

**IDENTIFICATION**  
-----

**PRODUCT CODE:** AC-F374B-MA  
-----

**PRODUCT NAME:** AJRLKB0 RL8-A/RL02 PERFORMANCE EXERCISER  
-----

**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
- 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 APT INITIALIZATION
- 4.3 ACCEPT MODE
- 4.3.1 OPERATOR PROMPTS/RESPONSE
- 4.3.2 PARAMETERS
- 4.3.3 INITIAL WRITE/READ OF PACKS
- 4.3.4 DATA PATTERNS
- 4.3.5 END OF PASS (SEEKS)
- 4.3.6 END OF PASS (DATA)
- 4.3.7 DRIVE ACCEPTANCE
- 4.4 MANUAL INTERVENTION MODE
- 4.4.1 OPERATOR PROMPTS/RESPONSE
- 4.4.2 PARAMETER CONFLICTS
- 4.4.3 INITIAL WRITE/READ OF PACKS
- 4.4.4 END OF PASS
- 4.5 REQUIREMENTS
- 4.6 OPERATION OF CONSOLE PACKAGE
- 5.0 ERRORS
- 5.1 ON APT ERRORS
- 5.2 NOT ON APT ERRORS
- 5.2.1 FATAL ERRORS
- 5.2.2 SOFT ERRORS
- 5.2.3 HARD ERRORS
- 5.2.4 ERROR LOGGING
- 5.2.5 RETRIES
- 5.2.6 ERROR REPORTING
- 5.2.7 BAD SECTOR FILES UNREADABLE
- 6.0 SWITCH REGISTER SETTINGS
- 6.1 NORMAL OPERATION SWITCHES
- 6.2 ERROR RELATED SWITCHES
- 6.3 OTHER SWITCHES
- 7.0 REVISIONS
- 8.0 ADDITIONAL INFORMATION
- 8.1 ADDING OR DROPPING DRIVES
- 8.2 HALTS
- 8.3 DRIVE DROPS
- 8.4 STATUS MESSAGES
- 8.5 POWER FAIL
- 9.0 PROGRAM LISTING

THE RL02 PERFORMANCE EXERCISER WILL EXERCISE FROM 1 TO 4 RL02 DISK DRIVES IN A MANNER WHICH SIMULATES A USER ENVIRONMENT. RANDOM OR FIXED SEEKS, READS, AND WRITES WILL BE PERFORMED IN AN ATTEMPT TO PROVE OPERATIONAL RELIABILITY OF THE RL01 SUBSYSTEM. THESE OPERATIONS ARE PERFORMED IN A RANDOM OVERLAPPED MANNER ON THE SELECTED DRIVES TO ENHANCE THROUGHPUT. PERFORMANCE STATISTICS ARE MAINTAINED FOR EACH DRIVE, INCLUDING ERROR SUMMARIES AND OPERATION COUNTS. THE OPERATOR MAY OBTAIN THESE PERFORMANCE STATISTICS ON THE CONSOLE TERMINAL BY TYPING CONTROL-R.

THE PROGRAM REQUIRES THAT AN OPERATOR'S TERMINAL BE PRESENT AND ON-LINE FOR ALL ERROR AND STATUS MESSAGES. THE STANDARD CONSOLE PACKAGE IS USED TO PERMIT USE OF A SOFTWARE SWITCH REGISTER. DRIVES UNDER TEST MAY BE ADDED TO OR DROPPED FROM BE SELECTED RANDOMLY FOR THE RL01 DRIVES UNDER TEST, NOT HAVING ERROR CONDITIONS PRESENT. THE DATA WRITTEN IS SELECTED RANDOMLY FROM A SET OF FIXED DATA PATTERNS. WHEN A DRIVE HAS AN UNRECOVERABLE ERROR OTHER THAN A DATA TRANSFER OR SEEK INCOMPLETE, OR HAS EXCEEDED AN ERROR THRESHOLD, THE DRIVE WILL BE AUTOMATICALLY DEASSIGNED UNLESS SWR10=1.

THE PROGRAM HAS TWO MODES OF OPERATION:

ACCEPT MODE

MANUAL INTERVENTION MODE

THE ACCEPT MODE OF OPERATION WILL USE ALL ERROR LIMITS AND OPERATION LIMITS AS SPECIFIED BY DISK ENGINEERING. ALL OPERATIONS AND OPERATION PARAMETERS WILL BE DETERMINED RANDOMLY. THE MANUAL INTERVENTION MODE IS AVAILABLE AS A HARDWARE DEBUGGING AID TO ALLOW THE OPERATOR TO SELECT DATA PATTERNS, TRANSFER LENGTHS, AND DISK ADDRESSING.

## 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: RL8A/RL02 OR RL278/RL02 DISK CONTROLLER

SPECIAL: 1 TO 4 RL02 DISK DRIVES

NOTE: AN OPTIONAL POWER FAIL MODULE (M848) OR EQUIVALENT MAY BE USED WITH AUTO-RESTART FROM LOCATION 0000 FIELD 0.

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 RL278/RL02 DISKLESS DIAGNOSTIC
- C. RL8A/RL02 RL278/RL02 DRIVE TESTS (PART 1 AND PART 2)

3.0 RESTRICTIONS:  
-----

- A. AN OPERATOR'S CONSOLE TERMINAL MUST BE PRESENT AND ON-LINE (TELETYPE OR EQUIVALENT)
- B. THE DISK PACKS 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 PACKS USED FOR RUNNING THIS PROGRAM HAVE NO BAD HEADERS, OR HEADER CRC ERRORS WILL 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 THE PACK MUST BE CHANGED ON THE APPROPRIATE DRIVE (SEE SECTION 5.2.7 FOR FURTHER INFORMATION).



4.1 **LOADING THE PROGRAM:**  
-----

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

4.2 **PROGRAM INITIALIZATION:**  
-----

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).
- B. 8K OF READ/WRITE MEMORY - THE MINIMUM REQUIREMENT
- C. CONSOLE PACKAGE ACTIVE
- D. DEVICE CODES: 60 AND 61

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.

4.2.1 **STARTING ADDRESS:**  
-----

THE ONLY STARTING ADDRESS IS LOCATION 0200 IN FIELD 0

4.2.2 **APT INITIALIZATION:**  
-----

A. PSEUDO SWITCH REGISTER -

SWR7=1 OVERLAPPING SEEKS ONLY ( NO DATA TRANSFERS )

ALL OTHER SWITCHES = 0

BIT0 = 0           USE PSEUDO SWITCH REGISTER

BITS 1:6 = 0

BITS               MEMORY SIZE  
-----7 8 9 10 11  
-----

0 0 1 1 1	(7) = 8K (MINIMUM)	
0 1 0 1 1	(13) = 12K	
0 1 1 1 1	(17) = 16K	
1 0 0 1 1	(23) = 20K	
1 0 1 1 1	(27) = 24K	
1 1 0 1 1	(33) = 28K	
1 1 1 1 1	(37) = 32K	(NOT ALLOWED, FIELD 7 RESERVED FOR APT PROM)

THE PROGRAM WILL AUTO-SIZE MEMORY UPON STARTUP AND WILL USE ALL AVAILABLE MEMORY SPECIFIED IN HCW1, UNLESS A FIELD IS NOT FOUND. IF A FIELD IS NOT FOUND, THE PROGRAM WILL USE THE FIELDS UPTO BUT NOT INCLUDING THE FIRST FIELD NOT FOUND.

## C. HARDWARE CONFIGURATION WORD #2           (LOC 0022)

BIT0 = 1           (ON APT)

BITS 0:3 = 0

BITS               DEFINITION  
-----4 5  
----

0 0	BITS 6:11 = HIGHEST DRIVE TO TEST
0 1	BITS 6:11 = SINGLE DRIVE TO TEST
BITS	DRIVE TO TEST

6 7 8 9 10 11  
-----

0 0 0 0 0 0	(00) DRIVE 0
0 0 0 0 0 1	(01) DRIVE 1
0 0 0 0 1 0	(02) DRIVE 2
0 0 0 0 1 1	(03) DRIVE 3

## EXAMPLES:

- 1)       HCW2 = 4003       ON APT, TEST 4 DRIVES, (0:3)
- 2)       HCW2 = 4001       ON APT, TEST 2 DRIVES, (0:1)
- 3)       HCW2 = 4103       ON APT, TEST ONLY DRIVE 3

D. HARDWARE CONFIGURATION WORD #3           (LOC 0023)  
NOT USED

IN ACCEPT MODE, THE PROGRAM WILL RUN IN A RANDOM EXERCISING MODE, USING DISK ENGINEERING SPECIFICATIONS FOR ERROR AND OPERATION LIMITS. ALL NON-BAD SECTORS WILL EVENTUALLY BE ACCESSED; READ, WRITE, AND READ HEADER. ALL TRACKS WILL EVENTUALLY BE ACCESSED BY SEEKS FROM EITHER DIRECTION. EACH TIME 10 BILLION BITS HAVE BEEN TRANSFERRED, AN END OF DATA PASS MESSAGE WILL BE PRINTED ( THIS WILL TAKE BETWEEN 24 AND 48 HOURS FOR ONE DRIVE ALONE WITH 32K OF MEMORY). EACH TIME 1000 SEEKS HAVE BEEN COMPLETED, AN END OF SEEKS PASS MESSAGE WILL BE PRINTED. WHEN A GIVEN DRIVE HAS COMPLETED BOTH 1000 SEEKS AND 10 BILLION DATA BITS TRANSFERRED, A DRIVE ACCEPTED END OF PASS MESSAGE WILL BE PRINTED, AND THE DRIVE WILL BE DROPPED FROM TESTING UNLESS SWR10=1.

#### 4.3.1 OPERATOR PROMPTS/RESPONSE:

-----

THE FOLLOWING IS THE DIALOGUE THE PROGRAM WILL CARRY ON WITH THE OPERATOR FOR ACCEPT MODE OF OPERATION:

TITLE: "AJRLKB RL8A/RL02 PERFORMANCE EXERCISER"

IF THE PROGRAM HAS BEEN RESTARTED (NOT RELOADED) THEN Q1 WILL BE ASKED, ELSE GO TO Q2.

\*\*\*\*\*

\* Q1: \* "CLR DRV STAT TBLS (Y/N)?"

\*\*\*\*\*

RESPONSE: "Y<CR>" WILL CLEAR ALL DRIVE STATISTICS FROM TABLES

RESPONSE: "N<CR>" WILL LEAVE DRIVE STATISTICS LOG INTACT

\*\*\*\*\*

\* Q2: \* "NUM FLDS EXTD R/W MEM (1-7)?"

\*\*\*\*\*

THE OPERATOR RESPONDS WITH ONE OCTAL DIGIT FOLLOWED BY A CARRIAGE RETURN AS FOLLOWS:

RESPONSE:	READ/WRITE MEMORY SIZE:
-----	-----

1	8K
2	12K
3	16K
4	20K
5	24K
6	28K
7	32K

AS MUCH OR AS LITTLE OF THIS EXTENDED R/W MEMORY MAY BE USED AS IS DESIRED (MINIMUM = 8K). THE PROGRAM WILL AUTOSIZE TO VERIFY THAT ALL FIELDS ENTERED EXIST. IF A FIELD IS NOT FOUND, THE FOLLOWING ERROR MESSAGE WILL RESULT:

MESSAGE: "NO FLD X"

WHERE "X" IS THE FIRST FIELD NOT FOUND. THE PROGRAM WILL THEN REASK Q2. IF Q2 IS ANSWERED CORRECTLY, Q3-Q6 WILL BE ASKED NEXT.

\*\*\*\*\*  
 \* Q3: \*  
 \*\*\*\*\*

"TEST DRIVE 0 (Y/N)?"

RESPONSE: "Y<CR>"

WILL ACTIVATE DRIVE 0 FOR TESTING

RESPONSE: "N<CR>"

WILL DEACTIVATE DRIVE 0 FROM TESTING

\*\*\*\*\*  
 \* Q4: \*  
 \*\*\*\*\*

"TEST DRIVE 1 (Y/N)?"

RESPONSE: "Y<CR>"

WILL ACTIVATE DRIVE 1 FOR TESTING

RESPONSE: "N<CR>"

WILL DEACTIVATE DRIVE 1 FROM TESTING

\*\*\*\*\*  
 \* Q5: \*  
 \*\*\*\*\*

"TEST DRIVE 2 (Y/N)?"

RESPONSE: "Y<CR>"

WILL ACTIVATE DRIVE 2 FOR TESTING

RESPONSE: "N<CR>"

WILL DEACTIVATE DRIVE 2 FROM TESTING

\*\*\*\*\*  
 \* Q6: \*  
 \*\*\*\*\*

"TEST DRIVE 3 (Y/N)?"

RESPONSE: "Y<CR>"

WILL ACTIVATE DRIVE 3 FOR TESTING

RESPONSE: "N<CR>"

WILL DEACTIVATE DRIVE 3 FROM TESTING

\*\*\*\*\*  
 \* Q7: \*  
 \*\*\*\*\*

"DC (60,61) (Y/N)?"

RESPONSE: "Y<CR>"

WILL USE DEVICE CODE 60 AND 61 (DEFAULT AND STANDARD)

RESPONSE: "N<CR>"

WILL USE DEVICE CODE 62 AND 63 (REQUIRES JUMPERING ON M8433)

\*\*\*\*\*  
 \* Q8: \*  
 \*\*\*\*\*

"SR=0000 \_"

RESPONSE:

1 TO 4 OCTAL DIGITS FOLLOWED BY CARRIAGE RETURN WILL MODIFY THE CONTENTS OF THE PSEUDO SWITCH REGISTER. TYPING JUST A LINE-FEED WILL RESTART THE PROGRAM (SEE SECTION 4.6 CONSOLE PACKAGE). TYPING JUST A CARRIAGE RETURN WILL CONTINUE THE PROGRAM WITHOUT MODIFYING THE PSEUDO SWITCH REGISTER.

\*\*\*\*\*

RESPONSE: "Y<CR>" PROGRAM WILL ASK Q10  
 RESPONSE: "N<CR>" PROGRAM WILL ENTER THE MANUAL INTERVENTION MODE OF  
 OPERATION AND CONTINUE WITH Q11 AS DESCRIBED IN SECTION 4.4.1

\*\*\*\*\*

\* Q10: \* "ARE YOU SURE <Y/N>?"  
 \*\*\*\*\*

RESPONSE: "Y<CR>" PROGRAM WILL NOW RUN IN ACCEPT MODE AS DESCRIBED IN  
 SECTION 4.3  
 RESPONSE: "N<CR>" PROGRAM WILL BE RESTARTED AND Q1 THROUGH Q10 WILL BE  
 REASKED

#### 4.3.2 PARAMETERS:

-----

THE FOLLOWING ARE THE ACCEPT MODE PARAMETERS USED BY THE PROGRAM:

- A. DATA DUMP ON DATA CRC ERROR (UP TO 254 DATA WORDS)
- B. DROP DRIVE ON ERROR LIMITS REACHED
- C. DROP DRIVE ON OPERATION LIMITS REACHED
- D. USE BOTH DISK SURFACES (ALL DRIVES)
- E. MINIMUM CYLINDER = 0
- F. MAXIMUM CYLINDER = 377(OCTAL) (NO SURFACE 1 BAD SECTOR FILE)
- G. MINIMUM SECTOR = 0
- H. MAXIMUM SECTOR = 47(OCTAL)
- I. READ/WRITE RATIO = 3:2 ( RANDOMLY SELECTED DATA PATTERNS FOR WRITES )
- J. RANDOM SEEKS
- K. RANDOM WORD COUNT (TRANSFER LENGTH)
- L. RANDOM BUFFER FIELD SELECTION
- M. RETRY LIMIT = 1
- N. SEEK TO TRACK RETRY LIMIT = 1
- O. SEEK ERROR LIMIT = 1
- P. DATA TRANSFER LIMIT = 10 BILLION BITS
- Q. SEEK LIMIT = 1000
- R. HARD ERROR LIMIT = 3
- S. SOFT ERROR LIMIT = 10
- T. DRIVE ERROR LIMIT = 3
- U. NUMBER OF DATA WORD ERRORS = 254(DECIMAL)

THE PACKS ON ALL DRIVES UNDER TEST WILL BE INITIALLY COMPLETELY WRITTEN WITH DATA PATTERN NUMBER 1 (SEE SECTION 4.3.4) AND READ VERIFIED FOR CORRECTNESS. THIS IS THE WORST CASE DATA PATTERN. THE TIME INVOLVED FOR THIS OPERATION IS AS FOLLOWS:

INITIAL WRITE: 1.5 MINUTES PER DRIVE

INITIAL READ/VERIFY: 6 MINUTES PER DRIVE

- A. BAD SECTORS (FROM THE BAD SECTOR FILES) WILL NOT BE WRITTEN OR READ
- B. EVEN NUMBERED TRACKS (0,2,4 ... 376) SURFACE 0 ARE 8 BIT MODE  
ODD NUMBERED TRACKS (1,3,5 ... 377) SURFACE 0 ARE 12 BIT MODE  
  
EVEN NUMBERED TRACKS SURFACE 1 ARE 12 BIT MODE  
ODD NUMBERED TRACKS SURFACE 1 ARE 8 BIT MODE
- C. THE FIRST WORD OF EACH SECTOR CONTAINS THE NUMBER OF DATA WORDS (NON-ZERO FILL) IN THE SECTOR. THE SECOND WORD OF EACH SECTOR CONTAINS THE DATA PATTERN NUMBER USED FOR THAT SECTOR. THIS ALLOWS EXERCISING TO BEGIN FROM A KNOWN STATE AND PROVIDES A QUICK VERIFY THAT ALL SECTORS CAN BE READ AND WRITTEN CORRECTLY USING THE WORST CASE DATA PATTERN. IT ALSO PERMITS RANDOM READS, SINCE THE SECTOR DATA IS DESCRIBED BY THE FIRST 2 WORDS OF THE SECTOR, AND THE DATA PATTERNS ARE KNOWN TO THE PROGRAM.
- D. AFTER A PACK HAS BEEN COMPLETELY WRITTEN, THE FOLLOWING MESSAGE WILL BE PRINTED:  
  
"DRIVE X PACK WRITTEN"  
  
WHERE "X" IS THE DRIVE NUMBER WHOSE PACK WAS SUCCESSFULLY WRITTEN.
- E. AFTER A PACK HAS BEEN READ AND VERIFIED THE FOLLOWING MESSAGE WILL BE PRINTED:  
  
"DRIVE X PACK READ OK"
- F. ALL ERRORS DURING THIS PHASE OF TESTING WILL BE HANDLED LIKE ANY OTHER ERRORS, SEE SECTION 5 FOR ERROR DESCRIPTIONS.
- G. THE MQ REGISTER WILL CONTAIN THE CYLINDER ADDRESS IN BITS 4:11 AND THE HEAD SELECT IN BIT 1 AS THE PACKS ARE WRITTEN TO PROVIDE A MEANS OF MONITORING THE PROGRESS OF THE INITIAL PACK WRITES AND READ/VERIFIES (IF A HARDWARE FRONT PANEL IS AVAILABLE).

\*\*\*\*\*

PATTERN 0	PATTERN 1	PATTERN 2	PATTERN 3
-----	-----	-----	-----
0052	0333	0242	0113
0252	0155	0105	0113
0125	0266	0321	0245
0125	0333	0042	0245
0125	0155	0150	0322
0125	0266	0221	0322
0252	0333	0264	0151
0252	0155	0110	0151
0252	0266	0132	0264
0252	0333	0044	0264
0252	0155	0055	0132
0252	0266	0022	0132
0125	0333	0026	0055
0125	0155	0211	0055
0125	0266	0213	0226
0125	0333	0104	0226
0252	0155	0105	0113
0252	0266	0242	0113
0252	0333	0042	0245
0252	0155	0321	0245
0125	0266	0221	0322
0125	0333	0150	0322
0252	0155	0110	0151
0252	0266	0264	0151
0125	0333	0044	0264
0125	0155	0132	0264
0252	0266	0022	0132
0252	0333	0055	0132
0125	0155	0211	0055
0125	0266	0026	0055
0252	0333	0104	0226
0252	0155	0213	0226

PATTERN 0 -----	PATTERN 1 -----	PATTERN 2 -----	PATTERN 3 -----
1252	6666	5044	2264
5125	6666	2721	5645
2525	6666	1046	5135
2525	6666	4221	1322
5252	6666	5504	3226
5252	6666	4132	4664
5252	6666	1102	5505
5252	6666	6422	5132
2525	6666	0550	1322
2525	6666	4613	6626
2532	6666	2104	4544
5252	6666	2642	5513
5252	6666	1055	5132
5125	6666	0621	2722
2532	6666	3204	6446
5252	6666	4264	4551
2525	6666	1105	5513
2652	6666	5022	2132
5245	6666	1330	2642
2525	6666	4426	6455
5252	6666	2110	4551
5252	6666	5422	3004

#### 4.3.5 END OF PASS (SEEKS): -----

EACH TIME A DRIVE COMPLETES 1000 SEEKS, THE FOLLOWING MESSAGE WILL BE PRINTED:

"END PASS DRIVE X SEEK"

WHERE "X" IS THE DRIVE NUMBER COMPLETING 1000 SEEKS. THE DRIVE WILL NOT BE DROPPED AND WILL NOT HAVE BEEN ACCEPTED (PER DISK ENGINEERING SPEC.) UNTIL 10 BILLION BITS OF DATA HAVE BEEN TRANSFERRED. THEREFORE, DURING THE MANY HOURS IT WILL TAKE TO TRANSFER 10 BILLION BITS OF DATA, 1000 SEEKS WILL BE COMPLETED MANY TIMES.

#### 4.3.6 END OF PASS (DATA): -----

EACH TIME A DRIVE COMPLETES THE TRANSFER OF 10 BILLION BITS, THE FOLLOWING MESSAGE WILL BE PRINTED:

"END PASS DRIVE X DATA"

NOTE: THIS WILL TAKE APPROXIMATELY 48 HOURS WITH 1 DRIVE AND 16K OF MEMORY, SINCE 12 BIT DATA TRANSFERS ARE LIMITED TO 1 SECTOR.



AFTER A DRIVE HAS COMPLETED THE TRANSFER OF 10 BILLION BITS AND 1000  
SEKS THE FOLLOWING MESSAGE WILL BE PRINTED:

"END PASS DRIVE X ACPTD"

WHERE "X" IS THE NUMBER OF THE DRIVE JUST ACCEPTED.

#### 4.4 MANUAL INTERVENTION MODE: -----

IN MANUAL INTERVENTION MODE, THE PROGRAM WILL RUN IN A RANDOM OR  
FIXED EXERCISING MODE, USING OPERATOR SPECIFIED AND/OR DISK ENGINEERING  
SPECIFICATIONS FOR ERROR AND OPERATION LIMITS.

##### 4.4.1 OPERATOR PROMPTS/RESPONSE: -----

ALL QUESTIONS IN SECTION 4.3.1 WILL BE ASKED UP TO AND INCLUDING Q9. IF THE  
WILL CONTINUE AS FOLLOWS:

\*\*\*\*\*  
\* Q11: \*  
\*\*\*\*\*

"RETRY LIMIT (1-4095)?"

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL  
BE THE ERROR RETRY LIMIT FOR ERRORS OF ALL TYPES EXCEPT FATAL ERRORS.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT LIMIT = 1

\*\*\*\*\*  
\* Q12: \*  
\*\*\*\*\*

"SEEK RETRY LIMIT (1-4095)?"

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE  
THE SEEK ERROR RETRY LIMIT (THE ANSWER TO Q11 TAKES PRECEDENCE OVER  
THIS LIMIT IF THE ANSWER TO Q11 IS LESS THAN THE ANSWER TO Q12).

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT LIMIT = 1.

\*\*\*\*\*  
\* Q13: \*  
\*\*\*\*\*

"DATA DUMP ON DCRC (Y/N)?"

RESPONSE: "Y<CR>" THIS WILL ENABLE THE PROGRAM TO PRINT DATA WORD ERRORS ON THE TERMINAL.

RESPONSE: "N<CR>" THIS WILL INHIBIT DATA WORD ERRORS FROM BEING PRINTED.

\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS IN THE RANGE 1 TO 4095 FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE SOFT ERROR LIMIT UPON WHICH TO DROP A DRIVE. INTERMITTENT ERRORS WHICH DO NOT PERMIT RETRIES, WILL INCREASE THE SOFT ERROR COUNT, BUT WILL NOT CAUSE THE DRIVE TO BE DROPPED IF THE SOFT ERROR LIMIT IS EXCEEDED, SINCE THE OPERATION IS ABORTED. THEREFORE IF THE SOFT ERROR LIMIT IS EXCEEDED IN THE ERROR MESSAGE SOME INTERMITTENT ERROR IS OCCURRING, AND THE OTHER RL8A/RL02 DIAGNOSTICS MUST BE RUN TO DETERMINE THE EXACT NATURE OF THE ERROR.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT = 12.

\*\*\*\*\*  
\* Q15: \*  
\*\*\*\*\*

"DROP DRV ON OPR LIMIT REACHED (Y/N)?"

RESPONSE: "Y<CR>" THIS WILL ALLOW A DRIVE TO BE DROPPED FROM TESTING WHEN A SPECIFIC OPERATION LIMIT HAS BEEN REACHED. Q16 IS NEXT.

RESPONSE: "N<CR>" THIS WILL INHIBIT A DRIVE FROM BEING DROPPED UNTIL ACCEPTED OR AN ERROR LIMIT HAS BEEN EXCEEDED. Q18 IS NEXT.

\*\*\*\*\*  
\* Q16: \*  
\*\*\*\*\*

"XFER LIMIT (1-4095)?"

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE NUMBER OF TIMES (1 TO 4095) THAT 10 BILLION BITS WILL BE TRANSFERRED BEFORE THE DRIVE WILL BE DROPPED.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT LIMIT = 1 TIMES 10 BILLION BITS.

\*\*\*\*\*  
\* Q17: \*  
\*\*\*\*\*

"SEEK OPR LIMIT (1-4095)?"

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE NUMBER OF TIMES (1 TO 4095) 1000 SEEKS WILL BE PERFORMED BEFORE THE DRIVE MAY BE DROPPED AUTOMATICALLY FROM TESTING.

\*\*\*\*\*  
\* Q18: \*  
\*\*\*\*\*

"MODIFY SEEK, R/W PARAMS (Y/N)?"

RESPONSE: "Y<CR>" THIS WILL ALLOW CHANGING OF THE SEEK AND READ/WRITE PARAMETERS. Q19 WILL BE NEXT.

RESPONSE: "N<CR>" THIS WILL BYPASS CHANGING OF THE SEEK AND READ/WRITE PARAMETERS. Q31 WILL BE NEXT.

\*\*\*\*\*

RESPONSE: "Y<CR>" THIS WILL ALLOW THE PROGRAM TO USE A RANDOMLY GENERATED TRANSFER LENGTH (WORD COUNT). Q21 WILL BE NEXT.

RESPONSE: "N<CR>" THIS WILL ALLOW THE OPERATOR TO SPECIFY A FIXED TRANSFER LENGTH (WORD COUNT). Q20 WILL BE NEXT.

\*\*\*\*\*

\* Q20: \* "WC (3-4096)?" IF MORE THAN 8K OF MEMORY SPECIFIED IN Q2 SECTION 4.3.1

\*\*\*\*\*

OR

"WC (3-1024)?" IF 8K OF MEMORY (LARGEST BUFFER IS 1K IN FIELD 1)

RESPONSE: 1 TO 4 DECIMAL DIGITS, WITHIN THE RANGE SPECIFIED BY THE PROMPT OF Q20, FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE FIXED TRANSFER LENGTH (WORD COUNT) USED BY THE PROGRAM. IF THE LIMITS ARE EXCEEDED, THEN Q20 WILL BE REASKED.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT FIXED WORD COUNT = 4096 OR 1024 DEPENDING ON MEMORY SIZE.

\*\*\*\*\*

\* Q21: \* "RD ONLY (Y/N)?"

\*\*\*\*\*

RESPONSE: "Y<CR>" THIS WILL PREVENT THE PROGRAM FROM DOING ANY WRITE OPERATIONS AFTER THE INITIAL WRITE OF THE PACKS. Q24 WILL BE NEXT.

RESPONSE: "N<CR>" THIS WILL ALLOW THE PROGRAM TO PERFORM WRITE OPERATIONS TO ALL DRIVES. Q22 WILL BE NEXT.

\*\*\*\*\*

\* Q22: \* "CONSTANT DATA (Y/N)?"

\*\*\*\*\*

RESPONSE: "Y<CR>" THIS WILL ALLOW THE OPERATOR TO SELECT ONE OF THE 4 DATA PATTERNS SHOWN IN SECTION 4.3.4 AS THE ONLY DATA PATTERN THE PROGRAM WILL USE, INCLUDING THE INITIAL WRITE OF THE PACKS. Q23 WILL BE NEXT.

RESPONSE: "N<CR>" THIS WILL ALLOW THE PROGRAM TO RANDOMLY SELECT A DATA PATTERN FROM THOSE SHOWN IN SECTION 4.3.4. Q24 WILL BE NEXT.

\*\*\*\*\*

\* Q23: \* "PAT (0-3)?"

\*\*\*\*\*

RESPONSE: 1 OCTAL DIGIT FOLLOWED BY A CARRIAGE RETURN. THIS WILL SELECT DATA PATTERN 0,1,2, OR 3 FOR ALL WRITE OPERATIONS (SEE SECTION 4.3.4)

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT DATA PATTERN FOR ALL WRITE OPERATIONS (PATTERN #1 WORST CASE).

\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS, IN THE RANGE 0 THROUGH 254, FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE LIMIT OF DATA WORD ERRORS PRINTED ON THE OPERATOR'S TERMINAL.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT LIMIT = 254.

\*\*\*\*\*

\* Q25: \* "SNGL SURFACE XFER (Y/N)?"  
\*\*\*\*\*

RESPONSE: "Y<CR>" THIS WILL RESTRICT THE DATA TRANSFERS TO AND FROM A SINGLE SURFACE OF ALL DRIVE'S PACKS. Q26 WILL BE NEXT.

RESPONSE: "N<CR>" THIS WILL ALLOW DATA TRANSFERS TO AND FROM BOTH SURFACES OF ALL DRIVE'S PACKS. Q27 WILL BE NEXT.

\*\*\*\*\*

\* Q26: \* "SURF (0 OR 1)?"  
\*\*\*\*\*

RESPONSE: "0" OR "1" FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE SURFACE USED FOR ALL DATA TRANSFERS ON ALL DRIVES.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL USE THE DEFAULT SURFACE WHICH IS SURFACE 0.

\*\*\*\*\*

\* Q27: \* "MAX CYL (0-255)?"  
\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE MAXIMUM CYLINDER USED FOR SEEKS AND DATA TRANSFERS.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT MAXIMUM CYLINDER = 255 (DECIMAL).

\*\*\*\*\*

\* Q28: \* "MIN CYL (0-255)?"  
\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE MINIMUM CYLINDER USED FOR SEEKS AND DATA TRANSFERS.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT MINIMUM CYLINDER = 0.

NOTE: THE MINIMUM CYLINDER MUST BE LESS THAN OR EQUAL TO THE MAXIMUM CYLINDER OF Q27 OR Q28 WILL BE REASKED.

\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE MAXIMUM SECTOR USED FOR START OF DATA TRANSFERS.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT MAXIMUM SECTOR = 39.

\*\*\*\*\*

\* Q30: \* "MIN SECTOR (0-39)?"

\*\*\*\*\*

RESPONSE: 1 TO 4 DECIMAL DIGITS FOLLOWED BY A CARRIAGE RETURN. THIS WILL BE THE MINIMUM SECTOR USED FOR START OF DATA TRANSFERS.

RESPONSE: "<CR>" TYPING JUST A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO USE THE DEFAULT MINIMUM SECTOR = 0.

NOTE: THE MINIMUM SECTOR MUST BE LESS THAN OR EQUAL TO THE MAXIMUM SECTOR OF Q29 OR Q30 WILL BE REASKED.

\*\*\*\*\*

\* Q31: \* "ARE YOU SURE (Y/N)?"

\*\*\*\*\*

RESPONSE: "Y<CR>" THIS WILL START THE PROGRAM, USING ALL PARAMETERS AS MODIFIED, IN MANUAL INTERVENTION MODE. ALL END OF PASS PARAMETERS REMAIN UNCHANGED (SEE SECTIONS 4.3.5-4.3.7)

RESPONSE: "N<CR>" THIS WILL CAUSE THE PROGRAM TO BE RESTARTED AND ALL QUESTIONS WILL BE REASKED.

- A. RETRY LIMITS:** THE RETRY LIMIT SET IN Q11 TAKES PRIORITY OVER ALL OTHER RETRY AND ERROR LIMITS OF SECTION 4.4.1
- B. SEEK OPERATION LIMIT:** THE SEEK OPERATION LIMIT OF Q17 TAKES PRIORITY OVER THE DATA TRANSFER LIMIT OF Q16 (SECTION 4.3.1) SINCE 1000 SEEKS ARE COMPLETED MANY TIMES BEFORE 10 BILLION BITS OF DATA ARE TRANSFERRED.
- C. FIXED WORD COUNT:** A FIXED WORD COUNT GREATER THAN 170 (DECIMAL) WILL FORCE THE PROGRAM TO PERFORM ONLY 8 BIT MODE TRANSFERS, SINCE THE MAXIMUM TRANSFER IN 12 BIT MODE IS ONE SECTOR (170 WORDS). WHENEVER THE HEADS ARE SELECTED AND POSITIONED ON A 12 BIT TRACK, A SEEK WILL BE PERFORMED. IF THE MAXIMUM AND MINIMUM CYLINDER LIMITS AND A SINGLE SURFACE HAVE BEEN SELECTED SUCH THAT ONLY A 12 BIT TRACK IS SELECTED, NO DATA TRANSFERS WILL OCCUR AND ONLY SEEK TO CURRENT TRACK WILL OCCUR.
- D. MAXIMUM/MINIMUM SECTOR:** IF THE MINIMUM STARTING SECTOR OF Q30 IS GREATER THAN SECTOR 23, THE ALLOWABLE WORD COUNT WILL DECREASE AS THE MINIMUM STARTING SECTOR INCREASES. THIS MAY CONFLICT WITH THE FIXED WORD COUNT OF Q20. IN THIS EVENT NO DATA TRANSFERS WILL OCCUR (SEEKS ONLY).

**4.4.3 INITIAL WRITE/READ OF PACKS:**  
-----

THE INITIAL WRITE AND READ VERIFY OF THE PACKS OF ALL DRIVES UNDER TEST WILL OCCUR, AS DESCRIBED IN SECTION 4.3.3, WITH THE EXCEPTION THAT IF A FIXED DATA PATTERN WAS SELECTED IN Q23 OF SECTION 4.4.1, THEN ALL PACKS WILL BE WRITTEN WITH THE SELECTED CONSTANT DATA PATTERN.

**4.4.4 END OF PASS:**  
-----

END OF PASS MESSAGES AND PARAMETERS ARE THE SAME AS THOSE DESCRIBED IN SECTION 4.3.5 THROUGH 4.3.7

**4.5 REQUIREMENTS:**  
-----

THE RUNNING OF THIS PROGRAM REQUIRES THAT ALL DRIVES TO BE TESTED WILL CONTAIN A PACK WITH 16 OR LESS BAD SECTORS; THAT ALL DRIVES ARE WRITE ENABLED; AND THAT ALL PRELIMINARY DIAGNOSTICS (SECTION 2.3) HAVE BEEN RUN WITH NO ERRORS.

IF ENTERED WITH AC=0000 THE SWITCH REGISTER  
MODIFICATION ROUTINE IS ENTERED AUTOMATICALLY.  
IF ENTERED WITH AC NOT EQUAL TO 0000, THE  
KEYBOARD INPUT DECODER IS ENTERED AND IT IS ASSUMED  
THAT THE AC CONTAINS THE ASCII CODE TO BE  
CHECKED FOR A VALID CONTROL CHARACTER.

# 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:

OPERATOR TYPES:

-----

RESULT:

-----

ANOTHER CONTROL G

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

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

**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 SHOULD CONTINUE TO RUN. IF THE OPERATOR TYPES CONTROL S WHILE A MESSAGE IS IN PROGRESS, THE PRINTOUT SHOULD BE INTERRUPTED AND THE PROGRAM SHOULD WAIT FOR A CONTROL Q OR CONTROL C.

**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 MONITOR.



THIS IS A NON-STANDARD CONTROL CHARACTER USED BY THIS PROGRAM AS AN INSTRUCTION TO PRINT THE STATUS OF ALL DRIVES ONCE FOR EACH TYPING OF CONTROL R. THERE IS NO AUTOMATIC PRINTOUT OF DRIVE STATUS EXCEPT IN THE CASE OF AN ERROR.

5.0 ERRORS:  
-----

5.1 ON APT:  
-----

ALL ERRORS WHICH REQUIRE DROPPING A DRIVE, EXCEEDING AN ERROR OR RETRY LIMIT, OR WHICH ARE FATAL TO THE RL02 SUBSYSTEM, WILL BE REPORTED TO APT (IF RUNNING ON APT) WITH THE ERROR PC CONTAINED IN THE AC.

5.2 NOT ON APT:  
-----

ALL ERRORS WILL BE REPORTED ON THE USER'S CONSOLE TERMINAL UNLESS SWR5=1

5.2.1 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.

"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 4.6)

NOTE: THIS MESSAGE CANNOT BE INHIBITED BY SETTING SWR5=1

#### 5.2.2 SOFT ERRORS: -----

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

#### 5.2.3 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  
 (THE DRIVE WILL BE DROPPED UNLESS SWR10=1)

C. DRIVE ERRORS WHICH CANNOT BE RESET

#### 5.2.4 ERROR LOGGING: -----

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 5.2.6 FOR ERROR REPORTING DESCRIPTION)

**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

**B. READ AND WRITE RETRIES CONSIST OF THE FOLLOWING OPERATIONS:**

- 1) ISSUE READ OR WRITE
- 2) WAIT FOR DONE
- 3) CHECK FOR ERRORS
- 4) CHECK DATA IF READ
- 5) IF SUCCESSFUL LOG SOFT ERROR
- 6) IF RETRY EXCEEDED LOG HARD ERROR

**5.2.6 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.**

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

NOTE: SEE BEGINNING OF PROGRAM LISTING FOR BIT DEFINITIONS OF THE CONTROLLER  
REGISTERS AND DRIVE STATUS WORDS.

SEQ 24

"ERROR PC: XXXX"  
"DN:000X"  
"ER: X00X CA: XXXX CB: XXXX SA: XXXX IWC: 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  
IWC: 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  
SEKS: 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 RD FACT BAD SECT FILE DRIVE X"

OR

"CAN'T RD FLD BAD SECT FILE DRIVE X"

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

THE DRIVE WILL NOW BE DEACTIVATED WITHOUT ANY FURTHER MESSAGES. THE PACK SHOULD BE CHANGED ON DRIVE "X", AND THE DRIVE MAY NOW BE ADDED (SEE SECTION 8.1) OR THE PROGRAM MAY BE RESTARTED.

6.0 SWITCH REGISTER SETTINGS:

-----

6.1 NORMAL OPERATION SWITCHES:

-----

ALL SWITCHES = 0

6.2 ERROR RELATED SWITCHES:

-----

SR0 = 1 INHIBIT ERROR HALT

SR5 = 1 INHIBIT ERROR PRINTOUTS

SR10 = 1 INHIBIT DRIVE DROP ON ERROR OR PASS COMPLETE.

6.3 OTHER SWITCHES:

-----

SR3 = 1 HALT AT END OF PASS

SR7 = 1 PERFORM ONLY OVERLAPPING SEEKS (NO DATA TRANSFERS)

SR8 = 1 ADD OR DROP DRIVES

SR9 = 1 INHIBIT SOFTWARE DATA CHECKING

(DATA CRC LOGIC WILL BE RELIED UPON FOR DATA CHECK)

8.0 ADDITIONAL INFORMATION:  
-----8.1 ADDING OR DROPPING DRIVES:  
-----

ONE OR MORE DRIVES MAY BE ADDED TO OR DROPPED FROM TESTING BY

\*\*\*\*\*  
\* Q32: \*  
\*\*\*\*\*

"ADD DRIVE (Y/N)?"

RESPONSE: "Y<CR>" USER DESIRES TO ADD DRIVE(S). GO TO Q33.

RESPONSE: "N<CR>" USER DESIRES TO DROP DRIVE(S). GO TO Q37.

\*\*\*\*\*  
\* Q33: \*  
\*\*\*\*\*

"DRIVE?"

USER IS ASKED FOR WHICH DRIVE  
NUMBER TO ADD. ANY VALID DRIVE  
NUMBER (0,1,2, OR 3) MAY BE  
ADDED, EVEN IF ALREADY ACTIVE OR  
DROPPED DUE TO ERRORS.

RESPONSE: ONE OCTAL DIGIT IN THE RANGE 0 TO 3 FOLLOWED BY A  
CARRIAGE RETURN. NO DEFAULT IS PROVIDED. Q33 WILL  
BE REASKED IF JUST A CARRIAGE RETURN IS TYPED.

\*\*\*\*\*  
\* Q34: \*  
\*\*\*\*\*

"CLR DRV STAT TBLS (Y/N)?"

RESPONSE: "Y<CR>" THE DRIVE STATE TABLE FOR THE DRIVE BEING ADDED  
ADDED WILL BE CLEARED. ALL ERROR TALLIES  
AND OPERATION COUNTS WILL BE CLEARED.

RESPONSE: "N<CR>" THE DRIVE STATE TABLE FOR THE DRIVE BEING  
ADDED WILL NOT BE CLEARED. ALL ERROR TALLIES  
AND OPERATION COUNTS WILL REMAIN IN THE  
TABLE.

\*\*\*\*\*  
\* Q35: \*  
\*\*\*\*\*

"GET BAD SECT FILE (Y/N)?"

RESPONSE: "Y<CR>" THE BAD SECTOR FILE FOR THE SELECTED DRIVE  
WILL BE RETRIEVED AND PROCESSED.

RESPONSE: "N<CR>" THE BAD SECTOR FILE FOR THE SELECTED DRIVE  
WILL NOT BE RETRIEVED. A PACK WITH NO BAD  
SECTORS WILL BE ASSUMED BY THE PROGRAM. ALL  
SECTORS ON THE PACK WILL BE WRITTEN AND READ,  
(EXCEPT BAD SECTOR FILE)

\*\*\*\*\*

RESPONSE: "Y<CR>" THE ENTIRE PACK OF THE SELECTED DRIVE WILL  
BE WRITTEN AND READ VERIFIED (SETTING  
SWR9 = 1 WILL DECREASE THE TIME, SEE SECTION  
4.3.3) Q39 WILL BE NEXT.

RESPONSE: "N<CR>" THE PROGRAM WILL ASSUME THAT THE PACK HAS  
BEEN WRITTEN AND READ VERIFIED. Q39 WILL BE NEXT.

\*\*\*\*\*

\* Q37: \* "DROP DRV (Y/N)?"

\*\*\*\*\*

RESPONSE: "Y<CR>" THIS ALLOWS THE USER TO DISCONTINUE TESTING  
A SINGLE DRIVE. GO TO Q38.

RESPONSE: "N<CR>" NO DRIVE WILL BE DROPPED. GO TO Q39.

\*\*\*\*\*

\* Q38: \* "DRIVE?" USER IS ASKED FOR WHICH DRIVE  
NUMBER TO DROP

\*\*\*\*\*

RESPONSE: 1 OCTAL DIGIT IN THE RANGE 0 TO 3 FOLLOWED BY A  
CARRIAGE RETURN. NO DEFAULT IS PROVIDED. Q38 WILL  
BE REASKED IF JUST A CARRIAGE RETURN IS TYPED.

\*\*\*\*\*

\* Q39: \* "SR=XXXX \_" THIS ALLOWS THE USER TO MODIFY  
THE PSEUDO SWITCH REGISTER BEFORE  
EXERCISING THE DRIVES. (SEE SECTION 4.6 FOR RESPONSES.)

\*\*\*\*\*

## 8.2 HALTS:

-----  
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 INT REQ"

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

"UNK INT"

- 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.

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, THE DRIVE WILL BE DEACTIVATED FROM TESTING, UNLESS SWR10 = 1.

#### 8.4 STATUS MESSAGES:

-----

WHEN THE USER TYPES A CONTROL-R (SEE SECTION 4.6) THE STATUS OF EACH DRIVE WILL BE PRINTED AS FOLLOWS:

"DN: 000X"  
 "ER: X00X CA: XXXX CB: XXXX SA: XXXX IWC: 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"  
 "DRIVE ACTV"

ETC.

SEE SECTION 5.2.6 (B) FOR EXPLANATION OF THESE CODES. THE STATUS FOR ALL POSSIBLE DRIVES WILL BE PRINTED ONCE EACH TIME THAT CONTROL-R IS TYPED. THE "DRIVE ACTV" OR "DRV NOT ACTV" MESSAGE TELLS WHETHER THE DRIVE IS CURRENTLY BEING EXERCISED. ONLY THE DRIVE LAST USING THE CONTROLLER WILL HAVE THE CONTROLLER REGISTERS PRINTED WITH THE DRIVE'S STATUS.



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:

- A. IF ON APT, REPORT ERROR TO APT
- B. IF NOT ON APT, THE FOLLOWING MESSAGE WILL BE PRINTED:

"PWR FAIL PC: XXXX"

- C. IF ALL QUESTIONS HAVE NOT BEEN ANSWERED, (POWER FAILED DURING INITIAL PROGRAM/USER DIALOGUE) OR DURING THE INITIAL WRITE/READ OF THE PACKS, THE PROGRAM WILL HAVE TO BE RESTARTED BY THE USER, SINCE THE POWER FAIL ROUTINE MUST ASSUME THAT ALL PACKS HAVE BEEN WRITTEN.
- D. IF ALL QUESTIONS HAVE BEEN ANSWERED, THE PROGRAM RECOVERY ROUTINE FOR POWER FAIL WILL BE ENTERED AND WILL CONSIST OF THE FOLLOWING STEPS:
  - 1. DELAY 40 SECONDS FOR DRIVES TO SPIN UP AND HEADS TO LOAD
  - 2. RESET ALL DRIVES
  - 3. IF THE LAST DRIVE USING THE CONTROLLER WAS PERFORMING A WRITE OPERATION, THAT DRIVE'S PACK WILL BE COMPLETELY WRITTEN AND READ/VERIFIED AS DESCRIBED IN SECTION 4.4.3 WITHOUT OPERATOR INTERVENTION.
  - 4. GET STATUS AND CHECK FOR ERRORS (ALL DRIVES)
  - 5. SEEK ALL DRIVES TO TRACK 0
  - 6. RESUME EXERCISING THE DRIVES

9.0 PROGRAM LISTING: ATTACHED

/AJRLK-B            PERFORMANCE EXERCISER FOR THE RL0A-RL02  
/  
/COPYRIGHT (C) 1981 DIGITAL EQUIPMENT CORP. MAYNARD, MASS. 01754  
/

/ REVISION HISTORY  
/-----

/ ORIGINAL RELEASE - DAVE ORIN  
/

/ MODIFICATIONS BY        DATE  
/-----  
/ H. POULTER               FEBURARY 1979  
/

/ HP 001 : MODIFICATIONS FOR RL02  
/

/ HP 002 : FIX FLD 2 OS-8 BOOT AREA OVERWRITE BY PROGRAM  
/

      MOVED SNGLHD TO END OF A ZERO FILL BUFFER, WAS IN ALL ONES BUFFER

      ALLOW CYLINDER AND HEAD TO BE PLACED IN MQ FOR DISPLAY DURING  
      EXERCISER PART OF THIS DIAGNOSTIC, NOT INITIAL WRITE AND READ.

      REMOVE CODE TO DISPLAY HEAD AND CYLINDER IN MQ.  
      DONE IN POSSET ROUTINE.

      CHANGE THE NUMBER OF TRACKS TO READ        RL02 CONTAINS 1023  
  RL01 CONTAINS 777

      REMOVE CODE TO DISPLAY HEAD AND CYLINDER IN MQ.  
      DONE IN POSSET ROUTINE.

      ADD INSTRUCTION ISZ BADCNT TO PREVENT MISSING ERROR CONDITION.

      ADD A DELAY TO WAIT FOR DISK TO SPIN UP.

      ADD CODE TO DISPLAY THE HEAD AND CYLINDER IN THE MQ.

      ADD CODE TO PREVENT STATUS REPORT FROM CONTINUING AFTER 4 DEVICES.

      MOVE THIS ROUTINE FROM ATER HDRERR TO NEXT PAGE TO MAKE  
      ROOM FOR ADDED CODE IN DERR ROUTINE.

      ADD CODE TO ALLOW CONTROL S AND Q TO WORK WHEN  
      PRINTING LIST OF DATA ERRORS.

```

/
/      CHANGE POINTER FOR FILLING ROUTINE.
/
/      FIX VALUE USED TO SET DATA ERROR ERRORS TO PRINT.
/
/      ADD CODE TO ENABLE CONTROL S TO WORK WHEN PRINTING STATUS REPORT.
/
/      FIX INSTRUCTION WHICH TESTS INMODE FLAG.
/
/ MODIFICATIONS BY      DATE
/ -----
/ MIKE LETENDRE      FEB. 1981
/ ML 001      VERSION "B" CHANGES FOR THE VT278 COMPATABILITY UPGRADE.
/
0200  PAGE
/
7002  BSW=7002      /SWAP HALVES
7421  MQL=7421      /LOAD MQ FROM AC THEN CLEAR AC
7501  MQA=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
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 D E F I N I T I O N S *****/
/*****
/*****
/*****
/

```

```

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      RRWC=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 RLRS SKIP ON DRIVE READY)
6617      RLSE=6617      /SKIP ON COMPOSITE ERROR FLAG, CLEAR FLAG IF SET

```

```

/      0=0      HEAD DIRECTION AWAY FROM CENTER (LOWER CYLINDER)
/      0=1      HEAD DIRECTION TOWARD CENTER (HIGHER CYLINDER)
/      1=0      SELECT UPPER HEAD (TRACK 0)
/      1=1      SELECT LOWER HEAD (TRACK 1)
/      2=SPARE
/      3:11     CYLINDER ADDRESS/DIFFERENCE WORD
/
/*****
/
/      * COMMAND REGISTER B BIT DEFINITIONS *
/
/      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      RL01
/      4=1      RL02
/      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  DSTATE=0          /DRIVE STATE 0=NON-ACTIVE 7777=ACTIVE
0001  OLD CYL=1         /PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
0002  CUR CYL=2         /CURRENT CYLINDER ADDRESS BY READ HEADER
0003  NEW CYL=3         /NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
0004  BITS1=4          /BITS TRANSFERRED LOW ORDER
0005  BITS2=5          /BITS TRANSFERRED WORD 2
0006  BITS3=6          /BITS TRANSFERRED HIGH ORDER
0007  BITS4=7          /MULTIPLIER TIMES 10 BILLION
0010  SEEKS1=10        /SEEK COUNT
0011  SEEKS2=11        /SEEK COUNT MULTIPLIER TIMES 1000
0012  HRDERR=12        /HARD ERROR COUNT
0013  SFTERR=13        /SOFT ERROR COUNT
0014  DRVERR=14        /DRIVE ERROR COUNT
0015  SEKERR=15        /SEEK ERROR COUNT
0016  DATERR=16        /DATA ERROR COUNT
0017  TRKERR=17        /TRACKING ERROR COUNT
0020  DCR CER=20        /DATA CRC ERROR COUNT
0021  HCR CER=21        /HEADER CRC ERROR COUNT
0022  DLTERR=22        /DATA LATE ERROR COUNT
0023  OPIERR=23        /OPERATION INCOMPLETE ERROR COUNT
0024  HNFERR=24        /HEADER NOT FOUND ERROR COUNT
0025  CTLERR=25        /CONTROLLER ERROR COUNT
0026  READS=26         /NUMBER OF READ OPERATIONS
0027  WRTS=27          /NUMBER OF WRITE OPERATIONS
0030  WRDCNT=30        /INITIAL WORD COUNT SENT
0031  INITCA=31        /INITIAL CURRENT DATA BUFFER ADDRESS
0032  SECADD=32        /SECTOR ADDRESS SENT
0033  XCOMA=33         /COMMAND REGISTER A SENT
0034  XCOMB=34         /COMMAND REGISTER B SENT
0035  DACNT=35         /DRIVE ACCESS ATTEMPT COUNT (DRIVE NOT READY)
0036  POSFLG=36        /POSITION VERIFICATION NEEDED IF = 7777
0037  RDYFLG=37        /7777=READY PENDING FOLLOWING ISSUE OF A SEEK 0=NOT SEEKING
0040  XENDSC=40        /EXPECTED FINAL SECTOR ADDR REG. CONTENTS
0041  DRVBSY=41        /7777=DRIVE IS BUSY DOING A READ OR WRITE 0=DRIVE NOT BUSY
/
/
/
0010  OFFSET=0010      /DEFINE FIELD OFFSET OF CONSOLE PACKAGE
0200  RSTART=200       /DEFINE PROGRAM RESTART ADDRESS
/
/DEFINE DATA BUFFERS
/
7200  BUF0=7200        /DEFINE BUFFER 0 FIELD 0 ADDRESS
5600  BUF1=5600        /DEFINE BUFFER 1 FIELD 1 ADDRESS
0000  BUF2=0000        /DEFINE BUFFER 2 FIELD 2 ADDRESS
/
0400  BUFSZ0=0400      /DEFINE BUFFER 0 FLD 0 SIZE (256 DECIMAL)
2000  BUFSZ1=2000      /DEFINE BUFFER 1 FLD 1 SIZE (1024 DECIMAL)
7600  BUFSZ2=7600      /DEFINE BUFFER 2 FLD 2 SIZE (3968 DECIMAL)
/

```

HP 002

HP 002

/\*\*\*\*\*

```

/
/
/      * START OF PAGE ZERO CODE *
/
0000      *0
/
0000 0302      LOCO,  *B      /REVISION B
0001 5402      ADDR51, JMP I      ADDR2 /GO TO INTERRUPT SERVICE ROUTINE
0002 4000      ADDR52, INT8VC      /ADDRESS OF INTERRUPT SERVICE ROUTINE
0003 4022      ADDR53, RESTR      /POWER FAIL RESTART ADDRESS
0004 0000      ADDR54, 0      /RESERVED FOR ODT
0005 0000      ADDR55, 0      /RESERVED FOR ODT
0006 0000      ADDR56, 0      /RESERVED FOR ODT
0007 0000      ADDR57, 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      HCN1, 0      /HARDWARE CONFIGURATION WORD #1
0022 0400      HCN2, 0400      /HARDWARE CONFIGURATION WORD #2
0023 0000      HCN3, 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      TBL5AV, 0
/
/
0036 0000      BITTMP, 0      /BITMOD SAVE AREA
0037 0000      SAVWRD, 0      /WRDCNT SAVE AREA
0040 0000      SECCNT, 0      /SECTOR COUNTER FOR DATA BUFFER
0041 0000      WDCNTR, 0      /WORD COUNTER FOR DATA BUFFER
0042 0003      K3, 0003
0043 0005      K5, 0005
0044 0007      K7, 0007
0045 0070      K70, 0070

```



```

0046 0100 K100, 0100
0047 0260 K260, 0260
0050 0377 K377, 0377
0051 0400 K400, 0400
0052 0777 K777, 0777
0053 1000 K1000, 1000
0054 2000 K2000, 2000
0055 7777 M1, 7777
0056 7774 M4, -4
0057 7400 M400, -400

```

HP 001

```

/*****
/

```

```

/THESE LOCATIONS ARE REINITIALIZED TO DEFAULT VALUES WHEN
/THE PROGRAM IS RESTARTED IN ACCEPT MODE. DO NOT INSERT, DELETE
/OR REARRANGE THESE VALUES.
/

```

0060 TBLSTR=.

```

0060 0000 DMPFLG, 0 /DATA CRC DATA DUMP FLAG 0=YES 7777=NO
0061 0000 LIMFLG, 0 /DROP DRIVE ON OPERATION LIMITS REACHED 0=YES 7777=NO
0062 0000 CBUSY, 0 /CONTROLLER BUSY FLAG (DSC 60,61) 0=NOT BUSY 7777=BUSY
0063 0000 MINCYL, 0 /MINIMUM CYLINDER TO USE
0064 0000 MINSEC, 0 /MINIMUM SECTOR TO USE
0065 0000 SNGLHD, 0 /SINGLE SURFACE TO USE DURING DATA TRANSFERS
0066 7777 BUFLD0, 7777 /DRIVE ZERO BUFFER FIELD
0067 7777 BUFLD1, 7777 /DRIVE ONE BUFFER FIELD
0070 7777 BUFLD2, 7777 /DRIVE TWO BUFFER FIELD
0071 7777 BUFLD3, 7777 /DRIVE THREE BUFFER FIELD
0072 7777 SNGFLG, 7777 /SINGLE SURFACE DATA TRANSFERS FLAG 0=YES 7777=NO
0073 7777 RDFLG, 7777 /READ ONLY FLAG 0=YES 7777=NO
0074 7777 CONFLG, 7777 /CONSTANT DATA FLAG 0=YES 7777=NO
0075 0001 FIXWRD, 0001 /FIXED NUMBER OF WORDS TO USE FOR DATA
/TRANSFERS. IF = 0001 MEANS USE RANDOM WORD COUNT
/IF NOT = 0001 THEN USE AS FIXED WORD COUNT

```

HP 003

```

0076 0001 RETLIM, 0001 /RETRY LIMIT
0077 0001 SEKRET, 0001 /SEEK TO TRACK RETRY LIMIT
0100 0001 XFRLIM, 0001 /DATA TRANSFER LIMIT TO DROP DRIVE TIMES 10(10)
0101 0001 SEKLIM, 0001 /SEEK OPERATION LIMIT TO DROP DRIVE TIMES 10(3)
0102 0001 NUMFLD, 0001 /NUMBER OF ADDITIONAL FIELDS R/W MEMORY
0103 0012 SFTLIM, 0012 /SOFT ERROR LIMIT TO DROP DRIVE
0104 0001 DATPAT, 0001 /DEFAULT CONSTANT DATA PATTERN TO USE (WORST CASE)
0105 0376 DATERN, 0376 /NUMBER OF DATA WORD ERRORS REPORTED
0106 0777 MAXCYL, 0777 /MAXIMUM CYLINDER TO USE
0107 0047 MAXSEC, 0047 /MAXIMUM SECTOR TO USE

```

HP 001

0110 TBLED=.

```

/
/ * END OF CRITICAL LOCATIONS *
/

```

```

/*****

```

```

/
/
0110 0000 RETCNT, 0 /RETRY COUNT
0111 0000 BITMOD, 0 /BIT MODE SAVE AREA
0112 0000 TBLPNT, 0 /DRIVE PARAMETER TABLE POINTER
0113 0000 PWRFLG, 0 /POWER FAIL RESTART FLAG 7777=POWER FAILED
0114 0000 RSTFLG, 0 /PROGRAM RESTART FLAG 0=NOT RESTARTED 7777=RESTARTED
0115 0000 FUNCOD, 0 /FUNCTION CODE
0116 7200 BUFAD0, BUF0 /FIELD 0 BUFFER ADDRESS
0117 5600 BUFAD1, BUF1 /FIELD 1 BUFFER ADDRESS
0120 0000 CNTRI, 0
0121 0000 CDFSAV, 0
0122 6001 KION, ION
0123 6201 KCDF, CDF
0124 0000 TEMP1, 0 /TEMPORARY STORAGE FOR USE IN MAIN CODE
0125 0000 BUFNUM, 0
0126 0000 NUMSEC, 0
0127 0000 DRVcnt, 0
0130 0000 FLGSAV, 0 /CONTROLLER FLAGS AFTER INTERRUPT, BIT 11 = COMPOSITE ERROR
/ BIT 10 = DONE

0131 0000 BADPNT, 0 /BAD SECTOR POINTER
0132 0000 PATCNT, 0
0133 0000 BDCNT, 0 /BAD DATA WORD COUNT
0134 0000 DERFLG, 0 /DATA ERROR FLG
0135 0000 DRVTMP, 0 /TEMPORARY DRIVE NUMBER STORAGE (CURDRV)
0136 0000 PATMP, 0
0137 0000 DTMPA, 0
0140 0000 SECSAV, 0 /SECTOR ADDRESS SAVE AREA
0141 0000 WRDSAV, 0 /WORD COUNT SAVE AREA
0142 0000 ASAV, 0 /COMMAND A SAVE AREA
0143 0000 NEWSAV, 0 /NEW CYLINDER SAVE AREA
0144 0000 WDCTMP, 0 /TEMPORARY WORD COUNT STORAGE
/
/*****
/*****
/***** SUBROUTINE CALLS *****/
/*****
/*****
/*****
/
0145 4545 APTCHK=JMS I . /CALL TO CHECK FOR ON APT
5163 CHKAPT /RETURN CALL+1 IF ON APT, CALL+2 IF NOT ON APT

0146 4546 C8CALL=JMS I . /CALL CONSOLE PACKAGE
6252 XC8CAL /AC=0 MEANS PRINT SR=XXXX
/AC NOT = 0 MEANS DECODE ASCII CHAR IN AC

0147 4547 SETPNT=JMS I . /ROUTINE TO SET UP "TBLPNT" IN DRIVE STATE TABLES
4313 XSETP /CALL+1 = TABLE ENTRY OFFSET

```

0150	4550 2151	GETSWR=JMS I . SWRGET	/ROUTINE TO GET HARDWARE OR SOFTWARE SWR /RETURN CALL+1 WITH VALUE OF SWR IN AC
0151	4551 4355	STAPRT=JMS I . PRTSTA	/ROUTINE TO PRINT STATUS MESSAGES /SEE LISTING OF SUBROUTINE FOR FURTHER DETAILS
0152	4552 5145	RANDOM=JMS I . XRAND	/ROUTINE TO GENERATE A RANDOM NUMBER /RETURN WITH NUMBER IN AC
0153	4553 5321	ERROR=JMS I . XERROR	/ROUTINE TO HANDLE GENERAL STATUS ERRORS /RETURN CALL+1 IF ERROR NOT RECOVERABLE /ELSE RETURN CALL+2 IF RECOVERABLE
0154	4554 4600	ERRCHK=JMS I . CHKERR	/CHECK FOR ERRORS STATUS WORD 1 OR COMPOSITE ERROR FLG /RETURN CALL+2 IF ERROR, CALL+1 IF OK
0155	4555 5067	GO=JMS I . XGO	/CALL TO ROUTINE TO EXECUTE A COMMAND
0156	4556 4272	WAITDN=JMS I . DNWAIT	/CALL TO ROUTINE TO WAIT FOR DONE INTERRUPT
0157	4557 2740	APTERR=JMS I . ERRAPT	/CALL TO ON APT ERROR ROUTINE
0160	4560 6645	XFER=JMS I . EXFER	/CALL TO ROUTINE TO EXECUTE A READ OR WRITE
0161	4561 4400	SETPOS=JMS I . POSSET	/CALL TO ROUTINE TO SEEK TO ABSOLUTE CYLINDER AND SURFACE /CALL+1 IS HEAD IN BIT 1,CYL IN BITS 3;11
0162	4562 5200	GETSTA=JMS I . STAGET	/CALL TO ROUTINE TO GET DRIVE STATUS /RETURN ONLY IF SUCCESSFUL
0163	4563 4205	STACHK=JMS I . CHKSTA	/CALL TO ROUTINE TO CHECK STATUS WORD #1
0164	4564 2747	MESAG=JMS I . MESGO	/CALL TO MESSAGE PRINT ROUTINE FLD 1
0165	4565 1350	YESRN=JMS I . YNGO	/CALL TO ROUTINE TO GET YES OR NO ANSWER
0166	4566 1357	PRNT=JMS I . PRNTGO	/CALL TO ROUTINE TO PRINT ONE CHARACTER
0167	4567 6607	GETOCT=JMS I . OCTGET	/CALL TO ROUTINE TO GET ONE OCTAL DIGIT
0170	4570 2164	PRNAC=JMS I . ACPRN	/CALL TO ROUTINE TO PRINT CONTENTS OF AC
	4571	DOCRLF=JMS I .	/CALL TO ROUTINE TO EXECUTE A <CR> AND <LF>

```

0171 6703          CRLFDO
          4572      SPACE2=JMS I .          /CALL TO ROUTINE TO PRINT 2 SPACES .
0172 4557          SPACES
          4573      RESET=JMS I .           /CALL TO ROUTINE TO RESET DRIVE
0173 3114          XRESET
          4574      ADDR=JMS I .            /CALL TO ROUTINE TO ADD DRIVE
0174 3200          XADDRV
          4575      DECPRN=JMS I .          /CALL TO ROUTINE TO PRINT FOUR DECIMAL DIGITS
0175 2160          PRNDEC
          4576      VT278= JMS I.
0176 3554          XVT278
          4577      WRDCHK= JMS I .
0177 0754          XWRDCH

```

```

/
/
/*****
/
/

```

```

/
/      * 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 6201      CDF      00          /CDF TO PRGM FLD
0211 1113      TAD      PWRFLG      /GET POWER FAILED RESTART FLAG
0212 7640      SZA CLA              /SKIP IF NOT POWER FAIL RESTART
0213 5774      JMP      PWRST        /GO DO A POWER FAIL RESTART
0214 1114      TAD      RSTFLG      /GET PROGRAM RESTART FLAG
0215 7650      SNA CLA              /SKIP IF PROGRAM RESTARTED
0216 5222      JMP      ACHK1        /PROGRAM NOT RESTARTED
0217 6213      CDI      10          /CDI TO FLD 1
0220 5621      JMP I    .+1          /GO TO FLD 1
0221 0203      CLRQ              /DESTINATION IN FLD 1
/
0222 4545      ACHK1, APTCHK        /CHECK FOR ON APT
0223 7610      SKP CLA              /ON APT RETURN
0224 5230      JMP      CONGO        /NOT ON APT RETURN
0225 6213      CDI      10          /CDI TO FLD 1
0226 5627      JMP I    .+1          /GO TO FLD 1
0227 0222      INIT              /FLD 1 DESTINATION

```

```

0230 7240 / CONGO, CLA CMA
0231 3114 DCA RSTFLG /SET PROGRAM RESTART FLAG
0232 6213 CDI 10 /CDI TO FLD 1
0233 5634 JMP I .+1 /GO TO FLD 1
0234 0200 CONSET /FLD 1 DESTINATION

/
/
/RESET ALL DRIVES, IGNORE ERRORS (DUE TO VOLUME CHECK)
/
0235 7301 MAIN, CLA CLL IAC /SET BIT 11 FOR KEYBOARD INTERRUPT ENABLE
0236 6035 KIE /ENABLE KEYBOARD INTERRUPTS
0237 6045 TIE
0240 7300 CLA CLL /KIE DOES NOT CLEAR AC
0241 1056 TAD M4 /4 DRIVES
0242 3127 DCA DRVCNT /SET UP DRIVE COUNTER
0243 3024 DCA CURDRV /START WITH DRV 0
0244 4573 SETUP, RESET /RESET A DRIVE IGNORING ERRORS
0245 4773 JMS CHKACT /GO CHECK FOR DRIVE ACTIVE
0246 5305 JMP NXTDV1 /DRV NOT ACTIVE GO DO NEXT DRV
0247 4561 SETPOS /SEEK TO BAD SECTOR FILE TRACK
0250 2777 2777 /CYL 777 HD 1 (BAD SECTOR FILE)
0251 5305 JMP NXTDV1 /NOT SUCCESSFUL - DRIVE DROPPED
0252 4772 JMS GETBSF /RETRIEVE BAD SECTOR FILE
0253 5305 JMP NXTDV1 /NOT SUCCESSFUL - DRIVE DROPPED

/
0254 4771 WRTPK, JMS WRTPAK /WRITE ENTIRE PACK WITH SINGLE DATA PATTERN
0255 5305 JMP NXTDV1 /NOT SUCCESSFUL - DRIVE DROPPED
0256 4545 APTCHK /CHECK FOR ON APT
0257 5270 JMP READPK /ON APT SKIP MESSAGE, GO READ ENTIRE PACK
0260 4571 DOCRLF /DO A <CR> AND <LF>
0261 4564 MESAG
0262 4075 DRVMSG /DRIVE
0263 1024 TAD CURDRV /GET CURRENT DRIVE NUMBER
0264 1047 TAD K260 /ADD ASCII BASE CODE
0265 4566 PRNT /PRINT DRIVE NUMBER
0266 4564 MESAG
0267 4552 PAKWRT /PACK WRITTEN OK

/
0270 4770 READPK, JMS RDPK /READ AND VERIFY ENTIRE PACK
0271 5305 JMP NXTDV1 /NOT SUCCESSFUL - DRIVE DROPPED
0272 4545 APTCHK /CHECK FOR ON APT
0273 5305 JMP NXTDV1 /ON APT SKIP MESSAGE, GO DO NEXT PACK
0274 4571 DOCRLF /DO A <CR> AND <LF>
0275 4564 MESAG
0276 4075 DRVMSG /DRIVE
0277 1024 TAD CURDRV /GET DRIVE NUMBER
0300 1047 TAD K260 /ADD ASCII BASE CODE
0301 4566 PRNT /PRINT DRIVE NUMBER
0302 4564 MESAG
0303 4562 PAKRD /PACK READ OK
0304 4571 DOCRLF /DO A <CR> AND <LF>

/
0305 2024 NXTDV1, ISZ CURDRV /INCREMENT DRIVE NUMBER
0306 2127 ISZ DRVCNT /INCREMENT DRIVE COUNT

```

HP 001

```

/
/
/ NOW EXERCISE DRIVES
/
0310 1056 ANYDRV, TAD M4
0311 3127 DCA DRVCNT /SET UP COUNTER TO CHECK FOR ANY ACTIVE DRIVES
0312 3024 DCA CURDRV /CURDRV=0
0313 4773' ANYLP, JMS CHKACT /CHECK FOR DRIVE ACTIVE
0314 7610 SKP CLA /SKIP IF NOT ACTIVE
0315 5327 JMP DRIVE /A DRIVE IS ACTIVE - PROGRAM IS ALIVE
0316 2024 ISZ CURDRV /INCREMENT DRIVE #
0317 2127 ISZ DRVCNT /INCREMENT DRIVE COUNTER
0320 5313 JMP ANYLP /TRY AGAIN FOR ANY ACTIVE DRIVE
0321 4545 APTCHK /CHECK FOR ON APT NO DRIVES ACTIVE
0322 4557 APTERR /ON APT - NO DRIVES ACTIVE
0323 4564 MESAG
0324 4315 NODRVS /NOT ON APT - NO ACTIVE DRIVES
0325 4574 ADDR /GO CHECK FOR ADDING DRIVES
0326 5310 JMP ANYDRV /GO CHECK FOR ANY ACTIVE DRIVES
/
0327 7301 DRIVE, CLA CLL IAC /SET BIT 11 FOR KEYBOARD INTERRUPT ENABLE
0330 6035 KIE /ENABLE KEYBOARD INTERRUPTS (DISABLED IN STAGET)
0331 6045 TIE
0332 7300 CLA CLL /AC NOT CLEARED BY KIE
0333 1367 TAD (-10 /SET UP FOR 8 ATTEMPTS AT RANDOMLY SELECTING A DRIVE
0334 3120 DCA CNTR1 /SET UP LOOP COUNTER
0335 4550 GETSWR /GET SWITCHES
0336 0366 AND (10 /MASK BIT 8 (ADD OR DROP DRIVES)
0337 7640 SZA CLA /SKIP IF NOT ADDING DRIVES
0340 4574 ADDR /GO ADD OR DROP DRIVE
0341 4552 DRVLP3, RANDOM /GET A RANDOM NUMBER
0342 0042 AND K3 /MASK DRIVE # BITS
0343 3024 DCA CURDRV /SAVE DRIVE # AS CURRENT DRIVE
0344 4773' JMS CHKACT /GO CHECK FOR DRIVE ACTIVE
0345 7610 SKP CLA /SKIP IF DRIVE NOT ACTIVE
0346 5765' JMP BSYCHK /GO CHECK FOR THIS DRIVE BUSY
0347 2120 ISZ CNTR1 /INCREMENT LOOP COUNT
0350 5341 JMP DRVLP3 /TRY AGAIN FOR ACTIVE DRIVE
0351 5310 JMP ANYDRV /TRY AGAIN FOR ANY ACTIVE DRIVES
/
/*****8
/
/ 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
/

```

```

0352 0000 /
0353 7300 DELAY, 0
0354 1752 CLA CLL
0355 3360 TAD I DELAY
0356 6212 DCA DLYCAL+1
0357 4764 CIF 10
0358 7402 DLYCAL, JMS I (XDELAY
0359 2352 HLT/DELAY COUNT
0360 5752 ISZ DELAY
0361 JMS I DELAY
0362 /
/
/
/
0364 1533
0365 0400
0366 0010
0367 7770
0370 2200
0371 1600
0372 3440
0373 1340
0374 1510
0375 0022
0376 0021
0377 0020
0378 0400

PAGE
/
/
0400 4547 BSYCHK, SETPNT
0401 0041 DRVBSY
0402 1512 TAD I TBLPNT
0403 7650 SNA CLA
0404 5250 JMP POS1
/
0405 4547 DBUSYA, SETPNT
0406 0037 RDYFLG
0407 1512 TAD I TBLPNT
0410 7650 SNA CLA
0411 5241 JMP INCTRY
0412 4556 WAITDN
0413 4777 JMS FATAL
0414 4554 ERRCHK
0415 5220 JMP .+3
0416 4553 ERROR
0417 5776 JMP DRIVE
0420 4550 GETSWR
0421 0375 AND (4
0422 7640 SZA CLA
0423 5227 JMP .+4
0424 4774 JMS DATCHK
0425 4553 ERROR
0426 5776 JMP DRIVE
0427 4562 GETSTA
0430 4554 ERRCHK

/GET 2'S COMPLEMENT OF NUMBER OF SECONDS TO DELAY
/SAVE FOR DELAY ROUTINE CALL
/GOING TO FLD 1
/GO TO FLD 1 DELAY ROUTINE
/SECONDS MULTIPLIER
/INCREMENT FOR RETURN
/RETURN

/SET TBLPNT TO DRIVE BUSY FLAG
/TABLE INDEX
/GET DRIVE BUSY FLAG
/SKIP IF DRIVE BUSY
/DRIVE NOT BUSY GO CHK POS VERIFY FLG

/SET TBLPNT TO READY PENDING FLG
/TBL INDX
/GET READY PENDING FLG
/SKIP IF READY PENDING,(DRIVE SEEKING)
/DRIVE NOT SEEKING, GO INCREMENT ATTEMPT COUNT
/WAIT FOR DONE
/NO DONE
/CHECK FOR ERRORS
/NO ERROR RETURN
/ERROR FROM LAST OPERATION
/GO GET NEW DRIVE, DRIVE DROPPED RETURN
/GET SWITCH REGISTER
/MASK BIT 9, INHIBIT SOFTWARE DATA CHECK
/SKIP IF SOFTWARE DATA CHECK NOT INHIBITED
/DO NOT CHECK DATA
/GO CHECK DATA IF LAST OPERATION WAS READ
/DATA ERROR ON READ
/GO GET NEW DRIVE, DRIVE DROPPED AFTER READ RETRY
/GET STATUS NEXT DRIVE
/CHECK FOR ERRORS

```

0431	5234	JMP	+.3	/NO ERROR RETURN
0432	4553	ERROR		/STATUS ERROR
0433	5776'	JMP	DRIVE	/DRIVE DROPPED RETURN
0434	4563	STACHK		/DRIVE SEEKING, CHECK STATUS WORD 1
0435	5241	JMP	INCTRY	/DRIVE STATUS WORD #1 OK, GO INCREMENT ATTEMPT COUNT
0436	4553	ERROR		/DRIVE STATUS ERROR
0437	7000	NOP		/ERROR NOT RECOVERABLE - DRIVE DROPPED
0440	5776'	JMP	DRIVE	/GO GET NEW DRIVE - ERROR RECOVERED
/				
0441	4547	INCTRY,	SETPNT	/SET TBLPNT TO DRV ATTEMPT COUNT
0442	0035		DACNT	/TABLE INDEX
0443	2512		ISZ I TBLPNT	/INCREMENT DRIVE ATTEMPT COUNT
0444	5776'	JMP	DRIVE	/DRIVE STILL LEGALLY BUSY
0445	4553	ERROR		/DRIVE BUSY TOO LONG
0446	7000	NOP		/NOT RECOVERABLE RETURN - DRIVE DROPPED
0447	5776'	JMP	DRIVE	/GO GET NEW DRIVE
/				
0450	4547	POS1,	SETPNT	/SET TBLPNT TO POS VERIFY FLG
0451	0036		POSFLG	/TABLE INDEX
0452	1512		TAD I TBLPNT	/GET POS VERIFY FLG
0453	7650		SNA CLA	/SKIP IF POS VERIFY NEEDED
0454	5330	JMP	FUNGEN	/GO GENERATE RANDOM FUNCTION
/				
0455	4556	POSVER,	WAITDN	/WAIT FOR CONTROLLER NOT BUSY
0456	4777'	JMS	FATAL	/NO DONE INTERRUPT
0457	4554	ERRCHK		/CHECK FOR ANY ERRORS LAST OPERATION
0460	5263	JMP	+.3	/NO ERROR RETURN
0461	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
0462	5776'	JMP	DRIVE	/DRIVE DROPPED, GO GET NEW DRIVE
0463	4550		GETSWR	/GET SWITCH REGISTER
0464	0375	AND	(4	/MASK BIT 9, INHIBIT SOFTWARE DATA CHECK
0465	7640		SZA CLA	/SKIP IF SOFTWARE DATA CHECK
0466	5272	JMP	+.4	/DO NOT CHECK DATA, NO DCRC ERROR
0467	4774'	JMS	DATCHK	/CHECK DATA IF LAST OPERATION WAS READ
0470	4553	ERROR		/DATA ERROR ON READ
0471	5776'	JMP	DRIVE	/GO GET NEW DRIVE, DRIVE DROPPED AFTER READ RETRY
0472	4562	GETSTA		/GET STATUS NEXT DRIVE
0473	4554	ERRCHK		/CHECK FOR ANY ERRORS
0474	5277	JMP	+.3	/NO ERROR RETURN
0475	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
0476	5776'	JMP	DRIVE	/DRIVE DROPPED - ERROR NOT RECOVERABLE RETURN
0477	4773'	JMS	RDYCHK	/CHECK FOR DRIVE READY
0500	5303	JMP	+.3	/DRIVE READY RETURN
0501	4553	ERROR		/DRIVE NOT READY
0502	5776'	JMP	DRIVE	/DRIVE DROPPED - ERROR NOT RECOVERABLE RETURN
/				
0503	4772'	GETHDR,	JMS RDHDR	/GO READ HEADER
0504	4554		ERRCHK	/CHECK FOR ERRORS
0505	5310	JMP	+.3	/NO ERROR RETURN
0506	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
0507	5776'	JMP	DRIVE	/DRIVE DROPPED - ERROR NOT RECOVERABLE RETURN
0510	4547	SETPNT		/SET TBLPNT TO CURRENT CYL AND HD
0511	0002	CURCYL		/TABLE INDEX
0512	1512	TAD I	TBLPNT	/GET CURRENT CYL AND HEAD
0513	7041	CIA		/NEGATE



```

0514 4547      SETPNT      /SET TBLPNT TO EXPECTED CYL AND HEAD
0515 0003      NEWCYL      /TABLE INDEX
0516 1512      TAD I      TBLPNT /ADD EXPECTED CYL AND HEAD
0517 7650      SNA        CLA    /SKIP IF CURRENT CYL AND HEAD NOT = EXPECTED
0520 5323      JMP        .+3    /SEEK AND READ HEADER OK
0521 4553      ERROR      /GO HANDLE SEEK ERROR
0522 5776      JMP        DRIVE  /DRIVE DROPPED, GO GET NEW DRIVE
0523 4547      SETPNT      /SET TBLPNT TO POSITION VERIFICATION NEEDED FLAG
0524 0036      POSFLG      /TBL INDX
0525 3512      DCA I      TBLPNT /CLEAR POS VERIFY FLG
0526 4771      JMS        SEEKUP /GO UPDATE SEEK COUNT
0527 5776      JMP        DRIVE  /GO GET NEW DRIVE

/
/
/
/GET A FUNCTION CODE RANDOMLY SELECTED FROM A TABLE
/OF FUNCTION CODES
/
0530 4552      FUNGEN, RANDOM /GET RANDOM INDEX INTO FUNCTION TABLE
0531 0044      AND        K7     /MASK INDEX TO 3 BITS
0532 3115      DCA        FUNCOD /SAVE FUNCTION CODE INDEX
0533 1370      TAD        (FUNTBL /GET ADDRESS OF FUNCTION CODE TABLE
0534 1115      TAD        FUNCOD /ADD INDEX
0535 3115      DCA        FUNCOD /SAVE ADDRESS OF FUNCTION CODE IN TABLE
0536 1515      TAD I      FUNCOD /GET FUNCTION CODE FROM TABLE
0537 7450      SNA        /SKIP IF VALID FUNCTION
0540 5330      JMP        FUNGEN /NULL FUNCTION FOR R/W WEIGHTING, GET NEW FUNCTION
0541 3115      DCA        FUNCOD /SAVE ACTUAL FUNCTION CODE FROM TABLE

/
/
/
/NOW CHECK FOR READ ONLY (NO WRITES ALLOWED)
/IF PROGRAM SO INITIALIZED DURING QUESTIONS (NOT ACCEPT MODE)
/
/CHECK FOR SEEKS ONLY (SWR 7=1)
/
/
0542 7307      CHKSEK, CLA CLL IAC RTL /AC=4
0543 7041      CIA        /AC=-4
0544 1115      TAD        FUNCOD /ADD FUNCTION CODE
0545 7750      SPA SNA CLA /SKIP IF READ OR WRITE
0546 5767      JMP        CYLGEN /GO DO A SEEK
0547 4550      GETSWR      /GET SWITCH REGISTER
0550 0366      AND        (20    /MASK BIT 7 (SEEKS ONLY)
0551 7640      SZA CLA      /SKIP IF NOT SEEKS ONLY
0552 5767      JMP        CYLGEN /GO DO A SEEK

/
0553 1073      RONLY, TAD    RDFLG /GET READ ONLY FLAG
0554 7640      SZA CLA      /SKIP IF READ ONLY (NO WRITES)
0555 5765      JMP        DOXFER /READS AND WRITES OK
0556 1043      TAD        K5
0557 7041      CIA        /NEGATE
0560 1115      TAD        FUNCOD /ADD FUNCTION CODE
0561 7640      SZA CLA      /SKIP IF RANDOM FUNCTION WAS A WRITE

```

0562	5765	JMP	DOXFER	/GO DO A READ	
0563	5330	JMP	FUNGEN	/GET A NEW RANDOM FUNCTION	
/					
0565	1000				
0566	0020				
0567	0600				
0570	0744				
0571	3753				
0572	5243				
0573	4347				
0574	6000				
0575	0004				
0576	0327				
0577	6614				
	0600				
PAGE					
/					
/					
/					
/					
/PREPARE TO DO A SEEK					
/					
0600	4552	CYLGEN, RANDOM		/GET A RANDOM NUMBER	
0601	0377	AND	(2777	/MASK HEAD SELECT AND CYLINDER	HP 001
0602	3143	DCA	NEWSAV	/SAVE NEW CYL AND HEAD	
0603	1072	TAD	SNGFLG	/GET SINGLE SURFACE FLAG	
0604	7640	SZA	CLA	/SKIP IF SINGLE SURFACE BEING USED	
0605	5216	JMP	BDTRK	/NOT SINGLE SURFACE - CHK BAD SECTOR FILE	
0606	1065	TAD	SNGLHD	/GET SINGLE HEAD TO USE IN BIT 11	
0607	7112	CLL	RTR	/MOVE TO BIT 0	
0610	7010	RAR		/MOVE TO BIT 1	
0611	3124	DCA	TEMP1	/SAVE TEMPORARILY	
0612	1143	TAD	NEWSAV	/GET NEW CYL AND HEAD	
0613	0052	AND	K777	/MASK OFF HEAD SELECT	HP 001
0614	1124	TAD	TEMP1	/ADD SINGLE SURFACE HEAD	
0615	3143	DCA	NEWSAV	/SAVE NEW CYL AND SINGLE SURFACE HEAD	
/					
0616	1143	BDTRK, TAD	NEWSAV	/GET NEW CYL AND HEAD	
0617	7041	CIA		/NEGATE	
0620	1377	TAD	(2777	/ADD DISK ADDR OF BAD SECTOR FILE	HP 001
0621	7650	SNA	CLA	/SKIP IF NOT BAD SECTOR FILE	
0622	5200	JMP	CYLGEN	/GO GET NEW CYL AND HEAD	
0623	1143	TAD	NEWSAV	/GET NEW CYL AND HEAD	
0624	0052	AND	K777	/MASK OFF HEAD SELECT	HP 001
0625	3142	DCA	ASAV	/SAVE CYL ADDR TEMPORARILY	
0626	1142	TAD	ASAV	/GET NEW CYL AND HEAD	
0627	7041	CIA		/NEGATE	
0630	1106	TAD	MAXCYL	/ADD MAXIMUM CYL TO USE	
0631	7710	SPA	CLA	/SKIP IF NEW CYL < OR = MAXIMUM CYL TO USE	
0632	5200	JMP	CYLGEN	/GO GET NEW CYL AND HEAD	
0633	1063	TAD	MINCYL	/GET MINIMUM CYL TO USE	
0634	7041	CIA		/NEGATE	
0635	1142	TAD	ASAV	/GET NEW CYL AND HEAD	
0636	7710	SPA	CLA	/SKIP IF NEW CYL > OR = MINIMUM CYL TO USE	
0637	5200	JMP	CYLGEN	/GO GET NEW CYL AND HEAD	

/

/CHECK FOR FIXED WORD COUNT AND IF > 170 (MULTIPLE SECTORS)

/THEN MUST SEEK TO AN 8 BIT MODE TRACK

/

/ONLY EVEN CYL HEAD 0 OR ODD CYL HEAD 1 ALLOWED

/

0640	1075	TAD	FIXWRD	/GET FIXED WORD COUNT
0641	7710	SPA CLA		/SKIP IF WORD COUNT < OR = 2048
0642	5250	JMP	SEKCHK	/CHECK NEWCYL FOR 8 BIT MODE
0643	1075	TAD	FIXWRD	/GET FIXED WORD COUNT
0644	7041	CIA		/NEGATE
0645	1376	TAD	(252	/ADD 170(10)
0646	7700	SMA CLA		/SKIP IF > 170
0647	5261	JMP	SEKOK	/SEEK OK FIXED WORD COUNT NOT > 170
/				
0650	1143	SEKCHK, TAD	NEWSAV	/GET NEW CYL AND HEAD
0651	0375	AND	(2001	/MASK LSB AND HEAD
0652	7650	SNA CLA		/SKIP IF NOT EVEN CYL AND HEAD 0
0653	5261	JMP	SEKOK	/SEEK OK EVEN CYL AND HEAD 0
0654	1143	TAD	NEWSAV	/GET NEW CYL AND HEAD
0655	0375	AND	(2001	/MASK LSB AND HEAD
0656	1374	TAD	(-2001	/ADD 2'S COMP OF ODD CYL AND HEAD 1
0657	7640	SZA CLA		/SKIP IF ODD CYL AND HEAD 1
0660	5200	JMP	CYLGEN	/GO GET NEW HEAD AND CYL
/				
0661	4547	SEKOK, SETPNT		/SET TBLPNT TO CURCYL IN DRV TBL
0662	0002	CURCYL		/TABLE INDEX
0663	1512	TAD I	TBLPNT	/GET CURRENT CYL ADDR AND HEAD
0664	0052	AND	K777	/MASK OFF HEAD SELECT
0665	7041	CIA		/NEGATE
0666	1142	TAD	ASAV	/ADD NEW CYL
0667	7500	SMA		/SKIP IF NEW CYL < CURRENT CYL
0670	5273	JMP	.+3	/NEW CYL > OR = CURRENT CYL
0671	7041	CIA		/MAKE CYL DIF POSITIVE
0672	7410	SKP		
0673	1373	TAD	(4000	/SET HEAD DIRECTION FOR INWARD (HIGHER CYL)
0674	3142	DCA	ASAV	/SAVE COMMAND A
/				
/				
/				
0675	1143	SETA, TAD	NEWSAV	/GET NEW CYL AND HEAD
0676	0054	AND	K2000	/MASK HEAD SELECT
0677	1142	TAD	ASAV	/ADD NEW CYL TO HEAD SELECT
0700	3142	DCA	ASAV	/SAVE COMMAND A
0701	4556	WAITDN		/WAIT FOR DONE FROM LAST OPERATION
0702	4772	JMS	FATAL	/NO DONE FLAG LAST OPERATION
0703	4554	ERRCHK		/CHECK FOR ERRORS FROM LAST OPERATION
0704	5307	JMP	.+3	/NO ERROR RETURN
0705	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
0706	5771	JMP	DRIVE	/DRIVE DROPPED, GO GET NEW DRIVE
0707	4550	GETSWR		/GET SWITCH REGISTER
0710	0370	AND	(4	/MASK BIT 9 INHIBIT SOFTWARE DATA CHECK
0711	7640	SZA	CLA	/SKIP IF SOFTWARE DATA CHECK
0712	5316	JMP	.+4	/DO NOT CHECK DATA, NO DCRC ERROR

HP 001

0713	4767'	JMS	DATCHK	/CHECK DATA IF LAST OPR WAS READ
0714	4553	ERROR		/DATA ERROR ON READ
0715	5771'	JMP	DRIVE	/GO GET NEW DRIVE, DRIVE DROPPED AFTER READ RETRY
0716	4562	GETSTA		/GET CURRENT DRIVE STATUS
0717	4554	ERRCHK		/CHECK FOR ERRORS CURRENT DRIVE
0720	5323	JMP	..+3	/NO ERROR RETURN
0721	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
0722	5771'	JMP	DRIVE	/GO GET NEW DRIVE - DRIVE DROPPED RETURN
0723	4766'	JMS	RDYCHK	/CHECK FOR DRIVE READY
0724	5327	JMP	..+3	/DRIVE READY RETURN
0725	4553	ERROR		/DRIVE NOT READY
0726	5771'	JMP	DRIVE	/SKIP SEEK - DRIVE DROPPED RETURN
0727	4547	SETPNT		/SET TBBLPNT TO COMMAND A
0730	0033	XCOMA		/TBL INDX
0731	1142	TAD	ASAV	/GET COMMAND A GENERATED
0732	3512	DCA I	TBLPNT	/SAVE COMMAND A IN DRIVE TABLE
0733	4547	SETPNT		/SET TBLPNT TO NEW CYL AND HD
0734	0003	NEWCYL		/TBL INDX
0735	1143	TAD	NEWSAV	/GET NEW CYL AND HD GENERATED
0736	3512	DCA I	TBLPNT	/SAVE IN DRIVE TABLE
0737	4547	SETPNT		/SET TBLPNT TO SECTOR ADDRESS
0740	0032	SECADD		/TBL INDX
0741	3512	DCA I	TBLPNT	/CLEAR SECTOR ADDRESS FOR SEEK
0742	4765'	JMS	SEEK	/ISSUE SEEK
0743	5771'	JMP	DRIVE	/GO GET NEW DRIVE

/  
 /  
 /FUNCTION CODE TABLE  
 /RANDOMLY SELECTED  
 /3 WRITES,3 READS,1 SEEK,1READ DATA NO HDR CHK  
 /

0744 FUNTBL=.

0744	0000	0000	/NULL FUNCTION FOR 3/2 READ/WRITE RATIO
0745	0005	0005	/WRITE
0746	0005	0005	/WRITE
0747	0006	0006	/READ
0750	0006	0006	/READ
0751	0006	0006	/READ
0752	0003	0003	/SEEK
0753	0000	0000	/NULL FUNCTION

0754	0000	XWRDCH, 0
0755	1141	TAD WRDSAV
0756	7710	SPA CLA
0757	5362	JMP ..+3
0760	1364	TAD (BUFSZ2
0761	5754	JMP I XWRDCH
0762	2354	ISZ XWRDCH
0763	5754	JMP I XWRDCH
0764	7600	
0765	6314	
0766	4347	
0767	6000	

0770 0004  
 0771 0327  
 0772 6614  
 0773 4000  
 0774 5777  
 0775 2001  
 0776 0252  
 0777 2777  
 1000

PAGE

/

/

/\*\*\*\*\*

/SET UP TO DO A READ OR WRITE

/

/

1000 4547	DOXFER.	SETPNT		/SET TBLPNT TO CURRENT CYL AND HD
1001 0002		CURCYL		/TABLE INDEX
1002 1512		TAD I	TBLPNT	/GET CURRENT CYL AND HD
1003 7110		CLL RAR		/MOVE LSB OF CURRENT CYL INTO LINK
1004 7630		SZL CLA		/SKIP IF CURRENT CYL IS EVEN LSB=0
1005 5214		JMP	ODCYL	/LSB=1 CURRENT CYL IS ODD
/				
1006 1512	EVNCYL.	TAD I	TBLPNT	/GET CURRENT CYL AND HD
1007 7106		CLL RTL		/MOVE HD SELECT INTO LINK
1010 7620		SNL CLA		/SKIP IF HEAD 1 USE 12 BIT MODE
/				
1011 1053	HDOA.	TAD	K1000	/USE 8 BIT MODE HD0 AND EVEN CYL
1012 3111		DCA	BITMOD	/SAVE BIT MODE
1013 5221		JMP	RANBUF	/GO DETERMINE BUFFER
/				
1014 1512	ODCYL.	TAD I	TBLPNT	/GET CURRENT CYL AND HD
1015 7106		CLL RTL		/MOVE HD SELECT INTO LINK
1016 7630		SZL CLA		/SKIP IF USING HD0
1017 1053		TAD	K1000	/USE HD1 8 BIT MODE
/				
1020 3111	HDOB.	DCA	BITMOD	/SAVE BIT MODE
/				
1021 4552	RANBUF.	RANDOM		/GET A RANDOM NUMBER
1022 0044		AND	K7	/MASK TO 3 BITS
1023 3125		DCA	BUFNUM	/SAVE AS POSSIBLE BUFFER NUMBER
1024 1102		TAD	NUMFLD	/GET NUMBER OF FIELDS MEMORY
1025 7041		CIA		
1026 1125		TAD	BUFNUM	/ADD RANDOM BUFFER NUMBER FIELD
1027 7740		SMA SZA	CLA	/SKIP IF VALID BUFFER BY MEM SIZE
1030 5221		JMP	RANBUF	/GO GET NEW RANDOM BUFFER
/				
1031 1056		TAD	M4	
1032 3127		DCA	DRVCNT	/SET UP COUNTER TO CHECK FOR BUFFER IN USE
1033 7340		CLA CLL	CMA	
1034 1377		TAD	(BUFLD0	/ADDR OF BUFFER TABLE-1
1035 3010		DCA	AUTO10	/SET BUFFER TABLE POINTER
1036 1125	BUFLPB.	TAD	BUFNUM	/GET BUFFER NUMBER SELECTED

1037	7041	CIA		/ADD BUFFER TABLE VALUE		
1040	1410	TAD I	AUTO10	/SKIP IF BUFFER NOT IN USE		
1041	7650	SNA CLA		/BUFFER IN USE - GO GET NEW RANDOM BUFFER		
1042	5221	JMP	RANBUF	/INCREMENT BUFFER TABLE COUNTER		
1043	2127	ISZ	DRVCNT	/GO CHECK NEXT DRIVE'S BUFFER		
1044	5236	JMP	BUFLPB			
/VALID BUFFER AND NOT IN USE						
1045	7340	CONWRD,	CLA CLL CMA	/AC=-1		
1046	1075	TAD	FIXWRD	/ADD FIXED WRD COUNT IF ANY		
1047	7650	SNA	CLA	/SKIP IF USING FIXED WORD COUNT		
1050	5255	JMP	RANWRD	/GO GET RANDOM WORD COUNT		
1051	1075	TAD	FIXWRD	/GET FIXED WORD COUNT		
1052	7041	CIA		/FORM 2'S COMPLEMENT WORD COUNT		
1053	3141	DCA	WRDSAV	/SAVE FIXED 2'S COMPLEMENT WRD CNT		
1054	5265	JMP	BUFCHK	/GO CHECK BUFFER SIZE		
1055	4552	RANWRD,	RANDOM	/GET A RANDOM NUMBER		
1056	3141	DCA	WRDSAV	/SAVE AS POSSIBLE ABSOLUTE WORD COUNT		
1057	1141	TAD	WRDSAV	/GET POSSIBLE WORD COUNT		
1060	7500	SMA		/SKIP IF NEGATIVE		
1061	5265	JMP	BUFCHK	/GO CHECK BUFFER SIZE, WORD COUNT > 3		
1062	1042	TAD	K3	/ADD 3 TO TEST FOR 1ST 2 WORDS + AT LEAST 1 DATA WORD		
1063	7740	SMA SZA	CLA	/SKIP IF WORD COUNT OK		
1064	5255	JMP	RANWRD	/GO GET NEW RANDOM WORD COUNT, TOO SMALL		
1065	7344	BUFCHK,	CLA CLL CMA RAL	/AC=-2	FLD 0 = 7776	HP 002
1066	1125	TAD	BUFNUM	/ADD BUFFER NUMBER (FIELD)	FLD 1 = 7777	HP 002
1067	7440	SZA		/SKIP IF FLD #2 BEING USED	FLD 2 = 0000	HP 002
1070	5275	JMP	CONT1			
1071	4577	WRDCHK				
1072	5307	JMP	EXIT1	/CHECK THAT BUFFER WILL FIT IN FIELD 2		HP 002
1073	7300	CLA CLL				
1074	5317	JMP	BITGET			
1075	7700	CONT1,	SMA CLA	/SKIP IF BUFFER #1 OR #0		HP 002
1076	5317	JMP	BITGET	/GO GET BIT MODE, NOT BUFFER 0, 1, OR 2.		
1077	1141	TAD	WRDSAV	/GET WORD COUNT		
1100	7700	SMA	CLA	/SKIP IF WORD COUNT < OR = 2048(10)		
1101	5312	JMP	FIXCHK	/WORD COUNT TOO LARGE FOR EITHER BUFFER #0 OR #1		
1102	1125	TAD	BUFNUM	/GET BUFFER NUMBER		
1103	7640	SZA	CLA	/SKIP IF USING BUFFER 0		
1104	1054	TAD	K2000	/USE BUFFER 1 SIZE		HP 002
1105	7450	SNA		/SKIP IF USING BUFFER 1		HP 002
1106	1051	TAD	K400	/USE BUFFER 0 SIZE		HP 002
1107	1141	EXIT1,	TAD WRDSAV			
1110	7700	SMA CLA				
1111	5317	JMP	BITGET			
/BUFFER SIZE AND WORD COUNT DO NOT MATCH						

```

1112 7340  FIXCHK, CLA CLL CMA      /AC=-1
1113 1075      TAD      FIXWRD  /ADD FIXED WORD COUNT IF ANY
1114 7640      SZA      CLA      /SKIP IF USING RANDOM WORD COUNT
1115 5776      JMP      CYLGEN  /DO A SEEK, WORD COUNT TOO LARGE AND FIXED
1116 5255      JMP      RANWRD  /GO GET NEW RANDOM WORD COUNT

/
1117 1111  BITGET, TAD      BITMOD /GET BIT MODE BASED ON CURRENT TRACK
1120 7640      SZA      CLA      /SKIP IF 12 BIT MODE
1121 5333      JMP      MODE8    /GO HANDLE 8 BIT MODE

/
1122 7301  MODE12, CLA CLL IAC     /NUMBER OF SECTORS = 1 (12 BIT MODE)
1123 3126      DCA      NUMSEC    /GET WORD COUNT
1124 1141      TAD      WRDSAV    /SKIP IF < OR = 2048
1125 7500      SMA      /WORD COUNT TOO LARGE FOR 12 BIT MODE
1126 5312      JMP      FIXCHK    /ADD MAXIMUM FOR 12 BIT SECTOR (170)
1127 1375      TAD      (252      /SKIP IF WORD COUNT NOT OK FOR 12 BIT MODE
1130 7700      SMA      CLA      /WORD COUNT OK, GO GET RANDOM SECTOR
1131 5774      JMP      RANSEC    /GO FIX WORD COUNT IF POSSIBLE
1132 5312      JMP      FIXCHK

/
1133 3126  MODE8,  DCA      NUMSEC /CLEAR NUMBER OF SECTORS COUNT
1134 1141      TAD      WRDSAV    /GET WORD COUNT
1135 7500      SMA      /SKIP IF < OR = 2048(10)
1136 5347      JMP      POSWRD    /GO HANDLE POSITIVE WORD COUNT

/
1137 1051  NEGWRD, TAD      K400    /ADD 256(10) = 1 SECTOR
1140 2126      ISZ      NUMSEC    /INCREMENT SECTOR COUNT
1141 7510      SPA      /SKIP IF DONE COUNTING SECTORS
1142 5337      JMP      NEGWRD    /NOT DONE COUNTING SECTORS, CONTINUE

/
1143 1373      TAD      (-375
1144 7740      SMA SZA CLA      /SKIP IF FINAL SECTOR IS AT LEAST 3 WORDS
1145 5354      JMP      ADJUST    /FINAL SECTOR TOO SMALL, GO ADJUST WORD COUNT
1146 5774      JMP      RANSEC    /FINAL SECTOR OK, GO GET RANDOM STARTING SECTOR

/
1147 1051  POSWRD, TAD      K400    /ADD 256(10) = 1 SECTOR
1150 2126      ISZ      NUMSEC    /INCREMENT SECTOR COUNT
1151 7500      SMA      /SKIP IF WORD COUNT LEFT BECOMES < OR = 2048(10)
1152 5347      JMP      POSWRD    /STILL MORE SECTORS AND REMAINING WORD COUNT STILL POSITIVE
1153 5337      JMP      NEGWRD    /NOW GO DO THE LAST 8 SECTORS

/
1154 7305  ADJUST, CLA CLL IAC RAL /AC=2
1155 1141      TAD      WRDSAV    /ADD WORD COUNT
1156 3141      DCA      WRDSAV    /SAVE ADJUSTED WORD COUNT
1157 1126      TAD      NUMSEC    /GET NUMBER OF SECTORS
1160 1055      TAD      M1        /SUBTRACT 1 FROM NUMBER OF SECTORS
1161 3126      DCA      NUMSEC    /SAVE ADJUSTED NUMBER OF SECTORS
1162 5774      JMP      RANSEC    /NOW GO GET RANDOM STARTING SECTOR

/
/ROUTINE TO DROP A DRIVE AND RELEASE THE BUFFER
/
/   CALLED BY:      JMS      KILDRV
/
/RETURN CALL+1

```

```

/
/
1163 0000 KILDRV, 0
1164 7300 CLA CLL
1165 4547 SETPNT /SET TBLPNT TO DRIVE STATE
1166 0000 DSTATE /TBL INDX
1167 3512 DCA I TBLPNT /DROP THE DRIVE IN DRIVE STATE TABLE
1170 4772 JMS RLSBUF /RELEASE THE BUFFER
1171 5763 JMP I KILDRV /RETURN
/
/
1172 3332
1173 7403
1174 1200
1175 0252
1176 0600
1177 0066
1200 PAGE
1200 1200
/
/
1200 4552 RANSEC, RANDOM /GET A RANDOM NUMBER
1201 0377 AND (77 /MASK TO SIX BITS
1202 3140 DCA SECSAV /SAVE AS POSSIBLE STARTING SECTOR
1203 1064 TAD MINSEC /GET MINIMUM SECTOR TO USE
1204 7041 CIA
1205 1140 TAD SECSAV /ADD POSSIBLE STARTING SECTOR
1206 7710 SPA CLA /SKIP IF STARTING SECTOR > OR = MINIMUM
1207 5200 JMP RANSEC /GO GET NEW RANDOM SECTOR
1210 7340 CLA CLL CMA /AC=-1
1211 1126 TAD NUMSEC /ADD NUMBER OF SECTORS
1212 1140 TAD SECSAV /ADD STARTING SECTOR - NOW HAVE LAST SECTOR
1213 7041 CIA /NEGATE
1214 1107 TAD MAXSEC /ADD MAXIMUM SECTOR TO USE
1215 7710 SPA CLA /SKIP IF LAST SECTOR < OR = MAXIMUM SECTOR
1216 5200 JMP RANSEC /GO GET NEW RANDOM STARTING SECTOR
1217 1024 TAD CURDRV /GET CURRENT DRIVE NUMBER AS INDEX
1220 1376 TAD (BDSECO /ADD ADDRESS OF BAD SECTOR TABLE
1221 3124 DCA TEMP1 /SAVE AS POINTER TO BAD SECTOR FILE
1222 1524 TAD I TEMP1 /GET ADDRESS OF BAD SECTOR FILE
1223 3124 DCA TEMP1 /SET UP POINTER TO BAD SECTOR FILE
1224 6211 CDF 10 /CHANGE TO DATA FIELD 1
1225 1524 TAD I TEMP1 /GET FIRST BAD SECTOR TABLE WORD
1226 6201 CDF 00 /CHANGE TO DATA FIELD 0
1227 7001 IAC /ADD ONE
1230 7650 SNA CLA /SKIP IF BAD SECTORS THIS DRIVE
1231 5301 JMP CLRBUF /GO CLEAR DATA BUFFER
/
/
1232 1375 TAD (-20
1233 3120 DCA CNTR1 /SET UP COUNTER TO CHK 16 BAD SECTORS
1234 4547 CYLLPA, SETPNT /SET TBLPNT TO CURRENT CYL AND HD
1235 0002 CURCYL /TABLE INDEX
1236 1512 TAD I TBLPNT /GET CURRENT CYL AND HD
1237 7041 CIA /NEGATE

```



1240	6211		CDF 10	/CHANGE TO DATA FIELD 1
1241	1524		TAD I TEMP1	/ADD A BAD SECTOR CYL AND HEAD
1242	6201		CDF 00	/DATA FIELD=0
1243	7650		SNA CLA	/SKIP IF BAD CYL AND HEAD NOT = CURRENT CYL AND HEAD
1244	5252		JMP SECCHK	/GO CHECK SECTOR - THIS CYL AND HD HAS BAD SECTOR
1245	2124		ISZ TEMP1	/INCREMENT BAD SECTOR POINTER
1246	2124	CYLLPB,	ISZ TEMP1	/TO NEXT BAD CYL AND HD
1247	2120		ISZ CNTR1	/INCREMENT BAD SECTOR COUNT
1250	5234		JMP CYLLPA	/TRY NEXT BAD SECTOR CYL AND HD
1251	5301		JMP CLRBUF	/NO BAD SECTORS THIS CYLINDER AND TRACK
/				
1252	2124	SECCHK,	ISZ TEMP1	/INCREMENT TO BAD SECTOR
1253	1140		TAD SECSAV	/GET STARTING SECTOR
1254	7041		CIA	/NEGATE
1255	6211		CDF 10	/DATA FIELD=1
1256	1524		TAD I TEMP1	/ADD BAD SECTOR
1257	6201		CDF 00	/DATA FIELD=0
1260	7710		SPA CLA	/SKIP IF BAD SECTOR > OR = STARTING SECTOR
1261	5246		JMP CYLLPB	/GO CHECK NEXT CYL AND HD
1262	7340		CLA CLL CMA	
1263	1126		TAD NUMSEC	/ADD NUMBER OF SECTORS
1264	1140		TAD SECSAV	/ADD STARTING SECTOR - NOW HAVE LAST SECTOR
1265	7041		CIA	/NEGATE LAST SECTOR
1266	6211		CDF 10	/DATA FIELD=1
1267	1524		TAD I TEMP1	/ADD BAD SECTOR
1270	6201		CDF 00	/DATA FLD=0
1271	7740		SNA SZA CLA	/SKIP IF LAST SECTOR > OR = BAD SECTOR
1272	5246		JMP CYLLPB	/GO CHECK NEXT CYL AND HD
/				
1273	1107	FOUNDB,	TAD MAXSEC	/FOUND BAD SECTOR - CHK FOR ONLY SECTOR
1274	7041		CIA	
1275	1064		TAD MINSEC	/ADD MINIMUM SECTOR TO -MAXIMUM SECTOR
1276	7640		SZA CLA	/SKIP IF MAXSEC=MINSEC BAD SECTOR IS ONLY SECTOR
1277	5774		JMP FIXCHK	/STARTING SECTOR PROBLEM, GET NEW WORD COUNT OR SEEK
1300	5773		JMP CYLGEN	/ONLY ONE SECTOR AND IT IS BAD, GO DO SEEK
/				
1301	1125	CLRBUF,	TAD BUFNUM	/GET BUFFER NUMBER (BUFFER FIELD)
1302	7106		CLL RTL	
1303	7004		RAL	/ROTATE TO BITS 6,7,8
1304	1123		TAD KCDF	/ADD CDF INSTRUCTION
1305	3121		DCA CDFSAV	/SET UP CDF INSTRUCTION TO BUFFER FIELD
1306	7346		CLA CLL CMA RTL	/AC= -3
1307	1125		TAD BUFNUM	/ADD BUFFER NUMBER
1310	7710		SPA CLA	/SKIP IF BUFFER > OR = 3
1311	5314		JMP CLRBF	/GO CLEAR BUFFER 0, 1 OR 2
1312	3120		DCA CNTR1	/CLEAR BUFFER COUNTER FOR 4096 LOCs TO CLEAR
1313	5772		JMP SETIDX	/GO SET UP BUFFER POINTER
1314	1125	CLRBF,	TAD BUFNUM	/GET BUFFER NUMBER 0, 1, 2
1315	7440		SZA	/SKIP IF USING BUFFER 0
1316	5324		JMP USEBF1	/USING BUFFER 1 OR 2
1317	1116		TAD BUFADO	/GET BUFFER 0 ADDRESS
1320	1055		TAD M1	/SUBTRACT 1 FOR AUTO INDEX
1321	3017		DCA AUTO17	/SET UP BUFFER POINTER FOR DATGEN ROUTINE
1322	4771		JMS CLRBF0	/CLEAR BUFFER 0

HP 002

HP 002

```

1323 5770'      JMP      CODCHK      /GO CHECK FUNCTION CODE

1324 1055      USEBF1, TAD      M1      /SUBTRACT ONE FROM BUFNUM
1325 7640      SZA CLA      /SKIP IF FIELD 1
1326 5334      JMP      USEBF2      /BUFFER FIELD 2 SELECTED
1327 1117      TAD      BUFAD1      /GET BUFFER 1 ADDRESS
1330 1055      TAD      M1      /SUB 1 FOR AUTO INDEX
1331 3017      DCA      AUTO17      /SET UP BUFFER POINTER FOR DATGEN ROUTINE
1332 4767'      JMS      CLRBF1      /CLEAR BUFFER 1
1333 5770'      JMP      CODCHK      /GO CHECK FUNCTION CODE

1334 1366      USEBF2, TAD      (BUFSZ2 /GET THE BUFFER 2 SIZE (7600)
1335 7041      CIA      /COMPLEMENT BUFFER SIZE
1336 3120      DCA      CNTR1      /COUNTER FOR 7600 LOCATIONS
1337 5772'      JMP      SETIDX      /SET UP BUFFER POINTER

/
/
/ROUTINE TO CHECK FOR "CURDRV" ACTIVE
/
/RETURN CALL+1 IF NOT ACTIVE
/RETURN CALL+2 IF ACTIVE
/
/      CALLED BY:      JMS      CHKACT

/
/
1340 0000      CHKACT, 0
1341 7300      CLA      CLL
1342 4547      SETPNT      /SET "TBLPNT" TO DRV STATE ENTRY IN TABLE
1343 0000      DSTATE      /TBL INDX
1344 1512      TAD I      TBLPNT      /GET DRIVE STATE
1345 7640      SZA      CLA      /SKIP IF DRIVE NOT ACTIVE
1346 2340      ISZ      CHKACT      /INCREMENT FOR ACTIVE RETURN
1347 5740      JMP I      CHKACT      /RETURN

/
/
/*****
/
/
1350 0000      YNGO, 0
1351 7300      CLA CLL
1352 6212      CIF      10      /INS FLD = 1
1353 4765      JMS I      (XYESNO      /GO TO YES OR NO ROUTINE FLD 1
1354 7610      SKP CLA      /YES RETURN
1355 2350      ISZ      YNGO      /NO RETURN
1356 5750      JMP I      YNGO      /EXIT

/
/
1357 0000      PRNTGO, 0
1360 6212      CIF      10      /INS FLD = 1
1361 4764      JMS I      (XPRINT      /GO TO PRINT ROUTINE FLD 1
1362 5757      JMP I      PRNTGO      /EXIT

/
/*****

```

1364 2130  
 1365 2073  
 1366 7600  
 1367 6536  
 1370 1413  
 1371 6554  
 1372 1400  
 1373 0600  
 1374 1112  
 1375 7760  
 1376 7124  
 1377 0077  
 1400

PAGE

1400 1055  
 1401 3017  
 1402 1121  
 1403 3204  
 1404 7402  
 1405 3417  
 1406 2120  
 1407 5205  
 1410 7240  
 1411 3017  
 1412 6201

SETIDX.

TAD M1  
 DCA AUTO17  
 TAD CDFSAV  
 DCA XCDFA  
 XCDFA, HLT/CDF  
 CBUF, DCA I AUTO17  
 ISZ CNTR1  
 JMP CBUF  
 STA  
 DCA AUTO17  
 CDF 00

/SUBTRACT ONE FROM BUFFER ADDRESS  
 /SET UP BUFFER POINTER TO BUFFER ADDR-1  
 /GET MODIFIED CDF TO BUFFER FIELD  
 /SAVE FOR EXECUTION  
 /MODIFIED CDF TO BUFFER FIELD  
 /CLEAR A BUFFER LOCATION  
 /INCREMENT BUFFER LOCATION COUNTER  
 /CONTINUE CLEARING BUFFER  
 /SET AUTO17 = -1  
 /DATA FIELD=0

1413 1043  
 1414 7041  
 1415 1115  
 1416 7640  
 1417 5222  
 1420 1141  
 1421 4777

CODCHK.

TAD K5  
 CIA  
 TAD FUNCOD  
 SZA CLA  
 JMP TRANS  
 TAD WRDSAV  
 JMS DATGEN

/ADD FUNCTION CODE TO -5  
 /SKIP IF DOING A WRITE CODE 5  
 /DOING A READ, DON'T GENERATE DATA IN BUFFER  
 /GET WORD COUNT  
 /GENERATE DATA FOR A WRITE

1422 4556  
 1423 4776  
 1424 4554  
 1425 5230  
 1426 4553  
 1427 5775  
 1430 4550  
 1431 0374  
 1432 7640  
 1433 5237  
 1434 4773  
 1435 4553  
 1436 5775  
 1437 4562  
 1440 4554  
 1441 5244

TRANS.

WAITDN  
 JMS FATAL  
 ERRCHK  
 JMP .+3  
 ERROR  
 JMP DRIVE  
 GETSWR  
 AND (4  
 SZA CLA  
 JMP .+4  
 JMS DATCHK  
 ERROR  
 JMP DRIVE  
 GETSTA  
 ERRCHK  
 JMP .+3

/WAIT FOR DONE FROM LAST OPERATION  
 /NO DONE FLAG  
 /CHECK FOR ERRORS LAST OPERATION  
 /NO ERROR RETURN  
 /STATUS ERROR OR CONTROLLER ERROR  
 /DRIVE DROPPED, GO GET NEW DRIVE  
 /GET SWITCH REGISTER  
 /MASK BIT 9 - INHIBIT SOFTWARE DATA CHECKING  
 /SKIP IF NOT INHIBITED  
 /DO NOT CHECK DATA, NO DCRC ERROR  
 /CHK DATA IF LAST OPR WAS A READ  
 /DATA ERROR ON READ  
 /GO GET NEW DRIVE, DRIVE DROPPED AFTER READ RETRY  
 /GET STATUS OF CURRENT DRIVE  
 /CHECK FOR ERRORS  
 /NO ERROR RETURN

1442	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
1443	5775'	JMP	DRIVE	/GO GET NEW DRIVE - DRIVE DROPPED RETURN
1444	4772'	JMS	RDYCHK	/CHECK FOR DRIVE READY
1445	5250	JMP	.+3	/DRIVE READY RETURN
1446	4553	ERROR		/DRIVE NOT READY
1447	5775'	JMP	DRIVE	/GO GET NEW DRIVE - DRIVE DROPPED RETURN
/				
1450	4547	SETPNT		/SET TBLPNT TO WORD COUNT
1451	0030	WRDCNT		/TBL INDX
1452	1141	TAD	WRDSAV	/GET WORD COUNT
1453	3512	DCA I	TBLPNT	/SAVE WORD COUNT IN DRV TBL
1454	4547	SETPNT		/SET TBLPNT TO SECTOR ADDRESS
1455	0032	SECADD		/TBL INDX
1456	1140	TAD	SECSAV	/GET SECTOR ADDRESS
1457	3512	DCA I	TBLPNT	/SAVE INITIAL SECTOR ADDRESS IN DRIVE STATE TABLE
1460	1512	TAD I	TBLPNT	/GET INITIAL SECTOR
1461	1126	TAD	NUMSEC	/ADD NUMBER OF SECTORS
1462	4547	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR
1463	0040	XENDSC		/TBL INDX
1464	3512	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR
1465	1371	TAD	(BUFLDO	/GET START OF BUFFER TABLE ADDRESS
1466	1024	TAD	CURDRV	/INDEX USING CURRENT DRIVE NUMBER
1467	3124	DCA	TEMP1	/SAVE TABLE ADDRESS
1470	1125	TAD	BUFNUM	/GET BUFFER NUMBER
1471	3524	DCA I	TEMP1	/ASSIGN BUFFER TO DRIVE
1472	4547	SETPNT		/SET TBLPNT TO INITIAL CURRENT ADDRESS
1473	0031	INITCA		/TBL INDX
1474	7344	CLA CLL	CMA RAL	/AC=-2
1475	1125	TAD	BUFNUM	/ADD BUFFER NUMBER
1476	7700	SMA CLA		/SKIP IF BUFFER 0 OR 1
1477	5305	JMP	CASTOR	/USING 4K BUFFER CURRENT ADDRESS = 0 OF BUFFER FLD
1500	1125	TAD	BUFNUM	/GET BUFFER NUMBER ( 0 OR 1 )
1501	7640	SZA	CLA	/SKIP IF BUFFER 0
1502	1117	TAD	BUFAD1	/USING BUFFER 1, GET BUFFER ADDRESS
1503	7450	SNA		/SKIP IF USING BUFFER 1
1504	1116	TAD	BUFAD0	/GET BUFFER 0 ADDRESS
1505	3512	CASTOR, DCA I	TBLPNT	/SAVE BUFFER ADDRESS AS INITIAL CURRENT ADDRESS
1506	4560	XFER		/GO DO A WRITE OR READ ACCORDING TO FUNCOD
1507	5775'	JMP	DRIVE	/GO GET A NEW DRIVE

/\*\*\*\*\*

/

/ROUTINE TO START UP AFTER POWER FAIL  
 /IF A DRIVE WAS DOING A WRITE, THAT DRIVE'S PACK WILL BE  
 /WRITTEN AND VERIFIED BEFORE CONTINUING THE EXERCISER  
 /ALL DRIVES ARE RESET, GET STATUS, READ HEADER, BEFORE CONTINUING  
 /ALL BUFFERS ARE RELEASED  
 /DRIVE STATUS TABLES REMAIN INTACT

/

/CALLED BY: JMP PWRST

/

/LAST DRIVE USING CONTROLLER IS IN "DRV60"

/

/

1510	1025	PWRST,	TAD	DRV60	/GET LAST DRIVE USING CONTROLLER
1511	3024		DCA	CURDRV	/SAVE AS CURRENT DRIVE
1512	4770		JMS	DELAY	/DELAY FOR 40 SECONDS SO DRIVES CAN SPIN-UP
1513	7730		-50		/40 SECOND TIMER
1514	4547		SETPNT		/SET TBLPNT TO COMMAND B
1515	0034		XCOMB		/TBL INDX
1516	1512		TAD I	TBLPNT	/GET LAST COMMAND B
1517	0044		AND	K7	/MASK FUNCTION CODE
1520	7041		CIA		/NEGATE
1521	1043		TAD	K5	/ADD FUNCTION CODE 5 (WRITE)
1522	7650		SNA CLA		/SKIP IF LAST OPERATION NOT WRITE
1523	4574		ADDRV		/GO WRITE AND READ PACK
1524	3113		DCA	PWRFLG	/CLEAR POWER FAIL FLAG
1525	1056		TAD	M4	/4 DRIVES TO CHECK
1526	3120		DCA	CNTR1	/SET UP DRIVE COUNTER
1527	3024		DCA	CURDRV	/SET TO DRIVE 0
1530	1055		TAD	M1	
1531	3066		DCA	BUFLD0	/RELEASE DRIVE 0 BUFFER
1532	1055		TAD	M1	
1533	3067		DCA	BUFLD1	/RELEASE DRIVE 1 BUFFER
1534	1055		TAD	M1	
1535	3070		DCA	BUFLD2	/RELEASE DRIVE 2 BUFFER
1536	1055		TAD	M1	
1537	3071		DCA	BUFLD3	/RELEASE DRIVE 3 BUFFER
/					
1540	4767	RSTLP1,	JMS	CHKACT	/CHECK FOR DRIVE ACTIVE
1541	5360		JMP	NXTDA	/GO GET NEXT DRIVE
1542	4573		RESET		/RESET VOLUME CHECK AND DRIVE ERRORS
1543	4573		RESET		/RESET DRIVE ERROR
1544	4554		ERRCHK		/CHECK FOR ERRORS
1545	5350		JMP	.+3	/NO ERRORS AFTER RESET
1546	4553		ERROR		/STATUS ERROR OR CONTROLLER ERROR
1547	5360		JMP	NXTDA	/GO GET NEXT DRIVE, DRIVE DROPPED
1550	4562		GETSTA		/GET STATUS
1551	4554		ERRCHK		/CHECK FOR ERRORS
1552	5355		JMP	.+3	/NO ERRORS
1553	4553		ERROR		/STATUS ERROR AFTER GET STATUS
1554	5360		JMP	NXTDA	/GO GET NEXT DRIVE
1555	4561		SETPOS		/SEEK TO CYL 0 HD 0
1556	0000		0		/CYL 0 HD 0
1557	7000		NOP		/DRIVE DROPPED RETURN
/					
1560	2024	NXTDA,	ISZ	CURDRV	/INCREMENT DRIVE #
1561	2120		ISZ	CNTR1	/INCREMENT DRIVE COUNTER
1562	5340		JMP	RSTLP1	/CONTINUE RECOVERING FROM POWER FAIL
1563	5766		JMP	ANYDRV	/GO CHECK FOR ACTIVE DRIVES
/					
1566	0310				
1567	1340				
1570	0352				
1571	0066				
1572	4347				
1573	6000				
1574	0004				



1632	3111	DCA	BITMOD	/BIT MODE = 12 BIT
1633	1116	TAD	BUFADO	/GET ADDRESS OF BUFFER 0
1634	1055	TAD	M1	/SUBTRACT 1 FOR AUTO INDEX
1635	3017	DCA	AUTO17	/SET UP BUFFER POINTER
1636	7340	CLA CLL	CMA	/AC=-1
1637	3332	DCA	BITSW1	/SET UP BIT MODE SWITCH
1640	1123	TAD	KCDF	/GET CDF TO FLD 0
1641	3121	DCA	CDFSAV	/SAVE FOR DATA GENERATION
1642	1371	TAD	(-252	/WORD COUNT=-170(10)
1643	4772	JMS	DATGEN	/GENERATE A SECTOR WORTH OF 12 BIT DATA
1644	4561	SETPOS		/SEEK TO CYL 0 HEAD 0
1645	0000	0		/CYL 0 HD 0
1646	5322	JMP	WRTPEX	/NON-RECOVERABLE ERROR SEEKING TO CYL 0
1647	3316	DCA	NXTCA	/CLEAR NEXT CYL AND HEAD STORAGE
1650	1053	TAD	K1000	/SET MODE BIT FOR 8 BIT MODE
1651	3111	DCA	BITMOD	/SAVE FOR WRITE
1652	4334	TWRT, JMS	WRTTRK	/WRITE TRACK
1653	5322	JMP	WRTPEX	/NON-RECOVERABLE ERROR RETURN
1654	2326	ISZ	TRKCNT	/INCREMENT TRACK COUNTER
1655	7610	SKP CLA		/SKIP IF NOT DONE
1656	5321	JMP	WRTPEX-1	/DONE WRITING ALL TRACKS!
1657	2332	ISZ	BITSW1	/INCREMENT BIT SWITCH
1660	5300	JMP	NEXTK1	/NOT TIME TO SWITCH BIT MODES
1661	7344	CLA CLL	CMA RAL	
1662	3332	DCA	BITSW1	/RESET BIT MODE SWITCH TO -2
1663	1111	TAD	BITMOD	/GET BIT MODE
1664	7650	SNA CLA		/SKIP IF 8 BIT, SET TO 12 BIT
1665	1053	TAD	K1000	/SET TO 8 BIT MODE
1666	3111	DCA	BITMOD	/SAVE INVERTED BIT MODE
1667	1327	TAD	BUFLD	/GET BUFFER FLD
1670	7650	SNA CLA		/SKIP IF FLD 1, SET TO FLD 0
1671	7001	IAC		/SET BUFFER FIELD TO FLD 1
1672	3327	DCA	BUFLD	/SAVE BUFFER FIELD
1673	1024	TAD	CURDRV	/GET CURRENT DRIVE AS INDEX
1674	1373	TAD	(BUFLDO	/INDEX INTO BUFFER IN USE TABLE
1675	3330	DCA	BUFP	/SET UP BUFFER FLD TABLE POINTER
1676	1327	TAD	BUFLD	/GET BUFFER FIELD
1677	3730	DCA I -	BUFP	/SAVE BUFFER FLD IN BUFFER TBL
1700	7301	NEXTK1, CLA CLL	IAC	/AC=1
1701	0326	AND	TRKCNT	/MASK LSB OF TRACK COUNT
1702	7640	SZA CLA		/SKIP IF EVEN COUNT
1703	2316	ISZ	NXTCA	/INCREMENT CYL ADDRESS FOR NEXT SEEK
1704	1316	TAD	NXTCA	/GET NEXT SEEK CYL AND HEAD
1705	0054	AND	K2000	/MASK HEAD SELECT
1706	7650	SNA CLA		/SKIP IF HEAD 1, SET TO HEAD 0
1707	1054	TAD	K2000	/SET TO HEAD 1
1710	3331	DCA	HDTMPA	/SAVE HEAD SELECT
1711	1316	TAD	NXTCA	/GET NEXT CYL ADDR
1712	0052	AND	K777	/MASK OFF HEAD SELECT BIT
1713	1331	TAD	HDTMPA	/ADD NEW HEAD SELECT
1714	3316	DCA	NXTCA	/SAVE NEW CYL AND HEAD SELECT
1715	4561	SETPOS		/SEEK TO NEXT CYL AND/OR HEAD
1716	0000	NXTCA, 0		/CYL ADDR AND HEAD SELECT
1717	5322	JMP	WRTPEX	/NON-RECOVERABLE ERROR SEEKING TO NEXT CYL/HEAD

		/HP 005 TAD	NXTCA	/GET NEXT CYL AND HEAD	HP 005
		/HP 005 MQL		/LOAD INTO MQ FOR OPERATOR DISPLAY	HP 005
1720	5252	JMP	TWRT	/GO DO NEXT TRACK	
		/			
1721	2200	ISZ	WRTPAK	/INCREMENT FOR SUCCESSFUL RETURN	
1722	4770	WRTPEX, JMS	RLSBUF	/RELEASE THE BUFFER	
1723	1333	TAD	CONSV	/GET OLD CONSTANT DATA FLAG	
1724	3074	DCA	CONFLG	/RESTORE CONSTANT DATA FLG	
1725	5600	JMP I	WRTPAK	/RETURN	
		/			
1726	0000	TRKCNT, 0		/TRACK COUNTER	
1727	0000	BUFLD, 0		/BUFFER FIELD	
1730	0000	BUFP, 0		/BUFFER POINTER	
1731	0000	HDTMPA, 0		/HEAD TO USE	
1732	0000	BITSWI, 0		/BIT MODE SWITCH	
1733	0000	CONSV, 0		/SAVE CONSTANT DATA FLAG	
		/			
		/*****			
		/			
		/			
		/ROUTINE TO WRITE A TRACK			
		/USE BUFFER 0 FLD 0 IF 12 BIT MODE			
		/USE BUFFER 1 FLD 1 IF 8 BIT MODE			
		/BUFFER MUST CONTAIN DATA WHEN THIS ROUTINE IS CALLED			
		/BIT MODE (BITMOD) MUST BE SET UP PRIOR TO CALL			
		/			
		/			
		/CALLED BY: JMS WRTTRK			
		/RETURN CALL+1 IF DRIVE DROPPED			
		/RETURN CALL+2 IF OK			
		/			
		/			
1734	0000	WRTTRK, 0			
1735	7300	CLA CLL			
1736	1367	TAD (-50		/FORTY SECTORS PER TRACK	
1737	3766	DCA WRTCNT		/SET UP SECTOR COUNTER -40(10)	
1740	4547	SETPNT		/SET TBLPNT TO WORD COUNT	
1741	0030	WRDCNT		/TBL INDX	
1742	1111	TAD BITMOD		/GET BIT MODE	
1743	7640	SZA CLA		/SKIP IF 12 BIT MODE	
1744	1057	TAD M400		/8 BIT MODE WORD COUNT (ONE SECTOR)=-256(10)	
1745	7450	SNA		/SKIP IF 8 BIT MODE	
1746	1371	TAD (-252		/12 BIT MODE WORD COUNT (ONE SECTOR)=-170(10)	
1747	3512	DCA I TBLPNT		/SAVE WORD COUNT	
1750	4547	SETPNT		/SET TBLPNT TO SECTOR ADDRESS	
1751	0032	SECADD		/TBL INDX	
1752	3512	DCA I TBLPNT		/CLEAR SECTOR ADDRESS IN DRV STATE TBL	
1753	3140	DCA SECSAV		/CLEAR SECTOR ADDRESS SAVE AREA	
1754	5765	JMP WRT1		/GO TO NEXT PAGE	
		/			
		/			
		/*****			
		/			
		/			



```

/ROUTINE TO CALL BITS TRANSFERRED ROUTINE IN FIELD 1
/
/   CALLED BY:      JMS      UPDAT
/
/GOES TO "UPDBIT" IN FIELD 1
/
1755 0000  UPDAT, 0
1756 6212      CIF      10      /GOING TO FLD 1
1757 4764      JMS I    (UPDBIT /GO TO BITS TRANSFERRED ROUTINE IN FLD 1
1760 5755      JMP I    UPDAT    /RETURN
/
/*****
/
1764 3000
1765 2000
1766 2074
1767 7730
1770 3332
1771 7526
1772 4702
1773 0066
1774 6536
1775 6554
1776 0010
1777 6001
1777 2000
PAGE
/
/
2000 4547  WRT1,  SETPNT      /SET TBLPNT TO INITIAL CURRENT ADDRESS
2001 0031      INITCA      /TBL INDX
2002 1111      TAD          /GET BIT MODE
2003 7640      SZA CLA      /SKIP IF 12 BIT MODE
2004 1117      TAD          /8 BIT MODE USE BUFFER 1 ADDRESS
2005 7450      SNA          /SKIP IF 8 BIT MODE
2006 1116      TAD          /12 BIT MODE USE BUFFER 0 ADDRESS
2007 3512      DCA I        /SAVE BUFFER ADDRESS AS INITIAL CURRENT ADDRESS
2010 1043      TAD          K5
2011 3115      DCA          FUNCOD
2012 4562      GETSTA
2013 4554      ERRCHK
2014 5217      JMP          .+3
2015 4553      ERROR
2016 5270      JMP          WRTEXA
2017 4563      STACHK
2020 5223      JMP          .+3
2021 4553      ERROR
2022 5270      JMP          WRTEXA
/
/
/CHECK FOR BAD SECTOR
/
2023 4276  WRTLPI, JMS      BADCHK      /GO CHECK FOR SECADD=BAD SECTOR

```

2024	5241	JMP	NXTSA	/BAD SECTOR, GO DO NEXT SECTOR
2025	1140	TAD	SECSAV	/GET SECTOR ADDRESS
2026	7001	IAC		/INCREMENT SECTOR ADDRESS FOR FINAL SECTOR REG.
2027	4547	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR REGISTER
2030	0040	XENDSC		/TBL INDX
2031	3512	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR ADDRESS
2032	4560	XFER		/WRITE SECTOR
2033	4556	WAITDN		/WAIT FOR DONE INTERRUPT
2034	4777	JMS	FATAL	/NO DONE INTERRUPT TIMEOUT
2035	4554	ERRCHK		/CHECK FOR ERRORS
2036	5241	JMP	NXTSA	/NO CONTROLLER OR STATUS WORD #1 ERROR
2037	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
2040	5270	JMP	WRTEXA	/NOT RECOVERABLE ERROR EXIT

/

/SET UP FOR NEXT SECTOR

/

2041	4547	NXTSA,	SETPNT	/SET TBLPNT TO SECTOR ADDRESS
2042	0032		SECADD	/TBL INDX
2043	1140	TAD	SECSAV	/GET SECTOR ADDRESS
2044	1043	TAD	K5	/SECTOR ADDRESS + 5
2045	3140	DCA	SECSAV	/SAVE NEW SECTOR ADDRESS
2046	1140	TAD	SECSAV	/GET NEW SECTOR ADDRESS
2047	3512	DCA I	TBLPNT	/SAVE NEXT SECTOR TO WRITE
2050	2274	ISZ	WRTCNT	/INCREMENT WRITE COUNT
2051	7410	SKP		/SKIP IF NOT DONE
2052	5267	JMP	WRTEXA-1	/DONE WRITING TRACK
2053	1140	TAD	SECSAV	/GET SECTOR ADDRESS
2054	7041	CIA		/NEGATE
2055	1376	TAD	(47	/ADD HIGHEST SECTOR
2056	7700	SMA	CLA	/SKIP IF 1/3 OF TRACK DONE
2057	5223	JMP	WRTLPI	/GO DO NEXT SECTOR
2060	1512	TAD I	TBLPNT	/GET SECTOR ADDRESS
2061	0042	AND	K3	/MASK OFF HIGH ORDER BITS
2062	7001	IAC		/ADD ONE FOR NEXT FIRST SECTOR
2063	3512	DCA I	TBLPNT	/SAVE NEXT SECTOR TO WRITE
2064	1512	TAD I	TBLPNT	/GET NEW SECTOR ADDRESS
2065	3140	DCA	SECSAV	/SAVE FOR NEXT TRANSFER
2066	5223	JMP	WRTLPI	/DO NEXT SECTOR START BUFFER OVER

/

2067	2775	ISZ	WRTRK	/INCREMENT RETURN ADDRESS FOR GOOD EXIT
2070	1775	WRTEXA,	TAD	WRTRK
2071	3273		DCA	EXADA
2072	5673		JMP I	EXADA
2073	0000	EXADA,	0	/RETURN ADDRESS

/

/

2074	0000	WRTCNT,	0	/WRITE COUNT
2075	0000	SCNTB,	0	/SECTOR COUNTER

/

/\*\*\*\*\*

/

/ROUTINE TO LOOKUP A SECTOR IN BAD SECTOR FILE

/SECTOR ADDRESS IN DRIVE STATE TABLE, "CURCYL" MUST CONTAIN CYL AND HEAD

/LOCATION "CURDRV" MUST CONTAIN CORRECT DRIVE NUMBER

```

/RETURN CALL+1 IF SECTOR IS IN BAD SECTOR FILE
/RETURN CALL+2 IF SECTOR NOT IN BAD SECTOR FILE
/
/CALLED BY:      JMS      BADCHK
/
/
2076 0000      BADCHK, 0
2077 7300          CLA CLL
2100 1374          TAD      (-20          /BAD SECTOR FILE = 16 MAX
2101 3347          DCA      SCNT          /SET UP BAD SECTOR COUNTER
2102 1024          TAD      CURDRV        /GET CURRENT DRIVE AS INDEX
2103 1373          TAD      (BDSECO      /INDEX INTO BAD SECTOR FILE INDEX TABLE
2104 3350          DCA      BADPT        /SAVE INTERMEDIATE ADDRESS
2105 1750          TAD I  . BADPT        /GET ADDRESS OF BAD SECTOR FILE
2106 3350          DCA      BADPT        /SET UP BAD SECTOR FILE POINTER
2107 4547          SETPNT          /SET TBLPNT TO CURRENT CYL AND HEAD
2110 0002          CURCYL          /TBL INDX
/
2111 6211      CYLLP1, CDF      10          /CDF TO BAD SECTOR FILE FLD 1
2112 1750          TAD I  BADPT          /GET A BAD CYL AND HEAD
2113 6201          CDF      00          /CDF TO FLD 0
2114 7001          IAC          /INCREMENT TO TEST FOR NO BAD SECTORS OR END OF BAD SECTOR TABLE
2115 7450          SNA          /SKIP IF NOT END OF TABLE AND BAD SECTORS EXIST
2116 5330          JMP      OKEX        /END OF TABLE OR NO BAD SECTORS
2117 1055          TAD      M1          /RETURN TO ORIGINAL VALUE
2120 7041          CIA          /NEGATE FOR CHECK
2121 1512          TAD I  TBLPNT        /ADD CURRENT CYL AND HEAD
2122 7650          SNA      CLA          /SKIP IF BAD CYL AND HEAD NOT=CURRENT CYL AND HEAD
2123 5332          JMP      SCHK        /GO CHECK SECTOR THIS CYL AND HEAD HAS BAD SECTOR
2124 2350          ISZ      BADPT        /INCREMENT BAD SECTOR POINTER
2125 2350      CYLLP2, ISZ      BADPT        /INCREMENT BAD SECTOR POINTER
2126 2347          ISZ      SCNT        /INCREMENT BAD SECTOR COUNTER
2127 5311          JMP      CYLLP1      /GO CHECK NEXT SECTOR
2130 2276      OKEX,  ISZ      BADCHK        /INCREMENT FOR GOOD RETURN - SECTOR NOT FOUND
2131 5676          JMP I  BADCHK        /RETURN CALL+2
/
2132 2350      SCHK,  ISZ      BADPT        /INCREMENT BAD SECTOR POINTER TO BAD SECTOR
2133 4547          SETPNT          /SET TBLPNT TO SECTOR ADDRESS
2134 0032          SECADD          /TBL INDX
2135 1512          TAD I  TBLPNT        /GET CURRENT SECTOR ADDRESS
2136 7041          CIA          /NEGATE
2137 6211          CDF      10          /CDF TO BAD SECTOR FILE FLD 1
2140 1750          TAD I  BADPT        /ADD BAD SECTOR
2141 6201          CDF      00          /CDF TO PROGRAM FLD 0
2142 7650          SNA      CLA          /SKIP IF BAD SECTOR NOT=CURRENT SECTOR
2143 5676          JMP I  BADCHK        /RETURN CALL+1, BAD SECTOR FOUND
2144 4547          SETPNT          /SET TBLPNT TO CURRENT CYL AND HEAD
2145 0002          CURCYL          /TBL INDX
2146 5325          JMP      CYLLP2      /GO CHECK NEXT BAD SECTOR
/
2147 0000      SCNT,  0          /BAD SECTOR COUNTER
2150 0000      BADPT, 0          /BAD SECTOR POINTER
/
/*****
/

```

```

/
/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
/
/
2151 0000  SWRGET, 0
2152 7300      CLA CLL
2153 1021      TAD      HCW1      /GET HCW1
2154 7710      SPA CLA      /SKIP IF USING SOFTWARE SWR
2155 7614      LAS SKP      /GET HARDWARE SWR
2156 1020      TAD      PSR      /GET PSEUDO SWR
2157 5751      JMP I  SWRGET      /EXIT
/
/
/*****
/ROUTINE TO CALL DECIMAL PRINT ROUTINE IN FIELD 1
/
/      CALLED BY:      DECPRN      (FROM FLD 0)
/
/
2160 0000  PRNDEC, 0
2161 6212      CIF      10      /GOING TO FIELD 1
2162 4772      JMS I  (PRNDEC      /GO TO FLD 1 DECIMAL PRINT ROUTINE
2163 5760      JMP I  PRNDEC      /RETURN
/
2164 0000  ACPRN, 0
2165 6212      CIF      10      /INS FLD=1
2166 4771      JMS I  (ACPRNT      /GO TO PRINT AC ROUTINE FLD 1
2167 5764      JMP I  ACPRN      /RETURN
/
/
/
2171 1476
2172 2400
2173 7124
2174 7760
2175 1734
2176 0047
2177 6614
2200
PAGE
/
/
/
/*****
/ROUTINE TO READ VERIFY AN ENTIRE PACK
/EVEN CYLS HD 0 ARE 8 BIT MODE
/ODD CYLS HD 0 ARE 12 BIT MODE
/EVEN CYLS HD 1 ARE 12 BIT MODE

```

/ODD CYLS HD 1 ARE 8 BIT MODE  
 /RETURN CALL+1 IF DRIVE DROPPED  
 /RETURN CALL+2 IF SUCCESSFUL

/ CALLED BY: JMS RDBAK

2200	0000	RDBAK,	0		
2201	7300		CLA CLL		
2202	1053		TAD	K1000	/SET BIT 2, 8 BIT MODE
2203	3111		DCA	BITMOD	/SAVE BIT MODE
2204	1377		TAD	(-1777	/1023(10) TRACKS TO CHECK
2205	3260		DCA	TRKCT	/SET UP TRACK COUNTER
2206	3217		DCA	NXTCB	/CLEAR CYL AND HEAD FOR SEEK TO CYL 0 HD 0
2207	1024		TAD	CURDRV	/GET CURRENT DRV
2210	1376		TAD	(BUFLDO	/INDEX INTO BUFFER TABLE
2211	3262		DCA	BUFP1	/SET UP BUFFER TABLE POINTER
2212	7001		IAC		/USING BUFFER (FLD) 1
2213	3662		DCA I	BUFP1	/SET DRIVE USING BUFFER 1
2214	7340		CLA CLL	CMA	
2215	3263		DCA	BITSWT	/SET UP BIT MODE SWITCH

HP 006

/READ AND VERIFY A TRACK LOOP

2216	4561	TRD,	SETPOS		/SEEK
2217	0000	NXTCB,	0		/CYL AND HD TO SEEK TO
2220	5255		JMP	RDPEX	/DRIVE DROPPED SEEKING TO NEXT CYL OR HD
2221	4264		JMS	RDTRK	/READ AND VERIFY A TRACK
2222	5255		JMP	RDPEX	/DRIVE DROPPED READING A TRACK
2223	2260		ISZ	TRKCT	/INCREMENT TRACK COUNT
2224	7610		SKP	CLA	/NOT DONE
2225	5254		JMP	RDPEX-1	/DONE CHECKING EXIT
2226	2263		ISZ	BITSWT	/INCREMENT BIT MODE SWITCH, SKIP IF TIME TO CHANGE BIT MODE
2227	5236		JMP	TRKCHK	/GO CHECK FOR EVEN OR ODD TRACK
2230	7344		CLA CLL	CMA RAL	/AC=-2 TO SWITCH BIT MODE EVERY 2 TRACKS
2231	3263		DCA	BITSWT	/SET UP BIT MODE SWITCH
2232	1111		TAD	BITMOD	/GET BIT MODE
2233	7650		SNA	CLA	/SKIP IF 8 BIT MODE (BIT 2 SET)
2234	1053		TAD	K1000	/SET BIT 2 (8 BIT MODE)
2235	3111		DCA	BITMOD	/SAVE NEW BIT MODE
2236	7301	TRKCHK,	CLA CLL	IAC	/AC=1
2237	0260		AND	TRKCT	/MASK TRACK COUNT LSB
2240	7640		SZA	CLA	/SKIP IF EVEN TRACK COUNT
2241	2217		ISZ	NXTCB	/INCREMENT CYL ADDRESS FOR SEEK
2242	1217		TAD	NXTCB	/GET CYL AND HEAD FOR NEXT SEEK
2243	0054		AND	K2000	/MASK HEAD SELECT BIT
2244	7650		SNA	CLA	/SKIP IF HEAD 1
2245	1054		TAD	K2000	/SET BIT 1 FOR HEAD 1
2246	3261		DCA	HDTEMP	/SAVE HEAD SELECT
2247	1217		TAD	NXTCB	/GET CYL AND HD
2250	0052		AND	K777	/MASK OFF HD SELECT
2251	1261		TAD	HDTEMP	/GET NEW HEAD SELECT
2252	3217		DCA	NXTCB	/SAVE NEW CYL AND HD FOR SEEK
2253	5216		JMP	TRD	/GO READ NEXT TRACK

HP 001

```

2254 2200      ISZ      RDBAK      /INCREMENT FOR GOOD RETURN
2255 4775'    RDPEX, JMS      RLSBUF /RELEASE THE BUFFER
2256 4573      RESET    /MAKE SURE LAST OPERATION IS NOT READ
2257 5600      JMP I     RDBAK      /EXIT
/
2260 0000      TRKCT, 0      /TRACK COUNTER
2261 0000      HDTEMP, 0     /TEMP STORAGE OF HEAD SELECT
2262 0000      BUFP1, 0      /BUFFER TABLE POINTER
2263 0000      BITSWT, 0     /BIT MODE SWITCH
/
/*****
/
/ROUTINE TO READ A TRACK AND VERIFY DATA
/
/      CALLED BY:      JMS      RDTRK
/RETURN CALL+1 IF DRIVE DROPPED
/RETURN CALL+2 IF OK
/
2264 0000      RDTRK, 0
2265 7300      CLA CLL
2266 1374      TAD      (-5      /READ 8 SECTORS 5 TIMES (40 SECTORS=TRACK)
2267 3773'    DCA      RDCNT     /SET UP READ COUNTER
2270 4547      SETPNT    /SET TBLPNT TO WORD COUNT
2271 0030      WRDCNT     /TBL INDX
2272 1111      TAD      BITMOD   /GET BIT MODE
2273 7640      SZA      CLA      /SKIP IF 12 BIT MODE
2274 1057      TAD      M400     /8 BIT MODE, WRD CNT = 256(10)
2275 7450      SNA      /SKIP IF USING 8 BIT MODE
2276 1372      TAD      (-252    /12 BIT MODE, WRD CNT = 170(10)
2277 3512      DCA I     TBLPNT   /SAVE WORD COUNT IN DRIVE STATE TABLE
2300 4547      SETPNT    /SET TBLPNT TO SECTOR ADDRESS
2301 0032      SECADD     /TBL INDX
2302 3512      DCA I     TBLPNT   /CLEAR SECTOR ADDRESS IN DRV STATE TBL
2303 3140      DCA      SECSAV   /CLEAR SECTOR ADDRESS SAVE AREA
2304 3771'    DCA      SECTMP    /CLEAR FIRST SECTOR OF SET OF 4 STORAGE
/
/READ AND VERIFY 8 SECTORS
/
2305 7344      SECLPA, CLA CLL CMA RAL /AC=-2
2306 3770'    DCA      BUFCT     /SET UP FOR 2 SETS OF 4 SECTORS
/
2307 4547      SECLP8, SETPNT    /SET TBLPNT TO CURRENT BUFFER ADDRESS
2310 0031      INITCA   /TBL INDX
2311 1117      TAD      BUFAD1  /GET BUFFER 1 ADDRESS
2312 3512      DCA I     TBLPNT  /SAVE AS CURRENT ADDRESS
2313 4550      GETSWR    /GET SWITCH REGISTER
2314 0367      AND      (4      /TEST BIT 9 FOR SKIP SOFTWARE DATA CHECK
2315 7650      SNA CLA     /SKIP IF BIT 9 SET = 1
2316 4766'    JMS      CLRBFI    /CLEAR BUFFER 1

```

2317	1056	TAD	M4	/4 SECTORS IN BUFFER
2320	3765'	DCA	SCNTC	/SET UP SECTOR COUNTER
/				
2321	4764'	SECLPB,	JMS	/CHECK FOR BAD SECTOR
2322	5344	JMP	SECRDA	/BAD SECTOR RETURN
2323	1140	TAD	SECSAV	/GET SECTOR ADDRESS
2324	7001	IAC		/ADD 1 TO SECTOR ADDRESS FOR FINAL SECTOR REG.
2325	4547	SETPNT		/SET TBLPNT TO EXPECTED FINAL SECTOR REGISTER
2326	0040	XENDSC		/TBL INDX
2327	3512	DCA I	TBLPNT	/SAVE EXPECTED FINAL SECTOR REGISTER
2330	7327	CLA CLL	CML IAC RTL	/FUNCTION CODE=6 (READ)
2331	3115	DCA	FUNCOD	/SAVE FUNCTION CODE FOR READ
2332	4560	XFER		/READ DATA (1 SECTOR)
2333	4556	WAITDN		/WAIT FOR DONE
2334	4763'	JMS	FATAL	/FATAL ERROR - NO DONE FLAG AFTER READ
2335	4554	ERRCHK		/CHECK FOR ERRORS
2336	5344	JMP	SECRDA	/NO ERROR RETURN GO TO NEXT PAGE
2337	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
2340	5343	JMP	RDEXA	/ERROR NOT RECOVERED - DRIVE DROPPED
2341	5344	JMP	SECRDA	/GO TO NEXT PAGE
/				
2342	2264	ISZ	RDTRK	/INCREMENT FOR GOOD RETURN
2343	5664	RDEXA,	JMP I RDTRK	/RETURN
/				
/				
/				
/SET UP FOR NEXT SECTOR READ				
/				
2344	4547	SECRDA,	SETPNT	/SET TBLPNT TO SECTOR ADDRESS
2345	0032		SECADD	/TBL INDX
2346	1512	TAD I	TBLPNT	/GET SECTOR ADDRESS
2347	1043	TAD	K5	/SECTOR ADDRESS+5
2350	3512	DCA I	TBLPNT	/SAVE NEXT SECTOR ADDRESS
2351	1512	TAD I	TBLPNT	/GET NEXT SECTOR ADDRESS
2352	3140	DCA	SECSAV	/SAVE FOR NEXT TRANSFER
2353	5762'	JMP	SECW	/GO TO NEXT PAGE
/				
2362	2400			
2363	6614			
2364	2076			
2365	2506			
2366	6536			
2367	0004			
2370	2507			
2371	2511			
2372	7526			
2373	2510			
2374	7773			
2375	3332			
2376	0066			
2377	6001			
	2400	PAGE		
/				
/				
2400	4547	SECW,	SETPNT	/SET TBLPNT TO WORD COUNT

2401	0030	WRDCNT		/TBL INDX
2402	1512	TAD I	TBLPNT	/GET WORD COUNT
2403	7041	CIA		/MAKE IT POSITIVE
2404	4547	SETPNT		/SET TBLPNT TO CURRENT BUFFER ADDRESS
2405	0031	INITCA		/TBL INDX
2406	1512	TAD I	TBLPNT	/ADD SECTOR WORD COUNT TO BUFFER ADDRESS
2407	3512	DCA I	TBLPNT	/SAVE NEW BUFFER ADDRESS
2410	2306	ISZ	SCNTC	/INCREMENT SECTOR COUNTER SKIP IF 4 SECTORS DONE
2411	5777	JMP	SECLPB	/GO READ NEXT SECTOR
2412	4547	SETPNT		/SET TBLPNT TO SECTOR ADDRESS
2413	0032	SECADD		/TBL INDX
2414	1311	TAD	SECTMP	/GET FIRST SECTOR OF SET OF 4 SECTORS
2415	3512	DCA I	TBLPNT	/SAVE NEW SECTOR ADDRESS
2416	1376	TAD	(CDF 10	/GET CDF INSTRUCTION
2417	3121	DCA	CDFSAV	/SAVE CDF TO FLD 1
2420	4547	SETPNT		/SET TBLPNT TO INITIAL CURRENT ADDRESS
2421	0031	INITCA		/TBL INDX
2422	1117	TAD	BUFAD1	/GET ADDRESS OF BUFFER
2423	3512	DCA I	TBLPNT	/RESTORE CA TO START OF BUFFER
/				
/NOW CHECK DATA IN BUFFER				
/				
2424	1056	TAD	M4	/SECTOR COUNT=-4
2425	3306	DCA	SCNTC	/SET UP SECTOR COUNTER
2426	4775	SECLPD, JMS	BADCHK	/CHECK FOR BAD SECTOR
2427	5240	JMP	NXTSD	/BAD SECTOR RETURN, SKIP THIS SECTOR
2430	4550	GETSWR		/GET SWITCH REGISTER
2431	0374	AND	(4	/MASK BIT 9 (INHIBIT SOFTWARE DATA CHECK)
2432	7640	SZA CLA		/SKIP IF CHECKING DATA
2433	5240	JMP	NXTSD	/NOT DOING SOFTWARE DATA CHECK, GO DO NEXT SECTOR
2434	4312	JMS	CHKSEC	/CHECK A SECTOR OF DATA
2435	5240	JMP	+.3	/NO DATA ERROR, CONTINUE
2436	4553	ERROR		/ERROR GO RETRY
2437	5773	JMP	RDEXA	/EXIT, DRIVE DROPPED
/				
2440	4547	NXTSD, SETPNT		/SET TBLPNT TO SECTOR ADDRESS
2441	0032	SECADD		/TBL INDX
2442	1512	TAD I	TBLPNT	/GET SECTOR ADDRESS
2443	1043	TAD	K5	/SECTOR ADDRESS+5
2444	3512	DCA I	TBLPNT	/SAVE NEW SECTOR ADDRESS
2445	4547	SETPNT		/SET TBLPNT TO WORD COUNT
2446	0030	WRDCNT		/TBL INDX
2447	1512	TAD I	TBLPNT	/GET WORD COUNT
2450	7041	CIA		/MAKE IT POSITIVE
2451	4547	SETPNT		/SET TBLPNT TO CURRENT BUFFER ADDRESS
2452	0031	INITCA		/TBL INDX
2453	1512	TAD I	TBLPNT	/SET CURRENT ADDRESS TO NEXT SECTOR IN BUFFER
2454	3512	DCA I	TBLPNT	/SAVE CURRENT ADDRESS
/				
2455	2306	ISZ	SCNTC	/INCREMENT SECTOR COUNT
2456	5226	JMP	SECLPD	/GO CHECK NEXT SECTOR
2457	2307	ISZ	BUFCT	/INCREMENT SETS OF 4 SECTORS COUNTER, SKIP IF 2 SETS
2460	7610	SKP CLA		/NOT 2 SETS OF 4
2461	5271	JMP	NXT8	
2462	4547	SETPNT		/SET TBLPNT TO SECTOR ADDRESS



```

2463 0032      SECADD
2464 1140      TAD      SECSAV      /GET NEXT FIRST SECTOR
2465 3512      DCA I    TBLPNT      /SET UP NEXT SECTOR ADDRESS
2466 1140      TAD      SECSAV      /GET NEXT FIRST SECTOR
2467 3311      DCA      SECTMP      /REMEMBER FOR CHKSEC ROUTINE
2470 5772'     JMP      SECLP8      /GO CLEAR BUFFER
2471 4547     NXT8,  SETPNT      /SET TBLPNT TO SECTOR ADDRESS
2472 0032      SECADD      /TBL INDX
2473 1512      TAD I    TBLPNT      /GET SECTOR ADDRESS
2474 0044      AND      K7          /MASK SECTOR ADDRESS LSB'S
2475 7001      IAC          /SECTOR ADDRESS+1
2476 3512      DCA I    TBLPNT      /SAVE NEW STARTING SECTOR FOR NEXT READ
2477 1512      TAD I    TBLPNT      /GET NEW STARTING SECTOR
2500 3140      DCA      SECSAV      /SAVE FOR NEXT TRANSFER
2501 1140      TAD      SECSAV      /GET NEW STARTING SECTOR
2502 3311      DCA      SECTMP      /REMEMBER FOR CHKSEC ROUTINE
2503 2310      ISZ      RDCNT      /INCREMENT READ COUNTER, SKIP IF TRACK DONE
2504 5771'     JMP      SECLPA      /GO READ NEXT 8 SECTOR SET
2505 5770'     JMP      RDEXA-1     /TRACK CHECKED GO TO EXIT

/
/
2506 0000      SCNTC, 0          /SECTOR COUNTER
2507 0000      BUFCT, 0          /BUFFER COUNTER
2510 0000      RDCNT, 0          /READ COUNTER
2511 0000      SECTMP, 0         /STORAGE OF FIRST SECTOR OF 4 SECTOR SET

/
/
/*****
/
/ROUTINE TO CHECK A SECTOR OF DATA
/FIRST WORD OF SECTOR GIVES NUMBER OF NON-ZERO FILL WORDS
/SECOND WORD OF SECTOR GIVES PATTERN USED
/
/      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
/AUTO14 USED FOR DATA PATTERN TABLES (GOOD DATA)
/AUTO15 USED FOR DATA BUFFER (DATA READ TO BE CHECKED)
/ENTER WITH BUFFER FIELD CDF IN LOCATION "CDFSAV"
/LOCATION "BITMOD" MUST CONTAIN BIT MODE IN BIT 2 (1000=8 BIT MODE, 0=12 BIT MODE)
/LOCATION "CURDRV" MUST CONTAIN DRIVE NUMBER
/
/
2512 0000      CHKSEC, 0
2513 7300      CLA CLL
2514 1121      TAD      CDFSAV      /GET CDF TO BUFFER FLD
2515 3767'     DCA      LPDF        /SAVE CDF FOR 8 BIT CHECK
2516 1121      TAD      CDFSAV      /GET CDF TO BUFFER FLD
2517 3766'     DCA      LPDF1       /SAVE CDF FOR 12 BIT CHECK
2520 1121      TAD      CDFSAV      /GET CDF TO BUFFER FLD
2521 3765'     DCA      ZCHKA       /SAVE CDF FOR ZERO FILL CHECK

```

2522	1121	TAD	CDFSAV	/GET CDF TO BUFFER FLD
2523	3764'	DCA	ZCHKB	/SAVE CDF FOR ZERO FILL CHECK
2524	1121	TAD	CDFSAV	/GET CDF TO BUFFER FLD
2525	3763'	DCA	CHK8	/SAVE CDF FOR BUFFER POINTER
2526	1121	TAD	CDFSAV	/GET CDF TO BUFFER FLD
2527	3762'	DCA	CHK12	/SAVE CDF FOR BUFFER POINTER
2530	4547	SETPNT		/SET TBLPNT TO BUFFER ADDRESS OF SECTOR DATA
2531	0031	INITCA		/TBL INDX
2532	1512	TAD I	TBLPNT	/GET BUFFER ADDRESS OF SECTOR TO CHECK
2533	1055	TAD	M1	/SUBTRACT 1 FOR AUTO INDX
2534	3015	DCA	AUTO15	/SET UP BUFFER POINTER
2535	4547	SETPNT		/SET TBLPNT TO WORD COUNT
2536	0030	WRDCNT		/TBL INDX
2537	7305	CLA CLL	IAC RAL	/AC=2
2540	1512	TAD I	TBLPNT	/-WRDCNT +2 FOR FIRST 2 WORDS
2541	7500	SMA		/SKIP IF GREATER THAN 2 WORD COUNT
2542	5350	JMP	CHKEX	/EXIT, WORD COUNT TOO SMALL
2543	3761'	DCA	WDCTR	/SET UP PARTIAL SECTOR WORD COUNTER
2544	3134	DCA	DERFLG	/CLEAR DATA ERROR FLAG
2545	3133	DCA	BDCNT	/CLEAR BAD DATA WORD COUNTER
2546	5760'	JMP	CHKB	/GO TO NEXT PAGE
/				
2547	2312	ISZ	CHKSEC	/INCREMENT FOR DATA ERROR RETURN
2550	6201	CHKEX,	CDF	/DF=0
2551	7300		CLA	
2552	5712		JMP I	/RETURN
/				
/				
2560	2600			
2561	3033			
2562	2704			
2563	2603			
2564	3035			
2565	2667			
2566	3012			
2567	2645			
2570	2342			
2571	2305			
2572	2307			
2573	2343			
2574	0004			
2575	2076			
2576	6211			
2577	2321			
	2600			
PAGE				
/				
/				
2600	1111	CHK8,	TAD	BITMOD
2601	7650		SNA CLA	/GET BIT MODE
2602	5304		JMP	CHK12
/				
2603	7402	CHK8,	HLT/CDF	/MODIFIED CDF TO BUFFER FLD
2604	1415		TAD I	AUTO15
2605	0050		AND	K377
2606	3144		DCA	WDCTMP
/SAVE TEMPORARILY				

2607	1144	TAD	WDCTMP	/GET NUMBER NON-ZERO FILL WRDS-1ST 2 WRDS
2610	7041	CIA		/NEGATE FOR USE AS COUNTER
2611	3041	DCA	WDCNTR	/SAVE WRD CNT TO CHK
2612	1415	TAD I	AUTO15	/GET DATA PATTERN NUMBER USED
2613	6201	CDF	00	/CDF TO PROGRAM FIELD
2614	0050	AND	K377	/MASK OFF GARBAGE BITS
2615	3136	DCA	PATMP	/SAVE DATA PATTERN
2616	1136	TAD	PATMP	/GET DATA PATTERN NUMBER
2617	7041	CIA		/NEGATE
2620	1042	TAD	K3	/ADD MAXIMUM PATTERN NUMBER
2621	7710	SPA CLA		/SKIP IF VALID PATTERN
2622	5777	JMP	PATER	/INVALID PATTERN, DATA ERROR, CANNOT CHK SECTOR
2623	1144	TAD	WDCTMP	/GET NUMBER NON-ZERO FILL WRDS
2624	7041	CIA		/NEGATE
2625	1376	TAD	(376	/SUBTRACT FROM 254(10) (ONE SECTOR OF DATA)
2626	7041	CIA		/NEGATE DIFFERENCE=ZERO FILL COUNT
2627	3144	DCA	WDCTMP	/SET UP ZERO FILL COUNTER
2630	1136	TAD	PATMP	/GET PATTERN NUMBER
2631	1375	TAD	(DPAT8	/INDEX INTO PATTERN POINTER TBL
2632	3136	DCA	PATMP	/SAVE POINTER
2633	1536	LPA, TAD I	PATMP	/GET ADDRESS OF PATTERN FROM TABLE
2634	1055	TAD	M1	/SUBTRACT 1 FOR AUTO INDX
2635	3014	DCA	AUTO14	/SET UP DATA PATTERN POINTER
2636	1374	TAD	(-40	/32 WRDS IN PATTERN
2637	3773	DCA	PCNTR	/SET UP PATTERN WRD CNTR
/				
2640	1105	LPA1, TAD	DATERN	/GET NUMBER OF DATA ERROR WORDS TO REPORT
2641	7041	CIA		/NEGATE
2642	1133	TAD	BDCNT	/ADD NUMBER OF BAD WORDS SO FAR
2643	7650	SNA	CLA	/SKIP IF LIMIT NOT REACHED
2644	5334	JMP	CHKRET	/DATA ERROR, GO TO EXIT
2645	7402	LPDF, HLT/CDF		/CDF TO BUFFER FLD
2646	1415	TAD I	AUTO15	/GET BUFFER WRD
2647	0050	AND	K377	/MASK OFF GARBAGE BITS
2650	6211	CDF	10	/CDF FLD 1
2651	7041	CIA		/NEGATE BUFFER WRD
2652	1414	TAD I	AUTO14	/ADD GOOD WRD
2653	6201	CDF	00	/CDF TO FLD 0 PRGM FLD
2654	7640	SZA CLA		/SKIP IF GOOD WORD=BUFFER WORD READ
2655	4772	JMS	DERR	/SECTOR DATA ERROR
2656	2771	ISZ	WDCTR	/INCREMENT PARTIAL SECTOR WORD COUNTER
2657	7610	SKP	CLA	/SKIP IF NOT END OF PARTIAL SECTOR
2660	5334	JMP	CHKRET	/PARTIAL SECTOR COMPLETELY CHECKED, EXIT
2661	2041	ISZ	WDCNTR	/INCREMENT SECTOR WRD CNTR-SKIP IF DONE
2662	7610	SKP CLA		/SKIP IF NOT DONE CHECKING SECTOR NON-ZERO
2663	5267	JMP	ZCHKA	/DONE CHECKING SECTOR NON-ZERO - GO CHECK ZERO FILL
2664	2773	ISZ	PCNTR	/INCREMENT PATTERN CNTR
2665	5240	JMP	LPA1	/CONTINUE WITH SECTOR AND PATTERN
2666	5233	JMP	LPA	/SET UP TO START PATTERN OVER
/				
/CHECK ZERO FILL WRDS				
/				
2667	7402	ZCHKA, HLT/CDF		/CDF TO BUFFER FLD
2670	1144	TAD	WDCTMP	/GET ZERO FILL WORD COUNT
2671	7650	SNA	CLA	/SKIP IF NOT ZERO

2672	5334	JMP	CHKRET	/NO ZERO FILL WORDS
2673	1415	TAD I	AUTO15	/GET A DATA WRD
2674	6201	CDF	00	/CDF TO PRGM FLD
2675	0050	AND	K377	/MASK TO 8 BITS (DELETE GARBAGE)
2676	7640	SZA CLA		/SKIP IF ZERO FILL WRD IS ZERO
2677	4772	JMS	DERR	/ZERO FILL DATA ERROR
2700	2144	ISZ	WDCTMP	/INCREMENT ZERO FILL WRD CNTR
2701	7610	SKP CLA		/SKIP IF NOT DONE CHECKING
2702	5334	JMP	CHKRET	/DONE CHECKING SECTOR - EXIT
2703	5267	JMP	ZCHKA	/CONTINUE CHECKING ZERO FILL
/				
2704	7402	CHK12,	HLT/CDF	/CDF TO BUFFER FLD
2705	1415	TAD I	AUTO15	/GET NUMBER OF NON-ZERO FILL WRDS
2706	7041	CIA		/NEGATE
2707	3041	DCA	WDCNTR	/SAVE TEMPORARILY
2710	1041	TAD	WDCNTR	/GET NUMBER NON-ZERO FILL WRDS
2711	1370	TAD	(250	/SUBTRACT FROM MAXIMUM
2712	7710	SPA CLA		/SKIP IF VALID NUMBER NON-ZERO FILL WRDS
2713	5777	JMP	PATER	/DATA ERROR - INVALID NON-ZERO FILL WRDS COUNT
2714	1415	TAD I	AUTO15	/GET DATA PATTERN NUMBER FROM BUFFER
2715	6201	CDF	00	/CDF TO PRGM FLD
2716	3136	DCA	PATMP	/SAVE DATA PATTERN NUMBER
2717	1136	TAD	PATMP	/GET DATA PATTERN NUMBER
2720	7041	CIA		/NEGATE
2721	1042	TAD	K3	/ADD MAXIMUM
2722	7710	SPA CLA		/SKIP IF VALID DATA PATTERN NUMBER
2723	5777	JMP	PATER	/DATA ERROR - INVALID DATA PATTERN NUMBER
2724	1041	TAD	WDCNTR	/GET NON-ZERO FILL WORDS (DATA)
2725	1370	TAD	(250	/SUBTRACT FROM MAXIMUM
2726	7041	CIA		/MAKE NEGATIVE
2727	3144	DCA	WDCTMP	/SET UP ZERO FILL WRD CNTR
2730	1136	TAD	PATMP	/GET PATTERN NUMBER
2731	1367	TAD	(DPAT12	/INDEX INTO PATTERN ADDRESS TBL
2732	3136	DCA	PATMP	/SAVE POINTER TO PATTERN ADDRESS TBL
2733	5766	JMP	LPB	/GO TO NEXT PAGE
/				
2734	1134	CHKRET,	TAD	DERFLG
2735	7640	SZA	CLA	/SKIP IF NO DATA ERRORS
2736	5765	JMP	CHKEX-1	/DATA ERROR EXIT
2737	5764	JMP	CHKEX	/NO DATA ERROR EXIT

\*\*\*\*\*

/APT ERROR HANDLER

/ CALLED BY: APTERR

```

2740 0000  ERRAPT, 0
2741 6002      IOF                      /INTERRUPT OFF
2742 7340      CLA CLL CMA
2743 1340      TAD  ERRAPT              /GET ERROR PC
2744 6272      CIF.  70                 /GOING TO APT FROM FLD 7
2745 5746      JMP I  .+1               /GOOD BYE!
2746 6520                      /FROM ADDRESS IN APT INTERFACE

```

```

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

```

```

/ROUTINE TO PRINT A MESSAGE

```

```

/      CALLED BY:      MESAG
/      FOLLOWED BY:    MESNAME          /ADDRESS OF MESSAGE
/

```

```

/CALLS ROUTINE MESPRT IN FLD 1
/

```

```

2747 0000  MESGO, 0
2750 7300      CLA CLL
2751 1747      TAD I  MESGO              /GET MESSAGE ADDRESS
2752 3355      DCA  MESSAD              /SAVE FOR FLD 1 CALL
2753 6212      CIF  10                 /INSTRUCTION FLD = 1
2754 4763      JMS I  (MESPRT)          /GO TO FLD 1 MESSAGE PRINTER
2755 0000      MESSAD, 0                /MESSAGE ADDRESS STORAGE
2756 2347      ISZ  MESGO               /INCREMENT RETURN ADDRESS
2757 5747      JMP I  MESGO             /RETURN

```

```

/
/
/
/*****
/
/

```

```

2763 2000
2764 2550
2765 2547
2766 3000
2767 7134
2770 0250
2771 3033
2772 6424
2773 3034
2774 7740
2775 7130
2776 0376
2777 3050
3000

```

```

PAGE
/
/

```

Address	Hex	Label	Op1	Op2	Description
3000	1536	LPB.	TAD I	PATMP	/GET ADDRESS OF DATA PATTERN FROM TABLE
3001	1055		TAD	M1	/SUBTRACT 1 FOR AUTO INDX
3002	3014		DCA	AUTO14	/SET UP DATA PATTERN POINTER (GOOD DATA)
3003	1377		TAD	(-26	/22(10) WRDS PER 12 BIT PATTERN
3004	3234		DCA	PCNTR	/SET UP PATTERN COUNTER
3005	1105	LPB1.	TAD	DATERN	/GET NUMBER OF BAD DATA WORDS TO REPORT
3006	7041		CIA		/NEGATE FOR CHECK
3007	1133		TAD	BDCNT	/ADD NUMBER OF BAD WORDS SO FOR
3010	7650		SNA	CLA	/SKIP IF LIMIT NOT REACHED
3011	5776		JMP	CHKRET	/NUMBER OF BAD WORDS LIMIT REACHED, EXIT
3012	7402	LPDF1.	HLT/	CDF	/CDF TO BUFFER FLD
3013	1415		TAD I	AUTO15	/GET DATA WRD
3014	6211		CDF	10	/CDF TO DATA PATTERN FLD 1
3015	7041		CIA		/NEGATE DATA WRD
3016	1414		TAD I	AUTO14	/ADD DATA PATTERN WRD
3017	6201		CDF	00	/CDF TO FLD 0
3020	7640		SZA	CLA	/SKIP IF DATA WRD = DATA PAT WRD
3021	4775		JMS	DERR	/DATA ERROR - GO REPORT
3022	2233		ISZ	WDCTR	/INCREMENT PARTIAL SECTOR WORD COUNTER
3023	7610		SKP	CLA	/SKIP IF NOT END OF PARTIAL SECTOR
3024	5776		JMP	CHKRET	/DONE CHECKING PARTIAL SECTOR
3025	2041		ISZ	WDCNTR	/INCREMENT NON-ZERO FILL WRD CNTR
3026	7610		SKP	CLA	/SKIP IF NOT DONE CHECKING
3027	5235		JMP	ZCHKB	/GO CHECK ZERO FILL WRDS
3030	2234		ISZ	PCNTR	/INCREMENT PATTERN COUNTER
3031	5205		JMP	LPB1	/GO CHECK NEXT WRD
3032	5200		JMP	LPB	/START PATTERN OYER
3033	0000	WDCTR.	0		
3034	0000	PCNTR.	0		
		/			
		/			
		/			
		/CHECK ZERO FILL WRDS			
		/			
3035	7402	ZCHKB.	HLT/	CDF	/CDF TO BUFFER FLD
3036	1144		TAD	WDCTMP	/GET ZERO FILL WORD COUNT
3037	7650		SNA	CLA	/SKIP IF NOT ZERO
3040	5776		JMP	CHKRET	/DONE, NO ZERO FILL, SECTOR FULL
3041	1415		TAD I	AUTO15	/GET DATA WRD TO CHK
3042	6201		CDF	00	/CDF TO FLD 0
3043	7640		SZA	CLA	/SKIP IF ZERO FILL WRD = 0
3044	4775		JMS	DERR	/DATA ERROR IN ZERO FILL
3045	2144		ISZ	WDCTMP	/INCREMENT ZERO FILL WRD CNTR
3046	5235		JMP	ZCHKB	/CONTINUE CHECKING ZERO FILL
3047	5776		JMP	CHKRET	/ZERO FILL OK - EXIT
		/			
		/			
3050	7340	PATER.	CLA	CLL CMA	/SET DATA ERROR FLAG
3051	3134		DCA	DERFLG	/CHECK FOR ON APT
3052	4545		APTCHK		/GO TO DATA ERROR RETURN
3053	5776		JMP	CHKRET	

```

3054 4550      GETSWR      /GET SWITCHES
3055 0046      AND        K100    /CHECK SWITCH 5 FOR INHIBIT ERROR TYPEOUT
3056 7640      SZA        CLA      /SKIP IF ERROR MSG OK
3057 5776'     JMP        CHKRET  /GO TO DATA ERROR RETURN
3060 4547      SETPNT     /SET TBLPNT TO INITIAL CURRENT ADDRESS
3061 0031      INITCA     /TBL INDX
3062 1512      TAD I      TBLPNT  /GET INITIAL BUFFER ADDRESS
3063 1055      TAD        M1      /SUB 1 FOR AUTO INDX
3064 3015      DCA        AUTO15  /SET UP BUFFER POINTER
3065 4547      SETPNT     /SET TBLPNT TO WORD COUNT
3066 0030      WRDCNT     /TBL INDX
3067 1512      TAD I      TBLPNT  /GET WORD COUNT
3070 3233      DCA        WDCTR   /SET UP SECTOR WORD COUNT FOR DUMP
3071 4564      MESAG
3072 3725      PATMSG
3073 4564      MESAG
3074 4057      ERR
3075 4571      DOCRLEF
3076 4571      DOCRLEF
3077 1105      TAD        DATERN  /GET NUMBER OF BAD DATA WORDS TO REPORT
3100 7041      CIA
3101 3774'     DCA        BADCNT  /SET UP DATA WORD COUNTER
3102 1121      TAD        CDFSAV  /GET CDF TO DATA FLD BUFFER
3103 3304      DCA        PATCDF  /SAVE FOR EXECUTION
3104 7402      PATCDF,    HLT/CDF /CDF TO BUFFER FLD
3105 1415      TAD I      AUTO15  /GET BUFFER DATA WORD READ
3106 6201      CDF        00
3107 4570      PRNAC
3110 4571      DOCRLEF
3111 2774'     ISZ        BADCNT  /PRINT DATA WORD
3112 5304      JMP        PATCDF  /<CR LF>
3113 5776'     JMP        CHKRET  /INCREMENT NUMBER OF WORDS REPORTED
                                   /LOOP
                                   /DONE REPORTING, EXIT

/
/
/*****
/
/
/
/
/ROUTINE TO RESET A DRIVE
/CALLED BY:      RESET
/RETURN ONLY IF SUCCESSFUL
/
/
3114 0000      XRESET, 0
3115 6002      IOF
3116 7300      CLA        CLL     /INTERRUPTS OFF
3117 4547      SETPNT     /SET TBLPNT TO COMMAND A
3120 0033      XCOMA
3121 3512      DCA I      TBLPNT  /TABLE INDEX
3122 4547      SETPNT     /CLEAR COMMAND A
3123 0032      SECADD     /SET TBLPNT TO SECTOR ADDRESS
3124 3512      DCA I      TBLPNT  /TABLE INDEX
3125 4547      SETPNT     /CLEAR SECTOR ADDRESS
                                   /SET TBLPNT TO EXPECTED FINAL SECTOR ADDRESS

```

3126	0040	XENDSC		/TBL INDX
3127	3512	DCA I	TBLPNT	/EXPECTED FINAL SECTOR ADDRESS = 0
3130	4547	SETPNT		/SET TBLPNT TO COMMAND B
3131	0034	XCOMB		/TABLE INDEX
3132	1024	TAD	CURDRV	/GET CURRENT DRIVE
3133	7002	BSW		/MOVE DRIVE SELECT TO BITS 4,5
3134	1373	TAD	(1401	/ADD INTERRUPT ENABLE, 8 BIT MODE, CODE 1
3135	3512	DCA I	TBLPNT	/SAVE COMMAND B
3136	4555	GO		/EXECUTE COMMAND
3137	4556	WAITDN		/WAIT FOR DONE OR ERROR
3140	4772	JMS	FATAL	/DONE DID NOT SET
3141	5714	JMP I	XRESET	/EXIT - RESET WAS SUCCESSFUL



3170 1163  
 3171 0040  
 3172 6614  
 3173 1401  
 3174 3752  
 3175 6424  
 3176 2734  
 3177 7752  
 3200

PAGE

/

/

/

/\*\*\*\*\*

/

/ROUTINE TO ADD A DRIVE OR DROP A DRIVE

/

/CALLED BY: ADDR

/

/ENTERED WHEN SR0=1, OR NO DRIVES ACTIVE

/

3200	0000	XADDRV, 0	
3201	4545	APTCHK	/CHECK FOR ON APT
3202	4557	APTERR	/CANNOT ADD DRIVES - ON APT
3203	1113	TAD PWRFLG	/GET POWER FAIL FLAG
3204	7640	SZA CLA	/SKIP IF NOT POWER FAIL
3205	5777'	JMP WRTF	/POWER FAILED
3206	4564	MESAG	
3207	4356	ADD	/ADD
3210	4564	MESAG	
3211	4075	DRVMSG	/DRIVE
3212	4564	MESAG	
3213	4002	YORN	/(Y/N)?
3214	4565	YESRN	/GET YES OR NO ANSWER
3215	7610	SKP CLA	/YES RETURN
3216	5776'	JMP DRPDVA	/GO DROP DRIVE - NO RETURN
3217	4343	JMS WCHDR	/GO GET DRIVE NUMBER
3220	1055	TAD M1	
3221	3066	DCA BUFLD0	/RELEASE BUFFER FOR DRIVE 0
3222	1055	TAD M1	
3223	3067	DCA BUFLD1	/RELEASE BUFFER FOR DRIVE 1
3224	1055	TAD M1	
3225	3070	DCA BUFLD2	/RELEASE BUFFER FOR DRIVE 2
3226	1055	TAD M1	
3227	3071	DCA BUFLD3	/RELEASE BUFFER FOR DRIVE 3
3230	4564	MESAG	
3231	4513	CLRTAB	/CLEAR DRIVE STATUS TABLES
3232	4564	MESAG	
3233	4002	YORN	/(Y/N)?
3234	4565	YESRN	/GET YES OR NO ANSWER
3235	7610	SKP CLA	/YES RETURN
3236	5253	JMP SETACT	/NO RETURN
3237	1024	TBLCLR, TAD CURDRV	/GET DRV # AS INDX

```

3240 1375      TAD      (DRVPNT      /ADD ADDR OF DRV STATE TBL INDX TBL
3241 3124      DCA      TEMP1      /SET UP POINTER TO DRV STATE TABLE
3242 1524      TAD I     TEMP1      /GET DRIVE STATE TABLE ADDRESS
3243 3124      DCA      TEMP1      /SET UP POINTER INTO TABLE
3244 1374      TAD      (DRIVE1-DRIVE0-1 /GET SIZE OF A DRIVE STATE TBL
3245 7041      CIA      /NEGATE FOR COUNTER
3246 3120      DCA      CNTR1      /SAVE TABLE COUNTER
3247 3524      TBLLP, DCA I TEMP1      /CLEAR A DRIVE STATE TBL LOC
3250 2124      ISZ      TEMP1      /INCREMENT TBL POINTER
3251 2120      ISZ      CNTR1      /INCREMENT COUNTER
3252 5247      JMP      TBLLP      /CONTINUE CLEARING TABLE
/
/
3253 4547      SETACT, SETPNT      /SET TBLPNT TO DRIVE STATE
3254 0000      DSTATE      /TBL INDX
3255 7240      CLA CMA
3256 3512      DCA I     TBLPNT      /SET DRIVE ACTIVE
3257 4564      MESAG
3260 4533      RETBAD      /RETRIEVE BAD SECTOR FILE
3261 4564      MESAG
3262 4002      YORN
3263 4565      YESRN      /(Y/N)?
3264 5304      JMP      GETBAD      /YES RETURN, GO RETRIEVE BAD SECTOR FILE
3265 1024      TAD      CURDRV      /NO RETURN, GET DRV #
3266 1373      TAD      (BDSECO      /INDEX INTO BAD SECTOR INDEX TABLE
3267 3124      DCA      TEMP1      /SAVE TEMPORARILY
3270 1524      TAD I     TEMP1      /GET ADDRESS OF BAD SECTOR FILE
3271 1055      TAD      M1          /-1 FOR AUTO-INDEX
3272 3016      DCA      AUTO16      /SET BAD SECTOR FILE POINTER
3273 1372      TAD      (-40        /32(10) LOCATIONS IN BAD SECTOR FILE
3274 3331      DCA      BCNT      /SET UP COUNTER
3275 6211      CDF      10          /DF=1
3276 1055      BADRES, TAD M1          /AC=7777
3277 3416      DCA I     AUTO16      /SAVE IN BAD SECTOR FILE
3300 2331      ISZ      BCNT      /INCREMENT COUNTER
3301 5276      JMP      BADRES      /CONTINUE RESTORING TO NO BAD SECTORS
3302 6201      CDF      00          /DF=0
3303 5312      JMP      WRTASK      /NO BAD SECTORS ASSUMED, GO ASK ABOUT WRITING PACK
/
3304 4573      GETBAD, RESET      /RESET DRIVE
3305 4561      SETPOS      /DO A SEEK TO BAD SECTOR FILE
3306 2777      2777          /CYL 777 HEAD 1 (BAD SECTOR TRACK)
3307 5325      JMP      ADDEX      /NOT SUCCESSFUL RETURN - EXIT
3310 4771      JMS      GETBSF      /SUCCESSFUL RETURN - GO RETRIEVE BAD SECTOR FILE
3311 5325      JMP      ADDEX      /NOT SUCCESSFUL RETURN - EXIT
/
3312 1113      WRTASK, TAD PWRFLG      /GET POWER FAIL FLAG
3313 7640      SZA CLA      /SKIP IF NOT POWER FAIL
3314 5777      JMP      WRTP      /POWER FAIL, GO WRITE PACK
3315 4564      MESAG
3316 4545      WPAK          /WRITE PACK
3317 4564      MESAG
3320 4002      YORN
3321 4565      YESRN      /(Y/N)?
3322 5777      JMP      WRTP      /GET YES OR NO ANSWER
/YES RETURN, GO TO NEXT PAGE

```

HP 001

```

3323 4561      SETPOS      /SEEK TO CYL 0 HD 0
3324 0000      0           /DISK ADDRESS

/
3325 6002      ADDEX, 10F
3326 4546      C8CALL      /CALL CONSOLE PACKAGE
3327 7000      NOP
3330 5600      ADDEX1, JMP I  XADDRV
/
/
3331 0000      BCNT, 0
/
/
/*****8
/
/ROUTINE TO RELEASE A DATA BUFFER
/SET BUFLDX ON PAGE ZERO TO 777, (DRV NOT USING ANY BUFFER)
/
/      CALLED BY:      JMS      RLSBUF
/
/
3332 0000      RLSBUF, 0
3333 7300      CLA      CLL
3334 1370      TAD      (BUFLDQ      /GET ADDRESS OF START OF BUFFER TABLE
3335 1024      TAD      CURDRV      /ADD CURRENT DRV # AS INDEX INTO TABLE
3336 3342      DCA      RLSPT      /SET UP BUFFER TABLE POINTER
3337 7340      CLA CLL CMA      /AC=7777
3340 3742      DCA I  RLSPT      /SET BUFFER(DRV) NOT IN USE
3341 5732      JMP I  RLSBUF      /RETURN

/
3342 0000      RLSPT, 0      /BUFFER TABLE POINTER
/
/
/
3343 0000      WCHDR, 0
3344 4571      DOCR LF      /DO A CR AND LF
3345 4564      MESAG
3346 4075      DRVMSG      /DRIVE
3347 1367      TAD      (277      /GET ASCII CODE FOR "?"
3350 4566      PRNT      /?
3351 4567      GETOCT      /GET ONE OCTAL DIGIT
3352 3024      DCA      CURDRV      /SAVE AS POSSIBLE CURRENT DRIVE
3353 6211      CDF      10      /DF=1
3354 1766      TAD I  (DIGFLG      /GET DIGIT INPUT FLAG
3355 6201      CDF      00      /DF=0
3356 7650      SNA      CLA      /SKIP IF DIGIT INPUT
3357 5344      JMP      WCHDR+1      /GO TRY AGAIN FOR DRIVE NUMBER
3360 7346      CLA CLL CMA RTL      /AC=-3
3361 1024      TAD      CURDRV      /ADD CURRENT DRIVE
3362 7740      SNA SZA CLA      /SKIP IF VALID DRV #
3363 5344      JMP      WCHDR+1      /GO ASK DRV QUESTION INVALID DRV #
3364 5743      JMP I  WCHDR      /EXIT
/
/

```

3366 0060  
 3367 0277  
 3370 0066  
 3371 3440  
 3372 7740  
 3373 7124  
 3374 0041  
 3375 6710  
 3376 3424  
 3377 3400  
 3400

PAGE

/

/

3400	4561	WRTP.	SETPOS	/SET HEADS TO CYL 0 HD 0
3401	0000		0	/DISK ADDRESS
3402	5777'	JMP	ADDEX	/NOT SUCCESSFUL, EXIT
3403	4776'	JMS	WRTPAK	/GO WRITE PACK
3404	5777'	JMP	ADDEX	/UNSUCCESSFUL RETURN
3405	4564		MESAG	
3406	4075		DRVMSG	/DRV
3407	1024	TAD	CURDRV	/GET CURRENT DRIVE NUMBER
3410	1047	TAD	K260	/ADD ASCII BASE CODE
3411	4566	PRNT		/PRINT DRIVE NUMBER
3412	4564		MESAG	
3413	4552		PAKWRT	/PACK WRTN OK
3414	4775'	JMS	RDPK	/GO READ PACK
3415	5220	JMP	.+3	/UNSUCCESSFUL RETURN
3416	4564		MESAG	
3417	4562		PAKRD	/PACK READ OK
3420	1113	TAD	PWRFLG	/GET POWER FAIL FLAG
3421	7640	SZA CLA		/SKIP IF NOT POWER FAIL
3422	5774'	JMP	ADDEX1	/POWER FAIL EXIT
3423	5777'	JMP	ADDEX	/GO ASK SR=
3424	4564	DRPDVA.	MESAG	
3425	4341		DROP	/DROP
3426	4564		MESAG	
3427	4075		DRVMSG	/DRIVE
3430	4564		MESAG	
3431	4002		YORN	/((Y/N)?
3432	4565		YESRN	/GET YES OR NO ANSWER
3433	7610		SKP CLA	/YES RETURN
3434	5777'	JMP	ADDEX	/NO RETURN
3435	4773'	JMS	WCHDR	/GO GET DRIVE NUMBER
3436	4772'	JMS	KILDRV	/GO DROP THE DRIVE AND RELEASE THE BUFFER
3437	5777'	JMP	ADDEX	/EXIT

/

/\*\*\*\*\*

/

```

/
/
/ROUTINE TO RETRIEVE AND PROCESS A BAD SECTOR FILE
/FROM A PACK
/CALLED BY:      JMS      GETBSF
/RETURN CALL+1 IF DRIVE DROPPED
/RETURN CALL+2 IF OK
/
/

```

```

3440 0000      GETBSF, 0
3441 7300      CLA CLL
3442 1053      TAD      K1000      /SET BIT 2
3443 3111      DCA      BITMOD     /8 BIT MODE
3444 4547      SETPNT     /SET TBLPNT TO INITIAL CURRENT ADDR
3445 0031      INITCA     /TBL INDX
3446 1117      TAD      BUFAD1     /GET ADDR OF FLD 1 BUFFER
3447 3512      DCA I     TBLPNT     /SAVE AS INITIAL CURRENT ADDRESS
3450 1371      TAD      (-5        /FIVE BAD SECTOR FILES FOR FACTORY
3451 3770      DCA      BADCNT     /SET UP ATTEMPT COUNT TO READ BAD SECTOR FILE
3452 1367      TAD      (FACBAD    /GET ADDRESS OF TABLE OF FACTORY BAD SECTOR FILE SECTOR NUMBERS
3453 3131      DCA      BADPNT     /SET UP TABLE POINTER
3454 1366      TAD      (BUFLD0    /GET BUFFER TABLE ADDR
3455 1024      TAD      CURDRV     /INDEX USING CURRENT DRIVE
3456 3124      DCA      TEMP1      /SAVE POINTER
3457 7001      IAC
3460 3524      DCA I     TEMP1      /SET CURRENT DRIVE USING BUFFER 1

/
3461 4765      TRYLPA, JMS      CLRBF1      /CLEAR BUFFER 1
3462 4562      GETSTA      /GET STATUS OF DRIVE
3463 4554      ERRCHK     /CHECK FOR ERRORS
3464 5267      JMP      .+3      /NO ERROR RETURN
3465 4553      ERROR      /STATUS ERROR OR CONTROLLER ERROR
3466 5640      JMP I     GETBSF     /DRIVE DROPPED ERROR NOT RECOVERABLE
3467 4764      JMS      RDYCHK     /CHECK FOR DRIVE READY
3470 5273      JMP      .+3      /DRIVE READY RETURN
3471 4553      ERROR      /DRIVE NOT READY ERROR
3472 5640      JMP I     GETBSF     /DRIVE DROPPED ERROR NOT RECOVERABLE
3473 1057      TAD      M400      /WORD COUNT=256(10) (ONE SECTOR)
3474 4547      SETPNT     /SET TBLPNT TO WORD COUNT
3475 0030      WRDCNT     /TABLE INDEX
3476 3512      DCA I     TBLPNT     /SAVE WORD COUNT
3477 4547      SETPNT     /SET TBLPNT TO SECTOR ADDRESS IN DRV STATE TBL
3500 0032      SECADD     /TBL INDX
3501 1531      TAD I     BADPNT     /GET A BAD SECTOR FILE SECTOR NUMBER
3502 3512      DCA I     TBLPNT     /SAVE AS SECTOR ADDRESS
3503 1512      TAD I     TBLPNT     /GET SECTOR ADDRESS
3504 7001      IAC      /ADD ONE FOR EXPECTED FINAL SECTOR REGISTER
3505 4547      SETPNT     /SET TBLPNT TO EXPECTED FINAL SECTOR REG.
3506 0040      XENDSC     /TBL INDX
3507 3512      DCA I     TBLPNT     /SAVE EXPECTED FINAL SECTOR REG. IN DRV TBL
3510 7327      CLA CLL    CML IAC RTL /CODE 6 READ DATA
3511 3115      DCA      FUNCOD     /SAVE AS FUNCTION CODE
3512 4560      XFER      /READ BAD SECTOR FILE
3513 4556      WAITDN     /WAIT FOR DONE
3514 4763      JMS      FATAL      /NO DONE INTERRUPT

```

```

3515 4554      ERRCHK      /CHECK FOR ERRORS
3516 5337      JMP      RDFLD      /NO ERROR RETURN GO READ FIELD BAD SECTOR FILE
3517 2131      ISZ      BADPNT      /INCREMENT BAD SECTOR POINTER
3520 2770      ISZ      BADCNT      /INCREMENT READ ATTEMPT COUNT
3521 5261      JMP      TRYLPA      /TRY NEXT BAD SECTOR FILE
3522 4545      APTCHK      /CHECK FOR ON APT
3523 4557      APTERR      /ON APT - CANNOT READ FACTORY BAD SECTOR FILE
3524 4564      MESAG
3525 4255      CNTRD1      /CAN'T READ FACTORY
3526 4564      MESAG
3527 4265      BDMSEC      /BAD SECTOR FILE ON
3530 4564      MESAG
3531 4075      DRVMSG      /DRIVE
3532 1024      TAD      CURDRV      /GET DRIVE #
3533 1047      TAD      K260      /ADD ASCII BASE CODE
3534 4566      PRNT      /PRINT DRIVE #
3535 4772      JMS      KILDRV      /GO DROP THE DRIVE AND RELEASE THE BUFFER
3536 5640      JMP I      GETBSF      /RETURN

/
3537 4547      RDFLD, SETPNT      /SET TBLPNT TO INITIAL CURRENT ADDR
3540 0031      INITCA      /TBL INDX
3541 1117      TAD      BUFAD1      /GET ADDR OF BUFFER
3542 1051      TAD      K400      /ADD 256(10) TO PRESERVE FACTORY BAD SECTOR
3543 3512      DCA I      TBLPNT      /SET UP INITIAL CURRENT ADDR
3544 1371      TAD      (-5      /
3545 3770      DCA      BADCNT      /SET UP TO READ ALL 5 FIELD BAD IF NECESSARY
3546 1362      TAD      (FLDBAD      /GET ADDR OF TABLE OF FIELD BAD SECTOR ADDR
3547 3131      DCA      BADPNT      /SET UP POINTER TO TABLE
3550 5761      JMP      TRYLPB      /GO TO NEXT PAGE

/
3551 2240      ISZ      GETBSF      /INCREMENT FOR GOOD RETURN
3552 4760      BSFEX, JMS      RLSBUF      /RELEASE THE BUFFER
3553 5640      JMP I      GETBSF      /RETURN

/
3554 0000      XVT278, 0
3555 6031      KSF
3556 6030      KCF
3557 5754      JMP I      XVT278

/
/*****
/
/
3560 3332
3561 3600
3562 7171
3563 6614
3564 4347
3565 6536
3566 0066
3567 7164
3570 3752
3571 7773
3572 1163
3573 3343

```

3574 3330  
3575 2200  
3576 1600  
3577 3325  
3600

PAGE

//  
//  
//

3600 4562 TRYLPB, GETSTA  
3601 4554 ERRCHK  
3602 5205 JMP .+3  
3603 4553 ERROR  
3604 5777' JMP BSFEX  
3605 4776' JMS RDYCHK  
3606 5211 JMP .+3  
3607 4553 ERROR  
3610 5777' JMP BSFEX  
3611 4547 SETPNT  
3612 0032 SECADD  
3613 1531 TAD I BADPNT  
3614 3512 DCA I TBLPNT  
3615 1512 TAD I TBLPNT  
3616 7001 IAC  
3617 4547 SETPNT  
3620 0040 XENDSC  
3621 3512 DCA I TBLPNT  
3622 4560 XFER  
3623 4556 WAITDN  
3624 4775' JMS FATAL  
3625 4554 ERRCHK  
3626 5247 JMP BADPRO  
3627 2131 ISZ BADPNT  
3630 2352 ISZ BADCNT  
3631 5200 JMP TRYLPB  
3632 4545 APTCHK  
3633 4557 APTERR  
3634 4564 MESAG  
3635 4275 CNTRD2  
3636 4564 MESAG  
3637 4265 BDSEC  
3640 4564 MESAG  
3641 4075 DRVMSG  
3642 1024 TAD CURDRV  
3643 1047 TAD K260  
3644 4566 PRNT  
3645 4774' JMS KILDRV  
3646 5777' JMP BSFEX

//  
//  
//  
/PROCESS THE BAD SECTORS  
/

3647 7344 BADPRO, CLA CLL CMA RAL  
3650 3350 DCA BADSWT  
3651 1373 TAD (-21

/GET DRV STATUS  
/CHECK FOR ERRORS  
/NO ERROR RETURN  
/STATUS ERROR OR CONTROLLER ERROR  
/NOT RECOVERABLE ERROR  
/CHECK FOR DRIVE READY  
/DRIVE READY RETURN  
/DRIVE NOT READY  
/NOT RECOVERABLE ERROR  
/SET TBLPNT TO SECTOR ADDRESS  
/TBL INDX  
/GET BAD SECTOR FILE SECTOR  
/SAVE AS SECTOR ADDRESS  
/GET SECTOR ADDRESS  
/ADD ONE FOR FINAL EXPECTED SECTOR REG.  
/SET TBLPNT TO FINAL SECTOR REG. IN DRV TBL  
/TBL INDX  
/SAVE FINAL EXPECTED SECTOR REG.  
/READ BAD SECTOR FILE  
/WAIT FOR DONE  
/NO DONE INTERRUPT  
/CHECK FOR ERRORS  
/GO PROCESS BAD SECTORS  
/INCR TBL POINTER TO NXT BAD SECTOR FILE  
/INCREMENT BAD SECTOR COUNT  
/TRY NEXT BAD SECTOR FILE  
/CHECK FOR ON APT  
/ON APT CAN'T READ FIELD BAD SECTOR FILE  
/CAN'T READ FIELD  
/BAD SECTOR FILE ON  
/DRIVE  
/GET DRIVE #  
/ADD ASCII BASE CODE  
/PRINT DRIVE #  
/GO DROP THE DRIVE AND RELEASE THE BUFFER  
/RETURN

HP 008

/SET UP FACTORY/FIELD SWITCH  
/16 BAD SECTORS ALLOWED NOT 17

3652	3352	DCA	BADCNT	/SET UP BAD SECTOR COUNTER
3653	1372	TAD	(13	/SET UP POINTER TO TRACK OF 1ST BAD SECTOR
3654	1117	TAD	BUFAD1	/ADD ADDR OF BUFFER
3655	3347	DCA	BUFPNT	/SET UP POINTER
3656	1024	TAD	CURDRV	/GET CURRENT DRV # AS INDEX
3657	1371	TAD	(BDSECO	/ADD ADDR OF BAD SECTOR INDX TBL
3660	3124	DCA	TEMP1	/SAVE POINTER TO INDX TBL
3661	1524	TAD I	TEMP1	/GET ADDR OF BAD SECTOR FILE FROM TABLE
3662	3124	DCA	TEMP1	/SET UP BAD SECTOR TBL POINTER
3663	6211	CDF	10	/DATA FLD 1
/				
3664	1747	BADLPB, TAD I	BUFPNT	/GET BAD TRACK
3665	0050	AND	K377	/MASK OFF UNUSED BITS
3666	7041	CIA		/NEGATE
3667	1050	TAD	K377	
3670	7650	SNA CLA		/SKIP IF TRACK NOT ALL ONES
3671	5340	JMP	FLDBD	/TRACK ALL ONES, END OF BAD SECTORS
3672	2352	ISZ	BADCNT	/INCREMENT BAD SECTOR COUNT
3673	7410	SKP		/SKIP IF < 16 BAD SECTORS
3674	5324	JMP	TOOBAD	/MORE THAN 16 BAD SECTORS
3675	7301	CLA CLL	IAC	/SET BIT 11
3676	0747	AND I	BUFPNT	/MASK TRACK BIT
3677	7112	CLL RTR		
3700	7010	RAR		/MOVE TRK BIT TO BIT 1
3701	3351	DCA	BADTRK	/SAVE TRK BIT TEMPORARILY
3702	7346	CLA CLL	CMA RTL	
3703	1347	TAD	BUFPNT	/SUBTRACT 3 FROM BUFFER POINTER
3704	3347	DCA	BUFPNT	/BACK UP BUFFER POINTER TO CYL ADDR
3705	1747	TAD I	BUFPNT	/GET CYL
3706	0052	AND	K777	/MASK OFF UNUSED BITS
3707	1351	TAD	BADTRK	/ADD TRK BIT
3710	3524	DCA I	TEMP1	/SAVE BAD TRK AND CYL
3711	2124	ISZ	TEMP1	/INCREMENT BAD SECTOR STORAGE POINTER
3712	2347	ISZ	BUFPNT	/INCREMENT BUFFER POINTER
3713	2347	ISZ	BUFPNT	/INCREMENT BUFFER POINTER
3714	1747	TAD I	BUFPNT	/GET BAD SECTOR #
3715	0052	AND	K777	/MASK OFF UNUSED BITS
3716	3524	DCA I	TEMP1	/SAVE BAD SECTOR #
3717	2124	ISZ	TEMP1	/INCREMENT BAD SECTOR STORAGE POINTER
3720	1043	TAD	K5	
3721	1347	TAD	BUFPNT	/ADD 5 TO BUFFER POINTER
3722	3347	DCA	BUFPNT	/BUFFER POINTER POINTS TO NEXT BAD TRK
3723	5264	JMP	BADLPB	/CONTINUE PROCESSING BAD SECTORS
3724	6201	TOOBAD, CDF	00	/DATA FLD 0
3725	4545	APTCHK		/CHECK FOR ON APT
3726	4557	APTERR		/TOO MANY BAD SECTORS ON DRIVE
3727	4564	MESAG		
3730	4075	DRVMSG		/DRIVE
3731	1024	TAD	CURDRV	
3732	1047	TAD	K260	/ADD ASCII BASE CODE
3733	4566	PRNT		/PRINT DRIVE #
3734	4564	MESAG		
3735	4304	PAKBAD		/PACK HAS MORE THAN 16 BAD SECTORS!
3736	4774	JMS	KILDRV	/GO DROP THE DRIVE AND RELEASE THE BUFFER
3737	5777	JMP	BSFEX	/RETURN

HP 001

HP 001



```

/
3740 1117  FLDBD,  TAD      BUFAD1      /GET BUFFER ADDR
3741 1370          TAD      (413        /ADD 267(10) TO POINT TO FLD BAD SECTORS
3742 3347          DCA      BUFPNT      /SET UP BUFFER POINTER TO FIRST FLD BAD
3743 2350          ISZ      BADSWT      /INCREMENT FLD BAD SECTOR LOOP SWITCH
3744 5264          JMP      BADLPB      /GO PROCESS FLD BAD SECTORS
3745 6201          CDF      00          /DATA FLD 0
3746 5767'      JMP      BSFEX-1      /BAD SECTORS COMPLETE

/
3747 0000  BUFPNT, 0      /BUFFER POINTER
3750 0000  BADSWT, 0      /FACTORY/FIELD SWITCH
3751 0000  BADTRK, 0      /BAD TRACK TEMP STORAGE
3752 0000  BADCNT, 0      /BAD SECTOR COUNTER

/
/*****
/
/
/ROUTINE TO CALL SEEK UPDATE ROUTINE "UPDSEK" IN FLD 1
/
/      CALLED BY:      JMS      SEEKUP
/
/
3753 0000  SEEKUP, 0
3754 6212          CIF      10
3755 4757          JMS I    SKUPAD
3756 5753          JMP I    SEEKUP      /RETURN

/
3757 3200  SKUPAD, UPDSEK      /ADDRESS OF ROUTINE IN FLD 1

/
/
/*****
/
/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
/
/
3760 0000  CHKRDY, 0
3761 7300          CLA      CLL
3762 1030          TAD      ERREG      /GET ERROR REGISTER AT TIME OF ERROR
3763 7110          CLL      RAR      /MOVE INTO LINK BIT 11 (DRIVE READY)
3764 7630          SZL      CLA      /SKIP IF DRIVE WAS NOT READY
3765 2360          ISZ      CHKRDY     /DRIVE WAS READY, INCREMENT RETURN
3766 5760          JMP I    CHKRDY     /RETURN

/
/
/*****
/

```

3767 3551  
 3770 0413  
 3771 7124  
 3772 0013  
 3773 7757  
 3774 1163  
 3775 6614  
 3776 4347  
 3777 3552  
 4000

PAGE

/

/

/

/\*\*\*\*\*

/ROUTINE TO SERVICE INTERRUPTS

4000	6003	INTSVC, SRO	/SKIP IF INTERRUPT REQUEST
4001	5235	JMP NOTINT	/NO INTERRUPT REQUEST - GO TO ERROR ROUTINE
4002	3217	DCA ACSAVE	/SAVE AC
4003	6004	GTF	/GET THE FLAGS
4004	3220	DCA FLAGS	/SAVE THE FLAGS
4005	6102	SPL	/SKIP ON POWER LOW
4006	5242	JMP FLGS	/POWER OK, GO CHECK FLAGS
4007	6103	CAL	/CLEAR AC LOW FLAG
4010	1000	TAD LOCO	/GET INTERRUPT PC
4011	3216	DCA PCSAVE	/SAVE INTERRUPT PC
4012	1215	TAD STRTIN	/GET RESTART INSTRUCTION
4013	3000	DCA LOCO	/SAVE FOR EXECUTION
4014	7402	HLT	/STOP EVERYTHING - POWER GOING DOWN
/			
4015	5403	STRTIN, JMP I ADDR3	/RESTART INSTRUCTION
4016	7402	PCSAVE, HLT	/INTERRUPT PC
4017	7402	ACSAVE, HLT	/INTERRUPT AC
4020	7402	FLAGS, HLT	/FLAG SAVE AREA
4021	7402	SETTMP, HLT	/STORAGE OF SETPNT RETURN ADDRESS
4022	4545	RESTR, APTCHK	/CHECK ON APT
4023	4557	APTERR	/GO TO APT PROM
4024	4777	JMS DELAY	/DELAY FOR 12 SECONDS OR SO FOR SPIN-UP
4025	7766	-12	/AFTER POWER FAIL
4026	4564	MESAG	
4027	4502	PWRFAIL	/POWER FAILED AT PC
4030	1216	TAD PCSAVE	/GET POWER FAIL PC
4031	4570	PRNAC	/PRINT PC
4032	7240	CLA CMA	
4033	3113	DCA PWRFLG	/SET POWER FAIL RESTART FLAG
4034	5776	JMP START	/START PROGRAM OVER
/			
4035	4545	NOTINT, APTCHK	/CHECK FOR ON APT
4036	4557	APTERR	/ON APT - INVALID INTERRUPT ERROR
4037	4564	MESAG	
4040	4525	NOINT	/NO INTERRUPT REQUEST

HP 009  
 HP 009

4041	7402	HLT		/FATAL ERROR - INTERRUPT SERVICE WAS /ENTERED BUT NO INTERRUPT REQUEST WAS /ACTIVE, TRY RELOADING PROGRAM
/				
4042	6601	FLGS,	RLSD	/SKIP ON FUNCTION DONE FLAG SET
4043	7410	SKP		/FUNCTION DONE NOT SET
4044	7305	CLA CLL IAC RAL		/SET BIT 10 FUNCTION DONE FLAG SET
/				
4045	6617	IOTA,	RLSE	/SKIP ON COMPOSITE ERROR FLAG SET
4046	7410	SKP		/COMPOSITE ERROR FLAG NOT SET
4047	7001	IAC		/SET BIT 11 - COMPOSITE ERROR FLAG SET
/				
4050	3130	DCA	FLGSAV	/SAVE STATUS OF CONTROLLER FLAGS
4051	1130	TAD	FLGSAV	/GET FLAGS
4052	7650	SNA CLA		/SKIP IF EITHER DONE OR ERROR SET
4053	5342	JMP	KBDCHK	/NOT DONE OR ERROR - GO CHECK KEYBOARD
4054	6610	IOTB,	RRER	/READ ERROR REGISTER
4055	7410	SKP		/IOT SHOULD NOT SKIP
4056	4775	JMS	FATAL	/IOT RRER SKIPPED
4057	3030	DCA	ERREG	/SAVE ERROR REGISTER
/				
4060	6611	IOTC,	RRWC	/READ FINAL WORD COUNT
4061	7410	SKP		/IOT SHOULD NOT SKIP
4062	4775	JMS	FATAL	/IOT RRWC SKIPPED
4063	3034	DCA	ENDWD	/SAVE FINAL WORD COUNT
/				
4064	6612	IOTD,	RRCA	/READ COMMAND A
4065	7410	SKP		/IOT SHOULD NOT SKIP
4066	4775	JMS	FATAL	/IOT RRCA SKIPPED
4067	3031	DCA	COMDA	/SAVE COMMAND A
/				
4070	6613	IOTE,	RRCB	/READ COMMAND B
4071	7410	SKP		/IOT SHOULD NOT SKIP
4072	4775	JMS	FATAL	/IOT RRCB SKIPPED
4073	3032	DCA	COMDB	/SAVE COMMAND B
/				
4074	6614	IOTF,	RRSA	/READ SECTOR ADDRESS
4075	7410	SKP		/IOT SHOULD NOT SKIP
4076	4775	JMS	FATAL	/IOT RRSA SKIPPED
4077	3033	DCA	ENDSC	/SAVE FINAL SECTOR ADDRESS
/				
4100	3026	DCA	STAT6A	/CLEAR STATUS WORD 1
4101	3027	DCA	STAT6B	/CLEAR STATUS WORD 2
/				
4102	3062	DCA	CBUSY	/CLEAR CONTROLLER BUSY FLAG
4103	1022	TAD	HCV2	/GET HARDWARE CONTROL WORD #2
4104	7700	SMA	CLA	/SKIP IF ON APT
4105	5310	JMP	.+3	/NOT ON APT
4106	6213	CDI	10	/GOING TO FLD 1 FOR APT TIMING
4107	4774	JMS I	(APTTIC	/GO UPDATE APT REPORT TIMER
4110	1024	TAD	CURDRV	/GET CURRENT DRIVE
4111	3216	DCA	PCSAVE	/SAVE TEMPORARILY
4112	1025	TAD	DRV60	/GET DRIVE USING CONTROLLER
4113	3024	DCA	CURDRV	/SAVE AS CURRENT DRIVE

4114	1773'	TAD	XSETP	/GET SETPNT (XSETP) RETURN ADDRESS
4115	3221	DCA	SETTMP	/STORE FOR EXIT
4116	1112	TAD	TBLPNT	/GET TABLE POINTER
4117	3035	DCA	TBLSAV	/SAVE TABLE POINTER FOR EXIT
4120	4547	SETPNT		/SET TBLPNT TO READY PENDING FLAG
4121	0037	RDYFLG		/TABLE INDEX
4122	1512	TAD I	TBLPNT	/GET READY PENDING FLAG
4123	7640	SZA CLA		/SKIP IF NOT READY PENDING (LAST OPR NOT SEEK)
4124	5334	JMP	DRVRES	/LAST OPR WAS A SEEK, RESTORE DRIVE
4125	1130	TAD	FLGSAV	/GET CONTROLLER FLAGS SAVED
4126	0372	AND	(2	/MASK BIT 10 (DONE)
4127	7650	SNA CLA		/SKIP IF DONE INTERRUPT
4130	5334	JMP	DRVRES	/NOT DONE - RESTORE CURRENT DRIVE
4131	4547	SETPNT		/SET TBLPNT TO DRIVE BUSY FLAG
4132	0041	DRVBSY		/TABLE INDEX
4133	3512	DCA I	TBLPNT	/CLEAR DRIVE BUSY FLAG
4134	1216	DRVRES, TAD	PCSAVE	/GET CURRENT DRIVE
4135	3024	DCA	CURDRV	/RESTORE CURRENT DRIVE
4136	1221	TAD	SETTMP	/GET SETPNT (XSETP) RETURN ADDRESS
4137	3773'	DCA	XSETP	/RESTORE SETPNT RETURN ADDRESS
4140	1035	TAD	TBLSAV	/GET TABLE POINTER
4141	3112	DCA	TBLPNT	/RESTORE TABLE POINTER
/				
4142	6031	KBDCHK, KSF		/SKIP IF KEYBOARD FLAG SET
4143	5346	JMP	..+3	/KEYBOARD FLAG NOT SET
4144	4576	VT278		
4145	5356	JMP	KYBDRD	/READ KEYBOARD
4146	6041	TTYCHK, TSF		/SKIP IF TELEPRINTER FLAG
4147	5352	JMP	..+3	/NOT TTY FLG
4150	6042	TCF		/CLEAR TTY FLG
4151	5363	JMP	INTRET	/EXIT
4152	1130	TAD	FLGSAV	/GET FLAGS
4153	7650	SNA CLA		/SKIP IF A CONTROLLER FLAG WAS SET
4154	5771'	JMP	UNKINT	/NO KNOWN FLAGS SET
4155	5363	JMP	INTRET	/GO TO EXIT
/				
4156	6036	KYBDRD, KRB		/READ KEYBOARD BUFFER
4157	0370	AND	(177	/MASK TO 7 BITS
4160	1376	TAD	(200	/SET PARITY BIT
4161	4546	C8CALL		/CALL CONSOLE PACKAGE
4162	7000	NOP		/CONSOLE NOT ACTIVE RETURN
4163	1220	INTRET, TAD	FLAGS	/GET FLAGS
4164	6005	RTF		/RESTORE THE FLAGS
4165	7200	CLA		
4166	1217	TAD	ACSAVE	/RESTORE AC
4167	5400	JMP I	LOCO	/RETURN FROM INTERRUPT
/				
4170	0177			
4171	4200			
4172	0002			
4173	4313			
4174	0351			
4175	6614			
4176	0200			

4177 0352  
4200 4200

PAGE

4200 4545  
4201 4557  
4202 4564  
4203 4572  
4204 7402

UNKINT, APTCHK  
APTERR  
MESAG  
UNINT  
HLT

/CHECK FOR ON APT  
/ON APT REPORT UNKNOWN INTERRUPT  
  
/UNKNOWN INTERRUPT  
/UNKNOWN INTERRUPT HALT

/\*\*\*\*\*

/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

4205 0000  
4206 7308  
4207 6000  
4210 7301  
4211 1122  
4212 3267  
4213 1026  
4214 0377  
4215 7640  
4216 5266

CHKSTA, 0

CLA CLL  
SKON  
CLA CLL IAC  
TAD KION  
DCA CKSOUT  
TAD STAT6A  
AND (40  
SZA CLA  
JMP CKSOUT-1

/SKIP IF INTERRUPT ON AND TURN OFF  
/SET BIT 11, INTERRUPT OFF  
/ADD 6001  
/SAVE ION OR IOF FOR EXIT  
/GET STATUS WRD #1  
/MASK COVER OPEN BIT 6  
/SKIP IF BOTH CLEAR  
/COVER OPEN OR SPARE BIT SET

HP 001

4217 4547  
4220 0003  
4221 1512  
4222 0054  
4223 7112  
4224 7012  
4225 3271  
4226 1026  
4227 0046  
4230 7041  
4231 1271  
4232 7640  
4233 5266

HDCHKA,

SETPNT  
NEWCYL  
TAD I TBLPNT  
AND K2000  
CLL RTR  
RTR  
DCA HDTMP  
TAD STAT6A  
AND K100  
CIA  
TAD HDTMP  
SZA CLA  
JMP CKSOUT-1

/SET TBLPNT TO CYL ADDR  
/TBL INDX  
/GET CYL ADDR AND HEAD SELECT  
/MASK HEAD SELECT BIT  
  
/MOVE HEAD SELECT TO BIT 5  
/SAVE HEAD SELECT TEMPORARILY  
/GET STATUS WORD #1  
/MASK HEAD SELECT BIT 5  
/NEGATE  
/ADD HEAD SELECT EXPECTED  
/SKIP IF EXPECTED HEAD=HEAD FROM GET STATUS  
/HEAD SELECT INCORRECT

4234 1030  
4235 7110  
4236 7620  
4237 5247  
4240 4547

DRDYA,

TAD ERREG  
CLL RAR  
SNL CLA  
JMP STATE4  
SETPNT

/GET ERROR REGISTER  
/MOVE DRIVE READY BIT INTO LINK  
/SKIP IF READY  
/GO CHECK FOR SEEKING (STATE 4) IF NOT READY  
/SET TBLPNT TO DRIVE BUSY FLG

```

4241 0041      DRVBSY      /TBL INDX
4242 3512      DCA I      TBLPNT /CLEAR DRIVE BUSY FLG
4243 4547      SETPNT     /SET TBLPNT TO READY PENDING FLAG
4244 0037      RDYFLG     /TBL INDX
4245 3512      DCA I      TBLPNT /CLEAR READY PENDING FLAG
4246 5254      JMP        STATES /GO CHECK FOR STATE 5 (TRACKING)

/
4247 1026      STATE4. TAD      STAT6A /GET STATUS WRD #1
4250 0044      AND        K7      /MASK STATE BITS 9:11
4251 1056      TAD        M4      /SUBTRACT 4
4252 7650      SNA CLA     /SKIP TO CHECK STATE 5
4253 5261      JMP        BHOME   /GO CHECK FOR BRUSHES HOME AND HEADS OUT

/
4254 1026      STATES. TAD      STAT6A /GET STATUS WRD #1
4255 0044      AND        K7      /MASK STATE BITS
4256 1376      TAD        (-5     /SUBTRACT 5
4257 7640      SZA CLA     /SKIP IF STATE 5 (TRACKING)
4260 5266      JMP        CKSOUT-1 / (NOT STATE 5 AND READY) OR (NOT STATE 5 OR 4)

/
4261 1026      BHOME.  TAD      STAT6A /GET STATUS WRD #1
4262 0375      AND        (30      /MASK BITS 7 (HDS OUT) AND 8 (BRUSH HOME)
4263 7041      CIA        /NEGATE
4264 1375      TAD        (30      /ADD BITS 7 AND 8 =1
4265 7640      SZA CLA     /SKIP IF BRUSHES HOME AND HEADS OVER DISK
4266 2205      ISZ        CHKSTA   /INCREMENT FOR BAD RETURN
4267 7402      CKSOUT. HLT/ION/IOF /MODIFIED ION OR IOF
4270 5605      STAEX2. JMP I      CHKSTA /RETURN

/
4271 0000      HDTMP, 0      /TEMPORARY STORAGE OF EXPECTED HEAD IN BIT 5

/
/*****
/
/
/ROUTINE TO WAIT FOR CONTROLLER NOT BUSY
/
4272 0000      DNWAIT, 0
4273 7300      CLA CLL
4274 1374      TAD        (-24     /20 TIMES 31.2 MSEC. = 624 MSEC. APPROX.
4275 3312      DCA        DNCNT1  /SET UP MULTIPLIER
4276 3311      DCA        DONCNT  /CLEAR WAIT FOR INTERRUPT COUNTER
4277 1062      DONLP, TAD      CBUSY /GET CONTROLLER BUSY FLAG
4300 7650      SNA CLA     /SKIP IF CONTROLLER BUSY
4301 5307      JMP        NOTBSY  /CONTROLLER NOT BUSY
4302 2311      ISZ        DONCNT  /INCREMENT WAIT COUNTER
4303 5277      JMP        DONLP   /CONTINUE WAITING FOR DONE OR ERROR
4304 2312      ISZ        DNCNT1  /INCREMENT MULTIPLIER
4305 5277      JMP        DONLP   /CONTINUE WAITING FOR DONE, COMP. ERROR

/
4306 5672      JMP I      DNWAIT  /FATAL ERROR - NO INTERRUPT
4307 2272      NOTBSY. ISZ      DNWAIT /INCREMENT FOR GOOD RETURN
4310 5672      JMP I      DNWAIT  /GOOD RETURN

/
4311 0000      DONCNT, 0      /INTERRUPT WAIT COUNTER
4312 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)
/
4313 0000 XSETP, 0
4314 6000 SKON
4315 5322 JMP      INTOFF      /SKIP IF INTERRUPT ON AND TURN OFF
4316 3345 DCA      ACSV      /INTERRUPT NOT ON
4317 1122 TAD      KION      /SAVE AC
4320 3343 DCA      XSETEX     /GET ION INSTRUCTION
4321 5325 JMP      LSAV      /SAVE FOR EXIT EXECUTION
4322 3345 INTOFF, DCA      LSAV /GO SAVE LINK
4323 1373 TAD      ACSV      /SAVE AC UPON ENTRY
4324 3343 TAD      (IOF      /GET IOF INSTRUCTION
4325 7004 DCA      XSETEX     /SAVE FOR EXIT EXECUTION
4326 3346 LSAV, RAL      /GET LINK
4327 1372 DCA      LKSV      /SAVE LINK UPON ENTRY
4330 1024 TAD      (DRVPNT    /GET ADDRESS OF INDEX TABLE TO DRIVE STATE TABLE
4331 3112 TAD      CURDRV     /ADD CURRENT DRIVE # AS INDEX INTO TABLE
4332 1512 DCA      TBLPNT    /SAVE TEMPORARILY
4333 3112 TAD I  TBLPNT    /GET POINTER FROM INDEX TABLE
4334 1713 DCA      TBLPNT    /SAVE TEMPORARILY
4335 1112 TAD I  XSETP     /GET TABLE OFFSET DESIGNATOR FROM CALL+1
4336 3112 TAD      TBLPNT    /ADD INDEX TABLE POINTER
4337 2313 DCA      TBLPNT    /SAVE DRIVE STATE TABLE POINTER
4340 1346 ISZ      XSETP     /INCREMENT RETURN ADDRESS
4341 7110 TAD      LKSV      /GET LINK UPON ENTRY
4342 1345 CLL RAR      /RESTORE LINK
4343 7402 XSETEX, HLT/ION/IOF /GET AC UPON ENTRY
4344 5713 JMP I  XSETP     /MODIFIED ION OR IOF DEPENDING ON INT STATE UPON ENTRY
/                   /EXIT
/
4345 0000 ACSV, 0      /AC SAVE AREA
4346 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
/
/
4347 0000  RDYCHK, 0
4350 6610  IOTG,  RRER      /READ ERROR REGISTER INSTRUCTION
4351 7110      CLL      RAR      /PUT DRIVE READY BIT 11 IN LINK
4352 7420      SNL      /SKIP IF DRIVE READY - RETURN CALL+1
4353 2347      ISZ      RDYCHK  /DRIVE IS NOT READY - RETURN CALL+2
4354 5747      JMP I    RDYCHK  /EXIT

```

/\*\*\*\*\*

/\*\*\*\*\*

/ROUTINE TO CALL STATUS PRINTER IN FLD 1

```

/
/      CALLED BY:      STAPRT
/      FOLLOWED BY:    ERTBL1      (EXAMPLE ERROR MESSAGE TABLE ADDRESS)
/
/
4355 0000  PRTSTA, 0
4356 7300      CLA      CLL
4357 1755      TAD I    PRTSTA      /GET CALL+1, START OF MESSAGE ADDRESS
4360 3363      DCA      STATGO     /SAVE FOR FLD 1 CALL
4361 6212      CIF      10        /GOING TO FLD 1
4362 4771      JMS I    (PRNSTA    /GO TO STATUS PRINTER
4363 7402  STATGO, HLT/MSG ADRS    /START OF MESSAGE ADDRESS
4364 2355      ISZ      PRTSTA     /INCREMENT FOR RETURN
4365 5755      JMP I    PRTSTA     /RETURN

```

/\*\*\*\*\*

```

4371 2441
4372 6710
4373 6002
4374 7754
4375 0030
4376 7773
4377 0040
4400

```

PAGE

/ROUTINE TO DO A SEEK TO AN ABSOLUTE CYLINDER ADDRESS  
 /AND HEAD SELECT



/CALL+1=CYL ADDR IN BITS 3:11, HEAD SELECT IN BIT 1  
 /CALLED BY: SETPOS  
 /FOLLOWED BY: NNNN /NNNN=HEAD SELECT IN BIT 1,CYLINDER ADDRESS IN 3:11  
 /RETURN CALL+2 IF UNSUCCESSFUL (DRIVE DROPPED)  
 /RETURN CALL+3 IF SUCCESSFUL  
 /

4400	0000	POSSET, 0			
4401	4573	RESET		/RESET IGNORING VOLUME CHECK	
4402	4573	RESET		/RESET THE DRIVE	
4403	4554	ERRCHK		/CHECK FOR ERRORS	
4404	5207	JMP	+.3	/NO ERROR RETURN	
4405	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
4406	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
4407	4777	JMS	RDYCHK	/CHECK FOR DRIVE READY	
4410	5213	JMP	+.3	/NO ERROR RETURN	
4411	4553	ERROR		/DRIVE NOT READY AFTER RESET	
4412	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
4413	4562	GETSTA		/GET DRIVE STATUS	
4414	4554	ERRCHK		/CHECK FOR ERRORS	
4415	5220	JMP	+.3	/NO ERROR RETURN	
4416	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
4417	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
4420	4777	JMS	RDYCHK	/CHECK DRIVE READY	
4421	5224	JMP	+.3	/NO ERROR RETURN	
4422	4553	ERROR		/DRIVE NOT READY	
4423	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
4424	4776	JMS	RDHDR	/READ HEADER	
4425	4554	ERRCHK		/CHECK FOR ERRORS	
4426	5231	JMP	+.3	/NO ERROR RETURN	
4427	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR	
4430	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT	
4431	1600	TAD I	POSSET	/GET CYL AND HEAD FROM CALL+1	HP 010
4432	7421	MQL		/AND DISPLAY IT IN THE MQ FOR ALL TO SEE	HP 010
4433	1600	TAD I	POSSET	/GET NEW CYL AND HEAD FROM CALL+1	
4434	0052	AND	K777	/MASK OFF HEAD SLCT BIT 1	HP 001
4435	4547	SETPNT		/SET TBLPNT TO COMMAND A	
4436	0033	XCOMA		/TBL INDX	
4437	3512	DCA I	TBLPNT	/SAVE NEW CYL - HD SLCT	
4440	4547	SETPNT		/SET TBLPNT TO CURRENT CYL AND HD	
4441	0002	CURCYL		/TBL INDX	
4442	1512	TAD I	TBLPNT	/GET CURRENT CYL AND HD	
4443	0052	AND	K777	/MASK OFF HD SLCT	HP 001
4444	7041	CIA		/NEGATE	
4445	4547	SETPNT		/SET TBLPNT TO COMMAND A	
4446	0033	XCOMA		/TBL INDX	
4447	1512	TAD I	TBLPNT	/SUBTRACT CURRENT CYL ADDR FROM NEW CYL ADDR	
4450	7500	SMA		/SKIP IF NEW CYL < CURRENT CYL	
4451	5254	JMP	+.3	/NEW CYL > OR = CURRENT CYL	
4452	7041	CIA		/MAKE CYL DIF POSITIVE	
4453	7410	SKP			
4454	1375	TAD	(4000	/SET HEAD DIRECTION FOR INWARD (HIGHER CYL)	
4455	3512	DCA I	TBLPNT	/SAVE COMMAND A CYL DIF WRD	
4456	1600	TAD I	POSSET	/GET NEWCYL AND HEAD FROM CALL+1	
4457	0054	AND	K2000	/MASK HEAD BIT	
4460	1512	TAD I	TBLPNT	/ADD CYL DIF WRD AND DIRECTION	

4461	3512	DCA I	TBLPNT	/SAVE CYL DIF WRD,DIRECTION,HD SLCT
4462	4547	SETPNT		/SET TBLPNT TO NEW CYL AND HD
4463	0003	NEWCYL		/TBL INDX
4464	1600	TAD I	POSSET	/GET NEW CYL AND HD FROM CALL+1
4465	3512	DCA I	TBLPNT	/SAVE NEW CYL AND HD SLCT
4466	4777'	JMS	RDYCHK	/CHECK FOR DRIVE READY
4467	5272	JMP	.+3	/DRIVE READY RETURN
4470	4553	ERROR		/DRIVE NOT READY
4471	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4472	4774'	JMS	SEEK	/SEEK TO NEW CYL AND HEAD
4473	4556	WAITDN		/WAIT FOR DONE
4474	4773'	JMS	FATAL	/NO DONE INTERRUPT
4475	4554	ERRCHK		/CHECK FOR ERRORS
4476	5301	JMP	.+3	/NO ERROR RETURN
4477	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
4500	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4501	4772'	JMS	WATRDY	/WAIT FOR DRIVE READY
4502	5305	JMP	.+3	/NO ERROR RETURN
4503	4553	ERROR		/DRIVE NOT READY TIME-OUT
4504	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4505	4562	GETSTA		/CHECK DRIVE STATUS
4506	4554	ERRCHK		/CHECK FOR ERRORS
4507	5312	JMP	.+3	/NO ERROR RETURN
4510	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
4511	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4512	4563	STACHK		/CHECK DRIVE STATUS WORD #1
4513	5316	JMP	.+3	/NO ERROR
4514	4553	ERROR		/DRIVE STATUS ERROR WORD #1
4515	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4516	7000	NOP		
4517	5322	JMP	.+3	/DRIVE READY RETURN
4520	4553	ERROR		/DRIVE NOT READY
4521	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4522	4776'	JMS	RDHDR	/READ HEADER
4523	4554	ERRCHK		/CHECK FOR ERRORS
4524	5327	JMP	.+3	/NO ERROR RETURN
4525	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
4526	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4527	4562	GETSTA		/GET STATUS
4530	4554	ERRCHK		/CHECK FOR ERRORS
4531	5334	JMP	.+3	/NO ERROR RETURN
4532	4553	ERROR		/STATUS ERROR OR CONTROLLER ERROR
4533	5355	JMP	BADEX2	/DRIVE DROPPED RETURN - EXIT
4534	4547	SETPNT		/SET TBLPNT TO CURRENT CYL AND HD
4535	0002	CURCYL		/TBL INDX
4536	1512	TAD I	TBLPNT	/GET CURRENT CYL AND HD FROM READ HEADER
4537	7041	CIA		/NEGATE
4540	4547	SETPNT		/SET TBLPNT TO EXPECTED CYL AND HD
4541	0003	NEWCYL		/TBL INDX
4542	1512	TAD I	TBLPNT	/ADD EXPECTED CYL AND HD TO CURRENT CYL AND HD
4543	7650	SNA CLA		/SKIP IF EXPECTED CYL AND HD NOT = CURRENT CYL AND HD
4544	5351	JMP	SOK	/NO SEEK ERROR
4545	4771'	JMS	LOGERR	/LOG A SEEK ERROR
4546	0015	SEKERR		/DRV TBL INDX
4547	4553	ERROR		/SEEK ERROR, GO TO ERROR HANDLER

```

4550 5355      JMP      BADEX2      /GO TO DRIVE DROPPED EXIT
4551 4547      SOK,    SETPNT      /SET TBLPNT TO POSITION VERIFY FLAG
4552 0036      POSFLG      /TBL INDX
4553 3512      DCA I    TBLPNT      /CLEAR POSITION VERIFICATION NEEDED FLAG
4554 2200      ISZ      POSSET      /INCREMENT FOR SUCCESSFUL RETURN
4555 2200      BADEX2, ISZ      POSSET /INCREMENT FOR GOOD OR BAD RETURN
4556 5600      JMP I    POSSET      /RETURN

```

```

/
/*****
/
/

```

```

/ROUTINE TO PRINT 2 SPACES
/

```

```

/CALLED BY:      SPACE2
/

```

```

4557 0000      SPACES, 0
4560 7300      CLA CLL
4561 1366      TAD      K240
4562 4566      PRNT
4563 1366      TAD      K240
4564 4566      PRNT
4565 5757      JMP I    SPACES      /RETURN
4566 0240      K240,    0240

```

```

4571 5756
4572 6400
4573 6614
4574 6314
4575 4000
4576 5243
4577 4347
4600

```

```

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
/

```

```

4600 0000      CHKERR, 0

```

4601	7300	CLA	CLL	
4602	6000	SKON		/SKIP IF INTERRUPT ON AND TURN OFF
4603	7301	CLA CLL	IAC	/SET BIT 11 FOR IOF
4604	1122	TAD	KION	/ADD 6001
4605	3300	DCA	CHKOUT	/SAVE ION OR IOF FOR EXECUTION
4606	4777'	JMS	SAVDRV	/GO GET DRV60 AND SAVE CURDRV
4607	4776'	JMS	CHKACT	/CHECK FOR DRIVE ACTIVE
4610	5276	JMP	EREX1	/DRIVE NOT ACTIVE RETURN
4611	1130	TAD	FLGSAV	/GET FLAGS FROM LAST INTERRUPT
4612	7110	CLL RAR		/MOVE COMPOSITE ERROR INTO LINK
4613	7630	SZL CLA		/SKIP IF COMPOSITE ERROR NOT SET
4614	5275	JMP	EREX1-1	/COMPOSITE ERROR FLG SET
4615	1030	TAD	ERREG	/GET ERROR REGISTER
4616	0375	AND	(7002	/MASK ERROR BITS AND DRIVE ERROR BIT
4617	7640	SZA CLA		/SKIP IF NO ERRORS
4620	5275	JMP	EREX1-1	/CONTROLLER ERROR BITS SET, COMPOSITE ERROR FLAG
				/MAY HAVE BEEN CLEARED BY GET STATUS
4621	1027	TAD	STAT6B	/GET DRIVE STATUS WORD #2
4622	7650	SNA CLA		/SKIP IF DRIVE ERROR BITS NOT CLEAR
4623	5226	JMP	.+3	/ERROR BITS CLEAR
4624	4774'	JMS	WRTLOC	/CHECK FOR WRITE LOCK ON DRIVE
4625	4773'	JMS	FATAL	/DRIVE ERROR BITS SET BUT NOT COMPOSITE ERROR FLG
4626	4547	SETPNT		/SET TBLPNT TO EXPECTED (SENT) COMMAND A
4627	0033	XCOMA		/TABLE INDEX
4630	1512	TAD I	TBLPNT	/GET COMMAND A SENT
4631	7041	CIA		/NEGATE
4632	1031	TAD	COMDA	/ADD COMMAND A READ
4633	7640	SZA CLA		/SKIP IF COMMAND A SENT = COMMAND A READ
4634	4773'	JMS	FATAL	/ERROR LOADING COMMAND A REGISTER
4635	4547	SETPNT		/SET TBLPNT TO EXPECTED (SENT) COMMAND B
4636	0034	XCOMB		/TABLE INDEX
4637	1512	TAD I	TBLPNT	/GET COMMAND B SENT
4640	7041	CIA		/NEGATE
4641	1032	TAD	COMDB	/ADD COMMAND B READ
4642	7640	SZA CLA		/SKIP IF COMMAND B SENT = COMMAND B READ
4643	4773'	JMS	FATAL	/ERROR LOADING COMMAND B REGISTER
4644	1512	TAD I	TBLPNT	/GET COMMAND B SENT
4645	0044	AND	K7	/MASK FUNCTION CODE BITS
4646	1372	TAD	(-5	/SUBTRACT 5
4647	7700	SMA CLA		/SKIP IF FUNCTION CODE < 5
4650	5261	JMP	ZWRD	/DATA TRANSFER, EXPECT ZERO FINAL WORD COUNT
4651	4547	SETPNT		/SET TBLPNT TO WORD COUNT LOADED
4652	0030	WRDCNT		/TBL INDX
4653	1512	TAD I	TBLPNT	/GET WORD COUNT LOADED
4654	7041	CIA		/NEGATE FOR CHECKING
4655	1034	TAD	ENDWD	/ADD FINAL WORD COUNT READ
4656	7640	SZA CLA		/SKIP IF FINAL WORD COUNT READ = WORD COUNT LOADED
4657	4773'	JMS	FATAL	/ERROR LOADING WORD COUNT REGISTER
4660	5264	JMP	FINSEC	/GO CHECK FINAL SECTOR REGISTER CONTENTS
4661	1034	TAD	ENDWD	/GET FINAL WORD COUNT READ
4662	7640	SZA CLA		/SKIP IF = ZERO
4663	4773'	JMS	FATAL	/FINAL WORD COUNT NOT 0 AND COMPOSITE ERROR FLAG NOT SET
4664	4547	SETPNT		/SET TBLPNT TO FINAL SECTOR EXPECTED
4665	0040	XENDSC		/TABLE INDEX
4666	1512	TAD I	TBLPNT	/GET FINAL SECTOR EXPECTED

ZWRD,

FINSEC,

```

4667 7002      BSW      /SWAP TO OTHER BYTE
4670 7041      CIA      /NEGATE FOR CHECKING
4671 1033      TAD      ENDSC /ADD FINAL SECTOR READ
4672 7640      SZA CLA   /SKIP IF FINAL SECTOR READ = FINAL SECTOR EXPECTED
4673 4773      JMS      FATAL /SECTOR ADDRESS REGISTER ERROR
4674 5276      JMP      EREX1 /GO TO GOOD RETURN

/
4675 2200      ISZ      CHKERR /INCREMENT FOR ERROR RETURN
4676 1135      EREX1, TAD  DRVTMP /GET ACTUAL CURRENT DRIVE #
4677 3024      DCA      CURDRV  /RESTORE CURRENT DRIVE
4700 7402      CHKOUT, HLT/ION/IOF /MODIFIED ION OR IOF
4701 5600      JMP I    CHKERR  /RETURN

/
/
/*****
/ROUTINE TO GENERATE DATA FOR DATA BUFFER
/
/CALLED BY: JMS DATGEN
/
/LOCATION "CDFSAV" MUST CONTAIN A CDF TO BUFFER FIELD
/ENTER WITH 2'S COMPLEMENT WORD COUNT IN AC
/LOCATION "BITMOD" MUST CONTAIN BIT MODE 0=12 BIT MODE 1000=8 BIT MODE
/LOCATION "AUTO17" MUST CONTAIN BUFFER ADDRESS-1 (AUTO INDX REG.)
/USES RANDOMLY SELECTED DATA PATTERNS UNLESS "CONFLG" = 0
/
/
4702 0000      DATGEN, 0
4703 3041      DCA      WDCNTR  /SET UP WORD COUNTER
4704 1074      TAD      CONFLG  /GET CONSTANT DATA FLAG
4705 7640      SZA CLA   /SKIP IF USING CONSTANT DATA PATTERN
4706 4552      RANDOM    /GET RANDOM NUMBER
4707 1104      TAD      DATPAT  /GET CONSTANT DATA PATTERN
4710 0042      AND      K3      /MASK TO 3 BITS
4711 3136      DCA      PATMP   /SAVE DATA PATTERN
4712 1111      TAD      BITMOD  /GET BIT MODE
4713 7650      SNA      CLA     /SKIP IF 8 BIT MODE
4714 5322      JMP      BITM12  /GO HANDLE 12 BIT MODE

/
/
4715 1121      BITM8, TAD  CDFSAV /GET CDF TO BUFFER FIELD
4716 3771      DCA      XCDF4    /SAVE FOR EXECUTION
4717 1121      TAD      CDFSAV  /GET CDF TO BUFFER FIELD
4720 3770      DCA      XCDF5    /SAVE FOR EXECUTION
4721 5767      JMP      PATLP8   /GO GENERATE 8 BIT DATA

/
4722 1121      BITM12, TAD CDFSAV /GET MODIFIED CDF TO BUFFER FIELD
4723 3333      DCA      XCDF2    /SAVE FOR EXECUTION
4724 1121      TAD      CDFSAV  /GET MODIFIED CDF TO BUFFER FIELD
4725 3352      DCA      XCDF3    /SAVE FOR EXECUTION
4726 7305      CLA CLL IAC RAL  /AC+=2
4727 1041      TAD      WDCNTR  /ADD WORD COUNT
4730 3040      DCA      SECCNT  /SAVE SECTOR DATA WORD COUNT ( NON-ZERO FILL )
4731 1040      TAD      SECCNT  /GET SECTOR NON-ZERO FILL WORD COUNT

```

4732	7041	CIA		/MAKE POSITIVE FOR 1ST WORD OF SECTOR
4733	7402	XCDF2, HLT/CDF		/MODIFIED CDF TO BUFFER FIELD
4734	3417	DCA I	AUTO17	/STORE 1ST WORD=NUMBER NON-ZERO FILL WORDS
4735	1136	TAD	PATMP	/GET DATA PATTERN NUMBER USED
4736	3417	DCA I	AUTO17	/STORE 2ND WORD=DATA PATTERN NUMBER
4737	6201	CDF 00		/CDF TO PRGM FLD
4740	1366	PATLPA, TAD	(-26	
4741	3132	DCA	PATCNT	/PATTERN WORD COUNT = -22(10)
4742	1136	TAD	PATMP	/GET DATA PATTERN NUMBER
4743	1365	TAD	(DPAT12	/ADD AS INDEX TO ADDR OF DATA PATTERN TBL
4744	3137	DCA	DTMPA	/SAVE DATA PATTERN POINTER TABLE ADDRESS
4745	1537	TAD I	DTMPA	/GET ADDRESS OF DATA PATTERN FROM TABLE
4746	1055	TAD	M1	/SUBTRACT ONE FOR AUTO INDEX
4747	3016	DCA	AUTO16	/SET UP PATTERN TABLE POINTER
4750	6211	FILBF1, CDF	10	/CDF TO PATTERN TABLE FIELD 1
4751	1416	TAD I	AUTO16	/GET A DATA PATTERN WORD
4752	7402	XCDF3, HLT/CDF		/MODIFIED CDF TO BUFFER FIELD
4753	3417	DCA I	AUTO17	/STORE WORD IN BUFFER
4754	6201	CDF 00		/CDF TO PRGM FLD
4755	2040	ISZ	SECCNT	/INCREMENT SECTOR NON-ZERO WORD COUNT
4756	7410	SKP		/SKIP IF NOT DONE GENERATING DATA
4757	5363	JMP	DATEXA	/DONE, EXIT
4760	2132	ISZ	PATCNT	/INCREMENT PATTERN WORD COUNT
4761	5350	JMP	FILBF1	/GO DO NEXT PATTERN WORD
4762	5340	JMP	PATLPA	/START PATTERN OVER
4763	5702	/		
		DATEXA, JMP I	DATGEN	/RETURN
		/		
		/		
		/		
		/		
		/		
		/		
		/		
4765	7134			
4766	7752			
4767	5000			
4770	5046			
4771	5027			
4772	7773			
4773	6614			
4774	3142			
4775	7002			
4776	1340			
4777	6600			
	5000	PAGE		
		/		
		/		
5000	1377	PATLPB, TAD	(-376	/SECTOR WRD CNT = 256(10) - 1ST 2 WORDS
5001	3120	DCA	CNTR1	/SET UP SECTOR WORD COUNT
5002	7305	CLA CLL	IAC RAL	/AC+=2
5003	1041	TAD	WDCNTR	/WORD COUNT LESS 2 FOR FIRST 2 WORDS
5004	3041	DCA	WDCNTR	/SAVE WORD COUNT
5005	1041	TAD	WDCNTR	/GET WORD COUNT

5006	7500	SMA		/SKIP IF NEGATIVE
5007	5220	JMP	RANCNT	/MORE THAN ONE SECTOR
5010	1376	TAD	(376	/ADD 254(10) TO SEE IF ONE SECTOR
5011	7710	SPA	CLA	/SKIP IF ONE SECTOR
5012	5220	JMP	RANCNT	/MORE THAN ONE SECTOR
5013	1041	TAD	WDCNTR	/GET WORD COUNT
5014	3040	DCA	SECCNT	/SAVE AS SECTOR DATA WORD COUNT
5015	1040	TAD	SECCNT	/GET DATA WORD COUNT
5016	7041	CIA		/MAKE IT POSITIVE FOR 1ST WORD OF SECTOR
5017	5227	JMP	XCDF4	/GO CDF TO BUFFER FLD
5020	4552	RANCNT,	RANDOM	/GET RANDOM NUMBER
5021	0050	AND	K377	/MASK TO 8 BITS
5022	1055	TAD	M1	/SUBTRACT 1 FOR MAX 254(10)
5023	7041	CIA		/NEGATE FOR USE AS COUNTER
5024	3040	DCA	SECCNT	/SAVE NUMBER OF NON-ZERO FILL WORDS - 1ST 2 WORDS
5025	1040	TAD	SECCNT	/GET NUMBER OF NON-ZERO FILL WRDS
5026	7041	CIA		/MAKE POSITIVE FOR FIRST WORD OF SECTOR
5027	7402	XCDF4,	HLT/CDF	/CDF TO BUFFER FIELD
5030	3417	DCA I	AUTO17	/STORE 1ST WRD=NUMBER NON-ZERO FILL WRDS
5031	1136	TAD	PATMP	/GET DATA PATTERN NUMBER
5032	3417	DCA I	AUTO17	/STORE 2ND WRD=DATA PATTERN NUMBER
5033	6201	CDF	00	/CDF TO PRGM FLD
5034	1375	PATLPC,	TAD (-40	
5035	3132	DCA	PATCNT	/32(10) WRDS FOR 8 BIT DATA PATTERN
5036	1136	TAD	PATMP	/GET DATA PATTERN NUM AS INDEX
5037	1374	TAD	(DPAT8	/ADD ADDRESS OF DATA PATTERN TABLE
5040	3137	DCA	DTMPA	/SAVE DATA PATTERN POINTER TABLE ADDRESS
5041	1537	TAD I	DTMPA	/GET ADDRESS OF DATA PATTERN FROM TABLE
5042	1055	TAD	M1	/SUBTRACT 1 FOR AUTO INDEX
5043	3016	DCA	AUTO16	/SET UP POINTER TO DATA PATTERN
5044	6211	FILBF2,	CDF 10	/CDF TO DATA PATTERN FIELD 1
5045	1416	TAD I	AUTO16	/GET A DATA WRD FROM DATA PAT TBL
5046	7402	XCDF5,	HLT/CDF	/CDF TO BUFFER FIELD
5047	3417	DCA I	AUTO17	/STORE DATA WRD IN BUFFER
5050	6201	CDF	00	/CDF TO PRGM FLD
5051	2041	ISZ	WDCNTR	/INCREMENT TOTAL WORD COUNT
5052	7410	SKP		
5053	5773	JMP	DATEXA	/RETURN BUFFER IS SET UP
5054	2120	ISZ	CNTR1	/INCREMENT SECTOR WORD COUNT
5055	7410	SKP		
5056	5200	JMP	PATLPB	/GO START A NEW SECTOR
5057	2040	ISZ	SECCNT	/INCREMENT SECTOR NON-ZERO WRD CNT
5060	5264	JMP	.+4	/
5061	7240	CLA	CMA	/SECCNT=-1 TO SKIP UNTIL SECTOR DONE
5062	3040	DCA	SECCNT	/KEEP FILLING WITH ZEROS
5063	5246	JMP	XCDF5	/GO FILL REST OF SECTOR WITH ZEROS
5064	2132	ISZ	PATCNT	/INCREMENT PATTERN WORD CNT
5065	5244	JMP	FILBF2	/GO DO NEXT PATTERN WORD
5066	5234	JMP	PATLPC	/START PATTERN OVER

/  
 /  
 /  
 /  
 /

```

/*****
/
/
/ROUTINE TO EXECUTE A COMMAND TO A DRIVE
/
/      CALLED BY:      GO
/
/
5067 0000 XGO,      0
5070 6002      IOF      /INHIBIT INTERRUPTS
5071 7340 SETBSY, CLA CLL CMA
5072 3062      DCA      CBUSY      /SET CONTROLLER BUSY FLAG
5073 1024      TAD      CURDRV      /GET CURRENT DRIVE
5074 3025      DCA      DRV60      /SAVE AS DRIVE USING CONTROLLER
5075 4547      SETPNT      /SET TBLPNT TO DRIVE BUSY FLAG
5076 0041      DRVBSY      /TABLE INDEX
5077 7340      CLA CLL CMA
5100 3512      DCA I      TBLPNT      /SET DRIVE BUSY FLAG
5101 4547      SETPNT      /SET TBLPNT TO DRIVE ACCESS ATTEMPT COUNT
5102 0035      DACNT      /TABLE INDEX
5103 3512      DCA I      TBLPNT      /CLEAR DRIVE ACCESS ATTEMPT COUNT
5104 4547      SETPNT
5105 0030      WRDCNT      /SET "TBLPNT" TO WORD COUNT IN DRIVE STATE TABLE
5106 1512      TAD I      TBLPNT      /GET WORD COUNT FROM DRIVE STATE TABLE
5107 6607 CNTLOD, RLWC      /LOAD WORD COUNT IOT
5110 7440      SZA      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
5111 4772'      JMS      FATAL      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
5112 4547      SETPNT
5113 0031      INITCA      /SET TBLPNT TO BREAK MEM ADDR
5114 1512      TAD I      TBLPNT      /GET BREAK MEM ADDR FROM DRIVE STATE TBL
5115 6602 BRKLOD, RLMA      /LOAD BREAK MEM ADDR IOT
5116 7440      SZA      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
5117 4772'      JMS      FATAL      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
5120 4547      SETPNT
5121 0032      SECADD      /SET TBLPNT TO SECTOR ADDRESS
5122 1512      TAD I      TBLPNT      /GET SECTOR ADDRESS
5123 7002      BSW      /MOVE TO BITS 0:5 FROM 6:11
5124 6605 SECLOD, RLSA      /LOAD OR CLEAR SECTOR ADDRESS IOT
5125 7440      SZA      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
5126 4772'      JMS      FATAL      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
5127 4547      SETPNT
5130 0033      XCOMA      /SET TBLPNT TO COMMAND A
5131 1512      TAD I      TBLPNT      /GET COMMAND A
5132 6603 ALODE, RLCA      /LOAD OR CLEAR COMMAND A
5133 7440      SZA      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
5134 4772'      JMS      FATAL      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
5135 4547      SETPNT
5136 0034      XCOMB      /SET TBLPNT TO COMMAND B
5137 1512      TAD I      TBLPNT      /GET COMMAND B
/
5140 6604 BLODE, RLCB      /LOAD COMMAND B, EXECUTE FUNCTION IOT
5141 7440      SZA      /SKIP IF AC CLEARED AND IOT DID NOT SKIP
5142 4772'      JMS      FATAL      /IOT SKIPPED OR FAILED TO CLEAR AC - FATAL TO SUBSYSTEM
/
/

```



```

5143 6001 GOEXIT, ION /ENABLE INTERRUPTS
5144 5667 JMP I XGO /EXIT

/
/*****
/
/ROUTINE TO GENERATE A RANDOM NUMBER
/RETURN WITH NUMBER IN AC
/
/ CALLED BY: RANDOM
/
5145 0000 XRAND, 0
5146 7301 CLA CLL IAC
5147 1361 TAD RAN1
5150 1362 TAD RAN2
5151 7106 CLL RTL
5152 3361 DCA RAN1
5153 1362 TAD RAN2
5154 7012 RTR
5155 1361 TAD RAN1
5156 3362 DCA RAN2
5157 1362 TAD RAN2
5160 5745 JMP I XRAND

/
5161 1234 RAN1, 1234
5162 5670 RAN2, 5670
/
/
/*****
/
/ROUTINE TO CHECK FOR ON APT
/RETURN CALL+1 IF ON APT
/RETURN CALL+2 IF NOT ON APT
/
/ CALLED BY: APTCHK
/
5163 0000 CHKAPT, 0
5164 7300 CLA CLL
5165 1022 TAD HCW2 /GET HCW2
5166 7700 SMA CLA /SKIP IF BIT 0=1, ON APT RETURN CALL+1
5167 2363 ISZ CHKAPT /INCREMENT RETURN POINTER - NOT ON APT
5170 5763 JMP I CHKAPT /EXIT

/
/*****
/
5172 6614
5173 4763
5174 7130

```

5175 7740  
5176 0376  
5177 7402  
5200

PAGE

//  
//  
//  
//  
//  
//  
//

/ROUTINE TO GET STATUS  
/CALLED BY: GETSTA  
/RETURN ONLY IF SUCCESSFUL  
/

5200 0000  
5201 7300  
5202 6035

STAGET, 0

CLA CLL  
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.

5203 6045  
5204 6002  
5205 4547  
5206 0033  
5207 3512  
5210 4547  
5211 0032  
5212 3512  
5213 4547  
5214 0040  
5215 3512  
5216 4547  
5217 0034  
5220 1024  
5221 7002  
5222 1377  
5223 3512  
5224 4555  
5225 4556  
5226 4776

TIE  
IOF  
SETPNT  
XCOMA  
DCA I TBLPNT  
SETPNT  
SECADD  
DCA I TBLPNT  
SETPNT  
XENDSC  
DCA I TBLPNT  
SETPNT  
XCOMB  
TAD CURDRV  
BSW  
TAD (1402  
DCA I TBLPNT  
GO  
WAITDN  
JMS FATAL

/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

5227 6002  
5230 6615  
5231 7410  
5232 4776  
5233 0050  
5234 3026  
5235 6615  
5236 7410  
5237 4776  
5240 0050

RDSIA,

IOF  
RRSI  
SKP  
JMS FATAL  
AND K377  
DCA STAT6A  
RDSIB, RRSI  
SKP  
JMS FATAL  
AND K377

/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  
/MASK OFF GARBAGE BITS

```

5241 3027      DCA      STAT6B      /SAVE STATUS WORD #2
5242 5600      JMP I     STAGET      /EXIT

/
/
/
/
/
/*****
/
/ROUTINE TO READ HEADER
/CALLED BY:      JMS      RDHDR
/RETURN ONLY IF SUCCESSFUL
/
/
5243 0000      RDHDR, 0
5244 7300      CLA      CLL
5245 6002      IOF
5246 4547      SETPNT
5247 0034      XCOMB
5250 1024      TAD      CURDRV
5251 7002      BSW
5252 1375      TAD      (1404
5253 3512      DCA I     TBLPNT
5254 4555      GO
5255 4556      WAITDN
5256 4776      JMS      FATAL

/INTERRUPTS OFF
/SET TBLPNT TO COMMAND B
/TABLE INDEX
/GET CURRENT DRIVE
/MOVE DRIVE SELECT TO BITS 4,5
/ADD INTERRUPT ENABLE, 8 BIT MODE, CODE 4
/STORE COMMAND B
/EXECUTE COMMAND
/WAIT FOR DONE OR ERROR
/DONE DID NOT SET

/
5257 6615      RDSI1, RRSI
5260 0050      AND      K377
5261 3317      DCA      SILO1

/READ SILO WORD #1, SECTOR ADDR, HEAD SELECT, AND CYL ADDR LSB
/MASK OFF GARBAGE BITS
/SAVE SILO WORD #1

/
5262 6615      RDSI2, RRSI
5263 0050      AND      K377
5264 3320      DCA      SILO2

/READ SILO WORD #2, CYL ADDR
/MASK OFF GARBAGE BITS
/SAVE SILO WORD #2

/
5265 6615      RDSI3, RRSI
5266 0050      AND      K377
5267 7640      SZA      CLA
5270 4774      JMS      HDRERR

/READ SILO WORD #3, SHOULD BE ALL ZEROES
/MASK OFF UNUSED BITS
/SKIP IF WORD #3 IS ZERO
/HEADER ERROR WORD #3 NOT ZERO

/
5271 6615      RDSI4, RRSI
5272 0050      AND      K377
5273 7640      SZA      CLA
5274 4774      JMS      HDRERR

/READ SILO WORD #4, SHOULD BE ALL ZEROES
/MASK OFF UNUSED BITS
/SKIP IF WORD #4 IS ZERO
/HEADER ERROR WORD #4 NOT ZERO

/
/
/
5275 1317      TAD      SILO1
5276 7002      BSW
5277 7012      RTR
5300 7200      CLA
5301 1320      TAD      SILO2

/GET SILO WORD #1
/MOVE CYL ADDR LSB FROM BIT 4 TO BIT 10
/MOVE CYL ADDR LSB FROM BIT 10 TO LINK
/GET RID OF SECTOR ADDR BITS
/GET SILO WORD #2

```

```

5302 7004      RAL          /PUT CYL ADDR LSB IN BIT 11
5303 0052      AND          /MASK OFF UNUSED BITS
5304 4547      SETPNT      K777      /SET TBLPNT TO CURRENT CYL
5305 0002      CURCYL      /TABLE INDEX
5306 3512      DCA I      TBLPNT    /SAVE CURRENT CYL
5307 7201      CLA IAC     /SET BIT 11
5310 7002      BSW        /SET BIT 5
5311 0317      AND          /MASK HEAD SELECT BIT
5312 7106      CLL RTL     /MOVE TO BIT 3
5313 7106      CLL RTL     /MOVE HEAD SELECT TO BIT 1
5314 1512      TAD I      TBLPNT    /ADD CURRENT CYL
5315 3512      DCA I      TBLPNT    /SAVE CURCYL AND HEAD SELECT IN CURCYL
5316 5643      JMP I      RDHDR     /RETURN

5317 0000      SILO1, 0      /SECTOR ADDR BITS 6:11, CYL ADDR LSB BIT 4, HEAD SELECT BIT 5
5320 0000      SILO2, 0      /CYL ADDR MSB=BIT 4, REST OF CYL ADDR IN BITS 6:11
/
/*****
/
/MAIN ERROR HANDLER
/
/      CALLED BY:      ERROR
/      FOLLOWED BY:     DRIVE DROPPED RETURN
/      FOLLOWED BY:     GOOD RETURN
/
5321 0000      XERROR, 0
5322 6002      IOF        /INTERRUPTS OFF
5323 7300      CLA        CLL
5324 4773      JMS        SAVDRV    /SAVE "CURDRV" AND GET "DRV60" (LAST DRV USING CONTROLLER)
5325 1772      TAD        ERRFLG    /GET ERROR FLAG TO SEE IF FIRST ERROR CALL
5326 7640      SZA        CLA
5327 5771      JMP        RETRY     /SKIP IF FIRST ERROR CALL
5330 7340      CLA CLL CMA    /NOT FIRST CALL, GO TO RETRY ROUTINE
5331 3772      DCA        ERRFLG    /SET ERROR FLAG FOR FUTURE ERROR CALLS
5332 1321      TAD        XERROR    /GET ERROR CALL RETURN PC
5333 3770      DCA        ERRPC     /SAVE FOR RETURN
5334 1767      TAD        POSSET    /GET SEEK ROUTINE RETURN PC IF HERE BECAUSE OF SEEK OR READ HEADER ERROR
5335 3766      DCA        POSPC     /SAVE RETURN PC FOR SEEK ROUTINE
5336 3110      DCA        RETCNT    /CLEAR RETRY COUNT
5337 6212      CIF        10       /GOING TO FLD 1
5340 4765      JMS I      (SAVALL)  /GO SAVE ALL REGISTERS AT TIME OF ERROR
5341 4545      APTCHK     /CHECK FOR ON APT
5342 5764      JMP        DETST    /ON APT NO REPORT
5343 4550      GETSWR     /GET SWITCHES
5344 0046      AND        K100     /TEST INHIBIT ERROR MESSAGE SWR5=1
5345 7640      SZA        CLA
5346 5764      JMP        DETST    /SKIP IF OK TO PRINT ERROR MSG
5347 4571      DOCLRF     /NO MSG, GO TEST FOR DRIVE ERROR
5350 4564      MESAG      /<CR LF>
5351 4627      ERRPCM     /"ERROR PC:"
5352 1321      TAD        XERROR    /GET CALL PC+1
5353 1055      TAD        M1        /SUBTRACT 1
5354 4570      PRNAC      /PRINT ERROR CALL PC

```

HP 001

```

5355 4571      DDCRLF      /<CR LF>
/
5356 4551      SPRNT, STAPRT /PRINT DRIVE #
5357 3165      ERTBL4      /START OF MSG ADDRESS
5360 4551      STAPRT      /PRINT CONTROLLER REGISTERS
5361 2541      ERTBL1      /START OF MSG ADDRESS
5362 5764      JMP         DETST /GO TO NEXT PAGE
/
/
5364 5400
5365 2654
5366 5731
5367 4400
5370 5730
5371 5625
5372 5732
5373 6600
5374 6352
5375 1404
5376 6614
5377 1402
5400          PAGE
/
/
/
5400 1030      DETST, TAD     ERREG      /GET ERROR REGISTER
5401 7112      CLL      RTR      /MOVE DRIVE ERROR BIT 10 INTO LINK
5402 7620      SNL      CLA      /SKIP IF DRIVE ERROR SET
5403 5275      JMP      NDE      /NO DRIVE ERROR, GO HANDLE NON-DRV ERROR
5404 4562      GETSTA      /GET DRIVE STATUS
5405 1030      TAD      ERREG      /GET ERROR REGISTER
5406 7106      CLL      RTL      /PUT OPI IN LINK
5407 7620      SNL      CLA      /SKIP IF OPI FROM GET STATUS, NO SYSTEM CLOCK
5410 5230      JMP      SCK1      /GO CHECK STATUS
5411 4777      JMS      LOGERR      /LOG OPI ERROR
5412 0023      OPIERR      /TABLE INDX
/
5413 4776      DRVGON, JMS      DELAY      /DELAY FOR 4 SECONDS
5414 7774      -4          /TIMER
5415 4562      GETSTA      /GET STATUS FOR PRINTOUT
5416 4777      JMS      LOGERR      /LOG DRIVE ERROR
5417 0014      DRVERR      /DRIVE STATUS TABLE INDEX
5420 4777      JMS      LOGERR      /LOG HARD ERROR
5421 0012      HRDERR      /DRIVE STATE TABLE INDEX
5422 4550      GETSWR      /GET SWITCHES
5423 0375      AND      (2.      /TEST BIT 10, INHIBIT DRIVE DROP
5424 7650      SNA      CLA      /SKIP IF NOT DROPPING DRIVE BIT 10 = 1
5425 4774      JMS      KILDRV      /DROP DRIVE
5426 4773      JMS      FINRPT      /GO FINISH REPORT
5427 5772      JMP      EREXT2      /RETURN CALL+1, DRIVE DROPPED
/
5430 1027      SCK1, TAD      STAT6B      /GET STATUS WORD #2
5431 0371      AND      (37      /CHECK FOR SKTO,SPIN,WGE,VOL CHK,OR DRV SLCT ERROR
5432 7650      SNA      CLA      /SKIP IF ONE OF THE ABOVE OR MORE
5433 5251      JMP      SCK2      /NONE OF THE ABOVE, GO CHECK FOR OTHER ERRORS

```

5434	1026	TAD	STAT6A	/GET STATUS WORD #1
5435	0044	AND	K7	/MASK DRIVE STATE BITS
5436	7650	SNA	CLA	/SKIP IF NOT LOAD STATE (0)
5437	5256	JMP	SCK3	/GO CHECK FOR OTHER ERRORS, LOAD STATE
5440	4573	RESET		/RESET THE DRIVE
5441	4776'	JMS	DELAY	/DELAY
5442	7774	-4		/4 SECS. (APPROX.)
5443	4562	GETSTA		/GET STATUS
5444	1027	TAD	STAT6B	/GET STATUS WORD #2
5445	0371	AND	(37	/TEST DRIVE ERROR BITS AGAIN
5446	7650	SNA	CLA	/SKIP IF ANY ERROR BITS SET
5447	5770'	JMP	RETRY	/GO TO RETRY ROUTINE
5450	5213	JMP	DRVGON	/GO DROP THE DRIVE
/				
5451	1027	SCK2,	TAD	STAT6B
5452	0367		AND	(140
5453	7650		SNA	CLA
5454	5256		JMP	SCK3
5455	5213		JMP	DRVGON
/				
5456	1027	SCK3,	TAD	STAT6B
5457	0366		AND	(210
5460	7650		SNA	CLA
5461	5770'		JMP	RETRY
5462	4776'		JMS	DELAY
5463	7777		-1	
5464	4573		RESET	
5465	4776'		JMS	DELAY
5466	7730		-50	
5467	4562		GETSTA	
5470	1026		TAD	STAT6A
5471	0044		AND	K7
5472	7640		SZA	CLA
5473	5770'		JMP	RETRY
5474	5213		JMP	DRVGON
/				
5475	1030	NDE,	TAD	ERREG
5476	0365		AND	(7000
5477	7450		SNA	
5500	5304		JMP	.+4
5501	1364		TAD	(6000
5502	7640		SZA	CLA
5503	5332		JMP	NOPION
5504	4763'		JMS	CHKRDY
5505	5321		JMP	STATC
/				
5506	1134	ABORT,	TAD	DERFLG
5507	7640		SZA	CLA
5510	5770'		JMP	RETRY
5511	4562		GETSTA	
5512	4777'		JMS	LOGERR
5513	0013		SFTERR	
5514	4777'		JMS	LOGERR
5515	0025		CTLERR	
5516	4773'		JMS	FINRPT

/GO FINISH REPORT

```

5517 4762' JMS RLSBUF /RELEASE BUFFER
5520 5772' JMP EREXT2 /RETURN AS IF DRIVE DROPPED (ABORT)

/
5521 4562 STATC, GETSTA /DRIVE NOT READY, GET STATUS
5522 1026 TAD STAT6A /GET STATUS WORD #1
5523 0044 AND K7 /MASK DRIVE STATE BITS
5524 1056 TAD M4 /SUBTRACT 4
5525 7640 SZA CLA /SKIP IF STATE 4 (SEEK-TRACK COUNTING)
5526 5213 JMP DRVGON /NOT STATE 4
5527 4776' JMS DELAY /DELAY
5530 7774 -4 /WAIT 4 SECS. (APPROX.) FOR SKTO
5531 5230 JMP SCK1 /SEEK TIME-OUT EXPECTED

/
5532 1030 NOPION, TAD ERREG /GET ERROR REGISTER
5533 0054 AND K2000 /TEST FOR OPI
5534 7650 SNA CLA /SKIP IF OPI SET
5535 5761' JMP NOTOPI /OPI NOT SET
5536 1030 TAD ERREG /GET ERROR REGISTER
5537 0053 AND K1000 /TEST FOR HEADER NOT FOUND (HNF)
5540 7650 SNA CLA /SKIP IF HNF SET
5541 5347 JMP CHKHCR /GO CHECK FOR HEADER CRC ERROR
5542 4763' JMS CHKRDY /SEE IF DRIVE WAS READY
5543 5760' JMP LOGTRK /DRIVE NOT READY, LOG TRACKING ERROR
5544 4777' LOGHNF, JMS LOGERR /DRIVE WAS READY
5545 0024 HNFERR /DRIVE STATE TABLE INDEX
5546 5770' JMP RETRY /GO TO RETRY ROUTINE

/
5547 1030 CHKHCR, TAD ERREG /GET ERROR REGISTER
5550 0357 AND (4000 /MASK HCRC BIT
5551 7650 SNA CLA /SKIP IF HCRC SET
5552 5306 JMP ABORT /ABORT OPERATION
5553 4763' JMS CHKRDY /CHECK FOR DRIVE READY
5554 5760' JMP LOGTRK /DRIVE NOT READY
5555 5756' JMP LOGHCR /GO TO NEXT PAGE

/
5556 5600
5557 4000
5560 5603
5561 5606
5562 3332
5563 3760
5564 6000
5565 7000
5566 0210
5567 0140
5570 5625
5571 0037
5572 5722
5573 5733
5574 1163
5575 0002
5576 0352
5577 5756
5600

```

```

/
/
5600 4356 LOGHCR, JMS LOGERR /DRIVE READY, LOG HCRC ERROR
5601 0021 HRCER
5602 5225 JMP RETRY /GO TO RETRY ROUTINE
/
/
5603 4356 LOGTRK, JMS LOGERR /LOG TRACKING ERROR
5604 0017 TRKERR
5605 5225 JMP RETRY /GO TO RETRY ROUTINE
/
/
5606 1030 NOTOPI, TAD ERREG /GET ERROR REGISTER
5607 0053 AND K1000 /TEST DATA LATE BIT (DLT)
/
/
5610 7650 CHKDLT, SNA CLA /SKIP IF DLT
5611 5215 JMP CHKCRC /NOT DLT, GO CHECK DATA CRC
/
5612 4356 LOGDLT, JMS LOGERR /LOG DATA LATE ERROR
5613 0022 DLTERR
5614 5225 JMP RETRY /GO TO RETRY ROUTINE
/
/
5615 1030 CHKCRC, TAD ERREG /GET ERROR REGISTER
5616 0377 AND (4000 /TEST FOR DCRC (DATA CRC ERROR)
5617 7650 SNA CLA /SKIP IF DCRC
5620 5776 JMP ABORT /ABORT OPERATION, INTERMITTENT ERROR
5621 4775 JMS CHKRDY /CHECK FOR DRIVE WAS READY
5622 5203 JMP LOGTRK /NOT READY GO LOG TRACKING ERROR
/
5623 4356 LOGDCR, JMS LOGERR /LOG DATA CRC ERROR
5624 0020 DRCER /DRIVE STATE TABLE INDEX
/
5625 2110 RETRY, ISZ RETCNT /INCREMENT RETRY COUNT
5626 7610 SKP CLA /SKIP IF NO OVERFLOW
5627 5774 JMP DRVGN /4096 RETRIES, THAT'S ENOUGH, DROP DRIVE
5630 1110 TAD RETCNT /GET RETRY COUNT
5631 7041 CIA /NEGATE FOR CHECK
5632 1076 TAD RETLIM /ADD RETRY LIMIT
5633 7710 SPA CLA /SKIP IF RETRY < OR = LIMIT
5634 5774 JMP DRVGN /GO DROP DRIVE, RETRY EXCEEDED
5635 4547 SETPNT /SET TBLPNT TO SOFT ERROR COUNT
5636 0013 SFTERR /TBL INDX
5637 1512 TAD I TBLPNT /GET SOFT ERROR COUNT
5640 7041 CIA /NEGATE FOR CHECK
5641 1103 TAD SFTLIM /ADD SOFT ERROR LIMIT
5642 7710 SPA CLA /SKIP IF SOFT ERROR LIMIT NOT EXCEEDED
5643 5774 JMP DRVGN /SOFT ERROR LIMIT EXCEEDED, GO DROP DRIVE
5644 6212 CIF 10 /GOING TO FLD 1
5645 4773 JMS RESALL /RESTORE ALL DRV TABLE ENTRIES
5646 4547 SETPNT /SET TBLPNT TO COMMAND B SENT
5647 0034 XCOMB /TBL INDX
5650 1512 TAD I TBLPNT /GET COMMAND B SENT

```



5651	0044	AND	K7	/MASK FUNCTION CODE BITS
5652	3115	DCA	FUNCOD	/SAVE FUNCTION CODE
5653	1115	TAD	FUNCOD	/GET FUNCTION CODE
5654	7041	CIA		/NEGATE FOR CHECK
5655	1043	TAD	K5	/ADD 5 TO SEE IF READ OR WRITE
5656	7750	SPA SNA	CLA	/SKIP IF FUNCTION 4 OR LESS
5657	5277	JMP	RWRET	/CODE 5 OR 6, GO TO R/W RETRY ROUTINE
5660	1110	TAD	RETCNT	/GET RETRY COUNT
5661	7041	CIA		
5662	1077	TAD	SEKRET	/ADD SEEK TO TRACK RETRY LIMIT
5663	7710	SPA	CLA	/SKIP IF LIMIT NOT EXCEEDED
5664	5774'	JMP	DRVGON	/DROP DRIVE, SEEK LIMIT EXCEEDED
5665	4356	JMS	LOGERR	/LOG SEEK ERROR
5666	0015	SEKERR		/DRIVE STATE TABLE INDEX
5667	4547	SETPNT		/SET TBLPNT TO NEW CYLINDER AND HEAD
5670	0003	NEWCYL		/TBL INDX
5671	1512	TAD I	TBLPNT	/GET NEW CYLINDER AND HEAD
5672	3274	DCA	POSGO	/SAVE FOR SEEK RETRY
5673	4561	SETPOS		/CALL SEEK TO ABSOLUTE TRACK ROUTINE
5674	7402	POSGO, HLT/NEWCYL		/CYL AND HEAD TO SEEK TO
5675	5774'	JMP	DRVGON	/DROP DRIVE, RETRY EXCEEDED
5676	5312	JMP	EREXT1	/SUCCESSFUL, EXIT
/				
5677	1512	RWRET, TAD I	TBLPNT	/GET COMMAND B SENT
5700	0053	AND	K1000	/MASK BIT MODE BIT
5701	3111	DCA	BITMOD	/SAVE AS CURRENT BIT MODE
5702	4560	XFER		/GO DO READ OR WRITE
5703	4556	WAITDN		/WAIT FOR DONE
5704	4772'	JMS	FATAL	/NO DONE FLAG AFTER READ OR WRITE
5705	4554	ERRCHK		/CHECK FOR ERRORS
5706	7610	SKP	CLA	/NO ERROR RETURN
5707	5225	JMP	RETRY	/ERROR RETURN
5710	4771'	JMS	DATCHK	/CHECK DATA IF READ
5711	5225	JMP	RETRY	/DATA ERROR
/				
5712	4356	EREXT1, JMS	LOGERR	/LOG SOFT ERROR
5713	0013	SFTERR		/DRIVE STATE TABLE INDEX
5714	4562	GETSTA		/GET STATUS
5715	4333	JMS	FINRPT	/FINISH REPORT
5716	6212	CIF	10	/GOING TO FLD 1
5717	4773'	JMS	RESALL	/RESTORE FUNCOD, BITMODE, ETC.
5720	4573	RESET		/MAKE RESET LAST OPERATION
5721	2330	ISZ	ERRPC	/INCREMENT FOR GOOD RETURN
5722	1331	EREXT2, TAD	POSPC	/GET POSSET RETURN PC
5723	3770'	DCA	POSSET	/RESTORE RETURN PC
5724	1135	TAD	DRVTMP	/GET CURDRV
5725	3024	DCA	CURDRV	/RESTORE CURRENT DRIVE
5726	3332	DCA	ERRFLG	/CLEAR ERROR FLAG
5727	5730	JMP I	ERRPC	/RETURN
/				
5730	0000	ERRPC, 0		/ERROR RETURN PC OF FIRST CALL
5731	0000	POSPC, 0		/POSSET RETURN PC OF FIRST CALL
5732	0000	ERRFLG, 0		/ERROR FLAG FOR FIRST ENTRY SWITCH
/				

```

/
/*****
/
/ROUTINE TO FINISH STATUS REPORT
/
/      CALLED BY:      JMS      FINRPT
/
/
5733 0000  FINRPT, 0
5734 4545          APTCHK          /CHECK FOR ON APT
5735 5733          JMP I    FINRPT  /ON APT, NO REPORT
5736 4550          GETSWR          /GET SWITCH REGISTER
5737 0046          AND      K100    /CHECK FOR INHIBIT ERROR TYPEOUT
5740 7640          SZA      CLA     /SKIP IF OK TO PRINT ERROR MESSAGE
5741 5733          JMP I    FINRPT  /NO ERROR REPORT, EXIT
5742 4551          STAPRT          /PRINT STATUS
5743 2567          ERTBL2          /START OF MESSAGE ADDRESS IN FLD 1
5744 4551          STAPRT          /PRINT STATUS
5745 2645          ERTBL3
5746 4547          SETPNT          /SET TBLPNT TO DRIVE STATE
5747 0000          DSTATE          /TBL INDX
5750 1512          TAD I    TBLPNT  /GET DRIVE STATE
5751 7640          SZA      CLA     /SKIP IF DRIVE DROPPED
5752 5733          JMP I    FINRPT  /RETURN
5753 4564          MESAG
5754 4345          DROPNG          /"DROPPING DRIVE"
5755 5733          JMP I    FINRPT  /EXIT
/
/
/*****
/
/ROUTINE TO LOG AN ERROR IN DRIVE STATE TABLE
/
/      CALLED BY:      JMS      LOGERR
/      FOLLOWED BY:     HRDERR          /OR ANY DRIVE STATE TABLE INDEX
/
/
5756 0000  LOGERR, 0
5757 7300          CLA      CLL
5760 1756          TAD I    LOGERR  /GET DRIVE STATE TABLE INDEX FROM CALL+1
5761 3363          DCA      ERRPNT  /SAVE AS INDEX FOR SETPNT CALL
5762 4547          SETPNT          /SET TABLE POINT TO DRIVE STATE TABLE ENTRY
5763 7402  ERRPNT, HLT/ERROR INDEX /MODIFIED DRIVE STATE TABLE INDEX FROM CALL+1
5764 2512          ISZ I    TBLPNT  /INCREMENT DRIVE STATE TABLE ERROR ENTRY
5765 7000          NOP            /IN CASE OF OVERFLOW
5766 2356          ISZ      LOGERR  /INCREMENT RETURN PC
5767 5756          JMP I    LOGERR  /EXIT
/
/
5770 4400

```

5771 6000  
5772 6614  
5773 2730  
5774 5413  
5775 3760  
5776 5506  
5777 4000

SEQ 111

6000 \*6000

/\*\*\*\*\*

/ROUTINE TO CHECK DATA AFTER A READ  
 /UPDATE READ OR WRITE OPERATION COUNT IN DRIVE STATE TABLES  
 /RELEASE BUFFER AND UPDATE BITS TRANSFERRED COUNT IF R/W  
 /ELSE RETURN IF LAST OPERATION OF DRIVE USING CONTROLLER  
 /WAS NOT A READ OR WRITE

/ CALLED BY: JMS DATCHK  
 / FOLLOWED BY: ERROR (CALL TO ERROR HANDLER FOR RETRY)  
 / FOLLOWED BY: JMP DRIVE (GET A NEW DRIVE, DRIVE DROPPED AFTER RETRY)  
 / GOOD RETURN (CALL+3)

6000	0000	DATCHK, 0	
6001	6002	IDF	/INTERRUPT OFF
6002	4777	JMS SAVDRV	/GO GET DRV60 AND SAVE CURDRV
6003	1111	TAD BITMOD	/GET BIT MODE OF NEXT DRIVE
6004	3036	DCA BITTMP	/SAVE TEMPORARILY
6005	4547	SETPNT	/SET TBLPNT TO INITIAL BREAK MA
6006	0031	INITCA	/TBL INDX
6007	1512	TAD I TBLPNT	/GET INITIAL BREAK MA
6010	3372	DCA SAVBMA	/SAVE BMA TEMPORARILY
6011	4547	SETPNT	/SET TBLPNT TO WORD COUNT
6012	0030	WRDCNT	/TBL INDX
6013	1512	TAD I TBLPNT	/GET WORD COUNT
6014	3037	DCA SAVWRD	/SAVE WC TEMPORARILY
6015	4776	JMS CHKACT	/CHECK FOR DRIVE ACTIVE
6016	5351	JMP DEX-2	/DRIVE NOT ACTIVE
6017	4547	SETPNT	/SET TBLPNT TO LAST COMMAND B
6020	0034	XCOMB	/TBL INDX
6021	1512	TAD I TBLPNT	/GET LAST COMMAND B
6022	0044	AND K7	/MASK FUNCTION CODE BITS
6023	7041	CIA	
6024	1043	TAD K5	/ADD CODE FOR WRITE
6025	7540	SMA SZA	/SKIP IF LAST OPR WAS A READ OR WRITE
6026	5351	JMP DEX-2	/NOT R/W, EXIT
6027	7650	SNA CLA	/SKIP IF LAST OPR WAS A READ (CODE 6)
6030	5343	JMP WRTUP	/GO PROCESS A WRITE
6031	1512	READOP, TAD I TBLPNT	/GET LAST COMMAND B
6032	0053	AND K1000	/MASK BIT MODE BIT 2
6033	3111	DCA BITMOD	/SAVE AS BIT MODE
6034	1512	TAD I TBLPNT	/GET LAST COMMAND B
6035	0045	AND K70	/MASK EMA FIELD BITS 6,7,8
6036	1123	TAD KCDF	/ADD CDF INSTRUCTION
6037	3121	DCA CDFSAV	/SAVE CDF TO BUFFER FIELD INSTRUCTION
6040	1111	TAD BITMOD	/GET BIT MODE
6041	7640	SZA CLA	/SKIP IF 12 BIT MODE

6042	5246		JMP	MOD8	/GO CHECK 8 BIT MODE DATA
6043	4775	/			
6043	4775	MOD12,	JMS	CHKSEC	/CHECK 12 BIT DATA (1 SECTOR)
6044	5332		JMP	CHKOK	/DATA GOOD RETURN
6045	5353		JMP	DEX	/DATA ERROR RETURN
6046	4547	/			
6046	4547	MOD8,	SETPNT		/SET TBLPNT TO WORD COUNT
6047	0030		WRDCNT		/TBL INDX
6050	1512		TAD I	TBLPNT	/GET WORD COUNT
6051	3371		DCA	WSAV	/SAVE WORD COUNT
6052	1371		TAD	WSAV	/GET WORD COUNT
6053	7700		SMA	CLA	/SKIP IF <= 2048(10)
6054	5307		JMP	MODLPB	/GO CHECK POSITIVE WORD COUNT > 2048(10)
6055	1371	/			
6055	1371	MODLPA,	TAD	WSAV	/GET WORD COUNT
6056	1051		TAD	K400	/ADD 256(10) = ONE 8 BIT SECTOR
6057	3371		DCA	WSAV	/SAVE REMAINING WORD COUNT TO CHECK
6060	1371		TAD	WSAV	/GET REMAINING WORD COUNT
6061	7700		SMA	CLA	/SKIP IF COMPLETE SECTOR TO CHECK
6062	5266		JMP	PARSEC	/GO CHECK PARTIAL SECTOR
6063	1057		TAD	M400	/-256(10) FOR WORD COUNTER ONE FULL SECTOR
6064	3512		DCA I	TBLPNT	/SAVE WORD COUNT TO CHECK
6065	5271		JMP	SECHK	/CHECK SECTOR OF DATA
6066	1371	/			
6066	1371	PARSEC,	TAD	WSAV	/GET REMAINING WORD COUNT
6067	1057		TAD	M400	/PARTIAL SECTOR, RESTORE TO NEGATIVE VALUE
6070	3512		DCA I	TBLPNT	/SAVE WORD COUNT TO CHECK
6071	4775	/			
6071	4775	SECHK,	JMS	CHKSEC	/CHECK WHOLE OR PARTIAL SECTOR
6072	7610		SKP	CLA	/SKIP IF DATA OK
6073	5353		JMP	DEX	/DATA ERROR, EXIT
6074	1371		TAD	WSAV	/GET REMAINING WORDS TO CHECK
6075	7700		SMA	CLA	/SKIP IF MORE TO CHECK
6076	5332		JMP	CHKOK	/DONE CHECKING DATA, GO TO GOOD EXIT
6077	4547		SETPNT		/SET TBLPNT TO BREAK MA
6100	0031		INITCA		/TBL INDX
6101	1512		TAD I	TBLPNT	/GET BMA OF THIS SECTOR
6102	1051		TAD	K400	/ADD 256(10) FOR NEXT SECTOR
6103	3512		DCA I	TBLPNT	/SAVE BREAK MA OF NEXT SECTOR
6104	4547		SETPNT		/SET TBLPNT TO WORD COUNT NEXT SECTOR
6105	0030		WRDCNT		/TBL INDX
6106	5255		JMP	MODLPA	/CONTINUE CHECKING DATA
6107	1371	/			
6107	1371	MODLPB,	TAD	WSAV	/GET WORD COUNT REMAINING TO CHECK
6110	1051		TAD	K400	/ADD 256(10) = ONE SECTOR
6111	3371		DCA	WSAV	/SAVE REMAINING WORD COUNT TO CHECK
6112	1057		TAD	M400	
6113	3512		DCA I	TBLPNT	/WRDCNT = -256(10) = ONE SECTOR TO CHECK
6114	4775		JMS	CHKSEC	/CHECK A COMPLETE SECTOR
6115	7610		SKP	CLA	/SKIP IF CHECK OK
6116	5353		JMP	DEX	/DATA ERROR, EXIT FOR RETRY OR DRIVE DROP
6117	4547		SETPNT		/SET TBLPNT TO BMA
6120	0031		INITCA		/TBL INDX
6121	1512		TAD I	TBLPNT	/GET BMA
6122	1051		TAD	K400	/ADD 256(10) = ONE SECTOR

6123	3512	DCA I	TBLPNT	/SAVE BMA OF NEXT SECTOR
6124	4547	SETPNT		/SET TBLPNT TO WRD COUNT
6125	0030	WRDCNT		/TBL INDX
6126	1371	TAD	WSAV	/GET WORDS LEFT TO CHECK
6127	7700	SMA	CLA	/SKIP IF NOW NEGATIVE (<= 2048(10))
6130	5307	JMP	MODLPB	/GO CHECK NEXT SECTOR (>2048)
6131	5255	JMP	MODLPA	/GO CHECK NEXT SECTOR (<=2048)
/				
6132	4547	CHKOK,	SETPNT	/SET TBLPNT TO WORD COUNT
6133	0030		WRDCNT	/TBL INDX
6134	1037		TAD	/GET WORD COUNT SAVED
6135	3512		DCA I	/SAVE FOR BITS UPDATE
6136	4547		SETPNT	/SET TBLPNT TO UPDATE READ OPERATION COUNT
6137	0026		READS	/TBL INDX
6140	2512		ISZ I	/INCREMENT READ COUNT
6141	7300		CLA	/IN CASE OF OVERFLOW
6142	5347		JMP	/EXIT
/				
6143	4547	WRTUP,	SETPNT	/SET TBLPNT TO UPDATE WRITE OPERATION COUNT
6144	0027		WRTS	/TBL INDX
6145	2512		ISZ I	/INCREMENT WRITE OPERATION COUNT
6146	7300		CLA	/IN CASE OF OVERFLOW
/				
6147	4774'	DATCEX,	JMS	/RELEASE THE BUFFER OF LAST DRIVE
6150	4773'		JMS	/UPDATE BITS TRANSFERRED, NO ERRORS OR RECOVERY OK
6151	2200		ISZ	/INCREMENT FOR GOOD RETURN
6152	2200		ISZ	/CALL+3
/				
6153	7300	DEX,	CLA	CLL
6154	4547		SETPNT	/SET TBLPNT TO BREAK MA
6155	0031		INITCA	/TBL INDX
6156	1372		TAD	/GET BMA SAVED
6157	3512		DCA I	/RESTORE BMA
6160	4547		SETPNT	/SET TBLPNT TO WORD COUNT
6161	0030		WRDCNT	/TBL INDX
6162	1037		TAD	/GET ORIGINAL WORD COUNT
6163	3512		DCA I	/RESTORE WORD COUNT
6164	1036		TAD	/GET ORIGINAL BIT MODE
6165	3111		DCA	/RESTORE BIT MODE
6166	1135		TAD	/GET NEXT DRIVE # SAVED
6167	3024		DCA	/RESTORE NEXT DRIVE #
6170	5600		JMP I	/RETURN
/				
6171	0000	WSAV,	0	/WRDCNT
6172	0000	SAVBMA,	0	/INITCA
/				
/				
/				
6173	1755			
6174	3332			
6175	2512			
6176	1340			
6177	6600			
6200				

PAGE

/\*\*\*\*\*

```

/
/
/ROUTINE TO PRINT STATUS ON ALL ACTIVE DRIVES
/
/      CALLED BY:      JMS      STATUS
/
/
6200 0000      STATUS, 0
6201 7300      CLA      CLL
6202 1024      TAD      CURDRV      /GET CURRENT DRIVE
6203 3135      DCA      DRVTMP      /SAVE TEMPORARILY
6204 3024      DCA      CURDRV      /CLEAR DRIVE #
6205 1056      TAD      M4
6206 3251      DCA      STACNT      /SET UP DRIVE COUNTER TO -4
6207 1251      STATLP, TAD      STACNT      /VERIFY COUNT DOES NOT GO POSITIVE      HP 011
6210 7700      SMA CLA      /SKIP IF COUNT IS OK STILL NEGATIVE      HP 011
6211 5246      JMP      BYSTAT      /COUNT IS POSITIVE LEAVE ROUTINE      HP 011
6212 4571      DOCLRF
6213 4551      STAPRT
6214 3165      ERTBL4
6215 1025      TAD      DRV60
6216 7041      CIA
6217 1024      TAD      CURDRV      /SEE IF DRV STATUS BEING PRINTED NEXT IS LAST DRV USING CONTROLLER
6220 7640      SZA      CLA
6221 5224      JMP      .+3      /SKIP IF IT IS
6222 4551      STAPRT      /ELSE DO NOT PRINT CONTROLLER REGISTERS
6223 2541      ERTBL1      /PRINT STATUS MSG
6224 4551      STAPRT      /MSG ADDRESS
6225 2567      ERTBL2
6226 4777      JMS      CHKACT      /CHECK FOR DRIVE ACTIVE
6227 5235      JMP      DNACTV      /GO PRINT DRIVE NOT ACTIVE
6230 4564      MESAG
6231 4075      DRVMSG      /"DRV "
6232 4564      MESAG
6233 4326      DRVACT      /"ACTV"
6234 5243      JMP      NXT      /GO DO NEXT DRIVE
6235 4564      DNACTV, MESAG
6236 4075      DRVMSG      /"DRV "
6237 4564      MESAG
6240 4332      NOTMSG      /" NOT"
6241 4564      MESAG
6242 4326      DRVACT      /"ACTV"
6243 2024      NXT,      ISZ      CURDRV      /INCR TO NEXT DRIVE
6244 2251      ISZ      STACNT      /INCR DRV COUNT, SKIP WHEN DONE
6245 5207      JMP      STATLP      /GO DO NEXT DRIVE
6246 1135      BYSTAT, TAD      DRVTMP      /GET ORIGINAL DRIVE NUMBER      HP 011
6247 3024      DCA      CURDRV      /RESTORE DRIVE NUMBER
6250 5600      JMP I      STATUS      /RETURN

6251 0000      STACNT, 0      /DRIVE STATUS COUNTER
/
/
/
/
/

```

```

/*****
/
/
/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
/
6252 0000 XC8CAL, 0
6253 6000 SKON /SKIP IF INTERRUPT ON AND TURN OFF
6254 5261 JMP C8OFF /INT OFF
6255 3313 DCA CHRTMP /SAVE AC
6256 1122 TAD KION /GET ION INSTRUCTION
6257 3310 DCA CALOUT /SAVE ION FOR EXIT EXECUTION
6260 5264 JMP GETHCW /GO CHECK FOR CONSOLE ACTIVE
6261 3313 C8OFF, DCA CHRTMP /SAVE AC
6262 1376 TAD (IOF /GET IOF INSTRUCTION
6263 3310 DCA CALOUT /SAVE FOR EXIT EXECUTION
6264 1022 GETHCW, TAD 22 /GET HCW2
6265 0051 AND K400 /TEST FOR BIT 3=1 CONSOLE ACTIVE
6266 7650 SNA CLA /SKIP IF CONSOLE ACTIVE
6267 5310 JMP CALOUT /EXIT, RETURN CALL+1 CONSOLE NOT ACTIVE
6270 1313 TAD CHRTMP /RESTORE AC
6271 1375 TAD (-222 /SEE IF CHAR WAS CTRL R FOR STATUS REPORT
6272 7650 SNA CLA /SKIP IF NOT CTRL R
6273 4200 JMS STATUS /IT WAS CTRL R, GO PRINT STATUS OF DRIVES
6274 6224 RIF /READ INSTRUCTION FIELD
6275 1374 TAD (OFFSET /ADD CONSOLE PACKAGE FIELD OFFSET
6276 1373 TAD (CIF /ADD CIF INSTRUCTION CODE
6277 3300 DCA .+1 /SAVE MODIFIED CIF FOR EXECUTION
6300 7402 HLT/CIF /MODIFIED CIF TO CONSOLE PACKAGE FIELD
6301 1313 TAD CHRTMP /GET CHARACTER AGAIN
6302 4712 JMS I C8LOC /GO TO CONSOLE PACKAGE
6303 2252 ISZ XC8CAL /INCREMENT RETURN ADDRESS
6304 1020 TAD PSR /GET PSR
6305 6211 CDF 10 /CDF TO FLD 1
6306 3772 DCA I (PSR1 /COPY PSR TO FLD 1 PG 0 LOC 20
6307 6201 CDF 00 /CDF TO PRGM FLD
6310 7402 CALOUT, HLT/ION/IOF /MODIFIED ION OR IOF UPON ENTRY
6311 5652 XC8RET, JMP I XC8CAL /RETURN CALL + 2 CONSOLE WAS ACTIVE
/
6312 5222 C8LOC, C8ENTR /POINTER TO CONSOLE PACKAGE ENTRY
6313 0000 CHRTMP, 0 /TEMPORARY AC SAVE AREA
/
/
/*****
/ROUTINE TO DO A SEEK
/CALLED BY: JMS SEEK
/
/

```



```

6314 0000  SEEK,  0
6315 6002      IOF
6316 4547      SETPNT
6317 0036      POSFLG
6320 7240      CLA CMA
6321 3512      DCA I   TBLPNT
6322 4547      SETPNT
6323 0037      RDYFLG
6324 7240      CLA CMA
6325 3512      DCA I   TBLPNT
6326 4547      SETPNT
6327 0034      XCOMB
6330 1024      TAD     CURDRV
6331 7002      BSW
6332 1371      TAD     (403
6333 3512      DCA I   TBLPNT
6334 4547      SETPNT
6335 0002      CURCYL
6336 1512      TAD I   TBLPNT
6337 4547      SETPNT
6340 0001      OLDCYL
6341 3512      DCA I   TBLPNT
6342 4547      SETPNT
6343 0032      SECADD
6344 3512      DCA I   TBLPNT
6345 4547      SETPNT
6346 0040      XENDSC
6347 3512      DCA I   TBLPNT
6350 4555      GO
6351 5714      JMP I   SEEK

```

/INTERRUPTS OFF  
 /SET TBLPNT TO POSITION VERIFICATION NEEDED FLAG  
  
 /SET POS VERIFY FLAG TO 7777  
 /SET TBLPNT TO READY PENDING FLAG  
 /TBL INDX  
 /AC=7777  
 /SET DRIVE SEEKING (READY PENDING FLG)  
 /SET TBLPNT TO COMMAND B  
 /TABLE INDEX  
 /GET CURRENT DRIVE  
 /MOVE DRIVE SELECT BITS TO BITS 4,5  
 /SET INTERRUPT ENABLE , CODE 3 FOR SEEK  
 /SAVE COMMAND B  
 /SET TBLPNT TO CURRENT CYL AND HEAD  
 /TABLE INDEX  
 /GET CURRENT CYL AND HEAD  
 /SET TBLPNT TO OLD CYL ADDR STORAGE  
 /TABLE INDEX  
 /SAVE CURRENT CYL AND HEAD  
 /SET TBLPNT TO SECTOR ADDR  
 /TABLE INDEX  
 /CLEAR SECTOR ADDR  
 /SET TBLPNT TO EXPECTED FINAL SECTOR ADDRESS  
 /TBL INDX  
 /EXPECTED FINAL SECTOR ADDRESS = 0  
 /ISSUE COMMAND  
 /EXIT

```

6352 0000  HDRERR, 0
6353 4545      APTCHK
6354 4557      APTERR
6355 4564      MESAG
6356 3731      HDRMSG
6357 4564      MESAG
6360 4057      ERR
6361 4571      DOCLRF
6362 4551      STAPRT
6363 3165      ERTBL4
6364 4551      STAPRT
6365 2541      ERTBL1
6366 5752      JMP I   HDRERR

```

/CHECK FOR ON APT  
 /BAD HEADER ON PACK  
  
 /"HEADER"  
  
 /" ERR"  
 /<CR LF>  
 /PRINT DRIVE NUMBER  
 /START OF MESSAGE ADDRESS  
 /PRINT CONTROLLER REGISTERS  
 /ADDRESS OF START OF MESSAGES  
 /RETURN

6371 0403  
 6372 0020  
 6373 6202  
 6374 0010  
 6375 7556  
 6376 6002  
 6377 1340  
 6400

PAGE

/ \*\*\*\*\*

/

/ REMOVED CRLFDO ROUTINE FROM HERE AND PLACED IT ON NEXT PAGE.

HP 012

/

/ \*\*\*\*\*

/

/ROUTINE TO WAIT FOR DRIVE READY AFTER SEEK

/

/CALLED BY: JMS WATRDY

/

/RETURN CALL+1 IF READY

/RETURN CALL+2 IF NOT READY

/

6400 0000

WATRDY, 0

6401 7300

CLA CLL

6402 3222

DCA WATCNT

/CLEAR DRIVE READY WAIT COUNTER

6403 1377

TAD (-12

/GET TIMES 10 MULTIPLIER

6404 3223

DCA WATCT1

/SET UP TIMES 10 COUNTER

6405 6610

WATLPA, RRER

/READ ERROR REGISTER

6406 7410

SKP

/IOT SHOULD NOT SKIP

6407 4776

JMS FATAL

/IOT RRER SKIPPED

6410 7110

CLL RAR

/PUT DRIVE READY BIT IN LINK

6411 7620

SNL CLA

/SKIP IF DRIVE READY

6412 5214

JMP WATMOR

/DRIVE NOT READY, CONTINUE WAITING

6413 5600

JMP I WATRDY

/DRIVE READY RETURN

/

6414 2222

WATMOR, ISZ WATCNT

/INCREMENT DRIVE READY WAIT COUNTER

6415 5205

JMP WATLPA

/CONTINUE WAITING FOR DRIVE READY

6416 2223

ISZ WATCT1

/INCREMENT DRIVE READY WAIT COUNTER TIMES 10

6417 5205

JMP WATLPA

/CONTINUE WAITING FOR DRIVE READY

6420 2200

ISZ WATRDY

/INCREMENT RETURN FOR DRIVE NOT READY TIME-OUT

6421 5600

JMP I WATRDY

/DRIVE NOT READY TIME-OUT RETURN

/

6422 0000

WATCNT, 0

/DRIVE READY WAIT COUNTER

6423 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 NOT ON APT AND 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
/
/
6424 0000 DERR, 0
6425 2133 ISZ   BDCNT /INCREMENT BAD WORD COUNT
6426 7000 NOP    /IN CASE OF OVERFLOW
6427 1134 TAD    DERFLG /GET DATA ERROR FLAG
6430 7640 SZA    CLA    /SKIP IF THIS IS THE FIRST BAD WORD
6431 5265 JMP    PRNBA /GO PRINT BUFFER ADDR, NOT FIRST BAD WRD
6432 7040 CMA    /AC=7777
6433 3134 DCA    DERFLG /SET DATA ERROR FLG TO 7777
6434 4545 APTCHK
6435 5624 JMP I   DERR    /ON APT NO DATA ERROR MSG
6436 4550 GETSWR /GET SWITCHES
6437 0046 AND    K100 /MASK BIT 5 (INHIBIT ERROR MSG)
6440 7640 SZA CLA /SKIP IF ERROR PRINTOUT OK
6441 5624 JMP I   DERR /EXIT NO ERROR MSG ALLOW
6442 4571 DOCRLF /<CR LF>
6443 4564 MESAG
6444 3735 DATMSG /"DATA"
6445 4564 MESAG
6446 4057 ERR    /" ERR"
6447 4571 DOCRLF /<CR LF>
6450 4551 STAPRT /PRINT THE DRIVE NUMBER
6451 3165 ERTBL4 /START OF MSG ADDRESS
6452 4775 JMS    LOGERR /LOG DATA ERROR
6453 0016 DATERR /TBL INDX
6454 1060 TAD    DMPFLG /GET DATA DUMP FLAG
6455 7640 SZA    CLA /SKIP IF OK TO PRINT BAD DATA WORDS
6456 5624 JMP I   DERR /NO DATA DUMP, EXIT
6457 4564 MESAG
6460 3740 BAGBM /"BA: GOOD BAD"
6461 4571 DOCRLF /<CR LF>
6462 1025 TAD    DRV60 /GET DRIVE NUMBER
6463 1374 TAD    (BUFLD0 /ADD ADDRESS OF BUFFER TABLE
6464 3335 DCA    BFPT /SET UP BUFFER TABLE POINTER
6465 4545 PRNBA, APTCHK /CHECK FOR ON APT
6466 5624 JMP I   DERR /ON APT, NO REPORT TO TTY
6467 4550 GETSWR /GET SWITCHES
6470 0046 AND    K100 /TEST FOR INHIBIT ERROR MESSAGE
6471 7640 SZA    CLA /SKIP IF OK TO PRINT ERROR MESSAGE
6472 5624 JMP I   DERR /NO MESSAGE, EXIT
6473 7240 STA    /AC =7777
6474 3046 DCA    INMODE /ALLOW CONTROL S WHILE PRINTING DATA ERROR
6475 1735 TAD I   BFPT /GET BUFFER NUMBER (FIELD)
6476 1047 TAD    K260 /ADD ASCII BASE CODE
6477 4566 PRNT /PRINT FIRST DIGIT AS BUFFER FIELD
6500 1015 TAD    AUTO15 /GET BUFFER ADDRESS OF BAD DATA
6501 4570 PRNAC /PRINT BUFFER ADDRESS
6502 4572 SPACE2 /PRINT 2 SPACES
6503 1015 TAD    AUTO15 /GET BUFFER ADDRESS POINTER

```

HP 013  
HP 013

```

6504 1055      TAD      M1          /SUBTRACT 1 FOR AUTO INDX
6505 3015      DCA      AUTO15      /SAVE BUFFER ADDRESS POINTER
6506 1014      TAD      AUTO14      /GET DATA PATTERN POINTER
6507 1055      TAD      M1          /SUBTRACT 1 FOR AUTO INDX
6510 3014      DCA      AUTO14      /SAVE GOOD DATA POINTER
6511 6211      CDF      10          /GOOD DATA IN FLD 1
6512 1414      TAD I    AUTO14      /GET GOOD DATA WORD
6513 6201      CDF      00
6514 4570      PRNAC                     /PRINT GOOD DATA WORD
6515 4572      SPACE2                    /PRINT 2 SPACES
6516 1121      TAD      CDFSAV        /GET CDF TO BUFFER FLD
6517 3320      DCA      BUFCDF        /SAVE FOR EXECUTION

/
6520 7402      BUFCDF, HLT/CDF
6521 1111      TAD      BITMOD        /CDF TO BUFFER FLD
6522 7650      SNA      CLA          /GET BIT MODE
6523 5327      JMP      .+4          /SKIP IF 8 BIT MODE
6524 1415      TAD I    AUTO15      /12 BIT MODE DATA
6525 0050      AND      K377        /GET BAD DATA WORD
6526 7410      SKP                     /MASK FOR UNUSED BITS
6527 1415      TAD I    AUTO15      /GET BAD DATA WORD
6530 6201      CDF      00
6531 4570      PRNAC                     /PRINT BAD DATA WORD
6532 4571      DOCRLF                    /DO A <CR> <LF>
6533 3046      DCA      INMODE        /CLEAR FLAG TO INDICATE NOT IN INPUT MODE
6534 5624      JMP I    DERR        /RETURN

6535 0000      BFPT, 0              /BUFFER TABLE POINTER

/
/ROUTINE TO CLEAR BUFFER 1 FIELD 1
/
/CALLED BY:      JMS      CLRBF1
/
/
6536 0000      CLRBF1, 0
6537 1373      TAD      (BUFSZ1      /GET BUFFER 1 SIZE
6540 7041      CIA
6541 3353      DCA      BUFCTR        /SET UP BUFFER LOCATION COUNTER
6542 1117      TAD      BUFAD1        /GET START OF BUFFER 1 ADDRESS
6543 1055      TAD      M1          /SUB ONE FOR AUTO-INDEX
6544 3012      DCA      AUTO12        /SET UP BUFFER POINTER
6545 6211      CDF      10          /CDF TO BUFFER 1 FLD 1
6546 3412      CLRBF1, DCA I    AUTO12 /CLEAR A BUFFER 1 LOCATION
6547 2353      ISZ      BUFCTR        /INCREMENT BUFFER LOCATION COUNTER
6550 5346      JMP      CLRBF1        /CONTINUE CLEARING BUFFER 1
6551 6201      CDF      00          /CDF TO PRGM FLD
6552 5736      JMP I    CLRBF1        /RETURN

/
6553 0000      BUFCTR, 0            /BUFFER LOCATION COUNTER
/
/
/*****
/
/ROUTINE TO CLEAR BUFFER 0 FIELD 0

```

HP 013

```

/
/CALLED BY:      JMS      CLRBFO
/
/
6554 0000  CLRBFO, 0
6555 1372      TAD      (BUFSZ0      /GET BUFFER 0 SIZE
6556 7041      CIA
6557 3353      DCA      BUFCTR      /SET UP BUFFER LOCATION COUNTER
6560 1116      TAD      BUFAD0      /GET ADDRESS OF START OF BUFFER 0
6561 1055      TAD      M1          /SUB ONE FOR AUTO-INDEX
6562 3012      DCA      AUTO12      /SET UP BUFFER POINTER
/
6563 3412  CLRLP0, DCA I  AUTO12      /CLEAR A BUFFER 0 LOCATION
6564 2353      ISZ      BUFCTR      /INCREMENT BUFFER LOC COUNTER
6565 5363      JMP      CLRLP0      /CONTINUE CLEARING BUFFER 0
6566 5754      JMP I  CLRBFO      /RETURN
/
/
/
6572 0400
6573 2000
6574 0066
6575 5756
6576 6614
6577 7766
6600
PAGE
/
/
/*****
/
/
/ROUTINE TO SAVE "CURDRV" AND GET "DRV60" AS CURRENT DRIVE
/
/      CALLED BY:      JMS      SAVDRV
/
/
6600 0000  SAVDRV, 0
6601 7300      CLA CLL
6602 1024      TAD      CURDRV
6603 3135      DCA      DRVTMP      /SAVE SOFTWARE "CURDRV"
6604 1025      TAD      DRV60      /GET DRIVE LAST USING CONTROLLER
6605 3024      DCA      CURDRV      /SAVE AS SOFTWARE "CURDRV"
6606 5600      JMP I  SAVDRV
/
/
/
/*****
/
6607 0000  OCTGET, 0
6610 6212      CIF      10          /INS FLD = 1
6611 4613      JMS I  OCTGO      /GO TO INPUT ONE OCTAL DIGIT FLD 1
6612 5607      JMP I  OCTGET      /RETURN
/
/
6613 1664  OCTGO, XOCT1          /ADDRESS OF SUBROUTINE

```

```

/*****
/
/
/*****
/ROUTINE TO HANDLE FATAL ERRORS
/
/      CALLED BY:      JMS      FATAL
/
/
6614 0000  FATAL, 0
6615 6002      IOF                      /INTERRUPT OFF
6616 3142      DCA      ASAV             /SAVE AC
6617 4545      APTCHK                    /CHECK FOR ON APT
6620 4557      APTERR                    /GO TO APT ERROR HANDLER
6621 4564      MESAG                     /PRINT MESSAGE
6622 3721      FATL                      /"FATAL"
6623 4564      MESAG
6624 4057      ERR                      /" ERR"
6625 4571      DOCR LF                   /<CR LF>
6626 4564      MESAG
6627 3716      PCMSG                     /"PC: "
6630 1214      TAD      FATAL            /GET PC+1
6631 1055      TAD      M1              /NOW HAVE PC
6632 4570      PRNAC                     /PRINT PC
6633 4572      SPACE2                    /2 SPACES
6634 4564      MESAG
6635 3750      ACMMSG                     /"AC: "
6636 1142      TAD      ASAV            /GET AC
6637 4570      PRNAC                     /PRINT AC
6640 4546      C8CALL                    /GO TO CONSOLE PACKAGE FOR SR=
6641 7000      NOP                      /NOT ACTIVE RETURN
6642 4562      GETSTA                    /GET STATUS FOR LAST OPERATION CONTINUE PROGRAM
6643 5644      JMP I    DRVGO            /PROGRAM CONTINUED, GO TO EXERCISER LOOP

/
6644 0327  DRVGO, DRIVE                /ADDRESS OF EXERCISER LOOP
/
/
/*****
/
/ROUTINE TO DO A DATA TRANSFER
/READ,WRITE,READ DATA NO HEADER CHECK
/CALLED BY:      XFER
/FUNCTION CODE MUST BE IN LOCATION "FUNCOD" AND MUST BE
/0005 (WRITE), 0006 (READ)
/
6645 0000  EXFER, 0

```

```

6646 7300      CLA      CLL
6647 6002      IOF              /INTERRUPTS OFF
6650 4547      SETPNT          /SET TBLPNT TO CURRENT CYL AND HD
6651 0002      CURCYL          /TABLE INDEX
6652 1512      TAD I    TBLPNT /GET CURRENT CYL AND HD
6653 4547      SETPNT          /SET TBLPNT TO COMMAND A
6654 0033      XCOMA          /TABLE INDEX
6655 3512      DCA I    TBLPNT /SAVE COMMAND A
6656 4547      SETPNT          /SET TBLPNT TO COMMAND B
6657 0034      XCOMB          /TABLE INDEX
6660 1024      TAD      CURDRV /GET CURRENT DRIVE AS BUFFER TABLE INDEX
6661 1302      TAD      BUFA0   /ADD ADDR OF BUFFER TABLE
6662 3301      DCA      FLDP1   /SAVE BUFFER TABLE POINTER
6663 1024      TAD      CURDRV /GET CURRENT DRIVE
6664 7002      BSW              /MOVE TO BITS 4,5
6665 3300      DCA      DTMP1   /SAVE SHIFTED DRIVE NUMBER
6666 1701      TAD I    FLDP1   /GET BUFFER FIELD
6667 7106      CLL RTL
6670 7004      RAL              /MOVE TO BITS 6:8
6671 1111      TAD      BITMOD  /ADD BIT MODE
6672 1300      TAD      DTMP1   /ADD DRIVE NUMBER
6673 1051      TAD      K400    /SET INTERRUPT ENABLE BIT 3
6674 1115      TAD      FUNCOD  /ADD FUNCTION CODE
6675 3512      DCA I    TBLPNT  /SAVE COMMAND B
6676 4555      GO              /ISSUE COMMAND
6677 5645      JMP I    EXFER    /EXIT

```

```

/
6700 0000      DTMP1, 0        /TEMP STORAGE OF SHIFTED DRIVE NUMBER
6701 0000      FLDP1, 0        /TEMP STORAGE OF BUFFER FIELD POINTER
/
6702 0066      BUFA0, BUFLD0    /ADDRESS OF BUFFER TABLE
/

```

```

/*****
/
/

```

```

/ROUTINE TO DO A <CR> AND <LF>
/

```

```

/      CALLED BY:      DOCRLF
/
/

```

```

6703 0000      CRLFDO, 0
6704 6212      CIF      10      /INS FLD = 1
6705 4707      JMS I    XXCRLF   /GO TO FLD 1
6706 5703      JMP I    CRLFDO   /RETURN
6707 2057      XXCRLF, XCRLF     /LOCATION OF CRLF ON FIELD 1

```

```

HP 012
HP 012
HP 012

```

```

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

```

```

/*****
/
/
/      D R I V E   S T A T E   T A B L E S
/
/
6710      DRVPNT=.      /START OF INDEX TABLE INTO DRIVE STATE TABLES
/
6710 6714      DRV0,      DRIVE0      /POINTER TO DRIVE 0 DRIVE STATE TABLE
6711 6756      DRV1,      DRIVE1      /POINTER TO DRIVE 1 DRIVE STATE TABLE
6712 7020      DRV2,      DRIVE2      /POINTER TO DRIVE 2 DRIVE STATE TABLE
6713 7062      DRV3,      DRIVE3      /POINTER TO DRIVE 3 DRIVE STATE TABLE
/
/
6714      DRIVE0=.      /START OF DRIVE 0 DRIVE STATE TABLE
/
6714 0000      DSTATE      /DRIVE STATE 0=NON-ACTIVE 7777=ACTIVE
6715 0001      OLD CYL      /PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
6716 0002      CUR CYL      /CURRENT CYLINDER ADDRESS BY READ HEADER
6717 0003      NEW CYL      /NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
6720 0004      BITS1      /BITS TRANSFERRED LOW ORDER
6721 0005      BITS2      /BITS TRANSFERRED WORD 2
6722 0006      BITS3      /BITS TRANSFERRED HIGH ORDER
6723 0007      BITS4      /MULTIPLIER TIMES 10 BILLION
6724 0010      SEEKS1      /SEEK COUNT
6725 0011      SEEKS2      /SEEK COUNT MULTIPLIER TIMES 1000
6726 0012      HRDERR      /HARD ERROR COUNT
6727 0013      SFTERR      /SOFT ERROR COUNT
6730 0014      DRVERR      /DRIVE ERROR COUNT
6731 0015      SEKERR      /SEEK ERROR COUNT
6732 0016      DATERR      /DATA ERROR COUNT
6733 0017      TRKERR      /TRACKING ERROR COUNT
6734 0020      DCR CER      /DATA CRC ERROR COUNT
6735 0021      HCR CER      /HEADER CRC ERROR COUNT
6736 0022      DL TERR      /DATA LATE ERROR COUNT
6737 0023      OPIERR      /OPERATION INCOMPLETE ERROR COUNT
6740 0024      HNFERR      /HEADER NOT FOUND ERROR COUNT
6741 0025      CTLERR      /CONTROLLER ERROR COUNT
6742 0026      READS      /NUMBER OF READ OPERATIONS
6743 0027      WRTS      /NUMBER OF WRT OPERATIONS
6744 0030      WRDCNT      /INITIAL WORD COUNT SENT
6745 0031      INITCA      /INITIAL CURRENT DATA BUFFER ADDRESS
6746 0032      SECADD      /SECTOR ADDRESS SENT
6747 0033      XCOMA      /COMMAND REGISTER A SENT
6750 0034      XCOMB      /COMMAND REGISTER B SENT
6751 0035      DACNT      /DRIVE ACCESS ATTEMPT COUNT (DRIVE NOT READY)
6752 0036      POSFLG      /POSITION VERIFICATION NEEDED FLAG
6753 0037      RDYFLG      /7777=DRIVE READY PENDING DUE TO SEEK 0=NOT SEEKING
6754 0040      XENDSC      /EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
6755 0041      DRVBSY      /7777=DRIVE BUSY R/W 0=DRIVE NOT BUSY R/W
/
/
6756      DRIVE1=.      /START OF DRIVE 1 DRIVE STATE TABLE
/
6756 0000      DSTATE      /DRIVE STATE 0=NON-ACTIVE 7777=ACTIVE

```



6757	0001	OLDCYL	/PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
6760	0002	CURCYL	/CURRENT CYLINDER ADDRESS BY READ HEADER
6761	0003	NEWCYL	/NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
6762	0004	BITS1	/BITS TRANSFERRED LOW ORDER
6763	0005	BITS2	/BITS TRANSFERRED WORD 2
6764	0006	BITS3	/BITS TRANSFERRED HIGH ORDER
6765	0007	BITS4	/MULTIPLIER TIMES 10 BILLION
6766	0010	SEEKS1	/SEEK COUNT
6767	0011	SEEKS2	/SEEK COUNT MULTIPLIER TIMES 1000
6770	0012	HRDERR	/HARD ERROR COUNT
6771	0013	SFTERR	/SOFT ERROR COUNT
6772	0014	DRVERR	/DRIVE ERROR COUNT
6773	0015	SEKERR	/SEEK ERROR COUNT
6774	0016	DATERR	/DATA ERROR COUNT
6775	0017	TRKERR	/TRACKING ERROR COUNT
6776	0020	DCRCER	/DATA CRC ERROR COUNT
6777	0021	HRCER	/HEADER CRC ERROR COUNT
7000	0022	DLTERR	/DATA LATE ERROR COUNT
7001	0023	OPIERR	/OPERATION INCOMPLETE ERROR COUNT
7002	0024	HNFERR	/HEADER NOT FOUND ERROR COUNT
7003	0025	CTLERR	/CONTROLLER ERROR COUNT
7004	0026	READS	/NUMBER OF READ OPERATIONS
7005	0027	WRTS	/NUMBER OF WRT OPERATIONS
7006	0030	WRDCNT	/INITIAL WORD COUNT SENT
7007	0031	INITCA	/INITIAL CURRENT DATA BUFFER ADDRESS
7010	0032	SECADD	/SECTOR ADDRESS SENT
7011	0033	XCOMA	/COMMAND REGISTER A SENT
7012	0034	XCOMB	/COMMAND REGISTER B SENT
7013	0035	DACNT	/DRIVE ACCESS ATTEMPT COUNT (DRIVE NOT READY)
7014	0036	POSFLG	/POSITION VERIFICATION NEEDED FLAG
7015	0037	RDYFLG	/7777=DRIVE READY PENDING DUE TO SEEK 0=NOT SEEKING
7016	0040	XENDSC	/EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
7017	0041	DRVBSY	/7777=DRIVE BUSY R/W 0=DRIVE NOT BUSY R/W
		/	
		/	
7020		DRIVE2=.	/START OF DRIVE 2 STATE TABLE
		/	
7020	0000	DSTATE	/DRIVE STATE 0=NON-ACTIVE 7777=ACTIVE
7021	0001	OLDCYL	/PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7022	0002	CURCYL	/CURRENT CYLINDER ADDRESS BY READ HEADER
7023	0003	NEWCYL	/NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7024	0004	BITS1	/BITS TRANSFERRED LOW ORDER
7025	0005	BITS2	/BITS TRANSFERRED WORD 2
7026	0006	BITS3	/BITS TRANSFERRED HIGH ORDER
7027	0007	BITS4	/MULTIPLIER TIMES 10 BILLION
7030	0010	SEEKS1	/SEEK COUNT
7031	0011	SEEKS2	/SEEK COUNT MULTIPLIER TIMES 1000
7032	0012	HRDERR	/HARD ERROR COUNT
7033	0013	SFTERR	/SOFT ERROR COUNT
7034	0014	DRVERR	/DRIVE ERROR COUNT
7035	0015	SEKERR	/SEEK ERROR COUNT
7036	0016	DATERR	/DATA ERROR COUNT
7037	0017	TRKERR	/TRACKING ERROR COUNT
7040	0020	DCRCER	/DATA CRC ERROR COUNT
7041	0021	HRCER	/HEADER CRC ERROR COUNT

7042	0022	DLTERR	/DATA LATE ERROR COUNT
7043	0023	OPIERR	/OPERATION INCOMPLETE ERROR COUNT
7044	0024	HNFFERR	/HEADER NOT FOUND ERROR COUNT
7045	0025	CTLERR	/CONTROLLER ERROR COUNT
7046	0026	READS	/NUMBER OF READ OPERATIONS
7047	0027	WRTS	/NUMBER OF WRT OPERATIONS
7050	0030	WRDCNT	/INITIAL WORD COUNT SENT
7051	0031	INITCA	/INITIAL CURRENT DATA BUFFER ADDRESS
7052	0032	SECADD	/SECTOR ADDRESS SENT
7053	0033	XCOMA	/COMMAND REGISTER A SENT
7054	0034	XCOMB	/COMMAND REGISTER B SENT
7055	0035	DACNT	/DRIVE ACCESS ATTEMPT COUNT (DRIVE NOT READY)
7056	0036	POSFLG	/POSITION VERIFICATION NEEDED FLAG
7057	0037	RDYFLG	/7777=DRIVE READY PENDING DUE TO SEEK 0=NOT SEEKING
7060	0040	XENDSC	/EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
7061	0041	DRVBSY	/7777=DRIVE BUSY R/W 0=DRIVE NOT BUSY R/W
		/	
		/	
	7062	DRIVE3=.	/START OF DRIVE 3 DRIVE STATE TABLE
		/	
7062	0000	DSTATE	/DRIVE STATE 0=NON-ACTIVE 7777=ACTIVE
7063	0001	OLDCYL	/PREVIOUS CYLINDER ADDRESS PRIOR TO SEEK
7064	0002	CURCYL	/CURRENT CYLINDER ADDRESS BY READ HEADER
7065	0003	NEWCYL	/NEW CYLINDER ADDRESS (RANDOM OR CONSTANT)
7066	0004	BITS1	/BITS TRANSFERRED LOW ORDER
7067	0005	BITS2	/BITS TRANSFERRED WORD 2
7070	0006	BITS3	/BITS TRANSFERRED HIGH ORDER
7071	0007	BITS4	/MULTIPLIER TIMES 10 BILLION
7072	0010	SEEKS1	/SEEK COUNT
7073	0011	SEEKS2	/SEEK COUNT MULTIPLIER TIMES 1000
7074	0012	HRDERR	/HARD ERROR COUNT
7075	0013	SFTERR	/SOFT ERROR COUNT
7076	0014	DRVERR	/DRIVE ERROR COUNT
7077	0015	SEKERR	/SEEK ERROR COUNT
7100	0016	DATERR	/DATA ERROR COUNT
7101	0017	TRKERR	/TRACKING ERROR COUNT
7102	0020	DCRCER	/DATA CRC ERROR COUNT
7103	0021	HRCRCER	/HEADER CRC ERROR COUNT
7104	0022	DLTERR	/DATA LATE ERROR COUNT
7105	0023	OPIERR	/OPERATION INCOMPLETE ERROR COUNT
7106	0024	HNFFERR	/HEADER NOT FOUND ERROR COUNT
7107	0025	CTLERR	/CONTROLLER ERROR COUNT
7110	0026	READS	/NUMBER OF READ OPERATIONS
7111	0027	WRTS	/NUMBER OF WRT OPERATIONS
7112	0030	WRDCNT	/INITIAL WORD COUNT SENT
7113	0031	INITCA	/INITIAL CURRENT DATA BUFFER ADDRESS
7114	0032	SECADD	/SECTOR ADDRESS SENT
7115	0033	XCOMA	/COMMAND REGISTER A SENT
7116	0034	XCOMB	/COMMAND REGISTER B SENT
7117	0035	DACNT	/DRIVE ACCESS ATTEMPT COUNT (DRIVE NOT READY)
7120	0036	POSFLG	/POSITION VERIFICATION NEEDED FLAG
7121	0037	RDYFLG	/7777=DRIVE READY PENDING DUE TO SEEK 0=NOT SEEKING
7122	0040	XENDSC	/EXPECTED FINAL SECTOR ADDRESS REGISTER AFTER R/W
7123	0041	DRVBSY	/7777=DRIVE BUSY R/W 0=DRIVE NOT BUSY R/W
		/	

```

/
7123  DRVEND=-1      /END OF DRIVE STATE TABLES
/
/*****
/
/POINTERS TO BAD SECTOR TABLE
/
7124  3400  BDSEC0, BADSEC      /DRIVE 0
7125  3440  BDSEC1, BADSEC+40    /DRIVE 1
7126  3500  BDSEC2, BADSEC+100   /DRIVE 2
7127  3540  BDSEC3, BADSEC+140   /DRIVE 3
/
/*****
/
/      DATA PATTERN INDEX TABLE
/
7130  DPAT8=.
/
7130  4636  DPAT80, PAT80    /DATA PATTERN 0 ADDRESS (8 BIT MODE)
7131  4676  DPAT81, PAT81    /DATA PATTERN 1 ADDRESS (8 BIT MODE)
7132  4736  DPAT82, PAT82    /DATA PATTERN 2 ADDRESS (8 BIT MODE)
7133  4776  DPAT83, PAT83    /DATA PATTERN 3 ADDRESS (8 BIT MODE)
/
7134  DPAT12=.
/
7134  5036  DP120, PAT120    /DATA PATTERN 0 ADDRESS (12 BIT MODE)
7135  5064  DP121, PAT121    /DATA PATTERN 1 ADDRESS (12 BIT MODE)
7136  5112  DP122, PAT122    /DATA PATTERN 2 ADDRESS (12 BIT MODE)
7137  5140  DP123, PAT123    /DATA PATTERN 3 ADDRESS (12 BIT MODE)
/
/
/*****
/
/      DEVICE CODE TABLES
/
/ADDRESSES OF ALL IOTS IN PROGRAM
/
/
7140  DCTB60=.      /START OF DSC 0600 OR 0620 TABLE
/
7140  5107  CNTLOD      /LOAD WORD COUNT IN ROUTINE "XGO"
7141  5115  BRKLOD      /LOAD BRKMA IN ROUTINE "XGO"
7142  5124  SECL0D      /LOAD SECTOR ADDR IN ROUTINE "XGO"
7143  5132  ALODE       /LOAD COMMAND A REG. IN ROUTINE "XGO"
7144  5140  BLODE       /LOAD COMMAND B REG. IN ROUTINE "XGO"
7145  4042  FLGS        /SKIP ON DONE FLAG IN INTSVC
/

```

```

7146 DCTB61=. /START OF DSC 0610 OR 0630 TABLE
/
7146 4045 IOTA /SKIP ON ERROR FLG IN INTSVC
7147 4054 IOTB /READ ERROR REG IN INTSVC
7150 4060 IOTC /READ WC IN INTSVC
7151 4064 IOTD /READ COM A IN INTSVC
7152 4070 IOTE /READ COM B IN INTSVC
7153 4074 IOTF /READ SECTOR ADR REG IN INTSVC
7154 4350 IOTG /READ ERROR REGISTER IN RDYCHK
7155 5257 RDSI1 /READ SILO IN READ HEADER ROUTINE
7156 5262 RDSI2 /SAME
7157 5265 RDSI3 /SAME
7160 5271 RDSI4 /SAME
7161 5230 RDSIA /READ SILO IN GET STATUS ROUTINE
7162 5235 RDSIB /SAME
7163 6405 WATLPA /RRER IN WATRDY ROUTINE
/
7164 DCEND=.
/
/*****
/
/FACTORY BAD SECTOR FILE SECTOR NUMBERS
/
7164 FACBAD=.
7164 0000 0000 /SECTOR ADDRESS 0 OF FACTORY BAD
7165 0004 0004 /SECTOR ADDRESS 4 OF FACTORY BAD
7166 0010 0010 /SECTOR ADDRESS 8 OF FACTORY BAD
7167 0014 0014 /SECTOR ADDRESS 12 OF FACTORY BAD
7170 0020 0020 /SECTOR ADDRESS 16 OF FACTORY BAD
/
/FIELD BAD SECTOR FILE SECTOR NUMBERS
/
7171 FLDBAD=.
7171 0024 0024 /SECTOR ADDRESS 20 OF FIELD BAD
7172 0030 0030 /SECTOR ADDRESS 24 OF FIELD BAD
7173 0034 0034 /SECTOR ADDRESS 28 OF FIELD BAD
7174 0040 0040 /SECTOR ADDRESS 32 OF FIELD BAD
7175 0044 0044 /SECTOR ADDRESS 36 OF FIELD BAD
/
/*****
/
/
7200 *BUFO /DATA BUFFER FIELD ZERO
/
7200 0000 ZBLOCK BUFSZO
/
/
7577 BUFENO=-1 /END OF FIELD ZERO BUFFER
/
/*****
/
/
0001 FIELD 1

```

[illegible]



```

0010      0010      *10
0010      0000      /
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      0020      /
0020      0000      *20
0021      0000      /
0021      0000      PSR1, 0      /SOFTWARE SWITCH REGISTER
0022      0000      HDW1, 0      /HARDWARE CONFIG WORD 1
0022      0000      HDW2, 0      /HARDWARE CONFIG WORD 2
0023      0000      HDW3, 0      /HARDWARE CONFIG WORD 3
0024      0024      *24
0024      4424      MESSAGE=JMS I .      /CALL TO MESSAGE PRINT ROUTINE
0024      2000      MESPRT      /FOLLOWED BY MESSAGE ADDRESS
0025      4425      YESRNO=JMS I .      /CALL TO YES OR NO ROUTINE
0025      2073      XYESNO      /RETURN CALL+1 IF YES, CALL+2 IF NO
0026      4426      OCT1=JMS I .      /CALL TO ROUTINE TO ACCEPT 1 OCTAL DIGIT INPUT
0026      1664      XOCT1
0027      4427      DEC4=JMS I .      /CALL TO ROUTINE TO ACCEPT 4 DECIMAL DIGITS INPUT
0027      1600      XDEC4      /AND CONVERT THEM TO OCTAL
0030      4430      LIMCHK=JMS I .      /CALL TO ROUTINE TO CHECK FOR ANSWER WITHIN LIMITS
0030      2200      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
0031      4431      PRNTAC=JMS I .      /PRINT CONTENTS OF ACCUMULATOR
0031      1476      ACPRNT
0032      4432      PRINT=JMS I .      /PRINT ONE ASCII CHARACTER
0032      2130      XPRINT
0033      1727      XTFIL      /CALL+1 IS STARTING ADDRESS OF TABLE
                                /CALL+2 IS THE MEMORY FIELD WHERE THE TABLE IS LOCATED
                                /      IN BITS 6 - 8
                                /CALL+3 IS NUMBER OF TABLE ENTRIES (UNSIGNED)
                                /CALL+4 IS THE VALUE TO PLACE IN EACH OF THE TABLE LOCATIONS
                                /CALL+5 IS RETURN WITH AC CLEAR
0034      4434      CRLF=JMS I .      /ROUTINE TO PRINT <CR> AND <LF>
0034      2057      XCRLF

```

0035	4435	INPUT=JMS I .	/ROUTINE TO WAIT FOR KEYBOARD INPUT
	2270	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
	2400	PRTDEC	
0037	4437	APTCK=JMS I .	/CALL TO CHECK FOR ON APT
	1757	CKAPT	
0040	4440	SETPT=JMS I .	/CALL TO ROUTINE TO SET "TBLPT" TO DRIVE STATE TABLE ENTRY FLD 0
	2314	PTSET	
0041	4441	C8CAL=JMS I .	/CALL TO CONSOLE PACKAGE
	2342	CALC8	
0042	4442	PASPT=JMS I .	/CALL TO ROUTINE TO SET "UPPNT" TO
	3142	PTPAS	
0043	4443	GETSR=JMS I .	/CALLO ROUTINE TO GET SOFT OR HARDWARE SWITCHES
	3364	SRGET	
0044	4444	VT278B=JMS I .	
	1527	XVT27B	
0045	0000	INCHAR, 0	/ASCII CHARACTER INPUT STORAGE
0046	0000	INMODE, 0	/MESSAGE ACTIVE FLAG 7777=ACTIVE 0=NON-ACTIVE
0047	0000	UPPNT, 0	/END OF PASS TABLE POINTER
0050	0000	DVTMP, 0	/TEMPORARY DRIVE NUMBER STORAGE
0051	0000	TBLPT, 0	/DRIVE STATE TABLE POINTER
0052	0000	CNT1, 0	
0053	0000	CNT2, 0	
0054	0000	CDFSV, 0	
0055	0000	CURDV, 0	
0056	0000	TMP1, 0	
0057	0000	QFLAG, 0	
0060	0000	DIGFLG, 0	/DIGIT INPUT FLAG
0061	0000	DSC6X, 0	
0062	0000	DSC6Y, 0	
0063	0003	K03, 0003	
0064	0007	K07, 0007	
0065	0040	K040, 0040	
0066	0047	K047, 0047	
0067	0070	K0070, 0070	
0070	0177	K0177, 0177	
0071	0212	K0212, 0212	
0072	0215	K0215, 0215	
0073	0240	K0240, 0240	
0074	0260	K0260, 0260	
0075	0277	K0277, 0277	
0076	0300	K0300, 0300	
0077	0777	K0777, 0777	
0100	1234	K1234, 1234	/RANDOM GENERATOR PRIME



```

0101 5670 K5670, 5670
0102 7007 K7007, 7007
0103 7777 M01, 7777
0104 7774 M04, -4
0105 7772 M06, -6
0106 7563 M215, -215
0107 6201 KXCDF, CDF
0110 6203 KXCDI, CDI
0111 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
/EXERCISER PORTION OF THE PROGRAM IN FIELD 0.
/
0112 TABSTR=.
/
/
0112 0000 DMFLG, 0 /DATA CRC DATA DUMP FLAG 0=YES 7777=NO
0113 0000 LMFLG, 0 /DROP DRIVE ON OPERATION LIMITS REACHED 0=YES 7777=NO
0114 0000 CBUSY1, 0 /CONTROLLER BUSY FLAG 0=NOT BUSY 7777=BUSY
0115 0000 MNCYL, 0 /MINIMUM CYLINDER TO USE
0116 0000 MNSEC, 0 /MINIMUM SECTOR TO USE
0117 0000 SGLHD, 0 /SINGLE SURFACE TO USE DURING DATA TRANSFERS
0120 7777 BFLD0, 7777 /DRIVE ZERO BUFFER FIELD
0121 7777 BFLD1, 7777 /DRIVE ONE BUFFER FIELD
0122 7777 BFLD2, 7777 /DRIVE TWO BUFFER FIELD
0123 7777 BFLD3, 7777 /DRIVE THREE BUFFER FIELD
0124 7777 SGFLG, 7777 /SINGLE SURFACE DATA TRANSFERS FLAG 0=YES 7777=NO
0125 7777 REDFLG, 7777 /READ ONLY FLAG 0=YES 7777=NO
0126 7777 CNFLG, 7777 /CONSTANT DATA FLAG 0=YES 7777=NO
0127 0001 FXWRD, 0001 /FIXED NUMBER OF WORDS TO USE FOR
/DATA TRANSFERS. IF = 0001 MEANS USE RANDOM WORD COUNT
/RETRY LIMIT
0130 0001 RTLIM, 0001 /SEEK TO TRACK RETRY LIMIT
0131 0001 SKRET, 0001 /DATA TRANSFER LIMIT TO DROP DRIVE (TIMES 10(10) BITS)
0132 0001 XFRLM, 0001 /SEEK OPERATION LIMIT TO DROP DRIVE (TIMES 10(3))
0133 0001 SKLIM, 0001 /NUMBER OF ADDITIONAL FIELDS R/W MEMORY
0134 0001 NMFLD, 0001 /SOFT ERROR LIMIT TO DROP DRIVE
0135 0012 SFLIM, 0012 /DEFAULT CONSTANT DATA PATTERN TO USE (WORST CASE)
0136 0001 DTPAT, 0001 /NUMBER OF DATA WORD ERRORS REPORTED
0137 0376 DTERN, 0376 /MAXIMUM CYLINDER TO USE
0140 0777 MXCYL, 0777 /MAXIMUM SECTOR TO USE
0141 0047 MXSEC, 0047
/
0142 TABED=.
/
/
* END OF CRITICAL LOCATIONS *
/
/*****

```

HP 003

HP 001

```

0200      *200
/
0200 4424  CONSET, MESSAGE
0201 4361      TITLE
0202 5222      JMP      INIT
0203 4424  CLRQ,  MESSAGE
0204 4361      TITLE
0205 6201      CDF      00
0206 1100      TAD      K1234
0207 3777      DCA I    (RAN1
0210 1101      TAD      K5670
0211 3776      DCA I    (RAN2
0212 6211      CDF      10
0213 4424      MESSAGE
0214 4513      CLR TAB
0215 4424      MESSAGE
0216 4002      YORN
0217 4425      YESRNO
0220 7610      SKP CLA
0221 5260      JMP      FLDS

/
0222 1375  INIT,  TAD      (DRIVE0-1
0223 3010      DCA      INDX10
0224 6201  INITLP, CDF      00
0225 3410      DCA I    INDX10
0226 6211      CDF      10
0227 1374      TAD      (DRVEND
0230 7041      CIA
0231 1010      TAD      INDX10
0232 7640      SZA CLA
0233 5224      JMP      INITLP
0234 4433      TABFIL
0235 3400      BADSEC
0236 0010
0237 0200      BADEND-BADSEC+1
0240 7777
0241 4437      APTCK
0242 5244      JMP      ONAPT
0243 5260      JMP      FLDS

/
0244 1021  ONAPT, TAD      HDW1
0245 7012      RTR
0246 0064      AND      K07
0247 3134      DCA      NMFLD
0250 1064      TAD      K07
0251 7041      CIA
0252 1134      TAD      NMFLD
0253 7710      SPA CLA
0254 5257      JMP      .+3
0255 7327      CLA CLL  CML IAC RTL
0256 3134      DCA      NMFLD
0257 5272      JMP      AUTSIZ

/
0260 3057  FLDS,  DCA      QFLAG
0261 4424      MESSAGE

```

```

/AJRLK-A RL8A PERFORMANCE EXERCISER FOR THE RL02
/GO CLEAR TABLES

```

```

/GET FIRST RANDOM GENERATOR PRIME
/RESTORE FIRST RANDOM PRIME
/GET SECOND RANDOM GENERATOR PRIME
/RESTORE SECOND RANDOM PRIME
/CDF TO PRGM FLD

```

```

/CLEAR DRIVE STATUS TABLE

```

```

/(Y/N)?
/GET YES OR NO ANSWER
/YES RETURN
/NO RETURN

```

```

/GET DRIVE TABLE ADDRESS-1
/INIT AUTO-INDEX TABLE POINTER
/DATA FLD 0
/CLEAR A DRIVE TABLE LOCATION
/DATA FLD 1
/GET ADDRESS OF END OF TABLE
/NEGATE
/ADD VALUE OF POINTER
/SKIP IF TABLE CLEARED
/CONTINUE CLEARING TABLE
/FILL BAD SECTOR TABLE
/ADDRESS OF TABLE
/TABLE FIELD (1)
/NUMBER OF TABLE ENTRIES
/WORD TO PLACE IN TABLE
/CHECK FOR ON APT
/ON APT RETURN
/NOT ON APT RETURN

```

```

/GET HCW1
/ROTATE FOR MEMORY SIZE
/MASK NUMBER OF EXT D R/W FIELDS
/SAVE NUMBER OF EXT D R/W FIELDS

```

```

/SKIP IF NUMBER OF FIELDS = 7

```

```

/FORCE TO 6 FIELDS (28K)
/GO SIZE MEMORY

```

```

/CLEAR ALL QUESTIONS ANSWERED FLAG

```

0262	4445	FLDMSG		/NUMBER OF FIELDS EXT D R/W MEM(1-7)?	
0263	4426	OCT1		/GET ONE OCTAL DIGIT	
0264	3134	DCA	NMFLD	/SAVE NUMBER OF FIELDS	
0265	1134	TAD	NMFLD	/GET ANSWER (NO. OF FLDS)	
0266	4430	LIMCHK		/CHECK ANSWER AGAINST LIMITS	
0267	0007	0007		/UPPER LIMIT	
0270	0001	0001		/LOWER LIMIT (MUST BE AT LEAST 8K CORE)	
0271	5260	JMP	FLDS	/INVALID ANSWER RETURN - TRY AGAIN	
0272	1134	AUTSIZ, TAD	NMFLD	/GET NUMBER OF FIELDS EXT D R/W MEM	
0273	7041	CIA			
0274	3052	DCA	CNT1	/SET UP FIELD COUNTER	
0275	1373	TAD	(CDF		
0276	3054	DCA	CDFSV		
0277	1372	/ FLDLP, TAD	(10		
0300	1054	TAD	CDFSV	/ADD ONE TO FIELD OF CDF INSTRUCTION	
0301	3054	DCA	CDFSV	/SAVE IT	
0302	1054	TAD	CDFSV	/GET MODIFIED CDF	
0303	3304	DCA	.+1	/SAVE FOR EXECUTION	
0304	7402	HLT/CDF		/MODIFIED CDF TO AUTO-SIZE	
0305	1771	TAD I	(7777	/SAVE THE CONTENTS OF THIS LOCATION ON PAGE	HP 002
0306	3056	DCA	TMP1	/SAVE IT IN TEMP LOC PAGE ZERO	HP 002
0307	3771	DCA I	(7777	/WRITE ALL ZEROS INTO HIGHEST FIELD LOCATION	
0310	1771	TAD I	(7777	/READ SAME LOCATION	
0311	7640	SZA CLA		/SKIP IF ALL ZEROS READ	
0312	5321	JMP	ERRMEM	/FIELD NOT FOUND ERROR	HP 002
0313	7040	CMA			
0314	3771	DCA I	(7777	/WRITE ALL ONES INTO SAME LOCATION	
0315	1771	TAD I	(7777	/READ SAME LOCATION	
0316	7001	IAC		/ADD 1	
0317	7100	CLL		/CLEAR THE LINK	HP 002
0320	7640	SZA CLA		/SKIP IF ALL ONES READ	
0321	7120	ERRMEM, STL		/SET THE LINK TO 1 ERROR INDICATOR	HP 002
0322	1056	TAD	TMP1	/GET BACK ORIGINAL DATA	HP 002
0323	3771	DCA I	(7777	/STORE SAVED DATA BACK	HP 002
0324	7430	SZL		/SKIP IF NO ERROR FLAG LINK NOT SET TO 1	HP 002
0325	5770	JMP	MEMERR	/ERROR WITH FIELD CANNOT WRITE CORRECTLY	HP 002
0326	6211	CDF 10		/CDF BACK TO PROGRAM FIELD	
0327	2052	ISZ	CNT1	/INCREMENT FIELD COUNT	
0330	5277	JMP	FLDLP	/TRY NEXT FIELD	
0331	4437	APTCK		/CHECK FOR ON APT	
0332	7610	SKP CLA		/ON APT SO SKIP	
0333	5767	JMP	DRVS	/NOT ON APT	
0334	1022	APTDRV, TAD	HDW2	/GET HDW2	
0335	0076	AND	K0300	/MASK BITS 4 AND 5	
0336	7640	SZA CLA		/SKIP IF BITS 9:11 ARE HIGHEST UNIT TO TEST	
0337	5766	JMP	ONEUNT	/ONE UNIT TO TEST	
0340	1022	TAD	HDW2	/GET HDW2	
0341	0063	AND	K03	/MASK DRIVE BITS	
0342	7450	SNA		/SKIP IF HIGHEST DRIVE IS NOT 0	
0343	5766	JMP	ONEUNT	/ONE DRIVE TO TEST	
0344	7001	IAC		/DRIVE 0 + 1 + ( HIGHEST DRIVE # ) = NUMBER OF DRIVES	
0345	7041	CIA			
0346	3052	DCA	CNT1	/SET UP DRIVE COUNTER	

```

0347 3055      DCA      CURDV      /INIT CURRENT DRIVE TO 0
0350 5765'     JMP      APTDLP     /GO TO NEXT PAGE

/
/
/*****
/
/ROUTINE TO UPDATE APT TIMER AND REPORT TO APT
/
/      CALLED BY:      JMS I      (APTTIC      FROM FLD 0
/
/
0351 0000      APTTIC, 0
0352 2362      ISZ      TIMER      /UPDATE APT TIMER
0353 5360      JMP      APTX1      /NOT TIME TO REPORT TO APT PROM
0354 1363      TAD      APTTIM     /GET APT TIMER VALUE
0355 3362      DCA      TIMER     /RESET APT TIMER FOR NEXT REPORT
0356 6272      CIF      70        /GOING TO APT PROM FLD 7
0357 4764      JMS I      (6500    /REPORT TO APT PROM

/
0360 6203      APTX1, CDI      00      /RETURNING TO FLD 0 INTERRUPT SERVICE
0361 5751      JMP I      APTTIC     /RETURN

/
0362 7716      TIMER, -62      /APT TIMER (ALLOW 50 INTERRUPTS BEFORE REPORT TO APT)
0363 7716      APTTIM, -62     /APT TIMER INITIALIZE VALUE

/
/
0364 6500
0365 0400
0366 0411
0367 0446
0370 0422
0371 7777
0372 0010
0373 6201
0374 7123
0375 6713
0376 5162
0377 5161
0400 0400      PAGE

/
APTDLP, SETPT
0400 4440      DSTATE
0401 0000      /SET "TBLPNT" TO DRIVE STATE
0402 7240      CLA CMA
0403 3451      DCA I      TBLPT    /SET DRIVE ACTIVE
0404 6211      CDF      10
0405 2055      ISZ      CURDV     /INCREMENT DRIVE NUMBER TO ACTIVATE
0406 2052      ISZ      CNT1      /INCREMENT DRIVE COUNTER - SKIP WHEN DONE
0407 5200      JMP      APTDLP     /CONTINUE
0410 5777'     JMP      MAINGO     /GO TO MAIN PROGRAM

/
0411 1022      ONEUNT, TAD      HDW2    /GET HDW2
0412 0063      AND      K03
0413 3055      DCA      CURDV     /SAVE SINGLE DRIVE TO TEST

```

```

0414 4440      SETPT
0415 0000      DSTATE
0416 7240      CLA CMA
0417 3451      DCA I   TBLPT
0420 6211      CDF     10
0421 5777      JMP     MAINGO
                /
                /
0422 6211      MEMERR, CDF 10
0423 4437      APTCK
0424 7610      SKP CLA
0425 5235      JMP     CONMEM
0426 1054      TAD     CDFSV
0427 0067      AND     K0070
0430 7012      RTR
0431 7010      RAR
0432 1103      TAD     MO1
0433 3134      DCA     NMFLD
0434 5776      JMP     APTDRV
                /
                /
0435 4424      CONMEM, MESSAGE
0436 4465      NOFLD
0437 1054      TAD     CDFSV
0440 0067      AND     K0070
0441 7012      RTR
0442 7010      RAR
0443 1074      TAD     K0260
0444 4432      PRINT
0445 5775      JMP     FLDS
                /
0446 3055      DRVS,   DCA     CURDV
0447 1374      TAD     (-4
0450 3052      DCA     CNT1
0451 4424      DRVLP1, MESSAGE
0452 3761      TORIVE
0453 1055      TAD     CURDV
0454 1074      TAD     K0260
0455 4432      PRINT
0456 4424      MESSAGE
0457 4002      YORN
0460 4425      YESRNO
0461 7610      CLA SKP
0462 7610      CLA SKP
0463 7040      CMA
0464 4440      SETPT
0465 0000      DSTATE
0466 3451      DCA I   TBLPT
0467 6211      CDF     10
0470 2055      NXTDRV, ISZ     CURDV
0471 2052      ISZ     CNT1
0472 5251      JMP     DRVLP1
0473 4424      ASKDSC, MESSAGE
0474 4477      DCODE

```

/SET "TBLPNT" TO DRIVE STATE  
 /ACTIVATE DRIVE  
 /GO TO MAIN PROGRAM  
 /CDF BACK TO PROGRAM FIELD  
 /CHECK FOR ON APT  
 /SKIP IF ON APT  
 /NOT ON APT  
 /MASK EMA FIELD BITS OF FIELD NOT FOUND  
 /SUBTRACT ONE FOR HIGHEST FIELD FOUND  
 /FORCE TO NUMBER OF FIELDS FOUND  
 /GO GET DRIVES TO TEST  
 /NO FIELD  
 /GET CDF (FIELD)  
 /MASK EMA BITS  
 /SET UP TO PRINT FIELD NUMBER  
 /PRINT THE FIELD NUMBER  
 /START EXT D MEM QUESTION OVER  
 /INIT CURRENT DRIVE INDICATOR TO ZERO  
 /SET UP FOR 4 DRIVE QUESTIONS  
 /TEST DRIVE  
 /GET CURRENT DRIVE  
 /PRINT DRIVE #  
 /(Y/N)?  
 /GET ANSWER  
 /YES RETURN  
 /NO RETURN  
 /REMEMBER ANSWER IN DRIVE TABLE  
 /INCREMENT DRIVE #  
 /INCREMENT LOOP COUNTER  
 /CONTINUE ASKING QUESTIONS  
 /DEVICE CODE

```

0475 4424      MESSAGE
0476 4040      DC6061      /((60,61)
0477 4424      MESSAGE
0500 4002      YORN      /(Y/N)?
0501 4425      YESRNO    /GET (Y)ES OR (N)O ANSWER
0502 5310      JMP      USE60
0503 1373      USE62,   TAD      (620
0504 3061      DCA      DSC6X
0505 1372      TAD      (630
0506 3062      DCA      DSC6Y
0507 5314      JMP      IOTCNG      /GO CHANGE IOT DEVICE CODES

/
0510 1371      USE60,   TAD      (600
0511 3061      DCA      DSC6X      /SAVE FIRST DEVICE CODE TO USE
0512 1370      TAD      (610
0513 3062      DCA      DSC6Y      /SAVE SECOND DEVICE CODE TO USE

/
0514 1367      IOTCNG,  TAD      (DCTB60
0515 3056      DCA      TMP1      /GET ADDRESS OF FIRST DEV CODE TABLE
0516 1366      TAD      (DCTB61-DCTB60 /SET UP POINTER TO DEV CODE TABLE
0517 7041      CIA      /GET SIZE OF TABLE
0520 3052      DCA      CNT1      /NEGATE FOR COUNTER
0521 6201      DC60LP,  CDF 00    /SET UP DEV CODE COUNTER
0522 1456      TAD I    TMP1      /CDF TO IOT TABLE FLD 0
0523 3051      DCA      TBLPT     /GET ADDR OF FIRST DEV CODE
0524 1451      TAD I    TBLPT     /SET UP IOT CHANGE POINTER TO IOT ADDR
0525 0102      AND      K7007    /GET IOT TO CHANGE
0526 1061      TAD      DSC6X    /MASK OFF DEVICE CODE
0527 3451      DCA I    TBLPT     /ADD FIRST DEV CODE
0530 2056      ISZ      TMP1      /SAVE MODIFIED IOT
0531 2052      ISZ      CNT1      /INCREMENT POINTER
0532 5321      JMP      DC60LP    /INCREMENT COUNTER
0533 6211      CDF      10        /CONTINUE CHANGING FIRST DEV CODE

/
0534 1365      TAD      (DCTB61    /GET ADDRESS OF SECOND DEV CODE TABLE
0535 3056      DCA      TMP1      /SET UP POINTER
0536 1364      TAD      (DCEND-DCTB61 /GET SIZE OF TABLE
0537 7041      CIA      /NEGATE FOR COUNTER
0540 3052      DCA      CNT1      /SET UP COUNTER

/
0541 6201      DC61LP,  CDF 00    /CDF TO IOT TABLE FLD 0
0542 1456      TAD I    TMP1      /GET ADDR OF SECOND DEV CODE IOT
0543 3051      DCA      TBLPT     /SET UP IOT CHANGE POINTER TO IOT ADDR
0544 1451      TAD I    TBLPT     /GET IOT TO CHANGE
0545 0102      AND      K7007    /MASK OFF DEV CODE BITS
0546 1062      TAD      DSC6Y    /ADD SECOND DEV CODE
0547 3451      DCA I    TBLPT     /SAVE MODIFIED IOT
0550 2056      ISZ      TMP1      /INCREMENT TABLE POINTER
0551 2052      ISZ      CNT1      /INCREMENT TABLE COUNTER
0552 5341      JMP      DC61LP    /CONTINUE CHANGING SECOND DEV CODE IOTS
0553 6211      CDF      10        /CDF TO PRGM FLD
0554 5771      JMP      SRQUES    /GO TO NEXT PAGE

0564 0016
0565 7146

```

0566 0006  
 0567 7140  
 0570 0610  
 0571 0600  
 0572 0630  
 0573 0620  
 0574 7774  
 0575 0260  
 0576 0334  
 0577 1456  
 0600

PAGE

/

0600 4441 SRQUES, C8CAL  
 0601 7402 HLT  
 0602 4424 QUEST, MESSAGE  
 0603 3753 ACCEPT  
 0604 4424 MESSAGE  
 0605 4002 YORN  
 0606 4425 YESRNO  
 0607 7410 SKP  
 0610 5243 JMP RETQ1  
 0611 4433 TINIT, TABFIL  
 0612 0112 DMFLG  
 0613 0010 0010  
 0614 0006 6  
 0615 0000 0  
 0616 4433 TABFIL  
 0617 0120 BFLDO  
 0620 0010 0010  
 0621 0007 7  
 0622 7777 7777  
 0623 4433 TABFIL  
 0624 0127 FXWRD  
 0625 0010 0010  
 0626 0005 5  
 0627 0001 0001

/

/

0630 1377 TINIT1, TAD (12  
 0631 3135 DCA SFLIM  
 0632 7301 CLA CLL IAC  
 0633 3136 DCA DTPAT  
 0634 1376 TAD (376  
 0635 3137 DCA DTERN  
 0636 1077 TAD K0777  
 0637 3140 DCA MXCYL  
 0640 1066 TAD K047  
 0641 3141 DCA MXSEC  
 0642 5775 JMP DBLCHK

/

0643 4434 RETQ1, CRLF  
 0644 4424 MESSAGE  
 0645 3770 RETLM

/CALL CONSOLE PACKAGE - ASK SR=  
 /NOT ACTIVE RETURN - PROGRAM ERROR?  
 /ACTIVE RETURN  
 /ACCEPT MODE

/(Y/N)?  
 /GET ANSWER  
 /YES RETURN - RESTORE DEFAULT PARAMETERS  
 /NO RETURN GO ASK MANUAL MODE QUESTIONS  
 /CALL TABLE FILLING SUBROUTINE  
 /STARTING ADDRESS OF TABLE  
 /TABLE FIELD (1)  
 /NUMBER OF TABLE ENTRIES  
 /VALUE TO PLACE IN TABLE  
 /CALL TABLE FILLING SUBROUTINE  
 /STARTING ADDRESS OF TABLE  
 /TABLE FIELD (1)  
 /NUMBER OF TABLE ENTRIES  
 /VALUE TO PLACE IN TABLE  
 /CALL TABLE FILLING SUBROUTINE  
 /STARTING ADDRESS OF TABLE  
 /TABLE FIELD (1)  
 /NUMBER OF TABLE ENTRIES  
 /VALUE TO PLACE IN TABLE

HP 014

/RESTORE SOFT ERROR LIMIT  
 /AC=1  
 /RESTORE DEFAULT CONSTANT DATA PATTERN  
 /  
 /RESTORE NUMBER OF DATA WORD ERRORS TO PRINT  
 /  
 /RESTORE MAXIMUM CYLINDER TO USE  
 /RESTORE MAXIMUM SECTOR TO USE  
 /GO ASK ARE YOU SURE

HP 015

HP 001

/DO A <CR> AND <LF>  
 /RETRY LIMIT

0646	4424	MESSAGE		/((1-4095)?
0647	4007	LM4095		/GET FOUR DECIMAL DIGITS - RETRY LIMIT
0650	4427	DEC4		/SKIP IF NOT 4096
0651	7430	SZL		/4096 NOT VALID - TRY AGAIN
0652	5243	JMP	RETQ1	/SKIP IF NON-ZERO
0653	7450	SNA		/USE DEFAULT = 1
0654	7301	CLA CLL IAC		/SAVE OCTAL RETRY LIMIT
0655	3130	DCA	RTLIM	
/				
0656	4434	RETQ2,	CRLF	/DO A <CR> AND <LF>
0657	4424	MESSAGE		
0660	3777	SEKMSG		/SEEK
0661	4424	MESSAGE		
0662	3770	RETLN		/RETRY LIMIT
0663	4424	MESSAGE		
0664	4007	LM4095		/((1-4095)?
0665	4427	DEC4		/GET 4 DECIMAL DIGITS - SEEK RETRY LIMIT
0666	7430	SZL		/SKIP IF NOT 4096
0667	5256	JMP	RETQ2	/4096 NOT VALID - TRY AGAIN
0670	7450	SNA		/SKIP IF NON-ZERO
0671	7301	CLA CLL IAC		/USE DEFAULT = 1
0672	3131	DCA	SKRET	/SAVE OCTAL SEEK RETRY LIMIT
0673	4424	MESSAGE		
0674	4045	DATDUM		/DATA DUMP ON DATA CRC
0675	4424	MESSAGE		
0676	4057	ERR		/ERROR
0677	4424	MESSAGE		
0700	4002	YORN		/((Y/N)?
0701	4425	YESRNO		/GET ANSWER
0702	7610	CLA SKP		/YES RETURN
0703	7240	CLA CMA		/NO RETURN
0704	3112	DCA	DMFLG	/REMEMBER ANSWER
/				
0705	4424	SLIM,	MESSAGE	
0706	4067	SOFT		/SOFT
0707	4424	MESSAGE		
0710	4057	ERR		/ERROR
0711	4424	MESSAGE		
0712	4063	LIMMSG		/LIMIT
0713	4424	MESSAGE		
0714	4007	LM4095		/((1-4095)?
0715	4427	DEC4		/GET FOUR DECIMAL DIGITS - SOFT ERROR LIMIT
0716	7430	SZL		/SKIP IF NOT 4096
0717	5305	JMP	SLIM	/4096 NOT VALID - TRY AGAIN
0720	7450	SNA		/SKIP IF NON-ZERO
0721	1377	TAD	(12	/USE DEFAULT = 10
0722	3135	DCA	SFLIM	/SAVE OCTAL SOFT ERROR LIMIT
0723	5774	JMP	OPLIM	/GO TO NEXT PAGE
/				
0774	1000			
0775	1445			
0776	0376			
0777	0012			
	1000	PAGE		



```

/
/
/
1000 4424 OPLIM, MESSAGE
1001 4341 DROP /DROP
1002 4424 MESSAGE
1003 4420 DRVON /DRIVE ON
1004 4424 MESSAGE
1005 4424 OPRAT /OPERATION
1006 4424 MESSAGE
1007 4063 LIMMSG /LIMIT
1010 4424 MESSAGE
1011 4413 REACHD /REACHED
1012 4424 MESSAGE
1013 4002 YORN /((Y/N)?
1014 4425 YESRNO /GET ANSWER
1015 5221 JMP .+4 /YES RETURN
1016 7240 CLA CMA /NO RETURN
1017 3113 DCA LMFLG /REMEMBER ANSWER
1020 5256 JMP CHNGP /GO CHECK FOR PARAMETER CHANGES
1021 3113 DCA LMFLG /REMEMBER YES ANSWER
1022 4434 CRLF /DO A <CR> AND <LF>

/
1023 4424 DXLIM, MESSAGE
1024 4101 DXFER /DATA TRANSFER
1025 4424 MESSAGE
1026 4063 LIMMSG /LIMIT
1027 4424 MESSAGE
1030 4007 LM4095 /((1-4095)?
1031 4427 DEC4 /GET LIMIT IN DECIMAL
1032 7430 SZL /SKIP IF NOT 4096
1033 5223 JMP DXLIM /4096 NOT VALID - TRY AGAIN
1034 7450 SNA /SKIP IF NON-ZERO
1035 7301 CLA CLL IAC /USE DEFAULT = 1
1036 3132 DCA XFRLM /SAVE OCTAL DATA TRANSFER LIMIT

/
1037 4434 SKLIM2, CRLF /DO A <CR> AND <LF>
1040 4424 MESSAGE
1041 3777 SEKMSG /SEEK
1042 4424 MESSAGE
1043 4424 OPRAT /OPERATION
1044 4424 MESSAGE
1045 4063 LIMMSG /LIMIT
1046 4424 MESSAGE
1047 4007 LM4095 /((1-4095)?
1050 4427 DEC4 /GET LIMIT IN DECIMAL
1051 7430 SZL /SKIP IF NOT 4096
1052 5237 JMP SKLIM2 /4096 NOT VALID - TRY AGAIN
1053 7450 SNA /SKIP IF NON-ZERO
1054 7301 CLA CLL IAC /USE DEFAULT = 1
1055 3133 DCA SKLIM /SAVE OCTAL SEEK OPERATION LIMIT (X 1000)

/
1056 4424 CHNGP, MESSAGE
1057 4111 CHNGSK
1060 4424 MESSAGE

```

/CHANGE SEEK, R/W PARAMETERS

1061	4002	YORN		/(Y/N)?
1062	4425	YESRNO		/GET ANSWER
1063	7410	SKP		/YES RETURN
1064	5777'	JMP	DBLCHK	/NO RETURN - GO ASK IF SURE
1065	4424	FIXWD, MESSAGE		
1066	4031	RAND		/RANDOM
1067	4424	MESSAGE		
1070	4036	WCOUNT		/WORD COUNT
1071	4424	MESSAGE		
1072	4002	YORN		/(Y/N)?
1073	4425	YESRNO		/GET YES OR NO ANSWER
1074	7610	SKP CLA		/YES RETURN
1075	5301	JMP	GETWRD	/NO RETURN
1076	7001	IAC		/NOT USING FIXED WORD COUNT
1077	3127	DCA	FXWRD	/FIXWRD=1=USE RANDOM WORD COUNT
1100	5776'	JMP	READON	/GO ASK READ ONLY QUESTION
/				
1101	4424	GETWRD, MESSAGE		
1102	4036	WCOUNT		/WORD COUNT
1103	7344	CLA CLL	CHA RAL	/AC=-2=7776
1104	1134	TAD	NMFLD	/ADD NUMBER OF FIELDS EXTEND R/W MEM
1105	7710	SPA CLA		/SKIP IF MORE THAN 8K
1106	5326	JMP	ONEFLD	/ONLY 8K - SMALL BUFFERS ONLY - GO CHK
1107	4424	MESSAGE		
1110	4015	LM4096		/(3-4096)?
1111	4427	DEC4		/GET 4 DECIMAL DIGITS
1112	3127	DCA	FXWRD	/SAVE WORD COUNT TO USE
1113	1060	TAD	DIGFLG	/GET DIGIT INPUT FLAG
1114	7650	SNA CLA		/SKIP IF DIGITS INPUT
1115	5776'	JMP	READON	/NO DIGITS INPUT - USE 4096 DEFAULT
1116	7430	SZL		/SKIP IF NOT 4096
1117	5776'	JMP	READON	/GO ASK READ ONLY QUESTION
1120	1127	TAD	FXWRD	/GET WORD COUNT
1121	4430	LIMCHK		/CHECK AGAINST LIMITS
1122	7777	7777		/4096 ALREADY KNOW IT'S NOT 4096
1123	0003	0003		/MINIMUM WORD COUNT
1124	5301	JMP	GETWRD	/INVALID ANSWER - TRY AGAIN
1125	5776'	JMP	READON	/GO ASK READ ONLY QUESTION
/				
1126	4424	ONEFLD, MESSAGE		
1127	4023	LM1024		/(3-1024)?
1130	4427	DEC4		/GET 4 DECIMAL DIGITS
1131	3127	DCA	FXWRD	/SAVE WORD COUNT
1132	1060	TAD	DIGFLG	/GET DIGIT INPUT FLAG
1133	7640	SZA CLA		/SKIP IF NOT DIGITS INPUT
1134	5337	JMP	.+3	/DIGITS INPUT
1135	1375	TAD	(2000	/USE DEFAULT = 1024(10)
1136	3127	DCA	FXWRD	/SAVE DEFAULT TRANSFER LENGTH
1137	1127	TAD	FXWRD	/GET WORD COUNT
1140	4430	LIMCHK		/CHECK AGAINST BUFFER SIZE
1141	2000	2000		/1024(10) WORDS MAXIMUM
1142	0003	0003		/3 WORDS MINIMUM
1143	5301	JMP	GETWRD	/INVALID GO GET NEW WORD COUNT
1144	5776'	JMP	READON	/GO TO NEXT PAGE

1175 2000  
1176 1200  
1177 1445  
1200

PAGE

```

1200 4424 READON, MESSAGE
1201 4427 RDONL /READ ONLY
1202 4424 MESSAGE
1203 4002 YORN /(Y/N)?
1204 4425 YESRNO /GET ANSWER
1205 7610 CLA SKP /YES RETURN
1206 7240 CLA CMA /NO RETURN
1207 3125 DCA REDFLG /REMEMBER ANSWER
1210 1125 TAD REDFLG /GET READ ONLY FLAG
1211 7650 SNA CLA /SKIP IF NOT READ ONLY
1212 5243 JMP NEXTQ1 /GO ASK FOR NUMBER OF DATA WORD ERRORS
1213 4424 CDATA, MESSAGE
1214 4134 CONSD /CONSTANT DATA
1215 4424 MESSAGE
1216 4002 YORN /(Y/N)?
1217 4425 YESRNO /GET ANSWER
1220 5224 JMP .+4 /YES RETURN
1221 7240 CLA CMA /NO RETURN
1222 3126 DCA CNFLG /INDICATE USING RANDOM DATA
1223 5243 JMP NEXTQ1 /GO TO NEXT QUESTION
1224 3126 DCA CNFLG /SET CONSTANT DATA FLAG

```

```

1225 4424 DPAT, MESSAGE
1226 4144 DPATN /DATA PATTERN (0-3)?
1227 4426 OCT1 /GET ONE OCTAL DIGIT
1230 3136 DCA DTPAT /SAVE DATA PATTERN # TO USE
1231 1060 TAD DIGFLG /GET DIGIT INPUT FLAG
1232 7640 SZA CLA /SKIP IF NOT DIGITS INPUT
1233 5236 JMP .+3 /GO USE INPUT VALUE
1234 7301 CLA CLL IAC /USE DEFAULT=1 WORST CASE PATTERN
1235 3136 DCA DTPAT /USE DEFAULT WORST CASE DATA PATTERN
1236 1136 TAD DTPAT /GET DATA PATTERN NUMBER
1237 4430 LIMCHK /CHECK AGAINST LIMITS
1240 0003 0003 /HIGHEST PATTERN NUMBER
1241 0000 0000 /LOWEST PATTERN NUMBER
1242 5225 JMP DPAT /INVALID, GO GET NEW PATTERN NUMBER AGAIN

```

```

1243 1112 NEXTQ1, TAD DMFLG /GET DATA DUMP FLAG
1244 7640 SZA CLA /SKIP IF OK TO DUMP DATA
1245 5266 JMP SNGLO /NO DATA DUMP, SKIP NEXT QUESTION
1246 4424 MESSAGE
1247 4153 NUMWRD /NUMBER OF DATA WORD ERRORS TO REPORT (0-254)?
1250 4427 DEC4 /GET DECIMAL ANSWER
1251 3137 DCA DTERN /SAVE OCTAL ANSWER
1252 7430 SZL /SKIP IF NOT 4096
1253 5243 JMP NEXTQ1 /4096(10) = 0000(8) TRY AGAIN
1254 1060 TAD DIGFLG /GET DIGIT INPUT FLAG

```

1255	7640	SZA CLA	/SKIP IF NO DIGIT INPUT	
1256	5261	JMP .+3	/DIGIT WAS INPUT	
1257	1377	TAD (376		
1260	3137	DCA DTERN	/SAVE DEFAULT=254 (DECIMAL)	
/				
1261	1137	TAD DTERN	/GET ANSWER	
1262	4430	LIMCHK	/CHECK ANSWER AGAINST LIMITS	
1263	0376	0376	/UPPER LIMIT - 254 DECIMAL	
1264	0000	0	/LOWER LIMIT	
1265	5243	JMP NEXTQ1	/INVALID ANSWER RETURN - TRY AGAIN	
1266	4424	SNGLO, MESSAGE		
1267	4167	SNGSUR	/SINGLE SURFACE	
1270	4424	MESSAGE		
1271	4101	DXFER	/DATA TRANSFER	
1272	4424	MESSAGE		
1273	4002	YORN	/(Y/N)?	
1274	4425	YESRNO	/GET ANSWER	
1275	5301	JMP .+4	/YES RETURN	
1276	7240	CLA CMA	/NO RETURN	
1277	3124	DCA SGFLG	/USE BOTH SURFACES	
1300	5314	JMP HICYL	/GO GET HIGHEST CYLINDER	
1301	3124	DCA SGFLG	/REMEMBER YES ANSWER	
/				
1302	4424	WCHSUR, MESSAGE		
1303	4434	WCHONE	/WHICH SURFACE (0 OR 1)?	
1304	4426	OCT1	/GET ONE OCTAL DIGIT	
1305	3117	DCA SGLHD	/REMEMBER SINGLE SURFACE TO USE	
1306	1117	TAD SGLHD	/GET ANSWER	
1307	4430	LIMCHK	/CHECK ANSWER AGAINST LIMITS	
1310	0001	1	/UPPER LIMIT	
1311	0000	0	/LOWER LIMIT	
1312	5302	JMP WCHSUR	/INVALID ANSWER - TRY AGAIN	
1313	5314	JMP HICYL	/GO TO NEXT PAGE	
/				
1314	4424	HICYL, MESSAGE		
1315	4126	MAXMSG	/MAXIMUM	
1316	4424	MESSAGE		
1317	4202	CYLMSG	/CYLINDER	
1320	4424	MESSAGE		
1321	4211	LMS11	/((0-511)?	HP 001
1322	4427	DEC4	/GET DECIMAL ANSWER	
1323	3140	DCA MXCYL	/SAVE MAXIMUM CYLINDER	
1324	7430	SZL	/SKIP IF NOT 4096	
1325	5314	JMP HICYL	/IT WAS 4096 TRY AGAIN	
1326	1060	TAD DIGFLG	/GET DIGIT INPUT FLAG	
1327	7640	SZA CLA	/SKIP IF NO DIGITS INPUT	
1330	5333	JMP .+3	/DIGITS WERE INPUT	
1331	4077	TAD K0777	/GET DEFAULT VALUE	HP 001
1332	3140	DCA MXCYL	/USE DEFAULT VALUE 777 (OCTAL) AS MAX CYL	HP 001
/				
1333	1140	TAD MXCYL	/GET ANSWER	
1334	4430	LIMCHK	/CHECK ANSWER AGAINST LIMITS	
1335	0777	0777	/UPPER LIMIT (511 DECIMAL)	HP 001
1336	0000	0	/LOWER LIMIT	
1337	5314	JMP HICYL	/INVALID ANSWER - TRY AGAIN	

```

/
/
1340 4424 LOCYL, MESSAGE
1341 4177 MINMSG /MINIMUM
1342 4424 MESSAGE
1343 4202 CYLMSG /CYLINDER
1344 4424 MESSAGE
1345 4211 LM511 /(0-511)? HP 001
1346 4427 DEC4 /GET DECIMAL ANSWER
1347 3115 DCA MNCYL /SAVE OCTAL MINIMUM CYLINDER
1350 7430 SZL /SKIP IF NOT 4096
1351 5340 JMP LOCYL /4096, TRY AGAIN
1352 1140 TAD MXCYL /GET MAXIMUM CYLINDER
1353 3356 DCA .+3 /STORE AS UPPER LIMIT FOR MINIMUM CYLINDER
1354 1115 TAD MNCYL /GET MINIMUM CYLINDER
1355 4430 LIMCHK /CHECK AGAINST LIMITS
1356 0777 0777/MAXCYL /UPPER LIMIT - MODIFIED TO MAXCYL HP 001
1357 0000 0 /LOWER LIMIT
1360 5340 JMP LOCYL /INVALID ANSWER - TRY AGAIN
1361 5776 JMP HISEC /GO TO NEXT PAGE

1376 1400
1377 0376
1400 PAGE
/
/
1400 4424 HISEC, MESSAGE
1401 4126 MAXMSG /MAXIMUM
1402 4424 MESSAGE
1403 4205 SECMSG /SECTOR
1404 4424 MESSAGE
1405 4217 LM39 /(0-39)?
1406 4427 DEC4 /GET DECIMAL ANSWER
1407 3141 DCA MXSEC /SAVE MAXIMUM SECTOR
1410 7430 SZL /SKIP IF NOT 4096
1411 5200 JMP HISEC /IT WAS 4096 TRY AGAIN
1412 1060 TAD DIGFLG /GET DIGIT INPUT FLAG
1413 7640 SZA CLA /SKIP IF NO DIGITS INPUT
1414 5217 JMP .+3 /DIGITS WERE INPUT
1415 1066 TAD K047 /GET DEFAULT VALUE
1416 3141 DCA MXSEC /USE DEFAULT MAXIMUM SECTOR=47
1417 1141 TAD MXSEC /GET ANSWER
1420 4430 LIMCHK /CHECK ANSWER AGAINST LIMITS
1421 0047 0047 /UPPER LIMIT - MAXIMUM SECTOR (39 DECIMAL)
1422 0000 0 /LOWER LIMIT
1423 5200 JMP HISEC /INVALID ANSWER - TRY AGAIN

/
/
1424 4424 LOSEC, MESSAGE
1425 4177 MINMSG /MINIMUM
1426 4424 MESSAGE
1427 4205 SECMSG /SECTOR
1430 4424 MESSAGE
1431 4217 LM39 /(0-39)?

```

```

1432 4427      DEC4      /GET MINIMUM SECTOR IN DECIMAL
1433 3116      DCA      MNSEC    /SAVE OCTAL MINIMUM SECTOR
1434 7430      SZL      /SKIP IF NOT 4096
1435 5224      JMP      LOSEC    /IT WAS 4096 TRY AGAIN
1436 1141      TAD      MXSEC    /GET MAXIMUM SECTOR
1437 3242      DCA      .+3      /SAVE AS MINIMUM SECTOR UPPER LIMIT
1440 1116      TAD      MNSEC    /GET MINIMUM SECTOR
1441 4430      LIMCHK    /CHECK ANSWER AGAINST LIMITS
1442 0047      0047/MXSEC /UPPER LIMIT - MODIFIED TO MAXSEC
1443 0000      0          /LOWER LIMIT
1444 5224      JMP      LOSEC    /INVALID ANSWER - TRY AGAIN

/
/
1445 4424      DBLCHK, MESSAGE
1446 4224      RUSURE      /ARE YOU SURE
1447 4424      MESSAGE
1450 4002      YORN      /(Y/N)?
1451 4425      YESRNO    /GET ANSWER
1452 7610      SKP CLA    /YES RETURN
1453 5777      JMP      CLRQ    /NO RETURN - START QUESTIONS OVER
1454 7040      CMA      /AC=7777
1455 3057      DCA      QFLAG    /SET ALL QUESTIONS ANSWERED FLAG FOR
                                /POWER FAIL RESTART
1456 1376      MAINGO, TAD    (TABSTR-1 /GET ADDRESS OF FIRST LOC TO MOVE
1457 3010      DCA      INDX10 /SET UP PAGE 0 FLD 1 POINTER
1460 1375      TAD      (TBLSTR-1 /GET ADDRESS OF FIRST LOC DESTINATION
1461 3011      DCA      INDX11 /SET UP PAGE 0 FLD 0 POINTER
1462 1374      TAD      (TABED-TABSTR /GET SIZE OF TABLE TO MOVE
1463 7041      CIA      /NEGATE FOR USE AS COUNTER
1464 3052      DCA      CNT1    /SET UP TBL CNTR
1465 6211      MOVLP, CDF      10    /DATA FLD 1
1466 1410      TAD I    INDX10 /GET FLD 1 PAGE 0 CONTENTS
1467 6201      CDF      00      /DATA FLD 0
1470 3411      DCA I    INDX11 /MOVE TO FLD 0 PAGE 0
1471 2052      ISZ      CNT1    /INCREMENT MOVE COUNTER
1472 5265      JMP      MOVLP    /CONTINUE COPYING FLD 1 PG 0 TO FLD 0 PG 0
1473 6203      CDI      00      /GOING TO FIELD 0
1474 5675      JMP I    .+1
1475 0235      MAIN      /GO TO FLD 0

/
/
/*****
/
/ROUTINE TO PRINT CONTENTS OF AC
/
/      CALLED BY:      PRNTAC
/
/
1476 0000      ACPRNT, 0
1477 3325      DCA      ACTMP    /SAVE AC
1500 6214      RDF      /READ CALLING DATA FIELD
1501 1110      TAD      KXCDI    /ADD CDI BASE INSTRUCTION
1502 3323      DCA      ACOU     /SAVE CDI TO CALLING FIELD
1503 6211      CDF      10      /DATA FLD 1

```

```

1504 1373      TAD      (-4
1505 3326      DCA      ACCNTR      /INIT DIGIT PRINT COUNTER
1506 1325      TAD      ACTMP      /GET AC
1507 7104      CLL RAL      /EXTRA ROTATE FOR LINK
1510 7004      ACLP,    RAL
1511 7006      RTL
1512 3325      DCA      ACTMP      /ROTATE LEFT AC OCTAL DIGIT FOR PRINTING
1513 1325      TAD      ACTMP      /SAVE ROTATED AC
1514 0064      AND      K07      /GET ROTATED AC
1515 1074      TAD      K0260     /MASK OCTAL DIGIT
1516 4432      PRINT     /ADD ASCII BASE CODE
1517 1325      TAD      ACTMP      /PRINT OCTAL DIGIT
1520 2326      ISZ      ACCNTR     /GET ROTATED AC
1521 5310      JMP      ACLP      /INCREMENT DIGIT COUNTER
1522 7300      CLA CLL      /CONTINUE
1523 7402      ACOUT,    HLT/CDI
1524 5676      JMP I    ACPRNT     /CDI TO CALLING FIELD MODIFIED
                                   /EXIT
/
1525 0000      ACTMP,    0      /TEMPORARY STORAGE OF ROTATED AC
1526 0000      ACCNTR,    0      /DIGIT COUNTER
1527 0000      XVT27B,    0
1530 6031      KSF
1531 6030      KCF
1532 5727      JMP I    XVT27B

/
/
/
/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
/
/
1533 0000      XDELAY,    0
1534 7300      CLA      CLL
1535 1733      TAD I    XDELAY      /GET NUMBER OF SECONDS TO DELAY FROM CALL+1
1536 3356      DCA      DLY1      /SET UP DELAY COUNTER
1537 1372      DLPA,    TAD      (-47 /39 TIMES 25.39MS (8E) OR 30.72MS (8A)
1540 3357      DCA      DLY2      /SET UP DELAY COUNTER
1541 3360      DCA      DLY3      /CLEAR DELAY COUNTER
1542 6102      DLPB,    SPL      /SKIP ON POWER LOW
1543 7410      SKP
1544 6001      ION      /SKIP ON POWER OK
1545 2360      ISZ      DLY3      /INTERRUPT ON FOR POWER FAIL
1546 5342      JMP      DLPB      /INCREMENT 4096 COUNTER
1547 2357      ISZ      DLY2      /KEEP INCREMENTING 4096 COUNTER
1550 5342      JMP      DLPB      /INCREMENT TIMES 4096 COUNTER
                                   /KEEP INCREMENTING UNTIL 1 SECOND DELAY

```

1556	0000	DLY1,	0	/SECONDS COUNTER
1557	0000	DLY2,	0	/ONE SECOND COUNTER
1560	0000	DLY3,	0	/1/64 SECOND COUNTER

1572	7731
1573	7774
1574	0030
1575	0057
1576	0111
1577	0203
	1600

```

/
/
/*****

```

```

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

```

CALL BY: DEC4

```

1600 0000      XDEC4, 0
1601 7300      CLA CLL
1602 3263      DCA     LNK1           /CLEAR LINK SAVE AREA
1603 1377      TAD     (-5
1604 3261      DCA     DIGCNT        /INIT DIGIT COUNTER
1605 3260      DCA     DECBLD        /CLEAR CONVERTED DIGIT AREA
1606 3060      DCA     DIGFLG        /CLEAR DIGIT INPUT FLAG

/
1607 4435      DEC4LP, INPUT          /GO WAIT FOR CHARACTER INPUT
1610 1106      TAD     M215
1611 7450      SNA
1612 5253      JMP     DEC4EX        /SKIP IF NOT <CR>
1613 1376      TAD     (-43         /<CR> GO TO EXIT
1614 7510      SPA
1615 5245      JMP     DEC4ER        /SKIP IF > OR = 260
1616 1375      TAD     (-12         /NOT DIGIT - START OVER
1617 7700      SMA CLA
1620 5245      JMP     DEC4ER        /SKIP IF < 272
1621 7040      CMA
1622 3060      DCA     DIGFLG        /NOT DIGIT OR <CR> - START OVER
1623 1045      TAD     DIGFLG        /SET DIGIT INPUT FLAG
1624 0374      AND     INCHAR        /GET INPUT CHARACTER AGAIN
1625 3262      DCA     (17           /MASK TO BINARY NUMBER
1626 3262      DCA     DECTMP        /STORE BINARY CHARACTER

```



1626	1260	TAD	DECBLD	/GET CONVERTED DIGITS
1627	7106	CLL	RTL	/MULTIPLY BY 4
1630	1260	TAD	DECBLD	/ADD DIGITS (NOW TIMES 5)
1631	7104	CLL	RAL	/MULTIPLY BY 2
1632	1262	TAD	DECTMP	/ADD NEXT DIGIT
1633	3260	DCA	DECBLD	/STORE CONVERTED DIGITS
1634	7420	SNL		/SKIP IF > 4095
1635	5243	JMP	.+6	/NOT > 4095
1636	7301	CLA	CLL IAC	/SET BIT 11 FOR EXIT
1637	3263	DCA	LNK1	/SAVE FOR EXIT
1640	1260	TAD	DECBLD	/GET CONVERTED NUMBER
1641	7640	SZA	CLA	/SKIP IF = 4096
1642	5245	JMP	DEC4ER	/ > 4096 NOT VALID
1643	2261	ISZ	DIGCNT	/INCREMENT DIGIT COUNT - SKIP IF 5 DIGITS
1644	5207	JMP	DEC4LP	/CONTINUE
/				
1645	7300	DEC4ER,	CLA CLL	
1646	3060	DCA	DIGFLG	/CLEAR DIGIT INPUT FLAG
1647	3263	DCA	LNK1	/CLEAR 4096 LINK INDICATOR WORD
1650	1075	TAD	K0277	
1651	4432	PRINT		/PRINT "?"
1652	5201	JMP	XDEC4+1	/START OVER.
/				
1653	7300	DEC4EX,	CLA CLL	
1654	1263	TAD	LNK1	/GET LINK SAVED
1655	7010	RAR		/PUT IT INTO LINK
1656	1260	TAD	DECBLD	/GET CONVERTED NUMBER
1657	5600	JMP	I XDEC4	/EXIT
/				
1660	0000	DECBLD,	0	
1661	0000	DIGCNT,	0	
1662	0000	DECTMP,	0	
1663	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				
/				
/				
1664	0000	XOCT1,	0	
1665	7300	CLA	CLL	
1666	6214	RDF		/READ DATA FIELD
1667	1110	TAD	KXCDI	/ADD CDI INSTRUCTION
1670	3317	DCA	OCTOUT	/SAVE CDI TO CALLING FLD
1671	6211	CDF	10	/DATA FLD 1
1672	3060	TRYOCT,	DCA	DIGFLG
1673	4435	INPUT		/CLEAR DIGIT INPUT FLAG
1674	1106	TAD	M215	/GO WAIT FOR KEYBOARD INPUT

```

1675 7450      SNA                /SKIP IF NOT <CR>
1676 5317      JMP      OCTOUT    /RETURN AC = 0
1677 1376      TAD      (-43
1700 7510      SPA                /SKIP IF > OR = 260
1701 5321      JMP      OCT1ER    /NOT OCTAL DIGIT - START OVER
1702 1373      TAD      (-10
1703 7700      SMA CLA            /SKIP IF < 270
1704 5321      JMP      OCT1ER    /NOT OCTAL DIGIT - START OVER
1705 7040      CMA
1706 3060      DCA      DIGFLG    /SET DIGIT INPUT FLAG
1707 1045      TAD      INCHAR    /GET CHARACTER AGAIN
1710 0064      AND      K07       /MASK TO OCTAL DIGIT
1711 3326      DCA      OCTONE    /SAVE OCTAL DIGIT
1712 4435      INPUT            /GO WAIT FOR <CR>
1713 1106      TAD      M215
1714 7640      SZA CLA            /SKIP IF <CR>
1715 5321      JMP      OCT1ER    /NOT <CR> - START OVER
1716 1326      TAD      OCTONE    /GET OCTAL DIGIT
1717 7402      OCTOUT, HLT/CDI    /CDI TO CALLING FLD MODIFIED
1720 5664      JMP I   XOCT1     /EXIT

/
1721 7300      OCT1ER, CLA CLL
1722 3060      DCA      DIGFLG    /CLEAR DIGIT INPUT FLAG
1723 1075      TAD      K0277
1724 4432      PRINT            /PRINT "?"
1725 5272      JMP      TRYOCT    /START OVER

/
1726 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)
/                      0010        (FIELD THE TABLE IS IN)
/                      7           (NUMBER OF TABLE ENTRIES)
/                      7777        (VALUE TO PLACE IN TABLE)
/                      (RETURN)
/
/
1727 0000      XTFIL, 0
1730 7340      CLA CLL CMA
1731 1727      TAD I   XTFIL      /GET STARTING ADDRESS OF TABLE-1
1732 3011      DCA      INDX11    /SET UP TABLE POINTER
1733 2327      ISZ      XTFIL      /INCREMENT PARAMETER POINTER
1734 1727      TAD I   XTFIL      /GET TABLE FIELD

```

```

1735 1107      TAD      KXCDF      /ADD CDF INSTRUCTION
1736 3346      DCA      XCDF1      /SAVE CDF TO TABLE FIELD FOR EXECUTION
1737 2327      ISZ      XTFIL      /INCREMENT PARAMETER POINTER
1740 1727      TAD I    XTFIL      /GET NUMBER OF TABLE ENTRIES
1741 7041      CIA
1742 3053      DCA      CNT2      /SET UP COUNTER
1743 2327      ISZ      XTFIL      /INCREMENT PARAMETER POINTER
1744 1727      TAD I    XTFIL      /GET WORD TO FILL TABLE WITH
1745 3356      DCA      TBLWRD     /SAVE TABLE WORD VALUE
1746 7402      XCDF1,  HLT/CDF     /MODIFIED CDF TO TABLE FIELD
1747 1356      XTLP,   TAD      TBLWRD /GET WORD VALUE
1750 3411      DCA I    INDX11     /FILL A TABLE WORD
1751 2053      ISZ      CNT2      /INCREMENT LOCATION COUNTER
1752 5347      JMP      XTLP      /CONTINUE FILLING TABLE
1753 6211      CDF      10        /CHANGE BACK TO PROGRAM DATA FIELD
1754 2327      ISZ      XTFIL      /BUMP RETURN POINTER
1755 5727      JMP I    XTFIL      /EXIT

```

```

1756 0000      /
                TBLWRD, 0
                /

```

```

                /*****
                /

```

```

                /ROUTINE TO CHECK FOR ON APT
                /

```

```

                /      CALLED BY:      APTCK
                /

```

```

                /RETURN CALL+1 IF ON APT
                /RETURN CALL+2 IF NOT ON APT
                /

```

```

1757 0000      CKAPT,  0
1760 7300      CLA CLL
1761 1022      TAD      HDW2
1762 7700      SMA CLA
1763 2357      ISZ      CKAPT
1764 5757      JMP I    CKAPT

```

```

                /
                /
                /*****
                /

```

```

1773 7770
1774 0017
1775 7766
1776 7735
1777 7773
2000

```

```

                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 "\*".

```

2000 0000      MESPRT, 0
2001 6214      RDF      /READ CALLING DATA FIELD
2002 1110      TAD      KXCDI      /ADD CDI TO CALLING FIELD BASE CODE
2003 3252      DCA      MESOUT     /SAVE CDI TO CALLING FLD INSTRUCTION
2004 1600      TAD I     MESPRT     /GET MESSAGE ADDRESS
2005 6211      CDF      10      /CDF TO PRGM FLD
2006 3255      DCA      MESPNT     /SAVE MESSAGE ADDRESS AS POINTER
2007 6000      SKON      /SKIP IF INT ON AND TURN OFF
2010 7301      CLA CLL IAC      /INTERRUPT NOT ON SET BIT 11
2011 1377      TAD      (ION      /GET ION INSTRUCTION
2012 3251      DCA      MESINT     /SAVE FOR EXECUTION UPON RETURN
2013 7240      STA      /AC=7777
2014 3046      DCA      INMODE     /SET MESSAGE ACTIVE FLAG
2015 7344      CLA CLL CMA RAL    /AC=-2
2016 3254      DCA      MESCNT     /SET UP 6-BIT CHARACTER COUNTER
2017 2200      ISZ      MESPRT     /INCREMENT RETURN ADDRESS
2020 1655      MESLUP, TAD I     MESPNT /GET TWO 6-BIT PACKED ASCII CHARACTERS
2021 2254      ISZ      MESCNT     /INCREMENT CHARACTER COUNTER
2022 5231      JMP      MESBSW     /FIRST CHARACTER OF TWO CHARACTERS
2023 2255      ISZ      MESPNT     /2ND CHARACTER INCREMENT MESSAGE POINTER
2024 3256      DCA      MESTMP     /SAVE CHARACTER
2025 7344      CLA CLL CMA RAL    /AC=-2
2026 3254      DCA      MESCNT     /RESET CHARACTER COUNTER
2027 1256      TAD      MESTMP     /GET CHARACTER
2030 7410      SKP      /SKIP THE BYTE SWAP
2031 7002      MESBSW, BSW      /SWAP 6 BIT BYTES FOR PRINTING
2032 0376      AND      (77      /MASK OFF LEFT BYTE
2033 7450      SNA      /SKIP IF NOT END OF MESSAGE
2034 5250      JMP      MESEXT     /END OF MESSAGE GO TO EXIT
2035 1375      TAD      (-43      /SUBTRACT 43
2036 7450      SNA      /SKIP IF NOT "*"
2037 5246      JMP      .+7      /IT WAS "*" GO DO A <CR> AND <LF>
2040 1374      TAD      (3      /ADD 3
2041 7510      SPA      /SKIP IF ASCII CODE IS 2XX
2042 1373      TAD      (100     /ADD 100 ASCII CODE IS 3XX
2043 1073      TAD      K0240     /ADD 40 TO RESTORE ASCII CODE AND ADD 200
                                   /FOR 8 BIT ASCII
2044 4432      PRINT     /PRINT THE CHARACTER
2045 7410      SKP      /SKIP NOT DOING A <CR><LF>
2046 4434      CRLF      /GO DO A <CR> AND <LF>
2047 5220      JMP      MESLUP     /CONTINUE PRINTING
2050 3046      MESEXT, DCA      INMODE /CLEAR MESSAGE FLAG
2051 7402      MESINT, HLT/ION/IOF /IOF OR ION DEPENDING ON INT STATE UPON ENTRY
2052 7402      MESOUT, HLT/CDI    /MODIFIED CDI TO CALLING FLD
2053 5600      JMP I      MESPRT     /RETURN
2054 0000      MESCNT, 0      /CHARACTER COUNTER
2055 0000      MESPNT, 0      /MESSAGE POINTER
2056 0000      MESTMP, 0      /TEMPORARY STORAGE
/
/
/*****
/

```

/ROUTINE TO EXECUTE A CARRIAGE RETURN AND LINE FEED

```

2057 0000  XCRLF, 0
2060 7300      CLA CLL
2061 6214      RDF          /READ CALLING DATA FLD
2062 1110      TAD          KXCDI      /ADD CDI INSTRUCTION
2063 3271      DCA          CROUT      /SAVE FOR EXECUTION
2064 6211      CDF          10         /CDF TO FLD 1
2065 1072      TAD          K0215      /GET ASCII CODE FOR CARRIAGE RETURN
2066 4432      PRINT        /EXECUTE A CARRIAGE RETURN
2067 1071      TAD          K0212      /GET ASCII CODE FOR LINE FEED
2070 4432      PRINT        /EXECUTE A LINE FEED
2071 7402      CROUT, HLT/CDI      /MODIFIED CDI TO CALLING FLD
2072 5657      JMP I    XCRLF      /RETURN

```

/\*\*\*\*\*

/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 CNA

(NO RETURN)

DCA ANSFLG (STORE ANSWER AS 0=YES 7777=NO)

```

2073 0000  XYESNO, 0
2074 6214      RDF          /READ CALLING DATA FLD
2075 1110      TAD          KXCDI      /ADD CDI TO CALLING FLD BASE CODE
2076 3326      DCA          YNOUT      /SAVE CDI TO CALLING FLD FOR EXECUTION
2077 6211      CDF          10         /CDF TO PRGM FLD
2100 4435      TRYYN, INPUT      /GO WAIT FOR INPUT OF Y OR N
2101 1372      TAD          (-316
2102 7450      SNA          /SKIP IF NOT (N)0
2103 5313      JMP          NEXIT      /IT WAS (N)0 GO TO NO EXIT
2104 1371      TAD          (-13
2105 7650      SNA CLA      /SKIP IF NOT (N)0 OR (Y)ES
2106 5321      JMP          YEXIT      /IT WAS (Y)ES GO TO YES EXIT
2107 4434      NOTYN, CRLF      /DO A <CR> AND <LF>
2110 1370      TAD          (277
2111 4432      PRINT        /PRINT "?"
2112 5300      JMP          TRYYN      /TRY AGAIN

2113 4435      NEXIT, INPUT      /WAIT FOR INPUT OF <CR>
2114 1106      TAD          M215      /ADD ASCII NEGATED CODE FOR <CR>
2115 7640      SZA CLA      /SKIP IF <CR> TYPED
2116 5307      JMP          NOTYN      /NOT <CR> TRY AGAIN
2117 2273      ISZ          XYESNO      /INCREMENT RETURN FOR "NO" EXIT
2120 5325      JMP          YNEXIT      /GO TO EXIT

2121 4435      YEXIT, INPUT      /WAIT FOR INPUT OF <CR>
2122 1106      TAD          M215

```

```

2123 7640          SZA CLA          /SKIP IF <CR>
2124 5307          JMP      NOTYN    /NOT <CR> TRY AGAIN FOR "Y" OR "N"
2125 4434          YNEXIT, CRLF      /DO A <CR> AND <LF>
2126 7402          YNOUT,  HLT/CDI    /MODIFIED CDI TO CALLING FLD
2127 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
/
2130 0000          XPRINT, 0
2131 3367          DCA      PRNTMP    /SAVE CHAR TO PRINT
2132 7004          RAL          /GET LINK IN AC 11
2133 3111          DCA      LINKSV    /SAVE FOR RETURN
2134 6000          SKON          /SKIP IF INT ON AND TURN OFF
2135 7301          CLA CLL IAC
2136 1377          TAD      (ION      /GET ION INSTRUCTION
2137 3365          DCA      PRNINT     /SAVE ION OR AND 0 FOR EXIT
2140 6214          RDF          /READ CALLING DATA FLD
2141 1110          TAD      KXCDI     /ADD CDI BASE CODE
2142 3364          DCA      PRNOUT     /SAVE CDI TO CALLING FLD INSTRUCTION
2143 6211          CDF      10        /DATA FLD 1
2144 1367          TAD      PRNTMP    /GET CHARACTER TO PRINT
2145 6046          TLS          /TRANSMIT A CHARACTER
2146 6102          SPL          /SKIP ON POWER LOW
2147 7410          SKP          /SKIP IF POWER OK
2150 6001          ION          /TURN ON INTERRUPT SYSTEM - POWER LOW
2151 6041          TSF          /TEST TELEPRINTER FLAG
2152 5346          JMP      .-4       /WAIT FOR TELEPRINTER FLAG
2153 6042          TCF          /CLEAR TELEPRINTER FLAG
2154 6031          KSF          /TEST KEYBOARD FLAG
2155 5361          JMP      .+4       /KEYBOARD FLAG NOT SET
2156 4444          VT278B
2157 6036          KRB          /KEYBOARD FLAG SET  READ KEYBOARD BUFFER
2160 4441          C8CAL        /GO TO CONSOLE PACKAGE IF ACTIVE
2161 7200          CLA
2162 1111          TAD      LINKSV    /GET LINK
2163 7110          CLL RAR        /RESTORE LINK
2164 7402          PRNOUT, HLT/CDI    /CDI TO CALLING FLD MODIFIED
2165 7402          PRNINT, HLT/ION/IOF /RESTORE INTERRUPT STATE
2166 5730          JMP I    XPRINT    /RETURN
2167 0000          PRNTMP, 0
/
/
/
2170 0277
2171 7765
2172 7462
2173 0100
2174 0003
2175 7735
2176 0077

```

2177 6001  
2200

SEQ 155

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)

2200 0000 XLCHK, 0  
2201 3267 DCA CHKTMP /SAVE VALUE TO CHECK  
2202 1600 TAD I XLCHK /GET UPPER LIMIT  
2203 3266 DCA UPCHK /SAVE UPPER LIMIT  
2204 2200 ISZ XLCHK /POINT TO LOWER LIMIT  
2205 1267 TAD CHKTMP /GET VALUE TO CHECK  
2206 7710 SPA CLA /SKIP IF POSITIVE  
2207 5226 JMP VALNEG /GO CHECK NEGATIVE VALUE  
2210 1266 TAD UPCHK /GET UPPER LIMIT  
2211 7710 SPA CLA /SKIP IF POSITIVE - ELSE GREATER THAN VALUE TO CHECK  
2212 5220 JMP LOCHK /GO CHECK LOWER LIMIT  
2213 1267 TAD CHKTMP /GET VALUE TO CHECK  
2214 7041 CIA /NEGATE  
2215 1266 TAD UPCHK /ADD UPPER LIMIT  
2216 7710 SPA CLA /SKIP IF VALUE < OR = UPPER LIMIT  
2217 5245 JMP BADEX1 /VALUE > UPPER LIMIT - GO TO BAD EXIT  
  
2220 1600 LOCHK, TAD I XLCHK /GET LOWER LIMIT  
2221 7041 CIA /NEGATE  
2222 1267 TAD CHKTMP /ADD VALUE TO CHECK  
2223 7710 SPA CLA /SKIP IF VALUE > OR = LOWER LIMIT  
2224 5245 JMP BADEX1 /VALUE < LOWER LIMIT - GO TO BAD EXIT  
2225 5242 JMP GOODEX /VALUE OK - GO TO GOOD EXIT  
  
2226 1266 VALNEG, TAD UPCHK /GET UPPER LIMIT  
2227 7700 SMA CLA /SKIP IF NEGATIVE  
2230 5245 JMP BADEX1 /VALUE TO CHECK IS > UPPER LIMIT  
2231 7240 CLA CMA  
2232 1267 TAD CHKTMP /VALUE TO CHECK - 1 IN AC  
2233 7700 SMA CLA /SKIP IF NOT 4000(8)  
2234 5242 JMP GOODEX /IT WAS 4000(8) MUST BE < OR = UPPER LIMIT

```

2235 1267      TAD      CHKTMP      /GET VALUE TO CHECK
2236 7041      CIA              /MAKE IT POSITIVE
2237 1266      TAD      UPCHK      /ADD NEGATIVE UPPER LIMIT
2240 7710      SPA CLA      /SKIP IF VALUE < OR = UPPER LIMIT
2241 5245      JMP      BADEX1     /VALUE > UPPER LIMIT - GO TO BAD EXIT
/
2242 2200      GOODEX, ISZ      XLCHK /INCREMENT RETURN FOR GOOD EXIT
2243 2200      ISZ      XLCHK      /INCREMENT RETURN
2244 5600      JMP I      XLCHK      /EXIT - VALUE WITHIN LIMITS
/
2245 4424      BADEX1, MESSAGE
2246 4235      LIMEXD
2247 4424      MESSAGE
2250 4246      HIMSG
2251 7240      CLA CMA
2252 1200      TAD      XLCHK
2253 3200      DCA      XLCHK      /BACK UP POINTER TO UPPER LIMIT
2254 1600      TAD I      XLCHK      /GET UPPER LIMIT
2255 4436      DECPRT      /PRINT UPPER LIMIT
2256 4424      MESSAGE
2257 4252      LOMSG
2260 2200      ISZ      XLCHK      /LOW -
2261 1600      TAD I      XLCHK      /INCREMENT TO LOWER LIMIT
2262 4436      DECPRT      /GET LOWER LIMIT
2263 4434      CRLF
2264 2200      ISZ      XLCHK      /PRINT LOWER LIMIT
2265 5600      JMP I      XLCHK      /<CR> AND <LF>
/                                     /INCREMENT FOR BAD RETURN
/                                     /EXIT
2266 0000      UPCHK, 0          /TEMPORARY STORAGE OF UPPER LIMIT
2267 0000      CHKTMP, 0        /TEMPORARY STORAGE OF VALUE TO CHECK
/
/
/
/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
/
/
2270 0000      XINPUT, 0
2271 6000      SKON
2272 7301      CLA CLL IAC      /SKIP IF INTERRUPT ON AND TURN OFF
2273 1377      TAD      (ION      /INTERRUPT NOT ON SET BIT 11
2274 3311      DCA      INPINT    /GET ION INSTRUCTION
2275 6102      SPL
2276 7410      SKP
2277 6001      ION
2278 6031      KSF
2279 5275      JMP      .-4      /SAVE ION OR AND 0 INSTRUCTION FOR EXIT
2280 6036      KRB
2281 0070      AND      K0177     /SKIP ON POWER LOW
2282 1376      TAD      (200      /SKIP IF POWER OK
2283 3045      DCA      INCHAR    /TURN ON INTERRUPT SYSTEM, POWER IS LOW
2284 1045      TAD      INCHAR    /SKIP ON KEYBOARD FLAG
/                                     /WAIT FOR KYBD FLG OR POWER LOW
/                                     /READ KEYBOARD BUFFER
/                                     /MASK TO 7 BITS
/                                     /SET PARITY BIT
/                                     /SAVE ASCII CHAR
/                                     /GET CHARACTER

```



```

2307 4441      C8CAL      /CHECK FOR CONTROL CHARACTER
2310 7000      NOP        /CONSOLE NOT ACTIVE RETURN
2311 7402      INPINT, HLT/ION/AND 0
2312 1045      TAD        INCHAR      /GET CHARACTER
2313 5670      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)
/
2314 0000      PTSET, 0
2315 3340      DCA        ACSAV      /SAVE AC
2316 7004      RAL        /GET LINK
2317 3341      DCA        LKSAV      /SAVE LINK
2320 1375      TAD        (DRVPT     /GET ADDRESS OF DRV STATE INDEX TABLE
2321 1055      TAD        CURDV      /ADD CURRENT DRIVE # AS INDEX (0-3)
2322 3051      DCA        TBLPT      /SAVE POINTER TO INDEX TABLE
2323 6201      CDF        00
2324 1451      TAD I      TBLPT      /GET DRIVE STATE TABLE ADDRESS FROM INDEX TABLE
2325 3051      DCA        TBLPT      /SAVE AS POINTER
2326 6211      CDF        10
2327 1714      TAD I      PTSET      /GET DRV STATE TABLE OFFSET FROM CALL+1
2330 1051      TAD        TBLPT      /ADD OFFSET TO START OF DRV STATE TABLE ADDRESS
2331 3051      DCA        TBLPT      /SET UP DRV TBL POINTER
2332 2314      ISZ        PTSET      /INCREMENT FOR RETURN
2333 1341      TAD        LKSAV      /GET LINK
2334 7010      RAR        /RESTORE LINK
2335 1340      TAD        ACSAV      /GET AC UPON ENTRY
2336 6201      CDF        00
2337 5714      JMP I      PTSET      /RETURN

/
/
2340 0000      ACSAV, Q
2341 0000      LKSAV, 0
/
/
/*****
/
/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
/
/      CALLED BY:      C8CAL

```

```

/
/
2342 0000  CALC8, 0
2343 3365      DCA      CHRTEM      /SAVE AC
2344 6000      SKON                      /SKIP IF INTERRUPT ON AND TURN OFF
2345 7301      CLA CLL IAC
2346 1377      TAD      (ION
2347 3363      DCA      C8OUT      /SAVE ION OR IOF FOR EXIT
2350 1022      TAD      22      /GET HDW2
2351 0374      AND      (400      /TEST FOR BIT 3=1 CONSOLE ACTIVE
2352 7650      SNA CLA      /SKIP IF CONSOLE ACTIVE
2353 5363      JMP      C8OUT      /RETURN CALL+1 CONSOLE NOT ACTIVE
2354 1365      TAD      CHRTEM      /GET CHARACTER TO CHECK
2355 4773      JMS      C8ENTR      /GO TO CONSOLE PACKAGE
2356 2342      ISZ      CALC8      /INCREMENT RETURN ADDRESS
2357 1020      TAD      PSR1      /GET PSR1
2360 6201      CDF      00      /CDF TO FLD 0
2361 3772      DCA I (PSR      /COPY SOFTWARE SWITCH REGISTER TO FLD 0 PG 0 LOC 20
2362 6211      CDF      10      /CDF TO PRGM FLD
2363 7402      C8OUT, HLT/ION/IOF      /MODIFIED ION OR IOF FOR EXIT
2364 5742      JMP I  CALC8      /RETURN CALL+2 CONSOLE WAS ACTIVE
/
/
2365 0000  CHRTEM, 0      /AC STORAGE
/
/
/
2372 0020
2373 5222
2374 0400
2375 6710
2376 0200
2377 6001
2400 2400

PAGE
/
/*****
/
/ROUTINE TO CONVERT OCTAL TO DECIMAL OUTPUT - SINGLE WORD PRECISION
/CALLED WITH OCTAL NUMBER TO OUTPUT IN AC
/
/      CALLED BY:      DECPRT
/
/
/
2400 0000  PRTDEC, 0
2401 3237      DCA      DTEMP1      /SAVE BINARY NUMBER TO OUTPUT
2402 3236      DCA      DECDIG      /CLEAR DECIMAL DIGIT STORAGE
2403 1104      TAD      M04
2404 3240      DCA      DCNT1      /SET UP DIGIT COUNTER FOR 4 DIGITS
2405 1231      TAD      ADDRSA      /GET POWER OF TEN TABLE ACCESS INSTRUCTION
2406 3213      DCA      DPNT1      /INIT POWER OF TEN TABLE ACCESS INSTRUCTION
2407 7410      SKP                      /NEXT INSTRUCTION NOT USED YET
2410 3237      DCA      DTEMP1      /SAVE PARTIAL QUOTIENT
2411 7100      CLL
2412 1237      TAD      DTEMP1      /GET BINARY NUMBER TO CONVERT
2413 1232      DPNT1, TAD      TENPWR      /SUBTRACT POWER OF TEN

```

```

2414 7430      SZL
2415 2236      ISZ      DECDIG      /INCREMENT CONVERTED DIGIT
2416 7430      SZL
2417 5210      JMP      DPNT1-3      /CONTINUE CONVERTING THIS DIGIT
2420 7200      CLA
2421 1236      TAD      DECDIG      /GET CONVERTED DIGIT
2422 1074      TAD      K0260      /ADD ASCII BASE CODE
2423 4432      PRINT     /PRINT CONVERTED DIGIT
2424 3236      DCA      DECDIG      /CLEAR CONVERTED DIGIT STORAGE
2425 2213      ISZ      DPNT1      /INCREMENT POWER OF TEN TABLE ACCESS INSTRUCTION
2426 2240      ISZ      DCNT1      /INCREMENT DIGIT COUNTER
2427 5212      JMP      DPNT1-1      /START NEXT DIGIT
2430 5600      JMP I     PRTDEC     /EXIT

/
2431 1232      ADDRSA, TAD      TENPWR      /POWER OF TEN TABLE ACCESS INSTRUCTION
/
2432 6030      TENPWR, -1750      /-1000
2433 7634      -144              /-100
2434 7766      -12              /-10
2435 7777      -1              /-1

/
2436 0000      DECDIG, 0
2437 0000      DTEMP1, 0
2440 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, EMSG      /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
/
/
2441 0000      PRNSTA, 0
2442 7300      CLA CLL
2443 6214      RDF      /READ DATA FLD
2444 1110      TAD      KXCDI      /ADD CDI INSTRUCTION FOR RETURN
2445 3331      DCA      STACDI      /SAVE FOR EXIT
2446 1641      TAD I     PRNSTA      /GET START OF MESSAGE ADDRESS
2447 3334      DCA      EPNTR      /SAVE AS POINTER

```

2450	6201	CDF	00		
2451	1740	TAD I	CDRV	/GET CURRENT DRIVE NUMBER	
2452	3055	DCA	CURDV	/SAVE IN FLD 1 LOCATION	
2453	6211	CDF	10	/DF=1	
2454	1105	MLP1, TAD	M06		
2455	3335	DCA	LCNTR	/SET UP LINE COUNTER	
2456	1734	MLP2, TAD I	EPNTR	/GET MESSAGE ADDRESS	
2457	7450	SNA		/SKIP IF NOT END OF MSG	
2460	5326	JMP	STAOUT	/END OF MSG EXIT	
2461	3263	DCA	MADR	/SAVE MSG ADDRESS	
2462	4424	MESSAGE		/CALL MESSAGE PRINT ROUTINE	
2463	7402	MADR, HLT/MSG	ADRS	/MESSAGE ADDRESS	
2464	2334	ISZ	EPNTR	/INCREMENT MSG TBL POINTER	
2465	1734	TAD I	EPNTR	/GET ADDRESS OF SUBROUTINE TO CALL FOR DATA PRINT	
2466	3333	DCA	PDATA	/SAVE ADDRESS OF PRINT SUBROUTINE	
2467	7240	STA		/SET THE AC =7777	ADDITION FOR CNTRL 8 HP 016
2470	3046	DCA	INMODE	/SET THE FLAG ACTIVE INDICATING PRINTING IN PROGRESS	HP 016
2471	2334	ISZ	EPNTR	/INCREMENT MSG TBL POINTER	
2472	1734	TAD I	EPNTR	/GET ADDRESS OF DATA TO PRINT	
2473	7500	SMA		/SKIP IF GREATER THAN OR EQUAL TO 4000	
2474	5300	JMP	GETDAT	/GO GET DATA TO PRINT	
2475	1337	TAD	MDRV0	/ADD +DRIVE0 TO -(DRIVE0+OFFSET) = OFFSET	
2476	7500	SMA		/SKIP IF DATA NOT IN DRIVE STATE TBL	
2477	5307	JMP	TBLPRN	/GO GET DATA FROM DRV STATE TBL	
2500	7300	GETDAT, CLA CLL			
2501	1734	TAD I	EPNTR	/GET DATA ADDRESS	
2502	3336	DCA	EPNTR1	/SAVE AS POINTER	
2503	6201	CDF	00		
2504	1736	TAD I	EPNTR1	/GET DATA TO PRINT	
2505	6211	CDF	10		
2506	5314	JMP	PRNDAT	/GO PRINT DATA	
2507	3311	TBLPRN, DCA	TBLSET	/SAVE FOR SETPT CALL	
2510	4440	SETPT			
2511	7402	TBLSET, HLT/TBL	INDX		
2512	1451	TAD I	TBLPT	/GET VALUE FROM DRV STATE TBL	
2513	6211	CDF	10		
2514	4733	PRNDAT, JMS I	PDATA	/GO TO ROUTINE TO PRINT DATA	
2515	2334	ISZ	EPNTR	/INCREMENT MESSAGE TABLE POINTER	
2516	1073	TAD	K0240		
2517	4432	PRINT		/ONE SPACE	
2520	1073	TAD	K0240		
2521	4432	PRINT		/TWO SPACES	
2522	2335	ISZ	LCNTR	/INCREMENT LINE COUNTER	
2523	5256	JMP	MLP2	/CONTINUE THIS LINE	
2524	4434	CRLF		/<CR LF>	
2525	5254	JMP	MLP1	/START NEXT LINE	
2526	4434	STAOUT, CRLF		/<CR LF>	
2527	2241	ISZ	PRNSTA	/INCREMENT RETURN	
2530	3046	DCA	INMODE	/CLEAR PRINTING INPROGRESS FLAG	HP 016
2531	7402	STACDI, HLT/CDI		/MODIFIED CDI TO CALLING FIELD	
2532	5641	JMP I	PRNSTA	/RETURN	
2533	0000	PDATA, 0		/ADDRESS OF PRINT ROUTINE	

2534	0000	EPNTR, 0	/ERROR MESSAGE TABLE POINTER
2535	0000	LCNTR, 0	/LINE COUNTER
2536	0000	EPNTR1, 0	/USED FOR DOUBLE INDIRECT ADDRESSING
2537	1064	MDRV0, -DRIVE0	
2540	0024	CDRV, CURDRV	

/

/

/

/\*\*\*\*\*

2541	3606	ERTBL1, ERMMSG	/ER:
2542	1476	ACPRNT	/PRNT 4 OCT DIG
2543	0030	ERREG	/ERROR REGISTER STORAGE
2544	3611	CAMSG	/CA:
2545	1476	ACPRNT	
2546	0031	COMDA	/COMMAND A REG. STORAGE
2547	3614	CBMSG	/CB:
2550	1476	ACPRNT	
2551	0032	COMDB	/COMMAND B REG. STORAGE
2552	3600	SAMSG	/SA:
2553	1476	ACPRNT	
2554	0033	ENDSC	/FINAL SECTOR REGISTER STORAGE
2555	3630	IWCMSG	/IWC:
2556	1476	ACPRNT	
2557	6744	DRIVE0+WRDCNT	/INITIAL WORD COUNT STORAGE ADDRESS
2560	3633	FNCMSG	/FNC:
2561	1476	ACPRNT	
2562	0034	ENDWD	/FINAL WORD COUNT READ FROM CONTROLLER
2563	3617	CFMSG	/CF:
2564	1476	ACPRNT	
2565	0130	FLGSAV	/CONTROLLER FLAGS LAST INTERRUPT
2566	0000	0	/END OF MESSAGES

/

2567	3701	ERTBL2, RDSMSG	/RDS:
2570	2400	PRTDEC	/PRINT 4 DECIMAL DIGITS
2571	6742	DRIVE0+READS	/NUMBER OF READS STORAGE ADDRESS
2572	3704	WRTMSG	/WRTS:
2573	2400	PRTDEC	
2574	6743	DRIVE0+WRTS	/NUMBER OF WRITES STORAGE ADDRESS
2575	3710	SKMSG	/SKS:
2576	2400	PRTDEC	
2577	6724	DRIVE0+SEEKS1	/NUMBER OF SEEKS CURRENT PASS STORAGE ADDRESS
2600	3622	HEMSG	/HE:
2601	2400	PRTDEC	/ADDRESS OF PRINT DECIMAL ROUTINE
2602	6726	DRIVE0+HRDERR	/DRIVE STATE TABLE STARTING ADDRESS PLUS TABLE OFFSET
2603	3625	SEMSG	/SE:
2604	2400	PRTDEC	
2605	6727	DRIVE0+SFTERR	/SOFT ERROR STORAGE
2606	3644	DEMSG	/DE:
2607	2400	PRTDEC	
2610	6730	DRIVE0+DRVERR	/DRIVE ERROR STORAGE
2611	3647	SKMSG	/SKE:
2612	2400	PRTDEC	
2613	6731	DRIVE0+SEKERR	/SEEK ERROR STORAGE
2614	3713	DATAM	/DAT:

```

2615 2400      PRTDEC
2616 6732      DRIVE0+DATERR /DATA ERROR STORAGE ADDRESS
2617 3652      TEMSG /TE:
2620 2400      PRTDEC
2621 6733      DRIVE0+TRKERR /TRACKING ERROR STORAGE
2622 3655      DCRCM /DCRC:
2623 2400      PRTDEC
2624 6734      DRIVE0+DCRCER /DATA CRC ERROR STORAGE ADDRESS
2625 3661      HRCM /HCRC:
2626 2400      PRTDEC
2627 6735      DRIVE0+HRCER /HEADER CRC ERROR STORAGE ADDRESS
2630 3665      DLTMSG /DLT:
2631 2400      PRTDEC
2632 6736      DRIVE0+DLTERR /DATA LATE ERROR TALLY STORAGE ADDRESS
2633 3670      OPIMSG /OPI:
2634 2400      PRTDEC
2635 6737      DRIVE0+OPIERR /OPERATION INCOMPLETE ERROR TALLY STORAGE ADDRESS
2636 3673      HNFMSG /HNF:
2637 2400      PRTDEC
2640 6740      DRIVE0+HNFERR /HEADER NOT FOUND ERROR TALLY STORAGE ADDRESS
2641 3676      CEMSG /CE:
2642 2400      PRTDEC
2643 6741      DRIVE0+CTLERR /CONTROLLER ERROR TALLY STORAGE ADDRESS
2644 0000      0 /END OF MESSAGE TERMINATOR

/
2645 3636      ERTBL3, S1MSG /S1:
2646 1476      ACPRNT /ADDRESS OF ROUTINE TO PRINT AC CONTENTS
2647 0026      STAT6A /STATUS WORD 1 STORAGE ADDRESS AFTER GET STATUS
2650 3641      S2MSG /S2:
2651 1476      ACPRNT
2652 0027      STAT6B /STATUS WORD 2 STORAGE ADDRESS AFTER GET STATUS
2653 0000      0 /END OF MESSAGE TERMINATOR

/
/
/
/*****
/
/
/ROUTINE TO SAVE ALL NECESSARY DRIVE STATE TABLE ENTRIES
/AND OTHER PARAMETERS
/
/ CALLED BY: JMS I (SAVALL /FROM FLD 0
/
/
2654 0000      SAVALL, 0
2655 1777      TAD I (CURDRV /GET CURRENT DRIVE FROM FLD 0
2656 3055      DCA CURDV /SAVE FOR FLD 1 USE
2657 1776      TAD I (FUNCOD /GET FUNCTION CODE DF=0
2660 3317      DCA FUNSAV /SAVE FUNCTION CODE
2661 1775      TAD I (BITMOD /GET BIT MODE FROM FLD 0
2662 3327      DCA BTSV /SAVE BIT MODE
2663 4440      SETPT /SET TBLPT TO WORD COUNT
2664 0030      WRDCNT /TBL INDX
2665 1451      TAD I TBLPT /GET WORD COUNT
2666 3320      DCA WSV /SAVE WORD COUNT

```

2667	4440	SETPT		/SET TBLPT TO BREAK MA
2670	0031	INITCA		/TBL INDX
2671	1451	TAD I	TBLPT	/GET BREAK MA
2672	3321	DCA	ISV	/SAVE BREAK MA
2673	4440	SETPT		/SET TBLPT TO SECTOR ADDRESS
2674	0032	SECADD		/TBL INDX
2675	1451	TAD I	TBLPT	/GET SECTOR ADDRESS
2676	3322	DCA	SSV	/SAVE SECTOR ADDRESS
2677	4440	SETPT		/SET TBLPT TO COMMAND A SENT
2700	0033	XCOMA		/TBL INDX
2701	1451	TAD I	TBLPT	/GET COMMAND A SENT
2702	3323	DCA	ASV	/SAVE COMMAND A
2703	4440	SETPT		/SET TBLPT TO COMMAND B SENT
2704	0034	XCOMB		/TBL INDX
2705	1451	TAD I	TBLPT	/GET COMMAND B SENT
2706	3324	DCA	BSV	/SAVE COMMAND B
2707	4440	SETPT		/SET TBLPT TO EXPECTED FINAL SECTOR
2710	0040	XENDSC		/TBL INDX
2711	1451	TAD I	TBLPT	/GET EXPECTED FINAL SECTOR
2712	3325	DCA	ENDSV	/SAVE EXPECTED FINAL SECTOR
2713	1774	TAD I	(ERREG	/GET ERROR REGISTER AT TIME OF ERROR
2714	3326	DCA	ERSV	/SAVE ERROR REGISTER AT TIME OF ERROR
2715	6203	CDI	00	/RETURNING TO FLD 0
2716	5654	JMP I	SAVALL	/RETURN TO FLD 0

2717	0000	FUNSAV,	0	/FUNCOD
2720	0000	WSV,	0	/WRDCNT
2721	0000	ISV,	0	/INITCA
2722	0000	SSV,	0	/SECADD
2723	0000	ASV,	0	/XCOMA
2724	0000	BSV,	0	/XCOMB
2725	0000	ENDSV,	0	/XENDSC
2726	0000	ERSV,	0	/ERREG
2727	0000	BTSV,	0	/BITMOD

/  
 /  
 /\*\*\*\*\*  
 /  
 /

/ROUTINE TO RESTORE DRIVE STATE TABLE ENTRIES

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

2730	0000	RESALL,	0	
2731	7300	CLA	CLL	
2732	1317	TAD	FUNSAV	/GET FUNCTION CODE
2733	3776	DCA I	(FUNCOD	/DF=0
2734	1327	TAD	BTSV	/GET BIT MODE
2735	3775	DCA I	(BITMOD	/RESTORE BIT MODE
2736	4440	SETPT		/SET TBLPT TO WORD COUNT
2737	0030	WRDCNT		/TBL INDX
2740	1320	TAD	WSV	/GET WORD COUNT SAVED
2741	3451	DCA I	TBLPT	/RESTORE WORD COUNT

2742	4440	SETPT		/SET TBLPT TO BREAK MA
2743	0031	INITCA		/TBL INDX
2744	1321	TAD	ISV	/GET BREAK MA (INITIAL CURRENT ADDRESS)
2745	3451	DCA I	TBLPT	/RESTORE BREAK MA
2746	4440	SETPT		/SET TBLPT TO SECTOR ADDRESS
2747	0032	SECADD		/TBL INDX
2750	1322	TAD	SSV	/GET SECTOR ADDRESS SAVED
2751	3451	DCA I	TBLPT	/RESTORE SECTOR ADDRESS
2752	4440	SETPT		/SET TBLPT TO COMMAND A SAVED
2753	0033	XCOMA		/TBL INDX
2754	1323	TAD	ASV	/GET COMMAND A SAVED
2755	3451	DCA I	TBLPT	/RESTORE COMMAND A SAVED
2756	4440	SETPT		/SET TBLPT TO COMMAND B SAVED
2757	0034	XCOMB		/TBL INDX
2760	1324	TAD	BSV	/GET COMMAND B SAVED
2761	3451	DCA I	TBLPT	/RESTORE COMMAND B SAVED
2762	4440	SETPT		/SET TBLPT TO EXPECTED FINAL SECTOR
2763	0040	XENDSC		/TBL INDX
2764	1325	TAD	ENDSV	/GET FINAL SECTOR SAVED
2765	3451	DCA I	TBLPT	/RESTORE FINAL SECTOR
2766	1326	TAD	ERSV	/GET ERROR REGISTER SAVED
2767	3774	DCA I	(ERREG	/RESTORE ERROR REGISTER
2770	6203	CDI	00	/RETURNING TO FLD 0
2771	5730	JMP I	RESALL	/RETURN

2774 0030  
2775 0111  
2776 0115  
2777 0024  
3000

PAGE

/ROUTINE TO UPDATE BITS TRANSFERRED COUNTERS  
/IN DRIVE STATE TABLES.

CALLLED BY: JMS UPDAT (FROM FIELD 0)

3000	0000	UPDBIT, 0	
3001	7300	CLA	CLL
3002	1777	TAD I	(DRV60 /GET LAST DRIVE USING CONTROLLER DF=0
3003	6211	CDF	10 /DF=1
3004	3055	DCA	CURDV /SAVE DRIVE FOR USE BY SETPT ROUTINE



3005	4440	SETPT		/SET "TBLPT" TO WORD COUNT OF CURRENT DRIVE
3006	0030	WRDCNT		
3007	1451	TAD I	TBLPT	/GET WORD COUNT
3010	6211	CDF	10	/DF=1
3011	3336	DCA	BITCNT	/SET UP WORD COUNTER
3012	4440	SETPT		/SET "TBLPNT" TO COMMAND B SENT
3013	0034	XCOMB		
3014	1451	TAD I	TBLPT	/GET LAST COMMAND B
3015	6211	CDF	10	/DF=1
3016	0376	AND	(1000	/MASK BIT MODE BIT
3017	7650	SNA CLA		/SKIP IF 8 BIT MODE
3020	7307	CLA CLL	IAC RTL	/12 BIT MODE, ADD 4 BITS PER WORD
3021	1375	TAD	(10	/AT LEAST 8 BITS PER WORD
				/SET UP TO ADD 8 BITS PER WORD
3022	3335	DCA	BITPER	/SET UP BITS PER WORD
3023	4440	SETPT		/SET "TBLPT" TO HIGH ORDER WORD OF BITS
3024	0007	BITS4		/TBL INDX
3025	6211	CDF	10	/DF=1
3026	1051	TAD	TBLPT	
3027	3341	DCA	BPNTR3	/SET UP POINTER TO MULTIPLIER (TIMES 10 BILLION)
3030	4440	SETPT		/SET "TBLPT" TO NEXT WORD FOR BITS
3031	0006	BITS3		/TBL INDX
3032	6211	CDF	10	/DF=1
3033	1051	TAD	TBLPT	/GET ADDRESS OF NEXT TABLE ENTRY
3034	3340	DCA	BPNTR2	/SET UP POINTER TO HIGH ORDER WORD
3035	4440	SETPT		/SET TBLPT TO NEXT TABLE ENTRY
3036	0005	BITS2		/TBL INDX
3037	6211	CDF	10	/DF=1
3040	1051	TAD	TBLPT	/GET TABLE ENTRY ADDRESS
3041	3337	DCA	BPNTR1	/SET UP TABLE POINTER
3042	4440	SETPT		/SET TBLPT TO LOW ORDER WORD
3043	0004	BITS1		/TBL INDX
3044	1451	TAD I	TBLPT	/GET LOW ORDER WORD
3045	1335	TAD	BITPER	/ADD NUMBER OF BITS PER WORD
3046	3451	DCA I	TBLPT	/SAVE UPDATED LOW ORDER BITS
3047	7430	SZL		/SKIP IF NO OVERFLOW
3050	2737	ISZ I	BPNTR1	/INCREMENT MIDDLE ORDER BITS
3051	7410	SKP		
3052	2740	ISZ I	BPNTR2	/INCREMENT HIGH ORDER BITS
3053	7000	NOP		
3054	1740	TAD I	BPNTR2	/GET HIGH ORDER WORD OF BIT COUNT
3055	7041	CIA		
3056	6211	CDF	10	/DF=1
3057	1374	TAD	(1124	/ADD HIGH ORDER WORD IF 10 BILLION
3060	7640	SZA CLA		/SKIP IF POSSIBLE 10 BILLION BITS
3061	5327	JMP	BITEND	/NOT 10 BILLION YET
3062	6201	CDF	00	/DF=0
3063	1737	TAD I	BPNTR1	/GET MIDDLE ORDER WORD OF BIT COUNT
3064	7041	CIA		
3065	6211	CDF	10	/DF=1
3066	1373	TAD	(276	/ADD MIDDLE ORDER WORD OF BIT COUNT
3067	7640	SZA CLA		/SKIP IF POSSIBLE 10 BILLION BITS
3070	5327	JMP	BITEND	/NOT 10 BILLION YET
3071	1372	TAD	(2000	/GET LOW ORDER WORD IF 10 BILLION OR GREATER
3072	7041	CIA		

BITLP,

```

3073 6201      CDF      00      /DF=0
3074 1451      TAD I    TBLPT    /ADD LOW ORDER WORD OF BIT COUNT
3075 7510      SPA      /SKIP IF > OR = 10 BILLION BITS
3076 5327      JMP      BITEND    /NOT YET QUITE 10 BILLION BITS
3077 3451      DCA I    TBLPT    /SET LOW ORDER = LOW ORDER -2000
3100 3737      DCA I    BPNT1    /CLEAR MIDDLE ORDER
3101 3740      DCA I    BPNT2    /CLEAR HIGH ORDER
3102 2741      ISZ I    BPNT3    /INCREMENT MULTIPLIER (TIMES 10 BILLION)
3103 7000      NOP      /SAFETY
3104 6211      CDF      10
3105 4437      APTCK     /CHECK FOR ON APT
3106 5327      JMP      BITEND    /ON APT, EXIT
3107 4424      MESSAGE
3110 4604      PASMMSG
3111 4424      MESSAGE
3112 4075      DRVMSG
3113 1055      TAD      CURDV      /"DRV "
3114 1074      TAD      K0260      /GET CURRENT DRIVE NUMBER
3115 4432      PRINT     /ADD ASCII BASE CODE
3116 1073      TAD      K0240      /PRINT DRIVE NUMBER
3117 4432      PRINT     /GET ASCII CODE FOR SPAC
3120 4424      MESSAGE
3121 4616      DATA
3122 4442      PASPT     /"DATA"
3123 3155      DPAS0     /SET "UPPNT" TO DATA PASS TABLE
3124 2447      ISZ I    UPPNT      /DATA PASS TABLE ADDRESS
3125 7000      NOP      /INCREMENT DATA PASS COUNT
3126 4771      JMS      PASCHK     /IN CASE OF OVERFLOW
3127 6201      /          /GO CHECK FOR END OF PASS
3130 7300      BITEND, CDF      00
3131 2336      CLA      CLL
3132 5244      ISZ      BITCNT     /INCREMENT WORD COUNTER
3133 6203      JMP      BITLP      /CONTINUE UPDATING BIT COUNT
3134 5600      CDI      00         /GOING BACK TO FIELD 0
3135 0000      JMP I    UPDBIT     /EXIT
3136 0000      BITPER, 0
3137 0000      BITCNT, 0
3138 0000      BPNT1, 0
3139 0000      BPNT2, 0
3140 0000      BPNT3, 0
3141 0000      /
3142 0000      /ROUTINE TO SET "UPPNT" TO A TABLE ADDRESS INDEXED USING DRIVE NUMBER
3143 7300      /
3144 1742      /      CALLED BY:    PASPT
3145 1055      /      FOLLOWED BY:   SKPAS0 OR DPAS0 OR APAS0      (START OF TABLE ADDRESS)
3146 3047      /
3147 2342      /      PTPAS, 0
3148 2342      /      CLA      CLL
3149 5742      /      TAD I    PTPAS      /GET START OF TABLE ADDRESS FROM CALL+1
3150 5742      /      TAD      CURDV      /INDEX INTO TABLE USING DRIVE NUMBER
3151 5742      /      DCA      UPPNT      /SET UP POINTER
3152 5742      /      ISZ      PTPAS      /INCREMENT RETURN
3153 5742      /      JMP I    PTPAS      /RETURN

```

```

/*****
/SEEK PASS TABLE
/
3151 0000 SKPAS0, 0
3152 0000 SKPAS1, 0
3153 0000 SKPAS2, 0
3154 0000 SKPAS3, 0
/
/DATA PASS TABLE
/
3155 0000 DPAS0, 0
3156 0000 DPAS1, 0
3157 0000 DPAS2, 0
3160 0000 DPAS3, 0
/
/ACCEPT PASS TABLE
/
3161 0000 APAS0, 0
3162 0000 APAS1, 0
3163 0000 APAS2, 0
3164 0000 APAS3, 0
/
3165 3603 ERTBL4, DNMSG      /DN:
3166 1476      ACPRNT      /PRINT 4 OCTAL DIGITS
3167 0024      CURDRV      /CURRENT DRIVE # STORAGE
3170 0000      0           /END OF TABLE
/
/
3171 3250
3172 2000
3173 0276
3174 1124
3175 0010
3176 1000
3177 0025
3200 3200
PAGE
/
/
/
/*****
/ROUTINE TO UPDATE SEEK OPERATION COUNT IN DRIVE STATE TABLES
/
/CALLED BY:      JMS      SEEKUP (FROM FLD 0)
/
/
3200 0000 UPDSEK, 0
3201 7300      CLA      CLL
3202 1777      TAD I    (DRV60      /DF=0, GET CURRENT DRIVE
3203 6211      CDF      10

```

```

3204 3055      DCA      CURDV      /SET UP CURRENT DRIVE IN FLD 1
3205 4440      SETPT     /SET TBLPT TO SEEK COUNT IN DRV TBL
3206 0010      SEEKS1    /TBL INDX
3207 2451      ISZ I     TBLPT     /INCREMENT SEEK COUNT
3210 1451      TAD I     TBLPT     /GET SEEK COUNT
3211 7041      CIA       /NEGATE FOR CHECK
3212 1376      TAD       (1750     /ADD 1000(10)
3213 7640      SZA       CLA       /SKIP IF 1000 SEEKS (PASS)
3214 5246      JMP       UPSEX     /GO TO EXIT
3215 3451      DCA I     TBLPT     /CLEAR SEEK OPERATION COUNTER
3216 6211      CDF       10
3217 4440      SETPT     /SET TBLPT TO PASS COUNTER
3220 0011      SEEKS2    /TBL INDX
3221 2451      ISZ I     TBLPT     /INCREMENT PASS COUNTER
3222 7000      NOP       /IN CASE OF OVERFLOW
3223 6211      CDF       10
3224 4437      APTCK     /CHECK FOR ON APT
3225 5246      JMP       UPSEX     /ON APT EXIT
3226 4424      MESSAGE   /NOT ON APT, PRINT END OF SEEK PASS MESSAGE
3227 4604      PASMSG    /"END OF PASS"
3230 4424      MESSAGE
3231 4075      DRVMSG    /"DRV"
3232 1055      TAD       CURDV     /GET DRIVE NUMBER
3233 1074      TAD       K0260     /ADD ASCII BASE CODE
3234 4432      PRINT     /PRINT DRIVE NUMBER
3235 1073      TAD       K0240
3236 4432      PRINT     /PRINT A SPACE
3237 4424      MESSAGE
3240 3777      SEKMSG    /"SEEKS"
3241 4442      PASPT     /POINT TO SEEK PASS TABLE
3242 3151      SKPASO    /START OF TABLE
3243 2447      ISZ I     UPNT      /INCREMENT SEEK PASS COUNT
3244 7000      NOP       /IN CASE OF OVERFLOW
3245 4250      JMS       PASCHK    /CHECK FOR DRIVE ACCEPT (1000 SEEKS AND 10 BILLION BITS TRANSFERRED)

/
3246 6203      UPSEX,   CDI       00      /RETURNING TO FLD 0
3247 5600      JMP I     UPDSEK    /EXIT

/
/*****
/
/
/ROUTINE TO CHECK FOR END OF PASS (1000 SEEKS AND 10 BILLION BITS TRANSFERRED)
/
/      CALLED BY:      JMS      PASCHK
/
/
/
PASCHK, 0
3251 7300      CLA       CLL
3252 4437      APTCK     /CHECK FOR ON APT
3253 5650      JMP I     PASCHK    /ON APT, EXIT
3254 4442      PASPT     /SET UPNT TO SEEK PASS COUNT
3255 3151      SKPASO    /TBL ADDRESS
3256 1447      TAD I     UPNT      /GET SEEK PASS COUNT
3257 7650      SNA       CLA       /SKIP IF NOT ZERO
3260 5650      JMP I     PASCHK    /NOT 1000 SEEKS YET, EXIT

```

3261	4442	PASPT		/SET UPNT TO DATA PASS COUNT
3262	3155	DPASO		/DATA PASS COUNT TABLE ADDRESS
3263	1447	TAD I	UPNT	/GET DATA PASS COUNT
3264	7650	SNA	CLA	/SKIP IF NOT ZERO
3265	5650	JMP I	PASCHK	/NO DATA PASS (10 BILLION BITS), EXIT
3266	3447	DCA I	UPNT	/CLEAR DATA PASS SWITCH
3267	4442	PASPT		
3270	3151	SKPASO		
3271	3447	DCA I	UPNT	/CLEAR SEEK PASS SWITCH
3272	4424	MESSAGE		
3273	4604	PASMSG		/"END OF PASS"
3274	4424	MESSAGE		
3275	4075	DRVMSG		/"DRV "
3276	1055	TAD	CURDV	/GET CURRENT DRIVE NUMBER
3277	1074	TAD	K0260	/ADD ASCII BASE CODE
3300	4432	PRINT		/PRINT DRIVE NUMBER
3301	1073	TAD	K0240	
3302	4432	PRINT		/PRINT A SPACE
3303	4424	MESSAGE		
3304	4612	ACPTDM		/"ACPTD"
3305	4442	PASPT		/SET UPNT TO END OF PASS TABLE
3306	3161	APASO		/ADDRESS OF END OF PASS TABLE
3307	2447	ISZ I	UPNT	/UPDATE PASS FLAG
3310	7000	NOP		/IN CASE OF OVERFLOW
3311	4443	GETSR		/GET SWITCHES
3312	0375	AND	(2	/MASK BIT 10, CHECK FOR DROP DRIVE AT END OF PASS
3313	7640	SZA	CLA	/SKIP IF MIGHT BE OK TO DROP DRIVE
3314	5324	JMP	ALLPAS	/DO NOT DROP DRIVE, CHECK FOR END OF PASS HALT
3315	1113	TAD	LMFLG	/GET DROP DRV AT OPERATION LIMITS REACHED FLAG
3316	7640	SZA	CLA	/SKIP IF OK TO DROP DRIVE
3317	5324	JMP	ALLPAS	/DO NOT DROP DRIVE, CHECK FOR END OF PASS HALT
3320	4440	SETPT		/SET TBLPT TO DRIVE STATE
3321	0000	DSTATE		/TBL INDX
3322	3451	DCA I	TBLPT	/DROP DRIVE
3323	6211	CDF	10	
/				
3324	4443	ALLPAS,	GETSR	/GET SWITCHES
3325	0374	AND	(400	/TEST SWITCH 3 FOR END OF PASS HLT
3326	7650	SNA	CLA	/SKIP IF HALT AT END OF PASS
3327	5650	JMP I	PASCHK	/NO END OF PASS HALT, EXIT
/				
3330	1104	TAD	M04	
3331	3053	DCA	CNT2	/4 DRIVES TO CHECK
3332	1055	TAD	CURDV	/SAVE CURRENT DRIVE TEMPORARILY
3333	3050	DCA	DVTMP	
3334	3055	DCA	CURDV	/SET DRIVE NUMBER TO 0
/				
3335	4440	PASLP,	SETPT	/SET TBLPT TO DRIVE STATE
3336	0000		DSTATE	/TBL INDX
3337	1451	TAD I	TBLPT	/GET A DRV STATE
3340	6211	CDF	10	/CDF TO PROGRAM FLD
3341	7650	SNA	CLA	/SKIP IF DRV ACTIVE
3342	5350	JMP	PCHK	/DRIVE NOT ACTIVE, GO INCREMENT DRV NUMBER
3343	4442	PASPT		/SET UPNT TO PASS TABLE

```

3344 3161      APASO      /TABLE ADDRESS
3345 1447      TAD I      /GET DRIVES END OF PASS FLAG
3346 7650      SNA        /SKIP IF DRIVE HAS COMPLETED A PASS
3347 5650      JMP I      /PASS NOT COMPLETE, EXIT
3350 2055      PCHK, ISZ   /INCREMENT DRIVE NUMBER
3351 2053      ISZ        /INCREMENT DRIVE COUNT
3352 5335      JMP        /CONTINUE CHECKING FOR ALL DRIVES PASSED
3353 1050      TAD        /GET DRIVE NUMBER
3354 3055      DCA        /RESTORE DRIVE NUMBER
3355 4424      MESSAGE
3356 4621      ALLMSG      /"ALL DRVS"
3357 4424      MESSAGE
3360 4612      ACPTDM      /"ACPTD"
3361 4441      C8CAL       /GO TO SR= IN CONSOLE PACKAGE
3362 7000      NOP
3363 5650      JMP I      PASCHK      /EXIT

```

```

/
/*****
/

```

```

/ROUTINE TO GET HARDWARE OR SOFTWARE SWITCHES
/

```

```

/      CALLED BY:      GETSR
/

```

```

3364 0000      SRGET, 0
3365 7300      CLA        CLL
3366 1021      TAD        HDW1      /GET HARDWARE CONFIG WORD 1
3367 7710      SPA        CLA      /SKIP IF USING PSEUDO SWITCH REGISTER
3370 7614      LAS        SKP      /GET HARDWARE SWITCHES
3371 1020      TAD        PSR1      /GET SOFTWARE SWITCHES
3372 5764      JMP I      SRGET      /RETURN

```

```

3374 0400
3375 0002
3376 1750
3377 0025
3400

```

```

PAGE
/
/

```

```

/*****
/

```

```

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

```

```

/BAD SECTORS FOR EACH DRIVE - MAXIMUM OF 16 PER DRIVE
/

```

```

/
/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 A DRIVES BAD SECTOR FILE = 7777 IF NO BAD SECTORS
/FOR THAT DRIVE
/
/
3400 BADSEC=. /START OF BAD SECTOR FILE
/
3400 0000 ZBLOCK 200
/
3577 BADEND=-1 /END OF BAD SECTOR FILE
/
/
/*****
/*****
/***** MESSAGES *****/
/*****
/*****
/*****
/*****
/
3600 2301 SAMSG, TEXT "SA: "
3601 7240
3602 0000
3603 0416 DNMSG, TEXT "DN: "
3604 7240
3605 0000
3606 0522 ERMSG, TEXT "ER: "
3607 7240
3610 0000
3611 0301 CAMSG, TEXT "CA: "
3612 7240
3613 0000
3614 0302 CBMSG, TEXT "CB: "
3615 7240
3616 0000
3617 0306 CFMSG, TEXT "CF: "
3620 7240
3621 0000
3622 1005 HEMSG, TEXT "HE: "
3623 7240
3624 0000
3625 2305 SEMSG, TEXT "SE: "
3626 7240
3627 0000
3630 1127 IWCMSG, TEXT "IWC: "
3631 0372
3632 4000
3633 0627 FWCMSG, TEXT "FWC: "

```

3634	0372		
3635	4000		
3636	2361	S1MSG, TEXT	"S1: "
3637	7240		
3640	0000		
3641	2362	S2MSG, TEXT	"S2: "
3642	7240		
3643	0000		
3644	0405	DEMSG, TEXT	"DE: "
3645	7240		
3646	0000		
3647	2313	SKEMSG, TEXT	"SKE: "
3650	0572		
3651	4000		
3652	2405	TEMSG, TEXT	"TE: "
3653	7240		
3654	0000		
3655	0403	DCRCM, TEXT	"DCRC: "
3656	2203		
3657	7240		
3660	0000		
3661	1003	HCRCM, TEXT	"HCRC: "
3662	2203		
3663	7240		
3664	0000		
3665	0414	DLTMSG, TEXT	"DLT: "
3666	2472		
3667	4000		
3670	1720	OPIMSG, TEXT	"OPI: "
3671	1172		
3672	4000		
3673	1016	HNFMSG, TEXT	"HNF: "
3674	0672		
3675	4000		
3676	0305	CEMSG, TEXT	"CE: "
3677	7240		
3700	0000		
3701	2204	RDSMSG, TEXT	"RDS: "
3702	2372		
3703	4000		
3704	2722	WRTMSG, TEXT	"WRTS: "
3705	2423		
3706	7240		
3707	0000		
3710	2313	SKSMMSG, TEXT	"SKS: "
3711	2372		
3712	4000		
3713	0401	DATAM, TEXT	"DAT: "
3714	2472		
3715	4000		
3716	2003	PCMSG, TEXT	"PC: "
3717	7240		
3720	0000		
3721	4306	FATL, TEXT	"#FATAL"
3722	0124		



3723	0114		
3724	0000		
3725	2001	PATMSG, TEXT	"PATTERN"
3726	2424		
3727	0522		
3730	1600		
3731	1005	HDRMSG, TEXT	"HEADER"
3732	0104		
3733	0522		
3734	0000		
3735	0401	DATMSG, TEXT	"DATA"
3736	2401		
3737	0000		
3740	0201	BAGBM, TEXT	"BA: GOOD BAD"
3741	7240		
3742	4040		
3743	0717		
3744	1704		
3745	4040		
3746	0201		
3747	0400		
3750	0103	ACMSG, TEXT	"AC: "
3751	7240		
3752	0000		
3753	4301	ACCEPT, TEXT	"#ACPT MODE"
3754	0320		
3755	2440		
3756	1517		
3757	0405		
3760	0000		
3761	4324	TDRIVE, TEXT	"#TEST DRIVE "
3762	0523		
3763	2440		
3764	0422		
3765	1126		
3766	0540		
3767	0000		
3770	2205	RETLN, TEXT	"RETRY LIMIT "
3771	2422		
3772	3140		
3773	1411		
3774	1511		
3775	2440		
3776	0000		
3777	2305	SEKMSG, TEXT	"SEEK "
4000	0513		
4001	4000		
4002	4050	YORN, TEXT	" (Y/N)? "
4003	3157		
4004	1651		
4005	7740		
4006	0000		
4007	4050	LM4095, TEXT	" (1-4095)? "
4010	6155		
4011	6460		

4012	7165		
4013	5177		
4014	4000		
4015	4050	LM4096, TEXT	" (3-4096)? "
4016	6355		
4017	6460		
4020	7166		
4021	5177		
4022	4000		
4023	4050	LM1024, TEXT	" (3-1024)? "
4024	6355		
4025	6160		
4026	6264		
4027	5177		
4030	4000		
4031	4322	RAND, TEXT	"#RANDOM "
4032	0116		
4033	0417		
4034	1540		
4035	0000		
4036	2703	WCOUNT, TEXT	"WC "
4037	4000		
4040	4050	DC6061, TEXT	" (60,61) "
4041	6660		
4042	5466		
4043	6151		
4044	4000		
4045	4304	DATDUM, TEXT	"#DATA DUMP ON DCRC"
4046	0124		
4047	0140		
4050	0425		
4051	1520		
4052	4017		
4053	1640		
4054	0403		
4055	2203		
4056	0000		
4057	4005	ERR, TEXT	" ERROR"
4060	2222		
4061	1722		
4062	0000		
4063	4014	LIMMSG, TEXT	" LIMIT"
4064	1115		
4065	1124		
4066	0000		
4067	4323	SOFT, TEXT	"#SOFT"
4070	1706		
4071	2400		
4072	4310	HARD, TEXT	"#HARD"
4073	0122		
4074	0400		
4075	0422	DRVMSG, TEXT	"DRIVE "
4076	1126		
4077	0540		
4100	0000		

4101	3006	DXFER, TEXT	"XFER"
4102	0522		
4103	0000		
4104	4323	SEKOP, TEXT	"#SEEK OPR"
4105	0505		
4106	1340		
4107	1720		
4110	2200		
4111	4315	CHNGSK, TEXT	"#MODIFY SEEK, R/W PARAMS"
4112	1704		
4113	1106		
4114	3140		
4115	2305		
4116	0513		
4117	5440		
4120	2257		
4121	2740		
4122	2001		
4123	2201		
4124	1523		
4125	0000		
4126	4315	MAXMSG, TEXT	"#MAX "
4127	0130		
4130	4000		
4131	2722	WRDXFR, TEXT	"WRD "
4132	0440		
4133	0000		
4134	4303	CONSD, TEXT	"#CONSTANT DATA"
4135	1716		
4136	2324		
4137	0116		
4140	2440		
4141	0401		
4142	2401		
4143	0000		
4144	4320	DPATN, TEXT	"#PATN (0-3)? "
4145	0124		
4146	1640		
4147	5060		
4150	5563		
4151	5177		
4152	4000		
4153	4304	NUMWRD, TEXT	"#DATA ERR REPT (0-254)?"
4154	0124		
4155	0140		
4156	0522		
4157	2240		
4160	2205		
4161	2024		
4162	4050		
4163	6055		
4164	6265		
4165	6451		
4166	7700		
4167	4323	SNLSUR, TEXT	"#SNGL SURFACE "

4170	1607		
4171	1440		
4172	2325		
4173	2206		
4174	0103		
4175	0540		
4176	0000		
4177	4315	MINMSG, TEXT	"#MIN "
4200	1116		
4201	4000		
4202	0331	CYLMMSG, TEXT	"CYL "
4203	1440		
4204	0000		
4205	2305	SECMSG, TEXT	"SECTOR "
4206	0324		
4207	1722		
4210	4000		
4211	4050	LM511, TEXT	" (0-511)? " /
4212	6055		
4213	6561		
4214	6151		
4215	7740		
4216	0000		
4217	4050	LM39, TEXT	" (0-39)? "
4220	6055		
4221	6371		
4222	5177		
4223	4000		
4224	4343	RUSURE, TEXT	"##ARE YOU SURE? "
4225	0122		
4226	0540		
4227	3117		
4230	2540		
4231	2325		
4232	2205		
4233	7740		
4234	0000		
4235	4314	LIMEXD, TEXT	"#LIMIT EXCEEDED! "
4236	1115		
4237	1124		
4240	4005		
4241	3003		
4242	0505		
4243	0405		
4244	0441		
4245	4000		
4246	4010	HIMSG, TEXT	" HIGH: "
4247	1107		
4250	1072		
4251	4000		
4252	4014	LOMSG, TEXT	" LO: "
4253	1772		
4254	4000		
4255	4303	CNTRD1, TEXT	"#CAN'T RD FACT"
4256	0116		

HP 001

4257	4724		
4260	4022		
4261	0440		
4262	0601		
4263	0324		
4264	0000		
4265	4002	BDSEC, TEXT	" BAD SECT FILE "
4266	0104		
4267	4023		
4270	0503		
4271	2440		
4272	0611		
4273	1405		
4274	4000		
4275	4303	CNTRD2, TEXT	"#CAN'T RD FLD"
4276	0116		
4277	4724		
4300	4022		
4301	0440		
4302	0614		
4303	0400		
4304	4076	PAKBAD, TEXT	" > 16 BAD SECTS#"
4305	4061		
4306	6640		
4307	0201		
4310	0440		
4311	2305		
4312	0324		
4313	2343		
4314	0000		
4315	4316	NODRV8, TEXT	"#NO DRVS TO TST!#"
4316	1740		
4317	0422		
4320	2623		
4321	4024		
4322	1740		
4323	2423		
4324	2441		
4325	4300		
4326	4001	DRVACT, TEXT	" ACTV#"
4327	0324		
4330	2643		
4331	0000		
4332	1617	NOTMSG, TEXT	"NOT"
4333	2400		
4334	4050	LM7, TEXT	" (0-7)? "
4335	6055		
4336	6751		
4337	7740		
4340	0000		
4341	4304	DROP, TEXT	"#DROP "
4342	2217		
4343	2040		
4344	0000		
4345	4304	DROPNG, TEXT	"#DROPPING DRIVE#"

4346	2217		
4347	2020		
4350	1116		
4351	0740		
4352	0422		
4353	1126		
4354	0543		
4355	0000		
4356	4301	ADD, TEXT	"#ADD "
4357	0404		
4360	4000		
4361	4301	TITLE, TEXT	"#AJRLK-B RL8A PERFORMANCE EXERCISER FOR THE RL02#"
4362	1222		
4363	1413		
4364	5502		
4365	4040		
4366	2214		
4367	7001		
4370	4020		
4371	0522		
4372	0617		
4373	2215		
4374	0116		
4375	0305		
4376	4005		
4377	3005		
4400	2203		
4401	1123		
4402	0522		
4403	4006		
4404	1722		
4405	4024		
4406	1005		
4407	4022		
4410	1460		
4411	6243		
4412	0000		
4413	4022	REACHD, TEXT	" REACHED"
4414	0501		
4415	0310		
4416	0504		
4417	0000		
4420	4004	DRVON, TEXT	" DRV ON"
4421	2226		
4422	4017		
4423	1600		
4424	4017	OPRAT, TEXT	" OPR"
4425	2022		
4426	0000		
4427	4322	RDONL, TEXT	"#RD ONLY"
4430	0440		
4431	1716		
4432	1431		
4433	0000		
4434	4340	WCHONE, TEXT	"# SURF (0 OR 1)? "

4435 2325  
4436 2206  
4437 4050  
4440 6040  
4441 1722  
4442 4061  
4443 5177  
4444 4000  
4445 4316  
4446 2515  
4447 4006  
4450 1404  
4451 2340  
4452 0530  
4453 2404  
4454 4022  
4455 5727  
4456 4015  
4457 0515  
4460 4050  
4461 6155  
4462 6751  
4463 7740  
4464 0000  
4465 4316  
4466 1740  
4467 0614  
4470 0440  
4471 0000  
4472 4322  
4473 0523  
4474 2401  
4475 2224  
4476 0000  
4477 4304  
4500 0340  
4501 0000  
4502 4320  
4503 2722  
4504 4006  
4505 0111  
4506 1440  
4507 4020  
4510 0372  
4511 4040  
4512 0000  
4513 4303  
4514 1422  
4515 4004  
4516 2226  
4517 4023  
4520 2401  
4521 2440  
4522 2402  
4523 1423

FLDMSG, TEXT    "#NUM FLDS EXTD R/W MEM (1-7)? "

NOFLD, TEXT    "#NO FLD "

RSTRT, TEXT    "#RESTART"

DCODE, TEXT    "#DC "

PWRFAL, TEXT    "#PWR FAIL PC: "

CLRTAB, TEXT    "#CLR DRV STAT TBLs"

4524	0000		
4525	4316	NOINT, TEXT	"#NO INT REQ"
4526	1740		
4527	1116		
4530	2440		
4531	2205		
4532	2100		
4533	4307	RETBAD, TEXT	"#GET BAD SECT FILE"
4534	0524		
4535	4002		
4536	0104		
4537	4023		
4540	0503		
4541	2440		
4542	0611		
4543	1405		
4544	0000		
4545	4327	WPAK, TEXT	"#WRT PAK"
4546	2224		
4547	4020		
4550	0113		
4551	0000		
4552	4020	PAKWRT, TEXT	" PACK WRITTEN#"
4553	0103		
4554	1340		
4555	2722		
4556	1124		
4557	2405		
4560	1643		
4561	0000		
4562	4020	PAKRD, TEXT	" PACK READ OK#"
4563	0103		
4564	1340		
4565	2205		
4566	0104		
4567	4017		
4570	1343		
4571	0000		
4572	2516	UNINT, TEXT	"UNK INT"
4573	1340		
4574	1116		
4575	2400		
4576	4027	WRTPH, TEXT	" WRT PROT#"
4577	2224		
4600	4020		
4601	2217		
4602	2443		
4603	0000		
4604	4305	PASMSG, TEXT	"#END PASS "
4605	1604		
4606	4020		
4607	0123		
4610	2340		
4611	0000		
4612	4001	ACPTDM, TEXT	" ACPTD#"



**DATA, TEXT "DATA"**

ALLMSG, TEXT " ALL DRVS#"

ERRPCM, TEXT "ERROR PC: "

\*\*\*\*\*

DATA PATTERN TABLES

## DATA PATTERN TABLES

4636	0052	PAT80,	0052	/PATTERN 0 (8BIT MODE)
4637	0252		0252	
4640	0125		0125	
4641	0125		0125	
4642	0125		0125	
4643	0125		0125	
4644	0252		0252	
4645	0252		0252	
4646	0252		0252	
4647	0252		0252	
4650	0252		0252	
4651	0252		0252	
4652	0125		0125	
4653	0125		0125	
4654	0125		0125	
4655	0125		0125	
4656	0252		0252	
4657	0252		0252	
4660	0252		0252	
4661	0252		0252	
4662	0125		0125	
4663	0125		0125	
4664	0252		0252	
4665	0252		0252	
4666	0125		0125	
4667	0125		0125	
4670	0252		0252	

4671	0252	0252
4672	0125	0125
4673	0125	0125
4674	0252	0252
4675	0252	0252

/			
4676	0333	PAT81,	0333
4677	0155		0155
4700	0266		0266
4701	0333		0333
4702	0155		0155
4703	0266		0266
4704	0333		0333
4705	0155		0155
4706	0266		0266
4707	0333		0333
4710	0155		0155
4711	0266		0266
4712	0333		0333
4713	0155		0155
4714	0266		0266
4715	0333		0333
4716	0155		0155
4717	0266		0266
4720	0333		0333
4721	0155		0155
4722	0266		0266
4723	0333		0333
4724	0155		0155
4725	0266		0266
4726	0333		0333
4727	0155		0155
4730	0266		0266
4731	0333		0333
4732	0155		0155
4733	0266		0266
4734	0333		0333
4735	0155		0155

/			
4736	0242	PAT82,	0242
4737	0105		0105
4740	0321		0321
4741	0042		0042
4742	0150		0150
4743	0221		0221
4744	0264		0264
4745	0110		0110
4746	0132		0132
4747	0044		0044
4750	0055		0055
4751	0022		0022
4752	0026		0026
4753	0211		0211
4754	0213		0213
4755	0104		0104

4756	0105	0105
4757	0242	0242
4760	0042	0042
4761	0321	0321
4762	0221	0221
4763	0150	0150
4764	0110	0110
4765	0264	0264
4766	0044	0044
4767	0132	0132
4770	0022	0022
4771	0055	0055
4772	0211	0211
4773	0026	0026
4774	0104	0104
4775	0213	0213

4776	0113	/	PAT03,	0113	/PATTERN 3 (8 BIT MODE)
4777	0113			0113	
5000	0245			0245	
5001	0245			0245	
5002	0322			0322	
5003	0322			0322	
5004	0151			0151	
5005	0151			0151	
5006	0264			0264	
5007	0264			0264	
5010	0132			0132	
5011	0132			0132	
5012	0055			0055	
5013	0055			0055	
5014	0226			0226	
5015	0226			0226	
5016	0113			0113	
5017	0113			0113	
5020	0245			0245	
5021	0245			0245	
5022	0322			0322	
5023	0322			0322	
5024	0151			0151	
5025	0151			0151	
5026	0264			0264	
5027	0264			0264	
5030	0132			0132	
5031	0132			0132	
5032	0055			0055	
5033	0055			0055	
5034	0226			0226	
5035	0226			0226	

5036	1252	/	PAT120,	1252	/PATTERN 0 (12 BIT MODE)
5037	5125			5125	
5040	2525			2525	
5041	2525			2525	
5042	5252			5252	

5043	5252	5252
5044	5252	5252
5045	5252	5252
5046	2525	2525
5047	2525	2525
5050	2532	2532
5051	5252	5252
5052	5252	5252
5053	5125	5125
5054	2532	2532
5055	5252	5252
5056	2525	2525
5057	2652	2652
5060	5245	5245
5061	2525	2525
5062	5252	5252
5063	5252	5252

5064	6666	PAT121, 6666	/PATTERN 1 (12 BIT MODE)
5065	6666	6666	
5066	6666	6666	
5067	6666	6666	
5070	6666	6666	
5071	6666	6666	
5072	6666	6666	
5073	6666	6666	
5074	6666	6666	
5075	6666	6666	
5076	6666	6666	
5077	6666	6666	
5100	6666	6666	
5101	6666	6666	
5102	6666	6666	
5103	6666	6666	
5104	6666	6666	
5105	6666	6666	
5106	6666	6666	
5107	6666	6666	
5110	6666	6666	
5111	6666	6666	

5112	5044	PAT122, 5044	/PATTERN 2 (12 BIT MODE)
5113	2721	2721	
5114	1046	1046	
5115	4221	4221	
5116	5504	5504	
5117	4132	4132	
5120	1102	1102	
5121	6422	6422	
5122	0550	0550	
5123	4613	4613	
5124	2104	2104	
5125	2642	2642	
5126	1055	1055	
5127	0621	0621	

5130	3204	3204
5131	4264	4264
5132	1105	1105
5133	5022	5022
5134	1330	1330
5135	4426	4426
5136	2110	2110
5137	5422	5422

5140	2264	PAT123, 2264	/PATTERN 3 (12 BIT MODE)
5141	5645	5645	
5142	5135	5135	
5143	1322	1322	
5144	3226	3226	
5145	4664	4664	
5146	5505	5505	
5147	5132	5132	
5150	1322	1322	
5151	6626	6626	
5152	4544	4544	
5153	5513	5513	
5154	5132	5132	
5155	2722	2722	
5156	6446	6446	
5157	4551	4551	
5160	5513	5513	
5161	2132	2132	
5162	2642	2642	
5163	6455	6455	
5164	4551	4551	
5165	3004	3004	

```

/*****
/*****
/*****
/*****  C O N S O L E  P A C K A G E  *****/
/*****
/*****
/*****
/
/IF ENTERED WITH AC=0000 THE SWITCH REGISTER
/MODIFICATION ROUTINE IS ENTERED AUTOMATICALLY.
/IF ENTERED WITH AC NOT EQUAL TO 0000, THE
/KEYBOARD INPUT DECODER IS ENTERED AND IT IS ASSUMED
/THAT THE AC CONTAINS THE ASCII CODE TO BE
/CHECKED FOR A VALID CONTROL CHARACTER.
/

```

```

/ 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:
/
/ OPERATOR TYPES:          RESULT:
/ -----          -----
/
/ 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    CHANGE PSEUDO SWITCH REGISTER AND
/ FOLLOWED BY CARRIAGE    RETURN TO THE PROGRAM, CPU FLAGS
/ RETURN                  AND STATUS RESTORED
/
/ 1 TO 4 OCTAL DIGITS    CHANGE PSEUDO SWITCH REGISTER AND
/ FOLLOWED BY LINE FEED   RESTART THE PROGRAM
/
/ 1 TO 4 OCTAL DIGITS    NO CHANGE TO PSEUDO SWITCH REGISTER
/ FOLLOWED BY CONTROL G   PRINT UPARROW G, EXECUTE A CARRIAGE
/                          RETURN AND LINE FEED AND PRINT
/                          SR=XXXX, WAIT FOR OPERATOR INPUT
/
/ ALL OTHER INPUT        NO CHANGE TO PSEUDO SWITCH REGISTER
/ (ILLEGAL CHARACTERS OR ECHO THE CHARACTER, PRINT A QUESTION
/ 5 OCTAL DIGITS)        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 SHOULD  
 /CONTINUE TO RUN.  
 /SUCH TIME AS A MESSAGE IS PENDING. WHEN A MESSAGE IS PENDING, THE  
 /PROGRAM SHOULD WAIT FOR A CONTROL Q OR A CONTROL C. IF THE  
 /OPERATOR TYPES CONTROL S WHILE A MESSAGE IS IN PROGRESS, THE  
 /PRINTOUT SHOULD BE INTERRUPTED AND THE PROGRAM SHOULD WAIT FOR A  
 /CONTROL Q OR CONTROL C.

/  
 / 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.

5200 PAGE

5200	0000	C8TEMP, 0	/TEMPORARY WORK AREA
5201	6203	C8CDI, C1F CDF	/USED TO CREATE CDI TO PROGRAM FIELD
5202	6201	C8CDF, CDF	/USED TO CREATE CDF TO CONSOLE FIELD
5203	0000	C8SWR, 0	/SWITCH REGISTER SAVE AREA
5204	0000	C8MODE, 0	/PRINT MODE SWITCH
5205	0000	C8CNTR, 0	/USED AS COUNTER
5206	7775	C8M3, -3	/CONSTANT
5207	7774	C8M4, -4	/CONSTANT
5210	7773	C8M5, -5	/CONSTANT
5211	7770	C8M10, -10	/CONSTANT
5212	7520	C8M260, -260	/CONSTANT
5213	0007	C8K7, 0007	/CONSTANT
5214	0240	C8K240, 0240	/CONSTANT
5215	0260	C8K260, 0260	/CONSTANT
5216	0275	C8K275, 0275	/CONSTANT
5217	0277	C8K277, 0277	/CONSTANT
5220	0322	C8K322, 0322	/CONSTANT
5221	0323	C8K323, 0323	/CONSTANT

5222	0000	C8ENTR, 0	
5223	3200	DCA	C8TEMP /SAVE AC

5224	6214	RDF		/READ PROGRAM FIELD
5225	1201	TAD	C8CDI	/ADD CDI INSTRUCTION
5226	3205	DCA	C8CNTR	/SAVE CDI TO PROGRAM FIELD TEMPORARILY
5227	6224	RIF		/READ CONSOLE FIELD
5230	1202	TAD	C8CDF	/ADD CDF INSTRUCTION
5231	3241	DCA	C8FLD	/SAVE CDF TO CONSOLE FIELD
5232	1777	TAD I	(21	/GET HCW1 FROM PROGRAM FIELD
5233	7710	SPA CLA		/SKIP IF USING PSEUDO SWR
5234	7614	LAS	SKP	/GET HARDWARE SWR AND SKIP
5235	1776	TAD I	(20	/GET PSEUDO SWR
5236	3203	DCA	C8SWR	/SAVE SWITCH REGISTER
5237	1046	TAD	INMODE	/GET MESSAGE ACTIVE FLAG
5240	3204	DCA	C8MODE	/SAVE MESSAGE ACTIVE FLAG
5241	7402	C8FLD, HLT		/MODIFIED CDF TO CONSOLE DATA FIELD
5242	1222	TAD	C8ENTR	/GET RETURN ADDRESS
5243	3775'	DCA	C8RTN	/SAVE FOR EXIT
5244	1205	TAD	C8CNTR	/GET CDI TO PROGRAM FIELD
5245	3774'	DCA	C8PFLD	/SAVE CDI TO PROGRAM FIELD FOR EXIT
5246	1200	TAD	C8TEMP	/GET AC UPON ENTRY
5247	7440	SZA		/SKIP IF IT WAS ZERO
5250	5773'	JMP	C8CNTL	/AC NOT ZERO, GO CHECK CTRL CHAR

HP 017

/

/PRINT OUT SR=XXXX WHERE XXXX IS THE CURRENT CONTENTS

/OF THE SWITCH REGISTER BEING USED (EITHER PSEUDO OR HARDWARE)

/

5251	4772'	C8PSW, JMS	C8CRLF	/DO A <CR> AND <LF>
5252	1221	TAD	C8K323	/GET ASCII CODE FOR "S"
5253	4771'	JMS	C8TYP	/PRINT "S"
5254	1220	TAD	C8K322	/GET ASCII CODE FOR "R"
5255	4771'	JMS	C8TYP	/PRINT "R"
5256	1216	TAD	C8K275	/GET ASCII CODE FOR "="
5257	4771'	JMS	C8TYP	/PRINT "="
5260	1207	TAD	C8M4	/AC=-4
5261	3205	DCA	C8CNTR	/SET UP OCTAL DIGIT COUNTER
5262	1203	TAD	C8SWR	/GET SWITCH REGISTER
5263	7004	RAL		/EXTRA ROTATE FOR LINK
5264	7004	C8LOPA, RAL		
5265	7006	RTL		/ROTATE OCTAL DIGITS FOR PRINTING
5266	3203	DCA	C8SWR	/SAVE ROTATED SWR
5267	1203	TAD	C8SWR	/GET ROTATED SWR
5270	0213	AND	C8K7	/MASK OFF DIGIT TO PRINT
5271	1215	TAD	C8K260	/ADD ASCII BASE CODE
5272	4771'	JMS	C8TYP	/PRINT AN OCTAL DIGIT
5273	1203	TAD	C8SWR	/GET SWR
5274	2205	ISZ	C8CNTR	/INCREMENT LOOP COUNTER
5275	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

/

5276 7300

CLA CLL



5277	1210	TAD	C8M5	/AC=-5
5300	3205	DCA	C8CNTR	/SET UP TO ACCEPT 5 CHARACTERS
5301	3770	DCA	C8BLD	/CLEAR SWITCH REG. BUILD AREA
5302	3767	DCA	C8FLG	/CLEAR SWR CHANGE SWITCH
5303	1214	TAD	C8K240	/GET ASCII CODE FOR SPACE
5304	4771	JMS	C8TYP	/SPACE OVER ONE POSITION
5305	4766	C8SRLP, JMS	C8TTY	/GO WAIT FOR KEYBOARD INPUT
5306	3200	DCA	C8TEMP	/SAVE INPUT CHARACTER
5307	1200	TAD	C8TEMP	/GET CHARACTER
5310	1365	TAD	(-203	
5311	7450	SNA		/SKIP IF NOT CTRL/C
5312	5764	JMP	C8CTLC	/GO TO CTRL/C ROUTINE
5313	1207	TAD	C8M4	/AC=-4
5314	7450	SNA		/SKIP IF NOT CTRL/G
5315	5763	JMP	C8CTLG	/GO TO CTRL/G ROUTINE
5316	1206	TAD	C8M3	/SUBTRACT 3
5317	7450	SNA		/SKIP IF NOT LINE FEED
5320	5762	JMP	C8EXT1	/GO TO LINE FEED EXIT
5321	1206	TAD	C8M3	/SUBTRACT 3
5322	7650	SNA CLA		/SKIP IF NOT CARRIAGE RETURN
5323	5761	JMP	C8EXT2	/GO TO CARRIAGE RETURN EXIT
5324	1200	TAD	C8TEMP	/GET CHARACTER
5325	4771	JMS	C8TYP	/ECHO IT
5326	1200	TAD	C8TEMP	/GET CHARACTER
5327	1212	TAD	C8M260	
5330	7510	SPA		/SKIP IF >= TO ASCII CODE FOR ZERO
5331	5351	JMP	C8ERR	/INVALID CHARACTER NOT OCTAL DIGIT
5332	1211	TAD	C8M10	
5333	7700	SNA CLA		/SKIP IF <= ASCII CODE FOR SEVEN
5334	5351	JMP	C8ERR	/INVALID CHARACTER NOT OCTAL DIGIT
5335	7240	STA		/AC=7777
5336	3767	DCA	C8FLG	/SET SWR CHANGE FLAG
5337	1200	TAD	C8TEMP	/GET CHARACTER
5340	0213	AND	C8K7	/MASK TO 3 BITS
5341	3200	DCA	C8TEMP	/SAVE OCTAL DIGIT
5342	1770	TAD	C8BLD	/GET SWR BUILD AREA CONTENTS
5343	7106	CLL RTL		
5344	7004	RAL		/ROTATE TO BUILD SWR
5345	1200	TAD	C8TEMP	/ADD NEXT OCTAL DIGIT
5346	3770	DCA	C8BLD	/SAVE NEW SWR
5347	2205	ISZ	C8CNTR	/INCREMENT OCTAL DIGIT COUNTER
5350	5305	JMP	C8SRLP	/CONTINUE ACCEPTING OCTAL DIGITS
/				
5351	7300	C8ERR, CLA CLL		
5352	1217	TAD	C8K277	/GET ASCII CODE FOR "?"
5353	4771	JMS	C8TYP	/PRINT "?"
5354	4772	JMS	C8CRLF	/DO A <CR> AND <LF>
5355	5251	JMP	C8PSW	/GO START OVER
/				
5361	5551			
5362	5534			
5363	5457			
5364	5465			
5365	7575			

5366 5514  
 5367 5403  
 5370 5400  
 5371 5477  
 5372 5526  
 5373 5420  
 5374 5552  
 5375 5402  
 5376 0020  
 5377 0021  
 5400

## PAGE

5400	0000	C8BLD, 0	/SWITCH REGISTER BUILD AREA
5401	0200	C8STRT, RSTART	/ADDRESS OF START OF PROGRAM
5402	0000	C8RTN, 0	/STORAGE FOR RETURN ADDRESS
5403	0000	C8FLG, 0	/SWR CHANGE SWITCH
5404	0000	C8SFLG, 0	/CTRL/S ACTIVE FLAG
5405	0177	C8K177, 0177	/CONSTANT
5406	0200	C8K200, 0200	/CONSTANT
5407	0077	C8K77, 0077	/CONSTANT
5410	7740	C8M40, -40	/CONSTANT
5411	0100	C8K100, 0100	/CONSTANT
5412	0215	C8K215, 0215	/CONSTANT
5413	0212	C8K212, 0212	/CONSTANT
5414	0303	C8K303, 0303	/CONSTANT
5415	0307	C8K307, 0307	/CONSTANT
5416	0336	C8K336, 0336	/CONSTANT
5417	7600	C87600, 7600	

\*\*\*\*\*

## CONTROL CHARACTER

## DECODE ROUTINE

5420	1377	C8CNTL, TAD	(-203	
5421	7450	SNA		/SKIP IF NOT CTRL/C
5422	5265	JMP	C8CTLC	/CTRL/C TYPED EXIT TO MONITOR
5423	1376	TAD	(-4	
5424	7450	SNA		/SKIP IF NOT CTRL/G
5425	5257	JMP	C8CTLG	/CTRL/G TYPED GO PRINT "G"
5426	1375	TAD	(-12	
5427	7450	SNA		/SKIP IF NOT CTRL/Q
5430	5255	JMP	C8CTLQ	/CTRL/Q TYPED
5431	1374	TAD	(-2	
5432	7450	SNA		/SKIP IF NOT CTRL/S
5433	5237	JMP	C8CTLs	/CTRL/S TYPED
5434	3773	DCA	C8MODE	/SET MESSAGE ACTIVE FLAG
5435	2204	ISZ	C8SFLG	/TEST CTRL/S ACTIVE FLAG
5436	5274	JMP	C8ECHO	/ECHO CHAR AND RETURN TO PROGRAM

/CTRL/S HANDLER

5437	7240	C8CTL8, STA		/AC=7777
5440	3204	DCA	C8SFLG	/SET CTRL/S ACTIVE FLAG
5441	1773	TAD	C8MODE	/GET MESSAGE ACTIVE FLAG
5442	7650	SNA CLA		/SKIP IF CTRL/S TYPED WHILE MESSAGE ACTIVE
5443	5352	JMP	C8PFLD	/RETURN TO PROGRAM
/				
5444	7240	C8WAIT, STA		/AC=7777
5445	3204	DCA	C8SFLG	/SET CTRL/S ACTIVE FLAG
5446	4314	JMS	C8TTY	/WAIT FOR KEYBOARD INPUT
5447	1377	TAD	(-203	
5450	7450	SNA		/SKIP IF NOT CTRL/C
5451	5265	JMP	C8CTLG	/CTRL/C TYPED EXIT TO MONITOR
5452	1372	TAD	(-16	
5453	7640	SZA CLA		/SKIP IF CTRL/Q
5454	5244	JMP	C8WAIT	/NOT CTRL/C OR CTRL/Q CONTINUE WAITING
5455	3204	C8CTLQ, DCA	C8SFLG	/CLEAR CTRL/S ACTIVE FLAG
5456	5352	JMP	C8PFLD	/RETURN TO MAIN PROGRAM
/				
/CONTROL G HANDLER				
/				
5457	4326	C8CTLG, JMS	C8CRLF	/DO A <CR> AND <LF>
5460	1216	TAD	C8K336	/GET ASCII CODE FOR UP ARROW
5461	4277	JMS	C8TYP	/PRINT UP ARROW
5462	1215	TAD	C8K307	/GET ASCII CODE FOR "G"
5463	4277	JMS	C8TYP	/PRINT "G"
5464	5771	JMP	C8PSW	/GO TO "SR=XXXX" ROUTINE
/				
/CONTROL C HANDLER				
/				
5465	3204	C8CTLG, DCA	C8SFLG	/CLEAR CTRL/S ACTIVE FLAG
5466	1216	TAD	C8K336	/GET ASCII CODE FOR UP ARROW
5467	4277	JMS	C8TYP	/PRINT UP ARROW
5470	1214	TAD	C8K303	/GET ASCII CODE FOR "C"
5471	4277	JMS	C8TYP	/PRINT "C"
5472	6203	CIF CDF		/CHANGE TO IF AND DF ZERO
5473	5617	JMP I	C87600	/RETURN TO MONITOR
/				
5474	1770	C8ECHO, TAD	C8TEMP	/GET CHARACTER
5475	4277	JMS	C8TYP	/ECHO IT
5476	5352	JMP	C8PFLD	/RETURN TO PROGRAM
/				
/PRINT ONE CHARACTER				
/				
5477	0000	C8TYP, 0		
5500	2204	ISZ	C8SFLG	/TEST CTRL/S ACTIVE FLAG
5501	7410	SKP		/SKIP IF CTRL/S NOT ACTIVE
5502	5244	JMP	C8WAIT	/GO WAIT FOR CTRL/Q OR CTRL/C
5503	6046	TLS		/TRANSMIT CHARACTER
5504	6102	SPL		/SKIP ON POWER LOW
5505	7410	SKP		/SKIP IF POWER OK
5506	6001	ION		/TURN INTERRUPT ON - POWER IS LOW
5507	6041	TSF		/TEST TTY FLAG
5510	5304	JMP	.-4	/WAIT FOR TTY FLAG
5511	6042	TCF		/CLEAR TTY FLAG

```

5512 7200          CLA                /CLEAR AC DO NOT CLEAR LINK
5513 5677          JMP I  C8TYP        /RETURN
/
/WAIT FOR KEYBOARD INPUT THEN EXIT WITH ASCII CODE IN AC
/
5514 0000          C8TTY, 0
5515 6102          SPL                /SKIP ON POWER LOW
5516 7410          SKP                /SKIP IF POWER OK
5517 6001          ION                /TURN INTERRUPT ON - POWER IS LOW
5520 6031          KSF                /SKIP IF KEYBOARD FLAG SET
5521 5315          JMP      .-4        /WAIT FOR KEYBOARD INPUT
5522 6036          KRB                /READ KEYBOARD BUFFER  CLEAR FLAG
5523 0205          AND      C8K177     /MASK TO 7 BITS
5524 1206          TAD      C8K200     /SET BIT 4
5525 5714          JMP I  C8TTY        /RETURN
/
/EXECUTE A CARRIAGE RETURN AND LINE FEED
/
5526 0000          C8CRLF, 0
5527 1212          TAD      C8K215     /GET ASCII CODE FOR CARRIAGE RETURN
5530 4277          JMS      C8TYP        /GO EXECUTE THE CARRIAGE RETURN
5531 1213          TAD      C8K212     /GET ASCII CODE FOR LINE FEED
5532 4277          JMS      C8TYP        /GO EXECUTE THE LINE FEED
5533 5726          JMP I  C8CRLF        /RETURN
/
/CONSOLE PACKAGE EXIT IF TERMINATED WITH LINE FEED
/
5534 4326          C8EXT1, JMS      C8CRLF        /DO A <CR> AND <LF>
5535 1352          TAD      C8PFLD     /GET MODIFIED CDI TO PROGRAM FIELD
5536 3337          DCA      .+1        /SAVE FOR EXECUTION
5537 7402          HLT/CDI            /MODIFIED CDI TO PROGRAM FIELD
5540 2203          ISZ      C8FLG      /TEST SWR CHANGE FLAG
5541 5601          JMP I  C8STRT       /RESTART PROGRAM WITHOUT CHANGE OF SWR
5542 6201          CDF      00         /CDF TO FIELD 0
5543 1200          TAD      C8BLD      /GET NEW SWITCH REGISTER
5544 3767          DCA I  (20         /SAVE IT IN FIELD 0 PAGE 0 LOC 20
5545 6211          CDF      10         /CDF TO FIELD 1
5546 1200          TAD      C8BLD      /GET NEW SWITCH REGISTER
5547 3767          DCA I  (20         /SAVE IT IN FIELD 1 PAGE 0 LOC 20
5550 5337          JMP      C8EXT1+3   /GO EXECUTE CDI INSTRUCTION AGAIN
/
/EXIT FROM CONSOLE PACKAGE IF TERMINATED WITH CARRIAGE RETURN
/
5551 4326          C8EXT2, JMS      C8CRLF        /DO A <CR> AND <LF>
5552 7402          C8PFLD, HLT        /MODIFIED CDI TO PROGRAM FIELD
5553 7300          CLA CLL            /CLEAR AC AND LINK FOR RETURN
5554 2203          ISZ      C8FLG      /TEST SWR CHANGE FLAG
5555 5602          JMP I  C8RTN       /RETURN TO PROGRAM WITHOUT CHANGE OF SWR
5556 1200          TAD      C8BLD      /GET NEW SWITCH REGISTER
5557 3767          DCA I  (20         /SAVE IT IN PROGRAM FIELD
5560 5602          JMP I  C8RTN       /RETURN TO PROGRAM
/
/*****
5567 0020

```

) ) ) ) )  
5570 5200  
5571 5251  
5572 7762  
5573 5204  
5574 7776  
5575 7766  
5576 7774  
5577 7575

SEQ 193

```
5600 *BUF1 /DATA BUFFER FIELD 1
/
5600 0000 ZBLOCK BUFSZ1
/
7577 BUFEN1=-1
/
/*****
/
0000 FIELD 0
```

[illegible]

7600  
7700



**SEQ 197**

**0200**

**\*200**

**END OF PROGRAM**

\*\*\*

ABORT	5506	AUTO14	0014	BRKLOD	5115	C8K100	5411
ACCEPT	3753	AUTO15	0015	BSFEX	3552	C8K177	5405
ACCNTR	1526	AUTO16	0016	BSV	2724	C8K200	5406
ACHK1	0222	AUTO17	0017	BSW	7002	C8K212	5413
ACL	7701	AUTSIZ	0272	BSYCHK	0400	C8K215	5412
ACLP	1510	BADCHK	2076	BTSV	2727	C8K240	5214
ACMSG	3750	BADCNT	3752	BUFO	7200	C8K260	5215
ACOUT	1523	BADEND	3577	BUF1	5600	C8K275	5216
ACPRN	2164	BADEX1	2245	BUF2	0000	C8K277	5217
ACPRNT	1476	BADEX2	4555	BUFA0	6702	C8K303	5414
ACPTDM	4612	BADLPB	3664	BUFAD0	0116	C8K307	5415
ACSAV	2340	BADPNT	0131	BUFAD1	0117	C8K322	5220
ACSAVE	4017	BADPRO	3647	BUFCDF	6520	C8K323	5221
ACSV	4345	BADPT	2150	BUFCHK	1065	C8K336	5416
ACTMP	1525	BADRES	3276	BUFFT	2507	C8K7	5213
ADD	4356	BADSEC	3400	BUFCTR	6553	C8K77	5407
ADDEX	3325	BADSWT	3750	BUFENO	7577	C8LOC	6312
ADDEX1	3330	BADTRK	3751	BUFEN1	7577	C8LOPA	5264
ADDRS1	0001	BAGBM	3740	BUFLD	1727	C8M10	5211
ADDRS2	0002	BCNT	3331	BUFLD0	0066	C8M260	5212
ADDRS3	0003	BDCNT	0133	BUFLD1	0067	C8M3	5206
ADDRS4	0004	BDSEC	4265	BUFLD2	0070	C8M4	5207
ADDRS5	0005	BDSEC0	7124	BUFLD3	0071	C8M40	5410
ADDRS6	0006	BDSEC1	7125	BUFLPB	1036	C8M5	5210
ADDRS7	0007	BDSEC2	7126	BUFNUM	0125	C8MODE	5204
ADDRSA	2431	BDSEC3	7127	BUFP	1730	C8OFF	6261
ADDRV	4574	BDTRK	0616	BUFP1	2262	C8OUT	2363
ADJUST	1154	BFLD0	0120	BUFPNT	3747	C8PFLD	5552
ALLMSG	4621	BFLD1	0121	BUFSZ0	0400	C8PSW	5251
ALLPAS	3324	BFLD2	0122	BUFSZ1	2000	C8RTN	5402
ALODE	5132	BFLD3	0123	BUFSZ2	7600	C8SFLG	5404
ANYDRV	0310	BFPT	6535	BYSTAT	6246	C8SRLP	5305
ANYLP	0313	BHOME	4261	C87600	5417	C8STRT	5401
APAS0	3161	BITCNT	3136	C8BLD	5400	C8SWR	5203
APAS1	3162	BITEND	3127	C8CAL	4441	C8TEMP	5200
APAS2	3163	BITGET	1117	C8CALL	4546	C8TTY	5514
APAS3	3164	BITLP	3044	C8CDF	5202	C8TYP	5477
APTCHK	4545	BITH12	4722	C8CDI	5201	C8WAIT	5444
APTCK	4437	BITH8	4715	C8CNTL	5420	CAF	6007
APTOLP	0400	BITMOD	0111	C8CNTR	5205	CAL	6103
APTDRV	0334	BITPER	3135	C8CRLF	5526	CALC8	2342
APTERR	4557	BITS1	0004	C8CTLG	5465	CALOUT	6310
APTEX1	0360	BITS2	0005	C8CTLG	5457	CAM	7621
APTTIC	0351	BITS3	0006	C8CTLQ	5455	CAMSG	3611
APTTIM	0363	BITS4	0007	C8CTL8	5437	CASTOR	1505
ASAV	0142	BITSW1	1732	C8ECHO	5474	CBMSG	3614
ASKDSC	0473	BITSWT	2263	C8ENTR	5222	CBUF	1405
ASV	2723	BITTMP	0036	C8ERR	5351	CBUSY	0062
AUTO10	0010	BLODE	5140	C8EXT1	5534	CBUSY1	0114
AUTO11	0011	BPNTR1	3137	C8EXT2	5551	CDATA	1213
AUTO12	0012	BPNTR2	3140	C8FLD	5241	CDF	6201
AUTO13	0013	BPNTR3	3141	C8FLG	5403	CDFSAV	0121

CDFSV 0054  
 CDI 6203  
 CDRV 2540  
 CEMSG 3676  
 CFMSG 3617  
 CHK12 2704  
 CHK0 2603  
 CHKACT 1340  
 CHKAPT 5163  
 CHKB 2600  
 CHKCRC 5615  
 CHKDLT 5610  
 CHKERR 4600  
 CHKEX 2550  
 CHKHCR 5547  
 CHKOK 6132  
 CHKOUT 4700  
 CHKRDI 3760  
 CHKRET 2734  
 CHKSEC 2512  
 CHKSEK 0542  
 CHKSTA 4205  
 CHKTMP 2267  
 CHNGP 1056  
 CHNGSK 4111  
 CHRTEM 2365  
 CHRTMP 6313  
 CIF 6202  
 CKAPT 1757  
 CKSOUT 4267  
 CLRBFI 1314  
 CLRBFO 6554  
 CLRBFI 6536  
 CLRBFI 1301  
 CLRLPO 6563  
 CLRLPI 6546  
 CLRO 0203  
 CLRTAB 4513  
 CNFLG 0126  
 CNT1 0052  
 CNT2 0053  
 CNTLOD 5107  
 CNTR1 0120  
 CNTRD1 4255  
 CNTRD2 4275  
 CODCHK 1413  
 COMDA 0031  
 COMDB 0032  
 CONFLG 0074  
 CONGO 0230  
 CONMEM 0435  
 CONSD 4134

CONSET 0200  
 CONSV 1733  
 CONT1 1075  
 CONWRD 1045  
 CRLF 4434  
 CRLFDO 6703  
 CROUT 2071  
 CTLERR 0025  
 CURCYL 0002  
 CURDRV 0024  
 CURDV 0055  
 CYLGEN 0600  
 CYLLP1 2111  
 CYLLP2 2125  
 CYLLPA 1234  
 CYLLPB 1246  
 CYLMSG 4202  
 DACNT 0035  
 DATA 4616  
 DATAM 3713  
 DATCEX 6147  
 DATCHK 6000  
 DATDUM 4045  
 DATERN 0105  
 DATERR 0016  
 DATEXA 4763  
 DATGEN 4702  
 DATMSG 3735  
 DATPAT 0104  
 DBLCHK 1445  
 DBUSYA 0405  
 DC6061 4040  
 DC60LP 0521  
 DC61LP 0541  
 DCEND 7164  
 DCNT1 2440  
 DCODE 4477  
 DCR CER 0020  
 DCR CM 3655  
 DCTB60 7140  
 DCTB61 7146  
 DEC4 4427  
 DEC4ER 1645  
 DEC4EX 1653  
 DEC4LP 1607  
 DECBLD 1660  
 DECDIG 2436  
 DECPRN 4575  
 DECPRT 4436  
 DECTMP 1662  
 DELAY 0352  
 DEMSG 3644

DERFLG 0134  
 DERR 6424  
 DETST 5400  
 DEX 6153  
 DIGCNT 1661  
 DIGFLG 0060  
 DLPA 1537  
 DLPB 1542  
 DLTERR 0022  
 DLTMSG 3665  
 DLY1 1556  
 DLY2 1557  
 DLY3 1560  
 DLYCAL 0357  
 DMFLG 0112  
 DMPFLG 0060  
 DNaCTV 6235  
 DNCNT1 4312  
 DNMSG 3603  
 DNWAIT 4272  
 DOCLRF 4571  
 DONCNT 4311  
 DONLP 4277  
 DOXFER 1000  
 DP120 7134  
 DP121 7135  
 DP122 7136  
 DP123 7137  
 DPAS0 3155  
 DPAS1 3156  
 DPAS2 3157  
 DPAS3 3160  
 DPAT 1225  
 DPAT12 7134  
 DPAT8 7130  
 DPAT80 7130  
 DPAT81 7131  
 DPAT82 7132  
 DPAT83 7133  
 DPATN 4144  
 DPNT1 2413  
 DRDYA 4234  
 DRIVE 0327  
 DRIVE0 6714  
 DRIVE1 6756  
 DRIVE2 7020  
 DRIVE3 7062  
 DROP 4341  
 DROPNG 4345  
 DRPDVA 3424  
 DRV0 6710  
 DRV1 6711

DRV2 6712  
 DRV3 6713  
 DRV60 0025  
 DRVACT 4326  
 DRVBSY 0041  
 DRVCNT 0127  
 DRVEND 7123  
 DRVERR 0014  
 DRVGO 6644  
 DRVGON 5413  
 DRVLPI 0451  
 DRVLPI 0341  
 DRVMSG 4075  
 DRVON 4420  
 DRVPNT 6710  
 DRVRES 4134  
 DRVS 0446  
 DRVTMP 0135  
 DSC6X 0061  
 DSC6Y 0062  
 DSTATE 0000  
 DTEMP1 2437  
 DTERN 0137  
 DTMP1 6700  
 DTNPA 0137  
 DTPAT 0136  
 DVTMP 0050  
 DXFER 4101  
 DXLIM 1023  
 ENDSC 0033  
 ENDSV 2725  
 ENDWD 0034  
 EPNTR 2534  
 EPNTR1 2536  
 EREX1 4676  
 EREXT1 5712  
 EREXT2 5722  
 ERMSG 3606  
 ERR 4057  
 ERRAPT 2740  
 ERRCHK 4554  
 ERREG 0030  
 ERRFLG 5732  
 ERRMEM 0321  
 ERROR 4553  
 ERRPC 5730  
 ERRPCM 4627  
 ERRPNT 5763  
 ERSV 2726  
 ERTBL1 2541  
 ERTBL2 2567  
 ERTBL3 2645

ERTBL4	3165	HDOB	1020	K0277	0075	LOGDLT	5612
EVNCYL	1006	HDCHKA	4217	K03	0063	LOGERR	5756
EXADA	2073	HDRERR	6352	K0300	0076	LOGHCR	5600
/EXFER	6645	HDRMSG	3731	K040	0065	LOGHNF	5544
EXIT1	1107	HDTEMP	2261	K047	0066	LOGTRK	5603
FACBAD	7164	HDTMP	4271	K07	0064	LOMSG	4252
FATAL	6614	HDTMPA	1731	K0777	0077	LOSEC	1424
FATL	3721	HDW1	0021	K100	0046	LPA	2633
FILBF1	4750	HDW2	0022	K1000	0053	LPA1	2640
FILBF2	5044	HDW3	0023	K1234	0100	LPB	3000
FINRPT	5733	HEMSG	3622	K2000	0054	LPB1	3005
FINSEC	4664	HICYL	1314	K240	4566	LPDF	2645
FIXCHK	1112	HIMSG	4246	K260	0047	LPDF1	3012
FIXWD	1065	HISEC	1400	K3	0042	LSAV	4325
FIXWRD	0075	HNERR	0024	K377	0050	M01	0103
FLAGS	4020	HNFMMSG	3673	K400	0051	M04	0104
FLDBAD	7171	HRDERR	0012	K5	0043	M06	0105
FLDBD	3740	INCHAR	0045	K5670	0101	M1	0055
FLDLP	0277	INTRY	0441	K7	0044	M215	0106
FLDMMSG	4445	INDX10	0010	K70	0045	M4	0056
FLDP1	6701	INDX11	0011	K7007	0102	M400	0057
FLDS	0260	INDX12	0012	K777	0052	MADR	2463
FLGS	4042	INDX13	0013	KBDCHK	4142	MAIN	0235
FLGSAV	0130	INDX14	0014	KCDF	0123	MAINGO	1456
FOUNDB	1273	INDX15	0015	KCF	6030	MAXCYL	0106
FUNCOD	0115	INDX16	0016	KIE	6035	MAXMSG	4126
FUNGEN	0530	INDX17	0017	KILDRV	1163	MAXSEC	0107
FUNSAV	2717	INIT	0222	KION	0122	MDRV0	2537
FUNTBL	0744	INITCA	0031	KSF	6031	MEMERR	0422
FWCMMSG	3633	INITLP	0224	KXCDF	0107	MESAG	4564
FXWRD	0127	INMODE	0046	KXCDI	0110	MESSAGE	4424
GETHAD	3304	INPINT	2311	KYBDRD	4156	MESBSW	2031
GETBSF	3440	INPUT	4435	LCNTR	2535	MESCNT	2054
GETDAT	2500	INTOFF	4322	LIMCHK	4430	MESEXT	2050
GETHCW	6264	INTRET	4163	LIMEXD	4235	MESGO	2747
GETHDR	0503	INTSVC	4000	LIMFLG	0061	MESINT	2051
GETOCT	4567	IOTA	4045	LIMMSG	4063	MESLUP	2020
GETSR	4443	IOTB	4054	LINKSV	0111	MESOUT	2052
GETSTA	4562	IOTC	4060	LKSAV	2341	MESPNT	2055
GETSWR	4550	IOTCNG	0514	LKSV	4346	MESPRT	2000
GETWRD	1101	IOTD	4064	LM1024	4023	MESSAD	2755
GO	4555	IOTE	4070	LM39	4217	MESTMP	2056
GOEXIT	5143	IOTF	4074	LM4095	4007	MINCYL	0063
GOODEX	2242	IOTG	4350	LM4096	4015	MINMSG	4177
GTF	6004	ISV	2721	LM511	4211	MINSEC	0064
HARD	4072	IWCMSG	3630	LM7	4334	MLP1	2454
HCR CER	0021	K0070	0067	LMFLG	0113	MLP2	2456
HCR CM	3661	K0177	0070	LNK1	1663	MNCYL	0115
HCW1	0021	K0212	0071	LOC0	0000	MNSEC	0116
HCW2	0022	K0215	0072	LOCHK	2220	MOD12	6043
HCW3	0023	K0240	0073	LOCYL	1340	MOD8	6046
HDOA	1011	K0260	0074	LOGDCR	5623	MODE12	1122

MODE8	1133	OPMSG	3670	PRTSTA	4355	RETQ2	0656
MODLPA	6055	OPLIM	1000	PSR	0020	RETRY	5625
MODLPB	6107	OPRAT	4424	PSR1	0020	RIF	6224
MOVL	1465	PAKBAD	4304	PTPAS	3142	RLCA	6603
MOA	7501	PAKRD	4562	PTSET	2314	RLCB	6604
MO	7421	PAKWRT	4552	PWRFAL	4502	RLDC	6600
MXCYL	0140	PARSEC	6066	PWRFLG	0113	RLMA	6602
MXSEC	0141	PASCHK	3250	PWRST	1510	RLSA	6605
NDE	5475	PASLP	3335	QFLAG	0057	RLSBUF	3332
NEGWRD	1137	PASMSG	4604	QUEST	0602	RLSD	6601
NEWCYL	0003	PASPT	4442	RAN1	5161	RLSE	6617
NEWSAV	0143	PAT120	5036	RAN2	5162	RLSPT	3342
NEXIT	2113	PAT121	5064	RANBUF	1021	RLWC	6607
NEXTK1	1700	PAT122	5112	RANCNT	5020	RMF	6244
NEXTQ1	1243	PAT123	5140	RAND	4031	RRCA	6612
NMFLD	0134	PAT80	4636	RANDOM	4552	RRCB	6613
NODRVS	4315	PAT81	4676	RANSEC	1200	RRER	6610
NOFLD	4465	PAT82	4736	RANWRD	1055	RRSA	6614
NOINT	4525	PAT83	4776	RDCNT	2510	RRSI	6615
NOP61	6616	PATCDF	3104	RDEXA	2343	RRWC	6611
NOPION	5532	PATCNT	0132	RDF	6214	RSTART	0200
NOTBSY	4307	PATER	3050	RDFLD	3537	RSTFLG	0114
NOTINT	4035	PATLPA	4740	RDFLG	0073	RSTLP1	1540
NOTMSG	4332	PATLPB	5000	RDHDR	5243	RSTRT	4472
NOTOPI	5606	PATLPC	5034	RDONL	4427	RTF	6005
NOTYN	2107	PATMP	0136	RDONLY	0553	RTLIM	0130
NUMFLD	0102	PATMSG	3725	RDPK	2200	RUSURE	4224
NUMSEC	0126	PCHK	3350	RDPEX	2255	RWRET	5677
NUMWRD	4153	PCMSG	3716	RDS11	5257	S1MSG	3636
NXT	6243	PCNTR	3034	RDS12	5262	S2MSG	3641
NXTB	2471	PCSAVE	4016	RDS13	5265	SAMSG	3600
NXTCA	1716	PDATA	2533	RDS14	5271	SAVALL	2654
NXTCB	2217	POS1	0450	RDSIA	5230	SAVBMA	6172
NXTDA	1560	POSFLG	0036	RDSIB	5235	SAVDRV	6600
NXTDRV	0470	POSGO	5674	RDSMSG	3701	SAVWRD	0037
NXTDV1	0305	POSPC	5731	RDTRK	2264	SBE	6101
NXTSA	2041	POSSET	4400	RDYCHK	4347	SCHK	2132
NXTSD	2440	POSVER	0455	RDYFLG	0037	SCK1	5430
OCT1	4426	POSWRD	1147	REACHD	4413	SCK2	5451
OCT1ER	1721	PRINT	4432	READON	1200	SCK3	5456
OCTGET	6607	PRNAC	4570	READOP	6031	SCNT	2147
OCTGO	6613	PRNBA	6465	READPK	0270	SCNTB	2075
OCTONE	1726	PRNDAT	2514	READS	0026	SCNTC	2506
OCTOUT	1717	PRNDEC	2160	REDFLG	0125	SECADD	0032
ODCYL	1014	PRNINT	2165	RESALL	2730	SECCHK	1252
OFFSET	0010	PRNOUT	2164	RESET	4573	SECCNT	0040
OKEX	2130	PRNSTA	2441	RESTRT	4022	SECHK	6071
OLDCYL	0001	PRNT	4566	RETBAD	4533	SECLD	5124
ONAPT	0244	PRNTAC	4431	RETCNT	0110	SECLP8	2307
ONEFLD	1126	PRNTGO	1357	RETLIM	0076	SECLPA	2305
ONEUNT	0411	PRNTMP	2167	RETLM	3770	SECLPB	2321
OPIERR	0023	PRTDEC	2400	RETQ1	0643	SECLPD	2426

SECMMSG	4205	SPACES	4557	TRD	2216	WRTPAK	1600
SECRDA	2344	SPARE	6606	TRKCHK	2236	WRTPEX	1722
SECSAV	0140	SPL	6102	TRKCNT	1726	WRTPK	0254
SECTMP	2511	SPRNT	5356	TRKCT	2260	WRTPM	4576
SECW	2400	SRGET	3364	TRKERR	0017	WRTS	0027
SEEK	6314	SRQ	6003	TRYLPA	3461	WRTTRK	1734
SEEKS1	0010	SRQUES	0600	TRYLPB	3600	WRTUP	6143
SEEKS2	0011	SSV	2722	TRYOCT	1672	WSAV	6171
SEEKUP	3753	STACDI	2531	TRYYN	2100	WSV	2720
SEKCHK	0650	STACHK	4563	TTYCHK	4146	XADDRV	3200
SEKERR	0015	STACNT	6251	TWRT	1652	XC8CAL	6252
SEKLIM	0101	STAEX2	4270	UNINT	4572	XC8RET	6311
SEKMSG	3777	STAGET	5200	UNKINT	4200	XCDF1	1746
SEKOK	0661	STAOUT	2526	UPCHK	2266	XCDF2	4733
SEKOP	4104	STAPRT	4551	UPDAT	1755	XCDF3	4752
SEKRET	0077	START	0200	UPDBIT	3000	XCDF4	5027
SEMSG	3625	STAT6A	0026	UPDSEK	3200	XCDF5	5046
SETA	0675	STAT6B	0027	UPPNT	0047	XCDFA	1404
SETACT	3253	STATC	5521	UPSEX	3246	XCOMA	0033
SETBSY	5071	STATE4	4247	USE60	0510	XCOMB	0034
SETIDX	1400	STATES	4254	USE62	0503	XCRLF	2057
SETPNT	4547	STATGO	4363	USEBF1	1324	XDEC4	1600
SETPOS	4561	STATLP	6207	USEBF2	1334	XDELAY	1533
SETPT	4440	STATUS	6200	VALNEG	2226	XENDSC	0040
SETTMP	4021	STR TIN	4015	VT278	4576	XERROR	5321
SETUP	0244	SWP	7521	VT278B	4444	XFER	4560
SFLIM	0135	SWRGET	2151	WAITDN	4556	XFRLIM	0100
SFTERR	0013	TABED	0142	WATCNT	6422	XFRLM	0132
SFTLIM	0103	TABFIL	4433	WATCT1	6423	XGO	5067
SGFLG	0124	TABSTR	0112	WATLPA	6405	XINPUT	2270
SGLHD	0117	TBLCLR	3237	WATMOR	6414	XLCHK	2200
SILO1	5317	TBLED	0110	WATRDY	6400	XOCT1	1664
SILO2	5320	TBLLP	3247	WCHDR	3343	XPRINT	2130
SKEMSG	3647	TBLPNT	0112	WCHONE	4434	XRAND	5145
SKLIM	0133	TBLPRN	2507	WCHSUR	1302	XRESET	3114
SKLIM2	1037	TBLPT	0051	WCOUNT	4036	XSETEX	4343
SKON	6000	TBLSAV	0035	WDCNTR	0041	XSETP	4313
SKPASO	3151	TBLSET	2511	WDCTMP	0144	XTFIL	1727
SKPAS1	3152	TBLSTR	0060	WDCTR	3033	XTLP	1747
SKPAS2	3153	TBLWRD	1756	WPAK	4545	XVT278	3554
SKPAS3	3154	TDRIVE	3761	WRDCHK	4577	XVT27B	1527
SKRET	0131	TEMP1	0124	WRDCNT	0030	XWRDCH	0754
SKMSG	3710	TEMSG	3652	WRDSAV	0141	XXCRLF	6707
SKUPAD	3757	TENPWR	2432	WRDXFR	4131	XYESNO	2073
SLIM	0705	TIE	6045	WRT1	2000	YESRN	4565
SNGFLG	0072	TIMER	0362	WRTASK	3312	YESRNO	4425
SNGLHD	0065	TINIT	0611	WRTCNT	2074	YEXIT	2121
SNGLO	1266	TINIT1	0630	WRTEXA	2070	YNEXIT	2125
SNGSUR	4167	TITLE	4361	WRTLLOC	3142	YNGO	1350
SOFT	4067	TMP1	0056	WRTLPI	2023	YNOUT	2126
SOK	4551	TOOBAD	3724	WRTMSG	3704	YORN	4002
SPACE2	4572	TRANS	1422	WRTP	3400	ZCHKA	2667