Digital Systems for VGA Monitor Control: Lab4

Douglas Guillen | dguillen3@horizon.csueastbay.edu
Alexandro Milanez | amilanez@horizon.csueastbay.edu
Keoni Sablan | ksablan3@horizon.csueastbay.edu

1| Introduction

In this lab, we studied the concepts of counting, state machines, and timing in digital systems. The term “digital systems” refers to the implementation of using different concepts to output a Vertical synchronization signal and a horizontal synchronization signal. In the last experiment, we used red, green, and blue components of an analog transmitter to display through a VGA monitor display.

2| Theory

Our group implemented a VGA control module in Verilog HDL that outputs the VSYNC, HSYNC, and digital RGB components.

Experiment 1:

In this experiment we created a counter that divided the 100 Mhz clock on the FPGA to go active high for 75% of the clock and active low on 25% of the clock. We ran a verilog simulation and displayed the waveform on an oscilloscope to ensure the correct waveform was being generated.This gave us a template for a counter/clock divider for experiment 2.

Experiment 2:

In experiment 2 we used the counter template from the previous experiment to create three different counter circuits, a pixel clock, a horizontal synchronization (hsync), and a vertical synchronization (vsync) counter. We used the pixel clock counter to manipulate the 100 Mhz clock signal generated from the FPGA to run a frequency of 25 Mhz, we used the same template to generate the hsync and vsync signals.

Experiment 3:

In this experiment we used the counter circuits from the previous experiment to synchronize and display a color output onto a monitor using a VGA cable. Understanding the addressable video range of 640x480 pixels, calculating the time between the front and back porch of the hsync and vsync signals.These colors will be displayed on a monitor by using a VGA cable from the FPGA board.

3| Experiment

For our lab in all 3 experiments we used the FPGA **(Figure 1)** and the Verilog software to code and create the counters for the VGA controller. In experiments 1 and 2 we used an oscilloscope **(Figure 4)** to display and measure the waveforms generated by the experiments, and in experiment 3 we used a monitor **(Figure 5)** and a VGA cable to display the graphics generated from the experiment onto the monitor.

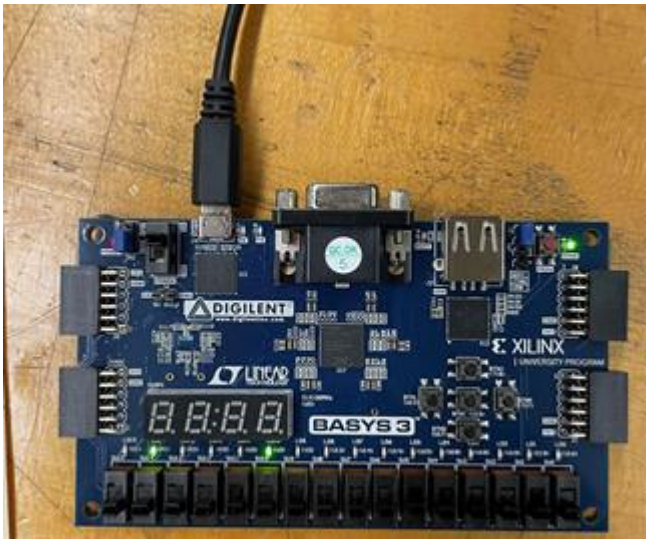

Figure 1: Field Programmable Gate Array.

4| Results

Experiment 1: For our first experiment, we implemented a counter in verilog to act as a clock divider creating an output high for 75% of the clock and low for 25% of the clock. We ran a simulation test bench **(Figure 2)** to make sure our clock divider was working properly making sure that the gclk waveform triggered at the positive edge of the clock or at active low rest. We then programmed the FPGA and used an oscilloscope **(Figure 4)** to display and measure the waveforms outputted from the FPGA.

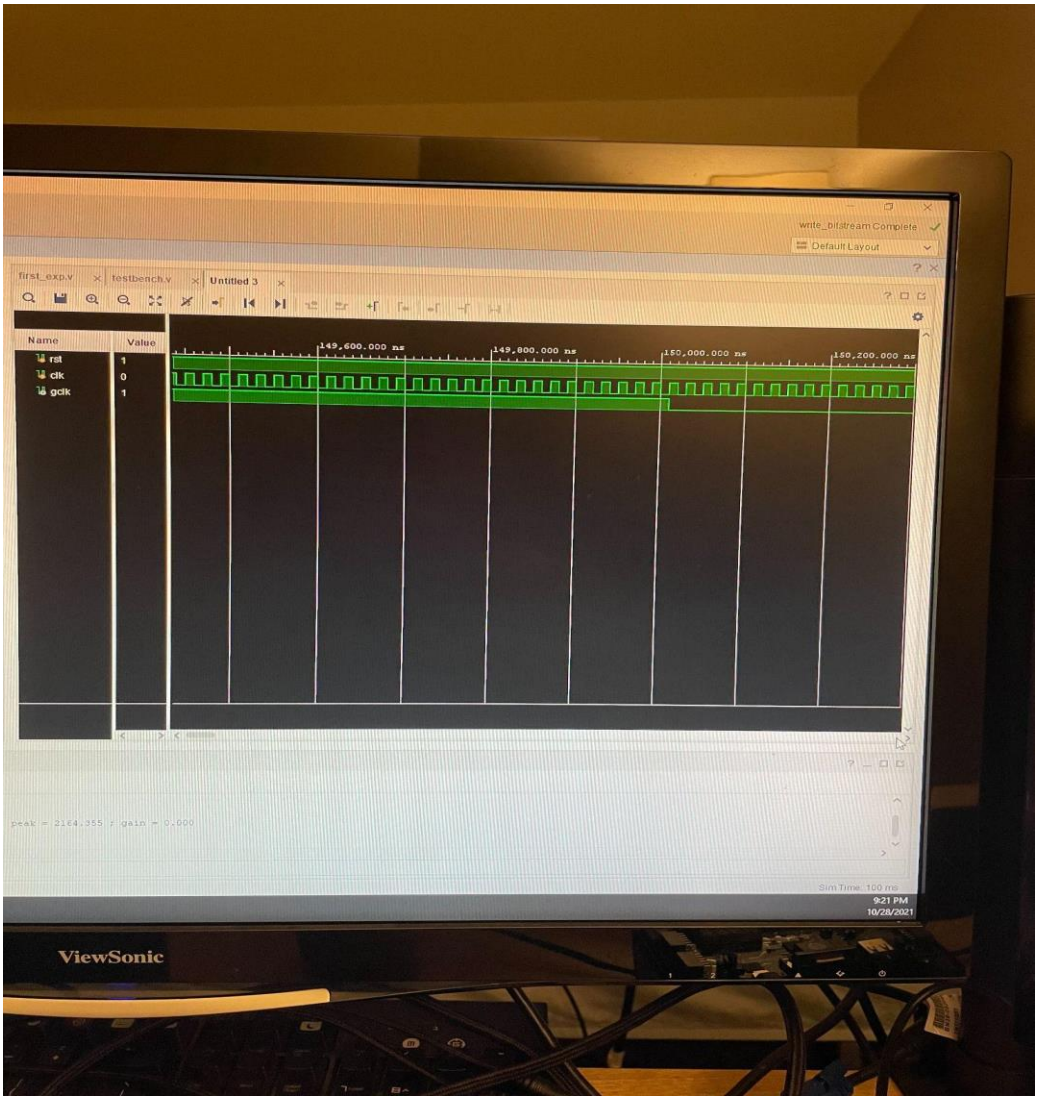

Figure 2: Simulated bench displaying "clk' and "gclk" waveforms.

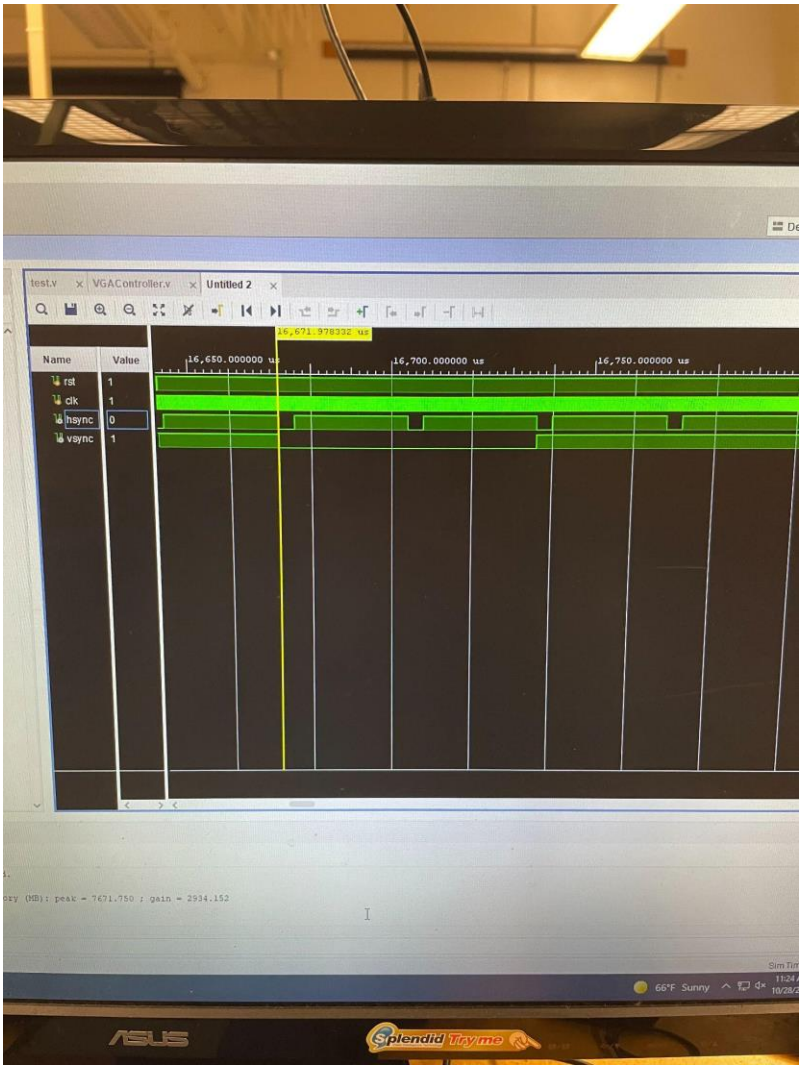

Figure 3: Simulation test bench of pixel clock(2nd waveform), hsync(3rd waveform), and vsync(4th waveform).

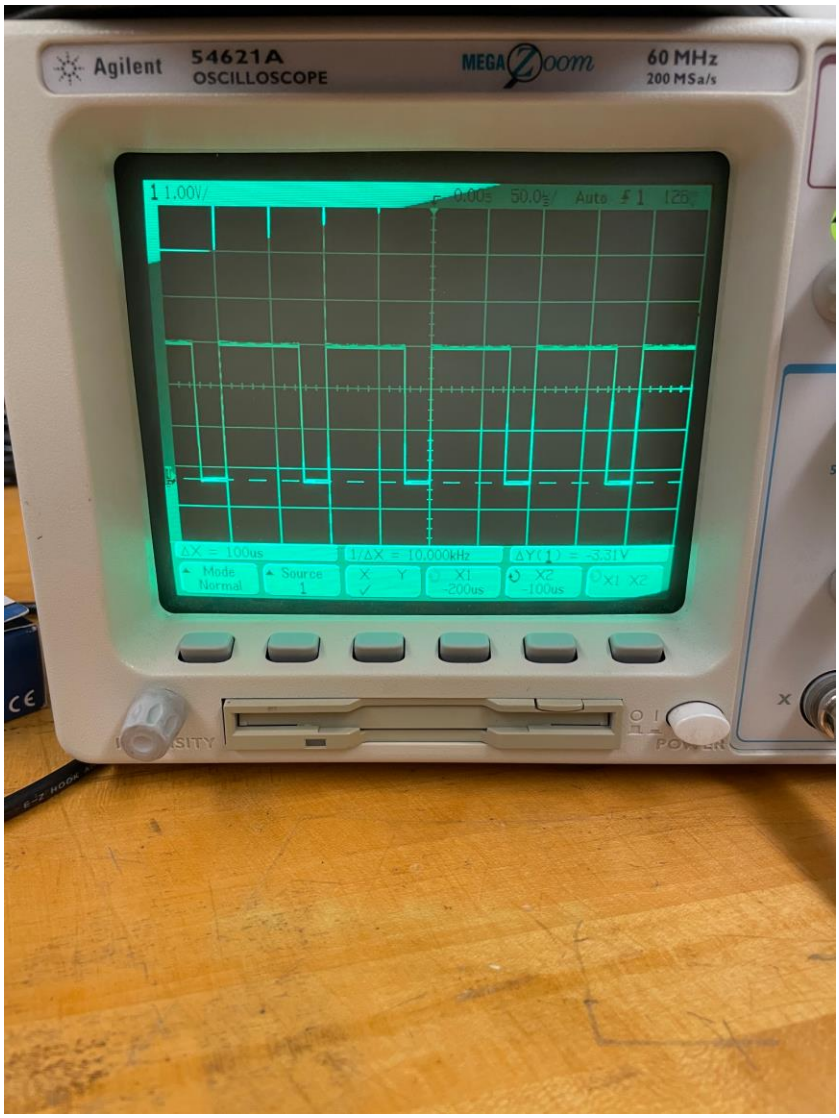

Figure 4: Oscilloscope displaying the waveform generated from "gclk" counter.

Experiment 2: For our second experiment, we developed three separate counters, the pixel clock, horizontal (hsync) and vertical (vsync) signals. Using the counter from experiment 1 as a template, we created the hsync and vsync counters. Hsync being active high from 0-799 counts (0 counting as 1) on the positive edge of the clock and active low from 96 counts, vsync went high from 0-520 counts (0 counting as 1) triggering on the negative edge of the hsync and low for 2 counts. We ran a simulation test bench **(Figure 5)** and loaded the verilog on the FPGA to display on the oscilloscope **(Figure 6).**

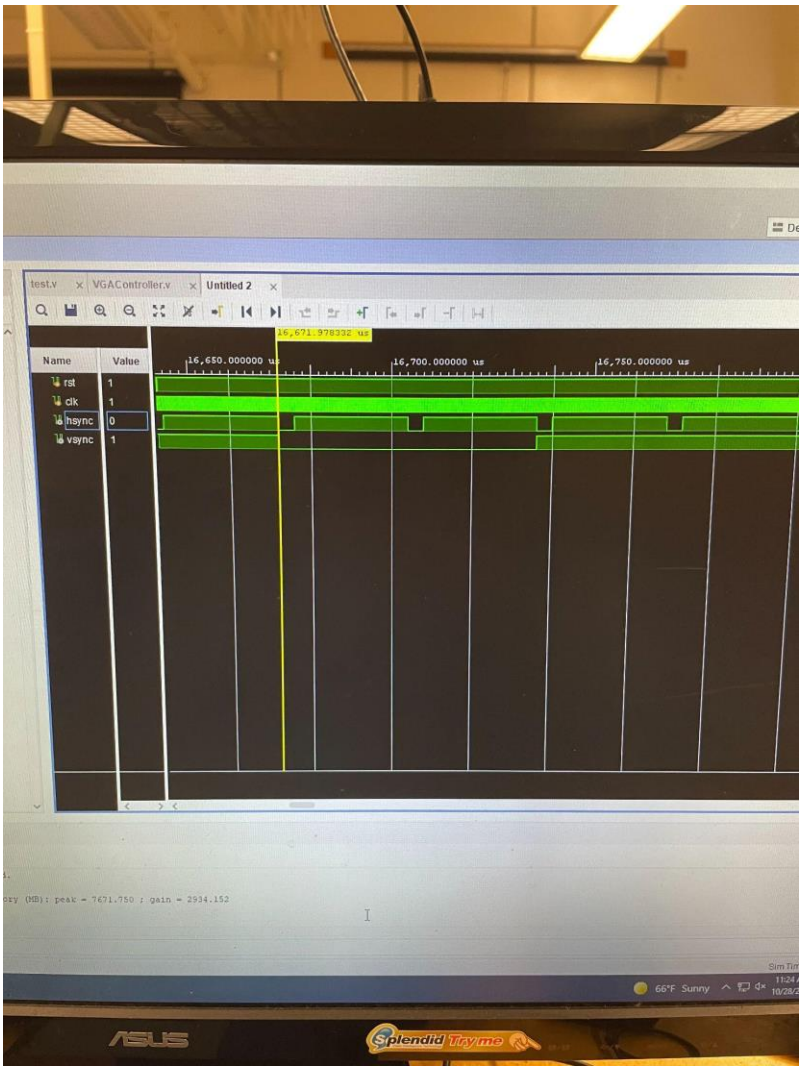

Figure 5: Simulation test bench of pixel clock(2nd waveform), hsync(3rd waveform), and vsync(4th waveform).

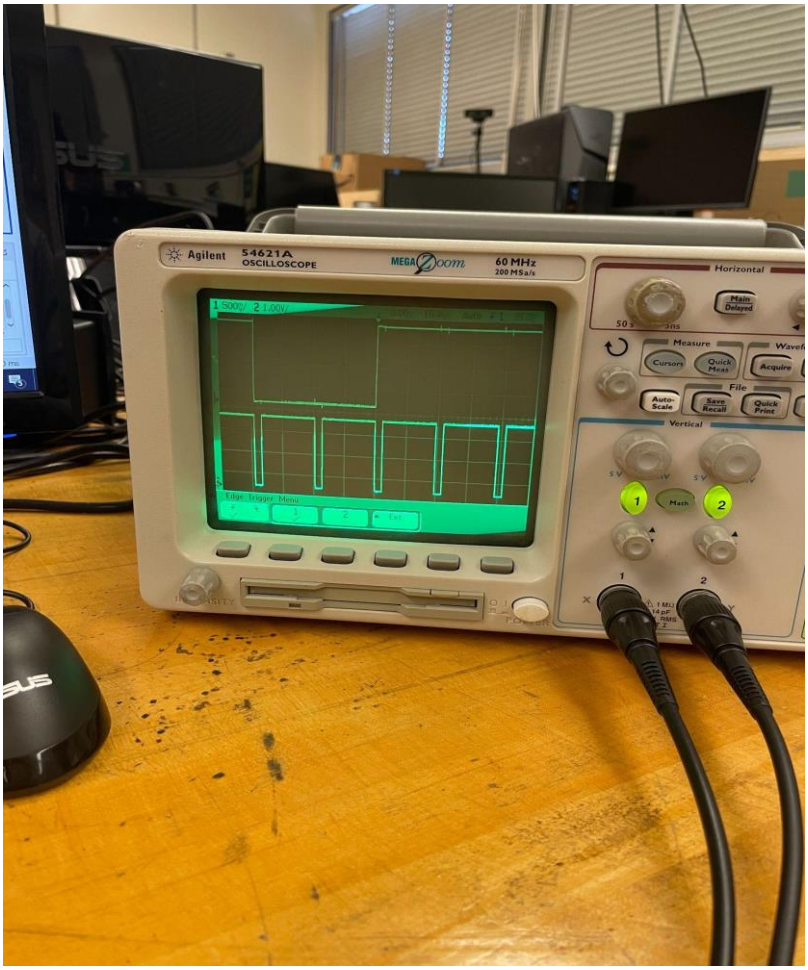

Figure 6: Oscilloscope displaying the waveforms generated from "hysnc"(bottom) and "vsync"(top) counters.

Experiment 3: For our third experiment, we implemented red, green, and blue components to a VGA controller. Using three 4-bit registers, one for each color, and calculating the addressable video time in between the front and back porch on the hsync (600 pixels) and vsync signals (480 pixels). We then set the bits in the red, green, and blue registers to display color on the monitor. **(Figure 8)**.

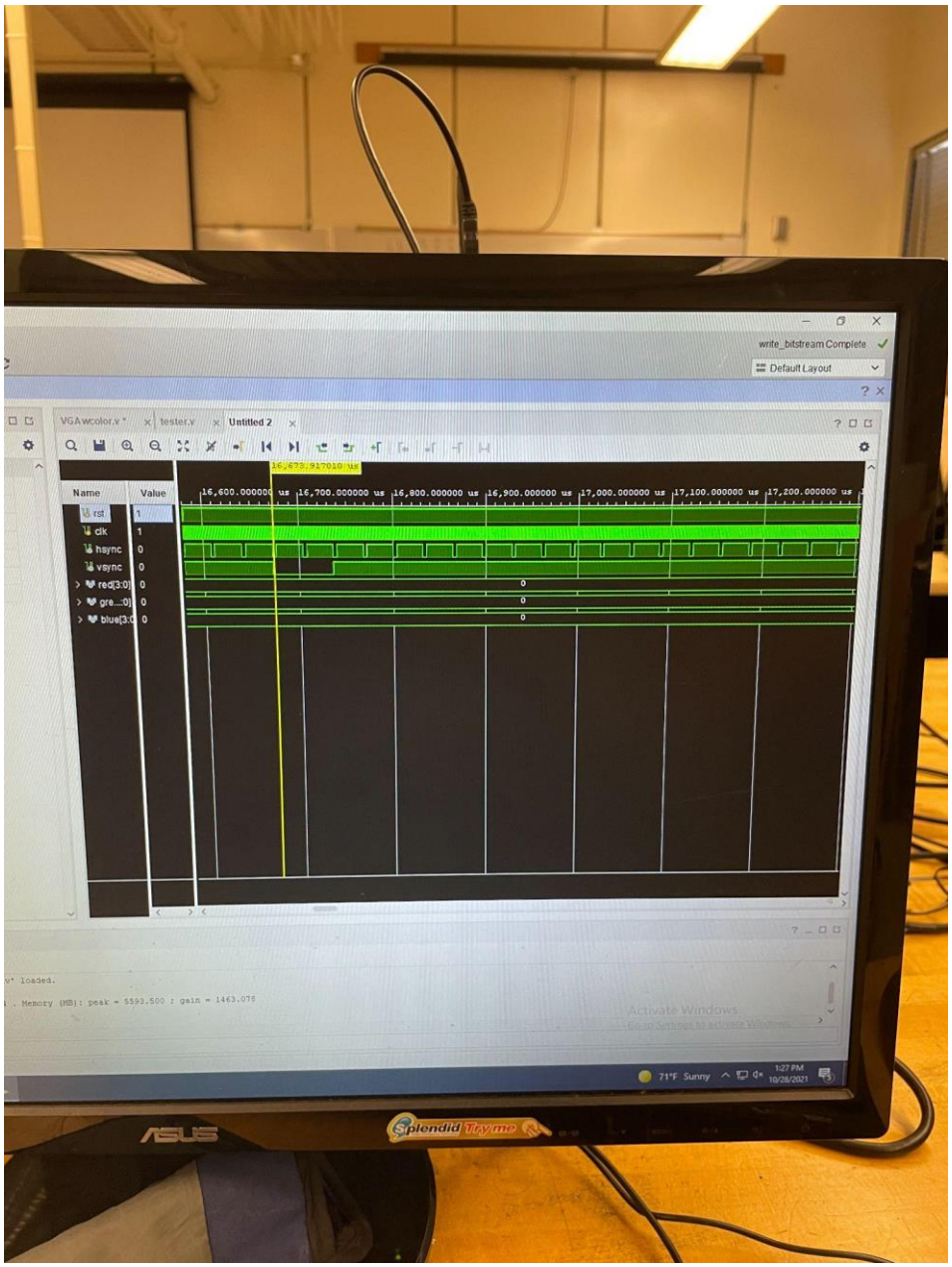

Figure 7: Simulation test bench for experiment 3.

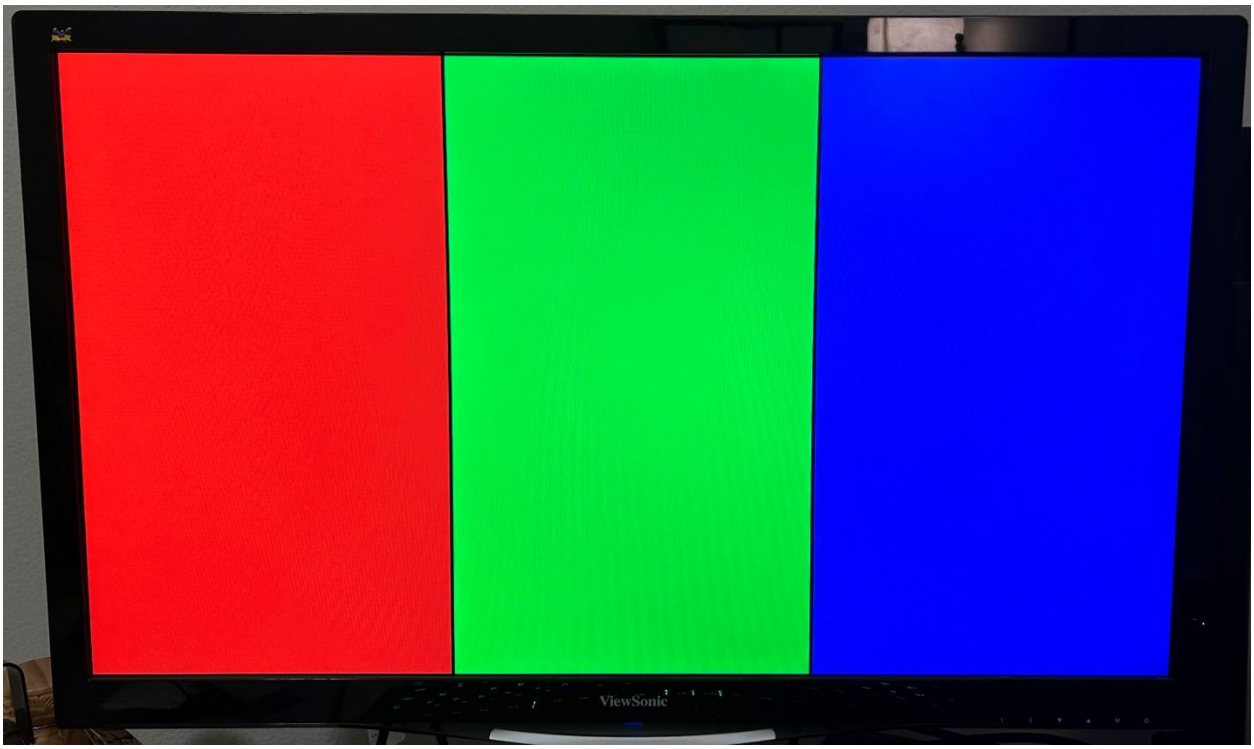

Figure 8: Red, Green, And Blue components on Monitor.

5| Discussion

The results we obtained from the lab were what we expected as discussed in the previous section.

Exp 1:
For experiment 1, with the given code which acted as a clock divider we were able to use the BASYS 3 connected to an oscilloscope to simulate our signal. The clock frequency we measure on the oscilloscope is 10.0kHz and the pulse width is 10ms.
Exp 2: in experiment 2 we needed to generate two signals at different frequencies. The first signal had to run at 31.25 kHz(hsync) and the second signal had to be at 59.98 Hz(vsync). In figure 6 we can see the two signals being displayed; the first signal being displayed is our vsync and the second one is hsync. We can compare the oscilloscope results with our simulation in verilog. In figure 5, we can see how our vsync and hsync signals run pretty similar as in the oscilloscope.
Exp 3:
In this experiment, we will use the VGA controller on our BASYS 3 to add red, blue and green components. In **figure 8** we show the colors on the screen using the VGA connection. Unfortunately, we were not able to get the colors for the flag of France. However, we were still able to create an RGB image on screen.

6| Conclusion

In conclusion, our group expanded our knowledge on working with the FPGA and its capabilities. This experiment helped us to understand how computers actually follow patterns, directions and logic implementation in order to come up with something on the screen.