IOJUPITER.ASM

```
0000          1: Listdefs  equ  0            ; don't list SDOS definitions
              2: ;
003C          3: memsize   equ   60          ; K of memory
8400          4: code      equ   $8400       ; top of user space
              5: ;
0001          6: VirtualFloppy equ 1         ; Use Virtual (joke) floppy drive
              7: ;
0002          8: PerSci    equ   2           ; 1 dual persci
0001          9: WMformat  equ   1           ;   set up for WM 256 byte format...
0001         10: IBMformat equ   1           ;   and IBM 3740 format
             11: ;
0002         12: DAMfloppy equ   2           ; 1 dual DAM floppy drive
             13: ;
0001         14: StorageDemon equ 1          ; Standard Storage Demon
0001         15: IMI7710S  equ   1           ;    IMI 7710s
0001         16: UseDemonAsClock equ 1       ; Use Demon interface as clock
             17: ;
0800         18: DesiredPoolSize equ 512*4   ; so poolsize is 3 sectors -> readahead happens
             19: ;
0000         20: InBufSize:$ffc4 equ  0      ; line printer doesn't need input buffer
0000         21: LineBufSize:$ffc4 equ 0     ; line printer doesn't need line buffer
```

CPUSpeed equ $87
; 1100 MHz

for 7711
9382 / 4A38

make D2: the default disk
8EB1 / 9155

9286+5 / 9379

90E4+5 / 0000

fix BUILDMAP1
8508 / DE068DEC

fix FORECORE
9E05 / 01
86A5
B7FF82
40
BBFF82
40 8600
46 8880
FF FF82

9DF8 / 7E9E05

make retry count high
8584 / 30

87D1 / E7
fix SAVEERRLSN

fix WDCRESET
8A39 / 86CA
B7FF4C
B6FF41
8612
B7FF4D
B7FF4E
CE1388
09
26FD
86EA

fix error recovery in floppy driver 4/2/83
87C4 / 36 EE2B
A603E604
FE9013
6F3F
7E8EE63F

8EE6 / A740E741
3239

3/3/85 patch
clock reset to make
CLOCK! more accurate
9E35 / C628
8639

3/22/85 patch LPT
for 10 second timeout
on EPSON
8D78 / 0266

8D9E / 0266

```
                2: ********************************************************************************
  0001          3: JUPITERII EQU    1                   THIS I/O PACKAGE IS FOR A JUPITER II!
  0001          4: WAVEMATE EQU   1                     IF'S ON THIS SHOULD BE IF'S ON JUPITERII
                5: ********************************************************************************
                6:
                9:          NAME  SDOSDRIVERS
  0001         10:          IFUND   M6800
  0001         11: M6800    EQU     1
  0000         12: M6801    EQU     0
  0000         13: M6809    EQU     0
               14:          FIN     M6800
               15:
  0001         16:          IF    M6800!M6801
               17:          WITH  LINCLUDE
               18:          FIN
               19:
               20: *
               21: *        BY SOFTWARE DYNAMICS
               22: *        AND A CAST OF THOUSANDS!
               23: *
  1231         24: EDITDATE EQU   $1231                 /82 MMDD IN HEX FORMAT
  1982         25: EDITYEAR EQU   $1982
               26: *
               27: *
               28: ********************************************************************************
               29: *   I/O PACKAGE STRUCTURE
               30: *   The I/O package is organized in the following fashion:
               31: *
               32: *   Low addresses: !                 !
               33: *                  !  Read-only code,  !
               34: *                  !   tables, etc.    !
               35: *                  !------------------!
               36: *                  !                  !
               37: *                  !  Interrupt poll   !
               38: *                  !     chains        !
               39: *                  !   (readonly)     !
               40: *                  !                  !
               41: *                  !------------------!
               42: *                  !                  !
               43: *                  !  Working storage, !
               44: *                  !  DCBs, TCBs, etc. !
               45: *                  !                  !
               46: *                  !------------------!
               47: *                  !                  !
               48: *                  !    Disk Buffer   !
               49: *                  !       Pool       !
               50: *                  !    I/O Driver    !
               51: *                  !    Reset code    !
               52: *                  !    (once-only)   !
               53: *                  !                  !
               54: *                  !------------------!
               55: *                  !                  !
               56: *                  !   VT Drivers     !
               57: *                  !      SDOS        !
               58: *                  !      ....        !
```

```
                59: *
                60: *       To make this arrangement possible, each I/O package source is
                61: *       organized in the following way:
                62: *            IOxxxxx.DO             is a file containing a configuration
                63: *                                    for machine xxxxx
                64: *            IOxxxxx.DO      is a file containing an I/O package "shell"
                65: *            IOyyyyy.ASM            is a file containing ALL driver-related code
                66: *                                    tables, etc. for the hardware device yyyyyy
                67: *
                68: *       Each I/O package shell uses conditional assembly switches to conditionally
                69: *       INCLUDE IOyyyyy.ASM in a particular configuration.  The IOyyyyy.ASM
                70: *       file is actually INCLUDEd 4 times, once for each of the 4 areas of
                71: *       the I/O package shown above.  The following conditional switches
                72: *       are used by the driver source module to distinguish between areas:
                73: *            IODRIVERBODY           selects the read-only code portion
                74: *            IODRIVERPOLL           selects the Interrupt poll chain portion
                75: *            IODRIVERRAM            selects read/write storage of driver
                76: *            IODRIVERINIT           selects the once-only I/O driver initializing code
                77: *
                78: *       Note: the driver source module should define all equates and (DCB)
                79: *       table displacements when the conditional switch IODRIVERBODY is enabled.
                80:
                81: *
                82: *       A Typical shell has the following source form:
                83: *
                84: *       *    SET DEFAULTS
                85: *            IFUND                  xxxx
                86: *            IFUND                  yyyy
                87: *            ...
                88: *       *    BUILD READ-ONLY CODE
                89: *       IODRIVERBODY                SET 1
                90: *       IODRIVERPOLL                SET 0
                91: *       IODRIVERRAM                 SET 0
                92: *       IODRIVERINIT                SET 0
                93: *            IF                     xxxx
                94: *            INCLUDE                IOxxxx.ASM
                95: *            FIN
                96: *            IF                     yyyy
                97: *            INCLUDE                IOyyyy.ASM
                98: *            FIN
                99: *            ...
               100: *       PATCHSPACE                  RTP zzzz
               101: *            SWI
               102: *       ***** Build Interrupt Poll Chains
               103: *       IODRIVERBODY                SET 0
               104: *       IODRIVERPOLL                SET 1
               105: *            IF                     xxxx Note: order of poll routines may be different than bodies
               106: *            INCLUDE                IOxxxx.ASM
               107: *            FIN
               108: *            IF                     yyyy
               109: *            INCLUDE                IOyyyy.ASM
               110: *            FIN
               111: *            ...
               112: *       ***** Build Working Storage
               113: *       IODRIVERPOLL                SET 0
```

```
114: *       IODRIVERRAM                SET 1
115: *            IF                    xxxx
116: *            INCLUDE               IOxxxx.ASM
117: *            FIN
118: *            IF                    xxxx
119: *            IF                    yyyy
120: *            INCLUDE               IOyyyy.ASM
121: *            FIN
122: *            ...
123: *       ***** Build Driver Reset routines
124: *       IODRIVERRAM                SET 0
125: *       IODRIVERINIT               SET 1
126: *            IF                    xxxx
127: *            INCLUDE               IOxxxx.ASM
128: *            FIN
129: *            IF                    yyyy
130: *            INCLUDE               IOyyyy.ASM
131: *            FIN
132: *            ...
133: *       ***** Finish out disk buffer pool, etc
134: *            ...
135: *            END
```

```
                137: *
  FC03          138: INICV    EQU   $FC03               INITIALIZE CONSOLE DEVICE VECTOR
  FC06          139: PUTCV    EQU   $FC06               PUT A CHARACTER TO CONSOLE VECTOR
  FC09          140: GETCV    EQU   $FC09               GET A CHARACTER TO CONSOLE VECTOR
  FC0C          141: TESTCV   EQU   $FC0C               TEST FOR ARRIVAL OF CHARACTER FROM CONSOLE
  FC12          142: INIDV    EQU   $FC12               INIT DEFAULT DEVICE (FOR LPT)
  FC15          143: PUTDV    EQU   $FC15               PUT A CHARACTER TO DEFAULT DEVICE VECTOR
                144:
  00FE          145: SYSPG    EQU   $FE
  00FD          146: SYSIIRQ  EQU   $FD
                147:
                148:
```

```
  0001          150: OUTASPACE      EQU    1  ; USE TRIMMED DEF FILES TO SAVE SYMBOL TABLE SPACE
                151:
  0001          152:          IFUND SDOSMT
  0000          153: SDOSMT   EQU   0                     ; DEFAULT IS "NOT SDOS/MT"
                154:          FIN
                155:
  0000          156:          IF    SDOSMT
                169:          FIN   SDOSMT
  0000          170:          IFUND MEMSIZE
                172:          FIN   MEMSIZE
                173:
  0000          174:          IFUND DESIREDPOOLSIZE
                176:          FIN
                177:
  0001          178:          IFUND NIOCHANNELS
  0000          179:          IF    SDOSMT
                181:          ELSE
  0008          182: NIOCHANNELS    EQU                   8
                183:          FIN
                184:          FIN   NIOCHANNELS
                185:
                186: *
  0400          187: K        EQU   1024                  # BYTES PER "K" OF MEMORY
                188: *
  0001          189:          IF    M6800!M6801
  0000          190:          IFUND CODE
                196:          FIN   CODE
                197: *
  0001          198:          IFUND SDOS
  BE00          199: SDOS     EQU   MEMSIZE*K-$3200
                200:          FIN   SDOS
  0001          201:          IFUND VTDRIVER
  A600          202: VTDRIVER EQU   SDOS-$1800
                203:          FIN
  0001          204:          ELSE  (M6809)
                222:          FIN
                223: *
  0001          224:          IFUND DRIVERBASE
  0001          225:          IF    CODE<<SDOS
  8400          226: DRIVERBASE     EQU                   CODE
  0002          227:          ELSE
                229:          FIN   CODE<<SDOS
                230:          FIN   DRIVERBASE
                231:
  0001          232:          IFUND REALTIMECLOCK         "THERE EXISTS A REAL PIECE OF CLOCK HARDWARE"
  0000          233:          IFUND STORAGEDEMON
                237:          ELSE
  0001          238:          IFUND USECONSOLEACIAASCLOCK
  0000          239: USECONSOLEACIAASCLOCK                EQU &STORAGEDEMON
                240:          FIN   USECONSOLEACIAASCLOCK
                241:          FIN   STORAGEDEMON
  0001          242: REALTIMECLOCK  EQU                   &USECONSOLEACIAASCLOCK
                243:          FIN   REALTIMECLOCK
                244:
  003C          245: TICKSPERSECOND EQU                   60
```

```
 0001          246:          IFUND CLOCK
 0001          247: CLOCK    EQU   1
               248:          FIN
 0001          249:          IF    CLOCK
 0001          250:          IFUND LISTCLOCK
 0001          251: LISTCLOCK      EQU                     1
               252:          FIN
               253:          FIN
               254:
 0001          255:          IFUND BLACKHOLE
 0000          256: BLACKHOLE      EQU                     0
               257:          FIN
 0000          258:          IF    BLACKHOLE
               262:          FIN
               263:
 0001          264:          IFUND SDLP
 0000          265: SDLP     EQU   0
               266:          FIN
 0000          267:          IF    SDLP
               271:          FIN
               272:
 0000          273:          IFUND VIRTUALFLOPPY
               275:          FIN   VIRTUALFLOPPY
 0001          276:          IF    VIRTUALFLOPPY
 0001          277:          IFUND LIST.VIRTUALFLOPPY
 0001          278: LIST.VIRTUALFLOPPY EQU  1
               279:          FIN   LISTVIRTUALFLOPPY
 0000          280:          IFUND PERSCI
               282:          FIN   PERSCI
 0000          283:          IFUND DAMFLOPPY
               285:          FIN   DAMFLOPPY
 0002          286:          IF    PERSCI
 0000          287:          IFUND WMFORMAT
               289:          FIN   WMFORMAT
 0000          290:          IFUND IBMFORMAT
               292:          FIN   IBMFORMAT
               293:          FIN   PERSCI
               294:          FIN   VIRTUALFLOPPY
 0001          295:          IFUND WMPERSCI
 0000          296: WMPERSCI EQU   0
               297:          FIN
 0000          298:          IF    WMPERSCI
               305:          FIN   WMPERSCI
 0001          306:          IFUND WMDAMFLOPPY
 0000          307: WMDAMFLOPPY    EQU                     0
               308:          FIN   WMDAMFLOPPY
 0000          309:          IF    WMDAMFLOPPY
               320:          FIN   WMDAMFLOPPY
               321:
 0000          322:          IFUND STORAGEDEMON
               324:          FIN
 0001          325:          IF    STORAGEDEMON
 0001          326:          IFUND LISTSTORAGEDEMON
 0001          327: LISTSTORAGEDEMON                     EQU 1
               328:          FIN
```

```
                329:          FIN
                330:
  0000          331:          IFUND LISTDEFS
                333:          FIN
  0001          334: IOPKDEFS EQU   1                     MAKE SURE WE GET I/O PACKAGE DEFINITIONS
```

```
                344: *
                345: *        SDOS-TO-IOPACKAGE COMMUNICATION REGION
                346: *
  BE01          347:          ORG   SDOS+1
                348: ****     FCB   $10                 SDOS VERSION NUMBER
BE01 0000       349:          FDB   0                   LAST ERROR ENCOUNTERED
BE03 BEB1       350:          FDB   CNFGTABLE           TELL SDOS WHERE ALL THE GOODIES ARE
BE05 0000       351:          FDB   0                   SERIAL NUMBER
BE07 0000       352:          FDB   0                   IOBLOCKPTR
BE09 0000       353:          FDB   0                   IOCB POINTER (FOR FILE-TYPE DEVICE DRIVERS)
BE0B 000000     354:          FCB   0,0,0               SET CLOCK TO "MIDNITE"
BE0E 00         355:          FCB   0                   DAY, LET COMMAND INTERPRETER KNOW
BE0F 00         356:          FCB   0                   MONTH, THAT THE TIME HASN'T BEEN SET
BE10 82         357:          FCB   EDITYEAR&$FF        YEAR
                358: *
  8400          359:          ORG   DRIVERBASE
8400 7E8400     360: SYSCALLIO      JMP                 SYSCALLIO SDOS SETS JMP ADDR TO ITS ENTRY POINT
                361:
8403 1231       362:         FDB    EDITDATE            RECORD I/O PACKAGE DATE IN OBJECT FILE
8405 1982       363:         FDB    EDITYEAR
                364:
                365: *
                366: *****    READ ONLY CODE SECTION
  0001          367: IODRIVERBODY   SET                 1
  0000          368: IODRIVERPOLL   SET                 0
  0000          369: IODRIVERRAM    SET                 0
  0000          370: IODRIVERINIT   SET                 0
                371:
                372: *
  0000          373: NEXTTCB  SET   0                   END OF TCB CHAIN
  0000          374: NEXTTIMEOUT    SET                0
  0000          375: NEXTDISKDCB    SET                0
  0000          376: NEXTDEVICEDCB  SET                0
  0000          377: NTIMEOUTS      SET                0
  0000          378: NDISKDCBS      SET                0
  BE15          379: INTERRUPTTARGET                   SET SDOS+SDOS:RTI ASSUME CONVENTIONAL INTERRUPT SCHEME
```

```
  0001          381:          IF    M6800
  8407          382: DEBUGSYSCALLHANDLER
8407 34         383:          DES                           MAKE SPACE FOR X
8408 34         384:          DES                           MAKE ROOM FOR INDEX REGISTER
8409 34         385:          DES                           MAKE ROOM FOR CONTENTS OF (A)
840A 34         386:          DES                           SAVE ACCB
840B 07         387:          TPA
840C 36         388:          PSHA                          SAVE CC BITS
840D BD8EC4     389:          JSR   INTDISABLE              TURN OFF INTERRUPTS
  8410          390: DEBUGINTERRUPT ; ^D: CONTEXT BLOCK IS ON TOP OF STACK
8410 FEFFFC     391:          LDX   $FFFC                    ; NMI VECTOR
8413 6E00       392:          JMP   0,X
  0001          393:          ELSE  (M6809)
                399:          FIN   M6800
  0001          400:          IF    CLOCK
                401:          INCLUDE                       IOCLOCK.ASM
  0001            1:                  IF      IODRIVERBODY
```

IOCLOCK.ASM

```
8415 842F          3: CLOCKDRIVER       FDB     CLOCKOPEN
8417 842F          4:                   FDB     CLOCKCLOSE
8419 848D          5:                   FDB     CLOCKREADA
841B 8EDB          6:                   FDB     ILLDEVICEOP     WRITEA IS A NO-NO
841D 8469          7:                   FDB     CLOCKREADB
841F 8448          8:                   FDB     CLOCKWRITEB
8421 8EDB          9:                   FDB     ILLDEVICEOP     YOU UPDATE THE CLOCK, NOT REBUILD IT (CREATE)
8423 8EDB         10:                   FDB     ILLDEVICEOP     RENAME IT TO WHAT? CLOCK-RADIO; ???
8425 8EDB         11:                   FDB     ILLDEVICEOP     YOU CAN'T GET RID OF THE CLOCK, NEITHER
8427 8EDB         12:                   FDB     ILLDEVICEOP     NO CONTROL FUNCTIONS
8429 8434         13:                   FDB     CLOCKSTATUS     SAY "I'M A CLOCK, TICK-TOCK"
842B 9E35         14:                   FDB     CLOCKRESET
842D 842F         15:                   FDB     CLOCKPFRESTART  WHO UNPLUGGED ME???
                  16: *
  842F            17: CLOCKCLOSE        EQU     *               WHAT AM I SUPPOSED TO DO, PUT THE CLOCK AWAY??
  842F            18: CLOCKOPEN         EQU     *               HOW ABOUT LOOKING AT YOUR $9 TI CHEAPIE, MAC!
  842F            19: CLOCKPFRESTART    EQU     *               AM I SUPPOSED TO KEEP TIME WITH NO POWER???
842F 0C39         20:                   OKRTS                   TOUGH!
                  21:
                  22: *
8431 7E8EDB       23: CLOCKSPRUNG       JMP     ILLDEVICEOP
                  24:
8434 8104         25: CLOCKSTATUS       CMPA    #SC:GETTYPE
8436 26F9         26:                   BNE     CLOCKSPRUNG
8438 FEBE07       27:                   LDX     SDOS+SDOS:IOBLOCKPTR
843B BDBE36       28:                   JSR     SDOS+SDOS:CHECKRDLEN HAS HE GOT A BYTE SPACE
843E 0001         29:                   FDB     1
8440 EE0A         30:                   LDX     SCBLK:RDBUF,X   GET THE BUFFER POINTER
8442 860B         31:                   LDAA    #DVTYP.CLOCK    I'M ALIVE AND TICKING (HOPEFULLY!)
8444 A700         32:                   STAA    DVTYP:TYPE,X
8446 0C39         33:                   OKRTS
                  34:
8448 BDBE39       35: CLOCKWRITEB       JSR     SDOS+SDOS:CHECKWRLEN HAS HE GOT 6 BYTES?
844B 0006         36:                   FDB     6
844D EE04         37:                   LDX     SCBLK:WRBUF,X   WRITE BUFFER POINTER
844F C606         38:                   LDAB    #6
8451 A600         39: CLOCKWB1          LDAA    0,X
8453 08           40:                   INX
8454 36           41:                   PSHA
8455 5A           42:                   DECB
8456 26F9         43:                   BNE     CLOCKWB1
8458 01           44:                   NOP                     DON'T WANT TO UPDATE THE CLOCK WHILE SETTING IT
8459 0F           45:                   SEI
845A CEBE00       46:                   LDX     #SDOS
845D C606         47:                   LDAB    #6
845F 32           48: CLOCKWB2          PULA
8460 A710         49:                   STAA    SDOS:CLOCK+5,X
8462 09           50:                   DEX
8463 5A           51:                   DECB
8464 26F9         52:                   BNE     CLOCKWB2
8466 0E           53:                   CLI
8467 0C39         54:                   OKRTS                   AND WE'S DONE!
                  55: *
                  56: *
                  57: *
```

```
8469 BDBE36      58: CLOCKREADB      JSR     SDOS+SDOS:CHECKRDLEN
846C 0006        59:                 FDB     6               HE BETTER HAVE 6 BYTES AT LEAST
846E 01          60:                 NOP                      DON'T WANT CLOCK UPDATED WHILE READING IT
846F 0F          61:                 SEI
8470 CEBE00      62:                 LDX     #SDOS
8473 C606        63:                 LDAB    #6
8475 A60B        64: CLOCKRB1        LDAA    SDOS:CLOCK,X
8477 08          65:                 INX
8478 36          66:                 PSHA
8479 5A          67:                 DECB
847A 26F9        68:                 BNE     CLOCKRB1
847C 0E          69:                 CLI
847D FEBE07      70:                 LDX     SDOS+SDOS:IOBLOCKPTR
8480 EE0A        71:                 LDX     SCBLK:RDBUF,X
8482 C606        72:                 LDAB    #6
8484 32          73: CLOCKRB2        PULA
8485 A705        74:                 STAA    5,X
8487 09          75:                 DEX
8488 5A          76:                 DECB
8489 26F9        77:                 BNE     CLOCKRB2
848B 0C39        78:                 OKRTS
                 79: *
                 80: *
                 81: *
848D BDBE36      82: CLOCKREADA      JSR     SDOS+SDOS:CHECKRDLEN
8490 0011        83:                 FDB     17              ENOUGH FOR HH:MM:SS MM/DD/YY
8492 8D25        84:                 BSR     CLOCKGETTD      GET TIME, DATE FROM SDOS
8494 8D41        85:                 BSR     CLOCKDATE       FORMAT DATE
8496 8D72        86:                 BSR     CLOCKTIME       FORMAT TIME
8498 FEBE07      87:                 LDX     SDOS+SDOS:IOBLOCKPTR
849B 8611        88:                 LDAA    #17
849D A709        89:                 STAA    SCBLK:RPLEN+1,X
849F EE0A        90:                 LDX     SCBLK:RDBUF,X
84A1 DF00        91:                 STX     TEMPX
84A3 CE9002      92:                 LDX     #TIME$
84A6 E600        93: CLOCKRA1        LDAB    0,X
84A8 08          94:                 INX
84A9 DF02        95:                 STX     TEMPX+2
84AB DE00        96:                 LDX     TEMPX
84AD E700        97:                 STAB    0,X
84AF 08          98:                 INX
84B0 DF00        99:                 STX     TEMPX
84B2 DE02       100:                 LDX     TEMPX+2
84B4 4A         101:                 DECA
84B5 26EF       102:                 BNE     CLOCKRA1
84B7 0C39       103:                 OKRTS
                104: *
                105: *
                106: *
  84B9          107: CLOCKGETTD
84B9 01         108:                 NOP
84BA 0F         109:                 SEI
84BB CEBE00     110:                 LDX     #SDOS
84BE C606       111:                 LDAB    #6
84C0 A60B       112: CLOCKGETTD1     LDAA    SDOS:CLOCK,X
```

```
84C2 08        113:                    INX
84C3 36        114:                    PSHA
84C4 5A        115:                    DECB
84C5 26F9      116:                    BNE     CLOCKGETTD1
84C7 0E        117:                    CLI
84C8 CE8FF4    118:                    LDX     #CLOCKBUFFER
84CB C606      119:                    LDAB    #6
84CD 32        120: CLOCKGETTD2        PULA
84CE A705      121:                    STAA    5,X
84D0 09        122:                    DEX
84D1 5A        123:                    DECB
84D2 26F9      124:                    BNE     CLOCKGETTD2
84D4 39        125:                    RTS
               126: *
               127: *
               128: *
84D5 8DE2      129: DATE               BSR     CLOCKGETTD
84D7 B68FF8    130: CLOCKDATE          LDAA    MONTH
84DA 8D22      131:                    BSR     BCDTOASC
84DC B7900B    132:                    STAA    DATE$:MONTH
84DF F7900C    133:                    STAB    DATE$:MONTH+1
84E2 B68FF7    134:                    LDAA    DAY
84E5 8D17      135:                    BSR     BCDTOASC
84E7 B7900E    136:                    STAA    DATE$:DAY
84EA F7900F    137:                    STAB    DATE$:DAY+1
84ED B68FF9    138:                    LDAA    YEAR
84F0 8D0C      139:                    BSR     BCDTOASC
84F2 B79011    140:                    STAA    DATE$:YEAR
84F5 F79012    141:                    STAB    DATE$:YEAR+1
84F8 CE900B    142:                    LDX     #DATE$
84FB 8608      143:                    LDAA    #8
84FD 39        144:                    RTS
               145: *
84FE 16        146: BCDTOASC           TAB
84FF C40F      147:                    ANDB    #$F
8501 CB30      148:                    ADDB    #'0
8503 44        149:                    LSRA
8504 44        150:                    LSRA
8505 44        151:                    LSRA
8506 44        152:                    LSRA
8507 8B30      153:                    ADDA    #'0
8509 39        154:                    RTS
               155: *
               156: *
850A BD852E    157: CLOCKTIME          JSR     DIVIDEBY60      NOW DIVIDEND HAS SECONDS
850D CE9008    158:                    LDX     #TIME$:SECONDS
8510 8D10      159:                    BSR     CLOCKMAKEXX
8512 CE9005    160:                    LDX     #TIME$:MINUTES
8515 8D0B      161:                    BSR     CLOCKMAKEXX
8517 CE9002    162:                    LDX     #TIME$:HOURS
851A 8D06      163:                    BSR     CLOCKMAKEXX
851C CE9002    164:                    LDX     #TIME$
851F 8608      165:                    LDAA    #8
8521 39        166:                    RTS
               167: *
```

```
8522 BD852E     168: CLOCKMAKEXX        JSR     DIVIDEBY60
8525 8B30       169:                    ADDA    #'0
8527 CB30       170:                    ADDB    #'0
8529 E700       171:                    STAB    0,X
852B A701       172:                    STAA    1,X
852D 39         173:                    RTS
                174: *
                175: *                  DIVIDE BY 60 -- DIVIDE 3 BYTE "DIV60DIVIDEND" BY 60
                176: *                          DIVIDEND:=DIVIDEND/60
                177: *                          (A):=REMAINDER MOD 10
                178: *                          (B):=INT(REMAINDER/10)
  852E          179: DIVIDEBY60         EQU     *
852E C619       180:                    LDAB    #3*8+1          NUMBER OF BITS
8530 4F         181:                    CLRA
  8531          182: DIVIDE60L          EQU     *
8531 49         183:                    ROLA
8532 813C       184:                    CMPA    #60
8534 2504       185:                    BCS     DIVIDE60L2
8536 803C       186:                    SUBA    #60
8538 0D         187:                    SEC
8539 85         188:                    $85                     SKIP THE NEXT INSTRUCTION
853A 0C         189: DIVIDE60L2         CLC
853B 798FF6     190:                    ROL     DIV60DIVIDEND+2
853E 798FF5     191:                    ROL     DIV60DIVIDEND+1
8541 798FF4     192:                    ROL     DIV60DIVIDEND+0
8544 5A         193:                    DECB
8545 26EA       194:                    BNE     DIVIDE60L
8547 C6FF       195:                    LDAB    #-1
8549 5C         196: DIVIDE60L3         INCB
854A 800A       197:                    SUBA    #10
854C 24FB       198:                    BCC     DIVIDE60L3
854E 8B0A       199:                    ADDA    #10
8550 39         200:                    RTS
                201:                    FIN     IODRIVERBODY
  0000          202:                    IF      IODRIVERRAM
                228:                    FIN     IODRIVERRAM
                229:
                230:
                402:            FIN
  0000          403:            IF      BLACKHOLE
                405:            FIN     BLACKHOLE
  0000          406:            IF      SDLP
                408:            FIN     SDLP
  0001          409:            IF      VIRTUALFLOPPY
                410:            INCLUDE                         IOVFD.ASM
  0001            1:                  IF      IODRIVERBODY
                  2: *                PHYSICAL DISK DRIVERS STORAGE "DEFS"
                  3:
  8551            4: ::               SET     *
  0042            5:                  ORG     DSKINFO:SIZE
                  6:
                  7: *                TACKS ON TO BOTTOM OF DISK INFO TABLE
                  8:
0042 0001         9: FDREADWRITE      RMB     1               0 IS READ, <>0 IS WRITE
0043 0001        10: FDDSTATEJ        RMB     1               JMP instruction
```

IOVFD.ASM

```
0044 0002        11: FDDSTATE      RMB     2               address for JMP instruction
0046 0001        12: FDSEEKRETRY   RMB     1               NUMBER OF RE-SEEKS
0047 0001        13: FDRETRY       RMB     1               READ/WRITE RETRY COUNT
0048 0001        14: FDDRIVE       RMB     1               DRIVE NUMBER TO USE
0049 0001        15: FDCYL         RMB     1               what track we're on (-1 if lost)
004A 0001        16: FDSECTOR      RMB     1               GIMME THIS ONE
004B 0001        17: FDCOMPLEMENT  RMB     1               COMPLEMENT DATA
004C 0001        18: FDFIRSTSEC    RMB     1               FIRST SECTOR ON TRACK
004D 0002        19: FDHEADCHAIN   RMB     2               head of shared-head queue
004F 0002        20: FDNEXTCHAIN   RMB     2               next on shared-head queue
0051 0002        21: FDCCB         RMB     2               address of controller table
0053 0002        22: FDMAPALG      RMB     2               current map algorithm
0055 0001        23: FDK1MODNSPT   RMB     1               SPIRALING CONSTANT MOD NSPT
0056 0001        24: FDK2MODNSPT   RMB     1               2*SC MOD NSPT
0057 0001        25: FDK4MODNSPT   RMB     1               4*SC MOD NSPT
0058 0001        26: FDK8MODNSPT   RMB     1               8*SC MOD NSPT
0059 0001        27: FDK16MODNSPT  RMB     1               16*SC MOD NSPT
005A 0001        28: FDK32MODNSPT  RMB     1               32*SC MOD NSPT
  005B           29: FDMAP         EQU     *               MAP FOR MAPPING
                 30: ; virtual floppy dcb allocates room needed for FDMAP
  005B           31: FDSIZE        EQU     *
```

```
                33: *               Controller Tables
                34:
  0000          35:                 ORG     0
                36:
0000 0001       37: CCB:BUSY        RMB     1           controller is busy
0001 0002       38: CCB:ADDR        RMB     2           controller address
0003 0001       39: CCB:TIMEOUT     RMB     1           seconds before controller times out
0004 0001       40: CCB:DRIVE       RMB     1           drive to access
0005 0001       41: CCB:CYL         RMB     1           cylinder to access on that drive
0006 0001       42: CCB:LASTCYL     RMB     1           last cylinder accessed, that drive
0007 0002       43: CCB:STARTIO     RMB     2           address of STARTIO routine
0009 0003       44: CCB:STATUS      RMB     3           call for status
000C 0003       45: CCB:RESET       RMB     3           call to abort and interrupt
000F 0003       46: CCB:ABORT       RMB     3           call to abort
0012 0003       47: CCB:RESTORE     RMB     3           call to restore drive
0015 0003       48: CCB:SETSEEK     RMB     3           call to set desired drive and track
0018 0003       49: CCB:SEEK        RMB     3           call to initiate seek
001B 0003       50: CCB:READSECTOR  RMB     3           call to read sector
001E 0003       51: CCB:WRITESECTOR RMB     3           call to write sector
0021 0003       52: CCB:VERIFYSECTOR RMB    3           call to verify sector just written
0024 0008       53: CCB:TIMEOUTBLK  RMB     TIMEOUT:SIZE timeout block for controller
002C 0002       54: CCB:CURRENTDCB  RMB     2           address of current DCB
  002E          55: CCB:SIZE        EQU     *
  8551          56:                 ORG     ::
                57:                 FIN     IODRIVERBODY
  0000          58:                 IF      IODRIVERRAM
                127:            FIN     IODRIVERRAM
  0000          128:               IF      IODRIVERINIT
                172:               FIN     IODRIVERINIT
  0001          173:            IF      IODRIVERBODY
```

```
8551 9DE3       175: FDDRIVER   FDB     FDRESTORE
8553 856D       176:            FDB     FDREAD
8555 8569       177:            FDB     FDWRITE
8557 858F       178:            FDB     FDWAITDONE
8559 8566       179:            FDB     FDSTATUS
855B 855D       180:            FDB     FDCONTROL
                181:
                182:
                183: *          FDCONTROL -- CONTROL OPERATION ENTRY POINT FOR SECTOR I/O DRIVER
                184:
855D B111       185: FDCONTROL  CMPA    #CC:DISMOUNTDISK        SINCE SDOS PASSES THIS THRU
855F 2703       186:            BEQ     FDDISMOUNT              B/ ITS A DISMOUNT!
8561 7E8EDB     187:            JMP     ILLDEVICEOP             NOT A LEGAL CONTROL CALL
                188:
8564 0C39       189: FDDISMOUNT OKRTS                           I'M HAPPY...
                190:
                191: *          FDSTATUS-- HANDLE STATUS REQUEST
                192:
  8566          193: FDSTATUS
8566 7E8EDB     194:            JMP      ILLDEVICEOP           ; NO SUCH STATUS AVALIABLE
                195:
```

```
                197: *          FDREAD/WRITE -- START SINGLE SECTOR TRANSFER
                198:
8569 8601       199: FDWRITE    LDAA    #1
856B 2001       200:            BRA     FDREAD.1
856D 4F         201: FDREAD     CLRA
856E BD85AF     202: FDREAD.1   JSR     FDSETUPDRIVE        GO SET UP ALL THE PARAMETERS IN THE DCB
8571 EE51       203:            LDX     FDCCB,X             see if controller is busy
8573 6D00       204:            TST     CCB:BUSY,X
8575 2603       205:            BNE     FDSTARTIO
8577 BDBE2A     206:            JSR     SDOS+SDOS:WAITEVENT
857A DE06       207: FDSTARTIO  LDX     DCBPOINTER          NOW NOBODY'S USING DRIVE
857C 6F00       208:            CLR     DCB:DONEFLAG,X      KICK INTERRUPT ROUTINE INTO MOTION
857E EE51       209:            LDX     FDCCB,X             point controller table at this DCB
8580 9606       210:            LDAA    DCBPOINTER
8582 D607       211:            LDAB    DCBPOINTER+1
8584 A72C       212:            STAA    CCB:CURRENTDCB,X
8586 E72D       213:            STAB    CCB:CURRENTDCB+1,X
8588 EE07       214:            LDX     CCB:STARTIO,X
858A BDBE24     215:            JSR     SDOS+SDOS:STARTIO
858D 0C39       216:            OKRTS
                217:
  858F          218: FDWAITDONE
858F A600       219:            LDAA    DCB:DONEFLAG,X      IS IT DONE?
8591 2605       220:            BNE     FDWAIT1             B/ YES
8593 BDBE2A     221:            JSR     SDOS+SDOS:WAITEVENT
8596 DE06       222:            LDX     DCBPOINTER
8598 EE01       223: FDWAIT1    LDX     DCB:LASTERROR,X
859A 2703       224:            BEQ     FDWAIT2             B/ NO ERRORS
859C 7E8EE0     225:            JMP     ERRETX
859F 0C39       226: FDWAIT2    OKRTS
```

```
  85A1          228: MODULONSPTB
85A1 E00C       229:          SUBB    DSKINFO:NSPT+1,X
85A3 24FC       230:          BCC     MODULONSPTB
85A5 EB0C       231:          ADDB    DSKINFO:NSPT+1,X
85A7 39         232:          RTS
                233:
  85A8          234: MODULONSPT
85A8 A00C       235:          SUBA    DSKINFO:NSPT+1,X
85AA 24FC       236:          BCC     MODULONSPT
85AC AB0C       237:          ADDA    DSKINFO:NSPT+1,X
85AE 39         238:          RTS
                239:
                240: *        FDSETUPDRIVE -- SETS UP FDDRIVE TABLE FOR INTERRUPT DRIVEN TRANSFER
                241:
  85AF          242: FDSETUPDRIVE
85AF DE06       243:          LDX     DCBPOINTER
85B1 A742       244:          STAA    FDREADWRITE,X     SAVE THE READ/WRITE FLAG
85B3 C60A       245:          LDAB    #10
85B5 E747       246:          STAB    FDRETRY,X         SAVE THE READ/WRITE RETRY COUNT
85B7 8604       247:          LDAA    #4                SETUP SEEK RETRY COUNT
85B9 A746       248:          STAA    FDSEEKRETRY,X
85BB 6F01       249:          CLR     DCB:LASTERROR,X   "NO ERRORS"
85BD 6F02       250:          CLR     DCB:LASTERROR+1,X
85BF E616       251:          LDAB    DSKINFO:MAPALGORITHM,X
85C1 A617       252:          LDAA    DSKINFO:MAPALGORITHM+1,X
85C3 E153       253:          CMPB    FDMAPALG,X
85C5 2604       254:          BNE     FDSETUP1          B/ MAP HAS CHANGED
85C7 A154       255:          CMPA    FDMAPALG+1,X
85C9 2742       256:          BEQ     FDSETUP2          B/ MAP HAS NOT CHANGED
85CB E753       257: FDSETUP1 STAB    FDMAPALG,X
85CD A754       258:          STAA    FDMAPALG+1,X
                259:
85CF DF00       260: BUILDMAP STX     TEMPX
85D1 40         261:          NEGA
85D2 E60C       262:          LDAB    DSKINFO:NSPT+1,X  NUMBER OF TIMES TO DO THIS
85D4 DE06       263: BUILDMAP1 LDX    DCBPOINTER
85D6 AB17       264:          ADDA    DSKINFO:MAPALGORITHM+1,X
85D8 8D06       265: BUILDMAP2 BSR    MODULONSPT
85DA DF00       266:          LDX     DCBPOINTER        SEE IF THIS SECTOR NUMBER
85DC 09         267:          DEX                         ALREADY USED
85DD 08         268: BUILDMAP3 INX
85DE 9C00       269:          CPX     TEMPX
85E0 2707       270:          BEQ     BUILDMAP4         B/ NOT USED, USE IT
85E2 A15B       271:          CMPA    FDMAP,X
85E4 26F7       272:          BNE     BUILDMAP3         B/ IT'S NOT THIS ONE, KEEP LOOKING
85E6 4C         273:          INCA                      OH WELL, LET'S BUMP IT AND TRY AGAIN
85E7 20EF       274:          BRA     BUILDMAP2
85E9 A75B       275: BUILDMAP4 STAA   FDMAP,X
85EB 08         276:          INX
85EC DF00       277:          STX     TEMPX
85EE 5A         278:          DECB                      ARE WE DONE BUILDING THE MAP?
85EF 26E3       279:          BNE     BUILDMAP1         B/ NOPE
                280:
                281: *        BUILD UP THE SPIRALING INFO
                282:
```

page

IOVFD.ASM

```
85F1 DE06      283:          LDX     DCBPOINTER
85F3 A653      284:          LDAA    FDMAPALG,X
85F5 8DB1      285:          BSR     MODULONSPT
85F7 A755      286:          STAA    FDK1MODNSPT,X
85F9 48        287:          ASLA
85FA 8DAC      288:          BSR     MODULONSPT
85FC A756      289:          STAA    FDK2MODNSPT,X
85FE 48        290:          ASLA
85FF 8DA7      291:          BSR     MODULONSPT
8601 A757      292:          STAA    FDK4MODNSPT,X
8603 48        293:          ASLA
8604 8DA2      294:          BSR     MODULONSPT
8606 A758      295:          STAA    FDK8MODNSPT,X
8608 48        296:          ASLA
8609 8D9D      297:          BSR     MODULONSPT
860B A759      298:          STAA    FDK16MODNSPT,X
```

```
  860D          300: FDSETUP2 ; COMPUTE TARGET CYLINDER AND SECTOR
860D EE2B       301:          LDX     DSKINFO:SECTORDB,X
860F A603       302:          LDAA    RDSI:LSN+1,X      GET LSN
8611 E604       303:          LDAB    RDSI:LSN+2,X
8613 DE06       304:          LDX     DCBPOINTER        SO WE CAN POKE AT DCB AGAIN
                305: ;
                306: ;  Now generate 8 quotient bits (enough for 255 tracks!)
                307: ;
8615 58         308:          ASLB                      it takes 8 ASLD's to shift sector
8616 49         309:          ROLA                        number into upper byte
8617 A00C       310:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
8619 2402       311:          BCC     *+4               b/ did go in, quotient bit is 1
861B AB0C       312:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
861D 694A       313:          ROL     FDSECTOR,X        save complement of quotient bit
861F 58         314:          ASLB                      double the dividend
8620 49         315:          ROLA
                316:
8621 A00C       317:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
8623 2402       318:          BCC     *+4               b/ did go in, quotient bit is 1
8625 AB0C       319:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
8627 694A       320:          ROL     FDSECTOR,X        save complement of quotient bit
8629 58         321:          ASLB                      double the dividend
862A 49         322:          ROLA
                323:
862B A00C       324:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
862D 2402       325:          BCC     *+4               b/ did go in, quotient bit is 1
862F AB0C       326:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
8631 694A       327:          ROL     FDSECTOR,X        save complement of quotient bit
8633 58         328:          ASLB                      double the dividend
8634 49         329:          ROLA
                330:
8635 A00C       331:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
8637 2402       332:          BCC     *+4               b/ did go in, quotient bit is 1
8639 AB0C       333:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
863B 694A       334:          ROL     FDSECTOR,X        save complement of quotient bit
863D 58         335:          ASLB                      double the dividend
863E 49         336:          ROLA
                337:
863F A00C       338:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
8641 2402       339:          BCC     *+4               b/ did go in, quotient bit is 1
8643 AB0C       340:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
8645 694A       341:          ROL     FDSECTOR,X        save complement of quotient bit
8647 58         342:          ASLB                      double the dividend
8648 49         343:          ROLA
                344:
8649 A00C       345:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
864B 2402       346:          BCC     *+4               b/ did go in, quotient bit is 1
864D AB0C       347:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
864F 694A       348:          ROL     FDSECTOR,X        save complement of quotient bit
8651 58         349:          ASLB                      double the dividend
8652 49         350:          ROLA
                351:
8653 A00C       352:          SUBA    DSKINFO:NSPT+1,X  Compute quotient bit
8655 2402       353:          BCC     *+4               b/ did go in, quotient bit is 1
8657 AB0C       354:          ADDA    DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
```

```
8659 694A      355:         ROL    FDSECTOR,X        save complement of quotient bit
865B 58        356:         ASLB                     double the dividend
865C 49        357:         ROLA
               358:
865D A00C      359:         SUBA   DSKINFO:NSPT+1,X  Compute quotient bit
865F 2402      360:         BCC    *+4               b/ did go in, quotient bit is 1
8661 AB0C      361:         ADDA   DSKINFO:NSPT+1,X  didn't go in, quotient bit is zero
8663 694A      362:         ROL    FDSECTOR,X        save complement of quotient bit
               363: ;       ASLB                     double the dividend
               364: ;       ROLA
               365:
8665 E64A      366:         LDAB   FDSECTOR,X        get complement of desired track
8667 53        367:         COMB                     invert the inverted quotient bits
8668 A74A      368:         STAA   FDSECTOR,X        save sector within track
866A EE2B      369:         LDX    DSKINFO:SECTORDB,X now save cylinder number in RDSI
866C 6F11      370:         CLR    RDSI:CYLINDER,X
866E E712      371:         STAB   RDSI:CYLINDER+1,X
8670 6F0D      372:         CLR    RDSI:SECTOR,X
8672 6F0E      373:         CLR    RDSI:SECTOR+1,X
8674 DE06      374:         LDX    DCBPOINTER
```

IOVFD.ASM

```
8676 EE53      376:          LDX     FDMAPALG,X        apply mapalgorithm...
8678 8C0001    377:          CPX     #$0001              unless it is :0001
867B 2734      378:          BEQ     FDSETUP4          B/ map algorithm :0001, all done!
867D 9B07      379:          ADDA    DCBPOINTER+1
867F 9701      380:          STAA    TEMPX+1
8681 9606      381:          LDAA    DCBPOINTER
8683 8900      382:          ADCA    #0
8685 9700      383:          STAA    TEMPX
8687 4F        384:          CLRA                      make spiral in (A)
8688 DE06      385:          LDX     DCBPOINTER        assert: cylinder number in (B)
868A BD85A1    386:          JSR     MODULONSPTB
868D 57        387:          ASRB
868E 2402      388:          BCC     MAP1
8690 AB55      389:          ADDA    FDK1MODNSPT,X
8692 57        390: MAP1     ASRB
8693 2402      391:          BCC     MAP2
8695 AB56      392:          ADDA    FDK2MODNSPT,X
8697 57        393: MAP2     ASRB
8698 2402      394:          BCC     MAP3
869A AB57      395:          ADDA    FDK4MODNSPT,X
869C 57        396: MAP3     ASRB
869D 2402      397:          BCC     MAP4
869F AB58      398:          ADDA    FDK8MODNSPT,X
86A1 57        399: MAP4     ASRB
86A2 2402      400:          BCC     MAP5
86A4 AB59      401:          ADDA    FDK16MODNSPT,X
86A6 DE00      402: MAP5     LDX     TEMPX
86A8 AB5B      403:          ADDA    FDMAP,X
86AA DE06      404:          LDX     DCBPOINTER
86AC BD85A8    405:          JSR     MODULONSPT
86AF A74A      406:          STAA    FDSECTOR,X
  86B1         407: FDSETUP4
86B1 DE06      408:          LDX     DCBPOINTER
86B3 0C39      409:          OKRTS
               410:          FIN     IODRIVERBODY
  0000         411:          IF      IODRIVERPOLL
               434:        FIN    IODRIVERPOLL
  0001         435:        IF     IODRIVERBODY
  86B5         436: DISKINTERRUPT ; entered with CCB address in (X)
86B5 8D02      437:        BSR    DISKINTSETUP          set up a working context area
86B7 6E43      438:        JMP    FDDSTATEJ,X           resume process waiting for interrupt
```

```
  86B9          440: DISKINTSETUP   ; set up a context area of sorts
86B9 FF9015     441:         STX    DISKINTCCB             remember interface table address
86BC EE2C       442:         LDX    CCB:CURRENTDCB,X
86BE FF9013     443:         STX    DISKINTDCB
86C1 39         444:         RTS
                445:
  0002          446:         IF     PERSCI
  86C2          447: DISKINTSTARTPERSCI                    ; ASSERT: INTERRUPTS ARE DISABLED HERE!
86C2 CE9017     448:         LDX    #CCB:PERSCI
  0002          449:         IF     DAMFLOPPY
86C5 2003       450:         BRA    DISKINTSTART
                451:
                452:         FIN    DAMFLOPPY
                453:         FIN    PERSCI
  0002          454:         IF     DAMFLOPPY
  86C7          455: DISKINTSTARTDAMFLOPPY                  ; ASSERT: INTERRUPTS ARE DISABLED HERE!
86C7 CE9045     456:         LDX    #CCB:DAMFLOPPY
                457:         FIN    DAMFLOPPY
  86CA          458: DISKINTSTART
86CA 6F00       459:         CLR    CCB:BUSY,X             mark controller busy
                460: *       CLI                           (allow interrupts)
86CC 8606       461:         LDAA   #6                     Set up for 6 1-second interrupts
86CE A703       462:         STAA   CCB:TIMEOUT,X            to keep disk spinning
86D0 8600       463:         LDAA   #(1*TICKSPERSECOND+NTIMEOUTBLOCKS)/256
86D2 C645       464:         LDAB   #(1*TICKSPERSECOND+NTIMEOUTBLOCKS)&$FF
86D4 A726       465:         STAA   CCB:TIMEOUTBLK+TIMEOUT:FUSE,X
86D6 E727       466:         STAB   CCB:TIMEOUTBLK+TIMEOUT:FUSE+1,X
86D8 BDDF       467:         BSR    DISKINTSETUP           set up a working context area
86DA A642       468:         LDAA   FDREADWRITE,X          a write to an IBM format disk
86DC 2707       469:         BEQ    SEEK                     must have the data complemented
86DE 6D4B       470:         TST    FDCOMPLEMENT,X             before it is written
86E0 2703       471:         BEQ    SEEK                         (and complemented back, after
86E2 BD87FB     472:         JSR    DISKCOMPLEMENT                 it has been written)
```

```
                    474: *      See if the head must be moved with a seek operation;
                    475: *      if it must, the seek is done without verification, as a seek
                    476: *      failure will be picked up by a subsequent read/write as
                    477: *      "record not found" status, for which the remedy will be
                    478: *      a restore operation.  The restore operation IS verified; if
                    479: *      it fails, a "seek error" is registered, and the restore is
                    480: *      retried, up to the seek-retry count.
                    481: *
                    482: *
86E5 FE9015         483: SEEK   LDX    DISKINTCCB           announce intentions
86E8 AD15           484:        JSR    CCB:SETSEEK,X
86EA 2412           485:        BCC    SEEKDONE             B/ no seek necessary
86EC C1FF           486:        CMPB   #-1                  a seek is necessary: if the B register
86EE 2730           487:        BEQ    SEEKHOME               contains a -1, then I am lost and
86F0 FE9015         488:        LDX    DISKINTCCB               must do a restore; otherwise, a
86F3 AD18           489:        JSR    CCB:SEEK,X                 standard seek will suffice
86F5 EE2B           490:        LDX    DSKINFO:SECTORDB,X   I assert that I am on the right track
86F7 A612           491:        LDAA   RDSI:CYLINDER+1,X
86F9 FE9013         492:        LDX    DISKINTDCB           remember which track we are on
86FC 8D76           493:        BSR    DISKSETCYLADD
  86FE              494: SEEKDONE
86FE EE2B           495:        LDX    DSKINFO:SECTORDB,X   pick up the buffer page number in the
8700 E605           496:        LDAB   RDSI:SECTORBASE,X      B register.  pick up the sector number
8702 FE9013         497:        LDX    DISKINTDCB               in the A register and offset by
8705 A64A           498:        LDAA   FDSECTOR,X                 the track base sector
8707 AB4C           499:        ADDA   FDFIRSTSEC,X
8709 6D42           500:        TST    FDREADWRITE,X        go off and do the read or write, as
870B 2673           501:        BNE    DISKWRITE              appropriate
870D 7E87D4         502:        JMP    DISKREAD
                    503:
                    504: **** THE WESTERN DIGITAL TRICK OF STEP IN ONE/STEP OUT ONE SHOULD
                    505: **** BE ADDED TO MAKE THE DRIVER MORE ROBUST.
                    506: **** BUT DENNIS PAINTER SEZ IT DOESN'T WORK ON A PERSCI.
```

should set timeout on seek, and on read/write instead of over entire operation.

```
8710 A733       508: SEEK3   STAA   DSKINFO:SEEKERRSTS+1,X  count a seek error, and
8712 6C31       509:         INC    DSKINFO:SEEKERRCNT+1,X    save the error status
8714 2602       510:         BNE    SEEK3.1
8716 6C30       511:         INC    DSKINFO:SEEKERRCNT,X
8718 86FF       512: SEEK3.1 LDAA   #-1                    Say that I lost my place...
871A 8D58       513:         BSR    DISKSETCYLADD
871C 6A46       514:         DEC    FDSEEKRETRY,X          DOWN COUNT # TRIES LEFT
871E 2715       515:         BEQ    DISKSEEKERROR          B/ GAK! CROAK! DIE....
  8720          516: SEEKHOME
8720 8D4D       517:         BSR    DISKABORT              KILL WHATEVER DISK IS DOING
8722 FE9015     518:         LDX    DISKINTCCB
8725 AD12       519:         JSR    CCB:RESTORE,X
8727 FE9015     520:         LDX    DISKINTCCB
872A AD09       521:         JSR    CCB:STATUS,X           *** why doesn't this check for success?
872C 8504       522:         BITA   #%00000100             (ON OTHER HAND, IF CYL 0, WHO CARES?)
872E 27E0       523:         BEQ    SEEK3                  B/ DIDN'T GET TO CYL 0 FOR SOME REASON!?
8730 4F         524:         CLRA
8731 8D41       525:         BSR    DISKSETCYLADD          SET "I'M AT CYLINDER 0"
8733 20B0       526:         BRA    SEEK                   GO TRY SEEK TO PROPER TRACK AGAIN
```

```
  8735          528: DISKSEEKERROR
8735 8604       529:          LDAA   #ERR:DISKSEEK/256       GET APPROPRIATE ERROR CODE
8737 C617       530:          LDAB   #ERR:DISKSEEK&$FF
8739 2012       531:          BRA    DISKERROR1
                532:
  873B          533: DISKWPERR
873B 8604       534:          LDAA   #ERR:DSKWRTPROT/256
873D C618       535:          LDAB   #ERR:DSKWRTPROT&$FF
873F 200C       536:          BRA    DISKERROR1
                537:
  8741          538: DISKERROR ; fatal read or write error occurred
8741 8604       539:          LDAA   #ERR:DISKREAD/256       ASSUME READ ERROR
8743 C615       540:          LDAB   #ERR:DISKREAD&$FF
8745 6D42       541:          TST    FDREADWRITE,X           WAS IT A READ OR A WRITE?
8747 2704       542:          BEQ    DISKERROR1              B/ IT'S A READ
8749 8604       543:          LDAA   #ERR:DISKWRITE/256
874B C616       544:          LDAB   #ERR:DISKWRITE&$FF
  874D          545: DISKERROR1
874D A701       546:          STAA   DCB:LASTERROR,X
874F E702       547:          STAB   DCB:LASTERROR+1,X
  8751          548: DISKDONE
8751 6C00       549:          INC    DCB:DONEFLAG,X          SIGNAL "DISK DONE"
8753 FE9015     550:          LDX    DISKINTCCB
8756 6C00       551:          INC    CCB:BUSY,X              SO TASK KNOWS WE'RE FREE
  8758          552: DISKDONE1
8758 FE9013     553:          LDX    DISKINTDCB
875B BD8828     554:          JSR    WAITFORINTERRUPT
  875E          555: DISKINTUNEXPECTED
875E 8D0F       556:          BSR    DISKABORT
8760 20F6       557:          BRA    DISKDONE1
```

```
  8762          559: CHECKDISKREADY
                560: ; It would be nice if this could be used to tell one that the drive
                561: ; did not have a diskette in it... Thank you Dennis Brown
8762 FE9015     562:         LDX    DISKINTCCB             return carry set if not ready;
8765 AD09       563:         JSR    CCB:STATUS,X             carry reset if ready--
8767 16         564:         TAB                               in either case, status is in A
8768 59         565:         ROLB
8769 39         566:         RTS
                567:
  876A          568: MAKEDISKREADY
876A FE9015     569:         LDX    DISKINTCCB
876D 6E0C       570:         JMP    CCB:RESET,X
                571:
  876F          572: DISKABORT
876F FE9015     573:         LDX    DISKINTCCB             abort whatever's being done
8772 6E0F       574:         JMP    CCB:ABORT,X
                575:
  8774          576: DISKSETCYLADD ; mark DCBs that share heads as all being on track (A)
8774 EE4D       577:         LDX    FDHEADCHAIN,X
  8776          578: DISKSETCYLADD.1
8776 A749       579:         STAA   FDCYL,X                all DCB's sharing the same head
8778 EE4F       580:         LDX    FDNEXTCHAIN,X            mechanism are chained together
877A 26FA       581:         BNE    DISKSETCYLADD.1            so that all FDCYL values will
877C FE9013     582:         LDX    DISKINTDCB                   be equally correct
877F 39         583:         RTS
```

```
  8780          585: DISKWRITE
8780 FE9015     586:         LDX    DISKINTCCB
8783 AD1E       587:         JSR    CCB:WRITESECTOR,X
8785 8DDB       588:         BSR    CHECKDISKREADY
8787 8540       589:         BITA   #%01000000              IS DISK WRITE PROTECTED ?
8789 26B0       590:         BNE    DISKWPERR               B/ YEP.
878B 2404       591:         BCC    DISKWRITE2              B/ READY
878D 8DDB       592:         BSR    MAKEDISKREADY           since the drive must have shut down,
878F 200F       593:         BRA    SEEKDONEJ                 we'll try this again
                594:
  8791          595: DISKWRITE2
8791 857C       596:         BITA   #%01111100              Is the write OK?
8793 2612       597:         BNE    DISKWRITE4              B/ NO
8795 FE9015     598:         LDX    DISKINTCCB
8798 AD21       599:         JSR    CCB:VERIFYSECTOR,X      do a verify
879A 8DC6       600:         BSR    CHECKDISKREADY          WELL?
879C 2405       601:         BCC    DISKWRITE3              B/ 10-4
879E 8DCA       602:         BSR    MAKEDISKREADY           It's not, so we'll try the write again
  87A0          603: SEEKDONEJ
87A0 7E86FE     604:         JMP    SEEKDONE
```

```
  87A3          606: DISKWRITE3
87A3 8518       607:         BITA   #%00011000             record not found or CRC error?
87A5 273E       608:         BEQ    DISKDONEJ1             B/ no--everything's OK
  87A7          609: DISKWRITE4
87A7 A736       610:         STAA   DSKINFO:WRITEERRSTS,X   SAVE WRITE ERROR STATUS
87A9 8D19       611:         BSR    DISKSAVEERRLSN         SAVE ERRORING LSN
87AB 6C35       612:         INC    DSKINFO:WRITEERRCNT+1,X COUNT # WRITE ERRORS
87AD 2602       613:         BNE    DISKWRITE5
87AF 6C34       614:         INC    DSKINFO:WRITEERRCNT,X
  87B1          615: DISKWRITE5
87B1 6A47       616:         DEC    FDRETRY,X
87B3 270C       617:         BEQ    DISKERRORJ             B/ NO MORE TRIES LEFT
87B5 E647       618:         LDAB   FDRETRY,X              ON LAST TRY ?
87B7 5A         619:         DECB                          (=1?)
87B8 2704       620:         BEQ    SEEKHOMEJ              B/ YES, TRY HARDER
87BA 8510       621:         BITA   #%00010000             NO, DID WE GET "RECORD NOT FOUND" ?
87BC 27E2       622:         BEQ    SEEKDONEJ              B/ NOPE, TRY READ/WRITE AGAIN
  87BE          623: SEEKHOMEJ
87BE 7E8720     624:         JMP    SEEKHOME               GO SEE IF RE-SEEK HELPS
                625:
  87C1          626: DISKERRORJ
87C1 7E8741     627:         JMP    DISKERROR
                628:
  87C4          629: DISKSAVEERRLSN ; save RDSI:LSN in DCB
87C4 EE2B       630:         LDX    DSKINFO:SECTORDB,X
87C6 A603       631:         LDAA   RDSI:LSN+1,X
87C8 E604       632:         LDAB   RDSI:LSN+2,X
87CA FE9013     633:         LDX    DISKINTDCB
87CD 6F3F       634:         CLR    DSKINFO:ERRLSN,X
87CF A740       635:         STAA   DSKINFO:ERRLSN+1,X
87D1 E741       636:         STAB   DSKINFO:ERRLSN+2,X
87D3 39         637:         RTS
```

PSHA

save disk status

PULA

X

B

IOVFD.ASM

```
  87D4          639: DISKREAD
87D4 FE9015     640:         LDX    DISKINTCCB
87D7 AD1B       641:         JSR    CCB:READSECTOR,X
87D9 8D87       642:         BSR    CHECKDISKREADY
87DB 2404       643:         BCC    DISKREAD1               B/ READY
87DD 8D8B       644:         BSR    MAKEDISKREADY
87DF 20BF       645:         BRA    SEEKDONEJ
                646:
  87E1          647: DISKREAD1
87E1 851C       648:         BITA   #%00011100             Is the read OK?
87E3 260A       649:         BNE    DISKREAD4               B/ no
  87E5          650: DISKDONEJ1
87E5 6D4B       651:         TST    FDCOMPLEMENT,X          complement data?
87E7 2703       652:         BEQ    DISKDONEJ               B/ NO
87E9 8D87FB     653:         JSR    DISKCOMPLEMENT          YES, COMPLEMENT DATA BEFORE WE QUIT!
  87EC          654: DISKDONEJ
87EC 7E8751     655:         JMP    DISKDONE
                656:
  87EF          657: DISKREAD4
87EF A73A       658:         STAA   DSKINFO:READERRSTS,X    SAVE READ ERROR STATUS
87F1 8DD1       659:         BSR    DISKSAVEERRLSN          save erroring LSN
87F3 6C39       660:         INC    DSKINFO:READERRCNT+1,X  COUNT # READ ERRORS
87F5 26BA       661:         BNE    DISKWRITE5
87F7 6C38       662:         INC    DSKINFO:READERRCNT,X
87F9 20B6       663:         BRA    DISKWRITE5              GO CHECK RETRY COUNT
```

IOVFD.ASM

```
  87FB          665: DISKCOMPLEMENT ; COMPLEMENT SECTOR CONTENTS (FOR IBM 3740 FORMAT)
87FB A609       666:         LDAA   DSKINFO:NBPS,X          get sector size
87FD E60A       667:         LDAB   DSKINFO:NBPS+1,X
87FF EE2B       668:         LDX    DSKINFO:SECTORDB,X
8801 EE05       669:         LDX    RDSI:SECTORBASE,X       MUST COMPLEMENT THE DATA FIRST
8803 4C         670:         INCA                           TO OFFSET "DECA" BELOW ON 1st PASS
  8804          671: DISKCOMPL
8804 6300       672:         COM    0,X                     COMPLEMENT A BYTE
8806 08         673:         INX                            BUMP POINTER
8807 6300       674:         COM    0,X                     COMPLEMENT A BYTE
8809 08         675:         INX                            BUMP POINTER
880A 6300       676:         COM    0,X                     COMPLEMENT A BYTE
880C 08         677:         INX                            BUMP POINTER
880D 6300       678:         COM    0,X                     COMPLEMENT A BYTE
880F 08         679:         INX                            BUMP POINTER
8810 C004       680:         SUBB   #4                      = # BYTES LEFT TO COMPLEMENT
8812 26F0       681:         BNE    DISKCOMPL
8814 4A         682:         DECA
8815 26ED       683:         BNE    DISKCOMPL
8817 FE9013     684:         LDX    DISKINTDCB              TO BE NICE TO CALLER
881A 39         685:         RTS
```

```
 0002          687:         IF     DAMFLOPPY!PERSCI
               688: *               WM FLOPPY DISK HARDWARE DEFINITIONS
               689:
 0002          690:                      IF         PERSCI
 FFA0          691: PERSCI:PIACA         EQU        $FFA0
 FFA1          692: PERSCI:PIACB         EQU        $FFA1
 FFA2          693: PERSCI:PIADA         EQU        $FFA2        DMA PAGE NUMBER
 FFA3          694: PERSCI:PIADB         EQU        $FFA3        drive select, misc. control
 FFA4          695: PERSCI:WDCMDSTS      EQU        $FFA4        COMMAND / STATUS REGISTER
 FFA5          696: PERSCI:WDTRACK       EQU        $FFA5        CURRENT TRACK REGISTER
 FFA6          697: PERSCI:WDSECTOR      EQU        $FFA6        TARGET SECTOR REGISTER
 FFA7          698: PERSCI:WDDATA        EQU        $FFA7        TARGET TRACK /  DATA REGISTER
               699:                      FIN        PERSCI
               700:
 0002          701:                      IF         DAMFLOPPY
 FF80          702: DAMFLOPPY:PIACA      EQU        $FF80
 FF81          703: DAMFLOPPY:PIACB      EQU        $FF81
 FF82          704: DAMFLOPPY:PIADA      EQU        $FF82        DMA PAGE NUMBER
 FF83          705: DAMFLOPPY:PIADB      EQU        $FF83        drive select, misc. control
 FF84          706: DAMFLOPPY:WDCMDSTS   EQU        $FF84        COMMAND / STATUS REGISTER
 FF85          707: DAMFLOPPY:WDTRACK    EQU        $FF85        CURRENT TRACK REGISTER
 FF86          708: DAMFLOPPY:WDSECTOR   EQU        $FF86        TARGET SECTOR REGISTER
 FF87          709: DAMFLOPPY:WDDATA     EQU        $FF87        TARGET TRACK /  DATA REGISTER
               710:                      FIN        DAMFLOPPY
```

```
                712:          FIN     DAMFLOPPY!PERSCI
  881B          713: COUNTCOMMAND
881B FE9013     714:          LDX     DISKINTDCB
881E 6C3E       715:          INC     DSKINFO:OPSCOUNT+2,X       COUNT # OPERATIONS ISSUED TO FLOPPY
8820 2606       716:          BNE     WAITFORINTERRUPT
8822 6C3D       717:          INC     DSKINFO:OPSCOUNT+1,X
8824 2602       718:          BNE     WAITFORINTERRUPT
8826 6C3C       719:          INC     DSKINFO:OPSCOUNT,X
  8828          720: WAITFORINTERRUPT
8828 32         721:          PULA
8829 33         722:          PULB
882A A744       723:          STAA    FDDSTATE,X
882C E745       724:          STAB    FDDSTATE+1,X
882E 7EBE15     725:          JMP     SDOS+SDOS:RTI
```

```
                727: *       Test if an actual seek is required
                728:
                729: *       If the drive number has not changed, and the cylinder (track) number
                730: *       has not changed, then no seek is necessary, and return is made with
                731: *       carry clear; otherwise, return is made with carry set and the
                732: *       previous cylinder number in the B register.
                733:
                734: *       a side effect is that the values of FDCYL, FDTARGETCYL, and FDDRIVE
                735: *       are copied to the CCB, regardless of whether a seek is necessary
                736: *       (this ensures that the CCB is set up for a subsequent restore, read,
                737: *       or write)
                738:
  8831          739: TESTFORSEEK
8831 A604       740:         LDAA   CCB:DRIVE,X
8833 FE9013     741:         LDX    DISKINTDCB
8836 A148       742:         CMPA   FDDRIVE,X
8838 2623       743:         BNE    DOSEEK
883A A649       744:         LDAA   FDCYL,X
883C EE2B       745:         LDX    DSKINFO:SECTORDB,X
883E A112       746:         CMPA   RDSI:CYLINDER+1,X
8840 261B       747:         BNE    DOSEEK
8842 FE9013     748:         LDX    DISKINTDCB
                749: *       BSR    COPYDCBTOCCB
                750: *       OKRTS
                751:
  8845          752: COPYDCBTOCCB
8845 A648       753:         LDAA   FDDRIVE,X
8847 36         754:         PSHA
8848 E649       755:         LDAB   FDCYL,X
884A EE2B       756:         LDX    DSKINFO:SECTORDB,X
884C A612       757:         LDAA   RDSI:CYLINDER+1,X
884E FE9015     758:         LDX    DISKINTCCB
8851 A705       759:         STAA   CCB:CYL,X
8853 E706       760:         STAB   CCB:LASTCYL,X
8855 32         761:         PULA
8856 A704       762:         STAA   CCB:DRIVE,X
8858 FE9013     763:         LDX    DISKINTDCB
885B 0C39       764:         OKRTS
                765:
  885D          766: DOSEEK ; seek is required
885D FE9013     767:         LDX    DISKINTDCB
8860 8DE3       768:         BSR    COPYDCBTOCCB
8862 0D39       769:         ERRORRTS
                770:
  0002          771:         IF     PERSCI
```

```
  8864          773: PERSCI:STATUS
8864 B6FFA4     774:         LDAA   PERSCI:WDCMDSTS
8867 43         775:         COMA
8868 FE9013     776:         LDX    DISKINTDCB
886B 39         777:         RTS
                778:
  886C          779: PERSCI:RESTORE
886C A604       780:         LDAA   CCB:DRIVE,X            get the drive address and add in
886E 8A08       781:         ORAA   #%00001000               slow step, read, no DMA
8870 B7FFA3     782:         STAA   PERSCI:PIADB
8873 86FD       783:         LDAA   #(\%00000010)&$FF      restore
  8875          784: PERSCI:ISSUECOMMAND
8875 B7FFA4     785:         STAA   PERSCI:WDCMDSTS
8878 7E881B     786:         JMP    COUNTCOMMAND
                787:
  887B          788: PERSCI:ABORT
                789: ; Note that the Series 2000 does something funny here; somebody should go look.
887B 862F       790:         LDAA   #(\%11010000)&$FF      abort with no interrupts
887D B7FFA4     791:         STAA   PERSCI:WDCMDSTS
8880 8D0E       792:         BSR    PERSCI:ABORT.RTS       wait about 30 uS for chip to settle
8882 8D0C       793:         BSR    PERSCI:ABORT.RTS
8884 8D0A       794:         BSR    PERSCI:ABORT.RTS
8886 F6FFA4     795:         LDAB   PERSCI:WDCMDSTS        return with status in B
8889 53         796:         COMB
888A B6FFA3     797:         LDAA   PERSCI:PIADB           clear possible interrupt
888D FE9013     798:         LDX    DISKINTDCB
  8890          799: PERSCI:ABORT.RTS
8890 39         800:         RTS
                801:
  8891          802: PERSCI:RESET
8891 862E       803:         LDAA   #(\%11010001)&$FF      abort with interrupt
8893 20E0       804:         BRA    PERSCI:ISSUECOMMAND
```

```
  8895          806: PERSCI:SETSEEK
8895 209A       807:          BRA     TESTFORSEEK
                808:
  8897          809: PERSCI:SEEK
8897 A604       810:          LDAA    CCB:DRIVE,X
8899 B7FFA3     811:          STAA    PERSCI:PIADB
889C A606       812:          LDAA    CCB:LASTCYL,X
889E E605       813:          LDAB    CCB:CYL,X
88A0 43         814:          COMA
88A1 B7FFA5     815:          STAA    PERSCI:WDTRACK
88A4 53         816:          COMB
88A5 F7FFA7     817:          STAB    PERSCI:WDDATA
88A8 86E6       818:          LDAA    #(\%00011001)&$FF       seek, load head, no verify
88AA 20C9       819:          BRA     PERSCI:ISSUECOMMAND
                820:
  88AC          821: PERSCI:VERIFYSECTOR
88AC A604       822:          LDAA    CCB:DRIVE,X             don't want either write or DMA!!
88AE B7FFA3     823:          STAA    PERSCI:PIADB
88B1 200E       824:          BRA     PERSCI:READSECTOR.2
```

```
  88B3          826: PERSCI:READSECTOR
88B3 F7FFA2     827:         STAB   PERSCI:PIADA           set DMA page number
88B6 E604       828:         LDAB   CCB:DRIVE,X            set up controller for read, DMA
88B8 CA40       829:         ORAB   #%01000000
88BA F7FFA3     830:         STAB   PERSCI:PIADB
88BD 43         831:         COMA
88BE B7FFA6     832:         STAA   PERSCI:WDSECTOR        set the sector number
  88C1          833: PERSCI:READSECTOR.2
88C1 BD887B     834:         JSR    PERSCI:ABORT           load head, if necessary
88C4 8677       835:         LDAA   #(\%10001000)&$FF      read sector
88C6 C520       836:         BITB   #%00100000             head load status
88C8 2602       837:         BNE    PERSCI:READSECTOR.1
88CA 8804       838:         EORA   #%00000100             make the head load
  88CC          839: PERSCI:READSECTOR.1
88CC 7E8875     840:         JMP    PERSCI:ISSUECOMMAND
                841:
  88CF          842: PERSCI:WRITESECTOR
88CF F7FFA2     843:         STAB   PERSCI:PIADA           set DMA page number
88D2 E604       844:         LDAB   CCB:DRIVE,X
88D4 CAC0       845:         ORAB   #%11000000             set up controller for write, DMA
88D6 F7FFA3     846:         STAB   PERSCI:PIADB
88D9 43         847:         COMA
88DA B7FFA6     848:         STAA   PERSCI:WDSECTOR        set the sector number
88DD BD887B     849:         JSR    PERSCI:ABORT           see if necessary to load heads
88E0 8657       850:         LDAA   #(\%10101000)&$FF      write sector
88E2 C520       851:         BITB   #%00100000
88E4 2602       852:         BNE    PERSCI:WRITESECTOR.1
88E6 8804       853:         EORA   #%00000100             load the heads
  88E8          854: PERSCI:WRITESECTOR.1
88E8 7E8875     855:         JMP    PERSCI:ISSUECOMMAND
                856:         FIN    PERSCI
   0002         857:         IF     DAMFLOPPY
```

IOVFD.ASM

```
  88EB          859: DAMFLOPPY:STATUS
88EB B6FF84     860:         LDAA   DAMFLOPPY:WDCMDSTS
88EE FE9013     861:         LDX    DISKINTDCB
88F1 39         862:         RTS
                863:
  88F2          864: DAMFLOPPY:RESTORE
88F2 A604       865:         LDAA   CCB:DRIVE,X
88F4 B7FF83     866:         STAA   DAMFLOPPY:PIADB
88F7 8602       867:         LDAA   #%00000010            restore
  88F9          868: DAMFLOPPY:ISSUECOMMAND
88F9 B7FF84     869:         STAA   DAMFLOPPY:WDCMDSTS
88FC 7E881B     870:         JMP    COUNTCOMMAND
                871:
  88FF          872: DAMFLOPPY:ABORT
88FF 86D0       873:         LDAA   #%11010000            abort with no interrupts
8901 B7FF84     874:         STAA   DAMFLOPPY:WDCMDSTS
8904 8D0D       875:         BSR    DAMFLOPPY:ABORT.RTS   wait about 30 uS for chip to settle
8906 8D0B       876:         BSR    DAMFLOPPY:ABORT.RTS
8908 8D09       877:         BSR    DAMFLOPPY:ABORT.RTS
890A F6FF84     878:         LDAB   DAMFLOPPY:WDCMDSTS    return with status in B
890D B6FF83     879:         LDAA   DAMFLOPPY:PIADB       clear possible interrupt
8910 FE9013     880:         LDX    DISKINTDCB
  8913          881: DAMFLOPPY:ABORT.RTS
8913 39         882:         RTS
                883:
  8914          884: DAMFLOPPY:RESET
8914 86D1       885:         LDAA   #%11010001            abort with interrupt
8916 20E1       886:         BRA    DAMFLOPPY:ISSUECOMMAND
```

```
  8918          888: DAMFLOPPY:SETSEEK
8918 7E8831     889:         JMP    TESTFORSEEK
                890:
  891B          891: DAMFLOPPY:SEEK
891B A604       892:         LDAA   CCB:DRIVE,X
891D B7FF83     893:         STAA   DAMFLOPPY:PIADB
8920 A606       894:         LDAA   CCB:LASTCYL,X
8922 E605       895:         LDAB   CCB:CYL,X
8924 B7FF85     896:         STAA   DAMFLOPPY:WDTRACK
8927 F7FF87     897:         STAB   DAMFLOPPY:WDDATA
892A 8619       898:         LDAA   #%00011001              seek, load head, 12 mS step, no verify
892C 20CB       899:         BRA    DAMFLOPPY:ISSUECOMMAND
                900:
  892E          901: DAMFLOPPY:VERIFYSECTOR
892E B6FF83     902:         LDAA   DAMFLOPPY:PIADB         turn off write, DMA
8931 842F       903:         ANDA   #%00101111
8933 B7FF83     904:         STAA   DAMFLOPPY:PIADB
8936 200D       905:         BRA    DAMFLOPPY:READSECTOR.2
```

```
  8938          907: DAMFLOPPY:READSECTOR
8938 F7FF82     908:         STAB    DAMFLOPPY:PIADA         set DMA page number
893B E604       909:         LDAB    CCB:DRIVE,X             set up controller for read, DMA
893D CA40       910:         ORAB    #%01000000              read, DMA
893F F7FF83     911:         STAB    DAMFLOPPY:PIADB
8942 B7FF86     912:         STAA    DAMFLOPPY:WDSECTOR      set the sector number
  8945          913: DAMFLOPPY:READSECTOR.2
8945 8DB8       914:         BSR     DAMFLOPPY:ABORT         load head, if necessary
8947 8680       915:         LDAA    #%10000000              read sector
8949 C520       916:         BITB    #%00100000              head load status
894B 2602       917:         BNE     DAMFLOPPY:READSECTOR.1
894D 8A04       918:         ORAA    #%00000100              make the head load
  894F          919: DAMFLOPPY:READSECTOR.1
894F 7E88F9     920:         JMP     DAMFLOPPY:ISSUECOMMAND
```

```
  8952          922: DAMFLOPPY:WRITESECTOR
8952 F7FF82     923:         STAB   DAMFLOPPY:PIADA         set DMA page number
8955 E604       924:         LDAB   CCB:DRIVE,X             set up controller for write, DMA
8957 CAC0       925:         ORAB   #%11000000
8959 B7FF86     926:         STAA   DAMFLOPPY:WDSECTOR      target sector
895C B6FF85     927:         LDAA   DAMFLOPPY:WDTRACK       check if write pre-compensation needed
895F 8115       928:         CMPA   #21
8961 2B02       929:         BMI    DAMFLOPPY:WRITESECTOR.2
8963 CA10       930:         ORAB   #%00010000              turn on write pre-compensation
  8965          931: DAMFLOPPY:WRITESECTOR.2
8965 F7FF83     932:         STAB   DAMFLOPPY:PIADB
8968 BD88FF     933:         JSR    DAMFLOPPY:ABORT         see if necessary to load heads
896B 86A0       934:         LDAA   #%10100000              write sector
896D C520       935:         BITB   #%00100000
896F 2602       936:         BNE    DAMFLOPPY:WRITESECTOR.1
8971 8A04       937:         ORAA   #%00000100              load the heads
  8973          938: DAMFLOPPY:WRITESECTOR.1
8973 7E88F9     939:         JMP    DAMFLOPPY:ISSUECOMMAND
                940:         FIN    DAMFLOPPY
```

IOVFD.ASM

```
  0002          942:          IF      PERSCI
  8976          943: PERSCI:TIMEOUT
8976 CE9017     944:          LDX     #CCB:PERSCI
  0002          945:          IF      DAMFLOPPY
8979 2003       946:          BRA     DISKTIMEOUT
                947:
                948:          FIN     DAMFLOPPY
                949:          FIN     PERSCI
  0002          950:          IF      DAMFLOPPY
  897B          951: DAMFLOPPY:TIMEOUT
897B CE9045     952:          LDX     #CCB:DAMFLOPPY
                953:          FIN     DAMFLOPPY
  897E          954: DISKTIMEOUT
897E FF9015     955:          STX     DISKINTCCB              save CCB address
8981 AD09       956:          JSR     CCB:STATUS,X            touch controller to keep drive going
8983 FE9015     957:          LDX     DISKINTCCB
8986 6A03       958:          DEC     CCB:TIMEOUT,X           COUNT OFF 1 SEC
8988 261F       959:          BNE     DISKTIMEOUT1            B/ TIMER NOT ZERO YET
898A EE2C       960:          LDX     CCB:CURRENTDCB,X        point at DCB, again
898C A600       961:          LDAA    DCB:DONEFLAG,X
898E 2631       962:          BNE     DISKTIMEOUT2            B/ DISK IS DONE, GO AWAY
  0002          963:          IF      DAMFLOPPY
                964: * should have code here to reset "load both heads" bit ??
                965:          FIN
8990 8604       966:          LDAA    #ERR:DEVICETIMEDOUT/256
8992 C612       967:          LDAB    #ERR:DEVICETIMEDOUT&$FF
  8994          968: DISKTIMEOUTERRORED ; timeout detected an error
8994 FF9013     969:          STX     DISKINTDCB              remember DCB address
8997 A701       970:          STAA    DCB:LASTERROR,X         remember device error code
8999 E702       971:          STAB    DCB:LASTERROR+1,X
899B 6C00       972:          INC     DCB:DONEFLAG,X          MARK DISK AS 'DONE'
899D FE9015     973:          LDX     DISKINTCCB              point at 'controller busy' flag
89A0 6C00       974:          INC     CCB:BUSY,X              and make it unbusy
89A2 6F04       975:          CLR     CCB:DRIVE,X             FORCE SEEK W/VERIFY ON NEXT READ/WRITE
89A4 6A04       976:          DEC     CCB:DRIVE,X
89A6 7E875E     977:          JMP     DISKINTUNEXPECTED
                978:
  89A9          979: DISKTIMEOUT1   ; (X) -> controller table
89A9 49         980:          ROLA                            save device ready status in carry
89AA EE2C       981:          LDX     CCB:CURRENTDCB,X        find DCB for device
89AC A600       982:          LDAA    DCB:DONEFLAG,X          is device done ?
89AE 2606       983:          BNE     DISKTIMEOUT1A           b/ yes, just keep it spinning
89B0 8604       984:          LDAA    #ERR:DEVICENOTREADY/256 assume the worst...
89B2 C624       985:          LDAB    #ERR:DEVICENOTREADY&$FF
89B4 25DE       986:          BCS     DISKTIMEOUTERRORED      b/ drive not ready after 1 second
  89B6          987: DISKTIMEOUT1A
89B6 FE9015     988:          LDX     DISKINTCCB
89B9 8600       989:          LDAA    #(1*TICKSPERSECOND+NTIMEOUTBLOCKS)/256
89BB C645       990:          LDAB    #(1*TICKSPERSECOND+NTIMEOUTBLOCKS)&$FF
89BD A726       991:          STAA    CCB:TIMEOUTBLK+TIMEOUT:FUSE,X    plant a 1-second fuse
89BF E727       992:          STAB    CCB:TIMEOUTBLK+TIMEOUT:FUSE+1,X
  89C1          993: DISKTIMEOUT2
89C1 7EBE15     994:          JMP     SDOS+SDOS:RTI
                995:          FIN     IODRIVERBODY
                996:
```

```
                997:
                411:            FIN
  0001          412:            IF      STORAGEDEMON
                413:            INCLUDE                         IOSTOREDEMON.ASM
  0001            1:                    IF      IODRIVERBODY
```

```
                3: *                    DRIVES IMI7710 WITH 7711 INTELLIGENT CONTROLLER...
                4: *                    VIA A "VIA" (A WONDERFUL ROCKWELL PART)
                5: *
                6: ; EQUATES FOR WINCHESTER DRIVER
                7:
  0200          8: WDCNBPS            EQU     512             ; 512 BYTES PER SECTOR (TRANSFER)
                9:
  0001         10:                    IFUND   IMI7711
  0000         11: IMI7711            EQU     0
               12:                    FIN
  0000         13:                    IF      IMI7711
               15:                    FIN
               16:
  0001         17:                    IFUND   IMI5007
  0000         18: IMI5007            EQU     0
               19:                    FIN
  0000         20:                    IF      IMI5007         mini-wini
               22:                    FIN
               23:
  0001         24:                    IFUND   WDCNSPT
  4E34         25: WDCNSPT            EQU     20020           DEFAULT TO 7710C
               26:                    FIN
  0001         27: WDCNTPC            EQU     1
  0001         28: WDCNCYL            EQU     1
  0080         29: WDCFATAL           EQU     $80             ; RETRY TYPE ERROR
               30:
  0001         31: WDCFORMAT          EQU     1               ; FORMAT ENTIRE DISK COMMAND
  0002         32: WDCREADCMD         EQU     2               ; CONTROL READ COMMAND
  0003         33: WDCWRITECMD        EQU     3               ; CONTROL WRITE COMMAND
               34:
  0005         35: WDCRETRY           EQU     5               ; FAILURE RETRY COUNT
               36:
               37:
               38: *                   WINCHESTER DISK CONTROLLER DCB DEFINITIONS
               39: *
  89C4         40: ::                 SET     *
  0042         41:                    ORG     DSKINFO:SIZE    TACKS ON TO BOTTOM OF DISK INFO TABLE
0042 0001      42: WDCREADWRITE       RMB     1               0 IS READ, <>0 IS WRITE
0043 0001      43: WDCDRIVE           RMB     1               DRIVE NUMBER
  0044         44: WDCSIZE            EQU     *
  89C4         45:                    ORG     ::
```

```
                47: ; BRANCH TABLE POINTED TO BY DCB AND USED BY SDOS
                48:
89C4 9E2C       49: WDCDRIVER       FDB   WDCINIT        ; ROUTINE TO CLEAR VIA SO NO INTERRUPTS
89C6 89E9       50:                 FDB   WDCREAD        ; READ SINGLE SECTOR
89C8 89E5       51:                 FDB   WDCWRITE       ; WRITE SINGLE SECTOR
89CA 8A22       52:                 FDB   WDCWAITDONE
89CC 8EDB       53:                 FDB   ILLDEVICEOP    SDOS HANDLES ALL THE DISK STATUSES NECESSARY !
89CE 89D0       54:                 FDB   WDCCONTROL     ; DISMOUNT OR FORMAT COMMAND
                55:
  89D0          56: WDCCONTROL      ;CMPA  #CC:DISMOUNTDISK
89D0 8111       57:                 CMPA  #CC:DISMOUNTDISK
89D2 274C       58:                 BEQ   WDCOKRTS       B/ DISMOUNT, NOTHING SPECIAL NEEDED.
                59:                 ;CMPA  #CC:FORMAT
89D4 8115       60:                 CMPA  #CC:FORMAT
89D6 2703       61:                 BEQ   WDCFORMATX     B/ FORMAT OPERATION
                62:                 ;JMP   ILLDEVICEOP
89D8 7E8EDB     63:                 JMP   ILLDEVICEOP
  89DB          64: WDCFORMATX      ;LDA   #WDCFORMAT     DO A "SECONDARY" FORMAT OPERATION
89DB 8601       65:                 LDAA  #WDCFORMAT
                66:                 ;LDX   DCBPOINTER
89DD DE06       67:                 LDX   DCBPOINTER
                68:                 ;STA   WDCREADWRITE,X
89DF A742       69:                 STAA  WDCREADWRITE,X
                70:                 ;LDA   #1             ; SET RETRY COUNT AT 1
89E1 8601       71:                 LDAA  #1
89E3 200C       72:                 BRA   WDCSETRETRY1
                73:
  89E5          74: WDCWRITE        ;LDA   #WDCWRITECMD
89E5 8603       75:                 LDAA  #WDCWRITECMD
89E7 2002       76:                 BRA   WDCOPSET
                77:
  89E9          78: WDCREAD         ;LDA   #WDCREADCMD
89E9 8602       79:                 LDAA  #WDCREADCMD
  89EB          80: WDCOPSET        ;LDX   DCBPOINTER
89EB DE06       81:                 LDX   DCBPOINTER
                82:                 ;STA   WDCREADWRITE,X  ; SET THE OPERATION
89ED A742       83:                 STAA  WDCREADWRITE,X
                84:                 ;LDA   #WDCRETRY       ; SET RETRY COUNT
89EF 8605       85:                 LDAA  #WDCRETRY
  89F1          86: WDCSETRETRY1 ; ENTRY POINT FOR WDCFORMATX
                87:                 ;STA   WDCRETRYCNT
89F1 B7932D     88:                 STAA  WDCRETRYCNT
                89:                 ;CLR   DCB:LASTERROR,X
89F4 6F01       90:                 CLR   DCB:LASTERROR,X
                91:                 ;CLR   DCB:LASTERROR+1,X
89F6 6F02       92:                 CLR   DCB:LASTERROR+1,X
                93:                 ;LDX   #WDCINTERFACE   ; WAIT FOR INTERFACE FREE
89F8 CE9325     94:                 LDX   #WDCINTERFACE
                95:                 ;LDA   0,X
89FB A600       96:                 LDAA  0,X
89FD 2603       97:                 BNE   WDCSETUP
                98:                 ;JSR   SDOS+SDOS:WAITEVENT
89FF BDBE2A     99:                 JSR   SDOS+SDOS:WAITEVENT
  8A02         100: WDCSETUP        ;CLR   WDCINTERFACE    ; SET INTERFACE BUSY
8A02 7F9325    101:                 CLR   WDCINTERFACE
```

```
                    *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
                    *** THE SD STORAGE DEMON DRIVER ***

                    ;LDX    DCBPOINTER
8A05 DE06           LDX     DCBPOINTER
8A07 FF9326         STX     WDCDCBPOINTER   ; INTERRUPTS SERVICE DCB ADDRESS
                    ;CLR    DCB:DONEFLAG,X  ; CLEAR DONE
8A0A 6F00           CLR     DCB:DONEFLAG,X
                    ;LDX    #0
8A0C CE0000         LDX     #0
8A0F FF9328         STX     WDCCONTINUEPC   ; SET INTERRUPTS NO GOOD
                    ;LDX    #WDCSTARTIO
8A12 CE8A5C         LDX     #WDCSTARTIO
                    ;JSR    SDOS+SDOS:STARTIO ; ENTER INTERRUPTS SERVICE CODE
8A15 BDBE24         JSR     SDOS+SDOS:STARTIO
                    ;LDX    DCBPOINTER      ; IF FORMAT CONTROL CALL OPERATION
8A18 DE06           LDX     DCBPOINTER
                    ;LDA    WDCREADWRITE,X
8A1A A642           LDAA    WDCREADWRITE,X
                    ;CMPA   #WDCFORMAT
8A1C 8101           CMPA    #WDCFORMAT
8A1E 2702           BEQ     WDCWAITDONE     B/ GO WAIT FOR FORMAT OPERATION COMPLETE
8A20 0C39           WDCOKRTS OKRTS

  8A22              WDCWAITDONE ;LDX  DCBPOINTER      ; WAIT FOR TRANSFER DONE
8A22 DE06           LDX     DCBPOINTER
                    ;LDA    DCB:DONEFLAG,X
8A24 A600           LDAA    DCB:DONEFLAG,X
8A26 2603           BNE     WDCWAIT1
                    ;JSR    SDOS+SDOS:WAITEVENT
8A28 BDBE2A         JSR     SDOS+SDOS:WAITEVENT
  8A2B              WDCWAIT1 ;LDX   DCBPOINTER
8A2B DE06           LDX     DCBPOINTER
                    ;LDX    DCB:LASTERROR,X
8A2D EE01           LDX     DCB:LASTERROR,X
8A2F 27EF           BEQ     WDCOKRTS
                    ;JMP    ERRETX
8A31 7E8EE0         JMP     ERRETX
8A34 0C39           OKRTS
```

```
                139: * VIA REGISTER DEFINITIONS
  0001          140:                    IFUND   CONRAC
  0000          141: CONRAC             EQU     0
                142:                    FIN
  0000          143:                    IF      CONRAC
                145:                    FIN
  0000          146:                    IFUND   WAVEMATE
                148:                    FIN
  0001          149:                    IF      WAVEMATE
  FF40          150: STORAGEDEMONVIA    EQU     $FF40
                151:                    FIN
  0001          152:                    IFUND   EXORCISOR
  0000          153: EXORCISOR          EQU     0
                154:                    FIN
  0000          155:                    IF      EXORCISOR
                157:                    FIN
                158:
  0001          159:                    IF      CONRAC!WAVEMATE!EXORCISOR
  FF4C          160: VIAPCR             EQU     STORAGEDEMONVIA+$C ; CONTROL REGISTER
  FF4D          161: VIAIFR             EQU     VIAPCR+1        ; INTERRUPT FLAG
  FF4E          162: VIAIER             EQU     VIAIFR+1        ; INTERRUPT ENABLE
                163:
  FF40          164: VIADRB             EQU     STORAGEDEMONVIA+$0 ; DATA REGISTERS
  FF41          165: VIADRA             EQU     VIADRB+1
  FF42          166: VIADDRB            EQU     VIADRA+1
  FF43          167: VIADDRA            EQU     VIADDRB+1       ; DATA DIRECTION REGISTERS
  FF44          168: VIAT1LL            EQU     VIADDRA+1
  FF45          169: VIAT1CH            EQU     VIAT1LL+1       ; INTERVAL TIMER HIGH BYTE
  FF46          170: VIAT1LLA           EQU     VIAT1CH+1
  FF47          171: VIAT1LH            EQU     VIAT1LLA+1
                172:
  FF4B          173: VIAACR             EQU     STORAGEDEMONVIA+$B ; AUXILIARY CONTROL REGISTER - USED FOR CLOCK
                174: *
  FF4F          175: VIADRAF            EQU     STORAGEDEMONVIA+$F ; PORT A - NO HANDSHAKE CONTROL
                176:
                177:                    FIN
  0001          178:                    IFUND   WMSERIES2000
  0000          179: WMSERIES2000       EQU     0
                180:                    FIN
  0000          181:                    IF      WMSERIES2000    WITH ITS INVERTED I/O ADDRESS LINES (YUK!)
                203:                    FIN     WMSERIES2000
```

```
MAL/6800 1.3F: 8A34  SDOSDRIVERS       *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 49; Form 1     *** THE SD STORAGE DEMON DRIVER ***
IOSTOREDEMON.ASM
  8A36          205: WDCRESET          ;CLR    VIADDRA         MAKE PA0 SIDE OF VIA BE AN "INPUT" PORT
8A36 7FFF43     206:                   CLR     VIADDRA
                207:                   ;LDA    #%11011010      ; WATCH FOR READY AND *BUSDIR= +
8A39 86DA       208:                   LDAA    #%11011010
                209:                   ;STA    VIAPCR          ; VIA PROG CONTROL REG
8A3B B7FF4C     210:                   STAA    VIAPCR
                211:                   ;LDA    VIADRA          ; ISSUE STROBE SD READY PULSE CAN BE SEEN
8A3E B6FF41     212:                   LDAA    VIADRA
                213:                   ;LDX    #(20000*2)//8   ; PULSE CB2 FOR 20Ms. AS PER 7711 RESET SPECIFICATION
8A41 CE1388     214:                   LDX     #(20000*2)//8
8A44 09         215: WDCRESETLP        DEX                     ; WAIT LONG ENOUGH FOR PULSE...
8A45 26FD       216:                   BNE     WDCRESETLP
                217:                   ;LDA    #%00010010      ; RESET INTERRUPT BITS
8A47 8612       218:                   LDAA    #%00010010
                219:                   ;STA    VIAIFR          ; ACK INTERRUPTS THAT MIGHT ACCIDENTALLY BE PENDING
8A49 B7FF4D     220:                   STAA    VIAIFR          (assume IOAI response can't come back this fast!)
                221:                   ;STA    VIAIER          ; AND CLEAR THE INTERRUPT ENABLE
8A4C B7FF4E     222:                   STAA    VIAIER
                223:                   ;LDA    #%11111010
8A4F 86FA       224:                   LDAA    #%11111010
                225:                   ;STA    VIAPCR          ; STOP PULSE, LEAVE CA2 IN "PULSE ON R/W"
8A51 B7FF4C     226:                   STAA    VIAPCR
8A54 0C39       227:                   OKRTS
```

Ø

GOING HIGH

GOING HIGH

Ø

pcr 4 = 0 → watch for CB1 ↓

= 1 → watch for CB1 ↑

```
                229: ; FEED THE WINCHESTER A COMMAND
                230:
  8A56          231: WDCFORMSERVJ       ;JMP    WDCFORMSERV     ; GO HANDLE "FORMAT" COMMAND
8A56 7E8B3B     232:                    JMP     WDCFORMSERV
                233:
  8A59          234: WDCREADSERVJ       ;JMP    WDCREADSERV     ; GO DO READ SECTOR LOGIC
8A59 7E8B4F     235:                    JMP     WDCREADSERV
                236:
  8A5C          237: WDCSTARTIO         EQU     *               CONTROL TRANSFERS HERE TO START DISK I/O
8A5C 0E         238: WDCCMDFEED         CLI                     ; RE-ENABLE INTERRUPTS
                239:                    ;LDX    WDCDCBPOINTER
8A5D FE9326     240:                    LDX     WDCDCBPOINTER
8A60 6C3E       241:                    INC     DSKINFO:OPSCOUNT+2,X
8A62 2606       242:                    BNE     WDCCMDFEED0
8A64 6C3D       243:                    INC     DSKINFO:OPSCOUNT+1,X
8A66 2602       244:                    BNE     WDCCMDFEED0
8A68 6C3C       245:                    INC     DSKINFO:OPSCOUNT,X
  8A6A          246: WDCCMDFEED0        ;JSR    WDCWAITAVAILABLE ; WAIT FOR COMMAND AND DATA BUS AVAILABLE
8A6A BD8C1E     247:                    JSR     WDCWAITAVAILABLE
                248:                    ;LDX    WDCDCBPOINTER
8A6D FE9326     249:                    LDX     WDCDCBPOINTER
                250:                    ;LDA    WDCREADWRITE,X  ; COMMAND TYPE
8A70 A642       251:                    LDAA    WDCREADWRITE,X
                252:                    ;JSR    WDCOUTDATA      ; OUTPUT COMMAND BYTE
8A72 BD8BE0     253:                    JSR     WDCOUTDATA
                254:                    ;LDA    WDCDRIVE,X      ; DRIVE SELECT
8A75 A643       255:                    LDAA    WDCDRIVE,X
                256:                    ;JSR    WDCOUTDATA
8A77 BD8BE0     257:                    JSR     WDCOUTDATA
                258:                    ;LDA    WDCREADWRITE,X  ; CHECK IF FORMAT COMMAND
8A7A A642       259:                    LDAA    WDCREADWRITE,X
                260:                    ;CMPA   #WDCFORMAT
8A7C 8101       261:                    CMPA    #WDCFORMAT
8A7E 27D6       262:                    BEQ     WDCFORMSERVJ    ; B/ ALL FORMAT PARAMETERS SENT!
  8A80          263: WDCCMDFEED1        ;LDX    DSKINFO:SECTORDB,X
8A80 EE2B       264:                    LDX     DSKINFO:SECTORDB,X
                265:                    ;LDA    RDSI:LSN+2,X    ; FEED LOW BYTE OF
8A82 A604       266:                    LDAA    RDSI:LSN+2,X
                267:                    ;JSR    WDCOUTDATA      ; DISK ADDRESS OUT
8A84 BD8BE0     268:                    JSR     WDCOUTDATA
                269:                    ;LDA    RDSI:LSN+1,X    ; AND HIGH BYTE LOGICAL DISK ADDRESS
8A87 A603       270:                    LDAA    RDSI:LSN+1,X
                271:                    ;JSR    WDCOUTDATA
8A89 BD8BE0     272:                    JSR     WDCOUTDATA
                273:                    ;LDX    WDCDCBPOINTER
8A8C FE9326     274:                    LDX     WDCDCBPOINTER
                275:                    ;LDA    WDCREADWRITE,X
8A8F A642       276:                    LDAA    WDCREADWRITE,X
                277:                    ;CMPA   #WDCREADCMD
8A91 8102       278:                    CMPA    #WDCREADCMD
8A93 27C4       279:                    BEQ     WDCREADSERVJ    ; GO DO READ TRANSFER AND CHECK
```

```
                281: ; WRITE TRANSFER SECTION
                282:
  8A95          283: WDCWRITESERV       ; WRITE A SECTOR TO 7710
                284:                    ;JSR    WDCSET4TRANS    ; SET X = PAGE ADR
8A95 BD8D01     285:                    JSR     WDCSET4TRANS
                286:                    ;LDA    #$FF            ; SET VIA TO OUTPUT MODE (AND SET UP 255 CYCLE COUNTER)
8A98 86FF       287:                    LDAA    #$FF
                288:                    ;STA    VIADDRA         ; SO AS TO OUTPUT A 512 BYTE SECTOR
8A9A B7FF43     289:                    STAA    VIADDRA
  8A9D          290: WDCWRITEWAIT1ST    ; WAIT FOR 1ST DATA REQUEST
8A9D F5FF4D     291:                    BITB    VIAIFR          FIRST DATA REQUEST ARRIVE ?
8AA0 2608       292:                    BNE     WDCWRITELOOP    B/ YES, GIVE THE 7710 ITS DATA
8AA2 08         293:                    INX                     NO, DELAY AWHILE (???? Us. MAX)
8AA3 09         294:                    DEX
8AA4 4A         295:                    DECA                    DOWN COUNT FUSE
8AA5 26F6       296:                    BNE     WDCWRITEWAIT1ST B/ MORE TIME TO WAIT
                297:                    ;JMP    WDCQUIET1       FUSE EXPIRED, SO DO WE!
8AA7 7E8B31     298:                    JMP     WDCQUIET1
                299:
  8AAA          300: WDCWRITELOOP       ; OUTPUT BYTES LOOP, OPTIMIZED FOR SPEED!
                301:                    ;LDX    WDCPOINTER      ; GET POINTER TO NEXT BLOCK OF 8 BYTES
8AAA FE932B     302:                    LDX     WDCPOINTER
                303:                    ;LDA    ,X              ; FETCH BYTE TO FEED TO CONTROLLER
8AAD A600       304:                    LDAA    0,X
8AAF F5FF4D     305:                    BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8AB2 2602       306:                    BNE     WDCWRITE0       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AB4 8D6E       307:                    BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8AB6          308: WDCWRITE0          ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8AB6 B7FF41     309:                    STAA    VIADRA
                310:                    ;LDA    1,X             ; FETCH BYTE TO FEED TO CONTROLLER
8AB9 A601       311:                    LDAA    1,X
8ABB F5FF4D     312:                    BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8ABE 2602       313:                    BNE     WDCWRITE1       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AC0 8D62       314:                    BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8AC2          315: WDCWRITE1          ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8AC2 B7FF41     316:                    STAA    VIADRA
                317:                    ;LDA    2,X             ; FETCH BYTE TO FEED TO CONTROLLER
8AC5 A602       318:                    LDAA    2,X
8AC7 F5FF4D     319:                    BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8ACA 2602       320:                    BNE     WDCWRITE2       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8ACC 8D56       321:                    BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8ACE          322: WDCWRITE2          ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8ACE B7FF41     323:                    STAA    VIADRA
                324:                    ;LDA    3,X             ; FETCH BYTE TO FEED TO CONTROLLER
8AD1 A603       325:                    LDAA    3,X
8AD3 F5FF4D     326:                    BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8AD6 2602       327:                    BNE     WDCWRITE3       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AD8 8D4A       328:                    BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8ADA          329: WDCWRITE3          ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8ADA B7FF41     330:                    STAA    VIADRA
                331:                    ;LDA    4,X             ; FETCH BYTE TO FEED TO CONTROLLER
8ADD A604       332:                    LDAA    4,X
8ADF F5FF4D     333:                    BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8AE2 2602       334:                    BNE     WDCWRITE4       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AE4 8D3E       335:                    BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
```

```
                        *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
                        *** THE SD STORAGE DEMON DRIVER ***

  8AE6          336: WDCWRITE4    ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8AE6 B7FF41     337:              STAA    VIADRA
                338:              ;LDA    5,X             ; FETCH BYTE TO FEED TO CONTROLLER
8AE9 A605       339:              LDAA    5,X
8AEB F5FF4D     340:              BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8AEE 2602       341:              BNE     WDCWRITE5       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AF0 8D32       342:              BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8AF2          343: WDCWRITE5    ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8AF2 B7FF41     344:              STAA    VIADRA
                345:              ;LDA    6,X             ; FETCH BYTE TO FEED TO CONTROLLER
8AF5 A606       346:              LDAA    6,X
8AF7 F5FF4D     347:              BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8AFA 2602       348:              BNE     WDCWRITE6       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8AFC 8D26       349:              BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8AFE          350: WDCWRITE6    ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8AFE B7FF41     351:              STAA    VIADRA
                352:              ;LDA    7,X             ; FETCH BYTE TO FEED TO CONTROLLER
8B01 A607       353:              LDAA    7,X
8B03 F5FF4D     354:              BITB    VIAIFR          ; IS 7710 READY FOR NEXT BYTE ?
8B06 2602       355:              BNE     WDCWRITE7       B/ USUAL CASE, 7710 IS READY FOR ANOTHER
8B08 8D1A       356:              BSR     WDCWRITEWAIT    SIGH... GO WAIT FOR 7710 TO BE READY
  8B0A          357: WDCWRITE7    ;STA    VIADRA          ; OUTPUT DATA BYTE AND ISSUE STROBE
8B0A B7FF41     358:              STAA    VIADRA
                359:              ;LDA    WDCPOINTER+1    ADVANCE POINTER BY 8 BYTES
8B0D B6932C     360:              LDAA    WDCPOINTER+1
8B10 8B08       361:              ADDA    #8
                362:              ;STA    WDCPOINTER+1
8B12 B7932C     363:              STAA    WDCPOINTER+1
8B15 2403       364:              BCC     WDCWRITED       B/ UPPER HALF DOES NOT NEED MODIFICATION
8B17 7C932B     365:              INC     WDCPOINTER      PROPAGATE CARRY TO UPPER HALF
8B1A 7A932A     366: WDCWRITED    DEC     WDCCOUNT        DOWN COUNT NUMBER OF 8 BYTE BLOCKS TO SEND
8B1D 268B       367:              BNE     WDCWRITELOOP    B/ MORE 8 BYTE BLOCKS TO WRITE
                368:              ;CLR    VIADDRA         ; MAKE VIA PORT AN INPUT PORT WHEN DONE
8B1F 7FFF43     369:              CLR     VIADDRA
8B22 2017       370:              BRA     WDCFORMSERV     GO WAIT FOR 7710 TO FINISH OPERATION
```

```
MAL/6800 1.3F: 8B22  SDOSDRIVERS        *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 53; Form 1      *** THE SD STORAGE DEMON DRIVER ***
IOSTOREDEMON.ASM
  8B24          372: WDCWRITEWAIT       ; WAIT FOR 7710 TO BE READY FOR NEXT BYTE
8B24 36         373:                    PSHA                    ; SAVE THE DATA BYTE TO SEND
8B25 4F         374:                    CLRA                    ; SET TIMEOUT LIMIT IN (A)
8B26 F5FF4D     375: WDCWRITEWAITLOOP   BITB    VIAIFR          ; LOOK AGAIN
8B29 260E       376:                    BNE     WDCWRITEWAITEXIT ; B/ FINALLY, IS READY!
8B2B 4A         377:                    DECA                    DOWN COUNT FUSE
8B2C 26F8       378:                    BNE     WDCWRITEWAITLOOP B/ SOME FUSE STILL LEFT
8B2E 31         379:                    INS                     BANG! TIME'S UP...POP DATA BYTE TO BE SENT
  8B2F          380: WDCQUIETERR        ; 7710 DID NOT RESPOND IN REASONABLE LENGTH OF TIME
                381:                    ;LEAS   2,S             POP RETURN ADDRESS
  0000          382:                    IF      2<0
                385:                    ELSE
  0002          386:                    RPT     2
8B2F 31         387:                    INS
                388:                    FIN
  8B31          389: WDCQUIET1          ; IMI DRIVE DID NOT RESPOND IN REASONABLE TIME
                390:                    ;JSR    WDCRESET        ; MAYBE HITTING BELOW THE BELT WILL RE-SYNCH
8B31 BD8A36     391:                    JSR     WDCRESET
                392:                    ;LDA    #%1001111       ; PICK UP VERY FUNNY ERROR STATUS
8B34 864F       393:                    LDAA    #%1001111
                394:                    ;JMP    WDCFATAL0       ; GO STORE ERROR AND RETRY
8B36 7E8C8E     395:                    JMP     WDCFATAL0
                396:
8B39 32         397: WDCWRITEWAITEXIT   PULA                    ; GET THE DATA BYTE BACK
8B3A 39         398:                    RTS
                399:
  8B3B          400: WDCFORMSERV        EQU     *
                401:                    ;JSR    WDCWAIT4INT     ; GO START INTERRUPT FOR COMMAND DONE
8B3B BD8C3C     402:                    JSR     WDCWAIT4INT
                403:                    ;JSR    WDCINDATA       ; CHECK DONE STATUS
8B3E BD8C07     404:                    JSR     WDCINDATA
                405:                    ;JSR    WDCPROCST       ; GO CHECK STATUS RETURN IF OK
8B41 BD8C84     406:                    JSR     WDCPROCST
  8B44          407: WDCDONE            EQU     *
                408:                    ;LDX    WDCDCBPOINTER
8B44 FE9326     409:                    LDX     WDCDCBPOINTER
8B47 6C00       410:                    INC     DCB:DONEFLAG,X  ; SET DONE
8B49 7C9325     411:                    INC     WDCINTERFACE    ; INTERFACE DONE
                412:                    ;JMP    SDOS+SDOS:RESCHEDULE
8B4C 7EBE18     413:                    JMP     SDOS+SDOS:RESCHEDULE
```

Can't optimize xfer for 7211


```
                415: ; READ TRANSFER SECTION
                416:
  8B4F          417: WDCREADSERV      ;JSR    WDCWAIT4INT     ; WAIT FOR 7710 INTERRUPT ON READ COMPLETE
8B4F BD8C3C     418:                  JSR     WDCWAIT4INT
                419:                  ;JSR    WDCINDATA       ; GET STATUS ON BUS
8B52 BD8C07     420:                  JSR     WDCINDATA
                421:                  ;JSR    WDCPROCST       ; GO PROCESS STATUS RETRY IF NEEDED
8B55 BD8C84     422:                  JSR     WDCPROCST
                423:                  ;JSR    WDCSET4TRANS
8B58 BD8D01     424:                  JSR     WDCSET4TRANS
  8B5B          425: WDCREADLOOP      ; INPUT BYTES FOR SECTOR LOOP, OPTIMIZED FOR SPEED
                426:                  ;LDX    WDCPOINTER      ; GET POINTER TO NEXT BLOCK OF 8 BYTES
8B5B FE932B     427:                  LDX     WDCPOINTER
8B5E F5FF4D     428:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B61 2602       429:                  BNE     WDCREAD0        B/ DATA IS READY
8B63 8D6E       430:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8B65          431: WDCREAD0         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8B65 B6FF41     432:                  LDAA    VIADRA
                433:                  ;STA    ,X              ; SAVE DATA IN SECTOR BUFFER
8B68 A700       434:                  STAA    0,X
8B6A F5FF4D     435:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B6D 2602       436:                  BNE     WDCREAD1        B/ DATA IS READY
8B6F 8D62       437:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8B71          438: WDCREAD1         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8B71 B6FF41     439:                  LDAA    VIADRA
                440:                  ;STA    1,X             ; SAVE DATA IN SECTOR BUFFER
8B74 A701       441:                  STAA    1,X
8B76 F5FF4D     442:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B79 2602       443:                  BNE     WDCREAD2        B/ DATA IS READY
8B7B 8D56       444:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8B7D          445: WDCREAD2         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8B7D B6FF41     446:                  LDAA    VIADRA
                447:                  ;STA    2,X             ; SAVE DATA IN SECTOR BUFFER
8B80 A702       448:                  STAA    2,X
8B82 F5FF4D     449:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B85 2602       450:                  BNE     WDCREAD3        B/ DATA IS READY
8B87 8D4A       451:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8B89          452: WDCREAD3         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8B89 B6FF41     453:                  LDAA    VIADRA
                454:                  ;STA    3,X             ; SAVE DATA IN SECTOR BUFFER
8B8C A703       455:                  STAA    3,X
8B8E F5FF4D     456:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B91 2602       457:                  BNE     WDCREAD4        B/ DATA IS READY
8B93 8D3E       458:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8B95          459: WDCREAD4         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8B95 B6FF41     460:                  LDAA    VIADRA
                461:                  ;STA    4,X             ; SAVE DATA IN SECTOR BUFFER
8B98 A704       462:                  STAA    4,X
8B9A F5FF4D     463:                  BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8B9D 2602       464:                  BNE     WDCREAD5        B/ DATA IS READY
8B9F 8D32       465:                  BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8BA1          466: WDCREAD5         ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8BA1 B6FF41     467:                  LDAA    VIADRA
                468:                  ;STA    5,X             ; SAVE DATA IN SECTOR BUFFER
8BA4 A705       469:                  STAA    5,X
```

```
8BA6 F5FF4D     470:                    BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8BA9 2602       471:                    BNE     WDCREAD6        B/ DATA IS READY
8BAB 8D26       472:                    BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8BAD          473: WDCREAD6           ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8BAD B6FF41     474:                    LDAA    VIADRA
                475:                    ;STA    6,X             ; SAVE DATA IN SECTOR BUFFER
8BB0 A706       476:                    STAA    6,X
8BB2 F5FF4D     477:                    BITB    VIAIFR          ; IS ANOTHER DATA BYTE READY ?
8BB5 2602       478:                    BNE     WDCREAD7        B/ DATA IS READY
8BB7 8D1A       479:                    BSR     WDCREADWAIT     ; GO WAIT FOR 7710 READY WITH ANOTHER BYTE
  8BB9          480: WDCREAD7           ;LDA    VIADRA          ; READ DATA AND ISSUE ACKNOWLEDGE PULSE
8BB9 B6FF41     481:                    LDAA    VIADRA
                482:                    ;STA    7,X             ; SAVE DATA IN SECTOR BUFFER
8BBC A707       483:                    STAA    7,X
                484:                    ;LDA    WDCPOINTER+1    ; ADVANCE BUFFER POINTER BY 8
8BBE B6932C     485:                    LDAA    WDCPOINTER+1
8BC1 8B08       486:                    ADDA    #8
                487:                    ;STA    WDCPOINTER+1
8BC3 B7932C     488:                    STAA    WDCPOINTER+1
8BC6 2403       489:                    BCC     WDCREADD        B/ DON'T HAVE TO BUMP UPPER HALF
8BC8 7C932B     490:                    INC     WDCPOINTER
8BCB 7A932A     491: WDCREADD           DEC     WDCCOUNT        DOWN COUNT # OF 8 BYTE BLOCKS TO SEND
8BCE 268B       492:                    BNE     WDCREADLOOP
                493:                    ;JMP    WDCDONE         ALL DONE READING SECTOR !
8BD0 7E8B44     494:                    JMP     WDCDONE
                495:
  8BD3          496: WDCREADWAIT        ; WAIT FOR 7710 TO BE READY TO GIVE US NEXT BYTE
8BD3 4F         497:                    CLRA
8BD4 F5FF4D     498: WDCREADWAITLOOP    BITB    VIAIFR          ; WAIT FOR READY SIGNAL
8BD7 2606       499:                    BNE     WDCREADWAITRTS  B/ 7710 IS NOW READY
8BD9 4A         500:                    DECA                    TIMED OUT ?
8BDA 26F8       501:                    BNE     WDCREADWAITLOOP B/ NOT YET
                502:                    ;JMP    WDCQUIETERR     TIMED OUT, SOMETHING'S WRONG!
8BDC 7E8B2F     503:                    JMP     WDCQUIETERR
                504:
8BDF 39         505: WDCREADWAITRTS     RTS
```

```
                    507: *              WDCOUTDATA --- SEND (A) TO 7710 WHEN 7710 READY
                    508: *                     USED ONLY FOR PARAMETER BYTES
                    509: *
  8BE0              510: WDCOUTDATA     EQU   *
                    511:                ;LDB  #$FF          ; SELECT VIA MODE = OUTPUT
8BE0 C6FF           512:                LDAB  #$FF
                    513:                ;STB  VIADDRA
8BE2 F7FF43         514:                STAB  VIADDRA
                    515:                ;STA  VIADRAF       ; OUTPUT THE DATA, DON'T ISSUE PULSE (YET)
8BE5 B7FF4F         516:                STAA  VIADRAF
8BE8 4F             517:                CLRA                ; SET A LONG FUSE
  8BE9              518: WDCOUTDATAL    ;LDB  VIAIFR        ; IS 7710 READY?
8BE9 F6FF4D         519:                LDAB  VIAIFR
8BEC C502           520:                BITB  #%00000010
8BEE 2606           521:                BNE   WDCOUTDATA1   ; YES
8BF0 4A             522:                DECA                ; NO, DOWN COUNT FUSE
8BF1 26F6           523:                BNE   WDCOUTDATAL
                    524:                ;JMP  WDCQUIETERR
8BF3 7E8B2F         525:                JMP   WDCQUIETERR
                    526:
  8BF6              527: WDCOUTDATA1    EQU   *
                    528:                ;LDB  #%11111010    ; WATCH FOR BUSDIR GOING LOW
8BF6 C6FA           529:                LDAB  #%11111010
                    530:                ;STB  VIAPCR
8BF8 F7FF4C         531:                STAB  VIAPCR
                    532:                ;LDB  #%00010010
8BFB C612           533:                LDAB  #%00010010
                    534:                ;STB  VIAIFR        ; SEE WDCWRITE1 FOR COMMENTS
8BFD F7FF4D         535:                STAB  VIAIFR
                    536:                ;LDA  VIADRA        ; ISSUE STROBE PULSE TO 7710
8C00 B6FF41         537:                LDAA  VIADRA
                    538:                ;CLR  VIADDRA       ; RESTORE VIA PORT TO INPUT MODE FOR SAFETY
8C03 7FFF43         539:                CLR   VIADDRA
8C06 39             540:                RTS
```

```
                542: *                  WDCINDATA --- GET (A) FROM 7710 WHEN 7710 IS READY
                543: *                  USED ONLY TO READ STATUS FROM 7710
                544: *                  ASSERT: VIADDRA=0 HERE
                545:
  8C07          546: WDCINDATA          EQU     *
8C07 4F         547:                    CLRA                    ; SET LONG FUSE
  8C08          548: WDCINDATA0         ;LDB    VIAIFR          ; IS 7710 READY WITH DATA?
8C08 F6FF4D     549:                    LDAB    VIAIFR
8C0B C502       550:                    BITB    #%00000010
8C0D 2606       551:                    BNE     WDCINDATA1
8C0F 4A         552:                    DECA
8C10 26F6       553:                    BNE     WDCINDATA0      ; B/ SOME MORE TIME LEFT
                554:                    ;JMP    WDCQUIETERR
8C12 7E8B2F     555:                    JMP     WDCQUIETERR
                556:
  8C15          557: WDCINDATA1         EQU     *               ; 7710 IS READY WITH DATA FOR US
                558:                    ;LDB    #%11101010      ; WATCH FOR IFACTIVE GOING HIGH
8C15 C6EA       559:                    LDAB    #%11101010
                560:                    ;STB    VIAPCR
8C17 F7FF4C     561:                    STAB    VIAPCR
  0000          562:                    IF      CONRAC          ??????? WHY ??????
                565:                    FIN     CONRAC
                566:                    ;LDA    VIADRA          ; GET STATUS, ACKNOWLEDGE 'READY' SIGNAL
8C1A B6FF41     567:                    LDAA    VIADRA
8C1D 39         568: WDCWAITRTS         RTS
```

```
                    570: *                  WDCWAITAVAILABLE -- WAIT FOR 7710 READY
                    571: *
  8C1E              572: WDCWAITAVAILABLE   EQU     *
  0001              573:                    IF      M6800
                    574:                    ;LDX    #(2*2500//256)*(1000//(4+2+2+4+2+4)) = 2.5 SECONDS AT 2MHZ
8C1E CE0460         575:                    LDX     #(2*2500//256)*(1000//(4+2+2+4+2+4))
  0003              576:                    ELSE    (M6809)
                    579:                    FIN
  8C21              580: WDCWAITAVAILABLELOOP ; WAIT AT MOST 2.5 SECONDS FOR DRIVE TO BE READY
                    581:                    ;LDB    VIAIFR
8C21 F6FF4D         582:                    LDAB    VIAIFR
8C24 C412           583:                    ANDB    #%00010010      CHECK: IS 7710 READY AND *BUS.DIR HIGH ?
                    584:                    ;CMPB   #%00010010      ...?
8C26 C112           585:                    CMPB    #%00010010
8C28 27F3           586:                    BEQ     WDCWAITRTS      ; B/ 7710 IS READY
8C2A 4A             587:                    DECA                    NO, DOWN COUNT LOWER 8 BITS OF FUSE
8C2B 26F4           588:                    BNE     WDCWAITAVAILABLELOOP B/ FUSE NOT BURNED UP
8C2D 09             589:                    DEX                     DOWN COUNT UPPER 16 BITS OF FUSE
8C2E 26F1           590:                    BNE     WDCWAITAVAILABLELOOP
                    591:                    ;LEAS   2,S             THROW RETURN ADDRESS AWAY
  0000              592:                    IF      2<0
                    595:                    ELSE
  0002              596:                    RPT     2
8C30 31             597:                    INS
                    598:                    FIN
                    599:                    ;JSR    WDCRESET        TRY TO GET DRIVE'S ATTENTION
8C32 BD8A36         600:                    JSR     WDCRESET
                    601:                    ;LDD    #ERR:DEVICENOTREADY DECLARE DEVICE NOT READY
8C35 C624           602:                    LDAB    #(ERR:DEVICENOTREADY)&$FF
8C37 8604           603:                    LDAA    #(ERR:DEVICENOTREADY)/256
                    604:                    ;JMP    WDCQUITWITHERR  GO STORE ERROR CODE IN DCB
8C39 7E8CE8         605:                    JMP     WDCQUITWITHERR
```

```
  8C3C          607: WDCWAIT4INT      EQU     *               WAIT FOR "DONE" INTERRUPT
                608:                  ;PULD
8C3C 32         609:                  PULA
8C3D 33         610:                  PULB
8C3E 01         611:                  NOP                     PREVENT WDC SELF INTERRUPT
8C3F 0F         612:                  SEI
                613:                  ;STD    WDCCONTINUEPC   ; SAVE WHERE TO GO ON CMD DONE INTERRUPT
8C40 F79329     614:                  STAB    WDCCONTINUEPC+1
8C43 B79328     615:                  STAA    WDCCONTINUEPC
                616:                  ;LDX    WDCDCBPOINTER
8C46 FE9326     617:                  LDX     WDCDCBPOINTER
                618:                  ;LDA    WDCREADWRITE,X
8C49 A642       619:                  LDAA    WDCREADWRITE,X
                620:                  ;CMPA   #WDCFORMAT
8C4B B101       621:                  CMPA    #WDCFORMAT
8C4D 2605       622:                  BNE     WDCWAIT4INT2    ; B/ STANDARD 3 SEC WAIT
                623:                  ;LDX    #0              ; FOREVER FOR FORMAT (TAKES 40 MINUTES ON 7710C)
8C4F CE0000     624:                  LDX     #0
8C52 2003       625:                  BRA     WDCWAIT4INT3
                626:
  8C54          627: WDCWAIT4INT2     EQU     *
                628:                  ;LDX    #5*TICKSPERSECOND+NTIMEOUTBLOCKS
8C54 CE0135     629:                  LDX     #5*TICKSPERSECOND+NTIMEOUTBLOCKS
8C57 FF93C2     630: WDCWAIT4INT3     STX     WDCTIMEOUTCOUNT
                631:                  ;LDB    #%10010000      ; SET VIA INTERRUPT ON *BUSDIR GOING HIGH
8C5A C690       632:                  LDAB    #%10010000
                633:                  ;STB    VIAIER
8C5C F7FF4E     634:                  STAB    VIAIER
                635:                  ;LDA    VIAIFR          ; IS DRIVE DONE WITH TRANSFER ?
8C5F B6FF4D     636:                  LDAA    VIAIFR
8C62 8510       637:                  BITA    #%00010000      ...?
8C64 2603       638:                  BNE     WDCINTERRUPT    B/ YES (THIS CODE HERE TO SIMPLIFY SINGLESTEPPING)
  8C66          639: WDCINTUNEXPECTED ;JMP    SDOS+SDOS:RTI   ; EXIT INTERRUPT SERVICE
8C66 7EBE15     640:                  JMP     SDOS+SDOS:RTI
                641:
  8C69          642: WDCINTERRUPT     ;LDX    #0
8C69 CE0000     643:                  LDX     #0
8C6C FF93C2     644:                  STX     WDCTIMEOUTCOUNT ; CLEAR TIME OUT
                645:                  ;LDB    #%00010000      ; KILL THE INTERRUPT ENABLE
8C6F C610       646:                  LDAB    #%00010000
                647:                  ;STB    VIAIER
8C71 F7FF4E     648:                  STAB    VIAIER
                649:                  ;LDX    WDCCONTINUEPC
8C74 FE9328     650:                  LDX     WDCCONTINUEPC
                651:                  ;LDD    #WDCINTUNEXPECTED RESET WHERE TO GO IF INTERRUPT
8C77 C666       652:                  LDAB    #(WDCINTUNEXPECTED)&$FF
8C79 868C       653:                  LDAA    #(WDCINTUNEXPECTED)/256
                654:                  ;STD    WDCCONTINUEPC
8C7B F79329     655:                  STAB    WDCCONTINUEPC+1
8C7E B79328     656:                  STAA    WDCCONTINUEPC
8C81 0E         657:                  CLI                     ; RE ENABLE INTERRUPTS - SO FLOPPY AND RTC CAN WORK
                658:                  ;JMP    0,X             ; RETURN TO CALLER
8C82 6E00       659:                  JMP     0,X
```

```
                        *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
                        *** THE SD STORAGE DEMON DRIVER ***
8C84 4D         661: WDCPROCST          TSTA                    PROCESS STATUS BITS
8C85 2706       662:                    BEQ     WDCPROCSTOKRTS  B/ NO TROUBLE AT ALL
8C87 8D1D       663:                    BSR     WDCSAVESTATUS   OOPS, HAD SOME KIND OF PROBLEM
8C89 8580       664:                    BITA    #WDCFATAL       WAS ERROR FATAL ?
8C8B 2605       665:                    BNE     WDCFATALERR     B/ YES
8C8D 39         666: WDCPROCSTOKRTS     RTS                     NON-FATAL ERROR, CONTINUE
                667:
8C8E 8D16       668: WDCFATAL0          BSR     WDCSAVESTATUS
8C90 2002       669:                    BRA     WDCFATAL2
                670:
  8C92          671: WDCFATALERR        ;LEAS   2,S             ; POP RETURN OFF STACK
  0000          672:                    IF      2<0
                675:                    ELSE
  0002          676:                    RPT     2
8C92 31         677:                    INS
                678:                    FIN
8C94 7A932D     679: WDCFATAL2          DEC     WDCRETRYCNT     ; ALL 9 LIVES USED UP ?
8C97 273D       680:                    BEQ     WDCQUIT         ; B/ YES, WE'RE DEAD
                681:                    ;LDA    WDCRETRYCNT     ON PENULTIMATE RETRY ?
8C99 B6932D     682:                    LDAA    WDCRETRYCNT
                683:                    ;CMPA   #1
8C9C 8101       684:                    CMPA    #1
8C9E 2603       685:                    BNE     JWDCCMDFEED     B/ NO, JUST SEND COMMANDS AGAIN
                686:                    ;JSR    WDCRESET        ; ON LAST TRY, HIT BELOW THE BELT
8CA0 BD8A36     687:                    JSR     WDCRESET
  8CA3          688: JWDCCMDFEED        ;JMP    WDCCMDFEED
8CA3 7E8A5C     689:                    JMP     WDCCMDFEED
                690:
  8CA6          691: WDCSAVESTATUS      ;LDX    WDCDCBPOINTER   ; SAVE ERROR STATUS
8CA6 FE9326     692:                    LDX     WDCDCBPOINTER
                693:                    ;LDX    DSKINFO:SECTORDB,X SAVE RDSI:LSN AS DSKINFO:ERRLSN
8CA9 EE2B       694:                    LDX     DSKINFO:SECTORDB,X
                695:                    ;LDB    RDSI:LSN,X
8CAB E602       696:                    LDAB    RDSI:LSN,X
                697:                    ;PSHD                   SAVE ERROR STATUS BYTE, UPPER 8 BITS OF LSN
8CAD 37         698:                    PSHB
8CAE 36         699:                    PSHA
                700:                    ;LDD    RDSI:LSN+1,X
8CAF E604       701:                    LDAB    (RDSI:LSN+1)+1,X
8CB1 A603       702:                    LDAA    RDSI:LSN+1,X
                703:                    ;LDX    WDCDCBPOINTER
8CB3 FE9326     704:                    LDX     WDCDCBPOINTER
                705:                    ;STD    DSKINFO:ERRLSN+1,X
8CB6 E741       706:                    STAB    (DSKINFO:ERRLSN+1)+1,X
8CB8 A740       707:                    STAA    DSKINFO:ERRLSN+1,X
                708:                    ;PULD                   RESTORE ERROR STATUS BYTE, UPPER 8 BITS OF LSN
8CBA 32         709:                    PULA
8CBB 33         710:                    PULB
                711:                    ;STB    DSKINFO:ERRLSN,X
8CBC E73F       712:                    STAB    DSKINFO:ERRLSN,X
                713:                    ;LDB    WDCREADWRITE,X
8CBE E642       714:                    LDAB    WDCREADWRITE,X
                715:                    ;CMPB   #WDCREADCMD     IS THIS A READ OR A WRITE COMMAND ?
8CC0 C102       716:                    CMPB    #WDCREADCMD
8CC2 2709       717:                    BEQ     WDCSAVEREADSTATUS
```

```
                718:                    ;STA    DSKINFO:WRITEERRSTS,X ; LAST WRITE (OR FORMAT) ERROR STATUS
8CC4 A736       719:                    STAA    DSKINFO:WRITEERRSTS,X
8CC6 6C35       720:                    INC     DSKINFO:WRITEERRCNT+1,X ; 2 BYTE ERROR COUNT
8CC8 2602       721:                    BNE     WDCSAVEWRITESTATUS1
8CCA 6C34       722:                    INC     DSKINFO:WRITEERRCNT,X
  8CCC          723: WDCSAVEWRITESTATUS1
8CCC 39         724:                    RTS
                725:
  8CCD          726: WDCSAVEREADSTATUS  ;STA    DSKINFO:READERRSTS,X ; LOG LAST READ ERROR
8CCD A73A       727:                    STAA    DSKINFO:READERRSTS,X
8CCF 6C39       728:                    INC     DSKINFO:READERRCNT+1,X
8CD1 2602       729:                    BNE     WDCSAVEREADSTATUS1
8CD3 6C38       730:                    INC     DSKINFO:READERRCNT,X
  8CD5          731: WDCSAVEREADSTATUS1
8CD5 39         732:                    RTS
```

```
MAL/6800 1.3F: 8CD5  SDOSDRIVERS        *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 62; Form 1      *** THE SD STORAGE DEMON DRIVER ***
IOSTOREDEMON.ASM
  8CD6          734: WDCQUIT            ;CMPA   #%10011111      A TIMEOUT IN MIDDLE OF TRANSFER ?
8CD6 819F       735:                    CMPA    #%10011111
8CD8 2718       736:                    BEQ     WDCTIMEDOUT1    B/ YES, LET THE USER KNOW!
                737:                    ;CMPB   #WDCREADCMD
8CDA C102       738:                    CMPB    #WDCREADCMD
8CDC 2706       739:                    BEQ     WDCQUITREAD
                740:                    ;LDD    #ERR:DISKWRITE
8CDE C616       741:                    LDAB    #(ERR:DISKWRITE)&$FF
8CE0 8604       742:                    LDAA    #(ERR:DISKWRITE)/256
8CE2 2004       743:                    BRA     WDCQUITWITHERR
                744:
  8CE4          745: WDCQUITREAD        ;LDD    #ERR:DISKREAD
8CE4 C615       746:                    LDAB    #(ERR:DISKREAD)&$FF
8CE6 8604       747:                    LDAA    #(ERR:DISKREAD)/256
  8CE8          748: WDCQUITWITHERR     ;LDX    WDCDCBPOINTER   JUST TO BE SAFE...
8CE8 FE9326     749:                    LDX     WDCDCBPOINTER
                750:                    ;STD    DCB:LASTERROR,X
8CEB E702       751:                    STAB    (DCB:LASTERROR)+1,X
8CED A701       752:                    STAA    DCB:LASTERROR,X
                753:                    ;JMP    WDCDONE
8CEF 7E8B44     754:                    JMP     WDCDONE
                755:
  8CF2          756: WDCTIMEDOUT        EQU     *
  8CF2          757: WDCTIMEDOUT1       ;LDX    #WDCINTUNEXPECTED REMEMBER THAT WE DON'T EXPECT AN INTERRUPT!
8CF2 CE8C66     758:                    LDX     #WDCINTUNEXPECTED
8CF5 FF9328     759:                    STX     WDCCONTINUEPC
                760:                    ;JSR    WDCRESET        ; HIT HIM SO MAYBE HE WILL WAKE UP
8CF8 BD8A36     761:                    JSR     WDCRESET
                762:                    ;LDD    #ERR:DEVICETIMEDOUT
8CFB C612       763:                    LDAB    #(ERR:DEVICETIMEDOUT)&$FF
8CFD 8604       764:                    LDAA    #(ERR:DEVICETIMEDOUT)/256
8CFF 20E7       765:                    BRA     WDCQUITWITHERR
                766:
  8D01          767: WDCSET4TRANS       ;LDA    #WDCNBPS/8      ; SET NUMBER OF 8 BYTE BLOCKS TO TRANSFER
8D01 8640       768:                    LDAA    #WDCNBPS/8
                769:                    ;STA    WDCCOUNT
8D03 B7932A     770:                    STAA    WDCCOUNT
                771:                    ;LDX    WDCDCBPOINTER
8D06 FE9326     772:                    LDX     WDCDCBPOINTER
                773:                    ;LDX    DSKINFO:SECTORDB,X
8D09 EE2B       774:                    LDX     DSKINFO:SECTORDB,X
                775:                    ;LDX    RDSI:SECTORBASE,X
8D0B EE05       776:                    LDX     RDSI:SECTORBASE,X
8D0D FF932B     777:                    STX     WDCPOINTER      ; SET UP POINTER TO 1ST BLOCK OF 8 TO MOVE
                778:                    ;LDB    #%00000010      ; GET 'READY' LINE SENSE MASK
8D10 C602       779:                    LDAB    #%00000010
8D12 39         780:                    RTS
                781:                    FIN     IODRIVERBODY
  0000          782:                    IF      IODRIVERPOLL
                807:                    FIN     IODRIVERPOLL
  0000          808:                    IF      IODRIVERINIT
                833:                    FIN     IODRIVERINIT
  0000          834:                    IF      IODRIVERRAM
                886:                    FIN     IODRIVERRAM
                887:                    END     ;UNEXPECTED EOF
```

```
                414:          FIN
                415:          INCLUDE                    IDVTCONFIG.ASM GENERATED BY MAKEVTCONFIG IMMEDIATELY BEFORE ASSY
  0001            1:                        if      iodriverbody
  0001            2:                        ifund   profile.MALVT
  0001            3: profile.MALVT          equ     1
                  4:                        fin     profile.MALVT
  0001            5:                        ifund   profile.MALLPT
  0001            6: profile.MALLPT         equ     1
                  7:                        fin     profile.MALLPT
  0000            8:                        ifund   profile.MALVT
                 10:                        fin     profile.MALVT
  0001           11:                        ifund   profile.RS232LPT
  0001           12: profile.RS232LPT       equ     1
                 13:                        fin     profile.RS232LPT
  0001           14:                        ifund   profile.EPSONLPT
  0001           15: profile.EPSONLPT       equ     1
                 16:                        fin     profile.EPSONLPT
  0001           17:                        ifund   profile.ADM3
  0001           18: profile.ADM3           equ     1
                 19:                        fin     profile.ADM3
  0001           20:                        ifund   profile.GT100
  0001           21: profile.GT100          equ     1
                 22:                        fin     profile.GT100
  0001           23:                        ifund   profile.H19
  0001           24: profile.H19            equ     1
                 25:                        fin     profile.H19
  0001           26:                        ifund   profile.SOROCIQ120
  0001           27: profile.SOROCIQ120     equ     1
                 28:                        fin     profile.SOROCIQ120
```

```
                    30:                 fin     iodriverbody
  0000              31:                 if      iodriverinit
                    39:                 fin     iodriverinit
  0001              40:                 if      iodriverbody
  8D13              41: ilputdev:$FFC0
8D13 B7FFC1         42:                 staa    $FFC1           output data
8D16 86B5           43:                 ldaa    #%10110101      enable output interrupts
8D18 B7FFC0         44:                 staa    $FFC0
  8D1B              45: rts:$FFC0
8D1B 39             46:                 rts
  8D1C              47: ilgetdev:$FFC0
8D1C F6FFC0         48:                 ldab    $FFC0           get the status first (becuz reading data clears it)
8D1F B6FFC1         49:                 ldaa    $FFC1           get the interrupt-causing data
8D22 7E8EA2         50:                 jmp     ilgetdevstatusfromacia the rest is common code
  8D25              51: tlcheckready:$FFC0
8D25 B6FFC0         52:                 ldaa    $FFC0           get the status
8D28 46             53:                 rora                    shift *DCD into carry bit
8D29 46             54:                 rora
8D2A 46             55:                 rora
8D2B 39             56:                 rts
                    57:                 fin     iodriverbody
  0000              58:                 if      iodriverinit
                    66:                 fin     iodriverinit
  0001              67:                 if      iodriverbody
  8D2C              68: ilputdev:$FFC4
8D2C B7FFC5         69:                 staa    $FFC5           output data
8D2F 86B5           70:                 ldaa    #%10110101      enable output interrupts
8D31 B7FFC4         71:                 staa    $FFC4
  8D34              72: rts:$FFC4
8D34 39             73:                 rts
  8D35              74: ilgetdev:$FFC4
8D35 F6FFC4         75:                 ldab    $FFC4           get the status first (becuz reading data clears it)
8D38 B6FFC5         76:                 ldaa    $FFC5           get the interrupt-causing data
8D3B 7E8EA2         77:                 jmp     ilgetdevstatusfromacia the rest is common code
  8D3E              78: tlcheckready:$FFC4
8D3E B6FFC4         79:                 ldaa    $FFC4           get the status
8D41 46             80:                 rora                    shift *DCD into carry bit
8D42 46             81:                 rora
8D43 46             82:                 rora
8D44 39             83:                 rts
                    84:                 fin     iodriverbody
  0000              85:                 if      iodriverinit
                    93:                 fin     iodriverinit
  0001              94:                 if      iodriverbody
  8D45              95: ilputdev:$FFC8
8D45 B7FFC9         96:                 staa    $FFC9           output data
8D48 86B5           97:                 ldaa    #%10110101      enable output interrupts
8D4A B7FFC8         98:                 staa    $FFC8
  8D4D              99: rts:$FFC8
8D4D 39            100:                 rts
  8D4E             101: ilgetdev:$FFC8
8D4E F6FFC8        102:                 ldab    $FFC8           get the status first (becuz reading data clears it)
8D51 B6FFC9        103:                 ldaa    $FFC9           get the interrupt-causing data
8D54 7E8EA2        104:                 jmp     ilgetdevstatusfromacia the rest is common code
  8D57             105: tlcheckready:$FFC8
```

IOVTCONFIG.ASM

```
8D57 B6FFC8     106:                    ldaa    $FFC8           get the status
8D5A 46         107:                    rora                    shift *DCD into carry bit
8D5B 46         108:                    rora
8D5C 46         109:                    rora
8D5D 39         110:                    rts
                111:                    fin     iodriverbody
  0000          112:                    if      iodriverpoll
                193:                    fin     iodriverpoll
  0000          194:                    if      iodriverram
                521:                    fin     iodriverram
                522:
                523:
                416:            INCLUDE                         IOVTDPBS.ASM
  0000            1:             if      iodriverpoll
                  3:             fin     iodriverpoll
  0001            4:             if      iodriverbody
  0001            5:             ifund   nexdpb
  0000            6: nextdpb     set     0
                  7:             fin     nextdpb
```

IOVTDPBS.ASM

```
  0001           9: profilenum.malvt equ 1
                10:
  8D5E          11: thisdpb   set     *
8D5E 01         12:           fcb     profilenum.malvt            profile name
8D5F 04         13:           fcb     dvtyp.console
8D60 0000       14:           fdb     nextdpb                     next profile
8D62 50         15:           fcb     80                          default width
8D63 18         16:           fcb     24                          default depth
8D64 05         17:           fcb     5                           flags
8D65 0171       18:           fdb     6*tickspersecond+ntimeoutblocks
8D67 0C39       19:           okrts                               input translation routine
  0001          20:           if      m6800!m6801
8D69 01         21:           nop
                22:           fin
8D6A 7EBDD0     23:           jmp     sdos+sdos:vtmalvt           perform control functions
8D6D 39         24:           rts                                 set output coloring
8D6E 01         25:           nop
8D6F 01         26:           nop
8D70 39         27:           rts                                 set background coloring
                28: *         nop
                29: *         nop
                30: *         fcb      0,0,0,0,0,0,0,0            gpinit data
  8D5E          31: nextdpb   set     thisdpb
```

```
  0009          33: profilenum.mallpt equ 9
                34:
  8D71          35: thisdpb   set     *
8D71 09         36:           fcb     profilenum.mallpt           profile name
8D72 05         37:           fcb     dvtyp.printer
8D73 8D5E       38:           fdb     nextdpb                     next profile
8D75 84         39:           fcb     132                         default width
8D76 42         40:           fcb     66                          default depth
8D77 03         41:           fcb     3                           flags
8D78 01710266   42:           fdb     6*tickspersecond+ntimeoutblocks
8D7A 0C39       43:           okrts                               input translation routine
  0001          44:           if      m6800!m6801
8D7C 01         45:           nop
                46:           fin
8D7D 7EBDCD     47:           jmp     sdos+sdos:vtmallpt          perform control functions
8D80 39         48:           rts     set output coloring
8D81 01         49:           nop
8D82 01         50:           nop
8D83 39         51:           rts                                 set background coloring
                52: *         nop
                53: *         nop
                54: *         fcb      0,0,0,0,0,0,0,0            gpinit data
  8D71          55: nextdpb   set     thisdpb
```

```
  0006          57: profilenum.hardcopyvt equ 6
                58:
  8D84          59: thisdpb   set     *
8D84 06         60:           fcb     profilenum.hardcopyvt       profile name
8D85 04         61:           fcb     dvtyp.console
8D86 8D71       62:           fdb     nextdpb                     next profile
8D88 50         63:           fcb     80                          default width
8D89 00         64:           fcb     0                           default depth
8D8A 14         65:           fcb     20                          flags
8D8B 0171       66:           fdb     6*tickspersecond+ntimeoutblocks
8D8D 0C39       67:           okrts                               perform input translation
  0001          68:           if      m6800!m6801
8D8F 01         69:           nop
                70:           fin
8D90 0D39       71:           errorrts                            perform default control functions
  0001          72:           if      m6800!m6801
8D92 01         73:           nop
                74:           fin
8D93 39         75:           rts                                 set output coloring
8D94 01         76:           nop
8D95 01         77:           nop
8D96 39         78:           rts                                 set background coloring
                79: *         nop
                80: *         nop
                81: *         fcb     0,0,0,0,0,0,0,0             gpinit data
  8D84          82: nextdpb   set     thisdpb
  0001          83:           ifund   profile.cenlpt
  0002          84:           else
                112:          fin     profile.cenlpt
  0000          113:          ifund   profile.rs232lpt
                114:          else
```

```
                116: ;   Operates a printer using standard ascii control codes for everything
                117:
  000B          118: profilenum.rs232lpt equ 11
                119:
  8D97          120: thisdpb   set     *
8D97 0B         121:           fcb     profilenum.rs232lpt          profile name
8D98 05         122:           fcb     dvtyp.printer
8D99 8D84       123:           fdb     nextdpb                      next profile
8D9B 84         124:           fcb     132                          default width
8D9C 42         125:           fcb     66                           default depth
8D9D 02         126:           fcb     2                            flags
8D9E 0171       127:           fdb     6*tickspersecond+ntimeoutblocks
8DA0 0C39       128:           okrts                                input translation routine
  0001          129:           if      m6800!m6801
8DA2 01         130:           nop
                131:           fin
8DA3 0D39       132:           errorrts                             perform default control functions
  0001          133:           if      m6800!m6801
8DA5 01         134:           nop
                135:           fin
8DA6 39         136:           rts                                  set output coloring
8DA7 01         137:           nop
8DA8 01         138:           nop
8DA9 39         139:           rts                                  set background coloring
                140: *         nop
                141: *         nop
                142: *         fcb      0,0,0,0,0,0,0,0             gpinit data
  8D97          143: nextdpb   set     thisdpb
                144:           fin     profile.rs232lpt
  0001          145:           ifund   profile.adm1
  0002          146:           else
                219:           fin     profile.adm1
  0000          220:           ifund   profile.adm3
                221:           else
```

```
  0003          223: profilenum.adm3 equ 3
                224:
  8DAA          225: thisdpb   set     *
8DAA 03         226:           fcb     profilenum.adm3              profile name
8DAB 04         227:           fcb     dvtyp.console
8DAC 8D97       228:           fdb     nextdpb                      next profile
8DAE 50         229:           fcb     80                           default width
8DAF 18         230:           fcb     24                           default depth
8DB0 04         231:           fcb     4                            flags
8DB1 0171       232:           fdb     6*tickspersecond+ntimeoutblocks
8DB3 7E8DC0     233:           jmp     xlatei:adm3                  perform input translation
8DB6 7E8DCF     234:           jmp     specialoutput:adm3           perform control functions
8DB9 39         235:           rts                                  set output coloring
8DBA 01         236:           nop
8DBB 01         237:           nop
8DBC 39         238:           rts                                  set background coloring
8DBD 01         239:           nop
8DBE 01         240:           nop
8DBF 00         241:           fcb     0                            initial XLATEI state byte
  8DAA          242: nextdpb   set     thisdpb
```

```
  0001          244:          ifund    dcb:xlatestate
  003D          245: dcb:xlatestate equ dcb:profile+dpb:gpinit        translate state byte is 1st gp byte
                246:          fin
  8DC0          247: xlatei:adm3
                248: ;   translate adm3 input found in (a), returning translated character in (a)
                249: ;   with carry clear; carry set will cause character to be lost
8DC0 C67F       250:          ldab     #$7f                         assume swap for underscore key
8DC2 815F       251:          cmpa     #$5f                         swap for underscore ?
8DC4 2706       252:          beq      xlatei:adm3.b                b/ yes, use (B) as translation
8DC6 C65F       253:          ldab     #$5f                         assume swap for DEL key
8DC8 817F       254:          cmpa     #$7f                         swap for DEL key ?
8DCA 2601       255:          bne      xlatei:adm3.done             b/ no
  8DCC          256: xlatei:adm3.b ; use (B) as translation
8DCC 17         257:          tba
  8DCD          258: xlatei:adm3.done
8DCD 0C39       259:          okrts
```

```
  8DCF          261: specialoutput:adm3
                262: ;   called to perform control functions for a Lear Siegler ADM 3-A
                263: ;   The position, and clear functions are implemented.  All others
                264: ;   must be simulated by the VT driver.
8DCF 8181       265:            cmpa    #specialfn:posn
8DD1 2706       266:            beq     specialoutput:adm3posn
8DD3 8182       267:            cmpa    #specialfn:clear
8DD5 271E       268:            beq     specialoutput:adm3clear
8DD7 0D39       269:            errorrts                        adm3a can't do anything else
                270:
  8DD9          271: specialoutput:adm3posn
8DD9 861B       272:            ldaa    #ascii:esc
8DDB AD7D       273:            jsr     dcb:tlbuffer,x
8DDD 863D       274:            ldaa    #'=                     set up to do a position fn
8DDF AD7D       275:            jsr     dcb:tlbuffer,x
8DE1 30         276:            tsx
8DE2 A602       277:            ldaa    2,x
8DE4 8B20       278:            adda    ##$20
8DE6 DE06       279:            ldx     dcbpointer
8DE8 AD7D       280:            jsr     dcb:tlbuffer,x
8DEA 30         281:            tsx
8DEB A603       282:            ldaa    3,x
8DED 8B20       283:            adda    ##$20
8DEF DE06       284:            ldx     dcbpointer
8DF1 AD7D       285:            jsr     dcb:tlbuffer,x
8DF3 0C39       286:            okrts
                287:
  8DF5          288: specialoutput:adm3clear
8DF5 861A       289:            ldaa    ##$1a                   performs a clear screen fn
8DF7 AD7D       290:            jsr     dcb:tlbuffer,x
8DF9 0C39       291:            okrts
                292:            fin     profile.adm3
  0001          293:            ifund   profile.tvi912c
  0002          294:            else
                360:            fin     profile.tvi912c
  0001          361:            ifund   profile.soroc120
  0002          362:            else
                431:            fin     profile.soroc120
  0000          432:            ifund   profile.h19
                433:            else
```

```
  0005          435: profilenum.h19 equ 5
                436:
  8DFB          437: thisdpb   set     *
8DFB 05         438:           fcb     profilenum.h19              profile name
8DFC 04         439:           fcb     dvtyp.console
8DFD 8DAA       440:           fdb     nextdpb                     next profile
8DFF 50         441:           fcb     80                          default width
8E00 18         442:           fcb     24                          default depth
8E01 04         443:           fcb     4                           flags
8E02 0171       444:           fdb     6*tickspersecond+ntimeoutblocks
8E04 7E8E11     445:           jmp     xlatei:h19                  perform input translation
8E07 7E8E50     446:           jmp     specialoutput:h19           perform control functions
8E0A 7E8E8E     447:           jmp     coloring:h19                set output coloring
8E0D 39         448:           rts                                 set background coloring (none)
8E0E 01         449:           nop
8E0F 01         450:           nop
8E10 00         451:           fcb     0                           initial XLATEI state byte
  8DFB          452: nextdpb   set     thisdpb
```

```
  0000          454:          ifund    dcb:xlatestate
                456:          fin
  8E11          457: xlatei:h19
                458: ;   translate h19 input found in (a), returning translated character in (a)
                459: ;   with carry clear; carry set will cause character to be lost
8E11 6D3D       460:          tst      dcb:xlatestate,x
8E13 260B       461:          bne      xlatei:h19.escape
8E15 811B       462:          cmpa     #ascii:esc
8E17 2605       463:          bne      xlatei:h19.ok
8E19 6C3D       464:          inc      dcb:xlatestate,x
8E1B 0D39       465:          errorrts
                466:
  8E1D          467: xlatei:h19.b ; use (B) as translation of character
8E1D 17         468:          tba
  8E1E          469: xlatei:h19.ok ; (A) is tranlated character
8E1E 0C39       470:          okrts
                471:
  8E20          472: xlatei:h19.escape
                473: ;   if character following <ESC> is not A, B, C, D, J, N, or Q,
                474: ;   then bitch and revert to the standard state
8E20 6F3D       475:          clr      dcb:xlatestate,x
8E22 C60B       476:          ldab     #ascii:vt
8E24 8141       477:          cmpa     #'A                        cursor up?
8E26 27F5       478:          beq      xlatei:h19.b
8E28 C60A       479:          ldab     #ascii:lf
8E2A 8142       480:          cmpa     #'B                        cursor down?
8E2C 27EF       481:          beq      xlatei:h19.b
8E2E C60C       482:          ldab     #ascii:ff
8E30 8143       483:          cmpa     #'C                        cursor right?
8E32 27E9       484:          beq      xlatei:h19.b
8E34 C608       485:          ldab     #ascii:bs
8E36 8144       486:          cmpa     #'D                        cursor left?
8E38 27E3       487:          beq      xlatei:h19.b
8E3A C605       488:          ldab     #ascii:enq
8E3C 814A       489:          cmpa     #'J                        ^E?
8E3E 27DD       490:          beq      xlatei:h19.b
8E40 C615       491:          ldab     #ascii:nak
8E42 814E       492:          cmpa     #'N                        ^U?
8E44 27D7       493:          beq      xlatei:h19.b
8E46 C61B       494:          ldab     #ascii:esc
8E48 8151       495:          cmpa     #'Q                        <ESC>?
8E4A 27D1       496:          beq      xlatei:h19.b
  8E4C          497: xlatei:h1932
8E4C 6C5E       498:          inc      dcb:beepcount,x
8E4E 0D39       499:          errorrts                            ignore character
```

```
  8E50          501: specialoutput:h19
                502: ;   called to perform control functions for a Heath H-19
                503: ;   The position, clear and erase to end of line functions are implemented.
8E50 8181       504:            cmpa    #specialfn:posn
8E52 270A       505:            beq     specialoutput:h19posn
8E54 8182       506:            cmpa    #specialfn:clear
8E56 2722       507:            beq     specialoutput:h19clear
8E58 8183       508:            cmpa    #specialfn:eeol
8E5A 2728       509:            beq     specialoutput:h19eeol
8E5C 0D39       510:            errorrts                                h19 can't do anything else
                511:
  8E5E          512: specialoutput:h19posn
8E5E 861B       513:            ldaa    #ascii:esc
8E60 AD7D       514:            jsr     dcb:tlbuffer,x
8E62 8659       515:            ldaa    #'Y                             set up to do a position fn
8E64 AD7D       516:            jsr     dcb:tlbuffer,x
8E66 30         517:            tsx
8E67 A602       518:            ldaa    2,x
8E69 8B20       519:            adda    #$20
8E6B DE06       520:            ldx     dcbpointer
8E6D AD7D       521:            jsr     dcb:tlbuffer,x
8E6F 30         522:            tsx
8E70 A603       523:            ldaa    3,x
8E72 8B20       524:            adda    #$20
8E74 DE06       525:            ldx     dcbpointer
8E76 AD7D       526:            jsr     dcb:tlbuffer,x
8E78 0C39       527:            okrts
                528:
  8E7A          529: specialoutput:h19clear
8E7A 861B       530:            ldaa    #ascii:esc
8E7C AD7D       531:            jsr     dcb:tlbuffer,x
8E7E 8645       532:            ldaa    #'E                             performs a clear screen fn
8E80 AD7D       533:            jsr     dcb:tlbuffer,x
8E82 0C39       534:            okrts
                535:
  8E84          536: specialoutput:h19eeol
8E84 861B       537:            ldaa    #ascii:esc
8E86 AD7D       538:            jsr     dcb:tlbuffer,x
8E88 864B       539:            ldaa    #'K                             performs a eeol fn
8E8A AD7D       540:            jsr     dcb:tlbuffer,x
8E8C 0C39       541:            okrts
```

```
  8E8E          543: coloring:h19
8E8E 8508       544:          bita    #%00001000                  reverse video desired ?
8E90 2608       545:          bne     coloring:h19reversevideo    b/ yes
8E92 861B       546:          ldaa    #ascii:esc                  send "normal video" command
8E94 AD7D       547:          jsr     dcb:tlbuffer,x
8E96 8671       548:          ldaa    #'q
8E98 6E7D       549:          jmp     dcb:tlbuffer,x
                550:
  8E9A          551: coloring:h19reversevideo
8E9A 861B       552:          ldaa    #ascii:esc                  send "reverse video" request
8E9C AD7D       553:          jsr     dcb:tlbuffer,x
8E9E 8670       554:          ldaa    #'p
8EA0 6E7D       555:          jmp     dcb:tlbuffer,x
                556:          fin     profile.h19
  0001          557:          ifund   profile.hazeltine
  0002          558:          else
                561:          fin     profile.hazeltine
  0001          562:          ifund   profile.beehive
  0002          563:          else
                566:          fin     profile.beehive
  0001          567:          ifund   profile.exorterm155
  0002          568:          else
                743:          fin     exorterm155
```

```
  8EA2          745: ilgetdevstatusfromacia
                746:
                747: ;   receives control from the acia device access routine, returns either
                748: ;   the available character, with carry clear, or the error status, with
                749: ;   carry set.
                750:
                751: ;   the data or error status is returned in (a)
                752:
                753: ;   the status is defined as:
                754:
                755: ;         %00000001               framing error (probably BREAK received)
                756: ;         %00000010               overrun (data lost)
                757: ;         %00000100               parity error
                758:
8EA2 C570       759:            bitb    #%01110000      check for parity, overrun, or framing errors
8EA4 2602       760:            bne     ilgetdevicestatusfromaciaerror
8EA6 0C39       761:            okrts
                762:
  8EA8          763: ilgetdevicestatusfromaciaerror
8EA8 17         764:            tba                     make a standard status byte
8EA9 44         765:            lsra
8EAA 44         766:            lsra
8EAB 44         767:            lsra
8EAC 44         768:            lsra
8EAD 8407       769:            anda    #%00000111
8EAF 0D39       770:            errorrts
                771:            fin     iodriverbody
                772:            end
```

```
  0001          418:          IFUND INTERRUPTSTACKSIZE
  0046          419: INTERRUPTSTACKSIZE                    EQU MINSTACK+7+7+7+7+7+7+7 7 FOR EACH POSSIBLE
                420: *                                     NESTED INTERRUPT RESULTING FROM A DEVICE
                421:          FIN   INTERRUPTSTACKSIZE
                422: *
8EB1 9377       423: CNFGTABLE      FDB                    DISKDCBS DEFAULT DISK MUST BE FIRST
8EB3 9627       424:          FDB   TTYDCB                 CONSOLE MUST BE FIRST
8EB5 9D8D       425:          FDB   IOCBPOINTERS
8EB7 08         426:          FCB   NIOCHANNELS
8EB8 9DE3       427:          FDB   DSKBUFFERPOOL
8EBA 081D       428:          FDB   DSKPOOLSIZE
8EBC BDD3       429:          FDB   SDOS+SDOS:VTATTNCHECK
8EBE 8407       430:          FDB   DEBUGSYSCALLHANDLER
8EC0 8400       431:          FDB   DRIVERBASE
8EC2 9D9D       432:          FDB   INTSETUP
  8EC4          433: INTDISABLE
  0001          434:          IF    M6800!M6801
8EC4 01         435:          NOP                          INT DISABLE
                436:          FIN
8EC5 0F         437:          SEI
8EC6 39         438:          RTS
  8EC7          439: INTENABLE
8EC7 0E         440:          CLI
8EC8 39         441:          RTS
  0001          442:          IF    M6800!M6801
8EC9 01         443:          NOP
                444:          FIN
8ECA 3B         445: INTRTI   RTI                          INT RTI
8ECB 01         446:          NOP
8ECC 01         447:          NOP
8ECD 9DE2       448:          FDB   INTERRUPTSTACKEND-1
8ECF 8FA2       449:          FDB   STACKSWITCHEDDEVICEPOLL     ROUTINE TO DETERMINE INTERRUPTING DEVICE
8ED1 9724       450:          FDB   TASKQUEUE
8ED3 96C8       451:          FDB   TIMEOUTQUEUE
8ED5 8DFB       452:          FDB   NEXTDPB
8ED7 8410       453:          FDB   DEBUGINTERRUPT         WITH CONTEXT BLOCK ON STACK
                454: *
  0000          455:         IF     SDOSMT
                457:         ELSE
8ED9 0000       458:         FDB    0                       NO SDOS/MT PRIMITIVES
                459:         FIN
  0000          460:         IF     SDOSMT
                541:          FIN   SDOSMT
                542:
8EDB BDBE2D     543: ILLDEVICEOP    JSR                    SDOS+SDOS:ERROR
8EDE 040A       544:          FDB   ERR:ILLDEVICEOP
                545: *
8EE0 BDBE30     546: ERRETX   JSR   SDOS+SDOS:ERRORSAVE
8EE3 7EBE33     547:          JMP   SDOS+SDOS:ERRORED
                548:
  0032          549: PATCHSPACE     RPT                    50
8EE6 3F         550:          SWI
```

```
  0000          552: IODRIVERBODY   SET                     0
  0001          553: IODRIVERPOLL   SET                     1
                554:
  8F18          555: STACKUNSWITCHEDDEVICEPOLL ; come here via IRQ vector
                556:          INCLUDE                       IOVTCONFIG.ASM
  0000            1:                    if      iodriverbody
                 30:                    fin     iodriverbody
  0000           31:                    if      iodriverinit
                 39:                    fin     iodriverinit
  0000           40:                    if      iodriverbody
                 57:                    fin     iodriverbody
  0000           58:                    if      iodriverinit
                 66:                    fin     iodriverinit
  0000           67:                    if      iodriverbody
                 84:                    fin     iodriverbody
  0000           85:                    if      iodriverinit
                 93:                    fin     iodriverinit
  0000           94:                    if      iodriverbody
                111:                    fin     iodriverbody
  0001          112:                    if      iodriverpoll
```

```
  8F18          114: vt:interruptpollchain      ; should be first in poll chain
8F18 B6FFC0     115:                    ldaa    $FFC0           test irq
8F1B 2A28       116:                    bpl     noint:$FFC0     b/ no int here
8F1D CE9627     117:                    ldx     #dcb:$FFC0      look more closely
8F20 8182       118:                    cmpa    #%10000010      handle output interrupts pronto!
8F22 2608       119:                    bne     notoutput:$FFC0
  8F24          120: gotoutput:$FFC0
8F24 8695       121:                    ldaa    #%10010101      disable output interrupt
8F26 B7FFC0     122:                    staa    $FFC0
8F29 7EBDDF     123:                    jmp     sdos+sdos:vtoutputint
  8F2C          124: notoutput:$FFC0
8F2C 8504       125:                    bita    #%00000100      data carrier detect dropped
8F2E 2708       126:                    beq     notdcddrop:$FFC0
8F30 F6FFC1     127:                    ldab    $FFC1           clear interrupt caused by ^dcd
8F33 8502       128:                    bita    #%00000010      output requested with ^dcd?
8F35 26ED       129:                    bne     gotoutput:$FFC0 b/ yup!
  8F37          130: rti:$FFC0
8F37 3B         131:                    rti
  8F38          132: notdcddrop:$FFC0
8F38 8571       133:                    bita    #%01110001      receiver register full or error
8F3A 2703       134:                    beq     notinput:$FFC0
8F3C 7EBDDC     135:                    jmp     sdos+sdos:vtinputint
  8F3F          136: notinput:$FFC0
8F3F 8508       137:                    bita    #%00001000      not CTS
8F41 26F4       138:                    bne     rti:$FFC0       ignore CTS interrupt glitch
8F43 20FE       139:                    bra     *               wierd condition
  8F45          140: noint:$FFC0
8F45 B6FFC4     141:                    ldaa    $FFC4           test irq
8F48 2A28       142:                    bpl     noint:$FFC4     b/ no int here
8F4A CE9763     143:                    ldx     #dcb:$FFC4      look more closely
8F4D 8182       144:                    cmpa    #%10000010      handle output interrupts pronto!
8F4F 2608       145:                    bne     notoutput:$FFC4
  8F51          146: gotoutput:$FFC4
8F51 8695       147:                    ldaa    #%10010101      disable output interrupt
8F53 B7FFC4     148:                    staa    $FFC4
8F56 7EBDDF     149:                    jmp     sdos+sdos:vtoutputint
  8F59          150: notoutput:$FFC4
8F59 8504       151:                    bita    #%00000100      data carrier detect dropped
8F5B 2708       152:                    beq     notdcddrop:$FFC4
8F5D F6FFC5     153:                    ldab    $FFC5           clear interrupt caused by ^dcd
8F60 8502       154:                    bita    #%00000010      output requested with ^dcd?
8F62 26ED       155:                    bne     gotoutput:$FFC4 b/ yup!
  8F64          156: rti:$FFC4
8F64 3B         157:                    rti
  8F65          158: notdcddrop:$FFC4
8F65 8571       159:                    bita    #%01110001      receiver register full or error
8F67 2703       160:                    beq     notinput:$FFC4
8F69 7EBDDC     161:                    jmp     sdos+sdos:vtinputint
  8F6C          162: notinput:$FFC4
8F6C 8508       163:                    bita    #%00001000      not CTS
8F6E 26F4       164:                    bne     rti:$FFC4       ignore CTS interrupt glitch
8F70 20FE       165:                    bra     *               wierd condition
  8F72          166: noint:$FFC4
8F72 B6FFC8     167:                    ldaa    $FFC8           test irq
8F75 2A28       168:                    bpl     noint:$FFC8     b/ no int here
```

```
MAL/6800 1.3F: 8F77  SDOSDRIVERS       *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 81; Form 1     VT Interrupt Poll Chain
IOVTCONFIG.ASM
8F77 CE98A2     169:                   ldx     #dcb:$FFC8      look more closely
8F7A 8182       170:                   cmpa    #%10000010      handle output interrupts pronto!
8F7C 2608       171:                   bne     notoutput:$FFC8
  8F7E          172: gotoutput:$FFC8
8F7E 8695       173:                   ldaa    #%10010101      disable output interrupt
8F80 B7FFC8     174:                   staa    $FFC8
8F83 7EBDDF     175:                   jmp     sdos+sdos:vtoutputint
  8F86          176: notoutput:$FFC8
8F86 8504       177:                   bita    #%00000100      data carrier detect dropped
8F88 2708       178:                   beq     notdcddrop:$FFC8
8F8A F6FFC9     179:                   ldab    $FFC9           clear interrupt caused by ^dcd
8F8D 8502       180:                   bita    #%00000010      output requested with ^dcd?
8F8F 26ED       181:                   bne     gotoutput:$FFC8 b/ yup!
  8F91          182: rti:$FFC8
8F91 3B         183:                   rti
  8F92          184: notdcddrop:$FFC8
8F92 8571       185:                   bita    #%01110001      receiver register full or error
8F94 2703       186:                   beq     notinput:$FFC8
8F96 7EBDDC     187:                   jmp     sdos+sdos:vtinputint
  8F99          188: notinput:$FFC8
8F99 8508       189:                   bita    #%00001000      not CTS
8F9B 26F4       190:                   bne     rti:$FFC8       ignore CTS interrupt glitch
8F9D 20FE       191:                   bra     *               wierd condition
  8F9F          192: noint:$FFC8
                193:                   fin     iodriverpoll
  0000          194:                   if      iodriverram
                521:                   fin     iodriverram
                522:
                523:
8F9F 7EBE12     557:           JMP   SDOS+SDOS:IOINT          go switch stacks now
                558:
  8FA2          559: STACKSWITCHEDDEVICEPOLL ; come here after switching stacks
  0001          560:           IF    CLOCK
                561:           INCLUDE                        IOCLOCK.ASM
  0000            1:                   IF      IODRIVERBODY
                201:                   FIN     IODRIVERBODY
  0000          202:                   IF      IODRIVERRAM
                228:                   FIN     IODRIVERRAM
                229:
                230:
                562:           FIN
  0001          563:           IF    STORAGEDEMON
                564:           INCLUDE                        IOSTOREDEMON.ASM
  0000            1:                   IF      IODRIVERBODY
                781:                   FIN     IODRIVERBODY
  0001          782:                   IF      IODRIVERPOLL
  0001          783:                   IF      USEDEMONASCLOCK
                784:                   ;LDA    VIAIFR          CHECK FOR CLOCK INTERRUPT FROM VIA
8FA2 B6FF4D     785:                   LDAA    VIAIFR
8FA5 84C0       786:                   ANDA    #%11000000      IRQ + CLOCK DONE ?
                787:                   ;CMPA   #%11000000      ...?
8FA7 81C0       788:                   CMPA    #%11000000
8FA9 2608       789:                   BNE     WDCPOLL1        B/ NO
                790:                   ;STA    VIAIFR          ACKNOWLEDGE THE CLOCK INTERRUPT
8FAB B7FF4D     791:                   STAA    VIAIFR
```

```
                792:                    ;LDA    #1              = 1/60TH SECOND
8FAE 8601       793:                    LDAA    #1
                794:                    ;JMP    SDOS+SDOS:CLOCKTICKED
8FB0 7E8E1B     795:                    JMP     SDOS+SDOS:CLOCKTICKED
  8FB3          796: WDCPOLL1           EQU     *
                797:                    FIN     USEDEMONASCLOCK
                798:                    ;LDA    VIAIFR          ACCEPT DISK INTERRUPT ONLY IF NO CLOCK INTERRUPT PENDING
8FB3 B6FF4D     799:                    LDAA    VIAIFR
8FB6 84D0       800:                    ANDA    #%11010000      MASK TO OBTAIN CLOCK AND DISK INT BITS
                801:                    ;CMPA   #%10010000      DISK ONLY ?
8FB8 8190       802:                    CMPA    #%10010000
8FBA 2603       803:                    BNE     WDCPOLLNEXT     B/ NO
                804:                    ;JMP    WDCINTERRUPT    YES, GO SERVICE DISK INTERRUPT
8FBC 7E8C69     805:                    JMP     WDCINTERRUPT
  8FBF          806: WDCPOLLNEXT        EQU     *
                807:                    FIN     IODRIVERPOLL
  0000          808:                    IF      IODRIVERINIT
                833:                    FIN     IODRIVERINIT
  0000          834:                    IF      IODRIVERRAM
                886:                    FIN     IODRIVERRAM
                887:                    END     ;UNEXPECTED EOF
                565:            FIN
  0001          566:            IF    VIRTUALFLOPPY
                567:            INCLUDE                       IOVFD.ASM
  0000            1:                IF      IODRIVERBODY
                 57:                    FIN     IODRIVERBODY
  0000           58:                    IF      IODRIVERRAM
                127:              FIN     IODRIVERRAM
  0000          128:                 IF      IODRIVERINIT
                172:                 FIN     IODRIVERINIT
  0000          173:              IF      IODRIVERBODY
                410:              FIN     IODRIVERBODY
  0001          411:              IF      IODRIVERPOLL
```

```
  8FBF          413: DISKINTSERVICE
  0002          414:          IF      PERSCI
8FBF B6FFA1     415:          LDAA    PERSCI:PIACB            PerSci Controller
  8FC3          416: PERSCIINTERRUPTMASK EQU *+1 ; set by RESET routines
8FC2 8580       417:          BITA    #$80                    Want interrupt ?
8FC4 2709       418:          BEQ     DISKINTPERSCI.NO
8FC6 CE9017     419:          LDX     #CCB:PERSCI             PerSci interrupted, so absorb the
8FC9 F6FFA3     420:          LDAB    PERSCI:PIADB              interrupt from that controller
8FCC 7E86B5     421:          JMP     DISKINTERRUPT
  8FCF          422: DISKINTPERSCI.NO ; not Persci
                423:          FIN     PERSCI
  0002          424:          IF      DAMFLOPPY
8FCF B6FF81     425:          LDAA    DAMFLOPPY:PIACB         DAM Floppy Controller
  8FD3          426: DAMFLOPPYINTERRUPTMASK equ *+1
8FD2 8580       427:          BITA    #$80                    Want interrupt ?
8FD4 2709       428:          BEQ     DISKINTDAMFLOPPY.NO
8FD6 CE9045     429:          LDX     #CCB:DAMFLOPPY          DAM floppy interrupted, so absorb
8FD9 F6FF83     430:          LDAB    DAMFLOPPY:PIADB           the interrupt from that controller
8FDC 7E86B5     431:          JMP     DISKINTERRUPT
  8FDF          432: DISKINTDAMFLOPPY.NO
                433:          FIN     DAMFLOPPY
                434:          FIN     IODRIVERPOLL
  0000          435:          IF      IODRIVERBODY
                995:          FIN     IODRIVERBODY
                996:
                997:
                568:           FIN
                569:
8FDF FE8FE9     570:           LDX    BADINTERRUPTCOUNT       CAN'T FIGURE OUT WHO IT IS...
8FE2 08         571:           INX                             BUMP CRAZY INTERRRUPT COUNTER
8FE3 FF8FE9     572:           STX    BADINTERRUPTCOUNT
8FE6 7E8E15     573:           JMP    SDOS+SDOS:RTI           AND HOPE IT WENT AWAY !
```

```
  0000          575: IODRIVERPOLL   SET                     0
  0001          576: IODRIVERRAM    SET                     1
                577:
8FE9 0000       578: BADINTERRUPTCOUNT FDB 0 # OF INTERRUPTS FROM UNKNOWN DEVICES
                579:
  0000          580:          IF    SDOSMT
                588:          FIN   SDOSMT
                589:
  0001          590:          IF    CLOCK
                591:          INCLUDE                     IOCLOCK.ASM
  0000            1:                    IF      IODRIVERBODY
                201:                    FIN     IODRIVERBODY
  0001          202:                    IF      IODRIVERRAM
8FEB 01         203: CLOCKDCB           FCB     1               CLOCK'S ALWAYS DONE
8FEC 0000       204:                    FDB     0               LASTER
8FEE 8FFB       205:                    FDB     CLOCKSTR
8FF0 0000       206:                    FDB     NEXTDEVICEDCB
8FF2 8415       207:                    FDB     CLOCKDRIVER
  8FF4          208: DIV60DIVIDEND      EQU     *
8FF4 000000     209: CLOCKBUFFER        FCB     0,0,0
8FF7 00         210: DAY                FCB     0
8FF8 00         211: MONTH              FCB     0
8FF9 00         212: YEAR               FCB     0
8FFA FF         213: CLOCKFRACTION      FCB     -1
8FFB 434C4F43   214: CLOCKSTR           FCC     'CLOCK:'
9001 00         215:                    FCB     0
                216: *
  8FEB          217: NEXTDEVICEDCB      SET     CLOCKDCB
                218: *
  9002          219: TIME$              EQU     *
9002 30303A     220: TIME$:HOURS        FCC     '00:'
9005 30303A     221: TIME$:MINUTES      FCC     '00:'
9008 303020     222: TIME$:SECONDS      FCC     '00 '
  900B          223: DATE$              EQU     *
900B 30302F     224: DATE$:MONTH        FCC     '00/'
900E 30302F     225: DATE$:DAY          FCC     '00/'
9011 3030       226: DATE$:YEAR         FCC     '00'
                227: *
                228:                    FIN     IODRIVERRAM
                229:
                230:
                592:          FIN
  0000          593:          IF    BLACKHOLE
                595:          FIN
  0000          596:          IF    SDLP
                599:          FIN
  0001          600:          IF    VIRTUALFLOPPY
                601:          INCLUDE                     IOVFD.ASM
  0000            1:              IF      IODRIVERBODY
                 57:                FIN     IODRIVERBODY
  0001           58:                IF      IODRIVERRAM
9013 0002        59: DISKINTDCB     RMB     2               address of DCB for interrupt service
9015 0002        60: DISKINTCCB     RMB     2               address of CCB for interrupt service
```

IOVFD.ASM

```
                62: *               Controller Definitions
                63:
  0002          64:                 IF      PERSCI
                65: *               PerSci Controller
                66:
  9017          67: CCB:PERSCI
9017 01         68:                 FCB     1               controller busy: 0 = yes, <>0 = no
9018 FFA0       69:                 FDB     $FFA0           address
901A 00         70:                 FCB     0               timeout counter
901B FF         71:                 FCB     $FF             drive to access
901C FF         72:                 FCB     $FF             cylinder to access
901D FF         73:                 FCB     $FF             last cylinder accessed
901E 86C2       74:                 FDB     DISKINTSTARTPERSCI
9020 7E8864     75:                 JMP     PERSCI:STATUS
9023 7E8891     76:                 JMP     PERSCI:RESET
9026 7E887B     77:                 JMP     PERSCI:ABORT
9029 7E886C     78:                 JMP     PERSCI:RESTORE
902C 7E8895     79:                 JMP     PERSCI:SETSEEK
902F 7E8897     80:                 JMP     PERSCI:SEEK
9032 7E88B3     81:                 JMP     PERSCI:READSECTOR
9035 7E88CF     82:                 JMP     PERSCI:WRITESECTOR
9038 7E88AC     83:                 JMP     PERSCI:VERIFYSECTOR
  903B          84: FDTIMEOUTBLOCK  SET     *
903B 0000       85:                 FDB     NEXTTIMEOUT     timeout block for PerSci floppies
903D 0000       86:                 FDB     0               fuse length
903F 8976       87:                 FDB     PERSCI:TIMEOUT
  0001          88: NTIMEOUTS       SET     NTIMEOUTS+1
  903B          89: NEXTTIMEOUT     SET     FDTIMEOUTBLOCK
  9043          90:                 ORG     FDTIMEOUTBLOCK+TIMEOUT:SIZE
9043 0000       91:                 FDB     0               current DCB
  0002          92:                 IF      DAMFLOPPY
```

```
                94:                     FIN     DAMFLOPPY
                95:                     FIN     PERSCI
  0002          96:                     IF      DAMFLOPPY
                97: *                   DAM Floppy Controller
                98:
  9045          99: CCB:DAMFLOPPY
9045 01        100:                     FCB     1               controller busy: 0 = yes, <>0 = no
9046 FF80      101:                     FDB     $FF80           address
9048 00        102:                     FCB     0               timeout counter
9049 FF        103:                     FCB     $FF             drive to access
904A FF        104:                     FCB     $FF             cylinder to access
904B FF        105:                     FCB     $FF             last cylinder accessed
904C 86C7      106:                     FDB     DISKINTSTARTDAMFLOPPY
904E 7E88EB    107:                     JMP     DAMFLOPPY:STATUS
9051 7E8914    108:                     JMP     DAMFLOPPY:RESET
9054 7E88FF    109:                     JMP     DAMFLOPPY:ABORT
9057 7E88F2    110:                     JMP     DAMFLOPPY:RESTORE
905A 7E8918    111:                     JMP     DAMFLOPPY:SETSEEK
905D 7E891B    112:                     JMP     DAMFLOPPY:SEEK
9060 7E8938    113:                     JMP     DAMFLOPPY:READSECTOR
9063 7E8952    114:                     JMP     DAMFLOPPY:WRITESECTOR
9066 7E892E    115:                     JMP     DAMFLOPPY:VERIFYSECTOR
  9069         116: FDTIMEOUTBLOCK      SET     *
9069 903B      117:                     FDB     NEXTTIMEOUT     timeout block for PerSci floppies
906B 0000      118:                     FDB     0               fuse length
906D 897B      119:                     FDB     DAMFLOPPY:TIMEOUT
  0002         120: NTIMEOUTS           SET     NTIMEOUTS+1
  9069         121: NEXTTIMEOUT         SET     FDTIMEOUTBLOCK
  9071         122:                     ORG     FDTIMEOUTBLOCK+TIMEOUT:SIZE
9071 0000      123:                     FDB     0               current DCB
                124:                     FIN     DAMFLOPPY
```

IOVFD.ASM

```
                126:            INCLUDE IOVFDDCBS.ASM
  0001            1:          IFUND :DCBNUMBER
  0006            2: NDISKDCBS      SET                   NDISKDCBS+PERSCI*WMFORMAT+PERSCI*IBMFORMAT+DAMFLOPPY
  9073            3: NEXTDISKDCB    SET                   *
  0000            4: :DRIVENUMBER   SET                   0
  0000            5: :DCBNUMBER     SET                   0
  0002            6: :DAMFLOPPY     SET                   DAMFLOPPY
  0002            7: :PERSCI        SET                   PERSCI
  9073            8: :HEADCHAIN     SET                   *
  0000            9: :NEXTCHAIN     SET                   0
  0002           10:          IF    PERSCI
  0001           11: :WMFORMAT     SET                   WMFORMAT
  0001           12: :IBMFORMAT    SET                   IBMFORMAT
                 13:          FIN   PERSCI
  0003           14:          ELSE
                 24:          FIN   :DCBNUMBER
```

```
  0002          26:          IF    :DAMFLOPPY
                27: *        DAM Floppy
                28:
  0100          29: :BPS     SET   256                    bytes per sector
  0012          30: :SPT     SET   18                     sectors per track
  0001          31: :TPC     SET   1                      tracks per cylinder
  0028          32: :CYL     SET   40                     cylinders
  0000          33: :DATA    SET   0                      don't complement data
  0000          34: :FIRST   SET   0                      first sector
  9045          35: :CONTROLLER    SET                    CCB:DAMFLOPPY
  0003          36:          ELSEIF                       :PERSCI
                62:          FIN   :DAMFLOPPY
                63:
                64: *        Device Control Block
                65:
  9073          66: :DCB     SET   *
  005B          68:          RPT   FDSIZE                 clear dcb
9073 00         69:          FCB   0
  9073          71:          ORG   :DCB
9073 01         72:          FCB   1
9074 00000000   73:          FDB   0,0,0,FDDRIVER
907C 01000012   74:          FDB   :BPS,:SPT,:TPC,:CYL
  90B6          75:          ORG   :DCB+FDDSTATEJ
90B6 7E875E     76:          JMP   DISKINTUNEXPECTED
  90BB          77:          ORG   :DCB+FDDRIVE
90BB 00FF       78:          FCB   :DRIVENUMBER,$FF
  90BE          79:          ORG   :DCB+FDCOMPLEMENT
90BE 0000       80:          FCB   :DATA,:FIRST
90C0 90730000   81:          FDB   :HEADCHAIN,0
90C4 9045       82:          FDB   :CONTROLLER
  90C6          83:          ORG   :DCB+FDMAPALG
90C6 0001       84:          FDB   1                      set mapalgorithm intially to 1
  90CE          85:          ORG   :DCB+FDMAP
  0012          87:          RPT   :SPT
90CE 00         88:          FCB   *-(:DCB+FDMAP)
  90E0          90: ::       SET   *
  9076          91:          ORG   :DCB+DCB:NAME
9076 90E0       92:          FDB   ::
  90E0          93:          ORG   ::
  0000          94:          IF    :DCBNUMBER>9
                96:          ELSE
90E0 44303A00   97:          FCB   'D,'0+:DCBNUMBER,':,0
                98:          FIN   :DCBNUMBER>9
```

IOVFDDCBS.ASM

```
  0001          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0001          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0001          102:          IF    &:NEXTCHAIN
  0001          103: :NEXTCHAIN     SET                     1
                104:          FIN   &:NEXTCHAIN
  0000          105:          IF    :DRIVENUMBER&%10
                143:          FIN   :DRIVENUMBER&%10
  0004          144:          IF    :PERSCI+:DAMFLOPPY
                145:          INCLUDE                       IOVFDDCBS.ASM
  0000            1:          IFUND :DCBNUMBER
                 14:          ELSE
  90E4           15: ::       SET   *
  9078           16:          ORG   :DCB+DCB:NEXTDCB
9078 90E4        17:          FDB   ::
  90E4           18:          ORG   ::
  0001           19:          IF    :NEXTCHAIN
  90C2           20:          ORG   :DCB+FDNEXTCHAIN
90C2 90E4        21:          FDB   ::
  90E4           22:          ORG   ::
                 23:          FIN   :NEXTCHAIN
                 24:          FIN   :DCBNUMBER
```

```
  0002          26:          IF    :DAMFLOPPY
                27: *        DAM Floppy
                28:
  0100          29: :BPS     SET   256                    bytes per sector
  0012          30: :SPT     SET   18                     sectors per track
  0001          31: :TPC     SET   1                      tracks per cylinder
  0028          32: :CYL     SET   40                     cylinders
  0000          33: :DATA    SET   0                      don't complement data
  0000          34: :FIRST   SET   0                      first sector
  9045          35: :CONTROLLER    SET                    CCB:DAMFLOPPY
  0004          36:          ELSEIF                       :PERSCI
                62:          FIN   :DAMFLOPPY
                63:
                64: *        Device Control Block
                65:
  90E4          66: :DCB     SET   *
  005B          68:          RPT   FDSIZE                 clear dcb
90E4 00         69:          FCB   0
  90E4          71:          ORG   :DCB
90E4 01         72:          FCB   1
90E5 00000000   73:          FDB   0,0,0,FDDRIVER
90ED 01000012   74:          FDB   :BPS,:SPT,:TPC,:CYL
  9127          75:          ORG   :DCB+FDDSTATEJ
9127 7E875E     76:          JMP   DISKINTUNEXPECTED
  912C          77:          ORG   :DCB+FDDRIVE
912C 01FF       78:          FCB   :DRIVENUMBER,$FF
  912F          79:          ORG   :DCB+FDCOMPLEMENT
912F 0000       80:          FCB   :DATA,:FIRST
9131 90730000   81:          FDB   :HEADCHAIN,0
9135 9045       82:          FDB   :CONTROLLER
  9137          83:          ORG   :DCB+FDMAPALG
9137 0001       84:          FDB   1                      set mapalgorithm intially to 1
  913F          85:          ORG   :DCB+FDMAP
  0012          87:          RPT   :SPT
913F 00         88:          FCB   *-(:DCB+FDMAP)
  9151          90: ::       SET   *
  90E7          91:          ORG   :DCB+DCB:NAME
90E7 9151       92:          FDB   ::
  9151          93:          ORG   ::
  0000          94:          IF    :DCBNUMBER>9
                96:          ELSE
9151 44313A00   97:          FCB   'D,'0+:DCBNUMBER,':,0
                98:          FIN   :DCBNUMBER>9
```

IOVFDDCBS.ASM

```
  0002          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0002          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0000          102:          IF    &:NEXTCHAIN
                104:          FIN   &:NEXTCHAIN
  0002          105:          IF    :DRIVENUMBER&%10
  0002          106:          IF    :DAMFLOPPY
  0000          107: :DAMFLOPPY     SET                     :DAMFLOPPY-2
  0001          108:          IF    :DAMFLOPPY<1
  0000          109: :DRIVENUMBER   SET                     0
                110:          FIN   :DAMFLOPPY<1
  0001          111:          IF    :NEXTCHAIN
  0000          112: :NEXTCHAIN     SET                     0
                113:          FIN   :NEXTCHAIN
  9155          114: :HEADCHAIN     SET                     *
  0005          115:          ELSEIF                        :PERSCI
                142:          FIN   :DAMFLOPPY
                143:          FIN   :DRIVENUMBER&%10
  0002          144:          IF    :PERSCI+:DAMFLOPPY
                145:          INCLUDE                       IOVFDDCBS.ASM
  0000            1:          IFUND :DCBNUMBER
                 14:          ELSE
  9155           15: ::       SET   *
  90E9           16:          ORG   :DCB+DCB:NEXTDCB
90E9 9155        17:          FDB   ::
  9155           18:          ORG   ::
  0000           19:          IF    :NEXTCHAIN
                 23:          FIN   :NEXTCHAIN
                 24:          FIN   :DCBNUMBER
```

```
  0000          26:          IF    :DAMFLOPPY
  0002          36:          ELSEIF                        :PERSCI
  0001          37:          IF    :IBMFORMAT
                38: *        IBM Disk
                39:
  0080          40: :BPS     SET   128                     bytes per sector
  001A          41: :SPT     SET   26                      sectors per track
  0001          42: :TPC     SET   1                       tracks per cylinder
  004D          43: :CYL     SET   77                      cylinders
  0001          44: :DATA    SET   1                       complement data
  0001          45: :FIRST   SET   1                       first sector
  9017          46: :CONTROLLER    SET                     CCB:PERSCI
  0006          47:          ELSEIF                        :WMFORMAT
                59:          FIN   :IBMFORMAT
  0005          60:          ELSE
                62:          FIN   :DAMFLOPPY
                63:
                64: *        Device Control Block
                65:
  9155          66: :DCB     SET   *
  005B          68:          RPT   FDSIZE                  clear dcb
9155 00         69:          FCB   0
  9155          71:          ORG   :DCB
9155 01         72:          FCB   1
9156 00000000   73:          FDB   0,0,0,FDDRIVER
915E 0080001A   74:          FDB   :BPS,:SPT,:TPC,:CYL
  9198          75:          ORG   :DCB+FDDSTATEJ
9198 7E875E     76:          JMP   DISKINTUNEXPECTED
  919D          77:          ORG   :DCB+FDDRIVE
919D 00FF       78:          FCB   :DRIVENUMBER,$FF
  91A0          79:          ORG   :DCB+FDCOMPLEMENT
91A0 0101       80:          FCB   :DATA,:FIRST
91A2 91550000   81:          FDB   :HEADCHAIN,0
91A6 9017       82:          FDB   :CONTROLLER
  91A8          83:          ORG   :DCB+FDMAPALG
91A8 0001       84:          FDB   1                       set mapalgorithm intially to 1
  91B0          85:          ORG   :DCB+FDMAP
  001A          87:          RPT   :SPT
91B0 00         88:          FCB   *-(:DCB+FDMAP)
  91CA          90: ::       SET   *
  9158          91:          ORG   :DCB+DCB:NAME
9158 91CA       92:          FDB   ::
  91CA          93:          ORG   ::
  0000          94:          IF    :DCBNUMBER>9
                96:          ELSE
91CA 44323A00   97:          FCB   'D,'0+:DCBNUMBER,':,0
                98:          FIN   :DCBNUMBER>9
```

IOVFDDCBS.ASM

```
  0003          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0001          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0001          102:          IF    &:NEXTCHAIN
  0001          103: :NEXTCHAIN     SET                     1
                104:          FIN   &:NEXTCHAIN
  0000          105:          IF    :DRIVENUMBER&%10
                143:          FIN   :DRIVENUMBER&%10
  0002          144:          IF    :PERSCI+:DAMFLOPPY
                145:          INCLUDE                       IOVFDDCBS.ASM
  0000            1:          IFUND :DCBNUMBER
                 14:          ELSE
  91CE           15: ::       SET   *
  915A           16:          ORG   :DCB+DCB:NEXTDCB
915A 91CE        17:          FDB   ::
  91CE           18:          ORG   ::
  0001           19:          IF    :NEXTCHAIN
  91A4           20:          ORG   :DCB+FDNEXTCHAIN
91A4 91CE        21:          FDB   ::
  91CE           22:          ORG   ::
                 23:          FIN   :NEXTCHAIN
                 24:          FIN   :DCBNUMBER
```

```
  0000          26:          IF    :DAMFLOPPY
  0002          36:          ELSEIF                        :PERSCI
  0001          37:          IF    :IBMFORMAT
                38: *        IBM Disk
                39:
  0080          40: :BPS     SET   128                     bytes per sector
  001A          41: :SPT     SET   26                      sectors per track
  0001          42: :TPC     SET   1                       tracks per cylinder
  004D          43: :CYL     SET   77                      cylinders
  0001          44: :DATA    SET   1                       complement data
  0001          45: :FIRST   SET   1                       first sector
  9017          46: :CONTROLLER    SET                     CCB:PERSCI
  0007          47:          ELSEIF                        :WMFORMAT
                59:          FIN   :IBMFORMAT
  0006          60:          ELSE
                62:          FIN   :DAMFLOPPY
                63:
                64: *        Device Control Block
                65:
  91CE          66: :DCB     SET   *
  005B          68:          RPT   FDSIZE                  clear dcb
91CE 00         69:          FCB   0
  91CE          71:          ORG   :DCB
91CE 01         72:          FCB   1
91CF 00000000   73:          FDB   0,0,0,FDDRIVER
91D7 0080001A   74:          FDB   :BPS,:SPT,:TPC,:CYL
  9211          75:          ORG   :DCB+FDDSTATEJ
9211 7E875E     76:          JMP   DISKINTUNEXPECTED
  9216          77:          ORG   :DCB+FDDRIVE
9216 01FF       78:          FCB   :DRIVENUMBER,$FF
  9219          79:          ORG   :DCB+FDCOMPLEMENT
9219 0101       80:          FCB   :DATA,:FIRST
921B 91550000   81:          FDB   :HEADCHAIN,0
921F 9017       82:          FDB   :CONTROLLER
  9221          83:          ORG   :DCB+FDMAPALG
9221 0001       84:          FDB   1                       set mapalgorithm intially to 1
  9229          85:          ORG   :DCB+FDMAP
  001A          87:          RPT   :SPT
9229 00         88:          FCB   *-(:DCB+FDMAP)
  9243          90: ::       SET   *
  91D1          91:          ORG   :DCB+DCB:NAME
91D1 9243       92:          FDB   ::
  9243          93:          ORG   ::
  0000          94:          IF    :DCBNUMBER>9
                96:          ELSE
9243 44333A00   97:          FCB   'D,'0+:DCBNUMBER,':,0
                98:          FIN   :DCBNUMBER>9
```

IOVFDDCBS.ASM

```
  0004          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0002          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0000          102:          IF    &:NEXTCHAIN
                104:          FIN   &:NEXTCHAIN
  0002          105:          IF    :DRIVENUMBER&%10
  0000          106:          IF    :DAMFLOPPY
  0002          115:          ELSEIF                        :PERSCI
  0001          116:          IF    :IBMFORMAT
  0000          117: :IBMFORMAT     SET                     0
  0001          118:          IF    :WMFORMAT
  0000          119: :DRIVENUMBER   SET                     :DRIVENUMBER-2
  0009          120:          ELSE
                130:          FIN   :WMFORMAT
  0008          131:          ELSEIF                        :WMFORMAT
                141:          FIN   :IBMFORMAT
                142:          FIN   :DAMFLOPPY
                143:          FIN   :DRIVENUMBER&%10
  0002          144:          IF    :PERSCI+:DAMFLOPPY
                145:          INCLUDE                       IOVFDDCBS.ASM
  0000            1:          IFUND :DCBNUMBER
                 14:          ELSE
  9247           15: ::       SET   *
  91D3           16:          ORG   :DCB+DCB:NEXTDCB
91D3 9247        17:          FDB   ::
  9247           18:          ORG   ::
  0001           19:          IF    :NEXTCHAIN
  921D           20:          ORG   :DCB+FDNEXTCHAIN
921D 9247        21:          FDB   ::
  9247           22:          ORG   ::
                 23:          FIN   :NEXTCHAIN
                 24:          FIN   :DCBNUMBER
```

```
  0000          26:          IF    :DAMFLOPPY
  0002          36:          ELSEIF                        :PERSCI
  0000          37:          IF    :IBMFORMAT
  0001          47:          ELSEIF                        :WMFORMAT
                48: *        Wavemate Disk
                49:
  0100          50: :BPS     SET   256                     bytes per sector
  0010          51: :SPT     SET   16                      sectors per track
  0001          52: :TPC     SET   1                       tracks per cylinder
  004D          53: :CYL     SET   77                      cylinders
  0000          54: :DATA    SET   0                       don't complement data
  0000          55: :FIRST   SET   0                       first sector
  9017          56: :CONTROLLER    SET                     CCB:PERSCI
  0008          57:          ELSE
                59:          FIN   :IBMFORMAT
  0007          60:          ELSE
                62:          FIN   :DAMFLOPPY
                63:
                64: *        Device Control Block
                65:
  9247          66: :DCB     SET   *
  005B          68:          RPT   FDSIZE                  clear dcb
9247 00         69:          FCB   0
  9247          71:          ORG   :DCB
9247 01         72:          FCB   1
9248 00000000   73:          FDB   0,0,0,FDDRIVER
9250 01000010   74:          FDB   :BPS,:SPT,:TPC,:CYL
  928A          75:          ORG   :DCB+FDDSTATEJ
928A 7E875E     76:          JMP   DISKINTUNEXPECTED
  928F          77:          ORG   :DCB+FDDRIVE
928F 00FF       78:          FCB   :DRIVENUMBER,$FF
  9292          79:          ORG   :DCB+FDCOMPLEMENT
9292 0000       80:          FCB   :DATA,:FIRST
9294 91550000   81:          FDB   :HEADCHAIN,0
9298 9017       82:          FDB   :CONTROLLER
  929A          83:          ORG   :DCB+FDMAPALG
929A 0001       84:          FDB   1                       set mapalgorithm intially to 1
  92A2          85:          ORG   :DCB+FDMAP
  0010          87:          RPT   :SPT
92A2 00         88:          FCB   *-(:DCB+FDMAP)
  92B2          90: ::       SET   *
  924A          91:          ORG   :DCB+DCB:NAME
924A 92B2       92:          FDB   ::
  92B2          93:          ORG   ::
  0000          94:          IF    :DCBNUMBER>9
                96:          ELSE
92B2 44343A00   97:          FCB   'D,'0+:DCBNUMBER,':,0
                98:          FIN   :DCBNUMBER>9
```

```
  0005          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0001          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0000          102:          IF    &:NEXTCHAIN
                104:          FIN   &:NEXTCHAIN
  0000          105:          IF    :DRIVENUMBER&%10
                143:          FIN   :DRIVENUMBER&%10
  0002          144:          IF    :PERSCI+:DAMFLOPPY
                145:          INCLUDE                       IOVFDDCBS.ASM
  0000            1:          IFUND :DCBNUMBER
                 14:          ELSE
  92B6           15: ::       SET   *
  924C           16:          ORG   :DCB+DCB:NEXTDCB
924C 92B6        17:          FDB   ::
  92B6           18:          ORG   ::
  0001           19:          IF    :NEXTCHAIN
  9296           20:          ORG   :DCB+FDNEXTCHAIN
9296 92B6        21:          FDB   ::
  92B6           22:          ORG   ::
                 23:          FIN   :NEXTCHAIN
                 24:          FIN   :DCBNUMBER
```

```
  0000          26:           IF    :DAMFLOPPY
  0002          36:           ELSEIF                        :PERSCI
  0000          37:           IF    :IBMFORMAT
  0001          47:           ELSEIF                        :WMFORMAT
                48: *         Wavemate Disk
                49:
  0100          50: :BPS      SET   256                     bytes per sector
  0010          51: :SPT      SET   16                      sectors per track
  0001          52: :TPC      SET   1                       tracks per cylinder
  004D          53: :CYL      SET   77                      cylinders
  0000          54: :DATA     SET   0                       don't complement data
  0000          55: :FIRST    SET   0                       first sector
  9017          56: :CONTROLLER     SET                     CCB:PERSCI
  0009          57:           ELSE
                59:           FIN   :IBMFORMAT
  0008          60:           ELSE
                62:           FIN   :DAMFLOPPY
                63:
                64: *         Device Control Block
                65:
  92B6          66: :DCB      SET   *
  005B          68:           RPT   FDSIZE                  clear dcb
92B6 00         69:           FCB   0
  92B6          71:           ORG   :DCB
92B6 01         72:           FCB   1
92B7 00000000   73:           FDB   0,0,0,FDDRIVER
92BF 01000010   74:           FDB   :BPS,:SPT,:TPC,:CYL
  92F9          75:           ORG   :DCB+FDDSTATEJ
92F9 7E875E     76:           JMP   DISKINTUNEXPECTED
  92FE          77:           ORG   :DCB+FDDRIVE
92FE 01FF       78:           FCB   :DRIVENUMBER,$FF
  9301          79:           ORG   :DCB+FDCOMPLEMENT
9301 0000       80:           FCB   :DATA,:FIRST
9303 91550000   81:           FDB   :HEADCHAIN,0
9307 9017       82:           FDB   :CONTROLLER
  9309          83:           ORG   :DCB+FDMAPALG
9309 0001       84:           FDB   1                       set mapalgorithm intially to 1
  9311          85:           ORG   :DCB+FDMAP
  0010          87:           RPT   :SPT
9311 00         88:           FCB   *-(:DCB+FDMAP)
  9321          90: ::        SET   *
  92B9          91:           ORG   :DCB+DCB:NAME
92B9 9321       92:           FDB   ::
  9321          93:           ORG   ::
  0000          94:           IF    :DCBNUMBER>9
                96:           ELSE
9321 44353A00   97:           FCB   'D,'0+:DCBNUMBER,':,0
                98:           FIN   :DCBNUMBER>9
```

```
  0006          100: :DCBNUMBER     SET                     :DCBNUMBER+1
  0002          101: :DRIVENUMBER   SET                     :DRIVENUMBER+1
  0000          102:          IF    &:NEXTCHAIN
                104:          FIN   &:NEXTCHAIN
  0002          105:          IF    :DRIVENUMBER&%10
  0000          106:          IF    :DAMFLOPPY
  0002          115:          ELSEIF                        :PERSCI
  0000          116:          IF    :IBMFORMAT
  0001          131:          ELSEIF                        :WMFORMAT
  0000          132: :PERSCI  SET   :PERSCI-2
  0000          133:          IF    :PERSCI
                140:          FIN   :PERSCI
                141:          FIN   :IBMFORMAT
                142:          FIN   :DAMFLOPPY
                143:          FIN   :DRIVENUMBER&%10
  0000          144:          IF    :PERSCI+:DAMFLOPPY
                146:          FIN   :PERSCI+:DAMFLOPPY
                147:
                148:
                146:          FIN   :PERSCI+:DAMFLOPPY
                147:
                148:
                146:          FIN   :PERSCI+:DAMFLOPPY
                147:
                148:
                146:          FIN   :PERSCI+:DAMFLOPPY
                147:
                148:
                146:          FIN   :PERSCI+:DAMFLOPPY
                147:
                148:
                127:            FIN     IODRIVERRAM
  0000          128:               IF      IODRIVERINIT
                172:               FIN     IODRIVERINIT
  0000          173:             IF      IODRIVERBODY
                410:            FIN      IODRIVERBODY
  0000          411:            IF       IODRIVERPOLL
                434:           FIN    IODRIVERPOLL
  0000          435:           IF     IODRIVERBODY
                995:           FIN    IODRIVERBODY
                996:
                997:
                602:          FIN
  0001          603:          IF    STORAGEDEMON
                604:          INCLUDE                       IOSTOREDEMON.ASM
  0000            1:                    IF      IODRIVERBODY
                781:                    FIN     IODRIVERBODY
  0000          782:                    IF      IODRIVERPOLL
                807:                    FIN     IODRIVERPOLL
  0000          808:                    IF      IODRIVERINIT
                833:                    FIN     IODRIVERINIT
  0001          834:                    IF      IODRIVERRAM
```

```
                                *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
                                *** STORAGE DEMON WORKING RAM ***
9325 01         836: WDCINTERFACE       FCB     1               ; WDC CONTROLLER IS AVAILABLE
9326 0000       837: WDCDCBPOINTER      FDB     0               ; CURRENT UNIT IN USE BY INTERRUPT ROUTINES
9328 8C66       838: WDCCONTINUEPC      FDB     WDCINTUNEXPECTED ; WHERE TO GO WHEN TRANSFER DONE INTERRUPT
932A 00         839: WDCCOUNT           FCB     0               ; COUNTS # OF 8 BYTE BLOCKS TO XFER TO 7710
932B 0000       840: WDCPOINTER         FDB     0               ; POINTER TO NEXT BLOCK OF 8 BYTES TO XFER
932D 00         841: WDCRETRYCNT        FCB     0               ; USED TO COUNT # OF READ/WRITE ATTEMPTS
                842:
932E 01         843: WDC1DCB            FCB     1               DCB:DONEFLAG
932F 0000       844:                    FDB     0               DCB:LASTERROR
9331 9372       845:                    FDB     WDC1STR
9333 9073       846:                    FDB     NEXTDISKDCB     DCB:NEXT
9335 89C4       847:                    FDB     WDCDRIVER       DCB:DRIVER
9337 0200       848:                    FDB     WDCNBPS
9339 4E34       849:                    FDB     WDCNSPT         DSKINFO:NSPT
933B 0001       850:                    FDB     WDCNTPC         DSKINFO:NTPC
933D 0001       851:                    FDB     WDCNCYL         DSKINFO:NCYL
  0031          852:                    RPT     WDC1DCB+DSKINFO:SIZE-*
933F 00         853:                    FCB     0
9370 00         854:                    FCB     0               WDCREADWRITE: 2 IS READ 3 IS WRITE ETC.
9371 01         855:                    FCB     1               DRIVE SELECT 1
9372 5744313A   856: WDC1STR            FCC     'WD1:'
9376 00         857:                    FCB     0
  932E          858: NEXTDISKDCB        SET     WDC1DCB
  0007          859: NDISKDCBS          SET     NDISKDCBS+1
                860:
9377 01         861: WDC0DCB            FCB     1               DCB:DONEFLAG
9378 0000       862:                    FDB     0               DCB:LASTERROR
937A 93BB       863:                    FDB     WDC0STR
937C 932E       864:                    FDB     NEXTDISKDCB     DCB:NEXT
937E 89C4       865:                    FDB     WDCDRIVER       DCB:DRIVER
9380 0200       866:                    FDB     WDCNBPS
9382 4E34       867:                    FDB     WDCNSPT         DSKINFO:NSPT
9384 0001       868:                    FDB     WDCNTPC         DSKINFO:NTPC
9386 0001       869:                    FDB     WDCNCYL         DSKINFO:NCYL
  0031          870:                    RPT     WDC0DCB+DSKINFO:SIZE-*
9388 00         871:                    FCB     0
93B9 00         872:                    FCB     0               WDCREADWRITE: CONTAINS DESIRED DISK OPCODE
93BA 00         873:                    FCB     0               DRIVE 0
93BB 5744303A   874: WDC0STR            FCC     'WD0:'
93BF 00         875:                    FCB     0
  9377          876: NEXTDISKDCB        SET     WDC0DCB
  0008          877: NDISKDCBS          SET     NDISKDCBS+1
                878:
93C0 9069       879: WDCTIMEOUTBLOCK    FDB     NEXTTIMEOUT
93C2 0000       880: WDCTIMEOUTCOUNT    FDB     0
93C4 8CF2       881:                    FDB     WDCTIMEDOUT
                882:
                883: *
  93C0          884: NEXTTIMEOUT        SET     WDCTIMEOUTBLOCK
  0003          885: NTIMEOUTS          SET     NTIMEOUTS+1
                886:                    FIN     IODRIVERRAM
                887:                    END     ;UNEXPECTED EOF
                605:            FIN
                606:            INCLUDE                 IOVTCONFIG.ASM
  0000            1:                    if      iodriverbody
```

IOVTCONFIG.ASM

```
                30:                     fin     iodriverbody
  0000          31:                     if      iodriverinit
                39:                     fin     iodriverinit
  0000          40:                     if      iodriverbody
                57:                     fin     iodriverbody
  0000          58:                     if      iodriverinit
                66:                     fin     iodriverinit
  0000          67:                     if      iodriverbody
                84:                     fin     iodriverbody
  0000          85:                     if      iodriverinit
                93:                     fin     iodriverinit
  0000          94:                     if      iodriverbody
               111:                     fin     iodriverbody
  0000         112:                     if      iodriverpoll
               193:                     fin     iodriverpoll
  0001         194:                     if      iodriverram
  0009         195: ntimeouts           set       6+ntimeouts
```

```
  93C6          197: ttybuffers
  0001          198:                    ifund   outbufsize:$FFC0
  0050          199: outbufsize:$FFC0   equ     80
                200:                    fin
  0001          201:                    ifund   linebufsize:$FFC0
  0064          202: linebufsize:$FFC0  equ     100
                203:                    fin
  0001          204:                    ifund   inbufsize:$FFC0
  0050          205: inbufsize:$FFC0    equ     80
                206:                    fin
                207: ; inbufsize should be less than linebufsize, in order to
                208: ; avoid too long typed-ahead line
  93C6          209: outbuf:$FFC0
93C6 0050       210:                    rmb     outbufsize:$FFC0
  9416          211: inbuf:$FFC0
9416 0050       212:                    rmb     inbufsize:$FFC0
  9466          213: linebuf:$FFC0
9466 0064       214:                    rmb     linebufsize:$FFC0
  0001          215:                    ifund   outbufsize:$FFC4
  0050          216: outbufsize:$FFC4   equ     80
                217:                    fin
  0000          218:                    ifund   linebufsize:$FFC4
                220:                    fin
  0000          221:                    ifund   inbufsize:$FFC4
                223:                    fin
                224: ; inbufsize should be less than linebufsize, in order to
                225: ; avoid too long typed-ahead line
  94CA          226: outbuf:$FFC4
94CA 0050       227:                    rmb     outbufsize:$FFC4
  951A          228: inbuf:$FFC4
951A 0000       229:                    rmb     inbufsize:$FFC4
  951A          230: linebuf:$FFC4
951A 0000       231:                    rmb     linebufsize:$FFC4
  0001          232:                    ifund   outbufsize:$FFC8
  0050          233: outbufsize:$FFC8   equ     80
                234:                    fin
  0001          235:                    ifund   linebufsize:$FFC8
  0064          236: linebufsize:$FFC8  equ     100
                237:                    fin
  0001          238:                    ifund   inbufsize:$FFC8
  0050          239: inbufsize:$FFC8    equ     80
                240:                    fin
                241: ; inbufsize should be less than linebufsize, in order to
                242: ; avoid too long typed-ahead line
  951A          243: outbuf:$FFC8
951A 0050       244:                    rmb     outbufsize:$FFC8
  956A          245: inbuf:$FFC8
956A 0050       246:                    rmb     inbufsize:$FFC8
  95BA          247: linebuf:$FFC8
95BA 0064       248:                    rmb     linebufsize:$FFC8
```

IOVTCONFIG.ASM

```
  961E          250: dcbname:$FFC0
961E 434F4E53   251:                 fcc     /CONSOLE:/
9626 00         252:                 fcb     0
  0001          253:                 ifund   ttydcb
  9627          254: ttydcb
                255:                 fin     ttydcb
  9627          256: dcb:$FFC0
  00FD          257:                 rpt     dcb:vtsize
9627 00         258:                 fcb     0
  962A          259:                 org     dcb:$FFC0+dcb:name
962A 961E       260:                 fdb     dcbname:$FFC0
962C 9763       261:                 fdb     dcb:$FFC4
962E BDE2       262:                 fdb     sdos+sdos:vtdispatch
  96DC          263:                 org     dcb:$FFC0+dcb:reset
96DC 7E9E4B     264:                 jmp     reset:$FFC0
96DF 0C39       265:                 okrts                   dump nothing
  0001          266:                 if      m6800!m6801
96E1 01         267:                 nop
                268:                 fin
96E2 7E8D13     269:                 jmp     ilputdev:$FFC0
96E5 7E8D1C     270:                 jmp     ilgetdev:$FFC0
96E8 7E8EDB     271:                 jmp     illdeviceop     no extra control calls defined
96EB 7E8EDB     272:                 jmp     illdeviceop     no extra status calls defined
96EE 7E8D25     273:                 jmp     tlcheckready:$FFC0 go check for acia ready
  96C8          274:                 org     dcb:$FFC0+dcb:outputtoblk
  0001          275:                 ifund   ttytimeouts
  96C8          276: ttytimeouts
                277:                 fin     ttytimeouts
  96C8          278: outputtoblk:$FFC0
96C8 96D0       279:                 fdb     inputtoblk:$FFC0
96CA 0000       280:                 fdb     0
96CC BDD9       281:                 fdb     sdos+sdos:vtoutputto
96CE 9627       282:                 fdb     dcb:$FFC0
  96D0          283:                 org     dcb:$FFC0+dcb:inputtoblk
  96D0          284: inputtoblk:$FFC0
96D0 9804       285:                 fdb     outputtoblk:$FFC4
96D2 0000       286:                 fdb     0
96D4 BDD6       287:                 fdb     sdos+sdos:vtinputto
96D6 9627       288:                 fdb     dcb:$FFC0
  96D8          289:                 org     dcb:$FFC0+dcb:tcb
96D8 9724       290:                 fdb     tcb:$FFC0
  96DA          291:                 org     dcb:$FFC0+dcb:taskstack
96DA 9756       292:                 fdb     tcbstack:$FFC0
  9674          293:                 org     dcb:$FFC0+dcb:ringinbase
9674 9416       294:                 fdb     inbuf:$FFC0
9676 0050       295:                 fdb     inbufsize:$FFC0
  9680          296:                 org     dcb:$FFC0+dcb:ringoutbase
9680 93C6       297:                 fdb     outbuf:$FFC0
9682 0050       298:                 fdb     outbufsize:$FFC0
  9684          299:                 org     dcb:$FFC0+dcb:ringoutthreshold
9684 08         300:                 fcb     outbufsize:$FFC0//10
  968E          301:                 org     dcb:$FFC0+dcb:linebuf
968E 9466       302:                 fdb     linebuf:$FFC0
  96A1          303:                 org     dcb:$FFC0+dcb:linebuflen
96A1 64         304:                 fcb     linebufsize:$FFC0
```

```
  96F1          305:                    org     dcb:$FFC0+dcb:clearin
96F1 7EBDB8     306:                    jmp     sdos+sdos:vtclearin
96F4 7EBDBB     307:                    jmp     sdos+sdos:vtclearout
96F7 7EBDBE     308:                    jmp     sdos+sdos:vttlputbuf
96FA 7EBDC1     309:                    jmp     sdos+sdos:vttlgetbuf
96FD 7EBDC4     310:                    jmp     sdos+sdos:vtilputbuf
9700 7EBDC7     311:                    jmp     sdos+sdos:vtilgetbuf
  964F          312:                    org     dcb:$FFC0+dcb:profile
964F 01         313:                    fcb     profilenum.MALVT profile name
  9627          314:                    org     dcb:$FFC0+dcb:doneflag
9627 01         315:                    fcb     1               device not busy
  963A          316:                    org     dcb:$FFC0+dcb:oilquiescent
963A 01         317:                    fcb     1               interrupt not expected
  9724          318:                    org     dcb:$FFC0+dcb:vtsize
  0001          319:                    ifund   ttytcb
  9724          320: ttytcb
                321:                    fin     ttytcb
  9724          322: tcb:$FFC0
9724 9860       323:                    fdb     tcb:$FFC4
9726 9756       324:                    fdb     tcbstack:$FFC0
9728 00000000   325:                    fdb     0,0,0,0,0,0
  9732          326:                    org     tcb:$FFC0+tcb:scratchpad+dcbpointer
9732 9627       327:                    fdb     dcb:$FFC0
  9734          328:                    org     tcb:$FFC0+tcb:size
9734 0022       329:                    rmb     env:minstack*2-env:size
  9756          330: tcbstack:$FFC0
  0008          331:                    rpt     env:size
9756 00         332:                    fcb     0
  975D          333:                    org     *-env:cc
  975D          334: stack:$FFC0
  975C          335:                    org     tcbstack:$FFC0+env:p
975C BDCA       336:                    fdb     sdos+sdos:vtedittask
  9757          337:                    org     tcbstack:$FFC0+env:cc
9757 00         338:                    fcb     $80*m6809       set 'E' flag for 6809
  975E          339:                    org     stack:$FFC0+env:cc
  975E          340: dcbname:$FFC4
975E 4C50543A   341:                    fcc     /LPT:/
9762 00         342:                    fcb     0
  0000          343:                    ifund   ttydcb
                345:                    fin     ttydcb
  9763          346: dcb:$FFC4
  00FD          347:                    rpt     dcb:vtsize
9763 00         348:                    fcb     0
  9766          349:                    org     dcb:$FFC4+dcb:name
9766 975E       350:                    fdb     dcbname:$FFC4
9768 98A2       351:                    fdb     dcb:$FFC8
976A BDE2       352:                    fdb     sdos+sdos:vtdispatch
  9818          353:                    org     dcb:$FFC4+dcb:reset
9818 7E9E5A     354:                    jmp     reset:$FFC4
981B 0C39       355:                    okrts                   dump nothing
  0001          356:                    if      m6800!m6801
981D 01         357:                    nop
                358:                    fin
981E 7E8D2C     359:                    jmp     ilputdev:$FFC4
9821 7E8D35     360:                    jmp     ilgetdev:$FFC4
```

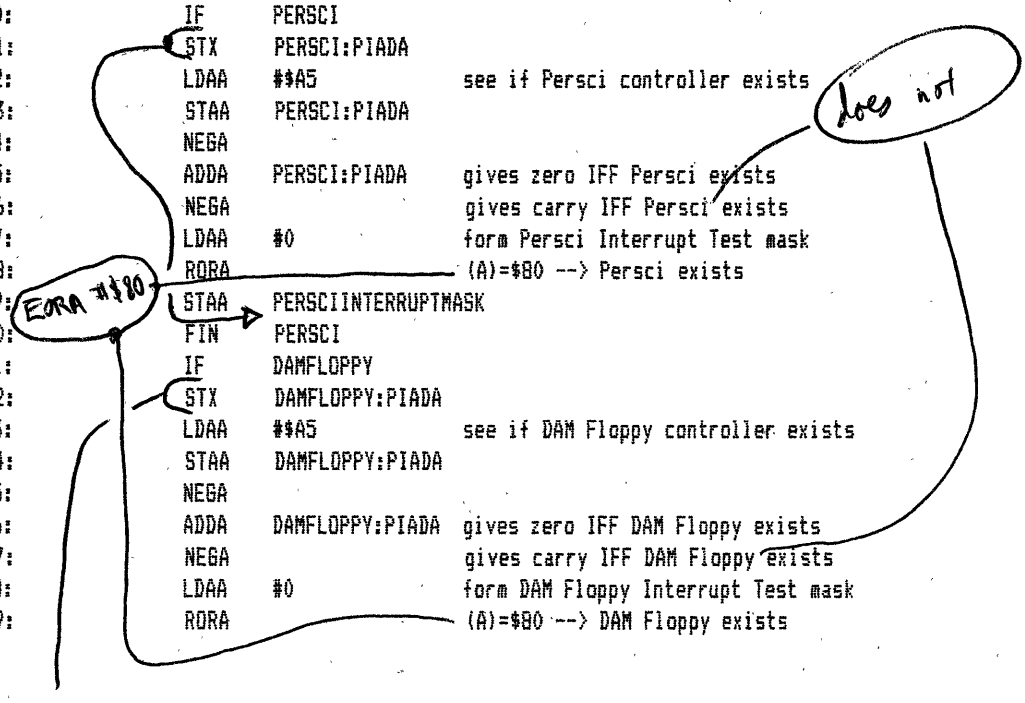
```
9824 7E8EDB     361:                    jmp     illdeviceop     no extra control calls defined
9827 7E8EDB     362:                    jmp     illdeviceop     no extra status calls defined
982A 7E8D3E     363:                    jmp     tlcheckready:$FFC4 go check for acia ready
  9804          364:                    org     dcb:$FFC4+dcb:outputtoblk
  0000          365:                    ifund   ttytimeouts
                367:                    fin     ttytimeouts
  9804          368: outputtoblk:$FFC4
9804 980C       369:                    fdb     inputtoblk:$FFC4
9806 0000       370:                    fdb     0
9808 BDD9       371:                    fdb     sdos+sdos:vtoutputto
980A 9763       372:                    fdb     dcb:$FFC4
  980C          373:                    org     dcb:$FFC4+dcb:inputtoblk
  980C          374: inputtoblk:$FFC4
980C 9943       375:                    fdb     outputtoblk:$FFC8
980E 0000       376:                    fdb     0
9810 BDD6       377:                    fdb     sdos+sdos:vtinputto
9812 9763       378:                    fdb     dcb:$FFC4
  9814          379:                    org     dcb:$FFC4+dcb:tcb
9814 9860       380:                    fdb     tcb:$FFC4
  9816          381:                    org     dcb:$FFC4+dcb:taskstack
9816 9892       382:                    fdb     tcbstack:$FFC4
  97B0          383:                    org     dcb:$FFC4+dcb:ringinbase
97B0 951A       384:                    fdb     inbuf:$FFC4
97B2 0000       385:                    fdb     inbufsize:$FFC4
  97BC          386:                    org     dcb:$FFC4+dcb:ringoutbase
97BC 94CA       387:                    fdb     outbuf:$FFC4
97BE 0050       388:                    fdb     outbufsize:$FFC4
  97C0          389:                    org     dcb:$FFC4+dcb:ringoutthreshold
97C0 08         390:                    fcb     outbufsize:$FFC4//10
  97CA          391:                    org     dcb:$FFC4+dcb:linebuf
97CA 951A       392:                    fdb     linebuf:$FFC4
  97DD          393:                    org     dcb:$FFC4+dcb:linebuflen
97DD 00         394:                    fcb     linebufsize:$FFC4
  982D          395:                    org     dcb:$FFC4+dcb:clearin
982D 7EBDB8     396:                    jmp     sdos+sdos:vtclearin
9830 7EBDBB     397:                    jmp     sdos+sdos:vtclearout
9833 7EBDBE     398:                    jmp     sdos+sdos:vttlputbuf
9836 7EBDC1     399:                    jmp     sdos+sdos:vttlgetbuf
9839 7EBDC4     400:                    jmp     sdos+sdos:vtilputbuf
983C 7EBDC7     401:                    jmp     sdos+sdos:vtilgetbuf
  978B          402:                    org     dcb:$FFC4+dcb:profile
978B 09         403:                    fcb     profilenum.MALLPT profile name
  9763          404:                    org     dcb:$FFC4+dcb:doneflag
9763 01         405:                    fcb     1               device not busy
  9776          406:                    org     dcb:$FFC4+dcb:oilquiescent
9776 01         407:                    fcb     1               interrupt not expected
  9860          408:                    org     dcb:$FFC4+dcb:vtsize
  0000          409:                    ifund   ttytcb
                411:                    fin     ttytcb
  9860          412: tcb:$FFC4
9860 999F       413:                    fdb     tcb:$FFC8
9862 9892       414:                    fdb     tcbstack:$FFC4
9864 00000000   415:                    fdb     0,0,0,0,0,0
  986E          416:                    org     tcb:$FFC4+tcb:scratchpad+dcbpointer
986E 9763       417:                    fdb     dcb:$FFC4
```

```
  9870          418:                    org     tcb:$FFC4+tcb:size
9870 0022       419:                    rmb     env:minstack*2-env:size
  9892          420: tcbstack:$FFC4
  0008          421:                    rpt     env:size
9892 00         422:                    fcb     0
  9899          423:                    org     *-env:cc
  9899          424: stack:$FFC4
  9898          425:                    org     tcbstack:$FFC4+env:p
9898 BDCA       426:                    fdb     sdos+sdos:vteditttask
  9893          427:                    org     tcbstack:$FFC4+env:cc
9893 00         428:                    fcb     $80*m6809       set 'E' flag for 6809
  989A          429:                    org     stack:$FFC4+env:cc
  989A          430: dcbname:$FFC8
989A 48595459   431:                    fcc     /HYTYPE:/
98A1 00         432:                    fcb     0
  0000          433:                    ifund   ttydcb
                435:                    fin     ttydcb
  98A2          436: dcb:$FFC8
  00FD          437:                    rpt     dcb:vtsize
98A2 00         438:                    fcb     0
  98A5          439:                    org     dcb:$FFC8+dcb:name
98A5 989A       440:                    fdb     dcbname:$FFC8
98A7 8FEB       441:                    fdb     nextdevicedcb
98A9 BDE2       442:                    fdb     sdos+sdos:vtdispatch
  9957          443:                    org     dcb:$FFC8+dcb:reset
9957 7E9E69     444:                    jmp     reset:$FFC8
995A 0C39       445:                    okrts                   dump nothing
  0001          446:                    if      m6800!m6801
995C 01         447:                    nop
                448:                    fin
995D 7E8D45     449:                    jmp     ilputdev:$FFC8
9960 7E8D4E     450:                    jmp     ilgetdev:$FFC8
9963 7E8EDB     451:                    jmp     illdeviceop     no extra control calls defined
9966 7E8EDB     452:                    jmp     illdeviceop     no extra status calls defined
9969 7E8D57     453:                    jmp     tlcheckready:$FFC8 go check for acia ready
  9943          454:                    org     dcb:$FFC8+dcb:outputtoblk
  0000          455:                    ifund   ttytimeouts
                457:                    fin     ttytimeouts
  9943          458: outputtoblk:$FFC8
9943 994B       459:                    fdb     inputtoblk:$FFC8
9945 0000       460:                    fdb     0
9947 BDD9       461:                    fdb     sdos+sdos:vtoutputto
9949 98A2       462:                    fdb     dcb:$FFC8
  994B          463:                    org     dcb:$FFC8+dcb:inputtoblk
  994B          464: inputtoblk:$FFC8
994B 93C0       465:                    fdb     nexttimeout
  96C8          466: nexttimeout        set     ttytimeouts
994D 0000       467:                    fdb     0
994F BDD6       468:                    fdb     sdos+sdos:vtinputto
9951 98A2       469:                    fdb     dcb:$FFC8
  9953          470:                    org     dcb:$FFC8+dcb:tcb
9953 999F       471:                    fdb     tcb:$FFC8
  9955          472:                    org     dcb:$FFC8+dcb:taskstack
9955 99D1       473:                    fdb     tcbstack:$FFC8
  98EF          474:                    org     dcb:$FFC8+dcb:ringinbase
```

```
98EF 956A       475:                    fdb     inbuf:$FFC8
98F1 0050       476:                    fdb     inbufsize:$FFC8
  98FB          477:                    org     dcb:$FFC8+dcb:ringoutbase
98FB 951A       478:                    fdb     outbuf:$FFC8
98FD 0050       479:                    fdb     outbufsize:$FFC8
  98FF          480:                    org     dcb:$FFC8+dcb:ringoutthreshold
98FF 08         481:                    fcb     outbufsize:$FFC8//10
  9909          482:                    org     dcb:$FFC8+dcb:linebuf
9909 95BA       483:                    fdb     linebuf:$FFC8
  991C          484:                    org     dcb:$FFC8+dcb:linebuflen
991C 64         485:                    fcb     linebufsize:$FFC8
  996C          486:                    org     dcb:$FFC8+dcb:clearin
996C 7EBDB8     487:                    jmp     sdos+sdos:vtclearin
996F 7EBDBB     488:                    jmp     sdos+sdos:vtclearout
9972 7EBDBE     489:                    jmp     sdos+sdos:vttlputbuf
9975 7EBDC1     490:                    jmp     sdos+sdos:vttlgetbuf
9978 7EBDC4     491:                    jmp     sdos+sdos:vtilputbuf
997B 7EBDC7     492:                    jmp     sdos+sdos:vtilgetbuf
  98CA          493:                    org     dcb:$FFC8+dcb:profile
98CA 01         494:                    fcb     profilenum.MALVT profile name
  98A2          495:                    org     dcb:$FFC8+dcb:doneflag
98A2 01         496:                    fcb     1               device not busy
  98B5          497:                    org     dcb:$FFC8+dcb:oilquiescent
98B5 01         498:                    fcb     1               interrupt not expected
  999F          499:                    org     dcb:$FFC8+dcb:vtsize
  0000          500:                    ifund   ttytcb
                502:                    fin     ttytcb
  999F          503: tcb:$FFC8
999F 0000       504:                    fdb     nexttcb
99A1 99D1       505:                    fdb     tcbstack:$FFC8
99A3 00000000   506:                    fdb     0,0,0,0,0,0
  99AD          507:                    org     tcb:$FFC8+tcb:scratchpad+dcbpointer
99AD 98A2       508:                    fdb     dcb:$FFC8
  99AF          509:                    org     tcb:$FFC8+tcb:size
99AF 0022       510:                    rmb     env:minstack*2-env:size
  99D1          511: tcbstack:$FFC8
  0008          512:                    rpt     env:size
99D1 00         513:                    fcb     0
  99D8          514:                    org     *-env:cc
  99D8          515: stack:$FFC8
  99D7          516:                    org     tcbstack:$FFC8+env:p
99D7 BDCA       517:                    fdb     sdos+sdos:vtedittask
  99D2          518:                    org     tcbstack:$FFC8+env:cc
99D2 00         519:                    fcb     $80*m6809       set 'E' flag for 6809
  99D9          520:                    org     stack:$FFC8+env:cc
                521:                    fin     iodriverram
                522:
                523:
  9377          607: DISKDCBS SET   NEXTDISKDCB
  0008          608: NDRIVES  SET   NDISKDCBS
  8FEB          609: DEVICEDCBS     SET                 NEXTDEVICEDCB
  96C8          610: TIMEOUTQUEUE   SET                 NEXTTIMEOUT
  0009          611: NTIMEOUTBLOCKS SET                 NTIMEOUTS
  9724          612: TASKQUEUE      EQU                 TTYTCB
  8DFB          613: PROFILECHAIN   EQU                 NEXTDPB
```

```
                615: *
                616: *       FCBS (MUST PRECEDE IOCBS)
                617: *
  02A4          618: FCBS     RPT   FCB:SIZE*(NIOCHANNELS+2*NDRIVES+NMAGICFCBS)
99D9 00         619:          FCB   0
                620: *
                621: *       IOCBS
                622: *
  0110          623: IOCBS    RPT   IOCB:SIZE*NIOCHANNELS
9C7D 00         624:          FCB   0
  0008          625: IOCBPOINTERS   RPT                   NIOCHANNELS
9D8D 9C7D       626:          FDB   IOCBS+IOCB:SIZE*(*-IOCBPOINTERS)/2
                628: *
  9D9D          629: INTERRUPTSTACK EQU * ; STACK SPACE FOR INTERRUPT ROUTINES
                630:
9D9D DEFE       631: INTSETUP LDX   SYSPG
9D9F 868F       632:          LDAA  #(STACKUNSWITCHEDDEVICEPOLL)/256 = WHERE TO GO ON INTERRUPT
9DA1 C618       633:          LDAB  #(STACKUNSWITCHEDDEVICEPOLL)\256
9DA3 A7FE       634:          STAA  SYSIIRQ+1,X
9DA5 E7FF       635:          STAB  SYSIIRQ+2,X
9DA7 0C39       636:          OKRTS
                637:
  003A          638:          RPT   INTERRUPTSTACKSIZE-(*-INTERRUPTSTACK)
9DA9 00         639:          FCB   0
                640:
  9DE3          641: INTERRUPTSTACKEND ; end of interruptstack
```

```
MAL/6800 1.3F: 9DE2  SDOSDRIVERS       *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 109; Form 1    *** DRIVER INIT (ONCE-ONLY) CODE ***
IOJUPITER.ASM
  0000          643: IODRIVERRAM    SET                     0
  0001          644: IODRIVERINIT   SET                     1
  19E3          645:          EQU   *-DRIVERBASE            SIZE OF READ-ONLY CODE
  9DE3          646: DSKBUFFERPOOL  EQU                     *
  0001          647:          IF    CLOCK
                648:          INCLUDE                       IOCLOCK.ASM
  0000            1:                    IF      IODRIVERBODY
                201:                    FIN     IODRIVERBODY
  0000          202:                    IF      IODRIVERRAM
                228:                    FIN     IODRIVERRAM
                229:
                230:
                649:          FIN
  0000          650:          IF    BLACKHOLE
                652:          FIN
  0000          653:          IF    SDLP
                655:          FIN
  0001          656:          IF    VIRTUALFLOPPY
                657:          INCLUDE                       IOVFD.ASM
  0000            1:              IF      IODRIVERBODY
                 57:                 FIN     IODRIVERBODY
  0000           58:                 IF      IODRIVERRAM
                127:            FIN     IODRIVERRAM
  0001          128:              IF      IODRIVERINIT
  9DE3          129: FDRESTORE
9DE3 CE0C39     130:              LDX     #OKRTS          do the following once only
9DE6 FF9DE3     131:              STX     FDRESTORE
9DE9 CE0000     132:              LDX     #0               reset the PIA(s)
  0002          133:              IF      PERSCI
9DEC FFFFA0     134:              STX     PERSCI:PIACA
                135:              FIN     PERSCI
  0002          136:              IF      DAMFLOPPY
9DEF FFFF80     137:              STX     DAMFLOPPY:PIACA
                138:              FIN     DAMFLOPPY
9DF2 CEFFFF     139:              LDX     ##FFFF
  0002          140:              IF      PERSCI
9DF5 FFFFA2     141:              STX     PERSCI:PIADA
9DF8 86A5       142:              LDAA    ##A5            see if Persci controller exists
9DFA B7FFA2     143:              STAA    PERSCI:PIADA
9DFD 40         144:              NEGA
9DFE BBFFA2     145:              ADDA    PERSCI:PIADA    gives zero IFF Persci exists
9E01 40         146:              NEGA                    gives carry IFF Persci exists
9E02 8600       147:              LDAA    #0              form Persci Interrupt Test mask
9E04 46         148:              RORA                    (A)=$80 --> Persci exists
9E05 B78FC3     149:              STAA    PERSCIINTERRUPTMASK
                150:              FIN     PERSCI
  0002          151:              IF      DAMFLOPPY
9E08 FFFF82     152:              STX     DAMFLOPPY:PIADA
9E0B 86A5       153:              LDAA    ##A5            see if DAM Floppy controller exists
9E0D B7FF82     154:              STAA    DAMFLOPPY:PIADA
9E10 40         155:              NEGA
9E11 BBFF82     156:              ADDA    DAMFLOPPY:PIADA gives zero IFF DAM Floppy exists
9E14 40         157:              NEGA                    gives carry IFF DAM Floppy exists
9E15 8600       158:              LDAA    #0              form DAM Floppy Interrupt Test mask
9E17 46         159:              RORA                    (A)=$80 --> DAM Floppy exists
```



```
MAL/6800 1.3F: 9E18  SDOSDRIVERS        *** SDOS I/O drivers for WaveMate Jupiter II (C) 1978 SOFTWARE DYNAMICS ***
01/14/83 11:39:33; Page 10; Form 1      *** DRIVER INIT (ONCE-ONLY) CODE ***
IOVFD.ASM
9E18 B78FD3     160:            STAA    DAMFLOPPYINTERRUPTMASK
                161:            FIN     DAMFLOPPY
9E1B CE2C07     162:            LDX     #%0010110000000111
  0002          163:            IF      PERSCI
9E1E FFFFA0     164:            STX     PERSCI:PIACA
9E21 B6FFA3     165:            LDAA    PERSCI:PIADB      clear possible interrupt
                166:            FIN     PERSCI
  0002          167:            IF      DAMFLOPPY
9E24 FFFF80     168:            STX     DAMFLOPPY:PIACA
9E27 B6FF83     169:            LDAA    DAMFLOPPY:PIADB   clear possible interrupt
                170:            FIN     DAMFLOPPY
9E2A 0C39       171:            OKRTS
                172:            FIN     IODRIVERINIT
  0000          173:          IF      IODRIVERBODY
                410:          FIN     IODRIVERBODY
  0000          411:          IF      IODRIVERPOLL
                434:        FIN     IODRIVERPOLL
  0000          435:        IF      IODRIVERBODY
                995:        FIN     IODRIVERBODY
                996:
                997:
                658:         FIN
  0001          659:         IF    STORAGEDEMON
                660:         INCLUDE                     IOSTOREDEMON.ASM
  0000            1:                 IF      IODRIVERBODY
                781:                 FIN     IODRIVERBODY
  0000          782:                 IF      IODRIVERPOLL
                807:                 FIN     IODRIVERPOLL
  0001          808:                 IF      IODRIVERINIT
  9E2C          809: WDCINIT         ;LDX    #OKRTS          INITIALIZE 7710 INTELLIGENT CONTROLLER
9E2C CE0C39     810:                 LDX     #OKRTS
9E2F FF9E2C     811:                 STX     WDCINIT         SO WE DON'T DO THIS MORE THAN ONCE!
                812:                 ;JMP    WDCRESET        INITIALIZE 7710 INTELLIGENT CONTROLLER
9E32 7E8A36     813:                 JMP     WDCRESET
  0001          814:                 IF      USEDEMONASCLOCK
  9E35          815: CLOCKRESET
                816:                 ;LDD    #2*16666        ASSUME 2MHZ CPU
9E35 C634       817:                 LDAB    #(2*16666)&$FF
9E37 8682       818:                 LDAA    #(2*16666)/256
                819:                 ;STB    VIAT1LL         ; SET INTERVAL IN LOW LATCH
9E39 F7FF44     820:                 STAB    VIAT1LL
                821:                 ;STA    VIAT1CH         ; LOAD HIGH LATCH AND INTO COUNTER
9E3C B7FF45     822:                 STAA    VIAT1CH
                823:                 ;LDA    #%01000000
9E3F 8640       824:                 LDAA    #%01000000
                825:                 ;STA    VIAACR          ; SET CONTINUOUS INTERRUPTS FROM COUNTER
9E41 B7FF4B     826:                 STAA    VIAACR
                827:                 ;LDA    #%11000000
9E44 86C0       828:                 LDAA    #%11000000
                829:                 ;STA    VIAIER          ; ENABLE INTERRUPT REQUEST FROM CLOCK
9E46 B7FF4E     830:                 STAA    VIAIER
9E49 0C39       831:                 OKRTS
                832:                 FIN     USEDEMONASCLOCK
                833:                 FIN     IODRIVERINIT
  0000          834:                 IF      IODRIVERRAM
```

(16666/100 * CPUSpeed

IF JUPITER USE DIFFERENT VALUE (850 KC CLOCK)

```
                886:                    FIN     IODRIVERRAM
                887:                    END     ;UNEXPECTED EOF
                661:            FIN
                662:            INCLUDE                 IOVTCONFIG.ASM
  0000           1:                    if      iodriverbody
                30:                    fin     iodriverbody
  0001          31:                    if      iodriverinit
  9E4B          32: reset:$FFC0
9E4B 8603       33:                    ldaa    #%00000011      reset ACIA
9E4D B7FFC0     34:                    staa    $FFC0
9E50 8695       35:                    ldaa    #%10010101      in int; 8 D + 1 S; no parity; /16
9E52 B7FFC0     36:                    staa    $FFC0
9E55 B6FFC1     37:                    ldaa    $FFC1           clear any input interrupt
9E58 0C39       38:                    okrts
                39:                    fin     iodriverinit
  0000          40:                    if      iodriverbody
                57:                    fin     iodriverbody
  0001          58:                    if      iodriverinit
  9E5A          59: reset:$FFC4
9E5A 8603       60:                    ldaa    #%00000011      reset ACIA
9E5C B7FFC4     61:                    staa    $FFC4
9E5F 8695       62:                    ldaa    #%10010101      in int; 8 D + 1 S; no parity; /16
9E61 B7FFC4     63:                    staa    $FFC4
9E64 B6FFC5     64:                    ldaa    $FFC5           clear any input interrupt
9E67 0C39       65:                    okrts
                66:                    fin     iodriverinit
  0000          67:                    if      iodriverbody
                84:                    fin     iodriverbody
  0001          85:                    if      iodriverinit
  9E69          86: reset:$FFC8
9E69 8603       87:                    ldaa    #%00000011      reset ACIA
9E6B B7FFC8     88:                    staa    $FFC8
9E6E 8695       89:                    ldaa    #%10010101      in int; 8 D + 1 S; no parity; /16
9E70 B7FFC8     90:                    staa    $FFC8
9E73 B6FFC9     91:                    ldaa    $FFC9           clear any input interrupt
9E76 0C39       92:                    okrts
                93:                    fin     iodriverinit
  0000          94:                    if      iodriverbody
               111:                    fin     iodriverbody
  0000         112:                    if      iodriverpoll
               193:                    fin     iodriverpoll
  0000         194:                    if      iodriverram
               521:                    fin     iodriverram
               522:
               523:
```

```
  0000          664:          IF    *>/VTDRIVER
                666:          ELSE
9E78 0788       667:          RMB   VTDRIVER-*
                668:          FIN   *>/VTDRIVER
  081D          669: DSKPOOLSIZE    EQU                     *-DSKBUFFERPOOL
  0000          670:          IF    DESIREDPOOLSIZE>>DSKPOOLSIZE
                672:          FIN
  2200          673:          EQU   *-CODE                  SO I CAN SEE HOW BIG THE WORLD IS
  0000          674:          END
```

Symbols Sorted by Name:

```
 ::/9321        :BPS/0100       :CONTROLLER/9017              :CYL/004D       :DAMFLOPPY/0000                :DATA/0000
 :DCB/92B6      :DCBNUMBER/0006              :DRIVENUMBER/0002               :FIRST/0000     :HEADCHAIN/9155
 :IBMFORMAT/0000                :NEXTCHAIN/0001               :PERSCI/0000    :SPT/0010       :TPC/0001       :WMFORMAT/0001
*ALTERPROFILE:CLIDLES/000D      *ALTERPROFILE:CLLEN/0008       *ALTERPROFILE:CLSEQ/0009       *ALTERPROFILE:COLDISP/0007
*ALTERPROFILE:CPIDLES/0005      *ALTERPROFILE:CPLEN/0000       *ALTERPROFILE:CPSEQ/0001       *ALTERPROFILE:EEOLIDLES/0013
*ALTERPROFILE:EEOLLEN/000E      *ALTERPROFILE:EEOLSEQ/000F     *ALTERPROFILE:ROWDISP/0006     *ALTERPROFILE:SIZE/0014
*ASCII:ACK/0006 *ASCII:BEL/0007  ASCII:BS/0008  *ASCII:CAN/0018 *ASCII:CR/000D  *ASCII:DC1/0011 *ASCII:DC2/0012 *ASCII:DC3/0013
*ASCII:DC4/0014 *ASCII:DLE/0010 *ASCII:EM/0019   ASCII:ENQ/0005 *ASCII:EOT/0004  ASCII:ESC/001B *ASCII:ETB/0017 *ASCII:ETX/0003
 ASCII:FF/000C  *ASCII:FS/001C  *ASCII:GS/001D  *ASCII:HT/0009   ASCII:LF/000A  *ASCII:MASK/007F                ASCII:NAK/0015
*ASCII:NULL/0000                *ASCII:RS/001E  *ASCII:RUBOUT/007F             *ASCII:SI/000F  *ASCII:SO/000E  *ASCII:SOH/0001
*ASCII:SPACE/0020               *ASCII:STX/0002 *ASCII:SUB/001A *ASCII:SYN/0016 *ASCII:US/001F   ASCII:VT/000B
 BADINTERRUPTCOUNT/8FE9         *BASICFLAGS/00F0               BCDTOASC/84FE   BLACKHOLE/0000 *BOOT:PARAMSIZE/0007
*BUILDMAP/85CF   BUILDMAP1/85D4  BUILDMAP2/85D8  BUILDMAP3/85DD  BUILDMAP4/85E9 *CC:ACTIVATIONCK/001D
*CC:ALTERPROFILE/0019           *CC:BACKGROUND/0021            *CC:CLRINPUT/0015              *CC:CLROUTPUT/0016
*CC:COLORING/0020               CC:DEVICESPECIFICOP/0010       CC:DISMOUNTDISK/0011           *CC:DUMPBUFFERS/0001
*CC:ECHO/0010    CC:FORMAT/0015 *CC:IDLES/0012  *CC:KILLENABLE/0023           *CC:KILLPROOF/0022
*CC:MULTISECTORREAD/0013        *CC:MULTISECTORWRITE/0014      *CC:NOECHO/0011 *CC:NOWRAP/001F *CC:POSITION/0000
*CC:POSITIONTOEND/0013          *CC:SETACTBLOCK/0014           *CC:SETEXCEPTION/0032          *CC:SETFIELDSIZE/001B
*CC:SETFILEDATE/0010            *CC:SETFILEPROT/0011           *CC:SETFILESIZE/0012           *CC:SETMAPALGORITHM/0012
*CC:SETPARAMS/001C              *CC:SETPROFILE/0018            *CC:SETREADTIMEOUT/0017        *CC:SETTIMESHARE/0031
*CC:TABS/0013   *CC:UNLOCKDISK/0010            *CC:WAITDONE/0016              *CC:WRAP/001E   *CC:WRITEANDWAIT/0030
*CC:WRITEBNOWAIT/0033           *CC:WRITEEDITLINE/001A         CCB:ABORT/000F *CCB:ADDR/0001   CCB:BUSY/0000   CCB:CURRENTDCB/002
 CCB:CYL/0005    CCB:DAMFLOPPY/9045             CCB:DRIVE/0004  CCB:LASTCYL/0006               CCB:PERSCI/9017
 CCB:READSECTOR/001B            CCB:RESET/000C  CCB:RESTORE/0012              CCB:SEEK/0018   CCB:SETSEEK/0015
*CCB:SIZE/002E   CCB:STARTIO/0007              CCB:STATUS/0009                CCB:TIMEOUT/0003               CCB:TIMEOUTBLK/002
 CCB:VERIFYSECTOR/0021          CCB:WRITESECTOR/001E           *CHANGED/0000   CHECKDISKREADY/8762            CLOCK/0001
 CLOCKBUFFER/8FF4               CLOCKCLOSE/842F                CLOCKDATE/84D7  CLOCKDCB/8FEB   CLOCKDRIVER/8415
*CLOCKFRACTION/8FFA             CLOCKGETTD/84B9                CLOCKGETTD1/84C0               CLOCKGETTD2/84CD
 CLOCKMAKEXX/8522               CLOCKOPEN/842F  CLOCKPFRESTART/842F            CLOCKRA1/84A6   CLOCKRB1/8475   CLOCKRB2/8484
 CLOCKREADA/848D                CLOCKREADB/8469                CLOCKRESET/9E35                CLOCKSPRUNG/8431
 CLOCKSTATUS/8434               CLOCKSTR/8FFB   CLOCKTIME/850A CLOCKWB1/8451   CLOCKWB2/845F   CLOCKWRITEB/8448
*CNFG:ATTNCHECK/000B            *CNFG:DEBUGGER/000D            *CNFG:DEVICEDCBS/0002          *CNFG:DISKDCBS/0000
*CNFG:DRIVERBASE/000F           *CNFG:DSKBUFFERPOOL/0007       *CNFG:DSKPOOLSIZE/0009         *CNFG:INTDISABLE/0013
*CNFG:INTENABLE/0016            *CNFG:INTERRUPTSTACK/001C      *CNFG:INTRTI/0019              *CNFG:INTSETUP/0011
*CNFG:IOCBPOINTERS/0004         *CNFG:IOINTPOLL/001E           *CNFG:MTPRIMS/0028             *CNFG:NIOCHANNELS/0006
*CNFG:TASKQUEUE/0020            CNFG:TIMEOUTLIST/0022          *CNFG:VTDEBUG/0026             *CNFG:VTPROFILES/0024
*CNFG:VTSIZE/002A               CNFGTABLE/8EB1  CODE/8400      COLORING:H19/8E8E              COLORING:H19REVERSEVIDEO/8E9A
 CONRAC/0000     CONTEXTBLOCK:SIZE/0007         COPYDCBTOCCB/8845              COUNTCOMMAND/881B              DAMFLOPPY/0002
 DAMFLOPPY:ABORT/88FF           DAMFLOPPY:ABORT.RTS/8913       DAMFLOPPY:ISSUECOMMAND/88F9    DAMFLOPPY:PIACA/FF80
 DAMFLOPPY:PIACB/FF81           DAMFLOPPY:PIADA/FF82           DAMFLOPPY:PIADB/FF83           DAMFLOPPY:READSECTOR/8938
 DAMFLOPPY:READSECTOR.1/894F    DAMFLOPPY:READSECTOR.2/8945    DAMFLOPPY:RESET/8914           DAMFLOPPY:RESTORE/88F2
 DAMFLOPPY:SEEK/891B            DAMFLOPPY:SETSEEK/8918         DAMFLOPPY:STATUS/88EB          DAMFLOPPY:TIMEOUT/897B
 DAMFLOPPY:VERIFYSECTOR/892E    DAMFLOPPY:WDCMDSTS/FF84        DAMFLOPPY:WDDATA/FF87          DAMFLOPPY:WDSECTOR/FF86
 DAMFLOPPY:WDTRACK/FF85         DAMFLOPPY:WRITESECTOR/8952     DAMFLOPPY:WRITESECTOR.1/8973   DAMFLOPPY:WRITESECTOR.2/8965
 DAMFLOPPYINTERRUPTMASK/8FD3   *DATE/84D5       DATE$/900B     DATE$:DAY/900E  DATE$:MONTH/900B               DATE$:YEAR/9011
 DAY/8FF7        DCB:$FFC0/9627 DCB:$FFC4/9763  DCB:$FFC8/98A2 *DCB:ACTCOL/007B                *DCB:ACTDISP/007C
*DCB:ACTIVATION/00ED            *DCB:BACKGROUND/0078           DCB:BEEPCOUNT/005E             *DCB:CALLERIOCB/0086
*DCB:CALLERSCB/0026             *DCB:CLEAR/0094 *DCB:CLEARIDLES/0098           DCB:CLEARIN/00CA               *DCB:CLEAROUT/00CD
*DCB:CLEARSL/0093               *DCB:COL/006E   *DCB:COLCT/000D *DCB:COLDISP/0092              *DCB:COLORING/0076
*DCB:CONTROL/00C1               *DCB:CTLCCOUNT/000E            *DCB:CTLCKILL/0075             *DCB:CURSORLOST/006C
*DCB:DISPLAYDEPTH/006A          *DCB:DISPLAYWIDTH/0069          DCB:DONEFLAG/0000             *DCB:DRIVER/0007
*DCB:EDITFLAGS/0009             *DCB:EEOL/009A  *DCB:EEOLIDLES/009E            *DCB:EEOLSL/0099               *DCB:ENDCOL/006B
*DCB:EXCEPT/0082                *DCB:FIELDEND/0074             *DCB:FIELDWIDTH/0088           *DCB:IDLECOUNT/00A0
```

```
#DCB:IDLETRIGGER/009F      #DCB:IILLFLGS/0012      #DCB:IILSPL/0011      #DCB:ILDATA/0062
#DCB:ILGETBUF/00D9         #DCB:ILGETDEV/00BE      #DCB:ILPUTBUF/00D6    #DCB:ILPUTDEV/00BB
#DCB:ILROOM/0080           #DCB:ILSW/000C  DCB:INPUTTOBLK/00A9  #DCB:ISDEVICEREADY/00C7  DCB:LASTERROR/0001
 DCB:LINEBUF/0067          #DCB:LINEBUFCOUNT/0066  DCB:LINEBUFLEN/007A  #DCB:LINEBUFPTR/000F
#DCB:LINEFLAGS/005F        DCB:NAME/0003  #DCB:NEWSTATUS/0079  DCB:NEXTDCB/0005
 DCB:OILQUIESCENT/0013     #DCB:OPENCOUNT/0085     DCB:OUTPUTTOBLK/00A1  #DCB:POSN/008C  #DCB:POSNIDLES/0090
#DCB:POSNSL/008B           #DCB:PROCESSID/0083     DCB:PROFILE/0028      #DCB:READAERR/0072
#DCB:READCOL/006F          #DCB:READPERIOD/0089    #DCB:REMINDERS/000A    DCB:RESET/00B5  DCB:RINGINBASE/004
#DCB:RINGINDATA/0047       #DCB:RINGINFETCH/0045   #DCB:RINGINLEN/004F   #DCB:RINGINROOM/004B
#DCB:RINGINSTORE/0049      DCB:RINGOUTBASE/0059    #DCB:RINGOUTDATA/0053 #DCB:RINGOUTFETCH/0051
#DCB:RINGOUTLEN/005B       #DCB:RINGOUTROOM/0057   #DCB:RINGOUTSTORE/0055  DCB:RINGOUTTHRESHOLD/005D
#DCB:ROW/006D  #DCB:ROWCT/0071 #DCB:ROWDISP/0091   #DCB:SCB/0014  DCB:SIZE/0009  #DCB:STATUS/00C4
#DCB:TABS/00DC  DCB:TASKSTACK/00B3  DCB:TCB/00B1   DCB:TLBUFFER/007D  #DCB:TLCLOSEDEV/00B8
#DCB:TLDATA/0060           #DCB:TLGETBUF/00D3      #DCB:TLPUTBUF/00D0    #DCB:TLROOM/0064
 DCB:VTSIZE/00FD           #DCB:WELFLAGS/000B      #DCB:WELPOS/0070       DCB:XLATESTATE/003D
#DCBEDITFLAGS:ACTIVATE/0010  #DCBEDITFLAGS:ESC/0001  #DCBEDITFLAGS:HCEDIT/0040  #DCBEDITFLAGS:INTO/0020
#DCBEDITFLAGS:KILLP/0080   #DCBEDITFLAGS:PAGE/0008  #DCBEDITFLAGS:READB/0002  #DCBEDITFLAGS:WRAP/0004
#DCBEXCEPT:SEDIT/0001      #DCBIILLFLGS:CTLB/0020  #DCBIILLFLGS:CTLG/0008  #DCBIILLFLGS:CTLT/0040
#DCBIILLFLGS:CTLV/0010     #DCBIILLFLGS:ESC/0080   #DCBIILSPL:CONTINUE/0008  #DCBIILSPL:DISCARD/0004
#DCBIILSPL:FREEZE/0002     #DCBIILSPL:INTO/0020    #DCBIILSPL:PAGE/0001   #DCBILSW:ALPHALOCK/0001
#DCBILSW:CTLC/0002         #DCBILSW:HCFREEZE/0080  #DCBILSW:OUTTO/0004    DCBNAME:$FFC0/961E
 DCBNAME:$FFC4/975E        DCBNAME:$FFC8/989A      DCBPOINTER/0006       #DCBREMINDERS:CTLO/0008
#DCBREMINDERS:CTLP/0004    #DCBREMINDERS:CTLS/0002  #DCBREMINDERS:INTO/0020  #DCBREMINDERS:RIP/0010
#DCBWELFLAGS:ECHO/0020     #DCBWELFLAGS:FLDE/0008  #DCBWELFLAGS:FLDW/0004  #DCBWELFLAGS:PREF/0001
#DCBWELFLAGS:RETYPE/0002   DEBUGINTERRUPT/8410     DEBUGSYSCALLHANDLER/8407  DESIREDPOOLSIZE/0800
#DEVICEDCBS/8FEB           DISKABORT/876F  DISKCOMPL/8804  DISKCOMPLEMENT/87FB  DISKDCBS/9377  DISKDONE/8751
 DISKDONE1/8758  DISKDONEJ/87EC  DISKDONEJ1/87E5   DISKERROR/8741  DISKERROR1/874D  DISKERRORJ/87C1
 DISKINTCCB/9015           DISKINTDAMFLOPPY.NO/8FDF  DISKINTDCB/9013  DISKINTERRUPT/86B5
 DISKINTPERSCI.NO/8FCF     #DISKINTSERVICE/8FBF    DISKINTSETUP/86B9     DISKINTSTART/86CA
 DISKINTSTARTDAMFLOPPY/86C7  DISKINTSTARTPERSCI/86C2  DISKINTUNEXPECTED/875E  DISKREAD/87D4  DISKREAD1/87E1
 DISKREAD4/87EF  DISKSAVEERRLSN/87C4  DISKSEEKERROR/8735  DISKSETCYLADD/8774
 DISKSETCYLADD.1/8776      DISKTIMEOUT/897E        DISKTIMEOUT1/89A9     DISKTIMEOUT1A/89B6
 DISKTIMEOUT2/89C1         DISKTIMEOUTERRORED/8994  DISKWPERR/873B  DISKWRITE/8780  DISKWRITE2/8791
 DISKWRITE3/87A3           DISKWRITE4/87A7         DISKWRITE5/87B1       DIV60DIVIDEND/8FF4
 DIVIDE60L/8531  DIVIDE60L2/853A  DIVIDE60L3/8549  DIVIDEBY60/852E  DOSEEK/885D
#DPB:DEFDEPTH/0005         #DPB:DEFWIDTH/0004      #DPB:DVTYP/0001 #DPB:FLAGS/0006  DPB:GPINIT/0015
#DPB:NEXT/0002  #DPB:OUTTO/0007 #DPB:PROFILE/0000  #DPB:SETBACKGROUND/0012  #DPB:SETCOLORING/000F
 DPB:SIZE/001D  #DPB:TLPUTDEV/000C  #DPB:XLATEI/0009  #DPBFLAGS:AUTONL/0008
#DPBFLAGS:HCEDIT/0010      #DPBFLAGS:MAL/0001      #DPBFLAGS:OUTPUT/0002  #DPBFLAGS:WRAP/0004
#DRIVER:CLOSE/0002         #DRIVER:CONTROL/0012    #DRIVER:CREATE/000C   #DRIVER:DELETE/0010
#DRIVER:DISKCONTROL/000A   #DRIVER:DISKREAD/0002   #DRIVER:DISKRESET/0000  #DRIVER:DISKSTATUS/0008
#DRIVER:DISKWAIT/0006      #DRIVER:DISKWRITE/0004  #DRIVER:OPEN/0000      DRIVER:PFRESTART/001A
#DRIVER:READA/0004         #DRIVER:READB/0008      #DRIVER:RENAME/000E   #DRIVER:RESET/0016
#DRIVER:STARTIO/0018       DSKBUFFERPOOL/9DE3      #DRIVER:WRITEA/0006   #DRIVER:WRITEB/000A
 DRIVERBASE/8400           #DSKINFO:LOG2NBPS/0018  #DSKINFO:BADLSN/002D  #DSKINFO:DIRFCB/0027
 DSKINFO:ERRLSN/003F       #DSKINFO:MIDALLOC/0014   DSKINFO:MAPALGORITHM/0016  #DSKINFO:MAPFCB/0029
#DSKINFO:MAPLSN/0024       #DSKINFO:NBPSM1/0019    #DSKINFO:MINALLOC/0012  #DSKINFO:NBPC/0020
 DSKINFO:NBPS/0009         DSKINFO:NSPC/0011       #DSKINFO:NCYL/000F    #DSKINFO:NLCN/001E
#DSKINFO:NLSN/001B         #DSKINFO:RANDMAP/0022    DSKINFO:NSPT/000B     #DSKINFO:NTPC/000D
 DSKINFO:OPSCOUNT/003C     DSKINFO:SEEKERRCNT/0030  DSKINFO:READERRCNT/0038  DSKINFO:READERRSTS/003A
 DSKINFO:SECTORDB/002B     DSKINFO:WRITEERRSTS/0036  DSKINFO:SEEKERRSTS/0032  DSKINFO:SIZE/0042
 DSKINFO:WRITEERRCNT/0034  #DVDAT:NCYL/0006        DSKPOOLSIZE/081D      #DVDAT:DEPTH/0001
 DVDAT:NBPS/0000           #DVDAT:WIDTH/0000       #DVDAT:NSPC/0002      #DVDAT:NSPT/0002
#DVDAT:NTPC/0004           #DVTYP.DTAPE/0003        DVTYP.CLOCK/000B      DVTYP.CONSOLE/0004
#DVTYP.DISK/0001                                   #DVTYP.DUMMY/000A     #DVTYP.FILE/0000
```

IOJUPITER.ASM

```
*DVTYP.PARIN/0009          *DVTYP.PAROUT/0008          DVTYP.PRINTER/0005          *DVTYP.SERIALIN/0007
*DVTYP.SERIALOUT/0006      *DVTYP.STAPE/0002           DVTYP:TYPE/0000             EDITDATE/1231  EDITYEAR/1982
*ENV:A/0003  *ENV:B/0002   ENV:CC/0001  ENV:MINSTACK/0015          ENV:P/0006      ENV:SIZE/0008  *ENV:X/0004
*ERR:ABNORMALSTOP/0068     *ERR:ACTIVATIONNOTINBUFFER/0773 *ERR:ACTIVATIONRECEIVED/0775  *ERR:ALLOCOCLUSTERS/042B
*ERR:ATTENTION/0001        *ERR:BADCMDFORMAT/0066      *ERR:BADFILENAME/03FF       *ERR:BADFILEVERSION/0405
*ERR:BADFNAMESIZE/03F5     *ERR:BADLOADRECORD/040C     *ERR:BADPOSITION/03EC       *ERR:BOOTCKSUMFAIL/03E8
*ERR:BRANCHFACTORSIZE/0435 *ERR:BUSYFORANOTHERPROCESS/0772 *ERR:CANTGOTO/0067       *ERR:CANTOPENMUSTCREATE/03FD
*ERR:CHBUSY/0407           *ERR:CHTOOBIG/0406          *ERR:CLOSED/0408            *ERR:CLUSTERSIZELIMITSFILE/041C
*ERR:DECRYPTIONKEYSDONTMATCH/0437          *ERR:DEVICEERRORED/0421          ERR:DEVICENOTREADY/0424
 ERR:DEVICETIMEDOUT/0412   *ERR:DIRECTORYDAMAGED/040F  *ERR:DISKMOUNTED/03FC       ERR:DISKREAD/0415
 ERR:DISKSEEK/0417          ERR:DISKWRITE/0416         *ERR:DISKWRITELOCKED/0419   ERR:DSKWRTPROT/0418
*ERR:DUPLICATEKEY/0434     *ERR:ENDOFMEDIUM/042F       *ERR:EOFHIT/03E9            *ERR:FATALCOMPILE/0064
*ERR:FILEALREADYDELETED/042C *ERR:FILEINCREATE/03FB    *ERR:FILEISOPEN/03EA        *ERR:FILENOTFOUND/03F3
*ERR:FILEWRTPROT/03F2      *ERR:HCSICTOOSMALL/0401     *ERR:IBUFOVERFLOW/0410      ERR:ILLDEVICEOP/040A
*ERR:ILLEGALLCN/03F4       *ERR:ILLEGALSYSCALL/0409    *ERR:ILLFILESIZE/0400       *ERR:ILLLSN/040E
*ERR:INPUTTIMEOUT/042E     *ERR:IOINPROGRESS/0771      *ERR:LCNWASNTALLOCATED/03F8 *ERR:MUSTBEDISK/0422
*ERR:NBPCTOOBIG/03ED       *ERR:NEWFILEEXISTS/03F6     *ERR:NLSNGE224/041B         *ERR:NODEBUGGER/03EB
*ERR:NODEFAULTPROGRAM/03F0 *ERR:NODISKMAP/03EE         *ERR:NODISKSPACE/03F7       *ERR:NOERRORMSGS/03FE
*ERR:NOFREEFCBS/03F9       *ERR:NOMATCHFCB/03EF        *ERR:NONE/0000  *ERR:NOSUCHDEVICE/0420        *ERR:NOSUCHKEY/0433
*ERR:NOSUCHLUN/0426        *ERR:NOSUCHPROGRAM/0428     *ERR:NOTALOADFILE/0404      *ERR:NOTENOUGHPOOL/0402
*ERR:NOTENOUGHROOM/040D    *ERR:NOTENUFMEM/0069        *ERR:NOTIMEOUTBLKS/0431     *ERR:NOTOPENTOCONSOLE/0423
*ERR:OLDFILEEXISTS/0429    *ERR:PRINTERNOTREADY/042D   *ERR:PROFILENOTFOUND/0777   *ERR:PROFILENOTMALLEABLE/0778
*ERR:PROGRAMKILLED/0411    *ERR:PWRFAILDISKF/0403      *ERR:RDBUFTOOSMALL/041E     *ERR:RENAMEDEVICE/040B
*ERR:SDOSCKSUM/041A        *ERR:SDOSMTALREADYRUNNING/04CE *ERR:SDOSMTPRIMSMISSING/04D0 *ERR:SDOSNOTREGISTERED/0436
*ERR:SECTORSIZE2/0413      *ERR:SELFTESTCKSUM/0430     *ERR:SERIALNOWRONG/0432     *ERR:STATUSHASCHANGED/04CF
*ERR:SYSCALLTOOSHORT/041D  *ERR:SYSTEMCROAKED/0414     *ERR:TIMEDINPUTEXPIRED/0776 *ERR:TIMENOTSET/0425
*ERR:WARNINGCOMPILE/0065   *ERR:WRBUFTOOSMALL/041F     *ERR:WRONGDISKTYPE/076E     *ERR:WRONGFILESYSTEM/03FA
*ERR:ZEROSTARTADDRESS/0427  ERRETX/8EE0  ERRORRTS/0D39  EXORCISOR/0000 *FCB:DAY/0016  *FCB:DIRDISP/0005
*FCB:DIRLSN/0002           *FCB:DISKINFO/0000          *FCB:FILESIZE/0011          *FCB:FLAGS/000A *FCB:HCSIC/000E
*FCB:HLCN/000C  *FCB:HLSN/0007  *FCB:MONTH/0017 *FCB:NCLUSTERS/000F          *FCB:PROT/0015  *FCB:REFCOUNT/000B
 FCB:SIZE/001A  *FCB:VERSION/0019          *FCB:YEAR/0018  *FCBS/99D9       FDCCB/0051      FDCOMPLEMENT/004B
 FDCONTROL/855D  FDCYL/0049      FDDISMOUNT/8564           FDDRIVE/0048    FDDRIVER/8551   FDDSTATE/0044   FDDSTATEJ/0043
 FDFIRSTSEC/004C                 FDHEADCHAIN/004D          FDK16MODNSPT/0059               FDK1MODNSPT/0055
 FDK2MODNSPT/0056                *FDK32MODNSPT/005A        FDK4MODNSPT/0057                FDK8MODNSPT/0058
 FDMAP/005B      FDMAPALG/0053   FDNEXTCHAIN/004F          FDREAD/856D     FDREAD.1/856E   FDREADWRITE/0042
 FDRESTORE/9DE3  FDRETRY/0047    FDSECTOR/004A   FDSEEKRETRY/0046          FDSETUP1/85CB   FDSETUP2/860D   FDSETUP4/86B1
 FDSETUPDRIVE/85AF               FDSIZE/005B     FDSTARTIO/857A  FDSTATUS/8566   FDTIMEOUTBLOCK/9069             FDWAIT1/8598
 FDWAIT2/859F    FDWAITDONE/858F                 FDWRITE/8569    *FILESYSTEMVERSION/0010         *GETCV/FC09
 GOTOUTPUT:$FFC0/8F24            GOTOUTPUT:$FFC4/8F51      GOTOUTPUT:$FFC8/8F7E            IBMFORMAT/0001 *IC:CREATE/0000
*IC:DESTROY/0001                *IC:LOCK/0003   *IC:RELEASE/0004          *IC:RESET/0002  *IC:TEST/0005   *IGNORED/0000
 ILGETDEV:$FFC0/8D1C             ILGETDEV:$FFC4/8D35       ILGETDEV:$FFC8/8D4E             ILGETDEVICESTATUSFROMACIAERROR/8EA
 ILGETDEVSTATUSFROMACIA/8EA2     ILLDEVICEOP/8EDB          ILPUTDEV:$FFC0/8D13             ILPUTDEV:$FFC4/8D2C
 ILPUTDEV:$FFC8/8D45             IMI5007/0000   *IMI7710S/0001  IMI7711/0000   INBUF:$FFC0/9416                INBUF:$FFC4/951A
 INBUF:$FFC8/956A                INBUFSIZE:$FFC0/0050       INBUFSIZE:$FFC4/0000          INBUFSIZE:$FFC8/0050
*INICV/FC03    *INIDV/FC12      INPUTTOBLK:$FFC0/96D0      INPUTTOBLK:$FFC4/980C         INPUTTOBLK:$FFC8/994B
 INTDISABLE/8EC4                *INTENABLE/8EC7  INTERRUPTSTACK/9D9D          INTERRUPTSTACKEND/9DE3
 INTERRUPTSTACKSIZE/0046        *INTERRUPTTARGET/BE15     *INTRTI/8ECA    INTSETUP/9D9D  *IOCB:BUFFERP/0005
*IOCB:BYTECOUNT/0016            *IOCB:COLCNT/000C         *IOCB:CURBYTE/000D            *IOCB:CURLCN/0018
*IOCB:CURLSN/0002               *IOCB:DCB/0000  *IOCB:DRDSI/0012          *IOCB:DRIVER/0009               *IOCB:DRSN/001A
*IOCB:EOFFLAG/000B              *IOCB:FCB/0007  *IOCB:HRDSI/001F          *IOCB:HRSN/0021 *IOCB:LOCATEDF/0011
*IOCB:NEXTBYTE/0014             *IOCB:RBN/001D  *IOCB:RDCN/001B  IOCB:SIZE/0022  IOCBPOINTERS/9D8D              IOCBS/9C7D
 IODRIVERBODY/0000               IODRIVERINIT/0001          IODRIVERPOLL/0000             IODRIVERRAM/0000
 IOPKDEFS/0001  *JUPITERII/0001  JWDCCMDFEED/8CA3          K/0400          LCN:SIZE/0002   LINEBUF:$FFC0/9466
 LINEBUF:$FFC4/951A              LINEBUF:$FFC8/95BA        LINEBUFSIZE:$FFC0/0064        LINEBUFSIZE:$FFC4/0000
 LINEBUFSIZE:$FFC8/0064         *LINEFLAGS/00F0 *LIST.VIRTUALFLOPPY/0001       *LISTCLOCK/0001  LISTDEFS/0000
```

Symbols Sorted by Value:

```
 :DAMFLOPPY/0000                :DATA/0000      :FIRST/0000     :IBMFORMAT/0000                :PERSCI/0000
*ALTERPROFILE:CPLEN/0000       *ASCII:NULL/0000                 BLACKHOLE/0000 *CC:POSITION/0000              CCB:BUSY/0000
*CHANGED/0000   *CNFG:DISKDCBS/0000             CONRAC/0000     DCB:DONEFLAG/0000             *DPB:PROFILE/0000
*DRIVER:DISKRESET/0000         *DRIVER:OPEN/0000                DVDAT:NBPS/0000               *DVDAT:WIDTH/0000
*DVTYP.FILE/0000                DVTYP:TYPE/0000                *ERR:NONE/0000   EXORCISOR/0000 *FCB:DISKINFO/0000
*IC:CREATE/0000 *IGNORED/0000    IMI5007/0000    IMI7711/0000    INBUFSIZE:$FFC4/0000          *IOCB:DCB/0000   IODRIVERBODY/0000
 IODRIVERPOLL/0000              IODRIVERRAM/0000                 LINEBUFSIZE:$FFC4/0000         LISTDEFS/0000   M6801/0000
 M6809/0000      NEXTTCB/0000   *RDSI:DISKINFO/0000             *RDSISTATE:IDLE/0000          *SC:GETPOS/0000 *SCBLK:OPCODE/0000
 SDLP/0000      *SDOS:VERSIONNUMBER/0000         SDOSMT/0000    *SECTORDB:DISKINFO/0000        SYSTEMDEFS/0000
*TCB:LNK/0000   *TEMP/0000      *TEMPA/0000      TEMPX/0000     *TIMEOUT:LINK/0000             USECONSOLEACIAASCLOCK/0000
 WMDAMFLOPPY/0000               WMPERSCI/0000   WMSERIES2000/0000              :NEXTCHAIN/0001                :TPC/0001
 :WMFORMAT/0001 *ALTERPROFILE:CPSEQ/0001        *ASCII:SOH/0001 *CC:DUMPBUFFERS/0001          *CCB:ADDR/0001   CLOCK/0001
 DCB:LASTERROR/0001            *DCBEDITFLAGS:ESC/0001           *DCBEXCEPT:SEDIT/0001         *DCBIILSPL:PAGE/0001
*DCBILSW:ALPHALOCK/0001         *DCBWELFLAGS:PREF/0001           *DPB:DVTYP/0001 *DPBFLAGS:MAL/0001             *DVDAT:DEPTH/0001
*DVTYP.DISK/0001                ENV:CC/0001     *ERR:ATTENTION/0001             IBMFORMAT/0001 *IC:DESTROY/0001
*IMI7710S/0001   IODRIVERINIT/0001               IOPKDEFS/0001  *JUPITERII/0001 *LIST.VIRTUALFLOPPY/0001       *LISTCLOCK/0001
*LISTSTORAGEDEMON/0001          M6800/0001      OUTASPACE/0001 *PROFILE.ADM3/0001             *PROFILE.EPSONLPT/0001
*PROFILE.GT100/0001            *PROFILE.H19/0001                *PROFILE.MALLPT/0001          *PROFILE.MALVT/0001
*PROFILE.RS232LPT/0001         *PROFILE.SOROCIQ120/0001          PROFILENUM.MALVT/0001         *PROT::BACKUP/0001
*RDSISTATE:READING/0001        *REALTIMECLOCK/0001              *REG:CC/0001    *SC:GETCOL/0001 *SCBLK:WLEN/0001
*SDOS:LASTERROR/0001            STORAGEDEMON/0001               *TEMPB/0001      USEDEMONASCLOCK/0001           VIRTUALFLOPPY/0001
 WAVEMATE/0001   WDCFORMAT/0001 WDCNCYL/0001    WDCNTPC/0001    WMFORMAT/0001   :DRIVENUMBER/0002             *ASCII:STX/0002
*CNFG:DEVICEDCBS/0002           DAMFLOPPY/0002 *DCBEDITFLAGS:READB/0002        *DCBIILSPL:FREEZE/0002        *DCBILSW:CTLC/0002
*DCBREMINDERS:CTLS/0002        *DCBWELFLAGS:RETYPE/0002         *DPB:NEXT/0002  *DPBFLAGS:OUTPUT/0002         *DRIVER:CLOSE/0002
*DRIVER:DISKREAD/0002          *DVDAT:NSPC/0002                 *DVDAT:NSPT/0002               *DVTYP.STAPE/0002
*ENV:B/0002     *FCB:DIRLSN/0002               *IC:RESET/0002  *IOCB:CURLSN/0002               LCN:SIZE/0002   NMAGICFCBS/0002
 PERSCI/0002     RDSI:LSN/0002 *RDSISTATE:WRITING/0002         *REG:B/0002     *SC:GETEOF/0002 *SCBLK:PARAMS/0002
*SECTORDB:LSN/0002             *TCB:STACK/0002  TIMEOUT:FUSE/0002               WDCREADCMD/0002               *ASCII:ETX/0003
 CCB:TIMEOUT/0003               DCB:NAME/0003  *DVTYP.DTAPE/0003                *ENV:A/0003     *IC:LOCK/0003   LSN:SIZE/0003
 PROFILENUM.ADM3/0003          *RDSISTATE:VERIFYING/0003        *REG:A/0003     *SC:GETFILESIZE/0003
*SDOS:CONFIGURATION/0003        WDCWRITECMD/0003                *ASCII:EOT/0004  CCB:DRIVE/0004 *CNFG:IOCBPOINTERS/0004
*DCBEDITFLAGS:WRAP/0004        *DCBIILSPL:DISCARD/0004          *DCBILSW:OUTTO/0004            *DCBREMINDERS:CTLP/0004
*DCBWELFLAGS:FLDW/0004         *DPB:DEFWIDTH/0004               *DPBFLAGS:WRAP/0004            *DRIVER:DISKWRITE/0004
*DRIVER:READA/0004             *DVDAT:NTPC/0004                 DVTYP.CONSOLE/0004            *ENV:X/0004     *IC:RELEASE/0004
*REG:X/0004      SC:GETTYPE/0004                SCBLK:WRBUF/0004               *TCB:COND/0004  *TIMEOUT:ROUTINE/0004
*ALTERPROFILE:CPIDLES/0005      ASCII:ENQ/0005  CCB:CYL/0005    DCB:NEXTDCB/0005               *DPB:DEFDEPTH/0005
 DVTYP.PRINTER/0005            *FCB:DIRDISP/0005                *IC:TEST/0005   *IOCB:BUFFERP/0005             PROFILENUM.H19/000
 RDSI:SECTORBASE/0005          *SC:GETPARAMS/0005               *SDOS:SERIALNUMBER/0005        *SECTORDB:ADDRESS/0005
 WDCRETRY/0005   :DCBNUMBER/0006               *ALTERPROFILE:ROWDISP/0006      *ASCII:ACK/0006  CCB:LASTCYL/0006
*CNFG:NIOCHANNELS/0006          DCBPOINTER/0006                 *DPB:FLAGS/0006 *DRIVER:DISKWAIT/0006         *DRIVER:WRITEA/0006
*DVDAT:NCYL/0006               *DVTYP.SERIALOUT/0006            ENV:P/0006      PROFILENUM.HARDCOPYVT/0006    *REG:PC/0006
*SCBLK:WRLEN/0006              *TCB:PARAM/0006 *TIMEOUT:DCB/0006               *ALTERPROFILE:COLDISP/0007     *ASCII:BEL/0007
*BOOT:PARAMSIZE/0007            CCB:STARTIO/0007               *CNFG:DSKBUFFERPOOL/0007        CONTEXTBLOCK:SIZE/0007
*DCB:DRIVER/0007               *DPB:OUTTO/0007 *DVTYP.SERIALIN/0007            *FCB:HLSN/0007  *IOCB:FCB/0007  *RDSI:FLINK/0007
 SDOS:IOBLOCKPTR/0007          *SECTORDB:SIZE/0007              *ALTERPROFILE:CLLEN/0008        ASCII:BS/0008
*DCBEDITFLAGS:PAGE/0008        *DCBIILLFLGS:CTLG/0008           *DCBIILSPL:CONTINUE/0008       *DCBREMINDERS:CTLO/0008
*DCBWELFLAGS:FLDE/0008         *DPBFLAGS:AUTONL/0008            *DRIVER:DISKSTATUS/0008        *DRIVER:READB/0008
*DVTYP.PAROUT/0008              ENV:SIZE/0008   NDISKDCBS/0008  NDRIVES/0008    NIOCHANNELS/0008               SCBLK:RPLEN/0008
 TCB:SCRATCHPAD/0008            TIMEOUT:SIZE/0008               *ALTERPROFILE:CLSEQ/0009       *ASCII:HT/0009   CCB:STATUS/0009
*CNFG:DSKPOOLSIZE/0009         *DCB:EDITFLAGS/0009               DCB:SIZE/0009  *DPB:XLATEI/0009               DSKINFO:NBPS/0009
*DVTYP.PARIN/0009              *IOCB:DRIVER/0009                 NTIMEOUTBLOCKS/0009            NTIMEOUTS/0009
 PROFILENUM.MALLPT/0009        *RDSI:BLINK/0009                *SDOS:IOCBPOINTER/0009          ASCII:LF/000A  *DCB:REMINDERS/000A
*DRIVER:DISKCONTROL/000A       *DRIVER:WRITEB/000A              *DVTYP.DUMMY/000A              *FCB:FLAGS/000A  SCBLK:RDBUF/000A
```

```
 ASCII:VT/000B  *CNFG:ATTNCHECK/000B          *DCB:WELFLAGS/000B            DSKINFO:NSPT/000B             DVTYP.CLOCK/000B
*FCB:REFCOUNT/000B            *IOCB:EOFFLAG/000B            PROFILENUM.RS232LPT/000B      *RDSI:MODIFIED/000B
 SDOS:CLOCK/000B               ASCII:FF/000C   CCB:RESET/000C *DCB:ILSW/000C  *DPB:TLPUTDEV/000C           *DRIVER:CREATE/000C
*FCB:HLCN/000C  *IOCB:COLCNT/000C             *RDSI:STATE/000C              *SCBLK:RDLEN/000C
*ALTERPROFILE:CLIDLES/000D    *ASCII:CR/000D  *CNFG:DEBUGGER/000D           *DCB:COLCT/000D *DSKINFO:NTPC/000D
*IOCB:CURBYTE/000D             RDSI:SECTOR/000D             *ALTERPROFILE:EEOLLEN/000E    *ASCII:SO/000E  *DCB:CTLCCOUNT/000E
*DRIVER:RENAME/000E           *FCB:HCSIC/000E *SCBLK:DATA/000E              SCBLK:END/000E *SDOS:DAY/000E
*ALTERPROFILE:EEOLSEQ/000F    *ASCII:SI/000F   CCB:ABORT/000F *CNFG:DRIVERBASE/000F         *DCB:LINEBUFPTR/000F
*DPB:SETCOLORING/000F         *DSKINFO:NCYL/000F           *FCB:NCLUSTERS/000F           *RDSI:TRACK/000F
*SDOS:MONTH/000F               :SPT/0010      *ASCII:DLE/0010  CC:DEVICESPECIFICOP/0010    *CC:ECHO/0010   *CC:SETFILEDATE/001
*CC:UNLOCKDISK/0010           *DCBEDITFLAGS:ACTIVATE/0010   *DCBIILLFLGS:CTLV/0010        *DCBREMINDERS:RIP/0010
*DPBFLAGS:HCEDIT/0010         *DRIVER:DELETE/0010           *FILESYSTEMVERSION/0010        SC:DEVICESPECIFICOP/0010
*SC:GETFILEDATE/0010          *SC:GETLASTBADLSN/0010        *SC:GETPROFILE/0010           *SDOS:YEAR/0010  TCB:SIZE/0010
*ASCII:DC1/0011  CC:DISMOUNTDISK/0011         *CC:NOECHO/0011 *CC:SETFILEPROT/0011          *CNFG:INTSETUP/0011
*DCB:IILSPL/0011               DSKINFO:NSPC/0011            *FCB:FILESIZE/0011            *IOCB:LOCATEDF/0011
 RDSI:CYLINDER/0011           *SC:GETACTCOL/0011            *SC:GETERRORSTATS/0011        *SC:GETFILEPROT/0011
*SDOS:STACKSWITCHED/0011      *SDOSVERSION/0011             *ASCII:DC2/0012 *CC:IDLES/0012  *CC:SETFILESIZE/0012
*CC:SETMAPALGORITHM/0012       CCB:RESTORE/0012             *DCB:IILLFLGS/0012            *DPB:SETBACKGROUND/0012
*DRIVER:CONTROL/0012          *DSKINFO:MINALLOC/0012        *IOCB:DRDSI/0012               SDOS:IOINT/0012
*ALTERPROFILE:EEOLIDLES/0013  *ASCII:DC3/0013 *CC:MULTISECTORREAD/0013      *CC:POSITIONTOEND/0013        *CC:TABS/0013
*CNFG:INTDISABLE/0013          DCB:OILQUIESCENT/0013        *RDSI:SIZE/0013 *ALTERPROFILE:SIZE/0014      *ASCII:DC4/0014
*CC:MULTISECTORWRITE/0014     *CC:SETACTBLOCK/0014          *DCB:SCB/0014   *DRIVER:STATUS/0014
*DSKINFO:MIDALLOC/0014        *IOCB:NEXTBYTE/0014           ASCII:NAK/0015 *CC:CLRINPUT/0015             CC:FORMAT/0015
 CCB:SETSEEK/0015              DPB:GPINIT/0015              ENV:MINSTACK/0015             *FCB:PROT/0015   MINSTACK/0015
 SDOS:RTI/0015  *ASCII:SYN/0016 *CC:CLROUTPUT/0016          *CC:WAITDONE/0016             *CNFG:INTENABLE/0016
*DRIVER:RESET/0016             DSKINFO:MAPALGORITHM/0016    *FCB:DAY/0016   *IOCB:BYTECOUNT/0016        *ASCII:ETB/0017
*CC:SETREADTIMEOUT/0017       *FCB:MONTH/0017 *ASCII:CAN/0018 *CC:SETPROFILE/0018          CCB:SEEK/0018  *DRIVER:STARTIO/001
*DSKINFO:LOG2NBPS/0018        *FCB:YEAR/0018  *IOCB:CURLCN/0018              SDOS:RESCHEDULE/0018          *ASCII:EM/0019
*CC:ALTERPROFILE/0019         *CNFG:INTRTI/0019             *DSKINFO:NBPSM1/0019          *FCB:VERSION/0019
*ASCII:SUB/001A *CC:WRITEEDITLINE/001A        DRIVER:PFRESTART/001A          FCB:SIZE/001A  *IOCB:DRSN/001A  ASCII:ESC/001B
*CC:SETFIELDSIZE/001B          CCB:READSECTOR/001B          *DSKINFO:NLSN/001B            *IOCB:RDCN/001B
 SDOS:CLOCKTICKED/001B        *ASCII:FS/001C  *CC:SETPARAMS/001C            *CNFG:INTERRUPTSTACK/001C     *ASCII:GS/001D
*CC:ACTIVATIONCK/001D          DPB:SIZE/001D  *IOCB:RBN/001D  *ASCII:RS/001E  *CC:WRAP/001E    CCB:WRITESECTOR/001E
*CNFG:IOINTPOLL/001E          *DSKINFO:NLCN/001E            *SDOS:CURRENTASK/001E         *ASCII:US/001F  *CC:NOWRAP/001F
*IOCB:HRDSI/001F              *ASCII:SPACE/0020             *CC:COLORING/0020             *CNFG:TASKQUEUE/0020
*DCBEDITFLAGS:INTO/0020       *DCBIILLFLGS:CTLB/0020        *DCBIILSPL:INTO/0020          *DCBREMINDERS:INTO/0020
*DCBWELFLAGS:ECHO/0020        *DSKINFO:NBPC/0020            *SDOS:KILLPROOF/0020          *CC:BACKGROUND/0021
 CCB:VERIFYSECTOR/0021        *IOCB:HRSN/0021 *SDOS:KILLUSERPROGRAM/0021    *CC:KILLPROOF/0022
 CNFG:TIMEOUTLIST/0022        *DSKINFO:RANDMAP/0022          IOCB:SIZE/0022 *CC:KILLENABLE/0023           CCB:TIMEOUTBLK/002
*CNFG:VTPROFILES/0024         *DSKINFO:MAPLSN/0024           SDOS:STARTID/0024             *CNFG:VTDEBUG/0026
*DCB:CALLERSCB/0026           *DSKINFO:DIRFCB/0027          *SDOS:WAITCOND/0027            *CNFG:MTPRIMS/0028
 DCB:PROFILE/0028             *DSKINFO:MAPFCB/0029          *CNFG:VTSIZE/002A              SDOS:WAITEVENT/002A
 DSKINFO:SECTORDB/002B         CCB:CURRENTDCB/002C          *SC:GETLINEFLAGS/002C         *DSKINFO:BADLSN/002D
 SDOS:ERROR/002D              *CCB:SIZE/002E  *CC:WRITEANOWAIT/0030          DSKINFO:SEEKERRCNT/0030       *SC:ATTENTIONCK/003
 SDOS:ERRORSAVE/0030          *CC:SETTIMESHARE/0031         *SC:STATUSCK/0031             *CC:SETEXCEPTION/0032
 DSKINFO:SEEKERRSTS/0032      *SC:GETTIMESHARE/0032         *CC:WRITEBNOWAIT/0033         *SC:ALLSTATUS/0033
 SDOS:ERRORED/0033             DSKINFO:WRITEERRCNT/0034     *SC:GETLINEFLAGSHINT/0034     *SC:GETFREECOUNT/0035
 DSKINFO:WRITEERRSTS/0036     *SC:GETDATACOUNT/0036          SDOS:CHECKRDLEN/0036          DSKINFO:READERRCNT/0038
 SDOS:CHECKWRLEN/0039          DSKINFO:READERRSTS/003A       DSKINFO:OPSCOUNT/003C         MEMSIZE/003C
*SDOS:CHECKSCLEN/003C          TICKSPERSECOND/003C           DCB:XLATESTATE/003D           DSKINFO:ERRLSN/003F
*SDOS:TABLEBRANCH/003F        *DCBEDITFLAGS:HCEDIT/0040     *DCBIILLFLGS:CTLT/0040        *PROT::WRITE/0040
 DSKINFO:SIZE/0042             FDREADWRITE/0042             *SDOS:BLOCKMOVE/0042           WDCREADWRITE/0042
 FDDSTATEJ/0043  WDCDRIVE/0043 FDDSTATE/0044  *WDCSIZE/0044 *DCB:RINGINFETCH/0045        *SDOS:ENTRYSIZE/0045
 FDSEEKRETRY/0046              INTERRUPTSTACKSIZE/0046      *DCB:RINGINDATA/0047           FDRETRY/0047    FDDRIVE/0048
 SDOS:EXTENSIONSIZE/0048      *DCB:RINGINSTORE/0049          FDCYL/0049     FDSECTOR/004A *DCB:RINGINROOM/004B
```

```
 FDCOMPLEMENT/004B          FDFIRSTSEC/004C            :CYL/004D      DCB:RINGINBASE/004D        FDHEADCHAIN/004D
*DCB:RINGINLEN/004F         FDNEXTCHAIN/004F           INBUFSIZE:$FFC0/0050        INBUFSIZE:$FFC8/0050
 OUTBUFSIZE:$FFC0/0050      OUTBUFSIZE:$FFC4/0050      OUTBUFSIZE:$FFC8/0050      *DCB:RINGOUTFETCH/0051
 FDCCB/0051     *DCB:RINGOUTDATA/0053       FDMAPALG/0053 *DCB:RINGOUTSTORE/0055      FDK1MODNSPT/0055
 FDK2MODNSPT/0056           *DCB:RINGOUTROOM/0057      FDK4MODNSPT/0057           FDK8MODNSPT/0058
 DCB:RINGOUTBASE/0059       FDK16MODNSPT/0059          *FDK32MODNSPT/005A         *DCB:RINGOUTLEN/005B
 FDMAP/005B      FDSIZE/005B DCB:RINGOUTTHRESHOLD/005D  DCB:BEEPCOUNT/005E         *DCB:LINEFLAGS/005F
*DCB:TLDATA/0060            *DCB:ILDATA/0062           *DCB:TLROOM/0064           *ERR:FATALCOMPILE/0064
 LINEBUFSIZE:$FFC0/0064     LINEBUFSIZE:$FFC8/0064     *ERR:WARNINGCOMPILE/0065   *DCB:LINEBUFCOUNT/0066
*ERR:BADCMDFORMAT/0066      DCB:LINEBUF/0067           *ERR:CANTGOTO/0067         *ERR:ABNORMALSTOP/0068
*DCB:DISPLAYWIDTH/0069      *ERR:NOTENUFMEM/0069       *DCB:DISPLAYDEPTH/006A     *DCB:ENDCOL/006B
*DCB:CURSORLOST/006C        *DCB:ROW/006D   *DCB:COL/006E *DCB:READCOL/006F       *DCB:WELPOS/0070
*DCB:ROWCT/0071 *DCB:READAERR/0072          *DCB:FIELDEND/0074         *DCB:CTLCKILL/0075          *DCB:COLORING/0076
*DCB:BACKGROUND/0078        *DCB:NEWSTATUS/0079        DCB:LINEBUFLEN/007A        *DCB:ACTCOL/007B
*DCB:ACTDISP/007C           DCB:TLBUFFER/007D          *ASCII:MASK/007F           *ASCII:RUBOUT/007F
*DCB:ILROOM/0080            *DCBEDITFLAGS:KILLP/0080   *DCBIILLFLGS:ESC/0080      *DCBILSW:HCFREEZE/0080
*SPECIALFN:NEWLINE/0080     WDCFATAL/0080   SPECIALFN:POSN/0081        *DCB:EXCEPT/0082
 SPECIALFN:CLEAR/0082       *DCB:PROCESSID/0083        SPECIALFN:EEOL/0083        *DCB:OPENCOUNT/0085
*DCB:CALLERIDCB/0086        *DCB:FIELDWIDTH/0088       *DCB:READPERIOD/0089       *DCB:POSNSL/008B
*DCB:POSN/008C  *DCB:POSNIDLES/0090         *DCB:ROWDISP/0091          *DCB:COLDISP/0092           *DCB:CLEARSL/0093
*DCB:CLEAR/0094 *DCB:CLEARIDLES/0098        *DCB:EEOLSL/0099           *DCB:EEOL/009A  *DCB:EEOLIDLES/009E
*DCB:IDLETRIGGER/009F       *DCB:IDLECOUNT/00A0        DCB:OUTPUTTOBLK/00A1       DCB:INPUTTOBLK/00A9
 DCB:TCB/00B1    DCB:TASKSTACK/00B3          DCB:RESET/00B5 *DCB:TLCLOSEDEV/00B8   *DCB:ILPUTDEV/00BB
*DCB:ILGETDEV/00BE          *DCB:CONTROL/00C1          *DCB:STATUS/00C4           *DCB:ISDEVICEREADY/00C7
 DCB:CLEARIN/00CA           *DCB:CLEAROUT/00CD         *DCB:TLPUTBUF/00D0         *DCB:TLGETBUF/00D3
*DCB:ILPUTBUF/00D6          *DCB:ILGETBUF/00D9         *DCB:TABS/00DC  *DCB:ACTIVATION/00ED        *BASICFLAGS/00F0
*LINEFLAGS/00F0  SYSDEPENDENT/00F0          *SYSCALL$/00FB  DCB:VTSIZE/00FD        SYSIIRQ/00FD    SYSPG/00FE
 :BPS/0100       WDCNBPS/0200   *ERR:BOOTCKSUMFAIL/03E8    *ERR:EOFHIT/03E9           *ERR:FILEISOPEN/03EA
*ERR:NODEBUGGER/03EB        *ERR:BADPOSITION/03EC      *ERR:NBPCTOOBIG/03ED       *ERR:NODISKMAP/03EE
*ERR:NOMATCHFCB/03EF        *ERR:NODEFAULTPROGRAM/03F0 *ERR:FILEWRTPROT/03F2      *ERR:FILENOTFOUND/03F3
*ERR:ILLEGALLCN/03F4        *ERR:BADFNAMESIZE/03F5     *ERR:NEWFILEEXISTS/03F6    *ERR:NODISKSPACE/03F7
*ERR:LCNWASNTALLOCATED/03F8 *ERR:NOFREEFCBS/03F9       *ERR:WRONGFILESYSTEM/03FA  *ERR:FILEINCREATE/03FB
*ERR:DISKMOUNTED/03FC       *ERR:CANTOPENMUSTCREATE/03FD *ERR:NOERRORMSGS/03FE     *ERR:BADFILENAME/03FF
*ERR:ILLFILESIZE/0400        K/0400         *ERR:HCSICTOOSMALL/0401     *ERR:NOTENOUGHPOOL/0402
*ERR:PWRFAILDISKF/0403      *ERR:NOTALOADFILE/0404     *ERR:BADFILEVERSION/0405   *ERR:CHTOOBIG/0406
*ERR:CHBUSY/0407            *ERR:CLOSED/0408           *ERR:ILLEGALSYSCALL/0409    ERR:ILLDEVICEOP/040A
*ERR:RENAMEDEVICE/040B      *ERR:BADLOADRECORD/040C    *ERR:NOTENOUGHROOM/040D    *ERR:ILLLSN/040E
*ERR:DIRECTORYDAMAGED/040F  *ERR:IBUFOVERFLOW/0410     *ERR:PROGRAMKILLED/0411     ERR:DEVICETIMEDOUT/0412
*ERR:SECTORSIZE2/0413        ERR:DSKWRTPROT/0418       ERR:DISKREAD/0415          ERR:DISKWRITE/0416
 ERR:DISKSEEK/0417          *ERR:CLUSTERSIZELIMITSFILE/041C *ERR:DISKWRITELOCKED/0419 *ERR:SDOSCKSUM/041A
*ERR:NLSNGE224/041B         *ERR:NOSUCHDEVICE/0420     *ERR:SYSCALLTOOSHORT/041D  *ERR:RDBUFTOOSMALL/041E
*ERR:WRBUFTOOSMALL/041F     *ERR:NOSUCHDEVICE/0420     *ERR:DEVICEERRORED/0421    *ERR:MUSTBEDISK/0422
*ERR:NOTOPENTOCONSOLE/0423  ERR:DEVICENOTREADY/0424    *ERR:TIMENOTSET/0425       *ERR:NOSUCHLUN/0426
*ERR:ZEROSTARTADDRESS/0427  *ERR:NOSUCHPROGRAM/0428    *ERR:OLDFILEEXISTS/0429    *ERR:ALLOCOCLUSTERS/042B
*ERR:FILEALREADYDELETED/042C *ERR:PRINTERNOTREADY/042D *ERR:INPUTTIMEOUT/042E     *ERR:ENDOFMEDIUM/042F
*ERR:SELFTESTCKSUM/0430     *ERR:NOTIMEOUTBLKS/0431    *ERR:SERIALNOWRONG/0432    *ERR:NOSUCHKEY/0433
*ERR:DUPLICATEKEY/0434      *ERR:BRANCHFACTORSIZE/0435 *ERR:SDOSNOTREGISTERED/0436 *ERR:DECRYPTIONKEYSDONTMATCH/0437
*ERR:SDOSMTALREADYRUNNING/04CE *ERR:STATUSHASCHANGED/04CF *ERR:SDOSMTPRIMSMISSING/04D0 *ERR:WRONGDISKTYPE/076E
*ERR:IOINPROGRESS/0771      *ERR:BUSYFORANOTHERPROCESS/0772 *ERR:ACTIVATIONNOTINBUFFER/0773 *ERR:ACTIVATIONRECEIVED/0775
*ERR:TIMEDINPUTEXPIRED/0776 *ERR:PROFILENOTFOUND/0777  *ERR:PROFILENOTMALLEABLE/0778  DESIREDPOOLSIZE/0800
 DSKPOOLSIZE/081D            OKRTS/0C39     ERRORRTS/0D39   EDITDATE/1231   EDITYEAR/1982   WDCNSPT/4E34    CODE/8400
 DRIVERBASE/8400             SYSCALLIO/8400 DEBUGSYSCALLHANDLER/8407  DEBUGINTERRUPT/8410          CLOCKDRIVER/8415
 CLOCKCLOSE/842F             CLOCKOPEN/842F CLOCKPFRESTART/842F       CLOCKSPRUNG/8431            CLOCKSTATUS/8434
 CLOCKWRITEB/8448            CLOCKWB1/8451  CLOCKWB2/845F   CLOCKREADB/8469          CLOCKRB1/8475  CLOCKRB2/8484
 CLOCKREADA/848D             CLOCKRA1/84A6  CLOCKGETTD/84B9          CLOCKGETTD1/84C0               CLOCKGETTD2/84CD
```

```
*DATE/84D5       CLOCKDATE/84D7  BCDTOASC/84FE   CLOCKTIME/850A  CLOCKMAKEXX/8522              DIVIDEBY60/852E
 DIVIDE60L/8531  DIVIDE60L2/853A                 DIVIDE60L3/8549                FDDRIVER/8551  FDCONTROL/855D  FDDISMOUNT/8564
 FDSTATUS/8566   FDWRITE/8569    FDREAD/856D     FDREAD.1/856E   FDSTARTIO/857A  FDWAITDONE/858F                FDWAIT1/8598
 FDWAIT2/859F    MODULONSPTB/85A1                MODULONSPT/85A8                FDSETUPDRIVE/85AF              FDSETUP1/85CB
*BUILDMAP/85CF   BUILDMAP1/85D4  BUILDMAP2/85D8  BUILDMAP3/85DD  BUILDMAP4/85E9  FDSETUP2/860D  MAP1/8692       MAP2/8697
 MAP3/869C       MAP4/86A1       MAP5/86A6       FDSETUP4/86B1   DISKINTERRUPT/86B5             DISKINTSETUP/86B9
 DISKINTSTARTPERSCI/86C2         DISKINTSTARTDAMFLOPPY/86C7      DISKINTSTART/86CA              SEEK/86E5       SEEKDONE/86FE
 SEEK3/8710      SEEK3.1/8718    SEEKHOME/8720   DISKSEEKERROR/8735              DISKWPERR/873B  DISKERROR/8741  DISKERROR1/874D
 DISKDONE/8751   DISKDONE1/8758  DISKINTUNEXPECTED/875E          CHECKDISKREADY/8762            MAKEDISKREADY/876A
 DISKABORT/876F  DISKSETCYLADD/8774              DISKSETCYLADD.1/8776            DISKWRITE/8780  DISKWRITE2/8791
 SEEKDONEJ/87A0  DISKWRITE3/87A3                 DISKWRITE4/87A7                 DISKWRITE5/87B1                SEEKHOMEJ/87BE
 DISKERRORJ/87C1                 DISKSAVEERRLSN/87C4             DISKREAD/87D4   DISKREAD1/87E1  DISKDONEJ1/87E5
 DISKDONEJ/87EC  DISKREAD4/87EF  DISKCOMPLEMENT/87FB             DISKCOMPL/8804  COUNTCOMMAND/881B
 WAITFORINTERRUPT/8828           TESTFORSEEK/8831                COPYDCBTOCCB/8845               DOSEEK/885D     PERSCI:STATUS/8864
 PERSCI:RESTORE/886C             PERSCI:ISSUECOMMAND/8875        PERSCI:ABORT/887B               PERSCI:ABORT.RTS/8890
 PERSCI:RESET/8891               PERSCI:SETSEEK/8895             PERSCI:SEEK/8897                PERSCI:VERIFYSECTOR/88AC
 PERSCI:READSECTOR/88B3          PERSCI:READSECTOR.2/88C1        PERSCI:READSECTOR.1/88CC        PERSCI:WRITESECTOR/88CF
 PERSCI:WRITESECTOR.1/88E8       DAMFLOPPY:STATUS/88EB           DAMFLOPPY:RESTORE/88F2          DAMFLOPPY:ISSUECOMMAND/88F9
 DAMFLOPPY:ABORT/88FF            DAMFLOPPY:ABORT.RTS/8913        DAMFLOPPY:RESET/8914            DAMFLOPPY:SETSEEK/8918
 DAMFLOPPY:SEEK/891B             DAMFLOPPY:VERIFYSECTOR/892E     DAMFLOPPY:READSECTOR/8938       DAMFLOPPY:READSECTOR.2/8945
 DAMFLOPPY:READSECTOR.1/894F     DAMFLOPPY:WRITESECTOR/8952      DAMFLOPPY:WRITESECTOR.2/8965    DAMFLOPPY:WRITESECTOR.1/8973
 PERSCI:TIMEOUT/8976             DAMFLOPPY:TIMEOUT/897B          DISKTIMEOUT/897E                DISKTIMEOUTERRORED/8994
 DISKTIMEOUT1/89A9               DISKTIMEOUT1A/89B6              DISKTIMEOUT2/89C1               WDCDRIVER/89C4  WDCCONTROL/89D0
 WDCFORMATX/89DB                 WDCWRITE/89E5   WDCREAD/89E9    WDCOPSET/89EB   WDCSETRETRY1/89F1               WDCSETUP/8A02
 WDCOKRTS/8A20   WDCWAITDONE/8A22                WDCWAIT1/8A2B   WDCRESET/8A36   WDCRESETLP/8A44                 WDCFORMSERVJ/8A56
 WDCREADSERVJ/8A59               WDCCMDFEED/8A5C                 WDCSTARTIO/8A5C                 WDCCMDFEED0/8A6A
*WDCCMDFEED1/8A80               *WDCWRITESERV/8A95               WDCWRITEWAIT1ST/8A9D            WDCWRITELOOP/8AAA
 WDCWRITE0/8AB6  WDCWRITE1/8AC2  WDCWRITE2/8ACE  WDCWRITE3/8ADA  WDCWRITE4/8AE6  WDCWRITE5/8AF2  WDCWRITE6/8AFE  WDCWRITE7/8B0A
 WDCWRITED/8B1A  WDCWRITEWAIT/8B24               WDCWRITEWAITLOOP/8B26           WDCQUIETERR/8B2F                WDCQUIET1/8B31
 WDCWRITEWAITEXIT/8B39           WDCFORMSERV/8B3B                WDCDONE/8B44    WDCREADSERV/8B4F                WDCREADLOOP/8B5B
 WDCREAD0/8B65   WDCREAD1/8B71   WDCREAD2/8B7D   WDCREAD3/8B89   WDCREAD4/8B95   WDCREAD5/8BA1   WDCREAD6/8BAD   WDCREAD7/8BB9
 WDCREADD/8BCB   WDCREADWAIT/8BD3                WDCREADWAITLOOP/8BD4            WDCREADWAITRTS/8BDF             WDCOUTDATA/8BE0
 WDCOUTDATAL/8BE9                WDCOUTDATA1/8BF6                WDCINDATA/8C07  WDCINDATA0/8C08                 WDCINDATA1/8C15
 WDCWAITRTS/8C1D                 WDCWAITAVAILABLE/8C1E           WDCWAITAVAILABLELOOP/8C21       WDCWAIT4INT/8C3C
 WDCWAIT4INT2/8C54               WDCWAIT4INT3/8C57               WDCINTUNEXPECTED/8C66           WDCINTERRUPT/8C69
 WDCPROCST/8C84  WDCPROCSTOKRTS/8C8D             WDCFATAL0/8C8E  WDCFATALERR/8C92               WDCFATAL2/8C94  JWDCCMDFEED/8CA3
 WDCSAVESTATUS/8CA6              WDCSAVEWRITESTATUS1/8CCC        WDCSAVEREADSTATUS/8CCD          WDCSAVEREADSTATUS1/8CD5
 WDCQUIT/8CD6    WDCQUITREAD/8CE4                WDCQUITWITHERR/8CE8             WDCTIMEDOUT/8CF2                WDCTIMEDOUT1/8CF2
 WDCSET4TRANS/8D01               ILPUTDEV:$FFC0/8D13             *RTS:$FFC0/8D1B ILGETDEV:$FFC0/8D1C
 TLCHECKREADY:$FFC0/8D25         ILPUTDEV:$FFC4/8D2C             *RTS:$FFC4/8D34 ILGETDEV:$FFC4/8D35
 TLCHECKREADY:$FFC4/8D3E         ILPUTDEV:$FFC8/8D45             *RTS:$FFC8/8D4D ILGETDEV:$FFC8/8D4E
 TLCHECKREADY:$FFC8/8D57         XLATEI:ADM3/8DC0                XLATEI:ADM3.B/8DCC              XLATEI:ADM3.DONE/8DCD
 SPECIALOUTPUT:ADM3/8DCF         SPECIALOUTPUT:ADM3POSN/8DD9     SPECIALOUTPUT:ADM3CLEAR/8DF5    NEXTDPB/8DFB   *PROFILECHAIN/8DFB
 THISDPB/8DFB    XLATEI:H19/8E11                 XLATEI:H19.B/8E1D               XLATEI:H19.OK/8E1E
 XLATEI:H19.ESCAPE/8E20         *XLATEI:H1932/8E4C               SPECIALOUTPUT:H19/8E50          SPECIALOUTPUT:H19POSN/8E5E
 SPECIALOUTPUT:H19CLEAR/8E7A     SPECIALOUTPUT:H19EEOL/8E84      COLORING:H19/8E8E               COLORING:H19REVERSEVIDEO/8E9A
 ILGETDEVSTATUSFROMACIA/8EA2     ILGETDEVICESTATUSFROMACIAERROR/8EA8              CNFGTABLE/8EB1  INTDISABLE/8EC4
*INTENABLE/8EC7 *INTRTI/8ECA     ILLDEVICEOP/8EDB                ERRETX/8EE0    *PATCHSPACE/8EE6
 STACKUNSWITCHEDDEVICEPOLL/8F18 *VT:INTERRUPTPOLLCHAIN/8F18      GOTOUTPUT:$FFC0/8F24            NOTOUTPUT:$FFC0/8F2C
 RTI:$FFC0/8F37  NOTDCDDROP:$FFC0/8F38           NOTINPUT:$FFC0/8F3F             NOINT:$FFC0/8F45
 GOTOUTPUT:$FFC4/8F51            NOTOUTPUT:$FFC4/8F59            RTI:$FFC4/8F64  NOTDCDDROP:$FFC4/8F65           NOTINPUT:$FFC4/8F6
 NOINT:$FFC4/8F72                GOTOUTPUT:$FFC8/8F7E            NOTOUTPUT:$FFC8/8F86            RTI:$FFC8/8F91
 NOTDCDDROP:$FFC8/8F92           NOTINPUT:$FFC8/8F99             NOINT:$FFC8/8F9F                STACKSWITCHEDDEVICEPOLL/8FA2
 WDCPOLL1/8FB3  *DISKINTSERVICE/8FBF             WDCPOLLNEXT/8FBF                PERSCIINTERRUPTMASK/8FC3
 DISKINTPERSCI.NO/8FCF           DAMFLOPPYINTERRUPTMASK/8FD3     DISKINTDAMFLOPPY.NO/8FDF        BADINTERRUPTCOUNT/8FE9
```

```
 CLOCKDCB/8FEB  *DEVICEDCBS/8FEB          NEXTDEVICEDCB/8FEB          CLOCKBUFFER/8FF4          DIV60DIVIDEND/8FF4
 DAY/8FF7       MONTH/8FF8     YEAR/8FF9     *CLOCKFRACTION/8FFA     CLOCKSTR/8FFB  TIME$/9002     TIME$:HOURS/9002
 TIME$:MINUTES/9005            TIME$:SECONDS/9008          DATE$/900B     DATE$:MONTH/900B          DATE$:DAY/900E
 DATE$:YEAR/9011               DISKINTDCB/9013             DISKINTCCB/9015           :CONTROLLER/9017
 CCB:PERSCI/9017               CCB:DAMFLOPPY/9045          FDTIMEOUTBLOCK/9069       :HEADCHAIN/9155
 :DCB/92B6      ::/9321        WDCINTERFACE/9325           WDCDCBPOINTER/9326        WDCCONTINUEPC/9328
 WDCCOUNT/932A  WDCPOINTER/932B          WDCRETRYCNT/932D          WDC1DCB/932E   WDC1STR/9372   DISKDCBS/9377
 NEXTDISKDCB/9377              WDC0DCB/9377   WDC0STR/93BB   WDCTIMEOUTBLOCK/93C0       WDCTIMEOUTCOUNT/93C2
 OUTBUF:$FFC0/93C6            *TTYBUFFERS/93C6             INBUF:$FFC0/9416          LINEBUF:$FFC0/9466
 OUTBUF:$FFC4/94CA             INBUF:$FFC4/951A            LINEBUF:$FFC4/951A        OUTBUF:$FFC8/951A
 INBUF:$FFC8/956A              LINEBUF:$FFC8/95BA          DCBNAME:$FFC0/961E        DCB:$FFC0/9627  TTYDCB/9627
 NEXTTIMEOUT/96C8             *OUTPUTTOBLK:$FFC0/96C8      TIMEOUTQUEUE/96C8         TTYTIMEOUTS/96C8
 INPUTTOBLK:$FFC0/96D0         TASKQUEUE/9724  TCB:$FFC0/9724  TTYTCB/9724    TCBSTACK:$FFC0/9756          STACK:$FFC0/975D
 DCBNAME:$FFC4/975E            DCB:$FFC4/9763  OUTPUTTOBLK:$FFC4/9804        INPUTTOBLK:$FFC4/980C        TCB:$FFC4/9860
 TCBSTACK:$FFC4/9892           STACK:$FFC4/9899            DCBNAME:$FFC8/989A        DCB:$FFC8/98A2
 OUTPUTTOBLK:$FFC8/9943        INPUTTOBLK:$FFC8/994B       TCB:$FFC8/999F  TCBSTACK:$FFC8/99D1          STACK:$FFC8/99D8
*FCBS/99D9      IOCBS/9C7D     IOCBPOINTERS/9D8D           INTERRUPTSTACK/9D9D       INTSETUP/9D9D  DSKBUFFERPOOL/9DE3
 FDRESTORE/9DE3  INTERRUPTSTACKEND/9DE3       WDCINIT/9E2C  CLOCKRESET/9E35          RESET:$FFC0/9E4B
 RESET:$FFC4/9E5A              RESET:$FFC8/9E69            VTDRIVER/A600  SDOS/BE00     *INTERRUPTTARGET/BE15
*INICV/FC03     *PUTCV/FC06    *GETCV/FC09    *TESTCV/FC0C  *INIDV/FC12    *PUTDV/FC15    STORAGEDEMONVIA/FF40
 VIADRB/FF40     VIADRA/FF41    VIADDRB/FF42   VIADDRA/FF43   VIAT1LL/FF44   VIAT1CH/FF45   VIAT1LLA/FF46  *VIAT1LH/FF47
 VIAACR/FF4B     VIAPCR/FF4C    VIAIFR/FF4D    VIAIER/FF4E   VIADRAF/FF4F   DAMFLOPPY:PIACA/FF80
 DAMFLOPPY:PIACB/FF81          DAMFLOPPY:PIADA/FF82        DAMFLOPPY:PIADB/FF83      DAMFLOPPY:WDCMDSTS/FF84
 DAMFLOPPY:WDTRACK/FF85        DAMFLOPPY:WDSECTOR/FF86     DAMFLOPPY:WDDATA/FF87     PERSCI:PIACA/FFA0
 PERSCI:PIACB/FFA1             PERSCI:PIADA/FFA2           PERSCI:PIADB/FFA3         PERSCI:WDCMDSTS/FFA4
 PERSCI:WDTRACK/FFA5           PERSCI:WDSECTOR/FFA6        PERSCI:WDDATA/FFA7        SDOS:VTCLEARIN/FFB8
 SDOS:VTCLEAROUT/FFBB          SDOS:VTTLPUTBUF/FFBE        SDOS:VTTLGETBUF/FFC1      SDOS:VTILPUTBUF/FFC4
 SDOS:VTILGETBUF/FFC7          SDOS:VTEDITTASK/FFCA        SDOS:VTMALLPT/FFCD        SDOS:VTMALVT/FFD0
 SDOS:VTATTNCHECK/FFD3         SDOS:VTINPUTTO/FFD6         SDOS:VTOUTPUTTO/FFD9      SDOS:VTINPUTINT/FFDC
 SDOS:VTOUTPUTINT/FFDF         SDOS:VTDISPATCH/FFE2       *SDOS:VTINTDCB/FFFE
```

1170 Symbols.

*** No Errors.