

IDENTIFICATION

PRODUCT CODE: AC-F366B-MA
PRODUCT NAME: AJRLIB0 RL8-A/RL02 DR PT2
PRODUCT DATE: SEPT, 1981
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP
AUTHOR: JACK RICH

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

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

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

COPYRIGHT (C) 1978,1981 DIGITAL EQUIPMENT CORPORATION

- S.1.1 PROGRAM PURPOSE (ABSTRACT)
- S.1.2 SYSTEM REQUIREMENTS
 - S.1.2.1 HARDWARE REQUIREMENTS
 - S.1.2.2 SOFTWARE REQUIREMENTS
- S.1.3 RELATED DOCUMENTS AND STANDARDS
- S.1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
- S.1.5 ASSUMPTIONS
- S.2.0 OPERATING INSTRUCTIONS
 - S.2.1 LOADING AND STARTING PROCEEDURES
 - S.2.2 SPECIAL ENVIRONMENTS
 - S.2.3 PROGRAM OPTIONS
 - S.2.3.1 OPERATOR PROMPTS
 - S.2.3.2 MEANING OF SWITCH REGISTER BITS
 - S.2.3.3 CONSOLE PACKAGE COMMAND SUMMARY
 - S.2.4 EXECUTION TIMES
- S.3.0 ERROR INFORMATION
 - S.3.1 ERROR REPORTING PROCEEDURES
 - S.3.2 ERROR HALTS
- S.4.0 PROGRESS REPORTS
- S.5.0 DEVICE INFORMATION TABLES
 - S.5.1 INSTRUCTION SET
 - S.5.2 CONTROLLER REGISTERS AND DRIVE WORDS
- S.6.0 SUBTEST SUMMARIES
- S.7.0 PROGRAM, SYMBOL TABLE, AND CROSS REFERENCE LISTING

S.1.1 PROGRAM PURPOSE (ABSTRACT)

THIS PROGRAM TESTS AN RL8A OR RL278 CONTROLLER AND UP TO FOUR
RL02 DRIVES FOR ALL READ AND WRITE RELATED FUNCTIONS.

S.1.2 SYSTEM REQUIREMENTS

S.1.2.1 HARDWARE REQUIREMENTS

PDP8/E,F,M,OR A WITH 8K MEMORY
VT278 WITH 16-32K OF MEMORY
CONSOLE DEVICE (ASR 33 OR EQUIVALENT)
RL8A CONTROLLER WITH ONE OR MORE RL02 DRIVES CONNECTED
RL278 CONTROLLER WITH ONE OR MORE RL02 DRIVES CONNECTED.

S.1.2.2 SOFTWARE REQUIREMENTS

THIS PROGRAM IS FULLY COMPATABLE WITH OS/8 AND APT8A.

S.1.3 RELATED DOCUMENTS AND STANDARDS

THIS PROGRAM ADHERES TO THE DIAGNOSTIC ENGINEERING STANDARDS AND
CONVENTIONS DOCUMENT NO. 175-003-009-02.

S.1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

AJRLA (DISKLESS CONTROLLER DIAGNOSTIC) AND AJRLB (DRIVE TEST PART 1)
SHOULD HAVE BEEN SUCCESSFULLY RUN ON THE CONTROLLER AND DRIVES
BEING USED PRIOR TO RUNNING THIS PROGRAM.

THIS PROGRAM IS NORMALLY THE THIRD PROGRAM RUN WHEN CHECKING
OUT AN RL8A/RL02 OR RL278/RL02 SUBSYSTEM.

S.1.5 ASSUMPTIONS

THE PACKS BEING USED MUST HAVE NO MORE THAN 16 MANUFACTURING BAD
AND 16 FIELD BAD SECTORS. ALL 40 HEADERS ON CYLINDER 0, HEAD 0
SHOULD BE GOOD.

THE CPU, CONSOLE DEVICE, AND THE FIRST 8K OF MEMORY ARE ASSUMED
TO BE FAULT FREE. IF A FAULT IS SUSPECTED IN THESE UNITS, THE
APPROPRIATE DIAGNOSTICS SHOULD BE RUN FIRST.

ANY USER ATTEMPTING TO DIAGNOSE THE HARDWARE (I.E. USE THE DIAGNOSTIC
FOR OTHER THAN GO/NO-GO TESTING) IS ASSUMED TO HAVE SOME KNOWLEDGE OF
THE CONTROLLER AND RL02 HARDWARE.

S.2.1 LOADING AND STARTING PROCEEDURES

THE PROGRAM MAY BE LOADED BY PAPER TAPE, APT, OR STANDARD OS/8 LOADING PROCEEDURES. THE STARTING ADDRESS IS 200. THERE IS NO OTHER START OR RESTART ADDRESS AND NO WAY TO SELECT A SUBSET OF THE TESTS TO BE EXECUTED.

S.2.2 SPECIAL ENVIRONMENTS

THE PROGRAM MAKES NO ATTEMPT TO USE THE CONSOLE DEVICE WHEN RUNNING UNDER APT (LOCATION 22, BIT 0=1). ALL ERRORS ARE FATAL AND REPORTED TO APT. APT IS NOTIFIED THAT THE PROGRAM IS STILL RUNNING IN THE STANDARD MANNER.

THE FOLLOWING BITS ARE RELAVENT IN THE HARDWARE CONFIGURATION WORDS:

LOC 21	BIT 0	=0	USE PSEUDO SWITCH REGISTER
		=1	USE HARDWARE SWITCH REGISTER
	BIT 1	=0	NO OPTION 1
		=1	OPTION 1 IS AVAILABLE
LOC 22	BIT 0	=0	NOT ON APT
		=1	ON APT 8A SYSTEM
	BITS 4,5	=0	BITS 10,11 SPECIFY HIGHEST UNIT NUMBER TO TEST. ALL UNITS FROM 0 TO THIS NUMBER ARE TESTED.
		=1	BITS 10,11 SPECIFY A SPECIFIC UNIT NUMBER. ONLY THAT UNIT IS TESTED.
		=2,3	RESERVED FOR FUTURE EXPANSION (THE PROGRAM WILL REPORT AN ERROR.)
	BITS 10,11		UNIT NUMBER AS DEFINED BY BITS 4,5.

S.2.3 PROGRAM OPTIONS

S.2.3.1 OPERATOR PROMPTS

THE CONSOLE PACKAGE (DESCRIBED BELOW) IS ALWAYS ACTIVE AND CANNOT BE DISABLED. THE PROGRAM DEFAULTS TO USE OF THE PSEUDO SWITCH REGISTER (LOCATION 20). THE HARDWARE SWITCH REGISTER CAN BE USED BY CHANGING LOCATION 21 BIT 0 TO A 1. THE PSEUDO SWITCH REGISTER CAN BE MODIFIED BY USING THE CONSOLE PACKAGE.

IF THE PSEUDO SWITCH REGISTER IS BEING USED, THE PROGRAM WILL ALLOW THE OPERATOR TO ENTER THE SWITCH REGISTER VALUE AT PROGRAM STARTUP.

"NO" IS THE DEFAULT ON ALL YES/NO QUESTIONS AND MAY BE GOTTEN BY TYPING "<CR>", AS WELL AS "N". "YES" IS GOTTEN BY TYPING "Y". NOTE THAT THE PROGRAM KNOWS ONLY ABOUT DRIVES 0-3. IF DRIVE ADDRESS PLUGS 4-7 ARE BEING USED, THEY ARE CONSIDERED EQUIVALENT TO DRIVES 0-3, RESPECTIVELY.

THE OPERATOR IS NEXT ASKED IF OPTION 1 IS AVAILABLE. THIS BOARD IS NEEDED FOR TIMING PURPOSES.

THE OPERATOR IS THEN ASKED IF HE WISHES TO SELECT NON-DEFAULT PARAMETERS. IF THE ANSWER IS "N" OR "<CR>", THE REMAINING QUESTIONS WILL NOT BE ASKED AND THE DEFAULTS FOR EACH OF THE QUESTIONS WILL BE USED. IF THE OPERATOR ANSWERS WITH "Y", THE QUESTIONS WILL BE ASKED. THE DEFAULT MAY BE GOTTEN BY TYPING "<CR>". "NO" IS THE DEFAULT TO ALL Y/N QUESTIONS. THE ANSWERS TO THE QUESTIONS APPLY TO ALL DRIVES UNDER TEST. THE QUESTIONS AND THEIR MEANINGS ARE LISTED BELOW:

"USE DEVICE CODES 62,63?"

IF NO, THE STANDARD DEVICE CODES 60,61 ARE USED.

"EXECUTE WRITE LOCK DATA PROTECTION TEST?"

IF YES, THE OPERATOR WILL BE REQUESTED TO PRESS WRITE LOCK FOR EACH DRIVE IN THE FIRST PASS ONLY.

"USE ALL CYLINDERS?"

IF YES, ALL CYLINDERS (EXCEPT 377) WILL BE USED FOR READ/WRITE TESTS ON SECOND AND SUBSEQUENT PASSES.

"USE LOWER (UPPER) CYLINDER LIMIT?"

IF NO, THE LIMIT WILL BE THE DEFAULT OF 0 (377).

IF YES, THE OPERATOR CAN ENTER THE DESIRED LIMIT IN OCTAL. THE PROGRAM WILL NOT USE CYLINDERS BELOW (OR ABOVE) THE CYLINDER LIMIT FOR READ/WRITE OPERATIONS EXCEPT WHERE NOTED.

"USE ONLY ONE SURFACE?"

IF NO, BOTH SURFACES OF THE DISK WILL BE USED.

IF YES, THE OPERATOR WILL BE ASKED TO ENTER THE SURFACE NUMBER (0 OR 1). ONLY THIS SURFACE WILL BE USED.

"SPECIFY ERROR LIMIT IN OCTAL:"

THE DEFAULT IS 24 (OCTAL). ANY VALUE FROM 0 TO 7777 MAY BE SELECTED. (0 IS EQUIVALENT TO 1.) IF SWITCH REGISTER BIT 8 IS SET, THE DRIVE UNDER TEST WILL BE DROPPED IF THIS ERROR LIMIT IS REACHED OR EXCEEDED.

 THE SWITCH REGISTER BITS AND THEIR MEANINGS ARE DESCRIBED BELOW:

SR BIT SETTING MEANING

SR0 =0 HALT ON ERROR
 (4000) =1 INHIBIT ERROR HALT

SR1 =0 DO NOT LOOP IF ERROR (OVERRIDDEN IF SR2=1)
 (2000) =1 LOOP ON ERROR (HARD OR SOFT)

SR2 =0 MAKE NORMAL LOOP DECISIONS
 (1000) =1 LOOP ON CURRENT TEST

SR3 =0 CONTINUE AFTER END OF PASS
 (0400) =1 HALT AT END OF PASS

SR4 =0 REPORT ERROR ON CONSOLE DEVICE
 (0200) =1 INHIBIT ERROR REPORT
 NOTE: CARE SHOULD BE TAKEN WHEN USING THE HARDWARE
 SWITCHES THAT THIS BIT IS NOT INADVERTENTLY
 LEFT SET WHEN STARTING THE PROGRAM.

SR5 =0 PROGRAM WILL PROMPT OPERATOR NORMALLY (AS
 (0100) DESCRIBED IN S.2.3.1
 =1 PROGRAM WILL USE CURRENT STATE ON A START OR RESTART.
 ON PROGRAM RESTART, THIS CORRESPONDS TO PREVIOUS
 RESPONSES, EXCEPT THAT ANY DRIVES THAT HAVE BEEN
 DROPPED WILL REMAIN UNSELECTED. ON FIRST PROGRAM
 START, NO DRIVES WILL BE SELECTED.

SR6 =0 REPORT END OF PASS
 (0040) =1 INHIBIT END OF PASS TYPEOUT

SR7 =0 TEST 35 (12 BIT MODE R/W TEST) WILL EXECUTE NORMALLY
 (0020) =1 TEST 35 WILL LOOP INDEFINITELY ON THE FIRST DATA PATTERN.

SR8 =0 THE DRIVE WILL NOT BE DROPPED UNLESS IT IS NOT READY
 (0010) =1 THE DRIVE WILL BE DROPPED WHEN THE ERROR LIMIT IS
 REACHED.

SR9 =0 NORMAL OPERATION--READ, VERIFY, AND USE THE BAD SECTOR FILES
 (0004) =1 DO NOT READ BAD SECTOR FILES. (FOR USE WHEN IT IS KNOWN
 THAT THE BAD SECTOR FILES ARE DESTROYED.)

 A MORE COMPLETE DESCRIPTION OF THE CONSOLE PACKAGE MAY BE FOUND IN DOCUMENT NO. 175-003-009-02. BELOW IS A SUMMARY OF AVAILABLE CONTROL FUNCTIONS. A "CONTROL" CHARACTER IS TYPED BY HOLDING DOWN THE "CONTROL" KEY AND HITTING THE CHARACTER.

CNTRL-G OPEN THE PSEUDO SWITCH REGISTER FOR MODIFICATION, THE PROGRAM WILL TYPE "^G" FOLLOWED BY:
 "SR=XXXX ", WHERE XXXX IS THE SWITCH REGISTER SETTING. THE VALUE MAY BE LEFT UNCHANGED OR A NEW VALUE MAY BE ENTERED. TYPING A <CR> WILL SAVE THE NEW VALUE (IF ONE WAS ENTERED) AND CONTINUE THE PROGRAM. TYPING A LINE FEED WILL SAVE THE NEW VALUE (IF ENTERED) AND RESTART THE PROGRAM AT LOCATION 200. TYPING ANY OTHER CHARACTER WILL CAUSE THE PROMPT TO BE RE-ISSUED.
 NOTE: IF THE HARDWARE SWITCHES ARE BEING USED, THEIR VALUE WILL BE TYPED OUT, BUT ANY NEW VALUE TYPED IN WILL BE SAVED IN LOCATION 20. THE HARDWARE SWITCHES WILL STILL CONTROL THE PROGRAM.

CNTRL-C RETURN TO OS/8 MONITOR (AT LOCATION 7600). IF THE MONITOR BOOT IS NOT PRESENT, THE RESULTS ARE INDETERMINATE.

CNTRL-S DISABLE ALL OUTPUT. THE PROGRAM WILL HANG WHEN IT ATTEMPTS TO OUTPUT TO THE CONSOLE DEVICE. AT THIS POINT, THE ONLY CHARACTERS THAT WILL BE RECOGNIZED ARE THE CONTROL-C AND CONTROL-Q.

CNTRL-Q REENABLE ALL OUTPUT (AFTER A ^S).

CNTRL-F THIS IS A NON-STANDARD CONTROL CHARACTER. THE CONSOLE FILL COUNT (LOCATION 23) IS OPENED FOR MODIFICATION. BEHAVIOR IS SIMILAR TO THAT FOR CONTROL-G, EXCEPT THAT A LINE FEED WILL NOT BE RECOGNIZED. THE DEFAULT FILL COUNT IS 1. A FILL COUNT OF 14 IS USUALLY REQUIRED FOR LA30'S.

ALL OTHER CHARACTERS WILL BE ECHOED, FOLLOWED BY A "?" AND <CR>.

S.2.4 EXECUTION TIMES

FIRST PASS IS QUICK VERIFY AND REQUIRES APPROX. 1 MINUTE/DRIVE. SUBSEQUENT PASSES REQUIRE APPROX. 5 MINUTES/DRIVE. ALL PASSES WILL BE DECREASED PROPORTIONATELY IF USING CYLINDER LIMITS OR ONLY ONE SURFACE. ALL PASS TIMES AFTER THE FIRST WILL BE INCREASED GREATLY BY USING ALL CYLINDERS.

S.3.1 ERROR REPORTING PROCEEDURES

ALL ERRORS HAVE THE FOLLOWING FORMAT:

ERROR MESSAGE
 DATA HEADERS
 DATA

THE ERROR MESSAGE DESCRIBES THE FAILURE. DATA HEADERS VARY IN NUMBER AND IDENTIFY THE DATA TYPED BELOW EACH HEADER. ALL DATA ARE TYPED IN OCTAL. THE PC OF THE ERROR CALL IS ALWAYS INCLUDED.

THE FOLLOWING ABBREVIATIONS ARE USED:

CA COMMAND A REGISTER
 CB COMMAND B REGISTER
 SA SECTOR ADDRESS REGISTER
 WC WORD COUNT REGISTER
 ER ERROR REGISTER
 MA MEMORY ADDRESS REGISTER
 DRV NO. THE DRIVE NUMBER OF THE FAILING DRIVE (0-3 ONLY)
 EXPCTD EXPECTED (I.E. THE CORRECT VALUE)
 GOOD SAME AS EXPCTD
 ACTUAL THE ACTUAL (INCORRECT) VALUE
 BAD SAME AS ACTUAL
 CYL CYLINDER ADDRESS
 HED-CYL ACTUAL POSITION OF THE HEADS (IN NORMAL COMMAND A FORMAT)
 WD1-WD2 WORD1 AND WORD2 OF HEADERS OR DRIVE STATUS
 CALLPC ERRORS REPORTED WITHIN SUBROUTINES INCLUDE THE PC OF THE SUBROUTINE CALL

S.3.2 ERROR HALTS

THE ONLY ACTUAL HALT INSTRUCTION IS WITHIN THE POWER FAIL ROUTINE. ALL OTHER "HALTS" ARE CALLS TO THE CONSOLE PACKAGE SWITCH REGISTER MODIFICATION ROUTINE. THIS IS TRUE EVEN IF THE HARDWARE SWITCHES ARE SELECTED. THE ONLY SUCH "HALTS" ARE AT THE SELECT FIRST DRIVE ROUTINE IF NO DRIVES ARE AVAILABLE, IN THE END OF PASS ROUTINE, AND THE ERROR REPORTING ROUTINE.

 ONLY ONE PROGRESS REPORT IS GIVEN; THAT IS GIVEN AT THE END OF
 PASS. IT HAS THE FOLLOWING FORMAT:

END PASS XXXX

WHERE XXXX IS THE PASS COUNT IN OCTAL. NO TOTAL ERROR
 COUNT IS REPORTED, SO WHEN RUNNING THE PROGRAM UNATTENDED,
 IT IS RECOMMENDED THAT EITHER ERROR HALTS, ERROR PRINTOUTS,
 OR LOOPING ON ERROR BE ENABLED TO AVOID PASSING A DEFECTIVE
 BOARD.

S.5.0 DEVICE INFORMATION TABLES

S.5.1 INSTRUCTION SET

TRANSFERS FROM THE AC TO REGISTER IN THE CONTROLLER
 CLEAR THE AC AFTER THE TRANSFER IS COMPLETE. TRANSFERS
 TO THE AC FROM REGISTERS IN THE CONTROLLER CLEAR THE AC
 FIRST THEN THE TRANSFER TAKES PLACE.

THE SKIP INSTRUCTIONS IN THIS INSTRUCTION SET ARE
 SKIP AND THEN CLEAR IOT'S. THIS MEANS THAT IF A
 GIVEN CONDITION IS TRUE, IE; "FUNCTION DONE" IS TRUE
 (SET TO A LOGIC ONE) THE FUNCTION DONE FLAG WILL
 BE CLEARED AT THE COMPLETION OF THE SKIP IOT.

THE DEVICE CODE IS JUMPER SELECTABLE FOR EITHER 60,61 OR 30,31.
 60,61 IS STANDARD. IOT CODES 30,31 MAY BE USED BY SETTING SWITCH
 REGISTER BIT 11 TO A 1.

----- 6600	----- RLDC	----- CLEAR DEVICE, ALL REGISTERS, AC AND FLAGS.
6601	RLSD	SKIP ON FUNCTION DONE, THEN CLEAR IF SET TO A ONE.
6602	RLMA	LOAD BREAK MA REGISTER FROM AC 0:11
6603	RLCA	LOAD COMMAND REGISTER "A" FROM AC 0:11
6604	RLCB	LOAD COMMAND REGISTER "B" FROM AC 0:11, EXECUTE COMMAND
6605	RLSA	LOAD SECTOR ADDRESS FROM AC 0:5
6606	----	SPARE (WILL CLEAR THE AC)
6607	RLWC	LOAD WORD COUNT FROM AC 0:11
6610	RRER	READ ERROR REGISTER INTO AC 0:2,11.
6611	RRWC	READ WORD COUNT INTO AC 0:11
6612	RRCA	READ COMMAND REGISTER "A" INTO AC 0:11
6613	RRCB	READ COMMAND REGISTER "B" INTO AC 0:11
6614	RRSA	READ SECTOR ADDRESS INTO AC 0:5
6615	RRSI	READ SILO WORD (8 BIT) INTO AC 4:11
6616	----	SPARE (AC NOT AFFECTED)
6617	RLSE	SKIP ON COMPOSITE ERROR, THEN CLEAR IF SET TO A ONE

BREAK MA REGISTER

AC0- MSB
"
"
AC11-LSB

COMMAND REGISTER "A"

AC0- DIRECTION; ZERO= MOVE HEADS AWAY
FROM SPINDLE (LOWER CYL. ADD.)
ONE= MOVE TOWARDS SPINDLE (HI ADD.)
AC1- HEAD SELECT- ZERO= UPPER HEAD
ONE= LOWER HEAD
AC2- SPARE
AC3- RESERVED FOR DISK EXPANSION
AC4- CYLINDER ADDRESS/DIFFERENCE WORD; MSB
"
"
AC11- CYLINDER ADDRESS/DIFFERENCE WORD; LSB

COMMAND REGISTER "B"

AC0- MAINT. INHIBIT. PREVENT THE FOLLOWING SIGNALS
FROM GOING TO/FROM THE DRIVE;
WRT GATE, WRT DATA, DRV CMND, DRV STAT., DRV RDY
SEC PLS, READ DATA, STAT. CLK, DRV ERR.
AC1- MAINTENANCE- LOOP DAR TO SILO SERIAL IN
AC2- MODE; ZERO= TRUNCATED (128 12 BIT WORDS/SECTOR)
ONE= BYTE (256 8 BIT WORDS/SECTOR)
MUST BE SET TO A 1 WHEN DOING
A "GET STATUS" OR "READ HEADER" COMMAND.
AC3- INTERRUPT ENABLE
AC4- DRIVE SELECT; MSB
AC5- DRIVE SELECT; LSB
AC6- EMA; MSB
AC7- "
AC8- EMA; LSB
AC9- FUNCTION BIT C
AC10- " " B
AC11-FUNCTION BIT A

FUNCTION BIT DEFINITION

BIT C	BIT B	BIT A	COMMAND
-----	-----	-----	-----
0	0	0	MAINTENANCE
0	0	1	RESET
0	1	0	GET STATUS
0	1	1	SEEK
1	0	0	READ HEADER
1	0	1	WRITE DATA
1	1	0	READ DATA
1	1	1	READ DATA WITHOUT HEADER CHECK

AC0- SECTOR ADDRESS; MSB

"

AC5- SECTOR ADDRESS; LSB

DISK STATUS REGISTER

WORD #1

AC4- SPARE (0)
 AC5- HEAD SELECT
 AC6- COVER OPEN *
 AC7- HEADS OUT
 AC8- BRUSH HOME
 AC9- STATE C
 AC10-STATE B
 AC11-STATE A

WORD #2

AC4- WRITE DATA ERROR *
 AC5- HEAD CURRENT ERROR *
 AC6- WRITE LOCK STATUS
 AC7- SEEK TIME OUT ERROR *
 AC8- SPIN UP TIME OUT ERROR *
 AC9- WRITE GATE ERROR *
 AC10-VOLUME CHECK *
 AC11-DRIVE SELECT ERROR *

*CAUSES DRIVE ERROR TO SET TO A ONE (1)

STATE BIT DEFINITION

BIT C	BIT B	BIT A	DEFINITION
----	----	----	-----
0	0	0	LOAD STATE
0	0	1	SPIN-UP
0	1	0	LOAD HEADS
0	1	1	BRUSH CYCLE
1	0	0	SEEK-TRACK COUNTING
1	0	1	SEEK LINEAR MODE
1	1	0	UNLOAD HEADS
1	1	1	SPIN-DOWN

HEADER WORDS

WORD #1

AC4- CYLINDER ADDRESS 0 (LSB)
 AC5- HEAD SELECT
 AC6- SECTOR ADDRESS 5 (MSB)
 "
 AC11-SECTOR ADDRESS 0 (LSB)

WORD #2

AC4- 0
 AC5- CYLINDER ADDRESS 7 (MSB)
 "
 AC11-CYLINDER ADDRESS 1

WORDS #3 & #4

ALL 0'S

AC0- CRC ERROR
 AC1- OPERATION INCOMPLETE (OPI)
 AC2- DATA LATE/HEADER NOT FOUND
 AC10-DRIVE ERROR
 AC11-DRIVE READY; A ONE= DRV RDY
 A ZERO= DRV RDY(NOT)
 NOTE:DRIVE ERROR AND DRIVE READY ARE ASSERTED WHEN
 DRIVE INTERFACE IS DISABLED (CB0 = 1).

WORD COUNT REGISTER; A 12 BIT WORD COUNT
 REGISTER HAS BEEN PROVIDED TO ALLOW UP TO
 4,096 DATA BREAKS (DMA'S) TO TAKE PLACE AT
 ONE TIME. THIS REGISTER IS LOADED WITH
 THE RLWC IOT FROM AC 0:11, WHICH MUST BE
 THE 2'S COMPLIMENT OF THE NUMBER OF TRANSFERS
 THAT ARE TO TAKE PLACE.

S.6.0 SUBTEST SUMMARIES -----

DRIVE PART 2 -- READ/WRITE TESTS

HEADER NOT FOUND TEST (CONTROLLER TEST)

POSITION HEADS AT CYLINDER 0, HEAD 0.

DO READ DATA TO HEAD 0, SECTOR 0, BUT TO CYLINDER 1. VERIFY
 OPI-HNF SET IN ERROR REG. FLOAT A 1 THROUGH CYLINDER ADDRESS
 FOR ATTEMPTED READS AND VERIFY OPI-HNF SETS.

ATTEMPT A READ TO CYLINDER 0, HEAD 0, SECTORS 50, 60, AND 70
 (OCTAL). VERIFY OPI-HNF SET.

ATTEMP READ TO CYLINDER 0, HEAD 1, SECTOR 0 AND VERIFY OPI-HNF
 SET. SELECT HEAD 1 AND VERIFY READ TO HEAD 0 CAUSES OPI-HNF.

NOTE: CYLINDER LIMITS AND SURFACE SELECT WILL BE IGNORED ON
 THIS TEST.

TEST 29 BASIC READ DATA TEST

POSITION HEADS AT CYLINDER 255.
 DO READ DATA, HEAD 1. CHECK FOR ANY ERRORS AND REPORT. IF
 ERROR, READ SECTOR 2 THROUGH 18 (EVEN ONLY) UNTIL NO ERROR ON
 READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF
 NONE CAN BE READ, SUCCESSFULLY, REPORT THAT FACTORY BAD SECTOR
 FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH
 READ OF SECTOR 20.

ARE NOT 0, WORD 2 AND 3 ARE 0, LOCATE FIRST WORD OF ALL ONE'S AND THAT WORD TO WORD 127 ARE ALL ONE'S.) STORE BAD SECTOR DATA.

READ DATA, HEAD ONE, SECTOR 20. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 22 THROUGH 38 (EVEN ONLY) UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ SUCCESSFULLY, REPORT THAT SOFTWARE BAD SECTOR FILES CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT TEST.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA AS ABOVE. STORE BAD SECTOR DATA.

NOTE 1: SURFACE SILECT WILL BE IGNORED ON THIS TEST. THIS TEST WILL BE PERFORMED EVEN IF SURFACE 0 ONLY IS SELECTED.

NOTE 2: A SWITCH REGISTER BIT WILL ALLOW THIS TEST TO BE BYPASSED. THIS FEATURE IS INTENDED FOR USE ONLY WHEN IT IS KNOWN THAT THE BAD SECTOR FILES HAVE BEEN DESTROYED.

TEST 30 WRITE/READ DATA TEST (PART 1)

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. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 31 CRC ERROR DETECTION TEST

POSITION HEADS AT CYLINDER 0, HEAD 0. (IF SURFACE 1 ONLY IS SELECTED, THEN HEAD 1 WILL BE USED.)

WRITE PATTERN 0 ON SECTOR 0. ISSUE A WRITE COMMAND TO WRITE PATTERN 1 ON SECTOR 0, BUT MONITOR THE WORD COUNT REGISTER AND WHEN THE 17TH (DECIMAL) DATA BREAK IS SEEN (INDICATING THAT THE WRITE OF THE SECTOR HAS STARTED), ISSUE A CAF TO TERMINATE THE WRITE.

READ BACK SECTOR 0 AND VERIFY THAT CRC ERROR SETS.

NOTE: CYLINDER LIMITS ARE IGNORED ON THIS TEST.

POSITION HEADS AT CYLINDER 0, HEAD 0. (USE HEAD 1 IF HEAD 1 ONLY IS SELECTED.)

WRITE ALL 1'S ON FIRST GOOD SECTOR. READ AND VERIFY DATA.

WRITE 1 WORD OF 1'S ON SAME SECTOR. READ AND VERIFY DATA AND ZERO FILL.

REPEAT FOR BOTH 8 AND 12 BIT MODES. NOTE THAT THIS WILL OVERWRITE CRC ERROR CAUSED BY PREVIOUS TEST.

TEST 33 WRITE/READ TEST (PART 2)

CC IS CURRENT CYLINDER SELECTED FROM SET.

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

SEEK TO 0. SEEK FORWARD TO CC. READ/COMPARE ALL DATA. SEEK TO 255. 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 TO 255. SEEK REVERSE TO CC. READ/COMPARE ALL DATA. SEEK REVERSE TO 0. 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, THE NEXT HIGHER CYLINDER WILL BE USED.

NOTE 2: IF PROGRAM MODE 2 IS USED AND 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 SIX OF THE CYLINDERS LISTED WITHIN THE DEFAULT CYLINDER SET (EVERY EIGHTH ENTRY).

NOTE 4: TESTING WILL BE DONE WITHIN THE UPPER AND LOWER CYLINDER LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

DO WRITE DATA PATTERN 0 AT CYLINDER 0, 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. VERIFY DRIVE 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.

TEST 35 12 BIT MODE WRITE/READ TEST (CONTROLLER TEST)

POSITION HEADS AT THE FIRST CYLINDER THAT IS ABOVE THE LOWER SEEK LIMIT. USE HEAD 0 UNLESS HEAD 1 ONLY IS SELECTED.

ISSUE WRITE COMMAND FOR A 2K TRANSFER IN TWELVE BIT MODE. READ BACK THE DATA AND VERIFY. REPEAT FOR EACH PATTERN.

THE FOLLOWING REPEATING THREE WORD PATTERNS WILL BE USED:
6666, 5555, 3333; 2525, 5252, 2525; 7777, 0000, 6314

NOTE: A SWITCH REGISTER BIT WILL FORCE LOOPING ON THIS TEST (USING THE FIRST PATTERN). THIS LOOPING FEATURE IS INTENDED TO FORCE MAXIMAL SILO USAGE ON THE CONTROLLER AND IS NOT INTENDED FOR USE IN DRIVE OR DISK PROBLEMS (BECAUSE OF THE NON-WHOLE WORD COUNT PER SECTOR ON 12-BIT MODE WRITES.)

WRITE AN ENTIRE TRACK WITH 0'S. CHOOSE ONE SECTOR AND WRITE IT WITH OTHER DATA. READ HEADERS UNTIL THE PREVIOUS HEADER IS READ AND ISSUE READ DATA WITHOUT HEADER CHECK. VERIFY THAT THE CORRECT SECTOR WAS READ.

TEST 37 ADJACENT CYLINDER INTERFERENCE TEST

CC IS CURRENT CYLINDER SELECTED FROM SET
DATA PATTERN IS WORST CASE.

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

SEEK FORWARD TO 255. SEEK REVERSE TO CC-1. WRITE PATTERN.
SEEK FORWARD TO 255. SEEK REVERSE TO CC+1. 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 0. SEEK FORWARD TO CC. READ/COMPARE DATA
FROM ALL SECTORS. ANY ERRORS (READ OR COMPARE) ARE ATTRIBUTED
TO ADJACENT CYLINDER INTERFERENCE.

SEEK FORWARD TO 255. SEEK REVERSE TO CC. WRITE DATA PATTERN.
SEEK REVERSE TO 0. SEEK FORWARD TO CC-1, WRITE PATTERN. SEEK
REVERSE TO 0, SEEK FORWARD TO CC+1, WRITE PATTERN. SEEK
FORWARD TO 255, 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,
THE NEXT HIGHER CYLINDER WILL BE TRIED.

NOTE 2: IF PROGRAM MODE 2 IS USED AND THE "USE ALL CYLINDERS"
PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE
ALL CYLINDERS (EXCEPT 0 AND 255) IN THE SELECTED
PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM, THIS TEST IS
EXECUTED ON ONLY THREE OF THE CYLINDERS LISTED IN THE
DEFAULT PARAMATER SET. THOSE USED WILL BE THE FIRST,
TWENTY-FIRST, AND FORTY-FIRST ENTRIES. ON SECOND AND
SUBSEQUENT PASSES, EVERY FOURTH CYLINDER SET ENTRY
WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER
LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
TO THAT SURFACE.

CC IS CURRENT CYLINDER SELECTED FROM SET
PATTERN A = WORST CASE
PATTERN B = 000,000

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

SEEK FORWARD TO 255, SEEK REVERSE TO CC. WRITE PATTERN B.
SEEK REVERSE 0, SEEK FORWARD TO CC, READ/COMPARE DATA.

SEEK FORWARD TO 255, SEEK REVERSE TO CC. WRITE DATA PATTERN
A. READ/COMPARE DATA. SEEK REVERSE TO 0, SEEK FORWARD TO CC.
WRITE PATTERN B. SEEK FORWARD TO 255, 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,
THE NEXT HIGHER CYLINDER IS TRIED.

NOTE 2: IF PROGRAM MODE 2 IS USED AND 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 THREE OF THE CYLINDERS LISTED IN THE
DEFAULT PARAMATER SET. THOSE USED WILL BE THE FIRST,
TWENTY-FIRST, AND FORTY-FIRST ENTRIES. ON SECOND AND
SUBSEQUENT PASSES, EVERY FOURTH CYLINDER SET ENTRY
WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER
LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
TO THAT SURFACE.

S.7.0 PROGRAM, SYMBOL TABLE, AND CROSS REFERENCE LISTING

/AJRLI-B RL8A/RL02 READ/WRITE DIAGNOSTIC
/ REVISION HISTORY

```

/-----
/ ORIGINAL BY JACK RICH
/ MODIFICATIONS BY DATE
/-----
/ H. POULTER FEBRUARY 1979
/ HP 001 MODIFICATIONS TO ALLOW THE RL02 DRIVE TO RUN.
/ HP 002 ALLOW THE HEAD AND CYLINDER TO BE
/ DISPLAYED IN THE MQ
/-----
/ M. LETENDRE JANUARY 1981
/ ML 001 VERSION "D" CHANGES FOR THE VT278
/ COMPATABILITY UPGRADE

```

```

0000 FIELD 0
0000 *0
0000 0302 "B /REVISION B
0001 5402 JMP I .+1 /GO TO POWER FAIL ROUTINE
0002 4751 PWRFAL /POINTER TO POWER FAIL ROUTINE
0003 4765 PWRUP /POINTER TO POWER UP ROUTINE (A JMP I 3
/ INSTRUCTION IS PLACED AT 0)

0010 *10
0010 0000 AUTO10, 0 /TEMP USE ONLY
0011 0000 AUTO11, 0
0012 0000 AUTO12, 0
0013 0000 AUTO13, 0 /TEMP USE ONLY
0014 0000 AUTO14, 0
0015 0000 AUTO15, 0
0020 *20
0020 0000 PSWR, 0
0021 0000 HCW1, 0 /DEFAULT TO SOFTWARE SWITCHES--4000 FOR HARDWARE SWR
0022 0000 HCW2, 0
0023 0001 FILLER, 1 /NUMBER OF FILLER CHARACTERS REQUIRED
4424 RLDC= JMS I . /DEVICE CLEAR IOT SUBROUTINE CALL
0024 4647 XRLDC /POINTERS TO IOT ROUTINES
4425 RLSD= JMS I . /SKIP ON DONE IOT SUBROUTINE CALL
0025 4652 XRLSD
4426 RLMA= JMS I . /LOAD BREAK MA IOT SUBROUTINE CALL
0026 4657 XRLMA
4427 RLCA= JMS I . /LOAD COMMAND REG A IOT SUBROUTINE CALL
0027 4662 XRLCA
4430 RLCB= JMS I . /LOAD COMMAND REG B IOT SUBROUTINE CALL
0030 4665 XRLCB
4431 RLSA= JMS I . /LOAD SECTOR ADDRESS IOT SUBROUTINE CALL
0031 4670 XRLSA
4432 RLWC= JMS I . /LOAD WORD COUNT REG SUBROUTINE CALL
0032 4673 XRLWC
4433 RRER= JMS I . /READ ERROR REG IOT SUBROUTINE CALL
0033 4676 XRRER
4434 RRWC= JMS I . /READ WORD COUNT IOT SUBROUTINE CALL
0034 4701 XRRWC
4435 RRCA= JMS I . /READ COMMAND REG A IOT SUBROUTINE CALL
0035 4704 XRRCA
4436 RRCB= JMS I . /READ COMMAND REG B IOT SUBROUTINE CALL

```

0036	4707		XRRCB	.	
	4437	RRSA=	JMS I	.	/READ SECTOR ADDR IOT SUBROUTINE CALL
0037	4712		XRRSA	.	
	4440	RRSI=	JMS I	.	/READ SILO WORD IOT SUBROUTINE CALL
0040	4715		XRRSI	.	
	4441	RLSE=	JMS I	.	/SKIP ON DRIVE ERROR IOT SUBROUTINE CALL
0041	4720		XRLSE	.	
	4442	GETSR=	JMS I	.	/READ EITHER HARDWARE OR PSEUDO SWITCHES
0042	5135		XGETSR	.	
	4443	APTCHK=	JMS I	.	/SKIP IF NOT RUNNING UNDER APT
0043	5144		XAPTCH	.	
	4444	TICK=	JMS I	.	//GENERATE TIMING FOR APT
0044	4562		XTICK	.	
	4445	SCOPE=	JMS I	.	/LOOP CONTROL SUBROUTINE
0045	4000		XSCOPE	.	
	4446	ERROR=	JMS I	.	/ERROR HANDLER
0046	4032		XERROR	.	
	4447	CONSOL=	JMS I	.	/TELETYPE INPUT HANDLER
0047	4200		XCONSO	.	
	4450	UPARG=	JMS I	.	/TYPE "^G" THEN CALL CNTRLG
0050	4275		XUPARG	.	
	4451	CNTRLG=	JMS I	.	/CONTROL-G HANDLER--SWITCH REG MODIFY ROUTINE
0051	4305		XCTRLG	.	
	4452	CNTRLF=	JMS I	.	/CONTROL-F HANDLER (FILL COUNT ADJUSTMENT)
0052	4244		XCTRLF	.	
	4453	GETRES=	JMS I	.	/GET OPERATORS RESPONSE TO YES-NO QUES
0053	5152		XGETRE	.	
	4454	ENTVAL=	JMS I	.	/ALLOW OPERATOR TO ENTER VALUE
0054	5241		XENTVA	.	
	4455	GETNUM=	JMS I	.	/GET OCTAL NUMBER
0055	5200		XGETNU	.	
	4456	WZITFG=	JMS I	.	/CHECK IF LAST CHARACTER WAS ^F OR ^G
0056	4350		XWZITF	.	
	4457	RESET=	JMS I	.	/RESET DRIVE AND WAIT FOR DONE
0057	5012		XRESET	.	
	4460	GETSTA=	JMS I	.	/GET STATUS AND WAIT FOR DONE
0060	5025		XGETST	.	
	4461	ERRHAN=	JMS I	.	/HANDLE POSSIBLE DRIVE ERROR
0061	5607		XERRHA	.	
	4462	SEEKTO=	JMS I	.	/SEEK TO TRACK AND CYL CONTAINED IN AC
0062	5400		XSEEKT	.	
	4463	SKOCYL=	JMS I	.	/SEEK TO 0 THEN TO HEAD+CYLNDR
0063	5552		XSKOCY	.	
	4464	SKHCYL=	JMS I	.	/SEEK TO 777 THEN TO HEAD+CYLNDR
0064	5564		XSKHCY	.	
	4465	REDHDR=	JMS I	.	/ISSUE A READ HEADER AND WAIT FOR DONE. HEADER LEFT IN SILO
0065	5321		XREDHD	.	
	4466	SETTIM=	JMS I	.	/SET UP REAL TIME CLOCK
0066	5062		XSETTI	.	
	4467	TIMCHK=	JMS I	.	/CHECK FOR REAL TIME PASSAGE
0067	5105		XTIMCH	.	
	4470	JMPPM1=	JMS I	.	/JMP .-1 EXCEPT ALLOW CONSOLE INPUT
0070	4741		XJMPPM	.	
	4471	CLRRBF=	JMS I	.	/CLEAR READ BUFFER (-# WORDS TO CLEAR FOLLOWS CALL)
0071	7041		XCLRRB	.	

0072	4472	FILBUF= JMS I .	/FILEE BUFFER POINTED TO BY ARGUMENT WITH PATTERN POINTED TO BY PATTRN
	6074	XFILBU	
	4473	CMPPAT= JMS I .	/COMPARE BUFFER TO PATTRN
0073	6124	XCMPPA	
	4474	WRTTRK= JMS I .	/WRITE AN ENTIRE TRACK WITH DATA IN BUFFER
0074	6400	XWRTTR	
	4475	CHKTRK= JMS I .	/CHECK ENTIRE TRACK AGAINST DATA PATTERNS
0075	6442	XCHKTR	
	4476	CYLBAD= JMS I .	/CHECK IF CYLNDR HAS BAD SECTORS
0076	6706	XCYLBA	
	4477	WRIT8= JMS I .	/ISSUE 8 BIT WRITE
0077	6200	XWRIT8	
	4500	READ8= JMS I .	/ISSUE 8 BIT READ
0100	6221	XREAD8	
	4501	WRIT12= JMS I .	/12 BIT WRITE
0101	6242	XWRIT1	
	4502	READ12= JMS I .	/12 BIT READ
0102	6263	XREAD1	
	4503	APTCOM= JMS I .	/COMPENSATE APT TIMER FOR A READ OR WRITE
0103	6676	XAPTCO	
	4504	DATERR= JMS I .	/REPORT A DATA ERROR
0104	7000	XDATER	
	4505	LISN= JMS I .	
0105	4463	XLISN	
	4506	MESSAGE= JMS I .	
0106	4600	MESAGX	
	4507	PRNT1= JMS I .	
0107	5600	XPRNT1	
	4510	PRNT2= JMS I .	
0110	4362	XPRNT2	
	4511	PRNT4= JMS I .	
0111	7027	XPRNT4	
	4512	SPACE2= JMS I .	
0112	5264	SPACX2	
	4513	TYPE= JMS I .	
0113	4400	XTYPE	
	4514	CRLF= JMS I .	
0114	4444	XCRLF	
0115	0000	ERRPC, 0	///COMMON DATA BLOCK MUST BE LEFT IN CURRENT ORDER FOR ERROR DATA TYPEOUTS
0116	0000	DRVNUM, 0	/PC OF ERROR CALL STORED HERE
0117	0000	DATA1, 0	/DRIVE NUMBER IS STORED HERE FOR ERROR REPORT
0120	0000	DATA2, 0	/DATA FOR ERROR TYPEOUTS STORED HERE
0121	0000	DATA3, 0	
0122	0000	DATA4, 0	
0123	0000	DATA5, 0	
0124	0000	DATA6, 0	
0125	0000	DATA7, 0	
			///END COMMON BLOCK
0126	0000	LOOPPT, 0	/POINTER (SET FROM PREVIOUS SCOPE
			/STATEMENT) USED FOR LOOPING
0127	0000	ERRFLG, 0	/ERROR FLAG USED BY SCOPE AND ERROR
0130	0000	LPRQST, 0	/LOOP REQUEST FLAG FOR SUBTEST FLAG TO SCOPE

0131	0000	TEMP1, 0	/TEMPORARY STORAGE ONLY!	
0132	0000	TEMP2, 0		
0133	0000	TEMP3, 0	/TEMP3 NOT TO BE ALTERED BY ANY SUBROUTINES!!	
0134	0000	TEMP4, 0	/DITTO FOR TEMP 4	
0135	0000	TEMP5, 0	/DITTO FOR TEMP 5	
0136	0000	CYLND, 0		
0137	0000	CYLPNT, 0	/POINTS INTO CYLINDER TABLE	
0140	0000	HEAD, 0		
0141	0000	SECTOR, 0		
0142	0000	NOPRNT, 0	/FLAG TO ALLOW OUTPUT (USED W/ CNTRL-S)	
0143	0000	PASCNT, 0	/NUMBER OF PASSES OF PROGRAM	
0144	0000	LASTIN, 0	/LAST CHARACTER INPUT IS STORED HERE	
0145	3600	PRDBUF, REDBUF	/POINTER TO READ BUFFER OR 2K BUFFER	
0146	0000	OPT1, 0	/OPTION 1 AVAILABLE FLAG	
0147	0000	DEVCOD, 0	/0 IF DEVICE CODE 60,61 IS TO BE USED	
			/20 IF DEVICE CODE 62,63 IS TO BE USED	
			/WHICH SURFACE TO USE FLAG (-1 = BOTH)	
0150	7777	SURFAC, -1	/ERROR LIMIT (DRIVE WILL BE DROPPED IF REACHED)	
0151	0024	ERRLIM, 24	/LOWER SEEK LIMIT	
0152	0000	LOLIM, 0	/UPPER SEEK LIMIT	HP 001
0153	0777	HILIM, 777	/EXECUTE MANUAL INTERVENTION TEST FLAG	
0154	0000	MANINT, 0	/USE ALL CYLINDERS FLAG	
0155	0000	ALLCYL, 0	/FLAGS BAD SECTOR FILE IS VALID IF NON-0	
0156	0000	BSVALD, 0		
0157	7162	PATPTR, PATPTA		
0160	0000	PATRN, 0		
0161	0777	K0777, 0777	/	HP 001
0162	7001	M777, -777	/	HP 001
0163	0377	K377, 377		
0164	7740	M40, -40		
0165	0200	K200, 200		
0166	1015	K1015, 1015		
0167	1016	K1016, 1016		
0170	7400	M400, -400		
0171	7770	M10, -10		
0172	7730	M50, -50	/ADD THESE CONSTENTS TO MAKE ROOM IN DIAG.	HP 001
0173	7401	M377, -377		
0174	7120	M660, -660		
0175	7773	M5, -0005		
0176	1000	K1000, 1000		
0177	7000	K7000, 7000	/END OF ADDED CONSTENTS	HP 001
6007		CAF=6007	/CLEAR ALL FLAGS	
6035		KIE=6035	/AC11 TO CONSOLE INTERRUPT ENABLE FF	
6030		KCF=6030		
6032		KCC=6032	/CLEAR KEYBOARD FLAG	
7002		BSW=7002		
6045		TIE=6045		
6042		TCF=6042		
6041		TSF=6041		
6046		TLS=6046		
6031		KSF=6031		
6102		SPL=6102	/SKIP ON POWER LOW	
7421		MQL=7421		
6055		ESIA=6055		

/DISK STATUS REGISTER BIT DEFINITIONS

0100	HDSLCT=	100		/HEAD SELECT
0040	COVERO=	40		/COVER OPEN
0020	HEDOUT=	20		/HEADS OUT (OVER DISK)
0010	BRUSHH=	10		/BRUSH HOME
	/STATE DEFINITIONS			
0001	SPINUP=	1		
0002	LODHED=	2		
0003	BRUSHC=	3		
0004	SEKCNT=	4		
0005	SEKLN=	5		
0006	UNLODH=	6		
0007	SPINDN=	7		
	/WORD 2			
0200	WRDERR=	200		/WRITE DATA ERROR
0100	HEDCUR=	100		/HEAD CURRENT ERROR
0040	WRLOCK=	40		/WRITE LOCK STATUS
0020	TIMOUT=	20		/SEEK TIME OUT
0010	SPUPTO=	10		/SPIN UP TIME OUT
0004	WRGATE=	4		/WRITE GATE ERROR
0002	VOLUME=	2		/VOLUME CHECK
0001	DRSLER=	1		/DRIVE SELECT ERROR
1000	MODE8=	1000		/8 BIT MODE IN CB
2000	HEAD1=	2000		/HEAD SELECT BIT IN CA
0200	*200			
0200	6007	CAF		/CLEAR ALL FLAGS
0201	3126	DCA	LOOPPT	/FLAG THAT LOOP POINTER IS NOT VALID (IN CASE OF A RESTART)
				/DISABLE CONSOLE INTERRUPTS
0202	6035	KIE		
0203	6045	TIE		
0204	3143	DCA	PASCNT	/CLEAR PASS COUNT
0205	4443	APTCHK		/SKIP IF NOT ON APT
0206	5777	JMP	APTSET	/SKIP ALL OPENING DIALOG
0207	4506	MESSAGE		/TYPE OUT MAINDEC NUMBER
0210	0500	MANDEC		/POINTER TO TEXT (LOCATED IN FIELD 1)
0211	4447	CONSOL		/CHECK FOR CONSOLE INPUT
0212	4442	GETSR		/GET SWITCHES (HARD OR PSEUDO)
0213	0370	AND	K100	/MASK OUT ALL BUT BIT 5
0214	7640	SZA CLA		/SKIP IF NOT USING PREVIOUS RESPONSES
0215	5776	JMP	FRSTDR	/SKIP OVER OPERATOR PROMPTS
0216	1021	TAD	HCW1	/GET HARDWARE CONFIGURATION WORD 1
0217	7700	SMA CLA		/SKIP IF USING HARDWARE SWITCH REGISTER
0220	4451	CNTRLG		/LET OPERATOR ENTER SWITCH REGISTER VALUE
0221	4442	GETSR		/GET SWITCHES (HARD OR PSEUDO)
0222	0370	AND	K100	/MASK OUT ALL BUT BIT 5
0223	7640	SZA CLA		/SKIP IF NOT USING PREVIOUS RESPONSES
0224	5776	JMP	FRSTDR	/SKIP OVER OPERATOR PROMPTS
0225	1371	TAD	M4	/SET UP A COUNTER FOR FOUR DRIVES
0226	3135	DCA	TEMP5	
0227	3116	DCA	DRVNUM	/ZERO OUT DRIVE NUMBER
0230	1375	TAD	(DRVACT-1	/SET UP POINTER TO DRIVE ACTIVE TABLE
0231	3011	DCA	AUTO11	

0232	1374	TAD	(DRVOER-1	/PICK UP POINTER TO DRIVE ERROR COUNTERS
0233	3010	DCA	AUTO10	/SAVE POINTER IN AUTO INC REG
0234	3410	DRVLUP, DCA I	AUTO10	/CLEAR OUT ERROR COUNT FOR THIS DRIVE
0235	4506	MESSAGE		/"TEST DRIVE "
0236	0543	TSTDRV		
0237	1116	TAD	DRVNUM	/GET DRIVE NUMBER
0240	4507	PRNT1		/TYPE IT
0241	4506	MESSAGE		/" ? "
0242	0551	SQSP		
0243	4453	GETRES		/GET OPERATORS RESPONSE
0244	5235	JMP	.-7	/REPROMPT IF NOT "Y","N", OR "<CR>"
0245	3411	DCA I	AUTO11	/SET THE DRIVE ACTIVE FLAG BASED ON OPR'S RESPONSE
				/RETURN FROM GETRES WAS "1" IF ANSWER
				/WAS "Y", AND "0" IF ANSWER WAS "N" OR "<CR>"
0246	2116	ISZ	DRVNUM	/GO TO NEXT DRIVE (FOR TYPEOUT)
0247	2135	ISZ	TEMP5	/ASKED FOR ALL DRIVES?
0250	5234	JMP	DRVLUP	/NO--GO PROMPT FOR NEXT DRIVE
0251	4506	MESSAGE		/YES--ISSUE NEXT PROMPT
0252	0563	OPT1AV		/IS 8/A OPTION 1 AVAILABLE?
0253	4453	GETRES		/GET OPERATOR'S RESPONSE
0254	5251	JMP	.-3	/WAIT FOR VALID RESPONSE
0255	3146	DCA	OPT1	/SAVE FLAG
0256	4442	GETSR		/GET SWITCHES (HARD OR PSEUDO)
0257	0370	AND	K100	/MASK OUT ALL BUT BIT 5
0260	7640	SZA CLA		/SKIP IF NOT USING PREVIOUS RESPONSES
0261	5776	JMP	FRSTDR	/SKIP OVER OPERATOR PROMPTS
0262	4506	MESSAGE		
0263	0601	USEDEF		/DO YOU WISH TO SELECT NON-DEFAULT PARAMETERS?
0264	4453	GETRES		/GET OPERATORS RESPONSE
0265	5256	JMP	.-7	/REPROMPT IF NOT YES,NO,OR CR
0266	7650	SNA CLA		/SKIP IF ANSWER WAS YES
0267	5773	JMP	DEFSET	/GO SET UP DEFAULTS
0270	4506	MESSAGE		/USE DEVICE CODES 62,63?
0271	0631	USE62		
0272	4453	GETRES		/GET OPERATOR'S RESPONSE
0273	5270	JMP	.-3	/WAIT FOR VALID RESPONSE
0274	7106	CLL RTL		/AC=1 IF RESPONSE WAS "Y" AND 0 IF "N" OR
0275	7106	CLL RTL		/"<CR>". MOVE BIT 11 OVER TO BIT 7
0276	3147	DCA	DEVCD	/AND SAVE DEVICE CODE FLAG
0277	4506	MESSAGE		
0300	0646	EXMAIN		/EXECUTE MANUAL INTERVENTION TEST
0301	4453	GETRES		/GET RESPONSE
0302	5277	JMP	.-3	/TRY AGAIN FOR VALID RESPONSE
0303	3154	DCA	MANINT	/SAVE FLAG
0304	4506	MESSAGE		/USE ALL CYLINDERS?
0305	0673	USALCY		
0306	4453	GETRES		/GET RESPONSE
0307	5304	JMP	.-3	/INVALID RESPONSE
0310	3155	DCA	ALLCYL	/SAVE FLAG
0311	4506	LIMITS, MESSAGE		
0312	0705	USLOLM		/USE LOWER CYLINDER LIMIT?
0313	4453	GETRES		/GET RESPONSE
0314	5311	JMP	.-3	/WAIT FOR GOOD ONE
0315	7640	SZA CLA		/SKIP IF RESPONSE WAS NO
0316	4454	ENTVAL		/ALLOW OPERATOR TO ENTER VALUE

0317	3152	DCA	LOLIM	/SAVE LOWER LIMIT
0320	4506	MESSAGE		/USE UPPER CYLINDER LIMIT?
0321	0737	USUPLM		
0322	4453	GETRES		/GET OPERATOR'S RESPONSE
0323	5320	JMP	.-3	/TRY AGAIN FOR VALID RESPONSE
0324	7640	SZA CLA		/SKIP IF ANSWER WAS NO
0325	5330	JMP	+.3	
0326	1161	TAD	K0777	/GET MAXIMUM UPPER LIMIT
0327	7410	SKP		
0330	4454	ENTVAL		/ALLOW OPERATOR TO ENTER VALUE
0331	3153	DCA	HILIM	/SAVE UPPER LIMIT
0332	1153	TAD	HILIM	
0333	7041	CIA		/SUBTRACT UPPER LIMIT FROM
0334	1152	TAD	LOLIM	/LOWER LIMIT
0335	7710	SPA CLA		/SKIP IF INVALID LIMITS
0336	5342	JMP	ASKSUR	
0337	4506	MESSAGE		/LOW LIMIT MUST BE LESS THAN HIGH LIMIT
0340	0755	LLLTHL		
0341	5311	JMP	LIMITS	/GO START OVER ON LIMIT QUESTIONS
0342	4506	ASKSUR, MESSAGE		
0343	1001	US1SUR		/USE ONLY ONE SURFACE?
0344	4453	GETRES		/GET OPERATOR'S RESPONSE
0345	5342	JMP	.-3	/TRY AGAIN
0346	7640	SZA CLA		/SKIP IF "NO" (OPERATOR'S RESPONSE)
0347	5353	JMP	SURMES	/PROMPT OPERATOR FOR WHICH SURFACE
0350	7240	SIA		/PUT -1 INTO SURFAC FLAG TO FLAG USE
0351	3150	DCA	SURFAC	/OF BOTH SURFACES
0352	5772'	JMP	ERLMAS	/GO PROMPT FOR ERROR LIMIT
0353	4506	SURMES, MESSAGE		/SPECIFY SURFACE (0 OR 1)
0354	1015	SPCSUR		
0355	4505	LISN		/WAIT FOR INPUT
0356	7520	- "0		/IF "0", GO SAVE IT
0357	0365	+.6		
0360	7517	- "1		/IF "1", GO SAVE IT
0361	0364	+.3		
0362	0000	0		/IF NOT 0 OR 1, ASK AGAIN
0363	0342	ASKSUR		
0364	7001	IAC		/FLAG SURFACE 1
0365	3150	DCA	SURFAC	/SAVE SURFACE TO BE USED
0366	4514	CRLF		/TYPE <CR><LF>
0367	5772'	JMP	ERLMAS	
0370	0100	K100,	100	
0371	7774	M4,	-4	
0372	0400			
0373	0451			
0374	7136			
0375	7125			
0376	0476			
0377	0412			
	0400	PAGE		
0400	4506	ERLMAS, MESSAGE		
0401	1033	SPERLM		/SPECIFY ERROR LIMIT IN OCTAL (DEFAULT=24)

HP 001

0402	4455	GETNUM		/GET NUMBER
0403	5210	JMP	.+5	/GO SEE WHAT LAST INPUTTED CHARACTER WAS
0404	5210	JMP	.+4	
0405	1336	TAD	K24	/PICK UP DEFAULT ERROR LIMIT
0406	3151	DCA	ERRLIM	/SAVE ERROR LIMIT (ENTERED VALUE OR DEFAULT)
0407	5263	JMP	IOTCNG	/GO SET UP IOT CODES
0410	4456	WZITFG		/WAS IT CONTROL F OR G?-IF SO, HANDLE IT
0411	5200	JMP	ERLMAS	/GO ASK FOR ERROR LIMIT AGAIN
0412	7325	APTSET, CLA STL	IAC RAL	/3 INTO AC FOR MASK
0413	0022	AND	HCW2	/GET UNIT NUMBER FROM APT HARDWARE CONFIGURATION WORD 2
0414	3116	DCA	DRVNUM	
0415	1022	TAD	HCW2	/GET BITS 4 & 5 OF HARDWARE CONFIGURATION
0416	0333	AND	K300	/ WORD 2
0417	7450	SNA		/SKIP IF NOT 0
0420	5234	JMP	USEALL	/USE ALL UNITS UP TO UNIT SPECIFIED IN BITS 6-11
0421	0165	AND	K200	/CHECK IF BIT 4 IS SET
0422	7650	SNA CLA		/SKIP IF IT IS
0423	5226	JMP	.+3	/SKIP OVER ERROR CALL
0424	4446	ERROR		/HANG UP--A "RESERVED FOR FUTURE EXPANSION" BIT WAS
0425	2774	PCONLY		/USED. THIS PROGRAM NOT SET UP TO WORK IN THE "FUTURE"
0426	1377	TAD	(DRVACT-1	/SET UP A POINTER TO THE UNIT NUMBER
0427	1116	TAD	DRVNUM	/WITHIN THE DRIVE ACTIVE TABLE
0430	3010	DCA	AUTO10	/SO THAT ONLY THAT DRIVE CAN BE
0431	7201	CLA IAC		/MADE ACTIVE
0432	3410	DCA I	AUTO10	
0433	5245	JMP	APTOP1	/GO GET OPTION 1 INFORMATION
0434	1116	USEALL, TAD	DRVNUM	/PICK UP THE MAXIMUM UNIT NUMBER TO USE
0435	7040	CMA		/MAKE INTO A COUNTER
0436	3131	DCA	TEMP1	
0437	1377	TAD	(DRVACT-1	/SET UP A POINTER TO THE DRIVE ACTIVE TABLE
0440	3010	DCA	AUTO10	
0441	7201	CLA IAC		/SET THE FLAG FOR THIS DRIVE ACTIVE
0442	3410	DCA I	AUTO10	
0443	2131	ISZ	TEMP1	/REACHED HIGHEST UNIT TO BE USED YET?
0444	5241	JMP	.-3	/NO--KEEP SETTING FLAGS
0445	1021	APTOP1, TAD	HCW1	/GET APT HARDWARE CONFIGURATION WORD 1
0446	7006	RTL		/PUT OPTION 1 BIT INTO THE LINK
0447	7204	CLA RAL		/PUT LINK INTO AC
0450	3146	DCA	OPT1	/SAVE OPTION 1 AVAILABLE FLAG
0451	3155	DEFSET, DCA	ALLCYL	/SET UP ALL DEFAULT PARAMETERS
0452	3147	DCA	DEVCD	
0453	3154	DCA	MANINT	
0454	3152	DCA	LOLIM	
0455	1161	TAD	K0777	/
0456	3153	DCA	HILIM	HP 001
0457	7240	STA		
0460	3150	DCA	SURFAC	
0461	1336	TAD	K24	
0462	3151	DCA	ERRLIM	
0463	1332	IOTCNG, TAD	PIOTS	/GET POINTER TO TABLE OF IOT POINTERS
0464	3010	DCA	AUTO10	/PUT IN AUTO INDEX REGISTER
0465	1410	IOTLUP, TAD I	AUTO10	/PICK UP POINTER TO AN IOT FROM TABLE
0466	7450	SNA		/SKIP IF NOT TABLE TERMINATOR
0467	5276	JMP	FRSTDR	/GO SELECT FIRST DRIVE
0470	3131	DCA	TEMP1	/SAVE POINTER TO IOT

```

0471 1531      TAD I   TEMP1      /PICK UP IOT THRU POINTER
0472 0335      AND     K7757      /MASK OUT BIT 7 OF CODE
0473 1147      TAD     DEVCOD     /ADD IN DESIRED CODE
0474 3531      DCA I   TEMP1      /SAVE IOT WITH NEW CODE
0475 5265      JMP     IOTLUP     /CONTINUE DOWN IOT POINTER TABLE
0476 7200      FRSTDR, CLA        /
0477 3116      DCA     DRVNUM     /ZERO DRIVE NUMBER
0500 1776      TAD     M4         /SET UP A COUNTER FOR 4 DRIVES
0501 3135      DCA     TEMP5      /
0502 1377      TAD     (DRVACT-1  /SET UP A POINTER TO THE DRIVE ACTIVE TABLE
0503 3010      DCA     AUTO10     /
0504 1410      TAD I   AUTO10     /PICK UP ACTIVE FLAG FOR DRIVE IN DRVNUM
0505 7640      SZA CLA          /SKIP IF DRIVE NOT AVAILABLE FOR TESTING
0506 5322      JMP     SCOPIN     /GOT A DRIVE--GO INIT SCOPE ROUTINE
0507 2116      ISZ     DRVNUM     /INCREMENT TO NEXT DRIVE
0510 2135      ISZ     TEMP5      /SKIP IF ALL DRIVES HAVE BEEN CHECKED
0511 5304      JMP     .-5        /NO--GET FLAG FOR THIS DRIVE
0512 4443      APTCHK          /SKIP IF NOT ON APT
0513 5316      JMP     .+3        /DON'T TYPE THE MESSAGE
0514 4506      MESSAGE          /NO DRIVES AVAILABLE FOR TESTING
0515 2254      NODRVS          /
0516 4446      ERROR          /TYPE THE PC (DISSAPEAR IF ON APT)
0517 2774      PCONLY          /
0520 4451      CNTRLG          /
0521 5314      JMP     .-5        /HANG UP
0522 3127      SCOPIN, DCA ERRFLG /NO ERRORS PRIOR TO TEST1
0523 3156      DCA     BSVALD     /FLAG BAD SECTOR FILE NOT VALID
0524 4424      RLDC             /CLEAR DEVICE
0525 1334      TAD     FSTTST     /PICK UP POINTER TO FIRST TEST
0526 3126      DCA     LOOPPT     /MAKE FIRST SCOPE LOOP TO FIRST TEST
0527 7201      CLA IAC          /FLAG ENTERING THE FIRST TEST FOR FIRST TIME
0530 3130      DCA     LPRQST     / (ALSO MAKE SURE FIRST TEST DOES NOT REQUEST A LOOP)
0531 5734      JMP I   FSTTST     /GO TO FIRST TEST

      /DATA AREA FOR THIS PAGE
0532 7142      PIOTS, IOTTAB-1   /POINTER TO TABLE OF IOT POINTERS
0533 0300      K300, 300
0534 0537      FSTTST, TEST0
0535 7757      K7757, 7757
0536 0024      K24, 24

      /*****
      /TEST 0 INITIAL STATE TEST
      /
      /      RESET THE DRIVE. VERIFY DRIVE READY IS SET AND DRIVE ERROR IS
      /      NOT SET. IF DRIVE ERROR IS SET, HANDLE THE DRIVE ERROR NORMALLY
      /      (THE DRIVE MAY GET DROPPED). IF DRIVE IS NOT READY, DROP THE
      /      DRIVE.
      /
0537 4457      TEST0, RESET      /ISSUE RESET
0540 4433      RRER             /READ THE ERROR REGISTER
0541 7012      RTR              /GET DRIVE ERR INTO LINK
0542 7430      SZL              /SKIP IF DRIVE ERROR NOT ASSERTED
0543 4461      ERRHAN          /HANDLE THE DRIVE ERROR
0544 4433      RRER             /GET DRIVE READY INTO SIGN BIT

```

```

0545 7012      RTR
0546 7510      SPA                /SKIP IF DRIVE NOT READY
0547 5353      JMP      .+4
0550 4506      MESSAGE            /? DRIVE NOT READY
0551 1221      NOTRDY
0552 5775'     JMP      DROPDR    /DROP THE DRIVE
0553 4445      SCOPE

/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**
/HEADER NOT FOUND TESTS
/
/      VERIFY HEADER NOT FOUND AND OPI ARE SET WHEN ISSUING READ TO THE
/      INCORRECT CYLINDER, INCORRECT HEAD, OR ILLEGAL SECTOR.
/
/HEADER NOT FOUND TEST 1
/
/FLOAT A 1 THROUGH CYLINDER ADDRESS BITS AND VERIFY HEADERS NOT FOUND FOR READS
/AT CYLINDER 0, HEAD 0, SECTOR 0.
/

0554 1130      TAD      LPRQST    /CHECK WHY SCOPE CAME HERE
0555 7450      SNA
0556 5366      JMP      ROTHNF    /I REQUESTED A LOOP
0557 7710      SPA CLA
0560 5774'     JMP      HNFCON    /NORMAL SCOPE LOOP
0561 1165      TAD      K200      /FIRST ENTRY INTO TEST
0562 3136      DCA      CYLNDR    /SET UP INITIAL INCORRECT CYLINDER VALUE
0563 3141      DCA      SECTOR    /READS WILL BE AT SECTOR 0
0564 3140      DCA      HEAD      /HEADS WILL BE AT CYL 0, HEAD 0
0565 5773'     JMP      HNFREQ1   /GO REQUEST A LOOP
0566 1136      ROTHNF, TAD      CYLNDR /GET INCORRECT CYLINDER VALUE USED LAST TIME
0567 7110      CLL RAR          /FLOAT THE 1 OVER
0570 5772'     JMP      HNFCON

0572 0600
0573 0606
0574 0610
0575 4161
0576 0371
0577 7125
      0600      PAGE

0600 7440      HNFCON, SZA        /SKIP IF NO MORE 1'S TO FLOAT
0601 5205      JMP      .+4        /GO SAVE NEW INCORRECT CYLINDER VALUE
0602 7330      STL CLA RAR        /DON'T REQUEST A LOOP
0603 3130      DCA      LPRQST
0604 5213      JMP      ENDHN1    /EXIT TEST
0605 3136      DCA      CYLNDR    /SAVE NEW CYLINDER TO TRY TO READ AT
0606 7240      HNFREQ1, STA      /REQUEST A LOOP
0607 3130      DCA      LPRQST
0610 4462      HNFCON, SEEKTO     /SEEK TO HEAD 0, CYL 0
0611 5210      JMP      HNFCON    /SEEK FAILURE--TRY AGAIN
0612 4777'     JMS      FRCHNF    /GO FORCE HEADER NOT FOUND AND REPORT IF ERROR
0613 4445      ENDHN1, SCOPE

/HEADER NOT FOUND TEST 2

```


POSITION HEADS AT CYLINDER 777.

DO READ DATA, HEAD 1. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 2 THROUGH 18 (EVEN ONLY) UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ, SUCCESSFULLY, REPORT THAT FACTORY BAD SECTOR FILE CANNOT BE READ, INCREMENT ERROR COUNT AND PROCEED WITH READ OF SECTOR 20.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA FORMAT (WORD 0 AND 1 ARE NOT 0, WORD 2 AND 3 ARE 0, LOCATE FIRST WORD OF ALL ONE'S AND THAT WORD TO WORD 127 ARE ALL ONE'S.) STORE BAD SECTOR DATA.

READ DATA, HEAD ONE, SECTOR 20. CHECK FOR ANY ERRORS AND REPORT. IF ERROR, READ SECTOR 22 THROUGH 38 (EVEN ONLY) UNTIL NO ERROR ON READ. REPORT ALL ERRORS BUT DO NOT INCREMENT ERROR COUNT. IF NONE CAN BE READ SUCCESSFULLY, REPORT THAT SOFTWARE BAD SECTOR FILES CANNOT BE READ, INCREMENT ERROR COUNT AND EXIT TEST.

ON SECTOR WITH NO CRC ERROR, VERIFY DATA AS ABOVE. STORE BAD SECTOR DATA.

NOTE 1: SURFACE SILECT WILL BE IGNORED ON THIS TEST. THIS TEST WILL BE PERFORMED EVEN IF SURFACE 0 ONLY IS SELECTED.

NOTE 2: SWR BIT 9 WILL SUPPRESS EXECUTION OF THIS TEST. THIS BIT SHOULD ONLY BE USED WHEN IT IS KNOWN THAT THE BAD SECTOR FILES HAVE BEEN DESTROYED.

0667	4457	TEST29, RESET	/RESET THE DRIVE	
0670	4442	GETSR	/GET THE SWITCHES	
0671	0363	AND K4	/SAVE ONLY BIT 9	
0672	7640	SZA CLA	/SKIP IF NORMAL EXECUTION	
0673	5776'	JMP END29	/BSF READING SUPPRESSED--EXIT TESTS	
0674	1130	TAD LPRQST	/PICK UP THE LOOP REQUEST FLAG TO TEST THE MODE OF ENTRY	
0675	7450	SNA	/SKIP IF FIRST TIME OR AN UNREQUESTED LOOP	
0676	5302	JMP .+4	/THIS TEST REQUESTED THIS LOOP--GO ADD 4 TO PREVIOUS SECTOR	
0677	7700	SMA CLA	/SKIP IF THE SCOPE LOOPED	
0700	5304	JMP .+4	/FIRST TIME INTO TEST--START AT SECTOR 0	
0701	5305	JMP .+4	/SET UP FOR NO BAD SECTORS YET	
0702	7307	CLL CLA IAC RTL	/ADD 4 TO PREVIOUS SECTOR	
0703	1141	TAD SECTOR		
0704	3141	DCA SECTOR	/STORE SECTOR TO READ	
0705	1321	TAD PBADSE	/SET UP POINTER THAT FIRST SOFTWARE BAD SECTOR	
0706	3775'	DCA FRSTSB	/ SHOULD BE PLACED AT BEGINNING OF BAD SECTOR	
			/ AREA (BECAUSE NO MANUFACTURING BAD YET)	
0707	1361	TAD K2777	/SET UP TO SEEK TO HD 1; CYL 777	HP 001
0710	4462	SEEKTO	/POSITION HEADS OVER THE BAD SECTOR FILES	
0711	5307	JMP .-2	/TRY TO SEEK TO BAD SECTOR FILES AGAIN	
0712	4774'	JMS REDBAD	/READ A BAD SECTOR FILE (SECTOR SPECIFIED IN SECTOR)	
0713	5325	JMP EREXMA	/ERROR EXIT	
0714	4773'	JMS SERCH	/CHECK THE SERIAL NUMBER AND 0 FILLED WORDS	

0715	5325	JMP	EREXMA	/ERROR EXIT
0716	4772'	JMS	CHKBSF	/VERIFY BAD SECTORS ARE VALID AND IN ASCENDING ORDER
0717	5325	JMP	EREXMA	/ERROR EXIT
0720	4771'	JMS	BADFIL	/FILL BAD SECTOR AREA FROM BAD SECTOR FILE IN READ BUFFER
0721	0000	PBADSE, BADSEC		/POINTER TO WHERE FIRST BAD SECTOR SHOULD BE PLACED
0722	1011	TAD	AUTO11	/GET POINTER TO WHERE LAST WORD (-1 TERMINATOR) WENT
0723	3775'	DCA	FRSTSB	/PLACE POINTER TO WHERE FIRST SOFTWARE BAD SECTOR GOES
0724	5333	JMP	NORQMA	/ALL OK--DON'T REQUEST LOOP AND EXIT TEST
0725	1141	EREXMA, TAD	SECTOR	/CHECK IF SECTOR READ WAS LAST BAD SECTOR
0726	1362	TAD	M20	/ IN MANUFACTURING BAD SECTOR FILE
0727	7650	SNA CLA		/SKIP IF IT WAS NOT
0730	5333	JMP	NORQMA	/DON'T REQUEST A LOOP
0731	7240	STA		/-1 INTO AC TO REQUEST A LOOP
0732	7410	SKP		
0733	7330	NORQMA, STL CLA	RAR	/DON'T REQUEST A LOOP
0734	3130	DCA	LPRQST	/SAVE LOOP REQUEST FLAG
0735	4445	SCOPE		

/SIMILAR TEST TO READ THE SOFTWARE BAD SECTOR FILES

0736	4457	RESET		/RESET THE DRIVE
0737	1130	TAD	LPRQST	/PICK UP THE LOOP REQUEST FLAG TO TEST THE MODE OF ENTRY
0740	7450	SNA		/SKIP IF FIRST TIME OR AN UNREQUESTED LOOP
0741	5346	JMP	ADD4	/THIS TEST REQUESTED THIS LOOP--GO ADD 4 TO PREVIOUS SECTOR
0742	7710	SPA CLA		/SKIP IF FIRST TIME INTO TEST
0743	5351	JMP	CON29B	/SCOPE LOOPED--REPEAT TEST AS LAST TIME
0744	1770'	TAD	K24	/FIRST TIME INTO TEST--START AT SECTOR 20
0745	5350	JMP	.+3	/SAVE SECTOR TO READ
0746	7307	ADD4, CLL CLA	IAC RTL	/ADD 4 TO PREVIOUS SECTOR
0747	1141	TAD	SECTOR	
0750	3141	DCA	SECTOR	/STORE SECTOR TO READ
0751	1361	CON29B, TAD	K2777	/SET UP TO SEEK TO HD 1; CYL 777 HP 001
0752	4462	SEEKTO		/POSITION HEADS OVER THE BAD SECTOR FILES
0753	5351	JMP	CON29B	/TRY TO SEEK TO BAD SECTOR FILES AGAIN
0754	4774'	JMS	REDBAD	/READ A BAD SECTOR FILE (SECTOR SPECIFIED IN SECTOR)
0755	5767'	JMP	EREXSW	/ERROR EXIT
0756	4773'	JMS	SERCH	/CHECK THE SERIAL NUMBER AND 0 FILLED WORDS
0757	5767'	JMP	EREXSW	/ERROR EXIT
0760	5766'	JMP	T29CON	

0761	2777	K2777, 2777	/	HP 001
0762	7760	M20, -20		
0763	0004	K4, 4		

0766	1000	
0767	1005	
0770	0536	
0771	5336	
0772	1200	
0773	1122	
0774	1020	
0775	1003	
0776	1015	
0777	6025	
	1000	PAGE

```

1000 4777' T29CON, JMS CHKBSF /VERIFY BAD SECTORS ARE VALID AND IN ASCENDING ORDER
1001 5205 JMS EREXSW /ERROR EXIT
1002 4776' JMS BADFIL /FILL BAD SECTOR AREA FROM BAD SECTOR FILE IN READ BUFFER
1003 0000 FRSTSB, BADSEC /PREVIOUS TEST WILL OVERWRITE THIS POINTER
/TO POINT AFTER LAST MANUFACTURING BAD SECTOR
1004 5213 JMS NORQSW /ALL OK--DON'T REQUEST LOOP AND EXIT TEST
1005 1141 EREXSW, TAD SECTOR /CHECK IF SECTOR READ WAS LAST BAD SECTOR
1006 1217 TAD M44 / IN SOFTWARE BAD SECTOR FILE
1007 7650 SNA CLA /SKIP IF IT WAS NOT
1010 5213 JMS NORQSW /DON'T REQUEST A LOOP
1011 7240 STA /-1 INTO AC TO REQUEST A LOOP
1012 7410 SKP
1013 7330 NORQSW, STL CLA RAR /DON'T REQUEST A LOOP
1014 3130 DCA LPRQST /SAVE LOOP REQUEST FLAG
1015 4445 END29, SCOPE
1016 5775' JMP TEST30

1017 7734 M44, -44

/THIS SUBROUTINE READS A BAD SECTOR FILE (EVEN-ODD SECTOR PAIR) AT THE SECTOR
/SPECIFIED IN SECTOR. ERRORS ARE CHECKED FOR AND REPORTED. ON ERROR, A NORMAL
/RETURN IS MADE, ELSE THE PROGRAM SKIPS ON RETURN.
/

1020 0000 REDBAD, 0
1021 7200 CLA
1022 1234 TAD M1000 /SET UP WORD COUNT FOR 2 SECTORS
1023 4432 RLWC
1024 1145 TAD PRDBUF /POINT CONTROLLER TO READ BUFFER
1025 4426 RLMA
1026 1141 TAD SECTOR /SET UP SA FOR SECTOR TO READ FROM
1027 7002 BSW /PUT INTO POSITION
1030 4431 RLSA
1031 1774' TAD K2777 /SET UP CA FOR HEAD 1; CYL 777
1032 4427 RLCA
1033 4471 CLRRBF /CLEAR THE READ BUFFER
1034 7000 M1000, -1000 /MINUS THE NUMBER OF WORDS TO BE CLEARED
1035 1116 TAD DRVNUM /ISSUE AN 8 BIT MODE READ COMMAND
1036 7002 BSW /GET DRIVE SELECT INTO POSITION
1037 1167 TAD K1016 /8 BIT MODE; FIELD 1; READ
1040 4430 RLCB
1041 4503 APTCOM /COMPENSATE APT TIMER FOR READ TIME
1042 4425 RLSD /WAIT FOR DONE
1043 4470 JMPPM1
1044 4433 RRER /READ AND SAVE ERROR REG
1045 3121 DCA DATA3
1046 1121 TAD DATA3 /GET IT BACK
1047 7700 SMA CLA /SKIP IF CRC ERROR
1050 5261 JMP EFCHK /CHECK ERROR FLAG FOR OTHER ERRORS
1051 4441 RLSE /CHECK THAT ERROR FLAG SET
1052 7410 SKP /REPORT THAT CRC DID NOT SET FLAG
1053 5263 JMP EFCHK+2 /CONTINUE WITH NORMAL ERROR REPORT
1054 1121 TAD DATA3 /PUT ER INTO DATA1 FOR TYPEOUT
1055 3117 DCA DATA1
1056 4446 ERROR /ERROR FLAG NOT SET BY CRC ERROR

```

HP 001

```

1057 2346      CRCFLG
1060 5320      JMP      RDBDRT      /PC DRVNO ER
1061 4441      EFCHK, RLSE      /MAKE ERROR RETURN
1062 5277      JMP      RBCHSA    /SKIP IF ERROR
1063 1141      TAD      SECTOR    /OK--GO CHECK THAT SA WAS CLOCKED
1064 3124      DCA      DATA6    /SAVE SECTOR READ FOR ERROR TYPEOUT
1065 1234      TAD      M1000
1066 3125      DCA      DATA7    /SAVE WORD COUNT
1067 1774      TAD      K2777
1070 3123      DCA      DATA5    /SAVE CA
1071 4436      RRCB
1072 3122      DCA      DATA4    /SAVE CB
1073 4460      GETSTA
1074 4446      ERROR
1075 2373      EF9
                                /GET AND SAVE STATUS IN DATA1 AND DATA2
                                /ERROR FLAG SET
                                /PC DRVNO STATUS ER CB CA SA WC
                                /IF SA >= 2400 THEN CALL WAS FROM SOFTWARE
                                / BAD SECTOR FILE TEST
                                /MAKE ERROR RETURN
                                /READ THE SECTOR ADDRESS REG
                                /SAVE IT
                                /GET INTO NORMAL POSITION
                                /COMPARE IT TO THE INITIAL
                                / VALUE BEFORE THE READ
                                /IT SHOULD BE 2 MORE
                                /SKIP IF IT ISN'T
                                /MAKE A GOOD RETURN
                                /GET WHAT SA SHOULD HAVE BEEN

1076 5320      JMP      RDBDRT
1077 4437      RBCHSA, RRSA
1100 3120      DCA      DATA2
1101 1120      TAD      DATA2
1102 7002      BSW
1103 7041      CIA
1104 1141      TAD      SECTOR
1105 1321      TAD      K2
1106 7650      SNA CLA
1107 5317      JMP      RDBDRT-1
1110 1141      TAD      SECTOR
1111 1321      TAD      K2
1112 7002      BSW
1113 3117      DCA      DATA1
1114 4446      ERROR
1115 2441      SANA2
1116 5320      JMP      RDBDRT
1117 2220      ISZ      REDBAD
1120 5620      RDBDRT, JMP I REDBAD
                                /SAVE IT FOR ERROR TYPEOUT
                                /SA NOT AS EXPECTED AFTER 2 SECTOR READ
                                /PC DRVNO EXPCTD ACTUAL
                                /MAKE ERROR EXIT
                                /SKIP ON RETURN FOR GOOD EXIT
                                /RETURN

1121 0002      K2,      2

/THIS SUBROUTINE VERIFIES THAT THE FIRST EIGHT WORDS OF THE BAD SECTOR FILE
/ARE OK. THE TWO BITS IN THE SERIAL NUMBER THAT MUST BE 0 ARE CHECKED, AND
/THE ZERO WORDS ARE CHECKED. IF ERRORS, AN ERROR CALL IS MADE AND THE RETURN
/IS TO CALL+1. IF NO ERROR, THE RETURN IS TO CALL+2
/
1122 0000      SERCH, 0
1123 7240      STA
1124 1145      TAD      PRDBUF
1125 3012      DCA      AUTO12
1126 4773      JMS      GET4
1127 1120      TAD      DATA2
1130 0165      AND      K200
1131 7640      SZA CLA
1132 5337      JMP      +5
1133 1122      TAD      DATA4
1134 0165      AND      K200
                                /GET A POINTER TO THE READ BUFFER FOR
                                /USE IN AN AUTO INCREMENT REG
                                /GET FOUR WORDS FROM READ BUFFER USING AUTO12
                                /CHECK "BIT15" OF BOTH 16 BIT SERIAL
                                / NUMBER WORDS. IF THIS BIT IS NOT
                                / 0, THEN THE SERIAL NUMBER IS NOT
                                / VALID, SO REPORT ERROR

```

HP 001

```

1135 7650          SNA CLA          /SKIP IF OTHER BIT15 IS NOT 0
1136 5344          JMP      SCHCON  /SERIAL NUMBER IS OK
1137 1141          TAD      SECTOR  /SAVE SECTOR THAT WAS READ
1140 3123          DCA      DATA5
1141 4446          ERROR          /INVALID CARTRIDGE SERIAL NUMBER IN BAD SECTOR FILE
1142 2472          BADSN          /PC DRVNO WD1-SERIAL-WD2 WD3-NUMBER-WD4 SECTOR
1143 5361          JMP      SCHRET  /MAKE ERROR RETURN
1144 4773' SCHCON, JMS      GET4    /GET NEXT FOUR WORDS OF FILE
1145 1117          TAD      DATA1  /NOW CHECK THAT ALL FOUR WORDS ARE ZERO.
1146 1120          TAD      DATA2  /SINCE TOP 4 BITS ARE MASKED, ALL
1147 1121          TAD      DATA3  /4 WORDS CAN BE ADDED AND IF ANY ARE NON-0,
1150 1122          TAD      DATA4  /THE SUM WILL BE NON-ZERO.
1151 7650          SNA CLA          /SKIP IF ANY WERE NOT 0
1152 5360          JMP      SCHRET-1 /MAKE GOOD RETURN
1153 1141          TAD      SECTOR  /SAVE SECTOR FOR ERROR TYPEOUT
1154 3123          DCA      DATA5
1155 4446          ERROR          /4 WORDS (8 BIT) OF 0'S BAD IN BAD SECTOR FILE
1156 2521          BAD4Z          /PC DRVNO WD1-WD4 SECTOR
1157 5361          JMP      SCHRET  /MAKE ERROR RETURN
1160 2322          ISZ      SERCH   /SKIP ON RETURN IF NO ERROR
1161 5722          SCHRET, JMP I SERCH /RETURN

1173 6004
1174 0761
1175 1340
1176 5336
1177 1200
1200 1200          PAGE

/THIS ROUTINE CHECKS THE SECTORS LISTED IN THE BAD SECTOR FILE (POINTED TO BY AUTO12).
/IT IS VERIFIED THAT ALL BAD SECTORS ARE WITHIN RANGE AND THAT THEY ARE
/IN ASCENDING ORDER. A MAXIMUM OF 16 BAD SECTORS IS ALLOWED (EACH,
/MANUFACTURING AND SOFTWARE DETECTED). IF MORE THAN THIS NUMBER ARE FOUND,
/A MESSAGE IS PRINTED AND THE DRIVE IS DROPPED.
/

1200 0000          CHKBSF, 0
1201 7200          CLA
1202 3333          DCA      PRECYL  /SET UP SMALLEST ALLOWABLE CYLINDER
1203 3334          DCA      PRESEC  /SET UP SMALLEST ALLOWABLE TRACK AND SECTOR
1204 1336          TAD      M21     /SET UP COUNTER FOR MAXIMUM NUMBER
1205 3335          DCA      CHKCNT  / OF BAD SECTORS IN FILE
1206 4777' CHKLUP, JMS      GET4    /GET 4 WORDS FROM READ BUFFER INTO
/ DATA1 THRU DATA4
1207 7344          STA CLL RAL      /CHECK THAT ALL BUT BIT 11 ARE 0
1210 0120          AND      DATA2  /CHECK OTHER CYLINDER BITS
1211 7640          SZA CLA          /SKIP IF THEY ARE
1212 5261          JMP      ALL1CH  /CHECK IF ALL 1'S
1213 7344          STA CLL RAL      /CHECK THAT ALL BITS EXCEPT BIT0
1214 0122          AND      DATA4  / (DON'T CARE) ARE 0 IN TRACK
1215 7640          SZA CLA          /SKIP IF TRACK IS WITHIN RANGE
1216 5310          JMP      INVSEC  /REPORT INVALID SECTOR LISTING
1217 1121          TAD      DATA3  /CHECK THAT SECTOR IS WITHIN RANGE
1220 1172          TAD      M50
1221 7700          SMA CLA
1222 5310          JMP      INVSEC  /SKIP IF IT IS
/REPORT INVALID SECTOR

```

HP 001

HP 001

```

/NOV CHECK THAT SECTOR IS LISTED IN ASCENDING ORDER. NOTE THAT IF CYLO,
/HD 0, SECTOR 0 IS LISTED AS BAD, AN OUT OF SEQUENCE WILL BE REPORTED.
1223 1120      TAD      DATA2      /GET SECOND WORD OF CYL ADD.      HP 001
1224 7002      BSW                      /PUT MSB IN BIT 5      HP 001
1225 7106      CLL RTL      /NOW PUT IT IN BIT 3      HP 001
1226 1117      TAD      DATA1      /GET FIRST WORD OF CYL ADD.      HP 001
1227 3117      DCA      DATA1      /SAVE COMPLETE CYLINDER ADDRESS      HP 001
1230 1117      TAD      DATA1      /COMPARE CYLINDER TO PREVIOUS
1231 7041      CIA                      / CYLINDER LISTED IN THE BAD
1232 1333      TAD      PRECYL      / SECTOR FILE
1233 7540      SMA SZA      /SKIP IF NEW CYLINDER IS >= OLD ONE
1234 5315      JMP      OUTSEQ      /REPORT OUT OF SEQUENCE ERROR
1235 7640      SZA CLA      /SKIP IF EQUAL TO OLD ONE
1236 5246      JMP      SAVNEW      /FILE IS ASCENDING--SAVE NEW PREVIOUS VALUES
1237 1122      TAD      DATA4      /GET TRACK BIT INTO ACS TO SERVE AS
1240 7002      BSW                      / AN MSB TO SECTOR
1241 1121      TAD      DATA3      /ADD IN SECTOR
1242 7041      CIA                      /COMPARE TO PREVIOUS BAD SECTOR
1243 1334      TAD      PRESEC
1244 7700      SMA CLA      /SKIP IF NEW > PREVIOUS
1245 5315      JMP      OUTSEQ      /REPORT OUT OF SEQUENCE ERROR
1246 1117      SAVNEW, TAD      DATA1      /SAVE PREVIOUS CYLINDER
1247 3333      DCA      PRECYL
1250 1122      TAD      DATA4      /FORMULATE COMBINATION TRACK AND
1251 7002      BSW                      / SECTOR WORD AND SAVE IT
1252 1121      TAD      DATA3
1253 3334      DCA      PRESEC
1254 2335      ISZ      CHKCNT
1255 5206      JMP      CHKLUP
1256 4506      MESSAGE
1257 1231      TOOMBS
1260 5776      JMP      DROPDR      /DROP THE DRIVE
1261 1117      ALL1CH, TAD      DATA1      /CHECK IF ALL 4 8 BIT WORDS WERE
1262 1120      TAD      DATA2      / 1'S FILLED
1263 1121      TAD      DATA3
1264 1122      TAD      DATA4
1265 1337      TAD      M1774
1266 7640      SZA CLA      /(-377)*4
1267 5310      JMP      INVSEC      /SKIP IF THEY ARE
1270 1335      TAD      CHKCNT      /REPORT IT AS AN INVALID BAD SECTOR
1271 7104      CLL RAL      /FORM A COUNTER BASED ON THE NUMBER
1272 7104      CLL RAL      /OF 4 WORD BAD SECTORS THAT WERE REMAINING
1273 1174      TAD      M660      /TO BE READ AND THE NUMBER OF
1274 3335      DCA      CHKCNT      /WORDS REMAINING IN THE FILE AFTER THE
1275 6211      CH1LUP, CDF      10      /LAST ALLOWABLE BAD SECTOR
1276 1412      TAD I      AUTO12
1277 6201      CDF      0      /PICK UP WORD
1300 0163      AND      K377      /MASK GARBAGE
1301 1173      TAD      M377      /COMPARE TO 377
1302 7640      SZA CLA      /SKIP IF OK
1303 5323      JMP      NOTAL1      /REPORT NON-ALL 1 WORD
1304 2335      ISZ      CHKCNT      /CHECK ALL REMAINING WORDS??
1305 5275      JMP      CH1LUP      /NO--CHECK ANOTHER
1306 2200      ISZ      CHKBSF      /NO ERRORS--SKIP ON RETURN
1307 5332      JMP      CHKRET      /RETURN

```

```

1310 1141 INVSEC, TAD SECTOR /SAVE SECTOR READ FOR ERROR TYPEOUT
1311 3123 DCA DATA5
1312 4446 ERROR /INVALID ENTRY IN BAD SECTOR FILE
1313 2543 INVENT /PC DRV NO. CYL 0 SECTOR TRACK SA
1314 5332 JMP CHKRET
1315 7200 OUTSEQ, CLA
1316 1141 TAD SECTOR /SAVE SA FOR ERROR TYPEOUT
1317 3123 DCA DATA5
1320 4446 ERROR /BAD SECTOR ENTRY DUPLICATED OR OUT OF SEQUENCE
1321 2563 BSDOS /PC DRV NO. CYL 0-WORD SECTOR TRACK SA
1322 5332 JMP CHKRET
1323 1141 NOTAL1, TAD SECTOR
1324 3120 DCA DATA2
1325 1335 TAD CHKCNT /FORMULATE WORD NUMBER IN BAD SECTOR
1326 1176 TAD K1000 / FILE--WORD NUMBER GREATER THAN
1327 3117 DCA DATA1 / OR EQUAL TO 400 INDICATES ERROR WAS
/ IN ODD SECTOR OF BAD SECTOR FILE PAIR
1330 4446 ERROR /NON 1 FILLED WORD FOUND AFTER LAST ENTRY IN BAD SECTOR FILE
1331 2611 NONIWD /PC DRV NO. WORD NO. SA
1332 5600 CHKRET, JMP I CHKBSF /RETURN
/
1333 0000 PRECYL, 0
1334 0000 PRESEC, 0
1335 0000 CHKCNT, 0
1336 7757 M21, -21
1337 6004 M1774, -1774

*****/***/***/***/***/***/***/***/***/***/***/***/***/***/***/***/
/TEST 30 WRITE/READ DATA TEST (PART 1)
/
/ 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. CHOOSING A SINGLE SURFACE WILL LIMIT
/ TESTING TO THAT SURFACE.
/
1340 4457 TEST30, RESET /RESET THE DRIVE
1341 1130 TAD LPRQST /PICK UP MODE OF ENTRY FLAG
1342 7450 SNA
1343 5356 JMP NXTP30 /THIS IS A REQUESTED LOOP--SELECT NEXT PATTERN
1344 7710 SPA CLA /SKIP IF FIRST ENTRY INTO TEST
1345 5775' JMP REPE30 /SCOPE LOOPED--REPEAT TEST AS LAST TIME
1346 1150 TAD SURFAC /CHECK IF OK TO USE HEAD 0
1347 7740 SMA SZA CLA /SKIP IF IT IS

```


1350	5774'	JMP	HD1T30	/GO SELECT HEAD 1
1351	3141	DCA	SECTOR	/SAVE SECTOR (0)
1352	3140	DCA	HEAD	/SAVE HEAD TO USE (0)
1353	1157	TAD	PATPTR	/GET POINTER TO TABLE OF PATTERN POINTERS
1354	3160	DCA	PATTRN	/SAVE IT
1355	5775'	JMP	REPE30	/START THE TEST
1356	2160	NXTP30, ISZ	PATTRN	/INCREMENT POINTER TO NEXT ENTRY IN PATTERN POINTER TABLE
1357	1560	TAD I	PATTRN	/PICK UP POINTER FROM TABLE FOR TESTING
1360	7640	SZA CLA		/SKIP IF PATTERN POINTER TABLE TERMINATOR
1361	5775'	JMP	REPE30	/REPEAT TEST WITH NEW PATTERN
1362	1140	TAD	HEAD	/PICK UP HEAD THAT'S BEING USED
1363	7640	SZA CLA		/SKIP IF HEAD 0
1364	5370	JMP	NORQ30	/HEAD 1 WAS USED--WE ARE DONE
1365	1150	TAD	SURFAC	/CHECK IF OK TO USE HEAD 1
1366	7640	SZA CLA		/SKIP IF NOT
1367	5774'	JMP	HD1T30	/GO SELECT HEAD 1
1370	7330	NORQ30, CLA STL	RAR	/DON'T REQUEST A LOOP
1371	3130	DCA	LPROST	
1372	5773'	JMP	END30	/EXIT TEST
1373	1553			
1374	1400			
1375	1410			
1376	4161			
1377	6004			
	1400	PAGE		
1400	7332	HD1T30, CLA STL	RTR	/SET BIT FOR HEAD 1 IN CA POSITION
1401	3140	DCA	HEAD	/SAVE IT
1402	4777'	JMS	TRK1G0	/RETURN WITH FIRST GOOD CYLO, TRK1 SECTOR
1403	3141	DCA	SECTOR	/THAT IS NOT IN BAD SECTOR FILE AND SAVE IT
1404	1157	TAD	PATPTR	/SET UP TO SELECT FIRST PATTERN
1405	3160	DCA	PATTRN	
1406	7240	STA		/REQUEST A LOOP
1407	3130	DCA	LPRQST	
1410	1140	REPE30, TAD	HEAD	/SEEK TO CYL 0, DESIRED HEAD
1411	4462	SEEKTO		
1412	5210	JMP	REPE30	/FAILURE--TRY AGAIN
1413	4472	FILBUF		/FILL UP 256 WORD BUFFER FROM POINTER IN PATTRN
1414	5600	PWRTBU, WRTBUF		/POINTER TO BUFFER TO FILL
1415	1214	TAD	PWRTBU	/POINT CONTROLLER TO WRITE BUFFER
1416	4426	RLMA		
1417	1170	TAD	M400	/SET UP WORD COUNT (1 SECTOR 8 BIT MODE)
1420	4432	RLWC		
1421	1141	TAD	SECTOR	/SET UP FOR SECTOR TO WRITE TO
1422	4431	RLSA		
1423	1140	TAD	HEAD	/WRITE TO DESIRED HEAD, CYL 0
1424	4427	RLCA		
1425	1116	TAD	DRVNUM	/GET DRIVE SELECT BITS SET UP
1426	7002	BSW		
1427	1166	TAD	K1015	/ADD IN 8 BIT WRITE TO FIELD 1
1430	4430	RLCB		
1431	1116	TAD	DRVNUM	/GET DRIVE SELECT BITS SET UP
1432	7002	BSW		
1433	1166	TAD	K1015	/ADD IN 8 BIT WRITE TO FIELD 1
1434	4430	RLCB		/ISSUE COMMAND

1435	4471	CLRRBF	/CLEAR READ BUFFER
1436	7400	-400	/MINUS NUMBER OF WORDS TO CLEAR
1437	4425	RLSD	/WAIT FOR DONE FROM WRITE COMMAND
1440	4470	JMPPM1	
1441	4441	RLSE	/SKIP IF ERROR
1442	5261	JMP SACH30	/GO CHECK SA FOR CORRECTNESS
1443	4433	RRER	/SAVE ERROR REG
1444	3121	DCA DATA3	
1445	4436	RRCB	/SAVE CB
1446	3122	DCA DATA4	
1447	1140	TAD HEAD	/SAVE CA
1450	3123	DCA DATA5	
1451	1141	TAD SECTOR	/SAVE INITIAL SA
1452	3124	DCA DATA6	
1453	1170	TAD M400	/SAVE WORD COUNT
1454	3125	DCA DATA7	
1455	4460	GETSTA	/GET AND SAVE STATUS
1456	4446	ERROR	/ERROR FLAG SET
1457	2373	EF9	/PC DRVNO STATUS ER CB CA SA WC
1460	5353	JMP END30	/EXIT TEST
1461	4437	SACH30, RRSA	/READ SECTOR ADDRESS REG
1462	3120	DCA DATA2	/SAVE ACTUAL SA
1463	1120	TAD DATA2	/GET IT BACK
1464	7041	CIA	/COMPARE TO THE INITIAL VALUE
1465	1141	TAD SECTOR	/BEFORE THE WRITE
1466	1776	TAD K100	/IT SHOULD HAVE BEEN 1 GREATER
1467	7650	SNA CLA	/SKIP IF IT WAS NOT
1470	5277	JMP READ30	/ISSUE THE READ
1471	1141	TAD SECTOR	/SAVE THE EXPECTED VALUE
1472	1776	TAD K100	
1473	3117	DCA DATA1	
1474	4446	ERROR	/SA NOT AS EXPECTED AFTER 1 SECTOR WRITE
1475	2646	SANAE1	/PC DRVNO EXPCTD ACTUAL
1476	5353	JMP END30	/EXIT TEST
1477	1141	READ30, TAD SECTOR	/SET UP SA
1500	4431	RLSA	
1501	1145	TAD PRDBUF	/SET UP BUFFER IN MA
1502	4426	RLMA	
1503	1170	TAD M400	/SET UP WORD COUNT (1 SECTOR)
1504	4432	RLWC	
1505	1116	TAD DRVNUM	/SET UP DRIVE SELECT BITS
1506	7002	BSW	
1507	1167	TAD K1016	/ISSUE 8 BIT READ
1510	4430	RLCB	
1511	4503	APTCOM	/COMPENSATE APT TIMER FOR READ
1512	4425	RLSD	/WAIT FOR DONE
1513	4470	JMPPM1	
1514	4433	RRER	/READ AND SAVE ERROR REG
1515	3121	DCA DATA3	
1516	1121	TAD DATA3	
1517	7700	SMA CLA	/SKIP IF CRC ERROR SET
1520	5331	JMP EFCH30	/GO CHECK ERROR FLAG FOR OTHER ERRORS
1521	4441	RLSE	/CHECK IF CRC SET FLAG
1522	7410	SKP	/IT DID NOT--REPORT AN ERROR
1523	5333	JMP EFCH30+2	/REPORT NORMAL ERROR WITH ERROR FLAG SET

```

1524 1121      TAD      DATA3      /MOVE ERROR REG OVER FOR THIS ERROR CALL
1525 3117      DCA      DATA1
1526 4446      ERROR
1527 2346      CRCFLG      /ERROR FLAG NOT SET BY CRC ERROR
1530 5353      JMP      END30      /PC DRV NO. ER
1531 4441      EFCH30, RLSE      /EXIT TEST
1532 5347      JMP      COMP30      /SKIP IF ERROR
1533 1170      TAD      M400      /UK--GO COMPARE DATA
1534 3125      DCA      DATA7      /SAVE WC
1535 1141      TAD      SECTOR      /SAVE INITIAL SA
1536 3124      DCA      DATA6
1537 1140      TAD      HEAD      /SAVE CA
1540 3123      DCA      DATA5
1541 4436      RRCB      /SAVE CB
1542 3122      DCA      DATA4
1543 4460      GETSTA      /SAVE STATUS FROM DRIVE
1544 4446      ERROR      /ERROR FLAG SET
1545 2373      EF9
1546 5353      JMP      END30      /EXIT TEST
1547 4473      COMP30, CMPPAT      /COMPARE BUFFER TO PATTERN
1550 3600      REDBUF      /POINTER TO BUFFER
1551 5353      JMP      END30      /GOOD RETURN--EXIT TEST
1552 4504      DATERR      /REPORT DATA ERROR
1553 4445      END30, SCOPE

```

```

/*****
/TEST 31 CRC ERROR DETECTION TEST
/

```

```

/      POSITION HEADS AT CYLINDER 0, HEAD 0. (IF SURFACE 1 ONLY IS
/      SELECTED, THEN HEAD 1 WILL BE USED.)
/

```

```

/      WRITE PATTERN 0 ON SECTOR 0. ISSUE A WRITE COMMAND TO WRITE
/      PATTERN 1 ON SECTOR 0, BUT MONITOR THE WORD COUNT REGISTER AND
/      WHEN THE 17TH (DECIMAL) DATA BREAK IS SEEN (INDICATING THAT
/      THE WRITE OF THE SECTOR HAS STARTED), ISSUE A CAF TO TERMINATE
/      THE WRITE.
/

```

```

/      READ BACK SECTOR 0 AND VERIFY THAT CRC ERROR SETS.
/

```

```

/      NOTE: CYLINDER LIMITS ARE IGNORED ON THIS TEST.
/

```

```

1554 4457      TEST31, RESET      /RESET THE DRIVE
1555 1150      TAD      SURFAC      /CHECK IF OK TO USE HEAD 0
1556 7750      SPA SNA CLA      /SKIP IF NOT
1557 5365      JMP      .+6      /SAVE HEAD 0, SECTOR 0
1560 7332      CLA STL RTR      /SET HEAD SELECT BIT FOR CA USE
1561 3140      DCA      HEAD      /SAVE IT
1562 4777      JMS      TRK1GO      /GET FIRST GOOD SECTOR
1563 3141      DCA      SECTOR      /SAVE SECTOR
1564 5367      JMP      .+3
1565 3141      DCA      SECTOR      /USE SECTOR 0
1566 3140      DCA      HEAD      /USE HEAD 0
1567 1140      TAD      HEAD      /GET DESIRED HEAD AND CYLINDER
1570 4462      SEEKTO      /GO THERE
1571 5367      JMP      .-2      /TRY AGAIN

```

1572	5775'	JMP	RTRY31	
1575	1600			
1576	0370			
1577	6324			
	1600	PAGE		
1600	1157	RTRY31, TAD	PATPTR	/SEEK OK--SET UP FOR PATTERN 1
1601	3160	DCA	PATRN	
1602	4472	FILBUF		/FILL BUFFER WITH ALL 0'S
1603	5600	PWB31, WRTBUF		/POINTER TO BUFFER TO FILL
1604	1203	TAD	PWB31	/SET UP MA
1605	4426	RLMA		
1606	1366	TAD	M200	/SET UP WORD COUNT
1607	4432	RLWC		
1610	1141	TAD	SECTOR	/SET UP SA
1611	4431	RLSA		
1612	1140	TAD	HEAD	/SET UP CA
1613	4427	RLCA		
1614	4477	WRIT8		/ISSUE 8 BIT MODE WRITE
1615	5311	JMP	END31	/ERROR--EXIT TEST
1616	2160	ISZ	PATRN	/SET UP TO WRITE PATTERN 2
1617	4472	FILBUF		/FILL THE BUFFER
1620	5600	WRTBUF		
1621	4777'	JMS	VTCHK	/CHECK IF VI278
1622	7410	SKP		
1623	4776'	JMS	SCNINT	/TURN OFF INTRPT CRTC
1624	1203	TAD	PWB31	/SET UP MA
1625	4426	RLMA		
1626	4775'	JMS	WRITE1	
1627	4434	T31LUP, RRWC		/READ WORD COUNT
1630	1365	TAD	K357	/CHECK IF 17 BREAKS HAVE OCCURRED (I.E. WRITING OF THE SECTOR
1631	7510	SPA		/HAS STARTED)--SKIP IF IT HAS
1632	5227	JMP	T31LUP	/WAIT FOR WRITE TO START
1633	3131	DCA	TEMP1	/SAVE WORD COUNT
1634	6007	CAF		/STOP THE WRITE FUNCTION
1635	4777'	JMS	VTCHK	/CHECK IF VT278
1636	7410	SKP		
1637	4774'	JMS	SETINT	/TURN ON INTRPT CRTC
1640	4503	APTCOM		/COMPENSATE APT FOR WRITE TIME
1641	1131	TAD	TEMP1	/CHECK THE WORD COUNT
1642	1172	TAD	M50	/COMPARE TO ARBITRARY NUMBER THAT WILL ASSURE
1643	7700	SMA CLA		/THAT SECTOR WAS ONLY PARTIALLY WRITTEN
1644	5200	JMP	RTRY31	/CAF WAS TOO LATE--RETRY
1645	1145	TAD	PRDBUF	/SET UP MA
1646	4426	RLMA		
1647	1170	TAD	M400	/SET UP WC
1650	4432	RLWC		
1651	1141	TAD	SECTOR	/SET UP SA
1652	4431	RLSA		
1653	1140	TAD	HEAD	/SET UP CA
1654	4427	RLCA		
1655	1116	TAD	DRVNUM	/SET UP DRIVE SELECT BITS
1656	7002	BSW		
1657	1167	TAD	K1016	/ADD IN 8 BIT READ COMMAND

1660	4430	RLCB		/ISSUE IT
1661	4503	APTCOM		/COMPENSATE APT FOR READ TIME
1662	4425	RLSD		/WAIT FOR DONE
1663	4470	JMPPM1		
1664	4433	RRER		/READ AND SAVE ERROR REG
1665	3117	DCA	DATA1	
1666	1117	TAD	DATA1	/VERIFY CRC ERROR SET
1667	7510	SPA		/SKIP IF NOT
1670	5274	JMP	+.4	/GO CHECK ERROR FLAG SET
1671	4446	ERROR		/CRC ERROR COULD NOT BE FORCED
1672	2677	NOCRC		/PC DRV NO. ER
1673	5311	JMP	END31	/EXIT TEST
1674	4441	RLSE		/VERIFY ERROR FLAG SET
1675	7410	SKP		/REPORT ERROR
1676	5302	JMP	+.4	/OK--GO CHECK RLDC CLEARS CRC
1677	4446	ERROR		/ERROR FLAG NOT SET BY CRC
1700	2346	CRCFLG		/PC DRVNO. ER
1701	5311	JMP	END31	/EXIT TEST
1702	4424	RLDC		/ISSUE A DEVICE CLEAR
1703	4433	RRER		/GET ERROR REG
1704	7500	SMA		/SKIP IF CRC STILL SET
1705	5311	JMP	END31	/OK--EXIT TEST
1706	3117	DCA	DATA1	/SAVE ERROR REG
1707	4446	ERROR		/RLDC DID NOT CLEAR CRC ERROR
1710	2723	CRCDC		/PC DRVNO ER
1711	4445	END31.	SCOPE	

/**
 /TEST 32 ZERO FILL TEST (8 AND 12 BIT MODES)

/
 / POSITION HEADS AT CYLINDER 0, HEAD 0. (USE HEAD 1 IF HEAD 1
 / ONLY IS SELECTED.)
 /
 / WRITE ALL 1'S ON FIRST GOOD SECTOR. READ AND VERIFY DATA.
 /
 / WRITE 1 WORD OF 1'S ON SAME SECTOR. READ AND VERIFY DATA AND
 / ZERO FILL.
 /
 / REPEAT FOR BOTH 8 AND 12 BIT MODES. NOTE THAT THIS WILL
 / OVERWRITE CRC ERROR CAUSED BY PREVIOUS TEST.
 /

1712	4457	TEST32.	RESET	/RESET THE DRIVE
1713	1373	TAD	(PATPTA+6	/PICK UP POINTER 7TH ENTRY IN PATTERN POINTER
1714	3160	DCA	PATTRN	/ TABLE. SAVE POINTER TO ALL 1'S PATTERN
1715	4472	FILBUF		/FILL THE WRITE BUFFER WITH PAT7
1716	5600	PWRBZ8.	WRIBUF	
1717	1150	TAD	SURFAC	/PICK UP SURFACE FLAG
1720	7750	SPA	SNA CLA	/SKIP IF HAVE TO USE HEAD 1
1721	5326	JMP	+.5	/SELECT HEAD 0, SECTOR 0
1722	7332	CLA	STL RTR	/SET HEAD 1 BIT
1723	3140	DCA	HEAD	
1724	4772	JMS	TRK1GO	/GET SECTOR TO USE ON TRACK 1
1725	7410	SKP		
1726	3140	DCA	HEAD	/SELECT HEAD 0
1727	3141	DCA	SECTOR	/SAVE SECTOR TO USE

1730	1140	TAD	HEAD	/SEEK TO CYL 0, HEAD DESIRED
1731	4462	SEEKTO		
1732	5330	JMP	.-2	/TRY SEEK AGAIN ON FAILURE
1733	1140	TAD	HEAD	/WRITE COMMAND A
1734	4427	RLCA		
1735	1141	TAD	SECTOR	/SELECT SECTOR TO WRITE
1736	4431	RLSA		
1737	1316	TAD	PWRBZ8	/SET UP MA TO POINT TO BUFFER
1740	4426	RLMA		
1741	1170	TAD	M400	/SET UP WORD COUNT
1742	4432	RLWC		
1743	4477	WRIT8		/ISSUE 8 BIT MODE WRITE
1744	5771	JMP	ENDZ8	/ERROR--EXIT TEST
1745	1145	TAD	PRDBUF	/PICK UP POINTER TO READ BUFFER
1746	4426	RLMA		/SET UP MA
1747	1170	TAD	M400	/WORD COUNT TO READ SECTOR
1750	4432	RLWC		
1751	1141	TAD	SECTOR	/SET UP SECTOR
1752	4431	RLSA		
1753	4500	READ8		/READ THE DATA
1754	5771	JMP	ENDZ8	/ERROR--EXIT TEST
1755	4473	CMPPAT		/COMPARE THE DATA
1756	3600	REDBUF		/ IN THE READ BUFFER
1757	5362	JMP	+.3	/NO ERRORS
1760	4504	DATERR		/REPORT DATA ERROR
1761	5771	JMP	ENDZ8	
1762	1316	TAD	PWRBZ8	/SET UP MA AGAIN
1763	4426	RLMA		
1764	5770	JMP	T32CON	
1765	0357	K357,	357	
1766	7600	M200,	-200	
1770	2000			
1771	2071			
1772	6324			
1773	7170			
1774	7072			
1775	7114			
1776	7065			
1777	7106			
	2000	PAGE		
2000	7240	T32CON, STA		/SET UP WORD COUNT OF 1
2001	4432	RLWC		
2002	1141	TAD	SECTOR	/WRITE SAME SECTOR
2003	4431	RLSA		
2004	4477	WRIT8		/WRITE 1 WORD FROM SAME WRITE BUFFER
2005	5271	JMP	ENDZ8	/ERROR-EXIT TEST
2006	1141	TAD	SECTOR	/READ BACK SAME SECTOR
2007	4431	RLSA		
2010	1170	TAD	M400	/READ ENTIRE SECTOR
2011	4432	RLWC		
2012	1145	TAD	PRDBUF	/READ IT INTO READ BUFFER
2013	4426	RLMA		
2014	4500	READ8		/ISSUE READ

```

2015 5271      JMP      ENDZ8      /ERROR--EXIT TEST
                /COMPARE THE 1 WORD AND 255 WORDS OF ZERO FILL
2016 1377      TAD      (REDBUF-1  /SET UP POINTER TO READ BUFFER
2017 3010      DCA      AUTO10
2020 6211      CDF      10        /FIELD OF BUFFER
2021 1410      TAD I    AUTO10    /PICK UP THE FIRST WORD
2022 6201      CDF      0
2023 0163      AND      K377      /MASK GARBAGE
2024 1173      TAD      M377      /COMPARE TO WORD OF ALL 1'S
2025 7440      SZA      /SKIP IF NO ERROR
2026 5242      JMP      DATEZ8    /GO REPORT DATA ERROR
2027 1173      TAD      M377      /SET UP COUNTER FOR REMAINING ZERO FILL
2030 3131      DCA      TEMP1
2031 6211      CDF      10
2032 1410      TAD I    AUTO10    /PICK UP WORD
2033 0163      AND      K377
2034 7440      SZA      /SKIP IF OK
2035 5261      JMP      NONZ8     /REPORT NON-ZERO FILLED WORD
2036 2131      ISZ      TEMP1
2037 5232      JMP      .-5       /CONTINUE CHECKING WORDS
2040 6201      CDF      0
2041 5271      JMP      ENDZ8     /EXIT TEST
2042 1163      DATEZ8, TAD      K377 /RESTORE DATA
2043 3120      DCA      DATA2    /SAVE FOR ERROR TYPEOUT
2044 1163      TAD      K377     /SAVE EXPECTED DATA
2045 3117      DCA      DATA1
2046 1140      TAD      HEAD      /SAVE CA
2047 3122      DCA      DATA4
2050 1141      TAD      SECTUR    /SAVE SECTOR
2051 3123      DCA      DATA5
2052 1010      TAD      AUTO10    /SAVE BUFFER ADDRESS
2053 3121      DCA      DATA3
2054 4436      RRCB      /SAVE COMMAND B (MODE BIT)
2055 3124      DCA      DATA6
2056 4446      ERROR    /DATA ERROR
2057 3063      DATAE1    /PC DRVNO EXPCTD ACTUAL CA SECTOR CB
2060 5271      JMP      ENDZ8
2061 6201      NONZ8, CDF      0
2062 3117      DCA      DATA1    /SAVE THE BAD WORD
2063 1010      TAD      AUTO10    /SAVE THE BUFFER ADDRESS
2064 3120      DCA      DATA2
2065 4436      RRCB      /SAVE CB
2066 3121      DCA      DATA3
2067 4446      ERROR    /ZERO FILL ERROR
2070 3102      ZFILER    /PC DRVNO ACTUAL MA CB
2071 4445      ENDZ8, SCOPE

/12 BIT MODE TEST FOR ZERO FILL
/
2072 4457      ZERO12, RESET    /RESET THE DRIVE
2073 1150      TAD      SURFAC    /PICK UP SURFACE FLAG
2074 7750      SPA SNA CLA      /SKIP IF HAVE TO USE HEAD 1
2075 5302      JMP      .+5       /SELECT HEAD 0, SECTOR 0
2076 7332      CLA STL RTR      /SET HEAD 1 BIT
2077 3140      DCA      HEAD

```

2100	4776'	JMS	TRK1G0	/GET SECTOR TO USE ON TRACK 1
2101	7410	SKP		
2102	3140	DCA	HEAD	/SELECT HEAD 0
2103	3141	DCA	SECTOR	/SAVE SECTOR TO USE
2104	1140	TAD	HEAD	/SEEK TO CYL 0, HEAD DESIRED
2105	4462	SEEKTO		
2106	5304	JMP	.-2	/TRY SEEK AGAIN ON FAILURE
2107	1362	TAD	M253	/SET UP COUNTER TO FILL WRITE BUFFER
2110	3363	DCA	CNT32	
2111	1375	TAD	(WRTBUF-1	/SET UP POINTER TO WRITE BUFFER
2112	3010	DCA	AUTO10	
2113	6211	CDF	10	/CHANGE TO FIELD OF BUFFER
2114	7240	STA		/PLACE A WORD OF ALL 1'S INTO THE BUFFER
2115	3410	DCA I	AUTO10	
2116	2363	ISZ	CNT32	/WRITTEN BUFFER?
2117	5314	JMP	.-3	/NO
2120	6201	CDF	0	/YES
2121	1140	TAD	HEAD	/WRITE COMMAND A
2122	4427	RLCA		
2123	1141	TAD	SECTOR	/SELECT SECTOR TO WRITE
2124	4431	RLSA		
2125	1374	TAD	(WRTBUF	/SET UP MA TO POINT TO BUFFER
2126	4426	RLMA		
2127	1773'	TAD	M252	/SET UP WORD COUNT
2130	4432	RLWC		
2131	4501	WRIT12		/ISSUE 12 BIT MODE WRITE
2132	5772'	JMP	ENDZ12	/ERROR--EXIT TEST
2133	1145	TAD	PRDBUF	/PICK UP POINTER TO READ BUFFER
2134	4426	RLMA		/SET UP MA
2135	1773'	TAD	M252	/WORD COUNT TO READ SECTOR
2136	4432	RLWC		
2137	1141	TAD	SECTOR	/SET UP SECTOR
2140	4431	RLSA		
2141	4502	READ12		/READ THE DATA
2142	5772'	JMP	ENDZ12	/ERROR--EXIT TEST
2143	1773'	TAD	M252	/SET UP COUNTER FOR NUMBER OF WORDS TO COMPARE
2144	3363	DCA	CNT32	
2145	1377	TAD	(REDBUF-1	/SET UP POINTER TO THE READ BUFFER
2146	3010	DCA	AUTO10	
2147	6211	CDF	10	/SWITCH DATA FIELD TO BUFFER FIELD
2150	1410	TAD I	AUTO10	/PICK UP WORD FROM BUFFER
2151	7040	CMA		/SHOULD COMPLEMENT TO 0
2152	7440	SZA		/SKIP IF DATA OK
2153	5771'	JMP	DATEZ12	/REPORT 12 BIT MODE DATA ERROR
2154	2363	ISZ	CNT32	/CHECK ALL WORDS?
2155	5350	JMP	.-5	/NO
2156	6201	CDF	0	/YES--BACK TO THIS FIELD
2157	1374	TAD	(WRTBUF	/SET UP MA AGAIN
2160	4426	RLMA		
2161	5770'	JMP	T32ACO	
2162	7525	M253,	-253	
2163	0000	CNT32,	0	/COUNTER FOR THE BUFFERS
2170	2200			

2171 2240
 2172 2270
 2173 2736
 2174 5600
 2175 5577
 2176 6324
 2177 3577
 2200

PAGE

2200	7240	T32ACO, STA	/SET UP WORD COUNT OF 1
2201	4432	RLWC	
2202	1141	TAD SECTOR	/WRITE SAME SECTOR
2203	4431	RLSA	
2204	4501	WRIT12	/WRITE 1 WORD FROM SAME WRITE BUFFER
2205	5270	JMP ENDZ12	/ERROR--EXIT TEST
2206	1141	TAD SECTOR	/READ BACK SAME SECTOR
2207	4431	RLSA	
2210	1777	TAD M252	/READ ENTIRE SECTOR
2211	4432	RLWC	
2212	1145	TAD PRDBUF	/READ IT INTO READ BUFFER
2213	4426	RLMA	
2214	4502	READ12	/ISSUE READ
2215	5270	JMP ENDZ12	/ERROR--EXIT TEST
		/COMPARE THE 1 WORD AND 169 WORDS OF ZERO FILL	
2216	1376	TAD (REDBUF-1	/SET UP POINTER TO READ BUFFER
2217	3010	DCA AUTO10	
2220	6211	CDF 10	/FIELD OF BUFFER
2221	1410	TAD I AUTO10	/PICK UP THE FIRST WORD
2222	6201	CDF 0	
2223	7040	CMA	/SHOULD COMPLEMENT TO 0
2224	7440	SZA	/SKIP IF NO ERROR
2225	5240	JMP DATEZ12	/GO REPORT DATA ERROR
2226	1366	TAD M251	/SET UP COUNTER FOR REMAINING ZERO FILL
2227	3131	DCA TEMP1	
2230	6211	CDF 10	
2231	1410	TAD I AUTO10	/PICK UP WORD
2232	7440	SZA	/SKIP IF OK
2233	5260	JMP NONZ12	/REPORT NON-ZERO FILLED WORD
2234	2131	ISZ TEMP1	
2235	5231	JMP .-4	/CONTINUE CHECKING WORDS
2236	6201	CDF 0	
2237	5270	JMP ENDZ12	/EXIT TEST
2240	6201	DATEZ12, CDF 0	
2241	7040	CMA	/RESTORE DATA
2242	3120	DCA DATA2	/SAVE FOR ERROR TYPEOUT
2243	7240	STA	/SAVE EXPECTED DATA
2244	3117	DCA DATA1	
2245	1140	TAD HEAD	/SAVE CA
2246	3122	DCA DATA4	
2247	1141	TAD SECTOR	/SAVE SECTOR
2250	3123	DCA DATA5	
2251	1010	TAD AUTO10	/SAVE BUFFER ADDRESS
2252	3121	DCA DATA3	
2253	4436	RRCB	/SAVE COMMAND B (MODE BIT)
2254	3124	DCA DATA6	

2255	4446	ERROR	/DATA ERROR
2256	3063	DATAE1	/PC DRVNO EXPCTD ACTUAL CA SECTOR CB
2257	5270	JMP ENDZ12	
2260	6201	NONZ12, CDF 0	
2261	3117	DCA DATA1	/SAVE THE BAD WORD
2262	1010	TAD AUTO10	/SAVE THE BUFFER ADDRESS
2263	3120	DCA DATA2	
2264	4436	RRCB	/SAVE CB
2265	3121	DCA DATA3	
2266	4446	ERROR	/ZERO FILL ERROR
2267	3102	ZFILER	/PC DRVNO ACTUAL MA CB
2270	4445	ENDZ12, SCOPE	

/**
 /TEST 33 WRITE/READ TEST (PART 2)

/
 / 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 TO 0. SEEK FORWARD TO CC. READ/COMPARE ALL DATA. SEEK
 / TO 777. 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 TO 777. SEEK REVERSE TO CC. READ/COMPARE ALL DATA.
 / SEEK REVERSE TO 0. 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, THE NEXT HIGHER CYLINDER WILL BE TRIED.
 /
 / NOTE 2: IF PROGRAM MODE 2 IS USED AND 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 SIX OF THE CYLINDERS LISTED WITHIN THE DEFAULT
 / CYLINDER SET (EVERY EIGHTH ENTRY).
 /
 / NOTE 4: TESTING WILL BE DONE WITHIN THE UPPER AND LOWER
 / CYLINDER LIMITS. CHUOSING A SINGLE SURFACE WILL LIMIT
 / TESTING TO THAT SURFACE.
 /

2271	1130	TEST33, TAD	LPRQST	/CHECK MODE OF ENTRY
2272	7450	SNA		
2273	5312	JMP	NXTC33	/1 REQUESTED LOOP--SELECT NEXT CYLINDER
2274	7710	SPA CLA		
2275	5775'	JMP	CON33A	/UNREQUESTED LOOP--REPEAT TEST

```

2276 1150      TAD      SURFAC      /FIRST ENTRY--CHECK IF OK TO USE HEAD 0
2277 7740      SMA SZA CLA      /SKIP IF IT IS OK
2300 7332      CLA STL RTR      /SET HEAD BIT FOR HEAD 1
2301 3140      DCA      HEAD      /SAVE HEAD TO USE
                /INITIALIZE TO SELECT FIRST CYLINDER (FIRST ENTRY OR CHANGE HEADS)
2302 1143      INIC33, TAD      PASCNT      /CHECK THE PASS COUNTER
2303 7650      SNA CLA      /SKIP IF NOT FIRST PASS
2304 1365      TAD      M7      /SET UP POINTER TO TABLE-8
2305 1374      TAD      (CYLTAB-1      /ADD IN POINTER TO CYLINDER TABLE - 1
2306 3137      DCA      CYLPNT      /SAVE POINTER
2307 7240      STA
2310 1152      TAD      LOLIM      /INITIALIZE CYLINDER USED LAST TIME
2311 3136      DCA      CYLNDR      / (IN CASE USING ALL CYLINDERS) TO LOLIM-1
                /SELECT THE NEXT CYLINDER, EITHER FROM TABLE OR INCREMENT IF USING
                /ALL CYLINDERS
2312 1143      NXC33, TAD      PASCNT
2313 7640      SZA CLA      /SKIP IF FIRST PASS
2314 5317      JMP      .+3
2315 1773      TAD      K10      /ADD 8 TO POINTER TO GET EVERY 8TH CYLINDER FROM TABLE
2316 5323      JMP      ADDP33      /GO ADD IN POINTER
2317 1155      TAD      ALLCYL      /ON 2ND AND LATER PASSES, CHECK IF USING
2320 7640      SZA CLA      / ALL CYLINDERS. SKIP IF NOT
2321 5361      JMP      INCC33      /GO INCREMENT TO NEXT CYLINDER
2322 7001      IAC      /ADD 1 TO TABLE POINTER
2323 1137      ADDP33, TAD      CYLPNT      /ADD ENTRY OFFSET TO POINTER TO GET POINTER
2324 3137      DCA      CYLPNT      /TO NEXT ENTRY TO RETRIEVE FROM TABLE
2325 1137      TAD      CYLPNT      /BUT FIRST, COMPARE POINTER TO END OF TABLE
2326 1372      TAD      (-CYLEND
2327 7740      SMA SZA CLA      /SKIP IF POINTER STILL POINTS WITHIN TABLE
2330 5345      JMP      DNCH33      /GO CHECK IF DONE OR IF ANOTHER HEAD
                /PLUCK THE ENTRY FROM THE DEFAULT CYLINDER SET TABLE AND VERIFY THAT
                /IT IS WITHIN LIMITS. IF BELOW LOLIM, GET NEXT ENTRY FROM TABLE. IF
                /ABOVE HILIM, GO CHECK IF DONE OR ANOTHER HEAD TO DO.
2331 1537      TAD I      CYLPNT      /PICK UP CYLINDER FROM TABLE
2332 3136      DCA      CYLNDR      /SAVE IT
2333 1152      TAD      LOLIM      /VERIFY THAT CYLINDER IS ABOVE LOW LIMIT
2334 7041      CIA
2335 1136      TAD      CYLNDR
2336 7710      SPA CLA      /SKIP IF CYLINDER ABOVE LOLIM
2337 5312      JMP      NXC33      /GET NEXT CYLINDER
2340 1136      TAD      CYLNDR      /VERIFY CYLINDER IS BELOW THE HIGH LIMIT
2341 7041      CIA
2342 1153      TAD      HILIM
2343 7740      SMA SZA CLA      /SKIP IF ABOVE THE LIMIT
2344 5771      JMP      BAD33      /GO CHECK IF THIS CYLINDER HAS A BAD SECTOR FILE LISTING
2345 1140      DNCH33, TAD      HEAD      /PICK UP HEAD BEING USED
2346 7650      SNA CLA      /SKIP IF USING HEAD 1
2347 5353      JMP      .+4
2350 7330      STL CLA RAR      /NO MORE HEADS TO USE--DON'T REQUEST A LOOP
2351 3130      DCA      LPRQST
2352 5770      JMP      END33      /EXIT TEST
2353 1150      TAD      SURFAC      /CHECK IF OK TO USE HEAD 1
2354 7650      SNA CLA      /SKIP IF IT IS OK
2355 5350      JMP      .-5      /DON'T REQUEST LOOP AND EXIT
2356 7332      STL CLA RTR      /SET HEAD BIT FOR HEAD 1

```

2357	3140	DCA	HEAD	/SAVE HEAD TO BE USED	
2360	5302	JMP	INIC33	/GO INITIALIZE CYLINDER TO BE USED	
2361	2136	INCC33, ISZ	CYLNR	/INCREMENT TO NEXT CYLINDER	
2362	7000	NOP		/NOP IN CASE LOLIM=0 (WHEN USING ALL CYLINDERS	
2363	1136	TAD	CYLNR	/CHECK IF OUT OF CYLINDERS	
2364	5767	JMP	T33CON		
2365	7771	M7,	-7		
2366	7527	M251,	-251		
2367	2400				
2370	2442				
2371	2410				
2372	0464				
2373	4553				
2374	7172				
2375	2414				
2376	3577				
2377	2736				
	2400	PAGE			
2400	7041	T33CON, CIA			
2401	1153	TAD	HILIM		
2402	7710	SPA CLA		/SKIP IF LESS THAN OR EQUAL TO HIGH LIMIT	
2403	5777	JMP	DNCH33	/GO CHECK IF DONE	
2404	1136	TAD	CYLNR	/CHECK IF AT CYLINDER 777 (IN CASE HILIM = 777)	
2405	1162	TAD	M777	/	HP 001
2406	7700	SMA CLA		/SKIP IF BELOW 777	
2407	5777	JMP	DNCH33	/GO CHECK IF DONE--DON'T USE CYL 777	
		/CHECK IF THE CYLINDER THAT HAS	FINALY BEEN SELECTED IS LISTED IN BAD SECTOR FILE		
2410	4476	BADC33, CYLBAD		/SKIP IF CYLINDER IS NOT BAD	
2411	5776	JMP	INCC33	/GO TRY OUT THE NEXT HIGHER CYLINDER	
2412	7240	STA		/FINALLY!--REQUEST A LOOP	
2413	3130	DCA	LPRQST		
2414	4463	CON33A, SKOCYL		/SEEK TO 0, THEN TO CYLNR	
2415	4775	JMS	BUFS2K	/SET UP PATTERNS IN 2K BUFFER	
2416	4474	WRTTRK		/WRITE THE ENTIRE TRACK	
2417	5242	JMP	END33	/ERROR RETURN--EXIT TEST	
2420	4475	CHKTRK		/READ AND COMPARE ENTIRE TRACK	
2421	5242	JMP	END33	/ERROR RETURN--EXIT TEST	
2422	4463	SKOCYL		/SEEK TO 0, THEN TO CYLNR	
2423	4475	CHKTRK		/READ AND COMPARE TRACK AGAIN	
2424	5242	JMP	END33	/COULDN'T REREAD AFTER SEEKING TO 0 AND BACK	
2425	4464	SKHCYL		/SEEK TO 777, THEN TO CYLNR	
2426	4475	CHKTRK		/READ AND COMPARE ENTIRE TRACK AGAIN	
2427	5242	JMP	END33	/COULDN'T READ AFTER SEEKING FROM OPPOSITE DIRECTION	
		/SINCE LAST READ WAS SUCCESSFUL--	THE BUFFER IS STILL GOOD AND CAN BE USED AS IS		
2430	4474	WRTTRK		/WRITE THE ENTIRE TRACK	
2431	5242	JMP	END33	/ERROR RETURN--EXIT TEST	
2432	4475	CHKTRK		/READ AND COMPARE ENTIRE TRACK	
2433	5242	JMP	END33	/ERROR RETURN--EXIT TEST	
2434	4464	SKHCYL		/SEEK TO 777, THEN TO CYLNR	
2435	4475	CHKTRK		/READ AND COMPARE TRACK AGAIN	
2436	5242	JMP	END33	/COULDN'T REREAD AFTER SEEKING TO 777 AND BACK	
2437	4463	SKOCYL		/SEEK TO 0, THEN TO CYLNR	

```

2440 4475      CHKTRK      /READ AND COMPARE ENTIRE TRACK AGAIN
2441 5242      JMP      END33      /COULDN'T READ AFTER SEEKING FROM OPPOSITE DIRECTION
2442 4445      END33, SCOPE

```

```

/*****/
/TEST 34 WRITE LOCK DATA PROTECTION TEST
/

```

```

/      DO WRITE DATA PATTERN 0 AT CYLINDER 0, 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.
/      VERIFY DRIVE 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.
/

```

```

2443 1154      TEST34, TAD      MANINT      /EXECUTE MANUAL INTERVENTION TESTS?
2444 7650      SNA CLA
2445 5774      JMP      SCOP34      /NO--BYPASS TEST
2446 1143      TAD      PASCNT      /YES--IS THIS THE FIRST PASS?
2447 7640      SZA CLA
2450 5774      JMP      SCOP34      /NO--BYPASS TEST
2451 1150      TAD      SURFAC      /YES--OK TO USE HEAD 0?
2452 7740      SMA SZA CLA      /YES--SKIP (USE HEAD0)
2453 7332      CLA STL RTR      /NO--SET HEAD BIT
2454 3140      DCA      HEAD      /SAVE HEAD BIT
2455 1140      TAD      HEAD      /SEEK TO CYLINDER 0, HEAD
2456 4462      SEEKTO
2457 5255      JMP      .-2      /TRY AGAIN ON SEEK FAILURE
2460 1140      TAD      HEAD
2461 7650      SNA CLA      /SKIP IF USING HEAD 0 (USE SECTOR 0)
2462 4773      JMS      TRK1G0      /GET FIRST GOOD SECTOR ON TRACK 1
2463 3141      DCA      SECTOR      /SAVE SECTOR TO USE
2464 1157      TAD      PATPTR      /GET POINTER TO FIRST ENTRY OF PATTERN
2465 3160      DCA      PATRN      /      POINTER TABLE
2466 4472      FILBUF      /FILL BUFFER
2467 5600      PWRB34, WRTBUF      /THE WRITE BUFFER IS TO BE FILLED
2470 1267      TAD      PWRB34      /GET POINTER TO WRITE BUFFER
2471 4426      RLMA      /      AND WRITE IT TO MA
2472 1170      TAD      M400      /SET UP WORD COUNT FOR 1 SECTOR
2473 4432      RLWC
2474 1140      TAD      HEAD      /SET UP CA WITH HEAD AND CYLINDER (0)
2475 4427      RLCA
2476 1141      TAD      SECTOR      /SET UP SA WITH SECTOR TO WRITE
2477 4431      RLSA
2500 4477      WRIT8      /WRITE THE DATA
2501 5774      JMP      SCOP34      /ERROR--EXIT TEST
2502 1141      TAD      SECTOR      /SET UP SA TO READ BACK
2503 4431      RLSA
2504 1170      TAD      M400      /SET UP WORD COUNT
2505 4432      RLWC

```

2506	1145	TAD	PRDBUF	/POINT MA TO READ BUFFER
2507	4426	RLMA		
2510	4500	READ8		/READ BACK THE SECTOR
2511	5774'	JMP	SCOP34	/ERROR--EXIT TEST
2512	4473	CMPPAT		/COMPARE THE READ BUFFER TO THE PATTERN
2513	3600	REDBUF		
2514	5317	JMP	.+3	/OK COMPARE--CONTINUE BELOW
2515	4504	DATERR		/REPORT DATA ERROR
2516	5774'	JMP	SCOP34	/ AND EXIT TEST
2517	4506	MESSAGE		/REQUEST OPERATOR TO WRITE LOCK DRIVE
2520	1110	PLWRLK		
2521	1116	TAD	DRVNUM	/TYPE THE DRIVE NUMBER
2522	4507	PRNT1		
2523	4512	SPACE2		/PRINT 2 SPACES
2524	4466	SETTIM		/SET UP REAL TIME CLOCK
		DECIMAL		
2525	2110	-3000		/30 SECONDS
2526	7470	-200		/GROSS TIMING ESTIMATE
		OCTAL		
2527	4460	S34CHK, GETSTA		/GET DRIVE STATUS
2530	1372	TAD	(WRLOCK	/GET MASK FOR WRITE LOCK BIT
2531	0120	AND	DATA2	
2532	7640	SZA CLA		/SKIP IF WRITE LOCK NOT YET SET
2533	5341	JMP	CON34A	/SET--CONTINUE TEST
2534	4467	TIMCHK		/CHECK REAL TIME CLOCK
2535	5327	JMP	S34CHK	/KEEP CHECKING STATUS
2536	4506	MESSAGE		/"TEST ABORTED"
2537	1247	TSTABR		
2540	5774'	JMP	SCOP34	/EXIT TEST
2541	2160	CON34A, ISZ	PATTRN	/SET UP TO WRITE PATTERN 1
2542	4472	FILBUF		/FILL WRITE BUFFER
2543	5600	PWB34B, WRTBUF		
2544	1343	TAD	PWB34B	/SET UP FOR WRITE
2545	4426	RLMA		
2546	1141	TAD	SECTOR	
2547	4431	RLSA		
2550	1170	TAD	M400	
2551	4432	RLWC		
2552	1116	TAD	DRVNUM	/GET DRIVE SELECT BITS INTO POSITION
2553	7002	BSW		
2554	1166	TAD	K1015	/ADD IN WRITE FROM FIELD 1 COMMAND
2555	4430	RLCB		/ISSUE COMMAND
2556	4425	RLSD		/WAIT FOR DONE
2557	4470	JMPPM1		
2560	4441	RLSE		/ERROR SHOULD HAVE SET
2561	5771'	JMP	NOER34	/REPRORT NO ERROR FLAG
2562	4457	RESET		/RESET THE DRIVE
2563	1145	TAD	PRDBUF	/READ THE SECTOR BACK
2564	4426	RLMA		
2565	5770'	JMP	T34CON	
2570	2600			
2571	2616			
2572	0040			
2573	6324			

PAGE

1

2

/

/

/

1

2625	1130
2626	7450
2627	5256
2630	7710
2631	5272
2632	1376
2633	3160
2634	1150
2635	7740
2636	7332
2637	3140
2640	1152

TEST 35.

```
TAD      LPRQST
SNA
JMP      NXTD35
SPA  CLA
JMP      CON35A
TAD      (DATA1
DCA      PATTRN
TAD      SURFAC
SMA  SZA  CLA
CLA  STL  RTR
DCA      HEAD
TAD      LOLIM
```

2641	3136	SAVC35,	DCA	CYLNR
2642	7410		SKP	
2643	2136		ISZ	CYLNR
2644	4476		CYLBAD	
2645	5243		JMP	.-2
2646	1136		TAD	CYLNR
2647	1162		TAD	M777
2650	7700		SMA	CLA
2651	5241		JMP	SAVC35
2652	1140		TAD	HEAD
2653	1136		TAD	CYLNR
2654	3136		DCA	CYLNR
2655	5270		JMP	RQAL35
2656	7325	NXTD35,	CLA	IAC
2657	1160		TAD	STL
2660	3160		DCA	RAL
2661	1160		TAD	PATRN
2662	1375		TAD	PATRN
2663	7640		TAD	(-LAST12
2664	5270		SZA	CLA
2665	7330		JMP	RQAL35
2666	3130		CLA	STL
2667	5272		RAR	
2670	7240		DCA	LPRQST
2671	3130		JMP	CON35A
2672	1136	RQAL35,	STA	
2673	4462		DCA	LPRQST
2674	5272	CON35A,	TAD	CYLNR
2675	1160		SEEKTO	
2676	3010		JMP	CON35A
2677	1410		TAD	PATRN
2700	3133		DCA	AUTO10
2701	1410		TAD	AUTO10
2702	3134		DCA	TEMP3
2703	1410		TAD	AUTO10
2704	3135		DCA	TEMP4
2705	1353		TAD	TEMP5
2706	3131		DCA	M70
2707	1374		TAD	TEMP1
2710	3010		DCA	(REDBUF-1
2711	6211		CDF	AUTO10
2712	5315		JMP	10
2713	1135			+.3
2714	3410	FILL35,	TAD	TEMP5
2715	1133		DCA	AUTO10
2716	3410		TAD	TEMP3
2717	1134		DCA	AUTO10
2720	3410		TAD	TEMP4
2721	2131		DCA	AUTO10
2722	5313		ISZ	TEMP1
			JMP	FILL35

2723	6201	CDF	0
------	------	-----	---

/TRY NEXT CYLINDER
 /IS CYLINDER IN CYLNR OK TO USE?
 /NO--TRY NEXT HIGHER ONE
 /CHECK THAT CYLNR IS 776 OR LESS IN CASE
 / THAT LOLIM IS VERY HIGH AND BAD
 /SKIP IF CYLINDER IS BELOW BAD SECTOR FILE
 /START AT CYLINDER 0 RATHER THAN AT LOLIM
 /PUT HEAD AND CYLINDER TOGETHER

HP 001

/REQUEST A LOOP
 /ADD 3 TO PREVIOUS POINTER TO GET
 / POINTER TO NEW DATA TO USE

/COMPARE POINTER TO LAST PATTERN

/SKIP IF THIS IS LAST PATTERN
 /GO REQUEST A LOOP
 /DON'T REQUEST A LOOP

/REQUEST A LOOP

/POSITION HEADS IN CORRECT SPOT
 /SEEK TO POSITION IN AC
 /TRY AGAIN
 /PUT POINTER TO PATTERN IN AN AUTO-INC REG

/PUT THE THREE WORDS OF THE DATA PATTERN
 / INTO TEMP LOCATIONS ON PAGE 0

/SET UP A COUNTER TO FILL 170 WORD BUFFER

/SET UP POINTER TO BUFFER

/SET DATA FIELD TO BUFFER FIELD

/SET UP 3RD WORD OF PATTERN INTO BUFFER

/SET UP 1ST WORD OF PATTERN IN BUFFER

/PLACE SECOND WORD

/DONE ENTIRE BUFFER?
 /NO--GO BACK
 /NUMBER OF WORDS WRITTEN TO BUFFER WAS
 /A MULTIPLE OF 3 (TEMP3, TEMP4, THEN TEMP5)
 /PLUS TWO (TEMP3 AND TEMP4).
 /((56*3 + 2 = 170)

		/NOW WRITE THE BUFFER TO THE DISK		
2724	4431	RLSA		/STARTING AT SECTOR 0
2725	1145	TAD	PRDBUF	/POINT CONTROLLER TO READ BUFFER
2726	4426	RLMA		
2727	1336	TAD	M252	/WORD COUNT OF 170
2730	4432	RLWC		
2731	1136	TAD	CYLNDR	/SET UP CA
2732	4427	RLCA		
2733	4501	WRIT12		/ISSUE 12 BIT WRITE
2734	5773'	JMP	END35	/EXIT TEST
2735	4471	CLRRBF		/CLEAR THE READ BUFFER
2736	7526	M252, -252		/MINUS THE NUMBER OF WORDS TO CLEAR
2737	4431	RLSA		/SET UP TO READ AT SECTOR 0
2740	1145	TAD	PRDBUF	
2741	4426	RLMA		
2742	1336	TAD	M252	/WORD COUNT
2743	4432	RLWC		
2744	4502	READ12		/ISSUE THE 12 BIT READ
2745	5773'	JMP	END35	/EXIT TEST ON READ ERROR
/NOW COMPARE DATA IN BUFFER TO THE PATTERN THAT WAS USED				
2746	1353	TAD	M70	/SET UP COUNTER
2747	3131	DCA	TEMP1	
2750	1374	TAD	(REDBUF-1	/SET UP POINTER TO THE BUFFER
2751	3010	DCA	AUTO10	
2752	5772'	JMP	CHK35+3	/START OFF WITH FIRST WORD OF PATTERN
2753	7710	M70, -70		
2772	3003			
2773	3023			
2774	3577			
2775	0456			
2776	7314			
2777	5300			
	3000	PAGE		
3000	1135	CHK35, TAD	TEMP5	/COMPARE WORD IN BUFFER TO 3RD WORD OF PATTERN
3001	4777'	JMS	CMPWRD	/COMPARE THE WORD
3002	5215	JMP	DATE35	/ERROR RETURN
3003	1133	TAD	TEMP3	/COMPARE WORD IN BUFFER TO 2ND WORD
3004	4777'	JMS	CMPWRD	/ OF PATTERN
3005	5215	JMP	DATE35	/ERROR RETURN
3006	1134	TAD	TEMP4	/LIKEWISE FOR SECOND WORD OF PATTERN
3007	4777'	JMS	CMPWRD	
3010	5215	JMP	DATE35	/GO REPORT DATA ERROR
3011	4444	TICK		/GENERATE APT. TIMING
3012	2131	ISZ	TEMP1	/CHECKED ENTIRE BUFFER?
3013	5200	JMP	CHK35	/NO--GO BACK
3014	5223	JMP	END35	/ENTIRE BUFFER WAS OK--EXIT TEST
3015	1010	DATE35, TAD	AUTO10	/SAVE MA
3016	3121	DCA	DATA3	
3017	4436	RRCB		/SAVE CB
3020	3122	DCA	DATA4	
3021	4446	ERROR		/DATA ERROR
3022	3076	DATAE2		/PC DRVNO EXPCTD ACTUAL MA CB

```

3023 4447      END35,      CONSOLE      /ALLOW CONSOLE INPUT IN CASE LOOPING
3024 4442      GETSR          /GET SWITCH REGISTER
3025 0776'     AND          K20         /CHECK IF SR7 IS SET
3026 7640      SZA CLA      /SKIP IF NOT (CALL SCOPE)
3027 5775'     JMP          CON35A     /LOCK ON TEST
3030 4445      SCOPE

/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**/**
/TEST 36 READ DATA WITHOUT HEADER CHECK TEST (CONTROLLER TEST)
/
/      WRITE AN ENTIRE TRACK WITH 0'S. CHOOSE ONE SECTOR AND WRITE
/      IT WITH OTHER DATA. READ HEADERS UNTIL THE PREVIOUS HEADER IS
/      READ AND ISSUE READ DATA WITHOUT HEADER CHECK. VERIFY THAT
/      THE CORRECT SECTOR WAS READ.
/

3031 1152      TEST36, TAD      LOLIM      /SET UP CYLINDER TO USE
3032 3136      DCA      CYLNDR
3033 4476      CYLBAD      /IS THE CYLINDER LISTED IN THE BAD SECTOR FILE?
3034 7410      SKP          /YES--INCREMENT TO NEXT CYLINDER
3035 5245      JMP          CON36A     /NO--USE THIS CYLINDER
3036 2136      ISZ      CYLNDR     /TRY NEXT CYLINDER
3037 1136      TAD      CYLNDR     /IS THE CYLINDER NOW TOO HIGH?
3040 1162      TAD      M777      /
3041 7750      SPA SNA CLA      /SKIP IF IT IS
3042 5233      JMP      TEST36+2    /CYL VALUE IS OK--CHECK IF IN BSF
3043 5232      JMP      TEST36+1    /START AT CYLINDER 0, RATHER THAN AT LOW LIM
3044 4457      RESET
3045 1150      CON36A, TAD      SURFAC   /OK TO USE HEAD 0?
3046 7740      SMA SZA CLA      /SKIP IF YES
3047 7332      STL CLA RTR      /SET BIT POSITION FOR HEAD 1
3050 3140      DCA      HEAD      /SAVE HEAD TO USE
3051 1140      TAD      HEAD      /SEEK TO HEAD AND CYLINDER
3052 1136      TAD      CYLNDR
3053 4462      SEEKTO
3054 5251      JMP      .-3        /TRY AGAIN ON SEEK FAILURE
3055 4471      CLRRBF          /CLEAR THE ENTIRE 2K BUFFER
3056 4000      4000          /-2K (NUMBER OF WORDS TO CLEAR
3057 4474      WRTTRK        /WRITE THE ENTIRE TRACK WITH 0'S
3060 5774'     JMP          END36     /ERROR--EXIT TEST
3061 1373      TAD      (PATPTA+1    /PICK UP POINTER TO ENTRY IN PATTERN POINTER
3062 3160      DCA      PATTRN      / TABLE THAT POINTS TO PATTERN 2
3063 4472      FILBUF        /FILL BUFFER WITH PATTERN 2
3064 5600      PWRB36, WRTBUF      / (THE WRITE BUFFER)
3065 1264      TAD      PWRB36     /SET UP TO WRITE THAT PATTERN
3066 4426      RLMA
3067 1344      TAD      K1500      / TO ANY OLD SECTOR
3070 3141      DCA      SECTOR     /SAVE SECTOR BEING USED
3071 1141      TAD      SECTOR
3072 4431      RLSA
3073 1170      TAD      M400
3074 4432      RLWC
3075 4477      WRIT8          /WRITE THE DATA
3076 5774'     JMP          END36     /ERROR ON WRITE
3077 1310      TAD      PBUF2      /SET UP TO READ DATA INTO BUF2
3100 4426      RLMA

```

HP 001

3101	1141	TAD	SECTOR	/SAME SECTOR
3102	4431	RLSA		
3103	1170	TAD	M400	
3104	4432	RLWC		
3105	4500	READ8		/READ THE SECTOR BACK
3106	5774'	JMP	END36	/ERROR ON READ
3107	4473	CMPPAT		/COMPARE BUFFER TO PATTERN
3110	4200	PBUF2, BUF2		/POINTER TO BUFFER TO CHECK
3111	5314	JMP	.+3	/COMPARED OK
3112	4504	DATERR		/REPORT THE DATA ERROR
3113	5774'	JMP	END36	/EXIT TEST
3114	4772'	LOOP36, JMS	SCNINT	/TURN OFF TTY INTRPT
3115	4465	REDHDR		/READ A HEADER
3116	4440	RRSI		/GET SILO WORD 1
3117	0771'	AND	K77	/SAVE ONLY SECTOR ADDRESS
3120	1346	TAD	M14	/COMPARE TO HEADER PRIOR TO ONE TO BE READ
3121	7640	SZA CLA		/SKIP IF THIS IS IT
3122	5314	JMP	LOOP36	
3123	7333	CLA IAC	STL RTR	/SET UP A BAD SECTOR ADDRESS
3124	4431	RLSA		
3125	1145	TAD	PRDBUF	/POINT CONTROLLER TO READ BUFFER
3126	4426	RLMA		
3127	1170	TAD	M400	/READ ENTIRE SECTOR
3130	4432	RLWC		
3131	1140	TAD	HEAD	/SET UP CA
3132	1136	TAD	CYLNDR	
3133	4427	RLCA		
3134	1116	TAD	DRVNUM	/SET UP DRIVE SELECT BITS
3135	7002	BSW		
3136	1343	TAD	K1017	/ADD IN COMMAND 7, 8 BIT MODE, FIELD 1
3137	4430	RLCB		/ISSUE READ DATA W/O HEADER CHECK
3140	4503	APTCOM		/COMPENSATE FOR THE TIME
3141	4770'	JMS	SETINT	/TURN ON TTY INTRPT
3142	5767'	JMP	T36CON	
3143	1017	K1017,	1017	
3144	1500	K1500,	1500	
3145	0015	K15,	15	
3146	7764	M14,	-14	
3167	3200			
3170	7072			
3171	4640			
3172	7065			
3173	7163			
3174	3245			
3175	2672			
3176	4443			
3177	6756			
	3200	PAGE		
3200	4425	T36CON, RLSD		/WAIT FOR DONE
3201	4470	JMPPM1		
3202	4441	RLSE		/CHECK ERROR FLAG
3203	5221	JMP	CON36B	/OK--CONTINUE

SEEK FORWARD TO 777. SEEK REVERSE TO CC. WRITE DATA PATTERN.
 SEEK REVERSE TO 0. SEEK FORWARD TO CC-1, WRITE PATTERN. SEEK
 REVERSE TO 0, SEEK FORWARD TO CC+1, WRITE PATTERN. SEEK
 FORWARD TO 777, 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,
 THE NEXT HIGHER CYLINDER WILL BE TRIED.

NOTE 2: IF PROGRAM MODE 2 IS USED AND THE "USE ALL CYLINDERS"
 PARAMETER IS SPECIFIED AS "Y", THE TEST WILL INCLUDE
 ALL CYLINDERS (EXCEPT 0 AND 777) IN THE SELECTED
 PARAMETER SET.

NOTE 3: IN THE FIRST PASS OF THE PROGRAM, THIS TEST IS
 EXECUTED ON ONLY THREE OF THE CYLINDERS LISTED IN THE
 DEFAULT PARAMATER SET. THOSE USED WILL BE THE FIRST,
 TWENTY-FIRST, AND FORTY-FIRST ENTRIES. ON SECOND AND
 SUBSEQUENT PASSES, EVERY FOURTH CYLINDER SET ENTRY
 WILL BE TESTED.

NOTE 4: TESTING WILL BE DONE BETWEEN UPPER AND LOWER CYLINDER
 LIMITS. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING
 TO THAT SURFACE.

3246	1130	TEST37, TAD	LPROST	/CHECK MODE OF ENTRY
3247	7450		SNA	
3250	5270		JMP	NXTC37
3251	7710		SPA	CLA
3252	5776'		JMP	CON37A
3253	1150		TAD	SURFAC
3254	7740		SMA	SZA CLA
3255	7332		CLA	STL RTR
3256	3140		DCA	HEAD
3257	1375		TAD	(PATPTA+4
3260	3160		DCA	PATRN
/INITIALIZE TO SELECT FIRST CYLINDER (FIRST ENTRY OR CHANGE HEADS)				
3261	1143	INIC37, TAD	PASCNT	/CHECK THE PASS COUNTER
3262	7650		SNA	CLA
3263	1774'		TAD	N20
3264	1373		TAD	(CYLTAB-4
3265	3137		DCA	CYLPNT
3266	1152		TAD	LOLIM
3267	3136		DCA	CYLNDR
/SELECT THE NEXT CYLINDER, EITHER FROM TABLE OR INCREMENT IF USING /ALL CYLINDERS				
3270	1143	NXTC37, TAD	PASCNT	
3271	7640		SZA	CLA
3272	5275		JMP	.+3
3273	1772'		TAD	K24
3274	5301		JMP	ADDP37

3275	1155	TAD	ALLCYL	/ON 2ND AND LATER PASSES, CHECK IF USING
3276	7640	SZA	CLA	/ ALL CYLINDERS, SKIP IF NOT
3277	5340	JMP	INCC37	/GO INCREMENT TO NEXT CYLINDER
3300	7307	CLA	CLL IAC RTL	/ADD 4 TO TABLE POINTER
3301	1137	ADDP37, TAD	CYLPNT	/ADD ENTRY OFFSET TO POINTER TO GET POINTER
3302	3137	DCA	CYLPNT	/TO NEXT ENTRY TO RETRIEVE FROM TABLE
3303	1137	TAD	CYLPNT	/BUT FIRST, COMPARE POINTER TO END OF TABLE
3304	1371	TAD	(-CYLEND	
3305	7740	SMA	SZA CLA	/SKIP IF POINTER STILL POINTS WITHIN TABLE
3306	5323	JMP	DNCH37	/GO CHECK IF DONE OR IF ANOTHER HEAD
		/PLUCK THE ENTRY FROM THE DEFAULT CYLINDER SET TABLE AND VERIFY THAT		
		/IT IS WITHIN LIMITS. IF BELOW LOLIM, GET NEXT ENTRY FROM TABLE. IF		
		/ABOVE HILIM, GO CHECK IF DONE OR ANOTHER HEAD TO DO.		
3307	1537	TAD	I CYLPNT	/PICK UP CYLINDER FROM TABLE
3310	3136	DCA	CYLNR	/SAVE IT
3311	1152	TAD	LOLIM	/VERIFY THAT CYLINDER IS ABOVE LOW LIMIT
3312	7041	CIA		
3313	1136	TAD	CYLNR	
3314	7710	SPA	CLA	/SKIP IF CYLINDER ABOVE LOLIM
3315	5270	JMP	NXTC37	/GET NEXT CYLINDER
3316	1136	TAD	CYLNR	/VERIFY CYLINDER IS BELOW THE HIGH LIMIT
3317	7041	CIA		
3320	1153	TAD	HILIM	
3321	7740	SMA	SZA CLA	/SKIP IF ABOVE THE LIMIT
3322	5352	JMP	BADC37	/GO CHECK IF THIS CYLINDER HAS A BAD SECTOR FILE LISTING
3323	1140	DNCH37, TAD	HEAD	/PICK UP HEAD BEING USED
3324	7650	SNA	CLA	/SKIP IF USING HEAD 1
3325	5331	JMP	.+4	
3326	7330	STL	CLA RAR	/NO MORE HEADS TO USE--DON'T REQUEST A LOOP
3327	3130	DCA	LPRQST	
3330	5770	JMP	END37	/EXIT TEST
3331	1150	TAD	SURFAC	/CHECK IF OK TO USE HEAD 1
3332	7650	SNA	CLA	/SKIP IF IT IS OK
3333	5326	JMP	.-5	/DON'T REQUEST LOOP AND EXIT
3334	7332	STL	CLA RTR	/SET HEAD BIT FOR HEAD 1
3335	3140	DCA	HEAD	/SAVE HEAD TO BE USED
3336	5261	JMP	INIC37	/GO INITIALIZE CYLINDER TO BE USED
3337	2136	ISZ	CYLNR	/INCREMENT CYLINDER BY 2 (IF ONE HAS BEEN FOUND BAD BELOW
				/ AT BADC)
3340	2136	INCC37, ISZ	CYLNR	/INCREMENT TO NEXT CYLINDER
3341	1136	TAD	CYLNR	/CHECK IF OUT OF CYLINDERS
3342	7041	CIA		
3343	1153	TAD	HILIM	
3344	7710	SPA	CLA	/SKIP IF LESS THAN OR EQUAL TO HIGH LIMIT
3345	5323	JMP	DNCH37	/GO CHECK IF DONE
3346	1136	TAD	CYLNR	/CHECK IF AT CYLINDER 376 (IN CASE HILIM > 376)
3347	1366	TAD	M376	
3350	7700	SMA	CLA	/SKIP IF BELOW 376
3351	5323	JMP	DNCH37	/GO CHECK IF DONE--DON'T USE CYL 776 OR 777
		/CHECK IF THE CYLINDER THAT HAS		FINALLY BEEN SELECTED IS LISTED IN BAD SECTOR FILE
		/OR IF CYLINDERS ON EITHER SIDE		ARE BAD.
3352	7240	BADC37, STA		/SUBTRACT ONE FROM CYLNR
3353	1136	TAD	CYLNR	/TO CHECK IF PREVIOUS CYLINDER IS BAD
3354	3136	DCA	CYLNR	
3355	4476	CYLBAD		/SKIP IF CYLNR IS OK

```

3356 5337      JMP      INCC37-1      /INCREMENT CYLINDER BY TWO AND TRY AGAIN
3357 2136      ISZ       CYLNR       /RESTORE CYLNR TO CENTER SELECTED CYLINDER
3360 4476      CYLBAD      /SKIP IF IT IS OK
3361 5337      JMP      INCC37-1      /INCREMENT CYLINDER BY TWO AND TRY AGAIN
3362 2136      ISZ       CYLNR       /TEST IF THE CYLINDER ON HIGHER SIDE IS OK
3363 4476      CYLBAD      /SKIP IF IT IS OK
3364 5337      JMP      INCC37-1      /INCREMENT CYLINDER BY TWO AND TRY AGAIN
3365 5767      JMP      T37CON
3366 7402      M376,      -376

3367 3400
3370 3504
3371 0464
3372 0536
3373 7167
3374 0762
3375 7166
3376 3405
3377 3145
3400 3400      PAGE

3400 7240      T37CON, STA      /RESTOR CYLNR TO SELECTED ONE
3401 1136      TAD       CYLNR
3402 3136      DCA       CYLNR
3403 7240      STA
3404 3130      DCA       LPRQST      /FINALLY!--REQUEST A LOOP
3405 4463      CON37A, SKOCYL      /SEEK TO 0 AND THEN TO CYLNR
3406 4777      JMS       FIL2KB     /FILL 2K BUFFER WITH PATTERN
3407 4474      WRTTRK      /WRITE THE ENTIRE TRACK
3410 5304      JMP      END37      /ERROR--EXIT TEST
3411 4776      JMS       CKTRK2     /CHECK ENTIRE TRACK
3412 5215      JMP      .+3        /NO ERROR--CONTINUE
3413 4504      DATERR      /REPORT DATA ERROR
3414 5304      JMP      END37      /EXIT TEST
3415 7240      STA
3416 1136      TAD       CYLNR      /NOW POINT CYLNR TO ADJACENT CYLINDER LOWER THAN CENTER CYLINDER
3417 3136      DCA       CYLNR
3420 4464      SKHCYL
3421 4474      WRTTRK
3422 5277      JMP      A1E37      /SEEK TO 777 AND THEN TO CYLNR
3423 2136      ISZ       CYLNR      /WRITE THAT ENTIRE TRACK
3424 2136      ISZ       CYLNR      /EXIT TEST BUT ADD 1 BACK TO CYLNR FIRST
3425 4464      SKHCYL      /ADD 2 TO CYLNR TO GET CC+1
3426 4474      WRTTRK
3427 5301      JMP      S1E37      /SEEK TO 777 THEN TO CYLNR
3430 7240      STA
3431 1136      TAD       CYLNR      /WRITE THE ENTIRE TRACK
3432 3136      DCA       CYLNR      /EXIT TEST BUT SUBTRACT 1 FROM CYLNR FIRST
3433 4463      SKOCYL      /RESTORE CYLNR TO CC
3434 4776      JMS       CKTRK2     /SEEK TO 0 THEN BACK TO CC
3435 7410      SKP
3436 5267      JMP      ERR37      /COMPARE DATA
3437 4464      SKHCYL      /DATA OK
                          /REPORT THE ERROR
                          /SINCE CKTRK2 MADE GOOD RETURN, THE DATA IN THE BUFFER CAN BE USED AS IS.
                          /SEEK TO 777 THEN TO CC

```

3440	4474	WRTRK		/WRITE THE ENTIRE TRACK
3441	5304	JMP	END37	/ERROR--EXIT TEST
3442	4776	JMS	CKTRK2	/CHECK THE ENTIRE TRACK
3443	5246	JMP	.+3	/DATA OK
3444	4504	DATERR		/REPORT THE DATA ERROR
3445	5304	JMP	END37	/EXIT TEST
3446	7240	STA		/SUBTRACT 1 FROM CYLNDR
3447	1136	TAD	CYLNDR	
3450	3136	DCA	CYLNDR	
3451	4463	SKOCYL		/SEEK TO 0 THEN TO CC-1
3452	4474	WRTRK		/WRITE THAT TRACK
3453	5277	JMP	A1E37	/EXIT TEST BUT ADD 1 TO CYLNDR FIRST
3454	2136	ISZ	CYLNDR	/ADD 2 TO CYLNDR TO GET TO CC+1
3455	2136	ISZ	CYLNDR	
3456	4463	SKOCYL		/SEEK TO 0 THEN TO CC+1
3457	4474	WRTRK		/WRITE THAT TRACK
3460	5301	JMP	S1E37	/RESTORE CYLNDR AND EXIT TEST
3461	7240	STA		/SUBTRACT 1 FROM CYLNDR TO GET CC
3462	1136	TAD	CYLNDR	
3463	3136	DCA	CYLNDR	
3464	4464	SKHCYL		/SEEK TO 777 THEN TO CC
3465	4776	JMS	CKTRK2	/CHECK THE TRACK
3466	5304	JMP	END37	/OK--EXIT TEST
3467	1140	TAD	HEAD	/SAVE CA FOR ERROR TYPEOUT
3470	1136	TAD	CYLNDR	
3471	3117	DCA	DATA1	
3472	1141	TAD	SECTOR	/SAVE SECTOR WHERE ERROR OCCURRED
3473	3120	DCA	DATA2	
3474	4446	ERROR		/ADJACENT CYLINDER INTERFERENCE
3475	3251	ADJINT		/PC DRVNO CA SECTOR
3476	5304	JMP	END37	
3477	2136	A1E37, ISZ	CYLNDR	/ADD 1 TO CYLNDR TO RESTORE IT
3500	5304	JMP	END37	/AND EXIT TEST
3501	7240	S1E37, STA		/SUBTRACT 1 FROM CYLNDR TO RESTORE IT
3502	1136	TAD	CYLNDR	
3503	3136	DCA	CYLNDR	
3504	4445	END37, SCOPE		

/***/
 /TEST 38 OVERWRITE TEST

/
 / CC IS CURRENT CYLINDER SELECTED FROM SET
 / SELECTED CYLINDER SET DEFINED IN PARAGRAPH 4.3.
 / PATTERN A = PAT5
 / PATTERN B = PAT1
 /
 / SEEK FORWARD TO CC. WRITE DATA OF PATTERN A IN ALL SECTORS.
 / HEAD 0. READ/COMPARE DATA.
 /
 / SEEK FORWARD TO 777, SEEK REVERSE TO CC. WRITE PATTERN B.
 / SEEK REVERSE 0, SEEK FORWARD TO CC, READ/COMPARE DATA.
 /
 / SEEK FORWARD TO 777, SEEK REVERSE TO CC. WRITE DATA PATTERN
 / A. READ/COMPARE DATA. SEEK REVERSE TO 0, SEEK FORWARD TO CC.
 / WRITE PATTERN B. SEEK FORWARD TO 777, SEEK REVERSE TO CC.


```

3505 1130 TEST38, TAD LPRQST /CHECK MODE OF ENTRY
3506 7450 SNA
3507 5326 JMP NXTC38 /I REQUESTED LOOP--SELECT NEXT CYLINDER
3510 7710 SPA CLA
3511 5775' JMP CON38A /UNREQUESTED LOOP--REPEAT TEST
3512 1150 TAD SURFAC /FIRST ENTRY--CHECK IF OK TO USE HEAD 0
3513 7740 SNA SZA CLA /SKIP IF IT IS OK
3514 7332 CLA STL RTR /SET HEAD BIT FOR HEAD 1
3515 3140 DCA HEAD /SAVE HEAD TO USE

/INITIALIZE TO SELECT FIRST CYLINDER (FIRST ENTRY OR CHANGE HEADS)
3516 1143 /INIC38, TAD PASCNT /CHECK THE PASS COUNTER
3517 7650 SNA CLA /SKIP IF NOT FIRST PASS
3520 1774' TAD M20 /SET UP POINTER TO TABLE-20 (DECIMAL)
3521 1373 TAD (CYLTAB-4 /ADD IN POINTER TO CYLINDER TABLE - 4
3522 3137 DCA CYLPNT /SAVE POINTER
3523 7240 STA
3524 1152 TAD LOLIM /INITIALIZE CYLINDER USED LAST TIME
3525 3136 DCA CYLNDR / (IN CASE USING ALL CYLINDERS) TO LOLIM-1

/SELECT THE NEXT CYLINDER, EITHER FROM TABLE OR INCREMENT IF USING
/ALL CYLINDERS
3526 1143 NXTC38, TAD PASCNT
3527 7640 SZA CLA /SKIP IF FIRST PASS
3530 5333 JMP .+3
3531 1772' TAD K24 /ADD 20 TO POINTER TO GET EVERY 20TH CYLINDER FROM TABLE
3532 5337 JMP ADDP38 /GO ADD IN POINTER
3533 1155 TAD ALLCYL /ON 2ND AND LATER PASSES, CHECK IF USING
3534 7640 SZA CLA / ALL CYLINDERS. SKIP IF NOT
3535 5771' JMP INCC38 /GO INCREMENT TO NEXT CYLINDER
3536 7307 CLA CLL IAC RTL /ADD 4 TO TABLE POINTER

```

3537	1137	ADDP38, TAD	CYLPNT	/ADD ENTRY OFFSET TO POINTER TO GET POINTER
3540	3137	DCA	CYLPNT	/TO NEXT ENTRY TO RETRIEVE FROM TABLE
3541	1137	TAD	CYLPNT	/BUT FIRST, COMPARE POINTER TO END OF TABLE
3542	1370	TAD	(-CYLEND	
3543	7740	SMA SZA	CLA	/SKIP IF POINTER STILL POINTS WITHIN TABLE
3544	5767	JMP	DNCH38	/GO CHECK IF DONE OR IF ANOTHER HEAD
		/PLUCK THE ENTRY FROM THE DEFAULT CYLINDER SET TABLE AND VERIFY THAT		
		/IT IS WITHIN LIMITS. IF BELOW LOLIM, GET NEXT ENTRY FROM TABLE. IF		
		/ABOVE HILIM, GO CHECK IF DONE OR ANOTHER HEAD TO DO.		
3545	1537	TAD I	CYLPNT	/PICK UP CYLINDER FROM TABLE
3546	3136	DCA	CYLNDR	/SAVE IT
3547	1152	TAD	LOLIM	/VERIFY THAT CYLINDER IS ABOVE LOW LIMIT
3550	7041	CIA		
3551	1136	TAD	CYLNDR	
3552	7710	SPA	CLA	/SKIP IF CYLINDER ABOVE LOLIM
3553	5326	JMP	NXTC38	/GET NEXT CYLINDER
3554	1136	TAD	CYLNDR	/VERIFY CYLINDER IS BELOW THE HIGH LIMIT
3555	7041	CIA		
3556	1153	TAD	HILIM	
3557	7740	SMA SZA	CLA	/SKIP IF ABOVE THE LIMIT
3560	5766	JMP	BADC38	/GO CHECK IF THIS CYLINDER HAS A BAD SECTOR FILE LISTING
3561	5767	JMP	DNCH38	
3566	3627			
3567	3600			
3570	0464			
3571	3614			
3572	0536			
3573	7167			
3574	0762			
3575	3633			
3576	6531			
3577	6657			
	3600			
		PAGE		
3600	1140	DNCH38, TAD	HEAD	/PICK UP HEAD BEING USED
3601	7650	SNA	CLA	/SKIP IF USING HEAD 1
3602	5206	JMP	.+4	
3603	7330	STL	CLA RAR	/NO MORE HEADS TO USE--DON'T REQUEST A LOOP
3604	3130	DCA	LPRQST	
3605	5307	JMP	END38	/EXIT TEST
3606	1150	TAD	SURFAC	/CHECK IF OK TO USE HEAD 1
3607	7650	SNA	CLA	/SKIP IF IT IS OK
3610	5203	JMP	.-5	/DON'T REQUEST LOOP AND EXIT
3611	7332	STL	CLA RTR	/SET HEAD BIT FOR HEAD 1
3612	3140	DCA	HEAD	/SAVE HEAD TO BE USED
3613	5777	JMP	INIC38	/GO INITIALIZE CYLINDER TO BE USED
3614	2136	INCC38, ISZ	CYLNDR	/INCREMENT TO NEXT CYLINDER
3615	7000	NOP		/IN CASE USING ALL CYLS & LOLIM=0
3616	1136	TAD	CYLNDR	/CHECK IF OUT OF CYLINDERS
3617	7041	CIA		
3620	1153	TAD	HILIM	
3621	7710	SPA	CLA	/SKIP IF LESS THAN OR EQUAL TO HIGH LIMIT
3622	5200	JMP	DNCH38	/GO CHECK IF DONE
3623	1136	TAD	CYLNDR	/CHECK IF AT CYLINDER 777 (IN CASE HILIM = 777)

3624	1162	TAD	M777	/	
3625	7700	SMA	CLA	/SKIP IF BELOW 777	HP 001
3626	5200	JMP	DNCH38	/GO CHECK IF DONE--DON'T USE CYL 777	
3627	4476	/CHECK IF THE CYLINDER THAT HAS			FINALLY BEEN SELECTED IS LISTED IN BAD SECTOR FILE
3630	5214	BADC38,	CYLBAD	/SKIP IF CYLNR IS OK	
3631	7240	JMP	INCC38	/INCREMENT CYLINDER AND TRY AGAIN	
3632	3130	STA		/FINALLY!--REQUEST A LOOP	
3633	4463	DCA	LPRQST		
3634	1376	CON38A,	SKOCYL	/SEEK TO 0 AND THEN TO CYLNR	
3635	3160	TAD	(PATPTA+4	/PICK UP ENTRY POINTING TO PAT5	
3636	4775'	DCA	PATRN	/SAVE FOR USE BY SUBROUTINES	
3637	4474	JMS	FIL2KB	/FILL 2K BUFFER WITH PATTERN	
3640	5307	WRTRK		/WRITE THE ENTIRE TRACK	
3641	4774'	JMP	END38	/ERROR--EXIT TEST	
3642	5245	JMS	CKTRK2	/CHECK ENTIRE TRACK	
3643	4504	JMP	+.3	/NO ERROR--CONTINUE	
3644	5307	DATERR		/REPORT DATA ERROR	
3645	4464	JMP	END38	/EXIT TEST	
3646	4471	SKHCYL		/SEEK TO 777 AND THEN TO CYLNR	
3647	4000	CLRRBF		/CLEAR OUT THE 2K BUFFER	
3650	4474	4000			
3651	5307	WRTRK		/WRITE THAT ENTIRE TRACK	
3652	4463	JMP	END38	/EXIT TEST	
3653	1157	SKOCYL		/SEEK TO 0 THEN BACK TO CC	
3654	3160	TAD	PATPTR	/SHOULD READ BACK PATTERN 1	
3655	4774'	DCA	PATRN		
3656	7410	JMS	CKTRK2	/COMPARE DATA	
3657	5300	SKP		/DATA OK	
3660	1376	JMP	ERR38	/REPORT THE ERROR	
3661	3160	TAD	(PATPTA+4	/SET UP FOR WRITING PAT5 AGAIN	
3662	4775'	DCA	PATRN		
3663	4464	JMS	FIL2KB	/FILL THE BUFFER WITH THE PATTERN	
3664	4474	SKHCYL		/SEEK TO 387 THEN TO CC	
3665	5307	WRTRK		/WRITE THE ENTIRE TRACK	
3666	4774'	JMP	END38	/ERROR--EXIT TEST	
3667	5272	JMS	CKTRK2	/CHECK THE ENTIRE TRACK	
3670	4504	JMP	+.3	/DATA OK	
3671	5307	DATERR		/REPORT THE DATA ERROR	
3672	4463	JMP	END38	/EXIT TEST	
3673	4474	SKOCYL		/SEEK TO 0 THEN TO CC	
3674	5307	WRTRK		/WRITE THAT TRACK	
3675	4464	JMP	END38	/EXIT TEST	
3676	4774'	SKHCYL		/SEEK TO 387 THEN TO CC	
3677	5307	JMS	CKTRK2	/CHECK THE TRACK	
3700	1140	JMP	END38	/OK--EXIT TEST	
3701	1136	ERR38,	TAD	/SAVE CA FOR ERROR TYPEOUT	
3702	3117	TAD	HEAD		
3703	1141	TAD	CYLNR		
3704	3120	DCA	DATA1		
3705	4446	TAD	SECTOR	/SAVE SECTOR WHERE ERROR OCCURRED	
3706	3276	DCA	DATA2		
3707	4445	ERROR		/OVERWRITE ERROR	
		OVRWRT		/PC DRVNO CA SECTOR	
		END38,	SCOPE		

/THIS ROUTINE SELECTS THE NEXT DRIVE THAT IS AVAILABLE FOR TESTING (AS

/INDICATED BY FLAGS IN DRIVE ACTIVE TABLE). CONTROL IS THEN PASSED BACK
 /TO THE SCOPE INITIALIZATION ROUTINE WHICH RESTARTS AT TEST 1. IF NO MORE
 /DRIVES ARE AVAILABLE, CONTROL IS PASSED TO THE END OF PASS ROUTINE.
 /

3710	7200	NXTDRV, CLA		
3711	2116	ISZ	DRVNUM	/BUMP UP TO NEXT DRIVE
3712	7307	CLA CLL	IAC RTL	/4 INTO AC FOR MASK
3713	0116	AND	DRVNUM	/CHECK IF AT DRIVE "4"
3714	7640	SZA CLA		/SKIP IF NOT
3715	5325	JMP	EOP	/NO MORE DRIVES--REPORT END OF PASS
3716	1116	TAD	DRVNUM	/FORM OFFSET INTO DRIVE ACTIVE TABLE
3717	1373	TAD	(DRVACT	/TO TEST IF THIS DRIVE
3720	3131	DCA	TEMP1	/IS ACTIVE
3721	1531	TAD I	TEMP1	/PICK UP DRIVE ACTIVE FLAG
3722	7650	SNA CLA		/SKIP IF DRIVE IS AVAILABLE FOR TESTING
3723	5310	JMP	NXTDRV	/TRY THE NEXT DRIVE
3724	5772'	JMP	SCOPIN	/RESART TESTS WITH THIS DRIVE ACTIVE

/END OF PASS ROUTINE. CONTROL IS PASSED HERE FROM NEXT DRIVE ROUTINE
 /WHEN NO MORE DRIVES ARE LEFT TO TEST ON THIS PASS. THE END PASS MESSAGE
 /IS TYPED AND CONTROL IS PASSED TO THE SELECT FIRST DRIVE ROUTINE.
 /SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE:

/	SR3	=0	LOOP PROGRAM
/		=1	HALT AT END OF PASS
/	SR6	=0	REPORT END OF PASS
/		=1	INHIBIT EOP MESSAGE
/			

3725	2143	EOP, ISZ	PASCNT	/INCREMENT PASS COUNT
3726	5331	JMP	.+3	
3727	7240	STA		/FREEZE PASS COUNT AT
3730	3143	DCA	PASCNT	/ 7777 AFTER 4097 PASSES
3731	4443	APTCHK		/SKIP IF NOT ON APT
3732	5771'	JMP	FRSTD	/DON'T TYPE END PASS MESSAGE OR HALT--GO SELECT FIRST DRIVE
3733	4442	GETSR		/GET SWITCH REGISTER
3734	7002	BSW		/GET SR6 INTO SIGN BIT
3735	7510	SPA		/SKIP IF BIT IS CLEAR
3736	5344	JMP	.+6	/INHIBIT TYPEOUT
3737	4506	MESSAGE		/TYPE "END PASS "
3740	1062	EOPMES		
3741	1143	TAD	PASCNT	/GET PASS COUNT
3742	4511	PRNT4		/TYPE PASS NUMBER
3743	4514	CRLF		/CARRIAGE RETURN, LINE FEED
3744	4442	GETSR		/GET SWITCHES
3745	0351	AND	K400	/MASK ALL BUT SWITCH 3
3746	7640	SZA CLA		/SKIP IF NO HALT DESIRED
3747	4451	CNTRLG		/"HALT"--GO TO CONTROL-G ROUTINE
3750	5771'	JMP	FRSTD	/GO SELECT FIRST DRIVE

3751	0400	K400, 400
------	------	-----------

3771	0476
3772	0522
3773	7126
3774	6531
3775	6657

3776 7166
3777 3516
4000

PAGE
/AJRLI SUBROUTINES AND UTILITIES

/SCOPE

/

/THIS ROUTINE IS CALLED AT THE END OF EACH SUBTEST. APT TIMING
/IS GENERATED AND CONSOLE INPUT IS PROCESSED. A DECISION IS THEN
/MADE WHETHER TO LOOP ON THE SUBTEST OR CONTINUE AT THE NEXT ONE.
/SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE (IN ORDER OF PRIORITY):
/ SR2 =1 LOOP ON CURRENT TEST
/ SR1 =1 LOOP ON ERROR (HARD OR SOFT)
/

/THE PROGRAM MAY REQUEST A LOOP (E.G. FOR A READ/WRITE TEST TO TWO DIFFERENT
/CYLINDERS) BY SETTING LPRQST TO -1. LPRQST MAY BE SET TO ANYTHING ELSE TO NOT
/REQUEST A LOOP. UPON ENTRY TO THE TEST, LPRQST WILL BE 0 IF ENTRY IS BECAUSE
/OF REQUESTED LOOP, UNCHANGED IF LOOPING BECAUSE OF A NORMAL SCOPE DECISION
/((I.E. THE PROGRAM MUST REPEAT THE TEST AS PREVIOUSLY EXECUTED), AND 1
/IF ENTERING THE TEST FOR THE FIRST TIME.
/

/CALL TICK;

/IF CONSOLE-INPUT THEN CALL CONSOL;

/IF SR2 = 1

/ THEN RETURN POINTER = LOOPPT;

/ ELSE IF (ERRFLG = 1) AND (SR1 = 1)

/ THEN RETURN POINTER = LOOPPT;

/ ELSE IF LOOP REQUEST FLAG = -1

/ THEN DO;

/ CLEAR LOOPREQUEST FLAG;
/ RETURN POINTER = LOOPPT

/ END;

/ ELSE DO;

/ LOOP REQUEST FLAG = 1;
/ LOOPPT = RETURN POINTER;
/ ERRFLG = 0;

/ END;

/RETURN VIA RETURN POINTER;

/

4000	0000	XSCOPE, 0	
4001	4444	TICK	/GENERATE APT TIMING
4002	6031	KSF	/CONSOLE INPUT?
4003	5206	JMP .+3	/NO--SKIP
4004	4777	JMS VT278	
4005	4447	CONSOL	/YES--PROCESS IT
4006	4442	GETSR	/GET SWITCHES (HARD OR PSEUDO)
4007	7006	RTL	/GET SR1 INTO LINK, SR2 INTO SIGN BIT
4010	7710	SPA CLA	/SKIP IF SR2 = 0
4011	5220	JMP RETSET	/LOOP ON TEST--GO SET RETURN POINTER
4012	1127	TAD ERRFLG	/GET ERROR FLAG
4013	7020	CML	/REVERSE STATE OF SR1 (SAVED IN LINK)
4014	7660	SZA SNL CLA	/SKIP IF NO ERROR OCCURRED OR NOT LOOPING ON ERROR
4015	5220	JMP RETSET	/LOOP ON ERROR--SET RETURN POINTER
4016	2130	ISZ LPRQST	/CHECK IF LOOP REQUEST FLAG IS SET
4017	5223	JMP LUPSET	/NOT SET--NO LOOPING

4020	1126	RETSET, TAD	LOOPPT	/PICK UP LOOP POINTER
4021	3200	DCA	XSCOPE	/PUT IT IN RETURN POINTER
4022	5230	JMP	.+6	/CLEAR DEVICE AND RETURN
4023	1200	LUPSET, TAD	XSCOPE	/PICK UP RETURN POINTER
4024	3126	DCA	LOOPPT	/SAVE IT AT LOOP POINTER
4025	3127	DCA	ERRFLG	/CLEAR ERROR FLAG
4026	7201	CLA IAC		/1 INTO AC TO FLAG ENTERING NEW TEST
4027	3130	DCA	LPRQST	/SAVE FLAG
4030	4424	RLDC		/CLEAR THE DEVICE
4031	5600	JMP I	XSCOPE	/RETURN VIA RETURN POINTER

```

/ERROR
/
/THIS ROUTINE IS CALLED BY A SUBTEST WHEN AN ERROR HAS BEEN DETECTED.
/THE PC OF THE ERROR CALL IS STORED AT "ERRPC" FOR TYPE-OUT. IF ON APT,
/NOTIFY APT OF THE ERROR. THE ARGUMENT IN THE LOCATION FOLLOWING THE ERROR
/CALL IS A POINTER TO AN ENTRY IN THE ERROR TABLE. SEE FIELD 1 FOR ERROR TABLE FORMAT.
/
/SWITCH REGISTER BITS AFFECTING THIS ROUTINE ARE:
/
/   SR0      =0      HALT ON ERROR
/   SR4      =1      INHIBIT ERROR HALT
/   SR4      =0      TIMEOUT ERROR
/   SR8      =1      INHIBIT ERROR REPORT
/   SR8      =0      DON'T DROP DRIVE REGARDLESS OF ERROR COUNT
/   SR8      =1      DROP DRIVE ON REACHING ERROR LIMIT
/
/GET ERROR PC;
/IF ON-APT THEN GO TO UVPPROM;
/IF SR4=0
/  THEN DO;
/    GET POINTER TO ERROR TABLE ENTRY;
/    GET ENTRY FROM ERROR TABLE;
/    IF ENTRY NOT= 0 THEN TYPE MESSAGE;
/    UPDATE POINTER TO TABLE;
/    GET ENTRY FROM ERROR TABLE;
/    IF ENTRY NOT= 0 THEN TYPE MESSAGE;
/    UPDATE POINTER TO TABLE;
/    GET ENTRY FROM ERROR TABLE;
/    SET POINTER TO DATA AREA (ERRPC-1);
/    DO I=(ENTRY FROM TABLE) TO 0;
/      INC POINTER TO DATA AREA;
/      GET DATUM;
/      TYPE IT (PRNT4);
/    END;
/  END;
/IF APTCODE NOT= 0
/  THEN DO;
/    INCREMENT ERROR COUNT FOR THIS DRIVE;
/    IF SR8 = 1 AND ERROR COUNT >= ERRLLIM
/      THEN DO;
/        DROP CURRENT DRIVE;
/        GO SELECT NEXT DRIVE;
/      END;
/  END;
/IF SR0=0 THEN HALT;
/SET ERRFLG;

```

```

/RETURN;
/
4032 0000      XERROR, 0
4033 7240      CLA CMA
4034 1232      TAD      XERROR
4035 3115      DCA      ERRPC
4036 1632      TAD I    XERROR
4037 3337      DCA      TABENT
4040 6211      CDF      10
4041 1737      TAD I    TABENT
4042 6201      CDF      0
4043 3356      DCA      PERNUM
4044 4442      GETSR
4045 0165      AND      K200
4046 7640      SZA CLA
4047 5267      JMP      DRPCHK
4050 2337      ISZ      TABENT
4051 4343      JMS      ERRTP
4052 4343      JMS      ERRTP
4053 6211      CDF      10
4054 1737      TAD I    TABENT
4055 6201      CDF      0
4056 3340      DCA      DATCNT
4057 1341      TAD      DATTAB
4060 3010      DCA      AUTO10
4061 1410      DATLUP, TAD I    AUTO10
4062 4511      PRNT4
4063 4512      SPACE2
4064 2340      ISZ      DATCNT
4065 5261      JMP      DATLUP
4066 4514      CRLF
4067 1356      DRPCHK, TAD      PERNUM
4070 7650      SNA CLA
4071 5315      JMP      HLTCHK
4072 1116      TAD      DRVNUM
4073 1357      TAD      PDRVOE
4074 3356      DCA      PERNUM
4075 2756      ISZ I    PERNUM
4076 5301      JMP      .+3
4077 7240      STA
4100 3756      DCA I    PERNUM
4101 4442      GETSR
4102 0360      AND      K10A
4103 7650      SNA CLA
4104 5315      JMP      HLTCHK
4105 1151      TAD      ERR LIM
4106 7141      CLL CIA
4107 1756      TAD I    PERNUM
4110 7620      SNL CLA
4111 5315      JMP      HLTCHK
4112 4506      MESSAGE
4113 0525      LIMEXC
4114 5361      JMP      DROPDR
4115 4443      HLTCHK, APTCHK
4116 5326      JMP      APTERR

/SUBTRACT 1 FROM RETURN
/ADDRESS TO GET ERROR PC
/SAVE ERROR PC
/GET ARGUMENT
/SAVE POINTER TO TABLE ENTRY

/GET THE APT ERROR CODE

/SAVE IT FOR LATER TESTING
/GET (HARD OR PSEUDO) SWITCH REG
/MASK ALL BUT SR4
/SKIP IF OK TO TYPE
/INHIBIT ERROR REPORT
/SKIP OVER APT CODE INFO
/TYPE MESSAGE FROM TABLE ENTRY
/TYPE DATA HEADER FROM TABLE ENTRY
/DATA FIELD OF ERROR TABLE
/GET NUMBER OF DATA ITEMS
/BACK TO HERE
/SAVE COUNT
/PICK UP POINTER TO DATA TABLE
/PUT IN AUTO-INC REGISTER
/PICK UP DATUM
/TYPE DATA
/FORMAT TO NEXT TAB STOP
/TYPED ALL NUMBERS?
/NO--GET NEXT DATUM
/YES--CARRIAGE RETURN
/GET THE APT CODE
/SKIP IF IT WAS A DRIVE RELATED ERROR
/DON'T TALLY AS A DRIVE ERROR;GO CHECK FOR HALT
/USE DRIVE NUMBER AS AN INDEX OFFSET
/ADD IN POINTER TO FIRST ENTRY
/SAVE THE POINTER TO CORRECT ENTRY FOR THIS DRIVE
/INCREMENT ERROR COUNT FOR THIS DRIVE

/FREEZE ERROR COUNT AT 7777

/GET SWITCHES
/CHECK DRIVE DROP SWITCH
/SKIP IF DRIVE DROPPING IS REQUESTED
/DON'T PERFORM ANY CHECKS ON ERROR COUNT
/PICK UP THE ERROR LIMIT

/COMPARE TO ERROR COUNT FOR CURRENT DRIVE
/SKIP IF ERROR LIMIT REACHED OR EXCEEDED (LINK TOGGLED)
/NOT REACHED--GO CHECK FOR HALT ON ERROR
/? ERROR LIMIT EXCEEDED -

/GO DROP THE DRIVE
/SKIP IF NOT ON APT
/GO TELL APT

```

4117	4442	GETSR		/GET SWITCHES
4120	7700	SMA CLA		/SR0=0
4121	4451	CNTRLG		/YES--"HALT" (GO TO CONTROL-G ROUTINE)
4122	7001	IAC		/SET AC NON-ZERO
4123	3127	DCA	ERRFLG	/SET ERROR FLAG
4124	2232	ISZ	XERROR	/SKIP OVER ARGUMENT
4125	5632	JMP I	XERROR	/AND RETURN W/AC CLEAR
/				
4126	7300	APTERR, CLA CLL		
4127	7450	SNA		/SKIP IF APT CODE-UNIT NUMBER INFORMATION
4130	5334	JMP	..+4	/IS TO BE TRANSMITTED
4131	1116	TAD	DRVNUM	/ONLY TELL APT THE ERROR PC
4132	6271	COF	70	/ADD IN THE UNIT NUMBER TO THE ERROR CODE
4133	7410	SKP		/FLAG THAT CODE-UNIT NUMBER IS BEING SENT
4134	1115	TAD	ERRPC	
4135	6272	CIF	70	/SEND ERROR PC TO PAPT
4136	5742	JMP I	PAPTER	/INSTRUCTION FIELD OF UVPRM
/GO TELL APT				
/				
4137	0000	TABENT, 0		/USED AS POINTER TO ERROR TABLE ENTRY
4140	0000	DATCNT, 0		/USED TO COUNT NUMBER OF DATA ITEMS TYPED
4141	0114	DATTAB, ERRPC-1		/POINTER TO DATA ENTRIES
4142	6520	PAPTER, 6520		/POINTER TO APT'S ERROR ROUTINE
/				
4143	0000	ERRTP, 0		
4144	6211	COF	10	/FIELD OF ERROR TABLE
4145	1737	TAD I	TABENT	/PICK UP TABLE ENTRY
4146	6201	COF	0	/BACK TO HERE
4147	7450	SNA		/SKIP IF VALID POINTER
4150	5354	JMP	..+4	/NULL ENTRY, IGNORE IT
4151	3353	DCA	..+2	/PLACE POINTER AS MESSAGE ARGUMENT
4152	4506	MESSAGE		/TYPE MESSAGE VIA POINTER FROM ERROR TABLE
4153	0000	0		/POINTER GETS PLACED HERE
4154	2337	ISZ	TABENT	/POINT TO NEXT TABLE ENTRY
4155	5743	JMP I	ERRTP	/RETURN TO W/AC CLEAR
/				
4156	0000	PERNUM, 0		/USED AS A POINTER INTO TABLE OF ERROR COUNTERS BELOW
4157	7137	PDRVOE, DRVOER		/POINTER TO FIRST ENTRY INTO TABLE
4160	0010	K10A, 10		
/DROP DRIVE BY CLEARING THE BIT SET IN LOCATION "DRIVES"				
4161	4506	DROPDR, MESSAGE		
4162	0553	DROPNG		/DROPPING DRIVE
4163	1116	TAD	DRVNUM	/TYPE THE DRIVE NUMBER
4164	4507	PRNT1		
4165	4514	CRLF		
4166	1116	TAD	DRVNUM	/SET UP A POINTER
4167	1376	TAD	(DRVACT	/ INTO DRIVE ACTIVE TABLE FOR THIS DRIVE
4170	3131	DCA	TEMP1	
4171	3531	DCA I	TEMP1	/SET THIS DRIVE NON-ACTIVE
4172	5775	JMP	NXTDRV	/GO SELECT NEXT DRIVE
4175	3710			
4176	7126			
4177	6000			

4200 PAGE

```

/CONSOL
/
/THIS ROUTINE IS USED TO PROCESS KEYBOARD INPUT. IF OUTPUT IS DISABLED, THEN
/ONLY CONTROL-Q AND CONTROL-C ARE RECOGNIZED, IF OUTPUT IS ENABLED, THEN
/IN ADDITION, CONTROL-G AND CONTROL-S ARE RECOGNIZED. THESE CHARACTERS
/HAVE THE FOLLOWING CONTROL FEATURES:
/
/   CNTRL-G -- ENTER SWITCH REGISTER MODIFICATION ROUTINE
/   CNTRL-C -- RETURN TO OS/8 MONITOR
/   CNTRL-S -- INHIBIT ALL FURTHER OUTPUT. ONLY CNTRL-Q OR CNTRL-C
/               WILL BE RECOGNIZED. (IF THE PROGRAM ATTEMPTS TO
/               OUTPUT, IT WILL HANG WAITING FOR A CNTRL-Q OR C.)
/   CNTRL-Q -- ENABLE OUTPUT
/   CNTRL-F -- ADJUST FILLER CHARACTER COUNT
/ALL OTHER CHARACTERS ARE ECHOED, FOLLOWED BY "?" AND CRLF.
/
/IF NOT ON-APT OR NO INPUT
/THEN IF OUTPUT-OFF
/THEN DO;
/   GET CHARACTER;
/   IF CHARACTER = ^Q
/   THEN TURN OUTPUT ON;
/   ELSE IF CHARACTER = ^C
/   THEN DO;
/       TURN OUTPUT ON;
/       TYPE "^C";
/       GOTO OS/8;
/   END;
/END;
/ELSE DO;
/   GET CHARACTER;
/   IF CHARACTER = ^G
/   THEN DO;
/       TYPE "^G";
/       CALL CNTRLG;
/   END;
/   ELSE IF CHARACTER = ^Q
/   THEN;
/   ELSE IF CHARACTER = ^C
/   THEN DO;
/       TYPE "^C";
/       GOTO OS/8;
/   END;
/   ELSE IF CHARACTER = ^S
/   THEN TURN OUTPUT OFF;
/   ELSE IF CHARACTER = ^F
/   THEN CALL CNTRLF;
/   ELSE TYPE "?<CR><LF>";
/END;
/RETURN;
/
4200 0000 XCONSO, 0
4201 6031 KSF
4202 5243 JMP CONRET
/SKIP IF KEYBOARD INPUT
/NO INPUT--RETURN

```

```

4203 4777' JMS VT278
4204 4443 APTCHK
4205 5243 JMP CONRET
4206 1142 TAD NOPRNT
4207 7650 SNA CLA
4210 5216 JMP ONLISN
4211 4505 LISN
4212 7557 -221
4213 4242 CNTRLQ
4214 0000 0
4215 4243 CONRET
4216 4505 ONLISN, LISN
4217 7571 M207, -207
4220 4236 CTLG
4221 7555 -223
4222 4240 CNTRLS
4223 7557 -221
4224 4243 CONRET
4225 7572 -206
4226 4234 CILF
4227 0000 0
4230 4231 .+1
4231 4506 MESSAGE
4232 1060 QESMRK
4233 5243 JMP CONRET
4234 4452 CTLF, CNTRLF
4235 5243 JMP CONRET
4236 4450 CTLG, UPARG
4237 5243 JMP CONRET
4240 2142 CNTRLS, ISZ NOPRNT
4241 5243 JMP CONRET
4242 3142 CNTRLQ, DCA NOPRNT
4243 5600 CONRET, JMP I XCONSO
/

/CNTRLF
/
/CONTROL-F HANDLER. ALLOWS MODIFICATION OF FILL COUNT FROM CONSOLE.
/
4244 0000 XCNTRLF, 0
4245 4506 MESSAGE
4246 1074 FILLEQ
4247 1023 TAD FILLER
4250 4511 PRNT4
4251 4455 GETNUM
4252 5256 JMP .+4
4253 5256 JMP .+3
4254 5274 JMP CNTRFT
4255 5273 JMP CNTRFT-1
4256 7240 STA
4257 4505 LISN
4260 7572 -206
4261 4245 XCNTRLF+1
4262 7571 -207
4263 4271 .+6

/SKIP IF NOT ON APT
/ON APT--IGNORE KEYBOARD
/PICK UP OUTPUT DISABLE FLAG
/OUTPUT DISABLED?
/NO--GO TO ONLISN
/YES--GET CHARACTER FROM KEYBOARD
/CNTRL-Q
/IF ^Q GO TO CNTRLQ
/IF NOT ^Q OR ^C THEN
/ RETURN
/GET CHARACTER
/CNTRL-G

/CNTRL-S
/GO DISABLE OUTPUT
/CNTRL-Q
/OUTPUT IS ALREADY ON, SO RETURN
/IF CONTROL-F GO TO
/ CTLF
/IF INVALID CHARACTER THEN
/ TYPE OUT
/ "?" FOLLOWED BY
/ <CR><LF>
/ AND RETURN
/CALL CNTRL-F HANDLER
/RETURN

/RETURN
/SET NOPRINT FLAG
/RETURN
/CLEAR NOPRINT FLAG
/RETURN W/AC CLEAR

/TYPE ^F<CR>FILL = "
/GET FILL COUNT
/PRINT OCTAL FILL COUNT
/GET OCTAL INPUT FROM CONSOLE
/GO SEE WHAT INVLAID INPUT WAS

/RETURN WITH FILL UNCHANGED
/MODIFY FILL AND RETURN
/USE PREVIOUS INPUT
/CHECK LAST INPUT
/IF CNTRL-F, THEN
/ REISSUE PROMPT
/IF CONTROL G, THEN GO HANDLE IT

```

```

4264 0000      0      /UNMATCHED INPUT
4265 4266      .+1
4266 4506      MESSAGE /TYPE "?<CR>"
4267 1060      QESMRK
4270 5245      JMP      XCTRLF+1 /GO START OVER
4271 4450      UPARG    /HANDLE CONTROL G
4272 5245      JMP      XCTRLF+1
4273 3023      DCA      FILLER /SAVE NEW FILL COUNT
4274 5644      CNTFRT, JMP I XCTRLF /RETURN W/ AC CLEAR

      /UPARG
      /
      /TYPE "^G", THEN CALL CONTROL G HANDLER
      /
4275 0000      XUPARG, 0
4276 4506      MESSAGE
4277 1072      UPARRG
4300 4451      CNTRLG
4301 5675      JMP I      XUPARG

      /CNTRLG
      /
      /CONTROL-G HANDLER. CALLED WITH "CNTRLG" WHENEVER CONTROL-G IS
      /RECEIVED FROM CONSOLE, OR ON PROGRAM "HALT", OR AT BEGINNING OF PROGRAM
      /IF SOFTWARE SWITCHES USED.
      /
4302 4506      MESSAGE /THESE THREE INSTRUCTIONS ARE FOR REISSUEING
4303 1072      UPARRG /THE "^G" PROMPT WHEN A CONTROL-G IS TYPED
4304 7410      SKP      /WHEN ALREADY IN CONTROL-G ROUTINE
4305 0000      XCTRLG, 0
4306 4506      MESSAGE /TYPE "SR="
4307 1102      SWRMSG
4310 4442      GETSR    /GET SWITCH REG VALUE INTO AC
4311 4511      PRNT4    /TYPE OUT OCTAL VALUE
4312 4455      GETNUM   /GET INPUT FORM OPERATOR
4313 5317      JMP      NOIN /FIND OUT WHAT INVALID CHARACTER WAS
4314 5323      JMP      YESIN /SAVE INPUT AND FIND OUT WHAT CHARACTER WAS
4315 5345      JMP      CNTGRT /RETURN WITHOUT SAVING NEW VALUE
4316 5344      JMP      CNTGRT-1 /SAVE NEW PSEUDO-SR AND RETURN
4317 7240      NOIN, STA
4320 3347      DCA      INFLG /SET FLAG THAT NO OCTAL INPUT WAS RECEIVED
4321 3346      DCA      DIGIN /CLEAR OCTAL NUMBER INPUT
4322 5325      JMP      .+3
4323 3346      YESIN, DCA DIGIN /SAVE OCTAL NUMBER INPUT
4324 3347      DCA      INFLG /CLEAR FLAG THAT NUMBER IS VALID
4325 7240      STA      /USE PREVIOUS INPUT
4326 4505      LISN     /USE OLD KEYBOARD INPUT
4327 7571      -207     /IF CONTROL-G, THEN
4330 4302      XCTRLG-3 / REISSUE PROMPT
4331 7566      -212     /ON LINE FEED
4332 4340      LNFEED   /GO TO LNFEED
4333 0000      0        /ON UNMATCHED INPUT
4334 4335      .+1      /CONTINUE
4335 4506      MESSAGE /TYPE "?" <CR><LF>
4336 1060      QESMRK

```

```

4337 5306          JMP      XCTRLG+1      /START OVER WITH PROMPT
4340 1165 LNFEED, TAD      K200           /PICK UP STARTING ADDRESS OF
4341 3305          DCA      XCTRLG       /PROGRAM AND PLACE FOR STARTUP
4342 1346          TAD      DIGIN        /GET INPUT DIGITS
4343 2347          ISZ      INFLG        /SKIP IF NOT VALID
4344 3020          DCA      PSWR         /SAVE INPUT DIGITS IN PSEUDO SWITCH REG
4345 5705 CNTGRT, JMP I   XCTRLG       /RETURN W/AC CLEAR
/
4346 0000          DIGIN, 0
4347 0000          INFLG, 0

/WZITFG
/
/WAS IT F OR G? SUBROUTINE CHECKS IF LAST CHARACTER INPUT WAS CONTROL-F OR CONTROL-G,
/AND IF SO, CALLS THE APPROPRIATE HANDLER. AC IS CLEAR ON RETURN
/
4350 0000 XWZITF, 0
4351 7300          CLA CLL
4352 1144          TAD      LASTIN       /GET LAST CHARACTER INPUT
4353 1217          TAD      M207         /CHECK IF ^G
4354 7450          SNA      M207        /SKIP IF NOT
4355 4450          UPARG                     /HANDLE ^G
4356 7001          IAC                     /CHECK IF ^F
4357 7650          SNA CLA                /SKIP IF NOT
4360 4452          CNTRLF                 /HANDLE CONTROL F
4361 5750          JMP I   XWZITF        /RETURN W/ AC CLEAR

/PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11
4362 0000 XPRNT2, 0
4363 3373          DCA      P2SAVE       /CALL BY "PRNT2"
4364 1373          TAD      P2SAVE
4365 7012          RTR
4366 7010          RAR
4367 4507          PRNT1
4370 1373          TAD      P2SAVE
4371 4507          PRNT1
4372 5762          JMP I   XPRNT2
4373 0000 P2SAVE, 0

4377 6000 PAGE
4400 4400

/TYPE
/
/TYPE THE ASCII CHARACTER IN THE AC
4400 0000 XTYPE, 0
4401 3241          DCA      CHARSV       /CALL BY "TYPE"
4402 1142          CHKOK, TAD      NOPRNT /SAVE CHARACTER TO BE TYPED
4403 7650          SNA CLA                /OK TO PRINT?
4404 5207          JMP      .+3          /YES
4405 4447          CONSOL                     /WAIT FOR INPUT
4406 5202          JMP      .-4          /CHECK IF OUTPUT REENABLED
4407 6031          KSF                     /SKIP IF READER FLAG SET

```

```

4410 5226      JMP      OK2TYP      /NOT SET, TYPE DATA
4411 4777'     JMS      VT278
4412 6034      KRS
4413 1242      TAD      M223      /READ KEYBOARD--LEAVE FLAG SET
4414 7440      SZA
4415 5221      JMP      .+4      /ADD IN MINUS CONTROL-S
4416 6032      KCC      /SKIP IF A CONTROL-S
4417 2142      ISZ      NOPRNT    /GO CHECK FOR CNTRL-C
4420 5202      JMP      CHKOK    /CLEAR KEYBOARD FLAG
4421 1243      TAD      K20      /SET NO PRINT FLAG
4422 7640      SZA      CLA      /GO GET IN CONTROL-Q WAIT LOOP
4423 5226      JMP      OK2TYP    /CHECK FOR CONTROL-C
4424 6032      KCC      /SKIP IF CONTROL-C
4425 5354      JMP      CNTRLC    /NOT ^C--TYPE THE CHARACTER
4426 1241      OK2TYP, TAD      CHARSV /CLEAR KEYBOARD FLAG
4427 6046      TLS
4430 7200      CLA
4431 6041      TSF
4432 7410      SKP
4433 5237      JMP      .+4      /HANDLE CONTROL-C
4434 6102      SPL      /GET CHARACTER
4435 5231      JMP      .-4
4436 4000      JMS      0
4437 6042      TCF
4440 5600      JMP I  XTYPE
/
4441 0000      CHARSV, 0
4442 7555      M223,  -223
4443 0020      K20,   20

```

```

/TYPE A CR AND LF WITH NUMBER OF FILLERS
/AS DETERMINED BY LOCATION "FILLER"

```

```

4444 0000      XCRLF,  0      /CALL BY "CRLF"
4445 7200      CLA
4446 1260      TAD      K215
4447 4513      TYPE
4450 1023      TAD      FILLER
4451 7040      CMA
4452 3262      DCA      CRLFSV
4453 1261      TAD      K212
4454 4513      TYPE
4455 2262      ISZ      CRLFSV
4456 5254      JMP      .-2
4457 5644      JMP I  XCRLF
4460 0215      K215,  0215
4461 0212      K212,  0212
4462 0000      CRLFSV, 0

```

```

/COMPARE INPUT TO LIST FOLLOWING CALL
/INPUT ONE CHARACTER IF AC=0
/USE LAST INPUT IF AC NON ZERO

```

4463	0000	XLISN, 0	/CALL BY "LISN"
4464	7640	SZA CLA	
4465	5316	JMP LISN1	/USE LAST INPUT SINCE AC NOT ZERO
4466	6031	KSF	
4467	5266	JMP .-1	
4470	6036	KRB	
4471	0347	AND K177	
4472	1165	TAD K200	
4473	3144	DCA LASTIN	
4474	1352	TAD M203	/CHECK IF CHARACTER WAS A CONTROL-C
4475	1144	TAD LASTIN	
4476	7650	SNA CLA	/SKIP IF NOT
4477	5354	JMP CNTRLC	/GO HANDLE CONTROL-C
4500	1142	TAD NOPRNT	/CHECK OUTPUT ENABLE FLAG
4501	7640	SZA CLA	/SKIP IF OK TO OUTPUT
4502	5316	JMP LISN1	/DON'T PRINT CHARACTER
4503	1144	TAD LASTIN	
4504	1350	TAD M212	
4505	7450	SNA	/IS IT A LF?
4506	5312	JMP .+4	/YES
4507	1351	TAD M3	
4510	7640	SZA CLA	/IS IT A CR?
4511	5314	JMP .+3	/NO
4512	4514	CRLF	
4513	5316	JMP LISN1	/PRINT THE CHARACTER
4514	1144	TAD LASTIN	/GET COMPARE VALUE
4515	4513	TYPE	
4516	1663	LISN1, TAD I XLISN	
4517	2263	ISZ XLISN	
4520	7450	SNA	/EXIT?
4521	5342	JMP LISN3	/YES
4522	7500	SMA	
4523	5332	JMP LISNUM	/LOOK FOR OCTAL NUMBER
4524	1144	TAD LASTIN	/COMPARE
4525	7650	SNA CLA	/EQUAL?
4526	5342	JMP LISN3	/YES
4527	7200	LISN2, CLA	
4530	2263	ISZ XLISN	
4531	5316	JMP LISN1	
4532	7200	LISNUM, CLA	/LOOK FOR OCTAL NUMBER
4533	1144	TAD LASTIN	
4534	1340	TAD M270	
4535	7500	SMA	/IS IT LESS THAN 8?
4536	5327	JMP LISN2	/NO, SO NOT AN OCTAL NUMBER
4537	1353	TAD K10	
4540	7510	M270, SPA	/IS IT GREATER THAN ZERO?
4541	5327	JMP LISN2	/NO, SO NOT A NUMBER
4542	3132	LISN3, DCA TEMP2	
4543	1663	TAD I XLISN	
4544	3263	DCA XLISN	
4545	1132	TAD TEMP2	
4546	5663	JMP I XLISN	/AC IS ZERO UNLESS OCTAL NUMBER
4547	0177	K177, 0177	
4550	7566	M212, 7566	
4551	7775	M3, 7775	

```

4552 7575 M203, -203
4553 0010 K10, 10

4554 7200 CNTRLC, CLA
4555 3142 DCA NOPRNT /TURN OUTPUT ON
4556 4506 MESSAGE /THEN TYPE "C<CR><LF>"
4557 1070 UPARRC
4560 5761 JMP I OS8 /GO TO OS/8 MONITOR
4561 7600 OS8, 7600 /POINTER TO OS/8 BOOT

/TIMING GENERATOR FOR NOTIFYING APT SYSTEM. TICKS SHOULD BE
/SPACED AT LEAST ONCE EVERY 1 MS AND NOT MORE OFTEN THAN EVERY 40 US
/WHEN NOT CALLING SCOPE
/
4562 0000 XTICK, 0
4563 4443 APTCHK /SKIP IF NOT ON APT
4564 7410 SKP /ON APT -- GENERATE TIMING
4565 5372 JMP TIKRET /DON'T GENERATE TIMING
4566 2375 ISZ TOCK /COUNT A TIME UNIT
4567 5372 JMP TIKRET /NOT READY FOR APT NOTIFICATION
4570 6272 CIF 70 /FIELD OF UV-PROM
4571 4776 JMS I PAPTIM /NOTIFY APT
4572 6102 TIKRET, SPL /SKIP ON POWER LOW
4573 5762 JMP I XTICK /RETURN
4574 4000 JMS 0 /POWER FAIL INTERRUPT
/
4575 0000 TOCK, 0 /TIME UNIT COUNTER
4576 6500 PAPTIM, 6500 /POINTER TO APT'S TIMING ROUTINE
4577 6000
4600 PAGE

/PRINT PACKED ASCII TEXT TERMINATED BY
/SIX-BIT 00

4600 0000 MESAGX, 0 /CALL BY "MESSAGE"
4601 7344 CLA CLL CMA RAL /-2 INTO AC
4602 3243 DCA MESCNT /SET UP CHARACTER COUNTER
4603 1600 TAD I MESAGX
4604 3244 DCA MESSAV
4605 2200 ISZ MESAGX /SET UP RETURN
4606 6211 MESLUP, CDF 10 /FIELD OF TEXT
4607 1644 TAD I MESSAV
4610 6201 CDF 0 /BACK TO HERE
4611 2243 ISZ MESCNT /FIRST OR SECOND CHARACTER OF WORD
4612 5221 JMP MESBSW /FIRST-PUT CHARACTER INTO LOW BYTE
4613 2244 ISZ MESSAV /2ND--INCREMENT POINTER FOR NEXT TIME
4614 3132 DCA TEMP2 /SAVE AC
4615 7344 CLA CLL CMA RAL /RESTORE CHARACTER COUNTER
4616 3243 DCA MESCNT
4617 1132 TAD TEMP2
4620 7410 SKP
4621 7002 MESBSW, BSW /GET FIRST CHARACTER INTO POSITION
4622 0240 AND K77
4623 7450 SNA /TERMINATOR (00)?

```

```

4624 5600      JMP I  MESAGX      /YES
4625 1241      TAD      M43
4626 7450      SNA              /CRLF?
4627 5236      JMP      .+7       /YES
4630 1245      TAD      K3A
4631 7510      SPA              /200 OR 300
4632 1246      TAD      K100A     /300
4633 1242      TAD      K240     /200
4634 4513      TYPE
4635 7410      SKP
4636 4514      CRLF
4637 5206      JMP      MESLUP
4640 0077      K77, 0077
4641 7735      M43, 7735
4642 0240      K240, 0240
4643 0000      MESCNT, 0
4644 0000      MESSAV, 0
4645 0003      K3A, 3
4646 0100      K100A, 100

```

/IOT SUBROUTINES

/CLEAR DEVICE, ALL REGISTERS, AC, AND FLAGS

```

/
4647 0000      XRLDC, 0
4650 6600      IOT0, 6600        /ISSUE DEVICE CLEAR
4651 5647      JMP I  XRLDC      /RETURN (AC CLEAR)

```

/SKIP ON FUNCTION DONE, THEN CLEAR IF SET TO A ONE

```

/
4652 0000      XRLSD, 0
4653 6601      IOT1, 6601        /SKIP IF DEVICE DONE
4654 7410      SKP              /NORMAL RETURN IF NO SKIP
4655 2252      ISZ      XRLSD    /INCREMENT RETURN ADDR FOR SKIP
4656 5652      JMP I  XRLSD    /RETURN (AC UNCHANGED)

```

/LOAD BREAK MA REG FROM AC

```

/
4657 0000      XRLMA, 0
4660 6602      IOT2, 6602        /LOAD MA REGISTER
4661 5657      JMP I  XRLMA     /RETURN (AC CLEAR)

```

/LOAD COMMAND REGISTER A FROM AC

```

/
4662 0000      XRLCA, 0
4663 6603      IOT3, 6603        /LOAD COMMAND REG A
4664 5662      JMP I  XRLCA     /RETURN (AC CLEAR)

```

/LOAD COMMAND REGISTER B FROM AC

```

/
4665 0000      XRLCB, 0
4666 6604      IOT4, 6604        /LOAD COMMAND REG B
4667 5665      JMP I  XRLCB     /RETURN (AC CLEAR)

```

/LOAD SECTOR ADDRESS REGISTER FROM AC 0:5


```

/
4670 0000 XRLSA, 0
4671 6605 IOT5, 6605 /LOAD SECTOR ADDRESS REG
4672 5670 JMP I XRLSA /RETURN (AC CLEAR)

/"SPARE/LOAD WORD COUNT FROM AC
/
4673 0000 XRLWC, 0
4674 6607 IOT7, 6607 /LOAD WORD COUNT
4675 5673 JMP I XRLWC /RETURN (AC CLEAR)

/READ ERROR REGISTER INTO AC 0:2
/
4676 0000 XRRER, 0
4677 6610 IOT10, 6610 /READ ERROR REGISTER
4700 5676 JMP I XRRER /RETURN (AC=E000)

/READ WORD COUNT REG INTO AC
/
4701 0000 XRRWC, 0
4702 6611 IOT11, 6611 /READ WORD COUNT
4703 5701 JMP I XRRWC /RETURN W/WC IN AC

/READ COMMAND REG A INTO AC
/
4704 0000 XRRCA, 0
4705 6612 IOT12, 6612 /READ COMMAND REG A INTO AC
4706 5704 JMP I XRRCA /RETURN W/CA IN AC

/READ COMMAND REG B INTO AC
/
4707 0000 XRRCB, 0
4710 6613 IOT13, 6613 /READ COMMAND REG B
4711 5707 JMP I XRRCB /RETURN W/ INFO IN AC

/READ SECTOR ADDRESS REGISTER INTO AC 0:5
/
4712 0000 XRRSA, 0
4713 6614 IOT14, 6614 /READ SECTOR ADDR
4714 5712 JMP I XRRSA /RETURN (AC=SA00)

/READ SILO WORD INTO AC 4:11
/
4715 0000 XRRSI, 0
4716 6615 IOT15, 6615 /READ SILO
4717 5715 JMP I XRRSI /RETURN (AC 0:3 CLEAR; 4:11=SILO)

/SKIP ON DRIVE ERROR, THEN CLEAR IF SET TO A ONE
/
4720 0000 XRLSE, 0
4721 6617 IOT17, 6617 /SKIP ON DRIVE ERROR
4722 7410 SKP /NORMAL RETURN (NO SKIP)
4723 2320 ISZ XRLSE /INC RETURN ADDR FOR SKIP
4724 5720 JMP I XRLSE /RETURN (AC UNCHANGED)

```

```

/STACHK
/
/CHECK STATUS RECEIVED FROM DRIVE. SKIP IF STATUS IS NOT AS EXPECTED.
/ACTUAL STATUS IS IN DATA1 AND DATA2. EXPECTED STATUS IS IN
/ATA3 AND DATA4.
/
4725 0000 XSTACH, 0
4726 1117 TAD DATA1 /COMPARE EXPECTED WORD 1
4727 7041 CIA /((BY SUBTRACTION)
4730 1121 TAD DATA3 /WITH ACTUAL WORD 1
4731 7640 SZA CLA /SKIP IF EQUAL
4732 5337 JMP .+5 /ERROR--EXIT WITH A SKIP
4733 1120 TAD DATA2 /COMPARE ACTUAL STATUS WORD 2
4734 7041 CIA /WITH EXPECTED
4735 1122 TAD DATA4
4736 7640 SZA CLA /SKIP IF EQUAL
4737 2325 ISZ XSTACH /ERROR--SKIP ON RETURN
4740 5725 JMP I XSTACH /RETURN W/AC CLEAR

/JMPPM1
/
/THIS SUBROUTINE ACTS AS A "JMP .-1" INSTRUCTION EXCEPT THAT THE AC IS
/CLEARED AND CONSOL INPUT CAN BE PROCESSED.
/
4741 0000 XJMPPM, 0
4742 4447 CONSOL /CHECK FOR CONSOLE INPUT
4743 7344 CLA CLL CMA RAL /-2 INTO AC
4744 1341 TAD XJMPPM /SUBTRACT 2 FROM RETURN PC
4745 3341 DCA XJMPPM /SO RETURN IS TO CALL-1
4746 6102 SPL /SKIP ON POWER LOW
4747 5741 JMP I XJMPPM /RETURN TO CALL-1 W/AC CLEAR
4750 4000 JMS 0 /POWER FAIL INTERRUPT

/POWER FAIL ROUTINES
/
4751 7200 PWRFAL, CLA
4752 1000 TAD 0 /GET PC
4753 3117 DCA DATA1 /SAVE FOR REFERENCE OR ERROR TYPE OUT
4754 6102 SPL /SKIP ON POWER LOW
4755 5361 JMP IMPINT /IMPOSSIBLE INTERRUPT--GO REPORT IT
4756 1364 TAD JMPUP /GET RESTART INSTRUCTION
4757 3000 DCA 0 /SAVE FOR POWER UP
4760 7402 HLT /WAIT FOR POWER UP RESTART

4761 4446 IMPINT, ERROR /IMPOSSIBLE INTERRUPT (INTERRUPT SYSTEM IS OFF)!
4762 2275 SFTWER / PC INTPC
4763 5565 JMP I K200 /RESTART PROGRAM

4764 5403 JMPUP, JMP I 3 /JUMPS TO POWER UP ROUTINE

4765 6035 PWRUP, KIE /DISABLE KEYBOARD INTERRUPTS
4766 3000 DCA 0 /CLEAR POWER UP INSTRUCTION
4767 4443 APTCHK /SKIP IF NOT ON APT
4770 5376 JMP NOPMES
4771 4506 MESSAGE /POWER FAIL PC=

```

```

4772 2242      POWER
4773 1117      TAD      DATA1      /GET PC OF POWER FAIL
4774 4511      PRNT4      /PRINT IT
4775 4514      CRLF
4776 4466      NOPMES, SETTIM      /SET UP 40 SECOND WAIT FOR DRIVE SPIN UP
DECIMAL
4777 0140      -4000
5000 7160      -400

OCTAL
5001 4467      TIMCHK
5002 4470      JMPPM1
5003 4457      RESET      /WAIT FOR TIMOUT
5004 1126      TAD      LOOPPT      /RESET THE DRIVE
5005 7650      SNA CLA      /GET LOOP POINTER
5006 5565      JMP I      K200      /SKIP IF POINTER IS VALID
5007 7201      CLA IAC      /RESTART PROGRAM FROM BEGINNING
5010 3130      DCA      LPRQST      /FLAG THAT TEST IS ENTERED FOR FIRST TIME
5011 5526      JMP I      LOOPPT      /RESTART INTERRUPTED TEST

/RESET
/
/EXECUTE RESET COMMAND TO DRIVE UNDER TEST.  WAIT FOR DONE.
/
5012 0000      XRESET, 0
5013 7200      CLA      /CLEAR OUT REGISTERS THAT LOAD DAR WITH GARBAGE
5014 4427      RLCA
5015 4431      RLSA
5016 1116      TAD      DRVNUM      /ADD IN DRIVE NUMBER
5017 7002      BSW      /PLACE INTO DRIVE SELECT BITS
5020 7001      IAC      /INCREMENT AC (FOR RESET COMMAND)
5021 4430      RLCB      /ISSUE COMMAND
5022 4425      RLSD      /WAIT FOR DONE
5023 4470      JMPPM1
5024 5612      JMP I      XRESET      /RETURN W/AC CLEAR

/GETSTA
/
/ISSUE GET STATUS TO DRIVE UNDER TEST.  WAIT FOR DONE.  SAVE STATUS.
/
5025 0000      XGETST, 0
5026 7200      CLA
5027 4431      RLSA      /CLEAR SECTOR ADDRESS (LOADS DAR WITH GARBAGE)
5030 4427      RLCA      /CLEAR COMMAND A
5031 1116      TAD      DRVNUM      /GET DRIVE NUMBER INTO
5032 7002      BSW      /DRIVE SELECT BITS FOR USE IN CB
5033 1377      TAD      (MODE8+2)      /SET 8 BIT MODE AND FUNCTION 2 (GET STATUS)
5034 4430      RLCB      /ISSUE COMMAND
5035 4425      RLSD      /WAIT FOR DONE
5036 4470      JMPPM1
5037 4433      RRER      /READ ERROR REG
5040 7004      RAL      /GET OPI INTO SIGN BIT
5041 7500      SMA      /SKIP IF OPI SET FROM GET STATUS
5042 5253      JMP      STAOK      /OK--GO SAVE STATUS
5043 7010      RAR      /RESTORE ERROR REG
5044 3117      DCA      DATA1      /SAVE IT FOR ERROR TYPEOUT

```

```

5045 7240          STA          /COMPUTE CALL PC
5046 1225          TAD          XGETST
5047 3120          DCA          DATA2
5050 4446          ERROR
5051 2746          NOCLK
5052 5776          JMP          FATAL
5053 4440          STAOK, RRSI
5054 0163          AND          K377
5055 3117          DCA          DATA1
5056 4440          RRSI
5057 0163          AND          K377
5060 3120          DCA          DATA2
5061 5625          JMP I       XGETST          /RETURN W/AC CLEAR

/SETTIM
/-(#MS/10)
/-(10MS/(EXECUTION TIME OF LOOP))
/
/SUBROUTINE INITIALIZES REAL TIME CLOCK(IF ONE). THE COUNT IS SET UP FOR THE
/AMOUNT OF TIME TO ELAPSE (ARG #1) BEFORE TIMCHK SUBROUTINE SKIPS. IF NO CLOCK
/IS PRESENT, THEN ARG #2 IS USED TO SET UP COUNT FOR TIMCHK TO COUNT NUMBER
/OF CALLS BEFORE ESTIMATING THAT 10MS HAVE PASSED.
/
5062 0000          XSETTI, 0
5063 7200          CLA
5064 1662          TAD I       XSETTI          /GET FIRST ARGUMENT AND SET
5065 3332          DCA          TIMCNT          /COUNTER
5066 2262          ISZ          XSETTI          /SKIP OVER ARGUMENT
5067 1146          TAD          OPT1          /CHECK IF OPTION 1 IS PRESENT
5070 7650          SNA CLA          /SKIP IF YES
5071 5277          JMP          .+6          /GO SET UP FOR GROSS TIMING
5072 6136          6136          /CLEAR CLOCK FLAG AND WAIT FOR BEGINNING OF NEW CLOCK PERIOD
5073 6137          6137          /SKIP ON CLOCK FLAG
5074 4470          JMPPM1        /WAIT FOR FLAG
5075 6136          6136          /CLEAR CLOCK FLAG
5076 5303          JMP          .+5          /EXIT
5077 1662          TAD I       XSETTI          /GET GROSS TIMING ADJUSTMENT
5100 3333          DCA          GRUSCT          /SET COUNTER
5101 1333          TAD          GRUSCT
5102 3334          DCA          GRSCNT
5103 2262          ISZ          XSETTI
5104 5662          JMP I       XSETTI          /SKIP OVER 2ND ARGUMENT

/
/TIMCHK
/
/USED IN CONJUNCTION WITH SETTIM TO TIME WAIT LOOPS
/((15 USEC EXECUTION TIME WHEN USED FOR GROSS TIMING))
/
5105 0000          XTIMCH, 0
5106 7200          CLA
5107 1146          TAD          OPT1          /IS OPTION 1 PRESENT?
5110 7650          SNA CLA          /SKIP IF YES
5111 5320          JMP          .+7          /GO ESTIMATE GROSS TIMING
5112 6137          6137          /SKIP IF CLOCK FLAG SET
5113 5327          JMP          TIMRET          /RETURN

```

```

5114 6136          6136          /CLEAR CLOCK FLAG
5115 2332          ISZ   TIMCNT   /WAIT FOR MORE CLOCK FLAGS?
5116 5327          JMP    TIMRET  /YES-- RETURN
5117 5326          JMP    TIMRET-1 /NO--SKIP ON RETURN
5120 2334          ISZ   GRSCNT   /COUNT A TIME THAT ROUTINE HAS BEEN CALLED
5121 5327          JMP    TIMRET  /RETURN
5122 1333          TAD    GROSCT   /RESTORE COUNTER
5123 3334          DCA    GRSCNT
5124 2332          ISZ   TIMCNT   /TICK 10MS TIMER
5125 7410          SKP                    /WAIT SOME MORE--RETURN
5126 2305          ISZ   XTIMCH   /SKIP ON RETURN
5127 6102          TIMRET, SPL     /SKIP ON POWER LOW
5130 5705          JMP I  XTIMCH  /RETURN
5131 4000          JMS    0       /POWER FAIL INTERRUPT
/
5132 0000          TIMCNT, 0
5133 0000          GROSCT, 0
5134 0000          GRSCNT, 0

/GETSR
/
/SUBROUTINE TO READ SWITCH REG OR PSEUDO-SR INTO AC, DEPENDING ON
/STATE OF BIT 0 IN LOCATION 21
/
5135 0000          XGETSR, 0
5136 7200          CLA
5137 1021          TAD    HCN1     /PICK UP HARDWARE CONFIGURATION WORD 1
5140 7710          SPA CLA       /SKIP IF USING SOFTWARE SWITCHES
5141 7614          LAS SKP       /READ HARDWARE SWITCHES
5142 1020          TAD    PSWR    /READ PSEUDO SWR
5143 5735          JMP I  XGETSR  /RETURN W/SR VALUE IN AC

/APTCHK
/
/SUBROUTINE SKIPS IF NOT RUNNING UNDER APT
/
5144 0000          XAPTCH, 0
5145 7200          CLA
5146 1022          TAD    HCN2     /GET HARDWARE CONFIGURATION WORD 2
5147 7700          SMA CLA       /SKIP IF ON APT (NORMAL RETURN)
5150 2344          ISZ   XAPTCH   /SKIP ON RETURN (NOT ON APT)
5151 5744          JMP I  XAPTCH  /RETURN W/AC CLEAR

/GETRES
/
/ THIS SUBROUTINE WAITS FOR OPERATOR RESPONSE TO YES, NO QUESTIONS. A <CR> IS
/TREATED AS A NO RESPONSE. IF AN INVALID RESPONSE IS RECEIVED, A NORMAL
/RETURN IS MADE, ELSE THE PROGRAM SKIPS. IF THE RESPONSE WAS "Y", THE AC
/CONTAINS A "1" ON RETURN, ELSE 0.
/
5152 0000          XGETRE, 0
5153 7200          CLA           /USE NEW INPUT
5154 4505          LISN
5155 7447          -"Y         /IF YES--SAVE FOR 1 IN AC ON RETURN
5156 5167          YES        /TYPE CRLF AN SKIP
5157 7462          -"N         /IF NO--SAVE FOR 0 IN AC ON RETURN

```

5160	5170	NO		/AND TYPE CRLF AND SKIP
5161	7563	-215		/IF <CR>--SKIP ON RETURN
5162	5173	RESRET		
5163	0000	0		/IF INVALID INPUT--CHECK FOR ^F OR ^G,
5164	5165	.+1		
5165	4456	WZITFG		/AND RETURN (NO SKIP)
5166	5374	JMP	RESRET+1	
5167	7001	YES, IAC		/SET AC TO ONE
5170	3375	NO, DCA	RESPON	/SAVE RETURN VALUE
5171	4514	CRLF		/TYPE CRLF
5172	1375	TAD	RESPON	/GET RETURN VALUE
5173	2352	RESRET, ISZ	XGETRE	/SKIP ON RETURN IF VALID INPUT
5174	5752	JMP I	XGETRE	/RETURN WITH 0 OR 1 IN AC
		/		
5175	0000	RESPON, 0		
5176	5741			
5177	1002			
	5200	PAGE		
		/GETNUM		
		/SUBROUTINE WAITS FOR OCTAL DIGITS TO BE INPUT FOLLOWED BY <CR>.		
		/THE FOLLOWING RETURNS ARE USED:		
		/ CALL +1		/NON-OCTAL, NON-CR CHARACTER INPUT (AC CLEAR)
		/ CALL +2		/5 DIGITS OR DIGITS FOLLOWED BY NON-CR (AC CONTAINS FIRST VALID DIGITS)
		/ CALL +3		/NO DIGITS ENTERED (AC CLEAR)
		/ CALL +4		/1 TO 4 DIGITS ENTERED (AC CONTAINS OCTAL INPUT)
		/		
5200	0000	XGETNU, 0		
5201	7200	CLA		
5202	1175	TAD	M5	/SET UP COUNTER FOR MAXIMUM OF FOUR
5203	3240	DCA	DIGCNT	/DIGITS (1'S COMPLEMENT IS NEEDED)
5204	3237	DCA	DIGITS	/ZERO OUT OCTAL NUMBER STORAGE
5205	4505	GTNMLP, LISN		
5206	7563	-215		
5207	5227	CARRET		/IF <CR>, RETURN
5210	0001	1		
5211	5214	.+3		/IF OCTAL INPUT PROCESS BELOW
5212	0000	0		
5213	5231	CARRET+2		/IF NON-OCTAL INPUT, SKIP OVER CARRIAGE
5214	2240	ISZ	DIGCNT	/RETURN HANDLER
5215	5220	JMP	.+3	/SKIP IF TOO MANY DIGITS
5216	7200	CLA		/OK--CONTINUE
5217	5234	JMP	TOOMNY	/GET RID OF LAST DIGIT ENTERED
5220	3131	DCA	TEMP1	/RETURN
5221	1237	TAD	DIGITS	/SAVE DIGIT JUST ENTERED
5222	7104	CLL RAL		/GET PREVIOUS DIGITS
5223	7006	RTL		/ROTATE BITS OVER ONE OCTAL DIGIT
5224	1131	TAD	TEMP1	/
5225	3237	DCA	DIGITS	POSITION
5226	5205	JMP	GTNMLP	/ADD IN NEW DIGIT
5227	2200	CARRET, ISZ	XGETNU	/SAVE DIGIT STRING
5230	2200	ISZ	XGETNU	/GO WAIT FOR MORE INPUT
5231	1777	TAD	K5	/ADD TWO TO RETURN ADDRESS SINCE
				/CARRIAGE RETURN WAS TYPED
				/ADD FIVE TO DIGITS COUNTER TO SEE IF

```

5232 1240      TAD      DIGCNT      /ANY DIGITS WERE ENTERED
5233 7640      SZA CLA      /SKIP IF NO DIGITS WERE ENTERED
5234 2200      TOOMNY, ISZ      XGETNU  /EXTRA SKIP ON RETURN
5235 1237      TAD      DIGITS      /GET DIGITS PRIOR TO RETURN
5236 5600      JMP I      XGETNU      /RETURN--AC HAS DIGITS OR 0
/
5237 0000      DIGITS, 0
5240 0000      DIGCNT, 0
/ENTVAL
/
/SUBROUTINE PROMPTS OPERATOR TO ENTER OCTAL VALUE.  VALUE MUST BE LESS THAN
/777.  RETURN IS MADE WITH VALUE IN AC.
/
5241 0000      XENTVA, 0
5242 4506      MESSAGE      /PROMPT OPERATOR
5243 0723      ENOVAL      /ENTER VALUE IN OCTAL
5244 4455      GETNUM      /GET OCTAL INPUT (UP TO THREE DIGITS)
5245 5260      JMP      ENTLIS      /GO CHECK WHAT LAST INPUT WAS
5246 5260      JMP      ENTLIS
5247 7000      NOP      /OK IF NO DIGITS ENTERED
5250 3131      DCA      TEMP1      /SAVE DIGITS ENTERED
5251 1131      TAD      TEMP1
5252 0177      AND      K7000      /SAVE ONLY INVALID BITS
5253 7650      SNA CLA      /SKIP IF OUT OF RANGE
5254 5262      JMP      ENTRET      /GET DIGITS AND RETURN
5255 4506      MESSAGE      /TYPE "?<CR>" AND TRY AGAIN
5256 1060      QESMRK
5257 5242      JMP      XENTVA+1
5260 4456      ENTLIS, WZITFG      /HANDLE CONTROL-G OR F IF IT WAS
5261 5242      JMP      XENTVA+1      /REISSUE PROMPT
5262 1131      ENTRET, TAD      TEMP1      /GET NUMBER INPUT
5263 5641      JMP I      XENTVA      /RETURN W/NUMBER IN AC
/
/PRINT 2 SPACES
5264 0000      SPACX2, 0      /CALL BY "SPACE2"
5265 4506      MESSAGE
5266 1106      TWOSPA
5267 5664      JMP I SPACX2
/
/JMS WAIT1
/
/SUBROUTINE REQUIRES APPROXIMATELY 1 MS TO EXECUTE ON PDP8/E
/
5270 0000      WAIT1, 0
5271 7200      CLA
5272 1277      TAD      M404
5273 3131      DCA      TEMP1
5274 2131      ISZ      TEMP1
5275 5274      JMP      .-1
5276 5670      JMP I      WAIT1
/
5277 7374      M404,      -404

```

HP 001

```

/JMS WRENWT
/
/IF NECESSARY, REQUEST OPERATOR TO WRITE ENABLE THE DRIVE & WAIT FOR CHANGE
/
5300 0000 WRENWT, 0
5301 4460 GETSTA /GET STATUS
5302 1120 TAD DATA2 /PICK UP STATUS WORD 2
5303 7002 BSW /PUT WRITE LOCK STATUS BIT INTO SIGN BIT
5304 7700 SMA CLA /SKIP IF DRIVE WRITE LOCKED
5305 5320 JMP WRERET /RETURN
5306 4506 MESSAGE /*PLEASE WRITE ENABLE DRIVE "
5307 1125 RLWREN
5310 1116 TAD DRVNUM /TYPE DRIVE NUMBER
5311 4507 PRNT1
5312 4460 GETSTA /GET STATUS
5313 1120 TAD DATA2 /PICK UP STATUS WORD 2
5314 7002 BSW /PUT WRITE LOCK STATUS BIT INTO SIGN BIT
5315 7710 SPA CLA /SKIP IF WRITE LOCK NOW RESET
5316 5312 JMP .-4 /KEEP MONITORING STATUS
5317 4514 CRLF
5320 5700 WRERET, JMP I WRENWT

/REDHDR
/
/THIS SUBROUTINE ISSUES A READ HEADER TO THE DRIVE UNDER TEST AND WAITS
/FOR DONE. THE HEADER WORDS ARE LEFT IN THE SILO.
/
5321 0000 XREDHD, 0
5322 7200 CLA
5323 1116 TAD DRVNUM /GET DRIVE SELECT BITS INTO POSITION
5324 7002 BSW
5325 1376 TAD (MODE8+4) /SET FOR 8 BIT MODE AND READ HEADER COMMAND
5326 4430 RLCB /ISSUE COMMAND
5327 2775' ISZ TOCK /COMPENSATE FOR APT TIMING
5330 5333 JMP .+3 /OK--WAIT FOR DONE
5331 7240 STA /CAN'T CALL APT YET--SET UP FOR NEXT "TICK"
5332 3775' DCA TOCK / CALL TO CALL APT
5333 4425 RLSD /WAIT FOR DONE
5334 4470 JMPPM1
5335 5721 JMP I XREDHD

/THIS SUBROUTINE MOVES BAD SECTOR ENTRIES OUT OF THE BAD SECTOR FILE IN THE
/READ BUFFER INTO A PERMANENT AREA POINTED TO BY AN ARGUMENT FOLLOWING THE
/SUBROUTINE CALL. THEY ARE STORED IN TWO WORD BLOCK AS FOLLOWS:
/ CYLINDER CYLINDER IN LSB'S
/ TRACK-SECTOR TRACK IN SIGN BIT; SECTOR IN LSB'S
/ AFTER THE LAST ENTRY, THE CYLINDER POSITION OF THE NEXT WORD IS ALL 1'S.
/ ((THE TRACK-SECTOR ENTRY IS NOT DEFINED.) NOTE THAT ASCENDING ORDER CANNOT
/ BE ASSUMED SINCE THE FIELD BAD WILL GET WRITTEN AFTER THE FACTORY BAD.
/
5336 0000 BADFIL, 0
5337 7240 STA
5340 1736 TAD I BADFIL /SUBTRACT 1 FROM POINTER ARGUMENT
5341 3011 DCA AUTO11 /FOR USE IN AN AUTO-INC REG
/ AUTO11 MUST BE USED BECAUSE IT'S VALUE

```


5342	2336	ISZ	BADFIL	/IS USED AFTER RETURN IN CALL FROM	
5343	1374	TAD	(REDBUF+7	/MANUFACTURING SECTION	
5344	3012	DCA	AUTO12	/SKIP OVER ARGUMENT	
5345	4773	FILLUP, JMS	GET4	/SET UP A POINTER TO 1 BEFORE FIRST	
5346	1120	TAD	DATA2	/ BAD SECTOR ENTRY	
5347	7010	RAR		/GET A BAD SECTOR ENTRY USING AUTO12	
5350	7640	SZA CLA		/CHECK IF END OF FILE	
5351	5364	JMP	BADEND	/CLEAR THE LINK AND SAVE CYL MSB	HP 001
5352	1117	TAD	DATA1	/SKIP IF NOT	
5353	7004	RAL		/END OF BAD SECTOR FILE	
5354	6211	CDF	10	/GET CYLINDER	
5355	3411	DCA I	AUTO11	/PUT THE MSB FROM LINK IN CYL WORD	HP 001
5356	1122	TAD	DATA4		
5357	7112	CLL RTR		/SAVE IT	
5360	1121	TAD	DATA3	/GET TRACK	
5361	3411	DCA I	AUTO11	/PLACE IN SIGN BIT	
5362	6201	CDF	0	/ADD IN SECTOR	
5363	5345	JMP	FILLUP	/PLACE IT	
5364	7240	BADEND, STA		/CONTINUE WITH FILE	
5365	6211	CDF	10	/FLAG END OF FILE	
5366	3411	DCA I	AUTO11		
5367	6201	CDF	0	/PLACE EOF MARKER	
5370	7201	CLA IAC		/FLAG BAD SECTOR FILE VALID	
5371	3156	DCA	BSVALD		
5372	5736	JMP I	BADFIL	/RETURN	

5373 6004
5374 3607
5375 4575
5376 1004
5377 6655
5400

PAGE

/SEEKTO

/

/THIS SUBROUTINE IS CALLED WITH THE DESIRED TRACK AND CYLINDER IN THE AC
/((IN AC1 AND AC4-AC11 RESPECTIVELY, AS IS REQUIRED IN COMMAND A).
/THE HEAD POSITION IS CHECKED AND IF DIFFERENT THAN THAT IN THE AC, A
/SEEK IS PERFORMED TO POSITION THE HEADS IN THE DESIRED POSITION. SEEK
/FAILURES ARE REPORTED. IF THE DRIVE DID NOT BECOME READY, THAT ERROR IS
/REPORTED, A CALL TO YNOTRY IS MADE, AND THE SUBROUTINE RETURNS TO
/CALL+1 (NORMAL RETURN). ON A SEEK FAILURE, A RETURN TO CALL +1 IS MADE. IF THE
/HEADS WERE IN THE DESIRED POSITION OR IF THE SEEK WAS SUCCESSFUL, THE RETURN
/IS TO CALL+2.

5400 0000
5401 3302
5402 1302
5403 7421
5404 4306
5405 5277
5406 3120
5407 7004
5410 3117

XSEEK, 0

DCA	CYLSAV	/SAVE THE CYLINDER TO SEEK TO	
TAD	CYLSAV	/GET THE HEAD AND CYLINDER	HP 002
MQL		/AND DISPLAY THEM IN THE MQ, CLEAR THE AC.	HP 002
JMS	COMPOS	/COMPARE HEAD POSITION	
JMP	SEEKOK	/NO SEEK NEEDED--RETURN	
DCA	DATA2	/SAVE CYLINDER WE ARE AT	
RAL		/GET HEAD THAT IS SELECTED	
DCA	DATA1	/SAVE IT	

HP 001

5411	1302	TAD	CYLSAV	/FORM A DIFFERENCE WORD
5412	0161	AND	K0777	/
5413	7041	CIA		
5414	1120	TAD	DATA2	
5415	7500	SMA		/SKIP IF DIRECTION 1 IS REQUIRED
5416	5221	JMP	.+3	/DON'T SET DIRECTION BIT
5417	7041	CIA		/FORM A POSITIVE DIFFERENCE WORD
5420	1305	TAD	K4000	/ADD IN DIRECTION TO DIFFERENCE
5421	3131	DCA	TEMP1	/SAVE DIRECTION AND DIFFERENCE
5422	7332	CLA STL	RTR	/SET HEAD BIT
5423	0302	AND	CYLSAV	/MASK IT OUT IF HEAD 0 IS DESIRED
5424	1131	TAD	TEMP1	/ADD IN DIRECTION AND DIFFERENCE
5425	3123	DCA	DATA5	/SAVE COMMAND A WORD
5426	1123	TAD	DATA5	
5427	4427	RLCA		/LOAD COMMAND A
5430	4431	RLSA		/CLEAR SECTOR ADDRESS REG
5431	1116	TAD	DRVNUM	/GET DRIVE SELECT BITS INTO POSITION
5432	7102	CLL BSW		/(CLEAR LINK FOR TOCK TEST BELOW)
5433	1304	TAD	K3	/SET FUNCTION BITS FOR SEEK COMMAND
5434	4430	RLCB		/ISSUE COMMAND
5435	1777	TAD	TOCK	/COMPENSATE APT TIMER FOR SEEK TIME
5436	1303	TAD	K144	/ADD 100
5437	7430	SZL		/SKIP IF NO OVERFLOW
5440	7240	STA		/BACK TIMER UP TO -1
5441	3777	DCA	TOCK	/SAVE NEW VALUE
5442	4425	RLSD		/WAIT FOR DONE
5443	4470	JMPPM1		
5444	4336	JMS	RDYWAT	/WAIT UP TO 3 SECONDS FOR READY
5445	5264	JMP	SEKCON	/DRIVE READY--CONTINUE
5446	4433	RRER		/READ AND STORE ERROR REG
5447	3121	DCA	DATA3	
5450	4436	RRCB		/READ AND STORE COMMAND B
5451	3122	DCA	DATA4	
5452	4435	RRCA		/READ AND STORE COMMAND A
5453	3123	DCA	DATA5	
5454	7240	STA		
5455	1200	TAD	XSEEKT	/COMPUTE CALL PC AND SAVE IT
5456	3124	DCA	DATA6	
5457	4460	GETSTA		/GET STATUS AND SAVE IT
5460	4446	ERROR		/READY NOT SET WITHIN 3 SECONDS
5461	3036	NORDY3		/PC DRVNO WD1-STATUS-WD2 ER CB CA CALPC
5462	4461	ERRHAN		/HANDLE DRIVE NOT READY
5463	5300	JMP	SEKRET	/MAKE FAILING RETURN
5464	4306	SEKCON, JMS	COMPOS	/COMPARE HEAD POSITION WITH DESIRED POSITION
5465	5277	JMP	SEEKOK	/OK--MAKE GOOD RETURN
5466	3122	DCA	DATA4	/NOT OK--SAVE ACTUAL CYLINDER
5467	7004	RAL		/GET THE HEAD SELECT OUT OF LINK
5470	3121	DCA	DATA3	/SAVE IT
5471	7240	STA		/SUBTRACT 1 FROM RETURN PC
5472	1200	TAD	XSEEKT	
5473	3124	DCA	DATA6	/SAVE CALL PC
5474	4446	ERROR		/SEEK FAILURE
5475	3022	SEKFAL		/PC DRV NO. HEAD-BEFOR-CYL HEAD-AFTER-CYL CA CALLPC
5476	7410	SKP		/MAKE FAILING RETURN
5477	2200	SEEKOK, ISZ	XSEEKT	/SKIP FOR AT POSITION

```

5500 4447 SEKRET, CONSOL          /ALLOW CONSOLE INPUT IN CASE OF FAILURE
5501 5600      JMP I   XSEKT      /NO SKIP FOR DRIVE NOT READY
/
5502 0000      CYLSAV, 0
5503 0144      K144,  144
5504 0003      K3,    3
5505 4000      K4000, 4000

/JMS COMPOS
/
/COMPARE HEAD POSITION TO DESIRED VALUE (IN CYLSAV). IF THE SAME, MAKE NORMAL
/RETURN WITH AC CLEAR. IF DIFFERENT, SKIP ON RETURN AND RETURN WITH CYLINDER
/ADDRESS IN AC AND HEAD SELECTED IN LINK.
/
5506 0000      COMPOS, 0
5507 4465      REDHDR              /READ A HEADER
5510 4440      RRSI                /GET HEADER WORD 1
5511 7002      BSW                 /PUT HEAD SEL INTO BIT 0, CYL LSB INTO LINK
5512 7012      RTR
5513 3131      DCA      TEMP1      /SAVE HEAD SELECT BIT
5514 4440      RRSI                /GET HEADER WORD 2
5515 7004      RAL                 /MOVE LSB INTO POSITION
5516 0161      AND      K0777      /MASK GARBAGE AND 0 BIT
5517 3132      DCA      TEMP2      /SAVE CYLINDER
5520 7330      STL CLA RAR         /GET MASK FOR HEAD SELECT BIT
5521 0131      AND      TEMP1      /PICK UP HEAD SELECT BIT
5522 7110      CLL RAR            /PUT INTO CORRECT POSITION
5523 1132      TAD      TEMP2      /ADD IN CYLINDER ADDRESS
5524 7041      CIA                 /COMPARE TO DESIRED HEAD AND CYLINDER
5525 1302      TAD      CYLSAV     / (AS SPECIFIED IN CYLSAV)
5526 7650      SNA CLA            /SKIP IF NOT EQUAL
5527 5335      JMP      COMRET     /OK--RETURN WITH AC CLEAR
5530 7330      STL CLA RAR         /GET HEAD SELECT BIT
5531 0131      AND      TEMP1
5532 7104      CLL RAL            /PUT HEAD SELECT INTO LINK
5533 1132      TAD      TEMP2      /PUT CYL ADDRESS INTO AC
5534 2306      ISZ      COMPOS     /SKIP ON RETURN
5535 5706      COMRET, JMP I  COMPOS /RETURN

/JMS RDYWAT
/
/THIS SUBROUTINE WAITS FOR DRIVE READY IN A 3 SECOND TIMEOUT LOOP.
/IF THE DRIVE BECAME READY, A NORMAL RETURN IS MADE. IF THE DRIVE DID NOT
/BECOME READY, THE ROUTINE SKIPS ON RETURN (NO ERROR IS REPORTED).
/
5536 0000      RDYWAT, 0
5537 4466      SETTIM              /SET UP THE TIMER
DECIMAL
5540 7324      -300                /3 SECONDS
5541 7160      -400                /COMPENSATION IN CASE OF NO CLOCK
OCTAL
5542 4433      RRER              /GET ERROR REGISTER
5543 7010      RAR               /ROTATE DRIVE READY BIT INTO LINK
5544 7630      SZL CLA           /SKIP IF DRIVE NOT READY
5545 5351      JMP      RDYRET    /DRIVE READY--GET OUT OF LOOP (RETURN)

```

```

5546 4467      TIMCHK      /CHECK UP ON REAL TIME CLOCK
5547 5342      JMP        .-5      /TIME IS NOT UP
5550 2336      ISZ        RDYWAT   /NOT READY--SKIP ON RETURN
5551 5736      RDYRET, JMP I  RDIWAT /RETURN

      /SKOCYL
      /
      /SEEK TO CYL 0, HEAD IN LOCATION HEAD. THEN SEEK TO HEAD, CYLNR.
      /ON FAILURE, START OVER.
      /
5552 0000      XSKOCY, 0
5553 7200      CLA
5554 1140      TAD        HEAD      /SEEK TO CYL 0, HEAD
5555 4462      SEEKTO
5556 5354      JMP        .-2      /TRY AGAIN ON FAILURE
5557 1140      TAD        HEAD      /SEEK TO CYLNR, HEAD
5560 1136      TAD        CYLNR
5561 4462      SEEKTO
5562 5354      JMP        .-6      /TRY AGAIN ON FAILURE
5563 5752      JMP I      XSKOCY

      /SKHCYL
      /
      /SEEK TO CYLINDER 777, HEAD IN "HEAD". THEN SEEK TO HEAD, CYLNR.
      /
5564 0000      XSKHCY, 0
5565 7200      CLA
5566 1140      TAD        HEAD      /PICK UP HEAD TO USE
5567 1161      TAD        K0777     /SEEK TO CYLINDER 777
5570 4462      SEEKTO
5571 5366      JMP        .-3      /TRY AGAIN ON FAILURE
5572 1140      TAD        HEAD      /SEEK TO HEAD, CYLNR
5573 1136      TAD        CYLNR
5574 4462      SEEKTO
5575 5366      JMP        .-7      /TRY AGAIN ON FAILURE
5576 5764      JMP I      XSKHCY

5577 4575
5600 5600      PAGE

      /PRINT THE OCTAL NUMBER IN AC 9 THRU 11
5600 0000      XPRNT1, 0          /CALL BY "JMS  XPRNT1"
5601 0206      AND        K7
5602 1205      TAD        K260
5603 4513      TYPE
5604 5600      JMP I      XPRNT1
5605 0260      K260,      0260
5606 0007      K7,        7

      /ERRHAN
      /
      /THIS SUBROUTINE HANDLES POSSIBLE DRIVE ERRORS. (E.G. AFTER ERROR FLAG
      /SET ON READ OR WRITE, OR DRIVE NOT READY AFTER SEEK.)
      /IT IS POSSIBLE THAT THE DRIVE WILL BE DROPPED IF THE ERROR IS
      /FATAL OR WILL NOT RESET.

```

HP 001

```

5607 0000 /XERRHA, 0
5610 4460 GETSTA /GET THE DRIVE STATUS
5611 1120 TAD DATA2 /CHECK THE STATUS FROM THE DRIVE
5612 0377 AND (TIMEOUT) /FOR SEEK TIME-OUT ERROR
5613 7650 SNA CLA /SKIP IF SEEK TIME-OUT ERROR
5614 5225 JMP YNCONB
5615 4457 RESET /RESET THE DRIVE
5616 4355 JMS WAT300 /WAIT 300 MILLISECONDS
5617 4460 GETSTA /GET DRIVE'S STATUS
5620 1120 TAD DATA2 /CHECK SEEK TIMEOUT
5621 0377 AND (TIMEOUT)
5622 7640 SZA CLA /SKIP IF CLEARED
5623 5344 JMP YNCNTR /PRINT "CAN'T RESET" MESSAGE AND DROP DRIVE
5624 5332 JMP YNEXIT /EXIT
5625 1120 YNCONB, TAD DATA2 /CHECK FOR SPEED ERROR
5626 0376 AND (SPUPTO)
5627 7650 SNA CLA /SKIP IF SPEED ERROR
5630 5267 JMP YNCONC
5631 4457 RESET /RESET THE DRIVE
5632 1117 TAD DATA1 /CHECK THE STATE BITS
5633 0206 AND K7
5634 7650 SNA CLA /SKIP IF NOT LOAD STATE
5635 5244 JMP .+7
5636 4775 JMS WAIT1 /WAIT 1 MS
5637 4433 RRER /CHECK IF DRIVE ERROR STILL SET
5640 7012 RTR
5641 7630 SZL CLA /SKIP IF STILL SET
5642 5332 JMP YNEXIT /EXIT
5643 5344 JMP YNCNTR /CAN'T RESET ERROR
5644 4355 JMS WAT300 /WAIT 300 MS
5645 4460 GETSTA /GET STATUS
5646 1117 TAD DATA1 /CHECK STATE BITS
5647 0206 AND K7
5650 7650 SNA CLA /SKIP IF NO LONGER IN LOAD STATE
5651 5347 JMP YNLOAD /DROP DRIVE
5652 4466 SETTIM /INIT REAL TIME CLOCK ROUTINE

DECIMAL
5653 0140 -4000 /40 SECONDS
5654 7014 -500

OCTAL
5655 4447 CONSOL
5656 4433 RRER /CHECK FOR READY
5657 7010 RAR
5660 7630 SZL CLA /SKIP IF NOT READY
5661 5265 JMP .+4 /GET OUT OF LOOP
5662 4467 TIMCHK /CHECK TIME PASSAGE
5663 5255 JMP .-6 /WAIT FOR READY
5664 5352 JMP YNCHRY /CAN'T MAKE READY--DROP DRIVE
5665 4457 RESET /RESET THE DRIVE
5666 5332 JMP YNEXIT /LET'S EXIT
5667 1120 YNCONC, TAD DATA2 /CHECK FOR HEAD CURRENT ERROR
5670 0374 AND (HEDCUR)
5671 7640 SZA CLA /SKIP IF NOT HEAD CURRENT ERROR
5672 5341 JMP FATAL /FATAL ERROR--DROP DRIVE

```

5673	1120	TAD	DATA2	/CHECK FOR WRITE DATA ERROR
5674	0373	AND	(WRDERR	
5675	7650	SNA CLA		/SKIP IF SET
5676	5316	JMP	YNCOND	
5677	4355	JMS	WAT300	/WAIT 300 MS
5700	4457	RESET		/RESET THE DRIVE
5701	4466	SETTIM		/SETUP FOR 40 SECOND WAIT
5702	0140	DECIMAL	-4000	
5703	7014		-500	
5704	4447	OCTAL	CONSOL	/ALLOW CONSOL INPUT
5705	4433		RRER	/CHECK FOR READY
5706	7010		RAR	
5707	7630		SZL CLA	/SKIP IF NOT READY
5710	5314		JMP .+4	/GET OUT OF LOOP
5711	4467		TIMCHK	/SKIP IF TIME IS UP
5712	5304		JMP .-6	/KEEP WAITING
5713	5352		JMP YNCMRY	/DROP DRIVE
5714	4457		RESET	/RESET THE DRIVE
5715	5332		JMP YNEXIT	/EXIT
5716	1120	YNCOND,	TAD DATA2	/CHECK FOR WRITE GATE ERROR
5717	0372		AND (WRGATE	
5720	7650		SNA CLA	/SKIP IF SET
5721	5330		JMP YNCONE	
5722	4457		RESET	/RESET THE DRIVE
5723	4460		GETSTA	/GET STATUS
5724	1120		TAD DATA2	/CHECK IF WRITE GATE CLEARED
5725	0372		AND (WRGATE	
5726	7640		SZA CLA	/SKIP IF IT DID
5727	5344		JMP YNCNTR	/CAN'T RESET--DROP DRIVE
5730	4457	YNONE,	RESET	/RESET DRIVE
5731	4775		JMS WAIT1	/WAIT 1 MS AFTER RESET
5732	4433	YNEXIT,	RRER	/BEFORE LEAVING, CHECK DRIVE IS
5733	7012		RTR	/READY AND NO ERROR
5734	7500		SMA	/SKIP IF DRIVE IS READY
5735	5352		JMP YNCMRY	
5736	7630		SZL CLA	/SKIP IF NO DRIVE ERROR
5737	5344		JMP YNCNTR	
5740	5607		JMP I XERRHA	/RETURN
5741	4506	FATAL,	MESSAGE	/"? FATAL ERROR -"
5742	1143		FATLER	
5743	5771		JMP DROPDR	/NON-STRUCTURED GOTO--DROP THIS DRIVE
5744	4506	YNCNTR,	MESSAGE	/"? CAN'T RESET ERROR-"
5745	1154		CANTRS	
5746	5771		JMP DROPDR	
5747	4506	YNLOAD,	MESSAGE	/"? DRIVE IN LOAD STATE-"
5750	1170		DRNLOD	
5751	5771		JMP DROPDR	
5752	4506	YNCMRY,	MESSAGE	/"? CAN'T MAKE READY -"
5753	1205		CNMRYD	
5754	5771		JMP DROPDR	

/SUBROUTINE WAITS 300 MS

```

5755 0000 WAT300, 0
5756 4466 SETTIM /SET UP TIMING ROUTINE FOR 300 MS
5757 7742 -36
5760 7230 -550
5761 4447 CONSOL /ALLOW KEYBOARD INPUT
5762 4467 TIMCHK /SKIP IF TIME IS UP
5763 5361 JMP -2
5764 5755 JMP I WAT300 /RETURN

5771 4161
5772 0004
5773 0200
5774 0100
5775 5270
5776 0010
5777 0020
6000 0000 PAGE
VT278, 0
6001 6031 KSF
6002 6030 KCF
6003 5600 JMP I VT278

/JMS GET4
/
/GET FOUR 8-BIT WORD FROM BUFFER (POINTED TO BY AUTO12) IN FIELD 1
/INTO DATA1-DATA4.
/
6004 0000 GET4, 0
6005 7300 CLA CLL /ADD CLEAR THE LINK FOR SAFTY IN FILLUP ROUTINE HP 001

6006 6211 CDF 10 /CHANGE TO FIELD OF BUFFER
6007 1412 TAD I AUTO12 /PICK UP THE WORD
6010 0163 AND K377 /MASK THE GARBAGE
6011 3117 DCA DATA1 /SAVE THE WORD
6012 1412 TAD I AUTO12 /PICK UP THE WORD
6013 0163 AND K377 /MASK THE GARBAGE
6014 3120 DCA DATA2 /SAVE THE WORD
6015 1412 TAD I AUTO12 /PICK UP THE WORD
6016 0163 AND K377 /MASK THE GARBAGE
6017 3121 DCA DATA3 /SAVE THE WORD
6020 1412 TAD I AUTO12 /PICK UP THE WORD
6021 6201 CDF 0 /BACK TO THIS DATA FIELD
6022 0163 AND K377 /MASK THE GARBAGE
6023 3122 DCA DATA4 /SAVE THE WORD
6024 5604 JMP I GET4 /RETURN

/JMS FRCHNF
/
/FORCE HEADER NOT FOUND. ISSUE A 1 WORD READ TO POSITION SPECIFIED BY
/LOCATIONS SECTOR AND CYLNR. VERIFY BOTH OPI AND HNF ARE
/SET IN THE ERROR REGISTER. IF EITHER IS NOT SET, REPORT AN ERROR.
/
6025 0000 FRCHNF, 0
6026 7240 STA /WORD COUNT OF 1
6027 4432 RLWC

```

6030	1145	TAD	PRDBUF	/POINT CONTROLLER TO READ BUFFER IN CASE
6031	4426	RLMA		/ THE READ WORKS
6032	1136	TAD	CYLNDR	/SET UP COMMAND A WITH CYLINDER AND HEAD
6033	4427	RLCA		
6034	1141	TAD	SECTOR	/SET UP SA
6035	4431	RLSA		
6036	1116	TAD	DRVNUM	/SET UP DRIVE SELECT
6037	7102	CLL BSW		
6040	1167	TAD	K1016	/ADD IN 8 BIT READ TO FIELD 1
6041	4430	RLCB		/ISSUE THE COMMAND
6042	1777	TAD	TOCK	/COMPENSATE APT CLOCK FOR OPI TIME
6043	1176	TAD	K1000	
6044	7430	SZL		/SKIP IF NO OVERFLOW
6045	7240	STA		/BACK TIMER UP
6046	3777	DCA	TOCK	/SAVE NEW TIMER VALUE
6047	4425	RLSD		/WAIT FOR DONE
6050	4470	JMPPM1		
6051	4433	RRER		/READ THE ERROR REG
6052	3117	DCA	DATA1	/SAVE IT
6053	1117	TAD	DATA1	
6054	7006	RTL		/GET OPI INTO LINK, HNF INTO SIGN BIT
6055	7060	CML CMA		/COMPLEMENT BITS SO EITHER CAN CAUSE SKIP
6056	7720	SNL SMA CLA		/SKIP IF EITHER OPI OR HNF WAS NOT SET
6057	5273	JMP	FRCRET	/OK--EXIT
6060	7240	STA		/PICK UP THE CALL PC
6061	1225	TAD	FRCHNF	
6062	3123	DCA	DATA5	/SAVE IT FOR ERROR TYPEOUT
6063	1136	TAD	CYLNDR	/SAVE CA FOR ERROR TYPEOUT
6064	3120	DCA	DATA2	
6065	1141	TAD	SECTOR	/SAVE SA FOR ERROR TYPEOUT
6066	3121	DCA	DATA3	
6067	1140	TAD	HEAD	/SAVE ACTUAL HEAD POSITION
6070	3122	DCA	DATA4	
6071	4446	ERROR		/COULD NOT FORCE OPI-HNF ON READ
6072	2414	NOHNF		/PC DRV NO. ER CA SA HED-CYL CALLPC
6073	5625	FRCRET, JMP I	FRCHNF	/RETURN
/FILBUF				
/				
/FILL THE BUFFER (POINTED TO BY ARGUMENT FOLLOWING CALL) IN FIELD 1 WITH				
/8-BIT PATTERN POINTED TO BY POINTER IN PATRN.				
/				
6074	0000	XFILBU, 0		
6075	7240	STA		
6076	1674	TAD I	XFILBU	/SET UP POINTER TO THE BUFFER
6077	3010	DCA	AUTO10	
6100	2274	ISZ	XFILBU	/SKIP OVER ARGUMENT
6101	1171	TAD	M10	/SET UP COUNTER FOR OUTER LOOP
6102	3132	DCA	TEMP2	/ (8*32=256)
6103	7240	FILBUL, STA		/SET UP POINTER TO PATTERN
6104	1560	TAD I	PATRN	
6105	3013	DCA	AUTO13	
6106	6102	SPL		/SKIP ON POWER LOW
6107	7410	SKP		
6110	4000	JMS	0	/POWER FAIL INTERRUPT

6111	1164	TAD	M40	/SET UP COUNTER FOR INNER LOOP
6112	3131	DCA	TEMP1	/ (NUMBER OF WORDS IN PATTERN)
6113	6211	CDF	10	
6114	1413	TAD I	AUTO13	/PICK UP A WORD OF PATTERN
6115	3410	DCA I	AUTO10	/PLACE IN BUFFER
6116	2131	ISZ	TEMP1	/RUN OUT OF PATTERN YET?
6117	5314	JMP	.-3	/NO--GO BACK
6120	6201	CDF	0	
6121	2132	ISZ	TEMP2	/YES--REPEATED PROCESS 10 TIMES?
6122	5303	JMP	FILBUL	/NO--LOOP BACK
6123	5674	JMP I	XFILBU	/YES--RETURN

/CMPPAT

/

/SUBROUTINE COMPARES A SECTOR OF 8-BIT DATA (POINTED TO BY ARGUMENT FOLLOWING /CALL) TO PATTERN POINTED TO BY ENTRY IN TABLE (POINTED TO BY PATRN). IF A DATA ERROR /OCCURS, THE ROUTINE SKIPS ON RETURN AND THE BAD WORD OF DATA IS POINTED /TO BY AUTO14, AND THE GOOD WORD OF DATA IS POINTED TO BY AUTO15. (NOTE: /THESE CANNOT BE USED DIRECTLY TO FETCH THE OPERANDS SINCE THEY ARE NOT /POINTING TO 1 BEFORE THE DATA.) IF NO DATA ERROR, A NORMAL RETURN IS MADE.

6124	0000	XCMPPA, 0		
6125	7240	STA		
6126	1724	TAD I	XCMPPA	/SET UP POINTER TO THE BUFFER TO CHECK
6127	3014	DCA	AUTO14	
6130	2324	ISZ	XCMPPA	/SKIP OVER ARGUMENT
6131	1171	TAD	M10	/SET UP COUNTER FOR 8 TIMES TO COMPARE
6132	3132	DCA	TEMP2	/ THE 32 WORD PATTERN SET
6133	1164	CMPLPO, TAD	M40	/SET UP COUNTER FOR 32 WORDS OF PATTERN
6134	3131	DCA	TEMP1	
6135	7240	STA		/SET UP POINTER TO THE PATTERN
6136	1560	TAD I	PATRN	
6137	3015	DCA	AUTO15	
6140	6102	SPL		/SKIP ON POWER LOW
6141	7410	SKP		
6142	4000	JMS	0	/POWER FAIL INTERRUPT
6143	6211	CDF	10	
6144	1414	CMPLPI, TAD I	AUTO14	/PICK UP WORD FROM BUFFER
6145	0163	AND	K377	/MASK GARBAGE
6146	7041	CIA		/COMPARE TO EXPECTED WORD FROM PATTERN
6147	1415	TAD I	AUTO15	
6150	7640	SZA CLA		/SKIP IF EQUAL
6151	5366	JMP	CMPEER	
6152	2131	ISZ	TEMP1	/CHECKED ALL 32 WORDS OF PATTERN
6153	5344	JMP	CMPLPI	/NO
6154	6201	CDF	0	
6155	2132	ISZ	TEMP2	/CHECK ENTIRE BUFFER (32 WORDS 8 TIMES)?
6156	5333	JMP	CMPLPO	/NO
6157	7300	CLA CLL		/CLEAR LINK FOR OVERFLOW TEST
6160	1371	TAD	K60	/ADD 60 TO APT TIMER TO COMPENSATE FOR
6161	1777	TAD	TOCK	/ COMPARE TIME
6162	7430	SZL		/SKIP IF NO OVERFLOW
6163	7240	STA		/BACK TIMER UP TO -1
6164	3777	DCA	TOCK	/SAVE NEW TOCK VALUE
6165	5370	JMP	.-3	/RETURN

```

6166 6201 CMPERR, CDF      0
6167 2324 ISZ      XCMPPA  /SKIP ON RETURN
6170 5724 JMP I      XCMPPA
/
6171 0060 K60,      60
6177 4575
6200 PAGE
/WRIT8
/
/ISSUE 8 BIT WRITE AND WAIT FOR DONE. COMPENSATE FOR APT TIMING. CHECK
/FOR ERROR FLAG AND REPORT. SKIP IF ERROR FLAG WAS NOT SET.
/
6200 0000 XWRIT8, 0
6201 7300 CLA CLL
6202 1116 TAD      DRVNUM      /SET UP DRIVE SELECT BITS
6203 7002 BSW
6204 1166 TAD      K1015      /ADD IN 8-BIT WRITE COMMAND
6205 4430 RLCB      /ISSUE IT
6206 4503 APTCOM      /COMPENSATE APT TIMER FOR WRITE TIME
6207 4425 RLSD      /WAIT FOR DONE
6210 4470 JMPPM1
6211 4441 RLSE
6212 5217 JMP      WR8RET      /SKIP IF ERROR
6213 7240 STA      /OK--SKIP ON RETURN
6214 1200 TAD      XWRIT8      /COMPUTE CALL PC
6215 4305 JMS      REPERR
6216 7410 SKP
6217 2200 WR8RET, ISZ      XWRIT8      /REPORT THE ERROR
6220 5600 JMP I      XWRIT8      /DON'T SKIP ON RETURN

/READ8
/
/ISSUE 8 BIT READ AND WAIT FOR DONE. COMPENSATE FOR APT TIMING. CHECK
/FOR ERROR FLAG AND REPORT. SKIP IF ERROR FLAG WAS NOT SET.
/
6221 0000 XREAD8, 0
6222 7300 CLA CLL
6223 1116 TAD      DRVNUM      /SET UP DRIVE SELECT BITS
6224 7002 BSW
6225 1167 TAD      K1016      /ADD IN 8-BIT READ COMMAND
6226 4430 RLCB      /ISSUE IT
6227 4503 APTCOM      /COMPENSATE APT TIMER FOR READ TIME
6230 4425 RLSD      /WAIT FOR DONE
6231 4470 JMPPM1
6232 4441 RLSE
6233 5240 JMP      RD8RET      /SKIP IF ERROR
6234 7240 STA      /OK--SKIP ON RETURN
6235 1221 TAD      XREAD8      /COMPUTE CALL PC
6236 4305 JMS      REPERR
6237 7410 SKP
6240 2221 RD8RET, ISZ      XREAD8      /REPORT THE ERROR
6241 5621 JMP I      XREAD8      /DON'T SKIP ON RETURN

```

```

/WRIT12
/
/ISSUE 12 BIT WRITE AND WAIT FOR DONE. COMPENSATE FOR APT TIMING. CHECK
/FOR ERROR FLAG AND REPORT. SKIP IF ERROR FLAG WAS NOT SET.
/
6242 0000 XWRIT1, 0
6243 7300 CLA CLL
6244 1116 TAD DRVNUM /SET UP DRIVE SELECT BITS
6245 7002 BSW
6246 1777' TAD K15 /ADD IN 12-BIT WRITE COMMAND
6247 4430 RLCB /ISSUE IT
6250 4503 APTCOM /COMPENSATE APT TIMER FOR WRITE TIME
6251 4425 RLSD /WAIT FOR DONE
6252 4470 JMPPM1
6253 4441 RLSE /SKIP IF ERROR
6254 5261 JMP WR12RE /OK--SKIP ON RETURN
6255 7240 STA /COMPUTE CALL PC
6256 1242 TAD XWRIT1
6257 4305 JMS REPERR /REPORT THE ERROR
6260 7410 SKP /DON'T SKIP ON RETURN
6261 2242 WR12RE, ISZ XWRIT1
6262 5642 JMP I XWRIT1

/READ12
/
/ISSUE 12 BIT READ AND WAIT FOR DONE. COMPENSATE FOR APT TIMING. CHECK
/FOR ERROR FLAG AND REPORT. SKIP IF ERROR FLAG WAS NOT SET.
/
6263 0000 XREAD1, 0
6264 7300 CLA CLL
6265 1116 TAD DRVNUM /SET UP DRIVE SELECT BITS
6266 7002 BSW
6267 1304 TAD K16 /ADD IN 12-BIT READ COMMAND
6270 4430 RLCB /ISSUE IT
6271 4503 APTCOM /COMPENSATE APT TIMER FOR READ TIME
6272 4425 RLSD /WAIT FOR DONE
6273 4470 JMPPM1
6274 4441 RLSE /SKIP IF ERROR
6275 5302 JMP RD12RE /OK--SKIP ON RETURN
6276 7240 STA /COMPUTE CALL PC
6277 1263 TAD XREAD1
6300 4305 JMS REPERR /REPORT THE ERROR
6301 7410 SKP /DON'T SKIP ON RETURN
6302 2263 RD12RE, ISZ XREAD1
6303 5663 JMP I XREAD1

/
6304 0016 K16, 16

/JMS REPERR
/
/REPORT THE ERROR FLAG SET ERROR. CALLED FROM ONE OF THE READ/WRITE
/ROUTINES WITH THE CALL PC OF THAT ROUTINE IN THE AC AS AN ARGUMENT.
/
6305 0000 REPERR, 0
6306 3125 DCA DATA7 /SAVE THE CALL PC ARGUMENT

```

6307	4433	RRER		/READ AND SAVE ERROR REG.
6310	3121	DCA	DATA3	
6311	4436	RRCB		/READ AND SAVE COMMAND B
6312	3122	DCA	DATA4	
6313	4435	RRCA		/DITTO COMMAND A
6314	3123	DCA	DATA5	
6315	4437	RRSA		
6316	3124	DCA	DATA6	
6317	4460	GETSTA		/GET AND SAVE DRIVE STATUS
6320	4446	ERROR		/ERROR FLAG SET
6321	2410	EF7		/PC DRVNO. STATUS ER CB CA SA CALLPC
6322	4461	ERRHAN		
6323	5705	JMP I	REPERR	/RETURN

```

/JMS TRK1GO
/
/SUBROUTINE RETURNS WITH FIRST GOOD SECTOR ON CYL 0, TRACK 1 IN AC 0:5
/
TRK1GO, 0
  CLA
  TAD BSVALD /IS THE BAD SECTOR FILE VALID?
  SNA CLA /SKIP IF YES
  JMP TRK1RT /RETURN WITH SECTOR 0
  DCA TEMP1 /START OUT CHECKING IF SECTOR 0 IS LISTED
  TAD PBSF /SET UP POINTER TO BSF
  DCA AUTO10
  TRKLUP, CDF 10
  TAD I AUTO10 /PICK UP A CYLINDER ENTRY
  CDF 0
  SPA
  JMP TRRTSE /SKIP IF NOT END OF FILE INDICATOR
  SNA CLA /RETURN WITH TEMP1 VALUE
  JMP .+3 /SKIP IF NOT CYL 0 THAT IS LISTED
  ISZ AUTO10 /SKIP OVER TRACK-SECTOR WORD
  JMP TRKLUP /CHECK NEXT ENTRY
  CDF 10
  TAD I AUTO10 /PICK UP TRK-SECTOR ENTRY
  CDF 0
  DCA TEMP2 /SAVE IT
  TAD TEMP2
  SMA CLA
  JMP TRKLUP /SKIP IF TRACK 1
  STA CLL RAR /NOT TRACK 1--CHECK NEXT ENTRY
  AND TEMP2 /MASK OUT TRACK BIT
  CIA
  TAD TEMP1 /COMPARE THE SECTOR WITH SECTOR
  SZA CLA / BEING CHECKED
  JMP TRKLUP /SKIP IF SECTOR IN TEMP1 IS LISTED
  ISZ TEMP1
  JMP TRKLUP-2 /CHECK NEXT SECTOR FOR LISTING
  TRRTSE, CLA /RECHECK ENTIRE BSF FOR SECTOR
  TAD TEMP1
  BSW
  TRK1RT, JMP I TRK1GO
/

```

6370 7777 PBSF, BADSEC-1

6377 3145
6400 PAGE

/WRITRK

/

/WRITE AN ENTIRE TRACK WITH THE 2K OF DATA IN THE BUFFER REPEATED FIVE
/TIMES (8 BIT MODE). 8 SECTORS ARE WRITTEN (2K), 8 SECTORS ARE SKIPPED, ETC.
/UNTIL TRACK IS WRITTEN. THIS REQUIRES 1 AND 4/5 SPINS (PLUS ROTATIONAL
/LATENCY PRIOR TO FIRST TRANSFER) TO WRITE THE TRACK. ROTATIONAL LATENCY
/PRIOR TO FIRST WRITE IS MINIMIZED (GURANTEED LESS THAN OR EQUAL TO 1/5 SPIN
/OR 5 MS) BY READING A HEADER AND BEGINNING FIRST WRITE AT SECTOR A
/MULTIPLE OF 8 HIGHER. WRITE ERRORS ARE REPORTED (BY WRIT8). THE SUBROUTINE SKIPS
/IF NO ERRORS.

6400 0000

XWRTR, 0

6401 4465

REDHDR

/READ A HEADER

6402 4440

RRSI

/GET HEADER WORD 1

6403 0241

AND K70

/SAVE ONLY HIGH ORDER BITS OF SECTOR

6404 1171

TAD M10

/SUBTRACT 8 TO LOOK LIKE A SECTOR JUST WRITTEN

6405 3141

DCA SECTOR

/SAVE "SECTOR WRITTEN LAST TIME"

6406 1175

TAD M5

/SET UP COUNTER FOR NUMBER OF WRITES

6407 3235

DCA WRTCNT

6410 1141

WRTLUP, TAD SECTOR

/GET STARTING SECTOR OF LAST WRITE

6411 1237

TAD M30

/ADD 20 (OCTAL) TO SECTOR AND COMPARE TO

6412 7510

SPA

/ILLEGAL SECTOR 50 IN ONE EASY STEP

6413 1240

TAD K50

/ADD 50 BACK IN IF SECTOR IS NEGATIVE

6414 3141

DCA SECTOR

/SAVE SECTOR THAT WRITE STARTED AT THIS TIME

6415 1141

TAD SECTOR

/SAVE SA

6416 7002

BSW

6417 4431

RLSA

6420 1236

TAD PNTRBF

/SET UP MA

6421 4426

RLMA

6422 7330

CLA STL RAR

/WORD COUNT OF 2K

6423 4432

RLWC

6424 1140

TAD HEAD

/SET UP HEAD AND CYLINDER IN CA

6425 1136

TAD CYLNR

6426 4427

RLCA

6427 4477

WRIT8

/ISSUE WRITE

6430 5234

JMP WTRRET

/MAKE ERROR RETURN ON ERROR

6431 2235

ISZ WRTCNT

/DONE 5 WRITES (WHOLE TRACK)?

6432 5210

JMP WRTLUP

/NO--DO IT AGAIN

6433 2200

ISZ XWRTR

/YES--AND NO ERRORS SO SKIP ON RETURN

6434 5600

WTRRET, JMP I XWRTR

/RETURN

/

6435 0000

WRTCNT, 0

6436 3600

PNTRBF, REDBUF

/POINTER TO 2K BUFFER

6437 7750

M30, -30

6440 0050

K50, 50

6441 0070

K70, 70

/CHKTRK

/

/THIS SUBROUTINE VERIFIES THE DATA PATTERNS ON AN ENTIRE TRACK, DATA PATTERNS 1-8
/ARE EXPECTED, REPEATED 5 TIMES STARTING AT SECTOR 0. 8 SECTORS (2K) ARE READ

/AT ONE TIME, AND EACH PORTION OF THE BUFFER (1 SECTOR) IS COMPARED TO ITS EXPECTED
 /PATTERN. THE SUBROUTINE READF LEAVES THE STARTING SECTOR OF THE TRANSFER DIVIDED
 /BY 8 IN LOCATION SECTOR. (I.E. A NUMBER FROM 0-4.) THIS IS USED FOR REPORTING
 /THE FAILING SECTOR.

6442	0000	XCHKTR, 0		
6443	4777	JMS	INITRF	/INITIALIZE THE READF SUBROUTINE (NO SECTORS READ YET)
6444	1175	TAD	M5	/SET UP COUNTER FOR NUMBER OF READS
6445	3327	DCA	CHKCN1	
6446	4776	ANOTHF, JMS	READF	/READ ANOTHER FIFTH OF THE TRACK
6447	5326	JMP	CHTRET	/ERROR--RETURN
6450	1157	TAD	PATPTR	/INITIALIZE POINTER TO CHECK PATTERN 1
6451	3160	DCA	PATRN	
6452	1145	TAD	PRDBUF	/INITIALIZE THE FIRST BUFFER TO CHECK
6453	3257	DCA	ANOTBF+1	
6454	1171	TAD	M10	/SET UP COUNTER FOR NUMBER OF SECTORS (BUFFERS) TO CHECK
6455	3330	DCA	CHKCN2	
6456	4473	ANOTBF, CMPPAT		/COMPARE A PATTERN IN A BUFFER
6457	3600	REDBUF		/THIS GETS CHANGED TO POINT TO NEXT BUFFER EACH TIME THRU LOOP
6460	7410	SKP		/OK--CONTINUE
6461	5274	JMP	CHTRER	/GO REPORT DATA ERROR
6462	1775	TAD	K400	/BUMP BUFFER POINTER UP TO POINT TO
6463	1257	TAD	ANOTBF+1	/ THE NEXT BUFFER TO CHECK
6464	3257	DCA	ANOTBF+1	
6465	2160	ISZ	PATRN	/GO TO NEXT PATTERN FOR NEXT BUFFER
6466	2330	ISZ	CHKCN2	/DONE ALL 8 BUFFERS?
6467	5256	JMP	ANOTBF	/NO--GO BACK AND CHECK ANOTHER
6470	2327	ISZ	CHKCN1	/YES--DONE 5 READS (ENTIRE TRACK)?
6471	5246	JMP	ANOTHF	/NO--DO ANOTHER
6472	2242	ISZ	XCHKTR	/YES--SKIP ON RETURN
6473	5326	JMP	CHTRET	/RETURN
6474	1141	CHTRER, TAD	SECTOR	/GET STARTING SECTOR/8
6475	7106	CLL	RTL	
6476	7104	CLL	RAL	
6477	1330	TAD	CHKCN2	/NOW WE HAVE STARTING SECTOR OF TRANSFER
6500	1774	TAD	K10	/COMPUTE FAILING SECTOR FROM VALUE
6501	3123	DCA	DATA5	/ LEFT IN COUNTER
6502	1014	TAD	AUTO14	/SAVE FAILING SECTOR FOR ERROR TYPEOUT
6503	3121	DCA	DATA3	/GET FAILING ADDRESS
6504	1015	TAD	AUTO15	/SAVE FOR TYPEOUT
6505	3131	DCA	TEMP1	/PICK UP POINTER TO GOOD DATA
6506	6211	CDF	10	
6507	1521	TAD	I DATA3	/PICK UP BAD DATA
6510	3117	DCA	DATA1	/SAVE IT
6511	1531	TAD	I TEMP1	/GET GOOD DATA
6512	6201	CDF	0	
6513	3120	DCA	DATA2	/SAVE IT
6514	1140	TAD	HEAD	/COMPUTE CA AND SAVE IT
6515	1136	TAD	CYLNR	
6516	3123	DCA	DATA5	
6517	4436	RRCB		/SAVE CB
6520	3124	DCA	DATA6	
6521	7240	STA		/COMPUTE CALL PC
6522	1242	TAD	XCHKTR	
6523	3125	DCA	DATA7	

```

6524 4446          ERROR          /DATA ERROR
6525 3245          DATAE4        /PC DRVNO BAD GOOD MA CA SECTOR CB CALLPC
6526 5642          CHTRET, JMP I   XCHKTR  /RETURN
/
6527 0000          CHKCN1, 0
6530 0000          CHKCN2, 0

/ JMS CKTRK2
/
/ THIS SUBROUTINE CHECKS AN ENTIRE TRACK (HEAD,CYLNR) AGAINST A SINGLE DATA
/ PATTERN (PATRN). ONE FIFTH OF THE TRACK (8 SECTORS) IS READ AT ONE
/ TIME, AS IS DONE IN CHKTRK. IF NO ERROR, A NORMAL RETURN IS MADE. ERRORS
/ ARE NOT REPORTED, BUT THE ROUTINE WILL SKIP ON RETURN IF AN ERROR.
/ ON ERROR RETURN, GOOD AND BAD DATA ARE POINTED TO BY AUTO14,15 (AS LEFT
/ BY CMPPAT), AND THE FAILING SECTOR IS LEFT IN SECTOR.
/
6531 0000          CKTRK2, 0
6532 4777'         JMS          INITRF      /INITIALIZE THE READF SUBROUTINE (NO SECTORS READ YET)
6533 1175          TAD          M5          /SET UP COUNTER FOR NUMBER OF READS
6534 3327          DCA          CHKCN1
6535 4776'         ANOTH8, JMS          READF      /READ ANOTHER 8 SECTORS
6536 5366          JMP          CT2RET      /ERROR--RETURN
6537 1145          TAD          PROBUF      /INITIALIZE THE FIRST BUFFER TO CHECK
6540 3344          DCA          BUFAGN+1
6541 1171          TAD          M10
6542 3330          DCA          CHKCN2      /SET UP COUNTER FOR NUMBER OF SECTORS (BUFFERS) TO CHECK
6543 4473          BUFAGN, CMPPAT
6544 3600          REDBUF
6545 7410          SKP
6546 5357          JMP          CHTRE2      /GO COMPUTE FAILING SECTOR
6547 1775'         TAD          K400        /BUMP BUFFER POINTER UP TO POINT TO
6550 1344          TAD          BUFAGN+1    / THE NEXT BUFFER TO CHECK
6551 3344          DCA          BUFAGN+1
6552 2330          ISZ          CHKCN2      /DONE ALL 8 BUFFERS?
6553 5343          JMP          BUFAGN      /NO--GO BACK AND CHECK ANOTHER
6554 2327          ISZ          CHKCN1      /YES--DONE 5 READS (ENTIRE TRACK)?
6555 5335          JMP          ANOTH8      /NO--DO ANOTHER
6556 5366          JMP          CT2RET      /RETURN
6557 1141          CHTRE2, TAD          SECTOR /GET STARTING SECTOR/8
6560 7106          CLL RTL
6561 7104          CLL RAL
6562 1330          TAD          CHKCN2      /NOW WE HAVE STARTING SECTOR OF TRANSFER
6563 1774'         TAD          K10        /COMPUTE FAILING SECTOR FROM VALUE
6564 3141          DCA          SECTOR     / LEFT IN COUNTER
6565 2331          ISZ          CKTRK2      /SAVE FAILING SECTOR
6566 5731          CT2RET, JMP I   CKTRK2   /SKIP ON RETURN TO FLAG ERROR
/RETURN

6574 4553
6575 3751
6576 6612
6577 6600
6600

```

PAGE

/JMS INITRF

/

```

/THIS SUBROUTINE INITIALIZES THE READF ROUTINE TO FLAG THAT NO SECTORS
/HAVE YET BEEN READ. (THIS IS DONE BY CLEARING ENTRIES IN SECTOR TABLE.)
/
6600 0000      INITRF, 0
6601 7240          STA
6602 1256          TAD      PSECTB      /PICK UP POINTER TO SECTOR TABLE-1
6603 3010          DCA      AUTO10
6604 3410          DCA I    AUTO10      /CLEAR FIRST ENTRY (FLAG SECTORS 0-7 NOT READ)
6605 3410          DCA I    AUTO10      /FLAG SECTORS 10-17 NOT YET READ
6606 3410          DCA I    AUTO10      /ETC.
6607 3410          DCA I    AUTO10
6610 3410          DCA I    AUTO10
6611 5600          JMP I    INITRF      /RETURN

/JMS READF
/
/READ THE NEXT FIFTH OF A TRACK (8 SECTORS) THAT HAS NOT YET BEEN READ (AS
/SPECIFIED BY SECTOR TABLE). IF CALLED WHEN ALL ENTRIES IN SECTAB AR
/NON-ZERO, THE ROUTINE WILL HANG! STARTING SECTOR OF READ DIVIDED BY 8 IS
/STORED IN "SECTOR". ERRORS ARE REPORTED BY READ8. IF NO ERRORS, THE ROUTINE
/SKIPS ON RETURN.
/
6612 0000      READF, 0
6613 4465          REDHDR
6614 4440          RRSI
6615 0777'        AND      K70
6616 7112          CLL RTR
6617 7010          RAR
6620 3141          DCA      SECTOR
6621 1141          NXTGRP, TAD      SECTOR
6622 1776'        TAD      M4
6623 7510          SPA
6624 1255          TAD      K5
6625 3141          DCA      SECTOR
6626 1141          TAD      SECTOR
6627 1256          TAD      PSECTB
6630 3131          DCA      TEMP1
6631 1531          TAD I    TEMP1
6632 7640          SZA CLA
6633 5221          JMP      NXTGRP
6634 7240          STA
6635 3531          DCA I    TEMP1
6636 1145          TAD      PRDBUF
6637 4426          RLMA
6640 1141          TAD      SECTOR
6641 7112          CLL RTR
6642 7012          RTR
6643 4431          RLCA
6644 7330          CLA STL RAR
6645 4432          RLWC
6646 1140          TAD      HEAD
6647 1136          TAD      CYLNRD
6650 4427          RLCA
6651 4500          READ8
6652 7410          SKP

/READ A HEADER TO GET POSITION
/READ HEADER WORD 1
/SAVE ONLY HIGH ORDER BITS OF SECTOR
/DIVIDE BY 8 (PUT IN LOW BITS)

/SAVE THE SECTOR GROUP THE HEADS ARE IN
/GO TO NEXT HIGHER SECTOR GROUP FOR THE READ
/ BY ADDING 1 AND COMPARING TO 5 (ILLEGAL
/ GROUP) AT SAME TIME. IF 0, LEAVE AS IS.
/ IF NEGATIVE, ADD 5 BACK IN.
/SAVE SECTOR GROUP TO READ
/CHECK IF THE SECTOR GROUP WAS ALREADY READ
/ ON A PREVIOUS CALL. (INDEX INTO TABLE.)
/SAVE POINTER INTO TABLE
/PICK UP FLAG FROM TABLE
/SKIP IF NOT YET READ
/TRY TO START READ AT NEXT HIGHER SECTOR GROUP
/FLAG THAT THIS GROUP OF 8 SECTORS HAS BEEN READ

/SET UP MA

/PICK UP SECTOR GROUP TO READ
/MOVE SECTOR BITS INTO POSITION FOR
/ VALID STARTING SECTOR (AC0-2)

/2K WORD COUNT

/SET UP HEAD AND CYLINDER FOR CA

/ISSUE READ
/ERROR ON READ

```



```

6653 2212      ISZ      READF      /SKIP ON RETURN IF NO ERRORS
6654 5612      JMP I    READF
/
6655 0005      K5,      5
6656 7132      PSECTB, SECTAB

/ JMS FIL2KB
/
/ FILL THE ENTIRE 2K BUFFER WITH THE DATA PATTERN SPECIFIED BY TABLE ENTRY
/ POINTED TO BY PATRN. IT'S SIMPLER THAN IT SOUNDS.
/
6657 0000      FIL2KB, 0
6660 7200      CLA
6661 1171      TAD      M10          /SET UP COUNTER FOR NUMBER OF 256 WORD
6662 3275      DCA      BFCNT2      / BUFFERS TO FILL
6663 1145      TAD      PRDBUF      /SET UP STARTING BUFFER POINTER
6664 3266      DCA      BFLUP2+1
6665 4472      BFLUP2, FILBUF      /FILL A 256 WORD BUFFER WITH THE PATTERN
6666 0000      0                  /POINTER TO BUFFER (GETS MODIFIED)
6667 1266      TAD      BFLUP2+1    /ADD 256 MORE TO BUFFER POINTER
6670 1775      TAD      K400        / AND SAVE NEW POINTER TO NEXT BUFFER
6671 3266      DCA      BFLUP2+1
6672 2275      ISZ      BFCNT2      /DONE 8 BUFFERS (2K) ??
6673 5265      JMP      BFLUP2      /NO--GO BACK AND FILL ANOTHER
6674 5657      JMP I    FIL2KB      /YES--RETURN
/
6675 0000      BFCNT2, 0

/ APTCOM
/
/ COMPENSATE APT TIMER FOR A READ OR WRITE TIME, ASSUMING AVERAGE DISK LATENCY.
/
6676 0000      XAPTCO, 0
6677 7300      CLA CLL
6700 1774      TAD      TOCK          /GET APT TIMER
6701 1773      TAD      K20          /ADD COMPENSATION
6702 7430      SZL              /SKIP IF NO OVERFLOW
6703 7240      STA              /BACK TIMER UP TO 7777 ON OVERFLOW
6704 3774      DCA      TOCK          /SAVE NEW TIMER VALUE
6705 5676      JMP I    XAPTCO      /RETURN

/ CYLBAD
/
/ THIS SUBROUTINE CHECKS IF THE CYLINDER CONTAINED IN CYLNDR HAS ANY BAD SECTORS
/ IN THE BSF. IF THE CYLINDER HAS BAD SECTORS, A NORMAL RETURN IS MADE.
/ IF NO BAD SECTORS (OR BSF NOT VALID), THE ROUTINE SKIPS ON RETURN.
6706 0000      XCYLBA, 0
6707 7200      CLA
6710 1156      TAD      BSVALD      /IS BAD SECTOR FILE VALID?
6711 7650      SNA CLA
6712 5330      JMP      CYLGUD      /NO--ASSUME CYLINDER IS GOOD
6713 1333      TAD      PBSM1      /YES--SET UP POINTER TO BSF AREA
6714 3010      DCA      AUTO10
6715 6211      CBLOOP, CDF 10
6716 1410      TAD I    AUTO10      /PICK UP BSF CYLINDER ENTRY

```

```

6717 6201      CDF      0
6720 7510      SPA
6721 5330      JMP      CYLGUD      /SKIP IF NOT BSF TERMINATOR
6722 7041      CIA      /CYLINDER WAS NOT LISTED
6723 1136      TAD      CYLNDR      /COMPARE ENTRY TO VALUE IN CYLNDR
6724 7650      SNA CLA
6725 5331      JMP      CYLGUD+1      /SKIP IF NOT THE SAME
6726 2010      ISZ      AUTO10      /CYLINDER IS BAD
6727 5315      JMP      CBLOOP      /SKIP OVER TRACK AND SECTOR LISTING
6730 2306      CYLGUD, ISZ      XCYLBA      /CHECK NEXT ENTRY
6731 7200      CLA      /SKIP ON RETURN IF CYLINDER WAS GOOD
6732 5706      JMP I      XCYLBA      /RETURN WITH AC CLEAR
/
6733 7777      PB5M1, BADSEC-1
/
/ JMS BUFS2K
/
/ SET UP 2K BUFFER WITH 8 BIT PATTERNS 1 THRU 8.
/
6734 0000      BUFS2K, 0
6735 7200      CLA
6736 1157      TAD      PATPTR      /SET UP POINTER TO PATTERN POINTER TABLE
6737 3160      DCA      PATTRN
6740 1171      TAD      M10
6741 3355      DCA      BUFCNT      /SET UP COUNTER FOR 8 PATTERNS
6742 1145      TAD      PRDBUF
6743 3345      DCA      P2KBUF      /INITIALIZE POINTER TO BUFFER
6744 4472      LUP2K, FILBUF      /SAVE POINTER TO BUFFER TO FILL
6745 0000      P2KBUF, 0      /FILL UP 256 WORD BUFFER
6746 2160      ISZ      PATTRN      /POINTER TO BUFFER (GETS CHANGED)
6747 1345      TAD      P2KBUF      /GO TO NEXT PATTERN
6750 1775      TAD      K400      /ADD 256 TO BUFFER POINTER
6751 3345      DCA      P2KBUF      /SAVE POINTER TO NEXT BUFFER
6752 2355      ISZ      BUFCNT      /DONE ALL 8 BUFFERS AND PATTERNS?
6753 5344      JMP      LUP2K      /NO
6754 5734      JMP I      BUFS2K      /YES
/
6755 0000      BUFCNT, 0
/
/ JMS CMPWRD
/
/ COMPARE WORD IN AC UPON ENTRY (EXPECTED VALUE) TO WORD IN BUFFER POINTED
/ TO BY AUTO10 (ACTUAL VALUE). SAVE EXPECTED AND ACTUAL VALUES IN DATA1
/ AND DATA2 RESPECTIVELY. SKIP IF NO ERROR.
/
6756 0000      CMPWRD, 0
6757 3117      DCA      DATA1      /SAVE EXPECTED
6760 6211      CDF      10
6761 1410      TAD I      AUTO10      /GET ACTUAL
6762 6201      CDF      0
6763 3120      DCA      DATA2      /SAVE IT
6764 1117      TAD      DATA1      /COMPARE THEM
6765 7041      CIA
6766 1120      TAD      DATA2
6767 7650      SNA CLA      /SKIP IF NOT EQUAL

```

```

6770 2356      ISZ      CMPWRD      /OK--SKIP ON RETURN
6771 5756      JMP I    CMPWRD      /RETURN

6773 4443
6774 4575
6775 3751
6776 0371
6777 6441
7000 7000      PAGE

/DATERR
/
/REPORT A DATA ERROR.  BAD DATA IS POINTED TO BY AUTO14, GOOD DATA BY AUTO15.
/SECTOR IS IN SECTOR.
7000 0000      XDATER, 0
7001 4436      RRCB              /SAVE COMMAND B FOR ERROR TYPEOUT
7002 3124      DCA      DATA6
7003 1141      TAD      SECTUR   /SAVE ALL PERTINENT REGISTERS
7004 3123      DCA      DATA5
7005 4435      RRCA
7006 3122      DCA      DATA4
7007 1014      TAD      AUTO14   /SAVE MA
7010 3121      DCA      DATA3
7011 1015      TAD      AUTO15   /SAVE POINTER TO GOOD DATA
7012 3131      DCA      TEMP1
7013 6211      CDF      10       /SWITCH TO FIELD OF DATA
7014 1521      TAD I    DATA3   /PICK UP BAD DATA
7015 3117      DCA      DATA1   /SAVE IT
7016 1531      TAD I    TEMP1    /PICK UP GOOD DATA
7017 6201      CDF      0
7020 3120      DCA      DATA2   /SAVE IT
7021 7240      STA              /COMPUTE CALL PC
7022 1200      TAD      XDATER
7023 3125      DCA      DATA7
7024 4446      ERROR
7025 3245      DATAE4          /DATA ERROR
7026 5600      JMP I    XDATER   /PC DRVNO BAD GOOD MA CA SECTOR CB CALLPC

/PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED
/BY TWO SPACES

7027 0000      XPRNT4, 0        /CALL BY "PRNT4"
7030 3240      DCA      P4SAVE
7031 1240      TAD      P4SAVE
7032 7002      BSW
7033 4510      PRNT2
7034 1240      TAD      P4SAVE
7035 4510      PRNT2
7036 4512      SPACE2
7037 5627      JMP I    XPRNT4
7040 0000      P4SAVE, 0

/CLRRBF
/
/CLEAR READ BUFFER.  ARGUMENT FOLLOWING CALL IS MINUS THE NUMBER OF WORDS

```

```

/TO CLEAR.
/
7041 0000 XCLRRB, 0
7042 7200 CLA
7043 1641 TAD I XCLRRB /SET UP A COUNTER FROM ARGUMENT
7044 3131 DCA TEMP1
7045 2241 ISZ XCLRRB /SKIP OVER ARGUMENT ON RETURN
7046 1264 TAD PRDBFM /SET UP POINTER TO READ BUFFER IN
7047 3010 DCA AUTO10 / AN AUTO-INC REG
7050 6102 SPL /SKIP ON POWER LOW
7051 7410 SKP
7052 4000 JMS 0 /POWER FAIL
7053 6211 CDF 10 /FIELD OF READ BUFFER
7054 3410 DCA I AUTO10 /CLEAR A WORD
7055 2131 ISZ TEMP1 /CLEARED ENUFF WORDS?
7056 5254 JMP .-2 /NO
7057 6201 CDF 0 /BACK TO HERE
7060 6102 SPL /SKIP ON POWER LOW
7061 7410 SKP
7062 4000 JMS 0 /POWER FAIL
7063 5641 JMP I XCLRRB /RETURN

/
7064 3577 PRDBFM, REDBUF-1
7065 0000 SCNINT, 0
7066 7300 CLA CLL
7067 6055 ESIA /TURN OFF INTRPT CRTC
7070 6115 6115 /TURN OFF INTRPT TTY
7071 5665 JMP I SCNINT
7072 0000 SETINT, 0
7073 6111 6111
7074 7000 NOP
7075 6051 6051
7076 7000 NOP
7077 6116 6116
7100 7200 CLA
7101 7201 CLA IAC
7102 6115 6115
7103 6055 6055
7104 7200 CLA
7105 5672 JMP I SETINT
7106 0000 VTCHK, 0
7107 6030 KCF
7110 6031 KSF
7111 5706 JMP I VTCHK
7112 2306 ISZ VTCHK
7113 5706 JMP I VTCHK
7114 0000 WRITE1, 0
7115 1170 TAD M400 /SET UP WC
7116 4432 RLWC
7117 1141 TAD SECTOR /SET UP SA
7120 4431 RLSA
7121 1116 TAD DRVNUM
7122 7002 BSW
7123 1166 TAD K1015
7124 4430 RLCB

```

```

7125 5714      JMP I   WRITE1

/DRIVE ACTIVE TABLE
/ACCESS IS BY INDEX--EACH WORD SET TO 1 IF DRIVE IS ACTIVE AND 0 IF DRIVE
/IS NOT AVAILABLE FOR TESTING
/
7126 0000      DRVACT, 0                      /DRIVE 0
7127 0000      0                            /1
7130 0000      0                            /2
7131 0000      0                            /3

7132 0000      SECTAB, ZBLOCK 5              /FIVE WORD TABLE FOR FLAGGING SECTOR GROUPS READ

/ERROR COUNTERS MAINTAINED SEPERATELY FOR EACH DRIVE.
/NOTE: THESE COUNTERS ARE ORDER DEPENDENT (ACCESS IS BY INDEX)
7137 0000      DRVOER, 0                    /ERROR COUNT FOR DRIVE 0
7140 0000      0                            /DRIVE 1
7141 0000      0                            /DRIVE 2
7142 0000      0                            /3

/TABLE OF POINTERS TO THE IOT CODES FOR THE RL8A CONTROLLER
/
7143 4650      IOTTAB, IOT0
7144 4653      IOT1
7145 4660      IOT2
7146 4663      IOT3
7147 4666      IOT4
7150 4671      IOT5
7151 4674      IOT7
7152 4677      IOT10
7153 4702      IOT11
7154 4705      IOT12
7155 4710      IOT13
7156 4713      IOT14
7157 4716      IOT15
7160 4721      IOT17
7161 0000      0                            /TABLE TERMINATOR

7162 0100      PATPTA, PAT1                 /TABLE OF POINTERS TO THE PATTERNS
7163 0140      PAT2
7164 0200      PAT3
7165 0240      PAT4
7166 0300      PAT5
7167 0340      PAT6
7170 0400      PAT7
7171 0440      PAT8
7172 0000      0                            /0 FOR TABLE TERMINATOR

DECIMAL
7173 0002      CYLTAB, 2                   /TABLE OF DEFAULT CYLINDERS
7174 0007      7
7175 0016      14
7176 0024      20
7177 0033      27
7200 0041      33

```

7201	0046	38
7202	0055	45
7203	0064	52
7204	0072	58
7205	0101	65
7206	0110	72
7207	0115	77
7210	0124	84
7211	0133	91
7212	0141	97
7213	0146	102
7214	0154	108
7215	0161	113
7216	0170	120
7217	0177	127
7220	0206	134
7221	0213	139
7222	0222	146
7223	0230	152
7224	0235	157
7225	0244	164
7226	0252	170
7227	0261	177
7230	0270	184
7231	0275	189
7232	0303	195
7233	0312	202
7234	0317	207
7235	0326	214
7236	0334	220
7237	0343	227
7240	0352	234
7241	0361	241
7242	0367	247
7243	0375	253
7244	0377	255
7245	0404	260
7246	0413	267
7247	0421	273
7250	0430	280
7251	0436	286
7252	0443	291
7253	0452	298
7254	0461	305
7255	0467	311
7256	0476	318
7257	0505	325
7260	0512	330
7261	0521	337
7262	0530	344
7263	0536	350
7264	0543	355
7265	0551	361
7266	0556	366
7267	0565	373

7270	0574	380
7271	0603	387
7272	0610	392
7273	0617	399
7274	0625	405
7275	0632	410
7276	0641	417
7277	0647	423
7300	0656	430
7301	0665	437
7302	0672	442
7303	0700	448
7304	0707	455
7305	0714	460
7306	0723	467
7307	0731	473
7310	0740	480
7311	0747	487
7312	0756	494
7313	0764	500
7314	0772	506

CYLEND, 506
OCTAL

7315	6666	6666
7316	5555	5555
7317	3333	3333

/THE 12 BIT DATA

7320	2525	2525
7321	5252	5252
7322	2222	2222

LAST12, 2222

7323	7777	7777
7324	0000	0000
7325	6314	6314
	0001	1

FIELD

0000	*0		
0000	0000	NOPUNCH	/DISABLE BINARY
		BADSEC, ZBLOCK	100
		ENPUNCH	/ENABLE BINARY
0100	*100		/ROOM FOR 32 BAD SECTORS TO BE STORED
/THE DATA PATTERNS			
0100	0000	PAT1, ZBLOCK	40
			/32 8 BIT WORDS OF 0'S
0140	0377	PAT2,	377
0141	0377		377
0142	0377		377
0143	0377		377
0144	0377		377
0145	0377		377
0146	0125		125
0147	0125		125
0150	0125		125
0151	0125		125
0152	0125		125
0153	0125		125
0154	0377		377
0155	0377		377
0156	0377		377
0157	0377		377
0160	0125		125
0161	0125		125
0162	0125		125
0163	0125		125
0164	0377		377
0165	0377		377
0166	0125		125
0167	0125		125
0170	0376		376
0171	0125		125
0172	0376		376
0173	0125		125
0174	0365		365
0175	0365		365
0176	0365		365
0177	0365		365
0200	0000	PAT3,	0
0201	0000		0
0202	0000		0
0203	0000		0
0204	0000		0
0205	0000		0
0206	0377		377
0207	0377		377
0210	0377		377
0211	0377		377
0212	0377		377

0213	0377	377
0214	0000	0
0215	0000	0
0216	0000	0
0217	0000	0
0220	0377	377
0221	0377	377
0222	0377	377
0223	0377	377
0224	0000	0
0225	0000	0
0226	0377	377
0227	0377	377
0230	0000	0
0231	0000	0
0232	0377	377
0233	0377	377
0234	0000	0
0235	0000	0
0236	0377	377
0237	0377	377
0240	0052	PAT4, 052
0241	0252	252
0242	0125	125
0243	0125	125
0244	0125	125
0245	0125	125
0246	0252	252
0247	0252	252
0250	0252	252
0251	0252	252
0252	0252	252
0253	0252	252
0254	0125	125
0255	0125	125
0256	0125	125
0257	0125	125
0260	0252	252
0261	0252	252
0262	0252	252
0263	0252	252
0264	0125	125
0265	0125	125
0266	0252	252
0267	0252	252
0270	0125	125
0271	0125	125
0272	0252	252
0273	0252	252
0274	0125	125
0275	0125	125
0276	0252	252
0277	0252	252

0300	0333	PAT5,	333
0301	0155		155
0302	0266		266
0303	0333		333
0304	0155		155
0305	0266		266
0306	0333		333
0307	0155		155
0310	0266		266
0311	0333		333
0312	0155		155
0313	0266		266
0314	0333		333
0315	0155		155
0316	0266		266
0317	0333		333
0320	0155		155
0321	0266		266
0322	0333		333
0323	0155		155
0324	0266		266
0325	0333		333
0326	0155		155
0327	0266		266
0330	0333		333
0331	0155		155
0332	0266		266
0333	0333		333
0334	0155		155
0335	0266		266
0336	0333		333
0337	0155		155

0340	0244	PAT6,	244
0341	0105		105
0342	0321		321
0343	0042		042
0344	0150		150
0345	0221		221
0346	0264		264
0347	0110		110
0350	0132		132
0351	0044		044
0352	0055		055
0353	0022		022
0354	0026		026
0355	0211		211
0356	0213		213
0357	0104		104
0360	0105		105
0361	0242		242
0362	0042		042
0363	0321		321
0364	0241		241
0365	0350		350

/110110110110110110110....

0366	0110	110
0367	0264	264
0370	0044	044
0371	0132	132
0372	0022	022
0373	0055	055
0374	0211	211
0375	0026	026
0376	0104	104
0377	0213	213

0400	0377	PAT7,	377
0401	0377		377
0402	0377		377
0403	0377		377
0404	0377		377
0405	0377		377
0406	0377		377
0407	0377		377
0410	0377		377
0411	0377		377
0412	0377		377
0413	0377		377
0414	0377		377
0415	0377		377
0416	0377		377
0417	0377		377
0420	0377		377
0421	0377		377
0422	0377		377
0423	0377		377
0424	0377		377
0425	0377		377
0426	0377		377
0427	0377		377
0430	0377		377
0431	0377		377
0432	0377		377
0433	0377		377
0434	0377		377
0435	0377		377
0436	0377		377
0437	0377		377

0440	0113	PAT8,	113
0441	0113		113
0442	0245		245
0443	0245		245
0444	0322		322
0445	0322		322
0446	0151		151
0447	0151		151
0450	0264		264
0451	0264		264
0452	0132		132

0453	0132	132
0454	0055	055
0455	0055	055
0456	0226	226
0457	0226	226
0460	0113	113
0461	0113	113
0462	0245	245
0463	0245	245
0464	0322	322
0465	0322	322
0466	0151	151
0467	0151	151
0470	0264	264
0471	0264	264
0472	0132	132
0473	0132	132
0474	0055	055
0475	0055	055
0476	0226	226
0477	0226	226

0500 4301 MANDEC, TEXT "#AJRLI-B RL8A/RL02 READ/WRITE DIAGNOSTIC#" /

HP

0501 1222
 0502 1411
 0503 5502
 0504 4022
 0505 1470
 0506 0157
 0507 2214
 0510 6062
 0511 4022
 0512 0501
 0513 0457
 0514 2722
 0515 1124
 0516 0540
 0517 0411
 0520 0107
 0521 1617
 0522 2324
 0523 1103
 0524 4300
 0525 4377
 0526 4005
 0527 2222
 0530 1722
 0531 4014
 0532 1115
 0533 1124
 0534 4005
 0535 3003
 0536 0505
 0537 0405
 0540 0440

LIMEXC, TEXT "?? ERROR LIMIT EXCEEDED - "

0541	5540		
0542	0000		
0543	2405	TSTDRV, TEXT	"TEST DRIVE "
0544	2324		
0545	4004		
0546	2211		
0547	2605		
0550	4000		
0551	4077	SPOS, TEXT	" ? "
0552	4000		
0553	0422	DROPNG, TEXT	"DROPPING DRIVE "
0554	1720		
0555	2011		
0556	1607		
0557	4004		
0560	2211		
0561	2605		
0562	4000		
0563	4311	OPT1AV, TEXT	"#IS OPTION 1 CLK ENABLE? "
0564	2340		
0565	4017		
0566	2024		
0567	1117		
0570	1640		
0571	6140		
0572	0314		
0573	1340		
0574	0516		
0575	0102		
0576	1405		
0577	7740		
0600	0000		
0601	4304	USEDEF, TEXT	"#DO YOU WISH TO SELECT NON-DEFAULT PARAMETERS? "
0602	1740		
0603	3117		
0604	2540		
0605	2711		
0606	2310		
0607	4024		
0610	1740		
0611	2305		
0612	1405		
0613	0324		
0614	4016		
0615	1716		
0616	5504		
0617	0506		
0620	0125		
0621	1424		
0622	4020		
0623	0122		
0624	0115		
0625	0524		
0626	0522		
0627	2377		

0630	4000		
0631	2523	USE62, TEXT	"USE DEVICE CODES 62,63? "
0632	0540		
0633	0405		
0634	2611		
0635	0305		
0636	4003		
0637	1704		
0640	0523		
0641	4066		
0642	6254		
0643	6663		
0644	7740		
0645	0000		
0646	0530	EXMAIN, TEXT	"EXECUTE WRITE LOCK DATA PROTECTION TEST? "
0647	0503		
0650	2524		
0651	0540		
0652	2722		
0653	1124		
0654	0540		
0655	1417		
0656	0313		
0657	4004		
0660	0124		
0661	0140		
0662	2022		
0663	1724		
0664	0503		
0665	2411		
0666	1716		
0667	4024		
0670	0523		
0671	2477		
0672	4000		
0673	2523	USALCY, TEXT	"USE ALL CYLINDERS? "
0674	0540		
0675	0114		
0676	1440		
0677	0331		
0700	1411		
0701	1604		
0702	0522		
0703	2377		
0704	4000		
0705	2523	USLOLM, TEXT	"USE LOWER CYLINDER LIMIT? "
0706	0540		
0707	1417		
0710	2705		
0711	2240		
0712	0331		
0713	1411		
0714	1604		
0715	0522		
0716	4014		

0717	1115		
0720	1124		
0721	7740		
0722	0000		
0723	0516	ENOVAL, TEXT	"ENTER VALUE IN OCTAL: "
0724	2405		
0725	2240		
0726	2601		
0727	1425		
0730	0540		
0731	1116		
0732	4017		
0733	0324		
0734	0114		
0735	7240		
0736	0000		
0737	2523	USUPLM, TEXT	"USE UPPER CYLINDER LIMIT? "
0740	0540		
0741	2520		
0742	2005		
0743	2240		
0744	0331		
0745	1411		
0746	1604		
0747	0522		
0750	4014		
0751	1115		
0752	1124		
0753	7740		
0754	0000		
0755	4314	LLLTHL, TEXT	"LOW LIMIT MUST BE LESS THAN HI LIMIT:"
0756	1727		
0757	4014		
0760	1115		
0761	1124		
0762	4015		
0763	2523		
0764	2440		
0765	0205		
0766	4014		
0767	0523		
0770	2340		
0771	2410		
0772	0116		
0773	4010		
0774	1140		
0775	1411		
0776	1511		
0777	2443		
1000	0000		
1001	2523	US1SUR, TEXT	"USE ONLY ONE SURFACE? "
1002	0540		
1003	1716		
1004	1431		
1005	4017		

1006	1605		
1007	4023		
1010	2522		
1011	0601		
1012	0305		
1013	7740		
1014	0000		
1015	2320	SPCSUR, TEXT	"SPECIFY SURFACE (0 OR 1): "
1016	0503		
1017	1106		
1020	3140		
1021	2325		
1022	2206		
1023	0103		
1024	0540		
1025	5060		
1026	4017		
1027	2240		
1030	6151		
1031	7240		
1032	0000		
1033	0516	SPERLM, TEXT	"ENTER ERROR LIMIT IN OCTAL (DEFAULT=24): "
1034	2405		
1035	2240		
1036	0522		
1037	2217		
1040	2240		
1041	1411		
1042	1511		
1043	2440		
1044	1116		
1045	4017		
1046	0324		
1047	0114		
1050	4050		
1051	0405		
1052	0601		
1053	2514		
1054	2475		
1055	6264		
1056	5172		
1057	4000		
1060	7743	QESMRK, TEXT	"?:"
1061	0000		
1062	4305	EOPMES, TEXT	"#END PASS "
1063	1604		
1064	4020		
1065	0123		
1066	2340		
1067	0000		
1070	3603	UPARRC, TEXT	"^C#"
1071	4300		
1072	3607	UPARRG, TEXT	"^G"
1073	0000		
1074	3606	FILLEQ, TEXT	"^F#FILL = "

1075	4306		
1076	1114		
1077	1440		
1100	7540		
1101	0000		
1102	4323	SWRMSG, TEXT	"#SR = "
1103	2240		
1104	7540		
1105	0000		
1106	4040	TWOSPA, TEXT	" "
1107	0000		
1110	4320	PLWRLK, TEXT	"#PLEASE WRITE LOCK DRIVE "
1111	1405		
1112	0123		
1113	0540		
1114	2722		
1115	1124		
1116	0540		
1117	1417		
1120	0313		
1121	4004		
1122	2211		
1123	2605		
1124	4000		
1125	4320	RLWREN, TEXT	"#PLEASE WRITE ENABLE DRIVE "
1126	1405		
1127	0123		
1130	0540		
1131	2722		
1132	1124		
1133	0540		
1134	0516		
1135	0102		
1136	1405		
1137	4004		
1140	2211		
1141	2605		
1142	4000		
1143	4377	FATLER, TEXT	"#? FATAL ERROR - "
1144	4006		
1145	0124		
1146	0114		
1147	4005		
1150	2222		
1151	1722		
1152	4055		
1153	4000		
1154	4377	CANTRS, TEXT	"#? CAN'T RESET ERROR - "
1155	4003		
1156	0116		
1157	4724		
1160	4022		
1161	0523		
1162	0524		
1163	4005		

1164	2222		
1165	1722		
1166	4055		
1167	4000		
1170	4377	DRNLOD, TEXT	"#? DRIVE IN LOAD STATE - "
1171	4004		
1172	2211		
1173	2605		
1174	4011		
1175	1640		
1176	1417		
1177	0104		
1200	4023		
1201	2401		
1202	2405		
1203	4055		
1204	4000		
1205	4377	CNMROY, TEXT	"#? CAN'T MAKE READY - "
1206	4003		
1207	0116		
1210	4724		
1211	4015		
1212	0113		
1213	0540		
1214	2205		
1215	0104		
1216	3140		
1217	5540		
1220	0000		
1221	4377	NOTRDY, TEXT	"#? NOT READY - "
1222	4016		
1223	1724		
1224	4022		
1225	0501		
1226	0431		
1227	4055		
1230	4000		
1231	4377	TOOMBS, TEXT	"#? TOO MANY BAD SECTORS - "
1232	4024		
1233	1717		
1234	4015		
1235	0116		
1236	3140		
1237	0201		
1240	0440		
1241	2305		
1242	0324		
1243	1722		
1244	2340		
1245	5540		
1246	0000		
1247	4324	TSTABR, TEXT	"#TEST ABORTED#"
1250	0523		
1251	2440		
1252	0102		

1253	1722				
1254	2405				
1255	0443				
1256	0000				
1257	4020	PC,	TEXT	" PC#"	
1260	0343				
1261	0000				
1262	4020	DRVNO,	TEXT	" PC	DRV NO.#"
1263	0340				
1264	4040				
1265	4004				
1266	2226				
1267	4016				
1270	1756				
1271	4300				
1272	4020	ER,	TEXT	" PC	DRV NO. ER#"
1273	0340				
1274	4040				
1275	4004				
1276	2226				
1277	4016				
1300	1756				
1301	4040				
1302	4005				
1303	2243				
1304	0000				
1305	4020	ERCALL,	TEXT	" PC	DRV NO. ER CALLPC#"
1306	0340				
1307	4040				
1310	4004				
1311	2226				
1312	4016				
1313	1756				
1314	4040				
1315	4005				
1316	2240				
1317	4040				
1320	4003				
1321	0114				
1322	1420				
1323	0343				
1324	0000				
1325	4020	EXAC2,	TEXT	" PC	DRV NO. WD1-ACTUAL-WD2 WD1-EXPCTD-WD2#"
1326	0340				
1327	4040				
1330	4004				
1331	2226				
1332	4016				
1333	1756				
1334	4027				
1335	0461				
1336	5501				
1337	0324				
1340	2501				
1341	1455				

1342 2704
1343 6240
1344 4027
1345 0461
1346 5505
1347 3020
1350 0324
1351 0455
1352 2704
1353 6243
1354 0000
1355 4020
1356 0340
1357 4040
1360 4004
1361 2226
1362 4016
1363 1756
1364 4010
1365 0501
1366 0455
1367 0205
1370 0617
1371 2255
1372 0331
1373 1440
1374 4010
1375 0501
1376 0455
1377 0106
1400 2405
1401 2255
1402 0331
1403 1440
1404 4040
1405 4003
1406 0140
1407 4040
1410 4003
1411 0114
1412 1420
1413 0343
1414 0000
1415 4020
1416 0340
1417 4040
1420 4004
1421 2226
1422 4016
1423 1756
1424 4040
1425 4005
1426 2240
1427 4040
1430 4040

HEDCYL, TEXT " PC DRV NO. HEAD-BEFOR-CYL HEAD-AFTER-CYL CA CALLPC#"

HNFERR, TEXT " PC DRV NO. ER CA SA HED-CYL CALLPC#"

1431 4003
 1432 0140
 1433 4040
 1434 4040
 1435 4023
 1436 0140
 1437 4040
 1440 1005
 1441 0455
 1442 0331
 1443 1440
 1444 4003
 1445 0114
 1446 1420
 1447 0343
 1450 0000
 1451 4020
 1452 0340
 1453 4040
 1454 4004
 1455 2226
 1456 4016
 1457 1756
 1460 4027
 1461 0461
 1462 5523
 1463 2401
 1464 2425
 1465 2355
 1466 2704
 1467 6240
 1470 4040
 1471 4005
 1472 2240
 1473 4040
 1474 4040
 1475 4003
 1476 0240
 1477 4040
 1500 4040
 1501 4003
 1502 0140
 1503 4040
 1504 4040
 1505 4023
 1506 0140
 1507 4040
 1510 4003
 1511 0114
 1512 1420
 1513 0343
 1514 0000
 1515 4020
 1516 0340
 1517 4040

ALLRG2. TEXT " PC DRV NO. WD1-STATUS-WD2 ER CB CA SA CALLPC#"

ALLREG. TEXT " PC DRV NO. WD1-STATUS-WD2 ER CB CA SA WC#"

1520 4004
1521 2226
1522 4016
1523 1756
1524 4027
1525 0461
1526 5523
1527 2401
1530 2425
1531 2355
1532 2704
1533 6240
1534 4040
1535 4005
1536 2240
1537 4040
1540 4040
1541 4003
1542 0240
1543 4040
1544 4040
1545 4003
1546 0140
1547 4040
1550 4040
1551 4023
1552 0140
1553 4040
1554 4040
1555 4027
1556 0343
1557 0000
1560 4020
1561 0340
1562 4040
1563 4004
1564 2226
1565 4016
1566 1756
1567 4027
1570 0461
1571 5523
1572 2401
1573 2425
1574 2355
1575 2704
1576 6240
1577 4040
1600 4005
1601 2240
1602 4040
1603 4040
1604 4003
1605 0240
1606 4040

ALREG3, TEXT " PC DRV NO. WD1-STATUS-WD2 ER CB CA SA#"

1607 4040
 1610 4003
 1611 0140
 1612 4040
 1613 4040
 1614 4023
 1615 0143
 1616 0000
 1617 4020
 1620 0340
 1621 4040
 1622 4004
 1623 2226
 1624 4016
 1625 1756
 1626 4005
 1627 3020
 1630 0324
 1631 0440
 1632 4001
 1633 0324
 1634 2501
 1635 1443
 1636 0000
 1637 4020
 1640 0340
 1641 4040
 1642 4004
 1643 2226
 1644 4016
 1645 1756
 1646 4040
 1647 2704
 1650 6140
 1651 4040
 1652 4040
 1653 2704
 1654 6240
 1655 4040
 1656 4040
 1657 2704
 1660 6340
 1661 4040
 1662 4040
 1663 2704
 1664 6440
 1665 4040
 1666 4023
 1667 0503
 1670 2417
 1671 2243
 1672 0000
 1673 4020
 1674 0340
 1675 4040

EXPACT, TEXT " PC DRV NO. EXPCTD ACTUAL#"

WD1WD4, TEXT " PC DRV NO. WD1 WD2 WD3 WD4 SECTOR#"

WDNOSA, TEXT " PC DRV NO. WORD NO. SA#"

1676 4004
1677 2226
1700 4016
1701 1756
1702 4027
1703 1722
1704 0440
1705 1617
1706 5640
1707 4023
1710 0143
1711 0000
1712 4020
1713 0340
1714 4040
1715 4004
1716 2226
1717 4016
1720 1756
1721 4027
1722 0461
1723 5523
1724 2401
1725 2425
1726 2355
1727 2704
1730 6240
1731 4040
1732 4005
1733 2243
1734 0000
1735 4020
1736 0340
1737 4040
1740 4004
1741 2226
1742 4016
1743 1756
1744 4027
1745 0461
1746 5523
1747 2401
1750 2425
1751 2355
1752 2704
1753 6240
1754 4040
1755 4005
1756 2240
1757 4040
1760 4040
1761 4003
1762 Q240
1763 4040
1764 4040

STATER, TEXT " PC DRV NO. WD1-STATUS-WD2 ER#"

STECCC, TEXT " PC DRV NO. WD1-STATUS-WD2 ER CB CA CALLPC#"

1765 4003
 1766 0140
 1767 4040
 1770 4003
 1771 0114
 1772 1420
 1773 0343
 1774 0000
 1775 4020
 1776 0340
 1777 4040
 2000 4004
 2001 2226
 2002 4016
 2003 1756
 2004 4005
 2005 3020
 2006 0324
 2007 0440
 2010 4001
 2011 0324
 2012 2501
 2013 1440
 2014 4040
 2015 4003
 2016 0140
 2017 4040
 2020 4023
 2021 0503
 2022 2417
 2023 2240
 2024 4040
 2025 4003
 2026 0243
 2027 0000
 2030 4020
 2031 0340
 2032 4040
 2033 4004
 2034 2226
 2035 4016
 2036 1756
 2037 4005
 2040 3020
 2041 0324
 2042 0440
 2043 4001
 2044 0324
 2045 2501
 2046 1440
 2047 4040
 2050 4015
 2051 0140
 2052 4040
 2053 4040

EAASB, TEXT " PC DRV NO. EXPCTD ACTUAL CA SECTOR CB#"

C7DAT, TEXT " PC DRV NO. EXPCTD ACTUAL MA CB SECTOR(EXPCTD)#"

2143 0340
2144 4040
2145 4004
2146 2226
2147 4016
2150 1756
2151 4040
2152 0201
2153 0440
2154 4040
2155 4040
2156 0717
2157 1704
2160 4040
2161 4040
2162 4015
2163 0140
2164 4040
2165 4040
2166 4003
2167 0140
2170 4040
2171 4023
2172 0503
2173 2417
2174 2240
2175 4040
2176 4003
2177 0240
2200 4040
2201 4003
2202 0114
2203 1420
2204 0343
2205 0000
2206 4020
2207 0340
2210 4040
2211 4004
2212 2226
2213 4016
2214 1756
2215 4040
2216 1116
2217 2420
2220 0343
2221 0000
2222 4020
2223 0340
2224 4040
2225 4004
2226 2226
2227 4016
2230 1756
2231 4040

PCINT, TEXT " PC DRV NO. INTPC#"

CASEC, TEXT " PC DRV NO. CA SECTOR#"

2232 4003
 2233 0140
 2234 4040
 2235 4023
 2236 0503
 2237 2417
 2240 2243
 2241 0000
 2242 4320
 2243 1727
 2244 0522
 2245 4006
 2246 0111
 2247 1440
 2250 4020
 2251 0340
 2252 7540
 2253 0000
 2254 4316
 2255 1740
 2256 0422
 2257 1126
 2260 0523
 2261 4001
 2262 2601
 2263 1114
 2264 0102
 2265 1405
 2266 4006
 2267 1722
 2270 4024
 2271 0523
 2272 2411
 2273 1607
 2274 4300

'POWER, TEXT "#POWER FAIL PC = "

NODRVS, TEXT "#NO DRIVES AVAILABLE FOR TESTING#"

/ERROR TABLE

/

/EACH ENTRY IN THE ERROR TABLE CONSISTS OF FOUR WORDS:

/ 1) ERROR CODE FOR APT USE WITH THE FOLLOWING MEANINGS:

/ 6000 STATUS ERROR

/ 6100 DATA ERROR

/ 6300 READ ERROR

/ 6500 WRITE ERROR

/ 0000 REPORT PC; NOT ERROR CODE

/ 2) POINTER TO ERROR MESSAGE (MAY BE 0 FOR NULL)

/ 3) POINTER TO DATA HEADER (MAY BE 0 FOR NULL)

/ 4) MINUS NUMBER OF DATA ITEMS (AT LEAST -1 (FOR PC))

/THE ERROR CALL CONSISTS OF THE "ERROR" STATEMENT FOLLOWED BY A

/POINTER TO AN ENTRY IN THE ERROR TABLE

/

2275 0000
 2276 2301
 2277 2206

SFTWER, 0

.+3

PCINT

/ PC DRV NO. INTPC

2300 7775
2301 4377
2302 4023
2303 1706
2304 2427
2305 0122
2306 0540
2307 0522
2310 2217
2311 2240
2312 5012
2313 1520
2314 4024
2315 1740
2316 6040
2317 1722
2320 4011
2321 1624
2322 0522
2323 2225
2324 2024
2325 4027
2326 1005
2327 1640
2330 1116
2331 2405
2332 2222
2333 2520
2334 2440
2335 2331
2336 2324
2337 0515
2340 4023
2341 5702
2342 4017
2343 0606
2344 5143
2345 0000
2346 0000
2347 2352
2350 1272
2351 7775
2352 4305
2353 2222
2354 1722
2355 4006
2356 1401
2357 0740
2360 1617
2361 2440
2362 2305
2363 2440
2364 0231
2365 4003
2366 2203

-3

TEXT

"?? SOFTWARE ERROR (JMP TO 0 OR INTERRUPT WHEN INTERRUPT SYSTEM S/B OFF)?"

CRCFLG, 0

. +3

ER

-3

TEXT

"ERROR FLAG NOT SET BY CRC ERROR?"

2367	4005		
2370	2222		
2371	1722		
2372	4300		
2373	6000	EF9,	6000
2374	2377		ERFLST
2375	1515		ALLREG
2376	7767		-11
2377	4305	ERFLST, TEXT	"/-9 "#ERROR FLAG SET#"
2400	2222		
2401	1722		
2402	4006		
2403	1401		
2404	0740		
2405	2305		
2406	2443		
2407	0000		
2410	6000	EF7,	6000
2411	2377		ERFLST
2412	1451		ALLRG2
2413	7767		-11
2414	0000	NOHNF,	0
2415	2420		.+3
2416	1415		HNFFERR
2417	7771		-7
2420	4303	TEXT	"#COULD NOT FORCE OPI-HNF ON READ#"
2421	1725		
2422	1404		
2423	4016		
2424	1724		
2425	4006		
2426	1722		
2427	0305		
2430	4017		
2431	2011		
2432	5510		
2433	1606		
2434	4017		
2435	1640		
2436	2205		
2437	0104		
2440	4300		
2441	0000	SANAE2, 0	
2442	2445		.+3
2443	1617		EXPACT
2444	7774		-4
2445	4323	TEXT	"#SA NOT AS EXPECTED AFTER 2 SECTOR WRITE#"
2446	0140		
2447	1617		
2450	2440		
2451	0123		
2452	4005		
2453	3020		
2454	0503		
2455	2405		

2456 0440
2457 0106
2460 2405
2461 2240
2462 6240
2463 2305
2464 0324
2465 1722
2466 4027
2467 2211
2470 2405
2471 4300
2472 6000
2473 2476
2474 1637
2475 7771
2476 4311
2477 1626
2500 0114
2501 1104
2502 4003
2503 0122
2504 2422
2505 1104
2506 0705
2507 4023
2510 0522
2511 1101
2512 1440
2513 1617
2514 5640
2515 1116
2516 4002
2517 2306
2520 4300
2521 6000
2522 2525
2523 1637
2524 7771
2525 4302
2526 0104
2527 4064
2530 4027
2531 1722
2532 0423
2533 4017
2534 0640
2535 6047
2536 2340
2537 1116
2540 4002
2541 2306
2542 4300
2543 6000
2544 2547

BADSN, 6000
.+3
WD1WD4
-7
TEXT "#INVALID CARTRIDGE SERIAL NO. IN BSF#"

BAD4Z, 6000
.+3
WD1WD4
-7
TEXT "#BAD 4 WORDS OF 0'S IN BSF#"

INVENT, 6000
.+3

2545	1637	
2546	7771	WD1WD4
2547	4311	-7
2550	1626	TEXT "#INVALID ENTRY IN BSF#"
2551	0114	
2552	1104	
2553	4005	
2554	1624	
2555	2231	
2556	4011	
2557	1640	
2560	0223	
2561	0643	
2562	0000	
2563	6000	BSDOS, 6000
2564	2567	.+3
2565	1637	WD1WD4
2566	7771	-7
2567	4302	TEXT "#BAD SECTOR OUT OF SEQUENCE IN BSF#"
2570	0104	
2571	4023	
2572	0503	
2573	2417	
2574	2240	
2575	1725	
2576	2440	
2577	1706	
2600	4023	
2601	0521	
2602	2505	
2603	1603	
2604	0540	
2605	1116	
2606	4002	
2607	2306	
2610	4300	
2611	6000	NON1WD, 6000
2612	2615	.+3
2613	1673	WDNOSA
2614	7774	-4
2615	4316	TEXT "#NON-1-FILLED WORD FOUND AFTER LAST ENTRY IN BSF#"
2616	1716	
2617	5561	
2620	5506	
2621	1114	
2622	1405	
2623	0440	
2624	2717	
2625	2204	
2626	4006	
2627	1725	
2630	1604	
2631	4001	
2632	0624	
2633	0522	

2634 4014
2635 0123
2636 2440
2637 0516
2640 2422
2641 3140
2642 1116
2643 4002
2644 2306
2645 4300
2646 0000
2647 2652
2650 1617
2651 7774
2652 4323
2653 0140
2654 1617
2655 2440
2656 0123
2657 4005
2660 3020
2661 0503
2662 2405
2663 0440
2664 0106
2665 2405
2666 2240
2667 6140
2670 2305
2671 0324
2672 1722
2673 4027
2674 2211
2675 2405
2676 4300
2677 0000
2700 2703
2701 1272
2702 7775
2703 4303
2704 2203
2705 4005
2706 2222
2707 1722
2710 4003
2711 1725
2712 1404
2713 4016
2714 1724
2715 4002
2716 0540
2717 0617
2720 2203
2721 0504
2722 4300

SANAE1, 0

.+3

EXPACT

-4

TEXT "#SA NOT AS EXPECTED AFTER 1 SECTOR WRITE#"

NOCRC, 0

.+3

ER

-3

TEXT "#CRC ERROR COULD NOT BE FORCED#"

2723	0000	CRCDC,	0	
2724	2727		.+3	
2725	1272		ER	
2726	7775		-3	
2727	4303	TEXT		"#CRC BIT NOT CLEARED BY RLDC#"
2730	2203			
2731	4002			
2732	1124			
2733	4016			
2734	1724			
2735	4003			
2736	1405			
2737	0122			
2740	0504			
2741	4002			
2742	3140			
2743	2214			
2744	0403			
2745	4300			
2746	6000	NOCLK,	6000	
2747	2752		.+3	
2750	1305		ERCALL	
2751	7774		-4	
2752	4316	TEXT		"#NO CLOCK (OPI SET ON GET STATUS)!"
2753	1740			
2754	0314			
2755	1703			
2756	1340			
2757	5017			
2760	2011			
2761	4023			
2762	0524			
2763	4017			
2764	1640			
2765	0705			
2766	2440			
2767	2324			
2770	0124			
2771	2523			
2772	5143			
2773	0000			
2774	0000	PONLY,	0	
2775	0000		0	/NULL ERROR MESSAGE
2776	1257		PC	
2777	7777		-1	
3000	6000	NORESE,	6000	
3001	3004		.+3	
3002	1325		EXAC2	
3003	7772		-6	
3004	4322	TEXT		"#RESET DID NOT RESET DRIVE!"
3005	0523			
3006	0524			
3007	4004			
3010	1104			
3011	4016			

```
3012 1724
3013 4022
3014 0523
3015 0524
3016 4004
3017 2211
3020 2605
3021 4300
3022 6000      SEKFAL, 6000
3023 3026      .+3
3024 1355      HEDCYL
3025 7770      -10
3026 4323      TEXT      "#SEEK FAILURE#" /-8
3027 0505
3030 1340
3031 0601
3032 1114
3033 2522
3034 0543
3035 0000
3036 6000      NORDY3, 6000
3037 3042      .+3
3040 1735      STECCC
3041 7770      -10
3042 4322      TEXT      "#READY NOT SET WITHIN 3 SECONDS#"
3043 0501
3044 0431
3045 4016
3046 1724
3047 4023
3050 0524
3051 4027
3052 1124
3053 1011
3054 1640
3055 6340
3056 2305
3057 0317
3060 1604
3061 2343
3062 0000
3063 6100      DATAE1, 6100
3064 3067      DAERTX
3065 1775      EAASB
3066 7771      -7
3067 4304      DAERTX, TEXT      "#DATA ERROR#"
3070 0124
3071 0140
3072 0522
3073 2217
3074 2243
3075 0000
3076 6100      DATAE2, 6100
3077 3067      DAERTX
3100 2113      EAMACB
```

3101	7772	-6	
3102	0000	ZFILER, 0	
3103	3106	.+3	
3104	2070	ACTMAB	
3105	7773	-5	
3106	4332	TEXT	"#ZERO FILL ERROR#"
3107	0522		
3110	1740		
3111	0611		
3112	1414		
3113	4005		
3114	2222		
3115	1722		
3116	4300		
3117	6000	WRLDAT, 6000	
3120	3123	.+3	
3121	1262	DRVNO	
3122	7776	-2	
3123	4327	TEXT	"#WRITE LOCK DID NOT PROTECT DATA#"
3124	2211		
3125	2405		
3126	4014		
3127	1703		
3130	1340		
3131	0411		
3132	0440		
3133	1617		
3134	2440		
3135	2022		
3136	1724		
3137	0503		
3140	2440		
3141	0401		
3142	2401		
3143	4300		
3144	6000	EFNSWL, 6000	
3145	3150	.+3	
3146	1712	STATER	
3147	7773	-5	
3150	4305	TEXT	"#ERROR FLAG NOT SET ON WRITE TO WRITE LOCKED DRIVE#"
3151	2222		
3152	1722		
3153	4006		
3154	1401		
3155	0740		
3156	1617		
3157	2440		
3160	2305		
3161	2440		
3162	1716		
3163	4027		
3164	2211		
3165	2405		
3166	4024		
3167	1740		

```
3170 2722
3171 1124
3172 0540
3173 1417
3174 0313
3175 0504
3176 4004
3177 2211
3200 2605
3201 4300
3202 6000 EFSC7, 6000
3203 3206 .+3
3204 1560 ALREG3
3205 7770 -10
3206 4305 TEXT "#ERROR FLAG SET ON COMMAND 7#"
3207 2222
3210 1722
3211 4006
3212 1401
3213 0740
3214 2305
3215 2440
3216 1716
3217 4003
3220 1715
3221 1501
3222 1604
3223 4067
3224 4300
3225 0000 C7DE, 0
3226 3231 .+3
3227 2030 C7DAT
3230 7771 -7
3231 4303 TEXT "#COMMAND 7 DATA ERROR#"
3232 1715
3233 1501
3234 1604
3235 4067
3236 4004
3237 0124
3240 0140
3241 0522
3242 2217
3243 2243
3244 0000
3245 6100 DATAE4, 6100
3246 3067 DAERTX
3247 2142 BGMS2
3250 7767 -11
3251 6100 ADJINT, 6100
3252 3255 .+3
3253 2222 CASEC
3254 7774 -4
3255 4301 TEXT "#ADJACENT CYLINDER INTERFERENCE#"
3256 0412
```

3257 0103
 3260 0516
 3261 2440
 3262 0331
 3263 1411
 3264 1604
 3265 0522
 3266 4011
 3267 1624
 3270 0522
 3271 0605
 3272 2205
 3273 1603
 3274 0543
 3275 0000
 3276 6100
 3277 3302
 3300 2222
 3301 7774
 3302 4317
 3303 2605
 3304 2227
 3305 2211
 3306 2405
 3307 4005
 3310 2222
 3311 1722
 3312 4300

OVRWRT, 6100
 .+3
 CASEC
 -4
 TEXT "#OVERWRITE ERROR#"

3600 *3600

/BEGINNING OF 2K BUFFER:

3600 0000 NOPUNCH
 4200 0000 REDBUF, ZBLOCK 400
 4600 0000 BUF2, ZBLOCK 400
 5200 0000 BUF3, ZBLOCK 400
 5600 0000 BUF4, ZBLOCK 400
 6200 0000 WRTBUF, ZBLOCK 400
 6600 0000 BUF6, ZBLOCK 400
 7200 0000 BUF7, ZBLOCK 400
 ENPUNCH

/DISABLE BINARY

/RESERVE EACH (8 BIT MODE SIZED) SECTOR SEPERATELY
 /SO THEY CAN BE INDIVIDUALLY ACCESSED

/SEPERATE READ AND WRITE BUFFERS FOR LESS THAN 1K TRANSFERS

/ENABLE BINARY

0000 FIELD 0

0200 *200
8

SEQ 143

A1E37	3477	C7DAT	2030	CRCFLG	2346	EF7	2410
ACTMAB	2070	C7DE	3225	CHLF	4514	EF9	2373
ADD4	0746	CAF	6007	CHLFSV	4462	EFCH30	1531
ADDP33	2323	CANTRS	1154	CT2RET	6566	EFCHK	1061
ADDP37	3301	CARRET	5227	CTLF	4234	EFNSWL	3144
ADDP38	3537	CASEC	2222	CTLG	4236	EFSC7	3202
ADJINT	3251	CBLOOP	6715	CYLBAD	4476	END29	1015
ALL1CH	1261	CH1LUP	1275	CYLEND	7314	END30	1553
ALLCYL	0155	CHARSV	4441	CYLGUD	6730	END31	1711
ALLREG	1515	CHK35	3000	CYLNDR	0136	END33	2442
ALLRG2	1451	CHKBSF	1200	CYLPNT	0137	END35	3023
ALREG3	1560	CHKCN1	6527	CYLSAV	5502	END36	3245
ANOTBF	6456	CHKCN2	6530	CYLTA8	7173	END37	3504
ANOTH8	6535	CHKCNT	1335	DAERTX	3067	END38	3707
ANOTHF	6446	CHKLUP	1206	DATA1	0117	ENDHNI	0613
APTCHK	4443	CHKOK	4402	DATA12	7315	ENDHNI2	0642
APTCOM	4503	CHKRET	1332	DATA2	0120	ENDHNI3	0666
APTERR	4126	CHKTRK	4475	DATA3	0121	ENDZ12	2270
APTOP1	0445	CHTRE2	6557	DATA4	0122	ENDZ8	2071
APTSET	0412	CHTRER	6474	DATA5	0123	ENOVAL	0723
ASKSUR	0342	CHTRET	6526	DATA6	0124	ENTLIS	5260
AUTO10	0010	CKTRK2	6531	DATA7	0125	ENTRET	5262
AUTO11	0011	CLRRBF	4471	DATAE1	3063	ENTVAL	4454
AUTO12	0012	CMPERR	6166	DATAE2	3076	EOP	3725
AUTO13	0013	CMPLPI	6144	DATAE4	3245	EOPMES	1062
AUTO14	0014	CMPLPO	6133	DATCNT	4140	ER	1272
AUTO15	0015	CMPPAT	4473	DATE35	3015	ERCALL	1305
BAD4Z	2521	CMPWRD	6756	DATERR	4504	EREXMA	0725
BADC33	2410	CNMRDY	1205	DATEZ1	2240	EREXSW	1005
BADC37	3352	CNT32	2163	DATEZ8	2042	ERFLST	2377
BADC38	3627	CNTFRT	4274	DATLUP	4061	ERLMA5	0400
BADEND	5364	CNTGRT	4345	DATTAB	4141	ERR37	3467
BADFIL	5336	CNTRLC	4554	DEFSET	0451	ERR38	3700
BADSEC	0000	CNTRLF	4452	DEVCOD	0147	ERRFLG	0127
BADSN	2472	CNTRLG	4451	DIGCNT	5240	ERRHAN	4461
BFCNT2	6675	CNTRLQ	4242	DIGIN	4346	ERRLIM	0151
BFLUP2	6665	CNTRLS	4240	DIGITS	5237	ERROR	4446
BGMS2	2142	COMP30	1547	DNCH33	2345	ERRPC	0115
BRUSHC	0003	COMPOS	5506	DNCH37	3323	ERRTYP	4143
BRUSHH	0010	COMRET	5535	DNCH38	3600	ESIA	6055
BSDOS	2563	CON29B	0751	DRNLOD	1170	EXAC2	1325
BSVALD	0156	CON33A	2414	DROPDR	4161	EXIT34	2623
BSW	7002	CON34A	2541	DROPNG	0553	EXMAIN	0646
BUF2	4200	CON35A	2672	DRPCHK	4067	EXPACT	1617
BUF3	4600	CON36A	3045	DRSLER	0001	FATAL	5741
BUF4	5200	CON36B	3221	DRVOER	7137	FATLER	1143
BUF6	6200	CON37A	3405	DRVACT	7126	FIL2KB	6657
BUF7	6600	CON38A	3633	DRVLUP	0234	FILBUF	4472
BUF8	7200	CONRET	4243	DRVNO	1262	FILBUL	6103
BUFAGN	6543	CONSOL	4447	DRVNUM	0116	FILL35	2713
BUFCNT	6755	COVERD	0040	EAASB	1775	FILLEQ	1074
BUFS2K	6734	CRCDC	2723	EAMACB	2113	FILLER	0023

FILLUP	5345	IOT17	4721	KIE	6035	MANINT	0154
FRCHNF	6025	IOT2	4660	KSF	6031	MESSAGE	4506
FRCRET	6073	IOT3	4663	LAST12	7322	MESAGX	4600
FRSTDR	0476	IOT4	4666	LASTIN	0144	MESBSW	4621
FRSTSB	1003	IOT5	4671	LIMEXC	0525	MESCNT	4643
FSTTST	0534	IOT7	4674	LIMITS	0311	MESLUP	4606
GET4	6004	IOTCNG	0463	LISN	4505	MESSAV	4644
GETNUM	4455	IOTLUP	0465	LISN1	4516	MODE8	1000
GETRES	4453	IOTTAB	7143	LISN2	4527	ML	7421
GETSR	4442	JMPPM1	4470	LISN3	4542	NU	5170
GETSTA	4460	JMPUP	4764	LISNUM	4532	NOCLK	2746
GROSCCT	5133	K0777	0161	LLLTHL	0755	NOCRC	2677
GRSCNT	5134	K10	4553	LNFEED	4340	NODRVS	2254
GTNMLP	5205	K100	0370	LODHED	0002	NOER34	2616
HCW1	0021	K1000	0176	LOLIM	0152	NOHNF	2414
HCW2	0022	K100A	4646	LOOP36	3114	NOIN	4317
HD1HNF	0655	K1015	0166	LOOPPT	0126	NON1WD	2611
HD1T30	1400	K1016	0167	LPRQST	0130	NONZ12	2260
HDSLCT	0100	K1017	3143	LUP2K	6744	NONZ8	2061
HEAD	0140	K10A	4160	LUPSET	4023	NOPMES	4776
HEAD1	2000	K144	5503	M10	0171	NOPRNT	0142
HEDCUR	0100	K15	3145	M1000	1034	NORDY3	3036
HEDCYL	1355	K1500	3144	M14	3146	NORESE	3000
HEDOUT	0020	K16	6304	M1774	1337	NORQ30	1370
HIL1M	0153	K177	4547	M20	0762	NORQMA	0733
HLTCHK	4115	K2	1121	M200	1766	NORQSW	1013
HNFCOA	0610	K20	4443	M203	4552	NOTAL1	1323
HNFCOB	0637	K200	0165	M207	4217	NOTRDY	1221
HNFCOC	0662	K212	4461	M21	1336	NXTC33	2312
HNFCON	0600	K215	4460	M212	4550	NXTC37	3270
HNFFERR	1415	K24	0536	M223	4442	NXTC38	3526
HNFRQ1	0606	K240	4642	M251	2366	NXTD35	2656
IMPINT	4761	K260	5605	M252	2736	NXTDRV	3710
INCC33	2361	K2777	0761	M253	2162	NXTGRP	6621
INCC37	3340	K3	5504	M270	4540	NXTP30	1356
INCC38	3614	K300	0533	M3	4551	OK2TYP	4426
INCHNF	0623	K357	1765	M30	6437	ONLISN	4216
INFLG	4347	K377	0163	M376	3366	OPT1	0146
INIC33	2302	K3A	4645	M377	0173	OPT1AV	0563
INIC37	3261	K4	0763	M4	0371	OS8	4561
INIC38	3516	K400	3751	M40	0164	OUTSEQ	1315
INITRF	6600	K4000	5505	M400	0170	OVRWRT	3276
INVENT	2543	K5	6655	M404	5277	P2KBUF	6745
INVSEC	1310	K50	6440	M43	4641	P2SAVE	4373
IOTO	4650	K60	6171	M44	1017	P4SAVE	7040
IOT1	4653	K7	5606	M5	0175	PAPTER	4142
IOT10	4677	K70	6441	M50	0172	PAPTIM	4576
IOT11	4702	K7000	0177	M660	0174	PASCNT	0143
IOT12	4705	K77	4640	M7	2365	PAT1	0100
IOT13	4710	K7757	0535	M70	2753	PAT2	0140
IOT14	4713	KCC	6032	M777	0162	PAT3	0200
IOT15	4716	KCF	6030	MANDEC	0500	PAT4	0240

PAT5	0300	REPERR	6305	SPACX2	5264	TRK1RT	6367
PAT6	0340	RESET	4457	SPCSUR	1015	TRKLUP	6334
PAT7	0400	RESPON	5175	SPERLM	1033	TRRTSE	6364
PAT8	0440	RESRET	5173	SPINDN	0007	TSF	6041
PATPTA	7162	RETSET	4020	SPINUP	0001	TSTABR	1247
PATPTR	0157	RLCA	4427	SPL	6102	TSTDRV	0543
PATTRN	0160	RLCB	4430	SPOSP	0551	TWOSPA	1106
PBADSE	0721	RLDC	4424	SPUPTO	0010	TYPE	4513
PBSF	6370	RLMA	4426	STAOK	5053	UNLODH	0006
PBSM1	6733	RLSA	4431	STATER	1712	UPARG	4450
PBUF2	3110	RLSD	4425	STECCE	1735	UPARRC	1070
PC	1257	RLSE	4441	SURFAC	0150	UPARRG	1072
PCINT	2206	RLWC	4432	SURMES	0353	US1SUR	1001
PCONLY	2774	RLWREN	1125	SWRMSG	1102	USALCY	0673
PDRVOE	4157	ROTHNF	0566	T29CON	1000	USE62	0631
PERNUM	4156	RQAL35	2670	T31LUP	1627	USEALL	0434
PIOTS	0532	RRCA	4435	T32ACO	2200	USEDEF	0601
PLWRLK	1110	RRCB	4436	T32CON	2000	USLOLM	0705
PNTRBF	6436	RRER	4433	T33CON	2400	USUPLM	0737
POWER	2242	RRSA	4437	T34CON	2600	VOLUME	0002
PRDBFM	7064	RRSI	4440	T36CON	3200	VT278	6000
PRDBUF	0145	RRWC	4434	T37CON	3400	VTCHK	7106
PRECYL	1333	RTRY31	1600	TABENT	4137	WAIT1	5270
PRESEC	1334	SIE37	3501	TCF	6042	WAT300	5755
PRNT1	4507	S34CHK	2527	TEMP1	0131	WD1WD4	1637
PRNT2	4510	SACH30	1461	TEMP2	0132	WDNOSA	1673
PRNT4	4511	SANAE1	2646	TEMP3	0133	WR12RE	6261
PSECTB	6656	SANAE2	2441	TEMP4	0134	WR8RET	6217
PSWR	0020	SAVC35	2641	TEMP5	0135	WRDERR	0200
PWB31	1603	SAVNEW	1246	TEST0	0537	WRENWT	5300
PWB34B	2543	SCHCON	1144	TEST29	0667	WRERET	5320
PWRB34	2467	SCHRET	1161	TEST30	1340	WRGATE	0004
PWRB36	3064	SCNINT	7065	TEST31	1554	WRIT12	4501
PWRBZ8	1716	SCOP34	2624	TEST32	1712	WRIT8	4477
PWRFAL	4751	SCOPE	4445	TEST33	2271	WRITE1	7114
PWRTBU	1414	SCOPIN	0522	TEST34	2443	WRLOAT	3117
PWRUP	4765	SECTAB	7132	TEST35	2625	WRLOCK	0040
QESMRK	1060	SECTOR	0141	TEST36	3031	WRTBUF	5600
RBCHSA	1077	SEEKOK	5477	TEST37	3246	WRTCNT	6435
RD12RE	6302	SEEKTO	4462	TEST38	3505	WRTLUP	6410
RD8RET	6240	SEKCNT	0004	TICK	4444	WRTTRK	4474
RDBDRT	1120	SEKCON	5464	TIE	6045	WTRRET	6434
RDYRET	5551	SEKFAL	3022	TIKRET	4572	WZITFG	4456
RDYWAT	5536	SEKLIN	0005	TIMCHK	4467	XAPTCH	5144
READ12	4502	SEKRET	5500	TIMCNT	5132	XAPTCO	6676
READ30	1477	SERCH	1122	TIMOUT	0020	XCHKTR	6442
READ8	4500	SETINT	7072	TIMRET	5127	XCLRRB	7041
READF	6612	SETTIM	4466	TLS	6046	XCMPPA	6124
REDBAD	1020	SFTWER	2275	TOCK	4575	XCONSO	4200
REDBUF	3600	SKOCYL	4463	TOOMBS	1231	XCRLF	4444
REDHDR	4465	SKHCYL	4464	TUOMNY	5234	XCTRLF	4244
REPE30	1410	SPACE2	4512	TRK1GO	6324	XCTRLG	4305

XCYLBA	6706	YNCONC	5667
XDATER	7000	YNCOND	5716
XENTVA	5241	YNCONC	5730
XERRHA	5607	YNEXIT	5732
XERROR	4032	YNLOAD	5747
XFILBU	6074	ZERO12	2072
XGETNU	5200	ZFILER	3102
XGETRE	5152		
XGETSR	5135		
XGETST	5025		
XJMPPM	4741		
XLISN	4463		
XPRNT1	5600		
XPRNT2	4362		
XPRNT4	7027		
XREAD1	6263		
XREAD8	6221		
XREDHD	5321		
XRESET	5012		
XRLCA	4662		
XRLCB	4665		
XRLDC	4647		
XRLMA	4657		
XRLSA	4670		
XRLSD	4652		
XRLSE	4720		
XRLWC	4673		
XRRCA	4704		
XRRCB	4707		
XRRER	4676		
XRRSA	4712		
XRRSI	4715		
XRRWC	4701		
XSCOPE	4000		
XSEEKT	5400		
XSETT1	5062		
XSKOCY	5552		
XSKHCY	5564		
XSTACH	4725		
XTICK	4562		
XTIMCH	5105		
XTYPE	4400		
XUPARG	4275		
XWRIT1	6242		
XWRIT8	6200		
XWRTTR	6400		
XWZITF	4350		
YES	5167		
YESIN	4323		
YNCMRY	5752		
YNCNTR	5744		
YNCONB	5625		

(

(

(

1