

# Acolyte Retro Computer

## Design Document

By: Steven Chad Burrow

This 8-bit computer runs at W65C02S processor at 3.14 MHz, contains 18KB of general purpose RAM, 30KB of Video RAM, and 16KB ROM (with up to 8 banks). VGA displays of 640x240 monochrome or 320x240 with 4-colors are available. Supports PS/2 Keyboard, SPI EEPROM, SPI Micro SD Card (through adapter), and 1-Voice Square Wave Audio through a 3.5mm audio jack. All I/O operations are through illegal opcodes \$\_3 or \$\_B.

### Memory Map:

\$0000-\$07FF = 2KB System RAM (used by general subroutines and system monitor)

\$0800-\$7FFF = 30KB Video RAM (overscanned)

\$8000-\$BFFF = 16KB General Purpose RAM (used by BASIC and other applications)

\$C000-\$FFFF = 16KB ROM (with up to 8 banks depending on size of ROM used)

### Inputs:

/NMI = KEY-CLK or KEY-DATA

/IRQ = KEY-CLK

/SO = SPI-MISO nor \$\_B-opcode

### Output Commands:

\$03 = SPI-CLK low

\$13 = SPI-MOSI low

\$23 = SPI-EEPROM low

\$33 = SPI-SDCARD low

\$43 = BANK-A low

\$53 = BANK-B low

\$63 = BANK-C low

\$73 = AUDIO-OUT low

\$0B = 4-Color Mode

\$83 = SPI-CLK high

\$93 = SPI-MOSI high

\$A3 = SPI-EEPROM high

\$B3 = SPI-SDCARD high

\$C3 = BANK-A high

\$D3 = BANK-B high

\$E3 = BANK-C high

\$F3 = AUDIO-OUT high

\$1B = Monochrome Mode

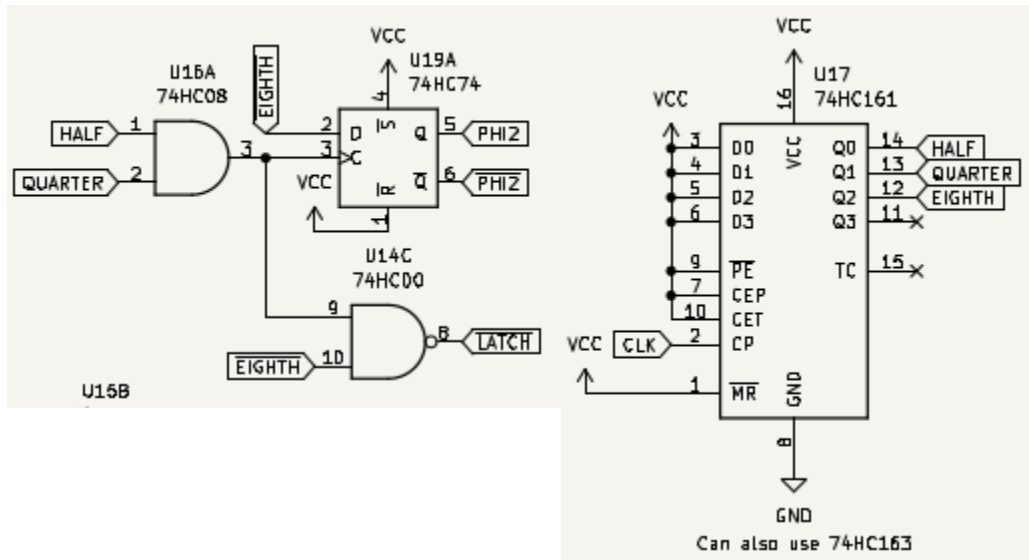
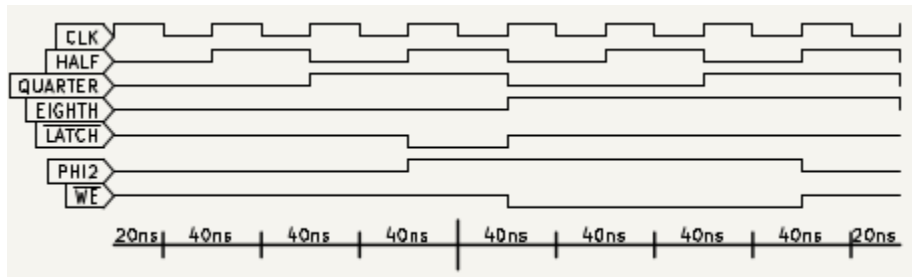


## Preliminary ISRs:

```
; 6502 running at 3.14 MHz,  
; PS/2 keyboard running at 17 kHz  
; That gives me 184 cycles between signals.  
; /IRQ = Keyboard-Clock  
; /NMI = Keyboard-Data OR Keyboard-Clock  
  
irq_isr                                ; 7  
    PHA                                ; 3, by now nmi_isr should trigger  
    LDA key_bit                        ; 4  
    STZ key_bit                        ; 4  
    ROR A                              ; 2  
    ROL key_data                      ; 6  
    DEC key_counter                   ; 6  
    BEQ irq_store                     ; 2  
    LDA key_counter                   ; 4  
    CMP #$FD                          ; 2  
    BEQ irq_reset                     ; 2  
    PLA                                ; 4  
    RTI                               ; 6, total = 52  
irq_store                              ; 1, sub-total = 36  
    PHX                                ; 3  
    LDA key_data                      ; 4  
    LDX key_write                     ; 4  
    STA key_array,X                  ; 5  
    INC key_write                     ; 6  
    PLX                                ; 4  
    PLA                                ; 4  
    RTI                               ; 6, total = 73  
irq_reset                              ; 1, sub-total = 44  
    LDA #$0A                         ; 2  
    STA key_counter                   ; 4  
    PLA                                ; 4  
    RTI                               ; 6, total = 61  
  
nmi_isr                                ; 7  
    INC key_bit                       ; 6  
    RTI                               ; 6, total = 19
```



## Clock Timing:



CLK runs at 25.175 MHz.

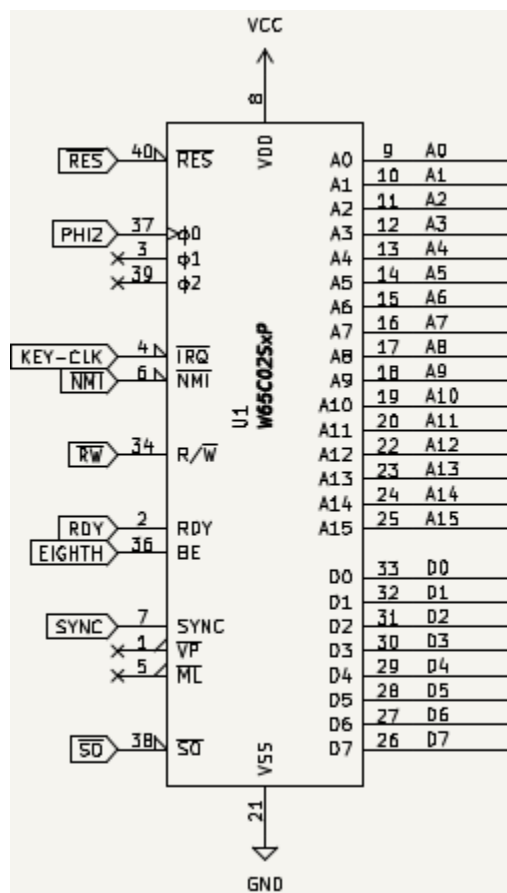
/LATCH is the time where the Video Sync / Reset signals from the ROM are latched, as well as the color data from the RAM is latched.

PHI2 is changed when both HALF and QUARTER go high, offsetting the EIGHTH signal some time.

/WE is qualified to last three-quarters of PHI2-high.



## Processor:



The W65C02S processor is running at 3.14 MHz. Most signals are typical, such as /RES, PHI2, and RDY. Some differences would be:

/IRQ is replaced with KEY-CLK. This only allows the single keyboard to use that interrupt line.

/NMI is derived from KEY-DATA. The NMI-IRQ would simply INC or DEC a byte value in memory, detecting if the KEY-DATA line went low, or didn't change.

/RW has a 10K pull-up when BE goes low.

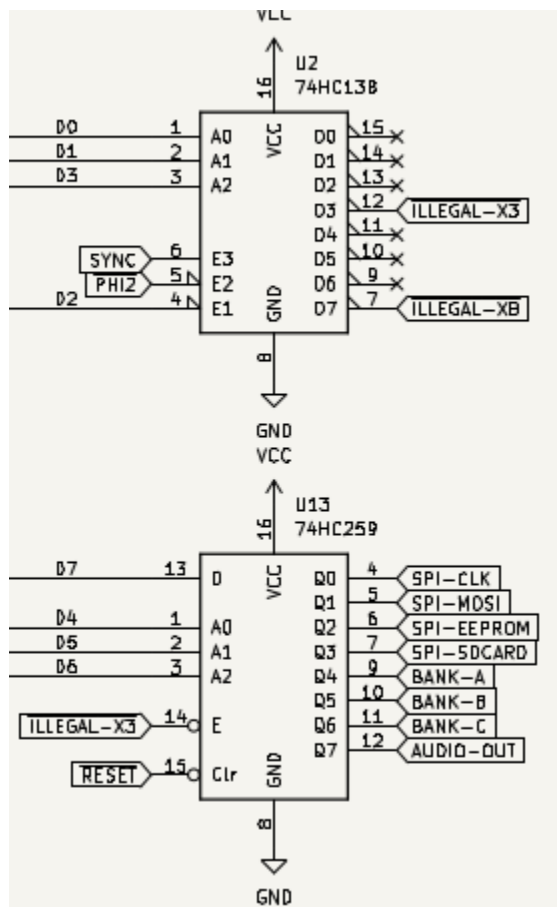
BE is replaced with EIGHTH. Thus half of the time the processor's address and data buses are high-Z, including /RW.

SYNC connects to a 74HC138 to decode the \$\_3 and \$\_B illegal opcodes.

/SO is derived from SPI-MISO, but requires the use of a \$\_B illegal opcode.



## Illegal Opcodes and Outputs:



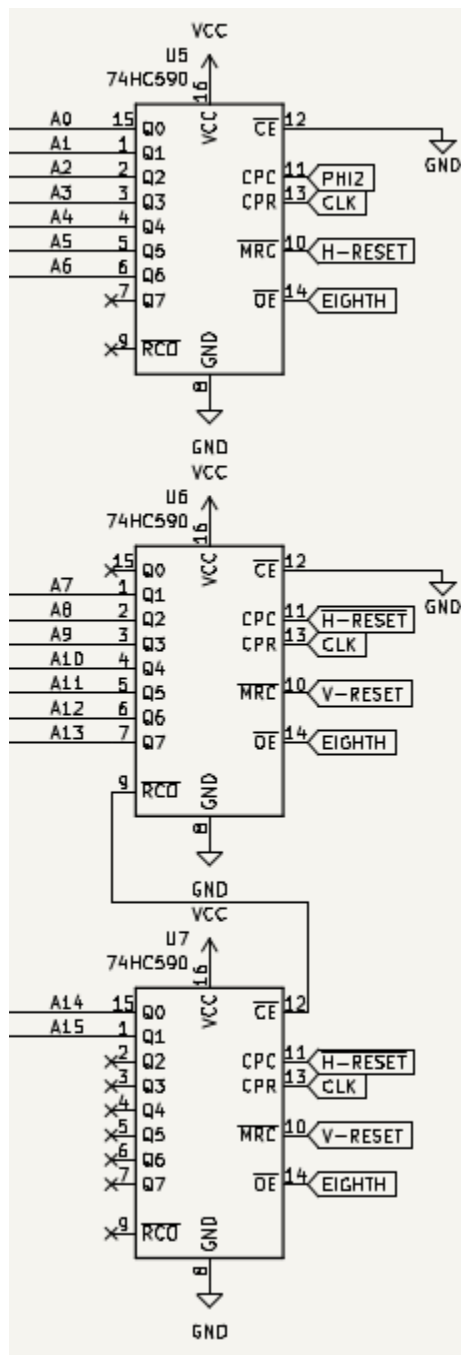
Illegal opcodes `$_3` and `$_B` are decoded with the 74HC138.

When a `$_3` opcode is used, D7 is captured as the output bit's high/low status, and stored in the 74HC259 addressable latch.

BANK-A, BANK-B, and BANK-C all have 10K pull-down resistors, in case the 74HC259 is absent. Likewise the `/ILLEGAL-XB` line has a 10K pull-up resistor in case the 74HC138 is absent.



## Counters:



The 74HC590 counters are strictly for VGA signal gathering. Each /OE line is connected to EIGHTH, which is the same as the BE line on the 6502. Thus when the 6502 is high-Z the 74HC590's are enabled, and when the 6502 is enabled the 74HC590's are high-Z.

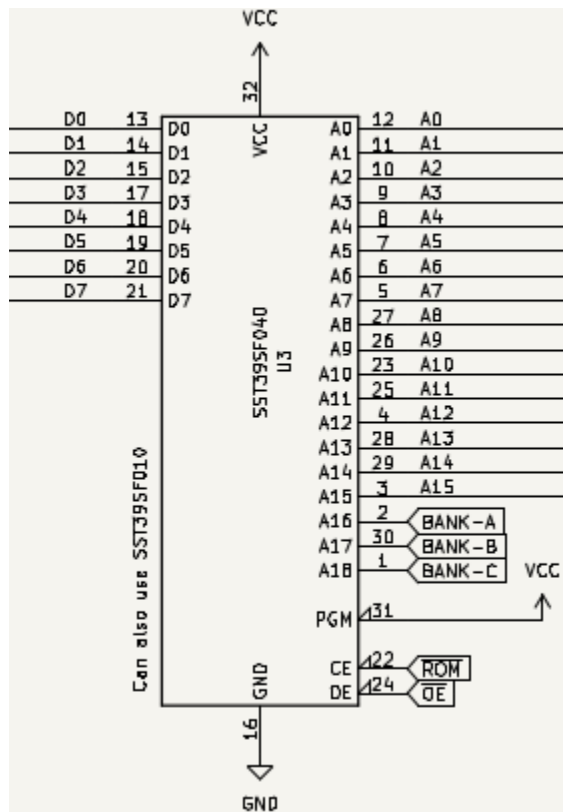
The first 74HC590 is the horizontal counters, reaching 128 bytes per scanline (overscanned 48 bytes). The other two 74HC590's are vertical counters. A15 is only high for a very brief period, thus the ROM it accesses for sync and reset signals must have video data from \$0000 to about \$8008, allowing the \$BFFF to \$FFFF range free for user ROM code. Each bank of the ROM must have duplicate video sync / reset signals.

Q0 on the middle 74HC590 is unconnected, thus each scanline is duplicated, creating a vertical resolution of 240 lines rather than the possible 480 lines.

Some of the scanlines are visibly blank, so those will be placed from \$0000 to \$07FF to allow zero-page and stack to not be visible on the screen.

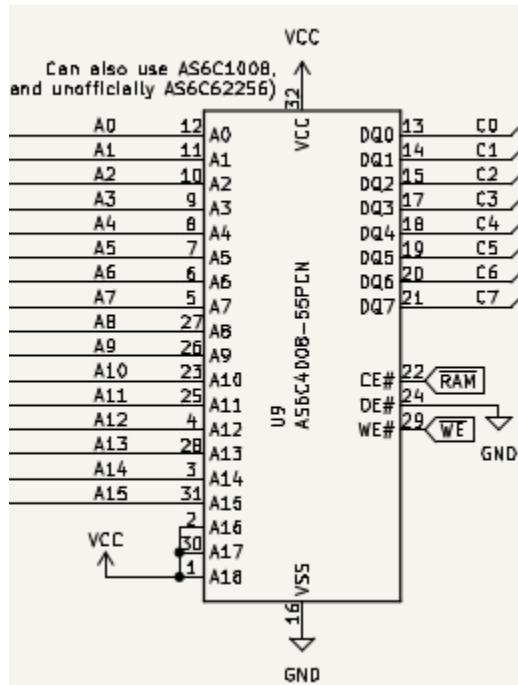


## ROM and Flip-Flop:





## RAM and Transceiver and Shift Registers:

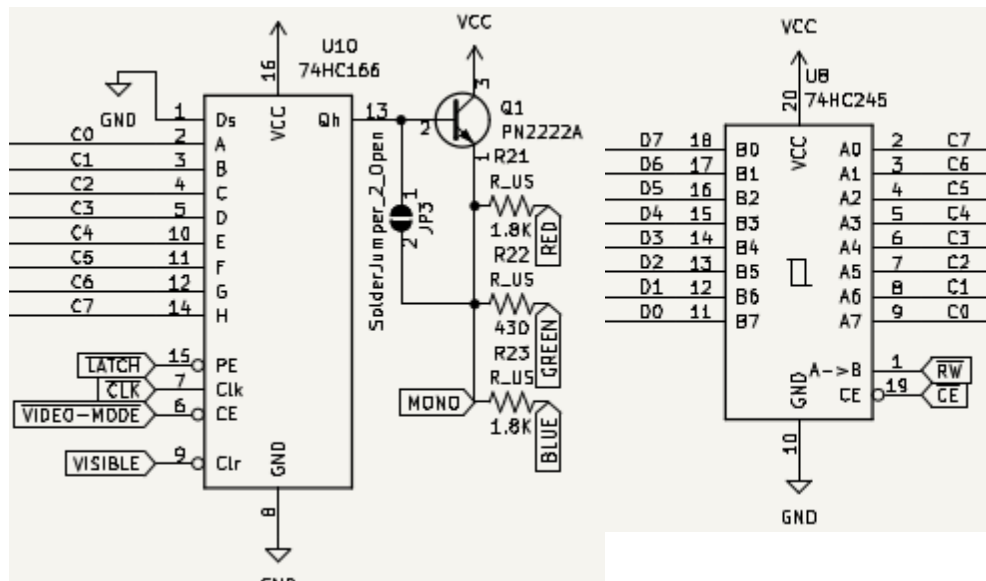


The RAM contains both Video color data, as well as user memory.

The /RAM signal is generated from glue logic, allowing it to be on during PHI2-low (for color data) and PHI2-high (for user memory).

Using an AS6C62256 would essentially remove all RAM from \$8000-\$BFFF, thus it is supported but not recommended.

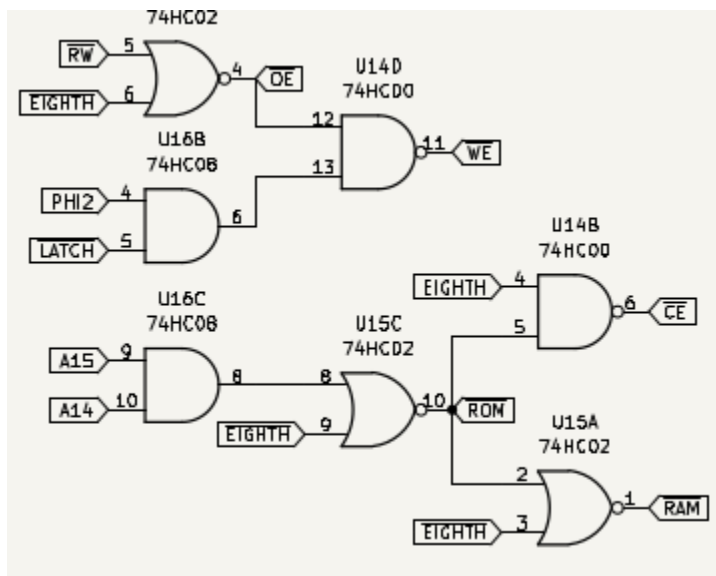
The 74HC245 transceiver directs the flow of the databus to and from the RAM. It is only allowed to be on while PHI2-high and accessing RAM.



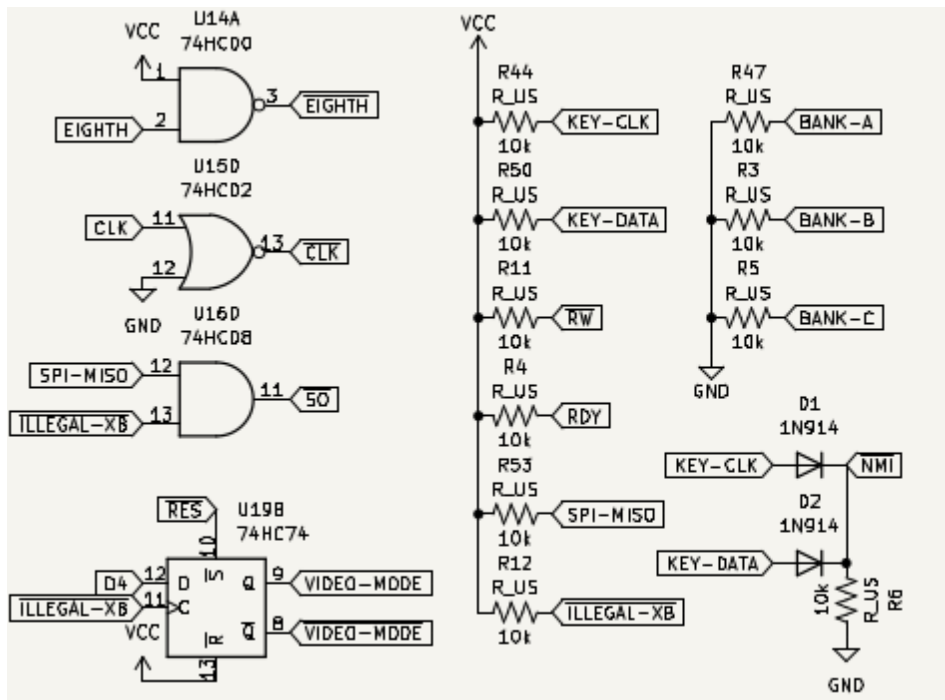
The 74HC166 shift registers hold the color data from the RAM, and shift it out to the VGA connection at 25.175 MHz. The color data is latched at the same time of the video sync / reset signals. Pictured here is only the monochrome IC. The PN2222A transistor is optional.



## Glue Logic:



This is the main glue logic for the board. The main purpose is to get the timing right on particular signals, as well as decode which addresses are used for RAM and ROM. Many signals are NOR'd with **/EIGHTH**, meaning it is low-enabled during the **PHI2**-low cycle for video signals and color data.



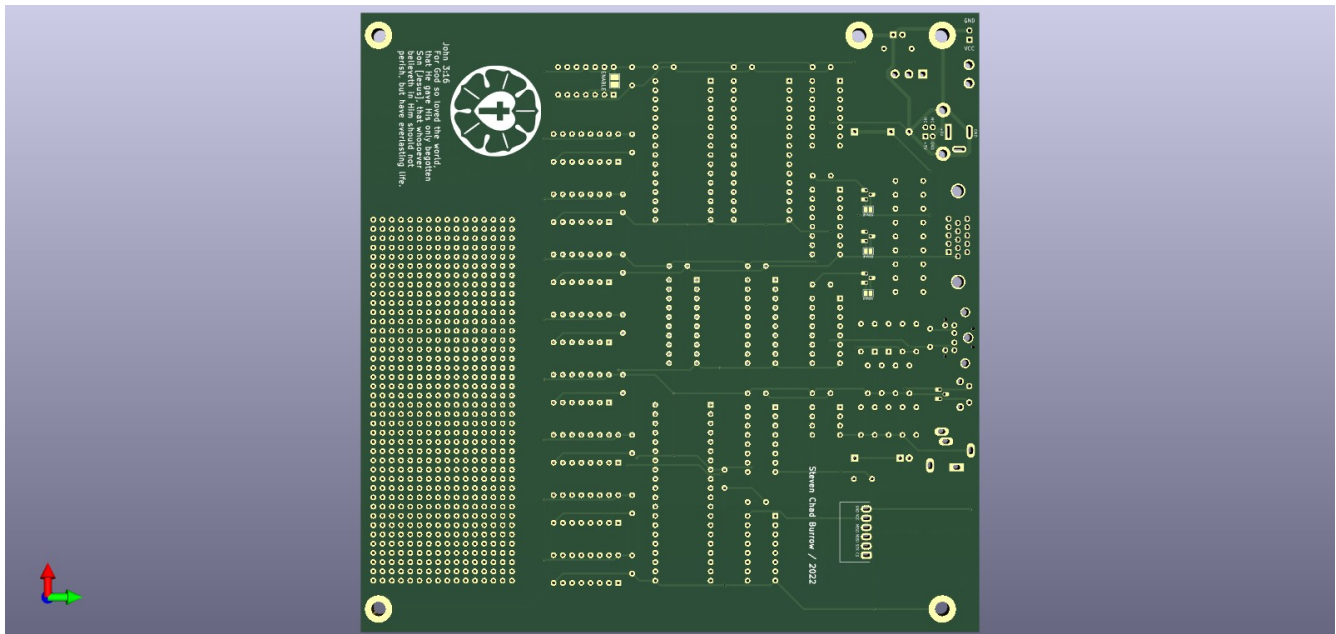
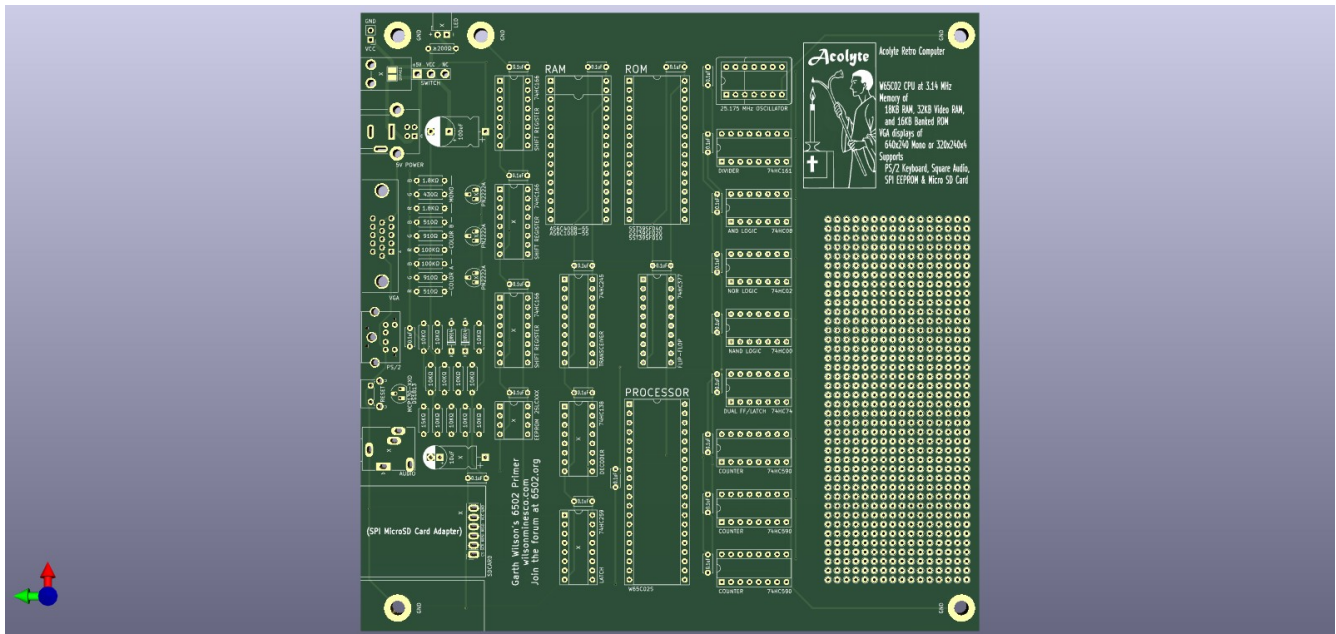
The **VIDEO-MODE** signal tells which shift registers to load, monochrome or color.

Using the **\$\_B** opcode will care must be taken while also using the **SPI-MISO** line (though this would not affect actual data transfer, just how it is displayed on the screen.)

Diode OR-logic is used for **/NMI** as the **PS/2** keyboard signal is slower and no other 74HC' logic gates are available for it.



## Board Layout:



The board itself has a Mini-ITX form factor, so that it could fit in any typical desktop PC case if so desired. There includes a prototyping through-hole area to one side, mainly to use up left-over room on the board, but also to help fix issues that might arise.



## **Pricing and Sourcing:**

The Acolyte Retro Computer is designed to be to simple and low-cost yet high feature. Some IC's are unnecessary for typical operation, and many connectors and switches are optional.

The price goal for a completed board is \$50 for a bare-bones model without a power switch, smaller IC sockets, ROM banking, SPI features, audio, and color. A fully featured board will be no more than \$75 with shipping included.

Prices could go even lower with better parts sourcing and buying in bulk. Currently the two suppliers are JLCPCB.com and Mouser.com

Other design decisions were to use only commonly available parts from major distributors, only through-hole components, and no programmable logic chips. This makes the board approachable to beginners and easy to understand.

Many designs on the board have cost savings in mind, such as using a single RAM and ROM chip, minimizing glue logic, and the exclusion of a 6522 VIA. Simplicity, efficiency, yet utility are the main design goals of this project.



