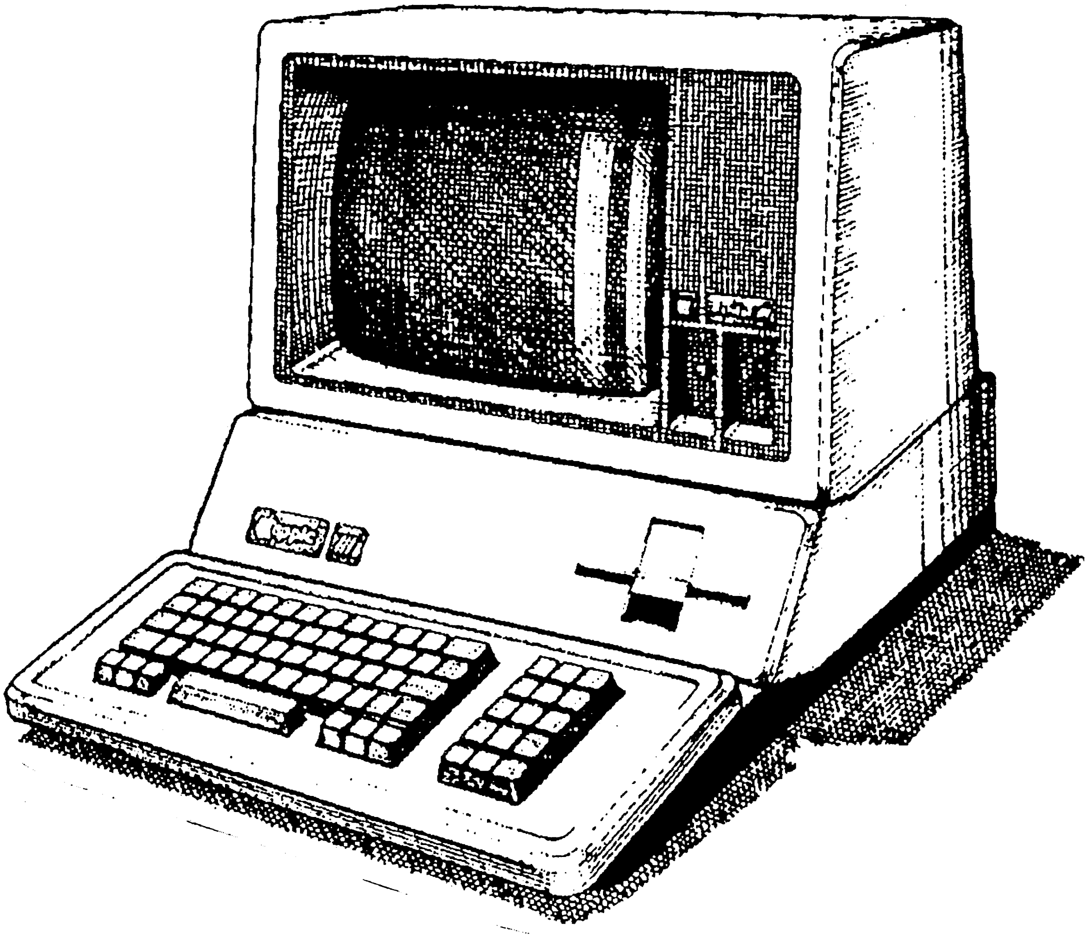
SEE DOC #193

Apple /// Computer Information



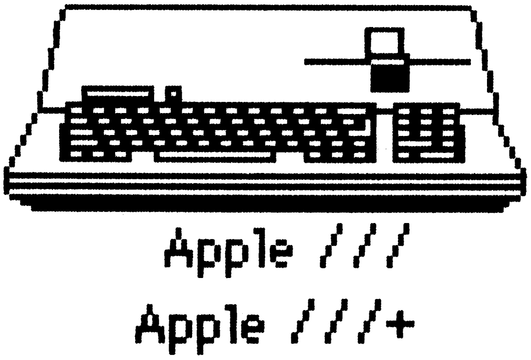
DOCUMENT NAME

SOS 1.3 FLOPPY BOOTSTRAP LOADER
SOURCE CODE LISTING

#
194

Ex Libris David T. Craig

"_188.PICT" 767 KB 2001-08-13 dpi: 600h x 600v pix: 4402h x 6048v



Apple /// SOS Technical Information

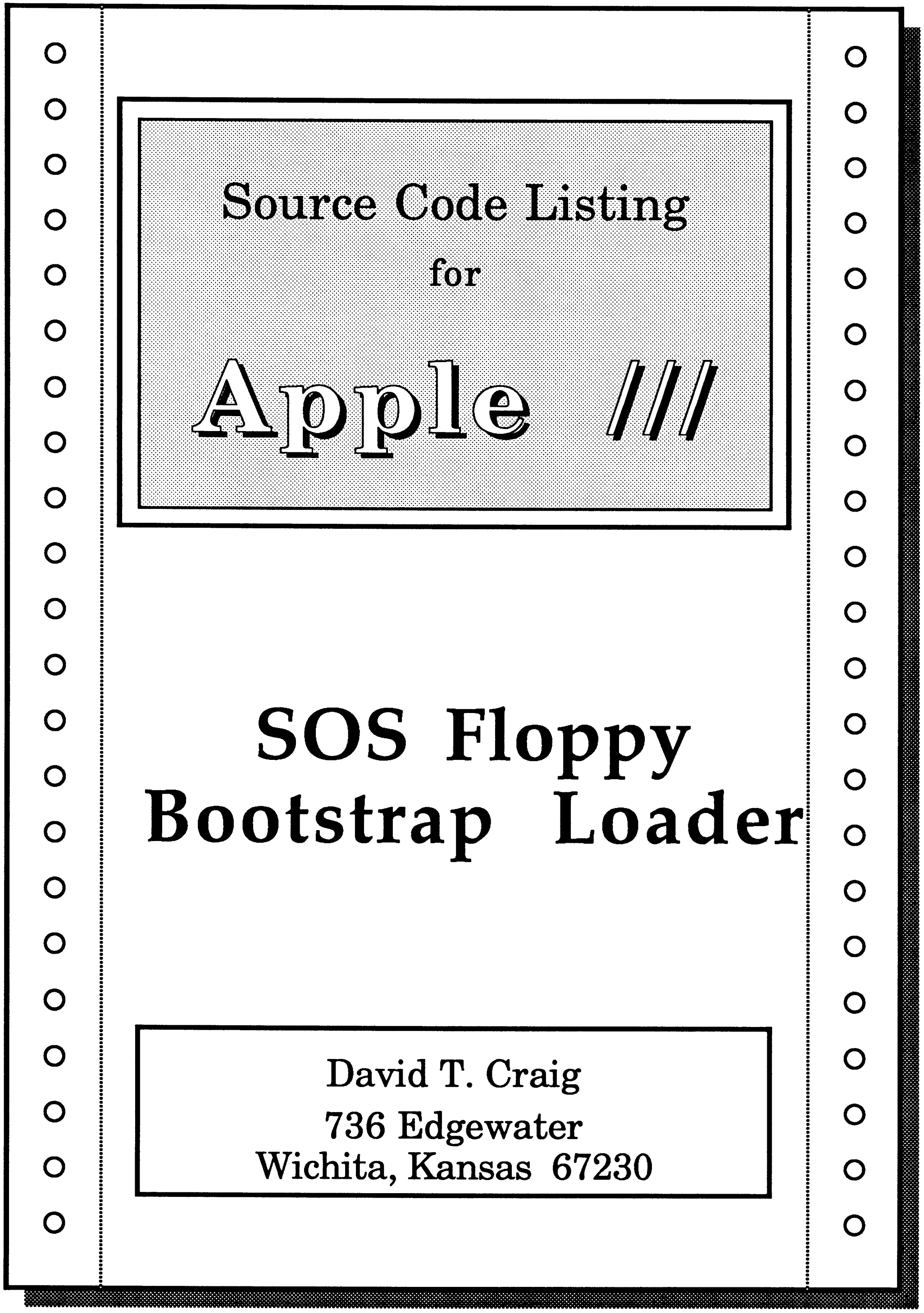
SOS 1.3 Floppy Bootstrap Loader Source Code Listing

This listing shows the code which is found at the beginning of a SOS boot disk. When the Apple /// computer starts the computer's ROM loads this code from the floppy disk and executes the code. This code loads the Apple ///'s operating system, SOS.

“_189.PICT” 411 KB 2001-08-13 dpi: 600h x 600v pix: 4320h x 4971v

Last version of BSLoader?.

DAVID T. CRAIG



Source Code Listing

for

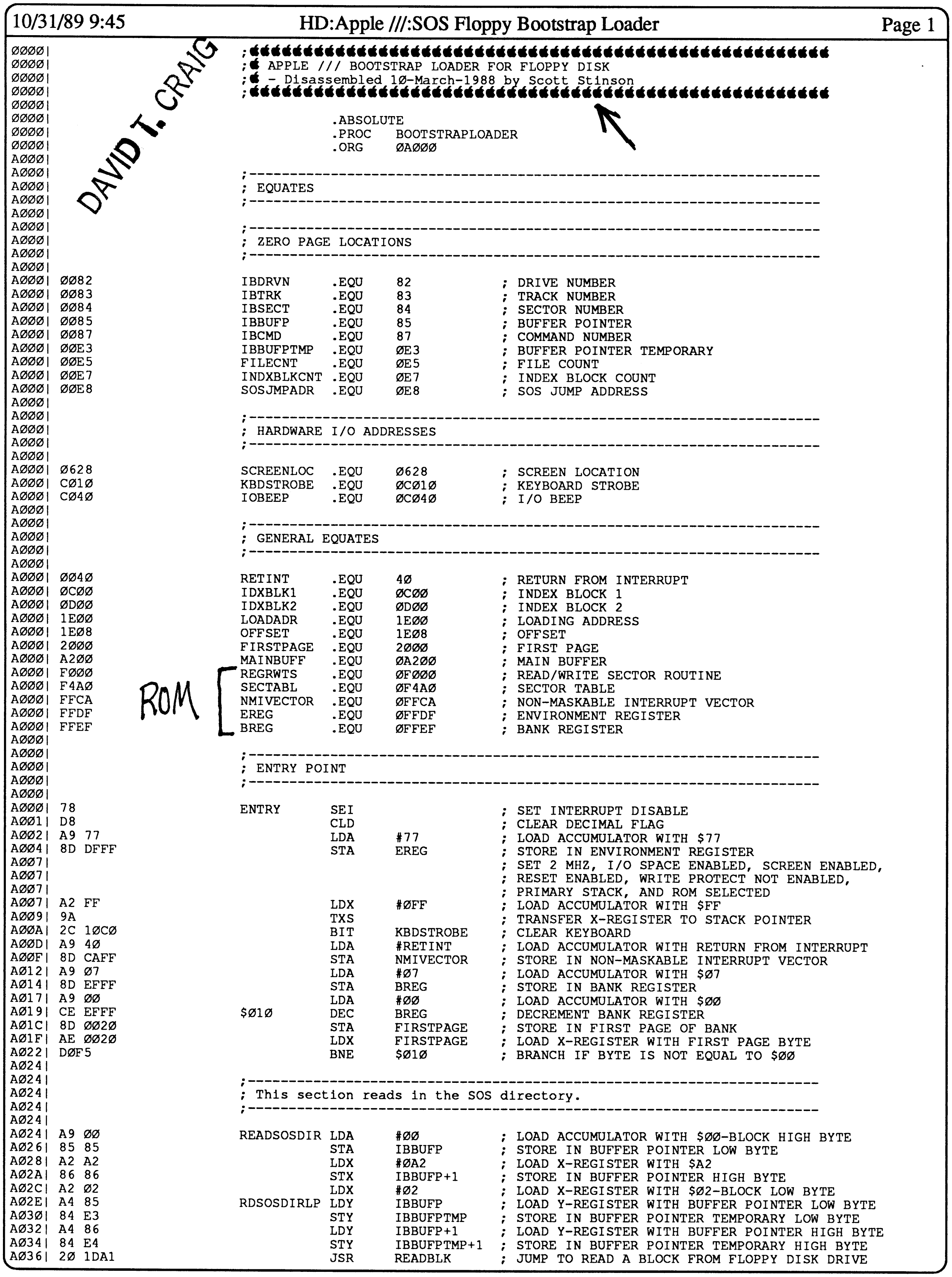
Apple ///

SOS Floppy Bootstrap Loader

David T. Craig
736 Edgewater
Wichita, Kansas 67230

Total pages: 5

"_190.PICT" 693 KB 2001-08-13 dpi: 600h x 600v pix: 3870h x 6415v

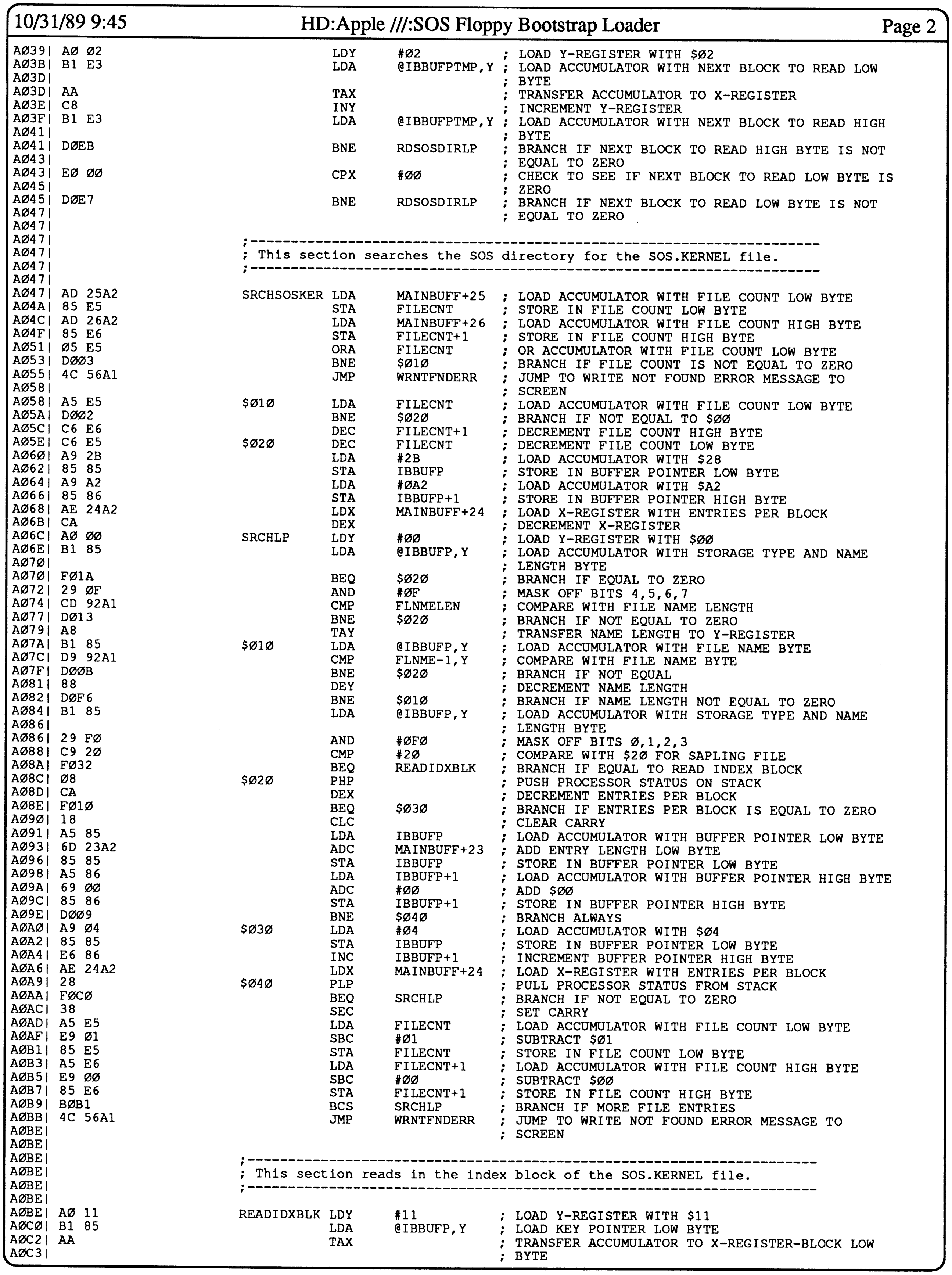


10/31/89 9:45 HD:Apple ///:SOS Floppy Bootstrap Loader Page 1

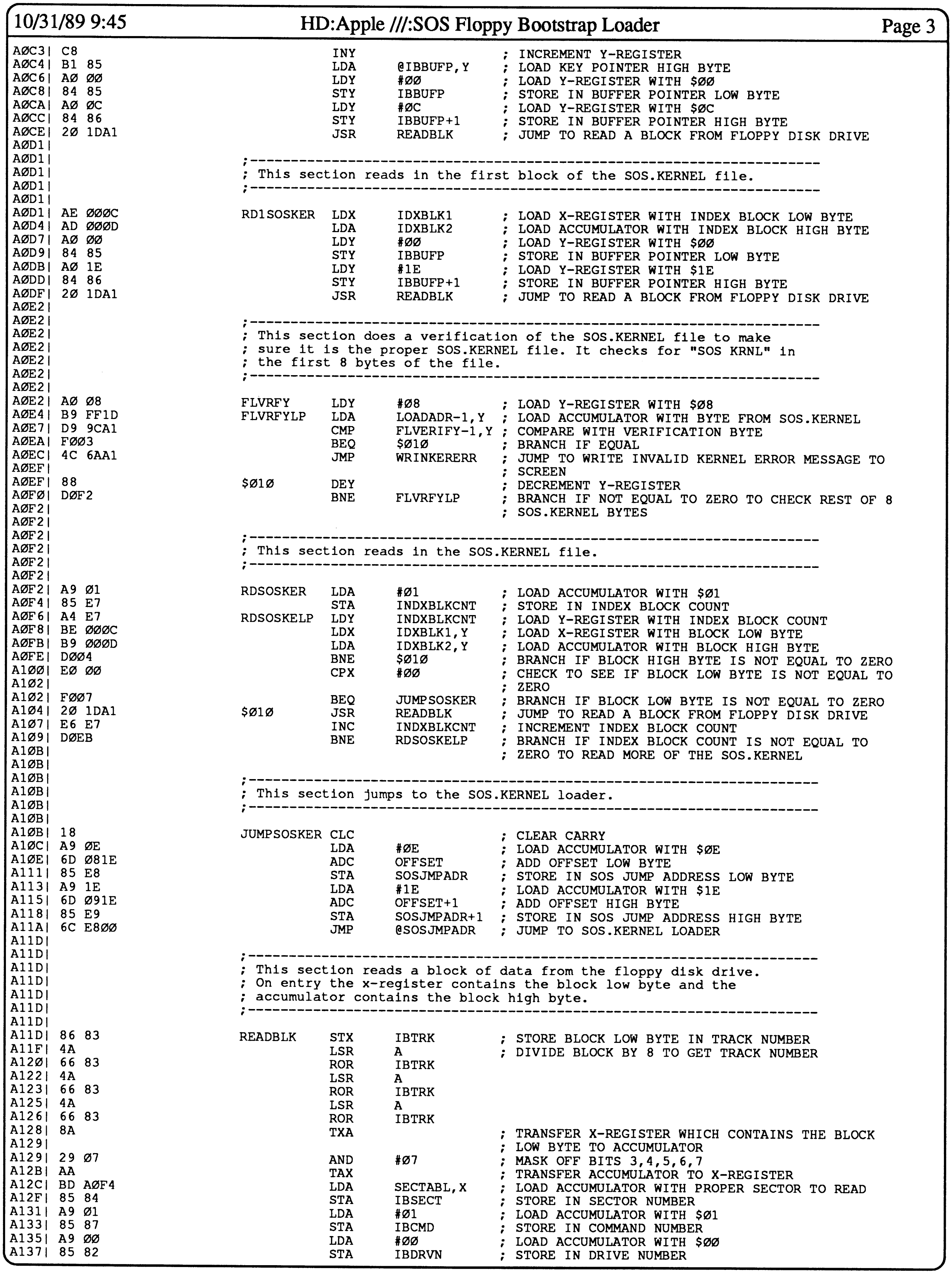
DAVID T. CRAIG

```
ØØØØ|                ;
ØØØØ|                ; APPLE /// BOOTSTRAP LOADER FOR FLOPPY DISK
ØØØØ|                ; - Disassembled 1Ø-March-1988 by Scott Stinson
ØØØØ|                ;
ØØØØ|
ØØØØ|                            .ABSOLUTE
ØØØØ|                            .PROC   BOOTSTRAPLOADER
ØØØØ|                            .ORG    ØAØØØ
AØØØ|
AØØØ|                ;------------------------------------------------------------------
AØØØ|                ; EQUATES
AØØØ|                ;------------------------------------------------------------------
AØØØ|
AØØØ|                ;------------------------------------------------------------------
AØØØ|                ; ZERO PAGE LOCATIONS
AØØØ|                ;------------------------------------------------------------------
AØØØ|
AØØØ| ØØ82           IBDRVN      .EQU    82           ; DRIVE NUMBER
AØØØ| ØØ83           IBTRK       .EQU    83           ; TRACK NUMBER
AØØØ| ØØ84           IBSECT      .EQU    84           ; SECTOR NUMBER
AØØØ| ØØ85           IBBUFP      .EQU    85           ; BUFFER POINTER
AØØØ| ØØ87           IBCMD       .EQU    87           ; COMMAND NUMBER
AØØØ| ØØE3           IBBUFPTMP   .EQU    ØE3          ; BUFFER POINTER TEMPORARY
AØØØ| ØØE5           FILECNT     .EQU    ØE5          ; FILE COUNT
AØØØ| ØØE7           INDXBLKCNT  .EQU    ØE7          ; INDEX BLOCK COUNT
AØØØ| ØØE8           SOSJMPADR   .EQU    ØE8          ; SOS JUMP ADDRESS
AØØØ|
AØØØ|                ;------------------------------------------------------------------
AØØØ|                ; HARDWARE I/O ADDRESSES
AØØØ|                ;------------------------------------------------------------------
AØØØ|
AØØØ| Ø628           SCREENLOC   .EQU    Ø628         ; SCREEN LOCATION
AØØØ| CØ1Ø           KBDSTROBE   .EQU    ØCØ1Ø        ; KEYBOARD STROBE
AØØØ| CØ4Ø           IOBEEP      .EQU    ØCØ4Ø        ; I/O BEEP
AØØØ|
AØØØ|                ;------------------------------------------------------------------
AØØØ|                ; GENERAL EQUATES
AØØØ|                ;------------------------------------------------------------------
AØØØ|
AØØØ| ØØ4Ø           RETINT      .EQU    4Ø           ; RETURN FROM INTERRUPT
AØØØ| ØCØØ           IDXBLK1     .EQU    ØCØØ         ; INDEX BLOCK 1
AØØØ| ØDØØ           IDXBLK2     .EQU    ØDØØ         ; INDEX BLOCK 2
AØØØ| 1EØØ           LOADADR     .EQU    1EØØ         ; LOADING ADDRESS
AØØØ| 1EØ8           OFFSET      .EQU    1EØ8         ; OFFSET
AØØØ| 2ØØØ           FIRSTPAGE   .EQU    2ØØØ         ; FIRST PAGE
AØØØ| A2ØØ           MAINBUFF    .EQU    ØA2ØØ        ; MAIN BUFFER
AØØØ| FØØØ           REGRWTS     .EQU    ØFØØØ        ; READ/WRITE SECTOR ROUTINE
AØØØ| F4AØ           SECTABL     .EQU    ØF4AØ        ; SECTOR TABLE
AØØØ| FFCA      ROM  NMIVECTOR   .EQU    ØFFCA        ; NON-MASKABLE INTERRUPT VECTOR
AØØØ| FFDF           EREG        .EQU    ØFFDF        ; ENVIRONMENT REGISTER
AØØØ| FFEF           BREG        .EQU    ØFFEF        ; BANK REGISTER
AØØØ|
AØØØ|                ;------------------------------------------------------------------
AØØØ|                ; ENTRY POINT
AØØØ|                ;------------------------------------------------------------------
AØØØ|
AØØØ| 78             ENTRY       SEI                  ; SET INTERRUPT DISABLE
AØØ1| D8                         CLD                  ; CLEAR DECIMAL FLAG
AØØ2| A9 77                      LDA     #77          ; LOAD ACCUMULATOR WITH $77
AØØ4| 8D DFFF                    STA     EREG         ; STORE IN ENVIRONMENT REGISTER
AØØ7|                                                 ; SET 2 MHZ, I/O SPACE ENABLED, SCREEN ENABLED,
AØØ7|                                                 ; RESET ENABLED, WRITE PROTECT NOT ENABLED,
AØØ7|                                                 ; PRIMARY STACK, AND ROM SELECTED
AØØ7| A2 FF                      LDX     #ØFF         ; LOAD ACCUMULATOR WITH $FF
AØØ9| 9A                         TXS                  ; TRANSFER X-REGISTER TO STACK POINTER
AØØA| 2C 1ØCØ                    BIT     KBDSTROBE    ; CLEAR KEYBOARD
AØØD| A9 4Ø                      LDA     #RETINT      ; LOAD ACCUMULATOR WITH RETURN FROM INTERRUPT
AØØF| 8D CAFF                    STA     NMIVECTOR    ; STORE IN NON-MASKABLE INTERRUPT VECTOR
AØ12| A9 Ø7                      LDA     #Ø7          ; LOAD ACCUMULATOR WITH $Ø7
AØ14| 8D EFFF                    STA     BREG         ; STORE IN BANK REGISTER
AØ17| A9 ØØ                      LDA     #ØØ          ; LOAD ACCUMULATOR WITH $ØØ
AØ19| CE EFFF        $Ø1Ø        DEC     BREG         ; DECREMENT BANK REGISTER
AØ1C| 8D ØØ2Ø                    STA     FIRSTPAGE    ; STORE IN FIRST PAGE OF BANK
AØ1F| AE ØØ2Ø                    LDX     FIRSTPAGE    ; LOAD X-REGISTER WITH FIRST PAGE BYTE
AØ22| DØF5                       BNE     $Ø1Ø         ; BRANCH IF BYTE IS NOT EQUAL TO $ØØ
AØ24|
AØ24|                ;------------------------------------------------------------------
AØ24|                ; This section reads in the SOS directory.
AØ24|                ;------------------------------------------------------------------
AØ24|
AØ24| A9 ØØ          READSOSDIR  LDA     #ØØ          ; LOAD ACCUMULATOR WITH $ØØ-BLOCK HIGH BYTE
AØ26| 85 85                      STA     IBBUFP       ; STORE IN BUFFER POINTER LOW BYTE
AØ28| A2 A2                      LDX     #ØA2         ; LOAD X-REGISTER WITH $A2
AØ2A| 86 86                      STX     IBBUFP+1     ; STORE IN BUFFER POINTER HIGH BYTE
AØ2C| A2 Ø2                      LDX     #Ø2          ; LOAD X-REGISTER WITH $Ø2-BLOCK LOW BYTE
AØ2E| A4 85          RDSOSDIRLP  LDY     IBBUFP       ; LOAD Y-REGISTER WITH BUFFER POINTER LOW BYTE
AØ3Ø| 84 E3                      STY     IBBUFPTMP    ; STORE IN BUFFER POINTER TEMPORARY LOW BYTE
AØ32| A4 86                      LDY     IBBUFP+1     ; LOAD Y-REGISTER WITH BUFFER POINTER HIGH BYTE
AØ34| 84 E4                      STY     IBBUFPTMP+1  ; STORE IN BUFFER POINTER TEMPORARY HIGH BYTE
AØ36| 2Ø 1DA1                    JSR     READBLK      ; JUMP TO READ A BLOCK FROM FLOPPY DISK DRIVE
```

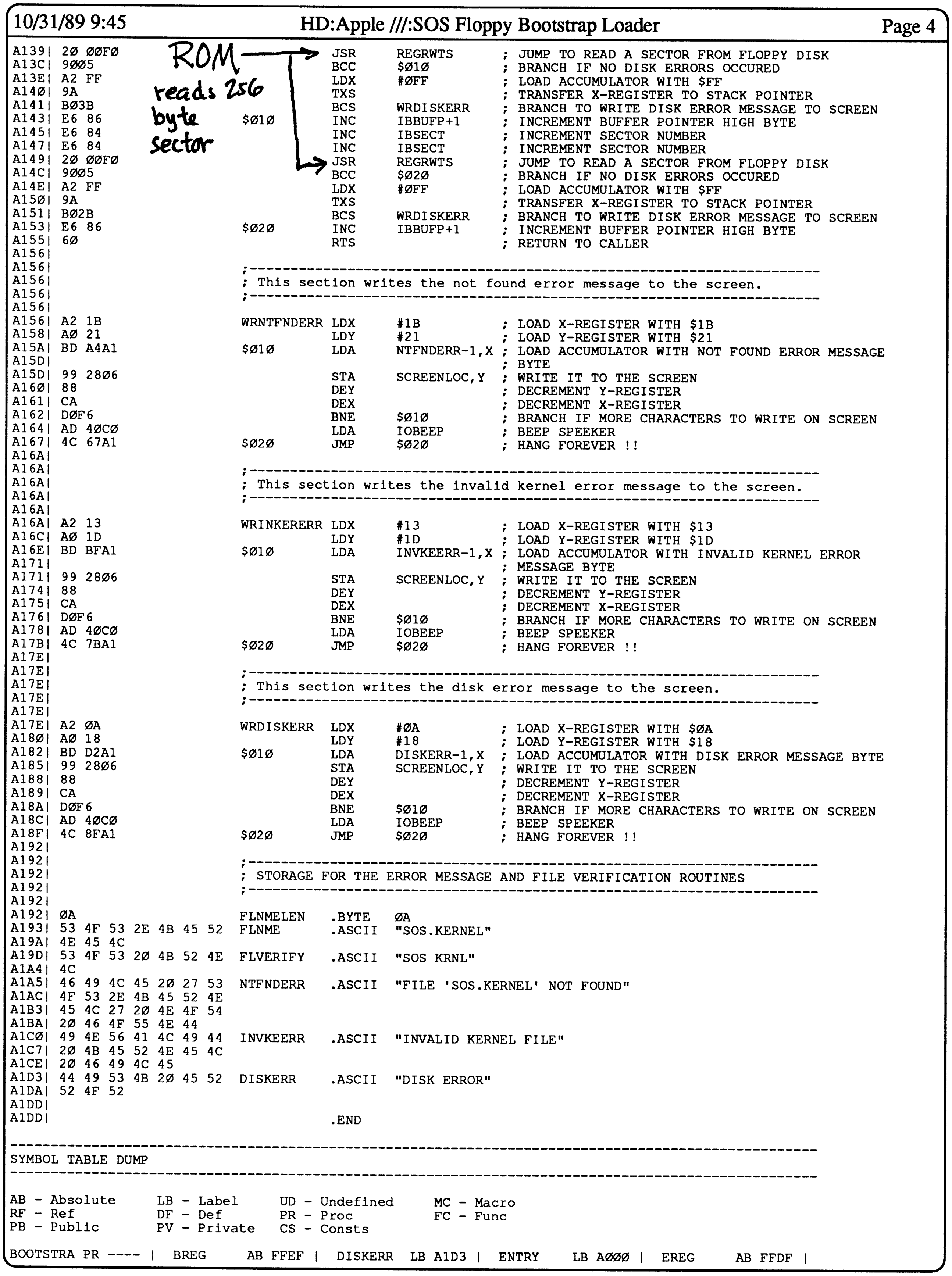
“_191.PICT” 779 KB 2001-08-13 dpi: 600h x 600v pix: 4655h x 6202v



```
AØ39| AØ Ø2                   LDY     #Ø2            ; LOAD Y-REGISTER WITH $Ø2
AØ3B| B1 E3                   LDA     @IBBUFPTMP,Y   ; LOAD ACCUMULATOR WITH NEXT BLOCK TO READ LOW
AØ3D|                                                ; BYTE
AØ3D| AA                      TAX                    ; TRANSFER ACCUMULATOR TO X-REGISTER
AØ3E| C8                      INY                    ; INCREMENT Y-REGISTER
AØ3F| B1 E3                   LDA     @IBBUFPTMP,Y   ; LOAD ACCUMULATOR WITH NEXT BLOCK TO READ HIGH
AØ41|                                                ; BYTE
AØ41| DØEB                    BNE     RDSOSDIRLP     ; BRANCH IF NEXT BLOCK TO READ HIGH BYTE IS NOT
AØ43|                                                ; EQUAL TO ZERO
AØ43| EØ ØØ                   CPX     #ØØ            ; CHECK TO SEE IF NEXT BLOCK TO READ LOW BYTE IS
AØ45|                                                ; ZERO
AØ45| DØE7                    BNE     RDSOSDIRLP     ; BRANCH IF NEXT BLOCK TO READ LOW BYTE IS NOT
AØ47|                                                ; EQUAL TO ZERO
AØ47|
AØ47|        ;-------------------------------------------------------------------
AØ47|        ; This section searches the SOS directory for the SOS.KERNEL file.
AØ47|        ;-------------------------------------------------------------------
AØ47|
AØ47| AD 25A2      SRCHSOSKER LDA     MAINBUFF+25    ; LOAD ACCUMULATOR WITH FILE COUNT LOW BYTE
AØ4A| 85 E5                   STA     FILECNT        ; STORE IN FILE COUNT LOW BYTE
AØ4C| AD 26A2                 LDA     MAINBUFF+26    ; LOAD ACCUMULATOR WITH FILE COUNT HIGH BYTE
AØ4F| 85 E6                   STA     FILECNT+1      ; STORE IN FILE COUNT HIGH BYTE
AØ51| Ø5 E5                   ORA     FILECNT        ; OR ACCUMULATOR WITH FILE COUNT LOW BYTE
AØ53| DØØ3                    BNE     $Ø1Ø           ; BRANCH IF FILE COUNT IS NOT EQUAL TO ZERO
AØ55| 4C 56A1                 JMP     WRNTFNDERR     ; JUMP TO WRITE NOT FOUND ERROR MESSAGE TO
AØ58|                                                ; SCREEN
AØ58| A5 E5      $Ø1Ø         LDA     FILECNT        ; LOAD ACCUMULATOR WITH FILE COUNT LOW BYTE
AØ5A| DØØ2                    BNE     $Ø2Ø           ; BRANCH IF NOT EQUAL TO $ØØ
AØ5C| C6 E6                   DEC     FILECNT+1      ; DECREMENT FILE COUNT HIGH BYTE
AØ5E| C6 E5      $Ø2Ø         DEC     FILECNT        ; DECREMENT FILE COUNT LOW BYTE
AØ6Ø| A9 2B                   LDA     #2B            ; LOAD ACCUMULATOR WITH $28
AØ62| 85 85                   STA     IBBUFP         ; STORE IN BUFFER POINTER LOW BYTE
AØ64| A9 A2                   LDA     #ØA2           ; LOAD ACCUMULATOR WITH $A2
AØ66| 85 86                   STA     IBBUFP+1       ; STORE IN BUFFER POINTER HIGH BYTE
AØ68| AE 24A2                 LDX     MAINBUFF+24    ; LOAD X-REGISTER WITH ENTRIES PER BLOCK
AØ6B| CA                      DEX                    ; DECREMENT X-REGISTER
AØ6C| AØ ØØ      SRCHLP       LDY     #ØØ            ; LOAD Y-REGISTER WITH $ØØ
AØ6E| B1 85                   LDA     @IBBUFP,Y      ; LOAD ACCUMULATOR WITH STORAGE TYPE AND NAME
AØ7Ø|                                                ; LENGTH BYTE
AØ7Ø| FØ1A                    BEQ     $Ø2Ø           ; BRANCH IF EQUAL TO ZERO
AØ72| 29 ØF                   AND     #ØF            ; MASK OFF BITS 4,5,6,7
AØ74| CD 92A1                 CMP     FLNMELEN       ; COMPARE WITH FILE NAME LENGTH
AØ77| DØ13                    BNE     $Ø2Ø           ; BRANCH IF NOT EQUAL TO ZERO
AØ79| A8                      TAY                    ; TRANSFER NAME LENGTH TO Y-REGISTER
AØ7A| B1 85      $Ø1Ø         LDA     @IBBUFP,Y      ; LOAD ACCUMULATOR WITH FILE NAME BYTE
AØ7C| D9 92A1                 CMP     FLNME-1,Y      ; COMPARE WITH FILE NAME BYTE
AØ7F| DØØB                    BNE     $Ø2Ø           ; BRANCH IF NOT EQUAL
AØ81| 88                      DEY                    ; DECREMENT NAME LENGTH
AØ82| DØF6                    BNE     $Ø1Ø           ; BRANCH IF NAME LENGTH NOT EQUAL TO ZERO
AØ84| B1 85                   LDA     @IBBUFP,Y      ; LOAD ACCUMULATOR WITH STORAGE TYPE AND NAME
AØ86|                                                ; LENGTH BYTE
AØ86| 29 FØ                   AND     #ØFØ           ; MASK OFF BITS Ø,1,2,3
AØ88| C9 2Ø                   CMP     #2Ø            ; COMPARE WITH $2Ø FOR SAPLING FILE
AØ8A| FØ32                    BEQ     READIDXBLK     ; BRANCH IF EQUAL TO READ INDEX BLOCK
AØ8C| Ø8         $Ø2Ø         PHP                    ; PUSH PROCESSOR STATUS ON STACK
AØ8D| CA                      DEX                    ; DECREMENT ENTRIES PER BLOCK
AØ8E| FØ1Ø                    BEQ     $Ø3Ø           ; BRANCH IF ENTRIES PER BLOCK IS EQUAL TO ZERO
AØ9Ø| 18                      CLC                    ; CLEAR CARRY
AØ91| A5 85                   LDA     IBBUFP         ; LOAD ACCUMULATOR WITH BUFFER POINTER LOW BYTE
AØ93| 6D 23A2                 ADC     MAINBUFF+23    ; ADD ENTRY LENGTH LOW BYTE
AØ96| 85 85                   STA     IBBUFP         ; STORE IN BUFFER POINTER LOW BYTE
AØ98| A5 86                   LDA     IBBUFP+1       ; LOAD ACCUMULATOR WITH BUFFER POINTER HIGH BYTE
AØ9A| 69 ØØ                   ADC     #ØØ            ; ADD $ØØ
AØ9C| 85 86                   STA     IBBUFP+1       ; STORE IN BUFFER POINTER HIGH BYTE
AØ9E| DØØ9                    BNE     $Ø4Ø           ; BRANCH ALWAYS
AØAØ| A9 Ø4      $Ø3Ø         LDA     #Ø4            ; LOAD ACCUMULATOR WITH $Ø4
AØA2| 85 85                   STA     IBBUFP         ; STORE IN BUFFER POINTER LOW BYTE
AØA4| E6 86                   INC     IBBUFP+1       ; INCREMENT BUFFER POINTER HIGH BYTE
AØA6| AE 24A2                 LDX     MAINBUFF+24    ; LOAD X-REGISTER WITH ENTRIES PER BLOCK
AØA9| 28         $Ø4Ø         PLP                    ; PULL PROCESSOR STATUS FROM STACK
AØAA| FØCØ                    BEQ     SRCHLP         ; BRANCH IF NOT EQUAL TO ZERO
AØAC| 38                      SEC                    ; SET CARRY
AØAD| A5 E5                   LDA     FILECNT        ; LOAD ACCUMULATOR WITH FILE COUNT LOW BYTE
AØAF| E9 Ø1                   SBC     #Ø1            ; SUBTRACT $Ø1
AØB1| 85 E5                   STA     FILECNT        ; STORE IN FILE COUNT LOW BYTE
AØB3| A5 E6                   LDA     FILECNT+1      ; LOAD ACCUMULATOR WITH FILE COUNT HIGH BYTE
AØB5| E9 ØØ                   SBC     #ØØ            ; SUBTRACT $ØØ
AØB7| 85 E6                   STA     FILECNT+1      ; STORE IN FILE COUNT HIGH BYTE
AØB9| BØB1                    BCS     SRCHLP         ; BRANCH IF MORE FILE ENTRIES
AØBB| 4C 56A1                 JMP     WRNTFNDERR     ; JUMP TO WRITE NOT FOUND ERROR MESSAGE TO
AØBE|                                                ; SCREEN
AØBE|
AØBE|        ;-------------------------------------------------------------------
AØBE|        ; This section reads in the index block of the SOS.KERNEL file.
AØBE|        ;-------------------------------------------------------------------
AØBE|
AØBE| AØ 11      READIDXBLK LDY     #11            ; LOAD Y-REGISTER WITH $11
AØCØ| B1 85                   LDA     @IBBUFP,Y      ; LOAD KEY POINTER LOW BYTE
AØC2| AA                      TAX                    ; TRANSFER ACCUMULATOR TO X-REGISTER-BLOCK LOW
AØC3|                                                ; BYTE
```



```
AØC3| C8                      INY                  ; INCREMENT Y-REGISTER
AØC4| B1 85                   LDA      @IBBUFP,Y   ; LOAD KEY POINTER HIGH BYTE
AØC6| AØ ØØ                   LDY      #ØØ         ; LOAD Y-REGISTER WITH $ØØ
AØC8| 84 85                   STY      IBBUFP      ; STORE IN BUFFER POINTER LOW BYTE
AØCA| AØ ØC                   LDY      #ØC         ; LOAD Y-REGISTER WITH $ØC
AØCC| 84 86                   STY      IBBUFP+1    ; STORE IN BUFFER POINTER HIGH BYTE
AØCE| 2Ø 1DA1                 JSR      READBLK     ; JUMP TO READ A BLOCK FROM FLOPPY DISK DRIVE
AØD1|
AØD1|              ;-------------------------------------------------------------------
AØD1|              ; This section reads in the first block of the SOS.KERNEL file.
AØD1|              ;-------------------------------------------------------------------
AØD1|
AØD1| AE ØØØC      RD1SOSKER  LDX      IDXBLK1     ; LOAD X-REGISTER WITH INDEX BLOCK LOW BYTE
AØD4| AD ØØØD                 LDA      IDXBLK2     ; LOAD ACCUMULATOR WITH INDEX BLOCK HIGH BYTE
AØD7| AØ ØØ                   LDY      #ØØ         ; LOAD Y-REGISTER WITH $ØØ
AØD9| 84 85                   STY      IBBUFP      ; STORE IN BUFFER POINTER LOW BYTE
AØDB| AØ 1E                   LDY      #1E         ; LOAD Y-REGISTER WITH $1E
AØDD| 84 86                   STY      IBBUFP+1    ; STORE IN BUFFER POINTER HIGH BYTE
AØDF| 2Ø 1DA1                 JSR      READBLK     ; JUMP TO READ A BLOCK FROM FLOPPY DISK DRIVE
AØE2|
AØE2|              ;-------------------------------------------------------------------
AØE2|              ; This section does a verification of the SOS.KERNEL file to make
AØE2|              ; sure it is the proper SOS.KERNEL file. It checks for "SOS KRNL" in
AØE2|              ; the first 8 bytes of the file.
AØE2|              ;-------------------------------------------------------------------
AØE2|
AØE2| AØ Ø8        FLVRFY     LDY      #Ø8         ; LOAD Y-REGISTER WITH $Ø8
AØE4| B9 FF1D      FLVRFYLP   LDA      LOADADR-1,Y ; LOAD ACCUMULATOR WITH BYTE FROM SOS.KERNEL
AØE7| D9 9CA1                 CMP      FLVERIFY-1,Y ; COMPARE WITH VERIFICATION BYTE
AØEA| FØØ3                    BEQ      $Ø1Ø        ; BRANCH IF EQUAL
AØEC| 4C 6AA1                 JMP      WRINKERERR  ; JUMP TO WRITE INVALID KERNEL ERROR MESSAGE TO
AØEF|                                              ; SCREEN
AØEF| 88           $Ø1Ø       DEY                  ; DECREMENT Y-REGISTER
AØFØ| DØF2                    BNE      FLVRFYLP    ; BRANCH IF NOT EQUAL TO ZERO TO CHECK REST OF 8
AØF2|                                              ; SOS.KERNEL BYTES
AØF2|
AØF2|              ;-------------------------------------------------------------------
AØF2|              ; This section reads in the SOS.KERNEL file.
AØF2|              ;-------------------------------------------------------------------
AØF2|
AØF2| A9 Ø1        RDSOSKER   LDA      #Ø1         ; LOAD ACCUMULATOR WITH $Ø1
AØF4| 85 E7                   STA      INDXBLKCNT  ; STORE IN INDEX BLOCK COUNT
AØF6| A4 E7        RDSOSKELP  LDY      INDXBLKCNT  ; LOAD Y-REGISTER WITH INDEX BLOCK COUNT
AØF8| BE ØØØC                 LDX      IDXBLK1,Y   ; LOAD X-REGISTER WITH BLOCK LOW BYTE
AØFB| B9 ØØØD                 LDA      IDXBLK2,Y   ; LOAD ACCUMULATOR WITH BLOCK HIGH BYTE
AØFE| DØØ4                    BNE      $Ø1Ø        ; BRANCH IF BLOCK HIGH BYTE IS NOT EQUAL TO ZERO
A1ØØ| EØ ØØ                   CPX      #ØØ         ; CHECK TO SEE IF BLOCK LOW BYTE IS NOT EQUAL TO
A1Ø2|                                              ; ZERO
A1Ø2| FØØ7                    BEQ      JUMPSOSKER  ; BRANCH IF BLOCK LOW BYTE IS NOT EQUAL TO ZERO
A1Ø4| 2Ø 1DA1      $Ø1Ø       JSR      READBLK     ; JUMP TO READ A BLOCK FROM FLOPPY DISK DRIVE
A1Ø7| E6 E7                   INC      INDXBLKCNT  ; INCREMENT INDEX BLOCK COUNT
A1Ø9| DØEB                    BNE      RDSOSKELP   ; BRANCH IF INDEX BLOCK COUNT IS NOT EQUAL TO
A1ØB|                                              ; ZERO TO READ MORE OF THE SOS.KERNEL
A1ØB|
A1ØB|              ;-------------------------------------------------------------------
A1ØB|              ; This section jumps to the SOS.KERNEL loader.
A1ØB|              ;-------------------------------------------------------------------
A1ØB|
A1ØB| 18           JUMPSOSKER CLC                  ; CLEAR CARRY
A1ØC| A9 ØE                   LDA      #ØE         ; LOAD ACCUMULATOR WITH $ØE
A1ØE| 6D Ø81E                 ADC      OFFSET      ; ADD OFFSET LOW BYTE
A111| 85 E8                   STA      SOSJMPADR   ; STORE IN SOS JUMP ADDRESS LOW BYTE
A113| A9 1E                   LDA      #1E         ; LOAD ACCUMULATOR WITH $1E
A115| 6D Ø91E                 ADC      OFFSET+1    ; ADD OFFSET HIGH BYTE
A118| 85 E9                   STA      SOSJMPADR+1 ; STORE IN SOS JUMP ADDRESS HIGH BYTE
A11A| 6C E8ØØ                 JMP      @SOSJMPADR  ; JUMP TO SOS.KERNEL LOADER
A11D|
A11D|              ;-------------------------------------------------------------------
A11D|              ; This section reads a block of data from the floppy disk drive.
A11D|              ; On entry the x-register contains the block low byte and the
A11D|              ; accumulator contains the block high byte.
A11D|              ;-------------------------------------------------------------------
A11D|
A11D| 86 83        READBLK    STX      IBTRK       ; STORE BLOCK LOW BYTE IN TRACK NUMBER
A11F| 4A                      LSR      A           ; DIVIDE BLOCK BY 8 TO GET TRACK NUMBER
A12Ø| 66 83                   ROR      IBTRK
A122| 4A                      LSR      A
A123| 66 83                   ROR      IBTRK
A125| 4A                      LSR      A
A126| 66 83                   ROR      IBTRK
A128| 8A                      TXA                  ; TRANSFER X-REGISTER WHICH CONTAINS THE BLOCK
A129|                                              ; LOW BYTE TO ACCUMULATOR
A129| 29 Ø7                   AND      #Ø7         ; MASK OFF BITS 3,4,5,6,7
A12B| AA                      TAX                  ; TRANSFER ACCUMULATOR TO X-REGISTER
A12C| BD AØF4                 LDA      SECTABL,X   ; LOAD ACCUMULATOR WITH PROPER SECTOR TO READ
A12F| 85 84                   STA      IBSECT      ; STORE IN SECTOR NUMBER
A131| A9 Ø1                   LDA      #Ø1         ; LOAD ACCUMULATOR WITH $Ø1
A133| 85 87                   STA      IBCMD       ; STORE IN COMMAND NUMBER
A135| A9 ØØ                   LDA      #ØØ         ; LOAD ACCUMULATOR WITH $ØØ
A137| 85 82                   STA      IBDRVN      ; STORE IN DRIVE NUMBER
```



10/31/89 9:45 HD:Apple ///:SOS Floppy Bootstrap Loader Page 4

ROM
reads 256
byte
sector

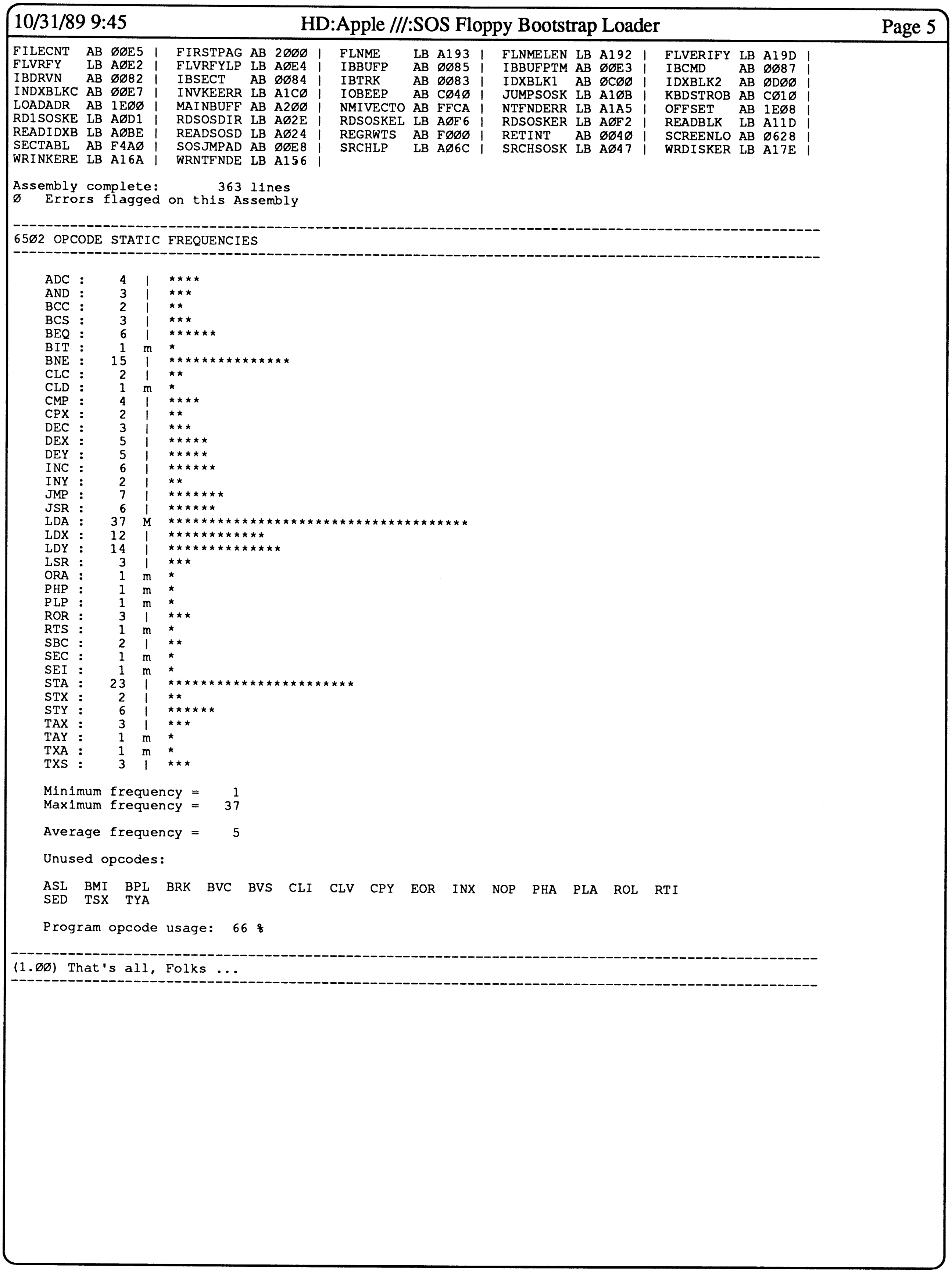
```
A139| 2Ø ØØFØ                         JSR      REGRWTS        ; JUMP TO READ A SECTOR FROM FLOPPY DISK
A13C| 9ØØ5                            BCC      $Ø1Ø           ; BRANCH IF NO DISK ERRORS OCCURED
A13E| A2 FF                           LDX      #ØFF           ; LOAD ACCUMULATOR WITH $FF
A14Ø| 9A                              TXS                     ; TRANSFER X-REGISTER TO STACK POINTER
A141| BØ3B                            BCS      WRDISKERR      ; BRANCH TO WRITE DISK ERROR MESSAGE TO SCREEN
A143| E6 86                $Ø1Ø       INC      IBBUFP+1       ; INCREMENT BUFFER POINTER HIGH BYTE
A145| E6 84                           INC      IBSECT         ; INCREMENT SECTOR NUMBER
A147| E6 84                           INC      IBSECT         ; INCREMENT SECTOR NUMBER
A149| 2Ø ØØFØ                         JSR      REGRWTS        ; JUMP TO READ A SECTOR FROM FLOPPY DISK
A14C| 9ØØ5                            BCC      $Ø2Ø           ; BRANCH IF NO DISK ERRORS OCCURED
A14E| A2 FF                           LDX      #ØFF           ; LOAD ACCUMULATOR WITH $FF
A15Ø| 9A                              TXS                     ; TRANSFER X-REGISTER TO STACK POINTER
A151| BØ2B                            BCS      WRDISKERR      ; BRANCH TO WRITE DISK ERROR MESSAGE TO SCREEN
A153| E6 86                $Ø2Ø       INC      IBBUFP+1       ; INCREMENT BUFFER POINTER HIGH BYTE
A155| 6Ø                              RTS                     ; RETURN TO CALLER
A156|
A156|                      ;------------------------------------------------------------------
A156|                      ; This section writes the not found error message to the screen.
A156|                      ;------------------------------------------------------------------
A156|
A156| A2 1B                WRNTFNDERR LDX      #1B            ; LOAD X-REGISTER WITH $1B
A158| AØ 21                           LDY      #21            ; LOAD Y-REGISTER WITH $21
A15A| BD A4A1              $Ø1Ø       LDA      NTFNDERR-1,X   ; LOAD ACCUMULATOR WITH NOT FOUND ERROR MESSAGE
A15D|                                                         ; BYTE
A15D| 99 28Ø6                         STA      SCREENLOC,Y    ; WRITE IT TO THE SCREEN
A16Ø| 88                              DEY                     ; DECREMENT Y-REGISTER
A161| CA                              DEX                     ; DECREMENT X-REGISTER
A162| DØF6                            BNE      $Ø1Ø           ; BRANCH IF MORE CHARACTERS TO WRITE ON SCREEN
A164| AD 4ØCØ                         LDA      IOBEEP         ; BEEP SPEEKER
A167| 4C 67A1              $Ø2Ø       JMP      $Ø2Ø           ; HANG FOREVER !!
A16A|
A16A|                      ;------------------------------------------------------------------
A16A|                      ; This section writes the invalid kernel error message to the screen.
A16A|                      ;------------------------------------------------------------------
A16A|
A16A| A2 13                WRINKERERR LDX      #13            ; LOAD X-REGISTER WITH $13
A16C| AØ 1D                           LDY      #1D            ; LOAD Y-REGISTER WITH $1D
A16E| BD BFA1              $Ø1Ø       LDA      INVKEERR-1,X   ; LOAD ACCUMULATOR WITH INVALID KERNEL ERROR
A171|                                                         ; MESSAGE BYTE
A171| 99 28Ø6                         STA      SCREENLOC,Y    ; WRITE IT TO THE SCREEN
A174| 88                              DEY                     ; DECREMENT Y-REGISTER
A175| CA                              DEX                     ; DECREMENT X-REGISTER
A176| DØF6                            BNE      $Ø1Ø           ; BRANCH IF MORE CHARACTERS TO WRITE ON SCREEN
A178| AD 4ØCØ                         LDA      IOBEEP         ; BEEP SPEEKER
A17B| 4C 7BA1              $Ø2Ø       JMP      $Ø2Ø           ; HANG FOREVER !!
A17E|
A17E|                      ;------------------------------------------------------------------
A17E|                      ; This section writes the disk error message to the screen.
A17E|                      ;------------------------------------------------------------------
A17E|
A17E| A2 ØA                WRDISKERR  LDX      #ØA            ; LOAD X-REGISTER WITH $ØA
A18Ø| AØ 18                           LDY      #18            ; LOAD Y-REGISTER WITH $18
A182| BD D2A1              $Ø1Ø       LDA      DISKERR-1,X    ; LOAD ACCUMULATOR WITH DISK ERROR MESSAGE BYTE
A185| 99 28Ø6                         STA      SCREENLOC,Y    ; WRITE IT TO THE SCREEN
A188| 88                              DEY                     ; DECREMENT Y-REGISTER
A189| CA                              DEX                     ; DECREMENT X-REGISTER
A18A| DØF6                            BNE      $Ø1Ø           ; BRANCH IF MORE CHARACTERS TO WRITE ON SCREEN
A18C| AD 4ØCØ                         LDA      IOBEEP         ; BEEP SPEEKER
A18F| 4C 8FA1              $Ø2Ø       JMP      $Ø2Ø           ; HANG FOREVER !!
A192|
A192|                      ;------------------------------------------------------------------
A192|                      ; STORAGE FOR THE ERROR MESSAGE AND FILE VERIFICATION ROUTINES
A192|                      ;------------------------------------------------------------------
A192|
A192| ØA                   FLNMELEN   .BYTE    ØA
A193| 53 4F 53 2E 4B 45 52 FLNME      .ASCII   "SOS.KERNEL"
A19A| 4E 45 4C
A19D| 53 4F 53 2Ø 4B 52 4E FLVERIFY   .ASCII   "SOS KRNL"
A1A4| 4C
A1A5| 46 49 4C 45 2Ø 27 53 NTFNDERR   .ASCII   "FILE 'SOS.KERNEL' NOT FOUND"
A1AC| 4F 53 2E 4B 45 52 4E
A1B3| 45 4C 27 2Ø 4E 4F 54
A1BA| 2Ø 46 4F 55 4E 44
A1CØ| 49 4E 56 41 4C 49 44 INVKEERR   .ASCII   "INVALID KERNEL FILE"
A1C7| 2Ø 4B 45 52 4E 45 4C
A1CE| 2Ø 46 49 4C 45
A1D3| 44 49 53 4B 2Ø 45 52 DISKERR    .ASCII   "DISK ERROR"
A1DA| 52 4F 52
A1DD|
A1DD|                                 .END

-----------------------------------------------------------------------------------------------
SYMBOL TABLE DUMP
-----------------------------------------------------------------------------------------------

AB - Absolute     LB - Label     UD - Undefined     MC - Macro
RF - Ref          DF - Def       PR - Proc          FC - Func
PB - Public       PV - Private   CS - Consts

BOOTSTRA PR ---- |  BREG     AB FFEF |  DISKERR  LB A1D3 |  ENTRY    LB AØØØ |  EREG     AB FFDF |
```

"_194.PICT" 752 KB 2001-08-13 dpi: 600h x 600v pix: 4679h x 6238v



```
FILECNT  AB ØØE5 |  FIRSTPAG AB 2ØØØ |  FLNME    LB A193 |  FLNMELEN LB A192 |  FLVERIFY LB A19D |
FLVRFY   LB AØE2 |  FLVRFYLP LB AØE4 |  IBBUFP   AB ØØ85 |  IBBUFPTM AB ØØE3 |  IBCMD    AB ØØ87 |
IBDRVN   AB ØØ82 |  IBSECT   AB ØØ84 |  IBTRK    AB ØØ83 |  IDXBLK1  AB ØCØØ |  IDXBLK2  AB ØDØØ |
INDXBLKC AB ØØE7 |  INVKEERR LB A1CØ |  IOBEEP   AB CØ4Ø |  JUMPSOSK LB A1ØB |  KBDSTROB AB CØ1Ø |
LOADADR  AB 1EØØ |  MAINBUFF AB A2ØØ |  NMIVECTO AB FFCA |  NTFNDERR LB A1A5 |  OFFSET   AB 1EØ8 |
RD1SOSKE LB AØD1 |  RDSOSDIR LB AØ2E |  RDSOSKEL LB AØF6 |  RDSOSKER LB AØF2 |  READBLK  LB A11D |
READIDXB LB AØBE |  READSOSD LB AØ24 |  REGRWTS  AB FØØØ |  RETINT   AB ØØ4Ø |  SCREENLO AB Ø628 |
SECTABL  AB F4AØ |  SOSJMPAD AB ØØE8 |  SRCHLP   LB AØ6C |  SRCHSOSK LB AØ47 |  WRDISKER LB A17E |
WRINKERE LB A16A |  WRNTFNDE LB A156 |

Assembly complete:       363 lines
Ø   Errors flagged on this Assembly

----------------------------------------------------------------------------------------------------
65Ø2 OPCODE STATIC FREQUENCIES
----------------------------------------------------------------------------------------------------

    ADC :    4  |  ****
    AND :    3  |  ***
    BCC :    2  |  **
    BCS :    3  |  ***
    BEQ :    6  |  ******
    BIT :    1  m  *
    BNE :   15  |  ***************
    CLC :    2  |  **
    CLD :    1  m  *
    CMP :    4  |  ****
    CPX :    2  |  **
    DEC :    3  |  ***
    DEX :    5  |  *****
    DEY :    5  |  *****
    INC :    6  |  ******
    INY :    2  |  **
    JMP :    7  |  *******
    JSR :    6  |  ******
    LDA :   37  M  *************************************
    LDX :   12  |  ************
    LDY :   14  |  **************
    LSR :    3  |  ***
    ORA :    1  m  *
    PHP :    1  m  *
    PLP :    1  m  *
    ROR :    3  |  ***
    RTS :    1  m  *
    SBC :    2  |  **
    SEC :    1  m  *
    SEI :    1  m  *
    STA :   23  |  ***********************
    STX :    2  |  **
    STY :    6  |  ******
    TAX :    3  |  ***
    TAY :    1  m  *
    TXA :    1  m  *
    TXS :    3  |  ***

    Minimum frequency =    1
    Maximum frequency =   37

    Average frequency =    5

    Unused opcodes:

    ASL  BMI  BPL  BRK  BVC  BVS  CLI  CLV  CPY  EOR  INX  NOP  PHA  PLA  ROL  RTI
    SED  TSX  TYA

    Program opcode usage:  66 %

----------------------------------------------------------------------------------------------------
(1.ØØ) That's all, Folks ...
----------------------------------------------------------------------------------------------------
```

"_195.PICT" 411 KB 2001-08-13 dpi: 600h x 600v pix: 4678h x 6249v

 Apple /// Computer Information

APPLE /// SOS BOOTSTRAP LOADER HEXADECIMAL DUMP

Source
DISK1.dofile as found with Chris Smolinski's Macintosh SARA emulator application

Printed by David T. Craig • December 1997

This hex dump, which was produced by the Apple Macintosh MPW DumpFile tool, lists the Apple /// SOS Bootstrap Loader. This 512 byte loader exists at block 0 of SOS disks and is loaded by the Apple /// ROM into memory addresses $A000-$A1FF. This code's purpose is to begin the loading of SOS from the floppy disk into the ///'s memory.

```
   0: 4C 6E A0 53 4F 53 20 42 4F 4F 54 20 20 31 2E 31 Ln†SOS.BOOT..1.1
  10: 20 0A 53 4F 53 2E 4B 45 52 4E 45 4C 20 20 20 20 ..SOS.KERNEL....
  20: 20 53 4F 53 20 4B 52 4E 4C 49 2F 4F 20 45 52 52 .SOS.KRNLI/O.ERR
  30: 4F 52 08 00 46 49 4C 45 20 27 53 4F 53 2E 4B 45 OR..FILE.'SOS.KE
  40: 52 4E 45 4C 27 20 4E 4F 54 20 46 4F 55 4E 44 25 RNEL'.NOT.FOUND%
  50: 00 49 4E 56 41 4C 49 44 20 4B 45 52 4E 45 4C 20 .INVALID.KERNEL.
  60: 46 49 4C 45 3A 00 00 0C 00 1E 0E 1E 04 A4 78 D8 FILE:........§xÿ
  70: A9 77 8D DF FF A2 FB 9A 2C 10 C0 A9 40 8D CA FF ©wçflˇ¢˚ö,.¿©@ç ˇ
  80: A9 07 8D EF FF A2 00 CE EF FF 8E 00 20 AD 00 20 ©.çÔˇ¢.ŒÔˇé..≠..
  90: D0 F5 A9 01 85 E0 A9 00 85 E1 A9 00 85 85 A9 A2 –ı©.Ö‡©.Ö·©.ÖÖ©¢
  A0: 85 86 20 BE A1 E6 E0 A9 00 85 E6 E6 86 E6 86 E6 ÖÜ.æ°Ê‡©.ÖÊÊÜÊÜÊ
  B0: E6 20 BE A1 A0 02 B1 85 85 E0 C8 B1 85 85 E1 D0 Ê.æ°†.±ÖÖ‡»±ÖÖ·–
  C0: EA A5 E0 D0 E6 AD 6C A0 85 E2 AD 6D A0 85 E3 18 Í•‡–Ê≠l†Ö‚≠m†Ö„.
  D0: A5 E3 69 02 85 E5 38 A5 E2 ED 23 A4 85 E4 A5 E5 •„i.ÖÂ8•‚Ì#§Ö‰•Â
  E0: E9 00 85 E5 A0 00 B1 E2 29 0F CD 11 A0 D0 21 A8 È.ÖÂ†.±‚).Õ.†–!®
  F0: B1 E2 D9 11 A0 D0 19 88 D0 F6 A0 00 B1 E2 29 F0 ±‚Ÿ.†–.à–ˆ†.±‚)
 100: 53 4F 53 20 4B 52 4E 4C 62 00 01 00 0E 2E 44 31 SOS.KRNLb.....D1
 110: 2F 53 4F 53 2E 49 4E 54 45 52 50 AA A5 A0 F9 A0 /SOS.INTERP™•†˘†
 120: A0 A5 A0 A0 A5 A0 A0 C5 A0 A0 98 A0 F0 A1 A0 CC †•††•††≈††ò†°†Ã
 130: A0 A0 C5 A0 A0 A0 A0 A0 EE A0 A0 C4 0E 2E 44 31 ††≈†††††Ó††ƒ..D1
 140: 2F 53 4F 53 2E 44 52 49 56 45 52 FF 9A A0 FF 9A /SOS.DRIVERˇö†ˇö
 150: A0 A0 A0 A0 D0 A0 A0 C1 A0 A0 8A A0 A0 F9 A0 C1 ††††–††¡††ä††˘†¡
 160: E9 A0 9E A1 A0 F5 A0 A0 A5 A0 A0 88 00 00 88 0C È†û°†ı††•††à..à.
 170: A9 00 AA 9D 00 1A 9D 00 16 9D 00 1B 9D 00 18 9D ©.™ù..ù..ù..ù..ù
 180: 00 14 9D 00 01 CA D0 EB A9 30 8D DF FF A2 FB 9A ..ù.. –Î©0çflˇ¢˚ö
 190: A9 1A 8D D0 FF 20 D4 1F AD DF FF 29 10 09 28 8D ©.ç–ˇ.‘.≠flˇ)..(ç
 1A0: DF FF A2 FF 9A A9 1A 8D D0 FF AD 01 19 8D EF FF flˇ¢ˇö©.ç–ˇ≠..çÔˇ
 1B0: 6C 02 00 AA AD EF FF 48 8E EF FF A5 27 05 26 F0 l..™≠ÔˇHéÔˇ•'.&
 1C0: 33 A5 26 D0 02 C6 27 C6 26 18 A5 23 65 27 85 23 3•&–.∆'∆&.•#e'Ö#
 1D0: A5 25 65 27 85 25 E6 27 A4 26 F0 07 B1 22 91 24 •%e'Ö%Ê'§&.±"ë$
 1E0: 88 D0 F9 B1 22 91 24 88 C6 23 C6 25 C6 27 D0 EC à–˘±"ë$à∆#∆%∆'–Ï
 1F0: E6 23 E6 25 68 8D EF FF 60 18 A5 24 65 10 85 10 Ê#Ê%hçÔˇ`.•$e.Ö.
```

###

seems like an
early version