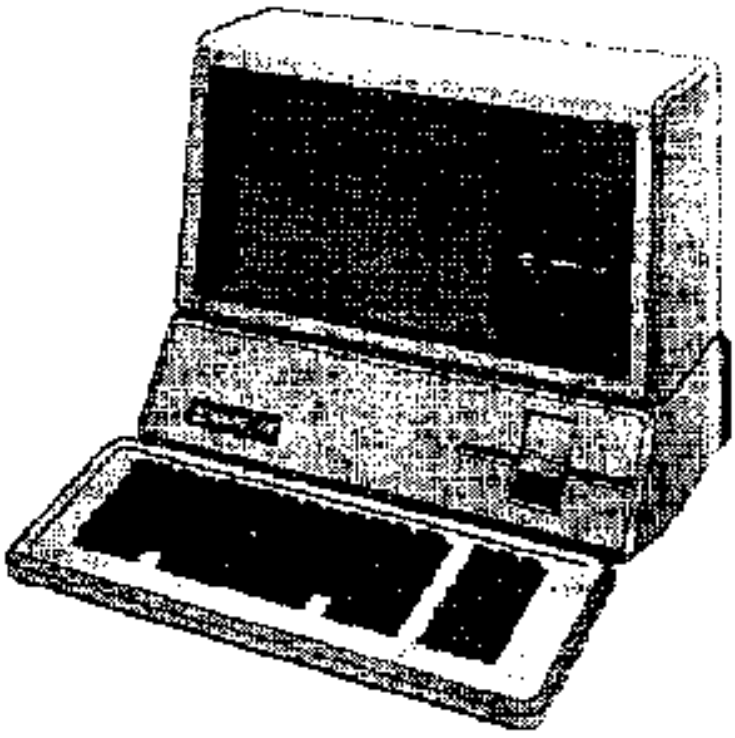
# Apple ///
# Disk /// Floppy Disk Formatter Driver 1.30 Source Code Listing



Created by David T. Craig
06 January 1998 • 71533.606@compuserve.com

```
; ##############################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTDX.TEXT
; ##############################################################################################################

000001                  .NOPATCHLIST
000002                  .NOMACROLIST
000003  ;---------------------------------------------------------------------
000004
000005  ;               SOS DISK /// Formatter Device Driver
000006
000007  ;               (C) Copyright 1980, 1981 by Apple Computer Inc.
000008
000009
000010  ;       Revisions: 11-SEP-81   V1.10   Modify Devtype and Subtype to
000011  ;                  reflect new conventions.  Activate all formatters.
000012
000013  ;                     19-AUG-82   V1.21   Include DSPEED checks and max  /RRA82237/
000014  ;                  retries on WTRACK16.  R. Auricchio. Changes marked by /RRA82237/
000015  ;                  '/RRA82237'.                                          /RRA82237/
000016
000017  ;                     15-SEP-82   V1.22   Fix WTRACK16/VTRACK code so    /RRA82258/
000018  ;                  it doesn't write 128 nybbles prior to sector 0 on a   /RRA82258/
000019  ;                  retry.  This caused 'too fast' error if several tries /RRA82258/
000020  ;                  were done: since 128-nibble seam is so large, VTRACK  /RRA82258/
000021  ;                  thinks NSYNC is too large, and decrements it.  The end/RRA82258/
000022  ;                  result is NSYNC getting too small. Also changed VTRACK/RRA82258/
000023  ;                  retries to 2 rotations MAX.  R. Auricchio.            /RRA82258/
000024  ;                  Changes marked by '/RRA82258'.                        /RRA82258/
000025
000026  ;                     06-DEC-82   V1.23   Fix VTRACK code to verify that /RRA82340/
000027  ;                  the gap between sectors 15 & 0 (the "seam") is large  /RRA82340/
000028  ;                  enough. R. Auricchio. Changes marked by '/RRA82340'.  /RRA82340/
000029
000030  ;                     04-Jan-82  V1.30    Installed comment field and set/HGL83004/
000031  ;                  the configuration block length to 0.  H. Lehtman.     /HGL83004/
000032  ;                  Deleted absolute assembly code.                       /HGL83004/
000033  ;                  Added code to support D_CONTROL 0 by returning rather /HGL83004/
000034  ;                  than generating an error.                             /HGL83004/
000035  ;                  Changes marked by /HGL83004.                          /HGL83004/
000036
000037  ;       Error codes returned:   $00 : good completion
000038  ;                               $27 : Unable to format (usually bad media)
000039  ;                               $28 : Write-Protected
000040  ;                               $33 : Drive too SLOW                  /RRA82237/
000041  ;                               $34 : Drive too FAST                  /RRA82237/
000042  ;---------------------------------------------------------------------
000043
000044  DEVTYPE         .EQU        011                     ;Disk Formatter
000045  SUBTYPE         .EQU        01                      ;For Disk ///
000046  APPLE           .EQU        0001
000047  RELEASE         .EQU        1300                    ;                     /HGL83301/
000048  DEVBLKS         .EQU        280.                    ;NUMBER OF BLKS ON VOLUME
000049  ;
000050  MAXTRKRETRY     .EQU        3.                      ;MAX RETRIES PER TRACK        /RRA82237/
000051  MAXSECTRETRY    .EQU        16.                     ;MAX RETRIES TO READ SECTORS  /RRA82258/
000052  ;
000053  ;     AVERAGE CORRECT SYNC VALUE HAS BEEN MEASURED AS FOLLOWS         /RRA82237/
000054  ;
000055  GOODSYNC        .EQU        22.                     ;CORRECT VALUE                /RRA82237/
000056  DELTAS          .EQU        3.                      ;                             /RRA82237/
000057  DELTAF          .EQU        3.                      ;DSPEED IS DELTA*16/2         /RRA82237/
000058  MINSYNC         .EQU        GOODSYNC-DELTAF         ;                             /RRA82237/
000059  MAXSYNC         .EQU        GOODSYNC+DELTAS         ;                             /RRA82237/
000060
000061                  .PAGE
000062  ;---------------------------------------------------------------------
000063  ;
000064  ;  The macro SWITCH performs an N way branch based on a switch index.  The
000065  ;  maximum value of the switch index is 127 with bounds checking provided
000066  ;  as an option.  The macro uses the A and Y registers and alters the C,
000067  ;  Z, and N flags of the status register, but the X register is unchanged.
000068  ;
000069  ;               SWITCH  [index], [bounds], adrs_table, [*]
000070  ;
000071  ;       index   This is the variable that is to be used as the switch index.
000072  ;               If omitted, the value in the accumulator is used.
000073  ;
000074  ;      bounds   This is the maximum allowable value for index.  If index
000075  ;               exceeds this value, the carry bit will be set and execution
000076  ;               will continue following the macro.  If bounds is omitted,
```

```
000077  ;               no bounds checking will be performed.
000078  ;
000079  ;  adrs_table   This is a table of addresses (low byte first) used by the
000080  ;               switch.  The first entry corresponds to index zero.
000081  ;
000082  ;           *   If an asterisk is supplied as the fourth parameter, the
000083  ;               macro will push the switch address but will not exit to
000084  ;               it; execution will continue following the macro.  The
000085  ;               program may then load registers or set the status before
000086  ;               exiting to the switch address.
000087  ;
000088  ;---------------------------------------------------------------------
000089  ;
000090                  .MACRO      SWITCH
000091                  .IF         "%1" <> ""              ;If PARM1 is present,
000092                  LDA         %1                      ;  Load A with switch index
000093                  .ENDC
000094                  .IF         "%2" <> ""              ;If PARM2 is present,
000095                  CMP         #%2+1                   ;  Perform bounds checking
000096                  BCS         $010                    ;  on switch index
000097                  .ENDC
000098                  ASL         A
000099                  TAY
000100                  LDA         %3+1,Y                  ;Get switch address from table
000101                  PHA                                 ;  and push onto stack
000102                  LDA         %3,Y
000103                  PHA
000104                  .IF         "%4" <> "*"             ;If PARM4 is omitted,
000105                  RTS                                 ;  Exit to code
000106                  .ENDC                               ;Otherwise, drop through
000107  $010            .ENDM
000108                  .PAGE
000109
000110  ;  SOS Global Data & Subroutines
000111
000112  SYSERR          .EQU        1928
000113
000114  ;  SOS Error Codes
000115
000116  XCTLCODE        .EQU        21                      ;Invalid control/status code
000117  XCTLPARM        .EQU        22                      ;Invalid control/status parm
000118  XNOTOPEN        .EQU        23                      ;Device not open
000119  XNOTAVAIL       .EQU        24                      ;Device not available
000120  XBADOP          .EQU        26                      ;Invalid operation for device
000121  XIOERR          .EQU        27                      ;I/O err, cannot format
000122  XWPROT          .EQU        2B                      ;Write Protected
000123  XDEVSPEC        .EQU        30                      ;Device-specific error        /RRA82237/
000124
000125  ;  Miscellaneous Equates
000126
000127  TRUE            .EQU        80
000128  FALSE           .EQU        00
000129
000130  ;---------------------------------------------------------------------
000131
000132  ;   SOS Device Driver Interface
000133
000134  ;---------------------------------------------------------------------
000135
000136  SOSINT          .EQU        0C0
000137  REQCODE         .EQU        SOSINT+0                ;SOS request code
000138  UNITNUM         .EQU        SOSINT+1                ;REQUESTED UNIT NUMBER
000139  CTLSTAT         .EQU        SOSINT+2                ;Control/status code
000140  CSLIST          .EQU        SOSINT+3                ;Control/status list pointer
000141
000142                  .PROC       FMTDISK3
000143  ;---------------------------------------------------------------------
000144
000145  ;  Device Identification Blocks: The first one is here so that the SOS
000146  ;    loader correctly pagealigns us. The other three (for .D2..4) are
000147  ;    at the end of the module, so they don't affect the alignment.
000148
000149  ;---------------------------------------------------------------------
000150                  .WORD       0FFFF                   ;                        /HGL83004/
000151                  .WORD       53.                     ;LENGTH OF COMMENT       /HGL83004/
000152                  .ASCII      "(C) Apple Computer, 1983.  "       ;  /HGL83004/"
000153                  .ASCII      "Disk /// Formatter Driver."        ;  /HGL83004/"
000154
000155
000156  DIBD1           .WORD       DIBD2                   ;Link to next device handler
000157                  .WORD       FMTMAIN                 ;Entry point address
```

```
000158                  .BYTE       6                       ;Length of device name
000159                  .ASCII      ".FMTD1         "
000160                  .BYTE       0C0                     ;Active, page-aligned
000161                  .BYTE       00,00                   ;Slot & Unit numbers
000162                  .BYTE       DEVTYPE
000163                  .BYTE       SUBTYPE
000164                  .BYTE       00
000165                  .WORD       DEVBLKS                 ;block count
000166                  .WORD       APPLE
000167                  .WORD       RELEASE
000168
000169                  .WORD       0                       ;                           /HGL83004/
000170  DCB_MINSYNC     .BYTE       MINSYNC                 ;                           /RRA82237/
000171  DCB_MAXSYNC     .BYTE       MAXSYNC                 ;                           /RRA82237/
000172  DCB_TRKRETRY    .BYTE       MAXTRKRETRY             ;                           /RRA82237/
000173  DCB_SECTRETRY   .BYTE       MAXSECTRETRY            ;                           /RRA82258/
000174                  .TITLE      "DRIVER GLOBAL EQUATES/DATA"
000175                  .INCLUDE    FMTDATA.TEXT            ;DATA
000176                  .INCLUDE    FMTREAD.TEXT            ;READ DATA
000177                  .INCLUDE    FMTWRADDR.TEXT          ;WRITE ADDRESS
000178                  .INCLUDE    FMTFORMAT.TEXT          ;FORMATTER CODE
000179                  .INCLUDE    FMTWTRACK.TEXT          ;WTRACK16/VTRACK            /RRA82258/
000180                  .TITLE      "FORMATTER MAINLINE"
000181                  .PAGE
000182  ;---------------------------------------------------------------------
000183
000184  ;  FMTDISK3-- Main entry point
000185
000186  ;---------------------------------------------------------------------
000187
000188  FMTMAIN         .EQU        *
000189
000190                  SWITCH      REQCODE,                8, REQSTSW
000191
000192  BADREQ          LDA         #XBADOP                 ;Invalid request code
000193                  JSR         SYSERR
000194
000195
000196  REQSTSW         .EQU        *                       ;FMTDISK3 request switch
000197                  .WORD       BADREQ-1                ;0 - READ
000198                  .WORD       BADREQ-1                ;1 - WRITE
000199                  .WORD       FMTSTATUS-1             ;2 - DSTATUS
000200                  .WORD       FMTCTRL-1               ;3 - DCNTRL
000201                  .WORD       BADREQ-1                ;4 - INVALID
000202                  .WORD       BADREQ-1                ;5 - INVALID
000203                  .WORD       BADREQ-1                ;6 - OPEN
000204                  .WORD       BADREQ-1                ;7 - CLOSE
000205                  .WORD       FMTINIT-1               ;8 - INIT
000206                  .PAGE
000207  ;---------------------------------------------------------------------
000208
000209  ;  FMTDISK3 -- Control Request
000210
000211  ;---------------------------------------------------------------------
000212
000213  FMTCTRL         .EQU        *
000214                  LDA         CTLSTAT                 ;get control code
000215                  CMP         #0                      ;reset must be supported /HGL82004/
000216                  BNE         $005                    ;    for Pascal.         /HGL82004/
000217                  RTS                                 ;nop if reset call       /HGL82004/
000218  $005            CMP         #0FE                    ;is it our special one?  /HGL82004/
000219                  BEQ         $020                    ;=>yes
000220                  LDA         #XCTLCODE               ;bad control code
000221  $010            JSR         SYSERR
000222
000223  ; Make sure the FORMATTER code likes everything:
000224
000225  $020            JSR         FMTSTARTUP              ;OK TO GO?
000226                  CMP         #0                      ;WELL?
000227                  BNE         $010                    ;=>NOPE
000228
000229  ;  SET UP CONTROL-LIST PARMS:
000230
000231                  LDA         CSLIST                  ;copy the address of
000232                  STA         USERBUF                 ; the caller's buffer
000233                  LDA         CSLIST+1
000234                  STA         USERBUF+1
000235                  LDA         1400+CSLIST+1           ;lest we forget the
000236                  STA         USERBUF+2               ; extend value...
000237
000238                  LDX         UNITNUM                 ;GET DRIVE TO BE FORMATTED
```

```
000239                  INX                                     ;MAKE RELATIVE TO ONE
000240
000241  ; FORMAT THE DISKETTE:
000242
000243                  JSR         DSKFORM                     ;** FORMAT IT **
000244
000245  ; EXIT PATH:
000246
000247                  CMP         #0                          ;CHECK RETURN CODE
000248                  BNE         $030                        ;=>AN ERROR
000249                  RTS                                     ;ALL'S WELL...
000250
000251  $030            CMP         #2                          ;IS IT WPROT?
000252                  BNE         $040                        ;=>NO, MUST BE IOERR
000253                  LDA         #XWPROT                     ;YES
000254                  JSR         SYSERR
000255
000256  $040            CMP         #1                          ;IS IT I/O ERR?       /RRA82237/
000257                  BNE         $050                        ;->NOPE               /RRA82237/
000258                  LDA         #XIOERR                     ;I/O ERROR
000259                  JSR         SYSERR
000260
000261  $050            CLC                                     ;SLOW/FAST IS         /RRA82237/
000262                  ADC         #XDEVSPEC                   ; SPECIFIC            /RRA82237/
000263                  JSR         SYSERR                      ;                     /RRA82237/
000264
000265  BADDRIVE        .EQU        *                           ;INVALID DRIVE REQUESTED
000266                  LDA         #XCTLPARM
000267                  JSR         SYSERR
000268                  .PAGE
000269  ;---------------------------------------------------------------------
000270
000271  ;  FMTDISK3 -- Status Request
000272
000273  ;---------------------------------------------------------------------
000274
000275  FMTSTATUS       .EQU        *
000276                  LDA         CTLSTAT
000277                  CMP         #0FE                        ;IS IT OUR SPECIAL ONE?
000278                  BEQ         $010                        ;=>YES
000279                  LDA         #XCTLCODE                   ;ILLEGAL CODE, BOZO!
000280                  JSR         SYSERR
000281
000282  $010            LDY         #0
000283                  LDA         #0FF                        ;WE HAVE NO BITMAP PREFERENCE
000284                  STA         (CSLIST),Y                  ; FOR BITMAP-STARTBLOCK FIELD
000285                  INY
000286                  STA         (CSLIST),Y
000287                  RTS
000288                  .PAGE
000289  ;---------------------------------------------------------------------
000290
000291  ;  FMTINIT -- Initialization Request
000292
000293  ;---------------------------------------------------------------------
000294
000295  FMTINIT         .EQU        *
000296                  CLC
000297                  RTS
000298                  .TITLE      "The other DIBs"
000299                  .PAGE
000300  DIBD2           .WORD       DIBD3                       ;Link to next device handler
000301                  .WORD       FMTMAIN                     ;Entry point address
000302                  .BYTE       6                           ;Length of device name
000303                  .ASCII      ".FMTD2         "
000304                  .BYTE       080                         ;Active
000305                  .BYTE       00,01                       ;Slot & Unit numbers
000306                  .BYTE       DEVTYPE
000307                  .BYTE       SUBTYPE
000308                  .BYTE       00
000309                  .WORD       DEVBLKS                     ;block count
000310                  .WORD       APPLE
000311                  .WORD       RELEASE
000312                  .WORD       0                           ;NO DCB
000313
000314  DIBD3           .WORD       DIBD4                       ;Link to next device handler
000315                  .WORD       FMTMAIN                     ;Entry point address
000316                  .BYTE       6                           ;Length of device name
000317                  .ASCII      ".FMTD3         "
000318                  .BYTE       080                         ;Active
000319                  .BYTE       00,02                       ;Slot & Unit numbers
```

```
000320                  .BYTE       DEVTYPE
000321                  .BYTE       SUBTYPE
000322                  .BYTE       00
000323                  .WORD       DEVBLKS                 ;block count
000324                  .WORD       APPLE
000325                  .WORD       RELEASE
000326                  .WORD       0                       ;NO DCB
000327
000328  DIBD4           .WORD       0000                    ;Link to next device handler
000329                  .WORD       FMTMAIN                 ;Entry point address
000330                  .BYTE       6                       ;Length of device name
000331                  .ASCII      ".FMTD4         "
000332                  .BYTE       080                     ;Active
000333                  .BYTE       00,03                   ;Slot & Unit numbers
000334                  .BYTE       DEVTYPE
000335                  .BYTE       SUBTYPE
000336                  .BYTE       00
000337                  .WORD       DEVBLKS                 ;block count
000338                  .WORD       APPLE
000339                  .WORD       RELEASE
000340                  .WORD       0                       ;NO DCB
000341                  .END

; ##########################################################################################################
; #   END OF FILE:  FMTDX.TEXT
; #   LINES      :  341
; #   CHARACTERS :  19337
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ##########################################################################################################
```

```
; ##########################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTDATA.TEXT
; ##########################################################################################################

000001                  .PAGE
000002  ; GLOBAL DATA AREAS:
000003
000004  USERBUF         .BLOCK      3                       ;SECTOR BUF POINTER
000005                  .TITLE      "FORMATTER LOCAL EQUATES/DATA"
000006                  .PAGE
000007  ; FORMATTER LOCAL DATA AREAS:
000008
000009  E_REG           .EQU        0FFDF
000010
000011  IBSLOT          .EQU        81                      ;SLOT
000012  IMASK           .EQU        IBSLOT+0A
000013  CURTRK          .EQU        IBSLOT+0B               ;CURRENT TRACK
000014  INTRTRY         .EQU        IBSLOT+0E
000015  RETRYCNT        .EQU        IBSLOT+12
000016  BUF             .EQU        IBSLOT+1A               ;DATA BUFFER FOR PRE/POSTNIB
000017  CKSUM           .EQU        IBSLOT+15               ;CHECKSUM BYTE
000018  CSSTV           .EQU        IBSLOT+16               ;FOUR BYTES,
000019  SECT            .EQU        CSSTV+1
000020  TRACK           .EQU        CSSTV+2
000021  ;                       CHECKSUM,SECTOR,TRACK,VOLUME
000022  TRKN            .EQU        IBSLOT+1D
000023  MONTIME         .EQU        CSSTV+2                 ;MOTOR-ON TIME
000024  ;
000025  ;    Zeropage Usage:
000026  ;
000027  AA              .EQU        0D0                     ;TIMING CONSTANT
000028  TRK             .EQU        0D1                     ;FORMATTER TRACK
000029  NSECT           .EQU        0D2                     ;FORMATTER SECTOR
000030  NVOL            .EQU        0D3                     ;FORMATTER VOLUME
000031  NSYNC           .EQU        0D4                     ;NUMBER SELFSYNC NYBBLES
000032  IDX             .EQU        0D5                     ;NBUF INDEX/COUNT FOR READ
000033  TRKRETRY        .EQU        0D6                     ;NO. RETRIES FOR WTRACK16
000034                  .PAGE
000035  ;************************
000036  ;                       *
000037  ;  HARDWARE ADDRESSES   *
000038  ;                       *
000039  ;************************
000040  MOTOROFF        .EQU        0C088
000041  MOTORON         .EQU        0C089
000042  Q6L             .EQU        0C08C
000043  Q6H             .EQU        0C08D
000044  Q7L             .EQU        0C08E
000045  Q7H             .EQU        0C08F
000046
000047  ;************************
000048  ;                       *
000049  ; ROM ROUTINE ADDRESSES *
000050  ;                       *
000051  ;************************
000052  RDADR16         .EQU        0F1B9                   ;READ ADDRESS
000053  WRITE16         .EQU        0F216                   ;WRITE DATA
000054  PRENIB16        .EQU        0F2C4                   ;PRENIBBLIZE FOR WRITE
000055  SEEK            .EQU        0F400                   ;SEEK
000056  MSWAIT          .EQU        0F456                   ;TIME DELAYER
000057  DNIBL           .EQU        0F300                   ;DNIBL TABLE ADDR
000058                  .PAGE
000059  ;---------------------------------------------------------------
000060  ;       Zeropage Savearea: This area is used to save the SOS ZeroPage
000061  ;        area $80..$9F, which is used exclusively by the ROM Core
000062  ;        Routines. These locations are clobbered by the Formatter
000063  ;        during its operation, and are then restored.
000064  ;---------------------------------------------------------------
000065
000066  ZEROSAVE        .BLOCK      32.                     ;SAVEAREA FOR $1800 PAGE
000067  XTNDSAVE        .BLOCK      32.                     ;SAVEAREA FOR $1400 PAGE
000068
000069  E_SAVE          .BLOCK      1                       ;SAVED CALLER ENVIRONMENT
000070  P_SAVE          .BLOCK      1                       ;SAVED CALLER STATUS
000071
000072  ;       DiskDrive Select Addresses
000073  ;                           .D1 .D2 .D3 .D4
000074  ;                           --  --  --  --
000075  SEL1            .BYTE       000,0D0,0D5,0D5,0D5
000076  SEL2            .BYTE       000,0D2,0D1,0D0,0D1
```

```
000077  SEL3            .BYTE       000,0D4,0D2,0D3,0D3
000078
000079  FOUND           .BLOCK      16.                     ;'SECTOR FOUND' TABLE.

; ##########################################################################################################
; #   END OF FILE:  FMTDATA.TEXT
; #   LINES      :  79
; #   CHARACTERS :  4060
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ##########################################################################################################
```

```
; ##############################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTREAD.TEXT
; ##############################################################################################################

000001                  .TITLE       "16-SECTOR READ"
000002                  .PAGE
000003  ;************************
000004  ;                       *
000005  ;     READ SUBROUTINE   *
000006  ;   (16-SECTOR FORMAT)   *
000007  ;                       *
000008  ;************************
000009  ;    READS 6-BIT NIBLS   *
000010  ;     (00ABCDEF) INTO    *
000011  ;     NBUF1 AND NBUF2    *
000012  ;    CONVERTING 7-BIT    *
000013  ;     NIBLS TO 6-BIT     *
000014  ;    VIA 'DNIBL' TABLE   *
000015  ;                       *
000016  ;  FIRST READS NBUF2     *
000017  ;          HIGH TO LOW,  *
000018  ;  THEN READS NBUF1      *
000019  ;          LOW TO HIGH.  *
000020  ;                       *
000021  ;   ---- ON ENTRY ----   *
000022  ;                       *
000023  ;  X-REG: SLOTNUM       *
000024  ;         TIMES $10.    *
000025  ;                       *
000026  ;  READ MODE (Q6L, Q7L)  *
000027  ;                       *
000028  ;   ---- ON EXIT -----   *
000029  ;                       *
000030  ;  CARRY SET IF ERROR.   *
000031  ;                       *
000032  ;  IF NO ERROR:         *
000033  ;     A-REG HOLDS $AA.  *
000034  ;     X-REG UNCHANGED.  *
000035  ;     Y-REG HOLDS $00.  *
000036  ;     CARRY CLEAR.      *
000037  ;                       *
000038  ;     NBUF1 AND NBUF2   *
000039  ;       HOLD 6-BIT NIBLS*
000040  ;       (00ABCDEF)      *
000041  ;                       *
000042  ;     USES TEMP 'IDX'.  *
000043  ;                       *
000044  ;   ---- CAUTION -----   *
000045  ;                       *
000046  ;        OBSERVE        *
000047  ;    'NO PAGE CROSS'    *
000048  ;      WARNINGS ON      *
000049  ;    SOME BRANCHES!!    *
000050  ;                       *
000051  ;   ---- ASSUMES ----   *
000052  ;                       *
000053  ;   1 USEC CYCLE TIME   *
000054  ;                       *
000055  ;************************
000056  READ16          LDY          #020                    ;'MUST FIND' COUNT.
000057  RSYNC           DEY                                  ;IF CAN'T FIND MARKS
000058                  BEQ          RDERR                   ;THEN EXIT WITH CARRY SET.
000059  RD1             LDA          Q6L,X                   ;READ NIBL.
000060                  BPL          RD1                     ;**; NO PAGE CROSS! ***
000061  RSYNC1          EOR          #0D5                    ;DATA MARK 1?
000062                  BNE          RSYNC                   ;LOOP IF NOT.
000063                  NOP                                  ;DELAY BETWEEN NIBLS.
000064  RD2             LDA          Q6L,X
000065                  BPL          RD2                     ;**; NO PAGE CROSS! ***
000066                  CMP          #0AA                    ;DATA MARK 2?
000067                  BNE          RSYNC1                  ;(IF NOT, IS IT DM1?)
000068                  LDY          #056                    ;INIT NBUF2 INDEX.
000069  ;              (ADDED NIBL DELAY)
000070  RD3             LDA          Q6L,X
000071                  BPL          RD3                     ;**; NO PAGE CROSS! ***
000072                  CMP          #0AD                    ;DATA MARK 3?
000073                  BNE          RSYNC1                  ;(IF NOT, IS IT DM1?)
000074  ;         (CARRY SET IF DM3!)
000075                  LDA          #0                      ;INIT CHECKSUM.
000076  RDATA1          DEY
```

```
000077                  STY         IDX
000078  RD4             LDY         Q6L,X
000079                  BPL         RD4                     ;**; NO PAGE CROSS! ***
000080  RDNIBL1         EOR         DNIBL,Y                 ;XOR 6-BIT NIBL.
000081                  LDY         IDX
000082
000083  ; NOTE: The storing of the Nybble into the NBUF has been deleted so that
000084  ;       the formatter doesn't have to prenibblize the data pattern again.
000085
000086  ;               STA     NBUF2,Y                 ;STORE IN NBUF2 PAGE.
000087
000088                  BNE         RDATA1                  ;TAKEN IF Y-REG NONZERO.
000089  RDATA2          STY         IDX
000090  RD5             LDY         Q6L,X
000091                  BPL         RD5                     ;**; NO PAGE CROSS! ***
000092  RDNIBL2         EOR         DNIBL,Y                 ;XOR 6-BIT NIBL.
000093                  LDY         IDX
000094
000095  ; NOTE: The storing of the Nybble into the NBUF has been deleted so that
000096  ;       the formatter doesn't have to prenibblize the data pattern again.
000097
000098  ;               STA     NBUF1,Y                 ;STORE IN NBUF1 PAGE.
000099
000100                  INY
000101                  BNE         RDATA2
000102  RD6             LDY         Q6L,X                   ;READ 7-BIT CSUM NIBL.
000103                  BPL         RD6                     ;**; NO PAGE CROSS! ***
000104  RDNIBL3         CMP         DNIBL,Y                 ;IF LAST NBUF1 NIBL NOT
000105                  BNE         RDERR                   ;EQUAL CHKSUM NIBL THEN ERR.
000106  RD7             LDA         Q6L,X
000107                  BPL         RD7                     ;**; NO PAGE CROSS! ***
000108                  CMP         #0DE                    ;FIRST BIT SLIP MARK?
000109                  BNE         RDERR                   ;(ERR IF NOT)
000110                  NOP                                 ;DELAY BETWEEN NIBLS.
000111  RD8             LDA         Q6L,X
000112                  BPL         RD8                     ;**; NO PAGE CROSS! ***
000113                  CMP         #0AA                    ;SECOND BIT SLIP MARK?
000114                  BEQ         RDEXIT                  ;(DONE IF IT IS)
000115  RDERR           SEC                                 ;INDICATE 'ERROR EXIT'.
000116                  RTS                                 ;RETURN FROM READ16 OR RDADR16.
000117  RDEXIT          CLC
000118                  RTS

; ##############################################################################################################
; #   END OF FILE:  FMTREAD.TEXT
; #   LINES      :  118
; #   CHARACTERS :  5863
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ##############################################################################################################
```

```
; ##########################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTWRADDR.TEXT
; ##########################################################################################################

000001                  .TITLE       "16-SECTOR WRITE ADDRESS"
000002                  .PAGE
000003  ;*******************************
000004  ;                              *
000005  ;  WRITE ADR FIELD SUBROUTINE  *
000006  ;      (16-SECTOR FORMAT)      *
000007  ;  WRITES SPECIFIED NUMBER OF  *
000008  ;  40-USEC (10-BIT) SELF-SYNC  *
000009  ;  NIBLS, ADR FIELDS 16-SECTOR *
000010  ;  START MARKS ($D5,$AA,$96),  *
000011  ;  BODY (VOLUME, TRACK, SECTOR,*
000012  ;  CHECKSUM), END FIELD MARKS, *
000013  ;  AND THE WRITE TURN-OFF NIBL.*
000014  ;                              *
000015  ;*******************************
000016  ;                              *
000017  ;   ------- ON ENTRY -------   *
000018  ;                              *
000019  ;  THE LOCATIONS VOLUME, TRK,  *
000020  ;  AND NSECT MUST CONTAIN THE  *
000021  ;  DESIRED VOLUME, TRACK, AND  *
000022  ;  SECTOR VALUES DESIRED.      *
000023  ;                              *
000024  ;  THE PROPER DRIVE MUST BE    *
000025  ;  ENABLED AND UP TO SPEED IN  *
000026  ;  READ MODE (Q7L, Q6L).       *
000027  ;                              *
000028  ;  X-REG CONTAINS SLOTNUM      *
000029  ;                    TIMES 16. *
000030  ;                              *
000031  ;  Y-REG CONTAINS NUMBER OF    *
000032  ;    SELF-SYNC NIBLS DESIRED   *
000033  ;    MINUS 1.                  *
000034  ;      (0 FOR 256 NIBLS)       *
000035  ;                              *
000036  ;*******************************
000037  ;                              *
000038  ;   ------- REQUIRES -------   *
000039  ;                              *
000040  ;         1 USEC CYCLE         *
000041  ;                              *
000042  ;*******************************
000043  ;                              *
000044  ;   ------- CAUTION --------   *
000045  ;                              *
000046  ;  MOST OF THIS CODE IS TIME   *
000047  ;  CRITICAL.  OBSERVE ALL      *
000048  ;  'NO PAGE CROSS!' WARNINGS   *
000049  ;  ON BRANCHES.                *
000050  ;                              *
000051  ;*******************************
000052  WADR16          SEC                                 ;ANTICIPATE WR PROT ERR.
000053                  LDA          Q6H,X                  ;INTO 'WR PROT SENSE' MODE.
000054                  LDA          Q7L,X                  ;SENSE IT (NEG=PROTECTED)
000055                  BMI          WADRTS                 ;ERR EXIT IF PROTECTED.
000056                  LDA          #0FF                   ;SELF-SYNC NIBL.
000057                  STA          Q7H,X                  ;WRITE FIRST NIBL.
000058                  CMP          Q6L,X                  ;(4) BACK TO WRITE MODE.
000059                  PHA                                 ;(3) FOR DELAY.
000060                  PLA                                 ;(4)
000061  WSYNC1          JSR          WADRTS1                ;(12) FOR 40-USEC NIBLS.
000062                  JSR          WADRTS1                ;(12)
000063                  STA          Q6H,X                  ;(5) WRITE NIBL.
000064                  CMP          Q6L,X                  ;(4) (BACK TO WRITE MODE)
000065                  NOP                                 ;(2) FOR DELAY.
000066                  DEY                                 ;(2) NEXT OF 'N' NIBLS.
000067                  BNE          WSYNC1                 ;(3) **; NO PAGE CROSS! ***
000068                  LDA          #0D5                   ;(2) ADR MARK 1.
000069                  JSR          WNIBLB2                ;(15,9,6) WRITE IT.
000070                  LDA          #0AA                   ;(2) ADR MARK 2.
000071                  JSR          WNIBLB2                ;(15,9,6) WRITE IT.
000072                  LDA          #096                   ;(2) 16-SECTOR ADR MARK 3.
000073                  JSR          WNIBLB2                ;(15,9,6) WRITE IT.
000074                  LDA          NVOL                   ;(3)
000075
              JSR          WBYTE                  ;(14,9,6) WRITE NVOL (ODD, THEN EVEN, BITS.)
```

```
000076                  LDA         TRK                     ;(3) WRITE TRACK NUMBER.
000077                  JSR         WBYTE                   ;(14,9,6) ODD, THEN EVEN, BITS)
000078                  LDA         NSECT                   ;(3) WRITE SECTOR NUMBER.
000079                  JSR         WBYTE                   ;(14,9,6) (ODD, THEN EVEN, BITS)
000080                  LDA         NVOL                    ;(3)
000081                  EOR         TRK                     ;(3) FORM ADR FIELD CHECKSUM.
000082                  EOR         NSECT                   ;(3)
000083                  PHA                                 ;(3) SAVE FOR EVEN BITS.
000084                  LSR         A                       ;(2) ALIGHN ODD BITS.
000085                  ORA         AA                      ;(3) SET CLOCK BITS.
000086  ;    (PRECISE TIMING, 32 CYCLES PER NIBL)
000087                  STA         Q6H,X                   ;(5) WRITE CHECKSUM ODD BITS.
000088                  LDA         Q6L,X                   ;(4) BACK TO WRITE MODE.
000089                  PLA                                 ;(4) RECOVER FOR EVEN BITS.
000090                  ORA         #0AA                    ;(2) SET CLOCK BITS.
000091                  JSR         WNIBLA                  ;(17,9,6) WRITE THEM.
000092                  LDA         #0DE                    ;(2) END MARK 1.
000093                  JSR         WNIBLB2                 ;(15,9,6) WRITE IT.
000094                  LDA         #0AA                    ;(2) END MARK 2.
000095                  JSR         WNIBLB2                 ;(15,9,6) WRITE IT.
000096                  LDA         #0EB                    ;(2) END MARK 3.
000097                  JSR         WNIBLB2                 ;(15,9,6) 'WRITE TURN-OFF'
000098                  CLC                                 ;INDICATE NO WR PROT ERR.
000099  WADRTS          LDA         Q7L,X                   ;OUT OF WRITE MODE.
000100                  LDA         Q6L,X                   ;TO READ MODE.
000101  WADRTS1         RTS                                 ;RETURN
000102  WBYTE           PHA                                 ;(3) PRESERVE FOR EVEN BITS.
000103                  LSR         A                       ;(2) ALIGN ODD BITS.
000104                  ORA         AA                      ;(3) SET CLOCK BITS.
000105                  STA         Q6H,X                   ;(5) WRITE NIBL.
000106                  CMP         Q6L,X                   ;(4)
000107                  PLA                                 ;(4) RECOVER FOR EVEN BITS.
000108                  NOP                                 ;(2)
000109                  NOP                                 ;(2) FOR DELAY.
000110                  NOP                                 ;(2)
000111                  ORA         #0AA                    ;(2) SET CLOCK BITS.
000112  WNIBLA          NOP                                 ;(2) (17,9,6) ENTRY.
000113  WNIBLB2         NOP                                 ;(2) (15,9,6) ENTRY.
000114                  PHA                                 ;(3) FOR
000115                  PLA                                 ;(4)   DELAY.
000116  WRNIBL          STA         Q6H,X                   ;(5) WRITE NIBL.
000117                  CMP         Q6L,X                   ;(4)
000118                  RTS                                 ;(6) RETURN.

; ###########################################################################################################
; #   END OF FILE:  FMTWRADDR.TEXT
; #   LINES      :  118
; #   CHARACTERS :  7495
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ###########################################################################################################
```

```
; ##############################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTFORMAT.TEXT
; ##############################################################################################################

000001                  .TITLE      "16-SECTOR FORMATTER"
000002  ;
000003  ; THIS IS THE 'SEEK' ROUTINE
000004  ;  SEEKS TRACK 'N' IN SLOT #X/$10
000005  ; DOES NOT CARE WHETHER ON DRIVE 0 OR 1 OR N...
000006  ;
000007  MYSEEK          ASL         A
000008                  ASL         CURTRK                  ;SEEK IS IN HALFTRACKS
000009                  STA         TRKN                    ;SAVE DESTINATION TRACK(*2)
000010                  TXA                                 ;TXY ROUTINE
000011                  LSR         A
000012                  LSR         A
000013                  LSR         A
000014                  LSR         A
000015                  TAY
000016                  LDA         TRKN                    ;AND WHERE I'M GOING TO
000017                  JSR         SEEK                    ;GO THERE!
000018                  LSR         CURTRK                  ;DIVIDE BACK DOWN
000019                  RTS
000020                  .PAGE
000021  ;***********************************************
000022  ;                                              *
000023  ;    FORMATTER  INITIALIZATION  SUBROUTINE     *
000024  ;                                              *
000025  ;       This subroutine is called by the       *
000026  ;       driver during OPEN; it will perform    *
000027  ;       all necessary startup initialization   *
000028  ;       prior to opening the FORMATTER driver.  *
000029  ;                                              *
000030  ;       INPUT : none                           *
000031  ;       OUTPUT: NONE                           *                    /RRA82237/
000032  ;                                              *
000033  ;***********************************************
000034
000035  ;       for 1.1 release: nothing to do since Rev0 ROM not supported   /RRA82237/
000036
000037  FMTSTARTUP      .EQU        *
000038                  LDA         #0                      ;GOOD
000039                  RTS
000040                  .PAGE
000041  ;***********************************************
000042  ;                                              *
000043  ;                FORMATTER SUBROUTINE          *
000044  ;                                              *
000045  ;  INPUT:  X = DRIVE TO BE FORMATTED (1..4)    *
000046  ;              'USERBUF' POINTER INITIALIZED.  *
000047  ;                                              *
000048  ; OUTPUT: AC = RETURN CODE AS FOLLOWS:         *
000049  ;                0 : GOOD COMPLETION           *
000050  ;                1 : UNABLE TO FORMAT          *
000051  ;                2 : WRITE-PROTECTED           *
000052  ;                3 : DRIVE TOO SLOW            *                    /RRA82237/
000053  ;                4 : DRIVE TOO FAST            *                    /RRA82237/
000054  ;                                              *
000055  ;***********************************************
000056
000057  DSKFORM         .EQU        *
000058
000059  ;  SELECT THE CORRECT DRIVE:
000060
000061                  LDY         SEL1,X                  ;GET FIRST SELECT
000062                  LDA         0C000,Y                 ;SELECT #1
000063                  LDY         SEL2,X                  ;GET SECOND SELECT
000064                  LDA         0C000,Y                 ;SELECT #2
000065                  LDY         SEL3,X                  ;GET THIRD SELECT
000066                  LDA         0C000,Y                 ;SELECT #3
000067
000068                  CPX         #1                      ;INTERNAL (.D1) ?
000069                  BNE         $030                    ;=>NO, IT'S EXTERNAL
000070                  LDA         0C0EA                   ;SELECT INTERNAL
000071                  JMP         $032
000072  $030            LDA         0C0EB                   ;SELECT EXTERNAL
000073
000074  ; SET UP THE ENVIRONMENT:
000075
000076  $032            LDA         E_REG
```

```
000077                  STA         E_SAVE                  ;SAVE CALLER ENV
000078                  ORA         #83                     ;1 MHz, ROM enabled
000079  ;               AND     #0FF-10                 ;inhibit the RESET
000080                  STA         E_REG
000081                  PHP                                 ;SAVE CALLER INHIBITS
000082                  PLA
000083                  STA         P_SAVE
000084
000085  ; SAVE THE ROM'S ZEROPAGE AREA:
000086
000087                  LDX         #31.                    ;INDEX
000088  $035            LDA         1480,X                  ;SAVE EXTEND
000089                  STA         XTNDSAVE,X
000090                  LDA         80,X
000091                  STA         ZEROSAVE,X
000092                  DEX
000093                  BPL         $035
000094
000095  ; START THE MOTOR:
000096
000097                  LDX         #060                    ;ALL APPLE-III DRIVES ARE
000098                  STX         IBSLOT
000099                  LDA         MOTORON,X               ;IN SLOT 6
000100
000101  ; PRENIBBLIZE THE DATA BUFFER PRIOR TO WRITING (WHILE MOTORING-UP):
000102  ;  Note: PRENIB16 returns to us in 1 MHz mode.
000103
000104                  LDA         E_REG                   ;SET 2 MHz MODE FOR
000105                  AND         #7F                     ; FASTER
000106                  STA         E_REG                   ;  PRENIB OPERATION
000107                  LDA         USERBUF                 ;SET DATA ADDRESS
000108                  STA         BUF                     ; TO POINT TO
000109                  LDA         USERBUF+1               ;  THE CALLER'S
000110                  STA         BUF+1                   ;   BUFFER...
000111                  LDA         USERBUF+2               ;ALSO SET XTND BYTE
000112                  STA         1400+BUF+1
000113                  JSR         PRENIB16                ;PRENIB THE BUFFER!
000114
000115                  LDA         #0D7                    ;ALMOST 1 SECOND
000116                  STA         MONTIME+1
000117
000118  ; RECALIBRATE THE HEAD WHILE THE MOTOR COMES UP TO SPEED:
000119
000120                  LDA         #80.                    ;FAKING TRACK 80 FORCES RECAL
000121                  STA         CURTRK
000122                  LDA         #0                      ;WE WANNA GO TO ZEEEEERO!
000123                  JSR         MYSEEK                  ;HONNNNNNNNK!
000124
000125  WAITUP          LDA         MONTIME+1               ;MOTOR UP TO SPEED?
000126                  BEQ         $040                    ;=>YES
000127                  JSR         MSWAIT                  ;NO, WAIT FOR A BIT...
000128                  JMP         WAITUP
000129
000130  ; INITIALIZE THINGS:
000131
000132  $040            SEI                                 ;NO IRQ DURING FORMATTING
000133                  LDA         #0                      ;SET INTERRUPT RETRIES
000134                  STA         INTRTRY                 ; TO ZERO (NONE OCCURRED)
000135                  LDA         #80                     ;SET INHIBIT INDICATOR
000136                  STA         IMASK                   ; TO SAY 'NO ENABLE,PLEASE'
000137
000138
          LDA           #1                              ;RANDY SAYS ALL SARA DISKETTES WILL BE VOLUME 1
000139                  STA         NVOL                    ;FOR FORMATTER.
000140                  LDA         #0AA                    ;SET Z-PAG LOC TO $AA FOR
000141                  STA         AA                      ;TIME DEPENDENT REFERENCES.
000142                  LDA         DCB_MAXSYNC             ;                       /RRA82237/
000143                  CLC                                 ;USE LARGER VALUE       /RRA82237/
000144                  ADC         #2                      ; TO START OFF          /RRA82237/
000145                  STA         NSYNC                   ;BEGINNING SELF-SYNC NIBLS.
000146
000147                  LDA         #0                      ;START ON TRACK ZERO
000148                  STA         TRK
000149
000150  FORMTRK         LDA         TRK
000151                  LDX         #60
000152                  JSR         MYSEEK                  ;GOTO NEXT TRACK.
000153
000154  ; See if the diskette is Write-Protected:
000155
000156                  LDX         #60
```

```
000157                  LDA         Q6H,X                   ;SENSE WRITEPROT MODE
000158                  LDA         Q7L,X
000159                  TAY                                 ;SAVE RESULT
000160                  LDA         Q7L,X                   ;BACK INTO
000161                  LDA         Q6L,X                   ; READ MODE
000162                  TYA
000163                  BPL         $010                    ;IT'S OK.
000164                  LDA         #2
000165                  JMP         FORMDONE                ;FLAG WRITE PROTECT.
000166
000167  $010            JSR         WT16                    ;WRITE AND VERIFY TRACK.    /RRA82237/
000168                  BCC         $020                    ;=>NO ERROR
000169
000170  ; BEFORE SAYING IT'S BAD MEDIA, SEE IF THE DRIVE'S WAY TOO FAST:      /RRA82237/
000171
000172                  LDA         #1                      ;ASSUME BAD MEDIA     /RRA82237/
000173                  LDY         NSYNC                   ;GET IT               /RRA82237/
000174                  CPY         DCB_MINSYNC             ;GAPS TOO SMALL?      /RRA82237/
000175                  BCS         $015                    ;GAPS OK: BAD MEDIA   /RRA82237/
000176                  LDA         #4                      ;"DRIVE TOO FAST"     /RRA82237/
000177  $015            JMP         FORMDONE
000178
000179  ; THE MEDIA FORMATTED OK...BUT THE DRIVE MIGHT STILL BE TOO SLOW OR TOO FAST.
000180
000181  $020            LDY         NSYNC                   ;GET IT               /RRA82237/
000182                  CPY         DCB_MINSYNC             ;GAPS TOO SMALL?      /RRA82237/
000183                  BCS         $025                    ;GAPS LARGE ENOUGH    /RRA82237/
000184                  LDA         #4                      ;"DRIVE TOO FAST"     /RRA82237/
000185                  JMP         FORMDONE                ;                     /RRA82237/
000186  $025            CPY         DCB_MAXSYNC             ;GAPS TOO LARGE?      /RRA82237/
000187                  BCC         $030                    ;->OK                 /RRA82237/
000188                  LDA         #3                      ;"DRIVE TOO SLOW"     /RRA82237/
000189                  JMP         FORMDONE                ;                     /RRA82237/
000190  $030            LDA         DCB_SECTRETRY           ;SECTOR RETRIES       /RRA82258/
000191                  STA         RETRYCNT                ; TO FIND SECTOR 0.
000192  FINDS0          .EQU        *
000193                  DEC         RETRYCNT                ;DONE RETRIES?
000194                  BNE         $010                    ;=>NO, KEEP RETRYING.
000195                  LDA         #1                      ;'UNABLE TO FORMAT' ERROR
000196                  JMP         FORMDONE
000197
000198  $010            LDX         #60
000199                  JSR         RDADR16                 ;READ ADR FIELD.
000200                  BCS         FINDS0                  ;RETRY IF ERR.
000201                  LDA         SECT                    ;CHECK SECTOR THAT WAS READ.
000202                  BNE         FINDS0                  ;CONTINUE SEARCHING IF NOT SECT 0.
000203                  LDX         #60
000204                  JSR         READ16                  ;NOW READ DATA FIELD.
000205                  BCS         FINDS0                  ;CONTINUE SEARCH IF ERR.
000206  ;       (NOW POSITIONED PROPERLY FOR NEXT TRACK)
000207                  INC         TRK                     ;INCREMENT TRACK NUMBER.
000208                  LDA         TRK
000209                  CMP         #35.                    ;CONTINUE IF LESS THAN 35.
000210                  BCC         FORMTRK
000211
000212                  LDA         #0                      ;GOOD COMPLETION
000213
000214  FORMDONE        PHA                                 ;SAVE RETURN CODE
000215                  LDX         #60
000216                  LDA         MOTOROFF,X              ;TURN MOTOR OFF.
000217                  LDA         #0
000218                  JSR         MYSEEK                  ;RESTORE HEAD TO TRACK 00
000219
000220  ; RESTORE THE ROM'S ZEROPAGE AREA:
000221
000222                  LDX         #31.                    ;INDEX
000223  $035            LDA         XTNDSAVE,X              ;RESTORE EXTEND
000224                  STA         1480,X
000225                  LDA         ZEROSAVE,X
000226                  STA         80,X
000227                  DEX
000228                  BPL         $035
000229
000230  ; Restore the caller's Environment:
000231
000232                  LDA         P_SAVE                  ;RESTORE
000233                  PHA                                 ; INHIBITS
000234                  PLP
000235                  LDA         E_SAVE                  ;RESTORE
000236                  STA         E_REG                   ; ENVIRONMENT
000237
```

```
000238                  PLA                                     ;RESTORE RETURN CODE
000239                  RTS                                     ; AND RETURN.

; ##############################################################################################################
; #   END OF FILE:  FMTFORMAT.TEXT
; #   LINES      :  239
; #   CHARACTERS :  13436
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ##############################################################################################################
```

```
; ##############################################################################################################
; #   PROJECT  :  Apple /// SOS DISK /// Formatter Device Driver 1.30 (6502 Assembly Source Code)
; #   FILE NAME:  FMTWTRACK.TEXT
; ##############################################################################################################

000001  ;****************************
000002  ;                           *
000003  ;   WRITE TRACK SUBROUTINE  *
000004  ;                           *
000005  ;****************************
000006  ;
000007  ; WE HAVE A COUNTER, TRKRETRY, WHICH PREVENTS WTRACK16 FROM RETRYING  /RRA82237/
000008  ;  FOREVER. THE EXTENSIVE RETRIES CAUSED THE TRACK TO PRECESS ON      /RRA82237/
000009  ;  SUCCESSIVE WRITES, POSSIBLY ALLOWING A DAMAGED SPOT ON THE MEDIA TO/RRA82237/
000010  ;  BE 'SLID' INTO AN INTERSECTOR GAP. SUBSEQUENT DATA WRITES COULD    /RRA82237/
000011  ;  THEN ENCOUNTER THE DAMAGED AREA.                                   /RRA82237/
000012  ;
000013  WT16            .EQU        *                       ;/RRA82237/
000014                  LDA         DCB_TRKRETRY            ;MAX RETRIES TO DO THE TRACK  /RRA82237/
000015                  STA         TRKRETRY                ; TO PREVENT TOO MUCH TRYING  /RRA82237/
000016
000017  ;  On the first attempt, deliberately write too large a seam. This
000018  ;     ensures that ALL of the track is erased.
000019
000020  WTRETRY         .EQU        *                       ;                     /RRA82258/
000021                  LDY         #128.                   ;128 NIBS PRIOR SECTOR 0
000022                  LDA         #0                      ;start at sector 0    /RRA82258/
000023                  STA         NSECT                   ;                     /RRA82258/
000024                  JMP         WSECT0                  ; TO INSURE NO BLANK SPOT BETW 15 & 0
000025
000026  WSECT           LDY         NSYNC                   ;CURRENT NUM OF GAP SELF-SYNC NIBLS.
000027  WSECT0          .EQU        *
000028                  LDX         #60
000029                  JSR         WADR16                  ;WRITE GAP AND ADR FIELD.
000030                  BCC         $010                    ;=>GOOD, NOT WRITE-PROTECTED.
000031                  JMP         WEXIT2                  ;ERR IF WRITE PROTECTED.
000032  $010            LDX         #60
000033                  JSR         WRITE16                 ;WRITE SECTOR FROM NBUF1, NBUF2.
000034                  BCC         $020                    ;=>OK
000035                  JMP         WEXIT2                  ;ERR IF WRITE PROTECTED.
000036  $020            INC         NSECT                   ;NEXT OF 16 SECTORS.
000037                  LDA         NSECT
000038                  CMP         #16.
000039                  BCC         WSECT                   ;CONTINUE IF NOT DONE.
000040                  .PAGE
000041  ;****************************
000042  ;                           *
000043  ;     VERIFY ROUTINE        *
000044  ;                           *
000045  ;  VERIFIES THAT THE FIRST  *
000046  ;  SECTOR ENCOUNTERED IS    *
000047  ;  SECTOR 0, AND THAT ALL   *
000048  ;  16 SECTORS ARE READABLE  *
000049  ;  WITH MINIMAL RETRIES.    *
000050  ;  (2 REVOLUTIONS MAXIMUM)  *
000051  ;                           *
000052  ;  IF FIRST SECTOR IS NOT   *
000053  ;  SECTOR 0 THEN THE        *
000054  ;  CURRENT NUMBER OF SELF-  *
000055  ;  SYNC NIBLS IS DECR'D BY  *
000056  ;  1.   THEN SECTOR         *
000057  ;  15 IS LOCATED SO AS TO   *
000058  ;  POSITION THE NEW TRACK   *
000059  ;  REWRITE.                 *
000060  ;                           *
000061  ;  IF UNABLE TO READ ANY    *
000062  ;  SECTOR THEN THE ENTIRE   *
000063  ;  TRACK IS REWRITTEN.      *
000064  ;                           *
000065  ;****************************
000066                  .PAGE
000067                  LDY         #16.-1                  ;sect0 is special     /RRA82258/
000068                  STY         NSECT                   ;SET 16 BYTES OF
000069                  LDA         DCB_SECTRETRY           ;SECTOR FOUND TABLE
000070                  STA         RETRYCNT                ;TO SOMETHING NONNEGATIVE
000071  CLRFOUND        STA         FOUND,Y
000072                  DEY
000073                  BPL         CLRFOUND
000074
000075  ; Delay for 40uSec per NSYNC byte, to ensure that the gap between     /RRA82340/
000076  ;  sector 15 & sector 0 is large enough.                              /RRA82340/
```

```
000077
000078                  LDA         NSYNC                   ;get current count    /RRA82340/
000079                  SEC                                 ;two already past and /RRA82340/
000080                  SBC         #5                      ; five for sync'ing   /RRA82340/
000081                  TAY                                 ;                     /RRA82340/
000082  S0DELAY         .EQU        *                       ;                     /RRA82340/
000083                  JSR         WEXIT2                  ;(12)                 /RRA82340/
000084                  JSR         WEXIT2                  ;(12)                 /RRA82340/
000085                  PHA                                 ;(3)                  /RRA82340/
000086                  PLA                                 ;(4)                  /RRA82340/
000087                  NOP                                 ;(2)                  /RRA82340/
000088                  NOP                                 ;(2)                  /RRA82340/
000089                  DEY                                 ;(2)                  /RRA82340/
000090                  BNE         S0DELAY                 ;(3)                  /RRA82340/
000091
000092  ; See if we have a sector-zero here:
000093
000094                  LDX         #60
000095                  JSR         RDADR16                 ;READ NEXT ADDRESS FIELD.
000096                  BCS         REWRITE                 ;ERR, LOCATE SECT 15 AND REWRITE TRK.
000097                  LDA         SECT                    ;WAS IT SECTOR 0?
000098                  BEQ         VDATA                   ;YES, NOW VERIFY DATA FIELD.
000099
000100  ; NO SECTOR 0? THEN SYNC GAPS ARE TOO LARGE. SHRINK 'EM.
000101
000102                  DEC         NSYNC
000103                  LDA         NSYNC
000104                  CMP         DCB_MINSYNC             ;IF TOO SMALL, UNRECOVERABLE /RRA82237/
000105                  BCS         REWRITE                 ;OK, REWRITE AFTER DATA FLD 15.
000106                  SEC                                 ;DRIVE EXTREMELY FAST    /RRA82237/
000107                  RTS                                 ; (CALLER WILL CHECK NSYNC)   /RRA82237/
000108
000109  ; Make sure we can read the track we just wrote...
000110
000111  VSECT           LDX         #60
000112                  JSR         RDADR16                 ;READ AN ADDRESS FIELD.
000113                  BCS         VERR1                   ;RETRY IF ERR.
000114
000115  VDATA           LDX         #60
000116                  JSR         READ16                  ;READ DATA FIELD.
000117                  BCS         VERR1                   ;=>it's bad...        /RRA82258/
000118
000119                  LDY         SECT                    ;THIS IS SECTOR READ.
000120                  LDA         FOUND,Y                 ;ALREADY FOUND?
000121                  BMI         VERR1                   ;YES, IGNORE IT.
000122
000123                  LDA         #0FF
000124                  STA         FOUND,Y                 ;INDICATE THIS SECT NOW FOUND.
000125                  DEC         NSECT                   ;FOUND 16 SECTORS?
000126                  BPL         VSECT                   ;NO, LOOK FOR NEXT.
000127  ;       REMOVED 'BLIND DECREMENT' OF NSYNC ON TRACKS 1..34            /RRA82237/
000128  WEXIT1          CLC                                 ;INDICATE NO ERROR.
000129                  RTS                                 ;RETURN.
000130
000131  VERR1           DEC         RETRYCNT                ;NEXT OF 48 SECTOR TRIES.
000132                  BNE         VSECT                   ;(KEEP TRYING)
000133
000134  ; We can't get a good verify...maybe we should rewrite the track.
000135
000136                  DEC         TRKRETRY                ;MORE RETRIES?        /RRA82237/
000137                  BNE         REWRITE                 ;->YES, TRY AGAIN     /RRA82237/
000138                  SEC                                 ;UNABLE TO WRITE & VERIFY   /RRA82237/
000139                  RTS                                 ;                     /RRA82237/
000140
000141  ; Read Sector 15 data to position for track retry...
000142
000143  REWRITE         .EQU        *                       ;                     /RRA82258/
000144                  LDA         DCB_SECTRETRY           ;MAX RETRIES          /RRA82258/
000145                  ASL         A                       ; *2 for good find    /RRA82258/
000146                  STA         RETRYCNT                ;                     /RRA82258/
000147
000148  S15LOC          LDX         #60
000149                  JSR         RDADR16                 ;READ ADDRESS FIELD.
000150                  BCS         NOTS15                  ;ERR, TRY UP TO 128 TIMES.
000151                  LDA         SECT                    ;SECTOR THAT WAS READ.
000152                  CMP         #15.                    ; SECTOR 15?
000153                  BEQ         PASS15                  ;YES, GO FOR IT.      /RRA82258/
000154  NOTS15          DEC         RETRYCNT
000155                  BNE         S15LOC                  ;TRY FOR SECT 15 AGAIN.
000156                  SEC                                 ;SET CARRY TO INDICATE VERIFY ERR.
000157  WEXIT2          RTS                                 ;AND RETURN TO FORMATTER.
```

```
000158
000159  ; Delay to let most of sector 15's data pass by, then go back and     /RRA82258/
000160  ;  rewrite the track (including the 128-nibbles of seam). This assures/RRA82258/
000161  ;  that we will rewrite the track entirely, and that sector 0 will be /RRA82258/
000162  ;  aligned at roughly the same place as it was on the prior try.      /RRA82258/
000163
000164  PASS15          .EQU        *                       ;                       /RRA82258/
000165                  LDX         #342.-128.              ;LET MOST GO BY         /RRA82258/
000166  VFYDELAY        .EQU        *                       ;                       /RRA82258/
000167                  JSR         WEXIT2                  ;(12)                   /RRA82258/
000168                  JSR         WEXIT2                  ;(12)                   /RRA82258/
000169                  BIT         0                       ;( 3)                   /RRA82258/
000170                  DEX                                 ;( 2)                   /RRA82258/
000171                  BNE         VFYDELAY                ;( 3)                   /RRA82258/
000172                  JMP         WTRETRY                 ;WRITE TRACK FROM HERE IF NO ERR.

; ###############################################################################################################
; #   END OF FILE:  FMTWTRACK.TEXT
; #   LINES      :  172
; #   CHARACTERS :  10797
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (06 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ###############################################################################################################

###
```