Lab Report 2

Using Quartus Prime 18.1

Created by:

Group 18

Rushi Bhatt

Ansh Pachauri

Siwei Fan

ECE 2060 Lab (9605)

Wednesdays, 5:45p.m. - 8:45p.m.

The Ohio State University

Columbus, OH

16 February 2024

Executive Summary

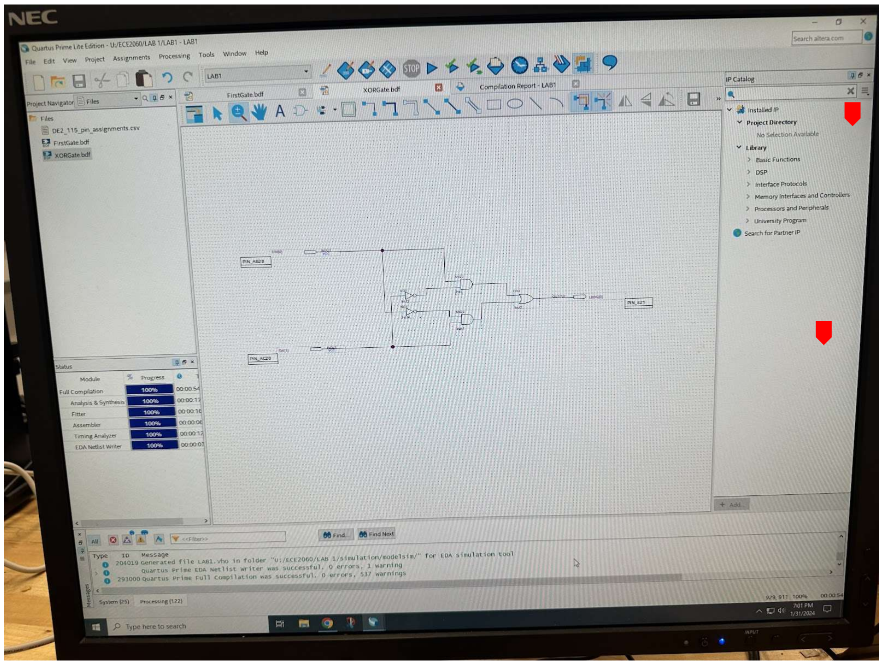
This lab report provides a comprehensive introduction to both ModelSim and VHDL, emphasizing their significance in digital circuit design and simulation. ModelSim is highlighted as a crucial tool for simulation and verification, enabling users to simulate digital circuits without physical hardware, which plays a crucial role in the design process. The lab focuses on setting up ModelSim for functional simulation, configuring simulation settings, and utilizing wave windows for visualization, aiming to provide a thorough understanding of ModelSim's functionality within Cordis Prime. In addition, the report delves into VHDL as a hardware description language, particularly focusing on combinatorial logic. VHDL is described as a language that compiles code into hardware rather than executing it like software. The structure of a VHDL file, including the entity and architecture sections, is explained, emphasizing the importance of adopting a hardware-oriented mindset when working with VHDL.
The experimental methodology section details the steps taken in the lab. By following this methodology, the results as shown in the pictures are being generated and the team gained practical experience in utilizing ModelSim and VHDL for design and simulation. Overall, this lab report provides a comprehensive overview of what the team did with ModelSim and VHDL including their importance, functionality, and practical applications.

Introduction

This lab mainly introduces ModelSim and VHDL by constructing simple projects through them. First of all, ModelSim is emphasized as a role for simulation and verification tools that are integrated by Cordis Prime. ModelSim enables users to simulate digital circuits without physical hardware, serving as a versatile platform for digital design and testing. Alongside other tools such as SignalTap 2 and the in-system memory editor, ModelSim plays a crucial role in the design process. This lab will mainly introduce how to set up ModelSim for functional simulation, configuring simulation settings, and utilize wave windows for visualization. Therefore, by conducting this lab, the team was able to provide a comprehensive understanding of ModelSim's importance and functionality within Cordis Prime. Secondly, VHDL is a hardware description language, focusing on asynchronous circuits, also known as combinatorial logic. VHDL stands for Very High Speed Integrated Circuit, and it is emphasized that VHDL code is compiled into hardware, rather than executed like software code. Therefore, it is important for the team to adopt the hardware-oriented mindset when working with VHDL. The structure of a VHDL includes two main sections: entity and architecture. In the entity section, pins are defined using port statements, allowing for the declaration of buses or vectors for multiple inputs or outputs. The architecture section defines the hardware, particularly combinatorial logic in the case of asynchronous circuits. Overall, this lab provides detailed introduction and fundamental insights towards ModelSim and VHDL and lays the groundwork for future applications.

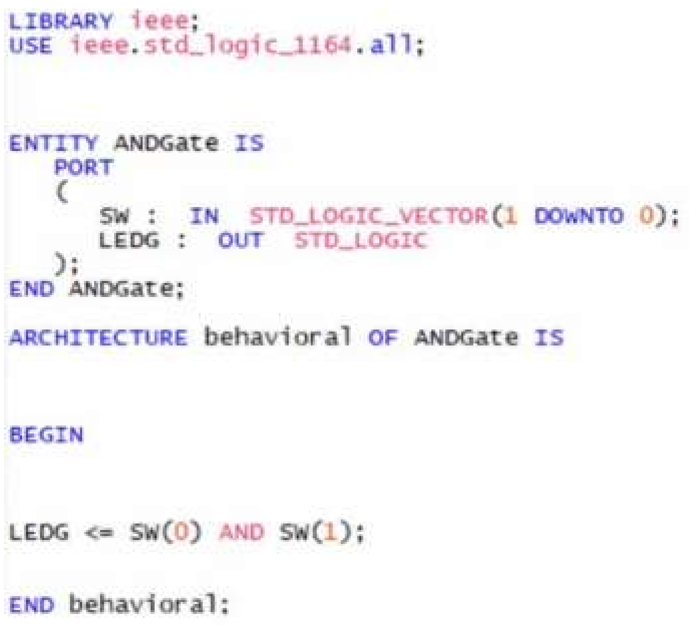
Experimental Methodology

In this lab, we utilized ModelSim to revisit the XOR gate schematic from Lab 1, setting it as the top-level entity. Following recompilation and Gate Level Simulation setup, we employed two clocks with different frequencies for input. To handle switches in a bus structure, we individually selected and configured them for the simulation. Maintaining the input and output names as instructed, we generated VHDL code for an AND gate and created a symbol file. Subsequently, we crafted a new .bdf file, integrated the AND gate symbol, and connected input and output pins accordingly. After setting the new .bdf file as the top-level entity, we compiled the project, configured waveform windows, and executed simulations in ModelSim, mirroring the methodology employed for the XOR gate. This comprehensive approach facilitated an exploration of gate logics, enhancing the understanding of the team towards the ModelSim and VHDL with the help of gate logics. Our XOR gate schematic and our AND gate VHDL code is shown below.



```
Type  ID     Message
      204019 Generated file LAB1.vho in folder "U:/ECE2060/LAB 1/simulation/modelsim/" for EDA simulation tool
             Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
      293000 Quartus Prime Full Compilation was successful. 0 errors, 537 warnings
```

Figure 1 - XOR Gate Schematic



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;


ENTITY ANDGate IS
   PORT
   (
      SW :  IN  STD_LOGIC_VECTOR(1 DOWNTO 0);
      LEDG :  OUT  STD_LOGIC
   );
END ANDGate;

ARCHITECTURE behavioral OF ANDGate IS


BEGIN


LEDG <= SW(0) AND SW(1);

END behavioral;
```

Figure 2 - AND Gate VHDL code

Results

Upon conducting and running the schematics and VHDL code through ModeSim, Group 18 acquired various simulation results which in turn verified the correctness of the schematic and the VHDL code. The schematic simulated a XOR gate whose results are shown in Figure 3. The VHDL code represents an AND Gate which takes two inputs and outputs the logic and operation of the two whose results are shown in Figure 4. The truth value of both of these simulations were tested through a truth table which identified all outputs given all possible combinations of inputs.

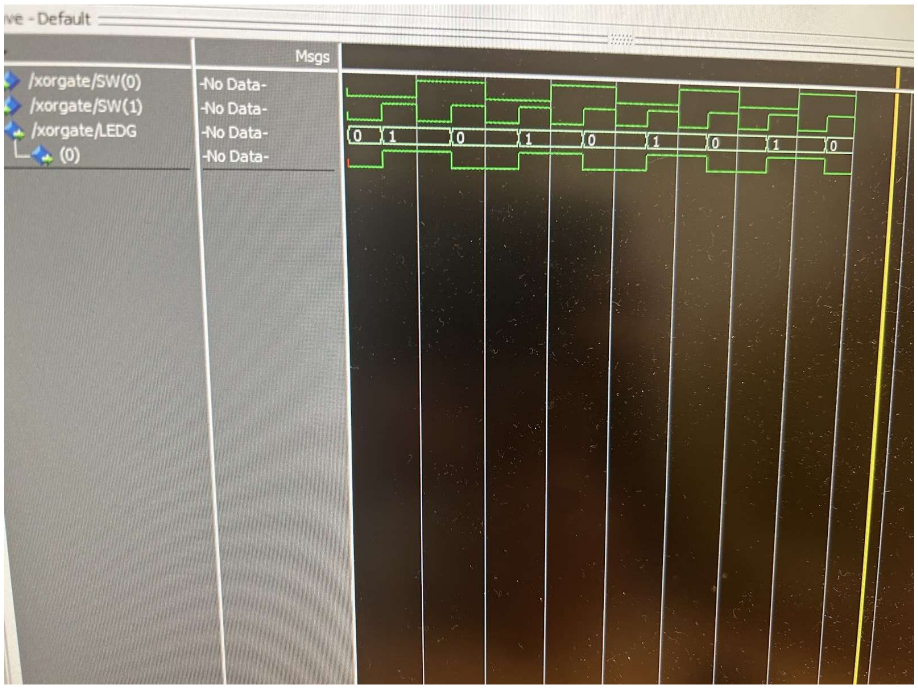
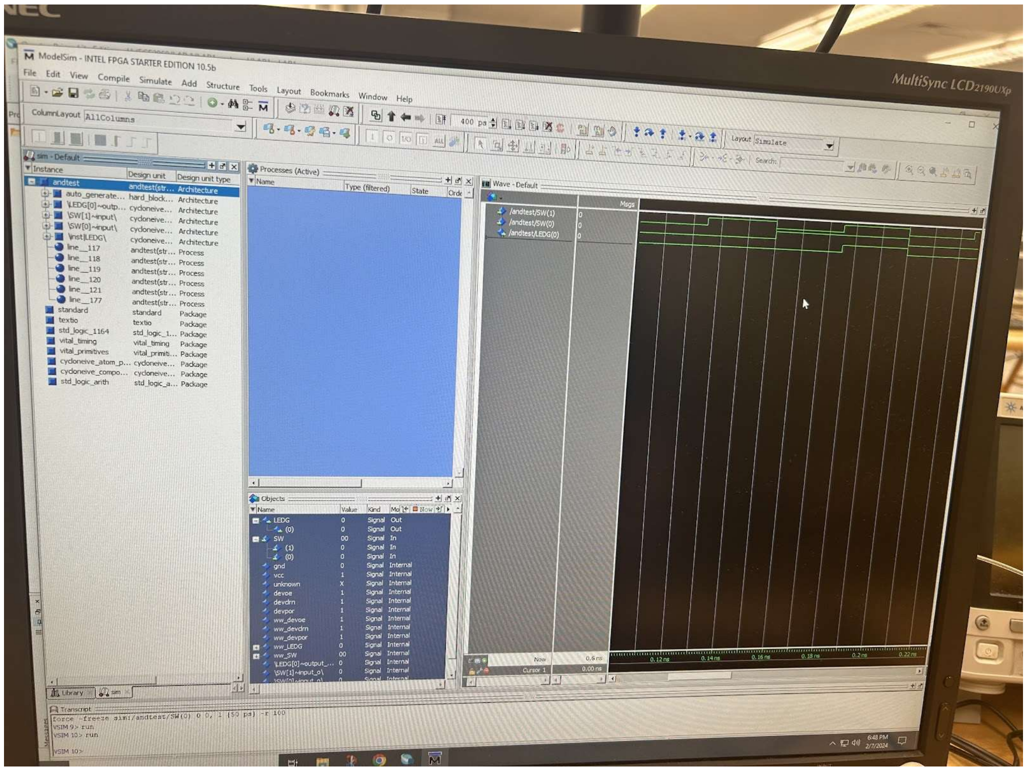


Figure 3: XOR Gate Simulation


ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help
Wave - Default
/andtest/SW(1)
/andtest/SW(0)
/andtest/LEDG(0)
Objects
Transcript
MultiSync LCD2190UXp

```
force -freeze sim:/andtest/SW(0) 0 0, 1 {50 ps} -r 100
VSIM 9> run
VSIM 10> run

VSIM 10>
```

Figure 4: AND Gate Simulation

Discussion

In digital logic, a functional simulation primarily focuses on ensuring the logical functionality of the circuit design, while a timing simulation's main purpose is to evaluate the behavior of the design while considering the delays and timing constraints. Both of them serve different purposes and are used in different parts of analysis to error-check circuit designs. In order to successfully bring up a wave window in ModelSim you need to first compile your code, then find the "tools" button, and run the simulation as a gate level simulation. When you prompt it to run as a gate level simulation, the ModelSim window will pop up. Next, you will need to click "view" at the top and scroll down to press "wave". Doing this will display your timing diagram in a wave form setting. Moving onto a VHDL file, the first section is known as the entity section, which defines the file name and all of the pins within that section. The second section is the architecture section, whose function is to define the hardware or combinatorial logic. The architecture section can be named anything, opposed to the entity section. Lastly, the primary difference between a programming language and a hardware language is that a hardware language does not execute; the compiler converts the code into hardware, which means there is no top down flow to the code. In programming languages, the code that you write for software allows for executing the code in a top down flow manner.

Conclusion

During Lab 2, Group 18 engaged in digital design exercises to enhance their proficiency in the field. They created schematics for an AND gate and an XOR gate using Quartus Prime, subsequently compiling and designating them as top entities. In ModelSim, they generated timed clock diagrams to verify the functionality against provided truth tables, ensuring accuracy. Furthermore, Group 18 undertook VHDL coding to regulate the behavior of the AND gate, facilitating more accurate predictions for various input combinations. By the end of Lab 2, each person in Group 18 had learned some new skills and a deeper understanding of digital design principles, positioning them effectively for future endeavors in the field.

Acknowledgments

All members of Group 18 actively contributed and participated equally in all lab activities. Each team member possesses a thorough understanding of the procedures undertaken and the results obtained during Lab 1.

References

- Lab report template - reportTemplate