ECE 2029 INTRODUCTION TO DIGITAL CIRCUIT DESIGN

Lab 2: Introduction to Xilinx Vivado Design Environment

Sign-Off Sheet

Date: 4/15/2020

Student 1: Jonathan Lopez

Student 2: Margaret Earnest

YOU ARE RESPONSIBLE TO COMPLETE ALL THE ASSIGNMENTS IN THE CHECK LIST BELOW IN ORDER TO GET FULL CREDIT

Check List	
Assignments	**TA Sign-off**
Pre-Lab (**MUST** be completed before the start of the lab) Download Vivado on your machine Watch the tutorial: https://youtu.be/tOwMmBI_XNo Create a project Design Entry & Synthesis	
Lab part Create and add a constraint file	
Synthesis & Simulation for Timing Verification	
Implementation and Downloading	
Programming the serial flash.	

Submission Details: (Each student should submit files individually on Canvas)

Project File(s) Upload on CANVAS

1. Upload Sign-off Sheet
2. Upload your project files (zip the folder)
   Name your file as: lastname_labn_D'20
3. Present your work
   Record a 2-3 minute video showing that all parts of your lab functioned properly.
   OR
   Show your work to TAs for the sign-off.
4. Upload the Writing Assignment (Debugging and Troubleshooting errors FAQ's) – OPTIONAL

Due Date: 04/22/2020

Pre-Lab

IMPORTANT: Watch the tutorial before you begin…

https://youtu.be/tOwMmBI_XNo 25:12 min

Usually, laboratory exercises will have pre-lab assignments which are to be completed before your lab session and must be signed-off by the TA during your lab session.

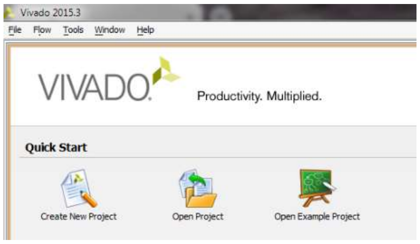
Pre-labs help you to become oriented to the problem before you enter lab, help complete your design in advance and prevent wasting time in lab. This second lab is a straightforward tutorial and does not have a pre-lab.

To complete this pre-lab, you will need to install the Vivado on your machine, the link to download the software is: https://www.xilinx.com/support/download.html

Assignment

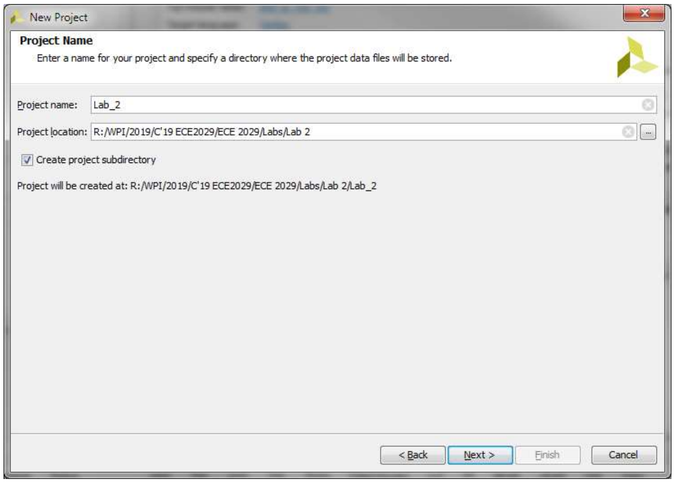
In this second lab you will start by implementing the basic logic gates AND, OR, NOT and the compound exclusive OR gate (XOR) and verify their truth tables. Then you will implement a multi-gate circuit. You will do this using the Verilog hardware description language.

1. Start Vivado Design Suite by clicking on Vivado icon and then selecting Create New Project from the menu. And click next to move to next step.



2. Enter a project name. You should use **your network drive** as the project location for your files. Make sure there are ***no spaces or special characters (- and _ are allowed)*** in the folder name that you create for your work. If you must create you project on the local disk be sure to back it up to your network drive when you're done at the end of lab. Lab machines can be re-imaged at any time and you could lose your work if it is not on your network drive! Each partner should have a copy of all the work!

Click *Next* and then select the RTL project type. Be sure to check the "Do not specify sources at this time" box and click Next:



Project Name

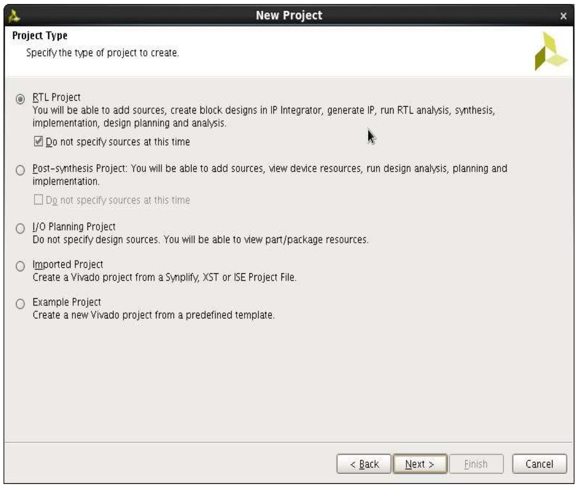
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: Lab_2

Project location: R:/WPI/2019/C'19 ECE2029/ECE 2029/Labs/Lab 2

☑ Create project subdirectory

Project will be created at: R:/WPI/2019/C'19 ECE2029/ECE 2029/Labs/Lab 2/Lab_2



Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☑ Do not specify sources at this time

○ Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.

□ Do not specify sources at this time

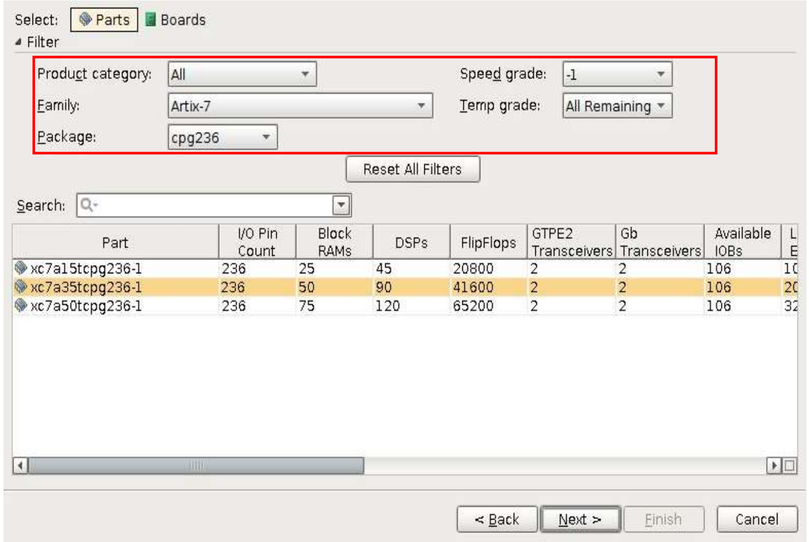
○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

○ Example Project
Create a new Vivado project from a predefined template.

Click next and select the correct Xilinx Artix 7 FPGA that is on the Basys3 board (XC7A35T-1CPG236C)

Click Next, and then Finish on the New Project Summary Page. The Project Window opens:



Select: Parts Boards

Filter

Product category: All

Speed grade: -1

Family: Artix-7

Temp grade: All Remaining

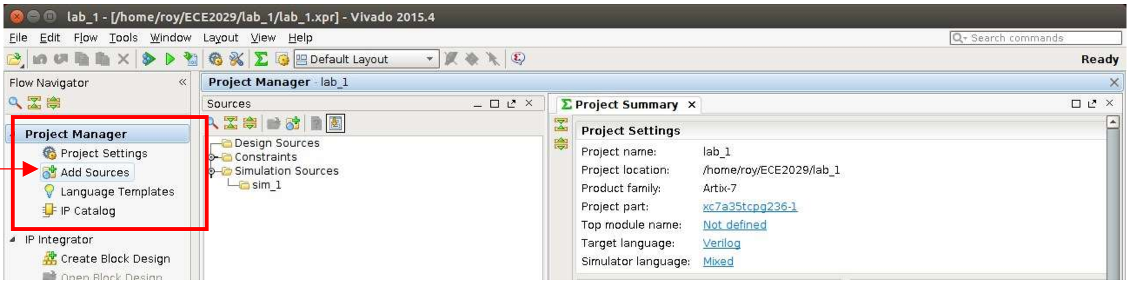
Package: cpg236

Search:

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	Gb Transceivers	Available IOBs	L E
xc7a15tcpg236-1	236	25	45	20800	2	2	106	1(
xc7a35tcpg236-1	236	50	90	41600	2	2	106	2(
xc7a50tcpg236-1	236	75	120	65200	2	2	106	32

Design Entry

1. Under the Project Manager, select Add Sources to begin designing your new project.



lab_1 - [/home/roy/ECE2029/lab_1/lab_1.xpr] - Vivado 2015.4

Project Settings

Project name: lab_1

Project location: /home/roy/ECE2029/lab_1

Product family: Artix-7

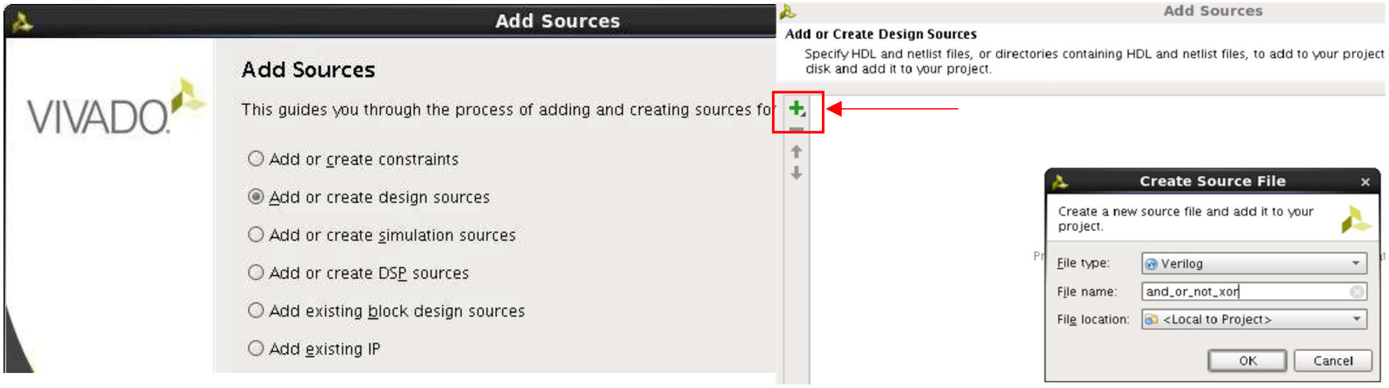
Project part: xc7a35tcpg236-1

Top module name: Not defined

Target language: Verilog

Simulator language: Mixed

2. Select Next and then select Create File (click on the + symbol) and enter the file name for the circuit.



Add Sources

This guides you through the process of adding and creating sources fo

○ Add or create constraints

● Add or create design sources

○ Add or create simulation sources

○ Add or create DSP sources

○ Add existing block design sources

○ Add existing IP

Add or Create Design Sources

Specify HDL and netlist files, or directories containing HDL and netlist files, to add to your project disk and add it to your project.

Create Source File

Create a new source file and add it to your project.

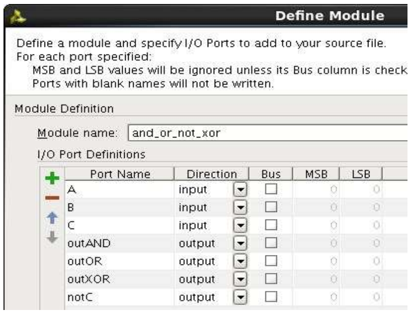
File type: Verilog

File name: and_or_not_xor

File location: <Local to Project>

3. Then click OK and Finish. We can now specify the inputs and outputs to create our AND, OR, NOT and XOR gates. We will use three slide switches as inputs and 4

LEDs on the Basys3 board to display the outputs of each gate. Inputs A and B will be the inputs to the AND, OR and XOR gates, and C will be the input to the NOT gate.



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is check
Ports with blank names will not be written.

Module Definition

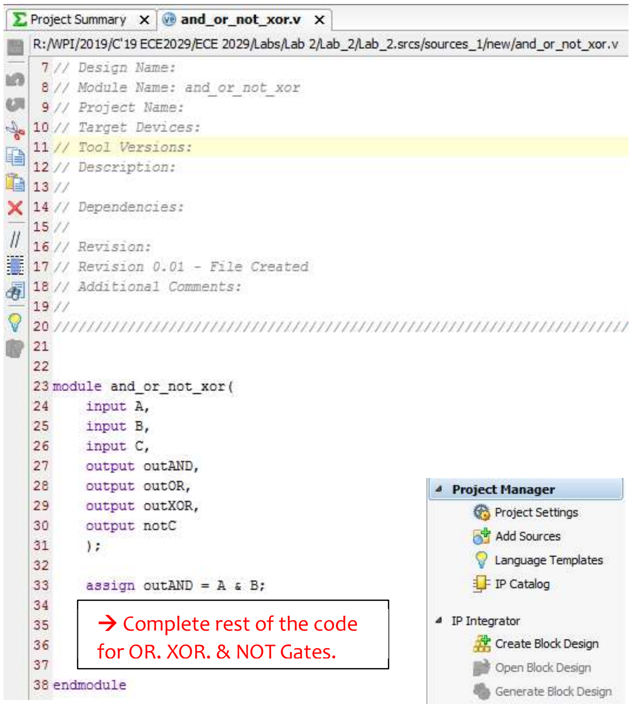
Module name: and_or_not_xor

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
A	input	☐	0	0
B	input	☐	0	0
C	input	☐	0	0
outAND	output	☐	0	0
outOR	output	☐	0	0
outXOR	output	☐	0	0
notC	output	☐	0	0

4. Click OK. Back in the Project Manager Sources window double-click the new and_or_not_xor.v file and you will then see the Verilog file appear in the window on the right. Add your name and a description of this file to the header description. You should ALWAYS complete the comment block at the top of each Verilog module that you write. This is basic professionalism.

5. We can now add the Verilog statements to design our gates. Enter the code to your and_or_not_xor.v file as shown in the figure on the right.



R:/WPI/2019/C'19 ECE2029/ECE 2029/Labs/Lab 2/Lab_2/Lab_2.srcs/sources_1/new/and_or_not_xor.v

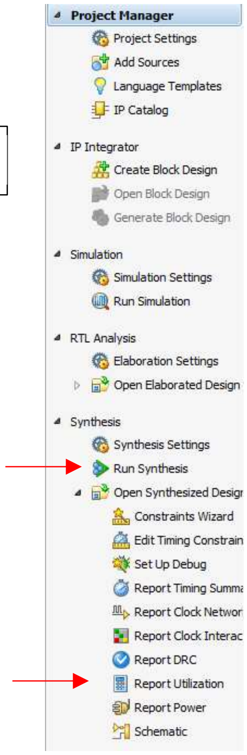
```
// Design Name:
// Module Name: and_or_not_xor
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////


module and_or_not_xor(
    input A,
    input B,
    input C,
    output outAND,
    output outOR,
    output outXOR,
    output notC
    );

    assign outAND = A & B;




endmodule
```

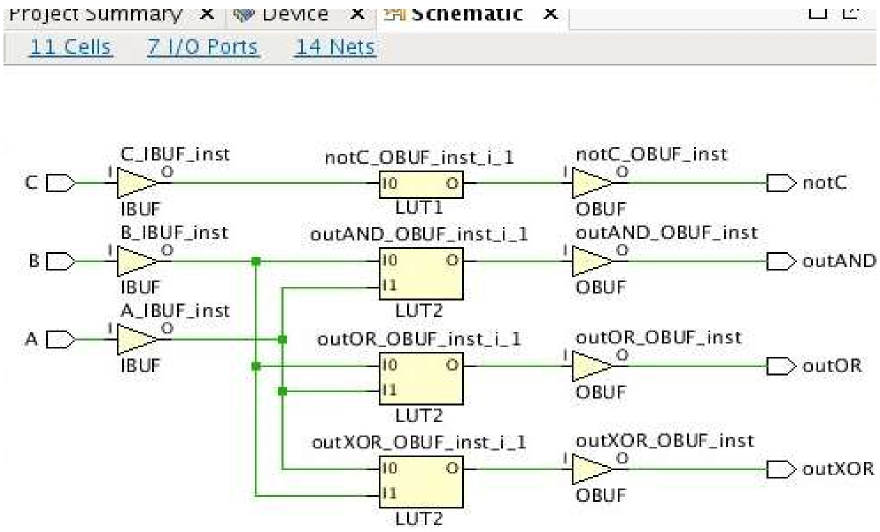
→ Complete rest of the code for OR. XOR. & NOT Gates.

6. Now we can synthesize the design. Click Run Synthesis in the Project Manager window. After synthesis is complete there should be no errors or warnings reported. If you open the synthesized design you can see a device level representation (this is mostly empty since we just have a very simple design that only uses a tiny, tiny fraction of the available FPGA resources).

7. Now we can synthesize the design. Click Run Synthesis in the Project Manager window. After synthesis is complete there should be no errors or warnings reported. If you open the synthesized design you can see a device level representation (this is mostly empty since we just have a very simple design that only uses a tiny, tiny fraction of the available FPGA resources).

8. You can also look at a schematic representation to see the input and output buffers and the Look Up Tables (LUT) used. LUTs can be used to implement any truth table with a certain number of inputs. Notice that since AND, OR and XOR all have A and B as inputs they are implemented in the same LUT.





9. Save your file, you will continue working from here in the lab.

See you in the Lab…☺