```
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 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
```

FILEID**DBDRIVER

```
DDDDDDDD    BBBBBBBB    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DDDDDDDD    BBBBBBBB    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BBBBBBBB    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
DD      DD  BBBBBBBB    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRPRR
DD      DD  BB      BB  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD  BB      BB  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD  BB      BB  DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
DD      DD  BB      BB  DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
DDDDDDDD    BBBBBBBB    DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR      ....
DDDDDDDD    BBBBBBBB    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
```

DBD
V04

Table of contents

```
0000     1          .TITLE  DBDRIVER - RP04/05/06 DISK DRIVER
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ; D. N. CUTLER 30-JAN-77
0000    29  ;
0000    30  ; MODIFIED BY:
0000    31  ;
0000    32  ;       V03-012 RAS0300         Ron Schaefer            27-Apr-1984
0000    33  ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    34  ;               devices will have the "node$" prefix.
0000    35  ;
0000    36  ;       V03-011 PRD0074         Paul R. DeStefano       28-Feb-1984
0000    37  ;               Modified ERROR routine so that software volume valid
0000    38  ;               isn't set if a pack acknowledge function is executed
0000    39  ;               and medium online isn't set.
0000    40  ;
0000    41  ;       V03-010 PRD0031         Paul R. DeStefano       09-Sep-1983
0000    42  ;               Added EXE$LCLDSKVALID to function decision table.
0000    43  ;
0000    44  ;       V03-009 ROW0211         Ralph O. Weber          16-AUG-1983
0000    45  ;               Change device-dependent UCB definition base from UCB$W_BCR to
0000    46  ;               UCB$K_LCL_DISK_LENGTH.  Also change UCB$L_DB_BCR to overlay
0000    47  ;               UCB$L_BCR, a field newly created to meet the needs of this
0000    48  ;               driver.
0000    49  ;
0000    50  ;       V03-008 PRD0022         Paul R. DeStefano       05-May-1983
0000    51  ;               Modified ERROR routine to attempt to clear a drive
0000    52  ;               unsafe condidtion.
0000    53  ;
0000    54  ;       V03-007 PRD53302        Paul R. DeStefano       04-May-1983
0000    55  ;       ECO 02  Modified RETRYERR routine to issue a Drive Clear before
0000    56  ;               retrying a function.  Modified FUNCXT routine to issue
0000    57  ;               a Drive Clear function before releasing the drive.
```

```
0000     58 ;
0000     59 ;       V03-006 PRD0017         Paul R. DeStefano       26-Apr-1983
0000     60 ;               Modified FATALERR routine to return SS$_PARITY only for
0000     61 ;               errors that possibly indicate bad media.  All other error
0000     62 ;               conditions which formerly returned SS$_PARITY now return
0000     63 ;               SS$_CNTLERR.
0000     64 ;
0000     65 ;       V03-005 PRD0012         Paul R. DeStefano       14-Apr-1983
0000     66 ;               Modified ECC correction logic so that ECC is only applied
0000     67 ;               when there is single bit ECC correctable error, or if there
0000     68 ;               is a multiple bit ECC correctable error and the error cannot
0000     69 ;               be corrected using retries.
0000     70 ;
0000     71 ;       V03-04  ROW47161        Ralph O. Weber          17-SEP-1982
0000     72 ;       ECO 01  Enhance ECC recovery logic to prevent bytes transfered counts
0000     73 ;               which are not exact multiples of 512 from causing transfer
0000     74 ;               parameters from being incorrectly updated.  Because a non-512-
0000     75 ;               intergal bytes transfered counts indicates an incomplete
0000     76 ;               transfer of the last block, this change also prevents ECC
0000     77 ;               corrections when such bytes transfered counts are encountered.
0000     78 ;
0000     79 ;       V03-003 KDM0002         Kathleen D. Morse       28-Jun-1982
0000     80 ;               Added $DCDEF, $DYNDEF, $PRDEF, and $SSDEF.
0000     81 ;
0000     82 ;       V03-002 KTA0100         Kerbey T. Altmann               07-Jun-1982
0000     83 ;               Add code to set UCB$L_MEDIA_ID field.
0000     84 ;
0000     85 ;
0000     86 ;**
0000     87 ;
0000     88 ; RP04/04/06 DISK DRIVER
0000     89 ;
0000     90 ; MACRO LIBRARY CALLS
0000     91 ;
0000     92
0000     93         $CRBDEF                         ;DEFINE CRB OFFSETS
0000     94         $DCDEF                          ;DEFINE DEVICE CLASSES
0000     95         $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS BITS
0000     96         $DDBDEF                         ;DEFINE DDB OFFSETS
0000     97         $DPTDEF                         ;DEFINE DPT OFFSETS
0000     98         $DYNDEF                         ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000     99         $EMBDEF                         ;DEFINE EMB OFFSETS
0000    100         $IDBDEF                         ;DEFINE IDB OFFSETS
0000    101         $IODEF                          ;DEFINE I/O FUNCTION CODES
0000    102         $IRPDEF                         ;DEFINE IRP OFFSETS
0000    103         $MBADEF                         ;DEFINE MBA REGISTER OFFSETS
0000    104         $PRDEF                          ;DEFINE PROCESSOR REGISTERS
0000    105         $SSDEF                          ;DEFINE SYSTEM STATUS CODES
0000    106         $UCBDEF                         ;DEFINE UCB OFFSETS
0000    107         $VECDEF                         ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000    108
0000    109 ;
0000    110 ; LOCAL MACROS
0000    111 ;
0000    112 ; EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000    113 ;
0000    114
```

```
0000    115          .MACRO  EXFUNC BDST,FCODE
0000    116                  .IF NB  FCODE
0000    117                  MOVZBL  #CD'FCODE,R0
0000    118                  .ENDC
0000    119                  BSBW    FEX
0000    120                  .SIGNED_WORD BDST-.-2
0000    121          .ENDM
0000    122
0000    123 ;
0000    124 ; GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000    125 ;
0000    126
0000    127          .MACRO  GENF FCODE
0000    128                  CD'FCODE=.-FTAB
0000    129                  .BYTE   FCODE!RP_CS1_M_GO
0000    130          .ENDM
0000    131
0000    132 ;
0000    133 ; LOCAL SYMBOLS
0000    134 ;
0000    135 ; RP04/05/06 MASSBUS REGISTER OFFSETS
0000    136 ;
0000    137
0000    138          $DEFINI RP
0000    139
0000    140 $DEF     RP_CS1          .BLKL   1       ;DRIVE CONTROL REGISTER
0004    141          _VIELD  RP_CS1,0,<-             ; DRIVE CONTROL REGISTER BIT DEFINITIONS
0004    142                  <GO,,M>,-               ; GO BIT
0004    143                  <FCODE,5>-              ; FUNCTION CODE
0004    144                  >                       ;
0004    145 $DEF     RP_DS           .BLKL   1       ;DRIVE STATUS REGISTER
0008    146          _VIELD  RP_DS,6,<-              ; DRIVE STATUS REGISTER BIT DEFINITIONS
0008    147                  <VV,,M>,-               ; VOLUME VALID
0008    148                  <DRY,,M>,-              ; DRIVE READY
0008    149                  <DPR,,M>,-              ; DRIVE PRESENT
0008    150                  <PGM,,M>,-              ; PROGRAMMABLE
0008    151                  <LST,,M>,-              ; LAST SECTOR TRANSFERED
0008    152                  <WRL,,M>,-              ; DRIVE WRITE LOCKED
0008    153                  <MOL,,M>,-              ; MEDIUM ONLINE
0008    154                  <PIP,,M>,-              ; POSITIONING IN PROGRESS
0008    155                  <ERR,,M>,-              ; COMPOSITE ERROR
0008    156                  <ATA,,M>-               ; ATTENTION ACTIVE
0008    157                  >                       ;
0008    158 $DEF     RP_ER1          .BLKL   1       ;ERROR REGISTER 1
000C    159          _VIELD  RP_ER1,0,<-             ; ERROR REGISTER 1 BIT DEFINITIONS
000C    160                  <ICF,,M>,-              ; ILLEGAL FUNCTION
000C    161                  <ILR,,M>,-              ; ILLEGAL REGISTER
000C    162                  <RMR,,M>,-              ; REGISTER MODIFY REFUSED
000C    163                  <PAR,,M>,-              ; PARITY ERROR
000C    164                  <FER,,M>,-              ; FORMAT ERROR
000C    165                  <WCF,,M>,-              ; WRITE CLOCK FAIL
000C    166                  <ECH,,M>,-              ; ECC HARD ERROR
000C    167                  <HCE,,M>,-              ; HEADER COMPARE ERROR
000C    168                  <HCRC,,M>,-             ; HEADER CRC ERROR
000C    169                  <AOE,,M>,-              ; ADDRESS OVERFLOW ERROR
000C    170                  <IAE,,M>,-              ; ILLEGAL ADDRESS ERROR
000C    171                  <WLE,,M>,-              ; WRITE LOCK ERROR
```

```
                        000C   172                     <DTE,,M>,-              ; DRIVE TIMING ERROR
                        000C   173                     <OPI,,M>,-              ; OPERATION INCOMPLETE
                        000C   174                     <UNS,,M>,-              ; DRIVE UNSAFE
                        000C   175                     <DCK,,M>-               ; DATA CHECK ERROR
                        000C   176             >                               ;
                        000C   177 $DEF     RP_MR               .BLKL   1       ;MAINTENANCE REGISTER
                        0010   178 $DEF     RP_AS               .BLKL   1       ;ATTENTION SUMMARY REGISTER
                        0014   179 $DEF     RP_DA               .BLKL   1       ;DESIRED SECTOR/TRACK ADDRESS REGISTER
                        0018   180             _VIELD  RP_DA,0,<-              ; DESIRED ADDRESS FIELD DEFINITIONS
                        0018   181                     <SA,5>,-                ; DESIRED SECTOR ADDRESS
                        0018   182                     <,3>,-                  ; RESERVED BITS
                        0018   183                     <TA,5>-                 ; DESIRED TRACK ADDRESS
                        0018   184             >                               ;
                        0018   185 $DEF     RP_DT               .BLKL   1       ;DRIVE TYPE REGISTER
                        001C   186             _VIELD  RP_DT,0,<-              ; DRIVE TYPE REGISTER FIELD DEFINITIONS
                        001C   187                     <DTN,9>,-               ; DRIVE TYPE NUMBER
                        001C   188                     <,2>,-                  ; RESERVED BITS
                        001C   189                     <DRQ,,M>-               ; DRIVE REQUEST REQUIRED
                        001C   190             >                               ;
                        001C   191 $DEF     RP_LA               .BLKL   1       ;LOOKAHEAD REGISTER
                        0020   192 $DEF     RP_ER2              .BLKL   1       ;ERROR REGISTER 2
                        0024   193 $DEF     RP_OF               .BLKL   1       ;OFFSET REGISTER
                        0028   194             _VIELD  RP_OF,0,<-              ; OFFSET REGISTER BIT DEFINITIONS
                        0028   195                     <OFF,8>,-               ; OFFSET VALUE
                        0028   196                     <DCK,,M>,-              ; DATA CHECK IN PROGRESS (SOFTWARE)
                        0028   197                     <,1>,-                  ; RESERVED BIT
                        0028   198                     <HCI,,M>,-              ; HEADER COMPARE INHIBIT
                        0028   199                     <ECI,,M>,-              ; ECC INHIBIT
                        0028   200                     <FMT,,M>-               ; 16-BIT FORMAT
                        0028   201             >                               ;
                        0028   202 $DEF     RP_DC               .BLKL   1       ;DESIRED CYLINDER ADDRESS
                        002C   203 $DEF     RP_CC               .BLKL   1       ;CURRENT CYLINDER ADDRESS
                        0030   204 $DEF     RP_SN               .BLKL   1       ;DRIVE SERIAL NUMBER
                        0034   205 $DEF     RP_ER3              .BLKL   1       ;ERROR REGISTER 3
                        0038   206             _VIELD  RP_ER3,14,<-            ; ERROR REGISTER 3 BIT DEFINITIONS
                        0038   207                     <SKI,,M>-               ; SEEK INCOMPLETE
                        0038   208             >                               ;
                        0038   209 $DEF     RP_EC1              .BLKL   1       ;ECC POSITION REGISTER
                        003C   210              VIELD  RP_EC1,0,<<POS,13>>     ; ECC POSITION FIELD
                        003C   211 $DEF     RP_EC2              .BLKL   1       ;ECC PATTERN REGISTER
                        0040   212             _VIELD  RP_EC2,0,<<PAT,11>>     ; ECC PATTERN FIELD
                        0040   213
                        0040   214             $DEFEND RP
                        0000   215
                        0000   216 ;
                        0000   217 ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
                        0000   218 ;
                        0000   219
                        0000   220             $DEFINI UCB
                        0000   221
                000000CC 0000  222 .=UCB$K_LCL_DISK_LENGTH                      ; Establish device-dependent UCB base
                        00CC   223
                000000C0 00CC  224 UCB$L_DB_BCR = UCB$L_BCR                     ; Local BCR longword overlays the
                        00CC   225                                              ; space reserved in the UCB.
                        00CC   226                                              ; N.B. most drivers only need a word.
                        00CC   227
                        00CC   228 $DEF     UCB$W_DB_ER3    .BLKW   1       ; Space to save RP_ER3 after operation.
```

```
                     00CE   229
                     00CE   230 $DEF    UCB$L_DB_SR     .BLKL   1       ;SAVE MBA STATUS REGISTER
                     00D2   231
                     00D2   232 $DEF    UCB$B_DB_ERL    .BLKB   1       ; Space for flag used to signal Medium
                     00D3   233                                         ;  offline at start of function.
                     00D3   234         _VIELD  ERL,0,<-
                     00D3   235                 <MEDOFF,,M>, -          ; MEDIUM OFFLINE FLAG
                     00D3   236                 <DUALPORT,,M>, -        ; DUALPORT KIT FLAG
                     00D3   237                 <ECC_DEFER,,M>, -       ; Flag to indicate that ECC correction
                     00D3   238         >                               ;  has been deferred until offset
                     00D3   239                                         ;  retries are exhausted.
                     00D3   240
000000D6             00D3   241                                 .BLKB   3       ; Reserved.
000000D6             00D6   242 UCB$K_DB_LENGTH=.                        ; Length of UCB for DB devices.
                     00D6   243         $DEFEND
                     0000   244
                     0000   245 ;
                     0000   246 ; HARDWARE FUNCTION CODES
                     0000   247 ;
                     0000   248
00000000             0000   249 F_NOP=0*2                               ;NO OPERATION
00000002             0000   250 F_UNLOAD=1*2                            ;UNLOAD DRIVE
00000004             0000   251 F_SEEK=2*2                              ;SEEK CYLINDER
00000006             0000   252 F_RECAL=3*2                             ;RECALIBRATE
00000008             0000   253 F_DRVCLR=4*2                            ;DRIVE CLEAR
0000000A             0000   254 F_RELEASE=5*2                           ;RELEASE DRIVE
0000000C             0000   255 F_OFFSET=6*2                            ;OFFSET HEADS
0000000E             0000   256 F_RETCENTER=7*2                         ;RETURN TO CENTERLINE
00000010             0000   257 F_READPRESET=8*2                        ;READ IN PRESET
00000012             0000   258 F_PACKACK=9*2                           ;PACK ACKNOWLEDGE
00000018             0000   259 F_SEARCH=12*2                           ;SEARCH FOR SECTOR
00000018             0000   260 F_SEARCHA=12*2                          ;SEARCH AHEAD FOR SECTOR
00000028             0000   261 F_WRITECHECK=20*2                       ;WRITE CHECK DATA
0000002A             0000   262 F_WRITECHECKH=21*2                      ;WRITE CHECK HEADER AND DATA
00000030             0000   263 F_WRITEDATA=24*2                        ;WRITE DATA
00000032             0000   264 F_WRITEHEAD=25*2                        ;WRITE HEADER AND DATA
00000038             0000   265 F_READDATA=28*2                         ;READ DATA
0000003A             0000   266 F_READHEAD=29*2                         ;READ HEADER AND DATA
                     0000   267
                     0000   268 ;
                     0000   269 ; LOCAL DATA
                     0000   270 ;
                     0000   271 ; DRIVER PROLOGUE TABLE
                     0000   272 ;
                     0000   273
                     0000   274         DPTAB   -                       ;DEFINE DRIVER PROLOGUE TABLE
                     0000   275                 END=DB_END,-            ;END OF DRIVER
                     0000   276                 ADAPTER=MBA,-           ;ADAPTER TYPE
                     0000   277                 FLAGS=DPT$M_SVP,-       ;SYSTEM PAGE TABLE ENTRY REQUIRED
                     0000   278                 UCBSIZE=UCB$K_DB_LENGTH,-;UCB SIZE
                     0000   279                 NAME=DBDRIVER           ;DRIVER NAME
                     0038   280         DPT_STORE INIT                  ;CONTROL BLOCK INIT VALUES
                     0038   281         DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\> ;DEFAULT ACP NAME
                     003F   282         DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_PACK ;ACP CLASS
                     0043   283         DPT_STORE UCB,UCB$B_FIPL,B,8    ;FORK IPL
                     0047   284         DPT_STORE UCB,UCB$L_DEVCHAR,L,- ;DEVICE CHARACTERISTICS
                     0047   285                 <DEV$M_FOD-             ; FILES ORIENTED
```

```
          0047          !DEV$M_DIR-                     ; DIRECTORY STRUCTURED
          0047          !DEV$M_AVL-                     ; AVAILABLE
          0047          !DEV$M_ELG-                     ; ERROR LOGGING ENABLED
          0047          !DEV$M_SHR-                     ; SHAREABLE
          0047          !DEV$M_IDV-                     ; INPUT DEVICE
          0047          !DEV$M_ODV-                     ; OUTPUT DEVICE
          0047          !DEV$M_RND>                     ; RANDOM ACCESS
          004E     DPT_STORE UCB,UCB$L_DEVCHAR2,L,-     ; DEVICE CHARACTERISTICS
          004E          <DEV$M_NNM>                     ; PREFIX NAME WITH "node$"
          0055     DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_DISK ;DEVICE CLASS
          0059     DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
          005E     DPT_STORE UCB,UCB$B_DIPL,B,21       ;DEVICE IPL
          0062     DPT_STORE UCB,UCB$B_ERTCNT,B,8      ;ERROR RETRY COUNT
          0066     DPT_STORE UCB,UCB$B_ERTMAX,B,8      ;MAX ERROR RETRY COUNT
          006A     DPT_STORE REINIT                    ;CONTROL BLOCK RE-INIT VALUES
          006A     DPT_STORE DDB,DDB$L_DDT,D,DB$DDT    ;DDT ADDRESS
          006F     DPT_STORE END                       ;
          0000
          0000 ;
          0000 ; DRIVER DISPATCH TABLE
          0000 ;
          0000
          0000     DDTAB   DB,-                        ;DRIVER DISPATCH TABLE
          0000             DB_STARTIO,-                ;START I/O OPERATION
          0000             DB_UNSOLNT,-                ;UNSOLICITED INTERRUPT
          0000             DB_FUNCTABLE,-              ;FUNCTION DECISION TABLE
          0000             0,-                         ;CANCEL I/O ENTRY POINT
          0000             DB_REGDUMP,-                ;REGISTER DUMP ROUTINE
          0000             <<RP_EC2+4+MBA$L_BCR+4+8>+<<3+5+1>*4>>,- ;DIAGNOSTIC BUFFER SIZE
          0000             <<RP_EC2+4+MBA$L_BCR+4+8>+<1*4>+<EMB$L_DV_REGSAV>>,- ;ERROR BUFFER S
          0000             DB_RPOX_INIT                ;UNIT INITIALIZATION
          0038
          0038 ;
          0038 ; DATA CHECK FUNCTION TRANSLATION TABLE
          0038 ;
          0038
          0038 CHECKTAB:                               ;
      0A' 0038     .BYTE   CDF_WRITECHECK              ;WRITE DATA
      0A' 0039     .BYTE   CDF_WRITECHECK              ;READ DATA
      0F' 003A     .BYTE   CDF_WRITECHECKH             ;WRITE HEADER AND DATA
      0F' 003B     .BYTE   CDF_WRITECHECKH             ;READ HEADER AND DATA
          003C
          003C ;
          003C ; RPOX DRIVE TYPE DESCRIPTOR TABLE
          003C ;
          003C
          003C DB_DTDESC:                              ;
     0010 003C     .WORD   ^X10                        ;RP04
       03 003E     .BYTE   DT$_RP04                    ;
       16 003F     .BYTE   22                          ;22 SECTORS
       13 0040     .BYTE   19                          ;19 TRACKS
     019B 0041     .WORD   411                         ;411 CYLINDERS PER PACK
 00029F16 0043     .LONG   411*19*22                   ;MAXIMUM BLOCKS PER PACK
 20A50004 0047     .LONG   ^X20A50004                  ;MEDIA ID "DB RP04"
 0000000F 004B DB_DTDESCLEN=.-DB_DTDESC                ;LENGTH OF DRIVE TYPE DESCRIPTOR
     0011 004B     .WORD   ^X11                        ;RP05
       04 004D     .BYTE   DT$_RP05                    ;
```

```
                16  004E   343          .BYTE   22                      ;22 SECTORS
                13  004F   344          .BYTE   19                      ;19 TRACKS
              019B  0050   345          .WORD   411                     ;411 CYLINDERS PER PACK
          00029F16  0052   346          .LONG   411*19*22               ;MAXIMUM BLOCKS PER PACK
          20A50004  0056   347          .LONG   ^X20A50004              ;MEDIA ID "DB RP04"
              0012  005A   348          .WORD   ^X12                    ;RP06
                05  005C   349          .BYTE   DT$_RP06                ;
                16  005D   350          .BYTE   22                      ;22 SECTORS
                13  005E   351          .BYTE   19                      ;19 TRACKS
              032F  005F   352          .WORD   815                     ;815 CYLINDERS PER PACK
          000532BE  0061   353          .LONG   815*19*22               ;MAXIMUM BLOCKS PER PACK
          20A50006  0065   354          .LONG   ^X20A50006              ;MEDIA ID "DB RP06"
                    0069   355
                0000  0069   356          .WORD   0                       ;END OF TABLE
          0000007A  006B   357          .BLKB   DB_DTDESCLEN            ;SPARE DRIVE TYPE SLOT
          00000089  007A   358          .BLKB   DB_DTDESCLEN            ;SPARE DRIVE TYPE SLOT
                    0089   359
                    0089   360 ;
                    0089   361 ; HARDWARE I/O FUNCTION CODE TABLE
                    0089   362 ;
                    0089   363
                    0089   364 FTAB:                                    ;
                    0089   365          GENF    F_NOP                   ;NO OPERATION
                    008A   366          GENF    F_UNLOAD                ;UNLOAD VOLUME
                    008B   367          GENF    F_SEEK                  ;SEEK CYLINDER
                    008C   368          GENF    F_RECAL                 ;RECALIBRATE
                    008D   369          GENF    F_DRVCLR                ;DRIVE CLEAR
                    008E   370          GENF    F_NOP                   ;RELEASE PORT (NOP)
                    008F   371          GENF    F_OFFSET                ;OFFSET HEADS
                    0090   372          GENF    F_RETCENTER             ;RETURN HEADS TO CENTERLINE
                    0091   373          GENF    F_PACKACK               ;PACK ACKNOWLEDGE
                    0092   374          GENF    F_SEARCH                ;SEARCH FOR SECTOR
                    0093   375          GENF    F_WRITECHECK            ;WRITE CHECK
                    0094   376          GENF    F_WRITEDATA             ;WRITE DATA
                    0095   377          GENF    F_READDATA              ;READ DATA
                    0096   378          GENF    F_WRITEHEAD             ;WRITE HEADER AND DATA
                    0097   379          GENF    F_READHEAD              ;READ HEADER AND DATA
                    0098   380          GENF    F_WRITECHECKH           ;WRITE CHECK HEADER AND DATA
                    0099   381          GENF    F_READPRESET            ;READ IN PRESET
                    009A   382          GENF    F_SEARCHA               ;SEARCH AHEAD FOR SECTOR
                    009B   383
                    009B   384 ;
                    009B   385 ; OFFSET TABLE FOR RP06 - RP04 VALUES = RP06 VALUES * 2 & ^XFF
                    009B   386 ;
                    009B   387
                    009B   388 OFFTAB:                                  ;
                00  009B   389          .BYTE   0                       ;RETURN TO CENTERLINE
                08  009C   390          .BYTE   ^X8                     ;+200 (+400)
                C8  009D   391          .BYTE   ^XC8                    ;-200 (-400)
                10  009E   392          .BYTE   ^X10                    ;+400 (+800)
                D0  009F   393          .BYTE   ^XD0                    ;-400 (-800)
                18  00A0   394          .BYTE   ^X18                    ;+600 (+1200)
                D8  00A1   395          .BYTE   ^XD8                    ;-600 (-1200)
                00  00A2   396          .BYTE   0                       ;RETURN TO CENTERLINE
          00000008  00A3   397 OFFSIZ=.-OFFTAB                          ;SIZE OF OFFSET TABLE
```

```
                        .SBTTL  RP04/05/06 FUNCTION DECISION TABLE
                ;+
                ; RP04/05/06 FUNCTION DECISION TABLE
                ;-

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

```
00AB   456                     ACCESS,-                ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00AB   457                     ACPCONTROL,-            ;ACP CONTROL FUNCTION
00AB   458                     CREATE,-                ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00AB   459                     DEACCESS,-              ;DEACCESS FILE
00AB   460                     DELETE,-                ;DELETE FILE AND/OR DIRECTORY ENTRY
00AB   461                     MODIFY,-                ;MODIFY FILE ATTRIBUTES
00AB   462                     MOUNT>                  ;MOUNT VOLUME
00B3   463             FUNCTAB +ACP$READBLK,-          ;READ FUNCTIONS
00B3   464                     <READHEAD,-             ;READ HEADER
00B3   465                      READLBLK,-             ;READ LOGICAL BLOCK
00B3   466                      READPBLK,-             ;READ PHYSICAL BLOCK
00B3   467                      READVBLK>              ;READ VIRTUAL BLOCK
00BF   468             FUNCTAB +ACP$WRITEBLK,-         ;WRITE FUNCTIONS
00BF   469                     <WRITECHECK,-           ;WRITE CHECK
00BF   470                      WRITECHECKH,-          ;WRITE CHECK HEADER AND DATA
00BF   471                      WRITEHEAD,-            ;WRITE HEADER
00BF   472                      WRITELBLK,-            ;WRITE LOGICAL BLOCK
00BF   473                      WRITEPBLK,-            ;WRITE PHYSICAL BLOCK
00BF   474                      WRITEVBLK>             ;WRITE VIRTUAL BLOCK
00CB   475             FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00D7   476             FUNCTAB +ACP$DEACCESS,<DEACCESS> ;DEACCESS FILE
00E3   477             FUNCTAB +ACP$MODIFY,-           ;
00E3   478                     <ACPCONTROL,-           ;ACP CONTROL FUNCTION
00E3   479                      DELETE,-               ;DELETE FILE OR DIRECTORY ENTRY
00E3   480                      MODIFY>                ;MODIFY FILE ATTRIBUTES
00EF   481             FUNCTAB +ACP$MOUNT,<MOUNT>      ;MOUNT VOLUME
00FB   482             FUNCTAB +EXE$LCLDSKVALID,-      ;LOCAL DISK VALID FUNCTIONS
00FB   483                     <UNLOAD,-               ;UNLOAD VOLUME
00FB   484                      AVAILABLE,-            ;UNIT AVAILABLE
00FB   485                      PACKACK>               ;PACK ACKNOWLEDGE
0107   486             FUNCTAB +EXE$ZEROPARM,-         ;ZERO PARAMETER FUNCTIONS
0107   487                     <NOP,-                  ;NO OPERATION
0107   488                      UNLOAD,-               ;UNLOAD VOLUME
0107   489                      RECAL,-                ;RECALIBRATE
0107   490                      DRVCLR,-               ;DRIVE CLEAR
0107   491                      RELEASE,-              ;RELEASE PORT
0107   492                      RETCENTER,-            ;RETURN HEADS TO CENTERLINE
0107   493                      READPRESET,-           ;READ IN PRESET
0107   494                      AVAILABLE,-            ;UNIT AVAILABLE
0107   495                      PACKACK>               ;PACK ACKNOWLEDGE
0113   496             FUNCTAB +EXE$ONEPARM,-          ;ONE PARAMETER FUNCTIONS
0113   497                     <SEEK,-                 ;SEEK CYLINDER
0113   498                      OFFSET,-               ;OFFSET HEADS
0113   499                      SEARCH>                ;SEARCH FOR SECTOR
011F   500             FUNCTAB +EXE$SENSEMODE,-        ;
011F   501                     <SENSECHAR,-            ;SENSE CHARACTERISTICS
011F   502                      SENSEMODE>             ;SENSE MODE
012B   503             FUNCTAB +EXE$SETCHAR,-          ;
012B   504                     <SETCHAR,-              ;SET CHARACTERISITCS
012B   505                      SETMODE>               ;SET MODE
```

```
                                0137   507          .SBTTL  START I/O OPERATION
                                0137   508 ;+
                                0137   509 ; DB_STARTIO - START I/O OPERATION ON DEVICE UNIT
                                0137   510 ;
                                0137   511 ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
                                0137   512 ;
                                0137   513 ; INPUTS:
                                0137   514 ;
                                0137   515 ;       R3 = ADDRESS OF I/O PACKET.
                                0137   516 ;       R5 = UCB ADDRESS OF DEVICE UNIT.
                                0137   517 ;
                                0137   518 ; OUTPUTS:
                                0137   519 ;
                                0137   520 ;       FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
                                0137   521 ;       RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
                                0137   522 ;       THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
                                0137   523 ;-
                                0137   524
                                0137   525 DB_STARTIO:                                    ;START I/O OPERATION
0080 C5   0081 C5   90          0137   526          MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
                      8A        013E   527          BICB    #<ERL_M_MEDOFF!-                  ; Clear flags used to signal medium
                                013F   528                  ERL_M_ECC_DEFER>,-                ;  offline and ECC correction deferred
     00D2 C5   05                013F   529                  UCB$B_DB_ERL(R5)                  ;  at start of function.
   009A C5   20 A3   B0          0143   530          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
         50   38 A3   D0          0149   531          MOVL    IRP$L_MEDIA(R3),R0            ;GET PARAMETER LONGWORD
                                014D   532
                                014D   533 ;
                                014D   534 ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                                014D   535 ;
                                014D   536
      06   00   EF              014D   537 10$:    EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODE
   51    20 A3                  0150   538                  IRP$W_FUNC(R3),R1           ;
      51   02   91              0153   539          CMPB    #IO$_SEEK,R1                ;SEEK FUNCTION?
           2F   13              0156   540          BEQL    20$                         ;IF EQL YES
      51   06   91              0158   541          CMPB    #IO$_OFFSET,R1              ;OFFSET FUNCTION?
           31   13              015B   542          BEQL    30$                         ;IF EQL YES
      51   09   91              015D   543          CMPB    #IO$_SEARCH,R1              ;SEARCH FUNCTION?
           33   13              0160   544          BEQL    40$                         ;IF EQL YES
      51   11   91              0162   545          CMPB    #IO$_AVAILABLE, R1          ;AVAILABLE function?
           0F   13              0165   546          BEQL    15$                         ;Branch if yes.
 00BC C5   50   D0              0167   547          MOVL    R0,UCB$W_DA(R5)             ;STORE PARAMETER LONGWORD
      51   18   91              016C   548          CMPB    #IO$_WRITECHECKH,R1         ;DISJOINT FUNCTION CODE?
           29   1A              016F   549          BGTRU   50$                         ;IF GTRU NO
      51   09   A2              0171   550          SUBW    #IO$_WRITECHECKH-IO$_READHEAD-1,R1 ;CONVERT TO DENSE FUNCTION CODE
           24   11              0174   551          BRB     50$                         ;
                                0176   552
                                0176   553 ;
                                0176   554 ; AVAILABLE FUNCTION - Clear software volume valid bit & exit
                                0176   555 ;
64 A5   0800 8F   AA            0176   556 15$:    BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear software volume valid bit.
      50   01   3C              017C   557          MOVZWL  #SS$_NORMAL, R0             ;Setup success status for zero
           51   D4              017F   558          CLRL    R1                          ;bytes transfered operation,
                                0181   559          REQCOM                              ;and complete request.
                                0187   560
                                0187   561 ;
                                0187   562 ; SEEK FUNCTION - SET CYLINDER ADDRESS
                                0187   563 ;
```

```
                     0187   564
00BE C5   50   B0     0187   565 20$:    MOVW    R0,UCB$W_DC(R5)         ;SET CYLINDER ADDRESS
          0C   11     018C   566          BRB     50$                     ;
               018E   567
               018E   568 ;
               018E   569 ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
               018E   570 ;
               018E   571
00C8 C5   50   90     018E   572 30$:    MOVB    R0,UCB$W_OFFSET(R5)     ;SET OFFSET VALUE
          05   11     0193   573          BRB     50$                     ;
               0195   574
               0195   575 ;
               0195   576 ; SEARCH FUNCTION - SET SECTOR ADDRESS
               0195   577 ;
               0195   578
00BC C5   50   90     0195   579 40$:    MOVB    R0,UCB$W_DA(R5)         ;SET SECTOR ADDRESS
               019A   580
               019A   581 ;
               019A   582 ; FINISH PREPROCESSING
               019A   583 ;
               019A   584
0092 C5   51   90     019A   585 50$:    MOVB    R1,UCB$B_FEX(R5)        ;SAVE FUNCTION DISPATCH INDEX
  54   24 A5   D0     019F   586          MOVL    UCB$L_CRB(R5),R4        ;GET ADDRESS OF CRB
  54   2C B4   D0     01A3   587          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4 ;GET FIRST CONTROLLER CSR ADDRESS
00 68 A5   00   E4     01A7   588          BBSC    #UCB$V_ECC,UCB$W_DEVSTS(R5),FDISPATCH ;CLEAR ECC CORRECTION MADE
               01AC   589
               01AC   590 ;
               01AC   591 ; CENTRAL FUNCTION DISPATCH
               01AC   592 ;
               01AC   593
               01AC   594 FDISPATCH:                                      ;FUNCTION DISPATCH
  53   58 A5   D0     01AC   595          MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF I/O PACKET
0D 2A A3   08   E0     01B0   596          BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
08 64 A5   0B   E0     01B5   597          BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
 50    0254 8F   3C     01BA   598          MOVZWL  #SS$_VOLINV,R0          ;SET VOLUME INVALID STATUS
          0582   31     01BF   599          BRW     RESETXFR                ;
               01C2   600
               01C2   601 ;
               01C2   602 ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
               01C2   603 ;
               01C2   604
 50   0092 C5   9A     01C2   605 10$:    MOVZBL  UCB$B_FEX(R5),R0        ;GET DISPATCH FUNCTION CODE
00C9 C5   10   90     01C7   606          MOVB    #RP_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
00CB C5   01   90     01CC   607          MOVB    #1,UCB$B_OFFRTC(R5)     ;SET INITIAL OFFSET RETRY COUNT
      00CA C5   94     01D1   608          CLRB    UCB$B_OFFNDX(R5)        ;CLEAR INITIAL OFFSET TABLE INDEX
               01D5   609          CASE    R0,<-                           ;DISPATCH TO FUNCTION HANDLING ROUTINE
               01D5   610                  NOP,-                           ;NO OPERATION
               01D5   611                  UNLOAD,-                        ;UNLOAD VOLUME
               01D5   612                  SEEK,-                          ;SEEK CYLINDER
               01D5   613                  RECAL,-                         ;RECALIBRATE
               01D5   614                  DRVCLR,-                        ;DRIVE CLEAR
               01D5   615                  RELEASE,-                       ;RELEASE PORT
               01D5   616                  OFFSET,-                        ;OFFSET HEADS
               01D5   617                  RETCENTER,-                     ;RETURN HEADS TO CENTER
               01D5   618                  PACKACK,-                       ;PACK ACKNOWLEDGE
               01D5   619                  SEARCH,-                        ;SEARCH FOR SECTOR
               01D5   620                  WRITECHECK,-                    ;WRITE CHECK DATA
```

```
                     01D5   621                WRITEDATA,-                     ;WRITE DATA
                     01D5   622                READDATA,-                      ;READ DATA
                     01D5   623                WRITEHEAD,-                     ;WRITE HEADER AND DATA
                     01D5   624                READHEAD,-                      ;READ HEADER AND DATA
                     01D5   625                WRITECHECKH,-                   ;WRITE CHECK HEADER AND DATA
                     01D5   626                READPRESET-                     ;READ IN PRESET
                     01D5   627                >
                     01FB   628
                     01FB   629 ;
                     01FB   630 ; IO$_UNLOAD INDICATES THE UNIT IS NOT MOUNTED SO WE CLEAR SOFTWARE VOLUME
                     01FB   631 ; VALID BEFORE EXECUTING THE OPERATION.  IO$_PACKACK INDICATES THAT SOFTWARE
                     01FB   632 ; IS READY TO MOUNT THE VOLUME SO WE SET SOFTWARE VOLUME VALID BEFORE
                     01FB   633 ; EXECUTING THE OPERATION.
                     01FB   634 ;
                     01FB   635 UNLOAD:
64 A5   0800 8F   AA 01FB   636         BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear software volume valid bit.
           06   11   0201   637         BRB     NOP                         ;Proceed with the unload operation.
                     0203   638
                     0203   639 PACKACK:
64 A5   0800 8F   A8 0203   640         BISW    #UCB$M_VALID, UCB$W_STS(R5) ;Set software volume valid bit.
                     0209   641 ;       BRB     NOP                         ;Proceed with the unload operation.
                     0209   642
                     0209   643 ;
                     0209   644 ; NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
                     0209   645 ; RETURN TO CENTER LINE, SEARCH, AND READ IN PRESET
                     0209   646 ;
                     0209   647
                     0209   648 NOP:                                          ;NO OPERATION
                     0209   649 SEEK:                                         ;SEEK CYLINDER
                     0209   650 RECAL:                                        ;RECALIBRATE
                     0209   651 DRVCLR:                                       ;DRIVE CLEAR
                     0209   652 RELEASE:                                      ;RELEASE PORT
                     0209   653 OFFSET:                                       ;OFFSET READ HEADS
                     0209   654 RETCENTER:                                    ;RETURN TO CENTERLINE
                     0209   655 SEARCH:                                       ;SEARCH FOR SECTOR
                     0209   656 READPRESET:                                   ;READIN PRESET
                     0209   657         EXFUNC  RETRY                       ;EXECUTE HOUSEKEEPING FUNCTION
           73   11   020E   658         BRB     NORMAL                      ;
                     0210   659
                     0210   660 ;
                     0210   661 ; WRITE CHECK DATA AND WRITE CHECK HEADER AND DATA
                     0210   662 ;
                     0210   663
                     0210   664 WRITECHECK:                                  ;WRITE CHECK DATA
                     0210   665 WRITECHECKH:                                 ;WRITE CHECK HEADER AND DATA
009A C5   4000 8F   AA 0210   666         BICW    #IO$M_DATACHECK,UCB$W_FUNC(R5) ;CLEAR DATA CHECK REQUEST
                     0217   667
                     0217   668 ;
                     0217   669 ; WRITE DATA, WRITE HEADER AND DATA, WRITE CHECK DATA, AND WRITE CHECK HEADER
                     0217   670 ; AND DATA
                     0217   671 ;
                     0217   672
                     0217   673 WRITEDATA:                                   ;WRITE DATA
                     0217   674 WRITEHEAD:                                   ;WRITE HEADER AND DATA
00C9 C5   08   88    0217   675         BISB    #RP_OF_M_ECI/256,UCB$W_OFFSET+1(R5) ;INHIBIT ECC CORRECTION
                     021C   676
                     021C   677 ;
```

```
                     021C   678 ; READ DATA, READ HEADER AND DATA, WRITE DATA, WRITE HEADER AND DATA, WRITE
                     021C   679 ; CHECK DATA, AND WRITE CHECK HEADER AND DATA
                     021C   680 ;
                     021C   681
                     021C   682 READDATA:                                    ;READ DATA
                     021C   683 READHEAD:                                    ;READ HEADER AND DATA
        08 009A C5   0C  E0  021C   684         BBS     #IO$V_INHSEEK,UCB$W_FUNC(R5),TRANRQCH ;IF SET, NO EXPLICIT SEEK
                     0222   685         EXFUNC  RETRY,F_SEARCHA                 ;SEARCH AHEAD OF STARTING SECTOR
                     022A   686
                     022A   687 ;
                     022A   688 ; DATA TRANSFER - REQUEST CHANNEL
                     022A   689 ;
                     022A   690
                     022A   691 TRANRQCH:                                    ;DATA TRANSFER REQUEST CHANNEL
                     022A   692         REQPCHAN LOW                         ;REQUEST PRIMARY CHANNEL FOR TRANSFER
                     0230   693
                     0230   694 ;
                     0230   695 ; DATA TRANSFER - CHANNEL ALREADY OWNED
                     0230   696 ;
                     0230   697
                     0230   698 TRANNOCH:                                    ;DATA TRANSFER CHANNEL OWNED
           50   0092 C5   9A  0230   699         MOVZBL  UCB$B_FEX(R5),R0             ;GET FUNCTION DISPATCH INDEX
                     0235   700         EXFUNC  TRANXT                       ;EXECUTE TRANSFER FUNCTION
                     023A   701
                     023A   702 ;
                     023A   703 ; DATA CHECK
                     023A   704 ;
                     023A   705
                     023A   706 DATACHECK:                                   ;DATA CHECK
        43 009A C5   0E  E1  023A   707         BBC     #IO$V_DATACHECK,UCB$W_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
           50   0639 8F   3C  0240   708         MOVZWL  #SS$_BASECC,R0               ;ASSUME ECC CORRECTION WAS MADE
        3C 68 A5   00  E0  0245   709         BBS     #UCB$V_ECC,UCB$W_DEVSTS(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
                     024A   710         RELCHAN                              ;RELEASE CHANNEL
           00C9 C5   19  90  0250   711         MOVB    #<RP_OF_M_DCK!-              ;SET DATA CHECK IN PROGRESS,
                     0255   712                 RP_OF_M_ECI!-                ;INHIBIT ECC CORRECTION, AND
                     0255   713                 RP_OF_M_FMT>/256,UCB$W_OFFSET+1(R5) ;SET FORMAT
           00CB C5   01  90  0255   714         MOVB    #1,UCB$B_OFFRTC(R5)          ;SET INITIAL OFFSET RETRY COUNT
                00CA C5   94  025A   715         CLRB    UCB$B_OFFNDX(R5)             ;CLEAR INITIAL OFFSET TABLE INDEX
              52   58 A5   D0  025E   716         MOVL    UCB$L_IRP(R5),R2             ;GET ADDRESS OF IRP
           78 A5   2C A2   7D  0262   717         MOVQ    IRP$L_SVAPTE(R2),UCB$L_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
        00BC C5   38 A2   D0  0267   718         MOVL    IRP$L_MEDIA(R2),UCB$W_DA(R5) ;
                     026D   719
                     026D   720 ;
                     026D   721 ; DATA CHECK RETRY
                     026D   722 ;
                     026D   723
                     026D   724 CHECKRETRY:                                  ;DATA CHECK RETRY
                     026D   725         REQPCHAN LOW                         ;REQUEST PRIMARY CHANNEL FOR DATA CHECK
           50   0092 C5   9A  0273   726         MOVZBL  UCB$B_FEX(R5),R0             ;GET FUNCTION DISPATCH INDEX
        50   FDB0 CF40   9A  0278   727         MOVZBL  CHECKTAB-CDF_WRITEDATA[R0],R0 ;GET CASE TABLE INDEX
                     027E   728         EXFUNC  TRANXT                       ;EXECUTE DATA CHECK FUNCTION
                     0283   729
                     0283   730 ;
                     0283   731 ; SUCCESSFUL OPERATION COMPLETION
                     0283   732 ;
                     0283   733
                     0283   734 NORMAL:                                      ;
```

```
                50   01   3C   0283   735              MOVZWL  S^#SS$_NORMAL,R0                ;SET NORMAL COMPLETION STATUS
                                0286   736  CHECKXT:                                        ;
                     01F6   31   0286   737              BRW     FUNCXT                      ;
                                0289   738
                                0289   739  ;
                                0289   740  ; TRANSFER ENDED WITH A RETRIABLE ERROR
                                0289   741  ;
                                0289   742
                                0289   743  TRANXT:                                         ;TRANSFER EXIT
              0093 C5   0B   91 0289   744              CMPB    #CDF_WRITEDATA,UCB$B_CEX(R5) ;WRITE DATA FUNCTION?
                        1B   13 028E   745              BEQL    RETRY                        ;IF EQL YES
              0093 C5   0D   91 0290   746              CMPB    #CDF_WRITEHEAD,UCB$B_CEX(R5) ;WRITE HEADER FUNCTION?
                        14   13 0295   747              BEQL    RETRY                        ;IF EQL YES
           51  00064F74 8F   D3 0297   748              BITL    #MBA$M_SR_DLT!-              ;DATA LATE OR,
                                029E   749                      MBA$M_SR_INVMAP!-            ;INVALID MAP REGISTER OR,
                                029E   750                      MBA$M_SR_MAPPE!-             ;MAP REGISTER PARITY ERROR OR,
                                029E   751                      MBA$M_SR_MCPE!-              ;MASSBUS CONTROL PARITY ERROR OR,
                                029E   752                      MBA$M_SR_SPE!-               ;SILO PARITY ERROR OR,
                                029E   753                      MBA$M_SR_MDPE!-              ;MASSBUS DATA PARITY ERROR OR,
                                029E   754                      MBA$M_SR_MXF!-               ;MISSED TRANSFER OR,
                                029E   755                      MBA$M_SR_NED!-               ;NONEXISTENT DISK OR,
                                029E   756                      MBA$M_SR_RDS!-               ;READ DATA SUBSTITUTE OR,
                                029E   757                      MBA$M_SR_WCKLWR!-            ;WRITE CHECK LOWER BYTE OR,
                                029E   758                      MBA$M_SR_WCKUPR,R1           ;WRITE CHECK UPPER BYTE?
                        0B   12 029E   759              BNEQ    RETRY                        ;IF NEQ YES - RETRY FUNCTION
                 0A 52  08   E0 02A0   760              BBS     #RP_ER1_V_HCRC,R2,ECC        ; First check HCRC. If bad go to ECC.
              52 20B8 8F   B3 02A4   761              BITW    #RP_ER1_M_FER!-              ;FORMAT ERROR OR,
                                02A9   762                      RP_ER1_M_HCE!-               ; Header Compare Error.
                                02A9   763                      RP_ER1_M_OPI!-               ;OPERATION INCOMPLETE OR,
                                02A9   764                      RP_ER1_M_PAR!-               ;PARITY ERROR OR,
                                02A9   765                      RP_ER1_M_WCF,R2              ;WRITE CLOCK FAIL?
                        03   13 02A9   766              BEQL    ECC                          ;IF EQL NO
                                02AB   767  RETRY:                                          ;
                     0110   31 02AB   768              BRW     RETRYERR                     ;RETRIABLE ERROR
                                02AE   769
                                02AE   770  ;
                                02AE   771  ; ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
                                02AE   772  ;
                                02AE   773
                                02AE   774  ECC:                                            ;ECC CORRECTION
     51   7E A5   00C0 C5   A1 02AE   775              ADDW3   UCB$W_BCR(R5), -             ; Compute bytes transfered then
                                02B5   776                      UCB$W_BCNT(R5), R1           ; clear byte offset bits and
     50   51   FFFF01FF 8F   CB 02B5   777              BICL3   #^XFFFF01FF, R1, R0          ; convert result to a longword.
                        6D   13 02BD   778              BEQL    OFF                          ; Branch if whole blocks xfered is zero.
                  51    01FF 8F   B3 02BF   779              BITW    #^X1FF, R1                   ; Was a partial block transfered?
                        66   12 02C4   780              BNEQ    OFF                          ; Branch if partial block transfered.
                  52    0180 8F   B3 02C6   781              BITW    #RP_ER1_M_HCE!-              ; Was there an error while processing
                                02CB   782                      RP_ER1_M_HCRC, R2            ; the header?
                        07   12 02CB   783              BNEQ    10$                          ; Branch if header error.
              50   00000200 8F   C2 02CD   784              SUBL2   #512, R0                     ; Else, reduce bytes xfered by a block.
                  52   11C0 8F   B3 02D4   785  10$:        BITW    #RP_ER1_M_DTE!-              ; For:  DRIVE TIMING ERROR
                                02D9   786                      RP_ER1_M_ECH!-               ;       ECC HARD ERROR
                                02D9   787                      RP_ER1_M_HCE!-               ;       HEADER COMPARE ERROR
                                02D9   788                      RP_ER1_M_HCRC,R2             ;       HEADER CRC ERROR
                        51   12 02D9   789              BNEQ    OFF                          ;         perform offset recovery.
              4B 00C8 C5   0B   E0 02DB   790              BBS     #RP_OF_V_ECI,UCB$W_OFFSET(R5),OFF  ; Branch if ECC inhibited.
                     7E   52   7D 02E1   791              MOVQ    R2,-(SP)                     ; Save work registers.
```

```
                52   00C6 C5    0B    00    EA  02E4   792          FFS     #0,#11,UCB$W_EC2(R5),R2 ; Find the first error bit in the ECC
                                                02EB   793                                          ;  pattern.
                          53    0A    52    C3  02EB   794          SUBL3   R2,#10,R3               ; Get the number of error bits
                                                02EF   795                                          ;  remaining in the pattern.
                                09    15    02EF   796          BLEQ    20$                     ; Branch if no other bits in pattern.
                                52    D6    02F1   797          INCL    R2                      ; Point ot next bit in pattern.
                52   00C6 C5    53    52    EF  02F3   798          EXTZV   R2,R3,UCB$W_EC2(R5),R2  ; Is there more than one error bit set?
                                      0C    BA  02FA   799 20$:     POPR    #^M<R3,R2>              ; Restore work registers without
                                                02FC   800                                          ;  affecting flags.
                                29    1A    02FC   801          BGTRU   DEFER_ECC               ; If more than one error bit set, don't
                                                02FE   802                                          ;  apply ECC correction.
                                                02FE   803 ;
                                                02FE   804 ; APPLY_ECC -
                                                02FE   805 ;
                                                02FE   806 ; Apply ECC correction to correct a single bit error.
                                                02FE   807 ;
                                                02FE   808
                                                02FE   809 APPLY_ECC:
                          7E    51    3C    02FE   810          MOVZWL  R1, -(SP)               ; Save total bytes transfered, inc. ECC.
                00000000'GF     16    0301   811          JSB     G^IOC$APPLYECC          ;APPLY ECC CORRECTION
                          50  8ED0    0307   812          POPL    R0                      ;RETRIEVE TRANSFERED BYTE COUNT
                00000000'GF     16    030A   813          JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
                      00CA C5   94    0310   814          CLRB    UCB$B_OFFNDX(R5)        ; Reset offset table index.
                                      0314   815          EXFUNC  FATALERR,F_RETCENTER    ; Return to centerline.
                             7E A5    B5    031C   816          TSTW    UCB$W_BCNT(R5)          ;ANY MORE TO TRANSFER?
                                03    13    031F   817          BEQL    20$                     ;IF EQL NO
                              FF0C    31    0321   818          BRW     TRANNOCH                ;TRANSFER NEXT SEGMENT
                              FF13    31    0324   819 20$:     BRW     DATACHECK               ;CHECK FOR WRITE CHECK
                                      0327   820
                                      0327   821 ;
                                      0327   822 ; DEFER_ECC -
                                      0327   823 ;
                                      0327   824 ; Don't apply ECC correction for multiple bit errors unless the error cannot
                                      0327   825 ; be recovered with offset retries.
                                      0327   826 ;
                                      0327   827
                                      0327   828 DEFER_ECC:
                          04    88    0327   829          BISB    #ERL_M_ECC_DEFER,-      ; Set flag to indicate that ECC
                      00D2 C5         0329   830                  UCB$B_DB_ERL(R5)        ;  can be used if offset recovery fails.
                                      032C   831
                                      032C   832 ;
                                      032C   833 ; OFF - OFFSET RECOVERY
                                      032C   834 ;
                                      032C   835 ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER COMPARE, OR ECC
                                      032C   836 ; HARD ERROR IS DETECTED ON A READ FUNCTION.
                                      032C   837 ;
                                      032C   838
                                      032C   839 OFF:                                    ;OFFSET RECOVERY
                          50    D5    032C   840          TSTL    R0                      ;ANY GOOD DATA TRANSFERED?
                          2E    13    032E   841          BEQL    20$                     ;IF EQL NO
                                      0330   842
                                      0330   843 ;
                                      0330   844 ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
                                      0330   845 ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
                                      0330   846 ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
                                      0330   847 ; POINT OF ERROR.
                                      0330   848 ;
```

```
                     0330   849
 00000000'GF   16    0330   850          JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
     00CA C5   94    0336   851          CLRB    UCB$B_OFFNDX(R5)        ;RESET OFFSET TABLE INDEX
00CB C5    10   90   033A   852 10$:     MOVB    #16,UCB$B_OFFRTC(R5)    ;SET OFFSET RETRY COUNT
00CA C5    08   91   033F   853          CMPB    #OFFSIZ,UCB$B_OFFNDX(R5) ;ALL OFFSETS TRIED?
           08   12   0344   854          BNEQ    15$                     ; Branch if not.
           02   E4   0346   855          BBSC    #ERL_V_ECC_DEFER,-      ; Correct the error with ECC if we can.
      00D2 C5         0348   856                  UCB$B_DB_ERL(R5),-      ;
           B2         034B   857                  APPLY_ECC               ;
           65   11    034C   858          BRB     OFFSETERR               ; Otherwise, fatal error.
                      034E   859 15$:     RELCHAN                         ;RELEASE CHANNEL
                      0354   860          EXFUNC  FATALERR,F_RETCENTER    ;RETURN TO CENTERLINE
           44   11    035C   861          BRB     50$                     ;
                      035E   862
                      035E   863 ;
                      035E   864 ; NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
                      035E   865 ;
                      035E   866
52   90C0 8F    B3    035E   867 20$:     BITW    #RP_ER1_M_DCK!-         ;DATA CHECK OR,
                      0363   868                  RP_ER1_M_DTE!-          ;DRIVE TIMING OR,
                      0363   869                  RP_ER1_M_ECH!-          ;ECC HARD ERROR OR,
                      0363   870                  RP_ER1_M_HCE,R2         ;HEADER COMPARE ERROR?
           05   12    0363   871          BNEQ    30$                     ;IF NEQ YES
00C9 C5    04   88    0365   872          BISB    #RP_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;SET HEADER COMPARE INHIBIT
     00CB C5   97    036A   873 30$:     DECB    UCB$B_OFFRTC(R5)        ;CHANGE CURRENT OFFSET?
           37   12    036E   874          BNEQ    60$                     ;IF NEQ NO
     00CA C5   96    0370   875          INCB    UCB$B_OFFNDX(R5)        ;UPDATE OFFSET TABLE INDEX
50   00CA C5   9A    0374   876          MOVZBL  UCB$B_OFFNDX(R5),R0     ;GET NEXT OFFSET TABLE INDEX
50 FD1C CF40   9A    0379   877          MOVZBL  OFFTAB-1[R0],R0         ;GET NEXT OFFSET VALUE?
           B9   13    037F   878          BEQL    10$                     ;IF EQL RETURN TO CENTERLINE
  18 A3    02   D3    0381   879          BITL    #2,RP_DT(R3)            ;RP06 DRIVE?
           03   12    0385   880          BNEQ    40$                     ;IF NEQ YES
        50  02   C4    0387   881          MULL    #2,R0                   ;CONVERT TO RP04 OFFSET VALUE
00C8 C5    50   90    038A   882 40$:     MOVB    R0,UCB$W_OFFSET(R5)     ;SET NEW OFFSET VALUE
00CB C5    02   90    038F   883          MOVB    #2,UCB$B_OFFRTC(R5)     ;SET OFFSET RETRY COUNT
                      0394   884          RELCHAN                         ;RELEASE CHANNEL
                      039A   885          EXFUNC  FATALERR,F_OFFSET       ;OFFSET TO NEXT POSITION
00C9 C5    04   8A    03A2   886 50$:     BICB    #RP_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;CLEAR HEADER COMPARE INHIBIT
03 00C8 C5    08   E0 03A7   887 60$:     BBS     #RP_OF_V_DCK,UCB$W_OFFSET(R5),70$ ;IF SET, DATA CHECK FUNCTION
          FE7A   31   03AD   888          BRW     TRANRQCH                ;TRY FUNCTION AGAIN
          FEBA   31   03B0   889 70$:     BRW     CHECKRETRY              ;TRY DATA CHECK AGAIN
                      03B3   890
                      03B3   891 ;
                      03B3   892 ; ALL OFFSETS TRIED - RETRIEVE FINAL TRANSFER STATUS
                      03B3   893 ;
                      03B3   894
                      03B3   895 OFFSETERR:                               ;OFFSET RECOVERY ERROR
   50   04 A3    D0   03B3   896          MOVL    RP_DS(R3),R0            ;RETRIEVE FINAL DRIVE STATUS
51    00CE C5    D0   03B7   897          MOVL    UCB$L_DB_SR(R5),R1      ;RETRIEVE FINAL ERROR STATUS
           2D   11    03BC   898          BRB     FATALERR                ; Branch around.
                      03BE   899
                      03BE   900 ;
                      03BE   901 ; RETRIABLE ERROR
                      03BE   902 ;
                      03BE   903
                      03BE   904 RETRYERR:                                ;RETRIABLE ERROR
           07   BB    03BE   905          PUSHR   #^M<R0,R1,R2>           ; Save volital error status registers.
```

```
                     03C0   906          RELCHAN                         ; Release channel before possible RECAL
            07   BA  03C6   907          POPR    #^M<R0,R1,R2>           ; Restore error status registers.
                     03C8   908
                     03C8   909 ;
                     03C8   910 ; Here we will do a RECAL if we had either a Seek Incomplete or a Header
                     03C8   911 ;       Compare Error.
                     03C8   912 ;
                     03C8   913
            0E   E0  03C8   914          BBS     #RP_ER3_V_SKI,-         ; If Seek Incomplete
 04 00CC C5          03CA   915                  UCB$W_DB_ER3(R5),10$    ;  then go do RECAL.
 08 52      07   E1  03CE   916          BBC     #RP_ER1_V_HCE,R2,20$    ; If NOT HCE, then branch around RECAL.
                     03D2   917 10$:     EXFUNC  FATALERR,F_RECAL        ; Do RECAL for SKI or HCE.
     0080 C5    97  03DA   918 20$:     DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
            0B   13  03DE   919          BEQL    FATALERR                ;IF EQL NO
                     03E0   920          EXFUNC  FATALERR,F_DRVCLR       ; Issue drive clear before retrying.
          FDC1   31  03E8   921          BRW     FDISPATCH               ;
                     03EB   922
                     03EB   923 ;
                     03EB   924 ; FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
                     03EB   925 ; INHIBITED, OR FINAL OFFSET TRIED
                     03EB   926 ;
                     03EB   927
                     03EB   928 FATALERR:                                ;FATAL ERROR - SET STATUS
   0E 50     0C  E0  03EB   929          BBS     #RP_DS_V_MOL,R0,10$     ; Branch if medium is online.
50  01A4 8F      3C  03EF   930          MOVZWL  #SS$_MEDOFL,R0          ; Otherwise, set medium offline status,
     0800 8F      AA  03F4   931          BICW    #UCB$M_VALID,-          ;  clear software volume valid,
        64 A5          03F8   932                  UCB$W_STS(R5)           ;
          0082   31  03FA   933          BRW     FUNCXT                  ;  and branch to common exit.
   79 50     06  E1  03FD   934 10$:     BBC     #RP_DS_V_VV,R0,20$      ;IF CLR, VOLUME INVALID
50  023C 8F      3C  0401   935          MOVZWL  #SS$_UNSAFE,R0          ;SET DRIVE UNSAFE STATUS
   75 52      0E   E0  0406   936          BBS     #RP_ER1_V_UNS,R2,FUNCXT ;IF SET, DRIVE UNSAFE
50  02D4 8F      3C  040A   937          MOVZWL  #SS$_OPINCOMPL,R0       ;SET OPERATION INCOMPLETE STATUS
   6C 52      0D   E0  040F   938          BBS     #RP_ER1_V_OPI,R2,FUNCXT ;IF SET, OPERATION INCOMPLETE
50  00BC 8F      3C  0413   939          MOVZWL  #SS$_FORMAT,R0          ;SET FORMAT ERROR STATUS
   63 52      04   E0  0418   940          BBS     #RP_ER1_V_FER,R2,FUNCXT ;IF SET, FORMAT ERROR
50  025C 8F      3C  041C   941          MOVZWL  #SS$_WRITLCK,R0         ;SET WRITE LOCK ERROR STATUS
   5A 52      0B   E0  0421   942          BBS     #RP_ER1_V_WLE,R2,FUNCXT ;IF SET, WRITE LOCK ERROR
50  0134 8F      3C  0425   943          MOVZWL  #SS$_IVADDR,R0          ;SET INVALID DISK ADDRESS STATUS
52  0600 8F      B3  042A   944          BITW    #RP_ER1_M_AOE!-         ;DISK ADDRESS OVERFLOW OR,
                     042F   945                  RP_ER1_M_IAE,R2         ;INVALID DISK ADDRESS ERROR?
            4E   12  042F   946          BNEQ    FUNCXT                  ;IF NEQ YES
50  008C 8F      3C  0431   947          MOVZWL  #SS$_DRVERR,R0          ;SET DRIVE ERROR STATUS
52  1027 8F      B3  0436   948          BITW    #RP_ER1_M_DTE!-         ;DRIVE TIMING ERROR OR,
                     043B   949                  RP_ER1_M_ILF!-          ;ILLEGAL FUNCTION OR,
                     043B   950                  RP_ER1_M_ILR!-          ;ILLEGAL REGISTER OR,
                     043B   951                  RP_ER1_M_RMR!-          ;REGISTER MODIFY REFUSE OR,
                     043B   952                  RP_ER1_M_WCF,R2         ;WRITE CLOCK FAIL ERROR?
            42   12  043B   953          BNEQ    FUNCXT                  ;IF NEQ YES
50  01F4 8F      3C  043D   954          MOVZWL  #SS$_PARITY,R0          ; Set parity error status.
52  8140 8F      B3  0442   955          BITW    #RP_ER1_M_DCK!-         ; Data check error or,
                     0447   956                  RP_ER1_M_ECH!-          ;  ECC hard error or,
                     0447   957                  RP_ER1_M_HCRC,R2        ;  header CRC error?
            36   12  0447   958          BNEQ    FUNCXT                  ; Branch if so.
50  0054 8F      3C  0449   959          MOVZWL  #SS$_CTRLERR,R0         ; Set fatal controller error status.
52  0088 8F      B3  044E   960          BITW    #RP_ER1_M_HCE!-         ; Header compare error or,
                     0453   961                  RP_ER1_M_PAR,R2         ;  parity error?
            2A   12  0453   962          BNEQ    FUNCXT                  ; Branch if so.
```

```
                51   00024064 8F    D3  0455   963          BITL    #MBA$M_SR_MAPPE!-          ;MAP PARITY ERROR OR,
                                        045C   964                  MBA$M_SR_MCPE!-            ;MASSBUS CONTROL PARITY ERROR OR,
                                        045C   965                  MBA$M_SR_SPE!-             ;SILO PARITY ERROR OR,
                                        045C   966                  MBA$M_SR_MDPE!-            ;MASSBUS DATA PARITY ERROR OR,
                                        045C   967                  MBA$M_SR_RDS,R1            ;READ DATA SUBSTITUTE?
                                 21     12  045C   968          BNEQ    FUNCXT                     ;IF NEQ YES
                     50   005C 8F     3C  045E   969          MOVZWL  #SS$_DATACHECK,R0          ;SET DATA CHECK ERROR STATUS
                     51   0600 8F     B3  0463   970          BITW    #MBA$M_SR_WCKLWR!-         ;WRITE CHECK ERROR LOWER BYTE OR,
                                        0468   971                  MBA$M_SR_WCKUPR,R1         ;WRITE CHECK ERROR UPPER BYTE?
                                 15     12  0468   972          BNEQ    FUNCXT                     ;IF NEQ YES
                     50   01C4 8F     3C  046A   973          MOVZWL  #SS$_NONEXDRV,R0           ;SET NONEXISTENT DRIVE STATUS
                       0C 51      12     E0  046F   974          BBS     #MBA$V_SR_NED,R1,FUNCXT    ;IF SET, NONEXISTENT DRIVE
                     50   0054 8F     3C  0473   975          MOVZWL  #SS$_CTRLERR,R0            ;SET CONTROLLER ERROR STATUS
                                 05     11  0478   976          BRB     FUNCXT                     ;
                     50   0254 8F     3C  047A   977 20$:     MOVZWL  #SS$_VOLINV,R0             ;SET VOLUME INVALID STATUS
                                        047F   978
                                        047F   979 ;
                                        047F   980 ; FUNCTION COMPLETION COMMON EXIT
                                        047F   981 ;
                                        047F   982
                                        047F   983 FUNCXT:                                     ;FUNCTION EXIT
                                 50     DD  047F   984          PUSHL   R0                         ;SAVE FINAL REQUEST STATUS
                  00000000'  GF     16  0481   985          JSB     G^IOC$DIAGBUFILL           ;FILL DIAGNOSTIC BUFFER IF PRESENT
                                        0487   986          RELCHAN                            ;RELEASE CHANNEL IF OWNED
                   0092 C5     0A     91  048D   987          CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                                 13     1A  0492   988          BGTRU   10$                        ;IF GTRU YES
                   0092 C5     10     91  0494   989          CMPB    #CDF_READPRESET,UCB$B_FEX(R5) ;READIN PRESET FUNCTION?
                                 0C     13  0499   990          BEQL    10$                        ;IF EQL YES
                     53   58    A5     D0  049B   991          MOVL    UCB$L_IRP(R5),R3           ;RETRIEVE ADDRESS OF IRP
          02 AE      32 A3    00C0   C5     A1  049F   992          ADDW3   UCB$W_BCR(R5),IRP$W_BCNT(R3),2(SP) ;CALCULATE BYTES TRANSFERED
                                 51     D4  04A7   993 10$:     CLRL    R1                         ;CLEAR SECOND STATUS LONGWORD
                                 50   8ED0  04A9   994          POPL    R0                         ;RETRIEVE FINAL REQUEST STATUS
                     53   0091 C5     9A  04AC   995          MOVZBL  UCB$B_SLAVE+1(R5),R3       ;Get drive offset constant
                  53   0400 C443     DE  04B1   996          MOVAL   MBA$L_ERB(R4)[R3],R3       ;Get address of driver registers
                          63     09     9A  04B7   997          MOVZBL  #F_DRVCLR!1,RP_CS1(R3)     ;Issue drive clear before release
                          63     0B     9A  04BA   998          MOVZBL  #F_RELEASE!1,RP_CS1(R3)    ;Release port
                                        04BD   999          REQCOM                             ;COMPLETE REQUEST
```

```
                        1001          .SBTTL  RP04/05/06 HARDWARE FUNCTION EXECUTION
                        1002 ;
                        1003 ; FEX - RP04/05/06 HARDWARE FUNCTION EXECUTION
                        1004 ;
                        1005 ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
                        1006 ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
                        1007 ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
                        1008 ; INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. IF THE FUNCTION IS AN
                        1009 ; IMMEDIATE FUNCTION CONTROL RETURNS IMMEDIATELY. ELSE THE RETURN ADDRESS
                        1010 ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                        1011 ; RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
                        1012 ;
                        1013 ; INPUTS:
                        1014 ;
                        1015 ;       R0 = FUNCTION TABLE DISPATCH INDEX.
                        1016 ;       R3 = ADDRESS OF DRIVE CONTROL STATUS REGISTER 1.
                        1017 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                        1018 ;       R5 = DEVICE UNIT UCB ADDRESS.
                        1019 ;
                        1020 ;       00(SP) = RETURN ADDRESS OF CALLER.
                        1021 ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
                        1022 ;
                        1023 ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
                        1024 ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
                        1025 ;
                        1026 ; OUTPUTS:
                        1027 ;
                        1028 ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
                        1029 ;
                        1030 ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
                        1031 ;             OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
                        1032 ;             ERROR ROUTINE.
                        1033 ;
                        1034 ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
                        1035 ;             ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
                        1036 ;             INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
                        1037 ;
                        1038 ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
                        1039 ;             OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
                        1040 ;             IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
                        1041 ;
                        1042 ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
                        1043 ;             DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
                        1044 ;
                        1045 ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
                        1046 ;
                        1047 ;       IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
                        1048 ;       THE GENERAL REGISTERS R0, R1, AND R2, AND THE UCB.
                        1049 ;
                        1050 ;       R0 = DRIVE STATUS REGISTER.
                        1051 ;       R1 = MBA STATUS REGISTER.
                        1052 ;       R2 = DRIVE ERROR REGISTER 1.
                        1053 ;
                        1054 ;       UCB$W_EC1(R5) = ECC POSITION REGISTER.
                        1055 ;       UCB$W_EC2(R5) = ECC PATTERN REGISTER.
                        1056 ;       UCB$W_BCR(R5) = BYTE COUNT REGISTER.
                        1057 ;
```

```
                        04C3  1058
                        04C3  1059 FEX:
            009C C5 8ED0  04C3  1060          POPL    UCB$L_DPC(R5)                   ;FUNCTION EXECUTOR
                                                                                      ;SAVE DRIVER PC VALUE
          0093 C5   50    90  04C8  1061          MOVB    R0,UCB$B_CEX(R5)            ;SAVE CASE INDEX
        53   0091 C5    9A  04CD  1062          MOVZBL  UCB$B_SLAVE+1(R5),R3        ;GET DRIVE OFFSET CONSTANT
    53     0400 C443    DE  04D2  1063          MOVAL   @DA$L_ERB(R4)[R3],R3        ;GET ADDRESS OF DRIVE REGISTERS
38 A5    00008000 8F    D3  04D8  1064          BITL    #DEV$M_DUA,UCB$L_DEVCHAR(R5) ;DUAL PORTED DRIVE?
                40    12  04E0  1065          BNEQ    SEIZE                       ;IF NEQ, YES
      50   0093 C5    9A  04E2  1066 GO:      MOVZBL  UCB$B_CEX(R5),R0            ;RESTORE CASE INDEX (FUNC. CODE)
                        04E7  1067          CASE    R0,<-                       ;DISPATCH TO PROPER FUNCTION ROUTINE
                        04E7  1068                  POSIT,-                     ;SEEK CYLINDER
                        04E7  1069                  EXFNC,-                     ;RECALIBRATE
                        04E7  1070                  IMMED,-                     ;DRIVE CLEAR
                        04E7  1071                  IMMED,-                     ;RELEASE DRIVE
                        04E7  1072                  EXFNC,-                     ;OFFSET HEADS
                        04E7  1073                  EXFNC,-                     ;RETURN TO CENTERLINE
                        04E7  1074                  IMMED,-                     ;PACK ACKNOWLEDGE
                        04E7  1075                  POSIT,-                     ;SEARCH FOR SECTOR
                        04E7  1076                  XFER,-                      ;WRITE CHECK
                        04E7  1077                  XFER,-                      ;WRITE DATA
                        04E7  1078                  XFER,-                      ;READ DATA
                        04E7  1079                  XFER,-                      ;WRITE HEADER AND DATA
                        04E7  1080                  XFER,-                      ;READ HEADER AND DATA
                        04E7  1081                  XFER,-                      ;WRITE CHECK HEADER AND DATA
                        04E7  1082                  IMMED,-                     ;READIN PRESET
                        04E7  1083                  SEARCHA,-                   ;SEARCH AHEAD FOR SECTOR
                        04E7  1084                  >,LIMIT=#CDF_SEEK           ;
                        050B  1085
                        050B  1086 ;
                        050B  1087 ; IMMEDIATE FUNCTION EXECUTION
                        050B  1088 ;
                        050B  1089 ;       FUNCTIONS INCLUDE:
                        050B  1090 ;
                        050B  1091 ;               NO OPERATION,
                        050B  1092 ;               UNLOAD VOLUME,
                        050B  1093 ;               DRIVE CLEAR,
                        050B  1094 ;               RELEASE PORT,
                        050B  1095 ;               READ IN PRESET, AND
                        050B  1096 ;               PACK ACKNOWLEDGE.
                        050B  1097 ;
                        050B  1098 ; THESE FUNCTIONS ARE EXECUTED IMMEDIATELY AND THE FINAL DEVICE REGISTERS
                        050B  1099 ; ARE RETURNED TO THE CALLER.
                        050B  1100 ;
                        050B  1101
                        050B  1102 IMMED:                                         ;IMMEDIATE FUNCTION EXECUTION
                        050B  1103          DSBINT                                ;DISABLE INTERRUPTS
  09 64 A5     05    E0  0511  1104          BBS     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF SET, POWER HAS FAILED
        63     09    9A  0516  1105          MOVZBL  #F_DRVCLR!1,RP_CS1(R3)        ;CLEAR DRIVE ERRORS
63    FB6B CF40    9A  0519  1106          MOVZBL  FTAB[R0],RP_CST(R3)           ;EXECUTE FUNCTION
           00B2    31  051F  1107 10$:     BRW     ENBXIT                        ;
                        0522  1108
                        0522  1109 ;
                        0522  1110 ; ATTEMPT TO SEIZE THE PORT ON A DUAL PORTED DISK.
                        0522  1111 ;
                        0522  1112
BA 00D2 C5    01    E1  0522  1113 SEIZE:  BBC     #ERL_V_DUALPORT, -           ;IF CLEAR, NO DUAL PORT KIT IN DRIVE
                        0528  1114                  UCB$B_DB_ERL(R5),GO
```

```
                     0528  1115          DSBINT                          ;DISABLE INTERRUPTS
          04 A3   D4 052E  1116          CLRL    RP_DS(R3)               ;ATTEMPT TO SEIZE PORT
00000100 8F   D3 0531  1117          BITL    #RP_DS_M_DPR,-          ;DID WE SEIZED THE PORT?
          04 A3      0537  1118                  RP_DS(R3)
             12   12 0539  1119          BNEQ    2$                      ;IF NEQ, WE SEIZED THE PORT
                     0538  1120          WFIKPCH RETREG,#15              ;LETS WAIT FOR THE PORT, ELSE TIMEOUT
                     0545  1121          IOFORK                          ;CREATE FORK PROCESS
             95   11 0548  1122          BRB     GO                      ;LETS CONTINUE WE HAVE THE PORT
                     054D  1123 2$:      ENBINT                          ;ENABLE INTERRUPTS
             90   11 0550  1124          BRB     GO                      ;LETS CONTINUE WE HAVE THE PORT
                     0552  1125
                     0552  1126 ; SEARCH AHEAD FUNCTION EXECUTION
                     0552  1127 ;
                     0552  1128 ; THIS FUNCTION MINIMIZES ROTATIONAL LATENCY BY SEARCHING FOR THE SECTOR THAT IS
                     0552  1129 ; FOUR SECTORS AHEAD OF THE STARTING SECTOR OF A TRANSFER.
                     0552  1130 ;
                     0552  1131 ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                     0552  1132 ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                     0552  1133 ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                     0552  1134 ;
                     0552  1135
                     0552  1136 SEARCHA:                                 ;
51   00BC C5   3C    0552  1137          MOVZWL  UCB$W_DA(R5),R1         ;GET DESIRED TRACK AND SECTOR ADDRESS
     51   04   82    0557  1138          SUBB    #4,R1                   ;COMPUTE FOUR SECTORS BEFORE SPECIFIED SECTO
          04   18    055A  1139          BGEQ    10$                     ;IF GEQ BEFORE SECTOR ZERO
  51   44 A5   80    055C  1140          ADDB    UCB$B_SECTORS(R5),R1    ;CONVERT TO AFTER SECTOR ZERO
  14 A3    51   D0 0560  1141 10$:      MOVL    R1,RP_DA(R3)            ;SET TRACK AND SECTOR ADDRESS
             15   11 0564  1142          BRB     LDCYL                   ;
                     0566  1143
                     0566  1144 ;
                     0566  1145 ; TRANSFER FUNCTION EXECUTION
                     0566  1146 ;
                     0566  1147 ;       FUNCTIONS INCLUDE:
                     0566  1148 ;
                     0566  1149 ;               WRITE CHECK,
                     0566  1150 ;               WRITE CHECK HEADER AND DATA,
                     0566  1151 ;               WRITE DATA,
                     0566  1152 ;               WRITE HEADER AND DATA,
                     0566  1153 ;               READ DATA, AND
                     0566  1154 ;               READ HEADER AND DATA.
                     0566  1155 ;
                     0566  1156 ; THE MAP REGISTERS, BYTE COUNT REGISTER, AND VIRTUAL ADDRESS REGISTER ARE
                     0566  1157 ; LOADED FOLLOWED BY THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS.
                     0566  1158 ; THE FUNCTION IS INITIATED AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
                     0566  1159 ; INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                     0566  1160 ;
                     0566  1161 ; IT ASSUMED THAT THE CALLER OWNS THE CHANNEL ON WHICH THE I/O IS TO OCCUR.
                     0566  1162 ;
                     0566  1163
                     0566  1164 XFER:                                   ;TRANSFER FUNCTION EXECUTION
  08 A4    00   D2 0566  1165          MCOML   #0,MBA$L_SR(R4)         ;CLEAR MASSBUS ADAPTER ERRORS
                     056A  1166          LOADMBA                         ;LOAD MAP, BYTE COUNT, AND VIRTUAL ADDRESS
50   0093 C5   9A    0570  1167          MOVZBL  UCB$B_CEX(R5),R0        ;RETRIEVE FUNCTION TABLE INDEX
                     0575  1168
                     0575  1169 ;
                     0575  1170 ; POSITIONING FUNCTION EXECUTION
                     0575  1171 ;
```

```
                     0575  1172 ;       FUNCTIONS INCLUDE:
                     0575  1173 ;
                     0575  1174 ;               SEEK CYLINDER, AND
                     0575  1175 ;               SEARCH FOR SECTOR.
                     0575  1176 ;
                     0575  1177 ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                     0575  1178 ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                     0575  1179 ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                     0575  1180 ;
                     0575  1181
                     0575  1182 POSIT:                                          ;POSITION FUNCTION EXECUTION
14 A3   00BC C5  3C  0575  1183         MOVZWL  UCB$W_DA(R5),RP_DA(R3)          ;SET DESIRED TRACK AND SECTOR ADDRESS
                     057B  1184 LDCYL:                                          ;
28 A3   00C2 C5  3C  057B  1185         MOVZWL  UCB$W_DC(R5),RP_DC(R3)          ;SET DESIRED CYLINDER ADDRESS
                     0581  1186
                     0581  1187 ;
                     0581  1188 ; INTERRUPT WAIT FUNCTION EXECUTION
                     0581  1189 ;
                     0581  1190 ;       FUNCTIONS INCLUDE:
                     0581  1191 ;
                     0581  1192 ;               OFFSET HEADS,
                     0581  1193 ;               RECALIBRATE, AND
                     0581  1194 ;               RETURN TO CENTERLINE.
                     0581  1195 ;
                     0581  1196 ; THE OFFSET REGISTER IS LOADED, THE FUNCTION IS INITIATED, AND A WAITFOR
                     0581  1197 ; INTERRUPT IS EXECUTED. WHEN THE INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS
                     0581  1198 ; ARE RETURNED TO THE CALLER.
                     0581  1199 ;
                     0581  1200
                     0581  1201 EXFNC:                                          ;EXECUTE FUNCTION
                     0581  1202         DSBINT                                  ;DISABLE INTERRUPTS
 48 64 A5   05   E0  0587  1203         BBS     #UCB$V_POWER,UCB$W_STS(R5),ENBXIT ;IF SET, POWER FAILED
       63   09   9A  058C  1204         MOVZBL  #F_DRVCLR!1,RP_CS1(R3)          ;CLEAR DRIVE ERRORS
24 A3  00C8 C5   3C  058F  1205         MOVZWL  UCB$W_OFFSET(R5),RP_OF(R3)      ;SET FORMAT, INHIBIT BITS, AND OFFSET
52   04 A3   13   78  0595  1206         ASHL    #31-RP_DS_V_MOL,RP_DS(R3),R2    ;MEDIUM ONLINE?
            1E   18  059A  1207         BGEQ    10$                             ;IF GEQ NO
 63   FAE8 CF40   9A  059C  1208         MOVZBL  FTAB[R0],RP_CS1(R3)             ;INITIATE FUNCTION
                     05A2  1209         WFIKPCH RETREG,#15                      ;WAITFOR INTERRUPT AND KEEP CHANNEL
00CE C5   08 A4   D0  05AC  1210         MOVL    MBA$L_SR(R4),UCB$L_DB_SR(R5)    ;SAVE FINAL CONTROLLER STATUS
                     05B2  1211         IOFORK                                  ;CREATE FORK PROCESS
            1D   11  05B8  1212         BRB     RETREG                          ;
                     05BA  1213
                     05BA  1214 ;
                     05BA  1215 ; MEDIUM OFFLINE AT START OF FUNCTION
                     05BA  1216 ;
                     05BA  1217
                     05BA  1218 10$:     ENBINT                                  ;ENABLE INTERRUPTS
       0093 C5   94  05BD  1219         CLRB    UCB$B_CEX(R5)                   ;FORCE DRIVE FUNCTION
 50    4000 8F   3C  05C1  1220         MOVZWL  #RP_DS_M_ERR,R0                 ;SET DRIVE ERROR
       00CE C5   D4  05C6  1221         CLRL    UCB$L_DB_SR(R5)                 ;CLEAR SAVED MBA STATUS REGISTER
             01   88  05CA  1222         BISB    #ERL_M_MEDOFF,-                 ; SET FLAG WHICH INDICATES THAT MEDIUM
       00D2 C5         05CC  1223                 UCB$B_DB_ERL(R5)             ;  WAS OFFLINE AT START OF FUNCTION.
            32   11  05CF  1224         BRB     ERROR                           ;
                     05D1  1225
                     05D1  1226
                     05D1  1227         .ENABL  LSB
                     05D1  1228
```

```
                 0117   31  05D1  1229 10$:    BRW     120$                    ; Branch to special condition handler.
                            05D4  1230
                            05D4  1231 ;
                            05D4  1232 ; ENABLE INTERRUPTS
                            05D4  1233 ;
                            05D4  1234
                            05D4  1235 ENBXIT:                                 ;
                            05D4  1236         ENBINT                          ;ENABLE INTERRUPTS
                            05D7  1237
                            05D7  1238 ;
                            05D7  1239 ; RETURN REGISTERS
                            05D7  1240 ;
                            05D7  1241
                            05D7  1242 RETREG:                                 ;RETURN FINAL DEVICE REGISTERS
                      34 A3 F7  05D7  1243         CVTLW   RP_ER3(R3),-          ; Save register after operation.
                   00CC C5      05DA  1244                 UCB$W_DB_ER3(R5)
          00C4 C5    38 A3 F7  05DD  1245         CVTLW   RP_ECT(R3),UCB$W_EC1(R5) ;SAVE ECC POSITION REGISTER
          00C6 C5    3C A3 F7  05E3  1246         CVTLW   RP_EC2(R3),UCB$W_EC2(R5) ;SAVE ECC PATTERN REGISTER
                            05E9  1247 ; Here we save the more conservative of the two byte counts contained in
                            05E9  1248 ;       the MBA$L_BCR register.  The high word of this register is the
                            05E9  1249 ;       (negative of the) number of bytes transferred to or from the
                            05E9  1250 ;       drive, while the low word is the (negative of the) number of
                            05E9  1251 ;       bytes transferred to or memory.  On a read, the more conservative
                            05E9  1252 ;       value is that of the number of bytes transferred to memory (low word)
                            05E9  1253 ;       while on a write the more conservative value is the number of
                            05E9  1254 ;       bytes transferred to the drive (high word).  Here we deposit
                            05E9  1255 ;       the entire register into a longword in the UCB.  If the operation
                            05E9  1256 ;       was a read we leave the value as is.  However if the operation
                            05E9  1257 ;       was a write (or anything but a read) we move the high word to
                            05E9  1258 ;       the low word in memory.  All other pieces of this driver use the
                            05E9  1259 ;       low word of this longword as the valid byte count.
                            05E9  1260
                      10 A4 D0  05E9  1261         MOVL    MBA$L_BCR(R4),-       ; Save entire byte count register
                   00C0 C5      05EC  1262                 UCB$L_DB_BCR(R5)      ;  in the UCB.
                50    58 A5 D0  05EF  1263         MOVL    UCB$L_IRP(R5),R0      ; Retrieve IRP pointer.
                         01 E0  05F3  1264         BBS     #IRP$V_FUNC,-         ; If we had a read operation then
                   07 2A A0     05F5  1265                 IRP$W_STS(R0),20$     ;  just branch around since all OK.
                   00C2 C5 B0   05F8  1266         MOVW    UCB$L_DB_BCR+2(R5),-  ; If NOT read, then copy high word to
                   00C0 C5      05FC  1267                 UCB$W_BCR(R5)         ;  low order word for later use.
                            05FF  1268 20$:
                50    04 A3 D0  05FF  1269         MOVL    RP_DS(R3),R0          ;GET CONTENTS OF DRIVE STATUS REGISTER
                            0603  1270 ERROR:                                  ;
             51    00CE C5 D0  0603  1271         MOVL    UCB$L_DB_SR(R5),R1    ;RETRIEVE FINAL CONTROLLER STATUS
                52    08 A3 D0  0608  1272         MOVL    RP_ERT(R3),R2         ;GET CONTENTS OF DRIVE ERROR REGISTER 1
          64 A5    0060 8F B3  060C  1273         BITW    #UCB$M_POWER!-        ;POWER FAIL OR DEVICE TIMEOUT?
                            0612  1274                 UCB$M_TIMOUT,UCB$W_STS(R5) ;
                         BD 12  0612  1275         BNEQ    10$                   ;IF NEQ YES - SPECIAL CONDITION
                0093 C5  0A 91  0614  1276         CMPB    #CDF_WRITECHECK,UCB$B_CEX(R5) ;DRIVE RELATED FUNCTION?
                         2E 1A  0619  1277         BGTRU   30$                   ;IF GTRU YES
                0093 C5  10 91  061B  1278         CMPB    #CDF_READPRESET,UCB$B_CEX(R5) ;OTHER DRIVE RELATED FUNCTION?
                         27 1B  0620  1279         BLEQU   30$                   ;IF EQL YES
                            0622  1280
                            0622  1281 ;
                            0622  1282 ; CONTROLLER RELATED FUNCTION
                            0622  1283 ;
                            0622  1284
          51    000E5FFF 8F   D3  0622  1285         BITL    #MBA$M_ERROR,R1       ;ANY CONTROLLER ERRORS?
```

```
                 6F   13   0629  1286          BEQL    80$                     ;IF EQL NO
      00000000'  GF   16   062B  1287          JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
  6C  009A  C5   0F   E0   0631  1288          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
  51  0008000B  8F   D3   0637  1289          BITL    #MBA$M_SR_ERCONF!-      ;ERROR CONFIRMATION OR,
                              063E  1290                  MBA$M_SR_ISTO!-         ;INTERFACE SEQUENCE TIMEOUT OR,
                              063E  1291                  MBA$M_SR_PGE!-          ;PROGRAMMING ERROR OR,
                              063E  1292                  MBA$M_SR_RDTO,R1        ;READ TIMEOUT?
                 63   12   063E  1293          BNEQ    90$                     ;IF NEQ YES - FATAL CONTROLLER ERROR
  51  00064FF4  8F   D3   0640  1294          BITL    #MBA$M_SR_DLT!-         ;DATA LATE OR,
                              0647  1295                  MBA$M_SR_INVMAP!-       ;INVALID MAP REGISTER OR,
                              0647  1296                  MBA$M_SR_MAPPE!-        ;MAP REGISTER PARITY ERROR OR,
                              0647  1297                  MBA$M_SR_MBEXC!-        ;MASSBUS EXCEPTION OR,
                              0647  1298                  MBA$M_SR_MCPE!-         ;MASSBUS CONTROL PARITY ERROR OR,
                              0647  1299                  MBA$M_SR_SPE!-          ;SILO PARITY ERROR OR,
                              0647  1300                  MBA$M_SR_MDPE!-         ;MASSBUS DATA PARITY ERROR OR,
                              0647  1301                  MBA$M_SR_MXF!-          ;MISSED TRANSFER OR,
                              0647  1302                  MBA$M_SR_NED!-          ;NONEXISTENT DRIVE OR,
                              0647  1303                  MBA$M_SR_RDS!-          ;READ DATA SUBSTITUTE OR,
                              0647  1304                  MBA$M_SR_WCKLWR!-       ;WRITE CHECK LOWER BYTE OR,
                              0647  1305                  MBA$M_SR_WCKUPR,R1      ;WRITE CHECK UPPER BYTE?
                 31   12   0647  1306          BNEQ    60$                     ;IF NEQ YES - RETRIABLE CONTROLLER ERROR
                              0649  1307
                              0649  1308 ;
                              0649  1309 ; DRIVE RELATED FUNCTION
                              0649  1310 ;
                              0649  1311
                 08   91   0649  1312 30$:    CMPB    #CDF_PACKACK,-          ; Packack function?
            0093  C5         064B  1313                  UCB$B_CEX(R5)           ;
                 0F   12   064E  1314          BNEQ    40$                     ; Branch if not.
        0B  50   0C   E0   0650  1315          BBS     #RP_DS_V_MOL,R0,40$     ; Success if medium online.
                 0F   E1   0654  1316          BBC     #IO$V_INHRETRY,-        ; Branch if retrys not inhibited.
        28  009A C5         0656  1317                  UCB$W_FUNC(R5),65$      ;
                 0B   E5   065A  1318          BBCC    #UCB$V_VALID,-          ; Otherwise, clear software volume
          44 64  A5         065C  1319                  UCB$W_STS(R5),90$       ;  valid and take fatal error path.
      37  50   0E   E1   065F  1320 40$:    BBC     #RP_DS_V_ERR,R0,80$     ;IF CLR, NO DRIVE ERRORS
00C0  C5    7E   A5   AE   0663  1321 50$:    MNEGW   UCB$W_BCNT(R5),UCB$W_BCR(R5) ;RESET BYTE COUNT - NO TRANSFER
      35  00D2 C5   E8   0669  1322          BLBS    UCB$B_DB_ERL(R5),90$    ; Do NOT log error if medium was offline
                              066E  1323                                          ;  at start of function.
      00000000'  GF   16   066E  1324          JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
29  009A  C5   0F   E0   0674  1325          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
      25  50   0C   E1   067A  1326 60$:    BBC     #RP_DS_V_MOL,R0,90$     ;IF CLR, MEDIUM OFFLINE
      21  50   06   E1   067E  1327          BBC     #RP_DS_V_VV,R0,90$      ;IF CLR, INVALID VOLUME
  52    0E07 8F   B3   0682  1328 65$:    BITW    #RP_ER1_M_AOE!-         ;ADDRESS OVERFLOW OR,
                              0687  1329                  RP_ER1_M_IAE!-          ;INVALID ADDRESS OR,
                              0687  1330                  RP_ER1_M_ILF!-          ;ILLEGAL FUNCTION OR,
                              0687  1331                  RP_ER1_M_ILR!-          ;ILLEGAL REGISTER OR,
                              0687  1332                  RP_ER1_M_RMR!-          ;REGISTER MODIFY REFUSE OR,
                              0687  1333                  RP_ER1_M_WLE,R2         ;WRITE LOCK ERROR?
                 1A   12   0687  1334          BNEQ    90$                     ;IF NEQ YES - FATAL DRIVE ERROR
  52    4000 8F   B3   0689  1335          BITW    #RP_ER1_M_UNS,R2        ; Is the drive unsafe?
                 16   12   068E  1336          BNEQ    100$                    ; Branch if so.
                              0690  1337
                              0690  1338 ;
                              0690  1339 ; RETRIABLE ERROR EXIT
                              0690  1340 ;
                              0690  1341
  7E    009C D5   32   0690  1342 70$:    CVTWL   @UCB$L_DPC(R5),-(SP)    ;GET BRANCH DISPLACEMENT
```

```
                009C C5   8E   C0  0695  1343          ADDL    (SP)+,UCB$L_DPC(R5)     ;CALCULATE RETURN ADDRESS - 2
                009C C5   02   C0  069A  1344 80$:     ADDL    #2,UCB$L_DPC(R5)        ;SKIP PAST BRANCH DISPLACEMENT WORD
                     009C D5   17  069F  1345          JMP     @UCB$L_DPC(R5)          ;RETURN TO DRIVER
                               06A3  1346
                               06A3  1347 ;
                               06A3  1348 ; FATAL CONTROLLER OR DRIVE ERROR
                               06A3  1349 ;
                               06A3  1350
                         FD45  31   06A3  1351 90$:     BRW     FATALERR                ;
                               06A6  1352
                               06A6  1353 ;
                               06A6  1354 ; Check for unsafe condition and attempt to clear it.
                               06A6  1355 ;
                               06A6  1356
                               06A6  1357 100$:   DSBINT                              ; Disable interupts.
                            05 E1    06AC  1358          BBC     #UCB$V_POWER,-          ; Branch if no power failure occured.
                03 64 A5                06AE  1359                  UCB$W_STS(R5),110$  ;
                         FF20  31   06B1  1360          BRW     ENBXIT                  ; Otherwise, enable interupts and
                               06B4  1361                                             ;  go process error.
                         63 09 9A   06B4  1362 110$:    MOVZBL  #F_DRVCLR!1,RP_CS1(R3)  ; Attempt to clear unsafe condition.
                               06B7  1363          TIMEWAIT -                         ; Wait for ten microseconds or until
                               06B7  1364                  TIME = #1,-                ;  unsafe condition clears.
                               06B7  1365                  BITVAL = #RP_ER1_M_UNS,-;
                               06B7  1366                  SOURCE = RP_ER1(R3),-      ;
                               06B7  1367                  CONTEXT = L,-              ;
                               06B7  1368                  SENSE = .FALSE.            ;
                               06DF  1369          ENBINT                             ; Enable interupts.
                      52 08 A3 D0   06E2  1370          MOVL    RP_ER1(R3),R2           ; Retrieve error status.
                         A7 50 E8   06E6  1371          BLBS    R0,70$                  ; Branch if drive is no longer unsafe.
                            B8 11   06E9  1372          BRB     90$                     ; Otherwise, fatal error.
                               06EB  1373
                               06EB  1374 ;
                               06EB  1375 ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
                               06EB  1376 ;
                               06EB  1377
                61 64 A5   05  E4   06EB  1378 120$:   BBSC    #UCB$V_POWER,UCB$W_STS(R5),150$ ;IF SET, POWER FAILURE
                               06F0  1379
                               06F0  1380 ;
                               06F0  1381 ; DEVICE TIME OUT
                               06F0  1382 ;
                               06F0  1383
                 00000000'GF   16   06F0  1384          JSB     G^ERL$DEVICTMO          ;LOG DEVICE TIME OUT
                   53    24 A5   D0   06F6  1385          MOVL    UCB$L_CRB(R5),R3        ;GET ADDRESS OF CRB
                   53    2C A3   D0   06FA  1386          MOVL    CRB$L_INTD+VEC$L_IDB(R3),R3 ;GET ADDRESS OF IDB
                   04 A3    55   D1   06FE  1387          CMPL    R5,IDB$L_OWNER(R3)      ;DEVICE OWN CONTROLLER?
                            22   12   0702  1388          BNEQ    140$                    ;IF NEQ NO
                               0704  1389          DSBINT                          ;DISABLE INTERRUPTS
                            06   D0   070A  1390          MOVL    #MBA$M_CR_ABORT!MBA$M_CR_IE,- ;ABORT THE DATA TRANSFER
                         04 A4        070C  1391                  MBA$L_CR(R4)            ;
                               070E  1392          WFIKPCH 130$,#15                        ;WAIT FOR ABORT AND KEEP CHANNEL
                               0718  1393          IOFORK                          ;CREATE FORK PROCESS
                               071E  1394 130$:
                   04 A4    01   D0   071E  1395          MOVL    #MBA$M_CR_INIT,MBA$L_CR(R4) ;INITIALIZE ENTIRE MBA
                   04 A4    04   D0   0722  1396          MOVL    #MBA$M_CR_IE,MBA$L_CR(R4) ;ENABLE DEVICE INTERRUPTS
                               0726  1397 140$:   SETIPL  UCB$B_FIPL(R5)          ;LOWER TO FORK LEVEL
                50    022C 8F   3C   072A  1398          MOVZWL  #SS$_TIMEOUT,R0         ;SET DEVICE TIMEOUT STATUS
                      0080 C5   97   072F  1399          DECB    UCB$B_ERTCNT(R5)        ;ANY ERROR RETRIES REMAINING?
```

```
                     0F   13  0733  1400          BEQL    RESETXFR                        ;IF EQL NO
                                0735  1401          RELCHAN                                 ;RELEASE CHANNEL IF OWNED
          64 A5  0040 8F   AA  073B  1402          BICW    #UCB$M_TIMOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
                    FA68   31  0741  1403          BRW     FDISPATCH                       ;
                                0744  1404
                                0744  1405 ;
                                0744  1406 ; RESET TRANSFER BYTE COUNT TO ZERO
                                0744  1407 ;
                                0744  1408
                                0744  1409 RESETXFR:                                      ;
               53   58 A5   D0  0744  1410          MOVL    UCB$L_IRP(R5),R3                ;RETRIEVE ADDRESS OF I/O PACKET
          00C0 C5   32 A3   AE  0748  1411          MNEGW   IRP$W_BCNT(R3),UCB$W_BCR(R5) ;RESET TRANSFER BYTE COUNT
                    FD2E   31  074E  1412          BRW     FUNCXT                          ;
                                0751  1413
                                0751  1414 ;
                                0751  1415 ; POWER FAILURE
                                0751  1416 ;
                                0751  1417
                                0751  1418 150$:   RELCHAN                                 ;RELEASE CHANNEL
               53   58 A5   D0  0757  1419          MOVL    UCB$L_IRP(R5),R3                ;RETRIEVE ADDRESS OF I/O PACKET
             78 A5  2C A3   7D  075B  1420          MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
                    F9D4   31  0760  1421          BRW     DB_STARTIO                      ;
                                0763  1422          .DSABL  LSB
```

```
                     0763  1424          .SBTTL  RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                     0763  1425 ;
                     0763  1426 ; DB_DTYPE - RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                     0763  1427 ;
                     0763  1428 ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
                     0763  1429 ; SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
                     0763  1430 ; SET UNIT PARAMETERS.
                     0763  1431 ;
                     0763  1432 ; INPUTS:
                     0763  1433 ;
                     0763  1434 ;       R3 = ADDRESS OF DRIVE CONTROL REGISTER.
                     0763  1435 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                     0763  1436 ;       R5 = DEVICE UNIT UCB ADDRESS.
                     0763  1437 ;
                     0763  1438 ; OUTPUTS:
                     0763  1439 ;
                     0763  1440 ;       THE DRIVE STATUS REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
                     0763  1441 ;
                     0763  1442
                     0763  1443 DB_DTYPE:                                    ;CLASSIFY DRIVE TYPE AND SET PARAMETERS
       18 A3   DD    0763  1444          PUSHL   RP_DT(R3)                   ;READ DRIVE TYPE REGISTER
6E FE00 8F   AA     0766  1445          BICW    #^C<^X1FF>,(SP)             ;CLEAR EXTRANEOUS BITS
52 F8CD CF   9E     076B  1446          MOVAB   DB_DTDESC,R2                ;GET ADDRESS OF DESCRIPTOR TABLE
    82   6E   B1    0770  1447 10$:     CMPW    (SP),(R2)+                  ;DRIVE TYPE MATCH?
         0E   13    0773  1448          BEQL    20$                         ;IF EQL YES
    52   0D   C0    0775  1449          ADDL    #DB_DTDESCLEN-2,R2          ;ADVANCE TO NEXT ENTRY
         62   B5    0778  1450          TSTW    (R2)                        ;END OF TABLE?
         F4   12    077A  1451          BNEQ    10$                         ;IF NEQ NO
  64 A5  10   AA    077C  1452          BICW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT OFFLINE
    52   0D   C2    0780  1453          SUBL    #DB_DTDESCLEN-2,R2          ;BACK UP TO LAST DRIVE DESCRIPTOR
  41 A5  82   90    0783  1454 20$:     MOVB    (R2)+,UCB$B_DEVTYPE(R5)     ;SET DEVICE TYPE
  44 A5  82   D0    0787  1455          MOVL    (R2)+,UCB$L_DEVDEPEND(R5)   ;SET DISK PACK GEOMETRY
00B0 C5  82   D0    078B  1456          MOVL    (R2)+,UCB$L_MAXBLOCK(R5)    ;SET MAXIMUM BLOCKS PER PACK
008C C5  62   D0    0790  1457          MOVL    (R2),UCB$L_MEDIA_ID(R5)     ;SET MEDIA IDENTIFICATION
         8E   D5    0795  1458          TSTL    (SP)+                       ;REMOVE DRIVE TYPE FROM STACK
              05    0797  1459          RSB                                 ;
```

```
                                0798  1461          .SBTTL  RP04/05/06 REGISTER DUMP ROUTINE
                                0798  1462 ;
                                0798  1463 ; DB_REGDUMP - RP04/05/06 REGISTER DUMP ROUTINE
                                0798  1464 ;
                                0798  1465 ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                                0798  1466 ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
                                0798  1467 ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                                0798  1468 ;
                                0798  1469 ; INPUTS:
                                0798  1470 ;
                                0798  1471 ;       R0 = ADDRESS OF REGISTER SAVE BUFFER.
                                0798  1472 ;       R4 = ADDRESS OF ADAPTER CONFIGURATION REGISTER.
                                0798  1473 ;       R5 = DEVICE UNIT UCB ADDRESS.
                                0798  1474 ;
                                0798  1475 ; OUTPUTS:
                                0798  1476 ;
                                0798  1477 ;       THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                                0798  1478 ;
                                0798  1479
                                0798  1480 DB_REGDUMP:                                    ;RP04/05/06 REGISTER DUMP ROUTINE
                    80    17    D0  0798  1481          MOVL    #<RP_EC2+4+MBA$L_BCR+4+8>/4,(R0)+ ;INSERT NUMBER OF DEVICE REGISTERS
                    80    64    D0  079B  1482          MOVL    MBA$L_CSR(R4),(R0)+         ;SAVE CONFIGURATION REGISTER
                 80     04  A4    D0  079E  1483          MOVL    MBA$L_CR(R4),(R0)+          ;SAVE CONTROL REGISTER
              80    00CE  C5    D0  07A2  1484          MOVL    UCB$L_DB_SR(R5),(R0)+       ;SAVE STATUS REGISTER
                 80     0C  A4    D0  07A7  1485          MOVL    MBA$L_VAR(R4),(R0)+         ;SAVE VIRTUAL ADDRESS REGISTER
                 80     10  A4    D0  07AB  1486          MOVL    MBA$L_BCR(R4),(R0)+         ;SAVE BYTE COUNT REGISTER
        51    F8  A0     08    09    EF  07AF  1487          EXTZV   #9,#8,-8(R0),R1             ;GET FINAL MAP REGISTER NUMBER
           80    0800  C441    D0  07B5  1488          MOVL    MBA$L_MAP(R4)[R1],(R0)+     ;SAVE FINAL MAP REGISTER CONTENTS
                    80    D4  07BB  1489          CLRL    (R0)+                       ;ASSUME NO PREVIOUS MAP REGISTER
                    51    D7  07BD  1490          DECL    R1                          ;CALCULATE PREVIOUS MAP REGISTER NUMBER
                    07    19  07BF  1491          BLSS    10$                         ;IF LSS NONE
     FC  A0    0800  C441    D0  07C1  1492          MOVL    MBA$L_MAP(R4)[R1],-4(R0)    ;SAVE PREVIOUS MAP REGISTER CONTENTS
                 51     10    9A  07C8  1493 10$:     MOVZBL  #<RP_EC2+4>/4,R1            ;SET NUMBER OF DRIVE REGISTERS TO SAVE
        52     0091  C5    9A  07CB  1494          MOVZBL  UCB$B_SLAVE+1(R5),R2        ;GET DRIVE OFFSET CONSTANT
     52    0400  C442    DE  07D0  1495          MOVAL   MBA$L_ERB(R4)[R2],R2        ;GET ADDRESS OF DRIVE REGISTERS
              80    82    D0  07D6  1496 20$:     MOVL    (R2)+,(R0)+                 ;SAVE DRIVE REGISTER
                 FA  51    F5  07D9  1497          SOBGTR  R1,20$                      ;ANY MORE TO SAVE?
                             05  07DC  1498          RSB                             ;
```

```
                        07DD  1500          .SBTTL  RP04/RP05/RP06 DISK DRIVE INITIALIZATION
                        07DD  1501 ;
                        07DD  1502 ; DB_RP0X_INIT - RP04/RP05/RP06 DISK DRIVE INITIALIZATION
                        07DD  1503 ;
                        07DD  1504 ; THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
                        07DD  1505 ; DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
                        07DD  1506 ;
                        07DD  1507 ; INPUTS:
                        07DD  1508 ;
                        07DD  1509 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                        07DD  1510 ;       R5 = DEVICE UNIT UCB ADDRESS.
                        07DD  1511 ;
                        07DD  1512 ; OUTPUTS:
                        07DD  1513 ;
                        07DD  1514 ;       UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
                        07DD  1515 ;
                        07DD  1516
                        07DD  1517 DB_RP0X_INIT:                            ;RP04/RP05/RP06 DISK DRIVE INITIALIZATION
          53   54 A5   3C  07DD  1518          MOVZWL  UCB$W_UNIT(R5),R3        ;GET DRIVE UNIT NUMBER
       0090 C5    53   90  07E1  1519          MOVB    R3,UCB$B_SLAVE(R5)       ;SET SLAVE UNIT NUMBER
            53    20   C4  07E6  1520          MULL    #<1@7>/4,R3              ;CALCULATE DRIVE OFFSET CONSTANT
       0091 C5    53   90  07E9  1521          MOVB    R3,UCB$B_SLAVE+1(R5)     ;SET DRIVE OFFSET CONSTANT
    53   0400 C443   DE  07EE  1522          MOVAL   MBA$L_ERB(R4)[R3],R3     ;GET ADDRESS OF DRIVE CONTROL REGISTER
                03   BB  07F4  1523          PUSHR   #^M<R0,R1>               ;SAVE THESE REGISTERS
                        07F6  1524          TIMEWAIT #100,#RP_DS_M_DPR,-     ;TRY TO SEIZE DRIVE
                        07F6  1525                   RP_DS(R3),L
              10 50   E9  0822  1526          BLBC    R0,5$                    ;NO PORT SEIZED
          50    18 A3   D0  0825  1527          MOVL    RP_DT(R3),R0             ;GET DRIVE TYPE
          08 50    0B   E1  0829  1528          BBC     #RP_DT_V_DRQ,R0,5$       ;IF CLEAR, LEAVE
                02   88  082D  1529          BISB    #ER[_M_DUALPORT,-        ; SET FLAG WHICH INDICATES THAT DISK
                00D2 C5      082F  1530                  UCB$B_DB_ERL(R5)         ;  HAS DUAL PORT OPTION
                63   09  9A  0832  1531          MOVZBL  #F_DRVCLR!1,RP_CS1(R3)   ;CLEAR DRIVE
                03   BA  0835  1532 5$:       POPR    #^M<R0,R1>               ;RESTORE REGISTERS
          7E    64   A5  3C  0837  1533          MOVZWL  UCB$W_STS(R5),-(SP)      ;SAVE CURRENT UNIT STATUS
                08   A4  DD  083B  1534          PUSHL   MBA$L_SR(R4)             ;READ MBA STATUS REGISTER
    64 A5     0810 8F  AA  083E  1535          BICW    #UCB$M_ONLINE!UCB$M_VALID,UCB$W_STS(R5) ;SET UNIT OFFLINE/INVALID
          32  6E    12  E0  0844  1536          BBS     #MBA$V_SR_NED,(SP),40$   ;IF SET, NONEXISTENT DISK
          64  A5    10  A8  0848  1537          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
                FF14    30  084C  1538          BSBW    DB_DTYPE                 ;CLASSIFY DRIVE TYPE
       23 64  A5    04  E1  084F  1539          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),30$ ;IF CLR, UNKNOWN DRIVE TYPE
       1E 04  AE    0B  E1  0854  1540          BBC     #UCB$V_VALID,4(SP),30$   ;IF CLR, VOLUME SOFTWARE INVALID
                63   09  9A  0859  1541 10$:     MOVZBL  #F_DRVCLR!1,RP_CS1(R3)   ;CLEAR DRIVE
    52    04 A3    13  78  085C  1542          ASHL    #3T-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
                     0B  19  0861  1543          BLSS    20$                      ;IF LSS YES
             00000000'GF  16  0863  1544          JSB     G^EXE$PWRTIMCHK          ;CHECK FOR MAXIMUM TIME EXCEEDED
                   ED 50  E8  0869  1545          BLBS    R0,10$                   ;IF LBS MORE TIME TO GO
                     09  11  086C  1546          BRB     30$                      ;
                63   13  9A  086E  1547 20$:     MOVZBL  #F_PACKACK!1,RP_CS1(R3)  ;ACKNOWLEDGE PACK
    64 A5     0800 8F  A8  0871  1548          BISW    #UCB$M_VALID,UCB$W_STS(R5) ;SET VOLUME SOFTWARE VALID
                63   0B  9A  0877  1549 30$:     MOVZBL  #F_RELEASE!1,RP_CST(R3)  ;Clear drive and release port
    08 A4       8E   8E  C9  087A  1550 40$:     BISL3   (SP)+,(SP)+,MBA$L_SR(R4) ;CLEAR MBA STATUS
                         05  087F  1551          RSB                              ;
```

```
                     0880  1553          .SBTTL  RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
                     0880  1554 ;
                     0880  1555 ; DB_UNSOLNT - RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
                     0880  1556 ;
                     0880  1557 ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED FOR
                     0880  1558 ; AN RP04, RP05, OR RP06 DRIVE.
                     0880  1559 ;
                     0880  1560 ; INPUTS:
                     0880  1561 ;
                     0880  1562 ;       R4 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                     0880  1563 ;       R5 = DEVICE UNIT UCB ADDRESS.
                     0880  1564 ;
                     0880  1565 ; OUTPUTS:
                     0880  1566 ;
                     0880  1567 ;       IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
                     0880  1568 ;       UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
                     0880  1569 ;       CLASSIFIED.
                     0880  1570 ;
                     0880  1571
                     0880  1572 DB_UNSOLNT:                                 ;RP04/RP05/RP06 UNSOLICITED INTERRUPTS
   53   0091 C5  9A  0880  1573          MOVZBL  UCB$B_SLAVE+1(R5),R3        ;GET DRIVE OFFSET CONSTANT
53   0400 C443  DE  0885  1574          MOVAL   MBA$L_ERB(R4)[R3],R3        ;GET ADDRESS OF DRIVE CONTROL REGISTER
     64 A5   10  A8  088B  1575          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
         FED1    30  088F  1576          BSBW    DB_DTYPE                    ;CLASSIFY DRIVE TYPE
  1F 64 A5   04  E1  0892  1577          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),10$ ;IF CLR, UNKNOWN DRIVE TYPE
  20 64 A5   0B  E1  0897  1578          BBC     #UCB$V_VALID,UCB$W_STS(R5),20$ ;IF CLR, VOLUME SOFTWARE INVALID
52   04 A3   13  78  089C  1579          ASHL    #31-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
             13  18  08A1  1580          BGEQ    10$                         ;IF GEQ NO
  07 64 A5   08  E1  08A3  1581          BBC     #UCB$V_BSY,UCB$W_STS(R5),5$ ;We know the drive is online; thus,
   0093 C5   08  91  08A8  1582          CMPB    #CDF_PACKACK,UCB$B_CEX(R5)  ;if busy doing a PACKACK function,
             0D  13  08AD  1583          BEQL    20$                         ;then don't clear software valid.
52   04 A3   19  78  08AF  1584 5$:      ASHL    #31-RP_DS_V_VV,RP_DS(R3),R2 ;VOLUME VALID?
             06  19  08B4  1585          BLSS    20$                         ;IF LSS YES
64 A5   0800 8F  AA  08B6  1586 10$:     BICW    #UCB$M_VALID,UCB$W_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
             05  08BC  1587 20$:     RSB                                     ;
                 08BD  1588 DB_END:                                          ;ADDRESS OF LAST LOCATION IN DRIVER
                 08BD  1589
                 08BD  1590          .END
```

```
$$$                     = 00000020 R      02
$$OP                    = 00000002
ACP$ACCESS                ********    X   03
ACP$DEACCESS              ********    X   03
ACP$MODIFY                ********    X   03
ACP$MOUNT                 ********    X   03
ACP$READBLK               ********    X   03
ACP$WRITEBLK              ********    X   03
APPLY_ECC                 000002FE  R       03
AT$_MBA                 = 00000000
CDF_DRVCLR              = 00000004
CDF_NOP                 = 00000005
CDF_OFFSET              = 00000006
CDF_PACKACK             = 00000008
CDF_READDATA            = 0000000C
CDF_READHEAD            = 0000000E
CDF_READPRESET          = 00000010
CDF_RECAL               = 00000003
CDF_RETCENTER           = 00000007
CDF_SEARCH              = 00000009
CDF_SEARCHA             = 00000011
CDF_SEEK                = 00000002
CDF_UNLOAD              = 00000001
CDF_WRITECHECK          = 0000000A
CDF_WRITECHECKH         = 0000000F
CDF_WRITEDATA           = 0000000B
CDF_WRITEHEAD           = 0000000D
CHECKRETRY                0000026D  R       03
CHECKTAB                  00000038  R       03
CHECKXT                   00000286  R       03
CRB$L_INTD              = 00000024
DATACHECK                 0000023A  R       03
DB$DDT                    00000000  RG      03
DB_DTDESC                 0000003C  R       03
DB_DTDESCLEN            = 0000000F
DB_DTYPE                  00000763  R       03
DB_END                    000008BD  R       03
DB_FUNCTABLE              000000A3  R       03
DB_REGDUMP                00000798  R       03
DB_RP0X_INIT              000007DD  R       03
DB_STARTIO                00000137  R       03
DB_UNSOLNT                00000880  R       03
DC$_DISK                = 00000001
DDB$K_PACK              = 00000001
DDB$L_ACPD              = 00000010
DDB$L_DDT               = 0000000C
DEFER_ECC                 00000327  R       03
DEV$M_AVL               = 00040000
DEV$M_DIR               = 00000008
DEV$M_DUA               = 00008000
DEV$M_ELG               = 00400000
DEV$M_FOD               = 00004000
DEV$M_IDV               = 04000000
DEV$M_NNM               = 00000200
DEV$M_ODV               = 08000000
DEV$M_RND               = 10000000
DEV$M_SHR               = 00010000
DPT$C_LENGTH            = 00000038
DPT$C_VERSION           = 00000004
DPT$INITAB                00000038  R       02
DPT$M_SVP               = 00000002
DPT$REINITAB              0000006A  R       02
DPT$TAB                   00000000  R       02
DRVCLR                    00000209  R       03
DT$_RP04                = 00000003
DT$_RP05                = 00000004
DT$_RP06                = 00000005
DYN$C_DDB               = 00000006
DYN$C_DPT               = 0000001E
DYN$C_UCB               = 00000010
ECC                       000002AE  R       03
EMB$L_DV_REGSAV         = 0000004E
ENBXIT                    000005D4  R       03
ERL$DEVICERR              ********    X   03
ERL$DEVICTMO              ********    X   03
ERL_M_DUALPORT          = 00000002
ERL_M_ECC_DEFER         = 00000004
ERL_M_MEDOFF            = 00000001
ERL_V_DUALPORT          = 00000001
ERL_V_ECC_DEFER         = 00000002
ERROR                     00000603  R       03
EXE$GL_TENUSEC            ********    X   03
EXE$GL_UBDELAY            ********    X   03
EXE$IOFORK                ********    X   03
EXE$LCLDSKVALID           ********    X   03
EXE$ONEPARM               ********    X   03
EXE$PWRTIMCHK             ********    X   03
EXE$SENSEMODE             ********    X   03
EXE$SETCHAR               ********    X   03
EXE$ZEROPARM              ********    X   03
EXFNC                     00000581  R       03
FATALERR                  000003EB  R       03
FDISPATCH                 000001AC  R       03
FEX                       000004C3  R       03
FTAB                      00000089  R       03
FUNCTAB_LEN             = 00000094
FUNCXT                    0000047F  R       03
F_DRVCLR                = 00000008
F_NOP                   = 00000000
F_OFFSET                = 0000000C
F_PACKACK               = 00000012
F_READDATA              = 00000038
F_READHEAD              = 0000003A
F_READPRESET            = 00000010
F_RECAL                 = 00000006
F_RELEASE               = 0000000A
F_RETCENTER             = 0000000E
F_SEARCH                = 00000018
F_SEARCHA               = 00000018
F_SEEK                  = 00000004
F_UNLOAD                = 00000002
F_WRITECHECK            = 00000028
F_WRITECHECKH           = 0000002A
F_WRITEDATA             = 00000030
```

```
F_WRITEHEAD                     = 00000032
GO                                000004E2 R      03
IDB$L_OWNER                     = 00000004
IMMED                             0000050B R      03
IO$M_DATACHECK                  = 00004000
IO$V_DATACHECK                  = 0000000E
IO$V_INHRETRY                   = 0000000F
IO$V_INHSEEK                    = 0000000C
IO$_ACCESS                      = 00000032
IO$_ACPCONTROL                  = 00000038
IO$_AVAILABLE                   = 00000011
IO$_CREATE                      = 00000033
IO$_DEACCESS                    = 00000034
IO$_DELETE                      = 00000035
IO$_DRVCLR                      = 00000004
IO$_MODIFY                      = 00000036
IO$_MOUNT                       = 00000039
IO$_NOP                         = 00000000
IO$_OFFSET                      = 00000006
IO$_PACKACK                     = 00000008
IO$_READHEAD                    = 0000000E
IO$_READLBLK                    = 00000021
IO$_READPBLK                    = 0000000C
IO$_READPRESET                  = 00000019
IO$_READVBLK                    = 00000031
IO$_RECAL                       = 00000003
IO$_RELEASE                     = 00000005
IO$_RETCENTER                   = 00000007
IO$_SEARCH                      = 00000009
IO$_SEEK                        = 00000002
IO$_SENSECHAR                   = 0000001B
IO$_SENSEMODE                   = 00000027
IO$_SETCHAR                     = 0000001A
IO$_SETMODE                     = 00000023
IO$_UNLOAD                      = 00000001
IO$_VIRTUAL                     = 0000003F
IO$_WRITECHECK                  = 0000000A
IO$_WRITECHECKH                 = 00000018
IO$_WRITEHEAD                   = 0000000D
IO$_WRITELBLK                   = 00000020
IO$_WRITEPBLK                   = 0000000B
IO$_WRITEVBLK                   = 00000030
IOC$APPLYECC                      ********  X   03
IOC$DIAGBUFILL                    ********  X   03
IOC$LOADMBAMAP                    ********  X   03
IOC$MNTVER                        ********  X   03
IOC$RELCHAN                       ********  X   03
IOC$REQCOM                        ********  X   03
IOC$REQPCHANL                     ********  X   03
IOC$RETURN                        ********  X   03
IOC$UPDATRANSP                    ********  X   03
IOC$WFIKPCH                       ********  X   03
IRP$L_MEDIA                     = 00000038
IRP$L_SVAPTE                    = 0000002C
IRP$S_FCODE                     = 00000006
IRP$V_FCODE                     = 00000000
IRP$V_FUNC                      = 00000001
IRP$V_PHYSIO                    = 00000008
IRP$W_BCNT                      = 00000032
IRP$W_FUNC                      = 00000020
IRP$W_STS                       = 0000002A
LDCYL                             0000057B R      03
MASKH                           = 00000008
MASKL                           = 04000000
MBA$L_BCR                       = 00000010
MBA$L_CR                        = 00000004
MBA$L_CSR                       = 00000000
MBA$L_ERB                       = 00000400
MBA$L_MAP                       = 00000800
MBA$L_SR                        = 00000008
MBA$L_VAR                       = 0000000C
MBA$M_CR_ABORT                  = 00000002
MBA$M_CR_IE                     = 00000004
MBA$M_CR_INIT                   = 00000001
MBA$M_ERROR                     = 000E5FFF
MBA$M_SR_DLT                    = 00000800
MBA$M_SR_ERCONF                 = 00000008
MBA$M_SR_INVMAP                 = 00000010
MBA$M_SR_ISTO                   = 00000002
MBA$M_SR_MAPPE                  = 00000020
MBA$M_SR_MBEXC                  = 00000080
MBA$M_SR_MCPE                   = 00020000
MBA$M_SR_MDPE                   = 00000040
MBA$M_SR_MXF                    = 00000100
MBA$M_SR_NED                    = 00040000
MBA$M_SR_PGE                    = 00080000
MBA$M_SR_RDS                    = 00000004
MBA$M_SR_RDTO                   = 00000001
MBA$M_SR_SPE                    = 00004000
MBA$M_SR_WCKLWR                 = 00000200
MBA$M_SR_WCKUPR                 = 00000400
MBA$V_SR_NED                    = 00000012
NOP                               00000209 R      03
NORMAL                            00000283 R      03
OFF                               0000032C R      03
OFFSET                            00000209 R      03
OFFSETERR                         000003B3 R      03
OFFSIZ                          = 00000008
OFFTAB                            0000009B R      03
PACKACK                           00000203 R      03
POSIT                             00000575 R      03
PR$_IPL                         = 00000012
READDATA                          0000021C R      03
READHEAD                          0000021C R      03
READPRESET                        00000209 R      03
RECAL                             00000209 R      03
RELEASE                           00000209 R      03
RESETXFR                          00000744 R      03
RETCENTER                         00000209 R      03
RETREG                            000005D7 R      03
RETRY                             000002AB R      03
RETRYERR                          000003BE R      03
RP_AS                             00000010
RP_CC                             0000002C
```

```
RP_CS1                          00000000
RP_CS1_M_GO                   = 00000001
RP_DA                           00000014
RP_DC                           00000028
RP_DS                           00000004
RP_DS_M_DPR                   = 00000100
RP_DS_M_ERR                   = 00004000
RP_DS_V_ERR                   = 0000000E
RP_DS_V_MOL                   = 0000000C
RP_DS_V_VV                    = 00000006
RP_DT                           00000018
RP_DT_V_DRQ                   = 0000000B
RP_ECT                          00000038
RP_EC2                          0000003C
RP_ER1                          00000008
RP_ER1_M_AOE                  = 00000200
RP_ER1_M_DCK                  = 00008000
RP_ER1_M_DTE                  = 00001000
RP_ER1_M_ECH                  = 00000040
RP_ER1_M_FER                  = 00000010
RP_ER1_M_HCE                  = 00000080
RP_ER1_M_HCRC                 = 00000100
RP_ER1_M_IAE                  = 00000400
RP_ER1_M_ILF                  = 00000001
RP_ER1_M_ILR                  = 00000002
RP_ER1_M_OPI                  = 00002000
RP_ER1_M_PAR                  = 00000008
RP_ER1_M_RMR                  = 00000004
RP_ER1_M_UNS                  = 00004000
RP_ER1_M_WCF                  = 00000020
RP_ER1_M_WLE                  = 00000800
RP_ER1_V_FER                  = 00000004
RP_ER1_V_HCE                  = 00000007
RP_ER1_V_HCRC                 = 00000008
RP_ER1_V_OPI                  = 0000000D
RP_ER1_V_UNS                  = 0000000E
RP_ER1_V_WLE                  = 0000000B
RP_ER2                          00000020
RP_ER3                          00000034
RP_ER3_V_SKI                  = 0000000E
RP_LA                           0000001C
RP_MR                           0000000C
RP_OF                           00000024
RP_OF_M_DCK                   = 00000100
RP_OF_M_ECI                   = 00000800
RP_OF_M_FMT                   = 00001000
RP_OF_M_HCI                   = 00000400
RP_OF_V_DCK                   = 00000008
RP_OF_V_ECI                   = 0000000B
RP_SN                           00000030
SEARCH                          00000209 R       03
SEARCHA                         00000552 R       03
SEEK                            00000209 R       03
SEIZE                           00000522 R       03
SIZ...                        = 00000001
SS$_CTRLERR                   = 00000054
SS$_DATACHECK                 = 0000005C
SS$_DRVERR                    = 0000008C
SS$_FORMAT                    = 000000BC
SS$_IVADDR                    = 00000134
SS$_MEDOFL                    = 000001A4
SS$_NONEXDRV                  = 000001C4
SS$_NORMAL                    = 00000001
SS$_OPINCOMPL                 = 000002D4
SS$_PARITY                    = 000001F4
SS$_TIMEOUT                   = 0000022C
SS$_UNSAFE                    = 0000023C
SS$_VOLINV                    = 00000254
SS$_WASECC                    = 00000639
SS$_WRITLCK                   = 0000025C
TRANNOCH                        00000230 R       03
TRANRQCH                        0000022A R       03
TRANXT                          00000289 R       03
UCB$B_CEX                     = 00000093
UCB$B_DB_ERL                    000000D2
UCB$B_DEVCLASS                = 00000040
UCB$B_DEVTYPE                 = 00000041
UCB$B_DIPL                    = 0000005E
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$K_DB_LENGTH               = 000000D6
UCB$K_LCL_DISK_LENGTH         = 000000CC
UCB$L_BCR                     = 000000C0
UCB$L_CRB                     = 00000024
UCB$L_DB_BCR                  = 000000C0
UCB$L_DB_SR                     000000CE
UCB$L_DEVCHAR                 = 00000038
UCB$L_DEVCHAR2                = 0000003C
UCB$L_DEVDEPEND               = 00000044
UCB$L_DPC                     = 0000009C
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$W_BCR                     = 000000C0
UCB$W_DA                      = 000000BC
UCB$W_DB_ER3                    000000CC
UCB$W_DC                      = 000000BE
```

```
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_EC1                       = 000000C4
UCB$W_EC2                       = 000000C6
UCB$W_FUNC                      = 0000009A
UCB$W_OFFSET                    = 000000C8
UCB$W_STS                       = 00000064
UCB$W_UNIT                      = 00000054
UNLOAD                            000001FB R      03
VEC$L_IDB                       = 00000008
WRITECHECK                        00000210 R      03
WRITECHECKH                       00000210 R      03
WRITEDATA                         00000217 R      03
WRITEHEAD                         00000217 R      03
XFER                              00000566 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	000000D6 (214.)	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	000008BD (2237.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.06	00:00:00.29
Command processing	129	00:00:00.40	00:00:01.70
Pass 1	585	00:00:19.40	00:01:11.47
Symbol table sort	0	00:00:02.51	00:00:07.70
Pass 2	285	00:00:04.18	00:00:17.00
Symbol table output	43	00:00:00.21	00:00:00.77
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1078	00:00:26.77	00:01:38.94

The working set limit was 2100 pages.
151533 bytes (296 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2322 non-local and 65 local symbols.
1590 source lines were read in Pass 1, producing 22 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$:DBDRIVER/OBJ=OBJ$:DBDRIVER MSRC$:DBDRIVER/UPDATE=(ENH$:DBDRIVER)+EXECML$/LIB
```

DBDRIVER
LIS

CRDRIVER
LIS

CVDRIVER
LIS

DDDRIVER
LIS

DDDMP
LIS