     100$            ;RESTORE R2
2825  025426  004737  016512          7$:     JSR     PC,TSTINT       ;BRANCH TO MAKE DRIVE READY FOR SUBSEQUENT TESTS
2826  025432  004737  016530                  JSR     PC,GSTATR       ;INITIALIZE TEST
2827  025436  027612                          T365$                   ;GET STATUS
2828  025440  005737  003060                  TST     T.STAT          ;TEST IF STATE 0
2829  025444  001440                          BEQ     2$              ;YES - SKIP
2830  025446                          1$:     PRINTF  #FMTOP1,#OPR5,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 025446  005046                          CLR     -(SP)
 (13) 025450  153716  003033                  BISB    RLDRV+1,(SP)
 (12) 025454  012746  006051                  MOV     #DRVNAM,-(SP)
 (11) 025460  013746  003026                  MOV     RLBAS,-(SP)
 (10) 025464  012746  006040                  MOV     #BASADD,-(SP)
  (9) 025470  012746  010011                  MOV     #OPR1A,-(SP)
  (8) 025474  012746  007513                  MOV     #OPR5,-(SP)
  (7) 025500  012746  011316                  MOV     #FMTOP1,-(SP)
  (6) 025504  012746  000007                  MOV     #7,-(SP)
  (3) 025510  010600                          MOV     SP,R0
  (4) 025512  104417                          TRAP    C$PNTF
  (4) 025514  062706  000020                  ADD     #20,SP
2831                                                                    ;PROMPT OPERATOR TO 'PRESS LOAD & WAIT FOR
2832                                                                    ;/LOAD LIGHT''
2833  025520  005037  004362                  CLR     OBUFF           ;CLEAR FOR RESPONSE
2834  025524                                  GMANIL  OPR002,OBUFF,1,NO
 (3)  025524  104443                          TRAP    C$GMAN
 (3)  025526  000404                          BR      10000$
 (4)  025530  004362                          .WORD   OBUFF
 (5)  025532  000120                          .WORD   T$CODE
 (5)  025534  007317                          .WORD   OPR002
 (5)  025536  000001                          .WORD   1
 (3)  025540                      10000$:
2835  025540  005737  004362                  TST     OBUFF           ;TEST IF RESPONSE YES
2836  025544  001740                          BEQ     1$              ;NO - SKIP
2837  025546                          2$:     PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 025546  005046                          CLR     -(SP)
 (13) 025550  153716  003033                  BISB    RLDRV+1,(SP)
 (12) 025554  012746  006051                  MOV     #DRVNAM,-(SP)
 (11) 025560  013746  003026                  MOV     RLBAS,-(SP)
 (10) 025564  012746  006040                  MOV     #BASADD,-(SP)
  (9) 025570  012746  010011                  MOV     #OPR1A,-(SP)
  (8) 025574  012746  007477                  MOV     #OPR3,-(SP)
  (7) 025600  012746  011316                  MOV     #FMTOP1,-(SP)
  (6) 025604  012746  000007                  MOV     #7,-(SP)
  (3) 025610  010600                          MOV     SP,R0
  (4) 025612  104417                          TRAP    C$PNTF
  (4) 025614  062706  000020                  ADD     #20,SP
2838                                                                    ;PROMPT OPERATOR TO 'PRESS LOAD''
2839  025620  012737  000004  003004          MOV     #CYLUP,OPFLAG   ;SET CYCLE UP FLAG
2840  025626  012703  000001                  MOV     #1,R3           ;SET EXPECTED STATE VALUE
2841  025632  012737  006414  003012          MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
2842  025640  012701  000454                  MOV     #300.,R1        ;SET WAIT COUNT FOR 30 SECONDS
2843  025644  004737  016544          3$:     JSR     PC,GSTATC       ;GET STATUS
2844  025650  027612                          T365$
2845  025652  005737  003060                  TST     T.STAT          ;TEST IF STATE IS STILL 0
2846  025656  001051                          BNE     10$             ;NO - SKIP
```

```
 2847  025660  005301                    DEC     R1              ;DEC WAIT COUNT
 2848  025662  001432                    BEQ     6$              ;EXIT IF WAIT DONE
 2849                              .LIST   ME
 2850  025664                            TIMDLY  #1000.
  (1)  025664                            SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  025664                    M$PUT   #3,#104,#CLKINT,#340
  (3)  025664                    M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  025664                    M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  025664                    M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  025664                    M$PUT1  #340
  (7)  025664                    M$GNINS <MOV    #340,-(SP)>
  (8)  025664  012746  000340              MOV     #340,-(SP)
  (8)                            .MEXIT
  (5)  025670                    M$PUT1  #CLKINT
  (6)  025670                    M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  025670  012746  016112              MOV     #CLKINT,-(SP)
  (7)                            .MEXIT
  (4)  025674                    M$PUT1  #104
  (5)  025674                    M$GNINS <MOV    #104,-(SP)>
  (6)  025674  012746  000104              MOV     #104,-(SP)
  (6)                            .MEXIT
  (3)  025700                    M$PUT1  #3
  (4)  025700                    M$GNINS <MOV    #3,-(SP)>
  (5)  025700  012746  000003              MOV     #3,-(SP)
  (5)                            .MEXIT
  (2)  025704                    M$SVC   C$SVEC
  (3)  025704                    M$TSTLAB
  (4)                             .MEXIT
  (3)  025704                    M$GNINS <TRAP   C$SVEC>
  (4)  025704  104437                      TRAP    C$SVEC
  (4)                            .MEXIT
  (2)  025706                    M$GNINS <ADD    #10,SP>
  (3)  025706  062706  000010              ADD     #10,SP
  (3)                            .MEXIT
  (1)  025712  012737  001750  003142      MOV     ##1000.,DLYCNT  ;INITIALIZE DELAY COUNT
  (1)  025720  012737  000001  172542      MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
  (1)                                                             ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
  (1)  025726  012737  000113  172540      MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
  (1)                                                             ;/10 KHZ RATE,START THE CLOCK
  (1)  025734  005737  003142      64$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
  (1)  025740  001375                      BNE     64$             ;BRANCH IF TIME NOT ELAPSED
  (1)  025742  005037  172540              CLR     @#172540        ;STOP THE CLOCK
 2851                              .NLIST  ME
 2852  025746  000736                    BR      3$
 2853  025750  005037  004362      6$:     CLR     OBUFF           ;CLEAR FOR RESPONSE
 2854  025754                            GMANIL  OPR003,OBUFF,',NO
  (3)  025754  104443                      TRAP    C$GMAN
  (3)  025756  000404                      BR      10001$
  (4)  025760  004362                      .WORD   OBUFF
  (5)  025762  000120                      .WORD   T$CODE
  (5)  025764  007344                      .WORD   OPR003
  (5)  025766  000001                      .WORD   1
  (3)  025770                      10001$:
 2855  025770  005737  004362              TST     OBUFF           ;TEST IF RESPONSE YES
 2856  025774  001005                      BNE     11$             ;YES - REPORT
 2857  025776  000137  025446              JMP     1$
```

```
 2858  026002  020337  003060          10$:    CMP     R3,T.STAT       ;CHECK IF NOW STATE 1
 2859  026006  001406                          BEQ     13$             ;YES - SKIP
 2860  026010                          11$:    ERRHRD  301.,ERR7
  (4)  026010  104456                          TRAP    C$ERHRD
  (5)  026012  000455                          .WORD   301
  (5)  026014  000000                          .WORD   0
  (5)  026016  013542                          .WORD   ERR7
 2861  026020                                  EXIT    TST
  (3)  026020  104432                          TRAP    C$EXIT
  (3)  026022  001570                          .WORD   L10026-.
 2862  026024  012701  000454          13$:    MOV     #300.,R1        ;INITIALIZE WAIT COUNT FOR 30 SECONDS
 2863  026030  012703  000002                  MOV     #2,R3           ;SET EXPECTED STATE VALUE
 2864  026034  004737  016544          14$:    JSR     PC,GSTATC       ;GET STATUS
 2865  026040  027612                          T365$
 2866  026042  020337  003060                  CMP     R3,T.STAT       ;CHECK IF STATE 2
 2867  026046  001466                          BEQ     20$             ;YES - SKIP
 2868  026050  101006                          BHI     17$             ;CHECK IF NO CHANGE - YES - SKIP
 2869  026052                                  ERRHRD  302.,ERR7
  (4)  026052  104456                          TRAP    C$ERHRD
  (5)  026054  000456                          .WORD   302
  (5)  026056  000000                          .WORD   0
  (5)  026060  013542                          .WORD   ERR7
 2870  026062                                  EXIT    TST
  (3)  026062  104432                          TRAP    C$EXIT
  (3)  026064  001526                          .WORD   L10026-.
 2871  026066  005301          17$:    DEC     R1              ;DEC WAIT COUNT
 2872  026070  001432                          BEQ     18$             ;SKIP IF 0
 2873                          .LIST   ME
 2874  026072                                  TIMDLY  #1000.
  (1)  026072                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  026072                  M$PUT   #3,#104,#CLKINT,#340
  (3)  026072                  M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  026072                  M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  026072                  M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  026072                  M$PUT1  #340
  (7)  026072                  M$GNINS <MOV    #340,-(SP)>
  (8)  026072  012746  000340                 MOV     #340,-(SP)
  (8)                          .MEXIT
  (5)  026076                  M$PUT1  #CLKINT
  (6)  026076                  M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  026076  012746  016112                 MOV     #CLKINT,-(SP)
  (7)                          .MEXIT
  (4)  026102                  M$PUT1  #104
  (5)  026102                  M$GNINS <MOV    #104,-(SP)>
  (6)  026102  012746  000104                 MOV     #104,-(SP)
  (6)                          .MEXIT
  (3)  026106                  M$PUT1  #3
  (4)  026106                  M$GNINS <MOV    #3,-(SP)>
  (5)  026106  012746  000003                 MOV     #3,-(SP)
  (5)                          .MEXIT
  (2)  026112                  M$SVC   C$SVEC
  (3)  026112                  M$TSTLAB
  (4)                           .MEXIT
  (3)  026112                  M$GNINS <TRAP   C$SVEC>
  (4)  026112  104437                         TRAP    C$SVEC
  (4)                          .MEXIT
```

```
   (2)  026114                                  M$GNINS <ADD     #10,SP>
   (3)  026114  062706  000010                          ADD     #10,SP
   (3)                                          .MEXIT
   (1)  026120  012737  001750  003142                  MOV     ##1000.,DLYCNT  ;INITIALIZE DELAY COUNT
   (1)  026126  012737  000001  172542                  MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                          ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  026134  012737  000113  172540                  MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                          ;/10 KHZ RATE,START THE CLOCK
   (1)  026142  005737  003142          65$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  026146  001375                          BNE     65$             ;BRANCH IF TIME NOT ELAPSED
   (1)  026150  005037  172540                  CLR     @#172540        ;STOP THE CLOCK
 2875                                           .NLIST  ME
 2876  026154  000727                          BR      14$
 2877  026156                                 18$:    ERRHRD  303.,,ERR7
  (4)  026156  104456                          TRAP    C$ERHRD
  (5)  026160  000457                          .WORD   303
  (5)  026162  000000                          .WORD   0
  (5)  026164  013542                          .WORD   ERR7
 2878  026166  032737  004000  003052          BIT     #SPDSTAT,T.MP   ;TEST IF SPINDLE TIMEOUT
 2879  026174  001011                          BNE     19$             ;YES - SKIP
 2880  026176  012737  006426  003012          MOV     #SPDERR,ERHEAD  ;SET ERROR HEADER
 2881  026204  012703  010536                  MOV     #MSPERR,R3      ;SET NAME MESSAGE POINTER
 2882  026210                                  ERRHRD  304.,,ERR3
  (4)  026210  104456                          TRAP    C$ERHRD
  (5)  026212  000460                          .WORD   304
  (5)  026214  000000                          .WORD   0
  (5)  026216  012454                          .WORD   ERR3
 2883  026220                                 19$:    EXIT    TST
  (3)  026220  104432                          TRAP    C$EXIT
  (3)  026222  001370                          .WORD   L10026-.
 2884  026224  012737  006371  003012  20$:    MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
 2885  026232  012704  011177                  MOV     #STATE2,R4      ;SET CONDITION MESSAGE POINTER
 2886  026236  012703  010451                  MOV     #MBHSTA,R3      ;SET NAME MESSAGE POINTER
 2887  026242  032737  000010  003052          BIT     #BHSTAT,T.MP    ;TEST IF BRUSH HOME STILL SET
 2888  026250  001006                          BNE     22$             ;YES - SKIP
 2889  026252                                  ERRHRD  305.,,ERR5
  (4)  026252  104456                          TRAP    C$ERHRD
  (5)  026254  000461                          .WORD   305
  (5)  026256  000000                          .WORD   0
  (5)  026260  012572                          .WORD   ERR5
 2890  026262                                 EXIT    TST
  (3)  026262  104432                          TRAP    C$EXIT
  (3)  026264  001326                          .WORD   L10026-.
 2891  026266  012701  000062          22$:    MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
 2892  026272  004737  016544          23$:    JSR     PC,GSTATC       ;GET STATUS
 2893  026276  027612                          T365$
 2894  026300  032737  000010  003052          BIT     #BHSTAT,T.MP    ;TEST IF BRUSH HOME RESET
 2895  026306  001442                          BEQ     27$             ;YES - SKIP
 2896  026310  005301                          DEC     R1              ;DEC WAIT COUNT
 2897  026312  001432                          BEQ     26$             ;SKIP IF ZERO
 2898                                         .LIST   ME
 2899  026314                                  TIMDLY  #1000.
  (1)  026314                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  026314                          M$PUT   #3,#104,#CLKINT,#340
  (3)  026314                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  026314                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
```

```
   (5)  026314                              M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
   (6)  026314                              M$PUT1  #340
   (7)  026314                              M$GNINS <MOV    #340,-(SP)>
   (8)  026314  012746  000340                      MOV     #340,-(SP)
   (8)                                      .MEXIT
   (5)  026320                              M$PUT1  #CLKINT
   (6)  026320                              M$GNINS <MOV    #CLKINT,-(SP)>
   (7)  026320  012746  016112                      MOV     #CLKINT,-(SP)
   (7)                                      .MEXIT
   (4)  026324                              M$PUT1  #104
   (5)  026324                              M$GNINS <MOV    #104,-(SP)>
   (6)  026324  012746  000104                      MOV     #104,-(SP)
   (6)                                      .MEXIT
   (3)  026330                              M$PUT1  #3
   (4)  026330                              M$GNINS <MOV    #3,-(SP)>
   (5)  026330  012746  000003                      MOV     #3,-(SP)
   (5)                                      .MEXIT
   (2)  026334                              M$SVC   C$SVEC
   (3)  026334                              M$TSTLAB
   (4)                                       .MEXIT
   (3)  026334                              M$GNINS <TRAP   C$SVEC>
   (4)  026334  104437                              TRAP    C$SVEC
   (4)                                      .MEXIT
   (2)  026336                              M$GNINS <ADD    #10,SP>
   (3)  026336  062706  000010                      ADD     #10,SP
   (3)                                      .MEXIT
   (1)  026342  012737  001750  003142              MOV     ##1000.,DLYCNT  ;INITIALIZE DELAY COUNT
   (1)  026350  012737  000001  172542              MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                      ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  026356  012737  000113  172540              MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                      ;/10 KHZ RATE,START THE CLOCK
   (1)  026364  005737  003142          66$:        TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  026370  001375                              BNE     66$             ;BRANCH IF TIME NOT ELAPSED
   (1)  026372  005037  172540                      CLR     @#172540        ;STOP THE CLOCK
2900                                    .NLIST  ME
2901    026376  000735                              BR      23$             ;LOOP
2902    026400                          26$:        ERRHRD  306.,,ERR4
   (4)  026400  104456                              TRAP    C$ERHRD
   (5)  026402  000462                              .WORD   306
   (5)  026404  000000                              .WORD   0
   (5)  026406  012522                              .WORD   ERR4
2903    026410                                      EXIT    TST
   (3)  026410  104432                              TRAP    C$EXIT
   (3)  026412  001200                              .WORD   L10026-.
2904    026414  012701  000454          27$:        MOV     #300.,R1        ;INITIALIZE WAIT COUNT FOR 30 SECONDS
2905    026420  004737  016544          28$:        JSR     PC,GSTATC       ;GET STATUS
2906    026424  027612                              T365$
2907    026426  032737  000010  003052              BIT     #BHSTAT,T.MP    ;TEST IF BRUSH HOME SF  AGAIN
2908    026434  001042                              BNE     32$             ;YES - SKIP
2909    026436  005301                              DEC     R1              ;ELSE DEC WAIT COUNT
2910    026440  001432                              BEQ     30$             ;SKIP IF 0
2911                                    .LIST   ME
2912    026442                                      TIMDLY  #1000.
   (1)  026442                                      SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
   (2)  026442                          M$PUT   #3,#104,#CLKINT,#340
   (3)  026442                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
```

```
 (4)  026442                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
 (5)  026442                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
 (6)  026442                          M$PUT1  #340
 (7)  026442                          M$GNINS <MOV    #340,-(SP)>
 (8)  026442  012746  000340                  MOV     #340,-(SP)
 (8)                                  .MEXIT
 (5)  026446                          M$PUT1  #CLKINT
 (6)  026446                          M$GNINS <MOV    #CLKINT,-(SP)>
 (7)  026446  012746  016112                  MOV     #CLKINT,-(SP)
 (7)                                  .MEXIT
 (4)  026452                          M$PUT1  #104
 (5)  026452                          M$GNINS <MOV    #104,-(SP)>
 (6)  026452  012746  000104                  MOV     #104,-(SP)
 (6)                                  .MEXIT
 (3)  026456                          M$PUT1  #3
 (4)  026456                          M$GNINS <MOV    #3,-(SP)>
 (5)  026456  012746  000003                  MOV     #3,-(SP)
 (5)                                  .MEXIT
 (2)  026462                          M$SVC   C$SVEC
 (3)  026462                          M$TSTLAB
 (4)                                   .MEXIT
 (3)  026462                          M$GNINS <TRAP   C$SVEC>
 (4)  026462  104437                          TRAP    C$SVEC
 (4)                                  .MEXIT
 (2)  026464                          M$GNINS <ADD    #10,SP>
 (3)  026464  062706  000010                  ADD     #10,SP
 (3)                                  .MEXIT
 (1)  026470  012737  001750  003142          MOV     ##1000.,DLYCNT  ;INITIALIZE DELAY COUNT
 (1)  026476  012737  000001  172542          MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
 (1)                                                                  ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
 (1)  026504  012737  000113  172540          MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
 (1)                                                                  ;/10 KHZ RATE,START THE CLOCK
 (1)  026512  005737  003142          67$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
 (1)  026516  001375                          BNE     67$             ;BRANCH IF TIME NOT ELAPSED
 (1)  026520  005037  172540                  CLR     @#172540        ;STOP THE CLOCK
2913                                  .NLIST  ME
2914  026524  000735                        BR      28$
2915  026526                          30$:    ERRHRD  307.,,ERR5
 (4)  026526  104456                        TRAP    C$ERHRD
 (5)  026530  000463                        .WORD   307
 (5)  026532  000000                        .WORD   0
 (5)  026534  012572                        .WORD   ERR5
2916  026536                          EXIT    TST
 (3)  026536  104432                        TRAP    C$EXIT
 (3)  026540  001052                        .WORD   L10026-.
2917  026542  012737  006414  003012  32$:    MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
2918  026550  012703  000003                  MOV     #3,R3           ;SET EXPECTED STATE VALUE
2919  026554  004737  016544                  JSR     PC,GSTATC       ;GET STATUS
2920  026560  027612                          T365$
2921  026562  020337  003060                  CMP     R3,T.STAT       ;CHECK IF STATE 3
2922  026566  001406                          BEQ     36$             ;YES - SKIP
2923  026570                                  ERRHRD  308.,,ERR7
 (4)  026570  104456                        TRAP    C$ERHRD
 (5)  026572  000464                        .WORD   308
 (5)  026574  000000                        .WORD   0
 (5)  026576  013542                        .WORD   ERR7
```

```
 2924  026600                                        EXIT    TST
  (3)  026600  104432                               TRAP    C$EXIT
  (3)  026602  001010                               .WORD   L10026-.
 2925  026604  012737  006371  003012  36$:        MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
 2926  026612  012704  011207                      MOV     #STATE3,R4      ;SET CONDITION MESSAGE POINTER
 2927  026616  012703  010475                      MOV     #MHOSTA,R3      ;SET NAME MESSAGE POINTER
 2928  026622  004737  016544                      JSR     PC,GSTATC       ;GET STATUS
 2929  026626  027612                              T365$
 2930  026630  032737  000020  003052              BIT     #HOSTAT,T.MP    ;TEST IF HEADS OUT SET
 2931  026636  001006                              BNE     38$             ;YES - SKIP
 2932  026640                                      ERRHRD  309.,,ERR5
  (4)  026640  104456                               TRAP    C$ERHRD
  (5)  026642  000465                               .WORD   309
  (5)  026644  000000                               .WORD   0
  (5)  026646  012572                               .WORD   ERR5
 2933  026650                                      EXIT    TST
  (3)  026650  104432                               TRAP    C$EXIT
  (3)  026652  000740                               .WORD   L10026-.
 2934  026654  012701  000012              38$:     MOV     #10.,R1
 2935                                              .LIST   ME
 2936  026660                                      381$:   TIMDLY  #1
  (1)  026660                                              SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  026660                                      M$PUT   #3,#104,#CLKINT,#340
  (3)  026660                                      M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  026660                                      M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (5)  026660                                      M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  026660                                      M$PUT1  #340
  (7)  026660                                      M$GNINS <MOV    #340,-(SP)>
  (8)  026660  012746  000340                              MOV     #340,-(SP)
  (8)                                              .MEXIT
  (5)  026664                                      M$PUT1  #CLKINT
  (6)  026664                                      M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  026664  012746  016112                              MOV     #CLKINT,-(SP)
  (7)                                              .MEXIT
  (4)  026670                                      M$PUT1  #104
  (5)  026670                                      M$GNINS <MOV    #104,-(SP)>
  (6)  026670  012746  000104                              MOV     #104,-(SP)
  (6)                                              .MEXIT
  (3)  026674                                      M$PUT1  #3
  (4)  026674                                      M$GNINS <MOV    #3,-(SP)>
  (5)  026674  012746  000003                              MOV     #3,-(SP)
  (5)                                              .MEXIT
  (2)  026700                                      M$SVC   C$SVEC
  (3)  026700                                      M$TSTLAB
  (4)                                               .MEXIT
  (3)  026700                                      M$GNINS <TRAP   C$SVEC>
  (4)  026700  104437                                      TRAP    C$SVEC
  (4)                                              .MEXIT
  (2)  026702                                      M$GNINS <ADD    #10,SP>
  (3)  026702  062706  000010                              ADD     #10,SP
  (3)                                              .MEXIT
  (1)  026706  012737  000001  003142              MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
  (1)  026714  012737  000001  172542              MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
  (1)                                                                      ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
  (1)  026722  012737  000113  172540              MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
  (1)                                                                      ;/10 KHZ RATE,START THE CLOCK
```

```
  (1)  026730  005737  003142          68$:   TST     DLYCNT          ;DELAY COUNT EXPIRED?
  (1)  026734  001375                         BNE     68$             ;BRANCH IF TIME NOT ELAPSED
  (1)  026736  005037  172540                 CLR     @#172540        ;STOP THE CLOCK
2937                                  .NLIST  ME
2938   026742  012700  000001                 MOV     #1,R0
2939   026746  004737  016544                 JSR     PC,GSTATC       ;GET THE STATUS AFTER SHORT DELAY
2940   026752  027612                         T365$
2941   026754  032737  001000  003052         BIT     #VCSTAT,T.MP    ;TEST IF VOLUME CHECK SET
2942   026762  001012                         BNE     40$
2943   026764  005301                         DEC     R1              ;DECREMENT COUNTER
2944   026766  001334                         BNE     381$            ;TRY FOR 'VC' AGAIN IF MORE TIME LEFT
2945   026770  012703  010425                 MOV     #MVOLCK,R3      ;SET NAME MESSAGE POINTER
2946   026774                                 ERRHRD  310.,,ERR5
  (4)  026774  104456                         TRAP    C$ERHRD
  (5)  026776  000466                         .WORD   310
  (5)  027000  000000                         .WORD   0
  (5)  027002  012572                         .WORD   ERR5
2947   027004                                 EXIT    TST
  (3)  027004  104432                         TRAP    C$EXIT
  (3)  027006  000604                         .WORD   L10026-.
2948   027010  032737  040000  003044  40$:   BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR SET
2949   027016  001010                         BNE     42$             ;YES - SKIP
2950   027020  012703  010402                 MOV     #MDRERR,R3      ;SET NAME MESSAGE POINTER
2951   027024                                 ERRHRD  311.,,ERR5
  (4)  027024  104456                         TRAP    C$ERHRD
  (5)  027026  000467                         .WORD   311
  (5)  027030  000000                         .WORD   0
  (5)  027032  012572                         .WORD   ERR5
2952   027034                                 EXIT    TST
  (3)  027034  104432                         TRAP    C$EXIT
  (3)  027036  000554                         .WORD   L10026-.
2953   027040  012701  005670          42$:   MOV     #3000.,R1               ;SET WAIT COUNT FOR 300 MS
2954   027044  012737  006414  003012         MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
2955   027052  012703  000004                 MOV     #4,R3           ;SET EXPECTED STATE VALUE
2956   027056  004737  016544          43$:   JSR     PC,GSTATC       ;GET STATUS
2957   027062  027612                         T365$
2958   027064  020337  003060                 CMP     R3,T.STAT       ;CHECK IF STATE 4
2959   027070  001442                         BEQ     49$             ;YES - SKIP
2960   027072  005301                         DEC     R1              ;DEC WAIT COUNT
2961   027074  001432                         BEQ     47$             ;SKIP IF 0
2962                                  .LIST   ME
2963   027076                                 TIMDLY  #1
  (1)  027076                                 SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  027076                          M$PUT  #3,#104,#CLKINT,#340
  (3)  027076                          M$PUT  <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  027076                          M$PUT  <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  027076                          M$PUT  <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  027076                          M$PUT1 #340
  (7)  027076                          M$GNINS <MOV    #340,-(SP)>
  (8)  027076  012746  000340                 MOV     #340,-(SP)
  (8)                                  .MEXIT
  (5)  027102                          M$PUT1 #CLKINT
  (6)  027102                          M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  027102  012746  016112                 MOV     #CLKINT,-(SP)
  (7)                                  .MEXIT
  (4)  027106                          M$PUT1 #104
```

```
   (5)  027106                              M$GNINS <MOV    #104,-(SP)>
   (6)  027106  012746  000104                     MOV     #104,-(SP)
   (6)                                      .MEXIT
   (3)  027112                              M$PUT1  #3
   (4)  027112                              M$GNINS <MOV    #3,-(SP)>
   (5)  027112  012746  000003                     MOV     #3,-(SP)
   (5)                                      .MEXIT
   (2)  027116                              M$SVC   C$SVEC
   (3)  027116                              M$TSTLAB
   (4)                                       .MEXIT
   (3)  027116                              M$GNINS <TRAP   C$SVEC>
   (4)  027116  104437                             TRAP    C$SVEC
   (4)                                      .MEXIT
   (2)  027120                              M$GNINS <ADD    #10,SP>
   (3)  027120  062706  000010                     ADD     #10,SP
   (3)                                      .MEXIT
   (1)  027124  012737  000001  003142             MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
   (1)  027132  012737  000001  172542             MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                     ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  027140  012737  000113  172540             MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                     ;/10 KHZ RATE,START THE CLOCK
   (1)  027146  005737  003142          69$:       TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  027152  001375                             BNE     69$             ;BRANCH IF TIME NOT ELAPSED
   (1)  027154  005037  172540                     CLR     @#172540        ;STOP THE CLOCK
  2964                                      .NLIST  ME
  2965  027160  000736                             BR      43$
  2966  027162                              47$:    ERRHRD  312,,ERR7
   (4)  027162  104456                             TRAP    C$ERHRD
   (5)  027164  000470                             .WORD   312
   (5)  027166  000000                             .WORD   0
   (5)  027170  013542                             .WORD   ERR7
  2967  027172                                     EXIT    TST
   (3)  027172  104432                             TRAP    C$EXIT
   (3)  027174  000416                             .WORD   L10026-.
  2968  027176  012701  000454          49$:       MOV     #300.,R1        ;SET WAIT COUNT FOR 30 MS
  2969  027202  012703  000005                     MOV     #5,R3           ;SET EXPECTED STATE VALUE
  2970  027206  004737  016544          50$:       JSR     PC,GSTATC       ;GET STATUS
  2971  027212  027612                             T365$
  2972  027214  020337  003060                     CMP     R3,T.STAT       ;CHECK IF STATE 5
  2973  027220  001442                             BEQ     55$             ;YES - SKIP
  2974  027222  005301                             DEC     R1              ;DEC WAIT COUNT
  2975  027224  001432                             BEQ     51$             ;ELSE SKIP
  2976                                    .LIST   ME
  2977  027226                                    TIMDLY  #1
   (1)  027226                                    SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
   (2)  027226                            M$PUT   #3,#104,#CLKINT,#340
   (3)  027226                            M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
   (4)  027226                            M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
   (5)  027226                            M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
   (6)  027226                            M$PUT1  #340
   (7)  027226                            M$GNINS <MOV    #340,-(SP)>
   (8)  027226  012746  000340                     MOV     #340,-(SP)
   (8)                                    .MEXIT
   (5)  027232                            M$PUT1  #CLKINT
   (6)  027232                            M$GNINS <MOV    #CLKINT,-(SP)>
   (7)  027232  012746  016112                     MOV     #CLKINT,-(SP)
```

```
(7)
(4)  027236                          .MEXIT
(5)  027236                          M$PUT1  #104
(6)  027236  012746  000104          M$GNINS <MOV    #104,-(SP)>
(6)                                          MOV    #104,-(SP)
(6)                                  .MEXIT
(3)  027242                          M$PUT1  #3
(4)  027242                          M$GNINS <MOV    #3,-(SP)>
(5)  027242  012746  000003                  MOV    #3,-(SP)
(5)                                  .MEXIT
(2)  027246                          M$SVC   C$SVEC
(3)  027246                          M$TSTLAB
(4)                                   .MEXIT
(3)  027246                          M$GNINS <TRAP   C$SVEC>
(4)  027246  104437                          TRAP   C$SVEC
(4)                                  .MEXIT
(2)  027250                          M$GNINS <ADD    #10,SP>
(3)  027250  062706  000010                  ADD    #10,SP
(3)                                  .MEXIT
(1)  027254  012737  000001  003142          MOV    ##1,DLYCNT      ;INITIALIZE DELAY COUNT
(1)  027262  012737  000001  172542          MOV    #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
(1)                                                                 ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
(1)  027270  012737  000113  172540          MOV    #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
(1)                                                                 ;/10 KHZ RATE,START THE CLOCK
(1)  027276  005737  003142          70$:    TST    DLYCNT          ;DELAY COUNT EXPIRED?
(1)  027302  001375                          BNE    70$             ;BRANCH IF TIME NOT ELAPSED
(1)  027304  005037  172540                  CLR    @#172540        ;STOP THE CLOCK
2978                                 .NLIST  ME
2979  027310  000736                         BR     50$
2980  027312                         51$:    ERRHRD  313.,,ERR7
(4)  027312  104456                         TRAP   C$ERHRD
(5)  027314  000471                         .WORD  313
(5)  027316  000000                         .WORD  0
(5)  027320  013542                         .WORD  ERR7
2981  027322                                 EXIT   TST
(3)  027322  104432                         TRAP   C$EXIT
(3)  027324  000266                         .WORD  L10026-.
2982  027326  012701  000120         55$:    MOV    #80.,R1         ;SET WAIT FOR 8 MS
2983  027332  004737  016544         56$:    JSR    PC,GSTATC       ;GET STATUS
2984  027336  027612                         T365$
2985  027340  032737  000001  003044         BIT    #DRDYMSK,T.CS   ;CHECK IF DRIVE READY
2986  027346  001121                         BNE    102$            ;YES - SKIP
2987  027350  005301                         DEC    R1              ;DEC COUNT
2988  027352  001432                         BEQ    60$             ;SKIP IF 0
2989                                 .LIST   ME
2990  027354                                 TIMDLY  #1
(1)  027354                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
(2)  027354                          M$PUT   #3,#104,#CLKINT,#340
(3)  027354                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
(4)  027354                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
(5)  027354                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
(6)  027354                          M$PUT1  #340
(7)  027354                          M$GNINS <MOV    #340,-(SP)>
(8)  027354  012746  000340                  MOV    #340,-(SP)
(8)                                  .MEXIT
(5)  027360                          M$PUT1  #CLKINT
(6)  027360                          M$GNINS <MOV    #CLKINT,-(SP)>
```

```
   (7)  027360  012746  016112                MOV     #CLKINT,-(SP)
   (7)                                .MEXIT
   (4)  027364                        M$PUT1  #104
   (5)  027364                        M$GNINS <MOV    #104,-(SP)>
   (6)  027364  012746  000104                MOV     #104,-(SP)
   (6)                                .MEXIT
   (3)  027370                        M$PUT1  #3
   (4)  027370                        M$GNINS <MOV    #3,-(SP)>
   (5)  027370  012746  000003                MOV     #3,-(SP)
   (5)                                .MEXIT
   (2)  027374                        M$SVC   C$SVEC
   (3)  027374                        M$TSTLAB
   (4)                                 .MEXIT
   (3)  027374                        M$GNINS <TRAP   C$SVEC>
   (4)  027374  104437                        TRAP    C$SVEC
   (4)                                .MEXIT
   (2)  027376                        M$GNINS <ADD    #10,SP>
   (3)  027376  062706  000010                ADD     #10,SP
   (3)                                .MEXIT
   (1)  027402  012737  000001  003142        MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
   (1)  027410  012737  000001  172542        MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  027416  012737  000113  172540        MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                ;/10 KHZ RATE,START THE CLOCK
   (1)  027424  005737  003142        71$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  027430  001375                        BNE     71$             ;BRANCH IF TIME NOT ELAPSED
   (1)  027432  005037  172540                CLR     @#172540        ;STOP THE CLOCK
 2991                                 .NLIST  ME
 2992  027436  000735                        BR      56$
 2993  027440  012737  006371  003012  60$:   MOV     #MISTST,ERHEAD  ;SET ERROR HEADER
 2994  027446  012704  011217                MOV     #STATE5,R4      ;SET CONDITION MESSAGE POINTER
 2995  027452  012703  010260                MOV     #MDRDY,R3       ;SET NAME MESSAGE POINTER
 2996  027456                                ERRHRD  314,,ERR5
  (4)  027456  104456                        TRAP    C$ERHRD
  (5)  027460  000472                        .WORD   314
  (5)  027462  000000                        .WORD   0
  (5)  027464  012572                        .WORD   ERR5
 2997  027466  000451                        BR      102$            ;EXIT TEST
 2998                                ;MAKE DRIVE READY FOR SUBSEQUENT TESTS
 2999  027470  004737  016512        100$:   JSR     PC,TSTINT       ;INITIALIZE TEST
 3000  027474  004737  016530                JSR     PC,GSTATR       ;GET STATUS WITH RESET
 3001  027500  027612                        T365$
 3002  027502  032737  000001  003044        BIT     #DRDYMSK,T.CS   ;CHECK IF DRIVE IS READY
 3003  027510  001040                        BNE     102$            ;BRANCH IF DRIVE IS READY
 3004  027512                        101$:   PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (13) 027512  005046                        CLR     -(SP)
  (13) 027514  153716  003033                BISB    RLDRV+1,(SP)
  (12) 027520  012746  006051                MOV     #DRVNAM,-(SP)
  (11) 027524  013746  003026                MOV     RLBAS,-(SP)
 (10)  027530  012746  006040                MOV     #BASADD,-(SP)
  (9)  027534  012746  010011                MOV     #OPR1A,-(SP)
  (8)  027540  012746  007555                MOV     #OPR6,-(SP)
  (7)  027544  012746  011316                MOV     #FMTOP1,-(SP)
  (6)  027550  012746  000007                MOV     #7,-(SP)
  (3)  027554  010600                        MOV     SP,R0
  (4)  027556  104417                        TRAP    C$PNTF
```

```
  (4)  027560  062706  000020                ADD     #20,SP
3005                                                    ;ELSE, PROMPT OPERATOR TO 'PRESS LOAD
3006                                                    ;/& WAIT FOR READY''
3007   027564  005037  004362                CLR     OBUFF           ;CLEAR FOR RESPONSE
3008   027570                                GMANIL  OPR002,OBUFF,1,NO       ;PROMPT OPERATOR FOR RESPONSE
  (3)  027570  104443                        TRAP    C$GMAN
  (3)  027572  000404                        BR      10002$
  (4)  027574  004362                        .WORD   OBUFF
  (5)  027576  000120                        .WORD   T$CODE
  (5)  027600  007317                        .WORD   OPR002
  (5)  027602  000001                        .WORD   1
  (3)  027604                           10002$:
3009   027604  005737  004362                TST     OBUFF           ;TEST IF RESPONSE IS YES
3010   027610  001740                        BEQ     101$            ;BRANCH IF NOT READY
3011   027612                           102$:
3012   027612                           T365$:
3013   027612                           ENDTST
  (3)  027612                           L10026:
  (3)  027612  104401                        TRAP    C$ETST
3014
3015
3016
3017                                    .SBTTL  *TEST 4          HEAD UNLOADING
3018   027614                           BGNTST                  ;TEST04
  (3)  027614                                                                    T4::
3019   027614  005737  003356                TST     PASNUM          ;TEST IF FIRST PASS
3020   027620  001003                        BNE     8$              ;NO - SKIP
3021   027622  005737  014202                TST     MISWIW          ;TEST IF MANUAL INTERVENTION
3022   027626  100402                        BMI     10$             ;YES - SKIP
3023   027630                           8$:    EXIT    TST
  (3)  027630  104432                        TRAP    C$EXIT
  (3)  027632  001054                        .WORD   L10027-.
3024                                    ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3025   027634  005737  003144           10$:   TST     CLKFLG          ;P-CLOCK?
3026   027640  001024                        BNE     TST4            ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3027   027642  012702  006662                MOV     #NOTST,R2       ;INITIALIZE POINTER FOR TEST MSG.
3028   027646  112762  000060  000004        MOVB    #'0,4(R2)       ;INSERT TEST NUMBER INTO MSG.
3029   027654  112762  000064  000005        MOVB    #'4,5(R2)       ;INSERT TEST NUMBER INTO MSG.
3030   027662                                PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 4 CANNOT BE PERFORMED...
  (8)  027662  012746  006662                MOV     #NOTST,-(SP)
  (7)  027666  012746  011627                MOV     #FMT9,-(SP)
  (6)  027672  012746  000002                MOV     #2,-(SP)
  (3)  027676  010600                        MOV     SP,R0
  (4)  027700  104417                        TRAP    C$PNTF
  (4)  027702  062706  000006                ADD     #6,SP
3031                                                    ;/NO P-CLK''
3032   027706  000750                        BR      8$              ;EXIT TEST
3033
3034   027710                           BGNSUB                                 T4.1:
  (3)  027710
  (3)  027710  104402                        TRAP    C$BSUB
3035   027712  012737  006414  003012  TST4:  MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
3036   027720  004737  016512                JSR     PC,TSTINT       ;INITIALIZE TEST
3037   027724  004737  016530                JSR     PC,GSTATR       ;GET STATUS
3038   027730  030576                        T465$
3039   027732  032737  000001  003044        BIT     #DRDYMSK,T.CS   ;CHECK IF DRIVE READY
```

```
                        BNE     3$                      ;YES - SKIP
                1$:     PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 027742  005046            CLR     -(SP)
 (13) 027744  153716  003033    BISB    RLDRV+1,(SP)
 (12) 027750  012746  006051    MOV     #DRVNAM,-(SP)
 (11) 027754  013746  003026    MOV     RLBAS,-(SP)
 (10) 027760  012746  006040    MOV     #BASADD,-(SP)
 (9)  027764  012746  010011    MOV     #OPR1A,-(SP)
 (8)  027770  012746  007555    MOV     #OPR6,-(SP)
 (7)  027774  012746  011316    MOV     #FMTOP1,-(SP)
 (6)  030000  012746  000007    MOV     #7,-(SP)
 (3)  030004  010600            MOV     SP,R0
 (4)  030006  104417            TRAP    C$PNTF
 (4)  030010  062706  000020    ADD     #20,SP
3042  030014  005037  004362    CLR     OBUFF                   ;CLEAR FOR RESPONSE
3043  030020                    GMANIL  OPR002,OBUFF,1,NO
 (3)  030020  104443            TRAP    C$GMAN
 (3)  030022  000404            BR      10000$
 (4)  030024  004362            .WORD   OBUFF
 (5)  030026  000120            .WORD   T$CODE
 (5)  030030  007317            .WORD   OPR002
 (5)  030032  000001            .WORD   1
 (3)  030034                10000$:
3044  030034  005737  004362    TST     OBUFF                   ;TST RESPONSE YES
3045  030040  001740            BEQ     1$                      ;NO - SKIP
3046
3047  030042  052737  000010  003004  3$:     BIS     #ULOAD,OPFLAG   ;SET UNLOAD OPERATION
3048  030050                    4$:     PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 030050  005046            CLR     -(SP)
 (13) 030052  153716  003033    BISB    RLDRV+1,(SP)
 (12) 030056  012746  006051    MOV     #DRVNAM,-(SP)
 (11) 030062  013746  003026    MOV     RLBAS,-(SP)
 (10) 030066  012746  006040    MOV     #BASADD,-(SP)
 (9)  030072  012746  010011    MOV     #OPR1A,-(SP)
 (8)  030076  012746  007477    MOV     #OPR3,-(SP)
 (7)  030102  012746  011316    MOV     #FMTOP1,-(SP)
 (6)  030106  012746  000007    MOV     #7,-(SP)
 (3)  030112  010600            MOV     SP,R0
 (4)  030114  104417            TRAP    C$PNTF
 (4)  030116  062706  000020    ADD     #20,SP
3049  030122  012703  000006    MOV     #6,R3                   ;SET EXPECTED STATE VALUE
3050  030126  012704  000144    MOV     #100.,R4                ;SET SECOND LEVEL COUNT
3051  030132  012701  001274    MOV     #700.,R1                ;SET WAIT COUNT FOR 30 SECONDS
3052  030136  004737  016544    5$:     JSR     PC,GSTATC       ;GET STATUS
3053  030142  030576            T465$
3054  030144  020337  003060    CMP     R3,T.STAT               ;CHECK IF STATE 6
3055  030150  001465            BEQ     11$                     ;YES - SKIP
3056  030152  022737  000005  003060  CMP     #5,T.STAT         ;TEST IF STATE 5
3057  030160  001053            BNE     9$                      ;NO - REPORT WRONG STATE
3058  030162  005304            8$:     DEC     R4              ;DEC 2ND LEVEL COUNT
3059  030164  001004            BNE     6$                      ;SKIP IF NOT 0
3060  030166  005301            DEC     R1                      ;ELSE DEC 1ST LEVEL COUNT
3061  030170  001434            BEQ     7$                      ;IF 0 - SKIP TO QUESTION
3062  030172  012704  000144    MOV     #100.,R4                ;ELSE RESET 2ND LEVEL
3063                        .LIST   ME
3064  030176                    6$:     TIMDLY  #1              ;WAIT 100 US
```

```
(1)  030176                                        SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
(2)  030176                                M$PUT   #3,#104,#CLKINT,#340
(3)  030176                                M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
(4)  030176                                M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
(5)  030176                                M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
(6)  030176                                M$PUT1  #340
(7)  030176                                M$GNINS <MOV    #340,-(SP)>
(8)  030176  012746  000340                        MOV     #340,-(SP)
(8)                                        .MEXIT
(5)  030202                                M$PUT1  #CLKINT
(6)  030202                                M$GNINS <MOV    #CLKINT,-(SP)>
(7)  030202  012746  016112                        MOV     #CLKINT,-(SP)
(7)                                        .MEXIT
(4)  030206                                M$PUT1  #104
(5)  030206                                M$GNINS <MOV    #104,-(SP)>
(6)  030206  012746  000104                        MOV     #104,-(SP)
(6)                                        .MEXIT
(3)  030212                                M$PUT1  #3
(4)  030212                                M$GNINS <MOV    #3,-(SP)>
(5)  030212  012746  000003                        MOV     #3,-(SP)
(5)                                        .MEXIT
(2)  030216                                M$SVC   C$SVEC
(3)  030216                                M$TSTLAB
(4)                                         .MEXIT
(3)  030216                                M$GNINS <TRAP   C$SVEC>
(4)  030216  104437                                TRAP    C$SVEC
(4)                                        .MEXIT
(2)  030220                                M$GNINS <ADD    #10,SP>
(3)  030220  062706  000010                        ADD     #10,SP
(3)                                        .MEXIT
(1)  030224  012737  000001  003142                MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
(1)  030232  012737  000001  172542                MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
(1)                                                                        ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
(1)  030240  012737  000113  172540                MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
(1)                                                                        ;/10 KHZ RATE,START THE CLOCK
(1)  030246  005737  003142                64$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
(1)  030252  001375                                BNE     64$             ;BRANCH IF TIME NOT ELAPSED
(1)  030254  005037  172540                        CLR     @#172540        ;STOP THE CLOCK
3065                                       .NLIST  ME
3066 030260  000726                                BR      5$
3067 030262  005037  004362                7$:     CLR     OBUFF           ;CLEAR FOR RESPONSE
3068 030266                                        GMANIL  OPR003,OBUFF,1,NO
 (3) 030266  104443                                TRAP    C$GMAN
 (3) 030270  000404                                BR      10001$
 (4) 030272  004362                                .WORD   OBUFF
 (5) 030274  000120                                .WORD   T$CODE
 (5) 030276  007344                                .WORD   OPR003
 (5) 030300  000001                                .WORD   1
 (3) 030302                                10001$:
3069 030302  005737  004362                        TST     OBUFF           ;TEST IF RESPONSE YES
3070 030306  001660                                BEQ     4$              ;NO - SKIP
3071 030310                                9$:     ERRHRD  401,,ERR7       ;ELSE REPORT STATE CHANGE WRONG
 (4) 030310  104456                                TRAP    C$ERHRD
 (5) 030312  000621                                .WORD   401
 (5) 030314  000000                                .WORD   0
 (5) 030316  013542                                .WORD   ERR7
```

```
3072  030320
 (3)  030320  104432
 (3)  030322  000262
3073  030324  012703  000007      11$:   MOV    #7,R3            ;SET EXPECTED STATE VALUE
3074  030330  012701  005670             MOV    #3000.,R1        ;SET COUNT FOR 300MS
3075  030334  004737  016544      12$:   JSR    PC,GSTATC        ;GET STATUS
3076  030340  030576                     T465$
3077  030342  020337  003060             CMP    R3,T.STAT        ;CHECK IF STATE 7
3078  030346  001442                     BEQ    18$              ;YES - SKIP
3079  030350  005301                     DEC    R1               ;DEC WAIT COUNT
3080  030352  001432                     BEQ    16$              ;SKIP IF 0
3081                              .LIST  ME
3082  030354                             TIMDLY #1
 (1)  030354                             SETVEC #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
 (2)  030354                      M$PUT  #3,#104,#CLKINT,#340
 (3)  030354                      M$PUT  <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
 (4)  030354                      M$PUT  <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
 (5)  030354                      M$PUT  <#340>,<>,<>,<>,<>,<>,<>,<>,<>
 (6)  030354                      M$PUT1 #340
 (7)  030354                      M$GNINS <MOV    #340,-(SP)>
 (8)  030354  012746  000340             MOV    #340,-(SP)
 (8)                              .MEXIT
 (5)  030360                      M$PUT1 #CLKINT
 (6)  030360                      M$GNINS <MOV    #CLKINT,-(SP)>
 (7)  030360  012746  016112             MOV    #CLKINT,-(SP)
 (7)                              .MEXIT
 (4)  030364                      M$PUT1 #104
 (5)  030364                      M$GNINS <MOV    #104,-(SP)>
 (6)  030364  012746  000104             MOV    #104,-(SP)
 (6)                              .MEXIT
 (3)  030370                      M$PUT1 #3
 (4)  030370                      M$GNINS <MOV    #3,-(SP)>
 (5)  030370  012746  000003             MOV    #3,-(SP)
 (5)                              .MEXIT
 (2)  030374                      M$SVC  C$SVEC
 (3)  030374                      M$TSTLAB
 (4)                               .MEXIT
 (3)  030374                      M$GNINS <TRAP   C$SVEC>
 (4)  030374  104437                     TRAP   C$SVEC
 (4)                              .MEXIT
 (2)  030376                      M$GNINS <ADD    #10,SP>
 (3)  030376  062706  000010             ADD    #10,SP
 (3)                              .MEXIT
 (1)  030402  012737  000001  003142      MOV    ##1,DLYCNT       ;INITIALIZE DELAY COUNT
 (1)  030410  012737  000001  172542      MOV    #1,@#172542      ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
 (1)                                                             ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
 (1)  030416  012737  000113  172540      MOV    #113,@#172540    ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
 (1)                                                             ;/10 KHZ RATE,START THE CLOCK
 (1)  030424  005737  003142      65$:   TST    DLYCNT           ;DELAY COUNT EXPIRED?
 (1)  030430  001375                     BNE    65$              ;BRANCH IF TIME NOT ELAPSED
 (1)  030432  005037  172540             CLR    @#172540         ;STOP THE CLOCK
3083                              .NLIST ME
3084  030436  000736                     BR     12$
3085  030440                      16$:   ERRHRD 402,,ERR7         ;REPORT WRONG STATE CHANGE
 (4)  030440  104456                     TRAP   C$ERHRD
 (5)  030442  000622                     .WORD  402
```

```
   (5)  030444  000000                        .WORD   0
   (5)  030446  013542                        .WORD   ERR7
 3086   030450                                EXIT    SUB
   (3)  030450  104432                        TRAP    C$EXIT
   (3)  030452  000132                        .WORD   L10030-.
 3087   030454  005003                18$:    CLR     R3              ;SET EXPECTED STATE VALUE
 3088   030456  012701  001130                MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
 3089   030462  004737  016544        20$:    JSR     PC,GSTATC       ;GET STATUS
 3090   030466  030576                        T465$
 3091   030470  005737  003060                TST     T.STAT          ;CHECK IF STATE 0
 3092   030474  001440                        BEQ     24$             ;YES - SKIP
 3093   030476  005301                        DEC     R1              ;DEC WAIT COUNT
 3094   030500  001432                        BEQ     22$             ;SKIP IF 0
 3095                                .LIST   ME
 3096   030502                               TIMDLY  #1000.
   (1)  030502                               SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
   (2)  030502                          M$PUT   #3,#104,#CLKINT,#340
   (3)  030502                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
   (4)  030502                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
   (5)  030502                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
   (6)  030502                          M$PUT1  #340
   (7)  030502                          M$GNINS <MOV    #340,-(SP)>
   (8)  030502  012746  000340                 MOV     #340,-(SP)
   (8)                                  .MEXIT
   (5)  030506                          M$PUT1  #CLKINT
   (6)  030506                          M$GNINS <MOV    #CLKINT,-(SP)>
   (7)  030506  012746  016112                 MOV     #CLKINT,-(SP)
   (7)                                  .MEXIT
   (4)  030512                          M$PUT1  #104
   (5)  030512                          M$GNINS <MOV    #104,-(SP)>
   (6)  030512  012746  000104                 MOV     #104,-(SP)
   (6)                                  .MEXIT
   (3)  030516                          M$PUT1  #3
   (4)  030516                          M$GNINS <MOV    #3,-(SP)>
   (5)  030516  012746  000003                 MOV     #3,-(SP)
   (5)                                  .MEXIT
   (2)  030522                          M$SVC   C$SVEC
   (3)  030522                          M$TSTLAB
   (4)                                   .MEXIT
   (3)  030522                          M$GNINS <TRAP   C$SVEC>
   (4)  030522  104437                         TRAP    C$SVEC
   (4)                                  .MEXIT
   (2)  030524                          M$GNINS <ADD    #10,SP>
   (3)  030524  062706  000010                 ADD     #10,SP
   (3)                                  .MEXIT
   (1)  030530  012737  001750  003142        MOV     ##1000.,DLYCNT  ;INITIALIZE DELAY COUNT
   (1)  030536  012737  000001  172542        MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                 ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  030544  012737  000113  172540        MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                 ;/10 KHZ RATE,START THE CLOCK
   (1)  030552  005737  003142        66$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  030556  001375                        BNE     66$             ;BRANCH IF TIME NOT ELAPSED
   (1)  030560  005037  172540                CLR     @#172540        ;STOP THE CLOCK
 3097                                 .NLIST  ME
 3098   030564  000736                        BR      20$
 3099   030566                        22$:    ERRHRD  403.,,ERR7      ;REPORT WRONG STATE CHANGE
```

```
   (4)  030566  104456                          TRAP    C$ERHRD
   (5)  030570  000623                          .WORD   403
   (5)  030572  000000                          .WORD   0
   (5)  030574  013542                          .WORD   ERR7
 3100   030576                              24$:
 3101   030576  012737  000002  003016      T465$:  MOV     #2,ERRSWI       ;INIT ERROR SWITCH
 3102
 3103   030604                              ENDSUB
  (3)   030604                              L10030:
  (3)   030604  104403                          TRAP    C$ESUB
 3104   030606                              26$:    PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CYCLE UP
  (13)  030606  005046                          CLR     -(SP)
  (13)  030610  153716  003033                  BISB    RLDRV+1,(SP)
  (12)  030614  012746  006051                  MOV     #DRVNAM,-(SP)
  (11)  030620  013746  003026                  MOV     RLBAS,-(SP)
  (10)  030624  012746  006040                  MOV     #BASADD,-(SP)
   (9)  030630  012746  010011                  MOV     #OPR1A,-(SP)
   (8)  030634  012746  007555                  MOV     #OPR6,-(SP)
   (7)  030640  012746  011316                  MOV     #FMTOP1,-(SP)
   (6)  030644  012746  000007                  MOV     #7,-(SP)
   (3)  030650  010600                          MOV     SP,R0
   (4)  030652  104417                          TRAP    C$PNTF
   (4)  030654  062706  000020                  ADD     #20,SP
 3105   030660  005037  004362                  CLR     OBUFF           ;CLEAR FOR RESPONSE
 3106   030664                                  GMANIL  OPR002,OBUFF,1,NO
  (3)   030664  104443                          TRAP    C$GMAN
  (3)   030666  000404                          BR      10000$
  (4)   030670  004362                          .WORD   OBUFF
  (5)   030672  000120                          .WORD   T$CODE
  (5)   030674  007317                          .WORD   OPR002
  (5)   030676  000001                          .WORD   1
  (3)   030700                              10000$:
 3107   030700  005737  004362                  TST     OBUFF           ;TEST IF RESPONSE YES
 3108   030704  001740                          BEQ     26$             ;NO - SKIP
 3109   030706                              29$:
 3110
 3111   030706                              ENDTST
  (3)   030706                              L10027:
  (3)   030706  104401                          TRAP    C$ETST
 3112
 3113
 3114
 3115                                       .SBTTL  *TEST 5         DRIVE SELECT
 3116   030710                              BGNTST                  ;TEST05
  (3)   030710                                                                  T5::
 3117   030710  012737  000002  003016          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
 3118   030716  005737  003356                  TST     PASNUM          ;TEST IF FIRST PASS
 3119   030722  001173                          BNE     EXT05           ;NO - SKIP
 3120   030724  032737  000004  014202          BIT     #DRSELT,MISWIW  ;TEST IF SELECT TESTS
 3121   030732  001567                          BEQ     EXT05           ;NO - SKIP
 3122   030734                              1$:     PRINTF  #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
  (13)  030734  005046                          CLR     -(SP)
  (13)  030736  153716  003033                  BISB    RLDRV+1,(SP)
  (12)  030742  012746  006051                  MOV     #DRVNAM,-(SP)
  (11)  030746  013746  003026                  MOV     RLBAS,-(SP)
  (10)  030752  012746  006040                  MOV     #BASADD,-(SP)
```

```
  (9)  030756  012746  010011                    MOV     #OPR1A,-(SP)
  (8)  030762  012746  007610                    MOV     #OPR7,-(SP)
  (7)  030766  012746  011316                    MOV     #FMTOP1,-(SP)
  (6)  030772  012746  000007                    MOV     #7,-(SP)
  (3)  030776  010600                            MOV     SP,R0
  (4)  031000  104417                            TRAP    C$PNTF
  (4)  031002  062706  000020                    ADD     #20,SP
3123
3124   031006  005037  004362                    CLR     OBUFF           ;REQUEST 'REMOVE ADD PLGS EXCPT ''
3125   031012                                    GMANIL  OPR002,OBUFF,1,NO       ;CLEAR FOR RESPONSE
  (3)  031012  104443                            TRAP    C$GMAN
  (3)  031014  000404                            BR      10000$
  (4)  031016  004362                            .WORD   OBUFF
  (5)  031020  000120                            .WORD   T$CODE
  (5)  031022  007317                            .WORD   OPR002
  (5)  031024  000001                            .WORD   1
  (3)  031026                       10000$:
3126   031026  005737  004362                    TST     OBUFF           ;TEST RESPONSE YES
3127   031032  001740                            BEQ     1$              ;NO - SKIP
3128   031034  012737  006526  003012  3$:       MOV     #T05ERR,ERHEAD  ;SET ERROR HEADER MESSAGE
3129   031042  004737  016512                    JSR     PC,TSTINT       ;INITIALIZE TEST
3130   031046  004737  016544                    JSR     PC,GSTATC       ;DO SELECT AND GET STATUS
3131   031052  031234                            T504$
3132   031054  013737  003032  003116            MOV     RLDRV,TEMP0     ;STORE ORIGINAL DRIVE NUMBER
3133   031062  013701  003032                    MOV     RLDRV,R1        ;PUT IT IN R1
3134   031066  012704  000004                    MOV     #4,R4           ;SET COUNT FOR NUMBER OF PLUGS
3135   031072  062701  000400            LPT05:  ADD     #400,R1         ;BUMP TO NEXT DRIVE
3136   031076  022701  002000                    CMP     #2000,R1        ;CHECK IF TO LARGE
3137   031102  001001                            BNE     4$              ;NO - SKIP
3138   031104  005001                            CLR     R1              ;ELSE CLEAR TO DRIVE 0
3139   031106  010137  003032            4$:     MOV     R1,RLDRV        ;PUT IT BACK IN RLDRV
3140   031112                            5$:     PRINTF  #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
  (11) 031112  012746  010025                    MOV     #UNDTST,-(SP)
  (10) 031116  012746  010015                    MOV     #OPR1B,-(SP)
  (9)  031122  005046                            CLR     -(SP)
  (9)  031124  153716  003033                    BISB    RLDRV+1,(SP)
  (8)  031130  012746  007637                    MOV     #OPR8,-(SP)
  (7)  031134  012746  011367                    MOV     #FMTOP3,-(SP)
  (6)  031140  012746  000005                    MOV     #5,-(SP)
  (3)  031144  010600                            MOV     SP,R0
  (4)  031146  104417                            TRAP    C$PNTF
  (4)  031150  062706  000014                    ADD     #14,SP
3141
3142   031154  005037  004362                    CLR     OBUFF           ;INSERT PLUG REQUEST
3143   031160                                    GMANIL  OPR002,OBUFF,1,NO       ;CLEAR FOR RESPONSE
  (3)  031160  104443                            TRAP    C$GMAN
  (3)  031162  000404                            BR      10001$
  (4)  031164  004362                            .WORD   OBUFF
  (5)  031166  000120                            .WORD   T$CODE
  (5)  031170  007317                            .WORD   OPR002
  (5)  031172  000001                            .WORD   1
  (3)  031174                       10001$:
3144   031174  005737  004362                    TST     OBUFF           ;TEST RESPONSE YES
3145   031200  001744                            BEQ     5$              ;NO - SKIP
3146   031202                            BGNSUB
  (3)  031202                                                                       T5.1:
```

```
  (3)  031202  104402                          TRAP    C$BSUB
3147   031204  004737  016544                  JSR     PC,GSTATC       ;GET STATUS - REPORT ANY ERROR
3148   031210  031212                          60$
3149   031212  012737  000002  003016  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3150
3151   031220                          ENDSUB
  (3)  031220                          L10032:
  (3)  031220  104403                          TRAP    C$ESUB
3152   031222  005304                          DEC     R4              ;DEC COUNT
3153   031224  001322                          BNE     LPT05           ;LOOP IF NOT ZERO
3154   031226  013737  003116  003032          MOV     TEMP0,RLDRV     ;ELSE RESTORE RLDRV
3155   031234                          T504$:
3156   031234                          4$:     PRINTF  #FMT4,#OPR8,#OPR9
  (9)  031234  012746  007656                  MOV     #OPR9,-(SP)
  (8)  031240  012746  007637                  MOV     #OPR8,-(SP)
  (7)  031244  012746  011432                  MOV     #FMT4,-(SP)
  (6)  031250  012746  000003                  MOV     #3,-(SP)
  (3)  031254  010600                          MOV     SP,R0
  (4)  031256  104417                          TRAP    C$PNTF
  (4)  031260  062706  000010                  ADD     #10,SP
3157   031264  005037  004362                  CLR     OBUFF           ;CLEAR FOR RESPONSE
3158   031270                                  GMANIL  OPR002,OBUFF,1,NO
  (3)  031270  104443                          TRAP    C$GMAN
  (3)  031272  000404                          BR      10000$
  (4)  031274  004362                          .WORD   OBUFF
  (5)  031276  000120                          .WORD   T$CODE
  (5)  031300  007317                          .WORD   OPR002
  (5)  031302  000001                          .WORD   1
  (3)  031304                          10000$:
3159   031304  005737  004362                  TST     OBUFF           ;TEST IF RESPONSE YES
3160   031310  001751                          BEQ     4$              ;NO - SKIP
3161   031312                          EXT05:
3162   031312                          ENDTST
  (3)  031312                          L10031:
  (3)  031312  104401                          TRAP    C$ETS1
3163
3164
3165
3166                                    .SBTTL  *TEST 6         DRIVE SELECT ERROR TEST
3167   031314                          BGNTST                  ;TEST06
  (3)  031314                                                                  T6::
3168   031314  005737  003356                  TST     PASNUM          ;CHECK IF FIRST PASS
3169   031320  001004                          BNE     1$              ;NO - SKIP
3170   031322  032737  000004  014202          BIT     #DRSELT,MISWIW  ;CHECK IF TEST DRIVE SELECT
3171   031330  001002                          BNE     4$              ;YES - SKIP
3172   031332                          1$:     EXIT    TST
  (3)  031332  104432                          TRAP    C$EXIT
  (3)  031334  001132                          .WORD   L10033-.
3173                                    ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3174   031336  005737  003144          4$:     TST     CLKFLG          ;P-CLOCK?
3175   031342  001023                          BNE     6$              ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3176   031344  012702  006662                  MOV     #NOTST,R2       ;INITIALIZE POINTER FOR TEST MSG.
3177   031350  112762  000060  000004          MOVB    #'0,4(R2)       ;INSERT TEST NUMBER INTO MSG.
3178   031356  112762  000066  000005          MOVB    #'6,5(R2)       ;INSERT TEST NUMBER INTO MSG.
3179   031364                                  PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 6 CANNOT BE PERFORMED...
  (8)  031364  012746  006662                  MOV     #NOTST,-(SP)
```

```
  (7)  031370  012746  011627                    MOV     #FMT9,-(SP)
  (6)  031374  012746  000002                    MOV     #2,-(SP)
  (3)  031400  010600                            MOV     SP,R0
  (4)  031402  104417                            TRAP    C$PNTF
  (4)  031404  062706  000006                    ADD     #6,SP
3180                                                                ;/NO P-CLK''
3181   031410  000750                            BR      1$          ;EXIT TEST
3182   031412  012737  006462  003012  6$:      MOV     #GSTER1,ERHEAD  ;SET ERROR HEADER
3183   031420  004737  016512                    JSR     PC,TSTINT   ;INITIALIZE TEST
3184   031424  013703  003360                    MOV     PSETNM,R3   ;GET PARAM SET NUMBER
3185   031430  023727  002012  000001            CMP     L$UNIT,#1   ;TEST IF MORE THAN 1 UNIT
3186   031436  101476                            BLOS    5$          ;NO - SKIP
3187   031440  005203                    2$:     INC     R3          ;BUMP PARAMETER SET NUMBER
3188   031442  020337  002012                    CMP     R3,L$UNIT   ;CHECK IF PAST VALID PARAMETER TABLE
3189   031446  101401                            BLOS    3$          ;NO - SKIP
3190   031450  005003                            CLR     R3          ;ELSE CLEAR TO POINT TO ENTRY 0
3191   031452                            3$:     GPHARD  R3,R0
  (3)  031452  010300                            MOV     R3,R0
  (3)  031454  104442                            TRAP    C$GPHRD
3192   031456                                    BNCOMPLETE      2$  ;SKIP IF NOT AVAILABLE
  (2)  031456  103370                            BCC     2$
3193   031460  010004                            MOV     R0,R4       ;PUT POINTER INTO R4
3194   031462  021437  003026                    CMP     (R4),RLBAS  ;CHECK IF SAME CONTROLLER
3195   031466  001364                            BNE     2$          ;NO - SKIP
3196   031470  005037  003006                    CLR     DONE        ;CLEAR DONE FLAG
3197   031474  012737  000104  003034            MOV     #GTSTAT,L.CS    ;LOAD GET STATUS
3198   031502  056437  000010  003034            BIS     10(R4),L.CS     ;INSERT DRIVE
3199   031510  012737  000013  003040            MOV     #GETSTAT!DRSET,L.DA     ;SET UP TO CLEAR DRIVE
3200   031516  013762  003040  000004            MOV     L.DA,RLDA(R2)   ;LOAD DA REG
3201   031524  013762  003034  000000            MOV     L.CS,RLCS(R2)   ;LOAD CS REG
3202                                    .LIST   ME
3203   031532                                    TIMDLY  #30.            ;WAIT 3 MS
  (1)  031532                                    SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  031532                           M$PUT   #3,#104,#CLKINT,#340
  (3)  031532                           M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  031532                           M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  031532                           M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  031532                           M$PUT1  #340
  (7)  031532                           M$GNINS <MOV    #340,-(SP)>
  (8)  031532  012746  000340                      MOV     #340,-(SP)
  (8)                                   .MEXIT
  (5)  031536                           M$PUT1  #CLKINT
  (6)  031536                           M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  031536  012746  016112                      MOV     #CLKINT,-(SP)
  (7)                                   .MEXIT
  (4)  031542                           M$PUT1  #104
  (5)  031542                           M$GNINS <MOV    #104,-(SP)>
  (6)  031542  012746  000104                      MOV     #104,-(SP)
  (6)                                   .MEXIT
  (3)  031546                           M$PUT1  #3
  (4)  031546                           M$GNINS <MOV    #3,-(SP)>
  (5)  031546  012746  000003                      MOV     #3,-(SP)
  (5)                                   .MEXIT
  (2)  031552                           M$SVC   C$SVEC
  (3)  031552                           M$TSTLAB
  (4)                                    .MEXIT
```

```
   (3)  031552                              MSGNINS <TRAP    C$SVEC>
   (4)  031552  104437                              TRAP    C$SVEC
   (4)                                      .MEXIT
   (2)  031554                              MSGNINS <ADD     #10,SP>
   (3)  031554  062706  000010                      ADD     #10,SP
   (3)                                      .MEXIT
   (1)  031560  012737  000036  003142              MOV     #^D30.,DLYCNT    ;INITIALIZE DELAY COUNT
   (1)  031566  012737  000001  172542              MOV     #1,@#172542      ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                              ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  031574  012737  000113  172540              MOV     #113,@#172540    ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                              ;/10 KHZ RATE,START THE CLOCK
   (1)  031602  005737  003142          64$:        TST     DLYCNT           ;DELAY COUNT EXPIRED?
   (1)  031606  001375                              BNE     64$              ;BRANCH IF TIME NOT ELAPSED
   (1)  031610  005037  172540                      CLR     @#172540         ;STOP THE CLOCK
  3204                                      .NLIST  ME
  3205  031614  005737  003006                      TST     DONE             ;TEST  IF INTERRUPT
  3206  031620  001707                              BEQ     2$               ;NO - SKIP
  3207  031622  032737  100000  003044              BIT     #ANYERR,T.CS     ;TEST IF ANY ERROR SET
  3208  031630  001415                              BEQ     7$               ;NO - GO TEST
  3209  031632  000702                              BR      2$               ;ELSE CHECK NEXT DRIVE
  3210  031634                           5$:        PRINTF  #FMT9,#OPR10     ;REPORT CAN'T FIND 2ND DRIVE
   (8)  031634  012746  007673                      MOV     #OPR10,-(SP)
   (7)  031640  012746  011627                      MOV     #FMT9,-(SP)
   (6)  031644  012746  000002                      MOV     #2,-(SP)
   (3)  031650  010600                              MOV     SP,R0
   (4)  031652  104417                              TRAP    C$PNTF
   (4)  031654  062706  000006                      ADD     #6,SP
  3211  031660  000137  032466                      JMP     LCLEXT
  3212  031664  016437  000010  003120  7$:         MOV     10(R4),TEMP1     ;STORE NEW ADDRESS
  3213                                              ;ASK FOR PLUG CHANGE
  3214  031672  013700  003032          9$:         MOV     RLDRV,R0         ;GET DRIVE UNDER TEST
  3215  031676  013705  003120                      MOV     TEMP1,R5         ;GET NEW ADDRESS
  3216  031702  042700  002000                      BIC     #2000,R0         ;CLEAR FOR ADDRESS 0 TO 3
  3217  031706  042705  002000                      BIC     #2000,R5
  3218  031712  020527  001400          20$:        CMP     R5,#1400         ;TEST IF DRIVE NUMBER 3
  3219  031716  001001                              BNE     21$              ;NO - SKIP
  3220  031720  005005                              CLR     R5               ;ELSE SET TO DRIVE NUMBER 0
  3221  031722  062705  000400          21$:        ADD     #400,R5          ;BUMP TO NEXT ADDRESS
  3222  031726  020500                              CMP     R5,R0            ;THIS EQUAL TO NEW ADDRESS?
  3223  031730  001770                              BEQ     20$              ;YES - SKIP
  3224  031732  052705  000200                      BIS     #CRDYMSK,R5      ;ELSE SET CONTROLLER READY BIT
  3225  031736  010562  000000                      MOV     R5,RLCS(R2)      ;AND LOAD CS REG
  3226  031742                                      PRINTF  #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
  (11)  031742  005046                              CLR     -(SP)
  (11)  031744  153716  003121                      BISB    TEMP1+1,(SP)
  (10)  031750  012746  010015                      MOV     #OPR1B,-(SP)
   (9)  031754  005046                              CLR     -(SP)
   (9)  031756  153716  003033                      BISB    RLDRV+1,(SP)
   (8)  031762  012746  007637                      MOV     #OPR8,-(SP)
   (7)  031766  012746  011345                      MOV     #FMTOP2,-(SP)
   (6)  031772  012746  000005                      MOV     #5,-(SP)
   (3)  031776  010600                              MOV     SP,R0
   (4)  032000  104417                              TRAP    C$PNTF
   (4)  032002  062706  000014                      ADD     #14,SP
  3227  032006  005037  004362                      CLR     OBUFF            ;CLEAR FOR RESPONSE
  3228  032012                                      GMANIL  OPR002,OBUFF,1,NO
```

```
   (3)  032012  104443                          TRAP    C$GMAN
   (3)  032014  000404                          BR      10000$
   (4)  032016  004362                          .WORD   OBUFF
   (5)  032020  000120                          .WORD   T$CODE
   (5)  032022  007317                          .WORD   OPR002
   (5)  032024  000001                          .WORD   1
   (3)  032026                          10000$:
3229    032026  005737  004362                  TST     OBUFF           ;TEST IF RESPONSE YES
3230    032032  001717                          BEQ     9$              ;NO - SKIP
3231    032034  012704  000012                  MOV     #10.,R4         ;SET COUNT
3232    032040                          BGNSUB
   (3)  032040                                                                          T6.1:
   (3)  032040  104402                          TRAP    C$BSUB
3233    032042  013737  003032  003034  8$:     MOV     RLDRV,L.CS      ;SET UP TO SELECT MULTIPLE DRIVES
3234    032050  013762  003034  000000          MOV     L.CS,RLCSR(R2)  ;DO IT
3235                                    .LIST   ME
3236    032056                                  TIMDLY  #100.
   (1)  032056                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
   (2)  032056                          M$PUT   #3,#104,#CLKINT,#340
   (3)  032056                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
   (4)  032056                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
   (5)  032056                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
   (6)  032056                          M$PUT1  #340
   (7)  032056                          M$GNINS <MOV    #340,-(SP)>
   (8)  032056  012746  000340                  MOV     #340,-(SP)
   (8)                                  .MEXIT
   (5)  032062                          M$PUT1  #CLKINT
   (6)  032062                          M$GNINS <MOV    #CLKINT,-(SP)>
   (7)  032062  012746  016112                  MOV     #CLKINT,-(SP)
   (7)                                  .MEXIT
   (4)  032066                          M$PUT1  #104
   (5)  032066                          M$GNINS <MOV    #104,-(SP)>
   (6)  032066  012746  000104                  MOV     #104,-(SP)
   (6)                                  .MEXIT
   (3)  032072                          M$PUT1  #3
   (4)  032072                          M$GNINS <MOV    #3,-(SP)>
   (5)  032072  012746  000003                  MOV     #3,-(SP)
   (5)                                  .MEXIT
   (2)  032076                          M$SVC   C$SVEC
   (3)  032076                          M$TSTLAB
   (4)                                   .MEXIT
   (3)  032076                          M$GNINS <TRAP   C$SVEC>
   (4)  032076  104437                          TRAP    C$SVEC
   (4)                                  .MEXIT
   (2)  032100                          M$GNINS <ADD    #10,SP>
   (3)  032100  062706  000010                  ADD     #10,SP
   (3)                                  .MEXIT
   (1)  032104  012737  000144  003142          MOV     ##100.,DLYCNT   ;INITIALIZE DELAY COUNT
   (1)  032112  012737  000001  172542          MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                  ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  032120  012737  000113  172540          MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                  ;/10 KHZ RATE,START THE CLOCK
   (1)  032126  005737  003142          64$     TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  032132  001375                          BNE     64$             ;BRANCH IF TIME NOT ELAPSED
   (1)  032134  005037  172540                  CLR     @#172540        ;STOP THE CLOCK
3237                                    .NLIST  ME
```

```
3238  032140  052737  000104  003034          BIS     #GTSTAT,L.CS    ;SET GET STATUS
3239  032146  012737  000013  003040          MOV     #GETSTAT!DRSET,L.DA     ;SET RESET BIT 3 IN THE DA REG FOR THE
3240                                                                          ;/DRIVE TO CLEAR ITS ERROR REGISTER
3241                                                                          ;/BEFORE SENDING A STATUS WORD TO THE
3242                                                                          ;/MP REG DURING GET STATUS COMMAND
3243
3244  032154  013762  003040  000004          MOV     L.DA,RLDA(R2)
3245  032162  005037  003006                  CLR     DONE
3246  032166  013762  003034  000000          MOV     L.CS,RLCSR(R2)  ;DO GET STATUS
3247  032174                                  WAITUS  #1              ;WAIT FOR INTERRUPT
 (3)  032174  012727  000001                  MOV     ###1,(PC)+
 (3)  032200  000000                          .WORD   0
 (3)  032202  013727  002116                  MOV     L$DLY,(PC)+
 (3)  032206  000000                          .WORD   0
 (3)  032210  005367  177772                  DEC     -6(PC)
 (3)  032214  001375                          BNE     .-4
 (3)  032216  005367  177756                  DEC     -22(PC)
 (3)  032222  001367                          BNE     .-20
3248  032224  005737  003006                  TST     DONE            ;CHECK IF INTERRUPTED
3249  032230  001012                          BNE     12$             ;YES - SKIP
3250  032232  004737  016320                  JSR     PC,WAITIN       ;WAIT FOR TIMEOUT
3251  032236  012603                          MOV     (SP)+,R3        ;GET ERROR POINTER
3252  032240  001406                          BEQ     12$             ;SKIP IF 0
3253  032242                                  ERRHRD  601.,GSTER1,ERR1
 (4)  032242  104456                          TRAP    C$ERHRD
 (5)  032244  001131                          .WORD   601
 (5)  032246  006462                          .WORD   GSTER1
 (5)  032250  012340                          .WORD   ERR1
3254  032252                                  EXIT    SUB
 (3)  032252  104432                          TRAP    C$EXIT
 (3)  032254  000136                          .WORD   L10034-.
3255                                  .LIST   ME
3256  032256                          12$:    TIMDLY  #20.            ;WAIT FOR DSE TO SET
 (1)  032256                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
 (2)  032256                          M$PUT   #3,#104,#CLKINT,#340
 (3)  032256                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
 (4)  032256                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
 (5)  032256                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
 (6)  032256                          M$PUT1  #340
 (7)  032256                          M$GNINS <MOV    #340,-(SP)>
 (8)  032256  012746  000340                  MOV     #340,-(SP)
 (8)                                  .MEXIT
 (5)  032262                          M$PUT1  #CLKINT
 (6)  032262                          M$GNINS <MOV    #CLKINT,-(SP)>
 (7)  032262  012746  016112                  MOV     #CLKINT,-(SP)
 (7)                                  .MEXIT
 (4)  032266                          M$PUT1  #104
 (5)  032266                          M$GNINS <MOV    #104,-(SP)>
 (6)  032266  012746  000104                  MOV     #104,-(SP)
 (6)                                  .MEXIT
 (3)  032272                          M$PUT1  #3
 (4)  032272                          M$GNINS <MOV    #3,-(SP)>
 (5)  032272  012746  000003                  MOV     #3,-(SP)
 (5)                                  .MEXIT
 (2)  032276                          M$SVC   C$SVEC
 (3)  032276                          M$TSTLAB
```

```
 (4)                                    .MEXIT
 (3)  032276                            M$GNINS <TRAP   C$SVEC>
 (4)  032276  104437                            TRAP    C$SVEC
 (4)                                    .MEXIT
 (2)  032300                            M$GNINS <ADD    #10,SP>
 (3)  032300  062706  000010                    ADD     #10,SP
 (3)                                    .MEXIT
 (1)  032304  012737  000024  003142            MOV     ##20.,DLYCNT    ;INITIALIZE DELAY COUNT
 (1)  032312  012737  000001  172542            MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
 (1)                                                                    ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
 (1)  032320  012737  000113  172540            MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
 (1)                                                                    ;/10 KHZ RATE,START THE CLOCK
 (1)  032326  005737  003142            65$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
 (1)  032332  001375                            BNE     65$             ;BRANCH IF TIME NOT ELAPSED
 (1)  032334  005037  172540                    CLR     @#172540        ;STOP THE CLOCK
3257                                    .NLIST  ME
3258  032340  004737  017652                    JSR     PC,GDRSTA       ;GET STATUS
3259  032344  032737  000400  003052            BIT     #DSESTAT,T.MP   ;TEST IF DRIVE SELECT ERROR SET
3260  032352  001010                            BNE     16$             ;YES - SKIP
3261  032354  012703  010506                    MOV     #MDSERR,R3      ;SET NAME MESSAGE POINTER
3262  032360                                    ERRHRD  602.,,ERR3
 (4)  032360  104456                            TRAP    C$ERHRD
 (5)  032362  001132                            .WORD   602
 (5)  032364  000000                            .WORD   0
 (5)  032366  012454                            .WORD   ERR3
3263  032370                                    EXIT    SUB
 (3)  032370  104432                            TRAP    C$EXIT
 (3)  032372  000020                            .WORD   L10034-.
3264  032374  010562  000000            16$:    MOV     R5,RLCS(R2)     ;LOAD IN DIFFERENT ADDRESS
3265  032400  005304                            DEC     R4              ;DEC COUNT
3266  032402  001217                            BNE     8$              ;LOOP IF NOT ZERO
3267  032404  012737  000002  003016    60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3268  032412                                    ENDSUB
 (3)  032412                                    L10034:
 (3)  032412  104403                            TRAP    C$ESUB
3269  032414                            15$:    PRINT   #FMT9,#OPR11    ;REQUEST PLUG CHANGE
 (8)  032414  012746  007741                    MOV     #OPR11,-(SP)
 (7)  032420  012746  011627                    MOV     #FMT9,-(SP)
 (6)  032424  012746  000002                    MOV     #2,-(SP)
 (3)  032430  010600                            MOV     SP,R0
 (4)  032432  104417                            TRAP    C$PNTF
 (4)  032434  062706  000006                    ADD     #6,SP
3270  032440  005037  004362                    CLR     OBUFF           ;CLEAR FOR RESPONSE
3271  032444                                    GMANIL  OPR002,OBUFF,1,NO
 (3)  032444  104443                            TRAP    C$GMAN
 (3)  032446  000404                            BR      10000$
 (4)  032450  004362                            .WORD   OBUFF
 (5)  032452  000120                            .WORD   T$CODE
 (5)  032454  007317                            .WORD   OPR002
 (5)  032456  000001                            .WORD   1
 (3)  032460                            10000$:
3272  032460  005737  004362                    TST     OBUFF           ;TEST IF RESPONSE YES
3273  032464  001753                            BEQ     15$             ;NO - SKIP
3274  032466                            LCLEXT:
3275  032466                                    ENDTST
 (3)  032466                                    L10033:
```

```
  (3)  032466  104401                          TRAP    C$ETST
3276
3277
3278
3279                                  .SBTTL  *TEST 7          INITIAL STATE
3280  032470                          BGNTST                   ;TEST 07
  (3)  032470                                                                  T7::
3281  032470  005737  003356                  TST     PASNUM          ;CHECK IF FIRST PASS
3282  032474  001003                          BNE     1$              ;NO - EXIT TEST
3283  032476  005737  014202                  TST     MISWIW          ;CHECK IF MANUAL INTERVENTION
3284  032502  100402                          BMI     2$              ;PERFORM TEST IF MANUAL INTERVENTION
3285  032504                          1$:     EXIT    TST
  (3)  032504  104432                          TRAP    C$EXIT
  (3)  032506  000664                          .WORD   L10035-.
3286                                  ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3287  032510  005737  003144          2$:     TST     CLKFLG          ;P-CLOCK?
3288  032514  001023                          BNE     3$              ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3289  032516  012702  006662                  MOV     #NOTST,R2       ;INITIALIZE POINTER FOR TEST MSG.
3290  032522  112762  000060  000004          MOVB    #'0,4(R2)       ;INSERT TEST NUMBER INTO MSG.
3291  032530  112762  000067  000005          MOVB    #'7,5(R2)       ;INSERT TEST NUMBER INTO MSG.
3292  032536                                  PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 7 CANNOT BE PERFORMED...
  (8)  032536  012746  006662                  MOV     #NOTST,-(SP)
  (7)  032542  012746  011627                  MOV     #FMT9,-(SP)
  (6)  032546  012746  000002                  MOV     #2,-(SP)
  (3)  032552  010600                          MOV     SP,R0
  (4)  032554  104417                          TRAP    C$PNTF
  (4)  032556  062706  000006                  ADD     #6,SP
3293                                                                   ;/NO P-CLK''
3294  032562  000750                          BR      1$              ;EXIT TEST
3295  032564  012737  006513  003012  3$:     MOV     #INITST,ERHEAD  ;SET ERROR HEADER
3296  032572  004737  016512                  JSR     PC,TSTINT       ;INITIALIZE TEST
3297                                  .LIST   ME
3298  032576                                  TIMDLY  #10.            ;WAIT 1 MS
  (1)  032576                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  032576                          M$PUT   #3,#104,#CLKINT,#340
  (3)  032576                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  032576                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  032576                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  032576                          M$PUT1  #340
  (7)  032576                          M$GNINS <MOV    #340,-(SP)>
  (8)  032576  012746  000340                  MOV     #340,-(SP)
  (8)                                  .MEXIT
  (5)  032602                          M$PUT1  #CLKINT
  (6)  032602                          M$GNINS <MOV    #CLKINT,-(SP)>
  (7)  032602  012746  016112                  MOV     #CLKINT,-(SP)
  (7)                                  .MEXIT
  (4)  032606                          M$PUT1  #104
  (5)  032606                          M$GNINS <MOV    #104,-(SP)>
  (6)  032606  012746  000104                  MOV     #104,-(SP)
  (6)                                  .MEXIT
  (3)  032612                          M$PUT1  #3
  (4)  032612                          M$GNINS <MOV    #3,-(SP)>
  (5)  032612  012746  000003                  MOV     #3,-(SP)
  (5)                                  .MEXIT
  (2)  032616                          M$SVC   C$SVEC
  (3)  032616                          M$TSTLAB
```

```
(4)                                     .MEXIT
(3)  032616                             M$GNINS <TRAP   C$SVEC>
(4)  032616  104437                             TRAP    C$SVEC
(4)                                     .MEXIT
(2)  032620                             M$GNINS <ADD    #10,SP>
(3)  032620  062706  000010                     ADD     #10,SP
(3)                                     .MEXIT
(1)  032624  012737  000012  003142             MOV     ##10.,DLYCNT    ;INITIALIZE DELAY COUNT
(1)  032632  012737  000001  172542             MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
(1)                                             ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
(1)  032640  012737  000113  172540             MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
(1)                                             ;/10 KHZ RATE,START THE CLOCK
(1)  032646  005737  003142          64$:       TST     DLYCNT          ;DELAY COUNT EXPIRED?
(1)  032652  001375                             BNE     64$             ;BRANCH IF TIME NOT ELAPSED
(1)  032654  005037  172540                     CLR     @#172540        ;STOP THE CLOCK
3299                                    .NLIST  ME
3300  032660  004737  016530                     JSR     PC,GSTATR       ;GET STATUS WITH RESET
3301  032664  033372                             65$
3302  032666  032737  000001  003044             BIT     #DRDYMSK,T.CS   ;CHECK IF DRIVE IS READY
3303  032674  001432                             BEQ     20$             ;BRANCH IF DRIVE IS NOT READY
3304
3305  032676  052737  000010  003004             BIS     #ULOAD,OPFLAG   ;SET UNLOAD OPERATION
3306  032704                                     PRINTF  #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 032704  005046                             CLR     -(SP)
 (13) 032706  153716  003033                     BISB    RLDRV+1,(SP)
 (12) 032712  012746  006051                     MOV     #DRVNAM,-(SP)
 (11) 032716  013746  003026                     MOV     RLBAS,-(SP)
 (10) 032722  012746  006040                     MOV     #BASADD,-(SP)
  (9) 032726  012746  010011                     MOV     #OPR1A,-(SP)
  (8) 032732  012746  007477                     MOV     #OPR3,-(SP)
  (7) 032736  012746  011316                     MOV     #FMTOP1,-(SP)
  (6) 032742  012746  000007                     MOV     #7,-(SP)
  (3) 032746  010600                             MOV     SP,R0
  (4) 032750  104417                             TRAP    C$PNTF
  (4) 032752  062706  000020                     ADD     #20,SP
3307                                                    ;PROMPT OPERATOR TO ''PRESS LOAD''
3308  032756  012703  000000                     MOV     #0,R3           ;SET ''LOAD CARTRIDGE'' STATE VALUE 0
3309
3310  032762  004737  016544          20$:       JSR     PC,GSTATC       ;GET STATUS
3311  032766  033372                             65$
3312  032770                                     BREAK                   ;MAKE A SUPERVISOR CALL
 (3)  032770  104422                             TRAP    C$BRK
3313  032772  022737  000000  003060             CMP     #0,T.STAT       ;TEST IF STATE 0
3314  033000  001370                             BNE     20$             ;WAIT FOR STATE 0
3315
3316  033002                             21$:    PRINTF  #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (13) 033002  005046                             CLR     -(SP)
 (13) 033004  153716  003033                     BISB    RLDRV+1,(SP)
 (12) 033010  012746  006051                     MOV     #DRVNAM,-(SP)
 (11) 033014  013746  003026                     MOV     RLBAS,-(SP)
 (10) 033020  012746  006040                     MOV     #BASADD,-(SP)
  (9) 033024  012746  010011                     MOV     #OPR1A,-(SP)
  (8) 033030  012746  007555                     MOV     #OPR6,-(SP)
  (7) 033034  012746  011316                     MOV     #FMTOP1,-(SP)
  (6) 033040  012746  000007                     MOV     #7,-(SP)
  (3) 033044  010600                             MOV     SP,R0
```

```
  (4)  033046  104417                        TRAP    C$PNTF
  (4)  033050  062706  000020                ADD     #20,SP
3317                                                    ;PROMPT OPERATOR TO 'PRESS LOAD &
3318                                                    ;/WAIT FOR READY''
3319  033054  005037  004362                CLR     OBUFF           ;CLEAR FOR RESPONSE
3320  033060                                GMANIL  OPR002,OBUFF,1,NO       ;PROMPT OPERATOR FOR RESPONSE
  (3)  033060  104443                        TRAP    C$GMAN
  (3)  033062  000404                        BR      10000$
  (4)  033064  004362                        .WORD   OBUFF
  (5)  033066  000120                        .WORD   T$CODE
  (5)  033070  007317                        .WORD   OPR002
  (5)  033072  000001                        .WORD   1
  (3)  033074                        10000$:
3321  033074  005737  004362                TST     OBUFF           ;TEST IF RESPONSE IS YES
3322  033100  001740                        BEQ     21$             ;BRANCH IF NOT READY
3323
3324  033102  004737  016544        22$:    JSR     PC,GSTATC       ;GET STATUS
3325  033106  033372                        65$
3326  033110                                BREAK                   ;MAKE A SUPERVISOR CALL
  (3)  033110  104422                        TRAP    C$BRK
3327  033112  022737  000005  003060        CMP     #5,T.STAT       ;CHECK IF STATE 5
3328  033120  001370                        BNE     22$             ;WAIT FOR STATE 5
3329
3330  033122  013701  003052                MOV     T.MP,R1         ;GET MP REG
3331  033126  032701  000020                BIT     #HOSTAT,R1      ;CHECK HEADS OUT
3332  033132  001003                        BNE     7$              ;YES-SKIP
3333  033134  012703  010475                MOV     #MHOSTA,R3      ;SET NAME MESSAGE PTR
3334  033140  000405                        BR      9$              ;GO REPORT
3335  033142  032701  000010        7$:     BIT     #BHSTAT,R1      ;CHECK BRUSH HOME SET
3336  033146  001010                        BNE     10$             ;YES-SKIP
3337  033150  012703  010451                MOV     #MBHSTA,R3      ;SET NAME MESSAGE PTR
3338  033154                        9$:     ERRHRD  702.,,ERR3      ;REPORT ERROR
  (4)  033154  104456                        TRAP    C$ERHRD
  (5)  033156  001276                        .WORD   702
  (5)  033160  000000                        .WORD   0
  (5)  033162  012454                        .WORD   ERR3
3339  033164                                EXIT    TST             ;EXIT
  (3)  033164  104432                        TRAP    C$EXIT
  (3)  033166  000204                        .WORD   L10035-.
3340  033170  005737  014202        10$:    TST     MISWIW          ;TEST IF MANUAL INTERVENTION RUN
3341  033174  100035                        BPL     16$             ;NO-SKIP
3342  033176  005737  003356                TST     PASNUM          ;CHECK IF FIRST PASS
3343  033202  001032                        BNE     16$             ;NO-SKIP
3344  033204  032701  000100                BIT     #HSSTAT,R1      ;ELSE CHECK HD 0 SELECTED
3345  033210  001412                        BEQ     13$             ;YES-SKIP
3346  033212  012703  010413                MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
3347  033216  012704  011266                MOV     #CCYLUP,R4      ;SET CONDITION POINTER
3348  033222                                ERRHRD  703.,,ERR4      ;REPORT ERROR
  (4)  033222  104456                        TRAP    C$ERHRD
  (5)  033224  001277                        .WORD   703
  (5)  033226  000000                        .WORD   0
  (5)  033230  012522                        .WORD   ERR4
3349  033232                                EXIT    TST             ;EXIT
  (3)  033232  104432                        TRAP    C$EXIT
  (3)  033234  000136                        .WORD   L10035-.
3350  033236  032701  001000        13$:    BIT     #VCSTAT,R1      ;CHECK VOL CHECK SET
```

```
3351  033242  001003                          BNE     15$             ;YES-SKIP
3352  033244  012703  010425                  MOV     #MVOLCK,R3      ;ELSE SET NAME MESSAGE PTR
3353  033250  000741                          BR      9$              ;GO REPORT
3354  033252  032737  040000  003044  15$:    BIT     #DRVERR,T.CS    ;TEST DRIVE ERROR SET
3355  033260  001003                          BNE     16$             ;YES-SKIP
3356  033262  012703  010402                  MOV     #MDRERR,R3      ;ELSE SET NAME MESSAGE PTR
3357  033266  000732                          BR      9$              ;GO REPORT
3358  033270  032701  020000          16$.    BIT     #WLSTAT,R1      ;CHECK WRITE LOCK STATUS
3359  033274  001406                          BEQ     17$             ;SKIP IF RESET
3360  033276  012703  010464                  MOV     #MWLSTA,R3      ;ELSE SET NAME MESSAGE PTR
3361  033302                                  ERRHRD  705,,ERR2
 (4)  033302  104456                          TRAP    C$ERHRD
 (5)  033304  001301                          .WORD   705
 (5)  033306  000000                          .WORD   0
 (5)  033310  012406                          .WORD   ERR2
3362  033312  042701  021177          17$:    BIC     #21177,R1       ;CLEAR STAUS EXCEPT FOR ERROR BITS
3363  033316  023727  002276  000001          CMP     T.DRIVE,#1
3364  033324  001404                          BEQ     99$
3365  033326  022701  000200                  CMP     #200,R1
3366  033332  001411                          BEQ     19$
3367  033334  000402                          BR      18$
3368  033336  005701                  99$:    TST     R1
3369  033340  001406                          BEQ     19$             ;NO-SKIP
3370  033342                          18$:    ERRHRD  704,,ERR6       ;ELSE REPORT ALL ERRORS
 (4)  033342  104456                          TRAP    C$ERHRD
 (5)  033344  001300                          .WORD   704
 (5)  033346  000000                          .WORD   0
 (5)  033350  012642                          .WORD   ERR6
3371  033352                                  EXIT    TST             ;EXIT
 (3)  033352  104432                          TRAP    C$EXIT
 (3)  033354  000016                          .WORD   L10035-.
3372  033356  013701  003044          19$:    MOV     T.CS,R1         ;GET CS REG
3373  033362  042701  141777                  BIC     #141777,R1      ;CLEAR ALL BUT ERROR BITS
3374  033366  005701                          TST     R1              ;TEST IF ANY ERROR SET
3375  033370  001364                          BNE     18$             ;YES-SKIP TO REPORT
3376  033372                          25$:
3377  033372                          65$:
3378  033372                          ENDTST
 (3)  033372                          L10035:
 (3)  033372  104401                          TRAP    C$ETST
3379
3380
3381
3382                                  .SBTTL  *TEST 8         INITIAL RESET STATE
3383  033374                          BGNTST                  ;TEST 8
 (3)  033374                                                                          T8::
3384  033374  012737  006513  003012          MOV     #INITST,ERHEAD
3385  033402  004737  016512                  JSR     PC,TSTINT       ;INITIALIZE TEST
3386
3387  033406  004737  016530                  JSR     PC,GSTATR       ;GET STATUS WITH RESET
3388  033412  033460                          65$
3389  033414  005737  014202                  TST     MISWIW          ;CHECK IF MAN INTERVENTION WAS RUN
3390  033420  100017                          BPL     4$              ;NO-SKIP
3391  033422  005737  003356                  TST     PASNUM          ;CHECK IF 1ST PASS
3392  033426  001014                          BNE     4$              ;NO-SKIP
3393  033430  032737  000100  003052          BIT     #HSSTAT,T.MP    ;CHECK HD SELECT STILL 0
```

```
                                        BEQ     4$                  ;YES-SKIP
                                        MOV     #MHSTA,R3           ;SET NAME MESSAGE PTR
                                        MOV     #CCYLUP,R4          ;SET CONDITION POINTER
                                        ERRHRD  801,,ERR4           ;REPORT ERROR
                                        TRAP    C$ERHRD
                                        .WORD   801
                                        .WORD   0
                                        .WORD   ERR4
                                4$:
                                65$:
                                ENDTST
                                L10036:
                                        TRAP    C$ETST


                                .SBTTL  *TEST 9         DRIVE READY
                                BGNTST                  ;TEST 9
                                                                        T9::
                                ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
                                        TST     CLKFLG              ;P-CLOCK?
                                        BNE     1$                  ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
                                        MOV     #NOTST,R2           ;INITIALIZE POINTER FOR TEST MSG.
                                        MOVB    #'0,4(R2)           ;INSERT TEST NUMBER INTO MSG.
                                        MOVB    #'9,5(R2)           ;INSERT TEST NUMBER INTO MSG.
                                        PRINTF  #FMT9,#NOTST        ;PRINT MSG. 'TST 9 CANNOT BE PERFORMED...
                                        MOV     #NOTST,-(SP)
                                        MOV     #FMT9,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTF
                                        ADD     #6,SP
                                                                    ;/NO P-CLK''
                                EXIT    TST
                                        TRAP    C$EXIT
                                        .WORD   L10037-.
                                1$:     MOV     #T09ERR,ERHEAD      ;SET ERROR HEADER
                                        MOV     #NEWCYL,R1          ;GET POINTER TO DESIRED LOC
                                        CLR     (R1)+               ;CLEAR NEW CYL
                                        CLR     (R1)+               ;CLEAR CURRENT CYL
                                        CLR     (R1)+               ;       DIFFERENCE
                                        CLR     (R1)                ;       SIGN
                                        JSR     PC,TSTINT           ;INITIALIZE TEST
                                        JSR     PC,GSTATR           ;GET STATUS WITH RESET
                                        65$
                                        JSR     PC,POSHSB           ;POSITION HEAD SELECTED BIT
                                        MOV     R5,DESHD            ;STORE AS DESIRED HEAD
                                        JSR     PC,SIMSEK           ;EXECUTE SIMPLE SEEK
                                        65$
                                        MOV     #MDRDY,R3           ;SET NAME MESSAGE PTR
                                        MOV     #CDRDY,R4           ;SET CONDITION POINTER
                                        JSR     PC,GSTAT            ;GET STATUS
                                        65$
                                        BIT     #DRDYMSK,T.CS       ;TEST READY SET
                                        BEQ     4$                  ;NO-SKIP
                                        ERRHRD  901,,ERR4           ;REPORT READY ERROR
```

```
   (4)  033640  104456                        TRAP    C$ERHRD
   (5)  033642  001605                        .WORD   901
   (5)  033644  000000                        .WORD   0
   (5)  033646  012522                        .WORD   ERR4
 3435   033650                        EXIT    TST             ;EXIT
   (3)  033650  104432                        TRAP    C$EXIT
   (3)  033652  000254                        .WORD   L10037-.
 3436   033654  012701  000121        4$:     MOV     #81.,R1         ;SET WAIT COUNT
 3437   033660  004737  016560        5$:     JSR     PC,GSTAT        ;GET STATUS
 3438   033664  034126                        65$
 3439   033666  012703  000005                MOV     #5,R3           ;SET EXPECTED STATE VALUE
 3440   033672  023703  003060                CMP     T.STAT,R3       ;CHECK STATE IS 5
 3441   033676  001406                        BEQ     7$              ;YES-SKIP
 3442   033700                                ERRHRD  902.,,ERR7      ;ELSE REPORT
   (4)  033700  104456                        TRAP    C$ERHRD
   (5)  033702  001606                        .WORD   902
   (5)  033704  000000                        .WORD   0
   (5)  033706  013542                        .WORD   ERR7
 3443   033710                        EXIT    TST
   (3)  033710  104432                        TRAP    C$EXIT
   (3)  033712  000214                        .WORD   L10037-.
 3444   033714  012703  010260        7$:     MOV     #MDRDY,R3
 3445   033720  032737  000001  003044        BIT     #DRDYMSK,T.CS   ;CHECK READY SET
 3446   033726  001042                        BNE     12$             ;YES-SKIP
 3447   033730  005301                        DEC     R1              ;ELSE DEC WAIT COUNT
 3448   033732  001432                        BEQ     9$              ;SKIP IF 0
 3449                          .LIST   ME
 3450   033734                        TIMDLY  #1
   (1)  033734                        SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
   (2)  033734                MSPUT   #3,#104,#CLKINT,#340
   (3)  033734                MSPUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
   (4)  033734                MSPUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
   (5)  033734                MSPUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
   (6)  033734                MSPUT1  #340
   (7)  033734                MSGNINS <MOV    #340,-(SP)>
   (8)  033734  012746  000340           MOV     #340,-(SP)
   (8)                        .MEXIT
   (5)  033740                MSPUT1  #CLKINT
   (6)  033740                MSGNINS <MOV    #CLKINT,-(SP)>
   (7)  033740  012746  016112           MOV     #CLKINT,-(SP)
   (7)                        .MEXIT
   (4)  033744                MSPUT1  #104
   (5)  033744                MSGNINS <MOV    #104,-(SP)>
   (6)  033744  012746  000104           MOV     #104,-(SP)
   (6)                        .MEXIT
   (3)  033750                MSPUT1  #3
   (4)  033750                MSGNINS <MOV    #3,-(SP)>
   (5)  033750  012746  000003           MOV     #3,-(SP)
   (5)                        .MEXIT
   (2)  033754                MSSVC   C$SVEC
   (3)  033754                MSTSTLAB
   (4)                         .MEXIT
   (3)  033754                MSGNINS <TRAP   C$SVEC>
   (4)  033754  104437                   TRAP    C$SVEC
   (4)                        .MEXIT
   (2)  033756                MSGNINS <ADD    #10,SP>
```

```
   (3)  033756  062706  000010                    ADD     #10,SP
   (3)                                .MEXIT
   (1)  033762  012737  000001  003142            MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
   (1)  033770  012737  000001  172542            MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                    ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  033776  012737  000113  172540            MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                    ;/10 KHZ RATE,START THE CLOCK
   (1)  034004  005737  003142          64$:      TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  034010  001375                            BNE     64$             ;BRANCH IF TIME NOT ELAPSED
   (1)  034012  005037  172540                    CLR     @#172540        ;STOP THE CLOCK
3451                                  .NLIST    ME
3452  034016  000720                            BR      5$
3453  034020                          9$:       ERRHRD  903,,ERR5       ;REPORT READY ERROR
   (4)  034020  104456                          TRAP    C$ERHRD
   (5)  034022  001607                          .WORD   903
   (5)  034024  000000                          .WORD   0
   (5)  034026  012572                          .WORD   ERR5
3454  034030                                    EXIT    TST
   (3)  034030  104432                          TRAP    C$EXIT
   (3)  034032  000074                          .WORD   L10037-.
3455
3456  034034  005737  003044          12$:      TST     T.CS            ;TEST IF ANY ERROR
3457  034040  100006                            BPL     15$             ;NO-SKIP
3458  034042                                    ERRHRD  904,,ERR6
   (4)  034042  104456                          TRAP    C$ERHRD
   (5)  034044  001610                          .WORD   904
   (5)  034046  000000                          .WORD   0
   (5)  034050  012642                          .WORD   ERR6
3459  034052                                    EXIT    TST
   (3)  034052  104432                          TRAP    C$EXIT
   (3)  034054  000052                          .WORD   L10037-.
3460  034056  012703  010413          15$:      MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
3461  034062  004737  022266                    JSR     PC,POSHSB       ;POSITION HEAD SELECT BIT FOR TEST
3462  034066  020537  003112                    CMP     R5,DESHD        ;CHECK IF CORRECT HEAD SELECTED
3463  034072  001415                            BEQ     20$             ;YES-SKIP
3464  034074  005737  003112                    TST     DESHD           ;ELSE TEST IF 1 DESIRED
3465  034100  001406                            BEQ     17$             ;NO-REPORT SB 0
3466  034102                                    ERRHRD  905,,ERR3       ;ELSE REPORT SB 1
   (4)  034102  104456                          TRAP    C$ERHRD
   (5)  034104  001611                          .WORD   905
   (5)  034106  000000                          .WORD   0
   (5)  034110  012454                          .WORD   ERR3
3467  034112                                    EXIT    TST
   (3)  034112  104432                          TRAP    C$EXIT
   (3)  034114  000012                          .WORD   L10037-.
3468  034116                          17$:      ERRHRD  906,,ERR2
   (4)  034116  104456                          TRAP    C$ERHRD
   (5)  034120  001612                          .WORD   906
   (5)  034122  000000                          .WORD   0
   (5)  034124  012406                          .WORD   ERR2
3469  034126                          20$:
3470  034126                          65$:
3471  034126                          ENDTST
   (3)  034126                          L10037:
   (3)  034126  104401                          TRAP    C$ETST
3472
```

```
3473
3474
3475                                        .SBTTL  *TEST 10         SEEK SIGN SWITCH
3476  034130                                BGNTST                   ;TEST 10
 (3)  034130                                                                        T10::
3477                                        ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3478  034130  005737  003144                        TST     CLKFLG           ;P-CLOCK?
3479  034134  001024                              BNE     1$               ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3480  034136  012702  006662                      MOV     #NOTST,R2        ;INITIALIZE POINTER FOR TEST MSG.
3481  034142  112762  000061  000004              MOVB    #'1,4(R2)        ;INSERT TEST NUMBER INTO MSG.
3482  034150  112762  000060  000005              MOVB    #'0,5(R2)        ;INSERT TEST NUMBER INTO MSG.
3483  034156                                      PRINTF  #FMT9,#NOTST     ;PRINT MSG. 'TST 10 CANNOT BE PERFORMED...
 (8)  034156  012746  006662                      MOV     #NOTST,-(SP)
 (7)  034162  012746  011627                      MOV     #FMT9,-(SP)
 (6)  034166  012746  000002                      MOV     #2,-(SP)
 (3)  034172  010600                              MOV     SP,R0
 (4)  034174  104417                              TRAP    C$PNTF
 (4)  034176  062706  000006                      ADD     #6,SP
3484                                                                    ;/NO P-CLK''
3485  034202                                      EXIT    TST
 (3)  034202  104432                              TRAP    C$EXIT
 (3)  034204  000412                              .WORD   L10040-.
3486  034206  012737  006551  003012  1$:         MOV     #T10ERR,ERHEAD   ;SET ERROR HEADER
3487  034214  012701  003102                      MOV     #NEWCYL,R1
3488  034220  005021                              CLR     (R1)+            ;CLEAR NEW CYL
3489  034222  005021                              CLR     (R1)+            ;CLEAR CURRENT CYLINDER
3490  034224  005021                              CLR     (R1)+            ;CLEAR DIFFERENCE
3491  034226  052721  000001                      BIS     #BIT0,(R1)+      ;SET FOR SIGN OF 1
3492  034232  004737  022266                      JSR     PC,POSHSB        ;GET SELECTED HEAD
3493  034236  010521                              MOV     R5,(R1)+         ;SET AS DESIRED HEAD
3494  034240                              T104$:
3495  034240                              BGNSUB
 (3)  034240                                                                        T10.1:
 (3)  034240  104402                              TRAP    C$BSUB
3496  034242  004737  016512                      JSR     PC,TSTINT        ;INITIALIZE TEST
3497  034246  004737  016530                      JSR     PC,GSTATR        ;GET STATUS
3498  034252  034576                              60$
3499  034254  004737  020622                      JSR     PC,SIMSEK        ;DO SEEK
3500  034260  034576                              60$
3501  034262  012703  010260                      MOV     #MDRDY,R3        ;SET NAME MESSAGE PTR
3502  034266  012704  011227                      MOV     #CDRDY,R4        ;SET CONDITION MESSAGE PTR
3503  034272  004737  016560                      JSR     PC,GSTAT         ;GET STATUS
3504  034276  034576                              60$
3505  034300  032737  000001  003044              BIT     #DRDYMSK,T.CS    ;CHECK READY RESET
3506  034306  001406                              BEQ     4$               ;YES-SKIP
3507  034310                                      ERRHRD  1001.,,ERR4      ;REPORT READY ERROR
 (4)  034310  104456                              TRAP    C$ERHRD
 (5)  034312  001751                              .WORD   1001
 (5)  034314  000000                              .WORD   0
 (5)  034316  012522                              .WORD   ERR4
3508  034320                                      EXIT    SUB              ;EXIT SUBTEST
 (3)  034320  104432                              TRAP    C$EXIT
 (3)  034322  000254                              .WORD   L10041-.
3509
3510
3511  034324  012701  000121          4$:         MOV     #81.,R1          ;SET WAIT COUNT
```

```
3512  034330  004737  016560          5$:     JSR     PC,GSTAT        ;GET STATUS
3513  034334  034576                          60$
3514  034336  012703  000005                  MOV     #5,R3           ;SET EXPECTED STATE
3515  034342  020337  003060                  CMP     R3,T.STAT       ;CHECK STATE IS 5
3516  034346  001406                          BEQ     7$              ;YES-SKIP
3517  034350                                  ERRHRD  1002.,,ERR7     ;REPORT STATE ERROR
 (4)  034350  104456                          TRAP    C$ERHRD
 (5)  034352  001752                          .WORD   1002
 (5)  034354  000000                          .WORD   0
 (5)  034356  013542                          .WORD   ERR7
3518  034360                                  EXIT    SUB             ;EXIT
 (3)  034360  104432                          TRAP    C$EXIT
 (3)  034362  000214                          .WORD   L10041-.
3519  034364  012703  010260          7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
3520  034370  032737  000001  003044          BIT     #DRDYMSK,T.CS   ;CHECK READY SET
3521  034376  001042                          BNE     12$             ;YES-SKIP
3522  034400  005301                          DEC     R1              ;DO WAIT COUNT
3523  034402  001432                          BEQ     9$              ;SKIP IF 0
3524                                  .LIST   ME
3525  034404                                  TIMDLY  #1
 (1)  034404                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
 (2)  034404                          M$PUT   #3,#104,#CLKINT,#340
 (3)  034404                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
 (4)  034404                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
 (5)  034404                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
 (6)  034404                          M$PUT1  #340
 (7)  034404                          M$GNINS <MOV    #340,-(SP)>
 (8)  034404  012746  000340                  MOV     #340,-(SP)
 (8)                                  .MEXIT
 (5)  034410                          M$PUT1  #CLKINT
 (6)  034410                          M$GNINS <MOV    #CLKINT,-(SP)>
 (7)  034410  012746  016112                  MOV     #CLKINT,-(SP)
 (7)                                  .MEXIT
 (4)  034414                          M$PUT1  #104
 (5)  034414                          M$GNINS <MOV    #104,-(SP)>
 (6)  034414  012746  000104                  MOV     #104,-(SP)
 (6)                                  .MEXIT
 (3)  034420                          M$PUT1  #3
 (4)  034420                          M$GNINS <MOV    #3,-(SP)>
 (5)  034420  012746  000003                  MOV     #3,-(SP)
 (5)                                  .MEXIT
 (2)  034424                          M$SVC   C$SVEC
 (3)  034424                          M$TSTLAB
 (4)                                   .MEXIT
 (3)  034424                          M$GNINS <TRAP   C$SVEC>
 (4)  034424  104437                          TRAP    C$SVEC
 (4)                                  .MEXIT
 (2)  034426                          M$GNINS <ADD    #10,SP>
 (3)  034426  062706  000010                  ADD     #10,SP
 (3)                                  .MEXIT
 (1)  034432  012737  000001  003142          MOV     ##1,DLYCNT      ;INITIALIZE DELAY COUNT
 (1)  034440  012737  000001  172542          MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
 (1)                                                                  ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
 (1)  034446  012737  000113  172540          MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
 (1)                                                                  ;/10 KHZ RATE,START THE CLOCK
 (1)  034454  005737  003142          64$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
```

```
  (1) 034460 001375                    BNE     64$             ;BRANCH IF TIME NOT ELAPSED
  (1) 034462 005037 172540             CLR     @#172540        ;STOP THE CLOCK
3526                                .NLIST  ME
3527 034466 000720                   BR      5$
3528
3529 034470                          9$:     ERRHRD  1003.,,ERR5     ;REPORT READY ERROR
  (4) 034470 104456                          TRAP    C$ERHRD
  (5) 034472 001753                          .WORD   1003
  (5) 034474 000000                          .WORD   0
  (5) 034476 012572                          .WORD   ERR5
3530 034500                          EXIT    SUB             ;EXIT
  (3) 034500 104432                          TRAP    C$EXIT
  (3) 034502 000074                          .WORD   L10041-.
3531 034504 005737 003044            12$:    TST     T.CS            ;TEST IF ANY OTHER ERROR
3532 034510 100006                           BPL     15$             ;NO-SKIP
3533 034512                                  ERRHRD  1004.,,ERR6     ;REPORT ALL ERRORS
  (4) 034512 104456                          TRAP    C$ERHRD
  (5) 034514 001754                          .WORD   1004
  (5) 034516 000000                          .WORD   0
  (5) 034520 012642                          .WORD   ERR6
3534 034522                          EXIT    SUB             ;EXIT
  (3) 034522 104432                          TRAP    C$EXIT
  (3) 034524 000052                          .WORD   L10041-.
3535
3536 034526 012703 010413            15$:    MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
3537 034532 004737 022266                    JSR     PC,POSHSB       ;GET SELECTED HEAD BIT
3538 034536 020537 003112                    CMP     R5,DESHD        ;CHECK IF CORRECT
3539 034542 001415                           BEQ     20$             ;YES - SKIP
3540 034544 005737 003112                    TST     DESHD           ;WAS IT SET
3541 034550 001406                           BEQ     17$             ;NO-SKIP
3542 034552                                  ERRHRD  1005.,,ERR3     ;REPORT SB 1
  (4) 034552 104456                          TRAP    C$ERHRD
  (5) 034554 001755                          .WORD   1005
  (5) 034556 000000                          .WORD   0
  (5) 034560 012454                          .WORD   ERR3
3543 034562                          EXIT    SUB
  (3) 034562 104432                          TRAP    C$EXIT
  (3) 034564 000012                          .WORD   L10041-.
3544 034566                          17$:    ERRHRD  1006.,,ERR2     ;REPORT SB 0
  (4) 034566 104456                          TRAP    C$ERHRD
  (5) 034570 001756                          .WORD   1006
  (5) 034572 000000                          .WORD   0
  (5) 034574 012406                          .WORD   ERR2
3545
3546 034576                          20$:
3547 034576                          60$:
3548 034576                          ENDSUB
  (3) 034576                         L10041:
  (3) 034576 104403                          TRAP    C$ESUB
3549 034600 005737 003110                    TST     DESSGN          ;CHECK IF BOTH SIGN USED
3550 034604 001404                           BEQ     25$             ;YES-SKIP
3551 034606 005037 003110                    CLR     DESSGN          ;SET FOR SIGN OF 0
3552 034612 000137 034240                    JMP     T104$           ;DO TEST AGAIN
3553 034616                          25$:
3554 034616                          ENDTST
  (3) 034616                         L10040:
```

```
 (3)  034616  104401                          TRAP    C$ETST
3555
3556
3557
3558                                 .SBTTL  *TEST 11        HEAD ALIGNMENT SUPPORT
3559  034620                         BGNTST                  ;TEST 11
 (3)  034620                                                                  T11::
3560  034620  032737  000010  014202         BIT     #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
3561  034626  001411                         BEQ     1$              ;NO-EXIT
3562  034630  005737  003356                 TST     PASNUM          ;TEST IF PASS 0
3563  034634  001006                         BNE     1$              ;NO-EXIT
3564  034636  023737  003032  003010         CMP     RLDRV,HADONE    ;TEST IF HEAD ALIGN DONE THIS DRIVE
3565  034644  001004                         BNE     2$              ;NO - SKIP
3566  034646  000137  035270                 JMP     T115$           ;GO CHECK WRITE LOCK
3567  034652                         1$:     EXIT    TST
 (3)  034652  104432                         TRAP    C$EXIT
 (3)  034654  000514                         .WORD   L10042-.
3568  034656  013737  003032  003010 2$:     MOV     RLDRV,HADONE    ;SET HEAD ALIGN DONE FLAG
3569  034664                               PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
 (11) 034664  005046                         CLR     -(SP)
 (11) 034666  153716  003033                 BISB    RLDRV+1,(SP)
 (10) 034672  012746  006051                 MOV     #DRVNAM,-(SP)
 (9)  034676  013746  003026                 MOV     RLBAS,-(SP)
 (8)  034702  012746  006040                 MOV     #BASADD,-(SP)
 (7)  034706  012746  011443                 MOV     #FMT5,-(SP)
 (6)  034712  012746  000005                 MOV     #5,-(SP)
 (3)  034716  010600                         MOV     SP,R0
 (4)  034720  104417                         TRAP    C$PNTF
 (4)  034722  062706  000014                 ADD     #14,SP
3570  034726                               PRINTF  #FMT9,#HAMES1   ;TYPE INSTRUCTIONS
 (8)  034726  012746  007130                 MOV     #HAMES1,-(SP)
 (7)  034732  012746  011627                 MOV     #FMT9,-(SP)
 (6)  034736  012746  000002                 MOV     #2,-(SP)
 (3)  034742  010600                         MOV     SP,R0
 (4)  034744  104417                         TRAP    C$PNTF
 (4)  034746  062706  000006                 ADD     #6,SP
3571  034752                               PRINTF  #FMT9,#HAMES2
 (8)  034752  012746  007213                 MOV     #HAMES2,-(SP)
 (7)  034756  012746  011627                 MOV     #FMT9,-(SP)
 (6)  034762  012746  000002                 MOV     #2,-(SP)
 (3)  034766  010600                         MOV     SP,R0
 (4)  034770  104417                         TRAP    C$PNTF
 (4)  034772  062706  000006                 ADD     #6,SP
3572
3573  034776                         BGNSUB
 (3)  034776                                                                  T11.1:
 (3)  034776  104402                         TRAP    C$BSUB
3574  035000  004737  016512         3$:     JSR     PC,TSTINT       ;INITIALIZE TEST
3575  035004  005037  003006                 CLR     DONE            ;CLEAR DONE
3576  035010  013737  003032  003034         MOV     RLDRV,L.CS      ;SET UP FOR GET STATUS
3577  035016  052737  000104  003034         BIS     #GTSTAT,L.CS
3578  035024  012737  000013  003040         MOV     #GETSTAT!DRSET,L.DA
3579  035032  013762  003040  000004         MOV     L.DA,RLDA(R2)
3580  035040  013762  003034  000000         MOV     L.CS,RLCSR(R2)  ;DO GET STATUS
3581  035046  012737  000031  003142         MOV     #25.,DLYCNT     ;INITIALIZE DELAY COUNT
3582  035054  006337  003142                 ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
```

```
3583  035060  006337  003142              ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
3584  035064                      4$:     DELAY   #250.           ;IMPLEMENT TIME DELAY
 (2)  035064  012727  000372              MOV     ##250.,(PC)+
 (2)  035070  000000                      .WORD   0
 (2)  035072  013727  002116              MOV     L$DLY,(PC)+
 (2)  035076  000000                      .WORD   0
 (2)  035100  005367  177772              DEC     -6(PC)
 (2)  035104  001375                      BNE     .-4
 (2)  035106  005367  177756              DEC     -22(PC)
 (2)  035112  001367                      BNE     .-20
3585  035114  005337  003142              DEC     DLYCNT          ;DECREMENT DELAY COUNT
3586  035120                              BREAK                   ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
 (3)  035120  104422                      TRAP    C$BRK
3587                                                            ;/BACK TO SUPERVISOR COMMAND MODE
3588  035122  001360                      BNE     4$              ;BRANCH IF TIME DELAY NOT EXPIRED
3589  035124  005737  003006              TST     DONE            ;CHECK IF DONE
3590  035130  001723                      BEQ     3$              ;NO-GO CLR CONTROLLER
3591
3592
3593  035132  012737  000021  003040 10$:     MOV     #HDSEL.MBSETO,L.DA;LOAD FOR HEAD 1
3594  035140  032737  020000  003052          BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK SET
3595  035146  001003                      BNE     12$             ;YES-SKIP
3596  035150  042737  000020  003040      BIC     #HDSEL,L.DA     ;ELSE CLEAR TO HEAD 0
3597  035156  013737  003032  003C34 12$:     MOV     RLDRV,L.CS      ;LOAD IN DRIVE NUMBER
3598  035164  052737  000106  003034          BIS     #SEEK,L.CS      ;SET FOR SEEK
3599  035172  013762  003040  000004          MOV     L.DA,RLDA(R2)   ;LOAD & EXECUTE SEEK
3600  035200  013762  003034  000000          MOV     L.CS,RLCSR(R2)
3601  035206  012737  000017  003142          MOV     #15.,DLYCNT     ;INITIALIZE DELAY COUNT
3602  035214  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2
3603  035220  006337  003142                  ASL     DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
3604  035224                      5$:     DELAY   #250.           ;IMPLEMENT TIME DELAY
 (2)  035224  012727  000372              MOV     ##250.,(PC)+
 (2)  035230  000000                      .WORD   0
 (2)  035232  013727  002116              MOV     L$DLY,(PC)+
 (2)  035236  000000                      .WORD   0
 (2)  035240  005367  177772              DEC     -6(PC)
 (2)  035244  001375                      BNE     .-4
 (2)  035246  005367  177756              DEC     -22(PC)
 (2)  035252  001367                      BNE     .-20
3605  035254  005337  003142              DEC     DLYCNT          ;DECREMENT DELAY COUNT
3606  035260                              BREAK                   ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
 (3)  035260  104422                      TRAP    C$BRK
3607                                                            ;/BACK TO SUPERVISOR COMMAND MODE
3608  035262  001360                      BNE     5$              ;BRANCH IF TIME DELAY NOT EXPIRED
3609  035264  000645                      BR      3$              ;LOOP
3610  035266                      59$:
3611  035266                      ENDSUB
 (3)  035266                      L10043:
 (3)  035266  104403                      TRAP    C$ESUB
3612  035270                      T115$:
3613  035270                      BGNSUB
 (3)  035270                                                                    T11.2:
 (3)  035270  104402                      TRAP    C$BSUB
3614  035272  004737  016512              JSR     PC,TSTINT       ;INITIALIZE TEST
3615  035276  004737  016530              JSR     PC,GSTATR       ;CLEAR DRIVE
3616  035302  035366                      60$
```

```
 3617  035304  032737  020000  003052          BIT     #WLSTAT,T.MP    ;CHECK WRITE LOCK RESET
 3618  035312  001425                          BEQ     19$             ;YES-SKIP
 3619  035314                          18$:    PRINTF  #FMT9,#OPR12    ;REQUEST WRITE LOCK RESET
  (8)  035314  012746  007772                  MOV     #OPR12,-(SP)
  (7)  035320  012746  011627                  MOV     #FMT9,-(SP)
  (6)  035324  012746  000002                  MOV     #2,-(SP)
  (3)  035330  010600                          MOV     SP,R0
  (4)  035332  104417                          TRAP    C$PNTF
  (4)  035334  062706  000006                  ADD     #6,SP
 3620  035340  005037  004362                  CLR     OBUFF           ;CLEAR FOR RESPONSE
 3621  035344                                  GMANIL  OPR002,OBUFF,1,NO       ;GET RESPONSE
  (3)  035344  104443                          TRAP    C$GMAN
  (3)  035346  000404                          BR      10000$
  (4)  035350  004362                          .WORD   OBUFF
  (5)  035352  000120                          .WORD   T$CODE
  (5)  035354  007317                          .WORD   OPR002
  (5)  035356  000001                          .WORD   1
  (3)  035360                          10000$:
 3622  035360  005737  004362                  TST     OBUFF           ;WAS ANSWER YES
 3623  035364  001753                          BEQ     18$             ;NO-REPEAT REQUEST
 3624  035366                          19$:
 3625  035366                          60$:
 3626  035366                          ENDSUB
  (3)  035366                          L10044:
  (3)  035366  104403                          TRAP    C$ESUB
 3627  035370                          20$:
 3628  035370                          ENDTST
  (3)  035370                          L10042:
  (3)  035370  104401                          TRAP    C$ETST
 3629
 3630
 3631
 3632                                  .SBTTL  *TEST 12        HEAD SWITCHING
 3633  035372                          BGNTST                  ;TEST 12
  (3)  035372                                                                     T12::
 3634                                  ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
 3635  035372  005737  003144                  TST     CLKFLG          ;P-CLOCK?
 3636  035376  001024                          BNE     1$              ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
 3637  035400  012702  006662                  MOV     #NOTST,R2       ;INITIALIZE POINTER FOR TEST MSG.
 3638  035404  112762  000061  000004          MOVB    #'1,4(R2)       ;INSERT TEST NUMBER INTO MSG.
 3639  035412  112762  000062  000005          MOVB    #'2,5(R2)       ;INSERT TEST NUMBER INTO MSG.
 3640  035420                                  PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 12 CANNOT BE PERFORMED...
  (8)  035420  012746  006662                  MOV     #NOTST,-(SP)
  (7)  035424  012746  011627                  MOV     #FMT9,-(SP)
  (6)  035430  012746  000002                  MOV     #2,-(SP)
  (3)  035434  010600                          MOV     SP,R0
  (4)  035436  104417                          TRAP    C$PNTF
  (4)  035440  062706  000006                  ADD     #6,SP
 3641                                                                     ;/NO P-CLK''
 3642  035444                                  EXIT TST
  (3)  035444  104432                          TRAP    C$EXIT
  (3)  035446  000406                          .WORD   L10045-.
 3643  035450  012737  006571  003012  1$:     MOV     #T12ERR,ERHEAD  ;SET ERROR HEADER
 3644  035456  012701  003102                  MOV     #NEWCYL,R1      ;GET POINTER TO DESIRED LOCATION
 3645  035462  005021                          CLR     (R1)+           ;CLEAR NEW CYLINDER
 3646  035464  005021                          CLR     (R1)+           ;CLEAR CURRENT CYL.
```

```
 3647  035466  005021                          CLR     (R1)+           ;CLEAR DIFFERENCE
 3648  035470  005021                          CLR     (R1)+           ;CLEAR SIGN
 3649  035472  012721  000001                  MOV     #1,(R1)+        ;SET FOR HEAD 1
 3650  035476                        T124$:
 3651  035476                        BGNSUB
  (3)  035476                                                                  T12.1:
  (3)  035476  104402                          TRAP    C$BSUB
 3652  035500  004737  016512                  JSR     PC,TSTINT       ;INITIALIZE TEST
 3653  035504  004737  016530                  JSR     PC,GSTATR       ;GET STATUS WITH RESET
 3654  035510  036034                          60$
 3655  035512  004737  020622                  JSR     PC,SIMSEK       ;DO SEEK
 3656  035516  036034                          60$
 3657  035520  012703  010260                  MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 3658  035524  012704  011227                  MOV     #CDRDY,R4       ;SET CONDITION POINTER
 3659  035530  004737  016560                  JSR     PC,GSTAT        ;GET STATUS
 3660  035534  036034                          60$
 3661  035536  032737  000001  003044          BIT     #DRDYMSK,T.CS   ;CHECK IF READY
 3662  035544  001406                          BEQ     5$              ;NO-SKIP
 3663  035546                                  ERRHRD  1201,,ERR4      ;REPORT READY ERROR
  (4)  035546  104456                          TRAP    C$ERHRD
  (5)  035550  002261                          .WORD   1201
  (5)  035552  000000                          .WORD   0
  (5)  035554  012522                          .WORD   ERR4
 3664  035556                                  EXIT    SUB             ;EXIT
  (3)  035556  104432                          TRAP    C$EXIT
  (3)  035560  000254                          .WORD   L10046-.
 3665
 3666  035562  012701  000121          5$:     MOV     #81.,R1         ;SET WAIT COUNT
 3667  035566  004737  016560          6$:     JSR     PC,GSTAT        ;GET STATUS
 3668  035572  036034                          60$
 3669  035574  012703  000005                  MOV     #5,R3           ;SET EXPECTED STATE VALUE
 3670  035600  020337  003060                  CMP     R3,T.STAT       ;CHECK IF STATE IS 5
 3671  035604  001406                          BEQ     7$              ;YES-SKIP
 3672  035606                                  ERRHRD  1202,,ERR7      ;REPORT STATE ERROR
  (4)  035606  104456                          TRAP    C$ERHRD
  (5)  035610  002262                          .WORD   1202
  (5)  035612  000000                          .WORD   0
  (5)  035614  013542                          .WORD   ERR7
 3673  035616                                  EXIT    SUB
  (3)  035616  104432                          TRAP    C$EXIT
  (3)  035620  000214                          .WORD   L10046-.
 3674
 3675  035622  012703  010260          7$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
 3676  035626  032737  000001  003044          BIT     #DRDYMSK,T.CS   ;CHECK DRIVE READY
 3677  035634  001042                          BNE     12$             ;YES-SKIP
 3678  035636  005301                          DEC     R1              ;DEC WAIT COUNT
 3679  035640  001432                          BEQ     9$              ;SKIP IF 0
 3680                                  .LIST   ME
 3681  035642                                  TIMDLY  #1
  (1)  035642                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
  (2)  035642                          M$PUT   #3,#104,#CLKINT,#340
  (3)  035642                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
  (4)  035642                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
  (5)  035642                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
  (6)  035642                          M$PUT1  #340
  (7)  035642                          M$GNINS <MOV    #340,-(SP)>
```

```
    (8)  035642  012746  000340          MOV     #340,-(SP)
    (8)                          .MEXIT
    (5)  035646                  M$PUT1  #CLKINT
    (6)  035646                  M$GNINS <MOV    #CLKINT,-(SP)>
    (7)  035646  012746  016112          MOV     #CLKINT,-(SP)
    (7)                          .MEXIT
    (4)  035652                  M$PUT1  #104
    (5)  035652                  M$GNINS <MOV    #104,-(SP)>
    (6)  035652  012746  000104          MOV     #104,-(SP)
    (6)                          .MEXIT
    (3)  035656                  M$PUT1  #3
    (4)  035656                  M$GNINS <MOV    #3,-(SP)>
    (5)  035656  012746  000003          MOV     #3,-(SP)
    (5)                          .MEXIT
    (2)  035662                  M$SVC   C$SVEC
    (3)  035662                  M$TSTLAB
    (4)                           .MEXIT
    (3)  035662                  M$GNINS <TRAP   C$SVEC>
    (4)  035662  104437                  TRAP    C$SVEC
    (4)                          .MEXIT
    (2)  035664  .               M$GNINS <ADD    #10,SP>
    (3)  035664  062706  000010          ADD     #10,SP
    (3)                          .MEXIT
    (1)  035670  012737  000001  003142  MOV     #1,DLYCNT       ;INITIALIZE DELAY COUNT
    (1)  035676  012737  000001  172542  MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SE; BUFFER REGISTER
    (1)                                                          ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
    (1)  035704  012737  000113  172540  MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
    (1)                                                          ;/10 KHZ RATE,START THE CLOCK
    (1)  035712  005737  003142  64$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
    (1)  035716  001375          BNE     64$             ;BRANCH IF TIME NOT ELAPSED
    (1)  035720  005037  172540  CLR     @#172540        ;STOP THE CLOCK
3682                             .NLIST  ME
3683  035724  000720             BR      6$
3684.
3685  035726                     9$:     ERRHRD  1203.,,ERR5     ;REPORT READY ERROR
 (4)  035726  104456             TRAP    C$ERHRD
 (5)  035730  002263             .WORD   1203
 (5)  035732  000000             .WORD   0
 (5)  035734  012572             .WORD   ERR5
3686  035736                     EXIT    SUB             ;EXIT
 (3)  035736  104432             TRAP    C$EXIT
 (3)  035740  000074             .WORD   L10046-.
3687
3688  035742  005737  003044     12$:    TST     T.CS            ;TEST IF ANY ERROR
3689  035746  100006             BPL     15$             ;NO-SKIP
3690  035750                     ERRHRD  1204.,,ERR6     ;REPORT ALL ERRORS
 (4)  035750  104456             TRAP    C$ERHRD
 (5)  035752  002264             .WORD   1204
 (5)  035754  000000             .WORD   0
 (5)  035756  012642             .WORD   ERR6
3691  035760                     EXIT    SUB
 (3)  035760  104432             TRAP    C$EXIT
 (3)  035762  000052             .WORD   L10046-.
3692  035764  012703  010413     15$:    MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
3693  035770  004737  022266             JSR     PC,POSHSB       ;POSITION HEAD SELECT BIT
3694  035774  023705  003112             CMP     DESHD,R5        ;CHECK IF CORRECT HEAD SELECTED
```

```
3695  036000  001415                          BEQ     20$             ;YES-SKIP
3696  036002  005737  003112                  TST     DESHD           ;WAS HEAD 0 SELECTED
3697  036006  001406                          BEQ     17$             ;YES-SKIP
3698  036010                                  ERRHRD  1205.,,ERR3     ;REPORT HEAD SB 1
 (4)  036010  104456                          TRAP    C$ERHRD
 (5)  036012  002265                          .WORD   1205
 (5)  036014  000000                          .WORD   0
 (5)  036016  012454                          .WORD   ERR3
3699  036020                                  EXIT    SUB             ;EXIT
 (3)  036020  104432                          TRAP    C$EXIT
 (3)  036022  000012                          .WORD   L10046-.
3700  036024                            17$:  ERRHRD  1206.,,ERR2     ;ELSE REPORT HEAD SB 0
 (4)  036024  104456                          TRAP    C$ERHRD
 (5)  036026  002266                          .WORD   1206
 (5)  036030  000000                          .WORD   0
 (5)  036032  012406                          .WORD   ERR2
3701
3702  036034                            20$:
3703  036034                            60$:
3704  036034                            ENDSUB
 (3)  036034                            L10046:
 (3)  036034  104403                          TRAP    C$ESUB
3705  036036  005737  003112                  TST     DESHD           ;CHECK IF HD 0 WAS DONE
3706  036042  001404                          BEQ     25$             ;YES-SKIP
3707  036044  005037  003112                  CLR     DESHD           ;ELSE SET TO HEAD 0
3708  036050  000137  035476                  JMP     T124$           ;REDO TEST
3709  036054                            25$:
3710  036054                            ENDTST
 (3)  036054                            L10045:
 (3)  036054  104401                          TRAP    C$ETST
3711
3712
3713
3714                              .SBTTL  *TEST 13        READ HEADER (PART 1)
3715  036056                      BGNTST                  ;TEST 13
 (3)  036056                                                                  T13::
3716  036056  012737  006603  003012          MOV     #T13ERR,ERHEAD  ;SET ERROR HEADER
3717  036064  012701  003102                  MOV     #NEWCYL,R1      ;GET ADDRESS OF DESIRED LOCATIONS
3718  036070  005021                          CLR     (R1)+           ;CLEAR NEW CYL
3719  036072  005021                          CLR     (R1)+           ;CLEAR CURRENT CYL
3720  036074  005021                          CLR     (R1)+           ;CLEAR DIFF
3721  036076  005021                          CLR     (R1)+           ;CLEAR SIGN
3722  036100  005021                          CLR     (R1)+           ;CLEAR HEAD
3723  036102                            T134$:
3724  036102                            BGNSUB
 (3)  036102                                                                  T13.1:
 (3)  036102  104402                          TRAP    C$BSUB
3725  036104  004737  016512                  JSR     PC,TSTINT       ;INITIALIZE TEST
3726  036110  004737  016530                  JSR     PC,GSTATR       ;GET STATUS W/RESET
3727  036114  036206                          60$
3728  036116  004737  020622                  JSR     PC,SIMSEK       ;DO SEEK
3729  036122  036206                          60$
3730  036124  012701  000121                  MOV     #81.,R1         ;SET WAIT COUNT
3731  036130  004737  022316                  JSR     PC,RDYWAIT      ;WAIT FOR READY
3732  036134  036206                          60$
3733
```

```
 3734  036136  004737  021576          10$:    JSR     PC,XRDHDC       ;DO READ HEADER
 3735  036142  036206                          60$
 3736  036144  012703  010413                  MOV     #MHSTA,R3       ;SET NAME MESSAGE PTR
 3737  036150  004737  022260                  JSR     PC,POSHW1       ;POSITION HS BIT IN HD WRD 1
 3738  036154  020537  003112                  CMP     R5,DESHD        ;CHECK IF HEAD CORRECT
 3739  036160  001412                          BEQ     15$             ;YES-SKIP
 3740  036162                                  ERRHRD  1301.,,ERR3     ;REPORT SB 1
  (4)  036162  104456                          TRAP    C$ERHRD
  (5)  036164  002425                          .WORD   1301
  (5)  036166  000000                          .WORD   0
  (5)  036170  012454                          .WORD   ERR3
 3741  036172                                  EXIT    SUB
  (3)  036172  104432                          TRAP    C$EXIT
  (3)  036174  000012                          .WORD   L10050-.
 3742  036176                          17$:    ERRHRD  1302.,,ERR2     ;REPORT SB 0
  (4)  036176  104456                          TRAP    C$ERHRD
  (5)  036200  002426                          .WORD   1302
  (5)  036202  000000                          .WORD   0
  (5)  036204  012406                          .WORD   ERR2
 3743
 3744  036206                          15$:
 3745  036206                          60$:
 3746  036206                          ENDSUB
  (3)  036206                          L10050:
  (3)  036206  104403                          TRAP    C$ESUB
 3747  036210  005737  003112                  TST     DESHD           ;TEST IF HEAD 1 DONE
 3748  036214  001007                          BNE     20$             ;YES-SKIP
 3749  036216  012737  000001  003112          MOV     #1,DESHD        ;ELSE SET TO HEAD 1
 3750  036224  013737  003052  003116          MOV     HDWRD1,TEMP0    ;STORE HDR WORD 1
 3751  036232  000723                          BR      T134$           ;DO TEST AGAIN
 3752  036234  042737  000177  003116  20$:    BIC     #177,TEMP0      ;CLEAR ALL BUT CYLINDER IN 1ST HEADER
 3753  036242  042737  000177  003052          BIC     #177,HDWRD1     ;CLEAR ALL BY CYL IN 2ND HEADER
 3754  036250  023737  003116  003052          CMP     TEMP0,HDWRD1    ;COMPARE IF EQUAL
 3755  036256  001406                          BEQ     22$             ;YES-SKIP
 3756  036260  012703  007044                  MOV     #CYLPER,R3      ;SET NAME MESSAGE PTR
 3757  036264                                  ERRHRD  1306.,,ERR1     ;REPORT HEAD ALIGNMENT PROBLEM
  (4)  036264  104456                          TRAP    C$ERHRD
  (5)  036266  002432                          .WORD   1306
  (5)  036270  000000                          .WORD   0
  (5)  036272  012340                          .WORD   ERR1
 3758  036274                          22$:
 3759  036274                          ENDTST
  (3)  036274                          L10047:
  (3)  036274  104401                          TRAP    C$ETST
 3760
 3761
 3762
 3763                                  .SBTTL  *TEST 14         READ HEADER (PART 2)
 3764  036276                          BGNTST          ;TEST 14
  (3)  036276                                                                  T14::
 3765  036276  012737  006617  003012          MOV     #T14ERR,ERHEAD  ;SET ERROR HEADER
 3766  036304  012701  003104                  MOV     #CURCYL,R1      ;GET ADDRESS OF DESIRED VALUE
 3767  036310  005021                          CLR     (R1)+           ;CLEAR CURRENT CYL
 3768  036312  005021                          CLR     (R1)+           ;CLEAR DESIRED DIFF
 3769  036314  005021                          CLR     (R1)+           ;CLEAR SIGN
 3770  036316  005021                          CLR     (R1)+           ;CLEAR DESIRED HEAD
```

```
 3771  036320                          T153$:
 3772  036320                          BGNSUB
  (3)  036320                                                          T14.1:
  (3)  036320  104402                      TRAP    C$BSUB
 3773  036322  004737  016512            JSR     PC,TSTINT       ;INITIALIZE TEST
 3774  036326  004737  016530            JSR     PC,GSTATR       ;CLEAR DRIVE
 3775  036332  036532                    60$
 3776  036334  004737  020622            JSR     PC,SIMSEK       ;DO SEEK
 3777  036340  036532                    60$
 3778  036342  012701  000310            MOV     #200.,R1        ;SET WAIT COUNT FOR 20 MS
 3779  036346  004737  022316            JSR     PC,RDYWAIT      ;WAIT FOR READY
 3780  036352  036532                    60$
 3781  036354  004737  023010            JSR     PC,RDALHD       ;DO READ HEADER ALL HEADERS
 3782  036360  036532                    60$
 3783  036362  005037  003014            CLR     MORECE          ;CLEAR MORE COMPARE ERRORS FOR REPORT
 3784  036366  052737  000002  003004    BIS     #HDRCMP,OPFLAG  ;SET HDR COMPARE FLAG
 3785  036374  005003                    CLR     R3              ;CLEAR FOR HDR COUNT
 3786  036376  012704  003762            MOV     #IBUFF,R4       ;GET POINTER FOR HDR TO BE CHECKED
 3787  036402  012705  003116            MOV     #TEMP0,R5       ;GET POINTER TO TEST AREA
 3788  036406  012701  000050            MOV     #40.,R1         ;SET HDR COUNT
 3789  036412  011415                    MOV     (R4),(R5)       ;GET FIRST HEADER WORD
 3790
 3791  036414  042715  000100            BIC     #HDHSEL,(R5)
 3792  036420  005737  003112            TST     DESHD           ;TEST IF HD 0 DESIRED
 3793  036424  001404                    BEQ     10$             ;YES-SKIP
 3794  036426  052715  000100            BIS     #HDHSEL,(R5)    ;ELSE SET HEAD BIT
 3795  036432  005065  000002            CLR     2(R5)           ;CLEAR 2ND WORD OF TEST AREA
 3796  036436  021524          10$:      CMP     (R5),(R4)+      ;COMPARE HEADER WORD
 3797  036440  001406                    BEQ     13$             ;SKIP IF OK
 3798  036442  005744                    TST     -(R4)           ;ELSE POSITION R4 TO BAD WORD
 3799  036444                          ERRHRD  1501.,,ERR10    ;REPORT ERROR
  (4)  036444  104456                    TRAP    C$ERHRD
  (5)  036446  002735                    .WORD   1501
  (5)  036450  000000                    .WORD   0
  (5)  036452  013752                    .WORD   ERR10
 3800  036454  005724                    TST     (R4)+           ;BUMP R4 TO NEXT WORD
 3801  036456  005203          13$:      INC     R3              ;BUMP WORD COUNT
 3802  036460  005724                    TST     (R4)+           ;TEST 2ND WORD IS 0
 3803  036462  001406                    BEQ     15$             ;YES - SKIP
 3804  036464  022544                    CMP     (R5)+,-(R4)     ;POSITION PTRS FOR REPORT
 3805  036466                          ERRHRD  1501.,,ERR10    ;REPORT ERROR
  (4)  036466  104456                    TRAP    C$ERHRD
  (5)  036470  002735                    .WORD   1501
  (5)  036472  000000                    .WORD   0
  (5)  036474  013752                    .WORD   ERR10
 3806  036476  024524                    CMP     -(R5),(R4)+     ;REPOSITION POINTER
 3807  036500  005724          15$:      TST     (R4)+           ;POSITION R4 PAST ECC WORD
 3808  036502  005203                    INC     R3              ;BUMP WORD COUNT
 3809  036504  005215                    INC     (R5)            ;BUMP SECTOR COUNT
 3810  036506  011500                    MOV     (R5),R0         ;CHECK IF SECTOR IS PAST LAST SECTOR
 3811  036510  042700  177700            BIC     #^CHDSEC,R0
 3812  036514  022700  000050            CMP     #40.,R0
 3813  036520  001002                    BNE     17$             ;NO-SKIP
 3814  036522  042715  000077            BIC     #HDSEC,(R5)     ;ELSE CLEAR SECTOR TO 0
 3815  036526  005301          17$:      DEC     R1              ;DEC HDR COUNT
 3816  036530  001342                    BNE     10$             ;YES-SKIP
```

```
3817
3818  036532                                  60$:
3819  036532                                  ENDSUB
 (3)  036532                                  L10052:
 (3)  036532  104403                                 TRAP    C$ESUB
3820  036534  005737  003112                        TST     DESHD           ;CHECK IF HD 1 TESTED
3821  036540  001005                                BNE     20$             ;YES-SKIP
3822  036542  012737  000001  003112                MOV     #1,DESHD        ;ELSE SET TO HEAD 1
3823  036550  000137  036320                        JMP     T153$           ;REDO TEST
3824  036554                                  20$:
3825  036554                                  ENDTST
 (3)  036554                                  L10051:
 (3)  036554  104401                                 TRAP    C$ETST
3826
3827
3828
3829                                          .SBTTL  *TEST 15        DIFFERENCE OF 1 SEEK (PART 1)
3830  036556                                  BGNTST                  ;TEST 15
 (5)  036556                                                                                    T15::
3831
3832                                          ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3833  036556  005737  003144                        TST     CLKFLG          ;P-CLOCK?
3834  036562  001024                                BNE     1$              ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3835  036564  012702  006662                        MOV     #NOTST,R2       ;INITIALIZE POINTER FOR TEST MSG.
3836  036570  112762  000061  000004                MOVB    #'1,4(R2)       ;INSERT TEST NUMBER INTO MSG.
3837  036576  112762  000065  000005                MOVB    #'5,5(R2)       ;INSERT TEST NUMBER INTO MSG.
3838  036604                                        PRINTF  #FMT9,#NOTST    ;PRINT MSG. 'TST 15 CANNOT BE PERFORMED...
 (8)  036604  012746  006662                        MOV     #NOTST,-(SP)
 (7)  036610  012746  011627                        MOV     #FMT9,-(SP)
 (6)  036614  012746  000002                        MOV     #2,-(SP)
 (3)  036620  010600                                MOV     SP,R0
 (4)  036622  104417                                TRAP    C$PNTF
 (4)  036624  062706  000006                        ADD     #6,SP
3839                                                                    ;/NO P-CLK''
3840  036630                                  EXIT    TST
 (3)  036630  104432                                TRAP    C$EXIT
 (3)  036632  000444                                .WORD   L10053-.
3841  036634  012737  006643  003012  1$:           MOV     #P2T01E,ERHEAD  ;SET ERROR HEADER
3842  036642  012737  000004  003116                MOV     #4,TEMP0        ;SET PASS COUNT
3843  036650  004737  016512                        JSR     PC,TSTINT       ;INITIALIZE TEST
3844  036654  004737  016530                        JSR     PC,GSTATR       ;GET STATUS
3845  036660  037276                                T1765$
3846  036662  022737  000001  002276                CMP     #1,T.DRIVE      ;RL01 OR RL02?
3847  036670  001404                                BEQ     2$              ;BRANCH TO SET UP DIFF ARGUMENT FOR RL01
3848  036672  012737  177776  003122                MOV     #-2,TEMP2       ;ELSE, SET -2 INTO DIFF ARGUMENT FOR RL02
3849                                                                    ;/(RL02 HAS DOUBLE THE TRACK DENSITY OF RL01)
3850  036700  000403                                BR      5$
3851  036702  012737  177777  003122  2$:           MOV     #-1,TEMP2       ;SET -1 INTO DIFF ARGUMENT FOR -1 SEEK
3852  036710  012704  003104          5$:           MOV     #CURCYL,R4      ;SET POINTERS
3853  036714  012705  003102                        MOV     #NEWCYL,R5
3854  036720  004737  021466                        JSR     PC,CHOSHD       ;GO CHOOSE HEAD
3855  036724                          T172$:
3856  036724                          BGNSUB
 (3)  036724                                                                                    T15.1:
 (3)  036724  104402                                TRAP    C$BSUB
3857  036726  004737  022662                        JSR     PC,GETPOS       ;GET POSITION
```

```
3858  036732  037234                                  60$
3859  036734                                  INLOOP                  ;CHECK IF IN ERROR LOOP
 (3)  036734  104420                          TRAP    C$INLP
3860  036736                                  BNCOMPLETE      3$      ;NO - SKIP
 (2)  036736  103005                          BCC     3$
3861  036740  021415                          CMP     (R4),(R5)       ;CHECK IF CURRENT = NEW
3862  036742  001005                          BNE     4$              ;NO - SKIP
3863  036744  004737  021552                  JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
3864  036750  000441                          BR      9$              ;SKIP TO SEEK
3865  036752  005437  003122          3$:     NEG     TEMP2           ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
3866  036756  011415                  4$:     MOV     (R4),(R5)       ;MOVE CURRENT INTO OLD
3867  036760  023714  002302                  CMP     HLMTW,(R4)      ;CHECK IF CURRENT AT 255
3868  036764  001014                          BNE     7$              ;NO - SKIP
3869  036766  022737  000001  002276          CMP     #1,T.DRIVE      ;RL01 OR RL02?
3870  036774  001404                          BEQ     6$              ;BRANCH IF RL01
3871  036776  012737  177776  003122          MOV     #-2,TEMP2       ;ELSE, SET UP DIFF ARGUMENT FOR RL02
3872  037004  000421                          BR      8$
3873  037006  012737  177777  003122  6$:     MOV     #-1,TEMP2       ;AT MAX CYL, MAKE NEXT SEEK REV
3874  037014  000415                          BR      8$              ;SKIP
3875  037016  005714                  7$:     TST     (R4)            ;TEST IF CURRENT AT 0
3876  037020  001013                          BNE     8$              ;NO - SKIP
3877  037022  022737  000001  002276          CMP     #1,T.DRIVE      ;RL01 OR RL02?
3878  037030  001404                          BEQ     11$             ;BRANCH IF RL01
3879  037032  012737  000002  003122          MOV     #2,TEMP2        ;ELSE, SET UP DIFF ARGUMENT FOR RL02
3880  037040  000403                          BR      8$
3881  037042  012737  000001  003122  11$:    MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
3882  037050  063715  003122          8$:     ADD     TEMP2,(R5)      ;ADD DIFF TO NEW CYL (+1 OR -1 FOR RL01,
3883                                                                  ;/+2 OR -2 FOR RL02)
3884  037054  004737  020032          9$:     JSR     PC,XSEEK        ;DO SEEK
3885  037060  037234                          60$
3886  037062  004737  017652                  JSR     PC,GDRSTA       ;GET DRIVE STATE
3887
3888  037066  012703  000004                  MOV     #4,R3           ;SET EXPECTED STATE
3889  037072  020337  003060                  CMP     R3,T.STAT       ;CHECK DRIVE STATE
3890  037076  001405                          BEQ     10$             ;YES-SKIP
3891  037100                                  ERRHRD  101.,,ERR7      ;REPORT STATE ERROR
 (4)  037100  104456                          TRAP    C$ERHRD
 (5)  037102  000145                          .WORD   101
 (5)  037104  000000                          .WORD   0
 (5)  037106  013542                          .WORD   ERR7
3892  037110  000444                          BR      16$             ;EXIT TEST
3893  037112  012703  000005          10$:    MOV     #5,R3           ;SET EXPECTED STATE
3894                                  .LIST   ME
3895  037116                                  TIMDLY  #50.            ;WAIT 5 MS FOR DRIVE STATE CHANGE FROM 4 TO 5
 (1)  037116                                  SETVEC  #104,#CLKINT,#340 ;SET P-CLOCK INTERRUPT VECTOR
 (2)  037116                          M$PUT   #3,#104,#CLKINT,#340
 (3)  037116                          M$PUT   <#104>,<#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>
 (4)  037116                          M$PUT   <#CLKINT>,<#340>,<>,<>,<>,<>,<>,<>,<>
 (5)  037116                          M$PUT   <#340>,<>,<>,<>,<>,<>,<>,<>,<>
 (6)  037116                          M$PUT1  #340
 (7)  037116                          M$GNINS <MOV    #340,-(SP)>
 (8)  037116  012746  000340                  MOV     #340,-(SP)
 (8)                                  .MEXIT
 (5)  037122                          M$PUT1  #CLKINT
 (6)  037122                          M$GNINS <MOV    #CLKINT,-(SP)>
 (7)  037122  012746  016112                  MOV     #CLKINT,-(SP)
```

```
   (7)                                          .MEXIT
   (4)  037126                                  M$PUT1  #104
   (5)  037126                                  M$GNINS <MOV    #104,-(SP)>
   (6)  037126  012746  000104                          MOV     #104,-(SP)
   (6)                                          .MEXIT
   (3)  037132                                  M$PUT1  #3
   (4)  037132                                  M$GNINS <MOV    #3,-(SP)>
   (5)  037132  012746  000003                          MOV     #3,-(SP)
   (5)                                          .MEXIT
   (2)  037136                                  M$SVC   C$SVEC
   (3)  037136                                  M$TSTLAB
   (4)                                           .MEXIT
   (3)  037136                                  M$GNINS <TRAP   C$SVEC>
   (4)  037136  104437                                  TRAP    C$SVEC
   (4)                                          .MEXIT
   (2)  037140                                  .$GNINS <ADD    #10,SP>
   (3)  037140  062706  000010                          ADD     #10,SP
   (3)                                          .MEXIT
   (1)  037144  012737  000062  003142                  MOV     ##50.,DLYCNT    ;INITIALIZE DELAY COUNT
   (1)  037152  012737  000001  172542                  MOV     #1,@#172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
   (1)                                                                          ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
   (1)  037160  012737  000113  172540                  MOV     #113,@#172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
   (1)                                                                          ;/10 KHZ RATE,START THE CLOCK
   (1)  037166  005737  003142          64$:    TST     DLYCNT          ;DELAY COUNT EXPIRED?
   (1)  037172  001375                          BNE     64$             ;BRANCH IF TIME NOT ELAPSED
   (1)  037174  005037  172540                  CLR     @#172540        ;STOP THE CLOCK
3896                                    .NLIST  ME
3897    037200  004737  017652          12$:    JSR     PC,GDRSTA       ;GET DRIVE STATE
3898    037204  020337  003060                  CMP     R3,T.STAT       ;IS STATE 5?
3899    037210  001404                          BEQ     16$             ;YES-SKIP
3900    037212                          14$:    ERRHRD  102.,,ERR7      ;REPORT STATE ERROR
 (4)    037212  104456                          TRAP    C$ERHRD
 (5)    037214  000146                          .WORD   102
 (5)    037216  000000                          .WORD   0
 (5)    037220  013542                          .WORD   ERR7
3901    037222  012701  000062          16$:    MOV     #50.,R1         ;INITIALIZE WAIT COUNT
3902    037226  004737  022316                  JSR     PC,RDYWAIT      ;GO WAIT FOR DRIVE READY
3903    037232  037234                          60$
3904    037234  012737  000002  003016  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3905    037242                          ENDSUB
 (3)    037242                          L10054:
 (3)    037242  104403                          TRAP    C$ESUB
3906    037244                                  ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)    037244  104410                          TRAP    C$ESCAPE
 (3)    037246  000030                          .WORD   L10053-.
3907    037250  005337  003116                  DEC     TEMP0           ;DEC PASS COUNT
3908    037254  001410                          BEQ     24$             ;SKIP IF 0-DONE
3909
3910    037256  032737  000001  003116          BIT     #BIT0,TEMP0     ;TEST IF PASS=2
3911    037264  001003                          BNE     23$             ;NO-SKIP
3912    037266  004737  021512                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
3913    037272  037276                          24$                     ;ABORT RETURN
3914    037274  000613                  23$:    BR      T172$
3915    037276                          24$:
3916    037276                          T1765$:
3917    037276                          ENDTST
```

```
  (3)  037276                          L10053:
  (3)  037276  104401                          TRAP    C$ETST
3918
3919
3920
3921                                   .SBTTL  *TEST 16        DIFFERENCE OF 1 SEEK (PART 2)
3922  037300                           BGNTST                  ;TEST 16
  (3)  037300                                                                    T16::
3923  037300  012737  006643  003012           MOV     #P2T02E,ERHEAD  ;SET ERROR HEADER
3924  037306  012737  000004  003116           MOV     #4,TEMP0        ;SET PASS COUNT
3925  037314  004737  016512                   JSR     PC,TSTINT       ;INITIALIZE TEST
3926  037320  004737  016530                   JSR     PC,GSTATR       ;GET STATUS, CLEAR DRIVE
3927  037324  037570                           T1865$
3928  037326  004737  021466                   JSR     PC,CHOSHD       ;GO CHOOSE HEAD
3929  037332  012737  177777  003122           MOV     #-1,TEMP2       ;SET DIFF ARGUMENT TO -1 (REVERSE)
3930  037340  012703  003102                   MOV     #NEWCYL,R3      ;GET ADDRESSES
3931  037344  012704  003104                   MOV     #CURCYL,R4
3932  037350  012705  003100                   MOV     #OLDCYL,R5
3933  037354                           T187$:
3934  037354                           BGNSUB
  (3)  037354                                                                    T16.1:
  (3)  037354  104402                          TRAP    C$BSUB
3935  037356  004737  022662                   JSR     PC,GETPOS       ;GET CURRENT POSITION
3936  037362  037526                           60$
3937  037364                                   INLOOP                  ;CHECK IF IN ERROR LOOP
  (3)  037364  104420                          TRAP    C$INLP
3938  037366                                   BNCOMPLETE      3$      ;NO - SKIP
  (2)  037366  103005                          BCC     3$
3939  037370  021413                           CMP     (R4),(R3)       ;CHECK IF CURRENT = NEW
3940  037372  001005                           BNE     4$              ;NO - SKIP
3941  037374  004737  021552                   JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
3942  037400  000421                           BR      9$              ;SKIP TO SEEK
3943  037402  005437  003122           3$:     NEG     TEMP2           ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
3944  037406  011413                   4$:     MOV     (R4),(R3)       ;MOV CURRENT INTO NEW
3945  037410  023714  002302                   CMP     HLMTW,(R4)      ;CHECK IF CURRENT AT 255
3946  037414  001004                           BNE     7$              ;NO - SKIP
3947  037416  012737  177777  003122           MOV     #-1,TEMP2       ;AT MAX CYL, MAKE NEXT SEEK REV
3948  037424  000405                           BR      8$              ;SKIP
3949  037426  005714                   7$:     TST     (R4)            ;TEST IF CURRENT AT 0
3950  037430  001003                           BNE     8$              ;NO - SKIP
3951  037432  012737  000001  003122           MOV     #1,TEMP2        ;AT CYL 0, MAKE NEXT SEEK FWRD
3952  037440  063713  003122           8$:     ADD     TEMP2,(R3)      ;ADD DIFF TO NEW CYL (+1 OR -1)
3953  037444  004737  020032           9$:     JSR     PC,XSEEK        ;DO SEEK
3954  037450  037526                           60$
3955  037452  012701  000226                   MOV     #150.,R1        ;SET WAIT COUNT FOR 15 MS
3956  037456  004737  022316                   JSR     PC,RDYWAIT      ;WAIT FOR READY
3957  037462  037526                           60$
3958  037464  004737  022662                   JSR     PC,GETPOS       ;STORE POSITION
3959  037470  037526                           60$
3960  037472  011501                           MOV     (R5),R1         ;GET OLD POSITION
3961  037474  161401                           SUB     (R4),R1         ;SUBTRACT FROM NEW POINTER (FORWARD)
3962  037476  005737  003110                   TST     DESSGN          ;CHECK IF SIGN FORWARD
3963  037502  001402                           BEQ     10$             ;YES-SKIP, ELSE SUB FOR SEEK REVERSE
3964  037504  011401                           MOV     (R4),R1         ;GET NEW CYLINDER
3965  037506  161501                           SUB     (R5),R1         ;SUBTRACT FROM OLD CYL
3966  037510  022701  000001           10$:    CMP     #1,R1           ;CHECK IF RESULT IS DIFFERENCE OF 1
```

```
3967  037514  001404                          BEQ     12$             ;YES-SKIP
3968  037516                                  ERRHRD  201,,ERR8       ;ELSE REPORT ERROR
 (4)  037516  104456                          TRAP    C$ERHRD
 (5)  037520  000311                          .WORD   201
 (5)  037522  000000                          .WORD   0
 (5)  037524  013612                          .WORD   ERR8
3969  037526                          12$:
3970  037526  012737  000002  003016  60$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
3971  037534                          ENDSUB
 (3)  037534                          L10056:
 (3)  037534  104403                          TRAP    C$ESUB
3972  037536                                  ESCAPE  TST             ;EXIT TEST IF ERROR
 (3)  037536  104410                          TRAP    C$ESCAPE
 (3)  037540  000030                          .WORD   L10055-.
3973  037542  005337  003116                  DEC     TEMP0           ;DEC PASS COUNT
3974  037546  001410                          BEQ     30$             ;EXIT IF DONE
3975
3976  037550  032737  000001  003116          BIT     #BIT0,TEMP0     ;TEST IF PASS 1 OR 3
3977  037556  001003                          BNE     20$             ;YES-SKIP
3978  037560  004737  021512                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
3979  037564  037570                          30$                     ;ABORT RETURN
3980  037566  000672                  20$:    BR      T187$           ;LOOP
3981  037570                          30$:
3982  037570                          T1865$:
3983  037570                          ENDTST
 (3)  037570                          L10055:
 (3)  037570  104401                          TRAP    C$ETST
3984  037572                          ENDMOD
3985
3986                                  .SBTTL  PARAMETER CODING
3987  037572                          BGNMOD  HRDPRM
3988  037572                          BGNHRD
 (3)  037572  000030                          .WORD L10057-L$HARD/2
3989
3990  037574                          GPRML   CNTYPE,CNT,1,YES
 (4)  037574  005130                          .WORD   T$CODE
 (4)  037576  037740                          .WORD   CNTYPE
 (4)  037600  000001                          .WORD   1
3991
3992  037602                          GPRMA   CSRMSG,CSR,0,160000,177776,YES
 (4)  037602  000031                          .WORD   T$CODE
 (4)  037604  037654                          .WORD   CSRMSG
 (4)  037606  160000                          .WORD   T$LOLIM
 (4)  037610  177776                          .WORD   T$HILIM
3993
3994  037612                          GPRMA   VECMSG,VECT,0,0,776,YES
 (4)  037612  001031                          .WORD   T$CODE
 (4)  037614  037670                          .WORD   VECMSG
 (4)  037616  000000                          .WORD   T$LOLIM
 (4)  037620  000776                          .WORD   T$HILIM
3995
3996  037622                          GPRMD   DRMSG,DRSB,0,3400,0,7,YES
 (4)  037622  004032                          .WORD   T$CODE
 (4)  037624  037732                          .WORD   DRMSG
 (4)  037626  003400                          .WORD   3400
 (4)  037630  000000                          .WORD   T$LOLIM
```

```
  (4)  037632  000007                          .WORD   T$HILIM
3997
3998  037634                                   GPRML   DRTYPE,TYPDR,1,YES
  (4)  037634  003130                          .WORD   T$CODE
  (4)  037636  037710                          .WORD   DRTYPE
  (4)  037640  000001                          .WORD   1
3999
4000  037642                                   GPRMD   BRMSG,PRIOR,0,340,0,7,YES
  (4)  037642  002032                          .WORD   T$CODE
  (4)  037644  037677                          .WORD   BRMSG
  (4)  037646  000340                          .WORD   340
  (4)  037650  000000                          .WORD   T$LOLIM
  (4)  037652  000007                          .WORD   T$HILIM
4001
4002  037654                                   ENDHRD
  (2)                                          .EVEN
  (3)  037654                                  L10057:
4003
4004                                           .EVEN
4005
4006  037654  052502  020123  042101  CSRMSG: .ASCIZ  /BUS ADDRESS/
      037662  051104  051505  000123
4007
4008  037670  042526  052103  051117  VECMSG: .ASCIZ  /VECTOR/
      037676     000
4009
4010  037677     102  020122  042514  BRMSG:  .ASCIZ  /BR LEVEL/
      037704  042526  000114
4011
4012  037710  051104  053111  020105  DRTYPE: .ASCIZ  /DRIVE TYPE = RL01/
      037716  054524  042520  036440
      037724  051040  030114  000061
4013
4014  037732  051104  053111  000105  DRMSG:  .ASCIZ  /DRIVE/
4015
4016  037740  046122  030461     000  CNTYPE: .ASCIZ  /RL11/
4017
4018  037745                                   ENDMOD
4019
4020          037746                           .EVEN
4021
4022  037746                                   BGNMOD  SFTPRM
4023  037746                                   BGNSFT
  (3)  037746  000016                          .WORD L10060-L$SOFT/2
4024
4025  037750                                   GPRML   SELQ,MISWI,4,YES
  (4)  037750  000130                          .WORD   T$CODE
  (4)  037752  040004                          .WORD   SELQ
  (4)  037754  000004                          .WORD   4
4026
4027  037756                                   GPRML   ALGNQ,MISWI,10,YES
  (4)  037756  000130                          .WORD   T$CODE
  (4)  037760  040037                          .WORD   ALGNQ
  (4)  037762  000010                          .WORD   10
4028
4029  037764                                   GPRML   MANQ,MISWI,100000,YES
```

```
 (4)  037764  000130                          .WORD   T$CODE
 (4)  037766  040076                          .WORD   MANQ
 (4)  037770  100000                          .WORD   100000
4030
4031  037772                                 3$:     GPRMD   ERLIMQ,ERLIM,D,377,0,377,YES
 (4)  037772  004052                          .WORD   T$CODE
 (4)  037774  040133                          .WORD   ERLIMQ
 (4)  037776  000377                          .WORD   377
 (4)  040000  000000                          .WORD   T$LOLIM
 (4)  040002  000377                          .WORD   T$HILIM
4032
4033  040004                                 ENDSFT
 (2)                                          .EVEN
 (3)  040004                                 L10060:
4034
4035                                         .EVEN
4036
4037  040004  054105  041505  052125  SELQ:   .ASCIZ  /EXECUTE DRIVE SELECT TESTS/
       040012  020105  051104  053111
       040020  020105  042523  042514
       040026  052103  052040  051505
       040034  051524     000
4038
4039  040037     105  042530  052503  ALGNQ:  .ASCIZ  /EXECUTE HEAD ALIGNMENT SUPPORT/
       040044  042524  044040  040505
       040052  020104  046101  043511
       040060  046516  047105  020124
       040066  052523  050120  051117
       040074  000124
4040
4041  040076  047504  046440  047101  MANQ:   .ASCIZ  /DO MANUAL INTERVENTION TESTS/
       040104  040525  020114  047111
       040112  042524  053122  047105
       040120  044524  047117  052040
       040126  051505  051524     000
4042
4043  040133     111  050116  052125  ERLIMQ: .ASCIZ  /INPUT ERROR LIMIT/
       040140  042440  051122  051117
       040146  046040  046511  052111
       040154     000
4044
4045          040156                         .EVEN
4046
4047  040156                                 ENDMOD
4048
4049  040156                                 LASTAD
 (2)                                          .EVEN
 (4)  040156  000000                          .WORD   0
 (4)  040160  000000                          .WORD   0
 (3)  040162                                 L$LAST::
4050                                         .EVEN
4051
4052  040162                                 L$LAST::
4053
4054          000001                         .END
```

```
ADR    = 000020 G          51#
ALGNQ    040037          4027    4039#
ALLCYL= 000001             70#
ALLSEC= 000002             71#
ANYERR= 100000            123#   1601    1946    2222    3207
ASSEMB= 000010              8
BADADD= 004000            102#
BAMSK  = 000060           134#
BANAM    006123           661#   2687
BASADD   006040           657#   1139    1324    1358    1369    1449    2685    2743    2783    2830    2837    3004    3041
                         3048    3104    3122    3306    3316    3569
BELL     011150           798#
BHSTAT= 000010            161#   2755    2887    2894    2907    3335
BIT0   = 000001 G          51#   3491    3910    3976
BIT00  = 000001 G          51#     70      92     111    2632
BIT01  = 000002 G          51#     71      91     112
BIT02  = 000004 G          51#     72      93     113
BIT03  = 000010 G          51#     73      94     114
BIT04  = 000020 G          51#     95
BIT05  = 000040 G          51#     96
BIT06  = 000100 G          51#     97
BIT07  = 000200 G          51#     98
BIT08  = 000400 G          51#     99
BIT09  = 001000 G          51#    100
BIT1   - 000002 G          51#
BIT10  = 002000 G          51#    101    1572    1650    1731    1794    2037    2321
BIT11  = 004000 G          51#    102
BIT12  = 010000 G          51#     74     103
BIT13  = 020000 G          51#     75     104
BIT14  = 040000 G          51#     76     105
BIT15  = 100000 G          51#     77     106    1033
BIT2   = 000004 G          51#
BIT3     000010 G          51#
BIT4   - 000020 G          51#
BIT5   - 000040 G          51#
BIT6   = 000100 G          51#
BIT7   = 000200 G          51#
BIT8   = 000400 G          51#
BIT9   = 001000 G          51#   1002
BOE    = 000400 G          51#
BRMSG    037677          4000    4010#
BSFLAG   003020           423#
BSFVAL   003370           519#
BSNSTR   010163           760#
BYPSNM   010114           757#
CAFDT    011277           816#   2050
CAMSK    002312           250#   1274*   1282*
CCYLUP   011266           815#   3347    3396
CDRDY    011227           811#   3429    3502    3658
CHOSHD   021466          1965#   3854    3928
CKDATA= 000102              80#
CKERLM   016126           918     932     946     961     976    1081    1095    1117    1131    1145    1444#
CLKADR   003146           504#   1202*
CLKFLG   003144           503#   1201*   1204*   2815    3025    3174    3287    3407    3478    3635    3833
CLKINT   016112 G        1423#   2850    2874    2899    2912    2936    2963    2977    2990    3064    3082    3096    3203
                         3236    3256    3298    3450    3525    3681    3895
```

```
CLNCOD  015660 G          1379#
CLRBYT  002304             247#  1277*  1286*
CLRPAR  024606            2649   2695#
CNT   = 000012              59#  3990
CNTYPE  037740            3990   4016#
COMPOP= 007777              90#  2628
CONHNG= 000004             113#
CONTIN  014470            1238#
COSTAT= 000040             163#  2751   2793
COUNT   003152             508#
CRDYMS= 000200             132#  1000   1312   1364   1385   1469   1490   2329   2335   3224
CSNAM   006116             660#  2687
CSR   = 000000              54#  3992
CSRMSG  037654            3992   4006#
CURCYL  003104             455#  1693   1710   2260*  2648   2689   3766   3852   3931
CYLPER  007044             730#  3756
CYLTBL  002604             330#
CYLUP = 000004              93#  2839
CYLWD   010101             755#  2648   2687
C$AU  = 000052               8#
C$AUTO= 000061               8#  1374
C$BRK = 000022               8#  3312   3326   3586   3606
C$BSEG= 000004               8#
C$BSUB= 000002               8#  3034   3146   3232   3495   3573   3613   3651   3724   3772   3856   3934
C$CEFG= 000045               8#
C$CLCK= 000062               8#  1202
C$CLEA= 000012               8#  1398
C$CLOS= 000035               8#
C$CLP1= 000006               8#
C$CVEC= 000036               8#  1373   1393   1397
C$DCLN= 000044               8#  1328   1452
C$DODU= 000051               8#  1327   1361   1372   1451
C$DRPT= 000024               8#
C$DU  = 000053               8#  1402
C$EDIT= 000003               8#    18
C$ERDF= 000055               8#
C$ERHR= 000056               8#  1557   1596   1609   1615   1758   1763   1821   1935   1948   2051   2062   2066
                          2074   2208   2220   2224   2342   2347   2754   2758   2762   2766   2796   2801   2860
                          2869   2877   2882   2889   2902   2915   2923   2932   2946   2951   2966   2980   2996
                          3071   3085   3099   3253   3262   3338   3348   3361   3370   3397   3434   3442   3453
                          3458   3466   3468   3507   3517   3529   3533   3542   3544   3663   3672   3685   3690
                          3698   3700   3740   3742   3757   3799   3805   3891   3900   3968
C$ERRO= 000060               8#
C$ERSF= 000054               8#
C$ERSO= 000057               8#
C$ESCA= 000010               8#  3906   3972
C$ESEG= 000005               8#
C$ESUB= 000003               8#  3103   3151   3268   3548   3611   3626   3704   3746   3819   3905   3971
C$ETST= 000001               8#  2770   2803   3013   3111   3162   3275   3378   3400   3471   3554   3628   3710
                          3759   3825   3917   3983
C$EXIT= 000032               8#  2813   2861   2870   2883   2890   2903   2916   2924   2933   2947   2952   2967
                          2981   3023   3072   3086   3172   3254   3263   3285   3339   3349   3371   3414   3435
                          3443   3454   3459   3467   3485   3508   3518   3530   3534   3543   3567   3642   3664
                          3673   3686   3691   3699   3741   3840
C$GETB= 000026               8#
C$GETW= 000027               8#
```

```
C$GMAN= 000043      8#  2745  2786  2834  2854  3008  3043  3068  3106  3125  3143  3158  3228
                3271    3320  3621
C$GPHR= 000042      8#  1256  3191
C$GPLO= 000030      8#
C$GPRI= 000040      8#
C$INIT= 000011      8#  1334
C$INLP= 000020      8#  1446  3859  3937
C$MANI= 000050      8#  1206  1298
C$MEM = 000031      8#
C$MSG = 000023      8#   919   933   947   962   977  1082  1096  1118  1132  1146
C$OPEN= 000034      8#
C$PNTB= 000014      8#  1020  1053  1067  1076  1139  1140  1142  2603  2604  2608  2621  2637
                2641    2645  2648  2662  2671  2672  2675  2685  2687  2688  2689
C$PNTF= 000017      8#  1323  1324  1326  1357  1358  1360  1368  1369  1371  1448  1449  1450
                2743    2783  2821  2830  2837  3004  3030  3041  3048  3104  3122  3140  3156
                3179    3210  3226  3269  3292  3306  3316  3412  3483  3569  3570  3571  3619
                3640    3838
C$PNTS= 000016      8#
C$PNTX= 000015      8#
C$QIO = 000377      8#
C$RDBU= 000007      8#
C$REFG= 000047      8#  1211  1216  1235  1239  1242
C$RESE= 000033      8#
C$REVI= 000003      8#    18
C$RFLA= 000021      8#
C$RPT = 000025      8#
C$SEFG= 000046      8#
C$SPRI= 000041      8#  1205  1292  1384
C$SVEC= 000037      8#  1291  1350  1382  2850  2874  2899  2912  2936  2963  2977  2990  3064
                3082    3096  3203  3236  3256  3298  3450  3525  3681  3895
C$TPRI= 000013      8#
C10MS   011247    813#
C5SEC   011310    817#  2061  2219
C500MS  011260    814#  1934
DANAM   006130    662#  2687
DATACM= 000001     92#
DCKERR= 004000    128#  1030  1054
DCLIM = 000012     67#
DCLIMW  014214   1175#
DESDIF  003106    456#  1703* 1717* 1726* 1733  1796  2645
DESHD   003112    458#  1741  1804  1965* 1968* 1977  1979* 2324  2645  2648  2689  3425* 3462
                3464    3538  3540  3694  3696  3705  3707* 3738  3747  3749* 3792  3820  3822*
DESSEC  003114    459#  2648
DESSGN  003110    457#  1702* 1713* 1716* 1722* 1738  1801  2645  3549  3551* 3962
DIAGMC= 000000       8
DIFAUG  003076    452#  1689* 1697* 1708* 1718  1726
DIFWD   010055    751#  2645
DIRBIT= 000004    144#  1740  1803
DIRMSK  002314    251#  1275* 1283*
DLTERR= 010000    126#  1034
DLYCNT  003142    502#  1319* 1391* 1412* 1424* 1479* 1480* 1481* 1534* 1552* 1589* 1942* 2215*
                2850* 2874* 2899* 2912* 2936* 2963* 2977* 2990* 3064* 3082* 3096* 3203* 3236*
                3256* 3298* 3450* 3525* 3581* 3582* 3583* 3585* 3601* 3602* 3603* 3605* 3681*
                3895*
DONE    003006    418#  1417* 1472  1495  1570* 1577  1647* 1659  1747* 1754  1808* 1813  2034*
                2045    3196* 3205  3245* 3248  3575* 3589
```

```
DRDYMS= 000001          135#  1315  1366  1388  1539  1586  1928  1939  2047  2055  2201  2212  2985
                       3002  3039  3302  3432  3445  3505  3520  3661  3676
DRMSG   037732         3996  4014#
DRSB  = 000010           58#  3996
DRSELT= 000004           72#  1208  3120  3170
DRSET = 000010          149#  1513  1529  1581  2624  3199  3239  3578
DRTYPE  037710         3998  4012#
DRVCNT  003074          451#  1220*  1245  1249*  1252*
DRVERR= 040000          124#  1531  1549  1607  2948  3354
DRVNAM  006051          658#  1139  1324  1358  1369  1449  2685  2743  2783  2830  2837  3004  3041
                       3048  3104  3122  3306  3316  3569
DSESTA= 000400          165#  3259
DSMSK = 001400          131#
DSPCOD  014216 G       1179#
EF.CON= 000036 G         51#  1239
EF.NEW= 000035 G         51#  1242
EF.PWR= 000034 G         51#  1211
EF.RES= 000037 G         51#  1235
EF.STA= 000040 G         51#  1216
ERHEAD  003012          420#  2608  2742*  2781*  2841*  2880*  2884*  2917*  2925*  2954*  2993*  3035*  3128*
                       3182*  3295*  3384*  3415*  3486*  3643*  3716*  3765*  3841*  3923*
ERLIM = 000010           66#  4031
ERLIMQ  040133         4031  4043#
ERLIMW  014212         1174#  1444  1448
ERRCNT  003156          510#  1222  1227  1248  1949*  2225*
ERRPOI  003154          509#   910*   922*   936*   950*   965*   982*  1085*  1099*  1121*  1137*  1227*  1248*
                       1253*  1444
ERRSWI  003016          422#  1528*  1616*  1629  1631  1688*  1759*  1764*  1770  1772  1788*  1822*  1828
                       1830  1924*  1950*  1956  1958  2024*  2057*  2075*  2088  2090  2198*  2226*  2232
                       2234  2265  2267  2313*  2343*  2348*  2360  2362  3101*  3117*  3149*  3267*  3904*
                       3970*
ERRVEC  003140          501#  1330*  1350  1373  1382  1397
ERR1    012340 G        907#  1557  1596  1615  1758  1821  2074  2342  3253  3757
ERR10   013752 G       1133#  3799  3805
ERR2    012406 G        921#  2796  2801  3361  3468  3544  3700  3742
ERR3    012454 G        935#  2208  2754  2758  2762  2882  3262  3338  3466  3542  3698  3740
ERR4    012522 G        949#  2902  3348  3397  3434  3507  3663
ERR5    012572 G        964#  1935  2051  2062  2220  2889  2915  2932  2946  2951  2996  3453  3529
                       3685
ERR6    012642 G        979#  1609  1763  1948  2066  2224  2347  3370  3458  3533  3690
ERR7    013542 G       1084#  2766  2860  2869  2877  2923  2966  2980  3071  3085  3099  3442  3517
                       3672  3891  3900
ERR8    013612 G       1098#  3968
ERR9    013706 G       1120#
EVL   = 000004 G         51#
EXTOS   031312         3119  3121  3161#
E$END = 002100           8#
E$LOAD= 000035           8#    18
FBSFIL  003566          522#
FMTOP1  011316          819#  2743  2783  2830  2837  3004  3041  3048  3104  3122  3306  3316
FMTOP2  011345          820#  3226
FMTOP3  011367          821#  3140
FMT1    011410          822#  2621  2675
FMT1.1  011415          823#  2662
FMT11   011634          832#  2665
FMT12   011642          833#  2668
```

```
FMT13   011650              834#  2645
FMT14   011714              835#  1140
FMT15   011746              836#  1067  1142
FMT16   012002              837#  2604
FMT17   012013              838#  1053
FMT18   012035              839#
FMT19   012067              840#
FMT2    011424              824#  2637  2641
FMT20   012124              841#
FMT21   012154              842#
FMT22   012177              843#  2648
FMT23   012233              844#
FMT24   012247              845#  1323  1357  1368
FMT25   012254              846#  1448
FMT26   012264              847#
FMT27   012310              848#  1076
FMT28   012327              849#  1020
FMT3    011427              825#  1326  1360  1371  1450
FMT4    011432              826#  2608  3156
FMT5    011443              827#  1139  1324  1358  1369  1449  2685  3569
FMT6    011463              828#  2687
FMT7    011525              829#  2689
FMT8    011575              830#  2688
FMT9    011627              831#  2603  2821  3030  3179  3210  3269  3292  3412  3483  3570  3571  3619
                           3640  3838
FOLWRT= 000100               97#   107
FRMWD   010106              756#  2645
FWDSKO= 002000              101#   107
FWDSKS= 000400               99#   107
F$AU  = 000015                8#
F$AUTO= 000020                8#  1348  1374
F$BGN - 000040                8#    16    23    49   174   220   623   631   850   859   907   921   935
                            949   964   979  1084  1098  1120  1133  1147  1150  1159  1161  1177  1179
                           1186  1189  1198  1199  1336  1348  1379  1380  1400  1404  1410  1423  1429
                           1438  2705  2731  2737  2770  2776  2803  2808  2813  2861  2870  2883  2890
                           2903  2916  2924  2933  2947  2952  2967  2981  3013  3018  3023  3034  3072
                           3086  3103  3111  3116  3146  3151  3162  3167  3172  3232  3254  3263  3268
                           3275  3280  3285  3339  3349  3371  3378  3383  3400  3405  3414  3435  3443
                           3454  3459  3467  3471  3476  3485  3495  3508  3518  3530  3534  3543  3548
                           3554  3559  3567  3573  3611  3613  3626  3628  3633  3642  3651  3664  3673
                           3686  3691  3699  3704  3710  3715  3724  3741  3746  3759  3764  3772  3819
                           3825  3830  3840  3856  3905  3906  3917  3922  3934  3971  3972  3983  3984
                           3987  3988  4018  4022  4023  4047
F$CLEA= 000007                8#  1380  1398
F$DU  = 000016                8#  1400  1402
F$END = 000041                8#    16    23    49   174   220   623   631   850   859   919   933   947
                            962   977  1082  1096  1118  1132  1146  1147  1150  1159  1161  1177  1179
                           1186  1198  1334  1336  1374  1379  1398  1402  1404  1419  1425  1431  1438
                           2705  2731  2737  2770  2776  2803  2808  2813  2861  2870  2883  2890  2903
                           2916  2924  2933  2947  2952  2967  2981  3013  3018  3023  3034  3072  3086
                           3103  3111  3116  3146  3151  3162  3167  3172  3232  3254  3263  3268  3275
                           3280  3285  3339  3349  3371  3378  3383  3400  3405  3414  3435  3443  3454
                           3459  3467  3471  3476  3485  3495  3508  3518  3530  3534  3543  3548  3554
                           3559  3567  3573  3611  3613  3626  3628  3633  3642  3651  3664  3673  3686
                           3691  3699  3704  3710  3715  3724  3741  3746  3759  3764  3772  3819  3825
                           3830  3840  3856  3905  3906  3917  3922  3934  3971  3972  3983  3984  3987
```

```
                    4002    4018    4022    4033    4047
F$HARD= 000004       8#     3988    4002
F$HW  = 000013       8#     1151    1158
F$INIT= 000006       8#     1199    1334
F$JMP = 000050       8#     2813    2861    2870    2883    2890    2903    2916    2924    2933    2947    2952    2967
                    2981    3023    3072    3086    3172    3254    3263    3285    3339    3349    3371    3414    3435
                    3443    3454    3459    3467    3485    3508    3518    3530    3534    3543    3567    3642    3664
                    3673    3686    3691    3699    3741    3840
F$MOD = 000000       8#      16      23      49     174     220     623     631     850     859    1147    1150    1159
                    1161    1177    1179    1186    1198    1336    1379    1404    1438    2705    2731    3984    3987
                    4018    4022    4047
F$MSG = 000011       8#     907     919     921     933     935     947     949     962     964     977     979    1082
                    1084    1096    1098    1118    1120    1132    1133    1146
F$PROT= 000021       8#     1189    1193
F$PWR = 000017       8#
F$RPT = 000012       8#
F$SEG = 000003       8#
F$SOFT= 000005       8#     4023    4033
F$SRV = 000010       8#     1410    1419    1423    1425    1429    1431
F$SUB = 000002       8#     3034    3103    3146    3151    3232    3268    3495    3548    3573    3611    3613    3626
                    3651    3704    3724    3746    3772    3819    3856    3905    3934    3971
F$SW  = 000014       8#     1162    1176
F$TEST= 000001       8#     2737    2770    2776    2803    2808    3013    3018    3111    3116    3162    3167    3275
                    3280    3378    3383    3400    3405    3471    3476    3554    3559    3628    3633    3710    3715
                    3759    3764    3825    3830    3917    3922    3983
GBND    002310      249#    1273*   1281*
GDRSTA  017652     1639#    3258    3886    3897
GETPOS  022662     1691     2244#   3857    3935    3958
GETSTA= 000003      148#     995    1513    1516    1566    1645    3199    3239    3578
GLBDAT  002224 G    220#
GLBEQA  002224 G     49#
GLBERR  012340 G    859#
GLBSUB  016126 G   1438#
GLBTXT  005240 G    631#
GSTAT   016560     1518#    1537    1593    1599    1926    1937    2053    2199    2210    3430    3437    3503    3512
                    3659    3667
GSTATC  016544     1515#    2749    2843    2864    2892    2905    2919    2928    2939    2956    2970    2983    3052
                    3075    3089    3130    3147    3310    3324
GSTATG  016570     1514     1517    1520#
GSTATR  016530     1512#    2767    2791    2826    3000    3037    3300    3387    3422    3497    3615    3653    3726
                    3774    3844    3926
GSTER1  006462      679#    3182    3253
GTSTAT= 000104       81#    1573    1651    3197    3238    3577
G$CNTO= 000200       8#
G$DELM= 000372       8#      999    1319    1391    1534    1552    1576    1589    1598    1753    1817    1930    1942
                    2044    2205    2215    3247    3584    3604
G$DISP= 000003       8#
G$EXCP= 000400       8#
G$HILI= 000002       8#
G$LOLI= 000001       8#
G$NO  = 000000       8#     2745    2786    2834    2854    3008    3043    3068    3106    3125    3143    3158    3228
                    3271    3320    3621
G$OFFS= 000400       8#     2745    2786    2834    2854    3008    3043    3068    3106    3125    3143    3158    3228
                    3271    3320    3621    3990    3992    3994    3996    3998    4000    4025    4027    4029    4031
G$OFSI= 000376       8#     2745    2786    2834    2854    3008    3043    3068    3106    3125    3143    3158    3228
                    3271    3320    3621    3990    3992    3994    3996    3998    4000    4025    4027    4029    4031
```

```
G$PRMA= 000001            8#  3992  3994
G$PRMD= 000002            8#  3996  4000  4031
G$PRML= 000000            8#  2745  2786  2834  2854  3008  3043  3068  3106  3125  3143  3158  3228
                        3271  3320  3621  3990  3998  4025  4027  4029
G$RADA= 000140            8#
G$RADB= 000000            8#
G$RADD= 000040            8#  4031
G$RADL= 000120            8#  2745  2786  2834  2854  3008  3043  3068  3106  3125  3143  3158  3228
                        3271  3320  3621  3990  3998  4025  4027  4029
G$RADO= 000020            8#  3992  3994  3996  4000
G$XFER= 000004            8#
G$YES = 000010            8#  3990  3992  3994  3996  3998  4000  4025  4027  4029  4031
HADONE  003010          419#  1229*  3564  3568*
HAMES1  007130          731#  3570
HAMES2  007213          732#  3571
HCESTA= 040000          171#  1543
HCRCER= 004000          129#  1017  1024
HDALIG= 000010           73#  1208  3560
HDCYL   002316          252#  1276*  1284*
HDHSEL= 000100          157#  3791  3794
HDMOVF  007005          728#
HDRCMP= 000002           91#  3784
HDR40 = 100000          106#  2315  2638
HDSEC = 000077          156#  3811  3814
HDSEL = 000020          145#  1743  1806  2326  3593  3596
HDWD    010070          753#  2645  2648  2687
HDWRD1  003052          439#  1105  2173  2254  3750  3753*  3754
HDWRD2  003054          440#  2068
HDWRD3  003056          441#
HEAD  = 000006           65#
HEADLM= 010000           74#  1966  1975
HEADW   014210         1173#  1968
HICYL = 020000           75#  1288
HILIM = 000004           64#
HILIMW  014206         1172#  1290*
HLMTW   002302          246#  1272*  1280*  1290  1694  1696  1698  1720  3867  3945
HNFERR= 010000          127#  1015
HOE   = 100000 G         51#
HOSTAT- 000020          162#  1541  2930  3331
HPTCOD  014162 G       1150#
HRDPRM  037572 G       3987#
HRDWTS  024636 G       2731#
HSMSK = 000100          140#
HSSTAT= 000100          164#  2177  3344  3393
IBE   = 010000 G         51#
IBUFF   003762          524#  1062  2316  3786
IDU   = 000040 G         51#
IER   = 020000 G         51#
INITCO  014266 G       1198#
INITST  006513          680#  3295  3384
INOUTS= 000020           95#   107
INTEBL= 000100          133#
INTHLR  016052 G       1291  1410#
ISR     000100 G         51#
IXE   = 004000 G         51#
I$AU  - 000041            8#
```

```
I$AUTO= 000041          8#  1348#  1374#
I$CLN = 000041          8#  1380#  1398#
I$DU  = 000041          8#  1400#  1402#
I$HRD = 000041       3988#  4002#
I$INIT= 000041          8#  1199#  1334#
I$MOD = 000041          8#    16#    23#    49#   174#   220#   623#   631#   850#   859#  1147#  1150#  1159#
                     1161#  1177#  1179#  1186#  1198#  1336#  1379#  1404#  1438#  2705#  2731#  3984#  3987#
                     4018#  4022#  4047#
I$MSG = 000041          8#   907#   919#   921#   933#   935#   947#   949#   962#   964#   977#   979#  1082#
                     1084#  1096#  1098#  1118#  1120#  1132#  1133#  1146#
I$PROT= 000040          8#  1189#
I$PTAB= 000041          8#
I$PWR = 000041          8#
I$RPT = 000041          8#
I$SEG = 000041          8#  2737   2776   2808   3018   3034   3116   3146   3167   3232   3280   3383   3405
                     3476   3495   3559   3573   3613   3633   3651   3715   3724   3764   3772   3830   3856
                     3922   3934
I$SETU= 000041          8#
I$SFT = 000041       4023#  4033#
I$SRV = 000041          8#  1410#  1419#  1423#  1425#  1429#  1431#
I$SUB = 000041          8#  2737   2776   2808   3018   3034#  3072   3086   3103#  3116   3146#  3151#  3167
                     3232#  3254   3263   3268#  3280   3383   3405   3476   3495#  3508   3518   3530   3534
                     3543   3548#  3559   3573#  3611#  3613#  3626#  3633   3651#  3664   3675   3686   3691
                     3699   3704#  3715   3724#  3741   3746#  3764   3772#  3819#  3830   3856#  3905#  3922
                     3934#  3971#
I$TST = 000041          8#  2737#  2770#  2776#  2803#  2808#  2813   2861   2870   2883   2890   2903   2916
                     2924   2933   2947   2952   2967   2981   3013#  3018#  3023   3034   3111#  3116#  3146
                     3162#  3167#  3172   3232   3275#  3280#  3285   3339   3349   3371   3378#  3383#  3400#
                     3405#  3414   3435   3443   3454   3459   3467   3471#  3476#  3485   3495   3554#  3559#
                     3567   3573   3613   3628#  3633#  3642   3651   3710#  3715#  3724   3759#  3764#  3772
                     3825#  3830#  3840   3856   3906   3917#  3922#  3934   3972   3983#
JJJ     002300        245#
J$JMP = 000167          8#
LAB     014442       1230#
LAB1    006142        664#  2688
LAB2    006155        665#  2689
LCLEXT  032466       3211   3274#
LOCERR  003362        513#
LOCYL = 040000         76#  1230
LOE   = 040000 G       51#
LOLIM = 000002         63#
LOLIMW  014204       1171#  1232*
LOT   = 000010 G       51#
LPTOS   031072       3135#  3153
L$ACP   002110 G       18#
L$APT   002036 G       18#
L$AUT   002070 G       18#
L$AUTO  015322 G       18   1348#
L$CCP   002106 G       18#
L$CLEA  015660 G       18   1380#
L$CO    002032 G       18#
L$DEPO  002011 G       18#
L$DESC  002122 G       18     25#
L$DESP  002076 G       18#
L$DEVP  002060 G       18#
L$DISP  014220 G       18   1181#
```

```
L$DLY   002116 G          18#   999   1319   1391   1484   1534   1552   1576   1589   1598   1753   1817   1930
                         1942  2044   2205   2215   3247   3584   3604
L$DTP   002040 G          18#
L$DTYP  002034 G          18#
L$DU    016046 G          18   1400#
L$DUT   002072 G          18#
L$DVTY  002212 G          18     27#
L$EF    002052 G          18#
L$ENVI  002044 G          18#
L$ETP   002102 G          18#
L$EXP1  002046 G          18#
L$EXP4  002064 G          18#
L$EXP5  002066 G          18#
L$HARD  037574 G          18   3988#
L$HIME  002120 G          18#
L$HPCP  002016 G          18#
L$HPTP  002022 G          18#
L$HW    014164 G          18   1151#
L$ICP   002104 G          18#
L$INIT  014266 G          18   1199#
L$LADP  002026 G          18#
L$LAST  040162 G          18   4049#  4052#
L$LOAD  002100 G          18#
L$LUN   002074 G          18#
L$MREV  002050 G          18#
L$NAME  002000 G          18#
L$PRIO  002042 G          18#
L$PROT  014260 G          18   1189#
L$PRT   002112 G          18#
L$REPP  002062 G          18#
L$REV   002010 G          18#
L$SOFT  037750 G          18   4023#
L$SPC   002056 G          18#
L$SPCP  002020 G          18#
L$SPTP  002024 G          18#
L$STA   002030 G          18#
L$SW    014202 G          18   1162#
L$TEST  002114 G          18#
L$TIML  002014 G          18#
L$UNIT  002012 G          18#   1213   1220   1249   3185   3188
L.BA    003036            432#  2322*  2328   2688
L.CS    003034            431#  1571*  1572*  1573*  1575   1619   1648*  1650*  1651*  1655   1663   1728   1791
                         2035   2078   2319*  2320*  2321*  2333   2610   2688   3197*  3198*  3201   3233*  3234
                         3238*  3246   3576*  3577*  3580   3597*  3598*  3600
L.DA    003040            433#  1566*  1568*  1574   1581   1645*  1653   2323*  2326*  2327   2624   2688   3199*
                         3200   3239*  3244   3578*  3579   3593*  3596*  3599
L.MP    003042            434#  1562   1641   2027   2688
L10000  012404            919#
L10001  012452            933#
L10002  012520            947#
L10003  012570            962#
L10004  012640            977#
L10005  013540           1082#
L10006  013610           1096#
L10007  013704           1118#
L10010  013750           1132#
```

```
L10011  014160          1146#
L10012  014200          1151    1158#
L10013  014216          1162    1176#
L10015  015320          1334#
L10016  015656          1374#
L10017  016044          1398#
L10020  016050          1402#
L10021  016110          1419#
L10022  016116          1425#
L10023  016124          1431#
L10024  025114          2770#
L10025  025322          2803#
L10026  027612          2813    2861    2870    2883    2890    2903    2916    2924    2933    2947    2952    2967    2981
                        3013#
L10027  030706          3023    3111#
L10030  030604          3072    3086    3103#
L10031  031312          3162#
L10032  031220          3151#
L10033  032466          3172    3275#
L10034  032412          3254    3263    3268#
L10035  033372          3285    3339    3349    3371    3378#
L10036  033460          3400#
L10037  034126          3414    3435    3443    3454    3459    3467    3471#
L10040  034616          3485    3554#
L10041  034576          3508    3518    3530    3534    3543    3548#
L10042  035370          3567    3628#
L10043  035266          3611#
L10044  035366          3626#
L10045  036054          3642    3710#
L10046  036034          3664    3673    3686    3691    3699    3704#
L10047  036274          3759#
L10050  036206          3741    3746#
L10051  036554          3825#
L10052  036532          3819#
L10053  037276          3840    3906    3917#
L10054  037242          3905#
L10055  037570          3972    3983#
L10056  037534          3971#
L10057  037654          3988    4002#
L10060  040004          4023    4033#
MANQ    040076          4029    4041#
MBADAD  005717           242     653#
MBADSF  005740           654#
MBHSTA  010451           776#   2757    2886    3337
MBSET0= 000001           143#   1744    1807    3593
MCERR   010271           255     765#
MCONHN  006271           670#   1493
MCOSTA  010436           775#   2753    2795
MCYLOC  010726           791#   1104    2666
MCYLUP  005443           233     645#
MDATCP  005325           231     638#
MDCRC   010313           767#   1032
MDHEDR  002000 G          16#
MDLT    010340           769#   1036
MDRDY   010260           764#   1933    2049    2207    2995    3428    3444    3501    3519    3657    3675
MDRERR  010402           256     771#   2950    3356
```

```
MDRRES  006216            668#  1477
MDRVST  010523            781#  1089
MDSERR  010506            270    780#  3261
MERRS   011141            797#  1020
MEXERS  011101            796#  1448
MFLERR  010667            258    789#
MFMTER  005771            655#
MFOLWR  005517            237    648#
MFWDSK  005574            239    650#
MFWSKO  005627            241    651#
MGTSTA  005313            225    637#
MHCERR  010607            264    785#
MHCRC   010303            766#  1022
MHDERR  010652            259    788#
MHDRCP  005344            232    639#
MHFCRC  010352            770#  1028
MHNF    010324            768#  1026
MHOSTA  010475            778#  2927  3333
MHSTA   010413            773#  3346  3395  3460  3536  3692  3736
MINOUT  005474            235    647#
MISTST  006371            676#  2742  2781  2884  2925  2993
MISWI =  000000             62#  4025  4027  4029
MISWIW  014202           1163#  1208* 1230  1288  1966  1975  2740  2779  2811  3021  3120  3170  3283
                         3340  3389  3560
MITEST= 100000             77#  1208
MNDRST  010733            261    792#  2663
MNEERR  010715            257    790#
MNOCLR  006305            671#
MNOINT  006236            669#  1498
MOPER   005412            642#  2621
MOPERR  010642            260    787#  1019
MORECE  003014            421#  1058* 1065  1068* 1073  1076  1135  1143* 1506* 3783*
MOUTIN  005453            236    646#
MPNAM   006135            663#  2687
MQUALS= 003760            107#
MREAD   005246            229    633#
MREADH  005257            227    634#
MRESKO  005663            240    652#
MREVSK  005541            238    649#
MRLFAL  011024            794#  1595
MRSLT   005421            643#  1020  1140  2662
MSEEK   005240            226    632#
MSPERR  010536            267    782#  2881
MSTERR  010571            266    784#
MTMBS   006016            656#
MTOSLO  006176            667#  1474
MULOAD  005432            234    644#
MUNDEF  010757            793#  1556
MVOLCK  010425            774#  2945  3352
MWDERR  010625            263    786#
MWGERR  010554            268    783#
MWLSTA  010464            777#  2761  2800  3360
MWORD   006170            666#  1067  1140  1142
MWRCHK  005267            224    635#
MWRITE  005301            228    636#
MWRSET  005376            230    641#
```

```
MWRTAB  011065          795#
M40HDR  005362          243   640#
NEWCYL  003102          454#  1112  1694  1696*  1697  1698*  1701*  1705  1707*  1708  1709*  1711  1720
                       1990  1991*  3416  3487  3644  3717  3853  3930
NOCLR = 000010          114#
NOCTLR  006730          691#  1357
NOERCT  003363          514#   908   980  1505*
NOIRPT= 000002          112#
NOOP  = 000100           87#
NOPWR   006056          659#  1323
NOTRDY  006757          692#  1368
NCTST   006662          690#  2818  2821  3027  3030  3176  3179  3289  3292  3409  3412  3480  3483
                       3637  3640  3835  3838
NSTACH  006414          677#  2841  2917  2954  3035
NXMERR= 020000          125#
NXTHL   002306          248#  1271* 1285*
NXTPAS  014510         1244#  1259  1261
OBUFF   004362          525#  1061  2744*  2745  2746  2785*  2786  2787  2833*  2834  2835  2853*  2854
                       2855  3007*  3008  3009  3042*  3043  3044  3067*  3068  3069  3105*  3106  3107
                       3124*  3125  3126  3142*  3143  3144  3157*  3158  3159  3227*  3228  3229  3270*
                       3271  3272  3319*  3320  3321  3620*  3621  3622
OLDCYL  003100          453#  1693*  1989  1990*  2645  3932
ONSWAP  021552         1988#  3863  3941
OPFLAG  003004          417#  1504*  1547*  1583  2315*  2609*  2614*  2617*  2620*  2628  2630  2638  2643
                       2646  2839*  3047*  3305*  3784*
OPIERR= 002000          130#  1013  1056  1475
OPMSGS  002224          223#  2621  2637  2641
OPR002  007317          733#  2745  2786  2834  3008  3043  3106  3125  3143  3158  3228  3271  3320
                       3621
OPR003  007344          73:#  2854  3068
OPR004  010040          750#
OPR1    007367          735#  2743
OPR1A   010011          747#  2743  2783  2830  2837  3004  3041  3048  3104  3122  3306  3316
OPR1B   010015          748#  3140  3226
OPR10   007673          743#  3210
OPR11   007741          744#  3269
OPR12   007772          746#  3619
OPR2    007445          736#  2783
OPR3    007477          737#  2837  3048  3306
OPR5    007513          738#  2830
OPR6    007555          739#  3004  3041  3104  3316
OPR7    007610          740#  3122
OPR8    007637          741#  3140  3156  3226
OPR9    007656          742#  3156
OUTINS= 000040           96#   107
O$APTS= 000000            8#    18
O$AU  = 000000            8#    18
O$BGNR= 000000            8#    18
O$BGNS= 000001            8#    14#   18
O$DU  = 000001            8#    14#   18
O$ERRT= 000000            8#    18
O$GNSW= 000001            8#    14#   18
O$POIN= 000001            8#    14#   18
O$SETU= 000000            8#    18   4049
PART1 = 000001 G          1#    17    675   729   772   810  1180  1296  1777
PART2 - ****** U         20    469    694  1183  1836  1995  2095  2272  2368
```

```
PASCNT  003150            507#
PASNEW  014516           1243   1247#
PASNUM  003356            511#  1221*  1247*  1301   2738   2777   2809   3019   3118   3168   3281   3342   3391
                         3562
PATTBL  002360            274#
PAT1    004762            274    527#
PAT10   005236            283    621#
PAT2    004764            275    528#
PAT3    005024            276    545#
PAT4    005064            277    562#
PAT5    005124            278    579#
PAT6    005132            279    583#
PAT7    005172            280    600#
PAT8    005174            281    602#
PAT9    005234            282    619#
PNT   = 001000 G           51#
POSHD0  022272           2174   2176#
POSHSB  022266           2175#  3424   3461   3492   3537   3693
POSHW1  022260           2173#  3737
PRI   = 002000 G           51#
PRIOR = 000004             56#  4000
PRI00 = 000000 G           51#
PRI01 = 000040 G           51#
PRI02 = 000100 G           51#
PRI03 = 000140 G           51#
PRI04 = 000200 G           51#
PRI05 = 000240 G           51#
PRI06 = 000300 G           51#
PRI07 = 000340 G           51#
PSETNM  003360            512#  1228*  1250*  1251*  1254   1327   1361   1372   1451   3184
PWCON   014766           1214   1240   1278   1288#
PWRFLG  003366            516#  1213*  1258   1260*  1308   1394   1396*
P2T01E  006643            688#  3841
P2T02E  006643            689#  3923
RDALHD  023010           2304#  3781
RDDATA= 000114             85#
RDHEAD= 000110             83#  2038
RDNOHR= 000116             86#
RDYCHK  021132           1745   1789   1915#  2032   2331
RDYWAI  022316           2189#  3731   3779   3902   3956
READRL  016266           1458#  1471   1492   1497   2339
RELDWT= 040000            105#  1547   1583
RESE3   011154            801#  1067   1140   1142   2671
RESE4   011160            802#  1067   1140   1142   2672
RESE5   011165            805#  2675
RESE6   011172            806#  1076
RESPAR  003062            445#  2660   2696   2701
RESTAR  014460           1217   1234#
RESTBL  002320            255#  1047
REVSKO= 001000            100#   107
REVSKS= 000200             98#   107
RLBA  = 000002            117#  1459   1751*  1811*  2042*  2328*
RLBAS   003026            427#  1139   1255   1293   1324   1353   1358   1369   1418   1449   2317   2685   2743
                         2783   2830   2837   3004   3041   3048   3104   3122   3194   3306   3316   3569
RLCS  - 000000            116#   998*  1000   1313*  1315   1354   1365*  1366   1385   1387*  1388   1490   1531
                         1586   1657   1752*  1812*  2329   2333*  2335   3201*  3225*  3264*
```

```
RLCSR = 000000          122#  1458  1469  1575*  1655*  2043*  3234*  3246*  3580*  3600*
RLDA  = 000004          118#   995* 1460  1574*  1653*  1750*  1810*  2041*  2327*  3200*  3244*  3579*  3599*
RLDRV   003032          429#   997  1139  1311  1324  1358  1363  1369  1387  1449  1571  1648  1730
                       1793  2036  2320  2685  2743  2783  2830  2837  3004  3041  3048  3104  3122
                       3132  3133  3139*  3140  3154*  3214  3226  3233  3306  3316  3564  3568  3569
                       3576  3597
RLMP  = 000006          119#  1004  1461  2069  2070  2318
RLVEC   003030          428#  1291  1393
RORWOP= 020000          104#  2609  2617  2620  2646
RPTOP   023356          912   924   938   952   967   987  1087  1102  1123  1138  2599#
RPTREM  024352          916   930   944   959   974  1050  1093  1114  1129  2685#
RPTRES  024144          915   929   943   958   973  1048  1092  1113  1128  2657#
RSTRT   014376         1221#  1236
SAMSK = 000077          139#
SBSFIL  003372          521#
SECWD   010074          754#  2648
SEEK  = 000106           82#  1729  1792  3598
SEEKOP= 010000          103#  2609  2614  2643
SELQ    040004         4025   4037#
SEQMES  010127          758#  2603
SETDON  014544         1233   1246  1251#
SFTPRM  037746 G       4022#
SGNWD   010063          752#  2645
SIMSEK  020622         1780#  3426  3499  3655  3728  3776
SPDERR  006426          678#  2880
SPDSTA= 004000          168#  1543  2878
SPTCOD  014200 G       1161#
SSINDX  003002          414#  1210*  1521  1525*  1624*  1680  1684*  1765*  1781  1785*  1824*  1916  1920*
                       1951*  2016  2020*  2083*  2190  2194*  2227*  2245  2249*  2261*  2305  2309*  2355*
                       2600  2606
STAMES  010152          759#  1053
STAMSK= 000007          160#  1580  1662
STATE2  011177          807#  2885
STATE3  011207          808#  2926
STATE5  011217          809#  2994
STOSTA= 010000          169#
SUBSTK  002404          287#  1523*  1524*  1682*  1683*  1783*  1784*  1918*  1919*  2018*  2019*  2192*  2193*
                       2247*  2248*  2307*  2308*  2604
SVCBGL= 000001           11#
SVCGBL= 000000            8#    16    18    25    27    49   220   631   859   907   921   935   949
                        964   979  1084  1098  1120  1133  1150  1151  1161  1162  1179  1181  1189
                       1198  1199  1348  1379  1380  1400  1410  1423  1429  1438  2731  3987  3988
                       4022  4023  4049#
SVCINS= 000000            8#    12#   18    25    27   919   933   947   962   977   999  1020  1053
                       1067  1076  1082  1096  1118  1132  1139  1140  1142  1146  1151  1162  1181
                       1202  1203  1205  1206  1207  1211  1212  1216  1217  1235  1236  1239  1240
                       1242  1243  1256  1257  1291  1292  1298  1299  1319  1323  1324  1326  1327
                       1328  1334  1350  1357  1358  1360  1361  1368  1369  1371  1372  1373  1374
                       1382  1384  1391  1393  1397  1398  1402  1419  1425  1431  1446  1447  1448
                       1449  1450  1451  1452  1534  1552  1557  1576  1589  1596  1598  1609  1615
                       1753  1758  1763  1817  1821  1930  1935  1942  1948  2044  2051  2062  2066
                       2074  2205  2208  2215  2220  2224  2342  2347  2603  2604  2608  2621  2637
                       2641  2645  2648  2662  2671  2672  2675  2685  2687  2688  2689  2743  2745
                       2754  2758  2762  2766  2770  2783  2786  2796  2801  2803  2813  2821  2830
                       2834  2837  2850  2854  2860  2861  2869  2870  2874  2877  2882  2883  2889
                       2890  2899  2902  2903  2912  2915  2916  2923  2924  2932  2933  2936  2946
```

```
                    2947  2951  2952  2963  2966  2967  2977  2980  2981  2990  2996  3004  3008
                    3013  3023  3030  3034  3041  3043  3048  3064  3068  3071  3072  3082  3085
                    3086  3096  3099  3103  3104  3106  3111  3122  3125  3140  3143  3146  3151
                    3156  3158  3162  3172  3179  3191  3192  3203  3210  3226  3228  3232  3236
                    3247  3253  3254  3256  3262  3263  3268  3269  3271  3275  3285  3292  3298
                    3306  3312  3316  3320  3326  3338  3339  3348  3349  3361  3370  3371  3378
                    3397  3400  3412  3414  3434  3435  3442  3443  3450  3453  3454  3458  3459
                    3466  3467  3468  3471  3483  3485  3495  3507  3508  3517  3518  3525  3529
                    3530  3533  3534  3542  3543  3544  3548  3554  3567  3569  3570  3571  3573
                    3584  3586  3604  3606  3611  3613  3619  3621  3626  3628  3640  3642  3651
                    3663  3664  3672  3673  3681  3685  3686  3690  3691  3698  3699  3700  3704
                    3710  3724  3740  3741  3742  3746  3757  3759  3772  3799  3805  3819  3825
                    3838  3840  3856  3859  3860  3891  3895  3900  3905  3906  3917  3934  3937
                    3938  3968  3971  3972  3983  3988  3990  3992  3994  3996  3998  4000  4002
                    4023  4025  4027  4029  4031  4033  4049
SVCSUB= 000001         8#    10#  3034  3146  3232  3495  3573  3613  3651  3724  3772  3856  3934
SVCTAG= 000000         8#    13#   919   933   947   962   977  1082  1096  1118  1132  1146  1158
                    1176  1334  1374  1398  1402  1419  1425  1431  2745  2770  2786  2803  2834
                    2854  3008  3013  3043  3068  3103  3106  3111  3125  3143  3151  3158  3162
                    3228  3268  3271  3275  3320  3378  3400  3471  3548  3554  3611  3621  3626
                    3628  3704  3710  3746  3759  3819  3825  3905  3917  3971  3983  4002  4033
SVCTST= 000001         8#     9#  2737  2776  2808  3018  3116  3167  3280  3383  3405  3476  3559
                    3633  3715  3764  3830  3922
SWAPHD  021512      1975#  3912  3978
S$LSYM= 010000         8#   919#  933#  947#  962#  977# 1082# 1096# 1118# 1132# 1146# 1158# 1176#
                    1334# 1374# 1398# 1402# 1419# 1425# 1431# 2745# 2770# 2786# 2803# 2834# 2854#
                    3008# 3013# 3043# 3068# 3103# 3106# 3111# 3125# 3143# 3151# 3158# 3162# 3228#
                    3268# 3271# 3275# 3320# 3378# 3400# 3471# 3548# 3554# 3611# 3621# 3626# 3628#
                    3704# 3710# 3746# 3759# 3819# 3825# 3905# 3917# 3971# 3983# 4002# 4033#
TBLSTR  003024       425#
TBT     002544       327#
TCERR   010237       761#  1076
TEMP0   003116       460#  3132*  3154  3750*  3752*  3754  3787  3842*  3907*  3910  3924*  3973*  3976
TEMP1   003120       461#  1676*  1678*  1748  3212*  3215  3226
TEMP2   003122       462#  3848*  3851*  3865*  3871*  3873*  3879*  3881*  3882  3929*  3943*  3947*  3951*
                    3952
TEMP3   003124       463#   990*  1005*  1006  1051  1053
TEMP4   003126       464#  1512  1513*  1515  1516*  1518  1519*  1529  1559  1568  1617  1628*  2012*
                    2014*  2025  2076
TEMP5   003130       465#
TEMP6   003132       466#
TEMP7   003134       467#
TEMP8   003136       468#
TOSLOW= 000001       111#
TRPFLG  003364       515#  1349*  1355  1430*
TRPHAN  016120 G    1350  1382  1429#
TSTINT  016512      1504#  2748  2790  2825  2999  3036  3129  3183  3296  3385  3421  3496  3574
                    3614  3652  3725  3773  3843  3925
TSTLAB  006363       674#  2608
TST4    027712      3026  3035#
TYPDR = 000006        57#  3998
T$ARGC= 000002        18#  1020#  1053#  1067#  1076#  1139#  1140#  1142#  1323#  1324#  1326#  1357#  1358#
                    1360# 1368# 1369# 1371# 1448# 1449# 1450# 2603# 2604# 2608# 2621# 2637# 2641#
                    2645# 2648# 2662# 2671# 2672# 2675# 2685# 2687# 2688# 2689# 2743# 2783# 2821#
                    2830# 2837# 3004# 3030# 3041# 3048# 3104# 3122# 3140# 3156# 3179# 3210# 3226#
                    3269# 3292# 3306# 3316# 3412# 3483# 3569# 3570# 3571# 3619# 3640# 3838#
```

```
T$CODE= 004052   2745#  2786#  2834#  2854#  3008#  3043#  3068#  3106#  3125#  3143#  3158#  3228#  3271#
                 3320#  3621#  3990#  3992#  3994#  3996#  3998#  4000#  4025#  4027#  4029#  4031#
T$ERRN= 000311      8#  1557#  1596#  1609#  1615#  1758#  1763#  1821#  1935#  1948#  2051#  2062#  2066#
                 2074#  2208#  2220#  2224#  2342#  2347#  2754#  2758#  2762#  2766#  2796#  2801#  2860#
                 2869#  2877#  2882#  2889#  2902#  2915#  2923#  2932#  2946#  2951#  2966#  2980#  2996#
                 3071#  3085#  3099#  3253#  3262#  3338#  3348#  3361#  3370#  3397#  3434#  3442#  3453#
                 3458#  3466#  3468#  3507#  3517#  3529#  3533#  3542#  3544#  3663#  3672#  3685#  3690#
                 3698#  3700#  3740#  3742#  3757#  3799#  3805#  3891#  3900#  3968#
T$EXCP= 000000   3992#  3994#  3996#  4000#  4031#
T$FLAG= 000040   2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#
                 3023#  3072#  3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#
                 3454#  3459#  3467#  3485#  3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#
                 3686#  3691#  3699#  3741#  3840#  3906#  3972#
T$GMAN= 000000      8#
T$HILI= 000377   3992#  3994#  3996#  4000#  4031#
T$LAST= 000001      8#  4049#
T$LOLI= 000000   3992#  3994#  3996#  4000#  4031#
T$LSYM= 010000      8#   919   933   947   962   977  1082  1096  1118  1132  1146  1158  1176
                 1334  1374  1398  1402  1419  1425  1431  2770  2803  3013  3103  3111  3151
                 3162  3268  3275  3378  3400  3471  3548  3554  3611  3626  3628  3704  3710
                 3746  3759  3819  3825  3905  3917  3971  3983  4002  4033
T$LTNO= 000020   4049#
T$NEST= 177777      8#    16#    23#    49#   174#   220#   623#   631#   850#   859#   907#   919#   921#
                  933#   935#   947#   949#   962#   964#   977#   979#  1082#  1084#  1096#  1098#  1118#
                 1120#  1132#  1133#  1146#  1147#  1150#  1151#  1158#  1159#  1161#  1162#  1176#  1177#
                 1179#  1186#  1189#  1193#  1198#  1199#  1334#  1336#  1348#  1374#  1379#  1380#  1398#
                 1400#  1402#  1404#  1410#  1419#  1423#  1425#  1429#  1431#  1438#  2705#  2731#  2737#
                 2770#  2776#  2803#  2808#  3013#  3018#  3034#  3103#  3111#  3116#  3146#  3151#  3162#
                 3167#  3232#  3268#  3275#  3280#  3378#  3383#  3400#  3405#  3471#  3476#  3495#  3548#
                 3554#  3559#  3573#  3611#  3613#  3626#  3628#  3633#  3651#  3704#  3710#  3715#  3724#
                 3746#  3759#  3764#  3772#  3819#  3825#  3830#  3856#  3905#  3917#  3922#  3934#  3971#
                 3983#  3984#  3987#  3988#  4002#  4018#  4022#  4023#  4033#  4047#
T$NS0 = 000000     16#    23     49#   174    220#   623    631#   850    859#  1147  1150#  1159  1161#
                 1177   1179#  1186   1189#  1193   1198#  1336   1348#  1374   1379#  1404   1410#  1419
                 1423#  1425   1429#  1431   1438#  2705   2731#  3984   3987#  4018   4022#  4047
T$NS1 = 000005    907#   919    921#   933    935#   947    949#   962    964#   977    979#  1082   1084#
                 1096   1098#  1118   1120#  1132   1133#  1146   1151#  1158   1162#  1176   1199#  1334
                 1380#  1398   1400#  1402   2737#  2770   2776#  2803   2808#  3013   3018#  3111   3116#
                 3162   3167#  3275   3280#  3378   3383#  3400   3405#  3471   3476#  3554   3559#  3628
                 3633#  3710   3715#  3759   3764#  3825   3830#  3917   3922#  3983   3988#  4002   4023#
                 4033
T$NS2 = 000002   3034#  3103   3146#  3151   3232#  3268   3495#  3548   3573#  3611   3613#  3626   3651#
                 3704   3724#  3746   3772#  3819   3856#  3905   3934#  3971
T$PTNU= 000000      8#
T$SAVL= 177777      8#
T$SEGL= 177777      8#
T$SUBN= 000001      8#  2737#  2776#  2808#  3018#  3034#  3116#  3146#  3167#  3232#  3280#  3383#  3405#
                 3476#  3495#  3559#  3573#  3613#  3633#  3651#  3715#  3724#  3764#  3772#  3830#  3856#
                 3922#  3934#
T$TAGL= 177777      8#
T$TAGN= 010061      8#   907#   921#   935#   949#   964#   979#  1084#  1098#  1120#  1133#  1151#  1162#
                 1189#  1199#  1348#  1380#  1400#  1410#  1423#  1429#  2737#  2776#  2808#  3018#  3034#
                 3116#  3146#  3167#  3232#  3280#  3383#  3405#  3476#  3495#  3559#  3573#  3613#  3633#
                 3651#  3715#  3724#  3764#  3772#  3830#  3856#  3922#  3934#  3988#  4023#
T$TEMP- 000000     23#   174#   623#   850#   919#   933#   947#   962#   977#  1082#  1096#  1118#  1132#
                 1146#  1147#  1158#  1159#  1176#  1177#  1181#  1186#  1193#  1334#  1336#  1374#  1398#
```

```
                    1402#  1404#  1419#  1425#  1431#  2705#  2745#  2770#  2786#  2803#  2813#  2834#  2854#
                    2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3008#
                    3013#  3023#  3043#  3068#  3072#  3086#  3103#  3106#  3111#  3125#  3143#  3151#  3158#
                    3162#  3172#  3228#  3254#  3263#  3268#  3271#  3275#  3285#  3320#  3339#  3349#  3371#
                    3378#  3400#  3414#  3435#  3443#  3454#  3459#  3467#  3471#  3485#  3508#  3518#  3530#
                    3534#  3543#  3548#  3554#  3567#  3611#  3621#  3626#  3628#  3642#  3664#  3673#  3686#
                    3691#  3699#  3704#  3710#  3741#  3746#  3759#  3819#  3825#  3840#  3905#  3906#  3917#
                    3971#  3972#  3983#  3984#  3990#  3992#  3994#  3996#  3998#  4000#  4002#  4018#  4025#
                    4027#  4029#  4031#  4033#  4047#
T$TEST= 000020         8#  2737#  2776#  2808#  3018#  3034   3116#  3146   3167#  3232   3280#  3383#  3405#
                    3476#  3495   3559#  3573   3613   3633#  3651   3715#  3724   3764#  3772   3830#  3856
                    3922#  3934   4049
T$TSTM= 177777         8#   919    933    947    962    977   1020   1053   1067   1076   1082   1096   1118
                    1132   1139   1140   1142   1146   1202   1205   1206   1211   1216   1235   1239   1242
                    1256   1291   1292   1298   1323   1324   1326   1327   1328   1334   1350   1357   1358
                    1360   1361   1368   1369   1371   1372   1373   1374   1382   1384   1393   1397   1398
                    1402   1446   1448   1449   1450   1451   1452   1557   1596   1609   1615   1758   1763
                    1821   1935   1948   2051   2062   2066   2074   2208   2220   2224   2342   2347   2603
                    2604   2608   2621   2637   2641   2645   2648   2662   2671   2672   2675   2685   2687
                    2688   2689   2743   2745   2754   2758   2762   2766   2770   2783   2786   2796   2801
                    2803   2813   2821   2830   2834   2837   2850   2854   2860   2861   2869   2870   2874
                    2877   2882   2883   2889   2890   2899   2902   2903   2912   2915   2916   2923   2924
                    2932   2933   2936   2946   2947   2951   2952   2963   2966   2967   2977   2980   2981
                    2990   2996   3004   3008   3013   3023   3030   3034   3041   3043   3048   3064   3068
                    3071   3072   3082   3085   3086   3096   3099   3103   3104   3106   3111   3122   3125
                    3140   3143   3146   3151   3156   3158   3162   3172   3179   3191   3203   3210   3226
                    3228   3232   3236   3253   3254   3256   3262   3263   3268   3269   3271   3275   3285
                    3292   3298   3306   3312   3316   3320   3326   3338   3339   3348   3349   3361   3370
                    3371   3378   3397   3400   3412   3414   3434   3435   3442   3443   3450   3453   3454
                    3458   3459   3466   3467   3468   3471   3483   3485   3495   3507   3508   3517   3518
                    3525   3529   3530   3533   3534   3542   3543   3544   3548   3554   3567   3569   3570
                    3571   3573   3586   3606   3611   3613   3619   3621   3626   3628   3640   3642   3651
                    3663   3664   3672   3673   3681   3685   3686   3690   3691   3698   3699   3700   3704
                    3710   3724   3740   3741   3742   3746   3757   3759   3772   3799   3805   3819   3825
                    3838   3840   3856   3859   3891   3895   3900   3905   3906   3917   3934   3937   3968
                    3971   3972   3983
T$TSTS= 000001         8#  2737#  2776#  2808#  3018#  3116#  3167#  3280#  3383#  3405#  3476#  3559#  3633#
                    3715#  3764#  3830#  3922#
T$$AUT= 010016      1348#  1374
T$$CLE= 010017      1380#  1398
T$$DU = 010020      1400#  1402
T$$HAR= 010057      3988#  4002
T$$HW = 010012      1151#  1158
T$$INI= 010015      1199#  1334
T$$MSG= 010011       907#   919    921#   933    935#   947    949#   962    964#   977    979#   1082   1084#
                    1096   1098#  1118   1120#  1132   1133#  1146
T$$PRO= 010014      1189#
T$$SOF= 010060      4023#  4033
T$$SRV= 010023      1410#  1419   1423#  1425   1429#  1431
T$$SUB= 010056      3034#  3072   3086   3103   3146#  3151   3232#  3254   3263   3268   3495#  3508   3518
                    3530   3534   3543   3548   3573#  3611   3613#  3626   3651#  3664   3673   3686   3691
                    3699   3704   3724#  3741   3746   3772#  3819   3856#  3905   3934#  3971
T$$SW = 010013      1162#  1176
T$$TES= 010055      2737#  2770   2776#  2803   2808#  2813   2861   2870   2883   2890   2903   2916   2924
                    2933   2947   2952   2967   2981   3013   3018#  3023   3111   3116#  3162   3167#  3172
                    3275   3280#  3285   3339   3349   3371   3378   3383#  3400   3405#  3414   3435   3443
```

```
                    3454    3459    3467    3471    3476#   3485    3554    3559#   3567    3628    3633#   3642    3710
                    3715#   3759    3764#   3825    3830#   3840    3906    3917    3922#   3972    3983
T.BA    003046       436#   1414*   1459*   2689
T.CS    003044       435#    991    1010    1054    1056    1413*   1458*   1475    1539    1549    1601    1607    1761
                    1928    1939    1946    2047    2055    2064    2201    2212    2222    2345    2689    2948    2985
                    3002    3039    3207    3302    3354    3372    3432    3445    3456    3505    3520    3531    3661
                    3676    3688
T.DA    003050       437#   1415*   1460*   2689
T.DRIV  002276       244#   1266*   1268    1699    1724    3363    3846    3869    3877
T.MP    003052       438#   1008    1416*   1461*   1541    1543    1579    1603    1661    2175    2689    2751    2755
                    2759    2793    2798    2878    2887    2894    2907    2930    2941    3259    3330    3393    3594
                    3617
T.STAT  003060       443#   1090    1579*   1580*   1661*   1662*   2763    2828    2845    2858    2866    2921    2958
                    2972    3054    3056    3077    3091    3313    3327    3440    3515    3670    3889    3898
T05ERR  006526       681#   3128
T09ERR  006541       682#   3415
T1      024636 G    1181    2737#
T10     034130 G    1181    3476#
T10ERR  006551       683#   3486
T10.1   034240      3495#
T104$   034240      3494#   3552
T11     034620 G    1181    3559#
T11.1   034776      3573#
T11.2   035270      3613#
T115$   035270      3566    3612#
T12     035372 G    1181    3633#
T12ERR  006571       684#   3643
T12.1   035476      3651#
T124$   035476      3650#   3708
T13     036056 G    1181    3715#
T13ERR  006603       685#   3716
T13.1   036102      3724#
T134$   036102      3723#   3751
T14     036276 G    1181    3764#
T14ERR  006617       686#   3765
T14.1   036320      3772#
T15     036556 G    1181    3830#
T15.1   035724      3856#
T153$   036320      3771#   3823
T16     037300 G    1181    3922#
T16ERR  006633       687#
T16.1   037354      3934#
T172$   036724      3855#   3914
T1765$  037276      3845    3916#
T1865$  037570      3927    3982#
T187$   037354      3933#   3980
T2      025116 G    1181    2776#
T25TBL  002430       299#
T25TB2  002456       312#
T3      025324 G    1181    2808#
T33TBL  002504       326#
T365$   027612      2827    2844    2865    2893    2906    2920    2929    2940    2957    2971    2984    3001    3012#
T4      027614 G    1181    3018#
T4.1    027710      3034#
T465$   030576      3038    3053    3076    3090    3101#
T5      030710 G    1181    3116#
```

```
T5.1     031202            3146#
T504$    031234            3131    3155#
T6       031314 G          1181    3167#
T6.1     032040            3232#
T7       032470 G          1181    3280#
T8       033374 G          1181    3383#
T9       033462 G          1181    3405#
UAM    = 000200 G            51#
ULOAD  = 000010              94#   3047    3305
UNDTST   010025             749#   3140
UNXERR   006346             673#   1611
VCNRST   006325             672#   1605
VCSTAT=  001000             166#   1603    2941    3350
VECMSG   037670            3994    4008#
VECT   = 000002              55#   3994
WAITIN   016320            1467#   1613    1669    1756    1819    2072    2340    3250
WCMSK  = 017777             152#
WCRNG  = 160000             153#
WDESTA=  100000             172#   1543
WGESTA=  002000             167#
WLSTAT=  020000             170#   2759    2798    3358    3594    3617
WRTSWI   003022             424#
WTDATA=  000112              84#
XRDHD    021606            2014#   2252
XRDHDC   021576            2012#   3734
XRDHDG   021612            2013    2015#
XSEEK    020032            1678#   3884    3953
XSEEKT   020022            1676#
XSEEK1   020036            1677    1679#
X$ALWA=  000000               8#
X$FALS=  000040               8#
X$OFFS=  000400               8#
X$TRUE=  000020               8#
.      = 040162               5#    326#    327#    510#    521#    522#    524#    525#    999   1319   1391   1487   1489
                  1534    1552    1576    1589    1598    1753    1817    1930    1942   2044   2205   2215   2813
                  2861    2870    2883    2890    2903    2916    2924    2933    2947   2952   2967   2981   3023
                  3072    3086    3172    3247    3254    3263    3285    3339    3349   3371   3414   3435   3443
                  3454    3459    3467    3485    3508    3518    3530    3534    3543   3567   3584   3604   3642
                  3664    3673    3686    3691    3699    3741    3840    3906    3972   4020#  4045#
```

```
BCOMPL  1207  1236  1240  1243  1257  1447
BGNAUT  1348
BGNCLN  1380
BGNDU   1400
BGNHRD  3988
BGNHW   1151
BGNINI  1199
BGNMOD    16    49   220   631   859  1150  1161  1179  1198  1379  1438  2731  3987  4022
BGNMSG   907   921   935   949   964   979  1084  1098  1120  1133
BGNPRO  1189
BGNSFT  4023
BGNSRV  1410  1423  1429
BGNSUB  3034  3146  3232  3495  3573  3613  3651  3724  3772  3856  3934
BGNSW   1162
BGNTST  2737  2776  2808  3018  3116  3167  3280  3383  3405  3476  3559  3633  3715  3764  3830
        3922
BNCOMP  1203  1212  1217  1299  3192  3860  3938
BREAK   3312  3326  3586  3606
CLOCK   1202
CLRVEC  1373  1393  1397
DELAY    999  1319  1391  1534  1552  1576  1589  1598  1753  1817  1930  1942  2044  2205  2215
        3247  3584  3604
DESCRI    25
DEVTYP    27
DISPAT  1181
DOCLN   1328  1452
DODU    1327  1361  1372  1451
ENDAUT  1374
ENDCLN  1398
ENDDU   1402
ENDHRD  4002
ENDHW   1158
ENDINI  1334
ENDMOD    23   174   623   850  1147  1159  1177  1186  1336  1404  2705  3984  4018  4047
ENDMSG   919   933   947   962   977  1082  1096  1118  1132  1146
ENDPRO  1193
ENDSFT  4033
ENDSRV  1419  1425  1431
ENDSUB  3103  3151  3268  3548  3611  3626  3704  3746  3819  3905  3971
ENDSW   1176
ENDTST  2770  2803  3013  3111  3162  3275  3378  3400  3471  3554  3628  3710  3759  3825  3917
        3983
EQUALS    51
ERRHRD  1557  1596  1609  1615  1758  1763  1821  1935  1948  2051  2062  2066  2074  2208  2220
        2224  2342  2347  2754  2758  2762  2766  2796  2801  2860  2869  2877  2882  2889  2902
        2915  2923  2932  2946  2951  2966  2980  2996  3071  3085  3099  3253  3262  3338  3348
        3361  3370  3397  3434  3442  3453  3458  3466  3468  3507  3517  3529  3533  3542  3544
        3663  3672  3685  3690  3698  3700  3740  3742  3757  3799  3805  3891  3900  3968
ESCAPE  3906  3972
EXIT    2813  2861  2870  2883  2890  2903  2916  2924  2933  2947  2952  2967  2981  3023  3072
        3086  3172  3254  3263  3285  3339  3349  3371  3414  3435  3443  3454  3459  3467  3485
        3508  3518  3530  3534  3543  3567  3642  3664  3673  3686  3691  3699  3741  3840
GMANIL  2745  2786  2834  2854  3008  3043  3068  3106  3125  3143  3158  3228  3271  3320  3621
GPHARD  1256  3191
GPRMA   3992  3994
GPRMD   3996  4000  4031
```

```
GPRML    2745#  2786#  2834#  2854#  3008#  3043#  3068#  3106#  3125#  3143#  3158#  3228#  3271#  3320#  3621#
         3990   3998   4025   4027   4029
HEADER     18
INLOOP   1446   3859   3937
LASTAD   4049
MANUAL   1206   1298
M$BYTE     18#
M$CHEC   2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3023#  3072#
         3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#  3454#  3459#  3467#  3485#
         3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#  3686#  3691#  3699#  3741#  3840#
M$CNTO   2745#  2786#  2834#  2854#  3008#  3043#  3068#  3106#  3125#  3143#  3158#  3228#  3271#  3320#  3621#
         3990#  3992#  3994#  3996#  3998#  4000#  4025#  4027#  4029#  4031#
M$COUN   1020#  1053#  1067#  1076#  1139#  1140#  1142#  1323#  1324#  1326#  1357#  1358#  1360#  1368#  1369#
         1371#  1448#  1449#  1450#  2603#  2604#  2608#  2621#  2637#  2641#  2645#  2648#  2662#  2671#  2672#
         2675#  2685#  2687#  2688#  2689#  2743#  2783#  2821#  2830#  2837#  3004#  3030#  3041#  3048#  3104#
         3122#  3140#  3156#  3179#  3210#  3226#  3269#  3292#  3306#  3316#  3412#  3483#  3569#  3570#  3571#
         3619#  3640#  3838#
M$DATA     18#    25#    27#
M$DECR     23#   174#   623#   850#   919#   933#   947#   962#   977#  1082#  1096#  1118#  1132#  1146#  1147#
         1158#  1159#  1176#  1177#  1186#  1193#  1334#  1336#  1374#  1398#  1402#  1404#  1419#  1425#  1431#
         2705#  2770#  2803#  3013#  3103#  3111#  3151#  3162#  3268#  3275#  3378#  3400#  3471#  3548#  3554#
         3611#  3626#  3628#  3704#  3710#  3746#  3759#  3819#  3825#  3905#  3917#  3971#  3983#  3984#  4002#
         4018#  4033#  4047#
M$DEFA   2745#  2786#  2834#  2854#  3008#  3043#  3068#  3106#  3125#  3143#  3158#  3228#  3271#  3320#  3621#
         3990#  3992#  3994#  3996#  3998#  4000#  4025#  4027#  4029#  4031#
M$ENDE     23#   174#   623#   850#   919#   933#   947#   962#   977#  1082#  1096#  1118#  1132#  1146#  1147#
         1158#  1159#  1176#  1177#  1186#  1334#  1336#  1374#  1398#  1402#  1404#  1419#  1425#  1431#  2705#
         2770#  2803#  3013#  3103#  3111#  3151#  3162#  3268#  3275#  3378#  3400#  3471#  3548#  3554#  3611#
         3626#  3628#  3704#  3710#  3746#  3759#  3819#  3825#  3905#  3917#  3971#  3983#  3984#  4002#  4018#
         4033#  4047#
M$ERRI   1557#  1596#  1609#  1615#  1758#  1763#  1821#  1935#  1948#  2051#  2062#  2066#  2074#  2208#  2220#
         2224#  2342#  2347#  2754#  2758#  2762#  2766#  2796#  2801#  2860#  2869#  2877#  2882#  2889#  2902#
         2915#  2923#  2932#  2946#  2951#  2966#  2980#  2996#  3071#  3085#  3099#  3253#  3262#  3338#  3348#
         3361#  3370#  3397#  3434#  3442#  3453#  3458#  3466#  3468#  3507#  3517#  3529#  3533#  3542#  3544#
         3663#  3672#  3685#  3690#  3698#  3700#  3740#  3742#  3757#  3799#  3805#  3891#  3900#  3968#
M$ESCA   3906#  3972#
M$ESCS   3906#  3972#
M$EXCP   3992#  3994#  3996#  4000#  4031#
M$EXIT   2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3023#  3072#
         3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#  3454#  3459#  3467#  3485#
         3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#  3686#  3691#  3699#  3741#  3840#
M$EXSE   2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3023#  3072#
         3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#  3454#  3459#  3467#  3485#
         3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#  3686#  3691#  3699#  3741#  3840#
M$EXTJ   2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3023#  3072#
         3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#  3454#  3459#  3467#  3485#
         3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#  3686#  3691#  3699#  3741#  3840#
M$GEN      16#    18#    25#    27#    49#   220#   631#   859#   907#   919#   921#   933#   935#   947#   949#
          962#   964#   977#   979#  1082#  1084#  1096#  1098#  1118#  1120#  1132#  1133#  1146#  1150#  1151#
         1158#  1161#  1162#  1176#  1179#  1181#  1189#  1198#  1199#  1334#  1348#  1374#  1379#  1380#  1398#
         1400#  1402#  1410#  1419#  1423#  1425#  1429#  1431#  1438#  2731#  2737#  2745#  2770#  2776#  2786#
         2803#  2808#  2834#  2854#  3008#  3013#  3018#  3034#  3043#  3068#  3103#  3106#  3111#  3116#  3125#
         3143#  3146#  3151#  3158#  3162#  3167#  3228#  3232#  3268#  3271#  3275#  3280#  3320#  3378#  3383#
         3400#  3405#  3471#  3476#  3495#  3548#  3554#  3559#  3573#  3611#  3613#  3621#  3626#  3628#  3633#
         3651#  3704#  3710#  3715#  3724#  3746#  3759#  3764#  3772#  3819#  3825#  3830#  3856#  3905#  3917#
         3922#  3934#  3971#  3983#  3987#  3988#  4002#  4022#  4023#  4033#  4049#
```

M$GENB 2745# 2786# 2834# 2854# 3008# 3043# 3068# 3106# 3125# 3143# 3158# 3228# 3271# 3320# 3621#
M$GETS 23# 174# 623# 850# 919# 933# 947# 962# 977# 1082# 1096# 1118# 1132# 1146# 1147#
1158# 1159# 1176# 1177# 1186# 1193# 1334# 1336# 1374# 1398# 1402# 1404# 1419# 1425# 1431#
2705# 2770# 2803# 3013# 3103# 3111# 3151# 3162# 3268# 3275# 3378# 3400# 3471# 3548# 3554#
3611# 3626# 3628# 3704# 3710# 3746# 3759# 3819# 3825# 3905# 3917# 3971# 3983# 3984# 4002#
4018# 4033# 4047#
M$GETT 2813# 2861# 2870# 2883# 2890# 2903# 2916# 2924# 2933# 2947# 2952# 2967# 2981# 3023# 3072#
3086# 3172# 3254# 3263# 3285# 3339# 3349# 3371# 3414# 3435# 3443# 3454# 3459# 3467# 3485#
3508# 3518# 3530# 3534# 3543# 3567# 3642# 3664# 3673# 3686# 3691# 3699# 3741# 3840# 3906#
3972#
M$GNGB 16# 18# 25# 27# 49# 220# 631# 859# 907# 921# 935# 949# 964# 979# 1084#
1098# 1120# 1133# 1150# 1151# 1161# 1162# 1179# 1181# 1189# 1198# 1199# 1348# 1379# 1380#
1400# 1410# 1423# 1429# 1438# 2731# 3987# 3988# 4022# 4023# 4049#
M$GNIN 18# 25# 27# 919# 933# 947# 962# 977# 999# 1020# 1053# 1067# 1076# 1082# 1096#
1118# 1132# 1139# 1140# 1142# 1146# 1151# 1162# 1181# 1202# 1203# 1205# 1206# 1207# 1211#
1212# 1216# 1217# 1235# 1236# 1239# 1240# 1242# 1243# 1256# 1257# 1291# 1292# 1298# 1299#
1319# 1323# 1324# 1326# 1327# 1328# 1334# 1350# 1357# 1358# 1360# 1361# 1368# 1369# 1371#
1372# 1373# 1374# 1382# 1384# 1391# 1393# 1397# 1398# 1402# 1419# 1425# 1431# 1446# 1447#
1448# 1449# 1450# 1451# 1452# 1534# 1552# 1557# 1576# 1589# 1596# 1598# 1609# 1615# 1753#
1758# 1763# 1817# 1821# 1930# 1935# 1942# 1948# 2044# 2051# 2062# 2066# 2074# 2205# 2208#
2215# 2220# 2224# 2342# 2347# 2603# 2604# 2608# 2621# 2637# 2641# 2645# 2648# 2662# 2671#
2672# 2675# 2685# 2687# 2688# 2689# 2743# 2745# 2754# 2758# 2762# 2766# 2770# 2783# 2786#
2796# 2801# 2803# 2813# 2821# 2830# 2834# 2837# 2850# 2854# 2860# 2861# 2869# 2870# 2874#
2877# 2882# 2883# 2889# 2890# 2899# 2902# 2903# 2912# 2915# 2916# 2923# 2924# 2932# 2933#
2936# 2946# 2947# 2951# 2952# 2963# 2966# 2967# 2977# 2980# 2981# 2990# 2996# 3004# 3008#
3013# 3023# 3030# 3034# 3041# 3043# 3048# 3064# 3068# 3071# 3072# 3082# 3085# 3086# 3096#
3099# 3103# 3104# 3106# 3111# 3122# 3125# 3140# 3143# 3146# 3151# 3156# 3158# 3162# 3172#
3179# 3191# 3192# 3203# 3210# 3226# 3228# 3232# 3236# 3247# 3253# 3254# 3256# 3262# 3263#
3268# 3269# 3271# 3275# 3285# 3292# 3298# 3306# 3312# 3316# 3320# 3326# 3338# 3339# 3348#
3349# 3361# 3370# 3371# 3378# 3397# 3400# 3412# 3414# 3434# 3435# 3442# 3443# 3450# 3453#
3454# 3458# 3459# 3466# 3467# 3468# 3471# 3483# 3485# 3495# 3507# 3508# 3517# 3518# 3525#
3529# 3530# 3533# 3534# 3542# 3543# 3544# 3548# 3554# 3567# 3569# 3570# 3571# 3573# 3584#
3586# 3604# 3606# 3611# 3613# 3619# 3621# 3626# 3628# 3640# 3642# 3651# 3663# 3664# 3672#
3673# 3681# 3685# 3686# 3690# 3691# 3698# 3699# 3700# 3704# 3710# 3724# 3740# 3741# 3742#
3746# 3757# 3759# 3772# 3799# 3805# 3819# 3825# 3838# 3840# 3856# 3859# 3860# 3891# 3895#
3900# 3905# 3906# 3917# 3934# 3937# 3938# 3968# 3971# 3972# 3983# 3988# 3990# 3992# 3994#
3996# 3998# 4000# 4002# 4023# 4025# 4027# 4029# 4031# 4033# 4049#
M$GNLS 2745# 2786# 2834# 2854# 3008# 3043# 3068# 3106# 3125# 3143# 3158# 3228# 3271# 3320# 3621#
M$GNSU 3034# 3146# 3232# 3495# 3573# 3613# 3651# 3724# 3772# 3856# 3934#
M$GNTA 919# 933# 947# 962# 977# 1082# 1096# 1118# 1132# 1146# 1158# 1176# 1334# 1374# 1398#
1402# 1419# 1425# 1431# 2770# 2803# 3013# 3103# 3111# 3151# 3162# 3268# 3275# 3378# 3400#
3471# 3548# 3554# 3611# 3626# 3628# 3704# 3710# 3746# 3759# 3819# 3825# 3905# 3917# 3971#
3983# 4002# 4033#
M$GNTE 2737# 2776# 2808# 3018# 3116# 3167# 3280# 3383# 3405# 3476# 3559# 3633# 3715# 3764# 3830#
3922#
M$HAPT 18#
M$HNAP 18#
M$INCR 16# 49# 220# 631# 859# 907# 919# 921# 933# 935# 947# 949# 962# 964# 977#
979# 1020# 1053# 1067# 1076# 1082# 1084# 1096# 1098# 1118# 1120# 1132# 1133# 1139# 1140#
1142# 1146# 1150# 1151# 1161# 1162# 1179# 1189# 1198# 1199# 1202# 1205# 1206# 1211# 1216#
1235# 1239# 1242# 1256# 1291# 1292# 1298# 1323# 1324# 1326# 1327# 1328# 1334# 1348# 1350#
1357# 1358# 1360# 1361# 1368# 1369# 1371# 1372# 1373# 1374# 1379# 1380# 1382# 1384# 1393#
1397# 1398# 1400# 1402# 1410# 1423# 1429# 1438# 1446# 1448# 1449# 1450# 1451# 1452# 1557#
1596# 1609# 1615# 1758# 1763# 1821# 1935# 1948# 2051# 2062# 2066# 2074# 2208# 2220# 2224#
2342# 2347# 2603# 2604# 2608# 2621# 2637# 2641# 2645# 2648# 2662# 2671# 2672# 2675# 2685#
2687# 2688# 2689# 2731# 2737# 2743# 2745# 2754# 2758# 2762# 2766# 2770# 2776# 2783# 2786#

	2796#	2801#	2803#	2808#	2813#	2821#	2830#	2834#	2837#	2850#	2854#	2860#	2861#	2869#	2870#
	2874#	2877#	2882#	2883#	2889#	2890#	2899#	2902#	2903#	2912#	2915#	2916#	2923#	2924#	2932#
	2933#	2936#	2946#	2947#	2951#	2952#	2963#	2966#	2967#	2977#	2980#	2981#	2990#	2996#	3004#
	3008#	3013#	3018#	3023#	3030#	3034#	3041#	3043#	3048#	3064#	3068#	3071#	3072#	3082#	3085#
	3086#	3096#	3099#	3103#	3104#	3106#	3111#	3116#	3122#	3125#	3140#	3143#	3146#	3151#	3156#
	3158#	3162#	3167#	3172#	3179#	3191#	3203#	3210#	3226#	3228#	3232#	3236#	3253#	3254#	3256#
	3262#	3263#	3268#	3269#	3271#	3275#	3280#	3285#	3292#	3298#	3306#	3312#	3316#	3320#	3326#
	3338#	3339#	3348#	3349#	3361#	3370#	3371#	3378#	3383#	3397#	3400#	3405#	3412#	3414#	3434#
	3435#	3442#	3443#	3450#	3453#	3454#	3458#	3459#	3466#	3467#	3468#	3471#	3476#	3483#	3485#
	3495#	3507#	3508#	3517#	3518#	3525#	3529#	3530#	3533#	3534#	3542#	3543#	3544#	3548#	3554#
	3559#	3567#	3569#	3570#	3571#	3573#	3586#	3606#	3611#	3613#	3619#	3621#	3626#	3628#	3633#
	3640#	3642#	3651#	3663#	3664#	3672#	3673#	3681#	3685#	3686#	3690#	3691#	3698#	3699#	3700#
	3704#	3710#	3715#	3724#	3740#	3741#	3742#	3746#	3757#	3759#	3764#	3772#	3799#	3805#	3819#
	3825#	3830#	3838#	3840#	3856#	3859#	3891#	3895#	3900#	3905#	3906#	3917#	3922#	3934#	3937#
	3968#	3971#	3972#	3983#	3987#	3988#	4022#	4023#							
M$LDRO	1202#	1205#	1211#	1216#	1235#	1239#	1242#	1256#	1292#	1327#	1361#	1372#	1373#	1384#	1393#
	1397#	1451#	3191#												
M$MCHI	8#														
M$MCLO	8#														
M$POP	23#	174#	623#	850#	919#	933#	947#	962#	977#	1082#	1096#	1118#	1132#	1146#	1147#
	1158#	1159#	1176#	1177#	1186#	1193#	1334#	1336#	1374#	1398#	1402#	1404#	1419#	1425#	1431#
	2705#	2770#	2803#	3013#	3103#	3111#	3151#	3162#	3268#	3275#	3378#	3400#	3471#	3548#	3554#
	3611#	3626#	3628#	3704#	3710#	3746#	3759#	3819#	3825#	3905#	3917#	3971#	3983#	3984#	4002#
	4018#	4033#	4047#												
M$PRIN	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1323#	1324#	1326#	1357#	1358#	1360#	1368#	1369#
	1371#	1448#	1449#	1450#	2603#	2604#	2608#	2621#	2637#	2641#	2645#	2648#	2662#	2671#	2672#
	2675#	2685#	2687#	2688#	2689#	2743#	2783#	2821#	2830#	2837#	3004#	3030#	3041#	3048#	3104#
	3122#	3140#	3156#	3179#	3210#	3226#	3269#	3292#	3306#	3316#	3412#	3483#	3569#	3570#	3571#
	3619#	3640#	3838#												
M$PUSH	16#	49#	220#	631#	859#	907#	921#	935#	949#	964#	979#	1084#	1098#	1120#	1133#
	1150#	1151#	1161#	1162#	1179#	1189#	1198#	1199#	1348#	1379#	1380#	1400#	1410#	1423#	1429#
	1438#	2731#	2737#	2776#	2808#	3018#	3034#	3116#	3146#	3167#	3232#	3280#	3383#	3405#	3476#
	3495#	3559#	3573#	3613#	3633#	3651#	3715#	3724#	3764#	3772#	3830#	3856#	3922#	3934#	3987#
	3988#	4022#	4023#												
M$PUT	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1291#	1323#	1324#	1326#	1350#	1357#	1358#	1360#
	1368#	1369#	1371#	1382#	1448#	1449#	1450#	2603#	2604#	2608#	2621#	2637#	2641#	2645#	2648#
	2662#	2671#	2672#	2675#	2685#	2687#	2688#	2689#	2743#	2783#	2821#	2830#	2837#	2850#	2874#
	2899#	2912#	2936#	2963#	2977#	2990#	3004#	3030#	3041#	3048#	3064#	3082#	3096#	3104#	3122#
	3140#	3156#	3179#	3203#	3210#	3226#	3236#	3256#	3269#	3292#	3298#	3306#	3316#	3412#	3450#
	3483#	3525#	3569#	3570#	3571#	3619#	3640#	3681#	3838#	3895#					
M$PUT1	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1291#	1323#	1324#	1326#	1350#	1357#	1358#	1360#
	1368#	1369#	1371#	1382#	1448#	1449#	1450#	2603#	2604#	2608#	2621#	2637#	2641#	2645#	2648#
	2662#	2671#	2672#	2675#	2685#	2687#	2688#	2689#	2743#	2783#	2821#	2830#	2837#	2850#	2874#
	2899#	2912#	2936#	2963#	2977#	2990#	3004#	3030#	3041#	3048#	3064#	3082#	3096#	3104#	3122#
	3140#	3156#	3179#	3203#	3210#	3226#	3236#	3256#	3269#	3292#	3298#	3306#	3316#	3412#	3450#
	3483#	3525#	3569#	3570#	3571#	3619#	3640#	3681#	3838#	3895#					
M$RADI	2745#	2786#	2834#	2854#	3008#	3043#	3068#	3106#	3125#	3143#	3158#	3228#	3271#	3320#	3621#
	3990#	3992#	3994#	3996#	3998#	4000#	4025#	4027#	4029#	4031#					
M$RNRO	1202#	1256#	3191#												
M$SETS	16#	49#	220#	631#	859#	907#	921#	935#	949#	964#	979#	1084#	1098#	1120#	1133#
	1150#	1151#	1161#	1162#	1179#	1189#	1198#	1199#	1348#	1379#	1380#	1400#	1410#	1423#	1429#
	1438#	2731#	2737#	2776#	2808#	3018#	3034#	3116#	3146#	3167#	3232#	3280#	3383#	3405#	3476#
	3495#	3559#	3573#	3613#	3633#	3651#	3715#	3724#	3764#	3772#	3830#	3856#	3922#	3934#	3987#
	3988#	4022#	4023#												
M$SVC	919#	933#	947#	962#	977#	1020#	1053#	1067#	1076#	1082#	1096#	1118#	1132#	1139#	1140#
	1142#	1146#	1202#	1205#	1206#	1211#	1216#	1235#	1239#	1242#	1256#	1291#	1292#	1298#	1323#

```
                1324#  1326#  1327#  1328#  1334#  1350#  1357#  1358#  1360#  1361#  1368#  1369#  1371#  1372#  1373#
                1374#  1382#  1384#  1393#  1397#  1398#  1402#  1446#  1448#  1449#  1450#  1451#  1452#  1557   1596
                1609   1615   1758   1763   1821   1935   1948   2051   2062   2066   2074   2208   2220   2224   2342
                2347   2603#  2604#  2608#  2621#  2637#  2641#  2645#  2648#  2662#  2671#  2672#  2675#  2685#  2687#
                2688#  2689#  2743#  2745#  2754   2758   2762   2766   2770#  2783#  2786#  2796   2801   2803#  2813#
                2821#  2830#  2834#  2837#  2850#  2854#  2860   2861#  2869   2870#  2874#  2877   2882   2883#  2889
                2890#  2899#  2902   2903#  2912#  2915   2916#  2923   2924#  2932   2933#  2936#  2946   2947#  2951
                2952#  2963#  2966   2967#  2977#  2980   2981#  2990#  2996   3004#  3008#  3013#  3023#  3030#  3034#
                3041#  3043#  3048#  3064#  3068#  3071   3072#  3082#  3085   3086#  3096#  3099   3103#  3104#  3106#
                3111#  3122#  3125#  3140#  3143#  3146#  3151#  3156#  3158#  3162#  3172#  3179#  3191#  3203#  3210#
                3226#  3228#  3232#  3236#  3253   3254#  3256#  3262   3263#  3268#  3269#  3271#  3275#  3285#  3292#
                3298#  3306#  3312#  3316#  3320#  3326#  3338   3339#  3348   3349#  3361   3370   3371#  3378#  3397
                3400#  3412#  3414#  3434   3435#  3442   3443#  3450#  3453   3454#  3458   3459#  3466   3467#  3468
                3471#  3483#  3485#  3495#  3507   3508#  3517   3518#  3525#  3529   3530#  3533   3534#  3542   3543#
                3544   3548#  3554#  3567#  3569#  3570#  3571#  3573#  3586#  3606#  3611#  3613#  3619#  3621#  3626#
                3628#  3640#  3642#  3651#  3663   3664#  3672   3673#  3681#  3685   3686#  3690   3691#  3698   3699#
                3700   3704#  3710#  3724#  3740   3741#  3742   3746#  3757   3759#  3772#  3799   3805   3819#  3825#
                3838#  3840#  3856#  3859#  3891   3895#  3900   3905#  3906#  3917#  3934#  3937#  3968   3971#  3972#
                3983#
MSTLAB           919#   933#   947#   962#   977#  1020#  1053#  1067#  1076#  1082#  1096#  1118#  1132#  1139#  1140#
                1142#  1146#  1202#  1205#  1206#  1211#  1216#  1235#  1239#  1242#  1256#  1291#  1292#  1298#  1323#
                1324#  1326#  1327#  1328#  1334#  1350#  1357#  1358#  1360#  1361#  1368#  1369#  1371#  1372#  1373#
                1374#  1382#  1384#  1393#  1397#  1398#  1402#  1446#  1448#  1449#  1450#  1451#  1452#  1557#  1596#
                1609#  1615#  1758#  1763#  1821#  1935#  1948#  2051#  2062#  2066#  2074#  2208#  2220#  2224#  2342#
                2347#  2603#  2604#  2608#  2621#  2637#  2641#  2645#  2648#  2662#  2671#  2672#  2675#  2685#  2687#
                2688#  2689#  2743#  2745#  2754#  2758#  2762#  2766#  2770#  2783#  2786#  2796#  2801#  2803#  2813#
                2821#  2830#  2834#  2837#  2850#  2854#  2860#  2861#  2869#  2870#  2874#  2877#  2882#  2883#  2889#
                2890#  2899#  2902#  2903#  2912#  2915#  2916#  2923#  2924#  2932#  2933#  2936#  2946#  2947#  2951#
                2952#  2963#  2966#  2967#  2977#  2980#  2981#  2990#  2996#  3004#  3008#  3013#  3023#  3030#  3034#
                3041#  3043#  3048#  3064#  3068#  3071#  3072#  3082#  3085#  3086#  3096#  3099#  3103#  3104#  3106#
                3111#  3122#  3125#  3140#  3143#  3146#  3151#  3156#  3158#  3162#  3172#  3179#  3191#  3203#  3210#
                3226#  3228#  3232#  3236#  3253#  3254#  3256#  3262#  3263#  3268#  3269#  3271#  3275#  3285#  3292#
                3298#  3306#  3312#  3316#  3320#  3326#  3338#  3339#  3348#  3349#  3361#  3370#  3371#  3378#  3397#
                3400#  3412#  3414#  3434#  3435#  3442#  3443#  3450#  3453#  3454#  3458#  3459#  3466#  3467#  3468#
                3471#  3483#  3485#  3495#  3507#  3508#  3517#  3518#  3525#  3529#  3530#  3533#  3534#  3542#  3543#
                3544#  3548#  3554#  3567#  3569#  3570#  3571#  3573#  3586#  3606#  3611#  3613#  3619#  3621#  3626#
                3628#  3640#  3642#  3651#  3663#  3664#  3672#  3673#  3681#  3685#  3686#  3690#  3691#  3698#  3699#
                3700#  3704#  3710#  3724#  3740#  3741#  3742#  3746#  3757#  3759#  3772#  3799#  3805#  3819#  3825#
                3838#  3840#  3856#  3859#  3891#  3895#  3900#  3905#  3906#  3917#  3934#  3937#  3968#  3971#  3972#
                3983#
M$TSTL           919#   933#   947#   962#   977#  1020#  1053#  1067#  1076#  1082#  1096#  1118#  1132#  1139#  1140#
                1142#  1146#  1202#  1205#  1206#  1211#  1216#  1235#  1239#  1242#  1256#  1291#  1292#  1298#  1323#
                1324#  1326#  1327#  1328#  1334#  1350#  1357#  1358#  1360#  1361#  1368#  1369#  1371#  1372#  1373#
                1374#  1382#  1384#  1393#  1397#  1398#  1402#  1446#  1448#  1449#  1450#  1451#  1452#  1557#  1596#
                1609#  1615#  1758#  1763#  1821#  1935#  1948#  2051#  2062#  2066#  2074#  2208#  2220#  2224#  2342#
                2347#  2603#  2604#  2608#  2621#  2637#  2641#  2645#  2648#  2662#  2671#  2672#  2675#  2685#  2687#
                2688#  2689#  2743#  2745#  2754#  2758#  2762#  2766#  2770#  2783#  2786#  2796#  2801#  2803#  2813#
                2821#  2830#  2834#  2837#  2850#  2854#  2860#  2861#  2869#  2870#  2874#  2877#  2882#  2883#  2889#
                2890#  2899#  2902#  2903#  2912#  2915#  2916#  2923#  2924#  2932#  2933#  2936#  2946#  2947#  2951#
                2952#  2963#  2966#  2967#  2977#  2980#  2981#  2990#  2996#  3004#  3008#  3013#  3023#  3030#  3034#
                3041#  3043#  3048#  3064#  3068#  3071#  3072#  3082#  3085#  3086#  3096#  3099#  3103#  3104#  3106#
                3111#  3122#  3125#  3140#  3143#  3146#  3151#  3156#  3158#  3162#  3172#  3179#  3191#  3203#  3210#
                3226#  3228#  3232#  3236#  3253#  3254#  3256#  3262#  3263#  3268#  3269#  3271#  3275#  3285#  3292#
                3298#  3306#  3312#  3316#  3320#  3326#  3338#  3339#  3348#  3349#  3361#  3370#  3371#  3378#  3397#
                3400#  3412#  3414#  3434#  3435#  3442#  3443#  3450#  3453#  3454#  3458#  3459#  3466#  3467#  3468#
                3471#  3483#  3485#  3495#  3507#  3508#  3517#  3518#  3525#  3529#  3530#  3533#  3534#  3542#  3543#
```

```
            3544#  3548#  3554#  3567#  3569#  3570#  3571#  3573#  3586#  3606#  3611#  3613#  3619#  3621#  3626#
            3628#  3640#  3642#  3651#  3663#  3664#  3672#  3673#  3681#  3685#  3686#  3690#  3691#  3698#  3699#
            3700#  3704#  3710#  3724#  3740#  3741#  3742#  3746#  3757#  3759#  3772#  3799#  3805#  3819#  3825#
            3838#  3840#  3856#  3859#  3891#  3895#  3900#  3905#  3906#  3917#  3934#  3937#  3968#  3971#  3972#
            3983#
M$WORD       18#  1181#  1557#  1596#  1609#  1615#  1758#  1763#  1821#  1935#  1948#  2051#  2062#  2066#  2074#
            2208#  2220#  2224#  2342#  2347#  2745#  2754#  2758#  2762#  2766#  2786#  2796#  2801#  2813#  2834#
            2854#  2860#  2861#  2869#  2870#  2877#  2882#  2883#  2889#  2890#  2902#  2903#  2915#  2916#  2923#
            2924#  2932#  2933#  2946#  2947#  2951#  2952#  2966#  2967#  2980#  2981#  2996#  3008#  3023#  3043#
            3068#  3071#  3072#  3085#  3086#  3099#  3106#  3125#  3143#  3158#  3172#  3228#  3253#  3254#  3262#
            3263#  3271#  3285#  3320#  3338#  3339#  3348#  3349#  3361#  3370#  3371#  3397#  3414#  3434#  3435#
            3442#  3443#  3453#  3454#  3458#  3459#  3466#  3467#  3468#  3485#  3507#  3508#  3517#  3518#  3529#
            3530#  3533#  3534#  3542#  3543#  3544#  3567#  3621#  3642#  3663#  3664#  3672#  3673#  3685#  3686#
            3690#  3691#  3698#  3699#  3700#  3740#  3741#  3742#  3757#  3799#  3805#  3840#  3891#  3900#  3968#
            3990#  3992#  3994#  3996#  3998#  4000#  4025#  4027#  4029#  4031#  4049
POINTE       14
PRINTB     1020   1053   1067   1076   1139   1140   1142   2603   2604   2608   2621   2637   2641   2645   2648
            2662   2671   2672   2675   2685   2687   2688   2689
PRINTF     1323   1324   1326   1357   1358   1360   1368   1369   1371   1448   1449   1450   2743   2783   2821
            2830   2837   3004   3030   3041   3048   3104   3122   3140   3156   3179   3210   3226   3269   3292
            3306   3316   3412   3483   3569   3570   3571   3619   3640   3838
READEF     1211   1216   1235   1239   1242
SETPRI     1205   1292   1384
SETVEC     1291   1350   1382   2850   2874   2899   2912   2936   2963   2977   2990   3064   3082   3096   3203
            3236   3256   3298   3450   3525   3681   3895
SVC          6#     8
TIMDLY      202#  2850   2874   2899   2912   2936   2963   2977   2990   3064   3082   3096   3203   3236   3256
            3298   3450   3525   3681   3895
WAITMS       184#  1319   1391   1534   1552   1589   1942   2215
WAITUS       196#   999   1576   1598   1753   1817   1930   2044   2205   3247
XFER        2813#  2861#  2870#  2883#  2890#  2903#  2916#  2924#  2933#  2947#  2952#  2967#  2981#  3023#  3072#
            3086#  3172#  3254#  3263#  3285#  3339#  3349#  3371#  3414#  3435#  3443#  3454#  3459#  3467#  3485#
            3508#  3518#  3530#  3534#  3543#  3567#  3642#  3664#  3673#  3686#  3691#  3699#  3741#  3840#

. ABS.  040162      000

 ERRORS DETECTED:  0

 ,CZRLIC.LST/CRF=SVC33/ML,CZRLIC.MAC
 RUN-TIME: 149 152 14 SECONDS
 RUN-TIME RATIO: 608/317=1.9
 CORE USED:  17K  (33 PAGES)
```