

RL11,RLV11

RL01,02 DRIVE CPT
CZRLLA0

AH-F130A-MC

COPYRIGHT © 1979
FICHE 1 OF 1

MAY 1979

digital

MADE IN USA

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-F131A-MC
PRODUCT NAME: CZRLLAO RL01/02 DRIVE COMPATABILITY TEST
DATE CREATED: 5-JAN-79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: 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
1.2	SYSTEM 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 SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	HOW TO CREATE A CHAINABLE FILE
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
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION
-----1.1 PROGRAM ABSTRACT
-----1.1.1 STRUCTURE OF PROGRAM

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

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

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

1.1.2 DIAGNOSTIC INFORMATION

THE RLO1 DRIVE COMPATABILITY TEST IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST INTERCHANGABILITY OF CARTRIDGES BETWEEN DRIVES. THE TEST PERFORMS WRITES, READS, OVERWRITES, ADJACENT CYLINDER WRITES TO PROVE COMPATABILITY.

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 RLO1 DRIVES WITH RLO1K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
 - 1 - 8 RLO2 DRIVES WITH RLO2K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'

- KW11P, KW11L (OPTIONAL)
- LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLLAO RLO1/02 DRIVE COMPATABILITY
(FORMERLY CZRLF8)

1.3 RELATED DOCUMENTS AND STANDARDS

RLO1 USERS MANUAL (EK-RLO1-UG-PRE)
XXDP USERS MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RLO1/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAAO	RLV11 RLO1/02 DISKLESS TEST (RLV11 ONLY)
CZRLGAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 1)
CZRLHAO	RL11/RLV11 RLO1/02 CONTROLLER TEST (PART 2)
CZRLIAO	RLO1/02 DRIVE TEST (PART 1)
CZRLJAO	RLO1/02 DRIVE TEST (PART 2)
CZRLKAO	RL11/RLV11 RLO1/02 PERFORMANCE EXERCISER

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RLO1/02 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 SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

* STEP 1 *

A SHORT SERIES OF "HARDCORE QUESTIONS" WILL BE ASKED:

QUESTION -----	MEANING -----
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
SOMZ (L) N ?	IS THE POWER 50 CYCLES (EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?
LPT (L) N ?	IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?	HOW MANY K OF MEMORY ARE THERE?

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

IF NEITHER P OR L CLOCK ARE ANSWERED YES THE OPERATOR WILL BE ASKED TO TYPE TWO CHARACTERS 4 SECONDS APART.

* STEP 2 *

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT 'DS-B>'. 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 SOME-

THING LIKE THIS:

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-B>" 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:

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 3 *

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 4 *

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 5 *

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 6 *

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 DS-B>).

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 OCCURED.

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 2, 3, 4, 5, AND 6 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

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 CZRLLA	O
CZRL	D
L-CLK (L) N ? Y	D,O
50HZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D
MEM (K) (D) 16 ?	D
DS-B>STA/PASS:1/FLAGS:HOE	D,O
# UNITS (D) ? 2	D,O
UNIT 1	D
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ?	D,O
DRIVE TYPE = RL01 (L) Y ?	D,O
UNIT 2	D
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)
CHANGE SW (L) ? N	D,O
CZRL HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DS-B>PRO/FLAGS:IER:LOE:HOE=0	D,O
***** AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT *****	
^C	O
DS-B>CON/FLAGS:HOE:IER:LOE=0	D,O

```

CHANGE SW (L) ? N           D.O
CZRL EOP 1                   D
^C
DS-B>RESTART/PASS:1         D.O
CHANGE SW (L) ? N           D.O
-----
-----
-----
-----

```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION 'BIN' INSTEAD OF 'BIC'. THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND 'CCI' ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>). THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION 'BIC'.

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```

.R UPD2
RESTART: XXXXXX
*CLR
*LOAD DIAG.BIN
XFER:200 CORE:0,60602
*START 200
L-CLK (L) N ?
-----
-----
-----

```

DS-B>CCI
UNITS (D) ? 4

(CHANGE SW (L) ? N
PTAB END: 60632

AT THIS POINT THE MACHINE HALTS AND
YOU MUST RESTART AT ADDRESS XXXXXX

*HICORE 60632
CORE: 0,60632
*DUMP DK0: DIAG.BIC

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN
FILE CONTAINING THE XXDP COMMAND

.R DIAG.BIC

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING
THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

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
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS
4. AN ERROR WAS ENCOUNTERED WITH THE MOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS

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 () 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, SUB-TEST, 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

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

"EOP-INCR" IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART
2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

 CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".R DIAG". THE COMMAND PROMPT "DS-B>" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

 DRO(P)/UNITS:UNIT-LIST

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

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

ADD/UNITS:UNIT-LIST

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

PRI(NT)

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

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

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

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

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

UNITS (D) ? 8

UNIT 1

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ?

UNIT 5

BUS ADDRESS (O) 174400 ? 175400

VECTOR (O) 160 ? 164

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ? N

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 RLO2 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 RLO2 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RLO2 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RLO2'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.

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

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.

THERE ARE NO SOFTWARE PARAMETERS.

3.0 ERROR INFORMATION

ERROR INFORMATION IS COMPLETE IN GIVING ALL INFORMATION NECESSARY. ALL REGISTERS ARE GIVEN AS WELL AS TRACK, SECTOR AND DRIVES INVOLVED IN ERROR.

3.1 ERROR REPORTING

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DEVICE. ERROR REPORTS ARE AIMED AT BEING SELF EXPLANATORY. THE GENERAL FORMAT IS:

DZRLX XXX ERR YYYYY TST ZZZ SUB PPP PC: RRRRRR

WHERE:

? IS PROGRAM LETTER
XXX IS SFT - SOFT ERROR
 HRD - HARD ERROR
 DV FAT - DEVICE FATAL ERROR
 SYS FAT - SYSTEM FATAL ERROR
YYYYY IS THE ERROR NUMBER
ZZZ IS THE TEST NUMBER
PPP IS THE SUBTEST NUMBER
RRRRR IS THE PROGRAM LISTING LOCATION

ERRORS GIVE THE REGISTER CONTENTS BEFORE AND AFTER THE ERROR ALONG WITH A ONE LINE DESCRIPTION AND RELEVANT DATA.

EXAMPLE:

ONE LINE DESCRIPTION
(OPTIONAL SECOND LINE)
(OPTIONAL THIRD LINE)
BEFORE CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX
AFTER CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX
OTHER PERTINENT INFORMATION IS GIVEN AT THIS TIME.

REGISTER DESCRIPTIONS CAN BE FOUND IN SECTION 5.0.

ERROR DESCRIPTIONS:

"ERROR READING SECTOR"

ERROR WAS ENCOUNTERED WHILE TRYING TO READ VERIFY THE SECTOR AFTER IT WAS WRITTEN BY THE SAME DRIVE.

"MINIMUM OF TWO DRIVES REQUIRED"

THE PROGRAM REQUIRES AT LEAST TWO DRIVES TO PROVE COMPATIBILITY.

"MAXIMUM OF FOUR DRIVES ALLOWED"

THE PROGRAM ONLY ALLOWS A MAXIMUM OF FOUR DRIVES.

"CAN'T FIND FIVE ADJACENT TRACKS"

THE PROGRAM REQUIRES TEN SETS OF FIVE ADJACENT TRACKS AT PREDETERMINED SPOTS ACROSS THE PACK. IT WAS UNABLE TO FIND FIVE COMPLETELY GOOD ADJACENT TRACKS IN THE LIMITS GIVEN.

"ERROR WRITING SECTOR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO WRITE THE GIVEN SECTOR.

"OVERWRITE ERROR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO READ DATA AFTER AN OVERWRITE BY ONE DRIVE. BOTH DRIVES INVOLVED ARE GIVEN.

"READ RECOVERY ERROR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO RECOVER ANOTHER DRIVES DATA.

"ADJACENT TRACK TEST"

AN ERROR WAS ENCOUNTERED WHILE IN THE ADJACENT TEST PART, A FURTHER DESCRIPTION IS GIVEN.

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 (XXXIX4)
-----FOR READ/WRITE FUNCTIONS

BIT 15-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (1-40.)

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)
BIT 9 - VOLUME CHECK (VC)
BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS 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

THE FOLLOWING IS A BRIEF DESCRIPTION OF THE WAY THE PROGRAM EXECUTES. THE PROGRAM WILL CHECK COMPATIBILITY BETWEEN 2 - 4 DRIVES USING THE SAME RLO1K CARTRIDGE OR SAME RLO2K CARTRIDGE. THE PROGRAM WILL ASK THE OPERATOR TO SEQUENCE THE PACK BETWEEN THE DRIVES GIVEN IN THE FOLLOWING MANNER.

PLACE PACK IN DRIVE N ON CONTROLLER X AND LOAD
 UNLOAD DRIVE N ON CONTROLLER X
 PLACE PACK IN DRIVE N+1 ON CONTROLLER X AND LOAD
 UNLOAD DRIVE N+1 ON CONTROLLER X
 ETC.....

THE PROGRAM WILL SEQUENCE IN THE ORDER THAT WAS GIVEN IN THE HARDWARE QUESTIONS. I.E.

DRIVE ? 0,1,2,3
 PROGRAM WILL SEQUENCE 0,1,2,3,2,1,0
 DRIVE ? 1,0,3,2
 PROGRAM WILL SEQUENCE 1,0,3,2,3,0,1

WHEN THE FIRST DRIVE IS LOADED THE PROGRAM WILL ATTEMPT TO FIND TEN SETS OF FIVE ADJACENT TRACKS AT PREDETERMINED SPOTS THAT CONTAIN NO BAD SECTORS USING THE BAD SECTOR FILE. THE 10 SPOTS ARE: ON BOTH SURFACES, INNER, OUTER, MIDDLE, ONE QUARTER AND THREE QUARTERS. AFTER THIS IS DONE THE OVERWRITE TEST IS PREPARED (FIRST DRIVE CAN'T OVERWRITE) AS WELL AS THE ADJACENT TEST.

AS THE PACK IS CYCLED BETWEEN DRIVES THE FOLLOWING CHECKS ARE MADE:

EACH DRIVE CAN OVERWRITE EACH OTHER DRIVE
 EACH DRIVE CAN RECOVER EACH OTHERS DATA
 EACH DRIVE CAN WRITE ADJACENT TO EVERY OTHER

DRIVE WITHOUT DISTURBING THE OTHER'S DATA.

READS AND WRITES TAKE PLACE AFTER SEEKS FROM
BOTH DIRECTIONS.

ADJACENT WRITES TAKE PLACE TO BOTH SIDES OF EACH WRITE

TESTS ARE PERFORMED AT ALL TEN SPOTS ACROSS THE PACK.

1-	22	GLOBAL EQUATES SECTION
1-	80	GLOBAL DATA SECTION
4-	1	GLOBAL TEXT SECTION
4-	41	GLOBAL ERROR REPORT SECTION
5-	1	INITIALIZATION SECTION
13-	1	GLOBAL SUBROUTINES SECTION
32-	1	DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

1		.TITLE	CZRLLAO RL01/2 DRIVE CPT
2		.ENABLE	AMA
3	000000	.ENABLE	ABS
4		.NLIST	ME,CND,MD
5		.MCALL	SVC
6			
7	002000	.-2000	
8			
9	002000	SVC	
10	000000	SVCINS=0	
11	000000	SVCTAG=0	
12			
13	002000	POINTER	NONE
14			
15	002000	BGNMOD	MDHEDR
16	002000	HEADER	CZRLLAO,0,0,0,0,RL01
	002000	.ASCII	/C/
	002001	.ASCII	/Z/
	002002	.ASCII	/R/
	002003	.ASCII	/L/
	002004	.ASCII	/L/
	002005	.BYTE	0
	002006	.BYTE	0
	002007	.BYTE	0
	002010	.ASCII	/A/
	002011	.ASCII	/O/
	002012	.WORD	0
	002014	.WORD	0
	002016	.WORD	L\$HARD
	002020	.WORD	0
	002022	.WORD	L\$HW
	002024	.WORD	0
	002026	.WORD	L\$LAST
	002030	.WORD	0
	002032	.WORD	0
	002034	.WORD	0
	002036	.WORD	0
	002040	.WORD	L\$DISPATCH
	002042	.WORD	0
	002044	.WORD	0
	002046	.WORD	0
	002050	.BYTE	C\$REVISION
	002051	.BYTE	C\$EDIT
	002052	.WORD	0
	002054	.WORD	0
	002056	.WORD	0
	002060	.WORD	0
	002062	.WORD	0
	002064	.WORD	L\$DVTYP
	002066	.WORD	0
	002070	.WORD	L\$DR
	002072	.WORD	L\$DRST
	002074	.WORD	0
	002076	.WORD	0
	002100	.WORD	14
	002102	.WORD	0
	002104	.WORD	L\$IN!!

```

002106 023562          .WORD  L$CLEAN
17 002110
18 002110          DEVREG
   002110 000000      .WORD  0
   .BLKW
19 002114          DEVTYP <RLO1,RLO2>
   002114 122 114 060      .ASCIZ /RLO1,RLO2/
   002117 061 054 122
   002122 114 060 062
   002125 000

          .EVEN

20
21
22          .SBTTL GLOBAL EQUATES SECTION
23
24          :DEFINITIONS
25
26
27 002126          BGNMOD  GLBEQAT
28
29 002126          EQUALS
30
31          000000      CS=0          ;CONTROL AND STATUS OFFSET
32          000002      BA=2          ;BUSS ADDRESS OFFSET
33          000004      DA=4          ;DISK ADDRESS OFFSET
34          000006      MP=6          ;MULTI PURPOSE OFFSET
35
36          ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
37
38          000000      CSR=0         ;CONTROLLER ADDRESS
39          000002      VEC=2         ;VECTOR OF CONTROLLER
40          000004      DSB=4         ;DRIVE SELECT
41          000006      PAT=6         ;PATTERN UNIQUE TO DRIVE
42
43
44          000001      DRDY=BIT0     ;DRIVE READY
45          000100      INTEN=BIT6    ;INTERRUPT ENABLE
46          100000      ERR=BIT15    ;COMPOSITE ERROR
47          040000      DERR=BIT14   ;DRIVE ERROR
48          020000      NXM=BIT13    ;NON-EXISTANT MEMORY ERROR
49          010000      DLT=BIT12    ;DATA LATE
50          004000      DCRC=BIT11   ;DATA CRC ERROR
51          004000      HCRC=BIT11   ;HEADER CRC ERROR
52          010000      HNF=BIT12    ;HEADER NOT FOUND ERROR
53          002000      OPI=BIT10    ;OPERATION INCOMPLETE ERROR
54          000200      CRDY=BIT7    ;CONTROLLER READY
55          000040      BA17=BIT5    ;EXTENDED BUS ADDRESS BIT 17
56          000020      BA16=BIT4    ;EXTENDED BUS ADDRESS BIT 16
57          000002      CRSET=BIT1   ;CONTROLLER RESET FUNCTION CODE
58          000004      GSTAT=BIT2   ;GET DRIVE STATUS FUNCTION CODE
59          000006      SEEK=BIT1:BIT2 ;SEEK FUNCTION CODE
60          000010      RDHDR=BIT3   ;READ HEADER FUNCTION CODE
61          000012      WRITE=BIT3:BIT1 ;WRITE FUNCTION CODE
62          000014      READ=BIT3:BIT2 ;READ FUNCTION CODE
63          000013      DRST=BIT3:BIT1:BIT0 ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
64          000003      GSBIT=BIT1:BIT0 ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
65          000001      MK=BIT0      ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)

```

```
66          000004          SIGN=BIT2          ;DIRECTION FOR SEEK(0 AWAY FROM SPINDLE)
67          000020          SKHS=BIT4          ;HEAD SELECT FOR SEEK
68          000100          HEAD=BIT6          ;HEAD SELECT FOR READ,WRITE,GET STATUS
69
70          ;OFFSET FOR HARDWARE P-TABLE
71
72          000000          CSR= 0
73          000002          VECT= 2
74          000004          TYPDR= 4
75          000006          DRBT= 6
76
77
78
79 002126          ENDMOD
80          .SBTTL GLOBAL DATA SECTION
81
82          ;
83
84
85 002126          BGNMOD GLBDAT
86
87 002126 000000          HDRFND: .WORD 0          ;1 HEADER IN BAD SECTOR LIST
88
89          ;HERE IS THE LIST OF TRACKS TO USE FOR THIS TEST
90          ;TRACKS ARE ENTERED BY 'FNDTRK' ROUTINE & 'FIXTRK' ROUTINE
91
92 002130 000000          OUT10: .WORD 0          ;OUTER TRK HEAD 0
93 002132 000000          OUT20: .WORD 0
94 002134 000000          OUT30: .WORD 0
95 002136 000000          OUT40: .WORD 0
96 002140 000000          OUT50: .WORD 0
97 002142 000000          OUT11: .WORD 0          ;OUTER TRK HEAD 1
98 002144 000000          OUT21: .WORD 0
99 002146 000000          OUT31: .WORD 0
100 002150 000000          OUT41: .WORD 0
101 002152 000000          OUT51: .WORD 0
102 002154 000000          OQU10: .WORD 0          ;1ST QUARTER TRK HEAD 0
103 002156 000000          OQU20: .WORD 0
104 002160 000000          OQU30: .WORD 0
105 002162 000000          OQU40: .WORD 0
106 002164 000000          OQU50: .WORD 0
107 002166 000000          OQU11: .WORD 0          ;1ST QUARTER TRK HEAD 1
108 002170 000000          OQU21: .WORD 0
109 002172 000000          OQU31: .WORD 0
110 002174 000000          OQU41: .WORD 0
111 002176 000000          OQU51: .WORD 0
112 002200 000000          MID10: .WORD 0          ;MIDDLE TRK HEAD 0
113 002202 000000          MID20: .WORD 0
114 002204 000000          MID30: .WORD 0
115 002206 000000          MID40: .WORD 0
116 002210 000000          MID50: .WORD 0
117 002212 000000          MID11: .WORD 0          ;MIDDLE TRK HEAD 1
118 002214 000000          MID21: .WORD 0
119 002216 000000          MID31: .WORD 0
120 002220 000000          MID41: .WORD 0
121 002222 000000          MID51: .WORD 0
122 002224 000000          TQU10: .WORD 0          ;3RD QUARTER TRK HEAD 0
```

123	002226	000000	TQU20:	.WORD	0	
124	002230	000000	TQU30:	.WORD	0	
125	002232	000000	TQU40:	.WORD	0	
126	002234	000000	TQU50:	.WORD	0	
127	002236	000000	TQU11:	.WORD	0	;3RD QUARTER TRK HEAD 1
128	002240	000000	TQU21:	.WORD	0	
129	002242	000000	TQU31:	.WORD	0	
130	002244	000000	TQU41:	.WORD	0	
131	002246	000000	TQU51:	.WORD	0	
132	002250	000000	INN10:	.WORD	0	;INNER TRK HEAD 0
133	002252	000000	INN20:	.WORD	0	
134	002254	000000	INN30:	.WORD	0	
135	002256	000000	INN40:	.WORD	0	
136	002260	000000	INN50:	.WORD	0	
137	002262	000000	INN11:	.WORD	0	;INNER TRK HEAD 1
138	002264	000000	INN21:	.WORD	0	
139	002266	000000	INN31:	.WORD	0	
140	002270	000000	INN41:	.WORD	0	
141	002272	000000	INN51:	.WORD	0	

142
143
144
145
146
147 002274
148
149
150
151 002334
152
153
154
155
156

.EVEN
;SECTOR LIST FOR LAST DRIVE WRITTEN
;MAP OF 16 SECTOR DRIVE BITS

157 002714 002134
158 002716 002160
159 002720 002204
160 002722 002230
161 002724 002254
162 002726 002146
163 002730 002172
164 002732 002216
165 002734 002242
166 002736 002266
167

SECLST: .BLKW 16.
;BUFFER TABLE FOR 24 X 5 MATRIX USED FOR ADJACENT CYLINDER TESTING.
SECBUF: .BLKW 5*24.
;LIST OF TRACKS USED TO OVERWRITE TEST.
;FIRST FIVE ARE CYLINDER ADDRESSES OF TOP SURFACE.
;LAST FIVE ARE CYLINDER ADDRESSES OF BOTTOM SURFACE.

157 002714 002134
158 002716 002160
159 002720 002204
160 002722 002230
161 002724 002254
162 002726 002146
163 002730 002172
164 002732 002216
165 002734 002242
166 002736 002266
167
168 002740 152525
169 002742 133333
170 002744 066666
171 002746 155555

OVWTRK: OUT30
OQU30
MID30
TQU30
INN30
OUT31
OQU31
MID31
TQU31
INN31
PATLST: .WORD 152525
.WORD 133333
.WORD 066666
.WORD 155555

1	002750	000000	TEM:	.WORD	0	
2	002752	000000	T.DRIVE:	.WORD	0	
3	002754	000000	FOUR:	.WORD	0	
4	002756	000000	FADJ:	.WORD	0	
5	002760	000000	TEMP:	.WORD	0	
6	002762	000000	LSTCLR:	.WORD	0	:LAST CONTROLLER
7	002764	000000	REASON:	.WORD	0	:DRIVE ERROR REASON
8	002766	000000	ERFLG:	.WORD	0	:ERROR FLAG
9	002770	000000	STFLG:	.WORD	0	:PROGRAM START UP FLAG
10	002772	000000	ADJLOC:	.WORD	0	:TRACK INDEX FOR ADJ. CYL TEST
11	002774	000000	ADJFLG:	.WORD	0	:FLAG FOR ADJ. STORE OR RETRIEVE
12	002776	000000	ADJDIR:	.WORD	0	:ADJACENT SEEK DIRECTION
13	003000	000000	DRSTAT:	.WORD	0	
14	003002	000000	HSFLG:	.WORD	0	
15	003004	000000	OSECT:	.WORD	0	
16	003006	000000	HEAD01:	.WORD	0	:SURFACE FLAG
17	003010	000000	DIRC:	.WORD	0	:DIRECTION OF SEEK
18	003012	000000	SURF:	.WORD	0	
19	003014	000000	CYL:	.WORD	0	
20	003016	000000	REVSK:	.WORD	0	:REVERSE SEEK
21	003020	000000	FORSK:	.WORD	0	:FORWARD SEEK
22	003022	000000	UUT:	.WORD	0	:UNIT UNDER TEST
23	003024	000000	SECT:	.WORD	0	:SECTOR
24	003026	000000	LSTDRV:	.WORD	0	:LAST DRIVE
25	003030	000000	GDATA:	.WORD	0	:GOOD DATA
26	003032	000000	BDATA:	.WORD	0	:BAD DATA
27	003034	000000	WCOUNT:	.WORD	0	:WORD COUNT
28	003036	000000	SECWRD:	.WORD	0	:SECTOR WORD
29	003040	000000	OFFSET:	.WORD	0	:INCREMENT
30	003042	000000	LSTRK:	.WORD	0	:LAST TRACK OF SEARCH
31	003044	000000	FRTTRK:	.WORD	0	:FIRST TRACK OF SEARCH
32	003046	000000	PRSTRK:	.WORD	0	:PRESENT TRACK
33	003050	000000	SURFACE:	.WORD	0	:SURFACE
34	003052	000000	TRKFND:	.WORD	0	:TRACK FOUND
35	003054	000000	TRKCNT:	.WORD	0	:TRACK COUNT
36	003056	000000	E.CS:	.WORD	0	:IMAGE OF CSR
37	003060	000000	E.BA:	.WORD	0	:IMAGE OF BUS ADDRESS
38	003062	000000	E.DA:	.WORD	0	:IMAGE OF DISK ADDRESS
39	003064	000000	E.MP:	.WORD	J	:IMAGE OF MULTI-PURPOSE WORD 1
40	003066	000000	E.MP1:	.WORD	0	:.. 2
41	003070	000000	E.MP2:	.WORD	0	:.. 3
42	003072	000000	BCS:	.WORD	0	:COMMAND LOADED
43	003074	000000	BBA:	.WORD	0	:BUS ADDRESS LOADED
44	003076	000000	BDA:	.WORD	0	:DISK ADDRESS LOADED
45	003100	000000	BMP:	.WORD	0	:WORD COUNT LOADED
46	003102	000000	SERNM1:	.WORD	0	:SERIAL NUMBER OF CARTRIDGE
47	003104	000000	SERNM2:	.WORD	0	:..
48	003106	000000	ADJTRK:	.WORD	0	:INSIDE/OUTSIDE FLAG
49	003110	000000	ADJUUT:	.WORD	0	:UUT FOR "ADJCYL"
50	003112	000000	ADJLC2:	.WORD	0	:TEMP LOC FOR "ADJCYL"
51	003114	000000	ADJLC3:	.WORD	0	:..
52	003116	000000	ADJLC4:	.WORD	C	:..
53	003120	000000	STSECT:	.WORD	0	:SECTORS TO WRITE "ADJCYL"
54	003122	000000	STSEC:	.WORD	0	:..
55	003124	000000	BUF:	.BLKW	3072.	:BUFFER FOR 24 SECTOR READS
56						
57	017124		DRBUF:			:DRIVE INFORMATION BUFFERS

```
4  
5  
12  
000004  
017124 000000 CSR ;CONTROLLER ADDRESS  
017126 000002 VEC ;VECTOR  
017130 000004 DSB ;DRIVE SELECT BITS  
017132 000006 PAT ;PATTERN UNIQUE TO DRIVE  
  
017134 000000 CSR ;CONTROLLER ADDRESS  
017136 000002 VEC ;VECTOR  
017140 000004 DSB ;DRIVE SELECT BITS  
017142 000006 PAT ;PATTERN UNIQUE TO DRIVE  
  
017144 000000 CSR ;CONTROLLER ADDRESS  
017146 000002 VEC ;VECTOR  
017150 000004 DSB ;DRIVE SELECT BITS  
017152 000006 PAT ;PATTERN UNIQUE TO DRIVE  
  
017154 000000 CSR ;CONTROLLER ADDRESS  
017156 000002 VEC ;VECTOR  
017160 000004 DSB ;DRIVE SELECT BITS  
017162 000006 PAT ;PATTERN UNIQUE TO DRIVE  
  
13  
17 017164 000000 ENDBUF: .WORD 0 ;END OF DRIVE BUFFERS  
18 017166 ENDMOD
```



```

1          .SBTTL GLOBAL TEXT SECTION
2 017166   BGNMOD GLBTXT
3
4          :GLOBAL TEXT
5
6
10
11 017166   103    117    116 CNTTOT: .ASCIZ /CONTROLLER TIMED OUT/
12 017213   105    122    122 INITWR: .ASCIZ /ERROR ON RECOVERING INITIAL WRITE BY FIRST DRIVE /
13 017275   105    122    122 DCKER: .ASCIZ /ERROR ON READ/
14 017313   115    111    116 FEW: .ASCIZ /MINIMUM OF TWO DRIVES REQUIRED/
15 017352   115    101    130 MANY: .ASCIZ /MAXIMUM OF FOUR DRIVES ALLOWED/
16 017411   124    105    123 NONE: .ASCIZ /TEST ABORTED - CAN'T FIND ANY GOOD SPOTS/
17 017462   124    122    131 OVMES: .ASCIZ /TRYING TO OVERWRITE DRIVE /
18 017515   124    122    131 RECMS: .ASCIZ /TRYING TO READ DATA WRITTEN BY DRIVE /
19 017563   103    101    116 ERRFND: .ASCIZ /CAN'T FIND FIVE ADJACENT TRACKS/
20 017623   117    126    105 OVWER: .ASCIZ /OVERWRITE ERROR/
21 017643   122    105    101 RECER: .ASCIZ /READ RECOVERY ERROR/
22 017667   105    122    122 FUNERR: .ASCIZ /ERROR IN SEEK OPERATION/
23 017717   115    111    123 SKER: .ASCIZ /MIS SEEK ERROR/
24 017736   106    117    122 FWD: .ASCIZ /FORWARD/
25 017746   122    105    126 REV: .ASCIZ /REVERSE/
26 017756   105    122    122 WRIT1: .ASCIZ /ERROR WRITING SECTOR/
27 020003   105    122    122 READ1: .ASCIZ /ERROR READING SECTOR/
28 020030   101    104    112 ADJTXT: .ASCIZ /ADJACENT CYLINDER TEST/
29
30
31
32
36
37          .EVEN
38
39 020060   ENDMOD
40
41          .SBTTL GLOBAL ERROR REPORT SECTION
42 020060   BGNMOD GLBERR
43
44 020060   BGNMSG ERR1
45
46 020060   PRINTB #FRM10,FRITRK,LSTTRK,SURFACE
47 020060   MOV SURFACE,-(SP)
48 020064   MOV LSTTRK,-(SP)
49 020070   MOV FRITRK,-(SP)
50 020074   MOV #FRM10,-(SP)
51 020100   MOV #4,-(SP)
52 020104   MOV SP,R0
53 020106   EMT C$PNTB
54 020110   ADD #12,SP
55
56
57          ENDMMSG
58 020114   L10000:
59 020114   EMT C$MSG
60 020114   104023
61
62          BGNMSG ERR2
63 020116   PRINTB #FRM4,(SR(R4),<B,DSB+1(R4)>
64 020116   CLR -(SP)
65 020120   BISB DSB+1(R4),(SP)

```

	020124	016446	000000	MOV	CSR(R4),-(SP)	
	020130	012746	021066	MOV	#FRM4,-(SP)	
	020134	012746	000003	MOV	#3,-(SP)	
	020140	010600		MOV	SP,R0	
	020142	104014		EMT	C\$PNTB	
	020144	062706	000010	ADD	#10,SP	
52	020150	004737	025506	JSR	PC,REGDMP	;REGISTER DUMP ROUTINE
53	020154			ENDMSG		
	020154			L10001:		
	020154	104023		EMT	C\$MSG	
54						
55	020156			BGNMSG	ERR3	
56	020156			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	
	020156	005046		CLR	-(SP)	
	020160	156416	000005	BISB	DSB+1(R4),(SP)	
	020164	016446	000000	MOV	CSR(R4),-(SP)	
	020170	012746	021066	MOV	#FRM4,-(SP)	
	020174	012746	000003	MOV	#3,-(SP)	
	020200	010600		MOV	SP,R0	
	020202	104014		EMT	C\$PNTB	
	020204	062706	000010	ADD	#10,SP	
57	020210	004737	025506	JSR	PC,REGDMP	;REGISTER DUMP ROUTINE
58	020214			PRINTB	#FRM5,<SURF>,<CYL>,SECT	
	020214	013746	003024	MOV	SECT,-(SP)	
	020220	013746	003014	MOV	CYL,-(SP)	
	020224	013746	003012	MOV	SURF,-(SP)	
	020230	012746	021127	MOV	#FRM5,-(SP)	
	020234	012746	000004	MOV	#4,-(SP)	
	020240	010600		MOV	SP,R0	
	020242	104014		EMT	C\$PNTB	
	020244	062706	000012	ADD	#12,SP	
59	020250			PRINTB	#FRM16,CSR(R3),<B,DSB+1(R3)>	
	020250	005046		CLR	-(SP)	
	020252	156316	000005	BISB	DSB+1(R3),(SP)	
	020256	016346	000000	MOV	CSR(R3),-(SP)	
	020262	012746	021716	MOV	#FRM16,-(SP)	
	020266	012746	000003	MOV	#3,-(SP)	
	020272	010600		MOV	SP,R0	
	020274	104014		EMT	C\$PNTB	
	020276	062706	000010	ADD	#10,SP	
60						
61	020302			ENDMSG		
	020302			L10002:		
	020302	104023		EMT	C\$MSG	
62						
63	020304			BGNMSG	ERR4	
64						
65	020304			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	
	020304	005046		CLR	-(SP)	
	020306	156416	000005	BISB	DSB+1(R4),(SP)	
	020312	016446	000000	MOV	CSR(R4),-(SP)	
	020316	012746	021066	MOV	#FRM4,-(SP)	
	020322	012746	000003	MOV	#3,-(SP)	
	020326	010600		MOV	SP,R0	
	020330	104014		EMT	C\$PNTB	
	020332	062706	000010	ADD	#10,SP	
66	020336	004737	025506	JSR	PC,REGDMP	;REGISTER DUMP ROUTINE

67	020342			PRINTB	#FRM5,<SURF>,<CYL>,SECT
	020342	013746	003024	MOV	SECT,-(SP)
	020346	013746	003014	MOV	CYL,-(SP)
	020352	013746	003012	MOV	SURF,-(SP)
	020356	012746	021127	MOV	#FRM5,-(SP)
	020362	012746	000004	MOV	#4,-(SP)
	020366	010600		MOV	SP,R0
	020370	104014		EMT	C\$PNTB
	020372	062706	000012	ADD	#12,SP
68	020376			PRINTB	#FRM6,REASON,LSTDRV,LSTCLR,LSTDRV
	020376	013746	003026	MOV	LSTDRV,-(SP)
	020402	013746	002762	MOV	LSTCLR,-(SP)
	020406	013746	003026	MOV	LSTDRV,-(SP)
	020412	013746	002764	MOV	REASON,-(SP)
	020416	012746	021176	MOV	#FRM6,-(SP)
	020422	012746	000005	MOV	#5,-(SP)
	020426	010600		MOV	SP,R0
	020430	104014		EMT	C\$PNTB
	020432	062706	000014	ADD	#14,SP
69	020436			PRINTB	#FRM7,DIRC
	020436	013746	003010	MOV	DIRC,-(SP)
	020442	012746	021217	MOV	#FRM7,-(SP)
	020446	012746	000002	MOV	#2,-(SP)
	020452	010600		MOV	SP,R0
	020454	104014		EMT	C\$PNTB
	020456	062706	000006	ADD	#6,SP
70					
71	020462			ENDMSG	
	020462			L10003:	
	020462	104023		EMT	C\$MSG
72					
73	020464			BGNMSG	ERR5
74	020464			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>
	020464	005046		CLR	-(SP)
	020466	156416	000005	BISB	DSB+1(R4),(SP)
	020472	016446	000000	MOV	CSR(R4),-(SP)
	020476	012746	021066	MOV	#FRM4,-(SP)
	020502	012746	000003	MOV	#3,-(SP)
	020506	010600		MOV	SP,R0
	020510	104014		EMT	C\$PNTB
	020512	062706	000010	ADD	#10,SP
75	020516	004737	025506	JSR	PC,REGDMP
76	020522			ENDMSG	
	020522			L10004:	
	020522	104023		EMT	C\$MSG
77					
78	020524			BGNMSG	ERR6
79	020524			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>
	020524	005046		CLR	-(SP)
	020526	156416	000005	BISB	DSB+1(R4),(SP)
	020532	016446	000000	MOV	CSR(R4),-(SP)
	020536	012746	021066	MOV	#FRM4,-(SP)
	020542	012746	000003	MOV	#3,-(SP)
	020546	010600		MOV	SP,R0
	020550	104014		EMT	C\$PNTB
	020552	062706	000010	ADD	#10,SP
80	020556	004737	025506	JSR	PC,REGDMP

```
81 020562          PRINTB #FRM17,R1,E.MP
    020562 013746 003064  MOV     E.MP,-(SP)
    020566 010146      MOV     R1,-(SP)
    020570 012746 022003  MOV     #FRM17,-(SP)
    020574 012746 000003  MOV     #3,-(SP)
    020600 010600      MOV     SP,R0
    020602 104014      EMT     C$PNTB
    020604 062706 000010  ADD     #10,SP
82 020610          ENDMSG
    020610          L10005:
    020610 104023      EMT     C$MSG
```

```
83
84
85
86          ;FORMAT STATEMENTS
87
91
```

```
92 020612      045      116      045 FRM1:  .ASCIZ  /%N%AUNLOAD DRIVE %01%A ON CONTROLLER %06%A AND REMOVE PACK%/
93 020707      045      116      045 FRM2:  .ASCIZ  /%N%APLACE PACK IN DRIVE %01%A ON CONTROLLER %06%A AND LOAD IT%/
94 021007      045      116      045 FRM3:  .ASCIZ  !%N%AWRONG PACK # IS %05%05%A # S/B %05%05%A%N%/
95 021066      045      101      103 FRM4:  .ASCIZ  /%ACONTROLLER: %06%A DRIVE: %01%A/
96 021127      045      101      110 FRM5:  .ASCIZ  /%AHEAD: %01%A CYL: %23%A SECTOR: %22%A/
97 021176      045      124      045 FRM6:  .ASCIZ  /%T%01%A ON %06%A/
98 021217      045      101      123 FRM7:  .ASCIZ  /%ASEEK DIRECTION: %T%A%ADATA:A%/
99 021257      045      101      127 FRM8:  .ASCIZ  !%AWORD: %23%A S/B: %06%A WAS: %06%A!
100 021323     045      104      063 FRM9:  .ASCIZ  /%D3%A WORDS BAD OUT OF 128 READ%/
101 021365     045      101      102 FRM10: .ASCIZ  /%ABETWEEN %23%A - %23%A HEAD: %01%A/
102 021431     045      116      045 FRM11: .ASCIZ  /%N%APWR FAIL NOT SUPPORTED%/
103 021466     045      101      102 FRM12: .ASCIZ  /%ABEFORE CS: %06%A BA: %06%A DA: %06%A MP: %06%/
104 021545     045      116      045 FRM13: .ASCIZ  /%N%AAFTER CS: %06%A BA: %06%A DA: %06%A MP: %06%A/
105 021630     045      116      045 FRM14: .ASCIZ  /%N%A DRIVE STATUS: %06%/
106 021657     045      116      045 FRM15: .ASCIZ  /%N%ACAN'T FIND BAD SECTOR FILE/
107 021716     045      101      101 FRM16: .ASCIZ  /%AADJACENT WRITTEN BY CONTROLLER: %06%A DRIVE: %01%A/
108 022003     045      101      105 FRM17: .ASCIZ  /%AEXP'D: %06%A REC'D: %06%A/
109 022037     045      116      045 FRM18: .ASCIZ  /%N%AUNLOAD ALL DRIVES TO BE USED%/
110 022102     045      116      045 ENDPAS: .ASCIZ  /%N%A END OF PASS%A%N%/
```

```
111
112
116
117
118
119
120          .EVEN
121
```

```
122 022130          ENDMOD
123
124 022130          BGNMOD HPTCODE
125 022130          BGNHW
    022130 000004      .WORD  L10006-L$HW/2
126 022132 174400      .WORD  174400 ;CSR
127 022134 000160      .WORD  160 ;VECTOR
128 022136 000001      .WORD  1 ;RL01 OR RL02 (RL01=1)
129 022140 000000      .WORD  0 ;DRIVE #
130 022142          ENDMW
    022142          L10006:
131
132 022142          ENDMOD
```

```
133  
134  
135 022142          BGNMOD  DSPCODE  
136  
137 022142          DISPATCH  1  
    022142 000001    .WORD    1  
    022144 031656    .WORD   11  
138  
139 022146          ENDMOD
```

```

1          .SBTTL  INITIALIZATION SECTION
2 022146   BGNMOD  INITCODE
3
4 022146   BGNINIT
5
6 022146   SETPRI  #340
  022146   012700 000340  MOV    #340,R0
  022152   104041  EMT    C$SPRI
7
8 022154   023727 002012 000002  CMP    LSUNIT,#2      ;MORE THAN TWO
9 022162   002005  BGE    90$           ;YES, OKAY
10
11 022164   ERRSF   19.,FEW
  022164   104421  TRAP  T$ERCODE
  022166   000023  .WORD 19
  022170   017313  .WORD FEW
12 022172   000137 023536  JMP    CMPENA       ;CLEAN CODE WHEN < 2 DRIVES
13
14 022176   023727 002012 000004 90$:  CMP    LSUNIT,#4     ;MORE THAN FOUR
15 022204   003405  BLE    91$           ;NO, OKAY
16
17 022206   ERRSF   20.,MANY
  022206   104421  TRAP  T$ERCODE
  022210   000024  .WORD 20
  022212   017352  .WORD MANY
18 022214   000137 023536  JMP    CMPENA       ;CLEAN CODE WHEN > 4 DRIVES
19
20 022220   013737 002012 003022 91$:  MOV    LSUNIT,UUT    ;GET NUMBER OF UNITS
21 022226   005001  CLR    R1            ;INIT P-TABLE
22 022230   012704 017124  MOV    #DRBUF,R4     ;SET UP DRIVE BUFFER
23 022234   012702 002740  MOV    #PATLST,R2    ;GET LIST OF PATTERNS
24 022240   005737 003022 1$:   TST    UUT           ;ANY P-TABLES LEFT?
25 022244   001423  BEQ    END           ;NO, GO TO END
26 022246   GPHARD  R1,R0           ;GET A P-TABLE
  022246   010100  MOV    R1,R0
  022250   104042  EMT    C$GPHRD
27 022252   012064 000000  MOV    (R0)+,CSR(R4) ;GET CSR
28 022256   012064 000002  MOV    (R0)+,VEC(R4) ;GET VECTOR
29 022262   012037 002752  MOV    (R0)+,T.DRIVE ;RL01/2 TYPE ... RL01-1
30 022266   011064 000004  MOV    (R0),DSB(R4)  ;GET DRIVE
31 022272   011264 000006  MOV    (R2),PAT(R4)
32 022276   005722  TST    (R2)+
33 022300   005201  INC    R1            ;NEXT P TABLE
34 022302   005337 003022  DEC    UUT           ;NEXT DRIVE
35 022306   062704 000010  ADD    #PAT+2,R4
36 022312   000752  BR     1$
37 022314   013737 002012 003022 END:  MOV    LSUNIT,UUT    ;GET BEGINNING OF BUFFER
38 022322   012704 017124  MOV    #DRBUF,R4     ;CLEAR ADJ. TEST FLAG
39 022326   005037 002756  CLR    FADJ          ;CLEAR OVERWRITE FLAG
40 022332   005037 002754  CLR    FOWR
41 022336   READEF #EF.PWR
  022336   012700 000034  MOV    #EF.PWR,R0
  022342   104050  EMT    C$REFG
42 022344   BNCOMPLETE SETUP
  022344   103010  BCC   SETUP
43 022346   PRINTF #FRM11
  022346   012746 021431  MOV    #FRM11,-(SP)

```

022352	012746	000001	MOV	#1,-(SP)
022356	010600		MOV	SP,R0
022360	104017		EMT	(SPNTF
022362	062706	000004	ADD	#4,SP

```

1      :INITIALIZE ROUTINE
2      :WE ATTEMPT TO LOCATE 5 PERFECT ADJACENT TRACKS AT 5 SPOTS
3      :ACROSS THE PACK.
4      :THE 5 SPOTS ARE: (EACH SURFACE)
5      :
6      :OUTER - TRACK 0 - 16 (BOTH RLO1 & RLO2)
7      :INNER - TRACK 238 - 254 (RLO1) OR 494 - 510 (RLO2)
8      :MIDDLE - TRACK 120 - 136 (RLO1) OR 248 - 264 (RLO2)
9      :ONE QUARTER - TRACK 56 - 72 (RLO1) OR 120 - 136 (RLO2)
10     :THREE QUARTER - TRACK 184 - 200 (RLO1) OR 376 - 392 (RLO2)
11     :
12     :IF WE FIND ANY BAD SPOTS, WE WILL REPORT SU.....
13
14
15 022366 005237 002770      SETUP:  INC      STFLG      ;INDICATE A START COMMAND
16 022372 012737 177777      003102  MOV      #-1,SERNM1
17 022400 012737 177777      003104  MOV      #-1,SERNM2
18 022406 012746 022037      PRINTF   #FRM18
    022406 012746 022037      MOV      #FRM18,-(SP)
    022412 012746 000001      MOV      #1,-(SF)
    022416 010600      MCV      SP,R0
    022420 104017      EMT      C$PNTF
    022422 062706 000004      ADD      #4,SP
19 022426 004537 031260      JSR      R5,LOAD      ; TELL OPERATOR TO LOAD
20 022432 004537 030512      JSR      R5,SERNUM    ; GET SERIAL NUMBER
21 022436 004537 027766      JSR      R5,MERGE     ; MERGE BAD SECTOR FILES
22 022442 012701 002130      MOV      #OUT10,R1    ; INITIALIZE ALL TRACKS
23 022446 012700 000062      MOV      #50,R0
24 022452 012721 177777      1$:     MOV      #177777,(R1)+
25 022456 005300      DEC      R0
26 022460 001374      BNE      1$
27
28 022462 004537 030214      JSR      R5,FNDTRK    ; TRY TO FIND FIVE TRACKS
29 022466 000001      1      ; INWARD SEARCH
30 022470 000000      0      ; TOP SURFACE
31
32 022472 000000 000020      .WORD    0,16.
33 022476 000000 000020      .WORD    0,16.
34
35 022502 005737 003052      TST      TRKFND      ; WAS SEARCH SUCCESSFUL???
36 022506 001005      BNE      2$          ; YES
37
38 022510 104463      EPRHRD   10,ERRFND,ERR1 ; NO TRACKS
    022510 000012      TRAP     T$ERCODE
    022512 000012      .WORD    10
    022514 017563      .WORD    ERRFND
    022516 020060      .WORD    ERR1
39 022520 000404      BR       3$
40
41 022522 012700 002130      2$:     MOV      #OUT10,R0      ; STORE AWAY TRACKS FOUND
42 022526 004537 030456      JSR      R5,FXCYL

```



```

1 022532 004537 030214      3$:   JSR      R5,FNDTRK      ;TRY TO FIND FIVE TRACKS
2 022536 000001              1      ;INWARD SEARCH
3 022540 000001              1      ;BOTTOM SURFACE
4 022542 000000 000020      .WORD  0,16.
5 022546 000000 000020      .WORD  0,16.
6
7 022552 005737 003052      TST      TRKFND      ;WAS SEARCH SUCCESSFUL???
8 022556 001005              BNE      4$          ;YES
9
10 022560              ERRHRD  10.,ERRFND,ERR1 ;NO TRACKS
    022560 104463          TRAP    T$ERCODE
    022562 000012          .WORD  10
    022564 017563          .WORD  ERRFND
    022566 020060          .WORD  ERR1
11 022570 000404          BR      5$
12
13 022572 012700 002142      4$:   MOV      #OUT11,RO      ;STORE TRACKS AWAY
14 022576 004537 030456      JSR      R5,FXCYL
15 022602 004537 030214      5$:   JSR      R5,FNDTRK      ;FIND NEXT 5 TRACK
16 022606 177777              -1     ;OUTWARD SEARCH
17 022610 000000              0      ;TOP SURFACE
18 022612 000376 000356      .WORD  254.,238.
19 022616 000776 000756      .WORD  510.,494.
20
21 022622 005737 003052      TST      TRKFND      ;WAS SEARCH SUCCESSFUL?
22 022626 001005              BNE      6$          ;YES
23
24 022630              ERRHRD  10.,ERRFND,ERR1 ;NO TRACKS
    022630 104463          TRAP    T$ERCODE
    022632 000012          .WORD  10
    022634 017563          .WORD  ERRFND
    022636 020060          .WORD  ERR1
25 022640 000404          BR      7$          ;SKIP
26
27 022642 012700 002250      6$:   MOV      #INN10,RO      ;STORE AWAY TRACKS FOUND
28 022646 004537 030456      JSR      R5,FXCYL

```

```

1 022652 004537 030214      7$:   JSR      R5,FNDTRK      ;NEXT SET
2 022656 177777              -1      ;OUTWARD SEARCH
3 022660 000001              1      ;BOTTOM SURFACE
4 022662 000376 000356      .WORD   254.,238.
5 022666 000776 000756      .WORD   510.,494.
6
7 022672 005737 003052      TST     TRKFND      ;SEARCH SUCCESSFUL?
8 022676 001005              BNE     8$          ;YES
9
10 022700      ERRHRD  10.,ERRFND,ERR1 ;NO TRACKS
    022700 104463      TRAP   T$ERCODE
    022702 000012      .WORD  10
    022704 017563      .WORD  ERRFND
    022706 020060      .WORD  ERR1
11 022710 000404      BR     9$
12
13 022712 012700 002262      8$:   MOV     #INN11,R0      ;STORE AWAY TRACKS FOUND
14 022716 004537 030456      JSR    R5,FXCYL
15
16 022722 004537 030214      9$:   JSR     R5,FNDTRK      ;NEXT SET
17 022726 000001              1      ;INWARD SEARCH
18 022730 000000              0      ;TOP SURFACE
19 022732 000176 000210      .WORD   126.,136.
20 022736 000376 000410      .WORD   254.,264.
21
22 022742 005737 003052      TST     TRKFND      ;DID WE FIND A SET
23 022746 001020              BNE     10$         ;YES
24
25 022750 004537 030214      JSR     R5,FNDTRK      ;NEXT SET (OTHER SIDE)
26 022754 177777              -1      ;OUTWARD SEARCH
27 022756 000000              0      ;TOP SURFACE
28 022760 000202 000170      .WORD   130.,120.
29 022764 000402 000370      .WORD   258.,248.
30 022770 005737 003052      TST     TRKFND      ;DID WE FIND A SET
31 022774 001005              BNE     10$         ;YES
32
33 022776      ERRHRD  10.,ERRFND,ERR1 ;NO TRACKS
    022776 104463      TRAP   T$ERCODE
    023000 000012      .WORD  10
    023002 017563      .WORD  ERRFND
    023004 020060      .WORD  ERR1
34 023006 000404      BR     11$
35
36 023010 012700 002200      10$:  MOV     #MID10,R0      ;STORE AWAY
37 023014 004537 030456      JSR    R5,FXCYL

```

```

1 023020 004537 030214      11$: JSR      R5,FNDTRK      ;NEXT SET
2 023024 000001              1      ;INWARD SEARCH
3 023026 000001              1      ;BOTTOM SURFACE
4 023030 000176 000210      .WORD    126.,136.      ;RANGE
5 023034 000376 000410      .WORD    254.,264.
6
7 023040 005737 003052      TST      TRKFND          ;SUCCESS?
8 023044 001020              BNE      12$            ;YES
9
10 023046 004537 030214     JSR      R5,FNDTRK      ;LOOK THE OTHER SIDE
11 023052 177777              -1     ;OUTWARD
12 023054 000001              1      ;BOTTOM SURFACE
13 023056 000202 000170      .WORD    130.,120.
14 023062 000402 000370      .WORD    258.,248.
15
16 023066 005737 003052      TST      TRKFND          ;SUCCESS?
17 023072 001005              BNE      12$            ;YES
18
19 023074              ERRHRD   10.,ERRFND,ERR1 ;NO TRACKS
   023074 104463          TRAP    T$ERCODE
   023076 000012          .WORD   10
   023100 017563          .WORD   ERRFND
   023102 020060          .WORD   ERR1
20 023104 000404          BR      13$
21
22 023106 012700 002212     12$: MOV      #MID11,R0      ;STORE AWAY THE TRACKS FOUND
23 023112 004537 030456     JSR      R5,FXCYL
24
25 023116 004537 030214     13$: JSR      R5,FNDTRK      ;NEXT SET
26 023122 000001              1      ;INWARD
27 023124 000000              0      ;TOP SURFACE
28 023126 000076 000110      .WORD    62.,72.
29 023132 000176 000210      .WORD    126.,136.
30
31 023136 005737 003052      TST      TRKFND          ;SUCCESS?
32 023142 001020              BNE      14$            ;YES
33
34 023144 004537 030214     JSR      R5,FNDTRK      ;LOOK OTHER SIDE
35 023150 177777              -1     ;OUTWARD
36 023152 000000              0      ;TOP SURFACE
37 023154 000102 000070      .WORD    66.,56.
38 023160 000202 000170      .WORD    130.,120.
39
40 023164 005737 003052      TST      TRKFND          ;SUCCESS?
41 023170 001005              BNE      14$            ;YES
42
43 023172              ERRHRD   10.,ERRFND,ERR1 ;NO TRACKS
   023172 104463          TRAP    T$ERCODE
   023174 000012          .WORD   10
   023176 017563          .WORD   ERRFND
   023200 020060          .WORD   ERR1
44 023202 000404          BR      15$
45
46 023204 012700 002154     14$: MOV      #00U10,R0     ;STORE AWAY NEXT SET
47 023210 004537 030456     JSR      R5,FXCYL

```

1	023214	004537	030214	15\$:	JSR	R5,FNDTRK	:LOOK FOR NEXT SET
2	023220	000001			1		:INWARD
3	023222	000001			1		:BOTTOM
4	023224	000076	000110		.WORD	62.,72.	:RANGE
5	023230	000176	000210		.WORD	126.,136.	
6							
7	023234	005737	003052		TST	TRKFND	:SUCCESS?
8	023240	001020			BNE	16\$:YES
9							
10	023242	004537	030214		JSR	R5,FNDTRK	:LOOK FOR ANOTHER SET
11	023246	177777			-1		:OUTWARD
12	023250	000001			1		:BOTTOM
13	023252	000102	000070		.WORD	66.,56.	:RANGE
14	023256	000202	000170		.WORD	130.,120.	
15							
16	023262	005737	003052		TST	TRKFND	:SUCCESS?
17	023266	001005			BNE	16\$:YES
18							
19	023270				ERRHRD	10.,ERRFND,ERR1	:NO TRACKS
	023270	104463			TRAP	T\$ERCODE	
	023272	000012			.WORD	10	
	023274	017563			.WORD	ERRFND	
	023276	020060			.WORD	ERR1	
20	023300	000404			BR	17\$	
21							
22	023302	012700	002166	16\$:	MOV	#0QU11,R0	:STORE AWAY TRACKS
23	023306	004537	030456		JSR	R5,FXCYL	
24							
25	023312	004537	030214	17\$:	JSR	R5,FNDTRK	:NEXT SET OF TRACKS
26	023316	000001			1		:INWARD
27	023320	000000			0		:TOP SURFACE
28	023322	000276	000310		.WORD	190.,200.	:RANGE
29	023326	000576	000610		.WORD	382.,392.	
30							
31	023332	005737	003052		TST	TRKFND	:SUCCESS?
32	023336	001020			BNE	18\$:YES
33							
34	023340	004537	030214		JSR	R5,FNDTRK	:LOOK OTHER SIDE
35	023344	177777			-1		:OUTWARD SEARCH
36	023346	000000			0		:TOP
37	023350	000302	000270		.WORD	194.,184.	
38	023354	000602	000570		.WORD	386.,376.	
39							
40	023360	005737	003052		TST	TRKFND	:SUCCESS
41	023364	001005			BNE	18\$:YES
42							
43	023366				ERRHRD	10.,ERRFND,ERR1	:NO TRACKS
	023366	104463			TRAP	T\$ERCODE	
	023370	000012			.WORD	10	
	023372	017563			.WORD	ERRFND	
	023374	020060			.WORD	ERR1	
44	023376	000404			BR	19\$	
45							
46	023400	012700	002224	18\$:	MOV	#1QU10,R0	:STORE TRACKS AWAY
47	023404	004537	030456		JSR	R5,FXCYL	

```

1 023410 004537 030214      19$: JSR      R5,FNDTRK      ;NEXT SET
2 023414 000001              1      ;INWARD
3 023416 000001              1      ;BOTTOM SURFACE
4 023420 000276 000310      .WORD    190.,200.      ;RANGE
5 023424 000576 000610      .WORD    382.,392.
6
7 023430 005737 003052      TST      TRKFND          ;SUCCESS?
8 023434 001020              BNE      20$            ;YES
9
10 023436 004537 030214      JSR      R5,FNDTRK      ;OTHER SET
11 023442 177777              -1      ;OUTWARD
12 023444 000001              1      ;BOTTOM SURFACE
13 023446 000302 000270      .WORD    194.,184.      ;RANGE
14 023452 000602 000570      .WORD    386.,376.
15
16 023456 005737 003052      TST      TRKFND          ;SUCCESS
17 023462 001005              BNE      20$            ;YES
18
19 023464              ERRHRD   10.,ERRFND,ERR1 ;NO TRACKS
   023464 104463          TRAP     T$ERCODE
   023466 000012          .WORD    10
   023470 017563          .WORD    ERRFND
   023472 020060          .WORD    ERR1
20 023474 000404          BR       21$
21
22 023476 012700 002236      20$: MOV      #TQU11,R0      ;STORE SET AWAY
23 023502 004537 030456      JSR      R5,FXCYL
24
25 023506 012700 002130      21$: MOV      #OUT10,R0      ;DID WE FIND ANY AT ALL
26 023512 012701 000062      MOV      #50.,R1
27 023516 022720 177777      22$: CMP      #-1,(R0)+
28 023522 001016              BNE      EXIT
29 023524 005301              DEC      R1
30 023526 001373              BNE      22$
31 023530              ERRSF   3.,NONE
   023530 104421          TRAP     T$ERCODE
   023532 000003          .WORD    3
   023534 017411          .WORD    NONE
32 023536 005001              CMPENA: CLR      R1
33 023540 013700 002012      MOV      L$UNIT,R0
34 023544              24$: DODU     R1          ;DO DROP UNIT
   023544 010100          MOV      R1,R0
   023546 104053          EMT      C$DODU
35 023550 005201              INC      R1
36 023552 005300              DEC      R0
37 023554 001373              BNE      24$
38 023556              DOCLN   C$DCLN
   023556 104044          EMT
39
40 023560              EXIT:
41 023560              L10007: ENDINIT
   023560              EMT      C$INIT
   023560 104011          ENDMOD
42 023562

```

1	023562		BGNMOD	CLNCODE	
2	023562			BGNCLN	
3					
4	023562	000240		NOP	
5					
6	023564			ENDCIN	
	023564		L10010:		
	023564	104012		EMT	CSCLEAN
7	023566			ENDMOD	
8					
9	023566		BGNMOD	DRPCODE	
10	023566			BGNDU	
11	023566	000240		NOP	
12	023570			ENDDU	
	023570		L10011:		
	023570	104055		EMT	CSDU
13	023572			ENDMOD	

```

1          .SBTTL GLOBAL SUBROUTINES SECTION
2 023572   BGNMOD GLBSUB
3
4          ;ROUTINE TO PERFORM OVERWRITE
5          ;CALL: JSR   R5,OVWPER
6          :       SECTORS TO WRITE FORWARD
7          :       SECTORS TO WRITE REVERSE
8
9 023572   010046   OVWPER: MOV   R0,-(SP)           ;SAVE R0, R1, R2, R3
10 023574   010146   MOV   R1,-(SP)
11 023576   010246   MOV   R2,-(SP)
12 023600   010346   MOV   R3,-(SP)
13 023602   005000   CLR   R0           ;R0 HAS COUNT IF R0<5.
14 023604   012537   003020  MOV   (R5)+,FORSK ;USE TOP SURFACE, IF R0>5.
15 023610   012537   003016  MOV   (R5)+,REVSK ;USE BOTTOM SURFACE, IF R0>1
16                                     ;DONE.
17 023614   012701   002714   MOV   #OVWTRK,R1 ;GET START OF LIST OF TRACKS
18 023620   011102   1$:   MOV   (R1),R2 ;GET POINTER TO TRACK
19 023622   021227   177777   CMP   (R2),#-1 ;LEGIT TRACK?????
20 023626   001500   BEQ   3$          ;NO, EXIT
21
22 023630   005037   003014   CLR   CYL         ;CLEAR CYLINDER/HEAD FOR SEEK
23 023634   005037   003012   CLR   SURF
24 023640   020027   000005   CMP   R0,#5      ;TOP/BOTTOM
25 023644   002402   BLT   2$          ;TOP, BRANCH
26 023646   005237   003012   INC   SURF       ;BOTTOM SURFACE
27 023652   004537   025240   2$:   JSR   R5,SKCYL ;SEEK TO CYLINDER
28 023656   005037   003014   CLR   CYL
29 023662   051237   003014   BIS   (R2),CYL
30 023666   004537   025240   JSR   R5,SKCYL ;SEEK TO PROPER CYLINDER
31 023672   013703   003020   MOV   FORSK,R3 ;SECTORS TO WRITE
32 023676   004537   024054   JSR   R5,WRSEC ;GO WRITE SECTORS
33 023702   000034   .WORD 28.
34 023704   012737   017736   003010  MOV   #FWD,DIRC ;SET FORWARD DIRECTION
35 023712   004537   026164   JSR   R5,VEROW ;VERIFY OVERWRITE
36 023716   004537   026550   JSR   R5,VEROD ;VERIFY OTHER DRIVES DATA
37 023722   005037   003014   CLR   CYL
38 023726   022737   000001   002752  CMP   #1,T.DRIVE ;RL01?
39 023734   001004   BNE   50$        ;NO
40 023736   052737   000377   003014  BIS   #377,CYL ;SET TO GO TO MAX CYL
41 023744   000403   BR    51$
42 023746   052737   000777   003014  50$:  BIS   #777,CYL ;MAX CYL FOR RL02
43 023754   004537   025240   51$:  JSR   R5,SKCYL ;SEEK TO MAX CYLINDER ON DRIVE
44 023760   005037   003014   CLR   CYL
45 023764   005037   003012   CLR   SURF
46 023770   051237   003014   BIS   (R2),CYL
47 023774   004537   025240   JSR   R5,SKCYL ;DO ANOTHER SEEK
48
49 024000   013703   003016   MOV   REVSK,R3 ;SECTORS TO WRITE
50 024004   004537   024054   JSR   R5,WRSEC ;WRITE THEM
51 024010   000034   .WORD 28.
52 024012   012737   017746   003010  MOV   #REV,DIRC ;SET DIRECTION
53 024020   004537   026164   JSR   R5,VEROW ;VERIFY OVERWRITE
54 024024   004537   026550   JSR   R5,VEROD ;VERIFY OTHER DRIVES DATA
55
56 024030   005721   3$:   TST   (R1)+      ;INCREMENT TO NEXT TRACK
57 024032   005200   INC   R0         ;ACCOUNT FOR IT

```

58	024034	020027	000012	CMP	R0,#10.	;DONE?
59	024040	001267		BNE	18	;NO, GO BACK
60						
61	024042	012603		MOV	(SP)+,R3	;RESTORE REG.
62	024044	012602		MOV	(SP)+,R2	
63	024046	012601		MOV	(SP)+,R1	
64	024050	012600		MOV	(SP)+,R0	
65	024052	000205		RTS	R5	;EXIT


```

1          ;ROUTINE TO WRITE SECTORS
2          ;USED IN OVERWRITE TEST;ADJACENT CYLINDER TEST
3          ;CALL JSR R5,WRSEC
4          ;
5          ;.WRD ;STARTING SECTOR
6          ;R3 HAS BITMAP OF SECTORS TO WRITE
7          ;R4 HAS DRIVE BUFFER POINTER
8 024054 010046 WRSEC: MOV R0,-(SP) ;SAVE R0
9 024056 010146 MOV R1,-(SP) ;SAVE R1
10 024060 010246 MOV R2,-(SP) ;SAVE R2
11 024062 012701 003124 MOV #BUF,R1 ;WRITE PATTERN INTO
12 024066 012702 000200 MOV #128.,R2 ;MEMORY THAT WE
13 024072 016421 000006 2$: MOV PAT(R4),(R1)+ ;WILL WRITE ONTO
14 024076 005302 DEC R2 ;PACK FOR THIS
15 024100 001374 BNE 2$ ;DRIVE
16 024102 012701 100000 MOV #100000,R1 ;MASK FOR BIT MAP
17 024106 012737 000007 002750 MOV #7,TEM
18 024114 053702 003014 BIS CYL,R2
19 024120 006302 120$: ASL R2
20 024122 005337 002750 DEC TEM
21 024126 001374 BNE 120$
22 024130 005737 003012 TST SURF
23 024134 001402 BEQ 3$ ;0, SKIP
24 024136 052702 000100 BIS #HEAD,R2 ;SET BOTTOM HEAD
25 024142 052502 3$: BIS (R5)+,R2 ;START AT SECTOR 28.
26 024144 030103 4$: BIT R1,R3 ;WRITE THIS SECTOR?
27 024146 001452 BEQ 5$ ;NO
28
29 024150 005037 003002 CLR HSFLG
30 024154 012737 177600 003100 MOV #-128.,BMP ;LOAD WORD COUNT
31 024162 010237 003076 MOV R2,BDA ;LOAD DISK ADDRESS
32 024166 010237 002760 MOV R2,TEMP ;SAVE DISK ADDRESS
33 024172 042702 177700 BIC #177700,R2
34 024176 020227 000047 CMP R2,#39.
35 024202 003403 BLE 6$
36 024204 162737 000050 003076 SUB #40.,BDA
37 024212 012737 003124 003074 6$: MOV #BUF,BBA ;LOAD BUS ADDRESS
38 024220 013702 002760 MOV TEMP,R2 ;RESTORE DISK ADDRESS
39 024224 004537 031360 11$: JSR R5,LDFUNC ;GO WRITE
40 024230 000012 WRITE
41 024232 005737 002766 TST ERFLG ;ERROR IN WRITING
42 024236 001416 BEQ 5$ ;NO,OKAY
43 024240 005737 003002 TST HSFLG
44 024244 001007 BNE 10$
45 024246 ERRSOFT 100.,WRIT1,ERR2
024246 104464 TRAP TSERCODE
024250 000144 .WORD 100
024252 017756 .WORD WRIT1
024254 020116 .WORD ERR2
46 024256 005237 003002 INC HSFLG
47 024262 000760 BR 11$
48 024264 10$: ERRHRD 110.,WRIT1,ERR2
024264 104463 TRAP TSERCODE
024266 000156 .WORD 110
024270 017756 .WORD WRIT1
024272 020116 .WORD ERR2

```

50	024274	005202	SS:	INC	R2	;NEXT SECTOR
51	024276	000241		CLC		;CLEAR CARRY BIT
52	024300	006001		ROR	R1	;DONE?
53	024302	103320		BCC	48	;NO GO BACK
54	024304	012602		MOV	(SP)+,R2	;REGISTERS AND EXIT
55	024306	012601		MOV	(SP)+,R1	
56	024310	012600		MOV	(SP)+,R0	
57	024312	000205		RTS	R5	

```

1 024314 005037 003106          ADJCYL: CLR      ADJTRK      ; INSIDE/OUTSIDE TRACK FLAG
2 024320 005037 003006          CLR      HEAD01      ; INIT TO TOP SURFACE
3 024324 012737 000001 003110  MOV      #1,ADJUUT    ; START OF TRACK LIST
4 024332 012701 002130          21$:  MOV      #OUT10,R1      ;
5 024336 012537 002772          20$:  MOV      (R5)+,ADJLOC    ; PICK UP TRACK OFFSET
6 024342 001003                    BNE      1$           ; IS THERE ONE?
7 024344 005037 002776          CLR      ADJDIR
8 024350 000205                    RTS      R5           ; NO EXIT
9 024352 012537 003112          1$:   MOV      (R5)+,ADJLC2    ; YES, GET REST OF INFO
10 024356 012537 003114          MOV      (R5)+,ADJLC3
11 024362 012537 003116          MOV      (R5)+,ADJLC4
12 024366 113700 002772          2$:   MOVB    ADJLOC,R0      ; GET OFFSET
13 024372 012737 000020 003122  MOV      #16.,STSEC    ; STARTING SECTOR IS 16
14
15 024400 010102                    MOV      R1,R2       ; GET START INTO R2
16
17 024402 005300          3$:   DEC      R0           ; DOWN COUNT OFFSET
18 024404 001414                    BEQ      4$           ; FOUND IT?
19
20 024406 005722                    TST     (R2)+         ; INDEX (R2)
21 024410 062737 000042 003122  ADD     #34.,STSEC    ; NO, NEXT SECTOR
22 024416 022737 000050 003122  CMP     #40.,STSEC
23 024424 003366                    BGT     3$
24 024426 162737 000050 003122  SUB     #40.,STSEC
25 024434 000762                    BR      3$           ; BACK FOR NEXT
26
27 024436 021227 177777          4$:   CMP     (R2),#-1     ; LEGAL TRACK?
28 024442 001002                    BNE     5$           ; YES, CONTINUE
29
30 024444 000137 025112          JMP     13$          ; NO PICK UP NEXT SET
31
32 024450 005037 003012          5$:   CLR     SURF         ; SET UP FOR OUTER TRACK
33 024454 005037 003014          CLR     CYL
34
35 024460 005737 003006          TST     HEAD01       ; WHICH HEAD?
36 024464 001403                    BEQ     6$           ; TOP, SKIP
37
38 024466 052737 000001 003012  BIS     #1,SURF ; LOWER HEAD, SET IT!
39
40 024474 004537 025240          6$:   JSR     R5,SKCYL     ; SEEK TO OUTER TRACK
41
42 024500 011237 003014          MOV     (R2),CYL     ; GET DESIRED TRACK
43
44 024504 004537 025240          JSR     R5,SKCYL     ; SEEK TO IT
45 024510 012737 017736 003010  MOV     #FWD,DIRC    ; SEEK DIRECTION
46 024516 113703 002773          MOVB   ADJLOC+1,P3   ; GET SECTORS TO WRITE
47 024522 000303                    SWAB   R3            ; ALIGN IT
48 024524 042703 000377          BIC     #377,R3      ; CLEAR OUT HIGH BYTE
49
50 024530 022737 000047 003122  CMP     #39.,STSEC    ; OVER FORTY?
51 024536 002003                    BGE     7$           ; NO, CONTINUE
52
53 024540 162737 000050 003122  SUB     #40.,STSEC    ; YES BACK II UP
54 024546 013737 003122 024560  7$:   MOV     STSEC,8$     ; STARTING SECTOR
55
56 024554 004537 024054          JSR     R5,WRSEC     ; WRITE SECTORS
57 024560 000000          .WORD  0

```

58	024562	013737	024560	024574	MOV	8\$,108\$	
59	024570	004537	027076		JSR	R5,VAJWR	:VERIFY THIS WRITE
60	024574	000000			.WORD	0	108\$:
61	024576	013737	024574	024610	MOV	108\$,208\$	
62	024604	004537	027342		JSR	R5,BSVWR	
63	024610	000000			.WORD	0	208\$:
64	024612	013737	003122	003120	MOV	STSEC,SISEC1	:GET OTHER SECTORS TO WRITE
65	024620	062737	000010	003120	ADD	#8.,SISEC1	:8 SECTORS GONE BY
66	024626	022737	000047	003120	CMP	#39.,SISEC1	:GONE PAST 40?
67	024634	002003			BGE	9\$:NO, OKAY
68							
69	024636	162737	000050	003120	SUB	#40.,STSEC1	:YES BACK IT UP
70							
71	024644	013703	003112		MOV	ADJLC2,R3	:GET SECTORS TO WRITE
72							
73	024650	013737	003120	024662	MOV	STSEC1,10\$:STARTING SECTORS
74							
75	024656	004537	024054		JSR	R5,WRSEC	:WRITE SECTORS
76	024662	000000			.WORD	0	10\$:
77	024664	013737	024662	024676	MOV	10\$,110\$	
78	024672	004537	027076		JSR	R5,VAJWR	:VERIFY THIS WRITE
79	024676	000000			.WORD	0	110\$:
80	024700	013737	024676	024712	MOV	110\$,210\$	
81	024706	004537	027342		JSR	R5,BSVWR	:VERIFY ADJ CYL + 1
82	024712	000000			.WORD	0	210\$:
83	024714	022737	000001	002752	CMP	#1,T.DRIVE	
84	024722	001004			BNE	77\$	
85	024724	012737	000377	003014	MOV	#377,CYL	
86	024732	000403			BR	88\$	
87							
88	024734	012737	000777	003014	MOV	#777,CYL	77\$:
89							
90	024742	004537	025240		JSR	R5,SKCYL	88\$:
91							
92	024746	011237	003014		MOV	(R2),CYL	:SEEK BACK TO PROPER TRACK
93							
94	024752	004537	025240		JSR	R5,SKCYL	:SEEK TO PROPER CYLINDER
95	024756	012737	017746	003010	MOV	#REV,DIRC	:SEEK DIRECTION
96	024764	113703	003115		MOVB	ADJLC3+1,R3	:GET SECTORS TO WRITE
97							
98	024770	000303			SWAB	R3	:ALIGN IT
99	024772	042703	000377		BIC	#377,R3	:CLEAR OUT HIGH BYTE
100	024776	013737	003122	025010	MOV	STSEC,11\$	
101							
102	025004	004537	024054		JSR	R5,WRSEC	:WRITE PROPER SECTOR
103	025010	000000			.WORD	0	11\$:
104							
105	025012	013737	025010	025024	MOV	11\$,111\$	
106	025020	004537	027076		JSR	R5,VAJWR	:VERIFY THIS WRITE
107	025024	000000			.WORD	0	111\$:
108	025026	013737	025024	025040	MOV	111\$,211\$	
109	025034	004537	027342		JSR	R5,BSVWR	
110	025040	000000			.WORD	0	211\$:
111	025042	013703	003116		MOV	ADJLC4,R3	:GET SECTORS
112	025046	013737	003120	025060	MOV	STSEC1,12\$:GET SECTORS TO WRITE
113							
114	025054	004537	024054		JSR	R5,WRSEC	:WRITE PROPER SECTORS

```

115 025060 000000          12$: .WORD 0
116
117
118 025062 013737 025060 025074      MOV 12$,112$
119 025070 004537 027076          JSR R5,VAJWR      ;VERIFY THIS WRITE
120 025074 000000          112$: .WORD 0
121
122
123 025076 013737 025074 025110      MOV 112$,212$
124 025104 004537 027342          JSR R5,BSVWR      ;VERIFY ADJ CYLINDERS + 1
125 025110 000000          212$: .WORD 0
126
127
128 025112 005737 003006          13$: TST HEAD01      ;WHICH HEAD WERE WE DOING?
129 025116 001003          BNE 14$
130 025120 005237 003006          INC HEAD01
131 025124 000402          BR 99$
132 025126 005037 003006          14$: CLR HEAD01      ;NEXT SET OF TRACKS
133 025132 062701 000012          99$: ADD #10.,R1      ;NEXT SET OF TRACKS
134 025136 020127 002272          CMP R1,#INNS1     ;END OF LIST
135 025142 002002          BGE 18$           ;END OF TRACK LIST
136 025144 000137 024366          JMP 2$            ;NO GO BACK
137
138          ;AT END OF TRACK LIST NEXT GROUP OF WRITES
139
140 025150 005737 002756          18$: TST FADJ        ;FIRST SET?
141 025154 001403          BEQ 15$           ;NO, CONTINUE
142 025156 005037 002756          CLR FADJ         ;YES, CLEAR FIRST
143 025162 000421          BR 17$           ;EXIT
144 025164 005737 003106          15$: TST ADJTRK     ;DONE BOTH INSIDE OUTSIDE
145 025170 001004          BNE 16$           ;TRACKS, YES 16$
146 025172 005237 003106          INC ADJTRK       ;NO, SET INSIDE FLAG
147 025176 000137 024332          JMP 21$          ;GO DO INSIDE TRACK
148 025202 005037 003106          16$: CLR ADJTRK     ;BACK TO OUTSIDE TRACK
149 025206 005237 003110          INC ADJUUT       ;DONE WITH ANOTHER
150 025212 023737 003110 003022      CMP ADJUUT,UUT    ;DONE TABLE FOR ALL UUT?
151 025220 001402          BEQ 17$           ;YES, FOR EXIT
152 025222 000137 024332          JMP 21$          ;NO, GO BACK FOR NEXT
153 025226 005725          17$: TST (R5)+     ;BUMP EXIT TO END OF
154 025230 001376          BNE 17$           ;TABLE FOR PROPER RETURN
155 025232 005037 002776          CLR ADJDIR       ;EXIT
156 025236 000205          RTS R5

```

1

```

1          ;ROUTINE TO SEEK TO A DESIRED CYLINDER
2          ;CALL: JSR    R5,SKCYL
3          ;ROUTINE HAS DESIRED CYLINDER IN LOC "CYL"
4          :
5          :
6 025240 010146          SKCYL: MOV    R1,-(SP)          ;SAVE R1
7 025242 004537 031360 90$: JSR    R5,LDFUNC          ;GET PRESENT POSITION
8 025246 000010          RDHDR
9          :
10 025250 005737 002766          TST    ERFLG          ;ERROR FLAG SET
11 025254 001104          BNE    5$          ;YES, SKIP
12          :
13 025256 005001          CLR    R1
14 025260 012737 000007 002750 MOV    #7,TEM
15 025266 053701 003014          BIS    CYL,R1          ;GET THE SELECTED CYLINDER NUMBER
16          :
17 025272 006301          120$: ASL    R1
18 025274 005337 002750          DEC    TEM
19 025300 001374          BNE    120$
20 025302 042737 000177 003064 BIC    #177,F.MP
21 025310 163701 003064          SUB    E.MP,R1          ;CALCULATE DIFFERENCE WORD
22 025314 103002          BCC    1$          ;IF POSITIVE SET DIRECTION
23 025316 005401          NEG    R1          ;NEGATE
24 025320 000402          BR    2$          ;SKIP SETTING DIRECTION
25 025322 052701 000004          1$: BIS    #SIGN,R1          ;SET FOR FORWARD SEEK
26 025326 052701 000001          2$: BIS    #MK,R1          ;SET MARKER BIT
27 025332 005737 003012          TST    SURF
28 025336 001402          BEQ    3$          ;TOP
29 025340 052701 000020          BIS    #SKHS,R1          ;BOTTOM
30 025344 010137 003076          3$: MOV    R1,BDA          ;LOAD DIFFERENCE WORD
31 025350 004537 031360          JSR    R5,LDFUNC          ;EXECUTE SEEK
32 025354 000006          SEEK
33          :
34 025356 005737 002766          TST    ERFLG          ;ERROR?
35 025362 001041          BNE    5$          ;YES, SKIP
36          :
37 025364 004537 031360          JSR    R5,LDFUNC          ;VERIFY POSITION?
38 025370 000010          RDHDR
39 025372 005737 002766          TST    ERFLG
40 025376 001033          BNE    5$
41 025400 042737 000077 003064 BIC    #77,E.MP          ;VERIFY POSITION
42 025406 005001          CLR    R1
43 025410 012737 000007 002750 MOV    #7,TEM
44 025416 053701 003014          BIS    CYL,R1
45 025422 006301          220$: ASL    R1
46 025424 005337 002750          DEC    TEM
47 025430 001374          BNE    220$
48 025432 005737 003012          TST    SURF
49 025436 001402          BEQ    4$
50 025440 052701 000100          BIS    #HEAD,R1
51 025444 020137 003064          4$: CMP    R1,E.MP
52 025450 001414          BEQ    6$
53          :
54 025452          ERDF  12,,SKER,ERR6          ;SEEK ERROR
   025452          TRAP  T$ERRCODE
   025454          .WORD 12
   025456          .WORD SKFR

```

```

55 025460 020524
56 025462 000137 025242
57 025466 104462
025466 000015
025470 017667
025474 020464
58 025476 000137 025242
59 025502 012601
60 025504 000205

      .WORD  FRR6
      JMP    908

58:   ERRDF  13.,FUNERR,ERR5 ;FUNCTION ERROR IN SEEK
      TRAP  T$ERCODE
      .WORD '3
      .WORD FUNE PR
      .WORD ERR5
      JMP   908

68:   MOV    (SP)+,R1      ;CANT GET THERE
      RTS   R5            ;EXIT

```

```

1      ;ROUTINE TO PERFORM REGISTER PRINTOUT DUMP
2      :CALL: JSR PC,REGDMP
3      REGDMP: PRINTB #FRM12,BCS,PBA,BDA,BMP
          MOV BMP,-(SP)
          MOV BDA,-(SP)
          MOV BBA,-(SP)
          MOV BCS,-(SP)
          MOV #FRM12,-(SP)
          MOV #5,-(SP)
          MOV SP,R0
          EMT ($PNTB
          ADD #14,SP
4      PRINTB #FRM13,E.CS,F.BA,E.DA,E.MP
          MOV E.MP,-(SP)
          MOV E.DA,-(SP)
          MOV E.BA,-(SP)
          MOV E.CS,-(SP)
          MCV #FRM13,-(SP)
          MOV #5,-(SP)
          MOV SP,R0
          EMT ($PNTB
          ADD #14,SP
5      025606 032737 040000 003056 BIT #BIT14,E.CS
6      025614 001437 BEQ 1$
7      025616 016403 000000 MOV (SR(R4),R3
8      025622 012763 000013 000004 MOV #13,DA(R3)
9      025630 012737 000004 003072 MOV #4,BCS
10     025636 056437 000004 003072 BIS DSB(R4),BCS
11     025644 013763 003072 00C000 MOV BCS,CS(R3)
12     025652 032763 000200 000000 2$: BIT #200,CS(R3)
13     025660 001774 BEQ 2$
14     025662 016337 000006 003000 MOV MP(R3),DRSTAT
15     025670 PRINTB #FRM14,DRSTAT
          MOV DRSTAT,-(SP)
          MOV #FRM14,-(SP)
          MOV #2,-(SP)
          MOV SP,R0
          EMT ($PNTB
          ADD #6,SP
16     025714 000207 1$: RTS PC

```



```

1 1 ;ROUTINE TO STORE OR RETRIEVE ADJACENT CYLINDER SECTOR DRIVE
2 ;INFORMATION FROM THE 24X5 "SECLST" BUFFER.
3 ;ENTER WITH R0 = SECTOR REQUEST
4 ;EXIT WITH R0 = ADJACENT CYLINDER DRIVE INFORMATION FOR SECTOR
5 ;EXIT WITH R0 = 0 IF SECTOR REQUESTED IS NOT IN BUFFER MAP
6 ;CALL 1: JSR R5,RSADJS
7 ; ;RETRIEVE SECTOR INFO.
8 ;CALL 2: JSR R5,RSADJS
9 ; ;STORE SECTOR INFO.
10 025716 010146 RSADJS: MOV R1,-(SP)
11 025720 010246 MOV R2,-(SP)
12 025722 010346 MOV R3,-(SP)
13 025724 042700 177700 BIC #177700,R0 ;SAVE SECTOR BITS
14 025730 012537 002774 MOV (R5)+,ADJFLG ;SAVE RETRIEVE/STORE FLAG
15 025734 012701 000001 MOV #1,R1 ;START WITH TRACK (N-2)
16 025740 012702 002334 MOV #SECBUF,R2 ;START OF 24X5 BUFFER
17 025744 012703 000020 MOV #16.,R3 ;SECTOR 16 START FOR (N-2) TRACK
18 025750 123701 002772 1$: CMPB ADJLOC,R1 ;CHECK TRACK INDEX
19 025754 001413 BEQ 2$ ;
20 025756 005201 INC R1 ;INDEX TRACK REFERENCE
21 025760 062702 000060 ADD #48.,R2 ;UPDATE BUFFER TO NEXT TRACK REF.
22 025764 062703 000042 ADD #34.,R3 ;UPDATE SECTOR START FOR NEXT TRACK
23 025770 020327 000050 CMP R3,#40.
24 025774 002765 BLT 1$
25 025776 162703 000050 SUB #40.,R3
26 026002 000762 BR 1$ ;
27 026004 012701 000030 2$: MOV #24.,R1 ;SET COUNTER FOR 24 SECTORS
28 026010 020003 3$: CMP R0,R3 ;COMPARE SECTOR TO SECTOR TABLE
29 026012 001413 BEQ 5$ ;YES,STORE OR RETRIEVE SECTOR INFO.
30 026014 005722 TST (R2)+ ;INDEX SECLST BUFFER IN WORD FORMAT
31 026016 005203 INC R3 ;INDEX SECTOR COUNT
32 026020 020327 000047 CMP R3,#39. ;COMPARE SECTOR COUNT FOR <40
33 026024 003402 BLE 4$
34 026026 162703 000050 SUB #40.,R3 ;KEEP SECTOR COUNT<40
35 026032 005301 4$: DEC R1 ;PASSED 24 SECTORS?
36 026034 001365 BNE 3$ ;COMPARE NEXT SECTOR
37 026036 005000 CLR R0 ;SETUP R0 FOR EXIT
38 026040 000405 BR 7$ ;EXIT ROUTINE,SECTOR NOT FOUND
39 026042 005737 002774 5$: TST ADJFLG ;FLAG=0 FOR RETRIEVE
40 026046 001401 BEQ 6$
41 026050 010412 MOV R4,(R2) ;STORE DRIVE INFO. INTO BUFFER
42 026052 011200 6$: MOV (R2),R0 ;SAVE DRIVE INFO. INTO R0 FOR EXIT
43 026054 012603 7$: MOV (SP)+,R3
44 026056 012602 MOV (SP)+,R2
45 026060 012601 MOV (SP)+,R1
46 026062 000205 RTS R5 ;EXIT

```

```

1          ;ROUTINE TO SET DRIVE IN SECTOR LIST
2          ;CALL: JSR   R5,SETLST      ;RO HAS SECTOR
3          ;DRIVE GOTTEN FROM R4
4
5 026064 010146  SETLST: MOV    R1,-(SP)      ;SAVE R1
6
7 026066 162700 000034      SUB    #28.,R0      ;START LIST AT 0
8 026072 100002      BPL    3$
9 026074 062700 000050      ADD    #40.,R0
10 026100 012701 002274  3$:  MOV    #SECLST,R1      ;BEGINNING OF SECTOR LIST
11 026104 005700      1$:  TST    R0      ;FOUND SECTOR?
12 026106 001403      BEQ    2$      ;BRANCH IF YES
13 026110 005300      DEC    R0      ;DECREMENT SECTOR
14 026112 005721      TST    (R1)+      ;NEXT ENTRY IN LIST
15 026114 000773      BR     1$      ;GO BACK
16 026116 010411  2$:  MOV    R4,(R1)      ;STORE DRIVE BITS IN LIST
17 026120 012601      MOV    (SP)+,R1      ;RESTORE R1
18 026122 000205      RTS    R5
19
20          ;ROUTINE TO LOCATE DRIVE THAT WROTE SECTOR LAST
21          ;CALL: JSR   R5,FNDDRV      ;RO-CONTAINS SECTOR
22          ;ON EXIT RO-DRIVE
23
24 026124 010146  FNDDRV: MOV    R1,-(SP)      ;SAVE R1
25 026126 162700 000034      SUB    #28.,R0      ;START LIST AT 0
26 026132 100002      BPL    3$
27 026134 062700 000050      ADD    #40.,R0
28 026140 012701 002274  3$:  MOV    #SECLST,R1      ;START OF LIST
29 026144 005700      1$:  TST    R0      ;FOUND SECTOR?
30 026146 001403      BEQ    2$      ;YES, GET DRIVE #, EXIT
31 026150 005300      DEC    R0      ;NO, DOWN COUNT SECTOR
32 026152 005721      TST    (R1)+      ;NEXT ENTRY IN LIST
33 026154 000773      BR     1$      ;GO BACK
34 026156 011100  1$:  MOV    (R1),R0      ;GET DRIVE BUFFER POINTER
35 026160 012601      MOV    (SP)+,R1      ;RESTORE R1
36 026162 000205      RTS    R5      ;EXIT

```

```

1
2
3
4
5
6
7
8
9 026164 010046
10 026166 010146
11 026170 010246
12 026172 012737 000034 003024
13 026200 012701 100000
14 026204 016437 000006 003030
15
16 026212 012737 177600 003100 1$: MOV #-128.,BMP ;SET UP READ-ONE SECTOR
17 026220 012737 003124 003074 MOV #BUF,BBA ;BUS ADDRESS
18 026226 042737 000077 003076 2$: BIC #77,BDA ;CLEAR OUT SECTOR BITS
19 026234 053737 003024 003076 BIS SECT,BDA ;SET SECTOR
20 026242 030103 BIT R1,R3 ;DO WE READ THIS ONE?
21 026244 001521 BEQ 5$ ;NO, BRANCH
22 026246 004537 031360 JSR R5,LDFUNC ;READ
23 026252 000014 READ
24
25 026254 005737 003056 TST E.CS ;ERROR
26 026260 100107 BPL 4$ ;NO CONTINUE
27
28 026262 005737 002754 TST F0WR ;INITIAL WRITE
29 026266 001412 BEQ 21$ ;NO
30 026270 012737 017213 002764 MOV #INITWR,REASON ;SETUP INITIAL WRITE OF SECTOR
31 026276 016437 000000 002762 MOV (SR(R4),LSTCLR
32 026304 016437 000005 003026 MOV DSB+1(R4),LSTDRV
33 026312 000415 BR 22$
34 026314 012737 017462 002764 21$: MOV #OVMS,REASON ;SET MESSAGE FOR OVERWRITE
35 026322 013700 003024 MOV SECT,R0 ;FIND DRIVE THAT LAST WROTE
36 026326 004537 026124 JSR R5,FNDDRV ;SECTOR
37 026332 016037 000000 002762 MOV (SR(R0),LSTCLR ;GET IT'S CSR
38 026340 116037 000005 003026 MOV DSB+1(R0),LSTDRV ;GET THE DRIVE
39 026346 104462 22$: ERDF 13.,OVWR,ERR4 ;PRINT ERROR
026346 TRAP T$RCODE
026350 .WORD 13
026352 .WORD OVWR
026354 .WORD ERR4
40 026356 005037 003034 CLR WCOUNT ;CLEAR BAD WORD COUNT W/IN SECTOR
41 026362 005037 003036 CLR SECRD ;CLEAR WORD IN SECTOR
42 026366 012702 003124 MOV #BUF,R2 ;GET BUFFER START
43 026372 023712 003030 3$: CMP GDATA,(R2) ;IS DATA CORRECT?
44 026376 001417 BEQ 31$ ;YES CHECK NEXT
45 026400 005237 003034 INC WCOUNT ;NO ACCOUNT FOR IT
46 026404 PRINTF #FRMB,SECRD,GDATA,(R2)
026404 MOV (R2),-(SP)
026406 013746 003030 MOV GDATA,-(SP,
026412 013746 003036 MOV SECRD,-(SP,
026416 012746 021257 MOV #FRMB,-(SP)
026422 012746 000004 MOV #4,-(SP)
026426 010600 MOV SP,R0
026430 104017 EMT ($PNT)

```

ROUTINE TO VERIFY THAT THE OVERWRITE DID ACTUALLY OVERWRITE THE
PREVIOUS DATA ON THE PACK.

CALL: JSR R5,VEROW USES R3 AS BIT MAP OF SECTORS TO
CHECK. R3 IS LOADED PRIOR TO
WRITING SECTORS.

VEROW: MOV R0,-(SP) ;SAVE REGISTER CONTENTS
MOV R1,-(SP)
MOV R2,-(SP)
MOV #28.,SECT ;START VERIFY AT SECTOR 28
MOV #100000,R1 ;BIT MASK FOR VERIFICATION
MOV PAT(R4),GDATA ;GET PATTERN FOR THIS DRIVE

1\$: MOV #-128.,BMP ;SET UP READ-ONE SECTOR
MOV #BUF,BBA ;BUS ADDRESS
2\$: BIC #77,BDA ;CLEAR OUT SECTOR BITS
BIS SECT,BDA ;SET SECTOR
BIT R1,R3 ;DO WE READ THIS ONE?
BEQ 5\$;NO, BRANCH
JSR R5,LDFUNC ;READ
READ

TST E.CS ;ERROR
BPL 4\$;NO CONTINUE

TST F0WR ;INITIAL WRITE
BEQ 21\$;NO
MOV #INITWR,REASON ;SETUP INITIAL WRITE OF SECTOR

MOV (SR(R4),LSTCLR
MOV DSB+1(R4),LSTDRV
BR 22\$
21\$: MOV #OVMS,REASON ;SET MESSAGE FOR OVERWRITE
MOV SECT,R0 ;FIND DRIVE THAT LAST WROTE
JSR R5,FNDDRV ;SECTOR
MOV (SR(R0),LSTCLR ;GET IT'S CSR
MOV DSB+1(R0),LSTDRV ;GET THE DRIVE
22\$: ERDF 13.,OVWR,ERR4 ;PRINT ERROR

TRAP T\$RCODE
.WORD 13
.WORD OVWR
.WORD ERR4

CLR WCOUNT ;CLEAR BAD WORD COUNT W/IN SECTOR
CLR SECRD ;CLEAR WORD IN SECTOR
MOV #BUF,R2 ;GET BUFFER START
3\$: CMP GDATA,(R2) ;IS DATA CORRECT?
BEQ 31\$;YES CHECK NEXT
INC WCOUNT ;NO ACCOUNT FOR IT

PRINTF #FRMB,SECRD,GDATA,(R2)
MOV (R2),-(SP)
MOV GDATA,-(SP,
MOV SECRD,-(SP,
MOV #FRMB,-(SP)
MOV #4,-(SP)
MOV SP,R0
EMT (\$PNT)

```

47 026432 062706 000012          ADD    #12,SP
48 026436 005722          31$:  TST    (R2)+          ;NEXT
49 026440 005237 003036          INC    SECWRD          ;NEXT
50 026444 023727 003036 000200  CMP    SECWRD,#128.    ;DONE WITH SECTOR?
51 026452 001347          BNE    3$              ;NO GO BACK
52
53 026454          PRINTF #FRM9,WCOUNT    ;PRINT SUMMARY
    026454 013746 003034      MOV    WCOUNT,-(SP)
    026460 012746 021323      MOV    #FRM9,-(SP)
    026464 012746 000002      MOV    #2,-(SP)
    026470 010600          MOV    SP,R0
    026472 104017          EMT    (SPNTF
    026474 062706 000006      ADD    #6,SP
54
55 026500 013700 003024          4$:  MOV    SECT,R0          ;SET SECTOR IN LIST TO THE
56 026504 004537 026064          JSR    R5,SETLST      ;CREDIT OF THIS DRIVE
57
58 026510 005237 003024          5$:  INC    SECT              ;NEXT SECTOR
59 026514 023727 003024 000050  CMP    SECT,#40.
60 026522 001003          BNE    6$
61 026524 162737 000050 003024  SUB    #40.,SECT
62 026532 000241          6$:  CLC                    ;CLEAR CARRY
63 026534 006001          ROR    R1              ;NEXT BIT
64 026536 103225          BCC    1$              ;IF CLEAR NEXT
65
66 026540 012602          MOV    (SP)+,R2        ;RESTORE R2-R0, EXIT
67 026542 012601          MOV    (SP)+,R1
68 026544 012600          MOV    (SP)+,R0
69 026546 000205          RTS    R5

```

```

1          ;ROUTINE TO VERIFY THAT A DRIVE CAN RECOVER ANOTHER DRIVE'S DATA.
2          ;
3          ;CALL: JSR      R5,VEROD      USES R3 AS BIT MAP OF SECTORS TO
4          ;                   CHECK.  R3 IS LOAD BY WRSEC (WE
5          ;                   USE R3 COMPLIMENTED.
6          ;
7          ;
8 026550 010046 VEROD: MOV      R0,-(SP)      ;SAVE R0-R2
9 026552 010146      MOV      R1,-(SP)
10 026554 010246      MOV      R2,-(SP)
11 026556 012701 100000      MOV      #100000,R1      ;BIT MASK FOR SECTORS
12 026562 012737 000034 003024      MOV      #28.,SECT      ;START WITH SECTOR 28
13 026570 005737 002754      TST      F0WR      ;CHECK FOR FIRST OVERWRITE
14 026574 001134      BNE      6$
15
16 026576 012737 177600 003100 1$: MOV      #-128.,BMP      ;SET UP READ (ONE SECTOR)
17 026604 012737 003124 003074      MOV      #BUF,BBA      ;BUS ADDRESS
18 026612 042737 000077 003076 2$: BIC      #77,BDA      ;CLEAR SECTOR BITS
19 026620 053737 003024 003076      BIS      SECT,BDA      ;SET IN SECTOR BITS
20 026626 030103      BIT      R1,R3      ;CHECK THIS SECTOR?
21 026630 001103      BNE      5$      ;NO BRANCH
22
23 026632 013700 003024      MOV      SECT,R0      ;FIND DRIVE THAT WROTE
24 026636 004537 026124      JSR      R5,FNDDRV      ;SECTOR LAST
25 026642 016037 000000 002762      MOV      CSR(R0),LSTCLR ;GET CSR OF DRIVE
26 026650 116037 000005 003026      MOV      DSB+1(R0),LSTDRV ;GET DRIVE
27 026656 016037 000006 003030      MOV      PAT(R0),GDATA ;GET PATTERN
28
29 026664 004537 031360      JSR      R5,LDFUNC      ;READ
30 026670 000014      READ
31
32 026672 005737 003056      TST      E.CS      ;ERROR?
33 026676 100060      BPL      5$      ;NO, NEXT SECTOR
34 026700 012737 017515 002764      MOV      #RECMS,REASON ;SET READ RECOVERY MESSAGE
35 026706      ERRDF 14.,RECER,ERR4 ;REPORT ERROR
    026706 104462      TRAP  T$ERCODE
    026710 000016      .WORD 14
    026712 017643      .WORD RECER
    026714 020304      .WORD ERR4
36
37 026716 005037 003034      CLR      WCOUNT      ;CLEAR BAD WORD COUNT
38 026722 005037 003036      CLR      SECWRD      ;CLEAR WORD W/I SECTOR
39 026726 012702 003124      MOV      #BUF,R2      ;START OF BUFFER
40 026732 023712 003030 3$: CMP      GDATA,(R2)      ;DATA COMPARE
41 026736 001417      BEQ      4$      ;YES, CHECK NEXT
42
43 026740 005237 003034      INC      WCOUNT      ;ACCOUNT FOR ERROR
44 026744      PRINTF #FRMB,SECWRD,GDATA,(R2) ;PRINT ERROR
    026744 011246      MOV      (R2),-(SP)
    026746 013746 003030      MOV      GDATA,-(SP)
    026752 013746 003036      MOV      SECWRD,-(SP)
    026756 012746 021257      MOV      #FRMB,-(SP)
    026762 012746 000004      MOV      #4,-(SP)
    026766 010600      MOV      SP,R0
    026770 104017      EMT      C$PNTF
    026772 062706 000012      ADD      #12,SP

```

```
46 026776 005722          4$:  TST      (R2)+      ;NEXT
47 027000 005237 003036    INC      SECWRD     ;NEXT WORD IN SECTOR
48 027004 023727 003036 000200  CMP      SECWRD,#128. ;DONE?
49 027012 001347          BNE      3$         ;NO
50 027014          PRINTF #FRM9,WCOUNT ;PRINT SUMMARY
    027014 013746 003034    MOV      WCOUNT,-(SP)
    027020 012746 021323    MOV      #FRM9,-(SP)
    027024 012746 000002    MOV      #2,-(SP)
    027030 010600          MOV      SP,R0
    027032 104017          EMI      C$PNTF
    027034 062706 000006    ADD      #6,SP

51
52 027040 005237 003024    5$:  INC      SECT      ;NEXT SECTOR
53 027044 023727 003024 000050  CMP      SECT,#40.
54 027052 001002          BNE      7$
55 027054 005037 003024    CLR      SECT
56 027060 000241          7$:  CLC
57 027062 006001          ROR      R1          ;NEXT BIT MAP
58 027064 103244          BCC      1$
59
60 027066 012602          6$:  MOV      (SP)+,R2    ;RESTORE R2-R0, EXIT
61 027070 012601          MOV      (SP)+,R1
62 027072 012600          MOV      (SP)+,R0
63 027074 000205          RTS      R5
```

```

1          ;ROUTINE TO VERIFY THE ADJ. CYL. WRITE IS GOOD
2          ;USES R3 AND WORD FOLLOWING CALL
3          ;IF WRITE WAS GOOD,SECTOR WILL BE STORED IN MAP
4          ;USING RSADJS/.WORD 1
5
6 027076 010046          VAJWR:  MOV    R0,-(SP)          ;SAVE REGISTERS
7 027100 010146          MOV    R1,-(SP)
8 027102 010246          MOV    R2,-(SP)
9 027104 012701 100000   MOV    #100000,R1      ;BIT MASK FOR CYLINDER
10 027110 012502          MOV    (R5)+,R2        ;STARTING SECTOR
11 027112 005000          CLR    R0
12 027114 053700 003014   BIS    CYL,R0
13 027120 012737 000007 002750   MOV    #7,TEM
14
15 027126 006300          2$:   ASL    R0
16 027130 005337 002750   DEC    TEM
17 027134 001374          BNE    2$
18 027136 005737 003012   TST    SURF
19 027142 001402          BEQ    3$
20 027144 052700 000100   BIS    #HEAD,R0
21 027150 050200          3$:   BIS    R2,R0
22 027152 030103          4$:   BIT    R1,R3
23 027154 001462          BEQ    5$
24 027156 012737 177600 003100   MOV    #-128.,BMP
25 027164 010037 003076   MOV    R0,BDA
26 027170 010037 002760   MOV    R0,TEMP
27 027174 042700 177700   BIC    #177700,R0
28 027200 020027 000047   CMP    R0,#39.
29 027204 003406          BLE    6$
30 027206 162737 000050 003076   SUB    #40.,BDA
31 027214 162737 000050 002760   SUB    #40.,TEMP
32 027222 012737 003124 003074 6$:   MOV    #BUF,BBA
33 027230 005037 003002   CLR    HSFLG
34 027234 013700 002760   MOV    TEMP,R0
35 027240 004537 031360   10$:  JSR    R5,LDFUNC      ;READ FUNCTION
36 027244 000014          READ
37 027246 005737 002766   TST    ERFLG
38 027252 001416          BEQ    7$
39 027254 005737 003002   TST    HSFLG
40 027260 001007          BNE    11$
41 027262          ERRSOFT 120.,READ1,ERR2
   027262 104464          TRAP  T$ERCODE
   027264 000170          .WORD 120
   027266 020003          .WORD READ1
   027270 020116          .WORD ERR2
42 027272 005237 003002   INC    HSFLG
43 027276 000760          BR    10$
44 027300          11$:  ERRHRD 130.,RFAD1,ERR2
   027300 104463          TRAP  T$ERCODE
   027302 000202          .WORD 130
   027304 020003          .WORD READ1
   027306 020116          .WORD ERR2
45 027310 010046          7$:   MOV    R0,-(SP)
46 027312 004537 025716   JSR    R5,RSADJS      ;STORE ADJ. CYL. SECTOR INFO.
47 027316 000001          .WORD 1
48 027320 012600          MOV    (SP)+,R0      ;RESTORE R0
49 027322 005200          5$:   INC    R0

```

50	027324	000241	CLC	
51	027326	006001	ROR	R1
52	027330	103310	BCC	48
53	027332	012602	MOV	(SP)+,R2
54	027334	012601	MOV	(SP)+,R1
55	027336	012600	MOV	(SP)+,R0
56	027340	000205	RTS	R5


```

1          ;ROUTINE TO VERIFY THAT WRITE DID NOT DISTURB ADJACENT TRACKS
2          ;WRITTEN BY OTHER DRIVES.
3          ;CALL JSR R5,BSVWR
4          ;          .WORD          ;STARTING SECTOR
5          ;
6          ;USES "ADJLOC" TO GET +1/-1 CYLINDER OFFSET
7          ;USES R3 FOR SECTOR MAP, USES MAP AT "SECBUF" FOR INFO
8
9 027342 010046      BSVWR:  MOV     R0,-(SP)          ;SAVE REGISTERS
10 027344 010146      MOV     R1,-(SP)
11 027346 010246      MOV     R2,-(SP)
12 027350 013746 003014  MOV     CYL,-(SP)
13 027354 013746 003012  MOV     SURF,-(SP)
14 027360 012546      MOV     (R5)+,-(SP)      ;GET STARTING SECTOR
15 027362 123727 002772 000003  CMPB   ADJLOC,#3      ;ON MIDDLE TRACK???
16 027370 001455      BEQ    BSEXIT        ;YES, THEN NO CHECK
17 027372 162716 000042      SUB     #34.,(SP)      ;SETUP SECTOR START FOR OUTSIDE
18 027376 100002      BPL    1$           ;IF POSITIVE OKAY ELSE FIX
19 027400 062716 000050      ADD     #40.,(SP)      ;FIX IT
20 027404 123727 002772 000001 1$:  CMPB   ADJLOC,#1      ;ON OUTER LIMIT???
21 027412 001412      BEQ    INAWR        ;YES,SKIP CHECK
22 027414 105337 002772      DECB   ADJLOC        ;OUTER ADJ TRACK
23 027420 005337 003014      DEC     CYL
24 027424 004537 027552      JSR    R5,CHECK      ;GO CHECK ADJ SECTORS
25 027430 005237 003014      INC     CYL          ;FIX BACK
26 027434 105237 002772      INCB   ADJLOC
27 027440 062716 000104      INAWR: ADD     #68.,(SP)      ;INNER SECTOR START
28 027444 021627 000050      CMP    (SP),#40.     ;WITHIN LIMITS???
29 027450 002407      BLT    1$           ;YES, OKAY
30 027452 162716 000050      SUB     #40.,(SP)     ;FIX SECTOR
31 027456 021627 000050      CMP    (SP),#40.
32 027462 002402      BLT    1$
33 027464 162716 000050      SUB     #40.,(SP)
34 027470 123727 002772 000005 1$:  CMPB   ADJLOC,#5      ;INNER LIMIT??
35 027476 001412      BEQ    BSEXIT        ;YES,SKIP CHECK
36 027500 105237 002772      INCB   ADJLOC        ;FIX FOR INNER
37 027504 005237 003014      INC     CYL
38 027510 004537 027552      JSR    R5,CHECK      ;GO CHECK ADJ SECTORS
39 027514 105337 002772      DECB   ADJLOC        ;FIX BACK
40 027520 005337 003014      DEC     CYL
41 027524 005726      BSEXIT: TST   (SP)+    ;THROW OFF SECTOR
42 027526 012637 003012      MOV    (SP)+,SURF
43 027532 012637 003014      MOV    (SP)+,CYL
44 027536 012602      NCHECK: MOV   (SP)+,R2
45 027540 012601      MOV   (SP)+,R1
46 027542 012600      MOV   (SP)+,R0
47 027544 004537 025240      JSR   R5,SKCYL      ;SEEK BACK
48 027550 000205      R.J          ;RETURN

```

```

1          ;ROUTINE TO VERIFY AN ADJACENT SECTOR
2          ;CALLED FROM BSVWR
3          ;
4
5 027552 012701 100000          CHECK:  MOV    #100000,R1          ;SECTOR MASK
6 027556 004537 025240          JSR    R5,SKCYL          ;GET TO DESIRED CYLINDER
7 027562 005002                  CLR    R2                  ;CREATE ADDRESS
8 027564 053702 003014          BIS    CYL,R2
9 027570 012737 000007 002750  MOV    #7,TEM
10 027576 006302                2$:  ASL    R2
11 027600 005337 002750          DEC    TEM
12 027604 001374                BNE    2$
13 027606 005737 003012          TST    SURF
14 027612 001402                BEQ    3$                  ;NO
15 027614 052702 000100          BIS    #HEAD,R2
16 027620 056602 000002          3$:  BIS    2(SP),R2          ;SET IN SECTOR
17 027624 030103                4$:  BIT    R1,R3            ;THIS SECTOR IN LIST???
18 027626 001452                BEQ    5$                  ;NO, NEXT
19 027630 010200                MOV    R2,R0              ;COPY SECTOR
20 027632 042700 177700          BIC    #177700,R0         ;ONLY SECTOR LEFT
21 027636 020027 000050          CMP    R0,#40.           ;SECTOR OKAY???
22 027642 002404                BLT    6$                  ;YES
23 027644 162700 000050          SUB    #40.,R0
24 027650 162702 000050          SUB    #40.,R2            ;FIX SECTOR
25 027654 004537 025716          6$:  JSR    R5,RSADJS        ;FIND IF SECTOR PREVIOUSLY WRITTEN
26 027660 000000                .WORD 0
27 027662 005700                TST    R0                  ;WAS IT??
28 027664 001433                BEQ    5$                  ;NO
29 027666 010237 003076          MOV    R2,BDA            ;LOAD DISK ADDRESS
30 027672 012737 177600 003100  MOV    #-128.,BMP        ;LOAD WC
31 027700 004537 031360          JSR    R5,LDFUNC         ;LOAD
32 027704 000014                READ
33 027706 005737 002766          TST    ERFLG             ;WAS READ GOOD
34 027712 001420                BEQ    5$
35
36 027714 010346                MOV    R3,-(SP)
37 027716 010237 003024          MOV    R2,SECT
38 027722 010003                MOV    R0,R3
39
40 027724 042737 177700 003024  BIC    #177700,SECT
41 027732                ERRHRD 140.,ADJTXT,ERR3
   027732 104463                TRAP  T$ERCODE
   027734 000214                .WORD 140
   027736 020030                .WORD ADJTXT
   027740 020156                .WORD ERR3
42 027742 012603                MOV    (SP)+,R3
43 027744                ERRHRD 110.,READ1,ERR2
   027744 104463                TRAP  T$ERCODE
   027746 000156                .WORD 110
   027750 020003                .WORD READ1
   027752 020116                .WORD ERR2
44
45 027754 005202                5$:  INC    R2                  ;NEXT SECTOR
46 027756 000241                CLC
47 027760 006001                ROR    R1                  ;SHIFT MASK
48 027762 103320                BCC    4$
49 027764 000205                RTS    R5

```

```

1          :ROUTINE TO MERGE BAD SECTOR FILES
2          :ENTRY INTO THIS ROUTINE WILL OCCUR AFTER THE "SERNUM" ROUTINE
3          :IS PERFORMED. THE FACTORY BAD SECTOR FILE WILL BE LOCATED IN
4          :FIRST 400(8) LOCATIONS.
5          :THIS ROUTINE WILL STORE THE FIELD BAD SECTORS INTO THE NEXT
6          :400 LOCATIONS AND THEN MERGE THE FACTORY BAD FILE
7          :WITH THE FIELD BAD FILE.
8
9          :FACTORY BAD AT BUF
10         :FIELD BAD AT BUF + 512.
11
12 027766 010146      MERGE:  MOV    R1,-(SP)          ;SAVE R1, R2, R3
13 027770 010246      MOV    R2,-(SP)
14 027772 010346      MOV    R3,-(SP)
15 027774 012737 003524 003074  MOV    #BUF+400,BBA      ;BUFFER START FOR FIELD BAD
16 030002 022737 000001 002752  CMP    #1,T.DRIVE
17 030010 001004      BNE    55$
18 030012 012737 077724 003076  MOV    #77724,BDA
19 030020 000403      BR     66$
20 030022 012737 177724 003076 55$:  MOV    #177724,BDA
21
22 030030 012737 177400 003100 66$:  MOV    #-256.,BMP
23 030036 004537 031360      97$:  JSR    R5,LDFUNC      ;LOAD READ FUNCTION
24 030042 000014      READ
25 030044 005737 002766      TST    ERFLG          ;TEST ERROR FLAG
26 030050 001431      BEQ    98$          ;YES;MERGE BAD SECTOR FILES
27 030052 062737 000004 003076  ADD    #4,BDA        ;TRY NEXT FIELD BAD SECTOR FILE
28 030060 022737 000001 002752  CMP    #1,T.DRIVE
29 030066 001004      BNE    400$
30 030070 022737 077750 003076  CMP    #77750,BDA
31 030076 001357      BNE    97$
32
33 030100 022737 177750 003076 400$:  CMP    #177750,BDA
34 030106 001353      BNE    97$          ;NO,DO NEXT FIELD BAD SECTOR
35 030110      PRINTF #FRM15
   030110 012746 021657      MOV    #FRM15,-(SP)
   030114 012746 000001      MOV    #1,-(SP)
   030120 010600      MOV    SP,R0
   030122 104017      EMT   C$PNTF
   030124 062706 000004      ADD    #4,SP
36 030130      999$:  BREAK
   030130 104022      EMT   C$BRK
37 030132 000776      BP     999$
38 030134 012701 003134      98$:  MOV    #BUF+10,R1      ;GET PAST ID ETC.
39 030140 012702 000176      MOV    #126.,R2      ;MAX = 126
40 030144 005721      1$:  TST   (R1)+          ;SECTOR OR END
41 030146 100404      BMI   2$          ;END, GO GET FIELD
42 030150 005721      TST   (R1)+          ;REST OF SECTOR
43 030152 005302      DEC   R2          ;MAX REACHED
44 030154 001373      BNE   1$          ;NO, KEEP GOING
45 030156 000401      RR    3$          ;YES, SKIP BACK UP
46 030160 005741      2$:  TST   -(R1)        ;BACK UP PAST TERMINATOR
47 030162 012703 000176      3$:  MOV    #126.,R3      ;SET 126 MAX
48 030166 012702 003534      MOV    #BUF+410,R2   ;GET FIELD SECTORS
49 030172 012221      4$:  MOV    (R2)+,(R1)+   ;MERGE AT END OF FACTORY
50 030174 100403      BMI   5$          ;DONE?
51 030176 012221      MOV    (R2)+,(R1)+   ;NO, MERGE REST OF SECTOR

```

```
52 030200 005303          DEC      R3          :DONE
53 030202 001373          BNE      48          :NO, GO BACK
54 030204 012603          58:  MOV     (SP)+,R3      :RESTORE R3, R2, R1
55 030206 012602          MOV     (SP)+,R2
56 030210 012601          MOV     (SP)+,R1
57 030212 000205          RTS      R5          :EXIT
```

1	030214	012537	003040		FNDTRK:	MOV	(R5)+,OFFSET	:GET INCREMENT/DECREMENT
2	030220	012537	003050			MOV	(R5)+,SURFACE	:GET HEAD (SURFACE)
3	030224	022737	000601	002752		CMP	#1,T.DRIVE	
4	030232	001001				BNE	80\$	
5	030234	000401				BR	90\$	
6	030236	022525			80\$:	CMP	(R5)+,(R5)+	
7	030240	012537	003044		90\$:	MOV	(R5)+,FRTRK	
8	030244	012537	003042			MOV	(R5)+,LSTTRK	
9	030250	005037	003052			CLR	TRKFND	:CLEAR OUT FLAG FOUND
10	030254	005037	003054			CLR	TRKCNT	:CLEAR OUT TRACK COUNT
11	030260	013737	003044	003046		MOV	FRTRK,PRSTRK	:GET FIRST TRACK
12	030266				1\$:			
13	030266	004537	030366			JSR	R5,FNDBSC	:IS TRACK IN BAD SECTOR FILE
14	030272	005737	002126			TST	HDRFND	:WAS IT?
15	030276	001003				BNE	2\$:YES, CLEAR TRKCNT
16	030300	005237	003054			INC	TRKCNT	:NO, INDICATE GOOD TRACK
17	030304	000402				BR	3\$:CONTINUE
18	030306	005037	003054		2\$:	CLR	TRKCNT	:START COUNT OVER
19	030312	023727	003054	000005	3\$:	CMP	TRKCNT,#5	:FIND 5 TRACKS YET?
20	030320	001011				BNE	4\$:NO, CONTINUE
21	030322	005237	003052			INC	TRKFND	:YES, EXIT WITH GOOD FLAG
22	030326	022737	000001	002752		CMP	#1,T.DRIVE	
23	030334	001002				BNE	81\$	
24	030336	062705	000004			ADD	#4,R5	
25								
26	030342	000205			81\$:	RTS	R5	
27	030344	023737	003046	003042	4\$:	CMP	PRSTRK,LSTTRK	:ARE WE DONE?
28	030352	001001				BNE	5\$:NO, KEEP LOOKING
29	030354	000205				RTS	R5	:EXIT WITH NOT FOUND
30	030356	063737	003040	003046	5\$:	ADD	OFFSET,PRSTRK	:NEXT TRACK
31	030364	000740				BR	1\$	

```

1          ;ROUTINE TO FIND BAD TRACK IN FILE
2
3 030366 005037 002126  FNDBSC: CLR      HDRFND      ;INITIALIZE FLAG
4 030372 010146          MOV      R1,-(SP)  ;SAVE R1, R2
5 030374 010246          MOV      R2,-(SP)
6 030376 012701 003134  MOV      #BUF+10,R1  ;SETUP FOR BEGINNING OF FILE
7 030402 005711          1$:  TST      (R1)          ;END?
8 030404 100421          BMI      2$          ;IF MINUS AT END, EXIT
9 030406 023721 003046  CMP      PRSTRK,(R1)+ ;CYLINDER CORRECT?
10 030412 001011          BNE      3$          ;NO, NEXT
11 030414 105724          TSTB     (R4)+        ;UPPER HALF OF WORD
12 030416 123711 003050  CMPB     SURFACE,(R1) ;CORRECT SURFACT
13 030422 001402          BEQ      4$          ;
14 030424 105744          TSTB     -(R4)
15 030426 000403          BR       3$          ;
16 030430 005237 002126  4$:  INC      HDRFND      ;GET FOUND
17 030434 000405          BR       2$          ;
18
19 030436 005721          3$:  TST      (R1)+        ;NEXT WORD
20 030440 005202          INC      R2          ;ACCOUNT FOR IT
21 030442 020227 000374  CMP      R2,#252.    ;DONE?
22 030446 001355          BNE      1$          ;NO, KEEP CHECKING
23 030450 012601          2$:  MOV      (SP)+,R1    ;RESTORE R2, R1, EXIT
24 030452 012602          MOV      (SP)+,R2
25 030454 000205          RTS      R5
26
27 030456 013701 003046  FIXCYL: MOV     PRSTRK,R1 ;GET TRACK WHICH IS GOOD
28 030462 005737 003040  TST     OFFSET      ;WHICH WAY WERE WE LOOKING
29 030466 100402          BMI     1$          ;IN WORD, BRANCH
30 030470 162701 000004  SUB     #4,R1        ;BACK IT UP BY FOUR
31 030474 012702 000005  1$:  MOV     #5,R2      ;GOING STORE AWAY 5 TRACKS
32 030500 010120          2$:  MOV     R1,(R0)+  ;STORE THEM 1 WD/PER
33 030502 005201          INC     R1
34 030504 005302          DEC     R2
35 030506 001374          BNE     2$
36 030510 000205          RTS     R5

```

```

1          ;ROUTINE TO GET SERIAL NUMBER
2
3          ;CALL JSR R5,SERNUM
4
5 030512 012737 000013 003076 SERNUM: MOV #13,BDA
6 030520 004537 031360 JSR R5,LDFUNC ;GET STATUS
7 030524 000004 GSTAT
8 030526 004537 031360 JSR R5,LDFUNC ;READ HEADER
9 030532 000010 RDHDR
10 030534 013700 003064 MOV E,MP,R0 ;GET THE HEADER
11 030540 042700 000077 18: BIC #77,R0 ;CLEAR SECTOR BITS
12 030544 022737 000001 002752 CMP #1,T.DRIVE
13 030552 001003 BNE 23$
14 030554 020027 077700 CMP R0,#77700
15 030560 001446 BEQ 28$
16 030562 020027 177700 23$: CMP R0,#177700
17 030566 001443 BEQ 28$
18 030570 042700 000100 BIC #100,R0 ;CLEAR HEAD
19 030574 022737 000001 002752 CMP #1,T.DRIVE
20 030602 001003 BNE 32$
21 030604 012701 077600 MOV #77600,R1
22 030610 000402 BR 33$
23 030612 012701 177600 32$: MOV #177600,R1
24
25 030616 160001 33$: SUB R0,R1
26 030620 010137 003076 MOV R1,BDA ;SET UP DIF WORD
27 030624 012737 000025 003076 BIS #25,BDA ;SEEK IN, HEAD 1
28 030632 004537 031360 JSR R5,LDFUNC ;SEEK
29 030636 000006 SEEK
30 030640 004537 031360 JSR R5,LDFUNC ;VERIFY POSITION
31 030644 000010 RDHDR
32 030646 013700 003064 MOV E,MP,R0 ;GET HEADER
33 030652 022737 000001 002752 CMP #1,T.DRIVE
34 030660 001003 BNE 42$
35 030662 022700 077700 CMP #77700,R0
36 030666 000402 BR 43$
37 030670 022700 177700 42$: CMP #177700,R0
38
39 030674 103321 43$: BHIS 18
40 030676 022737 000001 002752 28: CMP #1,T.DRIVE
41 030704 001004 BNE 52$
42 030706 012737 077700 003076 MOV #77700,BDA
43 030714 000403 BP 97$
44
45 030716 012737 177700 003076 52$: MOV #177700,BDA
46 030724 012737 003124 003074 97$: MOV #BUF,BBA
47 030732 012737 177400 003100 MOV #-256.,BMP
48 030740 004537 031360 JSR R5,LDFUNC ;READ
49 030744 000014 READ
50 030746 005737 002766 TST ERFLG ;TEST ERROR FLAG
51 030752 001421 BEQ 98$ ;YES,COMPARE SERIAL NUMBERS
52 030754 062737 000004 003076 ADD #4,BDA ;NO,SETUP FOR NEXT FACTORY BAD SECTOR
53 030762 022737 000001 002752 CMP #1,T.DRIVE
54 030770 001005 BNE 62$
55 030772 022737 077724 003076 CMP #77724,BDA
56 031000 001351 BNE 97$
57 031002 000453 BR 99$

```

```

58 031004 022737 177724 003076 62$: CMP #177724,BDA
59 031012 001344 BNE 97$ ;GET NEXT FACTORY BAD SECTOR
60 031014 000446 BR 99$ ;REPORT ERROR
61 031016 012701 003124 98$: MOV #BUF,R1 ;COMPARE SERIAL NUMBERS
62 031022 005737 003102 TST SERNM1 ;HAVE WE GOT ONE TO COMPARE
63 031026 100005 BPL 3$ ;YES, BRANCH
64 031030 011137 003102 MOV (R1),SERNM1 ;NO, CALL THIS ONE IT
65 031034 016137 000002 003104 MOV 2(R1),SERNM2 ;
66 031042 021137 003102 3$: CMP (R1),SERNM1 ;SERNUM OKAY
67 031046 001004 BNE 4$ ;NO, PRINT ERROR
68 031050 026137 000002 003104 CMP 2(R1),SERNM2 ;OTHER HALF OKAY
69 031056 001437 BEQ 5$ ;YES, EXIT
70 031060 4$: PRINTF #FRM3,2(R1),(R1),SERNM2,SERNM1
    031060 013746 003102 MOV SERNM1,-(SP)
    031064 013746 003104 MOV SERNM2,-(SP)
    031070 011146 MOV (R1),-(SP)
    031072 016146 000002 MOV 2(R1),-(SP)
    031076 012746 021007 MOV #FRM3,-(SP)
    031102 012746 000005 MOV #5,-(SP)
    031106 010600 MOV SP,R0
    031110 104017 EMT C$PNTF
    031112 062706 000014 ADD #14,SP
71 031116 004537 031160 JSR R5,UNLOAD ;LET OPERATOR CHANGE
72 031122 004537 031260 JSR R5,LOAD ;PACK
73 031126 000137 030512 JMP SERNUM ;GO CHECK IT AGAIN.
74 031132 99$: PRINTF #FRM15 ;MESSAGE
    031132 012746 021657 MOV #FRM15,-(SP)
    031136 012746 000001 MOV #1,-(SP)
    031142 010600 MOV SP,R0
    031144 104017 EMT C$PNTF
    031146 062706 000004 ADD #4,SP
75 031152 999$: BREAK
    031152 104022 EMT C$BRK
76 031154 000776 BR 999$
77 031156 000205 5$: RTS R5

```



```

1          ;ROUTINE UNLOAD
2
3          ;CALL JSR R5,UNLOAD
4
5 031160 UNLOAD: PRINTF #FRM1,<B,DSB+1(R4)>,(CSR(R4))
031160 016446 000000 MOV CSR(R4),-(SP)
031164 005045 CLR -(SP)
031166 156416 000005 BISB DSB+1(R4),(SP)
031172 012746 020612 MOV #FRM1,-(SP)
031176 012746 000003 MOV #3,-(SP)
031202 010600 MOV SP,R0
031204 104017 EMT C$PNTF
031206 062706 000010 ADD #10,SP
6 031212 012701 000074 MOV #60.,R1 ;SETUP 60 SECOND TIMER
7 031216 012700 000200 MOV #200,R0
8 031222 056400 000004 BIS DSB(R4),R0
9 031226 010074 000000 MOV R0,@CSR(R4)
10 031232 032774 000001 000C00 2$: BIT #DRDY,@CSR(R4) ;CHECK DRDY FOR ZERO
11 031240 001406 BEQ 3$ ;PACK UNLOADED
12 031242 WAITMS #10. ;WAIT 1 SECOND
031242 012700 000012 MOV #10.,R0
031246 104026 EMT C$WTM
13 031250 005301 DEC R1 ;HAS 60 SEC PASSED?
14 031252 001367 BNE 2$ ;NO, RETEST DRDY, CONTINUE WAIT
15 031254 000741 BR UNLOAD ;YES, REPEAT MESSAGE CONTINUE WAIT
16 031256 000205 3$: RTS R5 ;RETURN WITH PACK UNLOADED
17
18          ;ROUTINE LOAD
19
20          ;CALL JSR R5,LOAD
21
22 031260 LOAD: PRINTF #FRM2,<B,DSB+1(R4)>,(CSR(R4))
031260 016446 000000 MOV CSR(R4),-(SP)
031264 005046 CLR -(SP)
031266 156416 000005 BISB DSB+1(R4),(SP)
031272 012746 020707 MOV #FRM2,-(SP)
031276 012746 000003 MOV #3,-(SP)
031302 010600 MOV SP,R0
031304 104017 EMT C$PNTF
031306 062706 000010 ADD #10,SP
23 031312 012701 000170 MOV #120.,R1 ;SETUP 120 SEC TIMER
24 031316 012700 000200 MOV #200,R0 ;SETUP CONTROLLER READY BIT
25 031322 056400 000004 BIS DSB(R4),R0 ;SELECT DRIVE
26 031326 010074 000000 MOV R0,@CSR(R4)
27 031332 032774 000001 000000 2$: BIT #DRDY,@CSR(R4)
28 031340 001006 BNE 3$
29 031342 WAITMS #10.
031342 012700 000012 MOV #10.,R0
031346 104026 EMT C$WTM
30 031350 005301 DEC R1
31 031352 001367 BNE 2$
32 031354 000741 BR LOAD
33
34 031356 000205 3$: RTS R5

```

```

1          ;ROUTINE LDFUNC
2          ;CALL   JSR    R5,LDFUNC
3
4 031360 010046          LDFUNC: MOV    R0,-(SP)
5 031362 010346          MOV    R3,-(SP)
6 031364 010146          MOV    R1,-(SP)
7 031366 005037 002766  CLR    ERFLG          ;CLEAR ERROR FLAG
8 031372 016403 000000  MOV    (SR(R4),R3      ;GET CSR
9 031376 013763 003100 000006  MOV    BMP,MP(R3)     ;LOAD MULTIPURPOSE
10 031404 013763 003076 000004  MOV    BDA,DA(R3)    ;LOAD DISK ADDRESS
11 031412 013763 003074 000002  MOV    BBA,BA(R3)    ;LOAD BUS ADDRESS
12 031420 011537 003072          MOV    (R5),BCS      ;GET FUNCTION TO LOAD
13 031424 056437 0C0004 003072  BIS    DSB(R4),BCS   ;SELECT BITS
14 031432 012701 000031          MOV    #25.,R1       ;SET WATCHDOG TO 250MS
15 031436 052737 000200 003072  BIS    #200,BCS
16 031444 013763 0C3072 000000  MOV    BCS,CS(R3)    ;LOAD FUNCTION
17 031452 016337 000000 003072  MOV    CS(R3),BCS
18 031460 042763 00020C 000000  BIC    #200,CS(R3)
19 031466 032763 000200 000000 1$: BIT    #200,CS(R3)   ;CNTRL READY?
20 031474 001034          BNE    2$           ;YES, GO
21 031476          WAITUS. #100.    ;WAIT 10 MILLISECONDS
   031476 012700 000144  MOV    #100.,R0
   031502 104027          EMT    ($WTU
22 031504 005301          DEC    R1
23 031506 001367          BNE    1$
24
25 031510 016337 000000 003056  MOV    CS(R3),E.CS   ;READ ALL REGISTERS
26 031516 016337 000002 003060  MOV    BA(R3),E.BA
27 031524 016337 000004 003062  MOV    DA(R3),E.DA
28 031532 016337 000006 003064  MOV    MP(R3),E.MP
29 031540 016337 000006 003066  MOV    MP(R3),E.MP1
30 031546 016337 000006 003070  MOV    MP(R3),E.MP2
31 031554          ERRDF  210.,CNTTOT,ERR5;CNTRL TIMEOUT
   031554 104462          TRAP  TSERCODE
   031556 0C0322          .WORD 210
   031560 017166          .WORD CNTTOT
   031562 020464          .WORD ERR5
32 031564 000425          BR    4$
33
34 031566 016337 000000 003056 2$: MOV    CS(R3),E.CS   ;READ ALL REGISTERS
35 031574 016337 000002 003060  MOV    BA(R3),E.BA
36 031602 016337 000004 003062  MOV    DA(R3),E.DA
37 031610 016337 000006 0C3064  MOV    MP(R3),E.MP
38 031616 016337 000006 003066  MOV    MP(R3),E.MP1
39 031624 016337 00C006 003070  MOV    MP(R3),E.MP2
40
41 031632 005737 003056          TST    E.CS          ;ANY ERRORS?
42 031636 100002          BPL    3$           ;YES, GO SERVICE
43 031640 005237 002766          4$: INC    ERFLG
44 031644 005725          3$: TST    (R5)+
45 031646 012601          MOV    (SP)+,R1
46 031650 012603          MOV    (SP)+,R3
47 031652 012600          MOV    (SP)+,R0
48 031654 000205          RTS    R5
49
50 031656          ENDMOD

```

```

1 031656          BGNTST
2
3                :CONTROL SECTION COMPATABILITY PROGRAM
4                :PRINT UNLOAD AND LOAD DRIVE MESSAGES
5                :PERFORM SERIAL CHECK ROUTINE
6                :PERFORM READ/WRITE CHECKS ON DRIVES
7
8 031656 012701 002334 COMPAT: MOV #SECBUF,R1      :ADJ. CYLINDER BUFFER
9 031662 012700 000170      MOV #120.,R0      :ADJ. CYLINDER BUFFER COUNT
10 031666 005021          4$: CLR (R1)+      :CLEAR ADJ. CYL. BUFFER AT STARTUP
11 031670 005300          DEC R0      :BUFFER CLEARED?
12 031672 001375          BNE 4$      :CLEAR NEXT BUFFER WORD
13 031674 005237 002754   INC F0WR      :SET FIRST OVERWRITE FLAG
14 031700 004537 023572   JSR R5,OVWPER :PERFORM OVERWRITE ON FIRST DRIVE
15 031704 177400          177400
16 031706 000377          377
17 031710 005037 002754   CLR F0WR      :CLEAR FIRST OVERWRITE
18 031714 005237 002756   INC FADJ      :SET FIRST ADJ. FLAG
19 031720 005237 002776   INC ADJDIR    :UP = 1
20 031724 004537 024314   JSR R5,ADJCYL
21 031730 003 377        .BYTE 3,377      :TRACK AND SECTORS FOR
22 031732 170000          .WORD 170000     :INWARD SEEK
23 031734 003 000        .BYTE 3,0        :TRACK AND SECTORS FOR
24 031736 007777          .WORD 7777      :OUTWARD SEEK
25 031740 000000          .WORD 0        :TERMINATOR
26 031742 004537 031160   JSR R5,UNLOAD :UNLOAD PACK FROM DRIVE UNIT
27 031746 062704 000010   ADD #PAT+2,R4 :UPDATE POINTER FOR NEXT DRIVE
28 031752 004537 031260   JSR R5,LOAD   :LOAD INTO SECOND DRIVE UNIT
29 031756 004537 030512   JSR R5,SERNUM :CHECK PACK SERIAL NUMBER
30 031762 004537 023572   JSR R5,OVWPER :PERFORM R/W OVERWRITE
31 031766 000360          360
32 031770 000017          17
33 031772 005237 002776   INC ADJDIR
34 031776 004537 024314   JSR R5,ADJCYL
35 032002 002 360        .BYTE 2,360     :IN 1/0 OUTSIDE
36 032004 000000          .WORD 0
37 032006 002 017        .BYTE 2,17      :OUT 1/0 OUTSIDE
38 032010 000000          .WORD 0
39 032012 004 360        .BYTE 4,360     :IN 1/0 INSIDE
40 032014 000000          .WORD 0
41 032016 004 017        .BYTE 4,17      :OUT 1/0 INSIDE
42 032020 000000          .WORD 0
43 032022 000000          .WORD 0
44 032024 004537 031160   JSR R5,UNLOAD :UNLOAD PACK FROM DRIVE UNIT
45 032030 023727 003022 000002 CMP UUT,#2     :CHECK FOR > 2 DRIVES
46 032036 00100~          BNE 10$      :YES,GO TO NEXT DRIVE
47 032040 000137 032454   JMP 2$      :GO TO FIRST DRIVE
48 032044 062704 000010   10$: ADD #PAT+2,R4 :UPDATE DRIVE BUFFER FOR THIRD DRIVE
49 032050 004537 031260   JSR R5,LOAD   :LOAD PACK FOR THIRD DRIVE
50 032054 004537 030512   JSR R5,SERNUM :CHECK SERIAL NUMBERS
51 032060 004537 023572   JSR R5,OVWPER :PERFORM R/W OVERWRITE ON THIRD DRIVE
52 032064 006014          6014
53 032066 001403          1403
54 032070 005237 002776   INC ADJDIR
55 032074 004537 024314   JSR R5,ADJCYL
56 032100 002 000        .BYTE 2,0        :IN 2/0 OUTSIDE
57 032102 170000          .WORD 170000

```

58	032104	002	000	.BYTE	2,0	;OUT 2/0 OUTSIDE
59	032106	007400		.WORD	7400	
60	032110	004	000	.BYTE	4,0	;IN 2/0 INSIDE
61	032112	170000		.WORD	170000	
62	032114	004	000	.BYTE	4,0	;OUT 2/0 INSIDE
63	032116	007400		.WORD	7400	
64	032120	0C1	200	.BYTE	1,200	;IN 2/1 OUTSIDE
65	032122	000000		.WORD	0	
66	032124	001	100	.BYTE	1,100	;OUT 2/1 OUTSIDE
67	032126	000000		.WORD	0	
68	032130	005	200	.BYTE	5,200	;IN 2/1 INSIDE
69	032132	000000		.WORD	0	
70	032134	005	100	.BYTE	5,100	;OUT 2/1 INSIDE
71	032136	000000		.WORD	0	
72	032140	000000		.WORD	0	;TERMINATOR
73	032142	004537	031160	JSR	R5,UNLOAD	;UNLOAD PACK ON THIRD DRIVE
74	032146	023727	003022	000003	CMP	UUT,#3
75	032154	001500		BEQ	1\$;CHECK FOR > 3 DRIVES
76	032156	062704	000010	ADD	#PAT+2,R4	;NO, GO TO 2ND DRIVE
77	032162	004537	031260	JSR	R5,LOAD	;UPDATE DRIVE BUFFER FOR 4TH DRIVE
78	032166	004537	030512	JSR	R5,SERNUM	;LOAD PACK ON 4TH DRIVE
79	032172	004537	023572	JSR	R5,OVWPER	;CHECK PACK ON FOURTH DRIVE
80	032176	001042			1042	;PERFORM R/W OVERWRITE
81	032200	000421			421	
82	032202	005237	002776	INC	ADJDIR	
83	032206	004537	024314	JSR	R5,ADJCYL	
84	032212	002	000	.BYTE	2,0	;IN 3/0 OUTSIDE
85	032214	000360		.WORD	360	
86	032216	002	000	.BYTE	2,0	;OUT 3/0 OUTSIDE
87	032220	000017		.WORD	17	
88	032222	004	000	.BYTE	4,0	;IN 3/0 INSIDE
89	032224	000360		.WORD	360	
90	032226	004	000	.BYTE	4,0	;OUT 3/0 INSIDE
91	032230	000017		.WORD	17	
92	032232	001	040	.BYTE	1,40	;IN 3/1 OUTSIDE
93	032234	000000		.WORD	0	
94	032236	001	020	.BYTE	1,20	;OUT 3/1 OUTSIDE
95	032240	000000		.WORD	0	
96	032242	005	040	.BYTE	5,40	;IN 3/1 INSIDE
97	032244	000000		.WORD	0	
98	032246	005	020	.BYTE	5,20	;OUT 3/1 INSIDE
99	032250	000000		.WORD	0	
100	032252	001	000	.BYTE	1,0	;IN 3/2 OUTSIDE
101	032254	100000		.WORD	100000	
102	032256	001	000	.BYTE	1,0	;OUT 3/2 OUTSIDE
103	032260	040000		.WORD	40000	
104	032262	005	000	.BYTE	5,0	;IN 3/2 INSIDE
105	032264	0000		.WORD	100000	
106	032266	005	000	.BYTE	5,0	;OUT 3/2 INSIDE
107	032270	040000		.WORD	40000	
108	032272	000000		.WORD	0	;TERMINATOR
109	032274	004537	031160	JSR	R5,UNLOAD	;UNLOAD PACK FROM 4TH DRIVE
110	032300	162704	000010	SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 3RD DRIVE
111	032304	004537	031260	JSR	R5,LOAD	;LOAD PACK ON 3RD DRIVE
112	032310	004537	030512	JSR	R5,SERNUM	;CHECK FOR PACK SERIAL NUMBER
113	032314	004537	023572	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE ON 3RD DRIVE
114	032320	020000			20000	

115	032322	010000		10000		
116	032324	004537	024314	JSR	R5,ADJCYL	
117	032330	001	000	.BYTE	1,0	;IN 2/3 OUTSIDE
118	032332	000200		.WORD	200	
119	032334	001	000	.BYTE	1,0	;OUT 2/3 OUTSIDE
120	032336	000100		.WORD	100	
121	032340	005	000	.BYTE	5,0	;IN 2/3 INSIDE
122	032342	000200		.WORD	200	
123	032344	005	000	.BYTE	5,0	;OUT 2/3 INSIDE
124	032346	000100		.WORD	100	
125	032350	000000		.WORD	0	;TERMINATOR
126	032352	004537	031160	JSR	R5,UNLOAD	;UNLOAD PACK FROM 3RD DRIVE
127	032356	162704	000010	18: SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 2ND DRIVE
128	032362	004537	031260	JSR	R5,LOAD	;LOAD PACK ON THIRD DRIVE
129	032366	004537	030512	JSR	R5,SERNUM	;CHECK PACK SERIAL NUMBER
130	032372	004537	023572	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE ON 2ND DRIVE
131	032376	004040		4040		
132	032400	002020		2020		
133	032402	004537	024314	JSR	R5,ADJCYL	
134	032406	001	000	.BYTE	1,0	;IN 1/2 OUTSIDE
135	032410	020000		.WORD	20000	
136	032412	001	000	.BYTE	1,0	;OUT 1/2 OUTSIDE
137	032414	010000		.WORD	10000	
138	032416	005	000	.BYTE	5,0	;IN 1/2 INSIDE
139	032420	020000		.WORD	20000	
140	032422	005	000	.BYTE	5,0	;OUT 1/2 INSIDE
141	032424	010000		.WORD	10000	
142	032426	001	000	.BYTE	1,0	;IN 1/3 OUTSIDE
143	032430	000040		.WORD	40	
144	032432	001	000	.BYTE	1,0	;OUT 1/3 OUTSIDE
145	032434	000020		.WORD	20	
146	032436	005	000	.BYTE	5,0	;IN 1/3 INSIDE
147	032440	000040		.WORD	40	
148	032442	005	000	.BYTE	5,0	;OUT 1/3 INSIDE
149	032444	000020		.WORD	20	
150	032446	000000		.WORD	0	;TERMINATOR
151	032450	004537	031160	JSR	R5,UNLOAD	;UNLOAD PACK FROM 2ND DRIVE
152	032454	162704	000010	28: SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 1ST DRIVE
153	032460	004537	031260	JSR	R5,LOAD	;LOAD PACK INTO FIRST DRIVE UNIT
154	032464	004537	030512	JSR	R5,SERNUM	;CHECK SERIAL NUMBER
155	032470	004537	023572	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE
156	032474	001042		1042		
157	032476	000421		421		
158	032500	004537	024314	JSR	R5,ADJCYL	
159	032504	001	010	.BYTE	1,10	;IN 0/1 OUTSIDE
160	032506	000000		.WORD	0	
161	032510	001	004	.BYTE	1,4	;OUT 0/1 OUTSIDE
162	032512	000000		.WORD	0	
163	032514	005	010	.BYTE	5,10	;IN 0/1 INSIDE
164	032516	000000		.WORD	0	
165	032520	005	004	.BYTE	5,4	;OUT 0/1 INSIDE
166	032522	000000		.WORD	0	
167	032524	001	000	.BYTE	1,0	;IN 0/2 OUTSIDE
168	032526	004000		.WORD	4000	
169	032530	001	000	.BYTE	1,0	;OUT 0/2 OUTSIDE
170	032532	002000		.WORD	2000	
171	032534	005	000	.BYTE	5,0	;IN 0/2 INSIDE

172	032536	004000		.WORD	4000	
173	032540	005	000	.BYTE	5,0	:OUT 0/2 INSIDE
174	032542	002000		.WORD	2000	
175	032544	001	000	.BYTE	1,0	:IN 0/3 OUTSIDE
176	032546	000010		.WORD	10	
177	032550	001	000	.BYTF	1,0	:OUT 0/3 OUTSIDE
178	032552	000004		.WORD	4	
179	032554	005	000	.BYTE	5,0	:IN 0/3 INSIDE
180	032556	000010		.WORD	10	
181	032560	005	000	.BYTE	5,0	:OUT 0/3 INSIDE
182	032562	000004		.WORD	4	
183	032564	000000		.WORD	0	:TERMINATOR
184	032566	004537	031160	JSR	R5,UNLOAD	:UNLOAD PACK
185	032572			PRINTF	#ENDPAS	:END OF PASS
	032572	012746	022102	MOV	#ENDPAS,-(SP)	
	032576	012746	000001	MOV	#1,-(SP)	
	032602	010600		MOV	SP,R0	
	032604	104017		EMT	C\$PNTF	
	032606	062706	000004	ADD	#4,SP	
186	032612			3\$: BREAK		
	032612	104022		EMT	C\$BRK	
187	032614	000776		BR	3\$	
188						
189						
190						
191	032616			ENDTST		
	032616			L10012:		
	032616	104001		EMT	C\$ETST	
192						
193	032620			BGNMOD	HRDPRM	
194	032620			BGNHRD		
	032620	000020		.WORD	L10013-L\$HARD/2	
195						
196	032622			GPRMA	CSRMSG,CSR,0,160000,177776,YES	
	032622	000031		.WORD	T\$CODE	
	032624	032662		.WORD	CSRMSG	
	032626	160000		.WORD	T\$LLOLM	
	032630	177776		.WORD	T\$HILIM	
197	032632			GPRMA	VECMMSG,VECT,0,0,776,YES	
	032632	001031		.WORD	T\$CODE	
	032634	032720		.WORD	VECMMSG	
	032636	000000		.WORD	T\$LLOLM	
	032640	000776		.WORD	T\$HILIM	
198	032642			GPRMD	DRMSG,DRBT,0,03400,0,7,YES	
	032642	003032		.WORD	T\$CODE	
	032644	032727		.WORD	DRMSG	
	032646	003400		.WORD	03400	
	032650	000000		.WORD	T\$LLOLM	
	032652	000007		.WORD	T\$HILIM	
199	032654			GPRML	DRTYPE,TYPDR,1,YES	
	032654	002130		.WORD	T\$CODE	
	032656	032676		.WORD	DRTYPE	
	032660	000001		.WORD	1	
200	032662			ENDHRD		
				.EVEN		
	032662			L10013:		
201						

```
202 032662      102      125      123  CGRMSG: .ASCII? /BUS ADDRESS/
      032665      040      101      104
      032670      104      122      105
      032673      123      123      000
203 032676      104      122      111  DRTYPE: .ASCII? /DRIVE TYPE = RL01/
      032701      126      105      040
      032704      124      131      120
      032707      105      040      075
      032712      040      122      114
      032715      060      061      000
204 032720      126      105      103  VECMSG: .ASCII? /VECTOR/
      032723      124      117      122
      032726      000
205 032727      104      122      111  DRMSG: .ASCII? /DRIVE/
      032732      126      105      000
206
207 032736
208      033114
209 033114
      033114

      .EVEN
      ENDMOD
      .=33114 ;THIS IS A FIX FOR THE 'APT' MAILBOX
      LASTAD
      .EVEN
      L$LAST::
```

1

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

PWR.FAIL:

POWER INTERRUPT ROUTINE

SEQ 0082

35 063710 000000
36 063712 000000
37 063714 000000
38 063716 000000
39 063722
40 000200

.WORD 0
.WORD 0
.WORD 0
.WORD 0
END.SUPV=.*2
.END 200

:SPACE FOR USER POOL POINTER
:SIZE
:CHECKSUM (NOT CURRENTLY USED)
:SIZE OF H.W. PTAB. ALLOCATION

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:41:44 PAGE 181-1

F 7

SEQ 0083

ABOFLA 033440 G
 ABOPAS 033356 G
 ABO.FM 035720
 ADJCYL 024314
 ADJDIR 002776
 ADJFLG 002774
 ADJLC2 003112
 ADJLC3 003114
 ADJLC4 003116
 ADJLOC 002772
 ADJTRK 003106
 ADJTXT 020030
 ADJUUT 003110
 AFSI 033146 G
 ALLOC 054060
 APT.ER 035050
 ASSEMB= 000011
 ASAAV 037716
 ASAAW 037732
 ASAAZ 037744
 ASAAZ 037752
 ASAAZ 037766
 ASABA 037776
 BA = 000002
 BA16 = 000020
 BA17 = 000040
 BBA 003074
 BCS 003072
 BDA 003076
 BDATA 003032
 BGN.SU= 033114
 BINMSG 052370
 BIT0 = 000001 G
 BIT00 = 000001 G
 BIT01 = 000002 G
 BIT02 = 000004 G
 BIT03 = 000010 G
 BIT04 = 000020 G
 BIT05 = 000040 G
 BIT06 = 000100 G
 BIT07 = 000200 G
 BIT08 = 000400 G
 BIT09 = 001000 G
 BIT1 = 000002 G
 BIT10 = 002000 G
 BIT11 = 004000 G
 BIT12 = 010000 G
 BIT13 = 020000 G
 BIT14 = 040000 G
 BIT15 = 100000 G
 BIT2 = 060004 G
 BIT3 = 000010 G
 BIT4 = 000020 G
 BIT5 = 000040 G
 BIT6 = 000100 G
 BIT7 = 000200 G
 BIT8 = 000400 G

BIT9 = 001000 G
 BLD.MW 040602
 BLOCK 056214
 BMP 003100
 BSEXIT 027524
 BSVWR 027342
 BUF 003124
 B\$AAB 042204
 B\$AAF 042116
 CALLPC= 000022
 CALLPS= 000024
 CALLSP= 000026
 CALLTC= 000030
 CAL.CL 060602
 CAL.TI 060640 G
 CHECK 027552
 CHKLUP 042220
 CHKSTR 054422
 CHKTTY 052510
 CHK.MA 040360
 CHK.PC 045510
 CHK.SW 034550
 CHRCNT 053742
 CH.FLA 040066
 CH.PAS 040104
 CLEAR. 041502
 CLKACC 033354 G
 CLKBFR 060604
 CLKCNT 033352 G
 CLKJUM 061210 G
 CLKRES 062212 G
 CLKSER 062346 G
 CLKSON 033412 G
 CLK.SE 040162
 CLNCOD 023562 G
 CLR.MA 040436
 CMPENA 023536
 CNTTOT 017166
 CNVT 056660
 COMMAN 033164 G
 COMMTA 056474
 COMPAT 031656
 CONTCL 062272 G
 CRDY = 000200
 CRLF 052572
 CRSET = 000002
 CS = 000000
 CSR = 000000
 CSRMSG 032662
 CURR.S 033122 G
 CURR.T 033124 G
 CYL 003014
 C\$AAD 045462
 C\$AAE 045474
 C\$AAK 046472
 C\$AAL 046636
 C\$ABRT= 000021

C\$ADR = 000020
 C\$AU = 000054
 C\$BRK = 000022
 C\$BSEG= 000004
 C\$BSUB= 000002
 C\$BUFF= 000030
 C\$CEFG= 000046
 C\$CLEA= 000012
 C\$CLP1= 000006
 C\$CVEC= 000036
 C\$DCLM= 000044
 C\$DODU= 000053
 C\$DRPT= 000024
 C\$DU = 000055
 C\$EDIT= 000002
 C\$ERDF= 000002
 C\$ERHR= 000003
 C\$ERSF= 000001
 C\$ERSO= 000004
 C\$ESCA= 000010
 C\$ESEG= 000005
 C\$ESUB= 000003
 C\$ETST= 000001
 C\$EXIT= 000032
 C\$GMAN= 000043
 C\$GPHR= 000042
 C\$GPR1= 000040
 C\$GTIM= 000052
 C\$INIT= 000011
 C\$INLP= 000020
 C\$KWOF= 000035
 C\$KWON= 000034
 C\$LLOOP= 000100
 C\$MANI= 000051
 C\$MSG = 000023
 C\$PNTB= 000014
 C\$PNTF= 000017
 C\$PNTS= 000016
 C\$PNTX= 000015
 C\$POIN= 000040
 C\$Q10 = 000377
 C\$RDBU= 000007
 C\$REFG= 000050
 C\$REQT= 000045
 C\$RESE= 000033
 C\$REVI= 000002
 C\$RPT = 000025
 C\$SEFG= 000047
 C\$SPRI= 000041
 C\$SVEC= 000037
 C\$TPRI= 000013
 C\$UNBU= 000031
 C\$WTM = 000026
 C\$WTU = 000027
 DA = 000004
 DCKER 017275
 DCRC = 004000

DECMG 052404
 DERR = 040000
 DEV.CO 033126 G
 DIAGMC= 000000
 DIAG.T 033446 G
 DIRC 003010
 DLT = 010000
 DPDVD 043056 G
 DPMUL 043074 G
 DRBT 043006
 DRBUF 043124
 DRDY 0430001
 DRMSG 032727
 DRPCOD 023566 G
 DRST = 000013
 DRSTAT 003000
 DRTYPE 032676
 DSB = 000004
 DSPCOD 022142 G
 DUNIT. 033362 G
 DVC.FT 046442
 D\$AAG 047346
 D\$AAH 047364
 D\$AAI 052132
 D\$AAJ 052136
 D\$AAK 052154
 D\$AAL 052172
 D\$AAM 052202
 EF.CON= 000036 G
 EF.NEW= 000035 G
 EF.PWR= 000034 G
 EF.RES= 000037 G
 EF.STA= 000040 G
 EF01 = 000001 G
 EF02 = 000002 G
 EF03 = 000003 G
 EF04 = 000004 G
 EF05 = 000005 G
 EF06 = 000006 G
 EF07 = 000007 G
 EF08 = 000010 G
 EF09 = 000011 G
 EF10 = 000012 G
 EF11 = 000013 G
 EF12 = 000014 G
 EF13 = 000015 G
 EF14 = 000016 G
 EF15 = 000017 G
 EF16 = 000020 G
 EMT.TR 033444 G
 END 022314
 ENDBUF 017164
 ENDPAS 022102
 END.OF 041470
 END.SU= 063722
 ENVIRO 033166 G
 EOP.LM 062370 G

EOP.FM 035734
 EOP.IN 040100
 ERFLG 002766
 ERR = 100000
 ERRFND 017563
 ERRFOR 046714
 ERRHAN 045514
 ERR.HR 046452
 ERR.NU 033116 G
 ERR.SF 046456
 ERR1 020060 G
 ERR1FO 047000
 ERR2 020116 G
 ERR3 020156 G
 ERR4 020304 G
 ERR5 020464 G
 ERR6 020524 G
 ESC.PC 045506
 EV.COU 033120 G
 EXIT 023560
 E.BA 003060
 E.CS 003056
 E.DA 003062
 E.MP 003064
 E.MP1 003066
 E.MP2 003070
 FADJ 002756
 FEW 017313
 FILL 053240
 FILL.C 000204 G
 FIXCYL 030456
 FLAGS 033160 G
 FLAGSI 033162 G
 FLAGTA 056412
 FLAG.I 040146
 FLA.SE 056360
 FLG.MA 040106
 FNDBSC 030366
 FNDDR 026124
 FNDTRK 030214
 FORM.T 047010
 FORSK 003020
 FOWR 002754
 FREE 054316
 FRM1 020612
 FRM10 021365
 FRM11 021431
 FRM12 021466
 FRM13 021545
 FRM14 021630
 FRM15 021657
 FRM16 021716
 FRM17 022003
 FRM18 022037
 FRM2 020707
 FRM3 021007
 FRM4 021066

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:41:44 PAGE 181-2

G 7

SEQ 0084

FRM5	021127	GSXFER=	000004	JSJMP =	000167	L\$TIMU	00205' G	NUM.NO	033156 G
FRM6	021176	GSYES =	000010	KBPTR	033224 G	L\$TIM1	002052 G	NUM.UN	033564
FRM7	021217	HCORED	037656	KBUF	033226 G	L\$STSI	002100 G	NUNITS	042172
FRM8	021257	HCOREQ	037566	LDFUNC	031360	L\$UNIT	002012 G	NXM =	020000
FRM9	021323	HCORET	033402 G	LINE.F	033442 G	I.CLK.	037512	NXTFOR	056652
FRTTRK	003044	HCRC =	004000	LOAD	031260	L10000	020114	OCTMSG	052376
FUNERR	017667	HC.ADR	033152 G	LOAD.F	040102	L10001	020154	O\$FSET	003040
FWD	017736	HC.DEF	033144 G	LOGMSG	052412	L10002	020302	OPI =	002000
F\$AU =	000015	HC.DIA	033142 G	LPBFR	033222 G	L10003	020462	OQU10	002154
F\$BGM =	000040	HDRFND	002126	LPCNTR	033220 G	L10004	020522	OQU11	002166
F\$CLEA=	000007	HEAD =	000100	LPT.AD	037544	L10005	020610	OQU20	002156
F\$DU =	000016	HEAD01	003006	LPT.RE	037540	L10006	022142	OQU21	002170
F\$END =	000041	HERTZ.	037526	LSI.RE	037534	L10007	023560	OQU30	002160
F\$HARD=	000004	HNF =	010000	LSTCLR	002762	L10010	023564	OQU31	002172
F\$HW =	000013	HOLDSP=	000020	LSTDRV	003026	L10011	023570	OQU40	002162
F\$INIT=	000006	HPTCOD	022130 G	LSTTRK	003042	L10012	032616	OQU41	002174
F\$JMP =	000050	HRDPRM	032620 G	LUP	060506	L10013	032662	OQU50	002164
F\$MOD =	000000	HSFLG	003002	LUP.AD	045512	MAJ.IN	033176 G	OQU51	002176
F\$MSG =	000011	HW.ADR	033150 G	L\$APT	002036 G	MAJ.LO	060606	OSECT	003004
F\$PWR =	000017	H\$AAB	057206	L\$AUT	002074 G	MAJ.US	033200 G	OUT10	002130
F\$RPT =	000012	INAWR	027440	L\$CCP	002106 G	MANY	017352	OUT11	002142
F\$SEG =	000003	ININIT	033372 G	L\$CLEA	023562 G	MAN.TI	001244	OUT20	002132
F\$SOFT=	000005	INITCO	022146 G	L\$CO	002032 G	MAP16	063314 G	OUT21	002144
F\$SRV =	000010	INITIA	052420	L\$DEPO	002011 G	MASK.B	042216	OUT30	002134
F\$SUB =	000002	INITWR	017213	L\$DESC	002102 G	MASK.W	042214	OUT31	002146
F\$SW =	000014	INIT.M	040504	L\$DEVP	002064 G	MDHEDR	002000 G	OUT40	002136
F\$TEST=	000001	INIT.R	033206 G	L\$DISP	022144 G	MEM.SI	037554	OUT41	002150
GARBAG	053744	INN10	002250	L\$DR	002112 G	MERGE	027766	OUT50	002140
GDATA	003030	INN11	002262	L\$DRCT	002070 G	MID10	002200	OUT51	002152
GETCHR	052450	INN20	002252	L\$DRS	002072 G	MID11	002212	OVMS	017462
GETCMN	056034	INN21	002264	L\$DRST	002112 G	MID20	002202	OVWER	017623
GETPAR	047526	INN30	002254	L\$DTP	002040 G	MID21	002214	OVWPER	023572
GETSWI	055030	INN31	002266	L\$DU	023566 G	MID30	002204	OVWTRK	002714
GET.TW	054600	INN40	002256	L\$DUT	002076 G	MID31	002216	O\$APTS=	000000
GLBDAT	002126 G	INN41	002270	L\$DVTY	002114 G	MID40	002206	O\$AU =	000000
GLBEQA	002126 G G	INN50	002260	L\$EF	002056 G	MID41	002220	O\$BGNR=	000000
GLBERR	020060 G G	INN51	002272	L\$EFLG	002034 G	MID50	002210	O\$BGNS=	000000
GLBSUB	023572 G	INPUTA	053346	L\$EXP1	002042 G	MID51	002222	O\$DU =	000000
GLBXT	017166 G	INTEN =	000100	L\$EXP2	002044 G	MIN.IN	033172 G	O\$GNSW=	000000
GSBIT =	000003	INTFOR	046644	L\$EXP3	002046 G	MIN.US	033174 G	O\$POIN=	000001
GSTAT =	000004	INVAL.	037452	L\$HARD	032622 G	MK =	000001	O\$PWR =	000000
G\$EXCP=	000400	INVINT	046502	L\$HPCP	002016 G	MODR	062656 G	PARSES	056106
G\$HILI=	000002	INV.SW	034504	L\$HPTP	002022 G	MP =	000006	PAR.LA	052074
G\$LOLI=	000001	IN.SUF	041454	L\$HW	022132 C	MSG.AD	033140 G	PASS.C	033130 G
G\$NO =	000000	I\$AU =	000041	L\$ICP	002104 G	MSG.TY	033114 G	PAT =	000006
G\$OFFS=	000400	I\$CLN =	000041	L\$INIT	022146 G	MUL	062612 G	PATLST	002740
G\$OFFSI=	000376	I\$DU =	000041	L\$LADP	002026 G	NCHECK	027536	PRINTC	053720
G\$PRMA=	000001	I\$HRD =	000041	L\$LAST	033114 G	NEWPRI	062336 G	PRINTF	057226
G\$PRMD=	000002	I\$INIT=	000041	L\$MREV	002050 G	NEXAR	056576	PRI00 =	000000 G
G\$PRML=	000000	I\$MOD =	000041	L\$NAME	002000 G	NONE	017411	PRI01 =	000040 G
G\$RADA=	000140	I\$MSG =	000041	L\$REPP	002066 G	NO.CLK	037502	PRI02 =	000100 G
G\$RADB=	000000	I\$PWR =	000041	L\$REV	002010 G	NO.FLA	056372	PRI03 =	000140 G
G\$RADD=	000040	I\$RPT =	000041	L\$SPC	002062 G	NO.LPT	053710	PRI04 =	000200 G
G\$RADF=	000200	I\$SEG =	000041	L\$SPCP	002020 G	NO.PTA	037706	PRI05 =	000240 G
G\$RADL=	000120	I\$SRV =	000041	L\$SPTP	002024 G	NR =	000000	PRI06 =	000300 G
G\$RADN=	000020	I\$SUB =	000041	L\$STA	002030 G	NUMBIN	047034	PRI07 =	000340 G
G\$RADT	000100	I\$TST	000041	L\$TIML	002014 G	NUM.LA	047202	PRNTST	053610

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:41:44 PAGE 181-3

M 7

SEQ 0085

PRO.CM	040060	SERNM2	003104	SYS.FT	046432	TSCODE=	002130	VAL.LA	034454
PRSTRK	003046	SERNUM	030512	SLSYM=	010000	T\$ERCO=	000062	VAL.SW	040120
PTAB.S	033400 G	SETLST	026064	TEM	002750	T\$ERRN=	000322	VEC	= 000002
PUTCHR	052424	SETUP	022366	TEMP	002760	T\$EXCP=	000000	VECMSG	032720
PWR.FA	063550 G	SET.MA	040272	TERMI	060576	T\$HILI=	000007	VECT	= 000002
PWR.FL	033204 G	SHIFT	063376 G	TERMI1	056400	T\$LOLI=	000000	VEROD	026550
PWR.MS	063676	SIGN	= 000004	TERMTA	052362	T\$LSYM=	010000	VEROW	026164
PWR.SA	063672	SKCYL	025240	TEST.M	040020	T\$NEST=	177777	WCOUNT	003034
PWR.UP	063674	SKER	017717	TIM.FLG	033350 G	T\$NSKO=	000000	WIDTH	047402
P.CLK.	037520	SKHS	= 000020	TIM.CO	033202 G	T\$NSK1=	000004	WRITE	= 000012
RDHDR =	000010	SPEC.U	040006	TIM.OP	047006	T\$SAVL=	177777	WRIT1	017756
READ =	000014	SPV.SE	000400	TOO.MA	052342	T\$SEGL=	177777	WRSEC	024054
READ.P	060610 G	STARTC	062266 G	TQU10	002224	T\$SUBN=	000000	XEQDIA	062424 G
READ1	020003	STFLG	002770	TQU11	002236	T\$TAGL=	177777	XEQSUB	062412 G
REASON	002764	STRCHR	053300	TQU20	002226	T\$TAGN=	010014	XEQ.CL	042134
RECER	017643	STRT.T	040064	TQU21	002240	T\$TEMP=	000000	XEQ.CM	037444
RECMS	017515	STSEC	003122	TQU30	002230	T\$TEST=	000001	XEQ.IN	041616
REGBAC	063300 G	STSEC1	003120	TQU31	002242	T\$TSTM=	177777	XEQ.LA	035706
REGDMP	025506	ST.SET	034716	TQU40	002232	T\$TSTS=	000001	XEQ.OP	041710
REGSAV	063264 G	SUNIT.	040070	TQU41	002244	T\$SCLE=	010010	XEQ.PR	035110
REQN.P	033170 G	SUPERV	035752	TQU50	002234	T\$SDU =	010011	XEQ.TE	041754
REQN.T	040062	SUPFLA	033360 G	TQU51	002246	T\$SHAR=	010013	XTIME	061276 G
REV	017746	SUPV.T	033532 G	TRKCNT	003054	T\$SHW =	010006	XTIMEN	062122
REVSK	003016	SUP.PR	034470	TRKFND	003052	T\$SINI=	010007	XTIMST	061320
RE.SET	034652	SURF	003012	TST.AB	042330	T\$SMG=	010005	XXDP.D	037464
RSADJS	025716	SURFAC	003050	TST.TO	034532	T\$STES =	010012	X\$ALWA=	000000
RSTACK	062540 G	SVCGBL=	000000	TYPDR =	000004	T.DRIV	002752	X\$FALS=	000040
SAVEDO=	035050	SVCHAN	042406	TYPEC	052736	T1	031656 G	X\$OFFS=	000400
SEARCH	054546	SVCHNS=	000000	TYPEPC	046632	UNIT.D	033132 G	X\$TRUE=	000020
SECBUF	002334	SVCSUB=	177777	TYPEPLA	056254	UNI.MA	040010	\$BRFG	040160
SECLST	002274	SVCTAG=	000000	TYPI IN	052634	UNLOAD	031160	\$ENDAD	062376 G
SECT	003024	SVCTST=	177777	TYPNUM	052216	USER.P	033374 G	\$SAV2	063442 G
SECWRD	003036	SWCHAN	037700	TYPSTR	052654	USER.T	033376 G	\$SAV3	063456 G
SEEK =	000006	SWITCH	056552	TYP.ER	046462	UUT	003022	\$SAV4	063474 G
SEGSTA	033414 G	SW.ADR	033154 G	TY.UNI	041474	VAJWR	027076	\$SAV5	063514 G
SERNM1	003102	SW.PTA	037664	T\$ARGC=	000001	VALID.	033634		

. ABS. 063720 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 17720 WORDS (70 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
CZRLA.BIN,CZRLA-#SVCR/M,CZRLA.DOCTOR