ECE 524 Lab: VGA with VHDL

Instructor: Saba Janamian

Spring 2023

Goals

1. To understand the VGA protocol and timing requirements.
2. To learn VHDL coding for designing digital circuits.
3. To design and implement a VGA controller using an FPGA board.
4. To verify the functionality of the VGA controller by displaying a simple image on a VGA monitor.
5. To learn how to use simulation tools to test and debug VHDL code.
6. To understand the importance of timing constraints and how to apply them in VHDL code.
7. To gain experience in using FPGA development tools and hardware description languages.

Grading Rubrics (Total = 100 points)

1. Pre-lab assignment (10 points)
2. Documentation, functionality and correctness (40 points)
3. Lab demonstration (50 points)

Required Hardware

- Zybo Z7-10/20 (Note: If you do not have the dev board you need prove your code is functional by analyzing your simulation waveforms)

Required Software

- Xilinx Vivado
- Git for Windows (If using Windows as development platform)
- VGA Pmod (https://digilent.com/shop/pmod-vga-video-graphics-array/)

NOTE:

This assignment is an individual work. Using other students' work or sharing your work with others is considered cheating and will result in a grade of 0 and possibility of reporting to the CSUN office of academic affairs. By submitting your assignment to Canvas you indicate the project is your original work, you have not shared this with anyone else and you accept responsibility for your action in case of any detection of academic dishonesty.

Pre-lab:

1. What does VGA stand for, and what is its purpose?

2. What are the different VGA resolutions, and what are their corresponding refresh rates?

3. What is the difference between VGA and HDMI?

4. What are the horizontal and vertical sync signals in VGA, and why are they important?

5. How many colors can VGA display, and how are they represented?

6. What are the timing requirements for VGA signals, and how are they specified?

7. How does a VGA controller generate a video signal, and what components are required?

8. What are the common issues that can occur when implementing a VGA interface, and how can they be mitigated?

9. What is the difference between a VGA monitor and a VGA graphics card?

10. How has VGA technology evolved over time, and what are some current alternatives to VGA?

VGA

Design a VGA controller using VHDL in an FPGA board. The VGA controller should be able to generate a simple image on a VGA monitor.

Requirements:

1. The system should detect the position of the switches and display a corresponding image on the screen.
2. When the switches are in the position 0000, the system should display a black screen.
3. When the switches are in the position 0001, the system should display a solid red screen.
4. When the switches are in the position 0010, the system should divide the screen into three regions and display RGB colors in each region.
5. When the switches are in the position 0011, the system should divide the monitor into 8 regions and display white, yellow, cyan, green, magenta, red, blue, and black colors in each region.
6. When the switches are in the position 0100, the system should divide the monitor into 8 sections and display 8 shades of gray in each section.
7. When the switches are in the position 0101, the system should allow the user to use a button to create different horizontal stripes on the screen.
8. When the switches are in the position 0110, the system should allow the user to use a button to create different vertical stripes on the screen.
9. When the switches are in the position 0111, the system should create a checkerboard pattern with different sizes.
10. When the switches are in the position 1000, the system should create a checkerboard pattern with an inner repeat pattern.
11. When the switches are in the position 1001, the system should display a moving bouncing ball on the screen. The ball should bounce off the edges of the screen.

.

Report

1. Create a video demo to show the successful installation of the tools. (See the requirements of creating video demo in the course syllabus)
    a. In your demo you could successfully start Vivado.
    b. Show your simulation code and simulation waveforms when running SystemVerilog testbench.
    c. Show your simulation code and simulation waveforms when running VHDL testbench.
    d. Show the elaborated result schematic.
    e. Show resource utilization in Vivado.
2. Write your lab report. Make sure to include the prelab exercise, write a step-by-step instruction on how to complete the exercise. Make sure to include waveforms from step 2 and 3. Include elaborated results and resource utilization in your report.
3. Push your src and sim folders to GitHub. Do not include any other Vivado auto generated files and folders. If you create your project in the build folder the .gitignore file will prevent the auto generated files from being pushed to GitHub.