

RP04

DISK PACK FORMATTER
MD-11-DERPL-B

EP-DERPL-B-DL
COPYRIGHT 1975
FICHE 1 OF 1

MAY 1978
digital
MADE IN USA

The image shows a microfiche card with 60 frames of data. The frames are arranged in a 10x6 grid. Each frame contains a small, high-contrast image of a document page, likely technical specifications or code. The frames are dark with light text and graphics. The right side of the page is a large, blank, light-colored area.

IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DERPL-B-D
PRODUCT NAME: RP04 DISK PACK FORMATTER
DATE CREATED: MAY 21, 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R. MOORE/C, HESS

COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

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

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

CONTENTS

SEQ 0002

- 1.0 ABSTRACT
- 2.0 REQUIRMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
 - 4.1 STARTING ADDRESS
 - 4.2 RESTARTING ADDRESS
 - 4.3 RH70/RP04 ADDRESSES
 - 4.4 SWITCH SETTINGS
- 5.0 OPERATING PROCEDURE
 - 5.1 OPERATOR ACTION
 - 5.2 PROGRAM ACTION
 - 5.3 SUBROUTINE ABSTRACTS
 - 5.3.1 'RP04'
 - 5.3.2 'DRINIT'
- 6.0 ERRORS
 - 6.1 ENTRY ERRORS
 - 6.2 ERROR TYPEOUTS = REPORTED WITHIN THE DRIVER
 - 6.3 ERROR TYPEOUTS = REPORTED AT THE RETURN FROM THE DRIVER
- 7.0 MISCELLANEOUS
 - 7.1 EXECUTION TIME
 - 7.2 HALTING PROGRAM
 - 7.3 DATA PATTERN
- 8.0 PROGRAM DESCRIPTION
 - 8.1 FORMAT OPERATION
 - 8.2 FORMAT ERRORS
 - 8.3 POSITIONER ERRORS
- 9. PROGRAM LISTING

1.0 ABSTRACT

 THE RP04 FORMATTER IS DESIGNED TO WRITE AND VERIFY
 HEADER AND DATA INFORMATION ON ALL POSSIBLE DISK
 PACK ADDRESSES WITH THE INTENTION OF TESTING THE
 RETENTIVITY OF THE RECORDING SURFACES, THE
 FORMAT IS MAINTAINED ON A BASIS OF 411 CYLINDERS,
 19 TRACKS PER CYLINDER AND 22 SECTORS PER TRACK.

2.0 REQUIRMENTS

2.1 EQUIPMENT

PDP-11 SYSTEM (12K CORE)
 RH70 MASSBUS CONTROLLER
 ONE OR MORE RP04 DISK DRIVES

2.3 PRELIMINARY PROGRAMS

ALL RP04 DIAGNOSTICS SHOULD HAVE PREVIOUSLY
 BEEN RUN SUCCESSFULLY.

3.0 LOADING PROCEDURE

USE THE STANDARD ABSOLUTE LOADER FROM THE 12K OR
 HIGHER BANK OF MEMORY.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESS

LOAD AND START AT ADDRESS 200

4.2 RESTARTING ADDRESS

LOAD AND START AT ADDRESS 204

4.3 RH70/RP04 ADDRESSES

THE PROGRAM HAS BEEN ASSEMBLED TO USE 176700 AS THE RH70/RP04 ADDRESS;
 THE VECTOR IS ASSUMED TO BE 254. TO CHANGE THESE ADDRESSES, ENTER
 THE PROPER ADDRESS INTO THE LOCATION(S) GIVEN BELOW.

LOCN	VALUE	FUNCTION
----	-----	-----
1100	176700	RH70/RP04 UNIBUS ADDRESS
1102	254	RH70 INTERRUPT VECTOR ADDRESS

4.4 SWITCH SETTINGS (SW#1)

SW15 HALT ON ERROR--- THE OPERATION IN PROGRESS WHEN THE ERROR OCCURED IS POINTED TO BY THE PC LOCATED ON THE STACK,

SW14 LOOP ON THE CURRENT CYLINDER AND TRACK--- ALLOWS THE USER TO LOCK ON A FAILING TRACK ETC.,

SW13 INHIBIT ERROR TYPEOUT

5.0 SW0 FORMAT (OR CHECK) CONTINUOUSLY OPERATING PROCEDURE

5.1 OPERATOR ACTION

1. TURN THE SELECTED DRIVE ON LINE.
2. DISABLE THE WRITE PROTECT ON THE SELECTED DRIVE.
3. WAIT FOR THE READY LIGHT INDICATION.
4. THE OPERATOR WILL RESPOND TO THE TYPED REQUESTS BY INPUTTING THE FOLLOWING PARAMETERS:

PARAMETER	VALUES	DEFAULT
DRIVE #	0-7	0
22 SECTOR (10 BIT) MODE	Y OR N	Y
PROGRAM MODE	F OR C	F
STARTING CYL,TRK	0-410,0-10	0,0 *
ENDING CYL,TRK	0-410,0-10	410,10 *
SELECT PATTERN	ZERO'S ONE'S WORST CASE	WORST CASE *

* NORMAL FULL PACK FORMAT OPERATIONS SHOULD DEFAULT TO THESE VALUES,

1. TYPING THE CHARACTER 'D' WILL ASSUME THE DEFAULT VALUES.
2. A 'CONTROL C' COMBINATION WILL RETURN USER BACK TO THE DRIVE NUMBER REQUEST DURING THE INPUT REQUESTS.
3. A RUBOUT WILL RETYPE THE CURRENT LINE.
4. THE INPUT FOR PARTIAL PACK FORMATS MUST HAVE THE CYL AND TRK SEPARATED BY A COMMA AND THE LINE TERMINATED WITH A CARRIAGE RETURN, THE ENDING CYL,TRK ADDRESS MUST BE EQUAL TO OR GREATER THAN THE STARTING CYL,TRK ADDRESS, ALL LIMITS ARE INCLUSIVE.
5. THE DATA PATTERN TO BE WRITTEN ON EACH SECTOR IS SELECTED BY TYPING THE DIGIT ASSOCIATED WITH THE PATTERN TYPED.
6. THE PROGRAM MAY BE RUN IN 'READ ONLY' MODE BY ENTERING A 'C' (CHECK) IN THE RESPONSE TO THE 'PROGRAM MODE' REQUEST MESSAGE, THE NORMAL OR DEFAULT MODE IS 'F' (FORMAT & VERIFY); THIS MODE IS SELECTED BY ENTERING A 'D' (DEFAULT) OR BY

- ENTERING AN 'F' IN RESPONSE TO THE 'MODE' REQUEST.
- 7. STARTING AND ENDING CYLINDER AND TRACK ADDRESS ENTRIES MUST BE IN FORMAT 'CCC,TT' WHERE 'CCC' IS THE CYLINDER ADDRESS AND 'TT' IS THE TRACK ADDRESS.
- 8. ALL ENTRIES EXCEPT THE 'SELECT PATTERN' ENTRY AND THE 'D' (DEFAULT) ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN.
- 9. THE PACK MAY BE FORMATTED IN 22 SECTOR (16 BIT) MODE OR IN 20 SECTOR (10 BIT) MODE. IF THE PACK IS FORMATTED IN 20 SECTOR MODE, BITS 16 & 17 OF EACH WORD WILL BE ZERO. ALL OF THE PDP-11 RPO4 DIAGNOSTICS REQUIRE 22 SECTOR FORMATS.

5.2 PROGRAM ACTION

AFTER THE PARAMETERS HAVE BEEN INPUTTED THE PROGRAM WILL FORMAT OR CHECK WITHIN THE LIMITS SPECIFIED. UPON COMPLETION OF THE FORMAT OR CHECK (SW0 = 0), 'FORMAT COMPLETE' OR 'CHECK COMPLETE' WILL BE TYPED AND A NEW DRIVE NUMBER WILL BE REQUESTED.

5.3 SUBROUTINE ABSTRACTS

5.3.1 'RPO4'

THIS ROUTINE IS THE DRIVER FOR THE RPO4 DISK SYSTEM. IT IS ENTERED UPON ALL REQUESTS TO THE RPO4 DRIVE.

5.3.2 'DRINIT'

THIS ROUTINE IS USED TO INITIALIZE THE RPO4 DRIVER. AT THE EXIT OF THIS ROUTINE THE AVAILABILITY OF EACH DRIVE HAS BEEN RECORDED IN TABLE DRVSTA.

6.0 ERRORS

6.1 ENTRY ERRORS

- 1. THE '??' MESSAGE IDENTIFIES AN ILLEGAL INPUT CHARACTER HAS BEEN TYPED.
- 2. THE '?? CYLINDER LIMIT EXCEEDED' MESSAGE INDICATES THE USER REQUESTED A CYLINDER GREATER THAN 410.
- 3. THE '?? TRACK LIMIT EXCEEDED' MESSAGE INDICATES THE USER REQUESTED A TRACK GREATER THAN 18.
- 4. THE '?? ENDING DSK ADRS MUST BE EQUAL TO OR GREATER THAN THE STARTING ADRS' MESSAGE IS SELF-EXPLANATORY.

6.2 ERROR TYPEOUTS- REPORTED WITHIN THE DRIVER ROUTINE

- 01 ILLEGAL SUBSYSTEM INTERRUPT

A SUBSYSTEM INTERRUPT OCCURED, BUT THE 'SC' BIT IN RHCS1 IS NOT SET,

SEG 0006

#2 ILLEGAL DRIVE INTERRUPT
DRIVE= DRIVE NUMBER
RHA= CONTENTS OF THE ATTENTION SUMMARY REGISTER

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

THE INDICATED DRIVE INTERRUPTED BUT THE DRIVE REGISTERS DO NOT SHOW A REASON FOR THE INTERRUPT.

#3 MASSBUS CONTROL BUS PARITY ERROR
RD,ADR= ADDRESS OF THE REGISTER READ
RD,WRD= CONTENTS OF THE WORD READ

A CONTROL PARITY ERROR WAS DETECTED BY THE CONTROL UNIT WHEN THE INDICATED REGISTER WAS READ.

#4 CONTROL BUS PARITY ERROR
WRT,AD= ADDRESS OF REGISTER WRITTEN
WRT,WD= CONTENTS OF WORD WRITTEN
RD,WRD= WORD READ BACK

THE ADDRESSED DRIVE DETECTED A PARITY ERROR WHEN THE CONTROLLER ATTEMPTED TO WRITE INTO THE INDICATED DRIVE REGISTER.

#5 ATTENTION FROM AN UNAVAILABLE DRIVE
DRIVE= DRIVE NUMBER
RHA= CONTENTS OF THE ATTENTION SUMMARY REGISTER

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

THE ATTENTION REGISTER HAS AN ATTENTION BIT SET FOR A DRIVE WHICH IS NOT CURRENTLY ACTIVE IN AN OPERATION.

6.3 ERROR TYPEOUTS- REPORTED AT THE RETURN FROM THE DRIVER

#1 WRITE ERROR
DRIVE=X SN=X = DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX = CYL & TRK WHERE ERROR OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

THIS ERROR INDICATES AN ERROR OCCURED WHEN TRYING TO DO A WRITE HEADER & DATA COMMAND, THE PROGRAM WILL NOT ADVANCE BEYOND THE CYL,TRK TYPED UNTIL THE ERROR DISOLVES.

#2 WRITE CHECK ERROR
DRIVE=X SN=X = DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX = DISK ADDRESS OF BAD SECTOR
SECTOR=XX = BAD SECTOR
GOOD DATA=XXXXXX = DATA IN CORE
BAD DATA=XXXXXX = DATA READ FROM DISK
RHEC1=XXXXXX = LOCATION OF ERROR BURST
RHEC2=XXXXXX = ERROR BURST

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

SEQ 0007

RETRIES UNSUCCESSFUL - SECTOR NOT ACCEPTABLE

THIS IS THE TYPICAL ERROR TYPEOUT EXPECTED
WHEN A FORMAT ERROR OCCURS, IT SUGGESTS THAT THE
DISK SURFACE SPECIFIED IS QUESTIONABLE, IF
THE MESSAGE 'RETRIES UNSUCCESSFUL' IS TYPED,
THE PACK IS NOT ACCEPTABLE,

03 HEADER COMPARE NOT MADE
DRIVE=X SN=X - DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX - CYL & TRK COMPARISON ATTEMPTED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

REFORMAT PACK

THIS ERROR INDICATES THE FAILURE TO READ A
HEADER SUCCESSFULLY, THE POSITIONER COULD HAVE
ADVANCED TOO FAR DURING THE FORMAT OPERATION,

04 SOFTWARE HEADER COMPARE ERROR
DRIVE=X SN=X - DRIVE & SERIAL NUMBER
EXPECTED - CYL,TRK=XXX,XX
READ - CYL,TRK=XXX,XX

REFORMAT PACK

THIS ERROR SUGGESTS THAT THE POSITIONER FAILED
TO ADVANCE DURING THE FORMAT OPERATION,

05 READ OR WRITE UNSAFE ERROR
DRIVE=X SN=X - DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX - CYL & TRK WHEN UNSAFE OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

IF AN UNSAFE OCCURS THE FORMAT IS TERMINATED, THE
FORMAT SHOULD BE RESTARTED AT OR BEFORE THE CYLINDER
WHERE THE UNSAFE DEVELOPED, THE TYPE OF UNSAFE
IS INDICATED BY THE CONTENTS OF RHER1, RHER2 OR RHER3.

06 DRIVE ERROR
DRIVE=X SN=X - DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX - CYL & TRK WHEN ERROR OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

TWO RETRIES ARE MADE ON THE CURRENT DISK ADDRESS
IF THE DRIVE ERROR REMAINS THE FORMAT IS TERMINATED,
IF NOT THE FORMAT CONTINUES,

07 DRIVE UNSAFE ERROR
DRIVE=X SN=X - DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX - CYL & TRK WHEN UNSAFE OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

SEQ 0000

IF A DRIVE UNSAFE DEVELOPS THE FORMAT IS TERMINATED,
THE TYPE OF UNSAFE IS INDICATED BY THE CONTENTS
OF RHER1, RHER2 OR RHER3,

#8 UNCORRECTABLE MASSBUS PARITY ERROR
FORMAT TERMINATED
CYL,TRK=XXX,XX = CYL & TRK WHEN ERROR OCCURED

THE PROGRAM HAS TRIED 3 TIMES TO READ OR WRITE
A REGISTER AND A PARITY ERROR HAS OCCURED EACH TIME,

#9 FATAL MASSBUS PARITY ERROR
CYL,TRK=XXX,XX = CYL & TRK WHEN ERROR OCCURED

THE PROGRAM DETECTED A MASSBUS PARITY ERROR
WHILE ISSUING A 'DRIVE CLEAR' TO RESET A PREVIOUSLY
DETECTED MASSBUS PARITY ERROR, THE PROGRAM WILL
HALT AFTER THE ERROR IS REPORTED,

#10 SOFTWARE TIMEOUT - SYSTEM HUNG
DRIVE=X SN=X = DRIVE & SERIAL NUMBER
CYL,TRK=XXX,XX = CYL & TRK WHEN OPERATION TIMED OUT

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3
CONTENTS OF THE ABOVE REGISTERS

IF A DISK OPERATION FAILS TO COMPLETE WITHIN 1 SECOND
THE ABOVE MESSAGE IS TYPED, THIS INDICATION ONLY
OCCURES IF THE PDP11 SYSTEM IS EQUIPPED WITH EITHER
A KW11-L OR KW11-P CLOCK,

7.0 MISCELLANEOUS

7.1 EXECUTION TIME

TO FORMAT THE ENTIRE DISK TAKES APPROXIMATELY 9
MINUTES, IT TAKES ABOUT 1.2 SECONDS TO FORMAT
ONE CYLINDER PLUS 7 MS FOR EACH CYLINDER SEEK,

7.2 HALTING THE PROGRAM

THE FORMAT SHOULD NOT BE INTERRUPTED ONCE UNDERWAY
UNLESS A REFORMAT IS TO FOLLOW,

7.3 DATA PATTERN

THE WORST CASE PATTERN OF 165555 & 133333 IS MORE
LIKELY TO CREATE DATA CHECK (DCK) ERRORS
THAN ONES OR ZEROS, IF A SUCCESSFUL PASS WITH ZEROS
IS ACHIEVED WHERE WORST CASE FAILS THEN THE
RP04 DIAGNOSTICS SHOULD BE RUN TO VERIFY THE
DRIVES RELIABILITY,

8.0 PROGRAM DESCRIPTION

8.1 FORMAT OPERATION

SEG 0009

THE FORMAT PROGRAM FORMATS THE DISK PACK ON THE ASSIGNED DRIVE ONE TRACK AT A TIME, THE DATA FIELDS ARE WRITTEN WITH THE SELECTED PATTERN, KEY WORDS ARE WRITTEN WITH ZERO'S. EACH TRACK IS VERIFIED WITH A WRITE CHECK ORDER IMMEDIATELY AFTER IT IS WRITTEN.

THE PORTION OF THE PACK TO BE FORMATTED IS DETERMINED BY THE BEGINNING AND THE ENDING CYLINDER AND TRACK ADDRESSES; THESE LIMITS ARE INCLUSIVE, A SINGLE TRACK IS THE SMALLEST ELEMENT THAT MAY BE FORMATTED.

8.2 FORMAT ERRORS

WRITE CHECK ERRORS ARE REPORTED WHEN THEY ARE DETECTED, IF AN ERROR IS DETECTED, THE SECTOR MUST BE REWRITTEN AND VERIFIED CORRECTLY TWO SUCCESSIVE TIMES TO BE CONSIDERED USEABLE, SECTORS WHICH CANNOT BE WRITTEN CORRECTLY TWICE AFTER AN ERROR WILL BE DECLARED NOT ACCEPTABLE BY THE PROGRAM.

8.3 POSITIONER ERRORS

AFTER THE LAST TRACK HAS BEEN FORMATTED AND VERIFIED AN ADDITIONAL CHECK IS PERFORMED, THE HEADER OF TRACK ZERO AND SECTOR ZERO OF EACH CYLINDER IS READ AND COMPARED BY THE SOFTWARE, THIS CHECK IS PERFORMED TO ISOLATE A POSSIBLE POSITIONER ERROR THAT MAY HAVE OCCURED DURING THE FORMAT OPERATION, TWO SUCH CASES OF POSITIONER MALFUNCTION ARE: FAILURE OF THE POSITIONER TO ADVANCE TO THE NEXT CYLINDER, AND THE CASE WHERE THE POSITIONER ADVANCES PAST THE CYLINDER DESIRED.

9.0 PROGRAM LISTING

DOCUMENT

MAINDEC-11-DERPL-B "RPO4 FORMATTER PROGRAM"

COPYRIGHT 1975
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS, 01754

TABLE OF CONTENTS

12	OPERATIONAL SWITCH SETTINGS
23	TRAP CATCHER
38	BASIC DEFINITIONS
156	RP04 ADDRESSES
161	MAIN PROGRAM
904	SUPPORT SUBROUTINES
1020	RH11/RP04 DRIVER (REV. 0.9)

2 COPYRIGHT (C) 1974,1975
DIGITAL EQUIPMENT CORP,
MAYNARD, MASS, 01754

PROGRAM BY R. MOORE/C. HESS

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
PACKAGE (MAINDEC-11-DZGAC-A2).

12

OPERATIONAL SWITCH SETTINGS

13

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON THE CURRENT TRACK
13	INHIBIT ERROR TYPEOUTS
0	LOOP THE PROGRAM

23

TRAP CATCHER

26 ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ",+2,HALT"
SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

30 STARTING ADDRESSES

32 200 = NORMAL START

34 204 = RESTART

38

BASIC DEFINITIONS

40 INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

51 GENERAL PURPOSE REGISTER DEFINITIONS

63 PRIORITY LEVEL DEFINITIONS

73 "SWITCH REGISTER" SWITCH DEFINITIONS

101 DATA BIT DEFINITIONS (BIT00 TO BIT15)

129 BASIC "CPU" TRAP VECTOR ADDRESSES

156
 RP04 ADDRESSES

161
 MAIN PROGRAM

904
 SUPPORT SUBROUTINES

1020
 RH11/RPS4 DRIVER (REV. 0,9)

1030 COPYRIGHT (C) 1974
 DIGITAL EQUIPMENT CORP.
 MAYNARD, MA 01754
 AUTHOR: JIM LACEY

1036 STORAGE FOR RHDS1, RHER1, RHER2, AND RHER3 ON AN ERROR "2" OR "5"

1037 RPERRS = RHDS1
 RPERRS+2 = RHER1
 RPERRS+4 = RHER2
 RPERRS+6 = RHER3

1047 TABLE OF DRIVE ACTIVE INDICATORS (DRVACT=8 BYTES)
 DRVACT=0 IMPLIES DRIVE IS IDLE
 DRVACT>0 IMPLIES DRIVE IS ACTIVE WITH A COMMAND
 DRVACT<0 IMPLIES DRIVE IS ACTIVE WITH AN ERROR RECOVERY OPERATION

1061 TABLE OF DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
 DRVSTA=0 IMPLIES DRIVE IS OFFLINE
 DRVSTA>0 IMPLIES DRIVE IS ONLINE
 DRVSTA<0 IMPLIES DRIVE IS UNSAFE OR NONEXISTENT

1075 TABLE OF DRIVE TYPES (DRVTYP=8 WORDS)
 DRVTYP WILL CONTAIN THE DRIVE TYPE OF ALL ONLINE, OFFLINE, AND
 UNSAFE DRIVES, IF A DRIVE IS NONEXISTENT DRVTYP WILL BE ZERO.

1080 TRANSFER WAIT FLAG (TRNSWT=1 WORD)
 THIS IS A ONE WORD QUEUE, IT WILL CONTAIN THE ADDRESS OF
 "DPB" OF THE I/O OPERATION,

1094 SEARCH WAIT KEYS (SRCHWT=1 WORD)
 THIS IS A ONE WORD QUEUE THAT WILL CONTAIN A KEY FOR EACH OF
 THE DRIVES THAT ARE PERFORMING A SEARCH COMMAND FOR THE I/O
 REQUEST THAT IS AT THE TOP OF THEIR REQUEST QUEUE,
 EACH DRIVE IS ASSIGNED ONE BIT, STARTING AT BIT00 FOR DRIVE 0.

- 1902 RP04 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
ACTDRV=0 IMPLIES DRIVER IS INACTIVE
ACTDRV>0 IMPLIES DRIVER IS ACTIVE
- 1908 SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
ACTSTR=0 IMPLIES SOFTWARE TIMER ROUTINE IS INACTIVE
ACTSTR>0 IMPLIES SOFTWARE TIMER ROUTINE IS ACTIVE
- 1914 UNLOAD FLAG (ULDFLG=8 BYTES)
ULDFLG=0 IMPLIES NO UNLOAD COMMAND
ULDFLG>0 IMPLIES UNLOAD COMMAND IN PROGRESS
ULDFLG<0 IMPLIES UNLOAD COMMAND IN WAIT QUEUE
- 1920 LOOK AHEAD COUNT (LACNT=8 BYTES)
LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED
- 1940 SAVE REGISTERS FLAG (SAVEFG =1 WORD)
SAVEFG <0 IMPLIES SAVE THE RH11/RP04 REGISTERS WHEN THE
OPERATION IS COMPLETED AS PER (DPB+14).
SAVEFG=0 IMPLIES SAVE THE RH11/RP04 REGISTERS, AS PER
(DPB+14), AFTER AN ERROR.
- 1948 SEEK FLAG (SEEKFG=1 WORD)
SEEKFG=0 IMPLIES WHEN THE DISK ADDRESS ISN'T IN THE WINDOW
FOR A DATA TRANSFER START A SEARCH COMMAND
SEEKFG<0 IMPLIES DATA TRANSFER WILL DO IMPLIED SEEKS,
DISREGARD THE WINDOW
- 1956 TIMEOUT TABLE (TIMER=8 WORDS)
THIS TABLE CONTAINS THE TIME ALLOWED FOR AN OPERATION
- 1968 DATA TRANSFER UNDERWAY INDICATOR (DTUW=1 WORD)
DTUW<0 IMPLIES NO DATA TRANSFER UNDERWAY
DTUW=N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
- 1974 ATTENTION BITS TABLE (ATABIT=8 BYTES)
THIS TABLE CONTAINS THE CORRESPONDING BIT TO EACH DRIVES
ATTENTION BIT
- 1987 RP04 TO RH11 "MASS CONTROL BUS PARITY ERRORS" (MCPE) ALLOWED BEFORE
CALLING IT FATAL (MCPMX=1 WORD)
- 1992 STORAGE FOR RPADR (THE FIRST ADDRESS (776700) OF THE RH11/RP04),
RPVEC (THE VECTOR ADDRESS (254)), AND RPVEC+2 (THE BR LEVEL (5)).
- 1997 MAXIMUM NUMBER OF LOOK AHEADS ALLOWED IS 4 (MXLACT=1 WORD)
- 1999 MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)

2001 MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)

2003 MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWNDW=1 WORD)

2006 DEFINITIONS OF THE RH11/RP04 ADDRESS INDEXES

2029 RH11/RP04 DRIVER INIT, CODE
THIS ROUTINE WILL DETERMINE WHICH RP04 DRIVES ARE AVAILABLE FOR TESTING AND SET THE DRVSTA INDICATOR TO THE PROPER STATE FOR EACH DRIVE,
NOTE: THIS ROUTINE CALLS DRVINT
CALL
 JSR PC,RPINIT
 RETURN

2070 DRIVE INIT, ROUTINE
THIS ROUTINE DETERMINES IF A DRIVE EXIST AND IF IT IS AN RP04, IF IT IS, A "READ-IN PRESET" IS ISSUED AND FMT22 IS SET TO A "1", THEN MOL, DPR, DRY, AND VV ARE CHECKED TO INSURE THEY ARE ALL ON A "1", AND DEPENDING ON THEIR STATE, DRVSTA IS SET TO THE PROPER CONDITION,
CALL
 MOV @DRVNUM,R1 ;DRIVE NUMBER TO R1
 MOV RPADR,R4 ;UNIBUS ADDRESS OF RH11/RP04 (RHCS1)
 JSR R0,DRVINT ;CALLED BY A JSR
 RETURN1 ;ERROR OCCURRED (PARITY)
 RETURN2 ;NORMAL RETURN

2132 REQUEST PRE-PROCESSOR-HANDLES SUBSYSTEM REQUEST
CALL
 JSR R0,@RP04 ;CALL THE RP04 DRIVER
 PNTADR ;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK
 RETURN1 ;RETURN HERE IF QUEUE IS FULL
 RETURN2 ;RETURN HERE IF REQUEST IS IN QUEUE OR THERE

2178 OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
CALL
 MOV @DRVNUM,R1 ;DRIVE NUMBER TO R1
 JSR PC,OPT ;SETUP A COMMAND

2215 COMMAND INITIATOR
CALL
 MOV @DRVNUM,R1 ;DRIVE NUMBER
 MOV @DPB,R2 ;ADDRESS OF DPB
 JSR PC,C1? ;C1? = C11,C13, OR C14
 ;WHERE:
 ;C11=DATA TRANSFER
 ;C12=SEARCH REQUESTED BY DATA XFER
 ;C14=NOT DATA TRANSFER

```

2400 LOOK AHEAD ROUTINE
CALL
      MOV      $DRVNUM,R1      ;DRIVE NUMBER
      MOV      $DPB,R2        ;POINT TO DPB
      JSR      R0,LA          ;GO CHECK THE WINDOW
      RETURN1 ;ERROR RETURN
      RETURN2 ;START A SEARCH
      RETURN3 ;START A DATA TRANSFER

2443 INTERRUPT SERVICE ROUTINE

2457 TRANSFER DONE ROUTINE

2484 SPECIAL CONDITION ROUTINE

2640 RP04 TIMER ROUTINE
CALL
      MOV      $TIME,=(SP)    ;ELAPSED TIME IN MILLISECONDS ON THE STACK
      JSR      R0,RPTMR      ;CALL RP04 TIME ROUTINE

2674 SOFTWARE TIMEOUT ROUTINE
CALL:  STO
      MOV      $DRVNUM,R1      ;DRIVE NUMBER
      JSR      R0,STO        ;CALL==DRVACT MUST BE NONZERO
      RETURN

2739 ROUTINE TO READ A RH11/RP04 REGISTER
CALL
      JSR      R0,RD,RP      ;GO READ A REGISTER
      INDEX   ;REG, INDEX FROM BASE
      ERRADR  ;ERROR ADDRESS==PROCESS ERROR STARTING
              ;AT THIS ADDRESS
      RETURN  ;CONTENTS OF REG, IS ON THE STACK

2782 ROUTINE TO WRITE A RH11/RP04 REGISTER
CALL
      MOV      DATA,=(SP)    ;DATA TO BE LOADED ON THE STACK
      JSR      R0,WRT,RP     ;CALL THE ROUTINE TO LOAD(WRITE) THE REG,
      INDEX   ;INDEX OF THE REGISTER TO BE LOADED
      ERRADR  ;ADDRESS TO RETURN TO ON AN ERROR
      RETURN  ;ERROR FREE RETURN

2826 ROUTINE TO SAVE THE RH11/RP04 REGISTERS AS PER DPB+14
CALL
      MOV      $DPBNUM,R2    ;DPB POINTER TO R2
      JSR      PC,SVRH11    ;SAVE THE DRIVES REG'S

2854 ROUTINE TO SET THE INTERRUPT WITHOUT GETTING A "TRE"
CALL
      MOV      $DRVNUM,R1    ;DRIVE NUMBER TO R1
      JSR      PC,SET,IE    ;SET "IE"
      RETURN
    
```

```

2875  QUEUE COUNT
2929  ROUTINE TO CLEAR ALL OF THE REQUEST QUEUES
      CALL
            JSR      PC,CLRQUE
2953  EMPTY THE QUEUE SPECIFIED BY R1
      CALL
            MOV      DRVNUM,R1      ;DRIVE NUMBER TO R1
            JSR      PC,EMPTYQ
2965  ROUTINE TO PUT A REQUEST IN QUEUE
      CALL
            MOV      @DRVNUM,R1      ;DRIVE NUMBER
            MOV      @DPB,R2         ;ADDRESS OF PARAMETER BLOCK
            JSR      R0,DRVQUE       ;GO PUT REQUEST IN QUEUE
2971  RETURN1
      RETURN2
            ;RETURN HERE IF QUEUE IS FULL
            ;RETURN HERE IF REQUEST IS IN QUEUE
2987  ROUTINE TO GET THE "DPB" ADDRESS OF NEXT REQUEST IN QUEUE
      CALL
            MOV      @DRVNUM,R1      ;DRIVE NUMBER TO R1
            JSR      PC,GETREQ       ;GO GET THE REQUEST
            RETURN                  ;R2="DPB" ADDRESS OF THE REQUEST
            ;R2=0 IF NO REQUEST IN QUEUE
3003  ROUTINE TO "POP" THE REQUEST FROM QUEUE
      CALL
            MOV      @DRVNUM,R1      ;DRIVE NUMBER TO R1
            JSR      PC,POPQUE       ;CALL TO REMOVE REQUEST
            RETURN                  ;R2=ADDRESS OF DPB REMOVED
3020  ROUTINES TO SAVE R0-R5 AND R1-R5
      CALL:  SAVR05
            JSR      R0,SAVR05
            RETURN                  ;R0-R5 IS ON THE STACK
3025  CALL:  SAVR15
            JSR      R0,SAVR15
            RETURN                  ;R1-R5 IS ON THE STACK
      UPON RETURN FROM SAVR05 AND SAVR15 THE STACK WILL LOOK LIKE:

```

```

+12  R0
+10  R1
+06  R2
+04  R3
+02  R4
TOP  R5

```

3040 ROUTINES TO RESTORE R0-R5 AND R1-R5

CALL: GETR05
JSR R0,GETR05
RETURN ;R0-R5 HAVE BEEN RESTORED

CALL: GETR15
JSR R0,GETR15
RETURN ;R1-R5 HAVE BEEN RESTORED

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

;TITLE MAINDEC-11-DERPL-B "RP04 FORMATTER PROGRAM"
;COPYRIGHT (C) 1974,1975
;DIGITAL EQUIPMENT CORP.
;MAYNARD, MASS, 01754
;
;PROGRAM BY R. MOORE/C. HESS
;
;THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;PACKAGE (MAINDEC-11-DZQAC-A2).
;

.SBTTL OPERATIONAL SWITCH SETTINGS
;
; SWITCH USE
; -----
; 15 HALT ON ERROR
; 14 LOOP ON THE CURRENT TRACK
; 13 INHIBIT ERROR TYPEOUTS
; 0 LOOP THE PROGRAM

.SBTTL TRAP CATCHER
;
;=0
;ALL UNUSED LOCATIONS FROM 4 = 776 CONTAIN A ".+2,HALT"
;SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

;STARTING ADDRESSES
;=200
;200 = NORMAL START
; JMP BEGIN
;204 = RESTART
; JMP RESTRY

.SBTTL BASIC DEFINITIONS
;INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
;EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
;EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
PS= 177776 ;;PROCESSOR STATUS WORD
;EQUIV PS,PSW
STKLMT= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
SWR= 177570 ;;SWITCH REGISTER
DISPLAY=SWR

;GENERAL PURPOSE REGISTER DEFINITIONS
R0= 00 ;;GENERAL REGISTER
R1= 01 ;;GENERAL REGISTER
R2= 02 ;;GENERAL REGISTER
  
```

000000

000200

000200 000137 001104

000204 000137 001156

001100

177776

177774

177772

177570

177570

000000

000001

000002

55	000003	R3=	03	;;GENERAL REGISTER
56	000004	R4=	04	;;GENERAL REGISTER
57	000005	R5=	05	;;GENERAL REGISTER
58	000006	R6=	06	;;GENERAL REGISTER
59	000007	R7=	07	;;GENERAL REGISTER
60		,EQUIV	R6,SP	;;STACK POINTER
61		,EQUIV	R7,PC	;;PROGRAM COUNTER
62				
63		;PRIORITY LEVEL DEFINITIONS		
64	000000	PR0=	0	;;PRIORITY LEVEL 0
65	000040	PR1=	40	;;PRIORITY LEVEL 1
66	000100	PR2=	100	;;PRIORITY LEVEL 2
67	000140	PR3=	140	;;PRIORITY LEVEL 3
68	000200	PR4=	200	;;PRIORITY LEVEL 4
69	000240	PR5=	240	;;PRIORITY LEVEL 5
70	000300	PR6=	300	;;PRIORITY LEVEL 6
71	000340	PR7=	340	;;PRIORITY LEVEL 7
72				
73		; "SWITCH REGISTER" SWITCH DEFINITIONS		
74	100000	SW15=	100000	
75	040000	SW14=	40000	
76	020000	SW13=	20000	
77	010000	SW12=	10000	
78	004000	SW11=	4000	
79	002000	SW10=	2000	
80	001000	SW09=	1000	
81	000400	SW08=	400	
82	000200	SW07=	200	
83	000100	SW06=	100	
84	000040	SW05=	40	
85	000020	SW04=	20	
86	000010	SW03=	10	
87	000004	SW02=	4	
88	000002	SW01=	2	
89	000001	SW00=	1	
90		,EQUIV	SW09,SW9	
91		,EQUIV	SW08,SW8	
92		,EQUIV	SW07,SW7	
93		,EQUIV	SW06,SW6	
94		,EQUIV	SW05,SW5	
95		,EQUIV	SW04,SW4	
96		,EQUIV	SW03,SW3	
97		,EQUIV	SW02,SW2	
98		,EQUIV	SW01,SW1	
99		,EQUIV	SW00,SW0	
100				
101		;DATA BIT DEFINITIONS (BIT00 TO BIT15)		
102	100000	BIT15=	100000	
103	040000	BIT14=	40000	
104	020000	BIT13=	20000	
105	010000	BIT12=	10000	
106	004000	BIT11=	4000	
107	002000	BIT10=	2000	
108	001000	BIT09=	1000	

```

109      000400      BIT00= 400
110      000200      BIT07= 200
111      000100      BIT06= 100
112      000040      BIT05= 40
113      000020      BIT04= 20
114      000010      BIT03= 10
115      000004      BIT02= 4
116      000002      BIT01= 2
117      000001      BIT00= 1
118      ,EQUIV      BIT09,BIT9
119      ,EQUIV      BIT08,BIT8
120      ,EQUIV      BIT07,BIT7
121      ,EQUIV      BIT06,BIT6
122      ,EQUIV      BIT05,BIT5
123      ,EQUIV      BIT04,BIT4
124      ,EQUIV      BIT03,BIT3
125      ,EQUIV      BIT02,BIT2
126      ,EQUIV      BIT01,BIT1
127      ,EQUIV      BIT00,BIT0
128
129      ;BASIC "CPU" TRAP VECTOR ADDRESSES
130      000004      ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
131      000010      RESVEC= 10         ;;RESERVED AND ILLEGAL INSTRUCTIONS
132      000014      TBITVEC=14        ;; "T" BIT
133      000014      TRTVEC= 14         ;;TRACE TRAP
134      000014      BPTVEC= 14         ;;BREAKPOINT TRAP (BPT)
135      000020      IOTVEC= 20         ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
136      000024      PWRVEC= 24         ;;POWER FAIL
137      000030      EMTVEC= 30         ;;EMULATOR TRAP (EMT) **ERROR**
138      000034      TRAPVEC=34         ;; "TRAP" TRAP
139      000060      TKVEC= 60          ;;TTY KEYBOARD VECTOR
140      000064      TPVEC= 64          ;;TTY PRINTER VECTOR
141      000240      PIRQVEC=240        ;;PROGRAM INTERRUPT REQUEST VECTOR
142
143
144
145      ;REGISTER ADDRESSES
146      177560      TKB=177560
147      177562      TKB=177562
148      177564      TPA=177564
149      177566      TPA=177566
150      172540      0LKCSR=172540      ;ADRS OF KW11-P STATUS REG
151      172542      0LKCSB=172542     ;ADRS OF KW11-P COUNTER BUFFER
152      177546      0LKS=177546       ;ADRS OF KW11-L STATUS REG
153
154      001100      ,=1100
155
156      ,SBTTL      RP04 ADDRESSES
157
158      001100      176700      0RPADR: ,WORD 176700      ;RH11/RP04 UNIBUS ADDRESS
159      001102      000254      0RPVEC: ,WORD 254         ;RH11 INTERRUPT VECTOR
160
161      ,SBTTL      MAIN PROGRAM
162

```

```

163 001104 000005          BEGIN:  RESET          ;CLEAR WORLD
164 001106 012706 001100    MOV          0STACK,SP      ;SET UP STACK
165 001112 012737 003116 000030  MOV          0EMTHND,0030   ;SET UP EMT ADRS
166 001120 012737 000340 000032  MOV          0340,0032     ;SET PS TO 7
167 001126 013737 001100 013070  MOV          0RPADR,RPADR   ;MOVE RH11/RP04 ADDRESSES TO VECTOR
168 001134 013737 001102 013072  MOV          0RPVEC,RPVEC   ;RH11 VECTOR ADDRESS
169 001142 012737 000240 001156  MOV          0240,RESTRT    ;ALLOW RESTARTS AFTER INITIAL START
170 001150 004537 005166          JSR          R5,TYPOUT      ;GO TYPE MESSAGE
171 001154 007466          MTITLE          ;ADRS OF 'TITLE' MESSAGE
172 001156 000752          RESTRT: BR          BEGIN    ;CHANGE TO 'NOP' AFTER INITIAL START
173 001160 012706 001100    MOV          0STACK,SP      ;SET UP STACK
174 001164 000005          RESET          ;CLEAR WORLD
175
176          ;GO FIND OUT WHAT DRIVE
177
178 001166 004537 005166          M0:   JSR          R5,TYPOUT      ;GO TYPE MESSAGE
179 001172 007546          MUNIT          ;ADRS OF MESSAGE
180 001174 004737 005242          JSR          PC,TTI        ;GO WAIT FOR DRIVE 0
181 001200 022700 000104          CMP          0104,R0       ;IS IT THE DEFAULT CHAR?
182 001204 001006          BNE          10           ;BRANCH IF NOT THE DEFAULT CHAR
183 001206 005037 012540          CLR          USEL          ;SELECT DRIVE ZERO
184 001212 004537 005166          JSR          R5,TYPOUT      ;GO TYPE DEFAULT DRIVE NO.
185 001216 007561          MDRVD          ;ADRS OF MSG
186 001220 000413          BR          GSEL          ;GO SEE IF DRIVE ZERO IS THERE
187
188 001222 004737 005314          10:   JSR          PC,NUMTST     ;IS IT A DIGIT?
189 001226 000757          BR          M0           ;RETURN HERE IF DRIVE 0 NOT A DIGIT
190 001230 022700 000007          CMP          07,R0        ;IS IT LESS THAN 7?
191 001234 103003          BCC          20           ;YES - GO SELECT DRIVE
192 001236 004737 005406          JSR          PC,QUES       ;TYPE ??
193 001242 000751          BR          M0           ;RETYPE LINE
194 001244 010037 012540          20:   MOV          R0,USEL     ;SAVE IN USEL
195
196          ;THIS CODE CHECKS IF REQUIRED DRIVE IS AVAILABLE
197
198 001250 004737 013106          GSEL:  JSR          PC,RPINIT    ;GO SEE WHAT DRIVES ARE AVAILABLE
199 001254 012737 177777 013032  MOV          0177777,SEEKFG ;SET TO -1 IF NO OPTIMIZATION IN DRIVER
200 001262 013700 012540          MOV          USEL,R0       ;GET DRIVE NO.
201 001266 105760 012752          TSTB         DRVSTA(R0)    ;LOOK AT DRIVE STATUS
202 001272 003035          BGT          M1           ;BRANCH IF ONLINE
203 001274 002404          BLT          10           ;BRANCH IF UNSAFE OR NONEXISTENT
204 001276 004537 005166          JSR          R5,TYPOUT      ;TYPE 'DRIVE OFFLINE'
205 001302 007577          MOFFLN          ;ADRS OF MESSAGE
206 001304 000730          BR          M0           ;GO GET DRIVE 0 AGAIN
207 001306 006300          10:   ASL          R0         ;R0 INTO WORD INDEX
208 001310 005760 012762          TST         DRVSTYP(R0)    ;A DRIVE PRESENT?
209 001314 001004          BNE          20           ;BR IF 0
210 001316 004537 005166          JSR          R5,TYPOUT      ;TYPE 'DRIVE NOT PRESENT'
211 001322 007621          MDRNP          ;ADRS OF MSG
212 001324 000720          BR          M0           ;GO GET DRIVE NO, AGAIN
213 001326 022760 020020 012762  20:   CMP          020020,DRVSTYP(R0) ;SINGLE PORT?
214 001334 001410          BEQ          30           ;BR IF UNSAFE SINGLE PORT
215 001336 022760 024020 012762          CMP          024020,DRVSTYP(R0) ;DUAL PORT?
216 001344 001404          BEQ          30           ;BR IF UNSAFE DUAL PORT

```

```

217 001346 004537 005166 JSR R5,TYPOUT ;TYPE 'NOT AN RP04'
218 001352 007646 MNRP04 ;ADDR OF MSG
219 001354 000704 BR M0 ;GO GET DRIVE NO, AGAIN
220 001356 004537 005166 301 JSR R5,TYPOUT ;TYPE 'RP04 UNSAFE'
221 001362 007673 MUSD R ;ADDR OF MSG
222 001364 000700 BR M0 ;GO GET DRIVE NO, AGAIN
223
224 ;SEE WHICH MODE THE PROGRAM IS TO BE RUN IN,
225 ;MODES ARE: 'FORMAT & VERIFY' OR 'CHECK FORMAT'
226
227 001366 005037 012544 M11 CLR MODE ;SET MODE TO 'CHECK FORMAT'
228 001372 004537 005166 JSR R5,TYPOUT ;TYPE 'PROGRAM MODE'
229 001376 007725 MMODE ;ADDR OF MSG
230 001400 004737 005242 JSR PC,TTI ;WAIT FOR ENTRY
231 001404 022700 000104 CMP 0'D,R0 ;'D' (DEFAULT) ENTERED ?
232 001410 001415 BEQ 20 ;BR IF DEFAULT
233 001412 022700 000103 CMP 0'C,R0 ;CHECK FORMAT MODE ?
234 001416 001406 BEQ 10 ;BR IF CHECK ONLY
235 001420 022700 000106 CMP 0'F,R0 ;FORMAT & VERIFY ?
236 001424 001413 BEQ 30 ;BR IF YES
237 001426 004737 005406 JSR PC,QUES ;TYPE '??'
238 001432 000755 BR M1 ;TRY AGAIN
239 001434 004537 005166 101 JSR R5,TYPOUT ;TYPE REST OF 'CHECK'
240 001440 010004 MHECK ;ADDR OF MSG
241 001442 000412 BR M1A ;GET STARTING ADDRESS
242 001444 004537 005166 201 JSR R5,TYPOUT ;TYPE DEFAULT MESSAGE
243 001450 007761 MFORMT ;ADDR OF MSG
244 001452 000403 BR 40 ;SET UP MODE
245 001454 004537 005166 301 JSR R5,TYPOUT ;TYPE REST OF 'FORMAT'
246 001460 010016 MORMAT ;ADDR OF MSG
247 001462 012737 177777 012544 401 MOV 0=1,MODE ;SET MODE TO 'FORMAT & VERIFY'
248
249 ;FIND OUT IF FORMAT IS TO BE IN 20 OR 22 SECTOR MODE
250
251 001470 004537 005166 M1A1 JSR R5,TYPOUT ;TYPE FORMAT MODE REQUEST
252 001474 010030 MSIZE ;ADDRESS OF MESSAGE
253 001476 004737 005242 JSR PC,TTI ;WAIT FOR INPUT
254 001502 022700 000131 CMP 0'Y,R0 ;IS ENTRY A 'Y' ?
255 001506 001411 BEQ 10 ;BR IF IT IS
256 001510 022700 000116 CMP 0'N,R0 ;IS ENTRY A 'N' ?
257 001514 001431 BEQ 20 ;BR IF IT IS
258 001516 022700 000104 CMP 0'D,R0 ;IS ENTRY A 'D' ?
259 001522 001403 BEQ 10 ;BR IF IT IS
260 001524 004737 005406 JSR PC,QUES ;TYPE '??' - INVALID INPUT
261 001530 000757 BR M1A ;TRY AGAIN
262 001532 004537 005166 101 JSR R5,TYPOUT ;TYPEOUT MODE SELECTED
263 001536 010106 MSEC22 ;ADDRESS OF MESSAGE
264 001540 012737 013130 012602 MOV 0<260,022,>,MC ;TRACK SIZE IN 22 SECTOR MODE
265 001546 012737 164650 012604 MOV 0-<260,022,>,MWC ;2'S COMPLEMENT WORD COUNT
266 001554 012737 177777 012610 MOV 0=1,SEC20 ;22 SECTOR INDICATOR
267 001562 012737 000025 012612 MOV 021,,MAXSEC ;MAX SECTOR ADDRESS IN 22 SECTOR MODE
268 001570 112737 000020 012713 MOV 020,FMTDPB+1 ;LOAD FMT22 BIT
269 001576 000420 BR 30 ;LOAD FORMAT BIT
270 001600 004537 005166 201 JSR R5,TYPOUT ;TYPE OUT 20 SECTOR MODE SELECTED

```

```

271 001604 010165 MSEC20 ;ADDRESS OF MESSAGE
272 001606 012737 012120 012602 MOV 0<260,020,>,WC ;TRACK SIZE IN 20 SECTOR MODE
273 001614 012737 165660 012604 MOV 0-<260,020,>,MWC ;2'S COMPLEMENT WORD COUNT
274 001622 005037 012610 CLR SEC20 ;20 SECTOR INDICATOR
275 001626 012737 000023 012612 MOV 019,,MAXSEC ;MAX SECTOR ADDRESS IN 20 SECTOR MODE
276 001634 105037 012713 CLR0 FMTDPB+1 ;CLEAR THE FMT22 BIT
277 001640 113737 012540 012712 301 NOVB USEL,FMTDPB ;LOAD DRIVE NUMBER
278 001646 112737 000143 012714 NOVB 0143,FMTDPB+2 ;'SET FORMAT' COMMAND
279 001654 004037 013476 JSR R0,RPO4 ;GO TO THE DRIVER TO EXECUTE THE COMMAND
280 001660 012712 FMTDPB ;DPB ADDRESS
281 001662 000766 BR 30 ;QUEUE FULL RETURN - RETRY THE COMMAND
282 001664 005737 012730 401 TST FMTDPB+16 ;CHECK STATUS WORD
283 001670 001775 BEQ 40 ;BR IF NOT FINISHED
284 001672 100762 BMI 30 ;BR IF ERROR IN SET FORMAT
285 001674 000400 BR M2
286
287 ;GO GET PARAMETERS - STARTING CYL & TRK
288 ;AND ENDING CYL & TRK
289
290 001676 004537 005166 M21 JSR R5,TYPOUT ;GO TYPE MESSAGE 'STARTING CYL,TRK'
291 001702 010244 MNINCT ;ADRS OF MESSAGE
292 001704 012737 000001 012542 MOV 01,SOF5W ;SET UP SOF5W TO INDICATE STARTING INPUT
293 001712 004737 005446 JSR PC,GETDAT ;GO GET STARTING CYL & TRK
294 001716 000767 BR M2 ;RETURN HERE IF ERROR OR RUBOUT
295 001720 005037 012542 CLR SOF5W ;SET UP SOF5W TO INDICATE ENDING INPUT
296 001724 004537 005166 M31 JSR R5,TYPOUT ;GO TYPE MESSAGE 'ENDING CYL,TRK'
297 001730 010277 MNMAXCT ;ADRS OF MESSAGE
298 001732 004737 005446 JSR PC,GETDAT ;GO GET ENDING CYL & TRK
299 001736 000772 BR M3 ;RETURN HERE IF ERROR OR RUBOUT
300
301 ;THIS CODE CHECKS THE ORDER AND MAGNITUDE OF THE CYL & TRK ADRS
302 ;INPUTTED BY THE TTY, ALSO ESTABLISHES THE TOTAL TRK COUNT
303
304 001740 013700 012546 CKADRS: MOV ECYL,R0 ;SET R0 WITH ECYL
305 001744 163700 012550 SUB SCYL,R0 ;SEE IF END CYL IS SMALLER THAN START CYL
306 001750 103004 BCC 10 ;BRANCH IF NOT
307 001752 004537 005166 JSR R5,TYPOUT ;TYPE IMPROPER DSK ADRS
308 001756 010423 MADRER ;ADRS OF MESSAGE
309 001760 000746 BR M2 ;RETYPE REQUEST - INPUT CYL NOT IN PROPER FORM
310 001762 001014 101 BNE 30 ;BRANCH IF SCYL NOT EQUAL TO ECYL
311 001764 013700 012552 MOV ETRK,R0 ;SET UP R0 WITH ETRK
312 001770 163700 012554 SUB STRK,R0 ;SEE IF END TRK IS SMALLER THAN START TRK
313 001774 103004 BCC 20 ;BRANCH IF NOT
314 001776 004537 005166 JSR R5,TYPOUT ;TYPE IMPROPER DSK ADRS
315 002002 010423 MADRER ;ADRS OF MESSAGE
316 002004 000734 BR M2 ;RETYPE REQUEST - INPUT TRK NOT IN PROPER FORM
317 002006 010037 012556 201 MOV R0,TTRKS ;SAVE 0 OF TRACKS
318 002012 000417 BR M4 ;GO GET DATA PATTERN
319 002014 005037 012556 301 CLR TTRKS ;CLEAR TOTAL TRACK LOC
320 002020 062737 000023 012556 401 ADD 023,TTRKS ;ADD CYLINDER WORTH OF TRACKS
321 002026 005300 DEC R0 ;MAINTAIN CONTROL
322 002030 001373 BNE 40 ;BRANCH IF CYL DIFF NOT 0
323 002032 013700 012554 MOV STRK,R0 ;PUT STRK IN R0
324 002036 005400 NEG R0 ;MAKE TWO'S COMP

```

```

325 002040 060037 012556          ADD    R0,TRKS          ;SUB PARTICAL TRK
326 002044 063737 012552 012556  ADD    ETRK,TRKS       ;ADD IN END TRACKS
327
328
329                                ;GO GET DATA PATTERN FOR FORMAT
330 002052 004537 005166          M4:   JSR    R5,TYPOUT    ;GO TYPE 'SELECT PATTERN'
331 002056 010530                    MSELF ;ADRS OF MESSAGE
332 002060 004737 005242          JSR    PC,TTI          ;GO GET REPLY
333 002064 022700 000104          CMP    #104,R0        ;IS IT THE DEFAULT CHAR?
334 002070 001006                    BNE    10              ;BRANCH IF NOT
335 002072 004537 005166          JSR    R5,TYPOUT    ;TYPE DEFAULT PATTERN
336 002076 010632                    MPAID ;ADRS OF MSG
337 002100 012700 000002          MOV    #2,R0          ;SELECT PATTERN OF WORST CASE
338 002104 000411                    BR     20              ;GO SAVE PATTERN
339 002106 004737 005314          10:   JSR    PC,NUMTST    ;SEE IF A NUMBER
340 002112 000757                    BR     M4              ;BRANCH IF NOT
341 002114 020027 000003          CMP    R0,#3          ;IS # LARGER THAN 2
342 002120 103403                    BCS   20              ;BRANCH IF NOT
343 002122 004737 005406          JSR    PC,QUES        ;TYPE ??
344 002126 000751                    BR     M4              ;RETYPE LINE
345 002130 010037 012564          20:   MOV    R0,PSEL       ;SAVE PATTERN SELECTED
346 002134 004537 005166          JSR    R5,TYPOUT    ;TYPE 'SELECTED'
347 002140 007713                    MSEL ;ADRS OF MESSAGE
348
349                                ;GO TYPE 'STARTING FORMAT ON DRIVE # X'
350
351 002142 005737 012544          M5:   TST    MODE          ;'FORMAT' OR 'CHECK' MODE ?
352 002146 001404                    BEQ   10              ;BR IF 'CHECK' MODE
353 002150 004537 005166          JSR    R5,TYPOUT    ;TYPE 'STARTING FORMAT ON DRIVE #'
354 002154 010646                    MFOU  ;ADRS OF MESSAGE
355 002156 000403                    BR     20              ;
356 002160 004537 005166          10:   JSR    R5,TYPOUT    ;TYPE 'STARTING CHECK ON DRIVE #'
357 002164 010704                    MCHK  ;ADDR OF MSG
358 002166 004737 005350          20:   JSR    PC,UASH        ;GO PRINT DRIVE # IN USE
359
360
361                                ;THIS CODE SETS UP THE SELECTED DATA PATTERN TO BE WRITTEN ON EACH SECTOR IMAGE
362
363 002172 005037 012566          STUP: CLR    PATA          ;CLEAR DATA PATTERN A
364 002176 005037 012570          CLR    PATB          ;CLEAR DATA PATTERN B
365 002202 005737 012564          TST    PSEL          ;SEE IF PATTERN OF ONES
366 002206 001416                    BEQ   STUP           ;BR IF SO
367 002210 005137 012566          COM    PATA          ;SET PATA TO ONES
368 002214 005137 012570          COM    PATB          ;SET PATB TO ONES
369 002220 022737 000001 012564  CMP    #1,PSEL        ;SEE IF PATTERN OF ZEROS
370 002226 001406                    BEQ   STUP           ;BRANCH IF SO
371 002230 012737 165555 012566  MOV    #165555,PATA   ;SET UP WORST CASE
372 002236 012737 133333 012570  MOV    #133333,PATB   ;SET UP WORST CASE
373
374                                ;THIS CODE DUMP THE SELECT DATA PATTERN IN THE DATA BUFFER IN CORE
375
376 002244 013703 012602          STUP: MOV    #C,R3          ;SET UP COUNTER
377 002250 012700 020300          MOV    #0BFF,R0       ;SET UP MEMORY POINTER
378 002254 013701 012566          MOV    PATA,R1        ;SET UP PATTERN IN R1

```

```

379 002260 013702 012570      MOV      PATB,R2      ;SET UP PATTERN IN R2
380 002264 010120      101      MOV      R1,(R0)+    ;MOV 1ST PAT INTO MEM
381 002266 010220      MOV      R2,(R0)+    ;MOV 2ND PAT INTO MEM
382 002270 005303      DEC      R3          ;KEEP COUNT
383 002272 005303      DEC      R3          ;KEEP COUNT
384 002274 001373      BNE     10          ;DO IT AGAIN IF R3 NOT 0
385
386                          ;THIS CODE SETS UP FOR THE ACTUAL FORMAT
387
388 002276 004737 006016      WRHDR1: JSR     PC,SETTBL    ;GO SET UP DRIVER TABLE
389 002302 004737 006320      JSR     PC,SETHDR    ;GO INITIALIZE HEADERS IN THE SECTOR BUFFER IN CORE
390 002306 004737 007340      JSR     PC,CKCLK     ;GO START UP THE CLOCK IF AVAILABLE
391 002312 005037 012542      WRHDR2: CLR     SOFSW    ;CLEAR ERROR COUNTER
392 002316 005037 012574      CLR     RETRY       ;ZERO THE RETRY COUNTER
393 002322 105037 012722      CLRB   FMTDPB+10    ;RESTORE SECTOR
394 002326 013737 012604 012716      MOV     WWC,FMTDPB+4 ;RESTORE WC
395 002334 012737 020300 012720      MOV     $BUFF,FMTDPB+6 ;RESTORE CA
396 002342 005737 012544      TST     MODE        ;'FORMAT' OR 'CHECK' MODE ?
397 002346 001416      BEQ     CKHDR       ;BR IF 'CHECK' MODE
398 002350 112737 000163 012714      WRHDR: MOVB   $163,FMTDPB+2 ;SET WRITE HEADER & DATA COMMAND IN TBL
399 002356 004037 013476      101     JSR     R0,RPO4     ;GO FORMAT A TRACK
400 002362 012712      FMTDPB ;ADRS OF PARAMETERS = TBL
401 002364 000774      BR      10          ;WAIT FOR QUEUE IF FULL
402 002366 005737 012730      201     TST     FMTDPB+16    ;WAIT FOR COMMAND TO COMPLETE
403 002372 001775      BEQ     20          ;BRANCH IF NOT DONE
404 002374 100003      BPL     CKHDR       ;BRANCH IF NO ERROR
405 002376 004737 003716      JSR     PC,ER       ;GO RESOLVE ERROR
406 002402 000762      BR      WRHDR       ;GO TRY AGAIN
407
408                          ;THE CODE SETS UP FOR THE FORMAT CHECK
409
410 002404 112737 000153 012714      CKHDR: MOVB   $153,FMTDPB+2 ;SET WRITE CHECK HEADER & DATA COMMAND IN TBL
411 002412 004037 013476      101     JSR     R0,RPO4     ;GO CHECK THE TRK JUST FORMATTED
412 002416 012712      FMTDPB ;ADRS OF PARAMETERS = TBL
413 002420 000774      BR      10          ;WAIT FOR QUEUE IF FULL
414 002422 005737 012730      201     TST     FMTDPB+16    ;WAIT FOR COMMAND TO COMPLETE
415 002426 001775      BEQ     20          ;BRANCH IF NOT DONE
416 002430 100006      BPL     GOOD        ;BRANCH IF NO ERROR
417 002432 004737 003716      JSR     PC,ER       ;GO RESOLVE ERROR
418 002436 005737 012544      TST     MODE        ;'FORMAT' OR 'CHECK' MODE ?
419 002442 001760      BEQ     CKHDR       ;BR IF 'CHECK' MODE
420 002444 000741      BR      WRHDR       ;ER-REFORMAT SECTOR
421 002446 005737 012542      GOOD:  TST     SOFSW    ;SEE IF RETRY
422 002452 001421      BEQ     LOOPTS     ;BRANCH IF NOT
423 002454 022737 000002 012542      CMP     $2,SOFSW   ;SEE IF LAST RETRY
424 002462 001012      BNE     10          ;BRANCH IF NOT
425 002464 123737 012612 012722      CHPB   MAXSEC,FMTDPB+10 ;SEE IF LAST SECTOR IN TRK
426 002472 001411      BEQ     LOOPTS     ;BRANCH IF SO
427 002474 004737 006260      JSR     PC,SCAWC   ;GO SET UP TBL FOR REMAINING SECTORS
428 002500 005737 012544      TST     MODE        ;'FORMAT' OR 'CHECK' MODE ?
429 002504 001737      BEQ     CKHDR       ;BR IF 'CHECK' MODE
430 002506 000720      BR      WRHDR       ;GO FINISH FORMATTING TRK
431 002510 005237 012542      101     INC     SOFSW    ;RECORD GOOD RETRY
432 002514 000710      BR      WRHDR       ;TRY ONCE MORE

```

```

433 002516 032737 040000 177570 LOOPTS: BIT      #BIT14,#05WR      ;SEE IF LOOP ON CURRENT TRK
434 002524 001401          BEQ      NXTRK          ;BRANCH IF NOT
435 002526 000671          BR       WRHDR2         ;REPEAT SAME TRACK
436 002530 004737 006166          NXTRK: JSR      PC,TRKIST    ;GOOD FORMAT-GO SEE IF DONE-IF NOT, SET UP NEXT
437 002534 000401          BR       ENDCK         ;RETURN HERE IF DONE - GO DO QUICK CK
438 002536 000665          BR       WRHDR2         ;NOT DONE - GO WRITE NEXT TRACK
439
440
441                                     ;THIS CODE MAKES SURE EACH CYLINDER FORMATTED CONTAINS THE
442                                     ;PROPER CYLINDER ADDRESS, THE PROGRAM IS LOOKING FOR
443                                     ;POSSIBLE POSITIONER ERRORS THAT MAY HAVE OCCURED DURING THE FORMAT.
444 002540 005037 012606          ENDCK: CLR      HEDERR      ;CLEAR HEADER CHECK ERROR INDICATOR
445 002544 004737 006016          JSR      PC,SETTBL      ;GO SET UP DRIVER TABLE
446 002550 004737 006320          JSR      PC,SETHDR      ;GO SET UP HEADERS IN CORE
447 002554 013737 012550 012572    MOV      SCYL,TEMP      ;PUT SCYL IN TEMP
448 002562 012737 177776 012716    MOV      #2,FMTDPB+4    ;SET UP WORD COUNT
449 002570 012737 012622 012720    MOV      #RBUF,FMTDPB+6 ;SET UP BUFFER ADRS
450 002576 005037 012722          CLR      FMTDPB+10     ;CLEAR THE SECTOR & TRACK ADDRESS FIELD
451 002602 013737 012550 012724    MOV      SCYL,FMTDPB+12 ;SETUP THE CYLINDER FIELD
452 002610 112737 000173 012714    MOV      #173,FMTDPB+2 ;SET UP READ HEADER & DATA COMMAND
453 002616 004037 013476          HDREAD: JSR      R0,RP04   ;GO READ HEADER
454 002622 012712          FMTDPB      ;ADRS OF PARAMETER TBL
455 002624 000774          BR       HDREAD        ;WAIT IF QUEUE IS FULL
456 002626 005737 012730          18:      TST      FMTDPB+16 ;WAIT FOR COMMAND TO COMPLETE
457 002632 001775          BEQ      18            ;BRANCH IF NOT DONE
458 002634 100003          BPL      HDRCK         ;BRANCH IF DONE
459 002636 004737 003716          JSR      PC,ER         ;GO RESOLVE ERROR
460 002642 000406          BR       HDRCK1        ;BYPASS THE COMPARSION
461 002644 023737 012622 020300    HDRCK:  CMP      RBUF,BUFP ;SEE IF CYL READ EQUALS CYL EXPECTED
462 002652 001402          BEQ      ,+6           ;BR IF CYL CORRECT
463 002654 004737 002706          JSR      PC,CMPER      ;REPORT INCORRECT CYLINDER
464 002660 023737 012546 012572    HDRCK1: CMP      ECYL,TEMP    ;SEE IF LAST CYL
465 002666 001453          BEQ      DONE         ;BRANCH IF ALL DONE
466 002670 005237 012572          INC      TEMP         ;RECORD CYL CHECKED
467 002674 004737 006374          JSR      PC,UPDACY     ;SET UP FOR NEXT CYL
468 002700 005237 012724          INC      FMTDPB+12    ;ADVANCE TO NEXT CYL
469 002704 000744          BR       HDREAD        ;GO READ NEXT HEADER
470
471                                     ;THIS CODE REPORTS THAT THE HEADER READ DOES NOT
472                                     ;COMPARE WITH THE ONE EXPECTED.
473
474 002706 032737 020000 177570    CMPE:  BIT      #BIT13,#08WR ;SKIP TYPEOUT?
475 002714 001037          BNE      18            ;BRANCH IF SO
476 002716 004537 005166          JSR      R5,TYPOUT     ;GO TYPE SOFT HEADER COMPARE ERROR
477 002722 012474          MHD CMP
478 002724 004737 006662          JSR      PC,DRSN       ;GO TYPE DRIVES & SN
479 002730 004537 005166          JSR      R5,TYPOUT     ;GO TYPE EXPECTED
480 002734 012444          MEXPED
481 002736 004737 006744          JSR      PC,CYLTK      ;GO TYPE CYL,TRK EXPECTED
482 002742 004537 005166          JSR      R5,TYPOUT     ;GO TYPE READ
483 002746 012460          MREAD
484 002750 042737 010000 012622    BIC      #10000,RBUF    ;DUMP FMT22
485 002756 113737 012625 012614    MOV      RBUF+3,LOCTRK ;TRACK FROM HEADER
486 002764 013737 012622 012616    MOV      RBUF,LOCCYL   ;CYLINDER FROM HEADER

```

```

487 002772 004737 006760 JSR PC,CYLTRK ;GO TYPE CYL=TRK READ
488 002776 012737 177777 012606 MOV 0=1,HEDERR ;SET THE CYLINDER ERROR INDICATOR
489 003004 005737 177570 TST 00SWR ;HALT?
490 003010 100001 BPL 10 ;BRANCH IF NOT
491 003012 000000 HALT ;ERROR= SOFTWARE HEADER COMPARE ERROR
492 003014 000207 10: RTS PC ;RETURN
493
494 ;END OF FORMAT (OR CHECK) ROUTINE
495
496 003016 005737 012606 DONE: TST HEDERR ;CYLINDER ERROR ?
497 003022 001404 BEQ 10 ;BR IF NOT
498 003024 004537 005166 JSR RS,TYPOUT ;TYPEOUT 'REFORMAT MESSAGE'
499 003030 012332 HBDPMT ;MESSAGE ADDRESS
500 003032 000412 BR 30
501 003034 005737 012544 10: TST MODE ;'FORMAT' OR 'CHECK' MODE ?
502 003040 001404 BEQ 20 ;BR IF 'CHECK' MODE
503 003042 004537 005166 JSR RS,TYPOUT ;GO TYPE MESSAGE 'FORMAT COMPLETE'
504 003046 010752 HFCMPT ;ADRS OF MESSAGE
505 003050 000403 BR 30
506 003052 004537 005166 20: JSR RS,TYPOUT ;TYPE 'END OF CHECK' MESSAGE
507 003056 010777 HCCMPT ;ADDR OF MSG
508 003060 032737 000001 177570 30: BIT 0SW0,SWR ;SEE IF SWR 0 SET
509 003066 001002 BNE ,+6 ;BR IF SET
510 003070 000137 001156 JMP RSTRT ;ASK FOR NEXT DRIVE
511 003074 012706 001100 MOV 0STACK,SP ;RESET STACK POINTER
512 003100 000137 002142 JMP M5 ;DO THE FORMAT OR CHECK AGAIN
513
514 ;THIS CODE SERVICES A CLOCK INTERRUPT EVERY 16MS
515
516 003104 012746 000020 CLOCK: MOV 016,,-(SP) ;PUT MILLISECONDS ON THE STACK
517 003110 004037 016436 JSR R0,RPTNR ;GO REPORT TIME
518 003114 000002 RTI ;RETURN AND CONTINUE
519
520 ;THIS IS THE ENT ERROR HANDLER=SELECTS ADRS OF ERROR HANDLER
521
522
523 003116 010137 003162 ENTEND: MOV R1,SAVR ;SAVE THE REGISTERS
524 003122 010337 003164 MOV R3,SAVR+2 ;DRIVE NUMBER
525 003126 010437 003166 MOV R4,SAVR+4 ;ATA BIT
526 003132 010537 003170 MOV R5,SAVR+6 ;RHCS1 ADDRESS
527 003136 010037 003172 MOV R0,SAVR+10 ;MOL STATE (BIT 4)
528 003142 010237 003174 MOV R2,SAVR+12 ;SAVE UNKNOWN
529 003146 011600 MOV (SP),R0 ;SAVE UNKNOWN
530 003150 014000 MOV -(R0),R0 ;GET ENT PC
531 003152 006300 ASL R0 ;GET ENT CALL INSTR
532 003154 016000 173174 MOV ENTTL=10002(R0),R0 ;TIMES 2
533 003160 000110 JMP (R0) ;DEVELOP ENT ADDRESS
;GO TO ERROR REPORT ROUTINE
534
535 ;SAVE REGS 1,3,4,5 HERE AFTER DRIVER ERRORS 1->5
536
537 003162 000000 SAVR: 0 ;R1
538 003164 000000 0 ;R3
539 003166 000000 0 ;R4
540 003170 000000 0 ;R5

```

```

541 003172 000000      0      ;R0
542 003174 000000      0      ;R2
543
544      ;TABLE OF ERROR HANDLERS=DRIVER LEVEL
545
546 003176 003210      EMTTBL: ER1      ;ILLEGAL SUBSYSTEM INTERRUPT
547 003200 003244      ER2      ;ILLEGAL DRIVE INTERRUPT
548 003202 003354      ER3      ;MASSBUS CONTROL BUS PARITY ERROR
549 003204 003460      ER4      ;CONTROL BUS PARITY ERROR
550 003206 003606      ER5      ;ATTENTION FROM AN UNAVAILABLE DRIVE
551
552      ;HANDLES ILLEGAL SUBSYSTEM INTERRUPT
553
554 003210 032737 020000 177570 ER1:  BIT      0BIT13,00SWR      ;SKIP TYPEOUT?
555 003216 001003      BNE      10      ;BRANCH IF SO
556 003220 004537 005166      JSR      R5,TYPEOUT      ;TYPE ILLEGAL SUBSYSTEM INTERRUPT
557 003224 011023      MER1
558 003226 005737 177570      10:    TST      00SWR      ;HALT?
559 003232 100001      BPL      20      ;BRANCH IF NOT
560 003234 000000      HALT
561 003236 004737 007270      20:    JSR      PC,RESREG      ;ERROR= ILLEGAL SUBSYSTEM INTERRUPT
562 003242 000002      RTI      ;RESTORE R0-R5
563      ;RETURN TO DRIVER
564
565      ;HANDLES ILLEGAL DRIVE INTERRUPT
566 003244 032737 020000 177570 ER2:  BIT      0BIT13,00SWR      ;SKIP TYPEOUT?
567 003252 001031      BNE      10      ;BRANCH IF SO
568 003254 004537 005166      JSR      R5,TYPEOUT      ;TYPE ILLEGAL DRIVE INTERRUPT
569 003260 011065      MER2
570 003262 004537 005166      JSR      R5,TYPEOUT      ;TYPE DRIVE
571 003266 011762      MDRV
572 003270 013703 003162      MOV      SAVR,R3      ;GET DRIVE #
573 003274 004737 006724      JSR      PC,TDRNUM      ;GO TYPE DRIVE #
574 003300 004737 005376      JSR      PC,CRLF      ;SET MARGIN
575 003304 004537 005166      JSR      R5,TYPEOUT      ;TYPE RHAS
576 003310 011771      HAS
577 003312 013703 003170      MOV      SAVR+6,R3      ;GET RHAS
578 003316 004737 006604      JSR      PC,OCTYP      ;GO TYPE IT
579 003322 004737 005376      JSR      PC,CRLF      ;SET UP MARGIN
580 003326 004737 007064      JSR      PC,SETREG      ;GO PREPARE FOR TYPE
581 003332 004737 007150      JSR      PC,RHREGS      ;GO TYPE REGISTERS
582 003336 005737 177570      10:    TST      00SWR      ;HALT?
583 003342 100001      BPL      20      ;BRANCH IF NOT
584 003344 000000      HALT
585 003346 004737 007270      20:    JSR      PC,RESREG      ;ERROR= ILLEGAL DRIVE INTERRUPT
586 003352 000002      RTI      ;RESTORE R0-R5
587      ;RETURN TO DRIVER
588
589      ;HANDLES MASSBUS CONTROL BUS PARITY ERRORS
590 003354 032737 020000 177570 ER3:  BIT      0BIT13,00SWR      ;SKIP TYPEOUT?
591 003362 001027      BNE      10      ;BRANCH IF SO
592 003364 004537 005166      JSR      R5,TYPEOUT      ;TYPE MASSBUS CONTROL BUS PARITY ERROR
593 003370 011123      MER3
594 003372 004737 005376      JSR      PC,CRLF      ;SET MARGIN

```

595	003376	004537	005166			JSR	R5, TYP0UT	;TYPE RD,ADR
596	003402	012000				MRDADR		
597	003404	013703	017062			MOV	RD,ADR,R3	;GET ADRS
598	003410	004737	006604			JSR	PC,OCTYP	;TYPE IT
599	003414	004737	005376			JSR	PC,CRLF	;SET MARGIN
600	003420	004537	005166			JSR	R5, TYP0UT	;TYPE RD,WRD
601	003424	012010				MRDWRD		
602	003426	013703	017064			MOV	RD,WRD,R3	;GET WRD
603	003432	004737	006604			JSR	PC,OCTYP	;TYPE IT
604	003436	004737	005376			JSR	PC,CRLF	;SET MARGIN
605	003442	005737	177570	18:		TST	008WR	;HALT?
606	003446	100001				BPL	20	;BRANCH IF NOT
607	003450	000000				HALT		;ERROR= MASSBUS CONTROL BUS PARITY ERROR
608	003452	004737	007270	20:		JSR	PC,RESREG	;RESTORE R0-R5
609	003456	000002				RTI		;RETURN TO DRIVER
610								
611								
612								;HANDLES CONTROL BUS PARITY ERRORS
613	003460	032737	020000	177570	ER4:	BIT	0BIT13,008WR	;SKIP TYPEOUT?
614	003466	001040				BNE	10	;BRANCH IF 80
615	003470	004537	005166			JSR	R5, TYP0UT	;TYPE CONTROL BUS PARITY ERROR
616	003474	011172				NER4		
617	003476	004737	005376			JSR	PC,CRLF	;SET MARGIN
618	003502	004537	005166			JSR	R5, TYP0UT	;TYPE WRT,AD
619	003506	012020				MWRTAD		
620	003510	013703	017264			MOV	WRT,AD,R3	;GET ADRS
621	003514	004737	006604			JSR	PC,OCTYP	;TYPE IT
622	003520	004737	005376			JSR	PC,CRLF	;SET MARGIN
623	003524	004537	005166			JSR	R5, TYP0UT	;TYPE WRT,WD
624	003530	012030				MWRTWD		
625	003532	013703	017262			MOV	WRT,WD,R3	;GET WRD
626	003536	004737	006604			JSR	PC,OCTYP	;TYPE IT
627	003542	004737	005376			JSR	PC,CRLF	;SET MARGIN
628	003546	004537	005166			JSR	R5, TYP0UT	;TYPE RD,WRD
629	003552	012010				MRDWRD		
630	003554	013703	017064			MOV	RD,WRD,R3	;GET WRD
631	003560	004737	006604			JSR	PC,OCTYP	;TYPE IT
632	003564	004737	005376			JSR	PC,CRLF	;SET MARGIN
633	003570	005737	177570	18:		TST	008WR	;HALT?
634	003574	100001				BPL	20	;BRANCH IF NOT
635	003576	000000				HALT		;ERROR= CONTROL BUS PARITY ERROR
636	003600	004737	007270	20:		JSR	PC,RESREG	;RESTORE R0-R5
637	003604	000002				RTI		;RETURN TO DRIVER
638								
639								;HANDLES ATTENTION FROM AN UNAVAILABLE DRIVE
640								
641	003606	032737	020000	177570	ER5:	BIT	0BIT13,008WR	;SKIP TYPEOUT?
642	003614	001031				BNE	10	;BRANCH IF 80
643	003616	004537	005166			JSR	R5, TYP0UT	;TYPE ATTENTION FROM AN UNAVAILABLE DRIVE
644	003622	011231				NER5		
645	003624	004537	005166			JSR	R5, TYP0UT	;TYPE DRIVE
646	003630	011762				MDRV		
647	003632	013703	003162			MOV	SAVR,R3	;GET DRIVE #
648	003636	004737	006724			JSR	PC,TDRNUM	;TYPE IT

649	003642	004737	005376			JSR	PC,CRLF		;SET MARGIN
650	003646	004537	005166			JSR	R5,TYPEOUT		;TYPE RHAS
651	003652	011771				MAS			
652	003654	013703	003170			MOV	SAVR+6,R3		;GET RHAS
653	003660	004737	006604			JSR	PC,OCTYP		;TYPE IT
654	003664	004737	005376			JSR	PC,CRLF		;SET MARGIN
655	003670	004737	007064			JSR	PC,SETREG		;GO PREPARE FOR TYPE
656	003674	004737	007150			JSR	PC,RHREGS		;GO TYPE REGISTERS
657	003700	005737	177570	101		TST	008WR		;HALT?
658	003704	100001				BPL	20		;BRANCH IF NOT
659	003706	000000				HALT			;ERROR= ATTENTION FROM AN UNAVAILABLE DRIVE
660	003710	004737	007270	201		JSR	PC,RESREG		;RESTORE R0-R5
661	003714	000002				RTI			;RETURN TO DRIVER
662									
663									
664									;THIS CODE DECIDES WHAT TO DO WITH AN ERROR AT THE USER LEVEL
665									
666	003716	032737	000200	012730	ER1	BIT	0BIT7,FMTDPB+16		;DID THE COMMAND TERMINATE
667	003724	001402				BEQ	10		;BRANCH IF FATAL ERROR
668	003726	000137	004002			JMP	ER6		;GO DETERMINE WHAT KIND OF DATA ERROR
669	003732	032737	070000	012730	101	BIT	070000,FMTDPB+16		;SEE WHICH FATAL ERROR
670	003740	001402				BEQ	20		;BRANCH IF A PARITY ERROR
671	003742	000137	004776			JMP	ER7		;GO REPORT DRIVE UNSAFE
672	003746	032737	004000	012730	201	BIT	0BIT11,FMTDPB+16		;CHECK CONDITION OF PARITY ERROR
673	003754	001402				BEQ	30		;BRANCH IF FATAL MASSBUS PARITY ERROR
674	003756	000137	005042			JMP	ER8		;GO REPORT UNCORRECTABLE MASSBUS PARITY ER
675	003762	032737	002000	012730	301	BIT	0BIT10,FMTDPB+16		;CHECK FOR FATAL PARITY ERROR
676	003770	001402				BEQ	40		;BR IF NOT
677	003772	000137	005106			JMP	ER9		;GO REPORT FATAL MASSBUS PARITY ERROR
678	003776	000137	005136		401	JMP	ER10		;MUST BE SOFTWARE TIMEOUT
679									
680									;HANDLES THE DATA ERROR INDICATIONS
681									
682	004002	032737	000070	012730	ER61	BIT	070,FMTDPB+16		;SEE IF A DATA ERROR
683	004010	001402				BEQ	10		;BRANCH IF SO
684	004012	000137	004506			JMP	ER6A		;GO REPORT ERROR OTHER THAN DATA
685	004016	122737	000173	012714	101	CMPB	0173,FMTDPB+2		;SEE IF DOING A READ HEADER
686	004024	001002				BNE	20		;BRANCH IF NOT
687	004026	000137	004742			JMP	ER6D		;GO REPORT ERROR ON READ HEADER
688	004032	122737	000163	012714	201	CMPB	0163,FMTDPB+2		;SEE IF DOING A WRITE HEADER
689	004040	001002				BNE	30		;BRANCH IF NOT
690	004042	000137	004706			JMP	ER6C		;GO REPORT ERROR ON WRITE HEADER
691	004046	005737	012542		301	TST	00FSW		;IS THIS A RETRY?
692	004052	001002				BNE	40		;BRANCH IF SO
693	004054	000137	004140			JMP	ERROR6		;GO REPORT ERROR ON WRITE CHECK HEADER
694	004060	032737	020000	177570	401	BIT	0BIT13,008WR		;SKIP TYPEOUT?
695	004066	001006				BNE	50		;BRANCH IF SO
696	004070	004537	005166			JSR	R5,TYPEOUT		;GO TYPE RETRIES UNSUCCESSFUL
697	004074	012415				MREUNS			
698	004076	004537	005166			JSR	R5,TYPEOUT		;GO TYPE SECTOR NOT ACCEPTABLE
699	004102	012254				MUNCOR			
700	004104	005737	177570		501	TST	008WR		;HALT?
701	004110	100001				BPL	60		;BRANCH IF NOT
702	004112	000000				HALT			;ERROR= SECTOR FAILS TO FORMAT AFTER 1 OR 2 RETRIES

703	004114	022737	000025	012576	68:	CMP	025,SAVSEC	;IS THIS THE LAST SECTOR IN THE TRK?
704	004122	001403				BEQ	78	;BRANCH IF 50
705	004124	004737	006260			JSR	PC,SCANC	;GO SET UP TBL FOR REMAINING SECTORS
706	004130	000207				RTS	PC	;GO TRY NEXT SECTOR
707	004132	005726			78:	TST	(SP)+	;RESTORE SP
708	004134	000137	002516			JMP	LOOPTS	;TRY NEXT TRK
709								
710								
711								;HANDLES A DATA ERROR ON A WRITE CHECK COMMAND
712	004140	005237	012542		ERROR6:	INC	S0FSW	;RECORD DATA ERROR
713	004144	013704	012650			MOV	SAVREG+6,R4	;GO GET RHDA
714	004150	042704	177740			BIC	0177740,R4	;SAVE ONLY THE BAD SECTOR + 1
715	004154	001002				BNE	10	;BRANCH IF NOT ZERO
716	004156	012704	000026			MOV	026,R4	;RESTORE TO LAST SECTOR +1
717	004162	005304			10:	DEC	R4	;ADJUST SECTOR TO THE ONE THAT FAILED
718	004164	010437	012576			MOV	R4,SAVSEC	;SAVE IT
719	004170	032737	020000	177570		BIT	0BIT13,00SWR	;SKIP TYPEOUT?
720	004176	001076				BNE	SETCA	;BRANCH IF 50
721	004200	004537	005166			JSR	R5,TYPEOUT	;GO TYPE WRITE CHECK ERROR
722	004204	011303				NER6		
723	004206	004737	006662			JSR	PC,DRSN	;GO TYPE DR & SN
724	004212	004737	006744			JSR	PC,CYLTK	;GO TYPE CYL & TRK
725	004216	004537	005166			JSR	R5,TYPEOUT	;GO TYPE SECTORS
726	004222	012057				WSTOR		
727	004224	005000				CLR	R0	;CLR TENS DIGIT
728	004226	013703	012576			MOV	SAVSEC,R3	;GET SECTOR #
729	004232	162703	000012		20:	SUB	012,R3	;SEE HOW MANY TENS
730	004236	100402				BMI	30	;BRANCH IF TOO SMALL
731	004240	005200				INC	R0	;RECORD TEN
732	004242	000773				BR	20	;SUBTRACT AGAIN
733	004244	062703	000012		30:	ADD	012,R3	;MAKE SECTOR POSITIVE AGAIN
734	004250	062730	000260			ADD	0260,R0	;SET UP FOR PRINT
735	004254	004737	005204			JSR	PC,TTO	;GO PRINT IT
736	004260	010300				MOV	R3,R0	;MOVE LSD INTO R0
737	004262	062700	000260			ADD	0260,R0	;SET UP FOR PRINT
738	004266	004737	005204			JSR	PC,TTO	;GO PRINT LSD
739	004272	004537	005166			JSR	R5,TYPEOUT	;GO TYPE GOOD DATA
740	004276	012104				MGDATA		
741	004300	013703	012646			MOV	SAVREG+4,R3	;GET ADRS OF FAILURE
742	004304	162703	000002			SUB	02,R3	;BACK IT OFF BY TWO
743	004310	011303				MOV	(R3),R3	;GET THE DATA WORD
744	004312	004737	006604			JSR	PC,OCTYP	;GO TYPE IT
745	004316	004537	005166			JSR	R5,TYPEOUT	;GO TYPE BAD DATA
746	004322	012067				MBDATA		
747	004324	013703	012664			MOV	SAVREG+22,R3	;GET BAD DATA AT RHDB
748	004330	004737	006604			JSR	PC,OCTYP	;GO TYPE IT
749	004334	004537	005166			JSR	R5,TYPEOUT	;GO TYPE RHEC1
750	004340	012121				MRHEC1		
751	004342	013703	012706			MOV	SAVREG+44,R3	;GO GET RHEC1
752	004346	004737	006604			JSR	PC,OCTYP	;GO TYPE IT
753	004352	004537	005166			JSR	R5,TYPEOUT	;GO TYPE RHEC2
754	004356	012132				MRHEC2		
755	004360	013703	012710			MOV	SAVREG+46,R3	;GET RHEC2
756	004364	004737	006604			JSR	PC,OCTYP	;GO TYPE IT

```

757 004370 004737 007150      JSR      PC,RHREGS      ;GO TYPE RM REGS
758
759                               ;THIS CODE COMPUTES WC & CA FOR RETRIES ON WRITE CHECK ERRORS
760
761 004374 012700 020300      SETCA:  MOV      @BUFF,R0      ;GET STARTING ADRS OF BUFFER
762 004400 013701 012576      MOV      SAVSEC,R1      ;GET SECTOR
763 004404 005701      10:    TST      R1              ;SEE IF ZERO
764 004406 001404      BEQ      20              ;BRANCH IF SO
765 004410 062700 001010      ADD      @520,,R0      ;ADVANCE TO NEXT SECTOR
766 004414 005301      DEC      R1              ;KEEP COUNT
767 004416 000772      BR      10              ;GO CHECK NEXT SECTOR
768 004420 010037 012720      20:    MOV      R0,FMTDPB+6    ;SET UP CA OF BAD SECTOR
769 004424 013700 012604      SETWC:  MOV      MWC,R0      ;GET MAX WC
770 004430 013701 012576      MOV      SAVSEC,R1      ;GET SECTOR
771 004434 005701      10:    TST      R1              ;SEE IF ZERO
772 004436 001404      BEQ      20              ;BRANCH IF SO
773 004440 062700 000404      ADD      @260,,R0      ;ADVANCE TO NEXT SECTOR
774 004444 005301      DEC      R1              ;KEEP COUNT
775 004446 000772      BR      10              ;GO CHECK NEXT SECTOR
776 004450 062700 000404      20:    ADD      @260,,R0      ;POINT WC TO NEXT SECTOR
777 004454 010037 012600      MOV      R0,SAVWC      ;SAVE WC OF REMAINING SECTORS
778 004460 012737 177374 012716  MOV      @-260,,FMTDPB+4 ;SET UP WC FOR BAD SECTOR
779 004466 113737 012576 012722  MOV      SAVSEC,FMTDPB+10 ;SET UP SECTOR ADRS IN TBL
780 004474 005737 177570      TST      @05WR          ;HALT?
781 004500 100001      BPL      30              ;BRANCH IF NOT
782 004502 000000      HALT
783 004504 000207      30:    RTS      PC              ;GO REFORMAT FAILING SECTOR
784
785                               ;CHECK FOR READ OR WRITE UNSAFE ERROR
786
787 004506 032737 000020 012730  ER6A:  BIT      @20,FMTDPB+16  ;CHECK FOR AN UNSAFE
788 004514 001422      BEQ      ER6B          ;BRANCH IF NOT AN UNSAFE
789 004516 032737 020000 177570  BIT      @BIT13,@05WR  ;SKIP TYPEOUT?
790 004524 001010      BNE      10              ;BRANCH IF SO
791 004526 004537 005166      JSR      R5,TYPOUT      ;GO TYPE READ OR WRITE UNSAFE ERROR
792 004532 011332      MER6A
793 004534 004737 007322      JSR      PC,TYPALL      ;GO TYPE DATA
794 004540 004537 005166      JSR      R5,TYPOUT      ;GO TYPE REFORMAT PACK
795 004544 012307      MREFMT
796 004546 005737 177570      10:    TST      @05WR          ;HALT?
797 004552 100001      BPL      20              ;BRANCH IF NOT
798 004554 000000      HALT                    ;ERROR= READ OR WRITE UNSAFE OCCURED
799 004556 000137 001156      20:    JMP      RESTRT        ;GO GET NEXT DRIVE
800
801                               ;INDICATE A DRIVE ERROR HAS OCCURED
802
803 004562 005737 012574      ER6B:  TST      RETRY          ;IS THIS A RETRY?
804 004566 001020      BNE      30              ;BRANCH IF SO
805 004570 032737 020000 177570  BIT      @BIT13,@05WR  ;SKIP TYPEOUT?
806 004576 001005      BNE      10              ;BRANCH IF SO
807 004600 004537 005166      JSR      R5,TYPOUT      ;GO TYPE DRIVE ERROR
808 004604 011372      MER6B
809 004606 004737 007322      JSR      PC,TYPALL      ;GO TYPE DATA
810 004612 005237 012574      10:    INC      RETRY          ;RECORD ERROR

```

011	004616	005737	177570			TST	00SWR		;HALT?
012	004622	100001				BPL	28		;BRANCH IF NOT
013	004624	000000				HALT			;ERROR= DRIVE ERROR
014	004626	000207			201	RTS	PC		;TRY AGAIN
015	004630	022737	000002	012574	301	CHP	02,RETRY		;IS THIS THE LAST RETRY
016	004636	001403				BEG	48		;BRANCH IF SO
017	004640	005237	012574			INC	RETRY		;RECORD RETRY
018	004644	000207				RTS	PC		;TRY AGAIN
019	004646	032737	020000	177570	401	BIT	0BIT13,00SWR		;SKIP TYPEOUT?
020	004654	001006				BNE	58		;BRANCH IF SO
021	004656	004537	005166			JSR	R5,TYPEOUT		;GO TYPE RETRIES UNSUCCESSFUL
022	004662	012415				MREUNS			
023	004664	004537	005166			JSR	R5,TYPEOUT		;GO TYPE REFORMAT PACK
024	004670	012307				MREFMT			
025	004672	005737	177570		501	TST	00SWR		;HALT?
026	004676	100001				BPL	68		;BRANCH IF NOT
027	004700	000000				HALT			;ERROR= DRIVE ERROR STILL THERE AFTER TWO RETRIES
028	004702	000137	001156		601	JMP -	RESTRT		;GO ASK FOR DRIVE
029									
030									;HANDLES A DATA ERROR ON THE WRITE HEADER COMMAND
031									
032	004706	032737	020000	177570	ER6C1	BIT	0BIT13,00SWR		;SKIP TYPEOUT?
033	004714	001005				BNE	18		;BRANCH IF SO
034	004716	004537	005166			JSR	R5,TYPEOUT		;GO TYPE WRITE ERROR
035	004722	011413				MER6C			
036	004724	004737	007322			JSR	PC,TYPEALL		;GO TYPE DATA
037	004730	005737	177570		101	TST	00SWR		;HALT?
038	004734	100001				BPL	28		;BRANCH IF NOT
039	004736	000000				HALT			;ERROR= ERROR OCCURED ON WRITE COMMAND
040	004740	000207			201	RTS	PC		;TRY TO WRITE AGAIN
041									
042									;HANDLES A DATA ERROR ON THE READ HEADER COMMAND
043									
044	004742	032737	020000	177570	ER6D1	BIT	0BIT13,00SWR		;SKIP TYPEOUT?
045	004750	001005				BNE	18		;BRANCH IF SO
046	004752	004537	005166			JSR	R5,TYPEOUT		;GO TYPE HEADER COMPARE NOT MADE
047	004756	011434				MER6D			
048	004760	004737	007322			JSR	PC,TYPEALL		;GO TYPE DATA
049	004764	005737	177570		101	TST	00SWR		;HALT?
050	004770	100001				BPL	28		;BRANCH IF NOT
051	004772	000000				HALT			;ERROR= HEADER COMPARE NOT MADE
052	004774	000207			201	RTS	PC		;RETURN
053									
054									;HANDLES THE DRIVE UNSAFE ERRORS
055									
056	004776	032737	020000	177570	ER71	BIT	0BIT13,00SWR		;SKIP TYPEOUT?
057	005004	001010				BNE	18		;BRANCH IF SO
058	005006	004537	005166			JSR	R5,TYPEOUT		;GO TYPE DRIVE UNSAFE
059	005012	011556				MER7			
060	005014	004737	007322			JSR	PC,TYPEALL		;GO TYPE DATA
061	005020	004537	005166			JSR	R5,TYPEOUT		;GO TYPE FORMAT TERMINATED
062	005024	012226				MFTERM			
063	005026	005737	177570		101	TST	00SWR		;HALT?
064	005032	100001				BPL	28		;BRANCH IF NOT

```

865 005034 000000          HALT          ;ERROR= DRIVE UNSAFE ERROR
866 005036 000137 001156    20:  JMP          RESTRT      ;GO GET NEXT DRIVE
867
868                          ;HANDLES UNCORRECTABLE MASSBUS PARITY ERROR
869
870 005042 032737 020000 177570 ER8:  BIT          @BIT13,@@SWR  ;SKIP TYPEOUT?
871 005050 001010          BNE          10          ;BRANCH IF SO
872 005052 004537 005166    JSR          R5,TYPEOUT  ;GO TYPE UNCORRECTABLE MASSBUS PARITY ERROR
873 005056 011606          MER8
874 005060 004537 005166    JSR          R5,TYPEOUT  ;GO TYPE FORMAT TERMINATED
875 005064 012226          MFTERM
876 005066 004737 006744    JSR          PC,CYLTK    ;GO GIVE THEM CYL, TRK=XXX,XX
877 005072 005737 177570    10:  TST          @@SWR      ;HALT?
878 005076 100001          BPL          20          ;BRANCH IF NOT
879 005100 000000          HALT
880 005102 000137 001156    20:  JMP          RESTRT      ;ERROR= UNCORRECTABLE MASSBUS PARITY ERROR
881                          ;GO GET NEXT DRIVE
882
883                          ;HANDLES FATAL MASSBUS PARITY ERROR
884 005106 032737 020000 177570 ER9:  BIT          @BIT13,@@SWR  ;SKIP TYPEOUT?
885 005114 001005          BNE          10          ;BRANCH IF SO
886 005116 004537 005166    JSR          R5,TYPEOUT  ;GO TYPE FATAL MASSBUS PARITY ERROR
887 005122 011656          MER9
888 005124 004737 006744    JSR          PC,CYLTK    ;GO GIVE THEM CYL,TRK=XXX,XX
889 005130 000000          HALT
890 005132 000137 001156    10:  JMP          RESTRT      ;FATAL MASSBUS PARITY ERROR
891                          ;RESTART PROGRAM
892
893
894                          ;HANDLES A SOFTWARE TIMEOUT (HUNG RP04)
895
896 005136 032737 020000 177570 ER10: BIT          @BIT13,@@SWR  ;SKIP TYPEOUT?
897 005144 001005          BNE          10          ;BR IF SO
898 005146 004537 005166    JSR          R5,TYPEOUT  ;GO TYPE 'SOFTWARE TIMEOUT'
899 005152 011716          MER10
900 005154 004737 007322    JSR          PC,TYPEALL  ;ADRS OF MSG
901 005160 000000          HALT
902 005162 000137 001156    10:  JMP          RESTRT      ;GO TYPE DATA
903                          ;SYSTEM HUNG = OPERATION FAILED TO COMPLETE
904                          ;GO RESTART PROGRAM
905
906
907                          ;SBTTL SUPPORT SUBROUTINES
908
909                          ;TYPES OUT CHARACTERS POINTED TO BY THE ADRS IN R1
910 005166 012501          TYP0UT:  MOV          (R5)+,R1    ;SET UP ADRS OF MESSAGE
911 005170 112100    10:  MOVB         (R1)+,R0    ;GET CHARACTER
912 005172 001403          BEQ          20          ;ARE WE DONE TYPING?
913 005174 004737 005204    JSR          PC,TTO      ;GO TYPE IT
914 005200 000773          BR          10          ;GET NEXT CHARACTER
915 005202 000208    20:  RTS          R5        ;EXIT = DONE TYPING
916
917                          ;DOES THE ACTUAL TYPING = TYPES THE NUMBER OF NULL CHARS SPECIFIED IN 'NULCNT'
918

```

```

919 005204 010446          TTO1  MOV     R4,=(SP)      ;SAVE R4
920 005206 013704 012620  MOV     NULCNT,R4    ;SET UP NULL COUNT
921 005212 105737 177564  TTO1:  TSTB   TPS          ;IS PRINTER AVAILABLE?
922 005216 100375          BPL     TTO1         ;NO - KEEP LOOPING
923 005220 110037 177566  MOVB   R0,TPB       ;PRINT CHAR
924 005224 022700 000015  CMP     #15,R0      ;IS IT A RETURN?
925 005230 001002          BNE     TTO2        ;BRANCH IF NOT
926 005232 005304          DEC     R4          ;KEEP TRACK OF NULL CHARS
927 005234 001366          BNE     TTO1        ;BRANCH IF MORE NULL CHARS
928 005236 012604          TTO2:  MOV     (SP)+,R4  ;RESTORE R4
929 005240 000207          RTS     PC          ;EXIT FROM PRINTING
930
931                          ;READS THE KEYBOARD AND CHECKS FOR 'CONTROL C'
932
933 005242 105737 177560  TTI:  TSTB   TKB          ;CHECK FOR CHAR
934 005246 100375          BPL     TTI         ;WAIT FOR CHAR
935 005250 013700 177562  MOV     TKB,R0      ;PUT CHAR IN R0
936 005254 020027 000215  CMP     R0,#215    ;DON'T PRINT A CR
937 005260 001405          BEQ     10         ;SKIP OVER TYPE
938 005262 020027 000212  CMP     R0,#212    ;DON'T PRINT A LF
939 005266 001402          BEQ     10         ;SKIP OVER TYPE
940 005270 004737 005204  JSR     PC,TTO      ;ECHO CHAR TYPED
941 005274 042700 000200  10:   BIC     #200,R0    ;GET RID OF BIT 7
942 005300 022700 000003  CMP     #3,R0      ;IS CHAR IN R0 A CNTRL C?
943 005304 001002          BNE     CTRLR      ;NO - SKIP OVER JUMP
944 005306 000137 001156  JMP     RESTRY      ;YES - GO TO START OF PROGRAM
945 005312 000207          CTRLR: RTS     PC   ;RETURN TO MAIN PROGRAM
946
947                          ;TESTS THAT A INPUT CHARACTER WAS A NUMBER
948
949 005314 020027 000060  NUMTST: CMP     R0,#60   ;IS ASCII SMALLER THAN 60?
950 005320 100410          BMI     TYPERS     ;YES - GO TO TYPERS
951 005322 020027 000072  CMP     R0,#72     ;IS ASCII LARGER THAN 71?
952 005326 100005          BPL     TYPERS     ;YES - GO TO TYPERS
953 005330 042700 177760  BIC     #177760,R0 ;GET RID OF ASCII
954 005334 062716 000002  ADD     #2,(SP) ;ADD TWO TO RETURN ADRS
955 005340 000207          RTS     PC         ;RETURN - GOOD NUMBER
956 005342 004737 005406  TYPERS: JSR     PC,QUES   ;GO TYPE ??
957 005346 000207          RTS     PC         ;RETURN TO MAIN PROGRAM
958
959                          ;PRINTS OUT THE DRIVE SELECTED NUMBER
960
961 005350 013700 012540  UASMI: MOV     USEL,R0  ;GET DRIVE SELECTED
962 005354 062700 000260  ADD     #260,R0    ;MAKE IT ASCII
963 005360 004737 005204  JSR     PC,TTO     ;GO PRINT IT
964 005364 012700 000240  MOV     #240,R0    ;PROVIDE SPACE
965 005370 004737 005204  JSR     PC,TTO     ;GO PRINT IT
966 005374 000207          RTS     PC         ;RETURN
967
968                          ;THIS ROUTINE DOES A CARRIAGE RETURN AND LINE FEED
969
970 005376 004537 005166  CRLF: JSR     R5,TYPOUT ;PRINT LF + CR
971 005402 007574          MCRLF
972 005404 000207          RTS     PC         ;RETURN

```

```

973
974
975
976 005406 004537 005166      QUES: JSR    R5,TYPOUT      ;GO TYPE MESSAGE
977 005412 007567              MQUES  ;ADRS OF MESSAGE
978 005414 000207              RTS     PC                ;RETURN TO MAIN PROGRAM
979
980
981
982 005416 012703 012634      SETPC: MOV    0CPTR,R3      ;PUT CHAR PTR IN R3
983 005422 010204              MOV    R2,R4              ;PUT CHAR COUNT IN R4
984 005424 000207              RTS     PC                ;RETURN
985
986
987
988 005426 004737 005416      CLRBUF: JSR   PC,SETPC     ;GO SET UP POINTER + COUNT
989 005432 005023 101        CLR    (R3)+              ;MOVE 0'S THRU R3 R4 TIMES
990 005434 005304              DEC    R4                  ;KEEP COUNT
991 005436 001375              BNE    10                  ;DONE IF R4 EQ 0
992 005440 004737 005416      JSR   PC,SETPC           ;SET UP POINTER + COUNT AGAIN
993 005444 000207              RTS     PC                ;EXIT
994
995
996
997 005446 012702 000004      GETDAT: MOV   04,R2        ;SET UP MAX # OF DIGITS EXPECTED
998 005452 004737 005426      JSR   PC,CLRBUF         ;GO CLEAR DIGIT BUFFER
999 005456 004737 005242      JSR   PC,TTI            ;GO GET CHAR
1000 005462 022700 000104      CMP   0104,R0           ;IS IT THE DEFAULT CHART
1001 005466 001031              BNE    00                  ;BRANCH IF NOT
1002 005470 005737 012542      TST   SOFSW             ;IS THIS STARTING OR ENDING CYL?
1003 005474 001410              BEQ    10                  ;BRANCH IF ENDING CYL
1004 005476 005037 012550      CLR   SCYL              ;SET STARTING CYL TO 0
1005 005502 005037 012554      CLR   STRK              ;SET STARTING TRK TO 0
1006 005506 004537 005166      JSR   R5,TYPOUT         ;INDICATE DEFAULT VALUES
1007 005512 010272              MCVTK0 ;ADRS OF MSG
1008 005514 000411              BR     20                  ;GO SET UP RETURN
1009 005516 012737 000632 012546 101  MOV   0410,,ECYL        ;SET ENDING CYL TO 410
1010 005524 012737 000022 012552  MOV   010,,ETRK         ;SET ENDING TRK TO 10
1011 005532 004537 005166      JSR   R5,TYPOUT         ;INDICATE DEFAULT VALUES
1012 005536 010324              MCVTK1 ;ADRS OF MSG
1013 005540 062716 000002 201  ADD   02,(SP) ;ADD TWO TO RETURN ADRS
1014 005544 000207              RTS     PC                ;RETURN FROM DEFAULT COND.
1015 005546 004737 005242 301  JSR   PC,TTI            ;GO GET NEXT CHAR FOR CYLINDER
1016 005552 022700 000177 401  CMP   0177,R0           ;IS IT A RUBOUT?
1017 005556 001001              BNE    50                  ;BRANCH IF NOT A RUBOUT
1018 005560 000207              RTS     PC                ;GO RETYPE LINE
1019 005562 022700 000054 501  CMP   054,R0            ;IS IT THE SEPERATOR?
1020 005566 001024              BNE    00                  ;BRANCH IF NOT A COMMA
1021 005570 004737 006120      JSR   PC,DECBIN         ;CONVERT CYL # TO BINARY
1022 005574 023727 012572 000633  CMP   TEMP,0411.        ;SEE IF CYL REQUEST IS GREATER THAN 410
1023 005602 103404              BCS   60                  ;BRANCH IF WITHIN LIMITS
1024 005604 004537 005166      JSR   R5,TYPOUT         ;TYPE LIMIT EXCEEDED
1025 005610 010334              MCLINT ;ADRS OF MESSAGE
1026 005612 000207              RTS     PC                ;RETURN TO RETYPE LINE

```



```

1081 006100 000240      NOP
1082 006102 013737 012554 012562  MOV      STRK,TRKCNT      ;SET UP PARTIAL CYL TRACK COUNT
1083 006110 013737 012556 012560  MOV      TTRKS,TTRKSC    ;SET UP TOTAL TRACKS COUNTER
1084 006116 000207      RTS          PC           ;RETURN FROM SETUP
1085
1086                      ;THIS ROUTINE CONVERTS THE CYL OR TRK ADRS TO OCTAL
1087
1088 006120 014337 012572      DECBIN: MOV      =(R3),TEMP    ;SET UP ONES DIGIT
1089 006124 014300      MOV      =(R3),R0        ;PUT TENS DIGIT IN R0
1090 006126 162700 000001      ADD10: SUB      01,R0     ;SEE HOW MANY 10'S
1091 006132 103404      BCS      MSD            ;GO TRY HUNDREDS
1092 006134 062737 000012 012572  ADD      012,TEMP        ;RECORD 10
1093 006142 000771      BR       ADD10          ;TRY AGAIN
1094 006144 014300      MSD:  MOV      =(R3),R0    ;PUT HUNDREDS DIGIT IN R0
1095 006146 162700 000001      ADD100: SUB     01,R0     ;SEE HOW MANY 100'S
1096 006152 103404      BCS      DTBDNE         ;EXIT IF DONE
1097 006154 062737 000144 012572  ADD      0144,TEMP       ;RECORD 100
1098 006162 000771      BR       ADD100        ;TRY AGAIN
1099 006164 000207      DTBDNE: RTS          PC           ;EXIT
1100
1101                      ;THIS ROUTINE CONTROLS THE DISK ADDRESSING AND TOTAL TRK COUNT
1102                      ;IT IS ENTERED AFTER EVERY TRK OPERATION
1103
1104 006166 005737 012560      TRKTST: TST      TTRKSC    ;SEE IF LAST TRK
1105 006172 001001      BNE      10            ;BRANCH IF NOT LAST TRK
1106 006174 000207      RTS          PC           ;RETURN LAST TRK OF OPERATION
1107 006176 005337 012560      10:  DEC      TTRKSC      ;COUNT TRK JUST COMPLETED
1108 006202 105237 012723      INCB     FMTDPB+11      ;ADVANCE TRK #
1109 006206 022737 000022 012562  CMP      022,TRKCNT     ;IS THIS THE LAST TRK IN CYL?
1110 006214 001403      BEQ      20            ;BRANCH IF 80
1111 006216 005237 012562      INC      TRKCNT        ;COUNT UP TRK WITHIN CYLS
1112 006222 000411      BR       30            ;BRANCH TO SET UP RETURN
1113 006224 005037 012562      20:  CLR      TRKCNT      ;INITIALIZE TRKCNT FOR NEXT CYL
1114 006230 105037 012723      CLRB     FMTDPB+11     ;RESET TRK ADRS TO 0
1115 006234 005237 012724      INC      FMTDPB+12     ;UPDATE CYLINDER #
1116 006240 004737 006374      JSR      PC,UPDACY     ;UPDATE CYLINDER # IN MEM TABLE
1117 006244 000402      BR       40            ;GO TO RETURN SETUP
1118 006246 004737 006420      30:  JSR      PC,UPDATK   ;UPDATE TRACK # IN MEM TABLE
1119 006252 062716 000002      40:  ADD      02,(SP) ;ADD TWO TO RETURN ADRS
1120 006256 000207      RTS          PC           ;EXITS HERE IF NOT LAST TRACK
1121
1122                      ;THIS ROUTINE SETS UP WC & CA FOR REMAINING SECTORS
1123                      ;AFTER A WRITE CHECK ERROR
1124
1125 006260 013737 012600 012716  SCANC:  MOV      0AVWC,FMTDPB+4 ;SET UP WC FOR REMAINING SECTORS
1126 006266 062737 001010 012720  ADD      0520,,FMTDPB+6 ;SET UP CA FOR REMAINING SECTORS
1127 006274 005237 012722      INC      FMTDPB+10     ;ADVANCE TO NEXT SECTOR IN TBL
1128 006300 005037 012542      CLR      00FSW        ;RESET RETRY COUNTER
1129 006304 000207      RTS          PC           ;RETURN TO COMPLETE TRK FORMAT
1130
1131                      ;THIS ROUTINE INITIALIZES THE SECTOR COUNT AND HEADER POINTER
1132                      ;FOR THE HEADER & DATA BUFFER IN MEMORY
1133
1134

```

```

1135 006306 013701 012612      SECCNT: MOV     MAXSEC,R1      ;SET UP SECTOR COUNT
1136 006312 012700 020300      MOV     @BUFP,R0      ;SET UP HEADER POINTER IN R0
1137 006316 000207              RTS     PC              ;RETURN
1138
1139                               ;THIS ROUTINE SETS UP THE CYLINDER ADRS, FORMAT BIT, TRACK AND
1140                               ;SECTOR ADRS IN MEMORY WITH THE STARTING CYL = TRK INFORMATION
1141
1142 006320 004737 006306      SETHDR: JSR     PC,SECCNT    ;SET UP SECTOR COUNT AND HEADER POINTER
1143 006324 013702 012550      MOV     SCYL,R2      ;PUT STARTING CYL # IN R2
1144 006330 013703 012554      MOV     STPK,R3      ;PUT STARTING TRK # IN R3
1145 006334 000303              SWAB    R3            ;JUSTIFY TRACK ADRS
1146 006336 005737 012610      TST     SEC20        ;SEE IF 20 OR 22 SECTOR MODE
1147 006342 001402              BEQ     10            ;BR IF 20 SECTOR MODE
1148 006344 052702 010000      BIS     @10000,R2    ;SET THE 22 SECTOR FORMAT BIT
1149 006350 010220              10:    MOV     R2,(R0)+    ;WRITE IN HEADER AREA OF CORE THE CYL ADRS
1150 006352 010320              MOV     R3,(R0)+    ;WRITE IN HEADER AREA OF CORE THE TRK ADRS
1151 006354 005020              CLR     (R0)+        ;CLR 1ST KEYWORD
1152 006356 005010              CLR     (R0)         ;CLR 2ND KEYWORD
1153 006360 062700 001002      ADD     @514,,R0     ;SET UP FOR NEXT HEADER
1154 006364 005203              INC     R3            ;UPDATE SECTOR ADRS FOR NEXT HEADER
1155 006366 005301              DEC     R1            ;MAINTAIN COUNT OF SECTORS
1156 006370 002367              BGE     10            ;BRANCH IF NOT LAST SECTOR
1157 006372 000207              RTS     PC            ;EXIT - HEADERS ARE LOADED INTO CORE
1158
1159                               ;THIS ROUTINE UPDATES THE CYLINDER ADRS OF THE HEADER WORDS IN CORE
1160
1161 006374 004737 006306      UPDCY: JSR     PC,SECCNT    ;SET UP SECTOR COUNT AND HEADER POINTER
1162 006400 005220              10:    INC     (R0)+        ;INCREMENT FOR NEXT CYLINDER
1163 006402 042710 177400      BIC     @177400,(R0) ;RESET TRK ADRS TO 0
1164 006406 062700 001006      ADD     @518,,R0     ;SET UP FOR NEXT HEADER
1165 006412 005301              DEC     R1            ;COUNT SECTORS
1166 006414 002371              BGE     10            ;BRANCH IF NOT LAST SECTOR
1167 006416 000207              RTS     PC            ;EXIT
1168
1169                               ;THIS ROUTINE UPDATES THE TRACK ADRS OF THE HEADER WORDS IN CORE
1170
1171 006420 004737 006306      UPDATK: JSR     PC,SECCNT    ;SET UP SECTOR COUNT AND HEADER POINTER
1172 006424 005720              TST     (R0)+        ;POINT HEADER POINTER TO TRK = SEC ADRS
1173 006426 062710 000400      10:    ADD     @400,(R0)    ;INDEX TRK ADRS
1174 006432 062700 001010      ADD     @520,,R0     ;SET UP FOR NEXT HEADER
1175 006436 005301              DEC     R1            ;COUNT SECTORS
1176 006440 002372              BGE     10            ;BRANCH IF NOT LAST SECTOR
1177 006442 000207              RTS     PC            ;EXIT
1178
1179                               ;THIS ROUTINE CONVERTS FAILING CYL & TRK ADRS TO DECIMAL FOR TYPE
1180
1181 006444 012702 010741      BINDEC: MOV     @MCYLTK,R2  ;ESTABLISH POINTER FOR CYL & TRK TYPE
1182 006450 013700 012616      MOV     LOCCYL,R0    ;GO GET BINARY CYL ADRS
1183 006454 005001              CLR     R1            ;RESET WEIGHT COUNT
1184 006456 012703 000144      MOV     @144,R3      ;SET UP WEIGHT
1185 006462 162700 000144      10:    SUB     @144,R0      ;SUBTRACT 100 DEC
1186 006466 100402              BMI     20            ;BRANCH IF # NOT LARGE ENOUGH
1187 006470 005201              INC     R1            ;RECORD EACH 100 DEC
1188 006472 000773              BR     10            ;SUB AGAIN

```

```

1189 006474 004737 006572      28: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
1190 006500 005001              CLR    R1             ;RESET WEIGHT COUNT
1191 006502 012703 000012      MOV    #12,R3        ;SET UP WEIGHT
1192 006506 162700 000012      38: SUB    #12,R0      ;SUBTRACT 10 DEC
1193 006512 100402              BMI    48            ;BRANCH IF 0 NOT LARGE ENOUGH
1194 006514 005201              INC    R1            ;RECORD EACH 10 DEC
1195 006516 000773              BR     38            ;SUB AGAIN
1196 006520 004737 006572      48: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
1197 006524 062700 000260      ADD    #260,R0       ;MAKE LSD ASCII DIGIT
1198 006530 110022              MOVB   R0,(R2)+      ;AND STORE IT AWAY
1199 006532 112722 000254      MOVB   #254,(R2)+   ;STORE A COMMA AS A SEPARATOR
1200 006536 113700 012614      MOVB   LOCTRK,R0    ;LOAD R0 WITH CURRENT TRK ADRS
1201 006542 005001              CLR    R1             ;RESET WEIGHT COUNT
1202 006544 162700 000012      58: SUB    #12,R0      ;SUBTRACT 10 DEC
1203 006550 100402              BMI    68            ;BRANCH IF 0 NOT LARGE ENOUGH
1204 006552 005201              INC    R1            ;RECORD EACH 10 DEC
1205 006554 000773              BR     58            ;SUB AGAIN
1206 006556 004737 006572      68: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
1207 006562 062700 000260      ADD    #260,R0       ;MAKE LSD ASCII DIGIT
1208 006566 110022              MOVB   R0,(R2)+      ;STORE LSD OF TRK
1209 006570 000207              RTS     PC            ;GO RETURN AND TYPE BAD CYL & TRK
1210
1211                               ;THIS ROUTINE STORES THE FAILING CYL & TRK ADRS IN MCVLTK
1212
1213 006572 062701 000260      MKDIG: ADD    #260,R1      ;MAKE ASCII DIGIT
1214 006576 110122              MOVB   R1,(R2)+      ;PUSH DIGIT INTO MCVLTK
1215 006600 060300              ADD    R3,R0         ;MAKE 0 POS AGAIN
1216 006602 000207              RTS     PC            ;RETURN TO SET UP NEXT DIGIT
1217
1218                               ;THIS ROUTINE CONVERTS BINARY TO OCTAL FOR TYPE-R3 HAS 0
1219
1220 006604 012702 000006      OCTYP: MOV    #6,R2     ;SET UP DIGIT COUNTER
1221 006610 006103              ROL    R3            ;ADJUST FOR FIRST TYPE
1222 006612 006103              ROL    R3            ;
1223 006614 010300              MOV    R3,R0        ;SEND TO TYPE REG
1224 006616 042700 177776      BIC    #177776,R0   ;CLEAR OUT GARBAGE
1225 006622 000405              BR     CHOP         ;GO MAKE DIGIT
1226 006624 006103              NXD:  ROL    R3      ;ADJUST FOR NEXT TYPE
1227 006626 006103              ROL    R3            ;
1228 006630 006103              ROL    R3            ;
1229 006632 006103              ROL    R3            ;
1230 006634 010300              MOV    R3,R0        ;SEND TO TYPE REG
1231 006636 006003              CHOP: ROR    R3      ;SAVE CARRY
1232 006640 042700 177770      BIC    #177770,R0   ;CLEAR OUT GARBAGE
1233 006644 062700 000260      ADD    #260,R0      ;MAKE A DIGIT
1234 006650 004737 005204      TYOCT: JSR    PC,TTO   ;TYPE IT
1235 006654 005302              DEC    R2            ;IS IT THE LAST?
1236 006656 001362              BNE    NXD          ;BRANCH IF 0
1237 006660 000207              RTS     PC            ;DONE TYPING-RETURN
1238
1239                               ;THIS ROUTINE TYPES THE DRIVES AND S/N
1240
1241 006662 004537 005166      DREN: JSR    R5,TYPOUT ;GO TYPE DRIVES
1242 006666 011762              MDRV

```

1243	006670	013703	012652		MOV	SAVREG+10,R3	;GET DRIVE # FROM RMC52
1244	006674	004737	006724		JSR	PC,TDRNUM	;GO TYPE DRIVE #
1245	006700	004537	005166		JSR	R5,TYPEOUT	;TYPE SN#
1246	006704	012040			MSN		
1247	006706	013703	012672		MOV	SAVREG+30,R3	;GET DRIVE SN
1248	006712	004737	007020		JSR	PC,DECTYP	;GO TYPE IT
1249	006716	004737	005376		JSR	PC,CRLF	;SET UP MARGIN
1250	006722	000207			RTS	PC	;RETURN TO ER HANDLER
1251							
1252							;THIS ROUTINE TYPES THE FAILING DRIVE #
1253							
1254	006724	042703	177770		TDRNUM:	BIC	0177770,R3 ;CLEAR JUNK
1255	006730	010300			MOV	R3,R0	;PLACE DIGIT IN OUTPUT REG
1256	006732	062700	000260		ADD	0260,R0	;MAKE ASCII
1257	006736	004737	005204		JSR	PC,TTO	;GO TYPE IT
1258	006742	000207			RTS	PC	;RETURN
1259							
1260							;THIS ROUTINE TYPES THE CYL,TRK=XXX,XX
1261							
1262	006744	113737	012723	012614	CYLTK:	MOVB	FMTDPB+11,LOCTRK ;TRACK ADDRESS
1263	006752	013737	012724	012616	MOV	FMTDPB+12,LOCCYL ;CYLINDER ADDRESS	
1264	006760	004537	005166		CYLTRK:	JSR	R5,TYPEOUT ;GO TYPE CYL,TRK#
1265	006764	012046			MHCYTK		
1266	006766	004737	006774		JSR	PC,TCYLTK	;GO CONVERT AND TYPE CYL+TRK
1267	006772	000207			RTS	PC	;RETURN TO ER HANDLER
1268							
1269							;THIS ROUTINE ASSEMBLES & TYPES THE FAILING CYL & TRK ADRS
1270							
1271	006774	004737	006444		TCYLTK:	JSR	PC,BINDEC ;GO CONVERT BIN TO DEC
1272	007000	004537	005166		JSR	R5,TYPEOUT	;TYPE FAILING CYL,TRK
1273	007004	010741			MCYLTK		;ADRS OF MESSAGE WITH FAILING CYL-TRK
1274	007006	000207			RTS	PC	;RETURN
1275							
1276							;THIS ROUTINE TYPES TWO SPACES
1277							
1278	007010	004537	005166		TSPACE:	JSR	R5,TYPEOUT ;GO TYPE TWO SPACES
1279	007014	007564			MSPACE		
1280	007016	000207			RTS	PC	;RETURN
1281							
1282							;THIS ROUTINE TYPES OUT DECIMAL # IN R3
1283							
1284	007020	012701	000004		DECTYP:	MOV	04,R1 ;SET UP DIGIT COUNT
1285	007024	006103			NXDD:	ROL	R3 ;SET UP DIGIT
1286	007026	006103			ROL	R3	
1287	007030	006103			ROL	R3	
1288	007032	006103			ROL	R3	
1289	007034	006103			ROL	R3	
1290	007036	010300			MOV	R3,R0	;SET UP TYPE REG
1291	007040	006003			ROR	R3	;SAVE CARRY
1292	007042	042700	177760		BIC	0177760,R0	;GET RID OF JUNK
1293	007046	062700	000260		ADD	0260,R0	;MAKE ASCII
1294	007052	004737	005204		JSR	PC,TTO	;GO TYPE IT
1295	007056	005301			DEC	R1	;SEE IF LAST DIGIT
1296	007060	001361			BNE	NXDD	;BRANCH IF NOT

```

1297 007062 000207          RTS      PC              ;RETURN
1298
1299                          ;SETS UP RH REGS FOR THE 'RHREG' ROUTINE
1300
1301 007064 013737 012732 012654 SETREG: MOV      RPERRS,SAVREG+12      ;SET UP RHDS1
1302 007072 013737 012734 012656          MOV      RPERRS+2,SAVREG+14      ;SET UP RHER1
1303 007100 013737 012736 012702          MOV      RPERRS+4,SAVREG+40      ;SET UP RHER2
1304 007106 013737 012740 012704          MOV      RPERRS+6,SAVREG+42      ;SET UP RHER3
1305 007114 017737 174046 012642          MOV      @SAVR+4,SAVREG          ;SET UP RHCS1
1306 007122 013737 013070 007146          MOV      RPADR,IRHCS2           ;GET RP04 BASE ADRS
1307 007130 062737 000010 007146          ADD      @10,IRHCS2            ;POINT TO RHCS2 ADRS
1308 007136 017737 000004 012652          MOV      @IRHCS2,SAVREG+10      ;GET RHCS2
1309 007144 000207          RTS      PC              ;RETURN
1310 007146 000000          IRHCS2: 0                      ;CONTAINS RHCS2 ADRS
1311
1312                          ;THIS ROUTINE TYPES OUT THE CS1,CS2,DS1,ER1,ER2+ER3 REGISTERS
1313
1314 007150 004537 005166          RHREGS: JSR      RS,TYPOUT        ;GO TYPE THE RH HEADER
1315 007154 012143          HREGHD
1316 007156 013703 012642          MOV      SAVREG,R3              ;GET RHCS1
1317 007162 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1318 007166 004737 007010          JSR      PC,TSPACE             ;SPACE NX TYPE
1319 007172 013703 012652          MOV      SAVREG+10,R3          ;GET RHCS2
1320 007176 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1321 007202 004737 007010          JSR      PC,TSPACE             ;SPACE NX TYPE
1322 007206 013703 012654          MOV      SAVREG+12,R3          ;GET RHDS1
1323 007212 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1324 007216 004737 007010          JSR      PC,TSPACE             ;SPACE NX TYPE
1325 007222 013703 012656          MOV      SAVREG+14,R3          ;GET RHER1
1326 007226 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1327 007232 004737 007010          JSR      PC,TSPACE             ;SPACE NX TYPE
1328 007236 013703 012702          MOV      SAVREG+40,R3          ;GET RHER2
1329 007242 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1330 007246 004737 007010          JSR      PC,TSPACE             ;SPACE NX TYPE
1331 007252 013703 012704          MOV      SAVREG+42,R3          ;GET RHER3
1332 007256 004737 006604          JSR      PC,OCTYP              ;TYPE IT
1333 007262 004737 005376          JSR      PC,CRLF               ;SET MARGIN
1334 007266 000207          RTS      PC              ;RETURN TO ER HANDLER
1335
1336                          ;THIS ROUTINE RESTORES R0-R5 WHEN RETURNING TO THE DRIVER
1337
1338 007270 013701 003162          RESREG: MOV      SAVR,R1          ;RESTORE R0
1339 007274 013703 003164          MOV      SAVR+2,R3            ;RESTORE R3
1340 007300 013704 003166          MOV      SAVR+4,R4            ;RESTORE R4
1341 007304 013705 003170          MOV      SAVR+6,R5            ;RESTORE R5
1342 007310 013700 003172          MOV      SAVR+10,R0           ;RESTORE R0
1343 007314 013702 003174          MOV      SAVR+12,R2           ;RESTORE R2
1344 007320 000207          RTS      PC              ;RETURN TO DRIVER
1345
1346                          ;THIS ROUTINE TYPES THE DRIVE #, SERIAL NUMBER AND
1347                          ;THE CONTENTS OF THE RH11-RP04 REGISTERS
1348
1349 007322 004737 006662          TYPALL: JSR      PC,DRSN            ;GO TYPE DRIVE # AND SN
1350 007326 004737 006744          JSR      PC,CYLTK             ;GO TYPE CYL,TRK=XXX,XX

```

```

1351 007332 004737 007150 JSR PC,RHREGS ;GO TYPE RH REGS
1352 007336 000207 RTS PC ;EXIT
1353
1354 ;ROUTINE TO CHECK FOR KW11=L OR KW11=P CLOCKS
1355
1356 007340 012737 007410 000004 CKCLK1: MOV @CKCLK1,@ERRVEC ;SET UP VECTOR FOR P CLK
1357 007346 005037 000006 CLR @ERRVEC+2 ;NEW PSW
1358 007352 005777 163162 TST @LKCSR ;CHECK FOR KW11=P
1359 007356 012737 003104 000104 MOV @CLOCK,104 ;SET UP KW11=P VECTOR
1360 007364 012737 000300 000106 MOV @300,106 ;PSW = PRI 6
1361 007372 012777 177777 163142 MOV @-1,@LKCSB ;LOAD COUNTER BUFFER
1362 007400 012777 000135 163132 MOV @135,@LKCSR ;SET CLK = CNT UP
1363 007406 000423 BR CKCLK3
1364 007410 062706 000004 CKCLK1: ADD @4,SP ;RESTORE THE STACK POINTER
1365 007414 012737 007452 000004 MOV @CKCLK2,@ERRVEC ;CHANGE ERROR VECTOR
1366 007422 005777 170120 TST @LKS ;LOOK FOR KW11=L
1367 007426 012737 003104 000100 MOV @CLOCK,100 ;SET UP KW11=L VECTOR
1368 007434 012737 000300 000102 MOV @300,102 ;PSW = PRI 6
1369 007442 012777 000100 170076 MOV @100,@LKS ;SET KW11=L INTERRUPT ENABLE
1370 007450 000402 BR CKCLK3
1371 007452 062706 000004 CKCLK2: ADD @4,SP ;RESTORE THE STACK POINTER
1372 007456 012737 000006 000004 CKCLK3: MOV @6,@ERRVEC ;RESTORE THE ERROR VECTOR
1373 007464 000207 RTS PC
1374
1375 007466 005015 050122 032060 HTITLE: ,ASCII <15><12>/RP04 FORMATTER PROGRAM/<15><12>
1376 007474 043040 051117 040515
1377 007502 052124 051105 050040
1378 007510 047522 051107 046501
1379 007516 005015
1380 007520 040515 047111 042504 ,ASCII /MAINDEC-11-DERPL-B/<15><12><12>
1381 007526 026503 030461 042055
1382 007534 051105 046120 041055
1383 007542 005015 000012
1384
1385 007546 005015 051104 053111 MUNIT: ,ASCII <15><12>/DRIVE 0 /
1386 007554 020105 020043 000
1387
1388 007561 075 000060 MDRV: ,ASCII /0/
1389
1390 007564 020040 000 MSPACE: ,ASCII / /
1391
1392 007567 077 006477 000012 MQUES: ,ASCII /??/<15><12>
1393 007574 005015 000 MCRLF: ,ASCII <15><12>
1394
1395 007577 040 042040 044522 MOPFLN: ,ASCII / DRIVE OFFLINE/<15><12>
1396 007604 042526 047440 043106
1397 007612 044514 042516 005015
1398 007620 000
1399
1400 007621 040 051104 053111 MDRNP: ,ASCII / DRIVE NOT PRESENT/<15><12>
1401 007626 020105 047516 020124
1402 007634 051120 051505 047105
1403 007642 006524 000012
1404

```

1405	007646	042040	044522	042526	MNRP04: ,ASCIZ / DRIVE NOT AN RP04/<15><12>
1406	007654	047040	052117	040440	
1407	007662	020116	050122	032060	
1408	007670	005015	000		
1409					
1410	007673	040	051104	053111	MUSDRI: ,ASCIZ / DRIVE UNSAFE/<15><12>
1411	007700	020105	047125	040523	
1412	007706	042506	005015	000	
1413					
1414	007713	040	042523	042514	MSELD: ,ASCIZ / SELECTED/
1415	007720	052103	042105	000	
1416					
1417	007725	015	005012	051120	MNODE: ,ASCIZ <15><12><12>/PROGRAM MODE (C OR F) = /
1418	007732	043517	040522	020115	
1419	007740	047515	042504	024040	
1420	007746	020103	051117	043040	
1421	007754	020051	020075	000	
1422					
1423	007761	040	020075	047506	MFORMT: ,ASCIZ / = FORMAT & VERIFY/
1424	007766	046522	052101	023040	
1425	007774	053040	051105	043111	
1426	010002	000131			
1427					
1428	010004	042510	045503	047440	MHECK: ,ASCIZ /HECK ONLY/
1429	010012	046116	000131		
1430					
1431	010016	051117	040515	020124	MORMAT: ,ASCIZ /ORMAT & VERIFY/
1432	010024	020046	042526	044522	
1433	010032	054506	000		
1434					
1435	010035	015	005012	050117	MBSIZE: ,ASCIZ <15><12><12>/OPERATE IN 22 SECTOR MODE (Y OR N) ? /
1436	010042	051105	052101	020105	
1437	010050	047111	031040	020062	
1438	010056	042523	052103	051117	
1439	010064	046440	042117	020105	
1440	010072	054450	047440	020122	
1441	010100	024516	037440	000040	
1442					
1443	010106	005015	050117	051105	MSEC22: ,ASCIZ <15><12>/OPERATION WILL BE IN 22 SECTOR (16 BIT) MODE/
1444	010114	052101	047511	020116	
1445	010122	044527	046114	041040	
1446	010130	020105	047111	031040	
1447	010136	020062	042523	052103	
1448	010144	051117	024040	033061	
1449	010152	041040	052111	020051	
1450	010160	047515	042504	000	
1451					
1452	010165	015	047412	042520	MSEC20: ,ASCIZ <15><12>/OPERATION WILL BE IN 20 SECTOR (10 BIT) MODE/
1453	010172	040522	044524	047117	
1454	010200	053440	046111	020114	
1455	010206	042502	044440	020116	
1456	010214	030062	051440	041505	
1457	010222	047524	020122	030450	
1458	010230	020070	044502	024524	

1459	010236	046440	042117	000105	
1460					
1461	010244	005015	051412	040524	MMINCT: ,ASCIZ <15><12><12>/STARTING CYL,TRK# /
1462	010252	052122	047111	020107	
1463	010260	054503	026114	051124	
1464	010266	036513	000040		
1465					
1466	010272	030075	030054	000	MCYTK0: ,ASCIZ /#0,0/
1467					
1468	010277	015	042412	042116	MMAXCT: ,ASCIZ <15><12>/ENDING CYL,TRK# /
1469	010304	047111	020107	020040	
1470	010312	054503	026114	051124	
1471	010320	036513	000040		
1472					
1473	010324	032075	030061	030454	MCYTK1: ,ASCIZ /#410,10/
1474	010332	000070			
1475					
1476	010334	037477	041440	046131	MCLINT: ,ASCIZ /?? CYLINDER LIMIT EXCEEDED/<15><12>
1477	010342	047111	042504	020122	
1478	010350	044514	044515	020124	
1479	010356	054105	042503	042105	
1480	010364	042105	005015	000	
1481					
1482	010371	077	020077	051124	MTLINT: ,ASCIZ /?? TRACK LIMIT EXCEEDED/<15><12>
1483	010376	041501	020113	044514	
1484	010404	044515	020124	054105	
1485	010412	042503	042105	042105	
1486	010420	005015	000		
1487					
1488	010423	077	020077	047105	MADRER: ,ASCII /?? ENDING DSK ADRS MUST/<15><12>
1489	010430	044504	043516	042040	
1490	010436	045523	040440	051104	
1491	010444	020123	052515	052123	
1492	010452	005015			
1493	010454	042502	042440	052521	,ASCIZ /BE EQUAL TO OR GREATER THAN STARTING ADRS/<15><12>
1494	010462	046101	052040	020117	
1495	010470	051117	043440	042522	
1496	010476	052101	051105	052040	
1497	010504	040510	020116	052123	
1498	010512	051101	044524	043516	
1499	010520	040440	051104	006523	
1500	010526	000012			
1501					
1502	010530	005015	051412	046105	MHELP: ,ASCII <15><12><12>/SELECT DATA PATTERNI/<15><12>
1503	010536	041505	020124	040504	
1504	010544	040524	050040	052101	
1505	010552	042524	047122	006472	
1506	010560	012			
1507	010561	050	024460	055040	,ASCII / (0) ZERO'S/<15><12>
1508	010566	051105	023517	006523	
1509	010574	012			
1510	010575	050	024461	047440	,ASCII / (1) ONE'S/<15><12>
1511	010602	042516	051447	005015	
1512	010610	031050	020051	047527	,ASCIZ / (2) WORST CASE/<15><12><12>

1513	010616	051522	020124	040503		
1514	010624	042523	005015	000012		
1515						
1516	010632	053475	051117	052123	MPATDI	,ASCIZ /WORST CASE/
1517	010640	041440	051501	000105		
1518						
1519	010646	005015	051412	040524	MSFOU1	,ASCIZ <15><12><12>/STARTING FORMAT ON DRIVE 0/
1520	010654	052122	047111	020107		
1521	010662	047506	046522	052101		
1522	010670	047440	020116	051104		
1523	010676	053111	020105	000043		
1524						
1525	010704	005015	051412	040524	MSCHK1	,ASCIZ <15><12><12>/STARTING CHECK ON DRIVE 0/
1526	010712	052122	047111	020107		
1527	010720	044103	041505	020113		
1528	010726	047117	042040	044522		
1529	010734	042526	021440	000		
1530						
1531	010741	060	030060	030054	MCYLTK1	,ASCIZ /000,00/<15><12>
1532	010746	006460	000012			
1533						
1534	010752	005015	043012	051117	MFCMPT1	,ASCIZ <15><12><12>/FORMAT COMPLETE/<15><12>
1535	010760	040515	020124	047503		
1536	010766	050115	042514	042524		
1537	010774	005015	000			
1538						
1539	010777	015	005012	044103	MCCMPT1	,ASCIZ <15><12><12>/CHECK COMPLETE/<15><12>
1540	011004	041505	020113	047503		
1541	011012	050115	042514	042524		
1542	011020	005015	000			
1543						
1544	011023	015	005012	046111	MER11	,ASCIZ <15><12><12>/ILLEGAL SUBSYSTEM INTERRUPT/<15><12><12>
1545	011030	042514	040507	020114		
1546	011036	052523	051502	051531		
1547	011044	042524	020115	047111		
1548	011052	042524	051122	050125		
1549	011060	006524	005012	000		
1550						
1551	011065	015	005012	046111	MER21	,ASCIZ <15><12><12>/ILLEGAL DRIVE INTERRUPT/<15><12><12>
1552	011072	042514	040507	020114		
1553	011100	051104	053111	020105		
1554	011106	047111	042524	051122		
1555	011114	050125	006524	005012		
1556	011122	000				
1557						
1558	011123	015	005012	040515	MER31	,ASCIZ <15><12><12>/MSSBUS CONTROL BUS PARITY ERROR/<15><12><12>
1559	011130	051523	052502	020123		
1560	011136	047503	052116	047522		
1561	011144	020114	052502	020123		
1562	011152	040520	044522	054524		
1563	011160	042440	051122	051117		
1564	011166	005015	000012			
1565						
1566	011172	005015	041412	047117	MER41	,ASCIZ <15><12><12>/CONTROL BUS PARITY ERROR/<15><12><12>

1567	011200	051124	046117	041040		
1568	011206	051525	050040	051101		
1569	011214	052111	020131	051105		
1570	011222	047522	006522	005012		
1571	011230	000				
1572						
1573	011231	015	005012	052101	HER5:	,ASCIZ <15><12><12>/ATTENTION FROM AN UNAVAILABLE DRIVE/<15><12><12>
1574	011236	042524	052116	047511		
1575	011244	020116	051106	046517		
1576	011252	040440	020116	047125		
1577	011260	053101	044501	040514		
1578	011266	046102	020105	051104		
1579	011274	053111	006505	005012		
1580	011302	000				
1581						
1582	011303	015	005012	051127	HER6:	,ASCIZ <15><12><12>/WRITE CHECK ERROR/<15><12>
1583	011310	052111	020105	044103		
1584	011316	041505	020113	051105		
1585	011324	047522	006522	000012		
1586						
1587	011332	005015	051012	040505	HER6A:	,ASCIZ <15><12><12>/READ OR WRITE UNSAFE ERROR/<15><12>
1588	011340	020104	051117	053440		
1589	011346	044522	042524	052440		
1590	011354	051516	043101	020105		
1591	011362	051105	047522	006522		
1592	011370	000012				
1593						
1594	011372	005015	042012	044522	HER6B:	,ASCIZ <15><12><12>/DRIVE ERROR/<15><12>
1595	011400	042526	042440	051122		
1596	011406	051117	005015	000		
1597						
1598	011413	015	005012	051127	HER6C:	,ASCIZ <15><12><12>/WRITE ERROR/<15><12>
1599	011420	052111	020105	051105		
1600	011426	047522	006522	000012		
1601						
1602	011434	005015	042012	051511	HER6D:	,ASCII <15><12><12>/DISK ERROR TRYING TO PERFORM HEADER COMPARE/<15><12>
1603	011442	020113	051105	047522		
1604	011450	020122	051124	044531		
1605	011456	043516	052040	020117		
1606	011464	042520	043122	051117		
1607	011472	020115	042510	042101		
1608	011500	051105	041440	046517		
1609	011506	040520	042522	005015		
1610	011514	047520	044523	044524		,ASCIZ /POSITIONING CHECK NOT PERFORMED/<15><12>
1611	011522	047117	047111	020107		
1612	011530	044103	041505	020113		
1613	011536	047516	020124	042520		
1614	011544	043122	051117	042515		
1615	011552	006504	000012			
1616						
1617	011556	005015	042012	044522	HER7:	,ASCIZ <15><12><12>/DRIVE UNSAFE ERROR/<15><12>
1618	011564	042526	052440	051516		
1619	011572	043101	020105	051105		
1620	011600	047522	006522	000012		

1621						
1622	011606	005015	052412	041516	MER0:	,ASCIZ <15><12><12>/UNCORRECTABLE MASSBUS PARITY ERROR/<15><12>
1623	011614	051117	042522	052103		
1624	011622	041101	042514	046440		
1625	011630	051501	041123	051525		
1626	011636	050040	051101	052111		
1627	011644	020131	051105	047522		
1628	011652	006522	000012			
1629						
1630	011656	005015	043012	052101	MER9:	,ASCIZ <15><12><12>/FATAL MASSBUS PARITY ERROR/<15><12>
1631	011664	046101	046440	051501		
1632	011672	041123	051525	050040		
1633	011700	051101	052111	020131		
1634	011706	051105	047522	006522		
1635	011714	000012				
1636						
1637	011716	006415	051412	043117	MER10:	,ASCIZ <15><15><12>/SOFTWARE TIMEOUT - SYSTEM HUNG/<15><12>
1638	011724	053524	051101	020105		
1639	011732	044524	042515	052517		
1640	011740	020124	020055	054523		
1641	011746	052123	046505	044040		
1642	011754	047125	006507	000012		
1643						
1644	011762	051104	053111	036505	MDRV:	,ASCIZ /DRIVE# /
1645	011770	000				
1646						
1647	011771	122	040510	036523	MAS:	,ASCIZ /RHS# /
1648	011776	000040				
1649						
1650	012000	042122	040456	051104	MRDADR:	,ASCIZ /RD,ADR# /
1651	012006	000075				
1652						
1653	012010	042122	053456	042122	MRDWRD:	,ASCIZ /RD,WRD# /
1654	012016	000075				
1655						
1656	012020	051127	027124	042101	MWRTAD:	,ASCIZ /WRT,AD# /
1657	012026	000075				
1658						
1659	012030	051127	027124	042127	MWRTWD:	,ASCIZ /WRT,WD# /
1660	012036	000075				
1661						
1662	012040	020040	047123	000075	MSN:	,ASCIZ / SN# /
1663						
1664	012046	054503	026114	051124	MHCYTK:	,ASCIZ /CYL,TRK# /
1665	012054	036513	000			
1666						
1667	012057	123	041505	047524	MSTOR:	,ASCIZ /SECTOR# /
1668	012064	036522	000			
1669						
1670	012067	015	041012	042101	MBDATA:	,ASCIZ <15><12>/BAD DATA# /
1671	012074	020040	040504	040524		
1672	012102	000075				
1673						
1674	012104	005015	047507	042117	MGDATA:	,ASCIZ <15><12>/GOOD DATA# /

1675	012112	042040	052101	036501						
1676	012120	000								
1677										
1678	012121	015	051012	042510	MRHEC1:	,ASCIZ	<15><12>/RHEC1=			
1679	012126	030503	000075							
1680										
1681	012132	005015	044122	041505	MRHEC2:	,ASCIZ	<15><12>/RHEC2=			
1682	012140	036462	000							
1683										
1684	012143	015	020012	044122	MREGND:	,ASCIZ	<15><12>/ RHC81 RHC82 RHD81 RHE81 RHE82 RHE83/<15><12>			
1685	012150	051503	020061	020040						
1686	012156	044122	051503	020062						
1687	012164	020040	044122	051504						
1688	012172	020061	020040	044122						
1689	012200	051105	020061	020040						
1690	012206	044122	051105	020062						
1691	012214	020040	044122	051105						
1692	012222	006463	000012							
1693										
1694	012226	005015	047506	046522	MPTERM:	,ASCIZ	<15><12>/FORMAT TERMINATED/<15><12>			
1695	012234	052101	052040	051105						
1696	012242	044515	040516	042524						
1697	012250	006504	000012							
1698										
1699	012254	026440	051440	041500	MUNCOR:	,ASCIZ	/ = SECTOR NOT ACCEPTABLE/<15><12>			
1700	012262	047524	020122	047516						
1701	012270	020124	041501	042503						
1702	012276	052120	041101	042514						
1703	012304	005015	000							
1704										
1705	012307	015	005012	042522	MREFMT:	,ASCIZ	<15><12><12>/REFORMAT PACK/<15><12>			
1706	012314	047506	046522	052101						
1707	012322	050040	041501	006513						
1708	012330	000012								
1709										
1710	012332	005015	050012	051517	MDDFMT:	,ASCIZ	<15><12><12>/POSITIONING ERROR DURING FORMAT = REFORMAT PACK/			
1711	012340	052111	047511	044516						
1712	012346	043516	042440	051122						
1713	012354	051117	042040	051125						
1714	012362	047111	020107	047506						
1715	012370	046522	052101	026440						
1716	012376	051040	043105	051117						
1717	012404	040515	020124	040520						
1718	012412	045503	000							
1719										
1720	012415	015	051012	052105	MREUNS:	,ASCIZ	<15><12>/RETRIES UNSUCCESSFUL/			
1721	012422	044522	051505	052440						
1722	012430	051516	041525	042503						
1723	012436	051523	052506	000114						
1724										
1725	012444	054105	042520	052103	MEXPED:	,ASCIZ	/EXPECTED = /			
1726	012452	042105	026440	000040						
1727										
1728	012460	042522	042101	020040	MREAD:	,ASCIZ	/READ = /			

1729 012466 020040 026440 000040
 1730
 1731 012474 005015 051412 043117
 1732 012502 053524 051101 020105
 1733 012510 042510 042101 051105
 1734 012516 041440 046517 040520
 1735 012524 042522 042440 051122
 1736 012532 051117 005015 000

;LISTED BELOW ARE LOCATIONS CONTAINING
 ;PARAMETERS USED BY THE FORMAT OPERATION

1743 012540

.EVEN

1745 012540 000000
 1746 012542 000000
 1747 012544 000000
 1748 012546 000000
 1749 012550 000000
 1750 012552 000000
 1751 012554 000000
 1752 012556 000000
 1753 012560 000000
 1754 012562 000000
 1755 012564 000000
 1756 012566 000000
 1757 012570 000000
 1758 012572 000000
 1759 012574 000000
 1760 012576 000000
 1761 012600 000000
 1762 012602 000000
 1763 012604 000000
 1764 012606 000000
 1765 012610 000000

USEL: 0
 SOFSW: 0
 MODE: 0
 ECTL: 0
 SCYL: 0
 ETRK: 0
 STRK: 0
 TTRKS: 0
 TTRKSC: 0
 TRKCNT: 0
 PSEL: 0
 PATA: 0
 PATB: 0
 TEMP: 0
 RETRY: 0
 SAVSEC: 0
 SAVNC: 0
 NC: 0
 NWC: 0
 HEDERR: 0
 SEC20: 0

;CONTAINS # OF UNIT SELECTED
 ;CONTENTS ARE FOR SOFTWARE DECISIONS
 ;'FORMAT & VERIFY' OR 'CHECK' MODE INDICATOR
 ;ENDING CILINDER
 ;STARTING CYLINDER
 ;ENDING TRACK
 ;STARTING TRACK
 ;TOTAL # OF TRACKS TO BE FORMATTED
 ;TOTAL # OF TRACKS COUNTER
 ;COUNTS TRKS FROM 0-10 PER CYL
 ;CONTAINS PATTERN SELECTED
 ;PATTERN TO BE WRITTEN ON DISK
 ;DITTO
 ;UTILITY STORAGE
 ;MAINTAINS # OF RETRIES MADE
 ;CONTAINS LAST BAD SECTOR ON FORMAT
 ;CONTAINS NC FOR REMAINING SECTORS ON ERROR
 ;20 OR 22 SECTOR TRACK SIZE (IN WORDS)
 ;2'S COMPLEMENT OF 'NC'
 ;POSITIONING ERROR DURING FORMAT INDICATOR
 ;20 OR 22 SECTOR MODE INDICATOR
 ;0 = 20 SECTOR MODE
 ;1'S = 22 SECTOR MODE
 ;MAXIMUM SECTOR ADDRESS (FOR EITHER 20 OR 22 SECTOR
 ;FORMAT)
 ;TEMP STORAGE FOR TRACK CONVERSION FOR TYPEOUT
 ;TEMP STORAGE FOR CYL CONVERSION FOR TYPEOUT
 ;NULL (FILLER) CHARACTER COUNT FOR TYPEOUTS
 ;THIS LOCATION CONTAINS CYL ADRS ON HEADER CK
 ;THIS LOCATION CONTAINS TRK-SEC ADRS ON HEADER CK
 ;LEAVE SPACE FOR LEADING ZEROS
 ;TTY INPUT BUFFER
 ;MOVE POINTER TO END OF BUFFER

1766
 1767
 1768 012612 000000
 1769
 1770 012614 000000
 1771 012616 000000
 1772 012620 000001
 1773
 1774 012622 000000
 1775 012624 000000
 1776
 1777 012626 000000
 1778 012630 000000
 1779 012632 000000
 1780 012634 000000
 1781 012642
 1782

MAXSEC: 0
 LOCTRK: 0
 LOCCYL: 0
 NULCNT: 1
 RBUF: 0
 CPTR3: 0
 CPTR2: 0
 CPTR1: 0
 CPTR: 0

.B,+4

```

1783 012642 000000 SAVREG: 0 ;RHCS1
1784 012644 000000 0 ;RHWC
1785 012646 000000 0 ;RHBA
1786 012650 000000 0 ;RHDA
1787 012652 000000 0 ;RHCS2
1788 012654 000000 0 ;RHDS1
1789 012656 000000 0 ;RHER1
1790 012660 000000 0 ;RHAS
1791 012662 000000 0 ;RHLA
1792 012664 000000 0 ;RHDB
1793 012666 000000 0 ;RHMR
1794 012670 000000 0 ;RHDT
1795 012672 000000 0 ;RHSM
1796 012674 000000 0 ;RHOF
1797 012676 000000 0 ;RHCA
1798 012700 000000 0 ;RHCC
1799 012702 000000 0 ;RHER2
1800 012704 000000 0 ;RHER3
1801 012706 000000 0 ;RHEC1
1802 012710 000000 0 ;RHEC2
1803
1804
1805
1806 012712 000 FMTDPB: ;BYTE 0 ;DRIVE NUMBER
1807 012713 000 ;BYTE 0 ;OFFSET VALUE OR FMT22,ECI, AND HCI
1808 012714 000 ;BYTE 0 ;COMMAND
1809 012715 000 ;BYTE 0 ;PSEL AND A17 AND A16
1810 012716 000000 ;WORD 0 ;WORD COUNT (NEG)
1811 012720 000000 ;WORD 0 ;BUFFER ADDRESS
1812 012722 000 ;BYTE 0 ;SECTOR ADDRESS
1813 012723 000 ;BYTE 0 ;TRACK ADDRESS
1814 012724 000000 ;WORD 0 ;CYLINDER ADDRESS
1815 012726 000000 ;WORD 0 ;ERROR TABLE POINTER
1816 012730 000000 ;WORD 0 ;STATUS-ERROR INDICATOR
1817 ;BIT 15 = 1: ERROR OCCURED
1818 ;BIT 07 = 1: DONE
1819 ;BIT 14-16 AND BIT 06-03
1820 ;INDICATE TYPE OF ERROR
1821
1822
1823
1824
1825
1826
1827
1828 ;SBTTL RH11/RPS4 DRIVER (REV. 0.9)
1829
1830 ;COPYRIGHT (C) 1974
1831 ;DIGITAL EQUIPMENT CORP.
1832 ;MAYNARD, MA 01754
1833 ;AUTHOR: JIM LACEY
1834
1835
1836 ;STORAGE FOR RHDS1, RHER1, RHER2, AND RHER3 ON AN ERROR "2" OR "5"

```

1837
1838
1839
1840
1841
1842 012732 000000
1843 012734 000000
1844 012736 000000
1845 012740 000000
1846
1847
1848
1849
1850
1851
1852 012742 000
1853 012743 000
1854 012744 000
1855 012745 000
1856 012746 000
1857 012747 000
1858 012750 000
1859 012751 000
1860
1861
1862
1863
1864
1865
1866 012752 000
1867 012753 000
1868 012754 000
1869 012755 000
1870 012756 000
1871 012757 000
1872 012760 000
1873 012761 000
1874
1875
1876
1877
1878
1879 012762 000000
1880 012764 000000
1881 012766 000000
1882 012770 000000
1883 012772 000000
1884 012774 000000
1885 012776 000000
1886 013000 000000
1887
1888
1889
1890

;RPEPRS = RHDS1
;RPEPRS+2 = RHER1
;RPEPRS+4 = RHER2
;RPEPRS+6 = RHER3

RPEPRS: .WORD 0
.WORD 0
.WORD 0
.WORD 0

;TABLE OF DRIVE ACTIVE INDICATORS (DRVACT=8 BYTES)
;DRVACT=0 IMPLIES DRIVE IS IDLE
;DRVACT>0 IMPLIES DRIVE IS ACTIVE WITH A COMMAND
;DRVACT<0 IMPLIES DRIVE IS ACTIVE WITH AN ERROR RECOVERY OPERATION

DRVACT: .BYTE 0 ;DRIVE 0
.BYTE 0 ;DRIVE 1
.BYTE 0 ;DRIVE 2
.BYTE 0 ;DRIVE 3
.BYTE 0 ;DRIVE 4
.BYTE 0 ;DRIVE 5
.BYTE 0 ;DRIVE 6
.BYTE 0 ;DRIVE 7

;TABLE OF DRIVE STATUS INDICATORS (DRVSTA=8 BYTES)
;DRVSTA=0 IMPLIES DRIVE IS OFFLINE
;DRVSTA>0 IMPLIES DRIVE IS ONLINE
;DRVSTA<0 IMPLIES DRIVE IS UNSAFE OR NONEXISTENT

DRVSTA: .BYTE 0 ;DRIVE 0
.BYTE 0 ;DRIVE 1
.BYTE 0 ;DRIVE 2
.BYTE 0 ;DRIVE 3
.BYTE 0 ;DRIVE 4
.BYTE 0 ;DRIVE 5
.BYTE 0 ;DRIVE 6
.BYTE 0 ;DRIVE 7

;TABLE OF DRIVE TYPES (DRV TYP=8 WORDS)
;DRV TYP WILL CONTAIN THE DRIVE TYPE OF ALL ONLINE, OFFLINE, AND
;UNSAFE DRIVES, IF A DRIVE IS NONEXISTENT DRV TYP WILL BE ZERO.

DRV TYP: .WORD 0 ;DRIVE 0
.WORD 0 ;DRIVE 1
.WORD 0 ;DRIVE 2
.WORD 0 ;DRIVE 3
.WORD 0 ;DRIVE 4
.WORD 0 ;DRIVE 5
.WORD 0 ;DRIVE 6
.WORD 0 ;DRIVE 7

;TRANSFER WAIT FLAG (TRNSWT=1 WORD)
;THIS IS A ONE WORD QUEUE, IT WILL CONTAIN THE ADDRESS OF
;"DPB" OF THE I/O OPERATION,

```

1891
1892 013002 000000 TRNSWT: ,WORD 0
1893
1894 ;*SEARCH WAIT KEYS (SRCHWT=1 WORD)
1895 ;*THIS IS A ONE WORD QUEUE THAT WILL CONTAIN A KEY FOR EACH OF
1896 ;*THE DRIVES THAT ARE PERFORMING A SEARCH COMMAND FOR THE I/O
1897 ;*REQUEST THAT IS AT THE TOP OF THEIR REQUEST QUEUE,
1898 ;*EACH DRIVE IS ASSIGNED ONE BIT, STARTING AT BIT00 FOR DRIVE 0,
1899
1900 013004 000000 SRCHWT: ,WORD 0
1901
1902 ;*RP04 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
1903 ;*ACTDRV=0 IMPLIES DRIVER IS INACTIVE
1904 ;*ACTDRV>0 IMPLIES DRIVER IS ACTIVE
1905
1906 013006 000 ACTDRV: ,BYTE 0
1907
1908 ;*SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
1909 ;*ACTSTR=0 IMPLIES SOFTWARE TIMER ROUTINE IS INACTIVE
1910 ;*ACTSTR>0 IMPLIES SOFTWARE TIMER ROUTINE IS ACTIVE
1911
1912 013007 000 ACTSTR: ,BYTE 0
1913
1914 ;*UNLOAD FLAG (ULDFLG=8 BYTES)
1915 ;*ULDFLG=0 IMPLIES NO UNLOAD COMMAND
1916 ;*ULDFLG>0 IMPLIES UNLOAD COMMAND IN PROGRESS
1917 ;*ULDFLG<0 IMPLIES UNLOAD COMMAND IN WAIT QUEUE
1918
1919 013010 000 ULDFLG: ,BYTE 0 ;DRIVE 0
1920 013011 000 ,BYTE 0 ;DRIVE 1
1921 013012 000 ,BYTE 0 ;DRIVE 2
1922 013013 000 ,BYTE 0 ;DRIVE 3
1923 013014 000 ,BYTE 0 ;DRIVE 4
1924 013015 000 ,BYTE 0 ;DRIVE 5
1925 013016 000 ,BYTE 0 ;DRIVE 6
1926 013017 000 ,BYTE 0 ;DRIVE 7
1927
1928 ;*LOOK AHEAD COUNT (LACNT=8 BYTES)
1929 ;*LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED
1930
1931 013020 000 LACNT: ,BYTE 0 ;DRIVE 0
1932 013021 000 ,BYTE 0 ;DRIVE 1
1933 013022 000 ,BYTE 0 ;DRIVE 2
1934 013023 000 ,BYTE 0 ;DRIVE 3
1935 013024 000 ,BYTE 0 ;DRIVE 4
1936 013025 000 ,BYTE 0 ;DRIVE 5
1937 013026 000 ,BYTE 0 ;DRIVE 6
1938 013027 000 ,BYTE 0 ;DRIVE 7
1939
1940 ;*SAVE REGISTERS FLAG (SAVEFG =1 WORD)
1941 ;*SAVEFG <0 IMPLIES SAVE THE RH11/RP04 REGISTERS WHEN THE
1942 ;*OPERATION IS COMPLETED AS PER (DPB+14),
1943 ;*SAVEFG=0 IMPLIES SAVE THE RH11/RP04 REGISTERS, AS PER
1944 ;*(DPB+14), AFTER AN ERROR,

```

```

1945
1946 013030 000000 SAVEFG: ,WORD 0
1947
1948 ;*SEEK FLAG (SEEKFG=1 WORD)
1949 ;*SEEKFG=0 IMPLIES WHEN THE DISK ADDRESS ISN'T IN THE WINDOW
1950 ;*FOR A DATA TRANSFER START A SEARCH COMMAND
1951 ;*SEEKFG<0 IMPLIES DATA TRANSFER WILL DO IMPLIED SEEKS,
1952 ;*DISREGARD THE WINDOW
1953
1954 013032 000000 SEEKFG: ,WORD 0
1955
1956 ;*TIMEOUT TABLE (TIMER=8 WORDS)
1957 ;*THIS TABLE CONTAINS THE TIME ALLOWED FOR AN OPERATION
1958
1959 013034 177777 TIMER: ,WORD -1 ;DRIVE 0
1960 013036 177777 ,WORD -1 ;DRIVE 1
1961 013040 177777 ,WORD -1 ;DRIVE 2
1962 013042 177777 ,WORD -1 ;DRIVE 3
1963 013044 177777 ,WORD -1 ;DRIVE 4
1964 013046 177777 ,WORD -1 ;DRIVE 5
1965 013050 177777 ,WORD -1 ;DRIVE 6
1966 013052 177777 ,WORD -1 ;DRIVE 7
1967
1968 ;*DATA TRANSFER UNDERWAY INDICATOR (DTUN=1 WORD)
1969 ;*DTUN<0 IMPLIES NO DATA TRANSFER UNDERWAY
1970 ;*DTUN=N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
1971
1972 013054 177777 DTUN: ,WORD -1
1973
1974 ;*ATTENTION BITS TABLE (ATABIT=8 BYTES)
1975 ;*THIS TABLE CONTAINS THE CORRESPONDING BIT TO EACH DRIVES
1976 ;*ATTENTION BIT
1977
1978 013056 001 ATABIT: ,BYTE 1 ;DRIVE 0
1979 013057 002 ,BYTE 2 ;DRIVE 1
1980 013060 004 ,BYTE 4 ;DRIVE 2
1981 013061 010 ,BYTE 10 ;DRIVE 3
1982 013062 020 ,BYTE 20 ;DRIVE 4
1983 013063 040 ,BYTE 40 ;DRIVE 5
1984 013064 100 ,BYTE 100 ;DRIVE 6
1985 013065 200 ,BYTE 200 ;DRIVE 7
1986
1987 ;*RP04 TO RH11 "MASS CONTROL BUS PARITY ERRORS" (MCPE) ALLOWED BEFORE
1988 ;*CALLING IT FATAL (MCPENX=1 WORD)
1989
1990 013066 000003 MCPENX: ,WORD 3
1991
1992 ;*STORAGE FOR RPADR (THE FIRST ADDRESS (776700) OF THE RH11/RP04),
1993 ;*RPVEC (THE VECTOR ADDRESS (254)), AND RPVEC+2 (THE BR LEVEL (5)).
1994 013070 176700 RPADR: ,WORD 176700
1995 013072 000254 000240 RPVEC: ,WORD 254,5*32,
1996
1997 ;*MAXIMUM NUMBER OF LOOK AHEADS ALLOWED IS 4 (MXLACT=1 WORD)
1998 013076 000004 MXLACT: ,WORD 4

```

```

1999 ;*MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)
2000 013100 001000 MXDLTA: ,WORD 8,*64,
2001 ;*MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)
2002 013102 000200 MNDLTA: ,WORD 2*64,
2003 ;*MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWNDW=1 WORD)
2004 013104 000005 MXWNDW: ,WORD 5
2005
2006 ;*DEFINITIONS OF THE RH11/RP04 ADDRESS INDEXES
2007
2008 000000 RHCS1=0 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG, 00)
2009 000002 RHWC=2 ;WORD COUNT REGISTER (NOT A DRIVE REG)
2010 000004 RHBA=4 ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
2011 000006 RND=6 ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG, 05)
2012 000010 RHCS2=10 ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
2013 000012 RHDS1=12 ;DRIVE STATUS REGISTER (DRIVE REG #1)
2014 000014 RHER1=14 ;ERROR REGISTER #1 (DRIVE REG, 02)
2015 000016 RHAS=16 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG, 04)
2016 000020 RNLA=20 ;LOOK AHEAD REGISTER (DRIVE REG, 07)
2017 000022 RHDB=22 ;DATA BUFFER REGISTER (NOT A DRIVE REG,)
2018 000024 RHR=24 ;MAINTAINABILITY REGISTER (DRIVE REG, 03)
2019 000026 RNDT=26 ;DRIVE TYPE REGISTER (DRIVE REG, 06)
2020 000030 RSN=30 ;SERIAL NUMBER REGISTER (DRIVE REG, 10)
2021 000032 RNOF=32 ;OFFSET REGISTER (DRIVE REG, 11)
2022 000034 RHCA=34 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG, 12)
2023 000036 RHCC=36 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG, 13)
2024 000040 RHER2=40 ;ERROR REGISTER #2 (DRIVE REG, 14)
2025 000042 RHER3=42 ;ERROR REGISTER #3 (DRIVE REG, 15)
2026 000044 RHEC1=44 ;ECC POSITION REGISTER (DRIVE REG, 16)
2027 000046 RHEC2=46 ;ECC PATTERN REGISTER (DRIVE REG, 17)
2028
2029 ;*RH11/RP04 DRIVER INIT, CODE
2030 ;*THIS ROUTINE WILL DETERMINE WHICH RP04 DRIVES ARE
2031 ;*AVAILABLE FOR TESTING AND SET THE DRVSTA INDICATOR
2032 ;*TO THE PROPER STATE FOR EACH DRIVE,
2033 ;*NOTE: THIS ROUTINE CALLS DRVINT
2034 ;
2035 ;*CALL
2036 ;
2037 ;* JSR PC,RPINIT
2038 ;* RETURN
2039 ;
2040 013106 013746 177776 RPINIT: MOV 00PS,=(SP) ;SAVE PROCESSOR STATUS
2041 013112 013737 013074 177776 MOV RPVEC+2,00PS ;SET PROCESSOR STATUS TO RH11/RP04 LEVEL
2042 013120 004037 020236 JSR R0,SAVR15 ;GO SAVE R1-R5
2043 013124 004737 017774 JSR PC,CLRQUE ;CLEAR ALL REQUEST QUEUES
2044 013130 012701 012732 MOV 0RPERRS,R1 ;FIRST ADDRESS TO BE CLEARED
2045 013134 012702 013032 MOV 0SEEKFG,R2 ;LAST ADDRESS TO BE CLEARED
2046 013140 005021 101 CLR (R1)+ ;CLEAR
2047 013142 020102 101 CMP R1,R2 ;ARE WE DONE?
2048 013144 101775 101 BLOS 10 ;BRANCH IF NO
2049 013146 012702 013054 MOV 0DTUN,R2 ;LAST ADDRESS
2050 013152 012721 177777 201 MOV 0=1,(R1)+ ;INITIALIZE
2051 013156 020102 101 CMP R1,R2 ;DONE?
2052 013160 101774 101 BLOS 20 ;LOOP IF NO

```

```

2053 013162 005001          CLR      R1          ;DRIVE NUMBER WILL BE IN R1
2054 013164 013704 013070  MOV      RPADR,R4    ;FIRST ADDRESS OF RH11/RP04
2055 013170 012764 000040 000010  MOV      @BIT05,RHCS2(R4) ;MASSBUS INIT.
2056 013176 004037 013250 38:    JSR      R0,DRVINT   ;GO INIT, A DRIVE
2057 013202 000417          BR       78
2058 013204 005201          INC      R1          ;MOVE TO THE NEXT DRIVE
2059 013206 042701 177770 48:    BIC      @C7,R1     ;DON'T LET THE DRIVE NUMBER GET TO BIG
2060 013212 001371          BNE     38          ;BRANCH IF MORE DRIVES TO INIT.
2061 013214 013703 013072          MOV      RPVEC,R3    ;SETUP THE RH11/RP04 VECTOR
2062 013220 012723 015260          MOV      @ISR,(R3)+
2063 013224 013713 013074          MOV      RPVEC+2,(R3)
2064 013230 004037 020256          JSR      R0,GETR15   ;RESTORE R1-R5
2065 013234 012637 177776          MOV      (SP)+,@0PS  ;RESTORE THE PROCESSOR STATUS
2066 013240 000207          RTS     PC          ;BYE-BYE
2067 013242 105061 012752 78:    CLRB    DRVSTA(R1)  ;SET THE DRIVE STATUS TO OFFLINE
2068 013246 000756          BR      48          ;GO DO THE NEXT DRIVE
2069
2070          ;*DRIVE INIT, ROUTINE
2071          ;*THIS ROUTINE DETERMINES IF A DRIVE EXIST AND IF IT IS
2072          ;*AN RP04, IF IT IS, A "READ-IN PRESET" IS ISSUED AND FMT22
2073          ;*IS SET TO A "1", THEN MOL, DPR, DRY, AND VV ARE CHECKED TO
2074          ;*INSURE THEY ARE ALL ON A "1", AND DEPENDING ON THEIR STATE,
2075          ;*DRVSTA IS SET TO THE PROPER CONDITION,
2076          ;*CALL
2077          ;*
2078          ;*
2079          ;*
2080          ;*
2081          ;*
2082          ;*
2083 013250 004037 020236  DRVINT: JSR      R0,SAVR15   ;SAVE R1-R5
2084 013254 010164 000010          MOV      R1,RHCS2(R4) ;SELECT A DRIVE
2085 013260 010103          MOV      R1,R3       ;COPY THE DRIVE NUMBER INTO
2086 013262 006303          ASL     R3           ;R3 AND POSITION IT FOR TABLE INDEXING
2087 013264 112761 177777 012752  MOVB    @-1,DRVSTA(R1) ;START DRIVE STATUS AS NONEXISTENT
2088 013272 005063 012762          CLR     DRVSTYP(R3)  ;CLEAR THE DRIVE TYPE INDICATOR
2089 013276 112777 000111 177564  MOVB    @111,@RPADR  ;DO A "DRIVE CLEAR" COMMAND
2090 013304 032764 010000 000010  BIT     @BIT12,RHCS2(R4) ;NONEXISTENT DRIVE?
2091 013312 001403          BEQ    18           ;NO--BRANCH
2092 013314 004737 017430          JSR     PC,SET,IE   ;GO SET "IE" WITHOUT A "TRE"
2093 013320 000462          BR     68           ;LEAVE THIS ROUTINE
2094 013322 004037 017036 18:    JSR     R0,RD,RP    ;READ THE DRIVE TYPE REG.
2095 013326 000026          RMDT   78
2096 013330 013470          78
2097 013332 012605          MOV     (SP)+,R5    ;ERROR RETURN ADDRESS
2098 013334 010563 012762          MOV     R5,DRVSTYP(R3) ;PUT DRIVE TYPE IN R5
2099 013340 022705 020020          CMP     @20020,R5   ;SAVE THE DRIVE TYPE
2100 013344 001403          BEQ    28           ;IS IT A SINGLE PORT RP04?
2101 013346 022705 024020          CMP     @24020,R5   ;BRANCH IF YES
2102 013352 001043          BNE    58           ;IS IT A DUAL PORT RP04?
2103 013354 012746 000121 28:    MOV     @121,-(SP)  ;BRANCH IF NO
2104 013360 004037 017174          JSR     R0,WRT,RP   ;DO A "READ-IN PRESET"
2105 013364 000000          RHCS1
2106 013366 013470          78

```

2107	013370	012746	010000		MOV	0BIT12,=(SP)	;SET FMT22=1	
2108	013374	004037	017174		JSR	R0,WRT,RP		
2109	013400	000032			RHOF			
2110	013402	013470			78			
2111	013404	004037	017036		JSR	R0,RD,RP	;READ RHDS1	
2112	013410	000012			RHDS1			
2113	013412	013470			78			
2114	013414	012605			MOV	(SP)+,R5	;AND SAVE IT IN R5	
2115	013416	100011			BPL	48	;BRANCH IF ATA=0	
2116	013420	116164	013056	000016	MOVB	ATABIT(R1),RHAS(R4)	;CLEAR ATTENTION BIT	
2117	013426	004037	017036		JSR	R0,RD,RP	;FIND OUT WHY ATA=1	
2118	013432	000014			RHER1			
2119	013434	013470			78			
2120	013436	006126			ROL	(SP)+	;IS IT UNSAFE?	
2121	013440	100412			BMI	68	;BRANCH IF YES	
2122	013442	005105		48:	COM	R5	;CHECK MOL, DPR, DRY, AND VV	
2123	013444	042705	167077		BIC	0<BIT12 BIT00 BIT07 BIT06>,R5		
2124	013450	001004			BNE	58	;BRANCH IF MOL, DPR, DRY, OR VV IS CLEAR	
2125	013452	112761	000001	012752	MOVB	01,DRVSTA(R1)	;SET DRIVE STATUS TO ONLINE	
2126	013460	000402			BR	68		
2127	013462	105061	012752	58:	CLRB	DRVSTA(R1)	;DRIVE STATUS = OFFLINE	
2128	013466	005720		68:	TST	(R0)+	;STEP OVER THE ERROR RETURN	
2129	013470	004037	020256	78:	JSR	R0,GETR15	;RESTORE R1-R5	
2130	013474	000200			RTS	R0	;RETURN	
2131								
2132							;REQUEST PRE-PROCESSOR-HANDLES SUBSYSTEM REQUEST	
2133								
2134							;CALL	
2135								
2136							;CALL THE RP04 DRIVER	
2137							;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK	
2138							;RETURN HERE IF QUEUE IS FULL	
2139							;RETURN HERE IF REQUEST IS IN QUEUE OR THERE	
2140							;IS AN ERROR CONDITION	
2141								
2142	013476	013746	177776		RP04:	MOV	00PS,=(SP)	;SAVE THE CALLING STATUS
2143	013502	013737	013074	177776		MOV	RPVEC+2,00PS	;DON'T ALLOW ANY RP04 INTERRUPTS
2144	013510	112737	000001	013006		MOVB	01,ACTDRV	;SET "ACTIVE DRIVER" FLAG
2145	013516	004037	020236			JSR	R0,SAVR15	;SAVE R1-R5
2146	013522	012002				MOV	(R0)+,R2	;PICKUP THE DRIVE PARAMETER BLOCK POINTER
2147	013524	005062	000016			CLR	16(R2)	;CLEAR THE STATUS/ERROR INDICATOR
2148	013530	111201				MOVB	(R2),R1	;PICKUP THE DRIVE NUMBER
2149	013532	105761	012752			TSTB	DRVSTA(R1)	;CHECK DRIVES STATUS
2150	013536	003017				BGT	18	;BRANCH IF ONLINE
2151	013540	105761	013010			TSTB	ULDFLG(R1)	;UNLOAD COMMAND IN QUEUE?
2152	013544	001034				BNE	38	;BRANCH IF YES
2153	013546	013704	013070			MOV	RPADR,R4	;UNIBUS ADDRESS OF RHCS1
2154	013552	004037	013250			JSR	R0,DRVINT	;GO INIT, THE DRIVE
2155	013556	000442				BR	68	;ERROR RETURN
2156	013560	105761	012752			TSTB	DRVSTA(R1)	;IS DRIVE STATUS ONLINE?
2157	013564	003004				BGT	18	;BRANCH IF YES
2158	013566	052762	140000	000016		BIS	0BIT15 BIT14,16(R2)	;ERROR--DRIVE IS OFFLINE OR NONEXISTENT
2159	013574	000423				BR	48	;GO TO EXIT
2160	013576	004037	020076	18:		JSR	R0,DRVQUE	;PUT THIS REQUEST IN QUEUE

```

2161 013602 000421          BR      58          ;QUEUE IS FULL
2162 013604 122762 000103 000002  CMPB   0103,2(R2)   ;IS THIS REQ. FOR AN UNLOAD?
2163 013612 001003          BNE    28          ;BR IF NO
2164 013614 112761 177777 013010  MOVB   0=1,ULDFLG(R1) ;SET THE "UNLOAD IN QUEUE" FLAG
2165 013622 105761 012742          28:    TSTB   DRVACT(R1)   ;IS THIS DRIVE ACTIVE?
2166 013626 001006          BNE    48          ;BR IS YES
2167 013630 004737 013672          JSR    PC,OPT      ;CALL THE OPTIMIZER
2168 013634 000403          BR     48
2169 013636 052762 120000 000016  38:    BIS    0BIT15|BIT13,16(R2) ;SET THE "UNLOAD IN QUEUE" ERROR FLAG
2170 013644 005720          48:    TST    (R0)+
2171 013646 004037 020256          58:    JSR    R0,GETR15   ;RESTORE R1-R5
2172 013652 105037 013006          CLRB   ACTDRV      ;CLEAR "ACTIVE DRIVER" FLAG
2173 013656 012637 177776          MOV    (SP)+,00PS  ;RETURN "PS" TO USER LEVEL
2174 013662 000200          RTS    R0          ;RETURN TO CALLER
2175 013664 004737 014674          68:    JSR    PC,C17     ;GO HANDLE THE PARITY ERROR
2176 013670 000765          BR     48
2177
2178          ;=OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
2179
2180          ;=CALL
2181          ;=
2182          ;=
2183          ;
2184 013672 004037 020236          OPT:   JSR    R0,SAVR15   ;SAVE R1-R5
2185 013676 013746 177776          MOV    00PS,-(SP)  ;SAVE PROC, STATUS
2186 013702 146137 013056 013004  BICB   ATABIT(R1),SRCHWT ;CLEAR "SEARCH WAIT" KEY
2187 013710 004737 020152          JSR    PC,GETREQ   ;GET "DPB" POINTER OF REQUEST
2188 013714 005702          TST    R2          ;IS THERE A REQUEST IN QUEUE?
2189 013716 001444          BEQ    68          ;NO--BRANCH TO EXIT
2190 013720 105761 012752          TSTB   DRVSTA(R1) ;IS DRIVE ONLINE?
2191 013724 003006          BGT    18          ;YES--BRANCH
2192 013726 004737 020174          JSR    PC,POPQUE   ;NO--REMOVE REQUEST FROM QUEUE
2193 013732 052762 140000 000016  BIS    0BIT15|BIT14,16(R2) ;SET ERROR BIT OF STATUS/ERROR INDICATOR
2194 013740 000426          BR     58          ;BRANCH TO EXIT
2195 013742 122762 000150 000002  18:    CMPB   0150,2(R2)   ;IS THE REQUEST FOR I/O?
2196 013750 002403          BLT    28          ;YES--BRANCH
2197 013752 004737 014252          JSR    PC,C14      ;CALL THE COMMAND INITIATOR
2198 013756 000417          BR     58          ;BRANCH TO EXIT
2199 013760 005737 013054          28:    TST    DTUW       ;DATA TRANSFER UNDERWAY?
2200 013764 002012          BGE    48          ;YES--GO START A SEARCH
2201 013766 005737 013032          TST    SEEKFG      ;DO IMPLIED SEEKS?
2202 013772 100404          BMI    38          ;YES---BRANCH
2203 013774 004037 015122          JSR    R0,LA       ;NO--DO LOOK AHEAD
2204 014000 000406          BR     58          ;RETURN HERE ON A PARITY ERROR
2205 014002 000403          BR     48
2206 014004 004737 014036          38:    JSR    PC,C11      ;GO START A SEARCH
2207 014010 000402          BR     58          ;START A DATA TRANSFER
2208 014012 004737 014144          48:    JSR    PC,C13      ;START A SEARCH
2209 014016 012637 177776          58:    MOV    (SP)+,00PS  ;RESTORE PROC, STATUS
2210 014022 004037 020256          JSR    R0,GETR15   ;RESTORE R1-R5
2211 014026 000207          RTS    PC
2212 014030 004737 017430          68:    JSR    PC,SET,IE  ;SET "IE" WITHOUT A "TRE"
2213 014034 000770          BR     58          ;EXIT THE ROUTINE
2214

```

```

2215 ;COMMAND INITIATOR
2216 ;
2217 ;CALL
2218 ;* MOV #DRVNUM,R1 ;DRIVE NUMBER
2219 ;* MOV #DPB,R2 ;ADDRESS OF DPB
2220 ;* JSR PC,C17 ;C17= C11,C13, OR C14
2221 ;* ;WHERE:
2222 ;* ;C11=DATA TRANSFER
2223 ;* ;C12=SEARCH REQUESTED BY DATA XFEM
2224 ;* ;C14=NOT DATA TRANSFER
2225
2226 014036 004737 020174 C11: JSR PC,POPQUE ;REMOVE REQUEST FROM "DRIVES WAIT" QUEUE
2227 014042 010237 013002 MOV R2,TRNSW ;PUT REQ. IN TRANSFER WAIT QUEUE
2228 014046 010203 C12: MOV R2,R3 ;DPB ADDRESS TO R3
2229 014050 013704 013070 MOV RPADR,R4 ;RHC61 ADDRESS
2230 014054 010164 000010 MOV R1,RHC62(R4) ;SELECT DRIVE
2231 014060 062703 000004 ADD #4,R3 ;DESIRED WORD COUNT
2232 014064 062704 000002 ADD #2,R4 ;RHC ADDRESS
2233 014070 012324 MOV (R3)+,(R4)+ ;LOAD WORD COUNT
2234 014072 012324 MOV (R3)+,(R4)+ ;LOAD BUFFER ADDRESS
2235 014074 012346 MOV (R3)+,=(SP) ;LOAD SECTOR AND TRACK
2236 014076 004037 017174 JSR R0,WRT,RP ;CALL THE LOAD(WRITE) ROUTINE
2237 014102 000006 RHDA ;INDEX OF REGISTER TO LOAD
2238 014104 014674 C17 ;ERROR RETURN ADDRESS
2239 014106 012346 MOV (R3)+,=(SP) ;LOAD CYLINDER ADDRESS
2240 014110 004037 017174 JSR R0,WRT,RP
2241 014114 000034 RHCA
2242 014116 014674 C17
2243 014120 016246 000002 MOV 2(R2),=(SP) ;LOAD "COMMAND+GO", "A176A16", AND "PSEL"
2244 014124 004037 017174 JSR R0,WRT,RP
2245 014130 000000 RHC61
2246 014132 014674 C17
2247 014134 010137 013054 MOV R1,DTUN ;SET "DATA TRANSFER UNDERWAY"
2248 014140 000137 014630 JMP C15
2249 014144 013704 013070 C13: MOV RPADR,R4 ;RHC61 ADDRESS
2250 014150 010164 000010 MOV R1,RHC62(R4) ;SELECT DRIVE
2251 014154 016246 000012 MOV 12(R2),=(SP) ;DESIRED CYLINDER ADDRESS
2252 014160 004037 017174 JSR R0,WRT,RP
2253 014164 000034 RHCA
2254 014166 014674 C17
2255 014170 116203 000010 MOV# 10(R2),R3 ;PICKUP SECTOR ADDRESS
2256 014174 163703 013104 SUB MXWHDW,R3 ;BACKUP BY MAX, SEARCH FOR I/O WINDOW
2257 014200 002002 BGE 10
2258 014202 062703 000026 ADD #22,,R3
2259 014206 010346 10: MOV R3,=(SP) ;COMBINE THE ADJUSTED SECTOR WITH
2260 014210 116266 000011 000001 MOV# 11(R2),1(SP) ;THE DESIRED TRACK
2261 014216 004037 017174 JSR R0,WRT,RP ;LOAD DESIRED TRACK & SECTOR
2262 014222 000006 RHDA
2263 014224 014674 C17
2264 014226 012746 000131 MOV #131,=(SP) ;START A SEARCH
2265 014232 004037 017174 JSR R0,WRT,RP
2266 014236 000000 RHC61
2267 014240 014674 C17
2268 014242 156137 013056 013004 DISB ATABIT(R1),SRCHWT ;SET "SEARCH WAIT" KEY

```

2269	014250	000567			BR	C15	
2270	014252	013704	013070	CI4:	MOV	RPADR,R4	;RHCS1 ADDRESS
2271	014256	010164	000010		MOV	R1,RHCS2(R4)	;SELECT DRIVE
2272	014262	116203	000002		MOVB	2(R2),R3	;PICKUP THE REQUESTED COMMAND
2273	014266	122703	000131		CMPB	0131,R3	;IS IT A SEARCH COMMAND?
2274	014272	001007			BNE	18	;BRANCH IF NO
2275	014274	016246	000010		MOV	10(R2),=(SP)	;LOAD DESIRED TRACK & SECTOR
2276	014300	004037	017174		JSR	R0,WRT,RP	
2277	014304	000006			RHDA		
2278	014306	014674			CI7		
2279	014310	000403			BR	28	;GO LOAD CYLINDER
2280	014312	122703	000105	10:	CMPB	0105,R3	;IS IT A SEEK COMMAND
2281	014316	001007			BNE	38	;BRANCH IF NO
2282	014320	016246	000012	20:	MOV	12(R2),=(SP)	;LOAD DESIRED CYLINDER
2283	014324	004037	017174		JSR	R0,WRT,RP	
2284	014330	000034			RHCA		
2285	014332	014674			CI7		
2286	014334	000546			BR	C16	
2287	014336	122703	000115	30:	CMPB	0115,R3	;IS IT AN "OFFSET" COMMAND?
2288	014342	001013			BNE	48	;BR IF NO
2289	014344	004037	017036		JSR	R0,RD,RP	;MERGE THE OFFSET VALUE INTO RHOF
2290	014350	000032			RHOF		;BUT DON'T CHANGE THE UPPER
2291	014352	014674			CI7		
2292	014354	116216	000001		MOVB	1(R2),(SP)	;BYTE WHEN LOADING THE
2293	014360	004037	017174		JSR	R0,WRT,RP	;REGISTER (RHOF)
2294	014364	000032			RHOF		
2295	014366	014674			CI7		
2296	014370	000530			BR	C16	;GO START THE COMMAND
2297	014372	122703	000107	40:	CMPB	0107,R3	;IS IT A "RECALIBRATE" COMMAND?
2298	014376	001525			BEQ	C16	;BRANCH IF YES
2299	014400	122703	000117		CMPB	0117,R3	;IS IT A RETURN TO CENTER?
2300	014404	001522			BEQ	C16	;BRANCH IF YES
2301	014406	122703	000103		CMPB	0103,R3	;IS IT AN "UNLOAD" COMMAND?
2302	014412	001016			BNE	58	;BRANCH IF NO
2303	014414	112761	000001	012742	MOVB	01,DRVACT(R1)	;SET THE DRIVE ACTIVE INDICATOR
2304	014422	105061	012752		CLRB	DRVSTA(R1)	;PUT DRIVE STATUS TO OFFLINE
2305	014426	112761	000001	013010	MOVB	01,ULDFLG(R1)	;SET "UNLOAD IN PROGRESS" FLAG
2306	014434	010346			MOV	R3,=(SP)	;START THE "UNLOAD" COMMAND
2307	014436	004037	017174		JSR	R0,WRT,RP	
2308	014442	000000			RHCS1		
2309	014444	014674			CI7		
2310	014446	000207			RTS	PC	;RETURN TO USER
2311	014450	122703	000143	50:	CMPB	0143,R3	;IS IT A "SET FORMAT" COMMAND?
2312	014454	001014			BNE	68	;BRANCH IF NO
2313	014456	004037	017036		JSR	R0,RD,RP	;READ THE OFFSET REGISTER
2314	014462	000032			RHOF		
2315	014464	014674			CI7		
2316	014466	116266	000001	000001	MOVB	1(R2),1(SP)	;COMBINE "FMT22", "ECI", AND "HCI"
2317	014474	004037	017174		JSR	R0,WRT,RP	;LOAD "FMT22", "ECI", AND/OR "HCI".
2318	014500	000032			RHOF		
2319	014502	014674			CI7		
2320	014504	000436			BR	126	
2321	014506	122703	000141	60:	CMPB	0141,R3	;IS IT A "GET REGISTER" COMMAND?
2322	014512	001023			BNE	108	;BRANCH IF NO

2323	014514	016203	000006		70:	MOV	6(R2),R3		;POINTS TO 1ST ADDRESS OF WHERE
2324									;TO PUT THE REGISTER(S)
2325	014520	116237	000010	014536		MOVB	10(R2),R0		;INIT. THE INDEX FOR THE FIRST REG.
2326	014526	116205	000011			MOVB	11(R2),R5		;INDEX OF LAST REG, TO MOVE
2327	014532	004037	017036		80:	JSR	R0,RD,RP		;READ RP04 REGISTER
2328	014536	000000			90:	RHCS1			;INDEX OF REG, TO READ
2329	014540	014674				CI7			
2330	014542	012623				MOV	(SP)+,(R3)+		;GET THE CONTENTS OF RH11/RP04 REG.
2331	014544	023705	014536			CMP	R0,R5		;LAST REG, BEEN READ?
2332	014550	001414				BEG	120		;GET OUT IF YES
2333	014552	062737	000002	014536		ADD	R2,R0		;INCREASE THE INDEX BY 2
2334	014560	000764				BR	R0		;LOOP--MORE TO READ
2335	014562	122703	000145		100:	CMPB	R0,R3		;IS IT A "SELECT DRIVE" COMMAND?
2336	014566	001405				BEG	120		;BRANCH IF YES
2337	014570	010346			110:	MOV	R3,-(SP)		;LOAD THE COMMAND
2338	014572	004037	017174			JSR	R0,WRT,RP		
2339	014576	000000				RHCS1			
2340	014600	014674				CI7			
2341	014602	004737	020174		120:	JSR	PC,POPQUE		;REMOVE REG, FROM QUEUE
2342	014606	052762	000200	000016		BIS	R0,BIT07,16(R2)		;SET THE "DONE" BIT
2343	014614	005737	013030			TST	SAVEFG		;SAVE THE RH11/RP04 REGISTERS?
2344	014620	100002				BPL	R0		;BRANCH IF NO
2345	014622	004737	017336			JSR	PC,SVRH11		;YES--GO SAVE THE REGISTERS
2346	014626	000207			130:	RTS	PC		;RETURN TO USER
2347	014630	006301			CI5:	ASL	R1		
2348	014632	012761	001750	013034		MOV	R1,01000,,TIMER(R1)		;SET A ONE SECOND TIMER
2349	014640	006201				ASR	R1		
2350	014642	112761	000001	012742		MOVB	R1,DRVACT(R1)		;SET THE DRIVE ACTIVE
2351	014650	000207				RTS	PC		;RETURN TO THE USER
2352	014652	010346			CI6:	MOV	R3,-(SP)		;LOAD THE COMMAND
2353	014654	004037	017174			JSR	R0,WRT,RP		
2354	014660	000000				RHCS1			
2355	014662	014674				CI7			
2356	014664	000761				BR	CI5		
2357	014666	105761	012742		CI7A:	TSTB	DRVACT(R1)		;IS THE DRIVE ACTIVE?
2358	014672	001405				BEG	CI7B		;BRANCH IF NO
2359	014674	012762	104000	000016	CI7:	MOV	R0,BIT15;BIT11,16(R2)		;SET "PARITY" ERROR INDICATOR
2360	014702	004737	017336			JSR	PC,SVRH11		;GO SAVE THE RH11/RP04 REGISTERS
2361	014706	012746	000111		CI7B:	MOV	R0,0111,-(SP)		;DO A "DRIVE CLEAR"
2362	014712	004037	017174			JSR	R0,WRT,RP		
2363	014716	000000				RHCS1			
2364	014720	014760				CI8			
2365	014722	004737	020056			JSR	PC,EMPTYQ		;EMPTY THE QUEUE
2366	014726	105061	013010			CLRB	ULDFLG(R1)		;CLEAR THE UNLOAD IN QUEUE FLAG
2367	014732	105061	012742			CLRB	DRVACT(R1)		;DRIVE IS IDLE
2368	014736	020137	013054			CMP	R1,DTUW		;IF THIS DRIVE HAD AN I/O REQUEST
2369	014742	001005				BNE	R0		;IN PROGRESS CLEAR ALL OF THE FLAGS
2370	014744	005037	013002			CLR	TRNSWT		
2371	014750	012737	177777	013054		MOV	R0,0-1,DTUW		
2372	014756	000207			10:	RTS	PC		
2373	014760	004037	020236		CI8:	JSR	R0,SAVR15		;SAVE R1-R5
2374	014764	013704	013070			MOV	RPADR,R4		;PICKUP THE ADDRESS OF THE FIRST REGISTER
2375	014770	005001				CLR	R1		
2376	014772	005003				CLR	R3		

```

2377 014774 105761 012742      101  TSTB   DRVACT(R1)      ;DRIVE ACTIVE?
2378 015000 001421                BEQ    30             ;BRANCH IF NO
2379 015002 013702 013002      MOV    TRNSWT,R2     ;GET THE "TRANSFER WAIT" QUEUE
2380 015006 020137 013054      CMP    R1,DTUM      ;DID THIS DRIVE HAVE AN I/O IN PROGRESS?
2381 015012 001402                BEQ    20             ;BRANCH IF YES
2382 015014 004737 020152      JSR    PC,GETREG     ;GET THE DPB POINTER
2383 015020 004737 017336      JSR    PC,SVRH11     ;SAVE RH11/RP04 REGISTERS
2384 015024 052762 102000 000016  BIS    0BIT15,0BIT10,16(R2) ;SET "NON-CLEARABLE PARITY" ERROR INDICATOR
2385 015032 012763 177777 013034  MOV    0=1,TIMER(R3) ;STOP THE TIMER
2386 015040 105061 012742      CLRB  DRVACT(R1)     ;SET "DRIVE ACTIVE" TO IDLE
2387 015044 105061 013010      CLRB  ULDFLG(R1)    ;CLEAR UNLOAD FLAG
2388 015050 005201                INC    R1             ;MOVE TO THE NEXT DRIVE
2389 015052 062703 000002      ADD    02,R3
2390 015056 042701 177770      BIC    0=C7,R1
2391 015062 001344                BNE    10             ;BRANCH IF MORE DRIVES
2392 015064 012737 177777 013054  MOV    0=1,DTUM     ;NO DATA TRANSFERS UNDERWAY
2393 015072 005037 013002      CLR    TRNSWT       ;CLEAR THE "TRANSFER WAIT" QUEUE
2394 015076 004737 017774      JSR    PC,CLRQUE     ;CLEAR ALL OF THE REQUEST QUEUES
2395 015102 012764 000040 000010  MOV    0BIT05,RHC62(R4) ;DO A MASSBUS INIT.
2396 015110 004737 017430      JSR    PC,SET,IE     ;SET "IE" WITHOUT "TRE"
2397 015114 004037 020256      JSR    R0,GETR15    ;RESTORE THE REGISTERS
2398 015120 000207                RTS    PC             ;RETURN
2399
2400                                ;LOOK AHEAD ROUTINE
2401                                ;
2402                                ;CALL
2403                                ;
2404                                ;
2405                                ;
2406                                ;
2407                                ;
2408                                ;
2409                                ;
2410 015122 013704 013070      LAI   MOV    RPADR,R4 ;GET RHC61'S ADDRESS
2411 015126 010164 000010      MOV    R1,RHC62(R4) ;SELECT DRIVE
2412 015132 004037 017036      JSR    R0,RD,RP     ;READ CURRENT CYLINDER
2413 015136 000036                RHCC
2414 015140 015252                40
2415 015142 022662 000012      CMP    (SP)+,12(R2) ;IS CURRENT CYLINDER=DESIRED
2416                                ;CYLINDER?
2417 015146 001037                BNE    30             ;EXIT IF NO
2418 015150 105261 013020      INCB  LACNT(R1)     ;INCREMENT THE LOOK AHEAD COUNT
2419 015154 126137 013020 013076  CMPB  LACNT(R1),MXLACT ;EXCEED MAX?
2420 015162 003026                BGT    20             ;BRANCH IF YES
2421 015164 116203 000010      MOVB  10(R2),R3     ;GET DESIRED SECTOR ADDRESS AND
2422 015170 000303                SWAB  R3             ;MULT, BY 64--ALIGN WITH
2423 015172 006203                ASR   R3             ;LOOK AHEAD REGISTER
2424 015174 006203                ASR   R3
2425 015176 012737 000340 177776  MOV    0340,00PB    ;PRIORITY LEVEL "7"
2426 015204 004037 017036      JSR    R0,RD,RP     ;READ LOOK AHEAD REGISTER
2427 015210 000020                RHLA
2428 015212 015252                40
2429 015214 162603                SUB   (SP)+,R3      ;CALCULATE THE DELTA
2430 015216 002002                BGE   10

```

```

2431 015220 062703 002600          ADD      0<22,064,>,R3    ;MAKE THE DELTA POSITIVE
2432 015224 023703 013100          100     CMP      MXDLTA,R3    ;CHECK THE DELTA TO SEE
2433 015230 002400                    BLT      30                ;IF IT IS WITHIN THE
2434 015232 023703 013102          CMP      MNDLTA,R3    ;WINDOW---IF YES, ZERO
2435 015236 002003                    BGE      30                ;THE LOOK AHEAD COUNT
2436 015240 105061 013020          200     CLRB    LACNT(R1)    ;AND TAKE THE I/O EXIT
2437 015244 005720                    TST     (R0)+
2438 015246 005720          300     TST     (R0)+    ;ADJUST THE RETURN ADDRESS
2439 015250 000200                    RTS     R0
2440 015252 004737 014674          400     JSR     PC,C17    ;PROCESS THE ERROR
2441 015256 000200                    RTS     R0    ;TAKE ERROR RETURN
2442
2443          ;*INTERRUPT SERVICE ROUTINE
2444
2445 015260 112737 000001 013006  ISR:    MOVB    01,ACTDRV    ;SET "ACTIVE DRIVER" FLAG
2446 015266 004037 020236                    JSR     R0,SAVR15    ;SAVE R1-R5
2447 015272 013704 013070                    MOV     RPADR,R4    ;ADDRESS OF RMSCS1
2448 015276 013701 013054                    MOV     DTUH,R1    ;GET "DATA TRANSFER UNDERWAY" INDICATOR
2449 015302 002403                    BLT     10                ;BRANCH IF NO DATA TRANSFER UNDERWAY
2450 015304 004737 015330                    JSR     PC,TD        ;CALL TRANSFER DONE
2451 015310 000402                    BR     20                ;EXIT
2452 015312 004737 015456          100     JSR     PC,SC        ;CALL SPECIAL CONDITIONS
2453 015316 004037 020256          200     JSR     R0,GETR15    ;RESTORE R1-R5
2454 015322 105037 013006                    CLRB    ACTDRV    ;CLEAR "ACTIVE DRIVER" FLAG
2455 015326 000002                    RTI
2456
2457          ;*TRANSFER DONE ROUTINE
2458
2459 015330 105061 012742          TD:    CLRB    DRVACT(R1)    ;SET DRIVE ACTIVE INDICATOR TO IDLE
2460 015334 012737 177777 013054          MOV     0=1,DTUH    ;NO DATA TRANSFERS UNDERWAY
2461 015342 006301                    ABL     R1
2462 015344 012761 177777 013034          MOV     0=1,TIMER(R1) ;CANCEL TIMEOUT
2463 015352 006201                    ABR     R1
2464 015354 013702 013002          MOV     TRNSWT,R2    ;GET "DPB" ADDRESS FROM THE
2465 015360 005037 013002          CLR     TRNSWT    ;TRANSFER WAIT QUEUE--CLEAR QUEUE
2466 015364 052762 000200 000016          BIS     0BIT07,16(R2) ;SET DONE
2467 015372 010164 000010          MOV     R1,RHC82(R4) ;SELECT THE DRIVE
2468 015376 004037 017036          JSR     R0,RD,RP    ;TRANSFER ERROR(TRE=1)?
2469 015402 000000                    RHC81
2470 015404 014674                    CI7
2471 015406 006126                    ROL     (SP)+
2472 015410 100410                    BHI     20                ;BR IF YES
2473 015412 005737 013030          TST     SAVEFG    ;SAVE THE RH11/RP04 REGISTERS?
2474 015416 100002                    BPL     10                ;BRANCH IF NO
2475 015420 004737 017336          JSR     PC,SVRH11    ;YES--SAVE THE REGISTERS
2476 015424 004737 013672          100     JSR     PC,OPT    ;CALL OPTIMIZER
2477 015430 000456                    BR     SC2    ;SPECIAL CONDITION (ENTRY 02)
2478 015432 052762 100100 000016  200     BIS     0BIT15|BIT06,16(R2) ;SET DATA ERROR FLAG
2479 015440 004737 020056          JSR     PC,EMPTYQ    ;EMPTY THE "DRIVES WAIT" QUEUE
2480 015444 004737 017336          JSR     PC,SVRH11    ;SAVE THE RH11/RP04 REGISTERS
2481 015450 012714 040111          MOV     040111,(R4) ;ISSUE A "DRIVE CLEAR"
2482 015454 000444                    BR     SC2    ;SPECIAL CONDITION (ENTRY 02)
2483
2484          ;*SPECIAL CONDITION ROUTINE

```

2485							
2486	015456	005001		SC1	CLR	R1	;START WITH DRIVE 0
2487	015460	012702	000001		MOV	#1,R2	
2488	015464	105761	012752	101	TSTB	DRVSTA(R1)	;NONEXISTENT?
2489	015470	002016			BGE	20	;NO==BRANCH
2490	015472	005201			INC	R1	;YES==MOVE TO THE NEXT DRIVE
2491	015474	106302			ASLB	R2	;MORE DRIVES?
2492	015476	103372			BCC	10	;YES==BRANCH
2493	015500	013746	013032		MOV	SEEKFG,-(SP)	;SAVE THE "SEEK FLAG"
2494	015504	013746	013030		MOV	SAVEFG,-(SP)	;SAVE THE "SAVE FLAG"
2495	015510	004737	013106		JSR	PC,RPINIT	;GO INIT, THE SUBSYSTEM
2496	015514	012637	013030		MOV	(SP)+,SAVEFG	;RESTORE THE "SAVE FLAG"
2497	015520	012637	013032		MOV	(SP)+,SEEKFG	;RESTORE THE "SEEK FLAG"
2498	015524	000405			BR	SC1	;TAKE ERROR EXIT
2499	015526	010164	000010	201	MOV	R1,RHCS2(R4)	;SELECT DRIVE
2500	015532	116405	000001		NOVB	1(R4),R5	;IS "SC"=1?
2501	015536	100413			BMI	SC2	;BRANCH IF YES
2502	015540	012701	000010	SC11	MOV	#0,,R1	;DO EIGHT DRIVES
2503	015544	005301		101	DEC	R1	;NEXT DRIVE
2504	015546	002743			BLT	SC	;BRANCH IF OUT OF DRIVES
2505	015550	105761	012742		TSTB	DRVACT(R1)	;IS THIS DRIVE IDLE?
2506	015554	001373			BNE	10	;BRANCH IF NO
2507	015556	104001			ERROR	1	;REPORT THE ERROR
2508	015560	004737	017430		JSR	PC,SET,IE	;GO SET INTERRUPT ENABLE
2509	015564	000207			RTS	PC	
2510	015566	012701	000003	SC21	MOV	#3,R1	;READ RHAS UP TO THREE TIMES
2511	015572	116403	000016	101	NOVB	RHAS(R4),R3	;READ "RHAS"
2512	015576	001011			BNE	20	;BRANCH IF ANY ATA BITS = 1
2513	015600	005301			DEC	R1	;COUNT THIS READ
2514	015602	003373			BGT	10	;LOOP IF MORE READS ALLOWED
2515	015604	004037	017036		JSR	R0,RD,RP	;READ CONTROL AND STATUS REGISTER
2516	015610	000000			RHCS1		
2517	015612	014760			CIS		
2518	015614	106126			ROLB	(SP)+	;IS "IE"=1?
2519	015616	100350			BPL	SC1	;NO==TAKE ERROR EXIT
2520	015620	000207			RTS	PC	;YES==RETURN
2521	015622	005046		201	CLR	=(SP)	;PROCESS ALL DRIVES THAT HAVE
2522	015624	110316			NOVB	R3,(SP)	;AN "ATA"=1
2523	015626	012703	000001		MOV	#1,R3	
2524	015632	005001			CLR	R1	
2525	015634	030316		SC41	BIT	R3,(SP)	;ATA=1?
2526	015636	001005			BNE	SC5	;YES==BRANCH
2527	015640	005201		SC31	INC	R1	;MOVE TO THE NEXT DRIVE
2528	015642	106303			ASLB	R3	
2529	015644	001373			BNE	SC4	;BRANCH IF MORE TO CHECK?
2530	015646	005726			TST	(SP)+	;CLEAN OFF THE STACK
2531	015650	000207			RTS	PC	;RETURN TO USER
2532	015652	023701	013054	SC51	CMP	DTW,R1	;IS THIS DRIVE SETUP FOR I/O?
2533	015656	001002			BNE	10	;NO==BRANCH
2534	015660	005726			TST	(SP)+	;YES==CLEAN OFF THE STACK (RHAS)
2535	015662	000622			BR	TD	;BRANCH TO "TRANSFER DONE"
2536	015664	105761	012752	101	TSTB	DRVSTA(R1)	;CHECK THE DRIVE STATUS
2537	015670	003030			BGT	SC6	;BRANCH IF ONLINE
2538	015672	105761	013010		TSTB	ULDFLG(R1)	;UNLOAD IN PROGRESS?

2539	015676	003420			BLE	28		;BRANCH IF NO
2540	015700	004737	020152		JSR	PC,GETREG		;GET DPB POINTER
2541	015704	004737	017336		JSR	PC,SVRH11		;SAVE THE RH11/RP04 REGISTERS
2542	015710	004737	016366		JSR	PC,SC12		;SAVE RHDS1, RHER1, RHER2, AND RHER3
2543								;ALSO DO A DRIVE INIT (DRVINT)
2544	015714	105761	012752		TSTB	DRVSTA(R1)		;DID DRIVE COME ONLINE?
2545	015720	003411			BLE	38		;NO==BRANCH
2546	015722	032737	040000	012732	BIT	#BIT14,RPERRS		;WAS THERE AN ERROR?
2547	015730	001565			BEQ	SC11		;NO == BRANCH
2548	015732	013705	012734		MOV	RPEPRS+2,R5		;YES == PICKUP RHEH1 AND
2549	015736	000447			BR	SC6A		;GO PROCESS THE ERROR
2550	015740	004737	016366	281	JSR	PC,SC12		;SAVE RHDS1, RHER1, RHER2, AND RHER3
2551								;ALSO DO A DRVINT
2552	015744	011605		381	MOV	(SP),R5		;PICKUP (RHS) BEFORE THE ERROR CALL
2553	015746	104005			ERROR	5		;REPORT THE ERROR
2554	015750	000733			BR	SC3		;GO CHECK FOR MORE ATA'S
2555	015752	006301			ASL	R1		
2556	015754	012761	177777	013034	MOV	#=1,TIMER(R1)		;STOP THE TIMER
2557	015762	006201			ASR	R1		
2558	015764	004737	020152		JSR	PC,GETREG		;GET THE DPB POINTER FROM THE QUEUE
2559	015770	010364	000016		MOV	R3,RHAS(R4)		;CLEAR ATTENTION
2560	015774	010164	000010		MOV	R1,RHC52(R4)		;SELECT DRIVE
2561	016000	004037	017036		JSR	R0,RD,RP		;READ THE RP04'S STATUS REG,
2562	016004	000012			RHDS1			
2563	016006	016224			SC0			
2564	016010	011605			MOV	(SP),R5		;AND PUT IT IN R5
2565	016012	006126			ROL	(SP)+		;WAS THERE AN ERROR?
2566	016014	100115			BPL	SC9		;BR IF NO
2567	016016	105761	012742		TSTB	DRVACT(R1)		;CHECK THE DRIVE ACTIVE INDICATOR
2568	016022	001522			BEQ	SC10		;BRANCH IF IDLE
2569	016024	004037	017036		JSR	R0,RD,RP		;READ ERROR REGISTER #1
2570	016030	000014			RHER1			
2571	016032	016224			SC0			
2572	016034	012605			MOV	(SP)+,R5		;AND SAVE IT IN R5
2573	016036	004737	017336		JSR	PC,SVRH11		;SAVE RH11/RP04 REGISTERS
2574	016042	012746	000111		MOV	#111,=(SP)		;ISSUE A DRIVE CLEAR
2575	016046	004037	017174		JSR	R0,=RT,RP		
2576	016052	000000			RHC51			
2577	016054	016224			SC0			
2578	016056	006105		SC6A1	ROL	R5		;WAS "UNSAFE" CONDITION =1?
2579	016060	100404			BMI	18		;BRANCH IF YES
2580	016062	052762	100240	000016	BIS	#BIT15 BIT07 BIT05,16(R2)		;INFORM USER OF ERROR
2581	016070	000443			BR	SC7		
2582	016072	004037	017036	181	JSR	R0,RD,RP		;READ DRIVE STATUS REG, #1
2583	016076	000012			RHDS1			
2584	016100	016224			SC0			
2585	016102	011605			MOV	(SP),R5		;SAVE RHDS1 IN R5
2586	016104	006126			ROL	(SP)+		; "ERR"=1?
2587	016106	100013			BPL	28		;BR IF NO==UNSAFE CLEARED
2588	016110	112761	177777	012752	MOVB	#=1,DRVSTA(R1)		;DRIVE IS UNSAFE
2589	016116	004737	017336		JSR	PC,SVRH11		;SAVE RH11/RP04 REGISTERS
2590	016122	010364	000016		MOV	R3,RHAS(R4)		;CLEAR ATTENTION
2591	016126	052762	110000	000016	BIS	#BIT15 BIT12,16(R2)		;INFORM USER OF UNSAFE ERROR
2592	016134	000421			BR	SC7		