Lab 4 : VGA

EECE5117C

Team 3

**Purpose**

The purpose of this lab was to understand how use VGA ports with FPGA boards. To do this, used the board to display a background pattern by editing the pixels of the monitor.

**Procedure**

We acquired necessary background knowledge from the slides provided for this lab on Blackboard. In order to use larger files like pictures, we made use of a python tool that converts a jpg picture to a mem file which can be fed into Vivado to be uploaded to the Board. The following steps were used to implement our code:

**Steps**

1. Design pseudo code that displays the correct pixels and colors to the monitor
2. Tailor the code to meet the dimensions of the monitor
3. Use the Python tool to convert our jpg to mem file
4. Configure code to read the mem file and draw shapes and pixels
5. Test on board

**Test Bench Code**

It is not particularly helpful to simulate a VGA display since you can’t see whether the display is correct or not, so we did not make a testbench or simulation for this lab and instead we put our testing directly onto the board.

**Test Vectors**

There are no inputs in this lab, so they only thing you can test is to turn the system on and see if the correct display shows up on the connected monitor.

**Problems**

There were a few problems we had in this lab. At first, we did not understand how to create images on the monitor using VGA, but we realized that you can draw to specific areas of the screen using assignments. We also found that adding in text was a challenge because you can’t just draw squares, you have to use some additional modules for it.

**Takeaways**

One of the takeaways from this lab is that VGA programming in Verilog requires precise control over the pixels. By mapping pixels to specific areas of the screen, displays can be created that are different shapes and colors. Another takeaway is that adding in more than just a few shapes requires special handling of memory. For example, to create a game requires that a lot of assets be stored in memory and sometimes stored in multiple memories that that the code can run efficiently.

**Team Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Program Design | Behavioral Code | Diagrams/Reports |
| Ben Cohen | x | x | x |
| Moaz Abougabal | x | X | x |
| Quinn Nye | x | x | x |