



```

DDDDDDDD  BBBB8888  DDDDDDDD  RRRRRRRR  IIIIII  VV      VV  EEEEEEEEE  RRRRRRRR
DDDDDDDD  BBBB8888  DDDDDDDD  RRRRRRRR  IIIIII  VV      VV  EEEEEEEEE  RRRRRRRR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BBBB8888  DDDDDDDD  RRRRRRRR  II      II  VV      VV  EEEEEEEEE  RRRRRRRR
DD      DD  BBBB8888  DDDDDDDD  RRRRRRRR  II      II  VV      VV  EEEEEEEEE  RRRRRRRR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  BB      BB  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DDDDDDDD  BBBB8888  DDDDDDDD  RR      RR  IIIIII  VV      VV  EEEEEEEEE  RR      RR
DDDDDDDD  BBBB8888  DDDDDDDD  RR      RR  IIIIII  VV      VV  EEEEEEEEE  RR      RR

```

```

LL      LL      IIIIII  SSSSSSSS
LL      LL      IIIIII  SSSSSSSS
LL      LL      II      SS
LL      LL      II      SS
LL      LL      II      SS
LL      LL      II      SS
LL      LL      II      SSSSSS
LL      LL      II      SSSSSS
LL      LL      II      SS
LL      LL      II      SS
LL      LL      II      SS
LL      LL      II      SS
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS

```

(1)	399	RP04/05/06 FUNCTION DECISION TABLE
(1)	507	START I/O OPERATION
(1)	1001	RP04/05/06 HARDWARE FUNCTION EXECUTION
(1)	1424	RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
(1)	1461	RP04/05/06 REGISTER DUMP ROUTINE
(1)	1500	RP04/RP05/RP06 DISK DRIVE INITIALIZATION
(1)	1553	RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE

```

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

```

```

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

```

```

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

```

```

000C 172 <DTE,,M>,- : DRIVE TIMING ERROR
000C 173 <OPI,,M>,- : OPERATION INCOMPLETE
000C 174 <UNS,,M>,- : DRIVE UNSAFE
000C 175 <DCK,,M>- : DATA CHECK ERROR
000C 176 > :
000C 177 $DEF RP_MR .BLKL 1 : MAINTENANCE REGISTER
0010 178 $DEF RP_AS .BLKL 1 : ATTENTION SUMMARY REGISTER
0014 179 $DEF RP_DA .BLKL 1 : DESIRED SECTOR/TRACK ADDRESS REGISTER
0018 180 -VIELD RP_DA,0,<- : DESIRED ADDRESS FIELD DEFINITIONS
0018 181 <SA,5>,- : DESIRED SECTOR ADDRESS
0018 182 <3>- : RESERVED BITS
0018 183 <TA,5>- : DESIRED TRACK ADDRESS
0018 184 > :
0018 185 $DEF RP_DT .BLKL 1 : DRIVE TYPE REGISTER
001C 186 -VIELD RP_DT,0,<- : DRIVE TYPE REGISTER FIELD DEFINITIONS
001C 187 <DTN,9>,- : DRIVE TYPE NUMBER
001C 188 <2>- : RESERVED BITS
001C 189 <DRQ,,M>- : DRIVE REQUEST REQUIRED
001C 190 > :
001C 191 $DEF RP_LA .BLKL 1 : LOOKAHEAD REGISTER
0020 192 $DEF RP_ER2 .BLKL 1 : ERROR REGISTER 2
0024 193 $DEF RP_OF .BLKL 1 : OFFSET REGISTER
0028 194 -VIELD RP_OF,0,<- : OFFSET REGISTER BIT DEFINITIONS
0028 195 <OFF,8>,- : OFFSET VALUE
0028 196 <DCK,,M>,- : DATA CHECK IN PROGRESS (SOFTWARE)
0028 197 <1>- : RESERVED BIT
0028 198 <HCI,,M>,- : HEADER COMPARE INHIBIT
0028 199 <ECI,,M>,- : ECC INHIBIT
0028 200 <FMT,,M>- : 16-BIT FORMAT
0028 201 > :
0028 202 $DEF RP_DC .BLKL 1 : DESIRED CYLINDER ADDRESS
002C 203 $DEF RP_CC .BLKL 1 : CURRENT CYLINDER ADDRESS
0030 204 $DEF RP_SN .BLKL 1 : DRIVE SERIAL NUMBER
0034 205 $DEF RP_ER3 .BLKL 1 : ERROR REGISTER 3
0038 206 -VIELD RP_ER3,14,<- : ERROR REGISTER 3 BIT DEFINITIONS
0038 207 <SRI,,M>- : SEEK INCOMPLETE
0038 208 > :
0038 209 $DEF RP_EC1 .BLKL 1 : ECC POSITION REGISTER
003C 210 VIELD RP_EC1,0,<<POS,13>> : ECC POSITION FIELD
003C 211 $DEF RP_EC2 .BLKL 1 : ECC PATTERN REGISTER
0040 212 -VIELD RP_EC2,0,<<PAT,11>> : ECC PATTERN FIELD
0040 213 :
0040 214 $DEFEND RP
0000 215 :
0000 216 :
0000 217 : DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
0000 218 :
0000 219 :
0000 220 $DEFINI UCB
0000 221 :
000000CC 0000 222 .=UCB$%_LCL_DISK_LENGTH : Establish device-dependent UCB base
00CC 223 :
000000C0 00CC 224 UCBSL_DB_BCR = UCBSL_BCR : Local BCR longword overlays the
00CC 225 : space reserved in the UCB.
00CC 226 : N.B. most drivers only need a word.
00CC 227 :
00CC 228 $DEF UCBSW_DB_ER3 .BLKW 1 : Space to save RP_ER3 after operation.

```

```

00CE 229
00CE 230 $DEF UCBSL_DB_SR .BLKL 1 ;SAVE MBA STATUS REGISTER
00D3 231
00D3 232 $DEF UCBSB_DB_ERL .BLKB 1 ; Space for flag used to signal Medium
00D3 233 ; offline at start of function.
00D3 234 _VIELD ERL,0,<-
0CD3 235 <MEDOFF,,M>,- ; MEDIUM OFFLINE FLAG
00D3 236 <DUALPORT,,M>,- ; DUALPORT KIT FLAG
00D3 237 <ECC_DEFER,,M>,- ; Flag to indicate that ECC correction
00D3 238 > ; has been deferred until offset
00D3 239 ; retries are exhausted.
00D3 240
000000D6 00D3 241 .BLKB 3 ; Reserved.
000000D6 00D6 242 UCBSK_DB_LENGTH=. ; Length of UCB for DB devices.
00D6 243 $DEFEND
0000 244
0000 245 ;
0000 246 ; HARDWARE FUNCTION CODES
0000 247 ;
0000 248
00000000 0000 249 F_NOP=0*2 ;NO OPERATION
00000002 0000 250 F_UNLOAD=1*2 ;UNLOAD DRIVE
00000004 0000 251 F_SEEK=2*2 ;SEEK CYLINDER
00000006 0000 252 F_RECAL=3*2 ;RECALIBRATE
00000008 0000 253 F_DRVCLR=4*2 ;DRIVE CLEAR
0000000A 0000 254 F_RELEASE=5*2 ;RELEASE DRIVE
0000000C 0000 255 F_OFFSET=6*2 ;OFFSET HEADS
0000000E 0000 256 F_RETCENTER=7*2 ;RETURN TO CENTERLINE
00000010 0000 257 F_READPRESET=8*2 ;READ IN PRESET
00000012 0000 258 F_PACKACK=9*2 ;PACK ACKNOWLEDGE
00000018 0000 259 F_SEARCH=12*2 ;SEARCH FOR SECTOR
00000018 0000 260 F_SEARCHA=12*2 ;SEARCH AHEAD FOR SECTOR
00000028 0000 261 F_WRITECHECK=20*2 ;WRITE CHECK DATA
0000002A 0000 262 F_WRITECHECKH=21*2 ;WRITE CHECK HEADER AND DATA
00000030 0000 263 F_WRITEDATA=24*2 ;WRITE DATA
00000032 0000 264 F_WRITEHEAD=25*2 ;WRITE HEADER AND DATA
00000038 0000 265 F_READDATA=28*2 ;READ DATA
0000003A 0000 266 F_READHEAD=29*2 ;READ HEADER AND DATA
0000 267
0000 268 ;
0000 269 ; LOCAL DATA
0000 270 ;
0000 271 ; DRIVER PROLOGUE TABLE
0000 272 ;
0000 273
0000 274 DPTAB - ;DEFINE DRIVER PROLOGUE TABLE
0000 275 END=DB_END,- ;END OF DRIVER
0000 276 ADAPTER=MBA,- ;ADAPTER TYPE
0000 277 FLAGS=DPTSM_SVP,- ;SYSTEM PAGE TABLE ENTRY REQUIRED
0000 278 UCBSIZE=UCBSK_DB_LENGTH,- ;UCB SIZE
0000 279 NAME=DBDRIVER- ;DRIVER NAME
0038 280 DPT_STORE INIT ;CONTROL BLOCK INIT VALUES
0038 281 DPT_STORE DDB,DBS$ACPD,L,<^A\F11> ;DEFAULT ACP NAME
003F 282 DPT_STORE DDB,DBS$ACPD+3,B,DBSK_PACK ;ACP CLASS
0043 283 DPT_STORE UCB,UCBSB_FIPL,B,8 ;FORK IPL
0047 284 DPT_STORE UCB,UCBSL_DEVCHAR,L,- ;DEVICE CHARACTERISTICS
0047 285 <DEVSM_FOD- ; FILES ORIENTED

```



```

0047 286      :DEVSM_DIR-          : DIRECTORY STRUCTURED
0047 287      :DEVSM_AVL-          : AVAILABLE
0047 288      :DEVSM_ELG-          : ERROR LOGGING ENABLED
0047 289      :DEVSM_SHR-          : SHAREABLE
0047 290      :DEVSM_IDV-          : INPUT DEVICE
0047 291      :DEVSM_ODV-          : OUTPUT DEVICE
0047 292      :DEVSM_RND>      : RANDOM ACCESS
004E 293      DPT_STORE UCB,OCBSL_DEVCHAR2,L,- : DEVICE CHARACTERISTICS
004E 294      <DEVSM_NNM>    : PREFIX NAME WITH "node$"
0055 295      DPT_STORE UCB,OCBSB_DEVCLASS,B,DC$ :DISK ;DEVICE CLASS
0059 296      DPT_STORE UCB,UCBSW_DEVBUFSIZ,W,512 :DEFAULT BUFFER SIZE
005E 297      DPT_STORE UCB,UCBSB_DIPL,B,21      :DEVICE IPL
0062 298      DPT_STORE UCB,UCBSB_ERTCNT,B,8     :ERROR RETRY COUNT
0066 299      DPT_STORE UCB,UCBSB_ERTMAX,B,8     :MAX ERROR RETRY COUNT
006A 300      DPT_STORE REINIT                    :CONTROL BLOCK RE-INIT VALUES
006A 301      DPT_STORE DDB,DDBSL_DDT,D,DB$DDT   :DDT ADDRESS
006F 302      DPT_STORE END                        :
0000 303
0000 304      :
0000 305      : DRIVER DISPATCH TABLE
0000 306      :
0000 307
0000 308      DDTAB DB,-          :DRIVER DISPATCH TABLE
0000 309      DB_STARTIO,-       :START I/O OPERATION
0000 310      DB_UNSolNT,-      :UNSOLICITED INTERRUPT
0000 311      DB_FUNCTABLE,-    :FUNCTION DECISION TABLE
0000 312      C,-              :CANCEL I/O ENTRY POINT
0000 313      DB_REGDUMP,-      :REGISTER DUMP ROUTINE
0000 314      <<RP_EC2+4+MBASL_BCR+4+8>+<<3+5+1>*4>>,- :DIAGNOSTIC BUFFER SIZE
0000 315      <<RP_EC2+4+MBASL_BCR+4+8>+<1*4>+<EMBSL_DV_REGSAV>>,- :ERROR BUFFER S
0000 316      DB_RPOX_INIT      :UNIT INITIALIZATION
0038 317
0038 318      :
0038 319      : DATA CHECK FUNCTION TRANSLATION TABLE
0038 320      :
0038 321
0038 322 CHECKTAB:
0038 323      .BYTE CDF_WRITECHECK      :WRITE DATA
0038 324      .BYTE CDF_WRITECHECK      :READ DATA
003A 325      .BYTE CDF_WRITECHECKKH   :WRITE HEADER AND DATA
003B 326      .BYTE CDF_WRITECHECKKH   :READ HEADER AND DATA
003C 327
003C 328      :
003C 329      : RPOX DRIVE TYPE DESCRIPTOR TABLE
003C 330      :
003C 331
003C 332 DB_DTDESC:
0010 003C 333      .WORD ^X10              :RP04
003E 334      .BYTE DTS_RP04
003F 335      .BYTE 22                :22 SECTORS
0040 336      .BYTE 19                :19 TRACKS
019B 0041 337      .WORD 411            :411 CYLINDERS PER PACK
00029F16 0043 338      .LONG 411*19*22        :MAXIMUM BLOCKS PER PACK
20A50004 0047 339      .LONG ^X20A50004      :MEDIA ID "DB RP04"
0000000F 004B 340 DB_DTDESCLEN=-DB_DTDESC      :LENGTH OF DRIVE TYPE DESCRIPTOR
0011 004B 341      .WORD ^X11
004D 004D 342      .BYTE DTS_RP05

```

```

16 004E 343 .BYTE 22 :22 SECTORS
13 004F 344 .BYTE 19 :19 TRACKS
0198 0050 345 .WORD 411 :411 CYLINDERS PER PACK
00029F16 0052 346 .LONG 411*19*22 :MAXIMUM BLOCKS PER PACK
20A50004 0056 347 .LONG ^X20A50004 :MEDIA ID 'DB RP04'
0012 005A 348 .WORD ^X12 :RP06
05 005C 349 .BYTE DTB_RP06
16 005D 350 .BYTE 22 :22 SECTORS
13 005E 351 .BYTE 19 :19 TRACKS
032F 005F 352 .WORD 815 :815 CYLINDERS PER PACK
000532BE 0061 353 .LONG 815*19*22 :MAXIMUM BLOCKS PER PACK
20A50006 0065 354 .LONG ^X20A50006 :MEDIA ID 'DB RP06'
0069 355
0000 0069 356 .WORD 0 :END OF TABLE
0000007A 006B 357 .BLKB DB_DTDESCLEN :SPARE DRIVE TYPE SLOT
00000089 007A 358 .BLKB DB_DTDESCLEN :SPARE DRIVE TYPE SLOT
0089 359
0089 360 :
0089 361 : HARDWARE I/O FUNCTION CODE TABLE
0089 362 :
0C89 363
0089 364 FTAB:
0089 365 GENF F_NOP :NO OPERATION
008A 366 GENF F_UNLOAD :UNLOAD VOLUME
008B 367 GENF F_SEEK :SEEK CYLINDER
008C 368 GENF F_RECAL :RECALIBRATE
008D 369 GENF F_DRVCLR :DRIVE CLEAR
008E 370 GENF F_NOP :RELEASE PORT (NOP)
008F 371 GENF F_OFFSET :OFFSET HEADS
0090 372 GENF F_RETCENTER :RETURN HEADS TO CENTERLINE
0091 373 GENF F_PACKACK :PACK ACKNOWLEDGE
0092 374 GENF F_SEARCH :SEARCH FOR SECTOR
0093 375 GENF F_WRITECHECK :WRITE CHECK
0094 376 GENF F_WRITEDATA :WRITE DATA
0095 377 GENF F_READDATA :READ DATA
0096 378 GENF F_WRITEHEAD :WRITE HEADER AND DATA
0097 379 GENF F_READHEAD :READ HEADER AND DATA
0098 380 GENF F_WRITECHECKH :WRITE CHECK HEADER AND DATA
0099 381 GENF F_READPRESET :READ IN PRESET
009A 382 GENF F_SEARCHA :SEARCH AHEAD FOR SECTOR
009B 383
009B 384 :
009B 385 : OFFSET TABLE FOR RP06 - RP04 VALUES = RP06 VALUES * 2 & ^XFF
009B 386 :
009B 387
009B 388 OFFTAB:
00 009B 389 .BYTE 0 :RETURN TO CENTERLINE
08 009C 390 .BYTE ^X8 :+200 (+400)
C8 009D 391 .BYTE ^XC8 : -200 (-400)
10 009E 392 .BYTE ^X10 :+400 (+800)
D0 009F 393 .BYTE ^XD0 : -400 (-800)
18 00A0 394 .BYTE ^X18 :+600 (+1200)
D8 00A1 395 .BYTE ^XD8 : -600 (-1200)
00 00A2 396 .BYTE 0 :RETURN TO CENTERLINE
00000008 00A3 397 OFFSIZ=-OFFTAB :SIZE OF OFFSET TABLE

```

```
00A3 399 .SBTTL RP04/05/06 FUNCTION DECISION TABLE
00A3 400 :+
00A3 401 : RP04/05/06 FUNCTION DECISION TABLE
00A3 402 :-
00A3 403 :-
00A3 404 DB_FUNCTABLE:
00A3 405 FUNCTAB :-
00A3 406 <NOP,-
00A3 407 UNLOAD,-
00A3 408 SEEK,-
00A3 409 RECAL,-
00A3 410 DRVCLR,-
00A3 411 RELEASE,-
00A3 412 OFFSET,-
00A3 413 RETCENTER,-
00A3 414 PACKACK,-
00A3 415 SEARCH,-
00A3 416 READPRESET,-
00A3 417 SENSECHAR,-
00A3 418 SETCHAR,-
00A3 419 SENSEMODE,-
00A3 420 SETMODE,-
00A3 421 WRITECHECK,-
00A3 422 WRITEHEAD,-
00A3 423 READHEAD,-
00A3 424 WRITECHECKH,-
00A3 425 READBLK,-
00A3 426 WRITELBLK,-
00A3 427 READPBLK,-
00A3 428 WRITEPBLK,-
00A3 429 READVBLK,-
00A3 430 WRITEVBLK,-
00A3 431 AVAILABLE,-
00A3 432 ACCESS,-
00A3 433 ACPCONTROL,-
00A3 434 CREATE,-
00A3 435 DEACCESS,-
00A3 436 DELETE,-
00A3 437 MODIFY,-
00A3 438 MOUNT>
00AB 439 FUNCTAB :-
00AB 440 <NOP,-
00AB 441 UNLOAD,-
00AB 442 SEEK,-
00AB 443 RECAL,-
00AB 444 DRVCLR,-
00AB 445 RELEASE,-
00AB 446 OFFSET,-
00AB 447 RETCENTER,-
00AB 448 PACKACK,-
00AB 449 SEARCH,-
00AB 450 AVAILABLE,-
00AB 451 READPRESET,-
00AB 452 SENSECHAR,-
00AB 453 SETCHAR,-
00AB 454 SENSEMODE,-
00AB 455 SETMODE,-

:FUNCTION DECISION TABLE
:LEGAL FUNCTIONS
:NO OPERATION
:UNLOAD VOLUME
:SEEK CYLINDER
:RECALIBRATE
:DRIVE CLEAR
:RELEASE PORT
:OFFSET HEADS
:RETURN HEADS TO CENTERLINE
:PACK ACKNOWLEDGE
:SEARCH FOR SECTOR
:READ IN PRESET
:SENSE CHARACTERISTICS
:SET CHARACTERISTICS
:SENSE MODE
:SET MODE
:WRITE CHECK
:WRITE HEADER AND DATA
:READ HEADER AND DATA
:WRITE CHECK HEADER AND DATA
:READ LOGICAL BLOCK
:WRITE LOGICAL BLOCK
:READ PHYSICAL BLOCK
:WRITE PHYSICAL BLOCK
:READ VIRTUAL BLOCK
:WRITE VIRTUAL BLOCK
:UNIT AVAILABLE
:ACCESS FILE AND/OR FIND DIRECTORY ENTRY
:ACP CONTROL FUNCTION
:CREATE FILE AND/OR CREATE DIRECTORY ENTRY
:DEACCESS FILE
:DELETE FILE AND/OR DIRECTORY ENTRY
:MODIFY FILE ATTRIBUTES
:MOUNT VOLUME
:BUFFERED I/O FUNCTIONS
:NO OPERATION
:UNLOAD VOLUME
:SEEK CYLINDER
:RECALIBRATE
:DRIVE CLEAR
:RELEASE PORT
:OFFSET HEADS
:RETURN HEADS TO CENTERLINE
:PACK ACKNOWLEDGE
:SEARCH FOR SECTOR
:UNIT AVAILABLE
:READ IN PRESET
:SENSE CHARACTERISTICS
:SET CHARACTERISTICS
:SENSE MODE
:SET MODE
```

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

```

0137 507 .SBTTL START I/O OPERATION
0137 508 :
0137 509 : DB_STARTIO - START I/O OPERATION ON DEVICE UNIT
0137 510 :
0137 511 : THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
0137 512 :
0137 513 : INPUTS:
0137 514 :
0137 515 : R3 = ADDRESS OF I/O PACKET.
0137 516 : R5 = UCB ADDRESS OF DEVICE UNIT.
0137 517 :
0137 518 : OUTPUTS:
0137 519 :
0137 520 : FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
0137 521 : RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
0137 522 : THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
0137 523 :-
0137 524 :
0137 525 DB_STARTIO: ;START I/O OPERATION
0080 C5 0081 C5 90 0137 526 MOVW UCBSB_ERTMAX(R5),UCBSB_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
8A 013E 527 BICB #<ERL_M MEDOFF!- ; Clear flags used to signal medium
013F 528 ERL_M_ECC DEFER>,- ; offline and ECC correction deferred
00D2 C5 05 013F 529 UCBSB_DB_ERL(R5) ; at start of function.
009A C5 20 A3 B0 0143 530 MOVW IRPSW_FUNC(R3),UCBSW_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
50 38 A3 D0 0149 531 MOVL IRPSL_MEDIA(R3),R0 ;GET PARAMETER LONGWORD
014D 532
014D 533 :
014D 534 : MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
014D 535 :
014D 536 :
51 06 00 EF 014D 537 10$: EXTZV #IRPSV_FCODE,#IRPSS_FCODE,- ;EXTRACT I/O FUNCTION CODE
51 20 A3 0150 538 IRPSW_FUNC(R3),R1 ;
51 02 91 0153 539 CMPB #IOS_SEEK,R1 ;SEEK FUNCTION?
51 2F 13 0156 540 BEQL 20$ ;IF EQL YES
51 06 91 0158 541 CMPB #IOS_OFFSET,R1 ;OFFSET FUNCTION?
51 31 13 015B 542 BEQL 30$ ;IF EQL YES
51 09 91 015D 543 CMPB #IOS_SEARCH,R1 ;SEARCH FUNCTION?
51 33 13 0160 544 BEQL 40$ ;IF EQL YES
51 11 91 0162 545 CMPB #IOS_AVAILABLE,R1 ;AVAILABLE function?
00BC C5 50 D0 0165 546 BEQL 15$ ;Branch if yes.
51 18 91 016C 547 MOVL R0,UCBSW_DA(R5) ;STORE PARAMETER LONGWORD
51 29 1A 016F 548 CMPB #IOS_WRITECHECKH,R1 ;DISJOINT FUNCTION CODE?
51 09 A2 0171 549 BGTRU 50$ ;IF GTRU NO
24 11 0174 550 SUBW #IOS_WRITECHECKH-IOS_READHEAD-1,R1 ;CONVERT TO DENSE FUNCTION CODE
0176 551 BRB 50$ ;
0176 552 :
0176 553 :
0176 554 : AVAILABLE FUNCTION - Clear software volume valid bit & exit
0176 555 :
64 A5 080C 3F AA 0176 556 15$: BICW #UCBSM_VALID, UCBSW_STS(R5) ;Clear software volume valid bit.
50 01 3C 017C 557 MOVZWL #SS$_NORMAL, R0 ;Setup success status for zero
51 51 D4 017F 558 CLRL R1 ;bytes transfered operation,
0181 559 REOCOM ;and complete request.
0187 560 :
0187 561 :
0187 562 : SEEK FUNCTION - SET CYLINDER ADDRESS
0187 563 :

```

```

00BE C5 50 80 0187 564
OC 11 0187 565 20$: MOVW R0,UCBSW_DC(R5) ;SET CYLINDER ADDRESS
018C 566 BRB 50$ ;
018E 567 ;
018E 568 ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
018E 569 ;
018E 570 ;
018E 571 ;
00CB C5 50 90 018E 572 30$: MOVB R0,UCBSW_OFFSET(R5) ;SET OFFSET VALUE
05 11 0193 573 BRB 50$ ;
0195 574 ;
0195 575 ; SEARCH FUNCTION - SET SECTOR ADDRESS
0195 576 ;
0195 577 ;
00BC C5 50 90 0195 578 40$: MOVB R0,UCBSW_DA(R5) ;SET SECTOR ADDRESS
019A 580 ;
019A 581 ; FINISH PREPROCESSING
019A 582 ;
019A 583 ;
019A 584 ;
0092 C5 51 90 019A 585 50$: MOVB R1,UCBSB_FEX(R5) ;SAVE FUNCTION DISPATCH INDEX
54 24 A5 D0 019F 586 MOVL UCBSL_CRB(R5),R4 ;GET ADDRESS OF CRB
54 2C B4 D0 01A3 587 MOVL @CRB$C_INTD+VEC$SL_IDB(R4),R4 ;GET FIRST CONTROLLER CSR ADDRESS
00 68 A5 00 E4 01A7 588 BBSC #UCBSV_ECC,UCBSW_DEVSTS(R5),FDISPATCH ;CLEAR ECC CORRECTION MADE
01AC 589 ;
01AC 590 ; CENTRAL FUNCTION DISPATCH
01AC 591 ;
01AC 592 ;
01AC 593 ;
01AC 594 FDISPATCH: ;FUNCTION DISPATCH
53 58 A5 D0 01AC 595 MOVL UCBSL_IRP(R5),R3 ;RETRIEVE ADDRESS OF I/O PACKET
0D 2A A3 08 E0 01B0 596 BBS #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
08 64 A5 08 E0 01B5 597 BBS #UCBSV_VALID,UCBSW_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
50 0254 8F 3C 01BA 598 MOVZWL #SS$VOLINV,R0 ;SET VOLUME INVALID STATUS
0582 31 01BF 599 BRW RESETXFR ;
01C2 600 ;
01C2 601 ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
01C2 602 ;
01C2 603 ;
01C2 604 ;
50 0092 C5 9A 01C2 605 10$: MOVZBL UCBSB_FEX(R5),R0 ;GET DISPATCH FUNCTION CODE
00C9 C5 10 90 01C7 606 MOVB #RP_OF_M_FMT/256,UCBSW_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
00CB C5 01 90 01CC 607 MOVB #1,UCBSB_OFFRTC(R5) ;SET INITIAL OFFSET RETRY COUNT
00CA C5 94 01D1 608 CLRB UCBSB_OFFNDX(R5) ;CLEAR INITIAL OFFSET TABLE INDEX
01D5 609 CASE R0,- ;DISPATCH TO FUNCTION HANDLING ROUTINE
01D5 610 NOP,- ;NO OPERATION
01D5 611 UNLOAD,- ;UNLOAD VOLUME
01D5 612 SEEK,- ;SEEK CYLINDER
01D5 613 RECAL,- ;RECALIBRATE
01D5 614 DRVCLR,- ;DRIVE CLEAR
01D5 615 RELEASE,- ;RELEASE PORT
01D5 616 OFFSET,- ;OFFSET HEADS
01D5 617 RETCENTER,- ;RETURN HEADS TO CENTER
01D5 618 PACKACK,- ;PACK ACKNOWLEDGE
01D5 619 SEARCH,- ;SEARCH FOR SECTOR
01D5 620 WRITECHECK,- ;WRITE CHECK DATA

```

```

01D5 621 WRITEDATA,- ;WRITE DATA
01D5 622 READATA,- ;READ DATA
01D5 623 WRITEHEAD,- ;WRITE HEADER AND DATA
01D5 624 READHEAD,- ;READ HEADER AND DATA
01D5 625 WRITECHECKH,- ;WRITE CHECK HEADER AND DATA
01D5 626 READPRESET- ;READ IN PRESET
01D5 627 >
01FB 628
01FB 629 :
01FB 630 : IOS UNLOAD INDICATES THE UNIT IS NOT MOUNTED SO WE CLEAR SOFTWARE VOLUME
01FB 631 : VALID BEFORE EXECUTING THE OPERATION. IOS PACKACK INDICATES THAT SOFTWARE
01FB 632 : IS READY TO MOUNT THE VOLUME SO WE SET SOFTWARE VOLUME VALID BEFORE
01FB 633 : EXECUTING THE OPERATION.
01FB 634 :
01FB 635 UNLOAD:
64 A5 0800 BF AA 01FB 636 BICW #UCBSM_VALID, UCBSW_STS(R5) ;Clear software volume valid bit.
06 11 0201 637 BRB NOP ;Proceed with the unload operation.
0203 638
0203 639 PACKACK:
64 A5 0800 BF A8 0203 640 BISW #UCBSM_VALID, UCBSW_STS(R5) ;Set software volume valid bit.
0209 641 : BRB NOP ;Proceed with the unload operation.
0209 642 :
0209 643 :
0209 644 : NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
0209 645 : RETURN TO CENTER LINE, SEARCH, AND READ IN PRESET
0209 646 :
0209 647 :
0209 648 NOP: ;NO OPERATION
0209 649 SEEK: ;SEEK CYLINDER
0209 650 RECAL: ;RECALIBRATE
0209 651 DRVCLR: ;DRIVE CLEAR
0209 652 RELEASE: ;RELEASE PORT
0209 653 OFFSET: ;OFFSET READ HEADS
0209 654 RETCENTER: ;RETURN TO CENTERLINE
0209 655 SEARCH: ;SEARCH FOR SECTOR
0209 656 READPRESET: ;READIN PRESET
73 11 0209 657 EXFUNC RETRY ;EXECUTE HOUSEKEEPING FUNCTION
020E 658 BRB NORMAL ;
0210 659
0210 660 :
0210 661 : WRITE CHECK DATA AND WRITE CHECK HEADER AND DATA
0210 662 :
0210 663 :
0210 664 WRITECHECK: ;WRITE CHECK DATA
0210 665 WRITECHECKH: ;WRITE CHECK HEADER AND DATA
009A C5 4000 BF AA 0210 666 BICW #IOSM_DATACHECK,UCBSW_FUNC(R5) ;CLEAR DATA CHECK REQUEST
0217 667
0217 668 :
0217 669 : WRITE DATA, WRITE HEADER AND DATA, WRITE CHECK DATA, AND WRITE CHECK HEADER
0217 670 : AND DATA
0217 671 :
0217 672 :
0217 673 WRITEDATA: ;WRITE DATA
0217 674 WRITEHEAD: ;WRITE HEADER AND DATA
00C9 C5 08 88 0217 675 BISB #RP_OF_M_EC1/256,UCBSW_OFFSET+1(R5) ;INHIBIT ECC CORRECTION
021C 676
021C 677 :

```

```

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

```



```

50 01 3C 0283 735      MOVZWL  S^#SSS_NORMAL,R0      ;SET NORMAL COMPLETION STATUS
      01F6 31 0286 736 CHECKXT:
      0286 737      BRW      FUNCXT      :
      0289 738      :
      0289 739      :
      0289 740      : TRANSFER ENDED WITH A RETRIABLE ERROR
      0289 741      :
      0289 742      :
      0289 743 TRANXT:      :TRANSFER EXIT
0093 C5 0B 91 0289 744      CMPB      #CDF_WRITEDATA,UCBSB_CEX(R5) ;WRITE DATA FUNCTION?
      1B 13 028E 745      BEQL      RETRY?      ;IF EQL YES
0093 C5 0D 91 0290 746      CMPB      #CDF_WRITEHEAD,UCBSB_CEX(R5) ;WRITE HEADER FUNCTION?
      14 13 0295 747      BEQL      RETRY      ;IF EQL YES
51 00064F74 8F D3 0297 748      BITL      #MBASM_SR_DLT!-      ;DATA LATE OR,
      029E 749      MBASM_SR_INVMAP!-      ;INVALID MAP REGISTER OR,
      029E 750      MBASM_SR_MAPPE!-      ;MAP REGISTER PARITY ERROR OR,
      029E 751      MBASM_SR_MCPE!-      ;MASSBUS CONTROL PARITY ERROR OR,
      029E 752      MBASM_SR_SPE!-      ;SILO PARITY ERROR OR,
      029E 753      MBASM_SR_MDPE!-      ;MASSBUS DATA PARITY ERROR OR,
      029E 754      MBASM_SR_MXF!-      ;MISSED TRANSFER OR,
      029E 755      MBASM_SR_NED!-      ;NONEXISTENT DISK OR,
      029E 756      MBASM_SR_RDS!-      ;READ DATA SUBSTITUTE OR,
      029E 757      MBASM_SR_WCKLWR!-      ;WRITE CHECK LOWER BYTE OR,
      029E 758      MBASM_SR_WCKUPR,R1      ;WRITE CHECK UPPER BYTE?
      029E 759      BNEQ      RETRY      ;IF NEQ YES - RETRY FUNCTION
      OA 52 0B 12 029E 759      BNEQ      RETRY      ;IF NEQ YES - RETRY FUNCTION
52 208F 8F E0 02A0 760      BBS      #RP_ER1_V_HCRC,R2,ECC      ; First check HCRC. If bad go to ECC.
      02A4 761      BITW      #RP_ER1_M_FER!-      ;FORMAT ERROR OR,
      02A9 762      RP_ER1_M_HCE!-      ;Header Compare Error.
      02A9 763      RP_ER1_M_OPI!-      ;OPERATION INCOMPLETE OR,
      02A9 764      RP_ER1_M_PAR!-      ;PARITY ERROR OR,
      02A9 765      RP_ER1_M_WCF,R2      ;WRITE CLOCK FAIL?
      03 13 02A9 766      BEQL      ECC      ;IF EQL NO
      0110 31 02AB 767 RETRY:      :
      02AB 768      BRW      RETRYERR      ;RETRIABLE ERROR
      02AE 769      :
      02AE 770      :
      02AE 771      : ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
      02AE 772      :
      02AE 773      :
      02AE 774 ECC:      :ECC CORRECTION
51 7E A5 00C0 C5 A1 02AE 775      ADDW3      UCBSW_BCR(R5), -      ; Compute bytes transfered then
50 51 FFFF01FF 8F CB 02B5 776      BICL3      UCBSW_BCNT(R5), R1      ; clear byte offset bits and
      6D 13 02BD 777      BEQL      OFF      ; convert result to a longword.
      51 01FF 8F B3 02BF 778      BITW      #*X1FF, R1      ; Branch if whole blocks xfered is zero.
      66 12 02C4 779      BNEQ      OFF      ; Was a partial block transfered?
      52 0180 8F B3 02C6 780      BITW      #RP_ER1_M_HCE!-      ; Branch if partial block transfered.
      02CB 781      RP_ER1_M_HCRC, R2      ; Was there an error while processing
      02CB 782      10$      ; the header?
      07 12 02CB 783      BNEQ      10$      ; Branch if header error.
50 00000200 8F C2 02CD 784      SUBL2      #512, R0      ; Else, reduce bytes xfered by a block.
      52 11C0 8F B3 02D4 785      BITW      #RP_ER1_M_DTE!-      ; For: DRIVE TIMING ERROR
      02D9 786      RP_ER1_M_ECH!-      ; ECC HARD ERROR
      02D9 787      RP_ER1_M_HCE!-      ; HEADER COMPARE ERROR
      02D9 788      RP_ER1_M_HCRC,R2      ; HEADER CRC ERROR
      51 12 02D9 789      BNEQ      OFF      ; perform offset recovery.
4B 00C8 C5 0B E0 02DB 790      BBS      #RP_OF_V_ECI,UCBSW_OFFSET(R5),OFF ; Branch if ECC inhibited.
      7E 52 7D 02E1 791      MOVQ      R2,-(SP)      ; Save work registers.

```

```

52 00C6 C5 0B 00 EA 02E4 792 FFS #0,#11,UCBSW_EC2(R5),R2 ; Find the first error bit in the ECC
      02EB 793 ; pattern.
      53 0A 52 C3 02EB 794 SUBL3 R2,#10,R3 ; Get the number of error bits
      02EF 795 ; remaining in the pattern.
      09 15 02EF 796 BLEQ 20$ ; Branch if no other bits in pattern.
      52 D6 02F1 797 INCL R2 ; Point of next bit in pattern.
52 00C6 C5 53 52 EF 02F3 798 EXTZV R2,R3,UCBSW_EC2(R5),R2 ; Is there more than one error bit set?
      0C BA 02FA 799 20$: POPR #*M<R3,R2> ; Restore work registers without
      ; affecting flags.
      29 1A 02FC 800 BGTRU DEFER_ECC ; If more than one error bit set, don't
      02FE 801 ; apply ECC correction.
      02FE 802 ;
      02FE 803 ;
      02FE 804 ; APPLY_ECC -
      02FE 805 ;
      02FE 806 ; Apply ECC correction to correct a single bit error.
      02FE 807 ;
      02FE 808 ;
      02FE 809 APPLY_ECC:
      7E 51 3C 02FE 810 MOVZWL R1, -(SP) ; Save total bytes transfered, inc. ECC.
00000000'GF 16 0301 811 JSB G*IOCS$APPLYECC ; APPLY ECC CORRECTION
      50 8ED0 0307 812 POPL R0 ; RETRIEVE TRANSFERED BYTE COUNT
00000000'GF 16 030A 813 JSB G*IOCS$UPDATRANSF ; UPDATE TRANSFER PARAMETERS
      00CA C5 94 0310 814 CLR B UCBSW_OFFNDX(R5) ; Reset offset table index.
      0314 815 EXFUNC FATALERR,F RETCENTER ; Return to centerline.
      7E A5 B5 031C 816 TSTW UCBSW_BCNT(R5) ; ANY MORE TO TRANSFER?
      03 13 031F 817 BEQL 20$ ; IF EQL NO
      FFOC 31 0321 818 BRW TRANNOC ; TRANSFER NEXT SEGMENT
      FF13 31 0324 819 20$: BRW DATACHECK ; CHECK FOR WRITE CHECK
      0327 820 ;
      0327 821 ;
      0327 822 ; DEFER_ECC -
      0327 823 ;
      0327 824 ; Don't apply ECC correction for multiple bit errors unless the error cannot
      0327 825 ; be recovered with offset retries.
      0327 826 ;
      0327 827 ;
      0327 828 DEFER_ECC:
      00D2 C5 88 0327 829 BIS B #ERL M ECC DEFER,- ; Set flag to indicate that ECC
      0329 830 UCBSW_DB_ERL(R5) ; can be used if offset recovery fails.
      032C 831 ;
      032C 832 ;
      032C 833 ; OFF - OFFSET RECOVERY
      032C 834 ;
      032C 835 ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER COMPARE, OR ECC
      032C 836 ; HARD ERROR IS DETECTED ON A READ FUNCTION.
      032C 837 ;
      032C 838 ;
      032C 839 OFF: ; OFFSET RECOVERY
      50 D5 032C 840 TSTL R0 ; ANY GOOD DATA TRANSFERED?
      2E 13 032E 841 BEQL 20$ ; IF EQL NO
      0330 842 ;
      0330 843 ;
      0330 844 ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
      0330 845 ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
      0330 846 ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
      0330 847 ; POINT OF ERROR.
      0330 848 ;

```

```

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

```

```

07  BA 03C0 906 RELCHAN ; Release channel before possible RECAL
      03C6 907 POPR #*M<R0,R1,R2> ; Restore error status registers.
      03C8 908
      03C8 909
      03C8 910 : Here we will do a RECAL if we had either a Seek Incomplete or a Header
      03C8 911 : Compare Error.
      03C8 912
      03C8 913
      04 00CC OE E0 03C8 914 BBS #RP ER3 V SKI,- ; If Seek Incomplete
      08 52 07 E1 03CA 915 UCBSW DB ER3(R5),10$ ; then go do RECAL.
      0080 C5 97 03CE 916 BBC #RP ER1 V HCE,R2,20$ ; If NOT HCE, then branch around RECAL.
      08 13 03D2 917 10$: EXFUNC FATALERR,F RECAL ; Do RECAL for SKI or HCE.
      03DA 918 20$: DECB UCBSB ER1CNT(R5) ; ANY RETRIES LEFT?
      03DE 919 BEQL FATALERR ; IF EQL NO
      03E0 920 EXFUNC FATALERR,F_DRVCLR ; Issue drive clear before retrying.
      FDC1 31 03E8 921 BRW FDISPATCH
      03EB 922
      03EB 923
      03EB 924 : FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
      03EB 925 : INHIBITED, OR FINAL OFFSET TRIED
      03EB 926
      03EB 927
      03EB 928 FATALERR: ; FATAL ERROR - SET STATUS
      50 OE 50 0C E0 03EB 929 BBS #RP DS V MOL,R0,10$ ; Branch if medium is online.
      50 01A4 8F 3C 03EF 930 MOVZWL #SS$ MEDOFFL,R0 ; Otherwise, set medium offline status.
      0800 8F AA 03F4 931 BITW #UCBSM_VALID,- ; clear software volume valid,
      64 A5 03F8 932
      0082 31 03FA 933 BRW FUNCXT ; and branch to common exit.
      79 50 06 E1 03FD 934 10$: BBC #RP DS V VV,R0,20$ ; IF CLR, VOLUME INVALID
      50 023C 8F 3C 0401 935 MOVZWL #SS$ UNSAFE,R0 ; SET DRIVE UNSAFE STATUS
      75 52 0E E0 0406 936 BBS #RP ER1 V UNS,R2,FUNCXT ; IF SET, DRIVE UNSAFE
      50 02D4 8F 3C 040A 937 MOVZWL #SS$ OPINCOMPL,R0 ; SET OPERATION INCOMPLETE STATUS
      6C 52 0D E0 040F 938 BBS #RP ER1 V OPI,R2,FUNCXT ; IF SET, OPERATION INCOMPLETE
      50 00BC 8F 3C 0413 939 MOVZWL #SS$ FORMAT,R0 ; SET FORMAT ERROR STATUS
      63 52 04 E0 0418 940 BBS #RP ER1 V FER,R2,FUNCXT ; IF SET, FORMAT ERROR
      50 025C 8F 3C 041C 941 MOVZWL #SS$ WRITECK,R0 ; SET WRITE LOCK ERROR STATUS
      5A 52 0B E0 0421 942 BBS #RP ER1 V WLE,R2,FUNCXT ; IF SET, WRITE LOCK ERROR
      50 0134 8F 3C 0425 943 MOVZWL #SS$ IVADDR,R0 ; SET INVALID DISK ADDRESS STATUS
      52 0600 8F B3 042A 944 BITW #RP ER1 M AOE!- ; DISK ADDRESS OVERFLOW OR,
      042F 945 RP ER1 M_TAE,R2 ; INVALID DISK ADDRESS ERROR?
      50 008C 8F 12 042F 946 BNEQ FUNCXT ; IF NEQ YES
      52 1027 8F 3C 0431 947 MOVZWL #SS$ DRVERR,R0 ; SET DRIVE ERROR STATUS
      0436 948 BITW #RP ER1 M DTE!- ; DRIVE TIMING ERROR OR,
      0438 949 RP ER1 M_ILF!- ; ILLEGAL FUNCTION OR,
      0438 950 RP ER1 M_ILR!- ; ILLEGAL REGISTER OR,
      0438 951 RP ER1 M_RMR!- ; REGISTER MODIFY REFUSE OR,
      0438 952 RP ER1 M_WCF,R2 ; WRITE CLOCK FAIL ERROR?
      50 01F4 8F 12 0438 953 BNEQ FUNCXT ; IF NEQ YES
      52 8140 8F 3C 043D 954 MOVZWL #SS$ PARITY,R0 ; Set parity error status.
      0442 955 BITW #RP ER1 M DCK!- ; Data check error or,
      0447 956 RP ER1 M_ECH!- ; ECC hard error or,
      0447 957 RP ER1 M_HCRC,R2 ; header CRC error?
      50 0054 8F 12 0447 958 BNEQ FUNCXT ; Branch if so.
      52 0088 8F 3C 0449 959 MOVZWL #SS$ CTRLERR,R0 ; Set fatal controller error status.
      044E 960 BITW #RP ER1 M_HCE!- ; Header compare error or,
      0453 961 RP ER1 M_PAR,R2 ; parity error?
      2A 12 0453 962 BNEQ FUNCXT ; Branch if so.

```

```
51 00024064 8F D3 0455 963 BITL #MBASH SR MAPPE!- :MAP PARITY ERROR OR,
      045C 964 MBASH SR MCPE!- :MASSBUS CONTROL PARITY ERROR OR,
      045C 965 MBASH SR SPE!- :SILO PARITY ERROR OR,
      045C 966 MBASH SR MDPE!- :MASSBUS DATA PARITY ERROR OR,
      045C 967 MBASH SR_RDS,R1 :READ DATA SUBSTITUTE?
      21 12 045C 968 BNEQ FUNCXT :IF NEQ YES
50 005C 8F 3C 045E 969 MOVZWL #SS$ DATACHECK,R0 :SET DATA CHECK ERROR STATUS
51 0600 8F B3 0463 970 BITW #MBASH SR WCKLWR!- :WRITE CHECK ERROR LOWER BYTE OR,
      0468 971 MBASH SR_WCKUPR,R1 :WRITE CHECK ERROR UPPER BYTE?
      15 12 0468 972 BNEQ FUNCXT :IF NEQ YES
50 01C4 8F 3C 046A 973 MOVZWL #SS$ NONEXDRV,R0 :SET NONEXISTENT DRIVE STATUS
OC 51 12 E0 046F 974 BBS #MBASH SR_NED,R1, FUNCXT :IF SET, NONEXISTENT DRIVE
50 0054 8F 3C 0473 975 MOVZWL #SS$ CTRLERR,R0 :SET CONTROLLER ERROR STATUS
      05 11 0478 976 BRB FUNCXT :
50 0254 8F 3C 047A 977 20$: MOVZWL #SS$_VOLINV,R0 :SET VOLUME INVALID STATUS
      047F 978
      047F 979
      047F 980 : FUNCTION COMPLETION COMMON EXIT
      047F 981
      047F 982
      047F 983 FUNCXT: :FUNCTION EXIT
      50 DD 047F 984 PUSHL R0 :SAVE FINAL REQUEST STATUS
      00000000 GF 16 0481 985 JSB G^IOC$DIAGBUFILL :FILL DIAGNOSTIC BUFFER IF PRESENT
      0487 986 RELCHAN :RELEASE CHANNEL IF OWNED
      0092 C5 0A 91 048D 987 CMPB #CDF_WRITECHECK,UCB$B_FEX(R5) :DRIVE RELATED FUNCTION?
      13 1A 0492 988 BGTRU 10$ :IF GTRU YES
      0092 C5 10 91 0494 989 CMPB #CDF_READPRESET,UCB$B_FEX(R5) :READIN PRESET FUNCTION?
      0C 13 0499 990 BEQL 10$ :IF EQL YES
      53 58 A5 D0 049B 991 MOVL UCB$L_IRP(R5),R3 :RETRIEVE ADDRESS OF IRP
02 AE 32 A3 00C0 C5 A1 049F 992 ADDW3 UCB$W_BCR(R5),IRP$W_BCNT(R3),2(SP) :CALCULATE BYTES TRANSFERED
      51 D4 04A7 993 10$: CLRL R1 :CLEAR SECOND STATUS LONGWORD
      50 BED0 04A9 994 POPL R0 :RETRIEVE FINAL REQUEST STATUS
      53 0091 C5 9A 04AC 995 MOVZBL UCB$B_SLAVE+1(R5),R3 :Get drive offset constant
      53 0400 C443 DE 04B1 996 MOVAL MBASH_ERB(R4)[R3],R3 :Get address of driver registers
      63 09 9A 04B7 997 MOVZBL #F_DRVCLR!1,RP_CS1(R3) :Issue drive clear before release
      63 0B 9A 04BA 998 MOVZBL #F_RELEASE!1,RP_CS1(R3) :Release port
      04BD 999 REQCOM :COMPLETE REQUEST
```

```
04C3 1001 .SBTTL RP04/05/06 HARDWARE FUNCTION EXECUTION
04C3 1002
04C3 1003 FEX - RP04/05/06 HARDWARE FUNCTION EXECUTION
04C3 1004
04C3 1005 THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
04C3 1006 SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
04C3 1007 SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
04C3 1008 INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. IF THE FUNCTION IS AN
04C3 1009 IMMEDIATE FUNCTION CONTROL RETURNS IMMEDIATELY. ELSE THE RETURN ADDRESS
04C3 1010 IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
04C3 1011 RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
04C3 1012
04C3 1013 INPUTS:
04C3 1014
04C3 1015 R0 = FUNCTION TABLE DISPATCH INDEX.
04C3 1016 R3 = ADDRESS OF DRIVE CONTROL STATUS REGISTER 1.
04C3 1017 R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
04C3 1018 R5 = DEVICE UNIT UCB ADDRESS.
04C3 1019
04C3 1020 00(SP) = RETURN ADDRESS OF CALLER.
04C3 1021 04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
04C3 1022
04C3 1023 IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
04C3 1024 A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
04C3 1025
04C3 1026 OUTPUTS:
04C3 1027
04C3 1028 THERE ARE FOUR EXITS FROM THIS ROUTINE:
04C3 1029
04C3 1030 1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
04C3 1031 OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
04C3 1032 ERROR ROUTINE.
04C3 1033
04C3 1034 2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
04C3 1035 ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
04C3 1036 INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
04C3 1037
04C3 1038 3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
04C3 1039 OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
04C3 1040 IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
04C3 1041
04C3 1042 4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
04C3 1043 DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
04C3 1044
04C3 1045 IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
04C3 1046
04C3 1047 IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
04C3 1048 THE GENERAL REGISTERS R0, R1, AND R2, AND THE UCB.
04C3 1049
04C3 1050 R0 = DRIVE STATUS REGISTER.
04C3 1051 R1 = MBA STATUS REGISTER.
04C3 1052 R2 = DRIVE ERROR REGISTER 1.
04C3 1053
04C3 1054 UCBSW_EC1(R5) = ECC POSITION REGISTER.
04C3 1055 UCBSW_EC2(R5) = ECC PATTERN REGISTER.
04C3 1056 UCBSW_BCR(R5) = BYTE COUNT REGISTER.
04C3 1057
```

```

04C3 1058
04C3 1059 FEX:
04C3 1060 :FUNCTION EXECUTOR
0093 C5 50 8E00 04C8 1061 :SAVE DRIVER PC VALUE
53 0091 C5 9A 04CD 1062 :SAVE CASE INDEX
38 A5 53 0400 C443 DE 04D2 1063 :GET DRIVE OFFSET CONSTANT
00008000 8F D3 04D8 1064 :GET ADDRESS OF DRIVE REGISTERS
50 0093 C5 9A 04E0 1065 BITL #DEV$M_DUA,UCBSL_DEVCHAR(R5) :DUAL PORTED DRIVE?
04E2 1066 GO: BNEQ SEIZE :IF NEG, YES
04E7 1067 MOVZBL UCBSB_CEX(R5),R0 :RESTORE CASE INDEX (FUNC. CODE)
04E7 1068 CASE RO, <- :DISPATCH TO PROPER FUNCTION ROUTINE
04E7 1069 POSIT,- :SEEK CYLINDER
04E7 1070 EXFNC,- :RECALIBRATE
04E7 1071 IMMED,- :DRIVE CLEAR
04E7 1072 IMMED,- :RELEASE DRIVE
04E7 1073 EXFNC,- :OFFSET HEADS
04E7 1074 EXFNC,- :RETURN TO CENTERLINE
04E7 1075 IMMED,- :PACK ACKNOWLEDGE
04E7 1076 POSIT,- :SEARCH FOR SECTOR
04E7 1077 XFER,- :WRITE CHECK
04E7 1078 XFER,- :WRITE DATA
04E7 1079 XFER,- :READ DATA
04E7 1080 XFER,- :WRITE HEADER AND DATA
04E7 1081 XFER,- :READ HEADER AND DATA
04E7 1082 IMMED,- :WRITE CHECK HEADER AND DATA
04E7 1083 SEARCHA,- :READIN PRESET
04E7 1084 >,LIMIT=#CDF_SEEK :SEARCH AHEAD FOR SECTOR
050B 1085 :
050B 1086 :
050B 1087 : IMMEDIATE FUNCTION EXECUTION
050B 1088 :
050B 1089 : FUNCTIONS INCLUDE:
050B 1090 :
050B 1091 : NO OPERATION,
050B 1092 : UNLOAD VOLUME,
050B 1093 : DRIVE CLEAR,
050B 1094 : RELEASE PORT,
050B 1095 : READ IN PRESET, AND
050B 1096 : PACK ACKNOWLEDGE.
050B 1097 :
050B 1098 : THESE FUNCTIONS ARE EXECUTED IMMEDIATELY AND THE FINAL DEVICE REGISTERS
050B 1099 : ARE RETURNED TO THE CALLER.
050B 1100 :
050B 1101 :
050B 1102 IMMED: : IMMEDIATE FUNCTION EXECUTION
050B 1103 DSBINT :DISABLE INTERRUPTS
09 64 A5 05 E0 0511 1104 BBS #UCBSV_POWER,UCBSW_STS(R5),10$ :IF SET, POWER HAS FAILED
63 FB6B CF40 9A 0516 1105 MOVZBL #F_DRV[LR!1,RP_CST(R3) :CLEAR DRIVE ERRORS
00R2 31 0519 1106 MOVZBL FTAB[RO],RP_CST(R3) :EXECUTE FUNCTION
051F 1107 10$: BRW ENBXIT :
0522 1108 :
0522 1109 :
0522 1110 : ATTEMPT TO SEIZE THE PORT ON A DUAL PORTED DISK.
0522 1111 :
0522 1112 :
BA 00D2 C5 01 E1 0522 1113 SEIZE: BBC #ERL_V_DUALPORT,- :IF CLEAR, NO DUAL PORT KIT IN DRIVE
0528 1114 UCBSB_DB_ERL(R5),GO

```

```

00000100 04 A3 D4 0528 1115 DSBINT ;DISABLE INTERRUPTS
00000100 04 8F D3 052E 1116 CLRL RP_DS(R3) ;ATTEMPT TO SEIZE PORT
00000100 04 A3 0531 1117 BITL #RP_DS_M DPR,- ;DID WE SEIZED THE PORT?
00000100 04 12 12 0537 1118 RP_DS(R3)
00000100 0539 1119 BNEQ 2$ ;IF NEQ, WE SEIZED THE PORT
00000100 053B 1120 WFIKPCM RETREG,#15 ;LETS WAIT FOR THE PORT, ELSE TIMEOUT
00000100 0545 1121 IOFORK ;CREATE FORK PROCESS
00000100 95 11 054B 1122 BRB GO ;LETS CONTINUE WE HAVE THE PORT
00000100 90 11 054D 1123 2$: ENBINT ;ENABLE INTERRUPTS
00000100 0550 1124 BRB GO ;LETS CONTINUE WE HAVE THE PORT
00000100 0552 1125
00000100 0552 1126 : SEARCH AHEAD FUNCTION EXECUTION
00000100 0552 1127 :
00000100 0552 1128 : THIS FUNCTION MINIMIZES ROTATIONAL LATENCY BY SEARCHING FOR THE SECTOR THAT IS
00000100 0552 1129 : FOUR SECTORS AHEAD OF THE STARTING SECTOR OF A TRANSFER.
00000100 0552 1130 :
00000100 0552 1131 : THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
00000100 0552 1132 : FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
00000100 0552 1133 : RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
00000100 0552 1134 :
00000100 0552 1135 :
00000100 0552 1136 SEARCHA:
00000100 51 00BC C5 3C 0552 1137 MOVZWL UCBSW_DA(R5),R1 ;GET DESIRED TRACK AND SECTOR ADDRESS
00000100 51 04 82 0557 1138 SUBB #4,R1 ;COMPUTE FOUR SECTORS BEFORE SPECIFIED SECTO
00000100 04 18 055A 1139 BGEQ 10$ ;IF GEQ BEFORE SECTOR ZERO
00000100 51 44 A5 80 055C 1140 ADDB UCBSB_SECTORS(R5),R1 ;CONVERT TO AFTER SECTOR ZERO
00000100 14 A3 51 D0 0560 1141 10$: MOVL R1,RP_DA(R3) ;SET TRACK AND SECTOR ADDRESS
00000100 15 11 0564 1142 BRB ;
00000100 0566 1143 :
00000100 0566 1144 :
00000100 0566 1145 : TRANSFER FUNCTION EXECUTION
00000100 0566 1146 :
00000100 0566 1147 : FUNCTIONS INCLUDE:
00000100 0566 1148 :
00000100 0566 1149 : WRITE CHECK,
00000100 0566 1150 : WRITE CHECK HEADER AND DATA,
00000100 0566 1151 : WRITE DATA,
00000100 0566 1152 : WRITE HEADER AND DATA,
00000100 0566 1153 : READ DATA, AND
00000100 0566 1154 : READ HEADER AND DATA.
00000100 0566 1155 :
00000100 0566 1156 : THE MAP REGISTERS, BYTE COUNT REGISTER, AND VIRTUAL ADDRESS REGISTER ARE
00000100 0566 1157 : LOADED FOLLOWED BY THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS.
00000100 0566 1158 : THE FUNCTION IS INITIATED AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
00000100 0566 1159 : INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
00000100 0566 1160 :
00000100 0566 1161 : IT ASSUMED THAT THE CALLER OWNS THE CHANNEL ON WHICH THE I/O IS TO OCCUR.
00000100 0566 1162 :
00000100 0566 1163 :
00000100 0566 1164 XFER:
00000100 08 A4 00 D2 0566 1165 MCOML #0,MBASL_SR(R4) ;TRANSFER FUNCTION EXECUTION
00000100 50 0093 C5 9A 056A 1166 LOADMBA ;CLEAR MASSBUS ADAPTER ERRORS
00000100 0570 1167 MOVZBL UCBSB_CEX(R5),R0 ;LOAD MAP, BYTE COUNT, AND VIRTUAL ADDRESS
00000100 0575 1168 ;RETRIEVE FUNCTION TABLE INDEX
00000100 0575 1169 :
00000100 0575 1170 : POSITIONING FUNCTION EXECUTION
00000100 0575 1171 :

```

DBDRIVER  
V04-000  
K 8  
15-SEP-1984 23:45:36 VAX/VMS Macro V04-00  
5-SEP-1984 00:11:41 [DRIVER.SRC]DBDRIVER.MAR;1  
Page 21  
(1)



```

0575 1172 : FUNCTIONS INCLUDE:
0575 1173 :
0575 1174 : SEEK CYLINDER, AND
0575 1175 : SEARCH FOR SECTOR.
0575 1176 :
0575 1177 : THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
0575 1178 : FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
0575 1179 : RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
0575 1180 :
0575 1181 :
14 A3 00BC C5 3C 0575 1182 POSIT: : POSITION FUNCTION EXECUTION
0575 1183 : MOVZWL UCBSW_DA(R5),RP_DA(R3) : SET DESIRED TRACK AND SECTOR ADDRESS
28 A3 00B2 C5 3C 0578 1184 LDCYL: :
0578 1185 : MOVZWL UCBSW_DC(R5),RP_DC(R3) : SET DESIRED CYLINDER ADDRESS
0581 1186 :
0581 1187 :
0581 1188 : INTERRUPT WAIT FUNCTION EXECUTION
0581 1189 :
0581 1190 : FUNCTIONS INCLUDE:
0581 1191 :
0581 1192 : OFFSET HEADS,
0581 1193 : RECALIBRATE, AND
0581 1194 : RETURN TO CENTERLINE.
0581 1195 :
0581 1196 : THE OFFSET REGISTER IS LOADED, THE FUNCTION IS INITIATED, AND A WAITFOR
0581 1197 : INTERRUPT IS EXECUTED. WHEN THE INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS
0581 1198 : ARE RETURNED TO THE CALLER.
0581 1199 :
0581 1200 :
0581 1201 EXFNC: : EXECUTE FUNCTION
0581 1202 : DISABLE INTERRUPTS
48 64 A5 05 EU 0587 1203 DSBINT :
63 09 9A 058C 1204 BBS #UCBSV POWER,UCBSW STS(R5),ENBXIT ;IF SET, POWER FAILED
24 A3 00C8 C5 3C 058F 1205 MOVZBL #F DRVCLR!1,RP_CS1(R3) : CLEAR DRIVE ERRORS
52 04 A3 13 78 0595 1206 MOVZWL UCBSW_OFFSET(R5),RP_OF(R3) : SET FORMAT, INHIBIT BITS, AND OFFSET
1E 18 059A 1207 ASHL #31-RP_DS_V_MOL,RP_DS(R3),R2 : MEDIUM ONLINE?
63 FAE8 CF40 9A 059C 1208 BGEQ 10$ : IF SEQ NO
00CE C5 08 A4 D0 05A2 1209 MOVZBL FTAB[R0],RP_CS1(R3) : INITIATE FUNCTION
05A2 1209 WFIKPCM RETREG,#15 : WAITFOR INTERRUPT AND KEEP CHANNEL
1D 11 0582 1211 MOVL MBASL_SR(R4),UCBSL_DB_SR(R5) : SAVE FINAL CONTROLLER STATUS
0588 1212 IOFORK : CREATE FORK PROCESS
0588 1212 BRB RETREG :
058A 1213 :
058A 1214 :
058A 1215 : MEDIUM OFFLINE AT START OF FUNCTION
058A 1216 :
058A 1217 :
50 0093 C5 94 058A 1218 10$: ENBINT : ENABLE INTERRUPTS
4000 8F 3C 058D 1219 CLRB UCBSB_CEX(R5) : FORCE DRIVE FUNCTION
00CE C5 D4 05C1 1220 MOVZWL #RP_DS_M_ERR,R0 : SET DRIVE ERROR
00D2 C5 01 88 05C6 1221 CLRL UCBSL_DB_SR(R5) : CLEAR SAVED MBA STATUS REGISTER
32 11 05CA 1222 BISB #ERL M MEDOFF,- : SET FLAG WHICH INDICATES THAT MEDIUM
05CC 1223 : WAS OFFLINE AT START OF FUNCTION.
05CF 1224 BRB ERROR :
05D1 1225 :
05D1 1226 :
05D1 1227 :
05D1 1228 .ENABL LSB
  
```

```

0117 31 05D1 1229 10$: BRW 120$ ; Branch to special condition handler.
      05D4 1230
      05D4 1231 :
      05D4 1232 : ENABLE INTERRUPTS
      05D4 1233 :
      05D4 1234
      05D4 1235 ENBXIT: ;
      05D4 1236 ENBINT ;ENABLE INTERRUPTS
      05D7 1237
      05D7 1238 :
      05D7 1239 : RETURN REGISTERS
      05D7 1240 :
      05D7 1241
      05D7 1242 RETREG: ;RETURN FINAL DEVICE REGISTERS
      05D7 1243 CVTLW RP_ER3(R3),- ; Save register after operation.
      05DA 1244 UCBSW_DB_ER3(R5)
      05DD 1245 CVTLW RP_ECT(R3),UCBSW_EC1(R5) ;SAVE ECC POSITION REGISTER
      05E3 1246 CVTLW RP_EC2(R3),UCBSW_EC2(R5) ;SAVE ECC PATTERN REGISTER
      05E9 1247 : Here we save the more conservative of the two byte counts contained in
      05E9 1248 : the MBASL_BCR register. The high word of this register is the
      05E9 1249 : (negative of the) number of bytes transferred to or from the
      05E9 1250 : drive, while the low word is the (negative of the) number of
      05E9 1251 : bytes transferred to or memory. On a read, the more conservative
      05E9 1252 : value is that of the number of bytes transferred to memory (low word)
      05E9 1253 : while on a write the more conservative value is the number of
      05E9 1254 : bytes transferred to the drive (high word). Here we deposit
      05E9 1255 : the entire register into a longword in the UCB. If the operation
      05E9 1256 : was a read we leave the value as is. However if the operation
      05E9 1257 : was a write (or anything but a read) we move the high word to
      05E9 1258 : the low word in memory. All other pieces of this driver use the
      05E9 1259 : low word of this longword as the valid byte count.
      05E9 1260
      05E9 1261 MOVL MBASL_BCR(R4),- ; Save entire byte count register
      05EC 1262 UCBSL_DB_BCR(R5) ; in the UCB.
      50 00C0 C5 58 A5 DO 05EF 1263 MOVL UCBSL_IRP(R5),R0 ; Retrieve IRP pointer.
      07 2A A0 01 E0 05F3 1264 BBS #IRPSV_FUNC,- ; If we had a read operation then
      00C2 C5 00C0 C5 07 2A A0 05F5 1265 IRPSW_STS(R0),20$ ; just branch around since all OK.
      05F8 1266 MOVW UCBSL_DB_BCR+2(R5),- ; If NOT read, then copy high word to
      05FC 1267 UCBSW_BCR(R5) ; low order word for later use.
      50 04 A3 DO 05FF 1268 20$:
      0603 1270 ERROR: MOVL RP_DS(R3),R0 ;GET CONTENTS OF DRIVE STATUS REGISTER
      51 00CE C5 DO 0603 1271 MOVL UCBSL_DB_SR(R5),R1 ;RETRIEVE FINAL CONTROLLER STATUS
      52 08 A3 DO 0608 1272 MOVL RP_ERT(R3),R2 ;GET CONTENTS OF DRIVE ERROR REGISTER 1
      64 A5 0060 8F B3 060C 1273 BITW #UCBSM_POWER!- ;POWER FAIL OR DEVICE TIMEOUT?
      0612 1274 UCBSM_TIMEOUT,UCBSW_STS(R5) ;
      0093 C5 BD 12 0612 1275 BNEQ 10$ ;IF NEQ YES - SPECIAL CONDITION
      0093 C5 0A 91 0614 1276 CMPB #CDF_WRITECHECK,UCBSB_CEX(R5) ;DRIVE RELATED FUNCTION?
      0093 C5 2E 1A 0619 1277 BGTRU 30$ ;IF GTRU YES
      0093 C5 10 91 061B 1278 CMPB #CDF_READPRESET,UCBSB_CEX(R5) ;OTHER DRIVE RELATED FUNCTION?
      0620 1279 BLEQU 30$ ;IF EQL YES
      0622 1280
      0622 1281 :
      0622 1282 : CONTROLLER RELATED FUNCTION
      0622 1283 :
      0622 1284 :
      51 00DE5FFF 8F D3 0622 1285 BITL #MBASM_ERROR,R1 ;ANY CONTROLLER ERRORS?

```

```

00000000'GF 13 0629 1286 BEQL 80$ ;IF EQL NO
6C 009A C5 OF 16 062B 1287 JSB G^ERL$DEVICERR ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
51 0008000B 8F D3 0631 1288 BBS #IOSV INHRETRY,UCBSW_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
0637 1289 BITL #MBASM_SR_ERCONF!- ;ERROR CONFIRMATION OR,
063E 1290 MBASM_SR_I$TO!- ;INTERFACE SEQUENCE TIMEOUT OR,
063E 1291 MBASM_SR_PGE!- ;PROGRAMMING ERROR OR,
063E 1292 MBASM_SR_RDTO,R1 ;READ TIMEOUT?
51 00064FF4 63 12 063E 1293 BNEQ 90$ ;IF NEQ YES - FATAL CONTROLLER ERROR
8F D3 0640 1294 BITL #MBASM_SR_DLT!- ;DATA LATE OR,
0647 1295 MBASM_SR_INVMAP!- ;INVALID MAP REGISTER OR,
0647 1296 MBASM_SR_MAPPE!- ;MAP REGISTER PARITY ERROR OR,
0647 1297 MBASM_SR_MBEXC!- ;MASSBUS EXCEPTION OR,
0647 1298 MBASM_SR_MCPE!- ;MASSBUS CONTROL PARITY ERROR OR,
0647 1299 MBASM_SR_SPE!- ;SILO PARITY ERROR OR,
0647 1300 MBASM_SR_MDPE!- ;MASSBUS DATA PARITY ERROR OR,
0647 1301 MBASM_SR_MXF!- ;MISSED TRANSFER OR,
0647 1302 MBASM_SR_NED!- ;NONEXISTENT DRIVE OR,
0647 1303 MBASM_SR_RDS!- ;READ DATA SUBSTITUTE OR,
0647 1304 MBASM_SR_WCKLWR!- ;WRITE CHECK LOWER BYTE OR,
0647 1305 MBASM_SR_WCKUPR,R1 ;WRITE CHECK UPPER BYTE?
31 12 0647 1306 BNEQ 60$ ;IF NEQ YES - RETRIABLE CONTROLLER ERROR
0649 1307
0649 1308 ;
0649 1309 ; DRIVE RELATED FUNCTION
0649 1310 ;
0649 1311
0093 08 91 0649 1312 30$: CMPB #CDF PACKACK,- ; Packack function?
C5 OF 12 064B 1313 UCBSB_CEX(R5) ;
0B 50 OC E0 0650 1314 BNEQ 40$ ; Branch if not.
OF E1 0654 1315 BBS #RP_DS_V_MOL,R0,40$ ; Success if medium online.
28 009A C5 OF E1 0654 1316 BBC #IOSV INHRETRY,- ; Branch if retries not inhibited.
0656 1317 UCBSW_FUNC(R5),65$ ;
44 64 A5 E5 065A 1318 BBCC #UCBSV_VALID,- ; Otherwise, clear software volume
37 50 OE E1 065C 1319 UCBSW_STS(R5),90$ ; valid and take fatal error path.
00C0 C5 7E A5 AE 0663 1320 40$: BBC #RP_DS_V_ERR,R0,80$ ;IF CLR, NO DRIVE ERRORS
35 00D2 C5 E8 0669 1321 50$: MNEGW UCBSW_BCNT(R5),UCBSW_BCR(R5) ;RESET BYTE COUNT - NO TRANSFER
066E 1322 BLBS UCBSB_DB_ERL(R5),90$ ; Do NOT log error if medium was offline
066E 1323 ; at start of function.
00000000'GF 16 066E 1324 JSB G^ERL$DEVICERR ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
29 009A C5 OF E0 0674 1325 BBS #IOSV INHRETRY,UCBSW_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
25 50 OC E1 067A 1326 60$: BBC #RP_DS_V_MOL,R0,90$ ;IF CLR, MEDIUM OFFLINE
21 50 06 E1 067E 1327 BBC #RP_DS_V_VV,R0,90$ ;IF CLR, INVALID VOLUME
52 0E07 8F B3 0682 1328 65$: BITW #RP_ERT_M_AOE!- ;ADDRESS OVERFLOW OR,
0687 1329 RP_ER1_M_I$E!- ;INVALID ADDRESS OR,
0687 1330 RP_ER1_M_ILF!- ;ILLEGAL FUNCTION OR,
0687 1331 RP_ER1_M_ILR!- ;ILLEGAL REGISTER OR,
0687 1332 RP_ER1_M_RMR!- ;REGISTER MODIFY REFUSE OR,
0687 1333 RP_ER1_M_WLE,R2 ;WRITE LOCK ERROR?
52 4000 1A 12 0687 1334 BNEQ 90$ ;IF NEQ YES - FATAL DRIVE ERROR
8F B3 0689 1335 BITW #RP_ER1_M_UN$ ,R2 ; Is the drive unsafe?
16 12 068E 1336 BNEQ 100$ ; Branch if so.
0690 1337
0690 1338 ;
0690 1339 ; RETRIABLE ERROR EXIT
0690 1340 ;
0690 1341
7E 009C D5 32 0690 1342 70$: CVTWL @UCBSL_DPC(R5),-(SP) ;GET BRANCH DISPLACEMENT

```

```

009C C5 8E C0 0695 1343 ADDL (SP)+,UCBSL_DPC(R5) ;CALCULATE RETURN ADDRESS - 2
009C C5 02 C0 069A 1344 80$: ADDL #2,UCBSL_DPC(R5) ;SKIP PAST BRANCH DISPLACEMENT WORD
009C D5 17 069F 1345 JMP @UCBSL_DPC(R5) ;RETURN TO DRIVER
06A3 1346
06A3 1347 ;
06A3 1348 ; FATAL CONTROLLER OR DRIVE ERROR
06A3 1349 ;
06A3 1350
FD45 31 06A3 1351 90$: BRW FATALERR ;
06A6 1352
06A6 1353 ;
06A6 1354 ; Check for unsafe condition and attempt to clear it.
06A6 1355 ;
06A6 1356
06A6 1357 100$: DSBINT ; Disable interrupts.
03 64 05 E1 06AC 1358 BBC #UCBSV_POWER,- ; Branch if no power failure occurred.
FF20 31 06AE 1359 UCBSW_STS(R5),110$ ;
63 09 9A 06B1 1360 BRW ENBXIT ; Otherwise, enable interrupts and
06B4 1361 ; go process error.
06B4 1362 110$: MOVZBL #F_DRVCLR!1,RP_CS1(R3) ; Attempt to clear unsafe condition.
06B7 1363 TIMEWAIT - ; Wait for ten microseconds or until
06B7 1364 TIME = #1,- ; unsafe condition clears.
06B7 1365 BITVAL = #RP_ER1_M_UNSAFE,- ;
06B7 1366 SOURCE = RP_ER1(R3),- ;
06B7 1367 CONTEXT = L,- ;
06B7 1368 SENSE = .FALSE. ;
06DF 1369 ENBINT ; Enable interrupts.
52 08 A3 D0 06E2 1370 MOVL RP_ER1(R3),R2 ; Retrieve error status.
A7 50 E8 06E6 1371 BLBS R0,70$ ; Branch if drive is no longer unsafe.
B8 11 06E9 1372 BRB 90$ ; Otherwise, fatal error.
06EB 1373
06EB 1374 ;
06EB 1375 ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
06EB 1376 ;
06EB 1377 ;
61 64 A5 05 E4 06EB 1378 120$: BBSC #UCBSV_POWER,UCBSW_STS(R5),150$ ;IF SET, POWER FAILURE
06F0 1379
06F0 1380 ;
06F0 1381 ; DEVICE TIME OUT
06F0 1382 ;
06F0 1383 ;
00000000 GF 16 06F0 1384 JSB G^ERL$DEVICTMO ;LOG DEVICE TIME OUT
53 24 A5 D0 06F6 1385 MOVL UCBSL_CRB(R5),R3 ;GET ADDRESS OF CRB
53 2C A3 D0 06FA 1386 MOVL CRBSL_INTD+VECSL_IDB(R3),R3 ;GET ADDRESS OF IDB
04 A3 55 D1 06FE 1387 CML R5,IDBSL_OWNER(R3) ;DEVICE OWN CONTROLLER?
22 12 0702 1388 BNEQ 140$ ;IF NEQ NO
0704 1389 DSBINT ;DISABLE INTERRUPTS
04 06 D0 070A 1390 MOVL #MBASH_CR_ABORT!MBASH_CR_IE,- ;ABORT THE DATA TRANSFER
04 A4 070C 1391 MBASL_CR(R4) ;
070E 1392 WFIKPCB 130$,#15 ;WAIT FOR ABORT AND KEEP CHANNEL
0718 1393 IOFORK ;CREATE FORK PROCESS
071E 1394 130$:
04 A4 01 D0 071E 1395 MOVL #MBASH_CR_INIT,MBASL_CR(R4) ;INITIALIZE ENTIRE MBA
04 A4 04 D0 0722 1396 MOVL #MBASH_CR_IE,MBASL_CR(R4) ;ENABLE DEVICE INTERRUPTS
0726 1397 140$: SETIPL UCBSB_FIPC(R5) ;LOWER TO FORK LEVEL
50 022C 8F 3C 072A 1398 MOVZWL #SS$_TIMEOUT,R0 ;SET DEVICE TIMEOUT STATUS
0080 C5 97 072F 1399 DECB UCBSB_ERTCNT(R5) ;ANY ERROR RETRIES REMAINING?

```

```

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

```

```

0763 1424 .SBTTL RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
0763 1425 :
0763 1426 : DB_DTYPE - RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
0763 1427 :
0763 1428 : THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
0763 1429 : SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
0763 1430 : SET UNIT PARAMETERS.
0763 1431 :
0763 1432 : INPUTS:
0763 1433 :
0763 1434 : R3 = ADDRESS OF DRIVE CONTROL REGISTER.
0763 1435 : R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
0763 1436 : R5 = DEVICE UNIT UCB ADDRESS.
0763 1437 :
0763 1438 : OUTPUTS:
0763 1439 :
0763 1440 : THE DRIVE STATUS REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
0763 1441 :
0763 1442 :
0763 1443 DB_DTYPE: ;CLASSIFY DRIVE TYPE AND SET PARAMETERS
6E 18 A3 DD 0763 1444 PUSHL RP DT(R3) ;READ DRIVE TYPE REGISTER
52 FE00 8F AA 0766 1445 BICW #^C<^X1FF>,(SP) ;CLEAR EXTRANEIOUS BITS
52 FB0D CF 9E 076B 1446 MOVAB DB DTDESC,R2 ;GET ADDRESS OF DESCRIPTOR TABLE
82 6E B1 0770 1447 10$: CMPW (SP),(R2)+ ;DRIVE TYPE MATCH?
OE 13 0773 1448 BEQL 20$ ;IF EQL YES
52 OD C0 0775 1449 ADDL #DB DTDESCLEN-2,R2 ;ADVANCE TO NEXT ENTRY
62 B5 0778 1450 TSTW (R2) ;END OF TABLE?
F4 12 077A 1451 BNEQ 10$ ;IF NEQ NO
64 A5 10 AA 077C 1452 BICW #UCB$M ONLINE,UCB$W_STS(R5) ;SET UNIT OFFLINE
52 OD C2 0780 1453 SUBL #DB DTDESCLEN-2,R2 ;BACK UP TO LAST DRIVE DESCRIPTOR
41 A5 82 90 0783 1454 20$: MOVB (R2)+,UCB$B_DEVTYPE(R5) ;SET DEVICE TYPE
44 A5 82 D0 0787 1455 MOVL (R2)+,UCB$L_DEVDEPEND(R5) ;SET DISK PACK GEOMETRY
00B0 C5 82 D0 078B 1456 MOVL (R2)+,UCB$L_MAXBLOCK(R5) ;SET MAXIMUM BLOCKS PER PACK
00B0 C5 62 D0 0790 1457 MOVL (R2),UCB$L_MEDIA_ID(R5) ;SET MEDIA IDENTIFICATION
8E D5 0795 1458 TSTL (SP)+ ;REMOVE DRIVE TYPE FROM STACK
05 05 0797 1459 RSB ;

```

```

0798 1461      .SBTTL  RP04/05/06 REGISTER DUMP ROUTINE
0798 1462      :
0798 1463      : DB_REGDUMP - RP04/05/06 REGISTER DUMP ROUTINE
0798 1464      :
0798 1465      : THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
0798 1466      : SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
0798 1467      : FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
0798 1468      :
0798 1469      : INPUTS:
0798 1470      :
0798 1471      :     R0 = ADDRESS OF REGISTER SAVE BUFFER.
0798 1472      :     R4 = ADDRESS OF ADAPTER CONFIGURATION REGISTER.
0798 1473      :     R5 = DEVICE UNIT UCB ADDRESS.
0798 1474      :
0798 1475      : OUTPUTS:
0798 1476      :
0798 1477      :     THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
0798 1478      :
0798 1479      :
0798 1480      : DB_REGDUMP:
0798 1481      :     MOVL  #<RP EC2+4+MBASL_BCR+4+8>/4,(R0)+ ;RP04/05/06 REGISTER DUMP ROUTINE
0798 1482      :     MOVL  MBASL_CSR(R4),(R0)+ ;INSERT NUMBER OF DEVICE REGISTERS
0798 1483      :     MOVL  MBASL_CR(R4),(R0)+ ;SAVE CONFIGURATION REGISTER
0798 1484      :     MOVL  UCB$B_DB_SR(R5),(R0)+ ;SAVE CONTROL REGISTER
0798 1485      :     MOVL  MBASL_VAR(R4),(R0)+ ;SAVE STATUS REGISTER
0798 1486      :     MOVL  MBASL_VIRTUAL_ADDR(R4),(R0)+ ;SAVE VIRTUAL ADDRESS REGISTER
0798 1487      :     MOVL  MBASL_BYTE_COUNT(R4),(R0)+ ;SAVE BYTE COUNT REGISTER
0798 1488      :     EXTZV #9,#8,-8(R0),R1 ;GET FINAL MAP REGISTER NUMBER
0798 1489      :     MOVL  MBASL_MAP(R4)[R1],(R0)+ ;SAVE FINAL MAP REGISTER CONTENTS
0798 1490      :     CLRL  (R0)+ ;ASSUME NO PREVIOUS MAP REGISTER
0798 1491      :     DECL  R1 ;CALCULATE PREVIOUS MAP REGISTER NUMBER
0798 1492      :     BLSS  10$ ;IF LSS NONE
0798 1493      :     MOVL  MBASL_MAP(R4)[R1],-4(R0) ;SAVE PREVIOUS MAP REGISTER CONTENTS
0798 1494      :     MOVZBL #<RP EC2+4>/4,R1 ;SET NUMBER OF DRIVE REGISTERS TO SAVE
0798 1495      :     MOVZBL UCB$B_SLAVE+1(R5),R2 ;GET DRIVE OFFSET CONSTANT
0798 1496      :     MOVAL MBASL_ERB(R4)[R2],R2 ;GET ADDRESS OF DRIVE REGISTERS
0798 1497      :     MOVL  (R2)+,(R0)+ ;SAVE DRIVE REGISTER
0798 1498      :     SOBGTR R1,20$ ;ANY MORE TO SAVE?
0798 1499      :     RSB ;

```

```

07DD 1500 .SBTTL RP04/RP05/RP06 DISK DRIVE INITIALIZATION
07DD 1501 :
07DD 1502 : DB_RPOX_INIT - RP04/RP05/RP06 DISK DRIVE INITIALIZATION
07DD 1503 :
07DD 1504 : THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
07DD 1505 : DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
07DD 1506 :
07DD 1507 : INPUTS:
07DD 1508 :
07DD 1509 : R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
07DD 1510 : R5 = DEVICE UNIT UCB ADDRESS.
07DD 1511 :
07DD 1512 : OUTPUTS:
07DD 1513 :
07DD 1514 : UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
07DD 1515 :
07DD 1516 :
07DD 1517 DB_RPCX_INIT: ;RP04/RP05/RP06 DISK DRIVE INITIALIZATION
07DD 1518 MOVZWL UCBSW_UNIT(R5),R3 ;GET DRIVE UNIT NUMBER
0090 C5 53 3C 07E1 1519 MOVB R3,UCBSB_SLAVE(R5) ;SET SLAVE UNIT NUMBER
07DD 1520 MULL #<107>/4,R3 ;CALCULATE DRIVE OFFSET CONSTANT
0091 C5 53 C4 07E6 1521 MOVB R3,UCBSB_SLAVE+1(R5) ;SET DRIVE OFFSET CONSTANT
53 0400 C443 DE 07EE 1522 MOVAL MBASL_ERB(R4)[R3],R3 ;GET ADDRESS OF DRIVE CONTROL REGISTER
03 BB 07F4 1523 PUSHR #^M<R0,R1> ;SAVE THESE REGISTERS
07F6 1524 TIMEWAIT #100,#RP_DS_M_DPR,- ;TRY TO SEIZE DRIVE
07F6 1525 RP_DS(R3),L
07DD 1526 BLBC R0,5$ ;NO PORT SEIZED
50 10 50 E9 0822 1527 MOVL RP_DT(R3),R0 ;GET DRIVE TYPE
08 50 18 A3 D0 0825 1528 BBC #RP_DT_V_DRQ,R0,5$ ;IF CLEAR, LEAVE
08 50 08 E1 0829 1529 BISB #ERC_M_DOALPORT,- ; SET FLAG WHICH INDICATES THAT DISK
00D2 C5 02 88 082D 1530 UCBSB_DB_ERL(R5) ; HAS DUAL PORT OPTION
63 09 9A 0832 1531 MOVZBL #F_DRVCLR!1,RP_CS1(R3) ;CLEAR DRIVE
7E 64 A5 3C 0835 1532 5$: POPR #^M<R0,R1> ;RESTORE REGISTERS
08 08 A4 DD 083B 1533 MOVZWL UCBSW_STS(R5),-(SP) ;SAVE CURRENT UNIT STATUS
64 A5 0810 8F AA 083E 1534 PUSHL MBASL_SR(R4) ;READ MBA STATUS REGISTER
32 6E 12 E0 0844 1535 BICW #UCBSM_ONLINE!UCBSM_VALID,UCBSW_STS(R5) ;SET UNIT OFFLINE/INVALID
64 A5 10 A8 0848 1536 BBS #MBASV_SR_NED,(SP),40$ ;IF SET, NONEXISTENT DISK
FF14 30 084C 1537 BBSW #UCBSM_ONLINE,UCBSW_STS(R5) ;SET UNIT ONLINE
23 64 A5 04 E1 084F 1538 BSBW DB_DTYPE ;CLASSIFY DRIVE TYPE
1E 04 AE 08 E1 0854 1539 BBC #UCBSV_ONLINE,UCBSW_STS(R5),30$ ;IF CLR, UNKNOWN DRIVE TYPE
52 04 A3 13 78 085C 1540 BBC #UCBSV_VALID,4(SP),30$ ;IF CLR, VOLUME SOFTWARE INVALID
00000000'GF 16 0863 1541 10$: MOVZBL #F_DRVCLR!1,RP_CS1(R3) ;CLEAR DRIVE
ED 50 09 19 0859 1542 ASHL #3T-RP_DS_V_MOC,RP_DS(R3),R2 ;MEDIUM ONLINE?
09 11 0861 1543 BLSS 20$ ;IF LSS YES
64 A5 0800 8F AB 0863 1544 JSB G^EXESPWRTIMCHX ;CHECK FOR MAXIMUM TIME EXCEEDED
08 A4 8E 8E C9 0869 1545 BLBS R0,10$ ;IF LBS MORE TIME TO GO
09 11 086C 1546 BRB 30$ ;
64 A5 0800 8F AB 0871 1547 20$: MOVZBL #F_PACKACK!1,RP_CS1(R3) ;ACKNOWLEDGE PACK
08 A4 8E 8E C9 0877 1548 BBSW #UCBSM_VALID,UCBSW_STS(R5) ;SET VOLUME SOFTWARE VALID
05 087A 1549 30$: MOVZBL #F_RELEASE!1,RP_CST(R3) ;Clear drive and release port
05 087F 1550 40$: BISL3 (SP)+,(SP)+,MBASL_SR(R4) ;CLEAR MBA STATUS
05 087F 1551 RSB ;

```



```

0880 1553      .SBTTL  RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
0880 1554      :
0880 1555      : DB_UN SOLNT - RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
0880 1556      :
0880 1557      : THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED FOR
0880 1558      : AN RP04, RP05, OR RP06 DRIVE.
0880 1559      :
0880 1560      : INPUTS:
0880 1561      :
0880 1562      :     R4 = ADDRESS OF CONFIGURATION STATUS REGISTER.
0880 1563      :     R5 = DEVICE UNIT UCB ADDRESS.
0880 1564      :
0880 1565      : OUTPUTS:
0880 1566      :
0880 1567      :     IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
0880 1568      :     UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
0880 1569      :     CLASSIFIED.
0880 1570      :
0880 1571      :
0880 1572      DB_UN SOLNT:
0880 1573      MOVZBL UCBSB_SLAVE+1(R5),R3      ;RP04/RP05/RP06 UNSOLICITED INTERRUPTS
0885 1574      MOVAL  MBASL_ERB(R4)[R3],R3      ;GET DRIVE OFFSET CONSTANT
0888 1575      BISW   #UCBSM_ONLINE,UCBSW_STS(R5) ;SET UNIT ONLINE
088F 1576      BSBW   DB_DTYPE      ;CLASSIFY DRIVE TYPE
0892 1577      BBC    #UCBSV_ONLINE,UCBSW_STS(R5),10$ ;IF CLR, UNKNOWN DRIVE TYPE
0897 1578      BBC    #UCBSV_VALID,UCBSW_STS(R5),20$ ;IF CLR, VOLUME SOFTWARE INVALID
089C 1579      ASHL  #31-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
08A1 1580      BGEQ  10$      ;IF GEQ NO
08A3 1581      BBC    #UCBSV_BSY,UCBSW_STS(R5),5$ ;We know the drive is online; thus,
08A8 1582      CMPB  #CDF_PACKACK,UCBSB_CEX(R5) ;if busy doing a PACKACK function,
08AD 1583      BEQL  20$      ;then don't clear software valid.
08AF 1584      ASHL  #31-RP_DS_V_VV,RP_DS(R3),R2 ;VOLUME VALID?
08B4 1585      BLSS  20$      ;IF LSS YES
08B6 1586      BICW  #UCBSM_VALID,UCBSW_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
08BC 1587      RSB
08BD 1588      DB_END:      ;ADDRESS OF LAST LOCATION IN DRIVER
08BD 1589
08BD 1590      .END
  
```

53	0091	C5	9A	0880	1573				
53	0400	C443	DE	0885	1574				
	64	A5	10	AB	0888				
		FED1	30	088F	1576				
	1F	64	A5	04	E1	0892			
	20	64	A5	08	E1	0897			
52	04	A3	13	78	089C				
			13	18	08A1				
	07	64	A5	08	E1	08A3			
		0093	C5	08	91	08A8			
			0D	13	08AD				
52	04	A3	19	78	08AF		5\$:		
			06	19	08B4				
64	A5	0800	8F	AA	08B6		10\$:		
				05	08BC		20\$:		
					08BD				
					08BD				
					08BD				

```

SSS = 00000020 R 02
SSOP = 00000002
ACPSACCESS ***** X 03
ACPSDEACCESS ***** X 03
ACPSMODIFY ***** X 03
ACPSMOUNT ***** X 03
ACPSREADBLK ***** X 03
ACPSWRITEBLK ***** X 03
APPLY ECC = 000002FE R 03
ATS_MBA = 00000000
CDF_DRVCLR = 00000004
CDF_NOP = 00000005
CDF_OFFSET = 00000006
CDF_PACKACK = 00000008
CDF_READDATA = 0000000C
CDF_READHEAD = 0000000E
CDF_READPRESET = 00000010
CDF_RECAL = 00000003
CDF_RETCENTER = 00000007
CDF_SEARCH = 00000009
CDF_SEARCHA = 00000011
CDF_SEEK = 00000002
CDF_UNLOAD = 00000001
CDF_WRITECHECK = 0000000A
CDF_WRITECHECKH = 0000000F
CDF_WRITEDATA = 0000000B
CDF_WRITEHEAD = 0000000D
CHECKRETRY = 0000026D R 03
CHECKTAB = 00000038 R 03
CHECKXT = 00000286 R 03
CRBSL_INTD = 00000024
DATACHECK = 0000023A R 03
DBSDDT = 00000000 RG 03
DB_DTDESC = 0000003C R 03
DB_DTDESCLEN = 0000000F
DB_DTYPE = 00000763 R 03
DB_END = 000008BD R 03
DB_FUNCABLE = 000000A3 R 03
DB_REGDUMP = 00000798 R 03
DB_RPOX_INIT = 000007DD R 03
DB_STARTIO = 00000137 R 03
DB_UNSOINT = 00000880 R 03
DCS_DISK = 00000001
DDBSK_PACK = 00000001
DDBSL_ACPD = 00000010
DDBSL_DDT = 0000000C
DEFER_ECC = 00000327 R 03
DEVSM_AVL = 00040000
DEVSM_DIR = 00000008
DEVSM_DUA = 00008000
DEVSM_ELG = 00400000
DEVSM_FOD = 00004000
DEVSM_IDV = 04000000
DEVSM_NNM = 00000200
DEVSM_ODV = 08000000
DEVSM_RND = 10000000
DEVSM_SHR = 00010000

```

```

DPTSC_LENGTH = 00000038
DPTSC_VERSION = 00000004
DPTSINITAB = 00000038 R 02
DPTSM_SVP = 00000002
DPTSREINITAB = 0000006A R 02
DPTSTAB = 00000000 R 02
DRVCLR = 00000209 R 03
DTS_RP04 = 00000003
DTS_RP05 = 00000004
DTS_RP06 = 00000005
DYNBC_DDB = 00000006
DYNBC_DPT = 0000001E
DYNBC_UCB = 00000010
ECC = 000002AE R 03
EMBSL_DV_REGSAV = 0000004E
ENBXIT = 000005D4 R 03
ERL$DEVICERR ***** X 03
ERL$DEVICTMO ***** X 03
ERL_M_DUALPORT = 00000002
ERL_M_ECC_DEFER = 00000004
ERL_M_MEDOFF = 00000001
ERL_V_DUALPORT = 00000001
ERL_V_ECC_DEFER = 00000002
ERROR = 00000603 R 03
EXESGL_TENUSEC ***** X 03
EXESGL_UBDELAY ***** X 03
EXESIOFORK ***** X 03
EXESLCLDSKVALID ***** X 03
EXESONEPARM ***** X 03
EXESPWRTIMCHK ***** X 03
EXESSENSEMODE ***** X 03
EXESSETCHAR ***** X 03
EXESZEROPARM ***** X 03
EXFNC = 00000581 R 03
FATALERR = 000003EB R 03
FDISPATCH = 000001AC R 03
FEX = 000004C3 R 03
FTAB = 00000089 R 03
FUNCTAB_LEN = 00000094
FUNCXT = 0000047F R 03
F_DRVCLR = 00000008
F_NOP = 00000000
F_OFFSET = 0000000C
F_PACKACK = 00000012
F_READDATA = 00000038
F_READHEAD = 0000003A
F_READPRESET = 00000010
F_RECAL = 00000006
F_RELEASE = 0000000A
F_RETCENTER = 0000000E
F_SEARCH = 00000018
F_SEARCHA = 00000018
F_SEEK = 00000004
F_UNLOAD = 00000002
F_WRITECHECK = 00000028
F_WRITECHECKH = 0000002A
F_WRITEDATA = 00000030

```

DBDRIVER  
Symbol table

- RP04/05/06 DISK DRIVER

1 9

15-SEP-1984 23:45:36 VAX/VMS Macro V04-00  
5-SEP-1984 00:11:41 [DRIVER.SRC]DBDRIVER.MAR;1

Page 32  
(1)

```

F_WRITEHEAD = 00000032
GO = 000004E2 R 03
IDBSL_OWNER = 00000004
IMMED = 0000050B R 03
IOSM_DATACHECK = 00004000
IOSV_DATACHECK = 0000000E
IOSV_INHRETRY = 0000000F
IOSV_INHSEEK = 0000000C
IOS_ACCESS = 00000032
IOS_ACPCONTROL = 00000038
IOS_AVAILABLE = 00000011
IOS_CREATE = 00000033
IOS_DEACCESS = 00000034
IOS_DELETE = 00000035
IOS_DRVCLR = 00000004
IOS_MODIFY = 00000036
IOS_MOUNT = 00000039
IOS_NOP = 00000000
IOS_OFFSET = 00000006
IOS_PACKACK = 00000008
IOS_READHEAD = 0000000E
IOS_READBLK = 00000021
IOS_READPBLK = 0000000C
IOS_READPRESET = 00000019
IOS_READVBLK = 00000031
IOS_RECAL = 00000003
IOS_RELEASE = 00000005
IOS_RETCENTER = 00000007
IOS_SEARCH = 00000009
IOS_SEEK = 00000002
IOS_SENSECHAR = 0000001B
IOS_SENSEMODE = 00000027
IOS_SETCHAR = 0000001A
IOS_SETMODE = 00000023
IOS_UNLOAD = 00000001
IOS_VIRTUAL = 0000003F
IOS_WRITECHECK = 0000000A
IOS_WRITECHECKM = 00000018
IOS_WRITEHEAD = 0000000D
IOS_WRITEBLK = 00000020
IOS_WRITEPBLK = 0000000B
IOS_WRITEVBLK = 00000030
IOCSAPPLYECC ***** X 03
IOCSDIAGBUFILL ***** X 03
IOCSLOADMBAMAP ***** X 03
IOCSMNTVER ***** X 03
IOCSRELCHAN ***** X 03
IOCSREQCOM ***** X 03
IOCSREQPCANL ***** X 03
IOCSRETURN ***** X 03
IOCSUPDATRASP ***** X 03
IOCSWFKPCH ***** X 03
IRPSL_MEDIA = 00000038
IRPSL_SVAPE = 0000002C
IRPSL_FCODE = 00000006
IRPSV_FCODE = 00000000
IRPSV_FUNC = 00000001

```

```

IRPSV_PHYSIO = 00000008
IRPSW_BCNT = 00000032
IRPSW_FUNC = 00000020
IRPSW_STS = 0000002A
LDCYL = 0000057B R 03
MASKH = 00000008
MASKL = C4000000
MBASL_BCR = 00000010
MBASL_CR = 00000004
MBASL_CSR = 00000000
MBASL_ERB = 00000400
MBASL_MAP = 00000800
MBASL_SR = 00000008
MBASL_VAR = 0000000C
MBASH_CR_ABORT = 00000002
MBASH_CR_IE = 00000004
MBASH_CR_INIT = 00000001
MBASH_ERROR = 000E5FFF
MBASH_SR_DLT = 00000800
MBASH_SR_FRCONF = 00000008
MBASH_SR_INVMAP = 00000010
MBASH_SR_ISTO = 00000002
MBASH_SR_MAPPE = 00000020
MBASH_SR_MBEXC = 00000080
MBASH_SR_MCPE = 00020000
MBASH_SR_MDPE = 00000040
MBASH_SR_MXF = 00000100
MBASH_SR_NED = 00040000
MBASH_SR_PGE = 00080000
MBASH_SR_RDS = 00000004
MBASH_SR_RDTO = 00000001
MBASH_SR_SPE = 00004000
MBASH_SR_WCKLWR = 00000200
MBASH_SR_WCKUPR = 00000400
MBASV_SR_NED = 00000012
NOP = 00000209 R 03
NORMAL = 00000283 R 03
OFF = 0000032C R 03
OFFSET = 00000209 R 03
OFFSETErr = 00000383 R 03
OFFSIZ = 00000008
OFFTAB = 0000009B R 03
PACKACK = 00000203 R 03
POSIT = 00000575 R 03
PR$ IPL = 00000012
READDATA = 0000021C R 03
READHEAD = 0000021C R 03
READPRESET = 00000209 R 03
RECAL = 00000209 R 03
RELEASE = 00000209 R 03
RESETXFR = 00000744 R 03
RETCENTER = 00000209 R 03
RETREG = 000005D7 R 03
RETRY = 000002AB R 03
RETRYERR = 000003BE R 03
RP_AS = 00000010
RP_CC = 0000002C

```

RP\_CS1 = 00000000  
 RP\_CS1\_M\_GO = 00000001  
 RP\_DA = 00000014  
 RP\_DC = 00000028  
 RP\_DS = 00000004  
 RP\_DS\_M\_DPR = 00000100  
 RP\_DS\_M\_ERR = 00004000  
 RP\_DS\_V\_ERR = 0000000E  
 RP\_DS\_V\_MOL = 0000000C  
 RP\_DS\_V\_VV = 00000006  
 RP\_DT = 00000018  
 RP\_DT\_V\_DRQ = 0000000B  
 RP\_ECT = 00000038  
 RP\_EC2 = 0000003C  
 RP\_ER1 = 00000008  
 RP\_ER1\_M\_AOE = 00000200  
 RP\_ER1\_M\_DCK = 00008000  
 RP\_ER1\_M\_DTE = 00001000  
 RP\_ER1\_M\_ECH = 00000040  
 RP\_ER1\_M\_FER = 00000010  
 RP\_ER1\_M\_HCE = 00000080  
 RP\_ER1\_M\_HCRC = 00000100  
 RP\_ER1\_M\_IAE = 00000400  
 RP\_ER1\_M\_ILF = 00000001  
 RP\_ER1\_M\_ILR = 00000002  
 RP\_ER1\_M\_OPI = 00002000  
 RP\_ER1\_M\_PAR = 00000008  
 RP\_ER1\_M\_RMR = 00000004  
 RP\_ER1\_M\_UNE = 00004000  
 RP\_ER1\_M\_WCF = 00000020  
 RP\_ER1\_M\_WLE = 00000800  
 RP\_ER1\_V\_FER = 00000004  
 RP\_ER1\_V\_HCE = 00000007  
 RP\_ER1\_V\_HCRC = 00000008  
 RP\_ER1\_V\_OPI = 0000000D  
 RP\_ER1\_V\_UNE = 0000000E  
 RP\_ER1\_V\_WLE = 0000000B  
 RP\_ER2 = 00000020  
 RP\_ER3 = 00000034  
 RP\_ER3\_V\_SKI = 0000000E  
 RP\_LA = 0000001C  
 RP\_MR = 0000000C  
 RP\_OF = 00000024  
 RP\_OF\_M\_DCK = 00000100  
 RP\_OF\_M\_ECI = 00000800  
 RP\_OF\_M\_FMT = 00001000  
 RP\_OF\_M\_HCI = 00000400  
 RP\_OF\_V\_DCK = 00000008  
 RP\_OF\_V\_ECI = 0000000B  
 RP\_SN = 00000030  
 SEARCH = 00000209  
 SEARCHA = 00000552  
 SEEK = 00000209  
 SEIZE = 00000522  
 SIZ... = 00000001  
 SSS\_CTRLERR = 00000054  
 SSS\_DATACHECK = 0000005C

R 03  
 R 03  
 R 03  
 R 03

SSS\_DRVERR = 0000008C  
 SSS\_FORMAT = 0000008C  
 SSS\_IVADDR = 00000134  
 SSS\_MEDOFL = 000001A4  
 SSS\_NONEXDRV = 000001C4  
 SSS\_NORMAL = 00000001  
 SSS\_OPINCOMPL = 000002D4  
 SSS\_PARITY = 000001F4  
 SSS\_TIMEOUT = 0000022C  
 SSS\_UNSAFE = 0000023C  
 SSS\_VOLINV = 00000254  
 SSS\_WASECC = 00000639  
 SSS\_WRITLCK = 0000025C  
 TRANNOCH = 00000230  
 TRANRQCH = 0000022A  
 TRANXT = 00000289  
 UCBSB\_CEX = 00000093  
 UCBSB\_DB\_ERL = 000000D2  
 UCBSB\_DEVCLASS = 00000040  
 UCBSB\_DEVTYPE = 00000041  
 UCBSB\_DIPL = 0000005E  
 UCBSB\_ERTCNT = 00000080  
 UCBSB\_ERTMAX = 00000081  
 UCBSB\_FEX = 00000092  
 UCBSB\_FIPL = 0000000B  
 UCBSB\_OFFNDX = 000000CA  
 UCBSB\_OFFRTC = 000000CB  
 UCBSB\_SECTORS = 00000044  
 UCBSB\_SLAVE = 00000090  
 UCBSK\_DB\_LENGTH = 000000D6  
 UCBSK\_LCC\_DISK\_LENGTH = 000000CC  
 UCBSL\_BCR = 000000C0  
 UCBSL\_CRB = 00000024  
 UCBSL\_DB\_BCR = 000000C0  
 UCBSL\_DB\_SR = 000000CE  
 UCBSL\_DEVCHAR = 00000038  
 UCBSL\_DEVCHAR2 = 0000003C  
 UCBSL\_DEVDEPEND = 00000044  
 UCBSL\_DPC = 0000009C  
 UCBSL\_IRP = 00000058  
 UCBSL\_MAXBLOCK = 000000B0  
 UCBSL\_MEDIA\_ID = 0000008C  
 UCBSL\_SVAPTE = 00000078  
 UCBSM\_ONLINE = 00000010  
 UCBSM\_POWER = 00000020  
 UCBSM\_TIMEOUT = 00000040  
 UCBSM\_VALID = 00000080  
 UCBSV\_BSY = 00000008  
 UCBSV\_ECC = 00000000  
 UCBSV\_ONLINE = 00000004  
 UCBSV\_POWER = 00000005  
 UCBSV\_VALID = 0000000B  
 UCBSW\_BCNT = 0000007E  
 UCBSW\_BCR = 000000C0  
 UCBSW\_DA = 0000008C  
 UCBSW\_DB\_ER3 = 000000CC  
 UCBSW\_DC = 0000008E

R 03  
 R 03  
 R 03

DBDRIVER  
Symbol table

- RP04/05/06 DISK DRIVER

K 9

15-SEP-1984 23:45:36 VAX/VMS Macro V04-00  
5-SEP-1984 00:11:41 [DRIVER.SRC]DBDRIVER.MAR;1

Page 34  
(1)

```
UCBSW_DEVBUSIZ      = 00000042
UCBSW_DEVSTS        = 00000068
UCBSW_EC1           = 000000C4
UCBSW_EC2           = 000000C6
UCBSW_FUNC          = 0000009A
UCBSW_OFFSET        = 000000C8
UCBSW_STS           = 00000064
UCBSW_UNIT          = 00000054
UNLOAD              = 000001FB R    03
VECSL_IDB           = 00000008
WRITECHECK          = 00000210 R    03
WRITECHECKM         = 00000210 R    03
WRITEDATA           = 00000217 R    03
WRITEHEAD           = 00000217 R    03
XFER                = 00000566 R    03
```

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	000000D6 ( 214.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$105_PROLOGUE	00000070 ( 112.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$115_DRIVER	000008BD ( 2237.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

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

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

The working set limit was 2100 pages.  
151533 bytes (296 pages) of virtual memory were used to buffer the intermediate code.  
There were 120 pages of symbol table space allocated to hold 2322 non-local and 65 local symbols.  
1590 source lines were read in Pass 1, producing 22 object records in Pass 2.  
48 pages of virtual memory were used to define 45 macros.

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

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

2486 GETS were required to define 40 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DBDRIVER/OBJ=OBJ\$:DBDRIVER MSRC\$:DBDRIVER/UPDATE=(ENH\$:DBDRIVER)+EXECMLS/LIB



