C 8

```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
```

```
                                                                 I 16
**FILE**ID**DRDRIVER









          DDDDDDDD    RRRRRRRR    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
          DDDDDDDD    RRRRRRRR    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
          DD      DD  RR      RR  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
          DD      DD  RR      RR  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
          DD      DD  RR      RR  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
          DD      DD  RR      RR  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
          DD      DD  RRRRRRRR    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
          DD      DD  RRRRRRRR    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
          DD      DD  RR  RR      DD      DD  RR  RR          II      VV      VV  EE          RR  RR
          DD      DD  RR  RR      DD      DD  RR  RR          II      VV      VV  EE          RR  RR
          DD      DD  RR    RR    DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
          DD      DD  RR    RR    DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
          DDDDDDDD    RR      RR  DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR      ....
          DDDDDDDD    RR      RR  DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR      ....


          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  DRDRIVER - RM03/RM05/RM80/RP07 DISK DRIVER
0000            .IDENT  'V04-001'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ; D. N. CUTLER, LEN KAWELL 23-NOV-77
0000    ;
0000    ;  MODIFIED BY:
0000    ;
0000    ;       V04-001 PRD0112         Paul R. DeStefano       06-Sep-1984
0000    ;               Modify ECC routine to allow for RP07's handling of
0000    ;               HCRC errors as class A errors when HCI is set.
0000    ;
0000    ;               Add sanity check to offset recovery routine to insure
0000    ;               that there is data to be transferred before offset
0000    ;               recovery is performed.
0000    ;
0000    ;       V03-016 RAS0300         Ron Schaefer            27-Apr-1984
0000    ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    ;               devices will have the "node$" prefix.
0000    ;
0000    ;       V03-015 PRD0081         Paul R. DeStefano       19-Mar-1984
0000    ;               For dual ported drives, make sure the port isn't
0000    ;               reseized by the time we come off the I/O fork queue.
0000    ;
0000    ;       V03-014 PRD0048         Paul R. DeStefano       01-Feb-1984
0000    ;               Fix context used in TIMEWAIT macro when referencing
0000    ;               device registers.
0000    ;
0000    ;       V03-013 PRD0036         Paul R. DeStefano       09-Sep-1983
0000    ;               Added EXE$LCLDSKVALID to function decision table.
0000    ;
0000    ;       V03-012 ROW0211         Ralph O. Weber          16-AUG-1983
0000    ;               Change device-dependent UCB definition base from UCB$W_BCR+2
0000    ;               to UCB$K_LCL_DISK_LENGTH.
```

```
0000     58 ;
0000     59 ;       V03-011 WMC0001         Wayne Cardoza           09-Aug-1983
0000     60 ;               Missing G^.
0000     61 ;
0000     62 ;       V03-010 KDM0060         Kathleen D. Morse       14-Jul-1983
0000     63 ;               Replace reference to IPR TODR with call to cpu-dependent
0000     64 ;               routine, EXE$READ_TODR.
0000     65 ;               Add $DEVDEF.
0000     66 ;
0000     67 ;       V03-009 PRD0027         Paul R. DeStefano       17-Jun-1983
0000     68 ;               Modified EXFNC routine to bypass setting of offset mode
0000     69 ;               for RP07's to prevent RP07 microcode hang and system crash.
0000     70 ;
0000     71 ;       V03-008 PRD0023         Paul R. DeStefano       05-May-1983
0000     72 ;               Modified ERROR routine to attempt to clear a drive
0000     73 ;               unsafe condidtion.
0000     74 ;
0000     75 ;       V03-007 PRD53302        Paul R. DeStefano       05-May-1983
0000     76 ;       ECO 02  Modified RETRYERR routine to issue a Drive Clear before
0000     77 ;               retrying a function.  Modified FUNCXT routine to issue
0000     78 ;               a Drive Clear function before releasing the drive.
0000     79 ;
0000     80 ;       V03-006 PRD0018         Paul R. DeStefano       26-Apr-1983
0000     81 ;               Modified FATALERR routine to return SS$_PARITY only for
0000     82 ;               errors that possibly indicate bad media.  All other error
0000     83 ;               conditions which formerly returned SS$_PARITY now return
0000     84 ;               SS$_CNTLERR.
0000     85 ;
0000     86 ;       V03-005 PRD0015         Paul R. DeStefano       26-Apr-1983
0000     87 ;               Modified ECC correction logic so that ECC is only applied
0000     88 ;               when there is single bit ECC correctable error, or if there
0000     89 ;               is a multiple bit ECC correctable error and the error cannot
0000     90 ;               be corrected using retries.
0000     91 ;
0000     92 ;       V03-004 ROW47161        Ralph O. Weber          16-SEP-1982
0000     93 ;       ECO 01  Enhance ECC recovery logic to prevent bytes transfered counts
0000     94 ;               which are not exact multiples of 512 from causing transfer
0000     95 ;               parameters from being incorrectly updated.  Because a non-512-
0000     96 ;               intergal bytes transfered counts indicates an incomplete
0000     97 ;               transfer of the last block, this change also prevents ECC
0000     98 ;               corrections when such bytes transfered counts are encountered.
0000     99 ;
0000    100 ;       V03-003 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    101 ;               Added $DCDEF, $DYNDEF, and $SSDEF.
0000    102 ;
0000    103 ;       V03-002 KTA0100         Kerbey T. Altmann               07-Jun-1982
0000    104 ;               Add code to set UCB$L_MEDIA_ID.
0000    105 ;
0000    106 ;
0000    107 ; RM03/RM05/RM80/RP07 DISK DRIVER
0000    108 ;
0000    109 ; MACRO LIBRARY CALLS
0000    110 ;
0000    111
0000    112         $CRBDEF                                 ;DEFINE CRB OFFSETS
0000    113         $DCDEF                                  ;DEFINE DEVICE CLASSES
0000    114         $DDBDEF                                 ;DEFINE DDB OFFSETS
```

```
0000            $DEVDEF                                 ;DEFINE DEVICE CHARACTERISTICS
0000            $DPTDEF                                 ;DEFINE DPT OFFSETS
0000            $DYNDEF                                 ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000            $EMBDEF                                 ;DEFINE EMB OFFSETS
0000            $IDBDEF                                 ;DEFINE IDB OFFSETS
0000            $IODEF                                  ;DEFINE I/O FUNCTION CODES
0000            $IRPDEF                                 ;DEFINE IRP OFFSETS
0000            $MBADEF                                 ;DEFINE MBA REGISTER OFFSETS
0000            $PRDEF                                  ;DEFINE PROCESSOR REGISTER NUMBERS
0000            $SSDEF                                  ;DEFINE SYSTEM STATUS CODES
0000            $UCBDEF                                 ;DEFINE UCB OFFSETS
0000            $VECDEF                                 ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000
0000    ;
0000    ; LOCAL MACROS
0000    ;
0000    ; EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000    ;
0000
0000            .MACRO  EXFUNC BDST,FCODE
0000                    .IF NB  FCODE
0000                    MOVZBL  #CD'FCODE,R0
0000                    .ENDC
0000                    BSBW    FEX
0000                    .SIGNED_WORD BDST-.-2
0000            .ENDM
0000
0000    ;
0000    ; GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000    ;
0000
0000            .MACRO  GENF FCODE
0000                    CD'FCODE=.-FTAB
0000                    .BYTE   FCODE!RM_CS1_M_GO
0000            .ENDM
0000
0000    ;
0000    ; LOCAL SYMBOLS
0000    ;
0000    ; MASSBUS REGISTER OFFSETS
0000    ;
0000
0000            $DEFINI RM
0000
0000    $DEF    RM_CS1          .BLKL   1               ;DRIVE CONTROL REGISTER
0004            _VIELD  RM_CS1,0,<-                     ; DRIVE CONTROL REGISTER BIT DEFINITIONS
0004                    <GO,,M>,-                       ; GO BIT
0004                    <FCODE,5>-                      ; FUNCTION CODE
0004            >                                       ;
0004    $DEF    RM_DS           .BLKL   1               ;DRIVE STATUS REGISTER
0008            _VIELD  RM_DS,0,<-                      ; DRIVE STATUS REGISTER BIT DEFINITIONS
0008                    <OM,,M>,-                       ; OFFSET MODE
0008                    <,5>,-                          ; RESERVED BITS
0008                    <VV,,M>,-                       ; VOLUME VALID
0008                    <DRY,,M>,-                      ; DRIVE READY
0008                    <DPR,,M>,-                      ; DRIVE PRESENT
0008                    <PGM,,M>,-                      ; PROGRAMMABLE
```

```
0008                      <LST,,M>,-                    ; LAST SECTOR TRANSFERED
0008                      <WRL,,M>,-                    ; DRIVE WRITE LOCKED
0008                      <MOL,,M>,-                    ; MEDIUM ONLINE
0008                      <PIP,,M>,-                    ; POSITIONING IN PROGRESS
0008                      <ERR,,M>,-                    ; COMPOSITE ERROR
0008                      <ATA,,M>-                     ; ATTENTION ACTIVE
0008            >                                       ;
0008    $DEF    RM_ER1              .BLKL   1           ;ERROR REGISTER 1
000C            _VIELD    RM_ER1,0,<-                   ; ERROR REGISTER 1 BIT DEFINITIONS
000C                      <ILF,,M>,-                    ; ILLEGAL FUNCTION
000C                      <ILR,,M>,-                    ; ILLEGAL REGISTER
000C                      <RMR,,M>,-                    ; REGISTER MODIFY REFUSED
000C                      <PAR,,M>,-                    ; PARITY ERROR
000C                      <FER,,M>,-                    ; FORMAT ERROR
000C                      <WCF,,M>,-                    ; WRITE CLOCK FAIL
000C                      <ECH,,M>,-                    ; ECC HARD ERROR
000C                      <HCE,,M>,-                    ; HEADER COMPARE ERROR
000C                      <HCRC,,M>,-                   ; HEADER CRC ERROR
000C                      <AOE,,M>,-                    ; ADDRESS OVERFLOW ERROR
000C                      <IAE,,M>,-                    ; ILLEGAL ADDRESS ERROR
000C                      <WLE,,M>,-                    ; WRITE LOCK ERROR
000C                      <DTE,,M>,-                    ; DRIVE TIMING ERROR
000C                      <OPI,,M>,-                    ; OPERATION INCOMPLETE
000C                      <UNS,,M>,-                    ; DRIVE UNSAFE
000C                      <DCK,,M>-                     ; DATA CHECK ERROR
000C            >                                       ;
000C    $DEF    RM_MR               .BLKL   1           ;MAINTENANCE REGISTER
0010            _VIELD    RM_MR,0,<-                    ; MAINTENANCE REGISTER DEFINITIONS
0010                      <PAR,8>,-                     ; DIAGNOSTIC PARAMETER
0010                      <RTN,7>,-                     ; DIAGNOSTIC ROUTINE NUMBER
0010                      <DM,,M>,-                     ; DIAGNOSTIC MODE
0010            >                                       ;
0010    $DEF    RM_AS               .BLKL   1           ;ATTENTION SUMMARY REGISTER
0014    $DEF    RM_DA               .BLKL   1           ;DESIRED SECTOR/TRACK ADDRESS REGISTER
0018            _VIELD    RM_DA,0,<-                    ; DESIRED ADDRESS FIELD DEFINITIONS
0018                      <SA,5>,-                      ; DESIRED SECTOR ADDRESS
0018                      <,3>,-                        ; RESERVED BITS
0018                      <TA,5>-                       ; DESIRED TRACK ADDRESS
0018            >                                       ;
0018    $DEF    RM_DT               .BLKL   1           ;DRIVE TYPE REGISTER
001C            _VIELD    RM_DT,0,<-                    ; DRIVE TYPE REGISTER FIELD DEFINITIONS
001C                      <DTN,9>,-                     ; DRIVE TYPE NUMBER
001C                      <,2>,-                        ; RESERVED BITS
001C                      <DRQ,,M>-                     ; DRIVE REQUEST REQUIRED
001C            >                                       ;
001C    $DEF    RM_LA               .BLKL   1           ;LOOKAHEAD REGISTER
0020    $DEF    RM_SN               .BLKL   1           ;SERIAL NUMBER REGISTER
0024    $DEF    RM_OF               .BLKL   1           ;OFFSET REGISTER
0028            _VIELD    RM_OF,0,<-                    ; OFFSET REGISTER BIT DEFINITIONS
0028                      <OFF,8>,-                     ; OFFSET VALUE
0028                      <,1>,-                        ; RESERVED
0028                      <SSEI,,M>,-                   ; SKIP SECTOR INHIBIT (RM80)
0028                      <HCI,,M>,-                    ; HEADER COMPARE INHIBIT
0028                      <ECI,,M>,-                    ; ECC INHIBIT (avoid using this bit)
0028                      <FMT,,M>,-                    ; 16-BIT FORMAT
0028                      <,1>,-                        ; RESERVED
0028                      <MTD,,M>,-                    ; MOVE TRACK DESCRIPTOR
```

```
          0028                        <CMO,,M>-                   ; COMMAND MODIFIER
          0028                >                                   ;
          0028  $DEF     RM_DC             .BLKL   1              ;DESIRED CYLINDER ADDRESS
          002C  $DEF     RM_UNUSED         .BLKL   1              ;UNUSED
          0030  $DEF     RM_MR2            .BLKL   1              ;MAINTENANCE REGISTER 2
          0034  $DEF     RM_ER2            .BLKL   1              ;ERROR REGISTER 2
          0038           _VIELD  RM_ER2,3,<-                      ; ERROR REGISTER 2 BIT DEFINITIONS
          0038                   <DPE,,M>,-                       ; DATA PARITY ERROR
          0038                   <,1>,-                           ; RESERVED BIT
          0038                   <SSE,,M>,-                       ; SKIP SECTOR ERROR (RM80)
          0038                   <,1>,-                           ; RESERVED BIT
          0038                   <DVC,,M>,-                       ; DEVICE CHECK ERROR
          0038                   <,2>,-                           ; RESERVED BITS
          0038                   <LBC,,M>,-                       ; LOSS OF BIT CLOCK ERROR
          0038                   <LSC,,M>,-                       ; LOSS OF SYSTEM CLOCK ERROR
          0038                   <IVC,,M>,-                       ; INVALID COMMAND ERROR
          0038                   <OPE,,M>,-                       ; OPERATOR PLUG ERROR
          0038                   <SKI,,M>,-                       ; SEEK INCOMPLETE ERROR
          0038                   <BSE,,M>-                        ; BAD SECTOR ERROR
          0038           >                                        ;
          0038  $DEF     RM_EC1            .BLKL   1              ;ECC POSITION REGISTER
          003C           _VIELD  RM_EC1,0,<<POS,13>>              ; ECC POSITION FIELD
          003C  $DEF     RM_EC2            .BLKL   1              ;ECC PATTERN REGISTER
          0040           _VIELD  RM_EC2,0,<<PAT,11>>              ; ECC PATTERN FIELD
          0040
          0040           $DEFEND RM
          0000
          0000  ;
          0000  ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
          0000  ;
          0000
          0000           $DEFINI UCB
          0000
000000CC  0000  .=UCB$K_LCL_DISK_LENGTH                           ;Establish device-dependent UCB base
          00CC
          00CC  $DEF     UCB$L_DR_SR       .BLKL   1              ;SAVED MBA STATUS REGISTER
          00D0  $DEF     UCB$W_DR_ER2      .BLKW   1              ;SAVED ERROR REGISTER 2
          00D2  $DEF     UCB$W_DR_MR       .BLKW   1              ;MAINTENANCE REGISTER
          00D4  $DEF     UCB$B_DR_SSTS     .BLKB   1              ;SOFTWARE STATUS BYTE
          00D5           _VIELD  DR,0,<-                          ; SOFTWARE STATUS BIT DEFINITIONS
          00D5                   <DCK,,M>,-                       ; DATACHECK IN PROGRESS
          00D5                   <OM,,M>,-                        ; OFFSET MODE
          00D5                   <NOECC,,M>,-                     ; Don't correct with ECC
          00D5                   <DUALPORT,,M>-                   ; Drive has a dualport kit
          00D5                   <ECC_DEFER,,M>, -                ; Flag to indicate that ECC correction
          00D5           >                                        ;  has been deferred until offset
          00D5                                                    ;  retries are exhausted.
          00D5  $DEF     UCB$B_DR_ERL      .BLKB   1              ;ERROR LOGGING REGISTER FOR MED OFFLINE
          00D6  $DEF     UCB$W_DR_OFR      .BLKW   1              ;SAVED OFFSET REGISTER
          00D8  $DEF     UCB$L_DR_BCR      .BLKL   1              ; Saved (longword) MBA byte count reg.
000000DC  00DC  UCB$K_DR_LENGTH=.
          00DC
          00DC           $DEFEND UCB
          0000
          0000  ;
          0000  ; HARDWARE FUNCTION CODES
          0000  ;
          0000   286
00000000  0000   287 F_NOP=0*2                                  ;NO OPERATION
00000004  0000   288 F_SEEK=2*2                                 ;SEEK CYLINDER
00000006  0000   289 F_RECAL=3*2                                ;RECALIBRATE
00000008  0000   290 F_DRVCLR=4*2                               ;DRIVE CLEAR
0000000A  0000   291 F_RELEASE=5*2                              ;RELEASE DRIVE
0000000C  0000   292 F_OFFSET=6*2                               ;OFFSET HEADS
0000000E  0000   293 F_RETCENTER=7*2                            ;RETURN TO CENTERLINE
00000010  0000   294 F_READPRESET=8*2                           ;READ IN PRESET
00000012  0000   295 F_PACKACK=9*2                              ;PACK ACKNOWLEDGE
00000018  0000   296 F_SEARCH=12*2                              ;SEARCH FOR SECTOR
00000018  0000   297 F_SEARCHA=12*2                             ;SEARCH AHEAD FOR SECTOR
0000001C  0000   298 F_DIAGNOSE=14*2                            ;DIAGNOSE DRIVE
00000028  0000   299 F_WRITECHECK=20*2                          ;WRITE CHECK DATA
0000002A  0000   300 F_WRITECHECKH=21*2                         ;WRITE CHECK HEADER AND DATA
00000030  0000   301 F_WRITEDATA=24*2                           ;WRITE DATA
00000032  0000   302 F_WRITEHEAD=25*2                           ;WRITE HEADER AND DATA
00000034  0000   303 F_WRITETRACKD=26*2                         ;WRITE TRACK DESCRIPTOR
00000038  0000   304 F_READDATA=28*2                            ;READ DATA
0000003A  0000   305 F_READHEAD=29*2                            ;READ HEADER AND DATA
0000003C  0000   306 F_READTRACKD=30*2                          ;READ TRACK DESCRIPTOR
00000000  0000   307 F_AVAILABLE=F_NOP                          ;AVAILABLE
          0000   308
          0000   309 ;
          0000   310 ; LOCAL DATA
          0000   311 ;
          0000   312 ; DRIVER PROLOGUE TABLE
          0000   313 ;
          0000   314
          0000   315         DPTAB   -                          ;DEFINE DRIVER PROLOGUE TABLE
          0000   316                 END=DR_END,-               ;END OF DRIVER
          0000   317                 ADAPTER=MBA,-              ;ADAPTER TYPE
          0000   318                 FLAGS=DPT$M_SVP,-          ;SYSTEM PAGE TABLE ENTRY REQUIRED
          0000   319                 UCBSIZE=UCB$K_DR_LENGTH,-          ;UCB size
          0000   320                 NAME=DRDRIVER              ;DRIVER NAME
          0038   321         DPT_STORE INIT                     ;CONTROL BLOCK INIT VALUES
          0038   322         DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\> ;DEFAULT ACP NAME
          003F   323         DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_PACK ;ACP CLASS
          0043   324         DPT_STORE UCB,UCB$B_FIPL,B,8       ;FORK IPL
          0047   325         DPT_STORE UCB,UCB$L_DEVCHAR,L,-    ;DEVICE CHARACTERISTICS
          0047   326                 <DEV$M_FOD-                ; FILES ORIENTED
          0047   327                 !DEV$M_DIR-                ; DIRECTORY STRUCTURED
          0047   328                 !DEV$M_AVL-                ; AVAILABLE
          0047   329                 !DEV$M_ELG-                ; ERROR LOGGING ENABLED
          0047   330                 !DEV$M_SHR-                ; SHAREABLE
          0047   331                 !DEV$M_IDV-                ; INPUT DEVICE
          0047   332                 !DEV$M_ODV-                ; OUTPUT DEVICE
          0047   333                 !DEV$M_RND>                ; RANDOM ACCESS
          004E   334         DPT_STORE UCB,UCB$L_DEVCHAR2,L,-;  DEVICE CHARACTERISTICS
          004E   335                 <DEV$M_NNM>                ; PREFIX NAME WITH "node$"
          0055   336         DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_DISK ;DEVICE CLASS
          0059   337         DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
          005E   338         DPT_STORE UCB,UCB$B_DIPL,B,21      ;DEVICE IPL
          0062   339         DPT_STORE UCB,UCB$B_ERTCNT,B,8     ;ERROR RETRY COUNT
          0066   340         DPT_STORE UCB,UCB$B_ERTMAX,B,8     ;MAX ERROR RETRY COUNT
          006A   341         DPT_STORE REINIT                   ;CONTROL BLOCK RE-INIT VALUES
          006A   342         DPT_STORE DDB,DDB$L_DDT,D,DR$DDT   ;DDT ADDRESS
```

```
                006F         DPT_STORE END                                ;
                0000
                0000    ;
                0000    ; DRIVER DISPATCH TABLE
                0000    ;
                0000
                0000         DDTAB   DR,-                                 ;DRIVER DISPATCH TABLE
                0000                 DR_STARTIO,-                         ;START I/O OPERATION
                0000                 DR_UNSOLNT,-                         ;UNSOLICITED INTERRUPT
                0000                 DR_FUNCTABLE,-                       ;FUNCTION DECISION TABLE
                0000                 0,-                                  ;CANCEL I/O ENTRY POINT
                0000                 DR_REGDUMP,-                         ;REGISTER DUMP ROUTINE
                0000                 <<RM_EC2+4+4+MBA$L_BCR+4+8>+<<5+5+1>*4>>,- ;DIAG BUFF SIZE
                0000                 <<RM_EC2+4+4+MBA$L_BCR+4+8>+<1*4>+<EMB$L_DV_REGSAV>>,- ;ERLG BUFF SI
                0000                 DR_UNIT_INIT                         ;UNIT INITIALIZATION
                0038
                0038    ;
                0038    ; DATA CHECK FUNCTION TRANSLATION TABLE
                0038    ;
                0038
                0038    CHECKTAB:                                         ;
           0A'  0038         .BYTE   CDF_WRITECHECK                       ;WRITE DATA
           0A'  0039         .BYTE   CDF_WRITECHECK                       ;READ DATA
           12'  003A         .BYTE   CDF_WRITECHECKH                      ;WRITE HEADER AND DATA
           12'  003B         .BYTE   CDF_WRITECHECKH                      ;READ HEADER AND DATA
                003C
                003C    ;
                003C    ; DRIVE TYPE DESCRIPTOR TABLE
                003C    ;
                003C    DR_DTDESC:                                        ;
          0014  003C         .WORD   ^X14                                 ; RM03
            06  003E         .BYTE   DT$_RM03                             ;
            20  003F         .BYTE   32                                   ; 32 SECTORS
            05  0040         .BYTE   5                                    ; 5 TRACKS
          0337  0041         .WORD   823                                  ; 823 CYLINDERS
      00020260  0043         .LONG   823*5*32                             ; MAXIMUM BLOCKS
      24A4D003  0047         .LONG   ^X24A4D003                           ; MEDIA IDENT "DR RM03"
      0000000F  004B    DR_DTDESCLEN=.-DR_DTDESC                          ;LENGTH OF DRIVE TYPE DESCRIPTOR
                004B
          0016  004B         .WORD   ^X16                                 ; RM80
            0D  004D         .BYTE   DT$_RM80                             ;
            1F  004E         .BYTE   31                                   ; 31 SECTORS
            0E  004F         .BYTE   14                                   ; 14 TRACKS
          022F  0050         .WORD   559                                  ; 559 CYLINDERS
      0003B3AE  0052         .LONG   559*14*31                            ; MAXIMUM BLOCKS
      24A4D050  0056         .LONG   ^X24A4D050                           ; MEDIA IDENT "DR RM80"
          0017  005A         .WORD   ^X17                                 ; RM05
            0F  005C         .BYTE   DT$_RM05                             ;
            20  005D         .BYTE   32                                   ; 32 SECTORS
            13  005E         .BYTE   19                                   ; 19 TRACKS
          0337  005F         .WORD   823                                  ; 823 CYLINDERS
      0007A2A0  0061         .LONG   823*19*32                            ; MAXIMUM BLOCKS
      24A4D005  0065         .LONG   ^X24A4D005                           ; MEDIA IDENT "DR RM05"
          0022  0069         .WORD   ^X22                                 ; RP07
            07  006B         .BYTE   DT$_RP07                             ;
            32  006C         .BYTE   50                                   ; 50 SECTORS
            20  006D         .BYTE   32                                   ; 32 TRACKS
```

```
    0276  006E           .WORD   630                         ; 630 CYLINDERS
000F6180  0070           .LONG   630*32*50                   ; MAXIMUM BLOCKS
24A50007  0074           .LONG   ^X24A50007                  ; MEDIA IDENT "DR RP07"
          0078
    0000  0078           .WORD   0                           ; END OF TABLE
00000089  007A           .BLKB   DR_DTDESCLEN                ; SPARE DRIVE TYPE SLOT
00000098  0089           .BLKB   DR_DTDESCLEN                ; SPARE DRIVE TYPE SLOT
          0098
          0098   ;
          0098   ; HARDWARE I/O FUNCTION CODE TABLE
          0098   ;
          0098
          0098   FTAB:                                       ;
          0098           GENF    F_NOP                       ;NO OPERATION
          0099           GENF    F_NOP                       ;(NO UNLOAD FUNCTION)
          009A           GENF    F_SEEK                      ;SEEK CYLINDER
          009B           GENF    F_RECAL                     ;RECALIBRATE
          009C           GENF    F_DRVCLR                    ;DRIVE CLEAR
          009D           GENF    F_NOP                       ;(NO RELEASE PORT)
          009E           GENF    F_OFFSET                    ;OFFSET HEADS
          009F           GENF    F_RETCENTER                 ;RETURN HEADS TO CENTERLINE
          00A0           GENF    F_PACKACK                   ;PACK ACKNOWLEDGE
          00A1           GENF    F_SEARCH                    ;SEARCH FOR SECTOR
          00A2           GENF    F_WRITECHECK                ;WRITE CHECK
          00A3           GENF    F_WRITEDATA                 ;WRITE DATA
          00A4           GENF    F_READDATA                  ;READ DATA
          00A5           GENF    F_WRITEHEAD                 ;WRITE HEADER AND DATA
          00A6           GENF    F_READHEAD                  ;READ HEADER AND DATA
          00A7           GENF    F_WRITETRACKD               ;WRITE TRACK DESCRIPTOR
          00A8           GENF    F_READTRACKD                ;READ TRACK DESCRIPTOR
          00A9           GENF    F_AVAILABLE                 ;AVAILABLE
          00AA           GENF    F_WRITECHECKH               ;WRITE CHECK HEADER AND DATA
          00AB           GENF    F_READPRESET                ;READ IN PRESET
          00AC           GENF    F_DIAGNOSE                  ;DIAGNOSE THE DRIVE
          00AD           GENF    F_SEARCHA                   ;SEARCH AHEAD OF SECTOR
          00AE
          00AE   ;
          00AE   ; OFFSET TABLE
          00AE   ;
          00AE
          00AE   OFFTAB:                                     ;
      00  00AE           .BYTE   0                           ;RETURN TO CENTERLINE
      01  00AF           .BYTE   ^X01                        ;+ OFFSET (BIT 0 = OFFSET FLAG)
      81  00B0           .BYTE   ^X81                        ;- OFFSET (BIT 0 = OFFSET FLAG)
      00  00B1           .BYTE   0                           ;RETURN TO CENTERLINE
00000004  00B2   OFFSIZ=.-OFFTAB                             ;SIZE OF OFFSET TABLE
```

```
00B2            .SBTTL  FUNCTION DECISION TABLE
00B2    ;+
00B2    ; RM03 FUNCTION DECISION TABLE
00B2    ;-
00B2
00B2    DR_FUNCTABLE:                                   ;FUNCTION DECISION TABLE
00B2            FUNCTAB ,-                              ;LEGAL FUNCTIONS
00B2                    <NOP,-                          ;NO OPERATION
00B2                    UNLOAD,-                        ;UNLOAD VOLUME
00B2                    SEEK,-                          ;SEEK CYLINDER
00B2                    RECAL,-                         ;RECALIBRATE
00B2                    DRVCLR,-                        ;DRIVE CLEAR
00B2                    RELEASE,-                       ;RELEASE PORT
00B2                    OFFSET,-                        ;OFFSET HEADS
00B2                    RETCENTER,-                     ;RETURN HEADS TO CENTERLINE
00B2                    PACKACK,-                       ;PACK ACKNOWLEDGE
00B2                    SEARCH,-                        ;SEARCH FOR SECTOR
00B2                    READPRESET,-                    ;READ IN PRESET
00B2                    SENSECHAR,-                     ;SENSE CHARACTERISTICS
00B2                    SETCHAR,-                       ;SET CHARACTERISTICS
00B2                    SENSEMODE,-                     ;SENSE MODE
00B2                    SETMODE,-                       ;SET MODE
00B2                    WRITECHECK,-                    ;WRITE CHECK
00B2                    WRITEHEAD,-                     ;WRITE HEADER AND DATA
00B2                    READHEAD,-                      ;READ HEADER AND DATA
00B2                    WRITETRACKD,-                   ;WRITE TRACK DESCRIPTOR
00B2                    READTRACKD,-                    ;READ TRACK DESCRIPTOR
00B2                    WRITECHECKH,-                   ;WRITE CHECK HEADER AND DATA
00B2                    DIAGNOSE,-                      ;DIAGNOSE THE DRIVE
00B2                    READLBLK,-                      ;READ LOGICAL BLOCK
00B2                    WRITELBLK,-                     ;WRITE LOGICAL BLOCK
00B2                    READPBLK,-                      ;READ PHYSICAL BLOCK
00B2                    WRITEPBLK,-                     ;WRITE PHYSICAL BLOCK
00B2                    READVBLK,-                      ;READ VIRTUAL BLOCK
00B2                    WRITEVBLK,-                     ;WRITE VIRTUAL BLOCK
00B2                    AVAILABLE,-                     ;AVAILABLE
00B2                    ACCESS,-                        ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00B2                    ACPCONTROL,-                    ;ACP CONTROL FUNCTION
00B2                    CREATE,-                        ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00B2                    DEACCESS,-                      ;DEACCESS FILE
00B2                    DELETE,-                        ;DELETE FILE AND/OR DIRECTORY ENTRY
00B2                    MODIFY,-                        ;MODIFY FILE ATTRIBUTES
00B2                    MOUNT>                          ;MOUNT VOLUME
00BA            FUNCTAB ,-                              ;BUFFERED I/O FUNCTIONS
00BA                    <NOP,-                          ;NO OPERATION
00BA                     UNLOAD,-                       ;UNLOAD VOLUME
00BA                     SEEK,-                         ;SEEK CYLINDER
00BA                     RECAL,-                        ;RECALIBRATE
00BA                     DRVCLR,-                       ;DRIVE CLEAR
00BA                     RELEASE,-                      ;RELEASE PORT
00BA                     OFFSET,-                       ;OFFSET HEADS
00BA                     RETCENTER,-                    ;RETURN HEADS TO CENTERLINE
00BA                     PACKACK,-                      ;PACK ACKNOWLEDGE
00BA                     SEARCH,-                       ;SEARCH FOR SECTOR
00BA                     READPRESET,-                   ;READ IN PRESET
00BA                     DIAGNOSE,-                     ;DIAGNOSE DRIVE
00BA                     SENSECHAR,-                    ;SENSE CHARACTERISTICS
```

```
00BA                     SETCHAR,-                       ;SET CHARACTERISTICS
00BA                     SENSEMODE,-                     ;SENSE MODE
00BA                     SETMODE,-                       ;SET MODE
00BA                     AVAILABLE,-                     ;AVAILABLE
00BA                     ACCESS,-                        ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00BA                     ACPCONTROL,-                    ;ACP CONTROL FUNCTION
00BA                     CREATE,-                        ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00BA                     DEACCESS,-                      ;DEACCESS FILE
00BA                     DELETE,-                        ;DELETE FILE AND/OR DIRECTORY ENTRY
00BA                     MODIFY,-                        ;MODIFY FILE ATTRIBUTES
00BA                     MOUNT>                          ;MOUNT VOLUME
00C2            FUNCTAB +ACP$READBLK,-                   ;READ FUNCTIONS
00C2                    <READTRACKD,-                    ;READ TRACK DESCRIPTOR
00C2                     READHEAD,-                      ;READ HEADER
00C2                     READLBLK,-                      ;READ LOGICAL BLOCK
00C2                     READPBLK,-                      ;READ PHYSICAL BLOCK
00C2                     READVBLK>                       ;READ VIRTUAL BLOCK
00CE            FUNCTAB +ACP$WRITEBLK,-                  ;WRITE FUNCTIONS
00CE                    <WRITETRACKD,-                   ;WRITE TRACK DESCRIPTOR
00CE                     WRITECHECK,-                    ;WRITE CHECK
00CE                     WRITECHECKH,-                   ;WRITE CHECK HEADER AND DATA
00CE                     WRITEHEAD,-                     ;WRITE HEADER
00CE                     WRITELBLK,-                     ;WRITE LOGICAL BLOCK
00CE                     WRITEPBLK,-                     ;WRITE PHYSICAL BLOCK
00CE                     WRITEVBLK>                      ;WRITE VIRTUAL BLOCK
00DA            FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00E6            FUNCTAB +ACP$DEACCESS,<DEACCESS> ;DEACCESS FILE
00F2            FUNCTAB +ACP$MODIFY,-                    ;
00F2                    <ACPCONTROL,-                    ;ACP CONTROL FUNCTION
00F2                     DELETE,-                        ;DELETE FILE OR DIRECTORY ENTRY
00F2                     MODIFY>                         ;MODIFY FILE ATTRIBUTES
00FE            FUNCTAB +ACP$MOUNT,<MOUNT>               ;MOUNT VOLUME
010A            FUNCTAB +EXE$LCLDSKVALID,-               ;LOCAL DISK VALID FUNCTIONS
010A                    <UNLOAD,-                        ;UNLOAD VOLUME
010A                     AVAILABLE,-                     ;UNIT AVAILABLE
010A                     PACKACK>                        ;PACK ACKNOWLEDGE
0116            FUNCTAB +EXE$ZEROPARM,-                  ;ZERO PARAMETER FUNCTIONS
0116                    <NOP,-                           ;NO OPERATION
0116                     UNLOAD,-                        ;UNLOAD VOLUME
0116                     RECAL,-                         ;RECALIBRATE
0116                     DRVCLR,-                        ;DRIVE CLEAR
0116                     RELEASE,-                       ;RELEASE PORT
0116                     RETCENTER,-                     ;RETURN HEADS TO CENTERLINE
0116                     READPRESET,-                    ;READ IN PRESET
0116                     PACKACK,-                       ;PACK ACKNOWLEDGE
0116                     AVAILABLE>                      ;AVAILABLE
0122            FUNCTAB +EXE$ONEPARM,-                   ;ONE PARAMETER FUNCTIONS
0122                    <SEEK,-                          ;SEEK CYLINDER
0122                     OFFSET,-                        ;OFFSET HEADS
0122                     SEARCH,-                        ;SEARCH FOR SECTOR
0122                     DIAGNOSE>                       ;DIAGNOSE THE DRIVE
012E            FUNCTAB +EXE$SENSEMODE,-                 ;
012E                    <SENSECHAR,-                     ;SENSE CHARACTERISTICS
012E                     SENSEMODE>                      ;SENSE MODE
013A            FUNCTAB +EXE$SETCHAR,-                   ;
013A                    <SETCHAR,-                       ;SET CHARACTERISTICS
013A                     SETMODE>                        ;SET MODE
```

```
                              0146          .SBTTL  START I/O OPERATION
                              0146  ;+
                              0146  ; DR_STARTIO - START I/O OPERATION ON DEVICE UNIT
                              0146  ;
                              0146  ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
                              0146  ;
                              0146  ; INPUTS:
                              0146  ;
                              0146  ;       R3 = ADDRESS OF I/O PACKET.
                              0146  ;       R5 = UCB ADDRESS OF DEVICE UNIT.
                              0146  ;
                              0146  ; OUTPUTS:
                              0146  ;
                              0146  ;       FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
                              0146  ;       RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
                              0146  ;       THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
                              0146  ;-
                              0146
                              0146  DR_STARTIO:                                     ;START I/O OPERATION
0080 C5    0081 C5     90     0146          MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
  009A C5    20 A3     B0     014D          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
           00D2 C5     B4     0153          CLRW    UCB$W_DR_MR(R5)         ;CLEAR THE MAINTENANCE VALUE
00D4 C5    FFF7 8F     AA     0157          BICW    #^C<DR_M_DUALPORT, -    ;Clear software status and error log
                              015E                  UCB$B_DR_SSTS(R5)       ;bytes, except for dualport bit.
      50     38 A3     D0     015E          MOVL    IRP$L_MEDIA(R3),R0      ;GET PARAMETER LONGWORD
                              0162
                              0162  ;
                              0162  ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                              0162  ;
                              0162
           06    00    EF     0162  10$:    EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODE
      51     20 A3            0165                  IRP$W_FUNC(R3),R1       ;
            51   02    91     0168          CMPB    #IO$_SEEK,R1            ;SEEK FUNCTION?
                 1E    13     016B          BEQL    20$                     ;IF EQL YES
            51   06    91     016D          CMPB    #IO$_OFFSET,R1          ;OFFSET FUNCTION?
                 20    13     0170          BEQL    30$                     ;IF EQL YES
            51   09    91     0172          CMPB    #IO$_SEARCH,R1          ;SEARCH FUNCTION?
                 22    13     0175          BEQL    40$                     ;IF EQL YES
            51   1D    91     0177          CMPB    #IO$_DIAGNOSE,R1        ;DIAGNOSE FUNCTION?
                 24    13     017A          BEQL    45$                     ;IF EQL YES
       UOBC C5   50    D0     017C          MOVL    R0,UCB$W_DA(R5)         ;STORE PARAMETER LONGWORD
            51   18    91     0181          CMPB    #IO$_WRITECHECKH,R1     ;DISJOINT FUNCTION CODE?
                 22    1A     0184          BGTRU   50$                     ;IF GTRU NO
            51   06    A2     0186          SUBW    #IO$_WRITECHECKH-IO$_AVAILABLE-1,R1 ;MAKE FUNCTION TABLE INDEX
                 1D    11     0189          BRB     50$                     ;
                              018B
                              018B  ;
                              018B  ; SEEK FUNCTION - SET CYLINDER ADDRESS
                              018B  ;
                              018B
       00BE C5   50    B0     018B  20$:    MOVW    R0,UCB$W_DC(R5)         ;SET CYLINDER ADDRESS
                 16    11     0190          BRB     50$                     ;
                              0192
                              0192  ;
                              0192  ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
                              0192  ;
                              0192
```

```
   00C8 C5   50   90  0192   30$:    MOVB    R0,UCB$W_OFFSET(R5)           ;SET OFFSET VALUE
             0F   11  0197           BRB     50$                           ;
                      0199
                      0199   ;
                      0199   ; SEARCH FUNCTION - SET SECTOR ADDRESS
                      0199   ;
                      0199
   00BC C5   50   90  0199   40$:    MOVB    R0,UCB$W_DA(R5)               ;SET SECTOR ADDRESS
             08   11  019E           BRB     50$                           ;
                      01A0
                      01A0   ;
                      01A0   ; DIAGNOSE FUNCTION - SET MAINTENANCE VALUE
                      01A0   ;
                      01A0
   00D2 C5   50   B0  01A0   45$:    MOVW    R0,UCB$W_DR_MR(R5)            ;SET MAINTENANCE VALUE
        51   03   A2  01A5           SUBW    #IO$_DIAGNOSE-IO$_READPRESET-1,R1 ;MAKE A FUNCTION TABLE INDEX
                      01A8
                      01A8   ;
                      01A8   ; FINISH PREPROCESSING
                      01A8   ;
                      01A8
   0092 C5   51   90  01A8   50$:    MOVB    R1,UCB$B_FEX(R5)              ;SAVE FUNCTION DISPATCH INDEX
     54   24 A5   D0  01AD           MOVL    UCB$L_CRB(R5),R4              ;GET ADDRESS OF CRB
     54   2C B4   D0  01B1           MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4 ;GET FIRST CONTROLLER CSR ADDRESS
  00 68 A5   00   E4  01B5           BBSC    #UCB$V_ECC,UCB$W_DEVSTS(R5),FDISPATCH ;CLEAR ECC CORRECTION MADE
                      01BA
                      01BA   ;
                      01BA   ; CENTRAL FUNCTION DISPATCH
                      01BA   ;
                      01BA
                      01BA   FDISPATCH:                                    ;FUNCTION DISPATCH
     53   58 A5   D0  01BA           MOVL    UCB$L_IRP(R5),R3              ;RETRIEVE ADDRESS OF I/O PACKET
  0D 2A A3   08   E0  01BE           BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
  08 64 A5   0B   E0  01C3           BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
   50   0254 8F   3C  01C8           MOVZWL  #SS$_VOLINV,R0                ;SET VOLUME INVALID STATUS
            06D1  31  01CD           BRW     RESETXFR                      ;
                      01D0
                      01D0   ;
                      01D0   ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
                      01D0   ;
                      01D0
   50   0092 C5   9A  01D0   10$:    MOVZBL  UCB$B_FEX(R5),R0              ;GET DISPATCH FUNCTION CODE
   00C9 C5   10   90  01D5           MOVB    #RM_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
   00CB C5   01   90  01DA           MOVB    #1,UCB$B_OFFRTC(R5)           ;SET INITIAL OFFSET RETRY COUNT
        00CA C5   94  01DF           CLRB    UCB$B_OFFNDX(R5)              ;CLEAR INITIAL OFFSET TABLE INDEX
                      01E3
                      01E3   ;
                      01E3   ; CHECK FOR DIAGNOSTIC MODIFIERS
                      01E3   ;
  2F 2A A3   08   E1  01E3           BBC     #IRP$V_PHYSIO,IRP$W_STS(R3),40$  ;IF CLEAR, NOT PHYSICAL I/O
06 009A C5   06   E1  01E8           BBC     #IO$V_COMMOD,UCB$W_FUNC(R5),15$  ;IF CLEAR, NO COMMAND MODIFIER
00C9 C5   80 8F   88  01EE           BISB    #RM_OF_M_CMO/256,UCB$W_OFFSET+1(R5)  ;SET COMMAND MODIFIER
                      01F4
06 009A C5   07   E1  01F4   15$:    BBC     #IO$V_MOVETRACKD,UCB$W_FUNC(R5),20$  ;IF CLR, NO MOVE TRACK DESC
00C9 C5   40 8F   88  01FA           BISB    #RM_OF_M_MTD/256,UCB$W_OFFSET+1(R5)  ;SET MOVE TRACK DESCRIPTOR
                      0200
06 009A C5   08   E1  0200   20$:    BBC     #IO$V_DIAGNOSTIC,UCB$W_FUNC(R5),30$  ;IF CLEAR, NOT DIAG MODE
```

```
   00D3 C5    80 8F    88 0206              BISB    #RM_MR_M_DM/256,UCB$W_DR_MR+1(R5)  ;SET DIAGNOSTIC MODE
                          020C
05 009A C5    09       E1 020C      30$:    BBC     #IO$V_SKPSECINH,UCB$W_FUNC(R5),40$  ;IF CLEAR, NO SSEI MODIFIER
   00C9 C5    02       88 0212              BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)  ;SET SKIP SECTOR ERR INH
                          0217
                          0217      ;
                          0217      ; DISPATCH TO FUNCTION HANDLING ROUTINE
                          0217      ;
                          0217      40$:
                          0217              CASE    R0,<-                        ;DISPATCH TO FUNCTION HANDLING ROUTINE
                          0217                      NOP,-                        ;NO OPERATION
                          0217                      UNLOAD,-                     ;UNLOAD VOLUME
                          0217                      SEEK,-                       ;SEEK CYLINDER
                          0217                      RECAL,-                      ;RECALIBRATE
                          0217                      DRVCLR,-                     ;DRIVE CLEAR
                          0217                      RELEASE,-                    ;RELEASE PORT
                          0217                      OFFSET,-                     ;OFFSET HEADS
                          0217                      RETCENTER,-                  ;RETURN HEADS TO CENTER
                          0217                      PACKACK,-                    ;PACK ACKNOWLEDGE
                          0217                      SEARCH,-                     ;SEARCH FOR SECTOR
                          0217                      WRITECHECK,-                 ;WRITE CHECK DATA
                          0217                      WRITEDATA,-                  ;WRITE DATA
                          0217                      READDATA,-                   ;READ DATA
                          0217                      WRITEHEAD,-                  ;WRITE HEADER AND DATA
                          0217                      READHEAD,-                   ;READ HEADER AND DATA
                          0217                      WRITETRACKD,-                ;WRITE TRACK DESCRIPTOR
                          0217                      READTRACKD,-                 ;READ TRACK DESCRIPTOR
                          0217                      AVAILABLE,-                  ;AVAILABLE
                          0217                      WRITECHECKH,-                ;WRITE CHECK HEADER AND DATA
                          0217                      READPRESET,-                 ;READIN PRESET
                          0217                      DIAGNOSE>                    ;DIAGNOSE DRIVE
                          0245
                          0245      ; UNLOAD or AVAILABLE - Clear UCB$V_VALID
                          0245      ;   This is the only operation which these functions need to perform.  All
                          0245      ;   devices supported by this driver do not have an unload function, and the
                          0245      ;   available function code should only clear the UCB$V_VALID bit.
                          0245
                          0245      UNLOAD:
                          0245      AVAILABLE:
   64 A5    0800 8F    AA 0245              BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear the software volume valid
                 00AF  31 024B              BRW     NORMAL                       ;bit and complete function.
                          024E
                          024E      ;
                          024E      ; PACKACK - Set UCB$V_VALID and proceed with hardware pack acknowledge
                          024E      ;                 function
                          024E      ;
                          024E      PACKACK:
   64 A5    0800 8F    A8 024E              BISW    #UCB$M_VALID, UCB$W_STS(R5) ;Set the software volume valid
                          0254      ;       BRB     NOP                          ;bit and complete function.
                          0254
                          0254      ;
                          0254      ; NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
                          0254      ; RETURN TO CENTER LINE, SEARCH, AND READ IN PRESET
                          0254      ;
                          0254
                          0254      NOP:                                         ;NO OPERATION
                          0254      SEEK:                                        ;SEEK CYLINDER
```

```
                     0254  RECAL:                                    ;RECALIBRATE
                     0254  DRVCLR:                                   ;DRIVE CLEAR
                     0254  RELEASE:                                  ;RELEASE PORT
                     0254  OFFSET:                                   ;OFFSET READ HEADS
                     0254  RETCENTER:                                ;RETURN TO CENTERLINE
                     0254  SEARCH:                                   ;SEARCH FOR SECTOR
                     0254  READPRESET:                               ;READIN PRESET
                     0254          EXFUNC  RETRY                     ;EXECUTE HOUSEKEEPING FUNCTION
        00A1   31    0259          BRW     NORMAL                    ;
                     025C
                     025C  ;
                     025C  ; WRITE TRACK DESCRIPTOR and READ TRACK DESCRIPTOR
                     025C  ;       Both want to SEEK rather than to SEARCH to arrive on cylinder.
                     025C  ;
                     025C
                     025C  WRITETRACKD:                              ;WRITE TRACK DESCRIPTOR
   00D4 C5   04   88 025C          BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
                     0261
                     0261  READTRACKD:                               ;READ TRACK DESCRIPTOR
             0C   E0 0261          BBS     #IO$V_INHSEEK,-
     23 009A C5      0263                  UCB$W_FUNC(R5),TRANRQCH ; If set, NO explicit SEEK
                     0267          EXFUNC  RETRY,F_SEEK              ; Seek to cylinder
             19   11 026F          BRB     TRANRQCH                  ;  and branch around to common code.
                     0271
                     0271  ;
                     0271  ; WRITE CHECK DATA AND WRITE CHECK HEADER AND DATA
                     0271  ;
                     0271
                     0271  WRITECHECK:                               ;WRITE CHECK DATA
                     0271  WRITECHECKH:                              ;WRITE CHECK HEADER AND DATA
00 009A C5   0E   E4 0271          BBSC    #IO$V_DATACHECK,UCB$W_FUNC(R5),WRITEDATA ;CLEAR DATA CHECK REQUEST
                     0277
                     0277  ;
                     0277  ; WRITE DATA, WRITE HEADER AND DATA,
                     0277  ; WRITE CHECK DATA, AND WRITE CHECK HEADER AND DATA
                     0277  ;
                     0277
                     0277  WRITEDATA:                                ;WRITE DATA
                     0277  WRITEHEAD:                                ;WRITE HEADER AND DATA
   00D4 C5   04   88 0277          BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
                     027C
                     027C  ;
                     027C  ; READ DATA, READ HEADER AND DATA,
                     027C  ; WRITE DATA, WRITE HEADER AND DATA,
                     027C  ; WRITE CHECK DATA, AND WRITE CHECK HEADER AND DATA
                     027C  ;
                     027C
                     027C  READDATA:                                 ;READ DATA
                     027C  READHEAD:                                 ;READ HEADER AND DATA
08 009A C5   0C   E0 027C          BBS     #IO$V_INHSEEK,UCB$W_FUNC(R5),TRANRQCH ;IF SET, NO EXPLICIT SEEK
                     0282          EXFUNC  RETRY,F_SEARCHA           ;SEARCH AHEAD OF STARTING SECTOR
                     028A
                     028A  ;
                     028A  ; DATA TRANSFER OR DIAGNOSE - REQUEST CHANNEL
                     028A  ;
                     028A
                     028A  DIAGNOSE:                                 ;DIAGNOSE
```

```
                           028A  TRANRQCH:                                      ;DATA TRANSFER
                           028A          REQPCHAN LOW                           ;REQUEST PRIMARY CHANNEL
                           0290
                           0290  ;
                           0290  ; DATA TRANSFER - CHANNEL ALREADY OWNED
                           0290  ;
                           0290
                           0290  TRANNOCH:                                      ;DATA TRANSFER CHANNEL OWNED
     50   0092 C5     9A   0290          MOVZBL  UCB$B_FEX(R5),R0               ;GET FUNCTION DISPATCH INDEX
                           0295          EXFUNC  TRANXT                         ;EXECUTE TRANSFER FUNCTION
                           029A
                           029A  ;
                           029A  ; DATA CHECK
                           029A  ;
                           029A
                           029A  DATACHECK:                                     ;DATA CHECK
 5D 009A C5     0E    E1   029A          BBC     #IO$V_DATACHECK,UCB$W_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
    50   0639 8F      3C   02A0          MOVZWL  #SS$_WASECC,R0                 ;ASSUME ECC CORRECTION WAS MADE
   56 68 A5     00    E0   02A5          BBS     #UCB$V_ECC,UCB$W_DEVSTS(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
                           02AA          RELCHAN                                ;RELEASE CHANNEL
   00D4 C5      01    88   02B0          BISB    #DR_M_DCK,UCB$B_DR_SSTS(R5)   ;SET DATA CHECK IN PROGRESS
   00C9 C5      10    90   02B5          MOVB    #RM_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
   00D4 C5      04    88   02BA          BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
   00CB C5      01    90   02BF          MOVB    #1,UCB$B_OFFRTC(R5)            ;SET INITIAL OFFSET RETRY COUNT
        00CA    C5    94   02C4          CLRB    UCB$B_OFFNDX(R5)               ;CLEAR INITIAL OFFSET TABLE INDEX
     52     58  A5    D0   02C8          MOVL    UCB$L_IRP(R5),R2               ;GET ADDRESS OF IRP
  78 A5     2C  A2    7D   02CC          MOVQ    IRP$L_SVAPTE(R2),UCB$L_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
00BC C5     38  A2    D0   02D1          MOVL    IRP$L_MEDIA(R2),UCB$W_DA(R5)  ;
  0B 2A A2      08    E1   02D7          BBC     #IRP$V_PHYSIO,IRP$W_STS(R2),CHECKRETRY  ;IF CLEAR NOT PHYS I/O
05 009A C5      09    E1   02DC          BBC     #IO$V_SKPSECINH,UCB$W_FUNC(R5),CHECKRETRY  ;IF CLEAR NO SSEI MOD
   00C9 C5      02    88   02E2          BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)  ;SET SKIP SECTOR ERR INH
                           02E7
                           02E7  ;
                           02E7  ; DATA CHECK RETRY
                           02E7  ;
                           02E7
                           02E7  CHECKRETRY:                                    ;DATA CHECK RETRY
                           02E7          REQPCHAN LOW                           ;REQUEST PRIMARY CHANNEL FOR DATA CHECK
     50   0092 C5     9A   02ED          MOVZBL  UCB$B_FEX(R5),R0               ;GET FUNCTION DISPATCH INDEX
  50    FD36 CF40     9A   02F2          MOVZBL  CHECKTAB-CDF_WRITEDATA[R0],R0 ;GET CASE TABLE INDEX
                           02F8          EXFUNC  TRANXT                         ;EXECUTE DATA CHECK FUNCTION
                           02FD
                           02FD  ;
                           02FD  ; SUCCESSFUL OPERATION COMPLETION
                           02FD  ;
                           02FD
                           02FD  NORMAL:                                        ;
            50  01    3C   02FD          MOVZWL  #SS$_NORMAL,R0                 ;SET NORMAL COMPLETION STATUS
                           0300  CHECKXT:                                       ;
              0208    31   0300          BRW     FUNCXT                         ;
                           0303
                           0303  ;
                           0303  ; TRANSFER ENDED WITH A RETRIABLE ERROR
                           0303  ;
                           0303
                           0303  TRANXT:                                        ;TRANSFER EXIT
   0093 C5      0B    91   0303          CMPB    #CDF_WRITEDATA,UCB$B_CEX(R5) ;WRITE DATA FUNCTION?
```

```
                       24    13  0308    847              BEQL     RETRY                          ;IF EQL YES
           0093 C5     0D    91  030A    848              CMPB     #CDF_WRITEHEAD,UCB$B_CEX(R5)  ;WRITE HEADER FUNCTION?
                       1D    13  030F    849              BEQL     RETRY                          ;IF EQL YES
     51    00064F74 8F       D3  0311    850              BITL     #MBA$M_SR_DLT!-                ;DATA LATE OR,
                                 0318    851                       MBA$M_SR_INVMAP!-              ;INVALID MAP REGISTER OR,
                                 0318    852                       MBA$M_SR_MAPPE!-               ;MAP REGISTER PARITY ERROR OR,
                                 0318    853                       MBA$M_SR_MCPE!-                ;MASSBUS CONTROL PARITY ERROR OR,
                                 0318    854                       MBA$M_SR_SPE!-                 ;MBA SILO PARITY ERROR OR,
                                 0318    855                       MBA$M_SR_MDPE!-                ;MASSBUS DATA PARITY ERROR OR,
                                 0318    856                       MBA$M_SR_MXF!-                 ;MISSED TRANSFER OR,
                                 0318    857                       MBA$M_SR_NED!-                 ;NONEXISTENT DISK OR,
                                 0318    858                       MBA$M_SR_RDS!-                 ;READ DATA SUBSTITUTE OR,
                                 0318    859                       MBA$M_SR_WCKLWR!-              ;WRITE CHECK LOWER BYTE OR,
                                 0318    860                       MBA$M_SR_WCKUPR,R1             ;WRITE CHECK UPPER BYTE?
                       14    12  0318    861              BNEQ     RETRY                          ;IF NEQ YES - RETRY FUNCTION
     00D0 C5      1C88 8F    B3  031A    862              BITW     #RM_ER2_M_DPE!-                ;DATA PARITY ERROR OR,
                                 0321    863                       RM_ER2_M_DVC!-                 ;DEVICE CHECK OR,
                                 0321    864                       RM_ER2_M_LBC!-                 ;LOSS OF BIT CLOCK OR,
                                 0321    865                       RM_ER2_M_LSC!-                 ;LOSS OF SYSTEM CLOCK OR,
                                 0321    866                       RM_ER2_M_IVC,UCB$W_DR_ER2(R5)  ;INVALID COMMAND?
                       0B    12  0321    867              BNEQ     RETRY                          ;IF NEQ YES - RETRY FUNCTION
              0A 52    08    E0  0323    868              BBS      #RM_ER1_V_HCRC,R2,ECC          ;Test HCRC before HCE.
              52  20A8 8F    B3  0327    869              BITW     #RM_ER1_M_OPI!-                ;OPERATION INCOMPLETE OR,
                                 032C    870                       RM_ER1_M_PAR!-                 ;PARITY ERROR OR,
                                 032C    871                       RM_ER1_M_HCE!-                 ;HEADER COMPARE ERROR OR,
                                 032C    872                       RM_ER1_M_WCF,R2                ;WRITE CLOCK FAIL?
                       03    13  032C    873              BEQL     ECC                            ;IF EQL NO
                                 032E    874 RETRY:                                               ;
                     0110    31  032E    875              BRW      RETRYERR                       ;RETRIABLE ERROR
                                 0331    876
                                 0331    877 ;
                                 0331    878 ; ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
                                 0331    879 ;
                                 0331    880
                                 0331    881 ECC:                                                 ;ECC CORRECTION
 51   7E A5   00D8 C5        A1  0331    882              ADDW3    UCB$L_DR_BCR(R5), -            ; Compute bytes transfered then
                                 0338    883                       UCB$W_BCNT(R5), R1             ; clear byte offset bits and
 50   51   FFFF01FF 8F       CB  0338    884              BICL3    #^XFFFF01FF, R1, R0            ; convert result to a longword.
                       77    13  0340    885              BEQL     OFF                            ; Branch if whole blocks xfered is zero.
          51      01FF 8F    B3  0342    886              BITW     #^X1FF, R1                     ; Was a partial block transfered?
                       70    12  0347    887              BNEQ     OFF                            ; Branch if partial block transfered.
              10 52    08    E1  0349    888              BBC      #RM_ER1_V_HCRC, R2, 10$        ; Branch if error was not HCRC.
                       07    91  034D    889              CMPB     #DT$_RP07,-                    ; Is this drive an RP07?
                    41 A5        034F    890                       UCB$B_DEVTYPE(R5)              ;
                       11    12  0351    891              BNEQ     20$                            ; Branch if not.
              00000400 8F    E1  0353    892              BBC      #RM_OF_M_HCI,-                 ; Branch if header compare inhibit
                 07 00C8 C5      0359    893                       UCB$W_OFFSET(R5),20$           ;   isn't set.
      50      00000200 8F    C2  035D    894 10$:         SUBL2    #512, R0                       ; Else, reduce bytes xfered by a block.
              52  1140 8F    B3  0364    895 20$:         BITW     #RM_ER1_M_DTE!-                ; For:   DRIVE TIMING ERROR
                                 0369    896                       RM_ER1_M_ECH!-                 ;        ECC HARD ERROR
                                 0369    897                       RM_ER1_M_HCRC,R2               ;        HEADER CRC ERROR
                       4E    12  0369    898              BNEQ     OFF                            ;               perform offset recovery.
         48 00D4 C5    02    E0  036B    899              BBS      #DR_V_NOECC, -                 ;If it won't help, skip ECC correction.
                                 0371    900                       UCB$B_DR_SSTS(R5), OFF         ;
                    7E    52 7D  0371    901              MOVQ     R2,-(SP)                       ; Save work registers.
 52    00C6 C5     0B     00 EA  0374    902              FFS      #0,#11,UCB$W_EC2(R5),R2        ; Find the first error bit in the ECC
                                 037B    903                                                      ;  pattern.
```

```
        53   0A   52   C3  037B           SUBL3   R2,#10,R3                 ; Get the number of error bits
                           037F                                             ;  remaining in the pattern.
                  09   15  037F           BLEQ    30$                       ; Branch if no other bits in pattern.
                  52   D6  0381           INCL    R2                        ; Point to next bit in pattern.
   52   00C6 C5   53   52  EF  0383       EXTZV   R2,R3,UCB$W_EC2(R5),R2    ; Is there more than one error bit set?
                  0C   BA  038A   30$:    POPR    #^M<R3,R2>                ; Restore work registers without
                           038C                                             ;  affecting flags.
                  26   1A  038C           BGTRU   DEFER_ECC                 ; If more than one error bit set, don't
                           038E                                             ;  apply ECC correction.
                           038E   ;
                           038E   ; APPLY_ECC -
                           038E   ;
                           038E   ; Apply ECC correction to correct a single bit error.
                           038E   ;
                           038E
                           038E   APPLY_ECC:
             7E   51   3C  038E           MOVZWL  R1, -(SP)                 ; Save total bytes transfered, inc. ECC.
        00000000'GF    16  0391           JSB     G^IOC$APPLYECC            ; Apply ECC correction.
                  50 8ED0  0397           POPL    R0                        ; Retrieve transfered byte count.
        00000000'GF    16  039A           JSB     G^IOC$UPDATRANSP          ; Update transfer parameters.
             00CA C5   94  03A0           CLRB    UCB$B_OFFNDX(R5)          ; Reset offset table index.
                  02   8A  03A4           BICB    #DR_M_OM,-                ; Clear offset mode.
             00D4 C5       03A6                   UCB$B_DR_SSTS(R5)         ;
             7E A5     B5  03A9           TSTW    UCB$W_BCNT(R5)            ; Any more to transfer?
                  03   13  03AC           BEQL    20$                       ; If EQL no.
                FEDF   31  03AE           BRW     TRANNOCH                  ; Transfer next segment.
                FEE6   31  03B1   20$:    BRW     DATACHECK                 ; Check for write check.
                           03B4
                           03B4   ;
                           03B4   ; DEFER_ECC -
                           03B4   ;
                           03B4   ; Don't apply ECC correction for multiple bit errors unless the error cannot
                           03B4   ; be recovered with offset retries.
                           03B4   ;
                           03B4
                           03B4   DEFER_ECC:
                  10   88  03B4           BISB    #DR_M_ECC_DEFER,-         ; Set flag to indicate that ECC
             00D4 C5       03B6                   UCB$B_DR_SSTS(R5)         ;  can be used if offset recovery fails.
                           03B9
                           03B9   ;
                           03B9   ; OFF - OFFSET RECOVERY
                           03B9   ;
                           03B9   ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER CRC, OR ECC
                           03B9   ; HARD ERROR IS DETECTED ON A READ FUNCTION.
                           03B9   ;
                           03B9
                           03B9   OFF:                                      ;OFFSET RECOVERY
                  50   D5  03B9           TSTL    R0                        ;ANY GOOD DATA TRANSFERED?
                  33   13  03BB           BEQL    30$                       ;IF EQL NO
                           03BD
                           03BD   ;
                           03BD   ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
                           03BD   ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
                           03BD   ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
                           03BD   ; POINT OF ERROR.
                           03BD   ;
                           03BD
        00000000'GF   16  03BD            JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
              7E A5   B5  03C3            TSTW    UCB$W_BCNT(R5)          ; Any more data to transfer?
                 03   12  03C6            BNEQ    5$                      ; Branch if so.
               FECF   31  03C8            BRW     DATACHECK               ; Otherwise, go check for write check.
            00CA C5   94  03CB  5$:       CLRB    UCB$B_OFFNDX(R5)        ;RESET OFFSET TABLE INDEX
        00CB C5  10   90  03CF  10$:      MOVB    #16,UCB$B_OFFRTC(R5)    ;SET OFFSET RETRY COUNT
        00CA C5  04   91  03D4            CMPB    #OFFSIZ,UCB$B_OFFNDX(R5) ;ALL OFFSETS TRIED?
                 08   12  03D9            BNEQ    20$                     ; Branch if not.
                 04   E4  03DB            BBSC    #DR_V_ECC_DEFER,-       ; Correct the error with ECC if we can.
            00D4 C5       03DD                    UCB$B_DR_SSTS(R5),-     ;
                 AD       03E0                    APPLY_ECC               ;
                 53   11  03E1            BRB     90$                     ; Otherwise, fatal error.
                          03E3  20$:      RELCHAN                         ;RELEASE CHANNEL
        00D4 C5  02   8A  03E9            BICB    #DR_M_OM,UCB$B_DR_SSTS(R5)  ;CLEAR OFFSET MODE
                 35   11  03EE            BRB     60$                     ;
                          03F0
                          03F0  ;
                          03F0  ; NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
                          03F0  ;
                          03F0
       52   9040 8F   B3  03F0  30$:      BITW    #RM_ER1_M_DCK!-         ;DATA CHECK OR,
                          03F5                    RM_ER1_M_DTE!-          ;DRIVE TIMING OR,
                          03F5                    RM_ER1_M_ECH,R2         ;ECC HARD ERROR?
                 05   12  03F5            BNEQ    40$                     ;IF NEQ YES
        00C9 C5  04   88  03F7            BISB    #RM_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;SET HEADER COMPARE INHIBIT
            00CB C5   97  03FC  40$:      DECB    UCB$B_OFFRTC(R5)        ;CHANGE CURRENT OFFSET?
                 28   12  0400            BNEQ    70$                     ;IF NEQ NO
            00CA C5   96  0402            INCB    UCB$B_OFFNDX(R5)        ;UPDATE OFFSET TABLE INDEX
       50   00CA C5   9A  0406            MOVZBL  UCB$B_OFFNDX(R5),R0     ;GET NEXT OFFSET TABLE INDEX
00C8 C5   FC9D CF40   90  040B            MOVB    OFFTAB-1[R0],UCB$W_OFFSET(R5) ;GET NEXT OFFSET VALUE
                 BA   13  0413            BEQL    10$                     ;IF EQL RETURN TO CENTERLINE
        00CB C5  02   90  0415            MOVB    #2,UCB$B_OFFRTC(R5)     ;SET OFFSET RETRY COUNT
                          041A            RELCHAN                         ;RELEASE CHANNEL
        00D4 C5  02   88  0420            BISB    #DR_M_OM,UCB$B_DR_SSTS(R5)  ;SET OFFSET MODE
        00C9 C5  04   8A  0425  60$:      BICB    #RM_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;CLEAR HEADER COMPARE INHIBIT
     03 00D4 C5  00   E0  042A  70$:      BBS     #DR_V_DCK,UCB$B_DR_SSTS(R5),80$ ;IF SET, DATA CHECK FUNCTION
               FE57   31  0430            BRW     TRANRQCH                ;TRY FUNCTION AGAIN
               FEB1   31  0433  80$:      BRW     CHECKRETRY              ;TRY DATA CHECK AGAIN
                          0436
         50   04 A3   D0  0436  90$:      MOVL    RM_DS(R3),R0            ;GET DRIVE STATUS
       51   00CC C5   D0  043A            MOVL    UCB$L_DR_SR(R5),R1      ;GET MBA STATUS
                 32   11  043F            BRB     FATALERR                ;
                          0441
                          0441  ;
                          0441  ; RETRIABLE ERROR
                          0441  ;
                          0441
                          0441  RETRYERR:                                 ;RETRIABLE ERROR
                 07   BB  0441            PUSHR   #^M<R0,R1,R2>           ; Save error status registers.
                          0443            RELCHAN                         ; Release channel before possible RECAL
                 07   BA  0449            POPR    #^M<R0,R1,R2>           ; Restore error status registers.
     04 00D0 C5  0E   E0  044B            BBS     #RM_ER2_V_SKI,UCB$W_DR_ER2(R5),10$ ;IF SET, SEEK INCOMPLETE
          0D 52  07   E1  0451            BBC     #RM_ER1_V_HCE,R2,20$    ;IF CLR, HEADER COMPARED
                          0455  10$:      EXFUNC  FATALERR,F_RECAL        ;RECALIBRATE HEADS
       52   2000 8F   3C  045D            MOVZWL  #RM_ER1_M_DPI,R2        ;SET AN ERROR FOR CALLER TO SEE
            0080 C5   97  0462  20$:      DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
                 0B   13  0466            BEQL    FATALERR                ;IF EQL NO
```

```
                           0468          EXFUNC  FATALERR,F_DRVCLR        ; Issue drive clear before retrying.
           FD47      31    0470          BRW     FDISPATCH                ;
                           0473
                           0473  ;
                           0473  ; FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
                           0473  ; INHIBITED, OR FINAL OFFSET TRIED
                           0473  ;
                           0473
                           0473  FATALERR:                                ;FATAL ERROR - SET STATUS
      08 50    0C    E0    0473          BBS     #RM_DS_V_MOL,R0,10$      ; Branch if not offline.
   50   01A4 8F      3C    0477          MOVZWL  #SS$_MEDOFL,R0           ; Otherwise, set medium offline status
           008C      31    047C          BRW     FUNCXT                   ;   and branch to common completion exit.
      08 50    06    E0    047F  10$:    BBS     #RM_DS_V_VV,R0,20$       ; Branch if not volume invalid.
   50   0254 8F      3C    0483          MOVZWL  #SS$_VOLINV,R0           ; Otherwise, set volume invalid status.
           0080      31    0488          BRW     FUNCXT                   ;   and branch to common completion exit.
      08 52    0E    E1    048B  20$:    BBC     #RM_ER1_V_UNS,R2,30$     ; Branch if not drive unsafe.
   50   023C 8F      3C    048F          MOVZWL  #SS$_UNSAFE,R0           ; Otherwise, set drive unsafe status.
           0074      31    0494          BRW     FUNCXT                   ;   and branch to common completion exit.
   50   02D4 8F      3C    0497  30$:    MOVZWL  #SS$_OPINCOMPL,R0        ;SET OPERATION INCOMPLETE STATUS
      6B 52    0D    E0    049C          BBS     #RM_ER1_V_OPI,R2,FUNCXT  ;IF SET, OPERATION INCOMPLETE
   50   025C 8F      3C    04A0          MOVZWL  #SS$_WRITLCK,R0          ;SET WRITE LOCK ERROR STATUS
      62 52    0B    E0    04A5          BBS     #RM_ER1_V_WLE,R2,FUNCXT  ;IF SET, WRITE LOCK ERROR
   50   0134 8F      3C    04A9          MOVZWL  #SS$_IVADDR,R0           ;SET INVALID DISK ADDRESS STATUS
   52   0600 8F      B3    04AE          BITW    #RM_ER1_M_AOE!-          ;DISK ADDRESS OVERFLOW OR,
                           04B3                  RM_ER1_M_IAE,R2          ;INVALID DISK ADDRESS ERROR?
             56      12    04B3          BNEQ    FUNCXT                   ;IF NEQ YES
   50   008C 8F      3C    04B5          MOVZWL  #SS$_DRVERR,R0           ;SET DRIVE ERROR STATUS
   52   1027 8F      B3    04BA          BITW    #RM_ER1_M_DTE!-          ;DRIVE TIMING ERROR OR,
                           04BF                  RM_ER1_M_ILF!-           ;ILLEGAL FUNCTION OR,
                           04BF                  RM_ER1_M_ILR!-           ;ILLEGAL REGISTER OR,
                           04BF                  RM_ER1_M_RMR!-           ;REGISTER MODIFY REFUSE OR,
                           04BF                  RM_ER1_M_WCF,R2          ;WRITE CLOCK FAIL ERROR?
             4A      12    04BF          BNEQ    FUNCXT                   ;IF NEQ YES
   50   01F4 8F      3C    04C1          MOVZWL  #SS$_PARITY,R0           ; Set parity error status.
   52   8140 8F      B3    04C6          BITW    #RM_ER1_M_DCK!-          ; Data check error or,
                           04CB                  RM_ER1_M_ECH!-           ;   ECC hard error or,
                           04CB                  RM_ER1_M_HCRC,R2         ;   header CRC error?
             3E      12    04CB          BNEQ    FUNCXT                   ; Branch if so.
 38 00D0 C5    0F    E0    04CD          BBS     #RM_ER2_V_BSE,UCB$W_DR_ER2(R5),FUNCXT ;IF SET, BAD SECTOR ERROR
   50   0054 8F      3C    04D3          MOVZWL  #SS$_CTRLERR,R0          ; Set fatal controller error status.
   52   0088 8F      B3    04D8          BITW    #RM_ER1_M_HCE!-          ; Header compare error or,
                           04DD                  RM_ER1_M_PAR,R2          ;   parity error?
             2C      12    04DD          BNEQ    FUNCXT                   ; Branch if so.
51    00024064 8F    D3    04DF          BITL    #MBA$M_SR_MAPPE!-        ;MAP PARITY ERROR OR,
                           04E6                  MBA$M_SR_MCPE!-          ;MASSBUS CONTROL PARITY ERROR OR,
                           04E6                  MBA$M_SR_SPE!-           ;MBA SILO PARITY ERROR OR,
                           04E6                  MBA$M_SR_MDPE!-          ;MASSBUS DATA PARITY ERROR OR,
                           04E6                  MBA$M_SR_RDS,R1          ;READ DATA SUBSTITUTE?
             23      12    04E6          BNEQ    FUNCXT                   ;IF NEQ YES
   50   00BC 8F      3C    04E8          MOVZWL  #SS$_FORMAT,R0           ;SET FORMAT ERROR STATUS
      1A 52    04    E0    04ED          BBS     #RM_ER1_V_FER,R2,FUNCXT  ;IF SET, FORMAT ERROR
   50   005C 8F      3C    04F1          MOVZWL  #SS$_DATACHECK,R0        ;SET DATA CHECK ERROR STATUS
   51   0600 8F      B3    04F6          BITW    #MBA$M_SR_WCKLWR!-       ;WRITE CHECK ERROR LOWER BYTE OR,
                           04FB                  MBA$M_SR_WCKUPR,R1       ;WRITE CHECK ERROR UPPER BYTE?
             0E      12    04FB          BNEQ    FUNCXT                   ;IF NEQ YES
   50   01C4 8F      3C    04FD          MOVZWL  #SS$_NONEXDRV,R0         ;SET NONEXISTENT DRIVE STATUS
      05 51    12    E0    0502          BBS     #MBA$V_SR_NED,R1,FUNCXT  ;IF SET, NONEXISTENT DRIVE
```

```
     50    0054 8F    3C  0506           MOVZWL  #SS$_CTRLERR,R0         ;SET CONTROLLER ERROR STATUS
                          050B
                          050B     ;
                          050B     ; FUNCTION COMPLETION COMMON EXIT
                          050B     ;
                          050B
                          050B     FUNCXT:                                 ;FUNCTION EXIT
                  50  DD  050B           PUSHL   R0                      ;SAVE FINAL REQUEST STATUS
          00000000'GF  16  050D           JSB     G^IOC$DIAGBUFILL        ;FILL DIAGNOSTIC BUFFER IF PRESENT
                          0513           RELCHAN                         ;RELEASE CHANNEL IF OWNED
   0092 C5        0A  91  0519           CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                  1A  1A  051E           BGTRU   10$                     ;IF GTRU YES
   0092 C5        13  91  0520           CMPB    #CDF_READPRESET,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                  13  1B  0525           BLEQU   10$                     ;IF LEQU YES
   0092 C5        11  91  0527           CMPB    #CDF_AVAILABLE,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                  0C  13  052C           BEQL    10$                     ;IF EQL YES
     52       58 A5  D0  052E           MOVL    UCB$L_IRP(R5),R2        ;RETRIEVE ADDRESS OF IRP
          00D8 C5     A1  0532           ADDW3   UCB$L_DR_BCR(R5),-
   02 AE      32 A2       0536                   IRP$W_BCNT(R2),2(SP)    ; Calculate bytes transfered
                  51  D4  053A     10$:  CLRL    R1                      ;CLEAR SECOND STATUS LONGWORD
                  50 8ED0 053C           POPL    R0                      ;RETRIEVE FINAL REQUEST STATUS
    53     0091 C5    9A  053F           MOVZBL  UCB$B_SLAVE+1(R5),R3    ;GET DRIVE OFFSET CONSTANT
  53     0400 C443    DE  0544           MOVAL   MBA$L_ERB(R4)[R3],R3    ;GET ADDRESS OF DRIVE REGISTERS
          63      09  9A  054A           MOVZBL  #F_DRVCLR!1,RM_CS1(R3)  ; Issue a drive clear before release.
          63      0B  9A  054D           MOVZBL  #F_RELEASE!1,RM_CS1(R3) ;RELEASE PORT
                          0550           REQCOM                          ;COMPLETE REQUEST
```

```
0556          .SBTTL  HARDWARE FUNCTION EXECUTION
0556  ;
0556  ; FEX - HARDWARE FUNCTION EXECUTION
0556  ;
0556  ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
0556  ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
0556  ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
0556  ; INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. IF THE FUNCTION IS AN
0556  ; IMMEDIATE FUNCTION CONTROL RETURNS IMMEDIATELY. ELSE THE RETURN ADDRESS
0556  ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
0556  ; RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
0556  ;
0556  ; INPUTS:
0556  ;
0556  ;       R0 = FUNCTION TABLE DISPATCH INDEX.
0556  ;       R3 = ADDRESS OF DRIVE CONTROL STATUS REGISTER 1.
0556  ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
0556  ;       R5 = DEVICE UNIT UCB ADDRESS.
0556  ;
0556  ;       00(SP) = RETURN ADDRESS OF CALLER.
0556  ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
0556  ;
0556  ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
0556  ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
0556  ;
0556  ; OUTPUTS:
0556  ;
0556  ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
0556  ;
0556  ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
0556  ;               OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
0556  ;               ERROR ROUTINE.
0556  ;
0556  ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
0556  ;               ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
0556  ;               INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
0556  ;
0556  ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
0556  ;               OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
0556  ;               IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
0556  ;
0556  ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
0556  ;               DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
0556  ;
0556  ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
0556  ;
0556  ;       IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
0556  ;       THE GENERAL REGISTERS R0, R1, AND R2, AND THE UCB.
0556  ;
0556  ;       R0 = DRIVE STATUS REGISTER.
0556  ;       R1 = MBA STATUS REGISTER.
0556  ;       R2 = DRIVE ERROR REGISTER 1.
0556  ;
0556  ;       UCB$W_EC1(R5) = ECC POSITION REGISTER.
0556  ;       UCB$W_EC2(R5) = ECC PATTERN REGISTER.
0556  ;       UCB$W_BCR(R5) = BYTE COUNT REGISTER.
0556  ;       UCB$W_DR_ER2(R5) = DRIVE ERROR REGISTER 2.
```

```
                                 0556  ;
                                 0556
                                 0556  FEX:                                         ;FUNCTION EXECUTOR
                 009C C5 8ED0    0556          POPL    UCB$L_DPC(R5)                ;SAVE DRIVER PC VALUE
              0093 C5   50   90  055B          MOVB    R0,UCB$B_CEX(R5)             ;SAVE CASE INDEX
              53   0091 C5   9A  0560          MOVZBL  UCB$B_SLAVE+1(R5),R3         ;GET DRIVE OFFSET CONSTANT
            53   0400 C443   DE  0565          MOVAL   MBA$L_ERB(R4)[R3],R3         ;GET ADDRESS OF DRIVE REGISTERS
        38 A5   00008000 8F  D3  056B          BITL    #DEV$M_DUA,UCB$L_DEVCHAR(R5) ;DUAL PORTED DRIVE?
                        48   12  0573          BNEQ    SEIZE                        ;IF NEQ, YES
              50   0093 C5   9A  0575  GO:     MOVZBL  UCB$B_CEX(R5),R0             ;Restore case index (func. code)
                                 057A          CASE    R0,<-                        ;DISPATCH TO PROPER FUNCTION ROUTINE
                                 057A                  POSIT,-                      ;SEEK CYLINDER
                                 057A                  EXFNC,-                      ;RECALIBRATE
                                 057A                  IMMED,-                      ;DRIVE CLEAR
                                 057A                  IMMED,-                      ;RELEASE DRIVE
                                 057A                  IMMED,-                      ;OFFSET HEADS
                                 057A                  EXFNC,-                      ;RETURN TO CENTERLINE
                                 057A                  IMMED,-                      ;PACK ACKNOWLEDGE
                                 057A                  POSIT,-                      ;SEARCH FOR SECTOR
                                 057A                  XFER,-                       ;WRITE CHECK
                                 057A                  XFER,-                       ;WRITE DATA
                                 057A                  XFER,-                       ;READ DATA
                                 057A                  XFER,-                       ;WRITE HEADER AND DATA
                                 057A                  XFER,-                       ;READ HEADER AND DATA
                                 057A                  XFER,-                       ;WRITE TRACK DESCRIPTOR
                                 057A                  XFER,-                       ;READ TRACK DESCRIPTOR
                                 057A                  IMMED,-                      ;AVAILABLE
                                 057A                  XFER,-                       ;WRITE CHECK HEADER AND DATA
                                 057A                  IMMED,-                      ;READIN PRESET
                                 057A                  EXFNC,-                      ;DIAGNOSE
                                 057A                  SEARCHA,-                    ;SEARCH AHEAD
                                 057A                  >,LIMIT=#CDF_SEEK            ;
                                 05A6
                                 05A6  ;
                                 05A6  ; IMMEDIATE FUNCTION EXECUTION
                                 05A6  ;
                                 05A6  ;       FUNCTIONS INCLUDE:
                                 05A6  ;
                                 05A6  ;               NO OPERATION,
                                 05A6  ;               DRIVE CLEAR,
                                 05A6  ;               RELEASE PORT,
                                 05A6  ;               OFFSET,
                                 05A6  ;               READ IN PRESET, AND
                                 05A6  ;               PACK ACKNOWLEDGE.
                                 05A6  ;
                                 05A6  ; Two other functions which might (but hopefully don't) pass through this code
                                 05A6  ; are UNLOAD and AVAILABLE.  If such functions get here they are treated as
                                 05A6  ; NOPs.
                                 05A6  ;
                                 05A6  ; THESE FUNCTIONS ARE EXECUTED IMMEDIATELY AND THE FINAL DEVICE REGISTERS
                                 05A6  ; ARE RETURNED TO THE CALLER.
                                 05A6  ;
                                 05A6
                                 05A6  IMMED:                                       ;IMMEDIATE FUNCTION EXECUTION
                                 05A6          DSBINT                               ;DISABLE INTERRUPTS
              09 64 A5   05  E0  05AC          BBS     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF SET, POWER HAS FAILED
                    63   09  9A  05B1          MOVZBL  #F_DRVCLR!1,RM_CS1(R3)  ;CLEAR DRIVE ERRORS
```

```
      63    FADF CF40  9A  05B4                   MOVZBL  FTAB[R0],RM_CS1(R3)       ;EXECUTE FUNCTION
                 010B  31  05BA           10$:    BRW     ENBXIT                    ;
                           05BD
                           05BD           ;
                           05BD           ; ATTEMPT TO SEIZE THE PORT ON A DUAL PORTED DISK.
                           05BD           ;
                           05BD
      B2 00D4 C5    03  E1  05BD          SEIZE:  BBC     #DR_V_DUALPORT, -          ; IF CLEAR, DUALPORT KIT
                           05C3                           UCB$B_DR_SSTS(R5),GO      ;IS NOT PRESENT
      51    00000064 8F  D0  05C3                 MOVL    #100,R1                   ; Initialize count for the number of
                           05CA                                                     ;   times we will accept the loss of
                           05CA                                                     ;   the port while we are on the I/O
                           05CA                                                     ;   fork queue.
                           05CA           2$:     DSBINT                            ;DISABLE INTERRUPTS
                 04 A3  D4  05D0                  CLRL    RM_DS(R3)                 ;ATTEMPT TO SEIZE PORT
           00000100 8F  D3  05D3                  BITL    #RM_DS_M_DPR,-            ;DID WE SEIZE THE PORT?
                 04 A3      05D9                          RM_DS(R3)
                    16  12  05DB                  BNEQ    4$                        ;IF NEQ, WE SEIZED THE PORT
                            05DD                  WFIKPCH RETREG,#15                ;LETS WAIT FOR THE PORT, ELSE TIMEOUT
                            05E7                  IOFORK                            ;CREATE FORK PROCESS
                 DA 51  F4  05ED                  SOBGEQ  R1,2$                     ; Loop to make sure we really still
                            05F0                                                    ;   have the port after we are dequeued
                            05F0                                                    ;   off the I/O fork queue.
                  00D8  31  05F0                  BRW     RETREG                    ; Otherwise, error - We keep losing the
                            05F3                                                    ;   port and we've retried enough.
                            05F3          4$:     ENBINT                            ;ENABLE INTERRUPTS
                  FF7C  31  05F6                  BRW     GO                        ;LET'S CONTINUE, WE HAVE THE PORT
                            05F9
                            05F9          ;
                            05F9          ; SEARCH AHEAD FUNCTION EXECUTION
                            05F9          ;
                            05F9          ; THIS FUNCTION MINIMIZES ROTATIONAL LATENCY BY SEARCHING FOR THE SECTOR THAT IS
                            05F9          ; FOUR SECTORS AHEAD OF THE STARTING SECTOR OF A TRANSFER.
                            05F9          ;
                            05F9          ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                            05F9          ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                            05F9          ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                            05F9          ;
                            05F9
                            05F9          SEARCHA:                                  ;
      51    00BC C5     3C  05F9                  MOVZWL  UCB$W_DA(R5),R1           ;GET DESIRED TRACK AND SECTOR ADDRESS
            51   04     82  05FE                  SUBB    #4,R1                     ;COMPUTE FOUR SECTORS BEFORE IT
                 04     18  0601                  BGEQ    10$                       ;IF GEQ BEFORE SECTOR ZERO
      51    44   A5     80  0603                  ADDB    UCB$B_SECTORS(R5),R1      ;CONVERT TO AFTER SECTOR ZERO
      14 A3      51     D0  0607          10$:    MOVL    R1,RM_DA(R3)              ;SET TRACK AND SECTOR ADDRESS
                 15     11  060B                  BRB     LDCYL                     ;
                            060D
                            060D          ;
                            060D          ; TRANSFER FUNCTION EXECUTION
                            060D          ;
                            060D          ;       FUNCTIONS INCLUDE:
                            060D          ;
                            060D          ;               WRITE TRACK DESCRIPTOR,
                            060D          ;               WRITE CHECK,
                            060D          ;               WRITE CHECK HEADER AND DATA,
                            060D          ;               WRITE DATA,
                            060D          ;               WRITE HEADER AND DATA,
                          060D  ;                     READ TRACK DESCRIPTOR,
                          060D  ;                     READ DATA, AND
                          060D  ;                     READ HEADER AND DATA.
                          060D  ;
                          060D  ; THE MAP REGISTERS, BYTE COUNT REGISTER, AND VIRTUAL ADDRESS REGISTER ARE
                          060D  ; LOADED FOLLOWED BY THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS.
                          060D  ; THE FUNCTION IS INITIATED AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
                          060D  ; INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                          060D  ;
                          060D  ; IT ASSUMED THAT THE CALLER OWNS THE CHANNEL ON WHICH THE I/O IS TO OCCUR.
                          060D  ;
                          060D
                          060D  XFER:                                        ;TRANSFER FUNCTION EXECUTION
       08 A4     00  D2   060D          MCOML   #0,MBA$L_SR(R4)              ;CLEAR MASSBUS ADAPTER ERRORS
                          0611          LOADMBA                              ;LOAD MAP, BYTE COUNT, AND VIRTUAL ADDRESS
    50    0093 C5    9A   0617          MOVZBL  UCB$B_CEX(R5),R0             ;RETRIEVE FUNCTION TABLE INDEX
                          061C
                          061C  ;
                          061C  ; POSITIONING FUNCTION EXECUTION
                          061C  ;
                          061C  ;       FUNCTIONS INCLUDE:
                          061C  ;
                          061C  ;               SEEK CYLINDER, AND
                          061C  ;               SEARCH FOR SECTOR.
                          061C  ;
                          061C  ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                          061C  ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                          061C  ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                          061C  ;
                          061C
                          061C  POSIT:                                       ;POSITION FUNCTION EXECUTION
 14 A3    00BC C5    3C   061C          MOVZWL  UCB$W_DA(R5),RM_DA(R3)       ;SET DESIRED TRACK AND SECTOR ADDRESS
                          0622  LDCYL:                                       ;
 28 A3    00BE C5    3C   0622          MOVZWL  UCB$W_DC(R5),RM_DC(R3)       ;SET DESIRED CYLINDER ADDRESS
                          0628
                          0628  ;
                          0628  ; INTERRUPT WAIT FUNCTION EXECUTION
                          0628  ;
                          0628  ;       FUNCTIONS INCLUDE:
                          0628  ;
                          0628  ;               DIAGNOSE,
                          0628  ;               RECALIBRATE, AND
                          0628  ;               RETURN TO CENTERLINE.
                          0628  ;
                          0628  ; THE OFFSET REGISTER IS LOADED, THE FUNCTION IS INITIATED, AND A WAITFOR
                          0628  ; INTERRUPT IS EXECUTED. WHEN THE INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS
                          0628  ; ARE RETURNED TO THE CALLER.
                          0628  ;
                          0628
                          0628  EXFNC:                                       ;EXECUTE FUNCTION
          63     09  9A   0628          MOVZBL  #F DRVCLR!1,RM CS1(R3)       ;CLEAR DRIVE ERRORS
   30 00D4 C5    01  E1   062B          BBC     #DR V OM,UCB$B DR SSTS(R5),10$   ;IF CLR, NO OFFSET ACTIVE
          63     0D  9A   0631          MOVZBL  #F OFFSET!1,RM CST(R3)       ;SET DRIVE IN OFFSET MODE
                 07  91   0634          CMPB    #DT$ RP07,-                  ; Is this drive an RP07?
             41 A5        0636                  UCB$B_DEVTYPE(R5)            ;
                 27  12   0638          BNEQ    10$                          ; If not, no need to wait.
                          063A          TIMEWAIT -                           ; Wait for 5 milliseconds or until
```

```
                              063A              TIME = #5000,-           ;  go bit clears.
                              063A              BITVAL = #1,-            ;
                              063A              SOURCE = RM_CS1(R3),-    ;
                              063A              CONTEXT = L,-            ;
                              063A              SENSE = .FALSE.          ;
                              0661 10$:   DSBINT                        ;DISABLE INTERRUPTS
   5C 64 A5     05      E0    0667        BBS     #UCB$V_POWER,UCB$W_STS(R5),ENBXIT ;IF SET, POWER FAILED
 0C A3   00D2 C5        3C    066C        MOVZWL  UCB$W_DR_MR(R5),RM_MR(R3)      ;SET DIAGNOSTIC INFO
 24 A3   00C8 C5        3C    0672        MOVZWL  UCB$W_OFFSET(R5),RM_OF(R3) ;SET FORMAT, INHIBIT BITS, AND OFFSET
                09      E1    0678        BBC     #RM_OF_V_SSEI,-        ; If Skip Sector Error Inhibit is clear
       0F 00C8 C5             067A                UCB$W_OFFSET(R5),12$   ;  branch around.
                09      E0    067E        BBS     #IO$V_SKPSECINH,-      ; If reason for SSEI is EXPLICIT user
       09 009A C5             0680                UCB$W_FUNC(R5),12$     ;  request, then branch around.
                02      8A    0684        BICB    #RM_OF_M_SSEI/256,-    ; If here, SSEI set due to error on
          00C9 C5             0686                UCB$W_OFFSET+1(R5)     ;  this track. Hence clear SSEI in
          00BC C5       97    0689        DECB    UCB$W_DA(R5)           ;  software only and decrement sector
                              068D                                       ;  bias in software also.
                              068D 12$:
 52    04 A3    13      78    068D        ASHL    #31-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
                1E      18    0692        BGEQ    15$                    ;IF GEQ NO
 63    F9FF CF40        9A    0694        MOVZBL  FTAB[R0],RM_CS1(R3)    ;INITIATE FUNCTION
                              069A        WFIKPCH RETREG,#6              ;WAITFOR INTERRUPT AND KEEP CHANNEL
 00CC C5   08 A4        D0    06A4        MOVL    MBA$L_SR(R4),UCB$L_DR_SR(R5) ;SAVE FINAL CONTROLLER STATUS
                              06AA        IOFORK                         ;CREATE FORK PROCESS
                19      11    06B0        BRB     RETREG                 ;
                              06B2
                              06B2 ;
                              06B2 ; MEDIUM OFFLINE AT START OF FUNCTION
                              06B2 ;
                              06B2
                              06B2 15$:   ENBINT                         ;ENABLE INTERRUPTS
          0093 C5       94    06B5        CLRB    UCB$B_CEX(R5)          ;FORCE DRIVE FUNCTION
    50    4000 8F       3C    06B9        MOVZWL  #RM_DS_M_ERR,R0        ;SET DRIVE ERROR
          00CC C5       D4    06BE        CLRL    UCB$L_DR_SR(R5)        ;CLEAR SAVED MBA STATUS REGISTER
          00D5 C5       96    06C2        INCB    UCB$B_DR_ERL(R5)       ;SET MEDIUM OFFLINE FOR ERROR LOG
                35      11    06C6        BRB     ERROR                  ;
                              06C8
                              06C8 ;
                              06C8 ; ENABLE INTERRUPTS
                              06C8 ;
                              06C8
                              06C8 ENBXIT:                               ;
                              06C8        ENBINT                         ;ENABLE INTERRUPTS
                              06CB
                              06CB ;
                              06CB ; RETURN REGISTERS
                              06CB ;
                              06CB
                              06CB RETREG:                               ;RETURN FINAL DEVICE REGISTERS
 00D0 C5    34 A3       F7    06CB        CVTLW   RM_ER2(R3),UCB$W_DR_ER2(R5) ;SAVE ERROR REGISTER 2
 00C4 C5    38 A3       F7    06D1        CVTLW   RM_EC1(R3),UCB$W_ECT(R5) ;SAVE ECC POSITION REGISTER
 00C6 C5    3C A3       F7    06D7        CVTLW   RM_EC2(R3),UCB$W_EC2(R5) ;SAVE ECC PATTERN REGISTER
                              06DD ; Here we save the more conservative of the two byte counts contained in
                              06DD ;      the MBA$L_BCR register.  The high word of this register is the
                              06DD ;      (negative of the) number of bytes transferred to or from the
                              06DD ;      drive, while the low word is the (negative of the) number of
                              06DD ;      bytes transferred to or memory.  On a read, the more conservative
                              06DD  ;             value is that of the number of bytes transferred to memory (low word)
                              06DD  ;             while on a write the more conservative value is the number of
                              06DD  ;             bytes transferred to the drive (high word).  Here we deposit
                              06DD  ;             the entire register into a longword in the UCB.  If the operation
                              06DD  ;             was a read we leave the value as is.  However if the operation
                              06DD  ;             was a write (or anything but a read) we move the high word to
                              06DD  ;             the low word in memory.  All other pieces of this driver use the
                              06DD  ;             low word of this longword as the valid byte count.
                              06DD
               10 A4     D0   06DD                MOVL    MBA$L_BCR(R4),-          ; Save entire byte count register
             00D8 C5          06E0                        UCB$L_DR_BCR(R5)         ;   in the UCB.
        50     58 A5     D0   06E3                MOVL    UCB$L_IRP(R5),R0         ; Retrieve IRP pointer.
                  01     E0   06E7                BBS     #IRP$V_FUNC,-            ; If we had a read operation then
            07 2A A0          06E9                        IRP$W_STS(R0),5$         ;   just branch around since all OK.
             00DA C5     B0   06EC                MOVW    UCB$L_DR_BCR+2(R5),-     ; If NOT read, then copy high word to
             00D8 C5          06F0                        UCB$L_DR_BCR(R5)         ;   low order word for later use.
                              06F3  5$:
  00D6 C5      24 A3     F7   06F3                CVTLW   RM_OF(R3),UCB$W_DR_OFR(R5)   ;SAVE OFFSET REGISTER
       50      04 A3     D0   06F9                MOVL    RM_DS(R3),R0             ;GET CONTENTS OF DRIVE STATUS REGISTER
                              06FD  ERROR:                                         ;
     51      00CC C5     D0   06FD                MOVL    UCB$L_DR_SR(R5),R1       ;RETRIEVE FINAL CONTROLLER STATUS
       52      08 A3     D0   0702                MOVL    RM_ERT(R3),R2            ;GET CONTENTS OF DRIVE ERROR REGISTER 1
  64 A5      0060 8F     B3   0706                BITW    #UCB$M_POWER!-           ;POWER FAIL OR DEVICE TIMEOUT?
                              070C                        UCB$M_TIMOUT,UCB$W_STS(R5) ;
                  03     13   070C                BEQL    10$                      ;IF EQL - NO
                0137     31   070E                BRW     SPECOND                  ;BRANCH TO SPECIAL CONDITION
  0093 C5         15     91   0711  10$:          CMPB    #CDF_SEARCHA, UCB$B_CEX(R5) ;Search Ahead?
                  5F     13   0716                BEQL    SAFUNC                   ;Branch to special search-ahead code.
  0093 C5         0A     91   0718                CMPB    #CDF_WRITECHECK,UCB$B_CEX(R5)  ;DRIVE RELATED FUNCTION?
                  07     1A   071D                BGTRU   20$                      ;IF GTRU - YES
  0093 C5         13     91   071F                CMPB    #CDF_READPRESET,UCB$B_CEX(R5) ;DRIVE RELATED FUNCTION?
                  03     1A   0724                BGTRU   30$                      ;IF GTRU - NO
                0083     31   0726  20$:          BRW     DFUNC                    ;DRIVE FUNCTION
                              0729
                              0729  ;
                              0729  ; CHECK FOR RM80 SKIP SECTOR ERROR
                              0729  ;
                              0729
     41 A5        0D     91   0729  30$:          CMPB    #DT$_RM80,UCB$B_DEVTYPE(R5)   ;RM80?
                  56     12   072D                BNEQ    CFUNC                    ;IF NEQ - NO
                  05     E1   072F                BBC     #RM_ER2_V_SSE,-          ; If clear, No Skip Sectoring error,
          50 00D0 C5          0731                        UCB$W_DR_ER2(R5),CFUNC   ;   so branch around.
     52    0180 8F       B3   0735                BITW    #RM_ER1_M_HCE!RM_ER1_M_HCRC,R2   ;HEADER COMPARE OR HDR CRC ERR?
                  49     12   073A                BNEQ    CFUNC                    ;IF NEQ - YES
  00C9 C5         02     88   073C                BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)   ;SET SKIP SECTOR INHIBIT
                  50     DD   0741                PUSHL   R0                       ;SAVE R0 (DRIVE STATUS REGISTER)
  50       00D8 C5       3C   0743                MOVZWL  UCB$L_DR_BCR(R5),R0      ; Get negative bytes remaining
                  05     12   0748                BNEQ    40$                      ;IF NEQ - PARTIAL TRANSFER
  50       FFFF 8F       B0   074A                MOVW    #-1,R0                   ;FAKE A PARTIAL TRANSFER
     50      7E A5       A0   074F  40$:          ADDW    UCB$W_BCNT(R5),R0        ;CALCULATE BYTES TRANSFERRED
  50       01FF 8F       AA   0753                BICW    #^X1FF,R0                ;TRUNCATE TO LAST BLOCK TRANSFERRED
     00000000'GF         16   0758                JSB     G^IOC$UPDATRANSP         ;UPDATE TRANSFER PARAMETERS
             00BC C5     96   075E                INCB    UCB$W_DA(R5)             ;INCREMENT TO NEXT SECTOR
                 50    8ED0   0762                POPL    R0                       ;RESTORE R0 (DRIVE STATUS REGISTER)
  03 00D4 C5      00     E0   0765                BBS     #DR_V_DCK,UCB$B_DR_SSTS(R5),45$   ;IF SET - DATACHECK IN PROGRESS
                FB22     31   076B                BRW     TRANNOCH                 ;RESTART TRANSFER
                              076E  45$:          RELCHAN                          ;RELEASE CHANNEL
             FB70   31  0774            BRW     CHECKRETRY              ;RESTART DATA CHECK
                        0777                                            ;NO SSE - CHECK FOR TRACK-TRACK SSEI CLR
                        0777            .ENABL  LSB
                        0777
                        0777 ;
                        0777 ; SEARCH AHEAD ERROR CHECKING
                        0777 ;
                        0777
                        0777 SAFUNC:
     7A 00D0 C5    0E   E1  0777        BBC     #RM_ER2_V_SKI, -        ;The only error worth checking on
                        077D                    UCB$W_DR_ER2(R5), 30$   ;seach-ahead is seek incomplete.
        00000000'GF    16   077D        JSB     G^ERL$DEVICERR          ;SKI errors, however, must be logged
                   68  11   0783        BRB     25$                     ;and retried.
                        0785
                        0785 ;
                        0785 ; CONTROLLER RELATED FUNCTION
                        0785 ;
                        0785
                        0785 CFUNC:
51    000E5FFF 8F      D3   0785        BITL    #MBA$M_ERROR,R1         ;ANY CONTROLLER ERRORS?
                   69  13   078C        BEQL    30$                     ;IF EQL NO
        00000000'GF    16   078E        JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
     66 009A C5    0F  E0   0794        BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),40$ ;IF SET, RETRY INHIBITED
51    0008000B 8F      D3   079A        BITL    #MBA$M_SR_ERCONF!-      ;ERROR CONFIRMATION OR,
                        07A1                    MBA$M_SR_ISTO!-         ;INTERFACE SEQUENCE TIMEOUT OR,
                        07A1                    MBA$M_SR_PGE!-          ;PROGRAMMING ERROR OR,
                        07A1                    MBA$M_SR_RDTO,R1        ;READ TIMEOUT?
                   5D  12   07A1        BNEQ    40$                     ;IF NEQ YES - FATAL CONTROLLER ERROR
51    00064FF4 8F      D3   07A3        BITL    #MBA$M_SR_DLT!-         ;DATA LATE OR,
                        07AA                    MBA$M_SR_INVMAP!-       ;INVALID MAP REGISTER OR,
                        07AA                    MBA$M_SR_MAPPE!-        ;MAP REGISTER PARITY ERROR OR,
                        07AA                    MBA$M_SR_MBEXC!-        ;MASSBUS EXCEPTION OR,
                        07AA                    MBA$M_SR_MCPE!-         ;MASSBUS CONTROL PARITY ERROR OR,
                        07AA                    MBA$M_SR_SPE!-          ;MBA SILO PARITY ERROR OR,
                        07AA                    MBA$M_SR_MDPE!-         ;MASSBUS DATA PARITY ERROR OR,
                        07AA                    MBA$M_SR_MXF!-          ;MISSED TRANSFER OR,
                        07AA                    MBA$M_SR_NED!-          ;NONEXISTENT DRIVE OR,
                        07AA                    MBA$M_SR_RDS!-          ;READ DATA SUBSTITUTE OR,
                        07AA                    MBA$M_SR_WCKLWR!-       ;WRITE CHECK LOWER BYTE OR,
                        07AA                    MBA$M_SR_WCKUPR,R1      ;WRITE CHECK UPPER BYTE?
                   1B  12   07AA        BNEQ    20$                     ;IF NEQ YES - RETRIABLE CONTROLLER ERROR
                        07AC
                        07AC ;
                        07AC ; DRIVE RELATED FUNCTION
                        07AC ;
                        07AC
                        07AC DFUNC:
        47 50      0E  E1   07AC 10$:   BBC     #RM_DS_V_ERR,R0,30$     ;IF CLR, NO DRIVE ERRORS
           7E A5       AE   07B0        MNEGW   UCB$W_BCNT(R5),-
             00D8 C5        07B3                UCB$L_DR_BCR(R5)        ; Reset byte count - NO TRANSFER
     45 00D5 C5        E8   07B6        BLBS    UCB$B_DR_ERL(R5),40$    ; Don't log error if Medium offline at
                        07BB                                            ;   start of function.
        00000000'GF    16   07BB        JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
     39 009A C5    0F  E0   07C1        BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),40$ ;IF SET, RETRY INHIBITED
        35 50      0C  E1   07C7 20$:   BBC     #RM_DS_V_MOL,R0,40$     ;IF CLR, MEDIUM OFFLINE
        31 50      06  E1   07CB        BBC     #RM_DS_V_VV,R0,40$      ;IF CLR, INVALID VOLUME
52      0180 8F        B3   07CF        BITW    #RM_ERT_M_HCRC!-        ; Check HCRC and HCE before checking
```

```
                        07D4                    RM_ER1_M_HCE,R2                 ;  BSE and FER.
           17      12   07D4            BNEQ    25$                             ; NEQ means HCRC or HCE is set.
  52  0E17 8F      B3   07D6            BITW    #RM_ER1_M_AOE!-                 ;ADDRESS OVERFLOW OR,
                        07DB                    RM_ER1_M_FER!-                  ;FORMAT ERROR OR,
                        07DB                    RM_ER1_M_IAE!-                  ;INVALID ADDRESS OR,
                        07DB                    RM_ER1_M_ILF!-                  ;ILLEGAL FUNCTION OR,
                        07DB                    RM_ER1_M_ILR!-                  ;ILLEGAL REGISTER OR,
                        07DB                    RM_ER1_M_RMR!-                  ;REGISTER MODIFY REFUSE OR,
                        07DB                    RM_ER1_M_WLE,R2                 ;WRITE LOCK ERROR?
           23      12   07DB            BNEQ    40$                             ;IF NEQ YES - FATAL DRIVE ERROR
0000 C5  A000 8F   B3   07DD            BITW    #RM_ER2_M_BSE!-                 ;BAD SECTOR ERROR OR,
                        07E4                    RM_ER2_M_OPE,UCB$W_DR_ER2(R5) ;OPERATOR PLUG ERROR?
           1A      12   07E4            BNEQ    40$                             ;IF NEQ YES - FATAL DRIVE ERROR
  52  4000 8F      B3   07E6            BITW    #RM_ER1_M_UNS,R2                ; Is the drive unsafe?
           16      12   07EB            BNEQ    45$                             ; Branch if so.
                        07ED
                        07ED    ;
                        07ED    ; RETRIABLE ERROR EXIT
                        07ED    ;
                        07ED
  7E  009C D5      32   07ED    25$:    CVTWL   @UCB$L_DPC(R5),-(SP)            ;GET BRANCH DISPLACEMENT
009C C5    8E      C0   07F2            ADDL    (SP)+,UCB$L_DPC(R5)             ;CALCULATE RETURN ADDRESS - 2
009C C5    02      C0   07F7    30$:    ADDL    #2,UCB$L_DPC(R5)                ;SKIP PAST BRANCH DISPLACEMENT WORD
      009C D5      17   07FC            JMP     @UCB$L_DPC(R5)                  ;RETURN TO DRIVER
                        0800
                        0800    ;
                        0800    ; FATAL CONTROLLER OR DRIVE ERROR EXIT
                        0800    ;
                        0800
         FC70      31   0800    40$:    BRW     FATALERR                        ;
                        0803
                        0803    ;
                        0803    ; Check for unsafe condition and attempt to clear it.
                        0803    ;
                        0803
                        0803    45$:    DSBINT                                  ; Disable interupts.
           05      E1   0809            BBC     #UCB$V_POWER,-                  ; Branch if no power failure occured.
     03 64 A5           080B                    UCB$W_STS(R5),47$               ;
         FEB7      31   080E            BRW     ENBXIT                          ; Otherwise, enable interupts and
                        0811                                                    ;  go process error.
   63      09      9A   0811    47$:    MOVZBL  #F_DRVCLR!1,RM_CS1(R3)          ; Attempt to clear unsafe condition.
                        0814            TIMEWAIT -                              ; Wait for ten microseconds or until
                        0814                    TIME = #1,-                     ;  unsafe condition clears.
                        0814                    BITVAL = #RM_ER1_M_UNS,-        ;
                        0814                    SOURCE = RM_ER1(R3),-           ;
                        0814                    CONTEXT = L,-                   ;
                        0814                    SENSE = .FALSE.                 ;
                        083C            ENBINT                                  ; Enable interupts.
  52    08 A3      D0   083F            MOVL    RM_ER1(R3),R2                   ; Retrieve error status.
        A7 50      E8   0843            BLBS    R0,25$                          ; Branch if drive is no longer unsafe.
           B8      11   0846            BRB     40$                             ; Otherwise, fatal error.
                        0848
                        0848    ;
                        0848    ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
                        0848    ;
                        0848
                        0848    SPECOND:
```

```
   61 64 A5     05     E4  0848  50$:    BBSC    #UCB$V_POWER,UCB$W_STS(R5),70$ ;IF SET, POWER FAILURE
                           084D
                           084D  ;
                           084D  ; DEVICE TIME OUT
                           084D  ;
                           084D
     00000000'GF       16  084D          JSB     G^ERL$DEVICTMO              ;LOG DEVICE TIME OUT
     53    24 A5       D0  0853          MOVL    UCB$L_CRB(R5),R3            ;GET ADDRESS OF CRB
     53    2C A3       D0  0857          MOVL    CRB$L_INTD+VEC$L_IDB(R3),R3 ;GET ADDRESS OF IDB
     04 A3     55      D1  085B          CMPL    R5,IDB$L_OWNER(R3)          ;DEVICE OWN CONTROLLER?
               22      12  085F          BNEQ    60$                         ;IF NEQ NO
                           0861          DSBINT                              ;DISABLE INTERRUPTS
               06      D0  0867          MOVL    #MBA$M_CR_ABORT!MBA$M_CR_IE,- ;ABORT THE DATA TRANSFER
            04 A4          0869                  MBA$L_CR(R4)                ;
                           086B          WFIKPCH 55$,#T5                     ;WAIT FOR ABORT AND KEEP CHANNEL
                           0875          IOFORK                              ;CREATE FORK PROCESS
                           087B  55$:
     04 A4      01     D0  087B          MOVL    #MBA$M_CR_INIT,MBA$L_CR(R4) ;INITIALIZE ENTIRE MBA
     04 A4      04     D0  087F          MOVL    #MBA$M_CR_IE,MBA$L_CR(R4)  ;ENABLE DEVICE INTERRUPTS
                           0883  60$:    SETIPL  UCB$B_FIPL(R5)              ;LOWER TO FORK LEVEL
    50     022C 8F     3C  0887          MOVZWL  #SS$_TIMEOUT,R0             ;SET DEVICE TIMEOUT STATUS
           0080 C5     97  088C          DECB    UCB$B_ERTCNT(R5)            ;ANY ERROR RETRIES REMAINING?
                 0F    13  0890          BEQL    RESETXFR                    ;IF EQL NO
                           0892          RELCHAN                             ;RELEASE CHANNEL IF OWNED
  64 A5    0040 8F     AA  0898          BICW    #UCB$M_TIMOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
               F919    31  089E          BRW     FDISPATCH                   ;
                           08A1
                           08A1  ;
                           08A1  ; RESET TRANSFER BYTE COUNT TO ZERO
                           08A1  ;
                           08A1
                           08A1  RESETXFR:                                   ;
      53    58 A5      D0  08A1          MOVL    UCB$L_IRP(R5),R3            ;RETRIEVE ADDRESS OF I/O PACKET
            32 A3      AE  08A5          MNEGW   IRP$W_BCNT(R3),-
           00D8 C5         08A8                  UCB$L_DR_BCR(R5)            ; Reset transfer byte count
               FC5D    31  08AB          BRW     FUNCXT                      ;
                           08AE
                           08AE  ;
                           08AE  ; POWER FAILURE
                           08AE  ;
                           08AE
                           08AE  70$:    RELCHAN                             ;RELEASE CHANNEL
      53    58 A5      D0  08B4          MOVL    UCB$L_IRP(R5),R3            ;RETRIEVE ADDRESS OF I/O PACKET
   78 A5    2C A3      7D  08B8          MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
               F886    31  08BD          BRW     DR_STARTIO                        ;
                           08C0          .DSABL  LSB
                           08C0
```

```
                          08C0          .SBTTL  REGISTER DUMP ROUTINE
                          08C0  ;
                          08C0  ; DR_REGDUMP - REGISTER DUMP ROUTINE
                          08C0  ;
                          08C0  ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                          08C0  ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
                          08C0  ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                          08C0  ;
                          08C0  ; INPUTS:
                          08C0  ;
                          08C0  ;       R0 = ADDRESS OF REGISTER SAVE BUFFER.
                          08C0  ;       R4 = ADDRESS OF ADAPTER CONFIGURATION REGISTER.
                          08C0  ;       R5 = DEVICE UNIT UCB ADDRESS.
                          08C0  ;
                          08C0  ; OUTPUTS:
                          08C0  ;
                          08C0  ;       THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                          08C0  ;
                          08C0
                          08C0  DR_REGDUMP:                                             ;REGISTER DUMP ROUTINE
            80   18   D0  08C0          MOVL    #<RM_EC2+4+MBA$L_BCR+4+8+4>/4,(R0)+ ;INSERT NUMBER OF DEVICE REGS
            80   64   D0  08C3          MOVL    MBA$L_CSR(R4),(R0)+       ;SAVE CONFIGURATION REGISTER
         80   04 A4   D0  08C6          MOVL    MBA$L_CR(R4),(R0)+        ;SAVE CONTROL REGISTER
       80   00CC C5   D0  08CA          MOVL    UCB$L_DR_SR(R5),(R0)+     ;SAVE STATUS REGISTER
         80   0C A4   D0  08CF          MOVL    MBA$L_VAR(R4),(R0)+       ;SAVE VIRTUAL ADDRESS REGISTER
         80   10 A4   D0  08D3          MOVL    MBA$L_BCR(R4),(R0)+       ;SAVE BYTE COUNT REGISTER
 51   F8 A0   08   09   EF  08D7          EXTZV   #9,#8,-8(R0),R1           ;GET FINAL MAP REGISTER NUMBER
     80   0800 C441   D0  08DD          MOVL    MBA$L_MAP(R4)[R1],(R0)+   ;SAVE FINAL MAP REGISTER CONTENTS
                 80   D4  08E3          CLRL    (R0)+                     ;ASSUME NO PREVIOUS MAP REGISTER
                 51   D7  08E5          DECL    R1                        ;CALCULATE PREVIOUS MAP REGISTER NUMBER
                 07   19  08E7          BLSS    10$                       ;IF LSS NONE
  FC A0   0800 C441   D0  08E9          MOVL    MBA$L_MAP(R4)[R1],-4(R0)  ;SAVE PREVIOUS MAP REGISTER CONTENTS
            51   10   9A  08F0  10$:    MOVZBL  #<RM_EC2+4>/4,R1          ;SET NUMBER OF DRIVE REGISTERS TO SAVE
       52   0091 C5   9A  08F3          MOVZBL  UCB$B_SLAVE+1(R5),R2      ;GET DRIVE OFFSET CONSTANT
     52   0400 C442   DE  08F8          MOVAL   MBA$L_ERB(R4)[R2],R2      ;GET ADDRESS OF DRIVE REGISTERS
            80   82   D0  08FE  20$:    MOVL    (R2)+,(R0)+               ;SAVE DRIVE REGISTER
              FA 51   F5  0901          SOBGTR  R1,20$                    ;ANY MORE TO SAVE?
       80   00D5 C5   9A  0904          MOVZBL  UCB$B_DR_ERL(R5),(R0)+    ;SAVE MEDIUM OFFLINE INDICATOR
                      05  0909          RSB                               ;
                          090A
```

```
090A          .SBTTL  DISK DRIVE INITIALIZATION
090A    ;
090A    ; DR_UNIT_INIT - DISK DRIVE UNIT INITIALIZATION
090A    ;
090A    ; THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
090A    ; DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
090A    ;
090A    ; INPUTS:
090A    ;
090A    ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
090A    ;       R5 = DEVICE UNIT UCB ADDRESS.
090A    ;
090A    ; OUTPUTS:
090A    ;
090A    ;       UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
090A    ;
090A    ; SPECIAL NOTES:
090A    ;       This routine performs several special operations to support power
090A    ;       failure recovery in the RP07.  To provide an understanding of these
090A    ;       operations, power failure recovery within in the RP07 is discussed
090A    ;       first.  Then, the special actions taken by this routine are discussed.
090A    ;
090A    ;       The power up sequence in a RP07 drive is best described in terms of a
090A    ;       series of numbered states.  The state numbers are shown in the LED
090A    ;       readout on the micro-processor control pannel, the section labeled
090A    ;       "PROGRAM CODE."  The following lists these states and gives the
090A    ;       author's understanding what they mean.
090A    ;
090A    ;       STATE             COMMENTS
090A    ;       00,11,22,....,FF  These states occur upon restoration of DC power.
090A    ;                         Presumably they are related to micro-processor
090A    ;                         initialization and testing.  During these states,
090A    ;                         no MASSBUS interaction with the drive is possible.
090A    ;
090A    ;       00,01,02          These states occur after the above states during power
090A    ;                         failure recovery or after the START/STOP switch is
090A    ;                         moved from the STOP to the START position.  These
090A    ;                         states also are related to micro-processor and disk
090A    ;                         system testing.  The disk system is not spinning
090A    ;                         during these states.  During these states, no
090A    ;                         MASSBUS interaction with the drive is possible.
090A    ;
090A    ;       03                During power failure recovery, this is the state in
090A    ;                         which multiple RP07 drives on a single system will
090A    ;                         synchronize their attempts to spin their disk media.
090A    ;                         Limited communication with the drive via the MASSBUS
090A    ;                         is enabled while the drive is in this state.  The
090A    ;                         drive type register can be read, and the clear-drive
090A    ;                         command is accepted.  The drive status register also
090A    ;                         can be read while the drive is in this state.  For
090A    ;                         between 20 and 40 milliseconds after this state is
090A    ;                         entered, however, the drive status register contains
090A    ;                         garbage -- probably all bits except ATA and ERR set,
090A    ;                         a remnant of some internal test.  After this initial
090A    ;                         period, the drive status register contains reasonable,
090A    ;                         valid information.
090A    ;
```

```
                              090A  ;          04,05,06,07,08  These states occur while the disk medium is spining
                              090A  ;                          upto speed.  While in these states, no MASSBUS
                              090A  ;                          interaction with the drive is possible.
                              090A  ;
                              090A  ;          The following aspects of this routine relate specificially to dealing
                              090A  ;          with power failure recovery as practiced by the RP07.
                              090A  ;
                              090A  ;              o    The sieze port operation, performed near the beginning of this
                              090A  ;                   routine, also has the effect of waiting for the RP07 drive to
                              090A  ;                   reach state 03.  To allow both wait operations -- the sieze
                              090A  ;                   port function and the wait for RP07 to reach state 03 function
                              090A  ;                   -- to be combined, EXE$PWRTIMCHK is used to time both
                              090A  ;                   functions.  When this routine is called for reasons other than
                              090A  ;                   power failure recovery, it establishes a 20 millisecond wait
                              090A  ;                   interval for EXE$PWRTIMCHK.
                              090A  ;
                              090A  ;              o    Once access to the RP07 has been established, this routine
                              090A  ;                   proceeds to determine the drive type, that register can be
                              090A  ;                   read and contains valid.
                              090A  ;
                              090A  ;              o    Before proceeding to test for medium-online, however, this
                              090A  ;                   routine waits for 50 milliseconds.  This is intended to allow
                              090A  ;                   the drive status register to reach a valid state.
                              090A  ;
                              090A  ;              o    The medium-online test will wait for the drive to spin up.
                              090A  ;                   Because all drive registers show zero while MASSBUS access to
                              090A  ;                   the drive is disabled, it will correctly wait throughout
                              090A  ;                   states 03, 04, 05, 06, 07, and 08.
                              090A  ;
                              090A
                              090A  DR_UNIT_INIT:                                      ;DISK DRIVE UNIT INITIALIZATION
         53    54 A5    3C    090A             MOVZWL  UCB$W_UNIT(R5),R3               ;GET DRIVE UNIT NUMBER
      0090 C5      53   90    090E             MOVB    R3,UCB$B_SLAVE(R5)              ;SET SLAVE UNIT NUMBER
         53        20   C4    0913             MULL    #<1@7>/4,R3                     ;CALCULATE DRIVE OFFSET CONSTANT
      0091 C5      53   90    0916             MOVB    R3,UCB$B_SLAVE+1(R5)            ;SET SLAVE OFFSET CONSTANT
     53    0400 C443    DE    091B             MOVAL   MBA$L_ERB(R4)[R3],R3            ;GET ADDRESS OF DRIVE CONTROL REGISTER
    7E     00000000'GF  D0    0921             MOVL    G^EXE$GL_PWRDONE, -(SP) ;Save current powerfail limit time.
                   0E   12    0928             BNEQ    105$                    ;Non-zero value indicates powerfail.
           00000000'GF  16    092A             JSB     G^EXE$READ_TODR         ;If not powerfail, construct our
00000000'GF     50 02   C1    0930             ADDL3   #2,R0,G^EXE$GL_PWRDONE ;limit time for port seizure.
                04 A3   D4    0938  105$:      CLRL    RM_DS(R3)               ;Attempt to seize port.
     52      04 A3 17   78    093B  110$:      ASHL    #31-RM_DS_V_DPR, RM_DS(R3), R2 ;Did we seize the port?
                   0F   19    0940             BLSS    120$                    ;If LSS, we seized the port.
           00000000'GF  16    0942             JSB     G^EXE$PWRTIMCHK         ;Wait for port to be siezed.
                F0 50   E8    0948             BLBS    R0, 110$                ;Branch if haven't waited long enough.
         64 A5     10   AA    094B             BICW    #UCB$M_ONLINE, UCB$W_STS(R5) ;If never get the port,
                   5D   11    094F             BRB     15$                     ;mark the drive offline and invalid.
     52      18 A3 14   78    0951  120$:      ASHL    #31-RM_DT_V_DRQ, RM_DT(R3), R2 ;Is there a dualport kit?
                   05   18    0956             BGEQ    5$                      ;If GEQ, no dualport kit; continue.
      00D4 C5      08   88    0958             BISB    #DR_M_DUALPORT, -       ;Else, set flag indicating that disk
                              095D                     UCB$B_DR_SSTS(R5)       ;has a dualport kit.
         64 A5     10   A8    095D  5$:        BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
                 00A5   30    0961             BSBW    DR_DTYPE                        ;CLASSIFY DRIVE TYPE
      45 64 A5     04   E1    0964             BBC     #UCB$V_ONLINE,UCB$W_STS(R5),15$ ;IF CLR, UNKNOWN DRIVE TYPE
      4B 64 A5     0B   E1    0969             BBC     #UCB$V_VALID,UCB$W_STS(R5),30$  ;IF CLR, VOLUME SOFTWARE INVALID
         41 A5     07   91    096E             CMPB    #DT$_RP07, UCB$B_DEVTYPE(R5) ;Is this a RP07?
                   27   12    0972             BNEQ    10$                     ; Branch if not a RP07.
```

```
                              0974  1761          .SHOW   MEB
                              0974  1762          TIMEWAIT -                           ; If this is a RP07,
                              0974  1763                  time = #5000, -              ; wait for it to finish its
                              0974  1764                  bitval = #0, -               ; cup of coffee.
                              0974  1765                  source = #0, -
                              0974  1766                  context = B
                 50    01  3C 0974                MOVZWL  #SS$_NORMAL,R0
51  00000000'GF  00001388 8F  C5 0977                MULL3   #5000,G^EXE$GL_TENUSEC,R1
                       7E  D4 0983                CLRL    -(SP)
                 00    00  93 0985       30010$: BITB    #0,#0
                       0F  12 0988                BNEQ    30011$
           6E  00000000'GF  D0 098A                MOVL    G^EXE$GL_UBDELAY,(SP)
                    FD 6E  F5 0991       30012$: SOBGTR  (SP),30012$
                    EE 51  F5 0994                SOBGTR  R1,30010$
                       50  D4 0997                CLRL    R0
                              0999       30011$:
                       8E  D5 0999                TSTL    (SP)+
                              099B  1767          .NOSHOW MEB
                 63    09  9A 099B  1768 10$:     MOVZBL  #F_DRVCLR!1,RM_CS1(R3)   ;CLEAR DRIVE
         52   04 A3    13  78 099E  1769          ASHL    #31-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
                       11  19 09A3  1770          BLSS    20$                      ;IF LSS YES
              00000000'GF  16 09A5  1771          JSB     G^EXE$PWRTIMCHK          ;CHECK FOR MAXIMUM TIME EXCEEDED
                    ED 50  E8 09AB  1772          BLBS    R0,10$                   ;IF LBS MORE TIME TO GO
       64 A5      0800 8F  AA 09AE  1773 15$:     BICW    #UCB$M_VALID,UCB$W_STS(R5) ;MARK THE VOLUME INVALID
                       03  11 09B4  1774          BRB     30$
                 63    13  9A 09B6  1775 20$:     MOVZBL  #F_PACKACK!1,RM_CS1(R3) ;ACKNOWLEDGE PACK
                 63    0B  9A 09B9  1776 30$:     MOVZBL  #F_RELEASE!1,RM_CS1(R3) ;CLEAR DRIVE
        08 A4      08 A4  C8 09BC  1777 40$:     BISL    MBA$L_SR(R4), MBA$L_SR(R4) ;CLEAR MBA STATUS
                       8E  D5 09C1  1778          TSTL    (SP)+                    ;If powerfail limit time was zero
                       06  12 09C3  1779          BNEQ    50$                      ;when we started, make sure its
              00000000'GF  D4 09C5  1780          CLRL    G^EXE$GL_PWRDONE         ;zero when we leave.
                       05 09CB  1781 50$:     RSB
```

```
                             09CC          .SBTTL  UNSOLICITED INTERRUPT ROUTINE
                             09CC  ;
                             09CC  ; DR_UNSOLNT - UNSOLICITED INTERRUPT ROUTINE
                             09CC  ;
                             09CC  ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED.
                             09CC  ;
                             09CC  ; INPUTS:
                             09CC  ;
                             09CC  ;       R4 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                             09CC  ;       R5 = DEVICE UNIT UCB ADDRESS.
                             09CC  ;
                             09CC  ; OUTPUTS:
                             09CC  ;
                             09CC  ;       IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
                             09CC  ;       UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
                             09CC  ;       CLASSIFIED.
                             09CC  ;
                             09CC
                             09CC  DR_UNSOLNT:                                              ;UNSOLICITED INTERRUPT
      53    0091 C5     9A   09CC          MOVZBL  UCB$B_SLAVE+1(R5),R3                     ;GET DRIVE OFFSET CONSTANT
   53    0400 C443      DE   09D1          MOVAL   MBA$L_ERB(R4)[R3],R3                     ;GET ADDRESS OF DRIVE CONTROL REGISTER
        64 A5     10    A8   09D7          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
              002B      30   09DB          BSBW    DR_DTYPE                                 ;CLASSIFY DRIVE TYPE
     1F 64 A5     04    E1   09DE          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),10$ ;IF CLR, UNKNOWN DRIVE TYPE
     20 64 A5     0B    E1   09E3          BBC     #UCB$V_VALID,UCB$W_STS(R5),20$ ;IF CLR, VOLUME SOFTWARE INVALID
   52    04 A3    13    78   09E8          ASHL    #31-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
                  13    18   09ED          BGEQ    10$                              ;IF GEQ NO
     07 64 A5     08    E1   09EF          BBC     #UCB$V_BSY,UCB$W_STS(R5),5$ ;We know the drive is online; thus,
      0093 C5     08    91   09F4          CMPB    #CDF_PACKACK,UCB$B_CEX(R5)   ;if busy doing a PACKACK function,
                  0D    13   09F9          BEQL    20$                          ;then don't clear software valid.
   52    04 A3    19    78   09FB  5$:     ASHL    #31-RM_DS_V_VV,RM_DS(R3),R2 ;VOLUME VALID?
                  06    19   0A00          BLSS    20$                              ;IF LSS YES
  64 A5    0800 8F      AA   0A02  10$:    BICW    #UCB$M_VALID,UCB$W_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
                        05   0A08  20$:    RSB                                      ;
```

```
                      0A09         .SBTTL  CLASSIFY DRIVE TYPE AND SET PARAMETERS
                      0A09  ;
                      0A09  ; RM_DTYPE - CLASSIFY DRIVE TYPE AND SET PARAMETERS
                      0A09  ;
                      0A09  ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
                      0A09  ; SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
                      0A09  ; SET UNIT PARAMETERS.
                      0A09  ;
                      0A09  ; INPUTS:
                      0A09  ;
                      0A09  ;       R3 = ADDRESS OF DRIVE CONTROL REGISTER.
                      0A09  ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                      0A09  ;       R5 = DEVICE UNIT UCB ADDRESS.
                      0A09  ;
                      0A09  ; OUTPUTS:
                      0A09  ;
                      0A09  ;       THE DRIVE TYPE REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
                      0A09  ;
                      0A09
                      0A09  DR_DTYPE:                                        ;CLASSIFY DRIVE TYPE AND SET PARAMETERS
         18 A3   DD   0A09          PUSHL   RM_DT(R3)                        ;READ DRIVE TYPE REGISTER
6E    FE00 8F    AA   0A0C          BICW    #^C<^X1FF>,(SP)                  ;CLEAR EXTRANEOUS BITS
52    F627 CF    9E   0A11          MOVAB   DR_DTDESC,R2                     ;GET ADDRESS OF DESCRIPTOR TABLE
      82    6E   B1   0A16  10$:    CMPW    (SP),(R2)+                       ;DRIVE TYPE MATCH?
            0E   13   0A19          BEQL    20$                              ;IF EQL YES
      52    0D   C0   0A1B          ADDL    #DR_DTDESCLEN-2,R2               ;ADVANCE TO NEXT ENTRY
            62   B5   0A1E          TSTW    (R2)                             ;END OF TABLE?
            F4   12   0A20          BNEQ    10$                              ;IF NEQ NO
   64 A5    10   AA   0A22          BICW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT OFFLINE
      52    0D   C2   0A26          SUBL    #DR_DTDESCLEN-2,R2               ;BACK UP TO LAST DRIVE DESCRIPTOR
   41 A5    82   90   0A29  20$:    MOVB    (R2)+,UCB$B_DEVTYPE(R5)  ;SET DEVICE TYPE
   44 A5    82   D0   0A2D          MOVL    (R2)+,UCB$L_DEVDEPEND(R5)  ;SET DISK PACK GEOMETRY
 00B0 C5    82   D0   0A31          MOVL    (R2)+,UCB$L_MAXBLOCK(R5);SET MAXIMUM BLOCKS PER PACK
 008C C5    62   D0   0A36          MOVL    (R2),UCB$L_MEDIA_ID(R5)  ;SET MEDIA IDENT
            8E   D5   0A3B          TSTL    (SP)+                            ;REMOVE DRIVE TYPE FROM STACK
                 05   0A3D          RSB                                      ;
                      0A3E  DR_END:                                          ;ADDRESS OF LAST LOCATION IN DRIVER
                      0A3E
                      0A3E          .END
```

```
$$$               = 00000020 R       02    DPT$M_SVP         = 00000002
$$OP              = 00000002               DPT$REINITAB        0000006A R       02
ACP$ACCESS          ********      X  03    DPT$TAB             00000000 R       02
ACP$DEACCESS        ********      X  03    DR$DDT              00000000 RG      03
ACP$MODIFY          ********      X  03    DRVCLR              00000254 R       03
ACP$MOUNT           ********      X  03    DR_DTDESC           0000003C R       03
ACP$READBLK         ********      X  03    DR_DTDESCLEN      = 0000000F
ACP$WRITEBLK        ********      X  03    DR_DTYPE            00000A09 R       03
APPLY_ECC           0000038E R       03    DR_END              00000A3E R       03
AT$_MBA           = 00000000               DR_FUNCTABLE        000000B2 R       03
AVAILABLE           00000245 R       03    DR_M_DCK          = 00000001
CDF_AVAILABLE     = 00000011               DR_M_DUALPORT     = 00000008
CDF_DIAGNOSE      = 00000014               DR_M_ECC_DEFER    = 00000010
CDF_DRVCLR        = 00000004               DR_M_NOECC        = 00000004
CDF_NOP           = 00000005               DR_M_OM           = 00000002
CDF_OFFSET        = 00000006               DR_REGDUMP          000008C0 R       03
CDF_PACKACK       = 00000008               DR_STARTIO          00000146 R       03
CDF_READDATA      = 0000000C               DR_UNIT_INIT        0000090A R       03
CDF_READHEAD      = 0000000E               DR_UNSOLNT          000009CC R       03
CDF_READPRESET    = 00000013               DR_V_DCK          = 00000000
CDF_READTRACKD    = 00000010               DR_V_DUALPORT     = 00000003
CDF_RECAL         = 00000003               DR_V_ECC_DEFER    = 00000004
CDF_RETCENTER     = 00000007               DR_V_NOECC        = 00000002
CDF_SEARCH        = 00000009               DR_V_OM           = 00000001
CDF_SEARCHA       = 00000015               DT$_RM03          = 00000006
CDF_SEEK          = 00000002               DT$_RM05          = 0000000F
CDF_WRITECHECK    = 0000000A               DT$_RM80          = 0000000D
CDF_WRITECHECKH   = 00000012               DT$_RP07          = 00000007
CDF_WRITEDATA     = 0000000B               DYN$C_DDB         = 00000006
CDF_WRITEHEAD     = 0000000D               DYN$C_DPT         = 0000001E
CDF_WRITETRACKD   = 0000000F               DYN$C_UCB         = 00000010
CFUNC               00000785 R       03    ECC                 00000331 R       03
CHECKRETRY          000002E7 R       03    EMB$L_DV_REGSAV   = 0000004E
CHECKTAB            00000038 R       03    ENBXIT              000006C8 R       03
CHECKXT             00000300 R       03    ERL$DEVICERR        ********      X  03
CRB$L_INTD        = 00000024               ERL$DEVICTMO        ********      X  03
DATACHECK           0000029A R       03    ERROR               000006FD R       03
DC$_DISK          = 00000001               EXE$GL_PWRDONE      ********      X  03
DDB$K_PACK        = 00000001               EXE$GL_TENUSEC      ********      X  03
DDB$L_ACPD        = 00000010               EXE$GL_UBDELAY      ********      X  03
DDB$L_DDT         = 0000000C               EXE$IOFORK          ********      X  03
DEFER_ECC           000003B4 R       03    EXE$LCLDSKVALID     ********      X  03
DEV$M_AVL         = 00040000               EXE$ONEPARM         ********      X  03
DEV$M_DIR         = 00000008               EXE$PWRTIMCHK       ********      X  03
DEV$M_DUA         = 00008000               EXE$READ_TODR       ********      X  03
DEV$M_ELG         = 00400000               EXE$SENSEMODE       ********      X  03
DEV$M_FOD         = 00004000               EXE$SETCHAR         ********      X  03
DEV$M_IDV         = 04000000               EXE$ZEROPARM        ********      X  03
DEV$M_NNM         = 00000200               EXFNC               00000628 R       03
DEV$M_ODV         = 08000000               FATALERR            00000473 R       03
DEV$M_RND         = 10000000               FDISPATCH           000001BA R       03
DEV$M_SHR         = 00010000               FEX                 00000556 R       03
DFUNC               000007AC R       03    FTAB                00000098 R       03
DIAGNOSE            0000028A R       03    FUNCTAB_LEN       = 00000094
DPT$C_LENGTH      = 00000038               FUNCXT              0000050B R       03
DPT$C_VERSION     = 00000004               F_AVAILABLE       = 00000000
DPT$INITAB          00000038 R       02    F_DIAGNOSE        = 0000001C
```

```
F_DRVCLR                            = 00000008              IOS_UNLOAD                          = 00000001
F_NOP                               = 00000000              IOS_VIRTUAL                         = 0000003F
F_OFFSET                            = 0000000C              IOS_WRITECHECK                      = 0000000A
F_PACKACK                           = 00000012              IOS_WRITECHECKH                     = 00000018
F_READDATA                          = 00000038              IOS_WRITEHEAD                       = 0000000D
F_READHEAD                          = 0000003A              IOS_WRITELBLK                       = 00000020
F_READPRESET                        = 00000010              IOS_WRITEPBLK                       = 0000000B
F_READTRACKD                        = 0000003C              IOS_WRITETRACKD                     = 0000000F
F_RECAL                             = 00000006              IOS_WRITEVBLK                       = 00000030
F_RELEASE                           = 0000000A              IOC$APPLYECC                          ********   X    03
F_RETCENTER                         = 0000000E              IOC$DIAGBUFILL                        ********   X    03
F_SEARCH                            = 00000018              IOC$LOADMBAMAP                        ********   X    03
F_SEARCHA                           = 00000018              IOC$MNTVER                            ********   X    03
F_SEEK                              = 00000004              IOC$RELCHAN                           ********   X    03
F_WRITECHECK                        = 00000028              IOC$REQCOM                            ********   X    03
F_WRITECHECKH                       = 0000002A              IOC$REQPCHANL                         ********   X    03
F_WRITEDATA                         = 00000030              IOC$RETURN                            ********   X    03
F_WRITEHEAD                         = 00000032              IOC$UPDATRANSP                        ********   X    03
F_WRITETRACKD                       = 00000034              IOC$WFIKPCH                           ********   X    03
GO                                    00000575 R      03    IRP$L_MEDIA                         = 00000038
IDB$L_OWNER                         = 00000004              IRP$L_SVAPTE                        = 0000002C
IMMED                                 000005A6 R      03    IRP$S_FCODE                         = 00000006
IO$V_COMMOD                         = 00000006              IRP$V_FCODE                         = 00000000
IO$V_DATACHECK                      = 0000000E              IRP$V_FUNC                          = 00000001
IO$V_DIAGNOSTIC                     = 00000008              IRP$V_PHYSIO                        = 00000008
IO$V_INHRETRY                       = 0000000F              IRP$W_BCNT                          = 00000032
IO$V_INHSEEK                        = 0000000C              IRP$W_FUNC                          = 00000020
IO$V_MOVETRACKD                     = 00000007              IRP$W_STS                           = 0000002A
IO$V_SKPSECINH                      = 00000009              LDCYL                                 00000622 R      03
IO$_ACCESS                          = 00000032              MASKH                               = 00000008
IO$_ACPCONTROL                      = 00000038              MASKL                               = 04000000
IO$_AVAILABLE                       = 00000011              MBA$L_BCR                           = 00000010
IO$_CREATE                          = 00000033              MBA$L_CR                            = 00000004
IO$_DEACCESS                        = 00000034              MBA$L_CSR                           = 00000000
IO$_DELETE                          = 00000035              MBA$L_ERB                           = 00000400
IO$_DIAGNOSE                        = 0000001D              MBA$L_MAP                           = 00000800
IO$_DRVCLR                          = 00000004              MBA$L_SR                            = 00000008
IO$_MODIFY                          = 00000036              MBA$L_VAR                           = 0000000C
IO$_MOUNT                           = 00000039              MBA$M_CR_ABORT                      = 00000002
IO$_NOP                             = 00000000              MBA$M_CR_IE                         = 00000004
IO$_OFFSET                          = 00000006              MBA$M_CR_INIT                       = 00000001
IO$_PACKACK                         = 00000008              MBA$M_ERROR                         = 000E5FFF
IO$_READHEAD                        = 0000000E              MBA$M_SR_DLT                        = 00000800
IO$_READLBLK                        = 00000021              MBA$M_SR_ERCONF                     = 00000008
IO$_READPBLK                        = 0000000C              MBA$M_SR_INVMAP                     = 00000010
IO$_READPRESET                      = 00000019              MBA$M_SR_ISTO                       = 00000002
IO$_READTRACKD                      = 00000010              MBA$M_SR_MAPPE                      = 00000020
IO$_READVBLK                        = 00000031              MBA$M_SR_MBEXC                      = 00000080
IO$_RECAL                           = 00000003              MBA$M_SR_MCPE                       = 00020000
IO$_RELEASE                         = 00000005              MBA$M_SR_MDPE                       = 00000040
IO$_RETCENTER                       = 00000007              MBA$M_SR_MXF                        = 00000100
IO$_SEARCH                          = 00000009              MBA$M_SR_NED                        = 00040000
IO$_SEEK                            = 00000002              MBA$M_SR_PGE                        = 00080000
IO$_SENSECHAR                       = 0000001B              MBA$M_SR_RDS                        = 00000004
IO$_SENSEMODE                       = 00000027              MBA$M_SR_RDTO                       = 00000001
IO$_SETCHAR                         = 0000001A              MBA$M_SR_SPE                        = 00004000
IO$_SETMODE                         = 00000023              MBA$M_SR_WCKLWR                     = 00000200
```

```
MBA$M_SR_WCKUPR                 = 00000400                    RM_ER1_V_HCRC                 = 00000008
MBA$V_SR_NED                    = 00000012                    RM_ER1_V_OPI                  = 0000000D
NOP                               00000254 R       03         RM_ER1_V_UNS                  = 0000000E
NORMAL                            000002FD R       03         RM_ER1_V_WLE                  = 0000000B
OFF                               000003B9 R       03         RM_ER2                          00000034
OFFSET                            00000254 R       03         RM_ER2_M_BSE                  = 00008000
OFFSIZ                          = 00000004                    RM_ER2_M_DPE                  = 00000008
OFFTAB                            000000AE R       03         RM_ER2_M_DVC                  = 00000080
PACKACK                           0000024E R       03         RM_ER2_M_IVC                  = 00001000
POSIT                             0000061C R       03         RM_ER2_M_LBC                  = 00000400
PR$_IPL                         = 00000012                    RM_ER2_M_LSC                  = 00000800
READDATA                          0000027C R       03         RM_ER2_M_OPE                  = 00002000
READHEAD                          0000027C R       03         RM_ER2_V_BSE                  = 0000000F
READPRESET                        00000254 R       03         RM_ER2_V_SKI                  = 0000000E
READTRACKD                        00000261 R       03         RM_ER2_V_SSE                  = 00000005
RECAL                             00000254 R       03         RM_LA                           0000001C
RELEASE                           00000254 R       03         RM_MR                           0000000C
RESETXFR                          000008A1 R       03         RM_MR2                          00000030
RETCENTER                         00000254 R       03         RM_MR_M_DM                    = 00008000
RETREG                            000006CB R       03         RM_OF                           00000024
RETRY                             0000032E R       03         RM_OF_M_CMO                   = 00008000
RETRYERR                          00000441 R       03         RM_OF_M_FMT                   = 00001000
RM_AS                             00000010                    RM_OF_M_HCI                   = 00000400
RM_CS1                            00000000                    RM_OF_M_MTD                   = 00004000
RM_CS1_M_GO                     = 00000001                    RM_OF_M_SSEI                  = 00000200
RM_DA                             00000014                    RM_OF_V_SSEI                  = 00000009
RM_DC                             00000028                    RM_SN                           00000020
RM_DS                             00000004                    RM_UNUSED                       0000002C
RM_DS_M_DPR                     = 00000100                    SAFUNC                          00000777 R       03
RM_DS_M_ERR                     = 00004000                    SEARCH                          00000254 R       03
RM_DS_V_DPR                     = 00000008                    SEARCHA                         000005F9 R       03
RM_DS_V_ERR                     = 0000000E                    SEEK                            00000254 R       03
RM_DS_V_MOL                     = 0000000C                    SEIZE                           000005BD R       03
RM_DS_V_VV                      = 00000006                    SIZ...                        = 00000001
RM_DT                             00000018                    SPECOND                         00000848 R       03
RM_DT_V_DRQ                     = 0000000B                    SS$_CTRLERR                   = 00000054
RM_ECT                            00000038                    SS$_DATACHECK                 = 0000005C
RM_EC2                            0000003C                    SS$_DRVERR                    = 0000008C
RM_ER1                            00000008                    SS$_FORMAT                    = 000000BC
RM_ER1_M_AOE                    = 00000200                    SS$_IVADDR                    = 00000134
RM_ER1_M_DCK                    = 00008000                    SS$_MEDOFL                    = 000001A4
RM_ER1_M_DTE                    = 00001000                    SS$_NONEXDRV                  = 000001C4
RM_ER1_M_ECH                    = 00000040                    SS$_NORMAL                    = 00000001
RM_ER1_M_FER                    = 00000010                    SS$_OPINCOMPL                 = 000002D4
RM_ER1_M_HCE                    = 00000080                    SS$_PARITY                    = 000001F4
RM_ER1_M_HCRC                   = 00000100                    SS$_TIMEOUT                   = 0000022C
RM_ER1_M_IAE                    = 00000400                    SS$_UNSAFE                    = 0000023C
RM_ER1_M_ILF                    = 00000001                    SS$_VOLINV                    = 00000254
RM_ER1_M_ILR                    = 00000002                    SS$_WASECC                    = 00000639
RM_ER1_M_OPI                    = 00002000                    SS$_WRITLCK                   = 0000025C
RM_ER1_M_PAR                    = 00000008                    TRANNOCH                        00000290 R       03
RM_ER1_M_RMR                    = 00000004                    TRANRQCH                        0000028A R       03
RM_ER1_M_UNS                    = 00004000                    TRANXT                          00000303 R       03
RM_ER1_M_WCF                    = 00000020                    UCB$B_CEX                     = 00000093
RM_ER1_M_WLE                    = 00000800                    UCB$B_DEVCLASS                = 00000040
RM_ER1_V_FER                    = 00000004                    UCB$B_DEVTYPE                 = 00000041
RM_ER1_V_HCE                    = 00000007                    UCB$B_DIPL                    = 0000005E
```

```
UCB$B_DR_ERL                      000000D5
UCB$B_DR_SSTS                     000000D4
UCB$B_ERTCNT                    = 00000080
UCB$B_ERTMAX                    = 00000081
UCB$B_FEX                       = 00000092
UCB$B_FIPL                      = 0000000B
UCB$B_OFFNDX                    = 000000CA
UCB$B_OFFRTC                    = 000000CB
UCB$B_SECTORS                   = 00000044
UCB$B_SLAVE                     = 00000090
UCB$K_DR_LENGTH                 = 000000DC
UCB$K_LCC_DISK_LENGTH           = 000000CC
UCB$L_CRB                       = 00000024
UCB$L_DEVCHAR                   = 00000038
UCB$L_DEVCHAR2                  = 0000003C
UCB$L_DEVDEPEND                 = 00000044
UCB$L_DPC                       = 0000009C
UCB$L_DR_BCR                      000000D8
UCB$L_DR_SR                       000000CC
UCB$L_IRP                       = 00000058
UCB$L_MAXBLOCK                  = 000000B0
UCB$L_MEDIA_ID                  = 0000008C
UCB$L_SVAPTE                    = 00000078
UCB$M_ONLINE                    = 00000010
UCB$M_POWER                     = 00000020
UCB$M_TIMOUT                    = 00000040
UCB$M_VALID                     = 00000800
UCB$V_BSY                       = 00000008
UCB$V_ECC                       = 00000000
UCB$V_ONLINE                    = 00000004
UCB$V_POWER                     = 00000005
UCB$V_VALID                     = 0000000B
UCB$W_BCNT                      = 0000007E
UCB$W_DA                        = 000000BC
UCB$W_DC                        = 000000BE
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_DR_ER2                      000000D0
UCB$W_DR_MR                       000000D2
UCB$W_DR_OFR                      000000D6
UCB$W_ECT                       = 000000C4
UCB$W_EC2                       = 000000C6
UCB$W_FUNC                      = 0000009A
UCB$W_OFFSET                    = 000000C8
UCB$W_STS                       = 00000064
UCB$W_UNIT                      = 00000054
UNLOAD                            00000245 R     03
VEC$L_IDB                       = 00000008
WRITECHECK                        00000271 R     03
WRITECHECKH                       00000271 R     03
WRITEDATA                         00000277 R     03
WRITEHEAD                         00000277 R     03
WRITETRACKD                       0000025C R     03
XFER                              0000060D R     03
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           000000DC  (   220.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$$$105_PROLOGUE                 00000070  (   112.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$$$115_DRIVER                   00000A3E  (  2622.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

                                    +------------------------+
                                    ! Performance indicators !
                                    +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      31    00:00:00.04    00:00:03.11
Command processing                 109    00:00:00.38    00:00:05.85
Pass 1                             605    00:00:19.66    00:01:54.25
Symbol table sort                    0    00:00:02.56    00:00:22.81
Pass 2                             331    00:00:04.67    00:00:31.08
Symbol table output                 47    00:00:00.23    00:00:00.56
Psect synopsis output                2    00:00:00.01    00:00:00.25
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals              1127    00:00:27.57    00:02:57.92

The working set limit was 2250 pages.
159994 bytes (313 pages) of virtual memory were used to buffer the intermediate code.
There were 130 pages of symbol table space allocated to hold 2372 non-local and 84 local symbols.
1856 source lines were read in Pass 1, producing 24 object records in Pass 2.
48 pages of virtual memory were used to define 45 macros.

                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         30
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      10
TOTALS (all libraries)                                 40

2486 GETS were required to define 40 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:DRDRIVER/OBJ=OBJ$:DRDRIVER MSRC$:DRDRIVER/UPDATE=(ENH$:DRDRIVER)+EXECML$/LIB
```

0109 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DQDRIVER
LIS

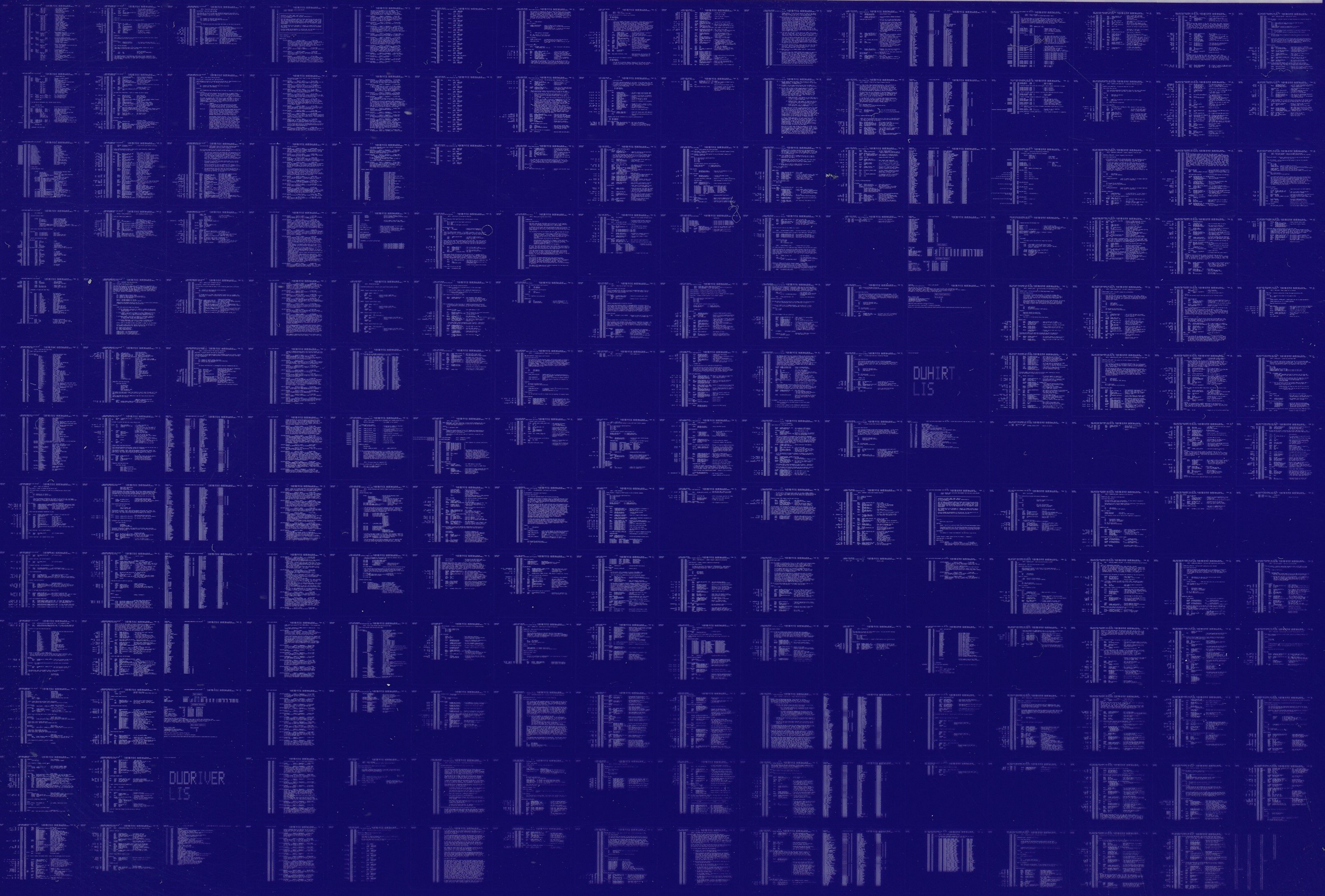
DLDRIVER
LIS

DRDRIVER
LIS

DMDRIVER
LIS

0110 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DUHIRT
LIS

DUDRIVER
LIS