Micro Fiche Scan

Name of device(s) tested:
    RL01/02,RL11,RLV11/12
Test description:
    RL01/02 DRIVE TST 3
MAINDEC Number or Package Identifier (after SEP 1977):
    CZRLNC0
Fiche Document Part Number:
    AH-F845C-MC
Fiche preparation date unknown, using copyright year:
    1985
Image resolution:
    8-bit gray levels, max. quality for archiving

```
.REM    @
```

IDENTIFICATION

PRODUCT CODE: AC-F843C-MC
PRODUCT NAME: CZRLNC0 RL01/02 DRIVE TEST 3
DATE CREATED: 05-JAN-1979
REVISED: 06-JAN-1986
MAINTAINER: CXO DIAGNOSTIC ENGINEERING
AUTHORS: D. DEKNIS, C. CAMPBELL
REVISED BY: M. LEAVITT

HISTORY

AUTHOR: DAN DEKNIS 05-JAN-1979 VERSION A0

MODIFIED BY:

CHUCK CAMPBELL 1983 VERSION B0
MIKE LEAVITT 06-JAN-1986 VERSION C0

B0 Problem:

Unknown

Solution:

Unknown

C0 Problem:

Prism Report PR00486. Diagnostic will not read Bad Sector File if TEST 2 is not included in test sequence.

Solution:

All tests in the diagnostic which require the bad sec file data, will test to see if the bad sec file had previously been read. If not, the test will read the bad sector file before executing the desired test sequence.

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

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

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

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

1.1.2 DIAGNOSTIC INFORMATION

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

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

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

1.1.3 DIAGNOSTIC RUN TIME

THIS DIAGNOSTIC TAKES 4 MINUTES TO RUN THE FIRST PASS AND 28.5 MINUTES FOR THE SECOND PASS.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

1.2.2 SOFTWARE REQUIREMENTS

CZRLJ RL01/02 DRIVE TEST PART 2

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

CVRLA	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLG	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CZRLH	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CZRLI	RL01/02 DRIVE TEST (PART 1)

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

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

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

TYPE "R" AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

****
* STEP 1 *
****

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

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

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

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

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

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

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

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

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

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

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

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

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

EACH UNIT TO BE TESTED.

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

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

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

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

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

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

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

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

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

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

   NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF

NO ERROR HAD OCCURRED.

2.1.2 SAMPLE RUN-THROUGH

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

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

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

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

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

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

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

PRO/FLAGS:IER:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG

2. TURN ON THE LOE FLAG

3. TURN OFF THE HOE FLAG

4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED:
.R CZRLN??	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. x mmm-yy	D
CZRLN-?-?	D
CZRLN TESTS SEEK AND ROTATIONAL TIMING & WRITE & READ DATA	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D,O
# UNITS (D) ? 2	D,O
UNIT 0	D
RL11 (L) Y ?	D,O

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

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

                                                CHANGE SW (L) ? Y                           D,O

                                                USE ALL CYL (L) N ?                         D,O
                                                USE ALL SECT (L) N ?                        D,O
                                                DO MANUAL INTERVENTION TEST (L) N ?         D,O
                                                LOW SEEK LIMIT (L) N ?                      D,O
                                                UPPER SEEK LIMIT (L) N ?                    D,O
                                                USE ONLY ONE SURF (L) N ?                   D,O
                                                INPUT ERROR LIMIT (D) 20 ?                  D,O
                                                DATA CMP ERR LMT (D) 10 ?                   D,O
                                                PRINT ERRORS DETECTED WHILE READING BAD SEC FILE (N) ? D,O


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

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

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

                                                ^C                                          O

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

                                                CHANGE SW (L) ? N                           D,O

                                                CZRLN EOP 1                                 D
                                                ^C

                                                DR>RESTART/PASS:1                           D,O

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

2.2 CHAIN MODE OPERATION

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

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

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

TO EXECUTE A CHAIN FILE THE USER TYPES:

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

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

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

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

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

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

2.3.2 COMMAND SYNTAX

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

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS

ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C, AFTER THE OPERATOR RESPONDS TO "# UNITS?" THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

ADR EXECUTE AUTODROP CODE

LOT LOOP ON TEST

EVL EVALUATE

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

****
EXIT
****

RETURN TO XXDP+ PROMPT MODE.

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

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

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

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

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

****
PRI(NT)

****

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

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

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

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

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

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

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

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

```
# UNITS (D) ? 8

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

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

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

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

2.5 HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

CHANGE S.W. ?

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

USE ALL CYLINDERS (N)?

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

USE ALL SECTORS (N)?

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

EXECUTE MANUAL INTERVENTION TESTS (N)?

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

LOWER SEEK LIMIT (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

ENTER VALUE (DECIMAL) (0)?

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

UPPER SEEK LIMIT (N)?

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

ENTER VALUE (DECIMAL) (255)?

USE ONLY ONE SURFACE (N)?

IF "YES", THE NEXT PARAMETER IS REQUESTED.

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

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

SPECIFY ERROR LIMIT (DECIMAL) (20)?

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

DATA COMPARE ERROR LIMIT (DECIMAL) (20)?

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

TOTAL IS REPORTED AT THE END OF THE COMPARE.

PRINT ERRORS DETECTED WHILE READING BAD SEC FILE (N)?

IF "YES", ALL ERRORS DETECTED WHILE READING THE BAD SECTOR FILE, WILL BE PRINTED TO THE OUTPUT DEVICE. IF "NO", ONLY HARD ERRORS WILL BE PRINTED TO THE OUTPUT DEVICE. THIS IS USEFUL IF THE USER WISHES TO SEE WHAT ERRORS ARE DETECTED IN ANY BAD COPIES OF THE BAD SECTOR FILES.

3.0 ERROR INFORMATION

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

3.1 ERROR REPORTING

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

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

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

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

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

READ HEADER - READ HEADER FOR 40 HEADERS - READ HEADER FOR 40 HEADERS WITH HEADER COMPARE - HAVE THE SAME GENERAL MEANING AS THE READ DATA AND READ DATA WITH DATA COMPARE. MESSAGES HAVING THE OPERATION OF READ HEADER OR READ HEADER FOR 40 HEADERS ARE THE RESULT OF ERRORS DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER

COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

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

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

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

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

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

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

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

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

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

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

3.1.2 SPECIFIC RESULT MESSAGES

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

THE GENERAL FORMAT FOR THE RESULT LINE IS:

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

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTANT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUSH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADER OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HDR CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
HDR NOT FND/DAT LATE	(HDR NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT THE RESULT ACTUALLY IS. THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT

CONDITIONS, A NUMBER 0 TO 7 TO INDICATE THE DRIVE STATE, OR A NUMBER 0 TO 377 (OCTAL) TO IDENTIFY A CYLINDER NUMBER.

VARIABLE 3 DEFINES THAT THE VALUE GIVEN IS VARIABLE 2 SHOULD BE. THE OPTIONAL QUALIFIER IS PROVIDED WHEN IT IS USEFUL TO KNOW WHEN THE ERROR WAS DETECTED IN THE OPERATION BEING PERFORMED. THIS QUALIFIER IS USED TO REPORT RESULTS SUCH AS:

```
BRUSH HME IS 1 SB 0 IN STATE 2
HEADS OUT IS 0 SB 1 IN STATE 3
DRV RDY   IS 0 SB 1 IN DATA XFER
SELECTED HEAD IS 1 SB 0 IN CYCLE UP
DRV RDY   IS 0 SB 1 IN STATE 5
DRV RDY   IS 1 SB 0 IN SEEK W/O MOTION
DRV RDY   IS 0 SB 1 IN 10MS
DRV RDY   IS 0 SB 1 IN 500MS
DRV RDY   IS 0 SB 1 IN 5SECONDS
```

THESE RESULTS, WHEN SEEN WITH THE OPERATION MESSAGE, WILL BE SELF EXPLANATORY.

OTHER RESULT MESSAGES THAT CAN BE PART OF AN ERROR REPORT ARE:

"INTERRUPT TOO LATE"

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

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"

THIS IS REPORTED WHEN AN ERROR CAUSES HEADS TO UNLOAD AND AFTER THE ERROR IS CLEARED THE HEADS DO NOT RELOAD.

"UNKN DRV STATE-NO RDY, NO ERR, HDS OUT"

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

"WRITE ABORTED"

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

"COULD NOT RETRIEVE DRIVE STATUS"

THIS IS REPORTED IF THE GET STATUS COMMAND DOES NOT COMPLETE SUCCESSFULLY WHEN THE STATUS IS REQUIRED TO REPORT AN ERROR.

"OPI SET-NO DRIVE RESPONSE"

THIS IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (OPI SETS) WHEN THAT COMMAND IS BEING USED IN THE

EARLY TESTS TO CHECK THE DRIVE INTERFACE.

"NO INTERRUPT ON CMND COMPLETE"

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

"ERR DID NOT CLEAR"

THIS IS REPORTED WHEN THE RESET COMMAND DOES NOT CLEAR THE CONTROLLER ERRORS. THIS IS A CONTROLER RELATED PROBLEM BUT IS REPORTED IF SEEN IN THE DRIVE TEST PROGRAMS.

"DRV ERR IS NOT CLEARED"

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

"UNEXPECTED ERR"

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

"BAD SEC FILE FMT ERR"

THIS IS REPORTED IF THE CONTENTS OF THE FILES DO NO CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

"*WARNING* ALL SECTORS ASSUMED GOOD FOR TESTS REQUIRING BAD SEC DATA"

THIS MESSAGE IS PRINTED WHEN THE BAD SECTOR FILES COULD NOT BE READ OR IF DATA READ IS CORRUPT. THIS WARNING IS TO PRINTED TO LET THE USER KNOW THAT ANY ERRORS COULD BE A RESULT OF TESTING A KNOWN BAD SECTOR.

"ERROR LIMIT EXCEEDED-UNIT DROPPED"

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

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

```
(1)   PROG NAME   ERR NUM   TEST NUM   SUBTEST NUM   ERR PC
(2)   ROUTINE TRACE SEQ (IN SEQ CALLED)
         (ADDRESS)
```

```
                        (ADDRESS)
                            .
                            .
                        (ADDRESS)
                (3)   TEST DESCRIPTION
                (4)   OPERATION:
                (5)   RESULT:
                (6)   ADDRESS OF UNIT UNDER TEST
                (7)                RLCS    RLDA    RLBA    RLMP    CYL     HD
                (8)   OP INIT
                (9)   OP DONE
                (10)       DRIVE STATUS
                (11)              WORD NUM IS (XXXXXX) SB (YYYYYY)
                (12)  TOTAL COMPARE ERRS: (ZZZ) OF (128)

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

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

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

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

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

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

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

ARE BEING READ FROM A TRACK.

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

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

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

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

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

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

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

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

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

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

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

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR

BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)
BIT 9 - VOLUME CHECK (VC)
BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS RL02 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 -STATE BITS
0 - LOAD STATE
1 - SPIN UP
2 - BRUSH CYCLE
3 - LOAD HEADS
4 - SEEK - TRACK COUNTING
5 - SEEK - LINEAR MODE
6 - UNLOAD HEADS
7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 SEEK TIMING

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

POSITION HEADS AT CYLINDER 0.

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

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

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

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

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

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

MID CYL FWD	X	X	X
MID CYL REV	X	X	X
MAX CYL FWD		X	X
MAX CYL REV		X	X

THE X INDICATES WHERE TIME WILL BE REPORTED.

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

POSITION HEADS AT MAX CYLINDER (BAD SEC FILE).

DO READ DATA ON 1ST COPY OF THE FACTORY BAD SEC FILE (SECTORS 0 & 1, HEAD 1). IF AN ERROR IS DETECTED, PROCEED BY READING THE NEXT COPY OF THE FACTORY BAD SEC FILE UNTIL A GOOD COPY IS FOUND (SECTORS 4 & 5, 8 & 9, 12 & 13, 16 & 17). REPORT ALL ERRORS IF BAD SEC FILE ERROR REPORTING IS ON (SEE SW QUESTIONS), BUT DO NOT INCREMENT ERROR COUNT. IF NO GOOD COPIES CAN BE FOUND, REPORT THAT FACTORY BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READING FIELD BAD SEC FILE AT SECTOR 20.

DO READ DATA ON 1ST COPY OF THE FIELD BAD SEC FILE (SECTORS 20 & 21, HEAD 1). IF AN ERROR IS DETECTED, PROCEED BY READING THE NEXT COPY OF THE FIELD BAD SEC FILE UNTIL A GOOD COPY IS FOUND (SECTORS 24 & 25, 28 & 29, 32 & 33, 36 & 37). REPORT ALL ERRORS IF BAD SEC FILE ERROR REPORTING IS ON (SEE SW QUESTIONS), BUT DO NOT INCREMENT ERROR COUNT. IF NO GOOD COPIES CAN BE FOUND, REPORT THAT FIELD BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT.

UPON FINDING A GOOD COPY OF THE BAD SEC FILE, VERIFY DATA FORMAT (WORD 0 & 1 ARE NOT 0 & NOT NEGATIVE, WORD 2 & 3 ARE 0, WORD 254 & 255 ARE ALL ONE'S, LOCATE 1ST WORD OF ALL ONE'S & MAKE SURE THAT ALL REMAIN WORDS TO WORD 255 ARE ALL 1'S) STORE BAD SECTOR DATA.

NOTE: IF HEAD 1 IS DESELECTED VIA THE SW QUESTIONS, THIS TEST WILL BE BYPASSED AND A MESSAGE PRINTED TO THAT AFFECT.

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

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

POSITION HEADS AT CYLINDER 0.

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

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

REPEAT FOR OTHER DATA PATTERNS (2 THROUGH 8).

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

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

TEST 4 ROTATIONAL TIMING TEST
******

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

POSITION HEADS TO CYLINDER 0.

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

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

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

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

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

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

READ/COMPARE ALL DATA.

REPEAT ABOVE TEST FOR HEAD 1.

REPEAT ABOVE TESTS FOR ALL CYLINDERS IN SELECTED CYLINDER SET.

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

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

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

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

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

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

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

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

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

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

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

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

OPPOSITE OF CENTER CYLINDER.)

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

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

TEST 8 OVERWRITE TEST
******

READ THE BAD SECTOR FILE IF NOT ALREADY READ.

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

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

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

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

SEEK FORWARD TO CC. WRITE PATTERN B. SEEK FORWARD TO "HILIMIT" SEEK REVERSE TO CC. READ/COMPARE DATA.

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

REPEAT ABOVE TESTS ON HEAD 1.

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

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

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

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

@

@

K3


```
                000001                          PART2==1
                                        .ENABL  LC,AMA,ABS
000000                                  .NLIST  MC,BEX,TOC
                002000                          .=2000
                                                .MCALL  SVC

                000000                          SVCTST=0
                000000                          SVCSUB=0
                000001                          SVCBGL=1
                000000                          SVCINS=0
                000000                          SVCTAG=0
```

```
                        .SBTTL  MACRO DEFINITIONS

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

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

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

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

                        .MACRO  STCLK                   ;MACRO START P-CLOCK
                                CLR     @#CLKCSB        ;CLEAR CLOCK COUNT SET BUFFER
                                CLR     @#CLKCTR        ;CLEAR CLOCK COUNTER
                                MOV     #23,@#CLKCSR    ;INITIALIZE CLOCK FOR COUNT-UP MODE,
                                                        ;/10 KHZ RATE, AND START CLOCK
                        .ENDM
```

M3


```
                                .NLIST  CND,MD,ME

                                .ASCII /C/
                                .ASCII /Z/
                                .ASCII /R/
                                .ASCII /L/
                                .ASCII /N/
                                .BYTE   0
                                .BYTE   0
                                .BYTE   0
                                .ASCII  /C/
                                .ASCII  /0/
                                .WORD   0
                                .WORD   30000
                                .WORD   L$HARD
                                .WORD   L$SOFT
                                .WORD   L$HW
                                .WORD   L$SW
                                .WORD   L$LAST
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   L$DISPATCH
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .BYTE   C$REVISION
                                .BYTE   C$EDIT
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   L$DVTYP
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   L$DU
                                .WORD   0
                                .WORD   L$DESC
                                EMT     E$LOAD
                                .WORD   0
                                .WORD   L$INIT
                                .WORD   L$CLEAN
                                .WORD   L$AUTO
                                .WORD   L$PROT
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .ASCIZ  /CZRLN TESTS SEEK, ROTATIONAL TIMING AND WRITE & READ DATA/
                                .EVEN
                                .ASCIZ  %RL01,RL02%
                                .EVEN


                        .SBTTL  GLOBAL EQUATE SECTION
```

```
                        ;
                        ; BIT DIFINITIONS
                        ;
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
                        ;
                        ;                                       ; BIT POSITION IN SECOND STATUS WORD
000040                  EF.START==      32.                     ; (100000) START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; (040000) RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; (020000) CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; (010000) A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; (004000) A POWER-FAIL/POWER-UP OCCURRED
000033                  EF.XM==         27.                     ; (002000) Diag is good of extended enviroment
                        ;
                        ;
                        ; 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
 16                             ;       OFFSETS FOR HARDWARE P-TABLE
 17             000000          CSR     =0                      ;BUS ADDRESS
 18             000002          VECT    =2                      ;VECTOR ADDRESS
 19             000004          PRIOR   =4                      ;PRIORITY
 20             000006          TYPDR   =6                      ;DRIVE TYPE
 21             000010          DRSB    =10                     ;DRIVE SELECT BIT
 22             000012          CNT     =12                     ;CONTROLLER TYPE
 23
 24                             ;       OFFSET FOR SOFTWARE P-TABLE
 25             000000          MISWI   =0                      ;SOFTWARE PARAMETERS SWITCHES
 26             000002          LOLIM   =2                      ;CYLINDER LOWER LIMIT
 27             000004          HILIM   =4                      ;CYLINDER HIGH LIMIT
 28             000006          HEAD    =6                      ;SELECTED HEAD FOR RUNNING TESTS
 29             000010          ERLIM   =10                     ;ERROR LIMIT
 30             000012          DCLIM   =12                     ;DATA COMPARE ERROR LIMIT
 31             000014          BSERR   =14                     ;BAD SEC FILE PRINT ERROR FLAG
 32
 33                             ;       BIT ASSIGNMENT FOR SOFTWARE P-TABLE SWITCHES
 34             000001          ALLCYL  =BIT00                  ;USE ALL CYLINDERS
 35             000002          ALLSEC  =BIT01                  ;USE ALL SECTORS
 36             000004          DRSELT  =BIT02                  ;EXECUTE DRIVE SELECT TEST
 37             000010          HDALIGN =BIT03                  ;EXECUTE HEAD ALIGNMENT TEST
 38             010000          HEADLM  =BIT12                  ;HEAD LIMIT SPECIFIED FLAG
 39             020000          HICYL   =BIT13                  ;HI LIMIT SPECIFIED FLAG
 40             040000          LOCYL   =BIT14                  ;LO LIMIT SPECIFIED
 41             100000          MITEST  =BIT15                  ;EXECUTE MANUAL INTERVENTION TESTS
 42
 43                             ;       SUBSYSTEM FUNCTIONS
 44             000102          CKDATA  =102                    ;WRITE CHECK
 45             000104          GTSTAT  =104                    ;GET STATUS
 46             000106          SEEK    =106                    ;SEEK
 47             000110          RDHEAD  =110                    ;READ HEADER
 48             000112          WTDATA  =112                    ;WRITE DATA
 49             000114          RDDATA  =114                    ;READ DATA
 50             000116          RDNOHR  =116                    ;READ DATA, IGNORE HEADERS
 51             000100          NOOP    =100                    ;NO OPERATION
 52
 53                             ;       OPERATION FLAGS
 54             007777          COMPOP  =7777                   ;COMPOSITE OPERATION FLAGS
 55             000002          HDRCMP  =BIT01                  ;HEADER COMPARE OPERATION
 56             000001          DATACMP =BIT00                  ;DATA COMPARE OPERATION
 57             000004          CYLUP   =BIT02                  ;CYCLE UP OPERATION
 58             000010          ULOAD   =BIT03                  ;UNLOAD OPERATION
```

```
                                ;
                                ;IN-OUT SEEK OPERATION
                                ;OUT-IN SEEK OPERATION
                                ;FOLLOWING WRITE OPERATION
                                ;REV SEEK SEQ (ADJ INTERFERENCE)
                                ;FWD SEEK SEQ (ADJ INTERFERENCE)
                                ;REV SEEK SEQ (OVERWRITE)
                                ;FWD SEEK SEQ (OVERWRITE)
                                ;BAD DISK ADDRESS
                                ;SEEK OPERATION
                                ;READ OR WRITE OPERATION
                                ;RELOAD WAIT
                                ;40 HEADER OPERATION
                                ;MESSAGE QUALIFIER BITS

                                ;ERROR FLAGS FROM SUBROUTINES
                                ;OPERATION TOOK TOO LONG
                                ;NO INTERRUPT FROM OPERATION
                                ;CONTROLLER HUNG
                                ;BAD CONTROLLER CLEAR

                                ;CONTROL AND STATUS REGISTER
                                ;BUS ADDRESS REGISTER
                                ;DISK ADDRESS REGISTER
                                ;MULTI-PURPOSE REGISTER

                                ;REGISTER BIT DEFINITIONS - CONTROL STATUS REGISTER
                                ;CONTROL AND STATUS REGISTER
                                ;ANY ERROR BIT
                                ;DRIVE ERROR BIT
                                ;NON-EXISTENT MEMORY ERROR
                                ;DATA LATE ERROR
                                ;HEADER NOT FOUND ERROR
                                ;DATA CHECK ERROR
                                ;HEADER CHECK ERROR
                                ;OPERATION INCOMPLETE ERROR
                                ;DRIVE SELECT MASK
                                ;CONTROLLER READY MASK
                                ;INTERRUPT ENABLE MASK
                                ;BUS ADDRESS UPPER MASK
                                ;DRIVE READY MASK

                                ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR DATA XFER
                                ;SECTOR ADDRESS MASK
                                ;HEAD SELECT MASK

                                ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR SEEK
                                ;MUST BE SET, BIT 0
                                ;DIRECTION BIT
                                ;HEAD SELECT BIT

                                ;REGISTER BIT DEFINITIONS - DISK ADDRESS FOR GET STATUS
                                ;GET STATUS SETUP
                                ;DRIVE RESET MASK

                                ;REGISTER BIT DEFINITIONS - MP FOR DATA XFER
                                ;WORD COUNT MASK
```

```
                160000          WCRNG   =160000                 ;WORD COUNT RANGE MASK

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

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

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


                                .SBTTL  GLOBAL DATA SECTION


                                ;       TABLE OF OPERATION MESSAGES

002226  000000  OPMSGS: .WORD   0                       ;FILLER
002230  005775         .WORD   MWRCHK                   ;MESSAGE FOR WRITE CHECK
002232  006020         .WORD   MGTSTA                   ;               GET STATUS
002234  005750         .WORD   MSEEK                    ;               SEEK
002236  005765         .WORD   MREADH                   ;               READ HEADER
002240  006006         .WORD   MWRITE                   ;               WRITE DATA
002242  005754         .WORD   MREAD                    ;               READ DATA
002244  006103         .WORD   MWRSET                   ;               WITH RESET
002246  006032         .WORD   MDATCP                   ;               WITH DATA COMPARE
002250  006051         .WORD   MHDRCP                   ;               WITH HEADER COMPARE
002252  006150         .WORD   MCYLUP                   ;               LOAD HEADS
002254  006137         .WORD   MULOAD                   ;               UNLOAD HEADS
002256  006177         .WORD   MINOUT                   ;                  IN-OUT SEQ
002260  006160         .WORD   MOUTIN                   ;                  OUT-IN SEQ
002262  006220         .WORD   MFOLWRT                  ;                  FOLLOWING WRITE
002264  006240         .WORD   MREVSK                   ;                  REV SEEK
002266  006271         .WORD   MFWDSK                   ;                  FWD SEEK
002270  006356         .WORD   MRESKO                   ;                  REV SEEK
002272  006322         .WORD   MFWSKO                   ;                  FWD SEEK
002274  006412         .WORD   MBADAD                   ;                  BAD DISK ADD FOR WRITE
002276  006067         .WORD   M40HDR                   ;               40 HEADER OPERATION

002300  000000  T.DRIVE: .WORD  0
002302  000000  JUNK:    .WORD  0
002304  000000  HLMTW:   .WORD  0
```

```
                                CLRBYT: .WORD   0
                                NXTHL:  .WORD   0
                                GBND:   .WORD   0
                                CAMSK:  .WORD   0
                                DIRMSK: .WORD   0
                                HDCYL:  .WORD   0

                                ;       TABLE OF RESULT NAME MESSAGE ADDRESSES
                                RESTBL: .WORD   MCERR           ;CONTROLLER ERROR
                                        .WORD   MDRERR          ;DRIVE ERROR
                                        .WORD   MNEERR          ;NON-EXISTANT MEMORY ERROR
                                        .WORD   MFLERR          ;HEADER NOT FOUND-DATA LATE
                                        .WORD   MHDERR          ;HEADER OR DATA ERROR
                                        .WORD   MOPERR          ;OPERATION INCOMPLETE
                                        .WORD   MNDRST          ;NO DRIVE STATUS AVAILABLE
                                        .WORD   0
                                        .WORD   MWDERR          ;WRITE DATA ERROR
                                        .WORD   MHCERR          ;HEAD CURRENT ERROR
                                        .WORD   0
                                        .WORD   MSTERR          ;SEEK TIMEOUT ERROR
                                        .WORD   MSPERR          ;SPINDLE ERROR
                                        .WORD   MWGERR          ;WRITE GATE ERROR
                                        .WORD   0
                                        .WORD   MDSERR          ;DRIVE SELECT ERROR

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


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


                                ;RL01 TABLE OF CYLINDERS
                                T25TBL: .WORD   2               ;TABLE OF DIFFERENCES
                                        .WORD   6
                                        .WORD   9.
                                        .WORD   12.
                                        .WORD   17.
                                        .WORD   22.
```

F4


```
 232 002446  000033                        .WORD   27.
 233 002450  000042                        .WORD   34.
 234 002452  000051                        .WORD   41.
 235 002454  000200                        .WORD   128.
 236 002456  000377                        .WORD   255.
 237
 238                                ;RL02 TABLE OF CYLINDERS
 239 002460  000004                T25TB2: .WORD   4
 240 002462  000014                        .WORD   12.
 241 002464  000022                        .WORD   18.
 242 002466  000030                        .WORD   24.
 243 002470  000042                        .WORD   34.
 244 002472  000054                        .WORD   44.
 245 002474  000066                        .WORD   54.
 246 002476  000104                        .WORD   68.
 247 002500  000122                        .WORD   82.
 248 002502  000400                        .WORD   256.
 249 002504  000777                        .WORD   511.
 250
 251                                ;       TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS
 252
 253 002506                         T33TBL: .BLKW   16.
 254 002546                         TBT:    .BLKW   16.
 255
 256
 257 002606      002                CYLTBL: .BYTE   2               ;TABLE OF DEFAULT CYLINDERS
 258 002607      007                        .BYTE   7.
 259 002610      016                        .BYTE   14.
 260 002611      024                        .BYTE   20.
 261 002612      033                        .BYTE   27.
 262 002613      041                        .BYTE   33.
 263 002614      046                        .BYTE   38.
 264 002615      055                        .BYTE   45.
 265 002616      064                        .BYTE   52.
 266 002617      072                        .BYTE   58.
 267 002620      101                        .BYTE   65.
 268 002621      110                        .BYTE   72.
 269 002622      115                        .BYTE   77.
 270 002623      124                        .BYTE   84.
 271 002624      133                        .BYTE   91.
 272 002625      141                        .BYTE   97.
 273 002626      146                        .BYTE   102.
 274 002627      154                        .BYTE   108.
 275 002630      161                        .BYTE   113.
 276 002631      170                        .BYTE   120.
 277 002632      177                        .BYTE   127.
 278 002633      206                        .BYTE   134.
 279 002634      213                        .BYTE   139.
 280 002635      222                        .BYTE   146.
 281 002636      230                        .BYTE   152.
 282 002637      235                        .BYTE   157.
 283 002640      244                        .BYTE   164.
 284 002641      252                        .BYTE   170.
 285 002642      261                        .BYTE   177.
 286 002643      270                        .BYTE   184.
 287 002644      275                        .BYTE   189.
 288 002645      303                        .BYTE   195.
```

```
289 002646     312                          .BYTE   202.
290 002647     317                          .BYTE   207.
291 002650     326                          .BYTE   214.
292 002651     334                          .BYTE   220.
293 002652     343                          .BYTE   227.
294 002653     352                          .BYTE   234.
295 002654     361                          .BYTE   241.
296 002655     367                          .BYTE   247.
297 002656     375                          .BYTE   253.
298 002657     000                          .BYTE   0
299 002660  000401                          .WORD   257.
300 002662  000406                          .WORD   262.
301 002664  000415                          .WORD   269.
302 002666  000423                          .WORD   275.
303 002670  000432                          .WORD   282.
304 002672  000445                          .WORD   293.
305 002674  000454                          .WORD   300.
306 002676  000463                          .WORD   307.
307 002700  000471                          .WORD   313.
308 002702  000500                          .WORD   320.
309 002704  000507                          .WORD   327.
310 002706  000514                          .WORD   332.
311 002710  000523                          .WORD   339.
312 002712  000532                          .WORD   346.
313 002714  000540                          .WORD   352.
314 002716  000545                          .WORD   357.
315 002720  000553                          .WORD   363.
316 002722  000560                          .WORD   368.
317 002724  000567                          .WORD   375.
318 002726  000576                          .WORD   382.
319 002730  000605                          .WORD   389.
320 002732  000612                          .WORD   394.
321 002734  000621                          .WORD   401.
322 002736  000627                          .WORD   407.
323 002740  000634                          .WORD   412.
324 002742  000643                          .WORD   419.
325 002744  000651                          .WORD   425.
326 002746  000660                          .WORD   432.
327 002750  000667                          .WORD   439.
328 002752  000674                          .WORD   444.
329 002754  000702                          .WORD   450.
330 002756  000711                          .WORD   457.
331 002760  000716                          .WORD   462.
332 002762  000725                          .WORD   469.
333 002764  000733                          .WORD   475.
334 002766  000742                          .WORD   482.
335 002770  000751                          .WORD   489.
336 002772  000760                          .WORD   496.
337 002774  000766                          .WORD   502.
338 002776  000774                          .WORD   508.
339 003000  000774                          .WORD   508.
340 003002  000000                          .WORD   0
341 003004  000000          SSINDX:         .WORD   0          ;SUBROUTINE STACK INDEX POINTER
342
343                          ;       OPERATIONAL FLAGS
344 003006  000000          OPFLAG:         .WORD   0          ;OPERATION FLAGS
345 003010  000000          DONE:           .WORD   0          ;OPERATION COMPLETE FLAG
```

H4


```
346 003012  000000                HADONE: .WORD   0               ;HEAD ALIGNMENT DONE FLAG
347 003014  000000                ERHEAD: .WORD   0               ;ADDRESS OF ERROR HEADER
348 003016  000000                MORECE: .WORD   0               ;MORE THAN 1 COMPARE ERROR
349 003020  000000                ERRSWI: .WORD   0               ;ERROR RETURN SWITCH
350 003022  000000                BSFLAG: .WORD   0               ;BAD SECTOR FLAGS
351 003024  000000                WRTSWI: .WORD   0               ;WRITE SWITCH
352 003026  000000                TBLSTR: .WORD   0               ;TABLE STORAGE
353
354 003030  000000                RLBAS:  .WORD   0               ;RL11 BASE ADDRESS
355 003032  000000                RLVEC:  .WORD   0               ;RL11 VECTOR ADDRESS
356 003034  000000                RLDRV:  .WORD   0               ;DRIVE NUMBER UNDER TEST
357
358 003036  000000                L.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
359 003040  000000                L.BA:   .WORD   0               ;BEFORE OPERATION
360 003042  000000                L.DA:   .WORD   0
361 003044  000000                L.MP:   .WORD   0
362 003046  000000                T.CS:   .WORD   0               ;CONTROLLER REGISTER STORAGE
363 003050  000000                T.BA:   .WORD   0               ;   AFTER OPERATION
364 003052  000000                T.DA:   .WORD   0
365 003054                        T.MP:
366 003054  000000                HDWRD1: .WORD   0               ;HEADER WORD STORAGE
367 003056  000000                HDWRD2: .WORD   0
368 003060  000000                HDWRD3: .WORD   0
369
370 003062  000000                T.STAT: .WORD   0               ;DRIVE STATE STORAGE
371
372 003064  000000                RESPARM: .WORD  0               ;PARAM BLOCK FOR REASON REPORT
373 003066  000000                         .WORD  0
374 003070  000000                         .WORD  0
375 003072  000000                         .WORD  0
376 003074  000000                         .WORD  0
377
378 003076  000000                DRVCNT: .WORD   0               ;DRIVE COUNT FOR DRIVES UNDER TEST
379 003100  000000                DIFAUG: .WORD   0               ;DIFFERENCE AUGMENT FOR SEEK
380 003102  000000                OLDCYL: .WORD   0               ;OLD CYLINDER
381 003104  000000                NEWCYL: .WORD   0               ;NEW CYLINDER
382 003106  000000                CURCYL: .WORD   0               ;CURRENT CYLINDER
383 003110  000000                DESDIF: .WORD   0               ;DESIRED DIFFERENCE
384 003112  000000                DESSGN: .WORD   0               ;DESIRED SIGN
385 003114  000000                DESHD:  .WORD   0               ;DESIRED HEAD
386 003116  000000                DESSEC: .WORD   0               ;DESIRED SECTOR
387 003120  000000                TEMP0:  .WORD   0               ;TEMPORARY STORAGE
388 003122  000000                TEMP1:  .WORD   0               ;TEMPORARY STARAGE
389 003124  000000                TEMP2:  .WORD   0               ;TEMPORARY STORAGE
390 003126  000000                TEMP3:  .WORD   0               ;TEMPORARY STORAGE
391 003130  000000                TEMP4:  .WORD   0               ;TEMPORARY STORAGE
392 003132  000000                TEMP5:  .WORD   0               ;TEMPORARY STORAGE
393 003134  000000                TEMP6:  .WORD   0               ;TEMPORARY STORAGE
394 003136  000000                TEMP7:  .WORD   0               ;TEMPORARY STORAGE
395 003140  000000                TEMP8:  .WORD   0               ;TEMPORARY STORAGE
397                               ;               TIMER STORAGE
398 003142  000000                OFIN:   .WORD   0               ;ONE CYLINDER FORWARD INNER
399 003144  000000                OFINU:  .WORD   0               ;       UPPER
400 003146  000000                OFMID:  .WORD   0               ;ONE CYLINDER FORWARD MIDDLE
401 003150  000000                OFMIDU: .WORD   0               ;       UPPER
402 003152  000000                OFOUT:  .WORD   0               ;ONE CYLINDER FORWARD OUTER
403 003154  000000                OFOUTU: .WORD   0               ;       UPPER
```

```
                                ORIN:    .WORD   0              ;ONE CYLINDER REVERSE INNER
                                ORINU:   .WORD   0              ;       UPPER
                                ORMID:   .WORD   0              ;ONE CYLINDER REVERSE MIDDLE
                                ORMIDU:  .WORD   0              ;       UPPER
                                OROUT:   .WORD   0              ;ONE CYLINDER REVERSE OUTER
                                OROUTU:  .WORD   0              ;       UPPER
                                HFIN:    .WORD   0              ;128 CYLINDER FORWARD INNER
                                HFINU:   .WORD   0              ;       UPPER
                                HFOUT:   .WORD   0              ;128 CYLINDER FORWARD OUTER
                                HFOUTU:  .WORD   0              ;       UPPER
                                HRIN:    .WORD   0              ;128 CYLINDER REVERSE INNER
                                HRINU:   .WORD   0              ;       UPPER
                                HROUT:   .WORD   0              ;128 CYLINDER REVERSE OUTER
                                HROUTU:  .WORD   0              ;       UPPER
                                AFMID:   .WORD   0              ;256 CYLINDER FORWARD
                                AFMIDU:  .WORD   0              ;       UPPER
                                ARMID:   .WORD   0              ;256 CYLINDER REVERSE
                                ARMIDU:  .WORD   0              ;       UPPER

                                EXOCYL:  .WORD   170.           ;EXPECTED TIME ONE CYLINDER
                                EXHCYL:  .WORD   550.           ;EXPECTED TIME 128 CYLINDER
                                EXACYL:  .WORD   1000.          ;EXPECTED TIME 256 CYLINDER
                                EXROT:   .WORD   250.           ;EXPECTED ROTATION TIME
                                ERRVEC:  .WORD   4              ;ERROR VECTOR

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


                                ;       BAD SECTOR TABLES AND POINTERS
                                BSFVAL:  .WORD   0              ;BAD SECTORS FILES VALID FLAG:
                                                                ;  0=NOT READ, 1=VALID, -1=NOT VALID
                                FCTBSF:  .BLKW   125.           ;FACTORY BAD SECTOR FILE STORAGE
                                         .WORD   -1             ;FULL TERMINATE
                                FLDBSF:  .BLKW   125.           ;FIELD BAD SECTOR FILE STORAGE
                                         .WORD   -1             ;FULL TERMINATE

                                IBUFF:   .BLKW   128.           ;INPUT BUFFER (1 sector of data)
```

J4


```
                                OBUFF:  .BLKW   128.            ;OUTPUT BUFFER          "
                                PAT1:   .WORD   0               ;PATTERN 1 (ALL ZEROS)
                                PAT2:   .WORD   177772
                                        .WORD   177777
                                        .WORD   177777
                                        .WORD   052525
                                        .WORD   052525
                                        .WORD   052525
                                        .WORD   177777
                                        .WORD   177777
                                        .WORD   052525
                                        .WORD   052525
                                        .WORD   177777
                                        .WORD   052525
                                        .WORD   177252
                                        .WORD   177252
                                        .WORD   172765
                                        .WORD   172765

                                PAT3:   .WORD   000003
                                        .WORD   000000
                                        .WORD   000000
                                        .WORD   177777
                                        .WORD   177777
                                        .WORD   177777
                                        .WORD   000000
                                        .WORD   000000
                                        .WORD   177777
                                        .WORD   177777
                                        .WORD   000000
                                        .WORD   177777
                                        .WORD   000000
                                        .WORD   177777

                                PAT4:   .WORD   025252
                                        .WORD   052525
                                        .WORD   052525
                                        .WORD   125252
                                        .WORD   125252
                                        .WORD   125252
                                        .WORD   052525
                                        .WORD   052525
                                        .WORD   125252
                                        .WORD   125252
                                        .WORD   052525
                                        .WORD   125252
                                        .WORD   052525
                                        .WORD   125252

                                PAT5:   .WORD   155555
                                        .WORD   133333
                                        .WORD   066666
```

K4


```
519
520 005642  121105                PAT6:   .WORD   121105
521 005644  150442                        .WORD   150442
522 005646  064221                        .WORD   064221
523 005650  132110                        .WORD   132110
524 005652  055044                        .WORD   055044
525 005654  026442                        .WORD   026442
526 005656  013211                        .WORD   013211
527 005660  105504                        .WORD   105504
528 005662  042642                        .WORD   042642
529 005664  021321                        .WORD   021321
530 005666  110550                        .WORD   110550
531 005670  044264                        .WORD   044264
532 005672  022132                        .WORD   022132
533 005674  011055                        .WORD   011055
534 005676  104426                        .WORD   104426
535 005700  042213                        .WORD   042213
536
537 005702  177777                PAT7:   .WORD   177777
538
539 005704  045513                PAT8:   .WORD   045513
540 005706  122645                        .WORD   122645
541 005710  151322                        .WORD   151322
542 005712  064551                        .WORD   064551
543 005714  132264                        .WORD   132264
544 005716  055132                        .WORD   055132
545 005720  026455                        .WORD   026455
546 005722  113226                        .WORD   113226
547 005724  045513                        .WORD   045513
548 005726  122645                        .WORD   122645
549 005730  151322                        .WORD   151322
550 005732  064551                        .WORD   064551
551 005734  132264                        .WORD   132264
552 005736  055132                        .WORD   055132
553 005740  026455                        .WORD   026455
554 005742  113226                        .WORD   113226
555
556 005744  125252                PAT9:   .WORD   125252
557
558 005746  155555                PAT10:  .WORD   155555
559
561
562                               .SBTTL  GLOBAL MESSAGES
563
565
568
569 005750     123     113     040  MSEEK:  .ASCIZ  /SK /
570 005754     122     104     040  MREAD:  .ASCIZ  /RD DATA /
571 005765     122     104     040  MREADH: .ASCIZ  /RD HDR /
572 005775     127     122     124  MWRCHK: .ASCIZ  /WRT CHCK/
573 006006     127     122     124  MWRITE: .ASCIZ  /WRT DATA /
574 006020     107     105     124  MGTSTA: .ASCIZ  /GET STAT /
575 006032     127     111     124  MDATCP: .ASCIZ  /WITH DATA CMP /
576 006051     127     111     124  MHDRCP: .ASCIZ  /WITH HDR CMP /
577 006067     106     117     122  M40HDR: .ASCIZ  /FOR 40 HDRS/
578 006103     127     111     124  MWRSET: .ASCIZ  /WITH RESET /
579 006117     117     120     105  MOPER:  .ASCIZ  /OPER: /
```

L4


```
 580 006126  122  105  123  MRSLT:  .ASCIZ  /RESULT:       /
 581 006137  125  116  114  MULOAD: .ASCIZ  /UNLD DRV/
 582 006150  114  104  040  MCYLUP: .ASCIZ  /LD DRV /
 583 006160  106  117  114  MOUTIN: .ASCIZ  /FOL 0 TO CC SK/
 584 006177  106  117  114  MINOUT: .ASCIZ  /FOL 255 TO CC SK/
 585 006220  106  117  114  MFOLWRT: .ASCIZ /FOL WRT (NO SK)/
 586 006240  101  104  112  MREVSK: .ASCIZ  /ADJ CYL WRTTN AFT REV SK/
 587 006271  101  104  112  MFWDSK: .ASCIZ  /ADJ CYL WRTTN AFT FWD SK/
 588 006322  123  113  040  MFWSKO: .ASCIZ  /SK FWD,WRT - SK REV,OVERWRT/
 589 006356  123  113  040  MRESKO: .ASCIZ  /SK REV,WRT - SK FWD,OVERWRT/
 590 006412  117  116  040  MBADAD: .ASCIZ  /ON BAD SEC FILES/
 591 006433  103  101  116  MFBSF:  .ASCIZ  /CAN'T FIND GOOD COPY OF FACTORY BAD SEC FILE/
 592 006510  103  101  116  MUBSF:  .ASCIZ  /CAN'T FIND GOOD COPY OF FIELD BAD SEC FILE/
 593 006563  102  101  104  MFMTER: .ASCIZ  /BAD SEC FILE FMT ERR/
 594 006610  102  125  123  BASADD: .ASCIZ  /BUS ADD=/
 595 006621  104  122  126  DRVNAM: .ASCIZ  /DRV=/
 596 006626  116  117  040  DRVNAV: .ASCIZ  /NO DRV FOR TST/
 597 006645  104  122  126  NOPWR:  .ASCIZ  /DRV DID NOT REC'R FROM PWR FAIL/
 598 006705  122  114  103  CSNAM:  .ASCIZ  /RLCS/
 599 006712  122  114  102  BANAM:  .ASCIZ  /RLBA/
 600 006717  122  114  104  DANAM:  .ASCIZ  /RLDA/
 601 006724  122  114  115  MPNAM:  .ASCIZ  /RLMP/
 602 006731  117  120  040  LAB1:   .ASCIZ  /OP INIT = /
 603 006744  117  120  040  LAB2:   .ASCIZ  /OP DONE = /
 604 006757  127  117  122  MWORD:  .ASCIZ  /WORD /
 605 006765  111  116  124  MTOSLOW: .ASCIZ /INTRPT TOO LATE/
 606 007005  116  117  040  MDRRES: .ASCIZ  /NO DRV RSPNSE/
 607 007023  116  117  040  MNOINT: .ASCIZ  /NO INTRPT ON CMND COMPLETE/
 608 007056  103  116  124  MCONHNG: .ASCIZ /CNTLR HUNG /
 609 007072  105  122  122  MNOCLR: .ASCIZ  /ERR DID NOT CLR/
 610 007112  126  117  114  VCNRST: .ASCIZ  /VOL CHK NOT RSET/
 611 007133  125  116  130  UNXERR: .ASCIZ  /UNXPCTED ERR/
 612 007150  040  124  105  TSTLAB: .ASCIZ  / TEST/
 630 007156  117  125  124  P2T03E: .ASCIZ  /OUT GRD BAND /
 631 007174  111  116  103  P2T04E: .ASCIZ  /INC SK FWD HD 0/
 632 007214  111  116  103  P2T05E: .ASCIZ  /INC SK REV HD 0/
 633 007234  111  116  103  P2T06E: .ASCIZ  /INC SK FWD HD 1/
 634 007254  111  116  116  P2T07E: .ASCIZ  /INN GRD BAND /
 635 007272  111  116  103  P2T08E: .ASCIZ  /INC SK REV HD 1/
 636 007312  123  113  000  P2T09E: .ASCIZ  /SK/
 637 007315  106  127  104  P2T10E: .ASCIZ  /FWD OSC SK/
 638 007330  122  105  126  P2T11E: .ASCIZ  /REV OSC SK/
 639 007343  123  113  040  P2T12E: .ASCIZ  /SK TIMING/
 640 007355  102  101  104  P2T13E: .ASCIZ  /BAD SEC FILE RD DATA/
 641 007402  127  122  124  P2T14E: .ASCIZ  &WRT/RD DATA (P1)&
 642 007423  123  120  111  P2T15E: .ASCIZ  /SPINDLE ROT TIMING/
 643 007446  127  122  124  P2T16E: .ASCIZ  &WRT/RD DATA (P2)&
 644 007467  127  122  124  P2T17E: .ASCIZ  /WRT LCK ERR AND DATA PROT/
 645 007521  101  104  112  P2T18E: .ASCIZ  /ADJ CYL INTERFNCE/
 646 007543  117  126  105  P2T19E: .ASCIZ  /OVERWRT/
 647 007553  123  113  040  SKTMES: .ASCIZ  /SK TIMES /
 648 007565  123  120  111  SRTMES: .ASCIZ  /SPINDLE ROT TIME /
 649 007607  050  111  116  VALDES: .ASCIZ  /(IN 100'S OF U-SEC)/
 650 007633  101  120  120  MAPROX: .ASCIZ  /APPROX /
 651 007643  111  116  116  LABIN:  .ASCIZ  /INNER/
 652 007651  115  111  104  LABMID: .ASCIZ  /MIDDLE/
 653 007660  117  125  124  LABOUT: .ASCIZ  /OUTER/
```

```
     654 007666     115     101     130  LABEXP: .ASCIZ  /MAX TIME/
     655 007677     061     040     103  LABOCF: .ASCIZ  /1 CYL FWD/
     656 007711     061     040     103  LABOCR: .ASCIZ  /1 CYL REV/
     657 007723     115     111     104  LABHCF: .ASCIZ  /MID CYL FWD/
     658 007737     115     111     104  LABHCR: .ASCIZ  /MID CYL REV/
     659 007753     115     101     130  LABACF: .ASCIZ  /MAX CYL FWD/
     660 007767     115     101     130  LABACR: .ASCIZ  /MAX CYL REV/
     662 010003     110     104     123  HDMOVF: .ASCIZ  /HDS FAILED TO MV IN 10 TRYS/
     680 010037     122     105     123  OPR12:  .ASCIZ  /RESET WRT LCK /
     681 010056     117     116     040  OPR1A:  .ASCIZ  /ON /
     682 010062     117     116     040  OPR1B:  .ASCIZ  /ON DRV /
     683 010072     125     116     104  UNDTST: .ASCIZ  /UNDER TEST/
     684 010105     123     105     124  OPR004: .ASCIZ  /SET WRT LCK /
     685 010122     104     111     106  DIFWD:  .ASCIZ  /DIFF /
     686 010130     123     107     116  SGNWD:  .ASCIZ  /SGN /
     687 010135     110     104     040  HDWD:   .ASCIZ  /HD /
     688 010141     123     105     103  SECWD:  .ASCIZ  /SEC /
     689 010146     103     131     114  CYLWD:  .ASCIZ  /CYL /
     690 010153     106     122     117  FRMWD:  .ASCIZ  /FROM /
     691 010161     040     102     131  BYPSNM: .ASCIZ  / BYPASSED /
     692 010174     122     117     125  SEQMES: .ASCIZ  /ROUTINE TRACE SEQ:/
     693 010217     104     122     126  STAMES: .ASCIZ  /DRV STAT/
     694 010230     124     117     124  TCERR:  .ASCIZ  /TOTAL CMP ERRS: /
     695 010251     104     122     111  NOCTLR: .ASCIZ  /DRIVE DROPPED - NO CONTROLLER/
     696 010307     104     122     111  NOTRDY: .ASCIZ  /DRIVE DROPPED - DID NOT RESPOND WITH "READY"/
     697 010364     045     116     045  NOTST:  .ASCIZ  /%N%ATEST %D2%A CANNOT BE PERFORMED...P-CLOCK NOT AVAILABLE/<CR><LF>
LF   698 010461     045     116     045  NOHD1:  .ASCIZ  /%N%ATEST %D2%A CANNOT READ BAD SEC FILE...HD 1 DISABLED BY SW QUESTION/<CR><
     699 010572     045     116     045  BSFNOT: .ASCIZ  /%N%A*WARNING* ALL SECTORS ASSUMED GOOD FOR TESTS REQUIRING BAD SEC DATA/
     700
     701                                  ;       RESULT NAMES
     702 010702     104     122     126  MDRDY:  .ASCIZ  /DRV RDY /
     703 010713     103     117     116  MCERR:  .ASCIZ  /CONT ERR /
     704 010725     110     104     122  MHCRC:  .ASCIZ  /HDR CRC/
     705 010735     104     101     124  MDCRC:  .ASCIZ  /DATA CRC/
     706 010746     110     104     122  MHNF:   .ASCIZ  /HDR NOT FND/
     707 010762     104     101     124  MDLT:   .ASCIZ  /DATA LATE/
     708 010774     110     104     122  MHFCRC: .ASCIZ  &HDR NOT FND/HDR CRC/OPI&
     709 011024     104     122     126  MDRERR: .ASCIZ  /DRV ERR /
     718 011035     104     122     126  MDSERR: .ASCIZ  /DRV SEL ERR /
     719 011052     104     122     126  MDRVST: .ASCIZ  /DRV STATE /
     720 011065     123     120     111  MSPERR: .ASCIZ  /SPIN TIMEOUT /
     721 011103     127     122     124  MWGERR: .ASCIZ  /WRT GAT ERR /
     722 011120     123     113     040  MSTERR: .ASCIZ  /SK TIMEOUT /
     723 011134     110     105     101  MHCERR: .ASCIZ  /HEAD CUR ERR /
     724 011152     127     122     124  MWDERR: .ASCIZ  /WRT DAT ERR /
     725 011167     117     120     122  MOPERR: .ASCIZ  /OPR-INC/
     726 011177     110     104     122  MHDERR: .ASCIZ  &HDR/DAT ERR &
     727 011214     110     104     122  MFLERR: .ASCIZ  &HDR NOT FND/DAT LATE &
     728 011242     116     117     116  MNEERR: .ASCIZ  /NON-EXISTENT MEMORY /
     729 011267     103     131     114  MCYLOC: .ASCIZ  /CYL /
     730 011274     103     101     116  MNDRST: .ASCIZ  /CAN'T GET DRV STAT/
     731 011317     125     116     113  MUNDEF: .ASCIZ  /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
     732 011364     106     101     111  MRLFAL: .ASCIZ  /FAIL TO RELD HDS AFTER ERR CLR/
     733 011423     127     122     124  MWRTAB: .ASCIZ  /WRT ABRTD/
     734 011435     040     117     126  MEXERS: .ASCIZ  / OVR ERR LIMIT - UNIT DRPPD /
     735 011472     040     105     122  MERRS:  .ASCIZ  / ERR/
     736 011477     207     377     377  BELL:   .ASCIZ  <207><377><377>
```

```
                                                ;
                                                ;RESULT SETTINGS
011503      111     123     040     RESE3:  .ASCIZ  /IS /
011507      040     123     102     RESE4:  .ASCIZ  / SB /
                                                ;
                                                ;RESULT CONDITIONS
011514      040     111     116     RESE5:  .ASCIZ  / IN /
011521      040     117     106     RESE6:  .ASCIZ  / OF /
011526      123     124     101     STATE2: .ASCIZ  /STATE 2/
011536      123     124     101     STATE3: .ASCIZ  /STATE 3/
011546      123     124     101     STATE5: .ASCIZ  /STATE 5/
011556      061     123     124     C10MS:  .ASCIZ  /1ST 3 MS/
011567      065     060     060     C500MS: .ASCIZ  /500MS/
011575      103     131     103     CCYLUP: .ASCIZ  /CYC UP/
011604      104     101     124     CAFDT:  .ASCIZ  /DATA XFR/
011615      065     040     123     C5SEC:  .ASCIZ  /5 SEC/

C11623      045     116     000     CRLF::  .ASCIZ  /%N/
011626      045     124     000     FMTXT:: .ASCIZ  /%T/
011631      045     116     045     FMTOP1: .ASCIZ  /%N%T%N%T%T%06%S%T%01%N/
011660      045     116     045     FMTOP2: .ASCIZ  /%N%T%01%S1%T%01%N/
011702      045     116     045     FMTOP3: .ASCIZ  /%N%T%01%S1%T%T%N/
011723      045     124     045     FMT1:   .ASCIZ  /%T%T/
011730      045     116     045     FMT2:   .ASCIZ  /%N%T%T/
011737                      FMT3:           ;unused
011737      045     116     045     FMT4:   .ASCIZ  /%N%T%T%N/
011750      045     116     045     FMT5:   .ASCIZ  /%N%T%06%S1%T%01/
011770      045     116     045     FMT6:   .ASCIZ  /%N%S11%T%S4%T%S4%T%S4%T%S4%T%S2%T/
012032      045     116     045     FMT7:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06%S3%03%S2%01%N/
012102      045     116     045     FMT8:   .ASCIZ  /%N%T%06%S2%06%S2%06%S2%06/
012134      045     116     045     FMT9:   .ASCIZ  /%N%T/
012141                      FMT10:          ;unused
012141      045     124     045     FMT11:  .ASCIZ  /%T%01/
012147      045     124     045     FMT12:  .ASCIZ  /%T%03/
012155      045     116     045     FMT13:  .ASCIZ  /%N%S11%T%03%S1%T%03%S1%T%01%S1%T%01/
012221      045     116     045     FMT14:  .ASCIZ  /%N%T%T%D3%S1%T%06%S1%T%06/
012253      045     116     045     FMT15:  .ASCIZ  /%N%S11%T%D3%S1%T%06%S1%T%06/
012307      045     116     045     FMT16:  .ASCIZ  /%N%S5%06/
012320      045     123     061     FMT17:  .ASCIZ  /%S10%T%N%S11%06%N/
012342      045     116     045     FMT18:  .ASCIZ  /%N%S15%T%S5%T%S4%T%S5%T%N/
012374      045     124     045     FMT19:  .ASCIZ  /%T%S4%D6%S4%D6%S4%D6%S4%D6%N/
012431      045     124     045     FMT20:  .ASCIZ  /%T%S2%D6%S14%D6%S4%D6%N/
012461      045     124     045     FMT21:  .ASCIZ  /%T%S12%D6%S14%D6%N/
012504      045     116     045     FMT22:  .ASCIZ  /%N%S11%T%03%S1%T%01%S1%T%02/
012540      045     124     045     FMT23:  .ASCIZ  /%T%T%T%01%N/
012554      045     116     045     FMT24:  .ASCIZ  /%N%T/
012561      045     116     045     FMT25:  .ASCIZ  /%N%D2%T/
012571      045     116     045     FMT26:  .ASCIZ  /%N%S1%T%D4%T%T%D3%N/
012615      045     116     045     FMT27:  .ASCIZ  /%N%T%D3%T%D3%N/
012634      045     116     045     FMT28:  .ASCIZ  /%N%T%T%T/
                                    .EVEN
```

```
                        .SBTTL  ERROR MESSAGES

                        ;       ERR1    R3 POINTS TO RESULT MESSAGE
                        ;               RESULT: (R3)

                        ;       ERR2    R3 POINTS TO RESULT NAME
                        ;               RESULT: (R3) IS 1 SB 0

                        ;       ERR3    R3 POINTS TO RESULT NAME
                        ;               RESULT: (R3) IS 0 SB 1

                        ;       ERR4    R3 POINTS TO RESULT NAME
                        ;               R4 POINTS TO RESULT CONDITIONS
                        ;               RESULT: (R3) IS 1 SB 0 (R4)

                        ;       ERR5    R3 POINTS TO RESULT NAME
                        ;               R4 POINTS TO RESULT CONDITIONS
                        ;               RESULT: (R3) IS 0 SB 1 (R4)

                        ;       ERR6    RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
                        ;               REPORTS ALL
                        ;               RESULT: "ERROR" IS 1 SB 0

                        ;       ERR7    DRIVE STATE ERROR REPORT
                        ;               R3 CONTAINS EXPECTED STATE
                        ;               T.STAT CONTAINS BAD STATE
                        ;               RESULT: DRIVE STATE IS (T.STAT) SB (R3)

                        ;       ERR8    HEAD POSITIONING ERROR REPORT
                        ;               NEWCYL CONTAINS EXPECTED CYLINDER
                        ;               HDWRD1 CONTAINS BAD CYLINDER
                        ;               RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)

                        ;       ERR9    UTILITY RESULT REPORT
                        ;               R3 POINTS TO RESULT NAME
                        ;               R4 POINTS TO VALUE 1
                        ;               R5 POINTS TO VALUE 2
                        ;               RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)

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

```
                                        TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
                                        BNE     1$              ;YES - SKIP
 2 012646  105737  003451               INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
 3 012652  001002                       MOV     R1,-(SP)        ;STORE R1
 4 012654  005277  170362       1$:     JSR     PC,RPTOP        ;REPORT OPERATION
 5 012660  010146                       MOV     #1,(R1)+        ;SET PARAM NUMBER
 6 012662  004737  026274               MOV     R3,(R1)+        ;INSERT MESSAGE ADDRESS POINTER
 7 012666  012721  000001               JSR     PC,RPTRES       ;REPORT RESULTS
 8 012672  010321                       JSR     PC,RPTREM       ;REPORT REMAINDER
 9 012674  004737  027062               MOV     (SP)+,R1        ;RESTORE R1
10 012700  004737  027270               JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
11 012704  012601
12 012706  004737  016616       L10000:
13 012712                               TRAP    C$MSG
   012712  104423
14
16 012714  005277  170322               INC     @ERRPOINT       ;BUMP ERROR COUNT
17 012720  010146                       MOV     R1,-(SP)        ;STORE R1
18 012722  004737  026274               JSR     PC,RPTOP        ;REPORT OPERATION
19 012726  012721  000003               MOV     #3,(R1)+        ;SET PARAM NUMBER
20 012732  010321                       MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
21 012734  012721  000001               MOV     #1,(R1)+        ;SET IS VALUE
22 012740  005021                       CLR     (R1)+           ;SET SB VALUE
23 012742  004737  027062               JSR     PC,RPTRES       ;REPORT RESULTS
24 012746  004737  027270               JSR     PC,RPTREM       ;REPORT REMAINDER
25 012752  012601                       MOV     (SP)+,R1        ;RESTORE R1
26 012754  004737  016616               JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
27 012760
   012760  104423              L10001:
28                                      TRAP    C$MSG
30 012762  005277  170254
31 012766  010146                       INC     @ERRPOINT       ;BUMP ERROR COUNT
32 012770  004737  026274               MOV     R1,-(SP)        ;STORE R1
33 012774  012721  000003               JSR     PC,RPTOP        ;REPORT OPERATION
34 013000  010321                       MOV     #3,(R1)+        ;SET PARAM NUMBER
35 013002  005021                       MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
36 013004  012721  000001               CLR     (R1)+           ;SET IS VALUE
37 013010  004737  027062               MOV     #1,(R1)+        ;SET SB VALUE
38 013014  004737  027270               JSR     PC,RPTRES       ;REPORT RESULTS
39 013020  012601                       JSR     PC,RPTREM       ;REPORT REMAINDER
40 013022  004737  016616               MOV     (SP)+,R1        ;RESTORE R1
41 013026                               JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
   013026  104423              L10002:
42                                      TRAP    C$MSG
44 013030  005277  170206
45 013034  010146                       INC     @ERRPOINT       ;BUMP ERROR COUNT
46 013036  004737  026274               MOV     R1,-(SP)        ;STORE R1
47 013042  012721  000004               JSR     PC,RPTOP        ;REPORT OPERATION
48 013046  010321                       MOV     #4,(R1)+        ;SET PARAM NUMBER
49 013050  012721  000001               MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
50 013054  005021                       MOV     #1,(R1)+        ;SET IS VALUE
51 013056  010411                       CLR     (R1)+           ;SET SB VALUE
52 013060  004737  027062               MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
53 013064  004737  027270               JSR     PC,RPTRES       ;REPORT RESULTS
54 013070  012601                       JSR     PC,RPTREM       ;REPORT REMAINDER
55 013072  004737  016616               MOV     (SP)+,R1        ;RESTORE R1
56 013076                               JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
   013076  104423              L10003:
57                                      TRAP    C$MSG
```

```
                                                INC     @ERRPOINT       ;BUMP ERROR COUNT
                                                MOV     R1,-(SP)        ;STORE R1
                                                JSR     PC,RPTOP        ;REPORT OPERATION
                                                MOV     #4,(R1)+        ;SET PARAM NUMBER
                                                MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
                                                CLR     (R1)+           ;SET IS VALUE
                                                MOV     #1,(R1)+        ;SET SB VALUE
                                                MOV     R4,(R1)         ;INSERT ADD OF CONDITION POINTER
                                                JSR     PC,RPTRES       ;REPORT RESULTS
                                                JSR     PC,RPTREM       ;REPORT REMAINDER
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
                                        L10004:
                                                TRAP    C$MSG

                                                TSTB    NOERCT          ;TEST IF ERROR COUNTING INHIBITED
                                                BNE     2$              ;YES - SKIP
                                                INC     @ERRPOINT       ;ELSE BUMP ERROR COUNT
                                        2$:     MOV     R1,-(SP)        ;STORE R1
                                                MOV     R3,-(SP)        ;STORE R3
                                                MOV     R4,-(SP)        ;STORE R4
                                                MOV     R5,-(SP)        ;STORE R5
                                                JSR     PC,RPTOP        ;REPORT OPERATION
                                                MOV     #3,(R1)+        ;SET PARAM NUMBER
                                                MOV     #1,2(R1)        ;INSERT IS VALUE
                                                CLR     TEMP3           ;CLEAR FOR STATUS STORAGE
                                                MOV     T.CS,R3         ;GET T.CS
                                                BIC     #177761,R3      ;AND CLEAR ALL BUT FUNCTION
                                                CMP     #4,R3           ;CHECK IF IT WAS GET STATUS
                                                BEQ     1$              ;YES - STATUS IS IN T.MP, SKIP
                                                MOV     #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
                                                MOV     #4,R3
                                                BIS     RLDRV,R3
                                                MOV     R3,RLCS(R2)
                                                MOV     #10.,XDELAY     ;SAVE ARGUMENT
                                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                                                BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY
                                                BNE     4$              ;YES - SKIP
                                        3$:     MOV     #BIT9,R3        ;ELSE SET NO DRIVE STATUS BIT
                                                BR      6$              ;IN MESSAGE WORD AND SKIP

                                        4$:     MOV     RLMP(R2),R3     ;STORE STATUS FOR REPORT
                                                MOV     R3,TEMP3
                                                MOVB    TEMP3+1,R3      ;GET ERROR BITS IN PROPER POSITION
                                                BR      5$

                                        1$:     MOVB    T.MP+1,R3       ;GET ERROR BITS FROM MP REG
                                        5$:     BIC     #177442,R3      ;CLEAR UNUSED BITS
                                        6$:     MOV     T.CS,R4         ;GET ERROR BITS FROM CS REG
                                                BIC     #1777,R4        ;CLEAR UNUSED BITS
                                                BIS     R4,R3           ;MAKE ONE WORD OF POSSIBLE ERRORS
                                                BIT     #OPIERR,R3      ;TEST IF OPI SET
                                                BEQ     11$             ;NO - SKIP
                                                BIT     #HNFERR,R3      ;TEST IF HDR NOT FOUND ERROR
                                                BNE     9$              ;YES - SKIP
                                                BIT     #HCRCERR,R3     ;TEST IF HDR CRC ERR
                                                BNE     8$              ;YES - SKIP
```

```
115 013366  012704  011167          MOV     #MOPERR,R4      ;SET OPI ALONE MESSAGE
116 013372                  7$:
    013372  012746  011472          MOV     #MERRS,-(SP)
    013376  010446                   MOV     R4,-(SP)
    013400  012746  006126           MOV     #MRSLT,-(SP)
    013404  012746  012634           MOV     #FMT28,-(SP)
    013410  012746  000004           MOV     #4,-(SP)
    013414  010600                   MOV     SP,R0
    013416  104414                   TRAP    C$PNTB
    013420  062706  000012           ADD     #12,SP
117 013424  000430                   BR      13$             ;SKIP
118
119 013426  012704  010725  8$:     MOV     #MHCRC,R4       ;HDR CRC MESSAGE
120 013432  000757                   BR      7$
121
122 013434  032703  004000  9$:     BIT     #HCRCERR,R3     ;TEST IF HCRC WITH HDR NOT FND
123 013440  001003                   BNE     10$             ;YES - SKIP
124 013442  012704  010746           MOV     #MHNF,R4        ;MESSAGE HEADER NOT FOUND
125 013446  000751                   BR      7$
126
127 013450  012704  010774  10$:    MOV     #MHFCRC,R4      ;HNF AND HCRC MESSAGE
128 013454  000746                   BR      7$              ;SKIP
129
130 013456  032703  004000  11$:    BIT     #DCKERR,R3      ;TEST IF DATA CHECK SET, NOT OPI
131 013462  001403                   BEQ     12$             ;NO - SKIP
132 013464  012704  010735           MOV     #MDCRC,R4       ;SET MESSAGE DATA CHECK
133 013470  000740                   BR      7$              ;SKIP
134
135 013472  032703  010000  12$:    BIT     #DLTERR,R3      ;TEST IF DATA LATE ERROR
136 013476  001403                   BEQ     13$             ;NO - SKIP
137 013500  012704  010762           MOV     #MDLT,R4        ;SET MESSAGE DATA LATE
138 013504  000732                   BR      7$              ;SKIP
139
140 013506  012705  100000  13$:    MOV     #BIT15,R5       ;SET BIT POINTER FOR TEST
141 013512  005004                   CLR     R4              ;CLEAR R4 FOR TABLE COUNT
142 013514  030503  14$:    BIT     R5,R3           ;TEST IF BIT IS SET
143 013516  001005                   BNE     16$             ;YES - SKIP TO REPORT
144 013520  005724  15$:    TST     (R4)+           ;ELSE BUMP TABLE POINTER
145 013522  000241                   CLC                     ;CLEAR CARRY
146 013524  006005                   ROR     R5              ;SHIFT BIT POINTER TO NEXT BIT
147 013526  001372                   BNE     14$             ;LOOP IF NOT 0
148 013530  000405                   BR      17$             ;ELSE REPORT REMAINDER
149
150 013532  016411  002322  16$:    MOV     RESTBL(R4),(R1) ;INSERT NAME ADDRESS
151 013536  004737  027062           JSR     PC,RPTRES       ;REPORT RESULTS
152 013542  000766                   BR      15$             ;GET NEXT BIT
153
154 013544  004737  027270  17$:    JSR     PC,RPTREM       ;REPORT REMAINDER
155 013550  005737  003126           TST     TEMP3           ;TEST IF ANY NEW STATUS
156 013554  001414                   BEQ     18$             ;NO - SKIP
157 013556  013746  003126           MOV     TEMP3,-(SP)
    013562  012746  010217           MOV     #STAMES,-(SP)
    013566  012746  012320           MOV     #FMT17,-(SP)
    013572  012746  000003           MOV     #3,-(SP)
    013576  010600                   MOV     SP,R0
    013600  104414                   TRAP    C$PNTB
    013602  062706  000010           ADD     #10,SP
```

```
158 013606 032737 004000 003046 18$:    BIT     #DCKERR,T.CS    ;TEST IF DATA CHECK ERROR
159 013614 001453                       BEQ     22$             ;NO - SKIP
160 013616 032737 002000 003046         BIT     #OPIERR,T.CS    ;TEST IF OPI SET
161 013624 001047                       BNE     22$             ;YES - SKIP
162 013626 005037 003016                CLR     MORECE          ;CLEAR COMPARE ERROR COUNT
163 013632 012701 000200                MOV     #128.,R1        ;SET COMPARE LENGTH
164 013636 012703 000001                MOV     #1,R3           ;SET WORD COUNT
165 013642 012705 005072                MOV     #OBUFF,R5       ;SET GOOD WORD POINTER
166 013646 012704 004472                MOV     #IBUFF,R4       ;SET TEST WORD POINTER
167 013652 021514               19$:    CMP     (R5),(R4)       ;CHECK WORD
168 013654 001427                       BEQ     21$             ;GOOD - SKIP
169 013656 023727 003016 000012         CMP     MORECE,#10.     ;TEST IF COMPARE LIMIT REACHED
170 013664 003021                       BGT     20$             ;YES - SKIP
171 013666 011546                       MOV     (R5),-(SP)
    013670 012746 011507                MOV     #RESE4,-(SP)
    013674 011446                       MOV     (R4),-(SP)
    013676 012746 011503                MOV     #RESE3,-(SP)
    013702 010346                       MOV     R3,-(SP)
    013704 012746 006757                MOV     #MWORD,-(SP)
    013710 012746 012253                MOV     #FMT15,-(SP)
    013714 012746 000007                MOV     #7,-(SP)
    013720 010600                       MOV     SP,R0
    013722 104414                       TRAP    C$PNTB
    013724 062706 000020                ADD     #20,SP
172 013730 005237 003016        20$:    INC     MORECE          ;BUMP ERROR COUNTER
173 013734 022524               21$:    CMP     (R5)+,(R4)+     ;BUMP POINTERS
174 013736 005203                       INC     R3              ;BUMP COUNTER
175 013740 005301                       DEC     R1              ;DEC LENGTH COUNT
176 013742 001343                       BNE     19$             ;LOOP IF NOT DONE
177 013744 005737 003016        22$:    TST     MORECE          ;TEST IF ANY COMPARE ERRORS
178 013750 001421                       BEQ     23$             ;NO - SKIP
179 013752 012701 000200                MOV     #128.,R1        ;SET COMPARE LENGTH
180 013756 010146                       MOV     R1,-(SP)
    013760 012746 011521                MOV     #RESE6,-(SP)
    013764 013746 003016                MOV     MORECE,-(SP)
    013770 012746 010230                MOV     #TCERR,-(SP)
    013774 012746 012615                MOV     #FMT27,-(SP)
    014000 012746 000005                MOV     #5,-(SP)
    014004 010600                       MOV     SP,R0
    014006 104414                       TRAP    C$PNTB
    014010 062706 000014                ADD     #14,SP
181 014014 012605               23$:    MOV     (SP)+,R5        ;RESTORE R5, 4, 3, 1
182 014016 012604                       MOV     (SP)+,R4
183 014020 012603                       MOV     (SP)+,R3
184 014022 012601                       MOV     (SP)+,R1
185 014024 004737 016616                JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
186 014030                      L10005:
    014030 104423                       TRAP    C$MSG
187
189 014032 005277 167204                INC     @ERRPOINT       ;BUMP ERROR COUNT
190 014036 010146                       MOV     R1,-(SP)        ;STORE R1
191 014040 004737 026274                JSR     PC,RPTOP        ;REPORT OPERATION
192 014044 012721 000003                MOV     #3,(R1)+        ;SET PARAM NUMBER
193 014050 012721 011052                MOV     #MDRVST,(R1)+   ;INSERT NAME ADD POINTER
194 014054 013721 003062                MOV     T.STAT,(R1)+    ;INSERT IS VALUE
195 014060 010311                       MOV     R3,(R1) ;INSERT SB VALUE
196 014062 004737 027062                JSR     PC,RPTRES       ;REPORT RESULTS
```

```
014066  004737  027270          JSR     PC,RPTREM       ;REPORT REMAINDER
014072  012601                  MOV     (SP)+,R1        ;RESTORE R1
014074  004737  016616          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
014100                  L10006:
014100  104423                  TRAP    C$MSG

014102  005277  167134          INC     @ERRPOINT       ;BUMP ERROR COUNT
014106  010146                  MOV     R1,-(SP)        ;STORE R1
014110  010346                  MOV     R3,-(SP)        ;STORE R3
014112  004737  026274          JSR     PC,RPTOP        ;REPORT OPERATION
014116  012721  000003          MOV     #3,(R1)+        ;SET PARAM NUMBER
014122  012721  011267          MOV     #MCYLOC,(R1)+   ;INSERT NAME ADD POINTER
014126  013711  003054          MOV     HDWRD1,(R1)     ;GET HEADER WORD
014132  012703  000007          MOV     #7,R3           ;SET SHIFT COUNT
014136  000241          14$:    CLC
014140  006011                  ROR     (R1)            ;ALIGN CHAR FOR PRINTING
014142  005303                  DEC     R3              ;  AS IS VALUE
014144  001374                  BNE     14$
014146  005721                  TST     (R1)+           ;BUMP PARAM POINTER
014150  013711  003104          MOV     NEWCYL,(R1)     ;INSERT SB VALUE
014154  004737  027062          JSR     PC,RPTRES       ;REPORT RESULTS
014160  004737  027270          JSR     PC,RPTREM       ;REPORT REMAINDER
014164  012603                  MOV     (SP)+,R3        ;RESTORE R3
014166  012601                  MOV     (SP)+,R1        ;RESTORE R1
014170  004737  016616          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
014174                  L10007:
014174  104423                  TRAP    C$MSG

014176  005277  167040          INC     @ERRPOINT       ;BUMP ERROR COUNT
014202  010146                  MOV     R1,-(SP)        ;STORE R1
014204  004737  026274          JSR     PC,RPTOP        ;REPORT OPERATION
014210  012721  000003          MOV     #3,(R1)+        ;SET PARAM NUMBER
014214  010321                  MOV     R3,(R1)+        ;INSERT NAME ADD POINTER
014216  010421                  MOV     R4,(R1)+        ;SET IS VALUE
014220  010521                  MOV     R5,(R1)+        ;SET SB VALUE
014222  004737  027062          JSR     PC,RPTRES       ;REPORT RESULTS
014226  004737  027270          JSR     PC,RPTREM       ;REPORT REMAINDER
014232  012601                  MOV     (SP)+,R1        ;RESTORE R1
014234  004737  016616          JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
014240                  L10010:
014240  104423                  TRAP    C$MSG

014242  010146                  MOV     R1,-(SP)        ;STORE R1
014244  005737  003016          TST     MORECE          ;TEST IF 2ND BAD LINE
014250  001051                  BNE     14$             ;YES - SKIP
014252  005277  166764          INC     @ERRPOINT       ;BUMP ERROR COUNT
014256  004737  026274          JSR     PC,RPTOP        ;REPORT OPERATION
014262  005046                  CLR     -(SP)
014264  153716  003035          BISB    RLDRV+1,(SP)
014270  012746  006621          MOV     #DRVNAM,-(SP)
014274  013746  003030          MOV     RLBAS,-(SP)
014300  012746  006610          MOV     #BASADD,-(SP)
014304  012746  011750          MOV     #FMT5,-(SP)
014310  012746  000005          MOV     #5,-(SP)
014314  010600                  MOV     SP,R0
014316  104414                  TRAP    C$PNTB
014320  062706  000014          ADD     #14,SP
```

```
245 014324  011546                    MOV     (R5),-(SP)
    014326  012746  011507              MOV     #RESE4,-(SP)
    014332  011446                    MOV     (R4),-(SP)
    014334  012746  011503              MOV     #RESE3,-(SP)
    014340  010346                    MOV     R3,-(SP)
    014342  012746  006757              MOV     #MWORD,-(SP)
    014346  012746  006126              MOV     #MRSLT,-(SP)
    014352  012746  012221              MOV     #FMT14,-(SP)
    014356  012746  000010              MOV     #10,-(SP)
    014362  010600                    MOV     SP,R0
    014364  104414                    TRAP    C$PNTB
    014366  062706  000022              ADD     #22,SP
246 014372  000421                    BR      15$
247
248 014374                      14$:
    014374  011546                    MOV     (R5),-(SP)
    014376  012746  011507              MOV     #RESE4,-(SP)
    014402  011446                    MOV     (R4),-(SP)
    014404  012746  011503              MOV     #RESE3,-(SP)
    014410  010346                    MOV     R3,-(SP)
    014412  012746  006757              MOV     #MWORD,-(SP)
    014416  012746  012253              MOV     #FMT15,-(SP)
    014422  012746  000007              MOV     #7,-(SP)
    014426  010600                    MOV     SP,R0
    014430  104414                    TRAP    C$PNTB
    014432  062706  000020              ADD     #20,SP
249 014436  005237  003016      15$:    INC     MORECE          ;INC COMPARE ERROR COUNT
250 014442  012601                    MOV     (SP)+,R1        ;RESTORE R1
251 014444  004737  016616              JSR     PC,CKERLM       ;GO CHECK IF ERROR COUNT EXCEEDED
252 014450                      L10011:
    014450  104423                    TRAP    C$MSG
```

```
                                ;LOAD PROTECTION TABLE

014452  000000                          .WORD   0               ;OFFSET OF CSR IN P-TABLE
014454  177777                          .WORD   -1              ;NOT A MASS-BUS DRIVE
014456  000011                          .WORD   DRSB+1          ;OFFSET OF DRIVE IN P-TABLE
                                        .EVEN

014460  000006                          .WORD   L10013-L$HW/2
014462  174400                          .WORD   174400          ;CSR BASE ADDRESS DEFAULT
014464  000160                          .WORD   160             ;VECTOR DEFAULT
014466  000240                          .WORD   240             ;PRIORITY DEFAULT
014470  000001                          .WORD   1               ;TYPE OF DRIVE
014472  000000                          .WORD   0               ;DRIVE NUMBER DEFAULT
014474  000001                          .WORD   1               ;RL11 CONTROLLER
014476                          L10013:

014476  000007                          .WORD   L10014-L$SW/2
014500  000000                  MISWIW: .WORD   0               ;BIT 0 = USE ALL CYLINDERS
                                                                ;BIT 1 = USE ALL SECTORS
                                                                ;BIT 2 = EXECUTE DRIVE SELECT TEST
                                                                ;BIT 3 = EXECUTE HEAD ALIGNMENT
                                                                ;BIT 12 = HEAD SELECT SUPPLIED FLAG
                                                                ;BIT 13 = HILIMIT SPECIFIED FLAG
                                                                ;BIT 14 = LO LIMIT SPECIFIED FLAG
                                                                ;BIT 15 = DO MANUAL INTERVENTION
014502  000000                  LOLIMW: .WORD   0
014504  000377                  HILIMW: .WORD   255.
014506  000000                  HEADW:  .WORD   0
014510  000024                  ERLIMW: .WORD   20.              ;ERROR LIMIT
014512  000012                  DCLIMW: .WORD   10.              ;COMPARE ERROR LIMIT
014514  000000                  BSERRS: .WORD   0               ;BSF ERROR OUTPUT FLAG
014516                          L10014:

014516  000010                          .WORD   8
014520  027554                          .WORD   T1
014522  031474                          .WORD   T2
014524  031526                          .WORD   T3
014526  031750                          .WORD   T4
014530  032564                          .WORD   T5
014532  033702                          .WORD   T6
014534  034726                          .WORD   T7
014536  036150                          .WORD   T8
```

```
                                .SBTTL  INITIALIZATION SECTION


                                ;CHECK FOR PRESENCE OF A P-CLOCK

014540  005037  003474                  CLR     CLKFLG          ;CLEAR CLOCK FLAG
014544  012700  000120                  MOV     #'P,R0
014550  104462                          TRAP    C$CLCK
014552  010037  003476                  MOV     R0,CLKADR
014556  103002                          BCC     1$
014560  005237  003474                  INC     CLKFLG          ;INDICATE PRESENCE OF A P-CLOCK
014564                          1$:
014564  012700  000340                  MOV     #340,R0
014570  104441                          TRAP    C$SPRI
014572  104433                          TRAP    C$RESET
014574  104450                          TRAP    C$MANI
014576  103403                          BCS     2$
014600  042737  100014  014500          BIC     #MITEST!DRSELT!HDALIGN,MISWIW ;CLEAR ALL MANUAL
                                                                              ; INTERVENTION FLAGS
014606  005037  003004          2$:     CLR     SSINDX          ;CLEAR SUBROUTINE STACK INDEX
014612  012700  000034                  MOV     #EF.PWR,R0
014616  104447                          TRAP    C$REFG
014620  103005                          BCC     3$
014622  013737  002012  003454          MOV     L$UNIT,PWRFLG   ;SET POWER FAIL FLAG
014630  000137  015246                  JMP     PWCON           ;GO SERVICE POWER FAIL

014634                          3$:
014634  012700  000040                  MOV     #EF.START,R0
014640  104447                          TRAP    C$REFG
014642  103034                          BCC     RESTART


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

014644  013737  002012  003076          MOV     L$UNIT,DRVCNT   ;SET UP UNIT COUNT
014652  005037  003444  RSTRT:  CLR     PASNUM          ;CLEAR PASS NUMBER
014656  012700  003244                  MOV     #ERRCNT,R0
014662  012701  000100                  MOV     #64.,R1         ;GET A COUNT
014666  005020                  1$:     CLR     (R0)+           ;CLEAR AN ERROR COUNTER STORAGE AREA
014670  005301                          DEC     R1
014672  001375                          BNE     1$              ;LOOP TILL ALL CLEARED
014674  012737  003242  003242          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
014702  012737  177777  003446          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
014710  012737  177777  003012          MOV     #-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
014716  032737  040000  014500          BIT     #LOCYL,MISWIW   ;TEST IF LO LIMIT SET
014724  001002                          BNE     2$              ;YES - SKIP
014726  005037  014502                  CLR     LOLIMW          ;ELSE CLEAR LO LIMIT
014732  000432                  2$:     BR      SETDON

014734                          RESTART:
014734  012700  000037                  MOV     #EF.RESTART,R0
014740  104447                          TRAP    C$REFG
014742  103743                          BCS     RSTRT
014744                          CONTINUE:
014744  012700  000036                  MOV     #EF.CONTINUE,R0
014750  104447                          TRAP    C$REFG
014752  103535                          BCS     PWCON
```

K5


```
                                ;       ON CONTINUE PICK UP UNIT LAST UNDER TEST
014754  012700  000035                  MOV     #EF.NEW,R0
014760  104447                          TRAP    C$REFG
014762  103403                          BCS     PASNEW

014764  005737  003076          NXTPAS: TST     DRVCNT          ;TEST IF ALL UNITS CHECKED
014770  001013                          BNE     SETDON          ;NO - SKIP

014772  005237  003444          PASNEW: INC     PASNUM          ;ELSE BUMP PASS COUNT
014776  012737  003242  003242          MOV     #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
015004  013737  002012  003076          MOV     L$UNIT,DRVCNT   ;GET ALL DRIVES
015012  012737  177777  003446          MOV     #-1,PSETNM      ;SET PARAM SELECT TO INITIAL

015020  005037  003500          SETDON: CLR     BSFVAL          ;ENABLE BAD SEC FILE READ
015024  005237  003446                  INC     PSETNM          ;NEXT SET OF PARAMETERS
015030  005337  003076                  DEC     DRVCNT          ;DOWN COUNT DRIVE TOTAL
015034  062737  000002  003242          ADD     #2,ERRPOINT     ;UPDATE THE ERROR POINTER
015042  013700  003446                  MOV     PSETNM,R0       ;SET UP TO GET PARAMETERS
015046  012702  003030                  MOV     #RLBAS,R2
015052  104442                          TRAP    C$GPHRD
015054  010001                          MOV     R0,R1
015056  103406                          BCS     1$
015060  005737  003454                  TST     PWRFLG          ;RECENT POWER FAILURE
015064  001737                          BEQ     NXTPAS  ;NO
015066  005337  003454                  DEC     PWRFLG          ;ACCOUNT FOR DRIVE
015072  000734                          BR      NXTPAS

015074  012122                  1$:     MOV     (R1)+,(R2)+     ;STORE PARAMETERS CSR
015076  012122                          MOV     (R1)+,(R2)+     ;               VECTOR
015100  005721                          TST     (R1)+           ;BUMP PAST PRIORITY
015102  012137  002300                  MOV     (R1)+,T.DRIVE
015106  012122                          MOV     (R1)+,(R2)+
015110  022737  000001  002300          CMP     #1,T.DRIVE      ;IS THIS AN RL01 TYPE DRIVE?
015116  001426                          BEQ     2$              ;BRANCH IF YES, ELSE
015120  012737  000776  002310          MOV     #510.,NXTHL     ;SETUP PARAMETERS FOR AN RL02 DRIVE
015126  012737  000777  002304          MOV     #511.,HLMTW
015134  012737  001000  002312          MOV     #512.,GBND
015142  012737  177600  002314          MOV     #177600,CAMSK
015150  012737  177600  002316          MOV     #177600,DIRMSK
015156  012737  177600  002320          MOV     #177600,HDCYL
015164  012737  177000  002306          MOV     #177000,CLRBYT
015172  000425                          BR      PWCON

015174  012737  000377  002304  2$:     MOV     #255.,HLMTW     ;SETUP PARAMETERS FOR AN RL01 DRIVE
015202  012737  000400  002312          MOV     #256.,GBND
015210  012737  077600  002314          MOV     #77600,CAMSK
015216  012737  077600  002316          MOV     #77600,DIRMSK
015224  012737  077600  002320          MOV     #77600,HDCYL
015232  012737  000376  002310          MOV     #254.,NXTHL
015240  012737  177400  002306          MOV     #177400,CLRBYT

015246  032737  020000  014500  PWCON:  BIT     #HICYL,MISWIW   ;SELECT HI CYLINDER ENABLED?
015254  001003                          BNE     1$              ;BRANCH IF NO
015256  013737  002304  014504          MOV     HLMTW,HILIMW    ;SETUP HI CYLINDER LIMIT WORD
015264                          1$:
015264  012746  000340                  MOV     #340,-(SP)
```

```
        015270  012746  016536                  MOV     #INTHLR,-(SP)
        015274  013746  003032                  MOV     RLVEC,-(SP)
        015300  012746  000003                  MOV     #3,-(SP)
        015304  104437                          TRAP    C$SVEC
        015306  062706  000010                  ADD     #10,SP
    105 015312  012700  000000                  MOV     #0,R0
        015316  104441                          TRAP    C$SPRI
    106 015320  013702  003030                  MOV     RLBAS,R2        ;SET RL11 BASE ADDRESS POINTER
    116
    117                                 ;CHECK IF POWER FAILURE WAIT IS NEEDED
    118
    119 015324  005737  003454                  TST     PWRFLG          ;NEEDED???
    120 015330  001472                          BEQ     3$              ;NO, SKIP
    121
    122 015332  013705  003034                  MOV     RLCRV,R5        ;DRIVE SELECT
    123 015336  052705  000200                  BIS     #CRDYMSK,R5     ;SET CRDY
    124 015342  010562  000000                  MOV     R5,RLCS(R2)     ;SELECT DRIVE
    125 015346  012701  000170                  MOV     #120.,R1        ;INITIALIZE WAIT COUNT
    126 015352  032762  000001  000000  2$:     BIT     #DRDYMSK,RLCS(R2)       ;DRIVE UP YET?
    127 015360  001056                          BNE     3$              ;YES START TEST
    128
    129 015362  012737  000012  003460          MOV     #10.,YDELAY     ;SAVE ARGUMENT
        015370  004737  016354                  JSR     PC,XTIME        ;CALL TIMING ROUTINE
    130 015374  005301                          DEC     R1              ;SIXTY GONE BY
    131 015376  001365                          BNE     2$              ;NO
    132 015400  012746  006645                  MOV     #NOPWR,-(SP)
        015404  012746  012554                  MOV     #FMT24,-(SP)
        015410  012746  000002                  MOV     #2,-(SP)
        015414  010600                          MOV     SP,R0
        015416  104417                          TRAP    C$PNTF
        015420  062706  000006                  ADD     #6,SP
    133 015424  005046                          CLR     -(SP)
        015426  153716  003035                  BISB    RLDRV+1,(SP)
        015432  012746  006621                  MOV     #DRVNAM,-(SP)
        015436  013746  003030                  MOV     RLBAS,-(SP)
        015442  012746  006610                  MOV     #BASADD,-(SP)
        015446  012746  011750                  MOV     #FMT5,-(SP)
        015452  012746  000005                  MOV     #5,-(SP)
        015456  010600                          MOV     SP,R0
        015460  104417                          TRAP    C$PNTF
        015462  062706  000014                  ADD     #14,SP
    134 015466  012746  011623                  MOV     #CRLF,-(SP)
        015472  012746  000001                  MOV     #1,-(SP)
        015476  010600                          MOV     SP,R0
        015500  104417                          TRAP    C$PNTF
        015502  062706  000004                  ADD     #4,SP
    135 015506  013700  003446                  MOV     PSETNM,R0
        015512  104451                          TRAP    C$DODU
    136 015514  104444                          TRAP    C$DCLN
    137 015516                          3$:
    138
    139 015516                          L10015:
        015516  104411                          TRAP    C$INIT
```

M5


```
                                .SBTTL  AUTO DROP SECTION

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

015520  005037  003452                CLR     TRPFLG          ;CLEAR TRAP FLAG
015524  012746  000340                MOV     #340,-(SP)
015530  012746  016530                MOV     #TRPHAN,-(SP)
015534  013746  003232                MOV     ERRVEC,-(SP)
015540  012746  000003                MOV     #3,-(SP)
015544  104437                       TRAP    C$SVEC
015546  062706  000010                ADD     #10,SP
                                                        ;/NON-EXISTENT CONTROLLER
015552  013702  003030                MOV     RLBAS,R2        ;GET RL11 BASE ADDRESS
015556  005762  000000                TST     RLCS(R2)        ;ACCESS DRIVE CONTROLLER ADDRESS
015562  005737  003452                TST     TRPFLG          ;DID TRAP OCCUR?
015566  001447                       BEQ     1$              ;BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
015570  012746  010251                MOV     #NOCTLR,-(SP)
015574  012746  012554                MOV     #FMT24,-(SP)
015600  012746  000002                MOV     #2,-(SP)
015604  010600                       MOV     SP,R0
015606  104417                       TRAP    C$PNTF
015610  062706  000006                ADD     #6,SP
015614  005046                       CLR     -(SP)
015616  153716  003035                BISB    RLDRV+1,(SP)
015622  012746  006621                MOV     #DRVNAM,-(SP)
015626  013746  003030                MOV     RLBAS,-(SP)
015632  012746  006610                MOV     #BASADD,-(SP)
015636  012746  011750                MOV     #FMT5,-(SP)
015642  012746  000005                MOV     #5,-(SP)
015646  010600                       MOV     SP,R0
015650  104417                       TRAP    C$PNTF
015652  062706  000014                ADD     #14,SP
                                                        ;PRINT DRIVE INFORMATION
015656  012746  011623                MOV     #CRLF,-(SP)
015662  012746  000001                MOV     #1,-(SP)
015666  010600                       MOV     SP,R0
015670  104417                       TRAP    C$PNTF
015672  062706  000004                ADD     #4,SP

015676  013700  003446                MOV     PSETNM,R0
015702  104451                       TRAP    C$DODU
015704  000460                       BR      2$              ;BRANCH TO EXIT

015706  013705  003034        1$:     MOV     RLDRV,R5        ;ELSE, GET DRIVE NUMBER
015712  052705  000200                BIS     #CRDYMSK,R5     ;SET CONTROLLER READY
015716  010562  000000                MOV     R5,RLCS(R2)     ;LOAD IN THE DRIVE NUMBER
015722  032762  000001  000000        BIT     #DRDYMSK,RLCS(R2)       ;IS DRIVE READY?
015730  001046                       BNE     2$              ;BRANCH TO PERFORM TESTS IF DRIVE IS READY
015732  012746  010307                MOV     #NOTRDY,-(SP)
015736  012746  012554                MOV     #FMT24,-(SP)
015742  012746  000002                MOV     #2,-(SP)
015746  010600                       MOV     SP,R0
```

N5


```
                015750  104417                  TRAP    C$PNTF
                015752  062706  000006          ADD     #6,SP
32
33              015756  005046                  CLR     -(SP)                   ;/WITH 'READY'"
                015760  153716  003035          BISB    RLDRV+1,(SP)
                015764  012746  006621          MOV     #DRVNAM,-(SP)
                015770  013746  003030          MOV     RLBAS,-(SP)
                015774  012746  006610          MOV     #BASADD,-(SP)
                016000  012746  011750          MOV     #FMT5,-(SP)
                016004  012746  000005          MOV     #5,-(SP)
                016010  010600                  MOV     SP,R0
                016012  104417                  TRAP    C$PNTF
                016014  062706  000014          ADD     #14,SP
34
35              016020  012746  011623          MOV     #CRLF,-(SP)             ;PRINT DRIVE INFORMATION
                016024  012746  000001          MOV     #1,-(SP)
                016030  010600                  MOV     SP,R0
                016032  104417                  TRAP    C$PNTF
                016034  062706  000004          ADD     #4,SP
36              016040  013700  003446          MOV     PSETNM,R0
                016044  104451                  TRAP    C$DODU
37              016046                  2$:
                016046  013700  003232          MOV     ERRVEC,R0
                016052  104436                  TRAP    C$CVEC
38              016054                  L10016:
                016054  104461                  TRAP    C$AUTO
```

```
                                        .SBTTL  CLEANUP CODE SECTION

                                                MOV     #340,-(SP)
                                                MOV     #TRPHAN,-(SP)
                                                MOV     ERRVEC,-(SP)
                                                MOV     #3,-(SP)
                                                TRAP    C$SVEC
                                                ADD     #10,SP
                                                MOV     #7,R0
                                                TRAP    C$SPRI
                                1$:             BIT     #CRDYMSK,RLCS(R2)       ;TEST IF CONTROLLER READY
                                                BEQ     2$                      ;NO LOOP UNTIL READY
                                                BIS     RLDRV,RLCS(R2)  ;SET DRIVE NUMBER
                                                BIT     #DRDYMSK,RLCS(R2)       ;TEST IF DRIVE BUSY
                                                BNE     3$                      ;NO - SKIP
                                2$:
                                                MOV     #3,YDELAY               ;SAVE ARGUMENT
                                                JSR     PC,XTIME                ;CALL TIMING ROUTINE
                                3$:
                                                MOV     RLVEC,R0
                                                TRAP    C$CVEC
                                                TST     PWRFLG                  ;PWR FAIL SET
                                                BEQ     4$                      ;NO
                                                DEC     PWRFLG
                                4$:
                                                MOV     ERRVEC,R0
                                                TRAP    C$CVEC
                                                TRAP    C$RESET
                                L10017:
                                                TRAP    C$CLEAN
                                                NOP
                                L10020:
                                                TRAP    C$DU
```

C6


```
                                .SBTTL  GLOBAL SUBROUTINES


016210  012737  000160  002116  TIME:   MOV     #160,L$DLY              ;GET OUTER DELAY LOOP
016216  005237  003466          ;       INC     TIM.US                  ;US-WAIT ROUTINE INDICATOR
016222  013737  003456  003462  ;       MOV     XDELAY,MININC           ;SAVE ORIGINAL US-WAIT
016230  005437  003456          ;       NEG     XDELAY                  ;GET NEGATIVE OF FACTOR
016234  104407                  ;       TRAP    C$RDBU
016236  103420                  ;       BCS     2$
016240                          1$:
016240  012727  000001          ;       MOV     #1.,(PC)+
016244  000000                  ;       .WORD   0
016246  013727  002116          ;       MOV     L$DLY,(PC)+
016252  000000                  ;       .WORD   0
016254  005367  177772          ;       DEC     -6(PC)
016260  001375                  ;       BNE     .-4
016262  005367  177756          ;       DEC     -22(PC)
016266  001367                  ;       BNE     .-20
016270  005237  003456          ;       INC     XDELAY                  ;WAIT FACTOR EXPIRED?
016274  002761                  ;       BLT     1$                      ;BRANCH - IF NO
016276  000422                  ;       BR      4$                      ;GET TIME

016300  012737  000065  002116  2$:     MOV     #65,L$DLY               ;GET OUTER DELAY LOOP
016306                          3$:
016306  012727  000001          ;       MOV     #1.,(PC)+
016312  000000                  ;       .WORD   0
016314  013727  002116          ;       MOV     L$DLY,(PC)+
016320  000000                  ;       .WORD   0
016322  005367  177772          ;       DEC     -6(PC)
016326  001375                  ;       BNE     .-4
016330  005367  177756          ;       DEC     -22(PC)
016334  001367                  ;       BNE     .-20
016336  005237  003456          ;       INC     XDELAY                  ;WAIT FACTOR EXPIRED?
016342  002761                  ;       BLT     3$                      ;BRANCH - IF NO
016344  063737  003462  003120  4$:     ADD     MININC,TEMPO            ;GET TIME EXPIRED
016352  000207                  ;       RTS     PC                      ;RETURN

016354  012737  000160  002116  XTIME:  MOV     #160,L$DLY              ;GET OUTER DELAY LOOP
016362  005037  003466          ;       CLR     TIM.US                  ;MS. WAIT INDICATOR
016366  013737  003460  003472  ;       MOV     YDELAY,MAJINC           ;SAVE ORIGINAL WAIT MS
016374  006337  003460          ;       ASL     YDELAY                  ;MULTIPLY BY FACTOR 4
016400  006337  003460          ;       ASL     YDELAY                  ;---------
016404  005437  003460          ;       NEG     YDELAY                  ;GET NEGATIVE OF RESULT
016410  104407                  ;       TRAP    C$RDBU
016412  103023                  ;       BCC     2$
016414  012737  000150  002116  ;       MOV     #150,L$DLY              ;GET OUTER DELAY LOOP
016422                          1$:
016422  012727  000020          ;       MOV     #20.,(PC)+
016426  000000                  ;       .WORD   0
016430  013727  002116          ;       MOV     L$DLY,(PC)+
016434  000000                  ;       .WORD   0
016436  005367  177772          ;       DEC     -6(PC)
016442  001375                  ;       BNE     .-4
016444  005367  177756          ;       DEC     -22(PC)
016450  001367                  ;       BNE     .-20
016452  005237  003460          ;       INC     YDELAY                  ;WAIT FACTOR EXPIRED
016456  002761                  ;       BLT     1$                      ;BRANCH - IF NO
```

D6


```
                                        BR      3$                      ;GET TIME

                                2$:
016462  012727  000010                MOV     #10,(PC)+
016466  000000                        .WORD   0
016470  013727  002116                MOV     L$DLY,(PC)+
016474  000000                        .WORD   0
016476  005367  177772                DEC     -6(PC)
016502  001375                        BNE     .-4
016504  005367  177756                DEC     -22(PC)
016510  001367                        BNE     .-20
016512  005237  003460                INC     YDELAY                  ;WAIT FACTOR EXPIRED?
016516  002761                        BLT     2$                      ;BRANCH - IF NO
016520  063737  003472  003464  3$:   ADD     MAJINC,TEMP             ;GET EXPIRED TIME
016526  000207                        RTS     PC                      ;RETURN


                                ;TRAP HANDLER INDICATES OCCURRENCE OF A TRAP.

016530  005237  003452        TRPHAN: INC     TRPFLG

016534                          L10021:
016534  000002                        RTI


                                ;INTERRUPT HANDLER. ABORTS WAIT TIMER AND STORES RL11 REGISTERS.

016536  012237  003046        INTHLR: MOV     (R2)+,T.CS              ;STORE RL REGISTERS
016542  012237  003050                MOV     (R2)+,T.BA
016546  012237  003052                MOV     (R2)+,T.DA
016552  011237  003054                MOV     (R2),T.MP
016556  012737  177777  003010        MOV     #-1,DONE                ;SET DONE FLAG
016564  013702  003030                MOV     RLBAS,R2                ;RESTORE R2
016570  013737  003456  003120        MOV     XDELAY,TEMPO            ;SAVE MICRO-SEC RUN TIME
016576  013737  003460  003464        MOV     YDELAY,TEMP             ;SAVE MILLI-SEC RUN TIME
016604  005037  003456                CLR     XDELAY                  ;ABORT MICRO-SEC WAIT
016610  005037  003460                CLR     YDELAY                  ;ABORT MILLI-SEC WAIT
016614                          L10022:
016614  000002                        RTI
```

E6


```
                                        ;       ERROR LIMIT CHECKING ROUTINE
                                        ;       DROPS DRIVE IF ERROR LIMIT EXCEEDED
                                        ;
016616  027737  164420  014510  CKERLM: CMP     @ERRPOINT,ERLIMW        ;TEST IF ERROR LIMIT EXCEEDED
016624  002453                          BLT     1$                      ;NO - SKIP
016626  104420                          TRAP    C$INLP
016630  103451                          BCS     1$
016632  012746  011435                  MOV     #MEXERS,-(SP)
016636  013746  014510                  MOV     ERLIMW,-(SP)
016642  012746  012561                  MOV     #FMT25,-(SP)
016646  012746  000003                  MOV     #3,-(SP)
016652  010600                          MOV     SP,R0
016654  104417                          TRAP    C$PNTF
016656  062706  000010                  ADD     #10,SP
016662  005046                          CLR     -(SP)
016664  153716  003035                  BISB    RLDRV+1,(SP)
016670  012746  006621                  MOV     #DRVNAM,-(SP)
016674  013746  003030                  MOV     RLBAS,-(SP)
016700  012746  006610                  MOV     #BASADD,-(SP)
016704  012746  011750                  MOV     #FMT5,-(SP)
016710  012746  000005                  MOV     #5,-(SP)
016714  010600                          MOV     SP,R0
016716  104417                          TRAP    C$PNTF
016720  062706  000014                  ADD     #14,SP
016724  012746  011623                  MOV     #CRLF,-(SP)
016730  012746  000001                  MOV     #1,-(SP)
016734  010600                          MOV     SP,R0
016736  104417                          TRAP    C$PNTF
016740  062706  000004                  ADD     #4,SP
016744  013700  003446                  MOV     PSETNM,R0
016750  104451                          TRAP    C$DODU
016752  104444                          TRAP    C$DCLN
016754  000207                  1$:     RTS     PC

                                        ;       READ AND STORE ALL RL11 REGISTERS

016756  016237  000000  003046  READRL: MOV     RLCSR(R2),T.CS  ;GET CS REG
016764  016237  000002  003050          MOV     RLBA(R2),T.BA   ;GET BUS ADDRESS REG
016772  016237  000004  003052          MOV     RLDA(R2),T.DA   ;GET DISK ADDRESS
017000  016237  000006  003054          MOV     RLMP(R2),T.MP   ;GET MULTI-PURPOSE REG
017006  000207                          RTS     PC              ;RETURN
```

F6


```
                                        ;       WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
                                        ;
017010  011646                  WAITIN: MOV     (SP),-(SP)      ;MAKE ROOM FOR ERROR POINTER
017012  005066  000002                  CLR     2(SP)           ;CLEAR FOR POINTER
017016  032762  000200  000000          BIT     #CRDYMSK,RLCSR(R2)      ;TEST IF CONTROLLER READY
017024  001420                          BEQ     3$              ;NO - SKIP TO WAIT
017026  004737  016756                  JSR     PC,READRL       ;READ ALL RL REGS
017032  005737  003010                  TST     DONE            ;TEST IF INTERRUPT OCCURRED
017036  001435                          BEQ     5$              ;NO - GO SET NO INTERRUPT ERR FLAG
017040  012766  006765  000002  1$:     MOV     #MTOSLOW,2(SP)  ;ELSE SET TOO SLOW ERROR POINTER
017046  032737  002000  003046          BIT     #OPIERR,T.CS    ;TEST IF OPI SET
017054  001403                          BEQ     2$              ;NO - SKIP
017056  012766  007005  000002          MOV     #MDRRES,2(SP)   ;SET MESSAGE FOR NO DRIVE RESPONSE
017064  000207                  2$:     RTS     PC              ;RETURN
017066                          3$:
017066  012737  000003  003460          MOV     #3,YDELAY       ;SAVE ARGUMENT
017074  004737  016354                  JSR     PC,XTIME        ;CALL TIMING ROUTINE
017100  032762  000200  000000          BIT     #CRDYMSK,RLCS(R2)       ;TEST IF READY NOW SET
017106  001006                          BNE     4$              ;YES - SKIP
017110  004737  016756                  JSR     PC,READRL       ;READ RL REGS
017114  012766  007056  000002          MOV     #MCONHNG,2(SP)  ;SET MESSAGE FOR CONTROLLER HUNG
017122  000760                          BR      2$              ;SKIP

017124  005737  003010          4$:     TST     DONE            ;ELSE CHECK IF INTERRUPT OCCURRED
017130  001343                          BNE     1$              ;YES - SKIP TO SET TOO SLOW
017132  004737  016756          5$:     JSR     PC,READRL       ;READ RL REGS
017136  012766  007023  000002          MOV     #MNOINT,2(SP)   ;ELSE SET NO INTERRUPT FLAG
017144  000747                          BR      2$              ;GO TO RETURN

                                        ;       OPERATION AND TEST INITIALIZE ROUTINE

017146  005037  003006          TSTINT: CLR     OPFLAG          ;CLEAR OPERATION FLAGS
017152  105037  003451                  CLRB    NOERCT          ;RESET INHIBIT ERROR COUNTING
017156  005037  003016                  CLR     MORECE          ;RESET MORE COMPARE ERRORS
017162  000207                          RTS     PC
```

G6


```
                                        ;       GET STATUS AND GET STATUS WITH RESET ROUTINE
                                        ;
017164  013746  003130                  GSTATR: MOV     TEMP4,-(SP)     ;STORE TEMP4
017170  012737  000013  003130                  MOV     #GETSTAT!DRSET,TEMP4    ;SET FOR RESET
017176  000412                          GSTATG: BR      GSTATG
                                        ;
017200  013746  003130                  GSTATC: MOV     TEMP4,-(SP)     ;STORE TEMP4
017204  012737  000003  003130                  MOV     #GETSTAT,TEMP4  ;SET FOR NO RESET
017212  000404                                  BR      GSTATG
                                        ;
017214  013746  003130                  GSTAT:  MOV     TEMP4,-(SP)     ;STORE TEMP4
017220  005037  003130                          CLR     TEMP4           ;SET FOR SAVE L. AND T. REGS
017224  010346                          GSTATG: MOV     R3,-(SP)        ;STORE R3
017226  013703  003004                          MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
017232  005723                                  TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
017234  016663  000004  002406                  MOV     4(SP),SUBSTK(R3) ;INSERT THIS CALL
017242  162763  000004  002406                  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
017250  010337  003004                          MOV     R3,SSINDX       ;STORE IT BACK
017254  010046                                  MOV     R0,-(SP)        ;STORE R0
017256  010146                                  MOV     R1,-(SP)        ;STORE R1
017260  012737  000002  003020                  MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
017266  032737  000010  003130                  BIT     #DRSET,TEMP4    ;TEST IF DRIVE RESET
017274  001460                                  BEQ     4$              ;NO - SKIP
017276  032762  040000  000000                  BIT     #DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
017304  001405                                  BEQ     1$              ;NO - SKIP
017306  012737  000003  003460                  MOV     #3,YDELAY       ;SAVE ARGUMENT
017314  004737  016354                          JSR     PC,XTIME        ;CALL TIMING ROUTINE
017320  012701  000062                  1$:     MOV     #50.,R1         ;INITIALIZE WAIT COUNT
017324  004737  017214                  2$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
017330  020014                                  16$
017332  032737  000001  003046                  BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
017340  001054                                  BNE     6$              ;YES - GO DO CLEAR
017342  032737  000020  003054                  BIT     #HOSTAT,T.MP    ;ELSE TEST IF HEADS OUT
017350  001010                                  BNE     3$              ;YES - BYPASS RELOAD WAIT FLAG SETTING
017352  032737  144000  003054                  BIT     #SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
                                                                        ;THAT CAUSED HEADS TO
                                                                        ;UNLOAD
017360  001444                                  BEQ     6$              ;NO - SKIP
017362  052737  040000  003006                  BIS     #RELDWT,OPFLAG  ;ELSE SET WAIT FLAG
017370  000440                                  BR      6$              ;SKIP TO CLEAR
                                        ;
017372  032737  040000  003046  3$:     BIT     #DRVERR,T.CS    ;TEST IF DRIVE ERROR NOW
017400  001034                                  BNE     6$              ;YES - SKIP TO CLEAR
017402  012737  000001  003460                  MOV     #1,YDELAY       ;SAVE ARGUMENT
017410  004737  016354                          JSR     PC,XTIME        ;CALL TIMING ROUTINE
017414  005301                                  DEC     R1              ;DEC WAIT COUNTER
017416  001342                                  BNE     2$              ;IF NOT DONE, LOOP
017420  012703  011317                          MOV     #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
017424  104456                                  TRAP    C$ERHRD
017426  023421                                  .WORD   10001
017430  000000                                  .WORD   0
017432  012646                                  .WORD   ERR1
017434  000565                                  BR      15$             ;EXIT
                                        ;
017436  005737  003130                  4$:     TST     TEMP4           ;TEST IF SAVE REGISTERS
017442  001013                                  BNE     6$              ;NO SKIP
017444  012701  000004                          MOV     #4,R1           ;SET SAVE COUNT
```

```
                                                MOV     #L.MP+2,R3      ;SET ADDRESS OF FIRST SAVE
                                        5$:     MOV     -(R3),-(SP)     ;PUT REG ON STACK
                                                DEC     R1              ;DEC COUNT
                                                BNE     5$              ;LOOP UNTIL ALL SAVED
                                                MOV     #GETSTAT,L.DA   ;SET FOR GET STATUS
                                                BR      7$              ;SKIP

                                        6$:     MOV     TEMP4,L.DA      ;INSERT PRESET FOR STATUS
                                        7$:
                                                CLR     DONE            ;CLEAR INTERRUPT FLAG
                                                MOV     RLDRV,L.CS      ;SET UP TO GET STATUS
                                                BIC     #BIT10,L.CS     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
                                                BIS     #GTSTAT,L.CS
                                                MOV     L.DA,RLDA(R2)   ;LOAD RL REGS
                                                MOV     L.CS,RLCSR(R2)  ;LOAD CS REG
                                                MOV     #1,XDELAY       ;SAVE ARGUMENT
                                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                                                TST     DONE            ;CHECK IF INTERRUPT OCCURRED
                                                BEQ     13$             ;NO - SKIP
                                        8$:     MOV     T.MP,T.STAT     ;STORE MP REGISTER
                                                BIC     #^C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
                                                BIT     #DRSET,L.DA     ;TEST IF RESET WAS SPECIFIED
                                                BEQ     16$             ;NO - SKIP TO EXIT
                                                BIT     #RELDWT,OPFLAG  ;TEST IF RELOAD WAIT FLAG SET
                                                BEQ     10$             ;NO - SKIP
                                                MOV     #600.,R1        ;SET WAIT COUNT FOR 60 SECONDS
                                        9$:     BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
                                                BNE     10$             ;YES - SKIP
                                                MOV     #1,YDELAY       ;SAVE ARGUMENT
                                                JSR     PC,XTIME        ;CALL TIMING ROUTINE
                                                DEC     R1              ;DEC COUNT
                                                BNE     9$              ;LOOP IF NOT 0
                                                JSR     PC,GSTAT        ;GET DRIVE STATUS
                                                16$                     ;ERROR RETURN
                                                MOV     #MRLFAL,R3      ;SET RESULT MESSAGE POINTER
                                                TRAP    C$ERHRD
                                                .WORD   10003
                                                .WORD   0
                                                .WORD   ERR1
                                                BR      15$             ;GO TO EXIT

                                        10$:
                                                MOV     #10.,XDELAY     ;SAVE ARGUMENT
                                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                                                JSR     PC,GSTAT        ;GET DRIVE STATUS
                                                16$
                                                BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR
                                                BEQ     16$             ;NO - SKIP
                                                BIT     #VCSTAT,T.MP    ;CHECK IF VOLUME CHECK RESET
                                                BEQ     11$             ;YES SKIP
                                                MOV     #VCNRST,R3      ;SET REASON POINTER
                                                BR      14$             ;EXIT

                                        11$:    BIT     #DRVERR,T.CS    ;CHECK IF DRIVE ERROR
                                                BEQ     12$             ;NO - SKIP
                                                TRAP    C$ERHRD
                                                .WORD   10004
```

I6


```
                017756  000000                          .WORD   0
                017760  013150                          .WORD   ERR6
    102         017762  000412                          BR      15$             ;EXIT
    103
    104         017764  012703  007133          12$:    MOV     #UNXERR,R3      ;SET REASON POINTER
    105         017770  000403                          BR      14$             ;EXIT
    106
    107         017772  004737  017010          13$:    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
    108         017776  012603                          MOV     (SP)+,R3        ;STORE REASON POINTER FOR RETURN
    109         020000                          14$:
                020000  104456                          TRAP    C$ERHRD
                020002  023422                          .WORD   10002
                020004  000000                          .WORD   0
                020006  012646                          .WORD   ERR1
    110         020010  005037  003020          15$:    CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
    111         020014  005737  003130          16$:    TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
    112         020020  001007                          BNE     18$             ;NO - SKIP
    113         020022  012703  003036                  MOV     #L.CS,R3        ;SET POINTER TO RESTORE
    114         020026  012701  000004                  MOV     #4,R1           ;SET REGISTER COUNT
    115         020032  012623                  17$:    MOV     (SP)+,(R3)+     ;RESTORE REG
    116         020034  005301                          DEC     R1              ;DEC COUNT
    117         020036  001375                          BNE     17$             ;LOOP UNTIL ALL ARE RESTORED
    118         020040  162737  000002  003004  18$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
    119         020046  012601                          MOV     (SP)+,R1        ;RESTORE R1
    120         020050  012600                          MOV     (SP)+,R0        ;RESTORE R0
    121         020052  012603                          MOV     (SP)+,R3        ;RESTORE R3
    122         020054  012637  003130                  MOV     (SP)+,TEMP4     ;RESTORE TEMP4
    123         020060  005737  003020                  TST     ERRSWI          ;TEST IF ERROR RETURN
    124         020064  001403                          BEQ     19$             ;YES - SKIP
    125         020066  063716  003020                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
    126         020072  000207                          RTS     PC
    127         020074  017616  000000          19$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
    128         020100  000207                          RTS     PC
```

```
                                        ;       SEEK ROUTINE
                                        ;
020102  012737  177777  003122  XSEEKT: MOV     #-1,TEMP1       ;SET SPECIAL TIMING SEEK FLAG
020110  000402                          BR      XSEEK1
                                        ;
020112  005037  003122          XSEEK:  CLR     TEMP1           ;CLEAR SPECIAL SEEK FOR TIMING FLAG
020116  010346                  XSEEK1: MOV     R3,-(SP)        ;STORE R3
020120  013703  003004                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
020124  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
020126  016663  000002  002406          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
020134  162763  000004  002406          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
020142  010337  003004                  MOV     R3,SSINDX       ;STORE IT BACK
020146  010046                          MOV     R0,-(SP)
020150  010146                          MOV     R1,-(SP)
020152  010546                          MOV     R5,-(SP)        ;STORE REG
020154  012737  000002  003020          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
020162  005037  003100                  CLR     DIFAUG          ;CLEAR DIFFERENCE AUGMENT (FOR SEEKING
                                        ;                       PAST GUARD BAND)
020166  004737  024054                  JSR     PC,GETPOS       ;GET PRESENT POSITION
020172  020624                          12$
020174  013737  003106  003102          MOV     CURCYL,OLDCYL   ;MOVE CURRENT TO OLD CYLINDER
020202  023737  003104  002304          CMP     NEWCYL,HLMTW    ;TEST IF NEW IS GREATER THAN MAX CYL
020210  003427                          BLE     1$              ;NO - SKIP
020212  163737  002304  003104          SUB     HLMTW,NEWCYL    ;ELSE SUBTRACT MAX CYL.
020220  013737  003104  003100          MOV     NEWCYL,DIFAUG   ;STORE DIFFERENCE AS AUGMENT
020226  013737  002304  003104          MOV     HLMTW,NEWCYL    ;SET NEWCYL AS MAX CYL.
020234  022737  000001  002300          CMP     #1,T.DRIVE
020242  001424                          BEQ     2$
020244  162737  000001  003104          SUB     #1,NEWCYL
020252  012737  000001  003112          MOV     #1,DESSGN
020260  012737  000001  003110          MOV     #1,DESDIF
020266  000451                          BR      6$
                                        ;
020270  005737  003104          1$:     TST     NEWCYL          ;TEST IF NEWCYL HAS NEGATIVE VALUE
020274  100007                          BPL     2$              ;NO - SKIP
020276  005437  003104                  NEG     NEWCYL          ;ELSE MAKE IT POSITIVE
020302  013737  003104  003100          MOV     NEWCYL,DIFAUG   ;AND STORE IT AS AUGMENT
020310  005037  003104                  CLR     NEWCYL          ;AND SET NEWCYL TO 0
020314  013705  003106          2$:     MOV     CURCYL,R5       ;COMPUTE DIFFERENCE AND NEW CYLINDER
020320  163705  003104                  SUB     NEWCYL,R5       ;SUB NEWCYL FROM CURCYL
020324  100005                          BPL     3$              ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
020326  012737  000001  003112          MOV     #1,DESSGN       ;ELSE SET SIGN FOR FORWARD
020334  005405                          NEG     R5              ;MAKE DIFFERENCE POSITIVE
020336  000402                          BR      4$              ;SKIP
                                        ;
020340  005037  003112          3$:     CLR     DESSGN          ;SET SIGN FOR REVERSE
020344  010537  003110          4$:     MOV     R5,DESDIF       ;STORE DIFFERENCE
020350  005737  003100                  TST     DIFAUG          ;IS THERE A DIFFERENCE AUGMENT
020354  001416                          BEQ     6$              ;NO - SKIP
020356  023737  003104  002304          CMP     NEWCYL,HLMTW    ;CHECK IF NEW CYL IS MAX CYL.
020364  001007                          BNE     5$              ;NO - SKIP
020366  012737  000001  003112          MOV     #1,DESSGN       ;ELSE FORCE SIGN FOR FORWARD
                                        ;(INNER GUARD BAND)
020374  022737  000001  002300          CMP     #1,T.DRIVE
020402  001003                          BNE     6$
020404  063737  003100  003110  5$:     ADD     DIFAUG,DESDIF
020412  012705  003036          6$:     MOV     #L.CS,R5        ;GET L REG ADDRESS
```

K6


```
                                                MOV     #SEEK,(R5)      ;SET FOR SEEK
                                                BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
                                                BIC     #BIT10,(R5)+    ;CLEAR IF DRIVE 4 - 7 SPEC'D
                                                CLR     (R5)+           ;CLEAR BUS ADDRESS
                                                MOV     DESDIF,(R5)     ;LOAD DIFFERENCE
                                                MOV     #7,R0           ;SET TO SHIFT DIFFERENCE
                                        7$:     ASL     (R5)
                                                DEC     R0
                                                BNE     7$              ;LOOP UNTIL ALIGNED
                                                TST     DESSGN          ;TEST SIGN
                                                BEQ     8$              ;SKIP IF 0
                                                BIS     #DIRBIT,(R5)    ;ELSE INSERT SIGN
                                        8$:     TST     DESHD           ;TEST IF HEAD 0
                                                BEQ     9$              ;YES - SKIP
                                                BIS     #HDSEL,(R5)     ;ELSE SET HEAD BIT
                                        9$:     BIS     #MBSET0,(R5)+   ;INSERT MARKER BIT
                                                JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
                                                12$
                                                CLR     DONE            ;CLEAR INTERRUPT FLAG
                                                TST     TEMP1           ;CHECK IF SPECIAL SEEK FLAG SET
                                                BNE     12$             ;YES - SKIP, DO NOT START SEEK
                                                MOV     -(R5),RLDA(R2)  ;LOAD RL REGISTERS
                                                MOV     -(R5),RLBA(R2)
                                                MOV     -(R5),RLCS(R2)
                                       10$:
                                                MOV     #10.,XDELAY     ;SAVE ARGUMENT
                                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                                                TST     DONE            ;TEST IF INTERRUPT DONE
                                                BNE     11$             ;YES - SKIP
                                                JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
                                                MOV     (SP)+,R3        ;GET RESULT MESSAGE POINTER
                                                TRAP    C$ERHRD
                                                .WORD   10005
                                                .WORD   0
                                                .WORD   ERR1
                                                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
                                                BR      12$

                                       11$:     TST     T.CS            ;TEST IF ANY ERROR
                                                BPL     12$             ;NO - SKIP
                                                TRAP    C$ERHRD
                                                .WORD   10006
                                                .WORD   0
                                                .WORD   ERR6
                                                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
                                       12$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
                                                MOV     (SP)+,R5        ;RESTORE REGISTERS
                                                MOV     (SP)+,R1
                                                MOV     (SP)+,R0
                                                MOV     (SP)+,R3
                                                TST     ERRSWI          ;TEST IF ERROR RETURN
                                                BEQ     13$             ;YES - SKIP
                                                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                                RTS     PC
                                       13$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                                RTS     PC
```

L6


```
                                ;       POSITION HEADS ROUTINE. POSITIONS HEADS USING 1 CYLINDER SEEKS
                                ;       TO CYLINDER SPECIFIED IN R5 BY THE CALLING ROUTINE
                                
020664  010346                  POSHDS: MOV     R3,-(SP)            ;SAVE REGS
020666  013703  003004                  MOV     SSINDX,R3           ;GET SUBROUTINE INDEX
020672  005723                          TST     (R3)+               ;BUMP IT FOR NEXT ENTRY
020674  016663  000002  002406          MOV     2(SP),SUBSTK(R3)    ;INSERT THIS CALL
020702  162763  000004  002406          SUB     #4,SUBSTK(R3)       ;ADJUST IT TO CALLING LOCATION
020710  010337  003004                  MOV     R3,SSINDX           ;STORE IT BACK
020714  010346                          MOV     R3,-(SP)
020716  010446                          MOV     R4,-(SP)
020720  012737  000002  003020          MOV     #2,ERRSWI           ;SET FOR NO ERROR RETURN
020726  004737  024054                  JSR     PC,GETPOS           ;GET CURRENT POSITION
020732  021172                          PH65$
020734  012704  000012                  MOV     #10.,R4             ;SET RETRY COUNT
020740  104404                          TRAP    C$BSEG

020742                          1$:
020742  104420                          TRAP    C$INLP
020744  103012                          BCC     2$
020746  004737  024054                  JSR     PC,GETPOS           ;ELSE GET POSITION
020752  021170                          10$
020754  023737  003106  003104          CMP     CURCYL,NEWCYL       ;CHECK IF AT INTENDED POSITION
020762  001017                          BNE     4$                  ;NO - SKIP
020764  004737  021570                  JSR     PC,ONSWAP           ;SWAP OLDCYL AND NEWCYL
020770  000414                          BR      4$                  ;SKIP

020772  013737  003106  003102  2$:     MOV     CURCYL,OLDCYL       ;IN NOT LOOPING, STORE CURCYL AS OLDCYL
021000  023705  003106                  CMP     CURCYL,R5           ;CHECK IF HDS AT FINAL POSITION
021004  001471                          BEQ     10$                 ;YES - GO TO EXIT
021006  003003                          BGT     3$                  ;IF CURCYL > FINAL POSITION - SKIP
021010  005237  003104                  INC     NEWCYL              ;ELSE BUMP NEWCYL (MOVE HDS IN)
021014  000402                          BR      4$                  ;SKIP

021016  005337  003104          3$:     DEC     NEWCYL              ;DEC NEWCYL (MOVE HDS OUT)
021022  004737  020112          4$:     JSR     PC,XSEEK            ;DO SEEK
021026  021170                          10$
021030  012701  005670                  MOV     #3000.,R1           ;SET WAIT COUNT 300 MS
021034  004737  023570                  JSR     PC,RDYWAIT          ;WAIT FOR DRIVE READY
021040  021170                          10$
021042  005737  003046                  TST     T.CS                ;TEST IF ANY ERROR
021046  100007                          BPL     5$                  ;NO - SKIP
021050  104456                          TRAP    C$ERHRD
021052  023430                          .WORD   10008
021054  000000                          .WORD   0
021056  013150                          .WORD   ERR6
021060  005037  003020                  CLR     ERRSWI              ;CLEAR FOR ERROR ERROR RETURN
021064  000441                          BR      10$

021066  004737  024054          5$:     JSR     PC,GETPOS           ;GET POSITION
021072  021170                          10$
021074  023737  003106  003104          CMP     CURCYL,NEWCYL       ;CHECK IF ARRIVED AT DESIRED PLACE
021102  001003                          BNE     7$                  ;NO - SKIP
021104  012704  000012          6$:     MOV     #10.,R4             ;ELSE INIT RETRY COUNT
021110  000714                          BR      1$                  ;GO DO NEXT SEEK

021112  005737  003112          7$:     TST     DESSGN              ;TEST IF GOING IN
```

M6


```
    54 021116  001017                        BNE     9$                ;YES - SKIP
    55 021120  023737  003106  003104        CMP     CURCYL,NEWCYL     ;CHECK IF HEADS DID NOT MOVE IN
    56 021126  003366                        BGT     6$                ;YES - SKIP
    57 021130  005304                8$:     DEC     R4                ;DEC RETRY COUNT
    58 021132  001333                        BNE     4$                ;DO ANOTHER SEEK IF NOT 0
    59 021134  012703  010003                MOV     #HDMOVF,R3        ;ELSE SET RESULT MESSAGE POINTER
    60 021140  104456                        TRAP    C$ERHRD
       021142  023431                        .WORD   10009
       021144  000000                        .WORD   0
       021146  012646                        .WORD   ERR1
    61 021150  005037  003020                CLR     ERRSWI            ;CLEAR FOR ERROR ERROR RETURN
    62 021154  000405                        BR      10$
    63
    64 021156  023737  003106  003104  9$:   CMP     CURCYL,NEWCYL     ;HDS SHOULD MOVE OUT, CHK THEY DID
    65 021164  002747                        BLT     6$                ;YES - SKIP
    66 021166  000760                        BR      8$                ;ELSE GO DEC AND RETRY
    67 021170                        10$:
       021170                        10000$:
       021170  104405                        TRAP    C$ESEG
    68
    69 021172  162737  000002  003004  PH65$: SUB     #2,SSINDX         ;REMOVE ENTRY FROM SUBROUT STACK
    70 021200  012604                        MOV     (SP)+,R4          ;RESTORE REGISTERS
    71 021202  012600                        MOV     (SP)+,R0
    72 021204  012603                        MOV     (SP)+,R3
    73 021206  005737  003020                TST     ERRSWI            ;TEST IF ERROR RETURN
    74 021212  001403                        BEQ     1$                ;YES - SKIP
    75 021214  063716  003020                ADD     ERRSWI,(SP)       ;ADD IN ERROR RETURN
    76 021220  000207                        RTS     PC
    77 021222  017616  000000        1$:     MOV     @(SP),(SP)        ;SET ERROR RETURN ADDRESS
    78 021226  000207                        RTS     PC
```

N6


```
                                ;       DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
                                ;       500MS FOR READY TO SET.
                                ;
021230  010346                  RDYCHK: MOV     R3,-(SP)        ;STORE REGS
021232  013703  003004                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
021236  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
021240  016663  000002  002406          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
021246  162763  000004  002406          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
021254  010337  003004                  MOV     R3,SSINDX       ;STORE IT BACK
021260  010046                          MOV     R0,-(SP)
021262  010146                          MOV     R1,-(SP)
021264  010446                          MOV     R4,-(SP)
021266  012737  000002  003020          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
021274  012701  011610                  MOV     #5000.,R1       ;SET WAIT COUNT
021300  004737  017214          1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
021304  021440                          4$
021306  032737  000001  003046          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
021314  001053                          BNE     5$              ;YES - EXIT
021316  012737  000001  003456          MOV     #1,XDELAY       ;SAVE ARGUMENT
021324  004737  016210                  JSR     PC,TIME         ;CALL TIMING ROUTINE
021330  005301                          DEC     R1              ;DEC WAIT COUNT
021332  001362                          BNE     1$              ;LOOP IF NOT 0
021334  012703  010702                  MOV     #MDRDY,R3       ;SET RESULT MESSAGE POINTER
021340  012704  011567                  MOV     #C500M$,R4      ;SET CONDITION MESSAGE POINTER
021344  104456                          TRAP    C$ERHRD
021346  023432                          .WORD   10010
021350  000000                          .WORD   0
021352  013100                          .WORD   ERR5
021354  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
021360  004737  017214          2$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
021364  021440                          4$
021366  032737  000001  003046          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
021374  001007                          BNE     3$              ;YES - SKIP
021376  012737  000001  003460          MOV     #1,YDELAY       ;SAVE ARGUMENT
021404  004737  016354                  JSR     PC,XTIME        ;CALL TIMING ROUTINE
021410  005301                          DEC     R1              ;DEC WAIT COUNTER
021412  001362                          BNE     2$              ;LOOP UNTIL TIME DONE
021414  032737  100000  003046  3$:     BIT     #ANYERR,T.CS    ;TEST IF ANYERR SET
021422  001406                          BEQ     4$              ;NO - SKIP
021424  104456                          TRAP    C$ERHRD
021426  023433                          .WORD   10011
021430  000000                          .WORD   0
021432  013150                          .WORD   ERR6
021434  C05337  003244                  DEC     ERRCNT          ;REDUCE ERROR COUNT FOR DUAL ERRORS
021440  005037  003020          4$:     CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
021444  162737  000002  003004  5$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
021452  012604                          MOV     (SP)+,R4        ;RESTORE REGS
021454  012601                          MOV     (SP)+,R1
021456  012600                          MOV     (SP)+,R0
021460  012603                          MOV     (SP)+,R3
021462  005737  003020                  TST     ERRSWI          ;TEST IF ERROR RETURN
021466  001403                          BEQ     6$              ;YES - SKIP
021470  063716  003020                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
021474  000207                          RTS     PC
021476  017616  000000          6$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
021502  000207                          RTS     PC
```

```
                                        ;       CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
                                        ;       SELECTED BY SOFTWARE PARAMETER.
                                        ;
021504  005037  003114                  CHOSHD: CLR     DESHD                   ;CLEAR TO HEAD 0
021510  032737  010000  014500                  BIT     #HEADLM,MISWIW          ;TEST IF HEAD SPECIFIED
021516  001403                                  BEQ     1$                      ;NO - SKIP
021520  013737  014506  003114                  MOV     HEADW,DESHD             ;INSERT SPECIFIED HEAD
021526  000207                          1$:     RTS     PC
                                        ;       SWAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
                                        ;       UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
021530  032737  010000  014500          SWAPHD: BIT     #HEADLM,MISWIW          ;TEST IF HEAD SPECIFIED
021536  001011                                  BNE     1$                      ;YES - TAKE ABORT EXIT
021540  005737  003114                          TST     DESHD                   ;TEST IF HEAD ONE USED
021544  001006                                  BNE     1$                      ;YES - TAKE ABORT EXIT
021546  012737  000001  003114                  MOV     #1,DESHD                ;ELSE SET FOR HEAD ONE
021554  062716  000002                          ADD     #2,(SP)                 ;BUMP PAST ABORT RETURN
021560  000207                                  RTS     PC                      ;RETURN
021562  017616  000000                  1$:     MOV     @(SP),(SP)              ;GET ABORT DESTINATION
021566  000207                          2$:     RTS     PC
                                        ;       SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
021570  010046                          ONSWAP: MOV     R0,-(SP)                ;STORE R0
021572  013700  003102                          MOV     OLDCYL,R0               ;MOVE OLD TO R0
021576  013737  003104  003102                  MOV     NEWCYL,OLDCYL           ;MOVE NEW TO OLD
021604  010037  003104                          MOV     R0,NEWCYL               ;PUT OLD IN NEW
021610  012600                                  MOV     (SP)+,R0                ;RESTORE R0
021612  000207                                  RTS     PC
```

```
                                ;       BAD SECTOR FILES VALID CHECK ROUTINE. CHECKS IF BAD SECTOR
                                ;       FILES HAVE BEEN READ AND STORED. IF NOT, READ BAD SECTOR
                                ;       FILES, ELSE EXIT ROUTINE.

021614  005737  003500        CKBSVD: TST     BSFVAL          ;TEST STATUS OF BAD SECTOR FILE
021620  001002                        BNE     1$              ;BR IF READ WITH ERRORS OR
                                                                ; VALID.
022622  004737  021630                JSR     PC,RDBSF        ;READ BAD SECTOR FILE
021626  000207                1$:     RTS     PC

                                ;       READ BAD SECTOR FILE ROUTINE

021630  012737  007355  003014  RDBSF:  MOV     #P2T13E,ERHEAD  ;SET ERROR HEADER
021636  012737  000001  003114          MOV     #1,DESHD        ;SET TO HEAD 1
021644  032737  010000  014500          BIT     #HEADLM,MISWIW  ;TEST IF HEAD SPEC'D
021652  001417                        BEQ     1$              ;NO - SKIP
021654  005737  014506                TST     HEADW           ;TEST IF HEAD 0
021660  001014                        BNE     1$              ;NO - SKIP, ELSE
021662  013746  003240                MOV     TSTNM,-(SP)
021666  012746  010461                MOV     #NOHD1,-(SP)
021672  012746  000002                MOV     #2.-(SP)
021676  010600                        MOV     SP,R0
021700  104417                        TRAP    C$PNTF
021702  062706  000006                ADD     #6,SP
021706  000137  022402                JMP     16$             ;EXIT

021712  013737  002304  003104  1$:   MOV     HLMTW,NEWCYL    ;POSITION HEADS AT LAST CYLINDER (BSF)
021720  004737  020112                JSR     PC,XSEEK        ;DO SEEK
021724  022350                        14$                     ;ERROR RETURN ADDRESS
021726  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
021732  004737  023570                JSR     PC,RDYWAIT      ;WAIT FOR INTERRUPT
021736  022350                        14$                     ;ERROR RETURN ADDRESS
021740  004737  024202                JSR     PC,VERPOS       ;VERIFY POSITION
021744  022350                        14$                     ;ERROR RETURN ADDRESS
021746  005037  003116                CLR     DESSEC          ;SET FOR SECTOR 0
021752  012737  003502  003132        MOV     #FCTBSF,TEMP5   ;SET TEMP STORAGE FOR FACTORY BS FILE
021760  012737  000020  003134        MOV     #16.,TEMP6      ;SET MAX SECTOR COUNT
021766  112737  000001  003451        MOVB    #1,NOERCT       ;SET FOR NO ERROR COUNTING
021774  105037  003450                CLRB    LOCERR          ;CLEAR LOCAL ERROR COUNTER
022000  005037  003126        2$:     CLR     TEMP3           ;CLEAR ONES DETECTED FLAG
022004  013701  003132                MOV     TEMP5,R1        ;INIT POINTERS
022010  013700  003134                MOV     TEMP6,R0
022014  012703  004472                MOV     #IBUFF,R3
022020  012737  000002  003020        MOV     #2,ERRSWI       ;SETUP NO ERROR SWITCH
022026  004737  025362                JSR     PC,XREAD        ;DO READ
022032  022242                        10$                     ;ERROR RETURN ADDRESS
022034  005723                        TST     (R3)+           ;TEST IF WORD 0 NOT NEG
022036  100470                        BMI     9$              ;YES, BAD FMT ERROR
022040  005723                        TST     (R3)+           ;ELSE TEST WORD 1 NOT NEG
022042  100466                        BMI     9$              ;YES - BAD FMT ERROR REPORT
022044  005723                        TST     (R3)+           ;TEST WORD 2 IS 0
022046  001064                        BNE     9$              ;NO - SKIP TO FMT ERROR RPT
022050  005723                        TST     (R3)+           ;TEST WORD 3 IS 0
022052  001062                        BNE     9$              ;NO - SKIP TO FMT ERROR RPT
022054  026327  000764  177777        CMP     764(R3),#-1     ;TEST IF NEXT TO LAST WORD IS ALL 1'S
022062  001056                        BNE     9$              ;NO  - SKIP
022064  026327  000766  177777        CMP     766(R3),#-1     ;TEST IF LAST WORD IS ALL 1'S
```

```
                                                BNE     9$              ;NO  - SKIP
                                        3$:     CMP     (R3),#-1        ;TEST IF NEXT WORD IS ALL 1'S
                                                BNE     4$              ;NO - SKIP
                                                MOV     #1,TEMP3        ;ELSE SET 1'S DETECTED FLAG
                                                CMP     (R3)+,(R3)      ;ADJUST POINTER
                                                BEQ     7$              ;BR IF THE SAME
                                        4$:     TST     TEMP3           ;TEST IF ONES HAVE BEEN DETECTED
                                                BNE     9$              ;YES - SKIP TO FMT ERROR RPT
                                                MOV     (R3)+,(R1)      ;STORE CYLINDER WORD
                                                MOV     #7,R5           ;ALIGN IT TO LOOK LIKE HEADER
                                        5$:     ASL     (R1)
                                                DEC     R5
                                                BNE     5$
                                                BIT     #BIT8,(R3)      ;TEST IF HEAD 1
                                                BEQ     6$              ;NO - SKIP
                                                BIS     #BIT6,(R1)      ;INSERT HEAD BIT
                                        6$:     BIC     #177400,(R3)    ;CLEAR ALL BUT SECTOR
                                        7$:     BIS     (R3)+,(R1)+     ;INSERT SECTOR NUMBER
                                                CMP     R3,#IBUFF+508.  ;CHECK IF IBUFF EMPTY
                                                BNE     3$              ;NO GET NEXT CYLINDER
                                                CMP     #36.,TEMP6      ;DONE CHECKING ALL BSF's YET?
                                                BEQ     15$             ;BRANCH IF YES, ELSE
                                        8$:     MOV     #FLDBSF,TEMP5   ;CHANGE POINTERS TO FIELD BS FILE
                                                MOV     #36.,TEMP6      ;MAX SECTOR NUMBER
                                                MOV     #20.,DESSEC     ;SECTOR NUMBER START
                                                BR      2$              ;DO READ

                                        9$:     TST     BSERRS          ;OUTPUT ALL BSF ERRORS?
                                                BEQ     11$             ;BRANCH IF NO
                                                MOV     #MFMTER,R3      ;SET RESULT MESSAGE POINTER
                                                TRAP    C$ERHRD
                                                .WORD   1302
                                                .WORD   0
                                                .WORD   ERR1
                                       10$:     TST     BSERRS          ;OUTPUT ALL BSF ERRORS?
                                                BEQ     11$             ;BRANCH IF NO
                                                TRAP    C$INLP
                                                BCS     2$

                                       11$:     CMP     DESSEC,TEMP6    ;CHECK IF ALL SECTORS READ
                                                BNE     13$             ;NO - SKIP
                                                INCB    LOCERR          ;BUMP LOCAL ERROR COUNTER
                                                MOV     #MFBSF,R3       ;SET ERROR MESSAGE POINTER
                                                CMP     #FLDBSF,TEMP5   ;IS THIS FIELD BS FILE?
                                                BNE     12$             ;BRANCH IF NO
                                                MOV     #MUBSF,R3       ;SET ERROR MESSAGE POINTER
                                       12$:     MOV     #-1,@TEMP5      ;TERMINATE FILE STORAGE
                                                TRAP    C$ERHRD
                                                .WORD   1301
                                                .WORD   0
                                                .WORD   ERR1
                                                CMP     #FLDBSF,TEMP5   ;DID WE CHECK FIELD BS FILE YET?
                                                BEQ     15$             ;BRANCH IF YES, ELSE
                                                BR      8$              ;GO CHECK FIELD BSF

                                       13$:     ADD     #4,DESSEC       ;BUMP TO NEXT SECTOR
                                                BR      2$              ;GO DO READ
```

```
104
105 022350  105237  003450          14$:    INCB    LOCERR          ;INC LOCAL ERROR COUNT
106 022354  012737  000002  003020  15$:    MOV     #2,ERRSWI       ;SETUP FOR NO ERROR RETURN
107 022362  012737  000001  003500          MOV     #1,BSFVAL       ;SET BAD SEC FILE VALID FLAG
108 022370  105737  003450                  TSTB    LOCERR          ;TEST IF LOCAL ERRORS
109 022374  001454                          BEQ     17$             ;NO - SKIP
110 022376  005237  003244                  INC     ERRCNT          ;BUMP ERROR COUNT
111 022402  012737  177777  003500  16$:    MOV     #-1,BSFVAL      ;SET BAD READ OR INVALID BAD SEC FILE
112 022410  012746  010572                  MOV     #BSFNOT,-(SP)
    022414  012746  000001                  MOV     #1,-(SP)
    022420  010600                          MOV     SP,R0
    022422  104417                          TRAP    C$PNTF
    022424  062706  000004                  ADD     #4,SP
113 022430  005046                          CLR     -(SP)
    022432  153716  003035                  BISB    RLDRV+1,(SP)
    022436  012746  006621                  MOV     #DRVNAM,-(SP)
    022442  013746  003030                  MOV     RLBAS,-(SP)
    022446  012746  006610                  MOV     #BASADD,-(SP)
    022452  012746  011750                  MOV     #FMT5,-(SP)
    022456  012746  000005                  MOV     #5,-(SP)
    022462  010600                          MOV     SP,R0
    022464  104417                          TRAP    C$PNTF
    022466  062706  000014                  ADD     #14,SP
114 022472  012746  011623                  MOV     #CRLF,-(SP)
    022476  012746  000001                  MOV     #1,-(SP)
    022502  010600                          MOV     SP,R0
    022504  104417                          TRAP    C$PNTF
    022506  062706  000004                  ADD     #4,SP
115 022512  012737  177777  003502          MOV     #-1,FCTBSF      ;TERMINATE FACTORY BSF LIST
116 022520  012737  177777  004076          MOV     #-1,FLDBSF      ;TERMINATE FIELD BSF LIST
117 022526  000207                  17$:    RTS     PC              ;RETURN
```

F7


```
                                        ;       READ HEADERS ROUTINE.
                                        ;
022530  012737  000001  003130  XRDHDC: MOV     #1,TEMP4        ;SET FLAG TO BYPASS REG STORAGE
022536  000402                          BR      XRDHDG          ;GO DO IT
                                        ;
022540  005037  00313(          XRDHD:  CLR     TEMP4           ;SET FLAG TO SAVE T. AMD L. REGS
022544  010346                  XRDHDG: MOV     R3,-(SP)        ;STORE REGISTERS
022546  013703  003004                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
022552  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
022554  016663  000002  002406          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
022562  162763  000004  002406          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
022570  010337  003004                  MOV     R3,SSINDX       ;STORE IT BACK
022574  010046                          MOV     R0,-(SP)
022576  010146                          MOV     R1,-(SP)
022600  010446                          MOV     R4,-(SP)
022602  012737  000002  003020          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
022610  005737  003130                  TST     TEMP4           ;TEST IF REGISTERS TO BE SAVED
022614  001007                          BNE     2$              ;NO - SKIP
022616  012703  003046                  MOV     #L.MP+2,R3      ;SET POINTER FOR REGS
022622  012701  000004                  MOV     #4,R1           ;SET COUNT
022626  014346                  1$:     MOV     -(R3),-(SP)     ;SAVE REGISTER
022630  005301                          DEC     R1              ;DEC COUNT
022632  001375                          BNE     1$              ;LOOP UNTIL ALL ARE SAVED
022634  004737  021230          2$:     JSR     PC,RDYCHK       ;CHECK DRIVE READY
022640  023110                          11$
022642  005037  003010                  CLR     DONE            ;CLEAR INTERRUPT FLAG
022646  012701  003036                  MOV     #L.CS,R1        ;GET ADDRESS OF LOAD REGS
022652  013711  003034                  MOV     RLDRV,(R1)      ;LOAD DRIVE NUMBER
022656  042711  002000                  BIC     #BIT10,(R1)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
022662  052721  000110                  BIS     #RDHEAD,(R1)+   ;INSERT COMMAND
022666  005021                          CLR     (R1)+           ;CLEAR BA
022670  005021                          CLR     (R1)+           ;CLEAR DA
022672  014162  000004                  MOV     -(R1),RLDA(R2)  ;LOAD RL11 REGS
022676  014162  000002                  MOV     -(R1),RLBA(R2)
022702  014162  000000                  MOV     -(R1),RLCSR(R2)
022706                          3$:
022720  005737  003010                  TST     DONE            ;TEST IN INTERRUPT FLAG SET
022724  001460                          BEQ     9$              ;NO - SKIP
022726  032737  000001  003046  4$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
022734  001035                          BNE     7$              ;YES - SKIP
022736  012703  010702                  MOV     #MDRDY,R3       ;SET NO READY MESSAGE
022742  012704  011604                  MOV     #CAFDT,R4       ;CONDITION OF AFTER DATA XFER
022746  104456                          TRAP    C$ERHRD
022750  023441                          .WORD   10017
022752  000000                          .WORD   0
022754  013100                          .WORD   ERR5
022756  012701  000062                  MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
022762  004737  017214          5$:     JSR     PC,GSTAT        ;GET STATUS
022766  023104                          10$
022770  032737  000001  003046          BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE HAS COME READY
022776  001403                          BEQ     6$              ;NO - SKIP
023000  005037  003020                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
023004  000411                          BR      7$              ;SKIP
                                        ;
023006  005301                  6$:     DEC     R1              ;DEC WAIT COUNT
023010  001364                          BNE     5$              ;LOOP UNTIL TIME DONE
023012  012704  011615                  MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECONDS
```

```
 55 023016  104456                    TRAP    C$ERHRD
    023020  023436                    .WORD   10014
    023022  000000                    .WORD   0
    023024  013100                    .WORD   ERR5
 56 023026  000426                    BR      10$             ;EXIT
 57
 58 023030  005737  003046    7$:     TST     T.CS            ;CHECK FOR ANY ERRORS
 59 023034  100005                    BPL     8$              ;NO - SKIP
 60 023036  104456                    TRAP    C$ERHRD
    023040  023440                    .WORD   10016
    023042  000000                    .WORD   0
    023044  013150                    .WORD   ERR6
 61 023046  000416                    BR      10$
 62
 63 023050  012701  003056    8$:     MOV     #HDWRD2,R1      ;GET POINTER
 64 023054  016221  000006            MOV     RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
 65 023060  016221  000006            MOV     RLMP(R2),(R1)+
 66 023064  000411                    BR      11$             ;EXIT
 67
 68 023066  004737  017010    9$:     JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
 69 023072  012603                    MOV     (SP)+,R3        ;GET RESULTS
 70 023074  104456                    TRAP    C$ERHRD
    023076  023437                    .WORD   10015
    023100  000000                    .WORD   0
    023102  012646                    .WORD   ERR1
 71 023104  005037  003020    10$:    CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
 72 023110  005737  003130    11$:    TST     TEMP4           ;TEST IF REGISTERS WERE SAVED
 73 023114  001007                    BNE     13$             ;NO - SKIP
 74 023116  012703  003036            MOV     #L.CS,R3        ;SET POINTER TO RESTORE REGS
 75 023122  012701  000004            MOV     #4,R1           ;SET COUNT
 76 023126  012623          12$:     MOV     (SP)+,(R3)+     ;RESTORE REGISTER
 77 023130  005301                    DEC     R1              ;DEC COUNT
 78 023132  001375                    BNE     12$             ;LOOP UNTIL ALL ARE RESTORED
 79 023134  162737  000002  003004  13$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
 80 023142  012604                    MOV     (SP)+,R4        ;RESTORE REGS
 81 023144  012601                    MOV     (SP)+,R1
 82 023146  012600                    MOV     (SP)+,R0
 83 023150  012603                    MOV     (SP)+,R3
 84 023152  005737  003020            TST     ERRSWI          ;TEST IF ERROR RETURN
 85 023156  001403                    BEQ     14$             ;YES - SKIP
 86 023160  063716  003020            ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
 87 023164  000207                    RTS     PC
 88 023166  017616  000000    14$:    MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
 89 023172  000207                    RTS     PC
```

H7


```
                                ;       VERIFY HEADERS ROUTINE. COMPARES 40 HEADERS FOR CONTENT AND
                                ;       SEQUENCE.
                                ;
023174  010346                  VERHDR: MOV     R3,-(SP)        ;STORE REGS
023176  013703  003004                  MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
023202  005723                          TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
023204  016663  000002  002406          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
023212  162763  000004  002406          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
023220  010337  003004                  MOV     R3,SSINDX       ;STORE IT BACK
023224  010046                          MOV     R0,-(SP)
023226  010146                          MOV     R1,-(SP)
023230  010446                          MOV     R4,-(SP)
023232  010546                          MOV     R5,-(SP)
023234  012737  000002  003020          MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
023242  052737  000002  003006          BIS     #HDRCMP,OPFLAG  ;SET HEADER COMPARE FLAG
023250  005037  003016                  CLR     MORECE          ;CLEAR MORE ERRORS FLAG
023254  012704  004472                  MOV     #IBUFF,R4       ;SET POINTER TO HEADERS
023260  012705  003120                  MOV     #TEMP0,R5       ;SET POINTER TO WORK AREA
023264  005003                          CLR     R3              ;CLEAR FOR WORD COUNTER
023266  011415                          MOV     (R4),(R5)       ;MOVE HDR WORD TO WORK AREA
023270  011401                          MOV     (R4),R1         ;PUT WORD IN REG 1
023272  042701  000177                  BIC     #177,R1 ;CLEAR ALL BUT CYLINDER
023276  012700  000007                  MOV     #7,R0           ;SET SHIFT COUNT
023302  006201                  1$:     ASR     R1              ;SHIFT
023304  005300                          DEC     R0              ;DEC
023306  001375                          BNE     1$              ;LOOP
023310  020137  003104                  CMP     R1,NEWCYL       ;CHECK IF CYLINDER PART GOOD
023314  001407                          BEQ     2$              ;YES - SKIP
023316  104456                          TRAP    C$ERHRD
023320  023442                          .WORD   10018
023322  000000                          .WORD   0
023324  014242                          .WORD   ERR10
023326  005037  003020                  CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
023332  000456                          BR      8$

023334  012701  000050          2$:     MOV     #40.,R1         ;SET HEADER COUNT
023340  042715  000100                  BIC     #HDHSEL,(R5)    ;CLEAR HEAD SELECT AND 0 BIT
023344  005737  003114                  TST     DESHD           ;ARE WE USING HD 0?
023350  001402                          BEQ     3$              ;YES - SKIP
023352  052715  000100                  BIS     #HDHSEL,(R5)    ;INSERT HEAD BIT
023356  005065  000002          3$:     CLR     2(R5)           ;CLEAR 2ND WORD OF WORK AREA
023362  021524                  4$:     CMP     (R5),(R4)+      ;TEST FIRST WORD OK
023364  001410                          BEQ     5$              ;YES - SKIP
023366  005744                          TST     -(R4)           ;ELSE SET POINTER FOR ERROR
023370  104456                          TRAP    C$ERHRD
023372  023442                          .WORD   10018
023374  000000                          .WORD   0
023376  014242                          .WORD   ERR10
023400  005037  003020                  CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
023404  005724                          TST     (R4)+           ;RESET POINTER
023406  005203                  5$:     INC     R3              ;BUMP WORD COUNTER
023410  005724                          TST     (R4)+           ;TEST 2ND WORD IS 0
023412  001410                          BEQ     6$              ;YES - SKIP
023414  022544                          CMP     (R5)+,-(R4)     ;ADJUST POINTERS FOR REPORT
023416  104456                          TRAP    C$ERHRD
023420  023442                          .WORD   10018
023422  000000                          .WORD   0
```

```
                023424  014242                          .WORD   ERR10
     50 023426  005037  003020                          CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
     51 023432  024524                                  CMP     -(R5),(R4)+     ;RESET POINTERS
     52 023434  005724                          6$:     TST     (R4)+           ;BUMP PAST ECC WORD
     53 023436  005203                                  INC     R3              ;BUMP WORD COUNTER
     54 023440  005215                                  INC     (R5)            ;BUMP SECTOR OF EXPECTED HEADER
     55 023442  011500                                  MOV     (R5),R0         ;MOVE EXPECTED HDR TO R0
     56 023444  042700  177700                          BIC     #^CHDSEC,R0     ;CLEAR ALL BUT SECTOR
     57 023450  022700  000050                          CMP     #40.,R0         ;TEST IF AT SECTOR 40
     58 023454  001002                                  BNE     7$              ;NO - SKIP
     59 023456  042715  000077                          BIC     #HDSEC,(R5)     ;CLEAR SECTOR TO 0
     60 023462  005203                          7$:     INC     R3              ;BUMP HDR WORD COUNTER
     61 023464  005301                                  DEC     R1              ;DEC HEADER COUNT
     62 023466  001335                                  BNE     4$              ;LOOP IF NOT YET DONE
     63 023470  162737  000002  003004          8$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
     64 023476  012605                                  MOV     (SP)+,R5        ;RESTORE REGISTERS
     65 023500  012604                                  MOV     (SP)+,R4
     66 023502  012601                                  MOV     (SP)+,R1
     67 023504  012600                                  MOV     (SP)+,R0
     68 023506  012603                                  MOV     (SP)+,R3
     69 023510  005737  003020                          TST     ERRSWI          ;TEST IF ERROR RETURN
     70 023514  001403                                  BEQ     9$              ;YES - SKIP
     71 023516  063716  003020                          ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
     72 023522  000207                                  RTS     PC
     73 023524  017616  000000                  9$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
     74 023530  000207                                  RTS     PC
     76
     77                                         ;       POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
     78
     79 023532  013705  003054                  POSHW1: MOV     HDWRD1,R5       ;START FOR POSITION HD BIT IN WD 1
     80 023536  000402                                  BR      POSHDO          ;SKIP
     81
     82 023540  013705  003054                  POSHSB: MOV     T.MP,R5         ;START FOR POSITION HD BIT IN MP
     83 023544  010146                          POSHDO: MOV     R1,-(SP)        ;STORE R1
     84 023546  042705  177677                          BIC     #^CHSSTAT,R5    ;CLEAR ALL BUT HEAD SEL BIT
     85 023552  012701  000006                          MOV     #6,R1           ;SET SHIFT COUNT
     86 023556  006205                          1$:     ASR     R5              ;SHIFT FOR RIGHT JUSTIFY
     87 023560  005301                                  DEC     R1
     88 023562  001375                                  BNE     1$
     89 023564  012601                                  MOV     (SP)+,R1        ;RESTORE R1
     90 023566  000207                                  RTS     PC              ;RETURN
     91
     92                                         ;       WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
     93                                         ;       FROM THE CALLING ROUTINE IN R1.
     94 023570  010346                          RDYWAIT:        MOV     R3,-(SP)        ;STORE R3
     95 023572  013703  003004                          MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
     96 023576  005723                                  TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
     97 023600  016663  000002  002406                  MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
     98 023606  162763  000004  002406                  SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
     99 023614  010337  003004                          MOV     R3,SSINDX       ;STORE IT BACK
    100 023620  010046                                  MOV     R0,-(SP)
    101 023622  010146                                  MOV     R1,-(SP)
    102 023624  010446                                  MOV     R4,-(SP)
    103 023626  012737  000002  003020                  MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
    104 023634  004737  017214                  1$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
    105 023640  024010                                  6$
    106 023642  032737  000001  003046                  BIT     #DRDYMSK,T.CS   ;CHECK IF READY
```

J7


```
                                                MOV     #1,XDELAY       ;SAVE ARGUMENT
                                                JSR     PC,TIME         ;CALL TIMING ROUTINE
                                                BR      1$
                                        2$:     MOV     #MDRDY,R3       ;SET NAME MESSAGE PTR
                                                TRAP    C$ERHRD
                                                .WORD   10020
                                                .WORD   0
                                                .WORD   ERR3
                                                MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECONDS
                                        3$:     JSR     PC,GSTAT        ;GET DRIVE STATUS
                                                6$
                                                BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
                                                BNE     4$              ;YES - SKIP
                                                MOV     #1,YDELAY       ;SAVE ARGUMENT
                                                JSR     PC,XTIME        ;CALL TIMING ROUTINE
                                                DEC     R1              ;DEC WAIT COUNT
                                                BNE     3$              ;LOOP UNTIL TIME DONE
                                                MOV     #C5SEC,R4       ;SET CONDITION AFTER 5 SECDS
                                                TRAP    C$ERHRD
                                                .WORD   10021
                                                .WORD   0
                                                .WORD   ERR5
                                                BR      5$              ;EXIT

                                        4$:     BIT     #ANYERR,T.CS    ;TEST IF ANY ERROR SET
                                                BEQ     6$              ;NO - SKIP
                                                TRAP    C$ERHRD
                                                .WORD   10022
                                                .WORD   0
                                                .WORD   ERR6
                                        5$:     DEC     ERRCNT          ;DEC FOR DOUBLE ERROR REPORT
                                        6$:     CLR     ERRSWI          ;CLEAR FOR ERROR ERROR RETURN
                                        7$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
                                                MOV     (SP)+,R4        ;RESTORE REGISTERS
                                                MOV     (SP)+,R1
                                                MOV     (SP)+,R0
                                                MOV     (SP)+,R3        ;RESTORE R3
                                                TST     ERRSWI          ;TEST IF ERROR RETURN
                                                BEQ     8$              ;YES - SKIP
                                                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                                RTS     PC
                                        8$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                                RTS     PC

                                        ;       GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
                                        ;       (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
                                        ;       NUMBER IN CURCYL.
                                        GETPOS: MOV     R3,-(SP)        ;STORE REGISTERS
                                                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
                                                TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
                                                MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
                                                SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
                                                MOV     R3,SSINDX       ;STORE IT BACK
```

```
                                                MOV     R0,-(SP)
                                                MOV     R5,-(SP)
                                                JSR     PC,XRDHD        ;DO READ HEADER
                                                2$
                                                MOV     HDWRD1,R3       ;GET HEADER WORD
                                                MOV     #7,R5           ;SET SHIFT COUNT
                                        1$:     ASR     R3              ;SHIFT TO RIGHT JUSTIFY
                                                DEC     R5
                                                BNE     1$
                                                BIC     #177000,R3
                                                MOV     R3,CURCYL       ;STORE AS CURRENT CYLINDER
                                        2$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
                                                MOV     (SP)+,R5        ;RESTORE REGISTERS
                                                MOV     (SP)+,R0
                                                MOV     (SP)+,R3
                                                TST     ERRSWI          ;TEST IF ERROR RETURN
                                                BEQ     3$              ;YES - SKIP
                                                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                                RTS     PC
                                        3$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                                RTS     PC

                                        ;       VERIFY POSITION ROUTINE. READS A HEADER (USING GETPOS) AND
                                        ;       CHECKS HEADS ARE POSITIONED AT NEW CYLINDER (CURCYL = NEWCYL).

                                        VERPOS: MOV     R3,-(SP)        ;STORE R3
                                                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
                                                TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
                                                MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
                                                SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
                                                MOV     R3,SSINDX       ;STORE IT BACK

                                                MOV     #2,ERRSWI       ;SET FOR NO ERROR RETURN
                                                JSR     PC,GETPOS       ;GET POSITION
                                                1$
                                                CMP     NEWCYL,CURCYL   ;CHECK IF CURRENT CYL IS NEW CYL
                                                BEQ     1$              ;YES - SKIP
                                                TRAP    C$ERHRD
                                                .WORD   10022
                                                .WORD   0
                                                .WORD   ERR8
                                                CLR     ERRSWI          ;CLEAR FOR ERROR RETURN
                                        1$:     SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
                                                MOV     (SP)+,R3        ;RESTORE R3
                                                TST     ERRSWI          ;TEST IF ERROR RETURN
                                                BEQ     2$              ;YES - SKIP
                                                ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                                RTS     PC
                                        2$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                                RTS     PC

                                        ;       READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
                                        ;       IN IBUFF.

                                        RDALHD: MOV     R3,-(SP)        ;STORE REGISTERS
                                                MOV     SSINDX,R3       ;GET SUBROUTINE INDEX
                                                TST     (R3)+           ;BUMP IT FOR NEXT ENTRY
```

L7


```
                                                ;INSERT THIS CALL
                                                ;ADJUST IT TO CALLING LOCATION
                                                ;STORE IT BACK
                                                ;SET FOR NO ERROR RETURN
                                                ;SET HEADER COUNT
                                                ;SET 40 HDR OP FLAG
                                                ;SET POINTER TO STORE HDRS
                                                ;GET BASE ADDRESS
                                                ;MAKE IT POINT TO MP REG
                                                ;LOAD FOR READ HEADER, NO INTERRUPT
                                                ;INSERT DRIVE NUMBER
                                                ;CLEAR FOR DRIVE 4 - 7 SPEC'D
                                                ;CLEAR BA
                                                ;CLEAR DA
                                                ;TEST IF HEAD 0
                                                ;YES - SKIP
                                                ;ELSE INSERT HEAD 0
                                                ;LOAD RLDA REG
                                                ;LOAD RLBA
                                                ;TEST IF CONTROLLER READY
                                                ;YES - SKIP
                                                ;ELSE CHECK READY
                                                ;LOAD RLCS REG
                                                ;SET COUNT FOR WAIT
                                                ;CHECK THAT OPERATION COMPLETED
                                                ;YES - SKIP
                                                ;DEC COUNT
                                                ;SKIP IF NOT YET 0
                                                ;ELSE GET ALL REGISTERS
                                                ;ELSE WAIT FOR TIMEOUT
                                                ;GET RESULT MESSAGE POINTER
                                                ;CLEAR FOR ERROR RETURN
                                                ;TEST FOR ANY ERRORS
                                                ;NO - SKIP
                                                ;CLEAR FOR ERROR RETURN
                                                ;STORE HEADER WORDS
                                                ;DEC HEADER COUNT
                                                ;REMOVE ENTRY FROM SUBROUT STACK
```

```
                                        MOV     (SP)+,R4        ;RESTORE REGISTERS
                                        MOV     (SP)+,R1
                                        MOV     (SP)+,R0
                                        MOV     (SP)+,R3
                                        TST     ERRSWI          ;TEST IF ERROR RETURN
                                        BEQ     7$              ;YES - SKIP
                                        ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
                                        RTS     PC
                                7$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
                                        RTS     PC

                                ;       GENERATE DATA ROUTINE. PATTERN TO BE GENERATED IS GIVEN
                                ;       IN THE WORD FOLLOWING THE CALL. 128 WORDS ARE GENERATED
                                ;       IN OBUFF.

                                DATGEN: MOV     R1,-(SP)        ;STORE REGISTERS
                                        MOV     R3,-(SP)
                                        MOV     R4,-(SP)
                                        MOV     #OBUFF,R1       ;SET POINTER TO OBUFF
                                        MOV     (R5)+,R4        ;GET DATA PATTERN SELECTOR
                                        ASL     R4              ;ADJUST IT FOR INDEXING
                                        MOV     PATTBL(R4),R3   ;GET ADDRESS OF PATTERN
                                        MOV     (R3),(R1)+      ;MOVE FIRST PATTERN WORD
                                        BEQ     5$              ;SKIP IF PATTERN IS 0
                                        CMP     (R3),#-1        ;CHECK IF PATTERN IS ALL 1'S
                                        BEQ     5$              ;YES - SKIP
                                        CMP     R4,#8.          ;TEST IF PATTERN 5
                                        BEQ     3$              ;YES - SKIP
                                        CMP     R4,#16.         ;CHECK IF PATTERN 9 OR 10
                                        BLT     6$              ;NO - SKIP
                                3$:     TST     (R3)+           ;BUMP SOURCE POINTER
                                        MOV     (R3)+,(R1)+     ;MOVE TWO MORE WORDS FORM SOURCE
                                        MOV     (R3)+,(R1)+
                                        MOV     #13.,R4         ;SET COUNT
                                        MOV     #OBUFF,R3       ;RESET POINTER
                                        BR      8$

                                5$:     MOV     #OBUFF,R3       ;ELSE SET OBUFF AS PATTERN SOURCE
                                        BR      7$              ;GO TO FILL

                                6$:     TST     (R3)+           ;BUMP SOURCE POINTER
                                7$:     MOV     #15.,R4         ;SET MOVE COUNT
                                8$:     MOV     (R3)+,(R1)+     ;MOVE 15 WORDS INTO BUFFER
                                        DEC     R4
                                        BNE     8$
                                        MOV     #OBUFF,R3       ;SET SOURCE TO TOP OF OBUFF
                                        MOV     #112.,R4        ;SET COUNT FOR REST OF BUFFER
                                10$:    MOV     (R3)+,(R1)+     ;REPEAT PATTERN IN BUFFER
                                        DEC     R4
                                        BNE     10$
                                        MOV     (SP)+,R4        ;RESTORE REGISTERS
                                        MOV     (SP)+,R3
                                        MOV     (SP)+,R1
                                        RTS     R5              ;RETURN
```

```
                                ;       DATA COMPARE ROUTINE. COMPARES THE CONTENTS OF IBUFF AND OBUFF.
                                ;       ERROR REPORTING IS LIMITED BY SOFTWARE PARAMETER.
                                ;
025032  010346                  DATCOM: MOV     R3,-(SP)        ;STORE R3
025034  013703  003004                  MOV     $SINDX,R3       ;GET SUBROUTINE STACK INDEX
025040  005723                          TST     (R3)+           ;BUMP INDEX TO NEXT ENTRY
025042  016663  000002  002406          MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
025050  162763  000004  002406          SUB     #4,SUBSTK(R3)   ;ADJUST IT TO CALLING LOCATION
025056  010337  003004                  MOV     R3,$SINDX       ;STORE IT BACK
025062  010146                          MOV     R1,-(SP)        ;STORE OTHER REGISTERS
025064  010446                          MOV     R4,-(SP)
025066  010546                          MOV     R5,-(SP)
025070  052737  000001  003006          BIS     #DATACMP,OPFLAG ;SET DATA COMPARE FLAG
025076  005037  003016                  CLR     MORECE          ;CLEAR MORE ERROR FLAG
025102  012705  005072                  MOV     #OBUFF,R5       ;SET POINTERS TO DATA FOR COMPARE
025106  012704  004472                  MOV     #IBUFF,R4
025112  012703  000001                  MOV     #1,R3           ;SET WORD COUNTER
025116  012701  000200                  MOV     #128.,R1        ;SET COMPARE COUNT
025122  022425                  1$:     CMP     (R4)+,(R5)+     ;COMPARE DATA
025124  001052                          BNE     6$              ;ERROR - SKIP TO REPORT
025126  005203                  2$:     INC     R3              ;BUMP WORD COUNT
025130  005301                          DEC     R1              ;DEC COMPARE COUNT
025132  001373                          BNE     1$              ;LOOP IF NOT 0
025134  042737  000001  003006  3$:     BIC     #DATACMP,OPFLAG ;CLEAR DATA COMPARE FLAG
025142  005737  003020                  TST     ERRSWI          ;TEST IF ANY COMPARE ERRORS
025146  001021                          BNE     4$              ;NO - SKIP
025150  012701  000200                  MOV     #128.,R1        ;SET REPORT VALUE
025154  010146                          MOV     R1,-(SP)
025156  012746  011521                  MOV     #RESE6,-(SP)
025162  013746  003016                  MOV     MORECE,-(SP)
025166  012746  010230                  MOV     #TCERR,-(SP)
025172  012746  012615                  MOV     #FMT27,-(SP)
025176  012746  000005                  MOV     #5,-(SP)
025202  010600                          MOV     SP,R0
025204  104414                          TRAP    C$PNTB
025206  062706  000014                  ADD     #14,SP
025212  162737  000002  003004  4$:     SUB     #2,$SINDX       ;REMOVE ENTRY FROM SUBROUT STACK
025220  012605                          MOV     (SP)+,R5        ;RESTORE REGS
025222  012604                          MOV     (SP)+,R4
025224  012601                          MOV     (SP)+,R1
025226  012603                          MOV     (SP)+,R3
025230  005737  003020                  TST     ERRSWI          ;TEST IF ERROR RETURN
025234  001403                          BEQ     5$              ;YES - SKIP
025236  063716  003020                  ADD     ERRSWI,(SP)     ;ADD IN ERROR RETURN
025242  000207                          RTS     PC
025244  017616  000000          5$:     MOV     @(SP),(SP)      ;SET ERROR RETURN ADDRESS
025250  000207                          RTS     PC
025252  023737  003016  014512  6$:     CMP     MORECE,DCLIMW   ;TEST IF COMPARE ERRORS LIMIT EXCEEDED
025260  002011                          BGE     7$              ;YES - SKIP
025262  024445                          CMP     -(R4),-(R5)     ;SET PTRS BACK TO ERROR WORDS
025264  104456                          TRAP    C$ERHRD
025266  023463                          .WORD   10035
025270  000000                          .WORD   0
025272  014242                          .WORD   ERR10
025274  005037  003020                  CLR     ERRSWI          ;CLEAR ERROR SWITCH
025300  022425                          CMP     (R4)+,(R5)+     ;BUMP PTRS PAST ERROR WORDS
025302  000711                          BR      2$              ;DO NEXT COMPARE
```

```
47
48 025304  005237  003016          7$:     INC     MORECE          ;BUMP ERROR COUNTER
49 025310  000706                          BR      2$              ;DO NEXT COMPARE
```

```
                                        ;       WRITE AND READ DATA ROUTINE.
                                        ;
025312 012737 177777 003122 XWRITT: MOV     #-1,TEMP1       ;SET SPECIAL WRITE FOR TIMING FLAG
025320 000402               BR      XWRITI
                                        ;
025322 005037 003122        XWRITE: CLR     TEMP1           ;CLEAR SPECIAL WRITE FLAG
025326 012737 000112 003136 XWRITI: MOV     #WTDATA,TEMP7   ;SET FOR WRITE
025334 023737 002304 003106         CMP     HLMTW,CURCYL    ;TEST IF CYLINDER MAX (BAD SEC)
025342 001006               BNE     1$              ;NO - SKIP
025344 005737 003114        TST     DESHD           ;TEST IF HEAD 1 (BAD SECTOR FILES)
025350 001403               BEQ     1$              ;NO - SKIP
025352 052737 004000 003006 BIS     #BADADD,OPFLAG  ;SET BAD ADDRESS FLAG
025360 000403               1$:     BR      XREADG          ;SKIP TO EXECUTE
                                        ;
025362 012737 000114 003136 XREAD:  MOV     #RDDATA,TEMP7   ;SET FOR READ
025370 010346               XREADG: MOV     R3,-(SP)        ;STORE R3
025372 013703 003004                MOV     SSINDX,R3       ;SET SUBROUTINE INDEX
025376 005723               TST     (R3)+           ;BUMP TO NEXT STACK ENTRY
025400 016663 000002 002406         MOV     2(SP),SUBSTK(R3) ;INSERT THIS CALL
025406 162763 000004 002406         SUB     #4,SUBSTK(R3)   ;ADJUST TO POINT TO CALL
025414 010337 003004                MOV     R3,SSINDX       ;STORE IT BACK
025420 010046               MOV     R0,-(SP)
025422 010146               MOV     R1,-(SP)        ;STORE OTHER REGISTERS
025424 010446               MOV     R4,-(SP)
025426 004737 021230                JSR     PC,RDYCHK       ;CHECK IF DRIVE READY
025432 026064               14$
025434 012703 003036                MOV     #L.CS,R3        ;GET ADDRESS OF LOAD REGS
025440 013713 003136                MOV     TEMP7,(R3)      ;SET COMMAND
025444 053713 003034                BIS     RLDRV,(R3)      ;INSERT DRIVE NUMBER
025450 042713 002000                BIC     #BIT10,(R3)     ;CLEAR FOR DRIVE 4 - 7 SPEC'D
025454 032723 000004                BIT     #BIT2,(R3)+     ;TEST IF WRITE DATA
025460 001403               BEQ     1$              ;YES - SKIP
025462 012723 004472                MOV     #IBUFF,(R3)+    ;ELSE SET BA FOR READ
025466 000402               BR      2$
                                        ;
025470 012723 005072        1$:     MOV     #OBUFF,(R3)+    ;SET BA FOR WRITE
025474 013713 003106        2$:     MOV     CURCYL,(R3)     ;GET CURRENT CYLINDER
025500 012704 000007                MOV     #7,R4           ;ALIGN IT IN DA
025504 006313               3$:     ASL     (R3)
025506 005304               DEC     R4
025510 001375               BNE     3$
025512 005737 003114        TST     DESHD           ;TEST IF HEAD 0
025516 001402               BEQ     4$              ;YES - SKIP
025520 052713 000100                BIS     #HSMSK,(R3)     ;SET FOR HEAD 1
025524 053723 003116        4$:     BIS     DESSEC,(R3)+    ;INSERT DESIRED SECTOR
025530 012713 177600                MOV     #177600,(R3)    ;INSERT WORD COUNT
025534 023737 003106 002304         CMP     CURCYL,HLMTW    ;IS THIS BSF CYLINDER?
025542 001004               BNE     5$              ;NO - SKIP
025544 005737 003114        TST     DESHD           ;TEST IF HEAD 1
025550 001401               BEQ     5$              ;NO - SKIP
025552 006313               ASL     (R3)            ;MAKE WORD COUNT 2 SECTORS
025554 005737 003122        5$:     TST     TEMP1           ;CHECK IF SPECIAL WRITE FOR TIMING
025560 001402               BEQ     6$              ;NO - SKIP
025562 012713 177777                MOV     #177777,(R3)    ;ELSE SET FOR 1 WORD TRANSFER
025566 032737 004000 003006 6$:     BIT     #BADADD,OPFLAG  ;TEST IF BAD ADDRESS FLAG SET
025574 001414               BEQ     7$              ;NO - SKIP
025576 042737 173777 003006         BIC     #^CBADADD,OPFLAG  ;CLEAR ALL BUT THIS FLAG
```

```
    58 025604  012703  011423                    MOV     #MWRTAB,R3      ;SET RESULT MESSAGE POINTER
    59 025610  104456                            TRAP    C$ERHRD
       025612  023460                            .WORD   10032
       025614  000000                            .WORD   0
       025616  012646                            .WORD   ERR1
    60 025620  005037  003006                    CLR     OPFLAG          ;CLEAR ALL FLAGS
    61 025624  000515                            BR      13$
    62
    63 025626  005037  003010            7$:     CLR     DONE            ;CLEAR INTERRUPT FLAG
    64 025632  005737  003122                    TST     TEMP1           ;CHECK IF SPECIAL WRITE FLAG SET
    65 025636  001112                            BNE     14$             ;YES - DO NOT START WRITE
    66 025640  011362  000006                    MOV     (R3),RLMP(R2)   ;LOAD RL REGS
    67 025644  014362  000004                    MOV     -(R3),RLDA(R2)
    68 025650  014362  000002                    MOV     -(R3),RLBA(R2)
    69 025654  014362  000000                    MOV     -(R3),RLCS(R2)
    70 025660                            8$:
       025660  012737  005670  003456            MOV     #3000.,XDELAY   ;SAVE ARGUMENT
       025666  004737  016210                    JSR     PC,TIME         ;CALL TIMING ROUTINE
    71 025672  005737  003010                    TST     DONE            ;CHECK IF INTERRUPT
    72 025676  001010                            BNE     9$              ;YES - SKIP
    73 025700  004737  017010                    JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
    74 025704  012603                            MOV     (SP)+,R3        ;GET RESULT MESSAGE
    75 025706  104456                            TRAP    C$ERHRD
       025710  023456                            .WORD   10030
       025712  000000                            .WORD   0
       025714  012646                            .WORD   ERR1
    76 025716  000460                            BR      13$
    77
    78 025720  032737  000001  003046    9$:     BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY
    79 025726  001033                            BNE     11$             ;YES - SKIP
    80 025730  012703  010702                    MOV     #MDRDY,R3       ;SET RESULT MESSAGE
    81 025734  012704  011604                    MOV     #CAFDT,R4       ;CONDITION AFTER DATA XFER
    82 025740  104456                            TRAP    C$ERHRD
       025742  023460                            .WORD   10032
       025744  000000                            .WORD   0
       025746  013100                            .WORD   ERR5
    83 025750  012701  000062                    MOV     #50.,R1         ;SET WAIT COUNT FOR 5 SECDS
    84 025754  004737  017214            10$:    JSR     PC,GSTAT        ;GET DRIVE STATUS
    85 025760  026060                            13$
    86 025762  032737  000001  003046            BIT     #DRDYMSK,T.CS   ;TEST IF DRIVE READY NOW
    87 025770  001012                            BNE     11$             ;YES - SKIP
    88 025772  005301                            DEC     R1              ;DEC WAIT COUNT
    89 025774  001367                            BNE     10$             ;LOOP IF NOT TIME DONE
    90 025776  012704  011615                    MOV     #C5SEC,R4       ;SET CONDITION 5 SECONDS
    91 026002  104456                            TRAP    C$ERHRD
       026004  023461                            .WORD   10033
       026006  000000                            .WORD   0
       026010  013100                            .WORD   ERR5
    92 026012  005037  003020                    CLR     ERRSWI          ;CLEAR ERROR SWITCH
    93 026016  005737  003046            11$:    TST     T.CS            ;CHECK IF ANY ERROR
    94 026022  100020                            BPL     14$             ;NO - SKIP
    95 026024  023737  003106  002304            CMP     CURCYL,HLMTW    ;IS THIS BSF CYLINDER?
    96 026032  001006                            BNE     12$             ;NO - SKIP
    97 026034  005737  003114                    TST     DESHD           ;TEST IF HEAD 1
    98 026040  001403                            BEQ     12$             ;NO - SKIP
    99 026042  005737  014514                    TST     BSERRS          ;OUTPUT ALL BSF ERRORS?
   100 026046  001404                            BEQ     13$             ;NO - SKIP
```

```
                                12$:
026050                                  TRAP    C$ERHRD
026050  104456                          .WORD   10031
026052  023457                          .WORD   0
026054  000000                          .WORD   ERR6
026056  013150
026060  005037  003020          13$:    CLR     ERRSWI          ;CLEAR ERROR SWITCH
026064  162737  000002  003004  14$:    SUB     #2,SSINDX       ;REMOVE ENTRY FROM SUBROUT STACK
026072  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
026074  012601                          MOV     (SP)+,R1
026076  012600                          MOV     (SP)+,R0
026100  012603                          MOV     (SP)+,R3
026102  005737  003020                  TST     ERRSWI          ;TEST IF ERROR RETURN
026106  001403                          BEQ     15$             ;YES - SKIP
026110  063716  003020                  ADD     ERRSWI,(SP)     ;ELSE ADD IN ERROR RETURN
026114  000207                          RTS     PC
026116  017616  000000          15$:    MOV     @(SP),(SP)      ;ADJUST FOR ERROR RETURN
026122  000207                          RTS     PC

                                ;       BAD SECTOR CHECK ROUTINE. CHECKS IF SECTOR SPECIFIED IN CURCYL,
                                ;       DESHD, AND DESSEC IS LISTED AS BAD IN THE BAD SECTOR FILES.

026124  010046                  BSCHK:  MOV     R0,-(SP)        ;STORE REGISTERS
026126  010146                          MOV     R1,-(SP)
026130  010346                          MOV     R3,-(SP)
026132  005037  003022                  CLR     BSFLAG          ;CLEAR FLAG
026136  012703  003502                  MOV     #FCTBSF,R3      ;GET POINTER TO FACTORY FILE
026142  022713  177777                  CMP     #-1,(R3)        ;CHECK IF ALL ONES
026146  001005                          BNE     2$              ;NO SKIP TO TEST
026150  012703  004076          1$:     MOV     #FLDBSF,R3      ;ELSE SET POINTER TO FIELD BS FILE
026154  022713  177777                  CMP     #-1,(R3)        ;CHECK IF ALL ONES
026160  001431                          BEQ     8$              ;YES - EXIT
026162  013700  003104          2$:     MOV     NEWCYL,R0       ;BUILD HEADER OF ADDRESS IN QUESTION
026166  012701  000007                  MOV     #7,R1           ;# OF POSITIONS TO SHIFT CYLINDER
026172  006300                  3$:     ASL     R0              ;SHIFT NUMBER
026174  005301                          DEC     R1              ;DONE YET?
026176  001375                          BNE     3$              ;NO, ANOTHER SHIFT PLEASE
026200  005737  003114                  TST     DESHD           ;CHECK IF HEAD 0
026204  001402                          BEQ     4$              ;YES - SKIP
026206  052700  000100                  BIS     #BIT6,R0        ;INSERT HEAD 1
026212  053700  003116          4$:     BIS     DESSEC,R0       ;INSERT SECTOR
026216  022300                  5$:     CMP     (R3)+,R0        ;DID WE FIND AN ENTRY MATCH?
026220  001402                          BEQ     6$              ;YES - EXIT
026222  101005                          BHI     7$              ;NO - FOUND FILE TERMINATOR
026224  000774                          BR      5$              ;NEITHER TRY NEXT ENTRY...

026226  012737  000001  003022  6$:     MOV     #1,BSFLAG       ;SET ERROR FLAG
026234  000403                          BR      8$              ;GO TO EXIT

026236  020327  004076          7$:     CMP     R3,#FLDBSF      ;DONE BOTH FILES?
026242  003742                          BLE     1$              ;NO, GO DO FIELD FILE
026244  012603                  8$:     MOV     (SP)+,R3        ;ELSE RESTORE REGISTERS
026246  012601                          MOV     (SP)+,R1
026250  012600                          MOV     (SP)+,R0
026252  005737  003022                  TST     BSFLAG          ;CHECK IF ERROR
026256  001003                          BNE     9$              ;YES - SKIP
026260  062716  000002                  ADD     #2,(SP)         ;ELSE BUMP ERROR RETURN
026264  000207                          RTS     PC
```

F8


```
                                        9$:     MOV     @(SP),(SP)      ;SET FOR ERROR RETURN
                                                RTS     PC

                                        ;       REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
                                        ;       OPERATION BEING PERFORMED PORTION OF ALL
                                        ;       ERROR MESSAGES.

                                        RPTOP:  MOV     R4,-(SP)
                                                TST     SSINDX          ;TEST SUBROUTINE INDEX 0
                                                BEQ     2$              ;SKIP IF 0
                                                MOV     #2,R4           ;SET INDEXER TO FIRST ENTRY
                                                MOV     #SEQMES,-(SP)
                                                MOV     #FMT9,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #6,SP
                                        1$:
                                                MOV     SUBSTK(R4),-(SP)
                                                MOV     #FMT16,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #6,SP
                                                ADD     #2,R4           ;BUMP INDEX
                                                CMP     R4,SSINDX       ;CHECK IF ALL PRINTED
                                                BLE     1$              ;LOOP IF NOT ALL PRINTED YET
                                        2$:
                                                MOV     #TSTLAB,-(SP)
                                                MOV     ERHEAD,-(SP)
                                                MOV     #FMT4,-(SP)
                                                MOV     #3,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #10,SP
                                                BIC     #SEEKOP!RORWOP,OPFLAG   ;CLEAR SK & RD OR WRT FLAG
                                                MOV     L.CS,R1         ;GET COMMAND EXECUTED
                                                BIC     #177741,R1      ;STRIP ALL BUT FUNCTION CODE
                                                CMP     #6,R1   ;TEST IF SEEK OPERATION
                                                BNE     3$              ;NO - SKIP
                                                BIS     #SEEKOP,OPFLAG  ;ELSE SET SEEK FLAG
                                        3$:     CMP     #12,R1          ;TEST IF WRITE
                                                BNE     4$              ;NO - SKIP
                                                BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
                                        4$:     CMP     #14,R1          ;TEST IF READ
                                                BNE     5$              ;NO - SKIP
                                                BIS     #RORWOP,OPFLAG  ;SET RD OR WRT FLAG
                                        5$:
                                                MOV     OPMSGS(R1),-(SP)
                                                MOV     #MOPER,-(SP)
                                                MOV     #FMT1,-(SP)
                                                MOV     #3,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #10,SP
                                                CMP     R1,#4           ;CHECK IF GET STATUS
                                                BNE     6$              ;NO - SKIP
```

```
                                                BIT     #DRSET,L.DA     ;TEST IF RESET INCLUDED
                                                BEQ     6$              ;NO - SKIP
                                                MOV     #16,R1          ;SET TO PRINT WITH RESET
                                                BR      10$
                                        6$:     BIT     #COMPOP,OPFLAG  ;TEST IF ANY OTHER OPERATION
                                                BEQ     9$              ;NO - SKIP
                                                MOV     OPFLAG,R4       ;SET UP TO DETERMINE WHICH ONE
                                                MOV     #20,R1          ;PRESET THE POINTER
                                        7$:     BIT     #BIT00,R4       ;CHECK THE BIT
                                                BNE     8$              ;IF SET - SKIP
                                                TST     (R1)+           ;BUMP POINTER
                                                ASR     R4
                                                BR      7$

                                        8$:
                                                MOV     OPMSGS(R1),-(SP)
                                                MOV     #FMTXT,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #6,SP
                                        9$:     BIT     #HDR40,OPFLAG   ;TEST IF 40 HEADER OPERATION
                                                BEQ     11$             ;NO - SKIP
                                                MOV     #50,R1          ;ELSE PRINT IT
                                        10$:
                                                MOV     OPMSGS(R1),-(SP)
                                                MOV     #FMTXT,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #6,SP
                                                BR      12$             ;SKIP

                                        11$:    BIT     #SEEKOP,OPFLAG  ;TEST IF SEEK
                                                BEQ     12$             ;NO - SKIP
                                                MOV     DESHD,-(SP)
                                                MOV     #HDWD,-(SP)
                                                MOV     DESSGN,-(SP)
                                                MOV     #SGNWD,-(SP)
                                                MOV     DESDIF,-(SP)
                                                MOV     #DIFWD,-(SP)
                                                MOV     OLDCYL,-(SP)
                                                MOV     #FRMWD,-(SP)
                                                MOV     #FMT13,-(SP)
                                                MOV     #11,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #24,SP
                                        12$:    BIT     #RORWOP,OPFLAG  ;TEST IF READ OR WRITE SET
                                                BEQ     13$             ;NO - SKIP
                                                MOV     DESSEC,-(SP)
                                                MOV     #SECWD,-(SP)
                                                MOV     DESHD,-(SP)
                                                MOV     #HDWD,-(SP)
                                                MOV     CURCYL,-(SP)
                                                MOV     #CYLWD,-(SP)
```

```
                027032  012746  012504                  MOV     #FMT22,-(SP)
                027036  012746  000007                  MOV     #7,-(SP)
                027042  010600                          MOV     SP,R0
                027044  104414                          TRAP    C$PNTB
                027046  062706  000020                  ADD     #20,SP
215             027052  004737  027524          13$:    JSR     PC,CLRPARM      ;CLEAR PARAM TABLE
216             027056  012604                          MOV     (SP)+,R4        ;RESTORE R4
217             027060  000207                          RTS     PC
218
219                                             ;       REPORT REASON ROUTINE
220                                             ;       PRINTS REASON PORTION FOR ALL ERROR REPORTS.
221             027062  010146                  RPTRES: MOV     R1,-(SP)        ;STORE R1
222             027064  010346                          MOV     R3,-(SP)        ;STORE R3
223             027066  010446                          MOV     R4,-(SP)        ;STORE R4
224             027070  012701  003064                  MOV     #RESPARM,R1     ;GET START OF PARAM
225             027074  012103                          MOV     (R1)+,R3        ;GET NUMBER OF PARAM
226             027076  011146                          MOV     (R1),-(SP)
                027100  012746  006126                  MOV     #MRSLT,-(SP)
                027104  012746  011730                  MOV     #FMT2,-(SP)
                027110  012746  000003                  MOV     #3,-(SP)
                027114  010600                          MOV     SP,R0
                027116  104414                          TRAP    C$PNTB
                027120  062706  000010                  ADD     #10,SP
227             027124  021127  011274                  CMP     (R1),#MNDRST    ;TEST IF MESSAGE IS NO DRV STATUS
228             027130  001453                          BEQ     2$              ;YES - SKIP REST OF REPORT
229             027132  012704  012141                  MOV     #FMT11,R4       ;PRISET FOR FORMAT 11
230             027136  022127  011267                  CMP     (R1)+,#MCYLOC   ;CHECK IF REPORTING CYLINDER LOC
231             027142  001002                          BNE     1$              ;NO - SKIP
232             027144  012704  012147                  MOV     #FMT12,R4       ;ELSE CHANGE TO FORMAT 12
233             027150  005303                  1$:     DEC     R3              ;DEC PARAM COUNT
234             027152  001442                          BEQ     2$              ;IF 0 - EXIT
235             027154  012146                          MOV     (R1)+,-(SP)
                027156  012746  011503                  MOV     #RESE3,-(SP)
                027162  010446                          MOV     R4,-(SP)
                027164  012746  000003                  MOV     #3,-(SP)
                027170  010600                          MOV     SP,R0
                027172  104414                          TRAP    C$PNTB
                027174  062706  000010                  ADD     #10,SP
236             027200  012146                          MOV     (R1)+,-(SP)
                027202  012746  011507                  MOV     #RESE4,-(SP)
                027206  010446                          MOV     R4,-(SP)
                027210  012746  000003                  MOV     #3,-(SP)
                027214  010600                          MOV     SP,R0
                027216  104414                          TRAP    C$PNTB
                027220  062706  000010                  ADD     #10,SP
237             027224  162703  000002                  SUB     #2,R3           ;DEC PARAM COUNT
238             027230  001413                          BEQ     2$              ;IF 0 - EXIT
239             027232  012146                          MOV     (R1)+,-(SP)
                027234  012746  011514                  MOV     #RESE5,-(SP)
                027240  012746  011723                  MOV     #FMT1,-(SP)
                027244  012746  000003                  MOV     #3,-(SP)
                027250  010600                          MOV     SP,R0
                027252  104414                          TRAP    C$PNTB
                027254  062706  000010                  ADD     #10,SP
240             027260  012604                  2$:     MOV     (SP)+,R4        ;RESTORE REGS
241             027262  012603                          MOV     (SP)+,R3
242             027264  012601                          MOV     (SP)+,R1
```

```
                                        RTS     PC              ;RETURN
                                ;       REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
                                ;       AND ALL REGISTER CONTENTS.
                                RPTREM:
                                        CLR     -(SP)
                                        BISB    RLDRV+1,(SP)
                                        MOV     #DRVNAM,-(SP)
                                        MOV     RLBAS,-(SP)
                                        MOV     #BASADD,-(SP)
                                        MOV     #FMT5,-(SP)
                                        MOV     #5,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #14,SP

                                ;       REPORT RL11 REGISTERS
                                        MOV     #HDWD,-(SP)
                                        MOV     #CYLWD,-(SP)
                                        MOV     #MPNAM,-(SP)
                                        MOV     #BANAM,-(SP)
                                        MOV     #DANAM,-(SP)
                                        MOV     #CSNAM,-(SP)
                                        MOV     #FMT6,-(SP)
                                        MOV     #7,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #20,SP
                                        MOV     L.MP,-(SP)
                                        MOV     L.BA,-(SP)
                                        MOV     L.DA,-(SP)
                                        MOV     L.CS,-(SP)
                                        MOV     #LAB1,-(SP)
                                        MOV     #FMT8,-(SP)
                                        MOV     #6,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #16,SP
                                        MOV     DESHD,-(SP)
                                        MOV     CURCYL,-(SP)
                                        MOV     T.MP,-(SP)
                                        MOV     T.BA,-(SP)
                                        MOV     T.DA,-(SP)
                                        MOV     T.CS,-(SP)
                                        MOV     #LAB2,-(SP)
                                        MOV     #FMT7,-(SP)
                                        MOV     #10,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #22,SP
                                        RTS     PC

                                ;       CLEAR PARAMETER BLOCK FOR REPORTING
                                CLRPARM: MOV    R5,-(SP)        ;STORE R5
                                        MOV     #RESPARM,R1     ;GET ADDRESS OF BLOCK
                                        MOV     #5,R5           ;SET COUNT
                                1$:     CLR     (R1)+           ;CLEAR WORD
```

J8


```
260 027540  005305                    DEC     R5              ;DEC COUNT
261 027542  001375                    BNE     1$              ;LOOP UNTIL 0
262 027544  012701  003064            MOV     #RESPARM,R1     ;RESET POINTER
263 027550  012605                    MOV     (SP)+,R5        ;RESTORE R5
264 027552  000207                    RTS     PC
265
```

K8


```
                                .TITLE  CZRLNC0 RL01/02 DRIVE TEST 3


                                .SBTTL  *TEST 1         **SEEK TIMING

027554                          T1::
027554  012737  000001  003240          MOV     #1,TSTNM        ;SAVE TEST NUMBER
027562  012737  007343  003014          MOV     #P2T12E,ERHEAD  ;SET ERROR HEADER

                                ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
027570  005737  003474                  TST     CLKFLG          ;P-CLOCK?
027574  001014                          BNE     1$              ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
027576  013746  003240                  MOV     TSTNM,-(SP)
027602  012746  010364                  MOV     #NOTST,-(SP)
027606  012746  000002                  MOV     #2,-(SP)
027612  010600                          MOV     SP,R0
027614  104417                          TRAP    C$PNTF
027616  062706  000006                  ADD     #6,SP
                                                                ;/P-CLOCK IS NOT AVAILABLE"
027622  000137  031472                  JMP     20$             ;EXIT TEST

027626  004737  017146          1$:     JSR     PC,TSTINT       ;INITIALIZE TEST
027632  004737  017164                  JSR     PC,GSTATR       ;CLEAR DRIVE
027636  031472                          20$
027640  012700  003142                  MOV     #OFIN,R0        ;GET ADDRESS OF 1ST TIME VALUE
027644  012701  000030                  MOV     #24.,R1         ;SET COUNT FOR CLEAR
027650  005020                  2$:     CLR     (R0)+           ;CLEAR TIMER STORAGE
027652  005301                          DEC     R1
027654  001375                          BNE     2$
027656  005037  003234                  CLR     PASCNT          ;CLEAR PASS COUNTER
027662  005037  003104                  CLR     NEWCYL          ;POSITION HEADS AT 0
027666  004737  020112                  JSR     PC,XSEEK        ;DO SEEK
027672  031472                          20$
027674  012701  005670                  MOV     #3000.,R1       ;SET WAIT FOR 300 MS
027700  004737  023570                  JSR     PC,RDYWAIT      ;WAIT FOR READY
027704  031472                          20$
027706  004737  024202                  JSR     PC,VERPOS       ;VERIFY POSITION
027712  031472                          20$
027714  004737  021504                  JSR     PC,CHOSHD       ;GO CHOSE HEAD
027720  012700  003152                  MOV     #OFOUT,R0       ;SET PTRS FOR 1 CYL FWD OUTER TIMER
027724  012701  003154                  MOV     #OFOUTU,R1
027730  012703  003166                  MOV     #OROUT,R3
027734  012704  003170                  MOV     #OROUTU,R4
027740  012737  000001  003104          MOV     #1,NEWCYL       ;SET NEWCYL TO CYL 1
027746  012737  000200  003236  3$:     MOV     #128.,COUNT     ;SET COUNTER FOR SEEK LOOP
027754  012737  000110  003140          MOV     #RDHEAD,TEMP8   ;BUILD READ HEADER COMMAND
027762  053737  003034  003140          BIS     RLDRV,TEMP8
027770  042737  002000  003140          BIC     #BIT10,TEMP8
027776  004737  020102          4$:     JSR     PC,XSEEKT       ;DO SEEK BUILD BUT DO NOT START
030002  031472                          20$
030004  013762  003042  000004          MOV     L.DA,RLDA(R2)   ;LOAD RL REGISTERS
030012  013762  003036  000000          MOV     L.CS,RLCS(R2)
030020  010046                          MOV     R0,-(SP)        ;STORE R0
030034  005737  003010                  TST     DONE            ;TEST IF INTERRUPT
030040  001011                          BNE     5$              ;YES - SKIP
030042  004737  017010                  JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
030046  012603                          MOV     (SP)+,R3        ;GET MESSAGE POINTER
```

L8


```
    57 030050  104456                        TRAP    C$ERHRD
       030052  002261                        .WORD   1201
       030054  000000                        .WORD   0
       030056  012646                        .WORD   ERR1
    58 030060  000137  031472                JMP     20$
    59
    60 030064  005737  003046        5$:     TST     T.CS            ;CHECK IF ANY ERRORS
    61 030070  100006                        BPL     6$              ;NO - SKIP
    62 030072  104456                        TRAP    C$ERHRD
       030074  002262                        .WORD   1202
       030076  000000                        .WORD   0
       030100  013150                        .WORD   ERR6
    63 030102  000137  031472                JMP     20$
    64
    65 030106  005037  003010        6$:     CLR     DONE            ;CLEAR INTERRUPT FLAG
    66 030112  005037  172542                CLR     @#CLKCSB        ;CLEAR CLOCK COUNT SET BUFFER
       030116  005037  172544                CLR     @#CLKCTR        ;CLEAR CLOCK COUNTER
       030122  012737  000023  172540        MOV     #23,@#CLKCSR    ;INITIALIZE CLOCK FOR COUNT-UP MODE,
    67                                                                 ;/OF TIME INTERVAL
    68 030130  013762  003140  000000        MOV     TEMP8,RLCS(R2)  ;LOAD RL11 CONTROL AND STATUS REGISTER
    69                                                                 ;/TO INITIATE SEEK OPERATION
    70 030136  012737  003720  003456        MOV     #2000.,XDELAY   ;SAVE ARGUMENT
       030144  004737  016210                JSR     PC,TIME         ;CALL TIMING ROUTINE
    71 030150  013705  172544                MOV     @#CLKCTR,R5     ;STORE CLOCK COUNTER CONTENTS
       030154  005037  172540                CLR     @#CLKCSR        ;EVENT FINISHED, STOP CLOCK
    72 030160  012600                        MOV     (SP)+,R0        ;RESTORE R0
    73 030162  013737  003140  003036        MOV     TEMP8,L.CS      ;SET IF ERROR TO REPORT
    74 030170  004737  024202                JSR     PC,VERPOS       ;VERIFY POSITION
    75 030174  031472                        20$
    76 030176  005737  003112                TST     DESSGN          ;CHECK WHICH SEEK DIRECTION
    77 030202  001403                        BEQ     7$              ;REVERSE - SKIP
    78 030204  060510                        ADD     R5,(R0)         ;ADD TO FORWARD TOTAL
    79 030206  005511                        ADC     (R1)            ;ADD IN OVERFLOW
    80 030210  000402                        BR      8$              ;SKIP
    81
    82 030212  060513                7$:     ADD     R5,(R3)         ;ADD TO REVERSE TOTAL
    83 030214  005514                        ADC     (R4)            ;ADD IN OVERFLOW
    84 030216  005337  003236        8$:     DEC     COUNT           ;DEC SEEK COUNT
    85 030222  001403                        BEQ     9$              ;SKIP IF 0
    86 030224  004737  021570                JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW CYL
    87 030230  000662                        BR      4$              ;REDO SEEK LOOP
    88
    89 030232  162710  000470        9$:     SUB     #312.,(R0)      ;SUB CONSTANT FOR READ HEADER TIME
    90 030236  162713  000470                SUB     #312.,(R3)
    91 030242  012705  000006                MOV     #6,R5           ;SET SHIFT COUNT TO DIVIDE BY 64
    92 030246  000241                10$:    CLC                     ;DIVIDE BOTH TOTALS BY 64
    93 030250  006011                        ROR     (R1)
    94 030252  006010                        ROR     (R0)
    95 030254  000241                        CLC
    96 030256  006014                        ROR     (R4)
    97 030260  006013                        ROR     (R3)
    98 030262  005305                        DEC     R5
    99 030264  001370                        BNE     10$
   100 030266  005237  003234                INC     PASCNT          ;BUMP PASS COUNT
   101 030272  022737  000001  003234        CMP     #1,PASCNT       ;TEST IF PASS 1
   102 030300  001051                        BNE     13$             ;NO - SKIP
   103 030302  012737  000177  003104        MOV     #127.,NEWCYL    ;ELSE SET TO POSITION HDS TO 127
```

M8


```
                                                    CMP     #1,T.DRIVE      ;DRIVE = RL01?
                                                    BEQ     11$             ;YUP
                                                    MOV     #255.,NEWCYL    ;NO - SET FOR A MID POS SEEK RL02
                                            11$:    JSR     PC,XSEEK        ;DO SEEK
                                                    20$
                                                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                    JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                    20$
                                                    JSR     PC,VERPOS       ;VERIFY POSITION
                                                    20$
                                                    MOV     #OFMID,R0       ;SET PTRS FOR TIMING 1 CYL SK AT 127
                                                    MOV     #OFMIDU,R1
                                                    MOV     #ORMID,R3
                                                    MOV     #ORMIDU,R4
                                                    MOV     #128.,NEWCYL    ;SET NEWCYL TO 128
                                                    CMP     #1,T.DRIVE      ;RL01?
                                                    BEQ     12$             ;YUP
                                                    MOV     #256.,NEWCYL    ;SET FOR RL02
                                            12$:    JMP     3$              ;DO SEEK LOOP

                                            13$:    CMP     #2,PASCNT       ;TEST IF PASS 2
                                                    BNE     14$             ;NO - SKIP
                                                    MOV     NXTHL,NEWCYL    ;SET UP TO TIME 1 CYL SEEK AT INNER
                                                    JSR     PC,XSEEK        ;  LIMIT
                                                    20$
                                                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                    JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                    20$
                                                    JSR     PC,VERPOS       ;VERIFY POSITION
                                                    20$
                                                    MOV     #OFIN,R0        ;SET POINTERS
                                                    MOV     #OFINU,R1
                                                    MOV     #ORIN,R3
                                                    MOV     #ORINU,R4
                                                    MOV     HLMTW,NEWCYL    ;LOAD NEW CYLINDER
                                                    JMP     3$              ;DO SEEK LOOP

                                            14$:    CMP     #3,PASCNT       ;TEST IF PASS 3
                                                    BNE     15$             ;NO - SKIP
                                                    CLR     NEWCYL          ;ELSE SET UP TO TIME 85/170 CYL SEEK
                                                    JSR     PC,XSEEK        ;  AT OUTER LIMIT
                                                    20$
                                                    MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                    JSR     PC,RDYWAIT      ;WAIT FOR DRIVE READY
                                                    20$
                                                    JSR     PC,VERPOS       ;VERIFY POSITION
                                                    20$
                                                    MOV     #HFOUT,R0       ;SET POINTERS
                                                    MOV     #HFOUTU,R1
                                                    MOV     #HROUT,R3
                                                    MOV     #HFOUTU,R4
                                                    MOV     #85.,NEWCYL     ;LOAD NEWCYL FOR 85 CYL SEEK
                                                    CMP     #1,T DRIVE      ;RL01?
                                                    BEQ     18$             ;YUP
                                                    MOV     #170.,NEWCYL    ;NO - SET FOR RL02
                                                    BR      18$
```

```
030632  022737  000004  003234  15$:   CMP     #4,PASCNT           ;TEST IF PASS 4
030640  001041                         BNE     17$                 ;NO - SKIP
030642  012737  000252  003104         MOV     #170.,NEWCYL        ;ELSE SET UP TO TIME 85 CYL SEEK
030650  022737  000001  002300         CMP     #1,T.DRIVE          ;RL01?
030656  001403                         BEQ     16$                 ;YES
030660  012737  000525  003104         MOV     #341.,NEWCYL        ;NO - SET FOR RL02
030666  004737  020112          16$:   JSR     PC,XSEEK            ; AT INNER LIMIT
030672  031472                         20$
030674  012701  005670                 MOV     #3000.,R1           ;SET WAIT COUNT FOR 300 MS
030700  004737  023570                 JSR     PC,RDYWAIT          ;WAIT FOR READY
030704  031472                         20$
030706  004737  024202                 JSR     PC,VERPOS           ;VERIFY POSITION
030712  031472                         20$
030714  012700  003172                 MOV     #HFIN,R0            ;SET POINTERS
030720  012701  003174                 MOV     #HFINU,R1
030724  012703  003202                 MOV     #HRIN,R3
030730  012704  003204                 MOV     #HRINU,R4
030734  013737  002304  003104         MOV     HLMTW,NEWCYL        ;SET NEWCYL TO MAX CYL
030742  000434                         BR      18$                 ;DO TIMING LOOP

030744  022737  000005  003234  17$:   CMP     #5,PASCNT           ;TEST IF PASS 5
030752  001032                         BNE     19$                 ;NO - SKIP
030754  005037  003104                 CLR     NEWCYL              ;ELSE SET UP TO TIME 256/512 CYL SEEK
030760  004737  020112                 JSR     PC,XSEEK            ;  OVER ALL SURFACE
030764  031472                         20$
030766  012701  005670                 MOV     #3000.,R1           ;SET WAIT COUNT FOR 300 MS
030772  004737  023570                 JSR     PC,RDYWAIT          ;WAIT FOR DRIVE READY
030776  031472                         20$
031000  004737  024202                 JSR     PC,VERPOS           ;VERIFY POSITION
031004  031472                         20$
031006  012700  003212                 MOV     #AFMID,R0           ;SET POINTERS
031012  012701  003214                 MOV     #AFMIDU,R1
031016  012703  003216                 MOV     #ARMID,R3
031022  012704  003220                 MOV     #ARMIDU,R4
031026  013737  002304  003104         MOV     HLMTW,NEWCYL        ;SET NEWCYL
031034  000137  027746          18$:   JMP     3$

031040                          19$:
031040  012746  007607                 MOV     #VALDES,-(SP)
031044  012746  007553                 MOV     #SKTMES,-(SP)
031050  012746  011730                 MOV     #FMT2,-(SP)
031054  012746  000003                 MOV     #3,-(SP)
031060  010600                         MOV     SP,R0
031062  104417                         TRAP    C$PNTF
031064  062706  000010                 ADD     #10,SP
031070  005046                         CLR     -(SP)
031072  153716  003035                 BISB    RLDRV+1,(SP)
031076  012746  006621                 MOV     #DRVNAM,-(SP)
031102  013746  003030                 MOV     RLBAS,-(SP)
031106  012746  006610                 MOV     #BASADD,-(SP)
031112  012746  011750                 MOV     #FMT5,-(SP)
031116  012746  000005                 MOV     #5,-(SP)
031122  010600                         MOV     SP,R0
031124  104417                         TRAP    C$PNTF
031126  062706  000014                 ADD     #14,SP
031132  012746  007666                 MOV     #LABEXP,-(SP)
031136  012746  007660                 MOV     #LABOUT,-(SP)
```

B9


```
        031142  012746  007651          MOV     #LABMID,-(SP)
        031146  012746  007643          MOV     #LABIN,-(SP)
        031152  012746  012342          MOV     #FMT18,-(SP)
        031156  012746  000005          MOV     #5,-(SP)
        031162  010600                  MOV     SP,R0
        031164  104417                  TRAP    C$PNTF
        031166  062706  000014          ADD     #14,SP
    201 031172  013746  003222          MOV     EXOCYL,-(SP)
        031176  013746  003152          MOV     OFOUT,-(SP)
        031202  013746  003146          MOV     OFMID,-(SP)
        031206  013746  003142          MOV     OFIN,-(SP)
        031212  012746  007677          MOV     #LABOCF,-(SP)
        031216  012746  012374          MOV     #FMT19,-(SP)
        031222  012746  000006          MOV     #6,-(SP)
        031226  010600                  MOV     SP,R0
        031230  104417                  TRAP    C$PNTF
        031232  062706  000016          ADD     #16,SP
    202 031236  013746  003222          MOV     EXOCYL,-(SP)
        031242  013746  003166          MOV     OROUT,-(SP)
        031246  013746  003162          MOV     ORMID,-(SP)
        031252  013746  003156          MOV     ORIN,-(SP)
        031256  012746  007711          MOV     #LABOCR,-(SP)
        031262  012746  012374          MOV     #FMT19,-(SP)
        031266  012746  000006          MOV     #6,-(SP)
        031272  010600                  MOV     SP,R0
        031274  104417                  TRAP    C$PNTF
        031276  062706  000016          ADD     #16,SP
    203 031302  013746  003224          MOV     EXHCYL,-(SP)
        031306  013746  003176          MOV     HFOUT,-(SP)
        031312  013746  003172          MOV     HFIN,-(SP)
        031316  012746  007723          MOV     #LABHCF,-(SP)
        031322  012746  012431          MOV     #FMT20,-(SP)
        031326  012746  000005          MOV     #5,-(SP)
        031332  010600                  MOV     SP,R0
        031334  104417                  TRAP    C$PNTF
        031336  062706  000014          ADD     #14,SP
    204 031342  013746  003224          MOV     EXHCYL,-(SP)
        031346  013746  003206          MOV     HROUT,-(SP)
        031352  013746  003202          MOV     HRIN,-(SP)
        031356  012746  007737          MOV     #LABHCR,-(SP)
        031362  012746  012431          MOV     #FMT20,-(SP)
        031366  012746  000005          MOV     #5,-(SP)
        031372  010600                  MOV     SP,R0
        031374  104417                  TRAP    C$PNTF
        031376  062706  000014          ADD     #14,SP
    205 031402  013746  003226          MOV     EXACYL,-(SP)
        031406  013746  003212          MOV     AFMID,-(SP)
        031412  012746  007753          MOV     #LABACF,-(SP)
        031416  012746  012461          MOV     #FMT21,-(SP)
        031422  012746  000004          MOV     #4,-(SP)
        031426  010600                  MOV     SP,R0
        031430  104417                  TRAP    C$PNTF
        031432  062706  000012          ADD     #12,SP
    206 031436  013746  003226          MOV     EXACYL,-(SP)
        031442  013746  003216          MOV     ARMID,-(SP)
        031446  012746  007767          MOV     #LABACR,-(SP)
        031452  012746  012461          MOV     #FMT21,-(SP)
```

C9


```
        031456  012746  000004                  MOV     #4,-(SP)
        031462  010600                          MOV     SP,R0
        031464  104417                          TRAP    C$ØNTF
        031466  062706  000012                  ADD     #12,SP
    207 031472                          20$:
    208 031472                          L10023:
        031472  104401                          TRAP    C$ETST
```

D9


```
                                        .SBTTL  *TEST 2          **BASIC READ DATA (BAD SECTOR FILE)

031474                                  T2::
031474  012737  000002  003240                  MOV     #2,TSTNM        ;SAVE TEST NUMBER
031502  004737  017146                          JSR     PC,TSTINT       ;INITIALIZE TEST
031506  004737  017164                          JSR     PC,GSTATR       ;CLEAR DRIVE
031512  031524                                  1$                      ;ERROR RETURN ADDRESS
031514  005037  003500                          CLR     BSFVAL          ;ENABLE BAD SEC FILE READ
031520  004737  021630                          JSR     PC,RDBSF        ;READ BAD SECTOR FILE
031524                                  1$:
031524                                  L10024:
031524  104401                                  TRAP    C$ETST
```

E9


```
                                .SBTTL  *TEST 3         **WRITE/READ DATA (PART 1)

                        T3::
031526  012737  000003  003240        MOV     #3,TSTNM        ;SAVE TEST NUMBER
031534  012737  007402  003014        MOV     #P2T14E,ERHEAD  ;SET ERROR HEADER
031542  004737  017146                JSR     PC,TSTINT       ;INITIALIZE TEST
031546  004737  017164                JSR     PC,GSTATR       ;CLEAR DRIVE
031552  031746                        T3065$
031554  004737  021614                JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
031560  004737  021504                JSR     PC,CHOSHD       ;GO CHOSE HEAD
031564  005037  003116                CLR     DESSEC          ;       SECTOR 0
031570  005037  003104                CLR     NEWCYL          ;       CYLINDER 0
031574  005037  031640                CLR     T310$           ;CLEAR PATTERN SELECT
031600  004737  020112        T306$:  JSR     PC,XSEEK        ;POSITION HEADS
031604  031746                        T3065$
031606  012701  005670                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
031612  004737  023570                JSR     PC,RDYWAIT      ;WAIT FOR READY
031616  031746                        T3065$
031620  004737  024202                JSR     PC,VERPOS       ;VERIFY POSITION
031624  031746                        T3065$
031626  005037  031640                CLR     T310$           ;CLEAR PATTERN SELECTOR

                        T307$:
                        T3.1:
031632  104402                        TRAP    C$BSUB
031634  004537  024672                JSR     R5,DATGEN       ;GENERATE DATA
031640  000000          T310$:  .WORD   0               ;PATTERN SELECT WORD
031642  004737  025322                JSR     PC,XWRITE       ;DO WRITE DATA
031646  031664                        1$
031650  004737  025362                JSR     PC,XREAD        ;DO READ DATA
031654  031664                        1$
031656  004737  025032                JSR     PC,DATCOM       ;COMPARE DATA
031662  031664                        1$
031664  012737  000002  003020  1$:   MOV     #2,ERRSWI       ;INIT ERROR SWITCH
031672                  L10026:
031672  104403                        TRAP    C$ESUB

031674  104410                        TRAP    C$ESCAPE
031676  000050                        .WORD   L10025-.
031700  022737  000010  031640        CMP     #8.,T310$       ;WAS DATA PAT 8 USED?
031706  001403                        BEQ     2$              ;YES - SKIP
031710  005237  031640                INC     T310$           ;ELSE BUMP TO NEXT PATTERN
031714  000746                        BR      T307$           ;DO TEST WITH NEW PATTERN

031716  004737  021530          2$:   JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
031722  031746                        T3065$                  ;ABORT RETURN
031724  005037  031640                CLR     T310$           ;SET PATTERN SELECT TO 0
031730  004737  026124          3$:   JSR     PC,BSCHK        ;CHECK IF SECTOR BAD
031734  031740                        4$                      ;YES RETURN - SKIP TO 4$
031736  000720                        BR      T306$           ;NO RETURN - DO TEST THIS SECTOR

031740  005237  003104          4$:   INC     NEWCYL          ;BUMP TO NEXT CYLINDER
031744  000771                        BR      3$              ;CHECK IF THIS ONE BAD

                        T3065$:
                        L10025:
031746  104401                        TRAP    C$ETST
```

F9


```
                                .SBTTL  *TEST 4         **ROTATIONAL TIMING

                        T4::
031750  012737  000004  003240            MOV     #4,TSTNM        ;SAVE TEST NUMBER
031756  012737  007423  003014            MOV     #P2T15E,ERHEAD  ;SET ERROR HEADER

                        ;CHECK FOR PRESENCE OF A P-CLOCK...BYPASS TEST IF NOT AVAILABLE
031764  005737  003474                    TST     CLKFLG          ;P-CLOCK?
031770  001014                            BNE     1$              ;BRANCH TO PERFORM TEST IF P-CLOCK IS PRESENT
031772  013746  003240                    MOV     TSTNM,-(SP)
031776  012746  010364                    MOV     #NOTST,-(SP)
032002  012746  000002                    MOV     #2,-(SP)
032006  010600                            MOV     SP,R0
032010  104417                            TRAP    C$PNTF
032012  062706  000006                    ADD     #6,SP
                                                                    ;/P-CLOCK IS NOT AVAILABLE"
032016  104432                            TRAP    C$EXIT
032020  000542                            .WORD   L10027-.

032022  005003                1$:         CLR     R3              ;CLEAR FOR TIMING STORAGE
032024  005004                            CLR     R4
032026  004737  017146                    JSR     PC,TSTINT       ;INITIALIZE TEST
032032  004737  017164                    JSR     PC,GSTATR       ;CLEAR DRIVE
032036  032554                            8$
032040  004537  024672                    JSR     R5,DATGEN       ;GENERATE DATA
032044  000000                            0                       ;PATTERN 0
032046  005037  003116                    CLR     DESSEC          ;CLEAR TO SECTOR 0
032052  004737  021504                    JSR     PC,CHOSHD       ;GO SELECT HEAD
032056  013737  014502  003104            MOV     LOLIMW,NEWCYL   ;SET FOR CYLINDER
032064  004737  020112                    JSR     PC,XSEEK        ;DO SEEK
032070  032554                            8$
032072  012701  005670                    MOV     #3000.,R1       ;SET WAIT FOR 300 MS
032076  004737  023570                    JSR     PC,RDYWAIT      ;WAIT FOR READY
032102  032554                            8$
032104  004737  024202                    JSR     PC,VERPOS       ;VERIFY POSITION
032110  032554                            8$
032112  012701  000100                    MOV     #64.,R1 ;SET LOOP COUNTER
032116  012705  003044        2$:         MOV     #L.MP,R5        ;SET A POINTER
032122  004737  025312                    JSR     PC,XWRITT       ;DO FIRST WRITE
032126  032554                            8$
032130  011562  000006                    MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS
032134  014562  000004                    MOV     -(R5),RLDA(R2)
032140  014562  000002                    MOV     -(R5),RLBA(R2)
032144  014562  000000                    MOV     -(R5),RLCS(R2)
032162  005737  003010                    TST     DONE            ;TEST IF INTERRUPT
032166  001011                            BNE     3$              ;YES - SKIP
032170  004737  017010                    JSR     PC,WAITIN       ;ELSE WAIT FOR TIMEOUT
032174  012603                            MOV     (SP)+,R3        ;GET MESSAGE POINTER
032176  104456                            TRAP    C$ERHRD
032200  002735                            .WORD   1501
032202  000000                            .WORD   0
032204  012646                            .WORD   ERR1
032206  000137  032554                    JMP     8$

032212  005737  003046        3$:         TST     T.CS            ;TEST IF ANY ERRORS
032216  100006                            BPL     4$              ;NO - SKIP
032220  104456                            TRAP    C$ERHRD
```

```
                032222  002736                          .WORD   1502
                032224  000000                          .WORD   0
                032226  013150                          .WORD   ERR6
     52         032230  000137  032554                 JMP     8$
     53
     54         032234  012705  003044          4$:     MOV     #L.MP,R5        ;SET POINTER TO RL LOAD REGS
     55         032240  005037  003010                  CLR     DONE            ;CLEAR INTERRUPT INDICATOR
     56         032244  005037  172542                  CLR     @#CLKCSB        ;CLEAR CLOCK COUNT SET BUFFER
                032250  005037  172544                  CLR     @#CLKCTR        ;CLEAR CLOCK COUNTER
                032254  012737  000023  172540          MOV     #23,@#CLKCSR    ;INITIALIZE CLOCK FOR COUNT-UP MODE,
     57                                                                         ;/OF TIME INTERVAL
     58         032262  011562  000006                  MOV     (R5),RLMP(R2)   ;LOAD RL REGISTERS FOR 2ND WRITE
     59         032266  014562  000004                  MOV     -(R5),RLDA(R2)
     60         032272  014562  000002                  MOV     -(R5),RLBA(R2)
     61         032276  014562  000000                  MOV     -(R5),RLCS(R2)
     62         032302  012737  005670  003456          MOV     #3000.,XDELAY   ;SAVE ARGUMENT
                032310  004737  016210                  JSR     PC,TIME         ;CALL TIMING ROUTINE
     63         032314  013700  172544                  MOV     @#CLKCTR,R0     ;STORE CLOCK COUNTER CONTENTS
                032320  005037  172540                  CLR     @#CLKCSR        ;EVENT FINISHED, STOP CLOCK
     64         032324  005737  003010                  TST     DONE            ;TEST IF INTERRUPT OCCURRED
     65         032330  001010                          BNE     5$              ;YES - SKIP
     66         032332  004737  017010                  JSR     PC,WAITIN       ;GO WAIT FOR INTERRUPT
     67         032336  012603                          MOV     (SP)+,R3        ;GET MESSAGE POINTER
     68         032340  104456                          TRAP    C$ERHRD
                032342  002737                          .WORD   1503
                032344  000000                          .WORD   0
                032346  012646                          .WORD   ERR1
     69         032350  000501                          BR      8$
     70
     71         032352  005737  003046          5$:     TST     T.CS            ;TEST IF ANY ERROR
     72         032356  100005                          BPL     6$              ;NO - SKIP
     73         032360  104456                          TRAP    C$ERHRD
                032362  002740                          .WORD   1504
                032364  000000                          .WORD   0
                032366  013150                          .WORD   ERR6
     74         032370  000471                          BR      8$
     75
     76         032372  060003          6$:     ADD     R0,R3           ;ADD IN TIME USED
     77         032374  005504                  ADC     R4              ;DOUBLE PRECISION
     78         032376  005301                  DEC     R1              ;DEC LOOP COUNTER
     79         032400  001246                  BNE     2$              ;LOOP UNTIL 0
     80         032402  012701  000006          MOV     #6,R1           ;SET DIVIDE COUNT
     81         032406  000241          7$:     CLC                     ;CLEAR CARRY FOR DIVIDE
     82         032410  006004                  ROR     R4              ;DIVIDE SUM BY 100(8)
     83         032412  006003                  ROR     R3
     84         032414  005301                  DEC     R1              ;DEC DIVIDE COUNT
     85         032416  001373                  BNE     7$              ;LOOP UNTIL DONE
     86         032420  012746  007607                  MOV     #VALDES,-(SP)
                032424  012746  007565                  MOV     #SRTMES,-(SP)
                032430  012746  011730                  MOV     #FMT2,-(SP)
                032434  012746  000003                  MOV     #3,-(SP)
                032440  010600                          MOV     SP,R0
                032442  104417                          TRAP    C$PNTF
                032444  062706  000010                  ADD     #10,SP
     87         032450  005046                          CLR     -(SP)
                032452  153716  003035                  BISB    RLDRV+1,(SP)
                032456  012746  006621                  MOV     #DRVNAM,-(SP)
```

```
        032462  013746  003030                 MOV     RLBAS,-(SP)
        032466  012746  006610                 MOV     #BASADD,-(SP)
        032472  012746  011750                 MOV     #FMT5,-(SP)
        032476  012746  000005                 MOV     #5,-(SP)
        032502  010600                         MOV     SP,R0
        032504  104417                         TRAP    C$PNTF
        032506  062706  000014                 ADD     #14,SP
     88 032512  013746  003230                 MOV     EXROT,-(SP)
        032516  012746  007633                 MOV     #MAPROX,-(SP)
        032522  012746  011507                 MOV     #RESE4,-(SP)
        032526  010346                         MOV     R3,-(SP)
        032530  012746  011503                 MOV     #RESE3,-(SP)
        032534  012746  012571                 MOV     #FMT26,-(SP)
        032540  012746  000006                 MOV     #6,-(SP)
        032544  010600                         MOV     SP,R0
        032546  104417                         TRAP    C$PNTF
        032550  062706  000016                 ADD     #16,SP
     89 032554  012737  000002  003020  8$:    MOV     #2,ERRSWI       ;INITIALIZE ERROR SWITCH
     90 032562                         L10027:
        032562  104401                         TRAP    C$ETST
```

```
                                .SBTTL  *TEST 5         **WRITE/READ DATA (PART 2)

032564                          T5::
032564  012737  000005  003240          MOV     #5,TSTNM        ;SAVE TEST NUMBER
032572  012737  007446  003014          MOV     #P2T16E,ERHEAD  ;SET ERROR HEADER
032600  004737  017146                  JSR     PC,TSTINT       ;INITIALIZE TEST
032604  004737  017164                  JSR     PC,GSTATR       ;CLEAR DRIVE
032610  033700                          T3165$
032612  004737  021614                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
032616  005037  003234                  CLR     PASCNT          ;CLEAR PASS TO 0
032622  012705  177776                  MOV     #-2,R5          ;SET
032626  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
032632  001006                          BNE     1$              ;NO - SKIP
032634  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
032642  001002                          BNE     1$              ;YES - SKIP
032644  012705  177760                  MOV     #-16.,R5                ;ELSE SET PEOPLE TO NEG 8
032650                          1$:
032650  012701  002506                  MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
032654  012737  000010  002302          MOV     #10,JUNK        ;SET CLEAR COUNT
032662  013721  014502          2$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LO LIMIT
032666  005337  002302                  DEC     JUNK            ;DEC COUNT
032672  001373                          BNE     2$              ;LOOP UNTIL 0
032674  013737  014504  002512          MOV     HILIMW,T33TBL+4 ;INSERT HILIMIT
032702  013737  014504  002514          MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
032710  013737  014504  002516          MOV     HILIMW,T33TBL+10

032716  062705  000002          T3100$: ADD     #2,R5           ;BUMP R5 BY 2
032722  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
032730  001031                          BNE     5$              ;YES - SKIP
032732  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
032736  001002                          BNE     1$              ;NO - SKIP
032740  062705  000016                  ADD     #16,R5          ;ELSE BUMP CYLINDER POINTER BY 7
032744  022737  000001  002300  1$:     CMP     #1,T.DRIVE      ;RL01 OR RL02?THAT IS THE Q
032752  001404                          BEQ     2$              ;ANS IS RL01
032754  020527  000244                  CMP     R5,#164.
032760  103013                          BHIS    4$
032762  000403                          BR      3$              ;TEST PAST TABLE-YES EXIT

032764  020527  000122          2$:     CMP     R5,#82.
032770  103007                          BHIS    4$              ;TES PAST THE TABLE
032772  016537  002606  002302  3$:     MOV     CYLTBL(R5),JUNK ;GET NEXT TABLE ENTRY
033000  043737  002306  002302          BIC     CLRBYT,JUNK     ;CLEAR UPPER BYTE
033006  001007                          BNE     6$
033010  000137  033700          4$:     JMP     T3165$          ;EXIT TEST

033014  023705  014504          5$:     CMP     HILIMW,R5       ;TEST IF ALL CYLINDERS USED
033020  001773                          BEQ     4$              ;YES - EXIT TEST
033022  010537  002302                  MOV     R5,JUNK         ;USE R5 AS NEXT CYLINDER
033026  023737  002302  014502  6$:     CMP     JUNK,LOLIMW     ;CHECK IF LOWER THAN LOLIMIT
033034  103730                          BLO     T3100$          ;YES - SKIP
033036  023737  002302  014504          CMP     JUNK,HILIMW     ;CHECK IF HIGHER THAN HILIMIT
033044  101324                          BHI     T3100$          ;YES - SKIP
033046  012703  002546                  MOV     #TBT,R3
033052  013713  002302                  MOV     JUNK,(R3)
033056  013763  002302  000002          MOV     JUNK,2(R3)
033064  013763  002302  000004          MOV     JUNK,4(R3)
033072  013763  002302  000006          MOV     JUNK,6(R3)
```

J9


```
                                T3101$:
                                T5.1:
 033124 104402                          TRAP    C$BSUB
 033126 042737 003760 003006            BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
 033134 005737 003234                   TST     PASCNT          ;TEST IF PASS 0
 033140 001414                          BEQ     2$              ;YES - SKIP
 033142 023727 003234 000003            CMP     PASCNT,#3       ;TEST IF PASS 3
 033150 001404                          BEQ     1$              ;YES - SKIP
 033152 002407                          BLT     2$              ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
 033154 012737 000003 003234            MOV     #3,PASCNT       ;ELSE SET TO 3
 033162 052737 000020 003006    1$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
 033170 000405                          BR      3$              ;SKIP

 033172 005037 003234           2$:     CLR     PASCNT          ;SET PASS COUNT TO 0
 033176 052737 000040 003006            BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
 033204 012737 000003 003024    3$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
 033212 013703 003026                   MOV     TBLSTR,R3       ;GET STORED TABLE ADDRESS
 033216 012701 002506                   MOV     #T33TBL,R1
 033222 012703 002546                   MOV     #TBT,R3
 033226 005037 003116           4$:     CLR     DESSEC          ;CLEAR TO SECTOR 0
 033232 012137 003104                   MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
 033236 004737 020112                   JSR     PC,XSEEK        ;DO SEEK
 033242 033606                          15$
 033244 012701 005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 033250 004737 023570                   JSR     PC,RDYWAIT      ;WAIT FOR READY
 033254 033606                          15$
 033256 012337 003104                   MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
 033262 004737 020112                   JSR     PC,XSEEK        ;DO SEEK
 033266 033606                          15$
 033270 012701 005670                   MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
 033274 004737 023570                   JSR     PC,RDYWAIT      ;WAIT FOR READY
 033300 033606                          15$
 033302 004737 024202                   JSR     PC,VERPOS       ;VERIFY POSITION
 033306 033606                          15$
 033310 004737 026124           5$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
 033314 033446                          9$                      ;"YES" RETURN
 033316 013737 003116 033336            MOV     DESSEC,6$       ;SET DATA PATTERN = TO SECTOR NUMBER
 033324 042737 177770 033336            BIC     #177770,6$      ;CLEAR ALL BUT LSD
 033332 004537 024672                   JSR     R5,DATGEN       ;GO GENERATE DATA
 033336 000000                  6$:     .WORD   0
 033340 032737 000001 003024            BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
 033346 001425                          BEQ     7$              ;NO - SKIP
 033350 004737 025322                   JSR     PC,XWRITE       ;DO WRITE
 033354 033606                          15$
 033356 005237 003116                   INC     DESSEC          ;INC SECTOR
 033362 022737 000050 003116            CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
 033370 001347                          BNE     5$              ;NO - SKIP
 033372 042737 000060 003006            BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
 033400 042737 000001 003024            BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
 033406 052737 000100 003006            BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
 033414 005037 003116                   CLR     DESSEC          ;CLEAR TO SECTOR 0
 033420 000733                          BR      5$              ;SKIP
```

K9


```
                                                    ;TEST IF READ THIS PASS
033422  032737  000002  003024  7$:     BIT     #BIT1,WRTSWI
033430  001414                        BEQ     10$             ;NO - SKIP
033432  004737  025362          8$:     JSR     PC,XREAD        ;ELSE DO READ
033436  033606                        15$
033440  004737  025032                JSR     PC,DATCOM       ;COMPARE DATA
033444  033606                        15$
033446  005237  003116          9$:     INC     DESSEC          ;BUMP SECTOR
033452  022737  000050  003116        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
033460  001313                        BNE     5$              ;NO - LOOP
033462  005037  003116          10$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
033466  005037  003024                CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
033472  005237  003234                INC     PASCNT          ;BUMP PASS COUNT
033476  042737  003760  003006        BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
033504  023727  003234  000003        CMP     PASCNT,#3       ;TEST IS PASS 3
033512  001435                        BEQ     15$             ;YES - SKIP
033514  023727  003234  000006        CMP     PASCNT,#6       ;TEST IF PASS 6
033522  001431                        BEQ     15$             ;YES - SKIP
033524  012737  000002  003024        MOV     #BIT1,WRTSWI    ;SET READ REQUIRED BIT
033532  023727  003234  000001        CMP     PASCNT,#1       ;TEST IF PASS 1
033540  001415                        BEQ     13$             ;YES - SKIP
033542  023727  003234  000005        CMP     PASCNT,#5       ;TEST IF PASS 4
033550  001411                        BEQ     13$             ;YES - SKIP
033552  000404                        BR      12$             ;SKIP

033554  052737  002000  003006  11$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
033562  000407                        BR      14$             ;GO DO NEXT PASS

033564  052737  000020  003006  12$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
033572  000403                        BR      14$             ;SKIP

033574  052737  000040  003006  13$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
033602  000137  033226          14$:    JMP     4$              ;GO DO NEXT PASS

033606  012737  000002  003020  15$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
033614                          L10031:
033614  104403                        TRAP    C$ESUB

033616  104410                        TRAP    C$ESCAPE
033620  000060                        .WORD   L10030-.
033622  012737  000003  003024        MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
033630  023727  003234  000003        CMP     PASCNT,#3       ;TEST IF PASS 3
033636  001004                        BNE     16$             ;NO - SKIP
033640  012737  002514  003026        MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
033646  000410                        BR      17$             ;GO START PASS 4

033650  005037  003234          16$:    CLR     PASCNT          ;CLEAR TO PASS 0
033654  004737  021530                JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
033660  032716                        T3100$                  ;ABORT RETURN
033662  012737  002506  003026        MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
033670  062703  000006          17$:    ADD     #6,R3
033674  000137  033124                JMP     T3101$

033700                          T3165$:
033700                          L10030:
033700  104401                        TRAP    C$ETST
```

L9


```
                                        .SBTTL  *TEST 6          **WRITE LOCK ERROR AND DATA PROTECTION

                                        T6::
033702  012737  000006  003240                MOV     #6,TSTNM        ;SAVE TEST NUMBER
033710  005737  003444                        TST     PASNUM          ;TEST IF FIRST PASS
033714  001003                              BNE     1$              ;NO - SKIP
033716  005737  014500                        TST     MISWIW          ;TEST IF RUN MANUAL INTERVENTION
033722  100402                              BMI     2$              ;YES - SKIP
033724  000137  034724          1$:         JMP     T3265$          ;EXIT TST

033730                          2$:
033730                          T6.1:
033730  104402                              TRAP    C$BSUB
033732  012737  007467  003014                MOV     #P2T17E,ERHEAD  ;SET ERROR HEADER
033740  004737  017146                        JSR     PC,TSTINT       ;INITIALIZE TEST
033744  004737  017164                        JSR     PC,GSTATR       ;CLEAR DRIVE
033750  034572                              11$
033752  005037  003114                        CLR     DESHD           ;SET TO HEAD 0
033756  005037  003116                        CLR     DESSEC          ;SET TO SECTOR 0
033762  005037  003104                        CLR     NEWCYL          ;CLEAR TO CYLINDER 0
033766  004737  020112                        JSR     PC,XSEEK        ;DO SEEK
033772  034572                              11$
033774  012701  013560                        MOV     #6000.,R1       ;INITIALIZE WAIT COUNT
034000  004737  023570                        JSR     PC,RDYWAIT      ;WAIT FOR READY
034004  034572                              11$
034006  004737  024202                        JSR     PC,VERPOS       ;VERIFY POSITION
034012  034572                              11$
034014  032737  020000  003054                BIT     #WLSTAT,T.MP    ;TEST IF WRITE LOCK SET
034022  001116                              BNE     4$              ;YES - SKIP
034024  004537  024672                        JSR     R5,DATGEN       ;GENERATE DATA
034030  000007                                                      ;PATTERN 7
034032  004737  025322                        JSR     PC,XWRITE       ;WRITE DATA
034036  034572                              11$
034040  004737  025362                        JSR     PC,XREAD        ;READ DATA
034044  034572                              11$
034046  004737  025032                        JSR     PC,DATCOM       ;CHECK DATA
034052  034572                              11$
034054  005046                              CLR     -(SP)
034056  153716  003035                        BISB    RLDRV+1,(SP)
034062  012746  006621                        MOV     #DRVNAM,-(SP)
034066  013746  003030                        MOV     RLBAS,-(SP)
034072  012746  006610                        MOV     #BASADD,-(SP)
034076  012746  010056                        MOV     #OPR1A,-(SP)
034102  012746  010105                        MOV     #OPR004,-(SP)
034106  012746  011631                        MOV     #FMTOP1,-(SP)
034112  012746  000007                        MOV     #7,-(SP)
034116  010600                              MOV     SP,R0
034120  104417                              TRAP    C$PNTF
034122  062706  000020                        ADD     #20,SP
034126  012701  000024                        MOV     #20.,R1         ;INITIALIZE WAIT COUNT
034132                          3$:
034144  004737  017164                        JSR     PC,GSTATR       ;GET STATUS
034150  034572                              11$
034152  032737  020000  003054                BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK SET
034160  001037                              BNE     4$              ;YES - SKIP
034162  012746  011477                        MOV     #BELL,-(SP)
034166  012746  011626                        MOV     #FMTXT,-(SP)
```

M9


```
                034172  012746  000002          MOV     #2,-(SP)
                034176  010600                  MOV     SP,R0
                034200  104417                  TRAP    C$PNTF
                034202  062706  000006          ADD     #6,SP
           46   034206  005301                  DEC     R1              ;DEC COUNT
           47   034210  001350                  BNE     3$              ;SKIP IF NOT 0
           48   034212  005046                  CLR     -(SP)
                034214  153716  003035          BISB    RLDRV+1,(SP)
                034220  012746  010056          MOV     #OPR1A,-(SP)
                034224  012746  010151          MOV     #BYPSNM,-(SP)
                034230  012746  007467          MOV     #P2T17E,-(SP)
                034234  012746  012540          MOV     #FMT23,-(SP)
                034240  012746  000005          MOV     #5,-(SP)
                034244  010600                  MOV     SP,R0
                034246  104417                  TRAP    C$PNTF
                034250  062706  000014          ADD     #14,SP
           49   034254  104432                  TRAP    C$EXIT
                034256  000446                  .WORD   L10032-.
           50
           51   034260  004537  024672  4$:     JSR     R5,DATGEN       ;GENERATE DATA
           52   034264  000001                  1                       ;PATTERN 1
           53   034266  012705  003036          MOV     #L.CS,R5        ;GET ADDRESS OF L REGS
           54   034272  012715  000112          MOV     #WTDATA,(R5)    ;LOAD WRITE COMMAND
           55   034276  053715  003034          BIS     RLDRV,(R5)      ;INSERT DRIVE NUMBER
           56   034302  042725  002000          BIC     #BIT10,(R5)+    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
           57   034306  012725  005072          MOV     #OBUFF,(R5)+    ;LOAD BUS ADDRESS
           58   034312  005025                  CLR     (R5)+           ;CYL 0, HD 0, SECTOR 0
           59   034314  012725  177600          MOV     #177600,(R5)+   ;128 WORDS
           60   034320  012701  000454          MOV     #300.,R1        ;SET WAIT COUNT FOR 30 MS
           61   034324  005037  003010          CLR     DONE            ;CLEAR INTERRUPT FLAG
           62   034330  014562  000006          MOV     -(R5),RLMP(R2)  ;LOAD RL REGS
           63   034334  014562  000004          MOV     -(R5),RLDA(R2)
           64   034340  014562  000002          MOV     -(R5),RLBA(R2)
           65   034344  014562  000000          MOV     -(R5),RLCS(R2)
           66   034350                  5$:
           67   034362  005737  003010          TST     DONE            ;CHECK IF INTERRUPT
           68   034366  001013                  BNE     6$              ;YES - SKIP
           69   034370  005301                  DEC     R1              ;DEC WAIT COUNT
           70   034372  001366                  BNE     5$              ;LOOP IF NOT 0
           71   034374  004737  017010          JSR     PC,WAITIN       ;WAIT FOR INTERRUPT
           72   034400  012603                  MOV     (SP)+,R3        ;GET RESULT MESSAGE
           73   034402  104456                  TRAP    C$ERHRD
                034404  003245                  .WORD   1701
                034406  000000                  .WORD   0
                034410  012646                  .WORD   ERR1
           74   034412  104432                  TRAP    C$EXIT
                034414  000164                  .WORD   L10033-.
           75   034416  004737  017214  6$:     JSR     PC,GSTAT        ;GET STATUS
           76   034422  034572                  11$
           77   034424  032737  040000  003046  BIT     #DRVERR,T.CS    ;TEST IF ANY ERROR SET
           78   034432  001006                  BNE     7$              ;YES - SKIP
           79   034434  012703  011024          MOV     #MDRERR,R3      ;SET RESULT MESSAGE POINTER
           80   034440  104456                  TRAP    C$ERHRD
                034442  003246                  .WORD   1702
                034444  000000                  .WORD   0
                034446  012762                  .WORD   ERR3
           81   034450  032737  002000  003054  7$:     BIT     #WGESTAT,T.MP   ;TEST IF WGE SET
```

N9


```
     034456  001006                          BNE     8$              ;YES - SKIP
     034460  012703  011103                  MOV     #MWGERR,R3      ;SET MESSAGE FOR WGE NOT SET
     034464  104456                          TRAP    C$ERHRD
     034466  003250                          .WORD   1704
     034470  000000                          .WORD   0
     034472  012762                          .WORD   ERR3
     034474  042737  040000  003046  8$:     BIC     #DRVERR,T.CS    ;CLEAR DRIVE ERROR BIT
     034502  042737  002000  003054          BIC     #WGESTAT,T.MP   ;CLEAR WGE BIT
     034510  032737  157400  003054          BIT     #157400,T.MP    ;TEST IF ANY OTHER ERRORS
     034516  001004                          BNE     9$              ;YES - GO REPORT
     034520  032737  036000  003046          BIT     #36000,T.CS     ;TEST ANY ERRORS IN CS REG
     034526  001405                          BEQ     10$             ;NO - SKIP
     034530                          9$:
     034530  104456                          TRAP    C$ERHRD
     034532  003247                          .WORD   1703
     034534  000000                          .WORD   0
     034536  013150                          .WORD   ERR6
     034540  000414                          BR      11$             ;EXIT TEST

     034542  004737  017164          10$:    JSR     PC,GSTATR       ;GET STATUS AND RESET ERROR
     034546  034572                          11$
     034550  004537  024672                  JSR     R5,DATGEN       ;GO GENERATE DATA
     034554  000007                          7                       ;PATTERN 7
     034556  004737  025362                  JSR     PC,XREAD        ;READ DATA
     034562  034572                          11$
     034564  004737  025032                  JSR     PC,DATCOM       ;COMPARE DATA
     034570  034572                          11$
     034572  012737  000002  003020  11$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
     034600                          L10033:
     034600  104403                          TRAP    C$ESUB

     034602  012737  000002  003020  T3204$: MOV     #2,ERRSWI       ;INIT ERROR SWITCH
     034610  005046                          CLR     -(SP)
     034612  153716  003035                  BISB    RLDRV+1,(SP)
     034616  012746  006621                  MOV     #DRVNAM,-(SP)
     034622  013746  003030                  MOV     RLBAS,-(SP)
     034626  012746  006610                  MOV     #BASADD,-(SP)
     034632  012746  010056                  MOV     #OPR1A,-(SP)
     034636  012746  010037                  MOV     #OPR12,-(SP)
     034642  012746  011631                  MOV     #FMTOP1,-(SP)
     034646  012746  000007                  MOV     #7,-(SP)
     034652  010600                          MOV     SP,R0
     034654  104417                          TRAP    C$PNTF
     034656  062706  000020                  ADD     #20,SP
     034662  012701  001274                  MOV     #700.,R1        ;INITIALIZE WAIT COUNT
     034666                          1$:
     034700  004737  017164                  JSR     PC,GSTATR       ;GET STATUS
     034704  034602                          T3204$
     034706  032737  020000  003054          BIT     #WLSTAT,T.MP    ;CHECK IF WRITE LOCK RESET
     034714  001403                          BEQ     T3265$
     034716  005301                          DEC     R1              ;DEC WAIT COUNT
     034720  001362                          BNE     1$              ;LOOP IF NOT 0
     034722  000727                          BR      T3204$          ;ELSE REPEAT MESSAGE

     034724                          T3265$:
     034724                          L10032:
     034724  104401                          TRAP    C$ETST
```

B10


```
                                .SBTTL  *TEST 7         **ADJACENT CYLINDER INTERFERENCE

034726                          T7::
034726  012737  000007  003240          MOV     #7,TSTNM        ;SAVE TEST NUMBER
034734  012737  007521  003014          MOV     #P2T18E,ERHEAD  ;SET ERROR HEADER
034742  004737  017146                  JSR     PC,TSTINT       ;INITIALIZE TEST
034746  004737  017164                  JSR     PC,GSTATR       ;CLEAR DRIVE
034752  036146                          T3365$
034754  004737  021614                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
034760  005037  003234                  CLR     PASCNT          ;CLEAR PASS TO 0
034764  012705  177776                  MOV     #-2,R5          ;SET R5
034770  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
034774  001007                          BNE     1$              ;NO - SKIP
034776  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035004  001003                          BNE     1$              ;YES - SKIP
035006  012705  177730                  MOV     #-40.,R5        ;ELSE SET R5 TO NEG 20
035012  000402                          BR      2$              ;SKIP

035014  012705  177770          1$:     MOV     #-10,R5         ;ELSE SET FOR NEG 4
035020  012701  002506          2$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
035024  012737  000010  002302          MOV     #10,JUNK        ;SET CLEAR COUNT
035032  013721  014502          3$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
035036  005337  002302                  DEC     JUNK            ;DEC COUNT
035042  001373                          BNE     3$              ;LOOP UNTIL 0
035044  004537  024672                  JSR     R5,DATGEN       ;GO GENERATE DATA
035050  000011                          9.                      ;PATTERN 9
035052  013737  014504  002510          MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
035060  013737  014504  002512          MOV     HILIMW,T33TBL+4 ;INTO APPROPRIATE LOCATIONS
035066  013737  014504  002516          MOV     HILIMW,T33TBL+10
035074  013737  014504  002524          MOV     HILIMW,T33TBL+16

035102  062705  000002          T3300$: ADD     #2,R5
035106  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
035114  001034                          BNE     6$              ;YES - SKIP
035116  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
035122  001403                          BEQ     1$              ;NO - SKIP
035124  062705  000006                  ADD     #6,R5           ;ELSE BUMP CYLINDER POINTER BY 3
035130  000402                          BR      2$              ;SKIP

035132  062705  000044          1$:     ADD     #36.,R5         ;BUMP TO NEXT ENTRY
035136  022737  000001  002300  2$:     CMP     #1,T.DRIVE
035144  001404                          BEQ     3$
035146  020537  000244                  CMP     R5,164.
035152  103013                          BHIS    5$
035154  000403                          BR      4$

035156  020527  000122          3$:     CMP     R5,#82.
035162  103007                          BHIS    5$
035164  016537  002606  002302  4$:     MOV     CYLTBL(R5),JUNK
035172  043737  002306  002302          BIC     CLRBYT,JUNK
035200  001013                          BNE     8$
035202  000137  033700          5$:     JMP     T3165$

035206  005705                  6$:     TST     R5              ;TEST IF R5 0
035210  001002                          BNE     7$              ;NO - SKIP
035212  062705  000002                  ADD     #2,R5
035216  023705  002304          7$:     CMP     HLMTW,R5        ;TEST IF ALL CYLINDERS USED
```

C10


```
                                                BEQ     5$              ;YES - EXIT TEST
                                                MOV     R5,JUNK         ;USE R5 AS NEXT CYLINDER
                                        8$:     CMP     JUNK,LOLIMW     ;CHECK IF LOWER THAN LOLIMIT
                                                BLO     T3300$          ;YES - SKIP
                                                CMP     JUNK,HILIMW     ;CHECK IF HIGHER THAN HILIMIT
                                                BHI     T3300$          ;YES - SKIP
                                                MOV     #TBT,R3
                                                MOV     JUNK,(R3)
                                                MOV     JUNK,6(R3)
                                                MOV     JUNK,10(R3)
                                                MOV     JUNK,12(R3)
                                                MOV     JUNK,16(R3)
                                                SUB     #1,JUNK
                                                MOV     JUNK,2(R3)
                                                MOV     JUNK,12(R3)
                                                ADD     #2,JUNK
                                                MOV     JUNK,4(R3)
                                                MOV     JUNK,14(R3)
                                                MOV     R3,TBLSTR
                                                JSR     PC,CHOSHD       ;GO CHOSE HEAD

                                        T3301$:
                                        T7.1:
                                                TRAP    C$BSUB
                                                BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
                                                TST     PASCNT          ;TEST IF PASS 0
                                                BEQ     2$              ;YES - SKIP
                                                CMP     PASCNT,#4       ;TEST IF PASS 4
                                                BEQ     1$              ;YES - SKIP
                                                BLT     2$              ;CHECK IF LESS THAN 4, IF YES CLEAR TO 0
                                                MOV     #4,PASCNT       ;ELSE SET TO 4
                                        1$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
                                                BR      3$              ;SKIP

                                        2$:     CLR     PASCNT          ;SET PASS COUNT TO 0
                                                BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
                                        3$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
                                                MOV     #T33TBL,R1
                                                MOV     #TBT,R3
                                        4$:     CLR     DESSEC          ;CLEAR TO SECTOR 0
                                                MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
                                                JSR     PC,XSEEK        ;DO SEEK
                                                15$
                                                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                15$
                                                MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
                                                JSR     PC,XSEEK        ;DO SEEK
                                                15$
                                                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                15$
                                                JSR     PC,VERPOS       ;VERIFY POSITION
                                                15$
                                        5$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
                                                8$                      ;"YES" RETURN
                                                BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
```

```
                                                BEQ     6$                      ;NO - SKIP
                                                JSR     PC,XWRITE               ;DO WRITE
                                                15$
                                                INC     DESSEC                  ;INC SECTOR
                                                CMP     #40.,DESSEC             ;TEST IF ALL SECTORS USED
                                                BNE     5$                      ;NO - SKIP
                                                BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
                                                BIC     #BIT0,WRTSWI            ;CLEAR WRITE REQUIRED SWITCH
                                                BIS     #FOLWRT,OPFLAG          ;SET FOLLOWING WRITE QUALIFIER
                                                CLR     DESSEC                  ;CLEAR TO SECTOR 0
                                                BR      5$                      ;SKIP

                                        6$:     BIT     #BIT1,WRTSWI            ;TEST IF READ THIS PASS
                                                BEQ     9$                      ;NO - SKIP
                                        7$:     JSR     PC,XREAD                ;ELSE DO READ
                                                15$
                                                JSR     PC,DATCOM               ;COMPARE DATA
                                                15$
                                        8$:     INC     DESSEC                  ;BUMP SECTOR
                                                CMP     #40.,DESSEC             ;TEST IF ALL SECTORS USED
                                                BNE     5$                      ;NO - LOOP
                                        9$:     CLR     DESSEC                  ;CLEAR DESIRED SECTOR
                                                CLR     WRTSWI                  ;CLEAR WRITE/READ SWITCH
                                                INC     PASCNT                  ;BUMP PASS COUNT
                                                BIC     #MQUALS,OPFLAG          ;CLEAR ALL QUALIFIERS
                                                CMP     PASCNT,#4               ;TEST IS PASS 4
                                                BEQ     15$                     ;YES - SKIP
                                                CMP     PASCNT,#8.              ;TEST IF PASS 8.
                                                BEQ     15$                     ;YES - SKIP
                                                CMP     PASCNT,#3               ;TEST IF PASS 3
                                                BEQ     12$                     ;YES - SKIP
                                                CMP     PASCNT,#7               ;TEST IF PASS 7
                                                BEQ     13$                     ;YES - SKIP
                                                MOV     #BIT0,WRTSWI            ;SET WRITE REQUIRED
                                                CMP     PASCNT,#1               ;TEST IF PASS 1
                                                BEQ     11$                     ;YES - SKIP
                                                CMP     PASCNT,#2               ;TEST IF PASS 2
                                                BEQ     11$                     ;YES - SKIP
                                                BIS     #OUTINS,OPFLAG          ;SET MESSAGE QUALIFIER
                                        10$:    JMP     4$                      ;GO DO NEXT PASS

                                        11$:    BIS     #INOUTS,OPFLAG          ;SET MESSAGE QUALIFIER
                                                BR      10$

                                        12$:    BIS     #REVSKS,OPFLAG          ;SET MESSAGE QUALIFIER
                                                BR      14$

                                        13$:    BIS     #FWDSKS,OPFLAG          ;SET MESSAGE QUALIFIER
                                        14$:    MOV     #BIT1,WRTSWI            ;SET READ REQUIRED
                                                BR      10$

                                        15$:    MOV     #2,ERRSWI               ;INIT ERROR SWITCH
                                        L10035:
                                                TRAP    C$ESUB

                                                TRAP    C$ESCAPE
                                                .WORD   L10034-.
```

E10


```
172 036070  012737  000003  003024          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
173 036076  023727  003234  000004          CMP     PASCNT,#4       ;TEST IF PASS 4
174 036104  001004                          BNE     16$             ;NO - SKIP
175 036106  012737  002516  003026          MOV     #T33TBL+10,TBLSTR ;STORE MID POINT IN TABLE
176 036114  000410                          BR      17$             ;GO START PASS 4
177
178 036116  005037  003234          16$:    CLR     PASCNT          ;CLEAR TO PASS 0
179 036122  004737  021530                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD 1 OR END TEST
180 036126  035102                          T3300$                  ;ABORT RETURN
181 036130  012737  002506  003026          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
182
183 036136  062703  000010          17$:    ADD     #10,R3
184 036142  000137  035364                  JMP     T3301$
185
186 036146                          T3365$:
    036146                          L10034:
    036146  104401                          TRAP    C$ETST
```

F10


```
                                .SBTTL  *TEST 8         **OVERWRITE

                        T8::
036150  012737  000010  003240          MOV     #10,TSTNM       ;SAVE TEST NUMBER
036156  012737  007543  003014          MOV     #P2T19E,ERHEAD  ;SET ERROR HEADER
036164  004737  017146                  JSR     PC,TSTINT       ;INITIALIZE TEST
036170  004737  017164                  JSR     PC,GSTATR       ;CLEAR DRIVE
036174  037346                          T3465$
036176  004737  021614                  JSR     PC,CKBSVD       ;GO CHECK IF BAD SECTOR FILES VALID
036202  005037  003234                  CLR     PASCNT          ;CLEAR PASS TO 0
036206  012705  177776                  MOV     #-2,R5          ;SET R5
036212  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
036216  001007                          BNE     1$              ;NO - SKIP
036220  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
036226  001003                          BNE     1$              ;YES - SKIP
036230  012705  177730                  MOV     #-40.,R5        ;ELSE SET R5 TO NEG 20
036234  000402                          BR      2$              ;SKIP

036236  012705  177770          1$:     MOV     #-10,R5         ;SET FOR NEXT ENTRY
036242  012701  002506          2$:     MOV     #T33TBL,R1      ;GET ADDRESS OF WORK TABLE
036246  012737  000010  002302          MOV     #10,JUNK        ;SET CLEAR COUNT
036254  013721  014502          3$:     MOV     LOLIMW,(R1)+    ;CLEAR LOCATIONS TO LOLIMIT
036260  005337  002302                  DEC     JUNK            ;DEC COUNT
036264  001373                          BNE     3$              ;LOOP UNTIL 0
036266  013737  014504  002510          MOV     HILIMW,T33TBL+2 ;INSERT HILIMIT
036274  013737  014504  002514          MOV     HILIMW,T33TBL+6 ;INTO APPROPRIATE LOCATIONS
036302  013737  014504  002520          MOV     HILIMW,T33TBL+12

036310  062705  000002          T3400$: ADD     #2,R5
036314  032737  000001  014500          BIT     #ALLCYL,MISWIW  ;TEST IF USE ALL CYLINDERS
036322  001034                          BNE     6$              ;YES - SKIP
036324  005737  003444                  TST     PASNUM          ;TEST IF FIRST PASS (QUICK VERIFY)
036330  001003                          BNE     1$              ;NO - SKIP
036332  062705  000046                  ADD     #38.,R5         ;ELSE BUMP CYLINDER POINTER BY 19
036336  000402                          BR      2$              ;SKIP

036340  062705  000006          1$:     ADD     #6,R5           ;BUMP CYLINDER POINTER BY 3
036344  022737  000001  002300  2$:     CMP     #1,T.DRIVE
036352  001404                          BEQ     3$
036354  020527  000244                  CMP     R5,#164.
036360  103013                          BHIS    5$
036362  000403                          BR      4$

036364  020527  000122          3$:     CMP     R5,#82.
036370  103007                          BHIS    5$
036372  016537  002606  002302  4$:     MOV     CYLTBL(R5),JUNK
036400  043737  002306  002302          BIC     CLRBYT,JUNK
036406  001013                          BNE     8$
036410  000137  037346          5$:     JMP     T3465$          ;EXIT TEST

036414  005705                  6$:     TST     R5              ;TEST IF R5 0
036416  001002                          BNE     7$              ;NO - SKIP
036420  062705  000002                  ADD     #2,R5
036424  022705  002304          7$:     CMP     #HLMTW,R5       ;TEST IF ALL CYLINDERS USED
036430  001767                          BEQ     5$              ;YES - EXIT TEST
036432  010537  002302                  MOV     R5,JUNK         ;USE R5 AS NEXT CYLINDER
036436  023737  002302  014502  8$:     CMP     JUNK,LOLIMW     ;TEST IF PAST LO LIMIT
```

```
                                                BLO     T3400$          ;YES - SKIP
                                                CMP     JUNK,HILIMW     ;TEST IF PAST HILIMIT
                                                BHI     T3400$          ;YES - SKIP
                                                MOV     #TBT,R3
                                                MOV     JUNK,(R3)
                                                MOV     JUNK,2(R3)
                                                MOV     JUNK,4(R3)
                                                MOV     JUNK,6(R3)
                                                MOV     JUNK,10(R3)
                                                MOV     JUNK,12(R3)
                                                MOV     R3,TBLSTR
                                                JSR     PC,CHOSHD       ;GO CHOSE HEAD

                                        T3401$:
                                        T8.1:
                                                TRAP    C$BSUB
                                                BIC     #MQUALS,OPFLAG  ;CLEAR ALL MESSAGE QUALIFIERS
                                                TST     PASCNT          ;TEST IF PASS 0
                                                BEQ     2$              ;YES - SKIP
                                                CMP     PASCNT,#3       ;TEST IF PASS 3
                                                BEQ     1$              ;YES - SKIP
                                                BLT     2$              ;CHECK IF LESS THAN 3, IF YES CLEAR TO 0
                                                MOV     #3,PASCNT       ;ELSE SET TO 3
                                        1$:     BIS     #INOUTS,OPFLAG  ;SET MESSAGE QUAL
                                                BR      3$              ;SKIP

                                        2$:     CLR     PASCNT          ;SET PASS COUNT TO 0
                                                BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUAL
                                        3$:     MOV     #3,WRTSWI       ;SET READ AND WRITE SWITCH
                                                MOV     #T33TBL,R1
                                                MOV     #TBT,R3
                                        4$:     CLR     DESSEC
                                                MOV     (R1)+,NEWCYL    ;GET NEXT TABLE ENTRY
                                                JSR     PC,XSEEK        ;DO SEEK
                                                18$
                                                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                18$
                                                MOV     (R3)+,NEWCYL    ;GET NEXT TABLE ENTRY
                                                JSR     PC,XSEEK        ;DO SEEK
                                                18$
                                                MOV     #3000.,R1       ;SET WAIT COUNT FOR 300 MS
                                                JSR     PC,RDYWAIT      ;WAIT FOR READY
                                                18$
                                                JSR     PC,VERPOS       ;VERIFY POSITION
                                                18$
                                        5$:     JSR     PC,BSCHK        ;CHECK FOR BAD SECTOR
                                                11$                     ;"YES" RETURN
                                                TST     PASCNT          ;TEST IF PASS 0
                                                BEQ     6$              ;YES - SKIP
                                                CMP     #3,PASCNT       ;TEST IF PASS 3
                                                BEQ     6$              ;YES - SKIP
                                                CLR     8$              ;ELSE CLEAR DATA PATTERN SELECTOR
                                                BR      7$

                                        6$:     MOV     #8.,8$          ;SET DATA PATTERN SELECTOR TO 8
                                        7$:     JSR     R5,DATGEN       ;GO GENERATE DATA
```

H10


```
                                                8$:     .WORD   0
                                                        BIT     #BIT0,WRTSWI    ;TEST IF WRITE THIS PASS
                                                        BEQ     9$              ;NO - SKIP
                                                        JSR     PC,XWRITE       ;DO WRITE
                                                        18$
                                                        INC     DESSEC          ;INC SECTOR
                                                        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
                                                        BNE     5$              ;NO - SKIP
                                                        BIC     #INOUTS!OUTINS,OPFLAG ;CLEAR QUALIFIERS
                                                        BIC     #BIT0,WRTSWI    ;CLEAR WRITE REQUIRED SWITCH
                                                        BIS     #FOLWRT,OPFLAG  ;SET FOLLOWING WRITE QUALIFIER
                                                        CLR     DESSEC          ;CLEAR TO SECTOR 0
                                                        BR      5$              ;SKIP

                                                9$:     BIT     #BIT1,WRTSWI    ;TEST IF READ THIS PASS
                                                        BEQ     12$             ;NO - SKIP
                                                10$:    JSR     PC,XREAD        ;ELSE DO READ
                                                        18$
                                                        JSR     PC,DATCOM       ;COMPARE DATA
                                                        18$
                                                11$:    INC     DESSEC          ;BUMP SECTOR
                                                        CMP     #40.,DESSEC     ;TEST IF ALL SECTORS USED
                                                        BNE     5$              ;NO - LOOP
                                                12$:    CLR     DESSEC          ;CLEAR DESIRED SECTOR
                                                        CLR     WRTSWI          ;CLEAR WRITE/READ SWITCH
                                                        INC     PASCNT          ;BUMP PASS COUNT
                                                        BIC     #MQUALS,OPFLAG  ;CLEAR ALL QUALIFIERS
                                                        CMP     PASCNT,#3       ;TEST IS PASS 3
                                                        BEQ     18$             ;YES - SKIP
                                                        CMP     PASCNT,#6       ;TEST IF PASS 6
                                                        BEQ     18$             ;YES - SKIP
                                                        CMP     PASCNT,#1       ;TEST IF PASS 1
                                                        BEQ     15$             ;YES - SKIP
                                                        CMP     PASCNT,#4       ;TEST IF PASS 4
                                                        BEQ     16$             ;YES - SKIP
                                                        MOV     #BIT1,WRTSWI    ;SET WRITE REQUIRED BIT
                                                        CMP     PASCNT,#2       ;TEST IF PASS 2
                                                        BEQ     14$             ;YES - SKIP
                                                        BIS     #REVSKO,OPFLAG  ;SET REVERSE QUALIFIER
                                                13$:    JMP     4$              ;GO DO NEXT PASS

                                                14$:    BIS     #FWDSKO,OPFLAG  ;SET FWD QUALIFIER
                                                        BR      13$             ;GO DO NEXT PASS

                                                15$:    BIS     #INOUTS,OPFLAG  ;SET QUALIFIER
                                                        BR      17$             ;SKIP

                                                16$:    BIS     #OUTINS,OPFLAG  ;SET MESSAGE QUALIFIER
                                                17$:    MOV     #BIT0,WRTSWI    ;SET WRITE REQUIRED BIT
                                                        BR      13$             ;GO DO NEXT PASS

                                                18$:    MOV     #2,ERRSWI       ;INIT ERROR SWITCH
                                                L10037:
037262  104403                                         TRAP    C$ESUB

037264  104410                                         TRAP    C$ESCAPE
037266  000060                                         .WORD   L10036-.
```

I10


```
  172 037270  012737  000003  003024          MOV     #3,WRTSWI       ;SET FOR READ AND WRITE REQ.
  173 037276  023727  003234  000003          CMP     PASCNT,#3       ;TEST IF PASS 3
  174 037304  001004                          BNE     19$             ;NO - SKIP
  175 037306  012737  002514  003026          MOV     #T33TBL+6,TBLSTR ;STORE MID POINT IN TABLE
  176 037314  000410                          BR      20$             ;GO START PASS 4
  177
  178 037316  005037  003234          19$:    CLR     PASCNT          ;CLEAR TO PASS 0
  179 037322  004737  021530                  JSR     PC,SWAPHD       ;GO SWAP TO HEAD ONE OR ABORT TEST
  180 037326  036310                          T3400$                  ;ABORT RETURN
  181 037330  012737  002506  003026          MOV     #T33TBL,TBLSTR  ;STORE START OF TABLE
  182 037336  062703  000006          20$:    ADD     #6,R3
  183 037342  000137  036534                  JMP     T3401$
  184
  185 037346                          T3465$:
      037346                          L10036:
      037346  104401                          TRAP    C$ETST
```

J10


```
                                .SBTTL  PARAMETER CODING

037350  000030                          .WORD L10040-L$HARD/2
037352  005130                          .WORD   T$CODE
037354  037516                          .WORD   CNTYPE
037356  000001                          .WORD   1
037360  000031                          .WORD   T$CODE
037362  037432                          .WORD   CSRMSG
037364  160000                          .WORD   T$LOLIM
037366  177776                          .WORD   T$HILIM
037370  001031                          .WORD   T$CODE
037372  037446                          .WORD   VECMSG
037374  000000                          .WORD   T$LOLIM
037376  000776                          .WORD   T$HILIM
037400  004032                          .WORD   T$CODE
037402  037510                          .WORD   DRMSG
037404  003400                          .WORD   3400
037406  000000                          .WORD   T$LOLIM
037410  000007                          .WORD   T$HILIM
037412  003130                          .WORD   T$CODE
037414  037466                          .WORD   DRTYPE
037416  000001                          .WORD   1
037420  002032                          .WORD   T$CODE
037422  037455                          .WORD   BRMSG
037424  000340                          .WORD   340
037426  000000                          .WORD   T$LOLIM
037430  000007                          .WORD   T$HILIM

                                        .EVEN
037432                          L10040:

037432  102  125  123  CSRMSG: .ASCIZ  /BUS ADDRESS/
037446  126  105  103  VECMSG: .ASCIZ  /VECTOR/
037455  102  122  040  BRMSG:  .ASCIZ  /BR LEVEL/
037466  104  122  111  DRTYPE: .ASCIZ  /DRIVE TYPE = RL01/
037510  104  122  111  DRMSG:  .ASCIZ  /DRIVE/
037516  122  114  061  CNTYPE: .ASCIZ  /RL11/
                                .EVEN
037524  000061                          .WORD L10041-L$SOFT/2
037526  000130                          .WORD   T$CODE
037530  037670                          .WORD   CYLQ
037532  000001                          .WORD   1
037534  000130                          .WORD   T$CODE
037536  037704                          .WORD   SECQ
037540  000002                          .WORD   2
037542  000130                          .WORD   T$CODE
037544  037721                          .WORD   MANQ
037546  100000                          .WORD   100000
037550  000130                          .WORD   T$CODE
037552  037755                          .WORD   LOLIMQ
037554  040000                          .WORD   40000
037556  006044                          .WORD   T$CODE
037560  001052                          .WORD   T$CODE
037562  037774                          .WORD   LIMVAL
037564  000777                          .WORD   777
037566  000000                          .WORD   T$LOLIM
037570  000375                          .WORD   T$HILIM
```

K10


```
 37 037572                                  1$:
    037572  000130                                .WORD   T$CODE
    037574  040002                                .WORD   HILIMQ
    037576  020000                                .WORD   20000
 38 037600  006044                                .WORD   T$CODE
 39 037602  002052                                .WORD   T$CODE
    037604  037774                                .WORD   LIMVAL
    037606  000777                                .WORD   777
    037610  000000                                .WORD   T$LOLIM
    037612  000377                                .WORD   T$HILIM
 40 037614                                  2$:
    037614  000130                                .WORD   T$CODE
    037616  040023                                .WORD   HEADQ
    037620  010000                                .WORD   10000
 41 037622  006044                                .WORD   T$CODE
 42 037624  003052                                .WORD   T$CODE
    037626  040045                                .WORD   HEADV
    037630  000017                                .WORD   17
    037632  000000                                .WORD   T$LOLIM
    037634  000001                                .WORD   T$HILIM
 44 037636                                  3$:
    037636  004052                                .WORD   T$CODE
    037640  040070                                .WORD   ERLIMQ
    037642  000377                                .WORD   377
    037644  000000                                .WORD   T$LOLIM
    037646  000377                                .WORD   T$HILIM
 46 037650  005052                                .WORD   T$CODE
    037652  040112                                .WORD   DCLIMQ
    037654  000377                                .WORD   377
    037656  000001                                .WORD   T$LOLIM
    037660  000377                                .WORD   T$HILIM
 47 037662  006130                                .WORD   T$CODE
    037664  040133                                .WORD   BSOUTQ
    037666  000001                                .WORD   1
 49                                             .EVEN
    037670                                  L10041:
 51
 52 037670      125     123     105     CYLQ:   .ASCIZ  /USE ALL CYL/
 53 037704      125     123     105     SECQ:   .ASCIZ  /USE ALL SECT/
 60 037721      104     117     040     MANQ:   .ASCIZ  /DO MANUAL INTERVENTION TEST/
 62 037755      114     117     127     LOLIMQ: .ASCIZ  /LOW SEEK LIMIT/
 63 037774      126     101     114     LIMVAL: .ASCIZ  /VALUE/
 64 040002      125     120     120     HILIMQ: .ASCIZ  /UPPER SEEK LIMIT/
 65 040023      125     123     105     HEADQ:  .ASCIZ  /USE ONLY ONE SURF/
 66 040045      127     110     101     HEADV:  .ASCIZ  /WHAT SURF (0 OR 1)/
 68 040070      111     116     120     ERLIMQ: .ASCIZ  /INPUT ERROR LIMIT/
 70 040112      104     101     124     DCLIMQ: .ASCIZ  /DATA CMP ERR LMT/
 71 040133      120     122     111     BSOUTQ: .ASCIZ  /PRINT ERRORS DETECTED WHILE READING BAD SEC FILE/
 74                                             .EVEN
    040214  000000                              .WORD   0
    040216  000000                              .WORD   0
    040220                                  L$LAST::
 75         000001                          .END
```

L10

Symbol table

```
ADR    = 000020 G      CKERLM  016616          C$GETW= 000027          DRSET  = 000010          FMT2    011730
AFMID    003212        CLKADR  003476          C$GMAN= 000043          DRTYPE  037466          FMT20   012431
AFMIDU   003214        CLKCSB= 172542          C$GPHR= 000042          DRVCNT  003076          FMT21   012461
ALLCYL= 000001         CLKCSR= 172540          C$GPRI= 000040          DRVERR= 040000          FMT22   012504
ALLSEC= 000002         CLKCTR= 172544          C$INIT= 000011          DRVNAM  006621          FMT23   012540
ANYERR= 100000         CLKFLG  003474          C$INLP= 000020          DRVNAV  006626          FMT24   012554
ARMID    003216        CLNCOD  016056 G        C$MANI= 000050          DSESTA= 000400          FMT25   012561
ARMIDU   003220        CLRBYT  002306          C$MAP  = 000102          DSMSK  = 001400          FMT26   012571
ASSEMB= 000010         CLRPAR  027524          C$MEM  = 000031          DSPCOD  014516 G        FMT27   012615
BADADD= 004000         CNT    = 000012          C$MMU  = 000103          EF.CON= 000036 G        FMT28   012634
BAMSK  = 000060        CNTYPE  037516          C$MSG  = 000023          EF.NEW= 000035 G        FMT3    011737
BANAM    006712        COMPOP= 007777          C$OPNR= 000034          EF.PWR= 000034 G        FMT4    011737
BASADD   006610        CONHNG= 000004          C$OPNW= 000104          EF.RES= 000037 G        FMT5    011750
BELL     011477        CONTIN  014744          C$PNTB= 000014          EF.STA= 000040 G        FMT6    011770
BHSTAT= 000010         COSTAT= 000040          C$PNTF= 000017          EF.XM  = 000033 G        FMT7    012032
BITO   = 000001 G      COUNT   003236          C$PNTS= 000016          ERHEAD  003014          FMT8    012102
BIT00  = 000001 G      CR     = 000015          C$PNTX= 000015          ERLIM  = 000010          FMT9    012134
BIT01  = 000002 G      CRDYMS= 000200          C$PUTB= 000072          ERLIMQ  040070          FOLWRT= 000100
BIT02  = 000004 G      CRLF    011623 G        C$PUTW= 000073          ERLIMW  014510          FRMWD   010153
BIT03  = 000010 G      CSNAM   006705          C$QIO  = 000377          ERRCNT  003244          FWDSKO= 002000
BIT04  = 000020 G      CSR    = 000000          C$RDBU= 000007          ERRPOI  003242          FWDSKS= 000400
BIT05  = 000040 G      CSRMSG  037432          C$REFG= 000047          ERRSWI  003020          F$AU   = 000015
BIT06  = 000100 G      CURCYL  003106          C$REL  = 000077          ERRVEC  003232          F$AUTO= 000020
BIT07  = 000200 G      CYLQ    037670          C$RESE= 000033          ERR1    012646 G        F$BGN  = 000040
BIT08  = 000400 G      CYLTBL  002606          C$REVI= 000004          ERR10   014242 G        F$CLEA= 000007
BIT09  = 001000 G      CYLUP  = 000004          C$RFLA= 000021          ERR2    012714 G        F$DU   = 000016
BIT1   = 000002 G      CYLWD   010146          C$RPT  = 000025          ERR3    012762 G        F$END  = 000041
BIT10  = 002000 G      C$AU   = 000052          C$SEFG= 000046          ERR4    013030 G        F$HARD= 000004
BIT11  = 004000 G      C$AUTO= 000061          C$SPRI= 000041          ERR5    013100 G        F$HW   = 000013
BIT12  = 010000 G      C$BRK  = 000022          C$SVEC= 000037          ERR6    013150 G        F$INIT= 000006
BIT13  = 020000 G      C$BSEG= 000004          C$TOME= 000076          ERR7    014032 G        F$JMP  = 000050
BIT14  = 040000 G      C$BSUB= 000002          C10MS   011556          ERR8    014102 G        F$MOD  = 000000
BIT15  = 100000 G      C$CLCK= 000062          C5SEC   011615          ERR9    014176 G        F$MSG  = 000011
BIT2   = 000004 G      C$CLEA= 000012          C500MS  011567          EVL    = 000004 G        F$PROT= 000021
BIT3   = 000010 G      C$CLOS= 000035          DANAM   006717          EXACYL  003226          F$PWR  = 000017
BIT4   = 000020 G      C$CLP1= 000006          DATACM= 000001          EXHCYL  003224          F$RPT  = 000012
BIT5   = 000040 G      C$CPBF= 000074          DATCOM  025032          EXOCYL  003222          F$SEG  = 000003
BIT6   = 000100 G      C$CPME= 000075          DATGEN  024672          EXROT   003230          F$SOFT= 000005
BIT7   = 000200 G      C$CVEC= 000036          DCKERR= 004000          E$END  = 002100          F$SRV  = 000010
BIT8   = 000400 G      C$DCLN= 000044          DCLIM  = 000012          E$LOAD= 000035          F$SUB  = 000002
BIT9   = 001000 G      C$DODU= 000051          DCLIMQ  040112          FCTBSF  003502          F$SW   = 000014
BOE    = 000400 G      C$DRPT= 000024          DCLIMW  014512          FLDBSF  004076          F$TEST= 000001
BRMSG    037455        C$DU   = 000053          DESDIF  003110          FMTOP1  011631          GBND    002312
BSCHK    026124        C$EDIT= 000001          DESHD   003114          FMTOP2  011660          GETPOS  024054
BSERR  = 000014        C$ERDF= 000055          DESSEC  003116          FMTOP3  011702          GETSTA= 000003
BSERRS   014514        C$ERHR= 000056          DESSGN  003112          FMTXT   011626 G        GLBDAT  002226 G
BSFLAG   003022        C$ERRO= 000060          DIAGMC= 000000          FMT1    011723          GLBEQA  002226 G
BSFNOT   010572        C$ERSF= 000054          DIFAUG  003100          FMT10   012141          GLBERR  012646 G
BSFVAL   003500        C$ERSO= 000057          DIFWD   010122          FMT11   012141          GLBSUB  016210 G
BSOUTQ   040133        C$ESCA= 000010          DIRBIT= 000004          FMT12   012147          GLBTXT  005750 G
BYPSNM   010161        C$ESEG= 000005          DIRMSK  002316          FMT13   012155          GSTAT   017214
CAFDT    011604        C$ESUB= 000003          DLTERR= 010000          FMT14   012221          GSTATC  017200
CAMSK    002314        C$ETST= 000001          DONE    003010          FMT15   012253          GSTATG  017224
CCYLUP   011575        C$EXIT= 000032          DRDYMS= 000001          FMT16   012307          GSTATR  017164
CHOSHD   021504        C$FREQ= 000101          DRMSG   037510          FMT17   012320          GTSTAT= 000104
CKBSVD   021614        C$FRME= 000100          DRSB   = 000010          FMT18   012342          G$CNTO= 000200
CKDATA= 000102         C$GETB= 000026          DRSELT= 000004          FMT19   012374          G$DELM= 000372
```

M10

Symbol table

```
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO  = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES = 000010
HADONE  003012
HCESTA= 040000
HCRCER= 004000
HDALIG= 000010
HDCYL   002320
HDHSEL= 000100
HDMOVF  010003
HDRCMP= 000002
HDR40 = 100000
HDSEC = 000077
HDSEL = 000020
HDWD    010135
HDWRD1  003054
HDWRD2  003056
HDWRD3  003060
HEAD  = 000006
HEADLM= 010000
HEADQ   040023
HEADV   040045
HEADW   014506
HFIN    003172
HFINU   003174
HFOUT   003176
HFOUTU  003200
HICYL = 020000
HILIM = 000004
HILIMQ  040002
HILIMW  014504
HLMTW   002304
HNFERR= 010000
HOE   = 100000 G
HOSTAT= 000020
HPTCOD  014460 G
HRDPRM  037350 G
HRDWTS  027554 G
HRIN    003202
HRINU   003204
HROUT   003206
HROUTU  003210
HSMSK = 000100
HSSTAT= 000100
IBE   = 010000 G
IBUFF   004472
IDU   = 000040 G
IER   = 020000 G
INITCO  014540 G
INOUTS= 000020
INTEBL= 000100
INTHLR  016536
ISR   = 000100 G
IXE   = 004000 G
I$AU  = 000041
I$AUTO= 000041
I$CLN = 000041
I$DU  = 000041
I$HRD = 000041
I$INIT= 000041
I$MOD = 000041
I$MSG = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR = 000041
I$RPT = 000041
I$SEG = 000041
I$SETU= 000041
I$SFT = 000041
I$SRV = 000041
I$SUB = 000041
I$TST = 000041
JUNK    002302
J$JMP = 000167
LABACF  007753
LABACR  007767
LABEXP  007666
LABHCF  007723
LABHCR  007737
LABIN   007643
LABMID  007651
LABOCF  007677
LABOCR  007711
LABOUT  007660
LAB1    006731
LAB2    006744
LF    = 000012
LIMVAL  037774
LOCERR  003450
LOCYL = 040000
LOE   = 040000 G
LOLIM = 000002
LOLIMQ  037755
LOLIMW  014502
LOT   = 000010 G
L$ACP   002110 G
L$APT   002036 G
L$AUT   002070 G
L$AUTO  015520 G
L$CCP   002106 G
L$CLEA  016056 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  002122 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  014520 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DU    016204 G
L$DUT   002072 G
L$DVTY  002214 G
L$EF    002052 G
L$ENVI  002044 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  037352 G
L$HIME  002120 G
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    014462 G
L$ICP   002104 G
L$INIT  014540 G
L$LADP  002026 G
L$LAST  040220 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  014452 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$SOFT  037526 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    014500 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L.BA    003040
L.CS    003036
L.DA    003042
L.MP    003044
L10000  012712
L10001  012760
L10002  013026
L10003  013076
L10004  013146
L10005  014030
L10006  014100
L10007  014174
L10010  014240
L10011  014450
L10013  014476
L10014  014516
L10015  015516
L10016  016054
L10017  016202
L10020  016206
L10021  016534
L10022  016614
L10023  031472
L10024  031524
L10025  031746
L10026  031672
L10027  032562
L10030  033700
L10031  033614
L10032  034724
L10033  034600
L10034  036146
L10035  036062
L10036  037346
L10037  037262
L10040  037432
L10041  037670
MAJINC  003472
MANQ    037721
MAPROX  007633
MBADAD  006412
MBSETO= 000001
MCERR   010713
MCONHN  007056
MCYLOC  011267
MCYLUP  006150
MDATCP  006032
MDCRC   010735
MDHEDR  002000 G
MDLT    010762
MDRDY   010702
MDRERR  011024
MDRRES  007005
MDRVST  011052
MDSERR  011035
MERRS   011472
MEXERS  011435
MFBSF   006433
MFLERR  011214
MFMTER  006563
MFOLWR  006220
MFWDSK  006271
MFWSKO  006322
MGTSTA  006020
MHCERR  011134
MHCRC   010725
MHDERR  011177
MHDRCP  006051
MHFCRC  010774
MHNF    010746
MININC  003462
MINOUT  006177
MISWI = 000000
MISWIW  014500
MITEST= 100000
MNDRST  011274
MNEERR  011242
MNOCLR  007072
MNOINT  007023
MOPEP   006117
MOPERR  011167
MORECE  003016
MOUTIN  006160
MPNAM   006724
MQUALS= 003760
MREAD   005754
MREADH  005765
MRESKO  006356
MREVSK  006240
MRLFAL  011364
MRSLT   006126
MSEEK   005750
MSPERR  011065
MSTERR  011120
MTOSLO  006765
MUBSF   006510
MULOAD  006137
MUNDEF  011317
MWDERR  011152
MWGERR  011103
MWORD   006757
MWRCHK  005775
MWRITE  006006
MWRSET  006103
MWRTAB  011423
M40HDR  006067
NEWCYL  003104
NOCLR = 000010
NOCTLR  010251
NOERCT  003451
NOHD1   010461
NOIRPT= 000002
NOOP  = 000100
NOPWR   006645
NOTRDY  010307
NOTST   010364
NXMERR= 020000
NXTHL   002310
NXTPAS  014764
OBUFF   005072
OFIN    003142
OFINU   003144
OFMID   003146
OFMIDU  003150
OFOUT   003152
```

N10

Symbol table

```
OFOUTU  003154          PRI06 = 000300 G        SECWD   010141          T$GMAN= 000000          T3265$  034724
OLDCYL  003102          PRI07 = 000340 G        SEEK    = 000106          T$HILI= 000377          T33TBL  002506
ONSWAP  021570          PSETNM  003446         SEEKOP= 010000          T$LAST= 000001          T3300$  035102
OPFLAG  003006          PWCON   015246          SEQMES  010174          T$LOLI= 000001          T3301$  035364
OPIERR= 002000          PWRFLG  003454          SETDON  015020          T$LSYM= 010000          T3365$  036146
OPMSGS  002226          P2T03E  007156          SFTPRM  037524 G        T$LTNO= 000010          T3400$  036310
OPR004  010105          P2T04E  007174          SGNWD   010130          T$NEST= 177777          T3401$  036534
OPR1A   010056          P2T05E  007214          SKTMES  007553          T$NSO  = 000000          T3465$  037346
OPR1B   010062          P2T06E  007234          SPDSTA= 004000          T$NS1  = 000005          T4      031750 G
OPR12   010037          P2T07E  007254          SPTCOD  014476 G        T$NS2  = 000002          T5      032564 G
ORIN    003156          P2T08E  007272          SRTMES  007565          T$PTNU= 000000          T5.1    033124
ORINU   003160          P2T09E  007312          SSINDX  003004          T$SAVL= 177777          T6      033702 G
ORMID   003162          P2T10E  007315          STAMES  010217          T$SEGL= 177777          T6.1    033730
ORMIDU  003164          P2T11E  007330          STAMSK= 000007          T$SEKO= 010000          T7      034726 G
OROUT   003166          P2T12E  007343          STATE2  011526          T$SUBN= 000001          T7.1    035364
OROUTU  003170          P2T13E  007355          STATE3  011536          T$TAGL= 177777          T8      036150 G
OUTINS= 000040          P2T14E  007402          STATE5  011546          T$TAGN= 010042          T8.1    036534
O$APTS= 000000          P2T15E  007423          STOSTA= 010000          T$TEMP= 000000          UAM     = 000200 G
O$AU  = 000000          P2T16E  007446          SUBSTK  002406          T$TEST= 000010          ULOAD = 000010
O$BGNR= 000000          P2T17E  007467          SVCBGL= 000001          T$TSTM= 177777          UNDTST  010072
O$BGNS= 000001          P2T18E  007521          SVCGBL= 000000          T$TSTS= 000001          UNXERR  007133
O$DU  = 000001          P2T19E  007543          SVCINS= 000000          T$$AUT= 010016          VALDES  007607
O$ERRT= 000000          RDALHD  024324          SVCSUB= 000000          T$$CLE= 010017          VCNRST  007112
O$GNSW= 000001          RDBSF   021630          SVCTAG= 000000          T$$DU  = 010020          VCSTAT= 001000
O$POIN= 000001          RDDATA= 000114          SVCTST= 000000          T$$HAR= 010040          VECMSG  037446
O$SETU= 000000          RDHEAD= 000110          SWAPHD  021530          T$$HW  = 010013          VECT    = 000002
PART2 = 000001 G        RDNOHR= 000116          S$LSYM= 010000          T$$INI= 010015          VERHDR  023174
PASCNT  003234          RDYCHK  021230          TAG     003470          T$$MSG= 010011          VERPOS  024202
PASNEW  014772          RDYWAI  023570          TBLSTR  003026          T$$PRO= 010012          WAITIN  017010
PASNUM  003444          READRL  016756          TBT     002546          T$$SEG= 010000          WCMSK   = 017777
PATTBL  002362          RELDWT= 040000          TCERR   010230          T$$SOF= 010041          WCRNG   = 160000
PAT1    005472          RESE3   011503          TEMP    003464          T$$SRV= 010022          WDESTA= 100000
PAT10   005746          RESE4   011507          TEMP0   003120          T$$SUB= 010037          WGESTA= 002000
PAT2    005474          RESE5   011514          TEMP1   003122          T$$SW  = 010014          WLSTAT= 020000
PAT3    005534          RESE6   011521          TEMP2   003124          T$$TES= 010036          WRTSWI  003024
PAT4    005574          RESPAR  003064          TEMP3   003126          T.BA    003050          WTDATA= 000112
PAT5    005634          RESTAR  014734          TEMP4   003130          T.CS    003046          XDELAY  003456
PAT6    005642          RESTBL  002322          TEMP5   003132          T.DA    003052          XRDHD   022540
PAT7    005702          REVSK0= 001000          TEMP6   003134          T.DRIV  002300          XRDHDC  022530
PAT8    005704          REVSKS= 000200          TEMP7   003136          T.MP    003054          XRDHDG  022544
PAT9    005744          RLBA    = 000002          TEMP8   003140          T.STAT  003062          XREAD   025362
PH65$   021172          RLBAS   003030          TIME    016210          T1      027554 G        XREADG  025370
PNT   = 001000 G        RLCS    = 000000          TIM.US  003466          T2      031474 G        XSEEK   020112
POSHDS  020664          RLCSR   = 000000          TOSLOW= 000001          T25TBL  002432          XSEEKT  020102
POSHDO  023544          RLDA    = 000004          TRPFLG  003452          T25TB2  002460          XSEEK1  020116
POSHSB  023540          RLDRV   003034          TRPHAN  016530          T3      031526 G        XTIME   016354
POSHW1  023532          RLMP    = 000006          TSTINT  017146          T3.1    031632          XWRITE  025322
PRI   = 002000 G        RLVEC   003032          TSTLAB  007150          T306$   031600          XWRITT  025312
PRIOR = 000004          RORWOP= 020000          TSTNM   003240          T3065$  031746          XWRIT1  025326
PRI00 = 000000 G        RPTOP   026274          TYPDR   = 000006          T307$   031632          X$ALWA= 000000
PRI01 = 000040 G        RPTREM  027270          T$ARGC= 000007          T310$   031640          X$FALS= 000040
PRI02 = 000100 G        RPTRES  027062          T$CODE= 006130          T3100$  032716          X$OFFS= 000400
PRI03 = 000140 G        RSTRT   014652          T$ERRN= 003247          T3101$  033124          X$TRUE= 000020
PRI04 = 000200 G        SAMSK   = 000077          T$EXCP= 000000          T3165$  033700          YDELAY  003460
PRI05 = 000240 G        SECQ    037704          T$FLAG= 000040          T3204$  034602

. ABS.  040220    000   (RW,I,GBL,ABS,OVR)
```

B11


```
       000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 964
Work  file writes: 775
Size of work file: 35176 Words  ( 138 Pages)
Size of core pool: 14080 Words  ( 55 Pages)
Operating  system: RT-11 (Under RTEM-11)

Elapsed time: 00:05:26.00
CZRLNC.BIC,CZRLNC.LST/C=CZRLNC.DOC,CZRLNC.MAC,SVC41R.MLB/M
```

```
ADR      39-15#
AFMID    39-418#  58-191  58-205
AFMIDU   39-419#  58-192
ALLCYL   39-34#   62-18   62-32   64-18   64-37   65-18   65-34
ALLSEC   39-35#
ANYERR   39-87#   49-92   52-33   55-127
ARMID    39-420#  58-193  58-206
ARMIDU   39-421#  58-194
ASSEMB   37-10    37-10
BADADD   39-66#   57-12   57-55   57-57
BAMSK    39-98#
BANAM    39-599#  57-250
BASADD   39-594#  41-244  43-133  44-19   44-33   47-9    53-113  57-247  58-199  61-87   63-38   63-106
BELL     39-736#  63-45
BHSTAT   39-124#
BIT0     39-15#   62-105  62-113  64-116  64-124  64-150  65-118  65-126  65-165
BIT00    39-15    39-15#  39-34   39-56   39-75   57-196
BIT01    39-15    39-15#  39-35   39-55   39-76
BIT02    39-15    39-15#  39-36   39-57   39-77
BIT03    39-15    39-15#  39-37   39-58   39-78
BIT04    39-15    39-15#  39-59
BIT05    39-15    39-15#  39-60
BIT06    39-15    39-15#  39-61
BIT07    39-15    39-15#  39-62
BIT08    39-15    39-15#  39-63
BIT09    39-15    39-15#  39-64
BIT1     39-15#   62-118  62-135  64-129  64-165  65-131  65-152
BIT10    39-15#   39-65   49-64   50-60   54-29   55-223  57-30   58-46   63-56
BIT11    39-15#   39-66
BIT12    39-15#   39-38   39-67
BIT13    39-15#   39-39   39-68
BIT14    39-15#   39-40   39-69
BIT15    39-15#   39-41   39-70   41-140
BIT2     39-15#   57-31
BIT3     39-15#
BIT4     39-15#
BIT5     39-15#
BIT6     39-15#   53-68   57-135
BIT7     39-15#
BIT8     39-15#   53-66
BIT9     39-15#   41-96
BOE      39-15#
BRMSG    66-10    66-16#
BSCHK    57-118#  60-48   62-99   64-114  65-106
BSERR    39-31#   66-47   66-47   66-47
BSERRS   42-36#   53-80   53-84   57-99
BSFLAG   39-350#  57-121* 57-142* 57-150
BSFNOT   39-699#  53-112
BSFVAL   39-454#  43-64*  53-5    53-107* 53-111* 59-10*
BSOUTQ   66-47    66-71#
BYPSNM   39-691#  63-48
C$AU     37-10#
C$AUTO   37-10#   44-38
C$BRK    37-10#
C$BSEG   37-10#   51-16
C$BSUB   37-10#   60-27   62-67   63-13   64-83   65-75
```

```
C$CLCK  37-10#  43-9
C$CLEA  37-10#  45-19
C$CLOS  37-10#
C$CLP1  37-10#
C$CPBF  37-10#
C$CPME  37-10#
C$CVEC  37-10#  44-37  45-13  45-17
C$DCLN  37-10#  43-136  47-12
C$DODU  37-10#  43-135  44-23  44-36  47-11
C$DRPT  37-10#
C$DU    37-10#  45-22
C$EDIT  37-10#  39-6
C$ERDF  37-10#
C$ERHR  37-10#  49-47  49-86  49-101  49-109  50-87  50-93  51-42  51-60  52-24  52-35  53-83  53-97  54-43
        54-55   54-60  54-70  55-29   55-42   55-49  55-114 55-124 55-129 55-191 55-244 55-250 56-43  57-59
        57-75   57-82  57-91  57-101  58-57   58-62  61-46  61-51  61-68  61-73  63-73  63-80  63-84  63-91
C$ERRO  37-10#
C$ERSF  37-10#
C$ERSO  37-10#
C$ESCA  37-10#  60-39  62-154  64-171  65-171
C$ESEG  37-10#  51-67
C$ESUB  37-10#  60-37  62-152  63-103  64-169  65-169
C$ETST  37-10#  58-208  59-12  60-55  61-90  62-168  63-117  64-186  65-185
C$EXIT  37-10#  61-14  63-4   63-74
C$FREQ  37-10#
C$FRME  37-10#
C$GETB  37-10#
C$GETW  37-10#
C$GMAN  37-10#
C$GPHP  37-10#  43-70
C$GPRI  37-10#
C$INIT  37-10#  43-139
C$INLP  37-10#  47-6  51-18  53-86
C$MANI  37-10#  43-14
C$MAP   37-10#
C$MEM   37-10#
C$MMU   37-10#
C$MSG   37-10#  41-13  41-27  41-41  41-56  41-71  41-186  41-200  41-222  41-236  41-252
C$OPNR  37-10#
C$OPNW  37-10#
C$PNTB  37-10#  41-116  41-157  41-171  41-180  41-244  41-245  41-248  56-28  57-166  57-167  57-171  57-184  57-202
        57-206  57-211  57-214  57-226  57-235  57-236  57-239  57-247  57-250  57-251  57-252
C$PNTF  37-10#  43-132  43-133  43-134  44-18  44-19  44-21  44-31  44-33  44-35  47-8  47-9  47-10  53-19
        53-112  53-113  53-114  58-16  58-198  58-199  58-200  58-201  58-202  58-203  58-204  58-205  58-206  61-12
        61-86   61-87  61-88  63-38  63-45  63-48  63-106
C$PNTS  37-10#
C$PNTX  37-10#
C$PUTB  37-10#
C$PUTW  37-10#
C$QIO   37-10#
C$RDBU  37-10#  46-9  46-29
C$REFG  37-10#  43-19  43-24  43-46  43-49  43-53
C$REL   37-10#
C$RESE  37-10#  37-10#  43-13  45-18
C$REVI  37-10#  39-6
C$RFLA  37-10#
```

E11

F11

```
EXHCYL  39-424#  58-203  58-204
EXOCYL  39-423#  58-201  58-202
EXROT   39-426#  61-88
F$AU    37-10#
F$AUTO  37-10#   44-10   44-38
F$BGN   37-10#   39-5    39-7    39-13   39-142  39-146  39-560  39-564  39-791  40-3    41-1    41-15   41-29   41-43
        41-58    41-73   41-188  41-202  41-224  41-238  41-253  42-3    42-10   42-19   42-21   42-38   42-40   42-47
        43-3     43-4    43-140  44-10   45-3    45-4    45-20   45-23   46-3    46-43   46-50   51-16   57-266  58-3
        58-7     58-208  59-3    59-12   60-3    60-27   60-27   60-37   60-39   60-55   61-3    61-14   61-90   62-3
        62-67    62-67   62-152  62-154  62-168  63-3    63-13   63-13   63-49   63-74   63-103  63-117  64-3    64-83
        64-83    64-165  64-171  64-186  65-3    65-75   65-75   65-169  65-171  65-185  65-186  66-3    66-4    66-20
        66-22    66-23   66-73
F$CLEA  37-10#   45-4    45-19
F$DU    37-10#   45-20   45-22
F$END   37-10    37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10   37-10
        37-10    37-10   37-10#  39-5    39-7    39-13   39-142  39-146  39-560  39-564  39-791  40-3    41-13   41-27
        41-41    41-56   41-71   41-186  41-200  41-222  41-236  41-252  41-253  42-10   42-19   42-21   42-38   42-40
        42-47    43-3    43-139  43-140  44-38   45-3    45-19   45-22   45-23   46-3    46-49   46-61   51-67   57-266
        58-3     58-7    58-7    58-7    58-208  58-208  59-3    59-3    59-3    59-12   59-12   60-3    60-3    60-3
        60-27    60-27   60-37   60-37   60-39   60-55   60-55   61-3    61-3    61-3    61-14   61-90   61-90   62-3
        62-3     62-3    62-67   62-67   62-152  62-152  62-154  62-168  62-168  63-3    63-3    63-3    63-13   63-13
        63-49    63-74   63-103  63-103  63-117  63-117  64-3    64-3    64-3    64-83   64-83   64-169  64-169  64-171
        64-186   64-186  65-3    65-3    65-3    65-75   65-75   65-169  65-169  65-171  65-185  65-185  65-186  66-3
        66-12    66-20   66-22   66-49   66-73
F$HARD  37-10#   66-4    66-12   66-35   66-38   66-41
F$HW    37-10#   42-11   42-18
F$INIT  37-10#   43-4    43-139
F$JMP   37-10#   61-14   63-49   63-74
F$MOD   37-10#   39-5    39-7    39-13   39-142  39-146  39-560  39-564  39-791  40-3    41-253  42-10   42-19   42-21
        42-38    42-40   42-47   43-3    43-140  45-3    45-23   46-3    57-266  58-3    65-186  66-3    66-20   66-22
        66-73
F$MSG   37-10#   41-1    41-13   41-15   41-27   41-29   41-41   41-43   41-56   41-58   41-71   41-73   41-186  41-188
        41-200   41-202  41-222  41-224  41-236  41-238  41-252
F$PROT  37-10#   42-3    42-7
F$PWR   37-10#
F$RPT   37-10#
F$SEG   37-10#   51-16   51-67
F$SOFT  37-10#   66-23   66-35   66-38   66-41   66-49
F$SRV   37-10#   46-43   46-49   46-50   46-61
F$SUB   37-10#   60-27   60-37   62-67   62-152  63-13   63-103  64-83   64-169  65-75   65-169
F$SW    37-10#   42-22   42-37
F$TEST  37-10#   58-7    58-208  59-3    59-12   60-3    60-55   61-3    61-90   62-3    62-168  63-3    63-117  64-3
        64-186   65-3    65-185
FCTBSF  39-456#  53-31   53-115* 57-122
FLDBSF  39-458#  53-75   53-93   53-98   53-116* 57-125  57-145
FMT1    39-762#  57-184  57-239
FMT10   39-771#
FMT11   39-772#  57-229
FMT12   39-773#  57-232
FMT13   39-774#  57-211
FMT14   39-775#  41-245
FMT15   39-776#  41-171  41-248
FMT16   39-777#  57-167
FMT17   39-778#  41-157
FMT18   39-779#  58-200
FMT19   39-780#  58-201  58-202
```

H11


```
FMT2     39-763#  57-226   58-198   61-86
FMT20    39-781#  58-203   58-204
FMT21    39-782#  58-205   58-206
FMT22    39-783#  57-214
FMT23    39-784#  63-48
FMT24    39-785#  43-132   44-18    44-31
FMT25    39-786#  47-8
FMT26    39-787#  61-88
FMT27    39-788#  41-180   56-28
FMT28    39-789#  41-116
FMT3     39-764#
FMT4     39-765#  57-171
FMT5     39-766#  41-244   43-133   44-19    44-33    47-9     53-113   57-247   58-199   61-87
FMT6     39-767#  57-250
FMT7     39-768#  57-252
FMT8     39-769#  57-251
FMT9     39-770#  57-166
FMTOP1   39-759#  63-38    63-106
FMTOP2   39-760#
FMTOP3   39-761#
FMTXT    39-758#  57-202   57-206   63-45
FOLWRT   39-61#   39-71    62-114   64-125   65-127
FRMWD    39-690#  57-211
FWDSKO   39-65#   39-71    62-142   65-158
FWDSKS   39-63#   39-71    64-164
G$CNTO   37-10#
G$DELM   37-10#   46-11    46-17    46-32    46-37
G$DISP   37-10#
G$EXCP   37-10#
G$HILI   37-10#
G$LOLI   37-10#
G$NO     37-10#
G$OFFS   37-10#   66-5     66-6     66-7     66-8     66-9     66-10    66-25    66-26    66-32    66-34    66-36    66-37    66-39
         66-40    66-42    66-44    66-46    66-47
G$OFSI   37-10#   66-5     66-6     66-7     66-8     66-9     66-10    66-25    66-26    66-32    66-34    66-36    66-37    66-39
         66-40    66-42    66-44    66-46    66-47
G$PRMA   37-10#   66-6     66-7
G$PRMD   37-10#   66-8     66-10    66-36    66-39    66-42    66-44    66-46
G$PRML   37-10#   66-5     66-9     66-25    66-26    66-32    66-34    66-37    66-40    66-47
G$RADA   37-10#
G$RADB   37-10#
G$RADD   37-10#   66-36    66-39    66-42    66-44    66-46
G$RADL   37-10#   66-5     66-9     66-25    66-26    66-32    66-34    66-37    66-40    66-47
G$RADO   37-10#   66-6     66-7     66-8     66-10
G$XFER   37-10#   66-35    66-38    66-41
G$YES    37-10#   66-5     66-6     66-7     66-8     66-9     66-10    66-25    66-26    66-32    66-34    66-36    66-37    66-39
         66-40    66-42    66-44    66-46    66-47
GBND     39-177#  43-86*   43-94*
GETPOS   50-19    51-13    51-20    51-46    55-147#  55-187
GETSTA   39-111#  41-89    49-4     49-8     49-57
GLBDAT   39-146#
GLBEQA   39-13#
GLBERR   40-3#
GLBSUB   46-3#
GLBTXT   39-564#
GSTAT    49-11#   49-28    49-83    49-90    52-15    52-26    54-45    55-104   55-116   57-84    63-75
```

I11

Cross reference table (CREF V05.03)

J11

K11


```
L$EXP1   39-6#
L$EXP4   39-6#
L$EXP5   39-6#
L$HARD   39-6      66-4      66-4#
L$HIME   39-6#
L$HPCP   39-6#
L$HPTP   39-6#
L$HW     39-6      42-11     42-11#
L$ICP    39-6#
L$INIT   39-6      43-4#
L$LADP   39-6#
L$LAST   39-6      66-74#
L$LOAD   39-6#
L$LUN    39-6#
L$MREV   39-6#
L$NAME   39-6#
L$PRIO   39-6#
L$PROT   39-6      42-3#
L$PRT    39-6#
L$REPP   39-6#
L$REV    39-6#
L$SOFT   39-6      66-23     66-23#
L$SPC    39-6#
L$SPCP   39-6#
L$SPTP   39-6#
L$STA    39-6#
L$SW     39-6      42-22     42-22#
L$TEST   39-6#
L$TIML   39-6#
L$UNIT   39-6#     43-21     43-30     43-61
L.BA     39-359#   55-224*   55-230    57-251
L.CS     39-358#   49-63*    49-64*    49-65*    49-67     49-113    50-57     54-27     54-74     55-221*   55-222*   55-223*   55-235    57-27
         57-173    57-251    58-50     58-73*    63-53
L.DA     39-360#   49-57*    49-60*    49-66     49-73     55-225*   55-228*   55-229    57-187    57-251    58-49
L.MP     39-361#   49-53     54-19     57-251    61-34     61-54
L10000   41-13#
L10001   41-27#
L10002   41-41#
L10003   41-56#
L10004   41-71#
L10005   41-186#
L10006   41-200#
L10007   41-222#
L10010   41-236#
L10011   41-252#
L10013   42-11     42-18#
L10014   42-22     42-37#
L10015   43-139#
L10016   44-38#
L10017   45-19#
L10020   45-22#
L10021   46-49#
L10022   46-61#
L10023   58-208#
L10024   59-12#
L10025   60-39     60-55#
```

L11


```
L10026  60-37#
L10027  61-14    61-90#
L10030  62-154   62-168#
L10031  62-152#
L10032  63-49    63-117#
L10033  63-74    63-103#
L10034  64-171   64-186#
L10035  64-169#
L10036  65-171   65-185#
L10037  65-169#
L10040  66-4     66-12#
L10041  66-23    66-49#
LAB1    39-602#  57-251
LAB2    39-603#  57-252
LABACF  39-659#  58-205
LABACR  39-660#  58-206
LABEXP  39-654#  58-200
LABHCF  39-657#  58-203
LABHCR  39-658#  58-204
LABIN   39-651#  58-200
LABMID  39-652#  58-200
LABOCF  39-655#  58-201
LABOCR  39-656#  58-202
LABOUT  39-653#  58-200
LF      39-567   39-697   39-698
LIMVAL  66-36    66-39    66-63#
LOCERR  39-438#  53-34*   53-91*   53-105*  53-108
LOCYL   39-40#   43-40
LOE     39-15#
LOLIM   39-26#   66-36    66-36    66-36
LOLIMQ  66-34    66-62#
LOLIMW  42-31#   43-42*   61-25    62-24    62-53    64-26    64-64    65-26    65-61
LOT     39-15#
M40HDR  39-170   39-577#
MAJINC  39-448#  46-25*   46-40
MANQ    66-32    66-60#
MAPROX  39-650#  61-88
MBADAD  39-169   39-590#
MBSETO  39-106#  50-73
MCERR   39-183   39-703#
MCONHN  39-608#  48-19
MCYLOC  39-729#  41-208   57-230
MCYLUP  39-160   39-582#
MDATCP  39-158   39-575#
MDCRC   39-705#  41-132
MDHEDR  39-5#
MDLT    39-707#  41-137
MDRDY   39-702#  52-22    54-41    55-113   57-80
MDRERR  39-184   39-709#  63-79
MDRRES  39-606#  48-13
MDRVST  39-719#  41-193
MDSERR  39-198   39-718#
MERRS   39-735#  41-116
MEXERS  39-734#  47-8
MFBSF   39-591#  53-92
MFLERR  39-186   39-727#
```

M11

Cross reference table (CREF V05.03)

C12

RESPAR	39-372#	57-224	57-257	57-262										
RESTAR	43-25	43-45#												
RESTBL	39-183#	41-150												
REVSKO	39-64#	39-71	65-155											
REVSKS	39-62#	39-71	64-161											
RLBA	39-81#	47-18	50-80*	54-34*	55-230*	57-68*	61-39*	61-60*	63-64*					
RLBAS	39-354#	41-244	43-69	43-106	43-133	44-14	44-19	44-33	46-59	47-9	53-113	55-219	57-247	58-199
	61-87	63-38	63-106											
RLCS	39-80#	41-92*	41-94	43-124*	43-126	44-15	44-28*	44-29	45-7	45-9*	45-10	48-16	49-24	49-78
	50-81*	55-231	55-235*	55-237	57-69*	58-50*	58-68*	61-40*	61-61*	63-65*				
RLCSR	39-86#	47-17	48-5	49-67*	54-35*									
RLDA	39-82#	41-89*	47-19	49-66*	50-79*	54-33*	55-229*	57-67*	58-49*	61-38*	61-59*	63-63*		
RLDRV	39-356#	41-91	41-244	43-122	43-133	44-19	44-26	44-33	45-9	47-9	49-63	50-59	53-113	54-28
	55-222	57-29	57-247	58-45	58-199	61-87	63-38	63-48	63-55	63-106				
RLMP	39-83#	41-99	47-20	54-64	54-65	55-220	57-66*	61-37*	61-58*	63-62*				
RLVEC	39-355#	43-104	45-13											
RORWOP	39-68#	57-172	57-180	57-183	57-212									
RPTOP	41-6	41-18	41-32	41-46	41-61	41-81	41-191	41-206	41-227	41-243	57-162#			
RPTREM	41-10	41-24	41-38	41-53	41-68	41-154	41-197	41-218	41-233	57-247#				
RPTRES	41-9	41-23	41-37	41-52	41-67	41-151	41-196	41-217	41-232	57-221#				
RSTRT	43-31#	43-47												
S$LSYM	37-10#	41-13#	41-27#	41-41#	41-56#	41-71#	41-186#	41-200#	41-222#	41-236#	41-252#	42-18#	42-37#	43-139#
	44-38#	45-19#	45-22#	46-49#	46-61#	51-16	51-16	51-16#	58-208#	59-12#	60-37#	60-55#	61-90#	62-152#
	62-168#	63-103#	63-117#	64-169#	64-186#	65-169#	65-185#	66-12#	66-49#					
SAMSK	39-102#													
SECQ	66-26	66-53#												
SECWD	39-688#	57-214												
SEEK	39-46#	50-58												
SEEKOP	39-67#	57-172	57-177	57-209										
SEQMES	39-692#	57-166												
SETDON	43-43	43-57	43-64#											
SFTPRM	66-22#													
SGNWD	39-686#	57-211												
SKTMES	39-647#	58-198												
SPDSTA	39-131#	49-34												
SPTCOD	42-21#													
SRTMES	39-648#	61-86												
SSINDX	39-341#	43-18*	49-14	49-18*	49-118*	50-8	50-12*	50-95*	51-5	51-9*	51-69*	52-5	52-9*	52-38*
	54-8	54-12*	54-79*	55-5	55-9*	55-63*	55-95	55-99*	55-132*	55-148	55-152*	55-164*	55-180	55-184*
	55-193*	55-207	55-211*	55-259*	56-5	56-9*	56-29*	57-17	57-21*	57-103*	57-163	57-169		
STAMES	39-693#	41-157												
STAMSK	39-123#	49-72												
STATE2	39-745#													
STATE3	39-746#													
STATE5	39-747#													
STOSTA	39-132#													
SUBSTK	39-214#	49-16*	49-17*	50-10*	50-11*	51-7*	51-8*	52-7*	52-8*	54-10*	54-11*	55-7*	55-8*	55-97*
	55-98*	55-150*	55-151*	55-182*	55-183*	55-209*	55-210*	56-7*	56-8*	57-19*	57-20*	57-167		
SVCBGL	37-13#													
SVCGBL	37-10#	39-5	39-5	39-5	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6

	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-8	39-8	39-8	39-9	39-9	39-9	39-13	39-13
	39-13	39-146	39-146	39-146	39-564	39-564	39-564	40-3	40-3	40-3	41-1	41-1	41-1	41-15
	41-15	41-15	41-29	41-29	41-29	41-43	41-43	41-43	41-58	41-58	41-58	41-73	41-73	41-73
	41-188	41-188	41-188	41-202	41-202	41-202	41-224	41-224	41-224	41-238	41-238	41-238	42-3	42-3
	42-3	42-10	42-10	42-10	42-11	42-11	42-11	42-21	42-21	42-21	42-22	42-22	42-22	42-40
	42-40	42-40	42-45	42-45	42-45	43-3	43-3	43-3	43-4	43-4	43-4	44-10	44-10	44-10
	45-3	45-3	45-3	45-4	45-4	45-4	45-20	45-20	45-20	46-3	46-3	46-3	58-3	58-3
	58-3	66-3	66-3	66-3	66-4	66-4	66-4	66-22	66-22	66-22	66-23	66-23	66-23	66-74
	66-74	66-74	66-74#											
SVCINS	37-10#	37-14#	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-8	39-8
	39-8	39-8	39-9	39-9	39-9	39-9	41-13	41-13	41-27	41-27	41-41	41-41	41-56	41-56
	41-71	41-71	41-116	41-116	41-116	41-116	41-116	41-116	41-116	41-116	41-116	41-116	41-116	41-116
	41-116	41-116	41-116	41-116	41-157	41-157	41-157	41-157	41-157	41-157	41-157	41-157	41-157	41-157
	41-157	41-157	41-157	41-157	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171
	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-180	41-180
	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180
	41-180	41-180	41-186	41-186	41-200	41-200	41-222	41-222	41-236	41-236	41-244	41-244	41-244	41-244
	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244
	41-244	41-244	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245
	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-248	41-248
	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248
	41-248	41-248	41-248	41-248	41-248	41-248	41-252	41-252	42-11	42-11	42-22	42-22	42-45	42-45
	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45
	42-45	42-45	43-9	43-9	43-9	43-9	43-9	43-9	43-10	43-10	43-12	43-12	43-12	43-12
	43-13	43-13	43-14	43-14	43-15	43-15	43-19	43-19	43-19	43-19	43-20	43-20	43-24	43-24
	43-24	43-24	43-25	43-25	43-46	43-46	43-46	43-46	43-47	43-47	43-49	43-49	43-49	43-49
	43-50	43-50	43-53	43-53	43-53	43-53	43-54	43-54	43-70	43-70	43-70	43-70	43-71	43-71
	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-104	43-105	43-105
	43-105	43-105	43-132	43-132	43-132	43-132	43-132	43-132	43-132	43-132	43-132	43-132	43-132	43-132
	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133	43-133
	43-133	43-133	43-133	43-133	43-133	43-133	43-134	43-134	43-134	43-134	43-134	43-134	43-134	43-134
	43-134	43-134	43-135	43-135	43-135	43-135	43-136	43-136	43-139	43-139	44-12	44-12	44-12	44-12
	44-12	44-12	44-12	44-12	44-12	44-12	44-12	44-12	44-18	44-18	44-18	44-18	44-18	44-18
	44-18	44-18	44-18	44-18	44-18	44-18	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19
	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-19	44-21	44-21
	44-21	44-21	44-21	44-21	44-21	44-21	44-21	44-21	44-23	44-23	44-23	44-23	44-31	44-31
	44-31	44-31	44-31	44-31	44-31	44-31	44-31	44-31	44-31	44-31	44-33	44-33	44-33	44-33
	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33	44-33
	44-33	44-33	44-35	44-35	44-35	44-35	44-35	44-35	44-35	44-35	44-35	44-35	44-36	44-36
	44-36	44-36	44-37	44-37	44-37	44-37	44-38	44-38	45-5	45-5	45-5	45-5	45-5	45-5
	45-5	45-5	45-5	45-5	45-5	45-5	45-6	45-6	45-6	45-6	45-13	45-13	45-13	45-13
	45-17	45-17	45-17	45-17	45-18	45-18	45-19	45-19	45-22	45-22	46-9	46-9	46-10	46-10
	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11	46-11
	46-11	46-11	46-17	46-17	46-17	46-17	46-17	46-17	46-17	46-17	46-17	46-17	46-17	46-17
	46-17	46-17	46-17	46-17	46-29	46-29	46-30	46-30	46-32	46-32	46-32	46-32	46-32	46-32
	46-32	46-32	46-32	46-32	46-32	46-32	46-32	46-32	46-32	46-32	46-37	46-37	46-37	46-37
	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-37	46-49	46-49
	46-61	46-61	47-6	47-6	47-7	47-7	47-8	47-8	47-8	47-8	47-8	47-8	47-8	47-8
	47-8	47-8	47-8	47-8	47-8	47-8	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9

E12

47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-9	47-10	47-10
47-10	47-10	47-10	47-10	47-10	47-10	47-10	47-10	47-11	47-11	47-11	47-11	47-12	47-12
49-47	49-47	49-47	49-47	49-47	49-47	49-47	49-47	49-86	49-86	49-86	49-86	49-86	49-86
49-86	49-86	49-101	49-101	49-101	49-101	49-101	49-101	49-101	49-101	49-109	49-109	49-109	49-109
49-109	49-109	49-109	49-109	50-87	50-87	50-87	50-87	50-87	50-87	50-87	50-87	50-93	50-93
50-93	50-93	50-93	50-93	50-93	50-93	51-16	51-16	51-18	51-18	51-19	51-19	51-42	51-42
51-42	51-42	51-42	51-42	51-42	51-42	51-60	51-60	51-60	51-60	51-60	51-60	51-60	51-60
51-67	51-67	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-35	52-35	52-35	52-35
52-35	52-35	52-35	52-35	53-19	53-19	53-19	53-19	53-19	53-19	53-19	53-19	53-19	53-19
53-19	53-19	53-83	53-83	53-83	53-83	53-83	53-83	53-83	53-83	53-86	53-86	53-87	53-87
53-97	53-97	53-97	53-97	53-97	53-97	53-97	53-97	53-112	53-112	53-112	53-112	53-112	53-112
53-112	53-112	53-112	53-112	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113
53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-113	53-114	53-114	53-114	53-114
53-114	53-114	53-114	53-114	53-114	53-114	54-43	54-43	54-43	54-43	54-43	54-43	54-43	54-43
54-55	54-55	54-55	54-55	54-55	54-55	54-55	54-55	54-60	54-60	54-60	54-60	54-60	54-60
54-60	54-60	54-70	54-70	54-70	54-70	54-70	54-70	54-70	54-70	55-29	55-29	55-29	55-29
55-29	55-29	55-29	55-29	55-42	55-42	55-42	55-42	55-42	55-42	55-42	55-42	55-49	55-49
55-49	55-49	55-49	55-49	55-49	55-49	55-114	55-114	55-114	55-114	55-114	55-114	55-114	55-114
55-124	55-124	55-124	55-124	55-124	55-124	55-124	55-124	55-129	55-129	55-129	55-129	55-129	55-129
55-129	55-129	55-191	55-191	55-191	55-191	55-191	55-191	55-191	55-191	55-244	55-244	55-244	55-244
55-244	55-244	55-244	55-244	55-250	55-250	55-250	55-250	55-250	55-250	55-250	55-250	56-28	56-28
56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28	56-28
56-28	56-28	56-43	56-43	56-43	56-43	56-43	56-43	56-43	56-43	57-59	57-59	57-59	57-59
57-59	57-59	57-59	57-59	57-75	57-75	57-75	57-75	57-75	57-75	57-75	57-75	57-82	57-82
57-82	57-82	57-82	57-82	57-82	57-82	57-91	57-91	57-91	57-91	57-91	57-91	57-91	57-91
57-101	57-101	57-101	57-101	57-101	57-101	57-101	57-101	57-166	57-166	57-166	57-166	57-166	57-166
57-166	57-166	57-166	57-166	57-166	57-166	57-167	57-167	57-167	57-167	57-167	57-167	57-167	57-167
57-167	57-167	57-167	57-167	57-171	57-171	57-171	57-171	57-171	57-171	57-171	57-171	57-171	57-171
57-171	57-171	57-171	57-171	57-184	57-184	57-184	57-184	57-184	57-184	57-184	57-184	57-184	57-184
57-184	57-184	57-184	57-184	57-202	57-202	57-202	57-202	57-202	57-202	57-202	57-202	57-202	57-202
57-202	57-202	57-206	57-206	57-206	57-206	57-206	57-206	57-206	57-206	57-206	57-206	57-206	57-206
57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211
57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-214	57-214
57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214
57-214	57-214	57-214	57-214	57-214	57-214	57-226	57-226	57-226	57-226	57-226	57-226	57-226	57-226
57-226	57-226	57-226	57-226	57-226	57-226	57-235	57-235	57-235	57-235	57-235	57-235	57-235	57-235
57-235	57-235	57-235	57-235	57-235	57-235	57-236	57-236	57-236	57-236	57-236	57-236	57-236	57-236
57-236	57-236	57-236	57-236	57-236	57-236	57-239	57-239	57-239	57-239	57-239	57-239	57-239	57-239
57-239	57-239	57-239	57-239	57-239	57-239	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247
57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-250	57-250
57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250
57-250	57-250	57-250	57-250	57-250	57-250	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251
57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-252	57-252
57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252
57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	58-16	58-16	58-16	58-16	58-16	58-16
58-16	58-16	58-16	58-16	58-16	58-16	58-57	58-57	58-57	58-57	58-57	58-57	58-57	58-57
58-62	58-62	58-62	58-62	58-62	58-62	58-62	58-62	58-198	58-198	58-198	58-198	58-198	58-198
58-198	58-198	58-198	58-198	58-198	58-198	58-198	58-198	58-199	58-199	58-199	58-199	58-199	58-199
58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199
58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200
58-200	58-200	58-200	58-200	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201
58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-202	58-202	58-202	58-202
58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202
58-202	58-202	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203
58-203	58-203	58-203	58-203	58-203	58-203	58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-204
58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-205	58-205	58-205	58-205

F12

	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-205	58-206	58-206
	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206	58-206
	58-208	58-208	59-12	59-12	60-27	60-27	60-37	60-37	60-39	60-39	60-39	60-39	60-55	60-55
	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-12	61-14	61-14
	61-14	61-14	61-46	61-46	61-46	61-46	61-46	61-46	61-46	61-46	61-51	61-51	61-51	61-51
	61-51	61-51	61-51	61-51	61-68	61-68	61-68	61-68	61-68	61-68	61-68	61-68	61-73	61-73
	61-73	61-73	61-73	61-73	61-73	61-73	61-86	61-86	61-86	61-86	61-86	61-86	61-86	61-86
	61-86	61-86	61-86	61-86	61-86	61-86	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87
	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-88	61-88
	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88
	61-88	61-88	61-88	61-88	61-90	61-90	62-67	62-67	62-152	62-152	62-154	62-154	62-154	62-154
	62-168	62-168	63-13	63-13	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38
	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38
	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-45	63-48	63-48
	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48
	63-48	63-48	63-48	63-48	63-49	63-49	63-49	63-49	63-73	63-73	63-73	63-73	63-73	63-73
	63-73	63-73	63-74	63-74	63-74	63-74	63-80	63-80	63-80	63-80	63-80	63-80	63-80	63-80
	63-84	63-84	63-84	63-84	63-84	63-84	63-84	63-84	63-91	63-91	63-91	63-91	63-91	63-91
	63-91	63-91	63-103	63-103	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106
	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106
	63-117	63-117	64-83	64-83	64-169	64-169	64-171	64-171	64-171	64-171	64-186	64-186	65-75	65-75
	65-169	65-169	65-171	65-171	65-171	65-171	65-185	65-185	66-4	66-4	66-5	66-5	66-5	66-5
	66-5	66-5	66-6	66-6	66-6	66-6	66-6	66-6	66-6	66-6	66-7	66-7	66-7	66-7
	66-7	66-7	66-7	66-7	66-8	66-8	66-8	66-8	66-8	66-8	66-8	66-8	66-8	66-8
	66-9	66-9	66-9	66-9	66-9	66-9	66-10	66-10	66-10	66-10	66-10	66-10	66-10	66-10
	66-10	66-10	66-12	66-12	66-23	66-23	66-25	66-25	66-25	66-25	66-25	66-25	66-26	66-26
	66-26	66-26	66-26	66-26	66-32	66-32	66-32	66-32	66-32	66-32	66-34	66-34	66-34	66-34
	66-34	66-34	66-35	66-35	66-36	66-36	66-36	66-36	66-36	66-36	66-36	66-36	66-36	66-36
	66-37	66-37	66-37	66-37	66-37	66-37	66-38	66-38	66-39	66-39	66-39	66-39	66-39	66-39
	66-39	66-39	66-39	66-39	66-40	66-40	66-40	66-40	66-40	66-40	66-41	66-41	66-42	66-42
	66-42	66-42	66-42	66-42	66-42	66-42	66-42	66-42	66-44	66-44	66-44	66-44	66-44	66-44
	66-44	66-44	66-44	66-44	66-46	66-46	66-46	66-46	66-46	66-46	66-46	66-46	66-46	66-46
	66-47	66-47	66-47	66-47	66-47	66-47	66-49	66-49	66-74	66-74	66-74	66-74	66-74	66-74
SVCSUB	37-10#	37-12#	60-27	60-27	60-27	62-67	62-67	62-67	63-13	63-13	63-13	64-83	64-83	64-83
	65-75	65-75	65-75											
SVCTAG	37-10#	37-15#	41-13	41-13	41-13	41-27	41-27	41-27	41-41	41-41	41-41	41-56	41-56	41-56
	41-71	41-71	41-71	41-186	41-186	41-186	41-200	41-200	41-200	41-222	41-222	41-222	41-236	41-236
	41-236	41-252	41-252	41-252	42-18	42-18	42-18	42-37	42-37	42-37	43-139	43-139	43-139	44-38
	44-38	44-38	45-19	45-19	45-19	45-22	45-22	45-22	46-49	46-49	46-49	46-61	46-61	46-61
	51-67	51-67	51-67	58-208	58-208	58-208	59-12	59-12	59-12	60-37	60-37	60-37	60-55	60-55
	60-55	61-90	61-90	61-90	62-152	62-152	62-152	62-168	62-168	62-168	63-103	63-103	63-103	63-117
	63-117	63-117	64-169	64-169	64-169	64-186	64-186	64-186	65-169	65-169	65-169	65-185	65-185	65-185
	66-12	66-12	66-12	66-49	66-49	66-49								
SVCTST	37-10#	37-11#	58-7	58-7	58-7	59-3	59-3	59-3	60-3	60-3	60-3	61-3	61-3	61-3
	62-3	62-3	62-3	63-3	63-3	63-3	64-3	64-3	64-3	65-3	65-3	65-3		
SWAPHD	52-62#	60-45	62-162	64-179	65-179									
T$$AUT	44-10#	44-38												
T$$CLE	45-4#	45-19												
T$$DU	45-20#	45-22												
T$$HAR	66-4	66-4#	66-12											
T$$HW	42-11	42-11#	42-18											
T$$INI	43-4#	43-139												
T$$MSG	41-1#	41-13	41-15#	41-27	41-29#	41-41	41-43#	41-56	41-58#	41-71	41-73#	41-186	41-188#	41-200
	41-202#	41-222	41-224#	41-236	41-238#	41-252								
T$$PRO	42-3#													
T$$SEG	51-16	51-16#	51-67	51-67#										

H12

I12

J12

Cross reference table (CREF V05.03)

K12

L12

Cross reference table (CREF V05.03)

M12

N12

	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-8	39-8#	39-9	39-9#	39-13	39-13#
	39-146	39-146#	39-564	39-564#	40-3	40-3#	41-1	41-1#	41-13	41-13#	41-15	41-15#	41-27	41-27#
	41-29	41-29#	41-41	41-41#	41-43	41-43#	41-56	41-56#	41-58	41-58#	41-71	41-71#	41-73	41-73#
	41-186	41-186#	41-188	41-188#	41-200	41-200#	41-202	41-202#	41-222	41-222#	41-224	41-224#	41-236	41-236#
	41-238	41-238#	41-252	41-252#	42-3	42-3#	42-10	42-10#	42-11	42-11#	42-11#	42-18	42-18#	42-21
	42-21#	42-22	42-22#	42-22#	42-37	42-37#	42-40	42-40#	42-45	42-45#	43-3	43-3#	43-4	43-4#
	43-139	43-139#	44-10	44-10#	44-38	44-38#	45-3	45-3#	45-4	45-4#	45-19	45-19#	45-20	45-20#
	45-22	45-22#	46-3	46-3#	46-43#	46-49	46-49#	46-50#	46-61	46-61#	51-67	51-67#	58-3	58-3#
	58-7	58-7#	58-208	58-208#	59-3	59-3#	59-12	59-12#	60-3	60-3#	60-27	60-27#	60-37	60-37#
	60-55	60-55#	61-3	61-3#	61-90	61-90#	62-3	62-3#	62-67	62-67#	62-152	62-152#	62-168	62-168#
	63-3	63-3#	63-13	63-13#	63-103	63-103#	63-117	63-117#	64-3	64-3#	64-83	64-83#	64-169	64-169#
	64-186	64-186#	65-3	65-3#	65-75	65-75#	65-169	65-169#	65-185	65-185#	66-3	66-3#	66-4	66-4#
	66-12	66-12#	66-22	66-22#	66-23	66-23#	66-49	66-49#	66-74	66-74#				
M$GETS	39-7	39-7#	39-142	39-142#	39-560	39-560#	39-791	39-791#	41-13	41-13#	41-27	41-27#	41-41	41-41#
	41-56	41-56#	41-71	41-71#	41-186	41-186#	41-200	41-200#	41-222	41-222#	41-236	41-236#	41-252	41-252#
	41-253	41-253#	42-7	42-7#	42-18	42-18#	42-19	42-19#	42-37	42-37#	42-38	42-38#	42-47	42-47#
	43-139	43-139#	43-140	43-140#	44-38	44-38#	45-19	45-19#	45-22	45-22#	45-23	45-23#	46-49	46-49#
	46-61	46-61#	51-67	51-67	51-67#	51-67#	57-266	57-266#	58-208	58-208#	59-12	59-12#	60-37	60-37#
	60-55	60-55#	61-90	61-90#	62-152	62-152#	62-168	62-168#	63-103	63-103#	63-117	63-117#	64-169	64-169#
	64-186	64-186#	65-169	65-169#	65-185	65-185#	65-186	65-186#	66-12	66-12#	66-20	66-20#	66-35	66-35#
	66-38	66-38#	66-41	66-41#	66-49	66-49#	66-73	66-73#						
M$GETT	60-39#	61-14#	62-154#	63-49#	63-74#	64-171#	65-171#	66-35	66-35#	66-38	66-38#	66-41	66-41#	
M$GNGB	39-5	39-5#	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-8	39-8#	39-9	39-9#	39-13	39-13#
	39-146	39-146#	39-564	39-564#	40-3	40-3#	41-1	41-1#	41-15	41-15#	41-29	41-29#	41-43	41-43#
	41-58	41-58#	41-73	41-73#	41-188	41-188#	41-202	41-202#	41-224	41-224#	41-238	41-238#	42-3	42-3#
	42-10	42-10#	42-11	42-11	42-11#	42-21	42-21#	42-22	42-22	42-22#	42-40	42-40#	42-45	42-45#
	43-3	43-3#	43-4	43-4#	44-10	44-10#	45-3	45-3#	45-4	45-4#	45-20	45-20#	46-3	46-3#
	46-43	46-43#	46-50	46-50#	58-3	58-3#	66-3	66-3#	66-4	66-4#	66-22	66-22#	66-23	66-23#
	66-74	66-74#												
M$GNIN	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6
	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#
	39-6#	39-6#	39-6#	39-6#	39-6#	39-6#	39-8	39-8	39-8#	39-8#	39-9	39-9	39-9#	39-9#
	41-13	41-13#	41-27	41-27#	41-41	41-41#	41-56	41-56#	41-71	41-71#	41-116	41-116	41-116	41-116
	41-116	41-116	41-116	41-116	41-116#	41-116#	41-116#	41-116#	41-116#	41-116#	41-116#	41-157	41-157	41-157
	41-157	41-157	41-157	41-157	41-157#	41-157#	41-157#	41-157#	41-157#	41-157#	41-171	41-171	41-171	41-171
	41-171	41-171	41-171	41-171	41-171	41-171	41-171	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#	41-171#
	41-171#	41-171#	41-171#	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180	41-180#	41-180#
	41-180#	41-180#	41-180#	41-180#	41-180#	41-180#	41-186	41-186#	41-200	41-200#	41-222	41-222#	41-236	41-236#
	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244	41-244#	41-244#	41-244#	41-244#
	41-244#	41-244#	41-244#	41-244#	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245	41-245
	41-245	41-245	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-245#	41-248
	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248	41-248#	41-248#	41-248#	41-248#
	41-248#	41-248#	41-248#	41-248#	41-248#	41-248#	41-252	41-252#	42-11	42-11#	42-22	42-22#	42-45	42-45

42-45 42-45 42-45 42-45 42-45 42-45 42-45 42-45# 42-45# 42-45# 42-45# 42-45# 42-45# 42-45#
42-45# 42-45# 43-9 43-9 43-9 43-9# 43-9# 43-9# 43-10 43-10# 43-12 43-12 43-12# 43-12#
43-13 43-13# 43-14 43-14# 43-15 43-15# 43-19 43-19 43-19# 43-19# 43-20 43-20# 43-24 43-24
43-24# 43-24# 43-25 43-25# 43-46 43-46 43-46# 43-46# 43-47 43-47# 43-49 43-49 43-49# 43-49#
43-50 43-50# 43-53 43-53 43-53# 43-53# 43-54 43-54# 43-70 43-70 43-70# 43-70# 43-70# 43-71
43-71# 43-104 43-104 43-104 43-104 43-104 43-104 43-104# 43-104# 43-104# 43-104# 43-104# 43-104# 43-105
43-105 43-105# 43-105# 43-132 43-132 43-132 43-132 43-132 43-132 43-132# 43-132# 43-132# 43-132# 43-132#
43-133 43-133 43-133 43-133 43-133 43-133 43-133 43-133 43-133 43-133 43-133# 43-133# 43-133# 43-133#
43-133# 43-133# 43-133# 43-133# 43-134 43-134 43-134 43-134 43-134 43-134# 43-134# 43-134# 43-134# 43-135
43-135 43-135# 43-135# 43-136 43-136# 43-139 43-139# 44-12 44-12 44-12 44-12 44-12 44-12# 44-12#
44-12# 44-12# 44-12# 44-12# 44-12# 44-18 44-18 44-18 44-18 44-18 44-18 44-18# 44-18# 44-18#
44-18# 44-18# 44-19 44-19 44-19 44-19 44-19 44-19 44-19 44-19 44-19 44-19# 44-19# 44-19#
44-19# 44-19# 44-19# 44-19# 44-19# 44-21 44-21 44-21 44-21 44-21 44-21# 44-21# 44-21#
44-21# 44-23 44-23 44-23# 44-23# 44-31 44-31 44-31 44-31 44-31 44-31# 44-31# 44-31#
44-31# 44-31# 44-33 44-33 44-33 44-33 44-33 44-33 44-33 44-33 44-33 44-33# 44-33# 44-33#
44-33# 44-33# 44-33# 44-33# 44-33# 44-35 44-35 44-35 44-35 44-35 44-35 44-35# 44-35# 44-35#
44-35# 44-36 44-36 44-36# 44-36# 44-37 44-37 44-37# 44-37# 44-38 44-38# 45-5 45-5 45-5
45-5 45-5 45-5 45-5# 45-5# 45-5# 45-5# 45-5# 45-5# 45-6 45-6 45-6# 45-6# 45-13
45-13 45-13# 45-13# 45-17 45-17 45-17# 45-17# 45-18 45-18# 45-19 45-19# 45-22 45-22# 46-9
46-9# 46-10 46-10# 46-11 46-11 46-11 46-11 46-11 46-11 46-11 46-11 46-11# 46-17 46-17
46-17 46-17 46-17 46-17 46-17 46-17 46-17# 46-29 46-29# 46-30 46-30# 46-32 46-32 46-32
46-32 46-32 46-32 46-32 46-32 46-32# 46-37 46-37 46-37 46-37 46-37 46-37 46-37 46-37
46-37# 46-49 46-49# 46-61 46-61# 47-6 47-6# 47-7 47-7# 47-8 47-8 47-8 47-8 47-8
47-8 47-8 47-8# 47-8# 47-8# 47-8# 47-8# 47-8# 47-9 47-9 47-9 47-9 47-9 47-9
47-9 47-9 47-9 47-9 47-9# 47-9# 47-9# 47-9# 47-9# 47-9# 47-9# 47-9# 47-9# 47-10
47-10 47-10 47-10 47-10# 47-10# 47-10# 47-10# 47-11 47-11 47-11# 47-11# 47-12 47-12# 47-47
49-47 49-47 49-47 49-47# 49-47# 49-47# 49-47# 49-47# 49-86 49-86 49-86 49-86 49-86# 49-86#
49-86# 49-86# 49-86# 49-101 49-101 49-101 49-101 49-101# 49-101# 49-101# 49-101# 49-101# 49-109 49-109
49-109 49-109 49-109# 49-109# 49-109# 49-109# 49-109# 50-87 50-87 50-87 50-87 50-87# 50-87# 50-87#
50-87# 50-87# 50-93 50-93 50-93 50-93 50-93# 50-93# 50-93# 50-93# 50-93# 51-16 51-16# 51-18
51-18# 51-19 51-19# 51-42 51-42 51-42 51-42 51-42# 51-42# 51-42# 51-42# 51-42# 51-60 51-60
51-60 51-60 51-60# 51-60# 51-60# 51-60# 51-60# 51-67 51-67# 52-24 52-24 52-24 52-24 52-24#
52-24# 52-24# 52-24# 52-24# 52-35 52-35 52-35 52-35# 52-35# 52-35# 52-35# 52-35# 52-35# 53-19
53-19 53-19 53-19 53-19 53-19 53-19# 53-19# 53-19# 53-19# 53-83 53-83 53-83 53-83
53-83# 53-83# 53-83# 53-83# 53-83# 53-86 53-86# 53-87 53-87# 53-97 53-97 53-97 53-97 53-97#
53-97# 53-97# 53-97# 53-97# 53-112 53-112 53-112 53-112 53-112 53-112# 53-112# 53-112# 53-112# 53-113
53-113 53-113 53-113 53-113 53-113 53-113 53-113 53-113 53-113 53-113# 53-113# 53-113# 53-113# 53-113#
53-113# 53-113# 53-113# 53-114 53-114 53-114 53-114 53-114 53-114# 53-114# 53-114# 53-114# 54-43 54-43
54-43 54-43 54-43# 54-43# 54-43# 54-43# 54-43# 54-55 54-55 54-55 54-55 54-55# 54-55# 54-55#
54-55# 54-55# 54-60 54-60 54-60 54-60 54-60 54-60# 54-60# 54-60# 54-60# 54-60# 54-70 54-70
54-70 54-70# 54-70# 54-70# 54-70# 54-70# 55-29 55-29 55-29 55-29# 55-29# 55-29# 55-29#
55-29# 55-42 55-42 55-42 55-42 55-42# 55-42# 55-42# 55-42# 55-49 55-49 55-49 55-49
55-49# 55-49# 55-49# 55-49# 55-49# 55-114 55-114 55-114 55-114 55-114# 55-114# 55-114# 55-114#
55-124 55-124 55-124 55-124 55-124# 55-124# 55-124# 55-124# 55-124# 55-129 55-129 55-129 55-129 55-129#
55-129# 55-129# 55-129# 55-129# 55-191 55-191 55-191 55-191 55-191# 55-191# 55-191# 55-191# 55-191# 55-244
55-244 55-244 55-244 55-244# 55-244# 55-244# 55-244# 55-244# 55-244# 55-250 55-250 55-250 55-250# 55-250#
55-250# 55-250# 55-250# 56-28 56-28 56-28 56-28 56-28 56-28 56-28 56-28# 56-28# 56-28# 56-28#
56-28# 56-28# 56-28# 56-28# 56-28# 56-28# 56-43 56-43 56-43 56-43 56-43# 56-43# 56-43#
56-43# 57-59 57-59 57-59 57-59 57-59# 57-59# 57-59# 57-59# 57-59# 57-75 57-75 57-75 57-75
57-75# 57-75# 57-75# 57-75# 57-75# 57-82 57-82 57-82 57-82 57-82# 57-82# 57-82# 57-82# 57-82#
57-91 57-91 57-91 57-91 57-91# 57-91# 57-91# 57-91# 57-91# 57-101 57-101 57-101 57-101 57-101#
57-101# 57-101# 57-101# 57-101# 57-166 57-166 57-166 57-166 57-166 57-166 57-166# 57-166# 57-166# 57-166# 57-166# 57-166# 57-166#
57-166# 57-167 57-167 57-167 57-167 57-167 57-167 57-167 57-167# 57-167# 57-167# 57-167# 57-167# 57-171 57-171
57-171 57-171 57-171 57-171 57-171 57-171# 57-171# 57-171# 57-171# 57-171# 57-171# 57-184 57-184 57-184
57-184 57-184 57-184 57-184 57-184# 57-184# 57-184# 57-184# 57-184# 57-184# 57-202 57-202 57-202 57-202
57-202 57-202 57-202# 57-202# 57-202# 57-202# 57-202# 57-206 57-206 57-206 57-206 57-206 57-206 57-206#

C13

57-206#	57-206#	57-206#	57-206#	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211	57-211
57-211	57-211	57-211	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#
57-211#	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214	57-214#	57-214#	57-214#
57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-226	57-226	57-226	57-226	57-226	57-226
57-226	57-226#	57-226#	57-226#	57-226#	57-226#	57-226#	57-226#	57-235	57-235	57-235	57-235	57-235	57-235	57-235
57-235#	57-235#	57-235#	57-235#	57-235#	57-235#	57-236	57-236	57-236	57-236	57-236	57-236	57-236	57-236	57-236#
57-236#	57-236#	57-236#	57-236#	57-236#	57-239	57-239	57-239	57-239	57-239	57-239	57-239	57-239#	57-239#	57-239#
57-239#	57-239#	57-239#	57-239#	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247	57-247
57-247#	57-247#	57-247#	57-247#	57-247#	57-247#	57-247#	57-247#	57-250	57-250	57-250	57-250	57-250	57-250	57-250
57-250	57-250	57-250	57-250	57-250	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#
57-250#	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251#	57-251#	57-251#
57-251#	57-251#	57-251#	57-251#	57-251#	57-251#	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252
57-252	57-252	57-252	57-252	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#
57-252#	58-16	58-16	58-16	58-16	58-16	58-16	58-16#	58-16#	58-16#	58-16#	58-16#	58-16#	58-57	58-57
58-57	58-57	58-57#	58-57#	58-57#	58-57#	58-57#	58-62	58-62	58-62	58-62	58-62	58-62#	58-62#	58-62#
58-62#	58-62#	58-198	58-198	58-198	58-198	58-198	58-198	58-198	58-198#	58-198#	58-198#	58-198#	58-198#	58-198#
58-198#	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199	58-199#	58-199#	58-199#
58-199#	58-199#	58-199#	58-199#	58-199#	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200	58-200
58-200#	58-200#	58-200#	58-200#	58-200#	58-200#	58-200#	58-200#	58-201	58-201	58-201	58-201	58-201	58-201	58-201
58-201	58-201	58-201	58-201	58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-201#	58-202
58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202#	58-202#	58-202#	58-202#	58-202#	58-202#
58-202#	58-202#	58-202#	58-202#	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203	58-203#
58-203#	58-203#	58-203#	58-203#	58-203#	58-203#	58-203#	58-204	58-204	58-204	58-204	58-204	58-204	58-204	58-204
58-204	58-204	58-204#	58-204#	58-204#	58-204#	58-204#	58-204#	58-204#	58-205	58-205	58-205	58-205	58-205	58-205
58-205	58-205	58-205	58-205	58-205#	58-205#	58-205#	58-205#	58-205#	58-205#	58-206	58-206	58-206	58-206	58-206
58-206	58-206	58-206	58-206	58-206	58-206#	58-206#	58-206#	58-206#	58-206#	58-206#	58-208	58-208#	58-208#	58-208#
59-12	59-12#	60-27	60-27#	60-37	60-37#	60-39	60-39	60-39#	60-39#	60-55	60-55#	61-12	61-12	61-12
61-12	61-12	61-12	61-12	61-12#	61-12#	61-12#	61-12#	61-12#	61-14	61-14	61-14#	61-14#	61-46	61-46
61-46	61-46	61-46	61-46#	61-46#	61-46#	61-46#	61-46#	61-51	61-51	61-51	61-51	61-51#	61-51#	61-51#
61-51#	61-51#	61-51#	61-68	61-68	61-68	61-68	61-68#	61-68#	61-68#	61-68#	61-68#	61-73	61-73	61-73
61-73	61-73	61-73#	61-73#	61-73#	61-73#	61-73#	61-86	61-86	61-86	61-86	61-86	61-86	61-86	61-86
61-86#	61-86#	61-86#	61-86#	61-86#	61-86#	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87	61-87
61-87	61-87	61-87#	61-87#	61-87#	61-87#	61-87#	61-87#	61-87#	61-87#	61-88	61-88	61-88	61-88	61-88
61-88	61-88	61-88	61-88	61-88	61-88	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#	61-88#
61-88#	61-90	61-90#	62-67	62-67#	62-152	62-152#	62-154	62-154	62-154#	62-154#	62-168	62-168#	63-13	63-13#
63-13#	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38#	63-38#
63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-38#	63-45	63-45	63-45	63-45	63-45	63-45
63-45	63-45#	63-45#	63-45#	63-45#	63-45#	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48	63-48
63-48	63-48	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-49	63-49	63-49#	63-49#	63-49#
63-73	63-73	63-73	63-73	63-73#	63-73#	63-73#	63-73#	63-73#	63-74	63-74	63-74#	63-74#	63-80	63-80
63-80	63-80	63-80	63-80#	63-80#	63-80#	63-80#	63-80#	63-84	63-84	63-84	63-84	63-84#	63-84#	63-84#
63-84#	63-84#	63-84#	63-91	63-91	63-91	63-91	63-91#	63-91#	63-91#	63-91#	63-91#	63-103	63-103#	63-103#
63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106	63-106#	63-106#	63-106#
63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-117	63-117#	64-83	64-83#	64-169	64-169#	64-169#
64-171	64-171	64-171#	64-171#	64-186	64-186#	65-75	65-75#	65-169	65-169#	65-171	65-171	65-171#	65-171#	65-171#
65-185	65-185#	66-4	66-4#	66-5	66-5	66-5	56-5#	66-6	66-6	66-6	66-6	66-6#	66-7	66-7
66-7	66-7	66-7	66-7#	66-8	66-8	66-8	66-8	66-8	66-8#	66-9	66-9	66-9	66-9#	66-9#
66-10	66-10	66-10	66-10	66-10	66-10#	66-12	66-12#	66-23	66-23#	66-25	66-25	66-25	66-25#	66-25#
66-26	66-26	66-26	66-26#	66-32	66-32	66-32	66-32#	66-34	66-34	66-34	66-34#	66-35	66-35#	66-35#
66-36	66-36	66-36	66-36	66-36	66-36#	66-37	66-37	66-37	66-37#	66-38	66-38#	66-39	66-39	66-39
66-39	66-39	66-39	66-39#	66-40	66-40	66-40	66-40#	66-41	66-41#	66-42	66-42	66-42	66-42	66-42
66-42	66-42#	66-44	66-44	66-44	66-44	66-44	66-44#	66-46	66-46	66-46	66-46	66-46	66-46#	66-46#
66-47	66-47	66-47	66-47#	66-49	66-49#	66-74	66-74	66-74	66-74#	66-74#	66-74#			
M$GNLS 51-67	51-67#													
M$GNSU 60-27	60-27#	62-67	62-67#	63-13	63-13#	64-83	64-83#	65-75	65-75#					
M$GNTA 41-13	41-13#	41-27	41-27#	41-41	41-41#	41-56	41-56#	41-71	41-71#	41-186	41-186#	41-200	41-200#	

D13

E13

M$PUSH
39-5 39-5# 39-13 39-13# 39-146 39-146# 39-564 39-564# 40-3 40-3# 41-1 41-1# 41-15 41-15#
41-29 41-29# 41-43 41-43# 41-58 41-58# 41-73 41-73# 41-188 41-188# 41-202 41-202# 41-224 41-224#
41-238 41-238# 42-3 42-3# 42-10 42-10# 42-11 42-11# 42-21 42-21# 42-22 42-22# 42-40 42-40#
43-3 43-3# 43-4 43-4# 44-10 44-10# 45-3 45-3# 45-4 45-4# 45-20 45-20# 46-3 46-3#
46-43 46-43# 46-50 46-50# 51-16 51-16 51-16# 58-3 58-3# 58-7 58-7# 59-3 59-3# 60-3
60-3# 60-27 60-27# 61-3 61-3# 62-3 62-3# 62-67 62-67# 63-3 63-3# 63-13 63-13# 64-3
64-3# 64-83 64-83# 65-3 65-3# 65-75 65-75# 66-3 66-3# 66-4 66-4# 66-22 66-22# 66-23
66-23#

M$PUT
41-116 41-116 41-116 41-116 41-116 41-116# 41-157 41-157 41-157 41-157 41-157# 41-171 41-171 41-171
41-171 41-171 41-171 41-171 41-171 41-171# 41-180 41-180 41-180 41-180 41-180 41-180# 41-180# 41-244
41-244 41-244 41-244 41-244 41-244 41-244# 41-245 41-245 41-245 41-245 41-245 41-245 41-245 41-245
41-245 41-245# 41-248 41-248 41-248 41-248 41-248 41-248 41-248 41-248 41-248# 43-104 43-104 43-104
43-104 43-104# 43-132 43-132 43-132 43-132# 43-133 43-133 43-133 43-133 43-133 43-133# 43-133# 43-134
43-134 43-134# 44-12 44-12 44-12 44-12 44-12# 44-18 44-18 44-18 44-18# 44-19 44-19 44-19
44-19 44-19 44-19 44-19# 44-21 44-21 44-21# 44-31 44-31 44-31 44-31# 44-33 44-33 44-33
44-33 44-33 44-33 44-33# 44-35 44-35 44-35# 45-5 45-5 45-5 45-5# 47-8 47-8 47-8
47-8 47-8 47-8# 47-9 47-9 47-9 47-9 47-9 47-9# 47-10 47-10 47-10# 53-19 53-19
53-19 53-19 53-19# 53-112 53-112 53-112# 53-113 53-113 53-113 53-113 53-113 53-113# 53-113# 53-114
53-114 53-114# 56-28 56-28 56-28 56-28 56-28 56-28# 57-166 57-166 57-166 57-166# 57-166# 57-167
57-167 57-167 57-167# 57-171 57-171 57-171 57-171 57-171# 57-184 57-184 57-184 57-184# 57-184# 57-202
57-202 57-202 57-202# 57-206 57-206 57-206 57-206# 57-211 57-211 57-211 57-211 57-211 57-211# 57-211
57-211 57-211 57-211 57-211# 57-214 57-214 57-214 57-214 57-214 57-214 57-214 57-214# 57-214# 57-226
57-226 57-226 57-226 57-226# 57-235 57-235 57-235 57-235 57-235# 57-236 57-236 57-236 57-236 57-236#
57-239 57-239 57-239 57-239 57-239# 57-247 57-247 57-247 57-247 57-247 57-247 57-247# 57-250 57-250
57-250 57-250 57-250 57-250 57-250 57-250 57-250# 57-251 57-251 57-251 57-251 57-251 57-251 57-251
57-251# 57-252 57-252 57-252 57-252 57-252 57-252 57-252 57-252 57-252 57-252# 58-16 58-16 58-16
58-16# 58-198 58-198 58-198 58-198 58-198# 58-199 58-199 58-199 58-199 58-199 58-199 58-199# 58-200
58-200 58-200 58-200 58-200 58-200 58-200# 58-201 58-201 58-201 58-201 58-201 58-201 58-201 58-201#
58-202 58-202 58-202 58-202 58-202 58-202 58-202 58-202# 58-203 58-203 58-203 58-203 58-203 58-203
58-203# 58-204 58-204 58-204 58-204 58-204 58-204 58-204# 58-205 58-205 58-205 58-205 58-205 58-205#
58-206 58-206 58-206 58-206 58-206 58-206# 61-12 61-12 61-12 61-12# 61-86 61-86 61-86 61-86
61-86# 61-87 61-87 61-87 61-87 61-87 61-87 61-87# 61-88 61-88 61-88 61-88 61-88 61-88
61-88 61-88# 63-38 63-38 63-38 63-38 63-38 63-38 63-38 63-38 63-38# 63-45 63-45 63-45
63-45# 63-48 63-48 63-48 63-48 63-48 63-48# 63-106 63-106 63-106 63-106 63-106 63-106
63-106 63-106 63-106#

M$PUT1
41-116 41-116 41-116 41-116 41-116 41-116# 41-116# 41-116# 41-116# 41-116# 41-157 41-157 41-157 41-157
41-157# 41-157# 41-157# 41-157# 41-171 41-171 41-171 41-171 41-171 41-171 41-171 41-171 41-171# 41-171#
41-171# 41-171# 41-171# 41-171# 41-171# 41-171# 41-180 41-180 41-180 41-180 41-180 41-180 41-180# 41-180#
41-180# 41-180# 41-180# 41-180# 41-244 41-244 41-244 41-244 41-244 41-244 41-244# 41-244# 41-244# 41-244#
41-244# 41-244# 41-245 41-245 41-245 41-245 41-245 41-245 41-245 41-245 41-245 41-245# 41-245# 41-245#
41-245# 41-245# 41-245# 41-245# 41-245# 41-245# 41-248 41-248 41-248 41-248 41-248 41-248 41-248 41-248
41-248# 41-248# 41-248# 41-248# 41-248# 41-248# 41-248# 41-248# 43-104 43-104 43-104 43-104 43-104# 43-104#
43-104# 43-104# 43-132 43-132 43-132 43-132# 43-132# 43-132# 43-133 43-133 43-133 43-133 43-133 43-133
43-133# 43-133# 43-133# 43-133# 43-133# 43-133# 43-134 43-134 43-134# 43-134# 44-12 44-12 44-12 44-12
44-12# 44-12# 44-12# 44-12# 44-18 44-18 44-18 44-18# 44-18# 44-18# 44-19 44-19 44-19 44-19
44-19 44-19 44-19# 44-19# 44-19# 44-19# 44-19# 44-19# 44-21 44-21 44-21# 44-21# 44-31 44-31
44-31 44-31# 44-31# 44-31# 44-33 44-33 44-33 44-33 44-33 44-33 44-33# 44-33# 44-33# 44-33#
44-33# 44-33# 44-35 44-35 44-35# 44-35# 45-5 45-5 45-5 45-5 45-5# 45-5# 45-5# 45-5#
47-8 47-8 47-8 47-8 47-8# 47-8# 47-8# 47-8# 47-9 47-9 47-9 47-9 47-9 47-9
47-9# 47-9# 47-9# 47-9# 47-9# 47-9# 47-10 47-10 47-10# 47-10# 53-19 53-19 53-19 53-19#
53-19# 53-19# 53-112 53-112 53-112# 53-112# 53-113 53-113 53-113 53-113 53-113 53-113 53-113# 53-113#
53-113# 53-113# 53-113# 53-113# 53-114 53-114 53-114# 53-114# 56-28 56-28 56-28 56-28 56-28 56-28
56-28# 56-28# 56-28# 56-28# 56-28# 56-28# 57-166 57-166 57-166 57-166# 57-166# 57-166# 57-167 57-167
57-167 57-167# 57-167# 57-167# 57-171 57-171 57-171 57-171 57-171# 57-171# 57-171# 57-171# 57-184 57-184
57-184 57-184 57-184# 57-184# 57-184# 57-184# 57-202 57-202 57-202 57-202# 57-202# 57-202# 57-206 57-206
57-206 57-206# 57-206# 57-206# 57-211 57-211 57-211 57-211 57-211 57-211 57-211 57-211 57-211 57-211

F13

	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-211#	57-214	57-214	57-214	57-214
	57-214	57-214	57-214	57-214	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-214#	57-226	57-226
	57-226	57-226	57-226#	57-226#	57-226#	57-226#	57-235	57-235	57-235	57-235	57-235#	57-235#	57-235#	57-235#
	57-236	57-236	57-236	57-236	57-236#	57-236#	57-236#	57-236#	57-239	57-239	57-239	57-239	57-239#	57-239#
	57-239#	57-239#	57-247	57-247	57-247	57-247	57-247	57-247	57-247#	57-247#	57-247#	57-247#	57-247#	57-247#
	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250	57-250#	57-250#	57-250#	57-250#	57-250#	57-250#
	57-250#	57-250#	57-251	57-251	57-251	57-251	57-251	57-251	57-251	57-251#	57-251#	57-251#	57-251#	57-251#
	57-251#	57-251#	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252	57-252#	57-252#	57-252#
	57-252#	57-252#	57-252#	57-252#	57-252#	57-252#	58-16	58-16	58-16	58-16#	58-16#	58-16#	58-198	58-198
	58-198	58-198	58-198#	58-198#	58-198#	58-198#	58-199	58-199	58-199	58-199	58-199	58-199	58-199#	58-199#
	58-199#	58-199#	58-199#	58-199#	58-200	58-200	58-200	58-200	58-200	58-200	58-200#	58-200#	58-200#	58-200#
	58-200#	58-200#	58-201	58-201	58-201	58-201	58-201	58-201	58-201	58-201#	58-201#	58-201#	58-201#	58-201#
	58-201#	58-201#	58-202	58-202	58-202	58-202	58-202	58-202	58-202	58-202#	58-202#	58-202#	58-202#	58-202#
	58-202#	58-202#	58-203	58-203	58-203	58-203	58-203	58-203	58-203#	58-203#	58-203#	58-203#	58-203#	58-203#
	58-204	58-204	58-204	58-204	58-204	58-204	58-204#	58-204#	58-204#	58-204#	58-204#	58-204#	58-205	58-205
	58-205	58-205	58-205	58-205#	58-205#	58-205#	58-205#	58-205#	58-206	58-206	58-206	58-206	58-206	58-206#
	58-206#	58-206#	58-206#	58-206#	61-12	61-12	61-12	61-12#	61-12#	61-12#	61-86	61-86	61-86	61-86
	61-86#	61-86#	61-86#	61-86#	61-87	61-87	61-87	61-87	61-87	61-87	61-87#	61-87#	61-87#	61-87#
	61-87#	61-87#	61-88	61-88	61-88	61-88	61-88	61-88	61-88	61-88#	61-88#	61-88#	61-88#	61-88#
	61-88#	61-88#	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38	63-38#	63-38#	63-38#	63-38#
	63-38#	63-38#	63-38#	63-38#	63-45	63-45	63-45	63-45#	63-45#	63-45#	63-48	63-48	63-48	63-48
	63-48	63-48	63-48#	63-48#	63-48#	63-48#	63-48#	63-48#	63-106	63-106	63-106	63-106	63-106	63-106
	63-106	63-106	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#	63-106#				
M$RADI	66-5	66-5#	66-6	66-6#	66-7	66-7#	66-8	66-8#	66-9	66-9#	66-10	66-10#	66-25	66-25#
	66-26	66-26#	66-32	66-32#	66-34	66-34#	66-36	66-36#	66-37	66-37#	66-39	66-39#	66-40	66-40#
	66-42	66-42#	66-44	66-44#	66-46	66-46#	66-47	66-47#						
M$RNRO	43-9	43-9#	43-70	43-70#										
M$SETS	39-5	39-5#	39-13	39-13#	39-146	39-146#	39-564	39-564#	40-3	40-3#	41-1	41-1#	41-15	41-15#
	41-29	41-29#	41-43	41-43#	41-58	41-58#	41-73	41-73#	41-188	41-188#	41-202	41-202#	41-224	41-224#
	41-238	41-238#	42-3	42-3#	42-10	42-10#	42-11	42-11#	42-21	42-21#	42-22	42-22#	42-40	42-40#
	43-3	43-3#	43-4	43-4#	44-10	44-10#	45-3	45-3#	45-4	45-4#	45-20	45-20#	46-3	46-3#
	46-43	46-43#	46-50	46-50#	51-16	51-16	51-16#	51-16#	58-3	58-3#	58-7	58-7#	59-3	59-3#
	60-3	60-3#	60-27	60-27#	61-3	61-3#	62-3	62-3#	62-67	62-67#	63-3	63-3#	63-13	63-13#
	64-3	64-3#	64-83	64-83#	65-3	65-3#	65-75	65-75#	66-3	66-3#	66-4	66-4#	66-22	66-22#
	66-23	66-23#												
M$SVC	41-13	41-13#	41-27	41-27#	41-41	41-41#	41-56	41-56#	41-71	41-71#	41-116	41-116#	41-157	41-157#
	41-171	41-171#	41-180	41-180#	41-186	41-186#	41-200	41-200#	41-222	41-222#	41-236	41-236#	41-244	41-244#
	41-245	41-245#	41-248	41-248#	41-252	41-252#	43-9	43-9#	43-12	43-12#	43-13	43-13#	43-14	43-14#
	43-19	43-19#	43-24	43-24#	43-46	43-46#	43-49	43-49#	43-53	43-53#	43-70	43-70#	43-104	43-104#
	43-105	43-105#	43-132	43-132#	43-133	43-133#	43-134	43-134#	43-135	43-135#	43-136	43-136#	43-139	43-139#
	44-12	44-12#	44-18	44-18#	44-19	44-19#	44-21	44-21#	44-23	44-23#	44-31	44-31#	44-33	44-33#
	44-35	44-35#	44-36	44-36#	44-37	44-37#	44-38	44-38#	45-5	45-5#	45-6	45-6#	45-13	45-13#
	45-17	45-17#	45-18	45-18#	45-19	45-19#	45-22	45-22#	46-9	46-9#	46-29	46-29#	47-6	47-6#
	47-8	47-8#	47-9	47-9#	47-10	47-10#	47-11	47-11#	47-12	47-12#	49-47	49-86	49-101	49-109
	50-87	50-93	51-16	51-16#	51-18	51-18#	51-42	51-60	51-67	51-67#	52-24	52-35	53-19	53-19#
	53-83	53-86	53-86#	53-97	53-112	53-112#	53-113	53-113#	53-114	53-114#	54-43	54-55	54-60	54-70
	55-29	55-42	55-49	55-114	55-124	55-129	55-191	55-244	55-250	56-28	56-28#	56-43	57-59	57-75
	57-82	57-91	57-101	57-166	57-166#	57-167	57-167#	57-171	57-171#	57-184	57-184#	57-202	57-202#	57-206
	57-206#	57-211	57-211#	57-214	57-214#	57-226	57-226#	57-235	57-235#	57-236	57-236#	57-239	57-239#	57-247
	57-247#	57-250	57-250#	57-251	57-251#	57-252	57-252#	58-16	58-16#	58-57	58-62	58-198	58-198#	58-199
	58-199#	58-200	58-200#	58-201	58-201#	58-202	58-202#	58-203	58-203#	58-204	58-204#	58-205	58-205#	58-206
	58-206#	58-208	58-208#	59-12	59-12#	60-27	60-27#	60-37	60-37#	60-39	60-39#	60-55	60-55#	61-12
	61-12#	61-14	61-14#	61-46	61-51	61-68	61-73	61-86	61-86#	61-87	61-87#	61-88	61-88#	61-90
	61-90#	62-67	62-67#	62-152	62-152#	62-154	62-154#	62-168	62-168#	63-13	63-13#	63-38	63-38#	63-45
	63-45#	63-48	63-48#	63-49	63-49#	63-73	63-74	63-74#	63-80	63-84	63-91	63-103	63-103#	63-106
	63-106#	63-117	63-117#	64-83	64-83#	64-169	64-169#	64-171	64-171#	64-186	64-186#	65-75	65-75#	65-169

G13

	65-169#	65-171	65-171#	65-185	65-185#										
M$TLAB	41-13#	41-27#	41-41#	41-56#	41-71#	41-116#	41-157#	41-171#	41-180#	41-186#	41-200#	41-222#	41-236#	41-244#	
	41-245#	41-248#	41-252#	43-9#	43-12#	43-13#	43-14#	43-19#	43-24#	43-46#	43-49#	43-53#	43-70#	43-104#	
	43-105#	43-132#	43-133#	43-134#	43-135#	43-136#	43-139#	44-12#	44-18#	44-19#	44-21#	44-23#	44-31#	44-33#	
	44-35#	44-36#	44-37#	44-38#	45-5#	45-6#	45-13#	45-17#	45-18#	45-19#	45-22#	46-9#	46-29#	47-6#	
	47-8#	47-9#	47-10#	47-11#	47-12#	49-47#	49-86#	49-101#	49-109#	50-87#	50-93#	51-16#	51-18#	51-42#	
	51-60#	51-67#	52-24#	52-35#	53-19#	53-83#	53-86#	53-97#	53-112#	53-113#	53-114#	54-43#	54-55#	54-60#	
	54-70#	55-29#	55-42#	55-49#	55-114#	55-124#	55-129#	55-191#	55-244#	55-250#	56-28#	56-43#	57-59#	57-75#	
	57-82#	57-91#	57-101#	57-166#	57-167#	57-171#	57-184#	57-202#	57-206#	57-211#	57-214#	57-226#	57-235#	57-236#	
	57-239#	57-247#	57-250#	57-251#	57-252#	58-16#	58-57#	58-62#	58-198#	58-199#	58-200#	58-201#	58-202#	58-203#	
	58-204#	58-205#	58-206#	58-208#	59-12#	60-27#	60-37#	60-39#	60-55#	61-12#	61-14#	61-46#	61-51#	61-68#	
	61-73#	61-86#	61-87#	61-88#	61-90#	62-67#	62-152#	62-154#	62-168#	63-13#	63-38#	63-45#	63-48#	63-49#	
	63-73#	63-74#	63-80#	63-84#	63-91#	63-103#	63-106#	63-117#	64-83#	64-169#	64-171#	64-186#	65-75#	65-169#	
	65-171#	65-185#													
M$TSTL	41-13	41-13#	41-27	41-27#	41-41	41-41#	41-56	41-56#	41-71	41-71#	41-116	41-116#	41-157	41-157#	
	41-171	41-171#	41-180	41-180#	41-186	41-186#	41-200	41-200#	41-222	41-222#	41-236	41-236#	41-244	41-244#	
	41-245	41-245#	41-248	41-248#	41-252	41-252#	43-9	43-9#	43-12	43-12#	43-13	43-13#	43-14	43-14#	
	43-19	43-19#	43-24	43-24#	43-46	43-46#	43-49	43-49#	43-53	43-53#	43-70	43-70#	43-104	43-104#	
	43-105	43-105#	43-132	43-132#	43-133	43-133#	43-134	43-134#	43-135	43-135#	43-136	43-136#	43-139	43-139#	
	44-12	44-12#	44-18	44-18#	44-19	44-19#	44-21	44-21#	44-23	44-23#	44-31	44-31#	44-33	44-33#	
	44-35	44-35#	44-36	44-36#	44-37	44-37#	44-38	44-38#	45-5	45-5#	45-6	45-6#	45-13	45-13#	
	45-17	45-17#	45-18	45-18#	45-19	45-19#	45-22	45-22#	46-9	46-9#	46-29	46-29#	47-6	47-6#	
	47-8	47-8#	47-9	47-9#	47-10	47-10#	47-11	47-11#	47-12	47-12#	49-47	49-47#	49-47#	49-86	
	49-86#	49-86#	49-101	49-101#	49-101#	49-109	49-109#	49-109#	50-87	50-87#	50-87#	50-93	50-93#	50-93#	
	51-16	51-16#	51-18	51-18#	51-42	51-42#	51-42#	51-60	51-60#	51-60#	51-67	51-67#	52-24	52-24#	
	52-24#	52-35	52-35#	52-35#	53-19	53-19#	53-83	53-83#	53-83#	53-86	53-86#	53-97	53-97#	53-97#	
	53-112	53-112#	53-113	53-113#	53-114	53-114#	54-43	54-43#	54-43#	54-55	54-55#	54-55#	54-60	54-60#	
	54-60#	54-70	54-70#	54-70#	55-29	55-29#	55-29#	55-42	55-42#	55-42#	55-49	55-49#	55-49#	55-114	
	55-114#	55-114#	55-124	55-124#	55-124#	55-129	55-129#	55-129#	55-191	55-191#	55-191#	55-244	55-244#	55-244#	
	55-250	55-250#	55-250#	56-28	56-28#	56-43	56-43#	56-43#	57-59	57-59#	57-59#	57-75	57-75#	57-75#	
	57-82	57-82#	57-82#	57-91	57-91#	57-91#	57-101	57-101#	57-101#	57-166	57-166#	57-167	57-167#	57-171	
	57-171#	57-184	57-184#	57-202	57-202#	57-206	57-206#	57-211	57-211#	57-214	57-214#	57-226	57-226#	57-235	
	57-235#	57-236	57-236#	57-239	57-239#	57-247	57-247#	57-250	57-250#	57-251	57-251#	57-252	57-252#	58-16	
	58-16#	58-57	58-57#	58-57#	58-62	58-62#	58-62#	58-198	58-198#	58-199	58-199#	58-200	58-200#	58-201	
	58-201#	58-202	58-202#	58-203	58-203#	58-204	58-204#	58-205	58-205#	58-206	58-206#	58-208	58-208#	59-12	
	59-12#	60-27	60-27#	60-37	60-37#	60-39	60-39#	60-55	60-55#	61-12	61-12#	61-14	61-14#	61-46	
	61-46#	61-46#	61-51	61-51#	61-51#	61-68	61-68#	61-68#	61-73	61-73#	61-73#	61-86	61-86#	61-87	
	61-87#	61-88	61-88#	61-90	61-90#	62-67	62-67#	62-152	62-152#	62-154	62-154#	62-168	62-168#	63-13	
	63-13#	63-38	63-38#	63-45	63-45#	63-48	63-48#	63-49	63-49#	63-73	63-73#	63-73#	63-74	63-74#	
	63-80	63-80#	63-80#	63-84	63-84#	63-84#	63-91	63-91#	63-91#	63-103	63-103#	63-106	63-106#	63-117	
	63-117#	64-83	64-83#	64-169	64-169#	64-171	64-171#	64-186	64-186#	65-75	65-75#	65-169	65-169#	65-171	
	65-171#	65-185	65-185#												
M$WORD	39-6	39-6#	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45	42-45#	49-47	49-47	
	49-47	49-47#	49-86	49-86	49-86	49-86#	49-101	49-101	49-101	49-101#	49-109	49-109	49-109	49-109#	
	50-87	50-87	50-87	50-87#	50-93	50-93	50-93	50-93#	51-42	51-42	51-42	51-42#	51-60	51-60	
	51-60	51-60#	52-24	52-24	52-24	52-24#	52-35	52-35	52-35	52-35#	53-83	53-83	53-83	53-83#	
	53-97	53-97	53-97	53-97#	54-43	54-43	54-43	54-43#	54-55	54-55	54-55	54-55#	54-60	54-60	
	54-60	54-60#	54-70	54-70	54-70	54-70#	55-29	55-29	55-29	55-29#	55-42	55-42	55-42	55-42#	
	55-49	55-49	55-49	55-49#	55-114	55-114	55-114	55-114#	55-124	55-124	55-124	55-124#	55-129	55-129	
	55-129	55-129#	55-191	55-191	55-191	55-191#	55-244	55-244	55-244	55-244#	55-250	55-250	55-250	55-250#	
	56-43	56-43	56-43	56-43#	57-59	57-59	57-59	57-59#	57-75	57-75	57-75	57-75#	57-82	57-82	
	57-82	57-82#	57-91	57-91	57-91	57-91#	57-101	57-101	57-101	57-101#	58-57	58-57	58-57	58-57#	
	58-62	58-62	58-62	58-62#	61-14#	61-46	61-46	61-46	61-46#	61-51	61-51	61-51	61-51#	61-68	
	61-68	61-68	61-68#	61-73	61-73	61-73	61-73	61-73#	63-49#	63-73	63-73	63-73#	63-74#	63-80	
	63-80	63-80	63-80#	63-84	63-84	63-84	63-84#	63-91	63-91	63-91	63-91#	66-5	66-5#	66-6	
	66-6#	66-7	66-7#	66-8	66-8#	66-9	66-9#	66-10	66-10#	66-25	66-25#	66-26	66-26#	66-32	

H13