Experiment Instruction for Digital Logic Circuit with Verilog

By Xiaoping Huang

School of Computer Science

Northwestern Polytechnical University

2018/09/20

Content

Task 1 Basic Practice of Verilog

1.1 Purpose

1: Learn the basic point of Verilog

2: Practice how to design the testbench and model stimulus

3: Practice how to perform simulation with Modelsim

4: Design the basic combinational circuit module using Verilog

1.2 Procedure

1: understand the function behavior of the MUX4to1

2: design a MUX4to1 with Verilog. The specific requirement is the width of all data input is 4.

3: deign the testbech file of the MUX4to1 using Verilog

4: Learn how to create a project under the Modelsim Environment

5: Perform the MUX4to1 simulation using Modelsim

6: Try to fix the errors you encounter.

7: design a Decoder3to8 using Verilog and perform the simulation, repeatedly perform the flow from step1 to step6 like MUX4to1.

8: design a priority Encoder8to3 using Verilog and perform the

simulation, repeatedly perform the flow from step1 to step6 like MUX4to1.

9: design a BinarytoLED converter using Verilog and perform the simulation, repeatedly perform the flow from step1 to step6 like MUX4to1.

Task 2 Addition Circuit Module

2.1 Purpose

1: Implement the 8-bit CLA architecture addition using Verilog

2: Continue to learn the Modelsim tool

3: Learn how to implement a module in FPGA

2.2 Procedure

1: Understand the work principle of CLA adder

2: design an 8-bit CLA adder using Verilog.

3: Design the testbench and try to implement an auto-compare testbench

4: perform the simulation using Modelsim

5: try to fix the appeared bug

6: read the datasheet of the FPGA board and understand how to use the board to verify or realize your design.

7: learn how to use Quartus II tool to implement your design on FPGA demo Board

8: Using the hardware resource in the FPGA board, such as

swith,LEDs, Implement decoder or encoder in the FPGA board. You could use your source code in task 1. In the task, switcher in the FPGA board can act as input signal; LED can act as output signal.

Task 3 Sequential Circuit Module and Implementation

3.1 Purpose

1: model basic sequential circuit module using Verilog

2: continue to learn how to implement the design on FPGA

3: continue to learn the design method of testbench

3.2 Procedure

1: Design an 8-bit registers with asynchronous reset and perform the simulation

2: Design a shift register with asynchronous reset and perform the simulation. Note: The depth of the shifter register is 5 and the width of each stage is 8.

3: Design an 8-bit counter with enable port and load port, then try to model the behavior of all input ports and perform the simulation. The ‘enable’ port means PAUSE function. That is when ‘enable’ is high, the counter will start to count and otherwise ,it will hold the number. When

the ‘load’ signal is valid, the ‘Data’ signal will be load into the counter.

4: Using the modules you have designed in previous task, implement a basic ALU (arithmetic logic unit). The ALU has the ability of adding, shifting, counting and Logic operation. Using the MUX4to1, the results can be selected and then connected into the LEDs to display the result. You may need to define several input signals to switch the functions.

5: Implement the ALU design into the FPGA board to verify the functions

6: you need to compose a report about the ALU design based on the report template. The electronic version should be sent to huangxp@nwpu.edu.cn and the printed version also should be submitted. The deadline is by the end of the experiment course. The index of this report is A.

Task 4 Comprehensive Design

4.1 Purpose

1: learn how to design a state machine with Verilog.

2: design the reaction time detector in Chapter 5 of your text book

3: Implement the reaction time detector on FPGA board

4.2 Procedure

1: design a state machine to detect the sequence pattern “101100”.

2: design the testbench to verify your state machine design

3: design the reaction time detector. You may refer to your text book to understand the SPEC of the reaction time detector.

4: implement and debug your design in the FPGA board.

4.1 you may adjust the frequency of clock to control the counter’s speed

4.2 You may use one level-switcher to act as the ‘w’ signal

4.3 you may add additional delay to postpone the change of LED’s state once ‘w’ is valid. This will allow the participant to prepare for the testing.

5: you need to compose a report about the reaction detector design based on the report template. The electronic version should be sent to huangxp@nwpu.edu.cn and the printed version also should be submitted. The deadline is by the end of the experiment course. The index of this report is B.