

IDENTIFICATION

PRODUCT CODE: AC-F370B-MA

PRODUCT NAME: AJRLJB0 RL0-A/RL02 DR CPT VFYR

PRODUCT DATE: SEPT 1981

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: DAVID ORIN

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 DOCUMENT.

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

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

COPYRIGHT (C) 1978,1981, BY DIGITAL EQUIPMENT CORPORATION

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
- 2.1 HARDWARE
- 2.2 STORAGE
- 2.3 PREREQUISITE SOFTWARE
- 2.4 RL8A/RL02 REGISTER DEFINITIONS
- 3.0 RESTRICTIONS
- 4.0 STANDARD TEST PROCEDURE
- 4.1 LOADING THE PROGRAM
- 4.2 PROGRAM INITIALIZATION
 - 4.2.1 STARTING ADDRESS
 - 4.2.2 OPERATOR PROMPTS/RESPONSE
- 4.3 DATA PATTERNS
- 4.4 END OF PASS
- 4.5 OPERATION OF CONSOLE PACKAGE
- 4.6 OVER-WRITE TEST DESCRIPTION
- 4.7 ADJACENT CYLINDER TEST DESCRIPTION
- 4.8 EXAMPLE OPERATIONS
- 5.0 ERRORS
 - 5.1 FATAL ERRORS
 - 5.2 SOFT ERRORS
 - 5.3 HARD ERRORS
 - 5.4 ERROR LOGGING
 - 5.5 RETRIES
 - 5.6 ERROR REPORTING
- 5.7 BAD SECTOR FILES UNREADABLE
- 5.8 OVER-WRITE TEST ERROR
- 5.9 ADJACENT CYLINDER TEST ERROR
- 6.0 SWITCH REGISTER SETTINGS
- 6.1 NORMAL OPERATION SWITCHES
- 6.2 ERROR RELATED SWITCHES
- 7.0 REVISIONS
- 8.0 ADDITIONAL INFORMATION
- 8.1 HALTS
- 8.2 DRIVE ERRORS
- 8.3 POWER FAIL
- 9.0 PROGRAM LISTING

THE PURPOSE OF THE RL02 DRIVE COMPATIBILITY PROGRAM IS TO DEMONSTRATE THAT THE RL02 DISK DRIVES ON ONE OR MORE SUBSYSTEMS ARE COMPATIBLE FOR DATA STORAGE AND RETRIEVAL. THE PROGRAM TESTS THAT INDIVIDUAL DRIVES CAN READ OVER-WRITE, AND ADJACENT WRITE DATA WRITTEN BY OTHER DRIVES. THIS PROGRAM WILL DEMONSTRATE DRIVE COMPATIBILITY BY PROVING THAT THE FOLLOWING OPERATIONS ARE POSSIBLE:

- A. EACH DRIVE CAN WRITE ON THE SELECTED TRACKS
- B. EACH DRIVE CAN WRITE ADJACENT TRACKS ON BOTH SURFACES OF THE TEST PACK.
- C. EACH DRIVE CAN OVER-WRITE DATA WRITTEN BY ANOTHER DRIVE
- D. EACH DRIVE CAN WRITE AFTER SEEKING FROM EITHER DIRECTION
- E. EACH DRIVE CAN READ ALL DATA

COMPATIBILITY CAN BE TESTED USING FROM 2 TO 4 DRIVES, DISTRIBUTED ON FROM 1 TO 4 RL02 SUBSYSTEMS. ONLY 2 RL8A OR RL278 CONTROLLERS MAY BE USED ON ANY ONE PDP-8 SYSTEM.

2.0 REQUIREMENTS: -----

2.1 HARDWARE: -----

THE FOLLOWING HARDWARE IS REQUIRED FOR EXECUTION OF THIS PROGRAM:

PROCESSOR: PDP-8A,8E,8F,8M OR VT278

MEMORY: MINIMUM OF 8K READ/WRITE MEMORY (FIELD 0 AND FIELD 1)

OPTIONS: 1 OR MORE RL8A/RL02 OR RL278/RL02 DISK CONTROLLERS

SPECIAL: 2 TO 4 RL02 DISK DRIVES
1 RL02 DISK PACK WITH NOT MORE THAN 16 BAD SECTORS

THE PROGRAM MUST BE LOADED INTO FIELD 0 AND 1 CONTAINING 4K OF READ/WRITE MEMORY IN EACH FIELD. THE PROGRAM OCCUPIES LOCATIONS 0000 THROUGH 7577 AND 10000 THROUGH 17577.

2.3 PREREQUISITE SOFTWARE: -----

THE FOLLOWING PROGRAMS MUST HAVE BEEN RUN SUCCESSFULLY PRIOR TO RUNNING THIS PROGRAM:

- A. ALL BASIC CPU AND MEMORY TESTS
- B. RL8A/RL02 OR RL278/RL02 DISKLESS DIAGNOSTIC
- C. RL8A/RL02 OR RL278/RL02 DRIVE TESTS (PART 1 AND PART 2)
- D. RL8A/RL02 OR RL278/RL02 PERFORMANCE EXERCISER

2.4 RL8A/RL02 REGISTER DEFINITIONS: -----

COMMAND REGISTER A

BIT	DEFINITION
---	-----
0=0	HEAD DIRECTION AWAY FROM CENTER (LOWER CYLINDER)
0=1	HEAD DIRECTION TOWARD CENTER (HIGHER CYLINDER)
1=0	SELECT UPPER HEAD (SURFACE 0)
1=1	SELECT LOWER HEAD (SURFACE 1)
2=SPARE	
3=RESERVED FOR FUTURE EXPANSION	
4 THRU 11	CYLINDER ADDRESS/DIFFERENCE WORD

COMMAND REGISTER B

BIT	DEFINITION
---	-----
0=1	MAINTENANCE INHIBIT (NOT USED IN THIS PROGRAM)
1=1	MAINTENANCE LOOP DAR TO SILO (NOT USED IN THIS PROGRAM)
2=0	12 BIT DATA MODE
2=1	8 BIT DATA MODE
3=1	INTERRUPT ENABLE
4:5	DRIVE SELECT
6:8	DATA BUFFER FIELD (EMA)

BITS	9 10 11	COMMAND
-----	-----	-----
0	0 0 0	(0) MAINTENANCE
0	0 0 1	(1) RESET
0	0 1 0	(2) GET STATUS
0	0 1 1	(3) SEEK
1	0 0 0	(4) READ HEADER
1	0 0 1	(5) WRITE DATA
1	1 0 0	(6) READ DATA
1	1 1 1	(7) READ DATA NO HEADER CHECK

ERROR REGISTER

BIT	DEFINITION
0=1	CRC ERROR (CYCLIC REDUNDANCY CHECK)
1=1	OPI ERROR (OPERATION INCOMPLETE)
2=1	DATA LATE/HEADER NOT FOUND
3 THRU 9	NOT USED
10=1	DRIVE ERROR
11=1	DRIVE READY

BITS	0 1 2	ERROR
0 0 0	(0)	
0 0 1	(1) DLT DATA LATE	
0 1 0	(2) OPI OPERATION INCOMPLETE	
0 1 1	(3) HNF HEADER NOT FOUND	
1 0 0	(4) DCRC DATA CRC ERROR	
1 0 1	(5)	
1 1 0	(6) HCRC HEADER CRC	
1 1 1	(7)	

DISK STATUS WORD #1

BIT	DEFINITION
5=1	HEAD SELECT ERROR (COMMAND REG. A DID NOT MATCH HEAD SELECTED)
6=1	COVER OPEN
7=0	HEADS HOME
7=1	HEADS OVER DISK
8=0	BRUSH OVER DISK
8=1	BRUSH HOME

BITS	9 10 11	DRIVE STATE
0 0 0	(0) LOAD STATE	
0 0 1	(1) SPIN-UP	
0 1 0	(2) LOAD HEADS	
0 1 1	(3) BRUSH CYCLE	
1 0 0	(4) SEEK-TRACK COUNTING (SEEKING)	
1 0 1	(5) SEEK LINEAR MODE (TRACKING)	
1 1 0	(6) UNLOAD HEADS	
1 1 1	(7) SPIN-DOWN	

BIT	DEFINITION
---	-----
4=1	WRITE DATA ERROR
5=1	HEAD CURRENT ERROR
6=1	WRITE-LOCKED DRIVE
7=1	SEEK TIME-OUT
8=1	SPIN-UP TIME-OUT
9=1	WRITE GATE ERROR
10=1	VOLUME CHECK (DRIVE SPUN DOWN THEN UP)
11=1	DRIVE SELECT ERROR (TWO OR MORE HAVE SAME DRIVE NUMBER PLUG)

3.0 RESTRICTIONS:

- A. AN OPERATOR'S CONSOLE TERMINAL MUST BE PRESENT AND ON-LINE (TELETYPE OR EQUIVALENT)
- B. THE DISK PACK USED MUST HAVE NO MORE THAN 16 BAD SECTORS LISTED IN THE FIELD AND FACTORY BAD SECTOR FILES COMBINED.
- C. THE BAD SECTOR FILES MUST BE INTACT AND UP-TO-DATE OR DATA ERRORS WILL RESULT. IT IS RECOMMENDED THAT THE SCRATCH PACK USED FOR RUNNING THIS PROGRAM HAVE NO BAD HEADERS, OR HEADER CRC ERRORS MAY RESULT.
- D. IF ALL 5 FACTORY AND/OR ALL 5 FIELD BAD SECTOR FILES CANNOT BE READ WITHOUT ERRORS, AN ERROR MESSAGE WILL RESULT AND A DIFFERENT PACK MUST BE USED. (SEE SECTION 5.7 FOR FURTHER INFORMATION).
- E. IF THE BAD SECTORS ON THE TEST PACK ARE DISTRIBUTED IN SUCH A MANNER THAT 5 ADJACENT PERFECT TRACKS CANNOT BE FOUND IN ONE OF THE 10 TEST AREAS ON THE TEST PACK, AN ERROR MESSAGE WILL RESULT AND A DIFFERENT TEST PACK MUST BE USED.

4.0 STANDARD TEST PROCEDURE:

4.1 LOADING THE PROGRAM:

LOAD THE PROGRAM INTO FIELDS 0 AND 1 USING THE STANDARD BINARY OR ABSOLUTE LOADER TECHNIQUES.

THE PROGRAM, WHEN LOADED, IS INITIALIZED TO RUN AS FOLLOWS:

- A. NO. HARDWARE SWITCH REGISTER (USES LOCATIONS 0020 AND 10020) AS A PSEUDO SWITCH REGISTER MODIFIED BY TYPING CTRL G). TO USE THE HARDWARE SWITCH REGISTER, MODIFY THE CONTENTS OF LOCATION 0021 IN FIELD 0 TO 4000. THIS IS HARDWARE CONFIGURATION WORD 1 AND SETTING BIT 0 TO A 1 WILL SIGNIFY THAT THE PROGRAM IS TO USE THE HARDWARE SWITCH REGISTER.
- B. 8K OF READ/WRITE MEMORY - THE MINIMUM REQUIREMENT
- C. CONSOLE PACKAGE ACTIVE
- D. DEVICE CODES: 60 AND 61

4.2.1 STARTING ADDRESS:

THE ONLY STARTING ADDRESS IS LOCATION 0200 IN FIELD 0

4.2.2 OPERATOR PROMPTS/RESPONSE:

THE FOLLOWING IS THE DIALOGUE THE PROGRAM WILL CARRY ON WITH THE OPERATOR FOR PROGRAM INITIALIZATION:

TITLE: "AJRLJB RL0A/RL02 DRIVE COMPATIBILITY TEST"

* Q1: *

"UNIT 0 = DRIVE (0-7)?"

RESPONSE: 1 OCTAL DIGIT (0 THRU 7). THIS WILL BE THE FIRST TEST UNIT DRIVE NUMBER.

RESPONSE: CARRIAGE RETURN Q1 WILL BE REASKED BECAUSE THE FIRST TEST UNIT MUST BE SPECIFIED.

* Q2: *

"UNIT 0 ON THIS SYSTEM (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT (Y)ES THE FIRST TEST UNIT (DRIVE) IS ON THE CURRENT PDP-8 SYSTEM. Q3 IS NEXT.

RESPONSE: "N" SPECIFIES THAT (N)O THE FIRST TEST UNIT IS NOT ON THE CURRENT PDP-8 SYSTEM. Q4 IS NEXT.

RESPONSE: "Y" SPECIFIES THAT CONTROLLER DEVICE CODES 60 AND 61 WILL BE USED FOR THE FIRST TEST UNIT ON THE CURRENT SYSTEM.

RESPONSE: "N" SPECIFIES THAT CONTROLLER DEVICE CODES 62 AND 63 WILL BE USED FOR THE FIRST TEST UNIT ON THE CURRENT SYSTEM.

* Q4: * "UNIT 1 = DRIVE (0-7)?"

RESPONSE: 1 OCTAL DIGIT (0 THRU 7). THIS WILL BE THE SECOND UNIT UNDER TEST DRIVE NUMBER.

RESPONSE: CARRIAGE RETURN Q4 WILL BE REASKED BECAUSE THE SECOND TEST UNIT MUST BE SPECIFIED.

* Q5: * "UNIT 1 ON THIS SYSTEM (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT (Y)ES THE SECOND TEST UNIT (DRIVE) IS ON THE CURRENT SYSTEM. Q6 IS NEXT.

RESPONSE: "N" SPECIFIES THAT (N)O THE SECOND TEST UNIT (DRIVE) IS NOT ON THE CURRENT SYSTEM. Q7 IS NEXT.

* Q6: * "DEVICE CODE (60,61) (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT CONTROLLER DEVICE CODES 60 AND 61 WILL BE USED FOR THE SECOND TEST UNIT ON THE CURRENT SYSTEM.

RESPONSE: "N" SPECIFIES THAT CONTROLLER DEVICE CODES 62 AND 63 WILL BE USED FOR THE SECOND TEST UNIT.

* Q7: * "UNIT 2 = DRIVE (0-7)?"

RESPONSE: 1 OCTAL DIGIT (0 THRU 7). THIS WILL BE THE DRIVE NUMBER USED FOR THE THIRD TEST UNIT (DRIVE). Q8 IS NEXT.

RESPONSE: CARRIAGE RETURN SPECIFIES THAT ONLY 2 UNITS (DRIVES) ARE BEING TESTED UNITS 0 AND 1 WILL BE TESTED, UNITS 2 AND 3 WILL NOT BE TESTED. Q13 IS NEXT.

RESPONSE: "Y" SPECIFIES THAT THE THIRD UNIT UNDER TEST (DRIVE)
IS ON THE CURRENT PDP-8 SYSTEM. Q9 IS NEXT.

RESPONSE: "N" SPECIFIES THAT THE THIRD UNIT UNDER TEST (DRIVE)
IS NOT ON THE CURRENT SYSTEM. Q10 IS NEXT.

* Q9: * "DEVICE CODE (60,61) (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT CONTROLLER DEVICE CODES 60 AND 61 WILL
WILL BE USED FOR THE THIRD UNIT UNDER TEST.

RESPONSE: "N" SPECIFIES THAT CONTROLLER DEVICE CODES 62 AND 63 WILL
BE USED FOR THE THIRD UNIT UNDER TEST.

* Q10: * "UNIT 3 = DRIVE (0-7)?"

RESPONSE: 1 OCTAL DIGIT (0 THRU 7). THIS WILL BE THE DRIVE NUMBER
USED FOR THE FOURTH UNIT UNDER TEST. Q11 IS NEXT.

RESPONSE: CARRIAGE RETURN SPECIFIES THAT ONLY 3 UNITS (DRIVES)
ARE BEING TESTED. UNITS 0,1, AND 2
WILL BE TESTED. UNIT 3 WILL NOT BE TESTED.
Q13 IS NEXT.

* Q11: * "UNIT 3 ON THIS SYSTEM (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT THE FOURTH UNIT UNDER TEST
IS ON THE CURRENT PDP-8 SYSTEM.

RESPONSE: "N" SPECIFIES THAT THE FOURTH UNIT UNDER TEST
IS NOT ON THE CURRENT SYSTEM.

* Q12: * "DEVICE CODE (60,61) (Y/N)?"

RESPONSE: "Y" SPECIFIES THAT CONTROLLER DEVICE CODES 60 AND 61
WILL BE USED FOR THE FOURTH UNIT UNDER TEST.

RESPONSE: "N" SPECIFIES THAT CONTROLLER DEVICE CODES 62 AND 63
WILL BE USED FOR THE FOURTH UNIT UNDER TEST.

RESPONSE: 1 TO 4 OCTAL DIGITS FOLLOWED BY CARRIAGE RETURN.
THIS WILL MODIFY THE SOFTWARE SWITCH REGISTER.
(SEE SECTION 4.5).

RESPONSE: CARRIAGE RETURN THIS WILL LEAVE THE SOFTWARE
SWITCH REGISTER UNMODIFIED
AND CONTINUE THE PROGRAM.

* Q14: * "ARE YOU SURE (Y/N)?"

RESPONSE: "N" THIS WILL RESTART THE PROGRAM

4.3 DATA PATTERNS: -----

THE FOLLOWING ARE THE UNIQUE DATA PATTERNS USED THROUGHOUT THIS PROGRAM. ALL DATA TRANSFERS ARE PERFORMED IN 8 BIT MODE, 1 SECTOR AT A TIME. EACH UNIT UNDER TEST HAS AN ASSOCIATED UNIQUE DATA PATTERN. ALL SECTORS ARE COMPLETELY WRITTEN OR READ USING THESE PATTERNS. TWO 8 BIT THESE DATA PATTERNS ARE THE SAME AS THOSE USED ON PDP-11 SYSTEMS.

8 BIT MODE DATA -----

UNIT 0 PATTERN -----	UNIT 1 PATTERN -----	UNIT 2 PATTERN -----	UNIT 3 PATTERN -----
0325	0266	0155	0333
0125	0333	0266	0155
0325	0266	0155	0333
0125	0333,	0266	0155
ETC.	ETC.	ETC.	ETC.

WHEN ALL OF THE STEPS HAVE BEEN COMPLETED, THE MESSAGE:

"TEST COMPLETE - ALL DRIVES COMPATIBLE!"

WILL BE PRINTED. THE SR= QUESTION WILL THEN BE ASKED. IF A BUT THE INITIAL OPERATOR/RESPONSE DIALOGUE DESCRIBED IN SECTION 4.2.2 WILL NOT BE REASKED. IF A LINEFEED IS TYPED, THE PROGRAM WILL BE RESTARTED COMPLETELY.

4.5 OPERATION OF CONSOLE PACKAGE:

CONTROL G

THIS CONTROL CHARACTER IS USED TO EXAMINE THE CONTENTS OF THE SWITCH REGISTER BEING USED BY THE PROGRAM, (AS DEFINED BY BIT 0 OF HARDWARE CONFIGURATION WORD ONE,) AND TO PERMIT MODIFICATION OF THE PSEUDO SWITCH REGISTER CONTENTS. IF THE PROGRAM IS USING THE HARDWARE SWITCH REGISTER, MODIFICATION OF THE PSEUDO SWITCH REGISTER SHOULD HAVE NO EFFECT ON THE OPERATION OF THE PROGRAM.

WHEN CONTROL G IS TYPED, THE PROGRAM WILL PRINT AN UPARROW FOLLOWED BY A G TO INDICATE THAT IT IS RESPONDING TO A CONTROL G. THE PROGRAM WILL THEN EXECUTE A CARRIAGE RETURN AND LINE FEED AND PRINT SR=XXXX WHERE XXXX IS THE 4 OCTAL DIGITS REPRESENTING THE CURRENT CONTENTS OF THE SWITCH REGISTER BEING USED. THE PROGRAM WILL THEN SPACE OVER TWO POSITIONS AND WAIT FOR THE OPERATOR TO TYPE A CHARACTER OR CHARACTERS. THE FOLLOWING IS A LIST OF POSSIBLE INPUTS FROM THE OPERATOR AND THE RESULT OF EACH INPUT:

ANOTHER CONTROL G	PRINT UPARROW G, DO A CARRIAGE RETURN AND LINE FEED, AND PRINT SR=XXXX, WAIT FOR INPUT FROM OPERATOR
CONTROL C	RETURN TO MONITOR
CARRIAGE RETURN	RETURN TO THE PROGRAM, NO CHANGE TO PSEUDO SWITCH REGISTER, CPU FLAGS AND STATUS RESTORED
LINE FEED	RESTART PROGRAM, NO CHANGE TO PSEUDO SWITCH REGISTER
1 TO 4 OCTAL DIGITS FOLLOWED BY CARRIAGE RETURN	CHANGE PSEUDO SWITCH REGISTER AND RETURN TO THE PROGRAM, CPU FLAGS AND STATUS RESTORED
1 TO 4 OCTAL DIGITS FOLLOWED BY LINE FEED	CHANGE PSEUDO SWITCH REGISTER AND RESTART THE PROGRAM
1 TO 4 OCTAL DIGITS FOLLOWED BY CONTROL G	NO CHANGE TO PSEUDO SWITCH REGISTER PRINT UPARROW G, EXECUTE A CARRIAGE RETURN AND LINE FEED AND PRINT SR=XXXX, WAIT FOR OPERATOR INPUT
ALL OTHER INPUT (ILLEGAL CHARACTERS OR 5 OCTAL DIGITS)	NO CHANGE TO PSEUDO SWITCH REGISTER ECHO THE CHARACTER, PRINT A QUESTION MARK, DO A CARRIAGE RETURN LINE FEED, PRINT SR=XXXX, WAIT FOR OPERATOR INPUT

THIS CONTROL CHARACTER IS USED TO INHIBIT TRANSMISSION OF DATA TO THE OPERATOR'S TERMINAL. WHEN CONTROL S IS TYPED BY THE OPERATOR IT SHOULD NOT BE ECHOED. SOME TERMINALS HAVE HARDWARE WHICH AUTOMATICALLY SENDS A CONTROL S WHEN THE TERMINAL BUFFER IS FULL. THE BUFFER IS EMPTIED AS THE DATA IS PRINTED AND AFTER ALL OF THE DATA IN THE BUFFER IS PRINTED, THE TERMINAL SENDS A CONTROL Q SIGNIFYING IT IS READY TO RECEIVE MORE DATA. IF THE OPERATOR TYPES CONTROL S WHILE THE PROGRAM IS RUNNING TESTS, THE PROGRAM WILL CONTINUE TO RUN THE TESTS. IF THE OPERATOR TYPES CONTROL S WHILE A MESSAGE IS IN PROGRESS, THE PRINTOUT WILL BE INTERRUPTED AND THE PROGRAM WILL WAIT FOR THE OPERATOR TO TYPE A CONTROL Q (CONTINUE MESSAGE) OR A CONTROL C (RETURN TO MONITOR).

CONTROL Q

THIS CONTROL CHARACTER IS USED TO ENABLE TRANSMISSION OF DATA TO THE OPERATOR'S TERMINAL. WHEN CONTROL Q IS TYPED BY THE OPERATOR IT SHOULD NOT BE ECHOED. SOME TERMINALS HAVE HARDWARE WHICH AUTOMATICALLY SENDS A CONTROL Q WHEN THE TERMINAL BUFFER IS EMPTY AND THE TERMINAL IS READY TO RECEIVE MORE DATA. CONTROL Q IS USED TO COUNTERACT THE EFFECT OF A PREVIOUSLY TYPED CONTROL S. IF CONTROL S WAS NOT IN EFFECT PRIOR TO THE TYPING OF CONTROL Q, THE CONTROL Q SHOULD HAVE NO EFFECT UPON THE PROGRAM STATUS OR CPU STATUS.

CONTROL C

INSTRUCTS PROGRAM TO RETURN TO THE OPERATING SYSTEM MONITOR.

THE MAIN PURPOSE OF THE OVER-WRITE TEST IS TO VERIFY THAT ALL DRIVES CAN OVER-WRITE ALL OTHER DRIVES DATA AND READ BACK ALL DATA. BEFORE BEGINNING THE TEST, THE PROGRAM RETRIEVES THE BAD SECTOR FILE AND LOCATES 10 SETS OF 5 PERFECT ADJACENT TRACKS. THESE TRACKS ARE LOCATED IN 5 SETS OF 5 ON EACH OF THE TWO DISK SURFACES. THE 5 GROUPS OF 5 TRACKS ARE LOCATED IN 5 DIFFERENT POSITIONS ON THE DISK SURFACE, OUTER, ONE-QUARTER, MIDDLE, THREE-QUARTER, INNER. THE RANGE OF TRACKS THE PROGRAM SELECTS FROM (IN OCTAL) ARE AS FOLLOWS:

OUTER: TRACKS 0-20
 1/4: TRACKS 70-110
 MIDDLE: TRACKS 170-210
 3/4: TRACKS 270-310
 INNER: TRACKS 356-376

THE OPTIMUM SETS OF TRACKS (IN OCTAL) ARE AS FOLLOWS:

OUTER: TRACKS 0,1,2,3,4
 1/4: TRACKS 70,71,72,73,74
 MIDDLE: TRACKS 170,171,172,173,174
 3/4: TRACKS 270,271,272,273,274
 INNER: TRACKS 356,357,360,361,362

THESE 10 SETS OF 5 TRACKS ARE NECESSARY FOR THE ADJACENT CYLINDER TEST. ONLY THE MIDDLE TRACK OF EACH SET OF 5 TRACKS IS USED FOR THE OVER-WRITE TEST. THEREFORE, THE FOLLOWING TRACKS ARE USED FOR THE OVER-WRITE TEST IF NO BAD SECTORS EXIST:

OUTER: TRACK 2
 1/4: TRACK 72
 MIDDLE: TRACK 172
 3/4: TRACK 272
 INNER: TRACK 360

THE SECTORS USED ON EACH OF THESE TRACKS ARE 34 THRU 47 AND 0 THRU 3 (OCTAL) FOR THE OVER-WRITE TEST.

THE DESCRIPTIONS AND EXAMPLES (SECTION 4.8) REFER TO "UNIT" NUMBERS. EACH UNIT NUMBER (0,1,2, OR 3) HAS AN ASSOCIATED DRIVE NUMBER (0 THRU 7) AS DETERMINED IN THE INITIAL DIALOGUE (SECTION 4.2.2). THE PROGRAM WILL

2 DRIVES: UNIT 0, UNIT 1, UNIT 0
 3 DRIVES: UNIT 0, UNIT 1, UNIT 2, UNIT 1, UNIT 0
 4 DRIVES: UNIT 0, UNIT 1, UNIT 2, UNIT 3, UNIT 2, UNIT 1, UNIT 0

THE PROGRAM IS DIVIDED INTO STEPS TO ALLOW THE RL02 TEST PACK TO BE MOVED FROM DRIVE TO DRIVE (UNIT TO UNIT). THE UNIT NUMBERS ARE USED INTERNALLY BY THE PROGRAM FOR INDEXING INTO TABLES. THE DRIVE NUMBERS ARE USED ONLY FOR ERROR REPORTING, I/O OPERATIONS, AND OPERATOR PROMPTS.

STEP NUMBER -----	SEEK DIRECTION -----	UNIT N OVER UNIT M (N/M) -----	SECTORS WRITTEN (OCTAL) -----
1	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 INITIALIZE UNIT 0 INITIALIZE	34,35,36,37,40,41,42,43 44,45,46,47,0,1,2,3
2	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 0 UNIT 1 OVER UNIT 0	44,45,46,47 0,1,2,3
3	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 1 UNIT 0 OVER UNIT 1	46 47

THE STEPS FOR TESTING 3 DRIVES ARE AS FOLLOWS:

STEP NUMBER -----	SEEK DIRECTION -----	UNIT N OVER UNIT M (N/M) -----	SECTORS WRITTEN (OCTAL) -----
1	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 INITIALIZE UNIT 0 INITIALIZE	34,35,36,37,40,41,42,43 44,45,46,47,0,1,2,3
2	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 0 UNIT 1 OVER UNIT 0	44,45,46,47 0,1,2,3
3	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 2 OVER UNIT 0 UNIT 2 OVER UNIT 0	40,41 42,43
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 2 OVER UNIT 1 UNIT 2 OVER UNIT 1	0,1 2,3
4	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 2 UNIT 1 OVER UNIT 2	40 41
5	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 1 UNIT 0 OVER UNIT 1	46 47
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 2 UNIT 0 OVER UNIT 2	2 3

STEP NUMBER -----	SEEK DIRECTION -----	UNIT N OVER UNIT M (N/M) -----	SECTORS WRITTEN (OCTAL) -----
1	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 INITIALIZE UNIT 0 INITIALIZE	34,35,36,37,40,41,42,43 44,45,46,47,0,1,2,3
2	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 0 UNIT 1 OVER UNIT 0	44,45,46,47 0,1,2,3
3	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 2 OVER UNIT 0 UNIT 2 OVER UNIT 0	40,41 42,43
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 2 OVER UNIT 1 UNIT 2 OVER UNIT 1	0,1 2,3
4	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 3 OVER UNIT 0 UNIT 3 OVER UNIT 0	36 37
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 3 OVER UNIT 2 UNIT 3 OVER UNIT 2	42 43
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 3 OVER UNIT 1 UNIT 3 OVER UNIT 1	46 47
5	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 2 OVER UNIT 3 UNIT 2 OVER UNIT 3	36 37
6	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 2 UNIT 1 OVER UNIT 2	40 41
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 1 OVER UNIT 3 UNIT 1 OVER UNIT 3	46 47
7	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 3 UNIT 0 OVER UNIT 3	42 43
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 1 UNIT 0 OVER UNIT 1	46 47
	IN FROM TRACK 0 OUT FROM TRACK 377	UNIT 0 OVER UNIT 2 UNIT 0 OVER UNIT 2	2 3

THE PURPOSE OF THE ADJACENT CYLINDER WRITE TEST IS TO VERIFY THAT ALL DRIVES UNDER TEST CAN WRITE A SECTOR WITHOUT DISTURBING ANOTHER DRIVES DATA ON A SECTOR WHICH IS ADJACENT TO (AND ON AND ADJACENT CYLINDER) THE SECTOR WRITTEN BY THE UNIT UNDER TEST. THIS TEST USES THE SAME 10 SETS OF 5 ADJACENT TRACKS AS DESCRIBED IN SECTION 4.6.

THE FOLLOWING SECTORS (IN OCTAL) ARE USED ON EACH SET OF 5 TRACKS:

TRACK:	SECTORS:
-----	-----
N-2	20-47
N-1	12 THRU 41
N	4 THRU 33
N+1	46, 47, AND 0 THRU 25
N+2	40 THRU 47, 0 THRU 17

TRACK "N" ABOVE REFERS TO THE CENTER TRACK OF EACH SET OF 5 TRACKS. THE NUMBERING OF THE SECTORS ON EACH TRACK IS OFFSET BY 6. THUS SECTOR 20 ON TRACK N-2 IS ADJACENT TO SECTOR 12 ON TRACK N-1. SECTOR 12 ON TRACK N-1 IS ADJACENT TO SECTOR 4 ON TRACK N. THE FOLLOWING IS A TABLE OF THE TRACKS, THE SECTORS USED ON EACH TRACK INDICATING WHICH SECTORS ARE ADJACENT, THE UNIT NUMBER WHICH WILL WRITE EACH SECTOR, AND THE DIRECTION OF THE SEEK PRIOR TO WRITING EACH SECTOR:

SECTOR	UNIT	SEEK	SECTOR	UNIT	SEEK	SECTOR	UNIT	SEEK	SECTOR	UNIT	SEEK	SECTOR	UNIT	SEEK
20	2	IN	12	1	IN	4	0	IN	46	1	IN	40	2	IN
21	2	OUT	13	1	IN	5	0	IN	47	1	IN	41	2	OUT
22	3	IN	14	1	IN	6	0	IN	0	1	IN	42	3	IN
23	3	OUT	15	1	IN	7	0	IN	1	1	IN	43	3	OUT
24	0	IN	16	1	OUT	10	0	IN	2	1	OUT	44	0	IN
25	0	OUT	17	1	OUT	11	0	IN	3	1	OUT	45	0	OUT
26	NOT USED		20	1	OUT	12	0	IN	4	1	OUT	46	NOT USED	
27	NOT USED		21	1	OUT	13	0	IN	5	1	OUT	47	NOT USED	
30	3	IN	22	2	IN	14	0	IN	6	2	IN	0	3	IN
31	3	OUT	23	2	IN	15	0	IN	7	2	IN	1	3	OUT
32	1	IN	24	2	IN	16	0	IN	10	2	IN	2	1	IN
33	1	OUT	25	2	IN	17	0	IN	11	2	IN	3	1	OUT
34	0	IN	26	2	OUT	20	0	OUT	12	2	OUT	4	0	IN
35	0	OUT	27	2	OUT	21	0	OUT	13	2	OUT	5	0	OUT
36	NOT USED		30	2	OUT	22	0	OUT	14	2	OUT	6	NOT USED	
37	NOT USED		31	2	OUT	23	0	OUT	15	2	OUT	7	NOT USED	
40	2	IN	32	3	IN	24	0	OUT	16	3	IN	10	2	IN
41	2	OUT	33	3	IN	25	0	OUT	17	3	IN	11	2	OUT
42	1	IN	34	3	IN	26	0	OUT	20	3	IN	12	1	IN
43	1	OUT	35	3	IN	27	0	OUT	21	3	IN	13	1	OUT
44	0	IN	36	3	OUT	30	0	OUT	22	3	OUT	14	0	IN
45	0	OUT	37	3	OUT	31	0	OUT	23	3	OUT	15	0	OUT
46	NOT USED		40	3	OUT	32	0	OUT	24	3	OUT	16	NOT USED	
47	NOT USED		41	3	OUT	33	0	OUT	25	3	OUT	17	NOT USED	

EXAMPLE 1: IF TESTING COMPATIBILITY BETWEEN 3 DRIVES ON 1 PDP-8 SYSTEM
USING 1 RL8A CONTROLLER, THE PROGRAM/USER DIALOGUE WOULD
APPEAR AS FOLLOWS:

AJRLDA RL8A/RL02 DRIVE COMPATIBILITY TEST

UNIT 0 = DRIVE (0-7)? 0
UNIT 0 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 1 = DRIVE (0-7)? 1
UNIT 1 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 2 = DRIVE (0-7)? 2
UNIT 2 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 3 = DRIVE (0-7)? <CR>
SR=0000

ARE YOU SURE (Y/N)? Y

MOUNT PACK ON DRIVE 0
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 2
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 0
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

TEST COMPLETE - ALL DRIVES COMPATIBLE!

SR=0000

TO MOUNT THE PACK ON A DIFFERENT DRIVE (UNIT), WAIT FOR THE DRIVE TO SPIN UP AND BECOME READY, WITH THE DRIVE WRITE ENABLED. WHEN THE DRIVE IS READY, THE OPERATOR TYPES CARRIAGE RETURN <RETURN> AND THE NEXT TEST STEP IS PERFORMED. THE PACK IS LEFT ON THIS DRIVE UNTIL THE PROGRAM INSTRUCTS THE OPERATOR TO MOVE THE PACK TO A DIFFERENT DRIVE. IF THE DRIVE IS NOT READY WHEN THE OPERATOR TYPES <RETURN>, THE FOLLOWING MESSAGE IS PRINTED:

DRIVE NOT READY

MOUNT PACK ON DRIVE X
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .

EXAMPLE 2: TO TEST 4 DRIVES, 2 DRIVES ON ONE PDP-8 SYSTEM AND 2 DRIVES
 ON A DIFFERENT PDP-8 SYSTEM:

- A. LOAD THIS PROGRAM INTO BOTH PDP-8 SYSTEM MEMORIES AND START 200.
- B. PERFORM THE DIALOGUES SHOWN BELOW ON THE 2 SYSTEMS
- C. MOVE THE TEST PACK AS INSTRUCTED BY THE PROGRAM RUNNING ON EACH SYSTEM
- D. ONLY 1 RL01 PACK IS USED FOR THE TEST

COMPATIBLE WITH DRIVES 0 AND 1 ON PDP-8 SYSTEM #2.

AJRLDA RL9A/RL01 DRIVE COMPATIBILITY TEST

UNIT 0 = DRIVE (0-7)? 0
UNIT 0 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 1 = DRIVE (0-7)? 1
UNIT 1 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 2 = DRIVE (0-7)? 0
UNIT 2 ON THIS SYSTEM (Y/N)? N

UNIT 3 = DRIVE (0-7)? 1
UNIT 3 ON THIS SYSTEM (Y/N)? N
SR=0000

START WITH STEP NUMBER (1-7)? 1

ARE YOU SURE (Y/N)? Y

MOUNT PACK ON DRIVE 0 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 0 OTHER SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1 OTHER SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 0 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

TEST COMPLETE - ALL DRIVES COMPATIBLE!

SR=0000

AJRLDA RL0A/RL01 DRIVE COMPATIBILITY TEST

UNIT 0 = DRIVE (0-7)? 0
UNIT 0 ON THIS SYSTEM (Y/N)? N

UNIT 1 = DRIVE (0-7)? 1
UNIT 1 ON THIS SYSTEM (Y/N)? N

UNIT 2 = DRIVE (0-7)? 0
UNIT 2 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y

UNIT 3 = DRIVE (0-7)? 1
UNIT 3 ON THIS SYSTEM (Y/N)? Y
DEVICE CODE (60,61) (Y/N)? Y
SR=0000
START WITH STEP NUMBER (1-7)? 3

ARE YOU SURE (Y/N)? Y

MOUNT PACK ON DRIVE 0 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 0 THIS SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>

MOUNT PACK ON DRIVE 1 OTHER SYSTEM
WAIT FOR DRIVE READY

TYPE <RETURN> TO CONTINUE . . .<CR>
MOUNT PACK ON DRIVE 0 OTHER SYSTEM

WAIT FOR DRIVE READY
TYPE <RETURN> TO CONTINUE . . .<CR>

TEST COMPLETE - ALL DRIVES COMPATIBLE!
SR=0000

STEP WAS STEP #3, SINCE STEPS 1 AND 2 WERE PERFORMED ON PDP-8
SYSTEM #1 BEFORE BRINGING THE PACK TO DRIVE 0 ON SYSTEM #2.
THIS ALLOWS THE OPERATOR TO SKIP THE DIALOGUE FOR STEPS 1 AND 2 ON
SYSTEM #2, SINCE THE TESTING FOR STEPS 1 AND 2 IS PERFORMED ON
SYSTEM #1. THE PROGRAM RUNNING ON SYSTEM #2 THEREFORE ASSUMES
THAT STEPS 1 AND 2 HAVE BEEN PERFORMED ON THE OTHER SYSTEM (#1)

5.1 FATAL ERRORS:

ERRORS WHICH OCCUR DURING THE RUNNING OF THIS PROGRAM, AND WHICH ARE CONSIDERED TO BE FATAL TO THE CONTINUATION OF THE PROGRAM INCLUDE THE FOLLOWING:

- A. NO DONE INTERRUPT - AFTER A COMMAND HAS BEEN ISSUED TO A DRIVE
- B. IOT ERRORS - AN IOT FAILS TO CLEAR THE AC OR SKIPS WHEN IT SHOULD NOT
- C. ERROR REGISTER - IF ERROR REGISTER BITS ARE SET AND THE COMPOSITE ERROR FLAG IS NOT SET
- D. DRIVE ERRORS - DRIVE ERROR BITS SET IN STATUS WORD #2 BUT COMPOSITE ERROR FLAG NOT SET
- E. REGISTER LOADING - LOADING A CONTROLLER REGISTER AND READING IT BACK SHOWED SOME DIFFERENCE
- F. SECTOR REGISTER - THE FINAL SECTOR REGISTER CONTENTS WERE INCORRECT AFTER A DATA TRANSFER, BUT THE COMPOSITE ERROR FLAG WAS NOT SET.

THE FOLLOWING ERROR REPORT WILL BE PRINTED:

"FATAL ERROR"
 "PC: XXXX AC: XXXX"
 "SR=XXXX _"

EXPLANATION: PC: = PROGRAM COUNTER AT THE TIME OF THE ERROR (REFER TO PROGRAM LISTING)
 AC: = ACCUMULATOR CONTENTS AT THE TIME OF THE ERROR
 SR= ALLOWS OPERATOR TO MODIFY PSEUDO SWITCH REGISTER AND RESTART OR CONTINUE THE PROGRAM (SEE SECTION 4.5)

5.2 SOFT ERRORS:

ANY NON-FATAL ERROR WHICH IS RECOVERABLE WITHIN THE RETRY LIMIT, WILL BE LOGGED AS A SOFT ERROR.

5.3 HARD ERRORS:

THE FOLLOWING TYPES OF ERRORS WILL BE LOGGED AS HARD ERRORS:

- A. ANY ERROR WHICH CANNOT BE RECOVERED BY RETRY WITHIN THE RETRY LIMIT
- B. WRITE PROTECT OR HEAD CURRENT ERRORS
- C. DRIVE ERRORS WHICH CANNOT BE RESET

THE PROGRAM MAINTAINS AN EXTENSIVE ERROR LOG FOR EACH DRIVE. THE FOLLOWING ERRORS ARE LOGGED IN THE DRIVE STATE TABLES:

HARD ERRORS
SOFT ERRORS
DRIVE ERRORS
SEEK ERRORS
DATA ERRORS
TRACKING ERRORS
DATA CRC ERRORS
HEADER CRC ERRORS
DATA LATE ERRORS
OPI ERRORS
HEADER NOT FOUND ERRORS
CONTROLLER ERRORS

(SEE SECTION 5.6 FOR ERROR REPORTING DESCRIPTION)

5.5 RETRIES: -----

A. SEEK RETRIES CONSIST OF THE FOLLOWING OPERATIONS IN THE

- 1) RESET DRIVE
- 2) CHECK DRIVE READY
- 3) GET STATUS AND CHECK
- 4) READ HEADER
- 5) SEEK TO TRACK, WAIT FOR DRIVE READY
- 6) GET STATUS AND CHECK
- 7) READ HEADER AND VERIFY POSITION

THERE WILL BE NO READ OR WRITE RETRIES. DRIVE INCOMPATIBILITY WILL BE ASSUMED IF THERE ARE ANY READ OR WRITE ERRORS.

5.6 ERROR REPORTING: -----

A. DATA ERRORS - THE FOLLOWING IS AN EXAMPLE OF A DATA ERROR REPORT:

```
"DATA ERROR"
"DN: 000X"
"BA: , GOOD    BAD"

"FXXXX XXXX   XXXX"
  ETC.
```

EXPLANATION:	DN:	IS THE DRIVE NUMBER (X=0-3)
	BA:	IS THE DATA BUFFER ADDRESS
	F	IS THE FIELD OF THE DATA BUFFER
	GOOD	IS THE EXPECTED DATA WORD
	BAD	IS THE ACTUAL DATA WORD READ

NOTE: SEE SECTION 2.4 FOR BIT DEFINITIONS OF THE RL8A CONTROLLER REGISTERS AND DRIVE STATUS WORDS.

"ERROR PC: XXXX"
 "DN:000X"
 "ER: X00X CA: XXXX CB: XXXX SA: XXXX INC: XXXX FWC: XXXX"
 "CF: 000X"
 "RDS: XXXX WRTS: XXXX SKS: XXXX HE: XXXX SE: XXXX DE: XXXX"
 "SKE: XXXX DAT: XXXX TE: XXXX DCRC: XXXX HCRC: XXXX DLT: XXXX"
 "OPI: XXXX HNF: XXXX CE: XXXX"
 "S1: XXXX S2: XXXX"

EXPLANATION:

PC: THE PROGRAM COUNTER (FIELD 0) AT THE TIME OF THE ERROR
 DN: THE DRIVE NUMBER (X = 0-3)
 ER: THE ERROR REGISTER READ AT THE TIME OF THE INTERRUPT
 CA: THE COMMAND A REGISTER READ AT THE TIME OF ERROR
 CB: THE COMMAND B REGISTER READ AT THE TIME OF ERROR
 SA: THE SECTOR ADDRESS REGISTER READ AT THE TIME OF ERROR
 INC: THE INITIAL WORD COUNT LOADED INTO CONTROLLER
 FWC: THE FINAL WORD COUNT READ AT TIME OF INTERRUPT
 CF: THE CONTROLLER FLAGS AT TIME OF INTERRUPT
 0001 = COMPOSITE ERROR FLAG SET
 0002 = DONE FLAG SET
 0003 = BOTH DONE AND COMPOSITE ERROR FLAGS SET
 0000 = NO FLAGS SET

RDS: UP TO 4096 (DECIMAL) READ OPERATIONS
 WRTS: UP TO 4096 (DECIMAL) WRITE OPERATIONS
 SEEKS: UP TO 1000 (DECIMAL) SEEK OPERATIONS (1 PASS)
 HE: HARD ERROR TALLY (DECIMAL)
 SE: SOFT ERROR TALLY (DECIMAL)
 DE: DRIVE ERROR TALLY (DECIMAL)
 SKE: SEEK ERROR TALLY (DECIMAL)
 DAT: DATA ERROR TALLY (DECIMAL)
 TE: TRACKING ERROR TALLY (DECIMAL) (HEADS DRIFT OFF TRACK)
 DCRC: DATA CRC ERROR TALLY (DECIMAL)
 HCRC: HEADER CRC ERROR TALLY (DECIMAL)
 DLT: DATA LATE ERROR TALLY (DECIMAL)
 OPI: OPERATION INCOMPLETE ERROR TALLY (DECIMAL)
 HNF: HEADER NOT FOUND ERROR TALLY (DECIMAL)
 CE: CONTROLLER ERROR TALLY (DECIMAL)
 S1: DISK STATUS WORD #1 (OCTAL)
 S2: DISK STATUS WORD #2 (OCTAL)

IF ALL 5 FACTORY AND/OR ALL 5 FIELD BAD SECTOR FILES CANNOT BE READ WITHOUT ERRORS, AN ERROR MESSAGE WILL RESULT AS FOLLOWS:

"CAN'T READ FACTOR BAD SECTOR FILE DRIVE X"

OR

"CAN'T READ FIELD BAD SECTOR FILE DRIVE X"

WHERE "X" IS THE DRIVE NUMBER UPON WHICH THE BAD SECTOR FILES CANNOT BE READ.

5.8 OVER-WRITE TEST ERROR: -----

IF A DATA ERROR OCCURS DURING AN OVER-WRITE TEST STEP, THE PROGRAM WILL ASSUME THAT A DRIVE INCOMPATIBILITY EXISTS. NO READ OR WRITE RETRIES WILL BE ATTEMPTED.

THE FOLLOWING IS AN EXAMPLE OF THE DATA ERROR REPORT:

"DATA ERROR"
"DN: 000X"
"BA: GOOD BAD"

"1XXX XXXX XXXX"
"1XXX XXXX XXXX"
ETC.

EXPLANATION: DN: IS THE DRIVE NUMBER (X=0-7)
BA: IS THE DATA BUFFER ADDRESS IN FIELD 1
GOOD: IS THE EXPECTED DATA WORD
BAD: IS THE ACTUAL DATA WORD READ

THE ABOVE PORTION OF THE ERROR MESSAGE MAY BE INHIBITED BY SETTING SWITCH REGISTER BIT 5=1.

THE FOLLOWING IS AN EXAMPLE OF THE OVER-WRITE ERROR REPORT:

"OVER-WRITE TEST ERROR"
"DRIVE X OVER DRIVE Y" "OTHER SYSTEM"
"SECTOR: 00XX"
"TRACK: 0XXX"
"SURFACE: X"
"SEEK IN" OR "SEEK OUT"
"ERROR AT PC: XXXX"

"SR=0000 _"

EXPLANATION:	DRIVE X	THE DRIVE ON WHICH THE PACK IS MOUNTED
	DRIVE Y	THE DRIVE WHOSE DATA WAS BEING OVER-WRITTEN
	OTHER SYSTEM	ONLY PRINTED IF DRIVE Y WAS ON A DIFFERENT PDP-8 SYSTEM
	SECTOR:	THE SECTOR NUMBER (OCTAL) WHICH WAS BEING WRITTEN (XX=0 THRU 47)
	TRACK:	THE TRACK ON WHICH THE SECTOR IS LOCATED (XXX=0 THRU 377 OCTAL)

SEEK IN THE SECTOR AND TRACK ARE LOCATED
 THE DIRECTION OF THE LAST SEEK TO
 TRACK PRIOR TO WRITING THE SECTOR
 SEEK OUT SEEK IN FROM TRACK 0
 THE DIRECTION OF THE LAST SEEK TO
 TRACK PRIOR TO WRITING THE SECTOR
 SEEK OUT FROM TRACK 377
 ERROR AT PC: THE ERROR CALL ADDRESS IN FIELD 0

THE SR=0000 MESSAGE IS AN EFFECTIVE PROGRAM HALT. THE OPERATOR MAY
 MODIFY THE PSEUDO SWITCH REGISTER. TYPING CARRIAGE RETURN WILL START
 THE TESTS OVER FROM THE BEGINNING WITHOUT REASKING THE INITIAL DIALOGUE
 DESCRIBED IN SECTION 4.2.2. TYPING A LINEFEED WILL START THE PROGRAM
 OVER FROM THE INITIAL DIALOGUE.

5.9 ADJACENT CYLINDER TEST ERROR:

IF A DATA ERROR OCCURS DURING AN ADJACENT WRITE TEST STEP THE PROGRAM
 WILL ASSUME THAT A DRIVE INCOMPATIBILITY EXISTS. NO READ OR WRITE
 RETRIES WILL BE ATTEMPTED.

THE FOLLOWING IS AN EXAMPLE OF THE DATA ERROR REPORT:

"DATA ERROR"
 "DN: 000X"
 "BA: GOOD BAD"

"1XXXX XXXX XXXX"
 "1XXXX XXXX XXXX"
 ETC.

EXPLANATION: DN: IS THE DRIVE NUMBER (X=7 THRU 7)
 BA: IS THE DATA BUFFER ADDRESS IN FIELD 1
 GOOD: IS THE EXPECTED DATA WORD
 BAD: IS THE ACTUAL DATA WORD READ

THE ABOVE PORTION OF THE ERROR MESSAGE MAY BE INHIBITED BY SETTING
 SWITCH REGISTER BIT 5=1.

THE FOLLOWING IS AN EXAMPLE OF THE ADJACENT ERROR REPORT:

"ADJACENT CYLINDER TEST ERROR"
 "DRIVE X ADJACENT TO DRIVE Y" "OTHER SYSTEM"
 "WRITTEN SECTOR: 00XX"
 "ADJACENT SECTOR: 00XX"
 "WRITTEN TRACK: 0XXX"
 "ADJACENT TRACK: 0XXX"
 "SURFACE: X"
 "SEEK IN" OR "SEEK OUT"
 "ERROR AT PC: XXXX"

"SR=0000 _"

OTHER SYSTEM THE DRIVE X DATA
 ONLY PRINTED IF DRIVE Y WAS ON A
 DIFFERENT PDP-8 SYSTEM THAN DRIVE X
 WRITTEN SECTOR: THE SECTOR JUST WRITTEN BY DRIVE X
 ADJACENT SECTOR: THE SECTOR, WRITTEN BY DRIVE Y, WHICH
 IS ADJACENT TO THE WRITTEN SECTOR.
 THIS IS THE SECTOR WHOSE DATA WAS
 REPORTED AS BEING INCORRECT
 WRITTEN TRACK: THE TRACK UPON WHICH THE LAST WRITE
 WAS PERFORMED BY DRIVE X.
 ("WRITTEN SECTOR" IS ON THIS TRACK)
 ADJACENT TRACK: THE TRACK UPON WHICH THE DATA ERROR
 OCCURRED ("ADJACENT SECTOR" IS ON
 THIS TRACK)
 SURFACE: THE DISK SURFACE (0 OR 1) ON WHICH
 THE TRACK AND SECTORS ARE LOCATED
 SEEK IN THE DIRECTION OF THE LAST SEEK PRIOR
 TO WRITING THE WRITTEN SECTOR
 SEEK IN FROM TRACK 0
 SEEK OUT THE DIRECTION OF THE LAST SEEK PRIOR
 TO WRITING THE WRITTEN SECTOR
 SEEK OUT FROM TRACK 377
 ERROR AT PC: THE ERROR CALL ADDRESS IN FIELD 0

THE SR=0000 MESSAGE IS AN EFFECTIVE PROGRAM HALT. THE OPERATOR MAY
 MODIFY THE PSEUDO SWITCH REGISTER. TYPING CARRIAGE RETURN WILL
 START THE TESTS OVER WITHOUT REASKING THE INITIAL DIALOGUE, DESCRIBED
 IN SECTION 4.2.2. TYPING A LINEFEED WILL START THE PROGRAM OVER FROM
 THE INITIAL DIALOGUE. THE ADJACENT CYLINDER ERROR MESSAGE WILL ONLY
 BE REPORTED IF THE "ADJACENT SECTOR" DATA WAS AFFECTED BY WRITING
 THE "WRITTEN SECTOR".

6.0 SWITCH REGISTER SETTINGS:

6.1 NORMAL OPERATION SWITCHES:

ALL SWITCHES = 0

6.2 ERROR RELATED SWITCHES:

SR5 = 1 INHIBIT DATA ERROR PRINTOUTS

7.0 REVISIONS:

8.0 ADDITIONAL INFORMATION:

THE ONLY CPU HALTS WILL OCCUR ONLY IF SOMETHING CATASTROPHIC HAPPENS. THE FOLLOWING CAUSES MAY EXIST:

A. INTERRUPT SERVICE ROUTINE ENTERED BUT NO INTERRUPT REQUEST ACTIVE ON THE BUS (SEE LISTING AT ADDRESS "NOTINT"). THE ERROR MESSAGE:

"NO INTERRUPT REQUEST"

B. AN INTERRUPT OCCURRED FROM A DEVICE UNKNOWN TO THIS PROGRAM (SEE LISTING AT ADDRESS "UNKINT"). THE ERROR MESSAGE:

"UNKNOWN INTERRUPT"

C. A LOCATION WHICH CONTAINED A "HLT" INSTRUCTION INITIALLY, AND WHICH WAS TO BE MODIFIED BY THE PROGRAM FOR USE AS A CDF, CIF OR PARAMETER, WAS NOT MODIFIED. EITHER THE PROGRAM IS NOT IN MEMORY CORRECTLY OR THE CPU HAS A HARDWARE FAULT.

8.2 DRIVE ERRORS:

CERTAIN ERRORS ARE CONSIDERED FATAL TO A DRIVE. THESE ARE:

A. THE DRIVE IS WRITE PROTECTED

B. THE DRIVE HAS A CURRENT HEAD ERROR

C. NO DRIVE SYSTEM CLOCK (OPI FROM GET STATUS)

D. DRIVE IS NOT SEEKING OR TRACKING (UNLOADED STATE)

IN THE EVENT THAT ANY OF THESE CONDITIONS EXIST, AN ERROR MESSAGE WILL RESULT, AND THE PROGRAM WILL BE RESTARTED AT THE INITIAL TEST (STEP 1) OR THE SR= QUESTION WILL BE ASKED. THE ERROR CONDITION SHOULD BE CORRECTED, AND THE PROGRAM RESTARTED.

8.4 POWER FAIL:

IN THE EVENT OF A SYSTEM POWER FAIL INTERRUPT, THE PROGRAM WILL TAKE THE FOLLOWING ACTIONS:

A. SAVE INTERRUPT PC

B. DEPOSIT PROGRAM RESTART INSTRUCTION IN LOC 0 FLD 0

C. HALT

WHEN THE SYSTEM POWER IS RESTORED, THE PROGRAM WILL TAKE THE FOLLOWING ACTIONS, PROVIDING THE AUTO-RESTART FUNCTION HAS BEEN ENABLED:

"POWER FAILED AT PC: XXXX"

THE PROGRAM WILL BE COMPLETELY RESTARTED AUTOMATICALLY.

9.0 PROGRAM LISTING:

ATTACHED

/AJRLJ-B RL8A/RL02 DRIVE COMPATIBILITY
/COPYRIGHT (C) 1981 DIGITAL EQUIPMENT CORP. MAYNARD,MASS. 01754

/ REVISION HISTORY

/ -----

/ ORIGINAL RELEASE - DAVE ORIN
/ MODIFICATIONS BY DATE

/ -----

/ H. POULTER FEBRUARY 1979

/ HP 001

/ MODIFICATIONS TO ALLOW THE RL02 DRIVE TO RUN.
/ ADD CODE TO DISPLAY THE HEAD AND TRACK IN THE MQ.
/ ADD CODE TO CORRECTLY READ THE BAD SECTOR FILE.
/ REMOVE MASK FOR BIT 200 WHICH IS NOW USED.
/ ADD CODE TO SET MESSAGE ACTIVE FLAG INMODE

/*****

/*****

/***** ASSEMBLY DEFINITIONS *****/

/*****

/*****

/*****

/

7002	BSW=7002	/SWAP HALVES
7421	MOQ=7421	/LOAD MQ FROM AC THEN CLEAR AC
7501	MOA=7501	/INCLUSIVE OR THE MQ WITH THE AC
7621	CAM=7621	/CLEAR AC AND MQ
7521	SWP=7521	/SWAP AC AND MQ
7701	ACL=7701	/LOAD MQ INTO AC
6000	SKON=6000	/SKIP IF INTERRUPT ON AND TURN OFF
6003	SRQ=6003	/SKIP ON INTERRUPT REQUEST
6004	GTF=6004	/GET THE FLAGS
6005	RTF=6005	/RESTORE THE FLAGS
6007	CAF=6007	/CLEAR ALL FLAGS
6035	KIE=6035	/KEYBOARD INTERRUPT ENABLE, AC11=1=ENABLE 0=DISABLE
6045	TIE=6045	
6030	KCF=6030	
6032	KCC=6032	
6031	KSF=6031	
6201	CDF=6201	/CHANGE DATA FIELD
6202	CIF=6202	/CHANGE INSTRUCTION FIELD
6203	CDI=6203	/CHANGE DATA AND INSTRUCTION FIELDS
6214	RDF=6214	/READ DATA FIELD
6224	RIF=6224	/READ INSTRUCTION FIELD
6244	RMF=6244	/RESTORE MEMORY FIELDS
6102	SPL=6102	/SKIP IF AC LOW FLAG SET - POWER FAIL
6103	CAL=6103	/CLEAR AC LOW INTERRUPT
6101	SBE=6101	/SKIP IF BATTERY EMPTY FLAG SET

/

/*****

/*****

/***** I O T DEFINITIONS *****/

/*****

/*****

```

6600      RLDC=6600      /CLEAR DEVICE, CLEAR AC
6601      RLSD=6601      /SKIP ON FUNCTION DONE FLAG, CLEAR FLAG IF SET
6602      RLMA=6602      /LOAD BREAK MEMORY ADDRESS FROM AC0:11, CLEAR AC
6603      RLCA=6603      /LOAD COMMAND REGISTER A FROM AC0:11, CLEAR AC
6604      RLCB=6604      /LOAD COMMAND REGISTER B FROM AC0:11, CLEAR AC
6605      RLSA=6605      /LOAD SECTOR ADDRESS REGISTER FROM AC0:5, CLEAR AC
6606      SPARE=6606      /SPARE IOT, WILL CLEAR AC
6607      RLWC=6607      /LOAD WORD COUNT FROM AC0:11, USE TWO'S COMPLEMENT, CLEAR AC
/
6610      RRER=6610      /READ ERROR REGISTER INTO AC0:2, AC11=DRIVE READY, AC10=DRIVE ERROR
6611      RRLC=6611      /READ WORD COUNT INTO AC0:11
6612      RRCA=6612      /READ COMMAND REGISTER A INTO AC0:11
6613      RRCB=6613      /READ COMMAND REGISTER B INTO AC0:11
6614      RRSA=6614      /READ SECTOR ADDRESS INTO AC0:5
6615      RRSI=6615      /READ 8 BIT SILO WORD INTO AC4:11
6616      NOP61=6616      /UNUSABLE IOT (FORMERLY RLSE SKIP ON DRIVE READY)
6617      RLSE=6617      /SKIP ON COMPOSITE ERROR FLAG, CLEAR FLAG IF SET

```

```

/ BIT      DEFINITION
/
/ 0=1      MAINTENANCE INHIBIT (NOT USED IN THIS PROGRAM)
/ 1=1      MAINTENANCE LOOP DAR TO SILO (NOT USED IN THIS PROGRAM)
/ 2=0      12 BIT DATA MODE
/ 2=1      8 BIT DATA MODE
/ 3=1      INTERRUPT ENABLE
/ 4:5      DRIVE SELECT
/ 6:8      DATA BUFFER FIELD (EMA)
/

```

```

/ BITS 9 10 11      COMMAND
/
/ 0 0 0      (0) MAINTENANCE
/ 0 0 1      (1) RESET
/ 0 1 0      (2) GET STATUS
/ 0 1 1      (3) SEEK
/ 1 0 0      (4) READ HEADER
/ 1 0 1      (5) WRITE DATA
/ 1 1 0      (6) READ DATA
/ 1 1 1      (7) READ DATA NO HEADER CHECK
/

```

* ERROR REGISTER *

```

/ BIT      DEFINITION
/
/ 0=1      CRC ERROR (CYCLIC REDUNDANCY CHECK)
/ 1=1      OPI ERROR (OPERATION INCOMPLETE)
/ 2=1      DATA LATE/HEADER NOT FOUND
/ 3 THRU 9 NOT USED
/ 10=1     DRIVE ERROR
/ 11=1     DRIVE READY
/

```

```

/ BITS 0 1 2      ERROR
/
/ 0 0 0      (0)
/ 0 0 1      (1) DLT DATA LATE
/ 0 1 0      (2) OPI OPERATION INCOMPLETE
/ 0 1 1      (3) HNF HEADER NOT FOUND
/ 1 0 0      (4) DCRC DATA CRC
/ 1 0 1      (5)
/ 1 1 0      (6) HCRC HEADER CRC
/ 1 1 1      (7)
/

```

* DISK STATUS WORD #1 *

```

/ BIT      DEFINITION
/

```

```

/      4=0    RL 01 IDENTIFICATION SETTING
/      4=1    RL 02 IDENTIFICATION SETTING
/      5=1    HEAD SELECT
/      6=1    COVER OPEN
/      7=0    HEADS HOME
/      7=1    HEADS OVER DISK
/      8=0    BRUSH OVER DISK
/      8=1    BRUSH HOME
/

```

```

/ BITS 9 10 11      DRIVE STATE
/
/      0 0 0      (0) LOAD STATE
/      0 0 1      (1) SPIN-UP
/      0 1 0      (2) LOAD HEADS
/      0 1 1      (3) BRUSH CYCLE
/      1 0 0      (4) SEEK-TRACK COUNTING
/      1 0 1      (5) SEEK LINEAR MODE
/      1 1 0      (6) UNLOAD HEADS
/      1 1 1      (7) SPIN-DOWN
/

```

```

/*****
/

```

```

/      * DISK STATUS WORD #2 *
/

```

```

/      BIT      DEFINITION
/
/      4=1      WRITE DATA ERROR
/      5=1      HEAD CURRENT ERROR
/      6=1      WRITE-LOCKED
/      7=1      SEEK TIME-OUT
/      8=1      SPIN-UP TIME-OUT
/      9=1      WRITE GATE ERROR
/      10=1     VOLUME CHECK
/      11=1     DRIVE SELECT ERROR (TWO OR MORE HAVE SAME DRIVE NUMBER)
/

```

```

/*****
/

```

```

/      * DRIVE STATE TABLE OFFSET DEFINITIONS *
/

```

```

0000  OLDCYL=0      /PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
0001  CURCYL=1     /CURRENT CYLINDER ADDRESS BY READ HEADER
0002  NEWCYL=2     /NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
0003  HRDERR=3     /HARD ERROR COUNT
0004  SFTERR=4     /SOFT ERROR COUNT
0005  DRVERR=5     /DRIVE ERROR COUNT
0006  SEKERR=6     /SEEK ERROR COUNT
0007  DATERR=7     /DATA ERROR COUNT
0010  TRKERR=10    /TRACKING ERROR COUNT

```

```

0011 DCR CER=11 /DATA CRC ERROR COUNT
0012 HCR CER=12 /HEADER CRC ERROR COUNT
0013 DL TERR=13 /DATA LATE ERROR COUNT
0014 OPI ERR=14 /OPERATION INCOMPLETE ERROR COUNT
0015 HNF ERR=15 /HEADER NOT FOUND ERROR COUNT
0016 CTL ERR=16 /CONTROLLER ERROR COUNT
0017 WRDCNT=17 /INITIAL WORD COUNT SENT
0020 INITCA=20 /INITIAL CURRENT DATA BUFFER ADDRESS
0021 SECADD=21 /SECTOR ADDRESS SENT
0022 XCOMA=22 /COMMAND REGISTER A SENT
0023 XCOMB=23 /COMMAND REGISTER B SENT
0024 XENDSC=24 /EXPECTED FINAL SECTOR ADDR REG. CONTENTS
/
/
/
0010 OFFSET=0010 /DEFINE FIELD OFFSET OF CONSOLE PACKAGE
0200 RSTART=200 /DEFINE PROGRAM RESTART ADDRESS
/
/DEFINE DATA BUFFERS
/
5600 BUF1=5600 /DEFINE BUFFER 1 FIELD 1 ADDRESS
6600 RBUF=6600 /DEFINE READ DATA BUFFER FIELD 1 ADDRESS
/
2000 BUFSZ1=2000 /DEFINE BUFFER 1 FLD 1 SIZE (1024)
1000 BUFSZR=1000 /DEFINE READ DATA BUFFER SIZE ( 512 )
/
/*****
/
/
/ * START OF PAGE ZERO CODE *
/
0000 *0
/
0000 0302 LOC0, "B /REVISION B
0001 5402 ADDR S1, JMP I ADDR S2 /GO TO INTERRUPT SERVICE ROUTINE
0002 2000 ADDR S2, INT SVC /ADDRESS OF INTERRUPT SERVICE ROUTINE
0003 2023 ADDR S3, RESTR T /POWER FAIL RESTART ADDRESS
0004 0000 ADDR S4, 0 /RESERVED FOR ODT
0005 0000 ADDR S5, 0 /RESERVED FOR ODT
0006 0000 ADDR S6, 0 /RESERVED FOR ODT
0007 0000 ADDR S7, 0 /RESERVED FOR ODT
/
0010 *10
/
0010 0000 AUTO10, 0 /AUTO-INCREMENT REGISTERS
0011 0000 AUTO11, 0
0012 0000 AUTO12, 0
0013 0000 AUTO13, 0
0014 0000 AUTO14, 0
0015 0000 AUTO15, 0
0016 0000 AUTO16, 0
0017 0000 AUTO17, 0
/
0020 *20

```

```

0020 0000 PSR, 0 /PSEUDO (SOFTWARE) SWITCH REGISTER
0021 0000 HCW1, 0 /HARDWARE CONFIGURATION WORD #1
0022 0400 HCW2, 0400 /HARDWARE CONFIGURATION WORD #2
0023 0000 HCW3, 0 /HARDWARE CONFIGURATION WORD #3
/
0024 *24
/
0024 0000 CURDRV, 0 /CURRENT DRIVE SELECTED
0025 0000 DRV60, 0 /DRIVE SELECTED LAST USING CONTROLLER
0026 0000 STAT6A, 0 /STATUS WORD #1
0027 0000 STAT6B, 0 /STATUS WORD #2
0030 0000 ERREG, 0 /ERROR REGISTER
0031 0000 COMDA, 0 /COMMAND REG. A READ
0032 0000 COMDB, 0 /COMMAND REG. B READ
0033 0000 ENDSC, 0 /SECTOR ADDRESS READ
0034 0000 ENDWD, 0 /FINAL WORD COUNT READ
0035 0000 CURUNT, 0 /CURRENT UNIT UNDER TEST
/
0036 0000 SECCNT, 0 /SECTOR COUNTER FOR DATA BUFFER
0037 0000 WDCNTR, 0 /WORD COUNTER FOR DATA BUFFER
0040 0003 K3, 0003
0041 0005 K5, 0005
0042 0007 K7, 0007
0043 0100 K100, 0100
0044 0260 K260, 0260
0045 0377 K377, 0377
0046 0400 K400, 0400
0047 0777 K0777, 0777
0050 1000 K1000, 1000
0051 2000 K2000, 2000
0052 7777 M1, 7777
0053 7774 M4, -4
0054 7400 M400, -400
/
/*****
/
/THESE LOCATIONS ARE REINITIALIZED TO DEFAULT VALUES WHEN
/THE PROGRAM IS RESTARTED. DO NOT INSERT, DELETE
/OR REARRANGE THESE VALUES.
/
0055 TBLSTR=.
/
/
0055 0000 CBUSY, 0 /CONTROLLER BUSY FLAG (DSC 60,61) 0=NOT BUSY 7777=BUSY
0056 7777 UNIT0, 7777 /UNIT 0 IN USE FLAG, 7777=NOT IN USE, 0,1,2, OR 3 = DRIVE #
0057 7777 UNIT1, 7777 /UNIT 1 IN USE FLAG
0060 7777 UNIT2, 7777 /UNIT 2 IN USE FLAG
0061 7777 UNIT3, 7777 /UNIT 3 IN USE FLAG
0062 0000 ONSYS0, 0 /UNIT 0 ON THIS SYSTEM FLAG 0=YES 7777=NO
0063 0000 ONSYS1, 0 /UNIT 1 ON THIS SYSTEM FLAG
0064 0000 ONSYS2, 0 /UNIT 2 ON THIS SYSTEM FLAG
0065 0000 ONSYS3, 0 /UNIT 3 ON THIS SYSTEM FLAG

```

HP 001

0066	0600	DSCU0, 0600	/UNIT 0 DEVICE CODE INDICATOR 0600=60,61 0620=62,63
0067	0600	DSCU1, 0600	/UNIT 1 DEVICE CODE INDICATOR
0070	0600	DSCU2, 0600	/UNIT 2 DEVICE CODE INDICATOR
0071	0600	DSCU3, 0600	/UNIT 3 DEVICE CODE INDICATOR
0072	0003	MAXSTP, 3	/MAXIMUM STARTING PROGRAM STEP IF 2 SYSTEMS
0073	0001	STRSTP, 1	/STARTING STEP IF 2 SYSTEMS
0074	0000	SYS2F, 0	/0=1 SYSTEM, 7777=2 SYSTEMS
0075	0000	NUMUNT, 0	/NUMBER OF UNITS UNDER TEST
0076	0001	SEKRET, 0001	/SEEK TO TRACK RETRY LIMIT
		/	
		/	
0077		TBLED=.	
		/	
		/	
		/	
		* END OF CRITICAL LOCATIONS *	
		/	

		/	
0077	0000	SERNM1, 0	/PACK SERIAL NUMBER 1ST WORD
0100	0000	SERNM2, 0	/PACK SERIAL NUMBER 2ND WORD
0101	0000	SERNM3, 0	/PACK SERIAL NUMBER 3RD WORD
0102	0000	SERNM4, 0	/PACK SERIAL NUMBER 4TH WORD
0103	0000	STPNUM, 0	/STEP NUMBER
0104	0000	TBLPNT, 0	/DRIVE PARAMETER TABLE POINTER
0105	0000	FUNCOD, 0	/FUNCTION CODE
0106	5600	BUFAD1, BUF1	/FIELD 1 BUFFER ADDRESS
0107	0000	CNTR1, 0	
0110	6001	KION, 10N	
0111	0000	TEMP1, 0	/TEMPORARY STORAGE FOR USE IN MAIN CODE
0112	0000	FLGSAV, 0	/CONTROLLER FLAGS AFTER INTERRUPT, BIT 11 = COMPOSITE ERROR /BIT 10 = DONE
0113	0000	PATNUM, 0	/DATA PATTERN #
0114	0000	PNTR1, 0	
0115	0000	RETCNT, 0	/RETRY COUNT
0116	0000	BADPNT, 0	/BAD SECTOR POINTER
0117	0000	DERFLG, 0	/DATA ERROR FLG
0120	0000	PATMP, 0	
0121	0000	SECSAV, 0	/SECTOR ADDRESS SAVE AREA
0122	0000	SECSV, 0	
0123	0000	SECPNT, 0	/SECTOR POINTER
0124	0000	CYLPNT, 0	/CYLINDER POINTER
0125	0000	TRKCNT, 0	
0126	0000	STPNT, 0	
0127	0000	WSEC, 0	
0130	0000	ASEC, 0	
0131	0000	WRTPNT, 0	
0132	0000	ADJPNT, 0	
0133	0000	SEKDIR, 0	/0=IN, 7777=OUT
0134	0000	OVRUNT, 0	/UNIT BEING OVER-WRITTEN
		/	
		/	
		/	


```

/
/
/*****
/*****
/***** SUBROUTINE CALLS *****/
/*****
/*****
/*****
/

```

0135	4535 7076	C8CALL=JMS I . XC8CAL	/CALL CONSOLE PACKAGE /AC=0 MEANS PRINT SR=XXXX /AC NOT = 0 MEANS DECODE ASCII CHAR IN AC
0136	4536 2277	SETPNT=JMS I . XSETP	/ROUTINE TO SET UP "TBLPNT" IN DRIVE STATE TABLES /CALL+1 = TABLE ENTRY OFFSET
0137	4537 1047	GETSWR=JMS I . SWRGET	/ROUTINE TO GET HARDWARE OR SOFTWARE SWR /RETURN CALL+1 WITH VALUE OF SWR IN AC
0140	4540 2341	STAPRT=JMS I . PRTSTA	/ROUTINE TO PRINT STATUS MESSAGES /SEE LISTING OF SUBROUTINE FOR FURTHER DETAILS
0141	4541 3256	ERROR=JMS I . XERROR	/ROUTINE TO HANDLE GENERAL STATUS ERRORS /RETURN CALL+1 IF ERROR NOT RECOVERABLE /ELSE RETURN CALL+2 IF RECOVERABLE
0142	4542 2600	ERRCHK=JMS I . CHKERR	/CHECK FOR ERRORS STATUS WORD 1 OR COMPOSITE ERROR FLG /RETURN CALL+2 IF ERROR, CALL+1 IF OK
0143	4543 3000	GO=JMS I . XGO	/CALL TO ROUTINE TO EXECUTE A COMMAND
0144	4544 2256	WAITDN=JMS I . DNWAIT	/CALL TO ROUTINE TO WAIT FOR DONE INTERRUPT
0145	4545 3120	XFER=JMS I . EXFER	/CALL TO ROUTINE TO EXECUTE A READ OR WRITE
0146	4546 2400	SETPOS=JMS I . POSSET	/CALL TO ROUTINE TO SEEK TO ABSOLUTE CYLINDER AND SURFACE /CALL+1 IS HEAD IN BIT 1,CYL IN BITS 4:11
0147	4547 3056	GETSTA=JMS I . STAGET	/CALL TO ROUTINE TO GET DRIVE STATUS /RETURN ONLY IF SUCCESSFUL
0150	4550 2200	STACHK=JMS I . CHKSTA	/CALL TO ROUTINE TO CHECK STATUS WORD #1
0151	4551 2124	MESAG=JMS I . MESGO	/CALL TO MESSAGE PRINT ROUTINE FLD 1

0152	4552 1077	YESRN=JMS I . YNGO	/CALL TO ROUTINE TO GET YES OR NO ANSWER
0153	4553 1106	PRNT=JMS I . PRNTGO	/CALL TO ROUTINE TO PRINT ONE CHARACTER
0154	4554 7162	GETOCT=JMS I . OCTGET	/CALL TO ROUTINE TO GET ONE OCTAL DIGIT
0155	4555 1062	PRNAC=JMS I . ACPRN	/CALL TO ROUTINE TO PRINT CONTENTS OF AC
0156	4556 7200	DOCRLF=JMS I . CRLFDO	/CALL TO ROUTINE TO EXECUTE A <CR> AND <LF>
0157	4557 2554	SPACE2=JMS I . SPACES	/CALL TO ROUTINE TO PRINT 2 SPACES
0160	4560 1266	RESET=JMS I . XRESET	/CALL TO ROUTINE TO RESET DRIVE
0161	4561 1056	DECPRN=JMS I . PRNDEC	/CALL TO ROUTINE TO PRINT FOUR DECIMAL DIGITS
0162	4562 4717	WAITCR=JMS I . CRWAIT	/CALL TO ROUTINE TO WAIT FOR CARRIAGE RETURN INPUT
0163	4563 4733	GETCHR=JMS I . CHRGET	/CALL TO ROUTINE TO GET TTY INPUT CHARACTER
0164	4564 3656	CNGIOT=JMS I . IOTCNG	/CALL TO ROUTINE TO CHANGE IOT DEVICE CODES
0165	4565 1400	GETBSF=JMS I . BSFGET	/CALL TO ROUTINE TO GET AND PROCESS BAD SECTOR FILE
0166	4566 5600	FNDTRK=JMS I . TRKFND	/CALL TO ROUTINE TO FIND 10 SETS OF 5 PERFECT ADJACENT TRACKS
0167	4567 3725	MNTPAK=JMS I . PAKMNT	/CALL TO ROUTINE TO WAIT FOR PACK MOUNTING
0170	4570 5200	OVRWRT=JMS I . WRTOVR	/CALL TO ROUTINE TO DO AN OVER-WRITE TEST
0171	4571 4121	ADJWRT=JMS I . WRTADJ	/CALL TO ROUTINE TO DO AN ADJACENT WRITE TEST
0172	4572 5000	OVRERR=JMS I . ERROVR	/CALL TO ROUTINE TO HANDLE AN OVER-WRITE ERROR
0173	4573 4600	ADJERR=JMS I . ERRADJ	/CALL TO ROUTINE TO HANDLE AN ADJACENT WRITE ERROR

0174 4574
4000

CMPSN=JMS I .
SNCMP

/CALL TO ROUTINE TO COMPARE PACK SERIAL NUMBER

```

/
/
/*****
/
/
/      * START OF MAIN PROGRAM CODE *
/
0200      *200
/
0200 6007  START,  CAF      /CLEAR ALL FLAGS
0201 6211      CDF      10  /CDF TO FLD 1
0202 1020      TAD      PSR  /GET SOFTWARE SWITCH REGISTER
0203 3777      DCA I    (PSR1 /MOVE TO FLD 1 PAGE 0
0204 1021      TAD      HCW1  /GET HARDWARE CONTROL WORD 1
0205 3776      DCA I    (HDW1  /MOVE TO FLD 1 PAGE 0
0206 1022      TAD      HCW2  /GET HARDWARE CONTROL WORD 2
0207 3775      DCA I    (HDW2  /MOVE TO FLD 1 PAGE 0
0210 6213      CDI      10    /CDI TO FLD 1
0211 5612      JMP I    .+1    /GO TO FLD 1
0212 0200      CONSET      /FLD 1 DESTINATION
/
/
0213 7301  MAIN,  CLA CLL IAC  /SET BIT 11 FOR KEYBOARD INTERRUPT ENABLE
0214 6035      KIE          /ENABLE KEYBOARD INTERRUPTS
0215 6045      TIE
0216 7300      CLA      CLL    /KIE DOES NOT CLEAR AC
/
0217 1074      TAD      SYS2F  /GET 2 SYSTEM FLAG
0220 7650      SNA      CLA    /SKIP IF 2 SYSTEMS
0221 5240      JMP      STEP1  /GO TO STEP 1, 1 SYSTEM
/
0222 1374      TAD      (STRTBL /GET START STEP ADDRESS TABLE ADDRESS
0223 1073      TAD      STRSTP  /INDEX USING STARTING STEP #
0224 3114      DCA      PNTR1   /SAVE TABLE POINTER
0225 1514      TAD I    PNTR1   /GET START ADDRESS FROM TABLE
0226 3114      DCA      PNTR1   /SAVE STARTING STEP ADDRESS
0227 5514      JMP I    PNTR1   /GO TO STARTING STEP
/
/
/*****
/
/
/      TABLE OF STARTING STEP ADDRESSES
/
0230 0000  STRTBL, 0      /NO STEP 0

```

0231	0240	STEP1	/STEP ADDRESSES
0232	0275	STEP2	
0233	0400	STEP3	
0234	0450	STEP4	
0235	0520	STEP5	
0236	0606	STEP6	
0237	0654	STEP7	

END OF TABLE

0240	7201	STEP1,	CLA	IAC	
0241	3103		DCA	STPNUM	/STEP # = 1
0242	3035		DCA	CURUNT	/CURRENT UNIT = 0
0243	4567		MNTPAK		/GO WAIT FOR PACK MOUNT
0244	1062		TAD	ONSYS0	/GET ON SYSTEM FLAG
0245	7640		SZA	CLA	/SKIP IF ON THIS SYSTEM
0246	5275		JMP	STEP2	/GO TO STEP 2
0247	1056		TAD	UNIT0	/GET DRIVE NUMBER
0250	0040		AND	K3	/MASK TO 2 BITS
0251	3024		DCA	CURDRV	/SAVE AS CURRENT DRIVE
0252	1066		TAD	DSCU0	/GET DEVICE CODE
0253	4564		CNGIOT		/CHANGE IOTS
0254	4560		RESET		/RESET DRIVE, SELECT DRIVE, RESET VOLUME CHECK
0255	4560		RESET		/RESET DRIVE, CLEAR COMPOSITE ERROR FLAG
0256	4773		JMS	RDYCHK	/GO CHECK FOR DRIVE READY
0257	5265		JMP	TST1	/DRIVE READY, GO DO FIRST TESTS
0260	4551		MESAG		
0261	2507		DRVMSG		/"DRIVE"
0262	4551		MESAG		
0263	2611		NOTRDY		/"NOT READY"
0264	5240		JMP	STEP1	/TRY AGAIN
0265	4565	TST1,	GETBSF		/GET BAD SECTOR FILE
0266	5213		JMP	MAIN	/TOO MANY BAD SECTORS
0267	4566		FNDTRK		/FIND 10 SETS OF 5 PERFECT ADJACENT TRACKS
0270	5213		JMP	MAIN	/TOO MANY BAD TRACKS
0271	4570	XTST1,	OVRWRT		/DO OVERWRITE TEST STEP 1
0272	5213		JMP	MAIN	/PROGRAM CONTINUED AFTER OVER-WRITE ERROR
0273	4571		ADJWRT		/DO ADJACENT WRITE TEST STEP 1
0274	5213		JMP	MAIN	/PROGRAM CONTINUED AFTER ADJACENT WRITE ERROR
0275	7305	STEP2,	CLA CLL	IAC RAL	/AC=2
0276	3103		DCA	STPNUM	/STEP # = 2
0277	7001		IAC		
0300	3035		DCA	CURUNT	/UNIT # = 1
0301	4567		MNTPAK		/GO WAIT FOR PACK MOUNTING
0302	1063		TAD	ONSYS1	/SEE IF UNIT 1 ON THIS SYS
0303	7640		SZA	CLA	/SKIP IF UNIT 1 ON THIS SYSTEM
0304	5772		JMP	STEP3	/UNIT 1 ON OTHER SYSTEM, GO TO STEP 3
0305	1057		TAD	UNIT1	/GET UNIT 1 DRIVE #
0306	0040		AND	K3	/MASK TO 3 BITS

0307	3024	DCA	CURDRV	/SAVE UNIT 1 DRIVE # AS CURRENT DRIVE
0310	1067	TAD	DSCU1	/GET UNIT 1 DEVICE CODE
0311	4564	CNGIOT		/CHANGE IOTS
0312	4560	RESET		/RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
0313	4560	RESET		/RESET DRIVE, CLEAR COMPOSITE ERROR FLG
0314	4773	JMS	RDYCHK	/CHECK FOR DRIVE READY
0315	5323	JMP	TST2	/DRIVE READY, GO TO TEST
0316	4551	MESAG		
0317	2507	DRVMSG		/"DRIVE"
0320	4551	MESAG		
0321	2611	NOTRDY		/"NOT READY"
0322	5275	JMP	STEP2	/TRY AGAIN
/				
0323	1073	TST2,	TAD	STRSTP
0324	7041		CIA	
0325	1103		TAD	STPNUM
0326	7640		SZA	CLA
0327	5334		JMP	XTST2
0330	4565		GETBSF	
0331	5213		JMP	MAIN
0332	4566		FNDTRK	
0333	5213		JMP	MAIN
/				
0334	4574	XTST2,	CMPSN	
0335	5275		JMP	STEP2
0336	4570		OVRWRT	
0337	5213		JMP	MAIN
0340	4571		ADJWRT	
0341	5213		JMP	MAIN
0342	5772		JMP	STEP3
/				
0372	0400			
0373	2333			
0374	0230			
0375	0022			
0376	0021			
0377	0020			
	0400			
PAGE				
/				
/				
/				
0400	7325	STEP3,	CLA CLL	CML IAC RAL
0401	3103		DCA	STPNUM
0402	7305		CLA CLL	IAC RAL
0403	3035		DCA	CURUNT
0404	1060		TAD	UNIT2
0405	7710		SPA	CLA
0406	5777		JMP	STEP35
0407	4567		MNTPAK	
0410	1064		TAD	ONSYS2
0411	7640		SZA	CLA
0412	5250		JMP	STEP4
0413	1060		TAD	UNIT2
0414	0040		AND	K3

				/AC=3
				/STEP # = 3
				/UNIT # = 2
				/GET UNIT 2 ACTIVE FLAG
				/SKIP IF UNIT 2 UNDER TEST
				/GO TO UNIT 0 OVERWRITE TEST
				/GO WAIT FOR MOUNTING PACK
				/GET UNIT 2 ON THIS SYS FLG
				/SKIP IF UNIT 2 ON THIS SYSTEM
				/UNIT 2 OTHER SYSTEM, GO TO NEXT STEP
				/GET UNIT 2 DRIVE #
				/MASK TO 2 BITS

0415	3024	DCA	CURDRV	/SAVE AS CURRENT DRIVE #
0416	1070	TAD	DSCU2	/GET UNIT 2 DEVICE CODE
0417	4564	CNGIOT		/CHANGE IOTS
0420	4560	RESET		/RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
0421	4560	RESET		/RESET DRIVE, CLEAR COMPOSITE ERROR FLG
0422	4776	JMS	RDYCHK	/CHECK FOR DRIVE READY
0423	5231	JMP	TST3	/DRIVE READY, GO TO TEST
0424	4551	MESAG		
0425	2507	DRVMSG		/"DRIVE"
0426	4551	MESAG		
0427	2611	NOTRDY		/"NOT READY"
0430	5200	JMP	STEP3	/GO TRY AGAIN
/				
0431	1073	TST3,	TAD	STRSTP
0432	7041		CIA	
0433	1103		TAD	STPNUM
0434	7640		SZA	CLA
0435	5242		JMP	XTST3
0436	4565		GETBSF	
0437	5775		JMP	MAIN
0440	4566		FNDTRK	
0441	5775		JMP	MAIN
/				
0442	4574	XTST3,	CMPSN	
0443	5200		JMP	STEP3
0444	4570		OVRWRT	
0445	5775		JMP	MAIN
0446	4571		ADJWRT	
0447	5775		JMP	MAIN
/				
0450	7307	STEP4,	CLA CLL	IAC RTL
0451	3103		DCA	STPNUM
0452	7325		CLA CLL	CML IAC RAL
0453	3035		DCA	CURUNT
0454	1061		TAD	UNIT3
0455	7710		SPA	CLA
0456	5774		JMP	STEP43
0457	4567		MNTPAK	
0460	1065		TAD	ONSY33
0461	7640		SZA	CLA
0462	5320		JMP	STEP5
0463	1061		TAD	UNIT3
0464	0040		AND	K3
0465	3024		DCA	CURDRV
0466	1071		TAD	DSCU3
0467	4564		CNGIOT	
0470	4560		RESET	
0471	4560		RESET	
0472	4776		JMS	RDYCHK
0473	5301		JMP	TST4
0474	4551		MESAG	
0475	2507		DRVMSG	
0476	4551		MESAG	
0477	2611		NOTRDY	
0500	5250		JMP	STEP4

```

/SAVE AS CURRENT DRIVE #
/GET UNIT 2 DEVICE CODE
/CHANGE IOTS
/RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
/RESET DRIVE, CLEAR COMPOSITE ERROR FLG
/CHECK FOR DRIVE READY
/DRIVE READY, GO TO TEST

/"DRIVE"

/"NOT READY"
/GO TRY AGAIN

/GET STARTING STEP

/COMPARE TO SEE IF THIS IS THE STARTING STEP
/SKIP IF THIS IS THE STARTING STEP
/GO EXECUTE TEST
/GET BAD SECTOR FILE
/TOO MANY BAD SECTORS
/FIND 10 SETS OF TRACKS
/TOO MANY BAD TRACKS

/GET AND COMPARE SERIAL NUMBER
/WRONG SERIAL NUMBER, TRY AGAIN
/DO OVER-WRITE TEST
/PROGRAM CONTINUED AFTER OVER-WRITE ERROR
/DO ADJACENT WRITE TEST
/PROGRAM CONTINUED AFTER ADJACENT WRITE ERROR

/AC = 4
/STEP # = 4

/UNIT # = 3
/GET UNIT 3 ACTIVE FLAG
/SKIP IF UNIT 3 ACTIVE
/UNIT 3 NOT UNDER TEST, GO DO UNIT 2 OVERWRITE
/GO WAIT FOR PACK MOUNT
/GET UNIT 3 ON THIS SYSTEM FLG
/SKIP IF UNIT 3 THIS SYSTEM
/UNIT 3 NOT ON THIS SYSTEM, GO TO NEXT STEP
/GET UNIT 3 DRIVE #
/MASK TO 3 BITS
/SAVE UNIT 3 DRIVE # AS CURRENT DRIVE
/GET UNIT 3 DEVICE CODE
/CHANGE IOTS
/RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
/RESET DRIVE, CLEAR COMPOSITE ERROR FLG
/CHECK FOR DRIVE READY
/DRIVE READY, GO TO TEST

/"DRIVE"

/"NOT READY"
/TRY AGAIN

```

0501	1073	/	TST4.	TAD	STRSTP	/GET STARTING STEP #
0502	7041			CIA		
0503	1103			TAD	STPNUM	/COMPARE WITH CURRENT STEP #
0504	7640			SZA	CLA	/SKIP IF EQUAL
0505	5312			JMP	XTST4	/THIS IS NOT THE STARTING STEP, GO DO TEST
0506	4565			GETBSF		/GET BAD SECTOR FILE
0507	5775'			JMP	MAIN	/TOO MANY BAD SECTORS
0510	4566			FNDTRK		/FIND 10 SETS OF 5 ADJACENT TRACKS
0511	5775'			JMP	MAIN	/TOO MANY BAD TRACKS
/						
0512	4574		XTST4.	CMPSN		/GET AND COMPARE SERIAL NUMBER
0513	5250			JMP	STEP4	/WRONG PACK, TRY AGAIN
0514	4570			OVRWRT		/DO OVER-WRITE TEST
0515	5775'			JMP	MAIN	/PROGRAM CONTINUED AFTER OVER-WRITE ERROR
0516	4571			ADJWRT		/DO ADJACENT WRITE TEST
0517	5775'			JMP	MAIN	/PROGRAM CONTINUE AFTER ADJACENT WRITE ERROR
/						
0520	1041		STEP5.	TAD	K5	
0521	3103			DCA	STPNUM	/STEP # = 5
0522	7305			CLA CLL	IAC RAL	
0523	3035			DCA	CURUNT	/UNIT # = 2
0524	1061			TAD	UNIT3	/GET UNIT 3 ACTIVE FLAG
0525	7710			SPA	CLA	/SKIP IF UNIT 3 ACTIVE
0526	5777'			JMP	STEP35	/GO TO STEP 5 FOR 2 UNITS UNDER TEST
0527	4567			MNTPAK		/WAIT FOR PACK MOUNT
0530	1064			TAD	ONSY82	/GET UNIT 2 ON SYS FLG
0531	7640			SZA	CLA	/SKIP IF UNIT 2 THIS SYSTEM
0532	5773'			JMP	STEP6	/UNIT 2 OTHER SYS, GO TO NEXT STEP
0533	1060			TAD	UNIT2	/GET UNIT 2 DRIVE #
0534	0040			AND	K3	/MASK TO 2 BITS
0535	3024			DCA	CURDRV	/SAVE AS CURRENT DRIVE #
0536	1070			TAD	DSCU2	/GET UNIT 2 DEVICE CODE
0537	4564			CNGIOT		/CHANGE IOTS
0540	4560			RESET		/RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
0541	4560			RESET		/RESET DRIVE, CLEAR COMPOSITE ERROR FLG
0542	4776'			JMS	RDYCHK	/CHECK FOR DRIVE READY
0543	5351			JMP	TST5	/DRIVE READY, GO TO TEST
0544	4551			MESAG		
0545	2507			DRVMSG		/"DRIVE"
0546	4551			MESAG		
0547	2611			NOTRDY		/"NOT READY"
0550	5320			JMP	STEP5	/TRY AGAIN
/						
0551	1073		TST5.	TAD	STRSTP	/GET STARTING STEP #
0552	7041			CIA		
0553	1103			TAD	STPNUM	/COMPARE WITH CURRENT STEP #
0554	7640			SZA	CLA	/SKIP IF THIS IS THE STARTING STEP
0555	5772'			JMP	XTST5	/GO DO TEST
0556	4565			GETBSF		/GET BAD SECTOR FILE
0557	5775'			JMP	MAIN	/TOO MANY BAD SECTORS
0560	4566			FNDTRK		/FIND ADJACENT TRACKS
0561	5775'			JMP	MAIN	/TOO MANY BAD TRACKS
0562	5772'			JMP	XTST5	/GO TO NEXT PAGE

0572 0600
 0573 0606
 0574 0610
 0575 0213
 0576 2333
 0577 0661
 0600

PAGE

```

0600 4574 XTST5, CMPSN /GET AND COMPARE SERIAL NUMBER
0601 5777' JMP STEP5 /WRONG PACK, TRY AGAIN
0602 4570 OVRWRT /DO OVER-WRITE TEST
0603 5776' JMP MAIN /PROGRAM CONTINUED AFTER OVER-WRITE ERROR
0604 4571 ADJWRT /DO ADJACENT WRITE TEST
0605 5776' JMP MAIN /PROGRAM CONTINUED AFTER ADJACENT WRITE ERROR

0606 7327 STEP6, CLA CLL CML IAC RTL
0607 3103 DCA STPNUM /STEP # = 6
0610 7001 STEP43, IAC /STEP #4 IF 3 UNITS UNDER TEST
0611 3035 DCA CURUNT /UNIT # = 1
0612 4567 PAK6, MNTPAK /CALL MOUNT PACK ROUTINE
0613 1063 TAD ONSYS1 /GET UNIT 1 ON SYS FLG
0614 7640 SZA CLA /SKIP IF UNIT 1 THIS SYS
0615 5254 JMP STEP7 /UNIT 1 OTHER SYS, GO TO NEXT STEP
0616 1057 TAD UNIT1 /GET UNIT 1 DRIVE #
0617 0040 AND K3 /MASK TO 2 BITS
0620 3024 DCA CURDRV /SAVE UNIT 1 DRIVE # AS CURRENT DRIVE
0621 1067 TAD DSCU1 /GET UNIT 1 DEVICE CODE
0622 4564 CNGIOT /CHANGE IOTS
0623 4560 RESET /RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
0624 4560 RESET /RESET DRIVE, CLEAR COMPOSITE ERROR FLG
0625 4775' JMS RDYCHK /CHECK FOR DRIVE READY
0626 5234 JMP TST6 /DRIVE READY, GO DO TEST
0627 4551 MESAG
0630 2507 DRVMSG
0631 4551 MESAG
0632 2611 NOTRDY
0633 5212 JMP PAK6 /"DRIVE"

0634 1073 TST6, TAD STRSTP /GET STARTING STEP #
0635 7041 CIA /COMPARE WITH CURRENT STEP #
0636 1103 TAD STPNUM /SKIP IF EQUAL
0637 7640 SZA CLA /GO DO TEST, THIS IS NOT THE STARTING STEP
0640 5246 JMP XTST6 /GET BAD SECTOR FILE
0641 4565 GETBSF /TOO MANY BAD SECTORS
0642 5776' JMP MAIN /FIND 10 SETS OF TRACKS
0643 4566 FNDTRK /TOO MANY BAD TRACKS
0644 5776' JMP MAIN /GO TO TEST, NEXT PAGE
0645 5246 JMP XTST6

0646 4574 XTST6, CMPSN /GET SERIAL NUMBER AND COMPARE
0647 5206 JMP STEP6 /WRONG PACK, TRY AGAIN
0650 4570 OVRWRT /DO OVER-WRITE TEST

```



```

0651 5776'      JMP      MAIN      /PROGRAM CONTINUED AFTER OVER-WRITE ERROR
0652 4571      ADJWRT /DO ADJACENT WRITE TEST
0653 5776'      JMP      MAIN      /PROGRAM CONTINUED AFTER ADJACENT WRITE ERROR

/
0654 1061      STEP7, TAD      UNIT3 /GET UNIT 3 ACTIVE FLAG
0655 7710      SPA      CLA      /SKIP IF UNIT 3 ACTIVE
0656 7344      CLA CLL CMA RAL /AC=-2
0657 1042      TAD      K7
0660 3103      DCA      STPNUM /STEP # = 5 OR 7
0661 3035      STEP35, DCA      CURUNT /UNIT # = 0, THIS IS STEP # 3 IF 2 UNITS, STEP # 5 IF 3 UNITS
0662 4567      PAK7,  MNTPAK /GO WAIT FOR PACK MOUNT
0663 1062      TAD      ONSYSO /GET UNIT 0 ON THIS SYS FLG
0664 7640      SZA      CLA      /SKIP IF UNIT 0 THIS SYS
0665 5323      JMP      ENDTST /UNIT 0 OTHER SYS, DONE
0666 1056      TAD      UNITO /GET UNIT 0 DRIVE #
0667 0040      AND      K3      /MASK TO 2 BITS
0670 3024      DCA      CURDRV /SAVE UNIT 0 DRIVE # AS CURRENT DRIVE
0671 1066      TAD      DSCUO /GET UNIT 0 DEVICE CODE
0672 4564      CNGIOT /CHANGE IOTS
0673 4560      RESET /RESET DRIVE, SELECT DRIVE, CLEAR VOLUME CHECK
0674 4560      RESET /RESET DRIVE, CLEAR COMPOSITE ERROR FLG
0675 4775'      JMS      RDYCHK /CHECK DRIVE READY
0676 5304      JMP      TST7 /DRIVE READY, GO DO TEST
0677 4551      MESAG
0700 2507      DRVMSG
0701 4551      MESAG
0702 2611      NOTRDY
0703 5262      JMP      PAK7 /"DRIVE"

/
0704 1073      TST7,  TAD      STRSTP /"NOT READY"
0705 7041      CIA
0706 1103      TAD      STPNUM /TRY AGAIN
0707 7640      SZA      CLA
0710 5315      JMP      XTST7 /GET STARTING STEP #
0711 4565      GETBSF /COMPARE WITH CURRENT STEP #
0712 5776'      JMP      MAIN /SKIP IF EQUAL
0713 4566      FNDTRK /THIS IS NOT THE STARTING STEP, GO DO TEST
0714 5776'      JMP      MAIN /GET BAD SECTOR FILE
/
0715 4574      XTST7, CHPSN /TOO MANY BAD SECTORS
0716 5254      JMP      STEP7 /FIND 10 SETS OF TRACKS
0717 4570      OVRWRT /TOO MANY BAD TRACKS
0720 5776'      JMP      MAIN
0721 4571      ADJWRT
0722 5776'      JMP      MAIN

/
0723 4551      ENDTST, MESAG /GET AND COMPARE SERIAL NUMBER
0724 2747      TSTCOM /WRONG PACK, TRY AGAIN
0725 4535      C8CALL /DO OVER-WRITE TEST
0726 7000      NOP /PROGRAM CONTINUED AFTER OVER-WRITE ERROR
0727 5776'      JMP      MAIN /DO ADJACENT WRITE TEST
/
/
/
/PROGRAM CONTINUED AFTER ADJACENT WRITE ERROR

0775 2333

```

0776 0213
0777 0520
1000

PAGE

/

/

/

***** SUBROUTINES *****

/ROUTINE TO WRITE A SECTOR

/USE BUFFER 1 FLD 1

/BUFFER MUST CONTAIN DATA WHEN THIS ROUTINE IS CALLED

/"SECSAV" CONTAINS SECTOR NUMBER IN 6:11

/DATA IS IN BUFFER FLD 1 STARTING AT BUFAD1, 1 SECTOR, 256 WORDS

1000 0000
1001 7300
1002 4536
1003 0017
1004 1054
1005 3504

WRTSEC, 0

CLA CLL

SETPNT

WRDCNT

TAD M400

DCA I TBLPNT

/SET TBLPNT TO WORD COUNT

/TBL INDX

/8 BIT MODE WORD COUNT (ONE SECTOR)=-256(10)

/SAVE WORD COUNT

1006 4536
1007 0020
1010 1106
1011 3504
1012 1041
1013 3105
1014 4547
1015 4542
1016 5221
1017 4541
1020 5246
1021 4550
1022 5225
1023 4541
1024 5246

WRT1,

SETPNT

INITCA

TAD BUFAD1

DCA I TBLPNT

TAD K5

DCA FUNCOD

GETSTA

ERRCHK

JMP .+3

ERROR

JMP WRTEXA

STACHK

JMP .+3

ERROR

JMP WRTEXA

/SET TBLPNT TO INITIAL CURRENT ADDRESS

/TBL INDX

/8 BIT MODE USE BUFFER 1 ADDRESS

/SAVE BUFFER ADDRESS AS INITIAL CURRENT ADDRESS

/FUNCTION CODE=5 (WRITE)

/GET STATUS OF DRIVE

/CHECK FOR ERRORS

/NO ERROR RETURN

/STATUS ERROR OR CONTROLLER ERROR

/NOT RECOVERABLE RETURN DRIVE DROPPED

/CHECK STATUS WORD #1

/NO STATUS ERROR

/STATUS ERROR

/NOT RECOVERABLE RETURN DRIVE DROPPED

```

1025 4536      SETPNT      /SET TBLPNT TO SECTOR ADDRESS
1026 0021      SECADD      /TABLE INDEX
1027 1121      TAD         SECSAV    /GET SECTOR
1030 3504      DCA I      TBLPNT    /SAVE FOR WRITE
1031 1121      TAD         SECSAV    /GET SECTOR ADDRESS
1032 7001      IAC         /INCREMENT SECTOR ADDRESS FOR FINAL SECTOR REG.
1033 4536      SETPNT      /SET TBLPNT TO EXPECTED FINAL SECTOR REGISTER
1034 0024      XENDSC      /TBL INDX
1035 3504      DCA I      TBLPNT    /SAVE EXPECTED FINAL SECTOR ADDRESS
1036 4545      XFER        /WRITE SECTOR
1037 4544      WAITDN      /WAIT FOR DONE INTERRUPT
1040 4777      JMS         FATAL     /NO DONE INTERRUPT TIMEOUT
1041 4542      ERRCHK      /CHECK FOR ERRORS
1042 5245      JMP         .+3       /NO CONTROLLER OR STATUS WORD #1 ERROR
1043 4541      ERROR       /STATUS ERROR OR CONTROLLER ERROR
1044 5246      JMP         WRTXA     /NOT RECOVERABLE ERROR EXIT

/
1045 2200      ISZ         WRTSEC     /INCREMENT RETURN ADDRESS FOR GOOD EXIT
1046 5600      WRTXA, JMP I WRTSEC    /RETURN
/
/
/*****
/
/ROUTINE TO GET HARDWARE OR SOFTWARE SWR
/IF BIT 0 OF HCW1 IS A ONE USE HARDWARE SWR
/ELSE USE LOC 20 (PSEUDO SWR)
/
/      CALLED BY:      GETSWR
/
/
1047 0000      SWRGET, 0
1050 7300      CLA CLL
1051 1021      TAD         HCW1      /GET HCW1
1052 7710      SPA CLA      /SKIP IF USING SOFTWARE SWR
1053 7614      LAS SKP      /GET HARDWARE SWR
1054 1020      TAD         PSR      /GET PSEUDO SWR
1055 5647      JMP I      SWRGET     /EXIT
/
/
/*****
/
/ROUTINE TO CALL DECIMAL PRINT ROUTINE IN FIELD 1
/
/      CALLED BY:      DECPRN      (FROM FLD 0)
/
/
1056 0000      PRNDEC, 0
1057 6212      CIF         10       /GOING TO FIELD 1
1060 4776      JMS I      (PRTDEC    /GO TO FLD 1 DECIMAL PRINT ROUTINE
1061 5656      JMP I      PRNDEC     /RETURN
/
/*****
/

```

/ROUTINE TO CALL AC PRINT ROUTINE IN FIELD 1

/ CALLED BY: PRNAC

1062	0000	ACPRN,	0		
1063	6212	CIF	10	/INS FLD#1	
1064	4775	JMS I	(ACPRNT	/GO TO PRINT AC ROUTINE FLD 1	
1065	5662	JMP I	ACPRN	/RETURN	

/*****

/ROUTINE TO DELAY FOR ONE SECOND TIMES UP TO 4096

/ CALLED BY: JMS DELAY
 / FOLLOWED BY: -XXXX WHERE XXXX IS THE 2'S COMPLEMENT OF NUMBER
 / OF SECONDS TO DELAY (OCTAL)

/CALLS ROUTINE IN FLD 1

1066	0000	DELAY,	0		
1067	7300	CLA	CLL		
1070	1666	TAD I	DELAY	/GET 2'S COMPLEMENT OF NUMBER OF SECONDS TO DELAY	
1071	3274	DCA	DLYCAL+1	/SAVE FOR DELAY ROUTINE CALL	
1072	6212	CIF	10	/GOING TO FLD 1	
1073	4774	DLYCAL,	JMS I (XDELAY	/GO TO FLD 1 DELAY ROUTINE	
1074	7402	HLT/DELAY	COUNT	/SECONDS MULTIPLIER	
1075	2266	ISZ	DELAY	/INCREMENT FOR RETURN	
1076	5666	JMP I	DELAY	/RETURN	

/*****

/ROUTINE TO CALL YES OR NO ANSWER ROUTINE IN FIELD 1

/ CALLED BY: YESRN

1077	0000	YNGO,	0		
1100	7300	CLA	CLL		
1101	6212	CIF	10	/INS FLD = 1	
1102	4773	JMS I	(XYESNO	/GO TO YES OR NO ROUTINE FLD 1	
1103	7610	SKP	CLA	/YES RETURN	
1104	2277	ISZ	YNGO	/NO RETURN	
1105	5677	JMP I	YNGO	/EXIT	

```

/*****
/
/ROUTINE TO CALL PRINT 1 CHARACTER ROUTINE IN FIELD 1
/
/   CALLED BY:      PRNT
/
/AC CONTAINS CHARACTER TO PRINT AT TIME OF CALL
/
/
/
1106 0000 PRNTGO, 0
1107 6212     CIF      10      /INS FLD = 1
1110 4772     JMS I    (XPRINT /GO TO PRINT ROUTINE FLD 1
1111 5706     JMP I    PRNTGO   /EXIT
/
/*****
/
/ROUTINE TO READ A SECTOR AND VERIFY DATA
/
/   CALLED BY:      JMS      RDSEC
/
/RETURN CALL+1 IF UNRECOVERABLE ERROR
/RETURN CALL+2 IF OK
/
/
/LOCATION "SECSAV" CONTAINS SECTOR NUMBER IN 6:11
/LOCATION "PATNUM" CONTAINS DATA PATTERN #
/
/
/
1112 0000 RDSEC, 0
1113 7300     CLA CLL
1114 4536     SETPNT      /SET TBLPNT TO WORD COUNT
1115 0017     WRDCNT      /TBL INDX
1116 1054     TAD      M400 /8 BIT MODE, WRD CNT = 256(10)
1117 3504     DCA, I    TBLPNT /SAVE WORD COUNT IN DRIVE STATE TABLE
/
1120 4547     GETSTA      /GET STATUS
1121 4542     ERRCHK      /CHECK FOR STATUS ERRORS
1122 5325     JMP      .+3 /NO ERROR
1123 4541     ERROR      /STATUS ERROR
1124 5364     JMP      RDEXA /ERROR NOT RECOVERABLE
1125 4550     STACHK      /CHECK STATUS
1126 5331     JMP      .+3 /NO ERROR
1127 4541     ERROR      /STATUS ERROR
1130 5364     JMP      RDEXA /ERROR NOT RECOVERABLE
/
1131 4536     SETPNT      /SET TBLPNT TO SECTOR ADDRESS
1132 0021     SECADD      /TBL INDX
1133 1121     TAD      SECSAV /GET SECTOR

```

1134	3504	DCA I	TBLPNT	
/				
1135	4536	SETPNT		/SET TBLPNT TO CURRENT BUFFER ADDRESS
1136	0020	INITCA		/TBL INDX
1137	1371	TAD	(RBUFF	/GET READ BUFFER ADDRESS
1140	3504	DCA I	TBLPNT	/SAVE AS CURRENT ADDRESS
1141	4770	JMS	CLRRBF	/GO CLEAR THE READ DATA BUFFER
/				
1142	1121	TAD	SECSAV	/GET SECTOR ADDRESS
1143	7001	IAC		/ADD 1 TO SECTOR ADDRESS FOR FINAL SECTOR REG.
1144	4536	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR REGISTER
1145	0024	XENDSC		/TBL INDX
1146	3504	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR REGISTER
1147	7327	CLA CLL	CML IAC RTL	/FUNCTION CODE=6 (READ)
1150	3105	DCA	FUNCOD	/SAVE FUNCTION CODE FOR READ
1151	4545	XFER		/READ DATA (1 SECTOR)
1152	4544	WAITDN		/WAIT FOR DONE
1153	4777	JMS	FATAL	/FATAL ERROR - NO DONE FLAG AFTER READ
1154	4542	ERRCHK		/CHECK FOR ERRORS
1155	5360	JMP	SECRDA	/NO ERROR RETURN, CHECK DATA
1156	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR
1157	5364	JMP	RDEXA	/ERROR NOT RECOVERED - DRIVE DROPPED
/				
1160	4767	SECRDA,	JMS	CHKSEC
1161	7410		SKP	
1162	5364		JMP	RDEXA
/				
1163	2312		ISZ	RDSEC
1164	5712	RDEXA,	JMP I	RDSEC
/				
/				
1167	1200			
1170	5131			
1171	6600			
1172	1316			
1173	1273			
1174	0670			
1175	0637			
1176	1600			
1177	7336			
	1200			

PAGE

/

/

/*****

/

/

/ROUTINE TO CHECK A SECTOR OF DATA

/

/

CALLED BY: JMS CHKSEC

/

/RETURN CALL+2 IF DATA ERROR

/RETURN CALL+1 IF DATA OK

/LOCATION "INITCA" OF DRV TBL MUST CONTAIN BUFFER ADDRESS FOR START OF SECTOR

/AUTO15 USED FOR DATA BUFFER (DATA READ TO BE CHECKED)

/LOCATION "CURDRV" MUST CONTAIN DRIVE NUMBER
 /LOCATION "PATNUM" MUST CONTAIN DATA PATTERN #

```

1200 0000      CHKSEC, 0
1201 7300          CLA CLL
1202 4536          SETPNT
1203 0020          INITCA
1204 1504          TAD I      TBLPNT
1205 1052          TAD      M1
1206 3015          DCA      AUTO15
1207 1054          TAD      M400
1210 3262          DCA      WDCTR
1211 3117          DCA      DERFLG
1212 3777          DCA      BDWCNT

1213 1113      CHK8,  TAD      PATNUM
1214 1376          TAD      (DPAT8
1215 3120          DCA      PATMP
1216 1520          TAD I      PATMP
1217 3120          DCA      PATMP
1220 1520          TAD I      PATMP
1221 3263          DCA      DATWD1
1222 2120          ISZ      PATMP
1223 1520          TAD I      PATMP
1224 3264          DCA      DATWD2
1225 3265          DCA      DATSWT

1226 6211      LPA1,  CDF      10
1227 1415          TAD I      AUTO15
1230 6201          CDF      00
1231 0045          AND      K377
1232 7041          CIA
1233 1263          TAD      DATWD1
1234 7640          SZA CLA
1235 4775          JMS      DERR
1236 2262          ISZ      WDCTR
1237 7610          SKP      CLA
1240 5256          JMP      CHKRET
1241 7240          CLA      CMA
1242 3265          DCA      DATSWT

1243 6211          CDF      10
1244 1415          TAD I      AUTO15
1245 6201          CDF      00
1246 0045          AND      K377
1247 7041          CIA
1250 1264          TAD      DATWD2
1251 7640          SZA      CLA
1252 4775          JMS      DERR
1253 3265          DCA      DATSWT
1254 2262          ISZ      WDCTR
1255 5226          JMP      LPA1

1256 1117      CHKRET, TAD      DERFLG

```

```

/SET TBLPNT TO BUFFER ADDRESS OF SECTOR DATA
/TBL INDX
/GET BUFFER ADDRESS OF SECTOR TO CHECK
/SUBTRACT 1 FOR AUTO INDX
/SET UP BUFFER POINTER
/WORD COUNT=256
/SET UP SECTOR WORD COUNTER
/CLEAR DATA ERROR FLAG
/CLEAR BAD DATA WORD COUNTER

/GET DATA PATTERN NUMBER
/INDEX INTO PATTERN POINTER TBL
/SAVE POINTER
/GET ADDRESS OF PATTERN FROM TABLE
/SAVE POINTER TO DATA WORD
/GET FIRST DATA WORD
/SAVE FIRST DATA WORD OF PATTERN
/INCREMENT POINTER
/GET SECOND WORD OF DATA PATTERN
/SAVE SECOND DATA WORD
/CLEAR DATA SWITCH FOR FIRST DATA WORD

/GET BUFFER WRD

/MASK OFF GARBAGE BITS
/NEGATE BUFFER WRD
/ADD ODD NUMBERED DATA WORD
/SKIP IF GOOD WORD=BUFFER WORD READ
/SECTOR DATA ERROR
/INCREMENT SECTOR WORD COUNTER
/SKIP IF NOT END OF SECTOR
/SECTOR COMPLETELY CHECKED, EXIT
/AC=-1
/SET DATA SWITCH FOR SECOND DATA WORD

/GET A DATA WRD
/CDF TO PRGM FLD
/MASK TO 8 BITS (DELETE GARBAGE)
/NEGATE FOR CHECK
/ADD EVEN NUMBERED DATA PATTERN WORD
/SKIP IF DATA WORD=BUFFER WORD
/SECTOR DATA ERROR
/CLEAR DATA SWITCH FOR ODD DATA WORD
/INCREMENT SECTOR WORD CNTR
/CONTINUE CHECKING SECTOR

```

/GET DATA ERROR FLAG

```

1257 7640          SZA   CLA      /SKIP IF NO DATA ERRORS
1260 2200          ISZ   CHKSEC  /INCREMENT FOR BAD RETURN
1261 5600          JMP I  CHKSEC  /RETURN

/
1262 0000          WDCTR, 0      /SECTOR WORD COUNTER
1263 0000          DATWD1, 0     /EVEN NUMBERED DATA WORD
1264 0000          DATWD2, 0     /ODD NUMBERED DATA WORD
1265 0000          DATSWT, 0     /0=DATWD1  7777=DATWD2
/
/
/*****
/
/ROUTINE TO RESET A DRIVE
/CALLED BY:      RESET
/RETURN ONLY IF SUCCESSFUL
/
1266 0000          XRESET, 0
1267 6002          IOF          /INTERRUPTS OFF
1270 7300          CLA          CLL
1271 4536          SETPNT       /SET TBLPNT TO COMMAND A
1272 0022          XCOMA       /TABLE INDEX
1273 3504          DCA I       TBLPNT /CLEAR COMMAND A
1274 4536          SETPNT       /SET TBLPNT TO SECTOR ADDRESS
1275 0021          SECADD       /TABLE INDEX
1276 3504          DCA I       TBLPNT /CLEAR SECTOR ADDRESS
1277 4536          SETPNT       /SET TBLPNT TO EXPECTED FINAL SECTOR ADDRESS
1300 0024          XENDSC       /TBL INDX
1301 3504          DCA I       TBLPNT /EXPECTED FINAL SECTOR ADDRESS = 0
1302 4536          SETPNT       /SET TBLPNT TO COMMAND B
1303 0023          XCOMB       /TABLE INDEX
1304 1024          TAD          CURDRV /GET CURRENT DRIVE
1305 7002          BSW          /MOVE DRIVE SELECT TO BITS 4,5
1306 1374          TAD          (1401 /ADD INTERRUPT ENABLE, 8 BIT MODE, CODE 1
1307 3504          DCA I       TBLPNT /SAVE COMMAND B
1310 4543          GO          /EXECUTE COMMAND
1311 4544          WAITDN       /WAIT FOR DONE OR ERROR
1312 4773          JMS          FATAL /DONE DID NOT SET
1313 5666          JMP I       XRESET /EXIT - RESET WAS SUCCESSFUL

/
/
/*****
/
/ROUTINE TO CHECK FOR DRIVE WRITE LOCKED
/
/      CALLED BY:      JMS      WRTLOC
/
/RETURN CALL+1 IF NOT WRITE LOCKED

```


/PRINT ERROR MESSAGE IF DRIVE WRITE PROTECTED

```

1314 0000 WRTLOC, 0
1315 1027 TAD STAT6B /GET STATUS WORD #2
1316 0372 AND (40 /TEST BIT 6 (WRT PROT)
1317 7650 SNA CLA /SKIP IF DRIVE WRT PROTECTED
1320 5714 JMP I WRTLOC /RETURN CALL+1 DRV NOT WRT PROT
1321 4556 DOCRLF /<CR LF>
1322 4551 MESAG
1323 2507 DRVMSG /"DRIVE "
1324 1025 TAD DRV60 /GET CURRENT DRIVE
1325 1044 TAD K260 /ADD ASCII BASE CODE
1326 4553 PRNT /PRINT DRIVE NUMBER
1327 4551 MESAG
1330 2711 WRTPH /"WRITE PROTECTED"
1331 4535 C8CALL /ASK SR = QUESTION
1332 7000 NOP
1333 5714 JMP I WRTLOC /PROGRAM CONTINUED

```

/*****

/ROUTINE TO ASK FOR AND ACCEPT A DRIVE NUMBER

/ CALLED BY: JMS WCHDR

```

1334 0000 WCHDR, 0
1335 4556 DOCRLF /DO A CR AND LF
1336 4551 MESAG
1337 2507 DRVMSG /DRIVE
1340 1371 TAD (277 /GET ASCII CODE FOR "?"
1341 4553 PRNT /?
1342 4554 GETOCT /GET ONE OCTAL DIGIT
1343 3024 DCA CURDRV /SAVE AS POSSIBLE CURRENT DRIVE
1344 6211 CDF 10 /DF=1
1345 1770 TAD I (DIGFLG /GET DIGIT INPUT FLAG
1346 6201 CDF 00 /DF=0
1347 7650 SNA CLA /SKIP IF DIGIT INPUT
1350 5335 JMP WCHDR+1 /GO TRY AGAIN FOR DRIVE NUMBER
1351 7346 CLA CLL CMA RTL /AC=-3
1352 1024 TAD CURDRV /ADD CURRENT DRIVE
1353 7740 SNA SZA CLA /SKIP IF VALID DRV #
1354 5335 JMP WCHDR+1 /GO ASK DRV QUESTION INVALID DRV #
1355 5734 JMP I WCHDR /EXIT

```

1370 0055
 1371 0277
 1372 0040
 1373 7336
 1374 1401
 1375 7230
 1376 2732
 1377 7317
 1400

PAGE

/

/

/*****

/

/

/

/ROUTINE TO RETRIEVE AND PROCESS A BAD SECTOR FILE

/FROM A PACK

/CALLED BY: GETBSF

/RETURN CALL+1 IF DRIVE DROPPED

/RETURN CALL+2 IF OK

/

/

1400 0000
 1401 4546
 1402 2777
 1403 5600
 1404 7300
 1405 4536
 1406 0020
 1407 1106
 1410 3504
 1411 1377
 1412 3776
 1413 1375
 1414 3116

BSFGET, 0

SETPOS

2777

/SEEK TO BAD SECTOR FILE TRACK

HP 001

/UNSUCCESSFUL RETURN, EXIT

JMP I BSFGET

CLA CLL

/SET TBLPNT TO INITIAL CURRENT ADDR

SETPNT

/TBL INDX

INITCA

TAD BUFAD1

/GET ADDR OF FLD 1 BUFFER

DCA I TBLPNT

/SAVE AS INITIAL CURRENT ADDRESS

TAD (-5

/FIVE BAD SECTOR FILES FOR FACTORY

DCA BDCNT

/SET UP ATTEMPT COUNT TO READ BAD SECTOR FILE

TAD (FACBAD

/GET ADDRESS OF TABLE OF FACTORY BAD SECTOR FILE SECTOR NUMBERS

DCA BADPNT

/SET UP TABLE POINTER

/

1415 4774
 1416 4547
 1417 4542
 1420 5223
 1421 4541
 1422 5600
 1423 4773
 1424 5227
 1425 4541
 1426 5600
 1427 4536
 1430 0017
 1431 1054
 1432 3504
 1433 4536
 1434 0021

TRYLPA, JMS CLRBF1

/CLEAR BUFFER 1

GETSTA

/GET STATUS OF DRIVE

ERRCHK

/CHECK FOR ERRORS

JMP .+3

/NO ERROR RETURN

ERROR

/STATUS ERROR OR CONTROLLER ERROR

JMP I BSFGET

/DRIVE DROPPED ERROR NOT RECOVERABLE

JMS RDYCHK

/CHECK FOR DRIVE READY

JMP .+3

/DRIVE READY RETURN

ERROR

/DRIVE NOT READY ERROR

JMP I BSFGET

/ERROR NOT RECOVERABLE

SETPNT

/SET TBLPNT TO WORD COUNT

WRDCNT

/TABLE INDEX

TAD M400

/WORD COUNT=256(10) (ONE SECTOR)

DCA I TBLPNT

/SAVE WORD COUNT

SETPNT

/SET TBLPNT TO SECTOR ADDRESS IN DRV STATE TBL

SECADD

/TBL INDX

1435	1516	TAD I	BADPNT	/GET A BAD SECTOR FILE SECTOR NUMBER
1436	3504	DCA I	TBLPNT	/SAVE AS SECTOR ADDRESS
1437	1504	TAD I	TBLPNT	/GET SECTOR ADDRESS
1440	7001	IAC		/ADD ONE FOR EXPECTED FINAL SECTOR REGISTER
1441	4536	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR REG.
1442	0024	XENDSC		/TBL INDX
1443	3504	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR REG. IN DRV TBL
1444	7327	CLA CLL	CML IAC RTL	/CODE 6 READ DATA
1445	3105	DCA	FUNCOD	/SAVE AS FUNCTION CODE
1446	4545	XFER		/READ BAD SECTOR FILE
1447	4544	WAITDN		/WAIT FOR DONE
1450	4772	JMS	FATAL	/NO DONE INTERRUPT
1451	4542	ERRCHK		/CHECK FOR ERRORS
1452	5275	JMP	RDFLD	/NO ERROR RETURN GO READ FIELD BAD SECTOR FILE
1453	2116	ISZ	BADPNT	/INCREMENT BAD SECTOR POINTER
1454	2776	ISZ	BDCNT	/INCREMENT READ ATTEMPT COUNT
1455	5215	JMP	TRYLPA	/TRY NEXT BAD SECTOR FILE
1456	4551	MESAG		
1457	2543	CNTRD1		/"CAN'T READ FACTORY"
1460	4551	MESAG		
1461	2555	BDSEC		/"BAD SECTOR FILE ON"
1462	4551	MESAG		
1463	2507	DRVMSG		/"DRIVE"
1464	1024	TAD	CURDRV	/GET DRIVE #
1465	1044	TAD	K260	/ADD ASCII BASE CODE
1466	4553	PRNT		/PRINT DRIVE #
1467	4551	MESAG		
1470	2732	BADPAK		/"BAD PACK1"
1471	4551	MESAG		
1472	2740	CHNGP		/"CHANGE PACK"
1473	4562	WAITCR		/WAIT FOR A CARRIAGE RETURN
1474	5600	JMP I	BSFGET	/RETURN
1475	4536	RDFLD,	SETPNT	/SET TBLPNT TO INITIAL CURRENT ADDR
1476	0020		INITCA	/TBL INDX
1477	1106	TAD	BUFAD1	/GET ADDR OF BUFFER
1500	1046	TAD	K400	/ADD 256(10) TO PRESERVE FACTORY BAD SECTOR
1501	3504	DCA I	TBLPNT	/SET UP INITIAL CURRENT ADDR
1502	1377	TAD	(-5	/
1503	3776	DCA	BDCNT	/SET UP TO READ ALL 5 FIELD BAD IF NECESSARY
1504	1371	TAD	(FLDBAD	/GET ADDR OF TABLE OF FIELD BAD SECTOR ADDR
1505	3116	DCA	BADPNT	/SET UP POINTER TO TABLE
1506	5311	JMP	TRYLPB	/GO TO NEXT PAGE
1507	2200	ISZ	BSFGET	/INCREMENT FOR GOOD RETURN
1510	5600	BSFEX,	JMP I	/RETURN
1511	4547	TRYLPB,	GETSTA	/GET DRV STATUS
1512	4542		ERRCHK	/CHECK FOR ERRORS
1513	5316	JMP	.+3	/NO ERROR RETURN
1514	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR
1515	5310	JMP	BSFEX	/NOT RECOVERABLE ERROR

1516	4773'	JMS	RDYCHK	/CHECK FOR DRIVE READY
1517	5322	JMP	.+3	/DRIVE READY RETURN
1520	4541	ERROR		/DRIVE NOT READY
1521	5310	JMP	BSFEX	/NOT RECOVERABLE ERROR
1522	4536	SETPNT		/SET TBLPNT TO SECTOR ADDRESS
1523	0021	SECADD		/TBL INDX
1524	1516	TAD I	BADPNT	/GET BAD SECTOR FILE SECTOR
1525	3504	DCA I	TBLPNT	/SAVE AS SECTOR ADDRESS
1526	1504	TAD I	TBLPNT	/GET SECTOR ADDRESS
1527	7001	IAC		/ADD ONE FOR FINAL EXPECTED SECTOR REG.
1530	4536	SETPNT		/SET TBLPNT TO FINAL SECTOR REG. IN DRV TBL
1531	0024	XENDSC		/TBL INDX
1532	3504	DCA I	TBLPNT	/SAVE FINAL EXPECTED SECTOR REG.
1533	4545	XFER		/READ BAD SECTOR FILE
1534	4544	WAITDN		/WAIT FOR DONE
1535	4772'	JMS	FATAL	/NO DONE INTERRUPT
1536	4542	ERRCHK		/CHECK FOR ERRORS
1537	5770'	JMP	BADPRO	/GO PROCESS BAD SECTORS
1540	2116	ISZ	BADPNT	/INCR TBL POINTER TO NXT BAD SECTOR FILE SECTOR
1541	2776'	ISZ	BDCNT	/INCREMENT READ ATTEMPT COUNTER
1542	5311	JMP	TRYLPB	/TRY NEXT BAD SECTOR FILE
1543	4551	MESAG		
1544	2566	CNTRD2		/"CAN'T READ FIELD"
1545	4551	MESAG		
1546	2555	BDSEC		/"BAD SECTOR FILE ON"
1547	4551	MESAG		
1550	2507	DRVMSG		/"DRIVE"
1551	1024	TAD CURDRV		/GET DRIVE #
1552	1044	TAD	K260	/ADD ASCII BASE CODE
1553	4553	PRNT		/PRINT DRIVE #
1554	4551	MESAG		
1555	2732	BADPAK		/"BAD PACK!"
1556	4551	MESAG		
1557	2740	CHNGP		/"CHANGE PACK"
1560	4562	WAITCR		/WAIT FOR A CARRIAGE RETURN
1561	5310	JMP	BSFEX	/RETURN

1570 1600
 1571 7561
 1572 7336
 1573 2333
 1574 7320
 1575 7554
 1576 1726
 1577 7773
 1600

PAGE

/PROCESS THE BAD SECTORS

1600 7344

BADPRO. CLA CLL CMA RAL

1601	3324	DCA	BADSWT	/SET UP FACTORY/FIELD SWITCH	
1602	1377	TAD	(-21	/16 BAD SECTORS ALLOWED NOT 17	
1603	3326	DCA	BDCNT	/SET UP BAD SECTOR COUNTER	
1604	1376	TAD	(13	/SET UP POINTER TO TRACK OF 1ST BAD SECTOR	
1605	1106	TAD	BUFAD1	/ADD ADDR OF BUFFER	
1606	3323	DCA	BUFPNT	/SET UP POINTER	
1607	1024	TAD	CURDRV	/GET CURRENT DRV # AS INDEX	
1610	1375	TAD	(BADSEC	/GET ADDR OF BAD SECTOR FILE	
1611	3111	DCA	TEMP1	/SET UP BAD SECTOR TBL POINTER	
1612	6211	CDP	10	/DATA FLD 1	
/					
1613	1723	BADLPB, TAD I	BUFPNT	/GET BAD CYLINDER	
1614	0045	AND	K377	/MASK OFF UNUSED BITS	
1615	7041	CIA		/NEGATE	
1616	1045	TAD	K377		
1617	7650	SNA CLA		/SKIP IF TRACK NOT ALL ONES	
1620	5275	JMP	FLDBD	/TRACK ALL ONES, END OF BAD SECTORS	
1621	2326	ISZ	BDCNT	/INCREMENT BAD SECTOR COUNT	
1622	7410	SKP		/SKIP IF < 16 BAD SECTORS	
1623	5261	JMP	TOOBAD	/MORE THAN 16 BAD SECTORS	
1624	7301	CLA CLL	IAC	/SET BIT 11	
1625	0723	AND I	BUFPNT	/MASK TRACK BIT	
1626	7112	CLL RTR			
1627	7010	RAR		/MOVE TRK BIT TO BIT 1	
1630	3325	DCA	BADTRK	/SAVE TRK BIT TEMPORARILY	
1631	7346	CLA CLL	CMA RTL		
1632	1323	TAD	BUFPNT	/SUBTRACT 3 FROM BUFFER POINTER	
1633	3323	DCA	BUFPNT	/BACK UP BUFFER POINTER TO CYL ADDR	
1634	1723	TAD I	BUFPNT	/GET CYL	
1635	0045	AND	K377	/MASK OFF UNUSED BITS	
1636	1325	TAD	BADTRK	/ADD TRK BIT	
1637	3511	DCA I	TEMP1	/SAVE BAD TRK AND CYL	
1640	2323	ISZ	BUFPNT	/INCREMENT BUFFER POINTER	
1641	7301	CLA CLL	IAC	/MASK FOR BIT 11 MSB OF CYL. ADD.	HP 003
1642	0723	AND I	BUFPNT	/GET THE LAST CYL WORD FROM BAD SECTOR FILE	HP 003
1643	7002	BSW		/PUT IT IN BIT 5 AND THEN MOVE IT	HP 003
1644	7106	CLL RTL		/LEFT TO THE CORRECT PLACE, IN BIT POSITION 3	HP 003
1645	1511	TAD I	TEMP1	/GET THE FIRST CYLINDER WORD AND HEAD BIT	HP 003
1646	3511	DCA I	TEMP1	/AND ADD THE MSB TO THE CYLINDER WORD	HP 003
1647	2111	ISZ	TEMP1	/BUMP POINTER TO STORE THE NEXT WORD	HP 003
				/FROM THE BAD SECTOR FILE, THE SECTOR #.	
1650	2323	ISZ	BUFPNT	/INCREMENT BUFFER POINTER	
1651	1723	TAD I	BUFPNT	/GET BAD SECTOR #	
1652	0045	AND	K377	/MASK OFF UNUSED BITS	
1653	3511	DCA I	TEMP1	/SAVE BAD SECTOR #	
1654	2111	ISZ	TEMP1	/INCREMENT BAD SECTOR STORAGE POINTER	
1655	1041	TAD	K5		
1656	1323	TAD	BUFPNT	/ADD 5 TO BUFFER POINTER	
1657	3323	DCA	BUFPNT	/BUFFER POINTER POINTS TO NEXT BAD CYLINDER	
1660	5213	JMP	BADLPB	/CONTINUE PROCESSING BAD SECTORS	
1661	6201	TOOBAD, CDP	00	/DATA FLD 0	
1662	4551	MESAG			
1663	2507	DRVMSG		/"DRIVE"	
1664	1024	TAD	CURDRV		
1665	1044	TAD	K260	/ADD ASCII BASE CODE	

1666	4553	PRNT		/PRINT DRIVE #
1667	4551	MESAG		
1670	2577	PAKBAD		/"PACK HAS MORE THAN 16 BAD SECTORS!"
1671	4551	MESAG		
1672	2740	CHNGP		/"CHANGE PACK"
1673	4562	WAITCR		/WAIT FOR A CARRIAGE RETURN
1674	5774	JMP	BSFEX	/RETURN
/				
1675	1106	FLDBD,	TAD	BUFAD1
1676	1373		TAD	(413
1677	3323		DCA	BUFPT
1700	2324		ISZ	BADSWT
1701	5213		JMP	BADLPB
/				
1702	1106	SNPROC,	TAD	BUFAD1
1703	1052		TAD	M1
1704	3016		DCA	AUTO16
1705	1416		TAD I	AUTO16
1706	0045		AND	K377
1707	3077		DCA	SERNM1
1710	1416		TAD I	AUTO16
1711	0045		AND	K377
1712	3100		DCA	SERNM2
1713	1416		TAD I	AUTO16
1714	0045		AND	K377
1715	3101		DCA	SERNM3
1716	1416		TAD I	AUTO16
1717	0045		AND	K377
1720	3102		DCA	SERNM4
/				
1721	6201		CDF	00
1722	5772		JMP	BSFEX-1
/				
1723	0000	BUFPT,	0	/BUFFER POINTER
1724	0000	BADSWT,	0	/FACTORY/FIELD SWITCH
1725	0000	BADTRK,	0	/BAD TRACK TEMP STORAGE
1726	0000	BDCNT,	0	/BAD SECTOR COUNTER
/				
/				
1772	1507			
1773	0413			
1774	1510			
1775	2200			
1776	0013			
1777	7757			
	2000	PAGE		
/				
/				
/				
/*****				
/				
/ROUTINE TO SERVICE INTERRUPTS				
/				
/				
2000	6003	INTSVC,	SRQ	/SKIP IF INTERRUPT REQUEST

2001	5230	JMP	NOTINT	/NO INTERRUPT REQUEST - GO TO ERROR ROUTINE
2002	3217	DCA	ACSAVE	/SAVE AC
2003	6004	GTF		/GET THE FLAGS
2004	3220	DCA	FLAGS	/SAVE THE FLAGS
2005	6102	SPL		/SKIP ON POWER LOW
2006	5233	JMP	FLGS	/POWER OK, GO CHECK FLAGS
2007	6103	CAL		/CLEAR AC LOW FLAG
2010	1000	TAD	LOCO	/GET INTERRUPT PC
2011	3216	DCA	PCSAVE	/SAVE INTERRUPT PC
2012	1215	TAD	STR TIN	/GET RESTART INSTRUCTION
2013	3000	DCA	LOCO	/SAVE FOR EXECUTION
2014	7402	HLT		/STOP EVERYTHING - POWER GOING DOWN
/				
2015	5403	STR TIN, JMP I	ADDR83	/RESTART INSTRUCTION
2016	7402	PCSAVE, HLT		/INTERRUPT PC
2017	7402	ACSAVE, HLT		/INTERRUPT AC
2020	7402	FLAGS, HLT		/FLAG SAVE AREA
2021	7402	SETTMP, HLT		/STORAGE OF SETPNT RETURN ADDRESS
2022	0000	TBL SAV, 0		/STORAGE OF TBLPNT
2023	4551	RESTR T, MESAG		
2024	2653	PWR FAL		/"POWER FAILED AT PC"
2025	1216	TAD	PCSAVE	/GET POWER FAIL PC
2026	4555	PRNAC		/PRINT PC
2027	5777	JMP	START	/START PROGRAM OVER
/				
2030	4551	NOTINT, MESAG		
2031	2665	NOINT		/"NO INTERRUPT REQUEST"
2032	7402	HLT		/FATAL ERROR - INTERRUPT SERVICE WAS /ENTERED BUT NO INTERRUPT REQUEST WAS /ACTIVE, TRY RELOADING PROGRAM
/				
/				
2033	6601	FLGS,	RLSD	/SKIP ON FUNCTION DONE FLAG SET
2034	7410		SKP	/FUNCTION DONE NOT SET
2035	7305		CLA CLL IAC RAL	/SET BIT 10 FUNCTION DONE FLAG SET
/				
2036	6617	IOTA,	RLSE	/SKIP ON COMPOSITE ERROR FLAG SET
2037	7410		SKP	/COMPOSITE ERROR FLAG NOT SET
2040	7001		IAC	/SET BIT 11 - COMPOSITE ERROR FLAG SET
/				
2041	3112		DCA	/SAVE STATUS OF CONTROLLER FLAGS
2042	1112		TAD	/GET FLAGS
2043	7650		SNA CLA	/SKIP IF EITHER DONE OR ERROR SET
2044	5274		JMP	/NOT DONE OR ERROR - GO CHECK KEYBOARD
2045	6610	IOTB,	RRER	/READ ERROR REGISTER
2046	7410		SKP	/IOT SHOULD NOT SKIP
2047	4776		JMS	/IOT RRER SKIPPED
2050	3030		DCA	/SAVE ERROR REGISTER
/				
2051	6611	IOTC,	RRWC	/READ FINAL WORD COUNT
2052	7410		SKP	/IOT SHOULD NOT SKIP
2053	4776		JMS	/IOT RRWC SKIPPED
2054	3034		DCA	/SAVE FINAL WORD COUNT
/				
2055	6612	IOTD,	RRCA	/READ COMMAND A

2056	7410	SKP		/IOT SHOULD NOT SKIP
2057	4776'	JMS	FATAL	/IOT RRCA SKIPPED
2060	3031	DCA	COMDA	/SAVE COMMAND A
/				
2061	6613	IOTE,	RRCB	/READ COMMAND B
2062	7410	SKP		/IOT SHOULD NOT SKIP
2063	4776'	JMS	FATAL	/IOT RRCB SKIPPED
2064	3032	DCA	COMDB	/SAVE COMMAND B
/				
2065	6614	IOTF,	RRSA	/READ SECTOR ADDRESS
2066	7410	SKP		/IOT SHOULD NOT SKIP
2067	4776'	JMS	FATAL	/IOT RRSA SKIPPED
2070	3033	DCA	ENDSC	/SAVE FINAL SECTOR ADDRESS
/				
2071	3026		DCA	STAT6A
2072	3027		DCA	STAT6B
2073	3055		DCA	CBUSY
/				
2074	6031	KBDCHK,	KSF	/SKIP IF KEYBOARD FLAG SET
2075	7610		SKP CLA	/KEYBOARD FLAG NOT SET
2076	5307		JMP	KEYBDRD
2077	6041	TTYCHK,	TSF	/READ KEYBOARD
2100	5303		JMP	+.3
2101	6042		TCF	/SKIP IF TELEPRINTER FLAG
2102	5314		JMP	INTRET
2103	1112		TAD	FLGSAV
2104	7650		SNA CLA	/NOT TTY FLG
2105	5321		JMP	UNKINT
2106	5314		JMP	INTRET
/				
2107	6036	KEYBDRD,	KRB	/READ KEYBOARD BUFFER
2110	0375		AND	(177
2111	1377		TAD	(200
2112	4535		C8CALL	/MASK TO 7 BITS
2113	7000		NOP	/SET PARITY BIT
2114	1220	INTRET,	TAD	FLAGS
2115	6005		RTF	/CALL CONSOLE PACKAGE
2116	7200		CLA	/CONSOLE NOT ACTIVE RETURN
2117	1217		TAD	ACSAVE
2120	5400		JMP I	LOCO
/				
2121	4551	UNKINT,	MESAG	
2122	2700		UNINT	
2123	7402		HLT	

/"UNKNOWN INTERRUPT"
/UNKNOWN INTERRUPT HALT

/*****

/ROUTINE TO PRINT A MESSAGE


```

/          CALLED BY:      MESAG
/          FOLLOWED BY:    MESNAME          /ADDRESS OF MESSAGE
/
/ CALLS ROUTINE MESPRT IN FLD 1
/
/
2124 0000 MESGO, 0
2125 7300 CLA CLL
2126 1724 TAD I MESGO /GET MESSAGE ADDRESS
2127 3332 DCA MESSAD /SAVE FOR FLD 1 CALL
2130 6212 CIF 10 /INSTRUCTION FLD = 1
2131 4774 JMS I (MESPRT /GO TO FLD 1 MESSAGE PRINTER
2132 0000 MESSAD, 0 /MESSAGE ADDRESS STORAGE
2133 2324 ISZ MESGO /INCREMENT RETURN ADDRESS
2134 5724 JMP I MESGO /RETURN
/
/
/
2174 1200
2175 0177
2176 7336
2177 0200
2200 PAGE
/
/
/
/
/*****
/
/ROUTINE TO CHECK DRIVE STATUS WORD #1 DURING A SEEK OR AFTER OTHER FUNCTIONS ARE DONE
/CALLED BY: STACHK
/RETURN CALL+2 IF COVER OPEN, SPARE BIT SET, DRIVE READY AND STATE NOT=5
/DRIVE NOT READY AND STATE NOT=4 OR 5, HEAD SELECT INCORRECT, HEADS BIT 7 NOT = 1,
/OR BRUSH HOME BIT 8 NOT = 1, CLEAR DRIVE BUSY AFTER A SEEK IF DRIVE READY
/RETURN CALL+1 IF ALL OF ABOVE OK
/
2200 0000 CHKSTA, 0
2201 7300 CLA CLL
2202 6000 SKON /SKIP IF INTERRUPT ON AND TURN OFF
2203 7301 CLA CLL IAC /SET BIT 11, INTERRUPT OFF
2204 1110 TAD KION /ADD 6001
2205 3253 DCA CKSOUT /SAVE ION OR IOF FOR EXIT
2206 1026 TAD STAT6A /GET STATUS WRD #1
2207 0377 AND (40 /MASK COVER OPEN BIT 6//REMOVE BIT MASK 200 HP 004
2210 7640 SZA CLA /SKIP IF BOTH CLEAR
2211 5252 JMP CKSOUT-1 /COVER OPEN OR SPARE BIT SET
/
2212 4536 HDCHKA, SETPNT /SET TBLPNT TO CYL ADDR
2213 0002 NEWCYL /TBL INDX
2214 1504 TAD I TBLPNT /GET CYL ADDR AND HEAD SELECT
2215 0051 AND K2000 /MASK HEAD SELECT BIT
2216 7112 CLL RTR
2217 7012 RTR /MOVE HEAD SELECT TO BIT 5
2220 3255 DCA HDTMP /SAVE HEAD SELECT TEMPORARILY

```

```

2221 1026      TAD      STAT6A      /GET STATUS WORD #1
2222 0043      AND      K100        /MASK HEAD SELECT BIT 5
2223 7041      CIA      /NEGATE
2224 1255      TAD      HDTMP        /ADD HEAD SELECT EXPECTED
2225 7640      SZA CLA      /SKIP IF EXPECTED HEAD=HEAD FROM GET STATUS
2226 5252      JMP      CKSOUT-1     /HEAD SELECT INCORRECT
/
2227 1030      DRDYA, TAD      ERREG      /GET ERROR REGISTER
2230 7110      CLL RAR      /MOVE DRIVE READY BIT INTO LINK
2231 7630      SZL CLA      /SKIP IF NOT READY
2232 5240      JMP      STATES5      /GO CHECK FOR STATE 5 (TRACKING)
/
2233 1026      STATE4, TAD      STAT6A      /GET STATUS WRD #1
2234 0042      AND      K7          /MASK STATE BITS 9:11
2235 1053      TAD      M4          /SUBTRACT 4
2236 7650      SNA CLA      /SKIP TO CHECK STATE 5
2237 5245      JMP      BHOME       /GO CHECK FOR BRUSHES HOME AND HEADS OUT
/
2240 1026      STATES5, TAD      STAT6A      /GET STATUS WRD #1
2241 0042      AND      K7          /MASK STATE BITS
2242 1376      TAD      (-5         /SUBTRACT 5
2243 7640      SZA CLA      /SKIP IF STATE 5 (TRACKING)
2244 5252      JMP      CKSOUT-1     / (NOT STATE 5 AND READY) OR (NOT STATE 5 OR 4)
/
2245 1026      BHOME, TAD      STAT6A      /GET STATUS WRD #1
2246 0375      AND      (30         /MASK BITS 7 (HDS OUT) AND 8 (BRUSH HOME)
2247 7041      CIA      /NEGATE
2250 1375      TAD      (30         /ADD BITS 7 AND 8 =1
2251 7640      SZA CLA      /SKIP IF BRUSHES HOME AND HEADS OVER DISK
2252 2200      ISZ      CHKSTA      /INCREMENT FOR BAD RETURN
2253 7402      CKSOUT, HLT/ION/IOF /MODIFIED ION OR IOF
2254 5600      STAEX2, JMP I  CHKSTA /RETURN
/
2255 0000      HDTMP, 0          /TEMPORARY STORAGE OF EXPECTED HEAD IN BIT 5
/
/*****
/
/ROUTINE TO WAIT FOR CONTROLLER NOT BUSY
/
2256 0000      DNWAIT, 0
2257 7300      CLA CLL
2260 1374      TAD      (-24        /20 TIMES 31.2 MSEC. = 624 MSEC. APPROX.
2261 3276      DCA      DNCNT1      /SET UP MULTIPLIER
2262 3275      DCA      DONCNT      /CLEAR WAIT FOR INTERRUPT COUNTER
2263 1055      DONLP, TAD      CBUSY      /GET CONTROLLER BUSY FLAG
2264 7650      SNA CLA      /SKIP IF CONTROLLER BUSY
2265 5273      JMP      NOTBSY      /CONTROLLER NOT BUSY
2266 2275      ISZ      DONCNT      /INCREMENT WAIT COUNTER
2267 5263      JMP      DONLP       /CONTINUE WAITING FOR DONE OR ERROR
2270 2276      ISZ      DNCNT1      /INCREMENT MULTIPLIER
2271 5263      JMP      DONLP       /CONTINUE WAITING FOR DONE, COMP. ERROR
/
2272 5656      JMP I  DNWAIT      /FATAL ERROR - NO INTERRUPT
2273 2256      NOTBSY, ISZ      DNWAIT /INCREMENT FOR GOOD RETURN

```

```

2274 5656      JMP I   DNWAIT      /GOOD RETURN
/
2275 0000      DONCNT, 0      /INTERRUPT WAIT COUNTER
2276 0000      DNCNT1, 0      /INTERRUPT WAIT MULTIPLIER
/
/
/*****
/
/ROUTINE TO SET UP "TBLPNT" (TABLE POINTER) TO
/POINT TO CORRECT DRIVE STATE TABLE ENTRY
/CALL+1 = TABLE OFFSET DESIGNATOR
/LOCATION "CURDRV" MUST CONTAIN CURRENT DRIVE NUMBER IN BITS 9:11
/
/      EXAMPLE:      SETPNT      (SUBROUTINE CALL)
/                   NEWCYL      (TABLE OFFSET DESIGNATOR)
/
/
2277 0000      XSETP, 0
2300 6000      SKON
2301 5306      JMP      INTOFF      /SKIP IF INTERRUPT ON AND TURN OFF
2302 3331      DCA      ACSV      /INTERRUPT NOT ON
2303 1110      TAD      KION      /SAVE AC
2304 3327      DCA      XSETEX      /GET ION INSTRUCTION
2305 5311      JMP      LSAV      /SAVE FOR EXIT EXECUTION
2306 3331      INTOFF, DCA      LSAV      /GO SAVE LINK
2307 1373      TAD      ACSV      /SAVE AC UPON ENTRY
2310 3327      DCA      (IOF      /GET IOF INSTRUCTION
2311 7004      LSAV, DCA      XSETEX      /SAVE FOR EXIT EXECUTION
2312 3332      RAL      /SAVE LINK
2313 1372      DCA      LKSV      /GET LINK UPON ENTRY
2314 1024      TAD      (DRVPNT      /GET ADDRESS OF INDEX TABLE TO DRIVE STATE TABLE
2315 3104      TAD      CURDRV      /ADD CURRENT DRIVE # AS INDEX INTO TABLE
2316 1504      DCA      TBLPNT      /SAVE TEMPORARILY
2317 3104      TAD I   TBLPNT      /GET POINTER FROM INDEX TABLE
2320 1677      DCA      TBLPNT      /SAVE TEMPORARILY
2321 1104      TAD I   XSETP      /GET TABLE OFFSET DESIGNATOR FROM CALL+1
2322 3104      TAD      TBLPNT      /ADD INDEX TABLE POINTER
2323 2277      DCA      TBLPNT      /SAVE DRIVE STATE TABLE POINTER
2324 1332      ISZ      XSETP      /INCREMENT RETURN ADDRESS
2325 7110      TAD      LKSV      /GET LINK UPON ENTRY
2326 1331      CLL RAR      /RESTORE LINK
2327 7402      TAD      ACSV      /GET AC UPON ENTRY
2330 5677      XSETEX, HLT/ION/IOF      /MODIFIED ION OR IOF DEPENDING ON INT STATE UPON ENTRY
                JMP I   XSETP      /EXIT
/
/
2331 0000      ACSV, 0      /AC SAVE AREA
2332 0000      LKSV, 0      /LINK SAVE AREA
/
/
/
/

```

/ROUTINE TO CHECK FOR DRIVE READY IN ERROR REGISTER
/RETURN CALL+1 IF READY
/RETURN CALL+2 IF NOT READY

 CALLED BY: JMS RDYCHK

2333	0000	RDYCHK, 0	
2334	6610	IOTG, RRER	/READ ERROR REGISTER INSTRUCTION
2335	7110	CLL RAR	/PUT DRIVE READY BIT 11 IN LINK
2336	7420	SNL	/SKIP IF DRIVE READY - RETURN CALL+1
2337	2333	ISZ RDYCHK	/DRIVE IS NOT READY - RETURN CALL+2
2340	5733	JMP I RDYCHK	/EXIT

/ROUTINE TO CALL STATUS PRINTER IN FLD 1

 CALLED BY: STAPRT
 FOLLOWED BY: ERTBL1 (EXAMPLE ERROR MESSAGE TABLE ADDRESS)

2341	0000	PRTSTA, 0	
2342	7300	CLA CLL	
2343	1741	TAD I PRTSTA	/GET CALL+1, START OF MESSAGE ADDRESS
2344	3347	DCA STATGO	/SAVE FOR FLD 1 CALL
2345	6212	CIF 10	/GOING TO FLD 1
2346	4771	JMS I (PRNSTA	/GO TO STATUS PRINTER
2347	7402	STATGO, HLT/MSG ADRS	/START OF MESSAGE ADDRESS
2350	2341	ISZ PRTSTA	/INCREMENT FOR RETURN
2351	5741	JMP I PRTSTA	/RETURN

2371 1641
2372 7400
2373 6002
2374 7754
2375 0030
2376 7773
2377 0040
2400

PAGE

```

/
/ROUTINE TO DO A SEEK TO AN ABSOLUTE CYLINDER ADDRESS
/AND HEAD SELECT
/CALL+1=CYL ADDR IN BITS 4:11, HEAD SELECT IN BIT 1
/CALLED BY: SETPOS
/FOLLOWED BY: NNNN          /NNNN=HEAD SELECT IN BIT 1.CYLINDER ADDRESS IN 4:11
/RETURN CALL+2 IF UNSUCCESSFUL (DRIVE DROPPED)
/RETURN CALL+3 IF SUCCESSFUL
/

```

2400	0000	POSSET, 0	
2401	4560	RESET	/RESET IGNORING VOLUME CHECK
2402	4560	RESET	/RESET THE DRIVE
2403	4542	ERRCHK	/CHECK FOR ERRORS
2404	5207	JMP .+3	/NO ERROR RETURN
2405	4541	ERROR	/STATUS ERROR OR CONTROLLER ERROR
2406	5352	JMP BADEX2	/DRIVE DROPPED RETURN - EXIT
2407	4777	JMS RDYCHK	/CHECK FOR DRIVE READY
2410	5213	JMP .+3	/NO ERROR RETURN
2411	4541	ERROR	/DRIVE NOT READY AFTER RESET
2412	5352	JMP BADEX2	/DRIVE DROPPED RETURN - EXIT
2413	4547	GETSTA	/GET DRIVE STATUS
2414	4542	ERRCHK	/CHECK FOR ERRORS
2415	5220	JMP .+3	/NO ERROR RETURN
2416	4541	ERROR	/STATUS ERROR OR CONTROLLER ERROR
2417	5352	JMP BADEX2	/DRIVE DROPPED RETURN - EXIT
2420	4777	JMS RDYCHK	/CHECK DRIVE READY
2421	5224	JMP .+3	/NO ERROR RETURN
2422	4541	ERROR	/DRIVE NOT READY
2423	5352	JMP BADEX2	/DRIVE DROPPED RETURN - EXIT
2424	4776	JMS RDHDR	/READ HEADER
2425	4542	ERRCHK	/CHECK FOR ERRORS
2426	5231	JMP .+3	/NO ERROR RETURN
2427	4541	ERROR	/STATUS ERROR OR CONTROLLER ERROR
2430	5352	JMP BADEX2	/DRIVE DROPPED RETURN - EXIT
2431	1600	TAD I POSSET	/GET NEW CYL AND HEAD FROM CALL+1
2432	0047	AND K0777	/MASK OFF HEAD SLCT BIT 1
2433	4536	SETPNT	/SET TBLPNT TO COMMAND A
2434	0022	XCOMA	/TBL INDX
2435	3504	DCA I TBLPNT	/SAVE NEW CYL - HD SLCT
2436	4536	SETPNT	/SET TBLPNT TO CURRENT CYL AND HD
2437	0001	CURCYL	/TBL INDX
2440	1504	TAD I TBLPNT	/GET CURRENT CYL AND HD
2441	0047	AND K0777	/MASK OFF HD SELECT
2442	7041	CIA	/NEGATE
2443	4536	SETPNT	/SET TBLPNT TO COMMAND A
2444	0022	XCOMA	/TBL INDX
2445	1504	TAD I TBLPNT	/SUBTRACT CURRENT CYL ADDR FROM NEW CYL ADDR
2446	7500	SMA	/SKIP IF NEW CYL < CURRENT CYL
2447	5252	JMP .+3	/NEW CYL > OR = CURRENT CYL
2450	7041	CIA	/MAKE CYL DIF POSITIVE
2451	7410	SKP	
2452	1375	TAD (4000	/SET HEAD DIRECTION FOR INWARD (HIGHER CYL)
2453	3504	DCA I TBLPNT	/SAVE COMMAND A CYL DIF WRD
2454	1600	TAD I POSSET	/GET NEWCYL AND HEAD FROM CALL+1
2455	0051	AND K2000	/MASK HEAD BIT

HP 001

HP 001

2456	1504	TAD I	TBLPNT	/ADD CYL DIF WRD AND DIRECTION	
2457	3504	DCA I	TBLPNT	/SAVE CYL DIF WRD,DIRECTION,HD SLCT	
2460	4536	SETPNT		/SET TBLPNT TO NEW CYL AND HD	
2461	0002	NEWCYL		/TBL INDX	
2462	1600	TAD I	POSSET	/GET NEW CYL AND HD FROM CALL+1	
2463	3504	DCA I	TBLPNT	/SAVE NEW CYL AND HD SLCT	
2464	4777	JMS	RDYCHK	/CHECK FOR DRIVE READY	
2465	5270	JMP	+.3	/DRIVE READY RETURN	
2466	4541	ERROR		/DRIVE NOT READY	
2467	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2470	1504	TAD I	TBLPNT	/GET THE NEW HEAD AND CYLINDER	HP 002
2471	7421	MQL		/ AND DISPLAY IT IN THE MQ	HP 002
2472	4774	JMS	SEEK	/SEEK TO NEW CYL AND HEAD	
2473	4544	WAITDN		/WAIT FOR DONE	
2474	4773	JMS	FATAL	/NO DONE INTERRUPT	
2475	4542	ERRCHK		/CHECK FOR ERRORS	
2476	5301	JMP	+.3	/NO ERROR RETURN	
2477	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
2500	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2501	4772	JMS	WATRDY	/WAIT FOR DRIVE READY	
2502	5305	JMP	+.3	/NO ERROR RETURN	
2503	4541	ERROR		/DRIVE NOT READY TIME-OUT	
2504	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2505	4547	GETSTA		/CHECK DRIVE STATUS	
2506	4542	ERRCHK		/CHECK FOR ERRORS	
2507	5312	JMP	+.3	/NO ERROR RETURN	
2510	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
2511	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2512	4550	STACHK		/CHECK DRIVE STATUS WORD #1	
2513	5316	JMP	+.3	/NO ERROR	
2514	4541	ERROR		/DRIVE STATUS ERROR WORD #1	
2515	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2516	4777	JMS	RDYCHK	/CHECK FOR DRIVE READY	
2517	5322	JMP	+.3	/DRIVE READY RETURN	
2520	4541	ERROR		/DRIVE NOT READY	
2521	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2522	4776	JMS	RDHDR	/READ HEADER	
2523	4542	ERRCHK		/CHECK FOR ERRORS	
2524	5327	JMP	+.3	/NO ERROR RETURN	
2525	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
2526	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2527	4547	GETSTA		/GET STATUS	
2530	4542	ERRCHK		/CHECK FOR ERRORS	
2531	5334	JMP	+.3	/NO ERROR RETURN	
2532	4541	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
2533	5352	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
2534	4536	SETPNT		/SET TBLPNT TO CURRENT CYL AND HD	
2535	0001	CURCYL		/TBL INDX	
2536	1504	TAD I	TBLPNT	/GET CURRENT CYL AND HD FROM READ HEADER	
2537	7041	CIA		/NEGATE	
2540	4536	SETPNT		/SET TBLPNT TO EXPECTED CYL AND HD	
2541	0002	NEWCYL		/TBL INDX	
2542	1504	TAD I	TBLPNT	/ADD EXPECTED CYL AND HD TO CURRENT CYL AND HD	
2543	7650	SNA CLA		/SKIP IF EXPECTED CYL AND HD NOT = CURRENT CYL AND HD	
2544	5351	JMP	SOK	/NO SEEK ERROR	

```

2545 4771 JMS LOGERR /LOG A SEEK ERROR
2546 0006 SEKERR /DRV TBL INDX
2547 4541 ERROR /SEEK ERROR, GO TO ERROR HANDLER
2550 5352 JMP BADEX2 /GO TO DRIVE DROPPED EXIT
2551 2200 SOK, ISZ POSSET /INCREMENT FOR SUCCESSFUL RETURN
2552 2200 BADEX2, ISZ POSSET /INCREMENT FOR GOOD OR BAD RETURN
2553 5600 JMP I POSSET /RETURN
/
/
/*****
/
/ROUTINE TO PRINT 2 SPACES
/
/CALLED BY: SPACE2
/
2554 0000 SPACES, 0
2555 7300 CLA CLL
2556 1363 TAD K240
2557 4553 PRNT
2560 1363 TAD K240
2561 4553 PRNT
2562 5754 JMP I SPACES /RETURN
2563 0240 K240, 0240
/
/
/
2571 3612
2572 7204
2573 7336
2574 7134
2575 4000
2576 3200
2577 2333
2600 PAGE
/
/
/
/*****
/
/ROUTINE TO CHECK FOR ERRORS AFTER INTERRUPT
/
/CALLED BY: ERRCHK
/
/RETURN CALL+2 IF COMPOSITE ERROR FLAG SET
/RETURN CALL+1 IF NO ERRORS
/ELSE GO TO FATAL ERROR ROUTINE IF ERROR BITS SET
/AND NO COMPOSITE ERROR FLAG, OR IF A CONTROLLER
/REGISTER LOADED IMPROPERLY
/
2600 0000 CHKERR, 0

```

2601	7300	CLA	CLL		
2602	6000	SKON			/SKIP IF INTERRUPT ON AND TURN OFF
2603	7301	CLA	CLL	IAC	/SET BIT 11 FOR IOF
2604	1110	TAD		KION	/ADD 6001
2605	3273	DCA		CHKOUT	/SAVE ION OR IOF FOR EXECUTION
2606	1112	TAD		FLGSAV	/GET FLAGS FROM LAST INTERRUPT
2607	7110	CLL	RAR		/MOVE COMPOSITE ERROR INTO LINK
2610	7630	SZL	CLA		/SKIP IF COMPOSITE ERROR NOT SET
2611	5272	JNP		EREX1	/COMPOSITE ERROR FLG SET
2612	1030	TAD		ERREG	/GET ERROR REGISTER
2613	0377	AND		(7002	/MASK ERROR BITS AND DRIVE ERROR BIT
2614	7640	SZA	CLA		/SKIP IF NO ERRORS
2615	5272	JMP		EREX1	/CONTROLLER ERROR BITS SET, COMPOSITE ERROR FLAG
					/MAY HAVE BEEN CLEARED BY GET STATUS
2616	1027	TAD		STAT6B	/GET DRIVE STATUS WORD #2
2617	7650	SNA	CLA		/SKIP IF DRIVE ERROR BITS NOT CLEAR
2620	5223	JMP		.,+3	/ERROR BITS CLEAR
2621	4776'	JMS		WRTLOC	/CHECK FOR WRITE LOCK ON DRIVE
2622	4775'	JMS		FATAL	/DRIVE ERROR BITS SET BUT NOT COMPOSITE ERROR FLG
2623	4536	SETPNT			/SET TBLPNT TO EXPECTED (SENT) COMMAND A
2624	0022	XCOMA			/TABLE INDEX
2625	1504	TAD	I	TBLPNT	/GET COMMAND A SENT
2626	7041	CIA			/NEGATE
2627	1031	TAD		COMDA	/ADD COMMAND A READ
2630	7640	SZA	CLA		/SKIP IF COMMAND A SENT = COMMAND A READ
2631	4775'	JMS		FATAL	/ERROR LOADING COMMAND A REGISTER
2632	4536	SETPNT			/SET TBLPNT TO EXPECTED (SENT) COMMAND B
2633	0023	XCOMB			/TABLE INDEX
2634	1504	TAD	I	TBLPNT	/GET COMMAND B SENT
2635	7041	CIA			/NEGATE
2636	1032	TAD		COMDB	/ADD COMMAND B READ
2637	7640	SZA	CLA		/SKIP IF COMMAND B SENT = COMMAND B READ
2640	4775'	JMS		FATAL	/ERROR LOADING COMMAND B REGISTER
2641	1504	TAD	I	TBLPNT	/GET COMMAND B SENT
2642	0042	AND		K7	/MASK FUNCTION CODE BITS
2643	1374	TAD		(-5	/SUBTRACT 5
2644	7700	SNA	CLA		/SKIP IF FUNCTION CODE < 5
2645	5256	JMP		ZWRD	/DATA TRANSFER, EXPECT ZERO FINAL WORD COUNT
2646	4536	SETPNT			/SET TBLPNT TO WORD COUNT LOADED
2647	0017	WRDCNT			/TBL INDX
2650	1504	TAD	I	TBLPNT	/GET WORD COUNT LOADED
2651	7041	CIA			/NEGATE FOR CHECKING
2652	1034	TAD		ENDWD	/ADD FINAL WORD COUNT READ
2653	7640	SZA	CLA		/SKIP IF FINAL WORD COUNT READ = WORD COUNT LOADED
2654	4775'	JMS		FATAL	/ERROR LOADING WORD COUNT REGISTER
2655	5261	JMP		FINSEC	/GO CHECK FINAL SECTOR REGISTER CONTENTS
2656	1034	ZWRD,	TAD	ENDWD	/GET FINAL WORD COUNT READ
2657	7640	SZA	CLA		/SKIP IF = ZERO
2660	4775'	JMS		FATAL	/FINAL WORD COUNT NOT 0 AND COMPOSITE ERROR FLAG NOT SET
2661	4536	FINSEC,	SETPNT		/SET TBLPNT TO FINAL SECTOR EXPECTED
2662	0024	XENDSC			/TABLE INDEX
2663	1504	TAD	I	TBLPNT	/GET FINAL SECTOR EXPECTED
2664	7002	BSW			/SWAP TO OTHER BYTE
2665	7041	CIA			/NEGATE FOR CHECKING
2666	1033	TAD		ENDSC	/ADD FINAL SECTOR READ


```

2667 7640          SZA CLA          /SKIP IF FINAL SECTOR READ = FINAL SECTOR EXPECTED
2670 4775'        JMS      FATAL    /SECTOR ADDRESS REGISTER ERROR
2671 5273          JMP      CHKOUT   /GO TO GOOD RETURN

/
2672 2200        EREX1, ISZ      CHKERR /INCREMENT FOR ERROR RETURN
2673 7402        CHKOUT, HLT/ION/IOF /MODIFIED ION OR IOF
2674 5600          JMP I      CHKERR   /RETURN

/
/
/*****
/ROUTINE TO GENERATE DATA FOR DATA BUFFER
/
/CALLED BY: JMS DATGEN
/
/
2675 0000        DATGEN, 0
2676 7300          CLA      CLL
2677 1373          TAD      (-200      /128 X 2 WORDS PER PATTERN
2700 3037          DCA      WDCNTR    /SET UP WORD COUNTER
2701 1106          TAD      BUFAD1    /GET ADDRESS OF BUFFER
2702 1052          TAD      M1        /-1 FOR AUTO INDEX
2703 3017          DCA      AUTO17    /SET AUTO INDEX POINTER TO BUFFER
2704 1035          TAD      CURUNT    /GET CURRENT UNIT #
2705 1372          TAD      (DPAT8    /INDEX INTO PATTERN TABLE
2706 3120          DCA      PATMP     /SAVE ADDRESS OF PATTERN TABLE
2707 1520          TAD I    PATMP     /GET ADDRESS OF DATA PATTERN
2710 3120          DCA      PATMP     /SAVE ADDRESS OF DATA PATTERN
2711 1520          TAD I    PATMP     /GET FIRST DATA PATTERN WORD
2712 3327          DCA      DWD1      /SAVE IT
2713 2120          ISZ      PATMP     /INCREMENT DATA PATTERN POINTER
2714 1520          TAD I    PATMP     /GET SECOND DATA PATTERN WORD
2715 3330          DCA      DWD2      /SAVE IT
2716 6211          CDF      10       /CDF TO BUFFER FLD 1

/
2717 1327        DATLP1, TAD      DWD1 /GET FIRST PATTERN WORD
2720 3417          DCA I    AUTO17    /LOAD IN DATA BUFFER
2721 1330          TAD      DWD2      /GET SECOND PATTERN WORD
2722 3417          DCA I    AUTO17    /LOAD IN DATA BUFFER
2723 2037          ISZ      WDCNTR    /INCREMENT WORD COUNT
2724 5317          JMP      DATLP1    /CONTINUE LOADING BUFFER
2725 6201          CDF      00       /DF BACK TO 0
2726 5675          JMP I    DATGEN    /RETURN

/
2727 0000          DWD1,  0          /DATA PATTERN WORD # 1
2730 0000          DWD2,  0          /DATA PATTERN WORD # 2

/
/
/
2731 5675        DATEXA, JMP I    DATGEN /RETURN

/
/*****

```

/
 / DATA PATTERN INDEX TABLE
 /

	2732	DPAT8=.	
2732	2736	DPAT80, PAT80	/DATA PATTERN 0 ADDRESS (8 BIT MODE)
2733	2740	DPAT81, PAT81	/DATA PATTERN 1 ADDRESS (8 BIT MODE)
2734	2742	DPAT82, PAT82	/DATA PATTERN 2 ADDRESS (8 BIT MODE)
2735	2744	DPAT83, PAT83	/DATA PATTERN 3 ADDRESS (8 BIT MODE)

 /
 / DATA PATTERN TABLE
 /

2736	0325	PAT80, 0325	/152525 (16 BIT)
2737	0125	0125	
2740	0266	PAT81, 0266	/133333 (16 BIT)
2741	0333	0333	
2742	0155	PAT82, 0155	/066666 (16 BIT)
2743	0266	0266	
2744	0333	PAT83, 0333	/155555 (16 BIT)
2745	0155	0155	

 /
 /
 /*****
 /

2772	2732
2773	7600
2774	7773
2775	7336
2776	1314
2777	7002
	3000

 /
 /
 /
 / PAGE
 /

 /*****
 /

 /
 / ROUTINE TO EXECUTE A COMMAND TO A DRIVE
 /

 /
 / CALLED BY: GO
 /

3000	0000
3001	6002

 /
 / XGO, 0
 / 10F

 / INHIBIT INTERRUPTS
 /

```

3002 1024      TAD      CURDRV      /GET CURRENT DRIVE
3003 3025      DCA      DRV60      /SAVE AS DRIVE USING CONTROLLER
3004 7240      CLA      CMA
3005 3055      DCA      CBUSY      /SET CONTROLLER BUSY FLG
3006 4536      SETPNT
3007 0017      WRDCNT
3010 1504      TAD I      TBLPNT      /SET "TBLPNT" TO WORD COUNT IN DRIVE STATE TABLE
3011 6607      CNTLOD, RLWC      /GET WORD COUNT FROM DRIVE STATE TABLE
3012 7440      SZA      /LOAD WORD COUNT IOT
3013 4777'      JMS      FATAL      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
3014 4536      SETPNT      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
3015 0020      INITCA
3016 1504      TAD I      TBLPNT      /SET TBLPNT TO BREAK MEM ADDR
3017 6602      BRKLOD, RLMA      /GET BREAK MEM ADDR FROM DRIVE STATE TBL
3020 7440      SZA      /LOAD BREAK MEM ADDR IOT
3021 4777'      JMS      FATAL      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
3022 4536      SETPNT      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
3023 0021      SECADD
3024 1504      TAD I      TBLPNT      /SET TBLPNT TO SECTOR ADDRESS
3025 7002      BSW
3026 6605      SECLOD, RLSA      /GET SECTOR ADDRESS
3027 7440      SZA      /MOVE TO BITS 0:5 FROM 6:11
3030 4777'      JMS      FATAL      /LOAD OR CLEAR SECTOR ADDRESS IOT
3031 4536      SETPNT      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
3032 0022      XCOMA      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
3033 1504      TAD I      TBLPNT      /SET TBLPNT TO COMMAND A
3034 6603      ALODE, RLCA      /GET COMMAND A
3035 7440      SZA      /LOAD OR CLEAR COMMAND A
3036 4777'      JMS      FATAL      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
3037 4536      SETPNT      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
3040 0023      XCOMB
3041 1504      TAD I      TBLPNT      /SET TBLPNT TO COMMAND B
3042 6604      BLODE, RLCB      /GET COMMAND B
3043 7440      SZA      /LOAD COMMAND B, EXECUTE FUNCTION IOT
3044 4777'      JMS      FATAL      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
3045 6001      GOEXIT, ION      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
3046 5600      JMP I      XGO      /ENABLE INTERRUPTS
                                   /EXIT
                                   /
                                   /
                                   /*****
                                   /
                                   /
                                   /ROUTINE TO CHECK FOR DRIVE WAS READY AT TIME OF LAST INTERRUPT
                                   /
                                   /      CALLED BY:      JMS      CHKRDY
                                   /
                                   /RETURN CALL+1 IF NOT READY
                                   /RETURN CALL+2 IF READY
                                   /
                                   /
3047 0000      CHKRDY, 0
3050 7300      CLA      CLL

```

```

3051 1030      TAD      ERREG      /GET ERROR REGISTER AT TIME OF ERROR
3052 7110      CLL      RAR        /MOVE INTO LINK BIT 11 (DRIVE READY)
3053 7630      SZL      CLA        /SKIP IF DRIVE WAS NOT READY
3054 2247      ISZ      CHKRDY     /DRIVE WAS READY, INCREMENT RETURN
3055 5647      JMP I    CHKRDY     /RETURN

/
/
/
/*****
/
/
/
/ROUTINE TO GET STATUS
/CALLED BY:  GETSTA
/RETURN ONLY IF SUCCESSFUL
/
/
3056 0000      STAGET, 0
3057 7300      CLA      CLL
3060 6035      KIE

/DISABLE KEYBOARD INTERRUPTS TO PREVENT
/AN UNTIMELY INTERRUPT FROM ALLOWING INTERRUPT
/SERVICE TO CLEAR THE STATUS SAVE AREA OR
/DRIVE READY TO COME BACK WHILE IN INTERRUPT
/SERVICE HANDLING KEYBOARD INTERRUPT. THIS IS
/VITAL TO AVOID STATUS ERRORS.
/INTERRUPTS OFF
/SET TBLPNT TO COMMAND A
/TABLE INDEX
/CLEAR COMMAND A
/SET TBLPNT TO SECTOR ADDRESS
/TABLE INDEX
/CLEAR SECTOR ADDRESS
/SET TBLPNT TO EXPECTED FINAL SECTOR ADDRESS
/TBL INDX
/EXPECTED FINAL SECTOR ADDRESS = 0
/SET TBLPNT TO COMMAND B
/TABLE INDEX
/GET CURRENT DRIVE NUMBER
/MOVE DRIVE SELECT TO BITS 4,5
/SET INTERRUPT ENABLE, 8 BIT MODE, CODE 2
/STORE COMMAND B
/EXECUTE COMMAND
/WAIT FOR DONE OR ERROR
/DONE DID NOT SET

/
3104 6002      IOF
3105 6615      RDSIA, RRSI
3106 7410      SKP
3107 4777'     JMS      FATAL
3110 0045      AND      K377
3111 3026      DCA      STAT6A
3112 6615      RDSIB, RRSI
3113 7410      SKP
3114 4777'     JMS      FATAL

/READ SILO
/IOT SHOULD NOT SKIP
/IOT SKIPPED - FATAL ERROR
/MASK OFF GARBAGE BITS
/SAVE STATUS WORD #1
/READ SILO
/IOT SHOULD NOT SKIP
/IOT SKIPPED - FATAL ERROR

```

```

3115 0045      AND      K377      /MASK OFF GARBAGE BITS
3116 3027      DCA      STAT6B    /SAVE STATUS WORD #2
3117 5656      JMP I    STAGET    /EXIT

```

```

/*****
/
/

```

```

/ROUTINE TO DO A DATA TRANSFER
/READ,WRITE
/CALLED BY:      XFER
/FUNCTION CODE MUST BE IN LOCATION "FUNCOD" AND MUST BE
/0005 (WRITE), 0006 (READ)
/
/

```

```

3120 0000      EXFER, 0
3121 7300      CLA      CLL
3122 6002      IOF
3123 4536      SETPNT
3124 0001      CURCYL
3125 1504      TAD I    TBLPNT
3126 4536      SETPNT
3127 0022      XCOMA
3130 3504      DCA I    TBLPNT
3131 4536      SETPNT
3132 0023      XCOMB
3133 1024      TAD      CURDRV
3134 7002      BSW
3135 3344      DCA      DTMP1
3136 1346      TAD      K1410
3137 1344      TAD      DTMP1
3140 1105      TAD      FUNCOD
3141 3504      DCA I    TBLPNT
3142 4543      GO
3143 5720      JMP I    EXFER

/INTERRUPTS OFF
/SET TBLPNT TO CURRENT CYL AND HD
/TABLE INDEX
/GET CURRENT CYL AND HD
/SET TBLPNT TO COMMAND A
/TABLE INDEX
/SAVE COMMAND A
/SET TBLPNT TO COMMAND B
/TABLE INDEX
/GET CURRENT DRIVE
/MOVE TO BITS 4,5
/SAVE SHIFTED DRIVE NUMBER
/BUFFER FLD 1, 8 BIT MODE, INTERRUPT ENABLE
/ADD DRIVE NUMBER
/ADD FUNCTION CODE
/SAVE COMMAND B
/ISSUE COMMAND
/EXIT

```

```

3144 0000      DTMP1, 0
3145 0000      FLDP1, 0
3146 1410      K1410, 1410

/TEMP STORAGE OF SHIFTED DRIVE NUMBER
/TEMP STORAGE OF BUFFER FIELD POINTER

```

```

3176 1402
3177 7336
3200
PAGE
/
/
/
/
/

```

```

/
/
/*****
/
/ROUTINE TO READ HEADER
/CALLED BY: JMS RDHDR
/RETURN ONLY IF SUCCESSFUL
/
/
RDHDR, 0
3200 0000
3201 7300 CLA CLL
3202 6002 IOF /INTERRUPTS OFF
3203 4536 SETPNT /SET TBLPNT TO COMMAND B
3204 0023 XCOMB /TABLE INDEX
3205 1024 TAD CURDRV /GET CURRENT DRIVE
3206 7002 BSW /MOVE DRIVE SELECT TO BITS 4,5
3207 1377 TAD (1404 /ADD INTERRUPT ENABLE, 8 BIT MODE, CODE 4
3210 3504 DCA I TBLPNT /STORE COMMAND B
3211 4543 GO /EXECUTE COMMAND
3212 4544 WAITDN /WAIT FOR DONE OR ERROR
3213 4776 JMS FATAL /DONE DID NOT SET

/
RDSI1, RRSI /READ SILO WORD #1, SECTOR ADDR, HEAD SELECT, AND CYL ADDR LSB
3214 6615 AND K377 /MASK OFF GARBAGE BITS
3215 0045 DCA SILO1 /SAVE SILO WORD #1
3216 3254

/
RDSI2, RRSI /READ SILO WORD #2, CYL ADDR
3217 6615 AND K377 /MASK OFF GARBAGE BITS
3220 0045 DCA SILO2 /SAVE SILO WORD #2
3221 3255

/
RDSI3, RRSI /READ SILO WORD #3, SHOULD BE ALL ZEROES
3222 6615 AND K377 /MASK OFF UNUSED BITS
3223 0045 SZA CLA /SKIP IF WORD #3 IS ZERO
3224 7640 JMS HDRERR /HEADER ERROR WORD #3 NOT ZERO
3225 4775

/
RDSI4, RRSI /READ SILO WORD #4, SHOULD BE ALL ZEROES
3226 6615 AND K377 /MASK OFF UNUSED BITS
3227 0045 SZA CLA /SKIP IF WORD #4 IS ZERO
3230 7640 JMS HDRERR /HEADER ERROR WORD #4 NOT ZERO
3231 4775

/
/
3232 1254 TAD SILO1 /GET SILO WORD #1
3233 7002 BSW /MOVE CYL ADDR LSB FROM BIT 4 TO BIT 10
3234 7012 RTR /MOVE CYL ADDR LSB FROM BIT 10 TO LINK
3235 7200 CLA /GET RID OF SECTOR ADDR BITS
3236 1255 TAD SILO2 /GET SILO WORD #2
3237 7004 RAL /PUT CYL ADDR LSB IN BIT 11
3240 0047 AND K0777 /MASK OFF UNUSED BITS
3241 4536 SETPNT /SET TBLPNT TO CURRENT CYL
3242 0001 CURCYL /TABLE INDEX
3243 3504 DCA I TBLPNT /SAVE CURRENT CYL
3244 7201 CLA IAC /SET BIT 11
3245 7002 BSW /SET BIT 5

```

HP 001

```

3246 0254      AND      SILO1      /MASK HEAD SELECT BIT
3247 7106      CLL RTL      /MOVE TO BIT 3
3250 7106      CLL RTL      /MOVE HEAD SELECT TO BIT 1
3251 1504      TAD I      TBLPNT    /ADD CURRENT CYL
3252 3504      DCA I      TBLPNT    /SAVE CURCYL AND HEAD SELECT IN CURCYL
3253 5600      JMP I      RDHDR      /RETURN

/
3254 0000      SILO1, 0      /SECTOR ADDR BITS 6:11,CYL ADDR LSB BIT 4, HEAD SELECT BIT 5
3255 0000      SILO2, 0      /CYL ADDR MSB=BIT 4, REST OF CYL ADDR IN BITS 5:11
/
/*****
/
/MAN ERROR HANDLER
/
/      CALLED BY:      ERROR
/      FOLLOWED BY:     DRIVE DROPPED RETURN
/      FOLLOWED BY:     GOOD RETURN
/
/
3256 0000      XERROR, 0
3257 6002      IOF          /INTERRUPTS OFF
3260 7300      CLA          CLL
3261 1774      TAD          ERRFLG   /GET ERROR FLAG TO SEE IF FIRST ERROR CALL
3262 7640      SZA          CLA       /SKIP IF FIRST ERROR CALL
3263 5773      JMP          RETRY     /NOT FIRST CALL, GO TO RETRY ROUTINE
3264 7340      CLA CLL CMA
3265 3774      DCA          ERRFLG   /SET ERROR FLAG FOR FUTURE ERROR CALLS
3266 1256      TAD          XERROR    /GET ERROR CALL RETURN PC
3267 3772      DCA          ERRPC     /SAVE FOR RETURN
3270 1771      TAD          POSSET    /GET SEEK ROUTINE RETURN PC IF HERE BECAUSE OF SEEK OR READ HEADER ERROR
3271 3770      DCA          POSPC     /SAVE RETURN PC FOR SEEK ROUTINE
3272 6212      CIF          10       /GOING TO FLD 1
3273 4767      JMS I      (SAVALL    /GO SAVE ALL REGISTERS AT TIME OF ERROR
3274 4556      DOCRLF
3275 4551      MESAG
3276 2722      ERRPCM
3277 1256      TAD          XERROR    /"ERROR PC:"
3300 1052      TAD          M1        /GET CALL PC+1
3301 4555      PRNAC
3302 4556      DOCRLF             /SUBTRACT 1
                                   /PRINT ERROR CALL PC
                                   /<CR LF>

/
3303 4540      SPRNT,      STAPRT     /PRINT DRIVE #
3304 2040      ERTBL4
3305 4540      STAPRT
3306 1736      ERTBL1           /START OF MSG ADDRESS
                                   /PRINT CONTROLLER REGISTERS
                                   /START OF MSG ADDRESS

/
3307 1030      DETST,      TAD          ERREG   /GET ERROR REGISTER
3310 7112      CLL          RTR          /MOVE DRIVE ERROR BIT 10 INTO LINK
3311 7620      SNL          CLA          /SKIP IF DRIVE ERROR SET
3312 5766      JMP          NDE          /NO DRIVE ERROR, GO HANDLE NON-DRV ERROR
3313 4547      GETSTA
3314 1030      TAD          ERREG   /GET ERROR REGISTER
3315 7106      CLL          RTL          /PUT OPI IN LINK
3316 7620      SNL          CLA          /SKIP IF OPI FROM GET STATUS, NO SYSTEM CLOCK

```

3317	5333	JMP	SCK1	/GO CHECK STATUS
3320	4765'	JMS	LOGERR	/LOG OPI ERROR
3321	0014	OPIERR		/TABLE INDX
/				
3322	4764'	DRVGN,	JMS	DELAY
3323	7774	-4		/DELAY FOR 4 SECONDS
3324	4547	GETSTA		/TIMER
3325	4765'	JMS	LOGERR	/GET STATUS FOR PRINTOUT
3326	0005	DRVERR		/LOG DRIVE ERROR
3327	4765'	JMS	LOGERR	/DRIVE STATUS TABLE INDEX
3330	0003	HRDERR		/LOG HARD ERROR
3331	4763'	JMS	FINRPT	/DRIVE STATE TABLE INDEX
3332	5762'	JMP	EREXT2	/GO FINISH REPORT
/				
3333	1027	SCK1,	TAD	STAT6B
3334	0361	AND	(37	/GET STATUS WORD #2
3335	7650	SNA	CLA	/CHECK FOR SKTO,SPIN,WGE,VOL CHK,OR DRV SLCT ERROR
3336	5760'	JMP	SCK2	/SKIP IF ONE OF THE ABOVE OR MORE
3337	1026	TAD	STAT6A	/NONE OF THE ABOVE, GO CHECK FOR OTHER ERRORS
3340	0042	AND	K7	/GET STATUS WORD #1
3341	7650	SNA	CLA	/MASK DRIVE STATE BITS
3342	5322	JMP	DRVGN	/SKIP IF NOT LOAD STATE (0)
3343	4560	RESET		/DRIVE IN LOAD STATE
3344	4764'	JMS	DELAY	/RESET THE DRIVE
3345	7774	-4		/DELAY
3346	4547	GETSTA		/4 SECS. (APPROX.)
3347	1027	TAD	STAT6B	/GET STATUS
3350	0361	AND	(37	/GET STATUS WORD #2
3351	7650	SNA	CLA	/TEST DRIVE ERROR BITS AGAIN
3352	5773'	JMP	RETRY	/SKIP IF ANY ERROR BITS SET
3353	5322	JMP	DRVGN	/GO TO RETRY ROUTINE
/				
/				
3360	3400			
3361	0037			
3362	3551			
3363	3600			
3364	1066			
3365	3612			
3366	3424			
3367	2044			
3370	3556			
3371	2400			
3372	3555			
3373	3530			
3374	3557			
3375	5107			
3376	7336			
3377	1404			
	3400			
PAGE				
/				
/				
/				
3400	1027	SCK2,	TAD	STAT6B
3401	0377	AND	(140	/GET STATUS WORD #2
				/CHECK FOR WRT PROT OR HEAD CURRENT ERROR

3402	7650	SNA	CLA	/SKIP IF EITHER OF ABOVE IS SET
3403	5205	JMP	SCK3	/GO CHECK FOR OTHER ERRORS
3404	5776'	JMP	DRVGON	/DROP DRIVE
/				
3405	1027	SCK3,	TAD	/GET STATUS WORD #2
3406	0375	AND	STAT6B	/CHECK FOR SPIN ERROR OR WDE (WRITE DATA ERROR)
3407	7650	SNA	CLA	/SKIP IF EITHER OF ABOVE
3410	5330	JMP	RETRY	/GO TO RETRY ROUTINE
3411	4774'	JMS	DELAY	
3412	7777	-1		/1 SEC. (APPROX.)
3413	4560	RESET		/RESET THE DRIVE
3414	4774'	JMS	DELAY	
3415	7730	-50		/40 SEC. (APPROX.)
3416	4547	GETSTA		/GET STATUS
3417	1026	TAD	STAT6A	/GET STATUS WORD #1
3420	0042	AND	K7	/MASK DRIVE STATE BITS
3421	7640	SZA	CLA	/SKIP IF LOAD STATE
3422	5330	JMP	RETRY	/GO TO RETRY ROUTINE
3423	5776'	JMP	DRVGON	/GO DROP DRIVE
/				
3424	1030	NDE,	TAD	/NO DRIVE ERROR
3425	0373	AND	ERREG	/MASK ERROR BITS
3426	7450	SNA	(7000	/SKIP IF ERROR BITS SET IN ERROR REGISTER
3427	5233	JMP	.+4	/NO ERROR BITS, MAYBE SOMEBODY PUSHED LOAD BUTTON
3430	1372	TAD	(6000	/ADD 2'S COMPLEMENT
3431	7640	SZA	CLA	/SKIP IF OPI ONLY
3432	5260	JMP	NOPION	/NOT OPI ONLY
3433	4771'	JMS	CHKRDY	/SEE IF DRIVE WAS READY
3434	5247	JMP	STATC	/DRIVE NOT READY
/				
3435	1117	ABORT,	TAD	/GET DATA ERROR FLAG (NO DCRQ, POSSIBLE WRITE ERROR)
3436	7640	SZA	CLA	/SKIP IF NO DATA ERROR
3437	5330	JMP	RETRY	/DATA ERROR, NO DCRQ, POSSIBLE WRITE ERROR, GO RETRY READ
3440	4547	GETSTA		/GET STATUS TO AVOID SOFT ERROR RETRIES
3441	4770'	JMS	LOGERR	/LOG SOFT ERROR
3442	0004	SFTERR		/DRIVE STATE TABLE INDEX
3443	4770'	JMS	LOGERR	/LOG CONTROLLER ERROR
3444	0016	CTLERR		
3445	4767'	JMS	FINRPT	/GO FINISH REPORT
3446	5351	JMP	EREXT2	/RETURN (ABORT OPERATION)
/				
3447	4547	STATC,	GETSTA	/DRIVE NOT READY, GET STATUS
3450	1026	TAD	STAT6A	/GET STATUS WORD #1
3451	0042	AND	K7	/MASK DRIVE STATE BITS
3452	1053	TAD	M4	/SUBTRACT 4
3453	7640	SZA	CLA	/SKIP IF STATE 4 (SEEK-TRACK COUNTING)
3454	5776'	JMP	DRVGON	/NOT STATE 4
3455	4774'	JMS	DELAY	/DELAY
3456	7774	-4		/WAIT 4 SECS. (APPROX.) FOR SKTO
3457	5766'	JMP	SCK1	/SEEK TIME-OUT EXPECTED
/				
3460	1030	NOPION,	TAD	/GET ERROR REGISTER
3461	0051	AND	K2000	/TEST FOR OPI
3462	7650	SNA	CLA	/SKIP IF OPI SET
3463	5311	JMP	NOTOPI	/OPI NOT SET

3464	1030	TAD	ERREG	/GET ERROR REGISTER
3465	0050	AND	K1000	/TEST FOR HEADER NOT FOUND (HNF)
3466	7650	SNA	CLA	/SKIP IF HNF SET
3467	5275	JMP	CHKHCR	/GO CHECK FOR HEADER CRC ERROR
3470	4771'	JMS	CHKRDY	/SEE IF DRIVE WAS READY
3471	5306	JMP	LOGTRK	/DRIVE NOT READY, LOG TRACKING ERROR
3472	4770'	LOGHNF, JMS	LOGERR	/DRIVE WAS READY
3473	0015	HNFFERR		/DRIVE STATE TABLE INDEX
3474	5330	JMP	RETRY	/GO TO RETRY ROUTINE
/				
3475	1030	CHKHCR, TAD	ERREG	/GET ERROR REGISTER
3476	0365	AND	(4000	/MASK HCRC BIT
3477	7650	SNA	CLA	/SKIP IF HCRC SET
3500	5235	JMP	ABORT	/ABORT OPERATION
3501	4771'	JMS	CHKRDY	/CHECK FOR DRIVE READY
3502	5306	JMP	LOGTRK	/DRIVE NOT READY
/				
3503	4770'	LOGHCR, JMS	LOGERR	/DRIVE READY, LOG HCRC ERROR
3504	0012	HRCRCR		
3505	5330	JMP	RETRY	/GO TO RETRY ROUTINE
/				
3506	4770'	LOGTRK, JMS	LOGERR	/LOG TRACKING ERROR
3507	0010	TRKERR		
3510	5330	JMP	RETRY	/GO TO RETRY ROUTINE
/				
3511	1030	NOTOPI, TAD	ERREG	/GET ERROR REGISTER
3512	0050	AND	K1000	/TEST DATA LATE BIT (DLT)
/				
3513	7650	CHKDLT, SNA	CLA	/SKIP IF DLT
3514	5320	JMP	CHKCRC	/NOT DLT, GO CHECK DATA CRC
/				
3515	4770'	LOGDLT, JMS	LOGERR	/LOG DATA LATE ERROR
3516	0013	DLTERR		
3517	5330	JMP	RETRY	/GO TO RETRY ROUTINE
/				
3520	1030	CHKCRC, TAD	ERREG	/GET ERROR REGISTER
3521	0365	AND	(4000	/TEST FOR DCRC (DATA CRC ERROR)
3522	7650	SNA	CLA	/SKIP IF DCRC
3523	5235	JMP	ABORT	/ABORT OPERATION, INTERMITTENT ERROR
3524	4771'	JMS	CHKRDY	/CHECK FOR DRIVE WAS READY
3525	5306	JMP	LOGTRK	/NOT READY GO LOG TRACKING ERROR
/				
3526	4770'	LOGDCR, JMS	LOGERR	/LOG DATA CRC ERROR
3527	0011	DCRCR		/DRIVE STATE TABLE INDEX
/				
3530	6212	RETRY, CIF	10	/GOING TO FLD 1
3531	4764'	JMS	RESALL	/RESTORE ALL DRV TABLE ENTRIES
3532	4536	SETPNT		/SET TBLPNT TO NEW CYLINDER AND HEAD
3533	0002	NEWCYL		/TBL INDX
3534	1504	TAD I	TBLPNT	/GET NEW CYLINDER AND HEAD
3535	3337	DCA	POSGO	/SAVE FOR SEEK RETRY

```

3536 4546      SETPOS      /CALL SEEK TO ABSOLUTE TRACK ROUTINE
3537 7402      POSGO, HLT/NEWCYL /CYL AND HEAD TO SEEK TO
3540 5776      JMP      DRVGO   /DROP DRIVE, RETRY EXCEEDED
/
3541 4770      EREXT1, JMS      LOGERR   /LOG SOFT ERROR
3542 0004      SFTERR      /DRIVE STATE TABLE INDEX
3543 4547      GETSTA      /GET STATUS
3544 4767      JMS      FINRPT   /FINISH REPORT
3545 6212      CIF      10      /GOING TO FLD 1
3546 4764      JMS      RESALL   /RESTORE FUNCOD, BITMODE, ETC.
3547 4560      RESET      /MAKE RESET LAST OPERATION
3550 2355      ISZ      ERRPC     /INCREMENT FOR GOOD RETURN
3551 1356      EREXT2, TAD      POSPC   /GET POSSET RETURN PC
3552 3763      DCA      POSSET   /RESTORE RETURN PC
3553 3357      DCA      ERRFLG    /CLEAR ERROR FLAG
3554 5755      JMP I      ERRPC     /RETURN
/
3555 0000      ERRPC, 0      /ERROR RETURN PC OF FIRST CALL
3556 0000      POSPC, 0      /POSSET RETURN PC OF FIRST CALL
3557 0000      ERRFLG, 0     /ERROR FLAG FOR FIRST ENTRY SWITCH
/
/
/*****
/
3563 2400
3564 2115
3565 4000
3566 3333
3567 3600
3570 3612
3571 3047
3572 6000
3573 7000
3574 1066
3575 0210
3576 3322
3577 0140
3600 3600
PAGE
/
/
/
/ROUTINE TO FINISH STATUS REPORT
/
/      CALLED BY:      JMS      FINRPT
/
/
3600 0000      FINRPT, 0
3601 4537      GETSWR      /GET SWITCH REGISTER
3602 0043      AND      K100  /CHECK FOR INHIBIT ERROR TYPEOUT
3603 7640      SZA      CLA   /SKIP IF OK TO PRINT ERROR MESSAGE
3604 5600      JMP I      FINRPT /NO ERROR REPORT, EXIT
3605 4540      STAPRT     /PRINT STATUS
3606 1764      ERTBL2    /START OF MESSAGE ADDRESS IN FLD 1
3607 4540      STAPRT     /PRINT STATUS

```

3610 2031
3611 5600

ERTBL3
JMP I FINRPT /RETURN

/ROUTINE TO LOG AN ERROR IN DRIVE STATE TABLE

CALLLED BY: JMS LOGERR
FOLLOWED BY: HRDERR /OR ANY DRIVE STATE TABLE INDEX

3612	0000	LOGERR, 0	
3613	7300	CLA CLL	
3614	1612	TAD I LOGERR	/GET DRIVE STATE TABLE INDEX FROM CALL+1
3615	3217	DCA ERRPNT	/SAVE AS INDEX FOR SETPNT CALL
3616	4536	SETPNT	/SET TABLE POINT TO DRIVE STATE TABLE ENTRY
3617	7402	ERRPNT, HLT/ERROR INDEX	/MODIFIED DRIVE STATE TABLE INDEX FROM CALL+1
3620	2504	ISZ I TBLPNT	/INCREMENT DRIVE STATE TABLE ERROR ENTRY
3621	7000	NOP	/IN CASE OF OVERFLOW
3622	2212	ISZ LOGERR	/INCREMENT RETURN PC
3623	5612	JMP I LOGERR	/EXIT

/ROUTINE TO READ AND CHECK ALL 16 SECTORS FOR THE OVERWRITE TEST

/USES THE SECTOR DATA MAP TO DETERMINE SECTOR DATA PATTERN NUMBER
/THE SECTOR DATA MAP IS CALLED "STBL16"

CALLLED BY: JMS CHK16

/RETURN CALL+1 IF UNRECOVERABLE ERROR
/RETURN CALL+3 IF OK

3624	0000	CHK16, 0	
3625	7300	CLA CLL	
3626	1377	TAD (34	
3627	3121	DCA SECSAV	/FIRST SECTOR IS 34
3630	1376	TAD (-20	
3631	3254	DCA CHKCNT	/16 SECTORS TO CHECK, SET UP COUNTER
3632	1375	TAD (STBL16	/GET ADDRESS OF SECTOR DATA MAP
3633	3255	DCA PTBL16	/SET UP SECTOR DATA MAP POINTER
3634	1655	LP16, TAD I PTBL16	/GET DATA PATTERN FROM SECTOR DATA MAP

```

3635 3113      DCA    PATNUM      /STORE DATA PATTERN FOR READ/VERIFY
3636 4774      JMS    RDSEC       /READ AND CHECK A SECTOR
3637 5253      JMP     EX16        /UNRECOVERABLE ERROR
3640 2121      ISZ    SECSAV      /INCREMENT SECTOR #
3641 1121      TAD     SECSAV      /GET SECTOR #
3642 7041      CIA
3643 1373      TAD     (47
3644 7710      SPA     CLA         /SKIP IF < OR = 47
3645 3121      DCA     SECSAV      /SECTOR=0
3646 2255      ISZ    PTBL16      /INCREMENT SECTOR DATA MAP POINTER
3647 2254      ISZ    CHKCNT      /INCREMENT SECTOR CHECK COUNTER
3650 5234      JMP     LP16        /CHECK NEXT SECTOR
3651 2224      ISZ    CHK16        /INCREMENT FOR GOOD RETURN
3652 2224      ISZ    CHK16        /INCREMENT FOR GOOD RETURN
3653 5624      EX16,  JMP I    CHK16 /EXIT

3654 0000      CHKCNT, 0          /SECTOR CHECK COUNTER
3655 0000      PTBL16, 0         /SECTOR DATA MAP POINTER

/
/
/*****
/
/ROUTINE TO CHANGE THE DEVICE CODE OF ALL IOTS USED IN PROGRAM
/
/      CALLED BY:      CNGIOT
/
/THE DEVICE CODE IS DETERMINED BY THE UNIT NUMBER'S DEVICE CODE
/ENTER WITH THE FIRST IOT BASE CODE IN AC (0600 OR 0620)
/
3656 0000      IOTCNG, 0
3657 3323      DCA     IOT60       /SAVE FIRST IOT BASE CODE
3660 1323      TAD     IOT60
3661 1372      TAD     (10         /FORM SECOND IOT BASE CODE
3662 3324      DCA     IOT61       /SAVE FOR IOT CHANGE
3663 1371      TAD     (DCTB60     /GET IOT TABLE ADDRESS
3664 3320      DCA     IOTP1       /SET UP TABLE POINTER
3665 1370      TAD     (DCTB61-DCTB60 /GET TABLE SIZE
3666 7041      CIA
3667 3322      DCA     IOTC1       /SAVE AS COUNTER

/
3670 1720      DC60LP, TAD I    IOTP1 /GET IOT ADDRESS
3671 3321      DCA     IOTP2       /SAVE POINTER
3672 1721      TAD I    IOTP2     /GET IOT
3673 0367      AND     (7007      /MASK OFF DEVICE CODE
3674 1323      TAD     IOT60       /ADD DEVICE CODE (0600 OR 0620)
3675 3721      DCA I    IOTP2     /SAVE NEW IOT
3676 2320      ISZ    IOTP1       /INCREMENT IOT LOCATION TABLE POINTER
3677 2322      ISZ    IOTC1       /INCREMENT IOT COUNTER
3700 5270      JMP     DC60LP     /CONTINUE MODIFYING FIRST SET OF IOTS

/
3701 1366      TAD     (DCTB61     /GET ADDRESS OF SECOND SET OF IOTS TABLE
3702 3320      DCA     IOTP1       /SET UP IOT ADDRESS TABLE POINTER

```

```

3703 1365      TAD      (DCEND-DCTB61  /GET TABLE SIZE
3704 7041      CIA
3705 3322      DCA      IOTC1          /SAVE AS IOT COUNTER
/
3706 1720      DC61LP, TAD I  IOTP1      /GET IOT ADDRESS FROM TABLE
3707 3321      DCA      IOTP2      /SET UP IOT POINTER
3710 1721      TAD I  IOTP2      /GET IOT
3711 0367      AND      (7007      /MASK OFF DEVICE CODE
3712 1324      TAD      IOT61      /ADD DEVICE CODE (0610 OR 0630)
3713 3721      DCA I  IOTP2      /SAVE MODIFIED IOT
3714 2320      ISZ      IOTP1      /INCREMENT IOT ADDRESS TABLE POINTER
3715 2322      ISZ      IOTC1      /INCREMENT IOT COUNTER
3716 5306      JMP      DC61LP      /CONTINUE CHANGING SECOND SET OF IOTS
/
3717 5656      JMP I  IOTCNG      /RETURN, ALL IOTS MODIFIED
/
/
3720 0000      IOTP1, 0          /IOT POINTER
3721 0000      IOTP2, 0          /IOT POINTER
3722 0000      IOTC1, 0          /IOT COUNTER
3723 0000      IOT60, 0          /FIRST SET OF IOTS DEVICE CODE (0600 OR 0620)
3724 0000      IOT61, 0          /SECOND SET OF IOTS DEVICE CODE (0610 OR 0630)
/*****
/
/ROUTINE TO WAIT FOR PACK MOUNTING
/
/      CALLED BY:      MNTPAK
/
/
3725 0000      PAKMNT, 0
3726 4551      MESAG
3727 2774      MOUNT
3730 1035      TAD      CURUNT      /"MOUNT PACK ON DRIVE "
3731 1364      TAD      (UNITO      /GET CURRENT UNIT NUMBER
3732 3360      DCA      UNTPNT      /USE AS INDEX INTO UNIT/DRIVE TABLE
3733 1760      TAD I  UNTPNT      /SAVE UNIT/DRIVE POINTER
3734 1044      TAD      K260      /GET DRIVE #
3735 4553      PRNT      /ADD ASCII BASE CODE
3736 1035      TAD      CURUNT      /PRINT DRIVE #
3737 1363      TAD ,  (ONSYSO      /GET CURRENT UNIT #
3740 3360      DCA      UNTPNT      /INDEX IN ON SYSTEM TABLE
3741 1760      TAD I  UNTPNT      /SAVE ON SYS POINTER
3742 7640      SZA      CLA      /GET ON SYS FLG
3743 5352      JMP      OTHRS      /SKIP IF ON THIS SYSTEM
3744 1074      TAD      SYS2F      /ON OTHER SYSTEM
3745 7650      SNA      CLA      /GET 2 SYSTEMS FLG
3746 5354      JMP      WAITOP      /SKIP IF 2 SYSTEMS UNDER TEST
3747 4551      MESAG      /WAIT FOR OPERATOR
3750 3151      THISYS
3751 5354      JMP      WAITOP      /"THIS SYSTEM"
/                                     /WAIT FOR OPERATOR
3752 4551      OTHRS,  MESAG
3753 3160      OTHRSY      /"OTHER SYSTEM"
/

```

```

3754 4551 WAITOP, MESAG
3755 3170 WAITM
3756 4562 WAITCR
3757 5725 JMP I PAKMNT
/
3760 0000 UNTPNT, 0
/
/
3763 0062
3764 0056
3765 0016
3766 7536
3767 7007
3770 0006
3771 7530
3772 0010
3773 0047
3774 1112
3775 6221
3776 7760
3777 0034
4000

PAGE
/
/
/
/*****
/
/ROUTINE TO COMPARE PACK SERIAL NUMBERS
/
/ CALLED BY: CMPSN
/
/
4000 0000 SNCMP, 0
4001 4546 SETPOS
4002 2777
4003 5600 JMP I SNCMP
4004 7300 CLA CLL
4005 4536 SETPNT
4006 0020 INITCA
4007 1106 TAD BUFAD1
4010 3504 DCA I TBLPNT
4011 4777 JMS CLRBF1
4012 4547 GETSTA
4013 4542 ERRCHK
4014 5217 JMP .+3
4015 4541 ERROR
4016 5600 JMP I SNCMP
4017 4776 JMS RDYCHK
4020 5223 JMP .+3
4021 4541 ERROR
4022 5600 JMP I SNCMP
4023 4536 SETPNT
4024 0017 WRDCNT
4025 1054 TAD M400
4026 3504 DCA I TBLPNT

```

```

/"WAIT FOR DRIVE READY"
/GO WAIT FOR <CR>
/RETURN

```

```

/UNIT POINTER

```

HP 001

```

/SEEK TO BAD SECTOR FILE TRACK
/HD 1,CYL 777
/UNRECOVERABLE ERROR
/SET TBLPNT TO INITIAL CURRENT ADDRESS
/TBL INDX
/GET BUFFER ADDRESS
/SAVE FOR READ
/CLEAR BUFFER
/GET DRIVE STATUS
/CHECK FOR STATUS ERRORS
/NO ERRORS
/UNRECOVERABLE ERROR
/CHECK FOR DRIVE READY
/DRIVE READY
/DRIVE NOT READY
/UNRECOVERABLE ERROR
/SET TBLPNT TO WORD COUNT
/TBL INDX
/WORD COUNT = 256, 1 SECTOR
/SAVE WORD COUNT

```

4027	1375	TAD	(FACBAD	/GET FACTORY BAD SECTOR TABLE ADDRESS
4030	3320	DCA	SPNTR	/SET UP TABLE POINTER
4031	4536	SETPNT		/SET TBLPNT TO SECTOR ADDRESS
4032	0021	SECADD		/TBL INDX
4033	1720	TAD I	SPNTR	/GET FACTORY BAD SECTOR FILE #
4034	3504	DCA I	TBLPNT	/SAVE FOR READ
4035	1504	TAD I	TBLPNT	/GET SECTOR
4036	7001	IAC		
4037	4536	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR REGISTER
4040	0024	XENDSC		/TBL INDX
4041	3504	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR
4042	7327	CLA CLL	CML IAC RTL	/CODE 6 FOR READ
4043	3105	DCA	FUNCOD	/READ DATA IS FUNCTION
4044	4545	XFER		/READ FACTORY BAD SECTOR FILE
4045	4544	WAITDN		/WAIT FOR DONE
4046	4774	JMS	FATAL	/NO DONE AFTER READ
4047	4542	ERRCHK		/CHECK FOR ERRORS
4050	5253	JMP	+.3	/NO ERROR
4051	4541	ERROR		/READ ERROR
4052	5600	JMP I	SNCMP	/UNRECOVERABLE ERROR
4053	1106	TAD	BUFAD1	/GET BUFFER ADDRESS
4054	1052	TAD	M1	/-1 FOR AUTO INDEX
4055	3016	DCA	AUTO16	/SET UP POINTER
4056	6211	CDF	10	/CDF TO BUFFER FLD
4057	1416	TAD I	AUTO16	/GET FIRST SERIAL NUMBER WORD
4060	0045	AND	K377	/MASK OFF GARBAGE
4061	7041	CIA		
4062	1077	TAD	SERNM1	/COMPARE WITH ORIGINAL SERIAL NUMBER
4063	7640	SZA	CLA	/SKIP IF SAME
4064	5312	JMP	SNERR	/WRONG PACK, GO HANDLE ERROR
4065	1416	TAD I	AUTO16	/GET SECOND SERIAL NUMBER WORD
4066	0045	AND	K377	/MASK OFF GARBAGE
4067	7041	CIA		
4070	1100	TAD	SERNM2	/COMPARE SECOND SERIAL NUMBER WORD
4071	7640	SZA	CLA	/SKIP IF EQUAL
4072	5312	JMP	SNERR	/WRONG PACK, GO HANDLE ERROR
4073	1416	TAD I	AUTO16	/GET THIRD SERIAL NUMBER WORD
4074	0045	AND	K377	/MASK OFF GARBAGE
4075	7041	CIA		
4076	1101	TAD	SERNM3	/COMPARE WITH ORIGINAL SERIAL NUMBER
4077	7640	SZA	CLA	/SKIP IF EQUAL
4100	5312	JMP	SNERR	/WRONG PACK, GO HANDLE ERROR
4101	1416	TAD I	AUTO16	/GET FOURTH SERIAL NUMBER WORD
4102	0045	AND	K377	/MASK OFF GARBAGE
4103	7041	CIA		
4104	1102	TAD	SERNM4	/COMPARE WITH ORIGINAL SERIAL NUMBER
4105	7640	SZA	CLA	/SKIP IF EQUAL
4106	5312	JMP	SNERR	/WRONG PACK, GO HANDLE ERROR
4107	6201	CDF	00	
4110	2200	ISZ	SNCMP	/INCREMENT FOR GOOD RETURN
4111	5600	JMP I	SNCMP	/EXIT
4112	6201	SNERR,	CDF	00


```

4113 4551      MESAG
4114 3225      WRNGPK
4115 4551      MESAG
4116 3234      SNNCMP
4117 5600      JMP I   SNCMP      /*"WRONG PACK!"
                                      /*"SERIAL NUMBER DOES NOT MATCH"
                                      /*BAD RETURN

4120 0000      SPNTR, 0
/
/
/
/
/
/
/ADJACENT CYLINDER WRITE TEST
/
/      CALLED BY:      ADJWRT
/
/RETURN CALL+1 IF UNSUCCESSFUL
/RETURN CALL+2 IF SUCCESSFUL
/
/
4121 0000      WRTADJ, 0
4122 7344      CLA CLL CMA RAL      /*AC=-2
4123 3773      DCA      SURSWT      /*SET UP 2 SURFACE SWITCH
4124 1041      TAD      K5
4125 7041      CIA
4126 3125      DCA      TRKCNT      /*5 TRKS PER SURFACE COUNTER
4127 3772      DCA      SEEK0      /*SEEK IN FROM CYL 0 SURFACE 0 INIT
4130 1047      TAD      K0777      /
4131 3771      DCA      SEK777      /*SEEK OUT FROM CYL 777 SURFACE 0 INIT
4132 4770      JMS      DATGEN      /*GENERATE 1 SECTOR OF DATA
4133 1367      TAD      (SURFO      /*GET CYL AND HEAD TABLE ADDRESS
4134 3124      DCA      CYLPNT      /*SET UP POINTER FOR CYL AND HEAD TO USE

/SET UP THE TRACK OFFSET TABLE POINTER, AND TRACK TO WRITE
/TRACK IS LOWEST TRACK OF A SET OF 5 TRACKS + OFFSET (0,1,2,3,4)
/
4135 1366      BEGTRK, TAD      (WTRK      /*GET TRACK OFFSET INDEX TABLE ADDRESS
4136 1075      TAD      NUMUNT      /*INDEX USING # OF UNITS UNDER TEST
4137 3131      DCA      WRTPNT      /*SAVE ADDRESS
4140 1531      TAD I   WRTPNT      /*GET ADDRESS OF STEP INDEX TABLE
4141 1103      TAD      STPNUM      /*INDEX USING STEP #
4142 3131      DCA      WRTPNT      /*SAVE ADDRESS OF STEP # IN OFFSET TABLE
4143 1531      TAD I   WRTPNT      /*GET STEP # ADDRESS
4144 3131      DCA      WRTPNT      /*SET UP STEP POINTER
4145 1524      TAD I   CYLPNT      /*GET TRACK AND HEAD FROM TABLE
4146 1531      TAD I   WRTPNT      /*ADD TRACK OFFSET
4147 3765      DCA      WTRK      /*SAVE TRACK TO SEEK TO
4150 5764      JMP      SECLPA      /*GO TO NEXT PAGE

/
/
4151 2321      ISZ      WRTADJ      /*INCREMENT FOR GOOD RETURN
4152 5721      ADJEX, JMP I   WRTADJ /*RETURN

```

HP 001
HP 001

4164 4200
 4165 4265
 4166 7071
 4167 5702
 4170 2675
 4171 4403
 4172 4262
 4173 4505
 4174 7336
 4175 7554
 4176 2333
 4177 7320
 4200

PAGE

/

/

/

/

/SET UP SECTOR TABLE POINTER, SUB-STEP TABLE POINTER, GET
 /NUMBER OF SECTORS AND STARTING SECTOR

4200	1377	SECLPA, TAD	(STBL	/GET ADDRESS OF # OF UNITS INDEX TABLE
4201	1075	TAD	NUMUNT	/INDEX USING # OF UNITS
4202	3123	DCA	SECPNT	/SAVE TABLE ADDRESS
4203	1523	TAD I	SECPNT	/GET STEP INDEX TABLE ADDRESS
4204	1103	TAD	STPNUM	/INDEX USING STEP #
4205	3123	DCA	SECPNT	/SAVE SUB-STEP TABLE ENTRY ADDRESS
4206	1523	TAD I	SECPNT	/GET ADDRESS OF SUB-STEP SECTOR TABLE ENTRY
4207	3126	DCA	STPNT	/SET UP SUB-STEP TABLE POINTER
4210	1526	TAD I	STPNT	/GET SUB-STEP SECTOR TABLE ENTRY ADDRESS
4211	3123	DCA	SECPNT	/SAVE ADDRESS
/				
4212	1523	SECLPB, TAD I	SECPNT	/GET # OF SECTORS
4213	7041	CIA		/NEGATE FOR USE AS COUNTER
4214	3036	DCA	SECCNT	/SET UP SECTOR COUNTER
4215	2123	ISZ	SECPNT	/INCREMENT SECTOR TABLE POINTER
4216	1523	TAD I	SECPNT	/GET STARTING SECTOR
4217	3121	DCA	SECSAV	/SAVE STARTING SECTOR
4220	1121	TAD	SECSAV	/GET SECTOR
4221	3127	DCA	WSEC	/SAVE AS SECTOR TO WRITE
/				
4222	1103	STPCHK, TAD	STPNUM	/GET CURRENT STEP #
4223	1052	TAD	M1	/SUB 1
4224	7650	SNA	CLA	/SKIP IF NOT STEP # 1
4225	5260	JMP	OUTSEK	/STEP # 1, INITIALIZE OF TRACK N, NO ADJ. CYL. TO CHECK

/

/CALCULATE THE ADJACENT TRACK #

/

/

4226	7344	CLA	CLL	CMA	RAL	/AC=-2
4227	1531	TAD I	WTRPNT			/ADD TRACK OFFSET
4230	7700	SNA	CLA			/SKIP IF TRACK OFFSET < 2, ATRK=WTRK+1
4231	5234	JMP	WTRKM1			/OFFSET > 2
4232	7001	WTRKP1, IAC				

4233	7410	SKP			
4234	1052	WTRKM1,	TAD	M1	/AC=-1
4235	1265		TAD	WTRK	/ADD TRACK TO WRITE
4236	3776		DCA	ATRK	/SAVE ADJACENT TRACK=WTRK + OR - 1
/					
/CALCULATE ADJACENT SECTOR					
/					
4237	7344	ADJCAL,	CLA	CLL	CMA RAL
4240	1531		TAD	I	WRTPT
4241	7700		SMA		CLA
4242	5245		JMP		ASECP6
4243	1375	ASECN6,	TAD		(-6
4244	7410		SKP		
4245	7327	ASECP6,	CLA	CLL	CML IAC RTL
4246	1127		TAD		WSEC
4247	3130		DCA		ASEC
4250	1130		TAD		ASEC
4251	7041		CIA		
4252	1374		TAD		(47
4253	7700		SMA		CLA
4254	5260		JMP		OUTSEK
4255	1130		TAD		ASEC
4256	1373		TAD		(-50
4257	3130		DCA		ASEC
/					
/					
4260	3133	OUTSEK,	DCA		SEKDIR
4261	4546		SETPOS		
4262	0000	SEEK0,	0/2000		
4263	5772		JMP		ADJEX
/					
4264	4546	SEKTRK,	SETPOS		
4265	0000	WTRK,	0		
4266	5772		JMP		ADJEX
4267	4771		JMS		WRTSEC
4270	5772		JMP		ADJEX
4271	1035		TAD		CURUNT
4272	3113		DCA		PATNUM
4273	4770		JMS		RDSEC
4274	5772		JMP		ADJEX
/					
4275	7340		CLA	CLL	CMA
4276	1103		TAD		STPNUM
4277	7650		SMA		CLA
4300	5304		JMP		NXTS1
/					
4301	4767		JMS		ADJCHK
4302	4573		ADJERR		
4303	5772		JMP		ADJEX
/					
4304	2127	NXTS1,	ISZ		WSEC
4305	1127		TAD		WSEC
4306	1366		TAD		(-47
4307	7740		SMA	SZA	CLA
4310	3127		DCA		WSEC

/SECTOR TO WRITE=0 FOR TRACK N+1

4311	2036	ISZ	SECCNT	/INCREMENT SECTOR WRITE COUNT
4312	7410	SKP		/SKIP IF NOT DONE WITH SECTOR SET
4313	5317	JMP	OUTST	/GO DO SEEK OUT
4314	1127	TAD	WSEC	/GET SECTOR TO WRITE
4315	3121	DCA	SECSAV	/SAVE IT
4316	5222	JMP	STPCHK	/GO CHECK STEP # AND WRITE NEXT SECTOR
/				
4317	2123	OUTST,	ISZ	SECPNT
4320	1523	TAD I	SECPNT	/INCREMENT SECTOR TABLE POINTER
4321	7041	CIA		/GET # OF SECTORS
4322	3036	DCA	SECCNT	/NEGATE FOR USE AS COUNTER
4323	2123	ISZ	SECPNT	/SAVE SECTOR COUNT
4324	1523	TAD I	SECPNT	/INCREMENT SECTOR TABLE POINTER
4325	3121	DCA	SECSAV	/GET STARTING SECTOR
4326	1121	TAD	SECSAV	/SAVE SECTOR TO WRITE
4327	3127	DCA	WSEC	/GET SECTOR
/				
4330	1103	STPCK,	TAD	STPNUM
4331	1052	TAD	M1	/GET STEP #
4332	7650	SNA	CLA	/SUB 1
4333	5765	JMP	INSEK	/SKIP IF NOT STEP # 1
/				
4334	7344	ACAL,	CLA CLL	CMA RAL
4335	1531	TAD I	WRTPT	/AC=-2
4336	7700	SNA	CLA	/ADD THE TRACK OFFSET
4337	5342	JMP	ASP6	/SKIP IF TRACK OFFSET < 2, ASEC=WSEC-6
/				
4340	1375	ASH6,	TAD	(-6
4341	7410	SKP		/WSEC-6 FOR ADJ. SECTOR
4342	7327	ASP6,	CLA CLL	CML IAC RTL
4343	1127	TAD	WSEC	/WSEC+6 FOR ADJ. SECTOR
4344	3130	DCA	ASEC	/ADD WRITTEN SECTOR
4345	1130	TAD	ASEC	/SAVE ADJ. SECTOR
4346	7041	CIA		/GET ADJ. SECTOR
4347	1374	TAD	(47	/ADD MAXIMUM SECTOR
4350	7700	SNA	CLA	/SKIP IF ADJ. SECTOR > 47
4351	5765	JMP	INSEK	/GO WRITE NEXT SECTOR
4352	1130	TAD	ASEC	/GET ADJ. SECTOR
4353	1373	TAD	(-50	/SUB 50
4354	3130	DCA	ASEC	/SAVE ACTUAL ADJACENT SECTOR
4355	5765	JMP	INSEK	/GO TO NEXT PAGE
/				
4365	4400			
4366	7731			
4367	4506			
4370	1112			
4371	1000			
4372	4152			
4373	7730			
4374	0047			
4375	7772			
4376	4510			
4377	6740			
	4400			

```

/
/
/
4400 1052 INSEK, TAD M1
4401 3133 DCA SEKDIR
4402 4546 SETPOS
4403 0777 SEK777, 0777/2777
4404 5777 JMP ADJEX
/SET SEEK DIRECTION FLG FOR SEEK OUT
/SEEK TO INNERMOST TRACK CYL 777
/INNERMOST TRACK, SURFACE 0 OR 1
/ERROR ATTEMPTING TO SEEK TO INNERMOST TRACK
HP 001

/
4405 1776 TAD WTRK
4406 3210 DCA WRTTRK
4407 4546 SETPOS
4410 0000 WRTTRK, 0
4411 5777 JMP ADJEX
/GET TRACK WRITTEN
/SAVE FOR SEEK OUT
/SEEK OUT TO TEST TRACK
/TEST TRACK CYL AND HEAD
/ERROR DURING ATTEMPT TO SEEK OUT TO TEST TRACK

/
4412 4775 WNXSEC, JMS WRTSEC
4413 5777 JMP ADJEX
4414 1035 TAD CURUNT
4415 3113 DCA PATNUM
4416 4774 JMS RDSEC
4417 5777 JMP ADJEX
4420 1052 TAD M1
4421 1103 TAD STPNUM
4422 7650 SNA CLA
4423 5227 JMP NXTS2
/WRITE TEST SECTOR
/ERROR ATTEMPTING TO WRITE TEST SECTOR
/GET CURRENT UNIT
/SAVE AS PATTERN NUMBER TO CHECK
/CHECK SECTOR JUST WRITTEN
/ERROR WRITING TEST SECTOR

/STEP # - 1
/SKIP IF NOT STEP # 1
/STEP # 1, NO ADJACENT SECTORS TO CHECK

/
4424 4306 JMS ADJCHK
4425 4573 ADJERR
4426 5777 JMP ADJEX
/CHECK ADJACENT SECTOR
/ADJ. SECTOR ERROR
/EXIT

/
4427 2127 NXTS2, ISZ WSEC
4430 1127 TAD WSEC
4431 1373 TAD (-47
4432 7740 SNA SZA CLA
4433 3127 DCA WSEC
4434 2036 ISZ SECCNT
4435 7410 SKP
4436 5242 JMP NXTIO
4437 1127 TAD WSEC
4440 3121 DCA SECSAV
4441 5772 JMP STPCK
/INCREMENT SECTOR TO WRITE
/GET SECTOR
/SKIP IF < OR = 47
/WSEC > 47, MAKE IT 0
/INCREMENT SECTOR WRITE COUNT
/SKIP IF NOT DONE
/DONE WITH SEEK IN AND SEEK OUT
/GET SECTOR TO WRITE
/SAVE IT
/GO CHECK FOR STEP 1 AND DO NEXT SECTOR

/COMPLETED A SEEK-IN/SEEK-OUT BLOCK OF SECTORS ON CURRENT TRACK
/NOW CHECK FOR MORE SEEK-IN/SEEK-OUT TESTS ON CURRENT TRACK

/
4442 2123 NXTIO, ISZ SECPNT
4443 1523 TAD I SECPNT
/INCREMENT SECTOR TABLE POINTER
/GET NEXT SECTOR COUNT, OR -1 IF END
/OF SUB-STEP, OR 4000 IF TIME FOR A TRACK
/CHANGE
/SKIP IF TRACK CHANGE OR END OF SUBSTEP
/GO PREPARE FOR MORE SECTORS THIS TRACK
/INCREMENT TO TEST FOR -1
/SKIP IF END OF SUB-STEP
/GO CHANGE TRACK TO WRITE
/INCREMENT SUB-STEP TABLE POINTER

4444 7500 SNA
4445 5771 JMP SECLPB
4446 7001 IAC
4447 7640 SZA CLA
4450 5260 JMP CNGTRK
4451 2126 ISZ STPNT

```


4511	5777	JMP	ADJEX	/ERROR ATTEMPTING TO SEEK TO ADJACENT TRACK
4512	1130	TAD	ASEC	/GET ADJACENT SECTOR
4513	3121	DCA	SECSAV	/SAVE AS SECTOR TO READ
4514	1364	TAD	(ADJU4	/GET ADDRESS OF ADJACENT SECTOR PATTERN # TABLE
4515	1127	TAD	WSEC	/ADD SECTOR WRITTEN
4516	3132	DCA	ADJPNT	/SAVE TABLE POINTER
4517	1531	TAD I	WRTPNT	/GET TRACK OFFSET
4520	7440	SZA		/SKIP IF TRACK WRITTEN=N-2 OFFSET=0
4521	5326	JMP	TRKNP2	/GO CHECK FOR TRACK N+2
4522	1363	TRKNM2, TAD	(-20	/TRACK N-2, FIRST SECTOR IS 20 SO SUB 20 FOR INDEX
4523	1132	TAD	ADJPNT	/ADD POINTER
4524	3132	DCA	ADJPNT	/SAVE ADJACENT SECTOR DATA PATTEN # TABLE POINTER
4525	5344	JMP	PATSV-2	/GO GET AND SAVE ADJACENT PATTERN #
4526	1053	TRKNP2, TAD	M4	/CHECK FOR TRACK N+2 OFFSET=4
4527	7640	SZA	CLA	/SKIP IF TRACK OFFSET=4, TRACK N+2
4530	5346	JMP	PATSV	/TRACK IS N + OR - 1
4531	1127	TAD	WSEC	/GET SECTOR WRITTEN
4532	1362	TAD	(-17	/CHECK FOR > 17
4533	7750	SPA SNA CLA		/SKIP IF SECTOR WRITTEN > 17
4534	5341	JMP	LE17	/SECTOR < OR = 17
4535	1361	TAD	(-40	/FIRST SECTOR IS 40, SUB 40 FOR TABLE INDEX
4536	1132	TAD	ADJPNT	/ADD ADJ. PATTERN NUMBER TABLE POINTER
4537	3132	DCA	ADJPNT	/SAVE POINTER
4540	5344	JMP	PATSV-2	/GO GET AND SAVE ADJACENT PATTERN #
4541	1360	LE17, TAD	(10	/SECTOR WRITTEN < OR = 17 ON TRACK N+2
4542	1132	TAD	ADJPNT	/ADD ADJ. SECTOR PATTERN # TABLE ADDRESS
4543	3132	DCA	ADJPNT	/SAVE TABLE POINTER
4544	1532	TAD I	ADJPNT	/GET ADJ. SECTOR PATTERN #
4545	0357	AND	(77	/MASK RIGHT BYTE FOR ADJACENT SECTOR PATTERN #
4546	3113	PATSV, DCA	PATNUM	/SAVE ADJACENT SECTOR PATTERN #
4547	4774	JMS	RDSEC	/READ AND CHECK SECTOR, SECTOR IN "SECSAV"
4550	5353	JMP	ACHKEX	/ERROR READING ADJ. SECTOR
4551	2306	ISZ	ADJCHK	/INCREMENT FOR GOOD RETURN
4552	2306	ISZ	ADJCHK	/INCREMENT FOR GOOD RETURN
4553	5706	ACHKEX, JMP I	ADJCHK	/RETURN
4557	0077			
4560	0010			
4561	7740			
4562	7761			
4563	7760			
4564	6745			
4565	4262			
4566	2000			
4567	4151			
4570	4135			
4571	4212			
4572	4330			
4573	7731			

4574 1112
 4575 1000
 4576 4265
 4577 4152
 4600

PAGE

/

/

/

/

/

/

/

/ADJACENT CYLINDER WRITE TEST ERROR HANDLER

CALLED BY: ADJERR

ERRADJ, 0

4600	0000			
4601	4551	MESAG		
4602	3007	ADJMSG		/"ADJACENT CYLINDER TEST ERROR"
4603	4551	MESAG		
4604	2507	DRVMSG		/"DRIVE"
4605	1024	TAD	CURDRV	
4606	1044	TAD	K260	
4607	4553	PRNT		/PRINT DRIVE #
4610	1113	TAD	PATNUM	/GET ADJACENT UNIT #
4611	1377	TAD	(UNITO	/INDEX INTO UNIT/DRIVE # TABLE
4612	3315	DCA	ADJDRV	/SAVE TABLE POINTER
4613	1715	TAD I	ADJDRV	/GET ADJ. UNIT DRIVE #
4614	3315	DCA	ADJDRV	/SAVE ADJACENT UNIT DRIVE #
4615	1103	TAD	STPNUM	
4616	1052	TAD	M1	
4617	7650	SNA	CLA	/SKIP IF NOT STEP # 1
4620	5240	JMP	SIERR	/STEP # 1, NO ADJACENT TRACK
4621	4551	MESAG		
4622	3027	ADJTO		/"ADJACENT TO"
4623	4551	MESAG		
4624	2507	DRVMSG		/"DRIVE"
4625	1315	TAD	ADJDRV	
4626	1044	TAD	K260	
4627	4553	PRNT		/PRINT ADJACENT DRIVE #
4630	1113	TAD	PATNUM	/GET DATA PATTERN # (UNIT #)
4631	1376	TAD	(ONSYSO	/INDEX INTO ON SYSTEM FLG TABLE
4632	3316	DCA	ADJUNT	/SAVE ON SYSTEM FLG TABLE ADDRESS
4633	1716	TAD I	ADJUNT	/GET ON SYSTEM FLG FROM TABLE
4634	7650	SNA	CLA	/SKIP IF ON OTHER SYSTEM
4635	5240	JMP	.+3	/ON THIS SYSTEM
4636	4551	MESAG		
4637	3160	OTHRSY		/"OTHER SYSTEM"
4640	4551	SIERR,		
4641	3036	MESAG		/"WRITTEN SECTOR: "
4642	1127	WRTNS		/GET SECTOR LAST WRITTEN
4643	4555	TAD	WSEC	/PRINT SECTOR # LAST WRITTEN
4644	4551	PRNAC		
4645	3047	MESAG		
4646	1130	ADJS		/"ADJACENT SECTOR: "
4647	4555	TAD	ASEC	/GET ADJACENT SECTOR #
4650	4551	PRNAC		/PRINT ADJACENT SECTOR #
		MESAG		


```

4651 3061      WRTNT      /*WRITTEN TRACK: "
4652 1775'     TAD        WTRK      /GET TRACK WRITTEN ON
4653 0047     AND        K0777     /MASK OFF HEAD SELECT
4654 4555     PRNAC      /PRINT TRACK WRITTEN
4655 4551     MESAG
4656 3072     ADJT
4657 1774'     TAD        ATRK      /*ADJACENT TRACK: "
4660 0047     AND        K0777     /GET ADJACENT TRACK
4661 4555     PRNAC      /MASK OFF HEAD SELECT
4662 4551     MESAG      /PRINT ADJACENT TRACK
4663 3103     SURFAC
4664 1775'     TAD        WTRK      /*SURFACE: "
4665 0373     AND        (2000     /GET TRACK WRITTEN
4666 7640     SZA        CLA      /MASK HEAD BIT
4667 7001     IAC        /SURFACE 0 SKIP
4670 1044     TAD        K260     /SURFACE 1
4671 4553     PRNT      /ADD ASCII BASE CODE
4672 4551     MESAG      /PRINT SURFACE 0 OR 1
4673 2465     SEKMSG
4674 1133     TAD        SEKDIR     /*SEEK "
4675 7640     SZA        CLA      /GET SEEK DIRECTION FLG
4676 5302     JMP        .+4      /SKIP IF SEEK-IN
4677 4551     MESAG      /SEEK OUT
4700 3111     INMSG
4701 5304     JMP        .+3      /*IN"
4702 4551     MESAG
4703 3114     OUTMSG      /*OUT"
4704 4551     MESAG
4705 2722     ERRPCH
4706 1200     TAD        ERRADJ     /*ERROR AT PC: "
4707 1052     TAD        M1      /GET ERROR CALL PC+1
4710 4555     PRNAC      /SUB 1 FOR PC
4711 4556     DOCRLF      /PRINT ERROR PC
4712 4535     C8CALL      /<CRLF>
4713 7000     NOP
4714 5600     JMP I      ERRADJ     /SR=
                                   /<CR> SO START TESTS OVER, RETURN

4715 0000     ADJDRV, 0
4716 0000     ADJUNT, 0
/
/
/*****
/
/ROUTINE TO WAIT FOR OPERATOR TO TYPE A CARRIAGE RETURN
/
/      CALLED BY:      WAITCR
/
4717 0000     CRWAIT, 0
4720 4551     MESAG
4721 3204     TYPCH      /*TYPE <RETURN> TO CONTINUE"
4722 4563     GETCHR     /GET INPUT FROM TTY KYBD
4723 1372     TAD        (-215     /ADD NEGATIVE ASCII CODE FOR <CR>

```

4724	7650	SNA	CLA	/SKIP IF NOT CARRIAGE RETURN
4725	5331	JMP	CREXT	/<CR> SO EXIT
4726	1371	TAD	(277	/GET ASCII CODE FOR QUESTION MARK
4727	4553	PRNT		/PRINT "?"
4730	5320	JMP	CRWAIT+1	/TRY AGAIN
4731	4556	CREXT,	DOCRLF	/<CRLF>
4732	5717	JMP I	CRWAIT	/RETURN

/

/

/*****

/

/

/ROUTINE TO CALL GET A CHARACTER ROUTINE IN FLD 1

/

/

CALLED BY: GETCHR

/

4733	0000	CHRGET,	0	
4734	7300	CLA	CLL	
4735	6212	CIF	10	
4736	4770	JMS I	(XINPUT	/GO TO FLD 1
4737	5733	JMP I	CHRGET	/RETURN

/

/

4770 1474
 4771 0277
 4772 7563
 4773 2000
 4774 4510
 4775 4265
 4776 0062
 4777 0056
 5000

PAGE

/

/

/

/

/

/OVER-WRITE TEST ERROR HANDLER

/

/

CALLED BY: OVRERR

/

5000	0000	ERROVR,	0	
5001	4551	MESAG		
5002	3117	OVRMSG		/"OVER-WRITE TEST ERROR"
5003	4551	MESAG		
5004	2507	DRVMSG		/"DRIVE"
5005	1024	TAD	CURDRV	/GET CURRENT DRIVE #
5006	1044	TAD	K260	/ADD ASCII BASE CODE
5007	4553	PRNT		/PRINT DRIVE #
5010	1134	TAD	OVRUNT	/GET UNIT # OVER-WRITTEN

5011	1377	TAD	(UNITO	/INDEX INTO UNIT/DRIVE TABLE
5012	3306	DCA	OVRDRV	/SAVE POINTER
5013	1706	TAD I	OVRDRV	/GET UNIT OVER-WRITTEN DRIVE #
5014	3306	DCA	OVRDRV	/SAVE DRIVE # OVER-WRITTEN
5015	1103	TAD	STPNUM	/GET STEP #
5016	1052	TAD	M1	/SUB 1
5017	7650	SNA	CLA	/SKIP IF NOT STEP # 1
5020	5240	JMP	STPIER	/STEP 1 INITIALIZE ERROR
5021	4551	MESAG		
5022	3133	OVER		/"OVER"
5023	4551	MESAG		
5024	2507	DRVMSG		/"DRIVE"
5025	1306	TAD	OVRDRV	/GET DRIVE # OVER-WRITTEN
5026	1044	TAD	K260	/ADD ASCII BASE CODE
5027	4553	PRNT		/PRINT DRIVE # OVER-WRITTEN
5030	1134	TAD	OVRUNT	/GET UNIT OVER-WRITTEN
5031	1376	TAD	(ONSYSO	/INDEX INTO ON SYSTEM TABLE
5032	3306	DCA	OVRDRV	/SAVE POINTER
5033	1706	TAD I	OVRDRV	/GET ON SYSTEM FLG
5034	7650	SNA	CLA	/SKIP IF UNIT ON OTHER SYSTEM
5035	5240	JMP	+.3	/ON THIS SYSTEM
5036	4551	MESAG		
5037	3160	OTHRSY		/"OTHER SYSTEM"
/				
5040	4551	STPIER, MESAG		
5041	3137	SECMMSG		/"SECTOR: "
5042	1121	TAD	SECSAV	/GET SECTOR #
5043	4555	PRNAC		/PRINT SECTOR #
5044	4551	MESAG		
5045	3144	TRKMSG		/"TRACK: "
5046	4536	SETPNT		/SET TBLPNT TO COMMAND A
5047	0022	XCOMA		/TBL INDX
5050	1504	TAD I	TBLPNT	/GET COMMAND A SENT
5051	0047	AND	K0777	/MASK CYL BITS
5052	4555	PRNAC		/PRINT TRACK #
5053	4551	MESAG		
5054	3103	SURFAC		/"SURFACE: "
5055	1504	TAD I	TBLPNT	/GET CYL AND HEAD
5056	0375	AND	(2000	/MASK HEAD BIT
5057	7640	SZA	CLA	/SKIP IF SURFACE 0
5060	7001	IAC		/SURFACE 1
5061	1044	TAD	K260	/ADD ASCII BASE CODE
5062	4553	PRNT		/PRINT SURFACE # 0 OR 1
5063	4551	MESAG		
5064	2465	SEKMSG		/"SEEK"
5065	1133	TAD	SEKDIR	
5066	7640	SZA	CLA	/SKIP IF SEEK-IN
5067	5273	JMP	+.4	/SEEK-OUT
5070	4551	MESAG		
5071	3111	INMSG		/"IN"
5072	5275	JMP	+.3	
5073	4551	MESAG		
5074	3114	OUTMSG		/"OUT"
5075	4551	MESAG		
5076	2722	ERRPCM		/"ERROR AT PC: "

HP 001

```

5077 1200      TAD      ERROVR      /GET PC+1
5100 1052      TAD      M1          /SUB 1
5101 4555      PRNAC          /PRINT ERROR PC
5102 4556      DOCRLF        /<CRLF>
5103 4535      C8CALL        /SR=
5104 7000      NOP
5105 5600      JMP I      ERROVR      /<CR> SO RETURN

/
5106 0000      OVRDRV, 0
/
/
/*****
/
/
/
/
5107 0000      HDRERR, 0
5110 4551      MESAG
5111 2365      HDRMSG          /"HEADER"
5112 4551      MESAG
5113 2476      ERR            /" ERROR"
5114 4556      DOCRLF        /<CR LF>
5115 4540      STAPRT        /PRINT DRIVE NUMBER
5116 2040      ERTBL4        /START OF MESSAGE ADDRESS
5117 4540      STAPRT        /PRINT CONTROLLER REGISTERS
5120 1736      ERTBL1        /ADDRESS OF START OF MESSAGES
5121 4551      MESAG
5122 2722      ERRPCM          /"ERROR AT PC:"
5123 1307      TAD      HDRERR      /GET PC+1
5124 1052      TAD      M1          /NOW HAVE PC
5125 4555      PRNAC          /PRINT ERROR PC
5126 4535      C8CALL        /SR=
5127 7000      NOP
5130 5707      JMP I      HDRERR      /RETURN

/
/
/*****
/
/
/ROUTINE TO CLEAR THE READ DATA BUFFER
/
/      CALLED BY:      JMS      CLRRBF
/
/
5131 0000      CLRRBF, 0
5132 7300      CLA      CLL
5133 1374      TAD      (BUFSZR      /GET READ DATA BUFFER SIZE
5134 7041      CIA          /NEGATE FOR USE AS COUNTER
5135 3347      DCA      BFCNT        /SET UP BUFFER COUNTER
5136 1373      TAD      (RBUFF        /GET ADDRESS OF BUFFER IN FLD 1
5137 1052      TAD      M1          /-1 FOR AUTO INDEX
5140 3012      DCA      AUTO12        /SET UP BUFFER LOCATION POINTER
5141 6211      CDF      10          /DF=1

```

5142	3412	CLRLOP, DCA 1	AUTO12	/CLEAR A BUFFER LOCATION
5143	2347	ISZ	BFCNT	/INCREMENT BUFFER COUNTER
5144	5342	JMP	CLRLOP	/CONTINUE CLEARING BUFFER
5145	6201	CDF	00	/DF=0
5146	5731	JMP 1	CLRRBF	/RETURN

5147	0000	BFCNT, 0	/BUFFER COUNTER
------	------	----------	-----------------

5173 6600
5174 1000
5175 2000
5176 0062
5177 0056
5200

PAGE

/OVER-WRITE TEST

CALLLED BY: OVRWRT

/THIS TEST WILL WRITE THE APPROPRIATE SECTORS FOR EACH STEP
/OF THE OVER-WRITE TEST. 10 TRACKS, 5 ON EACH SURFACE OF THE
/PACK BEING USED TO TEST DRIVE COMPATIBILITY, WILL BE USED
/OUT OF THE TOTAL OF 50 PERFECT TRACKS USED BY THE ADJACENT
/CYLINDER TEST. SECTORS 34,35,36,37,40,41,42,43,44,45,46,47,0,1,2, AND 3
/OF THE CENTER TRACK OF EACH SET OF 5 ADJACENT PERFECT TRACKS ARE
/USED FOR THIS TEST. THE STEP NUMBER DETERMINES WHICH DRIVE (UNIT)
/IS TESTED AND IS USED FOR INDEXING INTO SECTOR TABLES.

5200 0000
5201 7344
5202 3777
5203 1041
5204 7041
5205 3776
5206 3265
5207 1047
5210 3775
5211 4774
5212 1373

WRTOVR, 0

CLA CLL	CMA RAL	/AC=-2
DCA	SURCNT	/2 SURFACES
TAD	K5	
CIA		
DCA	CNTR5	/5 TRACKS FIRST SURFACE (0)
DCA	SEK1	/SURFACE 0 OUTER TRK (0)
TAD	K0777	/
DCA	SEK3	/SURFACE 0 INNER TRK (777)
JMS	DATGEN	/GENERATE DATA, 1 SECTOR, BUF 1, FOR UNIT
TAD	(SURFO	/GET CYL AND HEAD TBL ADDRESS

HP 001

5213	3124	DCA	CYLPNT	/SET UP POINTER FOR CYL AND HEAD TO USE
/				
5214	1372	SECLP1, TAD	(STBL16-1	/GET ADDRESS-1 OF SECTOR DATA MAP
5215	3011	DCA	AUTO11	/SET UP TABLE POINTER
5216	1075	TAD	NUMUNT	/GET NUMBER OF UNITS
5217	1371	TAD	(STPIDX	/USE # OF UNITS TO INDEX INTO UNIT TABLE
5220	3114	DCA	PNTR1	/SET UP TEMPORARY POINTER TO UNIT TABLE
5221	1514	TAD I	PNTR1	/GET ADDRESS OF STEP TABLE FROM UNIT TABLE
5222	3114	DCA	PNTR1	/SAVE TEMP POINTER TO STEP TABLE
5223	1103	TAD	STPNUM	/GET STEP NUMBER TO INDEX INTO STEP TABLE
5224	1114	TAD	PNTR1	/ADD ADDRESS OF STEP TABLE
5225	3114	DCA	PNTR1	/SAVE ADDRESS OF STEP TABLE ENTRY
5226	1514	TAD I	PNTR1	/GET ADDRESS OF SECTOR DATA MAP INITIALIZATION TABLE
5227	1052	TAD	M1	/-1 FOR AUTO INDEX
5230	3012	DCA	AUTO12	/SET UP POINTER TO SECTOR DATA MAP INIT TABLE
5231	1370	TAD	(-20	/16 TABLE ENTRIES
5232	3107	DCA	CNTR1	/SET UP TABLE COUNTER
/				
5233	1412	TBLIN, TAD I	AUTO12	/GET INIT VALUE
5234	3411	DCA I	AUTO11	/MOVE TO SECTOR DATA MAP
5235	2107	ISZ	CNTR1	/INCREMENT ENTRY COUNTER
5236	5233	JMP	TBLIN	/CONTINUE TO INIT SECTOR DATA MAP
/				
5237	1075	TAD	NUMUNT	/GET NUMBER OF UNITS
5240	1367	TAD	(OVRTBL	/INDEX INTO OVER-WRITE UNIT POINTER TABLE
5241	3123	DCA	SECPNT	/SAVE ADDRESS
5242	1523	TAD I	SECPNT	/GET ADDRESS OF STEP # TABLE
5243	1103	TAD	STPNUM	/USE STEP # TO INDEX INTO STEP # TABLE
5244	3123	DCA	SECPNT	
5245	1523	TAD I	SECPNT	/GET ADDRESS OF SECTOR TABLE ENTRY POINT
5246	3123	DCA	SECPNT	
5247	1523	TAD I	SECPNT	/GET STARTING SECTOR
5250	3122	DCA	SECSV	/SAVE FOR WRITE AND READ
5251	2123	ISZ	SECPNT	
5252	1523	TAD I	SECPNT	/GET # OF SECTORS
5253	7041	CIA		
5254	3766	DCA	SCNTSV	/SAVE SECTOR COUNT
5255	1122	TAD	SECSV	
5256	3121	DCA	SECSAV	
5257	1766	TAD	SCNTSV	
5260	3036	DCA	SECCNT	
5261	7340	CLA CLL	CMA	
5262	3765	DCA	SEKSWT	/SET UP SEEK OUT SWITCH
/				
5263	3133	OVRLP1, DCA	SEKDIR	/SET SEEK DIRECTION FLG FOR SEEK-IN
5264	4546	SETPOS		/SEEK TO OUTER TRACK
5265	0000	SEK1, 0/2000		/0 OR 2000 (TRK 0 SURF 0 OR 1)
5266	5330	JMP	OVREX	/ERROR ATTEMPTING TO SEEK TO OUTER CYLINDER
5267	7305	CLA CLL	IAC RAL	/AC=2
5270	1524	TAD I	CYLPNT	/LOWEST TRACK+2 = CENTER TRACK
5271	3273	DCA	SEK2	/SAVE CENTER TRK FOR SEEK IN
/				
5272	4546	SINOUT, SETPOS		/SEEK TO TRACK
5273	0000	SEK2, 0		/MODIFIED CYL AND HEAD FOR SEEK IN OR OUT
5274	5330	JMP	OVREX	/ERROR ATTEMPTING TO SEEK TO TEST CYLINDER

```

5275 4764' /OVRLP2, JMS WRTSEC /WRITE SECTOR, SECTOR NUMBER IN SECSAV
5276 5330 / JMP OVREX /ERROR ATTEMPTING TO WRITE TEST SECTOR

5277 7340 CLA CLL CMA /AC=-1
5300 1103 TAD STPNUM /ADD STEP NUMBER
5301 7650 SNA CLA /SKIP IF NOT STEP #1
5302 5763' JMP NXTSEC /DO NOT CHECK 16 SECTORS, FIRST STEP
5303 1121 TAD SECSAV
5304 1053 TAD M4
5305 7710 SPA CLA /SKIP IF SECTOR > OR = 34
5306 5314 JMP LOSEC /SECTOR 0,1,2, OR 3

5307 1121 HISEC, TAD SECSAV /GET SECTOR #
5310 1362 TAD (SEC34 /ADD ADDRESS OF SECTOR DATA MAP
5311 1361 TAD (-34 /SUBTRACK 34 FOR INDEX INTO SECTOR DATA MAP
5312 3331 DCA SECPTR /SET UP POINTER TO SECTOR DATA MAP
5313 5317 JMP STRPAT /GO STORE DATA PATTERN # IN SECTOR DATA MAP

5314 1121 LOSEC, TAD SECSAV /GET SECTOR # (0,1,2,OR 3)
5315 1360 TAD (SEC0 /ADD ADDRESS OF SECTOR DATA MAP SECTOR 0
5316 3331 DCA SECPTR /SET UP POINTER TO SECTOR DATA MAP

5317 1731 STRPAT, TAD I SECPTR /GET UNIT # BEING OVER-WRITTEN
5320 3134 DCA OVRUNT /SAVE FOR ERROR HANDLER
5321 1035 TAD CURUNT /GET CURRENT UNIT #
5322 3731 DCA I SECPTR /SAVE AS DATA PATTERN # IN SECTOR DATA MAP
5323 4757' JMS CHK16 /CHECK 16 SECTORS OF DATA
5324 4572 OVRERR /OVER-WRITE ERROR
5325 5330 JMP OVREX /OVER-WRITE ERROR, GO TO BAD EXIT
5326 5763' JMP NXTSEC /GO TO NEXT PAGE

5327 2200 / ISZ WRTOVR /INCREMENT FOR GOOD RETURN
5330 5600 OVREX, JMP I WRTOVR /RETURN

5331 0000 / SECPTR, 0
/
/
/

5357 3624
5360 6235
5361 7744
5362 6221
5363 5400
5364 1000
5365 5507
5366 5506
5367 6214
5370 7760
5371 6443
5372 6220
5373 5702
5374 2675
5375 5436

```

5376 5504
5377 5505
5400

PAGE

```

5400 2122  /
5401 1122  /
5402 7041  /
5403 1377  /
5404 7710  /
5405 3122  /
5406 1122  /
5407 3121  /
5410 2036  /
5411 5776' /
      /
5412 2123  /
5413 1523  /
5414 7710  /
5415 5231  /
5416 1523  /
5417 3122  /
5420 2123  /
5421 1523  /
5422 7041  /
5423 3306  /
5424 1122  /
5425 3121  /
5426 1306  /
5427 3036  /
5430 5776' /
      /
5431 2307  /
5432 5254  /
5433 1052  /
5434 3133  /
5435 4546  /
5436 0777  /
5437 5775' /
      /
5440 2123  /
5441 1523  /
5442 3122  /
5443 2123  /
5444 1523  /
5445 7041  /
5446 3306  /
5447 1306  /
5450 3036  /
5451 1122  /
5452 3121  /
5453 5774' /

```

NXTSEC.	ISZ	SECSV	/INCREMENT SECTOR TO OVER-WRITE
	TAD	SECSV	/GET NEXT SECTOR
	CIA		
	TAD	(47	/SEE IF NEXT SECTOR > 47 (INVALID)
	SPA	CLA	/SKIP IF NEXT SECTOR < OR = 47
	DCA	SECSV	/SET NEXT SECTOR TO 0
	TAD	SECSV	/GET NEXT SECTOR
	DCA	SECSAV	/SAVE FOR NEXT WRITE/READ
	ISZ	SECCNT	/INCREMENT SECTOR COUNT
	JMP	OVRLP2	/WRITE NEXT SECTOR
	ISZ	SECPNT	/SECTOR GROUP COMPLETE, INCREMENT SECTOR TABLE POINTER
	TAD I	SECPNT	/GET POSSIBLE NEXT STARTING SECTOR
	SPA	CLA	/SKIP IF MORE SECTORS
	JMP	SEKOUT	/PREPARE FOR SEEK OUT OR NEXT TRACK
	TAD I	SECPNT	/GET NEXT STARTING SECTOR THIS TRACK
	DCA	SECSV	/SAVE IT
	ISZ	SECPNT	/INCREMENT SECTOR TABLE POINTER
	TAD I	SECPNT	/GET NUMBER OF SECTORS
	CIA		
	DCA	SCNTSV	/SAVE SECTOR COUNT
	TAD	SECSV	/GET STARTING SECTOR
	DCA	SECSAV	/SAVE FOR NEXT WRITE/READ
	TAD	SCNTSV	/GET SECTOR COUNT
	DCA	SECCNT	/SET UP COUNTER
	JMP	OVRLP2	/CONTINUE WRITING/READING SECTORS
SEKOUT,	ISZ	SEKSWT	/SKIP IF SEEK OUT NOT DONE
	JMP	NXTRK1	/GO SET UP FOR NEXT TRACK
	TAD	M1	
	DCA	SEKDIR	/SET SEEK DIRECTION FLG FOR SEEK-OUT
	SETPOS		
SEK3,	0777/2777		/SEEK TO INNER CYL 0777 OR 2777
	JMP	OVREX	/ERROR ATTEMPTING TO SEEK TO INNER CYL
	ISZ	SECPNT	/INCREMENT SECTOR TABLE POINTER
	TAD I	SECPNT	/GET NEXT STARTING SECTOR
	DCA	SECSV	/SAVE IT
	ISZ	SECPNT	/INCREMENT SECTOR TABLE POINTER
	TAD I	SECPNT	/GET NUMBER OF SECTORS
	CIA		
	DCA	SCNTSV	/SAVE SECTOR COUNT
	TAD	SCNTSV	/GET SECTOR COUNT
	DCA	SECCNT	/SET UP SECTOR COUNTER
	TAD	SECSV	/GET NEXT SECTOR
	DCA	SECSAV	/SAVE FOR NEXT WRITE/READ
	JMP	SINOUT	/GO SEEK OUT AND FINISH TRACK

HP 001


```

5454 7340  NXTRK1. CLA CLL CMA
5455 1103      TAD      STPNUM      /ADD STEP NUMBER - 1
5456 7640      SZA      CLA          /SKIP IF STEP 1
5457 5263      JMP      NXTK1        /NOT STEP 1
5460 4773'     JMS      CHK16        /CHECK ALL 16 SECTORS FOR PATTERN 0
5461 5775'     JMP      OVREX        /INITIAL DATA ERROR OF STEP 1
5462 7000      NOP
5463 2124  NXTK1. ISZ      CYLPNT      /INCREMENT CYL TABLE POINTER
5464 7305      CLA CLL IAC RAL      /AC=2
5465 1524      TAD I  CYLPNT      /LOWEST CYL+2 = NEXT TRACK
5466 3772'     DCA      SEK2        /SAVE NEXT TRACK FOR SEEK
5467 2304      ISZ      CNTR5        /INCREMENT 5 TRACKS PER SURFACE COUNTER
5470 5771'     JMP      SECLP1       /CONTINUE THIS SURFACE
5471 2305      ISZ      SURCNT       /INCREMENT SURFACE SWITCH
5472 7610      SKP      CLA          /NOT DONE WITH TEST
5473 5770'     JMP      OVREX-1      /DONE WITH TEST
5474 1041      TAD      K5
5475 7041      CIA
5476 3304      DCA      CNTR5        /SET UP 5 TRKS PER SURFACE COUNT
5477 1051      TAD      K2000       /HEAD 1
5500 3767'     DCA      SEK1        /SURFACE 1 TRK 0
5501 1366      TAD      (2777       /HD 1 TRK 777
5502 3236      DCA      SEK3        /SURFACE 1 TRK 777
5503 5771'     JMP      SECLP1       /GO DO SURFACE 1

```

HP 001

```

5504 0000  CNTR5, 0
5505 0000  SURCNT, 0
5506 0000  SCNTSV, 0
5507 0000  SEKSWT, 0

```

```

/*****

```

```

5566 2777
5567 5265
5570 5327
5571 5214
5572 5273
5573 3624
5574 5272
5575 5330
5576 5275
5577 0047
5600

```

PAGE

/ROUTINE TO FIND 10 SETS OF 5 PERFECT ADJACENT TRACKS
/TRACKS ARE:

/ OUTER 0-20 (OCTAL)
/ QUARTER 170-210
/ MIDDLE 370-410
/ 3/4 570-610
/ INNER 756-776

/FIND 5 SETS OF THESE TRACKS ON EACH SURFACE
/IDEAL TRACKS ARE:

/ OUTER 0,1,2,3,4
/ QUARTER 170,171,172,173,174
/ MIDDLE 370,371,372,373,374
/ 3/4 570,571,572,573,574
/ INNER 756,757,760,761,762

/THE SEARCH FOR TRACKS STARTS WITH THE LOWEST TRACK OF EACH SET
/AND CONTINUES UNTIL LOWEST TRACK + 14 (OCTAL) IS TRIED
/THEN REPORT BAD PACK

/ CALLED BY: FNDTRK

5600	0000	TRKFND, 0		
5601	7300	CLA	CLL	
5602	1377	TAD	(-12	
5603	3274	DCA	SETCNT	/SET UP SETS OF TRACKS COUNTER FOR 10
5604	1376	TAD	(SURFO	/GET TRACK TABLE ADDRESS
5605	3275	DCA	TRKPNT	/SET UP LOWEST TRACK TABLE POINTER
5606	1375	TAD	(BADSEC	/GET BAD SECTOR FILE STORAGE ADDRESS
5607	3273	DCA	BADPT	/SET UP BAD SECTOR POINTER
5610	6211	CDF	10	/DF=1
5611	1673	TAD I	BADPT	/GET A BAD SECTOR CYLINDER AND HEAD
5612	6201	CDF	00	/DF=0
5613	7001	IAC		/INCREMENT TO TEST FOR NO BAD SECTORS
5614	7650	SNA	CLA	/SKIP IF BAD SECTORS
5615	5271	JMP	OKEX	/NO BAD SECTORS THIS PACK
5616	1374	LPINT1, TAD	(-16	
5617	3276	DCA	TRYCNT	/14 TRIES ALLOWED TO FIND 5 PERFECT TRACKS
5620	1675	LPINT2, TAD I	TRKPNT	/GET LOWEST TRACK OF SET OF 5
5621	1052	TAD	M1	/SUB 1 FOR FIRST TEST
5622	3277	DCA	TRKTRY	/SAVE TRK-1 TO TEST
5623	1373	LPINT3, TAD	(-5	
5624	3300	DCA	GDCNT	/NEED 5 ADJACENT PERFECT TRACKS
5625	1372	LPINT4, TAD	(-20	
5626	3301	DCA	BADCNT	/SET UP TO CHECK 16 BAD SECTORS

[illegible]

5702	0000	OUTER0, 0	/LOWEST TRACK OF 5, SURFACE 0, OUTER MOST SET
5703	0170	QUART0, 0170	/1/4
5704	0370	MIDDLE0, 0370	/MIDDLE
5705	0570	THREQ0, 0570	/3/4
5706	0756	INNER0, 0756	/INNER

5707 SURF1=.

5707	2000	OUTER1, 2000	/LOWEST TRACK OF 5, SURFACE 1, OUTER MOST SET
5710	2170	QUART1, 2170	/1/4
5711	2370	MIDDLE1, 2370	/MIDDLE
5712	2570	THREQ1, 2570	/3/4
5713	2756	INNER1, 2756	/INNER

5771 2176
5772 7760
5773 7773
5774 7762
5775 2200
5776 5702
5777 7766
6000

PAGE

/

/

/*****

/

TABLE OF SECTORS TO USE FOR 4 UNITS UNDER TEST

/

USED FOR OVER-WRITE TEST ONLY

/

.6000

STTBL4=.

/

6000	0034	UUT01, 34	/FIRST SECTOR
6001	0010	10	/8 SECTORS
6002	7777	-1	/DONE SEEK IN UNIT 0
6003	0044	44	/FIRST SECTOR
6004	0010	10	/8 SECTORS
6005	7777	-1	/DONE SEEK OUT UNIT 0

/

6006	0044	UUT12, 44	/FIRST SECTOR
6007	0004	4	/4 SECTORS
6010	7777	-1	/DONE SEEK IN UNIT 1
6011	0000	0	/FIRST SECTOR
6012	0004	4	/4 SECTORS
6013	7777	-1	/DONE SEEK OUT UNIT 1

/

6014	0040	UUT23, 40	/FIRST SECTOR
6015	0002	2	/2 SECTORS
6016	0000	0	/FIRST SECTOR
6017	0002	2	/2 SECTORS
6020	7777	-1	/DONE SEEK IN UNIT 2
6021	0042	42	/FIRST SECTOR
6022	0002	2	/2 SECTORS
6023	0002	2	/FIRST SECTOR
6024	0002	2	/2 SECTORS

6025	7777	-1	/DONE SEEK OUT UNIT 2
/			
6026	0036	UUT34, 36	/FIRST SECTOR
6027	0001	1	/1 SECTOR
6030	0042	42	/FIRST SECTOR
6031	0001	1	/1 SECTOR
6032	0046	46	/FIRST SECTOR
6033	0001	1	/1 SECTOR
6034	7777	-1	/DONE SEEK IN UNIT 3
6035	0037	37	/FIRST SECTOR
6036	0001	1	/1 SECTOR
6037	0043	43	/FIRST SECTOR
6040	0001	1	/1 SECTOR
6041	0047	47	/FIRST SECTOR
6042	0001	1	/1 SECTOR
6043	7777	-1	/DONE SEEK OUT UNIT 3
/			
6044	0036	UUT25, 36	/FIRST SECTOR
6045	0001	1	/1 SECTOR
6046	7777	-1	/DONE SEEK IN UNIT 2
6047	0037	37	/FIRST SECTOR
6050	0001	1	/1 SECTOR
6051	7777	-1	/DONE SEEK OUT UNIT 2
/			
6052	0040	UUT16, 40	/FIRST SECTOR
6053	0001	1	/1 SECTOR
6054	0046	46	/FIRST SECTOR
6055	0001	1	/1 SECTOR
6056	7777	-1	/DONE SEEK IN UNIT 1
6057	0041	41	/FIRST SECTOR
6060	0001	1	/1 SECTOR
6061	0047	47	/FIRST SECTOR
6062	0001	1	/1 SECTOR
6063	7777	-1	/DONE SEEK OUT UNIT 1 OVER UNIT 3
/			
6064	0002	UUT07, 2	/FIRST SECTOR
6065	0001	1	/1 SECTOR
6066	0042	42	/FIRST SECTOR
6067	0001	1	/1 SECTOR
6070	0046	46	/FIRST SECTOR
6071	0001	1	/1 SECTOR
6072	7777	-1	/DONE SEEK IN UNIT 0 OVER 2,3,1
6073	0003	3	/FIRST SECTOR
6074	0001	1	/1 SECTOR
6075	0043	43	/FIRST SECTOR
6076	0001	1	/1 SECTOR
6077	0047	47	/FIRST SECTOR
6100	0001	1	/1 SECTOR
6101	7777	-1	/DONE SEEK OUT UNIT 0 OVER 2,3,1
/			
6101	ENDTB4=-1		/END OF TABLE
/			
/			
/			
/*****			

//
 // TABLE OF ENTRY POINTS INTO STTBL4 BASED ON STEP NUMBER
 //

6102	0000	OVRTB4, 0	/NO STEP 0
6103	6000	UUT01	/STEP 1
6104	6006	UUT12	/STEP 2
6105	6014	UUT23	/STEP 3
6106	6026	UUT34	/STEP 4
6107	6044	UUT25	/STEP 5
6110	6052	UUT16	/STEP 6
6111	6064	UUT07	/STEP 7

 // *****
 // TABLE OF SECTORS TO USE FOR 3 UNITS UNDER TEST
 //

 // USED FOR OVER-WRITE TEST ONLY
 //

6112	STTBL3=.		
6112	0034	UUT001, 34	/FIRST SECTOR
6113	0010	10	/8 SECTORS
6114	7777	-1	/DONE SEEK IN UNIT 0
6115	0044	44	/FIRST SECTOR
6116	0010	10	/8 SECTORS
6117	7777	-1	/DONE SEEK OUT UNIT 0
6120	0044	UUT102, 44	/FIRST SECTOR
6121	0004	4	/4 SECTORS U1/U0
6122	7777	-1	/DONE SEEK IN UNIT 1
6123	0000	0	/FIRST SECTOR
6124	0004	4	/4 SECTORS U1/U0
6125	7777	-1	/DONE SEEK OUT UNIT 1
6126	0040	UUT203, 40	/FIRST SECTOR
6127	0002	2	/2 SECTORS U2/U0
6130	0000	0	/FIRST SECTOR
6131	0002	2	/2 SECTORS U2/U1
6132	7777	-1	/DONE SEEK IN UNIT 2
6133	0042	42	/FIRST SECTOR
6134	0002	2	/2 SECTORS U2/U0
6135	0002	2	/FIRST SECTOR
6136	0002	2	/2 SECTORS U2/U1
6137	7777	-1	/DONE SEEK OUT UNIT 2
6140	0040	UUT104, 40	/FIRST SECTOR
6141	0001	1	/1 SECTOR U1/U2
6142	7777	-1	/DONE SEEK IN UNIT 1
6143	0041	41	
6144	0001	1	/1 SECTOR U1/U2
6145	7777	-1	/DONE SEEK OUT UNIT 1
6146	0002	UUT005, 2	/FIRST SECTOR

6147	0001	1	/1 SECTOR U0/U2
6150	0046	46	/FIRST SECTOR
6151	0001	1	/1 SECTOR U0/U1
6152	7777	-1	/DONE SEEK IN UNIT 0
6153	0003	3	/FIRST SECTOR
6154	0001	1	/1 SECTOR U0/U2
6155	0047	47	/FIRST SECTOR
6156	0001	1	/1 SECTOR U0/U1
6157	7777	-1	/DONE SEEK OUT UNIT 0

6157 ENDTB3=-1 /END OF TABLE

/*****

SECTOR TABLE STTBL3 ENTRY POINTS BASED ON STEP NUMBER

6160	0000	OVRTB3, 0	/NO STEP 0
6161	6112	UUT001	/STEP 1, UNIT 0
6162	6120	UUT102	/STEP 2, UNIT 1
6163	6126	UUT203	/STEP 3, UNIT 2
6164	6140	UUT104	/STEP 4, UNIT 1
6165	6146	UUT005	/STEP 5, UNIT 0

/*****

TABLE OF SECTORS FOR 2 UNITS UNDER TEST

USED ONLY FOR OVER-WRITE TEST

6166 STTBL2=.

6166	0034	UT01, 34	/FIRST SECTOR
6167	0010	10	/8 SECTORS
6170	7777	-1	/DONE SEEK IN UNIT 0
6171	0044	44	/FIRST SECTOR
6172	0010	10	/8 SECTORS
6173	7777	-1	/DONE SEEK OUT UNIT 0
/			
6174	0044	UT12, 44	/FIRST SECTOR
6175	0004	4	/4 SECTORS U1/U0
6176	7777	-1	/DONE SEEK IN UNIT 1
6177	0000	0	/FIRST SECTOR
6200	0004	4	/4 SECTORS U1/U0
6201	7777	-1	/DONE SEEK OUT UNIT 1
/			
6202	0046	UT03, 46	/FIRST SECTOR
6203	0001	1	/1 SECTOR U0/U1
6204	7777	-1	/DONE SEEK IN UNIT 0
6205	0047	47	/FIRST SECTOR
6206	0001	1	/1 SECTOR U0/U1

```

6207 7777      -1      /DONE SEEK IN UNIT 0
6207 ENDTB2=-1    /END OF TABLE
/
/*****
/
/      TABLE OF SECTOR TABLE STBL2 ENTRY POINTS BASED ON STEP NUMBER
/
6210 0000      OVRTB2, 0      /NO STEP 0
6211 6166      UT01      /STEP 1 UNIT 0
6212 6174      UT12      /STEP 2 UNIT 1
6213 6202      UT03      /STEP 3 UNIT 0
/
/*****
/
/      TABLE OF ADDRESSES OF ENTRY POINT TABLES
/
6214 0000      OVRTBL, 0      /CANNOT HAVE 0 UNITS
6215 0000      0      /CANNOT HAVE 1 UNIT
6216 6210      OVRTB2      /ADDRESS OF 2 UNITS SECTOR ENTRY POINT TABLE
6217 6160      OVRTB3      /ADDRESS OF 3 UNITS SECTOR ENTRY POINT TABLE
6220 6102      OVRTB4      /ADDRESS OF 4 UNITS SECTOR ENTRY POINT TABLE
/
/*****
/
/SECTOR DATA MAP TABLE FOR OVER-WRITE TEST
/
6221 STBL16=.
/
6221 0000      SEC34, 0
6222 0000      SEC35, 0
6223 0000      SEC36, 0
6224 0000      SEC37, 0
6225 0000      SEC40, 0
6226 0000      SEC41, 0
6227 0000      SEC42, 0
6230 0000      SEC43, 0
6231 0000      SEC44, 0
6232 0000      SEC45, 0
6233 0000      SEC46, 0
6234 0000      SEC47, 0
6235 0000      SEC0, 0
6236 0000      SEC1, 0
6237 0000      SEC2, 0
6240 0000      SEC3, 0
/
/

```


/THIS TABLE IS PRESET TO 0 DURING THE FLD 1 PROGRAM INITIALIZATION

/IF A STARTING STEP OTHER THAN 1 OR 2 IS SELECTED, THIS TABLE WILL BE
/INITIALIZED AT THE START OF TESTS, TO THE PROPER VALUE ASSUMING THE
/PREVIOUS STEP HAD BEEN COMPLETED ON ANOTHER SYSTEM, AND THAT THE PACK
/BEING USED CONTAINS THE DATA REFLECTED BY THIS TABLE.
/EACH ENTRY IN THIS TABLE REPRESENTS THE DATA PATTERN # FOR EACH OF
/THE 16 SECTORS (34-47 AND 0-3) USED IN THE OVER-WRITE TEST.
/THIS TABLE IS UPDATED EACH TIME A SECTOR IS WRITTEN, AND
/IS USED TO INDICATE THE DATA PATTERN NUMBER EACH OF THE 16 SECTORS
/HAS BEEN WRITTEN WITH AS THE OVER-WRITE TEST PROGRESSES.

/

/

/*****

/

/TABLES USED FOR INITIALIZATION OF STBL16

/

6241	0000	STP124, 0	/STEP 1 OR 2 FOR 2,3, OR 4 UNITS
6242	0000	0	
6243	0000	0	
6244	0000	0	
6245	0000	0	
6246	0000	0	
6247	0000	0	
6250	0000	0	
6251	0000	0	
6252	0000	0	
6253	0000	0	
6254	0000	0	
6255	0000	0	
6256	0000	0	
6257	0000	0	
6260	0000	0	

/

/

6261	0000	STP3U4, 0	/STEP 3 FOR 2,3,OR 4 UNITS
6262	0000	0	
6263	0000	0	
6264	0000	0	
6265	0000	0	
6266	0000	0	
6267	0000	0	
6270	0000	0	
6271	0001	1	
6272	0001	1	
6273	0001	1	
6274	0001	1	
6275	0001	1	
6276	0001	1	
6277	0001	1	
6300	0001	1	

/

6301	0000	STP4U4, 0	/STEP 4 FOR 3 OR 4 UNITS
6302	0000	0	
6303	0000	0	

6304	0000	0
6305	0002	2
6306	0002	2
6307	0002	2
6310	0002	2
6311	0001	1
6312	0001	1
6313	0001	1
6314	0001	1
6315	0002	2
6316	0002	2
6317	0002	2
6320	0002	2

6321	0000	0	STP5U4, 0	/STEP 5 FOR 4 UNITS
6322	0000	0		
6323	0003	3		
6324	0003	3		
6325	0002	2		
6326	0002	2		
6327	0003	3		
6330	0003	3		
6331	0001	1		
6332	0001	1		
6333	0003	3		
6334	0003	3		
6335	0002	2		
6336	0002	2		
6337	0002	2		
6340	0002	2		

6341	0000	0	STP6U4, 0	/STEP 6 FOR 4 UNITS
6342	0000	0		
6343	0002	2		
6344	0002	2		
6345	0002	2		
6346	0002	2		
6347	0003	3		
6350	0003	3		
6351	0001	1		
6352	0001	1		
6353	0003	3		
6354	0003	3		
6355	0002	2		
6356	0002	2		
6357	0002	2		
6360	0002	2		

6361	0000	0	STP7U4, 0	/STEP 7 FOR 4 UNITS
6362	0000	0		
6363	0002	2		
6364	0002	2		
6365	0001	1		
6366	0001	1		
6367	0003	3		

6370	0003	3	
6371	0001	1	
6372	0001	1	
6373	0001	1	
6374	0001	1	
6375	0002	2	
6376	0002	2	
6377	0002	2	
6400	0002	2	
/			
6401	0000	STP5U3, 0	/STEP 5 FOR 3 UNITS
6402	0000	0	
6403	0000	0	
6404	0000	0	
6405	0001	1	
6406	0001	1	
6407	0002	2	
6410	0002	2	
6411	0001	1	
6412	0001	1	
6413	0001	1	
6414	0001	1	
6415	0002	2	
6416	0002	2	
6417	0002	2	
6420	0002	2	
/			
6421		SINDX4=.	/4 UNITS INDEX TABLE
6421	0000	0	/NO STEP 0
6422	6241	STP124	/STEP 1 ALL ZEROES
6423	6241	STP124	/STEP 2 ALL ZEROES
6424	6261	STP3U4	/STEP 3
6425	6301	STP4U4	/STEP 4
6426	6321	STP5U4	/STEP 5
6427	6341	STP6U4	/STEP 6
6430	6361	STP7U4	/STEP 7
/			
6431		SINDX3=.	/3 UNITS INDEX TABLE
6431	0000	0	/NO STEP 0
6432	6241	STP124	/STEP 1 ALL ZEROES
6433	6241	STP124	/STEP 2 ALL ZEROES
6434	6261	STP3U4	/STEP 3
6435	6301	STP4U4	/STEP 4
6436	6401	STP5U3	/STEP 5
/			
6437		SINDX2=.	/2 UNITS INDEX TABLE
6437	0000	0	/NO STEP 0
6440	6241	STP124	/STEP 1 ALL ZEROES
6441	6241	STP124	/STEP 2 ALL ZEROES
6442	6261	STP3U4	/STEP 3
/			
6443		STPIDX=.	
6443	0000	0	/0 UNITS NOT ALLOWED
6444	0000	0	/1 UNIT NOT ALLOWED
6445	6437	SINDX2	/2 UNITS

6446 6431
6447 6421

SINDX3 /3 UNITS
SINDX4 /4 UNITS

/

/TABLE OF STARTING SECTORS AND NUMBER OF SECTORS TO WRITE DURING
/ADJACENT CYLINDER TEST

6450 ADJTBL=.

/ S T E P 1
/*****

6450 0014 IN0, 14 /12 SECTORS, TRACK N, SEEK IN
6451 0004 4 /STARTING SECTOR
/

6452 0014 ON0, 14 /12 SECTORS, TRACK N, SEEK-OUT
6453 0020 20 /STARTING SECTOR
/

6454 7777 -1 /END OF UNIT 0 INITIALIZE OF TRACK N

/ S T E P 2
/*****

6455 0004 INM11, 4 /1 ADJ. TO 0, TRACK (N-1), UNIT 1, SEEK-IN
6456 0012 12 /STARTING SECTOR
/

6457 0004 ONM11, 4 /1 ADJ. TO 0, SEEK OUT, 4 SECTORS
6460 0016 16 /STARTING SECTOR
/

6461 4000 4000 /CHANGE TRACKS INDICATOR
/

6462 0004 INP11, 4 /TRACK N+1, UNIT 1, SEEK IN, 1 ADJ. TO 0, 4 SECTORS
6463 0046 46 /STARTING SECTOR
/

6464 0004 ONP11, 4 /4 SECTORS, SEEK-OUT
6465 0002 2 /STARTING SECTOR
/

6466 7777 -1 /END OF UNIT 1

/ S T E P 3
/*****

6467 0004 INM12, 4 /2 ADJ. TO 0, TRACK N-1, UNIT 2, SEEK-IN
6470 0022 22 /STARTING SECTOR
/

6471 0004 ONM12, 4 /4 SECTORS
6472 0026 26 /STARTING SECTOR
/

```

6473 4000          4000 /TRACK CHANGE INDICATOR
/
6474 0004 INP12, 4 /4 SECTORS, TRACK N+1, 2 ADJ. TO 0, SEEK-IN
6475 0006      6 /STARTING SECTOR
/
6476 0004 ONP12, 4 /4 SECTORS
6477 0012      12 /STARTING SECTOR
/
6500 4000          4000 /TRACK CHANGE INDICATOR
/
6501 0001 INM221, 1 /2 ADJ. TO 1, TRACK N-2, UNIT 2, SEEK-IN
6502 0020      20 /STARTING SECTOR
/
6503 0001 ONM221, 1 /TRACK N-2, 2 ADJ. TO 1, SEEK-OUT
6504 0021      21 /STARTING SECTOR
/
6505 4000          4000 /TRACK CHANGE INDICATOR
/
6506 0001 INP221, 1 /1 SECTOR, TRACK N+2, 2 ADJ. TO 1, SEEK-IN
6507 0040      40 /STARTING SECTOR
/
6510 0001 ONP221, 1 /1 SECTOR, SEEK-OUT
6511 0041      41 /STARTING SECTOR
/
6512 7777          -1 /END OF UNIT 2
/
/*****
/      S T E P  4
/*****
/
6513 0004 INM13, 4 /3 ADJ. TO 0, TRACK N-1, SEEK-IN
6514 0032      32 /STARTING SECTOR
/
6515 0004 ONM13, 4 /SEEK-OUT, TRACK N-1, 3 ADJ. TO 0, 4 SECTORS
6516 0036      36 /STARTING SECTOR
/
6517 4000          4000 /TRACK CHANGE INDICATOR
/
6520 0004 INP13, 4 /4 SECTORS
6521 0016      16 /STARTING SECTOR
/
6522 0004 ONP13, 4 /4 SECTORS
6523 0022      22 /STARTING SECTOR
/
6524 4000          4000 /TRACK CHANGE INDICATOR
/
6525 0001 INM231, 1 /1 SECTOR, TRACK N-2, 3 ADJ. TO 1, SEEK-IN
6526 0022      22 /STARTING SECTOR
/
6527 0001 ONM231, 1 /1 SECTOR, SEEK-OUT
6530 0023      23 /STARTING SECTOR
/
6531 0001 INM232, 1 /1 SECTOR, TRACK N-2, 3 ADJ. TO 2
6532 0030      30 /STARTING SECTOR
/

```

```

6533 0001 ONM232, 1 /1 SECTOR, SEEK-OUT
6534 0031 31 /STARTING SECTOR
/
6535 4000 4000 /TRACK CHANGE INDICATOR
/
6536 0001 INP231, 1 /3 ADJ. TO 1, TRACK N+2, 1 SECTOR, SEEK-IN
6537 0042 42 /STARTING SECTOR
/
6540 0001 ONP231, 1 /SEEK-OUT, 3 ADJ. TO 1, 1 SECTOR, TRACK N+2
6541 0043 43 /STARTING SECTOR
/
6542 0001 INP232, 1 /3 ADJ. TO 2, 1 SECTOR, SEEK-IN, TRACK N+2
6543 0000 0 /STARTING SECTOR
/
6544 0001 ONP232, 1 /1 SECTOR, SEEK-OUT
6545 0001 1 /STARTING SECTOR
/
6546 7777 -1 /END OF UNIT 3
/
/*****
/ S T E P 5
/*****
/
6547 0001 INM222, 1 /2 ADJ. 3, TRACK N-2, 1 SECTOR, SEEK-IN
6550 0040 40 /STARTING SECTOR
/
6551 0001 ONM222, 1 /SEEK-OUT, TRACK N-2, 2 ADJ. TO 3, 1 SECTOR
6552 0041 41 /STARTING SECTOR
/
6553 4000 4000 /TRACK CHANGE INDICATOR
/
6554 0001 INP222, 1 /1 SECTOR, TRACK N+2, 2 ADJ. TO 3
6555 0010 10 /STARTING SECTOR
/
6556 0001 ONP222, 1 /1 SECTOR, SEEK-OUT
6557 0011 11 /STARTING SECTOR
/
6560 7777 -1 /END OF UNIT 2
/
/*****
/ S T E P 6
/*****
/
6561 0001 INM211, 1 /1 ADJ. TO 2, TRACK N-2, 1 SECTOR, SEEK-IN
6562 0032 32 /STARTING SECTOR
/
6563 0001 ONM211, 1 /SEEK-OUT, TRACK N-2, 1 ADJ. TO 2, 1 SECTOR
6564 0033 33 /STARTING SECTOR
/
6565 7777 -1 /END OF SUB-TEST
/
6566 0001 INM212, 1 /1 SECTOR, 1 ADJ. TO 3
6567 0044 44 /STARTING SECTOR
/
6570 0001 ONM212, 1 /1 SECTOR

```

```

6571 0045      45      /STARTING SECTOR
6572 7777      -1      /END OF SUB-TEST
6573 0001      INP211, 1  /1 SECTOR, 1 ADJ. TO 2, SEEK-IN, TRACK N+2
6574 0002      2      /STARTING SECTOR
6575 0001      ONP211, 1  /1 SECTOR, SEEK-OUT
6576 0003      3      /STARTING SECTOR
6577 7777      -1      /END OF SUB-TEST
6600 0001      INP212, 1  /1 SECTOR, 1 ADJ. TO 3
6601 0012      12     /STARTING SECTOR
6602 0001      ONP212, 1  /1 SECTOR
6603 0013      13     /STARTING SECTOR
6604 7777      -1      /END OF UNIT 1

/
/*****
/      S T E P  7
/*****
/
6605 0001      INM201, 1  /0 ADJ. TO 1, TRACK N-2, SEEK-IN, 1 SECTOR
6606 0024      24     /STARTING SECTOR
6607 0001      ONM201, 1  /1 SECTOR, SEEK-OUT
6610 0025      25     /STARTING SECTOR
6611 7777      -1      /END OF SUB-TEST
6612 0001      INM202, 1  /0 ADJ. TO 2, SEEK-IN, TRACK N-2, 1 SECTOR
6613 0034      34     /STARTING SECTOR
6614 0001      ONM202, 1  /1 SECTOR, SEEK-OUT
6615 0035      35     /STARTING SECTOR
6616 7777      -1      /END OF SUB-TEST
6617 0001      INM203, 1  /1 SECTOR, SEEK-IN, TRACK N-2, 0 ADJ. TO 3
6620 0044      44     /STARTING SECTOR
6621 0001      ONM203, 1  /0 ADJ. TO 1, TRACK N-2, SEEK-OUT, 1 SECTOR
6622 0045      45     /STARTING SECTOR
6623 7777      -1      /END OF SUB-TEST
6624 0001      INP201, 1  /0 ADJ. TO 1, TRACK N+2, 1 SECTOR, SEEK-IN
6625 0044      44     /STARTING SECTOR
6626 0001      ONP201, 1  /SEEK-OUT, TRACK N+2, 1 SECTOR
6627 0045      45     /STARTING SECTOR

```

```

6630 7777      -1      /END OF SUB-TEST
/
6631 0001      INP202, 1      /0 ADJ. TO 2, 1 SECTOR, SEEK-IN, TRACK N+2
6632 0004      4      /STARTING SECTOR
/
6633 0001      ONP202, 1      /1 SECTOR, SEEK-OUT
6634 0005      5      /STARTING SECTOR
/
6635 7777      -1      /END OF SUB-TEST
/
6636 0001      INP203, 1      /0 ADJ. TO 3
6637 0014      14      /STARTING SECTOR
/
6640 0001      ONP203, 1      /SEEK-OUT, 0 ADJ. TO 3, 1 SECTOR, TRACK N+2
6641 0015      15      /STARTING SECTOR
/
6642 7777      -1      /END UNIT 0, END OF TEST
/
/*****
/
/TABLE USED TO INDEX INTO ADJTBL IF 4 UNITS UNDER TEST
/THIS IS THE SUB-STEP ADDRESS TABLE, THE ENTRY POINT OF
/WHICH IS DETERMINED BY INDEXING INTO S4TBL USING THE STEP NUMBER
/
/
6643      AJ4TBL=.
/
/ 4 DRIVES
/
6643 6450      STP14, INO      /UNIT 0
6644 7777      -1      /END OF 5 TRACKS
6645 6455      STP24, INM11    /1 ADJ. TO 0
6646 7777      -1      /END OF 5 TRACKS
6647 6467      STP34, INM12    /2 ADJ. 0 AND 1
6650 7777      -1      /END OF 5 TRACKS
6651 6513      STP44, INM13    /3 ADJ. TO 0,1,2
6652 7777      -1
6653 6547      STP54, INM222    /2 ADJ. TO 3
6654 7777      -1      /END OF 5 TRACKS
6655 6561      STP64, INM211    /1 ADJ. TO 2
6656 6566      INM212    /1 ADJ. TO 3
6657 6573      INP211    /1 ADJ. TO 2
6660 6600      INP212    /1 ADJ. TO 3
6661 7777      -1      /END OF 5 TRACKS
6662 6605      STP74, INM201    /0 ADJ. TO 1
6663 6612      INM202    /0 ADJ. TO 2
6664 6617      INM203    /0 ADJ. TO 3
6665 6624      INP201    /0 ADJ. TO 1 AND 2
6666 6631      INP202    /0 ADJ. TO 2
6667 6636      INP203    /0 ADJ. TO 3
6670 7777      -1      /END OF 5 TRACKS
/
/
/TABLE USED TO INDEX INTO AJ4TBL BASED ON STEP #
/

```



```

6671 S4TBL=.
/
/4 DRIVES
/
6671 0000 0 /NO STEP 0
6672 6643 STP14 /STEP 1. 4 UNITS
6673 6645 STP24 /STEP 2
6674 6647 STP34 /STEP 3
6675 6651 STP44 /STEP 4
6676 6653 STP54 /STEP 5
6677 6655 STP64 /STEP 6
6700 6662 STP74 /STEP 7
/
/
/*****
/
/
/SUB-STEP TABLE FOR 3 UNITS UNDER TEST
/
/
6701 AJ3TBL=. /3 DRIVES
/
/
6701 6450 STP13, IN0 /UNIT 0
6702 7777 -1 /END OF 5 TRACKS
6703 6455 STP23, INM11 /1 ADJ. TO 0
6704 7777 -1 /END OF 5 TRACKS
6705 6467 STP33, INM12 /2 ADJ. TO 0 AND 1
6706 7777 -1 /END OF 5 TRACKS
/
6707 6561 STP43, INM211 /1 ADJ. TO 2
6710 6573 INP211 /1 ADJ. TO 2
6711 7777 -1 /END OF 5 TRACKS
/
6712 6605 STP53, INM201 /0 ADJ. TO 1
6713 6612 INM202 /0 ADJ. TO 2
6714 6624 INP201 /0 ADJ. TO 1
6715 6631 INP202 /0 ADJ. TO 2
6716 7777 -1 /END OF 5 TRACKS
/
/
/TABLE USED TO INDEX INTO AJ3TBL BASED ON STEP #
/
/
6717 S3TBL=. /3 UNITS
/
6717 0000 0 /NO STEP 0
6720 6701 STP13 /STEP 1
6721 6703 STP23 /STEP 2
6722 6705 STP33 /STEP 3
6723 6707 STP43 /STEP 4
6724 6712 STP53 /STEP 5
/
/
/*****

```

```

/
/
/SUB-STEP TABLE FOR 2 UNITS UNDER TEST
/
6725 AJ2TBL=. /2 DRIVES
/
6725 6450 STP12, IN0 /UNIT 0
6726 7777 -1 /END OF 5 TRACKS
6727 6455 STP22, INM11 /1 ADJ. TO 0
6730 7777 -1 /END OF 5 TRACKS
6731 6605 STP32, INM201 /0 ADJ. TO 1
6732 6624 INP201 /0 ADJ. TO 1
6733 7777 -1 /END OF 5 TRACKS
/
/
/TABLE USED TO INDEX INTO AJ2TBL BASED ON STEP NUMBER
/
/
6734 S2TBL=. /2 UNITS
/
6734 0000 0 /NO STEP 0
6735 6725 STP12 /STEP 1
6736 6727 STP22 /STEP 2
6737 6731 STP32 /STEP 3
/
/*****
/
/
/TABLE USED TO LOCATE THE PROPER STEP INDEX TABLE BY USING
/THE NUMBER OF UNITS UNDER TEST
/
/
6740 STBL=.
/
6740 0000 0 /0 UNITS NOT VALID
6741 0000 0 /1 UNIT NOT VALID
6742 6734 S2TBL /2 UNITS STEP INDEX TABLE ADDRESS
6743 6717 S3TBL /3 UNITS STEP INDEX TABLE ADDRESS
6744 6671 S4TBL /4 UNITS STEP INDEX TABLE ADDRESS
/
/*****
/
/
/TABLE USED TO GET ADJACENT UNIT NUMBER (DATA PATTERN NUMBER)
/
/IF THE TRACK OFFSET IS 1 OR 3 ( TRACK N-1 OR N+1 ) THEN THE
/ADJACENT TRACK IS TRACK N AND THE UNIT NUMBER IS ALWAYS 0 ON TRACK N.
/IF THE TRACK OFFSET IS 0 OR 4 ( TRACK N-2 OR N+2 ) THEN THE
/ADJACENT TRACK IS TRACK N-1 OR N+1 AND THE UNIT NUMBER IS DETERMINED
/BY INDEXING INTO ADJU4 TABLE AND USING THE 6 RIGHTMOST BITS (LSB'S)
/OF THE TABLE ENTRY AS THE ADJACENT UNIT NUMBER. THE FOLLOWING
/ALGORITHMS ARE USED TO INDEX INTO THIS TABLE:
/
/ IF TRACK OFFSET = 0 (TRACK N-2) THEN:
/

```

```

/      ( ADJU4 + WSEC ) - 20   WILL POINT TO THE ADJACENT SECTORS UNIT NUMBER
/
/
/      IF TRACK OFFSET = 4 (TRACK N+2) THEN:
/
/      ( ADJU4 + WSEC ) -40   WILL POINT TO THE ADJACENT SECTORS UNIT NUMBER IF
/                               WSEC > 17
/
/      ( ADJU4 + WSEC ) + 10   WILL POINT TO THE ADJACENT SECTORS UNIT NUMBER IF
/                               WSEC < OR = 17
/
/
6745      ADJU4=.
/
6745 0201 0201 /UNIT 2 ADJ TO 1   TRACK N-2 ADJ TO TRACK N-1 OR N+2 ADJ TO N+1
6746 0201 0201 /SAME
6747 0301 0301 /UNIT 3 ADJ TO 1   TRACK N-2 ADJ TO TRACK N-1 OR N+2 ADJ TO N+1
6750 0301 0301 /SAME
6751 0001 0001 /UNIT 0 ADJ TO 1
6752 0001 0001 /UNIT 0 ADJ TO 1
6753 7701 7701 /THIS SECTOR NOT USED ON TRACK N+2 OR N-2
6754 7701 7701 /SAME
/
6755 0302 0302 /UNIT 3 ADJ TO 2
6756 0302 0302 /SAME
6757 0102 0102 /UNIT 1 ADJ TO 2
6760 0102 0102 /SAME
6761 0002 0002 /UNIT 0 ADJ TO 2
6762 0002 0002 /SAME
6763 7702 7702 /THIS SECTOR NOT USED ON TRACK N-2 OR N+2
6764 7702 7702 /SAME
/
6765 0203 0203 /UNIT 2 ADJ TO 3
6766 0203 0203 /SAME
6767 0103 0103 /UNIT 1 ADJ TO 3
6770 0103 0103 /SAME
6771 0003 0003 /UNIT 0 ADJ TO 3
6772 0003 0003 /SAME
6773 7703 7703 /THIS SECTOR NOT USED ON TRACK N-2 OR N+2
6774 7703 7703 /SAME
/
/*****
/
/
/      /TABLE USED TO CALCULATE THE TRACK TO WRITE ON FOR THE ADJACENT CYLINDER
/      /WRITE TEST. THE ENTRY INTO THIS TABLE IS FOUND USING THE STEP NUMBER
/      /TO INDEX INTO TABLE WRTRK4.
/
/
6775 0002 STP041, 2      /LOWEST TRACK + 2      TRACK N
6776 0001 STP042, 1      /LOWEST TRACK + 1      TRACK N-1
6777 0003      3      /LOWEST TRACK + 3      TRACK N+1
7000 0001 STP043, 1      /LOWEST TRACK + 1      TRACK N-1
7001 0003      3
7002 0000      0

```

7003	0004	4		
7004	0001	STP044, 1	/LOWEST TRACK + 1	TRACK N-1
7005	0003	3	/N+1	
7006	0000	0	/N-2	
7007	0004	4	/N+2	
7010	0000	STP045, 0	/N-2	
7011	0004	4	/N+2	
7012	0000	STP046, 0	/N-2	
7013	0000	0	/N-2	
7014	0004	4	/N+2	
7015	0004	4	/N+2	
7016	0000	STP047, 0	/N-2	
7017	0000	0	/N-2	
7020	0000	0	/N-2	
7021	0004	4	/N+2	
7022	0004	4	/N+2	
7023	0004	4	/N+2	
7024	7777	-1	/END OF TABLE	

//
//
//TABLE USED TO INDEX INTO TRACK OFFSET TABLE ABOVE BY USING
//THE STEP NUMBER
//

7025 WRTRK4=. /4 UNITS
//

7025	0000	0	/NO STEP 0	
7026	6775	STP041	/STEP 1	
7027	6776	STP042	/STEP 2	
7030	7000	STP043	/STEP 3	
7031	7004	STP044	/STEP 4	
7032	7010	STP045	/STEP 5	
7033	7012	STP046	/STEP 6	
7034	7016	STP047	/STEP 7	

//
//
//*****
//

//TABLE USED FOR TRACK OFFSET IF 3 UNITS UNDER TEST
//

7035	0002	STP031, 2	/TRACK N	
7036	0001	STP032, 1	/N-1	
7037	0003	3	/N+1	
7040	0001	STP033, 1	/N-1	
7041	0003	3	/N+1	
7042	0000	0	/N-2	
7043	0004	4	/N+2	
7044	0000	STP034, 0	/N-2	
7045	0004	4	/N+2	
7046	0000	STP035, 0	/N-2	
7047	0000	0	/N-2	
7050	0004	4	/N+2	

7051 0004 4 /N+2 0 ADJ. TO 2

//
/TABLE USED TO INDEX INTO ABOVE TRACK OFFSET TABLE
//

7052 WRTRK3=. /3 UNITS
//

7052 0000 0 /NO STEP 0
7053 7035 STP031 /STEP 1
7054 7036 STP032 /STEP 2
7055 7040 STP033 /STEP 3
7056 7044 STP034 /STEP 4
7057 7046 STP035 /STEP 5
//

//
/*****
//

//
/2 UNITS TABLE FOR TRACK OFFSET
//

7060 0002 STP021, 2 /TRACK N
7061 0001 STP022, 1 /N-1 1 ADJ. TO 0
7062 0003 3 /N+1 1 ADJ. TO 0
7063 0000 STP023, 0 /N-2 0 ADJ. TO 1
7064 0004 4 /N+2 0 ADJ. TO 1
//

//
/TABLE USED TO INDEX INTO ABOVE TRACK OFFSET TABLE
//

7065 WRTRK2=. /2 UNITS
//

7065 0000 0 /NO STEP 0
7066 7060 STP021 /STEP 1
7067 7061 STP022 /STEP 2
7070 7063 STP023 /STEP 3
//

//
/7071 WRTRK=.
//

7071 0000 0 /0 UNITS NOT ALLOWED
7072 0000 0 /1 UNIT NOT ALLOWED
7073 7065 WRTRK2 /2 UNITS
7074 7052 WRTRK3 /3 UNITS
7075 7025 WRTRK4 /4 UNITS
//
//
//

```

/*****
/
/ROUTINE TO CHECK FOR CONSOLE PACKAGE ACTIVE
/
/IF CONSOLE PACKAGE ACTIVE, GO TO CONSOLE PACKAGE
/RETURN CALL + 2 AC CLEAR
/
/IF CONSOLE PACKAGE NOT ACTIVE, RETURN CALL + 1 AC CLEAR
/
7076 0000 XC8CAL, 0
7077 6000 SKON /SKIP IF INTERRUPT ON AND TURN OFF
7100 5305 JMP C8OFF /INT OFF
7101 3333 DCA CHRTMP /SAVE AC
7102 1110 TAD KION /GET ION INSTRUCTION
7103 3330 DCA CALOUT /SAVE ION FOR EXIT EXECUTION
7104 5310 JMP GETHCW /GO CHECK FOR CONSOLE ACTIVE
7105 3333 C8OFF, DCA CHRTMP /SAVE AC
7106 1377 TAD (IOF /GET IOF INSTRUCTION
7107 3330 DCA CALOUT /SAVE FOR EXIT EXECUTION
7110 1022 GETHCW, TAD 22 /GET HCW2
7111 0046 AND K400 /TEST FOR BIT 3=1 CONSOLE ACTIVE
7112 7650 SNA CLA /SKIP IF CONSOLE ACTIVE
7113 5330 JMP CALOUT /EXIT, RETURN CALL+1 CONSOLE NOT ACTIVE
7114 6224 RIF /READ INSTRUCTION FIELD
7115 1376 TAD (OFFSET /ADD CONSOLE PACKAGE FIELD OFFSET
7116 1375 TAD (CIF /ADD CIF INSTRUCTION CODE
7117 3320 DCA .+1 /SAVE MODIFIED CIF FOR EXECUTION
7120 7402 HLT/CIF /MODIFIED CIF TO CONSOLE PACKAGE FIELD
7121 1333 TAD CHRTMP /GET CHARACTER AGAIN
7122 4732 JMS I C8LOC /GO TO CONSOLE PACKAGE
7123 2276 ISZ XC8CAL /INCREMENT RETURN ADDRESS
7124 1020 TAD PSR /GET PSR
7125 6211 CDF 10 /CDF TO FLD 1
7126 3774 DCA I (PSR1 /COPY PSR TO FLD 1 PG 0 LOC 20
7127 6201 CDF 00 /CDF TO PRGM FLD
7130 7402 CALOUT, HLT/ION/IOF /MODIFIED ION OR IOF UPON ENTRY
7131 5676 XC8RET, JMP I XC8CAL /RETURN CALL + 2 CONSOLE WAS ACTIVE
/
7132 3422 C8LOC, C8ENTR /POINTER TO CONSOLE PACKAGE ENTRY
7133 0000 CHRTMP, 0 /TEMPORARY AC SAVE AREA
/
/
/*****
/ROUTINE TO DO A SEEK
/CALLED BY: JMS SEEK
/
/
7134 0000 SEEK, 0
7135 6002 IOF /INTERRUPTS OFF
7136 4536 SETPNT /SET TBLPNT TO COMMAND B
7137 0023 XCOMB /TABLE INDEX
7140 1024 TAD CURDRV /GET CURRENT DRIVE

```

7141	7002	BSW		/MOVE DRIVE SELECT BITS TO BITS 4,5
7142	1373	TAD	(403	/SET INTERRUPT ENABLE , CODE 3 FOR SEEK
7143	3504	DCA I	TBLPNT	/SAVE COMMAND B
7144	4536	SETPNT		/SET TBLPNT TO CURRENT CYL AND HEAD
7145	0001	CURCYL		/TABLE INDEX
7146	1504	TAD I	TBLPNT	/GET CURRENT CYL AND HEAD
7147	4536	SETPNT		/SET TBLPNT TO OLD CYL ADDR STORAGE
7150	0000	OLDCYL		/TABLE INDEX
7151	3504	DCA I	TBLPNT	/SAVE CURRENT CYL AND HEAD
7152	4536	SETPNT		/SET TBLPNT TO SECTOR ADDR
7153	0021	SECADD		/TABLE INDEX
7154	3504	DCA I	TBLPNT	/CLEAR SECTOR ADDR
7155	4536	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR ADDRESS
7156	0024	XENDSC		/TBL INDX
7157	3504	DCA I	TBLPNT	/EXPECTED FINAL SECTOR ADDRESS = 0
7160	4543	GO		/ISSUE COMMAND
7161	5734	JMP I	SEEK	/EXIT

/ROUTINE TO CALL GET ONE OCTAL DIGIT ROUTINE IN FLD 1

 CALLED BY: GETOCT

7162	0000	OCTGET, 0		
7163	6212	CIF	10	/INS FLD = 1
7164	4766	JMS I	OCTGO	/GO TO INPUT ONE OCTAL DIGIT FLD 1
7165	5762	JMP I	OCTGET	/RETURN

7166	1064	OCTGO, XOCT1		/ADDRESS OF SUBROUTINE
------	------	--------------	--	------------------------

7173	0403
7174	0020
7175	6202
7176	0010
7177	6002
	7200

PAGE

```

/*****
/
/ROUTINE TO DO A <CR> AND <LF>
/
/      CALLED BY:      DOCRLF
/
7200 0000  CRLFDO, 0
7201 6212      CIF      10      /INS FLD = 1
7202 4777      JMS I    (XCRLF  /GO TO FLD 1
7203 5600      JMP I    CRLFDO  /RETURN
/
/*****
/
/ROUTINE TO WAIT FOR DRIVE READY AFTER SEEK
/
/      CALLED BY:      JMS      WATRDY
/
/RETURN CALL+1 IF READY
/RETURN CALL+2 IF NOT READY
/
7204 0000  WATRDY, 0
7205 7300      CLA CLL
7206 3226      DCA      WATCNT      /CLEAR DRIVE READY WAIT COUNTER
7207 1376      TAD      (-12      /GET TIMES 10 MULTIPLIER
7210 3227      DCA      WATCT1     /SET UP TIMES 10 COUNTER
7211 6610  WATLPA, RRER      /READ ERROR REGISTER
7212 7410      SKP      /IOT SHOULD NOT SKIP
7213 4336      JMS      FATAL      /IOT RRER SKIPPED
7214 7110      CLL RAR      /PUT DRIVE READY BIT IN LINK
7215 7620      SNL      CLA      /SKIP IF DRIVE READY
7216 5220      JMP      WATMOR     /DRIVE NOT READY, CONTINUE WAITING
7217 5604      JMP I    WATRDY     /DRIVE READY RETURN
/
7220 2226  WATMOR, ISZ      WATCNT      /INCREMENT DRIVE READY WAIT COUNTER
7221 5211      JMP      WATLPA      /CONTINUE WAITING FOR DRIVE READY
7222 2227      ISZ      WATCT1     /INCREMENT DRIVE READY WAIT COUNTER TIMES 10
7223 5211      JMP      WATLPA      /CONTINUE WAITING FOR DRIVE READY
7224 2204      ISZ      WATRDY     /INCREMENT RETURN FOR DRIVE NOT READY TIME-OUT
7225 5604      JMP I    WATRDY     /DRIVE NOT READY TIME-OUT RETURN
/
7226 0000  WATCNT, 0      /DRIVE READY WAIT COUNTER
7227 0000  WATCT1, 0     /TIMES 10 MULTIPLIER COUNTER
/
/*****
/

```



```

/
/
/ROUTINE TO PRINT BUFFER ADDRESS, GOOD DATA WORD, BAD DATA WORD
/FOR SECTOR DATA ERROR IF DATA CRC ERROR.
/IF ERROR MSG NOT INHIBITED, PRINT BUFFER ADDRESS,
/GOOD DATA, BAD DATA EACH TIME ROUTINE CALLED
/ENTER WITH BAD DATA WORD ADDRESS IN AUTO15
/ENTER WITH GOOD DATA WORD ADDRESS IN AUTO14
/RETURN CALL+1
/
/      CALLED BY:      JMS      DERR
/
/
7230 0000      DERR,      0
7231 2317      ISZ      BDWCNT      /INCREMENT BAD WORD COUNT
7232 1117      TAD      DERFLG      /GET DATA ERROR FLAG
7233 7640      SZA      CLA      /SKIP IF THIS IS THE FIRST BAD WORD
7234 5261      JMP      PRNBA      /GO PRINT BUFFER ADDR, NOT FIRST BAD WRD
7235 7040      CMA      /AC=7777
7236 3117      DCA      DERFLG      /SET DATA ERROR FLG TO 7777
7237 4556      DDCRLF      /<CR LF>
7240 4551      MESAG
7241 2371      DATMSG      /"DATA"
7242 4551      MESAG
7243 2476      ERR      /" ERROR"
7244 4556      DDCRLF      /<CR LF>
7245 4540      STAPRT      /PRINT THE DRIVE NUMBER
7246 2040      ERTBL4      /START OF MSG ADDRESS
7247 4775      JMS      LOGERR      /LOG DATA ERROR
7250 0007      DATERR      /TBL INDX
7251 4556      DDCRLF      /<CRLF>
7252 4537      GETSWR      /GET SWITCHES
7253 0043      AND      K100      /MASK SR5
7254 7640      SZA      CLA      /SR5=0 PRINT DATA ERROR
7255 5630      JMP I      DERR      /DATA ERROR REPORT INHIBITED, RETURN
7256 4551      MESAG
7257 2374      BAGBM      /"BA:      GOOD      BAD"
7260 4556      DDCRLF      /<CR LF>
7261 4537      PRNBA,      GETSWR      /GET SWITCHES
7262 0043      AND      K100      /MASK SR5
7263 7640      SZA      CLA      /SR5=0 PRINT DATA ERROR
7264 5630      JMP I      DERR      /DATA ERROR REPORT INHIBITED, RETURN
7265 7240      STA      /SET THE AC TO ALL ONES (7777)      HP 005
7266 3045      DCA      INMODE      /SET THE MESSAGE ACTIVE FLAG,SO CNTRL S AND Q      HP 005
/
7267 1374      TAD      (261      /BUFFER FLD 1 TO PRINT
7270 4553      PRNT      /PRINT "1"
7271 1015      TAD      AUTO15      /GET BUFFER ADDRESS OF BAD DATA
7272 4555      PRNAC      /PRINT BUFFER ADDRESS
7273 4557      SPACE2      /PRINT 2 SPACES
7274 1015      TAD      AUTO15      /GET BUFFER ADDRESS POINTER
7275 1052      TAD      M1      /SUBTRACT 1 FOR AUTO INDX
7276 3015      DCA      AUTO15      /SAVE BUFFER ADDRESS POINTER
7277 1773      TAD      DATSWT      /GET FIRST OR SECOND DATA WORD SWITCH
7300 7640      SZA      CLA      /SKIP IF FIRST DATA WORD OF PATTERN

```

```

7301 1772' TAD DATWD2 /GET SECOND DATA WORD OF PATTERN
7302 7450 SNA /SKIP IF SECOND DATA WORD
7303 1771' TAD DATWD1 /GET FIRST DATA WORD OF PATTERN
7304 4555 PRNAC /PRINT GOOD DATA WORD
7305 4557 SPACE2 /PRINT 2 SPACES

/
7306 6211 CDF 10
7307 1415 TAD I AUTO15 /GET BAD DATA WORD
7310 0045 AND K377 /MASK FOR UNUSED BITS
7311 6201 CDF 00
7312 4555 PRNAC /PRINT BAD DATA WORD
7313 4556 DDCRLF /DO A <CR> <LF>
7314 3045 DCA INMODE /CLEAR MESSAGE ACTIVE FLAG
7315 5630 JMP I DERR /RETURN

7316 0000 BFPT, 0 /BUFFER TABLE POINTER
7317 0000 BDCNT, 0

/
/*****
/
/ROUTINE TO CLEAR BUFFER 1 FIELD 1
/
/CALLED BY: JMS CLRBF1
/
7320 0000 CLRBF1, 0
7321 1370 TAD (BUFSZ1 /GET BUFFER 1 SIZE
7322 7041 CIA
7323 3335 DCA BUFCTR /SET UP BUFFER LOCATION COUNTER
7324 1106 TAD BUFAD1 /GET START OF BUFFER 1 ADDRESS
7325 1052 TAD M1 /SUB ONE FOR AUTO-INDEX
7326 3012 DCA AUTO12 /SET UP BUFFER POINTER
7327 6211 CDF 10 /CDF TO BUFFER 1 FLD 1
7330 3412 CLRLP1, DCA I AUTO12 /CLEAR A BUFFER 1 LOCATION
7331 2335 ISZ BUFCTR /INCREMENT BUFFER LOCATION COUNTER
7332 5330 JMP CLRLP1 /CONTINUE CLEARING BUFFER 1
7333 6201 CDF 00 /CDF TO PRGM FLD
7334 5720 JMP I CLRBF1 /RETURN

/
7335 0000 BUFCTR, 0 /BUFFER LOCATION COUNTER
/
/*****
/
/ROUTINE TO HANDLE FATAL ERRORS
/
/ CALLED BY: JMS FATAL
/
/
/
7336 0000 FATAL, 0
7337 6002 IOF /INTERRUPT OFF
7340 3364 DCA ASAV /SAVE AC
7341 4551 MESAG /PRINT MESSAGE
7342 2361 FATL /"FATAL"
7343 4551 MESAG

```

HP 005

```

7344 2476      ERR      /* ERROR"
7345 4556      DOCRLF    /*<CR LF>
7346 4551      MESAG
7347 2356      PCMSG
7350 1336      TAD      FATAL
7351 1052      TAD      M1
7352 4555      PRNAC
7353 4557      SPACE2
7354 4551      MESAG
7355 2405      ACM5G
7356 1364      TAD      ASAV
7357 4555      PRNAC
7360 4535      C8CALL
7361 7000      NOP
7362 4547      GETSTA
7363 5736      JMP I     FATAL

```

```

/*AC: "
/GET AC
/PRINT AC
/GO TO CONSOLE PACKAGE FOR SR=
/NOT ACTIVE RETURN
/GET STATUS FOR LAST OPERATION CONTINUE PROGRAM
/PROGRAM CONTINUED, GO DO NEXT DRIVE

```

```

/
/
7364 0000      ASAV,    0
/
/

```

```

7370 2000
7371 1263
7372 1264
7373 1265
7374 0261
7375 3612
7376 7766
7377 1257
7400

```

PAGE

```

/
/*****
/
/

```

D R I V E S T A T E T A B L E S

```

7400      DRVPNT=.      /*START OF INDEX TABLE INTO DRIVE STATE TABLES
/
7400 7404      DRV0,    DRIVE0      /*POINTER TO DRIVE 0 DRIVE STATE TABLE
7401 7431      DRV1,    DRIVE1      /*POINTER TO DRIVE 1 DRIVE STATE TABLE
7402 7456      DRV2,    DRIVE2      /*POINTER TO DRIVE 2 DRIVE STATE TABLE
7403 7503      DRV3,    DRIVE3      /*POINTER TO DRIVE 3 DRIVE STATE TABLE
/
/

```

```

7404      DRIVE0=.      /*START OF DRIVE 0 DRIVE STATE TABLE
/
7404 0000      OLDCYL    /*PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7405 0001      CURCYL    /*CURRENT CYLINDER ADDRESS BY READ HEADER
7406 0002      NEWCYL    /*NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7407 0003      HRDERR    /*HARD ERROR COUNT
7410 0004      SFTERR    /*SOFT ERROR COUNT
7411 0005      DRVERR    /*DRIVE ERROR COUNT
7412 0006      SEKERR    /*SEEK ERROR COUNT
7413 0007      DATERR    /*DATA ERROR COUNT

```

7414	0010	TRKERR	/TRACKING ERROR COUNT
7415	0011	DCRCER	/DATA CRC ERROR COUNT
7416	0012	HCR CER	/HEADER CRC ERROR COUNT
7417	0013	DLTERR	/DATA LATE ERROR COUNT
7420	0014	OPIERR	/OPERATION INCOMPLETE ERROR COUNT
7421	0015	HN FERR	/HEADER NOT FOUND ERROR COUNT
7422	0016	CTLERR	/CONTROLLER ERROR COUNT
7423	0017	WRDCNT	/INITIAL WORD COUNT SENT
7424	0020	INITCA	/INITIAL CURRENT DATA BUFFER ADDRESS
7425	0021	SECADD	/SECTOR ADDRESS SENT
7426	0022	XCOMA	/COMMAND REGISTER A SENT
7427	0023	XCOMB	/COMMAND REGISTER B SENT
7430	0024	XENDSC	/EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
/			
/			
7431	DRIVE1=.		/START OF DRIVE 1 DRIVE STATE TABLE
/			
7431	0000	OLD CYL	/PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7432	0001	CUR CYL	/CURRENT CYLINDER ADDRESS BY READ HEADER
7433	0002	NEW CYL	/NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7434	0003	HRDERR	/HARD ERROR COUNT
7435	0004	SFTERR	/SOFT ERROR COUNT
7436	0005	DRVERR	/DRIVE ERROR COUNT
7437	0006	SEKERR	/SEEK ERROR COUNT
7440	0007	DATERR	/DATA ERROR COUNT
7441	0010	TRKERR	/TRACKING ERROR COUNT
7442	0011	DCRCER	/DATA CRC ERROR COUNT
7443	0012	HCR CER	/HEADER CRC ERROR COUNT
7444	0013	DLTERR	/DATA LATE ERROR COUNT
7445	0014	OPIERR	/OPERATION INCOMPLETE ERROR COUNT
7446	0015	HN FERR	/HEADER NOT FOUND ERROR COUNT
7447	0016	CTLERR	/CONTROLLER ERROR COUNT
7450	0017	WRDCNT	/INITIAL WORD COUNT SENT
7451	0020	INITCA	/INITIAL CURRENT DATA BUFFER ADDRESS
7452	0021	SECADD	/SECTOR ADDRESS SENT
7453	0022	XCOMA	/COMMAND REGISTER A SENT
7454	0023	XCOMB	/COMMAND REGISTER B SENT
7455	0024	XENDSC	/EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
/			
/			
7456	DRIVE2=.		/START OF DRIVE 2 STATE TABLE
/			
7456	0000	OLD CYL	/PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7457	0001	CUR CYL	/CURRENT CYLINDER ADDRESS BY READ HEADER
7460	0002	NEW CYL	/NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7461	0003	HRDERR	/HARD ERROR COUNT
7462	0004	SFTERR	/SOFT ERROR COUNT
7463	0005	DRVERR	/DRIVE ERROR COUNT
7464	0006	SEKERR	/SEEK ERROR COUNT
7465	0007	DATERR	/DATA ERROR COUNT
7466	0010	TRKERR	/TRACKING ERROR COUNT
7467	0011	DCRCER	/DATA CRC ERROR COUNT
7470	0012	HCR CER	/HEADER CRC ERROR COUNT
7471	0013	DLTERR	/DATA LATE ERROR COUNT
7472	0014	OPIERR	/OPERATION INCOMPLETE ERROR COUNT

```

7473 0015 HNFERR /HEADER NOT FOUND ERROR COUNT
7474 0016 CTLERR /CONTROLLER ERROR COUNT
7475 0017 WRDCNT /INITIAL WORD COUNT SENT
7476 0020 INITCA /INITIAL CURRENT DATA BUFFER ADDRESS
7477 0021 SECADD /SECTOR ADDRESS SENT
7500 0022 XCOMA /COMMAND REGISTER A SENT
7501 0023 XCOMB /COMMAND REGISTER B SENT
7502 0024 XENDSC /EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
/
/
7503 DRIVE3=. /START OF DRIVE 3 DRIVE STATE TABLE
/
7503 0000 OLD CYL /PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7504 0001 CUR CYL /CURRENT CYLINDER ADDRESS BY READ HEADER
7505 0002 NEW CYL /NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7506 0003 HRDERR /HARD ERROR COUNT
7507 0004 SFTERR /SOFT ERROR COUNT
7510 0005 DRVERR /DRIVE ERROR COUNT
7511 0006 SEKERR /SEEK ERROR COUNT
7512 0007 DATERR /DATA ERROR COUNT
7513 0010 TRKERR /TRACKING ERROR COUNT
7514 0011 DCR CER /DATA CRC ERROR COUNT
7515 0012 HCR CER /HEADER CRC ERROR COUNT
7516 0013 DLTERR /DATA LATE ERROR COUNT
7517 0014 OPIERR /OPERATION INCOMPLETE ERROR COUNT
7520 0015 HNFERR /HEADER NOT FOUND ERROR COUNT
7521 0016 CTLERR /CONTROLLER ERROR COUNT
7522 0017 WRDCNT /INITIAL WORD COUNT SENT
7523 0020 INITCA /INITIAL CURRENT DATA BUFFER ADDRESS
7524 0021 SECADD /SECTOR ADDRESS SENT
7525 0022 XCOMA /COMMAND REGISTER A SENT
7526 0023 XCOMB /COMMAND REGISTER B SENT
7527 0024 XENDSC /EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
/
/
7527 DRVEND=-1 /END OF DRIVE STATE TABLES
/
/
/*****
/
/
/ D E V I C E C O D E T A B L E S
/
/
/ ADDRESSES OF ALL IOTS IN PROGRAM
/
/
7530 DCTB60=. /START OF DSC 0600 OR 0620 TABLE
/
7530 3011 CNTLOD /LOAD WORD COUNT IN ROUTINE "XGO"
7531 3017 BRKLOD /LOAD BRKMA IN ROUTINE "XGO"
7532 3026 SECLD /LOAD SECTOR ADDR IN ROUTINE "XGO"
7533 3034 ALODE /LOAD COMMAND A REG. IN ROUTINE "XGO"
7534 3042 BLODE /LOAD COMMAND B REG. IN ROUTINE "XGO"

```

```

7535 2033      FLGS      /SKIP ON DONE FLAG IN INTSVC
/
7536      DCTB61=.      /START OF DSC 0610 OR 0630 TABLE
/
7536 2036      IOTA      /SKIP ON ERROR FLG IN INTSVC
7537 2045      IOTB      /READ ERROR REG IN INTSVC
7540 2051      IOTC      /READ WC IN INTSVC
7541 2055      IOTD      /READ COM A IN INTSVC
7542 2061      IOTE      /READ COM B IN INTSVC
7543 2065      IOTF      /READ SECTOR ADR REG IN INTSVC
7544 2334      IOTG      /READ ERROR REGISTER IN RDYCHK
7545 3214      RDSI1     /READ SILO IN READ HEADER ROUTINE
7546 3217      RDSI2     /SAME
7547 3222      RDSI3     /SAME
7550 3226      RDSI4     /SAME
7551 3105      RDSIA     /READ SILO IN GET STATUS ROUTINE
7552 3112      RDSIB     /SAME
7553 7211      WATLPA    /RRER IN WATRDY ROUTINE
/
7554      DCEND=.
/
/*****
/FACTORY BAD SECTOR FILE SECTOR NUMBERS
/
7554      FACBAD=.
7554 0000      0000      /SECTOR ADDRESS 0 OF FACTORY BAD
7555 0004      0004      /SECTOR ADDRESS 4 OF FACTORY BAD
7556 0010      0010      /SECTOR ADDRESS 8 OF FACTORY BAD
7557 0014      0014      /SECTOR ADDRESS 12 OF FACTORY BAD
7560 0020      0020      /SECTOR ADDRESS 16 OF FACTORY BAD
/
/FIELD BAD SECTOR FILE SECTOR NUMBERS
/
7561      FLDBAD=.
7561 0024      0024      /SECTOR ADDRESS 20 OF FIELD BAD
7562 0030      0030      /SECTOR ADDRESS 24 OF FIELD BAD
7563 0034      0034      /SECTOR ADDRESS 28 OF FIELD BAD
7564 0040      0040      /SECTOR ADDRESS 32 OF FIELD BAD
7565 0044      0044      /SECTOR ADDRESS 36 OF FIELD BAD
/
/*****
/
0001      FIELD 1

```


0010	0000	*10	
		/	
0010	0000	INDX10, 0	/AUTO-INDEX
0011	0000	INDX11, 0	/AUTO-INDEX
0012	0000	INDX12, 0	/AUTO-INDEX
0013	0000	INDX13, 0	/AUTO-INDEX
0014	0000	INDX14, 0	/AUTO-INDEX
0015	0000	INDX15, 0	/AUTO-INDEX
0016	0000	INDX16, 0	/AUTO-INDEX
0017	0000	INDX17, 0	/AUTO-INDEX
		/	
	0020	*20	
		/	
0020	0000	PSR1, 0	/SOFTWARE SWITCH REGISTER
0021	0000	HDW1, 0	/HARDWARE CONFIG WORD 1
0022	0000	HDW2, 0	/HARDWARE CONFIG WORD 2
0023	0000	HDW3, 0	/HARDWARE CONFIG WORD 3
	0024	*24	
	4424	MESSAGE=JMS I .	/CALL TO MESSAGE PRINT ROUTINE
0024	1200	MESPRT	/FOLLOWED BY MESSAGE ADDRESS
	4425	YESRNO=JMS I .	/CALL TO YES OR NO ROUTINE
0025	1273	XYESNO	/RETURN CALL+1 IF YES, CALL+2 IF NO
	4426	OCT1=JMS I .	/CALL TO ROUTINE TO ACCEPT 1 OCTAL DIGIT INPUT
0026	1064	XOCT1	
	4427	DEC4=JMS I .	/CALL TO ROUTINE TO ACCEPT 4 DECIMAL DIGITS INPUT
0027	1000	XDEC4	/AND CONVERT THEM TO OCTAL
	4430	LIMCHK=JMS I .	/CALL TO ROUTINE TO CHECK FOR ANSWER WITHIN LIMITS
0030	1400	XLCHK	/CALL+1 = UPPER LIMIT, CALL+2 = LOWER LIMIT
			/RETURN CALL+3 IF LIMITS EXCEEDED
			/RETURN CALL+4 IF LIMITS NOT EXCEEDED
			/AC CONTAINS VALUE TO CHECK WHEN LIMCHK IS CALLED
			/RETURN WITH AC CLEAR
	4431	PRNTAC=JMS I .	/PRINT CONTENTS OF ACCUMULATOR
0031	0637	ACPRNT	
	4432	PRINT=JMS I .	/PRINT ONE ASCII CHARACTER
0032	1316	XPRINT	
0033	1121	XTFIL	/CALL+1 IS STARTING ADDRESS OF TABLE
			/CALL+2 IS NUMBER OF TABLE ENTRIES (UNSIGNED)
			/CALL+3 IS THE VALUE TO PLACE IN EACH OF THE TABLE LOCATIONS
			/CALL+4 IS RETURN WITH AC CLEAR
	4434	CRLF=JMS I .	/ROUTINE TO PRINT <CR> AND <LF>
0034	1257	XCRLF	
	4435	INPUT=JMS I .	/ROUTINE TO WAIT FOR KEYBOARD INPUT

```
0035 1474 XINPUT /RETURN WITH ASCII CODE IN AC AND STORED IN  
/LOCATION "INCHAR". CONSOLE PKG IS CALLED  
  
0036 4436 DECPRT=JMS I . /CALL TO ROUTINE TO CONVERT OCTAL TO DECIMAL OUTPUT  
1600 PRTDEC  
  
0037 4437 SETPT=JMS I . /CALL TO ROUTINE TO SET "TBLPT" TO DRIVE STATE TABLE ENTRY FLD 0  
1525 PTSET  
  
0040 4440 C8CAL=JMS I . /CALL TO CONSOLE PACKAGE  
0733 CALC8  
  
0041 4441 GETSR=JMS I . /CALL TO ROUTINE TO GET SOFT OR HARDWARE SWITCHES  
2155 SRGET  
  
0042 4442 GETDSC=JMS I . /CALL TO ROUTINE TO GET DEVICE CODE INDICATOR  
0716 DSCGET  
4443 VT278=JMS I .  
0043 1470 XVT278  
  
0044 0000 INCHAR, 0 /ASCII CHARACTER INPUT STORAGE  
0045 0000 INMODE, 0 /MESSAGE ACTIVE FLAG 7777=ACTIVE 0=NON-ACTIVE  
0046 0000 DVTMP, 0 /TEMPORARY DRIVE NUMBER STORAGE  
0047 0000 TBLPT, 0 /DRIVE STATE TABLE POINTER  
0050 0000 CNT1, 0  
0051 0000 CNT2, 0  
0052 0000 CDFS, 0  
0053 0000 CURDV, 0  
0054 0000 TMP1, 0  
0055 0000 DIGFLG, 0 /DIGIT INPUT FLAG  
0056 0003 K03, 0003  
0057 0007 K07, 0007  
0060 0177 K0177, 0177  
0061 0212 K0212, 0212  
0062 0215 K0215, 0215  
0063 0240 K0240, 0240  
0064 0260 K0260, 0260  
0065 0277 K0277, 0277  
0066 7774 M04, -4  
0067 7772 M06, -6  
0070 7563 M215, -215  
0071 6201 KXCDF, CDF  
0072 6203 KXCDI, CDI  
0073 0000 LINKSV, 0  
/  
/  
/*****  
/  
/  
/THE FOLLOWING LOCATIONS MUST NOT BE REARRANGED OR MODIFIED  
/UNLESS THE CORRESPONDING FIELD 0 PAGE 0 LOCATIONS ARE  
/MODIFIED CORRESPONDINGLY. THIS ENTIRE SECTION WILL BE MOVED  
/TO FIELD 0 PAGE 0 TO BE USED AS PARAMETERS FOR THE MAIN  
/PORTION OF THE PROGRAM IN FIELD 0.
```

```

0074 / TABSTR=.
/
0074 0000 CBUSY1, 0 /CONTROLLER BUSY FLAG 0=NOT BUSY 7777=BUSY
0075 7777 UNIT01, 7777 /UNIT 0 IN USE FLAG 0=USED 77=NOT USED
0076 7777 UNIT11, 7777 /UNIT 1 IN USE FLAG
0077 7777 UNIT21, 7777 /UNIT 2 IN USE FLAG
0100 7777 UNIT31, 7777 /UNIT 3 IN USE FLAG
0101 0000 SYSON0, 0 /UNIT 0 ON THIS SYSTEM FLAG 0=YES 7777=NO
0102 0000 SYSON1, 0 /UNIT 1 ON THIS SYSTEM FLAG
0103 0000 SYSON2, 0 /UNIT 2 ON THIS SYSTEM FLAG
0104 0000 SYSON3, 0 /UNIT 3 ON THIS SYSTEM FLAG
0105 0600 DSCU01, 0600 /UNIT 0 DEVICE CODE INDICATOR 0600=60,61 0620=62,63
0106 0600 DSCU11, 0600 /UNIT 1 DEVICE CODE INDICATOR
0107 0600 DSCU21, 0600 /UNIT 2 DEVICE CODE INDICATOR
0110 0600 DSCU31, 0600 /UNIT 3 DEVICE CODE INDICATOR
0111 0003 MXSTP, 3 /MAXIMUM PROGRAM STARTING STEP NUMBER
0112 0001 STSTP, 1 /STARTING STEP NUMBER
0113 0000 S2FLG, 0 /2 SYSTEM FLG, 0=1 SYSTEM 7777=2 SYSTEMS
0114 0000 NUMUUT, 0 /NUMBER OF UNITS TO TEST
0115 0001 SKRET, 0001 /SEEK TO TRACK RETRY LIMIT
/
0116 TABED=.
/
/ * END OF CRITICAL LOCATIONS *
/*****
/
0200 *200
/
0200 4424 CONSET, MESSAGE
0201 2617 TITLE /"AJRLJA RL0A/RL02 DRIVE COMPATIBILITY VERIFIER"
/
0202 1377 INIT, TAD (DRIVE0-1 /GET DRIVE TABLE ADDRESS-1
0203 3010 DCA INDX10 /INIT AUTO-INDEX TABLE POINTER
0204 6201 INITLP, CDF 00 /DATA FLD 0
0205 3410 DCA 1 INDX10 /CLEAR A DRIVE TABLE LOCATION
0206 6211 CDF 10 /DATA FLD 1
0207 1376 TAD (DRVEND /GET ADDRESS OF END OF TABLE
0210 7041 CIA /NEGATE
0211 1010 TAD INDX10 /ADD VALUE OF POINTER
0212 7640 SZA CLA /SKIP IF TABLE CLEARED
0213 5204 JMP INITLP /CONTINUE CLEARING TABLE
0214 4433 TABFIL /FILL BAD SECTOR TABLE
0215 2200 BADSEC /ADDRESS OF TABLE
0216 0010 0010 /TABLE FIELD (1)
0217 0040 BADEND-BADSEC+1 /NUMBER OF TABLE ENTRIES
0220 7777 7777 /WORD TO PLACE IN TABLE
/
0221 4433 TABFIL /FILL UNITS USED TABLE
0222 0075 UNIT01 /ADDRESS OF START OF TABLE
0223 0010 0010 /TABLE FIELD (1)
0224 0004 4 /4 TABLE ENTRIES
0225 7777 7777 /WORD TO PLACE IN ALL TABLE ENTRIES

```

0226	4433	TABFIL	/FILL UNITS/SYSTEMS TABLE
0227	0101	SYSONO	/ADDRESS OF START OF TABLE
0230	0010	0010	/TABLE FIELD (1)
0231	0004	4	/4 TABLE ENTRIES
0232	0000	0	/WORD TO PLACE IN ALL TABLE ENTRIES
/			
0233	4433	TABFIL	/FILL UNIT DEVICE CODE TABLE
0234	0105	DSCU01	/ADDRESS OF START OF TABLE
0235	0010	0010	/TABLE FIELD (1)
0236	0004	4	/4 TABLE ENTRIES
0237	0600	0600	/WORD TO PLACE IN ALL TABLE ENTRIES
/			
0240	4433	TABFIL	/FILL SECTOR DATA MAP
0241	6221	STBL16	/ADDRESS OF SECTOR DATA MAP
0242	0000	0	/FLD 0
0243	0020	20	/16 TABLE ENTRIES
0244	0000	0	/FILL WITH 0
/			
0245	3113	DCA S2FLG	/CLEAR 2 SYSTEMS FLAG
0246	7301	CLA CLL IAC	
0247	3112	DCA STSTP	/ASSUME STARTING STEP = 1 FOR PROGRAM RESTART
/			
0250	4434	UNTO, CRLF	/DO A <CRLF>
0251	4424	MESSAGE	/PRINT A MESSAGE
0252	2410	UNIT	/MESSAGE ADDRESS "UNIT "
0253	4424	MESSAGE	
0254	2413	ZMSG	/" 0 "
0255	4424	MESSAGE	
0256	2415	UNTDVR	/" = DRIVE (0-7)?"
0257	4426	OCT1	/GET ONE OCTAL DIGIT
0260	3075	DCA UNIT01	/SAVE AS POSSIBLE UNIT 0
0261	1055	TAD DIGFLG	/GET DIGIT INPUT FLAG
0262	7650	SNA CLA	/SKIP IF DIGIT TYPED
0263	5250	JMP UNTO	/UNIT 0 MUST BE ASSIGNED, REPEAT QUESTION
/			
0264	1075	TAD UNIT01	/GET UNIT 0 DRIVE NUMBER
0265	4430	LIMCHK	/CHECK AGAINST LIMITS
0266	0007	0007	/UPPER LIMIT
0267	0000	0000	/LOWER LIMIT
0270	5250	JMP UNTO	/INVALID ANSWER, REPEAT QUESTION
/			
0271	4434	CRLF	<CRLF>
0272	4424	MESSAGE	
0273	2410	UNIT	/"UNIT "
0274	4424	MESSAGE	
0275	2413	ZMSG	/" 0 "
0276	4424	MESSAGE	
0277	2426	QONSYs	/"ON THIS SYSTEM "
0300	4424	MESSAGE	
0301	2471	YORN	/"(Y/N)?"
0302	4425	YESRNO	/GET (Y)ES OR (N)O ANSWER
0303	7610	CLA SKP	/YES=0
0304	7240	CLA CMA	/NO=7777
0305	3101	DCA SYSONO	/REMEMBER ANSWER

0306	1101	TAD	SYSON0	/GET ANSWER
0307	7640	SZA	CLA	/SKIP IF UNIT 0 IS ON THIS SYSTEM
0310	5313	JMP	UNT1	/UNIT 0 IS ON OTHER SYSTEM, SKIP DEVICE CODE QUESTION
0311	4442	GETDSC		/GET UNIT 0 DEVICE CODE INDICATOR
0312	3105	DCA	DSCU01	/SAVE UNIT 0 DEVICE CODE INDICATOR
/				
0313	4434	UNT1,	CRLF	/DO A <CRLF>
0314	4424		MESSAGE	
0315	2410		UNIT	/"UNIT "
0316	4424		MESSAGE	
0317	2437		ONMSG	/" 1 "
0320	4424		MESSAGE	
0321	2415		UNTDRV	/" = DRIVE (0-7)?"
0322	4426		OCT1	/GET ONE OCTAL DIGIT INPUT
0323	3076	DCA	UNIT11	/SAVE AS POSSIBLE UNIT 1 DRIVE NUMBER
0324	1055	TAD	DIGFLG	/GET DIGIT INPUT FLAG
0325	7650	SNA	CLA	/SKIP IF DIGIT TYPED
0326	5313	JMP	UNT1	/MUST ASSIGN UNIT 1, REPEAT QUESTION
/				
0327	1076	TAD	UNIT11	/GET UNIT 1 DRIVE NUMBER
0330	4430	LIMCHK		/CHECK AGAINST UPPER AND LOWER LIMITS
0331	0007		0007	/UPPER LIMIT
0332	0000		0000	/LOWER LIMIT
0333	5313	JMP	UNT1	/INVALID ANSWER, REPEAT QUESTION
/				
0334	4434		CRLF	/<CRLF>
0335	4424		MESSAGE	
0336	2410		UNIT	/"UNIT "
0337	4424		MESSAGE	
0340	2437		ONMSG	/" 1 "
0341	4424		MESSAGE	
0342	2426		QONSYS	/"ON THIS SYSTEM "
0343	4424		MESSAGE	
0344	2471		YORN	/"(Y/N)?"
0345	4425		YESRNO	/GET YES OR NO ANSWER
0346	7610	CLA	SKP	/YES=0
0347	7240	CLA	CHA	/NO=7777
0350	3102	DCA	SYSON1	/REMEMBER ANSWER
0351	1102	TAD	SYSON1	/GET ANSWER
0352	7640	SZA	CLA	/SKIP IF UNIT 1 ON THIS SYSTEM
0353	5356	JMP	UNT2	/UNIT 1 ON OTHER SYSTEM, SKIP DEVICE CODE QUESTION
0354	4442	GETDSC		/GET UNIT 1 DEVICE CODE
0355	3106	DCA	DSCU11	/SAVE UNIT 1 DEVICE CODE INDICATOR
/				
0356	4434	UNT2,	CRLF	/DO A <CRLF>
0357	4424		MESSAGE	
0360	2410		UNIT	/"UNIT "
0361	4424		MESSAGE	
0362	2441		TWOMSG	/" 2 "
0363	4424		MESSAGE	
0364	2415		UNTDRV	/" = DRIVE (0-7)?"
0365	4426		OCT1	/GET ONE OCTAL DIGIT INPUT
0366	5775	JMP	NXTPGO	/GO TO NEXT PAGE
/				
0375	0400			

0376 7527
0377 7403

SEQ 138

```

0400 PAGE
/
/
0400 3077 NXXPGO, DCA UNIT21 /SAVE AS POSSIBLE UNIT 2 DRIVE NUMBER
0401 1055 TAD DIGFLG /GET DIGIT INPUT FLAG
0402 7640 SZA CLA /SKIP IF NO DIGIT
0403 5211 JMP U2CHK /DIGIT INPUT
0404 1377 TAD (2 /NO DIGIT, ONLY 2 UNITS TO TEST
0405 3114 DCA NUMUUT /SAVE AS 2 UNITS TO TEST
0406 7040 CMA
0407 3077 DCA UNIT21 /SET UNIT NOT ACTIVE FLG
0410 5312 JMP SYSCHK /GO CHECK FOR 2 SYSTEMS
0411 1077 U2CHK, TAD UNIT21 /GET UNIT 2 DRIVE NUMBER
0412 4430 LIMCHK /CHECK UNIT 2 DRIVE NUMBER AGAINST LIMITS
0413 0007 0007 /UPPER LIMIT
0414 0000 0000 /LOWER LIMIT
0415 5776' JMP UNT2 /INVALID DRIVE NUMBER, REPEAT QUESTION
/
0416 4434 CRLF /<CRLF>
0417 4424 MESSAGE
0420 2410 UNIT /"UNIT "
0421 4424 MESSAGE
0422 2441 TWMSG /" 2 "
0423 4424 MESSAGE
0424 2426 QON SYS /"ON THIS SYSTEM "
0425 4424 MESSAGE
0426 2471 YORN /"(Y/N)?"
0427 4425 YESRNO /GET (Y)ES OR (N)O ANSWER
0430 7610 CLA SKP /YES=0
0431 7240 CLA CMA /NO=7777
0432 3103 DCA SYSON2 /REMEMBER ANSWER
0433 1103 TAD SYSON2 /GET ANSWER
0434 7640 SZA CLA /SKIP IF UNIT 2 IS ON THIS SYSTEM
0435 5240 JMP UNT3 /UNIT 2 IS ON OTHER SYSTEM, SKIP DEVICE CODE QUESTION
0436 4442 GETDSC /GET UNIT 2 DEVICE CODE INDICATOR
0437 3107 DCA DSCU21 /REMEMBER UNIT 2 DEVICE CODE INDICATOR
/
0440 4434 UNT3, CRLF /DO A <CRLF>
0441 4424 MESSAGE
0442 2410 UNIT /"UNIT "
0443 4424 MESSAGE
0444 2443 THRM SG /" 3 "
0445 4424 MESSAGE
0446 2415 UNTDRV /" = DRIVE (0-7)?"
0447 4426 OCT1 /GET ONE OCTAL DIGIT INPUT
0450 3100 DCA UNIT31 /SAVE POSSIBLE UNIT 3 DRIVE NUMBER
0451 1055 TAD DIGFLG /GET DIGIT INPUT FLAG
0452 7640 SZA CLA /SKIP IF NO DIGITS INPUT
0453 5261 JMP U3CHK /DIGIT INPUT
0454 1056 TAD K03 /3 UNITS TO TEST
0455 3114 DCA NUMUUT /SAVE AS 3 UNITS TO TEST
0456 7040 CMA
0457 3100 DCA UNIT31 /SET UNIT 3 NOT ACTIVE
0460 5312 JMP SYSCHK /GO CHECK FOR 2 SYSTEMS

```

0461	1100	U3CHK,	TAD	UNIT31	/GET UNIT 3 DRIVE NUMBER
0462	4430		LIMCHK		/CHECK AGAINST LIMITS
0463	0007		0007		/UPPER LIMIT
0464	0000		0000		/LOWER LIMIT
0465	5240		JMP	UNT3	/INVALID DRIVE NUMBER
0466	7307		CLA CLL	IAC RTL	/AC=0004
0467	3114		DCA	NUMUUT	/4 UNITS TO TEST
/					
0470	4434		CRLF		/<CRLF>
0471	4424		MESSAGE		
0472	2410		UNIT		/"UNIT "
0473	4424		MESSAGE		
0474	2443		THRMSG		/" 3 "
0475	4424		MESSAGE		
0476	2426		QONSY8		/" ON THIS SYSTEM"
0477	4424		MESSAGE		
0500	2471		YORN		/"(Y/N)?"
0501	4425		YESRNO		/GET (Y)ES OR (N)O ANSWER
0502	7610		CLA	SKP	/YES=0
0503	7240		CLA	CMA	/NO=7777
0504	3104		DCA	SYSON3	/REMEMBER ANSWER
0505	1104		TAD	SYSON3	/GET ANSWER
0506	7640		SZA	CLA	/SKIP IF UNIT 3 IS ON THIS SYSTEM
0507	5312		JMP	SYSCHK	/UNIT 3 IS ON OTHER SYSTEM, SKIP DEVICE CODE QUESTION
0510	4442		GETDSC		/GET UNIT 3 DEVICE CODE INDICATOR
0511	3110		DCA	DSCU31	/SAVE UNIT 3 DEVICE CODE INDICATOR
/					
0512	1101	SYSCHK,	TAD	SYSON0	/GET UNIT 0 ON THIS SYSTEM FLG
0513	1102		TAD	SYSON1	/ADD UNIT 1 ON THIS SYSTEM FLG
0514	1103		TAD	SYSON2	/ADD UNIT 2 ON THIS SYSTEM FLG
0515	1104		TAD	SYSON3	/ADD UNIT 3 ON THIS SYSTEM FLG
0516	7450		SNA		/SKIP IF A UNIT IS NOT ON THIS SYSTEM
0517	5775'		JMP	SRQUES	/ALL UNITS THIS SYSTEM
0520	1374		TAD	(4	/ADD 4 TO SEE IF ANY UNITS THIS SYSTEM
0521	7640		SZA	CLA	/SKIP IF NO UNITS THIS SYSTEM
0522	5326		JMP	STPASK	/GO ASK STARTING STEP NUMBER
0523	4424		MESSAGE		
0524	3304		NOUNIT		/"NO UNITS THIS SYSTEM"
0525	5773'		JMP	INIT	/START OVER
/					
0526	7240	STPASK,	CLA	CMA	
0527	3113		DCA	S2FLG	/SET 2 SYSTEMS FLAG = 7777, 2 SYSTEMS UNDER TEST
0530	1077		TAD	UNIT21	/GET UNIT 2 IN USE FLAG
0531	7700		SMA	CLA	/SKIP IF ONLY 2 UNITS (ONLY UNIT 0 AND UNIT 1)
0532	5336		JMP	U31CHK	/UNIT 2 IN USE, GO CHECK FOR UNIT 3 IN USE
0533	1056		TAD	K03	
0534	3111		DCA	MXSTP	/MAXIMUM STARTING STEP NUMBER = 3
0535	5346		JMP	ASKSTP	/GO ASK STARTING STEP NUMBER
/					
0536	1100	U31CHK,	TAD	UNIT31	/GET UNIT 3 IN USE FLAG
0537	7700		SMA	CLA	/SKIP IF ONLY 3 UNITS (ONLY UNITS 0,1, AND 2)
0540	5344		JMP	ALLUNT	/GO USE MAXIMUM STARTING STEP OF 7 (UNITS 0,1,2, AND 3 IN USE)
0541	1372		TAD	(5	
0542	3111		DCA	MXSTP	/MAXIMUM STARTING STEP = 5


```

0543 5346      JMP      ASKSTP      /GO ASK STARTING STEP NUMBER
/
0544 1057      ALLUNT, TAD      K07
0545 3111      DCA      MXSTP      /MAXIMUM STARTING STEP = 7
/
0546 4434      ASKSTP, CRLF      /DO A <CRLF>
0547 4424      MESSAGE
0550 2445      FRSTP
0551 1111      TAD      MXSTP      /*START WITH STEP (1-
0552 1064      TAD      K0260      /GET MAXIMUM STARTING STEP
0553 4432      PRINT
0554 4424      MESSAGE      /PRINT MAXIMUM STARTING STEP
0555 2463      PARENQ
0556 4426      OCT1      /"? "
0557 3112      DCA      STSTP      /GET ONE OCTAL DIGIT
0560 1055      TAD      DIGFLG      /SAVE AS POSSIBLE STARTING STEP
0561 7650      SNA      CLA      /GET DIGIT INPUT FLAG
0562 5346      JMP      ASKSTP      /SKIP IF DIGIT TYPED
0563 1111      TAD      MXSTP      /GO REPEAT QUESTION, MUST HAVE STARTING STEP NUMBER
0564 3771      DCA      UPLSTP      /GET MAXIMUM STARTING STEP NUMBER
0565 5770      JMP      NXTPG1      /SAVE AS UPPER LIMIT TO CHECK
/
/GO TO NEXT PAGE

0570 0600
0571 0602
0572 0005
0573 0202
0574 0004
0575 0605
0576 0356
0577 0002
0600 0600      PAGE
/
/
0600 1112      NXTPG1, TAD      STSTP      /GET POSSIBLE STARTING STEP
0601 4430      LIMCHK      /CHECK AGAINST LIMITS
0602 7402      UPLSTP, HLT/ MAX STEP      /MODIFIED MAXIMUM STARTING STEP
0603 0001      0001      /MINIMUM STARTING STEP NUMBER
0604 5777      JMP      ASKSTP      /INVALID, REPEAT QUESTION
/
/
0605 7300      SRQUES, CLA      CLL
0606 4440      C8CAL
0607 7402      HLT      /CALL CONSOLE PACKAGE - ASK SR=
/NOT ACTIVE RETURN - PROGRAM ERROR?
/
/
0610 4424      DBLCHK, MESSAGE
0611 2513      RUSURE
0612 4424      MESSAGE
0613 2471      YORN
0614 4425      YESRNO
0615 7610      SKP CLA
0616 5776      JMP      INIT
0617 1375      MAINGO, TAD      (TABSTR-1
0620 3010      DCA      IND10
0621 1374      TAD      (TBLSTR-1

```

```

0622 3011      DCA      INDX11      /SET UP PAGE 0 FLD 0 POINTER
0623 1373      TAD      (TABED-TABSTR /GET SIZE OF TABLE TO MOVE
0624 7041      CIA      /NEGATE FOR USE AS COUNTER
0625 3050      DCA      CNT1        /SET UP TBL CNTR
0626 6211      MOVLP,   CDF      10   /DATA FLD 1
0627 1410      TAD I    INDX10      /GET FLD 1 PAGE 0 CONTENTS
0630 6201      CDF      00         /DATA FLD 0
0631 3411      DCA I    INDX11      /MOVE TO FLD 0 PAGE 0
0632 2050      ISZ      CNT1        /INCREMENT MOVE COUNTER
0633 5226      JMP      MOVLP       /CONTINUE COPYING FLD 1 PG 0 TO FLD 0 PG 0
0634 6203      CDI      00         /GOING TO FIELD 0
0635 5636      JMP I    .+1
0636 0213      MAIN

```

```

/GO TO FLD 0

```

```

/
/
/*****
/

```

```

/ROUTINE TO PRINT CONTENTS OF AC
/

```

```

/      CALLED BY:      PRNTAC
/

```

```

0637 0000      ACPRNT, 0
0640 3266      DCA      ACTMP       /SAVE AC
0641 6214      RDF      /READ CALLING DATA FIELD
0642 1072      TAD      KXCDI      /ADD CDI BASE INSTRUCTION
0643 3264      DCA      ACOUT       /SAVE CDI TO CALLING FIELD
0644 6211      CDF      10         /DATA FLD 1
0645 1066      TAD      M04
0646 3267      DCA      ACCNTR      /INIT DIGIT PRINT COUNTER
0647 1266      TAD      ACTMP       /GET AC
0650 7104      CLL RAL          /EXTRA ROTATE FOR LINK
0651 7004      ACLP,   RAL
0652 7006      RTL
0653 3266      DCA      ACTMP       /ROTATE LEFT AC OCTAL DIGIT FOR PRINTING
0654 1266      TAD      ACTMP       /SAVE ROTATED AC
0655 0057      AND      K07         /GET ROTATED AC
0656 1064      TAD      K0260      /MASK OCTAL DIGIT
0657 4432      PRINT          /ADD ASCII BASE CODE
0660 1266      TAD      ACTMP       /PRINT OCTAL DIGIT
0661 2267      ISZ      ACCNTR      /GET ROTATED AC
0662 5251      JMP      ACLP        /INCREMENT DIGIT COUNTER
0663 7300      CLA CLL          /CONTINUE
0664 7402      ACOUT,   HLT/CDI     /CDI TO CALLING FIELD MODIFIED
0665 5637      JMP I    ACPRNT      /EXIT

0666 0000      ACTMP, 0           /TEMPORARY STORAGE OF ROTATED AC
0667 0000      ACCNTR, 0         /DIGIT COUNTER
/
/
/
/

```

```

/ROUTINE TO DELAY IN INCREMENTS OF SECONDS (APPROXIMATELY)
/

```

```

/      CALLED BY:      JMS I      (XDELAY      /FROM FLD 0
/      FOLLOWED BY:     -XXXX      /WHERE XXXX IS THE 2'S COMPLEMENT
/                                     /OF THE NUMBER OF SECONDS TO DELAY
/
/RETURN CALL+2 TO FLD 0
/
/DELAY IS IN INCREMENTS OF APPROXIMATELY .996 SECOND ON A PDP-8E
/DELAY IS IN INCREMENTS OF APPROXIMATELY 1.18 SECOND ON A PDP-8A
/MAXIMUM DELAY IS 4096 TIMES APPROX 1 SECOND
/
/
0670 0000 XDELAY, 0
0671 7300      CLA      CLL
0672 1670      TAD I    XDELAY      /GET NUMBER OF SECONDS TO DELAY FROM CALL+1
0673 3313      DCA      DLY1      /SET UP DELAY COUNTER
0674 1372      DLPA,    TAD      (-47 /39 TIMES 25.39MS (8E) OR 30.72MS (8A)
0675 3314      DCA      DLY2      /SET UP DELAY COUNTER
0676 3315      DCA      DLY3      /CLEAR DELAY COUNTER
0677 6102      DLPB,    SPL      /SKIP ON POWER LOW
0700 7410      SKP      /SKIP ON POWER OK
0701 6001      IQN      /INTERRUPT ON FOR POWER FAIL
0702 2315      ISZ      DLY3      /INCREMENT 4096 COUNTER
0703 5277      JMP      DLPB      /KEEP INCREMENTING 4096 COUNTER
0704 2314      ISZ      DLY2      /INCREMENT TIMES 4096 COUNTER
0705 5277      JMP      DLPB      /KEEP INCREMENTING UNTIL 1 SECOND DELAY
0706 2313      ISZ      DLY1      /INCREMENT SECONDS COUNTER
0707 5274      JMP      DLPA      /MORE SECONDS TO GO
0710 2270      ISZ      XDELAY     /INCREMENT RETURN PC
0711 6203      CDI      00        /RETURNING TO FLD 0
0712 5670      JMP I    XDELAY     /RETURN
/
/
0713 0000      DLY1,    0          /SECONDS COUNTER
0714 0000      DLY2,    0          /ONE SECOND COUNTER
0715 0000      DLY3,    0          /1/64 SECOND COUNTER
/
/
/*****
/
/ROUTINE TO GET DEVICE CODE INDICATOR
/
/      CALLED BY:      GETDSC
/
/RETURN WITH DEVICE CODE INDICATOR IN AC
/
/
0716 0000      DSCGET, 0
0717 4424      MESSAGE
0720 2645      DCODE      /"DEVICE CODE"
0721 4424      MESSAGE
0722 2502      DC6061     /"(60,61)"
0723 4424      MESSAGE
0724 2471      YORN      /"(Y/N)?"

```

0725	4425	YESRNO	/GET (Y)ES OR (N)O ANSWER
0726	5331	JMP .+3	/YES
0727	1371	TAD (0620	/NO USE DEVICE CODE 62,63
0730	7410	SKP	
0731	1370	TAD (0600	/YES, USE DEVICE CODE 60,61
0732	5716	JMP I DSCGET	/RETURN WITH DEVICE CODE INDICATOR

/ROUTINE TO CHECK FOR CONSOLE PACKAGE ACTIVE

/IF CONSOLE PACKAGE ACTIVE, GO TO CONSOLE PACKAGE

/RETURN CALL+2 AC CLEAR

/IF CONSOLE PACKAGE NOT ACTIVE, RETURN CALL+1 AC CLEAR

CALLLED BY: C8CAL

0733	0000	CALC8, 0	
0734	3356	DCA CHRTEM	/SAVE AC
0735	6000	SKON	/SKIP IF INTERRUPT ON AND TURN OFF
0736	7301	CLA CLL IAC	
0737	1367	TAD (ION	
0740	3354	DCA C8OUT	/SAVE ION OR IOF FOR EXIT
0741	1022	TAD 22	/GET HDW2
0742	0366	AND (400	/TEST FOR BIT 3=1 CONSOLE ACTIVE
0743	7650	SNA CLA	/SKIP IF CONSOLE ACTIVE
0744	5354	JMP C8OUT	/RETURN CALL+1 CONSOLE NOT ACTIVE
0745	1356	TAD CHRTEM	/GET CHARACTER TO CHECK
0746	4765	JMS CBENTR	/GO TO CONSOLE PACKAGE
0747	2333	ISZ CALC8	/INCREMENT RETURN ADDRESS
0750	1020	TAD PSR1	/GET PSR1
0751	6201	CDF 00	/CDF TO FLD 0
0752	3764	DCA I (PSR	/COPY SOFTWARE SWITCH REGISTER TO FLD 0 PG 0 LOC 20
0753	6211	CDF 10	/CDF TO PRGM FLD
0754	7402	C8OUT, HLT/ION/IOF	/MODIFIED ION OR IOF FOR EXIT
0755	5733	JMP I CALC8	/RETURN CALL+2 CONSOLE WAS ACTIVE

0756	0000	CHRTEM, 0	/AC STORAGE
------	------	-----------	-------------

0764	0020
0765	3422
0766	0400
0767	6001
0770	0600
0771	0620
0772	7731
0773	0022

0774 0054
 0775 0073
 0776 0202
 0777 0546
 1000

PAGE

/ROUTINE TO ACCEPT 4 DECIMAL DIGITS INPUT FROM KEYBOARD
 /CONVERT TO OCTAL WORD
 /MAXIMUM = 4096 (DECIMAL) = 0000 (OCTAL)
 /RETURN WITH OCTAL EQUIVALENT IN AC

/ CALLED BY: DEC4

1000	0000	XDEC4, 0	
1001	7300	CLA CLL	
1002	3263	DCA LNK1	/CLEAR LINK SAVE AREA
1003	1377	TAD (-5	
1004	3261	DCA DIGCNT	/INIT DIGIT COUNTER
1005	3260	DCA DECBLD	/CLEAR CONVERTED DIGIT AREA
1006	3055	DCA DIGFLG	/CLEAR DIGIT INPUT FLAG
1007	4435	DEC4LP, INPUT	/GO WAIT FOR CHARACTER INPUT
1010	1070	TAD M215	
1011	7450	SNA	/SKIP IF NOT <CR>
1012	5253	JMP DEC4EX	<CR> GO TO EXIT
1013	1376	TAD (-43	
1014	7510	SPA	/SKIP IF > OR = 260
1015	5245	JMP DEC4ER	/NOT DIGIT - START OVER
1016	1375	TAD (-12	
1017	7700	SMA CLA	/SKIP IF < 272
1020	5245	JMP DEC4ER	/NOT DIGIT OR <CR> - START OVER
1021	7040	CMA	
1022	3055	DCA DIGFLG	/SET DIGIT INPUT FLAG
1023	1044	TAD INCHAR	/GET INPUT CHARACTER AGAIN
1024	0374	AND (17	/MASK TO BINARY NUMBER
1025	3262	DCA DECTMP	/STORE BINARY CHARACTER
1026	1260	TAD DECBLD	/GET CONVERTED DIGITS
1027	7106	CLL RTL	/MULTIPLY BY 4
1030	1260	TAD DECBLD	/ADD DIGITS (NOW TIMES 5)
1031	7104	CLL RAL	/MULTIPLY BY 2
1032	1262	TAD DECTMP	/ADD NEXT DIGIT
1033	3260	DCA DECBLD	/STORE CONVERTED DIGITS
1034	7420	SNL	/SKIP IF > 4095
1035	5243	JMP .+6	/NOT > 4095
1036	7301	CLA CLL IAC	/SET BIT 11 FOR EXIT
1037	3263	DCA LNK1	/SAVE FOR EXIT
1040	1260	TAD DECBLD	/GET CONVERTED NUMBER
1041	7640	SZA CLA	/SKIP IF = 4096
1042	5245	JMP DEC4ER	/ > 4096 NOT VALID
1043	2261	ISZ DIGCNT	/INCREMENT DIGIT COUNT - SKIP IF 5 DIGITS

```

1044 5207          JMP      DEC4LP          /CONTINUE
/
1045 7300  DEC4ER, CLA CLL
1046 3055          DCA      DIGFLG          /CLEAR DIGIT INPUT FLAG
1047 3263          DCA      LNK1           /CLEAR 4096 LINK INDICATOR WORD
1050 1065          TAD      K0277
1051 4432          PRINT
1052 5201          JMP      XDEC4+1         /PRINT "?"
/                                     /START OVER
1053 7300  DEC4EX, CLA CLL
1054 1263          TAD      LNK1           /GET LINK SAVED
1055 7010          RAR
1056 1260          TAD      DECBLD         /PUT IT INTO LINK
1057 5600          JMP I    XDEC4         /GET CONVERTED NUMBER
/                                     /EXIT
1060 0000  DECBLD, 0
1061 0000  DIGCNT, 0
1062 0000  DECTMP, 0
1063 0000  LNK1, 0
/                                     /LINK SAVE AREA
/
/
/*****
/
/ROUTINE TO ACCEPT 1 OCTAL DIGIT INPUT
/RETURN WITH OCTAL DIGIT RIGHT JUSTIFIED IN AC
/TERMINATED BY <CR>
/
/      CALLED BY:      OCT1
/
/
1064 0000  XOCT1, 0
1065 7300          CLA CLL
1066 6214          RDF
1067 1072          TAD      KXCDI          /READ DATA FIELD
1070 3311          DCA      OCTOUT         /ADD CDI INSTRUCTION
1071 6211          CDF      10           /SAVE CDI TO CALLING FLD
1072 3055  TRYOCT, DCA      DIGFLG         /DATA FLD 1
1073 4435          INPUT
1074 1070          TAD      M215          /CLEAR DIGIT INPUT FLAG
1075 7450          SNA
1076 5311          JMP      OCTOUT         /GO WAIT FOR KEYBOARD INPUT
1077 1376          TAD      (-43
1100 7510          SPA
1101 5313          JMP      OCT1ER         /SKIP IF NOT <CR>
1102 1373          TAD      (-10          /RETURN AC = 0
1103 7700          SNA CLA
1104 5313          JMP      OCT1ER         /SKIP IF < 270
1105 7040          CMA
1106 3055          DCA      DIGFLG         /NOT OCTAL DIGIT - START OVER
1107 1044          TAD      INCHAR
1110 0057          AND      K07           /SET DIGIT INPUT FLAG
1111 7402  OCTOUT, HLT/CDI
1112 5664          JMP I    XOCT1         /GET CHARACTER AGAIN
/                                     /MASK TO OCTAL DIGIT
/                                     /CDI TO CALLING FLD MODIFIED
/                                     /EXIT

```

```

1113 7300 / OCT1ER, CLA CLL
1114 3055 DCA DIGFLG /CLEAR DIGIT INPUT FLAG
1115 1065 TAD K0277
1116 4432 PRINT /PRINT "?"
1117 5272 JMP TRYOC /START OVER

1120 0000 / OCTONE, 0 /TEMPORARY STORAGE OF OCTAL DIGIT
/
/
/*****
/
/ A DATA WORD CONTAINED IN THE ARGUMENT LIST
/ CALL+1 IS STARTING ADDRESS OF TABLE
/ CALL+2 IS NUMBER OF TABLE ENTRIES (UNSIGNED) MAX = 7777 (4095)
/ CALL+3 IS THE VALUE TO PLACE IN EACH OF THE TABLE LOCATIONS
/ CALL+4 IS RETURN WITH AC CLEAR
/
/ EXAMPLE: TABFIL (SUBROUTINE CALL)
/ 7 (NUMBER OF TABLE ENTRIES)
/ 0 (VALUE TO PLACE IN TABLE)
/ (RETURN)
/

1121 0000 XTFIL, 0
1122 7340 CLA CLL CMA
1123 1721 TAD I XTFIL /GET STARTING ADDRESS OF TABLE-1
1124 3011 DCA INDX11 /SET UP TABLE POINTER
1125 2321 ISZ XTFIL /INCREMENT PARAMETER POINTER
1126 1721 TAD I XTFIL /GET TABLE FIELD
1127 1071 TAD KXCDF /ADD CDF INSTRUCTION
1130 3340 DCA XCDF1 /SAVE CDF TO TABLE FIELD FOR EXECUTION
1131 2321 ISZ XTFIL /INCREMENT PARAMETER POINTER
1132 1721 TAD I XTFIL /GET NUMBER OF TABLE ENTRIES
1133 7041 CIA
1134 3051 DCA CNT2 /SET UP COUNTER
1135 2321 ISZ XTFIL /INCREMENT PARAMETER POINTER
1136 1721 TAD I XTFIL /GET WORD TO FILL TABLE WITH
1137 3350 DCA TBLWRD /SAVE TABLE WORD VALUE
1140 7402 XCDF1, HLT/CDF /MODIFIED CDF TO TABLE FIELD
1141 1350 XTLP, TAD TBLWRD /GET WORD VALUE
1142 3411 DCA I INDX11 /FILL A TABLE WORD
1143 2051 ISZ CNT2 /INCREMENT LOCATION COUNTER
1144 5341 JMP XTLP /CONTINUE FILLING TABLE
1145 6211 CDF 10 /CHANGE BACK TO PROGRAM DATA FIELD
1146 2321 ISZ XTFIL /BUMP RETURN POINTER
1147 5721 JMP I XTFIL /EXIT

1150 0000 / TBLWRD, 0
/
/*****

```

```

/
/
/ROUTINE TO CHECK FOR CONTROL S ACTIVE FLAG SET
/AND IF SET, WAIT FOR IT TO BE CLEARED BY A CONTROL Q
/

```

```

/      CALLED BY:      JMS      CSCHK
/

```

```

1151 0000 CSCHK, 0
1152 7300 CLA      CLL
1153 1772 TAD      C8SFLG /GET CONTROL S ACTIVE FLAG
1154 7700 SNA      CLA /SKIP IF CONTROL S ACTIVE
1155 5751 JMP I     CSCHK /RETURN, CTRL S NOT ACTIVE
1156 4435 INPUT    /WAIT FOR KEYBOARD INPUT
1157 5352 JMP      CSCHK+1 /CHECK THE CONTROL S FLAG AGAIN AFTER INPUT

```

```

/
/
/*****
/

```

```

1172 3604
1173 7770
1174 0017
1175 7766
1176 7735
1177 7773
1200

```

```

PAGE
/
/

```

```

/*****
/

```

```

/ROUTINE TO PRINT A MESSAGE CONSISTING OF 6-BIT PACKED ASCII
/TWO CHARACTERS PER 12 BIT WORD
/MESSAGE TERMINATED BY A 6-BIT BYTE OF ALL ZEROS
/A "#" APPEARING IN THE TEXT WILL CAUSE A CARRIAGE RETURN
/AND LINE FEED TO BE EXECUTED INSTEAD OF A "#".
/

```

```

1200 0000 MESPRT, 0
1201 6214 RDF      /READ CALLING DATA FIELD
1202 1072 TAD      /ADD CDI TO CALLING FIELD BASE CODE
1203 3252 DCA      MESOUT /SAVE CDI TO CALLING FLD INSTRUCTION
1204 1600 TAD I     MESPRT /GET MESSAGE ADDRESS
1205 6211 CDF      10 /CDF TO PRGM FLD
1206 3255 DCA      MESPNT /SAVE MESSAGE ADDRESS AS POINTER
1207 6000 SKON     /SKIP IF INT ON AND TURN OFF
1210 7301 CLA CLL IAC /INTERRUPT NOT ON SET BIT 11
1211 1377 TAD      (ION /GET ION INSTRUCTION
1212 3251 DCA      MESINT /SAVE FOR EXECUTION UPON RETURN
1213 7240 STA      /AC=7777
1214 3045 DCA      INMODE /SET MESSAGE ACTIVE FLAG
1215 7344 CLA CLL CMA RAL /AC=-2
1216 3254 DCA      MESCNT /SET UP 6-BIT CHARACTER COUNTER

```



```

/
/ROUTINE TO ACCEPT A YES OR NO (Y OR N) ANSWER TO A QUESTION
/RETURN CALL+1 IF "YES" (Y)
/RETURN CALL+2 IF "NO" (N)
/

```

```

/      EXAMPLE:      YESRNO      (SUBROUTINE CALL)
/                    CLA SKP      (YES RETURN)
/                    CLA CMA      (NO RETURN)
/                    DCA      ANSFLG (STORE ANSWER AS 0=YES 7777=NO)
/

```

```

1273 0000  XYESNO, 0
1274 6214      RDF      /READ CALLING DATA FLD
1275 1072      TAD      /ADD CDI TO CALLING FLD BASE CODE
1276 3314      DCA      YNOUT /SAVE CDI TO CALLING FLD FOR EXECUTION
1277 6211      CDF      10 /CDF TO PRGM FLD
1300 4435  TRYYN, INPUT /GO WAIT FOR INPUT OF Y OR N
1301 1372      TAD      (-316
1302 7450      SNA      /SKIP IF NOT (N)O
1303 5312      JMP      NEXIT /IT WAS (N)O GO TO NO EXIT
1304 1371      TAD      (-13
1305 7650      SNA CLA /SKIP IF NOT (N)O OR (Y)ES
1306 5313      JMP      YEXIT /IT WAS (Y)ES GO TO YES EXIT
1307 1370  NOTYN, TAD      (277
1310 4432      PRINT     /PRINT "?"
1311 5300      JMP      TRYYN /TRY AGAIN
/
1312 2273  NEXIT, ISZ      XYESNO /INCREMENT RETURN FOR "NO" EXIT
1313 4434  YEXIT, CRLF     /DO A <CR> AND <LF>
1314 7402  YNOUT, HLT/CDI /MODIFIED CDI TO CALLING FLD
1315 5673      JMP I XYESNO /<CR> TYPED - EXIT

```

```

/*****
/
/

```

```

/ROUTINE TO PRINT ONE CHARACTER
/

```

```

/IF KEYBOARD FLAG IS SET AT THE END OF THIS ROUTINE
/THE CONSOLE PACKAGE WILL BE ENTERED IF ACTIVE
/

```

```

1316 0000  XPRINT, 0
1317 3356      DCA      PRNTMP /SAVE CHAR TO PRINT
1320 7004      RAL      /GET LINK IN AC 11
1321 3073      DCA      LINKSV /SAVE FOR RETURN
1322 6000      SKON     /SKIP IF INT ON AND TURN OFF
1323 7301      CLA CLL IAC
1324 1377      TAD      (ION /GET ION INSTRUCTION
1325 3354      DCA      PRNINT /SAVE ION OR AND 0 FOR EXIT
1326 6214      RDF      /READ CALLING DATA FLD
1327 1072      TAD      KXCDI /ADD CDI BASE CODE
1330 3353      DCA      PRNOUT /SAVE CDI TO CALLING FLD INSTRUCTION
1331 6211      CDF      10 /DATA FLD 1
1332 4767      JMS      CSCHK /CHECK FOR CONTROL S ACTIVE
1333 1356      TAD      PRNTMP /GET CHARACTER TO PRINT
1334 6046      TIS      /TRANSMIT A CHARACTER
1335 6102      SPL      /SKIP ON POWER LOW

```

```

1336 7410      SKP      /SKIP IF POWER OK
1337 6001      ION      /TURN ON INTERRUPT SYSTEM - POWER LOW
1340 6041      TSF      /TEST TELEPRINTER FLAG
1341 5335      JMP      .-4 /WAIT FOR TELEPRINTER FLAG
1342 6042      TCF      /CLEAR TELEPRINTER FLAG
1343 6031      KSF      /TEST KEYBOARD FLAG
1344 5350      JMP      .+4 /KEYBOARD FLAG NOT SET
1345 4443      VT278
1346 6036      KRB      /KEYBOARD FLAG SET  READ KEYBOARD BUFFER
1347 4440      C8CAL     /GO TO CONSOLE PACKAGE IF ACTIVE
1350 7200      CLA
1351 1073      TAD      LINKSV /GET LINK
1352 7110      CLL RAR   /RESTORE LINK
1353 7402      PRNOUT, HLT/CDI /CDI TO CALLING FLD MODIFIED
1354 7402      PRNINT, HLT/ION/IOF /RESTORE INTERRUPT STATE
1355 5716      JMP I    XPRINT /RETURN
1356 0000      PRNTMP, 0

```

```

1367 1151
1370 0277
1371 7765
1372 7462
1373 0100
1374 0003
1375 7735
1376 0077
1377 6001
1400

```

PAGE

```

/*****

```

```

/ROUTINE TO CHECK A VALUE CONTAINED IN AC TO BE WITHIN
/UPPER AND LOWER LIMITS AS SPECIFIED IN ARGUMENT LIST
/FOLLOWING SUBROUTINE CALL
/CALL+1 = UPPER LIMIT, CALL+2 = LOWER LIMIT (MUST BE POSITIVE)
/RETURN CALL+3 IF LIMITS EXCEEDED
/RETURN CALL+4 IF LIMITS NOT EXCEEDED
/AC CONTAINS VALUE TO CHECK WHEN "LIMCHK" IS CALLED
/RETURN WITH AC CLEAR

```

```

/      EXAMPLE:      LIMCHK      (SUBROUTINE CALL)
/                      176        (UPPER LIMIT)
/                      0          (LOWER LIMIT)
/                      JMP REASK   (LIMITS EXCEEDED RETURN)
/                      (LIMITS NOT EXCEEDED RETURN)
/

```

```

1400 0000      XLCHK, 0
1401 3267      DCA      CHKTMP    /SAVE VALUE TO CHECK
1402 1600      TAD I    XLCHK     /GET UPPER LIMIT

```

1403	3266	DCA	UPCHK	/SAVE UPPER LIMIT
1404	2200	ISZ	XLCHK	/POINT TO LOWER LIMIT
1405	1267	TAD	CHKTMP	/GET VALUE TO CHECK
1406	7710	SPA CLA		/SKIP IF POSITIVE
1407	5226	JMP	VALNEG	/GO CHECK NEGATIVE VALUE
1410	1266	TAD	UPCHK	/GET UPPER LIMIT
1411	7710	SPA CLA		/SKIP IF POSITIVE - ELSE GREATER THAN VALUE TO CHECK
1412	5220	JMP	LOCHK	/GO CHECK LOWER LIMIT
1413	1267	TAD	CHKTMP	/GET VALUE TO CHECK
1414	7041	CIA		/NEGATE
1415	1266	TAD	UPCHK	/ADD UPPER LIMIT
1416	7710	SPA CLA		/SKIP IF VALUE < OR = UPPER LIMIT
1417	5245	JMP	BADEX1	/VALUE > UPPER LIMIT - GO TO BAD EXIT
/				
1420	1600	LOCHK,	TAD I XLCHK	/GET LOWER LIMIT
1421	7041	CIA		/NEGATE
1422	1267	TAD	CHKTMP	/ADD VALUE TO CHECK
1423	7710	SPA CLA		/SKIP IF VALUE > OR = LOWER LIMIT
1424	5245	JMP	BADEX1	/VALUE < LOWER LIMIT - GO TO BAD EXIT
1425	5242	JMP	GOODEX	/VALUE OK - GO TO GOOD EXIT
/				
1426	1266	VALNEG,	TAD UPCHK	/GET UPPER LIMIT
1427	7700	SMA CLA		/SKIP IF NEGATIVE
1430	5245	JMP	BADEX1	/VALUE TO CHECK IS > UPPER LIMIT
1431	7240	CLA CMA		
1432	1267	TAD	CHKTMP	/VALUE TO CHECK - 1 IN AC
1433	7700	SMA CLA		/SKIP IF NOT 4000(8)
1434	5242	JMP	GOODEX	/IT WAS 4000(8) MUST BE < OR = UPPER LIMIT
1435	1267	TAD	CHKTMP	/GET VALUE TO CHECK
1436	7041	CIA		/MAKE IT POSITIVE
1437	1266	TAD	UPCHK	/ADD NEGATIVE UPPER LIMIT
1440	7710	SPA CLA		/SKIP IF VALUE < OR = UPPER LIMIT
1441	5245	JMP	BADEX1	/VALUE > UPPER LIMIT - GO TO BAD EXIT
/				
1442	2200	GOODEX,	ISZ XLCHK	/INCREMENT RETURN FOR GOOD EXIT
1443	2200	ISZ	XLCHK	/INCREMENT RETURN
1444	5600	JMP I	XLCHK	/EXIT - VALUE WITHIN LIMITS
/				
1445	4424	BADEX1,	MESSAGE	/"LIMIT EXCEEDED!"
1446	2523		LINEXD	
1447	4424		MESSAGE	
1450	2534		HMSG	/"HIGH:"
1451	7240		CLA CMA	
1452	1200	TAD	XLCHK	
1453	3200	DCA	XLCHK	/BACK UP POINTER TO UPPER LIMIT
1454	1600	TAD I	XLCHK	/GET UPPER LIMIT
1455	4436	DECPRT		/PRINT UPPER LIMIT
1456	4424		MESSAGE	
1457	2540		LMSG	/"LOW:"
1460	2200	ISZ	XLCHK	/INCREMENT TO LOWER LIMIT
1461	1600	TAD I	XLCHK	/GET LOWER LIMIT
1462	4436	DECPRT		/PRINT LOWER LIMIT
1463	4434	CRLF		/<CR> AND <LF>
1464	2200	ISZ	XLCHK	/INCREMENT FOR BAD RETURN
1465	5600	JMP I	XLCHK	/EXIT

```

1466 0000 /UPCHK, 0
1467 0000 /CHKTMP, 0
/
1470 0000 XVT278, 0
1471 6031 KSF
1472 6030 KCF
1473 5670 JMP I XVT278

/
/Routine to wait for keyboard input
/call console package to check for control character
/and return with 8 bit ASCII code in AC
/
/ CALLED BY: INPUT
/
1474 0000 XINPUT, 0
1475 6000 SKON /SKIP IF INTERRUPT ON AND TURN OFF
1476 7301 CLA CLL IAC /INTERRUPT NOT ON SET BIT 11
1477 1377 TAD (ION /GET ION INSTRUCTION
1500 3323 DCA INPINT /SAVE ION OR IOF INSTRUCTION FOR EXIT
1501 6214 RDF /READ DF
1502 1072 TAD KXCDI /FORM CDI INSTRUCTION FOR EXIT
1503 3322 DCA INPCDI /SAVE CDI FOR EXECUTION
1504 6211 CDF 10 /DF=1
1505 6102 SPL /SKIP ON POWER LOW
1506 7410 SKP /SKIP IF POWER OK
1507 6001 ION /TURN ON INTERRUPT SYSTEM, POWER IS LOW
1510 6031 KSF /SKIP ON KEYBOARD FLAG
1511 5305 JMP .-4 /WAIT FOR KYBD FLG OR POWER LOW
1512 6036 KRB /READ KEYBOARD BUFFER
1513 0060 AND K0177 /MASK TO 7 BITS
1514 1376 TAD (200 /SET PARITY BIT
1515 3044 DCA INCHAR /SAVE ASCII CHAR
1516 1044 TAD INCHAR /GET CHARACTER
1517 4440 C0CAL /CHECK FOR CONTROL CHARACTER
1520 7000 NOP /CONSOLE NOT ACTIVE RETURN
1521 1044 TAD INCHAR /GET CHARACTER
1522 7402 INPCDI, HLT/CDI
1523 7402 INPINT, HLT/ION/IOF
1524 5674 JMP I XINPUT /EXIT

/
/
/
/*****
/
/Routine to set "TBLPT" to point to entry in drive state table
/
/ CALLED BY: SETPT
/ FOLLOWED BY: DSTATE (EXAMPLE OF TABLE ENTRY NAME)
/

```

```

1525 0000 PTSET, 0
1526 3351 DCA ACSAV /SAVE AC
1527 7004 RAL /GET LINK
1530 3352 DCA LKSAV /SAVE LINK
1531 1375 TAD (DRVPNT /GET ADDRESS OF DRV STATE INDEX TABLE
1532 1053 TAD CURDV /ADD CURRENT DRIVE # AS INDEX (0-3)
1533 3047 DCA TBLPT /SAVE POINTER TO INDEX TABLE
1534 6201 CDF 00
1535 1447 TAD I TBLPT /GET DRIVE STATE TABLE ADDRESS FROM INDEX TABLE
1536 3047 DCA TBLPT /SAVE AS POINTER
1537 6211 CDF 10
1540 1725 TAD I PTSET /GET DRV STATE TABLE OFFSET FROM CALL+1
1541 1047 TAD TBLPT /ADD OFFSET TO START OF DRV STATE TABLE ADDRESS
1542 3047 DCA TBLPT /SET UP DRV TBL POINTER
1543 2325 ISZ PTSET /INCREMENT FOR RETURN
1544 1352 TAD LKSAV /GET LINK
1545 7010 RAR /RESTORE LINK
1546 1351 TAD ACSAV /GET AC UPON ENTRY
1547 6201 CDF 00
1550 5725 JMP I PTSET /RETURN

```

```

1551 0000 ACSAV, 0
1552 0000 LKSAV, 0

```

```

1575 7400
1576 0200
1577 6001
1600

```

PAGE

```

/*****
/
/
/
/ROUTINE TO CONVERT OCTAL TO DECIMAL OUTPUT - SINGLE WORD PRECISION
/CALLED WITH OCTAL NUMBER TO OUTPUT IN AC
/
/ CALLED BY: DECPRT
/
/
/

```

```

1600 0000 PRTDEC, 0
1601 3237 DCA DTEMP1 /SAVE BINARY NUMBER TO OUTPUT
1602 3236 DCA DECDIG /CLEAR DECIMAL DIGIT STORAGE
1603 1066 TAD M04
1604 3240 DCA DCNT1 /SET UP DIGIT COUNTER FOR 4 DIGITS
1605 1231 TAD ADDRSA /GET POWER OF TEN TABLE ACCESS INSTRUCTION
1606 3213 DCA DPNT1 /INIT POWER OF TEN TABLE ACCESS INSTRUCTION
1607 7410 SKP /NEXT INSTRUCTION NOT USED YET
1610 3237 DCA DTEMP1 /SAVE PARTIAL QUOTIENT
1611 7100 CLL
1612 1237 TAD DTEMP1 /GET BINARY NUMBER TO CONVERT
1613 1232 DPNT1, TAD TENPWR /SUBTRACT POWER OF TEN
1614 7430 SZL
1615 2236 ISZ DECDIG /INCREMENT CONVERTED DIGIT

```

```

1616 7430      SZL
1617 5210      JMP      DPNT1-3      /CONTINUE CONVERTING THIS DIGIT
1620 7200      CLA
1621 1236      TAD      DECDIG      /GET CONVERTED DIGIT
1622 1064      TAD      K0260      /ADD ASCII BASE CODE
1623 4432      PRINT     /PRINT CONVERTED DIGIT
1624 3236      DCA      DECDIG      /CLEAR CONVERTED DIGIT STORAGE
1625 2213      ISZ      DPNT1      /INCREMENT POWER OF TEN TABLE ACCESS INSTRUCTION
1626 2240      ISZ      DCNT1      /INCREMENT DIGIT COUNTER
1627 5212      JMP      DPNT1-1    /START NEXT DIGIT
1630 5600      JMP I     PRTDEC     /EXIT

/
1631 1232      ADDRSA, TAD      TENPWR      /POWER OF TEN TABLE ACCESS INSTRUCTION
/
1632 6030      TENPWR, -1750      /-1000
1633 7634      -144              /-100
1634 7766      -12               /-10
1635 7777      -1                /-1

/
1636 0000      DECDIG, 0
1637 0000      DTEMP1, 0
1640 0000      DCNT1, 0

/
/
/*****
/
/*****
/
/ROUTINE TO PRINT STATUS MESSAGES
/
/      CALLED BY:      STAPRT
/      FOLLOWED BY:     ERTBL1      (EXAMPLE ERROR MESSAGE TABLE ADDRESS)
/
/FORMAT OF THE TABLED ERROR MESSAGES:
/
/      ERTBL1, ERMSG      /LABEL OF START OF MESSAGES, LABEL OF TEXT MESSAGE
/      ACPRNT             /LABEL OF APPROPRIATE ROUTINE TO JMS TO
/      ERREG              /ADDRESS OF DATA TO PRINT
/      CAMSG              /LABEL OF TEXT MESSAGE
/      ACPRNT             /LABEL OF APPROPRIATE ROUTINE TO JMS TO
/      COMDA              /COMMAND A REGISTER STORAGE ADDRESS
/      0                  /END OF MESSAGE TERMINATOR

/
/
1641 0000      PRNSTA, 0
1642 7300      CLA CLL
1643 6214      RDF          /READ DATA FLD
1644 1072      TAD      KXCDI      /ADD CDI INSTRUCTION FOR RETURN
1645 3326      DCA      STACDI      /SAVE FOR EXIT
1646 1641      TAD I     PRNSTA      /GET START OF MESSAGE ADDRESS
1647 3331      DCA      EPNTR      /SAVE AS POINTER
1650 6201      CDF      00
1651 1735      TAD I     CDRV       /GET CURRENT DRIVE NUMBER

```

1652	3053	DCA	CURDV	/SAVE IN FLD 1 LOCATION
1653	6211	CDF	10	/DF=1
1654	1067	MLP1,	TAD M06	
1655	3332	DCA	LCNTR	/SET UP LINE COUNTER
1656	1731	MLP2,	TAD I EPNTR	/GET MESSAGE ADDRESS
1657	7450	SMA		/SKIP IF NOT END OF MSG
1660	5324	JMP	STAOUT	/END OF MSG EXIT
1661	3263	DCA	MADR	/SAVE MSG ADDRESS
1662	4424	MESSAGE		/CALL MESSAGE PRINT ROUTINE
1663	7402	MADR,	HLT/MSG ADRS	/MESSAGE ADDRESS
1664	2331	ISZ	EPNTR	/INCREMENT MSG TBL POINTER
1665	1731	TAD I	EPNTR	/GET ADDRESS OF SUBROUTINE TO CALL FOR DATA PRINT
1666	3330	DCA	PDATA	/SAVE ADDRESS OF PRINT SUBROUTINE
1667	2331	ISZ	EPNTR	/INCREMENT MSG TBL POINTER
1670	1731	TAD I	EPNTR	/GET ADDRESS OF DATA TO PRINT
1671	7500	SMA		/SKIP IF GREATER THAN OR EQUAL TO 4000
1672	5276	JMP	GETDAT	/GO GET DATA TO PRINT
1673	1334	TAD	MDRVO	/ADD +DRIVE0 TO -(DRIVE0+OFFSET) = OFFSET
1674	7500	SMA		/SKIP IF DATA NOT IN DRIVE STATE TBL
1675	5305	JMP	TBLPRN	/GO GET DATA FROM DRV STATE TBL
1676	7300	GETDAT,	CLA CLL	
1677	1731	TAD I	EPNTR	/GET DATA ADDRESS
1700	3333	DCA	EPNTR1	/SAVE AS POINTER
1701	6201	CDF	00	
1702	1733	TAD I	EPNTR1	/GET DATA TO PRINT
1703	6211	CDF	10	
1704	5312	JMP	PRNDAT	/GO PRINT DATA
1705	3307	TBLPRN,	DCA TBLSET	/SAVE FOR SETPT CALL
1706	4437	SETPT		
1707	7402	TBLSET,	HLT/TBL INDX	
1710	1447	TAD I	TBLPT	/GET VALUE FROM DRV STATE TBL
1711	6211	CDF	10	
1712	4730	PRNDAT,	JMS I PDATA	/GO TO ROUTINE TO PRINT DATA
1713	2331	ISZ	EPNTR	/INCREMENT MESSAGE TABLE POINTER
1714	1063	TAD	K0240	
1715	4432	PRINT		/ONE SPACE
1716	1063	TAD	K0240	
1717	4432	PRINT		/TWO SPACES
1720	2332	ISZ	LCNTR	/INCREMENT LINE COUNTER
1721	5256	JMP	MLP2	/CONTINUE THIS LINE
1722	4434	CRLF		/<CR LF>
1723	5254	JMP	MLP1	/START NEXT LINE
1724	4434	STAOUT,	CRLF	/<CR LF>
1725	2241	ISZ	PRNSTA	/INCREMENT RETURN
1726	7402	STACDI,	HLT/CDI	/MODIFIED CDI TO CALLING FIELD
1727	5641	JMP I	PRNSTA	/RETURN
1730	0000	PDATA,	0	/ADDRESS OF PRINT ROUTINE
1731	0000	EPNTR,	0	/ERROR MESSAGE TABLE POINTER
1732	0000	LCNTR,	0	/LINE COUNTER
1733	0000	EPNTR1,	0	/USED FOR DOUBLE INDIRECT ADDRESSING
1734	0374	MDRVO,	-DRIVE0	
1735	0024	CDRV,	CURDRV	


```

/
/
/
/*****
1736 2246 ERTBL1, ERMSG      /ER:
1737 0637        ACPRNT    /PRNT 4 OCT DIG
1740 0030        ERREG     /ERROR REGISTER STORAGE
1741 2251        CAMSG     /CA:
1742 0637        ACPRNT
1743 0031        COMDA     /COMMAND A REG. STORAGE
1744 2254        CBMSG     /CB:
1745 0637        ACPRNT
1746 0032        COMDB     /COMMAND B REG. STORAGE
1747 2240        SAMSG     /SA:
1750 0637        ACPRNT
1751 0033        ENDSC     /FINAL SECTOR REGISTER STORAGE
1752 2270        IWCMSG    /IWC:
1753 0637        ACPRNT
1754 7423        DRIVE0+WRDCNT /INITIAL WORD COUNT STORAGE ADDRESS
1755 2273        FNCMSG    /FWC:
1756 0637        ACPRNT
1757 0034        ENDWD     /FINAL WORD COUNT READ FROM CONTROLLER
1760 2257        CFMSG     /CF:
1761 0637        ACPRNT
1762 0112        FLGSAV    /CONTROLLER FLAGS LAST INTERRUPT
1763 0000        0         /END OF MESSAGES

/
1764 2262 ERTBL2, HEMSG    /HE:
1765 1600        PRTEC     /ADDRESS OF PRINT DECIMAL ROUTINE
1766 7407        DRIVE0+HRDERR /DRIVE STATE TABLE STARTING ADDRESS PLUS TABLE OFFSET
1767 2265        SEMSG     /SE:
1770 1600        PRTEC
1771 7410        DRIVE0+SFTERR /SOFT ERROR STORAGE
1772 2304        DEMSG     /DE:
1773 1600        PRTEC
1774 7411        DRIVE0+DRVERR /DRIVE ERROR STORAGE
1775 2307        SKEMSG    /SKE:
1776 1600        PRTEC
1777 7412        DRIVE0+SEKERR /SEEK ERROR STORAGE
2000 2353        DATAM     /DAT:
2001 1600        PRTEC
2002 7413        DRIVE0+DATERR /DATA ERROR STORAGE ADDRESS
2003 2312        TEMSG     /TE:
2004 1600        PRTEC
2005 7414        DRIVE0+TRKERR /TRACKING ERROR STORAGE
2006 2315        DCRCM     /DCRC:
2007 1600        PRTEC
2010 7415        DRIVE0+DCRCER /DATA CRC ERROR STORAGE ADDRESS
2011 2321        HRCM      /HCRC:
2012 1600        PRTEC
2013 7416        DRIVE0+HRCER /HEADER CRC ERROR STORAGE ADDRESS
2014 2325        DLTMSG    /DLT:
2015 1600        PRTEC
2016 7417        DRIVE0+DLTERR /DATA LATE ERROR TALLY STORAGE ADDRESS

```

2017	2330	OPIMSG	/OPI:
2020	1600	PRTDEC	
2021	7420	DRIVE0+OPIERR	/OPERATION INCOMPLETE ERROR TALLY STORAGE ADDRESS
2022	2333	HNFMSG	/HNF:
2023	1600	PRTDEC	
2024	7421	DRIVE0+HNFERR	/HEADER NOT FOUND ERROR TALLY STORAGE ADDRESS
2025	2336	CEMSG	/CE:
2026	1600	PRTDEC	
2027	7422	DRIVE0+CTLERR	/CONTROLLER ERROR TALLY STORAGE ADDRESS
2030	0000	0	/END OF MESSAGE TERMINATOR

/			
2031	2276	ERTBL3, SIMSG	/S1:
2032	0637	ACPRNT	/ADDRESS OF ROUTINE TO PRINT AC CONTENTS
2033	0026	STAT6A	/STATUS WORD 1 STORAGE ADDRESS AFTER GET STATUS
2034	2301	S2MSG	/S2:
2035	0637	ACPRNT	
2036	0027	STAT6B	/STATUS WORD 2 STORAGE ADDRESS AFTER GET STATUS
2037	0000	0	/END OF MESSAGE TERMINATOR

/			
2040	2243	ERTBL4, DNMSG	/DN:
2041	0637	ACPRNT	/PRINT 4 OCTAL DIGITS
2042	0024	CURDRV	/CURRENT DRIVE NUMBER
2043	0000	0	/END OF TABLE

/
 /
 /
 /*****
 /
 /

/ROUTINE TO SAVE ALL NECESSARY DRIVE STATE TABLE ENTRIES
 /AND OTHER PARAMETERS

/
 / CALLED BY: JMS I (SAVALL /FROM FLD 0
 /
 /

2044	0000	SAVALL, 0	
2045	1777	TAD I	(CURDRV /GET CURRENT DRIVE FROM FLD 0
2046	3053	DCA	CURDV /SAVE FOR FLD 1 USE
2047	1776	TAD I	(FUNCOD /GET FUNCTION CODE DF=0
2050	3305	DCA	FUNSAV /SAVE FUNCTION CODE
2051	4437	SETPT	/SET TBLPT TO WORD COUNT
2052	0017	WRDCNT	/TBL INDX
2053	1447	TAD I	TBLPT /GET WORD COUNT
2054	3306	DCA	WSV /SAVE WORD COUNT
2055	4437	SETPT	/SET TBLPT TO BREAK MA
2056	0020	INITCA	/TBL INDX
2057	1447	TAD I	TBLPT /GET BREAK MA
2060	3307	DCA	ISV /SAVE BREAK MA
2061	4437	SETPT	/SET TBLPT TO SECTOR ADDRESS
2062	0021	SECADD	/TBL INDX
2063	1447	TAD I	TBLPT /GET SECTOR ADDRESS
2064	3310	DCA	SSV /SAVE SECTOR ADDRESS
2065	4437	SETPT	/SET TBLPT TO COMMAND A SENT
2066	0022	XCOMA	/TBL INDX
2067	1447	TAD I	TBLPT /GET COMMAND A SENT

2070	3311	DCA	ASV	/SAVE COMMAND A
2071	4437	SETPT		/SET TBLPT TO COMMAND B SENT
2072	0023	XCOMB		/TBL INDX
2073	1447	TAD I	TBLPT	/GET COMMAND B SENT
2074	3312	DCA	BSV	/SAVE COMMAND B
2075	4437	SETPT		/SET TBLPT TO EXPECTED FINAL SECTOR
2076	0024	XENDSC		/TBL INDX
2077	1447	TAD I	TBLPT	/GET EXPECTED FINAL SECTOR
2100	3313	DCA	ENDSV	/SAVE EXPECTED FINAL SECTOR
2101	1775	TAD I	(ERREG	/GET ERROR REGISTER AT TIME OF ERROR
2102	3314	DCA	ERSV	/SAVE ERROR REGISTER AT TIME OF ERROR
2103	6203	CDI	00	/RETURNING TO FLD 0
2104	5644	JMP I	SAVALL	/RETURN TO FLD 0

2105	0000	FUNSAV, 0	/FUNCOD
2106	0000	WSV, 0	/WRDCNT
2107	0000	ISV, 0	/INITCA
2110	0000	SSV, 0	/SECADD
2111	0000	ASV, 0	/XCOMA
2112	0000	BSV, 0	/XCOMB
2113	0000	ENDSV, 0	/XENDSC
2114	0000	ERSV, 0	/ERREG

/ROUTINE TO RESTORE DRIVE STATE TABLE ENTRIES

/ CALLED BY: JMS I (RESALL /FROM FLD 0

2115	0000	RESALL, 0	
2116	7300	CLA	CLL
2117	1305	TAD	FUNSAV
2120	3776	DCA I	(FUNCOD
2121	4437	SETPT	/GET FUNCTION CODE
2122	0017	WRDCNT	/DF=0
2123	1306	TAD	/SET TBLPT TO WORD COUNT
2124	3447	DCA I	/TBL INDX
2125	4437	SETPT	/GET WORD COUNT SAVED
2126	0020	INITCA	/RESTORE WORD COUNT
2127	1307	TAD	/SET TBLPT TO BREAK MA
2130	3447	DCA I	/TBL INDX
2131	4437	SETPT	/GET BREAK MA (INITIAL CURRENT ADDRESS)
2132	0021	SECADD	/RESTORE BREAK MA
2133	1310	TAD	/SET TBLPT TO SECTOR ADDRESS
2134	3447	DCA I	/TBL INDX
2135	4437	SETPT	/GET SECTOR ADDRESS SAVED
2136	0022	XCOMA	/RESTORE SECTOR ADDRESS
2137	1311	TAD	/SET TBLPT TO COMMAND A SAVED
2140	3447	DCA I	/TBL INDX
2141	4437	SETPT	/GET COMMAND A SAVED
2142	0023	XCOMB	/RESTORE COMMAND A SAVED

2143	1312	TAD	BSV	/GET COMMAND B SAVED
2144	3447	DCA I	TBLPT	/RESTORE COMMAND B SAVED
2145	4437	SETPT		/SET TBLPT TO EXPECTED FINAL SECTOR
2146	0024	XENDSC		/TBL INDX
2147	1313	TAD	ENDSV	/GET FINAL SECTOR SAVED
2150	3447	DCA I	TBLPT	/RESTORE FINAL SECTOR
2151	1314	TAD	ERSV	/GET ERROR REGISTER SAVED
2152	3775	DCA I	(ERREG	/RESTORE ERROR REGISTER
2153	6203	CDI	00	/RETURNING TO FLD 0
2154	5715	JMP I	RESALL	/RETURN

/*****

/ROUTINE TO GET HARDWARE OR SOFTWARE SWITCHES

CALLED BY: GETSR

2155	0000	SRGET,	0		
2156	7300	CLA	CLL		
2157	1021	TAD	HDW1		/GET HARDWARE CONFIG WORD 1
2160	7710	SPA	CLA		/SKIP IF USING PSEUDO SWITCH REGISTER
2161	7614	LAS	SKP		/GET HARDWARE SWITCHES
2162	1020	TAD	PSR1		/GET SOFTWARE SWITCHES
2163	5755	JMP	I SRGET		/RETURN

2175	0030
2176	0105
2177	0024
.	2200

PAGE

B A D S E C T O R F I L E

/ BAD SECTORS ON PACK - MAXIMUM OF 16

```

/EVEN WORDS CONTAIN CYLINDER ADDRESS IN BITS 4:11, TRACK BIT (HEAD SELECT) IN BIT 1
/
/ODD WORDS CONTAIN SECTOR ADDRESS IN BITS 6:11
/
/FIRST WORD OF BAD SECTOR FILE = 7777 IF NO BAD SECTORS
/
/
2200 BADSEC=. /START OF BAD SECTOR FILE
/
2200 0000 ZBLOCK 40
/
2237 BADEND=-1 /END OF BAD SECTOR FILE
/
/
/*****
/*****
/***** MESSAGES *****/
/*****
/*****
/*****
/
/
2240 2301 SAMSG, TEXT "SA: "
2241 7240
2242 0000
2243 0416 DNMSG, TEXT "DN: "
2244 7240
2245 0000
2246 0522 ERMSG, TEXT "ER: "
2247 7240
2250 0000
2251 0301 CANMSG, TEXT "CA: "
2252 7240
2253 0000
2254 0302 CBMSG, TEXT "CB: "
2255 7240
2256 0000
2257 0306 CFMSG, TEXT "CF: "
2260 7240
2261 0000
2262 1005 HEMSG, TEXT "HE: "
2263 7240
2264 0000
2265 2305 SEMSG, TEXT "SE: "
2266 7240
2267 0000
2270 1127 INCMSG, TEXT "INC: "
2271 0372
2272 4000
2273 0627 FWCMSG, TEXT "FWC: "
2274 0372
2275 4000

```

2276	2361	S1MSG, TEXT	"S1: "
2277	7240		
2300	0000		
2301	2362	S2MSG, TEXT	"S2: "
2302	7240		
2303	0000		
2304	0405	DEMSG, TEXT	"DE: "
2305	7240		
2306	0000		
2307	2313	SKEMSG, TEXT	"SKE: "
2310	0572		
2311	4000		
2312	2405	TEMSG, TEXT	"TE: "
2313	7240		
2314	0000		
2315	0403	DCRCM, TEXT	"DCRC: "
2316	2203		
2317	7240		
2320	0000		
2321	1003	HRCRCM, TEXT	"HCRC: "
2322	2203		
2323	7240		
2324	0000		
2325	0414	DLTMSG, TEXT	"DLT: "
2326	2472		
2327	4000		
2330	1720	OPIMSG, TEXT	"OPI: "
2331	1172		
2332	4000		
2333	1016	HNFMMSG, TEXT	"HNF: "
2334	0672		
2335	4000		
2336	0305	CEMSG, TEXT	"CE: "
2337	7240		
2340	0000		
2341	2204	RDSMSG, TEXT	"RDS: "
2342	2372		
2343	4000		
2344	2722	WRTMSG, TEXT	"WRTS: "
2345	2423		
2346	7240		
2347	0000		
2350	2313	SKSMMSG, TEXT	"SKS: "
2351	2372		
2352	4000		
2353	0401	DATAN, TEXT	"DAT: "
2354	2472		
2355	4000		
2356	2003	PCMSG, TEXT	"PC: "
2357	7240		
2360	0000		
2361	4306	FATL, TEXT	"#FATAL"
2362	0124		
2363	0114		
2364	0000		

2365	1005	HDRMSG, TEXT	"HEADER"
2366	0104		
2367	0522		
2370	0000		
2371	0401	DATMSG, TEXT	"DATA"
2372	2401		
2373	0000		
2374	0201	BAGBN, TEXT	"BA: GOOD BAD"
2375	7240		
2376	4040		
2377	4007		
2400	1717		
2401	0440		
2402	4002		
2403	0104		
2404	0000		
2405	0103	ACMSG, TEXT	"AC: "
2406	7240		
2407	0000		
2410	2516	UNIT, TEXT	"UNIT "
2411	1124		
2412	4000		
2413	4060	ZMSG, TEXT	" 0 "
2414	4000		
2415	4075	UNTDREV, TEXT	" = DRIVE (0-7)? "
2416	4004		
2417	2211		
2420	2605		
2421	4050		
2422	6055		
2423	6751		
2424	7740		
2425	0000		
2426	4017	QONSY, TEXT	" ON THIS SYSTEM "
2427	1640		
2430	2410		
2431	1123		
2432	4023		
2433	3123		
2434	2405		
2435	1540		
2436	0000		
2437	4061	ONMSG, TEXT	" 1 "
2440	4000		
2441	4062	TWMSG, TEXT	" 2 "
2442	4000		
2443	4063	THRMSG, TEXT	" 3 "
2444	4000		
2445	4323	FRSSTP, TEXT	"#START WITH STEP NUMBER (1-"
2446	2401		
2447	2224		
2450	4027		
2451	1124		
2452	1040		
2453	2324		

2454	0520		
2455	4016		
2456	2515		
2457	0205		
2460	2240		
2461	5061		
2462	5500		
2463	5177	PARENQ, TEXT	")? "
2464	4000		
2465	4323	SEKMSG, TEXT	"#SEEK "
2466	0505		
2467	1340		
2470	0000		
2471	4050	YORN, TEXT	" (Y/N)? "
2472	3157		
2473	1651		
2474	7740		
2475	0000		
2476	4005	ERR, TEXT	" ERROR"
2477	2222		
2500	1722		
2501	0000		
2502	4050	DC6061, TEXT	" (60,61) "
2503	6660		
2504	5466		
2505	6151		
2506	4000		
2507	0422	DRVMSG, TEXT	"DRIVE "
2510	1126		
2511	0540		
2512	0000		
2513	4343	RUSURE, TEXT	"#ARE YOU SURE"
2514	0122		
2515	0540		
2516	3117		
2517	2540		
2520	2325		
2521	2205		
2522	0000		
2523	4314	LIMEXD, TEXT	"#LIMIT EXCEEDED! "
2524	1115		
2525	1124		
2526	4005		
2527	3003		
2530	0505		
2531	0405		
2532	0441		
2533	4000		
2534	4010	HIMSG, TEXT	" HIGH: "
2535	1107		
2536	1072		
2537	4000		
2540	4014	LOMSG, TEXT	" LO: "
2541	1772		
2542	4000		

2543	4303	CNTRD1, TEXT	"#CAN'T READ FACTORY"
2544	0116		
2545	4724		
2546	4022		
2547	0501		
2550	0440		
2551	0601		
2552	0324		
2553	1722		
2554	3100		
2555	4002	BDSEC, TEXT	" BAD SECTOR FILE "
2556	0104		
2557	4023		
2560	0503		
2561	2417		
2562	2240		
2563	0611		
2564	1405		
2565	4000		
2566	4303	CNTRD2, TEXT	"#CAN'T READ FIELD"
2567	0116		
2570	4724		
2571	4022		
2572	0501		
2573	0440		
2574	0611		
2575	0514		
2576	0400		
2577	4076	PAKBAD, TEXT	" > 16 BAD SECTORS#"
2600	4061		
2601	6640		
2602	0201		
2603	0440		
2604	2305		
2605	0324		
2606	1722		
2607	2343		
2610	0000		
2611	1617	NOTRDY, TEXT	"NOT READY#"
2612	2440		
2613	2205		
2614	0104		
2615	3143		
2616	0000		
2617	4301	TITLE, TEXT	"#AJRLJB RL8A/RL02 DRIVE COMPATIBILITY TEST#"
2620	1222		
2621	1412		
2622	0240		
2623	2214		
2624	7001		
2625	5722		
2626	1460		
2627	6240		
2630	0422		
2631	1126		

2632	0540		
2633	0317		
2634	1520		
2635	0124		
2636	1102		
2637	1114		
2640	1124		
2641	3140		
2642	2405		
2643	2324		
2644	4300		
2645	0405	DCODE, TEXT	"DEVICE CODE"
2646	2611		
2647	0305		
2650	4003		
2651	1704		
2652	0500		
2653	4320	PWRFAL, TEXT	"PWR FAIL AT PC: "
2654	2722		
2655	4006		
2656	0111		
2657	1440		
2660	0124		
2661	4040		
2662	2003		
2663	7240		
2664	4000		
2665	4316	NOINT, TEXT	"NO INTERRUPT REQUEST"
2666	1740		
2667	1116		
2670	2405		
2671	2222		
2672	2520		
2673	2440		
2674	2205		
2675	2125		
2676	0523		
2677	2400		
2700	2516	UNINT, TEXT	"UNKNOWN INTERRUPT"
2701	1316		
2702	1727		
2703	1640		
2704	1116		
2705	2405		
2706	2222		
2707	2520		
2710	2400		
2711	4027	WRTPM, TEXT	"WRITE PROTECTED"
2712	2211		
2713	2405		
2714	4020		
2715	2217		
2716	2405		
2717	0324		
2720	0504		

2721	4300		
2722	4305	ERRPCM, TEXT	"#ERROR AT PC: "
2723	2222		
2724	1722		
2725	4001		
2726	2440		
2727	4020		
2730	0372		
2731	4000		
2732	4302	BADPAK, TEXT	"#BAD PACK!#"
2733	0104		
2734	4020		
2735	0103		
2736	1341		
2737	4300		
2740	4303	CHNGP, TEXT	"#CHANGE PACK#"
2741	1001		
2742	1607		
2743	0540		
2744	2001		
2745	0313		
2746	4300		
2747	4324	TSTCOM, TEXT	"#TEST COMPLETE - ALL DRIVES COMPATIBLE!#"
2750	0523		
2751	2440		
2752	0317		
2753	1520		
2754	1405		
2755	2405		
2756	4055		
2757	4001		
2760	1414		
2761	4004		
2762	2211		
2763	2605		
2764	2340		
2765	0317		
2766	1520		
2767	0124		
2770	1102		
2771	1405		
2772	4143		
2773	0000		
2774	4315	MOUNT, TEXT	"#MOUNT PACK ON DRIVE "
2775	1725		
2776	1624		
2777	4020		
3000	0103		
3001	1340		
3002	1716		
3003	4004		
3004	2211		
3005	2605		
3006	4000		
3007	4301	ADJMSG, TEXT	"#ADJACENT CYLINDER TEST ERROR#"

3010	0412		
3011	0103		
3012	0516		
3013	2440		
3014	0331		
3015	1411		
3016	1604		
3017	0522		
3020	4024		
3021	0523		
3022	2440		
3023	0522		
3024	2217		
3025	2243		
3026	0000		
3027	4001	ADJTO, TEXT	" ADJACENT TO "
3030	0412		
3031	0103		
3032	0516		
3033	2440		
3034	2417		
3035	4000		
3036	4327	WRNS, TEXT	"#WRITTEN SECTOR: "
3037	2211		
3040	2424		
3041	0516		
3042	4023		
3043	0503		
3044	2417		
3045	2272		
3046	4000		
3047	4301	ADJS, TEXT	"#ADJACENT SECTOR: "
3050	0412		
3051	0103		
3052	0516		
3053	2440		
3054	2305		
3055	0324		
3056	1722		
3057	7240		
3060	0000		
3061	4327	WRNT, TEXT	"#WRITTEN TRACK: "
3062	2211		
3063	2424		
3064	0516		
3065	4024		
3066	2201		
3067	0313		
3070	7240		
3071	0000		
3072	4301	ADJT, TEXT	"#ADJACENT TRACK: "
3073	0412		
3074	0103		
3075	0516		
3076	2440		

3077	2422		
3100	0103		
3101	1372		
3102	4000		
3103	4323	SURFAC, TEXT	"#SURFACE: "
3104	2522		
3105	0601		
3106	0305		
3107	7240		
3110	0000		
3111	4011	INMSG, TEXT	" IN#"
3112	1643		
3113	0000		
3114	4017	OUTMSG, TEXT	"# OUT#"
3115	2524		
3116	4300		
3117	4317	OVRMSG, TEXT	"#OVER-WRITE TEST ERROR#"
3120	2605		
3121	2255		
3122	2722		
3123	1124		
3124	0540		
3125	2405		
3126	2324		
3127	4005		
3130	2222		
3131	1722		
3132	4300		
3133	4017	OVER, TEXT	" OVER "
3134	2605		
3135	2240		
3136	0000		
3137	4323	SECMSG, TEXT	"#SECTOR: "
3140	0503		
3141	2417		
3142	2272		
3143	4000		
3144	4324	TRKMSG, TEXT	"#TRACK: "
3145	2201		
3146	0313		
3147	7240		
3150	0000		
3151	4024	THISYS, TEXT	" THIS SYSTEM#"
3152	1011		
3153	2340		
3154	2331		
3155	2324		
3156	0515		
3157	4300		
3160	4017	OTHRSY, TEXT	" OTHER SYSTEM#"
3161	2410		
3162	0522		
3163	4023		
3164	3123		
3165	2405		

3166	1543		
3167	0000		
3170	4327	WAITM, TEXT	"#WAIT FOR DRIVE READY#"
3171	0111		
3172	2440		
3173	0617		
3174	2240		
3175	0422		
3176	1126		
3177	0540		
3200	2205		
3201	0104		
3202	3143		
3203	0000		
3204	4324	TYPGR, TEXT	"#TYPE <RETURN> TO CONTINUE . . ."
3205	3120		
3206	0540		
3207	7422		
3210	0524		
3211	2522		
3212	1676		
3213	4024		
3214	1740		
3215	0317		
3216	1624		
3217	1116		
3220	2505		
3221	4056		
3222	4056		
3223	4056		
3224	0000		
3225	4327	WRNGPK, TEXT	"#WRONG PACK!"
3226	2217		
3227	1607		
3230	4020		
3231	0103		
3232	1341		
3233	0000		
3234	4320	SNNCMP, TEXT	"#PACK SERIAL NUMBERS DO NOT MATCH!#"
3235	0103		
3236	1340		
3237	2305		
3240	2211		
3241	0114		
3242	4016		
3243	2515		
3244	0205		
3245	2223		
3246	4004		
3247	1740		
3250	1617		
3251	2440		
3252	1501		
3253	2403		
3254	1041		

NOUNIT, TEXT "NO UNITS TO TEST ON THIS SYSTEM!"

```

*****
*****
****          *****
**** CONSOLE PACKAGE ****
****          *****

```

CONTROL G

```

/ WHEN CONTROL G IS TYPED, THE PROGRAM WILL PRINT AN UPARROW
/ FOLLOWED BY A G TO INDICATE THAT IT IS RESPONDING TO A CONTROL G.
/ THE PROGRAM WILL THEN EXECUTE A CARRIAGE RETURN AND LINE FEED AND
/ PRINT SR=XXXX WHERE XXXX IS THE 4 OCTAL DIGITS REPRESENTING THE
/ CURRENT CONTENTS OF THE SWITCH REGISTER BEING USED. THE PROGRAM WILL
/ THEN SPACE OVER TWO POSITIONS AND WAIT FOR THE OPERATOR TO TYPE A
/ CHARACTER OR CHARACTERS. THE FOLLOWING IS A LIST OF POSSIBLE INPUTS
/ FROM THE OPERATOR AND THE RESULT OF EACH INPUT:

```

RESULT:

```
PRINT UPARROW G,DO A CARRIAGE RETURN
AND LINE FEED, AND PRINT SR=XXXX, WAIT
FOR INPUT FROM OPERATOR
```

RETURN TO MONITOR

**RETURN TO THE PROGRAM, NO CHANGE TO
PSEUDO SWITCH REGISTER, CPU FLAGS
AND STATUS RESTORED**

RESTART PROGRAM, NO CHANGE TO PSEUDO
SWITCH REGISTER

**CHANGE PSEUDO SWITCH REGISTER AND
RETURN TO THE PROGRAM, CPU FLAGS
AND STATUS RESTORED**

**CHANGE PSEUDO SWITCH REGISTER AND
RESTART THE PROGRAM**

NO CHANGE TO PSEUDO SWITCH REGISTER
PRINT UPARROW G, EXECUTE A CARRIAGE
RETURN AND LINE FEED AND PRINT


```

/
/
/ALL OTHER INPUT
/((ILLEGAL CHARACTERS OR
/5 OCTAL DIGITS)
/
/
/ SR=XXXX, WAIT FOR OPERATOR INPUT
/ NO CHANGE TO PSEUDO SWITCH REGISTER
/ ECHO THE CHARACTER, PRINT A QUESTION
/ MARK, DO A CARRIAGE RETURN LINE FEED,
/ PRINT SR=XXXX, WAIT FOR OPERATOR
/ INPUT

```

```

/ CONTROL S
/

```

```

/ THIS CONTROL CHARACTER IS USED TO INHIBIT TRANSMISSION OF
/ DATA TO THE OPERATOR'S TERMINAL. WHEN CONTROL S IS TYPED BY THE
/ OPERATOR IT SHOULD NOT BE ECHOED. SOME TERMINALS HAVE HARDWARE
/ WHICH AUTOMATICALLY SENDS A CONTROL S WHEN THE TERMINAL BUFFER IS
/ FULL. THE BUFFER IS EMPTIED AS THE DATA IS PRINTED AND AFTER ALL
/ OF THE DATA IN THE BUFFER IS PRINTED, THE TERMINAL SENDS A CONTROL
/ Q SIGNIFYING IT IS READY TO RECEIVE MORE DATA.
/ IF THE OPERATOR TYPES CONTROL S WHILE THE PROGRAM IS RUNNING TESTS
/ THE PROGRAM WILL CONTINUE TO RUN. IF THE OPERATOR TYPES CONTROL S
/ WHILE A MESSAGE IS BEING PRINTED, THE MESSAGE WILL BE INTERRUPTED AND
/ THE PROGRAM WILL WAIT FOR A CONTROL Q OR CONTROL C TO BE TYPED.
/
/

```

```

/ CONTROL Q
/

```

```

/ THIS CONTROL CHARACTER IS USED TO ENABLE TRANSMISSION OF DATA
/ TO THE OPERATOR'S TERMINAL. WHEN CONTROL Q IS TYPED BY THE OPERATOR
/ IT SHOULD NOT BE ECHOED. SOME TERMINALS HAVE HARDWARE WHICH
/ AUTOMATICALLY SENDS A CONTROL Q WHEN THE TERMINAL BUFFER IS EMPTY AND
/ THE TERMINAL IS READY TO RECEIVE MORE DATA. CONTROL Q IS USED TO
/ COUNTERACT THE EFFECT OF A PREVIOUSLY TYPED CONTROL S. IF CONTROL S
/ WAS NOT IN EFFECT PRIOR TO THE TYPING OF CONTROL Q, THE CONTROL Q
/ SHOULD HAVE NO EFFECT UPON THE PROGRAM STATUS OR CPU STATUS.
/
/

```

```

3400 PAGE
/

```

3400	0000	C8TEMP, 0	/TEMPORARY WORK AREA
3401	6203	C8CDI, C1F CDF	/USED TO CREATE CDI TO PROGRAM FIELD
3402	6201	C8CDF, CDF	/USED TO CREATE CDF TO CONSOLE FIELD
3403	0000	C8SWR, 0	/SWITCH REGISTER SAVE AREA
3404	0000	C8MODE, 0	/PRINT MODE SWITCH
3405	0000	C8CNTR, 0	/USED AS COUNTER
3406	7775	C8M3, -3	/CONSTANT
3407	7774	C8M4, -4	/CONSTANT
3410	7773	C8M5, -5	/CONSTANT
3411	7770	C8M10, -10	/CONSTANT
3412	7520	C8W260, -260	/CONSTANT
3413	0007	C8K7, 0007	/CONSTANT
3414	0240	C8K240, 0240	/CONSTANT
3415	0260	C8K260, 0260	/CONSTANT

```

3416 0275 C8K275, 0275 /CONSTANT
3417 0277 C8K277, 0277 /CONSTANT
3420 0322 C8K322, 0322 /CONSTANT
3421 0323 C8K323, 0323 /CONSTANT
/
/
3422 0000 C8ENTR, 0
3423 3200 DCA C8TEMP /SAVE AC
3424 6214 ROF /READ PROGRAM FIELD
3425 1201 TAD C8CDI /ADD CDI INSTRUCTION
3426 3205 DCA C8CNTR /SAVE CDI TO PROGRAM FIELD TEMPORARILY
3427 6224 RIF /READ CONSOLE FIELD
3430 1202 TAD C8CDF /ADD CDF INSTRUCTION
3431 3241 DCA C8FLD /SAVE CDF TO CONSOLE FIELD
3432 1777 TAD I (21 /GET HCW1 FROM PROGRAM FIELD
3433 7710 SPA CLA /SKIP IF USING PSEUDO SWR
3434 7614 LAS SKP /GET HARDWARE SWR AND SKIP
3435 1776 TAD I (20 /GET PSEUDO SWR
3436 3203 DCA C8SWR /SAVE SWITCH REGISTER
3437 1045 TAD INMODE /GET MESSAGE ACTIVE FLAG
3440 3204 DCA C8MODE /SAVE MESSAGE ACTIVE FLAG
3441 7402 C8FLD, HLT /MODIFIED CDF TO CONSOLE DATA FIELD
3442 1222 TAD C8ENTR /GET RETURN ADDRESS
3443 3775' DCA C8RTN /SAVE FOR EXIT
3444 1205 TAD C8CNTR /GET CDI TO PROGRAM FIELD
3445 3774' DCA C8PFLD /SAVE CDI TO PROGRAM FIELD FOR EXIT
3446 1200 TAD C8TEMP /GET AC UPON ENTRY
3447 7440 SZA /SKIP IF IT WAS ZERO
3450 5773' JMP C8CNTL /AC NOT ZERO, GO CHECK CTRL CHAR
/
/PRINT OUT SR=XXXX WHERE XXXX IS THE CURRENT CONTENTS
/OF THE SWITCH REGISTER BEING USED (EITHER PSEUDO OR HARDWARE)
/
3451 4772' C8PSW, JMS C8CRLF /DO A <CR> AND <LF>
3452 1221 TAD C8K323 /GET ASCII CODE FOR "S"
3453 4771' JMS C8TYP /PRINT "S"
3454 1220 TAD C8K322 /GET ASCII CODE FOR "R"
3455 4771' JMS C8TYP /PRINT "R"
3456 1216 TAD C8K275 /GET ASCII CODE FOR "="
3457 4771' JMS C8TYP /PRINT "="
3460 1207 TAD C8M4 /AC=-4
3461 3205 DCA C8CNTR /SET UP OCTAL DIGIT COUNTER
3462 1203 TAD C8SWR /GET SWITCH REGISTER
3463 7004 RAL /EXTRA ROTATE FOR LINK
3464 7004 C8LOPA, RAL
3465 7006 RTL /ROTATE OCTAL DIGITS FOR PRINTING
3466 3203 DCA C8SWR /SAVE ROTATED SWR
3467 1203 TAD C8SWR /GET ROTATED SWR
3470 0213 AND C8K7 /MASK OFF DIGIT TO PRINT
3471 1215 TAD C8K260 /ADD ASCII BASE CODE
3472 4771' JMS C8TYP /PRINT AN OCTAL DIGIT
3473 1203 TAD C8SWR /GET SWR
3474 2205 ISZ C8CNTR /INCREMENT LOOP COUNTER
3475 5264 JMP C8LOPA /GO PRINT NEXT DIGIT

```

/ACCEPT KEYBOARD INPUT OF OCTAL DIGITS, <CR>, <LF>
 /CTRL/C OR CTRL/G. ALL OTHER CHARACTERS ARE INVALID
 /AND WILL BE ECHOED, FOLLOWED BY A "?",
 /A CARRIAGE RETURN, LINE FEED, AND A RESTART OF
 /THE SR=XXXX ROUTINE
 /

3476	7300	CLA CLL		
3477	1210	TAD	C8M5	/AC=-5
3500	3205	DCA	C8CNTR	/SET UP TO ACCEPT 5 CHARACTERS
3501	3770'	DCA	C8BLD	/CLEAR SWITCH REG. BUILD AREA
3502	3767'	DCA	C8FLG	/CLEAR SWR CHANGE SWITCH
3503	1214	TAD	C8K240	/GET ASCII CODE FOR SPACE
3504	4771'	JMS	C8TYP	/SPACE OVER ONE POSITION
3505	4766'	C8SRLP, JMS	C8TTY	/GO WAIT FOR KEYBOARD INPUT
3506	3200	DCA	C8TEMP	/SAVE INPUT CHARACTER
3507	1200	TAD	C8TEMP	/GET CHARACTER
3510	1365	TAD	(-203	
3511	7450	SNA		/SKIP IF NOT CTRL/C
3512	5764'	JMP	C8CTLG	/GO TO CTRL/C ROUTINE
3513	1207	TAD	C8M4	/AC=-4
3514	7450	SNA		/SKIP IF NOT CTRL/G
3515	5763'	JMP	C8CTLG	/GO TO CTRL/G ROUTINE
3516	1206	TAD	C8M3	/SUBTRACT 3
3517	7450	SNA		/SKIP IF NOT LINE FEED
3520	5762'	JMP	C8EXT1	/GO TO LINE FEED EXIT
3521	1206	TAD	C8M3	/SUBTRACT 3
3522	7650	SNA CLA		/SKIP IF NOT CARRIAGE RETURN
3523	5761'	JMP	C8EXT2	/GO TO CARRIAGE RETURN EXIT
3524	1200	TAD	C8TEMP	/GET CHARACTER
3525	4771'	JMS	C8TYP	/ECHO IT
3526	1200	TAD	C8TEMP	/GET CHARACTER
3527	1212	TAD	C8M260	
3530	7510	SPA		/SKIP IF >= TO ASCII CODE FOR ZERO
3531	5351	JMP	C8ERR	/INVALID CHARACTER NOT OCTAL DIGIT
3532	1211	TAD	C8M10	
3533	7700	SNA CLA		/SKIP IF <= ASCII CODE FOR SEVEN
3534	5351	JMP	C8ERR	/INVALID CHARACTER NOT OCTAL DIGIT
3535	7240	STA		/AC=7777
3536	3767'	DCA	C8FLG	/SET SWR CHANGE FLAG
3537	1200	TAD	C8TEMP	/GET CHARACTER
3540	0213	AND	C8K7	/MASK TO 3 BITS
3541	3200	DCA	C8TEMP	/SAVE OCTAL DIGIT
3542	1770'	TAD	C8BLD	/GET SWR BUILD AREA CONTENTS
3543	7106	CLL RTL		
3544	7004	RAL		/ROTATE TO BUILD SWR
3545	1200	TAD	C8TEMP	/ADD NEXT OCTAL DIGIT
3546	3770'	DCA	C8BLD	/SAVE NEW SWR
3547	2205	ISZ	C8CNTR	/INCREMENT OCTAL DIGIT COUNTER
3550	5305	JMP	C8SRLP	/CONTINUE ACCEPTING OCTAL DIGITS
/				
3551	7300	C8ERR, CLA CLL		
3552	1217	TAD	C8K277	/GET ASCII CODE FOR "?"
3553	4771'	JMS	C8TYP	/PRINT "?"
3554	4772'	JMS	C8CRLF	/DO A <CR> AND <LF>

3555 5251 JMP C8PSW /GO START OVER

3561 3751
3562 3734
3563 3657
3564 3665
3565 7575
3566 3714
3567 3603
3570 3600
3571 3677
3572 3726
3573 3620
3574 3752
3575 3602
3576 0020
3577 0021
3600

PAGE

3600	0000	C8BLD, 0	/SWITCH REGISTER BUILD AREA
3601	0200	C8STRT, RSTART	/ADDRESS OF START OF PROGRAM
3602	0000	C8RTN, 0	/STORAGE FOR RETURN ADDRESS
3603	0000	C8FLG, 0	/SWR CHANGE SWITCH
3604	0000	C8SFLG, 0	/CTRL/S ACTIVE FLAG
3605	0177	C8K177, 0177	/CONSTANT
3606	0200	C8K200, 0200	/CONSTANT
3607	0077	C8K77, 0077	/CONSTANT
3610	7740	C8M40, -40	/CONSTANT
3611	0100	C8K100, 0100	/CONSTANT
3612	0215	C8K215, 0215	/CONSTANT
3613	0212	C8K212, 0212	/CONSTANT
3614	0303	C8K303, 0303	/CONSTANT
3615	0307	C8K307, 0307	/CONSTANT
3616	0336	C8K336, 0336	/CONSTANT
3617	7600	C87600, 7600	/CONSTANT

CONTROL CHARACTER

DECODE ROUTINE

3620	1377	C8CNTL, TAD	(-203	
3621	7450	SNA		/SKIP IF NOT CTRL/C
3622	5265	JMP	C8CTLG	/CTRL/C TYPED EXIT TO MONITOR
3623	1376	TAD	(-4	
3624	7450	SNA		/SKIP IF NOT CTRL/G
3625	5257	JMP	C8CTLG	/CTRL/G TYPED GO PRINT "G"
3626	1375	TAD	(-12	
3627	7450	SNA		/SKIP IF NOT CTRL/Q
3630	5255	JMP	C8CTLQ	/CTRL/Q TYPED
3631	1374	TAD	(-2	

```

3632 7450      SNA
3633 5237      JMP      C8CTL5
3634 3773      DCA      C8MODE
3635 2204      ISZ      C8SFLG
3636 5274      JMP      C8ECHO
/
/CTRL/S HANDLER
/
3637 7240      C8CTL5, STA
3640 3204      DCA      C8SFLG
3641 1773      TAD      C8MODE
3642 7700      SNA CLA
3643 5352      JMP      C8PFLD
/
3644 7240      C8WAIT, STA
3645 3204      DCA      C8SFLG
3646 4314      JMS      C8TTY
3647 1377      TAD      (-203
3650 7450      SNA
3651 5265      JMP      C8CTL5
3652 1372      TAD      (-16
3653 7640      SZA CLA
3654 5244      JMP      C8WAIT
3655 3204      C8CTLQ, DCA      C8SFLG
3656 5352      JMP      C8PFLD
/
/CONTROL G HANDLER
/
3657 4326      C8CTLG, JMS      C8CRLF
3660 1216      TAD      C8K336
3661 4277      JMS      C8TYP
3662 1215      TAD      C8K307
3663 4277      JMS      C8TYP
3664 5771      JMP      C8PSW
/
/CONTROL C HANDLER
/
3665 3204      C8CTLC, DCA      C8SFLG
3666 1216      TAD      C8K336
3667 4277      JMS      C8TYP
3670 1214      TAD      C8K303
3671 4277      JMS      C8TYP
3672 6203      CIF CDF
3673 5617      JMP I      C87600
/
3674 1770      C8ECHO, TAD      C8TEMP
3675 4277      JMS      C8TYP
3676 5352      JMP      C8PFLD
/
/PRINT ONE CHARACTER
/
3677 0000      C8TYP, 0

```

```

/SKIP IF NOT CTRL/S
/CTRL/S TYPED
/SET MESSAGE ACTIVE FLAG
/TEST CTRL/S ACTIVE FLAG
/GO ECHO CHARACTER AND RETURN TO PROGRAM

```

```

/AC=7777
/SET CTRL/S ACTIVE FLAG
/GET MESSAGE ACTIVE FLAG
/SKIP IF CTRL/S TYPED WHILE MESSAGE ACTIVE
/RETURN TO PROGRAM

```

```

/AC=7777
/SET CTRL/S ACTIVE FLAG
/WAIT FOR KEYBOARD INPUT

```

```

/SKIP IF NOT CTRL/C
/CTRL/C TYPED EXIT TO MONITOR
/SKIP IF CTRL/Q
/NOT CTRL/C OR CTRL/Q CONTINUE WAITING
/CLEAR CTRL/S ACTIVE FLAG
/RETURN TO MAIN PROGRAM

```

```

/DO A <CR> AND <LF>
/GET ASCII CODE FOR UP ARROW
/PRINT UP ARROW
/GET ASCII CODE FOR "G"
/PRINT "G"
/GO TO "SR=XXXX" ROUTINE

```

```

/CLEAR CTRL/S ACTIVE FLAG
/GET ASCII CODE FOR UP ARROW
/PRINT UP ARROW
/GET ASCII CODE FOR "C"
/PRINT "C"
/CHANGE TO IF AND DF ZERO
/RETURN TO MONITOR

```

```

/GET CHARACTER
/ECHO IT
/RETURN TO PROGRAM

```

```

3700 2204      ISZ      C8SFLG      /TEST CTRL/S ACTIVE FLAG
3701 7410      SKP      /SKIP IF CTRL/S NOT ACTIVE
3702 5244      JMP      C8WAIT      /GO WAIT FOR CTRL/Q OR CTRL/C
3703 6046      TLS      /TRANSMIT CHARACTER
3704 6102      SPL      /SKIP ON POWER LOW
3705 7410      SKP      /SKIP IF POWER OK
3706 6001      ION      /TURN INTERRUPT ON - POWER IS LOW
3707 6041      TSF      /TEST TTY FLAG
3710 5304      JMP      .-4         /WAIT FOR TTY FLAG
3711 6042      TCF      /CLEAR TTY FLAG
3712 7200      CLA      /CLEAR AC DO NOT CLEAR LINK

3713 5677      JMP I    C8TYP      /RETURN
/
/WAIT FOR KEYBOARD INPUT THEN EXIT WITH ASCII CODE IN AC
/
3714 0000      C8TTY, 0
3715 6102      SPL      /SKIP ON POWER LOW
3716 7410      SKP      /SKIP IF POWER OK
3717 6001      ION      /TURN INTERRUPT ON - POWER IS LOW
3720 6031      KSF      /SKIP IF KEYBOARD FLAG SET
3721 5315      JMP      .-4         /WAIT FOR KEYBOARD INPUT
3722 6036      KRB      /READ KEYBOARD BUFFER CLEAR FLAG
3723 0205      AND      C8K177      /MASK TO 7 BITS
3724 1206      TAD      C8K200      /SET BIT 4
3725 5714      JMP I    C8TTY      /RETURN
/
/EXECUTE A CARRIAGE RETURN AND LINE FEED
/
3726 0000      C8CRLF, 0
3727 1212      TAD      C8K215      /GET ASCII CODE FOR CARRIAGE RETURN
3730 4277      JMS      C8TYP      /GO EXECUTE THE CARRIAGE RETURN
3731 1213      TAD      C8K212      /GET ASCII CODE FOR LINE FEED
3732 4277      JMS      C8TYP      /GO EXECUTE THE LINE FEED
3733 5726      JMP I    C8CRLF      /RETURN
/
/CONSOLE PACKAGE EXIT IF TERMINATED WITH LINE FEED
/
3734 4326      C8EXT1, JMS      C8CRLF      /DO A <CR> AND <LF>
3735 1352      TAD      C8PFLD      /GET MODIFIED CDI TO PROGRAM FIELD
3736 3337      DCA      .+1         /SAVE FOR EXECUTION
3737 7402      HLT/CDI      /MODIFIED CDI TO PROGRAM FIELD
3740 2203      ISZ      C8FLG      /TEST SWR CHANGE FLAG
3741 5601      JMP I    C8STRT      /RESTART PROGRAM WITHOUT CHANGE OF SWR
3742 6201      CDF      00         /CDF TO FIELD 0
3743 1200      TAD      C8BLD      /GET NEW SWITCH REGISTER
3744 3767      DCA I    (20        /SAVE IT IN FIELD 0 PAGE 0 LOC 20
3745 6211      CDF      10         /CDF TO FIELD 1
3746 1200      TAD      C8BLD      /GET NEW SWITCH REGISTER
3747 3767      DCA I    (20        /SAVE IT IN FIELD 1 PAGE 0 LOC 20
3750 5337      JMP      C8EXT1+3    /GO EXECUTE CDI INSTRUCTION AGAIN
/
/EXIT FROM CONSOLE PACKAGE IF TERMINATED WITH CARRIAGE RETURN
/
3751 4326      C8EXT2, JMS      C8CRLF      /DO A <CR> AND <LF>

```

3752	7402	C8PFLD, HLT	/MODIFIED CDI TO PROGRAM FIELD
3753	7300	CLA CLL	/CLEAR AC AND LINK FOR RETURN
3754	2203	ISZ C8FLG	/TEST SWR CHANGE FLAG
3755	5602	JMP I C8RTN	/RETURN TO PROGRAM WITHOUT CHANGE OF SWR
3756	1200	TAD C8BLD	/GET NEW SWITCH REGISTER
3757	3767	DCA I (20	/SAVE IT IN PROGRAM FIELD
3760	5602	JMP I C8RTN	/RETURN TO PROGRAM

/

/*****

/

3767 0020

3770 3400

3771 3451

3772 7762

3773 3404

3774 7776

3775 7766

3776 7774

3777 7575

5600 *BUF1 /DATA BUFFER FIELD 1

6600 *RBUF /READ DATA BUFFER FIELD 1

/

/*****

/

0000 FIELD 0

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

0200

*200

END OF PROGRAM

\$\$\$

ABORT	3435	AUTO12	0012	C8ECHO	3674	CBUSY	0055
ACAL	4334	AUTO13	0013	C8ENTR	3422	CBUSY1	0074
ACCNTR	0667	AUTO14	0014	C8ERR	3551	CDF	6201
ACHKEX	4553	AUTO15	0015	C8EXT1	3734	CDFSV	0052
ACL	7701	AUTO16	0016	C8EXT2	3751	CDI	6203
ACLP	0651	AUTO17	0017	C8FLD	3441	CDRV	1735
ACMSG	2405	BADCNT	5701	C8FLG	3603	CEMSG	2336
ACOUT	0664	BADEND	2237	C8K100	3611	CFMSG	2257
ACPRN	1062	BADEX1	1445	C8K177	3605	CHK16	3624
ACPRNT	0637	BADEX2	2552	C8K200	3606	CHK8	1213
ACSAV	1551	BADLPB	1613	C8K212	3613	CHKBC	5651
ACSAVE	2017	BADPAK	2732	C8K215	3612	CHKCNT	3654
ACSV	2331	BADPNT	0116	C8K240	3414	CHKCRC	3520
ACTMP	0666	BADPRO	1600	C8K260	3415	CHKDLT	3513
ADDRS1	0001	BADPT	5673	C8K275	3416	CHKERR	2600
ADDRS2	0002	BADSEC	2200	C8K277	3417	CHKHCR	3475
ADDRS3	0003	BADSWT	1724	C8K303	3614	CHKOUT	2673
ADDRS4	0004	BADTRK	1725	C8K307	3615	CHKRDY	3047
ADDRS5	0005	BAGBM	2374	C8K322	3420	CHKRET	1256
ADDRS6	0006	BDCNT	1726	C8K323	3421	CHKSEC	1200
ADDRS7	0007	BDPAK	5661	C8K336	3616	CHKSTA	2200
ADDRSA	1631	BDSEC	2555	C8K7	3413	CHKTMP	1467
ADJCAL	4237	BDWCNT	7317	C8K77	3607	CHNGP	2740
ADJCHK	4506	BEGTRK	4135	C8LOC	7132	CHRGET	4733
ADJDRV	4715	BFCNT	5147	C8LOPA	3464	CHRTEN	0756
ADJERR	4573	BFPT	7316	C8M10	3411	CHRTMP	7133
ADJEX	4152	BHOME	2245	C8M260	3412	CIF	6202
ADJMSG	3007	BLODE	3042	C8M3	3406	CKSOUT	2253
ADJPNT	0132	BRKLOD	3017	C8M4	3407	CLRBF1	7320
ADJS	3047	BSFEX	1510	C8M40	3610	CLRLOP	5142
ADJT	3072	BSFGET	1400	C8M5	3410	CLRLP1	7330
ADJTBL	6450	BSV	2112	C8MODE	3404	CLRRBF	5131
ADJTO	3027	BSW	7002	C8OFF	7105	CMPSN	4574
ADJU4	6745	BUF1	5600	C8OUT	0754	CNGIOT	4564
ADJUNT	4716	BUFAD1	0106	C8PFLD	3752	CNGTRK	4460
ADJWRT	4571	BUFCTR	7335	C8PSW	3451	CNT1	0050
AJ2TBL	6725	BUFPNT	1723	C8RTN	3602	CNT2	0051
AJ3TBL	6701	BUFSZ1	2000	C8SFLG	3604	CNTFND	3256
AJ4TBL	6643	BUFSZR	1000	C8SRLP	3505	CNTLOD	3011
ALLUNT	0544	C87600	3617	C8STRT	3601	CNTR1	0107
ALODE	3034	C8BLD	3600	C8SWR	3403	CNTR5	5504
ASAV	7364	C8CAL	4440	C8TEMP	3400	CNTRD1	2543
ASEC	0130	C8CALL	4535	C8TTY	3714	CNTRD2	2566
ASECM6	4243	C8CDF	3402	C8TYP	3677	CONDA	0031
ASECP8	4245	C8CDI	3401	C8WAIT	3644	COMDB	0032
ASKSTP	0546	C8CNTL	3620	CAF	6007	CONSET	0200
ASM6	4340	C8CNTR	3405	CAL	6103	CREXT	4731
ASP6	4342	C8CRLF	3726	CALC8	0733	CRLF	4434
ASV	2111	C8CTLG	3665	CALOUT	7130	CRLFDO	7200
ATRK	4510	C8CTLG	3657	CAM	7621	CROUT	1271
AUTO10	0010	C8CTLQ	3655	CAMSG	2251	CRWAIT	4717
AUTO11	0011	C8CTLS	3637	CBMSG	2254	CCHK	1151

CTLERR 0016
 CURCYL 0001
 CURDRV 0024
 CURDV 0053
 CURUNT 0035
 CYLPNT 0124
 DATAM 2353
 DATERR 0007
 DATEXA 2731
 DATGEN 2675
 DATLP1 2717
 DATMSG 2371
 DATSWT 1265
 DATWD1 1263
 DATWD2 1264
 DBLCHK 0610
 DC6061 2502
 DC60LP 3670
 DC61LP 3706
 DCEND 7554
 DCNT1 1640
 DCODE 2645
 DCR CER 0011
 DCR CM 2315
 DCTB60 7530
 DCTB61 7536
 DEC4 4427
 DEC4ER 1045
 DEC4EX 1053
 DEC4LP 1007
 DECBLO 1060
 DECDIG 1636
 DECPRN 4561
 DECPRT 4436
 DECTMP 1062
 DELAY 1066
 DEMSG 2304
 DERFLG 0117
 DERR 7230
 DETST 3307
 DIGCNT 1061
 DIGFLG 0055
 DLPA 0674
 DLPB 0677
 DLTERR 0013
 DLTMSG 2325
 DLY1 0713
 DLY2 0714
 DLY3 0715
 DLYCAL 1073
 DNCNT1 2276
 DNMSG 2243

DNWAIT 2256
 DOCR LF 4556
 DONCNT 2275
 DONLP 2263
 DPAT8 2732
 DPAT80 2732
 DPAT81 2733
 DPAT82 2734
 DPAT83 2735
 DPNT1 1613
 DRDYA 2227
 DRIVE0 7404
 DRIVE1 7431
 DRIVE2 7456
 DRIVE3 7503
 DRV0 7400
 DRV1 7401
 DRV2 7402
 DRV3 7403
 DRV60 0025
 DRVEND 7527
 DRVERR 0005
 DRVGON 3322
 DRVMSG 2507
 DRVPNT 7400
 DSCGET 0716
 DSCU0 0066
 DSCU01 0105
 DSCU1 0067
 DSCU11 0106
 DSCU2 0070
 DSCU21 0107
 DSCU3 0071
 DSCU31 0110
 DTEMP1 1637
 DTMP1 3144
 DVTMP 0046
 DWD1 2727
 DWD2 2730
 ENDSC 0033
 ENDSV 2113
 ENDTB2 6207
 ENDTB3 6157
 ENDTB4 6101
 ENDTST 0723
 ENDWD 0034
 EPNTR 1731
 EPNTR1 1733
 EREX1 2672
 EREXT1 3541
 EREXT2 3551
 ERMSG 2246

ERR 2476
 ERRADJ 4600
 ERRCHK 4542
 ERREG 0030
 ERRFLG 3557
 ERROR 4541
 ERROVR 5000
 ERRPC 3555
 ERRPCM 2722
 ERRPNT 3617
 ERSV 2114
 ERTBL1 1736
 ERTBL2 1764
 ERTBL3 2031
 ERTBL4 2040
 EX16 3653
 EXFER 3120
 FACBAD 7554
 FATAL 7336
 FATL 2361
 FINRPT 3600
 FINSEC 2661
 FLAGS 2020
 FLDBAD 7561
 FLDBD 1675
 FLDP1 3145
 FLGS 2033
 FLGSAV 0112
 FNDTRK 4566
 FOUND5 5655
 FRSSSTP 2445
 FUNCOD 0105
 FUNSAV 2105
 FNCMSG 2273
 GDCNT 5700
 GETBSF 4565
 GETCHR 4563
 GETDAT 1676
 GETDSC 4442
 GETHCW 7110
 GETOCT 4554
 GETSR 4441
 GETSTA 4547
 GETSWR 4537
 GO 4543
 GOEXIT 3045
 GOODEX 1442
 GTF 6004
 HCR CER 0012
 HCR CM 2321
 HCW1 0021
 HCW2 0022

HCW3 0023
 HDCHKA 2212
 HDRERR 5107
 HDRMSG 2365
 HDTMP 2255
 HDW1 0021
 HDW2 0022
 HDW3 0023
 HEMSG 2262
 HINMSG 2534
 HISEC 5307
 HNFERR 0015
 HNFMSG 2333
 HRDERR 0003
 INO 6450
 INCHAR 0044
 INDX10 0010
 INDX11 0011
 INDX12 0012
 INDX13 0013
 INDX14 0014
 INDX15 0015
 INDX16 0016
 INDX17 0017
 INIT 0202
 INITCA 0020
 INITLP 0204
 INM11 6455
 INM12 6467
 INM13 6513
 INM201 6605
 INM202 6612
 INM203 6617
 INM211 6561
 INM212 6566
 INM221 6501
 INM222 6547
 INM231 6525
 INM232 6531
 INMODE 0045
 INMSG 3111
 INNER0 5706
 INNER1 5713
 INP11 6462
 INP12 6474
 INP13 6520
 INP201 6624
 INP202 6631
 INP203 6636
 INP211 6573
 INP212 6600
 INP221 6506

INP222	6554	KXCDI	0072	MIDDL1	5711	ONM222	6551
INP231	6536	KYBDRD	2107	MLP1	1654	ONM231	6527
INP232	6542	LCNTR	1732	MLP2	1656	ONM232	6533
INPCDI	1522	LE17	4541	MNTPAK	4567	ONP11	6464
INPINT	1523	LIMCHK	4430	MOUNT	2774	ONP12	6476
INPUT	4435	LINEXD	2523	MOVLP	0626	ONP13	6522
INSEK	4400	LINKSV	0073	MQA	7501	ONP201	6626
INTOFF	2306	LKSAV	1552	ML	7421	ONP202	6633
INTRET	2114	LKSV	2332	MXSTP	0111	ONP203	6640
INTSVC	2000	LNK1	1063	NDE	3424	ONP211	6575
IOT60	3723	LOCO	0000	NEWCYL	0002	ONP212	6602
IOT61	3724	LOCHK	1420	NEXIT	1312	ONP221	6510
IOTA	2036	LOGDCR	3526	NOINT	2665	ONP222	6556
IOTB	2045	LOGDLT	3515	NOP61	6616	ONP231	6540
IOTC	2051	LOGERR	3612	NOPION	3460	ONP232	6544
IOTC1	3722	LOGHCR	3503	NOTBSY	2273	ONSY80	0062
IOTCNG	3656	LOGHNF	3472	NOTINT	2030	ONSY81	0063
IOTD	2055	LOGTRK	3506	NOTOPI	3511	ONSY82	0064
IOTE	2061	LOMSG	2540	NOTRDY	2611	ONSY83	0065
IOTF	2065	LOSEC	5314	NOTYN	1307	OPIERR	0014
IOTG	2334	LP16	3634	NOUNIT	3304	OPIMSG	2330
IOTP1	3720	LPA1	1226	NTRK	4466	OTHR8	3752
IOTP2	3721	LPINT1	5616	NUMUNT	0075	OTHR5Y	3160
ISV	2107	LPINT2	5620	NUMUUT	0114	OUTERO	5702
IWCMSG	2270	LPINT3	5623	NXTIO	4442	OUTER1	5707
K0177	0060	LPINT4	5625	NXTK1	5463	OUTMSG	3114
K0212	0061	LSAV	2311	NXTPG0	0400	OUTSEK	4260
K0215	0062	M04	0066	NXTPG1	0600	OUTST	4317
K0240	0063	M06	0067	NXTRK1	5454	OVER	3133
K0260	0064	M1	0052	NXTS1	4304	OVRDRV	5106
K0277	0065	M215	0070	NXTS2	4427	OVRERR	4572
K03	0056	M4	0053	NXTSEC	5400	OVREX	5330
K07	0057	M400	0054	OCT1	4426	OVRLP1	5263
K0777	0047	MADR	1663	OCT1ER	1113	OVRLP2	5275
K100	0043	MAIN	0213	OCTGET	7162	OVRMSG	3117
K1000	0050	MAINGO	0617	OCTGO	7166	OVRTB2	6210
K1410	3146	MAXSTP	0072	OCTONE	1120	OVRTB3	6160
K2000	0051	MORVO	1734	OCTOUT	1111	OVRTB4	6102
K240	2563	MESAG	4551	OFFSET	0010	OVRTBL	6214
K260	0044	MESAGE	4424	OKEX	5671	OVRUNT	0134
K3	0040	MESBSW	1231	OLDCYL	0000	OVRWRT	4570
K377	0045	MESCNT	1254	ONO	6452	PAK6	0612
K400	0046	MESEXT	1250	ONEMSG	2437	PAK7	0662
K5	0041	MESGO	2124	ONM11	6457	PAKBAD	2577
K7	0042	MESINT	1251	ONM12	6471	PAKMNT	3725
KBDCHK	2074	MESLUP	1220	ONM13	6515	PARENQ	2463
KCC	6032	MESOUT	1252	ONM201	6607	PAT80	2736
KCF	6030	MESPNT	1255	ONM202	6614	PAT81	2740
KIE	6035	MESPRT	1200	ONM203	6621	PAT82	2742
KION	0110	MESSAD	2132	ONM211	6563	PAT83	2744
KSF	6031	MESTMP	1256	ONM212	6570	PATMP	0120
KXCDF	0071	MIDDL0	5704	ONM221	6503	PATNUM	0113

PATSV 4546
 PCMSG 2356
 PCSAVE 2016
 PDATA 1730
 PNTR1 0114
 POSGO 3537
 POSPC 3556
 POSSET 2400
 PRINT 4432
 PRNAC 4555
 PRNBA 7261
 PRNDAT 1712
 PRNDEC 1056
 PRNINT 1354
 PRNOUT 1353
 PRNSTA 1641
 PRNT 4553
 PRNTAC 4431
 PRNTGO 1106
 PRNTMP 1356
 PRTDEC 1600
 PRTSTA 2341
 PSR 0020
 PSR1 0020
 PTBL16 3655
 PTSET 1525
 PWRFAL 2653
 QONSYS 2426
 QUART0 5703
 QUART1 5710
 RBUFF 6600
 RDEXA 1164
 RDF 6214
 RDFLD 1475
 RDHDR 3200
 RDSEC 1112
 RDSI1 3214
 RDSI2 3217
 RDSI3 3222
 RDSI4 3226
 RDSIA 3105
 RDSIB 3112
 RDSMSG 2341
 RDYCHK 2333
 RESALL 2115
 RESET 4560
 RESTRT 2023
 RETCNT 0115
 RETRY 3530
 RIF 6224
 RLCA 6603
 RLCB 6604

RLDC 6600
 RLMA 6602
 RLSA 6605
 RLSD 6601
 RLSE 6617
 RLWC 6607
 RMF 6244
 RRCA 6612
 RRCB 6613
 RRER 6610
 RRSA 6614
 RRSI 6615
 RRWC 6611
 RSTART 0200
 RTF 6005
 RUSURE 2513
 S1ERR 4640
 S1MSG 2276
 S2FLG 0113
 S2MSG 2301
 S2TBL 6734
 S3TBL 6717
 S4TBL 6671
 SAMSG 2240
 SAVALL 2044
 SBE 6101
 SCK1 3333
 SCK2 3400
 SCK3 3405
 SCNTSV 5506
 SEC0 6235
 SEC1 6236
 SEC2 6237
 SEC3 6240
 SEC34 6221
 SEC35 6222
 SEC36 6223
 SEC37 6224
 SEC40 6225
 SEC41 6226
 SEC42 6227
 SEC43 6230
 SEC44 6231
 SEC45 6232
 SEC46 6233
 SEC47 6234
 SECADD 0021
 SECNT 0036
 SECLD 3026
 SECLP1 5214
 SECLPA 4200
 SECLPB 4212

SECMSG 3137
 SECPNT 0123
 SECPTR 5331
 SECRDA 1160
 SECSAV 0121
 SECSV 0122
 SEEK 7134
 SEEK0 4262
 SEK1 5265
 SEK2 5273
 SEK3 5436
 SEK777 4403
 SEKDIR 0133
 SEKERR 0006
 SEKMSG 2465
 SEKOUT 5431
 SEKRET 0076
 SEKSWT 5507
 SEKTRK 4264
 SEMSG 2265
 SERNM1 0077
 SERNM2 0100
 SERNM3 0101
 SERNM4 0102
 SETCNT 5674
 SETPNT 4536
 SETPOS 4546
 SETPT 4437
 SETTMP 2021
 SFTERR 0004
 SILO1 3254
 SILO2 3255
 SINDX2 6437
 SINDX3 6431
 SINDX4 6421
 SINOUT 5272
 SKEMSG 2307
 SKON 6000
 SKRET 0115
 SKSMMSG 2350
 SNCMP 4000
 SNERR 4112
 SNNCMP 3234
 SNPROC 1702
 SOK 2551
 SPACE2 4557
 SPACES 2554
 SPARE 6606
 SPL 6102
 SPNTR 4120
 SPRNT 3303
 SRGET 2155

SRQ 6003
 SRQUES 0605
 SSV 2110
 STACDI 1726
 STACHK 4550
 STAEX2 2254
 STAGET 3056
 STAOUT 1724
 STAFRT 4540
 START 0200
 STAT6A 0026
 STAT6B 0027
 STATC 3447
 STATE4 2233
 STATE5 2240
 STATGO 2347
 STBL 6740
 STBL16 6221
 STEP1 0240
 STEP2 0275
 STEP3 0400
 STEP35 0661
 STEP4 0450
 STEP43 0610
 STEP5 0520
 STEP6 0606
 STEP7 0654
 STP021 7060
 STP022 7061
 STP023 7063
 STP031 7035
 STP032 7036
 STP033 7040
 STP034 7044
 STP035 7046
 STP041 6775
 STP042 6776
 STP043 7000
 STP044 7004
 STP045 7010
 STP046 7012
 STP047 7016
 STP12 6725
 STP124 6241
 STP13 6701
 STP14 6643
 STP1ER 5040
 STP22 6727
 STP23 6703
 STP24 6645
 STP32 6731
 STP33 6705

STP34	6647	TEMP1	0111	UNT2	0356	WRTSEC	1000
STP3U4	6261	TEMSG	2312	UNT3	0440	WRTTRK	4410
STP43	6707	TENPWR	1632	UNTDVR	2415	WSEC	0127
STP44	6651	THISYS	3151	UNTPNT	3760	WSV	2106
STP4U4	6301	THREQ0	5705	UPCHK	1466	WTRK	4265
STP53	6712	THREQ1	5712	UPLSTP	0602	WTRKM1	4234
STP54	6653	THRMSG	2443	UT01	6166	WTRKP1	4232
STP5U3	6401	TIE	6045	UT03	6202	XC8CAL	7076
STP5U4	6321	TITLE	2617	UT12	6174	XC8RET	7131
STP64	6655	TMP1	0054	UUT001	6112	XCDF1	1140
STP6U4	6341	TOOBAD	1661	UUT005	6146	XCONA	0022
STP74	6662	TRKNT	0125	UUT01	6000	XCOMB	0023
STP7U4	6361	TRKERR	0010	UUT07	6064	XCRLF	1257
STPASK	0526	TRKFND	5600	UUT102	6120	XDEC4	1000
STPCHK	4222	TRKLP2	5633	UUT104	6140	XDELAY	0670
STPCK	4330	TRKMSG	3144	UUT12	6006	XENDSC	0024
STPIDX	6443	TRKNM2	4522	UUT16	6052	XERROR	3256
STPNT	0126	TRKNP2	4526	UUT203	6126	XFER	4545
STPNUM	0103	TRKPNT	5675	UUT23	6014	XGO	3000
STRPAT	5317	TRKTRY	5677	UUT25	6044	XINPUT	1474
STRSTP	0073	TRYCNT	5676	UUT34	6026	XLCHK	1400
STRTBL	0230	TRYLPA	1415	VALNEG	1426	XOCT1	1064
STRTIN	2015	TRYLPB	1511	VT278	4443	XPRINT	1316
STSTP	0112	TRYOCT	1072	WAITCR	4562	XRESET	1266
STTBL2	6166	TRYYN	1300	WAITDN	4544	XSETEX	2327
STTBL3	6112	TST1	0265	WAITM	3170	XSETP	2277
STTBL4	6000	TST2	0323	WAITOP	3754	XTFIL	1121
SURCNT	5505	TST3	0431	WATCNT	7226	XTLP	1141
SURFO	5702	TST4	0501	WATCT1	7227	XTST1	0271
SURF1	5707	TST5	0551	WATLPA	7211	XTST2	0334
SURFAC	3103	TST6	0634	WATMOR	7220	XTST3	0442
SURSWT	4505	TST7	0704	WATRDY	7204	XTST4	0512
SWP	7521	TSTCOM	2747	WCHDR	1334	XTST5	0600
SWRGET	1047	TTYCHK	2077	WDCNTR	0037	XTST6	0646
SYS2F	0074	TWOMSG	2441	WDCTR	1262	XTST7	0715
SYSCHK	0512	TYPCR	3204	WNXSEC	4412	XVT278	1470
SYSON0	0101	U2CHK	0411	WRDCNT	0017	XYESNO	1273
SYSON1	0102	U31CHK	0536	WRNGPK	3225	YESRN	4552
SYSON2	0103	U3CHK	0461	WRT1	1006	YESRNO	4425
SYSON3	0104	UNINT	2700	WRTADJ	4121	YEXIT	1313
TABED	0116	UNIT	2410	WRTXA	1046	YNGO	1077
TABFIL	4433	UNIT0	0056	WRTLOC	1314	YNOUT	1314
TABSTR	0074	UNIT01	0075	WRTMSG	2344	YORN	2471
TBLED	0077	UNIT1	0057	WRTNS	3036	ZMSG	2413
TBLIN	5233	UNIT11	0076	WRTNT	3061	ZWRD	2656
TBLPNT	0104	UNIT2	0060	WRTOVR	5200		
TBLPRN	1705	UNIT21	0077	WRTPM	2711		
TBLPT	0047	UNIT3	0061	WRTPNT	0131		
TBLSAV	2022	UNIT31	0100	WTRK	7071		
TBLSET	1707	UNKINT	2121	WTRK2	7065		
TBLSTR	0055	UNTO	0250	WTRK3	7052		
TBLWRD	1150	UNT1	0313	WTRK4	7025		

(

(

(

(

(

1