University of Cincinnati
Department of Electrical Engineering and Computing Systems
EECE 2060C – Digital Design ELTN 1040C – Digital Systems

Lab 5: ISE Installation and Practices for Verilog and FPGA

Fall 2017

Introduction

The purposes of this lab are to: (a) Install ISE 14.7 into your laptop that runs Windows 8.1 or 10; (2) Practice Verilog and ISE by implementing a simple module; (3) Download the bitstream (configuration) file into FPGA board BASYS2 for execution.

Task 1 – ISE 14.7 installation

Follow the first tutorial, **ISE and Adept Installation Guide (FPGA Tutorial 1)**, posted under **Course Documents/Labs/Verilog and FPGA Lab Orientation** to download ISE version 14.7 and Adept 2.3. Note that the web sites of Xilinx (for ISE) and Digilent (for Adept) have been greatly changed. However, you should not have difficulty in finding the links to download. Before downloading ISE, you have to register an account at Xilinx web site. ISE installation file is about 6.2G. It takes very long time (several hrs) to download during lab hours. Please try your best to download and install ISE 14.7 on your laptop before the lab session. Adept is a much smaller software, but it is also recommended that you download it before the lab. You do not need to register an account to download Adept. Use the following links for the tools, if you cannot find them:
http://store.digilentinc.com/digilent-adept-2-download-only (Adept) and
https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html (ISE)

Follow the procedures in **Guide to install ISE 14.7 to Windows 8.1 and 10 (FPGA Tutorial: Additional steps for Win 8.1/10 users)** to make ISE 14.7 compatible with Windows 8.1 and 10. Without this repairing, ISE will crash whenever it is executed. Thus, we have **step 1**: Download ISE 14.7; **step 2**: Run "xsetup" and let it install; **step 3**: Follow the repairing tutorial to fix bugs.

In summary, **ISE is used to** enter your Verilog codes and testbench, run simulation, synthesize the logic circuit, assign the pins for inputs and outputs, and generate the bit file that is used to configure the FPGA board. **Adept is used** to download the bit file from your laptop to the FPGA board where the FPGA chip can be configured into your logic circuit and allow you to use switches, push buttons, LEDs, and seven segments to operate the circuit.

Task 2 – Verilog programming, simulation, and synthesis

There is a great **Verilog Tutorial (FPGA Tutorial 2)** under the directory of Verilog and FPGA lab

Orientation to help you go through the entire design entry, Verilog syntax checking, logic simulation, pin assignment, logic synthesis, and bit file generation process. Follow the tutorial step by step will work out this lab.

First, under design view of **Implementation,** use ISE to **enter** the following module written by Verilog:

```
module test1(A, B, C, D, E);
input A, B, C:
output D, E;

assign D = (A && B) || (!C);
assign E = (!C);

endmodule
```

Then, under design view of **Simulation**, use ISE to **enter** the following testbench for logic simulation.

```
module sim;

        // Inputs
        reg A, B, C;

        // Outputs
        wire D, E;

        // Instantiate the Unit Under Test (UUT)
        test1 uut (A, B, C, D, E);

        initial begin
                // Initialize Inputs
                A = 0;
                B = 0;
                C = 0;


                // Wait 100 ns for global reset to finish
                #100;
                // Add stimulus here
                A = 1;
                B = 0;
                C = 1;
                #200
                A = 0;
                B = 1;
                C = 0;
              end

    endmodule
```

Third, under design view of **Simulation**, perform **Behavioral Check Syntax** and then **Simulate Behavioral Model.**

Show the TA the simulation waveform and make sure the output correctness. Have the TA sign the waveform below and take a photo of the waveform to be included in the final report.

TA: _____ observed and verified the circuit waveform here.

After the simulation waveform demonstrates the design correctness, use ucf (implementation constraint) file to specify (inputs / outputs) of your circuits into I/O devices (switches or push buttons / LEDs or seven-segment-displays) on the FPGA board. Enter the following **pin assignments** to fill up your ucf file.

```
NET "A" LOC = "N3";
NET "B" LOC = "E2";
NET "C" LOC = "F3";

NET "D" LOC = "M11";
NET "E" LOC = "M5";
```

Once the pin assignment have been finished, we are ready to **synthesize** the logic circuit. Pay special attention to the **Synthesis Final Report** that contains Final Results, Device Utilization Summary, and Timing REPORT. Include the Final Report in the lab final report. Finally, use **Implement Design** and **Create Programming File** to generate the bit file. You have finished the job by ISE and you are ready to use Adept to take over the next step.

Task 3: Bitstream (configuration) file generation, downloading and FPGA execution

Follow the **Run ISE and FPGA board tutorial (FPGA Tutorial 2)** to download the bitstream configuration file generated by ISE to the BASYS 2 FPGA board.

If you **get an error** while downloading the .bit file to your FPGA board that says "Clock is not named properly or Clock is named 'CCLK' instead of 'JTAG Clock' ", this means that the clock is not named in the standard format in the programming file (.bit file that is generated by ISE).

The FPGA board still works with this with this error. However, you can follow the steps given below to resolve this error.

1. On ISE, under implementation (Task 2), right click on "Generate Programming File".

2. Go to "Process properties".

3. Under "start up options", look at the property "FPGA start up clock". Change this value to "JTAG Clock" in the drop down menu.

4. Now that the clock is named correctly, rerun the "Generate Programming File", and generate the correct programming file.

5. Download this new .bit file on your FPGA.

Once the FPGA board configuration is done, design at least 5 input patterns to fully test the circuit. The circuit testing table is given below.

Circuit Testing:

Inputs			Expected outputs		Outputs observed on the FPGA board	
A	B	C	D	E	D	E

Show the circuit outputs to the TA who will sign below

TA _____ observed and verified the above table.

Pre-Lab Design Report

Not required (because the design has been provided as shown in module test1).

Final Report Due: Just follow the policy of one-week window.

Project Final Report (PFR) Format:

The PFR is computer-edited report that documents all aspects of the completed tested project.

Title page

Section number (at least 5% of the final grade)

The PFR must contain the following:

1. Purposes of this lab
2. Verilog code of the module.
3. Testbench of the module.
4. Testbench simulation results, waveform photo for one or two test patterns (Task 2).

5. Synthesis final report produced by ISE (Task 2).
6. FPGA board test results, FPGA photo (Task 3).
7. Attach the pages (above) that contains TA's signatures for waveform and FPGA board testing.
8. Define in detail the contribution of each team member to the accomplishment of the project for each task. The idea is that over the quarter all team members share equally in all aspects of the laboratory activity (requirements, design, simulation, test, implementation, and report writing).
9. Report the time used for each task.