The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Comp 541 Digital Logic and Computer Design
Prof. Montek Singh
Spring 2020

Lab #1: Getting Started
Issued Wed. 1/8/20; Due Wed. 1/15/20 (11:59pm)

This lab assignment consists of two parts. For the first part, detailed instructions, including screenshots of almost every step, are provided. Your task is to follow the instructions carefully, and make sure that your software is correctly installed and working properly. You will create a simple project, specify a simple design in the Verilog hardware description language, and simulate it using various inputs. In the second part, you will make a small modification to the design to add an extra output, re-simulate it, and show your results. Detailed instructions are provided below.

PART 0: Software Installation

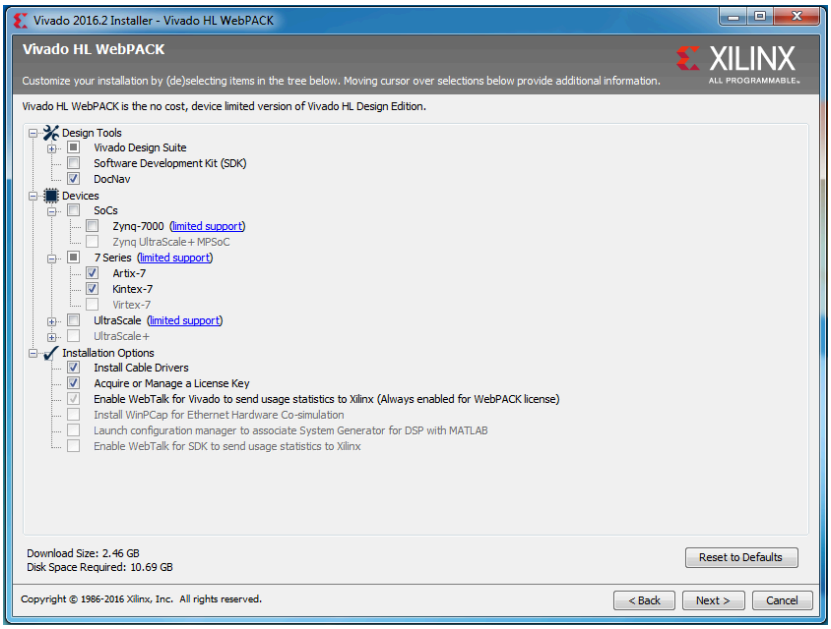
Download

Please follow the link on the class website to download the Vivado tools from the *Xilinx* website. On the Xilinx "Downloads" page, scroll down to the *Vivado Design Suite HLx Editions 2018.3 Windows/Linux Web Installer* section, and click on the link for Windows or Linux installation. (Mac users must run a virtual machine and install Windows or Linux, then follow the instructions here.) Register for an account when prompted, and log in. You will now download and run the web installer and follow the on-screen prompts. Make sure that you are installing on a supported OS (the installer shows a complete list of Windows and certain Linux releases). On the Xilinx page, there is a helpful video at the top of the page that covers installation.

Installation

No spaces allowed in installation path!

Choose *Download and Install Now*. Select the first option: *Vivado HL WebPACK*. On the next screen, under *Devices*, the only box that needs to be checked is *Artix-7* (under *7 Series*). You may uncheck SoCs and UltraScale* since we do not need them, to save disk space. On the following screens, use the default options. If you would like to change the location of the installed software (C:\Xilinx), ***be sure there are no spaces in the path name.*** Under *Installation Options,* be sure that the *Install Cable Drivers* option is checked. Hit *Install.*



Licensing

You DO NOT need to install any license; *Vivado HL WebPACK* comes with a free built-in license. This free license limits the size of the final design, but it is more than adequate for our purposes.

Congratulations! You have successfully installed the software and its license. To launch the software, click the *Vivado 201x.x* shortcut (not *Vivado HLS 201x.x*) from the desktop or the program menu.

Troubleshooting: In case of problems with installing the software, submit a post to the Piazza discussion board. It is likely that other students are encountering the same issue. Seek help early instead of the day before the assignment is due!

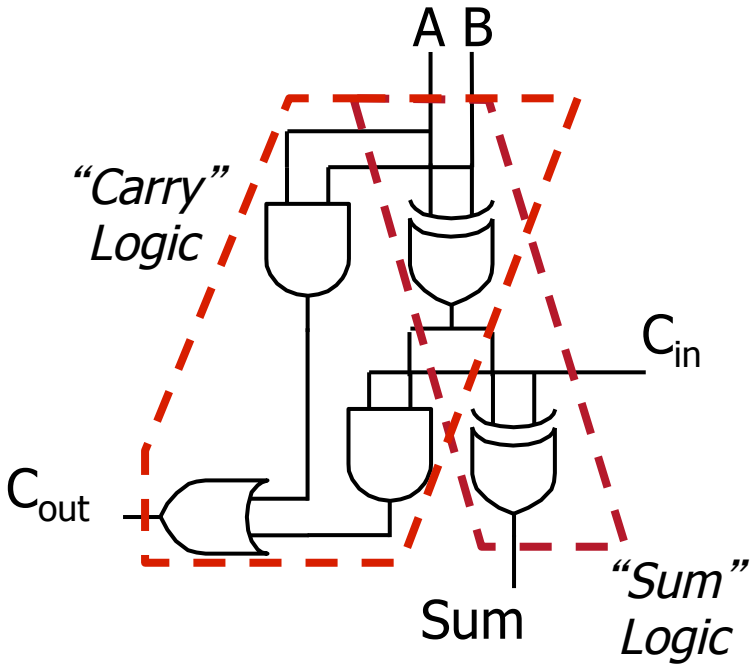
PART I: TUTORIAL

You are going to create a 3-input combinational circuit, with 1 output. The circuit is a single-bit *Full Adder,* with inputs *A, B* and $Carry_{in}$ (*Cin*), and the output is *Sum.* Note that a full adder has a second output, $Carry_{out}$; you will ignore it in this part, but add it in Part II.

The procedure will be to do the following:

- Create a new project
- Add a Verilog file that describes the circuit
- Add a Verilog file that provides the inputs
- Simulate the circuit
- Observe the outputs and verify they are correct

For your reference, here is the circuit and Boolean equations for a *Full Adder* (from Comp411).



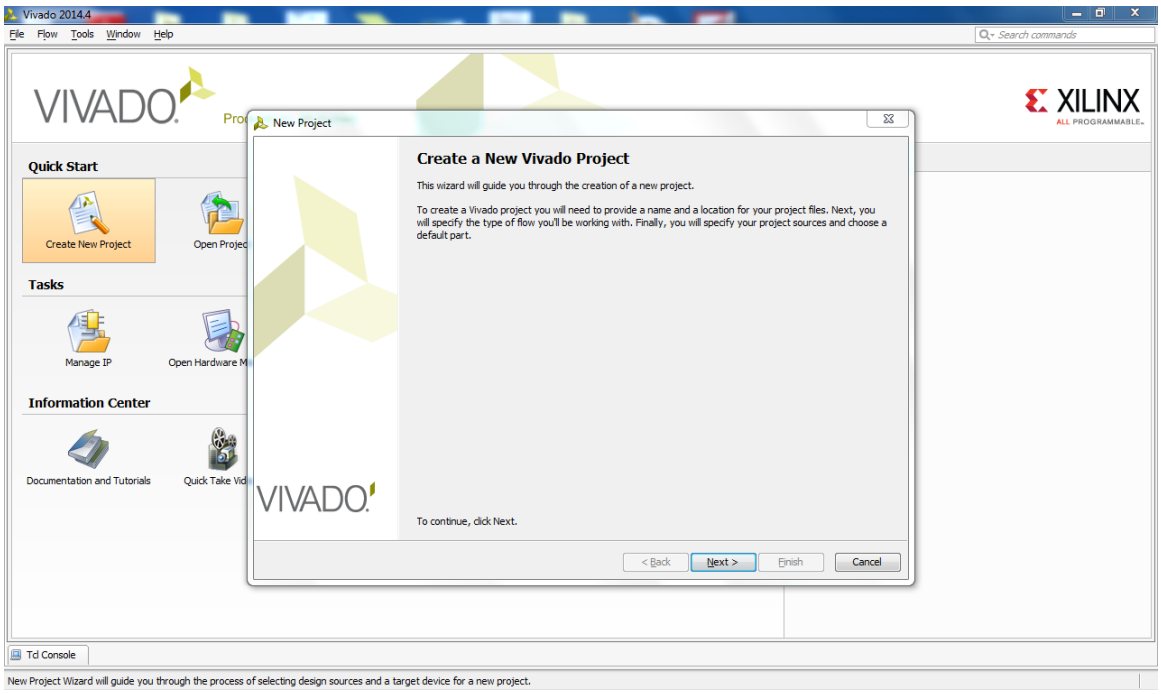
$$C_{out} = C_{in}\left(A \oplus B\right) + AB$$

$$Sum = C_{in} \oplus A \oplus B$$

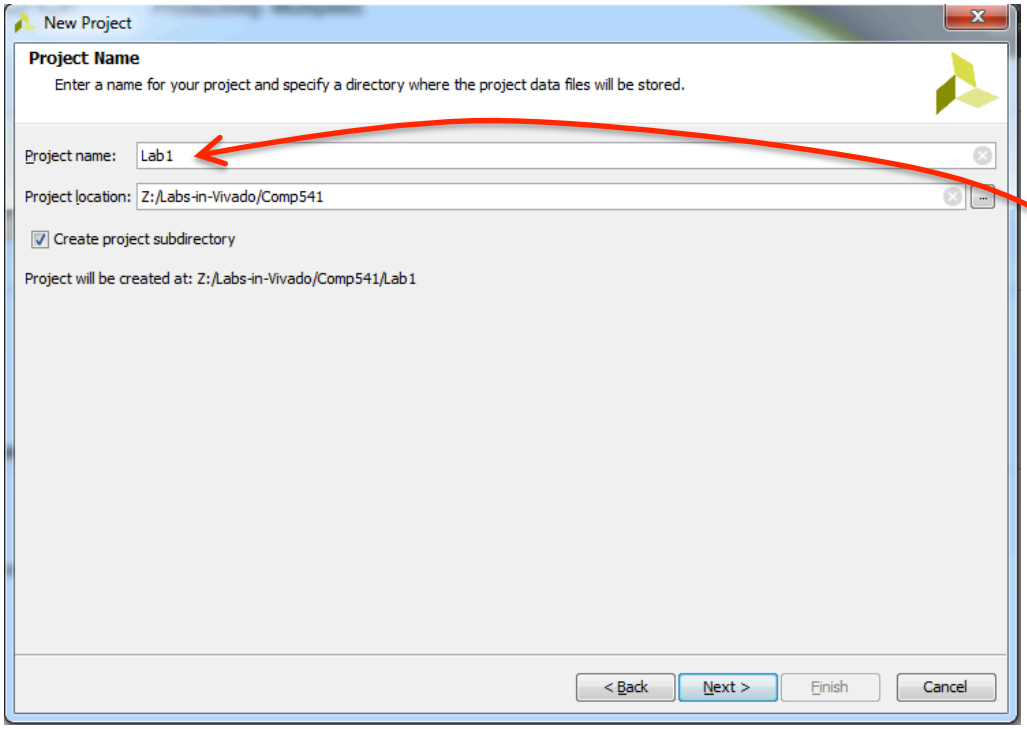
Creating a Project

No spaces allowed in assignments folder!

Create a folder on your computer's drive where you will save all of the lab assignments for Comp 541. (The path Z:/Labs-in-Vivado/Comp541 will be used in this tutorial, but your path may be different.) **Make sure there are no spaces in the path name!** Launch the design software by double-clicking on the Vivado icon on your desktop, and click *Create New Project.* This is what you will see.



Click *Next,* and then enter the name for your new project in the dialog box as follows:



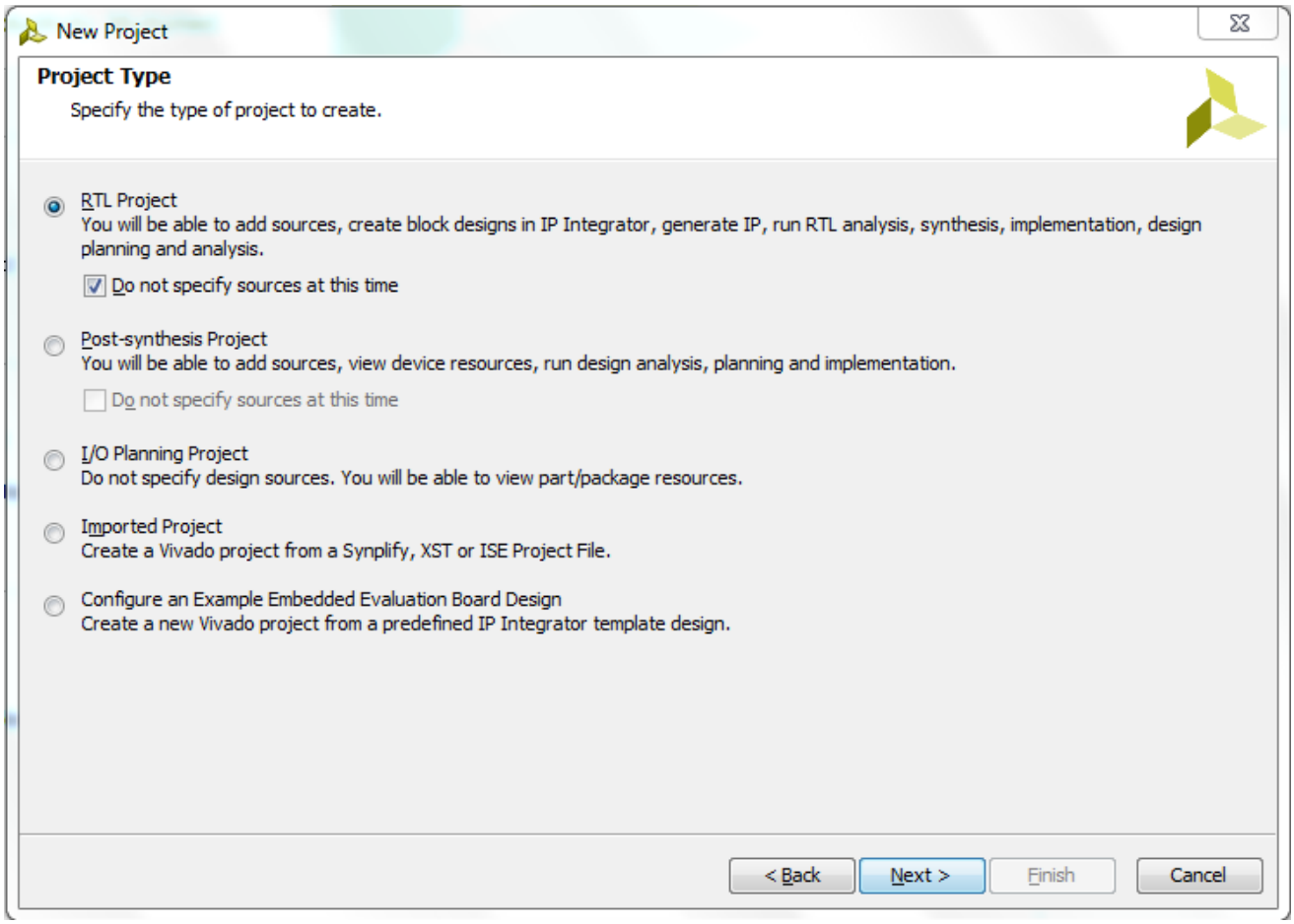
No spaces allowed in project name!

Choose a project name. This will also become a subdirectory of the location you choose next. Call it, say, Lab1.

Project Location: Browse and change the folder to where you will be saving all of your work this semester.

NOTE: Make sure there are ***NO*** spaces in the name and the path! Use "Lab1" instead of "Lab 1". Spaces **will** cause major headaches later on!

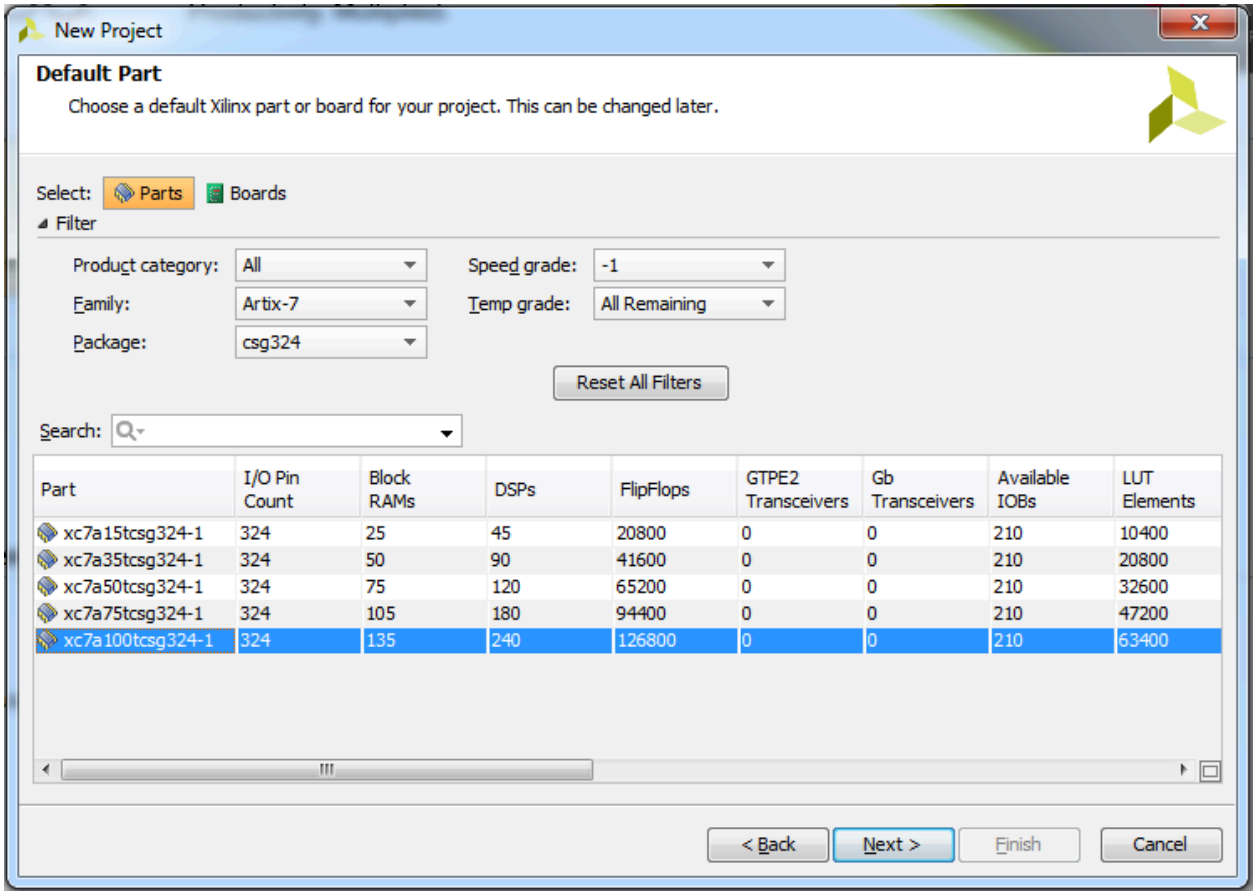
Click *Next.*



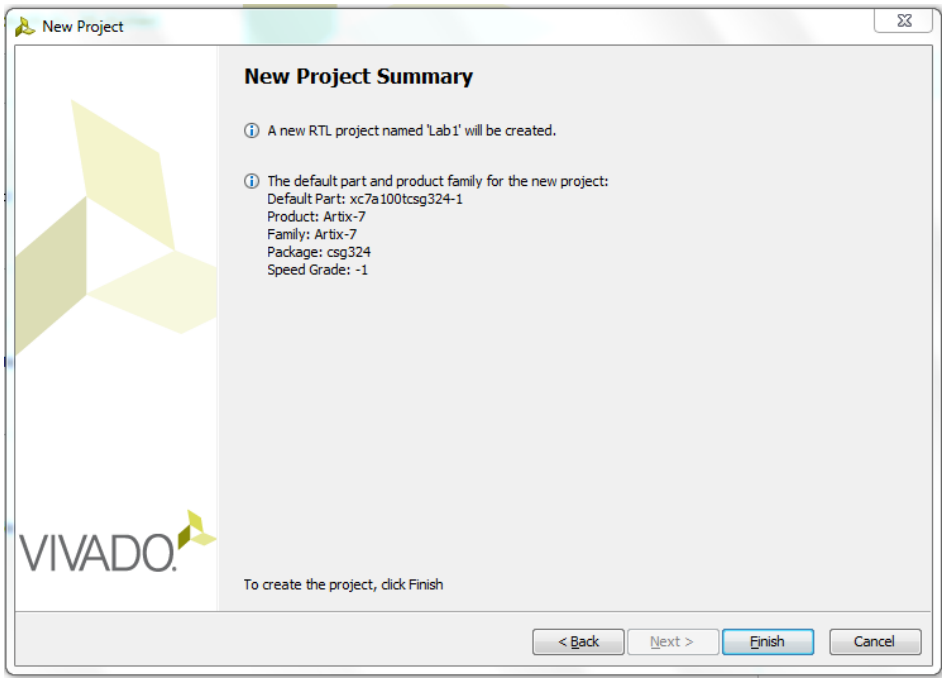
Choose RTL Project.

Click *Next.*

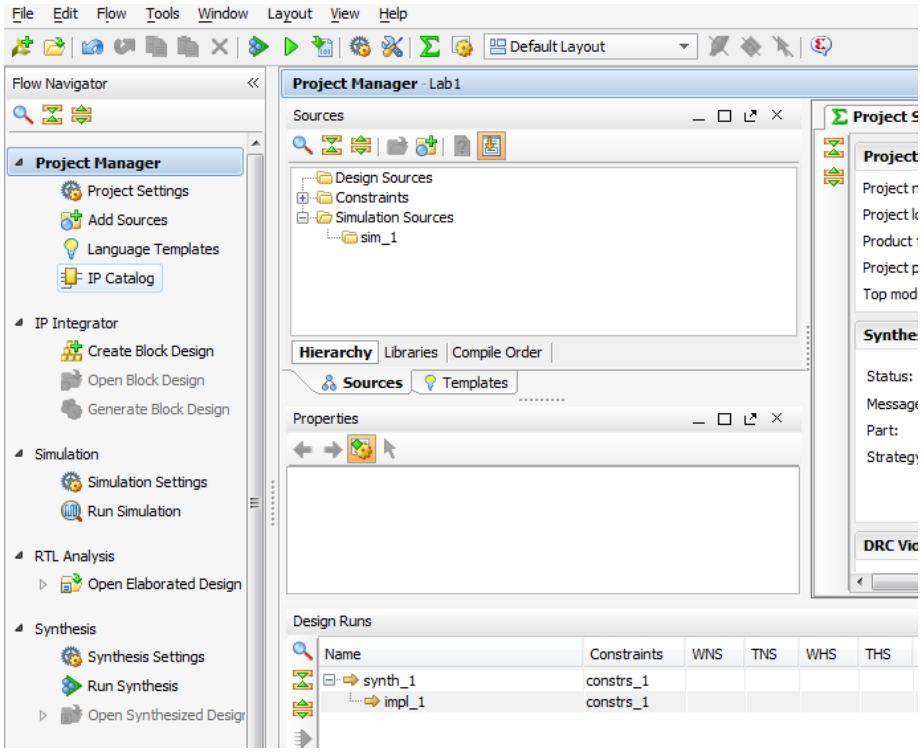
Select the part number corresponding to our development boards by making the selections shown below:



Click *Next* and then *Finish.*

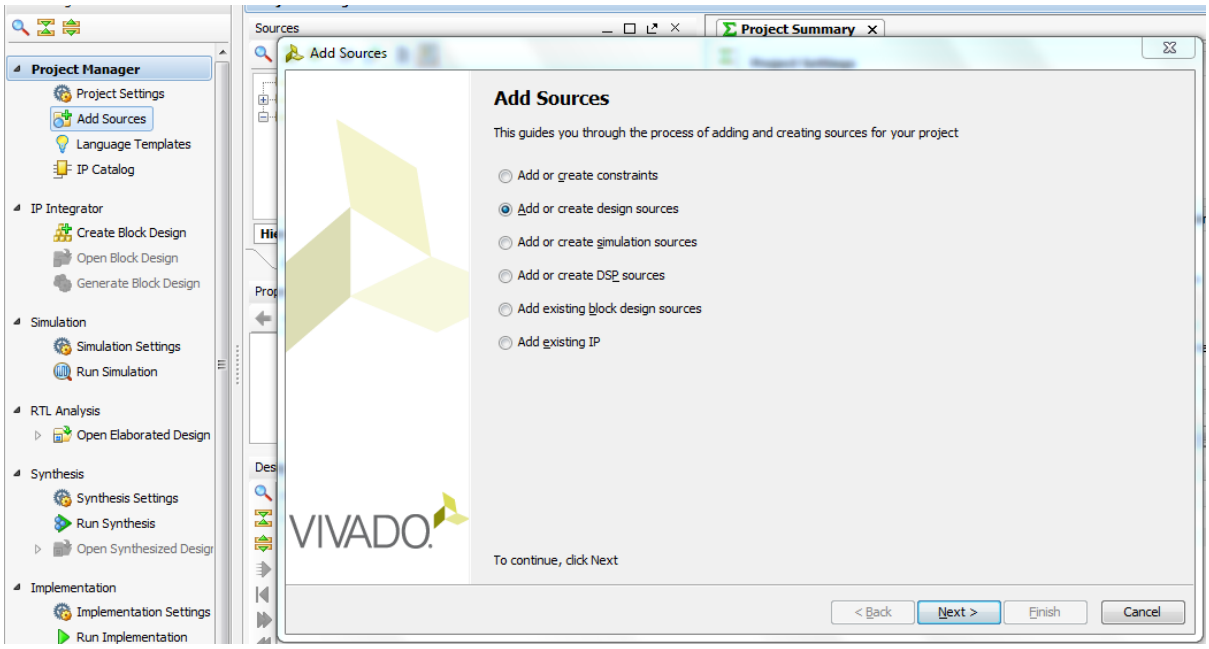


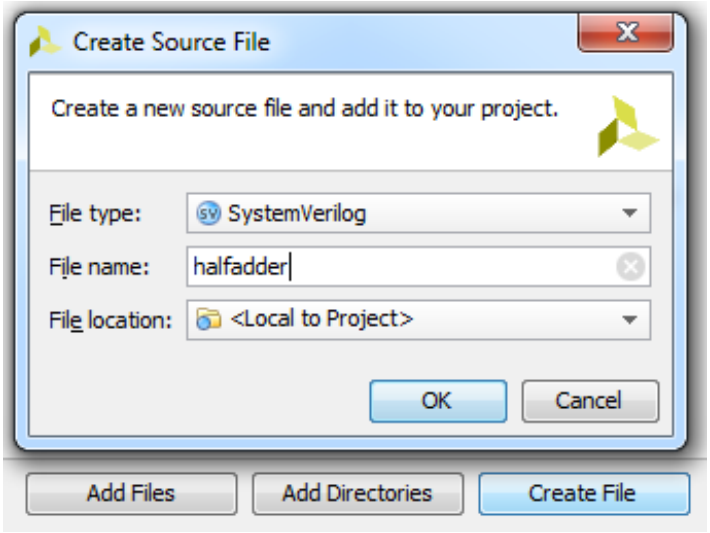
You have now created a new project, although an empty one. You should see this:



Adding a new source file.

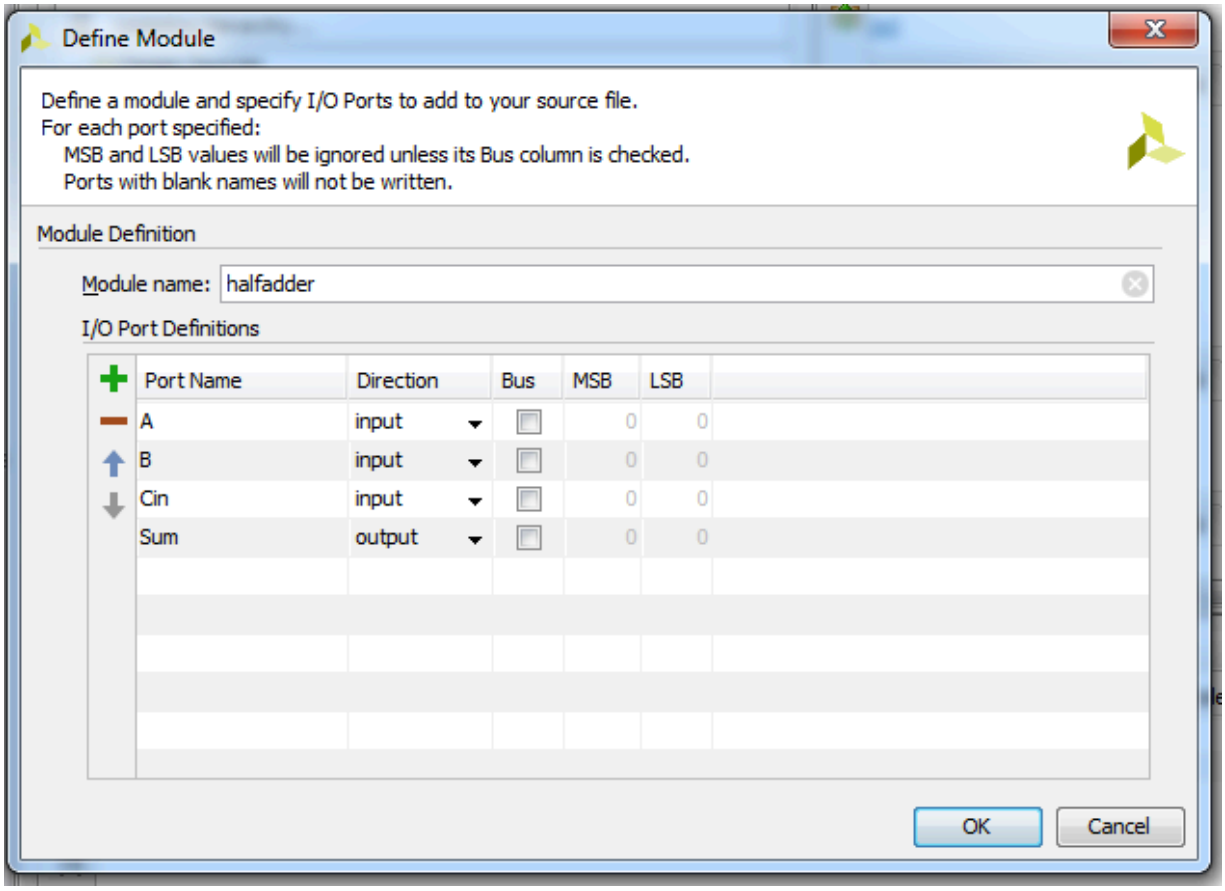
In the toolbar to the left, click on *Project Manager* and select *Add Sources*. Choose *Add or create design sources,* and click *Next.*





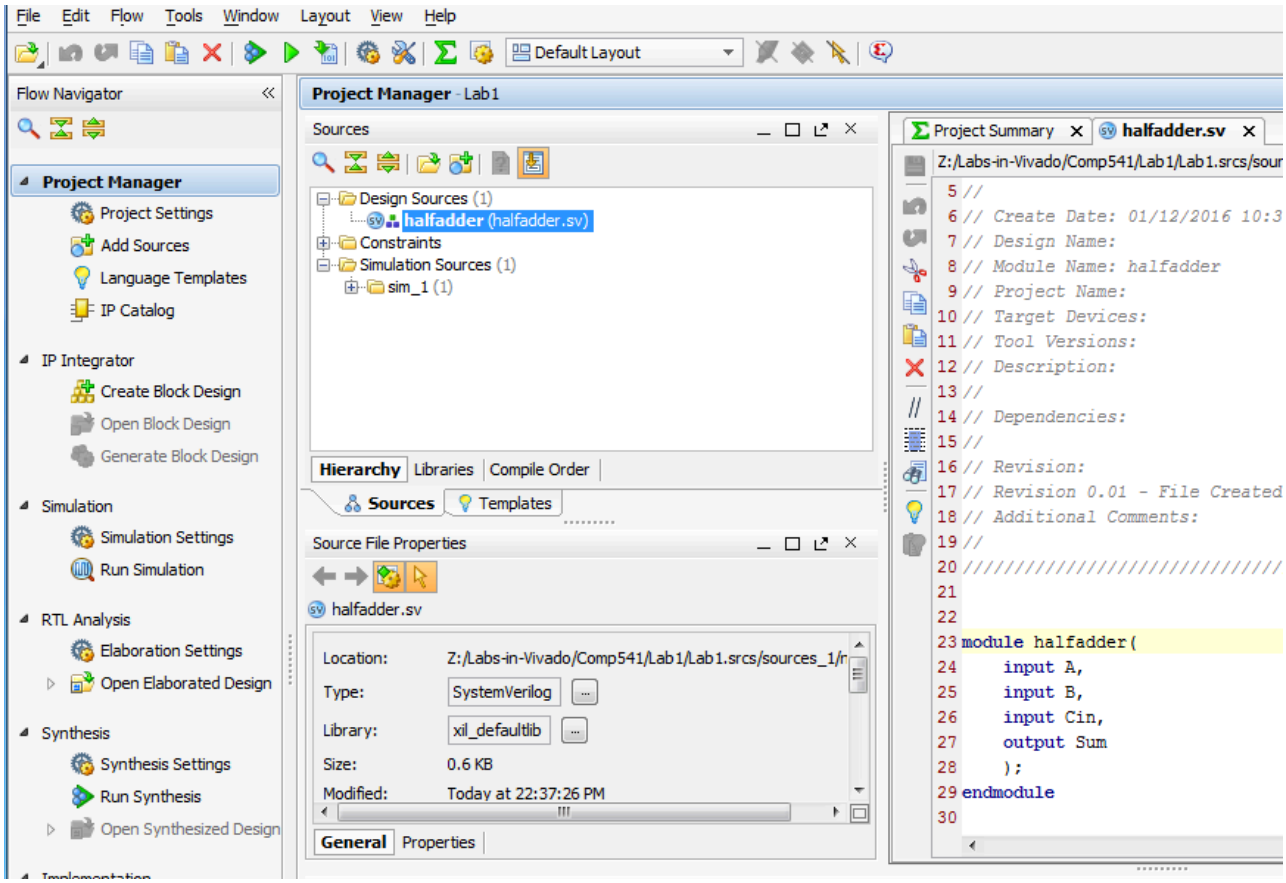
Click *Create File,* then choose *SystemVerilog* as the file type, name the file *halfadder,*[1] and click *OK,* then click *Finish.* Note: SystemVerilog is a new version of Verilog with many system-level extensions. We will be using SystemVerilog throughout this course, although we might loosely use the name Verilog in some of the lab write-ups and in the classroom.

Enter the input/output signal names as shown below, and click *OK*.



[1] Strictly, this design is not really a "half adder", but "half of a full adder"!

In the *Sources* window, double-click *halfadder* to open it for editing, and scroll down a bit. You should see the following:



The editor opens and shows you a skeleton of the Verilog file. Fill in the Verilog description below (Note: add the keyword `wire` to the input/output names):

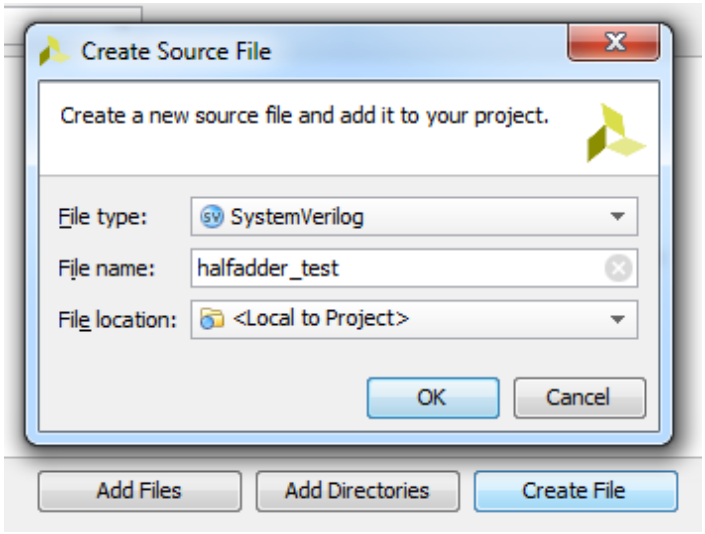
```
module halfadder(
    input wire A,
    input wire B,
    input wire Cin,
    output wire Sum
    );

    assign Sum = Cin ^ A ^ B;

endmodule
```

Note that '^' is the Verilog symbol for the exclusive-or (XOR) operation. This Verilog fragment corresponds to the Boolean equation for *Sum* from Page 2.

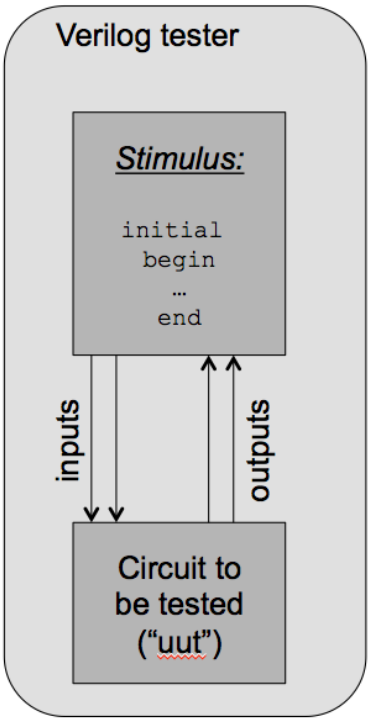
Hit *File* → *Save File.*

Simple Simulation

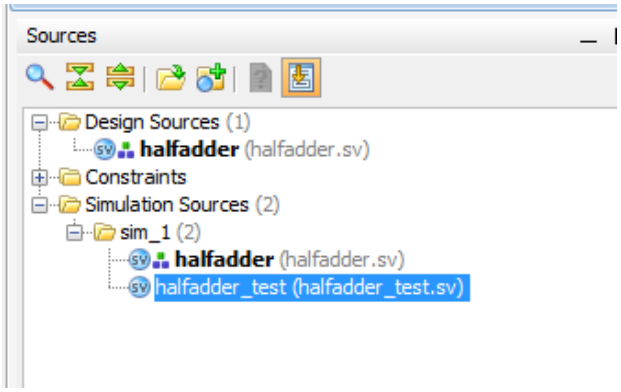
In order to simulate your circuit, you will need to specify a set of inputs (which change over time). To do so, again click *Add Source* under the *Project Manager,* but this time select *Add or create simulation sources* from the dialog box. Fill in the file name as shown, and click *OK* and *Finish* (and then click OK, then *Yes*).



The tester's job is to provide inputs to, and receive outputs from, the half adder module. Hence, the tester encloses the half adder, and as such does not have any inputs or outputs (see the block diagram). Therefore, do not add any input/output ports to the tester, and click *OK,* click *Yes* to confirm.



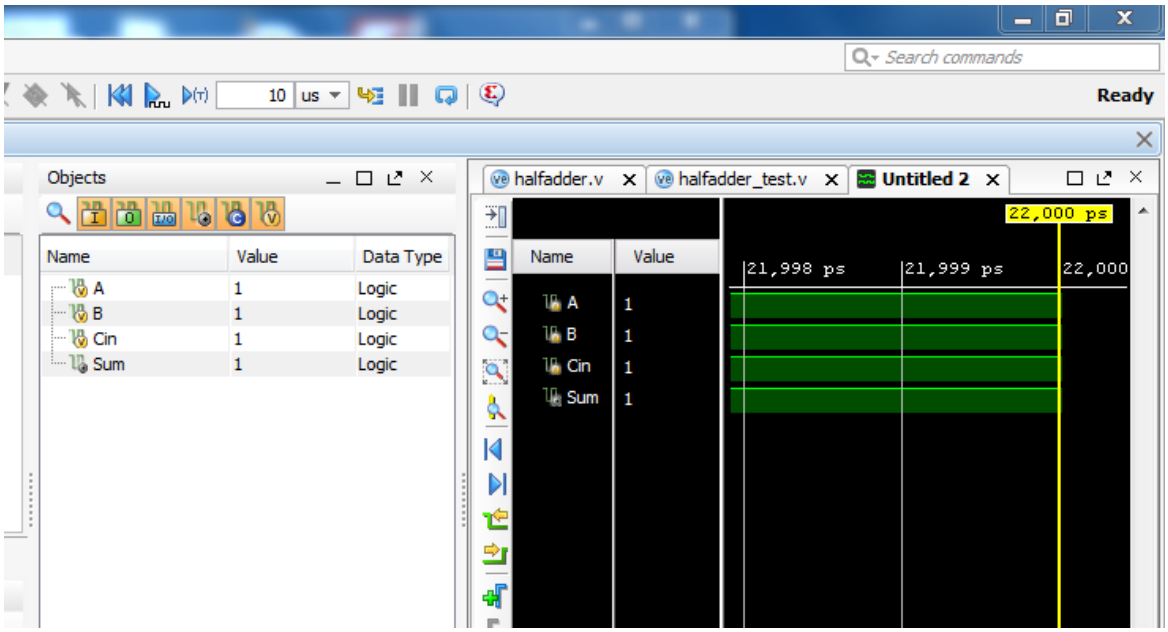
In the *Sources* window, under *Simulation Sources,* expand *sim_1*. You will see:



Double-click on *halfadder_test,* and in the editor that opens, copy-and-paste the contents of the file `halfadder_test.sv` provided on the class website, and hit *Save File.*

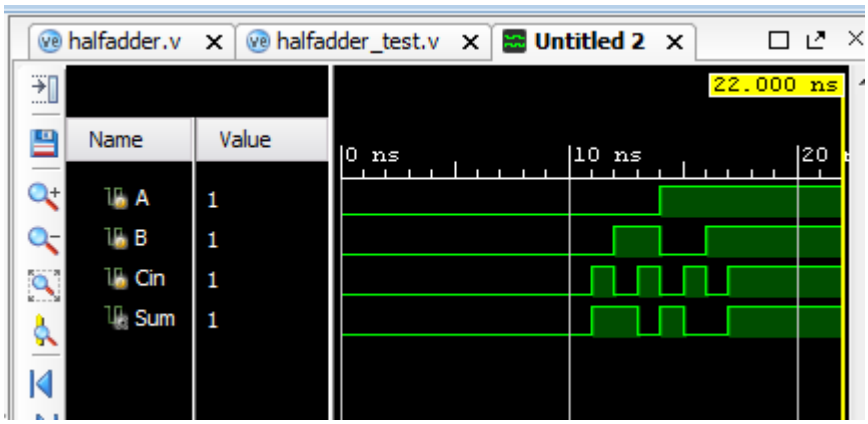
Click to highlight the module *halfadder_test* under *Simulation Sources* in the *Sources* window. Then click on *Run Simulation* under *Simulation* in the left toolbar, and choose *Run Behavioral Simulation.*

The simulator will launch and once it is done running, an output window will be created showing the waveforms corresponding to the inputs and outputs of the module being tested. This window will be in a new tab on the top right.

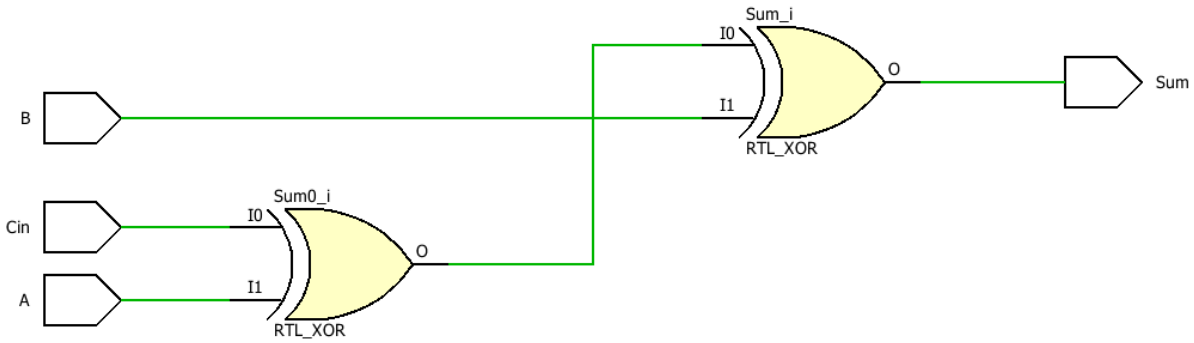


Click on the "Zoom to fit" icon.

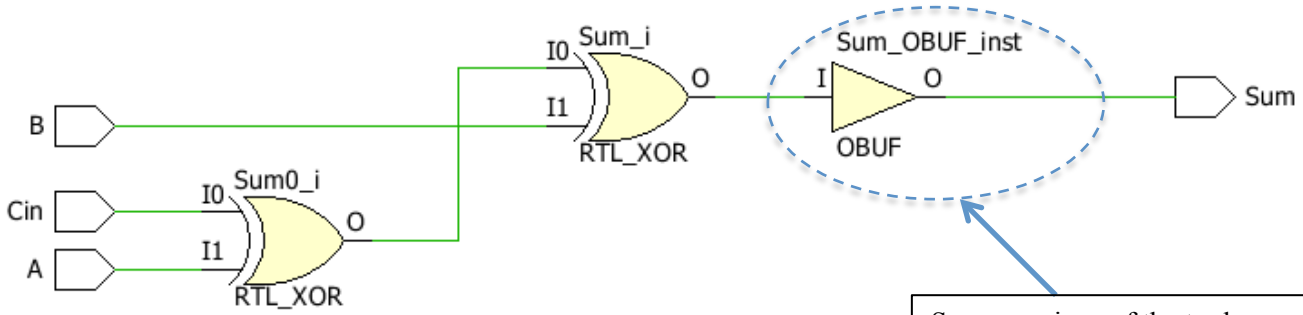
You will now see the resulting waveforms of all the inputs as well as the output!



Finally, let us generate a circuit diagram for the circuit we just described. Select *RTL Analysis* in the left toolbar, and click *Open Elaborated Design* to automatically generate a schematic (circuit) diagram. You should see the following:



With some (older) versions of the tool, you might see the following instead:



Some versions of the tool automatically insert a buffer at the output (and sometimes at the inputs). *Vivado 201x.x* may add them, but sometimes doesn't show them in the schematic diagram. Logically, these buffers do nothing; the output of the XOR to the right simply goes through the buffer to become the *Sum* output. However, buffers offer electrical advantages (noise removal and amplification of the signal strength), and the tool typically adds them to all outputs (and sometimes elsewhere, as needed).

If your waveforms and schematic diagram match those shown here (ignoring buffers), then you have successfully completed Part I!

Also note that your drawing may be laid out somewhat differently but still represent the same circuit topologically. These differences could be due to different tool versions, differences in the ordering of Verilog text, screen/window size, etc. Such variations are okay and expected.

PART II: EXERCISE

Modify your work from PART I to add the second output to your circuit: $Carry_{out}$. This requires two simple changes to your Verilog description from Part I:

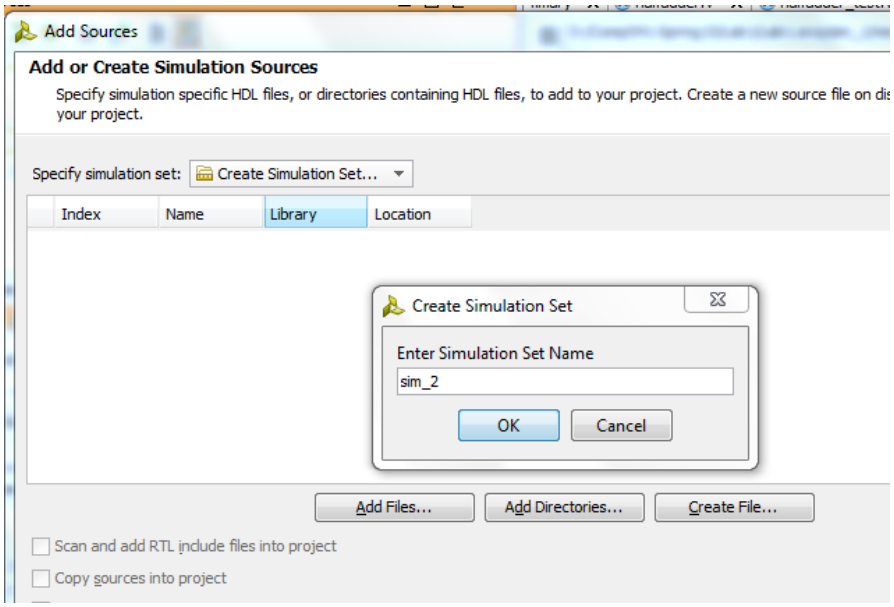
- Adding another output ($Carry_{out}$) to the module's interface
- Adding another continuous assignment (using keyword *assign*) to describe the Boolean equation for computing $Carry_{out}$. Please refer to Page 2 for the equation.

Note that the Verilog symbol for an AND operation is '&', for an OR operation is '|', and for an XOR operation is '^'. For a complete list of Verilog operators, please refer to the online Verilog reference (linked from the course website), Section 11. Be sure to use parentheses '(…)' if you are unsure of the precedence of '&', '|' and '^' operators.

Create a new source, and call it *fulladder*. Copy-and-paste the contents of *halfadder* into it, and then modify it by making the two changes listed above. They are shown highlighted below (fill in the blanks):

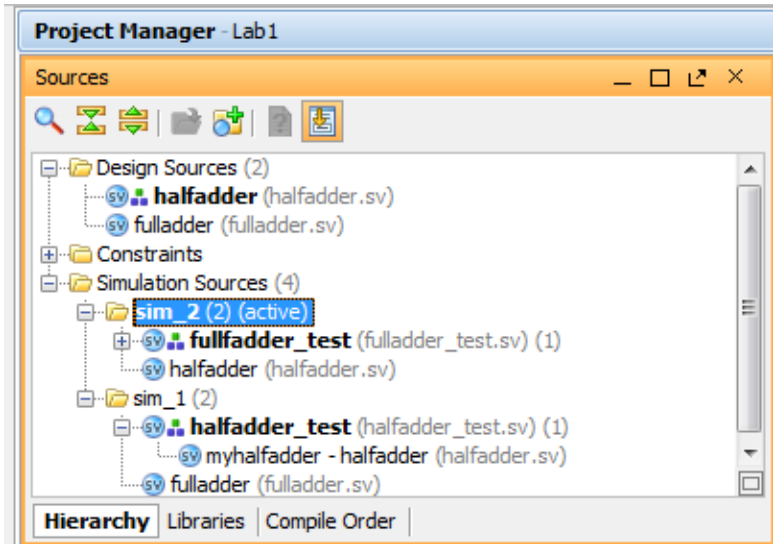
```
module fulladder(
    input wire A,
    input wire B,
    input wire Cin,
    output wire Sum,
    output wire Cout
    );

    assign Sum = Cin ^ A ^ B;
    assign …     = …

endmodule
```

You will now simulate the new *fulladder* module by repeating all of the simulation steps from Part I. First, create a new tester by adding a new simulation source. However, to keep this new simulation separate from the simulation of Part I, we will create a new "simulation set." Click *Add Sources* → *Add or create simulation sources* → *Specify simulation set,* choose *Create Simulation Set.* Use the default new name of *sim_2*.

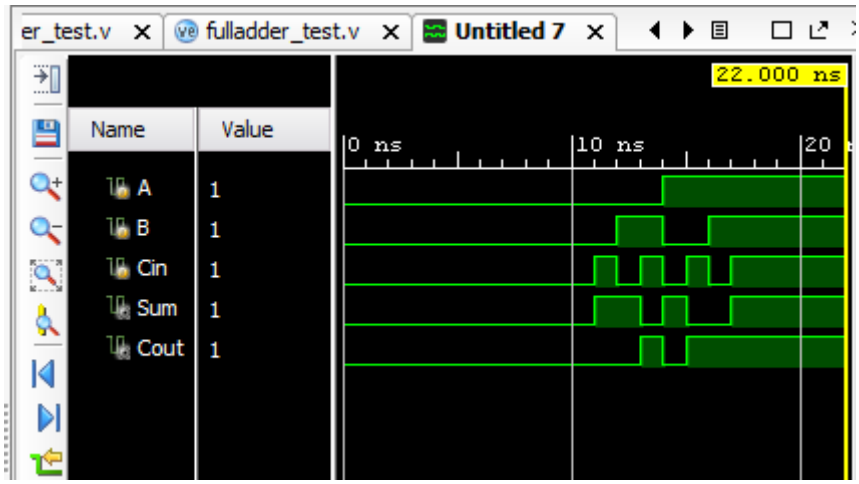


Create the new source, name it *fulladder_test,* and paste the contents of the file `fulladder_test.sv` provided on the class website.

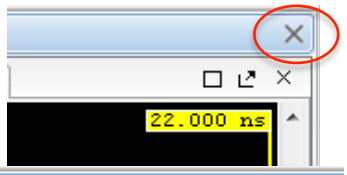
In the *Sources* window, right-click the new *sim_2* simulation set, and hit *Make Active.* This ensures that now when you run simulation, the second simulation set is run. The sources window should look like this:



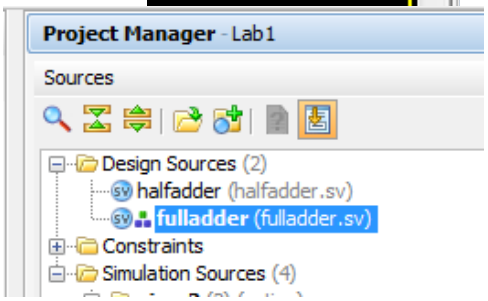
Click *Run Simulation* → *Run Behavioral Simulation.* Select the waveform window that appears, and zoom to fit. Observe the waveforms. Make sure the carry output is correct, along with the sum. If your work is correct, the waveform should look like this:



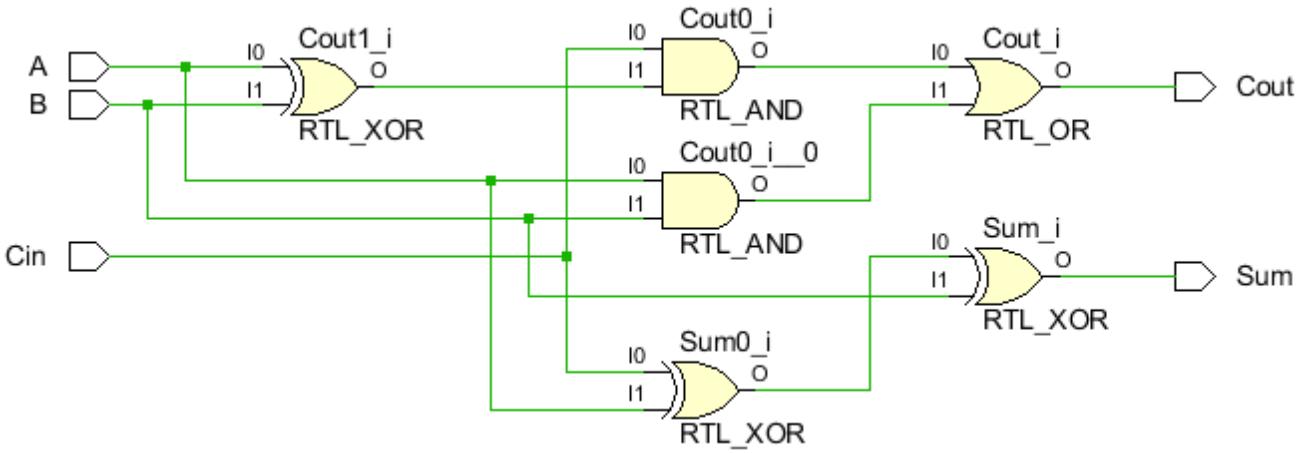
Congratulations! You have successfully created and simulated your first design in Verilog using the Xilinx tools. Finally, let us view the schematic diagram. First, close the behavioral simulation by clicking the X at the top right of the picture shown here.



Then, right-click the *fulladder* under *Design Sources,* and choose *Set as Top,* to select the full adder, as shown in this picture.



Now choose *RTL Analysis* → *Open Elaborated Design* → *Schematic* to create its schematic diagram. It should look like this:



If your waveforms and schematic diagram match those shown here (ignoring any buffers), then you have successfully completed Part II!

NOTE: The circuit diagram is not exactly the same as that on Page 2 (even ignoring the two buffers at the outputs). In particular, the circuit here has an extra XOR gate. This is because the diagram on Page 2 shared

an XOR gate among the *Sum* and *Cout* outputs, whereas the implementation here duplicates it. We will fix this in the next lab.

What to submit: **Two screenshots: (i) the waveform window for Part II only, clearly showing the 3 input and 2 output waveforms, and (ii) the schematic diagram generated by the tool for Part II.**

How to submit: **We will be using electronic submission (save a tree!). Please submit your work by email as follows:**

- **Send email to: `comp541-submit-s20@cs.unc.edu` (don't submit before Mon, Jan 13).**
- **Attach the simulator screenshot using the filename `simulation.png` (or other appropriate name/extension), and the schematic screenshot using the filename `schematic.png`.**
- **Include a statement in the email body that you confirm that: (i) you installed the Xilinx software in a location whose path has no spaces in it; and (ii) you created your Verilog files in a folder that has no spaces in its name. Otherwise, major headaches will arise in later labs.**
- **Submit your work by 11:59pm on Wednesday, Jan 15.**
