RP04

DISK PACK FORMATTER

MD-11-DERPL-B

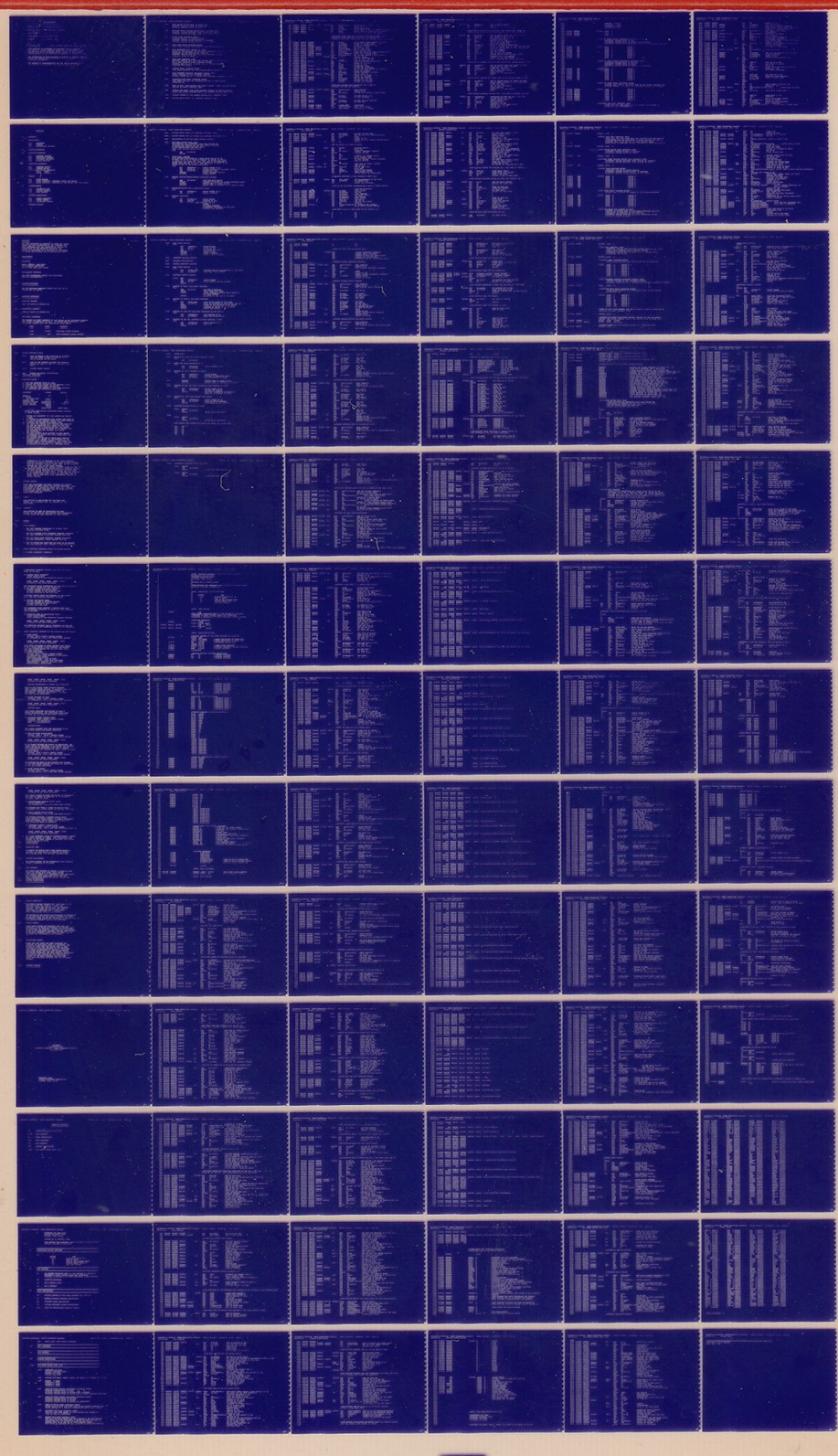
EP-DERPL-B-DL

FICHE 1 OF 1

MAY 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DEPPL-B-D

PRODUCT NAME: RP04 DISK PACK FORMATTER

DATE CREATED: MAY 21, 1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: R. MOORE/C. HESS

CONTENTS

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

C1

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

SW0 FORMAT (OR CHECK) CONTINUOUSLY

5.0 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 (18 BIT) MODE	Y OR N	Y
PROGRAM MODE	F OR C	F
STARTING CYL,TRK	0-410,0-18	0,0 *
ENDING CYL,TRK	0-410,0-18	410,18 *
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 (18 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 RP04 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 'RP04'

THIS ROUTINE IS THE DRIVER FOR THE RP04 DISK SYSTEM. IT IS ENTERED UPON ALL REQUESTS TO THE RP04 DRIVE.

5.3.2 'DRINIT'

THIS ROUTINE IS USED TO INITIALIZE THE RP04 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 DPIVER ROUTINE

01 ILLEGAL SUBSYSTEM INTERRUPT

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

#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
   RHAS= 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

   RETRIES UNSUCCESSFUL • SECTOR NOT ACCEPTABLE

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

#3 HEADER COMPARE NOT MADE
   DRIVE#=X  SN=X • DRIVE & SERIAL NUMBER
   CYL,TRK=XXX,XX • CYL & TRK COMPARSION 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,

#4 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,

#5 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 OCCURES 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,
#6 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,

#7 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

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 OCURRED

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

THE FORMAT PROGRAM FORMATS THE DISK PACK ON THE ASSIGNED DRIVE ONE TRACK AT A TIME, THE 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 :OT 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 "RP04 FORMATTER PROGRAM"

TABLE OF CONTENTS
****

PROGRAM BY R. MOORE/C. HESS

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
PACKAGE (MAINDEC-11-DZQAC-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
                ;*****************************************************************

46              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
                ;*****************************************************************

                ;*****************************************************************
        1828    ;RH11/RP04 DRIVER (REV. 0.9)
                ;*****************************************************************

                1830    ;COPYRIGHT (C) 1974
                        ;DIGITAL EQUIPMENT CORP.
                        ;MAYNARD, MA 01754
                        ;AUTHOR: JIM LACEY

                1836    ;STORAGE FOR RHDS1, RHER1, RHER2, AND RHER3 ON AN ERROR "2" OR "3"
                1837    RPERRS:  .= RHDS1
                        RPERRS+2 .= RHER1
                        RPERRS+4 .= RHER2
                        RPERRS+6 .= RHER3

                1847    ;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

                1861    ;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

                1875    ;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.

                1888    ;TRANSFER WAIT FLAG (TRNSWT=1 WORD)
                        ;THIS IS A ONE WORD QUEUE, IT WILL CONTAIN THE ADDRESS OF
                        ;"DPB" OF THE I/O OPERATION.

                1894    ;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 BIT=0 FOR DRIVE 0.
```

M1

```
                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=0 BYTES)
                        ULDFLG=0 IMPLIES NO UNLOAD COMMAND
                        ULDFLG>0 IMPLIES UNLOAD COMMAND IN PROGRESS
                        ULDFLG<0 IMPLIES UNLOAD COMMAND IN WAIT QUEUE

                1928    LOOK AHEAD COUNT (LACNT=0 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 (MCPEMX=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                 ;RETUPN 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,CI?  ;CI?= CI1,CI3, OR CI4
                                                ;WHERE:
                                                ;CI1=DATA TRANSFER
                                                ;CI2=SEARCH REQUESTED BY DATA XFER
                                                ;CI4=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

                2648    RP04 TIMER ROUTINE
                        CALL
                                MOV     #TIME,-(SP)     ;ELASPED 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                 ;RETURN HERE IF QUEUE IS FULL
                                RETURN2                 ;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
```

```
                3048    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
```

```
                                ;TITLE  MAINDEC-11-DERPL-B  "RP04 FORMATTER PROGRAM"
                                ;*COPYRIGHT (C) 1974,1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY R. MOORE/C. MESS
                                ;*
                                ;*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

000000                                  .=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
000200                                  .=200
                                ;*200 = NORMAL START
000200  000137  001104                  JMP     BEGIN
                                ;*204 = RESTART
000204  000137  001156                  JMP     RESTRT


                                ;SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                ;EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                ;EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                ;EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570          ;;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
```

```
                        R3=     %3              ;;GENERAL REGISTER
                        R4=     %4              ;;GENERAL REGISTER
                        R5=     %5              ;;GENERAL REGISTER
                        R6=     %6              ;;GENERAL REGISTER
                        R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
                        PR0=    0               ;;PRIORITY LEVEL 0
                        PR1=    40              ;;PRIORITY LEVEL 1
                        PR2=    100             ;;PRIORITY LEVEL 2
                        PR3=    140             ;;PRIORITY LEVEL 3
                        PR4=    200             ;;PRIORITY LEVEL 4
                        PR5=    240             ;;PRIORITY LEVEL 5
                        PR6=    300             ;;PRIORITY LEVEL 6
                        PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
```

```
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                        000014          TBITVEC=14              ;;"T" BIT
                        000014          TRTVEC= 14              ;;TRACE TRAP
                        000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                        000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                        000024          PWRVEC= 24              ;;POWER FAIL
                        000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                        000034          TRAPVEC=34              ;;"TRAP" TRAP
                        000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        000064          TPVEC=  64              ;;TTY PRINTER VECTOR
                        000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


                                        ;REGISTER ADDRESSES
                                                TKS=177560
                                                TKB=177562
                                                TPS=177564
                                                TPB=177566
                                                SLKCSR=172540           ;ADRS OF KW11-P STATUS REG
                                                SLKCSB=172542           ;ADRS OF KW11-P COUNTER BUFFER
                                                SLKS=177546             ;ADRS OF KW11-L STATUS REG

                        001100                  .=1100

                                        .SBTTL  RP04 ADDRESSES

001100  176700          SRPADR: .WORD   176700          ;RH11/RP04 UNIBUS ADDRESS
001102  000254          SRPVEC: .WORD   254             ;RH11 INTERRUPT VECTOR

                                        .SBTTL  MAIN PROGRAM
```

H2

```
001104  000005                          BEGIN:  RESET                   ;CLEAR WORLD
001106  012706  001100                          MOV     #STACK,SP       ;SET UP STACK
001112  012737  003116  000030                  MOV     #EMTHND,@#30    ;SET UP EMT ADRS
001120  012737  000340  000032                  MOV     #340,@#32       ;SET PS TO 7
001126  013737  001100  013070                  MOV     @RPADR,RPADR    ;MOVE RH11/RP04 ADDRESSES TO VECTOR
001134  013737  001102  013072                  MOV     @RPVEC,RPVEC    ;RH11 VECTOR ADDRESS
001142  012737  000240  001156                  MOV     #240,RESTRT     ;ALLOW RESTARTS AFTER INITIAL START
001150  004537  005166                          JSR     R5,TYPOUT       ;GO TYPE MESSAGE
001154  007466                                  MTITLE                  ;ADRS OF 'TITLE' MESSAGE
001156  000752                          RESTRT: BR      BEGIN           ;CHANGE TO 'NOP' AFTER INITIAL START
001160  012706  001100                          MOV     #STACK,SP       ;SET UP STACK
001164  000005                                  RESET                   ;CLEAR WORLD

                                        ;GO FIND OUT WHAT DRIVE

001166  004537  005166                  M0:     JSR     R5,TYPOUT       ;GO TYPE MESSAGE
001172  007546                                  MUNIT                   ;ADRS OF MESSAGE
001174  004737  005242                          JSR     PC,TTI          ;GO WAIT FOR DRIVE #
001200  022700  000104                          CMP     #104,R0         ;IS IT THE DEFAULT CHAR?
001204  001006                                  BNE     1$              ;BRANCH IF NOT THE DEFAULT CHAR
001206  005037  012540                          CLR     USEL            ;SELECT DRIVE ZERO
001212  004537  005166                          JSR     R5,TYPOUT       ;GO TYPE DEFAULT DRIVE NO.
001216  007561                                  MDRVD                   ;ADRS OF MSG
001220  000413                                  BR      GSEL            ;GO SEE IF DRIVE ZERO IS THERE

001222  004737  005314                  1$:     JSR     PC,NUMTST       ;IS IT A DIGIT?
001226  000757                                  BR      M0              ;RETURN HERE IF DRIVE # NOT A DIGIT
001230  022700  000007                          CMP     #7,R0           ;IS IT LESS THAN 7?
001234  103003                                  BCC     2$              ;YES - GO SELECT DRIVE
001236  004737  005406                          JSR     PC,QUES         ;TYPE ??
001242  000751                                  BR      M0              ;RETYPE LINE
001244  010037  012540                  2$:     MOV     R0,USEL         ;SAVE IN USEL

                                        ;THIS CODE CHECKS IF REQUIRED DRIVE IS AVAILABLE

001250  004737  013106                  GSEL:   JSR     PC,RPINIT       ;GO SEE WHAT DRIVES ARE AVAILABLE
001254  012737  177777  013032                  MOV     #177777,SEEKFG  ;SET TO -1 IF NO OPTIMIZATION IN DRIVER
001262  013700  012540                          MOV     USEL,R0         ;GET DRIVE NO.
001266  105760  012752                          TSTB    DRVSTA(R0)      ;LOOK AT DRIVE STATUS
001272  003035                                  BGT     M1              ;BRANCH IF ONLINE
001274  002404                                  BLT     1$              ;BRANCH IF UNSAFE OR NONEXISTENT
001276  004537  005166                          JSR     R5,TYPOUT       ;TYPE 'DRIVE OFFLINE'
001302  007577                                  MOFFLN                  ;ADRS OF MESSAGE
001304  000730                                  BR      M0              ;GO GET DRIVE # AGAIN
001306  006300                          1$:     ASL     R0              ;R0 INTO WORD INDEX
001310  005760  012762                          TST     DRVTYP(R0)      ;A DRIVE PRESENT?
001314  001004                                  BNE     2$              ;BR IF SO
001316  004537  005166                          JSR     R5,TYPOUT       ;TYPE 'DRIVE NOT PRESENT'
001322  007621                                  MDRNP                   ;ADRS OF MSG
001324  000720                                  BR      M0              ;GO GET DRIVE NO. AGAIN
001326  022760  020020  012762  2$:     CMP     #20020,DRVTYP(R0)       ;SINGLE PORT?
001334  001410                                  BEQ     3$              ;BR IF UNSAFE SINGLE PORT
001336  022760  024020  012762          CMP     #24020,DRVTYP(R0)       ;DUAL PORT?
001344  001404                                  BEQ     3$              ;BR IF UNSAFE DUAL PORT
```

I2

```
                                                JSR     R5,TYPOUT       ;TYPE 'NOT AN RP04'
                                                MNRP04                  ;ADRS OF MSG
                                                BR      M0              ;GO GET DRIVE NO, AGAIN
                                        38:     JSR     R5,TYPOUT       ;TYPE 'RP04 UNSAFE'
                                                MUSDR                   ;ADRS OF MSG
                                                BR      M0              ;GO GET DRIVE NO, AGAIN

                                                ;SEE WHICH MODE THE PROGRAM IS TO BE RUN IN,
                                                ;MODES ARE: 'FORMAT & VERIFY' OR 'CHECK FORMAT'

                                        M1:     CLR     MODE            ;SET MODE TO 'CHECK FORMAT'
                                                JSR     R5,TYPOUT       ;TYPE 'PROGRAM MODE'
                                                MMODE                   ;ADDR OF MSG
                                                JSR     PC,TTI          ;WAIT FOR ENTRY
                                                CMP     #'D,R0          ;'D' (DEFAULT) ENTERED ?
                                                BEQ     2$              ;BR IF DEFAULT
                                                CMP     #'C,R0          ;CHECK FORMAT MODE ?
                                                BEQ     1$              ;BR IF CHECK ONLY
                                                CMP     #'F,R0          ;FORMAT & VERIFY ?
                                                BEQ     3$              ;BR IF YES
                                                JSR     PC,QUES         ;TYPE '??'
                                                BR      M1              ;TRY AGAIN
                                        1$:     JSR     R5,TYPOUT       ;TYPE REST OF 'CHECK'
                                                MCHECK                  ;ADDR OF MSG
                                                BR      M1A             ;GET STARTING ADDRESS
                                        2$:     JSR     R5,TYPOUT       ;TYPE DEFAULT MESSAGE
                                                MFORMT                  ;ADDR OF MSG
                                                BR      4$              ;SET UP MODE
                                        3$:     JSR     R5,TYPOUT       ;TYPE REST OF 'FORMAT'
                                                MFORMT                  ;ADDR OF MSG
                                        4$:     MOV     #-1,MODE        ;SET MODE TO 'FORMAT & VERIFY'

                                                ;FIND OUT IF FORMAT IS TO BE IN 20 OR 22 SECTOR MODE

                                        M1A:    JSR     R5,TYPOUT       ;TYPE FORMAT MODE REQUEST
                                                MSIZE                   ;ADDRESS OF MESSAGE
                                                JSR     PC,TTI          ;WAIT FOR INPUT
                                                CMP     #'Y,R0          ;IS ENTRY A 'Y' ?
                                                BEQ     1$              ;BR IF IT IS
                                                CMP     #'N,R0          ;IS ENTRY A 'N' ?
                                                BEQ     2$              ;BR IF IT IS
                                                CMP     #'D,R0          ;IS ENTRY A 'D' ?
                                                BEQ     1$              ;BR IF IT IS
                                                JSR     PC,QUES         ;TYPE '??' - INVALID INPUT
                                                BR      M1A             ;TRY AGAIN
                                        1$:     JSR     R5,TYPOUT       ;TYPEOUT MODE SELECTED
                                                MSEC22                  ;ADDRESS OF MESSAGE
                                                MOV     #<20*,*22.>,WC  ;TRACK SIZE IN 22 SECTOR MODE
                                                MOV     #-<20*,*22.>,MWC  ;2'S COMPLEMENT WORD COUNT
                                                MOV     #-1,SEC20       ;22 SECTOR INDICATOR
                                                MOV     #21.,,MAXSEC    ;MAX SECTOR ADDRESS IN 22 SECTOR MODE
                                                MOVB    #20,FMTDPB+1    ;LOAD FMT22 BIT
                                                BR      3$              ;LOAD FORMAT BIT
                                        2$:     JSR     R5,TYPOUT       ;TYPE OUT 20 SECTOR MODE SELECTED
```

```
                                                MSEC20                  ;ADDRESS OF MESSAGE
001604  010165                                    MOV     #<20*,*20,>,MC  ;TRACK SIZE IN 20 SECTOR MODE
001606  012737  012120  012602                    MOV     #-<20*,*20,>,MNC  ;2'S COMPLEMENT WORD COUNT
001614  012737  165660  012604                    CLR     SEC20           ;20 SECTOR INDICATOR
001622  005037  012610                            MOV     #19.,MAXSEC     ;MAX SECTOR ADDRESS IN 20 SECTOR MODE
001626  012737  000023  012612                    CLRB    FMTDPB+1        ;CLEAR THE FMT22 BIT
001634  105037  012713                            MOVB    USEL,FMTDPB     ;LOAD DRIVE NUMBER
001640  113737  012540  012712  38$               MOVB    #143,FMTDPB+2   ;'SET FORMAT' COMMAND
001646  112737  000143  012714                    JSR     R0,RP04         ;GO TO THE DRIVER TO EXECUTE THE COMMAND
001654  004037  013476                            FMTDPB                  ;DPB ADDRESS
001660  012712                                    BR      3$              ;QUEUE FULL RETURN - RETRY THE COMMAND
001662  000766                                    TST     FMTDPB+16       ;CHECK STATUS WORD
001664  005737  012730          48$               BEQ     4$              ;BR IF NOT FINISHED
001670  001775                                    BMI     3$              ;BR IF ERROR IN SET FORMAT
001672  100762                                    BR      M2
001674  000400
                                                ;GO GET PARAMETERS - STARTING CYL & TRK
                                                ;AND ENDING CYL & TRK

001676  004537  005166          M2$:    JSR     R5,TYPOUT       ;GO TYPE MESSAGE 'STARTING CYL,TRK='
001702  010244                                  MMINCT          ;ADRS OF MESSAGE
001704  012737  000001  012542                  MOV     #1,SOFSW        ;SET UP SOFSW TO INDICATE STARTING INPUT
001712  004737  005446                          JSR     PC,GETDAT       ;GO GET STARTING CYL & TRK
001716  000767                                  BR      M2              ;RETURN HERE IF ERROR OR RUBOUT
001720  005037  012542                          CLR     SOFSW           ;SET UP SOFSW TO INDICATE ENDING INPUT
001724  004537  005166          M3$:    JSR     R5,TYPOUT       ;GO TYPE MESSAGE 'ENDING CYL,TRK='
001730  010277                                  MMAXCT          ;ADRS OF MESSAGE
001732  004737  005446                          JSR     PC,GETDAT       ;GO GET ENDING CYL & TRK
001736  000772                                  BR      M3              ;RETURN HERE IF ERROR OR RUBOUT
                                                ;THIS CODE CHECKS THE ORDER AND MAGNITUDE OF THE CYL & TRK ADRS
                                                ;INPUTTED BY THE TTY, ALSO ESTABLISHES THE TOTAL TRK COUNT

001740  013700  012546          CKADRS: MOV     ECYL,R0         ;SET R0 WITH ECYL
001744  163700  012550                  SUB     SCYL,R0         ;SEE IF END CYL IS SMALLER THAN START CYL
001750  103004                          BCC     1$              ;BRANCH IF NOT
001752  004537  005166                  JSR     R5,TYPOUT       ;TYPE IMPROPER DSK ADRS
001756  010423                          MADRER                  ;ADRS OF MESSAGE
001760  000746                          BR      M2              ;RETYPE REQUEST - INPUT CYL NOT IN PROPER FORM
001762  001014                  1$:     BNE     3$              ;BRANCH IF SCYL NOT EQUAL TO ECYL
001764  013700  012552                  MOV     ETRK,R0         ;SET UP R0 WITH ETRK
001770  163700  012554                  SUB     STRK,R0         ;SEE IF END TRK IS SMALLER THAN START TRK
001774  103004                          BCC     2$              ;BRANCH IF NOT
001776  004537  005166                  JSR     R5,TYPOUT       ;TYPE IMPROPER DSK ADRS
002002  010423                          MADRER                  ;ADRS OF MESSAGE
002004  000734                          BR      M2              ;RETYPE REQUEST - INPUT TRK NOT IN PROPER FORM
002006  010037  012556          2$:     MOV     R0,TTRKS        ;SAVE # OF TRACKS
002012  000417                          BR      M4              ;GO GET DATA PATTERN
002014  005037  012556          3$:     CLR     TTRKS           ;CLEAR TOTAL TRACK LOC
002020  062737  000023  012556  4$:     ADD     #23,TTRKS       ;ADD CYLINDER WORTH OF TRACKS
002026  005300                          DEC     R0              ;MAINTAIN CONTROL
002030  001373                          BNE     4$              ;BRANCH IF CYL DIFF NOT 0
002032  013700  012554                  MOV     STRK,R0         ;PUT STRK IN R0
002036  005400                          NEG     R0              ;MAKE TWO'S COMP
```

```
                                                ADD     R0,TTRKS        ;SUB PARTICAL TRK
                                                ADD     LTRK,TTRKS      ;ADD IN LND TRACKS

                                                ;GO GET DATA PATTERN FOR FORMAT

                                        M4:     JSR     R5,TYPOUT       ;GO TYPE 'SELECT PATTERN'
                                                MSELP                   ;ADRS OF MESSAGE
                                                JSR     PC,TTI          ;GO GET REPLY
                                                CMP     #104,R0         ;IS IT THE DEFAULT CHAR?
                                                BNE     1$              ;BRANCH IF NOT
                                                JSR     R5,TYPOUT       ;TYPE DEFAULT PATTERN
                                                MPATD                   ;ADRS OF MSG
                                                MOV     #2,R0           ;SELECT PATTERN OF WORST CASE
                                                BR      2$              ;GO SAVE PATTERN
                                        1$:     JSR     PC,NUMTST       ;SEE IF A NUMBER
                                                BR      M4              ;BRANCH IF NOT
                                                CMP     R0,#3           ;IS 0 LARGER THAN 2
                                                BCS     2$              ;BRANCH IF NOT
                                                JSR     PC,QUES         ;TYPE ??
                                                BR      M4              ;RETYPE LINE
                                        2$:     MOV     R0,PSEL         ;SAVE PATTERN SELECTED
                                                JSR     R5,TYPOUT       ;TYPE 'SELECTED'
                                                MSELD                   ;ADRS OF MESSAGE

                                                ;GO TYPE 'STARTING FORMAT ON DRIVE 0 X'

                                        M5:     TST     MODE            ;'FORMAT' OR 'CHECK' MODE ?
                                                BEQ     1$              ;BR IF 'CHECK' MODE
                                                JSR     R5,TYPOUT       ;TYPE 'STARTING FORMAT ON DRIVE 0'
                                                MSFOU                   ;ADRS OF MESSAGE
                                                BR      2$
                                        1$:     JSR     R5,TYPOUT       ;TYPE 'STARTING CHECK ON DRIVE 0'
                                                MSCHK                   ;ADDR OF MSG
                                        2$:     JSR     PC,UASN         ;GO PRINT DRIVE 0 IN USEL

                                                ;THIS CODE SETS UP THE SELECTED DATA PATTERN TO BE WRITTEN ON EACH SECTOR IMAGE

                                        STUFP:  CLR     PATA            ;CLEAR DATA PATTERN A
                                                CLR     PATB            ;CLEAR DATA PATTERN B
                                                TST     PSEL            ;SEE IF PATTERN OF ONES
                                                BEQ     STUF            ;BR IF SO
                                                COM     PATA            ;SET PATA TO ONES
                                                COM     PATB            ;SET PATB TO ONES
                                                CMP     #1,PSEL         ;SEE IF PATTERN OF ZEROS
                                                BEQ     STUF            ;BRANCH IF SO
                                                MOV     #165555,PATA    ;SET UP WORST CASE
                                                MOV     #133333,PATB    ;SET UP WORST CASE

                                                ;THIS CODE DUMP THE SELECT DATA PATTERN IN THE DATA BUFFER IN CORE

                                        STUF:   MOV     WC,R3           ;SET UP COUNTER
                                                MOV     #BUFP,R0        ;SET UP MEMORY POINTER
                                                MOV     PATA,R1         ;SET UP PATTERN IN R1
```

```
                                        MOV     PATB,R2         ;SET UP PATTERN IN R2
                                18:     MOV     R1,(R0)+        ;MOV 1ST PAT INTO MEM
                                        MOV     R2,(R0)+        ;MOV 2ND PAT INTO MEM
                                        DEC     R3              ;KEEP COUNT
                                        DEC     R3              ;KEEP COUNT
                                        BNE     18              ;DO IT AGAIN IF R3 NOT 0

                                        ;THIS CODE SETS UP FOR THE ACTUAL FORMAT

                                WRHDR1: JSR     PC,SETTBL       ;GO SET UP DRIVER TABLE
                                        JSR     PC,SETHDR       ;GO INITIALIZE HEADERS IN THE SECTOR BUFFER IN CORE
                                        JSR     PC,CKCLK        ;GO START UP THE CLOCK IF AVAILABLE
                                WRHDR2: CLR     SOFSW           ;CLEAR ERROR COUNTER
                                        CLR     RETRY           ;ZERO THE RETRY COUNTER
                                        CLRB    FMTDPB+10       ;RESTORE SECTOR
                                        MOV     #WC,FMTDPB+4    ;RESTORE WC
                                        MOV     #BUFP,FMTDPB+6  ;RESTORE CA
                                        TST     MODE            ;'FORMAT' OR 'CHECK' MODE ?
                                        BEQ     CKHDR           ;BR IF 'CHECK' MODE
                                WRHDR:  MOVB    #163,FMTDPB+2   ;SET WRITE HEADER & DATA COMMAND IN TBL
                                18:     JSR     R0,RP04         ;GO FORMAT A TRACK
                                        FMTDPB                  ;ADRS OF PARAMETERS = TBL
                                        BR      18              ;WAIT FOR QUEUE IF FULL
                                28:     TST     FMTDPB+16       ;WAIT FOR COMMAND TO COMPLETE
                                        BEQ     28              ;BRANCH IF NOT DONE
                                        BPL     CKHDR           ;BRANCH IF NO ERROR
                                        JSR     PC,ER           ;GO RESOLVE ERROR
                                        BR      WRHDR           ;GO TRY AGAIN

                                        ;THE CODE SETS UP FOR THE FORMAT CHECK

                                CKHDR:  MOVB    #153,FMTDPB+2   ;SET WRITE CHECK HEADER & DATA COMMAND IN TBL
                                18:     JSR     R0,RP04         ;GO CHECK THE TRK JUST FORMATTED
                                        FMTDPB                  ;ADRS OF PARAMETERS = TBL
                                        BR      18              ;WAIT FOR QUEUE IF FULL
                                28:     TST     FMTDPB+16       ;WAIT FOR COMMAND TO COMPLETE
                                        BEQ     28              ;BRANCH IF NOT DONE
                                        BPL     GOOD            ;BRANCH IF NO ERROR
                                        JSR     PC,ER           ;GO RESOLVE ERROR
                                        TST     MODE            ;'FORMAT' OR 'CHECK' MODE ?
                                        BEQ     CKHDR           ;BR IF 'CHECK' MODE
                                        BR      WRHDR           ;ER-REFORMAT SECTOR
                                GOOD:   TST     SOFSW           ;SEE IF RETRY
                                        BEQ     LOOPTS          ;BRANCH IF NOT
                                        CMP     #2,SOFSW        ;SEE IF LAST RETRY
                                        BNE     18              ;BRANCH IF NOT
                                        CMPB    MAXSEC,FMTDPB+10  ;SEE IF LAST SECTOR IN TRK
                                        BEQ     LOOPTS          ;BRANCH IF SO
                                        JSR     PC,SCAWC        ;GO SET UP TBL FOR REMAINING SECTORS
                                        TST     MODE            ;'FORMAT' OR 'CHECK' MODE ?
                                        BEQ     CKHDR           ;BR IF 'CHECK' MODE
                                        BR      WRHDR           ;GO FINISH FORMATTING TRK
                                18:     INC     SOFSW           ;RECORD GOOD RETRY
                                        BR      WRHDR           ;TRY ONCE MORE
```

```
433  002516  032737  040000  177570  LOOPTS: BIT     #BIT14,@#SWR    ;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,TRKTST       ;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                                          ;THIS CODE MAKES SURE EACH CYLINDER FORMATTED CONTAINS THE
441                                          ;PROPER CYLINDER ADDRESS. THE PROGRAM IS LOOKING FOR
442                                          ;POSSIBLE POSITIONER ERRORS THAT MAY HAVE OCCURED DURING THE FORMAT.
443
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          MOVB    #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  CMPER:  BIT     #BIT13,@#SWR    ;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                          MHDCMP
478  002724  004737  006662                  JSR     PC,DRSN         ;GO TYPE DRIVE & 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          MOVB    RBUF+3,LOCTRK   ;TRACK FROM HEADER
486  002764  013737  012622  012616          MOV     RBUF,LOCCYL     ;CYLINDER FROM HEADER
```

```
   487  003772  004737  006760                  JSR     PC,CYLTRK       ;GO TYPE CYL-TRK READ
   488  003776  012737  177777  012606          MOV     #-1,HEDERR      ;SET THE CYLINDER ERROR INDICATOR
   489  004004  005737  177570                  TST     @#SWR           ;HALT?
   490  004010  100001                          BPL     1$              ;BRANCH IF NOT
   491  004012  000000                          HALT                    ;ERROR- SOFT-ARE HEADER COMPARE ERROR
   492  004014  000207                  1$:     RTS     PC              ;RETURN
   493
   494                                          ;END OF FORMAT (OR CHECK) ROUTINE
   495
   496  004016  005737  012606          DONE:   TST     HEDERR          ;CYLINDER ERROR ?
   497  004022  001404                          BEQ     1$              ;BR IF NOT
   498  004024  004537  005166                  JSR     R5,TYPOUT       ;TYPEOUT 'REFORMAT MESSAGE'
   499  004030  012332                          HBDFMT                  ;MESSAGE ADDRESS
   500  004032  000412                          BR      3$
   501  004034  005737  012544          1$:     TST     MODE            ;'FORMAT' OR 'CHECK' MODE ?
   502  004040  001404                          BEQ     2$              ;BR IF 'CHECK' MODE
   503  004042  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE MESSAGE 'FORMAT COMPLETE'
   504  004046  010752                          MFCMPT                  ;ADRS OF MESSAGE
   505  004050  000403                          BR      3$
   506  004052  004537  005166          2$:     JSR     R5,TYPOUT       ;TYPE 'END OF CHECK' MESSAGE
   507  004056  010777                          MCCMPT                  ;ADDR OF MSG
   508  004060  032737  000001  177570  3$:     BIT     #SWO,@#SWR      ;SEE IF SWR 0 SET
   509  004066  001002                          BNE     .+6             ;BR IF SET
   510  004070  000137  001156                  JMP     RESTRT          ;ASK FOR NEXT DRIVE
   511  004074  012706  001100                  MOV     #STACK,SP       ;RESET STACK POINTER
   512  004100  000137  002142                  JMP     M5              ;DO THE FORMAT OR CHECK AGAIN
   513
   514                                          ;THIS CODE SERVICES A CLOCK INTERRUPT EVERY 16MS
   515
   516  004104  012746  000020          CLOCK:  MOV     #16.,-(SP)      ;PUT MILLISECONDS ON THE STACK
   517  004110  004037  016436                  JSR     R0,RPTMR        ;GO REPORT TIME
   518  004114  000002                          RTI                     ;RETURN AND CONTINUE
   519
   520                                          ;THIS IS THE EMT ERROR HANDLER-SELECTS ADRS OF ERROR HANDLER
   521
   522                                                                  ;SAVE THE REGISTERS
   523  004116  010137  004162          EMTHND: MOV     R1,SAVR         ;DRIVE NUMBER
   524  004122  010337  004164                  MOV     R3,SAVR+2       ;ATA BIT
   525  004126  010437  004166                  MOV     R4,SAVR+4       ;RHCS1 ADDRESS
   526  004132  010537  004170                  MOV     R5,SAVR+6       ;MOL STATE (BIT 4)
   527  004136  010037  004172                  MOV     R0,SAVR+10      ;SAVE UNKNOWN
   528  004142  010237  004174                  MOV     R2,SAVR+12      ;SAVE UNKNOWN
   529  004146  011600                          MOV     (SP),R0         ;GET EMT PC
   530  004150  014000                          MOV     -(R0),R0        ;GET EMT CALL INSTR
   531  004152  006300                          ASL     R0              ;TIMES 2
   532  004154  016000  173174                  MOV     EMTTBL-10002(R0),R0 ;DEVELOP EMT ADDRESS
   533  004160  000110                          JMP     (R0)            ;GO TO ERROR REPORT ROUTINE
   534
   535                                          ;SAVE REGS 1,3,4,6 5 HERE AFTER DRIVER ERRORS 1->5
   536
   537  004162  000000          SAVR:   0                               ;R1
   538  004164  000000                  0                               ;R3
   539  004166  000000                  0                               ;R4
   540  004170  000000                  0                               ;R5
```

```
003172  000000                          0                       ;R0
003174  000000                          0                       ;R2

                                        ;TABLE OF ERROR HANDLERS-DRIVER LEVEL

003176  003210          ERMTBL: ER1                             ;ILLEGAL SUBSYSTEM INTERRUPT
003200  003244                  ER2                             ;ILLEGAL DRIVE INTERRUPT
003202  003354                  ER3                             ;MASSBUS CONTROL BUS PARITY ERROR
003204  003460                  ER4                             ;CONTROL BUS PARITY ERROR
003206  003606                  ER5                             ;ATTENTION FROM AN UNAVAILABLE DRIVE

                                        ;HANDLES ILLEGAL SUBSYSTEM INTERRUPT

003210  032737  020000  177570  ER1:    BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
003216  001003                  BNE     1$                      ;BRANCH IF SO
003220  004537  005166                  JSR     R5,TYPOUT       ;TYPE ILLEGAL SUBSYSTEM INTERRUPT
003224  011023                  MER1
003226  005737  177570          1$:     TST     @#SWR           ;HALT?
003232  100001                  BPL     2$                      ;BRANCH IF NOT
003234  000000                  HALT                            ;ERROR- ILLEGAL SUBSYSTEM INTERRUPT
003236  004737  007270          2$:     JSR     PC,RESREG       ;RESTORE R0-R5
003242  000002                  RTI                             ;RETURN TO DRIVER

                                        ;HANDLES ILLEGAL DRIVE INTERRUPT

003244  032737  020000  177570  ER2:    BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
003252  001031                  BNE     1$                      ;BRANCH IF SO
003254  004537  005166                  JSR     R5,TYPOUT       ;TYPE ILLEGAL DRIVE INTERRUPT
003260  011065                  MER2
003262  004537  005166                  JSR     R5,TYPOUT       ;TYPE DRIVE
003266  011762                  MDRV
003270  013703  003162                  MOV     SAVR,R3         ;GET DRIVE #
003274  004737  006724                  JSR     PC,TDRNUM       ;GO TYPE DRIVE #
003300  004737  005376                  JSR     PC,CRLF         ;SET MARGIN
003304  004537  005166                  JSR     R5,TYPOUT       ;TYPE RHAS
003310  011771                  RHAS
003312  013703  003170                  MOV     SAVR+6,R3       ;GET RHAS
003316  004737  006604                  JSR     PC,OCTYP        ;GO TYPE IT
003322  004737  005376                  JSR     PC,CRLF         ;SET UP MARGIN
003326  004737  007064                  JSR     PC,SETREG       ;GO PREPARE FOR TYPE
003332  004737  007150                  JSR     PC,RHREGS       ;GO TYPE REGISTERS
003336  005737  177570          1$:     TST     @#SWR           ;HALT?
003342  100001                  BPL     2$                      ;BRANCH IF NOT
003344  000000                  HALT                            ;ERROR- ILLEGAL DRIVE INTERRUPT
003346  004737  007270          2$:     JSR     PC,RESREG       ;RESTORE R0-R5
003352  000002                  RTI                             ;RETURN TO DRIVER

                                        ;HANDLES MASSBUS CONTROL BUS PARITY ERRORS

003354  032737  020000  177570  ER3:    BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
003362  001027                  BNE     1$                      ;BRANCH IF SO
003364  004537  005166                  JSR     R5,TYPOUT       ;TYPE MASSBUS CONTROL BUS PARITY ERROR
003370  011123                  MER3
003372  004737  005376                  JSR     PC,CRLF         ;SET MARGIN
```

```
   595  003376  004537  005166                 JSR     R5,TYPOUT       ;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,TYPOUT       ;TYPE RD,*RD
   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$:   TST     @#SWR           ;HALT?
   606  003446  100001                         BPL     2$              ;BRANCH IF NOT
   607  003450  000000                         HALT                    ;ERROR- MASSBUS CONTROL BUS PARITY ERROR
   608  003452  004737  007270          2$:    JSR     PC,RESREG       ;RESTORE R0-R5
   609  003456  000002                         RTI                     ;RETURN TO DRIVER
   610
   611                                         ;HANDLES CONTROL BUS PARITY ERRORS
   612
   613  003460  032737  020000  177570  ER4$:  BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
   614  003466  001040                         BNE     1$              ;BRANCH IF SO
   615  003470  004537  005166                 JSR     R5,TYPOUT       ;TYPE CONTROL BUS PARITY ERROR
   616  003474  011172                         MER4
   617  003476  004737  005376                 JSR     PC,CRLF         ;SET MARGIN
   618  003502  004537  005166                 JSR     R5,TYPOUT       ;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,TYPOUT       ;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,TYPOUT       ;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$:   TST     @#SWR           ;HALT?
   634  003574  100001                         BPL     2$              ;BRANCH IF NOT
   635  003576  000000                         HALT                    ;ERROR- CONTROL BUS PARITY ERROR
   636  003600  004737  007270          2$:    JSR     PC,RESREG       ;RESTORE R0-R5
   637  003604  000002                         RTI                     ;RETURN TO DRIVER
   638
   639                                         ;HANDLES ATTENTION FROM AN UNAVAILABLE DRIVE
   640
   641  003606  032737  020000  177570  ER5$:  BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
   642  003614  001031                         BNE     1$              ;BRANCH IF SO
   643  003616  004537  005166                 JSR     R5,TYPOUT       ;TYPE ATTENTION FROM AN UNAVAILABLE DRIVE
   644  003622  011231                         MER5
   645  003624  004537  005166                 JSR     R5,TYPOUT       ;TYPE DRIVE
   646  003630  011762                         MDRV
   647  003632  013703  003162                 MOV     SAVR,R3         ;GET DRIVE #
   648  003636  004737  006724                 JSR     PC,TDRNUM       ;TYPE IT
```

```
                003642  004737  005376                  JSR     PC,CRLF         ;SET MARGIN
                003646  004537  005166                  JSR     R5,TYPOUT       ;TYPE RMAS
                003652  011771                          RMAS
                003654  013703  003170                  MOV     SAVR0,R3        ;GET RMAS
                003660  004737  006604                  JSR     PC,OCTYP        ;TYPE IT
                003664  004737  005376                  JSR     PC,CRLF         ;SET MARGIN
                003670  004737  007064                  JSR     PC,SETREG       ;GO PREPARE FOR TYPE
                003674  004737  007150                  JSR     PC,RHREGS       ;GO TYPE REGISTERS
                003700  005737  177570          18:     TST     @#SWR           ;HALT?
                003704  100001                          BPL     2$              ;BRANCH IF NOT
                003706  000000                          HALT                    ;ERROR- ATTENTION FROM AN UNAVAILABLE DRIVE
                003710  004737  007270          2$:     JSR     PC,RESREG       ;RESTORE R0-R5
                003714  000002                          RTI                     ;RETURN TO DRIVER


                                                        ;THIS CODE DECIDES WHAT TO DO WITH AN ERROR AT THE USER LEVEL

                003716  032737  000200  012730  ER:     BIT     #BIT7,FMTDPB+16 ;DID THE COMMAND TERMINATE
                003724  001402                          BEQ     1$              ;BRANCH IF FATAL ERROR
                003726  000137  004002                  JMP     ER6             ;GO DETERMINE WHAT KIND OF DATA ERROR
                003732  032737  070000  012730  1$:     BIT     #70000,FMTDPB+16        ;SEE WHICH FATAL ERROR
                003740  001402                          BEQ     2$              ;BRANCH IF A PARITY ERROR
                003742  000137  004776                  JMP     ER7             ;GO REPORT DRIVE UNSAFE
                003746  032737  004000  012730  2$:     BIT     #BIT11,FMTDPB+16        ;CHECK CONDITION OF PARITY ERROR
                003754  001402                          BEQ     3$              ;BRANCH IF FATAL MASSBUS PARITY ERROR
                003756  000137  005042                  JMP     ER8             ;GO REPORT UNCORRECTABLE MASSBUS PARITY ER
                003762  032737  002000  012730  3$:     BIT     #BIT10,FMTDPB+16        ;CHECK FOR FATAL PARITY ERROR
                003770  001402                          BEQ     4$              ;BR IF NOT
                003772  000137  005106                  JMP     ER9             ;GO REPORT FATAL MASSBUS PARITY ERROR
                003776  000137  005136          4$:     JMP     ER10            ;MUST BE SOFTWARE TIMEOUT


                                                        ;HANDLES THE DATA ERROR INDICATIONS

                004002  032737  000070  012730  ER6:    BIT     #70,FMTDPB+16   ;SEE IF A DATA ERROR
                004010  001402                          BEQ     1$              ;BRANCH IF SO
                004012  000137  004506                  JMP     ER6A            ;GO REPORT ERROR OTHER THAN DATA
                004016  122737  000173  012714  1$:     CMPB    #173,FMTDPB+2   ;SEE IF DOING A READ HEADER
                004024  001002                          BNE     2$              ;BRANCH IF NOT
                004026  000137  004742                  JMP     ER6D            ;GO REPORT ERROR ON READ HEADER
                004032  122737  000163  012714  2$:     CMPB    #163,FMTDPB+2   ;SEE IF DOING A WRITE HEADER
                004040  001002                          BNE     3$              ;BRANCH IF NOT
                004042  000137  004706                  JMP     ER6C            ;GO REPORT ERROR ON WRITE HEADER
                004046  005737  012542          3$:     TST     $OFSW           ;IS THIS A RETRY?
                004052  001002                          BNE     4$              ;BRANCH IF SO
                004054  000137  004140                  JMP     ERROR6          ;GO REPORT ERROR ON WRITE CHECK HEADER
                004060  032737  020000  177570  4$:     BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
                004066  001006                          BNE     5$              ;BRANCH IF SO
                004070  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE RETRIES UNSUCCESSFUL
                004074  012415                          NREUNS
                004076  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE SECTOR NOT ACCEPTABLE
                004102  012254                          NUNCOR
                004104  005737  177570          5$:     TST     @#SWR           ;HALT?
                004110  100001                          BPL     6$              ;BRANCH IF NOT
                004112  000000                          HALT                    ;ERROR- SECTOR FAILS TO FORMAT AFTER 1 OR 2 RETRIES
```

```
004114  022737  000025  012576  6$:     CMP     #25,SAVSEC      ;IS THIS THE LAST SECTOR IN THE TRK?
004122  001403                          BEQ     7$              ;BRANCH IF SO
004124  004737  006260                  JSR     PC,SCANC        ;GO SET UP TBL FOR REMAINING SECTORS
004130  000207                          RTS     PC              ;GO TRY NEXT SECTOR
004132  005726                  7$:     TST     (SP)+           ;RESTORE SP
004134  000137  002516                  JMP     LOOPTS          ;TRY NEXT TRK

                                        ;HANDLES A DATA ERROR ON A WRITE CHECK COMMAND

004140  005237  012542          ERROR6: INC     SOFS=           ;RECORD DATA ERROR
004144  013704  012650                  MOV     SAVREG+6,R4     ;GO GET RHDA
004150  042704  177740                  BIC     #177740,R4      ;SAVE ONLY THE BAD SECTOR + 1
004154  001002                          BNE     1$              ;BRANCH IF NOT ZERO
004156  012704  000026                  MOV     #26,R4          ;RESTORE TO LAST SECTOR +1
004162  005304                  1$:     DEC     R4              ;ADJUST SECTOR TO THE ONE THAT FAILED
004164  010437  012576                  MOV     R4,SAVSEC       ;SAVE IT
004170  032737  020000  177570          BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
004176  001076                          BNE     SETCA           ;BRANCH IF SO
004200  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE WRITE CHECK ERROR
004204  011303                          MER6
004206  004737  006662                  JSR     PC,DRSN         ;GO TYPE DR & SN
004212  004737  006744                  JSR     PC,CYLTK        ;GO TYPE CYL & TRK
004216  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE SECTOR=
004222  012057                          MSTOR
004224  005000                          CLR     R0              ;CLR TENS DIGIT
004226  013703  012576                  MOV     SAVSEC,R3       ;GET SECTOR #
004232  162703  000012          2$:     SUB     #12,R3          ;SEE HOW MANY TENS
004236  100402                          BMI     3$              ;BRANCH IF TOO SMALL
004240  005200                          INC     R0              ;RECORD TEN
004242  000773                          BR      2$              ;SUBTRACT AGAIN
004244  062703  000012          3$:     ADD     #12,R3          ;MAKE SECTOR POSITIVE AGAIN
004250  062700  000260                  ADD     #260,R0         ;SET UP FOR PRINT
004254  004737  005204                  JSR     PC,TTO          ;GO PRINT IT
004260  010300                          MOV     R3,R0           ;MOVE LSD INTO R0
004262  062700  000260                  ADD     #260,R0         ;SET UP FOR PRINT
004266  004737  005204                  JSR     PC,TTO          ;GO PRINT LSD
004272  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE GOOD DATA
004276  012104                          MGDATA
004300  013703  012646                  MOV     SAVREG+4,R3     ;GET ADRS OF FAILURE
004304  162703  000002                  SUB     #2,R3           ;BACK IT OFF BY TWO
004310  011303                          MOV     (R3),R3         ;GET THE DATA WORD
004312  004737  006604                  JSR     PC,OCTYP        ;GO TYPE IT
004316  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE BAD DATA
004322  012067                          MBDATA
004324  013703  012664                  MOV     SAVREG+22,R3    ;GET BAD DATA AT RHDB
004330  004737  006604                  JSR     PC,OCTYP        ;GO TYPE IT
004334  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE RHEC1=
004340  012121                          MRHEC1
004342  013703  012706                  MOV     SAVREG+44,R3    ;GO GET RHEC1
004346  004737  006604                  JSR     PC,OCTYP        ;GO TYPE IT
004352  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE RHEC2
004356  012132                          MRHEC2
004360  013703  012710                  MOV     SAVREG+46,R3    ;GET RHEC2
004364  004737  006604                  JSR     PC,OCTYP        ;GO TYPE IT
```

```
   757  004370  004737  007150                 JSR     PC,RHREGS       ;GO TYPE RH REGS
   758
   759                                         ;THIS CODE COMPUTES WC & CA FOR RETRIES ON WRITE CHECK ERRORS
   760
   761  004374  012700  020300         SETCA: MOV     #BUFP,R0        ;GET STARTING ADRS OF BUFFER
   762  004400  013701  012576                 MOV     SAVSEC,R1       ;GET SECTOR
   763  004404  005701                  18:    TST     R1              ;SEE IF ZERO
   764  004406  001404                         BEQ     2$              ;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      1$              ;GO CHECK NEXT SECTOR
   768  004420  010037  012720         2$:    MOV     R0,FMTDPB+6     ;SET UP CA OF BAD SECTOR
   769  004424  013700  012604         SETWC: MOV     MWC,R0          ;GET MAX WC
   770  004430  013701  012576                 MOV     SAVSEC,R1       ;GET SECTOR
   771  004434  005701                  1$:    TST     R1              ;SEE IF ZERO
   772  004436  001404                         BEQ     2$              ;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      1$              ;GO CHECK NEXT SECTOR
   776  004450  062700  000404         2$:    ADD     #260.,R0        ;POINT WC TO NEXT SECTOR
   777  004454  010037  012600                 MOV     R0,SAVWC        ;SAVE WC OF REMAINING SECTORS
   778  004460  012737  177374  012716         MOV     #-260.,FMTDPB+4 ;SET UP WC FOR BAD SECTOR
   779  004466  113737  012576  012722         MOVB    SAVSEC,FMTDPB+10        ;SET UP SECTOR ADRS IN TBL
   780  004474  005737  177570                 TST     @#MPR           ;HALT?
   781  004500  100001                         BPL     3$              ;BRANCH IF NOT
   782  004502  000000                         HALT                    ;ERROR- DATA ERROR
   783  004504  000207                  3$:    RTS     PC              ;GO REFORMAT FAILING SECTOR
   784
   785                                         ;CHECK FOR READ OR WRITE UNSAFE ERROR
   786
   787  004506  032737  000020  012730 ER6A:  BIT     #20,FMTDPB+16   ;CHECK FOR AN UNSAFE
   788  004514  001422                         BEQ     ER6B            ;BRANCH IF NOT AN UNSAFE
   789  004516  032737  020000  177570         BIT     @#BIT13,@#MPR   ;SKIP TYPEOUT?
   790  004524  001010                         BNE     1$              ;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          1$:    TST     @#MPR           ;HALT?
   797  004552  100001                         BPL     2$              ;BRANCH IF NOT
   798  004554  000000                         HALT                    ;ERROR- READ OR WRITE UNSAFE OCCURED
   799  004556  000137  001156          2$:    JMP     RESTRT          ;GO GET NEXT DRIVE
   800
   801                                         ;INDICATE A DRIVE ERROR HAS OCCURED
   802
   803  004562  005737  012574         ER6B:  TST     RETRY           ;IS THIS A RETRY?
   804  004566  001020                         BNE     3$              ;BRANCH IF SO
   805  004570  032737  020000  177570         BIT     @#BIT13,@#MPR   ;SKIP TYPEOUT?
   806  004576  001005                         BNE     1$              ;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          1$:    INC     RETRY           ;RECORD ERROR
```

```
004616  005737  177570                  TST     @#SWR           ;HALT?
004622  100001                          BPL     2$              ;BRANCH IF NOT
004624  000000                          HALT                    ;ERROR- DRIVE ERROR
004626  000207                  2$:     RTS     PC              ;TRY AGAIN
004630  022737  000002  012574  3$:     CMP     #2,RETRY        ;IS THIS THE LAST RETRY
004636  001403                          BEQ     4$              ;BRANCH IF SO
004640  005237  012574                  INC     RETRY           ;RECORD RETRY
004644  000207                          RTS     PC              ;TRY AGAIN
004646  032737  020000  177570  4$:     BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
004654  001006                          BNE     5$              ;BRANCH IF SO
004656  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE RETRIES UNSUCCESSFUL
004662  012415                          MREUNS
004664  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE REFORMAT PACK
004670  012307                          MREFMT
004672  005737  177570          5$:     TST     @#SWR           ;HALT?
004676  100001                          BPL     6$              ;BRANCH IF NOT
004700  000000                          HALT                    ;ERROR- DRIVE ERROR STILL THERE AFTER TWO RETRIES
004702  000137  001156          6$:     JMP     @#RESTRT        ;GO ASK FOR DRIVE

                                        ;HANDLES A DATA ERROR ON THE WRITE HEADER COMMAND

004706  032737  020000  177570  ER6C:   BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
004714  001005                          BNE     1$              ;BRANCH IF SO
004716  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE WRITE ERROR
004722  011413                          MER6C
004724  004737  007322                  JSR     PC,TYPALL       ;GO TYPE DATA
004730  005737  177570          1$:     TST     @#SWR           ;HALT?
004734  100001                          BPL     2$              ;BRANCH IF NOT
004736  000000                          HALT                    ;ERROR- ERROR OCCURED ON WRITE COMMAND
004740  000207                  2$:     RTS     PC              ;TRY TO WRITE AGAIN

                                        ;HANDLES A DATA ERROR ON THE READ HEADER COMMAND

004742  032737  020000  177570  ER6D:   BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
004750  001005                          BNE     1$              ;BRANCH IF SO
004752  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE HEADER COMPARE NOT MADE
004756  011434                          MER6D
004760  004737  007322                  JSR     PC,TYPALL       ;GO TYPE DATA
004764  005737  177570          1$:     TST     @#SWR           ;HALT?
004770  100001                          BPL     2$              ;BRANCH IF NOT
004772  000000                          HALT                    ;ERROR- HEADER COMPARE NOT MADE
004774  000207                  2$:     RTS     PC              ;RETURN

                                        ;HANDLES THE DRIVE UNSAFE ERRORS

004776  032737  020000  177570  ER7:    BIT     #BIT13,@#SWR    ;SKIP TYPEOUT?
005004  001010                          BNE     1$              ;BRANCH IF SO
005006  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE DRIVE UNSAFE
005012  011556                          MER7
005014  004737  007322                  JSR     PC,TYPALL       ;GO TYPE DATA
005020  004537  005166                  JSR     R5,TYPOUT       ;GO TYPE FORMAT TERMINATED
005024  012226                          MFTERM
005026  005737  177570          1$:     TST     @#SWR           ;HALT?
005032  100001                          BPL     2$              ;BRANCH IF NOT
```

```
                                        HALT                    ;ERROR- DRIVE UNSAFE ERROR
005036  000137  001156          28:   JMP     RESTRT            ;GO GET NEXT DRIVE

                                      ;HANDLES UNCORRECTABLE MASSBUS PARITY ERROR

005042  032737  020000  177570  ER8:  BIT     #BIT13,@#MSWR     ;SKIP TYPEOUT?
005050  001010                        BNE     1$                ;BRANCH IF SO
005052  004537  005166                JSR     R5,TYPOUT         ;GO TYPE UNCORRECTABLE MASSBUS PARITY ERROR
005056  011606                        MER8
005060  004537  005166                JSR     R5,TYPOUT         ;GO TYPE FORMAT TERMINATED
005064  012226                        MFTERM
005066  004737  006744                JSR     PC,CYLTK          ;GO GIVE THEM CYL, TRK=XXX,XX
005072  005737  177570          1$:   TST     @#MSWR            ;HALT?
005076  100001                        BPL     2$                ;BRANCH IF NOT
005100  000000                        HALT                      ;ERROR- UNCORRECTABLE MASSBUS PARITY ERROR
005102  000137  001156          2$:   JMP     RESTRT            ;GO GET NEXT DRIVE

                                      ;HANDLES FATAL MASSBUS PARITY ERROR

005106  032737  020000  177570  ER9:  BIT     #BIT13,@#MSWR     ;SKIP TYPEOUT?
005114  001005                        BNE     1$                ;BRANCH IF SO
005116  004537  005166                JSR     R5,TYPOUT         ;GO TYPE FATAL MASSBUS PARITY ERROR
005122  011606                        MER9
005124  004737  006744                JSR     PC,CYLTK          ;GO GIVE THEM CYL,TRK=XXX,XX
005130  000000                  1$:   HALT                      ;FATAL MASSBUS PARITY ERROR
005132  000137  001156                JMP     RESTRT            ;RESTART PROGRAM


                                      ;HANDLES A SOFTWARE TIMEOUT (HUNG RP04)

005136  032737  020000  177570  ER10: BIT     #BIT13,@#MSWR     ;SKIP TYPEOUT?
005144  001005                        BNE     1$                ;BR IF SO
005146  004537  005166                JSR     R5,TYPOUT         ;GO TYPE 'SOFTWARE TIMEOUT'
005152  011716                        MER10                     ;ADRS OF MSG
005154  004737  007322                JSR     PC,TYPALL         ;GO TYPE DATA
005160  000000                  1$:   HALT                      ;SYSTEM HUNG - OPERATION FAILED TO COMPLETE
005162  000137  001156                JMP     RESTRT            ;GO RESTART PROGRAM

                                .SBTTL  SUPPORT SUBROUTINES


                                      ;TYPES OUT CHARACTERS POINTED TO BY THE ADRS IN R1

005166  012501                  TYPOUT: MOV     (R5)+,R1          ;SET UP ADRS OF MESSAGE
005170  112100                  1$:   MOVB    (R1)+,R0          ;GET CHARACTER
005172  001403                        BEQ     2$                ;ARE WE DONE TYPING?
005174  004737  005204                JSR     PC,TTO            ;GO TYPE IT
005200  000773                        BR      1$                ;GET NEXT CHARACTER
005202  000205                  2$:   RTS     R5                ;EXIT - DONE TYPING

                                      ;DOES THE ACTUAL TYPING - TYPES THE NUMBER OF NULL CHARS SPECIFIED IN 'NULCNT'
```

```
   919  005204  010446                TTO:    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    TKS             ;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     1$              ;SKIP OVER TYPE
   938  005262  020027  000212               CMP     R0,#212         ;DON'T PRINT A LF
   939  005266  001402                       BEQ     1$              ;SKIP OVER TYPE
   940  005270  004737  005204               JSR     PC,TTO          ;ECHO CHAR TYPED
   941  005274  042700  000200        1$:    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     RESTRT          ;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     TYPER           ;YES - GO TO TYPER
   951  005322  020027  000072               CMP     R0,#72          ;IS ASCII LARGER THAN 71?
   952  005326  100005                       BPL     TYPER           ;YES - GO TO TYPER
   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        TYPER: 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        UASM:  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              ;RLTURN
```

```
                                        ;THIS ROUTINE PRINTS ?? ON IMPROPER KEYBOARD INPUTS
005406  004537  005166  QUES:  JSR     R5,TYPOUT       ;GO TYPE MESSAGE
005412  007567                 MQUES                   ;ADRS OF MESSAGE
005414  000207                 RTS     PC              ;RETURN  TO MAIN PROGRAM

                               ;THIS ROUTINE SETS UP THE TTY INPUT BUFFER AND COUNTER
005416  012703  012634  SETPC:  MOV     #CPTR,R3        ;PUT CHAR PTR IN R3
005422  010204                 MOV     R2,R4           ;PUT CHAR COUNT IN R4
005424  000207                 RTS     PC              ;RETURN

                               ;THIS ROUTINE CLEARS THE TTY INPUT BUFFER

005426  004737  005416  CLRBUF: JSR     PC,SETPC        ;GO SET UP POINTER & COUNT
005432  005023          1$:    CLR     (R3)+           ;MOVE 0'S THRU R3 R4 TIMES
005434  005304                 DEC     R4              ;KEEP COUNT
005436  001375                 BNE     1$              ;DONE IF R4 EQ 0
005440  004737  005416         JSR     PC,SETPC        ;SET UP POINTER & COUNT AGAIN
005444  000207                 RTS     PC              ;EXIT

                               ;THIS ROUTINE READS & CHECKS THE CYL & TRK INPUT FROM THE TTY

005446  012702  000004  GETDAT: MOV     #4,R2           ;SET UP MAX # OF DIGITS EXPECTED
005452  004737  005426         JSR     PC,CLRBUF       ;GO CLEAR DIGIT BUFFER
005456  004737  005242         JSR     PC,TTI          ;GO GET CHAR
005462  022700  000104         CMP     #104,R0         ;IS IT THE DEFAULT CHAR?
005466  001031                 BNE     4$              ;BRANCH IF NOT
005470  005737  012542         TST     $OFSW           ;IS THIS STARTING OR ENDING CYL?
005474  001410                 BEQ     1$              ;BRANCH IF ENDING CYL
005476  005037  012550         CLR     $CYL            ;SET STARTING CYL TO 0
005502  005037  012554         CLR     $TRK            ;SET STARTING TRK TO 0
005506  004537  005166         JSR     R5,TYPOUT       ;INDICATE DEFAULT VALUES
005512  010272                 MCYTK0                  ;ADRS OF MSG
005514  000411                 BR      2$              ;GO SET UP RETURN
005516  012737  000632  012546 1$:     MOV     #410.,,ECYL     ;SET ENDING CYL TO 410
005524  012737  000022  012552         MOV     #18.,,ETRK      ;SET ENDING TRK TO 18
005532  004537  005166         JSR     R5,TYPOUT       ;INDICATE DEFAULT VALUES
005536  010324                 MCYTK1                  ;ADRS OF MSG
005540  062716  000002  2$:     ADD     #2,(SP) ;ADD TWO TO RETURN ADRS
005544  000207                 RTS     PC              ;RETURN FROM DEFAULT COND.
005546  004737  005242  3$:     JSR     PC,TTI          ;GO GET NEXT CHAR FOR CYLINDER
005552  022700  000177  4$:     CMP     #177,R0         ;IS IT A RUBOUT?
005556  001001                 BNE     5$              ;BRANCH IF NOT A RUBOUT
005560  000207                 RTS     PC              ;GO RETYPE LINE
005562  022700  000054  5$:     CMP     #54,R0          ;IS IT THE SEPERATOR?
005566  001024                 BNE     6$              ;BRANCH IF NOT A COMMA
005570  004737  006120         JSR     PC,DECBIN       ;CONVERT CYL # TO BINARY
005574  023727  012572  000633 CMP     TEMP,#411.      ;SEE IF CYL REQUEST IS GREATER THAN 410
005602  103404                 BCS     6$              ;BRANCH IF WITHIN LIMITS
005604  004537  005166         JSR     R5,TYPOUT       ;TYPE LIMIT EXCEEDED
005610  010334                 MCLIMT                  ;ADRS OF MESSAGE
005612  000207                 RTS     PC              ;RETURN TO RETYPE LINE
```

```
1027  005614  005737  012542          68$:    TST     SOFSW           ;IS THIS THE STARTING OR ENDING CYL?
1028  005620  001403                          BEQ     7$              ;BRANCH IF ENDING CYL
1029  005622  013737  012572  012550          MOV     TEMP,SCYL       ;SAVE TEMP AT STARTING CYL
1030  005630  013737  012572  012546  78$:    MOV     TEMP,ECYL       ;SAVE TEMP AT ENDING CYL
1031  005636  000412                          BR      10$             ;GO GET TRK ADRS
1032  005640  004737  005314          88$:    JSR     PC,NUMTST       ;SEE IF A NUMBER
1033  005644  000207                          RTS     PC              ;RETURN = BAD TYPE
1034  005646  005304                          DEC     R4              ;KEEP COUNT OF DIGITS COMING IN
1035  005650  001003                          BNE     9$              ;BRANCH IF NOT THE 4TH
1036  005652  004737  005406                  JSR     PC,QUES         ;NO COMMA AFTER 3 DIGITS = TYPE ??
1037  005656  000207                          RTS     PC              ;RETURN = NUMBER TOO BIG
1038  005660  010023          98$:    MOV     R0,(R3)+        ;SAVE DIGIT IN BUFFER
1039  005662  000731                          BR      3$              ;GET NEXT CHAR
1040  005664  012702  000003          108$:   MOV     #3,R2           ;SET UP MAX # OF DIGITS EXPECTED
1041  005670  004737  005426                  JSR     PC,CLRBUF       ;GO CLEAR DIGIT BUFFER
1042  005674  004737  005242          118$:   JSR     PC,TTI          ;GO GET NEXT CHAR
1043  005700  022700  000177                  CMP     #177,R0         ;IS IT A RUBOUT?
1044  005704  001001                          BNE     12$             ;BRANCH IF NOT A RUBOUT
1045  005706  000207                          RTS     PC              ;GO RETYPE LINE
1046  005710  022700  000015          128$:   CMP     #15,R0          ;IS IT A TERMINATOR?
1047  005714  001026                          BNE     15$             ;BRANCH IF NOT A CR
1048  005716  004737  006120                  JSR     PC,DECBIN       ;CONVERT TRK # TO BINARY
1049  005722  023727  012572  000023          CMP     TEMP,#19.       ;SEE IF TRK REQUEST IS GREATER THAN 19
1050  005730  103404                          BCS     13$             ;BRANCH IF WITHIN LIMITS
1051  005732  004537  005166                  JSR     R5,TYPOUT       ;TYPE LIMIT EXCEEDED
1052  005736  010371                          MTLIMT                  ;ADRS OF MESSAGE
1053  005740  000207                          RTS     PC              ;RETURN = TRK ADRS TO LARGE
1054  005742  005737  012542          138$:   TST     SOFSW           ;IS THIS THE STARTING OR ENDING TRK?
1055  005746  001403                          BEQ     14$             ;BRANCH IF ENDING TRK
1056  005750  013737  012572  012554          MOV     TEMP,STRK       ;SAVE TEMP AT STARTING TRK
1057  005756  013737  012572  012552  148$:   MOV     TEMP,ETRK       ;SAVE TEMP AT ENDING TRK
1058  005764  062716  000002                  ADD     #2,(SP) ;ADD TWO TO RETURN ADRS
1059  005770  000207                          RTS     PC              ;RETURN = CYL & TRK ADRS ARE SAVED
1060  005772  004737  005314          158$:   JSR     PC,NUMTST       ;SEE IF A NUMBER
1061  005776  000207                          RTS     PC              ;RETURN = BAD TYPE
1062  006000  005304                          DEC     R4              ;KEEP COUNT OF DIGITS COMING IN
1063  006002  001003                          BNE     16$             ;BRANCH IF NOT THE 3RD
1064  006004  004737  005406                  JSR     PC,QUES         ;NO CR AFTER 2 DIGITS = TYPE ??
1065  006010  000207                          RTS     PC              ;RETURN = NUMBER TOO BIG
1066  006012  010023          168$:   MOV     R0,(R3)+        ;SAVE DIGIT IN BUFFER
1067  006014  000727                          BR      11$             ;GET NEXT CHAR
1068
1069                                  ;THIS ROUTINE SETS UP THE INPUT TABLE FOR THE DRIVER ROUTINE
1070
1071  006016  113737  012540  012712  SETTBL: MOVB    USEL,FMTDPB     ;SET UP DRIVE #
1072  006024  105037  012715                  CLRB    FMTDPB+3        ;CLEAR HIGH ORDER ADRS BITS
1073  006030  013737  012604  012716          MOV     #WC,FMTDPB+4    ;LOAD UP WORD COUNT
1074  006036  012737  020300  012720          MOV     #BUFP,FMTDPB+6  ;LOAD UP CURRENT ADRS
1075  006044  105037  012722                  CLRB    FMTDPB+10       ;SET SECTOR TO ZERO
1076  006050  113737  012554  012723          MOVB    STRK,FMTDPB+11  ;SET UP STARTING TRK ADRS
1077  006056  013737  012550  012724          MOV     SCYL,FMTDPB+12  ;SET UP STARTING CYL
1078  006064  012737  012642  012726          MOV     #SAVREG,FMTDPB+14       ;LOCATE POINTER TO 'RM' REGS
1079  006072  005037  012730                  CLR     FMTDPB+16       ;CLEAR RP04 STATUS
1080  006076  000240                          NOP
```

```
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     #1,R0           ;SEE HOW MANY 10'S
1091  006132  103404                        BCS     MSD             ;GO TRY HUNDREDS
1092  006134  062737  000012  012572        ADD     #12,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     #1,R0           ;SEE HOW MANY 100'S
1096  006152  103404                        BCS     DTBDNE          ;EXIT IF DONE
1097  006154  062737  000144  012572        ADD     #144,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  027737  000022  012562        CMP     #22,TRKCNT      ;IS THIS THE LAST TRK IN CYL?
1110  006214  001403                        BEQ     20              ;BRANCH IF SO
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     #2,(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  SCAWC:  MOV     SAVWC,FMTDPB+4  ;SET UP WC FOR REMAINING SECTORS
1126  006266  062737  001000  012720        ADD     #520,,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     SOFSW           ;RESET RETRY COUNTER
1129  006304  000207                        RTS     PC              ;RETURN TO COMPLETE TRK FORMAT
1130
1131
1132                                        ;THIS ROUTINE INITIALIZES THE SECTOR COUNT AND HEADER POINTER
1133                                        ;FOR THE HEADER & DATA BUFFER IN MEMORY
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     STRK,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     1$              ;BR IF 20 SECTOR MODE
1148  006344  052702  010000                  BIS     #10000,R2       ;SET THE 22 SECTOR FORMAT BIT
1149  006350  010220                  1$:     MOV     R2,(R0)+        ;WRITE IN HEADER AREA OF CORE THE CYL ADRS
1150  006352  010320                          MOV     R3,(R0)+        ;WRITE IN HEADER AREA OF CORE THE TRK ADRS
1151  006354  005020                          CLR     (R0)+           ;CLR 1ST KEY-WRD
1152  006356  005010                          CLR     (R0)            ;CLR 2ND KEY-WRD
1153  006360  062700  001002                  ADD     #514.,R0        ;SET UP FOR NEXT HEADER
1154  006364  005203                          INC     R3              ;UPDATE SECTOR ADRS FOR NEXT HEADER
1155  006366  005301                          DEC     R1              ;MAINTAIN COUNT OF SECTORS
1156  006370  002367                          BGE     1$              ;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          UPDACY: JSR     PC,SECCNT       ;SET UP SECTOR COUNT AND HEADER POINTER
1162  006400  005220                  1$:     INC     (R0)+           ;INCREMENT FOR NEXT CYLINDER
1163  006402  042710  177400                  BIC     #177400,(R0)    ;RESET TRK ADRS TO 0
1164  006406  062700  001006                  ADD     #518.,R0        ;SET UP FOR NEXT HEADER
1165  006412  005301                          DEC     R1              ;COUNT SECTORS
1166  006414  002371                          BGE     1$              ;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          1$:     ADD     #400,(R0)       ;INDEX TRK ADRS
1174  006432  062700  001010                  ADD     #520.,R0        ;SET UP FOR NEXT HEADER
1175  006436  005301                          DEC     R1              ;COUNT SECTORS
1176  006440  002372                          BGE     1$              ;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          1$:     SUB     #144,R0         ;SUBTRACT 100 DEC
1186  006466  100402                          BMI     2$              ;BRANCH IF # NOT LARGE ENOUGH
1187  006470  005201                          INC     R1              ;RECORD EACH 100 DEC
1188  006472  000773                          BR      1$              ;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 # 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 # 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 MCYLTK
1212
1213  006572  062701  000260          MKDIG:  ADD     #260,R1         ;MAKE ASCII DIGIT
1214  006576  110122                          MOVB    R1,(R2)+        ;PUSH DIGIT INTO MCYLTK
1215  006600  060300                          ADD     R3,R0           ;MAKE # 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 #
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 SO
1237  006660  000207                          RTS     PC              ;DONE TYPING-RETURN
1238
1239                                          ;THIS ROUTINE TYPES THE DRIVE# AND S/N
1240
1241  006662  004537  005166          DRSN:   JSR     R5,TYPOUT       ;GO TYPE DRIVE#
1242  006666  011702                          MDRV
```

```
1243  006670  013703  012652                  MOV     SAVREG+10,R3    ;GET DRIVE # FROM RMCS2
1244  006674  004737  006724                  JSR     PC,TDRNUM       ;GO TYPE DRIVE #
1245  006700  004537  005166                  JSR     R5,TYPOUT       ;TYPE S/N
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     #177770,R3      ;CLEAR JUNK
1255  006730  010300                          MOV     R3,R0           ;PLACE DIGIT IN OUTPUT REG
1256  006732  062700  000260                  ADD     #260,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,TYPOUT       ;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,TYPOUT       ;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,TYPOUT       ;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     #4,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     #177760,R0      ;GET RID OF JUNK
1293  007046  062700  000260                ADD     #260,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     RPERR8,SAVREG+12        ;SET UP RHDS1
1302  007072  013737  012734  012656          MOV     RPERR8+2,SAVREG+14      ;SET UP RHER1
1303  007100  013737  012736  012702          MOV     RPERR8+4,SAVREG+40      ;SET UP RHER2
1304  007106  013737  012740  012704          MOV     RPERR8+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 ADR8
1307  007130  062737  000010  007146          ADD     #10,IRHCS2              ;POINT TO RHCS2 ADR8
1308  007136  017737  000004  012652          MOV     @IRHCS2,SAVREG+10       ;GET RHCS2
1309  007144  000207                          RTS     PC                      ;RETURN
1310  007146  000000                  IRHCS2: 0                       ;CONTAINS RHCS2 ADR8
1311
1312                                          ;THIS ROUTINE TYPES OUT THE CS1,CS2,DS1,ER1,ER2+ER3 REGISTERS
1313
1314  007150  004537  005166          RHREG8: JSR     R5,TYPOUT       ;GO TYPE THE RH HEADER
1315  007154  012143                          RHREGHD
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#XX,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  CKCLK:  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  MTITLE: ,ASCII  <15><12>/RP04 FORMATTER PROGRAM/<15><12>
1376  007474  043040  051117  046515
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 # /
1386  007554  020105  020043     000
1387
1388  007561     075  000060          MDRVD:  ,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  MOFFLN: ,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  MUSDR:  .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  MMODE:  .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  MMECK:  .ASCIZ  /MECK 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  MSIZE:  .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 (18 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  MCLIMT: ,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  MTLIMT: ,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  000523
1500  010526  000012
1501
1502  010530  005015  051412  046105  MSELP:  ,ASCII  <15><12><12>/SELECT DATA PATTERN:/<15><12>
1503  010536  041505  020124  040504
1504  010544  040524  050040  052101
1505  010552  042524  047122  000472
1506  010560     012
1507  010561     050  024460  055040          ,ASCII  /(0) ZERO'S/<15><12>
1508  010566  051105  023517  000523
1509  010574     012
1510  010575     050  024461  047440          ,ASCII  /(1) ONE'S/<15><12>
1511  010602  042516  051447  005015
1512  010610  031060  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  MPATD:  ,ASCIZ  /=>ORST CASE/
1517  010640  041440  051501  000105
1518
1519  010646  005015  051412  040524  MSFOU:  ,ASCIZ  <15><12><12>/STARTING FORMAT ON DRIVE #/
1520  010654  052122  047111  020107
1521  010662  047506  046522  052101
1522  010670  047440  020116  051104
1523  010676  053111  020103  000043
1524
1525  010704  005015  051412  040524  MSCHK:  ,ASCIZ  <15><12><12>/STARTING CHECK ON DRIVE #/
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  MCYLTK: ,ASCIZ  /000,00/<15><12>
1532  010746  006460  000012
1533
1534  010752  005015  043012  051117  MFCMPT: ,ASCIZ  <15><12><12>/FORMAT COMPLETE/<15><12>
1535  010760  040515  020124  047503
1536  010766  050116  042514  042524
1537  010774  005015     000
1538
1539  010777     015  005012  044103  MCCMPT: ,ASCIZ  <15><12><12>/CHECK COMPLETE/<15><12>
1540  011004  041505  020113  047503
1541  011012  050116  042514  042524
1542  011020  005015     000
1543
1544  011023     015  005012  046111  MER1:   ,ASCIZ  <15><12><12>/ILLEGAL SUBSYSTEM INTERRUPT/<15><12><12>
1545  011030  042514  040507  020114
1546  011036  052523  051502  051531
1547  011044  042524  020116  047111
1548  011052  042524  051122  050125
1549  011060  006524  005012     000
1550
1551  011065     015  005012  046111  MER2:   ,ASCIZ  <15><12><12>/ILLEGAL DRIVE INTERRUPT/<15><12><12>
1552  011072  042514  040507  020114
1553  011100  051104  053111  020103
1554  011106  047111  042524  051122
1555  011114  050125  006524  005012
1556  011122     000
1557
1558  011123     015  005012  040515  MER3:   ,ASCIZ  <15><12><12>/MASSBUS 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  MER4:   ,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  MER5:   .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  MER6:   .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  MER6A:  .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  MER6B:  .ASCIZ  <15><12><12>/DRIVE ERROR/<15><12>
1595  011400  042526  042440  051122
1596  011406  051117  005015     000
1597
1598  011413     015  005012  051127  MER6C:  .ASCIZ  <15><12><12>/WRITE ERROR/<15><12>
1599  011420  052111  020105  051105
1600  011426  047522  006522  000012
1601
1602  011434  005015  042012  051511  MER6D:  .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          .ASCII  /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  MER7:   .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  MER8:   ,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  /RHAS= /
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  MRMSC1: .ASCIZ  <15><12>/RMEC1=/
1679  012126  030583  000075
1680
1681  012132  005015  044122  041505  MRMEC2: .ASCIZ  <15><12>/RMEC2=/
1682  012140  036462     000
1683
1684  012143     015  020012  044122  MREGMD: .ASCIZ  <15><12>/ RMCS1   RMCS2   RMDS1   RMER1   RMER2   RMLR3/<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  MFTERM: .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  041505  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  MBDFMT: .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  MHDCMP: .ASCIZ  <15><12><12>/SOFTWARE HEADER COMPARE ERROR/<15><12>
  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
  1737
  1738
  1739
  1740                                  ;LISTED BELOW ARE LOCATIONS CONTAINING
  1741                                  ;PARAMETERS USED BY THE FORMAT OPERATION
  1742
  1743          012540                          .EVEN
  1744
  1745  012540  000000                  USEL:   0               ;CONTAINS # OF UNIT SELECTED
  1746  012542  000000                  SOFSW:  0               ;CONTENTS ARE FOR SOFTWARE DECISIONS
  1747  012544  000000                  MODE:   0               ;'FORMAT & VERIFY' OR 'CHECK' MODE INDICATOR
  1748  012546  000000                  ECYL:   0               ;ENDING CILINDER
  1749  012550  000000                  SCYL:   0               ;STARTING CYLINDER
  1750  012552  000000                  ETRK:   0               ;ENDING TRACK
  1751  012554  000000                  STRK:   0               ;STARTING TRACK
  1752  012556  000000                  TTRKS:  0               ;TOTAL # OF TRACKS TO BE FORMATTED
  1753  012560  000000                  TTRKSC: 0               ;TOTAL # OF TRACKS COUNTER
  1754  012562  000000                  TRKCNT: 0               ;COUNTS TRKS FROM 0-10 PER CYL
  1755  012564  000000                  PSEL:   0               ;CONTAINS PATTERN SELECTED
  1756  012566  000000                  PATA:   0               ;PATTERN TO BE WRITTEN ON DISK
  1757  012570  000000                  PATB:   0               ;DITTO
  1758  012572  000000                  TEMP:   0               ;UTILITY STORAGE
  1759  012574  000000                  RETRY:  0               ;MAINTAINS # OF RETRIES MADE
  1760  012576  000000                  SAVSEC: 0               ;CONTAINS LAST BAD SECTOR ON FORMAT
  1761  012600  000000                  SAVWC:  0               ;CONTAINS WC FOR REMAINING SECTORS ON ERROR
  1762  012602  000000                  WC:     0               ;20 OR 22 SECTOR TRACK SIZE (IN WORDS)
  1763  012604  000000                  MWC:    0               ;2'S COMPLEMENT OF 'WC'
  1764  012606  000000                  HEDERR: 0               ;POSITIONING ERROR DURING FORMAT INDICATOR
  1765  012610  000000                  SEC20:  0               ;20 OR 22 SECTOR MODE INDICATOR
  1766                                                          ;0 = 20 SECTOR MODE
  1767                                                          ;1'S = 22 SECTOR MODE
  1768  012612  000000                  MAXSEC: 0               ;MAXIMUM SECTOR ADDRESS (FOR EITHER 20 OR 22 SECTOR
  1769                                                          ;FORMAT)
  1770  012614  000000                  LOCTRK: 0               ;TEMP STORAGE FOR TRACK CONVERSION FOR TYPEOUT
  1771  012616  000000                  LOCCYL: 0               ;TEMP STORAGE FOR CYL CONVERSION FOR TYPEOUT
  1772  012620  000001                  NULCNT: 1               ;NULL (FILLER) CHARACTER COUNT FOR TYPEOUTS
  1773
  1774  012622  000000                  RBUF:   0               ;THIS LOCATION CONTAINS CYL ADRS ON HEADER CK
  1775  012624  000000                          0               ;THIS LOCATION CONTAINS TRK-SEC ADRS ON HEADER CK
  1776
  1777  012626  000000                  CPTR3:  0               ;LEAVE SPACE FOR LEADING ZEROS
  1778  012630  000000                  CPTR2:  0
  1779  012632  000000                  CPTR1:  0
  1780  012634  000000                  CPTR:   0               ;TTY INPUT BUFFER
  1781          012642                          .+.+4           ;MOVE POINTER TO END OF BUFFER
  1782
```

```
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               ;RHSN
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
                                                      ;BIT 15 = 1: ERROR OCCURED
                                                      ;BIT 07 = 1: DONE
                                                      ;BIT 14-10 AND BIT 06-03
                                                      ;INDICATE TYPE OF ERROR
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828                          .SBTTL  RH11/RP04 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"
```

```
                                        ;*RPERRS   = RHDS1
                                        ;*RPERRS+2 = RHER1
                                        ;*RPERRS+4 = RHER2
                                        ;*RPERRS+6 = RHER3

012732  000000          RPERRS: .WORD   0
012734  000000                  .WORD   0
012736  000000                  .WORD   0
012740  000000                  .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

012742  000          DRVACT: .BYTE   0                ;DRIVE 0
012743  000                  .BYTE   0                ;DRIVE 1
012744  000                  .BYTE   0                ;DRIVE 2
012745  000                  .BYTE   0                ;DRIVE 3
012746  000                  .BYTE   0                ;DRIVE 4
012747  000                  .BYTE   0                ;DRIVE 5
012750  000                  .BYTE   0                ;DRIVE 6
012751  000                  .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

012752  000          DRVSTA: .BYTE   0                ;DRIVE 0
012753  000                  .BYTE   0                ;DRIVE 1
012754  000                  .BYTE   0                ;DRIVE 2
012755  000                  .BYTE   0                ;DRIVE 3
012756  000                  .BYTE   0                ;DRIVE 4
012757  000                  .BYTE   0                ;DRIVE 5
012760  000                  .BYTE   0                ;DRIVE 6
012761  000                  .BYTE   0                ;DRIVE 7

                        ;*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.

012762  000000          DRVTYP: .WORD   0                ;DRIVE 0
012764  000000                  .WORD   0                ;DRIVE 1
012766  000000                  .WORD   0                ;DRIVE 2
012770  000000                  .WORD   0                ;DRIVE 3
012772  000000                  .WORD   0                ;DRIVE 4
012774  000000                  .WORD   0                ;DRIVE 5
012776  000000                  .WORD   0                ;DRIVE 6
013000  000000                  .WORD   0                ;DRIVE 7

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

```
                                ;*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 BIT#0 FOR DRIVE 0.

                                ;*RP04 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
                                ;*ACTDRV=0 IMPLIES DRIVER IS INACTIVE
                                ;*ACTDRV>0 IMPLIES DRIVER IS ACTIVE

                                ;*SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
                                ;*ACTSTR=0 IMPLIES SOFTWARE TIMER ROUTINE IS INACTIVE
                                ;*ACTSTR>0 IMPLIES SOFTWARE TIMER ROUTINE IS ACTIVE

                                ;*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

                                ;*LOOK AHEAD COUNT (LACNT=8 BYTES)
                                ;*LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED

                                ;*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.
```

```
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 (DTUW=1 WORD)
1969                                        ;*DTUW<0 IMPLIES NO DATA TRANSFER UNDERWAY
1970                                        ;*DTUW=+N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
1971
1972  013054  177777                DTUW:   ,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 (MCPEMX=1 WORD)
1989
1990  013066  000003                MCPEMX: ,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
```

```
                                ;*MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)
013100  001000                  MXDLTA: .WORD   8.*64.
                                ;*MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)
013102  000200                  MNDLTA: .WORD   2*64.
                                ;*MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWNDW=1 WORD)
013104  000005                  MXWNDW: .WORD   5

                                ;*DEFINITIONS OF THE RH11/RP04 ADDRESS INDEXES

000000                          RHCS1=0                 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG, 00)
000002                          RHWC=2                  ;WORD COUNT REGISTER (NOT A DRIVE REG)
000004                          RHBA=4                  ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
000006                          RHDA=6                  ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG, 05)
000010                          RHCS2=10                ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
000012                          RHDS1=12                ;DRIVE STATUS REGISTER (DRIVE REG #1)
000014                          RHER1=14                ;ERROR REGISTER #1 (DRIVE REG, 02)
000016                          RHAS=16                 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG, 04)
000020                          RHLA=20                 ;LOOK AHEAD REGISTER (DRIVE REG, 07)
000022                          RHDB=22                 ;DATA BUFFER REGISTER (NOT A DRIVE REG,)
000024                          RHMR=24                 ;MAINTAINABILITY REGISTER (DRIVE REG, 03)
000026                          RHDT=26                 ;DRIVE TYPE REGISTER (DRIVE REG, 06)
000030                          RHSN=30                 ;SERIAL NUMBER REGISTER (DRIVE REG, 10)
000032                          RHOF=32                 ;OFFSET REGISTER (DRIVE REG, 11)
000034                          RHCA=34                 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG, 12)
000036                          RHCC=36                 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG, 13)
000040                          RHER2=40                ;ERROR REGISTER #2 (DRIVE REG, 14)
000042                          RHER3=42                ;ERROR REGISTER #3 (DRIVE REG, 15)
000044                          RHEC1=44                ;ECC POSITION REGISTER (DRIVE REG, 16)
000046                          RHEC2=46                ;ECC PATTERN REGISTER (DRIVE REG,  17)

                                ;*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 DRVINI
                                ;
                                ;*CALL
                                ;
                                ;*      JSR     PC,RPINIT
                                ;*      RETURN
                                ;
013106  013746  177776          RPINIT: MOV     @#PS,-(SP)      ;SAVE PROCESSOR STATUS
013112  013737  013074  177776          MOV     RPVEC+2,@#PS    ;SET PROCESSOR STATUS TO RH11/RP04 LEVEL
013120  004037  020236                  JSR     R0,$AVR1S       ;GO SAVE R1-R5
013124  004737  017774                  JSR     PC,CLRQUE       ;CLEAR ALL REQUEST QUEUES
013130  012701  012732                  MOV     #RPERRS,R1      ;FIRST ADDRESS TO BE CLEARED
013134  012702  013032                  MOV     #SEEKFG,R2      ;LAST ADDRESS TO BE CLEARED
013140  005021                  10:     CLR     (R1)+           ;CLEAR
013142  020102                          CMP     R1,R2           ;ARE WE DONE?
013144  101775                          BLOS    10              ;BRANCH IF NO
013146  012702  013054                  MOV     #DTUN,R2        ;LAST ADDRESS
013152  012721  177777          20:     MOV     #-1,(R1)+       ;INITIALIZE
013156  020102                          CMP     R1,R2           ;DONE?
013160  101774                          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      7$
2058  013204  005201                  4$:     INC     R1              ;MOVE TO THE NEXT DRIVE
2059  013206  042701  177770                  BIC     #^C7,R1         ;DON'T LET THE DRIVE NUMBER GET TO BIG
2060  013212  001371                          BNE     3$              ;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)+,@#PS      ;RESTORE THE PROCESSOR STATUS
2066  013240  000207                          RTS     PC              ;BYE-BYE
2067  013242  105061  012752          7$:     CLRB    DRVSTA(R1)      ;SET THE DRIVE STATUS TO OFFLINE
2068  013246  000756                          BR      4$              ;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 NOL, 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                                  ;+      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
2078                                  ;+      MOV     RPADR,R4        ;UNIBUS ADDRESS OF RH11/RP04 (RHCS1)
2079                                  ;+      JSR     R0,DRVINT       ;CALLED BY A JSR
2080                                  ;+      RETURN1                 ;ERROR OCCURRED (PARITY)
2081                                  ;+      RETURN2                 ;NORMAL RETURN
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     DRVTYP(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     1$              ;NO---BRANCH
2092  013314  004737  017430                  JSR     PC,SET.IE       ;GO SET "IE" WITHOUT A "TRE"
2093  013320  000462                          BR      6$              ;LEAVE THIS ROUTINE
2094  013322  004037  017036          1$:     JSR     R0,RD,RP        ;READ THE DRIVE TYPE REG.
2095  013326  000026                          RHDT
2096  013330  013470                          7$                      ;ERROR RETURN ADDRESS
2097  013332  012605                          MOV     (SP)+,R5        ;PUT DRIVE TYPE IN R5
2098  013334  010563  012762                  MOV     R5,DRVTYP(R3)   ;SAVE THE DRIVE TYPE
2099  013340  022705  020020                  CMP     #20020,R5       ;IS IT A SINGLE PORT RP04?
2100  013344  001403                          BEQ     2$              ;BRANCH IF YES
2101  013346  022705  024020                  CMP     #24020,R5       ;IS IT A DUAL PORT RP04?
2102  013352  001043                          BNE     5$              ;BRANCH IF NO
2103  013354  012746  000121          2$:     MOV     #121,-(SP)      ;DO A "READ-IN PRESET"
2104  013360  004037  017174                  JSR     R0,WRT,RP
2105  013364  000000                          RHCS1
2106  013366  013470                          7$
```

```
2107  013370  012746  010000                  MOV     #BIT12,-(SP)    ;SET PMI22=1
2108  013374  004037  017174                  JSR     R0,@RI,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     #^C<BIT12!BIT09!BIT07!BIT06>,R5
2124  013450  001004                          BNE     58              ;BRANCH IF MOL, DPR, DRY, OR VV IS CLEAR
2125  013452  112761  000001  012752          MOVB    #1,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                                  ;*      JSR     R0,@#RP04       ;CALL THE RP04 DRIVER
2137                                  ;*      PNTADR                  ;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK
2138                                  ;*      RETURN1                 ;RETURN HERE IF QUEUE IS FULL
2139                                  ;*      RETURN2                 ;RETURN HERE IF REQUEST IS IN QUEUE OR THERE
2140                                                                  ;IS AN ERROR CONDITION
2141
2142  013476  013746  177776          RP04:   MOV     @#PS,-(SP)      ;SAVE THE CALLING STATUS
2143  013502  013737  013674  177776          MOV     RPVEC+2,@#PS    ;DON'T ALLOW ANY RP04 INTERRUPTS
2144  013510  112737  000001  013006          MOVB    #1,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     #BIT15!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
```

```
013602  000421                          BR      5$                      ;QUEUE IS FULL
013604  122762  000103  000002          CMPB    #103,2(R2)              ;IS THIS REQ. FOR AN UNLOAD?
013612  001003                          BNE     2$                      ;BR IF NO
013614  112761  177777  013010          MOVB    #-1,ULDFLG(P1)          ;SET THE "UNLOAD IN QUEUE" FLAG
013622  105761  012742          2$:     TSTB    DRVACT(R1)              ;IS THIS DRIVE ACTIVE?
013626  001006                          BNE     4$                      ;BR IS YES
013630  004737  013672                  JSR     PC,OPT                  ;CALL THE OPTIMIZER
013634  000403                          BR      4$
013636  052762  120000  000016  3$:     BIS     #BIT15!BIT13,16(R2)     ;SET THE "UNLOAD IN QUEUE" ERROR FLAG
013644  005720                  4$:     TST     (R0)+
013646  004037  020256          5$:     JSR     R0,GETR15               ;RESTORE R1-R5
013652  105037  013006                  CLRB    ACTDRV                  ;CLEAR "ACTIVE DRIVER" FLAG
013656  012637  177776                  MOV     (SP)+,@#PS              ;RETURN "PS" TO USER LEVEL
013662  000200                          RTS     R0                      ;RETURN TO CALLER
013664  004737  014674          6$:     JSR     PC,C17                  ;GO HANDLE THE PARITY ERROR
013670  000765                          BR      4$

                                ;*OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
                                ;
                                ;*CALL
                                ;*      MOV     @DRVNUM,R1              ;DRIVE NUMBER TO R1
                                ;*      JSR     PC,OPT                  ;SETUP A COMMAND
                                ;
013672  004037  020236          OPT:    JSR     R0,SAVR15               ;SAVE R1-R5
013676  013746  177776                  MOV     @#PS,-(SP)              ;SAVE PROC. STATUS
013702  146137  013056  013004          BICB    ATABIT(R1),SRCHNT       ;CLEAR "SEARCH WAIT" KEY
013710  004737  020152                  JSR     PC,GETREQ               ;GET "DPB" POINTER OF REQUEST
013714  005702                          TST     R2                      ;IS THERE A REQUEST IN QUEUE?
013716  001444                          BEQ     6$                      ;NO--BRANCH TO EXIT
013720  105761  012752                  TSTB    DRVSTA(R1)              ;IS DRIVE ONLINE?
013724  003006                          BGT     1$                      ;YES--BRANCH
013726  004737  020174                  JSR     PC,POPQUE               ;NO--REMOVE REQUEST FROM QUEUE
013732  052762  140000  000016          BIS     #BIT15!BIT14,16(R2)     ;SET ERROR BIT OF STATUS/ERROR INDICATOR
013740  000426                          BR      5$                      ;BRANCH TO EXIT
013742  122762  000150  000002  1$:     CMPB    #150,2(R2)              ;IS THE REQUEST FOR I/O?
013750  002403                          BLT     2$                      ;YES--BRANCH
013752  004737  014252                  JSR     PC,CI4                  ;CALL THE COMMAND INITIATOR
013756  000417                          BR      5$                      ;BRANCH TO EXIT
013760  005737  013054          2$:     TST     DTUN                    ;DATA TRANSFER UNDERWAY?
013764  002012                          BGE     4$                      ;YES--GO START A SEARCH
013766  005737  013032                  TST     SEEKFG                  ;DO IMPLIED SEEKS?
013772  100404                          BMI     3$                      ;YES---BRANCH
013774  004037  015122                  JSR     R0,LA                   ;NO--DO LOOK AHEAD
014000  000406                          BR      5$                      ;RETURN HERE ON A PARITY ERROR
014002  000403                          BR      4$                      ;GO START A SEARCH
014004  004737  014036          3$:     JSR     PC,CI1                  ;START A DATA TRANSFER
014010  000402                          BR      5$
014012  004737  014144          4$:     JSR     PC,CI3                  ;START A SEARCH
014016  012637  177776          5$:     MOV     (SP)+,@#PS              ;RESTORE PROC. STATUS
014022  004037  020256                  JSR     R0,GETR15               ;RESTORE R1-R5
014026  000207                          RTS     PC
014030  004737  017430          6$:     JSR     PC,SET,IE               ;SET "IE" WITHOUT A "TRE"
014034  000770                          BR      5$                      ;EXIT THE ROUTINE
```

```
2215                                    ;*COMMAND INITIATOR
2216                                    ;
2217                                    ;*CALL
2218                                    ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER
2219                                    ;*      MOV     #DPB,R2         ;ADDRESS OF DPB
2220                                    ;*      JSR     PC,CI?  ;CI?= CI1,CI3, OR CI4
2221                                    ;*                              ;WHERE:
2222                                    ;*                              ;CI1=DATA TRANSFER
2223                                    ;*                              ;CI2=SEARCH REQUESTED BY DATA XFER
2224                                    ;*                              ;CI4=NOT DATA TRANSFER
2225
2226  014036  004737  020174       CI1:    JSR     PC,POPQUE       ;REMOVE REQUEST FROM "DRIVES WAIT" QUEUE
2227  014042  010237  013002               MOV     R2,IRWSWT       ;PUT REQ. IN TRANSFER WAIT QUEUE
2228  014046  010203               CI2:    MOV     R2,R3           ;DPB ADDRESS TO R3
2229  014050  013704  013070               MOV     RPADR,R4        ;RHC81 ADDRESS
2230  014054  010164  000010               MOV     R1,RHC82(R4)    ;SELECT DRIVE
2231  014060  062703  000004               ADD     #4,R3           ;DESIRED WORD COUNT
2232  014064  062704  000002               ADD     #2,R4           ;RHWC 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                       CI7                     ;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                       CI7
2243  014120  016246  000002               MOV     2(R2),-(SP)     ;LOAD "COMMAND+GO", "A17&A16", AND "PSEL"
2244  014124  004037  017174               JSR     R0,WRT,RP
2245  014130  000000                       RHC81
2246  014132  014674                       CI7
2247  014134  010137  013056               MOV     R1,DTUW         ;SET "DATA TRANSFER UNDERWAY"
2248  014140  000137  014630               JMP     CI5
2249  014144  013704  013070       CI3:    MOV     RPADR,R4        ;RHC81 ADDRESS
2250  014150  010164  000010               MOV     R1,RHC82(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                       CI7
2255  014170  116203  000010               MOVB    10(R2),R3       ;PICKUP SECTOR ADDRESS
2256  014174  163703  013104               SUB     MXWNDW,R3       ;BACKUP BY MAX, SEARCH FOR I/O WINDOW
2257  014200  002002                       BGE     1$
2258  014202  062703  000026               ADD     #22.,R3
2259  014206  010346               1$:     MOV     R3,-(SP)        ;COMBINE THE ADJUSTED SECTOR WITH
2260  014210  116266  000011  000001       MOVB    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                       CI7
2264  014226  012746  000131               MOV     #131,-(SP)      ;START A SEARCH
2265  014232  004037  017174               JSR     R0,WRT,RP
2266  014236  000000                       RHC81
2267  014240  014674                       CI7
2268  014242  156137  013056  013004       BISB    ATABIT(R1),SRCHWT       ;SET "SEARCH WAIT" KEY
```

```
 2269  014250  000567                          BR      CI5
 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    #131,R3         ;IS IT A SEARCH COMMAND?
 2274  014272  001007                          BNE     1$              ;BRANCH IF NO
 2275  014274  016246  000010                  MOV     10(R2),-(SP)    ;LOAD DESIRED TRACK & SECTOR
 2276  014300  004037  017174                  JSR     R0,#RT,RP
 2277  014304  000006                          RHDA
 2278  014306  014674                          CI7
 2279  014310  000403                          BR      2$              ;GO LOAD CYLINDER
 2280  014312  122703  000105          1$:     CMPB    #105,R3         ;IS IT A SEEK COMMAND
 2281  014316  001007                          BNE     3$              ;BRANCH IF NO
 2282  014320  016246  000012          2$:     MOV     12(R2),-(SP)    ;LOAD DESIRED CYLINDER
 2283  014324  004037  017174                  JSR     R0,#RT,RP
 2284  014330  000034                          RHCA
 2285  014332  014674                          CI7
 2286  014334  000566                          BR      CI6
 2287  014336  122703  000115          3$:     CMPB    #115,R3         ;IS IT AN "OFFSET" COMMAND?
 2288  014342  001013                          BNE     4$              ;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,#RT,RP       ;REGISTER (RHOF)
 2294  014364  000032                          RHOF
 2295  014366  014674                          CI7
 2296  014370  000530                          BR      CI6             ;GO START THE COMMAND
 2297  014372  122703  000107          4$:     CMPB    #107,R3         ;IS IT A "RECALIBRATE" COMMAND?
 2298  014376  001525                          BEQ     CI6             ;BRANCH IF YES
 2299  014400  122703  000117                  CMPB    #117,R3         ;IS IT A RETURN TO CENTER?
 2300  014404  001522                          BEQ     CI6             ;BRANCH IF YES
 2301  014406  122703  000103                  CMPB    #103,R3         ;IS IT AN "UNLOAD" COMMAND?
 2302  014412  001016                          BNE     5$              ;BRANCH IF NO
 2303  014414  112761  000001  012742          MOVB    #1,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    #1,ULDFLG(R1)   ;SET "UNLOAD IN PROGRESS" FLAG
 2306  014434  010346                          MOV     R3,-(SP)        ;START THE "UNLOAD" COMMAND
 2307  014436  004037  017174                  JSR     R0,#RT,RP
 2308  014442  000000                          RHCS1
 2309  014444  014674                          CI7
 2310  014446  000207                          RTS     PC              ;RETURN TO USER
 2311  014450  122703  000143          5$:     CMPB    #143,R3         ;IS IT A "SET FORMAT" COMMAND?
 2312  014454  001014                          BNE     6$              ;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,#RT,RP       ;LOAD "FMT22", "ECI", AND/OR "HCI",
 2318  014500  000032                          RHOF
 2319  014502  014674                          CI7
 2320  014504  000436                          BR      12$
 2321  014506  122703  000141          6$:     CMPB    #141,R3         ;IS IT A "GET REGISTER" COMMAND?
 2322  014512  001023                          BNE     10$             ;BRANCH IF NO
```

```
2323  014514  016203  000006          78$:    MOV     6(R2),R3        ;POINTS TO 1ST ADDRESS OF WHERE
2324                                            ;TO PUT THE REGISTER(S)
2325  014520  116237  000010  014536          MOVB    10(R2),R0       ;INIT. THE INDEX FOR THE FIRST REG.
2326  014526  116205  000011                  MOVB    11(R2),R5       ;INDEX OF LAST REG. TO MOVE
2327  014532  004037  017036          88$:    JSR     R0,RD,RP        ;READ RP04 REGISTER
2328  014536  000000                  98$:    RHCS1                   ;INDEX OF REG. TO READ
2329  014540  014674                          CI7
2330  014542  012623                          MOV     (SP)+,(R3)+     ;GET THE CONTENTS OF RH11/RP04 REG.
2331  014544  023705  014536                  CMP     R0,R5           ;LAST REG. BEEN READ?
2332  014550  001414                          BEQ     12$             ;GET OUT IF YES
2333  014552  062737  000002  014536          ADD     #2,R0           ;INCREASE THE INDEX BY 2
2334  014560  000764                          BR      8$              ;LOOP--MORE TO READ
2335  014562  122703  000145          108$:   CMPB    #145,R3         ;IS IT A "SELECT DRIVE" COMMAND?
2336  014566  001405                          BEQ     12$             ;BRANCH IF YES
2337  014570  010346                  118$:   MOV     R3,-(SP)        ;LOAD THE COMMAND
2338  014572  004037  017174                  JSR     R0,WRT,RP
2339  014576  000000                          RHCS1
2340  014600  014674                          CI7
2341  014602  004737  020174          128$:   JSR     PC,POPQUE       ;REMOVE REQ. FROM QUEUE
2342  014606  052762  000200  000016          BIS     #BIT07,16(R2)   ;SET THE "DONE" BIT
2343  014614  005737  013030                  TST     SAVEFG          ;SAVE THE RH11/RP04 REGISTERS?
2344  014620  100002                          BPL     13$             ;BRANCH IF NO
2345  014622  004737  017336                  JSR     PC,SVRH11       ;YES--GO SAVE THE REGISTERS
2346  014626  000207                  138$:   RTS     PC              ;RETURN TO USER
2347  014630  006301                  CIS:    ASL     R1
2348  014632  012761  001750  013034          MOV     #1000.,TIMER(R1)        ;SET A ONE SECOND TIMER
2349  014640  006201                          ASR     R1
2350  014642  112761  000001  012742          MOVB    #1,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      CIS
2357  014666  105761  012742          CI7A:   TSTB    DRVACT(R1)              ;IS THE DRIVE ACTIVE?
2358  014672  001405                          BEQ     CI7B                    ;BRANCH IF NO
2359  014674  012762  104000  000016  CI7:    MOV     #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     #111,-(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     18              ;IN PROGRESS CLEAR ALL OF THE FLAGS
2370  014744  005037  013002                  CLR     TRNS+T
2371  014750  012737  177777  013054          MOV     #-1,DTUW
2372  014756  000207                  18:     RTS     PC
2373  014760  004037  020236          CI8:    JSR     R0,SAVR1S       ;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  015774  105761  012742          181     TSTB    DRVACT(R1)          ;DRIVE ACTIVE?
2378  016000  001421                          BEQ     38                  ;BRANCH IF NO
2379  016002  013702  013002                  MOV     TRNSWT,R2           ;GET THE "TRANSFER WAIT" QUEUE
2380  016006  020137  013054                  CMP     R1,DTUN             ;DID THIS DRIVE HAVE AN I/O IN PROGRESS?
2381  016012  001402                          BEQ     28                  ;BRANCH IF YES
2382  016014  004737  020152                  JSR     PC,GETREQ           ;GET THE DPB POINTER
2383  016020  004737  017336          281     JSR     PC,SVRH11           ;SAVE RH11/RP04 REGISTERS
2384  016024  052762  102000  000016          BIS     #BIT15!BIT10,16(R2) ;SET "NON-CLEARABLE PARITY" ERROR INDICATOR
2385  016032  012763  177777  013034          MOV     #-1,TIMER(R3)       ;STOP THE TIMER
2386  016040  105061  012742                  CLRB    DRVACT(R1)          ;SET "DRIVE ACTIVE" TO IDLE
2387  016044  105061  013010          381     CLRB    ULDFLG(R1)          ;CLEAR UNLOAD FLAG
2388  016050  005201                          INC     R1                  ;MOVE TO THE NEXT DRIVE
2389  016052  062703  000002                  ADD     #2,R3
2390  016056  042701  177770                  BIC     #^C7,R1
2391  016062  001344                          BNE     18                  ;BRANCH IF MORE DRIVES
2392  016064  012737  177777  013054          MOV     #-1,DTUN            ;NO DATA TRANSFERS UNDERWAY
2393  016072  005037  013002                  CLR     TRNSWT              ;CLEAR THE "TRANSFER WAIT" QUEUE
2394  016076  004737  017774                  JSR     PC,CLRQUE           ;CLEAR ALL OF THE REQUEST QUEUES
2395  016102  012764  000040  000010          MOV     #BIT05,RMCS2(R4)    ;DO A MASSBUS INIT.
2396  016110  004737  017430                  JSR     PC,SET.IE           ;SET "IE" WITHOUT "TRE"
2397  016114  004037  020256                  JSR     R0,GETR15           ;RESTORE THE REGISTERS
2398  016120  000207                          RTS     PC                  ;RETURN
2399
2400                                  ;*LOOK AHEAD ROUTINE
2401                                  ;
2402                                  ;*CALL
2403                                  ;*      MOV     #DRVNUM,R1          ;DRIVE NUMBER
2404                                  ;*      MOV     #DPB,R2             ;POINT TO DPB
2405                                  ;*      JSR     R0,LA               ;GO CHECK THE WINDOW
2406                                  ;*      RETURN1                     ;ERROR RETURN
2407                                  ;*      RETURN2                     ;START A SEARCH
2408                                  ;*      RETURN3                     ;START A DATA TRANSFER
2409
2410  016122  013704  013070          LA:     MOV     RPADR,R4            ;GET RMCS1'S ADDRESS
2411  016126  010164  000010                  MOV     R1,RMCS2(R4)        ;SELECT DRIVE
2412  016132  004037  017036                  JSR     R0,RD,RP            ;READ CURRENT CYLINDER
2413  016136  000036                          RHCC
2414  016140  016252                          48                          ;ERROR RETURN ADDRESS
2415  016142  022662  000012                  CMP     (SP)+,12(R2)        ;IS CURRENT CYLINDER=DESIRED
2416                                                                      ;CYLINDER?
2417  016146  001037                          BNE     38                  ;EXIT IF NO
2418  016150  105261  013020                  INCB    LACNT(R1)           ;INCREMENT THE LOOK AHEAD COUNT
2419  016154  126137  013020  013076          CMPB    LACNT(R1),MXLACT    ;EXCEED MAX?
2420  016162  003026                          BGT     28                  ;BRANCH IF YES
2421  016164  116203  000010                  MOVB    10(R2),R3           ;GET DESIRED SECTOR ADDRESS AND
2422  016170  000303                          SWAB    R3                  ;MULT. BY 64--ALIGN WITH
2423  016172  006203                          ASR     R3                  ;LOOK AHEAD REGISTER
2424  016174  006203                          ASR     R3
2425  016176  012737  000340  177776          MOV     #340,@#PS           ;PRIORITY LEVEL "7"
2426  016204  004037  017036                  JSR     R0,RD,RP            ;READ LOOK AHEAD REGISTER
2427  016210  000020                          RHLA
2428  016212  016252                          48
2429  016214  162603                          SUB     (SP)+,R3            ;CALCULATE THE DELTA
2430  016216  002002                          BGE     18
```

```
2431  015220  062703  002600            ADD     #<22.+64.+>,R3  ;MAKE THE DELTA POSITIVE
2432  015224  023703  013100      18:   CMP     MXDLTA,R3       ;CHECK THE DELTA TO SEE
2433  015230  002406                    BLT     38              ;IF IT IS WITHIN THE
2434  015232  023703  013102            CMP     MNDLTA,R3       ;WINDO----IF YES, ZERO
2435  015236  002003                    BGE     38              ;THE LOOK AHEAD COUNT
2436  015240  105061  013020      28:   CLRB    LACNT(R1)       ;AND TAKE THE I/O EXIT
2437  015244  005720                    TST     (R0)+
2438  015246  005720                38: TST     (R0)+           ;ADJUST THE RETURN ADDRESS
2439  015250  000200                    RTS     R0
2440  015252  004737  014674        48: JSR     PC,CI7          ;PROCESS THE ERROR
2441  015256  000200                    RTS     R0              ;TAKE ERROR RETURN
2442
2443                                ;*INTERRUPT SERVICE ROUTINE
2444
2445  015260  112737  000001  013006  ISR:   MOVB    #1,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     DTUN,R1         ;GET "DATA TRANSFER UNDERWAY" INDICATOR
2449  015302  002403                    BLT     18              ;BRANCH IF NO DATA TRANSFER UNDER-AY
2450  015304  004737  015330            JSR     PC,TD           ;CALL TRANSFER DONE
2451  015310  000402                    BR      28              ;EXIT
2452  015312  004737  015456      18:   JSR     PC,SC           ;CALL SPECIAL CONDITIONS
2453  015316  004037  020256      28:   JSR     R0,GETR15       ;RESTORE R1-R5
2454  015322  105037  013006            CLRB    ACTDRV          ;CLEAR "ACTIVE DRIVER" FLAG
2455  015326  000002                    RTI                     ;RETURN
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     #-1,DTUN        ;NO DATA TRANSFERS UNDERWAY
2461  015342  006301                    ASL     R1
2462  015344  012761  177777  013034    MOV     #-1,TIMER(R1)   ;CANCEL TIMEOUT
2463  015352  006201                    ASR     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     #BIT07,16(R2)   ;SET DONE
2467  015372  010164  000010            MOV     R1,RMCS2(R4)    ;SELECT THE DRIVE
2468  015376  004037  017036            JSR     R0,RD,RP        ;TRANSFER ERROR(TRE=1)?
2469  015402  000000                    RMCS1
2470  015404  014674                    CI7
2471  015406  006126                    ROL     (SP)+
2472  015410  100410                    BMI     28              ;BR IF YES
2473  015412  005737  013030            TST     SAVEFG          ;SAVE THE RH11/RP04 REGISTERS?
2474  015416  100002                    BPL     18              ;BRANCH IF NO
2475  015420  004737  017336            JSR     PC,SVRH11       ;YES--SAVE THE REGISTERS
2476  015424  004737  013672      18:   JSR     PC,OPT          ;CALL OPTIMIZER
2477  015430  000456                    BR      SC2             ;SPECIAL CONDITION (ENTRY #2)
2478  015432  052762  100100  000016  28:   BIS     #BIT15!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     #40111,(R4)     ;ISSUE A "DRIVE CLEAR"
2482  015454  000444                    BR      SC2             ;SPECIAL CONDITION (ENTRY #2)
2483
2484                                ;*SPECIAL CONDITION ROUTINE
```

```
2485
2486  015456  005001              SC:   CLR     R1              ;START WITH DRIVE 0
2487  015460  012702  000001            MOV     #1,R2
2488  015464  105761  012752      1$:   TSTB    DRVSTA(R1)      ;NONEXISTENT?
2489  015470  002016                    BGE     2$              ;NO--BRANCH
2490  015472  005201                    INC     R1              ;YES--MOVE TO THE NEXT DRIVE
2491  015474  106302                    ASLB    R2              ;MORE DRIVES?
2492  015476  103372                    BCC     1$              ;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      2$:   MOV     R1,RHCS2(R4)    ;SELECT DRIVE
2500  015532  116405  000001            MOVB    1(R4),R5        ;IS "SC"=1?
2501  015536  100413                    BMI     SC2             ;BRANCH IF YES
2502  015540  012701  000010      SC1:  MOV     #8.,R1          ;DO EIGHT DRIVES
2503  015544  005301              1$:   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     1$              ;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      SC2:  MOV     #3,R1           ;READ RMAS UP TO THREE TIMES
2511  015572  116403  000016      1$:   MOVB    RMAS(R4),R3     ;READ "RMAS"
2512  015576  001011                    BNE     2$              ;BRANCH IF ANY ATA BITS = 1
2513  015600  005301                    DEC     R1              ;COUNT THIS READ
2514  015602  003373                    BGT     1$              ;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                    CIB
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              2$:   CLR     -(SP)           ;PROCESS ALL DRIVES THAT HAVE
2522  015624  110316                    MOVB    R3,(SP)         ;AN "ATA"=1
2523  015626  012703  000001            MOV     #1,R3
2524  015632  005001                    CLR     R1
2525  015634  030316              SC4:  BIT     R3,(SP)         ;ATA=1?
2526  015636  001005                    BNE     SC5             ;YES--BRANCH
2527  015640  005201              SC3:  INC     R1              ;MOVE TO THE NEXT DRIVE
2528  015642  106303                    ASLB    R3
2529  015644  001373                    BNE     SC4             ;BRANCH IF MORE TO CHECK?
2530  015646  005726                    TST     (SP)+           ;CLEAN OFF THE STACK
2531  015650  000207                    RTS     PC              ;RETURN TO USER
2532  015652  023701  013054      SC5:  CMP     DTUN,R1         ;IS THIS DRIVE SETUP FOR I/O?
2533  015656  001002                    BNE     1$              ;NO---BRANCH
2534  015660  005726                    TST     (SP)+           ;YES---CLEAN OFF THE STACK (RMAS)
2535  015662  000622                    BR      TD              ;BRANCH TO "TRANSFER DONE"
2536  015664  105761  012752      1$:   TSTB    DRVSTA(R1)      ;CHECK THE DRIVE STATUS
2537  015670  003030                    BGT     SC6             ;BRANCH IF ONLINE
2538  015672  105761  013018            TSTB    ULDFLG(R1)      ;UNLOAD IN PROGRESS?
```

```
2539  015676  003420                          BLE     28              ;BRANCH IF NO
2540  015700  004737  020152                  JSR     PC,GETREQ       ;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 (DRVIN1)
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     8C11            ;NO -- BRANCH
2548  015732  013705  012734                  MOV     RPERRS+2,R5     ;YES -- PICKUP RHER1 AND
2549  015736  000447                          BR      SC6A            ;GO PROCESS THE ERROR
2550  015740  004737  016366          28:     JSR     PC,SC12         ;SAVE RHDS1, RHER1, RHER2, AND RHER3
2551                                                                  ;ALSO DO A DRVINT
2552  015744  011605                  38:     MOV     (SP),R5         ;PICKUP (RHAS) BEFORE THE ERROR CALL
2553  015746  104005                          ERROR   5               ;REPORT THE ERROR
2554  015750  000733                          BR      8C3             ;GO CHECK FOR MORE ATA'S
2555  015752  006301                  8C6:    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,GETREQ       ;GET THE DPB POINTER FROM THE QUEUE
2559  015770  010364  000016                  MOV     R3,RHAS(R4)     ;CLEAR ATTENTION
2560  015774  010164  000010                  MOV     R1,RHCS2(R4)    ;SELECT DRIVE
2561  016000  004037  017036                  JSR     R0,RD,RP        ;READ THE RP04'S STATUS REG.
2562  016004  000012                          RHDS1
2563  016006  016224                          8C8
2564  016010  011605                          MOV     (SP),R5         ;AND PUT IT IN R5
2565  016012  006126                          ROL     (SP)+           ;WAS THERE AN ERROR?
2566  016014  100115                          BPL     8C9             ;BR IF NO
2567  016016  105761  012742                  TSTB    DRVACT(R1)      ;CHECK THE DRIVE ACTIVE INDICATOR
2568  016022  001522                          BEQ     8C10            ;BRANCH IF IDLE
2569  016024  004037  017036                  JSR     R0,RD,RP        ;READ ERROR REGISTER #1
2570  016030  000014                          RHER1
2571  016032  016224                          8C8
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,WRT,RP
2576  016052  000000                          RHCS1
2577  016054  016224                          8C8
2578  016056  006105          8C6A:   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      8C7
2582  016072  004037  017036          18:     JSR     R0,RD,RP        ;READ DRIVE STATUS REG, #1
2583  016076  000012                          RHDS1
2584  016100  016224                          8C8
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      8C7
```