ECE3270 Digital Computer Design: Lab 2

OPENCL LIBRARIES - 16-BIT COMPARATOR

February 23, 2019

Aayahna Herbert
Clemson University
Department of Electrical and Computer Engineering
aayahnh@g.clemson.edu

Abstract

The goal of this lab was to ensure the student could successfully use OpenCL and familiarize themselves with the concept of Libraries for OpenCL. This design lab was split into two parts: (1) designing a 16-bit comparator and (2) compiling and and executing the device in OpenCL. In this lab, all parts of the lab were successfully designed, tested, and executed in OpenCL and through simulation done in ModelSim.

1 Introduction

A comparator is a type of arithmetic circuit that compares the relative sizes of two binary numbers, A and B. Unlike the comparator built for this lab, most comparators produce three outputs that go to high depending on if A is less than B, equal to B, or greater than B. Because this lab is testing our ability to work with OpenCL, the circuit was simplified down to one output that checks if A is equal to B. The most important part of this lab is making sure the comparator is functioning correctly in VHDL mainly because the compiling process for OpenCL can be time-consuming and tedious to have to constantly update due to simple mistakes that could be detected by a good simulated test. The next sections will go into details about how each part of the lab was constructed and the reason it was done the way it was.

2 16-Bit Comparator

2.1 Design Method

The design of the 16-bit comparator relied on two components, a fragment of one comparator and the NAND gate operation. For the fragmented component of the comparator, it was designed to represent the logic gates that produce the output that is used for the NAND gate's input. From looking at Figure 6.1, that fragment consists of two inverted inputs going into an XNOR gate. The design for this fragment consisted of an entity and its data flow architecture. The entity consisted of two single bit inputs, A and B, and a single bit output, C. The architecture for the entity was made to be a data flow since the logic diagram for it was very simple and straightforward to be converted into a SOP equation.

The other component used was for the NAND gate section of the comparator. The design for this component consisted of a generic entity and its behavioral architecture. This component was provided by the TA and was made generic as a way to make its design easily changeable in the bit size/width of the NAND gate. Because this lab is wanting a 16-bit comparator, the generic integer WIDTH was set to 16. The entity consisted of a 16-bit input, a, and an output called output. The behavioral architecture was a for loop that went through the input and AND gated input a bit by bit and then inverted the final result to create the complete NAND gate.

The design for the 16-bit comparator consisted of an entity with two 16-bit inputs, A16 and B16, and an output O16 and a structural architecture. The architecture used the comparator fragment and the NAND gate components and mapped the inputs A16 and B16 to input A and B, created a 16-bit signal C16 to map to output C that was then mapped as the input for the NAND gate while O16 was mapped to the NAND gate component's output. Because the first component would require 16 separate port maps to create the 16-bit comparator, a generate statement, as explained in Chapter 6.6.4 of Brown [1], was used to basically create loop that would create 16 port maps while only typing it all up on one line, making the code shorter and more efficient.

2.2 Testing and Simulation

When it came to simulating the 16-bit comparator design with a test bench, the test cases chosen, as shown in Figure 6.2, were meant to test cases where the inputs were equal, slightly off from one another, and complete opposites of one another - 2's complement was used. When simulated in ModelSim, the wave forms, as displayed in Figure 6.3, reflected the expected results from the test bench, making it successful.

3 OpenCL

3.1 Design Method

In order the 16-bit comparator to function in OpenCL, a VHD file was provided that laid out the entity of the OpenCL comparator while leaving the structural architecture empty to be completed. As instructed in the README.txt file provided to the students for this entity, VHDL version of the 16-bit comparator was used as the component for this entity. Using a port map, the first 16 bits of datain were mapped to A16, the remaining 16 bits of datain were mapped to B16, the least significant bit of dataout was mapped to O16, while the remaining bits of dataout were set to zero. After this was done, it was compiled to make sure there were errors before moving into the compiling stage in OpenCL.

3.2 Testing and Execution

Following the steps and procedures that were laid out in Smith [2], the provided XML file was updated to include the VHD files created for the comparator fragment entity, the NAND gate entity, and the 16-bit comparator entity. Afterwards the device and host codes were compiled successfully and their corresponding executables were made. These files were then copied onto the SD card - which was then inserted into the Altera board - and tested. The execution was successful.

4 Results

When both parts of the lab were modified to be tested on the Altera DE1-SoC Board using guidance from Smith [2], they compiled and functioned correctly. The common issue I ran into - outside of simple syntax issues - when working on the lab was the ModelSim not being able to compile as a result of me not using the correct name for the input variables. When the final part was tested on the board with all of the other parts combined with it, the terminal said the output was valid, meaning it was a success.

5 Conclusion

This lab has given me the chance to test and apply my knowledge about OpenCL while improving my skills and knowledge on VHDL coding, behavioral and entity elements and techniques, learned from class lectures. By doing this lab, I have a clearer understanding

on how each of the components made in this lab work and what they all need in order to do their jobs correctly. If I could go back and change how things were done throughout the lab, I would change a couple of lines of code to be written in a way that makes the VHD file shorter and easier to read. Overall, this was a great lab to get used to OpenCl.

References

[1] S. Brown and Z. Vranesic. Fundamental of digital logic with vhdl design. Textbook, New York, USA, 2009.

[2] M. Smith. Digital computer design lab manual. Manual, South Carolina, USA, 2019.

6 Appendix

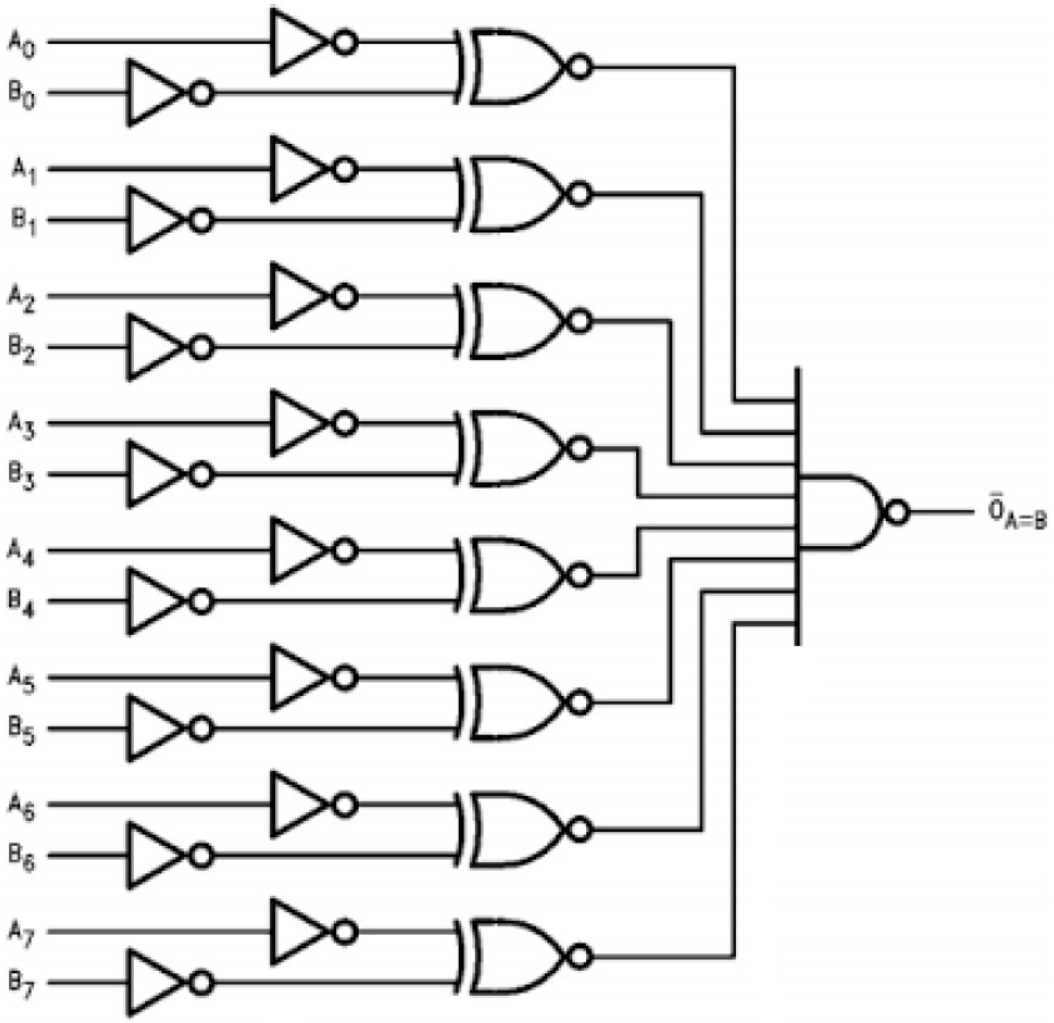

Figure 6.1: 8-Bit Comparator Logic Diagram

```
BEGIN
        -- code executes for every event on sensitivity list
   A16 <= "1111000011110000";
   B16 <= "1111000011110000";
   WAIT FOR 10ns;                    --A16 AND B16 exactly the same; expecting "not 1" or '0'

   A16 <= "1010001111100101";
   B16 <= "1010001011100101";
   WAIT FOR 10ns;                    --A16 AND B16 slightly different; expecting "not 0" or '1'

   A16 <= "1101011010110001";
   B16 <= "0010100101001111";
   WAIT FOR 10ns;                    --A16 AND B16 two's complement of each other; expecting "not 0" or '1'

WAIT;
END PROCESS always;
END comparator16_arch;
```

Figure 6.2: 16-Bit Comparator Test Bench Cases

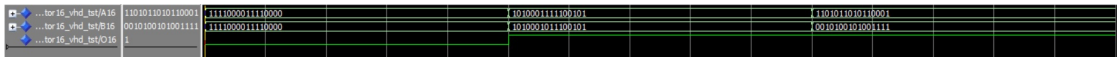

Figure 6.3: 16-Bit Comparator Simulated Test Waves