```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      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**DRDRIVER

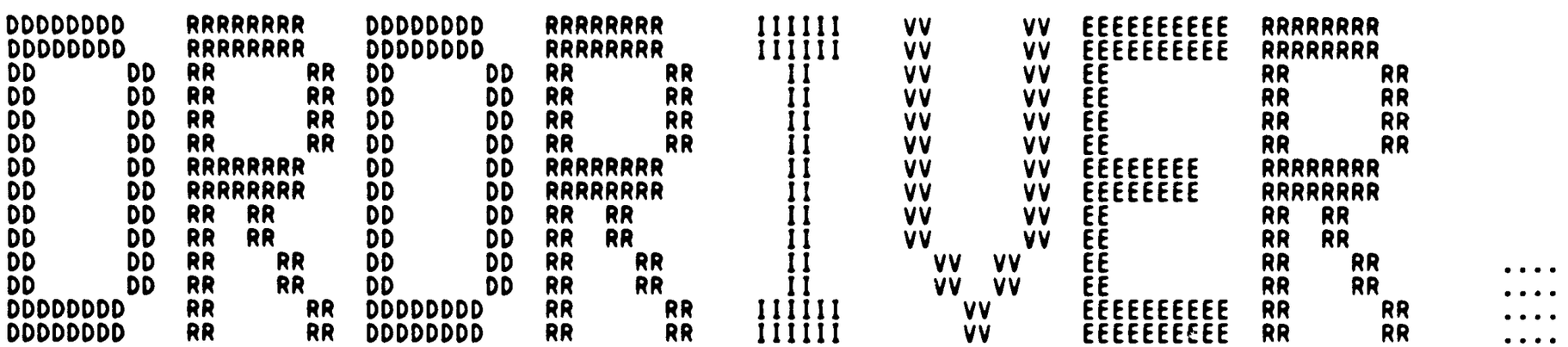

Table of contents

```
0000     1          .TITLE  DRDRIVER - RM03/RM05/RM80/RP07 DISK DRIVER
0000     2          .IDENT  'V04-001'
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, LEN KAWELL 23-NOV-77
0000    29 ;
0000    30 ;  MODIFIED BY:
0000    31 ;
0000    32 ;       V04-001 PRD0112         Paul R. DeStefano       06-Sep-1984
0000    33 ;               Modify ECC routine to allow for RP07's handling of
0000    34 ;               HCRC errors as class A errors when HCI is set.
0000    35 ;
0000    36 ;               Add sanity check to offset recovery routine to insure
0000    37 ;               that there is data to be transferred before offset
0000    38 ;               recovery is performed.
0000    39 ;
0000    40 ;       V03-016 RAS0300         Ron Schaefer            27-Apr-1984
0000    41 ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    42 ;               devices will have the "node$" prefix.
0000    43 ;
0000    44 ;       V03-015 PRD0081         Paul R. DeStefano       19-Mar-1984
0000    45 ;               For dual ported drives, make sure the port isn't
0000    46 ;               reseized by the time we come off the I/O fork queue.
0000    47 ;
0000    48 ;       V03-014 PRD0048         Paul R. DeStefano       01-Feb-1984
0000    49 ;               Fix context used in TIMEWAIT macro when referencing
0000    50 ;               device registers.
0000    51 ;
0000    52 ;       V03-013 PRD0036         Paul R. DeStefano       09-Sep-1983
0000    53 ;               Added EXE$LCLDSKVALID to function decision table.
0000    54 ;
0000    55 ;       V03-012 ROW0211         Ralph O. Weber          16-AUG-1983
0000    56 ;               Change device-dependent UCB definition base from UCB$W_BCR+2
0000    57 ;               to UCB$K_LCL_DISK_LENGTH.
```

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

```
0000    115          $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS
0000    116          $DPTDEF                         ;DEFINE DPT OFFSETS
0000    117          $DYNDEF                         ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000    118          $EMBDEF                         ;DEFINE EMB OFFSETS
0000    119          $IDBDEF                         ;DEFINE IDB OFFSETS
0000    120          $IODEF                          ;DEFINE I/O FUNCTION CODES
0000    121          $IRPDEF                         ;DEFINE IRP OFFSETS
0000    122          $MBADEF                         ;DEFINE MBA REGISTER OFFSETS
0000    123          $PRDEF                          ;DEFINE PROCESSOR REGISTER NUMBERS
0000    124          $SSDEF                          ;DEFINE SYSTEM STATUS CODES
0000    125          $UCBDEF                         ;DEFINE UCB OFFSETS
0000    126          $VECDEF                         ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000    127
0000    128 ;
0000    129 ; LOCAL MACROS
0000    130 ;
0000    131 ; EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000    132 ;
0000    133
0000    134          .MACRO  EXFUNC BDST,FCODE
0000    135                  .IF NB  FCODE
0000    136                  MOVZBL  #CD'FCODE,R0
0000    137                  .ENDC
0000    138                  BSBW    FEX
0000    139                  .SIGNED_WORD BDST-.-2
0000    140          .ENDM
0000    141
0000    142 ;
0000    143 ; GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000    144 ;
0000    145
0000    146          .MACRO  GENF FCODE
0000    147                  CD'FCODE=.-FTAB
0000    148                  .BYTE   FCODE!RM_CS1_M_GO
0000    149          .ENDM
0000    150
0000    151 ;
0000    152 ; LOCAL SYMBOLS
0000    153 ;
0000    154 ; MASSBUS REGISTER OFFSETS
0000    155 ;
0000    156
0000    157          $DEFINI RM
0000    158
0000    159 $DEF     RM_CS1              .BLKL   1           ;DRIVE CONTROL REGISTER
0004    160          _VIELD  RM_CS1,0,<-                     ; DRIVE CONTROL REGISTER BIT DEFINITIONS
0004    161                  <GO,,M>,-                      ; GO BIT
0004    162                  <FCODE,5>-                     ; FUNCTION CODE
0004    163                  >                              ;
0004    164 $DEF     RM_DS               .BLKL   1           ;DRIVE STATUS REGISTER
0008    165          _VIELD  RM_DS,0,<-                      ; DRIVE STATUS REGISTER BIT DEFINITIONS
0008    166                  <OM,,M>,-                      ; OFFSET MODE
0008    167                  <,5>,-                         ; RESERVED BITS
0008    168                  <VV,,M>,-                      ; VOLUME VALID
0008    169                  <DRY,,M>,-                     ; DRIVE READY
0008    170                  <DPR,,M>,-                     ; DRIVE PRESENT
0008    171                  <PGM,,M>,-                     ; PROGRAMMABLE
```

```
0008   172                  <LST,,M>,-              ; LAST SECTOR TRANSFERED
0008   173                  <WRL,,M>,-              ; DRIVE WRITE LOCKED
0008   174                  <MOL,,M>,-              ; MEDIUM ONLINE
0008   175                  <PIP,,M>,-              ; POSITIONING IN PROGRESS
0008   176                  <ERR,,M>,-              ; COMPOSITE ERROR
0008   177                  <ATA,,M>-               ; ATTENTION ACTIVE
0008   178          >                               ;
0008   179 $DEF     RM_ER1          .BLKL   1       ;ERROR REGISTER 1
000C   180          _VIELD  RM_ER1,0,<-             ; ERROR REGISTER 1 BIT DEFINITIONS
000C   181                  <ICF,,M>,-              ; ILLEGAL FUNCTION
000C   182                  <ILR,,M>,-              ; ILLEGAL REGISTER
000C   183                  <RMR,,M>,-              ; REGISTER MODIFY REFUSED
000C   184                  <PAR,,M>,-              ; PARITY ERROR
000C   185                  <FER,,M>,-              ; FORMAT ERROR
000C   186                  <WCF,,M>,-              ; WRITE CLOCK FAIL
000C   187                  <ECH,,M>,-              ; ECC HARD ERROR
000C   188                  <HCE,,M>,-              ; HEADER COMPARE ERROR
000C   189                  <HCRC,,M>,-             ; HEADER CRC ERROR
000C   190                  <AOE,,M>,-              ; ADDRESS OVERFLOW ERROR
000C   191                  <IAE,,M>,-              ; ILLEGAL ADDRESS ERROR
000C   192                  <WLE,,M>,-              ; WRITE LOCK ERROR
000C   193                  <DTE,,M>,-              ; DRIVE TIMING ERROR
000C   194                  <OPI,,M>,-              ; OPERATION INCOMPLETE
000C   195                  <UNS,,M>,-              ; DRIVE UNSAFE
000C   196                  <DCK,,M>-               ; DATA CHECK ERROR
000C   197          >                               ;
000C   198 $DEF     RM_MR           .BLKL   1       ;MAINTENANCE REGISTER
0010   199          _VIELD  RM_MR,0,<-              ; MAINTENANCE REGISTER DEFINITIONS
0010   200                  <PAR,8>,-               ; DIAGNOSTIC PARAMETER
0010   201                  <RTN,7>,-               ; DIAGNOSTIC ROUTINE NUMBER
0010   202                  <DM,,M>,-               ; DIAGNOSTIC MODE
0010   203          >                               ;
0010   204 $DEF     RM_AS           .BLKL   1       ;ATTENTION SUMMARY REGISTER
0014   205 $DEF     RM_DA           .BLKL   1       ;DESIRED SECTOR/TRACK ADDRESS REGISTER
0018   206          _VIELD  RM_DA,0,<-              ; DESIRED ADDRESS FIELD DEFINITIONS
0018   207                  <SA,5>,-                ; DESIRED SECTOR ADDRESS
0018   208                  <,3>,-                  ; RESERVED BITS
0018   209                  <TA,5>-                 ; DESIRED TRACK ADDRESS
0018   210          >                               ;
0018   211 $DEF     RM_DT           .BLKL   1       ;DRIVE TYPE REGISTER
001C   212          _VIELD  RM_DT,0,<-              ; DRIVE TYPE REGISTER FIELD DEFINITIONS
001C   213                  <DTN,9>,-               ; DRIVE TYPE NUMBER
001C   214                  <,2>,-                  ; RESERVED BITS
001C   215                  <DRQ,,M>-               ; DRIVE REQUEST REQUIRED
001C   216          >                               ;
001C   217 $DEF     RM_LA           .BLKL   1       ;LOOKAHEAD REGISTER
0020   218 $DEF     RM_SN           .BLKL   1       ;SERIAL NUMBER REGISTER
0024   219 $DEF     RM_OF           .BLKL   1       ;OFFSET REGISTER
0028   220          _VIELD  RM_OF,0,<-              ; OFFSET REGISTER BIT DEFINITIONS
0028   221                  <OFF,8>,-               ; OFFSET VALUE
0028   222                  <,1>,-                  ; RESERVED
0028   223                  <SSEI,,M>,-             ; SKIP SECTOR INHIBIT (RM80)
0028   224                  <HCI,,M>,-              ; HEADER COMPARE INHIBIT
0028   225                  <ECI,,M>,-              ; ECC INHIBIT (avoid using this bit)
0028   226                  <FMT,,M>,-              ; 16-BIT FORMAT
0028   227                  <,1>,-                  ; RESERVED
0028   228                  <MTD,,M>,-              ; MOVE TRACK DESCRIPTOR
```

```
                                                <CMO,,M>-                       ; COMMAND MODIFIER
                                                >                               ;
                                $DEF    RM_DC                   .BLKL   1       ;DESIRED CYLINDER ADDRESS
                                $DEF    RM_UNUSED               .BLKL   1       ;UNUSED
                                $DEF    RM_MR2                  .BLKL   1       ;MAINTENANCE REGISTER 2
                                $DEF    RM_ER2                  .BLKL   1       ;ERROR REGISTER 2
                                        _VIELD  RM_ER2,3,<-                     ; ERROR REGISTER 2 BIT DEFINITIONS
                                                <DPE,,M>,-                      ; DATA PARITY ERROR
                                                <,1>,-                          ; RESERVED BIT
                                                <SSE,,M>,-                      ; SKIP SECTOR ERROR (RM80)
                                                <,1>,-                          ; RESERVED BIT
                                                <DVC,,M>,-                      ; DEVICE CHECK ERROR
                                                <,2>,-                          ; RESERVED BITS
                                                <LBC,,M>,-                      ; LOSS OF BIT CLOCK ERROR
                                                <LSC,,M>,-                      ; LOSS OF SYSTEM CLOCK ERROR
                                                <IVC,,M>,-                      ; INVALID COMMAND ERROR
                                                <OPE,,M>,-                      ; OPERATOR PLUG ERROR
                                                <SKI,,M>,-                      ; SEEK INCOMPLETE ERROR
                                                <BSE,,M>-                       ; BAD SECTOR ERROR
                                                >                               ;
                                $DEF    RM_EC1                  .BLKL   1       ;ECC POSITION REGISTER
                                        _VIELD  RM_EC1,0,<<POS,13>>             ; ECC POSITION FIELD
                                $DEF    RM_EC2                  .BLKL   1       ;ECC PATTERN REGISTER
                                        _VIELD  RM_EC2,0,<<PAT,11>>             ; ECC PATTERN FIELD

                                        $DEFEND RM

                                ;
                                ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
                                ;

                                        $DEFINI UCB

000000CC                        .=UCB$K_LCL_DISK_LENGTH                         ;Establish device-dependent UCB base

                                $DEF    UCB$L_DR_SR             .BLKL   1       ;SAVED MBA STATUS REGISTER
                                $DEF    UCB$W_DR_ER2            .BLKW   1       ;SAVED ERROR REGISTER 2
                                $DEF    UCB$W_DR_MR             .BLKW   1       ;MAINTENANCE REGISTER
                                $DEF    UCB$B_DR_SSTS           .BLKB   1       ;SOFTWARE STATUS BYTE
                                        _VIELD  DR,0,<-                         ; SOFTWARE STATUS BIT DEFINITIONS
                                                <DCK,,M>,-                      ; DATACHECK IN PROGRESS
                                                <OM,,M>,-                       ; OFFSET MODE
                                                <NOECC,,M>,-                    ; Don't correct with ECC
                                                <DUALPORT,,M>-                  ; Drive has a dualport kit
                                                <ECC_DEFER,,M>, -               ; Flag to indicate that ECC correction
                                                >                               ;  has been deferred until offset
                                                                                ;  retries are exhausted.
                                $DEF    UCB$B_DR_ERL            .BLKB   1       ;ERROR LOGGING REGISTER FOR MED OFFLINE
                                $DEF    UCB$W_DR_OFR            .BLKW   1       ;SAVED OFFSET REGISTER
                                $DEF    UCB$L_DR_BCR            .BLKL   1       ; Saved (longword) MBA byte count reg.
000000DC                        UCB$K_DR_LENGTH=.

                                        $DEFEND UCB

                                ;
                                ; HARDWARE FUNCTION CODES
                                ;
```

```
                     0000   286
00000000             0000   287 F_NOP=0*2                          ;NO OPERATION
00000004             0000   288 F_SEEK=2*2                         ;SEEK CYLINDER
00000006             0000   289 F_RECAL=3*2                        ;RECALIBRATE
00000008             0000   290 F_DRVCLR=4*2                       ;DRIVE CLEAR
0000000A             0000   291 F_RELEASE=5*2                      ;RELEASE DRIVE
0000000C             0000   292 F_OFFSET=6*2                       ;OFFSET HEADS
0000000E             0000   293 F_RETCENTER=7*2                    ;RETURN TO CENTERLINE
00000010             0000   294 F_READPRESET=8*2                   ;READ IN PRESET
00000012             0000   295 F_PACKACK=9*2                      ;PACK ACKNOWLEDGE
00000018             0000   296 F_SEARCH=12*2                      ;SEARCH FOR SECTOR
00000018             0000   297 F_SEARCHA=12*2                     ;SEARCH AHEAD FOR SECTOR
0000001C             0000   298 F_DIAGNOSE=14*2                    ;DIAGNOSE DRIVE
00000028             0000   299 F_WRITECHECK=20*2                  ;WRITE CHECK DATA
0000002A             0000   300 F_WRITECHECKH=21*2                 ;WRITE CHECK HEADER AND DATA
00000030             0000   301 F_WRITEDATA=24*2                   ;WRITE DATA
00000032             0000   302 F_WRITEHEAD=25*2                   ;WRITE HEADER AND DATA
00000034             0000   303 F_WRITETRACKD=26*2                 ;WRITE TRACK DESCRIPTOR
00000038             0000   304 F_READDATA=28*2                    ;READ DATA
0000003A             0000   305 F_READHEAD=29*2                    ;READ HEADER AND DATA
0000003C             0000   306 F_READTRACKD=30*2                  ;READ TRACK DESCRIPTOR
00000000             0000   307 F_AVAILABLE=F_NOP                  ;AVAILABLE
                     0000   308
                     0000   309 ;
                     0000   310 ; LOCAL DATA
                     0000   311 ;
                     0000   312 ; DRIVER PROLOGUE TABLE
                     0000   313 ;
                     0000   314
                     0000   315         DPTAB   -                  ;DEFINE DRIVER PROLOGUE TABLE
                     0000   316                 END=DR_END,-       ;END OF DRIVER
                     0000   317                 ADAPTER=MBA,-      ;ADAPTER TYPE
                     0000   318                 FLAGS=DPT$M_SVP,-  ;SYSTEM PAGE TABLE ENTRY REQUIRED
                     0000   319                 UCBSIZE=UCB$K_DR_LENGTH,-      ;UCB size
                     0000   320                 NAME=DRDRIVER      ;DRIVER NAME
                     0038   321         DPT_STORE INIT             ;CONTROL BLOCK INIT VALUES
                     0038   322         DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\> ;DEFAULT ACP NAME
                     003F   323         DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_PACK ;ACP CLASS
                     0043   324         DPT_STORE UCB,UCB$B_FIPL,B,8   ;FORK IPL
                     0047   325         DPT_STORE UCB,UCB$L_DEVCHAR,L,- ;DEVICE CHARACTERISTICS
                     0047   326                 <DEV$M_FOD-        ; FILES ORIENTED
                     0047   327                 !DEV$M_DIR-        ; DIRECTORY STRUCTURED
                     0047   328                 !DEV$M_AVL-        ; AVAILABLE
                     0047   329                 !DEV$M_ELG-        ; ERROR LOGGING ENABLED
                     0047   330                 !DEV$M_SHR-        ; SHAREABLE
                     0047   331                 !DEV$M_IDV-        ; INPUT DEVICE
                     0047   332                 !DEV$M_ODV-        ; OUTPUT DEVICE
                     0047   333                 !DEV$M_RND>        ; RANDOM ACCESS
                     004E   334         DPT_STORE UCB,UCB$L_DEVCHAR2,L,-; DEVICE CHARACTERISTICS
                     004E   335                 <DEV$M_NNM>        ; PREFIX NAME WITH "node$"
                     0055   336         DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_DISK ;DEVICE CLASS
                     0059   337         DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
                     005E   338         DPT_STORE UCB,UCB$B_DIPL,B,21  ;DEVICE IPL
                     0062   339         DPT_STORE UCB,UCB$B_ERTCNT,B,8 ;ERROR RETRY COUNT
                     0066   340         DPT_STORE UCB,UCB$B_ERTMAX,B,8 ;MAX ERROR RETRY COUNT
                     006A   341         DPT_STORE REINIT           ;CONTROL BLOCK RE-INIT VALUES
                     006A   342         DPT_STORE DDB,DDB$L_DDT,D,DR$DDT ;DDT ADDRESS
```

```
                006F   343          DPT_STORE END                   ;
                0000   344
                0000   345 ;
                0000   346 ; DRIVER DISPATCH TABLE
                0000   347 ;
                0000   348
                0000   349          DDTAB   DR,-                    ;DRIVER DISPATCH TABLE
                0000   350                  DR_STARTIO,-            ;START I/O OPERATION
                0000   351                  DR_UNSOLNT,-            ;UNSOLICITED INTERRUPT
                0000   352                  DR_FUNCTABLE,-          ;FUNCTION DECISION TABLE
                0000   353                  0,-                     ;CANCEL I/O ENTRY POINT
                0000   354                  DR_REGDUMP,-            ;REGISTER DUMP ROUTINE
                0000   355                  <<RM_EC2+4+4+MBA$L_BCR+4+8>+<<5+5+1>*4>>,- ;DIAG BUFF SIZE
                0000   356                  <<RM_EC2+4+4+MBA$L_BCR+4+8>+<1*4>+<EMB$L_DV_REGSAV>>,- ;ERLG BUFF SI
                0000   357                  DR_UNIT_INIT            ;UNIT INITIALIZATION
                0038   358
                0038   359 ;
                0038   360 ; DATA CHECK FUNCTION TRANSLATION TABLE
                0038   361 ;
                0038   362
                0038   363 CHECKTAB:                                ;
          0A'   0038   364          .BYTE   CDF_WRITECHECK          ;WRITE DATA
          0A'   0039   365          .BYTE   CDF_WRITECHECK          ;READ DATA
          12'   003A   366          .BYTE   CDF_WRITECHECKH         ;WRITE HEADER AND DATA
          12'   003B   367          .BYTE   CDF_WRITECHECKH         ;READ HEADER AND DATA
                003C   368
                003C   369 ;
                003C   370 ; DRIVE TYPE DESCRIPTOR TABLE
                003C   371 ;
                003C   372 DR_DTDESC:                               ;
          0014  003C   373          .WORD   ^X14                    ; RM03
            06  003E   374          .BYTE   DT$_RM03                ;
            20  003F   375          .BYTE   32                      ; 32 SECTORS
            05  0040   376          .BYTE   5                       ; 5 TRACKS
          0337  0041   377          .WORD   823                     ; 823 CYLINDERS
      00020260  0043   378          .LONG   823*5*32                ; MAXIMUM BLOCKS
      24A4D003  0047   379          .LONG   ^X24A4D003              ; MEDIA IDENT "DR RM03"
      0000000F  004B   380 DR_DTDESCLEN=.-DR_DTDESC                 ;LENGTH OF DRIVE TYPE DESCRIPTOR
                004B   381
          0016  004B   382          .WORD   ^X16                    ; RM80
            0D  004D   383          .BYTE   DT$_RM80                ;
            1F  004E   384          .BYTE   31                      ; 31 SECTORS
            0E  004F   385          .BYTE   14                      ; 14 TRACKS
          022F  0050   386          .WORD   559                     ; 559 CYLINDERS
      0003B3AE  0052   387          .LONG   559*14*31               ; MAXIMUM BLOCKS
      24A4D050  0056   388          .LONG   ^X24A4D050              ; MEDIA IDENT "DR RM80"
          0017  005A   389          .WORD   ^X17                    ; RM05
            0F  005C   390          .BYTE   DT$_RM05                ;
            20  005D   391          .BYTE   32                      ; 32 SECTORS
            13  005E   392          .BYTE   19                      ; 19 TRACKS
          0337  005F   393          .WORD   823                     ; 823 CYLINDERS
      0007A2A0  0061   394          .LONG   823*19*32               ; MAXIMUM BLOCKS
      24A4D005  0065   395          .LONG   ^X24A4D005              ; MEDIA IDENT "DR RM05"
          0022  0069   396          .WORD   ^X22                    ; RP07
            07  006B   397          .BYTE   DT$_RP07                ;
            32  006C   398          .BYTE   50                      ; 50 SECTORS
            20  006D   399          .BYTE   32                      ; 32 TRACKS
```

```
                0276  006E    400          .WORD   630                     ; 630 CYLINDERS
            000F6180  0070    401          .LONG   630*32*50               ; MAXIMUM BLOCKS
            24A50007  0074    402          .LONG   ^X24A50007              ; MEDIA IDENT "DR RP07"
                      0078    403
                0000  0078    404          .WORD   0                       ; END OF TABLE
            00000089  007A    405          .BLKB   DR_DTDESCLEN            ; SPARE DRIVE TYPE SLOT
            00000089  0089    406          .BLKB   DR_DTDESCLEN            ; SPARE DRIVE TYPE SLOT
                      0098    407
                      0098    408 ;
                      0098    409 ; HARDWARE I/O FUNCTION CODE TABLE
                      0098    410 ;
                      0098    411
                      0098    412 FTAB:                                     ;
                      0098    413          GENF    F_NOP                    ;NO OPERATION
                      0099    414          GENF    F_NOP                    ;(NO UNLOAD FUNCTION)
                      009A    415          GENF    F_SEEK                   ;SEEK CYLINDER
                      009B    416          GENF    F_RECAL                  ;RECALIBRATE
                      009C    417          GENF    F_DRVCLR                 ;DRIVE CLEAR
                      009D    418          GENF    F_NOP                    ;(NO RELEASE PORT)
                      009E    419          GENF    F_OFFSET                 ;OFFSET HEADS
                      009F    420          GENF    F_RETCENTER              ;RETURN HEADS TO CENTERLINE
                      00A0    421          GENF    F_PACKACK                ;PACK ACKNOWLEDGE
                      00A1    422          GENF    F_SEARCH                 ;SEARCH FOR SECTOR
                      00A2    423          GENF    F_WRITECHECK             ;WRITE CHECK
                      00A3    424          GENF    F_WRITEDATA              ;WRITE DATA
                      00A4    425          GENF    F_READDATA               ;READ DATA
                      00A5    426          GENF    F_WRITEHEAD              ;WRITE HEADER AND DATA
                      00A6    427          GENF    F_READHEAD               ;READ HEADER AND DATA
                      00A7    428          GENF    F_WRITETRACKD            ;WRITE TRACK DESCRIPTOR
                      00A8    429          GENF    F_READTRACKD             ;READ TRACK DESCRIPTOR
                      00A9    430          GENF    F_AVAILABLE              ;AVAILABLE
                      00AA    431          GENF    F_WRITECHECKH            ;WRITE CHECK HEADER AND DATA
                      00AB    432          GENF    F_READPRESET             ;READ IN PRESET
                      00AC    433          GENF    F_DIAGNOSE               ;DIAGNOSE THE DRIVE
                      00AD    434          GENF    F_SEARCHA                ;SEARCH AHEAD OF SECTOR
                      00AE    435
                      00AE    436 ;
                      00AE    437 ; OFFSET TABLE
                      00AE    438 ;
                      00AE    439
                      00AE    440 OFFTAB:                                   ;
                   00 00AE    441          .BYTE   0                        ;RETURN TO CENTERLINE
                   01 00AF    442          .BYTE   ^X01                     ;+ OFFSET (BIT 0 = OFFSET FLAG)
                   81 00B0    443          .BYTE   ^X81                     ;- OFFSET (BIT 0 = OFFSET FLAG)
                   00 00B1    444          .BYTE   0                        ;RETURN TO CENTERLINE
            00000004  00B2    445 OFFSIZ=.-OFFTAB                           ;SIZE OF OFFSET TABLE
```

```
00B2    447          .SBTTL  FUNCTION DECISION TABLE
00B2    448 ;+
00B2    449 ; RM03 FUNCTION DECISION TABLE
00B2    450 ;-
00B2    451
00B2    452 DR_FUNCTABLE:                               ;FUNCTION DECISION TABLE
00B2    453          FUNCTAB ,-                         ;LEGAL FUNCTIONS
00B2    454                  <NOP,-                     ;NO OPERATION
00B2    455                  UNLOAD,-                   ;UNLOAD VOLUME
00B2    456                  SEEK,-                     ;SEEK CYLINDER
00B2    457                  RECAL,-                    ;RECALIBRATE
00B2    458                  DRVCLR,-                   ;DRIVE CLEAR
00B2    459                  RELEASE,-                  ;RELEASE PORT
00B2    460                  OFFSET,-                   ;OFFSET HEADS
00B2    461                  RETCENTER,-                ;RETURN HEADS TO CENTERLINE
00B2    462                  PACKACK,-                  ;PACK ACKNOWLEDGE
00B2    463                  SEARCH,-                   ;SEARCH FOR SECTOR
00B2    464                  READPRESET,-               ;READ IN PRESET
00B2    465                  SENSECHAR,-                ;SENSE CHARACTERISTICS
00B2    466                  SETCHAR,-                  ;SET CHARACTERISTICS
00B2    467                  SENSEMODE,-                ;SENSE MODE
00B2    468                  SETMODE,-                  ;SET MODE
00B2    469                  WRITECHECK,-               ;WRITE CHECK
00B2    470                  WRITEHEAD,-                ;WRITE HEADER AND DATA
00B2    471                  READHEAD,-                 ;READ HEADER AND DATA
00B2    472                  WRITETRACKD,-              ;WRITE TRACK DESCRIPTOR
00B2    473                  READTRACKD,-               ;READ TRACK DESCRIPTOR
00B2    474                  WRITECHECKH,-              ;WRITE CHECK HEADER AND DATA
00B2    475                  DIAGNOSE,-                 ;DIAGNOSE THE DRIVE
00B2    476                  READLBLK,-                 ;READ LOGICAL BLOCK
00B2    477                  WRITELBLK,-                ;WRITE LOGICAL BLOCK
00B2    478                  READPBLK,-                 ;READ PHYSICAL BLOCK
00B2    479                  WRITEPBLK,-                ;WRITE PHYSICAL BLOCK
00B2    480                  READVBLK,-                 ;READ VIRTUAL BLOCK
00B2    481                  WRITEVBLK,-                ;WRITE VIRTUAL BLOCK
00B2    482                  AVAILABLE,-                ;AVAILABLE
00B2    483                  ACCESS,-                   ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00B2    484                  ACPCONTROL,-               ;ACP CONTROL FUNCTION
00B2    485                  CREATE,-                   ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00B2    486                  DEACCESS,-                 ;DEACCESS FILE
00B2    487                  DELETE,-                   ;DELETE FILE AND/OR DIRECTORY ENTRY
00B2    488                  MODIFY,-                   ;MODIFY FILE ATTRIBUTES
00B2    489                  MOUNT>                     ;MOUNT VOLUME
00BA    490          FUNCTAB ,-                         ;BUFFERED I/O FUNCTIONS
00BA    491                  <NOP,-                     ;NO OPERATION
00BA    492                   UNLOAD,-                  ;UNLOAD VOLUME
00BA    493                   SEEK,-                    ;SEEK CYLINDER
00BA    494                   RECAL,-                   ;RECALIBRATE
00BA    495                   DRVCLR,-                  ;DRIVE CLEAR
00BA    496                   RELEASE,-                 ;RELEASE PORT
00BA    497                   OFFSET,-                  ;OFFSET HEADS
00BA    498                   RETCENTER,-               ;RETURN HEADS TO CENTERLINE
00BA    499                   PACKACK,-                 ;PACK ACKNOWLEDGE
00BA    500                   SEARCH,-                  ;SEARCH FOR SECTOR
00BA    501                   READPRESET,-              ;READ IN PRESET
00BA    502                   DIAGNOSE,-                ;DIAGNOSE DRIVE
00BA    503                   SENSECHAR,-               ;SENSE CHARACTERISTICS
```

```
00BA   504                 SETCHAR,-                ;SET CHARACTERISTICS
00BA   505                 SENSEMODE,-              ;SENSE MODE
00BA   506                 SETMODE,-                ;SET MODE
00BA   507                 AVAILABLE,-              ;AVAILABLE
00BA   508                 ACCESS,-                 ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00BA   509                 ACPCONTROL,-             ;ACP CONTROL FUNCTION
00BA   510                 CREATE,-                 ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00BA   511                 DEACCESS,-               ;DEACCESS FILE
00BA   512                 DELETE,-                 ;DELETE FILE AND/OR DIRECTORY ENTRY
00BA   513                 MODIFY,-                 ;MODIFY FILE ATTRIBUTES
00BA   514                 MOUNT>                   ;MOUNT VOLUME
00C2   515         FUNCTAB +ACP$READBLK,-           ;READ FUNCTIONS
00C2   516                 <READTRACKD,-            ;READ TRACK DESCRIPTOR
00C2   517                  READHEAD,-              ;READ HEADER
00C2   518                  READLBLK,-              ;READ LOGICAL BLOCK
00C2   519                  READPBLK,-              ;READ PHYSICAL BLOCK
00C2   520                  READVBLK>               ;READ VIRTUAL BLOCK
00CE   521         FUNCTAB +ACP$WRITEBLK,-          ;WRITE FUNCTIONS
00CE   522                 <WRITETRACKD,-           ;WRITE TRACK DESCRIPTOR
00CE   523                  WRITECHECK,-            ;WRITE CHECK
00CE   524                  WRITECHECKH,-           ;WRITE CHECK HEADER AND DATA
00CE   525                  WRITEHEAD,-             ;WRITE HEADER
00CE   526                  WRITELBLK,-             ;WRITE LOGICAL BLOCK
00CE   527                  WRITEPBLK,-             ;WRITE PHYSICAL BLOCK
00CE   528                  WRITEVBLK>              ;WRITE VIRTUAL BLOCK
00DA   529         FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00E6   530         FUNCTAB +ACP$DEACCESS,<DEACCESS> ;DEACCESS FILE
00F2   531         FUNCTAB +ACP$MODIFY,-            ;
00F2   532                 <ACPCONTROL,-            ;ACP CONTROL FUNCTION
00F2   533                  DELETE,-                ;DELETE FILE OR DIRECTORY ENTRY
00F2   534                  MODIFY>                 ;MODIFY FILE ATTRIBUTES
00FE   535         FUNCTAB +ACP$MOUNT,<MOUNT>       ;MOUNT VOLUME
010A   536         FUNCTAB +EXE$LCLDSKVALID,-       ;LOCAL DISK VALID FUNCTIONS
010A   537                 <UNLOAD,-                ;UNLOAD VOLUME
010A   538                  AVAILABLE,-             ;UNIT AVAILABLE
010A   539                  PACKACK>                ;PACK ACKNOWLEDGE
0116   540         FUNCTAB +EXE$ZEROPARM,-          ;ZERO PARAMETER FUNCTIONS
0116   541                 <NOP,-                   ;NO OPERATION
0116   542                  UNLOAD,-                ;UNLOAD VOLUME
0116   543                  RECAL,-                 ;RECALIBRATE
0116   544                  DRVCLR,-                ;DRIVE CLEAR
0116   545                  RELEASE,-               ;RELEASE PORT
0116   546                  RETCENTER,-             ;RETURN HEADS TO CENTERLINE
0116   547                  READPRESET,-            ;READ IN PRESET
0116   548                  PACKACK,-               ;PACK ACKNOWLEDGE
0116   549                  AVAILABLE>              ;AVAILABLE
0122   550         FUNCTAB +EXE$ONEPARM,-           ;ONE PARAMETER FUNCTIONS
0122   551                 <SEEK,-                  ;SEEK CYLINDER
0122   552                  OFFSET,-                ;OFFSET HEADS
0122   553                  SEARCH,-                ;SEARCH FOR SECTOR
0122   554                  DIAGNOSE>               ;DIAGNOSE THE DRIVE
012E   555         FUNCTAB +EXE$SENSEMODE,-         ;
012E   556                 <SENSECHAR,-             ;SENSE CHARACTERISTICS
012E   557                  SENSEMODE>              ;SENSE MODE
013A   558         FUNCTAB +EXE$SETCHAR,-           ;
013A   559                 <SETCHAR,-               ;SET CHARACTERISTICS
013A   560                  SETMODE>                ;SET MODE
```

```
                                0146   562          .SBTTL  START I/O OPERATION
                                0146   563 ;+
                                0146   564 ; DR_STARTIO - START I/O OPERATION ON DEVICE UNIT
                                0146   565 ;
                                0146   566 ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
                                0146   567 ;
                                0146   568 ; INPUTS:
                                0146   569 ;
                                0146   570 ;       R3 = ADDRESS OF I/O PACKET.
                                0146   571 ;       R5 = UCB ADDRESS OF DEVICE UNIT.
                                0146   572 ;
                                0146   573 ; OUTPUTS:
                                0146   574 ;
                                0146   575 ;       FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
                                0146   576 ;       RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
                                0146   577 ;       THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
                                0146   578 ;-
                                0146   579
                                0146   580 DR_STARTIO:                                    ;START I/O OPERATION
0080 C5   0081 C5   90          0146   581          MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
  009A C5   20 A3   B0          014D   582          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
        00D2 C5     B4          0153   583          CLRW    UCB$W_DR_MR(R5)           ;CLEAR THE MAINTENANCE VALUE
00D4 C5   FFF7 8F   AA          0157   584          BICW    #^CDR_M_DUALPORT, -       ;Clear software status and error log
                                015E   585                  UCB$B_DR_SSTS(R5)         ;bytes, except for dualport bit.
      50   38 A3   D0          015E   586          MOVL    IRP$L_MEDIA(R3),R0        ;GET PARAMETER LONGWORD
                                0162   587
                                0162   588 ;
                                0162   589 ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                                0162   590 ;
                                0162   591
    06   00   EF                0162   592 10$:     EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODE
  51    20 A3                   0165   593                  IRP$W_FUNC(R3),R1         ;
    51   02   91                0168   594          CMPB    #IO$_SEEK,R1              ;SEEK FUNCTION?
         1E   13                016B   595          BEQL    20$                       ;IF EQL YES
    51   06   91                016D   596          CMPB    #IO$_OFFSET,R1            ;OFFSET FUNCTION?
         20   13                0170   597          BEQL    30$                       ;IF EQL YES
    51   09   91                0172   598          CMPB    #IO$_SEARCH,R1            ;SEARCH FUNCTION?
         22   13                0175   599          BEQL    40$                       ;IF EQL YES
    51   1D   91                0177   600          CMPB    #IO$_DIAGNOSE,R1          ;DIAGNOSE FUNCTION?
         24   13                017A   601          BEQL    45$                       ;IF EQL YES
00BC C5  50   D0                017C   602          MOVL    R0,UCB$W_DA(R5)           ;STORE PARAMETER LONGWORD
    51   18   91                0181   603          CMPB    #IO$_WRITECHECKH,R1       ;DISJOINT FUNCTION CODE?
         22   1A                0184   604          BGTRU   50$                       ;IF GTRU NO
    51   06   A2                0186   605          SUBW    #IO$_WRITECHECKH-IO$_AVAILABLE-1,R1 ;MAKE FUNCTION TABLE INDEX
         1D   11                0189   606          BRB     50$                       ;
                                018B   607
                                018B   608 ;
                                018B   609 ; SEEK FUNCTION - SET CYLINDER ADDRESS
                                018B   610 ;
                                018B   611
00BE C5   50   B0               018B   612 20$:     MOVW    R0,UCB$W_DC(R5)           ;SET CYLINDER ADDRESS
         16   11                0190   613          BRB     50$                       ;
                                0192   614
                                0192   615 ;
                                0192   616 ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
                                0192   617 ;
                                0192   618
```

```
                     00C8 C5   50   90  0192   619 30$:    MOVB    R0,UCB$W_OFFSET(R5)     ;SET OFFSET VALUE
                               0F   11  0197   620          BRB     50$                     ;
                                        0199   621
                                        0199   622 ;
                                        0199   623 ; SEARCH FUNCTION - SET SECTOR ADDRESS
                                        0199   624 ;
                                        0199   625
                     00BC C5   50   90  0199   626 40$:    MOVB    R0,UCB$W_DA(R5)         ;SET SECTOR ADDRESS
                               08   11  019E   627          BRB     50$                     ;
                                        01A0   628
                                        01A0   629 ;
                                        01A0   630 ; DIAGNOSE FUNCTION - SET MAINTENANCE VALUE
                                        01A0   631 ;
                                        01A0   632
                     00D2 C5   50   B0  01A0   633 45$:    MOVW    R0,UCB$W_DR_MR(R5)      ;SET MAINTENANCE VALUE
                          51   03   A2  01A5   634          SUBW    #IO$_DIAGNOSE-IO$_READPRESET-1,R1 ;MAKE A FUNCTION TABLE INDEX
                                        01A8   635
                                        01A8   636 ;
                                        01A8   637 ; FINISH PREPROCESSING
                                        01A8   638 ;
                                        01A8   639
                     0092 C5   51   90  01A8   640 50$:    MOVB    R1,UCB$B_FEX(R5)        ;SAVE FUNCTION DISPATCH INDEX
                       54   24 A5   D0  01AD   641          MOVL    UCB$L_CRB(R5),R4        ;GET ADDRESS OF CRB
                       54   2C B4   D0  01B1   642          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4 ;GET FIRST CONTROLLER CSR ADDRESS
                    00 68 A5   00   E4  01B5   643          BBSC    #UCB$V_ECC,UCB$W_DEVSTS(R5),FDISPATCH ;CLEAR ECC CORRECTION MADE
                                        01BA   644
                                        01BA   645 ;
                                        01BA   646 ; CENTRAL FUNCTION DISPATCH
                                        01BA   647 ;
                                        01BA   648
                                        01BA   649 FDISPATCH:                               ;FUNCTION DISPATCH
                       53   58 A5   D0  01BA   650          MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF I/O PACKET
                    0D 2A A3    08   E0  01BE   651          BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
                    08 64 A5    0B   E0  01C3   652          BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
                     50    0254 8F   3C  01C8   653          MOVZWL  #SS$_VOLINV,R0          ;SET VOLUME INVALID STATUS
                             06D1   31  01CD   654          BRW     RESETXFR                ;
                                        01D0   655
                                        01D0   656 ;
                                        01D0   657 ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
                                        01D0   658 ;
                                        01D0   659
                       50   0092 C5   9A  01D0   660 10$:    MOVZBL  UCB$B_FEX(R5),R0        ;GET DISPATCH FUNCTION CODE
                     00C9 C5    10   90  01D5   661          MOVB    #RM_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
                     00CB C5    01   90  01DA   662          MOVB    #1,UCB$B_OFFRTC(R5)     ;SET INITIAL OFFSET RETRY COUNT
                          00CA C5   94  01DF   663          CLRB    UCB$B_OFFNDX(R5)        ;CLEAR INITIAL OFFSET TABLE INDEX
                                        01E3   664
                                        01E3   665 ;
                                        01E3   666 ; CHECK FOR DIAGNOSTIC MODIFIERS
                                        01E3   667 ;
                      2F 2A A3    08   E1  01E3   668          BBC     #IRP$V_PHYSIO,IRP$W_STS(R3),40$  ;IF CLEAR, NOT PHYSICAL I/O
                    06 009A C5    06   E1  01E8   669          BBC     #IO$V_COMMOD,UCB$W_FUNC(R5),15$  ;IF CLEAR, NO COMMAND MODIFIER
                    00C9 C5    80 8F   88  01EE   670          BISB    #RM_OF_M_CMO/256,UCB$W_OFFSET+1(R5)  ;SET COMMAND MODIFIER
                                        01F4   671
                    06 009A C5    07   E1  01F4   672 15$:    BBC     #IO$V_MOVETRACKD,UCB$W_FUNC(R5),20$  ;IF CLR, NO MOVE TRACK DESC
                    00C9 C5    40 8F   88  01FA   673          BISB    #RM_OF_M_MTD/256,UCB$W_OFFSET+1(R5)  ;SET MOVE TRACK DESCRIPTOR
                                        0200   674
                    06 009A C5    08   E1  0200   675 20$:    BBC     #IO$V_DIAGNOSTIC,UCB$W_FUNC(R5),30$  ;IF CLEAR, NOT DIAG MODE
```

```
                00D3 C5   80 8F   88  0206   676          BISB    #PM_MR_M_DM/256,UCB$W_DR_MR+1(R5)  ;SET DIAGNOSTIC MODE
                                    020C   677
     05 009A C5   09   E1   020C   678 30$:     BBC     #IO$V_SKPSECINH,UCB$W_FUNC(R5),40$  ;IF CLEAR, NO SSEI MODIFIER
        00C9 C5   02   88   0212   679          BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)  ;SET SKIP SECTOR ERR INH
                            0217   680
                            0217   681 ;
                            0217   682 ; DISPATCH TO FUNCTION HANDLING ROUTINE
                            0217   683 ;
                            0217   684 40$:
                            0217   685          CASE    R0,<-                       ;DISPATCH TO FUNCTION HANDLING ROUTINE
                            0217   686                  NOP,-                       ;NO OPERATION
                            0217   687                  UNLOAD,-                    ;UNLOAD VOLUME
                            0217   688                  SEEK,-                      ;SEEK CYLINDER
                            0217   689                  RECAL,-                     ;RECALIBRATE
                            0217   690                  DRVCLR,-                    ;DRIVE CLEAR
                            0217   691                  RELEASE,-                   ;RELEASE PORT
                            0217   692                  OFFSET,-                    ;OFFSET HEADS
                            0217   693                  RETCENTER,-                 ;RETURN HEADS TO CENTER
                            0217   694                  PACKACK,-                   ;PACK ACKNOWLEDGE
                            0217   695                  SEARCH,-                    ;SEARCH FOR SECTOR
                            0217   696                  WRITECHECK,-                ;WRITE CHECK DATA
                            0217   697                  WRITEDATA,-                 ;WRITE DATA
                            0217   698                  READDATA,-                  ;READ DATA
                            0217   699                  WRITEHEAD,-                 ;WRITE HEADER AND DATA
                            0217   700                  READHEAD,-                  ;READ HEADER AND DATA
                            0217   701                  WRITETRACKD,-               ;WRITE TRACK DESCRIPTOR
                            0217   702                  READTRACKD,-                ;READ TRACK DESCRIPTOR
                            0217   703                  AVAILABLE,-                 ;AVAILABLE
                            0217   704                  WRITECHECKH,-               ;WRITE CHECK HEADER AND DATA
                            0217   705                  READPRESET,-                ;READIN PRESET
                            0217   706                  DIAGNOSE>                   ;DIAGNOSE DRIVE
                            0245   707
                            0245   708 ; UNLOAD or AVAILABLE - Clear UCB$V_VALID
                            0245   709 ;   This is the only operation which these functions need to perform.  All
                            0245   710 ;   devices supported by this driver do not have an unload function, and the
                            0245   711 ;   available function code should only clear the UCB$V_VALID bit.
                            0245   712
                            0245   713 UNLOAD:
                            0245   714 AVAILABLE:
     64 A5    0800 8F   AA  0245   715          BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear the software volume valid
                00AF   31   024B   716          BRW     NORMAL                      ;bit and complete function.
                            024E   717
                            024E   718 ;
                            024E   719 ; PACKACK - Set UCB$V_VALID and proceed with hardware pack acknowledge
                            024E   720 ;           function
                            024E   721 ;
                            024E   722 PACKACK:
     64 A5    0800 8F   A8  024E   723          BISW    #UCB$M_VALID, UCB$W_STS(R5) ;Set the software volume valid
                            0254   724 ;        BRB     NOP                         ;bit and complete function.
                            0254   725
                            0254   726 ;
                            0254   727 ; NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
                            0254   728 ; RETURN TO CENTER LINE, SEARCH, AND READ IN PRESET
                            0254   729 ;
                            0254   730
                            0254   731 NOP:                                        ;NO OPERATION
                            0254   732 SEEK:                                       ;SEEK CYLINDER
```

```
                     0254   733 RECAL:                                  ;RECALIBRATE
                     0254   734 DRVCLR:                                 ;DRIVE CLEAR
                     0254   735 RELEASE:                                ;RELEASE PORT
                     0254   736 OFFSET:                                 ;OFFSET READ HEADS
                     0254   737 RETCENTER:                              ;RETURN TO CENTERLINE
                     0254   738 SEARCH:                                 ;SEARCH FOR SECTOR
                     0254   739 READPRESET:                             ;READIN PRESET
                     0254   740         EXFUNC  RETRY                   ;EXECUTE HOUSEKEEPING FUNCTION
           00A1  31  0259   741         BRW     NORMAL                  ;
                     025C   742
                     025C   743 ;
                     025C   744 ; WRITE TRACK DESCRIPTOR and READ TRACK DESCRIPTOR
                     025C   745 ;       Both want to SEEK rather than to SEARCH to arrive on cylinder.
                     025C   746 ;
                     025C   747
                     025C   748 WRITETRACKD:                             ;WRITE TRACK DESCRIPTOR
00D4 C5   04  88     025C   749         BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
                     0261   750
                     0261   751 READTRACKD:                              ;READ TRACK DESCRIPTOR
           0C  E0    0261   752         BBS     #IO$V_INHSEEK,-
         23 009A C5          0263   753                 UCB$W_FUNC(R5),TRANRQCH ; If set, NO explicit SEEK
                     0267   754         EXFUNC  RETRY,F_SEEK            ; Seek to cylinder
           19  11    026F   755         BRB     TRANRQCA                ;  and branch around to common code.
                     0271   756
                     0271   757 ;
                     0271   758 ; WRITE CHECK DATA AND WRITE CHECK HEADER AND DATA
                     0271   759 ;
                     0271   760
                     0271   761 WRITECHECK:                             ;WRITE CHECK DATA
                     0271   762 WRITECHECKH:                            ;WRITE CHECK HEADER AND DATA
   00 009A C5  0E    E4 0271   763         BBSC    #IO$V_DATACHECK,UCB$W_FUNC(R5),WRITEDATA ;CLEAR DATA CHECK REQUEST
                     0277   764
                     0277   765 ;
                     0277   766 ; WRITE DATA, WRITE HEADER AND DATA,
                     0277   767 ; WRITE CHECK DATA, AND WRITE CHECK HEADER AND DATA
                     0277   768 ;
                     0277   769
                     0277   770 WRITEDATA:                              ;WRITE DATA
                     0277   771 WRITEHEAD:                              ;WRITE HEADER AND DATA
   00D4 C5  04  88   0277   772         BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
                     027C   773
                     027C   774 ;
                     027C   775 ; READ DATA, READ HEADER AND DATA,
                     027C   776 ; WRITE DATA, WRITE HEADER AND DATA,
                     027C   777 ; WRITE CHECK DATA, AND WRITE CHECK HEADER AND DATA
                     027C   778 ;
                     027C   779
                     027C   780 READDATA:                               ;READ DATA
                     027C   781 READHEAD:                               ;READ HEADER AND DATA
   08 009A C5  0C    E0 027C   782         BBS     #IO$V_INHSEEK,UCB$W_FUNC(R5),TRANRQCH ;IF SET, NO EXPLICIT SEEK
                     0282   783         EXFUNC  RETRY,F_SEARCHA         ;SEARCH AHEAD OF STARTING SECTOR
                     028A   784
                     028A   785 ;
                     028A   786 ; DATA TRANSFER OR DIAGNOSE - REQUEST CHANNEL
                     028A   787 ;
                     028A   788
                     028A   789 DIAGNOSE:                               ;DIAGNOSE
```

```
                     028A   790 TRANRQCH:                                       ;DATA TRANSFER
                     028A   791         REQPCHAN LOW                            ;REQUEST PRIMARY CHANNEL
                     0290   792
                     0290   793 ;
                     0290   794 ; DATA TRANSFER - CHANNEL ALREADY OWNED
                     0290   795 ;
                     0290   796
                     0290   797 TRANNOCH:                                       ;DATA TRANSFER CHANNEL OWNED
   50    0092 C5    9A  0290   798         MOVZBL  UCB$B_FEX(R5),R0             ;GET FUNCTION DISPATCH INDEX
                     0295   799         EXFUNC  TRANXT                          ;EXECUTE TRANSFER FUNCTION
                     029A   800
                     029A   801 ;
                     029A   802 ; DATA CHECK
                     029A   803 ;
                     029A   804
                     029A   805 DATACHECK:                                       ;DATA CHECK
5D 009A C5    0E    E1  029A   806         BBC     #IO$V_DATACHECK,UCB$W_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
   50    0639 8F    3C  02A0   807         MOVZWL  #SS$_BASECC,R0               ;ASSUME ECC CORRECTION WAS MADE
 56 68 A5    00    E0  02A5   808         BBS     #UCB$V_ECC,UCB$W_DEVSTS(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
                     02AA   809         RELCHAN                                 ;RELEASE CHANNEL
   00D4 C5    01    88  02B0   810         BISB    #DR_M_DCK,UCB$B_DR_SSTS(R5)  ;SET DATA CHECK IN PROGRESS
   00C9 C5    10    90  02B5   811         MOVB    #RM_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
   00D4 C5    04    88  02BA   812         BISB    #DR_M_NOECC, UCB$B_DR_SSTS(R5) ; Signal don't correct with ECC.
   00CB C5    01    90  02BF   813         MOVB    #1,UCB$B_OFFRTC(R5)          ;SET INITIAL OFFSET RETRY COUNT
        00CA C5    94  02C4   814         CLRB    UCB$B_OFFNDX(R5)             ;CLEAR INITIAL OFFSET TABLE INDEX
     52    58 A5    D0  02C8   815         MOVL    UCB$L_IRP(R5),R2             ;GET ADDRESS OF IRP
  78 A5    2C A2    7D  02CC   816         MOVQ    IRP$L_SVAPTE(R2),UCB$L_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
00BC C5    38 A2    D0  02D1   817         MOVL    IRP$L_MEDIA(R2),UCB$W_BA(R5) ;
  0B 2A A2    08    E1  02D7   818         BBC     #IRP$V_PHYSIO,IRP$W_STS(R2),CHECKRETRY  ;IF CLEAR NOT PHYS I/O
05 009A C5    09    E1  02DC   819         BBC     #IO$V_SKPSECINH,UCB$W_FUNC(R5),CHECKRETRY  ;IF CLEAR NO SSEI MOD
   00C9 C5    02    88  02E2   820         BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)  ;SET SKIP SECTOR ERR INH
                     02E7   821
                     02E7   822 ;
                     02E7   823 ; DATA CHECK RETRY
                     02E7   824 ;
                     02E7   825
                     02E7   826 CHECKRETRY:                                      ;DATA CHECK RETRY
                     02E7   827         REQPCHAN LOW                            ;REQUEST PRIMARY CHANNEL FOR DATA CHECK
  50    0092 C5    9A  02ED   828         MOVZBL  UCB$B_FEX(R5),R0             ;GET FUNCTION DISPATCH INDEX
50   FD36 CF40    9A  02F2   829         MOVZBL  CHECKTAB-CDF_WRITEDATA[R0],R0 ;GET CASE TABLE INDEX
                     02F8   830         EXFUNC  TRANXT                          ;EXECUTE DATA CHECK FUNCTION
                     02FD   831
                     02FD   832 ;
                     02FD   833 ; SUCCESSFUL OPERATION COMPLETION
                     02FD   834 ;
                     02FD   835
                     02FD   836 NORMAL:                                          ;
      50    01    3C  02FD   837         MOVZWL  #SS$_NORMAL,R0               ;SET NORMAL COMPLETION STATUS
                     0300   838 CHECKXT:                                         ;
         0208    31  0300   839         BRW     FUNCXT                          ;
                     0303   840
                     0303   841 ;
                     0303   842 ; TRANSFER ENDED WITH A RETRIABLE ERROR
                     0303   843 ;
                     0303   844
                     0303   845 TRANXT:                                        ;TRANSFER EXIT
0093 C5    0B    91  0303   846         CMPB    #CDF_WRITEDATA,UCB$B_CEX(R5) ;WRITE DATA FUNCTION?
```

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

```
                53   0A   52    C3  037B    904          SUBL3   R2,#10,R3                       ; Get the number of error bits
                                     037F    905                                                  ;  remaining in the pattern.
                          09    15  037F    906          BLEQ    30$                             ; Branch if no other bits in pattern.
                          52    D6  0381    907          INCL    R2                              ; Point to next bit in pattern.
        52   00C6 C5     53    52    EF  0383    908          EXTZV   R2,R3,UCB$W_EC2(R5),R2  ; Is there more than one error bit set?
                          0C    BA  038A    909 30$:     POPR    #^M<R3,R2>                      ; Restore work registers without
                                     038C    910                                                  ;  affecting flags.
                          26    1A  038C    911          BGTRU   DEFER_ECC                       ; If more than one error bit set, don't
                                     038E    912                                                  ;  apply ECC correction.
                                     038E    913 ;
                                     038E    914 ; APPLY_ECC -
                                     038E    915 ;
                                     038E    916 ; Apply ECC correction to correct a single bit error.
                                     038E    917 ;
                                     038E    918
                                     038E    919 APPLY_ECC:
                     7E   51    3C  038E    920          MOVZWL  R1, -(SP)                       ; Save total bytes transfered, inc. ECC.
          00000000'GF     16    JSB     G^IOC$APPLYECC                  ; Apply ECC correction.
                     50 8ED0  0397    922          POPL    R0                              ; Retrieve transfered byte count.
          00000000'GF     16    JSB     G^IOC$UPDATRANSP                ; Update transfer parameters.
              00CA C5     94    CLRB    UCB$B_OFFNDX(R5)                ; Reset offset table index.
                          02    8A  03A4    925          BICB    #DR_M_OM,-                      ; Clear offset mode.
              00D4 C5           03A6    926                      UCB$B_DR_SSTS(R5)               ;
                 7E A5    B5  03A9    927          TSTW    UCB$W_BCNT(R5)                  ; Any more to transfer?
                    03    13  03AC    928          BEQL    20$                             ; If EQL no.
                  FEDF    31  03AE    929          BRW     TRANNOCH                        ; Transfer next segment.
                  FEE6    31  03B1    930 20$:     BRW     DATACHECK                       ; Check for write check.
                              03B4    931
                              03B4    932 ;
                              03B4    933 ; DEFER_ECC -
                              03B4    934 ;
                              03B4    935 ; Don't apply ECC correction for multiple bit errors unless the error cannot
                              03B4    936 ; be recovered with offset retries.
                              03B4    937 ;
                              03B4    938
                              03B4    939 DEFER_ECC:
                          10    88  03B4    940          BISB    #DR_M_ECC_DEFER,-               ; Set flag to indicate that ECC
                      00D4 C5     03B6    941                  UCB$B_DR_SSTS(R5)               ;  can be used if offset recovery fails.
                                     03B9    942
                                     03B9    943 ;
                                     03B9    944 ; OFF - OFFSET RECOVERY
                                     03B9    945 ;
                                     03B9    946 ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER CRC, OR ECC
                                     03B9    947 ; HARD ERROR IS DETECTED ON A READ FUNCTION.
                                     03B9    948 ;
                                     03B9    949
                                     03B9    950 OFF:                                             ;OFFSET RECOVERY
                          50    D5  03B9    951          TSTL    R0                              ;ANY GOOD DATA TRANSFERED?
                          33    13  03BB    952          BEQL    30$                             ;IF EQL NO
                                     03BD    953
                                     03BD    954 ;
                                     03BD    955 ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
                                     03BD    956 ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
                                     03BD    957 ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
                                     03BD    958 ; POINT OF ERROR.
                                     03BD    959 ;
                                     03BD    960
```

```
                                                                C 2
00000000'GF   16  03BD   961          JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
      7E A5   B5  03C3   962          TSTW    UCB$W_BCNT(R5)          ; Any more data to transfer?
        03    12  03C6   963          BNEQ    5$                      ; Branch if so.
       FECF   31  03C8   964          BRW     DATACHECK               ; Otherwise, go check for write check.
    00CA C5   94  03CB   965 5$:      CLRB    UCB$B_OFFNDX(R5)        ;RESET OFFSET TABLE INDEX
00CB C5   10  90  03CF   966 10$:     MOVB    #16,UCB$B_OFFRTC(R5)    ;SET OFFSET RETRY COUNT
00CA C5   04  91  03D4   967          CMPB    #OFFSIZ,UCB$B_OFFNDX(R5) ;ALL OFFSETS TRIED?
          08  12  03D9   968          BNEQ    20$                     ; Branch if not.
          04  E4  03DB   969          BBSC    #DR_V_ECC_DEFER,-       ; Correct the error with ECC if we can.
    00D4 C5       03DD   970                  UCB$B_DR_SSTS(R5),-     ;
          AD      03E0   971                  APPLY_ECC               ;
          53  11  03E1   972          BRB     90$                     ; Otherwise, fatal error.
              03E3   973 20$:     RELCHAN                             ;RELEASE CHANNEL
00D4 C5   02  8A  03E9   974          BICB    #DR_M_OM,UCB$B_DR_SSTS(R5)  ;CLEAR OFFSET MODE
          35  11  03EE   975          BRB     60$                     ;
              03F0   976
              03F0   977 ;
              03F0   978 ; NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
              03F0   979 ;
              03F0   980
52  9040 8F   B3  03F0   981 30$:     BITW    #RM_ER1_M_DCK!-         ;DATA CHECK OR,
              03F5   982                  RM_ER1_M_DTE!-          ;DRIVE TIMING OR,
              03F5   983                  RM_ER1_M_ECH,R2         ;ECC HARD ERROR?
          05  12  03F5   984          BNEQ    40$                     ;IF NEQ YES
00C9 C5   04  88  03F7   985          BISB    #RM_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;SET HEADER COMPARE INHIBIT
    00CB C5   97  03FC   986 40$:     DECB    UCB$B_OFFRTC(R5)        ;CHANGE CURRENT OFFSET?
          28  12  0400   987          BNEQ    70$                     ;IF NEQ NO
    00CA C5   96  0402   988          INCB    UCB$B_OFFNDX(R5)        ;UPDATE OFFSET TABLE INDEX
 50 00CA C5   9A  0406   989          MOVZBL  UCB$B_OFFNDX(R5),R0     ;GET NEXT OFFSET TABLE INDEX
00C8 C5 FC9D CF40   90  040B   990          MOVB    OFFTAB-1[R0],UCB$W_OFFSET(R5) ;GET NEXT OFFSET VALUE
          BA  13  0413   991          BEQL    10$                     ;IF EQL RETURN TO CENTERLINE
    00CB C5   02  90  0415   992          MOVB    #2,UCB$B_OFFRTC(R5)     ;SET OFFSET RETRY COUNT
              041A   993          RELCHAN                             ;RELEASE CHANNEL
    00D4 C5   02  88  0420   994          BISB    #DR_M_OM,UCB$B_DR_SSTS(R5)  ;SET OFFSET MODE
    00C9 C5   04  8A  0425   995 60$:     BICB    #RM_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;CLEAR HEADER COMPARE INHIBIT
03  00D4 C5   00  E0  042A   996 70$:     BBS     #DR_V_DCK,UCB$B_DR_SSTS(R5),80$ ;IF SET, DATA CHECK FUNCTION
        FE57  31  0430   997          BRW     TRANRCH                 ;TRY FUNCTION AGAIN
        FEB1  31  0433   998 80$:     BRW     CHECKRETRY              ;TRY DATA CHECK AGAIN
              0436   999
  50    04 A3   D0  0436  1000 90$:     MOVL    RM_DS(R3),R0            ;GET DRIVE STATUS
  51  00CC C5   D0  043A  1001          MOVL    UCB$L_DR_SR(R5),R1      ;GET MBA STATUS
          32  11  043F  1002          BRB     FATALERR                ;
              0441  1003
              0441  1004 ;
              0441  1005 ; RETRIABLE ERROR
              0441  1006 ;
              0441  1007
              0441  1008 RETRYERR:                                    ;RETRIABLE ERROR
          07  BB  0441  1009          PUSHR   #^M<R0,R1,R2>           ; Save error status registers.
              0443  1010          RELCHAN                             ; Release channel before possible RECAL
          07  BA  0445  1011          POPR    #^M<R0,R1,R2>           ; Restore error status registers.
04  00D0 C5   0E  E0  044B  1012          BBS     #RM_ER2_V_SKI,UCB$W_DR_ER2(R5),10$ ;IF SET, SEEK INCOMPLETE
    0D 52     07  E1  0451  1013          BBC     #RM_ER1_V_HCE,R2,20$    ;IF CLR, HEADER COMPARED
              0455  1014 10$:     EXFUNC  FATALERR,F_RECAL            ;RECALIBRATE HEADS
  52    2000 8F   3C  045D  1015          MOVZWL  #RM_ER1_M_OPI,R2        ;SET AN ERROR FOR CALLER TO SEE
        0080 C5   97  0462  1016 20$:     DECB    UCB$B_ERTCNT(R5)        ;ANY RETRIES LEFT?
          0B  13  0466  1017          BEQL    FATALERR                ;IF EQL NO
```

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

```
                50    0054 8F    3C  0506  1075          MOVZWL  #SS$_CTRLERR,R0         ;SET CONTROLLER ERROR STATUS
                                     050B  1076
                                     050B  1077 ;
                                     050B  1078 ; FUNCTION COMPLETION COMMON EXIT
                                     050B  1079 ;
                                     050B  1080
                                     050B  1081 FUNCXT:                                 ;FUNCTION EXIT
                          50    DD   050B  1082          PUSHL   R0                     ;SAVE FINAL REQUEST STATUS                    51
                 00000000'GF    16   050D  1083          JSB     G^IOC$DIAGBUFILL       ;FILL DIAGNOSTIC BUFFER IF PRESENT
                                     0513  1084          RELCHAN                        ;RELEASE CHANNEL IF OWNED
                0092 C5    0A    91   0519  1085          CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                           1A    1A   051E  1086          BGTRU   10$                    ;IF GTRU YES
                0092 C5    13    91   0520  1087          CMPB    #CDF_READPRESET,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                           13    18   0525  1088          BLEQU   10$                    ;IF LEQU YES
                0092 C5    11    91   0527  1089          CMPB    #CDF_AVAILABLE,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                           0C    13   052C  1090          BEQL    10$                    ;IF EQL YES
                  52    58 A5    D0   052E  1091          MOVL    UCB$L_IRP(R5),R2       ;RETRIEVE ADDRESS OF IRP
                     00D8 C5    A1   0532  1092          ADDW3   UCB$L_DR_BCR(R5),-
               02 AE    32 A2        0536  1093                  IRP$W_BCNT(R2),2(SP)   ; Calculate bytes transfered
                           51    D4   053A  1094 10$:     CLRL    R1                     ;CLEAR SECOND STATUS LONGWORD
                           50 8ED0    053C  1095          POPL    R0                     ;RETRIEVE FINAL REQUEST STATUS
                 53    0091 C5    9A   053F  1096          MOVZBL  UCB$B_SLAVE+1(R5),R3   ;GET DRIVE OFFSET CONSTANT
               53   0400 C443    DE   0544  1097          MOVAL   MBA$L_ERB(R4)[R3],R3   ;GET ADDRESS OF DRIVE REGISTERS
                      63    09    9A   054A  1098          MOVZBL  #F_DRVCLR!1,RM_CS1(R3) ; Issue a drive clear before release.
                      63    0B    9A   054D  1099          MOVZBL  #F_RELEASE!1,RM_CS1(R3) ;RELEASE PORT
                                     0550  1100          REQCOM                         ;COMPLETE REQUEST
```

```
0556  1102          .SBTTL  HARDWARE FUNCTION EXECUTION
0556  1103 ;
0556  1104 ; FEX - HARDWARE FUNCTION EXECUTION
0556  1105 ;
0556  1106 ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
0556  1107 ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
0556  1108 ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
0556  1109 ; INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. IF THE FUNCTION IS AN
0556  1110 ; IMMEDIATE FUNCTION CONTROL RETURNS IMMEDIATELY. ELSE THE RETURN ADDRESS
0556  1111 ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
0556  1112 ; RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
0556  1113 ;
0556  1114 ; INPUTS:
0556  1115 ;
0556  1116 ;       R0 = FUNCTION TABLE DISPATCH INDEX.
0556  1117 ;       R3 = ADDRESS OF DRIVE CONTROL STATUS REGISTER 1.
0556  1118 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
0556  1119 ;       R5 = DEVICE UNIT UCB ADDRESS.
0556  1120 ;
0556  1121 ;       00(SP) = RETURN ADDRESS OF CALLER.
0556  1122 ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
0556  1123 ;
0556  1124 ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
0556  1125 ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
0556  1126 ;
0556  1127 ; OUTPUTS:
0556  1128 ;
0556  1129 ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
0556  1130 ;
0556  1131 ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
0556  1132 ;             OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
0556  1133 ;             ERROR ROUTINE.
0556  1134 ;
0556  1135 ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
0556  1136 ;             ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
0556  1137 ;             INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
0556  1138 ;
0556  1139 ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
0556  1140 ;             OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
0556  1141 ;             IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
0556  1142 ;
0556  1143 ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
0556  1144 ;             DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
0556  1145 ;
0556  1146 ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
0556  1147 ;
0556  1148 ;       IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
0556  1149 ;       THE GENERAL REGISTERS R0, R1, AND R2, AND THE UCB.
0556  1150 ;
0556  1151 ;       R0 = DRIVE STATUS REGISTER.
0556  1152 ;       R1 = MBA STATUS REGISTER.
0556  1153 ;       R2 = DRIVE ERROR REGISTER 1.
0556  1154 ;
0556  1155 ;       UCB$W_EC1(R5) = ECC POSITION REGISTER.
0556  1156 ;       UCB$W_EC2(R5) = ECC PATTERN REGISTER.
0556  1157 ;       UCB$W_BCR(R5) = BYTE COUNT REGISTER.
0556  1158 ;       UCB$W_DR_ER2(R5) = DRIVE ERROR REGISTER 2.
```

```
                                0556  1159 ;
                                0556  1160
                                0556  1161 FEX:                                                  ;FUNCTION EXECUTOR
                     009C C5 8ED0  0556  1162          POPL    UCB$L_DPC(R5)                     ;SAVE DRIVER PC VALUE
                   0093 C5   50    90  055B  1163          MOVB    R0,UCB$B_CEX(R5)                  ;SAVE CASE INDEX
                  53   0091 C5    9A  0560  1164          MOVZBL  UCB$B_SLAVE+1(R5),R3              ;GET DRIVE OFFSET CONSTANT
               53    0400 C443    DE  0565  1165          MOVAL   @MBA$L_ERB(R4)[R3],R3             ;GET ADDRESS OF DRIVE REGISTERS
        38 A5   00008000 8F    D3  056B  1166          BITL    #DEV$M_DUA,UCB$L_DEVCHAR(R5) ;DUAL PORTED DRIVE?
                      48    12  0573  1167          BNEQ    SEIZE                             ;IF NEQ, YES
                   50   0093 C5    9A  0575  1168 GO:      MOVZBL  UCB$B_CEX(R5),R0                  ;Restore case index (func. code)
                                057A  1169          CASE    R0,<-                             ;DISPATCH TO PROPER FUNCTION ROUTINE
                                057A  1170                  POSIT,-                           ;SEEK CYLINDER
                                057A  1171                  EXFNC,-                           ;RECALIBRATE
                                057A  1172                  IMMED,-                           ;DRIVE CLEAR
                                057A  1173                  IMMED,-                           ;RELEASE DRIVE
                                057A  1174                  IMMED,-                           ;OFFSET HEADS
                                057A  1175                  EXFNC,-                           ;RETURN TO CENTERLINE
                                057A  1176                  IMMED,-                           ;PACK ACKNOWLEDGE
                                057A  1177                  POSIT,-                           ;SEARCH FOR SECTOR
                                057A  1178                  XFER,-                            ;WRITE CHECK
                                057A  1179                  XFER,-                            ;WRITE DATA
                                057A  1180                  XFER,-                            ;READ DATA
                                057A  1181                  XFER,-                            ;WRITE HEADER AND DATA
                                057A  1182                  XFER,-                            ;READ HEADER AND DATA
                                057A  1183                  XFER,-                            ;WRITE TRACK DESCRIPTOR
                                057A  1184                  XFER,-                            ;READ TRACK DESCRIPTOR
                                057A  1185                  IMMED,-                           ;AVAILABLE
                                057A  1186                  XFER,-                            ;WRITE CHECK HEADER AND DATA
                                057A  1187                  IMMED,-                           ;READIN PRESET
                                057A  1188                  EXFNC,-                           ;DIAGNOSE
                                057A  1189                  SEARCHA,-                         ;SEARCH AHEAD
                                057A  1190                  >,LIMIT=#CDF_SEEK                 ;
                                05A6  1191
                                05A6  1192 ;
                                05A6  1193 ; IMMEDIATE FUNCTION EXECUTION
                                05A6  1194 ;
                                05A6  1195 ;       FUNCTIONS INCLUDE:
                                05A6  1196 ;
                                05A6  1197 ;               NO OPERATION,
                                05A6  1198 ;               DRIVE CLEAR,
                                05A6  1199 ;               RELEASE PORT,
                                05A6  1200 ;               OFFSET,
                                05A6  1201 ;               READ IN PRESET, AND
                                05A6  1202 ;               PACK ACKNOWLEDGE.
                                05A6  1203 ;
                                05A6  1204 ; Two other functions which might (but hopefully don't) pass through this code
                                05A6  1205 ; are UNLOAD and AVAILABLE.  If such functions get here they are treated as
                                05A6  1206 ; NOPs.
                                05A6  1207 ;
                                05A6  1208 ; THESE FUNCTIONS ARE EXECUTED IMMEDIATELY AND THE FINAL DEVICE REGISTERS
                                05A6  1209 ; ARE RETURNED TO THE CALLER.
                                05A6  1210 ;
                                05A6  1211
                                05A6  1212 IMMED:                                                 ;IMMEDIATE FUNCTION EXECUTION
                                05A6  1213          DSBINT                                        ;DISABLE INTERRUPTS
                  09 64 A5    05    E0  05AC  1214          BBS     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF SET, POWER HAS FAILED
                        63    09    9A  05B1  1215          MOVZBL  #F_DRVCLR!1,RM_CS1(R3)        ;CLEAR DRIVE ERRORS
```

```
                63    FADF CF40    9A  05B4  1216          MOVZBL  FTAB[R0],RM_CS1(R3)      ;EXECUTE FUNCTION
                           010B    31  05BA  1217 10$:     BRW     ENBXIT                  ;
                                       05BD  1218
                                       05BD  1219 ;
                                       05BD  1220 ; ATTEMPT TO SEIZE THE PORT ON A DUAL PORTED DISK.
                                       05BD  1221 ;
                                       05BD  1222
             B2 00D4 C5      03    E1  05BD  1223 SEIZE:  BBC     #DR_V_DUALPORT,-        ; IF CLEAR, DUALPORT KIT
                                       05C3  1224                 UCB$B_DR_SSTS(R5),GO    ;IS NOT PRESENT
             51    00000064 8F     D0  05C3  1225          MOVL    #100,R1                 ; Initialize count for the number of
                                       05CA  1226                                         ;  times we will accept the loss of
                                       05CA  1227                                         ;  the port while we are on the I/O
                                       05CA  1228                                         ;  fork queue.
                                       05CA  1229 2$:      DSBINT                          ;DISABLE INTERRUPTS
                           04 A3       D4  05D0  1230          CLRL    RM_DS(R3)           ;ATTEMPT TO SEIZE PORT
                00000100 8F     04 A3   D3  05D3  1231          BITL    #RM_DS_M_DPR,-      ;DID WE SEIZE THE PORT?
                                       05D9  1232                 RM_DS(R3)
                              16       12  05DB  1233          BNEQ    4$                  ;IF NEQ, WE SEIZED THE PORT
                                       05DD  1234          WFIKPCH RETREG,#15              ;LETS WAIT FOR THE PORT, ELSE TIMEOUT
                                       05E7  1235          IOFORK                          ;CREATE FORK PROCESS
                           DA 51       F4  05ED  1236          SOBGEQ  R1,2$               ; Loop to make sure we really still
                                       05F0  1237                                         ;  have the port after we are dequeued
                                       05F0  1238                                         ;  off the I/O fork queue.
                            00D8       31  05F0  1239          BRW     RETREG              ; Otherwise, error - We keep losing the
                                       05F3  1240                                         ;  port and we've retried enough.
                                       05F3  1241 4$:      ENBINT                          ;ENABLE INTERRUPTS
                            FF7C       31  05F6  1242          BRW     GO                  ;LET'S CONTINUE, WE HAVE THE PORT
                                       05F9  1243
                                       05F9  1244 ;
                                       05F9  1245 ; SEARCH AHEAD FUNCTION EXECUTION
                                       05F9  1246 ;
                                       05F9  1247 ; THIS FUNCTION MINIMIZES ROTATIONAL LATENCY BY SEARCHING FOR THE SECTOR THAT IS
                                       05F9  1248 ; FOUR SECTORS AHEAD OF THE STARTING SECTOR OF A TRANSFER.
                                       05F9  1249 ;
                                       05F9  1250 ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                                       05F9  1251 ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                                       05F9  1252 ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                                       05F9  1253 ;
                                       05F9  1254
                                       05F9  1255 SEARCHA:                                ;
           51    00BC C5      3C      05F9  1256          MOVZWL  UCB$W_DA(R5),R1         ;GET DESIRED TRACK AND SECTOR ADDRESS
                51    04     82      05FE  1257          SUBB    #4,R1                   ;COMPUTE FOUR SECTORS BEFORE IT
                      04     18      0601  1258          BGEQ    10$                     ;IF GEQ BEFORE SECTOR ZERO
             51    44 A5     80      0603  1259          ADDB    UCB$B_SECTORS(R5),R1    ;CONVERT TO AFTER SECTOR ZERO
          14 A3    51     D0      0607  1260 10$:     MOVL    R1,RM_DA(R3)            ;SET TRACK AND SECTOR ADDRESS
                   15     11      060B  1261          BRB     LDCYL                   ;
                                       060D  1262
                                       060D  1263 ;
                                       060D  1264 ; TRANSFER FUNCTION EXECUTION
                                       060D  1265 ;
                                       060D  1266 ;       FUNCTIONS INCLUDE:
                                       060D  1267 ;
                                       060D  1268 ;               WRITE TRACK DESCRIPTOR,
                                       060D  1269 ;               WRITE CHECK,
                                       060D  1270 ;               WRITE CHECK HEADER AND DATA,
                                       060D  1271 ;               WRITE DATA,
                                       060D  1272 ;               WRITE HEADER AND DATA,
```

```
                     060D  1273 ;                  READ TRACK DESCRIPTOR,
                     060D  1274 ;                  READ DATA, AND
                     060D  1275 ;                  READ HEADER AND DATA.
                     060D  1276 ;
                     060D  1277 ; THE MAP REGISTERS, BYTE COUNT REGISTER, AND VIRTUAL ADDRESS REGISTER ARE
                     060D  1278 ; LOADED FOLLOWED BY THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS.
                     060D  1279 ; THE FUNCTION IS INITIATED AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
                     060D  1280 ; INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                     060D  1281 ;
                     060D  1282 ; IT ASSUMED THAT THE CALLER OWNS THE CHANNEL ON WHICH THE I/O IS TO OCCUR.
                     060D  1283 ;
                     060D  1284
                     060D  1285 XFER:                                        ;TRANSFER FUNCTION EXECUTION
      08 A4    00    D2  060D  1286          MCOML   #0,MBA$L_SR(R4)         ;CLEAR MASSBUS ADAPTER ERRORS
                     0611  1287          LOADMBA                         ;LOAD MAP, BYTE COUNT, AND VIRTUAL ADDRESS
   50   0093 C5    9A  0617  1288          MOVZBL  UCB$B_CEX(R5),R0        ;RETRIEVE FUNCTION TABLE INDEX
                     061C  1289
                     061C  1290 ;
                     061C  1291 ; POSITIONING FUNCTION EXECUTION
                     061C  1292 ;
                     061C  1293 ;          FUNCTIONS INCLUDE:
                     061C  1294 ;
                     061C  1295 ;                  SEEK CYLINDER, AND
                     061C  1296 ;                  SEARCH FOR SECTOR.
                     061C  1297 ;
                     061C  1298 ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                     061C  1299 ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                     061C  1300 ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                     061C  1301 ;
                     061C  1302
                     061C  1303 POSIT:                                       ;POSITION FUNCTION EXECUTION
   14 A3    00BC C5    3C  061C  1304          MOVZWL  UCB$W_DA(R5),RM_DA(R3)  ;SET DESIRED TRACK AND SECTOR ADDRESS
                     0622  1305 LDCYL:                                       ;
   28 A3    00BE C5    3C  0622  1306          MOVZWL  UCB$W_DC(R5),RM_DC(R3)  ;SET DESIRED CYLINDER ADDRESS
                     0628  1307
                     0628  1308 ;
                     0628  1309 ; INTERRUPT WAIT FUNCTION EXECUTION
                     0628  1310 ;
                     0628  1311 ;          FUNCTIONS INCLUDE:
                     0628  1312 ;
                     0628  1313 ;                  DIAGNOSE,
                     0628  1314 ;                  RECALIBRATE, AND
                     0628  1315 ;                  RETURN TO CENTERLINE.
                     0628  1316 ;
                     0628  1317 ; THE OFFSET REGISTER IS LOADED, THE FUNCTION IS INITIATED, AND A WAITFOR
                     0628  1318 ; INTERRUPT IS EXECUTED. WHEN THE INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS
                     0628  1319 ; ARE RETURNED TO THE CALLER.
                     0628  1320 ;
                     0628  1321
                     0628  1322 EXFNC:                                       ;EXECUTE FUNCTION
      63    09    9A  0628  1323          MOVZBL  #F_DRVCLR!1,RM_CS1(R3)  ;CLEAR DRIVE ERRORS
   30 00D4 C5    01    E1  062B  1324          BBC     #DR_V_OM,UCB$B_DR_STS(R5),10$  ;IF CLR, NO OFFSET ACTIVE
      63    0D    9A  0631  1325          MOVZBL  #F_OFFSET!1,RM_CST(R3)  ;SET DRIVE IN OFFSET MODE
            07    91  0634  1326          CMPB    #DT$_RP07,-             ; Is this drive an RP07?
         41 A5          0636  1327                  UCB$B_DEVTYPE(R5)       ;
            27    12  0638  1328          BNEQ    10$                     ; If not, no need to wait.
                     063A  1329          TIMEWAIT -                      ; Wait for 5 milliseconds or until
```

```
                     063A  1330                 TIME = #5000,-          ;  go bit clears.
                     063A  1331                 BITVAL = #1,-           ;
                     063A  1332                 SOURCE = RM_CS1(R3),-   ;
                     063A  1333                 CONTEXT = L,-           ;
                     063A  1334                 SENSE = .FALSE.         ;
                     0661  1335 10$:     DSBINT                         ;DISABLE INTERRUPTS
  5C 64 A5    05     E0  0667  1336           BBS     #UCB$V_POWER,UCB$W_STS(R5),ENBXIT ;IF SET, POWER FAILED
0C A3    00D2 C5     3C  066C  1337           MOVZWL  UCB$W_DR_MR(R5),RM_MR(R3)       ;SET DIAGNOSTIC INFO
24 A3    00C8 C5     3C  0672  1338           MOVZWL  UCB$W_OFFSET(R5),RM_OF(R3) ;SET FORMAT, INHIBIT BITS, AND OFFSET
                  09     E1  0678  1339           BBC     #RM_OF_V_SSEI,-         ; If Skip Sector Error Inhibit is clear
      0F 00C8 C5              067A  1340                   UCB$W_OFFSET(R5),12$    ;  branch around.
                  09     E0  067E  1341           BBS     #IO$V_SKPSECINH,-       ; If reason for SSEI is EXPLICIT user
      09 009A C5              0680  1342                   UCB$W_FUNC(R5),12$      ;  request, then branch around.
                  02     8A  0684  1343           BICB    #RM_OF_M_SSEI/256,-     ; If here, SSEI set due to error on
         00C9 C5              0686  1344                   UCB$W_OFFSET+1(R5)      ;  this track. Hence clear SSEI in
         00BC C5     97       0689  1345           DECB    UCB$W_DA(R5)            ;  software only and decrement sector
                              068D  1346                                           ;  bias in software also.
                              068D  1347 12$:
52    04 A3    13     78  068D  1348           ASHL    #31-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
                  1E     18  0692  1349           BGEQ    15$                     ;IF GEQ NO
 63    F9FF  CF40     9A  0694  1350           MOVZBL  FTAB[R0],RM_CS1(R3)     ;INITIATE FUNCTION
                              069A  1351           WFIKPCH RETREG,#6               ;WAITFOR INTERRUPT AND KEEP CHANNEL
00CC C5    08 A4     D0  06A4  1352           MOVL    MBA$L_SR(R4),UCB$L_DR_SR(R5) ;SAVE FINAL CONTROLLER STATUS
                              06AA  1353           IOFORK                          ;CREATE FORK PROCESS
                  19     11  06B0  1354           BRB     RETREG                  ;
                              06B2  1355
                              06B2  1356 ;
                              06B2  1357 ; MEDIUM OFFLINE AT START OF FUNCTION
                              06B2  1358 ;
                              06B2  1359
                              06B2  1360 15$:     ENBINT                          ;ENABLE INTERRUPTS
      0093 C5     94  06B5  1361           CLRB    UCB$B_CEX(R5)           ;FORCE DRIVE FUNCTION
   50 4000 8F     3C  06B9  1362           MOVZWL  #RM_DS_M_ERR,R0         ;SET DRIVE ERROR
      00CC C5     D4  06BE  1363           CLRL    UCB$L_DR_SR(R5)         ;CLEAR SAVED MBA STATUS REGISTER
      00D5 C5     96  06C2  1364           INCB    UCB$B_DR_ERL(R5)        ;SET MEDIUM OFFLINE FOR ERROR LOG
                  35     11  06C6  1365           BRB     ERROR                   ;
                              06C8  1366
                              06C8  1367 ;
                              06C8  1368 ; ENABLE INTERRUPTS
                              06C8  1369 ;
                              06C8  1370
                              06C8  1371 ENBXIT:                              ;
                              06C8  1372           ENBINT                          ;ENABLE INTERRUPTS
                              06CB  1373
                              06CB  1374 ;
                              06CB  1375 ; RETURN REGISTERS
                              06CB  1376 ;
                              06CB  1377
                              06CB  1378 RETREG:                              ;RETURN FINAL DEVICE REGISTERS
00D0 C5    34 A3     F7  06CB  1379           CVTLW   RM_ER2(R3),UCB$W_DR_ER2(R5) ;SAVE ERROR REGISTER 2
00C4 C5    38 A3     F7  06D1  1380           CVTLW   RM_EC1(R3),UCB$W_ECT(R5) ;SAVE ECC POSITION REGISTER
00C6 C5    3C A3     F7  06D7  1381           CVTLW   RM_EC2(R3),UCB$W_EC2(R5) ;SAVE ECC PATTERN REGISTER
                              06DD  1382 ; Here we save the more conservative of the two byte counts contained in
                              06DD  1383 ;       the MBA$L_BCR register.  The high word of this register is the
                              06DD  1384 ;       (negative of the) number of bytes transferred to or from the
                              06DD  1385 ;       drive, while the low word is the (negative of the) number of
                              06DD  1386 ;       bytes transferred to or memory.  On a read, the more conservative
```

```
                     06DD  1387 ;       value is that of the number of bytes transferred to memory (low word)
                     06DD  1388 ;       while on a write the more conservative value is the number of
                     06DD  1389 ;       bytes transferred to the drive (high word).  Here we deposit
                     06DD  1390 ;       the entire register into a longword in the UCB.  If the operation
                     06DD  1391 ;       was a read we leave the value as is.  However if the operation
                     06DD  1392 ;       was a write (or anything but a read) we move the high word to
                     06DD  1393 ;       the low word in memory.  All other pieces of this driver use the
                     06DD  1394 ;       low word of this longword as the valid byte count.
                     06DD  1395
          10 A4   D0 06DD  1396         MOVL    MBA$L_BCR(R4),-         ; Save entire byte count register
        00D8 C5      06E0  1397                 UCB$L_DR_BCR(R5)        ;  in the UCB.
     50   58 A5   D0 06E3  1398         MOVL    UCB$L_IRP(R5),R0        ; Retrieve IRP pointer.
             01   E0 06E7  1399         BBS     #IRP$V_FUNC,-           ; If we had a read operation then
      07 2A A0      06E9  1400                 IRP$W_STS(R0),5$        ;  just branch around since all OK.
        00DA C5   B0 06EC  1401         MOVW    UCB$L_DR_BCR+2(R5),-    ; If NOT read, then copy high word to
        00D8 C5      06F0  1402                 UCB$L_DR_BCR(R5)        ;  low order word for later use.
                     06F3  1403 5$:
00D6 C5   24 A3   F7 06F3  1404         CVTLW   RM_OF(R3),UCB$W_DR_OFR(R5)  ;SAVE OFFSET REGISTER
     50   04 A3   D0 06F9  1405         MOVL    RM_DS(R3),R0            ;GET CONTENTS OF DRIVE STATUS REGISTER
                     06FD  1406 ERROR:                                   ;
   51   00CC C5   D0 06FD  1407         MOVL    UCB$L_DR_SR(R5),R1      ;RETRIEVE FINAL CONTROLLER STATUS
     52   08 A3   D0 0702  1408         MOVL    RM_ERT(R3),R2           ;GET CONTENTS OF DRIVE ERROR REGISTER 1
64 A5   0060 8F   B3 0706  1409         BITW    #UCB$M_POWER!-          ;POWER FAIL OR DEVICE TIMEOUT?
                     070C  1410                 UCB$M_TIMOUT,UCB$W_STS(R5) ;
            03   13 070C  1411         BEQL    10$                     ;IF EQL - NO
          0137   31 070E  1412         BRW     SPECOND                 ;BRANCH TO SPECIAL CONDITION
   0093 C5   15   91 0711  1413 10$:    CMPB    #CDF_SEARCHA, UCB$B_CEX(R5) ;Search Ahead?
             5F   13 0716  1414         BEQL    SAFUNC                  ;Branch to special search-ahead code.
   0093 C5   0A   91 0718  1415         CMPB    #CDF_WRITECHECK,UCB$B_CEX(R5)  ;DRIVE RELATED FUNCTION?
             07   1A 071D  1416         BGTRU   20$                     ;IF GTRU - YES
   0093 C5   13   91 071F  1417         CMPB    #CDF_READPRESET,UCB$B_CEX(R5) ;DRIVE RELATED FUNCTION?
             03   1A 0724  1418         BGTRU   30$                     ;IF GTRU - NO
          0083   31 0726  1419 20$:     BRW     DFUNC                   ;DRIVE FUNCTION
                     0729  1420
                     0729  1421 ;
                     0729  1422 ; CHECK FOR RM80 SKIP SECTOR ERROR
                     0729  1423 ;
                     0729  1424
     41 A5   0D   91 0729  1425 30$:     CMPB    #DT$_RM80,UCB$B_DEVTYPE(R5)  ;RM80?
             56   12 072D  1426         BNEQ    CFUNC                   ;IF NEQ - NO
             05   E1 072F  1427         BBC     #RM_ER2_V_SSE,-         ; If clear, No Skip Sectoring error,
      50 00D0 C5      0731  1428                 UCB$W_DR_ER2(R5),CFUNC  ;  so branch around.
   52   0180 8F   B3 0735  1429         BITW    #RM_ER1_M_HCE!RM_ER1_M_HCRC,R2  ;HEADER COMPARE OR HDR CRC ERR?
             49   12 073A  1430         BNEQ    CFUNC                   ;IF NEQ - YES
   00C9 C5   02   88 073C  1431         BISB    #RM_OF_M_SSEI/256,UCB$W_OFFSET+1(R5)  ;SET SKIP SECTOR INHIBIT
             50   DD 0741  1432         PUSHL   R0                      ;SAVE R0 (DRIVE STATUS REGISTER)
   50   00D8 C5   3C 0743  1433         MOVZWL  UCB$L_DR_BCR(R5),R0     ; Get negative bytes remaining
             05   12 0748  1434         BNEQ    40$                     ;IF NEQ - PARTIAL TRANSFER
   50   FFFF 8F   B0 074A  1435         MOVW    #-1,R0                  ;FAKE A PARTIAL TRANSFER
     50   7E A5   A0 074F  1436 40$:     ADDW    UCB$W_BCNT(R5),R0       ;CALCULATE BYTES TRANSFERRED
   50   01FF 8F   AA 0753  1437         BICW    #^X1FF,R0               ;TRUNCATE TO LAST BLOCK TRANSFERRED
    00000000'GF   16 0758  1438         JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
        00BC C5   96 075E  1439         INCB    UCB$W_DA(R5)            ;INCREMENT TO NEXT SECTOR
             50 8ED0 0762  1440         POPL    R0                      ;RESTORE R0 (DRIVE STATUS REGISTER)
03 00D4 C5   00   E0 0765  1441         BBS     #DR_V_DCK,UCB$B_DR_SSTS(R5),45$  ;IF SET - DATACHECK IN PROGRESS
          FB22   31 076B  1442         BRW     TRANNOCH                ;RESTART TRANSFER
                     076E  1443 45$:     RELCHAN                         ;RELEASE CHANNEL
```

```
                FB70   31  0774  1444          BRW     CHECKRETRY              ;RESTART DATA CHECK
                                                                               ;NO SSE - CHECK FOR TRACK-TRACK SSEI CLR
                                                .ENABL  LSB
                                                ;
                                                ; SEARCH AHEAD ERROR CHECKING
                                                ;
                                                SAFUNC:
 7A 00D0 C5    0E   E1  0777  1453          BBC     #RM_ER2_V_SKI, -        ;The only error worth checking on
                                                     UCB$W_DR_ER2(R5), 30$   ;seach-ahead is seek incomplete.
     00000000'GF   16  077D  1455          JSB     G^ERL$DEVICERR          ;SKI errors, however, must be logged
                   68  11  0783  1456          BRB     25$                     ;and retried.
                                                ;
                                                ; CONTROLLER RELATED FUNCTION
                                                ;
                                                CFUNC:
 51   000E5FFF 8F   D3  0785  1463          BITL    #MBA$M_ERROR,R1         ;ANY CONTROLLER ERRORS?
                   69  13  078C  1464          BEQL    30$                     ;IF EQL NO
     00000000'GF   16  078E  1465          JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
  66 009A C5    0F   E0  0794  1466          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),40$ ;IF SET, RETRY INHIBITED
 51   00080008 8F   D3  079A  1467          BITL    #MBA$M_SR_ERCONF!-      ;ERROR CONFIRMATION OR,
                                                     MBA$M_SR_ISTO!-         ;INTERFACE SEQUENCE TIMEOUT OR,
                                                     MBA$M_SR_PGE!-          ;PROGRAMMING ERROR OR,
                                                     MBA$M_SR_RDTO,R1        ;READ TIMEOUT?
                   5D  12  07A1  1471          BNEQ    40$                     ;IF NEQ YES - FATAL CONTROLLER ERROR
 51   00064FF4 8F   D3  07A3  1472          BITL    #MBA$M_SR_DLT!-         ;DATA LATE OR,
                                                     MBA$M_SR_INVMAP!-       ;INVALID MAP REGISTER OR,
                                                     MBA$M_SR_MAPPE!-        ;MAP REGISTER PARITY ERROR OR,
                                                     MBA$M_SR_MBEXC!-        ;MASSBUS EXCEPTION OR,
                                                     MBA$M_SR_MCPE!-         ;MASSBUS CONTROL PARITY ERROR OR,
                                                     MBA$M_SR_SPE!-          ;MBA SILO PARITY ERROR OR,
                                                     MBA$M_SR_MDPE!-         ;MASSBUS DATA PARITY ERROR OR,
                                                     MBA$M_SR_MXF!-          ;MISSED TRANSFER OR,
                                                     MBA$M_SR_NED!-          ;NONEXISTENT DRIVE OR,
                                                     MBA$M_SR_RDS!-          ;READ DATA SUBSTITUTE OR,
                                                     MBA$M_SR_WCKLWR!-       ;WRITE CHECK LOWER BYTE OR,
                                                     MBA$M_SR_WCKUPR,R1      ;WRITE CHECK UPPER BYTE?
                   1B  12  07AA  1484          BNEQ    20$                     ;IF NEQ YES - RETRIABLE CONTROLLER ERROR
                                                ;
                                                ; DRIVE RELATED FUNCTION
                                                ;
                                                DFUNC:
     47 50    0E   E1  07AC  1491  10$:      BBC     #RM_DS_V_ERR,R0,30$     ;IF CLR, NO DRIVE ERRORS
          7E A5   AE  07B0  1492          MNEGW   UCB$W_BCNT(R5),-
              00D8 C5  07B3  1493                  UCB$L_DR_BCR(R5)        ; Reset byte count - NO TRANSFER
     45 00D5 C5    E8  07B6  1494          BLBS    UCB$B_DR_ERL(R5),40$    ; Don't log error if Medium offline at
                                                                           ;  start of function.
     00000000'GF   16  07BB  1496          JSB     G^ERL$DEVICERR          ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
 39 009A C5    0F   E0  07C1  1497          BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),40$ ;IF SET, RETRY INHIBITED
     35 50    0C   E1  07C7  1498  20$:      BBC     #RM_DS_V_MOL,R0,40$     ;IF CLR, MEDIUM OFFLINE
     31 50    06   E1  07CB  1499          BBC     #RM_DS_V_VV,R0,40$      ;IF CLR, INVALID VOLUME
   52   0180 8F   B3  07CF  1500          BITW    #RM_ERT_M_HCRC!-        ; Check HCRC and HCE before checking
```

```
                     07D4  1501                 RM_ER1_M_HCE,R2          ;  BSE and FER.
                17   12  07D4  1502          BNEQ    25$                 ; NEQ means HCRC or HCE is set.
          52    0E17 8F  B3  07D6  1503          BITW    #RM_ER1_M_AOE!-     ;ADDRESS OVERFLOW OR,
                           07DB  1504                  RM_ER1_M_FER!-      ;FORMAT ERROR OR,
                           07DB  1505                  RM_ER1_M_IAE!-      ;INVALID ADDRESS OR,
                           07DB  1506                  RM_ER1_M_ILF!-      ;ILLEGAL FUNCTION OR,
                           07DB  1507                  RM_ER1_M_ILR!-      ;ILLEGAL REGISTER OR,
                           07DB  1508                  RM_ER1_M_RMR!-      ;REGISTER MODIFY REFUSE OR,
                           07DB  1509                  RM_ER1_M_WLE,R2     ;WRITE LOCK ERROR?
                23   12  07DB  1510          BNEQ    40$                 ;IF NEQ YES - FATAL DRIVE ERROR
          00D0  C5  A000 8F  B3  07DD  1511          BITW    #RM_ER2_M_BSE!-     ;BAD SECTOR ERROR OR,
                           07E4  1512                  RM_ER2_M_OPE,UCB$W_DR_ER2(R5) ;OPERATOR PLUG ERROR?
                1A   12  07E4  1513          BNEQ    40$                 ;IF NEQ YES - FATAL DRIVE ERROR
          52    4000 8F  B3  07E6  1514          BITW    #RM_ER1_M_UNS,R2    ; Is the drive unsafe?
                16   12  07EB  1515          BNEQ    45$                 ; Branch if so.
                           07ED  1516
                           07ED  1517 ;
                           07ED  1518 ; RETRIABLE ERROR EXIT
                           07ED  1519 ;
                           07ED  1520
          7E    009C D5  32  07ED  1521 25$:    CVTWL   @UCB$L_DPC(R5),-(SP)    ;GET BRANCH DISPLACEMENT
          009C  C5   8E  C0  07F2  1522          ADDL    (SP)+,UCB$L_DPC(R5)     ;CALCULATE RETURN ADDRESS - 2
          009C  C5   02  C0  07F7  1523 30$:     ADDL    #2,UCB$L_DPC(R5)        ;SKIP PAST BRANCH DISPLACEMENT WORD
                009C D5  17  07FC  1524          JMP     @UCB$L_DPC(R5)          ;RETURN TO DRIVER
                           0800  1525
                           0800  1526 ;
                           0800  1527 ; FATAL CONTROLLER OR DRIVE ERROR EXIT
                           0800  1528 ;
                           0800  1529
                     FC70  31  0800  1530 40$:     BRW     FATALERR            ;
                           0803  1531
                           0803  1532 ;
                           0803  1533 ; Check for unsafe condition and attempt to clear it.
                           0803  1534 ;
                           0803  1535
                           0803  1536 45$:     DSBINT                          ; Disable interupts.
                05   E1    0809  1537          BBC     #UCB$V_POWER,-          ; Branch if no power failure occured.
          03 64 A5         080B  1538                  UCB$W_STS(R5),47$       ;
                FEB7  31   080E  1539          BRW     ENBXIT                  ; Otherwise, enable interupts and
                           0811  1540                                          ;  go process error.
           63   09   9A    0811  1541 47$:     MOVZBL  #F_DRVCLR!1,RM_CS1(R3)  ; Attempt to clear unsafe condition.
                           0814  1542          TIMEWAIT -                      ; Wait for ten microseconds or until
                           0814  1543                  TIME = #1,-             ;  unsafe condition clears.
                           0814  1544                  BITVAL = #RM_ER1_M_UNS,-;
                           0814  1545                  SOURCE = RM_ER1(R3),-   ;
                           0814  1546                  CONTEXT = L,-           ;
                           0814  1547                  SENSE = .FALSE.         ;
                           083C  1548          ENBINT                          ; Enable interupts.
          52    08 A3   D0  083F  1549          MOVL    RM_ER1(R3),R2           ; Retrieve error status.
                A7 50   E8  0843  1550          BLBS    R0,25$                  ; Branch if drive is no longer unsafe.
                   B8   11  0846  1551          BRB     40$                     ; Otherwise, fatal error.
                           0848  1552
                           0848  1553 ;
                           0848  1554 ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
                           0848  1555 ;
                           0848  1556
                           0848  1557 SPECOND:
```

```
                61 64 A5    05    E4  0848  1558 50$:    BBSC    #UCB$V_POWER,UCB$W_STS(R5),70$ ;IF SET, POWER FAILURE
                                      084D  1559
                                      084D  1560 ;
                                      084D  1561 ; DEVICE TIME OUT
                                      084D  1562 ;
                                      084D  1563
                  00000000'GF    16   084D  1564         JSB     G^ERL$DEVICTMO          ;LOG DEVICE TIME OUT
                   53    24 A5   D0   0853  1565         MOVL    UCB$L_CRB(R5),R3        ;GET ADDRESS OF CRB
                   53    2C A3   D0   0857  1566         MOVL    CRB$L_INTD+VEC$L_IDB(R3),R3 ;GET ADDRESS OF IDB
                   04 A3    55   D1   085B  1567         CMPL    R5,IDB$L_OWNER(R3)      ;DEVICE OWN CONTROLLER?
                            22   12   085F  1568         BNEQ    60$                     ;IF NEQ NO
                                      0861  1569         DSBINT                          ;DISABLE INTERRUPTS
                            06   D0   0867  1570         MOVL    #MBA$M_CR_ABORT!MBA$M_CR_IE,- ;ABORT THE DATA TRANSFER
                         04 A4         0869  1571                 MBA$L_CR(R4)            ;
                                      086B  1572         WFIKPCH 55$,#T5                 ;WAIT FOR ABORT AND KEEP CHANNEL
                                      0875  1573         IOFORK                          ;CREATE FORK PROCESS
                                      087B  1574 55$:
                   04 A4    01   D0   087B  1575         MOVL    #MBA$M_CR_INIT,MBA$L_CR(R4) ;INITIALIZE ENTIRE MBA
                   04 A4    04   D0   087F  1576         MOVL    #MBA$M_CR_IE,MBA$L_CR(R4) ;ENABLE DEVICE INTERRUPTS
                                      0883  1577 60$:     SETIPL  UCB$B_FIPL(R5)          ;LOWER TO FORK LEVEL
                   50    022C 8F    3C   0887  1578         MOVZWL  #SS$_TIMEOUT,R0         ;SET DEVICE TIMEOUT STATUS
                         0080 C5    97   088C  1579         DECB    UCB$B_ERTCNT(R5)        ;ANY ERROR RETRIES REMAINING?
                              0F    13   0890  1580         BEQL    RESETXFR                ;IF EQL NO
                                      0892  1581         RELCHAN                         ;RELEASE CHANNEL IF OWNED
                64 A5    0040 8F    AA   0898  1582         BICW    #UCB$M_TIMOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
                            F919    31   089E  1583         BRW     FDISPATCH               ;
                                      08A1  1584
                                      08A1  1585 ;
                                      08A1  1586 ; RESET TRANSFER BYTE COUNT TO ZERO
                                      08A1  1587 ;
                                      08A1  1588
                                      08A1  1589 RESETXFR:                               ;
                   53    58 A5    D0   08A1  1590         MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF I/O PACKET
                         32 A3    AE   08A5  1591         MNEGW   IRP$W_BCNT(R3),-
                      00D8 C5         08A8  1592                 UCB$L_DR_BCR(R5)        ; Reset transfer byte count
                            FC5D    31   08AB  1593         BRW     FUNCXT                  ;
                                      08AE  1594
                                      08AE  1595 ;
                                      08AE  1596 ; POWER FAILURE
                                      08AE  1597 ;
                                      08AE  1598
                                      08AE  1599 70$:     RELCHAN                         ;RELEASE CHANNEL
                   53    58 A5    D0   08B4  1600         MOVL    UCB$L_IRP(R5),R3        ;RETRIEVE ADDRESS OF I/O PACKET
                78 A5    2C A3    7D   08B8  1601         MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
                            F886    31   08BD  1602         BRW     DR_STARTIO              ;
                                      08C0  1603         .DSABL  LSB
                                      08C0  1604
```

```
                                08C0  1606          .SBTTL  REGISTER DUMP ROUTINE
                                08C0  1607 ;
                                08C0  1608 ; DR_REGDUMP - REGISTER DUMP ROUTINE
                                08C0  1609 ;
                                08C0  1610 ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                                08C0  1611 ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
                                08C0  1612 ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                                08C0  1613 ;
                                08C0  1614 ; INPUTS:
                                08C0  1615 ;
                                08C0  1616 ;       R0 = ADDRESS OF REGISTER SAVE BUFFER.
                                08C0  1617 ;       R4 = ADDRESS OF ADAPTER CONFIGURATION REGISTER.
                                08C0  1618 ;       R5 = DEVICE UNIT UCB ADDRESS.
                                08C0  1619 ;
                                08C0  1620 ; OUTPUTS:
                                08C0  1621 ;
                                08C0  1622 ;       THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                                08C0  1623 ;
                                08C0  1624
                                08C0  1625 DR_REGDUMP:                                  ;REGISTER DUMP ROUTINE
            80    18    D0    08C0  1626          MOVL    #<RM_EC2+4+MBA$L_BCR+4+8+4>/4,(R0)+ ;INSERT NUMBER OF DEVICE REGS
            80    64    D0    08C3  1627          MOVL    MBA$L_CSR(R4),(R0)+     ;SAVE CONFIGURATION REGISTER
         80    04 A4    D0    08C6  1628          MOVL    MBA$L_CR(R4),(R0)+      ;SAVE CONTROL REGISTER
      80    00CC C5    D0    08CA  1629          MOVL    UCB$L_DR_SR(R5),(R0)+   ;SAVE STATUS REGISTER
         80    0C A4    D0    08CF  1630          MOVL    MBA$L_VAR(R4),(R0)+     ;SAVE VIRTUAL ADDRESS REGISTER
         80    10 A4    D0    08D3  1631          MOVL    MBA$L_BCR(R4),(R0)+     ;SAVE BYTE COUNT REGISTER
   51 F8 A0    08    09    EF    08D7  1632          EXTZV   #9,#8,-8(R0),R1         ;GET FINAL MAP REGISTER NUMBER
      80    0800 C441    D0    08DD  1633          MOVL    MBA$L_MAP(R4)[R1],(R0)+ ;SAVE FINAL MAP REGISTER CONTENTS
               80    D4    08E3  1634          CLRL    (R0)+                   ;ASSUME NO PREVIOUS MAP REGISTER
               51    D7    08E5  1635          DECL    R1                      ;CALCULATE PREVIOUS MAP REGISTER NUMBER
               07    19    08E7  1636          BLSS    10$                     ;IF LSS NONE
FC A0    0800 C441    D0    08E9  1637          MOVL    MBA$L_MAP(R4)[R1],-4(R0) ;SAVE PREVIOUS MAP REGISTER CONTENTS
         51    10    9A    08F0  1638 10$:     MOVZBL  #<RM_EC2+4>/4,R1        ;SET NUMBER OF DRIVE REGISTERS TO SAVE
   52    0091 C5    9A    08F3  1639          MOVZBL  UCB$B_SLAVE+1(R5),R2    ;GET DRIVE OFFSET CONSTANT
52    0400 C442    DE    08F8  1640          MOVAL   MBA$L_ERB(R4)[R2],R2    ;GET ADDRESS OF DRIVE REGISTERS
         80    82    D0    08FE  1641 20$:     MOVL    (R2)+,(R0)+             ;SAVE DRIVE REGISTER
            FA 51    F5    0901  1642          SOBGTR  R1,20$                  ;ANY MORE TO SAVE?
   80    00D5 C5    9A    0904  1643          MOVZBL  UCB$B_DR_ERL(R5),(R0)+  ;SAVE MEDIUM OFFLINE INDICATOR
               05    0909  1644          RSB                             ;
                     090A  1645
```

```
090A  1647          .SBTTL  DISK DRIVE INITIALIZATION
090A  1648 ;
090A  1649 ; DR_UNIT_INIT - DISK DRIVE UNIT INITIALIZATION
090A  1650 ;
090A  1651 ; THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
090A  1652 ; DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
090A  1653 ;
090A  1654 ; INPUTS:
090A  1655 ;
090A  1656 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
090A  1657 ;       R5 = DEVICE UNIT UCB ADDRESS.
090A  1658 ;
090A  1659 ; OUTPUTS:
090A  1660 ;
090A  1661 ;       UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
090A  1662 ;
090A  1663 ; SPECIAL NOTES:
090A  1664 ;       This routine performs several special operations to support power
090A  1665 ;       failure recovery in the RP07.  To provide an understanding of these
090A  1666 ;       operations, power failure recovery within in the RP07 is discussed
090A  1667 ;       first.  Then, the special actions taken by this routine are discussed.
090A  1668 ;
090A  1669 ;       The power up sequence in a RP07 drive is best described in terms of a
090A  1670 ;       series of numbered states.  The state numbers are shown in the LED
090A  1671 ;       readout on the micro-processor control pannel, the section labeled
090A  1672 ;       "PROGRAM CODE."  The following lists these states and gives the
090A  1673 ;       author's understanding what they mean.
090A  1674 ;
090A  1675 ;       STATE           COMMENTS
090A  1676 ;       00,11,22,...,FF These states occur upon restoration of DC power.
090A  1677 ;                       Presumably they are related to micro-processor
090A  1678 ;                       initialization and testing.  During these states,
090A  1679 ;                       no MASSBUS interaction with the drive is possible.
090A  1680 ;
090A  1681 ;       00,01,02        These states occur after the above states during power
090A  1682 ;                       failure recovery or after the START/STOP switch is
090A  1683 ;                       moved from the STOP to the START position.  These
090A  1684 ;                       states also are related to micro-processor and disk
090A  1685 ;                       system testing.  The disk system is not spinning
090A  1686 ;                       during these states.  During these states, no
090A  1687 ;                       MASSBUS interaction with the drive is possible.
090A  1688 ;
090A  1689 ;       03              During power failure recovery, this is the state in
090A  1690 ;                       which multiple RP07 drives on a single system will
090A  1691 ;                       synchronize their attempts to spin their disk media.
090A  1692 ;                       Limited communication with the drive via the MASSBUS
090A  1693 ;                       is enabled while the drive is in this state.  The
090A  1694 ;                       drive type register can be read, and the clear-drive
090A  1695 ;                       command is accepted.  The drive status register also
090A  1696 ;                       can be read while the drive is in this state.  For
090A  1697 ;                       between 20 and 40 milliseconds after this state is
090A  1698 ;                       entered, however, the drive status register contains
090A  1699 ;                       garbage -- probably all bits except ATA and ERR set,
090A  1700 ;                       a remnant of some internal test.  After this initial
090A  1701 ;                       period, the drive status register contains reasonable,
090A  1702 ;                       valid information.
090A  1703 ;
```

```
                     090A  1704 ;       04,05,06,07,08  These states occur while the disk medium is spining
                     090A  1705 ;                       upto speed.  While in these states, no MASSBUS
                     090A  1706 ;                       interaction with the drive is possible.
                     090A  1707 ;
                     090A  1708 ;       The following aspects of this routine relate specificially to dealing
                     090A  1709 ;       with power failure recovery as practiced by the RP07.
                     090A  1710 ;
                     090A  1711 ;           o   The sieze port operation, performed near the beginning of this
                     090A  1712 ;               routine, also has the effect of waiting for the RP07 drive to
                     090A  1713 ;               reach state 03.  To allow both wait operations -- the sieze
                     090A  1714 ;               port function and the wait for RP07 to reach state 03 function
                     090A  1715 ;               -- to be combined, EXE$PWRTIMCHK is used to time both
                     090A  1716 ;               functions.  When this routine is called for reasons other than
                     090A  1717 ;               power failure recovery, it establishes a 20 millisecond wait
                     090A  1718 ;               interval for EXE$PWRTIMCHK.
                     090A  1719 ;
                     090A  1720 ;           o   Once access to the RP07 has been established, this routine
                     090A  1721 ;               proceeds to determine the drive type, that register can be
                     090A  1722 ;               read and contains valid.
                     090A  1723 ;
                     090A  1724 ;           o   Before proceeding to test for medium-online, however, this
                     090A  1725 ;               routine waits for 50 milliseconds.  This is intended to allow
                     090A  1726 ;               the drive status register to reach a valid state.
                     090A  1727 ;
                     090A  1728 ;           o   The medium-online test will wait for the drive to spin up.
                     090A  1729 ;               Because all drive registers show zero while MASSBUS access to
                     090A  1730 ;               the drive is disabled, it will correctly wait throughout
                     090A  1731 ;               states 03, 04, 05, 06, 07, and 08.
                     090A  1732 ;
                     090A  1733
                     090A  1734 DR_UNIT_INIT:                                ;DISK DRIVE UNIT INITIALIZATION
          53   54 A5  3C  090A  1735          MOVZWL  UCB$W_UNIT(R5),R3       ;GET DRIVE UNIT NUMBER
        0090 C5  53  90  090E  1736          MOVB    R3,UCB$B_SLAVE(R5)      ;SET SLAVE UNIT NUMBER
             53  20  C4  0913  1737          MULL    #<1a7>/4,R3             ;CALCULATE DRIVE OFFSET CONSTANT
        0091 C5  53  90  0916  1738          MOVB    R3,UCB$B_SLAVE+1(R5)    ;SET SLAVE OFFSET CONSTANT
     53   0400 C443  DE  091B  1739          MOVAL   MBA$L_ERB(R4)[R3],R3    ;GET ADDRESS OF DRIVE CONTROL REGISTER
   7E   00000000'GF  D0  0921  1740          MOVL    G^EXE$GL_PWRDONE, -(SP) ;Save current powerfail limit time.
             0E  12  0928  1741          BNEQ    105$                    ;Non-zero value indicates powerfail.
        00000000'GF  16  092A  1742          JSB     G^EXE$READ_TODR         ;If not powerfail, construct our
00000000'GF   50  02  C1  0930  1743          ADDL3   #2,R0,G^EXE$GL_PWRDONE ;limit time for port seizure.
           04 A3  D4  0938  1744 105$:   CLRL    RM_DS(R3)               ;Attempt to seize port.
     52   04 A3  17  78  093B  1745 110$:   ASHL    #3T-RM_DS_V_DPR, RM_DS(R3), R2 ;Did we seize the port?
             0F  19  0940  1746          BLSS    120$                    ;If LSS, we seized the port.
        00000000'GF  16  0942  1747          JSB     G^EXE$PWRTIMCHK         ;Wait for port to be siezed.
             F0 50  E8  0948  1748          BLBS    R0, 110$                ;Branch if haven't waited long enough.
          64 A5  10  AA  094B  1749          BICW    #UCB$M_ONLINE, UCB$W_STS(R5) ;If never get the port,
             5D  11  094F  1750          BRB     15$                     ;mark the drive offline and invalid.
     52   18 A3  14  78  0951  1751 120$:   ASHL    #31-RM_DT_V_DRQ, RM_DT(R3), R2 ;Is there a dualport kit?
             05  18  0956  1752          BGEQ    5$                      ;If GEQ, no dualport kit; continue.
        00D4 C5  08  88  0958  1753          BISB    #DR_M_DUALPORT, -       ;Else, set flag indicating that disk
                     095D  1754                  UCB$B_DR_SSTS(R5)       ;has a dualport kit.
          64 A5  10  A8  095D  1755 5$:     BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
           00A5  30  0961  1756          BSBW    DR_DTYPE                ;CLASSIFY DRIVE TYPE
     45 64 A5  04  E1  0964  1757          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),15$ ;IF CLR, UNKNOWN DRIVE TYPE
     48 64 A5  0B  E1  0969  1758          BBC     #UCB$V_VALID,UCB$W_STS(R5),30$  ;IF CLR, VOLUME SOFTWARE INVALID
        41 A5  07  91  096E  1759          CMPB    #DT$_RP07, UCB$B_DEVTYPE(R5) ;Is this a RP07?
             27  12  0972  1760          BNEQ    10$                     ; Branch if not a RP07.
```

```
                                0974  1761          .SHOW   MEB
                                0974  1762          TIMEWAIT -                      ; If this is a RP07,
                                0974  1763                   time = #5000, -        ; wait for it to finish its
                                0974  1764                   bitval = #0, -         ; cup of coffee.
                                0974  1765                   source = #0, -
                                0974  1766                   context = B
                     50   01   3C   0974                MOVZWL  #SS$_NORMAL,R0
51   00000000'GF   00001388 8F   C5   0977                MULL3   #5000,G^EXE$GL_TENUSEC,R1
                           7E   D4   0983                CLRL    -(SP)
                        00   00   93   0985      30010$: BITB    #0,#0
                           0F   12   0988                BNEQ    30011$
                  6E   00000000'GF   D0   098A                MOVL    G^EXE$GL_UBDELAY,(SP)
                        FD   6E   F5   0991      30012$: SOBGTR  (SP),30012$
                        EE   51   F5   0994              SOBGTR  R1,30010$
                           50   D4   0997                CLRL    R0
                                0999      30011$:
                           8E   D5   0999                TSTL    (SP)+
                                099B  1767          .NOSHOW MEB
                     63    09   9A   099B  1768 10$:     MOVZBL  #F_DRVCLR!1,RM_CS1(R3)  ;CLEAR DRIVE
                52   04 A3    13   78   099E  1769          ASHL    #3T-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
                           11   19   09A3  1770          BLSS    20$                     ;IF LSS YES
                00000000'GF   16   09A5  1771          JSB     G^EXE$PWRTIMCHK         ;CHECK FOR MAXIMUM TIME EXCEEDED
                        ED   50   E8   09AB  1772          BLBS    R0,10$                  ;IF LBS MORE TIME TO GO
             64 A5    0800 8F   AA   09AE  1773 15$:     BICW    #UCB$M_VALID,UCB$W_STS(R5) ;MARK THE VOLUME INVALID
                           03   11   09B4  1774          BRB     30$
                     63    13   9A   09B6  1775 20$:     MOVZBL  #F_PACKACK!1,RM_CS1(R3) ;ACKNOWLEDGE PACK
                     63    0B   9A   09B9  1776 30$:     MOVZBL  #F_RELEASE!1,RM_CS1(R3) ;CLEAR DRIVE
                08 A4    08 A4   C8   09BC  1777 40$:     BISL    MBA$L_SR(R4), MBA$L_SR(R4) ;CLEAR MBA STATUS
                           8E   D5   09C1  1778          TSTL    (SP)+                   ;If powerfail limit time was zero
                           06   12   09C3  1779          BNEQ    50$                     ;when we started, make sure its
                00000000'GF   D4   09C5  1780          CLRL    G^EXE$GL_PWRDONE        ;zero when we leave.
                           05   09CB  1781 50$:     RSB
```

```
                     09CC  1783          .SBTTL  UNSOLICITED INTERRUPT ROUTINE
                     09CC  1784 ;
                     09CC  1785 ; DR_UNSOLNT - UNSOLICITED INTERRUPT ROUTINE
                     09CC  1786 ;
                     09CC  1787 ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED.
                     09CC  1788 ;
                     09CC  1789 ; INPUTS:
                     09CC  1790 ;
                     09CC  1791 ;       R4 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                     09CC  1792 ;       R5 = DEVICE UNIT UCB ADDRESS.
                     09CC  1793 ;
                     09CC  1794 ; OUTPUTS:
                     09CC  1795 ;
                     09CC  1796 ;       IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
                     09CC  1797 ;       UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
                     09CC  1798 ;       CLASSIFIED.
                     09CC  1799 ;
                     09CC  1800
                     09CC  1801 DR_UNSOLNT:                                  ;UNSOLICITED INTERRUPT
      53   0091 C5   9A  09CC  1802          MOVZBL  UCB$B_SLAVE+1(R5),R3     ;GET DRIVE OFFSET CONSTANT
   53   0400 C443   DE  09D1  1803          MOVAL   MBA$L_ERB(R4)[R3],R3     ;GET ADDRESS OF DRIVE CONTROL REGISTER
      64 A5   10   A8  09D7  1804          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
            002B   30  09DB  1805          BSBW    DR_DTYPE                 ;CLASSIFY DRIVE TYPE
    1F 64 A5   04   E1  09DE  1806          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),10$ ;IF CLR, UNKNOWN DRIVE TYPE
    20 64 A5   0B   E1  09E3  1807          BBC     #UCB$V_VALID,UCB$W_STS(R5),20$ ;IF CLR, VOLUME SOFTWARE INVALID
52     04 A3   13   78  09E8  1808          ASHL    #31-RM_DS_V_MOL,RM_DS(R3),R2 ;MEDIUM ONLINE?
            13   18  09ED  1809          BGEQ    10$                      ;IF GEQ NO
    07 64 A5   08   E1  09EF  1810          BBC     #UCB$V_BSY,UCB$W_STS(R5),5$ ;We know the drive is online; thus,
     0093 C5   08   91  09F4  1811          CMPB    #CDF_PACKACK,UCB$B_CEX(R5)  ;if busy doing a PACKACK function,
            0D   13  09F9  1812          BEQL    20$                      ;then don't clear software valid.
52     04 A3   19   78  09FB  1813 5$:      ASHL    #31-RM_DS_V_VV,RM_DS(R3),R2 ;VOLUME VALID?
            06   19  0A00  1814          BLSS    20$                      ;IF LSS YES
64 A5    0800 8F   AA  0A02  1815 10$:     BICW    #UCB$M_VALID,UCB$W_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
                   05  0A08  1816 20$:     RSB                              ;
```

```
                     0A09  1818          .SBTTL  CLASSIFY DRIVE TYPE AND SET PARAMETERS
                     0A09  1819 ;
                     0A09  1820 ; RM_DTYPE - CLASSIFY DRIVE TYPE AND SET PARAMETERS
                     0A09  1821 ;
                     0A09  1822 ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
                     0A09  1823 ; SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
                     0A09  1824 ; SET UNIT PARAMETERS.
                     0A09  1825 ;
                     0A09  1826 ; INPUTS:
                     0A09  1827 ;
                     0A09  1828 ;       R3 = ADDRESS OF DRIVE CONTROL REGISTER.
                     0A09  1829 ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                     0A09  1830 ;       R5 = DEVICE UNIT UCB ADDRESS.
                     0A09  1831 ;
                     0A09  1832 ; OUTPUTS:
                     0A09  1833 ;
                     0A09  1834 ;       THE DRIVE TYPE REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
                     0A09  1835 ;
                     0A09  1836
                     0A09  1837 DR_DTYPE:                                ;CLASSIFY DRIVE TYPE AND SET PARAMETERS
          18 A3   DD 0A09  1838          PUSHL   RM_DT(R3)               ;READ DRIVE TYPE REGISTER
6E   FE00 8F   AA 0A0C  1839          BICW    #^C<^X1FF>,(SP)         ;CLEAR EXTRANEOUS BITS
52   F627 CF   9E 0A11  1840          MOVAB   DR_DTDESC,R2            ;GET ADDRESS OF DESCRIPTOR TABLE
     82   6E   B1 0A16  1841 10$:     CMPW    (SP),(R2)+              ;DRIVE TYPE MATCH?
          0E   13 0A19  1842          BEQL    20$                     ;IF EQL YES
     52   0D   C0 0A1B  1843          ADDL    #DR_DTDESCLEN-2,R2      ;ADVANCE TO NEXT ENTRY
          62   B5 0A1E  1844          TSTW    (R2)                    ;END OF TABLE?
          F4   12 0A20  1845          BNEQ    10$                     ;IF NEQ NO
  64 A5   10   AA 0A22  1846          BICW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT OFFLINE
     52   0D   C2 0A26  1847          SUBL    #DR_DTDESCLEN-2,R2      ;BACK UP TO LAST DRIVE DESCRIPTOR
  41 A5   82   90 0A29  1848 20$:     MOVB    (R2)+,UCB$B_DEVTYPE(R5) ;SET DEVICE TYPE
  44 A5   82   D0 0A2D  1849          MOVL    (R2)+,UCB$L_DEVDEPEND(R5) ;SET DISK PACK GEOMETRY
0080 C5   82   D0 0A31  1850          MOVL    (R2)+,UCB$L_MAXBLOCK(R5);SET MAXIMUM BLOCKS PER PACK
0080 C5   62   D0 0A36  1851          MOVL    (R2),UCB$L_MEDIA_ID(R5) ;SET MEDIA IDENT
          8E   D5 0A3B  1852          TSTL    (SP)+                   ;REMOVE DRIVE TYPE FROM STACK
               05 0A3D  1853          RSB                             ;
                     0A3E  1854 DR_END:                                ;ADDRESS OF LAST LOCATION IN DRIVER
                     0A3E  1855
                     0A3E  1856          .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                 0000038E R       03
ATS_MBA                 = 00000000
AVAILABLE                 00000245 R       03
CDF_AVAILABLE           = 00000011
CDF_DIAGNOSE            = 00000014
CDF_DRVCLR              = 00000004
CDF_NOP                 = 00000005
CDF_OFFSET              = 00000006
CDF_PACKACK             = 00000008
CDF_READDATA            = 0000000C
CDF_READHEAD            = 0000000E
CDF_READPRESET          = 00000013
CDF_READTRACKD          = 00000010
CDF_RECAL               = 00000003
CDF_RETCENTER           = 00000007
CDF_SEARCH              = 00000009
CDF_SEARCHA             = 00000015
CDF_SEEK                = 00000002
CDF_WRITECHECK          = 0000000A
CDF_WRITECHECKH         = 00000012
CDF_WRITEDATA           = 0000000B
CDF_WRITEHEAD           = 0000000D
CDF_WRITETRACKD         = 0000000F
CFUNC                     00000785 R       03
CHECKRETRY                000002E7 R       03
CHECKTAB                  00000038 R       03
CHECKXT                   00000300 R       03
CRB$L_INTD              = 00000024
DATACHECK                 0000029A R       03
DC$_DISK                = 00000001
DDB$K_PACK              = 00000001
DDB$L_ACPD              = 00000010
DDB$L_DDT               = 0000000C
DEFER_ECC                 00000384 R       03
DEV$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$M_SHR               = 00010000
DFUNC                     000007AC R       03
DIAGNOSE                  000026A R        03
DPT$C_LENGTH            = 00000038
DPT$C_VERSION           = 00000004
DPT$INITAB                00000038 R       02
DPT$M_SVP                       = 00000002
DPT$REINITAB                      0000006A R       02
DPT$TAB                           00000000 R       02
DR$DDT                            00000000 RG      03
DRVCLR                            00000254 R       03
DR_DTDESC                         0000003C R       03
DR_DTDESCLEN                    = 0000000F
DR_DTYPE                          00000A09 R       03
DR_END                            00000A3E R       03
DR_FUNCTABLE                      000000B2 R       03
DR_M_DCK                        = 00000001
DR_M_DUALPORT                   = 00000008
DR_M_ECC_DEFER                  = 00000010
DR_M_NOECC                      = 00000004
DR_M_OM                         = 00000002
DR_REGDUMP                        000008C0 R       03
DR_STARTIO                        00000146 R       03
DR_UNIT_INIT                      0000090A R       03
DR_UNSOCNT                        000009CC R       03
DR_V_DCK                        = 00000000
DR_V_DUALPORT                   = 00000003
DR_V_ECC_DEFER                  = 00000004
DR_V_NOECC                      = 00000002
DR_V_OM                         = 00000001
DT$_RM03                        = 00000006
DT$_RM05                        = 0000000F
DT$_RM80                        = 0000000D
DT$_RP07                        = 00000007
DYN$C_DDB                       = 00000006
DYN$C_DPT                       = 0000001E
DYN$C_UCB                       = 00000010
ECC                               00000331 R       03
EMB$L_DV_REGSAV                 = 0000004E
ENBXIT                            000006C8 R       03
ERL$DEVICERR                      ********   X     03
ERL$DEVICTMO                      ********   X     03
ERROR                             000006FD R       03
EXE$GL_PWRDONE                    ********   X     03
EXE$GL_TENUSEC                    ********   X     03
EXE$GL_UBDELAY                    ********   X     03
EXE$IOFORK                        ********   X     03
EXE$LCLDSKVALID                   ********   X     03
EXE$ONEPARM                       ********   X     03
EXE$PWRTIMCHK                     ********   X     03
EXE$READ_TODR                     ********   X     03
EXE$SENSEMODE                     ********   X     03
EXE$SETCHAR                       ********   X     03
EXE$ZEROPARM                      ********   X     03
EXFNC                             00000628 R       03
FATALERR                          00000473 R       03
FDISPATCH                         000001BA R       03
FEX                               00000556 R       03
FTAB                              00000098 R       03
FUNCTAB_LEN                     = 00000094
FUNCXT                            0000050B R       03
F_AVAILABLE                     = 00000000
F_DIAGNOSE                      = 0000001C
```

```
F_DRVCLR                = 00000008
F_NOP                   = 00000000
F_OFFSET                = 0000000C
F_PACKACK               = 00000012
F_READDATA              = 00000038
F_READHEAD              = 0000003A
F_READPRESET            = 00000010
F_READTRACKD            = 0000003C
F_RECAL                 = 00000006
F_RELEASE               = 0000000A
F_RETCENTER             = 0000000E
F_SEARCH                = 00000018
F_SEARCHA               = 00000018
F_SEEK                  = 00000004
F_WRITECHECK            = 00000028
F_WRITECHECKH           = 0000002A
F_WRITEDATA             = 00000030
F_WRITEHEAD             = 00000032
F_WRITETRACKD           = 00000034
GO                        00000575 R      03
IDB$L_OWNER             = 00000004
IMMED                     000005A6 R      03
IO$V_COMMOD             = 00000006
IO$V_DATACHECK          = 0000000E
IO$V_DIAGNOSTIC         = 00000008
IO$V_INHRETRY           = 0000000F
IO$V_INHSEEK            = 0000000C
IO$V_MOVETRACKD         = 00000007
IO$V_SKPSECINH          = 00000009
IO$_ACCESS              = 00000032
IO$_ACPCONTROL          = 00000038
IO$_AVAILABLE           = 00000011
IO$_CREATE              = 00000033
IO$_DEACCESS            = 00000034
IO$_DELETE              = 00000035
IO$_DIAGNOSE            = 0000001D
IO$_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$_READPRESET          = 00000019
IO$_READTRACKD          = 00000010
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$_WRITEPBLK           = 0000000B
IO$_WRITETRACKD         = 0000000F
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                     00000622 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            = 000008C0
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                           00000254 R      03
NORMAL                        000002FD R      03
OFF                           000003B9 R      03
OFFSET                        00000254 R      03
OFFSIZ                      = 00000004
OFFTAB                        000000AE R      03
PACKACK                       0000024E R      03
POSIT                         0000061C R      03
PR$_IPL                     = 00000012
READDATA                      0000027C R      03
READHEAD                      0000027C R      03
READPRESET                    00000254 R      03
READTRACKD                    00000261 R      03
RECAL                         00000254 R      03
RELEASE                       00000254 R      03
RESETXFR                      000008A1 R      03
RETCENTER                     00000254 R      03
RETREG                        000006CB R      03
RETRY                         0000032E R      03
RETRYERR                      00000441 R      03
RM_AS                         00000010
RM_CS1                        00000000
RM_CS1_M_GO                 = 00000001
RM_DA                         00000014
RM_DC                         00000028
RM_DS                         00000004
RM_DS_M_DPR                 = 00000100
RM_DS_M_ERR                 = 00004000
RM_DS_V_DPR                 = 00000008
RM_DS_V_ERR                 = 0000000E
RM_DS_V_MOL                 = 0000000C
RM_DS_V_VV                  = 00000006
RM_DT                         00000018
RM_DT_V_DRQ                 = 0000000B
RM_EC1                        00000038
RM_EC2                        0000003C
RM_ER1                        00000008
RM_ER1_M_AOE                = 00000200
RM_ER1_M_DCK                = 00008000
RM_ER1_M_DTE                = 00001000
RM_ER1_M_ECH                = 00000040
RM_ER1_M_FER                = 00000010
RM_ER1_M_HCE                = 00000080
RM_ER1_M_HCRC               = 00000100
RM_ER1_M_IAE                = 00000400
RM_ER1_M_ILF                = 00000001
RM_ER1_M_ILR                = 00000002
RM_ER1_M_OPI                = 00002000
RM_ER1_M_PAR                = 00000008
RM_ER1_M_RMR                = 00000004
RM_ER1_M_UNS                = 00004000
RM_ER1_M_WCF                = 00000020
RM_ER1_M_WLE                = 00000800
RM_ER1_V_FER                = 00000004
RM_ER1_V_HCE                = 00000007
RM_ER1_V_HCRC               = 00000008
RM_ER1_V_OPI                = 0000000D
RM_ER1_V_UNS                = 0000000E
RM_ER1_V_WLE                = 0000000B
RM_ER2                        00000034
RM_ER2_M_BSE                = 00008000
RM_ER2_M_DPE                = 00000008
RM_ER2_M_DVC                = 00000080
RM_ER2_M_IVC                = 00001000
RM_ER2_M_LBC                = 00000400
RM_ER2_M_LSC                = 00000800
RM_ER2_M_OPE                = 00002000
RM_ER2_V_BSE                = 0000000F
RM_ER2_V_SKI                = 0000000E
RM_ER2_V_SSE                = 00000005
RM_LA                         0000001C
RM_MR                         0000000C
RM_MR2                        00000030
RM_MR_M_DM                  = 00008000
RM_OF                         00000024
RM_OF_M_CMD                 = 00008000
RM_OF_M_FMT                 = 00001000
RM_OF_M_HCI                 = 00000400
RM_OF_M_MTD                 = 00004000
RM_OF_M_SSEI                = 00000200
RM_OF_V_SSEI                = 00000009
RM_SN                         00000020
RM_UNUSED                     0000002C
SAFUNC                        00000777 R      03
SEARCH                        00000254 R      03
SEARCHA                       000005F9 R      03
SEEK                          00000254 R      03
SEIZE                         000005BD R      03
SIZ...                      = 00000001
SPECOND                       00000848 R      03
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                      00000290 R      03
TRANRQCH                      0000028A R      03
TRANXT                        00000303 R      03
UCB$B_CEX                   = 00000093
UCB$B_DEVCLASS              = 00000040
UCB$B_DEVTYPE               = 00000041
UCB$B_DIPL                  = 0000005E
```

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

Psect synopsis

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

Performance indicators

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

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

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	30
_255DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	40

2486 GETS were required to define 40 macros.

There were no errors, warnings or information messages.

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

DQDRIVER
LIS

DLDRIVER
LIS

DRDRIVER
LIS

DMDRIVER
LIS

DUHIRT
LIS

DUDRIVER
LIS