

```

J U  U P P P  I I I I  T T T T  E E E E  R R R R  B B B B  O O O  O O O  T T T T  L   P P P P  T T T T
J U  U P P  P I   T E   R R B B O O O  O T   L   P P P T
J U  U P P  P I   T E   R R B B O O O  O T   L   P P P T
J U  U P P P P  I   T E E E E E  R R R R  B B B B  O O O  O T   L   P P P P T
J U  U P   I   T E   R R B B O O O  O T   L   P P T
J  J U  U P   I   T E   R R B B O O O  O T  ..  L   P P T
J J J  U U U P   I I I I  T E E E E  R R B B B B  O O O  O T  ..  L L L L L P T

```

```

1  222      000 33333      000 55555      1      1  666      888 33333
11 2 2 2 ::: 0 0 3 ::: 0 0 5      11 / 11 6      / 8 8 3
1  2 2 ::: 0 00 3 ::: 0 00 5555      1 / 1 6      / 8 8 3
1  22      0 0 0 33      0 0 0 5      1 / 1 6666 / 888 33
1  2 2 ::: 00 0 3 ::: 00 0 5      1 / 1 6 6 / 8 8 3
1  2 2 ::: 0 0 3 3 ::: 0 0 5 5      1 / 1 6 6 / 8 8 3 3
11111 22222      000 333      000 555      11111      11111 666      888 333

```

MAL/6800 1.3F: 0000 SDOS BOOT  
01/16/83 12:01:39; Page 1; Form 1  
jupiterboot.asm

Jupiter II Boot Program for SDOS  
RAM & EQU'S

```
2:      NAME    SDOS BOOT
3: *    BOOT.SYS program for SDOS 1.0 or SDOS 1.1 6800/6809
4: *    Operates with Jupiter II 6800 and Percsi disk drive
5: *    Boots from IBM format disk
6: *    WARNING: THIS BOOT DOES NOT WORK IF OVERLAYED BY LOADED RECORDS!
7: *
0001    8:      IFUND  M6800
0001    9: M6800  EQU    1      ASSUME 6800
0000   10: M6801  EQU    0
0000   11: M6809  EQU    0
12:      FIN
13:
0001   14: PERSCI EQU    1
FFA0   15: FD1771DISK EQU $FFA0      BASE OF 1771 DMA DISK REGISTERS
FFA1   16: FD1771PIADONE EQU $FFA1
FFA2   17: FD1771DMAPAGE EQU $FFA2
FFA3   18: FD1771DRVSEL EQU $FFA3
FFA4   19: FD1771CMDSTS EQU $FFA4
FFA5   20: FD1771CURTRK EQU $FFA5
FFA6   21: FD1771SECTOR EQU $FFA6
FFA7   22: FD1771NEWTRK EQU $FFA7
23:
0080   24: NBPS   EQU    128    NUMBER OF BYTES PER SECTOR
0100   25: SECTORROFFSET EQU    $100
001A   26: NSPT   EQU    26
27: *
```

```

0011      29: BOOT:NSPC      EQU    $11
0016      30: BOOT:MAPAL6    EQU    $16
0040      31: BOOT:SIZE      EQU    $40
0000      32: LOAD:TYPE0     EQU    0
0000      33:      IF      M6809
*5*      34: LOAD:TYPE1     EQU    2
          35:      ELSE    (M6800)
0001      36: LOAD:TYPE1     EQU    1
          37:      FIN
0002      38: LOAD:TYPE2     EQU    2
0003      39: LOAD:TYPE3     EQU    3
FFFF      40:      IF      \m6809
*5*      41: DKRTS      EQU    $0C39
*5*      42: ERRORRTS     EQU    $0D39
          43:      Fin
          44:
0000      45:      ORG      0
0000 0080 46: IBMSECTOR RMB  NBPS      WHERE BOOT PROGRAM READS IBM SECTORS
          47:
0080      48:      ORG      $80      SO WE DON'T STEP ON SECTORS READ
          49:
          50: * STORAGE FOR DISK READ SUBROUTINE
0080 0002 51: READTARGET RMB  2      WHERE TO READ SECTOR INTO
          52:
          53: * OTHER STORAGE
0082 0002 54: TEMPX   RMB  2
0084 0002 55: CNT     RMB  2
0086 0002 56: DEST   RMB  2
0088 0002 57: SOURCE RMB  2
008A 0002 58: SECTOR RMB  2
008C 0002 59: COUNT  RMB  2
008E 0002 60: LOADADDRESS RMB  2
0090 0002 61: STARTAD RMB  2
0092 0002 62: DISKBUFFERPOINTER RMB  2
0094 0001 63: LSN     RMB  1
0095 0001 64: TRACK  RMB  1      FUNNY FOR READNEXTLSN
0096 001A 65: LSNTOPSNMAP RMB  NSPT
00B0      66: LSNTOPSNMAPEND EQU    *
00B0 0001 67: K1MODNSPT RMB  1
00B1 0001 68: K2MODNSPT RMB  1
00B2 0001 69: K4MODNSPT RMB  1
00B3 0001 70: K8MODNSPT RMB  1
00B4 0001 71: K16MODNSPT RMB  1
00B5 0001 72: K32MODNSPT RMB  1
          73:
00B6 002B 74:      RMB  40      STACK SPACE
00DE 0001 75: STACKBASE RMB  1
          76:
0F00      77:      ORG      $F00
0F00 0080 78: DISKBUFFER RMB  NBPS
0F80      79: DISKBUFFEREND EQU    *
          80:
1000      81:      ORG      $1000
1000      82: TRACKBUF EQU    *
1000 0D00 83:      RMB  NBPS*NSPT
    
```

```
0100      85:      ORG      $100
          86: *
          87: *      Assert: DP=0, CC=$FF when control is passed here by boot rom!
          88: *
0100 8E00DE  89: BOOT   LDS      #STACKBASE
0103 B60117  90:      LDAA   BOOT:MAPALG+SECTOROFFSET+1
0106 CE0180  91:      LDX    ##180
0109 8D35   92:      BSR    READIBMWITHINTRACK
010B 2073   93:      BRA    MAKEMAP
          94:
          95:      ORG      $140
          96: READIBMWITHINTRACK ; READ LOGICAL SECTOR 0..NSPT-1 WITHIN TRACK
0140 4C     97:      INCA           CONVERT LOGICAL TO IBM SECTOR NUMBERING
          98: READWITHINTRACK ; READS PHYSICAL SECTOR SPECIFIED BY (A)
          99: ; (X) POINTS TO DESIRED TARGET BUFFER
          100: ; ASSERT: HEADS ON PROPER TRACK, FD1771DRVSEL INITZD FOR DRIVE 0 AND "READ"
0141 43     101:      COMA           REMEMBER, 1771 IS INVERTED...
0142 B7FFA6  102:      STAA   FD1771SECTOR      TELL 1771 WHAT SECTOR TO READ
0145 B677   103:      LDAA   ##FF-$10001000    GET "READ" COMMAND (w/o headload)
          104: READWITHINTRACKAGAIN ; LOOP HERE TO RE-READ SECTOR
0147 7FFFA2  105:      CLR    FD1771DMAPAGE      TELL 1771 WHERE TO READ INTO
014A B7FFA4  106:      STAA   FD1771CMDSTS      TELL 1771 WHAT TO DO
          107: ; ALWAYS READ INTO PAGE ZERO!
          108: READWITHINTRACKWAITDONE ; WAIT FOR TRANSFER COMPLETE
014D F6FFA1  109:      LDAB   FD1771PIADONE      DONE YET ?
0150 2AFB   110:      BPL    READWITHINTRACKWAITDONE B/ NO
0152 F6FFA4  111:      LDAB   FD1771CMDSTS      GET 1771 READ STATUS
0155 F5FFA3  112:      BITB   FD1771DRVSEL      ACK PIA DONE BIT
0158 53     113:      COMB           CONVERT TO "TRUE" LOGIC
0159 C59F   114:      BITB   %Z10011111       PROPER COMPLETION ?
015B 26EA   115:      BNE    READWITHINTRACKAGAIN B/ NO, DO THE READ AGAIN!
          116: ; NOW COPY SECTOR FROM BUFFER TO TARGET, COMPLEMENTING WHILE COPYING!
          117: READWITHINTRACKCOPYLOOP ; COPY BYTES FROM DMA BUFFER TO TARGET, COMPLEMENTING!
          118: IBMSECTORPTR1 EQU *+1      THIS IS UGLY BUT FAST AND SMALL...
          119:      LDAA   IBMSECTOR      GET 2 BYTES
          120: IBMSECTORPTR2 EQU *+1
015F D601   121:      LDAB   IBMSECTOR+1
0161 43     122:      COMA           COMPLEMENT BYTES (STUPID DESIGNER!)
0162 53     123:      COMB
0163 A700   124:      STAA   0,X
0165 E701   125:      STAB   1,X
0167 08     126:      INX
0168 08     127:      INX
0169 7C015E  128:      INC    IBMSECTORPTR1
016C 7C015E  129:      INC    IBMSECTORPTR1
016F 7C0160  130:      INC    IBMSECTORPTR2
0172 7C0160  131:      INC    IBMSECTORPTR2
0175 2AE6   132:      BPL    READWITHINTRACKCOPYLOOP
0177 7F015E  133:      CLR    IBMSECTORPTR1
017A C601   134:      LDAB   #1
017C F70160  135:      STAB   IBMSECTORPTR2
017F 39     136:      RTS
```

MAL/6800 1.3F: 0180 SDDS BOOT  
01/16/83 12:01:39; Page 4; Form 1  
jupiterboot.asm

Jupiter II Boot Program for SDDS  
RAM & EQU'S

```
0180 4F      138: MAKEMAP CLRA
0181 B00117  139:      SUBA   BOOT:MAPAL6+SECTOR0OFFSET+1
0184 CE0096  140:      LDY   #LSNTOPSNMAP
0187 DF82    141:      STX   TEMPX
0189 BB0117  142: MAP1  ADDA   BOOT:MAPAL6+SECTOR0OFFSET+1
018C B01A    143: MAP2  SUBA   #NSPT
018E 24FC    144:      BCC   MAP2
0190 BB1A    145:      ADDA   #NSPT
0192 CE0095  146:      LDY   #LSNTOPSNMAP-1
0195 08      147: MAP3  INX
0196 9C82    148:      CPX   TEMPX
0198 2707    149:      BEQ   MAP4
019A A100    150:      CMPA  0,X
019C 26F7    151:      BNE   MAP3
019E 4C      152:      INCA
019F 20EB    153:      BRA   MAP2
01A1 A700    154: MAP4  STAA  0,X
01A3 08      155:      INX
01A4 DF82    156:      STX   TEMPX
01A6 8C00B0  157:      CPX   #LSNTOPSNMAPEND
01A9 26DE    158:      BNE   MAP1
159: *
160: *      READ IN REST OF BOOT
161: *
01AB 969B    162:      LDAA  LSNTOPSNMAP+2
01AD CE0200  163:      LDY   #SECTOR0OFFSET+2*NBPS
01B0 8D8E    164:      BSR   READIBMWITHINTRACK
01B2 9699    165:      LDAA  LSNTOPSNMAP+3
01B4 CE0280  166:      LDY   #SECTOR0OFFSET+3*NBPS
01B7 8D87    167:      BSR   READIBMWITHINTRACK
01B9 969A    168:      LDAA  LSNTOPSNMAP+4
01BB CE0300  169:      LDY   #SECTOR0OFFSET+4*NBPS
01BE 8D80    170:      BSR   READIBMWITHINTRACK
01C0 969B    171:      LDAA  LSNTOPSNMAP+5
01C2 CE0380  172:      LDY   #SECTOR0OFFSET+5*NBPS
01C5 8D0140  173:      JSR   READIBMWITHINTRACK
01CB 7E02C3  174:      JMP   BUILDSPIRAL
```

```

176: *
177: *      READNEXTLSN -- INTO MEMORY AT (X) ACCORDING TO MAP ALGORITHM
178: *
01CB 179: READNEXTLSN
01CB DF86 180: STX DEST REMEMBER WHERE TO READ
01CD 9694 181: LDAA LSN GET DESIRED LSN
01CF BD02B9 182: JSR MODNSPTA
01D2 8B96 183: ADDA #LSNTOFPSNMAP
01D4 9783 184: STAA TEMPX+1
01D6 7F0082 185: CLR TEMPX
01D9 D795 186: STAB TRACK SAVE TARGET TRACK NUMBER
01DB DE82 187: LDX TEMPX PERFORM MAPPING WITHIN TRACK
01DD 4F 188: CLRA
01DE 57 189: ASRB
01DF 2402 190: BCC KMAP1
01E1 98B0 191: ADDA K1MODNSPT
01E3 57 192: KMAP1 ASRB
01E4 2402 193: BCC KMAP2
01E6 98B1 194: ADDA K2MODNSPT
01E8 57 195: KMAP2 ASRB
01E9 2402 196: BCC KMAP3
01EB 98B2 197: ADDA K4MODNSPT
01ED 57 198: KMAP3 ASRB
01EE 2402 199: BCC KMAP4
01F0 98B3 200: ADDA K8MODNSPT
01F2 57 201: KMAP4 ASRB
01F3 2402 202: BCC KMAP5
01F5 98B4 203: ADDA K16MODNSPT
01F7 57 204: KMAP5 ASRB
01F8 2402 205: BCC KMAP6
01FA 98B5 206: ADDA K32MODNSPT
01FC AB00 207: KMAP6 ADDA 0,X
01FE BD02B9 208: JSR MODNSPTA
0201 36 209: PSHA
0202 9694 210: LDAA LSN
0204 7C0094 211: INC LSN FOR NEXT TIME...
0207 BD02B9 212: JSR MODNSPTA
020A 4D 213: TSTA START OF NEW TRACK???
020B 2635 214: BNE READFROMBUFFERPOOL B/ NO
020D 9695 215: LDAA TRACK GET DESIRED TRACK NUMBER
020F 43 216: COMA REMEMBER, 1771 REGISTERS ARE INVERTED
0210 B7FFA7 217: STAA FD1771NEWTRK
0213 B6E3 218: LDAA #(\Z00011100)&%FF SEEK TRACK
0215 B7FFA4 219: STAA FD1771CMDSTS
0218 220: WAITSEEKDONE
0218 B6FFA1 221: LDAA FD1771PIADONE
021B 2AFB 222: BPL WAITSEEKDONE
021D B6FFA4 223: LDAA FD1771CMDSTS GET RESULTS OF OPERATION
0220 B5FFA3 224: BITA FD1771DRVSEL ACK PIA DONE BIT
0223 43 225: COMA CONVERT TO TRUE STATUS
0224 8599 226: BITA #Z10011001 SEEK FAILURE ?
0226 26FE 227: BNE * B/ YES....(DIE)
228:
0228 CE1000 229: LDX #TRACKBUF READ A WHOLE TRACK, PLEASE...
022B 8601 230: LDAA #1 READ ODD NUMBERED SECTORS, START AT 1
    
```

```
022D 978A 231: STAA SECTOR
022F 8D2B 232: BSR READ8SECTORSEVERYOTHER
0231 8D2B 233: BSR READ4SECTORSEVERYOTHER
0233 8D2A 234: BSR READ1SECTOREVERYOTHER
235:
0235 CE1080 236: LDX #TRACKBUF+NBPS WHERE TO READ 1ST EVEN SECTOR
023B 8602 237: LDAA #2 READ EVEN NUMBERED SECTORS, START AT 2
023A 978A 238: STAA SECTOR
023C 8D1B 239: BSR READ8SECTORSEVERYOTHER
023E 8D1B 240: BSR READ4SECTORSEVERYOTHER
0240 8D1D 241: BSR READ1SECTOREVERYOTHER
242:
0242 243: READFROMBUFFERPOOL
0242 32 244: PULA
0243 978A 245: STAA SECTOR
0245 8610 246: LDAA #TRACKBUF/256
0247 C600 247: LDAB #TRACKBUF&#xFF
0249 7A008A 248: GETAD DEC SECTOR COMPUTE DISPLACEMENT INTO TRACK BUFFER
024C 2B2D 249: BMI COPYBYTES
024E CB80 250: ADDB #NBPS&#xFF
0250 8900 251: ADCA #NBPS/256
0252 20F5 252: BRA GETAD
253:
0254 52454144 254: FCC "READ8" MARKER SO IRA CAN PLANT BKPT
0259 255: READ8SECTORSEVERYOTHER ; READ 8 SECTORS, SKIPPING ONE PHYSICAL SECTOR BETWEEN
0259 8D00 256: BSR READ4SECTORSEVERYOTHER
025B 257: READ4SECTORSEVERYOTHER ; READ 4 SECTORS, SKIPPING ONE PHYSICAL SECTOR BETWEEN
025B 8D00 258: BSR READ2SECTORSEVERYOTHER
025D 259: READ2SECTORSEVERYOTHER ; READ 2 SECTORS, SKIPPING ONE PHYSICAL SECTOR BETWEEN
025D 8D00 260: BSR READ1SECTOREVERYOTHER
025F 261: READ1SECTOREVERYOTHER ; READ 1 SECTOR, THEN SKIP NEXT SECTOR
025F 968A 262: LDAA SECTOR GET DESIRED SECTOR
0261 8D0141 263: JSR READWITHINTRACK ISSUE THE READ REQUEST
0264 DF82 264: STX TEMPX ADVANCE POINTER TO SECTOR BUFFER
0266 9682 265: LDAA TEMPX
0268 D683 266: LDAB TEMPX+1
026A CB80 267: ADDB #NBPS&#xFF
026C 8900 268: ADCA #NBPS/256
026E 9782 269: STAA TEMPX
0270 D783 270: STAB TEMPX+1
0272 DEB2 271: LDX TEMPX
0274 7C008A 272: INC SECTOR ADVANCE SECTOR NUMBER DESIRED
0277 7C008A 273: INC SECTOR
027A 39 274: RTS
275:
027B 276: COPYBYTES
027B 9788 277: STAA SOURCE
027D D789 278: STAB SOURCE+1
027F 8620 279: LDAA #NBPS/4 SINCE EACH ITERATION COPIES 4 BYTES
0281 9784 280: STAA CNT
281:
0283 282: FCOPY ; ****MAGIC 25 US/BYTE COPY ROUTINE****
283: **** THIS ROUTINE COULD BE SPED UP FOR 6809!
0283 DE88 284: LDX SOURCE
0285 A600 285: LDAA 0,X
```

```
0287 E601      286:      LDAB    1,X
0289 EE02      287:      LDX     2,X
028B DF82      288:      STX     TEMPX
028D DE86      289:      LDX     DEST
028F A700      290:      STAA   0,X
0291 E701      291:      STAB   1,X
0293 9682      292:      LDAA   TEMPX
0295 D683      293:      LDAB   TEMPX+1
0297 A702      294:      STAA   2,X
0299 E703      295:      STAB   3,X
029B 9688      296:      LDAA   SOURCE NOW ADD 4 TO SOURCE & DEST
029D D689      297:      LDAB   SOURCE+1
029F CB04      298:      ADDB   #4
02A1 8900      299:      ADCA   #0
02A3 9788      300:      STAA   SOURCE
02A5 D789      301:      STAB   SOURCE+1
02A7 9686      302:      LDAA   DEST
02A9 D687      303:      LDAB   DEST+1
02AB CB04      304:      ADDB   #4
02AD 8900      305:      ADCA   #0
02AF 9786      306:      STAA   DEST
02B1 D787      307:      STAB   DEST+1
02B3 7A0084    308:      DEC    CNT
02B6 26CB      309:      BNE    FCOPY **** END OF FAST COPY ****
              310:
02B8 39        311:      RTS           END OF SECTOR TRANSFER
```



MAL/6800 1.3F: 0288 SDOS BOOT  
01/16/83 12:01:39; Page 8; Form 1  
jupiterboot.asm

Jupiter II Boot Program for SDOS  
RAM & EQU'S

313: \*  
314: \*  
315: \*  
316: \*  
317: \*

MODNSPTA:

B:=A/NSPT

A:=A MOD NSPT

02B9 5F	318: MODNSPTA	CLRB	SET QUOTIENT TO ZERO
02BA 5C	319: MODNSPTAL	INCB	BUMP QUOTIENT
02BB 801A	320: SUBA	#NSPT	
02BD 24FB	321: BCC	MODNSPTAL	
02BF 8B1A	322: ADDA	#NSPT	
02C1 5A	323: DECB		
02C2 39	324: RTS		

```
326: *  
327: *      BUILDSPIRAL--GENERATE SPIRALLING CONSTANSTS  
328: *  
02C3 B60116 329: BUILDSPIRAL   LDAA   BOOT:MAPAL6+SECTOR0ROFFSET  
02C6 BDF1   330:      BSR     MODNSPTA  
02C8 97B0   331:      STAA   K1MODNSPT  
02CA 48     332:      ASLA  
02CB BDEC   333:      BSR     MODNSPTA  
02CD 97B1   334:      STAA   K2MODNSPT  
02CF 48     335:      ASLA  
02D0 BDE7   336:      BSR     MODNSPTA  
02D2 97B2   337:      STAA   K4MODNSPT  
02D4 48     338:      ASLA  
02D5 BDE2   339:      BSR     MODNSPTA  
02D7 97B3   340:      STAA   K8MODNSPT  
02D9 48     341:      ASLA  
02DA BDDD   342:      BSR     MODNSPTA  
02DC 97B4   343:      STAA   K16MODNSPT  
02DE 48     344:      ASLA  
02DF BDD8   345:      BSR     MODNSPTA  
02E1 97B5   346:      STAA   K32MODNSPT
```

```
02E3 7F0094 348: LOAD CLR LSN FAKE OUT READNEXTSECTOR TO START THINGS OFF
02E6 CE0F00 349: LDY #DISKBUFFER
02E9 BD01CB 350: JSR READNEXTLSN AS GOOD A PLACE AS ANY
02EC B60111 351: LDAA BOOT:NSPC+SECTORROFFSET
02EF 9794 352: STAA LSN
02F1 CE0FB0 353: LDY #DISKBUFFEREND
02F4 DF92 354: STX DISKBUFFERPOINTER
02F6 BD038B 355: JSR GETBYTE
02F9 8101 356: CMPA #LOAD:TYPE1
02FB 26FE 357: BNE *
358: *
359: * GET TYPE 1--STARTADDRESS
360: *
02FD BD03A1 361: JSR GETWORD
0300 9790 362: STAA STARTAD
0302 D791 363: STAB STARTAD+1
0304 BD03A1 364: JSR GETWORD SEE IF COMPLEMENT CHECKS!
0307 0D 365: SEC
0308 D991 366: ADCB STARTAD+1
030A 26FE 367: BNE *
030C 9990 368: ADCA STARTAD
030E 26FE 369: BNE * B/ COMPLEMENT FAILURE
0310 2002 370: BRA LOADNEXT
371:
0312 372: LOADTYPE2
0312 BD2A 373: BSR LOAD2AND3
0314 374: LOADNEXT
0314 BD038B 375: JSR GETBYTE
0317 8103 376: CMPA #LOAD:TYPE3
0319 271D 377: BEQ LOADTYPE3
031B 8102 378: CMPA #LOAD:TYPE2
031D 27F3 379: BEQ LOADTYPE2
031F 8100 380: CMPA #LOAD:TYPE0
0321 26FE 381: BNE * BAD RECORD TYPE
382: *
383: * SKIP BYTES
384: *
0323 BD03A1 385: JSR GETWORD
0326 978C 386: STAA COUNT
0328 D78D 387: STAB COUNT+1
032A DEBC 388: LDY COUNT
032C 389: LOADTYPE0
032C DF8C 390: STX COUNT
032E 27E4 391: BEQ LOADNEXT DONE!
0330 BD038B 392: JSR GETBYTE
0333 DEBC 393: LDY COUNT
0335 09 394: DEX
0336 20F4 395: BRA LOADTYPE0
396: *
397: * LOAD AND QUIT
398: *
0338 399: LOADTYPE3
0338 BD04 400: BSR LOAD2AND3
033A DE90 401: LDY STARTAD
033C 6E00 402: JMP 0,X BYE-BYE!
```

```

403: *
404: *   GET BYTES FROM DISK
405: *
033E 406: LOAD2AND3
033E 8D61 407:   BSR   GETWORD
0340 978E 408:   STAA  LOADADDRESS
0342 D78F 409:   STAB  LOADADDRESS+1
0344 8D5B 410:   BSR   GETWORD
0346 978C 411:   STAA  COUNT
0348 D78D 412:   STAB  COUNT+1
034A     413: LOAD2AND3T
034A DE8C 414:   LDX   COUNT
034C 273C 415:   BEQ   LOAD2AND3RTS   B/ NOT DONE
416: *
417: *   PROCESS TYPE 2 OR 3 RECORD CONTENTS
418: *
034E 419: LOAD2AND3L   EQU   *
034E DE92 420:   LDX   DISKBUFFERPOINTER   IS DISK BUFFER EMPTY ?
0350 BC0F80 421:   CPX   #DISKBUFFEREND   ... ?
0353 2625 422:   BNE   LOAD2AND3L1   B/ NO, GET A BYTE FROM THE BUFFER
0355 968C 423:   LDAA  COUNT   YES, AT LEAST A SECTOR'S WORTH OF BYTES...
0357 D68D 424:   LDAB  COUNT+1 LEFT IN THIS LOAD RECORD ?
0359 C080 425:   SUBB  #NBPS&#xFF   ...?
035B 8200 426:   SBCA  #NBPS/256   ...?
035D 2517 427:   BCS   LOAD2AND3L2   B/ NOPE, PROCESS BYTES THE HARD WAY
035F 978C 428:   STAA  COUNT   YES, ADJUST COUNT AND ADDRESS
0361 D78D 429:   STAB  COUNT+1
0363 DE8E 430:   LDX   LOADADDRESS   WHERE TO READ THE SECTOR
0365 968E 431:   LDAA  LOADADDRESS   ADJUST LOAD ADDRESS PAST SECTOR'S WORTH OF BYTES
0367 D68F 432:   LDAB  LOADADDRESS+1
0369 CB80 433:   ADDB  #NBPS&#xFF
036B 8900 434:   ADCA  #NBPS/256
036D 978E 435:   STAA  LOADADDRESS
036F D78F 436:   STAB  LOADADDRESS+1
0371 BD01CB 437:   JSR   READNEXTLSN   GO DO OPTIMIZED LOAD RECORD TRANSFER
0374 20D4 438:   BRA   LOAD2AND3T   CHECK FOR DONE
439:
0376 440: LOAD2AND3L2   EQU   *
0376 8D1A 441:   BSR   GETBYTE0
0378 2002 442:   BRA   LOAD2AND3L1A
443:
037A 444: LOAD2AND3L1   EQU   *
037A 8D1F 445:   BSR   GETBYTE1
037C     446: LOAD2AND3L1A
037C DE8E 447:   LDX   LOADADDRESS
037E A700 448:   STAA  0,X
0380 08 449:   INX
0381 DF8E 450:   STX   LOADADDRESS
0383 DE8C 451:   LDX   COUNT
0385 09 452:   DEX
0386 DF8C 453:   STX   COUNT
0388 26CA 454:   BNE   LOAD2AND3L   MORE TO GO
038A 39 455: LOAD2AND3RTS   RTS
456: *
457: *   GET A BYTE
```

MAL/6800 1.3F: 038A SDOS BOOT  
01/16/83 12:01:39; Page 12; Form 1  
jupiterboot.asm

Jupiter II Boot Program for SDOS  
COMMON ROUTINES

```
458: *
038B DE92 459: GETBYTE LDX DISKBUFFERPOINTER
038D 8C0F80 460: CPX #DISKBUFFEREND
0390 2609 461: BNE GETBYTE1 B/ BUFFER NOT EMPTY
0392 462: GETBYTE0
0392 CE0F00 463: LDX #DISKBUFFER WHERE TO READ
0395 BD01CB 464: JSR READNEXTLSN
0398 CE0F00 465: LDX #DISKBUFFER
039B 466: GETBYTE1
039B A600 467: LDAA 0,X
039D 08 468: INX
039E DF92 469: STX DISKBUFFERPOINTER
03A0 39 470: RTS
03A1 BDE8 471: GETWORD BSR GETBYTE
03A3 36 472: PSHA
03A4 BDE5 473: BSR GETBYTE
03A6 16 474: TAB
03A7 32 475: PULA
03A8 39 476: RTS
0100 477: END BOOT
```

MAL/6800 1.3F: 03AB SDOS BOOT  
01/16/83 12:01:39; Page 13; Form 1  
jupiterboot.asm  
Symbols Sorted by Name:

Jupiter II Boot Program for SDOS  
Symbols Sorted by Name

BOOT/0100	BOOT:MAPAL6/0016	BOOT:NSPC/0011	*BOOT:SIZE/0040	BUILDSPIRAL/02C3	CNT/00B4	
COPYBYTES/027B	COUNT/008C	DEST/0086	DISKBUFFER/0F00	DISKBUFFEREND/0F80		
DISKBUFFERPDINTER/0092	FCOPY/0283	FD1771CMDSTS/FFA4	*FD1771CURTRK/FFA5	*FD1771DISK/FFA0		
FD1771DMPAGE/FFA2	FD1771DRVSEL/FFA3	FD1771NEWTRK/FFA7	FD1771PIADONE/FFA1			
FD1771SECTOR/FFA6	GETAD/0249	GETBYTE/038B	GETBYTE0/0392	GETBYTE1/039B	GETWORD/03A1	IBMSECTOR/0000
IBMSECTORPTR1/015E	IBMSECTORPTR2/0160	K16MODNSPT/00B4	K1MODNSPT/00B0	K2MODNSPT/00B1		
K32MODNSPT/00B5	K4MODNSPT/00B2	K8MODNSPT/00B3	KMAP1/01E3	KMAP2/01E8	KMAP3/01ED	KMAP4/01F2
KMAP5/01F7	KMAP6/01FC	*LOAD/02E3	LOAD2AND3/033E	LOAD2AND3L/034E	LOAD2AND3L1/037A	
LOAD2AND3L1A/037C	LOAD2AND3L2/0376	LOAD2AND3L2/0376	LOAD2AND3RTS/038A	LOAD2AND3T/034A		
LOAD:TYPE0/0000	LOAD:TYPE1/0001	LOAD:TYPE2/0002	LOAD:TYPE3/0003			
LOADADDRESS/008E	LOADNEXT/0314	LOADTYPE0/032C	LOADTYPE2/0312	LOADTYPE3/0338	LSN/0094	LSNTOPSNMAP/0096
LSNTOPSNMAPEND/00B0	*M6800/0001	*M6801/0000	M6809/0000	MAKEMAP/0180	MAP1/0189	MAP2/018C
MAP3/0195	MAP4/01A1	MODNSPTA/02B9	MODNSPTAL/02BA	NBPS/0080	NSPT/001A	*PERSCI/0001
READ1SECTOREVERYOTHER/025F	READ2SECTORSEVERYOTHER/025D	READ4SECTOREVERYOTHER/025B	READ8SECTOREVERYOTHER/0259			
READFROMBUFFERPOOL/0242	READIBMWITHINTRACK/0140	READNEXTLSN/01CB	*READTARGET/0080			
READWITHINTRACK/0141	READWITHINTRACKAGAIN/0147	READWITHINTRACKCOPYLOOP/015D	READWITHINTRACKWAITDONE/014D			
SECTOR/008A	SECTROFFSET/0100	SOURCE/008B	STACKBASE/00DE	STARTAD/0090	TEMPX/00B2	TRACK/0095
TRACKBUF/1000	WAITSEEKDONE/0218					

MAL/6800 1.3F: 03A8 SDDS BOOT  
01/16/83 12:01:39; Page 14; Form 1  
jupiterboot.asm  
Symbols Sorted by Value:

Jupiter II Boot Program for SDDS  
Symbols Sorted by Value

IBMSECTOR/0000	LOAD:TYPE0/0000	*M6801/0000	M6809/0000	LOAD:TYPE1/0001	*M6800/0001	
*PERSCI/0001	LOAD:TYPE2/0002	LOAD:TYPE3/0003		BOOT:NSPC/0011	BOOT:MAPALB/0016	
NSPT/001A	*BOOT:SIZE/0040	NBPS/00B0	*READTARGET/0080	TEMPX/00B2	CNT/00B4	DEST/0086
SOURCE/0088	SECTOR/008A	COUNT/008C	LOADADDRESS/008E	STARTAD/0090	DISKBUFFERPOINTER/0092	
LSN/0094	TRACK/0095	LSNTOPSNMAP/0096		K1MODNSPT/00B0	LSNTOPSNMAPEND/00B0	K2MODNSPT/00B1
K4MODNSPT/00B2	K8MODNSPT/00B3	K16MODNSPT/00B4		K32MODNSPT/00B5	STACKBASE/00DE	BOOT/0100
SECTOROFFSET/0100		READIBMWITHINTRACK/0140		READWITHINTRACK/0141	READWITHINTRACKAGAIN/0147	
READWITHINTRACKWAITDONE/014D		READWITHINTRACKCOPYLOOP/015D		IBMSECTORPTR1/015E	IBMSECTORPTR2/0160	
MAKEMAP/0180	MAP1/0189	MAP2/018C	MAP3/0195	MAP4/01A1	READNEXTLSN/01CB	KMAP1/01E3
KMAP2/01E8	KMAP3/01ED	KMAP4/01F2	KMAP5/01F7	KMAP6/01FC	WAITSEEKDONE/021B	
READFROMBUFFERPOOL/0242		GETAD/0249	READ8SECTORSEVERYOTHER/0259	READ4SECTORSEVERYOTHER/025B		
READ2SECTORSEVERYOTHER/025D		READ1SECTOREVERYOTHER/025F	COPYBYTES/027B	FCOPY/0283	MODNSPTA/02B9	MODNSPTAL/02BA
BUILDSPIRAL/02C3	*LOAD/02E3	LOADTYPE2/0312	LOADNEXT/0314	LOADTYPE0/032C	LOADTYPE3/0338	LOAD2AND3/033E
LOAD2AND3T/034A		LOAD2AND3L/034E	LOAD2AND3L2/0376		LOAD2AND3L1/037A	
LOAD2AND3L1A/037C		LOAD2AND3RTS/038A	GETBYTE/038B	GETBYTE0/0392	GETBYTE1/039B	GETWORD/03A1
DISKBUFFER/0F00		DISKBUFFEREND/0F80	TRACKBUF/1000	*FD1771DISK/FFA0		FD1771PIADDNE/FFA1
FD1771DMAPAGE/FFA2		FD1771DRVSEL/FFA3	FD1771CMDSTS/FFA4		*FD1771CURTRK/FFA5	
FD1771SECTOR/FFA6		FD1771NEWTRK/FFA7				

95 Symbols.

MAL/6800 1.3F: 03AB SDOS BOOT  
01/16/83 12:01:39; Page 15; Form 1  
jupiterboot.asm

Jupiter II Boot Program for SDOS  
Symbols Sorted by Value

\*\*\* No Errors.