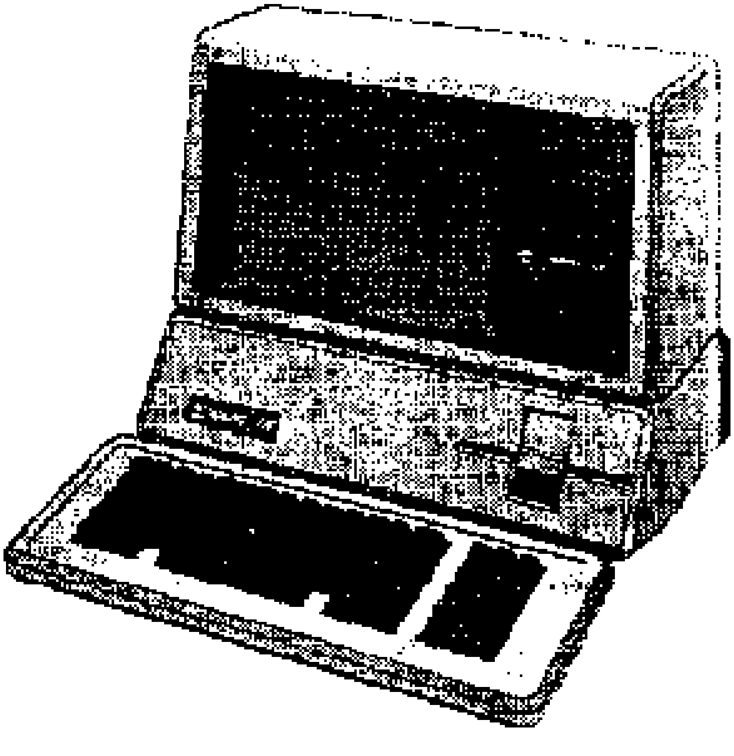
 Apple /// Computer Technical Information

Apple ///
Parallel Printer Port Driver 1.31 Source Code Listing



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

FORMATTED LISTING

```
; ####################################################################################################
; #   PROJECT  :  Apple /// SOS Parallel Printer Driver 1.31 (6502 Assembly Source Code)
; #   FILE NAME:  PARPRINT.text
; ####################################################################################################

                .NOPATCHLIST
                .TITLE       "SOS Parallel Printer Driver -- 1.31  17-Mar-83"
;-----------------------------------------------------------------------
;
;
;               SOS Parallel Printer Driver
;
;               Copyright (C) 1983 by Apple Computer Inc.
;               All rights reserved.
;
;
;       Revisions:
;
;       1.30    14-Jan-83
;               New driver design, replaces version 1.01 22-Sep-81.
;               Default slot number now undefined (was 1).
;               Defined DCB INVERT bit for printers with inverted logic.
;
;       1.31    17-Mar-83
;               Fixed test for ASC_CR in Auto Line Feed processing.
;
;-----------------------------------------------------------------------

DEVTYPE         .EQU         41                        ;Character device, output only, printer
SUBTYPE         .EQU         03
APPLE           .EQU         0001
RELEASE         .EQU         1310
                .PAGE
;-----------------------------------------------------------------------
;
;  The macro SWITCH performs an N way branch based on a switch index.  The
;  maximum value of the switch index is 127 with bounds checking provided
;  as an option.  The macro uses the A and Y registers and alters the C,
;  Z, and N flags of the status register, but the X register is unchanged.
;
;               SWITCH  [index], [bounds], adrs_table, [*]
;
;       index   This is the variable that is to be used as the switch index.
;               If omitted, the value in the accumulator is used.
;
;      bounds   This is the maximum allowable value for index.  If index
;               exceeds this value, the carry bit will be set and execution
;               will continue following the macro.  If bounds is omitted,
;               no bounds checking will be performed.
;
;  adrs_table   This is a table of addresses (low byte first) used by the
;               switch.  The first entry corresponds to index zero.
;
;           *   If an asterisk is supplied as the fourth parameter, the
;               macro will push the switch address but will not exit to
;               it; execution will continue following the macro.  The
;               program may then load registers or set the status before
;               exiting to the switch address.
;
;-----------------------------------------------------------------------

                .MACRO       SWITCH
                .IF          "%1" <> ""                ;If PARM1 is present,
                LDA          %1                        ;  Load A with switch index
                .ENDC
                .IF          "%2" <> ""                ;If PARM2 is present,
                CMP          #%2+1                     ;  Perform bounds checking
                BCS          $3579                     ;  on switch index
                .ENDC
                ASL          A
                TAY
                LDA          %3+1,Y                    ;Get switch address from table
                PHA                                    ;  and push onto stack
                LDA          %3,Y
                PHA
                .IF          "%4" <> "*"               ;If PARM4 is omitted,
                RTS                                    ;  Exit to code
                .ENDC                                  ;Otherwise, drop through
                .IF          "%2" <> ""
$3579
                .ENDC
                .ENDM
                .PROC        PRINTER
                .WORD        0FFFF
                .WORD        68.
                .ASCII       "Parallel Printer Driver -- "
                .ASCII       "Copyright (C) 1983 by Apple Computer Inc."
;-----------------------------------------------------------------------
;
```

```
;       Device Handler Identification Block
;
;----------------------------------------------------------------------
IDBLK           .WORD        0000                      ;Link to next device handler
                .WORD        LP_MAIN                   ;Entry point address
                .BYTE        8                         ;Length of device name
                .ASCII       ".PRINTER       "
                .BYTE        80                        ;Device # (Mark active)
SLOT            .BYTE        0FF                       ;Slot # of Parallel card
                .BYTE        00                        ;Unit #
                .BYTE        DEVTYPE
                .BYTE        SUBTYPE
                .BYTE        00
                .WORD        0000
                .WORD        APPLE
                .WORD        RELEASE
;----------------------------------------------------------------------
;
;       Device Handler Configuration Block
;
;----------------------------------------------------------------------
                .WORD        5
RDYMASK         .BYTE        060                       ;Mask for valid ready bits
RDYSTAT         .BYTE        040                       ;Printer ready status value
;--------+--------+--------+--------+--------+--------+--------+--------+
;  Power | Select |Pap. Out|Rib. Out|  Check |        |        |        |
;--------+--------+--------+--------+--------+--------+--------+--------+
PROPTNS         .BYTE        00                        ;Printer options
;--------+--------+--------+--------+--------+--------+--------+--------+
; Invert | Auto LF|        |        |        |        |        |        |
;--------+--------+--------+--------+--------+--------+--------+--------+
CTRLVAL         .BYTE        00                        ;Control word for UPIC
;--------+--------+--------+--------+--------+--------+--------+--------+
;    0   |    0   | STB pos|    0   | ACK pos| STB +8 | STB +4 | STB -2 |
;--------+--------+--------+--------+--------+--------+--------+--------+
TIMEOUT         .BYTE        64                        ;Timeout count for ACK
                .PAGE
;----------------------------------------------------------------------
;
;       SOS Device Handler Interface
;
;----------------------------------------------------------------------

SOSINT          .EQU         0C0
REQCODE         .EQU         SOSINT+0                  ;SOS request code
BUFFER          .EQU         SOSINT+2                  ;Buffer pointer
REQCNT          .EQU         SOSINT+4                  ;Requested count
CTLSTAT         .EQU         SOSINT+2                  ;Control/status code
CSLIST          .EQU         SOSINT+3                  ;Control/status list pointer


;----------------------------------------------------------------------
;
;       SOS Global Subroutines
;
;----------------------------------------------------------------------

ALLOCSIR        .EQU         1913                      ;SOS resource allocation
DEALCSIR        .EQU         1916                      ;SOS resource deallocation
SYSERR          .EQU         1928                      ;SOS error return


;----------------------------------------------------------------------
;
;       SOS Error Codes
;
;----------------------------------------------------------------------

XREQCODE        .EQU         20                        ;Invalid request code
XCTLCODE        .EQU         21                        ;Invalid control/status code
XNOTOPEN        .EQU         23                        ;Device not open
XNOTAVIL        .EQU         24                        ;Device not available
XNORESRC        .EQU         25                        ;Resouce not available
XBADOP          .EQU         26                        ;Invalid operation for device


;----------------------------------------------------------------------
;
;       Misecllaneous Equates
;
;----------------------------------------------------------------------

TRUE            .EQU         80
FALSE           .EQU         00
ASC_LF          .EQU         0A
ASC_CR          .EQU         0D
BITON7          .EQU         80
BITOFF7         .EQU         7F
                .PAGE
;----------------------------------------------------------------------
;
;       Local Variables
;
```

```
;------------------------------------------------------------------------

OPENFLG         .BYTE        FALSE                     ;Device open flag
CRFLAG          .BYTE        FALSE                     ;Previous character was CR
INVERT          .BYTE        0                         ;XOR with data for inverted logic
DEVOFF          .BYTE        0                         ;Offset to I/O addresses
CTRLWRD         .BYTE        0                         ;Parallel card control word

;  Consumer State -- The Consumer State is the current operational status
;  of the interrupt handler.  The Idle state indicates that the consumer
;  has no data to send, and there is no pending ACK from the printer.  The
;  Waiting for ACK state indicates that data has been STROBEd but the ACK
;  is still pending.  The Waiting for Ready state indicates that data was
;  STROBEd when the printer was not Ready to accept it.  The printer must
;  ACK when it becomes Ready, then the consumer will STROBE again.

CSTATE          .BYTE        0                         ;Consumer State
IDLE            .EQU         000                       ;  Idle, waiting for data
WT_ACK          .EQU         080                       ;  Waiting for ACK
WT_RDY          .EQU         0C0                       ;  Waiting for Ready


;  Local Buffer -- Logically, the Local Buffer contains $FF bytes and the
;  Buffer Count may range from $00 through $FF.  Physically, however, the
;  Local Buffer contains $100 bytes so the producer and consumer pointers
;  can be incremented without testing them for end of buffer or wrap around.

BUFCNT          .BYTE        0                         ;Local buffer byte count
PRODPTR         .BYTE        0                         ;Producer buffer pointer
CSMRPTR         .BYTE        0                         ;Consumer buffer pointer
LOCBUF          .EQU         *                         ;Local buffer
                .ASCII       "Copyright (C) 1983 by Apple Computer Inc."
                .BLOCK       *-LOCBUF^0FF+1            ;Save $100 bytes for Local Buffer


SIR_TBL         .WORD        SIR_NUM                   ;System Resource Table
SIR_NUM         .BYTE        0                         ;SIR Number
                .BYTE        0
                .WORD        LP_INTR                   ;Address of interrupt handler
INTR_BANK       .BYTE        0                         ;Bank number
SIR_CNT         .EQU         *-SIR_NUM
                .PAGE
;------------------------------------------------------------------------
;
;       Hardware I/O Addresses
;
;------------------------------------------------------------------------

B_REG           .EQU         0FFEF                     ;Bank Register
E_REG           .EQU         0FFDF                     ;Environment Register
ANYSLOT         .EQU         0FFDD                     ;Any Slot Interrupt Flag


;------------------------------------------------------------------------
;
;       Universal Parallel Interface Card
;
;   Note:  to generate a STROBE (write to C082, or write to C080 with
;     AutoStrobe enabled), the CPU must be in Fixed Speed mode.  All
;     other registers may be accessed in either Fixed or Full Speed mode.
;
;------------------------------------------------------------------------

PORTA           .EQU         0C080                     ;Output port A
PORTB           .EQU         0C081                     ;Output port B
READB           .EQU         0C083                     ;Input port
STROBE          .EQU         0C082                     ;Send strobe
DATARDY         .EQU         0C085                     ;Set DRDY, clear DRAK
CLEAR           .EQU         0C087                     ;Clear DRDY, AutoStrobe
;  Set ACK, DRAK

STATUS          .EQU         0C084                     ;UPIC Status Register

;    7        6        5        4        3        2        1        0
;--------+--------+--------+--------+--------+--------+--------+--------+
;   ACK  |  DRAK  |                                            |  ACK   |
; Latched| Latched|                                            |  Level |
;--------+--------+--------+--------+--------+--------+--------+--------+


CTRLREG         .EQU         0C086                     ;UPIC Control Register

;    7        6        5        4        3        2        1        0
;--------+--------+--------+--------+--------+--------+--------+--------+
;    Int Enable   |    Strobe/Ack Polarity   |    Strobe Width = 3 uS   |
;--------+--------+--------+--------+--------+--------+--------+--------+
;   ACK  |  DRAK  |   STB  |  DRDY  |ACK/DRAK|   +8   |   +4   |   -2   |
;--------+--------+--------+--------+--------+--------+--------+--------+
                .PAGE
;------------------------------------------------------------------------
;
;       Parallel Printer Driver -- Mainline
;
```

```
;----------------------------------------------------------------------

LP_MAIN         SWITCH       REQCODE,8,LP_TBL

BADREQ          LDA          #XREQCODE                 ;Invalid request code
                JSR          SYSERR
NOTOPEN         LDA          #XNOTOPEN                 ;Device not open
                JSR          SYSERR

LP_TBL          .WORD        LP_READ-1
                .WORD        LP_WRITE-1
                .WORD        LP_STAT-1
                .WORD        LP_CNTL-1
                .WORD        BADREQ-1
                .WORD        BADREQ-1
                .WORD        LP_OPEN-1
                .WORD        LP_CLOSE-1
                .WORD        LP_INIT-1



;----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Initialization
;
;----------------------------------------------------------------------

LP_INIT         .EQU         *
                LDY          SLOT                      ;Check for valid slot #
                DEY
                CPY          #04                       ;If carry set out of range
                RTS                                    ;SOS will mark inactive
                .PAGE
;----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Open
;
;----------------------------------------------------------------------

LP_OPEN         .EQU         *
                BIT          OPENFLG                   ;Printer open?
                BPL          $010                      ;No
                LDA          #XNOTAVIL
                JSR          SYSERR

$010            LDA          SLOT
                ORA          #10
                STA          SIR_NUM                   ;Set up resource number
                ASL          A
                ASL          A
                ASL          A
                ASL          A
                STA          DEVOFF                    ;Offset to slot addresses
                LDA          B_REG
                AND          #0F
                STA          INTR_BANK                 ;Set up interrupt bank
                LDA          #SIR_CNT
                LDX          SIR_TBL
                LDY          SIR_TBL+1
                JSR          ALLOCSIR                  ;Allocate resource
                BCC          $020
                LDA          #XNORESRC                 ;Can't allocate resource
                JSR          SYSERR

$020            LDA          #IDLE
                STA          CSTATE
                JSR          CNTL00
                LDA          #TRUE
                STA          OPENFLG
                RTS
                .PAGE
;----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Close
;
;----------------------------------------------------------------------

LP_CLOSE        .EQU         *
                ASL          OPENFLG                   ;Printer open ?
                BCS          $010                      ;Yes, closed now
                JMP          NOTOPEN                   ;No, error request

$010            JSR          CNTL00                    ;Reset UPIC
                LDA          #SIR_CNT
                LDX          SIR_TBL
                LDY          SIR_TBL+1
                JSR          DEALCSIR                  ;Deallocate resource
                RTS



;----------------------------------------------------------------------
;
```

```
;       Parallel Printer Driver -- Read
;
;-----------------------------------------------------------------------

LP_READ         .EQU         *
                BIT          OPENFLG                   ;Printer open ?
                BMI          $010                      ;Yes
                JMP          NOTOPEN                   ;No, return error

$010            LDA          #XBADOP
                JSR          SYSERR
                .PAGE
;-----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Write / Producer
;
;-----------------------------------------------------------------------

LP_WRITE        .EQU         *
                BIT          OPENFLG                   ;Printer open ?
                BMI          $010                      ;Yes
                JMP          NOTOPEN                   ;No, return error
$010            LDA          REQCNT                    ;One's Complement of REQCNT
                EOR          #0FF                      ;  to implement main loop:
                STA          REQCNT                    ;  While (RC:=RC+1)<>0 Do ...
                LDA          REQCNT+1
                EOR          #0FF
                STA          REQCNT+1

$020            INC          REQCNT                    ;Increment byte count
                BNE          $030
                INC          REQCNT+1
                BEQ          $090                      ;All done
$030            LDY          #0
                LDA          (BUFFER),Y                ;Get character
                INC          BUFFER                    ;Increment buffer address
                BNE          $040
                INC          BUFFER+1
                BNE          $040
                LDX          #80
                STX          BUFFER+1
                INC          1401+BUFFER
$040            BIT          PROPTNS                   ;Auto Line Feed?
                BVC          $060                      ;  No
                ASL          CRFLAG                    ;Last character CR?
                BCC          $050                      ;  No
                CMP          #ASC_LF                   ;This character LF?
                BEQ          $020                      ;  Yes -- ignore it
$050            CMP          #ASC_CR                   ;This character CR?
                BNE          $060                      ;  No
                ROR          CRFLAG                    ;  Yes -- set CR flag
$060            LDX          PRODPTR
                STA          LOCBUF,X                  ;Store character in local buffer
$070            LDX          BUFCNT                    ;Check buffer count for max
                INX
                BEQ          $080
                INC          PRODPTR                   ;Bump producer pointer
                INC          BUFCNT                    ;  and buffer count
                JMP          $020                      ;Loop

$080            JSR          PRIME                     ;Prime the consumer and
                JMP          $070                      ;  recheck buffer count

$090            JSR          PRIME                     ;Make sure consumer is running
                RTS
                .PAGE
;-----------------------------------------------------------------------
;
;       Subroutine PRIME
;
;       Start up Consumer, and wait until printer is ready.
;
;-----------------------------------------------------------------------

PRIME           .EQU         *
                BIT          CSTATE
                BVC          $020
$010            LDY          DEVOFF
                LDA          READB,Y                   ;Wait for Ready
                EOR          RDYSTAT
                AND          RDYMASK
                BNE          $010
$020            PHP
                SEI
                JSR          LP_INTR                   ;Call interrupt handler
                LDA          E_REG
                AND          #BITOFF7
                STA          E_REG                     ;Restore Full Speed
                PLP
                BIT          CSTATE
                BVS          $010
                RTS

```

```
;-----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Interrupt Handler / Consumer
;
;     This routine must be CALLED with INTERRUPTS DISABLED.  During
;     execution, it selects Fixed Speed mode and ENABLES INTERRUPTS.
;     It always EXITS in Fixed Speed mode with INTERRUPTS DISABLED.
;
;-----------------------------------------------------------------------

LP_INTR         LDA          E_REG
                ORA          #BITON7                   ;Set Fixed Speed
                STA          E_REG
                LDY          DEVOFF                    ;Get offset to card addresses
                LDA          CTRLWRD
                AND          #BITOFF7                  ;Disable card's interrupt
                STA          CTRLWRD
                STA          CTRLREG,Y
                LDA          #02
                STA          ANYSLOT                   ;Clear 'Any Slot' interrupt
                CLI
                .PAGE
                BIT          CSTATE                    ;Test Int.Handler State
                BPL          $020                      ;  Send a character
                LDX          STATUS,Y
                BVC          $010                      ;  Waiting for ACK
                LDA          READB,Y                   ;Check for READY
                EOR          RDYSTAT
                AND          RDYMASK
                BEQ          $030                      ;Re-strobe printer
                TXA
                EOR          #BITON7
                LDX          #WT_RDY                   ;Wait for Ready
                BNE          $070                      ;  (unconditional branch)

$010            BPL          $050                      ;Wait for ACK

$020            LDA          BUFCNT                    ;Check buffer count
                BEQ          $080                      ;Wait for data
                LDX          CSMRPTR
                LDA          LOCBUF,X                  ;Get character from buffer
                EOR          INVERT
                STA          PORTA,Y                   ;  and send it to printer
                DEC          BUFCNT
                INC          CSMRPTR
$030            STA          STROBE,Y                  ;Stobe Printer
                LDX          #WT_RDY
                LDA          READB,Y                   ;Printer Ready?
                EOR          RDYSTAT
                AND          RDYMASK
                BNE          $060                      ;  No
                LDX          TIMEOUT
$040            LDA          STATUS,Y                  ;Wait for ACK or Time Out
                BMI          $020                      ;  ACK, send next character
                DEX
                BPL          $040
$050            LDX          #WT_ACK                   ;Wait for ACK

$060            LDA          #BITON7                   ;Enable Interrupt

$070            SEI
                STX          CSTATE                    ;Set Int.Handler State
                AND          #BITON7                   ;Enable Interrupt with
                ORA          CTRLWRD                   ;  bit 7 of Accumulator
                STA          CTRLWRD
                STA          CTRLREG,Y
                RTS

$080            SEI
                LDX          #IDLE
                STX          CSTATE                    ;Wait for data
                RTS
                .PAGE
;-----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Status
;
;-----------------------------------------------------------------------

LP_STAT         BIT          OPENFLG                   ;Printer open ?
                BMI          $010
                JMP          NOTOPEN                   ;No, return error

$010            LDY          #00
                LDX          CTLSTAT
                BEQ          STAT00
                DEX
                BEQ          STAT01
                DEX
                BEQ          STAT02
                DEX
                BEQ          STAT03
BADCTL          LDA          #XCTLCODE                 ;Invalid control code
                JSR          SYSERR

STAT00          RTS                                    ;0 -- NOP

STAT01          TYA                                    ;1 -- Status table
                STA          (CSLIST),Y
                RTS

STAT02          TYA                                    ;2 -- New line
                STA          (CSLIST),Y
                RTS

STAT03          LDX          DEVOFF                    ;3 -- Error status and buffer size
                LDA          READB,X
                EOR          RDYSTAT
                STA          (CSLIST),Y
                INY                                    ;Error status byte
                LDA          #0FF                      ;Buffer size, low byte
                STA          (CSLIST),Y
                JSR          $010                      ; high byte
                INY
                LDA          BUFCNT                    ;Number of chars in buffer, low
                STA          (CSLIST),Y
$010            INY
                LDA          #00                       ; high byte
                STA          (CSLIST),Y
                RTS
                .PAGE
;----------------------------------------------------------------------
;
;       Parallel Printer Driver -- Control
;
;----------------------------------------------------------------------

LP_CNTL         BIT          OPENFLG                   ;Printer open ?
                BMI          $010                      ;Yes
                JMP          NOTOPEN

$010            LDX          CTLSTAT
                BEQ          CNTL00
                DEX
                BEQ          CNTL01
                DEX
                BEQ          CNTL02
                JMP          BADCTL                    ;Invalid request number

CNTL00          .EQU         *
                BIT          CSTATE                    ;Test consumer state
                BPL          $020                      ;  Ready
$010            JSR          PRIME                     ;Prime consumer and
                BIT          CSTATE                    ;  recheck state
                BMI          $010
$020            LDA          #00
                STA          BUFCNT                    ;Set buffer count to 0
                STA          PRODPTR                   ;Set producer pointer
                STA          CSMRPTR                   ;Set consumer pointer
                STA          CRFLAG                    ;Set CR switch false
                BIT          PROPTNS                   ;Check invert bit and
                BPL          $030
                LDA          #0FF
$030            STA          INVERT                    ;  set INVERT value
                PHP
                SEI
                LDY          DEVOFF
                LDA          CTRLVAL
                AND          #2F
                STA          CTRLWRD                   ;Init UPIC control register
                STA          CTRLREG,Y
                STA          CLEAR,Y                   ;Disable auto strobe
                PLP
                RTS


CNTL01          RTS


CNTL02          RTS                                    ;2 New line
                .END

```

```
; ####################################################################################################
; #   END OF FILE:  PARPRINT.text
; #   LINES      :  630
; #   CHARACTERS :  32172
; #   Formatter  :  Assembly Language Reformatter 1.0.2 (07 January 1998)
; #   Author     :  David T. Craig -- 71533.606@compuserve.com -- Santa Fe, New Mexico USA
; ####################################################################################################
```

###