

RL11,RLV11

RL01/02 DRIVE TEST1
CZRLIB0

AH-F118B-MC
FICHE 1 OF 1

MAR 1980
COPYRIGHT 77 80
MADE IN USA



A large grid of data tables, likely representing test results or drive logs. The grid consists of approximately 12 columns and 20 rows of smaller tables. Each cell in the grid contains a small table with multiple columns and rows of data, which is too small to read clearly. The overall layout is a dense matrix of data points.



IDENTIFICATION

PRODUCT CODE: AC-F119B-MC
PRODUCT NAME: CZRLIB0 RL01/02 DRIVE TEST 1
DATE CREATED: 5-JAN-79
REVISED: 7-DEC-79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHORS: D. DEKNIS, C. CAMPBELL

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1979, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	CHAIN MODE OPERATION
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
3.1	ERROR REPORTING
3.1.1	SPECIFIC OPERATION MESSAGES
3.1.2	SPECIFIC RESULT MESSAGES
3.1.3	OTHER MESSAGES
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION1.1 PROGRAM ABSTRACT1.1.1 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

CZRLIB0 RL01/02 DRIVE TEST 1
(FORMERLY CZRLCB)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 DISK SUBSYSTEM USER'S GUIDE (EK-RL01-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)
CZRLGBO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLHBO	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:

```
CHMDKAO 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 CZRLIB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. D APR-79	D
CZRLI-B-0	D
CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC	D
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                                0
DR>CON/FLAGS:HOE:IER:LOE=0        D,0
CHANGE SW (L) ? N                 D,0
CZRLI EOP 1                       D
^C
DR>RESTART/PASS:1                 D,0
CHANGE SW (L) ? N                 D,0
-----
-----
-----
```

2.2 CHAIN MODE OPERATION

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

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

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

TO EXECUTED 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:

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
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 ERROR WAS ENCOUNTERED
WITH THE HOE FLAG SET SET

START
RESTART
CONTINUE
PROCEED
PRINT
DISPLAY
FLAGS
ZFLAGS
EXIT

2.3.2 COMMAND SYNTAX

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

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

'TEST-LIST' IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

'PASS-CNT' IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING TEST EXECUTION. 'FLAG-LIST' IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR

IER INHIBIT ERROR REPORTING
IBE INHIBIT BASIC ERROR REPORTS
IXE INHIBIT EXTENDED ERROR REPORTS
PRI DIRECT ALL MESSAGES TO A LINE PRINTER
PNT PRINT NUMBER OF TEST BEING EXECUTED
BOE BELL ON ERROR
UAM RUN IN UNATTENDED 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 COMMAND 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

 PRJ(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 RLO1'S AND THE LAST 4 DRIVES ARE RLO2'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 0

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RLO1 (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 = RLO1 (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 RLO1'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

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

2.5 HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

CHANGE S.W. ?

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

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 BEING 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 CONTROLLER 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 NOT 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 RLO2 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.

STANDARD TESTS

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

TEST 7 INITIAL STATE TEST

(P-CLOCK REQUIRED)

DO GET STATUS, WAIT FOR INTERRUPT.

IF OPI OCCURS:

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

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

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

DRIVE CYCLE UP DID NOT SELECT HEAD 0

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

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

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

CHECK ALL ERROR BITS ARE 0.

CHECK STATE IS 5. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD

TEST 8 INITIAL RESET STATE TEST

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

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

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

HEAD SELECTED BIT SHOULD STILL BE ZERO. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
HEAD SELECT SHIFT REGISTER NOT LOADING

TEST 9 DRIVE READY TEST

(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)

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

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

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

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

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

FAILURE IN READY LATCH OR INTEGRATOR

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

FAILURE IN INTEGRATOR
UNEXPECTED GUARD BAND DETECTED

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

REPEAT TEST USING HEAD 1.

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

TEST 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.

a

47	BIT AND OFFSET DEFINITIONS
179	MACRO DEFINITIONS
218	GLOBAL DATA AND CONSTANTS
626	GLOBAL MESSAGES
857	ERROR MESSAGES
1196	INITIALIZATION CODE
1338	AUTO DROP SECTION
1408	INTERRUPT SERVICE ROUTINES
1435	GLOBAL SUBROUTINES
2680	*TEST 1 BASIC INTERFACE (PART 1)
2718	*TEST 2 BASIC INTERFACE (PART 2)
2747	*TEST 3 HEAD LOADING
2930	*TEST 4 HEAD UNLOADING
3020	*TEST 5 DRIVE SELECT
3070	*TEST 6 DRIVE SELECT TEST
3172	*TEST 7 INITIAL STATE
3250	*TEST 8 INITIAL RESET STATE
3273	*TEST 9 DRIVE READY
3340	*TEST 10 SEEK SIGN SWITCH
3421	*TEST 11 HEAD ALIGNMENT SUPPORT
3495	*TEST 12 HEAD SWITCHING
3575	*TEST 13 READ HEADER (PART 1)
3622	*TEST 14 READ HEADER (PART 2)
3686	*TEST 15 DIFFERENCE OF 1 SEEK (PART 1)
3774	*TEST 16 DIFFERENCE OF 1 SEEK (PART 2)
3839	PARAMETER CODING

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,3GNSFT,BGNDU
15			
16	002000	BGNMOD	MDHEDR
17	002000	HEADER	CZRLI,B,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	102	.ASCII /B/
(4)	002011	060	.ASCII /O/
(4)	002012	000000	.WORD 0
(4)	002014	000001	.WORD 1
(4)	002016	037270	.WORD L\$HARD
(4)	002020	037444	.WORD L\$SOFT
(4)	002022	014170	.WORD L\$HW
(4)	002024	014206	.WORD L\$SW
(4)	002026	037656	.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	014224	.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	016052	.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	014272				.WORD	LSINIT
(4)	002106	015664				.WORD	LSCLEAN
(4)	002110	015326				.WORD	LSAUTO
(4)	002112	014264				.WORD	LSPROT
(4)	002114	000000				.WORD	0
(4)	002116	000000				.WORD	0
(4)	002120	000000				.WORD	0

23 002122
24
25

ENDMOD

(3)	002122	055103	046122	020111
(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		

DESCRIPT .ASCIZ <CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC>
/CZRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC/

(2)
26
27 002212
(3) 002212 046122 030460 051054
(3) 002220 030114 000062

.EVEN
DEVTYP <RL01,RL02>
.ASCIZ /RL01,RL02/

(2)
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

.EVEN
:COPYRIGHT (C) 1979
:THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY
:ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
:THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
:SOFTWARE, OR ANY COPIES THEREOF, MAY NOT BE PROVIDED
:OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT
:FOR USE ON SUCH SYSTEM, AND TO ONE WHO AGREES TO THESE
:LICENSE TERMS. TITLE TO OWNERSHIP OF THE SOFTWARE SHALL
:AT ALL TIMES REMAIN IN DEC.
:
:THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
:WITHOUT NOTICE AND SHALL NOT BE CONSTRUED AS A COMMITMENT
:BY DIGITAL EQUIPMENT CORPORATION.
:
:DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
:OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

```
47          .SBTTL BIT AND OFFSET DEFINITIONS
48
49 002224    BGNMOD GLBEQAT
50
51 002224    EQUALS
(1)          :
(1)          : BIT DIFINITIONS
(1)          :
(1)          BIT15== 100000
(1)          BIT14== 40000
(1)          BIT13== 20000
(1)          BIT12== 10000
(1)          BIT11== 4000
(1)          BIT10== 2000
(1)          BIT09== 1000
(1)          BIT08== 400
(1)          BIT07== 200
(1)          BIT06== 100
(1)          BIT05== 40
(1)          BIT04== 20
(1)          BIT03== 10
(1)          BIT02== 4
(1)          BIT01== 2
(1)          BIT00== 1
(1)          :
(1)          BIT9== BIT09
(1)          BIT8== BIT08
(1)          BIT7== BIT07
(1)          BIT6== BIT06
(1)          BIT5== BIT05
(1)          BIT4== BIT04
(1)          BIT3== BIT03
(1)          BIT2== BIT02
(1)          BIT1== BIT01
(1)          BIT0== BIT00
(1)          :
(1)          : EVENT FLAG DEFINITIONS
(1)          : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1)          :
(1)          EF.START== 32.          ; START COMMAND WAS ISSUED
(1)          EF.RESTART== 31.       ; RESTART COMMAND WAS ISSUED
(1)          EF.CONTINUE== 30.      ; CONTINUE COMMAND WAS ISSUED
(1)          EF.NEW== 29.           ; A NEW PASS HAS BEEN STARTED
(1)          EF.PWR== 28.           ; A POWER-FAIL/POWER-UP OCCURRED
(1)          :
(1)          : PRIORITY LEVEL DEFINITIONS
(1)          :
(1)          PRI07== 340
(1)          PRI06== 300
(1)          PRI05== 240
(1)          PRI04== 200
(1)          PRI03== 140
(1)          PRI02== 100
(1)          PRI01== 40
(1)          PRI00== 0
```



```
(1)
(1)
(1)
(1) 000004
(1) 000010
(1) 000020
(1) 000040
(1) 000100
(1) 000200
(1) 000400
(1) 001000
(1) 002000
(1) 004000
(1) 010000
(1) 020000
(1) 040000
(1) 100000
52
53
54 000000
55 000002
56 000004
57 000006
58 000010
59 000012
60
61
62 000000
63 000002
64 000004
65 000006
66 000010
67 000012
68
69
70 000001
71 000002
72 000004
73 000010
74 010000
75 020000
76 040000
77 100000
78
79
80 000102
81 000104
82 000106
83 000110
84 000112
85 000114
86 000116
87 000100
88
89
90 007777
```

```

: OPERATOR FLAG BITS
:
EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
HOE== 100000

: OFFSETS FOR HARDWARE P-TABLE
CSR =0 :BUS ADDRESS
VECT =2 :VECTOR ADDRESS
PRIOR =4 :PRIORITY
TYPDR =6 :DRIVE TYPE
DRSB =10 :DRIVE SELECT
CNT =12 :CONTROLLER TYPE

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

: BIT ASSIGNMENTS FOR SOFTWARE P-TABLE SWITCHES
ALLCYL =BIT00 :USE ALL CYLINDERS
ALLSEC =BIT01 :USE ALL SECTORS
DRSELT =BIT02 :EXECUTE DRIVE SELECT TEST
HDALIGN =BIT03 :EXECUTE HEAD ALIGNMENT TEST
HEADLM =BIT12 :HEAD LIMIT SPECIFIED FLAG
HICYL =BIT13 :HI LIMIT SPECIFIED FLAG
LOCYL =BIT14 :LO LIMIT SPECIFIED
MITEST =BIT15 :EXECUTE MANUAL INTERVENTION TESTS

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

: OPERATION FLAGS
COMPOP =7777 :COMPOSITE OPERATION FLAGS
```

91	000002	HDRCMP	=BIT01	:HEADER COMPARE OPERATION
92	000001	DATAcmp	=BIT00	:DATA COMPARE OPERATION
93	000004	CYLUP	=BIT02	:CYCLE UP OPERATION
94	000010	ULOAD	=BIT03	:UNLOAD OPERATION
95	000020	INOUTS	=BIT04	:IN-OUT SEEK OPERATION
96	000040	OUTINS	=BIT05	:OUT-IN SEEK OPERATION
97	000100	FOLWRT	=BIT06	:FOLLOWING WRITE OPERATION
98	000200	REVSks	=BIT07	:REV SEEK SEQ (ADJ INTERFERENCE)
99	000400	FWDSks	=BIT08	:FWD SEEK SEQ (ADJ INTERFERENCE)
100	001000	REVSKO	=BIT09	:REV SEEK SEQ (OVERWRITE)
101	002000	FWDSKO	=BIT10	:FWD SEEK SEQ (OVERWRITE)
102	004000	BADADD	=BIT11	:BAD DISK ADDRESS
103	010000	SEEKOP	=BIT12	:SEEK OPERATION
104	020000	RORWOP	=BIT13	:READ OR WRITE OPERATION
105	040000	RELDWT	=BIT14	:RELOAD WAIT
106	100000	HDR40	=BIT15	:40 HEADER OPERATION
107	003760	MQUALS	=OUTINS!INOUTS!FOLWRT!REVSks!FWDSks!REVSKO!FWDSKO	:MESSAGE QUALIFIER BITS
108				
109				
110		:	ERROR FLAGS FROM SUBROUTINES	
111	000001	TOSLOW	=BIT00	:OPERATION TOOK TOO LONG
112	000002	NOIRPT	=BIT01	:NO INTERRUPT FROM OPERATION
113	000004	CONHNG	=BIT02	:CONTROLLER HUNG
114	000010	NOCLR	=BIT03	:BAD CONTROLLER CLEAR
115				
116	000000	RLCS	=0	:CONTROL AND STATUS REGISTER
117	000002	RLBA	=2	:BUS ADDRESS REGISTER
118	000004	RLDA	=4	:DISK ADDRESS REGISTER
119	000006	RLMP	=6	:MULTI-PURPOSE REGISTER
120				
121		:	REGISTER BIT DEFINITIONS	
122	000000	RLCSR	=0	:CONTROL AND STATUS REGISTER
123	100000	ANYERR	=100000	:ANY ERROR BIT
124	040000	DRVERR	=40000	:DRIVE ERROR BIT
125	020000	NXMERR	=20000	:NON-EXISTENT MEMORY ERROR
126	010000	DLTERR	=10000	:DATA LATE ERROR
127	010000	HNFERR	=10000	:HEADER NOT FOUND ERROR
128	004000	DCKERR	=4000	:DATA CHECK ERROR
129	004000	HRCERR	=4000	:HEADER CHECK ERROR
130	002000	OPIERR	=2000	:OPERATION INCOMPLETE ERROR
131	001400	DSMSK	=1400	:DRIVE SELECT MASK
132	000200	CRDYMSK	=200	:CONTROLLER READY MASK
133	000100	INTEBL	=100	:INTERRUPT ENABLE MASK
134	000060	BAMSK	=60	:BUS ADDRESS UPPER MASK
135	000001	DRDYMSK	=1	:DRIVE READY MASK
136				


```
138          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
139          000077 ;SAMSK =77 ;SECTOR ADDRESS MASK
140          000100 HSMSK =100 ;HEAD SELECT MASK
141
142          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
143          000001 ;MBSETO =1 ;MUST BE SET, BIT 0
144          000004 ;DIRBIT =4 ;DIRECTION BIT
145          000020 ;HDSEL =20 ;HEAD SELECT BIT
146
147          : REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
148          000003 ;GETSTAT =3 ;GET STATUS SETUP
149          000010 ;DRSET =10 ;DRIVE RESET MASK
150
151          : REGISTER BIT DEFINITIONS - MP FOR DATA XFER
152          017777 ;WCMSK =17777 ;WORD COUNT MASK
153          160000 ;WCRNG =160000 ;WORD COUNT RANGE MASK
154
155          : REGISTER BIT DEFINITIONS - MP FOR READ HEADER
156          000077 ;HDSEC =77 ;SECTOR MASK
157          000100 ;HDHSEL =100 ;HEAD SELECT MASK
158
159          : REGISTER BIT DEFINITIONS - MP FOR GET STATUS
160          000007 ;STAMSK =7 ;STATE MASK
161          000010 ;BHSTAT =10 ;BRUSH HOME STATUS
162          000020 ;HOSTAT =20 ;HEADS OUT STATUS
163          000040 ;COSTAT =40 ;COVER OPEN STATUS
164          000100 ;HSSTAT =100 ;HEAD SELECT STATUS
165          000400 ;DSESTAT =400 ;DRIVE SELECT ERROR STATUS
166          001000 ;VCSTAT =1000 ;VOLUME CHECK STATUS
167          002000 ;WGESTAT =2000 ;WRITE GATE ERROR STATUS
168          004000 ;SPDSTAT =4000 ;SPIN ERROR STATUS
169          010000 ;STOSTAT =10000 ;SEEK TIMEOUT ERROR STATUS
170          020000 ;WLSTAT =20000 ;WRITE LOCK STATUS
171          040000 ;HCESTAT =40000 ;HEAD CURRENT ERROR STATUS
172          100000 ;WDESTAT =100000 ;WRITE DATA ERROR STATUS
173
174          002224 ENDMOD
175
176
```

178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

```
.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
```



```
217  
218 .SBTTL GLOBAL DATA AND CONSTANTS  
219  
220 002224 BGNMOD GLBDAT  
221  
222 ; TABLE OF OPERATION MESSAGES  
223 002224 000000 OPMSGs: .WORD 0 ; FILLER  
224 002226 005267 .WORD MWRCHK ; MESSAGE FOR WRITE CHECK  
225 002230 005313 .WORD MGTSTA ; GET STATUS  
226 002232 005240 .WORD MSEEK ; SEEK  
227 002234 005257 .WORD MREADH ; READ HEADER  
228 002236 005301 .WORD MWRITE ; WRITE DATA  
229 002240 005246 .WORD MREAD ; READ DATA  
230 002242 005376 .WORD MWRSET ; WITH RESET  
231 002244 005325 .WORD MDATCP ; WITH DATA COMPARE  
232 002246 005344 .WORD MHDRCP ; WITH HEADER COMPARE  
233 002250 005443 .WORD MCYLUP ; LOAD HEADS  
234 002252 005432 .WORD MLOAD ; UNLOAD HEADS  
235 002254 005474 .WORD MINOUT ; IN-OUT SEQ  
236 002256 005453 .WORD MOUTIN ; OUT-IN SEQ  
237 002260 005517 .WORD MFLWRT ; FOLLOWING WRITE  
238 002262 005541 .WORD MREVSK ; REV SEEK  
239 002264 005574 .WORD MFWDSK ; FWD SEEK  
240 002266 005663 .WORD MRESKO ; REV SEEK  
241 002270 005627 .WORD MFWSKO ; FWD SEEK  
242 002272 005717 .WORD MBADAD ; BAD DISK ADD FOR WRITE  
243 002274 005362 .WORD M4OHDR ; 40 HEADER OPERATION  
244 002276 000000  
245 002300 000000 T.DRIVE: .WORD 0  
246 002302 000000 JJJ: .WORD 0  
247 002304 000000 HLMTW: .WORD 0  
248 002306 000000 CLRBYT: .WORD 0  
249 002310 000000 NXTHL: .WORD 0  
250 002312 000000 GBND: .WORD 0  
251 002314 000000 CAMSK: .WORD 0  
252 002316 000000 DIRMSK: .WORD 0  
253 HDCYL: .WORD 0  
254 ; TABLE OF RESULT NAME MESSAGE ADDRESSES  
255 002320 010274 RESTBL: .WORD MCERR ; CONTROLLER ERROR  
256 002322 010405 .WORD MDRERR ; DRIVE ERROR  
257 002324 010720 .WORD MNEERR ; NON-EXISTENT MEMORY ERROR  
258 002326 010672 .WORD MFLERR ; HEADER NOT FOUND-DATA LATE  
259 002330 010655 .WORD MHDERR ; HEADER OR DATA ERROR  
260 002332 010645 .WORD MOPERR ; OPERATION INCOMPLETE  
261 002334 010736 .WORD MNDRST ; NO DRIVE STATUS AVAILABLE  
262 002336 000000 .WORD 0  
263 002340 010630 .WORD MWDERR ; WRITE DATA ERROR  
264 002342 010612 .WORD MHCERR ; HEAD CURRENT ERROR  
265 002344 000000 .WORD 0  
266 002346 010574 .WORD MSTERR ; SEEK TIMEOUT ERROR  
267 002350 010541 .WORD MSPERR ; SPINDLE ERROR  
268 002352 010557 .WORD MWGERR ; WRITE GATE ERROR  
269 002354 000000 .WORD 0  
270 002356 010511 .WORD MDSERR ; DRIVE SELECT ERROR
```

272
273
274 002360 004762
275 002362 004764
276 002364 005024
277 002366 005064
278 002370 005124
279 002372 005132
280 002374 005172
281 002376 005174
282 002400 005234
283 002402 005236
284
285
286
287 002404 000000
288 002406 000000
289 002410 000000
290 002412 000000
291 002414 000000
292 002416 000000
293 002420 000000
294 002422 000000
295 002424 000000
296 002426 000000
297
298
299 002430 000002
300 002432 000006
301 002434 000011
302 002436 000014
303 002440 000021
304 002442 000026
305 002444 000033
306 002446 000042
307 002450 000051
308 002452 000200
309 002454 000377
310
311
312 002456 000004
313 002460 000014
314 002462 000022
315 002464 000030
316 002466 000042
317 002470 000054
318 002472 000066
319 002474 000104
320 002476 000122
321 002500 000400
322 002502 000777
323
324
325
326 002504 000020
327 002544 000020

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

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

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

: RL02 TABLE OF CYLINDERS
T25TB2: .WORD 4
.WORD 12.
.WORD 18.
.WORD 24.
.WORD 34.
.WORD 44.
.WORD 54.
.WORD 68.
.WORD 82.
.WORD 256.
.WORD 511.

: TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS
T33TBL: .BLKW 16.
TBT: .BLKW 16.

328					
329					
330	002604	002	CYLTBL: .BYTE	2	;TABLE OF DEFAULT CYLINDERS
331	002605	007	.BYTE	7.	
332	002606	016	.BYTE	14.	
333	002607	024	.BYTE	20.	
334	002610	033	.BYTE	27.	
335	002611	041	.BYTE	33.	
336	002612	046	.BYTE	38.	
337	002613	055	.BYTE	45.	
338	002614	064	.BYTE	52.	
339	002615	072	.BYTE	58.	
340	002616	101	.BYTE	65.	
341	002617	110	.BYTE	72.	
342	002620	115	.BYTE	77.	
343	002621	124	.BYTE	84.	
344	002622	133	.BYTE	91.	
345	002623	141	.BYTE	97.	
346	002624	146	.BYTE	102.	
347	002625	154	.BYTE	108.	
348	002626	161	.BYTE	113.	
349	002627	170	.BYTE	120.	
350	002630	177	.BYTE	127.	
351	002631	206	.BYTE	134.	
352	002632	213	.BYTE	139.	
353	002633	222	.BYTE	146.	
354	002634	230	.BYTE	152.	
355	002635	235	.BYTE	157.	
356	002636	244	.BYTE	164.	
357	002637	252	.BYTE	170.	
358	002640	261	.BYTE	177.	
359	002641	270	.BYTE	184.	
360	002642	275	.BYTE	189.	
361	002643	303	.BYTE	195.	
362	002644	312	.BYTE	202.	
363	002645	317	.BYTE	207.	
364	002646	326	.BYTE	214.	
365	002647	334	.BYTE	220.	
366	002650	343	.BYTE	227.	
367	002651	352	.BYTE	234.	
368	002652	361	.BYTE	241.	
369	002653	367	.BYTE	247.	
370	002654	375	.BYTE	253.	
371	002655	000	.BYTE	0	
372	002656	000401	.WORD	257.	
373	002660	000406	.WORD	262.	
374	002662	000415	.WORD	269.	
375	002664	000423	.WORD	275.	
376	002666	000432	.WORD	282.	
377	002670	000445	.WORD	293.	
378	002672	000454	.WORD	300.	
379	002674	000463	.WORD	307.	
380	002676	000471	.WORD	313.	
381	002700	000500	.WORD	320.	
382	002702	000507	.WORD	327.	
383	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	000000	T.MP: .WORD	0	
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 STORAGE
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

584	005134	150442	.WORD	150442
585	005136	064221	.WORD	064221
586	005140	132110	.WORD	132110
587	005142	055044	.WORD	055044
588	005144	026442	.WORD	026442
589	005146	013211	.WORD	013211
590	005150	105504	.WORD	105504
591	005152	042642	.WORD	042642
592	005154	021321	.WORD	021321
593	005156	110550	.WORD	110550
594	005160	044264	.WORD	044264
595	005162	022132	.WORD	022132
596	005164	011055	.WORD	011055
597	005166	104426	.WORD	104426
598	005170	042213	.WORD	042213

599
600 005172 177777 PAT7: .WORD 177777

601
602 005174 045513 PAT8: .WORD 045513
603 005176 122645 .WORD 122645
604 005200 151322 .WORD 151322
605 005202 064551 .WORD 064551
606 005204 132264 .WORD 132264
607 005206 055132 .WORD 055132
608 005210 026455 .WORD 026455
609 005212 113226 .WORD 113226
610 005214 045513 .WORD 045513
611 005216 122645 .WORD 122645
612 005220 151322 .WORD 151322
613 005222 064551 .WORD 064551
614 005224 132264 .WORD 132264
615 005226 055132 .WORD 055132
616 005230 026455 .WORD 026455
617 005232 113226 .WORD 113226

618
619 005234 125252 PAT9: .WORD 125252

620
621 005236 155555 PAT10: .WORD 155555

622
623 005240 ENDMOD

624
625
626 .SBTTL GLOBAL MESSAGES

627
631 005240 BGNMOD GLBXTX
632 005240 042523 045505 000040 MSEEK: .ASCIZ /SEEK /
633 005246 042122 042040 052101 MREAD: .ASCIZ /RD DATA /
634 005257 122 020104 042110 MREADH: .ASCIZ /RD HDR /
635 005267 127 052122 041440 MWRCHK: .ASCIZ /WRT CHECK/
636 005301 127 052122 042040 MWRITE: .ASCIZ /WRT DATA /
637 005313 107 052105 051440 MGTSTA: .ASCIZ /GET STAT /
638 005325 127 052111 020110 MDATCP: .ASCIZ /WITH DATA CMP /
639 005344 044527 044124 044040 MHDRCP: .ASCIZ /WITH HDR CMP /
640 005362 047506 020122 030064 M4OHDR: .ASCIZ /FOR 40 HDRS/
641 005376 044527 044124 051040 MWRSET: .ASCIZ /WITH RESET /
642 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	051440	MFOLWRT: .ASCIZ	/FOL WRT (NO SEEK)/
649	005541	101	045104	041440	MREVSK: .ASCIZ	/ADJ CYL WRTTN AFTER REV SK/
650	005574	042101	020112	051503	MFWDISK: .ASCIZ	/ADJ CYL WRTTN AFTER FWD SK/
651	005627	123	020113	051506	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	MFMTERR: .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	NOPIR: .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	MRRRES: .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	VNRST: .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/

728	007005	110	051504	043040	HDMOVF: .ASCIZ	/HDS FAILED TO MOVE IN 10 TRIES/	
730	007044	054503	020114	047520	CYLPFR: .ASCIZ	/CYL PORTION OF HDS DIFFER WHEN READ FROM TRK 0 & 1/	
731	007130	042510	042101	040440	HAMES1: .ASCIZ	/HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/	
732	007213	124	050131	020105	HAMES2: .ASCIZ	&TYPE "CTL/C" TO GET BACK TO SUPVR COMMAND MODE AND THEN TYPE "CON"	&
733	007322	041101	053117	020105	OPR002: .ASCIZ	/ABOVE CONDITIONS MET/	
734	007347	127	051501	046040	OPR003: .ASCIZ	/WAS LOAD DEPRESSED/	
735	007372	044103	020113	051104	OPR1: .ASCIZ	/CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /	
736	007450	046103	042523	041440	OPR2: .ASCIZ	/CLSE COVER & RST WRT LCK /	
737	007502	051120	051505	020123	OPR3: .ASCIZ	/PRESS LOAD /	
738	007516	051120	051505	020123	OPR5: .ASCIZ	/PRESS LOAD & WAIT FOR LOAD LIGHT /	
739	007560	051120	051505	020123	OPR6: .ASCIZ	/PRESS LOAD & WAIT FOR RDY /	
740	007613	122	046505	053117	OPR7: .ASCIZ	/REMOVE ADD PLGS EXCPT /	
741	007642	047111	051123	020124	OPR8: .ASCIZ	/INSRT ADD PLG /	
742	007661	111	020116	046101	OPR9: .ASCIZ	/IN ALL DRVS /	
743	007676	047111	052523	043106	OPR10: .ASCIZ	/INSUFFICIENT DRVS FOR DRV SEL ERR TST/	
744	007744	050122	041514	020105	OPR11: .ASCIZ	/RPLCE ADD PLGS AS BEFORE/	
746	007775	122	051505	052105	OPR12: .ASCIZ	/RESET WRT LCK /	
747	010014	047117	000040		OPR1A: .ASCIZ	/ON /	
748	010020	047117	042040	053122	OPR1B: .ASCIZ	/ON DRV /	
749	010030	047125	042504	020122	UNDTST: .ASCIZ	/UNDER TEST/	
750	010043	123	052105	053440	OPR004: .ASCIZ	/SET WRT LCK /	
751	010060	044504	043106	000040	DIFWD: .ASCIZ	/DIFF /	
752	010066	043523	020116	000	SGNWD: .ASCIZ	/SGN /	
753	010073	110	020104	000	HDWD: .ASCIZ	/HD /	
754	010077	123	041505	000040	SECWD: .ASCIZ	/SEC /	
755	010104	054503	020114	000	CYLWD: .ASCIZ	/CYL /	
756	010111	106	047522	020115	FRMWD: .ASCIZ	/FROM /	
757	010117	040	054502	040520	BYPSPM: .ASCIZ	/ BYPASSED /	
758	010132	047522	052125	047111	SEQMES: .ASCIZ	/ROUTINE TRACE SEQ:/	
759	010155	104	053122	051440	STAMES: .ASCIZ	/DRV STAT/	
760	010166	040502	020104	042523	BSNSTR: .ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./	
761	010242	047524	040524	020114	TCERR: .ASCIZ	/TOTAL CMP ERRS: /	
762							
763							
764	010263	104	053122	051040	MDRDY: .ASCIZ	/DRV RDY /	
765	010274	047503	052116	042440	MCERR: .ASCIZ	/CONT ERR /	
766	010306	042110	020122	051103	MHCRC: .ASCIZ	/HDR CRC/	
767	010316	040504	040524	041440	MDCRC: .ASCIZ	/DATA CRC/	
768	010327	110	051104	047040	MHNF: .ASCIZ	/HDR NOT FND/	
769	010343	104	052101	020101	MDLT: .ASCIZ	/DATA LATE/	
770	010355	110	051104	047040	MHFCRC: .ASCIZ	&HDR NOT FND/HDR CRC/OPI&	
771	010405	104	053122	042440	MDRERR: .ASCIZ	/DRV ERR /	
773	010416	042523	023514	020104	MHSTA: .ASCIZ	/SEL'D HD /	
774	010430	047526	020114	044103	MVOLCK: .ASCIZ	/VOL CHK /	
775	010441	103	053117	051105	MCOSTA: .ASCIZ	/COVER OPN /	
776	010454	051102	051525	020110	MBHSTA: .ASCIZ	/BRUSH HME /	
777	010467	127	052122	046040	MWLSTA: .ASCIZ	/WRT LCK /	
778	010500	042110	020123	052517	MHOSTA: .ASCIZ	/HDS OUT /	
780	010511	104	053122	051440	MDSERR: .ASCIZ	/DRV SEL ERR /	
781	010526	051104	020126	052123	MDRVST: .ASCIZ	/DRV STATE /	
782	010541	123	044520	020116	MSPERR: .ASCIZ	/SPIN TIMEOUT /	
783	010557	127	052122	043440	MWGERR: .ASCIZ	/WRT GAT ERR /	
784	010574	042523	045505	052040	MSTERR: .ASCIZ	/SEEK TIMEOUT /	
785	010612	042510	042101	041440	MHCERR: .ASCIZ	/HEAD CUR ERR /	
786	010630	051127	020124	040504	MWDERR: .ASCIZ	/WRT DAT ERR /	
787	010645	117	051120	044455	MOPERR: .ASCIZ	/OPR-INC/	

788	010655	110	051104	042057	MHDERR:	.ASCIZ	&HDR/DAT ERR &
789	010672	042110	020122	047516	MFLERR:	.ASCIZ	&HDR NOT FND/DAT LATE &
790	010720	026516	026530	042515	MNEERR:	.ASCIZ	/N-X-MEM /
791	010731	103	046131	000040	MCYLOC:	.ASCIZ	/CYL /
792	010736	040503	047116	052117	MNDRST:	.ASCIZ	/CANNOT GET DRV STAT/
793	010762	047125	047113	042040	MUNDEF:	.ASCIZ	/UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
794	011027	106	044501	020114	MRLFAL:	.ASCIZ	/FAIL TO RELD HDS AFTER ERR CLEAR/
795	011070	051127	020124	041101	MWRTAB:	.ASCIZ	/WRT ABORTED/
796	011104	047440	042526	020122	MEXERS:	.ASCIZ	/ OVER ERR LIMIT - UNIT DROPPED /
797	011144	042440	051122	051117	MERRS:	.ASCIZ	/ ERROR/
798	011153	207	177777	000	BELL:	.ASCIZ	<207><377><377>
799							
800					:	RESULT SETTINGS	
801	011157	111	020123	000	RESE3:	.ASCIZ	/IS /
802	011163	040	041123	000040	RESE4:	.ASCIZ	/ SB /
803							
804					:	RESULT CONDITIONS	
805	011170	044440	020116	000	RESE5:	.ASCIZ	/ IN /
806	011175	040	043117	000040	RESE6:	.ASCIZ	/ OF /
807	011202	052123	052101	020105	STATE2:	.ASCIZ	/STATE 2/
808	011212	052123	052101	020105	STATE3:	.ASCIZ	/STATE 3/
809	011222	052123	052101	020105	STATE5:	.ASCIZ	/STATE 5/
811	011232	042523	045505	053440	CDRDY:	.ASCIZ	&SEEK W/O MOTION&
813	011252	051461	020124	020063	C10MS:	.ASCIZ	/1ST 3 MS/
814	011263	065	030060	051515	C500MS:	.ASCIZ	/500MS/
815	011271	103	041531	042514	CCYLUP:	.ASCIZ	/CYCLE UP/
816	011302	040504	040524	054040	CAFDT:	.ASCIZ	/DATA XFR/
817	011313	065	051440	041505	C5SEC:	.ASCIZ	/5 SEC/
818							
819	011321	045	022516	022524	FMTOP1:	.ASCIZ	/&N&T&N&T&T&O6&S&T&O1&N/
820	011350	047045	052045	047445	FMTOP2:	.ASCIZ	/&N&T&O1&S1&T&O1&N/
821	011372	047045	052045	047445	FMTOP3:	.ASCIZ	/&N&T&O1&S1&T&T&N/
822	011413	045	022524	000124	FMT1:	.ASCIZ	/&T&T/
823	011420	047045	052045	052045	FMT1.1:	.ASCIZ	/&N&T&T/
824	011427	045	000124		FMT2:	.ASCIZ	/&T/
825	011432	047045	000		FMT3:	.ASCIZ	/&N/
826	011435	045	022516	022524	FMT4:	.ASCIZ	/&N&T&T&N/
827	011446	047045	052045	047445	FMT5:	.ASCIZ	/&N&T&O6&S1&T&O1/
828	011466	047045	051445	030461	FMT6:	.ASCIZ	/&N&S11&T&S4&T&S4&T&S4&T&S4&T&S2&T/
829	011530	047045	052045	047445	FMT7:	.ASCIZ	/&N&T&O6&S2&O6&S2&O6&S2&O6&S3&O3&S2&O1&N/
830	011600	047045	052045	047445	FMT8:	.ASCIZ	/&N&T&O6&S2&O6&S2&O6&S2&O6/
831	011632	047045	052045	000	FMT9:	.ASCIZ	/&N&T/
832	011637	045	022524	030517	FMT11:	.ASCIZ	/&T&O1/
833	011645	045	022524	031517	FMT12:	.ASCIZ	/&T&O3/
834	011653	045	022516	030523	FMT13:	.ASCIZ	/&N&S11&T&O3&S1&T&O3&S1&T&O1&S1&T&O1/
835	011717	045	022516	022524	FMT14:	.ASCIZ	/&N&T&T&D3&S1&T&O6&S1&T&O6/
836	011751	045	022516	030523	FMT15:	.ASCIZ	/&N&S11&T&D3&S1&T&O6&S1&T&O6/
837	012005	045	022516	032523	FMT16:	.ASCIZ	/&N&S5&O6/
838	012016	051445	030061	052045	FMT17:	.ASCIZ	/&S10&T&N&S11&O6&N/
839	012040	047045	051445	032461	FMT18:	.ASCIZ	/&N&S15&T&S5&T&S4&T&S5&T&N/
840	012072	052045	051445	022464	FMT19:	.ASCIZ	/&T&S4&D6&S4&D6&S4&D6&S4&D6&N/
841	012127	045	022524	031123	FMT20:	.ASCIZ	/&T&S2&D6&S14&D6&S4&D6&N/
842	012157	045	022524	030523	FMT21:	.ASCIZ	/&T&S12&D6&S14&D6&N/
843	012202	047045	051445	030461	FMT22:	.ASCIZ	/&N&S11&T&O3&S1&T&O1&S1&T&O2/
844	012236	052045	052045	052045	FMT23:	.ASCIZ	/&T&T&T&O1&N/
845	012252	047045	052045	000	FMT24:	.ASCIZ	/&N&T/

846 012257 045 022516 031104 FMT25: .ASCIZ /%N%D2%T/
847 012267 045 022516 030523 FMT26: .ASCIZ /%N%S1%T%D4%T%T%D3%N/
848 012313 045 022516 022524 FMT27: .ASCIZ /%N%T%D3%T%D3%N/
849 012332 047045 052045 052045 FMT28: .ASCIZ /%N%T%T%T/
850 012343 ENDMOD

855
856
857
858
859 012344
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905

.SBTTL ERROR MESSAGES

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 012344          BGNMSG  ERR1
908 012344 105737 003363      TSTB      NOERCT      ;TEST IF ERROR COUNTING INHIBITED
909 012350 001002          BNE      1$      ;YES - SKIP
910 012352 005277 170576      INC      @ERRPOINT ;ELSE BUMP ERROR COUNT
911 012356 010146          1$:  MOV      R1,-(SP) ;STORE R1
912 012360 004737 023362      JSR      PC,RPTOP  ;REPORT OPERATION
913 012364 012721 000001      MOV      #1,(R1)+ ;SET PARAM NUMBER
914 012370 010321          MOV      R3,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
915 012372 004737 024150      JSR      PC,RPTRES ;REPORT RESULTS
916 012376 004737 024356      JSR      PC,RPTREM ;REPORT REMAINDER
917 012402 012601          MOV      (SP)+,R1  ;RESTORE R1
918 012404 004737 016132      JSR      PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
919 012410          ENDMSG
(3) 012410          L10000:
(3) 012410 104423          TRAP      C$MSG
920
921 012412          BGNMSG  ERR2
922 012412 005277 170536      INC      @ERRPOINT ;BUMP ERROR COUNT
923 012416 010146          MOV      R1,-(SP) ;STORE R1
924 012420 004737 023362      JSR      PC,RPTOP  ;REPORT OPERATION
925 012424 012721 000003      MOV      #3,(R1)+ ;SET PARAM NUMBER
926 012430 010321          MOV      R3,(R1)+ ;INSERT NAME ADD POINTER
927 012432 012721 000001      MOV      #1,(R1)+ ;SET IS VALUE
928 012436 005021          CLR      (R1)+    ;SET SB VALUE
929 012440 004737 024150      JSR      PC,RPTRES ;REPORT RESULTS
930 012444 004737 024356      JSR      PC,RPTREM ;REPORT REMAINDER
931 012450 012601          MOV      (SP)+,R1  ;RESTORE R1
932 012452 004737 016132      JSR      PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
933 012456          ENDMSG
(3) 012456          L10001:
(3) 012456 104423          TRAP      C$MSG
934
935 012460          BGNMSG  ERR3
936 012460 005277 170470      INC      @ERRPOINT ;BUMP ERROR COUNT
937 012464 010146          MOV      R1,-(SP) ;STORE R1
938 012466 004737 023362      JSR      PC,RPTOP  ;REPORT OPERATION
939 012472 012721 000003      MOV      #3,(R1)+ ;SET PARAM NUMBER
940 012476 010321          MOV      R3,(R1)+ ;INSERT NAME ADD POINTER
941 012500 005021          CLR      (R1)+    ;SET IS VALUE
942 012502 012721 000001      MOV      #1,(R1)+ ;SET SB VALUE
943 012506 004737 024150      JSR      PC,RPTRES ;REPORT RESULTS
944 012512 004737 024356      JSR      PC,RPTREM ;REPORT REMAINDER
945 012516 012601          MOV      (SP)+,R1  ;RESTORE R1
946 012520 004737 016132      JSR      PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
947 012524          ENDMSG
(3) 012524          L10002:
(3) 012524 104423          TRAP      C$MSG
948
949 012526          BGNMSG  ERR4
950 012526 005277 170422      INC      @ERRPOINT ;BUMP ERROR COUNT
951 012532 010146          MOV      R1,-(SP) ;STORE R1
952 012534 004737 023362      JSR      PC,RPTOP  ;REPORT OPERATION
953 012540 012721 000004      MOV      #4,(R1)+ ;SET PARAM NUMBER
954 012544 010321          MOV      R3,(R1)+ ;INSERT NAME ADD POINTER
955 012546 012721 000001      MOV      #1,(R1)+ ;SET IS VALUE
```


956	012552	005021		CLR	(R1)+	:SET SB VALUE	
957	012554	010411		MOV	R4,(R1)	:INSERT ADD OF CONDITION POINTER	
958	012556	004737	024150	JSR	PC,RPTRES	:REPORT RESULTS	
959	012562	004737	024356	JSR	PC,RPTREM	:REPORT REMAINDER	
960	012566	012601		MOV	(SP)+,R1	:RESTORE R1	
961	012570	004737	016132	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED	
962	012574			ENDMSG			
(3)	012574			L10003:			
(3)	012574	104423		TRAP	C\$MSG		
963							
964	012576			BGNMSG	ERR5		
965	012576	005277	170352	INC	@ERRPOINT	:BUMP ERROR COUNT	
966	012602	010146		MOV	R1,-(SP)	:STORE R1	
967	012604	004737	023362	JSR	PC,RPTOP	:REPORT OPERATION	
968	012610	012721	000004	MOV	#4,(R1)+	:SET PARAM NUMBER	
969	012614	010321		MOV	R3,(R1)+	:INSERT NAME ADD POINTER	
970	012616	005021		CLR	(R1)+	:SET IS VALUE	
971	012620	012721	000001	MOV	#1,(R1)+	:SET SB VALUE	
972	012624	010411		MOV	R4,(R1)	:INSERT ADD OF CONDITION POINTER	
973	012626	004737	024150	JSR	PC,RPTRES	:REPORT RESULTS	
974	012632	004737	024356	JSR	PC,RPTREM	:REPORT REMAINDER	
975	012636	012601		MOV	(SP)+,R1	:RESTORE R1	
976	012640	004737	016132	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED	
977	012644			ENDMSG			
(3)	012644			L10004:			
(3)	012644	104423		TRAP	C\$MSG		
978							
979	012646			BGNMSG	ERR6		
980	012646	105737	003363	TSTB	NOERCT	:TEST IF ERROR COUNTING INHIBITED	
981	012652	001002		BNE	17\$:YES - SKIP	
982	012654	005277	170274	INC	@ERRPOINT	:ELSE BUMP ERROR COUNT	
983	012660	010146		17\$:	MOV	R1,-(SP)	:STORE R1
984	012662	010346		MOV	R3,-(SP)	:STORE R3	
985	012664	010446		MOV	R4,-(SP)	:STORE R4	
986	012666	010546		MOV	R5,-(SP)	:STORE R5	
987	012670	004737	023362	JSR	PC,RPTOP	:REPORT OPERATION	
988	012674	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER	
989	012700	012761	000001	000002	MOV	#1,2(R1)	:INSERT IS VALUE
990	012706	005037	003124	CLR	TEMP3	:CLEAR FOR STATUS STORAGE	
991	012712	013703	003044	MOV	T.CS,R3	:GET T.CS	
992	012716	042703	177761	BIC	#177761,R3	:AND CLEAR ALL BUT FUNCTION	
993	012722	022703	000004	CMP	#4,R3	:CHECK IF IT WAS GET STATUS	
994	012726	001443		BEQ	1\$:YES - STATUS IS IN T.MP, SKIP	
995	012730	012762	000003	000004	MOV	#GETSTAT,RLDA(R2)	:ELSE DO GET STATUS
996	012736	012703	000004	MOV	#4,R3		
997	012742	053703	003032	BIS	RLDRV,R3		
998	012746	010362	000000	MOV	R3,RLCS(R2)		
999	012752			WAITUS	#10.	:WAIT FOR CONTROLLER READY	
(3)	012752	012727	000012	MOV	###10.,(PC)+		
(3)	012756	000000		.WORD	0		
(3)	012760	013727	002116	MOV	LSDLY,(PC)+		
(3)	012764	000000		.WORD	0		
(3)	012766	005367	177772	DEC	-6(PC)		
(3)	012772	001375		BNE	-.4		
(3)	012774	005367	177756	DEC	-22(PC)		
(3)	013000	001367		BNE	.-20		

1000	013002	032762	000200	000000	BIT	#CRDYMSK,RLCS(R2)	:TEST IF READY
1001	013010	001003			BNE	10\$:YES - SKIP
1002	013012	012703	001000		9\$: MOV	#BIT9,R3	:ELSE SET NO DRIVE STATUS BIT
1003	013016	000413			BR	2\$:IN MESSAGE WORD AND SKIP
1004	013020	016203	000006		10\$: MOV	RLMP(R2),R3	:STORE STATUS FOR REPORT
1005	013024	010337	003124		MOV	R3,TEMP3	
1006	013030	113703	003125		MOV	TEMP3+1,R3	:GET ERROR BITS IN PROPER POSITION
1007	013034	000402			BR	13\$	
1008	013036	113703	003053		1\$: MOV	T.MP+1,R3	:GET ERROR BITS FROM MP REG
1009	013042	042703	177442		13\$: BIC	#177442,R3	:CLEAR UNUSED BITS
1010	013046	013704	003044		2\$: MOV	T.CS,R4	:GET ERROR BITS FROM CS REG
1011	013052	042704	001777		BIC	#1777,R4	:CLEAR UNUSED BITS
1012	013056	050403			BIS	R4,R3	:MAKE ONE WORD OF POSSIBLE ERRORS
1013	013060	032703	002000		BIT	#OPIERR,R3	:TEST IF OPI SET
1014	013064	001442			BEQ	115\$:NO - SKIP
1015	013066	032703	010000		BIT	#HNFERR,R3	:TEST IF HDR NOT FOUND ERROR
1016	013072	001026			BNE	107\$:YES - SKIP
1017	013074	032703	004000		BIT	#HRCRCERR,R3	:TEST IF HDR CRC ERR
1018	013100	001020			BNE	105\$:YES - SKIP
1019	013102	012704	010645		MOV	#MOPERR,R4	:SET OPI ALONE MESSAGE
1020	013106				100\$: PRINTB	#FMT28,#MRSLT,R4,#MERRS	:REPORT ERROR
(10)	013106	012746	011144		MOV	#MERRS,-(SP)	
(9)	013112	010446			MOV	R4,-(SP)	
(8)	013114	012746	005421		MOV	#MRSLT,-(SP)	
(7)	013120	012746	012332		MOV	#FMT28,-(SP)	
(6)	013124	012746	000004		MOV	#4,-(SP)	
(3)	013130	010600			MOV	SP,R0	
(4)	013132	104414			TRAP	C\$PNTB	
(4)	013134	062706	000012		ADD	#12,SP	
1021	013140	000430			BR	120\$:SKIP
1022	013142	012704	010306		105\$: MOV	#MHCRC,R4	:HDR CRC MESSAGE
1023	013146	000757			BR	100\$	
1024	013150	032703	004000		107\$: BIT	#HRCRCERR,R3	:TEST IF HCRC WITH HDR NOT FND
1025	013154	001003			BNE	109\$:YES - SKIP
1026	013156	012704	010327		MOV	#MHNF,R4	:MESSAGE HEADER NOT FOUND
1027	013162	000751			BR	100\$	
1028	013164	012704	010355		109\$: MOV	#MHFCRC,R4	:HNF AND HCRC MESSAGE
1029	013170	000746			BR	100\$:SKIP
1030	013172	032703	004000		115\$: BIT	#DCKERR,R3	:TEST IF DATA CHECK SET, NOT OPI
1031	013176	001403			BEQ	118\$:NO - SKIP
1032	013200	012704	010316		MOV	#MDCRC,R4	:SET MESSAGE DATA CHECK
1033	013204	000740			BR	100\$:SKIP
1034	013206	032703	010000		118\$: BIT	#DLTERR,R3	:TEST IF DATA LATE ERROR
1035	013212	001403			BEQ	120\$:NO - SKIP
1036	013214	012704	010343		MOV	#MDLT,R4	:SET MESSAGE DATA LATE
1037	013220	000732			BR	100\$:SKIP
1038	013222	012705	100000		120\$: MOV	#BIT15,R5	:SET BIT POINTER FOR TEST
1039	013226	005004			CLR	R4	:CLEAR R4 FOR TABLE COUNT
1040	013230	030503			3\$: BIT	R5,R3	:TEST IF BIT IS SET
1041	013232	001005			BNE	6\$:YES - SKIP TO REPORT
1042	013234	005724			4\$: TST	(R4)+	:ELSE BUMP TABLE POINTER
1043	013236	000241			CLC		:CLEAR CARRY
1044	013240	006005			ROR	R5	:SHIFT BIT POINTER TO NEXT BIT
1045	013242	001372			BNE	3\$:LOOP IF NOT 0
1046	013244	000405			BR	7\$:ELSE REPORT REMAINDER
1047	013246	016411	002320		6\$: MOV	RESTBL(R4),(R1)	:INSERT NAME ADDRESS


```

1048 013252 004737 024150      JSR    PC,RPTRES      ;REPORT RESULTS
1049 013256 000766              BR      4$           ;GET NEXT BIT
1050 013260 004737 024356      7$: JSR    PC,RPTREM   ;REPORT REMAINDER
1051 013264 005737 003124      TST    TEMP3         ;TEST IF ANY NEW STATUS
1052 013270 001414              BEQ    15$           ;NO - SKIP
1053 013272              PRINTB #FMT17,#STAMES,TEMP3
      (9) 013272 013746 003124      MOV    TEMP3,-(SP)
      (8) 013276 012746 010155      MOV    #STAMES,-(SP)
      (7) 013302 012746 012016      MOV    #FMT17,-(SP)
      (6) 013306 012746 000003      MOV    #3,-(SP)
      (3) 013312 010600              MOV    SP,R0
      (4) 013314 104414              TRAP   C$PNTB
      (4) 013316 062706 000010      ADD    #10,SP
1054 013322 032737 004000 003044 15$: BIT    #DCKERR,T.CS   ;TEST IF DATA CHECK ERROR
1055 013330 001453              BEQ    25$           ;NO - SKIP
1056 013332 032737 002000 003044  BIT    #OPIERR,T.CS   ;TEST IF OPI SET
1057 013340 001047              BNE    25$           ;YES - SKIP
1058 013342 005037 003014      CLR    MORECE        ;CLEAR COMPARE ERROR COUNT
1059 013346 012701 000200      MOV    #128,R1       ;SET COMPARE LENGTH
1060 013352 012703 000001      MOV    #1,R3          ;SET WORD COUNT
1061 013356 012705 004362      MOV    #OBUFF,R5      ;SET GOOD WORD POINTER
1062 013362 012704 003762      MOV    #IBUFF,R4      ;SET TEST WORD POINTER
1063 013366 021514              18$: CMP    (R5),(R4)    ;CHECK WORD
1064 013370 001427              BEQ    19$           ;GOOD - SKIP
1065 013372 023727 003014 000012  CMP    MORECE,#10.    ;TEST IF COMPARE LIMIT REACHED
1066 013400 003021              BGT    20$           ;YES - SKIP
1067 013402              PRINTB #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
      (13) 013402 011546              MOV    (R5),-(SP)
      (12) 013404 012746 011163      MOV    #RESE4,-(SP)
      (11) 013410 011446              MOV    (R4),-(SP)
      (10) 013412 012746 011157      MOV    #RESE3,-(SP)
      (9) 013416 010346              MOV    R3,-(SP)
      (8) 013420 012746 006170      MOV    #MWORD,-(SP)
      (7) 013424 012746 011751      MOV    #FMT15,-(SP)
      (6) 013430 012746 000007      MOV    #7,-(SP)
      (3) 013434 010600              MOV    SP,R0
      (4) 013436 104414              TRAP   C$PNTB
      (4) 013440 062706 000020      ADD    #20,SP
1068 013444 005237 003014      20$: INC    MORECE      ;BUMP ERROR COUNTER
1069 013450 022524              19$: CMP    (R5)+,(R4)+ ;BUMP POINTERS
1070 013452 005203              INC    R3             ;BUMP COUNTER
1071 013454 005301              DEC    R1             ;DEC LENGTH COUNT
1072 013456 001343              BNE    18$           ;LOOP IF NOT DONE
1073 013460 005737 003014      25$: TST    MORECE      ;TEST IF ANY COMPARE ERRORS
1074 013464 001421              BEQ    27$           ;NO - SKIP
1075 013466 012701 000200      MOV    #128,R1       ;SET COMPARE LENGTH
1076 013472              PRINTB #FMT27,#TCERR,MORECE,#RESE6,R1
      (11) 013472 010146              MOV    R1,-(SP)
      (10) 013474 012746 011175      MOV    #RESE6,-(SP)
      (9) 013500 013746 003014      MOV    MORECE,-(SP)
      (8) 013504 012746 010242      MOV    #TCERR,-(SP)
      (7) 013510 012746 012313      MOV    #FMT27,-(SP)
      (6) 013514 012746 000005      MOV    #5,-(SP)
      (3) 013520 010600              MOV    SP,R0
      (4) 013522 104414              TRAP   C$PNTB
      (4) 013524 062706 000014      ADD    #14,SP

```

1077	013530	012605	27\$:	MOV	(SP)+,R5	:RESTORE R5, 4, 3, 1
1078	013532	012604		MOV	(SP)+,R4	
1079	013534	012603		MOV	(SP)+,R3	
1080	013536	012601		MOV	(SP)+,R1	
1081	013540	004737	016132	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1082	013544			ENDMSG		
(3)	013544			L10005:		
(3)	013544	104423		TRAP	C\$MSG	
1083						
1084	013546			BGNMSG	ERR7	
1085	013546	005277	167402	INC	@ERRPOINT	:BUMP ERROR COUNT
1086	013552	010146		MOV	R1,-(SP)	:STORE R1
1087	013554	004737	023362	JSR	PC,RPTOP	:REPORT OPERATION
1088	013560	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER
1089	013564	012721	010526	MOV	#MDRVST,(R1)+	:INSERT NAME ADD POINTER
1090	013570	013721	003060	MOV	T,STAT,(R1)+	:INSERT IS VALUE
1091	013574	010311		MOV	R3,(R1)	:INSERT SB VALUE
1092	013576	004737	024150	JSR	PC,RPTRES	:REPORT RESULTS
1093	013602	004737	024356	JSR	PC,RPTREM	:REPORT REMAINDER
1094	013606	012601		MOV	(SP)+,R1	:RESTORE R1
1095	013610	004737	016132	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1096	013614			ENDMSG		
(3)	013614			L10006:		
(3)	013614	104423		TRAP	C\$MSG	
1097						
1098	013616			BGNMSG	ERR8	
1099	013616	005277	167332	INC	@ERRPOINT	:BUMP ERROR COUNT
1100	013622	010146		MOV	R1,-(SP)	:STORE R1
1101	013624	010346		MOV	R3,-(SP)	:STORE R3
1102	013626	004737	023362	JSR	PC,RPTOP	:REPORT OPERATION
1103	013632	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER
1104	013636	012721	010731	MOV	#MCYLOC,(R1)+	:INSERT NAME ADD POINTER
1105	013642	013711	003052	MOV	HDWRD1,(R1)	:GET HEADER WORD
1106	013646	012703	000007	MOV	#7,R3	:SET SHIFT COUNT
1107	013652	000241		3\$:	CLC	
1108	013654	006011		ROR	(R1)	:ALIGN CHAR FOR PRINTING
1109	013656	005303		DEC	R3	: AS IS VALUE
1110	013660	001374		BNE	3\$	
1111	013662	005721		TST	(R1)+	:BUMP PARAM POINTER
1112	013664	013711	003102	MOV	NEWCYL,(R1)	:INSERT SB VALUE
1113	013670	004737	024150	JSR	PC,RPTRES	:REPORT RESULTS
1114	013674	004737	024356	JSR	PC,RPTREM	:REPORT REMAINDER
1115	013700	012603		MOV	(SP)+,R3	:RESTORE R3
1116	013702	012601		MOV	(SP)+,R1	:RESTORE R1
1117	013704	004737	016132	JSR	PC,CKERLM	:GO CHECK IF ERROR COUNT EXCEEDED
1118	013710			ENDMSG		
(3)	013710			L10007:		
(3)	013710	104423		TRAP	C\$MSG	
1119						
1120	013712			BGNMSG	ERR9	
1121	013712	005277	167236	INC	@ERRPOINT	:BUMP ERROR COUNT
1122	013716	010146		MOV	R1,-(SP)	:STORE R1
1123	013720	004737	023362	JSR	PC,RPTOP	:REPORT OPERATION
1124	013724	012721	000003	MOV	#3,(R1)+	:SET PARAM NUMBER
1125	013730	010321		MOV	R3,(R1)+	:INSERT NAME ADD POINTER
1126	013732	010421		MOV	R4,(R1)+	:SET IS VALUE


```
1127 013734 010521          MOV      R5,(R1)+      ;SET SB VALUE
1128 013736 004737 024150    JSR      PC,RPTRES     ;REPORT RESULTS
1129 013742 004737 024356    JSR      PC,RPTREM     ;REPORT REMAINDER
1130 013746 012601          MOV      (SP)+,R1      ;RESTORE R1
1131 013750 004737 016132    JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
1132 013754          ENDMSG
(3) 013754          L10010:
(3) 013754 104423          BGNMSG  TRAP      C$MSG
1133 013756          ERR10
1134 013756 010146          MOV      R1,-(SP)      ;STORE R1
1135 013760 005737 003014    TST      MORECE       ;TEST IF 2ND BAD LINE
1136 013764 001051          BNE      3$           ;YES - SKIP
1137 013766 005277 167162    INC      @ERRPOINT     ;BUMP ERROR COUNT
1138 013772 004737 023362    JSR      PC,RPTOP     ;REPORT OPERATION
1139 013776          PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT ID
(11) 013776 005046          CLR      -(SP)
(11) 014000 153716 003033    BISB    RLDRV+1,(SP)
(10) 014004 012746 006051    MOV      #DRVNAM,-(SP)
(9) 014010 013746 003026    MOV      RLBAS,-(SP)
(8) 014014 012746 006040    MOV      #BASADD,-(SP)
(7) 014020 012746 011446    MOV      #FMT5,-(SP)
(6) 014024 012746 000005    MOV      #5,-(SP)
(3) 014030 010600          MOV      SP,R0
(4) 014032 104414          TRAP    C$PNTB
(4) 014034 062706 000014    ADD      #14,SP
1140 014040          PRINTB  #FMT14,#MRSLT,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)
(14) 014040 011546          MOV      (R5),-(SP)
(13) 014042 012746 011163    MOV      #RESE4,-(SP)
(12) 014046 011446          MOV      (R4),-(SP)
(11) 014050 012746 011157    MOV      #RESE3,-(SP)
(10) 014054 010346          MOV      R3,-(SP)
(9) 014056 012746 006170    MOV      #MWORD,-(SP)
(8) 014062 012746 005421    MOV      #MRSLT,-(SP)
(7) 014066 012746 011717    MOV      #FMT14,-(SP)
(6) 014072 012746 000010    MOV      #10,-(SP)
(3) 014076 010600          MOV      SP,R0
(4) 014100 104414          TRAP    C$PNTB
(4) 014102 062706 000022    ADD      #22,SP
1141 014106 000421          BR      4$
1142 014110          3$: PRINTB  #FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5) ;REPORT DATA
(13) 014110 011546          MOV      (R5),-(SP)
(12) 014112 012746 011163    MOV      #RESE4,-(SP)
(11) 014116 011446          MOV      (R4),-(SP)
(10) 014120 012746 011157    MOV      #RESE3,-(SP)
(9) 014124 010346          MOV      R3,-(SP)
(8) 014126 012746 006170    MOV      #MWORD,-(SP)
(7) 014132 012746 011751    MOV      #FMT15,-(SP)
(6) 014136 012746 000007    MOV      #7,-(SP)
(3) 014142 010600          MOV      SP,R0
(4) 014144 104414          TRAP    C$PNTB
(4) 014146 062706 000020    ADD      #20,SP
1143 014152 005237 003014          4$: INC      MORECE     ;INC COMPARE ERROR COUNT
1144 014156 012601          MOV      (SP)+,R1      ;RESTORE R1
1145 014160 004737 016132    JSR      PC,CKERLM     ;GO CHECK IF ERROR COUNT EXCEEDED
1146 014164          ENDMSG
(3) 014164          L10011:
```

1147	(3)	014164	104423	TRAP	C\$MSG	
1148		014166		ENDMOD		
1149					.EVEN	
1150		014166		BGNMOD	HPTCODE	
1151		014166		BGNHW		
1152	(3)	014166	000006		.WORD	L10012-L\$HW/2
1153		014170	174400		.WORD	174400
1154		014172	000160		.WORD	160
1155		014174	000240		.WORD	240
1156		014176	000001		.WORD	1
1157		014200	000000		.WORD	0
1158		014202	000001		.WORD	1
1159	(3)	014204		ENDHW		
1160		014204		L10012:		
1161		014204		ENDMOD		
1162		014204		BGNMOD	SPTCODE	
1163	(3)	014204	000006	BGNSW		
1164		014206	000000	MISWIW:	.WORD	L10013-L\$SW/2
1165					.WORD	0
1166						:BIT 0 = USE ALL CYLINDERS
1167						:BIT 1 = USE ALL SECTORS
1168						:BIT 2 = EXECUTE DRIVE SELECT TEST
1169						:BIT 3 = EXECUTE HEAD ALIGNMENT
1170						:BIT 12 = HEAD SELECT SUPPLIED FLAG
1171		014210	000000			:BIT 13 = HILIMIT SPECIFIED FLAG
1172		014212	000377	LOLIMW:	.WORD	0
1173		014214	000000	HILIMW:	.WORD	255.
1174		014216	000024	HEADW:	.WORD	0
1175		014220	000012	ERLIMW:	.WORD	20.
1176		014222		DCLIMW:	.WORD	10.
1177	(3)	014222		ENDSW		:ERROR LIMIT
1178		014222		L10013:		:COMPARE ERROR LIMIT
1179		014222		ENDMOD		
1181		014222		BGNMOD	DSPCODE	
1182	(4)	014222	000020	DISPATCH		16
1183	(6)	014224	024642		.WORD	16
1184	(6)	014226	025122		.WORD	T1
1185	(6)	014230	025330		.WORD	T2
1186	(6)	014232	027514		.WORD	T3
1187	(6)	014234	030610		.WORD	T4
1188	(6)	014236	031214		.WORD	T5
1189	(6)	014240	032372		.WORD	T6
1190	(6)	014242	033070		.WORD	T7
1191	(6)	014244	033156		.WORD	T8
1192	(6)	014246	033624		.WORD	T9
1193	(6)	014250	034314		.WORD	T10
1194	(6)	014252	035066		.WORD	T11
1195	(6)	014254	035552		.WORD	T12
1196	(6)	014256	035772		.WORD	T13
1197	(6)	014260	036252		.WORD	T14
1198	(6)	014262	036774		.WORD	T15
					.WORD	T16


```
1186 014264 ENDMOD
1187
1188 ;LOAD PROTECTION TABLE
1189 014264 BGNPROT
1190 014264 000000 .WORD 0 ;P-TABLE OFFSET OF CSR
1191 014266 177777 .WORD -1 ;NOT A MASS-BUSS DRIVE
1192 014270 000010 .WORD 10 ;P-TABLE OFFSET OF DRIVE
1193 014272 ENDPROT
1194
1195
1196 .SBTTL INITIALIZATION CODE
1197
1198 014272 BGNMOD INITCODE
1199 014272 BGNINIT
1200 ;CHECK FOR PRESENCE OF A P-CLOCK
1201 014272 005037 003144 CLR CLKFLG ;CLEAR CLOCK FLAG
1202 014276 CLOCK P,CLKADR ;P-CLOCK?
(3) 014276 012700 000120 MOV #P,RO
(3) 014302 104462 TRAP C$CLCK
(3) 014304 010037 003146 MOV RO,CLKADR
1203 014310 BNCOMPLETE 1$ ;BRANCH IF NO P-CLOCK
(2) 014310 103002 BCC 1$
1204 014312 005237 003144 INC CLKFLG ;INDICATE PRESENCE OF A P-CLOCK
1205 014316 1$: SETPRI #340 ;SET PRIORITY TO 7 TO INHIBIT INTERRUPTS
(3) 014316 012700 000340 MOV #340,RO
(3) 014322 104441 TRAP C$SPRI
1206 014324 MANUAL ;CHECK IF MANUAL INTERVENTION ALLOWED
(3) 014324 104450 TRAP C$MANI
1207 014326 BCOMPLETE 2$ ;YES - SKIP
(2) 014326 103403 BCS 2$
1208 014330 042737 100014 014206 BIC #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
1209 ; INTERVENTION FLAGS
1210 014336 005037 003002 2$: CLR SSINDX ;CLEAR SUBROUTINE STACK INDEX
1211 014342 READEF #EF.PWR ;POWER FAILURE?
(3) 014342 012700 000034 MOV #EF.PWR,RO
(3) 014346 104447 TRAP C$REFG
1212 014350 BNCOMPLETE 4$ ;NO, GO CHECK NEW PASS
(2) 014350 103005 BCC 4$
1213 014352 013737 002012 003366 MOV LSUNIT,PWRFLG ;SET POWER FAIL FLAG
1214 014360 000137 014772 JMP PWCON ;GO SERVICE POWER FAIL
1215 ;'START' COMMAND SEQUENCE
1216 014364 4$: READEF #EF.START ;CHECK IF START
(3) 014364 012700 000040 MOV #EF.START,RO
(3) 014370 104447 TRAP C$REFG
1217 014372 BNCOMPLETE RESTART ;NO - SKIP
(2) 014372 103034 BCC RESTART
1218 ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
1219 ; PASS COUNT, AND ERROR COUNT.
1220 014374 013737 002012 003074 MOV LSUNIT,DRVCNT ;SET UP UNIT COUNT
1221 014402 005037 003356 RSTRT: CLR PASNUM ;CLEAR PASS NUMBER
1222 014406 012700 003156 MOV #ERRCNT,RO
1223 014412 012701 000100 MOV #64,R1 ;GET A COUNT
1224 014416 005020 1$: CLR (RO)+ ;CLEAR ERROR COUNTER STORAGE AREA
1225 014420 005301 DEC R1
1226 014422 001375 BNE 1$ ;LOOP TILL ALL CLEARED
1227 014424 012737 003154 003154 MOV #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
```



```

1228 014432 012737 177777 003360      MOV      #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
1229 014440 012737 177777 003010      MOV      #-1,HADONE     ;PRESET HEAD ALIGN DONE FLAG
1230 014446 032737 040000 014206 LAB:    BIT      #LOCYL,MISWIW  ;TEST IF LO LIMIT SET
1231 014454 001002                BNE      5$             ;YES - SKIP
1232 014456 005037 014210                CLR      LOLIMW        ;ELSE CLEAR LO LIMIT
1233 014462 000432                5$:      BR       SETDON
1234 014464                RESTART:
1235 014464                READEF   #EF.RESTART    ;CHECK IF RESTART
(3) 014464 012700 000037                MOV      #EF.RESTART,R0
(3) 014470 104447                TRAP     CSREFG
1236 014472                BCOMPLETE RSTRT        ;NO - SKIP
(2) 014472 103743                BCS     RSTRT
1237                ;'CONTINUE' COMMAND SEQUENCE
1238 014474                CONTINUE:
1239 014474                READEF   #EF.CONTINUE  ;TEST IF CONTINUE
(3) 014474 012700 000036                MOV      #EF.CONTINUE,R0
(3) 014500 104447                TRAP     CSREFG
1240 014502                BCOMPLETE PWCON
(2) 014502 103533                BCS     PWCON
1241                ;
1242 014504                ON CONTINUE PICK UP UNIT LAST UNDER TEST
(3) 014504 012700 000035                READEF   #EF.NEW        ;CHECK IF STARTING NEW PASS
(3) 014510 104447                MOV      #EF.NEW,R0
1243 014512                BCOMPLETE PASNEW
(2) 014512 103403                BCS     PASNEW
1244 014514                NXPAS:
1245 014514 005737 003074                TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
1246 014520 001013                BNE     SETDON          ;NO - SKIP
1247 014522 005237 003356                PASNEW: INC    PASNUM    ;ELSE BUMP PASS COUNT
1248 014526 012737 003154 003154                MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
1249 014534 013737 002012 003074                MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
1250 014542 012737 177777 003360                MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL
1251 014550 005237 003360                SETDON: INC    PSETNM    ;NEXT SET OF PARAMETERS
1252 014554 005337 003074                DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
1253 014560 062737 000002 003154                ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
1254 014566 013700 003360                MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
1255 014572 012702 003026                MOV     #RLBAS,R2       ;GET POINTER TO RL11 BASE ADDRESS
1256 014576                GPHARD  R0,R1
(3) 014576 104442                TRAP    C$GPHRD
(3) 014600 010001                MOV     R0,R1
1257 014602                BCOMPLETE 7$           ;SKIP IF GOOD PARAM
(2) 014602 103406                BCS     7$
1258 014604 005737 003366                TST     PWRFLG          ;RECENT POWER FAILURE
1259 014610 001741                BEQ     NXPAS          ;NO
1260 014612 005337 003366                DEC     PWRFLG          ;ACCOUNT FOR DRIVE
1261 014616 000736                BR      NXPAS
1262                ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
1263 014620 012122                7$:     MOV     (R1)+,(R2)+ ;STORE CSR
1264 014622 012122                MOV     (R1)+,(R2)+    ;STORE VECTOR
1265 014624 005721                TST     (R1)+          ;BUMP PAST PRIORITY
1266 014626 012137 002276                MOV     (R1)+,T.DRIVE  ;STORE DRIVE TYPE
1267 014632 012122                MOV     (R1)+,(R2)+
1268 014634 022737 000001 002276                CMP     #1,T.DRIVE
1269 014642 001426                BEQ     65$
1270                ;INITIALIZE RL02 PARAMETERS
1271 014644 012737 000776 002306                MOV     #510.,NXTHL

```



```

1272 014652 012737 000777 002302      MOV      #511.,HLMTW
1273 014660 012737 001000 002310      MOV      #512.,GBND
1274 014666 012737 177600 002312      MOV      #177600,CAMSK
1275 014674 012737 177600 002314      MOV      #177600,DIRMSK
1276 014702 012737 177600 002316      MOV      #177600,HDCYL
1277 014710 012737 177000 002304      MOV      #177000,CLRBYT
1278 014716 000425
1279                                     BR      PWCON
1280 014720 012737 000377 002302      ;INITIALIZE RL01 PARAMETERS
1281 014726 012737 000400 002310      65$:    MOV      #255.,HLMTW
1282 014734 012737 077600 002312      MOV      #256.,GBND
1283 014742 012737 077600 002314      MOV      #77600,CAMSK
1284 014750 012737 077600 002316      MOV      #77600,DIRMSK
1285 014756 012737 000376 002306      MOV      #77600,HDCYL
1286 014764 012737 177400 002304      MOV      #254.,NXTHL
1287                                     MOV      #177400,CLRBYT
1288 014772 032737 020000 014206      PWCON:  BIT      #HICYL,MISWIW
1289 015000 001003
1290 015002 013737 002302 014212      BNE     1$
1291 015010                                     MOV      HLMTW,HILIMW
1292 (7) 015010 012746 000340      1$:    SETVEC  RLVEC,#INTHLR,#340      ;SET UP INTERRUPT VECTOR FOR DRIVE
1293 (6) 015014 012746 016056      MOV      #340,-(SP)
1294 (5) 015020 013746 003030      MOV      #INTHLR,-(SP)
1295 (4) 015024 012746 000003      MOV      RLVEC,-(SP)
1296 (3) 015030 104437      MOV      #3,-(SP)
1297 (2) 015032 062706 000010      TRAP    C$$VEC
1298 015036      ADD     #10,SP
1299 (3) 015036 012700 000000      SETPRI  #0      ;SET PRIORITY TO 0 TO ALLOW INTERRUPTS
1300 (3) 015042 104441      MOV     #0,R0
1301 015044 013702 003026      TRAP    C$$PRI
1302                                     MOV     RLBAS,R2      ;SET RL11 BASE ADDRESS POINTER
1303
1304
1305
1306
1307
1308 015050      MANUAL
1309 (3) 015050 104450      TRAP    C$MANI      ;MANUAL INTERVENTION ALLOWED?
1310 015052      BNCOMPLETE 4$
1311 (2) 015052 103004      BCC     4$      ;NO
1312
1313 015054 005737 003356      TST     PASNUM
1314 015060 001001      BNE     4$      ;YES, CHECK PASS NUMBER
1315 015062 000520      BR      8$      ;NOT FIRST PASS, NEED DRIVE UP
1316                                     ;FIRST PASS, PROGRAM WILL INSTRUCT USER
1317
1318                                     ;CHECK IF POWER FAILURE WAIT IS NEEDED
1319 015064 005737 003366      4$:    TST     PWRFLG
1320 015070 001515      BEQ     8$      ;NEEDED?
1321                                     ;NO, SKIP
1322
1323 015072 013705 003032      MOV     RLDRV,R5      ;DRIVE SELECT
1324 015076 052705 000200      BIS     #CRDYMSK,R5   ;SET CRDY
1325 015102 010562 000000      MOV     R5,RLCS(R2)   ;SELECT DRIVE
1326 015106 012701 000170      MOV     #120.,R1     ;INITIALIZE WAIT COUNT
1327 015112 032762 000001 000000 9$:    BIT     #DRDYMSK,RLCS(R2) ;DRIVE UP YET
1328 015120 001101      BNE     8$      ;YES START TEST
1329
1330 015122      WAITMS #10.
1331 (3) 015140 012727 000372      MOV     #N250.,(PC)+ ;WAIT A SECOND

```

```
(3) 015144 000000 .WORD 0
(3) 015146 013727 002116 MOV L$DLY,(PC)+
(3) 015152 000000 .WORD 0
(3) 015154 005367 177772 DEC -6(PC)
(3) 015160 001375 BNE .-4
(3) 015162 005367 177756 DEC -22(PC)
(3) 015166 001367 BNE .-20
1319 015176 005301 DEC R1 ;SIXTY GONE BY
1320 015200 001344 BNE 9$ ;NO
1321 015202 PRINTF #FMT24,#NOPWR ;REPORT 'DRV DID NOT REC'R FROM PWR FAIL'
(8) 015202 012746 006056 MOV #NOPWR,-(SP)
(7) 015206 012746 012252 MOV #FMT24,-(SP)
(6) 015212 012746 000002 MOV #2,-(SP)
(3) 015216 010600 MOV SP,R0
(4) 015220 104417 TRAP C$PNTF
(4) 015222 062706 000006 ADD #6,SP
1322 015226 PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT DRIVE UNIBUS
(11) 015226 005046 CLR -(SP)
(11) 015230 153716 003033 BISB RLDRV+1,(SP)
(10) 015234 012746 006051 MOV #DRVNAM,-(SP)
(9) 015240 013746 003026 MOV RLBAS,-(SP)
(8) 015244 012746 006040 MOV #BASADD,-(SP)
(7) 015250 012746 011446 MOV #FMT5,-(SP)
(6) 015254 012746 000005 MOV #5,-(SP)
(3) 015260 010600 MOV SP,R0
(4) 015262 104417 TRAP C$PNTF
(4) 015264 062706 000014 ADD #14,SP
1323 ;/ADDRESS AND DRIVE NUMBER
1324 015270 PRINTF #FMT3 ;NEW LINE
(7) 015270 012746 011432 MOV #FMT3,-(SP)
(6) 015274 012746 000001 MOV #1,-(SP)
(3) 015300 010600 MOV SP,R0
(4) 015302 104417 TRAP C$PNTF
(4) 015304 062706 000004 ADD #4,SP
1325 015310 DODU PSETNM ;DO DROP UNIT ON DRIVE
(3) 015310 013700 003360 MOV PSETNM,R0
(3) 015314 104451 TRAP C$DODU
1326 015316 DOCLN ;INVOKE CLEAN-UP CODE TO RESTORE DRIVE
(3) 015316 104444 TRAP C$DCLN
1327 ;/TO STATIC STATE
1328 015320 005037 003140 CLR ERRVEC ;CLEAR ERROR VECTOR
1329
1330 015324 8$:
1331
1332 015324 ENDINIT
(3) 015324 L10015:
(3) 015324 104411 TRAP C$INIT
1333
1334 015326 ENDMOD
1335
1336
```


1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347 015326
 1348 015326 005037 003364
 1349 015332
 (7) 015332 012746 000340
 (6) 015336 012746 016124
 (5) 015342 013746 003140
 (4) 015346 012746 000003
 (3) 015352 104437
 (2) 015354 062706 000010
 1350
 1351
 1352 015360 013702 003026
 1353 015364 005762 000000
 1354 015370 005737 003364
 1355 015374 001447
 1356 015376
 (8) 015376 012746 006730
 (7) 015402 012746 012252
 (6) 015406 012746 000002
 (3) 015412 010600
 (4) 015414 104417
 (4) 015416 062706 000006
 1357 015422
 (11) 015422 005046
 (11) 015424 153716 003033
 (10) 015430 012746 006051
 (9) 015434 013746 003026
 (8) 015440 012746 006040
 (7) 015444 012746 011446
 (6) 015450 012746 000005
 (3) 015454 010600
 (4) 015456 104417
 (4) 015460 062706 000014
 1358
 1359 015464
 (7) 015464 012746 011432
 (6) 015470 012746 000001
 (3) 015474 010600
 (4) 015476 104417
 (4) 015500 062706 000004
 1360 015504
 (3) 015504 013700 003360
 (3) 015510 104451
 1361 015512 000460
 1362 015514 013705 003032
 1363 015520 052705 000200
 1364 015524 010562 000000

.SBTTL AUTO DROP SECTION

:THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
 : 'ADR' FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
 : CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
 : IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
 : DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
 : AFTER WHICH THE NEXT DRIVE IS ACCESSED.

BGNAUTO

```

CLR      TRPFLG      ;CLEAR TRAP FLAG
SETVEC  ERRVEC,#TRPHAN,#340 ;SET UP TRAP VECTOR TO DETECT
MOV     #340,-(SP)
MOV     #TRPHAN,-(SP)
MOV     ERRVEC,-(SP)
MOV     #3,-(SP)
TRAP    C$SVEC
ADD     #10,SP

                                           ;/NON-EXISTENT CONTROLLER UNIBUS
                                           ;/ADDRESS
MOV     RLBAS,R2      ;GET RL11 BASE ADDRESS
TST     RLCS(R2)     ;ACCESS DRIVE CONTROLLER UNIBUS ADDRESS
TST     TRPFLG       ;DID TRAP OCCUR?
BEQ     1$           ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
PRINTF  #FMT24,#NOCTLR ;ELSE, PRINT MSG. 'DRV DROPPED - NO CNTLR'
MOV     #NOCTLR,-(SP)
MOV     #FMT24,-(SP)
MOV     #2,-(SP)
MOV     SP,R0
TRAP    C$PNTF
ADD     #6,SP
PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
CLR     -(SP)
BISB   RLDRV+1,(SP)
MOV     #DRVNAM,-(SP)
MOV     RLBAS,-(SP)
MOV     #BASADD,-(SP)
MOV     #FMT5,-(SP)
MOV     #5,-(SP)
MOV     SP,R0
TRAP    C$PNTF
ADD     #14,SP

                                           ;PRINT DRIVE INFORMATION
PRINTF  #FMT3
MOV     #FMT3,-(SP)
MOV     #1,-(SP)
MOV     SP,R0
TRAP    C$PNTF
ADD     #4,SP
DODU   PSETNM      ;DO DROP UNIT ON DRIVE
MOV     PSETNM,R0
TRAP    C$DODU
BR     2$           ;BRANCH TO EXIT
1$:    MOV     RLDRV,R5 ;ELSE, GET DRIVE NUMBER
BIS     #CRDYMSK,R5 ;SET CONTROLLER READY
MOV     R5,RLCS(R2) ;LOAD IN THE DRIVE NUMBER
  
```

```
1365 015530 032762 000001 000000 BIT #DRDYMSK,RLCS(R2) ;IS DRIVE READY?
1366 015536 001046 BNE 2$ ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
1367 015540 PRINTF #FMT24,#NOTRDY ;PRINT MSG. 'DRV DROPPED - NOT RDY'
(8) 015540 012746 006757 MOV #NOTRDY,-(SP)
(7) 015544 012746 012252 MOV #FMT24,-(SP)
(6) 015550 012746 000002 MOV #2,-(SP)
(3) 015554 010600 MOV SP,R0
(4) 015556 104417 TRAP C$PNTF
(4) 015560 062706 000006 ADD #6,SP
1368 ;/WITH 'READY''
1369 015564 PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 015564 005046 CLR -(SP)
(11) 015566 153716 003033 BISB RLDRV+1,(SP)
(10) 015572 012746 006051 MOV #DRVNAM,-(SP)
(9) 015576 013746 003026 MOV RLBAS,-(SP)
(8) 015602 012746 006040 MOV #BASADD,-(SP)
(7) 015606 012746 011446 MOV #FMT5,-(SP)
(6) 015612 012746 000005 MOV #5,-(SP)
(3) 015616 010600 MOV SP,R0
(4) 015620 104417 TRAP C$PNTF
(4) 015622 062706 000014 ADD #14,SP
1370 ;PRINT DRIVE INFORMATION
1371 015626 PRINTF #FMT3
(7) 015626 012746 011432 MOV #FMT3,-(SP)
(6) 015632 012746 000001 MOV #1,-(SP)
(3) 015636 010600 MOV SP,R0
(4) 015640 104417 TRAP C$PNTF
(4) 015642 062706 000004 ADD #4,SP
1372 015646 DODU PSETNM ;DO DROP UNIT ON DRIVE
(3) 015646 013700 003360 MOV PSETNM,R0
(3) 015652 104451 TRAP C$DODU
1373 015654 2$: CLRVEC ERRVEC ;RELEASE THE ERROR VECTOR
(3) 015654 013700 003140 MOV ERRVEC,R0
(3) 015660 104436 TRAP C$CVEC
1374 015662 ENDAUTO
(3) 015662 L10016:
(3) 015662 104461 TRAP C$AUTO
1375
1376
1377
1378
```


Line	Address	Offset	Code	Label	Instruction	Comment
1380	015664		BGNMOD	CLNCODE		
1381	015664		BGNCLN			
1382						
1383	015664				SETVEC ERRVEC,#TRPHAN,#340	
(7)	015664	012746	000340		MOV #340,-(SP)	
(6)	015670	012746	016124		MOV #TRPHAN,-(SP)	
(5)	015674	013746	003140		MOV ERRVEC,-(SP)	
(4)	015700	012746	000003		MOV #3,-(SP)	
(3)	015704	104437			TRAP C\$SVEC	
(2)	015706	062706	000010		ADD #10,SP	
1384						
1385	015712				SETPRI #7	;SET PRIORITY TO 7
(3)	015712	012700	000007		MOV #7,R0	
(3)	015716	104441			TRAP C\$SPRI	
1386	015720	032762	000200	000000	2\$: BIT #CRDYMSK,RLCS(R2)	;TEST IF CONTROLLER READY
1387	015726	001407			BEQ 3\$;NO LOOP UNTIL READY
1388	015730	053762	003032	000000	BIS RLDRV,RLCS(R2)	;SET DRIVE NUMBER
1389	015736	032762	000001	000000	BIT #DRDYMSK,RLCS(R2)	;TEST IF DRIVE BUSY
1390	015744	001026			BNE 5\$;NO - SKIP
1391	015746				3\$: WAITMS #3	;WAIT 300 MS
(3)	015764	012727	000372		MOV ##250.,(PC)+	
(3)	015770	000000			.WORD 0	
(3)	015772	013727	002116		MOV LSDLY,(PC)+	
(3)	015776	000000			.WORD 0	
(3)	016000	005367	177772		DEC -6(PC)	
(3)	016004	001375			BNE -4	
(3)	016006	005367	177756		DEC -22(PC)	
(3)	016012	001367			BNE -20	
1392	016022				5\$: CLRVEC RLVEC	;RELEASE DRIVE VECTOR
(3)	016022	013700	003030		MOV RLVEC,R0	
(3)	016026	104436			TRAP C\$CVEC	
1393	016030	005737	003366		TST PWRFLG	;PWR FAIL SET
1394	016034	001402			BEQ 7\$;NO
1395	016036	005337	003366		DEC PWRFLG	
1396	016042				7\$: CLRVEC ERRVEC	
(3)	016042	013700	003140		MOV ERRVEC,R0	
(3)	016046	104436			TRAP C\$CVEC	
1397	016050				ENDCLN	
(3)	016050				L10017:	
(3)	016050	104412			TRAP C\$CLEAN	
1398						
1399	016052				BGNDU	
1400	016052	000240			NOP	
1401	016054				ENDDU	
(3)	016054				L10020:	
(3)	016054	104453			TRAP C\$DU	
1402						
1403	016056				ENDMOD	
1404						
1405						
1406						

```
1408 .SBTTL INTERRUPT SERVICE ROUTINES
1409
1410 016056 BGNSRV INTHLR
1411 ;INTERRUPT HANDLER FOR DRIVE ABORTS WAIT TIMER AND STORES ALL RL11 REGISTERS
1412 016056 005037 003142 CLR DLYCNT ;CLEAR UNELAPSED DELAY COUNT
1413 016062 012237 003044 MOV (R2)+,T.CS ;STORE RL REGISTERS
1414 016066 012237 003046 MOV (R2)+,T.BA
1415 016072 012237 003050 MOV (R2)+,T.DA
1416 016076 011237 003052 MOV (R2),T.MP
1417 016102 012737 177777 003006 MOV #-1,DONE ;SET DONE FLAG
1418 016110 013702 003026 MOV RLBAS,R2 ;RESTORE R2
1419 016114 ENDSRV
(3) 016114 L10021:
(2) 016114 000002 RTI
1420
1421 ;INTERRUPT SERVICE ROUTINE FOR P-CLOCK DECREMENTS DELAY COUNTER AT 100-MICROSECOND
1422 ;TIME INTERVALS
1423 016116 BGNSRV CLKINT
1424 016116 005337 003142 DEC DLYCNT ;DECREMENT CLOCK DELAY COUNTER
1425 016122 ENDSRV
(3) 016122 L10022:
(2) 016122 000002 RTI
1426
1427 ;INTERRUPT SERVICE ROUTINE SETS TRAP FLAG WHEN A NON-EXISTENT UNIBUS ADDRESS IS
1428 ;ACCESSED
1429 016124 BGNSRV TRPHAN
1430 016124 005237 003364 INC TRPFLG ;INDICATE THAT TRAP OCCURRED
1431 016130 ENDSRV
(3) 016130 L10023:
(2) 016130 000002 RTI
1432
1433
```



```
1435      .SBTTL  GLOBAL SUBROUTINES
1436
1437 016132  BGNMOD  GLBSUB
1438
1439      :      ERROR LIMIT CHECKING ROUTINE
1440      :      DROPS DRIVE IF ERROR LIMIT EXCEEDED
1441 016132 027737 165016 014216  CKERLM: CMP      @ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED
1442 016140 002453      BLT      1$ ;NO - SKIP
1443 016142      INLOOP      ;CHECK IF IN ERROR LOOP
1444 (3) 016142 104420      TRAP      C$INLP
1444 (2) 016144 103451      BCOMPLETE 1$ ;YES - SKIP
1445 016146      PRINTF #FMT25,ERLIMW,#MEXERS ;PRINT MSG. 'OVER ERROR LIMIT - UNIT DROPPED'
1445 (9) 016146 012746 011104      MOV      #MEXERS,-(SP)
1445 (8) 016152 013746 014216      MOV      ERLIMW,-(SP)
1445 (7) 016156 012746 012257      MOV      #FMT25,-(SP)
1445 (6) 016162 012746 000003      MOV      #3,-(SP)
1445 (3) 016166 010600      MOV      SP,RO
1445 (4) 016170 104417      TRAP      C$PNTF
1445 (4) 016172 062706 000010      ADD      #10,SP
1446 016176      PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;PRINT DRIVE INFORMATION
1446 (11) 016176 005046      CLR      -(SP)
1446 (11) 016200 153716 003033      BISB     RLDRV+1,(SP)
1446 (10) 016204 012746 006051      MOV      #DRVNAM,-(SP)
1446 (9) 016210 013746 003026      MOV      RLBAS,-(SP)
1446 (8) 016214 012746 006040      MOV      #BASADD,-(SP)
1446 (7) 016220 012746 011446      MOV      #FMT5,-(SP)
1446 (6) 016224 012746 000005      MOV      #5,-(SP)
1446 (3) 016230 010600      MOV      SP,RO
1446 (4) 016232 104417      TRAP      C$PNTF
1446 (4) 016234 062706 000014      ADD      #14,SP
1447 016240      PRINTF #FMT3
1447 (7) 016240 012746 011432      MOV      #FMT3,-(SP)
1447 (6) 016244 012746 000001      MOV      #1,-(SP)
1447 (3) 016250 010600      MOV      SP,RO
1447 (4) 016252 104417      TRAP      C$PNTF
1447 (4) 016254 062706 000004      ADD      #4,SP
1448 016260      DODU     PSETNM ;DROP DRIVE
1448 (3) 016260 013700 003360      MOV      PSETNM,RO
1448 (3) 016264 104451      TRAP      C$DODU
1449 016266      DOCLN   ;GO TO CLEAN UP
1449 (3) 016266 104444      TRAP      C$DCLN
1450 016270 000207      RTS      PC
1451
1452      :      READ AND STORE ALL RL11 REGISTERS
1453 016272 016237 000000 003044  READRL: MOV      RLCSR(R2),T.CS ;GET CS REG
1454 016300 016237 000002 003046      MOV      RLBA(R2),T.BA ;GET BUS ADDRESS REG
1455 016306 016237 000004 003050      MOV      RLDA(R2),T.DA ;GET DISK ADDRESS
1456 016314 016237 000006 003052      MOV      RLMP(R2),T.MP ;GET MULTI-PURPOSE REG
1457 016322 000207      RTS      PC ;RETURN
1458
1459      :      WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
1460 016324 011646      WAITIN: MOV      (SP),-(SP) ;MAKE ROOM FOR ERROR POINTER
1461 016326 005066 000002      CLR      2(SP) ;CLEAR FOR POINTER
1462 016332 032762 000200 000000      BIT      #CRDYMSK,RLCSR(R2) ;TEST IF CONTROLLER READY
1463 016340 001420      BEQ     4$ ;NO - SKIP TO WAIT
```

```

1464 016342 004737 016272      JSR    PC,READRL      ;READ ALL RL REGS
1465 016346 005737 003006      TST    DONE          ;TEST IF INTERRUPT OCCURRED
1466 016352 001453              BEQ    5$            ;NO - GO SET NO INTERRUPT ERR FLAG
1467 016354 012766 006176 000002 1$:  MOV    #MTOSLOW,2(SP) ;ELSE SET TOO SLOW ERROR POINTER
1468 016362 032737 002000 003044  BIT    #OPIERR,T.CS  ;TEST IF OPI SET
1469 016370 001403              BEQ    2$            ;NO - SKIP
1470 016372 012766 006216 000002  MOV    #MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
1471 016400 000207              RTS    PC            ;RETURN
1472 016402 012737 000001 003142 4$:  MOV    #1,DLYCNT     ;INITIALIZE DELAY COUNT
1473 016410 006337 003142      ASL    DLYCNT        ;MULTIPLY BY 2
1474 016414 006337 003142      ASL    DLYCNT        ;MULTIPLY BY 2 AGAIN
1475 016420 012727 000012      MOV    #10.,(PC)+   ;IMPLEMENT TIME DELAY LOOP
1476 016424 000000      .WORD 0
1477 016426 013727 002116      MOV    L$DLY,(PC)+
1478 016432 000000      .WORD 0
1479 016434 005367 177772      DEC    -6(PC)
1480 016440 001375              BNE    -.4
1481 016442 005367 177756      DEC    -22(PC)
1482 016446 001367              BNE    -.20
1483 016450 032762 000200 000000  BIT    #CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
1484 016456 001006              BNE    3$            ;YES - SKIP
1485 016460 004737 016272      JSR    PC,READRL     ;READ RL REGS
1486 016464 012766 006271 000002  MOV    #MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
1487 016472 000742              BR     2$            ;SKIP
1488 016474 005737 003006      3$:  TST    DONE          ;ELSE CHECK IF INTERRUPT OCCURRED
1489 016500 001325              BNE    1$            ;YES - SKIP TO SET TOO SLOW
1490 016502 004737 016272      5$:  JSR    PC,READRL     ;READ RL REGS
1491 016506 012766 006236 000002  MOV    #MNOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
1492 016514 000731              BR     2$            ;GO TO RETURN
1493
1494      ; OPERATION AND TEST INITIALIZE ROUTINE
1495 016516 005037 003004      TSTINT: CLR    OPFLAG      ;CLEAR OPERATION FLAGS
1496 016522 105037 003363      CLR    NOERCT       ;RESET INHIBIT ERROR COUNTING
1497 016526 005037 003014      CLR    MORECE       ;RESET MORE COMPARE ERRORS
1498 016532 000207      RTS    PC
1499
1500      ; GET STATUS AND GET STATUS WITH RESET ROUTINE
1501 016534 013746 003126      GSTATR: MOV    TEMP4,-(SP)  ;STORE TEMP4
1502 016540 012737 000013 003126  MOV    #GETSTAT!DRSET,TEMP4 ;SET FOR RESET
1503 016546 000412              BR     GSTATG
1504 016550 013746 003126      GSTATC: MOV    TEMP4,-(SP)  ;STORE TEMP4
1505 016554 012737 000003 003126  MOV    #GETSTAT,TEMP4  ;SET FOR NO RESET
1506 016562 000404              BR     GSTATG
1507 016564 013746 003126      GSTAT:  MOV    TEMP4,-(SP)  ;STORE TEMP4
1508 016570 005037 003126      CLR    TEMP4        ;SET FOR SAVE L. AND T. REGS
1509 016574 010346      GSTATG: MOV    R3,-(SP)     ;STORE R3
1510 016576 013703 003002      MOV    SSINDX,R3    ;GET SUBROUTINE INDEX
1511 016602 005723              TST    (R3)+        ;BUMP IT FOR NEXT ENTRY
1512 016604 016663 000004 002404  MOV    4(SP),SUBSTK(R3) ;INSERT THIS CALL
1513 016612 162763 000004 002404  SUB    #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1514 016620 010337 003002      MOV    R3,SSINDX    ;STORE IT BACK
1515 016624 010046      MOV    R0,-(SP)     ;STORE R0
1516 016626 010146      MOV    R1,-(SP)     ;STORE R1
1517 016630 012737 000002 003016  MOV    #2,ERRSWI    ;SET FOR NO ERROR RETURN
1518 016636 032737 000010 003126  BIT    #DRSET,TEMP4 ;TEST IF DRIVE RESET
1519 016644 001523              BEQ    11$          ;NO - SKIP

```



```

1520 016646 032762 040000 000000 BIT #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
1521 016654 001426 BEQ 49$ ;NO - SKIP
1522 016656 WAITMS #1 ;WAIT FOR DRIVE TO SETTLE
(3) 016674 012727 000372 MOV ##250.,(PC)+
(3) 016700 000000 .WORD 0
(3) 016702 013727 002116 MOV L$DLY,(PC)+
(3) 016706 000000 .WORD 0
(3) 016710 005367 177772 DEC -6(PC)
(3) 016714 001375 BNE -4
(3) 016716 005367 177756 DEC -22(PC)
(3) 016722 001367 BNE -20
1523 016732 012701 000030 49$: MOV #24.,R1 ;INITIALIZE WAIT COUNTER
1524 016736 004737 016564 50$: JSR PC,GSTAT ;GET DRIVE STATUS
1525 016742 017570 3$
1526 016744 032737 000001 003044 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1527 016752 001076 BNE 5$ ;YES - GO DO CLEAR
1528 016754 032737 000020 003052 BIT #HOSTAT,T.MP ;ELSE TEST IF HEADS OUT
1529 016762 001010 BNE 51$ ;YES - BYPASS RELOAD WAIT FLAG SETTING
1530 016764 032737 144000 003052 BIT #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
1531 ;THAT CAUSED HEADS TO
1532 ;UNLOAD
1533 016772 001466 BEQ 5$ ;NO - SKIP
1534 016774 052737 040000 003004 BIS #RELDWT,OPFLAG ;ELSE SET WAIT FLAG
1535 017002 000462 BR 5$ ;SKIP TO CLEAR
1536 017004 032737 040000 003044 51$: BIT #DRVERR,T.CS ;TEST IF DRIVE ERROR NOW
1537 017012 001056 BNE 5$ ;YES - SKIP TO CLEAR
1538 017014 WAITMS #1 ;WAIT FOR DRIVE TO GET ERROR, READY, OR HEADS OUT
(3) 017032 012727 000372 MOV ##250.,(PC)+
(3) 017036 000000 .WORD 0
(3) 017040 013727 002116 MOV L$DLY,(PC)+
(3) 017044 000000 .WORD 0
(3) 017046 005367 177772 DEC -6(PC)
(3) 017052 001375 BNE -4
(3) 017054 005367 177756 DEC -22(PC)
(3) 017060 001367 BNE -20
1539 017070 005301 DEC R1 ;DEC WAIT COUNTER
1540 017072 001321 BNE 50$ ;IF NOT DONE, LOOP
1541 017074 012703 010762 MOV #MUNDEF,R3 ;MESSAGE FOR UNDEFINED STATE
1542 017100 ERRHRD 10001.,ERR1
(4) 017100 104456 TRAP C$ERRHD
(5) 017102 023421 .WORD 10001
(5) 017104 000000 .WORD 0
(5) 017106 012344 .WORD ERR1
1543 017110 000137 017564 JMP 14$ ;EXIT
1544 017114 005737 003126 11$: TST TEMP4 ;TEST IF SAVE REGISTERS
1545 017120 001013 BNE 5$ ;NO SKIP
1546 017122 012701 000004 MOV #4,R1 ;SET SAVE COUNT
1547 017126 012703 003044 MOV #L.MP+2,R3 ;SET ADDRESS OF FIRST SAVE
1548 017132 014346 8$: MOV -(R3),-(SP) ;PUT REG ON STACK
1549 017134 005301 DEC R1 ;DEC COUNT
1550 017136 001375 BNE 8$ ;LOOP UNTIL ALL SAVED
1551 017140 012737 000003 003040 MOV #GETSTAT,L.DA ;SET FOR GET STATUS
1552 017146 000403 BR 6$ ;SKIP
1553 017150 013737 003126 003040 5$: MOV TEMP4,L.DA ;INSERT PRESET FOR STATUS
1554 017156 6$:
1555 017156 005037 003006 CLR DONE ;CLEAR INTERRUPT FLAG
    
```


1556	017162	013737	003032	003034		MOV	RLDRV,L.CS	:SET UP TO GET STATUS
1557	017170	042737	002000	003034		BIC	#BIT10,L.CS	:CLEAR FOR DRIVE 4 - 7 SPEC'D
1558	017176	052737	000104	003034		BIS	#GTSTAT,L.CS	
1559	017204	013762	003040	000004		MOV	L.DA,RLDA(R2)	:LOAD RL REGS
1560	017212	013762	003034	000000		MOV	L.CS,RLCSR(R2)	:LOAD CS REG
1561	017220					WAITUS	#1	:WAIT 100 US FOR INTERRUPT
(3)	017220	012727	000001			MOV	###1,(PC)+	
(3)	017224	000000				.WORD	0	
(3)	017226	013727	002116			MOV	LSDLY,(PC)+	
(3)	017232	000000				.WORD	0	
(3)	017234	005367	177772			DEC	-6(PC)	
(3)	017240	001375				BNE	.-4	
(3)	017242	005367	177756			DEC	-22(PC)	
(3)	017246	001367				BNE	.-20	
1562	017250	005737	003006			TST	DONE	:CHECK IF INTERRUPT OCCURRED
1563	017254	001534				BEQ	1\$:NO - SKIP
1564	017256	013737	003052	003060	4\$:	MOV	T.MP,T.STAT	:STORE MP REGISTER
1565	017264	042737	177770	003060		BIC	#*C<STAMSK>,T.STAT	:CLEAR ALL BUT STATE
1566	017272	032737	000010	003040		BIT	#DRSET,L.DA	:TEST IF RESET WAS SPECIFIED
1567	017300	001533				BEQ	3\$:NO - SKIP TO EXIT
1568	017302	032737	040000	003004		BIT	#RELDWT,OPFLAG	:TEST IF RELOAD WAIT FLAG SET
1569	017310	001450				BEQ	12\$:NO - SKIP
1570	017312	012701	000144			MOV	#100.,R1	:INITIALIZE WAIT COUNTER
1571	017316	032762	000001	000000	13\$:	BIT	#DRDYMSK,RLCS(R2)	:TEST IF DRIVE NOW READY
1572	017324	001042				BNE	12\$:YES - SKIP
1573	017326					WAITMS	#1	:CALL WAIT
(3)	017344	012727	000372			MOV	##250.,(PC)+	
(3)	017350	000000				.WORD	0	
(3)	017352	013727	002116			MOV	LSDLY,(PC)+	
(3)	017356	000000				.WORD	0	
(3)	017360	005367	177772			DEC	-6(PC)	
(3)	017364	001375				BNE	.-4	
(3)	017366	005367	177756			DEC	-22(PC)	
(3)	017372	001367				BNE	.-20	
1574	017402	005301				DEC	R1	:DEC COUNT
1575	017404	001344				BNE	13\$:LOOP IF NOT 0
1576	017406	004737	016564			JSR	PC,GSTAT	:GET DRIVE STATUS
1577	017412	017570				3\$:ERROR RETURN
1578	017414	012703	011027			MOV	#MRLFAL,R3	:SET RESULT MESSAGE POINTER
1579	017420					ERRHRD	10003.,,ERR1	
(4)	017420	104456				TRAP	C\$ERRHRD	
(5)	017422	023423				.WORD	10003	
(5)	017424	000000				.WORD	0	
(5)	017426	012344				.WORD	ERR1	
1580	017430	000455				BR	14\$:GO TO EXIT
1581	017432				12\$:	WAITUS	#5	:WAIT
(3)	017432	012727	000005			MOV	###5,(PC)+	
(3)	017436	000000				.WORD	0	
(3)	017440	013727	002116			MOV	LSDLY,(PC)+	
(3)	017444	000000				.WORD	0	
(3)	017446	005367	177772			DEC	-6(PC)	
(3)	017452	001375				BNE	.-4	
(3)	017454	005367	177756			DEC	-22(PC)	
(3)	017460	001367				BNE	.-20	
1582	017462	004737	016564			JSR	PC,GSTAT	:GET DRIVE STATUS
1583	017466	017570				3\$		


```

1584 017470 032737 100000 003044 BIT #ANYERR,T.CS :TEST IF ANY ERROR
1585 017476 001434 BEQ 3$ :NO - SKIP
1586 017500 032737 001000 003052 BIT #VCSTAT,T.MP :CHECK IF VOLUME CHECK RESET
1587 017506 001403 BEQ 7$ :YES SKIP
1588 017510 012703 006325 MOV #VCNRST,R3 :SET REASON POINTER
1589 017514 000417 BR 2$ :EXIT
1590 017516 032737 040000 003044 7$: BIT #DRVERR,T.CS :CHECK IF DRIVE ERROR
1591 017524 001405 BEQ 9$ :NO - SKIP
1592 017526 ERRHRD 10004...ERR6
(4) 017526 104456 TRAP C$ERHRD
(5) 017530 023424 .WORD 10004
(5) 017532 000000 .WORD 0
(5) 017534 012646 .WORD ERR6
1593 017536 000412 BR 14$ :EXIT
1594 017540 012703 006346 9$: MOV #UNXERR,R3 :SET REASON POINTER
1595 017544 000403 BR 2$ :EXIT
1596 017546 004737 016324 1$: JSR PC,WAITIN :WAIT FOR INTERRUPT
1597 017552 012603 MOV (SP)+,R3 :STORE REASON POINTER FOR RETURN
1598 017554 2$: ERRHRD 10002...ERR1
(4) 017554 104456 TRAP C$ERHRD
(5) 017556 023422 .WORD 10002
(5) 017560 000000 .WORD 0
(5) 017562 012344 .WORD ERR1
1599 017564 005037 003016 14$: CLR ERRSWI :CLEAR FOR ERROR RETURN
1600 017570 005737 003126 3$: TST TEMP4 :TEST IF REGISTERS WERE SAVED
1601 017574 001007 BNE 22$ :NO - SKIP
1602 017576 012703 003034 MOV #L.CS,R3 :SET POINTER TO RESTORE
1603 017602 012701 000004 MOV #4,R1 :SET REGISTER COUNT
1604 017606 012623 20$: MOV (SP)+,(R3)+ :RESTORE REG
1605 017610 005301 DEC R1 :DEC COUNT
1606 017612 001375 BNE 20$ :LOOP UNTIL ALL ARE RESTORED
1607 017614 162737 000002 003002 22$: SUB #2,SSINDX :REMOVE ENTRY FROM SUBROUTINE STACK
1608 017622 012601 MOV (SP)+,R1 :RESTORE R1
1609 017624 012600 MOV (SP)+,R0 :RESTORE R0
1610 017626 012603 MOV (SP)+,R3 :RESTORE R3
1611 017630 012637 003126 MOV (SP)+,TEMP4 :RESTORE TEMP4
1612 017634 005737 003016 TST ERRSWI :TEST IF ERROR RETURN
1613 017640 001403 BEQ 99$ :YES - SKIP
1614 017642 063716 003016 ADD ERRSWI,(SP) :ADD IN ERROR RETURN
1615 017646 000207 RTS PC
1616 017650 017616 000000 99$: MOV @ (SP),(SP) :SET ERROR RETURN ADDRESS
1617 017654 000207 RTS PC
1618
1619 ;
1620 017656 010346 ;GDRSTA: GET DRIVE STATE ROUTINE
1621 017660 012701 000004 MOV R3,-(SP) :SAVE R3
1622 017664 012703 003044 MOV #4,R1 :INITIALIZE REGISTER SAVE COUNT
1623 017670 014346 1$: MOV #L.MP+2,R3 :INITIALIZE ADDRESS OF FIRST SAVE
1624 017672 005301 MOV -(R3),-(SP) :SAVE REGISTER ON STACK
1625 017674 001375 DEC R1 :DECREMENT REGISTER SAVE COUNT
1626 017676 012737 000003 003040 BNE 1$ :LOOP UNTIL ALL 4 REGISTERS ARE SAVED
1627 MOV #GETSTAT,L.DA :SET UP DISK ADDRESS REGISTER FOR GET STATUS
1628 017704 005037 003006 CLR DONE :/COMMAND
1629 017710 013737 003032 003034 MOV RLDRV,L.CS :CLEAR INTERRUPT FLAG
1630 :SET UP CONTROL STATUS REGISTER WITH
1631 017716 042737 002000 003034 BIC #BIT10,L.CS :/DRIVE NUMBER
:CLEAR FOR DRIVES 4-7 SPECIFIED

```



```

1632 017724 052737 000104 003034      BIS      #GTSTAT,L.CS      ;INITIALIZE CONTROL STATUS REGISTER FOR
1633                                     ;/GET STATUS COMMAND
1634 017732 013762 003040 000004      MOV      L.DA,RLDA(R2)   ;INITIALIZE DISK ADDRESS REGISTER FOR
1635                                     ;/GET STATUS COMMAND
1636 017740 013762 003034 000000      MOV      L.CS,RLCSR(R2) ;LOAD CONTROL STATUS REGISTER TO EXECUTE
1637                                     ;/GET STATUS COMMAND
1638 017746 105762 000000      5$:     TSTB     RLCS(R2)  ;WAIT FOR CONTROLLER READY INDICATING
1639 017752 001775                BEQ      5$              ;/RECEIPT OF GET STATUS COMMAND
1640 017754 005737 003006                TST     DONE            ;INTERRUPT OCCURRED?
1641 017760 001416                BEQ      3$              ;BRANCH IF NOT
1642 017762 013737 003052 003060      MOV      T.MP,T.STAT    ;GET CONTENTS OF MULTI-PURPOSE REGISTER
1643 017770 042737 177770 003060      BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE DRIVE BITS
1644 017776 012703 003034                MOV     #L.CS,R3        ;INITIALIZE POINTER TO RESTORE RL REGISTERS
1645 020002 012701 000004                MOV     #4,R1           ;INITIALIZE REGISTER SAVE COUNT
1646 020006 012623                2$:     MOV     (SP)+,(R3)+ ;RESTORE REGISTERS
1647 020010 005301                DEC     R1              ;DECREMENT REGISTER SAVE COUNT
1648 020012 001375                BNE     2$              ;LOOP UNTIL ALL 4 REGISTERS ARE RESTORED
1649 020014 000402                BR      4$
1650 020016 004737 016324      3$:     JSR     PC,WAITIN   ;WAIT FOR INTERRUPT
1651 020022 012603      4$:     MOV     (SP)+,R3    ;RESTORE R3
1652 020024 000207                RTS     PC              ;RETURN
1653
1654
1655
1656                                     ; XSEEK: SEEK ROUTINE
1657 020026 012737 177777 003120      XSEEK:  MOV     #-1,TEMP1 ;SET SPECIAL TIMING SEEK FLAG
1658 020034 000402                BR      XSEEK1
1659 020036 005037 003120      XSEEK:  CLR     TEMP1    ;CLEAR SPECIAL TIMING SEEK FLAG
1660 020042 010346      XSEEK1: MOV     R3,-(SP)    ;STORE R3
1661 020044 013703 003002                MOV     SSINDX,R3      ;GET SUBROUTINE INDEX
1662 020050 005723                TST     (R3)+          ;BUMP IT FOR NEXT ENTRY
1663 020052 016663 000002 002404      MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
1664 020060 162763 000004 002404      SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
1665 020066 010337 003002                MOV     R3,SSINDX     ;STORE IT BACK
1666 020072 010046                MOV     R0,-(SP)
1667 020074 010146                MOV     R1,-(SP)
1668 020076 010546                MOV     R5,-(SP)      ;STORE REG
1669 020100 012737 000002 003016      MOV     #2,ERRSWI     ;SET FOR NO ERROR RETURN
1670 020106 005037 003076                CLR     DIFAUG        ;CLEAR DIFFERENCE ARGUMENT (FOR SEEKING
1671                                     ; PAST GUARD BAND)
1672 020112 004737 022666                JSR     PC,GETPOS     ;GET PRESENT POSITION
1673 020116 020566      65$
1674 020120 013737 003104 003100      MOV     CURCYL,OLDCYL ;MOVE CURRENT TO OLD CYLINDER
1675 020126 023737 003102 002302      CMP     NEWCYL,HLMTW  ;TEST IF NEW IS GREATER THAN 255
1676 020134 003427                BLE     3$              ;NO - SKIP
1677 020136 163737 002302 003102      SUB     HLMTW,NEWCYL  ;ELSE SUBTRACT 255.
1678 020144 013737 003102 003076      MOV     NEWCYL,DIFAUG ;STORE DIFFERENCE AS ARGUMENT
1679 020152 013737 002302 003102      MOV     HLMTW,NEWCYL ;SET NEWCYL AS 255.
1680 020160 022737 000001 002276      CMP     #1,T.DRIVE
1681 020166 001424                BEQ     6$
1682 020170 162737 000001 003102      SUB     #1,NEWCYL
1683 020176 012737 000001 003110      MOV     #1,DESSGN
1684 020204 012737 000001 003106      MOV     #1,DESDIF
1685 020212 000451                BR      18$
1686 020214 005737 003102      3$:     TST     NEWCYL    ;TEST IF .NEWCYL HAS NEGATIVE VALUE
1687 020220 100007                BPL     6$              ;NO - SKIP

```


1688	020222	005437	003102		NEG	NEWCYL	:ELSE MAKE IT POSITIVE
1689	020226	013737	003102	003076	MOV	NEWCYL,DIFAUG	:AND STORE IT AS ARGUMENT
1690	020234	005037	003102		CLR	NEWCYL	:AND SET NEWCYL TO 0
1691	020240	013705	003104	6\$:	MOV	CURCYL,R5	:COMPUTE DIFFERENCE AND NEW CYLINDER
1692	020244	163705	003102		SUB	NEWCYL,R5	:SUB NEWCYL FROM CURCYL
1693	020250	100005			BPL	13\$:IF DIFF IS POSITIVE - SKIP(REV SEEK)
1694	020252	012737	000001	003110	MOV	#1,DESSGN	:ELSE SET SIGN FOR FORWARD
1695	020260	005405			NEG	R5	:MAKE DIFFERENCE POSITIVE
1696	020262	000402			BR	14\$:SKIP
1697	020264	005037	003110	13\$:	CLR	DESSGN	:SET SIGN FOR REVERSE
1698	020270	010537	003106	14\$:	MOV	R5,DESDIF	:STORE DIFFERENCE
1699	020274	005737	003076		TST	DIFAUG	:IS THERE A DIFFERENCE ARGUMENT
1700	020300	001416			BEQ	18\$:NO - SKIP
1701	020302	023737	003102	002302	CMP	NEWCYL,HLMTW	:CHECK IF NEW CYL IS 255.
1702	020310	001007			BNE	17\$:NO - SKIP
1703	020312	012737	000001	003110	MOV	#1,DESSGN	:ELSE FORCE SIGN FOR FORWARD
1704							: (INNER GUARD BAND)
1705	020320	022737	000001	002276	CMP	#1,T.DRIVE	
1706	020326	001003			BNE	18\$	
1707	020330	063737	003076	003106	17\$:	ADD	DIFAUG,DESDIF
1708	020336				18\$:		
1709	020336	012705	003034		MOV	#L,CS,R5	:GET L REG ADDRESS
1710	020342	012715	000106		MOV	#SEEK,(R5)	:SET FOR SEEK
1711	020346	053715	003032		BIS	RLDRV,(R5)	:INSERT DRIVE NUMBER
1712	020352	042725	002000		BIC	#BIT10,(R5)+	:CLEAR IF DRIVE 4 - 7 SPEC'D
1713	020356	005025			CLR	(R5)+	:CLEAR BUS ADDRESS
1714	020360	013715	003106		MOV	DESDIF,(R5)	:LOAD DIFFERENCE
1715	020364	012700	000007		MOV	#7,R0	:SET TO SHIFT DIFFERENCE
1716	020370	006315			21\$:	ASL	(R5)
1717	020372	005300			DEC	R0	
1718	020374	001375			BNE	21\$:LOOP UNTIL ALIGNED
1719	020376	005737	003110		TST	DESSGN	:TEST SIGN
1720	020402	001402			BEQ	23\$:SKIP IF 0
1721	020404	052715	000004		BIS	#DIRBIT,(R5)	:ELSE INSERT SIGN
1722	020410	005737	003112	23\$:	TST	DESHD	:TEST IF HEAD 0
1723	020414	001402			BEQ	25\$:YES - SKIP
1724	020416	052715	000020		BIS	#HSEL,(R5)	:ELSE SET HEAD BIT
1725	020422	052725	000001	25\$:	BIS	#MBSSET0,(R5)+	:INSERT MARKER BIT
1726	020426	004737	021136		JSR	PC,RDYCHK	:CHECK IF DRIVE READY
1727	020432	020566			65\$		
1728	020434	005037	003006		CLR	DONE	:CLEAR INTERRUPT FLAG
1729	020440	005737	003120		TST	TEMP1	:CHECK IF SPECIAL SEEK FLAG SET
1730	020444	001050			BNE	65\$:YES - SKIP, DO NOT START SEEK
1731	020446	014562	000004		MOV	-(R5),RLDA(R2)	:LOAD RL REGISTERS
1732	020452	014562	000002		MOV	-(R5),RLBA(R2)	
1733	020456	014562	000000		MOV	-(R5),RLCS(R2)	:PERFORM SEEK OPERATION
1734	020462			30\$:	WAITUS	#1	:ALLOW TIME FOR RECEIPT OF SEEK COMMAND
(3)	020462	012727	000001		MOV	###1,(PC)+	
(3)	020466	000000			.WORD	0	
(3)	020470	013727	002116		MOV	LSDLY,(PC)+	
(3)	020474	000000			.WORD	0	
(3)	020476	005367	177772		DEC	-6(PC)	
(3)	020502	001375			BNE	.-4	
(3)	020504	005367	177756		DEC	-22(PC)	
(3)	020510	001367			BNE	.-20	
1735	020512	005737	003006		TST	DONE	:TEST IF INTERRUPT DONE

1736	020516	001012				BNE	32\$:YES - SKIP
1737	020520	004737	016324			JSR	PC, WAITIN		:GO WAIT FOR INTERRUPT
1738	020524	012603				MOV	(SP)+, R3		:GET RESULT MESSAGE POINTER
1739	020526					ERRHRD	10005,,,ERR1		
(4)	020526	104456				TRAP	C\$ERHRD		
(5)	020530	023425				.WORD	10005		
(5)	020532	000000				.WORD	0		
(5)	020534	012344				.WORD	ERR1		
1740	020536	005037	003016			CLR	ERRSWI		:CLEAR FOR ERROR RETURN
1741	020542	000411				BR	65\$		
1742	020544	005737	003044		32\$:	TST	T.CS		:TEST IF ANY ERROR
1743	020550	100006				BPL	65\$:NO - SKIP
1744	020552					ERRHRD	10006,,,ERR6		
(4)	020552	104456				TRAP	C\$ERHRD		
(5)	020554	023426				.WORD	10006		
(5)	020556	000000				.WORD	0		
(5)	020560	012646				.WORD	ERR6		
1745	020562	005037	003016			CLR	ERRSWI		:CLEAR FOR ERROR RETURN
1746	020566	162737	000002	003002	65\$:	SUB	#2, SSINDX		:REMOVE ENTRY FROM SUBROUTINE STACK
1747	020574	012605				MOV	(SP)+, R5		:RESTORE REGISTER
1748	020576	012601				MOV	(SP)+, R1		
1749	020600	012600				MOV	(SP)+, R0		
1750	020602	012603				MOV	(SP)+, R3		:RESTORE R3
1751	020604	005737	003016			TST	ERRSWI		:TEST IF ERROR RETURN
1752	020610	001403				BEQ	99\$:YES - SKIP
1753	020612	063716	003016			ADD	ERRSWI, (SP)		:ADD IN ERROR RETURN
1754	020616	000207				RTS	PC		
1755	020620	017616	000000		99\$:	MOV	@(SP), (SP)		:SET ERROR RETURN ADDRESS
1756	020624	000207				RTS	PC		
1757									
1759	020626	010346				SIMSEK: MOV	R3, -(SP)		:STORE REGISTERS
1760	020630	013703	003002			MOV	SSINDX, R3		:GET SUBROUTINE INDEX
1761	020634	005723				TST	(R3)+		:BUMP IT FOR NEXT ENTRY
1762	020636	016663	000002	002404		MOV	2(SP), SUBSTK(R3)		:INSERT THIS CALL
1763	020644	162763	000004	002404		SUB	#4, SUBSTK(R3)		:ADJUST IT TO CALLING LOCATION
1764	020652	010337	003002			MOV	R3, SSINDX		:STORE IT BACK
1765	020656	010046				MOV	R0, -(SP)		
1766	020660	010446				MOV	R4, -(SP)		
1767	020662	012737	000002	003016		MOV	#2, ERRSWI		:SET FOR NO ERROR RETURN
1768	020670	004737	021136			JSR	PC, RDYCHK		:CHECK IF DRIVE READY
1769	020674	021100				65\$			
1770	020676	012704	003034			MOV	#L.CS, R4		:GET POINTER TO L REGS
1771	020702	012714	000106			MOV	#SEEK, (R4)		:SET FOR SEEK
1772	020706	053714	003032			BIS	RLDRV, (R4)		:INSERT DRIVE NUMBER
1773	020712	042724	002000			BIC	#BIT10, (R4)+		:CLEAR FOR DRIVE 4 - 7 SPEC'D
1774	020716	005024				CLR	(R4)+		:CLEAR BUS ADDRESS
1775	020720	013714	003106			MOV	DESDIF, (R4)		:LOAD DIFFERENCE
1776	020724	012703	000007			MOV	#7, R3		:SET COUNT FOR SHIFT TO ALIGN
1777	020730	006314			3\$:	ASL	(R4)		:ALIGN DIFFERENCE IN DA
1778	020732	005303				DEC	R3		
1779	020734	001375				BNE	3\$		
1780	020736	005737	003110			TST	DESSGN		:TEST IF SIGN SET
1781	020742	001402				BEQ	5\$:NO - SKIP
1782	020744	052714	000004			BIS	#DIRBIT, (R4)		:INSERT SIGN
1783	020750	005737	003112		5\$:	TST	DESHD		:TEST IF HEAD 0
1784	020754	001402				BEQ	7\$:YES - SKIP


```

1785 020756 052714 000020      BIS      #HDSSEL,(R4)      ;INSERT HEAD BIT
1786 020762 052724 000001      7$:    BIS      #MBSSET0,(R4)+ ;INSERT MARKER BIT
1787 020766 005037 003006      CLR      DONE           ;CLEAR INTERRUPT FLAG
1788 020772 012701 000012      MOV      #10,R1         ;SET WAIT COUNT FOR 800US
1789 020776 014462 000004      MOV      -(R4),RLDA(R2) ;LOAD RL REGISTERS
1790 021002 014462 000002      MOV      -(R4),RLBA(R2)
1791 021006 014462 000000      MOV      -(R4),RLCS(R2)
1792 021012 005737 003006      10$:    TST      DONE           ;CHECK IF INTERRUPTED
1793 021016 001030                BNE      65$           ;YES - SKIP
1794 021020 005301                DEC      R1            ;DEC WAIT COUNT
1795 021022 001415                BEQ      13$           ;IF 0 - SKIP
1796 021024                WAITUS  #1
   (3) 021024 012727 000001      MOV      ###1,(PC)+
   (3) 021030 000000                .WORD  0
   (3) 021032 013727 002116      MOV      L$DLY,(PC)+
   (3) 021036 000000                .WORD  0
   (3) 021040 005367 177772      DEC      -6(PC)
   (3) 021044 001375                BNE      -4
   (3) 021046 005367 177756      DEC      -22(PC)
   (3) 021052 001367                BNE      -20
1797 021054 000756                BR       10$           ;GO CHECK DONE
1798 021056 004737 016324      13$:    JSR      PC,WAITIN    ;GO WAIT FOR TIMEOUT
1799 021062 012603                MOV      (SP)+,R3      ;GET RESULT MESSAGE POINTER
1800 021064                ERRHRD 10011,ERR1
   (4) 021064 104456                TRAP    C$ERRHRD
   (5) 021066 023433                .WORD  10011
   (5) 021070 000000                .WORD  0
   (5) 021072 012344                .WORD  ERR1
1801 021074 005037 003016      CLR      ERRSWI        ;CLEAR FOR ERROR RETURN
1802 021100                14$:
1803 021100 162737 000002 003002 65$:    SUB      #2,SSINDEX    ;REMOVE ENTRY FROM SUBROUT STACK
1804 021106 012604                MOV      (SP)+,R4      ;RESTORE REGS
1805 021110 012600                MOV      (SP)+,R0
1806 021112 012603                MOV      (SP)+,R3
1807 021114 005737 003016      TST      ERRSWI        ;TEST IF ERROR RETURN
1808 021120 001403                BEQ      99$           ;YES - SKIP
1809 021122 063716 003016      ADD      ERRSWI,(SP)   ;ADD IN ERROR RETURN
1810 021126 000207                RTS      PC
1811 021130 017616 000000      99$:    MOV      @ (SP),(SP)   ;SET ERROR RETURN ADDRESS
1812 021134 000207                RTS      PC
1814
1890
1891      ; DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
1892      ; 500MS FOR READY TO SET.
1892 021136 010346                RDYCHK: MOV      R3,-(SP)    ;STORE REGS
1893 021140 013703 003002      MOV      SSINDEX,R3   ;GET SUBROUTINE INDEX
1894 021144 005723                TST      (R3)+        ;BUMP IT FOR NEXT ENTRY
1895 021146 016663 000002 002404      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
1896 021154 162763 000004 002404      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
1897 021162 010337 003002      MOV      R3,SSINDEX   ;STORE IT BACK
1898 021166 010046                MOV      R0,-(SP)
1899 021170 010146                MOV      R1,-(SP)
1900 021172 010446                MOV      R4,-(SP)
1901 021174 012737 000002 003016      MOV      #2,ERRSWI    ;SET FOR NO ERROR RETURN
1902 021202 012701 011610      MOV      #5000,R1     ;SET WAIT COUNT
1903 021206 004737 016564      1$:    JSR      PC,GSTAT     ;GET DRIVE STATUS
1904 021212 021426                4$

```

```

1905 021214 032737 000001 003044 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1906 021222 001103 BNE 5$ ;YES - EXIT
1907 021224 WAITUS #1
(3) 021224 012727 000001 MOV ##1,(PC)+
(3) 021230 000000 .WORD 0
(3) 021232 013727 002116 MOV L$DLY,(PC)+
(3) 021236 000000 .WORD 0
(3) 021240 005367 177772 DEC -6(PC)
(3) 021244 001375 BNE -.4
(3) 021246 005367 177756 DEC -22(PC)
(3) 021252 001367 BNE -.20
1908 021254 005301 DEC R1 ;DEC WAIT COUNT
1909 021256 001353 BNE 1$ ;LOOP IF NOT 0
1910 021260 012703 010263 MOV #MDRDY,R3 ;SET RESULT MESSAGE POINTER
1911 021264 012704 011263 MOV #C500MS,R4 ;SET CONDITION MESSAGE POINTER
1912 021270 ERRHRD 10010.,,ERR5
(4) 021270 104456 TRAP C$ERRD
(5) 021272 023432 .WORD 10010
(5) 021274 000000 .WORD 0
(5) 021276 012576 .WORD ERR5
1913 021300 012701 000030 MOV #24.,R1 ;INITIALIZE WAIT COUNT
1914 021304 004737 016564 2$: JSR PC,G$STAT ;GET DRIVE STATUS
1915 021310 021426 4$
1916 021312 032737 000001 003044 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
1917 021320 001030 BNE 3$ ;YES - SKIP
1918 021322 WAITMS #1 ;WAIT FOR 100MS
(3) 021340 012727 000372 MOV ##250.,(PC)+
(3) 021344 000000 .WORD 0
(3) 021346 013727 002116 MOV L$DLY,(PC)+
(3) 021352 000000 .WORD 0
(3) 021354 005367 177772 DEC -6(PC)
(3) 021360 001375 BNE -.4
(3) 021362 005367 177756 DEC -22(PC)
(3) 021366 001367 BNE -.20
1919 021376 005301 DEC R1 ;DEC WAIT COUNTER
1920 021400 001341 BNE 2$ ;LOOP UNTIL TIME DONE
1921 021402 032737 100000 003044 3$: BIT #ANYERR,T.CS ;TEST IF ANYERR SET
1922 021410 001406 BEQ 4$ ;NO - SKIP
1923 021412 ERRHRD 10011.,,ERR6 ;REPORT ALL ERRORS
(4) 021412 104456 TRAP C$ERRD
(5) 021414 023433 .WORD 10011
(5) 021416 000000 .WORD 0
(5) 021420 012646 .WORD ERR6
1924 021422 005337 003156 DEC ERRCNT ;REDUCE ERROR COUNT FOR DUAL ERRORS
1925 021426 005037 003016 4$: CLR ERRSWI ;CLEAR FOR ERROR RETURN
1926 021432 162737 000002 003002 5$: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
1927 021440 012604 MOV (SP)+,R4 ;RESTORE REGS
1928 021442 012601 MOV (SP)+,R1
1929 021444 012600 MOV (SP)+,R0
1930 021446 012603 MOV (SP)+,R3
1931 021450 005737 003016 TST ERRSWI ;TEST IF ERROR RETURN
1932 021454 001403 BEQ 99$ ;YES - SKIP
1933 021456 063716 003016 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
1934 021462 000207 RTS PC
1935 021464 017616 000000 99$: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
1936 021470 000207 RTS PC
  
```



```

1937
1938
1939
1940 021472 005037 003112
1941 021476 032737 010000 014206
1942 021504 001403
1943 021506 013737 014214 003112
1944 021514 000207
1945
1946
1947
1948 021516 032737 010000 014206
1949 021524 001011
1950 021526 005737 003112
1951 021532 001006
1952 021534 012737 000001 003112
1953 021542 062716 000002
1954 021546 000207
1955 021550 017616 000000
1956 021554 000207
1957
1958
1959 021556 010046
1960 021560 013700 003100
1961 021564 013737 003102 003100
1962 021572 010037 003102
1963 021576 012600
1964 021600 000207
1965
1980
1981 021602 012737 000001 003126
1982 021610 000402
1983 021612 005037 003126
1984 021616 010346
1985 021620 013703 003002
1986 021624 005723
1987 021626 016663 000002 002404
1988 021634 162763 000004 002404
1989 021642 010337 003002
1990 021646 010046
1991 021650 010146
1992 021652 010446
1993 021654 012737 000002 003016
1994 021662 005737 003126
1995 021666 001007
1996 021670 012703 003044
1997 021674 012701 000004
1998 021700 014346
1999 021702 005301
2000 021704 001375
2001 021706 004737 021136
2002 021712 022200
2003 021714 005037 003006
2004 021720 012701 003034
2005 021724 013711 003032
2006 021730 042711 002000

: CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
: SELECTED BY SOFTWARE PARAMETER.
CHOSHD: CLR DESHD ;CLEAR TO HEAD 0
BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
BEQ 1$ ;NO - SKIP
MOV HEADW,DESHD ;INSERT SPECIFIED HEAD
1$: RTS PC

: SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
: UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
SWAPHD: BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
BNE 2$ ;YES - TAKE ABORT EXIT
TST DESHD ;TEST IF HEAD ONE USED
BNE 2$ ;YES - TAKE ABORT EXIT
MOV #1,DESHD ;ELSE SET FOR HEAD ONE
ADD #2,(SP) ;BUMP PAST ABORT RETURN
RTS PC ;RETURN
2$: MOV @ (SP),(SP) ;GET ABORT DESTINATION
3$: RTS PC

: SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
ONSWAP: MOV R0,-(SP) ;STORE R0
MOV OLDCYL,R0 ;MOVE OLD TO R0
MOV NEWCYL,OLDCYL ;MOVE NEW TO OLD
MOV R0,NEWCYL ;PUT OLD IN NEW
MOV (SP)+,R0 ;RESTORE R0
RTS PC

: READ HEADERS ROUTINE.
XRDHDC: MOV #1,TEMP4 ;SET FLAG TO BYPASS REG STORAGE
BR XRDHDG ;GO DO IT
XRDHD: CLR TEMP4 ;SET FLAG TO SAVE T. AND L. REGS
XRDHDG: MOV R3,-(SP) ;STORE REGISTERS
MOV SSINDX,R3 ;GET SUBROUTINE INDEX
TST (R3)+ ;BUMP IT FOR NEXT ENTRY
MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
MOV R3,SSINDX ;STORE IT BACK
MOV R0,-(SP)
MOV R1,-(SP)
MOV R4,-(SP)
MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
TST TEMP4 ;TEST IF REGISTERS TO BE SAVED
BNE 2$ ;NO - SKIP
MOV #L.MP+2,R3 ;SET POINTER FOR REGS
MOV #4,R1 ;SET COUNT
1$: MOV -(R3),-(SP) ;SAVE REGISTER
DEC R1 ;DEC COUNT
BNE 1$ ;LOOP UNTIL ALL ARE SAVED
2$: JSR PC,RDYCHK ;CHECK DRIVE READY
65$
CLR DONE ;CLEAR INTERRUPT FLAG
MOV #L.CS,R1 ;GET ADDRESS OF LOAD REGS
MOV RLDRV,(R1) ;LOAD DRIVE NUMBER
BIC #BIT10,(R1) ;CLEAR FOR DRIVE 4 - 7 SPEC'D

```

```

2007 021734 052721 000110      BIS      #RDHEAD,(R1)+  ;INSERT COMMAND
2008 021740 005021              CLR      (R1)+        ;CLEAR BA
2009 021742 005021              CLR      (R1)+        ;CLEAR DA
2010 021744 014162 000004      MOV      -(R1),RLDA(R2) ;LOAD RL11 REGS
2011 021750 014162 000002      MOV      -(R1),RLBA(R2)
2012 021754 014162 000000      MOV      -(R1),RLCSR(R2)
2013 021760          3$:      WAITUS   #10.        ;WAIT 1 MS FOR INTERRUPT
(3) 021760 012727 000012      MOV      ###10.,(PC)+
(3) 021764 000000              .WORD    0
(3) 021766 013727 002116      MOV      L$DLY,(PC)+
(3) 021772 000000              .WORD    0
(3) 021774 005367 177772      DEC      -6(PC)
(3) 022000 001375              BNE      -4
(3) 022002 005367 177756      DEC      -22(PC)
(3) 022006 001367              BNE      -20
2014 022010 005737 003006      TST     DONE          ;TEST IF INTERRUPT FLAG SET
2015 022014 001460              BEQ     14$           ;NO - SKIP
2016 022016 032737 000001 003044 5$:      BIT     #DRDYMSK,T.CS ;TEST IF DRIVE READY
2017 022024 001035              BNE     10$           ;YES - SKIP
2018 022026 012703 010263      MOV     #MDRDY,R3     ;SET NO READY MESSAGE
2019 022032 012704 011302      MOV     #CAFDT,R4     ;CONDITION OF AFTER DATA XFER
2020 022036          (4) 022036 104456      ERRHRD  10017...ERR5
(5) 022040 023441              TRAP   C$ERHRD
(5) 022042 000000              .WORD  10017
(5) 022044 012576              .WORD  0
(5) 022044 012576              .WORD  ERR5
2021 022046 012701 000030      MOV     #24.,R1       ;INITIALIZE WAIT COUNT
2022 022052 004737 016564          4$:      JSR     PC,GSTAT      ;GET STATUS
2023 022056 022174              60$
2024 022060 032737 000001 003044      BIT     #DRDYMSK,T.CS ;TEST IF DRIVE HAS COME READY
2025 022066 001403              BEQ     11$           ;NO - SKIP
2026 022070 005037 003016      CLR     ERRSWI        ;CLEAR ERROR SWITCH
2027 022074 000411              BR      10$           ;SKIP
2028 022076 005301          11$:      DEC     R1            ;DEC WAIT COUNT
2029 022100 001364              BNE     4$            ;LOOP UNTIL TIME DONE
2030 022102 012704 011313      MOV     #C5SEC,R4     ;SET CONDITION AFTER 5 SECONDS
2031 022106          (4) 022106 104456      ERRHRD  10014...ERR5
(5) 022110 023436              TRAP   C$ERHRD
(5) 022112 000000              .WORD  10014
(5) 022114 012576              .WORD  0
(5) 022114 012576              .WORD  ERR5
2032 022116 000426              BR      60$           ;EXIT
2033 022120 005737 003044          10$:      TST     T.CS          ;CHECK FOR ANY ERRORS
2034 022124 100005              BPL     12$           ;NO - SKIP
2035 022126          (4) 022126 104456      ERRHRD  10016...ERR6
(5) 022130 023440              TRAP   C$ERHRD
(5) 022132 000000              .WORD  10016
(5) 022134 012646              .WORD  0
(5) 022134 012646              .WORD  ERR6
2036 022136 000416              BR      60$
2037 022140 012701 003054          12$:      MOV     #HDWRD2,R1    ;GET POINTER
2038 022144 016221 000006      MOV     RLMP(R2),(R1)+ ;STORE LAST TWO HEADER WORDS
2039 022150 016221 000006      MOV     RLMP(R2),(R1)+
2040 022154 000411              BR      65$
2041 022156 004737 016324          14$:      JSR     PC,WAITIN    ;WAIT FOR INTERRUPT
2042 022162 012603              MOV     (SP)+,R3     ;GET RESULTS
  
```



```

2043 022164          ERRHRD 10015,,ERR1 ;REPORT
(4) 022164 104456   TRAP   C$ERHRD
(5) 022166 023437   .WORD 10015
(5) 022170 000000   .WORD 0
(5) 022172 012344   .WORD ERR1
2044 022174 005037 003016 60$: CLR   ERRSWI ;CLEAR FOR ERROR RETURN
2045 022200 005737 003126 65$: TST  TEMP4 ;TEST IF REGISTERS WERE SAVED
2046 022204 001007   BNE   22$ ;NO - SKIP
2047 022206 012703 003034   MOV  #L.CS,R3 ;SET POINTER TO RESTORE REGS
2048 022212 012701 000004   MOV  #4,R1 ;SET COUNT
2049 022216 012623   20$: MOV  (SP)+,(R3)+ ;RESTORE REGISTER
2050 022220 005301   DEC  R1 ;DEC COUNT
2051 022222 001375   BNE  20$ ;LOOP UNTIL ALL ARE RESTORED
2052 022224 162737 000002 003002 22$: SUB  #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
2053 022232 012604   MOV  (SP)+,R4 ;RESTORE REGS
2054 022234 012601   MOV  (SP)+,R1
2055 022236 012600   MOV  (SP)+,R0
2056 022240 012603   MOV  (SP)+,R3
2057 022242 005737 003016   TST  ERRSWI ;TEST IF ERROR RETURN
2058 022246 001403   BEQ  99$ ;YES - SKIP
2059 022250 063716 003016   ADD  ERRSWI,(SP) ;ADD IN ERROR RETURN
2060 022254 000207   RTS  PC
2061 022256 017616 000000 99$: MOV  @ (SP),(SP) ;SET ERROR RETURN ADDRESS
2062 022262 000207   RTS  PC
2063
2139 ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
2140 022264 013705 003052 POSHW1: MOV  HDWRD1,R5 ;START FOR POSITION HD BIT IN WD 1
2141 022270 000402   BR   POSHDO ;SKIP
2142 022272 013705 003052 POSHSB: MOV  T.MP,R5 ;START FOR POSITION HD BIT IN MP
2143 022276 010146   POSHDO: MOV  R1,-(SP) ;STORE R1
2144 022300 042705 177677   BIC  #^CHSSTAT,R5 ;CLEAR ALL BUT HEAD SEL BIT
2145 022304 012701 000006   MOV  #6,R1 ;SET SHIFT COUNT
2146 022310 006205   1$: ASR  R5 ;SHIFT FOR RIGHT JUSTIFY
2147 022312 005301   DEC  R1
2148 022314 001375   BNE  1$
2149 022316 012601   MOV  (SP)+,R1 ;RESTORE R1
2150 022320 000207   RTS  PC ;RETURN
2151
2152 ; WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
2153 ; FROM THE CALLING ROUTINE IN R1.
2154 022322 010346   RDYWAIT: MOV  R3,-(SP) ;STORE R3
2155 022324 013703 003002   MOV  SSINDX,R3 ;GET SUBROUTINE INDEX
2156 022330 005723   TST  (R3)+ ;BUMP IT FOR NEXT ENTRY
2157 022332 016663 000002 002404   MOV  2(SP),SUBSTK(R3) ;INSERT THIS CALL
2158 022340 162763 000004 002404   SUB  #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
2159 022346 010337 003002   MOV  R3,SSINDX ;STORE IT BACK
2160 022352 010046   MOV  R0,-(SP)
2161 022354 010146   MOV  R1,-(SP)
2162 022356 010446   MOV  R4,-(SP)
2163 022360 012737 000002 003016   MOV  #2,ERRSWI ;SET FOR NO ERROR RETURN
2164 022366 004737 016564 5$: JSR  PC,GSTAT ;GET DRIVE STATUS
2165 022372 022622   10$
2166 022374 032737 000001 003044   BIT  #DRDYMSK,T.CS ;CHECK IF READY
2167 022402 001111   BNE  9$ ;YES - SKIP
2168 022404 005301   DEC  R1 ;DEC WAIT COUNT
2169 022406 001415   BEQ  7$ ;SKIP IF 0

```

2170	022410				WAITUS	#1		
(3)	022410	012727	000001		MOV	###1,(PC)+		
(3)	022414	000000			.WORD	0		
(3)	022416	013727	002116		MOV	L\$DLY,(PC)+		
(3)	022422	000000			.WORD	0		
(3)	022424	005367	177772		DEC	-6(PC)		
(3)	022430	001375			BNE	-.4		
(3)	022432	005367	177756		DEC	-22(PC)		
(3)	022436	001367			BNE	.-20		
2171	022440	000752			BR	5\$		
2172	022442	012703	010263	7\$:	MOV	#MDRDY,R3	:SET NAME MESSAGE PTR	
2173	022446				ERRHRD	10020,,,ERR3	:REPORT READY ERROR	
(4)	022446	104456			TRAP	C\$ERHRD		
(5)	022450	023444			.WORD	10020		
(5)	022452	000000			.WORD	0		
(5)	022454	012460			.WORD	ERR3		
2174	022456	012701	000030		MOV	#24,,R1	:INITIALIZE WAIT COUNT	
2175	022462	004737	016564	6\$:	JSR	PC,GSTAT	:GET DRIVE STATUS	
2176	022466	022622			10\$			
2177	022470	032737	000001	003044	BIT	#DRDYMSK,T.CS	:TEST IF DRIVE READY	
2178	022476	001037			BNE	8\$:YES - SKIP	
2179	022500				WAITMS	#1	:WAIT 100 MS	
(3)	022516	012727	000372		MOV	##250,,(PC)+		
(3)	022522	000000			.WORD	0		
(3)	022524	013727	002116		MOV	L\$DLY,(PC)+		
(3)	022530	000000			.WORD	0		
(3)	022532	005367	177772		DEC	-6(PC)		
(3)	022536	001375			BNE	-.4		
(3)	022540	005367	177756		DEC	-22(PC)		
(3)	022544	001367			BNE	.-20		
2180	022554	005301			DEC	R1	:DEC WAIT COUNT	
2181	022556	001341			BNE	6\$:LOOP UNTIL TIME DONE	
2182	022560	012704	011313		MOV	#C5SEC,R4	:SET CONDITION AFTER 5 SECDS	
2183	022564				ERRHRD	10021,,,ERR5		
(4)	022564	104456			TRAP	C\$ERHRD		
(5)	022566	023445			.WORD	10021		
(5)	022570	000000			.WORD	0		
(5)	022572	012576			.WORD	ERR5		
2184	022574	000410			BR	11\$:EXIT	
2185	022576	032737	100000	003044	8\$:	BIT	#ANYERR,T.CS	:TEST IF ANY ERROR SET
2186	022604	001406			BEQ	10\$:NO - SKIP	
2187	022606				ERRHRD	10022,,,ERR6	:REPORT ALL ERRORS	
(4)	022606	104456			TRAP	C\$ERHRD		
(5)	022610	023446			.WORD	10022		
(5)	022612	000000			.WORD	0		
(5)	022614	012646			.WORD	ERR6		
2188	022616	005337	003156	11\$:	DEC	ERRCNT	:DECREMENT FOR DOUBLE ERROR REPORT	
2189	022622	005037	003016	10\$:	CLR	ERRSWI	:CLEAR FOR ERROR ERROR RETURN	
2190	022626	162737	000002	003002	9\$:	SUB	#2,SSINDX	:REMOVE ENTRY FROM SUBROUT STACK
2191	022634	012604			MOV	(SP)+,R4	:RESTORE REGISTERS	
2192	022636	012601			MOV	(SP)+,R1		
2193	022640	012600			MOV	(SP)+,R0		
2194	022642	012603			MOV	(SP)+,R3	:RESTORE R3	
2195	022644	005737	003016		TST	ERRSWI	:TEST IF ERROR RETURN	
2196	022650	001403			BEQ	99\$:YES - SKIP	
2197	022652	063716	003016		ADD	ERRSWI,(SP)	:ADD IN ERROR RETURN	


```

2198 022656 000207
2199 022660 017616 000000
2200 022664 000207
2201
2202
2203
2204
2205 022666 010346
2206 022670 013703 003002
2207 022674 005723
2208 022676 016663 000002 002404
2209 022704 162763 000004 002404
2210 022712 010337 003002
2211 022716 010046
2212 022720 010546
2213 022722 004737 021612
2214 022726 022756
2215 022730 013703 003052
2216 022734 012705 000007
2217 022740 006203
2218 022742 005305
2219 022744 001375
2220 022746 042703 177000
2221 022752 010337 003104
2222 022756 162737 000002 003002
2223 022764 012605
2224 022766 012600
2225 022770 012603
2226 022772 005737 003016
2227 022776 001403
2228 023000 063716 003016
2229 023004 000207
2230 023006 017616 000000
2231 023012 000207
2232
2261
2262
2263 023014 010346
2264 023016 013703 003002
2265 023022 005723
2266 023024 016663 000002 002404
2267 023032 162763 000004 002404
2268 023040 010337 003002
2269 023044 010046
2270 023046 010146
2271 023050 010446
2272 023052 012737 000002 003016
2273 023060 012701 000050
2274 023064 052737 100000 003004
2275 023072 012703 003762
2276 023076 013704 003026
2277 023102 062704 000006
2278 023106 012737 000010 003034
2279 023114 053737 003032 003034
2280 023122 042737 002000 003034
2281 023130 005037 003036

          99$: RTS PC
          MOV @ (SP), (SP) ;SET ERROR RETURN ADDRESS
          RTS PC

          ; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
          ; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
          ; NUMBER IN CURCYL.
GETPOS: MOV R3, -(SP) ;STORE REGISTERS
        MOV SSINDX, R3 ;GET SUBROUTINE INDEX
        TST (R3)+ ;BUMP IT FOR NEXT ENTRY
        MOV 2(SP), SUBSTK(R3) ;INSERT THIS CALL
        SUB #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
        MOV R3, SSINDX ;STORE IT BACK
        MOV R0, -(SP)
        MOV R5, -(SP)
        JSR PC, XRDHD ;DO READ HEADER
        65$
        MOV HDWRD1, R3 ;GET HEADER WORD
        MOV #7, R5 ;SET SHIFT COUNT
        4$: ASR R3 ;SHIFT TO RIGHT JUSTIFY
        DEC R5
        BNE 4$
        BIC #177000, R3
        MOV R3, CURCYL ;STORE AS CURRENT CYLINDER
        65$: SUB #2, SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
        MOV (SP)+, R5 ;RESTORE REGISTERS
        MOV (SP)+, R0
        MOV (SP)+, R3
        TST ERRSWI ;TEST IF ERROR RETURN
        BEQ 99$ ;YES - SKIP
        ADD ERRSWI, (SP) ;ADD IN ERROR RETURN
        RTS PC
        99$: MOV @ (SP), (SP) ;SET ERROR RETURN ADDRESS
        RTS PC

          ; READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
          ; IN Ibuff.
RDALHD: MOV R3, -(SP) ;STORE REGISTERS
        MOV SSINDX, R3 ;GET SUBROUTINE INDEX
        TST (R3)+ ;BUMP IT FOR NEXT ENTRY
        MOV 2(SP), SUBSTK(R3) ;INSERT THIS CALL
        SUB #4, SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
        MOV R3, SSINDX ;STORE IT BACK
        MOV R0, -(SP)
        MOV R1, -(SP)
        MOV R4, -(SP)
        MOV #2, ERRSWI ;SET FOR NO ERROR RETURN
        MOV #40, R1 ;SET HEADER COUNT
        BIS #HDR40, OPFLAG ;SET 40 HDR OP FLAG
        MOV #IBUFF, R3 ;SET POINTER TO STORE HDRS
        MOV RLBAS, R4 ;GET BASE ADDRESS
        ADD #RLMP, R4 ;MAKE IT POINT TO MP REG
        MOV #10, L.CS ;LOAD FOR READ HEADER, NO INTERRUPT
        BIS RLDRV, L.CS ;INSERT DRIVE NUMBER
        BIC #BIT10, L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
        CLR L.BA ;CLEAR BA
  
```

```

2282 023134 005037 003040 CLR L.DA :CLEAR DA
2283 023140 005737 003112 TST DESHD :TEST IF HEAD 0
2284 023144 001403 BEQ 3$ :YES - SKIP
2285 023146 052737 000020 003040 BIS #HDSSEL,L.DA :ELSE INSERT HEAD 0
2286 023154 013762 003040 000004 3$: MOV L.DA,RLDA(R2) :LOAD RLDA REG
2287 023162 013762 003036 000002 MOV L.BA,RLBA(R2) :LOAD RLBA
2288 023170 032762 000200 000000 BIT #CRDYMSK,RLCS(R2) :TEST IF CONTROLLER READY
2289 023176 001003 BNE 6$ :YES - SKIP
2290 023200 004737 021136 JSR PC,RDYCHK :ELSE CHECK READY
2291 023204 023322 65$
2292 023206 013762 003034 000000 6$: MOV L.CS,RLCS(R2) :LOAD RLCS REG
2293 023214 012700 077777 MOV #77777,R0 :SET COUNT FOR WAIT
2294 023220 032762 000200 000000 7$: BIT #CRDYMSK,RLCS(R2) :CHECK THAT OPERATION COMPLETED
2295 023226 001016 BNE 8$ :YES - SKIP
2296 023230 005300 DEC R0 :DEC COUNT
2297 023232 001372 BNE 7$ :SKIP IF NOT YET 0
2298 023234 004737 016272 JSR PC,READRL :ELSE GET ALL REGISTERS
2299 023240 004737 016324 JSR PC,WAITIN :ELSE WAIT FOR TIMEOUT
2300 023244 012603 MOV (SP)+,R3 :GET RESULT MESSAGE POINTER
2301 023246 ERRHRD 10025...ERR1
(4) 023246 104456 TRAP C$ERRHD
(5) 023250 023451 .WORD 10025
(5) 023252 000000 .WORD 0
(5) 023254 012344 .WORD ERR1
2302 023256 005037 003016 CLR ERRSWI :CLEAR FOR ERROR RETURN
2303 023262 000417 BR 65$
2304 023264 005737 003044 8$: TST T.CS :TEST FOR ANY ERRORS
2305 023270 100007 BPL 12$ :NO - SKIP
2306 023272 ERRHRD 10026...ERR6
(4) 023272 104456 TRAP C$ERRHD
(5) 023274 023452 .WORD 10026
(5) 023276 000000 .WORD 0
(5) 023300 012646 .WORD ERR6
2307 023302 005037 003016 CLR ERRSWI :CLEAR FOR ERROR RETURN
2308 023306 000405 BR 65$
2309 023310 011423 12$: MOV (R4),(R3)+ :STORE HEADER WORDS
2310 023312 011423 MOV (R4),(R3)+
2311 023314 011423 MOV (R4),(R3)+
2312 023316 005301 DEC R1 :DEC HEADER COUNT
2313 023320 001332 BNE 6$
2314 023322 162737 000002 003002 65$: SUB #2,SSINDX :REMOVE ENTRY FROM SUBROUT STACK
2315 023330 012604 MOV (SP)+,R4 :RESTORE REGISTERS
2316 023332 012601 MOV (SP)+,R1
2317 023334 012600 MOV (SP)+,R0
2318 023336 012603 MOV (SP)+,R3
2319 023340 005737 003016 TST ERRSWI :TEST IF ERROR RETURN
2320 023344 001403 BEQ 99$ :YES - SKIP
2321 023346 063716 003016 ADD ERRSWI,(SP) :ADD IN ERROR RETURN
2322 023352 000207 RTS PC
2323 023354 017616 000000 99$: MOV @ (SP),(SP) :SET ERROR RETURN ADDRESS
2324 023360 000207 RTS PC
2325
2326
2554 : REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
2555 : OPERATION BEING PERFORMED PORTION OF ALL
2556 : ERROR MESSAGES.

```


2557	023362	010446		RPTOP:	MOV	R4,-(SP)	
2558	023364	005737	003002		TST	SSINDEX	;TEST SUBROUTINE INDEX 0
2559	023370	001433			BEQ	1\$;SKIP IF 0
2560	023372	012704	000002		MOV	#2,R4	;SET INDEXER TO FIRST ENTRY
2561	023376				PRINTB	#FMT9,#SEQMES	;PRINT 'SUBROUTINE CALL SEQ'
(8)	023376	012746	010132		MOV	#SEQMES,-(SP)	
(7)	023402	012746	011632		MOV	#FMT9,-(SP)	
(6)	023406	012746	000002		MOV	#2,-(SP)	
(3)	023412	010600			MOV	SP,R0	
(4)	023414	104414			TRAP	C\$PNTB	
(4)	023416	062706	000006		ADD	#6,SP	
2562	023422			3\$:	PRINTB	#FMT16,SUBSTK(R4)	;PRINT CALLING LOCATION
(8)	023422	016446	002404		MOV	SUBSTK(R4),-(SP)	
(7)	023426	012746	012005		MOV	#FMT16,-(SP)	
(6)	023432	012746	000002		MOV	#2,-(SP)	
(3)	023436	010600			MOV	SP,R0	
(4)	023440	104414			TRAP	C\$PNTB	
(4)	023442	062706	000006		ADD	#6,SP	
2563	023446	062704	000002		ADD	#2,R4	;BUMP INDEX
2564	023452	020437	003002		CMP	R4,SSINDEX	;CHECK IF ALL PRINTED
2565	023456	003761			BLE	3\$;LOOP IF NOT ALL PRINTED YET
2566	023460			1\$:	PRINTB	#FMT4,ERHEAD,#TSTLAB	;PRINT ERROR HEADER
(9)	023460	012746	006363		MOV	#TSTLAB,-(SP)	
(8)	023464	013746	003012		MOV	ERHEAD,-(SP)	
(7)	023470	012746	011435		MOV	#FMT4,-(SP)	
(6)	023474	012746	000003		MOV	#3,-(SP)	
(3)	023500	010600			MOV	SP,R0	
(4)	023502	104414			TRAP	C\$PNTB	
(4)	023504	062706	000010		ADD	#10,SP	
2567	023510	042737	030000	003004	BIC	#SEEKOP!RORWOP,OPFLAG	;CLEAR SK & RD OR WRT FLAG
2568	023516	013701	003034		MOV	L.CS,R1	;GET COMMAND EXECUTED
2569	023522	042701	177741		BIC	#177741,R1	;STRIP ALL BUT FUNCTION CODE
2570	023526	022701	000006		CMP	#6,R1	;TEST IF SEEK OPERATION
2571	023532	001003			BNE	2\$;NO - SKIP
2572	023534	052737	010000	003004	BIS	#SEEKOP,OPFLAG	;ELSE SET SEEK FLAG
2573	023542	022701	000012		2\$:	CMP	#12,R1
2574	023546	001003			BNE	20\$;NO - SKIP
2575	023550	052737	020000	003004	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG
2576	023556	022701	000014		20\$:	CMP	#14,R1
2577	023562	001003			BNE	22\$;NO - SKIP
2578	023564	052737	020000	003004	BIS	#RORWOP,OPFLAG	;SET RD OR WRT FLAG
2579	023572			22\$:	PRINTB	#FMT1,#MOPER,OPMSGS(R1)	;PRINT OPERATION
(9)	023572	016146	002224		MOV	OPMSGS(R1),-(SP)	
(8)	023576	012746	005412		MOV	#MOPER,-(SP)	
(7)	023602	012746	011413		MOV	#FMT1,-(SP)	
(6)	023606	012746	000003		MOV	#3,-(SP)	
(3)	023612	010600			MOV	SP,R0	
(4)	023614	104414			TRAP	C\$PNTB	
(4)	023616	062706	000010		ADD	#10,SP	
2580	023622	020127	000004		CMP	R1,#4	;CHECK IF GET STATUS
2581	023626	001007			BNE	4\$;NO - SKIP
2582	023630	032737	000010	003040	BIT	#DRSET,L.DA	;TEST IF RESET INCLUDED
2583	023636	001403			BEQ	4\$;NO - SKIP
2584	023640	012701	000016		MOV	#16,R1	;SET TO PRINT WITH RESET
2585	023644	000436			BR	9\$	
2586	023646	032737	007777	003004	4\$:	BIT	#COMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION


```

2587 023654 001424          BEQ      8$          ;NO - SKIP
2588 023656 013704 003004  MOV      OPFLAG,R4    ;SET UP TO DETERMINE WHICH ONE
2589 023662 012701 000020  MOV      #20,R1       ;PRESET THE POINTER
2590 023666 032704 000001  5$:     BIT      #BIT00,R4 ;CHECK THE BIT
2591 023672 001003          BNE      6$          ;IF SET - SKIP
2592 023674 005721          TST      (R1)+        ;BUMP POINTER
2593 023676 006204          ASR      R4
2594 023700 000772          BR       5$
2595 023702          6$:     PRINTB   #FMT2,OPMSGs(R1)
(8) 023702 016146 002224  MOV      OPMSGs(R1),-(SP)
(7) 023706 012746 011427  MOV      #FMT2,-(SP)
(6) 023712 012746 000002  MOV      #2,-(SP)
(3) 023716 010600          MOV      SP,R0
(4) 023720 104414          TRAP    C$PNTB
(4) 023722 062706 000006  ADD      #6,SP
2596 023726 032737 100000 003004  8$:     BIT      #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
2597 023734 001415          BEQ      10$         ;NO - SKIP
2598 023736 012701 000050  MOV      #50,R1       ;ELSE PRINT IT
2599 023742          9$:     PRINTB   #FMT2,OPMSGs(R1)
(8) 023742 016146 002224  MOV      OPMSGs(R1),-(SP)
(7) 023746 012746 011427  MOV      #FMT2,-(SP)
(6) 023752 012746 000002  MOV      #2,-(SP)
(3) 023756 010600          MOV      SP,R0
(4) 023760 104414          TRAP    C$PNTB
(4) 023762 062706 000006  ADD      #6,SP
2600 023766 000434          BR       15$         ;SKIP
2601 023770 032737 010000 003004  10$:    BIT      #SEEKOP,OPFLAG ;TEST IF SEEK
2602 023776 001430          BEQ      15$         ;NO - SKIP
2603 024000          PRINTB   #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
(15) 024000 013746 003112  MOV      DESHD,-(SP)
(14) 024004 012746 010073  MOV      #HDWD,-(SP)
(13) 024010 013746 003110  MOV      DESSGN,-(SP)
(12) 024014 012746 010066  MOV      #SGNWD,-(SP)
(11) 024020 013746 003106  MOV      DESDIF,-(SP)
(10) 024024 012746 010060  MOV      #DIFWD,-(SP)
(9) 024030 013746 003100  MOV      OLDCYL,-(SP)
(8) 024034 012746 010111  MOV      #FRMWD,-(SP)
(7) 024040 012746 011653  MOV      #FMT13,-(SP)
(6) 024044 012746 000011  MOV      #11,-(SP)
(3) 024050 010600          MOV      SP,R0
(4) 024052 104414          TRAP    C$PNTB
(4) 024054 062706 000024  ADD      #24,SP
2604 024060 032737 020000 003004  15$:    BIT      #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
2605 024066 001424          BEQ      17$         ;NO - SKIP
2606 024070          PRINTB   #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
(13) 024070 013746 003114  MOV      DESSEC,-(SP)
(12) 024074 012746 010077  MOV      #SECWD,-(SP)
(11) 024100 013746 003112  MOV      DESHD,-(SP)
(10) 024104 012746 010073  MOV      #HDWD,-(SP)
(9) 024110 013746 003104  MOV      CURCYL,-(SP)
(8) 024114 012746 010104  MOV      #CYLWD,-(SP)
(7) 024120 012746 012202  MOV      #FMT22,-(SP)
(6) 024124 012746 000007  MOV      #7,-(SP)
(3) 024130 010600          MOV      SP,R0
(4) 024132 104414          TRAP    C$PNTB
(4) 024134 062706 000020  ADD      #20,SP
  
```



```

2607 024140 004737 024612      17$: JSR    PC,CLRPARM      ;CLEAR PARAM TABLE
2608 024144 012604              MOV    (SP)+,R4          ;RESTORE R4
2609 024146 000207              RTS     PC
2610
2611      ; REPORT REASON ROUTINE
2612      ; PRINTS REASON PORTION FOR ALL ERROR REPORTS.
2613 024150 010146      RPTRES: MOV    R1,-(SP)      ;STORE R1
2614 024152 010346      MOV    R3,-(SP)      ;STORE R3
2615 024154 010446      MOV    R4,-(SP)      ;STORE R4
2616 024156 012701 003062      MOV    #RESPARM,R1    ;GET START OF PARAM
2617 024162 012103      MOV    (R1)+,R3       ;GET NUMBER OF PARAM
2618 024164              PRINTB #FMT1.1,#MRSLT,(R1) ;PRINT NAME
(9) 024164 011146      MOV    (R1),-(SP)
(8) 024166 012746 005421      MOV    #MRSLT,-(SP)
(7) 024172 012746 011420      MOV    #FMT1.1,-(SP)
(6) 024176 012746 000003      MOV    #3,-(SP)
(3) 024202 010600      MOV    SP,R0
(4) 024204 104414      TRAP  C$PNTB
(4) 024206 062706 000010      ADD    #10,SP
2619 024212 021127 010736      CMP    (R1),#MNRST    ;TEST IF MESSAGE IS NO DRV STATUS
2620 024216 001453      BEQ    6$             ;YES - SKIP REST OF REPORT
2621 024220 012704 011637      MOV    #FMT11,R4      ;PRESET FOR FORMAT 11
2622 024224 022127 010731      CMP    (R1)+,#MCYLOC ;CHECK IF REPORTING CYLINDER LOC
2623 024230 001002      BNE    3$             ;NO - SKIP
2624 024232 012704 011645      MOV    #FMT12,R4      ;ELSE CHANGE TO FORMAT 12
2625 024236 005303      3$: DEC    R3           ;DEC PARAM COUNT
2626 024240 001442      BEQ    6$             ;IF 0 - EXIT
2627 024242              PRINTB R4,#RESE3,(R1)+ ;REPORT IS VALUE
(9) 024242 012146      MOV    (R1)+,-(SP)
(8) 024244 012746 011157      MOV    #RESE3,-(SP)
(7) 024250 010446      MOV    R4,-(SP)
(6) 024252 012746 000003      MOV    #3,-(SP)
(3) 024256 010600      MOV    SP,R0
(4) 024260 104414      TRAP  C$PNTB
(4) 024262 062706 000010      ADD    #10,SP
2628 024266              PRINTB R4,#RESE4,(R1)+ ;REPORT SB VALUE
(9) 024266 012146      MOV    (R1)+,-(SP)
(8) 024270 012746 011163      MOV    #RESE4,-(SP)
(7) 024274 010446      MOV    R4,-(SP)
(6) 024276 012746 000003      MOV    #3,-(SP)
(3) 024302 010600      MOV    SP,R0
(4) 024304 104414      TRAP  C$PNTB
(4) 024306 062706 000010      ADD    #10,SP
2629 024312 162703 000002      SUB    #2,R3          ;DEC PARAM COUNT
2630 024316 001413      BEQ    6$             ;IF 0 - EXIT
2631 024320              PRINTB #FMT1,#RESE5,(R1)+ ;REPORT CONDITION
(9) 024320 012146      MOV    (R1)+,-(SP)
(8) 024322 012746 011170      MOV    #RESE5,-(SP)
(7) 024326 012746 011413      MOV    #FMT1,-(SP)
(6) 024332 012746 000003      MOV    #3,-(SP)
(3) 024336 010600      MOV    SP,R0
(4) 024340 104414      TRAP  C$PNTB
(4) 024342 062706 000010      ADD    #10,SP
2632 024346 012604      6$: MOV    (SP)+,R4          ;RESTORE REGS
2633 024350 012603      MOV    (SP)+,R3
2634 024352 012601      MOV    (SP)+,R1

```



```

2635 024354 000207          RTS      PC          ;RETURN
2636
2637          :          REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
2638          :          AND ALL REGISTER CONTENTS.
2639 024356 005046          RPTREM: PRINTB  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 024356 153716 003033          CLR      -(SP)
(11) 024360 012746 006051          BISB    RLDRV+1,(SP)
(10) 024364 012746 003026          MOV     #DRVNAM,-(SP)
(9) 024370 013746 006040          MOV     RLBAS,-(SP)
(8) 024374 012746 011446          MOV     #BASADD,-(SP)
(7) 024400 012746 000005          MOV     #FMT5,-(SP)
(6) 024404 012746          MOV     #5,-(SP)
(3) 024410 010600          MOV     SP,R0
(4) 024412 104414          TRAP   C$PNTB
(4) 024414 062706 000014          ADD     #14,SP
2640          :          REPORT RL11 REGISTERS
2641 024420          PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
(13) 024420 012746 010073          MOV     #HDWD,-(SP)
(12) 024424 012746 010104          MOV     #CYLWD,-(SP)
(11) 024430 012746 006135          MOV     #MPNAM,-(SP)
(10) 024434 012746 006123          MOV     #BANAM,-(SP)
(9) 024440 012746 006130          MOV     #DANAM,-(SP)
(8) 024444 012746 006116          MOV     #CSNAM,-(SP)
(7) 024450 012746 011466          MOV     #FMT6,-(SP)
(6) 024454 012746 000007          MOV     #7,-(SP)
(3) 024460 010600          MOV     SP,R0
(4) 024462 104414          TRAP   C$PNTB
(4) 024464 062706 000020          ADD     #20,SP
2642 024470          PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
(12) 024470 013746 003042          MOV     L.MP,-(SP)
(11) 024474 013746 003036          MOV     L.BA,-(SP)
(10) 024500 013746 003040          MOV     L.DA,-(SP)
(9) 024504 013746 003034          MOV     L.CS,-(SP)
(8) 024510 012746 006142          MOV     #LAB1,-(SP)
(7) 024514 012746 011600          MOV     #FMT8,-(SP)
(6) 024520 012746 000006          MOV     #6,-(SP)
(3) 024524 010600          MOV     SP,R0
(4) 024526 104414          TRAP   C$PNTB
(4) 024530 062706 000016          ADD     #16,SP
2643 024534          PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
(14) 024534 013746 003112          MOV     DESHD,-(SP)
(13) 024540 013746 003104          MOV     CURCYL,-(SP)
(12) 024544 013746 003052          MOV     T.MP,-(SP)
(11) 024550 013746 003046          MOV     T.BA,-(SP)
(10) 024554 013746 003050          MOV     T.DA,-(SP)
(9) 024560 013746 003044          MOV     T.CS,-(SP)
(8) 024564 012746 006155          MOV     #LAB2,-(SP)
(7) 024570 012746 011530          MOV     #FMT7,-(SP)
(6) 024574 012746 000010          MOV     #10,-(SP)
(3) 024600 010600          MOV     SP,R0
(4) 024602 104414          TRAP   C$PNTB
(4) 024604 062706 000022          ADD     #22,SP
2644 024610 000207          RTS      PC
2645
2646          :          CLEAR PARAMETER BLOCK FOR REPORTING
2647 024612 010546          CLRPARAM: MOV     R5,-(SP)          ;STORE R5
  
```



```
2648 024614 012701 003062      MOV    #RESPARM,R1      ;GET ADDRESS OF BLOCK
2649 024620 012705 000005      MOV    #5,R5           ;SET COUNT
2650 024624 005021              2$: CLR    (R1)+        ;CLEAR WORD
2651 024626 005305              DEC    R5              ;DEC COUNT
2652 024630 001375              BNE    2$              ;LOOP UNTIL 0
2653 024632 012701 003062      MOV    #RESPARM,R1      ;RESET POINTER
2654 024636 012605              MOV    (SP)+,R5        ;RESTORE R5
2655 024640 000207              RTS    PC
```

2656
2657 024642

ENDMOD

.TITLE CZRLIB0 RL01/02 DRIVE TEST 1

;DISK STATE FUNCTIONS

;BITS 0-2 OF THE MULTIPURPOSE REGISTER DURING GET STATUS COMMAND DEFINE THE
;STATE OF THE DRIVE

```
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
: STATE 0          LOAD CARTRIDGE  
: STATE 1          SPIN UP  
: STATE 2          BRUSH CYCLE  
: STATE 3          LOAD HEADS  
: STATE 4          SEEK  
: STATE 5          LOCKON  
: STATE 6          UNLOAD HEADS  
: STATE 7          SPIN DOWN
```

```

2679 024642          BGNMOD  HRDWTST
2680                .SBTTL  *TEST 1          BASIC INTERFACE (PART 1)
2681
2682 024642          BGNTST                ;TEST01
(3) 024642
2683 024642 005737 003356          TST      PASNUM          ;CHECK IF FIRST PASS          T1::
2684 024646 001124          BNE      65$            ;EXIT IF NO
2685 024650 005737 014206          TST      MISWIW         ;CHECK IF MANUAL INTERVENTION
2686 024654 100121          BPL      65$            ;NO - EXIT TEST
2687 024656 012737 006371 003012  MOV      #MISTST,ERHEAD ;LOAD ERR HEADER
2688 024664          2$: PRINTF    #FMTCP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 024664 005046          CLR      -(SP)
(13) 024666 153716 003033          BISB    RLDRV+1,(SP)
(12) 024672 012746 006051          MOV     #DRVNAM,-(SP)
(11) 024676 013746 003026          MOV     RLBAS,-(SP)
(10) 024702 012746 006040          MOV     #BASADD,-(SP)
(9) 024706 012746 010014          MOV     #OPR1A,-(SP)
(8) 024712 012746 007372          MOV     #OPR1,-(SP)
(7) 024716 012746 011321          MOV     #FMTOP1,-(SP)
(6) 024722 012746 000007          MOV     #7,-(SP)
(3) 024726 010600          MOV     SP,R0
(4) 024730 104417          TRAP   C$PNTF
(4) 024732 062706 000020          ADD     #20,SP
2689 024736 005037 004362          CLR     OBUFF           ;CLEAR FOR RESPONSE
2690 024742          GMANIL  OPR002,OBUFF,1,NO
(3) 024742 104443          TRAP   C$GMAN
(3) 024744 000404          BR      10000$
(4) 024746 004362          .WORD  OBUFF
(5) 024750 000120          .WORD  T$CODE
(5) 024752 007322          .WORD  OPR002
(5) 024754 000001          .WORD  1
(3) 024756          10000$:
2691 024756 005737 004362          TST     OBUFF           ;TEST RESPONSE YES
2692 024762 001740          BEQ     2$             ;YES - SKIP
2693 024764 004737 016516          1$: JSR    PC,TSTINT    ;INITIALIZE TEST
2694 024770 004737 016550          JSR    PC,GSTATC      ;GO GET STATUS (NO RESET)
2695 024774 025120          65$:
2696 024776 032737 000040 003052  BIT     #COSTAT,T.MP    ;CHECK IF COVER OPEN SET
2697 025004 001006          BNE     7$             ;YES - SKIP
2698 025006 012703 010441          MOV     #MCOSTA,R3     ;SET NAME POINTER
2699 025012          ERRHRD  101,,,ERR3
(4) 025012 104456          TRAP   C$ERRHD
(5) 025014 000145          .WORD  101
(5) 025016 000000          .WORD  0
(5) 025020 012460          .WORD  ERR3
2700 025022 032737 000010 003052  7$: BIT     #BHSTAT,T.MP    ;TEST IF BRUSHES HOME
2701 025030 001006          BNE     9$             ;YES - SKIP
2702 025032 012703 010454          MOV     #MBHSTA,R3     ;SET POINTER FOR BRUSH HOME ERROR
2703 025036          ERRHRD  102,,,ERR3
(4) 025036 104456          TRAP   C$ERRHD
(5) 025040 000146          .WORD  102
(5) 025042 000000          .WORD  0
(5) 025044 012460          .WORD  ERR3
2704 025046 032737 020000 003052  9$: BIT     #WLSTAT,T.MP    ;TEST IF WRITE LOCK SET
2705 025054 001006          BNE     11$            ;YES - SKIP
2706 025056 012703 010467          MOV     #MWLSTA,R3     ;SET NAME POINTER
  
```


2707	025062			ERRHRD	103,,,ERR3	
(4)	025062	104456		TRAP	C\$ERHRD	
(5)	025064	000147		.WORD	103	
(5)	025066	000000		.WORD	0	
(5)	025070	012460		.WORD	ERR3	
2708	025072	005737	003060	TST	T,STAT	;TEST IF STATE ZERO
2709	025076	001405		BEQ	15\$;YES - SKIP
2710	025100	005003		CLR	R3	;SET STATE EXPECTED
2711	025102			ERRHRD	104,,,ERR7	
(4)	025102	104456		TRAP	C\$ERHRD	
(5)	025104	000150		.WORD	104	
(5)	025106	000000		.WORD	0	
(5)	025110	013546		.WORD	ERR7	
2712	025112	004737	016534	JSR	PC,GSTATR	;DO DRIVE RESET
2713	025116	025120		65\$		
2714	025120			65\$:		
2715	025120			ENDTST		
(3)	025120			L10024:		
(3)	025120	104401		TRAP	C\$ETST	
2716						

```
2718 .SBTTL *TEST 2 BASIC INTERFACE (PART 2)
2719
2720 025122 BGNTST ;TEST 2
(3) 025122 T2::
2721 025122 005737 003356 TST PASNUM ;TEST IF PASS 0
2722 025126 001077 BNE 65$ ;NO - SKIP
2723 025130 005737 014206 TST MISWIW ;TEST IF MANUAL INTERVENTION
2724 025134 100074 BPL 65$ ;NO - SKIP
2725 025136 012737 006371 003012 MOV #MISTST,ERHEAD ;SET ERROR HEADER
2726 025144 2$: PRINTF #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CLOSE
(13) 025144 005046 CLR -(SP)
(13) 025146 153716 003033 BISB RLDRV+1,(SP)
(12) 025152 012746 006051 MOV #DRVNAM,-(SP)
(11) 025156 013746 003026 MOV RLBAS,-(SP)
(10) 025162 012746 006040 MOV #BASADD,-(SP)
(9) 025166 012746 010014 MOV #OPR1A,-(SP)
(8) 025172 012746 007450 MOV #OPR2,-(SP)
(7) 025176 012746 011321 MOV #FMTOP1,-(SP)
(6) 025202 012746 000007 MOV #7,-(SP)
(3) 025206 010600 MOV SP,R0
(4) 025210 104417 TRAP C$PNTF
(4) 025212 062706 000020 ADD #20,SP
2727 ;COVER AND RESET WRITE LOCK
2728 025216 005037 004362 CLR OBUFF ;CLEAR FOR RESPONSE
2729 025222 GMANIL OPR002,OBUFF,1,NO
(3) 025222 104443 TRAP C$GMAN
(3) 025224 000404 BR 10000$
(4) 025226 004362 .WORD OBUFF
(5) 025230 000120 .WORD T$CODE
(5) 025232 007322 .WORD OPR002
(5) 025234 000001 .WORD 1
(3) 025236 10000$:
2730 025236 005737 004362 TST OBUFF ;TEST IF RESPONSE YES
2731 025242 001740 BEQ 2$ ;NO - SKIP
2732 025244 004737 016516 1$: JSR PC,TSTINT ;INITIALIZE TEST
2733 025250 004737 016534 JSR PC,GSTATR ;GET STATUS WITH RESET
2734 025254 025326 65$
2735 025256 032737 000040 003052 BIT #COSTAT,T.MP ;TEST IF COVER OPEN RESET
2736 025264 001406 BEQ 9$ ;YES - SKIP
2737 025266 012703 010441 MOV #MCOSTA,R3 ;SET NAME MESSAGE POINTER
2738 025272 ERRHRD 201,,,ERR2
(4) 025272 104456 TRAP C$ERRHD
(5) 025274 000311 .WORD 201
(5) 025276 000000 .WORD 0
(5) 025300 012412 .WORD ERR2
2739 025302 032737 020000 003052 9$: BIT #WLSTAT,T.MP ;TEST IF WRITE LOCK RESET
2740 025310 001406 BEQ 65$ ;YES - SKIP
2741 025312 012703 010467 MOV #MWLSTA,R3 ;SET NAME MESSAGE POINTER
2742 025316 ERRHRD 202,,,ERR2
(4) 025316 104456 TRAP C$ERRHD
(5) 025320 000312 .WORD 202
(5) 025322 000000 .WORD 0
(5) 025324 012412 .WORD ERR2
2743 025326 65$:
2744 025326 ENDTST
(3) 025326 L10025:
```


CZRLIB0 RL01/02 DRIVE TEST 1 MACY11 30A(1052) 17-DEC-79 13:08 ^{G 8} PAGE 2-42
CZRLIB.MAC 12-DEC-79 14:02 *TEST 2 BASIC INTERFACE (PART 2)

SEQ 0097

(3) 025326 104401
2745

TRAP C\$ETST

```

2747          .SBTTL *TEST 3          HEAD LOADING
2748 025330    BGNTST                ;TEST03
(3) 025330
2749 025330    005737 003356          TST PASNUM          ;TEST IF PASS 0
2750 025334    001003                BNE 4$              ;NO SKIP
2751 025336    005737 014206          TST MISWIW         ;TEST IF MANUAL INTERVENTION
2752 025342    100402                BMI 7$              ;YES - SKIP
2753 025344    4$: EXIT TST
(3) 025344    104432                TRAP C$EXIT
(3) 025346    002144                .WORD L10026-
2754 025350    004737 016516          7$: JSR PC,TSTINT   ;INITIALIZE TEST
2755 025354    004737 016534          JSR PC,GSTATR     ;GET STATUS
2756 025360    027512
2757 025362    005737 003060          TST T.STAT        ;TEST IF STATE 0
2758 025366    001440                BEQ 2$              ;YES - SKIP
2759 025370    1$: PRINTF #FMTOP1,#OPR5,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST DRIVE BE
(13) 025370    005046                CLR -(SP)
(13) 025372    153716 003033          BISB RLDRV+1,(SP)
(12) 025376    012746 006051          MOV #DRVNAM,-(SP)
(11) 025402    013746 003026          MOV RLBAS,-(SP)
(10) 025406    012746 006040          MOV #BASADD,-(SP)
(9) 025412    012746 010014          MOV #OPR1A,-(SP)
(8) 025416    012746 007516          MOV #OPR5,-(SP)
(7) 025422    012746 011321          MOV #FMTOP1,-(SP)
(6) 025426    012746 000007          MOV #7,-(SP)
(3) 025432    010600                MOV SP,R0
(4) 025434    104417                TRAP C$PNTF
(4) 025436    062706 000020          ADD #20,SP
2760 025442    005037 004362          CLR OBUF           ;CLEAR FOR RESPONSE
2761 025446    GMANIL OPR002,OBUF,1,NO
(3) 025446    104443                TRAP C$GMAN
(3) 025450    000404                BR 10000$
(4) 025452    004362                .WORD OBUF
(5) 025454    000120                .WORD T$CODE
(5) 025456    007322                .WORD OPR002
(5) 025460    000001                .WORD 1
(3) 025462    10000$:
2762 025462    005737 004362          TST OBUF           ;TEST IF RESPONSE YES
2763 025466    001740                BEQ 1$              ;NO - SKIP
2764 025470    2$: PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 025470    005046                CLR -(SP)
(13) 025472    153716 003033          BISB RLDRV+1,(SP)
(12) 025476    012746 006051          MOV #DRVNAM,-(SP)
(11) 025502    013746 003026          MOV RLBAS,-(SP)
(10) 025506    012746 006040          MOV #BASADD,-(SP)
(9) 025512    012746 010014          MOV #OPR1A,-(SP)
(8) 025516    012746 007502          MOV #OPR3,-(SP)
(7) 025522    012746 011321          MOV #FMTOP1,-(SP)
(6) 025526    012746 000007          MOV #7,-(SP)
(3) 025532    010600                MOV SP,R0
(4) 025534    104417                TRAP C$PNTF
(4) 025536    062706 000020          ADD #20,SP
2765 025542    012737 000004 003004    MOV #CYLUP,OPFLAG ;SET CYCLE UP FLAG
2766 025550    012703 000001          MOV #1,R3          ;SET EXPECTED STATE VALUE
2767 025554    012737 006414 003012    MOV #NSTACHG,ERHEAD ;SET ERROR HEADER
2768 025562    012701 000454          MOV #300.,R1      ;SET WAIT COUNT FOR 30 SECONDS
  
```



```

2769 025566 004737 016550      3$:   JSR    PC,GSTATC      ;GET STATUS
2770 025572 027512              T365$
2771 025574 005737 003060      TST    T,STAT          ;TEST IF STATE IS STILL 0
2772 025600 001046              BNE    10$             ;NO - SKIP
2773 025602 005301              DEC    R1              ;DEC WAIT COUNT
2774 025604 001427              BEQ    6$              ;EXIT IF WAIT DONE
2775 025606              WAITMS #1
   (3) 025624 012727 000372      MOV    ##250.,(PC)+
   (3) 025630 000000              .WORD 0
   (3) 025632 013727 002116      MOV    L$DLY,(PC)+
   (3) 025636 000000              .WORD 0
   (3) 025640 005367 177772      DEC    -6(PC)
   (3) 025644 001375              BNE    -4
   (3) 025646 005367 177756      DEC    -22(PC)
   (3) 025652 001367              BNE    -20
2776 025662 000741              BR     3$
2777 025664 005037 004362      6$:   CLR    OBUF          ;CLEAR FOR RESPONSE
2778 025670              GMANIL OPR003,OBUF,1,NO
   (3) 025670 104443              TRAP  C$GMAN
   (3) 025672 000404              BR     10001$
   (4) 025674 004362              .WORD OBUF
   (5) 025676 000120              .WORD T$CODE
   (5) 025700 007347              .WORD OPR003
   (5) 025702 000001              .WORD 1
   (3) 025704              10001$:
2779 025704 005737 004362      TST    OBUF          ;TEST IF RESPONSE YES
2780 025710 001005              BNE    11$           ;YES - REPORT
2781 025712 000137 025370      JMP    1$
2782 025716 020337 003060      10$:  CMP    R3,T,STAT    ;CHECK IF NOW STATE 1
2783 025722 001406              BEQ    5$            ;YES - SKIP
2784 025724              11$:  ERRHRD 301,,,ERR7
   (4) 025724 104456              TRAP  C$ERRHRD
   (5) 025726 000455              .WORD 301
   (5) 025730 000000              .WORD 0
   (5) 025732 013546              .WORD ERR7
2785 025734              EXIT  TST
   (3) 025734 104432              TRAP  C$EXIT
   (3) 025736 001554              .WORD L10026-
2786              ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
2787 025740 005737 003144      5$:   TST    CLKFLG      ;P-CLOCK?
2788 025744 001002              BNE    13$          ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
2789 025746 000137 027416      JMP    100$         ;ELSE, REPORT THAT TEST CANNOT BE PERFORMED
2790 025752 012701 000454      13$:  MOV    #300.,R1    ;INITIALIZE WAIT COUNT FOR 30 SECONDS
2791 025756 012703 000002      MOV    #2,R3        ;SET EXPECTED STATE VALUE
2792 025762 004737 016550      14$:  JSR    PC,GSTATC   ;GET STATUS
2793 025766 027512              T365$
2794 025770 020337 003060      CMP    R3,T,STAT    ;CHECK IF STATE 2
2795 025774 001466              BEQ    20$          ;YES - SKIP
2796 025776 101006              BHI    17$          ;CHECK IF NO CHANGE - YES - SKIP
2797 026000              ERRHRD 302,,,ERR7
   (4) 026000 104456              TRAP  C$ERRHRD
   (5) 026002 000456              .WORD 302
   (5) 026004 000000              .WORD 0
   (5) 026006 013546              .WORD ERR7
2798 026010              EXIT  TST
   (3) 026010 104432              TRAP  C$EXIT
  
```


2826	026326				26\$:	ERRHRD	306...ERR4	
(4)	026326	104456				TRAP	C\$ERHRD	
(5)	026330	000462				.WORD	306	
(5)	026332	000000				.WORD	0	
(5)	026334	012526				.WORD	ERR4	
2827	026336					EXIT	TST	
(3)	026336	104432				TRAP	C\$EXIT	
(3)	026340	001152				.WORD	L10026-	
2828	026342	012701	000454		27\$:	MOV	#300,R1	:INITIALIZE WAIT COUNT FOR 30 SECONDS
2829	026346	004737	016550		28\$:	JSR	PC,GSTATC	:GET STATUS
2830	026352	027512				T365\$		
2831	026354	032737	000010	003052		BIT	#BHSTAT,T.MP	:TEST IF BRUSH HOME SET AGAIN
2832	026362	001042				BNE	32\$:YES - SKIP
2833	026364	005301				DEC	R1	:ELSE DEC WAIT COUNT
2834	026366	001432				BEQ	30\$:SKIP IF 0
2835	026370					TIMDLY	#1000.	
(8)	026370	012746	000340			MOV	#340,-(SP)	
(7)	026374	012746	016116			MOV	#CLKINT,-(SP)	
(6)	026400	012746	000104			MOV	#104,-(SP)	
(5)	026404	012746	000003			MOV	#3,-(SP)	
(4)	026410	104437				TRAP	C\$SVEC	
(3)	026412	062706	000010			ADD	#10,SP	
2836	026452	000735				BR	28\$	
2837	026454				30\$:	ERRHRD	307...ERR5	
(4)	026454	104456				TRAP	C\$ERHRD	
(5)	026456	000463				.WORD	307	
(5)	026460	000000				.WORD	0	
(5)	026462	012576				.WORD	ERR5	
2838	026464					EXIT	TST	
(3)	026464	104432				TRAP	C\$EXIT	
(3)	026466	001024				.WORD	L10026-	
2839	026470	012737	006414	003012	32\$:	MOV	#NSTACHG,ERHEAD	:SET ERROR HEADER
2840	026476	012703	000003			MOV	#3,R3	:SET EXPECTED STATE VALUE
2841	026502	004737	016550			JSR	PC,GSTATC	:GET STATUS
2842	026506	027512				T365\$		
2843	026510	020337	003060			CMP	R3,T.STAT	:CHECK IF STATE 3
2844	026514	001406				BEQ	36\$:YES - SKIP
2845	026516					ERRHRD	308...ERR7	
(4)	026516	104456				TRAP	C\$ERHRD	
(5)	026520	000464				.WORD	308	
(5)	026522	000000				.WORD	0	
(5)	026524	013546				.WORD	ERR7	
2846	026526					EXIT	TST	
(3)	026526	104432				TRAP	C\$EXIT	
(3)	026530	000762				.WORD	L10026-	
2847	026532	012737	006371	003012	36\$:	MOV	#MISTST,ERHEAD	:SET ERROR HEADER
2848	026540	012704	011212			MOV	#STATE3,R4	:SET CONDITION MESSAGE POINTER
2849	026544	012703	010500			MOV	#MHOSTA,R3	:SET NAME MESSAGE POINTER
2850	026550	004737	016550			JSR	PC,GSTATC	:GET STATUS
2851	026554	027512				T365\$		
2852	026556	032737	000020	003052		BIT	#HOSTAT,T.MP	:TEST IF HEADS OUT SET
2853	026564	001006				BNE	38\$:YES - SKIP
2854	026566					ERRHRD	309...ERR5	
(4)	026566	104456				TRAP	C\$ERHRD	
(5)	026570	000465				.WORD	309	
(5)	026572	000000				.WORD	0	

(5)	026574	012576				.WORD	ERR5	
2855	026576					EXIT	TST	
(3)	026576	104432				TRAP	C\$EXIT	
(3)	026600	000712				.WORD	L10026-	
2856	026602	012701	000012		38\$:	MOV	#10.,R1	
2857	026606				381\$:	TIMDLY	#1	
(8)	026606	012746	000340			MOV	#340,-(SP)	
(7)	026612	012746	016116			MOV	#CLKINT,-(SP)	
(6)	026616	012746	000104			MOV	#104,-(SP)	
(5)	026622	012746	000003			MOV	#3,-(SP)	
(4)	026626	104437				TRAP	C\$SVEC	
(3)	026630	062706	000010			ADD	#10,SP	
2858	026670	012700	000001			MOV	#1,R0	
2859	026674	004737	016550			JSR	PC,GSTATC	:GET THE STATUS AFTER SHORT DELAY
2860	026700	027512				T365\$		
2861	026702	032737	001000	003052		BIT	#VCSTAT,T.MP	:TEST IF VOLUME CHECK SET
2862	026710	001012				BNE	40\$	
2863	026712	005301				DEC	R1	:DECREMENT COUNTER
2864	026714	001334				BNE	381\$:TRY FOR 'VC' AGAIN IF MORE TIME LEFT
2865	026716	012703	010430			MOV	#MVOLCK,R3	:SET NAME MESSAGE POINTER
2866	026722					ERRHRD	310.,,ERR5	
(4)	026722	104456				TRAP	C\$ERHRD	
(5)	026724	000466				.WORD	310	
(5)	026726	000000				.WORD	0	
(5)	026730	012576				.WORD	ERR5	
2867	026732					EXIT	TST	
(3)	026732	104432				TRAP	C\$EXIT	
(3)	026734	000556				.WORD	L10026-	
2868	026736	032737	040000	003044	40\$:	BIT	#DRVERR,T.CS	:TEST IF DRIVE ERROR SET
2869	026744	001010				BNE	42\$:YES - SKIP
2870	026746	012703	010405			MOV	#MDRERR,R3	:SET NAME MESSAGE POINTER
2871	026752					ERRHRD	311.,,ERR5	
(4)	026752	104456				TRAP	C\$ERHRD	
(5)	026754	000467				.WORD	311	
(5)	026756	000000				.WORD	0	
(5)	026760	012576				.WORD	ERR5	
2872	026762					EXIT	TST	
(3)	026762	104432				TRAP	C\$EXIT	
(3)	026764	000526				.WORD	L10026-	
2873	026766	012701	005670		42\$:	MOV	#3000.,R1	:SET WAIT COUNT FOR 300 MS
2874	026772	012737	006414	003012		MOV	#NSTACHG,ERHEAD	:SET ERROR HEADER
2875	027000	012703	000004			MOV	#4,R3	:SET EXPECTED STATE VALUE
2876	027004	004737	016550		43\$:	JSR	PC,GSTATC	:GET STATUS
2877	027010	027512				T365\$		
2878	027012	020337	003060			CMP	R3,T.STAT	:CHECK IF STATE 4
2879	027016	001442				BEQ	49\$:YES - SKIP
2880	027020	005301				DEC	R1	:DEC WAIT COUNT
2881	027022	001432				BEQ	47\$:SKIP IF 0
2882	027024					TIMDLY	#1	
(8)	027024	012746	000340			MOV	#340,-(SP)	
(7)	027030	012746	016116			MOV	#CLKINT,-(SP)	
(6)	027034	012746	000104			MOV	#104,-(SP)	
(5)	027040	012746	000003			MOV	#3,-(SP)	
(4)	027044	104437				TRAP	C\$SVEC	
(3)	027046	062706	000010			ADD	#10,SP	
2883	027106	000736				BR	43\$	

2884	027110			47\$:	ERRHRD	312,,ERR7		
(4)	027110	104456			TRAP	C\$ERHRD		
(5)	027112	000470			.WORD	312		
(5)	027114	000000			.WORD	0		
(5)	027116	013546			.WORD	ERR7		
2885	027120				EXIT	TST		
(3)	027120	104432			TRAP	C\$EXIT		
(3)	027122	000370			.WORD	L10026-		
2886	027124	012701	000454	49\$:	MOV	#300,,R1	:SET WAIT COUNT FOR 30 MS	
2887	027130	012703	000005		MOV	#5,R3	:SET EXPECTED STATE VALUE	
2888	027134	004737	016550	50\$:	JSR	PC,GSTATC	:GET STATUS	
2889	027140	027512			T365\$			
2890	027142	020337	003060		CMP	R3,T.STAT	:CHECK IF STATE 5	
2891	027146	001442			BEQ	55\$:YES - SKIP	
2892	027150	005301			DEC	R1	:DEC WAIT COUNT	
2893	027152	001432			BEQ	51\$:ELSE SKIP	
2894	027154				TIMDLY	#1		
(8)	027154	012746	000340		MOV	#340,-(SP)		
(7)	027160	012746	016116		MOV	#CLKINT,-(SP)		
(6)	027164	012746	000104		MOV	#104,-(SP)		
(5)	027170	012746	000003		MOV	#3,-(SP)		
(4)	027174	104437			TRAP	C\$SVEC		
(3)	027176	062706	000010		ADD	#10,SP		
2895	027236	000736			BR	50\$		
2896	027240			51\$:	ERRHRD	313,,ERR7		
(4)	027240	104456			TRAP	C\$ERHRD		
(5)	027242	000471			.WORD	313		
(5)	027244	000000			.WORD	0		
(5)	027246	013546			.WORD	ERR7		
2897	027250				EXIT	TST		
(3)	027250	104432			TRAP	C\$EXIT		
(3)	027252	000240			.WORD	L10026-		
2898	027254	012701	000120	55\$:	MOV	#80,,R1	:SET WAIT FOR 8 MS	
2899	027260	004737	016550	56\$:	JSR	PC,GSTATC	:GET STATUS	
2900	027264	027512			T365\$			
2901	027266	032737	000001	003044	BIT	#DRDYMSK,T.CS	:CHECK IF DRIVE READY	
2902	027274	001106			BNE	102\$:YES - SKIP	
2903	027276	005301			DEC	R1	:DEC COUNT	
2904	027300	001432			BEQ	60\$:SKIP IF 0	
2905	027302				TIMDLY	#1		
(8)	027302	012746	000340		MOV	#340,-(SP)		
(7)	027306	012746	016116		MOV	#CLKINT,-(SP)		
(6)	027312	012746	000104		MOV	#104,-(SP)		
(5)	027316	012746	000003		MOV	#3,-(SP)		
(4)	027322	104437			TRAP	C\$SVEC		
(3)	027324	062706	000010		ADD	#10,SP		
2906	027364	000735			BR	56\$		
2907	027366	012737	006371	003012	60\$:	MOV	#MISTST,ERHEAD	:SET ERROR HEADER
2908	027374	012704	011222		MOV	#STAT5,R4	:SET CONDITION MESSAGE POINTER	
2909	027400	012703	010263		MOV	#MDRDY,R3	:SET NAME MESSAGE POINTER	
2910	027404				ERRHRD	314,,ERR5		
(4)	027404	104456			TRAP	C\$ERHRD		
(5)	027406	000472			.WORD	314		
(5)	027410	000000			.WORD	0		
(5)	027412	012576			.WORD	ERR5		
2911	027414	000436			BR	102\$:EXIT TEST	

```

2912                                     :REPORT THAT TEST CANNOT BE PERFORMED
2913 027416 012702 006662                100$: MOV #NOTST,R2 :INITIALIZE POINTER FOR TEST MESSAGE
2914 027422 112762 000060 000004        MOVB #'0,4(R2) :INSERT TEST NUMBER INTO MSG.
2915 027430 112762 000063 000005        MOVB #'3,5(R2) :INSERT TEST NUMBER INTO MSG.
2916 027436                                     PRINTF #FMT9,#NOTST :PRINT MSG. 'TST 3 CANNOT BE PERFORMED...
(8) 027436 012746 006662                MOV #NOTST,-(SP)
(7) 027442 012746 011632                MOV #FMT9,-(SP)
(6) 027446 012746 000002                MOV #2,-(SP)
(3) 027452 010600                        MOV SP,R0
(4) 027454 104417                        TRAP C$PNTF
(4) 027456 062706 000006                ADD #6,SP

2917                                     :/NO P-CLK''
2918                                     ;MAKE DRIVE READY FOR SUBSEQUENT TESTS
2919 027462 013702 003026                MOV RLBAS,R2 :GET RL11 BASE ADDRESS
2920 027466 013705 003032                MOV RLDRV,R5 :GET DRIVE NUMBER
2921 027472 052705 000200                BIS #CRDYMSK,R5 :SET CONTROLLER READY
2922 027476 010562 000000                MOV R5,RLCS(R2) :LOAD CONTROL STATUS REGISTER
2923 027502 032762 000001 000000 101$: BIT #DRDYMSK,RLCS(R2) :IS DRIVE READY?
2924 027510 001774                                     BEQ 101$ :REMAIN IN WAIT LOOP UNTIL DRIVE IS READY
2925 027512
2926 027512
2927 027512
(3) 027512
(3) 027512 104401
2928                                     102$:
                                     T365$:
                                     ENDTST
                                     L10026:
                                     TRAP C$ETST
  
```



```
2930          .SBTTL *TEST 4          HEAD UNLOADING
2931 027514    BGNTST                ;TEST04
(3) 027514
2932 027514    005737 003356          TST PASNUM          ;TEST IF FIRST PASS
2933 027520    001003                BNE 8$              ;NO - SKIP
2934 027522    005737 014206          TST MISWIW         ;TEST IF MANUAL INTERVENTION
2935 027526    100402                BMI 10$             ;YES - SKIP
2936 027530    8$: EXIT TST
(3) 027530    104432                TRAP C$EXIT
(3) 027532    001054                .WORD L10027-
2937          ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
2938 027534    005737 003144          10$: TST CLKFLG     ;P-CLOCK?
2939 027540    001024                BNE TST4           ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
2940 027542    012702 006662          MOV #NOTST,R2     ;INITIALIZE POINTER FOR TEST MSG.
2941 027546    112762 000060 000004  MOVB #'0,4(R2)    ;INSERT TEST NUMBER INTO MSG.
2942 027554    112762 000064 000005  MOVB #'4,5(R2)    ;INSERT TEST NUMBER INTO MSG.
2943 027562    8: PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 4 CANNOT BE PERFORMED...
(8) 027562    012746 006662          MOV #NOTST,-(SP)
(7) 027566    012746 011632          MOV #FMT9,-(SP)
(6) 027572    012746 000002          MOV #2,-(SP)
(3) 027576    010600                MOV SP,R0
(4) 027600    104417                TRAP C$PNTF
(4) 027602    062706 000006          ADD #6,SP
2944
2945 027606    000750                BR 8$              ;/NO P-CLK''
2946
2947 027610    BGNSUB
(3) 027610
(3) 027610    104402                T4.1:
2948 027612    012737 006414 003012  TST4: TRAP C$BSUB
2949 027620    004737 016516                MOV #NSTACHG,ERHEAD ;SET ERROR HEADER
2950 027624    004737 016534                JSR PC,TSTINT       ;INITIALIZE TEST
2951 027630    030476                JSR PC,GSTATR       ;GET STATUS
2952 027632    032737 000001 003044  T465$
2953 027640    001040                BIT #DRDYMSK,T.CS  ;CHECK IF DRIVE READY
2954 027642    13: PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 027642    005046                CLR -(SP)
(13) 027644    153716 003033          BISB RLDRV+1,(SP)
(12) 027650    012746 006051          MOV #DRVNAM,-(SP)
(11) 027654    013746 003026          MOV RLBAS,-(SP)
(10) 027660    012746 006040          MOV #BASADD,-(SP)
(9) 027664    012746 010014          MOV #OPR1A,-(SP)
(8) 027670    012746 007560          MOV #OPR6,-(SP)
(7) 027674    012746 011321          MOV #FMTOP1,-(SP)
(6) 027700    012746 000007          MOV #7,-(SP)
(3) 027704    010600                MOV SP,R0
(4) 027706    104417                TRAP C$PNTF
(4) 027710    062706 000020          ADD #20,SP
2955 027714    005037 004362          CLR OBUF          ;CLEAR FOR RESPONSE
2956 027720    GMANIL OPR002,OBUF,1,NO
(3) 027720    104443                TRAP C$GMAN
(3) 027722    000404                BR 10000$
(4) 027724    004362                .WORD OBUF
(5) 027726    000120                .WORD T$CODE
(5) 027730    007322                .WORD OPR002
(5) 027732    000001                .WORD 1
```



```
(3) 027734 10000$:
2957 027734 005737 004362 TST OBUFF ;TST RESPONSE YES
2958 027740 001740 BEQ 1$ ;NO - SKIP
2959
2960 027742 052737 000010 003004 3$: BIS #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
2961 027750 4$: PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 027750 005046 CLR -(SP)
(13) 027752 153716 003033 BISB RLDRV+1,(SP)
(12) 027756 012746 006051 MOV #DRVNAM,-(SP)
(11) 027762 013746 003026 MOV RLBAS,-(SP)
(10) 027766 012746 006040 MOV #BASADD,-(SP)
(9) 027772 012746 010014 MOV #OPR1A,-(SP)
(8) 027776 012746 007502 MOV #OPR3,-(SP)
(7) 030002 012746 011321 MOV #FMTOP1,-(SP)
(6) 030006 012746 000007 MOV #7,-(SP)
(3) 030012 010600 MOV SP,R0
(4) 030014 104417 TRAP C$PNTF
(4) 030016 062706 000020 ADD #20,SP
2962 030022 012703 000006 MOV #6,R3 ;SET EXPECTED STATE VALUE
2963 030026 012704 000144 MOV #100.,R4 ;SET SECOND LEVEL COUNT
2964 030032 012701 001274 MOV #700.,R1 ;SET WAIT COUNT FOR 30 SECONDS
2965 030036 004737 016550 5$: JSR PC,GSTATC ;GET STATUS
2966 030042 030476 TIMDLY #1 ;WAIT 100 US
2967 030044 020337 003060 CMP R3,T.STAT ;CHECK IF STATE 6
2968 030050 001465 BEQ 11$ ;YES - SKIP
2969 030052 022737 000005 003060 CMP #5,T.STAT ;TEST IF STATE 5
2970 030060 001053 BNE 9$ ;NO - REPORT WRONG STATE
2971 030062 005304 8$: DEC R4 ;DEC 2ND LEVEL COUNT
2972 030064 001004 BNE 6$ ;SKIP IF NOT 0
2973 030066 005301 DEC R1 ;ELSE DEC 1ST LEVEL COUNT
2974 030070 001434 BEQ 7$ ;IF 0 - SKIP TO QUESTION
2975 030072 012704 000144 MOV #100.,R4 ;ELSE RESET 2ND LEVEL
2976 030076 6$: TIMDLY #1 ;WAIT 100 US
(8) 030076 012746 000340 MOV #340,-(SP)
(7) 030102 012746 016116 MOV #CLKINT,-(SP)
(6) 030106 012746 000104 MOV #104,-(SP)
(5) 030112 012746 000003 MOV #3,-(SP)
(4) 030116 104437 TRAP C$SVEC
(3) 030120 062706 000010 ADD #10,SP
2977 030160 000726 BR 5$
2978 030162 005037 004362 7$: CLR OBUFF ;CLEAR FOR RESPONSE
2979 030166 GMANIL OPR003,OBUFF,1,NO
(3) 030166 104443 TRAP C$GMAN
(3) 030170 000404 BR 10001$
(4) 030172 004362 .WORD OBUFF
(5) 030174 000120 .WORD T$CODE
(5) 030176 007347 .WORD OPR003
(5) 030200 000001 .WORD 1
(3) 030202 10001$:
2980 030202 005737 004362 TST OBUFF ;TEST IF RESPONSE YES
2981 030206 001660 BEQ 4$ ;NO - SKIP
2982 030210 9$: ERRHRD 401.,ERR7 ;ELSE REPORT STATE CHANGE WRONG
(4) 030210 104456 TRAP C$ERRHD
(5) 030212 000621 .WORD 401
(5) 030214 000000 .WORD 0
(5) 030216 013546 .WORD ERR7
```



```
2983 030220          EXIT SUB
(3) 030220 104432    TRAP C$EXIT
(3) 030222 000262    .WORD L10030-.
2984 030224 012703 000007    11$: MOV #7,R3 ;SET EXPECTED STATE VALUE
2985 030230 012701 005670    MOV #3000.,R1 ;SET COUNT FOR 300MS
2986 030234 004737 016550    12$: JSR PC,GSTATC ;GET STATUS
2987 030240 030476    T465$
2988 030242 020337 003060    CMP R3,T.STAT ;CHECK IF STATE 7
2989 030246 001442    BEQ 18$ ;YES - SKIP
2990 030250 005301    DEC R1 ;DEC WAIT COUNT
2991 030252 001432    BEQ 16$ ;SKIP IF 0
2992 030254          TIMDLY #1
(8) 030254 012746 000340    MOV #340,-(SP)
(7) 030260 012746 016116    MOV #CLKINT,-(SP)
(6) 030264 012746 000104    MOV #104,-(SP)
(5) 030270 012746 000003    MOV #3,-(SP)
(4) 030274 104437    TRAP C$SVEC
(3) 030276 062706 000010    ADD #10,SP
2993 030336 000736          BR 12$
2994 030340          16$: ERRHRD 402.,,ERR7 ;REPORT WRONG STATE CHANGE
(4) 030340 104456    TRAP C$ERHRD
(5) 030342 000622    .WORD 402
(5) 030344 000000    .WORD 0
(5) 030346 013546    .WORD ERR7
2995 030350          EXIT SUB
(3) 030350 104432    TRAP C$EXIT
(3) 030352 000132    .WORD L10030-.
2996 030354 005003    18$: CLR R3 ;SET EXPECTED STATE VALUE
2997 030356 012701 001130    MOV #600.,R1 ;SET WAIT COUNT FOR 60 SECONDS
2998 030362 004737 016550    20$: JSR PC,GSTATC ;GET STATUS
2999 030366 030476    T465$
3000 030370 005737 003060    TST T.STAT ;CHECK IF STATE 0
3001 030374 001440    BEQ 24$ ;YES - SKIP
3002 030376 005301    DEC R1 ;DEC WAIT COUNT
3003 030400 001432    BEQ 22$ ;SKIP IF 0
3004 030402          TIMDLY #1000.
(8) 030402 012746 000340    MOV #340,-(SP)
(7) 030406 012746 016116    MOV #CLKINT,-(SP)
(6) 030412 012746 000104    MOV #104,-(SP)
(5) 030416 012746 000003    MOV #3,-(SP)
(4) 030422 104437    TRAP C$SVEC
(3) 030424 062706 000010    ADD #10,SP
3005 030464 000736          BR 20$
3006 030466          22$: ERRHRD 403.,,ERR7 ;REPORT WRONG STATE CHANGE
(4) 030466 104456    TRAP C$ERHRD
(5) 030470 000623    .WORD 403
(5) 030472 000000    .WORD 0
(5) 030474 013546    .WORD ERR7
3007 030476          24$:
3008 030476 012737 000002 003016 24$: T465$: MOV #2,ERRSWI ;INIT ERROR SWITCH
3009
3010 030504          ENDSUB
(3) 030504          L10030:
(3) 030504 104403
3011 030506          26$: TRAP C$ESUB
(13) 030506 005046    PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REQUEST CYCLE UP
(13) 030506          CLR -(SP)
```

(13)	030510	153716	003033	BISB	RLDRV+1,(SP)	
(12)	030514	012746	006051	MOV	#DRVNAM,-(SP)	
(11)	030520	013746	003026	MOV	RLBAS,-(SP)	
(10)	030524	012746	006040	MOV	#BASADD,-(SP)	
(9)	030530	012746	010014	MOV	#OPR1A,-(SP)	
(8)	030534	012746	007560	MOV	#OPR6,-(SP)	
(7)	030540	012746	011321	MOV	#FMTOP1,-(SP)	
(6)	030544	012746	000007	MOV	#7,-(SP)	
(3)	030550	010600		MOV	SP,R0	
(4)	030552	104417		TRAP	C\$PNTF	
(4)	030554	062706	000020	ADD	#20,SP	
3012	030560	005037	004362	CLR	OBUFF	;CLEAR FOR RESPONSE
3013	030564			GMANIL	OPR002,OBUFF,1,NO	
(3)	030564	104443		TRAP	C\$GMAN	
(3)	030566	000404		BR	10000\$	
(4)	030570	004362		.WORD	OBUFF	
(5)	030572	000120		.WORD	T\$CODE	
(5)	030574	007322		.WORD	OPR002	
(5)	030576	000001		.WORD	1	
(3)	030600					
3014	030600	005737	004362	10000\$: TST	OBUFF	;TEST RESPONSE YES
3015	030604	001740		BEQ	26\$;NO - SKIP
3016	030606			29\$:		
3017						
3018	030606			ENDTST		
(3)	030606			L10027:		
(3)	030606	104401		TRAP	C\$ETST	


```

3020
3021 030610 .SBTTL *TEST 5 DRIVE SELECT
(3) 030610 BGNTST ;TEST05
3022 030610 012737 000002 003016 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN T5::
3023 030616 005737 003356 TST PASNUM ;TEST IF FIRST PASS
3024 030622 001173 BNE EXT05 ;NO - SKIP
3025 030624 032737 000004 014206 BIT #DRSELT,MISWIW ;TEST IF SELECT TESTS
3026 030632 001567 BEQ EXT05 ;NO - SKIP
3027 030634 1$: PRINTF #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(13) 030634 005046 CLR -(SP)
(13) 030636 153716 003033 BISB RLDRV+1,(SP)
(12) 030642 012746 006051 MOV #DRVNAM,-(SP)
(11) 030646 013746 003026 MOV RLBAS,-(SP)
(10) 030652 012746 006040 MOV #BASADD,-(SP)
(9) 030656 012746 010014 MOV #OPR1A,-(SP)
(8) 030662 012746 007613 MOV #OPR7,-(SP)
(7) 030666 012746 011321 MOV #FMTOP1,-(SP)
(6) 030672 012746 000007 MOV #7,-(SP)
(3) 030676 010600 MOV SP,R0
(4) 030700 104417 TRAP C$PNTF
(4) 030702 062706 000020 ADD #20,SP
3028 ;REQUEST 'REMOVE ADD PLGS EXCPT ''
3029 030706 005037 004362 CLR OBUFF ;CLEAR FOR RESPONSE
3030 030712 GMANIL OPR002,OBUFF,1,NO
(3) 030712 104443 TRAP C$GMAN
(3) 030714 000404 BR 10000$
(4) 030716 004362 .WORD OBUFF
(5) 030720 000120 .WORD T$CODE
(5) 030722 007322 .WORD OPR002
(5) 030724 000001 .WORD 1
(3) 030726 10000$:
3031 030726 005737 004362 TST OBUFF ;TEST RESPONSE YES
3032 030732 001740 BEQ 1$ ;NO - SKIP
3033 030734 012737 006526 003012 3$: MOV #T05ERR,ERHEAD ;SET ERROR HEADER MESSAGE
3034 030742 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3035 030746 004737 016550 JSR PC,GSTATC ;DO SELECT AND GET STATUS
3036 030752 031134 T504$
3037 030754 013737 003032 003116 MOV RLDRV,TEMPO ;STORE ORIGINAL DRIVE NUMBER
3038 030762 013701 003032 MOV RLDRV,R1 ;PUT IT IN R1
3039 030766 012704 000004 MOV #4,R4 ;SET COUNT FOR NUMBER OF PLUGS
3040 030772 062701 000400 LPT05: ADD #400,R1 ;BUMP TO NEXT DRIVE
3041 030776 022701 002000 CMP #2000,R1 ;CHECK IF TOO LARGE
3042 031002 001001 BNE 4$ ;NO - SKIP
3043 031004 005001 CLR R1 ;ELSE CLEAR TO DRIVE 0
3044 031006 010137 003032 4$: MOV R1,RLDRV ;PUT IT BACK IN RLDRV
3045 031012 5$: PRINTF #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
(11) 031012 012746 010030 MOV #UNDTST,-(SP)
(10) 031016 012746 010020 MOV #OPR1B,-(SP)
(9) 031022 005046 CLR -(SP)
(9) 031024 153716 003033 BISB RLDRV+1,(SP)
(8) 031030 012746 007642 MOV #OPR8,-(SP)
(7) 031034 012746 011372 MOV #FMTOP3,-(SP)
(6) 031040 012746 000005 MOV #5,-(SP)
(3) 031044 010600 MOV SP,R0
(4) 031046 104417 TRAP C$PNTF
(4) 031050 062706 000014 ADD #14,SP
  
```

```

3046
3047 031054 005037 004362 CLR OBUFF ;INSERT PLUG REQUEST
3048 031060 GMANIL OPR002,OBUFF,1,NO ;CLEAR FOR RESPONSE
(3) 031060 104443 TRAP C$GMAN
(3) 031062 000404 BR 10001$
(4) 031064 004362 .WORD OBUFF
(5) 031066 000120 .WORD T$CODE
(5) 031070 007322 .WORD OPR002
(5) 031072 000001 .WORD 1
(3) 031074 10001$:
3049 031074 005737 004362 TST OBUFF ;TEST RESPONSE YES
3050 031100 001744 BEQ 5$ ;NO - SKIP
3051 031102 BGNSUB
(3) 031102 T5.1:
(3) 031102 104402 TRAP C$BSUB
3052 031104 004737 016550 JSR PC,GSTATC ;GET STATUS - REPORT ANY ERROR
3053 031110 031112 60$:
3054 031112 012737 000002 003016 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
3055
3056 031120 ENDSUB
(3) 031120 L10032:
(3) 031120 104403 TRAP C$ESUB
3057 031122 005304 DEC R4 ;DEC COUNT
3058 031124 001322 BNE LPT05 ;LOOP IF NOT ZERO
3059 031126 013737 003116 003032 MOV TEMPO,RLDRV ;ELSE RESTORE RLDRV
3060 031134 T504$:
3061 031134 4$: PRINTF #FMT4,#OPR8,#OPR9
(9) 031134 012746 007661 MOV #OPR9,-(SP)
(8) 031140 012746 007642 MOV #OPR8,-(SP)
(7) 031144 012746 011435 MOV #FMT4,-(SP)
(6) 031150 012746 000003 MOV #3,-(SP)
(3) 031154 010600 MOV SP,R0
(4) 031156 104417 TRAP C$PNTF
(4) 031160 062706 000010 ADD #10,SP
3062 031164 005037 004362 CLR OBUFF ;CLEAR FOR RESPONSE
3063 031170 GMANIL OPR002,OBUFF,1,NO
(3) 031170 104443 TRAP C$GMAN
(3) 031172 000404 BR 10000$
(4) 031174 004362 .WORD OBUFF
(5) 031176 000120 .WORD T$CODE
(5) 031200 007322 .WORD OPR002
(5) 031202 000001 .WORD 1
(3) 031204 10000$:
3064 031204 005737 004362 TST OBUFF ;TEST RESPONSE YES
3065 031210 001751 BEQ 4$ ;NO - SKIP
3066 031212 EXT05:
3067 031212 ENDTST
(3) 031212 L10031:
(3) 031212 104401 TRAP C$ETST
3068
  
```



```
3070 .SBTTL *TEST 6 DRIVE SELECT TEST
3071 BGNSTST ;TEST06
(3) 031214
3072 031214 005737 003356 TST PASNUM ;CHECK IF FIRST PASS
3073 031220 001004 BNE 1$ ;NO - SKIP
3074 031222 032737 000004 014206 BIT #DRSELT,MISWIW ;CHECK IF TEST DRIVE SELECT
3075 031230 001002 BNE 4$ ;YES - SKIP
3076 031232 1$: EXIT TST
(3) 031232 104432 TRAP C$EXIT
(3) 031234 001134 .WORD L10033-
3077 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3078 031236 005737 003144 4$: TST CLKFLG ;P-CLOCK?
3079 031242 001023 BNE 6$ ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3080 031244 012702 006662 MOV #NOTST,R2 ;INITIALIZE POINTER FOR TEST MSG.
3081 031250 112762 000060 000004 MOV #0,4(R2) ;INSERT TEST NUMBER INTO MSG.
3082 031256 112762 000066 000005 MOV #6,5(R2) ;INSERT TEST NUMBER INTO MSG.
3083 031264 PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 6 CANNOT BE PERFORMED...
(8) 031264 012746 006662 MOV #NOTST,-(SP)
(7) 031270 012746 011632 MOV #FMT9,-(SP)
(6) 031274 012746 000002 MOV #2,-(SP)
(3) 031300 010600 MOV SP,R0
(4) 031302 104417 TRAP C$PNTF
(4) 031304 062706 000006 ADD #6,SP
3084 ;/NO P-CLK''
3085 031310 000750 BR 1$ ;EXIT TEST
3086 031312 012737 006462 003012 6$: MOV #GSTER1,ERHEAD ;SET ERROR HEADER
3087 031320 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3088 031324 013703 003360 MOV PSETNM,R3 ;GET PARAM SET NUMBER
3089 031330 023727 002012 000001 CMP L$UNIT,#1 ;TEST IF MORE THAN 1 UNIT
3090 031336 101476 BLOS 5$ ;NO - SKIP
3091 031340 005203 2$: INC R3 ;BUMP PARAMETER SET NUMBER
3092 031342 020337 002012 CMP R3,L$UNIT ;CHECK IF PAST VALID PARAMETER TABLE
3093 031346 101401 BLOS 3$ ;NO - SKIP
3094 031350 005003 CLR R3 ;ELSE CLEAR TO POINT TO ENTRY 0
3095 031352 3$: GPHARD R3,R0
(3) 031352 010300 MOV R3,R0
(3) 031354 104442 TRAP C$GPHRD
3096 031356 BNCOMPLETE 2$ ;SKIP IF NOT AVAILABLE
(2) 031356 103370 BCC 2$
3097 031360 010004 MOV R0,R4 ;PUT POINTER INTO R4
3098 031362 021437 003026 CMP (R4),RLBAS ;CHECK IF SAME CONTROLLER
3099 031366 001364 BNE 2$ ;NO - SKIP
3100 031370 005037 003006 CLR DONE ;CLEAR DONE FLAG
3101 031374 012737 000104 003034 MOV #GTSTAT,L.CS ;LOAD GET STATUS
3102 031402 056437 000010 003034 BIS 10(R4),L.CS ;INSERT DRIVE
3103 031410 012737 000013 003040 MOV #GETSTAT!DRSET,L.DA ;SET UP TO CLEAR DRIVE
3104 031416 013762 003040 000004 MOV L.DA,RLDA(R2) ;LOAD DA REG
3105 031424 013762 003034 000000 MOV L.CS,RLCS(R2) ;LOAD CS REG
3106 031432 TIMDLY #3 ;WAIT 300 US
(8) 031432 012746 000340 MOV #340,-(SP)
(7) 031436 012746 016116 MOV #CLKINT,-(SP)
(6) 031442 012746 000104 MOV #104,-(SP)
(5) 031446 012746 000003 MOV #3,-(SP)
(4) 031452 104437 TRAP C$SVEC
(3) 031454 062706 000010 ADD #10,SP
3107 031514 005737 003006 TST DONE ;TEST IF INTERRUPT
```

```

3108 031520 001707          BEQ      2$          ;NO - SKIP
3109 031522 032737 100000 003044 BIT      #ANYERR,T.CS ;TEST IF ANY ERROR SET
3110 031530 001415          BEQ      7$          ;NO - GO TEST
3111 031532 000702          BR       2$          ;ELSE CHECK NEXT DRIVE
3112 031534          5$: PRINTF  #FMT9,#OPR10 ;REPORT CAN'T FIND 2ND DRIVE
(8) 031534 012746 007676 MOV     #OPR10,-(SP)
(7) 031540 012746 011632 MOV     #FMT9,-(SP)
(6) 031544 012746 000002 MOV     #2,-(SP)
(3) 031550 010600          MOV     SP,R0
(4) 031552 104417          TRAP   C$PNTF
(4) 031554 062706 000006 ADD     #6,SP
3113 031560 000137 032370 JMP     LCLEXT
3114 031564 016437 000010 003120 7$: MOV     10(R4),TEMP1 ;STORE NEW ADDRESS
3115          ;ASK FOR PLUG CHANGE
3116 031572 013700 003032 9$: MOV     RLDRV,R0 ;GET DRIVE UNDER TEST
3117 031576 013705 003120 MOV     TEMP1,R5 ;GET NEW ADDRESS
3118 031602 042700 002000 BIC     #2000,R0 ;CLEAR FOR ADDRESS 0 TO 3
3119 031606 042705 002000 BIC     #2000,R5
3120 031612 020527 001400 20$: CMP     R5,#1400 ;TEST IF DRIVE NUMBER 3
3121 031616 001001          BNE     21$          ;NO - SKIP
3122 031620 005005          CLR     R5          ;ELSE SET TO DRIVE NUMBER 0
3123 031622 062705 000400 21$: ADD     #400,R5 ;BUMP TO NEXT ADDRESS
3124 031626 020500          CMP     R5,R0 ;THIS EQUAL TO NEW ADDRESS?
3125 031630 001770          BEQ     20$          ;YES - SKIP
3126 031632 052705 000200 BIS     #CRDYMSK,R5 ;ELSE SET CONTROLLER READY BIT
3127 031636 010562 000000 MOV     R5,RLCS(R2) ;AND LOAD CS REG
3128 031642          PRINTF  #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
(11) 031642 005046          CLR     -(SP)
(11) 031644 153716 003121 BISB    TEMP1+1,(SP)
(10) 031650 012746 010020 MOV     #OPR1B,-(SP)
(9) 031654 005046          CLR     -(SP)
(9) 031656 153716 003033 BISB    RLDRV+1,(SP)
(8) 031662 012746 007642 MOV     #OPR8,-(SP)
(7) 031666 012746 011350 MOV     #FMTOP2,-(SP)
(6) 031672 012746 000005 MOV     #5,-(SP)
(3) 031676 010600          MOV     SP,R0
(4) 031700 104417          TRAP   C$PNTF
(4) 031702 062706 000014 ADD     #14,SP
3129 031706 005037 004362 CLR     OBUFF ;CLEAR FOR RESPONSE
3130 031712          GMANIL  OPR002,OBUFF,1,NO
(3) 031712 104443          TRAP   C$GMAN
(3) 031714 000404          BR     10000$
(4) 031716 004362          .WORD  OBUFF
(5) 031720 000120          .WORD  T$CODE
(5) 031722 007322          .WORD  OPR002
(5) 031724 000001          .WORD  1
(3) 031726          10000$:
3131 031726 005737 004362 TST     OBUFF ;TEST IF RESPONSE YES
3132 031732 001717          BEQ     9$          ;NO - SKIP
3133 031734 012704 000012 MOV     #10.,R4 ;SET COUNT
3134 031740          BGNSUB
(3) 031740          T6.1:
(3) 031740 104402          TRAP   C$BSUB
3135 031742 013737 003032 003034 8$: MOV     RLDRV,L.CS ;SET UP TO SELECT MULTIPLE DRIVES
3136 031750 013762 003034 000000 MOV     L.CS,RLCSR(R2) ;DO IT
3137 031756          TIMDLY #10.

```


(8)	031756	012746	000340		MOV	#340,-(SP)	
(7)	031762	012746	016116		MOV	#CLKINT,-(SP)	
(6)	031766	012746	000104		MOV	#104,-(SP)	
(5)	031772	012746	000003		MOV	#3,-(SP)	
(4)	031776	104437			TRAP	C\$\$VEC	
(3)	032000	062706	000010		ADD	#10,SP	
3138	032040	052737	000104	003034	BIS	#GTSTAT,L.CS	;SET GET STATUS
3139	032046	012737	000003	003040	MOV	#GETSTAT,L.DA	
3140	032054	013762	003040	000004	MOV	L.DA,RLDA(R2)	
3141	032062	005037	003006		CLR	DONE	
3142	032066	013762	003034	000000	MOV	L.CS,RLCSR(R2)	;DO GET STATUS
3143	032074				WAITUS	#1	;WAIT FOR INTERRUPT
(3)	032074	012727	000001		MOV	###1,(PC)+	
(3)	032100	000000			.WORD	0	
(3)	032102	013727	002116		MOV	L\$DLY,(PC)+	
(3)	032106	000000			.WORD	0	
(3)	032110	005367	177772		DEC	-6(PC)	
(3)	032114	001375			BNE	.-4	
(3)	032116	005367	177756		DEC	-22(PC)	
(3)	032122	001367			BNE	.-20	
3144	032124	005737	003006		TST	DONE	;CHECK IF INTERRUPTED
3145	032130	001012			BNE	12\$;YES - SKIP
3146	032132	004737	016324		JSR	PC,WAITIN	;WAIT FOR TIMEOUT
3147	032136	012603			MOV	(SP)+,R3	;GET ERROR POINTER
3148	032140	001406			BEQ	12\$;SKIP IF 0
3149	032142				ERRHRD	601.,GSTER1,ERR1	
(4)	032142	104456			TRAP	C\$ERHRD	
(5)	032144	001131			.WORD	601	
(5)	032146	006462			.WORD	GSTER1	
(5)	032150	012344			.WORD	ERR1	
3150	032152				EXIT	SUB	
(3)	032152	104432			TRAP	C\$EXIT	
(3)	032154	000140			.WORD	L10034-	
3151	032156			12\$:	TIMDLY	#2	;WAIT FOR DSE TO SET
(8)	032156	012746	000340		MOV	#340,-(SP)	
(7)	032162	012746	016116		MOV	#CLKINT,-(SP)	
(6)	032166	012746	000104		MOV	#104,-(SP)	
(5)	032172	012746	000003		MOV	#3,-(SP)	
(4)	032176	104437			TRAP	C\$\$VEC	
(3)	032200	062706	000010		ADD	#10,SP	
3152	032240	004737	016564		JSR	PC,GSTAT	;GET STATUS
3153	032244	032306			60\$		
3154	032246	032737	000400	003052	BIT	#DSESTAT,T.MP	;TEST IF DRIVE SELECT ERROR SET
3155	032254	001010			BNE	16\$;YES - SKIP
3156	032256	012703	010511		MOV	#MDSERR,R3	;SET NAME MESSAGE POINTER
3157	032262				ERRHRD	602.,,ERR3	
(4)	032262	104456			TRAP	C\$ERHRD	
(5)	032264	001132			.WORD	602	
(5)	032266	000000			.WORD	0	
(5)	032270	012460			.WORD	ERR3	
3158	032272				EXIT	SUB	
(3)	032272	104432			TRAP	C\$EXIT	
(3)	032274	000020			.WORD	L10034-	
3159	032276	010562	000000		MOV	R5,RLCS(R2)	;LOAD IN DIFFERENT ADDRESS
3160	032302	005304			DEC	R4	;DEC COUNT
3161	032304	001216			BNE	8\$;LOOP IF NOT ZERO

```

3162 032306 012737 000002 003016 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
3163 032314 ENDSUB
(3) 032314 L10034:
(3) 032314 104403
3164 032316 15$: TRAP C$ESUB
(8) 032316 012746 007744 PRINTF #FMT9,#OPR11 ;REQUEST PLUG CHANGE
(7) 032322 012746 011632 MOV #OPR11,-(SP)
(6) 032326 012746 000002 MOV #FMT9,-(SP)
(3) 032332 010600 MOV #2,-(SP)
(4) 032334 104417 MOV SP,RO
(4) 032336 062706 000006 TRAP C$PNTF
3165 032342 005037 004362 ADD #6,SP
3166 032346 CLR OBUFF ;CLEAR FOR RESPONSE
(3) 032346 104443 GMANIL OPR002,OBUFF,1,NO
(3) 032350 000404 TRAP C$GMAN
(4) 032352 004362 BR 10000$
(5) 032354 000120 .WORD OBUFF
(5) 032356 007322 .WORD T$CODE
(5) 032360 000001 .WORD OPR002
(3) 032362 10000$: .WORD 1
3167 032362 005737 004362 TST OBUFF ;TEST RESPONSE YES
3168 032366 001753 BEQ 15$ ;NO - SKIP
3169 032370
3170 032370 LCLEXT:
(3) 032370 ENDTST
(3) 032370 104401 L10033:
TRAP C$ETST
  
```



```

3172 .SBTTL *TEST 7 INITIAL STATE
3173 BGNTST ;TEST 07
(3) 032372
3174 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE T7::
3175 032372 005737 003144 TST CLKFLG ;P-CLOCK?
3176 032376 001024 BNE 1$ ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3177 032400 012702 006662 MOV #NOTST,R2 ;INITIALIZE POINTER FOR TEST MSG.
3178 032404 112762 000060 000004 MOVB #'0,4(R2) ;INSERT TEST NUMBER INTO MSG.
3179 032412 112762 000067 000005 MOVB #'7,5(R2) ;INSERT TEST NUMBER INTO MSG.
3180 032420 PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 7 CANNOT BE PERFORMED...
(8) 032420 012746 006662 MOV #NOTST,-(SP)
(7) 032424 012746 011632 MOV #FMT9,-(SP)
(6) 032430 012746 000002 MOV #2,-(SP)
(3) 032434 010600 MOV SP,R0
(4) 032436 104417 TRAP C$PNTF
(4) 032440 062706 000006 ADD #6,SP
3181 ;/NO P-CLK''
3182 032444 EXIT TST
(3) 032444 104432 TRAP C$EXIT
(3) 032446 000420 .WORD L10035-
3183 032450 012737 006513 003012 1$: MOV #INITST,ERHEAD ;SET ERROR HEADER
3184 032456 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3185 032462 TIMDLY #10. ;WAIT 1 MS
(8) 032462 012746 000340 MOV #340,-(SP)
(7) 032466 012746 016116 MOV #CLKINT,-(SP)
(6) 032472 012746 000104 MOV #104,-(SP)
(5) 032476 012746 000003 MOV #3,-(SP)
(4) 032502 104437 TRAP C$SVEC
(3) 032504 062706 000010 ADD #10,SP
3186 032544 004737 016550 JSR PC,GSTATC ;GET STATUS
3187 032550 033066 65$
3188 032552 032737 000001 003044 BIT #DRDYMSK,T.CS ;CHECK IF DRIVE READY
3189 032560 001003 BNE 3$ ;YES-SKIP
3190 032562 012703 010263 MOV #MDRDY,R3 ;SET NAME MESSAGE POINTER
3191 032566 000430 BR 9$ ;GO REPORT
3192 032570 012703 000005 3$: MOV #5,R3 ;SET EXPECTED STATE VALUE
3193
3194 032574 020337 003060 CMP R3,T.STAT ;CHECK IF STATE OK
3195 032600 001406 BEQ 5$ ;YES-SKIP
3196 032602 ERRHRD 701,,ERR7 ;ELSE REPORT STATE ERROR
(4) 032602 104456 TRAP C$ERRHD
(5) 032604 001275 .WORD 701
(5) 032606 000000 .WORD 0
(5) 032610 013546 .WORD ERR7
3197 032612 EXIT TST ;EXIT
(3) 032612 104432 TRAP C$EXIT
(3) 032614 000252 .WORD L10035-
3198 032616 013701 003052 5$: MOV T.MP,R1 ;GET MP REG
3199 032622 032701 000020 BIT #HOSTAT,R1 ;CHECK HEADS OUT
3200 032626 001003 BNE 7$ ;YES-SKIP
3201 032630 012703 010500 MOV #MHOSTA,R3 ;SET NAME MESSAGE PTR
3202 032634 000405 BR 9$ ;GO REPORT
3203 032636 032701 000010 7$: BIT #BHSTAT,R1 ;CHECK BRUSH HOME SET
3204 032642 001010 BNE 10$ ;YES-SKIP
3205 032644 012703 010454 MOV #MBHSTA,R3 ;SET NAME MESSAGE PTR
3206 032650 9$: ERRHRD 702,,ERR3 ;REPORT ERROR

```


(4)	032650	104456			TRAP	C\$ERHRD	
(5)	032652	001276			.WORD	702	
(5)	032654	000000			.WORD	0	
(5)	032656	012460			.WORD	ERR3	
3207	032660				EXIT	TST	:EXIT
(3)	032660	104432			TRAP	C\$EXIT	
(3)	032662	000204			.WORD	L10035-	
3208	032664	005737	014206	10\$:	TST	MISWIW	:TEST IF MANUAL INTERVENTION RUN
3209	032670	100035			BPL	16\$:NO-SKIP
3210	032672	005737	003356		TST	PASNUM	:CHECK IF FIRST PASS
3211	032676	001032			BNE	16\$:NO-SKIP
3212	032700	032701	000100		BIT	#HSSTAT,R1	:ELSE CHECK HD 0 SELECTED
3213	032704	001412			BEQ	13\$:YES-SKIP
3214	032706	012703	010416		MOV	#MHSTA,R3	:SET NAME MESSAGE PTR
3215	032712	012704	011271		MOV	#CCYLUP,R4	:SET CONDITION POINTER
3216	032716				ERRHRD	703...ERR4	:REPORT ERROR
(4)	032716	104456			TRAP	C\$ERHRD	
(5)	032720	001277			.WORD	703	
(5)	032722	000000			.WORD	0	
(5)	032724	012526			.WORD	ERR4	
3217	032726				EXIT	TST	:EXIT
(3)	032726	104432			TRAP	C\$EXIT	
(3)	032730	000136			.WORD	L10035-	
3218	032732	032701	001000	13\$:	BIT	#VCSTAT,R1	:CHECK VOL CHECK SET
3219	032736	001003			BNE	15\$:YES-SKIP
3220	032740	012703	010430		MOV	#MVOLCK,R3	:ELSE SET NAME MESSAGE PTR
3221	032744	000741			BR	9\$:GO REPORT
3222	032746	032737	040000	003044	15\$:	BIT	#DRVERR,T.CS
3223	032754	001003			BNE	16\$:YES-SKIP
3224	032756	012703	010405		MOV	#MDRERR,R3	:ELSE SET NAME MESSAGE PTR
3225	032762	000732			BR	9\$:GO REPORT
3226	032764	032701	020000	16\$:	BIT	#WLSTAT,R1	:CHECK WRITE LOCK STATUS
3227	032770	001406			BEQ	17\$:SKIP IF RESET
3228	032772	012703	010467		MOV	#MWLSTA,R3	:ELSE SET NAME MESSAGE PTR
3229	032776				ERRHRD	705...ERR2	
(4)	032776	104456			TRAP	C\$ERHRD	
(5)	033000	001301			.WORD	705	
(5)	033002	000000			.WORD	0	
(5)	033004	012412			.WORD	ERR2	
3230	033006	042701	021177	17\$:	BIC	#21177,R1	:CLEAR STAUUS EXCEPT FOR ERROR BITS
3231	033012	023727	002276	000001	CMP	T.DRIVE,#1	
3232	033020	001404			BEQ	99\$	
3233	033022	022701	000200		CMP	#200,R1	
3234	033026	001411			BEQ	19\$	
3235	033030	000402			BR	18\$	
3236	033032	005701			99\$:	TST	R1
3237	033034	001406			BEQ	19\$:NO-SKIP
3238	033036				18\$:	ERRHRD	704...ERR6
(4)	033036	104456			TRAP	C\$ERHRD	:ELSE REPORT ALL ERRORS
(5)	033040	001300			.WORD	704	
(5)	033042	000000			.WORD	0	
(5)	033044	012646			.WORD	ERR6	
3239	033046				EXIT	TST	:EXIT
(3)	033046	104432			TRAP	C\$EXIT	
(3)	033050	000016			.WORD	L10035-	
3240	033052	013701	003044	19\$:	MOV	T.CS,R1	:GET CS REG

3241	033056	042701	141777	BIC	#141777,R1	:CLEAR ALL BUT ERROR BITS
3242	033062	005701		TST	R1	:TEST IF ANY ERROR SET
3243	033064	001364		BNE	18\$:YES-SKIP TO REPORT
3244	033066					
3245	033066					
3246	033066					
(3)	033066					
(3)	033066	104401		TRAP	C\$ETST	

25\$:
65\$:
ENDTST
L10035:

```
3248
3249
3250          .SBTTL *TEST 8          INITIAL RESET STATE
3251 033070    BGNTST                  ;TEST 8
3252 (3) 033070
3253 033070 012737 006513 003012    MOV #INITST,ERHEAD          T8::
3254 033076 004737 016516          JSR PC,TSTINT              ;INITIALIZE TEST
3255 033102 004737 016534          JSR PC,GSTATR            ;GET STATUS WITH RESET
3256 033106 033154 65$
3257 033110 005737 014206          TST MISWIW              ;CHECK IF MAN INTERVENTION WAS RUN
3258 033114 100017 4$              BPL 4$                  ;NO-SKIP
3259 033116 005737 003356          TST PASNUM              ;CHECK IF 1ST PASS
3260 033122 001014 4$              BNE 4$                  ;NO-SKIP
3261 033124 032737 000100 003052  BIT #HSSTAT,T.MP        ;CHECK HD SELECT STILL 0
3262 033132 001410 4$              BEQ 4$                  ;YES-SKIP
3263 033134 012703 010416          MOV #MHSTA,R3           ;SET NAME MESSAGE PTR
3264 033140 012704 011271          MOV #CCYLUP,R4         ;SET CONDITION POINTER
3265 033144          ERRHRD 801,,,ERR4      ;REPORT ERROR
3266 (4) 033144 104456          TRAP CSERHRD
3267 (5) 033146 001441          .WORD 801
3268 (5) 033150 000000          .WORD 0
3269 (5) 033152 012526          .WORD ERR4
3266 033154          4$:
3267 033154          65$:
3268 033154          ENDTST
3269 (3) 033154          L10036:
3269 (3) 033154 104401          TRAP C$ETST
```



```

3271
3272
3273
3274 033156          .SBTTL *TEST 9          DRIVE READY
      (3) 033156      BGNTST          ;TEST 9
3275                                T9::
3276 033156 005737 003144 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3277 033162 001024      TST          CLKFLG          ;P-CLOCK?
3278 033164 012702 006662      BNE          1$          ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
3279 033170 112762 000060 000004      MOV          #NOTST,R2      ;INITIALIZE POINTER FOR TEST MSG.
3280 033176 112762 000071 000005      MOVB         #'0,4(R2)      ;INSERT TEST NUMBER INTO MSG.
3281 033204          PRINTF        #FMT9,#NOTST      ;INSERT TEST NUMBER INTO MSG.
      (8) 033204 012746 006662      MOV          #NOTST,-(SP)   ;PRINT MSG. 'TEST 9 CANNOT BE PERFORMED...
      (7) 033210 012746 011632      MOV          #FMT9,-(SP)
      (6) 033214 012746 000002      MOV          #2,-(SP)
      (3) 033220 010600          MOV          SP,R0
      (4) 033222 104417          TRAP         C$PNTF
      (4) 033224 062706 000006      ADD          #6,SP
3282
3283 033230          EXIT          TST
      (3) 033230 104432          TRAP         C$EXIT
      (3) 033232 000370          .WORD        L10037-
3284 033234 012737 006541 003012 1$:  MOV          #T09ERR,ERHEAD ;SET ERROR HEADER
3285 033242 012701 003102          MOV          #NEWCYL,R1    ;GET POINTER TO DESIRED LOC
3286 033246 005021          CLR          (R1)+        ;CLEAR NEW CYL
3287 033250 005021          CLR          (R1)+        ;CLEAR CURRENT CYL
3288 033252 005021          CLR          (R1)+        ;DIFFERENCE
3289 033254 005011          CLR          (R1)         ;SIGN
3290 033256 004737 016516      JSR          PC,TSTINT     ;INITIALIZE TEST
3291 033262 004737 016534      JSR          PC,GSTATR    ;GET STATUS WITH RESET
3292 033266 033622          65$
3293 033270 004737 022272      JSR          PC,POSHSB    ;POSITION HEAD SELECTED BIT
3294 033274 010537 003112      MOV          R5,DESHD     ;STORE AS DESIRED HEAD
3295 033300 004737 020626      JSR          PC,SIMSEK    ;EXECUTE SIMPLE SEEK
3296 033304 033622          65$
3297 033306 012703 010263      MOV          #MDRDY,R3    ;SET NAME MESSAGE PTR
3298 033312 012704 011232      MOV          #CDRDY,R4    ;SET CONDITION POINTER
3299 033316 004737 016564      JSR          PC,GSTAT     ;GET STATUS
3300 033322 033622          65$
3301 033324 032737 000001 003044  BIT          #DRDYMSK,T.CS ;TEST READY SET
3302 033332 001406          BEQ          4$          ;NO-SKIP
3303 033334          ERRHRD        901,,,ERR4 ;REPORT READY ERROR
      (4) 033334 104456          TRAP         C$ERHRD
      (5) 033336 001605          .WORD        901
      (5) 033340 000000          .WORD        0
      (5) 033342 012526          .WORD        ERR4
3304 033344          EXIT          TST          ;EXIT
      (3) 033344 104432          TRAP         C$EXIT
      (3) 033346 000254          .WORD        L10037-
3305 033350 012701 000121          4$: MOV          #81,R1      ;SET WAIT COUNT
3306 033354 004737 016564          5$: JSR          PC,GSTAT     ;GET STATUS
3307 033360 033622          65$
3308 033362 012703 000005      MOV          #5,R3        ;SET EXPECTED STATE VALUE
3309 033366 023703 003060      CMP          T,STAT,R3    ;CHECK STATE IS 5
3310 033372 001406          BEQ          7$          ;YES-SKIP
3311 033374          ERRHRD        902,,,ERR7 ;ELSE REPORT

```

(4)	033374	104456				TRAP	C\$ERHRD	
(5)	033376	001606				.WORD	902	
(5)	033400	000000				.WORD	0	
(5)	033402	013546				.WORD	ERR7	
3312	033404					EXIT	TST	
(3)	033404	104432				TRAP	C\$EXIT	
(3)	033406	000214				.WORD	L10037-	
3313	033410	012703	010263		7\$:	MOV	#MDRDY,R3	
3314	033414	032737	000001	003044		BIT	#DRDYMSK,T.CS	:CHECK READY SET
3315	033422	001042				BNE	12\$:YES-SKIP
3316	033424	005301				DEC	R1	:ELSE DEC WAIT COUNT
3317	033426	001432				BEQ	9\$:SKIP IF 0
3318	033430					TIMDLY	#1	
(8)	033430	012746	000340			MOV	#340,-(SP)	
(7)	033434	012746	016116			MOV	#CLKINT,-(SP)	
(6)	033440	012746	000104			MOV	#104,-(SP)	
(5)	033444	012746	000003			MOV	#3,-(SP)	
(4)	033450	104437				TRAP	C\$SVEC	
(3)	033452	062706	000010			ADD	#10,SP	
3319	033512	000720				BR	5\$	
3320	033514				9\$:	ERRHRD	903,,,ERR5	:REPORT READY ERROR
(4)	033514	104456				TRAP	C\$ERHRD	
(5)	033516	001607				.WORD	903	
(5)	033520	000000				.WORD	0	
(5)	033522	012576				.WORD	ERR5	
3321	033524					EXIT	TST	
(3)	033524	104432				TRAP	C\$EXIT	
(3)	033526	000074				.WORD	L10037-	
3322								
3323	033530	005737	003044		12\$:	TST	T.CS	:TEST IF ANY ERROR
3324	033534	100006				BPL	15\$:NO-SKIP
3325	033536					ERRHRD	904,,,ERR6	
(4)	033536	104456				TRAP	C\$ERHRD	
(5)	033540	001610				.WORD	904	
(5)	033542	000000				.WORD	0	
(5)	033544	012646				.WORD	ERR6	
3326	033546					EXIT	TST	
(3)	033546	104432				TRAP	C\$EXIT	
(3)	033550	000052				.WORD	L10037-	
3327	033552	012703	010416		15\$:	MOV	#MHSTA,R3	:SET NAME MESSAGE PTR
3328	033556	004737	022272			JSR	PC,POSHSB	:POSITION HEAD SELECT BIT FOR TEST
3329	033562	020537	003112			CMP	R5,DESHD	:CHECK IF CORRECT HEAD SELECTED
3330	033566	001415				BEQ	20\$:YES-SKIP
3331	033570	005737	003112			TST	DESHD	:ELSE TEST IF 1 DESIRED
3332	033574	001406				BEQ	17\$:NO-REPORT SB 0
3333	033576					ERRHRD	905,,,ERR3	:ELSE REPORT SB 1
(4)	033576	104456				TRAP	C\$ERHRD	
(5)	033600	001611				.WORD	905	
(5)	033602	000000				.WORD	0	
(5)	033604	012460				.WORD	ERR3	
3334	033606					EXIT	TST	
(3)	033606	104432				TRAP	C\$EXIT	
(3)	033610	000012				.WORD	L10037-	
3335	033612				17\$:	ERRHRD	906,,,ERR2	
(4)	033612	104456				TRAP	C\$ERHRD	
(5)	033614	001612				.WORD	906	

CZRLIB0 RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 E 10 PAGE 2-66
*TEST 9 DRIVE READY

SEQ 0121

(5) 033616 000000
(5) 033620 012412
3336 033622
3337 033622
3338 033622
(3) 033622
(3) 033622 104401

.WORD 0
.WORD ERR2
20\$:
65\$:
ENDTST
L10037:
TRAP C\$ETST

```
3340 .SBTTL *TEST 10 SEEK SIGN SWITCH
3341 033624 BGNSTST ;TEST 10
(3) 033624 T10::
3342 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3343 033624 005737 003144 TST CLKFLG ;P-CLOCK?
3344 033630 001024 BNE 1$ ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3345 033632 012702 006662 MOV #NOTST,R2 ;INITIALIZE POINTER FOR TEST MSG.
3346 033636 112762 000061 000004 MOV #1,4(R2) ;INSERT TEST NUMBER INTO MSG.
3347 033644 112762 000060 000005 MOV #0,5(R2) ;INSERT TEST NUMBER INTO MSG.
3348 033652 PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 10 CANNOT BE PERFORMED...
(8) 033652 012746 006662 MOV #NOTST,-(SP)
(7) 033656 012746 011632 MOV #FMT9,-(SP)
(6) 033662 012746 000002 MOV #2,-(SP)
(3) 033666 010600 MOV SP,R0
(4) 033670 104417 TRAP C$PNTF
(4) 033672 062706 000006 ADD #6,SP
3349 ;/NO P-CLK''
3350 033676 EXIT TST
(3) 033676 104432 TRAP C$EXIT
(3) 033700 000412 .WORD L10040-
3351 033702 012737 006551 003012 1$: MOV #T10ERR,ERHEAD ;SET ERROR HEADER
3352 033710 012701 003102 MOV #NEWCYL,R1
3353 033714 005021 CLR (R1)+ ;CLEAR NEW CYL
3354 033716 005021 CLR (R1)+ ;CLEAR CURRENT CYLINDER
3355 033720 005021 CLR (R1)+ ;CLEAR DIFFERENCE
3356 033722 052721 000001 BIS #BIT0,(R1)+ ;SET FOR SIGN OF 1
3357 033726 004737 022272 JSR PC,POSHSB ;GET SELECTED HEAD
3358 033732 010521 MOV R5,(R1)+ ;SET AS DESIRED HEAD
3359 033734 T104$:
3360 033734 BGNSUB
(3) 033734 T10.1:
(3) 033734 104402 TRAP C$BSUB
3361 033736 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3362 033742 004737 016534 JSR PC,GSTAT ;GET STATUS
3363 033746 034272 60$
3364 033750 004737 020626 JSR PC,SIMSEK ;DO SEEK
3365 033754 034272 60$
3366 033756 012703 010263 MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
3367 033762 012704 011232 MOV #CDRDY,R4 ;SET CONDITION MESSAGE PTR
3368 033766 004737 016564 JSR PC,GSTAT ;GET STATUS
3369 033772 034272 60$
3370 033774 032737 000001 003044 BIT #DRDYMSK,T.CS ;CHECK READY RESET
3371 034002 001406 BEQ 4$ ;YES-SKIP
3372 034004 ERRHRD 1001...ERR4 ;REPORT READY ERROR
(4) 034004 104456 TRAP C$ERHRD
(5) 034006 001751 .WORD 1001
(5) 034010 000000 .WORD 0
(5) 034012 012526 .WORD ERR4
3373 034014 EXIT SUB ;EXIT SUBTEST
(3) 034014 104432 TRAP C$EXIT
(3) 034016 000254 .WORD L10041-
3374
3375
3376 034020 012701 000121 4$: MOV #81.,R1 ;SET WAIT COUNT
3377 034024 004737 016564 5$: JSR PC,GSTAT ;GET STATUS
3378 034030 034272 60$
```


3379	034032	012703	000005			MOV	#5,R3	:SET EXPECTED STATE
3380	034036	020337	003060			CMP	R3,T.STAT	:CHECK STATE IS 5
3381	034042	001406				BEQ	7\$:YES-SKIP
3382	034044					ERRHRD	1002...ERR7	:REPORT STATE ERROR
(4)	034044	104456				TRAP	C\$ERHRD	
(5)	034046	001752				.WORD	1002	
(5)	034050	000000				.WORD	0	
(5)	034052	013546				.WORD	ERR7	
3383	034054					EXIT	SUB	:EXIT
(3)	034054	104432				TRAP	C\$EXIT	
(3)	034056	000214				.WORD	L10041-	
3384	034060	012703	010263		7\$:	MOV	#MDRDY,R3	:SET NAME MESSAGE PTR
3385	034064	032737	000001	003044		BIT	#DRDYMSK,T.CS	:CHECK READY SET
3386	034072	001042				BNE	12\$:YES-SKIP
3387	034074	005301				DEC	R1	:DO WAIT COUNT
3388	034076	001432				BEQ	9\$:SKIP IF 0
3389	034100					TIMDLY	#1	
(8)	034100	012746	000340			MOV	#340,-(SP)	
(7)	034104	012746	016116			MOV	#CLKINT,-(SP)	
(6)	034110	012746	000104			MOV	#104,-(SP)	
(5)	034114	012746	000003			MOV	#3,-(SP)	
(4)	034120	104437				TRAP	C\$SVEC	
(3)	034122	062706	000010			ADD	#10,SP	
3390	034162	000720				BR	5\$	
3391								
3392	034164				9\$:	ERRHRD	1003...ERR5	:REPORT READY ERROR
(4)	034164	104456				TRAP	C\$ERHRD	
(5)	034166	001753				.WORD	1003	
(5)	034170	000000				.WORD	0	
(5)	034172	012576				.WORD	ERR5	
3393	034174					EXIT	SUB	:EXIT
(3)	034174	104432				TRAP	C\$EXIT	
(3)	034176	000074				.WORD	L10041-	
3394	034200	005737	003044		12\$:	TST	T.CS	:TEST IF ANY OTHER ERROR
3395	034204	100006				BPL	15\$:NO-SKIP
3396	034206					ERRHRD	1004...ERR6	:REPORT ALL ERRORS
(4)	034206	104456				TRAP	C\$ERHRD	
(5)	034210	001754				.WORD	1004	
(5)	034212	000000				.WORD	0	
(5)	034214	012646				.WORD	ERR6	
3397	034216					EXIT	SUB	:EXIT
(3)	034216	104432				TRAP	C\$EXIT	
(3)	034220	000052				.WORD	L10041-	
3398								
3399	034222	012703	010416		15\$:	MOV	#MHSTA,R3	:SET NAME MESSAGE PTR
3400	034226	004737	022272			JSR	PC,POSHSB	:GET SELECTED HEAD BIT
3401	034232	020537	003112			CMP	R5,DESHD	:CHECK IF CORRECT
3402	034236	001415				BEQ	20\$:YES - SKIP
3403	034240	005737	003112			TST	DESHD	:WAS IT SET
3404	034244	001406				BEQ	17\$:NO-SKIP
3405	034246					ERRHRD	1005...ERR3	:REPORT SB 1
(4)	034246	104456				TRAP	C\$ERHRD	
(5)	034250	001755				.WORD	1005	
(5)	034252	000000				.WORD	0	
(5)	034254	012460				.WORD	ERR3	
3406	034256					EXIT	SUB	

CZRLIB0 RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 H 10 PAGE 2-69
*TEST 10 SEEK SIGN SWITCH

SEQ 0124

(3) 034256 104432
(3) 034260 000012
3407 034262
(4) 034262 104456
(5) 034264 001756
(5) 034266 000000
(5) 034270 012412
3408
3409 034272
3410 034272
3411 034272
(3) 034272
(3) 034272 104403
3412 034274 005737 003110
3413 034300 001404
3414 034302 005037 003110
3415 034306 000137 033734
3416 034312
3417 034312
(3) 034312
(3) 034312 104401

17\$: TRAP C\$EXIT
.WORD L10041-
ERRHRD 1006.,ERR2 ;REPORT SB 0
TRAP C\$ERHRD
.WORD 1006
.WORD 0
.WORD ERR2
20\$:
60\$:
ENDSUB
L10041:
TRAP C\$ESUB
TST DESSGN ;CHECK IF BOTH SIGN USED
BEQ 25\$;YES-SKIP
CLR DESSGN ;SET FOR SIGN OF 0
JMP T104\$;DO TEST AGAIN
25\$:
ENDTST
L10040:
TRAP C\$ETST


```

3419
3420
3421          .SBTTL *TEST 11          HEAD ALIGNMENT SUPPORT
3422 034314    BGNSTST                ;TEST 11
(3) 034314
3423 034314 032737 000010 014206    BIT    #HDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
3424 034322 001411                    BEQ    1$                ;NO-EXIT
3425 034324 005737 003356            TST    PASNUM          ;TEST IF PASS 0
3426 034330 001006                    BNE    1$                ;NO-EXIT
3427 034332 023737 003032 003010    CMP    RLDRV,HADONE    ;TEST IF HEAD ALIGN DONE THIS DRIVE
3428 034340 001004                    BNE    2$                ;NO - SKIP
3429 034342 000137 034764            JMP    T115$           ;GO CHECK WRITE LOCK
3430 034346          1$:              EXIT    TST
(3) 034346 104432                    TRAP   C$EXIT
(3) 034350 000514                    .WORD  L10042-
3431 034352 013737 003032 003010 2$:  MOV    RLDRV,HADONE    ;SET HEAD ALIGN DONE FLAG
3432 034360          PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
(11) 034360 005046                    CLR    -(SP)
(11) 034362 153716 003033            BISB  RLDRV+1,(SP)
(10) 034366 012746 006051            MOV   #DRVNAM,-(SP)
(9) 034372 013746 003026            MOV   RLBAS,-(SP)
(8) 034376 012746 006040            MOV   #BASADD,-(SP)
(7) 034402 012746 011446            MOV   #FMT5,-(SP)
(6) 034406 012746 000005            MOV   #5,-(SP)
(3) 034412 010600                    MOV   SP,R0
(4) 034414 104417                    TRAP  C$PNTF
(4) 034416 062706 000014            ADD   #14,SP
3433 034422          PRINTF #FMT9,#HAMES1 ;TYPE INSTRUCTIONS
(8) 034422 012746 007130            MOV   #HAMES1,-(SP)
(7) 034426 012746 011632            MOV   #FMT9,-(SP)
(6) 034432 012746 000002            MOV   #2,-(SP)
(3) 034436 010600                    MOV   SP,R0
(4) 034440 104417                    TRAP  C$PNTF
(4) 034442 062706 000006            ADD   #6,SP
3434 034446          PRINTF #FMT9,#HAMES2
(8) 034446 012746 007213            MOV   #HAMES2,-(SP)
(7) 034452 012746 011632            MOV   #FMT9,-(SP)
(6) 034456 012746 000002            MOV   #2,-(SP)
(3) 034462 010600                    MOV   SP,R0
(4) 034464 104417                    TRAP  C$PNTF
(4) 034466 062706 000006            ADD   #6,SP
3435
3436 034472          BGNSUB
(3) 034472
(3) 034472 104402
3437 034474 004737 016516          3$:  TRAP  C$BSUB                T11.1:
3438 034500 005037 003006          JSR   PC,TSTINT        ;INITIALIZE TEST
3439 034504 013737 003032 003034    CLR   DONE            ;CLEAR DONE
3440 034512 052737 000104 003034    MOV   RLDRV,L.CS      ;SET UP FOR GET STATUS
3441 034520 012737 000013 003040    BIS   #GTSTAT,L.CS
3442 034526 013762 003040 000004    MOV   #GETSTAT!DRSET,L.DA
3443 034534 013762 003034 000000    MOV   L.DA,RLDA(R2)
3444 034542 012737 000031 003142    MOV   L.CS,RLCSR(R2) ;DO GET STATUS
3445 034550 006337 003142          MOV   #25.,DLYCNT     ;INITIALIZE DELAY COUNT
3446 034554 006337 003142          ASL   DLYCNT          ;MULTIPLY ARGUMENT BY 2
3447 034560          4$:  ASL   DLYCNT          ;MULTIPLY ARGUMENT BY 2 AGAIN
          DELAY #250.    ;IMPLEMENT TIME DELAY

```

```

(2) 034560 012727 000372      MOV    ##250.,(PC)+
(2) 034564 000000              .WORD 0
(2) 034566 013727 002116      MOV    L$DLY,(PC)+
(2) 034572 000000              .WORD 0
(2) 034574 005367 177772      DEC    -6(PC)
(2) 034600 001375              BNE    -4
(2) 034602 005367 177756      DEC    -22(PC)
(2) 034606 001367              BNE    -20
3448 034610 005337 003142      DEC    DLYCNT          ;DECREMENT DELAY COUNT
3449 034614              BREAK          ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
(3) 034614 104422              TRAP    C$BRK
3450                          ;/BACK TO SUPERVISOR COMMAND MODE
3451 034616 001360              BNE    4$              ;BRANCH IF TIME DELAY NOT EXPIRED
3452 034620 005737 003006      TST    DONE            ;CHECK IF DONE
3453 034624 001723              BEQ    3$              ;NO-GO CLR CONTROLLER
3454
3455
3456 034626 012737 000021 003040 10$:  MOV    #HDSEL!MBSETO,L.DA;LOAD FOR HEAD 1
3457 034634 032737 020000 003052      BIT    #WLSTAT,T.MP   ;CHECK IF WRITE LOCK SET
3458 034642 001003              BNE    12$            ;YES-SKIP
3459 034644 042737 000020 003040      BIC    #HDSEL,L.DA    ;ELSE CLEAR TO HEAD 0
3460 034652 013737 003032 003034 12$:  MOV    RLDRV,L.CS     ;LOAD IN DRIVE NUMBER
3461 034660 052737 000106 003034      BIS    #SEEK,L.CS    ;SET FOR SEEK
3462 034666 013762 003040 000004      MOV    L.DA,RLDA(R2) ;LOAD & EXECUTE SEEK
3463 034674 013762 003034 000000      MOV    L.CS,RLCSR(R2)
3464 034702 012737 000017 003142      MOV    #15.,DLYCNT   ;INITIALIZE DELAY COUNT
3465 034710 006337 003142      ASL    DLYCNT         ;MULTIPLY ARGUMENT BY 2
3466 034714 006337 003142      ASL    DLYCNT         ;MULTIPLY ARGUMENT BY 2 AGAIN
3467 034720              5$:  DELAY  #250.          ;IMPLEMENT TIME DELAY
(2) 034720 012727 000372      MOV    ##250.,(PC)+
(2) 034724 000000              .WORD 0
(2) 034726 013727 002116      MOV    L$DLY,(PC)+
(2) 034732 000000              .WORD 0
(2) 034734 005367 177772      DEC    -6(PC)
(2) 034740 001375              BNE    -4
(2) 034742 005367 177756      DEC    -22(PC)
(2) 034746 001367              BNE    -20
3468 034750 005337 003142      DEC    DLYCNT          ;DECREMENT DELAY COUNT
3469 034754              BREAK          ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
(3) 034754 104422              TRAP    C$BRK
3470                          ;/BACK TO SUPERVISOR COMMAND MODE
3471 034756 001360              BNE    5$              ;BRANCH IF TIME DELAY NOT EXPIRED
3472 034760 000645              BR     3$              ;LOOP
3473
3474
3475 034762              59$:  ENDSUB
(3) 034762 L10043:
(3) 034762 104403              TRAP    C$ESUB
3475 034764              T115$:  BGNSUB
3476 034764
(3) 034764
(3) 034764 104402              TRAP    C$BSUB          T11.2:
3477 034766 004737 016516      JSR    PC,TSTINT      ;INITIALIZE TEST
3478 034772 004737 016534      JSR    PC,GSTATR     ;CLEAR DRIVE
3479 034776 035062      60$
3480 035000 032737 020000 003052      BIT    #WLSTAT,T.MP  ;CHECK WRITE LOCK RESET
3481 035006 001425              BEQ    19$            ;YES-SKIP

```


3482	035010			18\$:	PRINTF	#FMT9,#OPR12	:REQUEST WRITE LOCK RESET
(8)	035010	012746	007775		MOV	#OPR12,-(SP)	
(7)	035014	012746	011632		MOV	#FMT9,-(SP)	
(6)	035020	012746	000002		MOV	#2,-(SP)	
(3)	035024	010600			MOV	SP,R0	
(4)	035026	104417			TRAP	C\$PNTF	
(4)	035030	062706	000006		ADD	#6,SP	
3483	035034	005037	004362		CLR	OBUFF	:CLEAR FOR RESPONSE
3484	035040				GMANIL	OPR002,OBUFF,1,NO	:GET RESPONSE
(3)	035040	104443			TRAP	C\$GMAN	
(3)	035042	000404			BR	10000\$	
(4)	035044	004362			.WORD	OBUFF	
(5)	035046	000120			.WORD	T\$CODE	
(5)	035050	007322			.WORD	OPR002	
(5)	035052	000001			.WORD	1	
(3)	035054			10000\$:			
3485	035054	005737	004362		TST	OBUFF	:WAS ANSWER YES
3486	035060	001753			BEQ	18\$:NO-REPEAT REQUEST
3487	035062			19\$:			
3488	035062			60\$:			
3489	035062			ENDSUB			
(3)	035062			L10044:			
(3)	035062	104403			TRAP	C\$ESUB	
3490	035064			20\$:			
3491	035064			ENDTST			
(3)	035064			L10042:			
(3)	035064	104401			TRAP	C\$ETST	

```

3493
3494
3495
3496 035066 .SBTTL *TEST 12 HEAD SWITCHING
(3) 035066 BGNSTST ;TEST 12
3497 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE T12::
3498 035066 005737 003144 TST CLKFLG ;P-CLOCK?
3499 035072 001024 BNE 1$ ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3500 035074 012702 006662 MOV #NOTST,R2 ;INITIALIZE POINTER FOR TEST MSG.
3501 035100 112762 000061 000004 MOVB #1,4(R2) ;INSERT TEST NUMBER INTO MSG.
3502 035106 112762 000062 000005 MOVB #2,5(R2) ;INSERT TEST NUMBER INTO MSG.
3503 035114 PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 12 CANNOT BE PERFORMED...
(8) 035114 012746 006662 MOV #NOTST,-(SP)
(7) 035120 012746 011632 MOV #FMT9,-(SP)
(6) 035124 012746 000002 MOV #2,-(SP)
(3) 035130 010600 MOV SP,R0
(4) 035132 104417 TRAP C$PNTF
(4) 035134 062706 000006 ADD #6,SP
3504 ;/NO P-CLK''
3505 035140 EXIT TST
(3) 035140 104432 TRAP C$EXIT
(3) 035142 000406 .WORD L10045-
3506 035144 012737 006571 003012 1$: MOV #T12ERR,ERHEAD ;SET ERROR HEADER
3507 035152 012701 003102 MOV #NEWCYL,R1 ;GET POINTER TO DESIRED LOCATION
3508 035156 005021 CLR (R1)+ ;CLEAR NEW CYLINDER
3509 035160 005021 CLR (R1)+ ;CLEAR CURRENT CYL.
3510 035162 005021 CLR (R1)+ ;CLEAR DIFFERENCE
3511 035164 005021 CLR (R1)+ ;CLEAR SIGN
3512 035166 012721 000001 MOV #1,(R1)+ ;SET FOR HEAD 1
3513 035172
3514 035172 T124$:
(3) 035172 BGNSUB
(3) 035172 104402 TRAP C$BSUB T12.1:
3515 035174 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3516 035200 004737 016534 JSR PC,GSTATR ;GET STATUS WITH RESET
3517 035204 035530 60$
3518 035206 004737 020626 JSR PC,SIMSEK ;DO SEEK
3519 035212 035530 60$
3520 035214 012703 010263 MOV #MDRDY,R3 ;SET NAME MESSAGE PTR
3521 035220 012704 011232 MOV #CDRDY,R4 ;SET CONDITION POINTER
3522 035224 004737 016564 JSR PC,GSTAT ;GET STATUS
3523 035230 035530 60$
3524 035232 032737 000001 003044 BIT #DRDYMSK,T.CS ;CHECK IF READY
3525 035240 001406 BEQ 5$ ;NO-SKIP
3526 035242 ERRHRD 1201...ERR4 ;REPORT READY ERROR
(4) 035242 104456 TRAP C$ERHRD
(5) 035244 002261 .WORD 1201
(5) 035246 000000 .WORD 0
(5) 035250 012526 .WORD ERR4
3527 035252 EXIT SUB ;EXIT
(3) 035252 104432 TRAP C$EXIT
(3) 035254 000254 .WORD L10046-
3528
3529 035256 012701 000121 5$: MOV #81,R1 ;SET WAIT COUNT
3530 035262 004737 016564 6$: JSR PC,GSTAT ;GET STATUS
3531 035266 035530 60$

```


3532	035270	012703	000005		MOV	#5,R3	:SET EXPECTED STATE VALUE
3533	035274	020337	003060		CMP	R3,T.STAT	:CHECK IF STATE IS 5
3534	035300	001406			BEQ	7\$:YES-SKIP
3535	035302				ERRHRD	1202...ERR7	:REPORT STATE ERROR
(4)	035302	104456			TRAP	C\$ERHRD	
(5)	035304	002262			.WORD	1202	
(5)	035306	000000			.WORD	0	
(5)	035310	013546			.WORD	ERR7	
3536	035312				EXIT	SUB	
(3)	035312	104432			TRAP	C\$EXIT	
(3)	035314	000214			.WORD	L10046-	
3537							
3538	035316	012703	010263		MOV	#MDRDY,R3	:SET NAME MESSAGE PTR
3539	035322	032737	000001	003044	BIT	#DRDYMSK,T.CS	:CHECK DRIVE READY
3540	035330	001042			BNE	12\$:YES-SKIP
3541	035332	005301			DEC	R1	:DEC WAIT COUNT
3542	035334	001432			BEQ	9\$:SKIP IF 0
3543	035336				TIMDLY	#1	
(8)	035336	012746	000340		MOV	#340,-(SP)	
(7)	035342	012746	016116		MOV	#CLKINT,-(SP)	
(6)	035346	012746	000104		MOV	#104,-(SP)	
(5)	035352	012746	000003		MOV	#3,-(SP)	
(4)	035356	104437			TRAP	C\$SVEC	
(3)	035360	062706	000010		ADD	#10,SP	
3544	035420	000720			BR	6\$	
3545							
3546	035422				ERRHRD	1203...ERR5	:REPORT READY ERROR
(4)	035422	104456			TRAP	C\$ERHRD	
(5)	035424	002263			.WORD	1203	
(5)	035426	000000			.WORD	0	
(5)	035430	012576			.WORD	ERR5	
3547	035432				EXIT	SUB	:EXIT
(3)	035432	104432			TRAP	C\$EXIT	
(3)	035434	000074			.WORD	L10046-	
3548							
3549	035436	005737	003044		TST	T.CS	:TEST IF ANY ERROR
3550	035442	100006			BPL	15\$:NO-SKIP
3551	035444				ERRHRD	1204...ERR6	:REPORT ALL ERRORS
(4)	035444	104456			TRAP	C\$ERHRD	
(5)	035446	002264			.WORD	1204	
(5)	035450	000000			.WORD	0	
(5)	035452	012646			.WORD	ERR6	
3552	035454				EXIT	SUB	
(3)	035454	104432			TRAP	C\$EXIT	
(3)	035456	000052			.WORD	L10046-	
3553	035460	012703	010416		MOV	#MHSTA,R3	:SET NAME MESSAGE PTR
3554	035464	004737	022272		JSR	PC,POSHSB	:POSITION HEAD SELECT BIT
3555	035470	023705	003112		CMP	DESHD,R5	:CHECK IF CORRECT HEAD SELECTED
3556	035474	001415			BEQ	20\$:YES-SKIP
3557	035476	005737	003112		TST	DESHD	:WAS HEAD 0 SELECTED
3558	035502	001406			BEQ	17\$:YES-SKIP
3559	035504				ERRHRD	1205...ERR3	:REPORT HEAD SB 1
(4)	035504	104456			TRAP	C\$ERHRD	
(5)	035506	002265			.WORD	1205	
(5)	035510	000000			.WORD	0	
(5)	035512	012460			.WORD	ERR3	

```
3560 035514          EXIT      SUB          :EXIT
      (3) 035514 104432 TRAP      C$EXIT
      (3) 035516 000012 .WORD    L10046-
3561 035520          17$:     ERRHRD   1206...ERR2  :ELSE REPORT HEAD SB 0
      (4) 035520 104456 TRAP      C$ERHRD
      (5) 035522 002266 .WORD    1206
      (5) 035524 000000 .WORD    0
      (5) 035526 012412 .WORD    ERR2
3562
3563 035530          20$:
3564 035530          60$:
3565 035530          ENDSUB
      (3) 035530          L10046:
      (3) 035530 104403 TRAP      C$ESUB
3566 035532 005737 003112 TST      DESHD          :CHECK IF HD 0 WAS DONE
3567 035536 001404 BEQ      25$          :YES-SKIP
3568 035540 005037 003112 CLR      DESHD          :ELSE SET TO HEAD 0
3569 035544 000137 035172 JMP      T124$        :REDO TEST
3570 035550          25$:
3571 035550          ENDTST
      (3) 035550          L10045:
      (3) 035550 104401 TRAP      C$ETST
```



```

3573
3574
3575
3576 035552          .SBTTL *TEST 13          READ HEADER (PART 1)
      (3) 035552          BGNTST          ;TEST 13
3577 035552 012737 006603 003012          MOV #T13ERR,ERHEAD ;SET ERROR HEADER
3578 035560 012701 003102          MOV #NEWCYL,R1 ;GET ADDRESS OF DESIRED LOCATIONS
3579 035564 005021          CLR (R1)+ ;CLEAR NEW CYL
3580 035566 005021          CLR (R1)+ ;CLEAR CURRENT CYL
3581 035570 005021          CLR (R1)+ ;CLEAR DIFF
3582 035572 005021          CLR (R1)+ ;CLEAR SIGN
3583 035574 005021          CLR (R1)+ ;CLEAR HEAD
3584 035576          T134$:
3585 035576          BGNSUB
      (3) 035576          TRAP C$BSUB          T13.1:
      (3) 035576 104402          JSR PC,TSTINT ;INITIALIZE TEST
3586 035600 004737 016516          JSR PC,GSTATR ;GET STATUS W/RESET
3587 035604 004737 016534          60$
3588 035610 035702          JSR PC,SIMSEK ;DO SEEK
3589 035612 004737 020626          60$
3590 035616 035702          MOV #81.,R1 ;SET WAIT COUNT
3591 035620 012701 000121          JSR PC,RDYWAIT ;WAIT FOR READY
3592 035624 004737 022322          60$
3593 035630 035702
3594
3595 035632 004737 021602          10$: JSR PC,XRDHDC ;DO READ HEADER
3596 035636 035702          60$
3597 035640 012703 010416          MOV #MHSTA,R3 ;SET NAME MESSAGE PTR
3598 035644 004737 022264          JSR PC,POSHW1 ;POSITION HS BIT IN HD WRD 1
3599 035650 020537 003112          CMP R5,DESHD ;CHECK IF HEAD CORRECT
3600 035654 001412          BEQ 15$ ;YES-SKIP
3601 035656          ERRHRD 1301.,,ERR3 ;REPORT SB 1
      (4) 035656 104456          TRAP C$ERHRD
      (5) 035660 002425          .WORD 1301
      (5) 035662 000000          .WORD 0
      (5) 035664 012460          .WORD ERR3
3602 035666          EXIT SUB
      (3) 035666 104432          TRAP C$EXIT
      (3) 035670 000012          .WORD L10050-
3603 035672          17$: ERRHRD 1302.,,ERR2 ;REPORT SB 0
      (4) 035672 104456          TRAP C$ERHRD
      (5) 035674 002426          .WORD 1302
      (5) 035676 000000          .WORD 0
      (5) 035700 012412          .WORD ERR2
3604
3605 035702          15$:
3606 035702          60$:
3607 035702          ENDSUB
      (3) 035702          L10050:
      (3) 035702 104403          TRAP C$ESUB
3608 035704 005737 003112          TST DESHD ;TEST IF HEAD 1 DONE
3609 035710 001007          BNE 20$ ;YES-SKIP
3610 035712 012737 000001 003112          MOV #1,DESHD ;ELSE SET TO HEAD 1
3611 035720 013737 003052 003116          MOV HDWRD1,TEMPO ;STORE HDR WORD 1
3612 035726 000723          BR T134$ ;DO TEST AGAIN
3613 035730 042737 000177 003116 20$: BIC #177,TEMPO ;CLEAR ALL BUT CYLINDER IN 1ST HEADER

```

3614	035736	042737	000177	003052	BIC	#177,HDWRD1	:CLEAR ALL BY CYL IN 2ND HEADER
3615	035744	023737	003116	003052	CMP	TEMP0,HDWRD1	:COMPARE IF EQUAL
3616	035752	001406			BEQ	22\$:YES-SKIP
3617	035754	012703	007044		MOV	#CYLPER,R3	:SET NAME MESSAGE PTR
3618	035760				ERRHRD	1306.,,ERR1	:REPORT HEAD ALIGNMENT PROBLEM
(4)	035760	104456			TRAP	C\$ERHRD	
(5)	035762	002432			.WORD	1306	
(5)	035764	000000			.WORD	0	
(5)	035766	012344			.WORD	ERR1	
3619	035770						
3620	035770						
(3)	035770						
(3)	035770	104401					

22\$:
ENDTST
L10047:
TRAP C\$ETST


```

3622          .SBTTL *TEST 14      READ HEADER (PART 2)
3623 035772   BGNTST          ;TEST 14
(3) 035772
3624 035772   012737 006617 003012  MOV #T14ERR,ERHEAD ;SET ERROR HEADER      T14::
3625 036000   012701 003104  MOV #CURCYL,R1    ;GET ADDRESS OF DESIRED VALUE
3626 036004   005021          CLR (R1)+         ;CLEAR CURRENT CYL
3627 036006   005021          CLR (R1)+         ;CLEAR DESIRED DIFF
3628 036010   005021          CLR (R1)+         ;CLEAR SIGN
3629 036012   005021          CLR (R1)+         ;CLEAR DESIRED HEAD
3630 036014
3631 036014   T153$:
(3) 036014   BGNSUB
(3) 036014   104402          TRAP CS$SUB          T14.1:
3632 036016   004737 016516  JSR PC,TSTINT    ;INITIALIZE TEST
3633 036022   004737 016534  JSR PC,GSTATR   ;CLEAR DRIVE
3634 036026   036226          60$
3635 036030   004737 020626  JSR PC,SIMSEK   ;DO SEEK
3636 036034   036226          60$
3637 036036   012701 000310  MOV #200.,R1    ;SET WAIT COUNT FOR 20 MS
3638 036042   004737 022322  JSR PC,RDYWAIT  ;WAIT FOR READY
3639 036046   036226          60$
3640 036050   004737 023014  JSR PC,RDALHD   ;DO READ HEADER ALL HEADERS
3641 036054   036226          60$
3642 036056   005037 003014  CLR MORECE      ;CLEAR MORE COMPARE ERRORS FOR REPORT
3643 036062   052737 000002 003004  BIS #HDRCMP,OPFLAG ;SET HDR COMPARE FLAG
3644 036070   005003          CLR R3          ;CLEAR FOR HDR COUNT
3645 036072   012704 003762  MOV #IBUFF,R4   ;GET POINTER FOR HDR TO BE CHECKED
3646 036076   012705 003116  MOV #TEMPO,R5   ;GET POINTER TO TEST AREA
3647 036102   012701 000050  MOV #40.,R1     ;SET HDR COUNT
3648 036106   011415          MOV (R4),(R5)   ;GET FIRST HEADER WORD
3649
3650 036110   042715 000100  BIC #HDHSEL,(R5)
3651 036114   005737 003112  TST DESHD      ;TEST IF HD 0 DESIRED
3652 036120   001404          BEQ 10$        ;YES-SKIP
3653 036122   052715 000100  BIS #HDHSEL,(R5) ;ELSE SET HEAD BIT
3654 036126   005065 000002  CLR 2(R5)      ;CLEAR 2ND WORD OF TEST AREA
3655 036132   021524          10$: CMP (R5),(R4)+  ;COMPARE HEADER WORD
3656 036134   001406          BEQ 13$        ;SKIP IF OK
3657 036136   005744          TST -(R4)      ;ELSE POSITION R4 TO BAD WORD
3658 036140          ERRHRD 1501...ERR10 ;REPORT ERROR
(4) 036140   104456          TRAP C$ERHRD
(5) 036142   002735          .WORD 1501
(5) 036144   000000          .WORD 0
(5) 036146   013756          .WORD ERR10
3659 036150   005724          TST (R4)+      ;BUMP R4 TO NEXT WORD
3660 036152   005203          13$: INC R3          ;BUMP WORD COUNT
3661 036154   005724          TST (R4)+      ;TEST 2ND WORD IS 0
3662 036156   001406          BEQ 15$        ;YES - SKIP
3663 036160   022544          CMP (R5)+,-(R4) ;POSITION PTRS FOR REPORT
3664 036162          ERRHRD 1501...ERR10 ;REPORT ERROR
(4) 036162   104456          TRAP C$ERHRD
(5) 036164   002735          .WORD 1501
(5) 036166   000000          .WORD 0
(5) 036170   013756          .WORD ERR10
3665 036172   024524          15$: CMP -(R5),(R4)+ ;REPOSITION POINTER
3666 036174   005724          TST (R4)+      ;POSITION R4 PAST ECC WORD
  
```

```
3667 036176 005203          INC      R3          ;BUMP WORD COUNT
3668 036200 005215          INC      (R5)        ;BUMP SECTOR COUNT
3669 036202 011500          MOV      (R5),R0     ;CHECK IF SECTOR IS PAST LAST SECTOR
3670 036204 042700 177700   BIC      #^CHDSEC,R0
3671 036210 022700 000050   CMP      #40.,R0
3672 036214 001002          BNE     17$          ;NO-SKIP
3673 036216 042715 000077   BIC      #HDSEC,(R5) ;ELSE CLEAR SECTOR TO 0
3674 036222 005301          DEC     R1          ;DEC HDR COUNT
3675 036224 001342          BNE     10$          ;YES-SKIP
3676
3677 036226          60$:
3678 036226          ENDSUB
(3) 036226          L10052:
(3) 036226 104403          TRAP   C$ESUB
3679 036230 005737 003112   TST     DESHD        ;CHECK IF HD 1 TESTED
3680 036234 001005          BNE     20$          ;YES-SKIP
3681 036236 012737 000001 003112   MOV     #1,DESHD    ;ELSE SET TO HEAD 1
3682 036244 000137 036014   JMP     T153$        ;REDO TEST
3683 036250          20$:
3684 036250          ENDTST
(3) 036250          L10051:
(3) 036250 104401          TRAP   C$ETST
```



```
3686 .SBTTL *TEST 15 DIFFERENCE OF 1 SEEK (PART 1)
3687 BGNTST ;TEST 15
(3) 036252
3688 T15::
3689 ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
3690 036252 005737 003144 TST CLKFLG ;P-CLOCK?
3691 036256 001024 BNE 1$ ;BRANCH TO PERFORM TEST IF CLOCK IS PRESENT
3692 036260 012702 006662 MOV #NOTST,R2 ;INITIALIZE POINTER FOR TEST MSG.
3693 036264 112762 000061 000004 MOVB #'1,4(R2) ;INSERT TEST NUMBER INTO MSG.
3694 036272 112762 000065 000005 MOVB #'5,5(R2) ;INSERT TEST NUMBER INTO MSG.
3695 036300 PRINTF #FMT9,#NOTST ;PRINT MSG. 'TST 15 CANNOT BE PERFORMED...
(8) 036300 012746 006662 MOV #NOTST,-(SP)
(7) 036304 012746 011632 MOV #FMT9,-(SP)
(6) 036310 012746 000002 MOV #2,-(SP)
(3) 036314 010600 MOV SP,R0
(4) 036316 104417 TRAP C$PNTF
(4) 036320 062706 000006 ADD #6,SP
3696 ;/NO P-CLK''
3697 036324 EXIT TST
(3) 036324 104432 TRAP C$EXIT
(3) 036326 000444 .WORD L10053-
3698 036330 012737 006643 003012 1$: MOV #P2T01E,ERHEAD ;SET ERROR HEADER
3699 036336 012737 000004 003116 MOV #4,TEMP0 ;SET PASS COUNT
3700 036344 004737 016516 JSR PC,TSTINT ;INITIALIZE TEST
3701 036350 004737 016534 JSR PC,GSTATR ;GET STATUS
3702 036354 036772 T1765$
3703 036356 022737 000001 002276 CMP #1,T.DRIVE ;RLO1 OR RLO2?
3704 036364 001404 BEQ 2$ ;BRANCH TO SET UP DIFF ARGUMENT FOR RLO1
3705 036366 012737 177776 003122 MOV #-2,TEMP2 ;ELSE, SET -2 INTO DIFF ARGUMENT FOR RLO2
3706 ;/(RLO2 HAS DOUBLE THE TRACK DENSITY OF RLO1)
3707 036374 000403 BR 5$
3708 036376 012737 177777 003122 2$: MOV #-1,TEMP2 ;SET -1 INTO DIFF ARGUMENT FOR -1 SEEK
3709 036404 012704 003104 5$: MOV #CURCYL,R4 ;SET POINTERS
3710 036410 012705 003102 MOV #NEWCYL,R5
3711 036414 004737 021472 JSR PC,CHOSHD ;GO CHOOSE HEAD
3712 036420 T172$:
3713 036420 BGNSUB
(3) 036420
(3) 036420 104402 TRAP C$BSUB T15.1:
3714 036422 004737 022666 JSR PC,GETPOS ;GET POSITION
3715 036426 036730 60$
3716 036430 INLOOP ;CHECK IF IN ERROR LOOP
(3) 036430 104420 TRAP C$INLP
3717 036432 BNCOMplete 3$ ;NO - SKIP
(2) 036432 103005 BCC 3$
3718 036434 021415 CMP (R4),(R5) ;CHECK IF CURRENT = NEW
3719 036436 001005 BNE 4$ ;NO - SKIP
3720 036440 004737 021556 JSR PC,ONSWAP ;ELSE SWAP OLD AND NEW
3721 036444 000441 BR 9$ ;SKIP TO SEEK
3722 036446 005437 003122 3$: NEG TEMP2 ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
3723 036452 011415 4$: MOV (R4),(R5) ;MOVE CURRENT INTO OLD
3724 036454 023714 002302 CMP HLMTW,(R4) ;CHECK IF CURRENT AT 255
3725 036460 001014 BNE 7$ ;NO - SKIP
3726 036462 022737 000001 002276 CMP #1,T.DRIVE ;RLO1 OR RLO2?
3727 036470 001404 BEQ 6$ ;BRANCH IF RLO1
3728 036472 012737 177776 003122 MOV #-2,TEMP2 ;ELSE, SET UP DIFF ARGUMENT FOR RLO2
```

CZRLIBO RLO1/02 DRIVE TEST 1		MACY11 30A(1052) 17-DEC-79 13:08		G 11 PAGE 2-81		DIFFERENCE OF 1 SEEK (PART 1)		SEQ 0136
CZRLIB.MAC 12-DEC-79 14:02		*TEST 15						
3729	036500	000421				BR	8\$	
3730	036502	012737	177777	003122	6\$:	MOV	#-1,TEMP2	:AT MAX CYL, MAKE NEXT SEEK REV
3731	036510	000415				BR	8\$:SKIP
3732	036512	005714			7\$:	TST	(R4)	:TEST IF CURRENT AT 0
3733	036514	001013				BNE	8\$:NO - SKIP
3734	036516	022737	000001	002276		CMP	#1,T.DRIVE	:RLO1 OR RLO2?
3735	036524	001404				BEQ	11\$:BRANCH IF RLO1
3736	036526	012737	000002	003122		MOV	#2,TEMP2	:ELSE, SET UP DIFF ARGUMENT FOR RLO2
3737	036534	000403				BR	8\$	
3738	036536	012737	000001	003122	11\$:	MOV	#1,TEMP2	:AT CYL 0, MAKE NEXT SEEK FWRD
3739	036544	063715	003122		8\$:	ADD	TEMP2,(R5)	:ADD DIFF TO NEW CYL (+1 OR -1 FOR RLO1, :/+2 OR -2 FOR RLO2)
3740								
3741	036550	004737	020036		9\$:	JSR	PC,XSEEK	:DO SEEK
3742	036554	036730				60\$		
3743	036556	004737	017656			JSR	PC,GDRSTA	:GET DRIVE STATE
3744								
3745	036562	012703	000004			MOV	#4,R3	:SET EXPECTED STATE
3746	036566	020337	003060			CMP	R3,T.STAT	:CHECK DRIVE STATE
3747	036572	001405				BEQ	10\$:YES-SKIP
3748	036574					ERRHRD	101,,,ERR7	:REPORT STATE ERROR
(4)	036574	104456				TRAP	C\$ERRHD	
(5)	036576	000145				.WORD	101	
(5)	036600	000000				.WORD	0	
(5)	036602	013546				.WORD	ERR7	
3749	036604	000444				BR	16\$:EXIT TEST
3750	036606	012703	000005		10\$:	MOV	#5,R3	:SET EXPECTED STATE
3751	036612					TIMDLY	#50.	:WAIT 5 MS FOR DRIVE STATE CHANGE FROM 4 TO 5
(8)	036612	012746	000340			MOV	#340,-(SP)	
(7)	036616	012746	016116			MOV	#CLKINT,-(SP)	
(6)	036622	012746	000104			MOV	#104,-(SP)	
(5)	036626	012746	000003			MOV	#3,-(SP)	
(4)	036632	104437				TRAP	C\$SVEC	
(3)	036634	062706	000010			ADD	#10,SP	
3752	036674	004737	017656		12\$:	JSR	PC,GDRSTA	:GET DRIVE STATE
3753	036700	020337	003060			CMP	R3,T.STAT	:IS STATE 5?
3754	036704	001404				BEQ	16\$:YES-SKIP
3755	036706				14\$:	ERRHRD	102,,,ERR7	:REPORT STATE ERROR
(4)	036706	104456				TRAP	C\$ERRHD	
(5)	036710	000146				.WORD	102	
(5)	036712	000000				.WORD	0	
(5)	036714	013546				.WORD	ERR7	
3756	036716	012701	000062		16\$:	MOV	#50.,R1	:INITIALIZE WAIT COUNT
3757	036722	004737	022322			JSR	PC,RDYWAIT	:GO WAIT FOR DRIVE READY
3758	036726	036730				60\$		
3759	036730	012737	000002	003016	60\$:	MOV	#2,ERRSWI	:INIT ERROR SWITCH
3760	036736					ENDSUB		
(3)	036736					L10054:		
(3)	036736	104403				TRAP	C\$ESUB	
3761	036740					ESCAPE	TST	:EXIT TEST IF ERROR
(3)	036740	104410				TRAP	C\$ESCAPE	
(3)	036742	000030				.WORD	L10053-	
3762	036744	005337	003116			DEC	TEMPO	:DEC PASS COUNT
3763	036750	001410				BEQ	24\$:SKIP IF 0-DONE
3764								
3765	036752	032737	000001	003116		BIT	#BIT0,TEMPO	:TEST IF PASS=2
3766	036760	001003				BNE	23\$:NO-SKIP

CZRLIB0 RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 H 11
*TEST 15 DIFFERENCE OF 1 SEEK (PART 1) PAGE 2-82

SEQ 0137

3767 036762 004737 021516
3768 036766 036772
3769 036770 000613
3770 036772
3771 036772
3772 036772
(3) 036772
(3) 036772 104401

JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
24\$;ABORT RETURN
BR T172\$
23\$:
24\$:
T1765\$:
ENDTST
L10053:
TRAP C\$ETST

```

3774      .SBTTL *TEST 16      DIFFERENCE OF 1 SEEK (PART 2)
3775 036774      BGNTST      ;TEST 16
(3) 036774
3776 036774 012737 006643 003012      MOV      #P2T02E,ERHEAD ;SET ERROR HEADER
3777 037002 012737 000004 003116      MOV      #4,TEMP0      ;SET PASS COUNT
3778 037010 004737 016516      JSR      PC,TSTINT      ;INITIALIZE TEST
3779 037014 004737 016534      JSR      PC,GSTATR      ;GET STATUS, CLEAR DRIVE
3780 037020 037264      T1865$
3781 037022 004737 021472      JSR      PC,CHOSHD      ;GO CHOOSE HEAD
3782 037026 012737 177777 003122      MOV      #-1,TEMP2      ;SET DIFF ARGUMENT TO -1 (REVERSE)
3783 037034 012703 003102      MOV      #NEWCYL,R3      ;GET ADDRESSES
3784 037040 012704 003104      MOV      #CURCYL,R4
3785 037044 012705 003100      MOV      #OLDCYL,R5
3786 037050      T187$:
3787 037050      BGNSUB
(3) 037050
(3) 037050 104402      TRAP     C$BSUB      T16.1:
3788 037052 004737 022666      JSR      PC,GETPOS      ;GET CURRENT POSITION
3789 037056 037222      60$
3790 037060      INLOOP      ;CHECK IF IN ERROR LOOP
(3) 037060 104420      TRAP     C$INLP
3791 037062      BNCOMPLETE 3$      ;NO - SKIP
(2) 037062 103005      BCC      3$
3792 037064 021413      CMP      (R4),(R3)      ;CHECK IF CURRENT = NEW
3793 037066 001005      BNE      4$      ;NO - SKIP
3794 037070 004737 021556      JSR      PC,ONSWAP      ;ELSE SWAP OLD AND NEW
3795 037074 000421      BR       9$      ;SKIP TO SEEK
3796 037076 005437 003122      3$:      NEG      TEMP2      ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
3797 037102 011413      4$:      MOV      (R4),(R3)      ;MOV CURRENT INTO NEW
3798 037104 023714 002302      CMP      HLMTW,(R4)      ;CHECK IF CURRENT AT 255
3799 037110 001004      BNE      7$      ;NO - SKIP
3800 037112 012737 177777 003122      MOV      #-1,TEMP2      ;AT MAX CYL, MAKE NEXT SEEK REV
3801 037120 000405      BR       8$      ;SKIP
3802 037122 005714      7$:      TST      (R4)      ;TEST IF CURRENT AT 0
3803 037124 001003      BNE      8$      ;NO - SKIP
3804 037126 012737 000001 003122      MOV      #1,TEMP2      ;AT CYL 0, MAKE NEXT SEEK FWRD
3805 037134 063713 003122      8$:      ADD      TEMP2,(R3)      ;ADD DIFF TO NEW CYL (+1 OR -1)
3806 037140 004737 020036      9$:      JSR      PC,XSEEK      ;DO SEEK
3807 037144 037222      60$
3808 037146 012701 000226      MOV      #150.,R1      ;SET WAIT COUNT FOR 15 MS
3809 037152 004737 022322      JSR      PC,RDYWAIT      ;WAIT FOR READY
3810 037156 037222      60$
3811 037160 004737 022666      JSR      PC,GETPOS      ;STORE POSITION
3812 037164 037222      60$
3813 037166 011501      MOV      (R5),R1      ;GET OLD POSITION
3814 037170 161401      SUB      (R4),R1      ;SUBTRACT FROM NEW POINTER (FORWARD)
3815 037172 005737 003110      TST      DESSGN      ;CHECK IF SIGN FORWARD
3816 037176 001402      BEQ      10$      ;YES-SKIP, ELSE SUB FOR SEEK REVERSE
3817 037200 011401      MOV      (R4),R1      ;GET NEW CYLINDER
3818 037202 161501      SUB      (R5),R1      ;SUBTRACT FROM OLD CYL
3819 037204 022701 000001      10$:      CMP      #1,R1      ;CHECK IF RESULT IS DIFFERENCE OF 1
3820 037210 001404      BEQ      12$      ;YES-SKIP
3821 037212      ERRHRD 201.,ERR8      ;ELSE REPORT ERROR
(4) 037212 104456      TRAP     C$ERHRD
(5) 037214 000311      .WORD   201
(5) 037216 000000      .WORD   0

```



```

(5) 037220 013616 .WORD ERR8
3822 037222
3823 037222 012737 000002 003016 12$:
3824 037230 60$: MOV #2,ERRSWI ;INIT ERROR SWITCH
(3) 037230 ENDSUB
(3) 037230 104403 L10056:
3825 037232 TRAP C$ESUB
(3) 037232 104410 ESCAPE TST ;EXIT TEST IF ERROR
(3) 037234 000030 TRAP C$ESCAPE
3826 037236 005337 003116 .WORD L10055-
3827 037242 001410 DEC TEMPO ;DEC PASS COUNT
3828 BEQ 30$ ;EXIT IF DONE
3829 037244 032737 000001 003116 BIT #BIT0,TEMPO ;TEST IF PASS 1 OR 3
3830 037252 001003 BNE 20$ ;YES-SKIP
3831 037254 004737 021516 JSR PC,SWAPHD ;GO SWAP TO HEAD 1 OR END TEST
3832 037260 037264 30$ ;ABORT RETURN
3833 037262 000672 20$: BR T187$ ;LOOP
3834 037264 30$:
3835 037264 T1865$:
3836 037264 ENDTST
(3) 037264 L10055:
(3) 037264 104401 TRAP C$ETST
3837 037266 ENDMOD
3838
3839 .SBTTL PARAMETER CODING
3840 037266 BGNMOD HRDPRM
3841 037266 BGNHRD
(3) 037266 000030 .WORD L10057-L$HARD/2
3842
3843 037270 GPRML CNTYPE,CNT,1,YES
(4) 037270 005130 .WORD T$CODE
(4) 037272 037434 .WORD CNTYPE
(4) 037274 000001 .WORD 1
3844
3845 037276 GPRMA CSRMSG,CSR,0,160000,177776,YES
(4) 037276 000031 .WORD T$CODE
(4) 037300 037350 .WORD CSRMSG
(4) 037302 160000 .WORD T$LLOLIM
(4) 037304 177776 .WORD T$HILIM
3846
3847 037306 GPRMA VECMSG,VECT,0,0,776,YES
(4) 037306 001031 .WORD T$CODE
(4) 037310 037364 .WORD VECMSG
(4) 037312 000000 .WORD T$LLOLIM
(4) 037314 000776 .WORD T$HILIM
3848
3849 037316 GPRMD DRMSG,DRSB,0,3400,0,7,YES
(4) 037316 004032 .WORD T$CODE
(4) 037320 037426 .WORD DRMSG
(4) 037322 003400 .WORD 3400
(4) 037324 000000 .WORD T$LLOLIM
(4) 037326 000007 .WORD T$HILIM
3850
3851 037330 GPRML DRTYPE,TYPDR,1,YES
(4) 037330 003130 .WORD T$CODE
(4) 037332 037404 .WORD DRTYPE

```

3852	(4) 037334	000001								.WORD	1
3853	(4) 037336	002032				GPRMD	BRMSG,PRIOR,0,340,0,7,YES				
	(4) 037340	037373								.WORD	T\$CODE
	(4) 037342	000340								.WORD	BRMSG
	(4) 037344	000000								.WORD	340
	(4) 037346	000007								.WORD	T\$LOLIM
										.WORD	T\$HILIM
3855	(2) 037350					ENDHRD					
	(3) 037350					L10057:				.EVEN	
3856											
3857										.EVEN	
3858											
3859	037350	052502	020123	042101		CSRMSG:	.ASCIZ /BUS ADDRESS/				
	037356	051104	051505	000123							
3860											
3861	037364	042526	052103	051117		VECMMSG:	.ASCIZ /VECTOR/				
	037372	000									
3862											
3863	037373	102	020122	042514		BRMSG:	.ASCIZ /BR LEVEL/				
	037400	042526	000114								
3864											
3865	037404	051104	053111	020105		DRTYPE:	.ASCIZ /DRIVE TYPE = RL01/				
	037412	054524	042520	036440							
	037420	051040	030114	000061							
3866											
3867	037426	051104	053111	000105		DRMSG:	.ASCIZ /DRIVE/				
3868											
3869	037434	046122	030461	000		CNTYPE:	.ASCIZ /RL11/				
3870											
3871	037441					ENDMOD					
3872											
3873		037442								.EVEN	
3874											
3875	037442					BGNMOD	SFTPRM				
3876	037442					BGNSFT					
	(3) 037442	000016								.WORD	L10060-L\$SOFT/2
3877											
3878	037444					GPRML	SELQ,MISWI,4,YES				
	(4) 037444	000130								.WORD	T\$CODE
	(4) 037446	037500								.WORD	SELQ
	(4) 037450	000004								.WORD	4
3879											
3880	037452					GPRML	ALGNQ,MISWI,10,YES				
	(4) 037452	000130								.WORD	T\$CODE
	(4) 037454	037533								.WORD	ALGNQ
	(4) 037456	000010								.WORD	10
3881											
3882	037460					GPRML	MANQ,MISWI,100000,YES				
	(4) 037460	000130								.WORD	T\$CODE
	(4) 037462	037572								.WORD	MANQ
	(4) 037464	100000								.WORD	100000
3883											
3884	037466					3\$:	GPRMD	ERLIMQ,ERLIM,D,377,0,377,YES			


```

(4) 037466 004052          .WORD  T$CODE
(4) 037470 037627          .WORD  ERLIMQ
(4) 037472 000377          .WORD  377
(4) 037474 000000          .WORD  T$LOLIM
(4) 037476 000377          .WORD  T$HILIM
3885
3886 037500                ENDSFT
(2)
(3) 037500                L10060: .EVEN
3887
3888                      .EVEN
3889
3890 037500 054105 041505 052125 SELQ:  .ASCIZ /EXECUTE DRIVE SELECT TESTS/
      037506 020105 051104 053111
      037514 020105 042523 042514
      037522 052103 052040 051505
      037530 051524      000
3891
3892 037533      105 042530 052503 ALGNQ: .ASCIZ /EXECUTE HEAD ALIGNMENT SUPPORT/
      037540 042524 044040 040505
      037546 020104 046101 043511
      037554 046516 047105 020124
      037562 052523 050120 051117
      037570 000124
3893
3894 037572 047504 046440 047101 MANQ:  .ASCIZ /DO MANUAL INTERVENTION TESTS/
      037600 040525 020114 047111
      037606 042524 053122 047105
      037614 044524 047117 052040
      037622 051505 051524      000
3895
3896 037627      111 050116 052125 ERLIMQ: .ASCIZ /INPUT ERROR LIMIT/
      037634 042440 051122 051117
      037642 046040 046511 052111
      037650      000
3897
3898                      037652          .EVEN
3899
3900 037652                ENDMOD
3901
3902 037652                LASTAD
(2)
(4) 037652 000000          .EVEN
(4) 037654 000000          .WORD  0
(3) 037656                .WORD  0
3903                      L$LAST::
3904                      .EVEN
3905 037656                L$LAST::
3906
3907                      000001          .END

```


CZRLIBO RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0144

CSGMAN= 000043	8# 3484	2690	2729	2761	2778	2956	2979	3013	3030	3048	3063	3130	3166
CSGPHR= 000042	8#	1256	3095										
CSGPLO= 000030	8#												
CSGPRI= 000040	8#												
CSINIT= 000011	8#	1332											
CSINLP= 000020	8#	1443	3716	3790									
CSMANI= 000050	8#	1206	1298										
CSMEM = 000031	8#												
CSMSG = 000023	8#	919	933	947	962	977	1082	1096	1118	1132	1146		
CSOPEN= 000034	8#												
CSPNTB= 000014	8#	1020	1053	1067	1076	1139	1140	1142	2561	2562	2566	2579	2595
	2599	2603	2606	2618	2627	2628	2631	2639	2641	2642	2643		
CSPNTF= 000017	8#	1321	1322	1324	1356	1357	1359	1367	1369	1371	1445	1446	1447
	2688	2726	2759	2764	2916	2943	2954	2961	3011	3027	3045	3061	3083
	3112	3128	3164	3180	3281	3348	3432	3433	3434	3482	3503	3695	
CSPNTS= 000016	8#												
CSPNTX= 000015	8#												
CSQIO = 000377	8#												
CSRDBU= 000007	8#												
CSREFG= 000047	8#	1211	1216	1235	1239	1242							
CSRESE= 000033	8#												
CSREVI= 000003	8#	18											
CSRFLA= 000021	8#												
CSRPT = 000025	8#												
CSSEFG= 000046	8#												
CSSPRI= 000041	8#	1205	1292	1385									
CSSVEC= 000037	8#	1291	1349	1383	2801	2824	2835	2857	2882	2894	2905	2976	2992
	3004	3106	3137	3151	3185	3318	3389	3543	3751				
CSTPRI= 000013	8#												
C10MS 011252	813#												
C5SEC 011313	817#	2030	2182										
C500MS 011263	814#	1911											
DANAM 006130	662#	2641											
DATACM= 000001	92#												
DCKERR= 004000	128#	1030	1054										
DCLIM = 000012	67#												
DCLIMW 014220	1175#												
DESDIF 003106	456#	1684*	1698*	1707*	1714	1775	2603						
DESHD 003112	458#	1722	1783	1940*	1943*	1950	1952*	2283	2603	2606	2643	3294*	3329
	3331	3401	3403	3555	3557	3566	3568*	3599	3608	3610*	3651	3679	3681*
DESSEC 003114	459#	2606											
DESSGN 003110	457#	1683*	1694*	1697*	1703*	1719	1780	2603	3412	3414*	3815		
DIAGMC= 000000	8												
DIFAUG 003076	452#	1670*	1678*	1689*	1699	1707							
DIFWD 010060	751#	2603											
DIRBIT= 000004	144#	1721	1782										
DIRMSK 002314	251#	1275*	1283*										
DLTERR= 010000	126#	1034											
DLYCNT 003142	502#	1318*	1391*	1412*	1424*	1472*	1473*	1474*	1522*	1538*	1573*	1918*	2179*
	2775*	2801*	2824*	2835*	2857*	2882*	2894*	2905*	2976*	2992*	3004*	3106*	3137*
	3151*	3185*	3318*	3389*	3444*	3445*	3446*	3448*	3464*	3465*	3466*	3468*	3543*
	3751*												
DONE 003006	418#	1417*	1465	1488	1555*	1562	1628*	1640	1728*	1735	1787*	1792	2003*
	2014	3100*	3107	3141*	3144	3438*	3452						
DRDYMS= 000001	135#	1315	1365	1389	1526	1571	1905	1916	2016	2024	2166	2177	2901

CZRLIBO RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 PAGE 3-7
CROSS REFERENCE TABLE -- USER SYMBOLS

I\$CLN = 000041	8#	1381#	1397#											
I\$DU = 000041	8#	1399#	1401#											
I\$HRD = 000041	3841#	3855#												
I\$INIT= 000041	8#	1199#	1332#											
I\$MOD = 000041	8#	16#	23#	49#	174#	220#	623#	631#	850#	859#	1147#	1150#	1159#	
	1161#	1177#	1179#	1186#	1198#	1334#	1380#	1403#	1437#	2657#	2679#	3837#	3840#	
	3871#	3875#	3900#											
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#	2682	2720	2748	2931	2947	3021	3051	3071	3134	3173	3251	3274	
	3341	3360	3422	3436	3476	3496	3514	3576	3585	3623	3631	3687	3713	
	3775	3787												
I\$SETU= 000041	8#													
I\$SFT = 000041	3876#	3886#												
I\$SRV = 000041	8#	1410#	1419#	1423#	1425#	1429#	1431#							
I\$SUB = 000041	8#	2682	2720	2748	2931	2947#	2983	2995	3010#	3021	3051#	3056#	3071	
	3134#	3150	3158	3163#	3173	3251	3274	3341	3360#	3373	3383	3393	3397	
	3406	3411#	3422	3436#	3474#	3476#	3489#	3496	3514#	3527	3536	3547	3552	
	3560	3565#	3576	3585#	3602	3607#	3623	3631#	3678#	3687	3713#	3760#	3775	
	3787#	3824#												
I\$TST = 000041	8#	2682#	2715#	2720#	2744#	2748#	2753	2785	2798	2809	2816	2827	2838	
	2846	2855	2867	2872	2885	2897	2927#	2931#	2936	2947	3018#	3021#	3051	
	3067#	3071#	3076	3134	3170#	3173#	3182	3197	3207	3217	3239	3246#	3251#	
	3268#	3274#	3283	3304	3312	3321	3326	3334	3338#	3341#	3350	3360	3417#	
	3422#	3430	3436	3476	3491#	3496#	3505	3514	3571#	3576#	3585	3620#	3623#	
	3631	3684#	3687#	3697	3713	3761	3772#	3775#	3787	3825	3836#			
	245#													
JJJ	002300													
J\$JMP = 000167	8#													
LAB	014446	1230#												
LAB1	006142	664#	2642											
LAB2	006155	665#	2643											
LCLEXT	032370	3113	3169#											
LOCERR	003362	513#												
LOCYL = 040000	76#	1230												
LOE = 040000 G	51#													
LOLIM = 000002	63#													
LOLIMW	014210	1171#	1232*											
LOT = 000010 G	51#													
LPT05	030772	3040#	3058											
L\$ACP	002110 G	18#												
L\$APT	002036 G	18#												
L\$AUT	002070 G	18#												
L\$AUTO	015326 G	18	1347#											
L\$CCP	002106 G	18#												
L\$CLEA	015664 G	18	1381#											
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	014224 G	18	1181#											
L\$DLY	002116 G	18#	999	1318	1391	1477	1522	1538	1561	1573	1581	1734	1796	1907

CZRLIB0 RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 PAGE 3-12
CROSS REFERENCE TABLE -- USER SYMBOLS

PASNUM	003356	511#	1221*	1247*	1301	2683	2721	2749	2932	3023	3072	3210	3259	3425
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#												
POSHDO	022276	2141	2143#											
POSHSB	022272	2142#	3293	3328	3357	3400	3554							
POSHW1	022264	2140#	3598											
PRI =	002000 G	51#												
PRIOR =	000004	56#	3853											
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	1325	1360	1372	1448	3088			
PWCON	014772	1214	1240	1278	1288#									
PWRFLG	003366	516#	1213*	1258	1260*	1308	1393	1395*						
P2T01E	006643	688#	3698											
P2T02E	006643	689#	3776											
RDALHD	023014	2263#	3640											
RDDATA=	000114	85#												
RDHEAD=	000110	83#	2007											
RDNOHR=	000116	86#												
RDYCHK	021136	1726	1768	1892#	2001	2290								
RDYWAI	022322	2154#	3592	3638	3757	3809								
READRL	016272	1453#	1464	1485	1490	2298								
RELDWT=	040000	105#	1534	1568										
RESE3	011157	801#	1067	1140	1142	2627								
RESE4	011163	802#	1067	1140	1142	2628								
RESE5	011170	805#	2631											
RESE6	011175	806#	1076											
RESPAR	003062	445#	2616	2648	2653									
RESTAR	014464	1217	1234#											
RESTBL	002320	255#	1047											
REVSKO=	001000	100#	107											
REVSKS=	000200	98#	107											
RLBA =	000002	117#	1454	1732*	1790*	2011*	2287*							
RLBAS	003026	427#	1139	1255	1293	1322	1352	1357	1369	1418	1446	2276	2639	2688
		2726	2759	2764	2919	2954	2961	3011	3027	3098	3432			
RLCS =	000000	116#	998*	1000	1313*	1315	1353	1364*	1365	1386	1388*	1389	1483	1520
		1571	1638	1733*	1791*	2288	2292*	2294	2922*	2923	3105*	3127*	3159*	
RLCSR =	000000	122#	1453	1462	1560*	1636*	2012*	3136*	3142*	3443*	3463*			
RLDA =	000004	118#	995*	1455	1559*	1634*	1731*	1789*	2010*	2286*	3104*	3140*	3442*	3462*
RLDRV	003032	429#	997	1139	1311	1322	1357	1362	1369	1388	1446	1556	1629	1711

CZRLIBO RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 PAGE 3-13
CROSS REFERENCE TABLE -- USER SYMBOLS

	1772	2005	2279	2639	2688	2726	2759	2764	2920	2954	2961	3011	3027
	3037	3038	3044*	3045	3059*	3116	3128	3135	3427	3431	3432	3439	3460
RLMP = 000006	119#	1004	1456	2038	2039	2277							
RLVEC 003030	428#	1291	1392										
RORWOP= 020000	104#	2567	2575	2578	2604								
RPTOP 023362	912	924	938	952	967	987	1087	1102	1123	1138	2557#		
RPTREM 024356	916	930	944	959	974	1050	1093	1114	1129	2639#			
RPTRES 024150	915	929	943	958	973	1048	1092	1113	1128	2613#			
RSTRT 014402	1221#	1236											
SAMSK = 000077	139#												
SBSFIL 003372	521#												
SECWD 010077	754#	2606											
SEEK = 000106	82#	1710	1771	3461									
SEEKOP= 010000	103#	2567	2572	2601									
SELQ 037500	3878	3890#											
SEQMES 010132	758#	2561											
SETDON 014550	1233	1246	1251#										
SFTPRM 037442 G	3875#												
SGNWD 010066	752#	2603											
SIMSEK 020626	1759#	3295	3364	3518	3589	3635							
SPDERR 006426	678#	2806											
SPDSTA= 004000	168#	1530	2804										
SPTCOD 014204 G	1161#												
SSINDX 003002	414#	1210*	1510	1514*	1607*	1661	1665*	1746*	1760	1764*	1803*	1893	1897*
	1926*	1985	1989*	2052*	2155	2159*	2190*	2206	2210*	2222*	2264	2268*	2314*
	2558	2564											
STAMES 010155	759#	1053											
STAMSK= 000007	160#	1565	1643										
STATE2 011202	807#	2811											
STATE3 011212	808#	2848											
STATE5 011222	809#	2908											
STOSTA= 010000	169#												
SUBSTK 002404	287#	1512*	1513*	1663*	1664*	1762*	1763*	1895*	1896*	1987*	1988*	2157*	2158*
	2208*	2209*	2266*	2267*	2562								
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	1347	1380	1381	1399	1410	1423	1429	1437	2679	3840	3841
	3875	3876	3902#										
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	1318	1321	1322	1324	1325
	1326	1332	1349	1356	1357	1359	1360	1367	1369	1371	1372	1373	1374
	1383	1385	1391	1392	1396	1397	1401	1419	1425	1431	1443	1444	1445
	1446	1447	1448	1449	1522	1538	1542	1561	1573	1579	1581	1592	1598
	1734	1739	1744	1796	1800	1907	1912	1918	1923	2013	2020	2031	2035
	2043	2170	2173	2179	2183	2187	2301	2306	2561	2562	2566	2579	2595
	2599	2603	2606	2618	2627	2628	2631	2639	2641	2642	2643	2688	2690
	2699	2703	2707	2711	2715	2726	2729	2738	2742	2744	2753	2759	2761
	2764	2775	2778	2784	2785	2797	2798	2801	2803	2808	2809	2815	2816
	2824	2826	2827	2835	2837	2838	2845	2846	2854	2855	2857	2866	2867
	2871	2872	2882	2884	2885	2894	2896	2897	2905	2910	2916	2927	2936
	2943	2947	2954	2956	2961	2976	2979	2982	2983	2992	2994	2995	3004
	3006	3010	3011	3013	3018	3027	3030	3045	3048	3051	3056	3061	3063
	3067	3076	3083	3095	3096	3106	3112	3128	3130	3134	3137	3143	3149

	3150	3151	3157	3158	3163	3164	3166	3170	3180	3182	3185	3196	3197
	3206	3207	3216	3217	3229	3238	3239	3246	3265	3268	3281	3283	3303
	3304	3311	3312	3318	3320	3321	3325	3326	3333	3334	3335	3338	3348
	3350	3360	3372	3373	3382	3383	3389	3392	3393	3396	3397	3405	3406
	3407	3411	3417	3430	3432	3433	3434	3436	3447	3449	3467	3469	3474
	3476	3482	3484	3489	3491	3503	3505	3514	3526	3527	3535	3536	3543
	3546	3547	3551	3552	3559	3560	3561	3565	3571	3585	3601	3602	3603
	3607	3618	3620	3631	3658	3664	3678	3684	3695	3697	3713	3716	3717
	3748	3751	3755	3760	3761	3772	3787	3790	3791	3821	3824	3825	3836
	3841	3843	3845	3847	3849	3851	3853	3855	3876	3878	3880	3882	3884
	3886	3902											
SVCSUB= 000001	8#	10#	2947	3051	3134	3360	3436	3476	3514	3585	3631	3713	3787
SVCTAG= 000000	8#	13#	919	933	947	962	977	1082	1096	1118	1132	1146	1158
	1176	1332	1374	1397	1401	1419	1425	1431	2690	2715	2729	2744	2761
	2778	2927	2956	2979	3010	3013	3018	3030	3048	3056	3063	3067	3130
	3163	3166	3170	3246	3268	3338	3411	3417	3474	3484	3489	3491	3565
SVCTST= 000001	3571	3607	3620	3678	3684	3760	3772	3824	3836	3855	3886		
	8#	9#	2682	2720	2748	2931	3021	3071	3173	3251	3274	3341	3422
	3496	3576	3623	3687	3775								
SWAPHD 021516	1948#	3767	3831										
S\$LSYM= 010000	8#	919#	933#	947#	962#	977#	1082#	1096#	1118#	1132#	1146#	1158#	1176#
	1332#	1374#	1397#	1401#	1419#	1425#	1431#	2690#	2715#	2729#	2744#	2761#	2778#
	2927#	2956#	2979#	3010#	3013#	3018#	3030#	3048#	3056#	3063#	3067#	3130#	3163#
	3166#	3170#	3246#	3268#	3338#	3411#	3417#	3474#	3484#	3489#	3491#	3565#	3571#
	3607#	3620#	3678#	3684#	3760#	3772#	3824#	3836#	3855#	3886#			
TBLSTR 003024	425#												
TBT 002544	327#												
TCERR 010242	761#	1076											
TEMPO 003116	460#	3037*	3059	3611*	3613*	3615	3646	3699*	3762*	3765	3777*	3826*	3829
TEMP1 003120	461#	1657*	1659*	1729	3114*	3117	3128						
TEMP2 003122	462#	3705*	3708*	3722*	3728*	3730*	3736*	3738*	3739	3782*	3796*	3800*	3804*
	3805												
TEMP3 003124	463#	990*	1005*	1006	1051	1053							
TEMP4 003126	464#	1501	1502*	1504	1505*	1507	1508*	1518	1544	1553	1600	1611*	1981*
	1983*	1994	2045										
TEMP5 003130	465#												
TEMP6 003132	466#												
TEMP7 003134	467#												
TEMP8 003136	468#												
TOSLOW= 000001	111#												
TRPFLG 003364	515#	1348*	1354	1430*									
TRPHAN 016124 G	1349	1383	1429#										
TSTINT 016516	1495#	2693	2732	2754	2949	3034	3087	3184	3253	3290	3361	3437	3477
	3515	3586	3632	3700	3778								
TSTLAB 006363	674#	2566											
TST4 027612	2939	2948#											
TYPDR = 000006	57#	3851											
T\$ARGC= 000002	18#	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1321#	1322#	1324#	1356#	1357#
	1359#	1367#	1369#	1371#	1445#	1446#	1447#	2561#	2562#	2566#	2579#	2595#	2599#
	2603#	2606#	2618#	2627#	2628#	2631#	2639#	2641#	2642#	2643#	2688#	2726#	2759#
	2764#	2916#	2943#	2954#	2961#	3011#	3027#	3045#	3061#	3083#	3112#	3128#	3164#
	3180#	3281#	3348#	3432#	3433#	3434#	3482#	3503#	3695#				
T\$CODE= 004052	2690#	2729#	2761#	2778#	2956#	2979#	3013#	3030#	3048#	3063#	3130#	3166#	3484#
	3843#	3845#	3847#	3849#	3851#	3853#	3878#	3880#	3882#	3884#			
T\$ERRN= 000311	8#	1542#	1579#	1592#	1598#	1739#	1744#	1800#	1912#	1923#	2020#	2031#	2035#
	2043#	2173#	2183#	2187#	2301#	2306#	2699#	2703#	2707#	2711#	2738#	2742#	2784#

CZRLIBO RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 PAGE 3-15
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0157

	2797#	2803#	2808#	2815#	2826#	2837#	2845#	2854#	2866#	2871#	2884#	2896#	2910#
	2982#	2994#	3006#	3149#	3157#	3196#	3206#	3216#	3229#	3238#	3265#	3303#	3311#
	3320#	3325#	3333#	3335#	3372#	3382#	3392#	3396#	3405#	3407#	3526#	3535#	3546#
	3551#	3559#	3561#	3601#	3603#	3618#	3658#	3664#	3748#	3755#	3821#		
TSEXCP= 000000	3845#	3847#	3849#	3853#	3884#								
T\$FLAG= 000040	2753#	2785#	2798#	2809#	2816#	2827#	2838#	2846#	2855#	2867#	2872#	2885#	2897#
	2936#	2983#	2995#	3076#	3150#	3158#	3182#	3197#	3207#	3217#	3239#	3283#	3304#
	3312#	3321#	3326#	3334#	3350#	3373#	3383#	3393#	3397#	3406#	3430#	3505#	3527#
	3536#	3547#	3552#	3560#	3602#	3697#	3761#	3825#					
T\$GMAN= 000000	8#												
T\$HILI= 000377	3845#	3847#	3849#	3853#	3884#								
T\$LAST= 000001	8#	3902#											
T\$LOLI= 000000	3845#	3847#	3849#	3853#	3884#								
T\$LSYM= 010000	8#	919	933	947	962	977	1082	1096	1118	1132	1146	1158	1176
	1332	1374	1397	1401	1419	1425	1431	2715	2744	2927	3010	3018	3056
	3067	3163	3170	3246	3268	3338	3411	3417	3474	3489	3491	3565	3571
	3607	3620	3678	3684	3760	3772	3824	3836	3855	3886			
T\$LTND= 000020	3902#												
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#	1332#	1334#	1347#	1374#	1380#	1381#	1397#
	1399#	1401#	1403#	1410#	1419#	1423#	1425#	1429#	1431#	1437#	2657#	2679#	2682#
	2715#	2720#	2744#	2748#	2927#	2931#	2947#	3010#	3018#	3021#	3051#	3056#	3067#
	3071#	3134#	3163#	3170#	3173#	3246#	3251#	3268#	3274#	3338#	3341#	3360#	3411#
	3417#	3422#	3436#	3474#	3476#	3489#	3491#	3496#	3514#	3565#	3571#	3576#	3585#
	3607#	3620#	3623#	3631#	3678#	3684#	3687#	3713#	3760#	3772#	3775#	3787#	3824#
	3836#	3837#	3840#	3841#	3855#	3871#	3875#	3876#	3886#	3900#			
T\$NS0 = 000000	16#	23	49#	174	220#	623	631#	850	859#	1147	1150#	1159	1161#
	1177	1179#	1186	1189#	1193	1198#	1334	1347#	1374	1380#	1403	1410#	1419
	1423#	1425	1429#	1431	1437#	2657	2679#	3837	3840#	3871	3875#	3900	
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#	1332
	1381#	1397	1399#	1401	2682#	2715	2720#	2744	2748#	2927	2931#	3018	3021#
	3067	3071#	3170	3173#	3246	3251#	3268	3274#	3338	3341#	3417	3422#	3491
	3496#	3571	3576#	3620	3623#	3684	3687#	3772	3775#	3836	3841#	3855	3876#
	3886												
T\$NS2 = 000002	2947#	3010	3051#	3056	3134#	3163	3360#	3411	3436#	3474	3476#	3489	3514#
	3565	3585#	3607	3631#	3678	3713#	3760	3787#	3824				
T\$PTNU= 000000	8#												
T\$SAVL= 177777	8#												
T\$SEGL= 177777	8#												
T\$SUBN= 000001	8#	2682#	2720#	2748#	2931#	2947#	3021#	3051#	3071#	3134#	3173#	3251#	3274#
	3341#	3360#	3422#	3436#	3476#	3496#	3514#	3576#	3585#	3623#	3631#	3687#	3713#
	3775#	3787#											
T\$TAGL= 177777	8#												
T\$TAGN= 010061	8#	907#	921#	935#	949#	964#	979#	1084#	1098#	1120#	1133#	1151#	1162#
	1189#	1199#	1347#	1381#	1399#	1410#	1423#	1429#	2682#	2720#	2748#	2931#	2947#
	3021#	3051#	3071#	3134#	3173#	3251#	3274#	3341#	3360#	3422#	3436#	3476#	3496#
	3514#	3576#	3585#	3623#	3631#	3687#	3713#	3775#	3787#	3841#	3876#		
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#	1332#	1334#	1374#	1397#
	1401#	1403#	1419#	1425#	1431#	2657#	2690#	2715#	2729#	2744#	2753#	2761#	2778#
	2785#	2798#	2809#	2816#	2827#	2838#	2846#	2855#	2867#	2872#	2885#	2897#	2927#
	2936#	2956#	2979#	2983#	2995#	3010#	3013#	3018#	3030#	3048#	3056#	3063#	3067#
	3076#	3130#	3150#	3158#	3163#	3166#	3170#	3182#	3197#	3207#	3217#	3239#	3246#

	3268#	3283#	3304#	3312#	3321#	3326#	3334#	3338#	3350#	3373#	3383#	3393#	3397#
	3406#	3411#	3417#	3430#	3474#	3484#	3489#	3491#	3505#	3527#	3536#	3547#	3552#
	3560#	3565#	3571#	3602#	3607#	3620#	3678#	3684#	3697#	3760#	3761#	3772#	3824#
	3825#	3836#	3837#	3843#	3845#	3847#	3849#	3851#	3853#	3855#	3871#	3878#	3880#
	3882#	3884#	3886#	3900#									
T\$TEST= 000020	8#	2682#	2720#	2748#	2931#	2947	3021#	3051	3071#	3134	3173#	3251#	3274#
	3341#	3360	3422#	3436	3476	3496#	3514	3576#	3585	3623#	3631	3687#	3713
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	1321	1322	1324	1325	1326	1332	1349	1356	1357
	1359	1360	1367	1369	1371	1372	1373	1374	1383	1385	1392	1396	1397
	1401	1443	1445	1446	1447	1448	1449	1542	1579	1592	1598	1739	1744
	1800	1912	1923	2020	2031	2035	2043	2173	2183	2187	2301	2306	2561
	2562	2566	2579	2595	2599	2603	2606	2618	2627	2628	2631	2639	2641
	2642	2643	2688	2690	2699	2703	2707	2711	2715	2726	2729	2738	2742
	2744	2753	2759	2761	2764	2778	2784	2785	2797	2798	2801	2803	2808
	2809	2815	2816	2824	2826	2827	2835	2837	2838	2845	2846	2854	2855
	2857	2866	2867	2871	2872	2882	2884	2885	2894	2896	2897	2905	2910
	2916	2927	2936	2943	2947	2954	2956	2961	2976	2979	2982	2983	2992
	2994	2995	3004	3006	3010	3011	3013	3018	3027	3030	3045	3048	3051
	3056	3061	3063	3067	3076	3083	3095	3106	3112	3128	3130	3134	3137
	3149	3150	3151	3157	3158	3163	3164	3166	3170	3180	3182	3185	3196
	3197	3206	3207	3216	3217	3229	3238	3239	3246	3265	3268	3281	3283
	3303	3304	3311	3312	3318	3320	3321	3325	3326	3333	3334	3335	3338
	3348	3350	3360	3372	3373	3382	3383	3389	3392	3393	3396	3397	3405
	3406	3407	3411	3417	3430	3432	3433	3434	3436	3449	3469	3474	3476
	3482	3484	3489	3491	3503	3505	3514	3526	3527	3535	3536	3543	3546
	3547	3551	3552	3559	3560	3561	3565	3571	3585	3601	3602	3603	3607
	3618	3620	3631	3658	3664	3678	3684	3695	3697	3713	3716	3748	3751
	3755	3760	3761	3772	3787	3790	3821	3824	3825	3836			
T\$TSTS= 000001	8#	2682#	2720#	2748#	2931#	3021#	3071#	3173#	3251#	3274#	3341#	3422#	3496#
	3576#	3623#	3687#	3775#									
T\$\$AUT= 010016	1347#	1374											
T\$\$CLE= 010017	1381#	1397											
T\$\$DU = 010020	1399#	1401											
T\$\$HAR= 010057	3841#	3855											
T\$\$HW = 010012	1151#	1158											
T\$\$INI= 010015	1199#	1332											
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	3876#	3886											
T\$\$SRV= 010023	1410#	1419	1423#	1425	1429#	1431							
T\$\$SUB= 010056	2947#	2983	2995	3010	3051#	3056	3134#	3150	3158	3163	3360#	3373	3383
	3393	3397	3406	3411	3436#	3474	3476#	3489	3514#	3527	3536	3547	3552
	3560	3565	3585#	3602	3607	3631#	3678	3713#	3760	3787#	3824		
T\$\$SW = 010013	1162#	1176											
T\$\$TES= 010055	2682#	2715	2720#	2744	2748#	2753	2785	2798	2809	2816	2827	2838	2846
	2855	2867	2872	2885	2897	2927	2931#	2936	3018	3021#	3067	3071#	3076
	3170	3173#	3182	3197	3207	3217	3239	3246	3251#	3268	3274#	3283	3304
	3312	3321	3326	3334	3338	3341#	3350	3417	3422#	3430	3491	3496#	3505
	3571	3576#	3620	3623#	3684	3687#	3697	3761	3772	3775#	3825	3836	
T.BA 003046	436#	1414*	1454*	2643									
T.CS 003044	435#	991	1010	1054	1056	1413*	1453*	1468	1526	1536	1584	1590	1742
	1905	1916	1921	2016	2024	2033	2166	2177	2185	2304	2643	2868	2901

GPRML	2690# 3878	2729# 3880	2761# 3882	2778#	2956#	2979#	3013#	3030#	3048#	3063#	3130#	3166#	3484#	3843	3851
HEADER	18														
INLOOP	1443	3716	3790												
LASTAD	3902														
MANUAL	1206	1298													
MSBYTE	18#														
MSCHEC	2753# 2995# 3350#	2785# 3076# 3373#	2798# 3150# 3383#	2809# 3158# 3393#	2816# 3182# 3397#	2827# 3197# 3406#	2838# 3207# 3430#	2846# 3217# 3505#	2855# 3239# 3527#	2867# 3283# 3536#	2872# 3304# 3547#	2885# 3312# 3552#	2897# 3321# 3560#	2936# 3326# 3602#	2983# 3334# 3697#
MSCNTO	2690# 3847#	2729# 3849#	2761# 3851#	2778# 3853#	2956# 3878#	2979# 3880#	3013# 3882#	3030# 3884#	3048#	3063#	3130#	3166#	3484#	3843#	3845#
MSCOUN	1020# 1371# 2631# 3045#	1053# 1445# 2639# 3061#	1067# 1446# 2641# 3083#	1076# 1447# 2642# 3112#	1139# 2561# 2643# 3128#	1140# 2562# 2688# 3164#	1142# 2566# 2726# 3180#	1321# 2579# 2759# 3281#	1322# 2595# 2764# 3348#	1324# 2599# 2916# 3432#	1356# 2603# 2943# 3433#	1357# 2606# 2954# 3434#	1359# 2618# 2961# 3482#	1367# 2627# 3011# 3503#	1369# 2628# 3027# 3695#
MSDATA	18#	25#	27#												
MSDECR	23# 1158# 2657# 3474# 3871#	174# 1159# 2715# 3489# 3886#	623# 1176# 2744# 3491# 3900#	850# 1177# 2927# 3565#	919# 1186# 3010# 3571#	933# 1193# 3018# 3607#	947# 1332# 3056# 3620#	962# 1334# 3067# 3678#	977# 1374# 3163# 3684#	1082# 1397# 3170# 3760#	1096# 1401# 3246# 3772#	1118# 1403# 3268# 3824#	1132# 1419# 3338# 3836#	1146# 1425# 3411# 3837#	1147# 1431# 3417# 3855#
MSDEFA	2690# 3847#	2729# 3849#	2761# 3851#	2778# 3853#	2956# 3878#	2979# 3880#	3013# 3882#	3030# 3884#	3048#	3063#	3130#	3166#	3484#	3843#	3845#
MSENDE	23# 1158# 2715# 3489# 3886#	174# 1159# 2744# 3491# 3900#	623# 1176# 2927# 3565#	850# 1177# 3010# 3571#	919# 1186# 3018# 3607#	933# 1332# 3056# 3620#	947# 1334# 3067# 3678#	962# 1374# 3163# 3684#	977# 1397# 3170# 3760#	1082# 1401# 3246# 3772#	1096# 1403# 3268# 3824#	1118# 1419# 3338# 3836#	1132# 1425# 3411# 3837#	1146# 1431# 3417# 3855#	1147# 2657# 3474# 3871#
MSERRI	1542# 2187# 2837# 3216# 3407#	1579# 2301# 2845# 3229# 3526#	1592# 2306# 2854# 3238# 3535#	1598# 2699# 2866# 3265# 3546#	1739# 2703# 2871# 3303# 3551#	1744# 2707# 2884# 3311# 3559#	1800# 2711# 2896# 3320# 3561#	1912# 2738# 2910# 3325# 3601#	1923# 2742# 2982# 3333# 3603#	2020# 2784# 2994# 3335# 3618#	2031# 2797# 3006# 3372# 3658#	2035# 2803# 3149# 3382# 3664#	2043# 2808# 3157# 3392# 3748#	2173# 2815# 3196# 3396# 3755#	2183# 2826# 3206# 3405# 3821#
MSERCA	3761#	3825#													
MSERCS	3761#	3825#													
MSEXCP	3845#	3847#	3849#	3853#	3884#										
MSEXIT	2753# 2995# 3350#	2785# 3076# 3373#	2798# 3150# 3383#	2809# 3158# 3393#	2816# 3182# 3397#	2827# 3197# 3406#	2838# 3207# 3430#	2846# 3217# 3505#	2855# 3239# 3527#	2867# 3283# 3536#	2872# 3304# 3547#	2885# 3312# 3552#	2897# 3321# 3560#	2936# 3326# 3602#	2983# 3334# 3697#
MSEXSE	2753# 2995# 3350#	2785# 3076# 3373#	2798# 3150# 3383#	2809# 3158# 3393#	2816# 3182# 3397#	2827# 3197# 3406#	2838# 3207# 3430#	2846# 3217# 3505#	2855# 3239# 3527#	2867# 3283# 3536#	2872# 3304# 3547#	2885# 3312# 3552#	2897# 3321# 3560#	2936# 3326# 3602#	2983# 3334# 3697#
MSEX TJ	2753# 2995# 3350#	2785# 3076# 3373#	2798# 3150# 3383#	2809# 3158# 3393#	2816# 3182# 3397#	2827# 3197# 3406#	2838# 3207# 3430#	2846# 3217# 3505#	2855# 3239# 3527#	2867# 3283# 3536#	2872# 3304# 3547#	2885# 3312# 3552#	2897# 3321# 3560#	2936# 3326# 3602#	2983# 3334# 3697#
MSGEN	16# 962# 1158# 1399# 2744# 3051# 3338# 3571# 3824#	18# 964# 1161# 1401# 2748# 3056# 3341# 3576# 3836#	25# 977# 1162# 1410# 2761# 3063# 3360# 3585# 3840#	27# 979# 1176# 1419# 2778# 3067# 3411# 3607# 3841#	49# 1082# 1179# 1423# 2927# 3071# 3417# 3620# 3855#	220# 1084# 1181# 1425# 2931# 3130# 3422# 3623# 3875#	631# 1096# 1189# 1429# 2947# 3134# 3436# 3631# 3876#	859# 1098# 1198# 1431# 2956# 3163# 3474# 3678# 3886#	907# 1118# 1199# 1437# 2979# 3166# 3476# 3684# 3902#	919# 1120# 1199# 1437# 3010# 3170# 3484# 3687#	921# 1132# 1332# 2679# 3013# 3173# 3489# 3713#	933# 1133# 1374# 2690# 3018# 3246# 3491# 3760#	935# 1146# 1380# 2715# 3021# 3251# 3496#	947# 1150# 1381# 2720# 3030# 3268# 3514#	949# 1151# 1397# 2729# 3048# 3274# 3565# 3787#
MSGENB	2690#	2729#	2761#	2778#	2956#	2979#	3013#	3030#	3048#	3063#	3130#	3166#	3484#		

MSGETS	23#	174#	623#	850#	919#	933#	947#	962#	977#	1082#	1096#	1118#	1132#	1146#	1147#
	1158#	1159#	1176#	1177#	1186#	1193#	1332#	1334#	1374#	1397#	1401#	1403#	1419#	1425#	1431#
	2657#	2715#	2744#	2927#	3010#	3018#	3056#	3067#	3163#	3170#	3246#	3268#	3338#	3411#	3417#
	3474#	3489#	3491#	3565#	3571#	3607#	3620#	3678#	3684#	3760#	3772#	3824#	3836#	3837#	3855#
	3871#	3886#	3900#												
MSGETT	2753#	2785#	2798#	2809#	2816#	2827#	2838#	2846#	2855#	2867#	2872#	2885#	2897#	2936#	2983#
	2995#	3076#	3150#	3158#	3182#	3197#	3207#	3217#	3239#	3283#	3304#	3312#	3321#	3326#	3334#
	3350#	3373#	3383#	3393#	3397#	3406#	3430#	3505#	3527#	3536#	3547#	3552#	3560#	3602#	3697#
	3761#	3825#													
MSGNGB	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#	1347#	1380#	1381#
	1399#	1410#	1423#	1429#	1437#	2679#	3840#	3841#	3875#	3876#	3902#				
MSGNIN	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#
	1318#	1321#	1322#	1324#	1325#	1326#	1332#	1349#	1356#	1357#	1359#	1360#	1367#	1369#	1371#
	1372#	1373#	1374#	1383#	1385#	1391#	1392#	1396#	1397#	1401#	1419#	1425#	1431#	1443#	1444#
	1445#	1446#	1447#	1448#	1449#	1522#	1538#	1542#	1561#	1573#	1579#	1581#	1592#	1598#	1734#
	1739#	1744#	1796#	1800#	1907#	1912#	1918#	1923#	2013#	2020#	2031#	2035#	2043#	2170#	2173#
	2179#	2183#	2187#	2301#	2306#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#
	2628#	2631#	2639#	2641#	2642#	2643#	2688#	2690#	2699#	2703#	2707#	2711#	2715#	2726#	2729#
	2738#	2742#	2744#	2753#	2759#	2761#	2764#	2775#	2778#	2784#	2785#	2797#	2798#	2801#	2803#
	2808#	2809#	2815#	2816#	2824#	2826#	2827#	2835#	2837#	2838#	2845#	2846#	2854#	2855#	2857#
	2866#	2867#	2871#	2872#	2882#	2884#	2885#	2894#	2896#	2897#	2905#	2910#	2916#	2927#	2936#
	2943#	2947#	2954#	2956#	2961#	2976#	2979#	2982#	2983#	2992#	2994#	2995#	3004#	3006#	3010#
	3011#	3013#	3018#	3027#	3030#	3045#	3048#	3051#	3056#	3061#	3063#	3067#	3076#	3083#	3095#
	3096#	3106#	3112#	3128#	3130#	3134#	3137#	3143#	3149#	3150#	3151#	3157#	3158#	3163#	3164#
	3166#	3170#	3180#	3182#	3185#	3196#	3197#	3206#	3207#	3216#	3217#	3229#	3238#	3239#	3246#
	3265#	3268#	3281#	3283#	3303#	3304#	3311#	3312#	3318#	3320#	3321#	3325#	3326#	3333#	3334#
	3335#	3338#	3348#	3350#	3360#	3372#	3373#	3382#	3383#	3389#	3392#	3393#	3396#	3397#	3405#
	3406#	3407#	3411#	3417#	3430#	3432#	3433#	3434#	3436#	3447#	3449#	3467#	3469#	3474#	3476#
	3482#	3484#	3489#	3491#	3503#	3505#	3514#	3526#	3527#	3535#	3536#	3543#	3546#	3547#	3551#
	3552#	3559#	3560#	3561#	3565#	3571#	3585#	3601#	3602#	3603#	3607#	3618#	3620#	3631#	3658#
	3664#	3678#	3684#	3695#	3697#	3713#	3716#	3717#	3748#	3751#	3755#	3760#	3761#	3772#	3787#
	3790#	3791#	3821#	3824#	3825#	3836#	3841#	3843#	3845#	3847#	3849#	3851#	3853#	3855#	3876#
	3878#	3880#	3882#	3884#	3886#	3902#									
MSGNLS	2690#	2729#	2761#	2778#	2956#	2979#	3013#	3030#	3048#	3063#	3130#	3166#	3484#		
MSGNSU	2947#	3051#	3134#	3360#	3436#	3476#	3514#	3585#	3631#	3713#	3787#				
MSGNTA	919#	933#	947#	962#	977#	1082#	1096#	1118#	1132#	1146#	1158#	1176#	1332#	1374#	1397#
	1401#	1419#	1425#	1431#	2715#	2744#	2927#	3010#	3018#	3056#	3067#	3163#	3170#	3246#	3268#
	3338#	3411#	3417#	3474#	3489#	3491#	3565#	3571#	3607#	3620#	3678#	3684#	3760#	3772#	3824#
	3836#	3855#	3886#												
MSGNTE	2682#	2720#	2748#	2931#	3021#	3071#	3173#	3251#	3274#	3341#	3422#	3496#	3576#	3623#	3687#
	3775#														
MSHAPT	18#														
MSHNAP	18#														
MSINCR	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#	1321#	1322#	1324#	1325#	1326#	1332#	1347#	1349#
	1356#	1357#	1359#	1360#	1367#	1369#	1371#	1372#	1373#	1374#	1380#	1381#	1383#	1385#	1392#
	1396#	1397#	1399#	1401#	1410#	1423#	1429#	1437#	1443#	1445#	1446#	1447#	1448#	1449#	1542#
	1579#	1592#	1598#	1739#	1744#	1800#	1912#	1923#	2020#	2031#	2035#	2043#	2173#	2183#	2187#
	2301#	2306#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#	2628#	2631#	2639#
	2641#	2642#	2643#	2679#	2682#	2688#	2690#	2699#	2703#	2707#	2711#	2715#	2720#	2726#	2729#
	2738#	2742#	2744#	2748#	2753#	2759#	2761#	2764#	2778#	2784#	2785#	2797#	2798#	2801#	2803#

	2808#	2809#	2815#	2816#	2824#	2826#	2827#	2835#	2837#	2838#	2845#	2846#	2854#	2855#	2857#
	2866#	2867#	2871#	2872#	2882#	2884#	2885#	2894#	2896#	2897#	2905#	2910#	2916#	2927#	2931#
	2936#	2943#	2947#	2954#	2956#	2961#	2976#	2979#	2982#	2983#	2992#	2994#	2995#	3004#	3006#
	3010#	3011#	3013#	3018#	3021#	3027#	3030#	3045#	3048#	3051#	3056#	3061#	3063#	3067#	3071#
	3076#	3083#	3095#	3106#	3112#	3128#	3130#	3134#	3137#	3149#	3150#	3151#	3157#	3158#	3163#
	3164#	3166#	3170#	3173#	3180#	3182#	3185#	3196#	3197#	3206#	3207#	3216#	3217#	3229#	3238#
	3239#	3246#	3251#	3265#	3268#	3274#	3281#	3283#	3303#	3304#	3311#	3312#	3318#	3320#	3321#
	3325#	3326#	3333#	3334#	3335#	3338#	3341#	3348#	3350#	3360#	3372#	3373#	3382#	3383#	3389#
	3392#	3393#	3396#	3397#	3405#	3406#	3407#	3411#	3417#	3422#	3430#	3432#	3433#	3434#	3436#
	3449#	3469#	3474#	3476#	3482#	3484#	3489#	3491#	3496#	3503#	3505#	3514#	3526#	3527#	3535#
	3536#	3543#	3546#	3547#	3551#	3552#	3559#	3560#	3561#	3565#	3571#	3576#	3585#	3601#	3602#
	3603#	3607#	3618#	3620#	3623#	3631#	3658#	3664#	3678#	3684#	3687#	3695#	3697#	3713#	3716#
	3748#	3751#	3755#	3760#	3761#	3772#	3775#	3787#	3790#	3821#	3824#	3825#	3836#	3840#	3841#
	3875#	3876#													
M\$LDRO	1202#	1205#	1211#	1216#	1235#	1239#	1242#	1256#	1292#	1325#	1360#	1372#	1373#	1385#	1392#
	1396#	1448#	3095#												
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#	1332#	1334#	1374#	1397#	1401#	1403#	1419#	1425#	1431#
	2657#	2715#	2744#	2927#	3010#	3018#	3056#	3067#	3163#	3170#	3246#	3268#	3338#	3411#	3417#
	3474#	3489#	3491#	3565#	3571#	3607#	3620#	3678#	3684#	3760#	3772#	3824#	3836#	3837#	3855#
	3871#	3886#	3900#												
M\$PRIN	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1321#	1322#	1324#	1356#	1357#	1359#	1367#	1369#
	1371#	1445#	1446#	1447#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#	2628#
	2631#	2639#	2641#	2642#	2643#	2688#	2726#	2759#	2764#	2916#	2943#	2954#	2961#	3011#	3027#
	3045#	3061#	3083#	3112#	3128#	3164#	3180#	3281#	3348#	3432#	3433#	3434#	3482#	3503#	3695#
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#	1347#	1380#	1381#	1399#	1410#	1423#	1429#
	1437#	2679#	2682#	2720#	2748#	2931#	2947#	3021#	3051#	3071#	3134#	3173#	3251#	3274#	3341#
	3360#	3422#	3436#	3476#	3496#	3514#	3576#	3585#	3623#	3631#	3687#	3713#	3775#	3787#	3840#
	3841#	3875#	3876#												
M\$PUT	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1291#	1321#	1322#	1324#	1349#	1356#	1357#	1359#
	1367#	1369#	1371#	1383#	1445#	1446#	1447#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#
	2618#	2627#	2628#	2631#	2639#	2641#	2642#	2643#	2688#	2726#	2759#	2764#	2801#	2824#	2835#
	2857#	2882#	2894#	2905#	2916#	2943#	2954#	2961#	2976#	2992#	3004#	3011#	3027#	3045#	3061#
	3083#	3106#	3112#	3128#	3137#	3151#	3164#	3180#	3185#	3281#	3318#	3348#	3389#	3432#	3433#
	3434#	3482#	3503#	3543#	3695#	3751#									
M\$PUT1	1020#	1053#	1067#	1076#	1139#	1140#	1142#	1291#	1321#	1322#	1324#	1349#	1356#	1357#	1359#
	1367#	1369#	1371#	1383#	1445#	1446#	1447#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#
	2618#	2627#	2628#	2631#	2639#	2641#	2642#	2643#	2688#	2726#	2759#	2764#	2801#	2824#	2835#
	2857#	2882#	2894#	2905#	2916#	2943#	2954#	2961#	2976#	2992#	3004#	3011#	3027#	3045#	3061#
	3083#	3106#	3112#	3128#	3137#	3151#	3164#	3180#	3185#	3281#	3318#	3348#	3389#	3432#	3433#
	3434#	3482#	3503#	3543#	3695#	3751#									
M\$RADI	2690#	2729#	2761#	2778#	2956#	2979#	3013#	3030#	3048#	3063#	3130#	3166#	3484#	3843#	3845#
	3847#	3849#	3851#	3853#	3878#	3880#	3882#	3884#							
M\$RNRO	1202#	1256#	3095#												
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#	1347#	1380#	1381#	1399#	1410#	1423#	1429#
	1437#	2679#	2682#	2720#	2748#	2931#	2947#	3021#	3051#	3071#	3134#	3173#	3251#	3274#	3341#
	3360#	3422#	3436#	3476#	3496#	3514#	3576#	3585#	3623#	3631#	3687#	3713#	3775#	3787#	3840#
	3841#	3875#	3876#												
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#	1321#
	1322#	1324#	1325#	1326#	1332#	1349#	1356#	1357#	1359#	1360#	1367#	1369#	1371#	1372#	1373#
	1374#	1383#	1385#	1392#	1396#	1397#	1401#	1443#	1445#	1446#	1447#	1448#	1449#	1542	1579

	1592	1598	1739	1744	1800	1912	1923	2020	2031	2035	2043	2173	2183	2187	2301
	2306	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#	2628#	2631#	2639#	2641#
	2642#	2643#	2688#	2690#	2699	2703	2707	2711	2715#	2726#	2729#	2738	2742	2744#	2753#
	2759#	2761#	2764#	2778#	2784	2785#	2797	2798#	2801#	2803	2808	2809#	2815	2816#	2824#
	2826	2827#	2835#	2837	2838#	2845	2846#	2854	2855#	2857#	2866	2867#	2871	2872#	2882#
	2884	2885#	2894#	2896	2897#	2905#	2910	2916#	2927#	2936#	2943#	2947#	2954#	2956#	2961#
	2976#	2979#	2982	2983#	2992#	2994	2995#	3004#	3006	3010#	3011#	3013#	3018#	3027#	3030#
	3045#	3048#	3051#	3056#	3061#	3063#	3067#	3076#	3083#	3095#	3106#	3112#	3128#	3130#	3134#
	3137#	3149	3150#	3151#	3157	3158#	3163#	3164#	3166#	3170#	3180#	3182#	3185#	3196	3197#
	3206	3207#	3216	3217#	3229	3238	3239#	3246#	3265	3268#	3281#	3283#	3303	3304#	3311
	3312#	3318#	3320	3321#	3325	3326#	3333	3334#	3335	3338#	3348#	3350#	3360#	3372	3373#
	3382	3383#	3389#	3392	3393#	3396	3397#	3405	3406#	3407	3411#	3417#	3430#	3432#	3433#
	3434#	3436#	3449#	3469#	3474#	3476#	3482#	3484#	3489#	3491#	3503#	3505#	3514#	3526	3527#
	3535	3536#	3543#	3546	3547#	3551	3552#	3559	3560#	3561	3565#	3571#	3585#	3601	3602#
	3603	3607#	3618	3620#	3631#	3658	3664	3678#	3684#	3695#	3697#	3713#	3716#	3748	3751#
	3755	3760#	3761#	3772#	3787#	3790#	3821	3824#	3825#	3836#					
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#	1321#
	1322#	1324#	1325#	1326#	1332#	1349#	1356#	1357#	1359#	1360#	1367#	1369#	1371#	1372#	1373#
	1374#	1383#	1385#	1392#	1396#	1397#	1401#	1443#	1445#	1446#	1447#	1448#	1449#	1542#	1579#
	1592#	1598#	1739#	1744#	1800#	1912#	1923#	2020#	2031#	2035#	2043#	2173#	2183#	2187#	2301#
	2306#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#	2628#	2631#	2639#	2641#
	2642#	2643#	2688#	2690#	2699#	2703#	2707#	2711#	2715#	2726#	2729#	2738#	2742#	2744#	2753#
	2759#	2761#	2764#	2778#	2784#	2785#	2797#	2798#	2801#	2803#	2808#	2809#	2815#	2816#	2824#
	2826#	2827#	2835#	2837#	2838#	2845#	2846#	2854#	2855#	2857#	2866#	2867#	2871#	2872#	2882#
	2884#	2885#	2894#	2896#	2897#	2905#	2910#	2916#	2927#	2936#	2943#	2947#	2954#	2956#	2961#
	2976#	2979#	2982#	2983#	2992#	2994#	2995#	3004#	3006#	3010#	3011#	3013#	3018#	3027#	3030#
	3045#	3048#	3051#	3056#	3061#	3063#	3067#	3076#	3083#	3095#	3106#	3112#	3128#	3130#	3134#
	3137#	3149#	3150#	3151#	3157#	3158#	3163#	3164#	3166#	3170#	3180#	3182#	3185#	3196#	3197#
	3206#	3207#	3216#	3217#	3229#	3238#	3239#	3246#	3265#	3268#	3281#	3283#	3303#	3304#	3311#
	3312#	3318#	3320#	3321#	3325#	3326#	3333#	3334#	3335#	3338#	3348#	3350#	3360#	3372#	3373#
	3382#	3383#	3389#	3392#	3393#	3396#	3397#	3405#	3406#	3407#	3411#	3417#	3430#	3432#	3433#
	3434#	3436#	3449#	3469#	3474#	3476#	3482#	3484#	3489#	3491#	3503#	3505#	3514#	3526#	3527#
	3535#	3536#	3543#	3546#	3547#	3551#	3552#	3559#	3560#	3561#	3565#	3571#	3585#	3601#	3602#
	3603#	3607#	3618#	3620#	3631#	3658#	3664#	3678#	3684#	3695#	3697#	3713#	3716#	3748#	3751#
	3755#	3760#	3761#	3772#	3787#	3790#	3821#	3824#	3825#	3836#					
MSTSTL	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#	1321#
	1322#	1324#	1325#	1326#	1332#	1349#	1356#	1357#	1359#	1360#	1367#	1369#	1371#	1372#	1373#
	1374#	1383#	1385#	1392#	1396#	1397#	1401#	1443#	1445#	1446#	1447#	1448#	1449#	1542#	1579#
	1592#	1598#	1739#	1744#	1800#	1912#	1923#	2020#	2031#	2035#	2043#	2173#	2183#	2187#	2301#
	2306#	2561#	2562#	2566#	2579#	2595#	2599#	2603#	2606#	2618#	2627#	2628#	2631#	2639#	2641#
	2642#	2643#	2688#	2690#	2699#	2703#	2707#	2711#	2715#	2726#	2729#	2738#	2742#	2744#	2753#
	2759#	2761#	2764#	2778#	2784#	2785#	2797#	2798#	2801#	2803#	2808#	2809#	2815#	2816#	2824#
	2826#	2827#	2835#	2837#	2838#	2845#	2846#	2854#	2855#	2857#	2866#	2867#	2871#	2872#	2882#
	2884#	2885#	2894#	2896#	2897#	2905#	2910#	2916#	2927#	2936#	2943#	2947#	2954#	2956#	2961#
	2976#	2979#	2982#	2983#	2992#	2994#	2995#	3004#	3006#	3010#	3011#	3013#	3018#	3027#	3030#
	3045#	3048#	3051#	3056#	3061#	3063#	3067#	3076#	3083#	3095#	3106#	3112#	3128#	3130#	3134#
	3137#	3149#	3150#	3151#	3157#	3158#	3163#	3164#	3166#	3170#	3180#	3182#	3185#	3196#	3197#
	3206#	3207#	3216#	3217#	3229#	3238#	3239#	3246#	3265#	3268#	3281#	3283#	3303#	3304#	3311#
	3312#	3318#	3320#	3321#	3325#	3326#	3333#	3334#	3335#	3338#	3348#	3350#	3360#	3372#	3373#
	3382#	3383#	3389#	3392#	3393#	3396#	3397#	3405#	3406#	3407#	3411#	3417#	3430#	3432#	3433#
	3434#	3436#	3449#	3469#	3474#	3476#	3482#	3484#	3489#	3491#	3503#	3505#	3514#	3526#	3527#
	3535#	3536#	3543#	3546#	3547#	3551#	3552#	3559#	3560#	3561#	3565#	3571#	3585#	3601#	3602#
	3603#	3607#	3618#	3620#	3631#	3658#	3664#	3678#	3684#	3695#	3697#	3713#	3716#	3748#	3751#
	3755#	3760#	3761#	3772#	3787#	3790#	3821#	3824#	3825#	3836#					

CZRLIB RL01/02 DRIVE TEST 1
CZRLIB.MAC 12-DEC-79 14:02

MACY11 30A(1052) 17-DEC-79 13:08 PAGE 4-5
CROSS REFERENCE TABLE -- MACRO NAMES

MSWORD	18#	1181#	1542#	1579#	1592#	1598#	1739#	1744#	1800#	1912#	1923#	2020#	2031#	2035#	2043#
	2173#	2183#	2187#	2301#	2306#	2690#	2699#	2703#	2707#	2711#	2729#	2738#	2742#	2753#	2761#
	2778#	2784#	2785#	2797#	2798#	2803#	2808#	2809#	2815#	2816#	2826#	2827#	2837#	2838#	2845#
	2846#	2854#	2855#	2866#	2867#	2871#	2872#	2884#	2885#	2896#	2897#	2910#	2936#	2956#	2979#
	2982#	2983#	2994#	2995#	3006#	3013#	3030#	3048#	3063#	3076#	3130#	3149#	3150#	3157#	3158#
	3166#	3182#	3196#	3197#	3206#	3207#	3216#	3217#	3229#	3238#	3239#	3265#	3283#	3303#	3304#
	3311#	3312#	3320#	3321#	3325#	3326#	3333#	3334#	3335#	3350#	3372#	3373#	3382#	3383#	3392#
	3393#	3396#	3397#	3405#	3406#	3407#	3430#	3484#	3505#	3526#	3527#	3535#	3536#	3546#	3547#
	3551#	3552#	3559#	3560#	3561#	3601#	3602#	3603#	3618#	3658#	3664#	3697#	3748#	3755#	3821#
	3843#	3845#	3847#	3849#	3851#	3853#	3878#	3880#	3882#	3884#	3902				
POINTE	14														
PRINTB	1020	1053	1067	1076	1139	1140	1142	2561	2562	2566	2579	2595	2599	2603	2606
	2618	2627	2628	2631	2639	2641	2642	2643							
PRINTF	1321	1322	1324	1356	1357	1359	1367	1369	1371	1445	1446	1447	2688	2726	2759
	2764	2916	2943	2954	2961	3011	3027	3045	3061	3083	3112	3128	3164	3180	3281
	3348	3432	3433	3434	3482	3503	3695								
READEF	1211	1216	1235	1239	1242										
SETPRI	1205	1292	1385												
SETVEC	1291	1349	1383	2801	2824	2835	2857	2882	2894	2905	2976	2992	3004	3106	3137
	3151	3185	3318	3389	3543	3751									
SVC	6#	8													
TIMDLY	202#	2801	2824	2835	2857	2882	2894	2905	2976	2992	3004	3106	3137	3151	3185
	3318	3389	3543	3751											
WAITMS	184#	1318	1391	1522	1538	1573	1918	2179	2775						
WAITUS	196#	999	1561	1581	1734	1796	1907	2013	2170	3143					
XFER	2753#	2785#	2798#	2809#	2816#	2827#	2838#	2846#	2855#	2867#	2872#	2885#	2897#	2936#	2983#
	2995#	3076#	3150#	3158#	3182#	3197#	3207#	3217#	3239#	3283#	3304#	3312#	3321#	3326#	3334#
	3350#	3373#	3383#	3393#	3397#	3406#	3430#	3505#	3527#	3536#	3547#	3552#	3560#	3602#	3697#

. ABS. 037656 000

ERRORS DETECTED: 0

.CZRLIB.LST/CRF=SVC33/ML.CZRLIB.MAC
RUN-TIME: 144 147 14 SECONDS
RUN-TIME RATIO: 596/307=1.9
CORE USED: 17K (33 PAGES)