

RL01, RL02

RL01/02 DRIVE COMPT
CZRL LCO

AH-F130C-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 79-82
MADE IN USA



The main body of the document is a large, dark grid containing numerous small, illegible data points or tables. The grid is organized into approximately 10 columns and 20 rows. Each cell in the grid appears to contain a small table or set of data, but the text is too faint to be read. The overall appearance is that of a dense data matrix or a series of microfilm frames.

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F131C-MC
PRODUCT NAME: CZRLLC0 RL01/02 DRIVE COMPATABILITY
DATE CREATED: 5-JAN-79
REVISED: 4-FEB-82
MAINTAINER: DIAGNOSTIC ENGINEERING - COLORADC
AUTHORS: D. DEKNIS, C. CAMPBELL

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

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

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

COPYRIGHT (C) 1979, 1982 DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH XXDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP+, AND CAN BE CHAINED UNDER XXDP+, ACT AND APT IN ACT MODE (SEE 2.2 "CHAIN MODE OPERATION" FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES.

WHEN THIS DIAGNOSTIC IS STARTED, CONTROL GOES FIRST TO THE SUPERVISOR PORTION WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

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

1.1.2 DIAGNOSTIC INFORMATION

THE RLO1 DRIVE COMPATABILITY TEST IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES. THE TEST PERFORMS WRITES, READS, OVERWRITES, ADJACENT CYLINDER WRITES TO PROVE COMPATABILITY. SINCE THE PROGRAM RELIES ON MANUAL INTERVENTION, A TOTAL TEST TIME IS NOT APPLICABLE. HOWEVER, TO TEST TWO DRIVES REQUIRES A MINIMUM OF THREE PASS. EACH PASS REQUIRES APPROXIMATELY 70 SECONDS.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

- * PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- * CONSOLE DEVICE (LA30, LA36, VT50, ETC.)

* 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:

- 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'

* LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLLC0 RL01/02 DRIVE COMPATABILITY
(FORMERLY CZRLF0)

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

CVRLAB0	RLV11 RL01/02 DISKLESS TEST (RLV11 ONLY)
CZRLGCO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLHCO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLIDO	RL01/02 DRIVE TEST (PART 1)
CZRLJBO	RL01/02 DRIVE TEST (PART 2)
CZRLKBO	RL11/RLV11 RL01/02 PERFORMANCE EXERCISER
CZRLNAO	RL01/02 DRIVE TEST (PART 3)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

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

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

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

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

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

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING 5 STEPS WILL OCCUR:

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

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

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

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

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE 'DR>' LEVEL NEED TO BE TYPED.
2. THE 'PASS' SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE 'FLAGS' SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

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

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

 * STEP 2 *

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

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

 * STEP 3 *

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

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

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

* STEP 4 *

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

* STEP 5 *

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

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DR>).
2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

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

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

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

2.1.2 SAMPLE RUN-THROUGH

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

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

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN)
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURRED. NO QUESTIONS ASKED.)
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT)

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 CZRLLC	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. C APR-79	D
CZRLC-B-0	D
(CZRLC VERIFIES INTERCHANGEABILITY OF	D
CARTRIDGES BETWEEN DRIVES	
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D,O
CHANGE HW (L) ? Y	D,O
# UNITS (D) ? 2	D,O
UNIT 0	D
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ?	D,O
DRIVE TYPE = RL01 (L) Y ?	D,O
UNIT 1	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)
CZRLC HRD ERR 00004 TST 003 SUB 002 PC:004130	
ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D,O

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

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

CZRL EOP 1 D
^C

DR>RESTART/PASS:1 D,0

2.2 CHAIN MODE OPERATION

NOT THIS PROGRAM IS NOT CHAINABLE. CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION.

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

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

TO EXECUTE A CHAIN FILE THE USER TYPES:

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

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

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

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

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT
4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

2.3.2 COMMAND SYNTAX

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

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

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, 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
 ADR EXECUTE AUTODROP CODE
 LOT LOOP ON TEST
 EVL EVALUATE

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

 EXIT

RETURN TO XXDP+ PROMPT MODE.

 DRO(P)/UNITS:UNIT-LIST

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

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

 ADD/UNITS:UNIT-LIST

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

 PRI(NT)

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

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

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

 FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

 ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT 'BR LEVEL' 5. THE FIRST 4 DRIVES ARE RLO1'S AND THE LAST 4 DRIVES ARE RLO2'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 0
 BUS ADDRESS (D) 174400 ?
 VECTOR (D) 160 ?
 DRIVE (D) 0 ? 0-3
 DRIVE TYPE = RLO1 (L) Y ?

UNIT 4
 BUS ADDRESS (D) 174400 ? 175400
 VECTOR (D) 160 ? 164
 DRIVE (D) 0 ? 0-3
 DRIVE TYPE = RLO1 (L) Y ? N

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CSR ADDRESS OF THE CONTROLLER (QUESTION #1), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #2), AND THE DRIVE TYPE (QUESTION #4). THE ACTUAL UNIT NUMBERS OF THE RLO1'S FOR QUESTION #3 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS THE FIRST 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 #2. THE RLO2 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #3 AND THE DRIVE TYPE WAS SET FOR RLO2'S FOR THE REMAINING 4 UNITS IN QUESTION #4.

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 FLEXABILITY 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 AT 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
 RRRRRR 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 COMPATABILITY.

'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 (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)

BIT 9 - VOLUME CHECK (VC)
 BIT 8 - DRIVE SELECT ERROR (DSE)
 BIT 7 - DRIVE TYPE IS 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.

a

CZRLCO RL01/02 DRIVE COMPAT MACRO V04.00 16-FEB-82 13:32:06
TABLE OF CONTENTS

2-	8	MACRO DEFINITIONS
2-	36	GLOBAL EQUATES SECTION
3-	2	GLOBAL DATA SECTION
5-	1	GLOBAL TEXT SECTION
5-	35	GLOBAL ERROR REPORT SECTION
7-	1	INITIALIZATION SECTION
9-	1	GLOBAL SUBROUTINES SECTION
27-	51	CONTROL ROUTINE

```

1      .TITLE  CZRLLC0 RL01/02 DRIVE COMPAT
2
3      .ENABLE AMA
4      .ENABLE ABS
5      .MCALL  SVC
6      .=2000
7
8
9      .SBTTL  MACRO DEFINITIONS
10     .MACRO  WAITUS  ARG          :MACRO MICRO-SECOND WAIT
11           MOV      ARG,XDELAY  :SAVE ARGUMENT
12           JSR      PC,TIME     :CALL TIMING ROUTING
13     .ENDM
14
15     .MACRO  WAITMS  ARG          :MACRO MILLI-SECOND WAIT
16           MOV      ARG,YDELAY  :SAVE ARGUMENT
17           JSR      PC,XTIME    :CALL TIMING ROUTINE
18     .ENDM
19
20     .NLIST  CND,MD,ME
21
22     SVC
23     SVCINS=0
24     SVCTAG=0
25
26     POINTER NONE
27
28     BGNMOD  MDHEDR
29     HEADER  CZRLLC0,0,0,1
30     .ASCII  /C/
31     .ASCII  /Z/
32     .ASCII  /R/
33     .ASCII  /L/
34     .ASCII  /L/
35     .BYTE   0
36     .BYTE   0
37     .BYTE   0
38     .ASCII  /C/
39     .ASCII  /O/
40     .WORD   0
41     .WORD   0
42     .WORD   LSHARD
43     .WORD   0
44     .WORD   LSHW
45     .WORD   0
46     .WORD   LSLAST
47     .WORD   0
48     .WORD   0
49     .WORD   0
50     .WORD   1
51     .WORD   0
52     .WORD   LSDISPATCH
53     .WORD   0
54     .WORD   0
55     .WORD   0
56     .BYTE   CSREVISION
57     .BYTE   CSREDIT
58     .WORD   0

```

```

002054 000000 .WORD 0
002056 000000 .WORD 0
002060 002222 .WORD L$DVTYP
002062 000000 .WORD 0
002064 000000 .WORD 0
002066 000000 .WORD 0
002070 000000 .WORD 0
002072 000000 .WORD 0
002074 000000 .WORD 0
002076 002122 .WORD L$DESC
002100 104035 EMT ESLOAD
002102 000000 .WORD 0
002104 022466 .WORD L$INIT
002106 024316 .WORD L$CLEAN
002110 024312 .WORD L$AUTO
002112 022440 .WORD L$PROT
002114 000000 .WORD 0
002116 000000 .WORD 0
002120 000000 .WORD 0
30 002122 ENDMOD
31
32 002122 DESCRIPT <CZRL VERIFIES INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES>
002122 103 132 122 .ASCIZ /CZRL VERIFIES INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES/
002125 114 114 040
002130 126 105 122
002133 111 106 111
002136 105 123 040
002141 111 116 124
002144 105 122 103
002147 110 101 116
002152 107 105 101
002155 102 111 114
002160 111 124 131
002163 040 117 106
002166 040 103 101
002171 122 124 122
002174 111 104 107
002177 105 123 040
002202 102 105 124
002205 127 105 105
002210 116 040 104
002213 122 111 126
002216 105 123 000
.EVEN
33
34 002222 DEVTYP <RL01,RL02>
002222 122 114 060 .ASCIZ /RL01,RL02/
002225 061 054 122
002230 114 060 062
002233 000
.EVEN
35
36 .SBTTL GLOBAL EQUATES SECTION
37
38 ;DEFINITIONS
39
40 002234 BGNMCD GLBEQAT
    
```


41
42 002234

EQUALS

: BIT DEFINITIONS

100000	BIT15==	100000
040000	BIT14==	40000
020000	BIT13==	20000
010000	BIT12==	10000
004000	BIT11==	4000
002000	BIT10==	2000
001000	BIT09==	1000
000400	BIT08==	400
000200	BIT07==	200
000100	BIT06==	100
000040	BIT05==	40
000020	BIT04==	20
000010	BIT03==	10
000004	BIT02==	4
000002	BIT01==	2
000001	BIT00==	1

001000	BIT9==	BIT09
000400	BIT8==	BIT08
000200	BIT7==	BIT07
000100	BIT6==	BIT06
000040	BIT5==	BIT05
000020	BIT4==	BIT04
000010	BIT3==	BIT03
000004	BIT2==	BIT02
000002	BIT1==	BIT01
000001	BIT0==	BIT00

: EVENT FLAG DEFINITIONS
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START==	32.	: START COMMAND WAS ISSUED
000037	EF.RESTART==	31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE==	30.	: CONTINUE COMMAND WAS ISSUED
000035	EF.NEW==	29.	: A NEW PASS HAS BEEN STARTED
000034	EF.PWR==	28.	: A POWER-FAIL/POWER-!P OCCURRED

: PRIORITY LEVEL DEFINITIONS

000340	PRI07==	340
000300	PRI06==	300
000240	PRI05==	240
000200	PRI04==	200
000140	PRI03==	140
000100	PRI02==	100
000040	PRI01==	40
000000	PRI00==	0

: OPERATOR FLAG BITS

000004	EVL==	4
--------	-------	---

```

000010      LOT==      10
000020      ADR==      20
000040      IDU==      40
000100      ISR==     100
000200      UAM==     200
000400      BOE==     400
001000      PNT==    1000
002000      PRI==    2000
004000      IXE==    4000
010000      IBE==   10000
020000      IER==   20000
040000      LOE==   40000
100000      HOE==  100000

43
44      000000      CS=0      ;CONTROL AND STATUS OFFSET
45      000002      BA=2      ;BUSS ADDRESS OFFSET
46      000004      DA=4      ;DISK ADDRESS OFFSET
47      000006      MP=6      ;MULTI PURPOSE OFFSET
48
49      ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
50
51      000000      CSR=0      ;CONTROLLER ADDRESS
52      000002      VEC=2      ;VECTOR OF CONTROLLER
53      000004      DSB=4      ;DRIVE SELECT
54      000006      PAT=6      ;PATTERN UNIQUE TO DRIVE
55
56
57      000001      DRDY=BIT0   ;DRIVE READY
58      000100      INTEN=BIT6  ;INTERRUPT ENABLE
59      100000      ERR=BIT15   ;COMPOSITE ERROR
60      040000      DERR=BIT14  ;DRIVE ERROR
61      020000      NXM=BIT13   ;NON-EXISTANT MEMORY ERROR
62      010000      DLT=BIT12   ;DATA LATE
63      004000      DCRC=BIT11  ;DATA CRC ERROR
64      004000      HCRC=BIT11  ;HEADER CRC ERROR
65      010000      HNF=BIT12   ;HEADER NOT FOUND ERROR
66      002000      OPI=BIT10   ;OPERATION INCOMPLETE ERROR
67      000200      CRDY=BIT7   ;CONTROLLER READY
68      000040      BA17=BIT5   ;EXTENDED BUS ADDRESS BIT 17
69      000020      BA16=BIT4   ;EXTENDED BUS ADDRESS BIT 16
70      000002      CRSET=BIT1  ;CONTROLLER RESET FUNCTION CODE
71      000004      GSTAT=BIT2  ;GET DRIVE STATUS FUNCTION CODE
72      000006      SEEK=BIT1!BIT2 ;SEEK FUNCTION CODE
73      000010      RDHDR=BIT3  ;READ HEADER FUNCTION CODE
74      000012      WRITE=BIT3!BIT1 ;WRITE FUNCTION CODE
75      000014      READ=BIT3!BIT2 ;READ FUNCTION CODE
76      000013      DRST=BIT3!BIT1!BIT0 ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
77      000003      GSBIT=BIT1!BIT0 ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
78      000001      MK=BIT0     ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
79      000004      SIGN=BIT2   ;DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
80      000020      SKHS=BIT4   ;HEAD SELECT FOR SEEK
81      000100      HEAD=BIT6   ;HEAD SELECT FOR READ,WRITE,GET STATUS
82
83      ;OFFSET FOR HARDWARE P-TABLE
84
85      000000      CSR= 0      ;BUS ADDRESS
86      000002      VECT= 2     ;VECTOR ADDRESS
    
```

87	000004	PRIOR= 4	;PRIORITY (BREAK LEVEL)
88	C00006	TYPDR= 6	;DRIVE TYPE
89	000010	DRBT= 10	;DRIVE SELECT
90			
91	002234	ENDMOD	
92			
93			
94			

```

1
2
3
4 002234
5
6 002234 000000
7
8
9
10
11 002236 000000
12 002240 000000
13 002242 000000
14 002244 000000
15 002246 000000
16 002250 000000
17 002252 000000
18 002254 000000
19 002256 000000
20 002260 000000
21 002262 000000
22 002264 000000
23 002266 000000
24 002270 000000
25 002272 000000
26 002274 000000
27 002276 000000
28 002300 000000
29 002302 000000
30 002304 000000
31 002306 000000
32 002310 000000
33 002312 000000
34 002314 000000
35 002316 000000
36 002320 000000
37 002322 000000
38 002324 000000
39 002326 000000
40 002330 000000
41 002332 000000
42 002334 000000
43 002336 000000
44 002340 000000
45 002342 000000
46 002344 000000
47 002346 000000
48 002350 000000
49 002352 000000
50 002354 000000
51 002356 000000
52 002360 000000
53 002362 000000
54 002364 000000
55 002366 000000
56 002370 000000
57 002372 000000

.SBTTL GLOBAL DATA SECTION
BGNMOD GLBDAT
HDRFND: .WORD 0 ;1=HEADER IN BAD SECTOR LIST

;HERE IS THE LIST OF TRACKS TO USE FOR THIS TEST
;TRACKS ARE ENTERED BY 'FNDTRK' ROUTINE & 'FIXTRK' ROUTINE

OUT10: .WORD 0 ;OUTER TRK HEAD 0
OUT20: .WORD 0
OUT30: .WORD 0
OUT40: .WORD 0
OUT50: .WORD 0
OUT11: .WORD 0 ;OUTER TRK HEAD 1
OUT21: .WORD 0
OUT31: .WORD 0
OUT41: .WORD 0
OUT51: .WORD 0
OQU10: .WORD 0 ;1ST QUARTER TRK HEAD 0
OQU20: .WORD 0
OQU30: .WORD 0
OQU40: .WORD 0
OQU50: .WORD 0
OQU11: .WORD 0 ;1ST QUARTER TRK HEAD 1
OQU21: .WORD 0
OQU31: .WORD 0
OQU41: .WORD 0
OQU51: .WORD 0
MID10: .WORD 0 ;MIDDLE TRK HEAD 0
MID20: .WORD 0
MID30: .WORD 0
MID40: .WORD 0
MID50: .WORD 0
MID11: .WORD 0 ;MIDDLE TRK HEAD 1
MID21: .WORD 0
MID31: .WORD 0
MID41: .WORD 0
MID51: .WORD 0
TQU10: .WORD 0 ;3RD QUARTER TRK HEAD 0
TQU20: .WORD 0
TQU30: .WORD 0
TQU40: .WORD 0
TQU50: .WORD 0
TQU11: .WORD 0 ;3RD QUARTER TRK HEAD 1
TQU21: .WORD 0
TQU31: .WORD 0
TQU41: .WORD 0
TQU51: .WORD 0
INN10: .WORD 0 ;INNER TRK HEAD 0
INN20: .WORD 0
INN30: .WORD 0
INN40: .WORD 0
INN50: .WORD 0
INN11: .WORD 0 ;INNER TRK HEAD 1
INN21: .WORD 0
    
```

58 002374 000000
59 002376 000000
60 002400 000000
61
62
63
64
65
66 002402
67
68
69
70 002442
71
72
73
74
75
76 003022 002242
77 003024 002266
78 003026 002312
79 003030 002336
80 003032 002362
81 003034 002254
82 003036 002300
83 003040 002324
84 003042 002350
85 003044 002374
86
87 003046 152525
88 003050 133333
89 003052 066666
90 003054 155555
91

INN31: .WORD 0
INN41: .WORD 0
INN51: .WORD 0
.EVEN

:SECTOR LIST FOR LAST DRIVE WRITTEN
:MAP OF 16 SECTOR DRIVE BITS

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.

OVWTRK: OUT30
OQU30
MID30
TQU30
INN30
OUT31
OQU31
MID31
TQU31
INN31

PATLST: .WORD 152525
.WORD 133333
.WORD 066666
.WORD 155555

1					
2	003056	000000	TEM:	.WORD	0
3	003060	000000	T.DRIVE:	.WORD	0
4	003062	000000	FOWR:	.WORD	0
5	003064	000000	FADJ:	.WORD	0
6	003066	000000	TEMP:	.WORD	0
7	003070	000000	LSTCLR:	.WORD	0
8	003072	000000	REASON:	.WORD	0
9	003074	000000	ERFLG:	.WORD	0
10	003076	000000	STFLG:	.WORD	0
11	003100	000000	ADJLOC:	.WORD	0
12	003102	000000	ADJFLG:	.WORD	0
13	003104	000000	ADJDIR:	.WORD	0
14	003106	000000	DRSTAT:	.WORD	0
15	003110	000000	HSFLG:	.WORD	0
16	003112	000000	OSECT:	.WORD	0
17	003114	000000	HEAD01:	.WORD	0
18	003116	000000	DIRC:	.WORD	0
19	003120	000000	SURF:	.WORD	0
20	003122	000000	CYL:	.WORD	0
21	003124	000000	REVSK:	.WORD	0
22	003126	000000	FORSK:	.WORD	0
23	003130	000000	UUT:	.WORD	0
24	003132	000000	SECT:	.WORD	0
25	003134	000000	LSTDRV:	.WORD	0
26	003136	000000	GDATA:	.WORD	0
27	003140	000000	BDATA:	.WORD	0
28	003142	000000	WCOUNT:	.WORD	0
29	003144	000000	SECWRD:	.WORD	0
30	003146	000000	OFFSET:	.WORD	0
31	003150	000000	LSTTRK:	.WORD	0
32	003152	000000	FRTTRK:	.WORD	0
33	003154	000000	PRSTRK:	.WORD	0
34	003156	000000	SURFACE:	.WORD	0
35	003160	000000	TRKFND:	.WORD	0
36	003162	000000	TRKCNT:	.WORD	0
37	003164	000000	E.CS:	.WORD	0
38	003166	000000	E.BA:	.WORD	0
39	003170	000000	E.DA:	.WORD	0
40	003172	000000	E.MP:	.WORD	0
41	003174	000000	E.MP1:	.WORD	0
42	003176	000000	E.MP2:	.WORD	0
43	003200	000000	BCS:	.WORD	0
44	003202	000000	BBA:	.WORD	0
45	003204	000000	BDA:	.WORD	0
46	003206	000000	BMP:	.WORD	0
47	003210	000000	SERNM1:	.WORD	0
48	003212	000000	SERNM2:	.WORD	0
49	003214	000000	ADJTRK:	.WORD	0
50	003216	000000	ADJUUT:	.WORD	0
51	003220	000000	ADJLC2:	.WORD	0
52	003222	000000	ADJLC3:	.WORD	0
53	003224	000000	ADJLC4:	.WORD	0
54	003226	000000	STSEC1:	.WORD	0
55	003230	000000	STSEC:	.WORD	0
56	003232		BUF:	.BLKW	3072.
57	017232	000000	XDELAY:	.WORD	0

```

:LAST CONTROLLER
:DRIVE ERROR REASON
:ERROR FLAG
:PROGRAM START UP FLAG
:TRACK INDEX FOR ADJ. CYL TEST
:FLAG FOR ADJ. STORE OR RETRIEVE
:ADJACENT SEEK DIRECTION

:SURFACE FLAG
:DIRECTION OF SEEK

:REVERSE SEEK
:FORWARD SEEK
:UNIT UNDER TEST
:SECTOR
:LAST DRIVE
:GOOD DATA
:BAD DATA
:WORD COUNT
:SECTOR WORD
:INCREMENT
:LAST TRACK OF SEARCH
:FIRST TRACK OF SEARCH
:PRESENT TRACK
:SURFACE
:TRACK FOUND
:TRACK COUNT
:IMAGE OF CSF
:IMAGE OF BUS ADDRESS
:IMAGE OF DISK ADDRESS
:IMAGE OF MULTI-PURPOSE WORD 1
:.. .. .. .. .. 2
:.. .. .. .. .. 3
:COMMAND LOADED
:BUS ADDRESS LOADED
:DISK ADDRESS LOADED
:WORD COUNT LOADED
:SERIAL NUMBER OF CATRIDGE
:INSIDE/OUTSIDE FLAG
:UJT FOR "ADJCYL"
:TEMP LOC FOR "ADJCYL"
:.. .. .. ..
:SECTORS TO WRITE "ADJCYL"
:BUFFER FOR 24 SECTOR READS
:DELAY FOR WAIT MICRO-SECOND MACRO
    
```

```
58 017234 000000      YDELAY: .WORD 0      :DELAY FOR WAIT MILLI-SECOND MACRO
59 017236 000000      OBUF:  .WORD 0      :RESPONSE BUFFER
60
61
62 017240      DRBUF:      :DRIVE INFCRMATION BUFFERS
63
64
65
66
67
68      000004      .REPT 4.
69
70      017240 000000      CSR      :CONTROLLER ADDRESS
71      017242 000002      VEC      :VECTOR
72      017244 000004      DSB      :DRIVE SELECT BITS
73      017246 000006      PAT      :PATTERN UNIQUE TO DRIVE
74
75
76
77      017250 000000      CSR      :CONTROLLER ADDRESS
78      017252 000002      VEC      :VECTOR
79      017254 000004      DSB      :DRIVE SELECT BITS
80      017256 000006      PAT      :PATTERN UNIQUE TO DRIVE
81
82
83
84
85
86
87      017260 000000      CSR      :CONTROLLER ADDRESS
88      017262 000002      VEC      :VECTOR
89      017264 000004      DSB      :DRIVE SELECT BITS
90      017266 000006      PAT      :PATTERN UNIQUE TO DRIVE
91
92
93
94
95
96
97
98
99
100      017270 000000      CSR      :CONTROLLER ADDRESS
101      017272 000002      VEC      :VECTOR
102      017274 000004      DSB      :DRIVE SELECT BITS
103      017276 000006      PAT      :PATTERN UNIQUE TO DRIVE
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

1          .SBTTL GLOBAL TEXT SECTION
2 017302   BGNMOD  GLBTXT
3
4          :GLOBAL TEXT
5
6
10
11 017302   103      117      116  OPR001: .ASCIZ  /CONTINUE TEST?/
12 017321   101      102      117  OPR002: .ASCIZ  /ABOVE CONDITIONS MET/
13 017346   103      117      116  CNTTOT: .ASCIZ  /CONTROLLER TIMED OUT/
14 017373   105      122      122  INITWR: .ASCIZ  /ERROR ON RECOVERING INITIAL WRITE BY FIRST DRIVE /
15 017455   105      122      122  DCKER:  .ASCIZ  /ERROR ON READ/
16 017473   115      111      116  FEW:    .ASCIZ  /MINIMUM OF TWO DRIVES REQUIRED/
17 017532   115      101      130  MANY:   .ASCIZ  /MAXIMUM OF FOUR DRIVES ALLOWED/
18 017571   124      105      123  NONE:   .ASCIZ  /TEST ABORTED - CAN'T FIND ANY GOOD SPOTS/
19 017642   124      122      131  OVMES:  .ASCIZ  /TRYING TO OVERWRITE DRIVE /
20 017675   124      122      131  RECMS:  .ASCIZ  /TRYING TO READ DATA WRITTEN BY DRIVE /
21 017743   103      101      116  ERRFND: .ASCIZ  /CAN'T FIND FIVE ADJACENT TRACKS/
22 020003   117      126      105  OVWER:  .ASCIZ  /OVERWRITE ERROR/
23 020023   122      105      101  RECER:  .ASCIZ  /READ RECOVERY ERROR/
24 020047   105      122      122  FUNERR: .ASCIZ  /ERROR IN SEEK OPERATION/
25 020077   115      111      123  SKER:   .ASCIZ  /MIS SEEK ERROR/
26 020116   106      117      122  FWD:    .ASCIZ  /FORWARD/
27 020126   122      105      126  REV:    .ASCIZ  /REVERSE/
28 020136   105      122      122  WRIT1:  .ASCIZ  /ERROR WRITING SECTOR/
29 020163   105      122      122  READ1:  .ASCIZ  /ERROR READING SECTOR/
30 020210   101      104      112  ADJTXT: .ASCIZ  /ADJACENT CYLINDER TEST/
31          .EVEN
32
33 020240   ENDMOD
34
35          .SBTTL GLOBAL ERROR REPORT SECTION
36
37 020240   BGNMOD  GLBERR
38
39 020240   BGNMSG  ERR1
40
41 020240   PRINTB  #FRM10,FRTTRK,LSTTRK,SURFACE ;BETWEEN _ _ HEAD _
    020240   013746  003156  MOV      SURFACE,-(SP)
    020244   013746  003150  MOV      LSTTRK,-(SP)
    020250   013746  003152  MOV      FRTTRK,-(SP)
    020254   012746  021545  MOV      #FRM10,-(SP)
    020260   012746  000004  MOV      #4,-(SP)
    020264   010600  TRAP     SP,R0
    020266   104414  TRAP     C$PNTB
    020270   062706  000012  ADD      #12,SP
42
43 020274   ENDMMSG
    020274   L10000:
    020274   104+23  TRAP     C$MSG
44
45 020276   BGNMSG  ERR2
46 020276   PRINTB  #FRM4,CSR(R4),<B,DSB+1(R4)> ;CONTROLLER _ DRIVE _
    020276   005046  CLR      -(SP)
    020300   156416  000005  BISB    DSB+1(R4),(SP)
    020304   016446  000000  MOV     CSR(R4),-(SP)
    020310   012746  021246  MOV     #FRM4,-(SP)
    
```


	020314	012746	000003	MOV	#3,-(SP)	
	020320	010600		MOV	SP,R0	
	020322	104414		TRAP	C\$PNTB	
	020324	062706	000010	ADD	#10,SP	
47	020330	004737	026522	JSR	PC,REGDMP	:REGISTER DUMP ROUTINE
48	020334			ENDMSG		
	020334	104423		L10001: TRAP	C\$MSG	
49				BGNMSG	ERR3	
50	020336			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	:CONTROLLER _ DRIVE _
51	020336			CLR	-(SP)	
	020336	005046		BISB	DSB+1(R4),(SP)	
	020340	156416	000005	MOV	CSR(R4),-(SP)	
	020344	016446	000000	MOV	#FRM4,-(SP)	
	020350	012746	021246	MOV	#3,-(SP)	
	020354	012746	000003	MOV	SP,R0	
	020360	010600		TRAP	C\$PNTB	
	020362	104414		ADD	#10,SP	
	020364	062706	000010	JSR	PC,REGDMP	:REGISTER DUMP ROUTINE
52	020370	004737	026522	PRINTB	#FRM5,<SURF>,<CYL>,SECT	:HEAD _ CYLINDER _ SECTOR _
53	020374			MOV	SECT,-(SP)	
	020374	013746	003132	MOV	CYL,-(SP)	
	020400	013746	003122	MOV	SURF,-(SP)	
	020404	013746	003120	MOV	#FRM5,-(SP)	
	020410	012746	021307	MOV	#4,-(SP)	
	020414	012746	000004	MOV	SP,R0	
	020420	010600		TRAP	C\$PNTB	
	020422	104414		ADD	#12,SP	
	020424	062706	000012	PRINTB	#FRM16,CSR(R3),<B,DSB+1(R3)>	:ADJACENT WRITTEN BY CONTROLLER :_ DRIVE _
54	020430			CLR	-(SP)	
55	020430	005046		BISB	DSB+1(R3),(SP)	
	020432	156316	000005	MOV	CSR(R3),-(SP)	
	020436	016346	000000	MOV	#FRM16,-(SP)	
	020442	012746	022076	MOV	#3,-(SP)	
	020446	012746	000003	MOV	SP,R0	
	020452	010600		TRAP	C\$PNTB	
	020454	104414		ADD	#10,SP	
	020456	062706	000010	ENDMSG		
56	020462			L10002: TRAP	C\$MSG	
57	020462			BGNMSG	ERR4	
	020462	104423		PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	:CONTROLLER _ DRIVE _
58				CLR	-(SP)	
59	020464			BISB	DSB+1(R4),(SP)	
60				MOV	CSR(R4),-(SP)	
61	020464			MOV	#FRM4,-(SP)	
	020464	005046		MOV	#3,-(SP)	
	020466	156416	000005	MOV	SP,R0	
	020472	016446	000000	TRAP	C\$PNTB	
	020476	012746	021246	ADD	#10,SP	
	020502	012746	000003	JSR	PC,REGDMP	:REGISTER DUMP ROUTINE
	020506	010600		PRINTB	#FRM5,<SURF>,<CYL>,SECT	:HEAD _ CYLINDER _ SECTOR _
	020510	104414				
	020512	062706	000010			
62	020516	004737	026522			
63	020522					

	020522	013746	003132	MOV	SECT,-(SP)	
	020526	013746	003122	MOV	CYL,-(SP)	
	020532	013746	003120	MOV	SURF,-(SP)	
	020536	012746	021307	MOV	#FRM5,-(SP)	
	020542	012746	000004	MOV	#4,-(SP)	
	020546	010600		MOV	SP,R0	
	020550	104414		TRAP	C\$PNTB	
	020552	062706	000012	ADD	#12,SP	
64	020556			PRINTB	#FRM6,REASON,LSTDRV,LSTCLR,LSTDRV	
	020556	013746	003134	MOV	LSTDRV,-(SP)	
	020562	013746	003070	MOV	LSTCLR,-(SP)	
	020566	013746	003134	MOV	LSTDRV,-(SP)	
	020572	013746	003072	MOV	REASON,-(SP)	
	020576	012746	021356	MOV	#FRM6,-(SP)	
	020602	012746	000005	MOV	#5,-(SP)	
	020606	010600		MOV	SP,R0	
	020610	104414		TRAP	C\$PNTB	
	020612	062706	000014	ADD	#14,SP	
65	020616			PRINTB	#FRM7,DIRC	;SFEK DIRECTION
	020616	013746	003116	MOV	DIRC,-(SP)	
	020622	012746	021377	MOV	#FRM7,-(SP)	
	020626	012746	000002	MOV	#2,-(SP)	
	020632	010600		MOV	SP,R0	
	020634	104414		TRAP	C\$PNTB	
	020636	062706	000006	ADD	#6,SP	
66						
67	020642			ENDMSG		
	020642			L10003:		
	020642	104423		TRAP	C\$MSG	
68						
69	020644			BGNMSG	ERR5	
70	020644			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	;CONTROLLER _ DRIVE _
	020644	005046		CLR	-(SP)	
	020646	156416	000005	BISB	DSB+1(R4),(SP)	
	020652	016446	000000	MOV	CSR(R4),-(SP)	
	020656	012746	021246	MOV	#FRM4,-(SP)	
	020662	012746	000003	MOV	#3,-(SP)	
	020666	010600		MOV	SP,R0	
	020670	104414		TRAP	C\$PNTB	
	020672	062706	000010	ADD	#10,SP	
71	020676	004737	026522	JSR	PC,REGDMP	
72	020702			ENDMSG		
	020702			L10004:		
	020702	104423		TRAP	C\$MSG	
73						
74	020704			BGNMSG	ERR6	
75	020704			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)>	
	020704	005046		CLR	-(SP)	
	020706	156416	000005	BISB	DSB+1(R4),(SP)	
	020712	016446	000000	MOV	CSR(R4),-(SP)	
	020716	012746	021246	MOV	#FRM4,-(SP)	
	020722	012746	000003	MOV	#3,-(SP)	
	020726	010600		MOV	SP,R0	
	020730	104414		TRAP	C\$PNTB	
	020732	062706	000010	ADD	#10,SP	
76	020736	004737	026522	JSR	PC,REGDMP	
77	020742			PRINTB	#FRM17,R1,E.MP	

020742	013746	003172	MOV	E.MP, -(SD)
020746	010146		MOV	R1, -(SP)
020750	012746	022163	MOV	#FRM17, (SP)
020754	012746	000003	MOV	#3, -(SP)
020760	010600		MOV	SP, R0
020762	104414		TRAP	CSPNTR
020764	062706	000010	ADD	#10, S
78 020770			ENDMSG	
020770			L10005:	
020770	104423		TRAP	C\$MS;

;FORMAT STATEMENTS

88 020772	045	116	045	FRM1:	.ASCIZ	/%N%AUNLOAD DRIVE %01%A ON CONTROLLER %06%A AND REMOVE PACK%N/
89 021067	045	116	045	FRM2:	.ASCIZ	/%N%APLACE PACK IN DRIVE %01%A ON CONTROLLER %06%A AND LOAD IT%N/
90 021167	045	116	045	FRM3:	.ASCIZ	!%N%A WRONG PACK # IS %05%05%A # S/B %05%05%A%N!
91 021246	045	101	103	FRM4:	.ASCIZ	/%A CONTROLLER: %06%A DRIVE: %01%A%N/
92 021307	045	101	110	FRM5:	.ASCIZ	/%A HEAD: %01%A CYL: %Z3%A SECTOR: %Z2%A%N/
93 021356	045	124	045	FRM6:	.ASCIZ	/%T%01%A ON %06%A%N/
94 021377	045	101	123	FRM7:	.ASCIZ	/%A SEEK DIRECTION: %T%N%A DATA: %N/
95 021437	045	101	127	FRM8:	.ASCIZ	!%A WORD: %Z3%A S/B: %06%A WAS: %06%A%N!
96 021503	045	104	063	FRM9:	.ASCIZ	/%D3%A WORDS BAD OUT OF 128 READ%N/
97 021545	045	101	102	FRM10:	.ASCIZ	/%A BETWEEN %Z3%A - %Z3%A HEAD: %01%A%N/
98 021611	045	116	045	FRM11:	.ASCIZ	/%N%A PWR FAIL NOT SUPPORTED%N/
99 021646	045	101	102	FRM12:	.ASCIZ	/%A BEFORE CS: %06%A BA: %06%A DA: %06%A MP: %06%A%N/
100 021725	045	116	045	FRM13:	.ASCIZ	/%N%A AFTER CS: %06%A BA: %06%A DA: %06%A MP: %06%A%N/
101 022010	045	116	045	FRM14:	.ASCIZ	/%N%A DRIVE STATUS: %06%A%N/
102 022037	045	116	045	FRM15:	.ASCIZ	/%N%A CAN'T FIND BAD SECTOR FILE%N/
103 022076	045	101	101	FRM16:	.ASCIZ	/%A ADJACENT WRITTEN BY CONTROLLER: %06%A DRIVE: %01%A%N/
104 022163	045	101	105	FRM17:	.ASCIZ	/%A EXP'D: %06%A REC'D: %06%A%N/
105 022217	045	116	045	FRM18:	.ASCIZ	/%N%A UNLOAD AND WRITE ENABLE ALL DRIVES TO BE USED%N/
106 022303	045	116	045	FRM19:	.ASCIZ	/%N%A DRIVE TYPE IS DIFFERENT%N/
107 022341	045	116	045	FRM20:	.ASCIZ	/%N%A DRIVE NUMBER PREVIOUSLY SPECIFIED%N/
108 022412	045	116	045	ENDPAS:	.ASCIZ	/%N%A END OF TEST%N%N/
109						
113				.EVEN		
114						
115 022440				ENDMOD		
116						

```

1
2
3      ;LOAD PROTECTION TABLE
4 022440      BGNPROT
5
6 022440 000000      .WORD 0      ;OFFSET OF CSR IN P-TABLE
7 022442 177777      .WORD -1     ;NOT A MASS-BUS DRIVE
8 022444 000006      .WORD 6      ;OFFSET OF DRIVE IN P-TABLE
9
10 022446      ENDPROT
11
12
13 022446      BGNMOD HPTCODE
14 022446      BGNHW
15 022446 000005      .WORD L10007-LSHW/2
16 022450 174400      .WORD 174400    ;BASE ADDRESS DEFAULT
17 022452 000160      .WORD 160      ;VECTOR DEFAULT
18 022454 000240      .WORD 240      ;PRIORITY DEFAULT
19 022456 000001      .WORD 1      ;RL01 OR R'02 (RL01=1)
20 022460 000000      .WORD 0      ;DRIVE NUMBER DEFAULT
21 022462      ENDDHW
22 022462      L10007:
23      ENDMOD
24
25 022462      BGNMOD DSPCODE
26
27 022462      DISPATCH 1
28 022462 000001      .WORD 1
29 022464 032706      .WORD T1
30
31      ENDMOD
    
```

```

1          .SBTTL INITIALIZATION SECTION
2
3 022466   BGNMOD  INITCODE
4
5 022466   BGNINIT
6
7 022466   SETPRI  #340
   022466   MOV     #340,R0
   022472   TRAP   C$SPRI
8
9 022474   023727 002012 000002   CMP     LSUNIT,#2           ;MORE THAN TWO
10 022502   002006   BGE     90$                ;YES, OKAY
11
12 022504   ERRSF  19.,FEW           ;MINIMUM OFF TWO DRIVE REQUIRED
   022504   TRAP   C$ERSF
   022506   .WORD  19
   022510   .WORD  FEW
   022512   .WORD  0
13 022514   000137 024266   JMP     CMPENA            ;CLEAN CODE WHEN < 2 DRIVES
14
15 022520   023727 002012 000004 90$:  CMP     LSUNIT,#4           ;MORE THAN FOUR
16 022526   003406   BLE     91$                ;NO, OKAY
17
18 022530   ERRSF  20.,MANY           ;MAXIMUM OF FOUR DRIVES ALLOWED
   022530   TRAP   C$ERSF
   022532   .WORD  20
   022534   .WORD  MANY
   022536   .WORD  0
19 022540   000137 024266   JMP     CMPENA            ;CLEAN CODE WHEN > 4 DRIVES
20
21 022544   013737 002012 003130 91$:  MOV     LSUNIT,UUT         ;GET NUMBER OF UNITS
22 022552   005001   CLR     R1                 ;INIT P-TABLE
23 022554   012704 017240   MOV     #DRBUF,R4          ;SET UP DRIVE BUFFER
24 022560   012702 003046   MOV     #PATLST,R2        ;GET LIST OF PATTERNS
25 022564   005737 003130 1$:   TST     UUT                ;ANY P-TABLES LEFT?
26 022570   001513   BEQ    END                 ;NO, GO TO END
27 022572   GPHARD  R1,R0              ;GET A P-TABLE
   022572   MOV     R1,R0
   022574   TRAP   C$GPHRD
28 022576   012064 000000   MOV     (R0)+,CSR(R4)      ;GET CSR
29 022602   012064 000002   MOV     (R0)+,VEC(R4)     ;GET VECTOR
30 022606   012064 000004   MOV     (R0)+,PRIOR(R4)   ;GET BREAK LEVEL
31 022612   012037 003060   MOV     (R0)+,T.DRIVE     ;RL01/2 TYPE ... RL01=1
32 022616   011064 000004   MOV     (R0),DSB(R4)      ;GET DRIVE
33 022622   011264 000006   MOV     (R2),PAT(R4)
34 022626   005722   TST     (R2)+
35
36                                     ;TEST FOR DRIVES OF SAME TYPE AND NO REPEATED DRIVE NUMBERS
37
38 022630   023737 002012 003130   CMP     LSUNIT,UUT         ;SKIP TEST FOR FIRST DRIVE
39 022636   001462   BEQ    6$
40
41 022640   GPHARD  #0,R5              ;BASE ADDRESS OF FIRST P TABLE
   022640   MOV     #0,R0
   022644   TRAP   C$GPHRD
   022646   MOV     R0,R5
42
    
```



```

023056 012746 000001      MOV    #1,-(SP)
023062 010600      MOV    SP,R0
023064 104417      TRAP  C$PNTF
023066 062706 000004      ADD   #4,SP

72
73
74      :INITIALIZE ROUTINE
75      :WE ATTEMPT TO LOCATE 5 PERFECT ADJACENT TRACKS AT 5 SPOTS
76      :ACROSS THE PACK.
77      :THE 5 SPOTS ARE: (EACH SURFACE)
78      :
79      :OUTER - TRACK 0 - 16 (BOTH RL01 & RL02)
80      :INNER - TRACK 238 - 254 (RL01) OR 494 - 510 (RL02)
81      :MIDDLE - TRACK 120 - 136 (RL01) OR 248 - 264 (RL02)
82      :ONE QUARTER - TRACK 56 - 72 (RL01) OR 120 - 136 (RL02)
83      :THREE QUARTER - TRACK 184 - 200 (RL01) OR 376 - 392 (RL02)
84      :
85      :IF WE FIND ANY BAD SPCTS, WE WILL REPORT SO.....
86
87
88 023072 005237 00307c      SETUP:  INC    STFLG      :INDICATE A START COMMAND
89 023076 012737 177777 003210      MOV    #-1,SERNM1
90 023104 012737 177777 003212      MOV    #-1,SERNM2
91 023112      1$:  PRINTF  #FRM18      :PROMPT - UNLOAD DRIVES TO BE USED
      023112 012746 022217      MOV    #FRM18,-(SP)
      023116 012746 000001      MOV    #1,-(SP)
      023122 010600      MOV    SP,R0
      023124 104417      TRAP  C$PNTF
      023126 062706 000004      ADD   #4,SP
92 023132      GMANIL  OPR002,OBUFF,1, NO  :PROMPT - ABOVE CONDITIONS MET
      023132 104443      TRAP  C$GMAN
      023134 000404      BR    10002$
      023136 017236      .WORD OBUFF
      023140 000120      .WORD T$CODE
      023142 017321      .WORD OPR002
      023144 000001      .WORD 1
      023146      10002$:
93 023146 005737 017236      TST   OBUFF      :NO - ASK AGAIN
94 023152 001757      BEQ   1$
95
96 023154 004537 032300      JSR   R5,LOAD      :TELL OPERATOR TO LOAD
97 023160 004537 031526      JSR   R5,SERNUM    :GET SERIAL NUMBER
98 023164 004537 031002      JSR   R5,MERGE     :MERGE BAD SECTOR FILES
99 023170 012701 002236      MOV   #OUT10,R1   :INITIALIZE ALL TRACKS
100 023174 012700 000062      MOV   #50,R0
101 023200 012721 177777      3$:  MOV   #177777,(R1)+
102 023204 005300      DEC   R0
103 023206 001374      BNE   3$
104
105 023210 004537 031230      JSR   R5,FNDTRK   :TRY TO FIND FIVE TRACKS
106 023214 000001      1
107 023216 000000      0      :INWARD SEARCH
      :TOP SURFACE
108
109 023220 000000 000020      .WORD 0,16.
110 023224 000000 000020      .WORD 0,16.
111
112 023230 005737 003160      TST   TRKFND      :WAS SEARCH SUCCESSFUL????
    
```

113	023234	001005		BNE	5\$:YES
114							
115	023236			ERRHRD	10.,ERRFND,ERR1		:CAN'T FIND 5 ADJACENT TRACKS
	023236	104456		TRAP	C\$ERHRD		
	023240	000012		.WORD	10		
	023242	017743		.WORD	ERRFND		
	023244	020240		.WORD	ERR1		
116	023246	000404		BR	7\$		
117							
118	023250	012700	002236	5\$:	MOV	#OUT10,R0	:STORE AWAY TRACKS FOUND
119	023254	004537	031472		JSR	R5,FIXCYL	
120							
121	023260	004537	031230	7\$:	JSR	R5,FNDTRK	:TRY TO FIND FIVE TRACKS
122	023264	000001			1		:INWARD SEARCH
123	023266	000001			1		:BOTTOM SURFACE
124	023270	000000	000020		.WORD	0,16.	
125	023274	000000	000020		.WORD	0,16.	
126							
127	023300	005737	003160		TST	TRKFND	:WAS SEARCH SUCCESSFUL???
128	023304	001005			BNE	9\$:YES
129							
130	023306			ERRHRD	10.,ERRFND,ERR1		:CAN'T FIND 5 ADJACENT TRACKS
	023306	104456		TRAP	C\$ERHRD		
	023310	000012		.WORD	10		
	023312	017743		.WORD	ERRFND		
	023314	020240		.WORD	ERR1		
131	023316	000404		BR	10\$		
132							
133	023320	012700	002250	9\$:	MOV	#OUT11,R0	:STORE TRACKS AWAY
134	023324	004537	031472		JSR	R5,FIXCYL	
135	023330	004537	031230	10\$:	JSR	R5,FNDTRK	:FIND NEXT 5 TRACK
136	023334	177777			-1		:OUTWARD SEARCH
137	023336	000000			0		:TOP SURFACE
138	023340	000376	000356		.WORD	254.,238.	:TRACK RANGE
139	023344	000776	000756		.WORD	510.,494.	
140							
141	023350	005737	003160		TST	TRKFND	:WAS SEARCH SUCCESSFUL?
142	023354	001005			BNE	12\$:YES
143							
144	023356			ERRHRD	10.,ERRFND,ERR1		:CAN'T FIND 5 ADJACENT TRACKS
	023356	104456		TRAP	C\$EP 7D		
	023360	000012		.WORD	10		
	023362	017743		.WORD	ERRFND		
	023364	020240		.WORD	ERR1		
145	023366	000404		BR	14\$:SKIP
146							
147	023370	012700	002356	12\$:	MOV	#INN10,R0	:STORE AWAY TRACKS FOUND
148	023374	004537	031472		JSR	R5,FIXCYL	
149							
150	023400	004537	031230	14\$:	JSR	R5,FNDTRK	:NEXT SET
151	023404	177777			-1		:OUTWARD SEARCH
152	023406	000001			1		:BOTTOM SURFACE
153	023410	000376	000356		.WORD	254.,238.	
154	023414	000776	000756		.WORD	510.,494.	
155							
156	023420	005737	003160		TST	TRKFND	:SEARCH SUCCESSFUL?
157	023424	001005			BNE	16\$:YES


```

158
159 023426          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS
      023426 104456 TRAP      C$ERHRD
      023430 000012 .WORD    10
      023432 017743 .WORD    ERRFND
      023434 020240 .WORD    ERR1
160 023436 000404 BR        18$
161
162 023440 012700 002370 16$: MOV    #INN11,R0      ;STORE AWAY TRACKS FOUND
163 023444 004537 031472 JSR      R5,FXCYL
164
165 023450 004537 031230 18$: JSR      R5,FNDTRK      ;NEXT SET
166 023454 000001 1      ;INWARD SEARCH
167 023456 000000 0      ;TOP SURFACE
168 023460 000176 000210 .WORD    126.,136.      ;TRACK RANGE
169 023464 000376 000410 .WORD    254.,264.
170
171 023470 005737 003160 TST      TRKFND          ;DID WE FIND A SET
172 023474 001020 BNE      20$            ;YES
173
174 023476 004537 031230 JSR      R5,FNDTRK      ;NEXT SET (OTHER SIDE)
175 023502 177777 -1      ;OUTWARD SEARCH
176 023504 000000 0      ;TOP SURFACE
177 023506 000202 000170 .WORD    130.,120.      ;TRACK RANGE
178 023512 000402 000370 .WORD    258.,248.
179 023516 005737 003160 TST      TRKFND          ;DID WE FIND A SET
180 023522 001005 BNE      20$            ;YES
181
182 023524          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS
      023524 104456 TRAP      C$ERHRD
      023526 000012 .WORD    10
      023530 017743 .WORD    ERRFND
      023532 020240 .WORD    ERR1
183 023534 000404 BR        22$
184
185 023536 012700 002306 20$: MOV    #MID10,R0      ;STORE AWAY
186 023542 004537 031472 JSR      R5,FXCYL
187 023546 004537 031230 22$: JSR      R5,FNDTRK      ;NEXT SET
188 023552 000001 1      ;INWARD SEARCH
189 023554 000001 1      ;BOTTOM SURFACE
190 023556 000176 000210 .WORD    126.,136.      ;RANGE
191 023562 000376 000410 .WORD    254.,264.
192
193 023566 005737 003160 TST      TRKFND          ;SUCCESS?
194 023572 001020 BNE      24$            ;YES
195
196 023574 004537 031230 JSR      R5,FNDTRK      ;LOOK THE OTHER SIDE
197 023600 177777 -1      ;OUTWARD
198 023602 000001 1      ;BOTTOM SURFACE
199 023604 000202 000170 .WORD    130.,120.
200 023610 000402 000370 .WORD    258.,248.
201
202 023614 005737 003160 TST      TRKFND          ;SUCCESS?
203 023620 001005 BNE      24$            ;YES
204
205 023622          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS
      023622 104456 TRAP      C$ERHRD
    
```

206	023624	000012		.WORD	10	
207	023626	017743		.WORD	ERRFND	
208	023630	020240		.WORD	ERR1	
209	023632	000404		BR	26\$	
210			24\$:	MOV	#MID11,RO	:STORE AWAY THE TRACKS FOUND
211	023634	012700		JSR	R5,FXCYL	
212	023640	004537				
213			26\$:	JSR	R5,FNDTRK	:NEXT SET
214	023644	004537		1		:INWARD
215	023650	000001		0		:TOP SURFACE
216	023652	000000		.WORD	62.,72.	:RANGE
217	023654	000076	000110	.WORD	126.,136.	
218	023660	000176	000210			
219				TST	TRKFND	:SUCCESS?
220	023664	005737	003160	BNE	28\$:YES
221						
222	023672	004537	031230	JSR	R5,FNDTRK	:LOOK OTHER SIDE
223	023676	177777		-1		:OUTWARD
224	023700	000000		0		:TOP SURFACE
225	023702	000102	000070	.WORD	66.,56.	:RANGE
226	023706	000202	000170	.WORD	130.,120.	
227				TST	TRKFND	:SUCCESS?
228	023712	005737	003160	BNE	28\$:YES
229						
230	023720			ERRHRD	10.,ERRFND,ERR1	:CAN'T FIND 5 ADJACENT TRACKS
231	023720	104456		TRAP	C\$ERHRD	
232	023722	000012		.WORD	10	
233	023724	017743		.WORD	ERRFND	
234	023726	020240		.WORD	ERR1	
235	023730	000404		BR	30\$	
236			28\$:	MOV	#OQU10,RO	:STORE AWAY NEXT SET
237	023732	012700		JSR	R5,FXCYL	
238	023736	004537		JSR	R5,FNDTRK	:LOOK FOR NEXT SET
239	023742	004537	031230	1		:INWARD
240	023746	000001		1		:BOTTOM
241	023750	000001		.WORD	62.,72.	:RANGE
242	023752	000076	000110	.WORD	126.,136.	
243	023756	000176	000210			
244				TST	TRKFND	:SUCCESS?
245	023762	005737	003160	BNE	32\$:YES
246						
247	023770	004537	031230	JSR	R5,FNDTRK	:LOOK FOR ANOTHER SET
248	023774	177777		-1		:OUTWARD
249	023776	000001		1		:BOTTOM
250	024000	000102	000070	.WORD	66.,56.	:RANGE
251	024004	000202	000170	.WORD	130.,120.	
252				TST	TRKFND	:SUCCESS?
253	024010	005737	003160	BNE	32\$:YES
254						
255	024016			ERRHRD	10.,ERRFND,ERR1	:CAN'T FIND 5 ADJACENT TRACKS
256	024016	104456		TRAP	C\$ERHRD	
257	024020	000012		.WORD	10	
258	024022	017743		.WORD	ERRFND	

253	024024	020240		.WORD	ERR1	
254	024026	000404		BR	34\$	
255	024030	012700	002274	32\$: MOV	#0QU11,RO	:STORE AWAY TRACKS
256	024034	004537	031472	JSR	R5,FXCYL	
257						
258	024040	004537	031230	34\$: JSR	R5,FNDTRK	:NEXT SET OF TRACKS
259	024044	000001		1		:INWARD
260	024046	000000		0		:TOP SURFACE
261	024050	000276	000310	.WORD	190.,200.	:RANGE
262	024054	000576	000610	.WORD	382.,392.	
263						
264	024060	005737	003160	TST	TRKFND	:SUCCESS?
265	024064	001020		BNE	36\$:YES
266						
267	024066	004537	031230	JSR	R5,FNDTRK	:LOOK OTHER SIDE
268	024072	177777		-1		:OUTWARD SEARCH
269	024074	000000		0		:TOP
270	024076	000302	000270	.WORD	194.,184.	
271	024102	000602	000570	.WORD	386.,376.	
272						
273	024106	005737	003160	TST	TRKFND	:SUCCESS
274	024112	001005		BNE	36\$:YES
275						
276	024114			ERRHRD	10.,ERRFND,ERR1	:CAN'T FIND 5 ADJACENT TRACKS
	024114	104456		TRAP	C\$ERHRD	
	024116	000012		.WORD	10	
	024120	017743		.WORD	ERRFND	
	024122	020240		.WORD	ERR1	
277	024124	000404		BR	38\$	
278						
279	024126	012700	002332	36\$: MOV	#TQU10,RO	:STORE TRACKS AWAY
280	024132	004537	031472	JSR	R5,FXCYL	
281	024136	004537	031230	38\$: JSR	R5,FNDTRK	:NEXT SET
282	024142	000001		1		:INWARD
283	024144	000001		1		:BOTTOM SURFACE
284	024146	000276	000310	.WORD	190.,200.	:RANGE
285	024152	000576	000610	.WORD	382.,392.	
286						
287	024156	005737	003160	TST	TRKFND	:SUCCESS?
288	024162	001020		BNE	40\$:YES
289						
290	024164	004537	031230	JSR	R5,FNDTRK	:OTHER SET
291	024170	177777		-1		:OUTWARD
292	024172	000001		1		:BOTTOM SURFACE
293	024174	000302	000270	.WORD	194.,184.	:RANGE
294	024200	000602	000570	.WORD	386.,376.	
295						
296	024204	005737	003160	TST	TRKFND	:SUCCESS
297	024210	001005		BNE	40\$:YES
298						
299	024212			ERRHRD	10.,ERRFND,ERR1	:CAN'T FIND 5 ADJACENT TRACKS
	024212	104456		TRAP	C\$ERHRD	
	024214	000012		.WORD	10	
	024216	017743		.WORD	ERRFND	
	024220	020240		.WORD	ERR1	
300	024222	000404		BR	42\$	

```

301
302 024224 012700 002344      40$  MOV  #TQU11,RO      ;STORE SET AWAY
303 024230 004537 031472      JSR  R5,FIXCYL
304
305 024234 012700 002236      42$:  MOV  #OUT10,RO      ;DID WE FIND ANY AT ALL
306 024240 012701 000062      MOV  #50.,R1
307 024244 022720 177777      44$:  CMP  #-1,(R0)+
308 024250 001017      BNE  EXIT
309 024252 005301      DEC  R1
310 024254 001373      BNE  44$
311 024256      ERRSF 3.,NONE
      TRAP C$ERSF
      .WORD 3
      .WORD NONE
      .WORD 0
312 024266 005001      CMPENA: CLR R1
313 024270 013700 002012      MOV  LSUNIT,RO
314 024274      48$:  DODU R1      ;DO DROP UNIT
      MOV  R1,RO
      TRAP C$DODU
315 024300 005201      INC  R1
316 024302 005300      DEC  RO
317 024304 001373      BNE  48$
318 024306      DOCLN
      TRAP C$DCLN
      104444
319
320 024310      EXIT:
321 024310      L10010: ENDINIT
      TRAP C$INIT
      104411
322 024312      ENDMOD
323

```

```
1
2 024312          BGNMOD  AUTOCODE          ;AUTO DROP SECTION
3 024312          BGNAUTO
4
5 024312 000240   NOP                      ;DO NOTHING
6
7 024314          ENDAUTO
  024314          L10011:
  024314 104461   TRAP    C$AUTO
8 024316          ENDMOD
9
10
11 024316          BGNMOD  CLNCODE
12 024316          BGNCLN
13
14 024316 000240   NOP
15
16 024320          L10012:
  024320          ENDCLN
  024320 104412   TRAP    C$CLEAN
17 024322          ENDMOD
18
19 024322          BGNMOD  DRPCODE
20 024322          BGNDU
21 024322 000240   NOP
22 024324          ENDDU
  024324          L10013:
  024324 104453   TRAP    C$DU
23 024326          ENDMOD
24
25
26
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

.SBTTL GLOBAL SUBROUTINES SECTION
BGNMOD GLBSUB

:TIMING ROUTINES
:CALL 1:      JSR      PC,TIME
:CALL 2:      JSR      PC,XTIME

024326 012737 000160 002116 TIME:  MOV      #160, LSDLY      ;GET OUTER DELAY LOOP
024334 005437 017232          NEG      XDELAY        ;GET NEGATIVE OF MULTIPLY FACTOR
024340          READBUS          ;Q-BUS?
024340 104407          TRAP     CSRDBU
024342          BCOMPLETE      2$      ;BRANCH - IF YES
024342 103420          BCS      2$
024344          DELAY      1          ;WAIT
024344 012727 000001          MOV      #1,(PC)+
024350 000000          .WORD    0
024352 013727 002116          MOV      LSDLY,(PC)+
024356 000000          .WORD    0
024360 005367 177772          DEC      -6(PC)
024364 001375          BNE      -4
024366 005367 177756          DEC      -22(PC)
024372 001367          BNE      -20
024374 005237 017232          INC      XDELAY        ;WAIT FACTOR EXPIRED?
024400 002761          BLT      1$          ;BRANCH - IF NO
024402 000422          BR       4$          ;EXIT
024404 012737 000150 002116 2$:  MOV      #150,LSDLY      ;GET OUTER DELAY LOOP
024412          3$:  DELAY      1          ;WAIT WITH RESPECT TO FONZ BUS
024412 012727 000001          MOV      #1,(PC)+
024416 000000          .WORD    0
024420 013727 0021 6          MOV      LSDLY,(PC)+
024424 000000          .WORD    0
024426 005367 177772          DEC      -6(PC)
024432 001375          BNE      -4
024434 005367 177756          DEC      -22(PC)
024440 001367          BNE      -20
024442 005237 017232          INC      XDELAY        ;WAIT FACTOR EXPIRED?
024446 002761          BLT      3$          ;BRANCH - IF NO
024450 000207          4$:  RTS      PC          ;RETURN

024452 012737 000160 002116 XTIME: MOV      #160,LSDLY      ;GET OUTER DELAY LOOP
024460 006337 017234          ASL      YDELAY        ;MULTIPLY FACTOR BY 4
024464 006337 017234          ASL      YDELAY
024470 005437 017234          NEG      YDELAY
024474          READBUS          ;-----
024474 104407          TRAP     CSRDBU          ;GET NEGATIVE OF RESULT
024476          BCOMPLETE      1$      ;Q-BUS?
024476 103023          BCC      1$          ;BRANCH - IF NO
024500 012737 000150 002116 2$:  MOV      #150,LSDLY      ;GET OUTER DELAY LOOP
024506          DELAY      20          ;WAIT WITH RESPECT TO FONZ BUS
024506 012727 000020          MOV      #20,(PC)+
024512 000000          .WORD    0
024514 013727 002116          MOV      LSDLY,(PC)+
    
```

	024520	000000		.WORD	0	
	024522	005367	177772	DEC	-6(PC)	
	024526	001375		BNE	.-4	
	024530	005367	177756	DEC	-22(PC)	
	024534	001367		BNE	.-20	
35	024536	005237	017234	INC	YDELAY	:WAIT FACTOR EXPIRED?
36	024542	002761		BLT	2\$:BRANCH - IF NO
37	024544	000417		BR	3\$:EXIT
38	024546			DELAY	50	:WAIT
	024546	012727	000050	MOV	#50,(PC)+	
	024552	000000		.WORD	0	
	024554	013727	002116	MOV	L\$DLY,(PC)+	
	024560	000000		.WORD	0	
	024562	005367	177772	DEC	-6(PC)	
	024566	001375		BNE	.-4	
	024570	005367	177756	DEC	-22(PC)	
	024574	001367		BNE	.-20	
39	024576	005237	017234	INC	YDELAY	:WAIT FACTOR EXPIRED?
40	024602	002761		BLT	1\$:BRANCH - IF NO
41	024604	000207		RTS	PC	:RETURN
42						
43						

```

1
2
3
4
5
6
7 024606 010046
8 024610 010146
9 024612 010246
10 024614 010346
11 024616 005000
12 024620 012537 003126
13 024624 012537 003124
14
15 024630 012701 003022
16 024634 011102
17 024636 021227 177777
18 024642 001500
19
20 024644 005037 003122
21 024650 005037 003120
22 024654 020027 000005
23 024660 002402
24 024662 005237 003120
25 024666 004537 026254
26 024672 005037 003122
27 024676 051237 003122
28 024702 004537 026254
29 024706 013703 003126
30 024712 004537 025070
31 024716 000034
32 024720 012737 020116 003116
33 024726 004537 027200
34 024732 004537 027564
35 024736 005037 003122
36 024742 022737 000001 003060
37 024750 001004
38 024752 052737 000377 003122
39 024760 000403
40 024762 052737 000777 003122
41 024770 004537 026254
42 024774 005037 003122
43 025000 005037 003120
44 025004 051237 003122
45 025010 004537 026254
46
47 025014 013703 003124
48 025020 004537 025070
49 025024 000034
50 025026 012737 020126 003116
51 025034 004537 027200
52 025040 004537 027564
53
54 025044 005721
55 025046 005200
56 025050 020027 000012
57 025054 001267

:ROUTINE TO PERFORM OVERWRITE
:CALL: JSR R5,OVWPER
: SECTORS TO WRITE FORWARD
: SECTORS TO WRITE REVERSE

OVWPER: MOV R0,-(SP) ;SAVE R0, R1, R2, R3
MOV R1,-(SP)
MOV R2,-(SP)
MOV R3,-(SP)
CLR R0 ;R0 HAS COUNT IF R0<5.
MOV (R5)+,FORSK ;USE TOP SURFACE, IF R0>5.
MOV (R5)+,REVSK ;USE BOTTOM SURFACE, IF R0>1
;DONE.
1$: MOV #OVWTRK,R1 ;GET START OF LIST OF TRACKS
MOV (R1),R2 ;GET POINTER TO TRACK
CMP (R2),#-1 ;LEGIT TRACK?????
BEQ 3$ ;NO, EXIT

CLR CYL ;CLEAR CYLINDER/HEAD FOR SEEK
CLR SURF
CMP R0,#5 ;TOP/BOTTOM
BLT 2$ ;TOP, BRANCH
INC SURF ;BOTTOM SURFACE
2$: JSR R5,SKCYL ;SEEK TO CYLINDER
CLR CYL
BIS (R2),CYL ;SEEK TO PROPER CYLINDER
JSR R5,SKCYL ;SECTORS TO WRITE
MOV FORSK,R3 ;GO WRITE SECTORS
JSR R5,WRSEC
WORD 28.
MOV #FWD,DIRC ;SET FORWARD DIRECTION
JSR R5,VEROW ;VERIFY OVERWRITE
JSR R5,VEROD ;VERIFY OTHER DRIVES DATA
CLR CYL
CMP #1,T.DRIVE ;RLO1?
BNE 50$ ;NO
BIS #377,CYL ;SET TO GO TO MAX CYL
BR 51$
50$: BIS #777,CYL ;MAX CYL FOR RLO2
51$: JSR R5,SKCYL ;SEEK TO MAX CYLINDER ON DRIVE
CLR CYL
CLR SURF
BIS (R2),CYL ;DO ANOTHER SEEK
JSR R5,SKCYL
MOV REVSK,R3 ;SECTORS TO WRITE
JSR R5,WRSEC ;WRITE THEM
WORD 28.
MOV #REV,DIRC ;SET DIRECTION
JSR R5,VEROW ;VERIFY OVERWRITE
JSR R5,VEROD ;VERIFY OTHER DRIVES DATA
3$: TST (R1)+ ;INCREMENT TO NEXT TRACK
INC R0 ;ACCOUNT FOR IT
CMP R0,#10. ;DONE?
BNE 1$ ;NO, GO BACK
    
```


58					
59	025056	012603	MOV	(SP)+,R3	;RESTORE REG.
60	025060	012602	MOV	(SP)+,R2	
61	025062	012601	MOV	(SP)+,R1	
62	025064	012600	MOV	(SP)+,R0	
63	025066	000205	RTS	R5	;EXIT

```

1          :ROUTINE TO WRITE SECTORS
2          :USED IN OVERWRITE TEST;ADJACENT CYLINDER TEST
3          :CALL   JSR   R5,WRSEC
4          :      .WRD           :STARTING SECTOR
5          :R3 HAS BITMAP OF SECTORS TO WRITE
6          :R4 HAS DRIVE BUFFER POINTER
7
8 025070 010046      WRSEC:  MOV   R0,-(SP)           :SAVE R0
9 025072 010146      MOV   R1,-(SP)           :SAVE R1
10 025074 010246     MOV   R2,-(SP)           :SAVE R2
11 025076 012701 003232  MOV   #BUF,R1           :WRITE PATTERN INTO
12 025102 012702 000200  MOV   #128.,R2          :MEMORY THAT WE
13 025106 016421 000006  2$:  MOV   PAT(R4),(R1)+       :WILL WRITE ONTO
14 025112 005302      DEC   R2              :PACK FOR THIS
15 025114 001374      BNE   2$              :DRIVE
16 025116 012701 100000  MOV   #100000,R1        :MASK FOR BIT MAP
17 025122 012737 000007 003056  MOV   #7,TEM
18 025130 053702 003122      BIS   CYL,R2
19 025134 006302      120$: ASL   R2
20 025136 005337 003056      DEC   TEM
21 025142 001374      BNE   120$
22 025144 005737 003120      TST   SURF
23 025150 001402      BEQ   3$              :0, SKIP
24 025152 052702 000100  3$:  BIS   #HEAD,R2          :SET BOTTOM HEAD
25 025156 052502      4$:  BIS   (R5)+,R2        :START AT SECTOR 28.
26 025160 030103      4$:  BIT   R1,R3              :WRITE THIS SECTOR?
27 025162 001452      BEQ   5$              :NO
28
29 025164 005037 003110      CLR   HSFLG
30 025170 012737 177600 003206  MOV   #-128.,BMP       :LOAD WORD COUNT
31 025176 010237 003204      MOV   R2,BDA           :LOAD DISK ADDRESS
32 025202 010237 003066      MOV   R2,TEMP        :SAVE DISK ADDRESS
33 025206 042702 177700      BIC   #177700,R2
34 025212 020227 000047      CMP   R2,#39.
35 025216 003403      BLE   6$
36 025220 162737 000050 003204  SUB   #40.,BDA
37 025226 012737 003232 003202  6$:  MOV   #BUF,BBA       :LOAD BUS ADDRESS
38 025234 013702 003066      MOV   TEMP,R2        :RESTORE DISK ADDRESS
39 025240 004537 032404  11$:  JSR   R5,LDFUNC      :GO WRITE
40 025244 000012      WRITE
41 025246 005737 003074      TST   ERFLG          :ERROR IN WRITING
42 025252 001416      BEQ   5$              :NO,OKAY
43 025254 005737 003110      TST   HSFLG
44 025260 001007      BNE   10$
45 025262      ERRSOFT 100.,WRIT1,ERR2
46 025262 104457      TRAP  CSERSOFT
47 025264 000144      .WORD 100
48 025266 020136      .WORD WRIT1
49 025270 020276      .WORD ERR2
46 025272 005237 003110      INC   HSFLG
47 025276 000760      BR   11$
48 025300  10$:  ERRHRD 110.,WRIT1,ERR2
49 025300 104456      TRAP  CSERHRD
49 025302 000156      .WORD 110
49 025304 020136      .WORD WRIT1
49 025306 020276      .WORD ERR2

```

50	025310	005202	5\$:	INC	R2	:NEXT SECTOR
51	025312	000241		CLC		:CLEAR CARRY BIT
52	025314	006001		ROR	R1	:DONE?
53	025316	103320		BCC	4\$:NO GO BACK
54	025320	012602		MOV	(SP)+,R2	:RESTORE REGISTERS AND EXIT
55	025322	012601		MOV	(SP)+,R1	
56	025324	012600		MOV	(SP)+,R0	
57	025326	000205		RTS	R5	

1	025330	005037	003214		ADJCYL:	CLR	ADJTRK	:INSIDE/OUTSIDE TRACK FLAG
2	025334	005037	003114			CLR	HEAD01	:INIT TO TOP SURFACE
3	025340	012737	000001	003216		MOV	#1,ADJUUT	:START OF TRACK LIST
4	025346	012701	002236		21\$:	MOV	#OUT10,R1	:
5	025352	012537	003100		20\$:	MOV	(R5)+,ADJLOC	:PICK UP TRACK OFFSET
6	025356	001003				BNE	1\$:IS THERE ONE?
7	025360	005037	003104			CLR	ADJDIR	
8	025364	000205				RTS	R5	:NO EXIT
9	025366	012537	003220		1\$:	MOV	(R5)+,ADJLC2	:YES, GET REST OF INFO
10	025372	012537	003222			MOV	(R5)+,ADJLC3	
11	025376	012537	003224			MOV	(R5)+,ADJLC4	
12	025402	113700	003100		2\$:	MOVB	ADJLOC,R0	:GET OFFSET
13	025406	012737	000020	003230		MOV	#16.,STSEC	:STARTING SECTOR IS 16
14								
15	025414	010102				MOV	R1,R2	:GET START INTO R2
16								
17	025416	005300			3\$:	DEC	R0	:DOWN COUNT OFFSET
18	025420	001414				BEQ	4\$:FOUND IT?
19								
20	025422	005722				TST	(R2)+	:INDEX (R2)
21	025424	062737	000042	003230		ADD	#34.,STSEC	:NO, NEXT SECTOR
22	025432	022737	000050	003230		CMP	#40.,STSEC	
23	025440	003366				BGT	3\$	
24	025442	162737	000050	003230		SUB	#40.,STSEC	
25	025450	000762				BR	3\$:BACK FOR NEXT
26								
27	025452	021227	177777		4\$:	CMP	(R2),#-1	:LEGAL TRACK?
28	025456	001002				BNE	5\$:YES, CONTINUE
29								
30	025460	000137	026126			JMP	13\$:NO PICK UP NEXT SET
31								
32	025464	005037	003120		5\$:	CLR	SURF	:SET UP FOR OUTER TRACK
33	025470	005037	003122			CLR	CYL	
34								
35	025474	005737	003114			TST	HEAD01	:WHICH HEAD?
36	025500	001403				BEQ	6\$:TOP, SKIP
37								
38	025502	052737	000001	003120		BIS	#1,SURF	:LOWER HEAD, SET IT!
39								
40	025510	004537	026254		6\$:	JSR	R5,SKCYL	:SEEK TO OUTER TRACK
41								
42	025514	011237	003122			MOV	(R2),CYL	:GET DESIRED TRACK
43								
44	025520	004537	026254			JSR	R5,SKCYL	:SEEK TO IT
45	025524	012737	020116	003116		MOV	#FWD,DIRC	:SEEK DIRECTION
46	025532	113703	003101			MOVB	ADJLOC+1,R3	:GET SECTORS TO WRITE
47	025536	000303				SWAB	R3	:ALIGN IT
48	025540	042703	000377			BIC	#377,R3	:CLEAR OUT HIGH BYTE
49								
50	025544	022737	000047	003230		CMP	#39.,STSEC	:OVER FORTY?
51	025552	002003				BGE	7\$:NO, CONTINUE
52								
53	025554	162737	000050	003230		SUB	#40.,STSEC	:YES BACK IT UP
54	025562	013737	003230	025574	7\$:	MOV	STSEC,8\$:STARTING SECTOR
55								
56	025570	004537	025070			JSR	R5,WRSEC	:WRITE SECTORS
57	025574	000000			8\$:	.WORD	0	

58	025576	013737	025574	025610		MOV	8\$,108\$	
59	025604	004537	030112			JSR	R5,VAJWR	;VERIFY THIS WRITE
60	025610	000000			108\$:	.WORD	0	
61	025612	013737	025610	025624		MOV	108\$,208\$	
62	025620	004537	030356			JSR	R5,BSVWR	
63	025624	000000			208\$:	.WORD	0	
64	025626	013737	003230	003226		MOV	STSEC,STSEC1	;GET OTHER SECTORS TO WRITE
65	025634	062737	000010	003226		ADD	#8.,STSEC1	;8 SECTORS GONE BY
66	025642	022737	000047	003226		CMP	#39.,STSEC1	;GONE PAST 40?
67	025650	002003				BGE	9\$;NO, OKAY
68								
69	025652	162737	000050	003226		SUB	#40.,STSEC1	;YES BACK IT UP
70								
71	025660	013703	003220		9\$:	MOV	ADJLC2,R3	;GET SECTORS TO WRITE
72								
73	025664	013737	003226	025676		MOV	STSEC1,10\$;STARTING SECTORS
74								
75	025672	004537	025070			JSR	R5,WRSEC	;WRITE SECTORS
76	025676	000000			10\$:	.WORD	0	
77	025700	013737	025676	025712		MOV	10\$,110\$	
78	025706	004537	030112			JSR	R5,VAJWR	;VERIFY THIS WRITE
79	025712	000000			110\$:	.WORD	0	
80	025714	013737	025712	025726		MOV	110\$,210\$	
81	025722	004537	030356			JSR	R5,BSVWR	;VERIFY ADJ CYL + 1
82	025726	000000			210\$:	.WORD	0	
83	025730	022737	000001	003060		CMP	#1,T.DRIVE	
84	025736	001004				BNE	77\$	
85	025740	012737	000377	003122		MOV	#377,CYL	
86	025746	000403				BR	88\$	
87								
88	025750	012737	000777	003122	77\$:	MOV	#777,CYL	
89								
90	025756	004537	026254		88\$:	JSR	R5,SKCYL	
91								
92	025762	011237	003122			MOV	(R2),CYL	;SEEK BACK TO PROPER TRACK
93								
94	025766	004537	026254			JSR	R5,SKCYL	;SEEK TO PROPER CYLINDER
95	025772	012737	020126	003116		MOV	#REV,DIRC	;SEEK DIRECTION
96	026000	113703	003223			MOVB	ADJLC3+1,R3	;GET SECTORS TO WRITE
97								
98	026004	000303				SWAB	R3	;ALIGN IT
99	026006	042703	000377			BIC	#377,R3	;CLEAR OUT HIGH BYTE
100	026012	013737	003230	026024		MOV	STSEC,11\$	
101								
102	026020	004537	025070			JSR	R5,WRSEC	;WRITE PROPER SECTOR
103	026024	000000			11\$:	.WORD	0	
104								
105	026026	013737	026024	026040		MOV	11\$,111\$	
106	026034	004537	030112			JSR	R5,VAJWR	;VERIFY THIS WRITE
107	026040	000000			111\$:	.WORD	0	
108	026042	013737	026040	026054		MOV	111\$,211\$	
109	026050	004537	030356			JSR	R5,BSVWR	
110	026054	000000			211\$:	.WORD	0	
111	026056	013703	003224			MOV	ADJLC4,R3	;GET SECTORS
112	026062	013737	003226	026074		MOV	STSEC1,12\$;GET SECTORS TO WRITE
113								
114	026070	004537	025070			JSR	R5,WRSEC	;WRITE PROPER SECTORS

```

115 026074 000000      12$:  .WORD  0
116
117
118 026076 013737 026074 026110      MOV    12$,112$
119 026104 004537 030112      JSR    R5,VAJWR      ;VERIFY THIS WRITE
120 026110 000000      112$:  .WORD  0
121
122
123 026112 013737 026110 026124      MOV    112$,212$
124 026120 004537 030356      JSR    R5,BSVWR      ;VERIFY ADJ CYLINDERS + 1
125 026124 000000      212$:  .WORD  0
126
127
128 026126 005737 003114      13$:  TST    HEAD01      ;WHICH HEAD WERE WE DOING?
129 026132 001003      BNE    14$
130 026134 005237 003114      INC    HEAD01
131 026140 000402      BR     99$
132 026142 005037 003114      14$:  CLR    HEAD01      ;NEXT SET OF TRACKS
133 026146 062701 000012      99$:  ADD    #10,,R1      ;NEXT SET OF TRACKS
134 026152 020127 002400      CMP    R1,#INN51     ;END OF LIST
135 026156 002002      BGE    18$           ;END OF TRACK LIST
136 026160 000137 025402      JMP    2$           ;NO GO BACK
137
138      ;AT END OF TRACK LIST NEXT GROUP OF WRITES
139
140 026164 005737 003064      18$:  TST    FADJ          ;FIRST SET?
141 026170 001403      BEQ    15$          ;NO, CONTINUE
142 026172 005037 003064      CLR    FADJ        ;YES, CLEAR FIRST
143 026176 000421      BR     17$          ;EXIT
144 026200 005737 003214      15$:  TST    ADJTRK      ;DONE BOTH INSIDE OUTSIDE
145 026204 001004      BNE    16$          ;TRACKS, YES 16$
146 026206 005237 003214      INC    ADJTRK      ;NO, SET INSIDE FLAG
147 026212 000137 025346      JMP    21$          ;GO DO INSIDE TRACK
148 026216 005037 003214      16$:  CLR    ADJTRK      ;BACK TO OUTSIDE TRACK
149 026222 005237 003216      INC    ADJUUT      ;DONE WITH ANOTHER
150 026226 023737 003216 003130      CMP    ADJUUT,UUT  ;DONE TABLE FOR ALL UUT?
151 026234 001402      BLS    17$          ;YES, FOR EXIT
152 026236 000137 025346      JMP    21$          ;NO, GO BACK FOR NEXT
153 026242 005725      17$:  TST    (R5)+        ;BUMP EXIT TO END OF
154 026244 001376      BNE    17$          ;TABLE FOR PROPER RETURN
155 026246 005037 003104      CLR    ADJDIR      ;EXIT
156 026252 000205      RTS    R5
    
```

```

1          ;ROUTINE TO SEEK TO A DESIRED CYLINDER
2          ;CALL: JSR      R5,SKCYL
3          ;ROUTINE HAS DESIRED CYLINDER IN LOC "CYL"
4          ;
5          ;
6 026254 010146          SKCYL: MOV      R1,-(SP)          ;SAVE R1
7 026256 004537 032404 90$: JSR      R5,LDFUNC          ;GET PRESENT POSITION
8 026262 000010          RDHDR
9
10 026264 005737 003074          TST      ERFLG          ;ERROR FLAG SET
11 026270 001104          BNE      5$          ;YES, SKIP
12
13 026272 005001          CLR      R1
14 026274 012737 000007 003056 MOV      #7,TEM
15 026302 053701 003122          BIS      CYL,R1          ;GET THE SELECTED CYLINDER NUMBER
16
17 026306 006301          120$: ASL      R1
18 026310 005337 003056          DEC      TEM
19 026314 001374          BNE      120$
20 026316 042737 000177 003172 BIC      #177,E.MP
21 026324 163701 003172          SUB      E.MP,R1          ;CALCULATE DIFFERENCE WORD
22 026330 103002          BCC      1$          ;IF POSITIVE SET DIRECTION
23 026332 005401          NEG      R1          ;NEGATE
24 026334 000402          BR      2$          ;SKIP SETTING DIRECTION
25 026336 052701 000004          1$: BIS      #SIGN,R1          ;SET FOR FORWARD SEEK
26 026342 052701 000001          2$: BIS      #MK,R1          ;SET MARKER BIT
27 026346 005737 003120          TST      SURF
28 026352 001402          BEQ      3$          ;TOP
29 026354 052701 000020          BIS      #SKHS,R1          ;BOTTOM
30 026360 010137 003204          3$: MOV      R1,BDA          ;LOAD DIFFERENCE WORD
31 026364 004537 032404          JSR      R5,LDFUNC          ;EXECUTE SEEK
32 026370 000006          SEEK
33
34 026372 005737 003074          TST      ERFLG          ;ERROR?
35 026376 001041          BNE      5$          ;YES, SKIP
36
37 026400 004537 032404          JSR      R5,LDFUNC          ;VERIFY POSITION?
38 026404 000010          RDHDR
39 026406 005737 003074          TST      ERFLG
40 026412 001033          BNE      5$
41 026414 042737 000077 003172 BIC      #77,E.MP          ;VERIFY POSITION
42 026422 005001          CLR      R1
43 026424 012737 000007 003056 MOV      #7,TEM
44 026432 053701 003122          BIS      CYL,R1
45 026436 006301          220$: ASL      R1
46 026440 005337 003056          DEC      TEM
47 026444 001374          BNE      220$
48 026446 005737 003120          TST      SURF
49 026452 001402          BEQ      4$
50 026454 052701 000100          BIS      #HEAD,R1
51 026460 020137 003172          4$: CMP      R1,E.MP
52 026464 001414          BEQ      6$
53
54 026466          ERRDF 12.,SKER,ERR6 ;MIS SEEK ERROR
    026466 104455          TRAP  C$ERRDF
    026470 000014          .WORD 12
    026472 020077          .WORD SKER
    
```

```

55 026474 020704          .WORD  ERR6
56 026476 000137 026256  JMP    90$
57 026502          5$:  ERRDF  13.,FUNERR,ERR5 ;ERROR IN SEEK OPERATION
   026502 104455      TRAP   C$ERDF
   026504 000015      .WORD  13
   026506 020047      .WORD  FUNERR
   026510 020644      .WORD  ERR5
58 026512 000137 026256  JMP    90$
59 026516 012601      6$:  MOV   (SP)+,R1      ;CANT GET THERE
60 026520 000205      RTS   R5          ;EXIT
```



```

1      ;ROUTINE TO PERFORM REGISTER PRINTOUT DUMP
2      ;CALL: JSR      PC,REGDMP
3
4      REGDMP: PRINTB #FRM12,BCS,BBA,BDA,BMP
5
6      ;PROMPT - BEFORE CS: _ BA: _ DA: _ MP: _
7      MOV      BMP,-(SP)
8      MOV      BDA,-(SP)
9      MOV      BBA,-(SP)
10     MOV      BCS,-(SP)
11     MOV      #FRM12,-(SP)
12     MOV      #5,-(SP)
13     MOV      SP,R0
14     TRAP     C$PNTB
15     ADD      #14,SP
16
17     ;PROMPT - AFTER CS: _ BA: _ DA: _ MP: _
18     PRINTB   #FRM13,E.CS,E.BA,E.DA,E.MP
19     MOV      E.MP,-(SP)
20     MOV      E.DA,-(SP)
21     MOV      E.BA,-(SP)
22     MOV      E.CS,-(SP)
23     MOV      #FRM13,-(SP)
24     MOV      #5,-(SP)
25     MOV      SP,R0
26     TRAP     C$PNTB
27     ADD      #14,SP
28     BIT      #BIT14,E.CS
29     BEQ      1$
30     MOV      CSR(R4),R3
31     MOV      #13,DA(R3)
32     MOV      #4,BCS
33     BIS      DSB(R4),BCS
34     MOV      BCS,CS(R3)
35     BIT      #200,CS(R3)
36     BEQ      2$
37     MOV      MP(R3),DRSTAT
38     PRINTB   #FRM14,DRSTAT
39     MOV      DRSTAT,-(SP)
40     MOV      #FRM14,-(SP)
41     MOV      #2,-(SP)
42     MOV      SP,R0
43     TRAP     C$PNTB
44     ADD      #6,SP
45     RTS      PC
46
47     ;PROMPT - DRIVE STATUS
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

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      ;              .WORD 0      ;RETRIEVE SECTOR INFO.
8      ;CALL 2:      JSR R5,RSADJS
9      ;              .WORD 1      ;STORE SECTOR INFO.
10     RSADJS: MOV    R1,-(SP)
11     MOV    R2,-(SP)
12     MOV    R3,-(SP)
13     BIC    #177700,R0      ;SAVE SECTOR BITS
14     MOV    (R5)+,ADJFLG    ;SAVE RETRIEVE/STORE FLAG
15     MOV    #1,R1          ;START WITH TRACK (N-2)
16     MOV    #SECBUF,R2     ;START OF 24X5 BUFFER
17     MOV    #16.,R3       ;SECTOR 16 START FOR (N-2) TRACK
18     1$:  CMPB   ADJLOC,R1  ;CHECK TRACK INDEX
19     BEQ    2$
20     INC    R1            ;INDEX TRACK REFERENCE
21     ADD    #48.,R2       ;UPDATE BUFFER TO NEXT TRACK REF.
22     ADD    #34.,R3       ;UPDATE SECTOR START FOR NEXT TRACK
23     CMP    R3,#40.
24     BLT    1$
25     SUB    #40.,R3
26     BR    1$
27     2$:  MOV    #24.,R1    ;SET COUNTER FOR 24 SECTORS
28     3$:  CMP    R0,R3     ;COMPARE SECTOR TO SECTOR TABLE
29     BEQ    5$           ;YES,STORE OR RETRIEVE SECTOR INFO.
30     TST   (R2)+        ;INDEX SECLST BUFFER IN WORD FORMAT
31     INC    R3          ;INDEX SECTOR COUNT
32     CMP    R3,#39.     ;COMPARE SECTOR COUNT FOR <40
33     BLE    4$
34     SUB    #40.,R3     ;KEEP SECTOR COUNT<40
35     4$:  DEC    R1        ;PASSED 24 SECTORS?
36     BNE    3$         ;COMPARE NEXT SECTOR
37     CLR    R0         ;SETUP R0 FOR EXIT
38     BR    7$         ;EXIT ROUTINE,SECTOR NOT FOUND
39     5$:  TST   ADJFLG    ;FLAG=0 FOR RETRIEVE
40     BEQ    6$
41     MOV    R4,(R2)     ;STORE DRIVE INFO. INTO BUFFER
42     6$:  MOV    (R2),R0  ;SAVE DRIVE INFO. INTO R0 FOR EXIT
43     7$:  MOV    (SP)+,R3
44     MOV    (SP)+,R2
45     MOV    (SP)+,R1
46     RTS    R5         ;EXIT
    
```

```

1      ;ROUTINE TO SET DRIVE IN SECTOR LIST
2      ;CALL: JSR   R5,SETLST      ;R0 HAS SECTOR
3      ;DRIVE GOTTEN FROM R4
4
5 027100 010146      SETLST: MOV   R1,-(SP)      ;SAVE R1
6
7 027102 162700 000034      SUB   #28.,R0      ;START LIST AT 0
8 027106 100002      BPL   3$
9 027110 062700 000050      ADD   #40.,R0
10 027114 012701 002402     3$:  MOV   #SECLST,R1      ;BEGINNING OF SECTOR LIST
11 027120 005700      1$:  TST   R0      ;FOUND SECTOR?
12 027122 001403      BEQ   2$      ;BRANCH IF YES
13 027124 005300      DEC   R0      ;DECREMENT SECTOR
14 027126 005721      TST   (R1)+      ;NEXT ENTRY IN LIST
15 027130 000773      BR    1$      ;GO BACK
16 027132 010411     2$:  MOV   R4,(R1)      ;STORE DRIVE BITS IN LIST
17 027134 012601      MOV   (SP)+,R1      ;RESTORE R1
18 027136 000205      RTS   R5
19
20      ;ROUTINE TO LOCATE DRIVE THAT WROTE SECTOR LAST
21      ;CALL: JSR   R5,FNDDRV      ;R0-CONTAINS SECTOR
22      ;ON EXIT R0-DRIVE
23
24 027140 010146      FNDDRV: MOV  R1,-(SP)      ;SAVE R1
25 027142 162700 000034      SUB  #28.,R0      ;START LIST AT 0
26 027146 100002      BPL  3$
27 027150 062700 000050      ADD  #40.,R0
28 027154 012701 002402     3$:  MOV  #SECLST,R1      ;START OF LIST
29 027160 005700      1$:  TST  R0      ;FOUND SECTOR?
30 027162 001403      BEQ  2$      ;YES, GET DRIVE #, EXIT
31 027164 005300      DEC  R0      ;NO, DOWN COUNT SECTOR
32 027166 005721      TST  (R1)+      ;NEXT ENTRY IN LIST
33 027170 000773      BR   1$      ;GO BACK
34 027172 011100     2$:  MOV  (R1),R0      ;GET DRIVE BUFFER POINTER
35 027174 012601      MOV  (SP)+,R1      ;RESTORE R1
36 027176 000205      RTS  R5      ;EXIT
    
```

```

1
2
3
4
5
6
7
8
9 027200 010046
10 027202 010146
11 027204 010246
12 027206 012737 000034 003132
13 027214 012701 100000
14 027220 016437 000006 003136
15
16 027226 012737 177600 003206 1$: MOV #-128.,BMP ;SET UP READ-ONE SECTOR
17 027234 012737 003232 003202 MOV #BUF,BBA ;BUS ADDRESS
18 027242 042737 000077 003204 2$: BIC #77,BDA ;CLEAR OUT SECTOR BITS
19 027250 053737 003132 003204 BIS SECT,BDA ;SET SECTOR
20 027256 030103 BIT R1,R3 ;DO WE READ THIS ONE?
21 027260 001521 BEQ 5$ ;NO, BRANCH
22 027262 004537 032404 JSR R5,LDFUNC ;READ
23 027266 000014 READ
24
25 027270 005737 003164 TST E.CS ;ERROR
26 027274 100107 BPL 4$ ;NO CONTINUE
27
28 027276 005737 003062 TST F0WR ;INITIAL WRITE
29 027302 001412 BEQ 21$ ;NO
30 027304 012737 017373 003072 MOV #INITWR,REASON ;SETUP INITIAL WRITE OF SECTOR
31 027312 016437 000000 003070 MOV CSR(R4),LSTCLR
32 027320 016437 000005 003134 MOV DSB+1(R4),LSTDRV
33 027326 000415 BR 22$
34 027330 012737 017642 003072 21$: MOV #OVMS,REASON ;SET MESSAGE FOR OVERWRITE
35 027336 013700 003132 MOV SECT,R0 ;FIND DRIVE THAT LAST WROTE
36 027342 004537 027140 JSR R5,FNDDRV ;SECTOR
37 027346 016037 000000 003070 MOV CSR(R0),LSTCLR ;GET IT'S CSR
38 027354 116037 000005 003134 MOV DSB+1(R0),LSTDRV ;GET THE DRIVE
39 027362 104455 22$: ERDF ;PRINT ERROR
    027362 000015 TRAP C$ERDF
    027366 020003 .WORD 13
    027370 020464 .WORD OVWER
    027372 005037 003142 CLR WCOUNT ;CLEAR BAD WORD COUNT W/IN SECTOR
41 027376 005037 003144 CLR SECWRD ;CLEAR WORD IN SECTOR
42 027402 012702 003232 MOV #BUF,R2 ;GET BUFFER START
43 027406 023712 003136 3$: CMP GDATA,(R2) ;IS DATA CORRECT?
44 027412 001417 BEQ 31$ ;YES CHECK NEXT
45 027414 005237 003142 INC WCOUNT ;NO ACCOUNT FOR IT
46 027420 PRINTF #FRM8,SECWRD,GDATA,(R2)
    027420 011246 MOV (R2),-(SP)
    027422 013746 003136 MOV GDATA,-(SP)
    027426 013746 003144 MOV SECWRD,-(SP)
    027432 012746 021437 MOV #FRM8,-(SP)
    027436 012746 000004 MOV #4,-(SP)
    027442 010600 MOV SP,R0
    027444 104417 TRAP C$PNTF
    
```

:ROUTINE TO VERIFY THAT THE OVERWRITE D.D 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 CSR(R4),LSTCLR
 MOV DSB+1(R4),LSTDRV

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

TRAP C\$ERDF
 .WORD 13
 .WORD OVWER
 .WORD ERR4
 CLR WCOUNT ;CLEAR BAD WORD COUNT W/IN SECTOR
 CLR SECWRD ;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 #FRM8,SECWRD,GDATA,(R2)
 MOV (R2),-(SP)
 MOV GDATA,-(SP)
 MOV SECWRD,-(SP)
 MOV #FRM8,-(SP)
 MOV #4,-(SP)
 MOV SP,R0
 TRAP C\$PNTF

```

027446 062706 000012      ADD      #12,SP
47
48 027452 005722          31$:  TST      (R2)+      ;NEXT
49 027454 005237 003144  INC      SECWRD      ;NEXT
50 027460 023727 003144 000200  CMP      SECWRD,#128. ;DONE WITH SECTOR?
51 027466 001347          BNE      3$          ;NO GO BACK
52
53 027470          PRINTF  #FRM9,WCOUNT ;PRINT SUMMARY
    027470 013746 003142  MOV      WCOUNT,-(SP)
    027474 012746 021503  MOV      #FRM9,-(SP)
    027500 012746 000002  MOV      #2,-(SP)
    027504 010600          MOV      SP,R0
    027506 104417          TRAP    C$PNTF
    027510 062706 000006  ADD      #6,SP
54
55 027514 013700 003132  4$:  MOV      SECT,R0      ;SET SECTOR IN LIST TO THE
56 027520 004537 027100  JSR      R5,SETLST     ;CREDIT OF THIS DRIVE
57
58 027524 005237 003132  5$:  INC      SECT          ;NEXT SECTOR
59 027530 023727 003132 000050  CMP      SECT,#40.
60 027536 001003          BNE      6$
61 027540 162737 000050 003132  SUB      #40.,SECT
62 027546 000241          6$:  CLC
63 027550 006001          ROR      R1
64 027552 103225          BCC     1$
65
66 027554 012602          MOV      (SP)+,R2      ;RESTORE R2-R0, EXIT
67 027556 012601          MOV      (SP)+,R1
68 027560 012600          MOV      (SP)+,R0
69 027562 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 027564 010046          VEROD: MOV      R0,-(SP)      ;SAVE R0-R2
9 027566 010146          MOV      R1,-(SP)
10 027570 010246          MOV      R2,-(SP)
11 027572 012701 100000   MOV      #100000,R1      ;BIT MASK FOR SECTORS
12 027576 012737 000034 003132   MOV      #28.,SECT      ;START WITH SECTOR 28
13 027604 005737 003062   TST      F0WR          ;CHECK FOR FIRST OVERWRITE
14 027610 001134          BNE      6$
15
16 027612 012737 177600 003206 1$:  MOV      #-128.,BMP     ;SET UP READ (ONE SECTOR)
17 027620 012737 003232 003202   MOV      #BUF,BBA      ;BUS ADDRESS
18 027626 042737 000077 003204 2$:  BIC      #77,BDA       ;CLEAR SECTOR BITS
19 027634 053737 003132 003204   BIS      SECT,BDA      ;SET IN SECTOR BITS
20 027642 030103          BIT      R1,R3         ;CHECK THIS SECTOR?
21 027644 001103          BNE      5$           ;NO BRANCH
22
23 027646 013700 003132   MOV      SECT,R0       ;FIND DRIVE THAT WROTE
24 027652 004537 027140   JSR      R5,FNDDRV     ;SECTOR LAST
25 027656 016037 000000 003070   MOV      CSR(R0),LSTCLR ;GET CSR OF DRIVE
26 027664 116037 000005 003134   MOV      DSB+1(R0),LSTDRV ;GET DRIVE
27 027672 016037 000006 003136   MOV      PAT(R0),GDATA ;GET PATTERN
28
29 027700 004537 032404   JSR      R5,LDFUNC     ;READ
30 027704 000014          READ
31
32 027706 005737 003164   TST      E.CS         ;ERROR?
33 027712 100060          BPL      5$           ;NO, NEXT SECTOR
34 027714 012737 017675 003072   MOV      #RECMS,REASON ;SET READ RECOVERY MESSAGE
35 027722 005737 003164   ERRDF   14.,RECER,ERR4 ;REPORT ERROR
    027722 104455          TRAP    C$ERDF
    027724 000016          .WORD  14
    027726 020023          .WORD  RECER
    027730 020464          .WORD  ERR4
36
37 027732 005037 003142   CLR      WCOUNT      ;CLEAR BAD WORD COUNT
38 027736 005037 003144   CLR      SECWRD       ;CLEAR WORD W/I SECTOR
39 027742 012702 003232   MOV      #BUF,R2      ;START OF BUFFER
40 027746 023712 003136 3$:  CMP      GDATA,(R2)    ;DATA COMPARE
41 027752 001417          BEQ      4$           ;YES, CHECK NEXT
42
43 027754 005237 003142   INC      WCOUNT      ;ACCOUNT FOR ERROR
44 027760          PRINTF #FRMB,SECWRD,GDATA,(R2) ;PRINT ERROR
    027760 011246          MOV      (R2),-(SP)
    027762 013746 003136   MOV      GDATA,-(SP)
    027766 013746 003144   MOV      SECWRD,-(SP)
    027772 012746 02437   MOV      #FRMB,-(SP)
    027776 012746 000004   MOV      #4,-(SP)
    030002 010600          MOV      SP,R0
    030004 104417          TRAP    C$PNTF
    030006 062706 000012   ADD      #12,SP
45
    
```

46	030012	005722		4\$:	TST	(R2)+	:NEXT
47	030014	005237	003144		INC	SECWRD	:NEXT WORD IN SECTOR
48	030020	023727	003144	000200	CMP	SECWRD,#128.	:DONE?
49	030026	001347			BNE	3\$:NO
50	030030				PRINTF	#FRM9,WCOUNT	:PRINT SUMMARY
	030030	013746	003142		MOV	WCOUNT,-(SP)	
	030034	012746	021503		MOV	#FRM9,-(SP)	
	030040	012746	000002		MOV	#2,-(SP)	
	030044	010600			MOV	SP,R0	
	030046	104417			TRAP	C\$PNTF	
	030050	062706	000006		ADD	#6,SP	
51							
52	030054	005237	003132	5\$:	INC	SECT	:NEXT SECTOR
53	030060	023727	003132	000050	CMP	SECT,#40.	
54	030066	001002			BNE	7\$	
55	030070	005037	003132		CLR	SECT	
56	030074	000241		7\$:	CLC		
57	030076	006001			ROR	R1	:NEXT BIT MAP
58	030100	103244			BCC	1\$	
59							
60	030102	012602		6\$:	MOV	(SP)+,R2	:RESTORE R2-R0, EXIT
61	030104	012601			MOV	(SP)+,R1	
62	030106	012600			MOV	(SP)+,R0	
63	030110	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 030112 010046          VAJWR:  MOV     R0,-(SP)          ;SAVE REGISTERS
7 030114 010146          MOV     R1,-(SP)
8 030116 010246          MOV     R2,-(SP)
9 030120 012701 100000   MOV     #100000,R1      ;BIT MASK FOR CYLINDER
10 030124 012502          MOV     (R5)+,R2      ;STARTING SECTOR
11 030126 005000          CLR     R0
12 030130 053700 003122   BIS     CYL,R0
13 030134 012737 0000C7 003056   MOV     #7,TEM
14
15 030142 006300          2$:    ASL     R0
16 030144 005337 003056   DEC     TEM
17 030150 001374          BNE    2$
18 030152 005737 003120   TST    SURF
19 030156 001402          BEQ    3$
20 030160 052700 000100   BIS    #HEAD,R0
21 030164 050200          3$:    BIS    R2,R0
22 030166 030103          4$:    BIT    R1,R3
23 030170 001462          BEQ    5$
24 030172 012737 177600 003206   MOV     #-128.,BMP
25 030200 010037 003204   MOV     R0,BDA
26 030204 010037 003066   MOV     R0,TEMP
27 030210 042700 177700   BIC    #177700,R0
28 030214 020027 000047   CMP    R0,#39.
29 030220 003406          BLE    6$
30 030222 162737 000050 003204   SUB    #40.,BDA
31 030230 162737 000050 003066   SUB    #40.,TEMP
32 030236 012737 003232 003202 6$:    MOV     #BUF,BBA
33 030244 005037 003110   CLR    HSFLG
34 030250 013700 003066   MOV    TEMP,R0
35 030254 004537 032404 10$:    JSR    R5,LDFUNC      ;READ FUNCTION
36 030260 000014          READ
37 030262 005737 003074   TST    ERFLG
38 030266 001416          BEQ    7$
39 030270 005737 003110   TST    HSFLG
40 030274 001007          BNE    11$
41 030276          ERRSOFT 120.,READ1,ERR2
42 030276 104457          TRAP  CSERSOFT
43 030300 000170          .WORD 120
44 030302 020163          .WORD READ1
45 030304 020276          .WORD ERR2
46 030306 005237 003110   INC    HSFLG
47 030312 000760          BR    10$
48 030314          11$:   ERRHRD 130.,READ1,ERR2
49 030314 104456          TRAP  CSERHRD
50 030316 000202          .WORD 130
51 030320 020163          .WORD READ1
52 030322 020276          .WORD ERR2
53 030324 010046          7$:    MOV     R0,-(SP)
54 030326 004537 026732   JSR    R5,RSADJS      ;STORE ADJ. CYL. SECTOR INFO.
55 030332 000001          .WORD 1
56 030334 012600          MOV    (SP)+,R0      ;RESTORE R0
57 030336 005200          5$:    INC    R0
    
```


50	030340	000241	CLC		
51	030342	006001	ROR	R1	
52	030344	103310	BCC	4\$	
53	030346	012602	MOV	(SP)+,R2	;RESTORE REGISTERS AND EXIT
54	030350	012601	MOV	(SP)+,R1	
55	030352	012600	MOV	(SP)+,R0	
56	030354	000205	RTS	R5	
57					

```

1      ;ROUTINE TO VERIFY THAT WRITE DID NOT DISTURB ADJACENT TRACKS
2      ;WRITTEN BY OTHER DRIVES.
3      ;CALL JSR R5,BSVWR
4      ;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      BSVWR:  MOV     R0,-(SP)      ;SAVE REGISTERS
10     MOV     R1,-(SP)
11     MOV     R2,-(SP)
12     MOV     CYL,-(SP)
13     MOV     SURF,-(SP)
14     MOV     (R5)+,-(SP)      ;GET STARTING SECTOR
15     CMPB   ADJLOC,#3        ;ON MIDDLE TRACK???
16     BEQ    BSEXIT          ;YES, THEN NO CHECK
17     SUB    #34.,(SP)      ;SETUP SECTOR START FOR OUTSIDE
18     BPL    1$             ;IF POSITIVE OKAY ELSE FIX
19     ADD    #40.,(SP)      ;FIX IT
20     CMPB   ADJLOC,#1        ;ON OUTER LIMIT???
21     BEQ    INAWR          ;YES,SKIP CHECK
22     DECB   ADJLOC          ;OUTER ADJ TRACK
23     DEC    CYL
24     JSR    R5,CHECK        ;GO CHECK ADJ SECTORS
25     INC    CYL            ;FIX BACK
26     INCB   ADJLOC
27     INAWR:  ADD    #68.,(SP)   ;INNER SECTOR START
28     CMP    (SP),#40.      ;WITHIN LIMITS???
29     BLT    1$             ;YES, OKAY
30     SUB    #40.,(SP)      ;FIX SECTOR
31     CMP    (SP),#40.
32     BLT    1$
33     SUB    #40.,(SP)
34     CMPB   ADJLOC,#5        ;INNER LIMIT??
35     BEQ    BSEXIT          ;YES,SKIP CHECK
36     INCB   ADJLOC          ;FIX FOR INNER
37     INC    CYL
38     JSR    R5,CHECK        ;GO CHECK ADJ SECTORS
39     DECB   ADJLOC          ;FIX BACK
40     DEC    CYL
41     BSEXIT: TST    (SP)+      ;THROW OFF SECTOR
42     MOV    (SP)+,SURF
43     MOV    (SP)+,CYL
44     NCHECK: MOV    (SP)+,R2
45     MOV    (SP)+,R1
46     MOV    (SP)+,R0
47     JSR    R5,SKCYL        ;SEEK BACK
48     RTS     R5            ;RETURN
49

```

```

1          :ROUTINE TO VERIFY AN ADJACENT SECTOR
2          :CALLED FROM BSVWR
3          :CALL JSR R5,CHECK
4          :
5
6 030566 012701 100000 CHECK: MOV #100000,R1 ;SECTOR MASK
7 030572 004537 026254 JSR R5,SKCYL ;GET TO DESIRED CYLINDER
8 030576 005002 CLR R2 ;CREATE ADDRESS
9 030600 053702 003122 BIS CYL,R2
10 030604 012737 000007 003056 MOV #7,TEM
11 030612 006302 2$: ASL R2
12 030614 005337 003056 DEC TEM
13 030620 001374 BNE 2$
14 030622 005737 003120 TST SURF
15 030626 001402 BEQ 3$ ;NO
16 030630 052702 000100 BIS #HEAD,R2
17 030634 056602 000002 3$: BIS 2(SP),R2 ;SET IN SECTOR
18 030640 030103 4$: BIT R1,R3 ;THIS SECTOR IN LIST???
19 030642 001452 BEQ 5$ ;NO, NEXT
20 030644 010200 MOV R2,R0 ;COPY SECTOR
21 030646 042700 177700 BIC #177700,R0 ;ONLY SECTOR LEFT
22 030652 020027 000050 CMP R0,#40. ;SECTOR OKAY???
23 030656 002404 BLT 6$ ;YES
24 030660 162700 000050 SUB #40.,R0
25 030664 162702 000050 SUB #40.,R2 ;FIX SECTOR
26 030670 004537 026732 6$: JSR R5,RSADJS ;FIND IF SECTOR PREVIOUSLY WRITTEN
27 030674 000000 .WORD 0
28 030676 005700 TST R0 ;WAS IT??
29 030700 001433 BEQ 5$ ;NO
30 030702 010237 003204 MOV R2,BDA ;LOAD DISK ADDRESS
31 030706 012737 177600 003206 MOV #-128.,BMP ;LOAD WC
32 030714 004537 032404 JSR R5,LDFUNC ;LOAD
33 030720 000014 READ
34 030722 005737 003074 TST ERFLG ;WAS READ GOOD
35 030726 001420 BEQ 5$
36 030730 010346 MOV R3,-(SP)
37 030732 010237 003132 MOV R2,SECT
38 030736 010003 MOV R0,R3
39 030740 042737 177700 003132 BIC #177700,SECT
40 030746 ERRHRD 140.,ADJTXT,ERR3
    030746 104456 TRAP C$ERHRD
    030750 000214 .WORD 140
    030752 020210 .WORD ADJTXT
    030754 020336 .WORD ERR3
41 030756 012603 MOV (SP)+,R3
42 030760 ERFHRD 110.,READ1,ERR2
    030760 104456 TRAP C$ERHRD
    030762 000156 .WORD 110
    030764 020163 .WORD READ1
    030766 020276 .WORD ER2
43 030770 005202 5$: INC R2 ;NEXT SECTOR
44 030772 000241 CLC
45 030774 006001 ROR R1 ;SHIFT MASK
46 030776 103320 BCC 4$
47 031000 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 031002 010146      MERGE:  MOV      R1,-(SP)      ;SAVE R1, R2, R3
13 031004 010246      MOV      R2,-(SP)
14 031006 010346      MOV      R3,-(SP)
15 031010 012737 003632 003202  MOV      #BUF+400,BBA  ;BUFFER START FOR FIELD BAD
16 031016 022737 000001 003060  CMP      #1,T.DRIVE
17 031024 001004      BNE      55$
18 031026 012737 077724 003204  MOV      #77724,BDA
19 031034 000403      BR       66$
20 031036 012737 177724 003204 55$:  MOV      #177724,BDA
21
22 031044 012737 177400 003206 66$:  MOV      #-256.,BMP
23 031052 004537 032404 97$:  JSR      R5,LDFUNC  ;LOAD READ FUNCTION
24 031056 000014      READ
25 031060 005737 003074      TST      ERFLG      ;TEST ERROR FLAG
26 031064 001431      BEQ      98$      ;YES;MERGE BAD SECTOR FILES
27 031066 062737 000004 003204  ADD      #4,BDA      ;TRY NEXT FIELD BAD SECTOR FILE
28 031074 022737 000001 003060  CMP      #1,T.DRIVE
29 031102 001004      BNE      400$
30 031104 022737 077750 003204  CMP      #77750,BDA
31 031112 001357      BNE      97$
32
33 031114 022737 177750 003204 400$:  CMP      #177750,BDA
34 031122 001353      BNE      97$      ;NO,DO NEXT FIELD BAD SECTOR
35 031124      PRINTF  #FRM15
36 031124 012746 022037  MOV      #FRM15,-(SP)
37 031130 012746 000001  MOV      #1,-(SP)
38 031134 010600      MOV      SP,R0
39 031136 104417      TRAP    C$PNTF
40 031140 062706 000004      ADD      #4,SP
41 031144      999$:  BREAK
42 031144 104422      TRAP    C$BRK
43 031146 000776      BR       999$
44 031150 012701 003242 98$:  MOV      #BUF+10,R1  ;GET PAST ID ETC.
45 031154 012702 000176  MOV      #126.,R2  ;MAX = 126
46 031160 005721 1$:  TST      (R1)+      ;SECTOR OR END
47 031162 100404      BMI     2$      ;END, GO GET FIELD
48 031164 005721      TST      (R1)+      ;REST OF SECTOR
49 031166 005302      DEC     R2      ;MAX REACHED
50 031170 001373      BNE     1$      ;NO, KEEP GOING
51 031172 000401      BR      3$      ;YES, SKIP BACK UP
52 031174 005741 2$:  TST      -(R1)      ;BACK UP PAST TERMINATOR
53 031176 012703 000176 3$:  MOV      #126.,R3  ;SET 126 MAX
54 031202 012702 003642  MOV      #BUF+410,R2 ;GET FIELD SECTORS
55 031206 012221 4$:  MOV      (R2)+,(R1)+ ;MERGE AT END OF FACTORY
56 031210 100403      BMI     5$      ;DONE?
57 031212 012221      MOV      (R2)+,(R1)+ ;NO, MERGE REST OF SECTOR
    
```

```
52 031214 005303          DEC      R3          :DONE
53 031216 001373          BNE     4$          :NO, GO BACK
54 031220 012603          5$: MOV    (SP)+,R3    :RESTORE R3, R2, R1
55 031222 012602          MOV    (SP)+,R2
56 031224 012601          MOV    (SP)+,R1
57 031226 000205          RTS     R5          :EXIT
```

1	031230	012537	003146		FNDTRK:	MOV	(R5)+,OFFSET	:GET INCREMENT/DECREMENT
2	031234	012537	003156			MOV	(R5)+,SURFACE	:GET HEAD (SURFACE)
3	031240	022737	000001	003060		CMP	#1,T.DRIVE	
4	031246	001001				BNE	80\$	
5	031250	000401				BR	90\$	
6	031252	022525			80\$:	CMP	(R5)+,(R5)+	
7	031254	012537	003152		90\$:	MOV	(R5)+,FRTRK	
8	031260	012537	003150			MOV	(R5)+,LSTTRK	
9	031264	005037	003160			CLR	TRKFND	:CLEAR OUT FLAG FOUND
10	031270	005037	003162			CLR	TRKCNT	:CLEAR OUT TRACK COUNT
11	031274	013737	003152	003154		MOV	FRTRK,PRSTRK	:GET FIRST TRACK
12	031302				1\$:			
13	031302	004537	031402			JSR	R5,FNDBSC	:IS TRACK IN BAD SECTOR FILE
14	031306	005737	002234			TST	HDRFND	:WAS IT?
15	031312	001003				BNE	2\$:YES, CLEAR TRKCNT
16	031314	005237	003162			INC	TRKCNT	:NO, INDICATE GOOD TRACK
17	031320	000402				BR	3\$:CONTINUE
18	031322	005037	003162		2\$:	CLR	TRKCNT	:START COUNT OVER
19	031326	023727	003162	000005	3\$:	CMP	TRKCNT,#5	:FIND 5 TRACKS YET?
20	031334	001011				BNE	4\$:NO, CONTINUE
21	031336	005237	003160			INC	TRKFND	:YES, EXIT WITH GOOD FLAG
22	031342	022737	000001	003060		CMP	#1,T.DRIVE	
23	031350	001002				BNE	81\$	
24	031352	062705	000004			ADD	#4,R5	
25								
26	031356	000205			81\$:	RTS	R5	
27	031360	023737	003154	003150	4\$:	CMP	PRSTRK,LSTTRK	:ARE WE DONE?
28	031366	001001				BNE	5\$:NO, KEEP LOOKING
29	031370	000205				RTS	R5	:EXIT WITH NOT FOUND
30	031372	063737	003146	003154	5\$:	ADD	OFFSET,PRSTRK	:NEXT TRACK
31	031400	000740				BR	1\$	
32								

```

1          ;ROUTINE TO FIND BAD TRACK IN FILE
2          ;CALL   JSR   R5,FNDBSC
3
4 031402 005037 002234  FNDBSC: CLR   HDRFND      ;INITIALIZE FLAG
5 031406 010146          MOV   R1,-(SP)    ;SAVE R1, R2
6 031410 010246          MOV   R2,-(SP)
7 031412 012701 003242  MOV   #BUF+10,R1    ;SETUP FOR BEGINNING OF FILE
8 031416 005711          1$:  TST   (R1)          ;END?
9 031420 010421          BMI   2$          ;IF MINUS AT END, EXIT
10 031422 023721 003154  CMP   PRSTRK,(R1)+  ;CYLINDER CORRECT?
11 031426 001011          BNE   3$          ;NO, NEXT
12 031430 0105724        TSTB  (R4)+        ;UPPER HALF OF WORD
13 031432 0123711 003156  CMPB  SURFACE,(R1) ;CORRECT SURFACT
14 031436 001402          BEQ   4$          ;
15 031440 0105744        TSTB  -(R4)
16 031442 000403          BR    3$          ;
17 031444 005237 002234  4$:  INC   HDRFND      ;SET FOUND
18 031450 000405          BR    2$
19
20 031452 005721          3$:  TST   (R1)+        ;NEXT WORD
21 031454 005202          INC   R2          ;ACCOUNT FOR IT
22 031456 020227 000374  CMP   R2,#252.    ;DONE?
23 031462 001355          BNE   1$          ;NO, KEEP CHECKING
24 031464 012601          2$:  MOV   (SP)+,R1    ;RESTORE R2, R1, EXIT
25 031466 012602          MOV   (SP)+,R2
26 031470 000205          RTS   R5
27
28 031472 013701 003154  FIXCYL: MOV  PRSTRK,R1 ;GET TRACK WHICH IS GOOD
29 031476 005737 003146  TST   OFFSET      ;WHICH WAY WERE WE LOOKING
30 031502 0100402        BMI   1$          ;IN WORD, BRANCH
31 031504 0162701 000004  SUB   #4,R1        ;BACK IT UP BY FOUR
32 031510 012702 000005  1$:  MOV   #5,R2        ;GOING STORE AWAY 5 TRACKS
33 031514 010120          2$:  MOV   R1,(R0)+    ;STORE THEM 1 WD/PER
34 031516 005201          INC   R1
35 031520 005302          DEC   R2
36 031522 001374          BNE   2$
37 031524 000205          RTS   R5
    
```

```

1          ;ROUTINE TO GET SERIAL NUMBER
2
3          ;CALL JSR R5,SERNUM
4
5 031526 012737 000013 003204 SERNUM: MOV #13,BDA
6 031534 004537 032404 JSR R5,LDFUNC ;GET STATUS
7 031540 000004 GSTAT
8 031542 004537 032404 JSR R5,LDFUNC ;READ HEADER
9 031546 000010 RDHDR
10 031550 013700 003172 MOV E.MP,R0 ;GET THE HEADER
11 031554 042700 000077 1$: BIC #77,R0 ;CLEAR SECTOR BITS
12 031560 022737 000001 003060 CMP #1,T.DRIVE
13 031566 001003 BNE 23$
14 031570 020027 077700 CMP R0,#77700
15 031574 001446 BEQ 2$
16 031576 020027 177700 23$: CMP R0,#177700
17 031602 001443 BEQ 2$
18 031604 042700 000100 BIC #100,R0 ;CLEAR HEAD
19 031610 022737 000001 003060 CMP #1,T.DRIVE
20 031616 001003 BNE 32$
21 031620 012701 077600 MOV #77600,R1
22 031624 000402 BR 33$
23 031626 012701 177600 32$: MOV #177600,R1
24
25 031632 160001 33$: SUB R0,R1
26 031634 010137 003204 MOV R1,BDA ;SET UP DIF WORD
27 031640 052737 000025 003204 BIS #25,BDA ;SEEK IN, HEAD 1
28 031646 004537 032404 JSR R5,LDFUNC ;SEEK
29 031652 000006 SEEK
30 031654 004537 032404 JSR R5,LDFUNC ;VERIFY POSITION
31 031660 000010 RDHDR
32 031662 013700 003172 MOV E.MP,R0 ;GET HEADER
33 031666 022737 000001 003060 CMP #1,T.DRIVE
34 031674 001003 BNE 42$
35 031676 022700 077700 CMP #77700,R0
36 031702 000402 BR 43$
37 031704 022700 177700 42$: CMP #177700,R0
38
39 031710 103321 43$: BHIS 1$
40 031712 022737 000001 003060 2$: CMP #1,T.DRIVE
41 031720 001004 BNE 52$
42 031722 012737 077700 003204 MOV #77700,BDA
43 031730 000403 BR 97$
44
45 031732 012737 177700 003204 52$: MOV #177700,BDA
46 031740 012737 003232 003202 97$: MOV #BUF,BBA
47 031746 012737 177400 003206 MOV #-256.,BMP
48 031754 004537 032404 JSR R5,LDFUNC ;READ
49 031760 000014 READ
50 031762 005737 003074 TST ERFLG ;TEST ERROR FLAG
51 031766 001421 BEQ 98$ ;YES,COMPARE SERIAL NUMBERS
52 031770 062737 000004 003204 ADD #4,BDA ;NO,SETUP FOR NEXT FACTORY BAD SECTOR
53 031776 022737 000001 003060 CMP #1,T.DRIVE
54 032004 001005 BNE 62$
55 032006 022737 077724 003204 CMP #77724,BDA
56 032014 001351 BNE 97$
57 032016 000453 BR 99$
    
```



```

58 032020 022737 177724 003204 62$:  CMP      #177724,BDA
59 032026 001344                BNE      97$
60 032030 000446                BR       99$
61 032032 012701 003232 98$:  MOV      #BUF,R1
62 032036 005737 003210        TST      SERNM1
63 032042 100005                BPL      3$
64 032044 011137 003210        MOV      (R1),SERNM1
65 032050 016137 000002 003212  MOV      2(R1),SERNM2
66 032056 021137 003210 3$:   CMP      (R1),SERNM1
67 032062 001004                BNE      4$
68 032064 026137 000002 003212  CMP      2(R1),SERNM2
69 032072 001437                BEQ      5$
70 032074                4$:   PRINTF  #FRM3,2(R1),(R1),SERNM2,SERNM1
    032074 013746 003210        MOV      SERNM1,-(SP)
    032100 013746 003212        MOV      SERNM2,-(SP)
    032104 011146                MOV      (R1),-(SP)
    032106 016146 000002        MOV      2(R1),-(SP)
    032112 012746 021167        MOV      #FRM3,-(SP)
    032116 012746 000005        MOV      #5,-(SP)
    032122 010600                MOV      SP,R0
    032124 104417                TRAP    C$PNTF
    032126 062706 000014        ADD     #14,SP
71 032132 004537 032174        JSR     R5,UNLOAD
72 032136 004537 032300        JSR     R5,LOAD
73 032142 000137 031526        JMP     SERNUM
74 032146                99$:  PRINTF  #FRM15
    032146 012746 022037        MOV      #FRM15,-(SP)
    032152 012746 000001        MOV      #1,-(SP)
    032156 010600                MOV      SP,R0
    032160 104417                TRAP    C$PNTF
    032162 062706 000004        ADD     #4,SP
75 032166                999$: BREAK
    032166 104422                TRAP    C$BRK
76 032170 000776                BR      999$
77 032172 000205                5$:   RTS     R5
    
```

```

;GET NEXT FACTORY BAD SECTOR
;REPORT ERROR
;COMPARE SERIAL NUMBERS
;HAVE WE GOT ONE TO COMPARE
;YES, BRANCH
;NO, CALL THIS ONE IT
;
;SERNUM OKAY
;NO, PRINT ERROR
;OTHER HALF OKAY
;YES, EXIT
;LET OPERATOR CHANGE
;PACK
;GO CHECK IT AGAIN.
;MESSAGE
    
```

```

1      ;ROUTINE UNLOAD
2      ;CALL   JSR   R5,UNLOAD
3
4
5      UNLOAD: PRINTF #FRM1,<B,DSB+1(R4)>,CSR(R4) ;PROMPT - UNLOAD DRIVE ON CONTROLLER _
6      MOV   CSR(R4),-(SP) ;AND REMOVE PACK
7      CLR   -(SP)
8      BISB  DSB+1(R4),(SP)
9      MOV   #FRM1,-(SP)
10     MOV   #3,-(SP)
11     MOV   SP,R0
12     TRAP  C$PNTF
13     ADD   #10,SP
14     MOV   #60.,R1 ;SETUP 60 SECOND TIMER
15     MOV   #200,R0
16     BIS   DSB(R4),R0
17     MOV   R0,@CSR(R4)
18     2$:   BIT   #DRDY,@CSR(R4) ;CHECK DRDY FOR ZERO
19     BEQ   3$ ;PACK UNLOADED
20     WAITMS #10. ;WAIT 1 SECOND
21     DEC   R1 ;HAS 60 SEC PASSED?
22     BNE   2$ ;NO, RETEST DRDY, CONTINUE WAIT
23     BR   UNLOAD ;YES, REPEAT MESSAGE CONTINUE WAIT
24     3$:   RTS   R5 ;RETURN WITH PACK UNLOADED
25
26     ;ROUTINE LOAD
27     ;CALL   JSR   R5,LOAD
28
29
30     LOAD: PRINTF #FRM2,<B,DSB+1(R4)>,CSR(R4) ;PLACE PACK IN DRIVE ON CONTROLLER _ AND
31     MOV   CSR(R4),-(SP) ;LOAD IT
32     CLR   -(SP)
33     BISB  DSB+1(R4),(SP)
34     MOV   #FRM2,-(SP)
35     MOV   #3,-(SP)
36     MOV   SP,R0
37     TRAP  C$PNTF
38     ADD   #10,SP
39     MOV   #120.,R1 ;SETUP 120 SEC TIMER
40     MOV   #200,R0 ;SETUP CONTROLLER READY BIT
41     BIS   DSB(R4),R0 ;SELECT DRIVE
42     MOV   R0,@CSR(R4)
43     2$:   BIT   #DRDY,@CSR(R4)
44     BNE   3$
45     WAITMS #10.
46     DEC   R1
47     BNE   2$
48     BR   LOAD
49     3$:   RTS   R5

```

```

1          ;ROUTINE LDFUNC
2          ;CALL   JSR      R5,LDFUNC
3
4 032404 010046          LDFUNC: MOV      R0,-(SP)
5 032406 010346          MOV      R3,-(SP)
6 032410 010146          MOV      R1,-(SP)
7 032412 005037 003074  CLR      ERFLG          ;CLEAR ERROR FLAG
8 032416 016403 000000  MOV      CSR(R4),R3      ;GET CSR
9 032422 013763 003206 000006  MOV      BMP,MP(R3)      ;LOAD MULTIPURPOSE
10 032430 013763 003204 000004  MOV      BDA,DA(R3)      ;LOAD DISK ADDRESS
11 032436 013763 003202 000002  MOV      BBA,BA(R3)      ;LOAD BUS ADDRESS
12 032444 011537 003200          MOV      (R5),BCS        ;GET FUNCTION TO LOAD
13 032450 056437 000004 003200  BIS      DSB(R4),BCS      ;SELECT BITS
14 032456 012701 000031          MOV      #25.,R1        ;SET WATCHDOG TO 250MS
15 032462 052737 000200 003200  BIS      #200,BCS
16 032470 013763 003200 000000  MOV      BCS,CS(R3)      ;LOAD FUNCTION
17 032476 016337 000000 003200  MOV      CS(R3),BCS
18 032504 042763 000200 000000  BIC      #200,CS(R3)
19 032512 032763 000200 000000  1$: BIT      #200,CS(R3)  ;CNTLR READY?
20 032520 001036          BNE      2$              ;YES, GO
21 032522          WAITUS #100.      ;WAIT 10 MILLISECONDS
22 032534 005301          DEC      R1
23 032536 001365          BNE      1$
24
25 032540 016337 000000 003164  MOV      CS(R3),E.CS      ;READ ALL REGISTERS
26 032546 016337 000002 003166  MOV      BA(R3),E.BA
27 032554 016337 000004 003170  MOV      DA(R3),E.DA
28 032562 016337 000006 003172  MOV      MP(R3),E.MP
29 032570 016337 000006 003174  MOV      MP(R3),E.MP1
30 032576 016337 000006 003176  MOV      MP(R3),E.MP2
31 032604          ERRDF 210.,CNTTOT,ERR5;CNTLR TIMEOUT
   032604 104455  TRAP  C$ERDF
   032606 000322  .WORD 210
   032610 017346  .WORD CNTTOT
   032612 020644  .WORD ERR5
32 032614 000425  BR      4$
33
34 032616 016337 000000 003164  2$: MOV      CS(R3),E.CS      ;READ ALL REGISTERS
35 032624 016337 000002 003166  MOV      BA(R3),E.BA
36 032632 016337 000004 003170  MOV      DA(R3),E.DA
37 032640 016337 000006 003172  MOV      MP(R3),E.MP
38 032646 016337 000006 003174  MOV      MP(R3),E.MP1
39 032654 016337 000006 003176  MOV      MP(R3),E.MP2
40
41 032662 005737 003164          TST      E.CS          ;ANY ERRORS?
42 032666 100002          BPL      3$              ;YES, GO SERVICE
43 032670 005237 003074          4$: INC      ERFLG
44 032674 005725          3$: TST      (R5)+
45 032676 012601          MOV      (SP)+,R1
46 032700 012603          MOV      (SP)+,R3
47 032702 012600          MCV      (SP)+,R0
48 032704 000205          RTS      R5
49
50 032706          ENDMOD
51          .SBTTL CONTROL ROUTINE
    
```

1	032706			BGNMOD	HRDWTST	
2	032706			BGNTST		
3						
4						:CONTROL SECTION COMPATIBILITY PROGRAM
5						:PRINT UNLOAD AND LOAD DRIVE MESSAGES
6						:PERFORM SERIAL CHECK ROUTINE
7						:PERFORM READ/WRITE CHECKS ON DRIVES
8						
9	032706	012701	002442	COMPAT:	MOV	#SECBUF,R1 ;ADJ. CYLINDER BUFFER
10	032712	012700	000170		MOV	#120.,R0 ;ADJ. CYLINDER BUFFER COUNT
11	032716	005021		4\$:	CLR	(R1)+ ;CLEAR ADJ. CYL. BUFFER AT STARTUP
12	032720	005300			DEC	R0 ;BUFFER CLEARED?
13	032722	001375			BNE	4\$;CLEAR NEXT BUFFER WORD
14	032724	005237	003062		INC	F0WR ;SET FIRST OVERWRITE FLAG
15	032730	004537	024606		JSR	R5,OVWPER ;PERFORM OVERWRITE ON FIRST DRIVE
16	032734	177400				
17	032736	000377				
18	032740	005037	003062		CLR	F0WR ;CLEAR FIRST OVERWRITE
19	032744	005237	003064		INC	FADJ ;SET FIRST ADJ. FLAG
20	032750	005237	003104		INC	ADJDIR ;UP = 1
21	032754	004537	025330		JSR	R5,ADJCYL
22	032760	003	377		.BYTE	3,377 ;TRACK AND SECTORS FOR
23	032762	170000			.WORD	170000 ;INWARD SEEK
24	032764	003	000		.BYTE	3,0 ;TRACK AND SECTORS FOR
25	032766	007777			.WORD	7777 ;OUTWARD SEEK
26	032770	000000			.WORD	0 ;TERMINATOR
27	032772	004537	032174		JSR	R5,UNLOAD ;UNLOAD PACK FROM DRIVE UNIT
28	032776	062704	000010		ADD	#PAT+2,R4 ;UPDATE POINTER FOR NEXT DRIVE
29	033002	004537	032300		JSR	R5,LOAD ;LOAD INTO SECOND DRIVE UNIT
30	033006	004537	031526		JSR	R5,SERNUM ;CHECK PACK SERIAL NUMBER
31	033012	004537	024606		JSR	R5,OVWPER ;PERFORM R/W OVERWRITE
32	033016	000360				
33	033020	000017				
34	033022	005237	003104		INC	ADJDIR
35	033026	004537	025330		JSR	R5,ADJCYL
36	033032	002	360		.BYTE	2,360 ;IN 1/0 OUTSIDE
37	033034	000000			.WORD	0
38	033036	002	017		.BYTE	2,17 ;OUT 1/0 OUTSIDE
39	033040	000000			.WORD	0
40	033042	004	360		.BYTE	4,360 ;IN 1/0 INSIDE
41	033044	000000			.WORD	0
42	033046	004	017		.BYTE	4,17 ;OUT 1/0 INSIDE
43	033050	000000			.WORD	0
44	033052	000000			.WORD	0
45	033054	004537	032174		JSR	R5,UNLOAD ;UNLOAD PACK FROM DRIVE UNIT
46	033060	023727	003130	000002	CMP	UUT,#2 ;CHECK FOR > 2 DRIVES
47	033066	001002			BNE	10\$;YES,GO TO NEXT DRIVE
48	033070	000137	033504		JMP	2\$;GO TO FIRST DRIVE
49	033074	062704	000010	10\$:	ADD	#PAT+2,R4 ;UPDATE DRIVE BUFFER FOR THIRD DRIVE
50	033100	004537	032300		JSR	R5,LOAD ;LOAD PACK FOR THIRD DRIVE
51	033104	004537	031526		JSR	R5,SERNUM ;CHECK SERIAL NUMBERS
52	033110	004537	024606		JSR	R5,OVWPER ;PERFORM R/W OVERWRITE ON THIRD DRIVE
53	033114	006014				
54	033116	001403				
55	033120	005237	003104		INC	ADJDIR
56	033124	004537	025330		JSR	R5,ADJCYL
57	033130	002	000		.BYTE	2,0 ;IN 2/0 OUTSIDE

58	033132	170000			.WORD	170000	
59	033134	002	000		.BYTE	2,0	:OUT 2/0 OUTSIDE
60	033136	007400			.WORD	7400	
61	033140	004	000		.BYTE	4,0	:IN 2/0 INSIDE
62	033142	170000			.WORD	170000	
63	033144	004	000		.BYTE	4,0	:OUT 2/0 INSIDE
64	033146	007400			.WORD	7400	
65	033150	001	200		.BYTE	1,200	:IN 2/1 OUTSIDE
66	033152	000000			.WORD	0	
67	033154	001	100		.BYTE	1,100	:OUT 2/1 OUTSIDE
68	033156	000000			.WORD	0	
69	033160	005	200		.BYTE	5,200	:IN 2/1 INSIDE
70	033162	000000			.WORD	0	
71	033164	005	100		.BYTE	5,100	:OUT 2/1 INSIDE
72	033166	000000			.WORD	0	
73	033170	000000			.WORD	0	:TERMINATOR
74	033172	004537	032174		JSR	R5,UNLOAD	:UNLOAD PACK ON THIRD DRIVE
75	033176	023727	003130	000003	CMP	UUT,#3	:CHECK FOR > 3 DRIVES
76	033204	001500			BEQ	1\$:NO, GO TO 2ND DRIVE
77	033206	062704	000010		ADD	#PAT+2,R4	:UPDATE DRIVE BUFFER FOR 4TH DRIVE
78	033212	004537	032300		JSR	R5,LOAD	:LOAD PACK ON 4TH DRIVE
79	033216	004537	031526		JSR	R5,SERNUM	:CHECK PACK ON FOURTH DRIVE
80	033222	004537	024606		JSR	R5,OVWPER	:PERFORM R/W OVERWRITE
81	033226	001042				1042	
82	033230	000421				421	
83	033232	005237	003104		INC	ADJDIR	
84	033236	004537	025330		JSR	R5,ADJCYL	
85	033242	002	000		.BYTE	2,0	:IN 3/0 OUTSIDE
86	033244	000360			.WORD	360	
87	033246	002	000		.BYTE	2,0	:OUT 3/0 OUTSIDE
88	033250	000017			.WORD	17	
89	033252	004	000		.BYTE	4,0	:IN 3/0 INSIDE
90	033254	000360			.WORD	360	
91	033256	004	000		.BYTE	4,0	:OUT 3/0 INSIDE
92	033260	000017			.WORD	17	
93	033262	001	040		.BYTE	1,40	:IN 3/1 OUTSIDE
94	033264	000000			.WORD	0	
95	033266	001	020		.BYTE	1,20	:OUT 3/1 OUTSIDE
96	033270	000000			.WORD	0	
97	033272	005	040		.BYTE	5,40	:IN 3/1 INSIDE
98	033274	000000			.WORD	0	
99	033276	005	020		.BYTE	5,20	:OUT 3/1 INSIDE
100	033300	000000			.WORD	0	
101	033302	001	000		.BYTE	1,0	:IN 3/2 OUTSIDE
102	033304	100000			.WORD	100000	
103	033306	001	000		.BYTE	1,0	:OUT 3/2 OUTSIDE
104	033310	040000			.WORD	40000	
105	033312	005	000		.BYTE	5,0	:IN 3/2 INSIDE
106	033314	100000			.WORD	100000	
107	033316	005	000		.BYTE	5,0	:OUT 3/2 INSIDE
108	033320	040000			.WORD	40000	
109	033322	000000			.WORD	0	:TERMINATOR
110	033324	004537	032174		JSR	R5,UNLOAD	:UNLOAD PACK FROM 4TH DRIVE
111	033330	162704	000010		SUB	#PAT+2,R4	:SET DRIVE BUFFER FOR 3RD DRIVE
112	033334	004537	032300		JSR	R5,LOAD	:LOAD PACK ON 3RD DRIVE
113	033340	004537	031526		JSR	R5,SERNUM	:CHECK FOR PACK SERIAL NUMBER
114	033344	004537	024606		JSR	R5,OVWPER	:PERFORM R/W OVERWRITE ON 3RD DRIVE

115	033350	020000		20000		
116	033352	010000		10000		
117	033354	004537	025330	JSR	R5,ADJCYL	
118	033360	001	000	.BYTE	1,0	:IN 2/3 OUTSIDE
119	033362	000200		.WORD	200	
120	033364	001	000	.BYTE	1,0	:OUT 2/3 OUTSIDE
121	033366	000100		.WORD	100	
122	033370	005	000	.BYTE	5,0	:IN 2/3 INSIDE
123	033372	000200		.WORD	200	
124	033374	005	000	.BYTE	5,0	:OUT 2/3 INSIDE
125	033376	000100		.WORD	100	
126	033400	000000		.WORD	0	:TERMINATOR
127	033402	004537	032174	JSR	R5,UNLOAD	:UNLOAD PACK FROM 3RD DRIVE
128	033406	162704	000010	1\$: SUB	#PAT+2,R4	:SET DRIVE BUFFER FOR 2ND DRIVE
129	033412	004537	032300	JSR	R5,LOAD	:LOAD PACK ON THIRD DRIVE
130	033416	004537	031526	JSR	R5,SERNUM	:CHECK PACK SERIAL NUMBER
131	033422	004537	024606	JSR	R5,OVWPER	:PERFORM R/W OVERWRITE ON 2ND DRIVE
132	033426	004040		4040		
133	033430	002020		2020		
134	033432	004537	025330	JSR	R5,ADJCYL	
135	033436	001	000	.BYTE	1,0	:IN 1/2 OUTSIDE
136	033440	020000		.WORD	20000	
137	033442	001	000	.BYTE	1,0	:OUT 1/2 OUTSIDE
138	033444	010000		.WORD	10000	
139	033446	005	000	.BYTE	5,0	:IN 1/2 INSIDE
140	033450	020000		.WORD	20000	
141	033452	005	000	.BYTE	5,0	:OUT 1/2 INSIDE
142	033454	010000		.WORD	10000	
143	033456	001	000	.BYTE	1,0	:IN 1/3 OUTSIDE
144	033460	000040		.WORD	40	
145	033462	001	000	.BYTE	1,0	:OUT 1/3 OUTSIDE
146	033464	000020		.WORD	20	
147	033466	005	000	.BYTE	5,0	:IN 1/3 INSIDE
148	033470	000040		.WORD	40	
149	033472	005	000	.BYTE	5,0	:OUT 1/3 INSIDE
150	033474	000020		.WORD	20	
151	033476	000000		.WORD	0	:TERMINATOR
152	033500	004537	032174	JSR	R5,UNLOAD	:UNLOAD PACK FROM 2ND DRIVE
153	033504	162704	000010	2\$: SUB	#PAT+2,R4	:SET DRIVE BUFFER FOR 1ST DRIVE
154	033510	004537	032300	JSR	R5,LOAD	:LOAD PACK INTO FIRST DRIVE UNIT
155	033514	004537	031526	JSR	R5,SERNUM	:CHECK SERIAL NUMBER
156	033520	004537	024606	JSR	R5,OVWPER	:PERFORM R/W OVERWRITE
157	033524	001042		1042		
158	033526	000421		421		
159	033530	004537	025330	JSR	R5,ADJCYL	
160	033534	001	010	.BYTE	1,10	:IN 0/1 OUTSIDE
161	033536	000000		.WORD	0	
162	033540	001	004	.BYTE	1,4	:OUT 0/1 OUTSIDE
163	033542	000000		.WORD	0	
164	033544	005	010	.BYTE	5,10	:IN 0/1 INSIDE
165	033546	000000		.WORD	0	
166	033550	005	004	.BYTE	5,4	:OUT 0/1 INSIDE
167	033552	000000		.WORD	0	
168	033554	001	000	.BYTE	1,0	:IN 0/2 OUTSIDE
169	033556	004000		.WORD	4000	
170	033560	001	000	.BYTE	1,0	:OUT 0/2 OUTSIDE
171	033562	002000		.WORD	2000	

172	033564	005	000	.BYTE	5,0	;IN 0/2 INSIDE
173	033566	004000		.WORD	4000	
174	033570	005	000	.BYTE	5,0	;OUT 0/2 INSIDE
175	033572	002000		.WORD	2000	
176	033574	001	000	.BYTE	1,0	;IN 0/3 OUTSIDE
177	033576	000010		.WORD	10	
178	033600	001	000	.BYTE	1,0	;OUT 0/3 OUTSIDE
179	033602	000004		.WORD	4	
180	033604	005	000	.BYTE	5,0	;IN 0/3 INSIDE
181	033606	000010		.WORD	10	
182	033610	005	000	.BYTE	5,0	;OUT 0/3 INSIDE
183	033612	000004		.WORD	4	
184	033614	000000		.WORD	0	;TERMINATOR
185	033616	004537	032174	JSR	RS,UNLOAD	;UNLOAD PACK
186	033622			PRINTF	#ENDPAS	;END OF PASS
	033622	012746	022412	MOV	#ENDPAS,-(SP)	
	033626	012746	000001	MOV	#1,-(SP)	
	033632	010600		MOV	SP,R0	
	033634	104417		TRAP	C\$PNTF	
	033636	062706	000004	ADD	#4,SP	
187						
188	033642	000137	024266	JMP	CMPEA	;RETURN TO SUPERVISOR
189						
190						
191	033646			ENDTST		
	033646			L10014:		
	033646	104401		TRAP	C\$ETST	
192	033650			ENDMOD		
193						
194	033650			BGNMOD	HRDPRM	
195	033650			BGNHRD		
	033650	000025		.WORD	L10015-L\$HARD/2	
196						
197	033652			GPRMA	CSRMSG,CSR,0,160000,177776,YES	
	033652	000031		.WORD	T\$CODE	
	033654	033724		.WORD	CSRMSG	
	033656	160000		.WORD	T\$LOLIM	
	033660	177776		.WORD	T\$HILIM	
198						
199	033662			GPRMA	VECMMSG,VECT,0,0,776,YES	
	033662	001031		.WORD	T\$CODE	
	033664	033762		.WORD	VECMMSG	
	033666	000000		.WORD	T\$LOLIM	
	033670	000776		.WORD	T\$HILIM	
200						
201	033672			GPRMD	DRMSG,DRBT,0,03400,0,7,YES	
	033672	004032		.WORD	T\$CODE	
	033674	033771		.WORD	DRMSG	
	033676	003400		.WORD	03400	
	033700	000000		.WORD	T\$LOLIM	
	033702	000007		.WORD	T\$HILIM	
202						
203	033704			GPRML	DRTYPE,TYPDR,1,YES	
	033704	003130		.WORD	T\$CODE	
	033706	033740		.WORD	DRTYPE	
	033710	000001		.WORD	1	
204						

```

205 033712          GPRMD  BRMSG,PRIOR,0,340,0,7,YES
    033712 002032   .WORD  T$CODE
    033714 033777   .WORD  BRMSG
    033716 000340   .WORD  340
    033720 000000   .WORD  T$LOLIM
    033722 000007   .WORD  T$HILIM

206
207 033724          ENDHRD
    033724          .EVEN
    033724          L10015:

208
209 033724          102      125      123  CSRMSG: .ASCIZ /BUS ADDRESS/
    033727          040      101      104
    033732          104      122      105
    033735          123      123      000
210 033740          104      122      111  DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
    033743          126      105      040
    033746          124      131      120
    033751          105      040      075
    033754          040      122      114
    033757          060      061      000
211 033762          126      105      103  VECMSG: .ASCIZ /VECTOR/
    033765          124      117      122
    033770          000
212 033771          104      122      111  DRMSG: .ASCIZ /DRIVE/
    033774          126      105      000
213 033777          102      122      040  BRMSG: .ASCIZ /BR LEVEL/
    034002          114      105      126
    034005          105      114      000

214
215          .EVEN
216
217 034010          ENDMOD
218
219 034010          LASTAD
    034010 000000   .EVEN
    034012 000000   .WORD  0
    034014          .WORD  0
    034014          L$LAST::

220
221          000001   .END
    
```


ADJCYL 025330	CRDY = 000200	C\$SVEC= 000037	FRM12 021646	G\$OFSI= 000376
ADJDIR 003104	CRSET = 000002	C\$TPRI= 000013	FRM13 021725	G\$PRMA= 000001
ADJFLG 003102	CS = 000000	DA = 000004	FRM14 022010	G\$PRMD= 000002
ADJLC2 003220	CSR = 000000	DCKER 017455	FRM15 022037	G\$PRML= 00000C
ADJLC3 003222	CSRMSG 033724	DCRC = 004000	FRM16 022076	G\$RADA= 000140
ADJLC4 003224	CYL 003122	DERR = 040000	FRM17 022163	G\$RADB= 000000
ADJLOC 003100	C\$AU = 000052	DIAGMC= 000000	FRM18 022217	G\$RADD= 000040
ADJTRK 003214	C\$AUTO= 000061	DIRC 003116	FRM19 022303	G\$RADL= 000120
ADJTXT 020210	C\$BRK = 000022	DLT = 010000	FRM2 021067	G\$RADO= 000020
ADJUUT 003216	C\$BSEG= 000004	DRBT = 000010	FRM20 022341	G\$XFER= 000004
ADR = 000020 G	C\$BSUB= 000002	DRBUF 017240	FRM3 021167	G\$YES = 000010
ASSEMB= 000010	C\$CEFG= 000045	DRDY = 000001	FRM4 021246	HCRC = 004000
AUTOCO 024312 G	C\$CLCK= 000062	DRMSG 033771	FRM5 021307	HDRFND 002234
BA = 000002	C\$CLEA= 000012	DRPCOD 024322 G	FRM6 021356	HEAD = 000100
BA16 = 000020	C\$CLOS= 000035	DRST = 000013	FRM7 021377	HEAD01 003114
BA17 = 000040	C\$CLP1= 000006	DRSTAT 003106	FRM8 021437	HNF = 010000
BBA 003202	C\$CVEC= 000036	DRTYPE 033740	FRM9 021503	HOE = 100000 G
BCS 003200	C\$DCLN= 000044	DSB = 000004	FRTTRK 003152	HPTCOD 022446 G G
BDA 003204	C\$DODU= 000051	DSPCOD 022462 G	FUNERR 020047	HRDPRM 033650 G G
BDATA 003140	C\$DRPT= 000024	EF.COM= 000036 G G	FWD 020116	HRDWT 032706 G
BIT0 = 000001 G	C\$DU = 000053	EF.NEW= 000035 G G	F\$AU = 000015	HSFLG 003110
BIT00 = 000001 G G	C\$EDIT= 000003	EF.PWR= 000034 G G	F\$AUTO= 000020	IBE = 010000 G G
BIT01 = 000002 G G	C\$ERDF= 000055	EF.RES= 000037 G G	F\$BGN = 000040	IDU = 000040 G G
BIT02 = 000004 G G	C\$ERHR= 000056	EF.STA= 000040 G	F\$CLEA= 000007	IER = 020000 G
BIT03 = 000010 G G	C\$ERRO= 000060	END 023020	F\$DU = 000016	INAWR 030454
BIT04 = 000020 G G	C\$ERSF= 000054	ENDBUF 017300	F\$END = 000041	INITCO 022466 G
BIT05 = 000040 G G	C\$ERSO= 000057	ENDPAS 022412	F\$HARD= 000004	INITWR 017373
BIT06 = 000100 G G	C\$ESCA= 000010	ERFLG 003074	F\$HW = 000013	INN10 002356
BIT07 = 000200 G G	C\$ESEG= 000005	ERR = 100000	F\$INIT= 000006	INN11 002370
BIT08 = 000400 G G	C\$ESUB= 000003	ERRFND 017743	F\$JMP = 000050	INN20 002360
BIT09 = 001000 G G	C\$ETST= 000001	ERR1 020240 G	F\$MOD = 000000	INN21 002372
BIT1 = 000002 G G	C\$EXIT= 000032	ERR2 020276 G G	F\$MSG = 000011	INN30 002362
BIT10 = 002000 G G	C\$GETB= 000026	ERR3 020336 G G	F\$PROT= 000021	INN31 002374
BIT11 = 004000 G G	C\$GETW= 000027	ERR4 020464 G G	F\$PWR = 000017	INN40 002364
BIT12 = 010000 G G	C\$GMAN= 000043	ERR5 020644 G G	F\$RPT = 000012	INN41 002376
BIT13 = 020000 G G	C\$GPHR= 000042	ERR6 020704 G G	F\$SEG = 000003	INN50 002366
BIT14 = 040000 G G	C\$GPLO= 000030	EVL = 000004 G	F\$SOFT= 000005	INN51 002400
BIT15 = 100000 G G	C\$GPRI= 000040	EXIT 024310	F\$SRV = 000010	INTEN = 000100
BIT2 = 000004 G G	C\$INIT= 000011	E\$END = 002100	F\$SUB = 000002	ISR = 000100 G
BIT3 = 000010 G G	C\$INLP= 000020	E\$LOAD= 000035	F\$SW = 000014	IXE = 004000 G
BIT4 = 000020 G G	C\$MANI= 000050	E.BA 003166	F\$TEST= 000001	ISAU = 000041
BIT5 = 000040 G G	C\$MEM = 000031	E.CS 003164	GDATA 003136	ISAUTO= 000041
BIT6 = 000100 G G	C\$MSG = 000023	E.DA 003170	GLBDAT 002234 G	ISCLN = 000041
BIT7 = 000200 G G	C\$OPEN= 000034	E.MP 003172	GLBEQA 002234 G G	ISDU = 000041
BIT8 = 000400 G G	C\$PNTB= 000014	E.MP1 003174	GLBERR 020240 G G	ISHRD = 000041
BIT9 = 001000 G G	C\$PNTF= 000017	E.MP2 003176	GLBSUB 024326 G G	ISINIT= 000041
BMP 003206	C\$PNTS= 000016	FADJ 003064	GLBTXT 017302 G	ISMOD = 000041
BOE = 000400 G	C\$PNTX= 000015	FEW 017473	GSBIT = 000003	ISMSG = 000041
BRMSG 033777	C\$QIO = 000377	FIXCYL 031472	GSTAT = 000004	ISPROT= 000040
BSEXIT 030540	C\$RDBU= 000007	FNDASC 031402	G\$CNT0= 000200	ISPTAB= 000041
BSVWR 030356	C\$REFG= 000047	FNDDRV 027140	G\$DELM= 000372	ISPWR = 000041
BUF 003232	C\$RESE= 000033	FNDTRK 031230	G\$DISP= 000003	ISRPT = 000041
CHECK 030566	C\$REVI= 000003	FORSK 003126	G\$EXCP= 000400	ISSEG = 000041
CLNCOD 024316 G	C\$RFLA= 000021	FOWR 003062	G\$HILI= 000002	ISSETU= 000041
COMPEN 024266	C\$RPT = 000025	FRM1 020772	G\$LOLI= 000001	ISSRV = 000041
CNTTOT 017346	C\$SEFG= 000046	FRM10 021545	G\$NO = 000000	ISSUB = 000041
COMPAT 032706	C\$SPRI= 000041	FRM11 021611	G\$OFFS= 000400	ISTST = 000041

JSJMP = 000167	L\$SPC 002056 G	OQU40 002270	REV 020126	T\$HILI= 000007
LDFUNC 032404	L\$SPCP 002020 G	OQU41 002302	REVSX 003124	T\$LAST= 000001
LOAD 032300	L\$SPTP 002024 G	OQU50 002272	RSADJS 026732	T\$LOLI= 000000
LOE = 040000 G	L\$STA 002030 G	OQU51 002304	SECBUF 002442	T\$LSYM= 010000
LOT = 000010 G	L\$TEST 002114 G	OSECT 003112	SECLST 002402	T\$LTNO= 000001
LSTCLR 003070	L\$TIML 002014 G	OUT10 002236	SECT 003132	T\$NEST= 177777
LSTDRV 003134	L\$UNIT 002012 G	OUT11 002250	SECWRD 003144	T\$NSO = 000000
LSTTRK 003150	L10000 020274	OUT20 002240	SEEK = 000006	T\$NS1 = 000004
L\$ACP 002110 G	L10001 020334	OUT21 002252	SERNM1 003210	T\$PTNU= 000000
L\$APT 002036 G	L10002 020462	OUT30 002242	SERNM2 003212	T\$SAVL= 177777
L\$AUT 002070 G	L10003 020642	OUT31 002254	SERNUM 031526	T\$SEGL= 177777
L\$AUTO 024312 G	L10004 020702	OUT40 002244	SETLST 027100	T\$SUBN= 000000
L\$CCP 002106 G	L10005 020770	OUT41 002256	SETUP 023072	T\$TAGL= 177777
L\$CLEA 024316 G	L10007 022462	OUT50 002246	SIGN = 000004	T\$TAGN= 010016
L\$CO 002032 G	L10010 024310	OUT51 002260	SKCYL 026254	T\$TEMP= 000000
L\$DEPO 002011 G	L10011 024314	OVMS 017642	SKER 020077	T\$TEST= 000001
L\$DESC 002122 G	L10012 024320	OVWER 020003	SKHS = 000020	T\$TSTM= 177777
L\$DESC 002076 G	L10013 024324	OVWPER 024606	STFLG 003076	T\$TSTS= 000001
L\$DEVP 002060 G	L10014 033646	OVWTRK 003022	STSEC 003230	T\$SAUT= 010011
L\$DISP 022464 G	L10015 033724	OSAPTS= 000000	STSEC1 003226	T\$SCLE= 010012
L\$DLY 002116 G	MANY 017532	OSAU = 000000	SURF 003120	T\$SDU = 010013
L\$DTP 002040 G	MDHEDR 002000 G	OSBGNR= 000000	SURFAC 003156	T\$SHAR= 010015
L\$DTYP 002034 G	MERGE 031002	OSBGNS= 000000	SVCGBL= 000000	T\$SHW = 010007
L\$DU 024322 G	MID10 002306	OSDU = 000000	SVCINS= 000000	T\$SINI= 010010
L\$DUT 002072 G	MID11 002320	OSERRT= 000000	SVCSUB= 177777	T\$SMMSG= 010005
L\$DVTY 002222 G	MID20 002310	OSGNSW= 000000	SVCTAG= 000000	T\$SPRO= 010006
L\$EF 002052 G	MID21 002322	OSPOIN= 000001	SVCTST= 177777	T\$STES= 010014
L\$ENVI 002044 G	MID30 002312	OSSETU= 000000	S\$LSYM= 010000	T.DRIV 003060
L\$ETP 002102 G	MID31 002324	PAT = 000006	TEM 003056	T1 032706 G
L\$EXP1 002046 G	MID40 002314	PATLST 003046	TEMP 003066	UAM = 000200 G
L\$EXP4 002064 G	MID41 002326	PNT = 001000 G	TIME 024326	UNLOAD 032174
L\$EXP5 002066 G	MID50 002316	PRI = 002000 G	TQU10 002332	UUT 003130
L\$SHARD 033652 G	MID51 002330	PRIOR = 000004	TQU11 002344	VAJWR 030112
L\$HIME 002120 G	MK = 000001	PRI00 = 000000 G	TQU20 002334	VEC = 000002
L\$HPCP 002016 G	MP = 000006	PRI01 = 000040 G	TQU21 002346	VECMG 033762
L\$HPTP 002022 G	NCHECK 030552	PRI02 = 000100 G	TQU30 002336	VECT = 000002
L\$HW 022450 G	NONE 017571	PRI03 = 000140 G	TQU31 002350	VEROD 027564
L\$ICP 002104 G	NXM = 020000	PRI04 = 000200 G	TQU40 002340	VEROW 027200
L\$INIT 022466 G	OBUFF 017236	PRI05 = 000240 G	TQU41 002352	WCOUNT 003142
L\$LADP 002026 G	OFFSET 003146	PRI06 = 000300 G	TQU50 002342	WRITE = 000012
L\$LAST 034014 G	OPI = 002000	PRI07 = 000340 G	TQU51 002354	WRIT1 020136
L\$LOAD 002100 G	OPR001 017302	PRSTRK 003154	TRKCN 003162	WRSEC 025070
L\$LUN 002074 G	OPR002 017321	RDHDR = 000010	TRKFND 003160	XDELAY 017232
L\$MREV 002050 G	OQU10 002262	READ = 000014	TYPDR = 000006	XTIME 024452
L\$NAME 002000 G	OQU11 002274	READ1 020163	T\$ARGC= 000001	X\$ALWA= 000000
L\$PRIO 00204 G	OQU20 002264	REASON 003072	T\$CODE= 002032	X\$FALS= 000040
L\$PROT 02244 G	OQU21 002275	RECER 020023	T\$ERRN= 000322	X\$OFFS= 000400
L\$PRT 002112 G	OQU30 002265	RECMS 017675	T\$EXCP= 000000	X\$TRUE= 000020
L\$REPP 002062 G	OQU31 002300	REGDMP 026522	T\$GMAN= 000000	YDELAY 017234
L\$REV 002010 G				

. ABS. 034014 000
 000000 001
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28672 WORDS (112 PAGES)
 DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
 .A:CZRLLC/C=SVC33.SRC/P:1,A:CZRLLC

ISPWR	1-22#									
ISRPT	1-22#									
ISSEG	1-22#	27-2								
ISSETU	1-22#									
ISSRV	1-22#									
ISSUB	1-22#	27-2								
ISTST	1-22#	27-2	27-2#	27-191	27-191#	27-191#				
IBE	1-42#									
IDU	1-42#									
IER	1-42#									
INAWR	19-21	19-27#								
INITCO	6-3#									
INITWR	4-14#	16-30								
INN10	2-51#	6-147								
INN11	2-56#	6-162								
INN20	2-52#									
INN21	2-57#									
INN30	2-53#	2-80								
INN31	2-58#	2-85								
INN40	2-54#									
INN41	2-59#									
INN50	2-55#									
INN51	2-60#	11-134								
INTEN	1-58#									
ISR	1-42#									
IXE	1-42#									
JSJMP	1-22#									
LSACP	1-29#									
LSAPT	1-29#									
LSAUT	1-29#									
LSAUTO	1-29	7-3#								
LSCCP	1-29#									
LSCLEA	1-29	7-12#								
LSCO	1-29#									
LSDEPO	1-29#									
LSDESC	1-29	1-32#								
LSDESP	1-29#									
LSDEVP	1-29#									
LSDISP	1-29	5-27#								
LSDLY	1-29#	8-13*	8-17	8-21*	8-22	8-27*	8-33*	8-34	8-38	
LSCTP	1-29#									
LSDTYP	1-29#									
LSDU	7-20#									
LSDUT	1-29#									
LSDVTY	1-29	1-34#								
LSEF	1-29#									
LSEVI	1-29#									
LSETP	1-29#									
LSEXP1	1-29#									
LSEXP4	1-29#									
LSEXP5	1-29#									
LSHARD	1-29	27-195	27-195#							
LSHIME	1-29#									
LSHPCP	1-29#									
LSHPTP	1-29#									
LSHW	1-29	5-14	5-14#							

	27-197	27-197	27-199	27-199	27-199	27-199	27-199	27-199	27-199	27-199	27-201	27-201	27-201	27-201
	27-201	27-201	27-201	27-201	27-201	27-201	27-203	27-203	27-203	27-203	27-203	27-203	27-205	27-205
	27-205	27-205	27-205	27-205	27-205	27-205	27-205	27-205	27-207	27-207	27-219	27-219	27-219	27-219
	27-219	27-219												
SVCSUB	1-22#													
SVCTAG	1-22#	1-24#	4-43	4-43	4-43	4-48	4-48	4-48	4-57	4-57	4-57	4-67	4-67	4-67
	4-72	4-72	4-72	4-78	4-78	4-78	5-20	5-20	5-20	6-47	6-47	6-47	6-56	6-56
	6-56	6-92	6-92	6-92	6-321	6-321	6-321	7-7	7-7	7-7	7-16	7-16	7-16	7-22
	7-22	7-22	27-191	27-191	27-191	27-207	27-207	27-207						
SVCTST	1-22#	27-2	27-2	27-2										
TSSAUT	7-3#	7-7												
TSSCLE	7-12#	7-16												
TSSDU	7-20#	7-22												
TSSHAR	27-195	27-195#	27-207											
TSSHW	5-14	5-14#	5-20											
TSSINI	6-5#	6-321												
TSSMSG	4-39#	4-43	4-45#	4-48	4-50#	4-57	4-59#	4-67	4-69#	4-72	4-74#	4-78		
TSSPRO	5-4#													
TSSTES	27-2#	27-191												
TSARGC	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	4-41	4-41	4-41	4-41	4-41	4-41#	4-41#	4-41#	4-41#	4-46	4-46
	4-46	4-46	4-46#	4-46#	4-46#	4-51	4-51	4-51	4-51	4-51#	4-51#	4-51#	4-53	4-53
	4-53	4-53	4-53	4-53#	4-53#	4-53#	4-53#	4-55	4-55	4-55	4-55	4-55#	4-55#	4-55#
	4-61	4-61	4-61	4-61	4-61#	4-61#	4-61#	4-63	4-63	4-63	4-63	4-63	4-63#	4-63#
	4-63#	4-63#	4-64	4-64	4-64	4-64	4-64	4-64	4-64#	4-64#	4-64#	4-64#	4-64#	4-65
	4-65	4-65	4-65#	4-65#	4-70	4-70	4-70	4-70	4-70#	4-70#	4-70#	4-70#	4-75	4-75
	4-75	4-75#	4-75#	4-75#	4-77	4-77	4-77	4-77	4-77#	4-77#	4-77#	4-77#	6-45	6-45
	6-54	6-54	6-54#	6-71	6-71	6-71#	6-91	6-91	6-91#	13-4	13-4	13-4	13-4	13-4
	13-4	13-4#	13-4#	13-4#	13-4#	13-4#	13-6	13-6	13-6	13-6	13-6	13-6	13-6#	13-6#
	13-6#	13-6#	13-6#	13-17	13-17	13-17	13-17#	13-17#	16-46	16-46	16-46	16-46	16-46	16-46#
	16-46#	16-46#	16-46#	16-53	16-53	16-53	16-53#	16-53#	17-44	17-44	17-44	17-44	17-44	17-44#
	17-44#	17-44#	17-44#	17-50	17-50	17-50	17-50#	17-50#	21-35	21-35	21-35#	24-70	24-70	24-70
	24-70	24-70	24-70	24-70#	24-70#	24-70#	24-70#	24-70#	24-74	24-74	24-74#	25-5	25-5	25-5
	25-5	25-5#	25-5#	25-5#	25-22	25-22	25-22	25-22	25-22#	25-22#	25-22#	27-186	27-186	27-186#
TS CODE	6-47	6-47	6-47	6-47#	6-47#	6-47#	6-56	6-56	6-56	6-56#	6-56#	6-56#	6-92	6-92
	6-92	6-92#	6-92#	6-92#	27-197	27-197	27-197	27-197#	27-197#	27-197#	27-199	27-199	27-199	27-199#
	27-199#	27-199#	27-201	27-201	27-201	27-201#	27-201#	27-201#	27-203	27-203	27-203	27-203#	27-203#	27-203#
TSERRN	27-205	27-205	27-205	27-205#	27-205#	27-205#								
	1-22#	6-12	6-12#	6-18	6-18#	6-115	6-115#	6-130	6-130#	6-144	6-144#	6-159	6-159#	6-182
	6-182#	6-205	6-205#	6-229	6-229#	6-252	6-252#	6-276	6-276#	6-299	6-299#	6-311	6-311#	10-45
	10-45#	10-48	10-48#	12-54	12-54#	12-57	12-57#	16-39	16-39#	17-35	17-35#	18-41	18-41#	18-44
	18-44#	20-40	20-40#	20-42	20-42#	26-31	26-31#							
TSEXCP	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-205	27-205#						
TSGMAN	1-22#													
TSHILI	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-205	27-205#						
TSLAST	1-22#	27-219#												
TSLOLI	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-205	27-205#						
TSLSYM	1-22	1-22#	4-43	4-48	4-57	4-67	4-72	4-78	5-20	6-321	7-7	7-16	7-22	27-191
	27-207													
TSLTNO	27-219#													
TSNEST	1-22#	1-28	1-28	1-28#	1-30	1-30	1-30	1-30#	1-40	1-40	1-40#	1-91	1-91	1-91
	1-91#	2-4	2-4	2-4#	3-81	3-81	3-81	3-81#	4-2	4-2	4-2#	4-33	4-33	4-33
	4-33#	4-37	4-37	4-37#	4-39	4-39	4-39	4-39#	4-43	4-43	4-43#	4-45	4-45	4-45#
	4-48	4-48	4-48	4-48#	4-50	4-50	4-50	4-50#	4-57	4-57	4-57#	4-59	4-59	4-59#
	4-67	4-67	4-67	4-67#	4-69	4-69	4-69	4-69#	4-72	4-72	4-72#	4-74	4-74	4-74#
	4-78	4-78	4-78	4-78#	4-115	4-115	4-115	4-115#	5-4	5-4	5-4#	5-10	5-10	5-10

	5-10#	5-13	5-13	5-13#	5-14	5-14	5-14#	5-20	5-20	5-20	5-20#	5-22	5-22	5-22
	5-22#	5-25	5-25	5-25#	5-29	5-29	5-29	5-29#	6-3	6-3	6-3#	6-5	6-5	6-5#
	6-321	6-321	6-321	6-321#	6-322	6-322	6-322	6-322#	7-2	7-2	7-2#	7-3	7-3	7-3#
	7-7	7-7	7-7	7-7#	7-8	7-8	7-8	7-8#	7-11	7-11	7-11#	7-12	7-12	7-12#
	7-16	7-16	7-16	7-16#	7-17	7-17	7-17	7-17#	7-19	7-19	7-19#	7-20	7-20	7-20#
	7-22	7-22	7-22	7-22#	7-23	7-23	7-23	7-23#	8-3	8-3	8-3#	26-50	26-50	26-50
	26-50#	27-1	27-1	27-1#	27-2	27-2	27-2#	27-191	27-191	27-191	27-191#	27-192	27-192	27-192
	27-192#	27-194	27-194	27-194#	27-195	27-195	27-195#	27-207	27-207	27-207	27-207#	27-217	27-217	27-217
	27-217#													
TSNSO	1-28#	1-30	1-40#	1-91	2-4#	3-81	4-2#	4-33	4-37#	4-115	5-4#	5-10	5-13#	5-22
	5-25#	5-29	6-3#	6-322	7-2#	7-8	7-11#	7-17	7-19#	7-23	8-3#	26-50	27-1#	27-192
	27-194#	27-217												
TSNS1	4-39#	4-43	4-45#	4-48	4-50#	4-57	4-59#	4-67	4-69#	4-72	4-74#	4-78	5-14#	5-20
	6-5#	6-321	7-3#	7-7	7-12#	7-16	7-20#	7-22	27-2#	27-191	27-195#	27-207		
TSPTNU	1-22#													
TSSAVL	1-22#													
TSSSEGL	1-22#													
TSSUBN	1-22#	27-2#												
TSTAGL	1-22#													
TSTAGN	1-22#	4-39	4-39	4-39#	4-45	4-45	4-45#	4-50	4-50	4-50#	4-59	4-59	4-59#	4-69
	4-69	4-69#	4-74	4-74	4-74#	5-4	5-4	5-4#	5-14	5-14	5-14#	6-5	6-5	6-5#
	7-3	7-3	7-3#	7-12	7-12	7-12#	7-20	7-20	7-20#	27-2	27-2	27-2#	27-195	27-195
	27-195#													
TSTEMP	1-30	1-30#	1-91	1-91#	3-81	3-81#	4-33	4-33#	4-43	4-43#	4-48	4-48#	4-57	4-57#
	4-67	4-67#	4-72	4-72#	4-78	4-78#	4-115	4-115#	5-10	5-10#	5-20	5-20#	5-22	5-22#
	5-27	5-27	5-27#	5-27#	5-29	5-29#	6-47	6-47	6-47	6-47#	6-47#	6-47#	6-56	6-56
	6-56	6-56#	6-56#	6-56#	6-92	6-92	6-92	6-92#	6-92#	6-92#	6-321	6-321#	6-322	6-322#
	7-7	7-7#	7-8	7-8#	7-16	7-16#	7-17	7-17#	7-22	7-22#	7-23	7-23#	26-50	26-50#
	27-191	27-191#	27-192	27-192#	27-197	27-197	27-197	27-197#	27-197#	27-197#	27-199	27-199	27-199	27-199#
	27-199#	27-199#	27-201	27-201	27-201	27-201#	27-201#	27-201#	27-201#	27-203	27-203	27-203	27-203#	27-203#
	27-205	27-205	27-205	27-205#	27-205#	27-205#	27-207	27-207#	27-217	27-217#	27-217#	27-217#	27-217#	27-217#
TSTEST	1-22#	27-2	27-2	27-2#	27-219									
TSTSTM	1-22#	4-41	4-43	4-46	4-48	4-51	4-53	4-55	4-57	4-61	4-63	4-64	4-65	4-67
	4-70	4-72	4-75	4-77	4-78	6-7	6-12	6-18	6-27	6-41	6-45	6-47	6-54	6-56
	6-69	6-71	6-91	6-92	6-115	6-130	6-144	6-159	6-182	6-205	6-229	6-252	6-276	6-299
	6-311	6-314	6-318	6-321	7-7	7-16	7-22	8-15	8-31	10-45	10-48	12-54	12-57	13-4
	13-6	13-17	16-39	16-46	16-53	17-35	17-44	17-50	18-41	18-44	20-40	20-42	21-35	21-36
	24-70	24-74	24-75	25-5	25-22	26-31	27-186	27-191						
TSTSTS	1-22#	27-2#												
T.DRIV	3-3#	6-31*	6-43	9-36	11-83	21-16	21-28	22-3	22-22	24-12	24-19	24-33	24-40	24-53
T1	5-27	27-2#												
TEM	3-2#	10-17*	10-20*	12-14*	12-18*	12-43*	12-46*	18-13*	18-16*	20-10*	20-12*			
TEMP	3-6#	10-32*	10-38	18-26*	18-31*	18-34								
TIME	8-13#	26-21												
TQU10	2-41#	6-279												
TQU11	2-46#	6-302												
TQU20	2-42#													
TQU21	2-47#													
TQU30	2-43#	2-79												
TQU31	2-48#	2-84												
TQU40	2-44#													
TQU41	2-49#													
TQU50	2-45#													
TQU51	2-50#													
TRKCNT	3-36#	22-10*	22-16*	22-18*	22-19									
TRKFND	3-35#	6-112	6-127	6-141	6-156	6-171	6-179	6-193	6-202	6-217	6-226	6-240	6-249	6-264

TYPDR	6-273	6-287	6-296	22-9*	22-21*					
UAM	1-88#	6-43	27-203	27-203	27-203					
UNLOAD	1-42#									
UUT	24-71	25-5#	25-15	27-27	27-45	27-74	27-110	27-127	27-152	27-185
VAJWR	3-23#	6-21*	6-25	6-38	6-62*	6-65*	11-150	27-46	27-75	
VEC	11-59	11-78	11-106	11-119	18-6#					
VECMG	1-52#	3-75	3-75	3-75	3-75	6-29*				
VECT	27-199	27-211#								
VEROD	1-86#	27-199	27-199	27-199						
VEROW	9-34	9-52	17-8#							
WCOUNT	9-33	9-51	16-9#							
WRIT1	3-28#	16-40*	16-45*	16-53	17-37*	17-43*	17-50			
WRITE	4-28#	10-45	10-48							
WRSEC	1-74#	10-40								
XSALWA	9-30	9-48	10-8#	11-56	11-75	11-102	11-114			
XSALS	1-22#									
XSOFFS	1-22#									
XSTRUE	1-22#									
XDELAY	3-57#	8-14*	8-18*	8-23*	26-21*					
XTIME	8-27#	25-12	25-29							
YDELAY	3-9#	8-28*	8-29*	8-30*	8-35*	8-39*	25-12*	25-29*		

BCOMPL	8-16													
BGNAUT	7-3													
BGNCLM	7-12													
BGNDU	7-20													
BGNHRD	27-195													
BGNHW	5-14													
BGNINI	6-5													
BGNMOD	1-28	1-40	2-4	4-2	4-37	5-13	5-25	6-3	7-2	7-11	7-19	8-3	27-1	27-194
BGNMSG	4-39	4-45	4-50	4-59	4-69	4-74								
PGNPRO	5-4													
BGNTST	27-2													
BNCOMP	6-70	8-32												
BREXK	21-36	24-75												
DELAY	8-17	8-22	8-34	8-38										
DESCRI	1-32													
DEVTYP	1-34													
DISPAT	5-27													
DOCLN	6-318													
DODU	6-314													
ENDAUT	7-7													
ENDCLN	7-16													
ENDDU	7-22													
ENDHRD	27-207													
FNDHW	5-20													
ENDINI	6-321													
ENDMOD	1-30	1-91	3-81	4-33	4-115	5-22	5-29	6-322	7-8	7-17	7-23	26-50	27-192	27-217
ENDMSG	4-43	4-48	4-57	4-67	4-72	4-78								
ENDPRO	5-10													
ENDST	27-191													
EQUALS	1-42													
ERRDF	12-54	12-57	16-39	17-35	26-31									
ERRHRD	6-115	6-130	6-144	6-159	6-182	6-205	6-229	6-252	6-276	6-299	10-48	18-44	20-40	20-42
ERRSF	6-12	6-18	6-311											
ERRSOF	10-45	18-41												
GMANIL	6-47	6-56	6-92											
GPHARD	6-27	6-41												
GPRMA	27-197	27-199												
GPRMD	27-201	27-205												
GPRML	6-47	6-47#	6-56	6-56#	6-92	6-92#	27-203							
HEADER	1-29													
LASTAD	27-219													
MSBYTE	1-29	1-29	1-29	1-29#										
MSCNTO	6-47	6-47#	6-56	6-56#	6-92	6-92#	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-203	27-203#
MSCOUN	27-205	27-205#												
	4-41	4-41	4-41	4-41#	4-46	4-46	4-46#	4-51	4-51	4-51#	4-53	4-53	4-53	4-53#
	4-55	4-55	4-55#	4-61	4-61	4-61#	4-63	4-63	4-63	4-63#	4-64	4-64	4-64	4-64
	4-64#	4-65	4-65#	4-70	4-70	4-70#	4-75	4-75	4-75#	4-77	4-77	4-77#	6-45	6-45#
	6-54	6-54#	5-71	6-71#	6-91	6-91#	13-4	13-4	13-4	13-4	13-4#	13-6	13-6	13-6
	13-6	13-6#	13-17	13-17#	16-46	16-46	16-46	16-46#	16-53	16-53#	17-44	17-44	17-44	17-44#
	17-50	17-50#	21-35	21-35#	24-70	24-70	24-70	24-70	24-70	24-70#	24-74	24-74#	25-5	25-5#
	25-22	25-22	25-22#	27-186	27-186#									
MSDATA	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29#	1-29#	1-32	1-32#	1-34
	1-34#													
MSDECR	1-30	1-30#	1-91	1-91#	3-81	3-81#	4-33	4-33#	4-43	4-43#	4-48	4-48#	4-57	4-57#

	4-67	4-67#	4-72	4-72#	4-76	4-78#	4-115	4-115#	5-10	5-10#	5-20	5-20#	5-22	5-22#
	5-29	5-29#	6-321	6-321#	6-322	6-322#	7-7	7-7#	7-8	7-8#	7-16	7-16#	7-17	7-17#
	7-22	7-22#	7-23	7-23#	26-50	26-50#	27-191	27-191#	27-192	27-192#	27-207	27-207#	27-217	27-217#
MSDEFA	6-47	6-47#	6-56	6-56#	6-92	6-92#	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-203	27-203#
	27-205	27-205#												
MSENDE	1-30#	1-91#	3-81#	4-33#	4-43#	4-48#	4-57#	4-67#	4-72#	4-78#	4-115#	5-20#	5-22#	5-29#
	6-321#	6-322#	7-7#	7-8#	7-16#	7-17#	7-22#	7-23#	26-50#	27-191#	27-192#	27-207#	27-217#	
MSERRI	6-12	5-12#	6-18	6-18#	6-115	6-115#	6-130	6-130#	6-144	6-144#	6-159	6-159#	6-182	6-182#
	6-205	6-205#	6-229	6-229#	6-252	6-252#	6-276	6-276#	6-299	6-299#	6-311	6-311#	10-45	10-45#
	10-48	10-48#	12-54	12-54#	12-57	12-57#	16-39	16-39#	17-35	17-35#	18-41	18-41#	18-44	18-44#
	20-40	20-40#	20-42	20-42#	26-31	26-31#								
MSEXCP	27-197	27-197	27-197#	27-199	27-199	27-199#	27-201	27-201	27-201#	27-205	27-205	27-205#		
MSGEN	1-28	1-28#	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	2-4	2-4#	4-2	4-2#	4-37	4-37#	4-39	4-39#	4-43	4-43#	4-45	4-45#	4-48	4-48#
	4-50	4-50#	4-57	4-57#	4-59	4-59#	4-67	4-67#	4-69	4-69#	4-72	4-72#	4-74	4-74#
	4-78	4-78#	5-4	5-4#	5-13	5-13#	5-14	5-14#	5-14#	5-20	5-20#	5-25	5-25#	5-27
	5-27#	6-3	6-3#	6-5	6-5#	6-47	6-47#	6-56	6-56#	6-92	6-92#	6-321	6-321#	7-2
	7-2#	7-3	7-3#	7-7	7-7#	7-11	7-11#	7-12	7-12#	7-16	7-16#	7-19	7-19#	7-20
	7-20#	7-22	7-22#	8-3	8-3#	2 -1	27-1#	27-2	27-2#	27-191	27-191#	27-194	27-194#	27-195
	27-195#	27-207	27-207#	27-219	27-219#									
MSGENB	6-47	6-47#	6-56	6-56#	6-92	6-92#								
MSGETS	1-30	1-30#	1-91	1-91#	3-81	3-81#	4-33	4-33#	4-43	4-43#	4-48	4-48#	4-57	4-57#
	4-67	4-67#	4-72	4-72#	4-78	4-78#	4-115	4-115#	5-10	5-10#	5-20	5-20#	5-22	5-22#
	5-29	5-29#	6-321	6-321#	6-322	6-322#	7-7	7-7#	7-8	7-8#	7-16	7-16#	7-17	7-17#
MSGNGB	7-22	7-22#	7-23	7-23#	26-50	26-50#	27-191	27-191#	27-192	27-192#	27-207	27-207#	27-217	27-217#
	1-28	1-28#	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	2-4	2-4#	4-2	4-2#	4-37	4-37#	4-39	4-39#	4-45	4-45#	4-50	4-50#	4-59	4-59#
	4-69	4-69#	4-74	4-74#	5-4	5-4#	5-13	5-13#	5-14	5-14#	5-25	5-25#	5-27	5-27#
	5-27#	6-3	6-3#	6-5	6-5#	7-2	7-2#	7-3	7-3#	7-11	7-11#	7-12	7-12#	7-19
	7-19#	7-20	7-20#	8-3	8-3#	27-1	27-1#	27-194	27-194#	27-195	27-195#	27-219	27-219#	
MSGNIN	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29	1-29
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#	1-29#
	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-41	4-41#
	4-41#	4-43	4-43#	4-46	4-46#	4-46	4-46	4-46#	4-46	4-46#	4-46	4-46#	4-46	4-46#
	4-46#	4-46#	4-46#	4-48	4-48#	4-51	4-51	4-51#	4-51	4-51#	4-51	4-51#	4-51	4-51#
	4-51#	4-51#	4-51#	4-51#	4-51#	4-53	4-53	4-53#	4-53	4-53#	4-53	4-53#	4-53	4-53#
	4-53#	4-53#	4-53#	4-53#	4-53#	4-55	4-55	4-55#	4-55	4-55#	4-55	4-55#	4-55	4-55#
	4-55#	4-55#	4-55#	4-55#	4-55#	4-57	4-57	4-57#	4-61	4-61#	4-61	4-61#	4-61	4-61#
	4-61	4-61	4-61#	4-61#	4-61#	4-61#	4-61#	4-61#	4-63	4-63#	4-63	4-63#	4-63	4-63#
	4-63	4-63	4-63#	4-63#	4-63#	4-63#	4-63#	4-63#	4-63#	4-64	4-64#	4-64	4-64#	4-64
	4-64	4-64	4-64	4-64	4-64	4-64	4-64	4-64#	4-64#	4-64#	4-64#	4-64#	4-65	4-65

	4-65	4-65	4-65	4-65	4-65#	4-65#	4-65#	4-65#	4-65#	4-65#	4-67	4-67#	4-70	4-70	4-70
	4-70	4-70	4-70	4-70	4-70#	4-70#	4-70#	4-70#	4-70#	4-70#	4-70#	4-70#	4-72	4-72#	4-75
	4-75	4-75	4-75	4-75	4-75	4-75	4-75	4-75	4-75	4-75	4-75#	4-75#	4-75#	4-75#	4-77
	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77#	4-77#	4-77#	4-77#	4-78
	5-14	5-14#	5-27	5-27	5-27#	5-27#	6-7	6-7	6-7#	6-7#	6-7#	6-7#	6-12	6-12	6-12
	6-12#	6-12#	6-12#	6-12#	6-12#	6-18	6-18	6-18	6-18	6-18	6-18#	6-18#	6-18#	6-18#	6-18#
	6-27	6-27	6-27#	6-27#	6-27#	6-41	6-41	6-41	6-41	6-41	6-41#	6-41#	6-45	6-45	6-45
	6-45	6-45	6-45#	6-45#	6-45#	6-47	6-47	6-47	6-47	6-47	6-47#	6-47#	6-47	6-47#	6-47#
	6-47#	6-47#	6-54	6-54	6-54	6-54	6-54	6-54	6-54	6-54	6-54#	6-54#	6-56	6-56	6-56
	6-56	6-56	6-56	6-56#	6-56#	6-56#	6-56#	6-56#	6-56#	6-56#	6-56#	6-56#	6-69	6-69	6-69
	6-71	6-71	6-71	6-71	6-71#	6-71#	6-71#	6-71#	6-71#	6-71#	6-71#	6-71#	6-91	6-91	6-91
	6-91#	6-91#	6-91#	6-92	6-92	6-92	6-92	6-92	6-92	6-92	6-92#	6-92#	6-92#	6-92#	6-92#
	6-115	6-115	6-115	6-115#	6-115#	6-115#	6-115#	6-115#	6-115#	6-115#	6-130	6-130	6-130	6-130	6-130#
	6-130#	6-130#	6-130#	6-144	6-144	6-144	6-144	6-144	6-144	6-144#	6-144#	6-144#	6-144#	6-144#	6-159
	6-159	6-159	6-159#	6-159#	6-159#	6-159#	6-159#	6-159#	6-159#	6-159#	6-182	6-182	6-182	6-182	6-182#
	6-182#	6-182#	6-205	6-205	6-205	6-205	6-205	6-205	6-205	6-205#	6-205#	6-205#	6-205#	6-205#	6-229
	6-229	6-229#	6-229#	6-229#	6-229#	6-229#	6-229#	6-229#	6-229#	6-229#	6-252	6-252	6-252	6-252	6-252#
	6-252#	6-276	6-276	6-276	6-276	6-276	6-276	6-276	6-276	6-276#	6-276#	6-276#	6-299	6-299	6-299
	6-299#	6-299#	6-299#	6-299#	6-299#	6-311	6-311	6-311	6-311	6-311#	6-311#	6-311#	6-311#	6-311#	6-311#
	6-314	6-314	6-314#	6-314#	6-318	6-318#	6-321	6-321	6-321#	7-7	7-7#	7-16	7-16#	7-22	7-22#
	8-15	8-15#	8-16	8-16#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17#	8-22
	8-22	8-22	8-22	8-22	8-22	8-22	8-22	8-22	8-22#	8-31	8-31#	8-32	8-32#	8-34	8-34
	8-34	8-34	8-34	8-34	8-34	8-34	8-34	8-34	8-38	8-38	8-38	8-38	8-38	8-38	8-38
	8-38	8-38#	10-45	10-45	10-45	10-45	10-45#	10-45#	10-45#	10-45#	10-45#	10-45#	10-48	10-48	10-48
	10-48	10-48#	10-48#	10-48#	10-48#	10-48#	12-54	12-54	12-54	12-54	12-54	12-54#	12-54#	12-54#	12-54#
	12-54#	12-57	12-57	12-57	12-57	12-57#	12-57#	12-57#	12-57#	12-57#	12-57#	12-57#	13-4	13-4	13-4
	13-4	13-4	13-4	13-4	13-4	13-4#	13-4#	13-4#	13-4#	13-4#	13-4#	13-4#	13-4#	13-4#	13-6
	13-6	13-6	13-6	13-6	13-6	13-6	13-6	13-6	13-6	13-6#	13-6#	13-6#	13-6#	13-6#	13-6#
	13-6#	13-6#	13-17	13-17	13-17	13-17	13-17	13-17	13-17	13-17#	13-17#	13-17#	13-17#	13-17#	13-17#
	16-39	16-39	16-39	16-39#	16-39#	16-39#	16-39#	16-39#	16-39#	16-46	16-46	16-46	16-46	16-46	16-46
	16-46	16-46	16-46#	16-46#	16-46#	16-46#	16-46#	16-46#	16-46#	16-46#	16-53	16-53	16-53	16-53	16-53
	16-53	16-53#	16-53#	16-53#	16-53#	16-53#	17-35	17-35	17-35	17-35	17-35	17-35#	17-35#	17-35#	17-35#
	17-35#	17-44	17-44	17-44	17-44	17-44	17-44	17-44	17-44	17-44	17-44#	17-44#	17-44#	17-44#	17-44#
	17-44#	17-44#	17-50	17-50	17-50	17-50	17-50	17-50	17-50	17-50#	17-50#	17-50#	17-50#	17-50#	17-50#
	18-41	18-41	18-41	18-41#	18-41#	18-41#	18-41#	18-41#	18-41#	18-44	18-44	18-44	18-44	18-44	18-44
	18-44#	18-44#	18-44#	20-40	20-40	20-40	20-40	20-40	20-40#	20-40#	20-40#	20-40#	20-40#	20-40#	20-42
	20-42	20-42	20-42#	20-42#	20-42#	20-42#	20-42#	20-42#	21-35	21-35	21-35	21-35	21-35	21-35#	21-35#
	21-35#	21-35#	21-36	21-36#	24-70	24-70	24-70	24-70	24-70	24-70	24-70	24-70	24-70	24-70	24-70#
	24-70#	24-70#	24-70#	24-70#	24-70#	24-70#	24-70#	24-70#	24-74	24-74	24-74	24-74	24-74	24-74	24-74#
	24-74#	24-74#	24-75	24-75#	25-5	25-5	25-5	25-5	25-5	25-5	25-5	25-5	25-5	25-5#	25-5#
	25-5#	25-5#	25-5#	25-5#	25-22	25-22	25-22	25-22	25-22	25-22	25-22	25-22	25-22	25-22#	25-22#
	25-22#	25-22#	25-22#	25-22#	26-31	26-31	26-31	26-31	26-31	26-31#	26-31#	26-31#	26-31#	26-31#	26-31#
	27-186	27-186	27-186	27-186	27-186#	27-186#	27-186#	27-186#	27-186#	27-191	27-191#	27-195	27-195#	27-197	27-186
	27-197	27-197	27-197#	27-199	27-199	27-199	27-199	27-199	27-199#	27-201	27-201	27-201	27-201	27-201	27-197
	27-203	27-203	27-203	27-203#	27-205	27-205	27-205	27-205	27-205	27-205	27-205#	27-207	27-207#	27-201	27-201#
	27-219	27-219#	27-219#	27-219#										27-219	27-219
MSGNLS	6-47	6-47#	6-56	6-56#	6-92	6-92#									
MSGNTA	4-43	4-43#	4-48	4-48#	4-57	4-57#	4-67	4-67#	4-72	4-72#	4-78	4-78#	4-78#	5-20	5-20#
	6-321	6-321#	7-7	7-7#	7-16	7-16#	7-22	7-22#	27-191	27-191#	27-207	27-207#			
MSGNTE	27-2	27-2#													
MSHAPT	1-29	1-29#													
MSHNAP	1-29	1-29#													
MSINCR	1-28	1-28#	1-40	1-40#	2-4	2-4#	4-2	4-2#	4-37	4-37#	4-39	4-39	4-39#	4-39#	4-39#
	4-41#	4-43#	4-45	4-45#	4-45#	4-45#	4-46#	4-48#	4-50	4-50	4-50#	4-50#	4-51#	4-53#	4-53#
	4-55#	4-57#	4-59	4-59#	4-59#	4-59#	4-61#	4-63#	4-64#	4-65#	4-67#	4-69	4-69	4-69#	4-69#
	4-69#	4-70#	4-72#	4-74	4-74	4-74#	4-74#	4-75#	4-77#	4-78#	5-4	5-4	5-4#	5-4#	5-4#

	5-13	5-13#	5-14	5-14	5-14#	5-14#	5-25	5-25#	6-3	6-3#	6-5	6-5	6-5#	6-5#
	6-7#	6-12#	6-18#	6-27#	6-41#	6-45#	6-47	6-47#	6-47#	6-54#	6-56	6-56#	6-56#	6-69#
	6-71#	6-91#	6-92	6-92#	6-92#	6-115#	6-130#	6-144#	6-159#	6-182#	6-205#	6-229#	6-252#	6-276#
	6-299#	6-311#	6-314#	6-318#	6-321#	7-2	7-2#	7-3	7-3	7-3#	7-3#	7-7#	7-11	7-11#
	7-12	7-12	7-12#	7-12#	7-16#	7-19	7-19#	7-20	7-20	7-20#	7-20#	7-22#	8-3	8-3#
	8-15#	8-31#	10-45#	10-48#	12-54#	12-57#	13-4#	13-6#	13-17#	16-39#	16-46#	16-53#	17-35#	17-44#
	17-50#	18-41#	18-44#	20-40#	20-42#	21-35#	21-36#	24-70#	24-74#	24-75#	25-5#	25-22#	26-31#	27-1
	27-1#	27-2	27-2	27-2	27-2#	27-2#	27-2#	27-186#	27-191#	27-194	27-194#	27-195	27-195	27-195#
	27-195#													
MSLDRO	6-7	6-7#	6-27	6-27#	6-41	6-41#	6-69	6-69#	6-314	6-314#				
MSMCHI	1-22	1-22#												
MSMCLO	1-22	1-22#												
MSPOP	1-30	1-30#	1-91	1-91#	3-81	3-81#	4-33	4-33#	4-43	4-43#	4-48	4-48#	4-57	4-57#
	4-67	4-67#	4-72	4-72#	4-78	4-78#	4-115	4-115#	5-10	5-10#	5-20	5-20#	5-22	5-22#
	5-29	5-29#	6-321	6-321#	6-322	6-322#	7-7	7-7#	7-8	7-8#	7-16	7-16#	7-17	7-17#
MSPRIN	7-22	7-22#	7-23	7-23#	26-50	26-50#	27-191	27-191#	27-192	27-192#	27-207	27-207#	27-217	27-217#
	4-41	4-41#	4-46	4-46#	4-51	4-51#	4-53	4-53#	4-55	4-55#	4-61	4-61#	4-63	4-63#
	4-64	4-64#	4-65	4-65#	4-70	4-70#	4-75	4-75#	4-77	4-77#	6-45	6-45#	6-54	6-54#
	6-71	6-71#	6-91	6-91#	13-4	13-4#	13-6	13-6#	13-17	13-17#	16-46	16-46#	16-53	16-53#
	17-44	17-44#	17-50	17-50#	21-35	21-35#	24-70	24-70#	24-74	24-74#	25-5	25-5#	25-22	25-22#
	27-186	27-186#												
MSPUSH	1-28	1-28#	1-40	1-40#	2-4	2-4#	4-2	4-2#	4-37	4-37#	4-39	4-39#	4-45	4-45#
	4-50	4-50#	4-59	4-59#	4-69	4-69#	4-74	4-74#	5-4	5-4#	5-13	5-13#	5-14	5-14#
	5-25	5-25#	6-3	6-3#	6-5	6-5#	7-2	7-2#	7-3	7-3#	7-11	7-11#	7-12	7-12#
	7-19	7-19#	7-20	7-20#	8-3	8-3#	27-1	27-1#	27-2	27-2#	27-194	27-194#	27-195	27-195#
MSPUT	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-46	4-46#	4-46	4-46#	4-46#	4-51	4-51#	4-51
	4-51	4-51#	4-53	4-53#	4-53	4-53#	4-53	4-53#	4-55	4-55#	4-55	4-55	4-55#	4-61
	4-61	4-61#	4-61	4-61#	4-63	4-63#	4-63	4-63#	4-63	4-63#	4-64	4-64#	4-64	4-64#
	4-64	4-64#	4-64#	4-65	4-65	4-65#	4-70	4-70#	4-70	4-70#	4-70	4-70#	4-75	4-75#
	4-75	4-75#	4-75#	4-77	4-77	4-77#	4-77	4-77#	4-77	4-77#	6-45	6-45#	6-54	6-54#
	6-71	6-71#	6-71#	6-91	6-91	6-91#	13-4	13-4#	13-4	13-4#	13-4	13-4#	13-4#	13-6
	13-6	13-6#	13-6	13-6#	13-6	13-6#	13-17	13-17#	13-17	13-17#	16-46	16-46#	16-46	16-46#
	16-46	16-46#	16-53	16-53#	16-53	16-53#	17-44	17-44#	17-44	17-44#	17-44	17-44#	17-50	17-50#
	17-50	17-50#	21-35	21-35#	21-35	21-35#	24-70	24-70#	24-70	24-70#	24-70	24-70#	24-74	24-74#
	24-74#	25-5	25-5	25-5#	25-5	25-5#	25-22	25-22#	25-22	25-22#	25-22	25-22#	27-186	27-186#
MSPUT1	4-41	4-41#	4-41	4-41#	4-41	4-41#	4-41#	4-41#	4-41#	4-41#	4-46	4-46#	4-46	4-46#
	4-46#	4-46#	4-46#	4-46#	4-51	4-51#	4-51	4-51#	4-51#	4-51#	4-51#	4-51#	4-53	4-53#
	4-53	4-53#	4-53	4-53#	4-53#	4-53#	4-53#	4-53#	4-55	4-55#	4-55	4-55#	4-55#	4-55#
	4-55#	4-55#	4-61	4-61#	4-61	4-61#	4-61#	4-61#	4-61#	4-61#	4-63	4-63#	4-63	4-63#
	4-63	4-63#	4-63#	4-63#	4-63#	4-63#	4-64	4-64#	4-64	4-64#	4-64	4-64#	4-64#	4-64#
	4-64#	4-64#	4-64#	4-64#	4-65	4-65#	4-65	4-65#	4-65#	4-65#	4-70	4-70#	4-70	4-70#
	4-70#	4-70#	4-70#	4-70#	4-75	4-75#	4-75	4-75#	4-75	4-75#	4-75#	4-75#	4-77	4-77#
	4-77	4-77#	4-77#	4-77#	4-77#	4-77#	6-45	6-45#	6-45	6-45#	6-45#	6-54	6-54#	6-54#
	6-71	6-71#	6-71#	6-71#	6-91	6-91#	6-91	6-91#	13-4	13-4#	13-4	13-4#	13-4	13-4#
	13-4#	13-4#	13-4#	13-4#	13-4#	13-4#	13-6	13-6#	13-6	13-6#	13-6	13-6#	13-6#	13-6#
	13-6#	13-6#	13-6#	13-6#	13-17	13-17#	13-17	13-17#	13-17#	13-17#	16-46	16-46#	16-46	16-46#
	16-46	16-46#	16-46#	16-46#	16-46#	16-46#	16-53	16-53#	16-53	16-53#	16-53#	16-53#	17-44	17-44#
	17-44	17-44#	17-44	17-44#	17-44#	17-44#	17-44#	17-44#	17-50	17-50#	17-50	17-50#	17-50#	17-50#
	21-35	21-35#	21-35#	21-35#	24-70	24-70#	24-70	24-70#	24-70	24-70#	24-70	24-70#	24-70#	24-70#
	24-70#	24-70#	24-74	24-74#	24-74	24-74#	25-5	25-5#	25-5	25-5#	25-5	25-5#	25-5#	25-5#
	25-22	25-22#	25-22	25-22#	25-22	25-22#	25-22#	25-22#	27-186	27-186#	27-186	27-186#	27-186#	27-203
MSRADI	6-47	6-47#	6-56	6-56#	6-92	6-92#	27-197	27-197#	27-199	27-199#	27-201	27-201#	27-201#	27-203#
	27-205	27-205#												
MSRNRO	6-27	6-27#	6-41	6-41#										
MSSETS	1-28	1-28#	1-40	1-40#	2-4	2-4#	4-2	4-2#	4-37	4-37#	4-39	4-39#	4-45	4-45#
	4-50	4-50#	4-59	4-59#	4-69	4-69#	4-74	4-74#	5-4	5-4#	5-13	5-13#	5-14	5-14#

