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

Comp 541 Digital Logic and Computer Design
Fall 2014

Lab #1: Getting Started
Issued Wed. 8/20/14; Due Wed. 8/27/14 (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

Please follow the link on the class website to download the Xilinx ISE WebPack from the Xilinx website. You will need to register with a user ID and password. Download the file (.tar), and uncompress it (if you need WinZip to uncompress it, you can obtain it for free from shareware.unc.edu). Double click on the file *xsetup* to launch the installation wizard, and follow on-screen prompts.

Once installation is complete, create a shortcut to the Project Navigator on your desktop, so it is easier to launch it. Following these steps:

> *Start Menu → All Programs → Xilinx ISE Design Suite 14.7 → ISE Design Tools → Project Navigator (right click) → Send to → Desktop (create shortcut).*

Licensing

You will need the free ISE WebPack license. Follow these steps:

- First, make sure that Internet Explorer is set to be your default browser (you can change it, and restore it to your favorite browser after you have downloaded and installed your license file).
- Launch the Project Navigator by clicking on the desktop shortcut you just made. (Dismiss the Tip of the Day.)
- Obtain a license file by following either of these two steps:
  - *Help → Obtain a License Key → Get Free ISE WebPack License;* OR
  - Use your browser to go to www.xilinx.com → *Support → Downloads and Licensing → Get License File.*
- Log in with your User ID and Password, and verify your contact information. Then, go to *Create a New License File* → check the box for *ISE WebPack License*, and follow on-screen instructions.
- The license file will be emailed to you as an attachment (*Xilinx.lic)*. Save it to your desktop or downloads folder. Go back to the Project Navigator, and launch the License Manager (click *Help → Manage License*), click *Copy License,* browse to the license file you just downloaded from your email, select it and click *Open.* Now your copy of the software should be appropriately licensed, and you can start using it. Congratulations!

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!

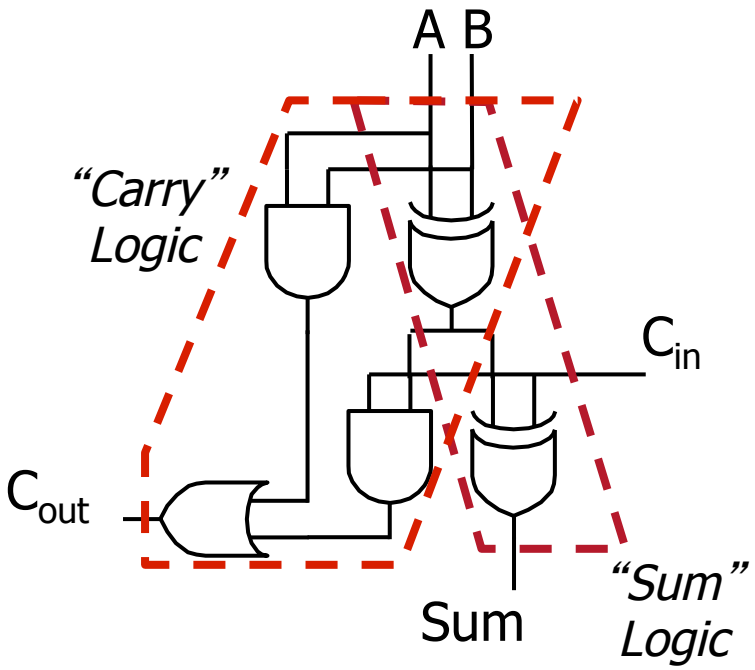
PART I: TUTORIAL

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).

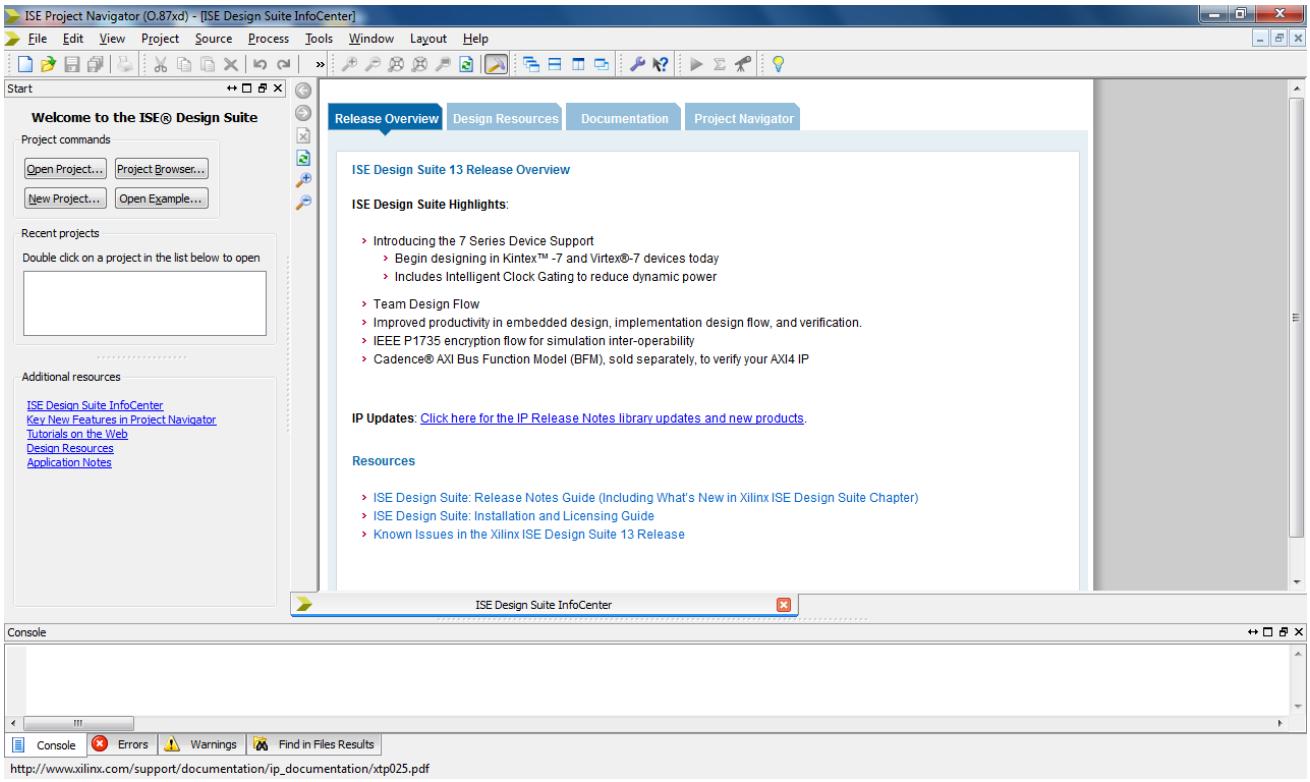


$$C_{out} = C_{in}\left(A \oplus B\right) + AB$$

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

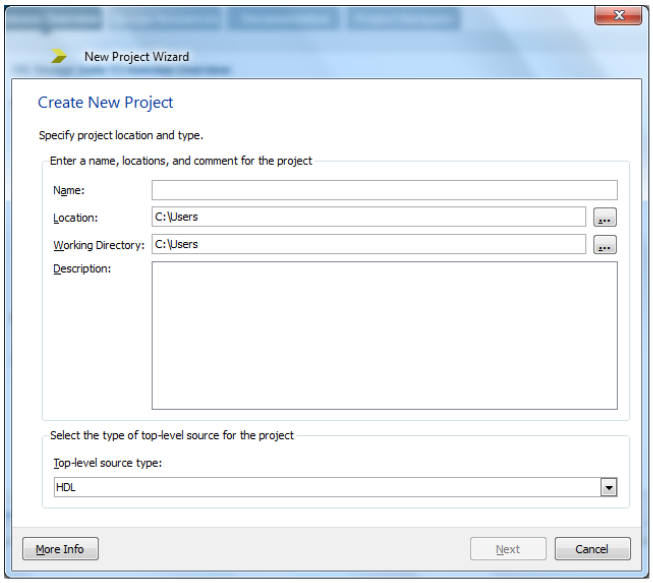
Creating a Project

Start by double-clicking on the Project Navigator icon on your desktop. This is what you will see.



You can close the InfoCenter window by clicking on the X below it.

To create a new project, select *File* → *New Project,* which will bring up the following window:

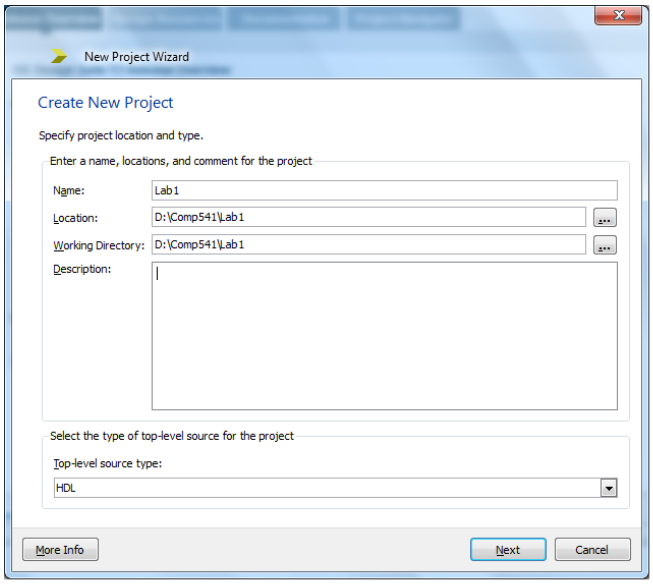


Choose a project name. This will also become a subdirectory of the location you choose next. Call it, say, Lab1.

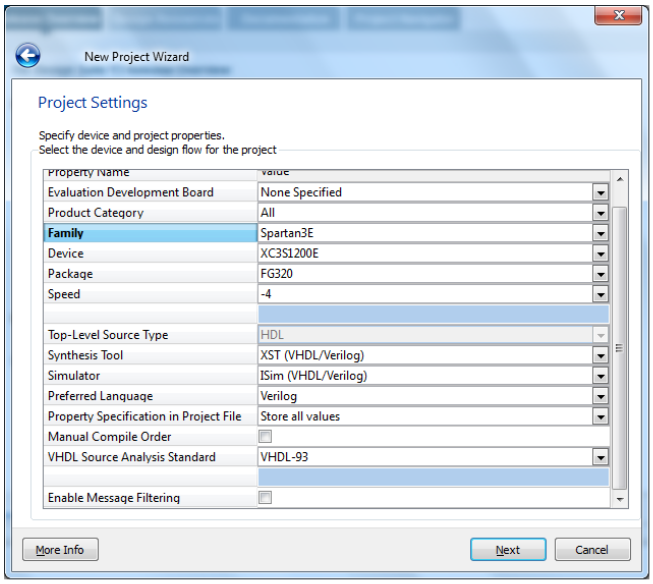
Project Location: Browse and change the folder to some convenient place.

NOTE: Make sure there are ***NO*** spaces in the name and the path! This has caused a problem in the past.

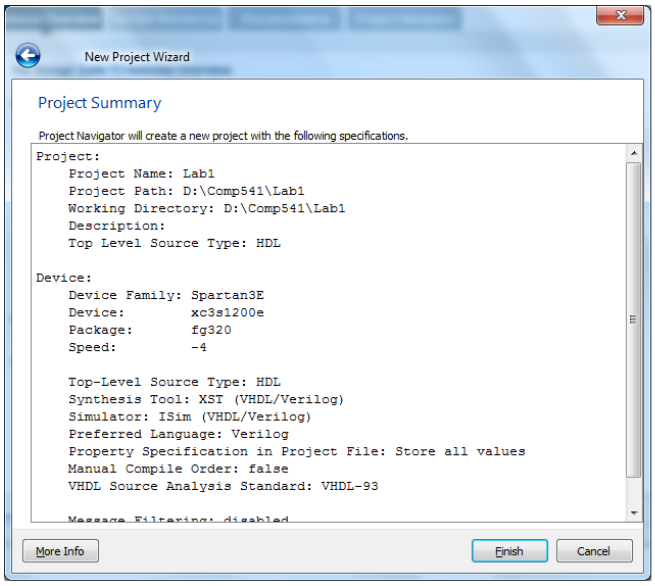
Choose HDL (Hardware Description Languages).



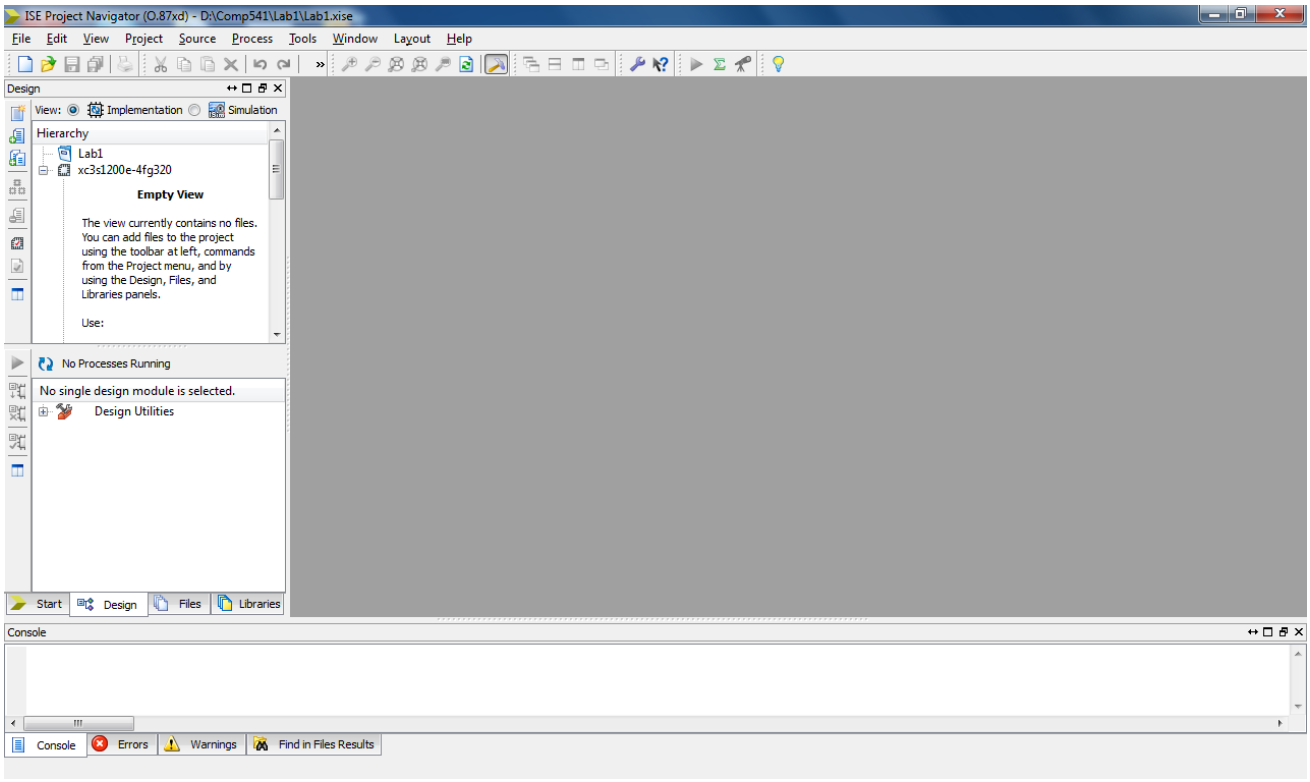
Click *Next.*



Click *Next* and then *Finish.*

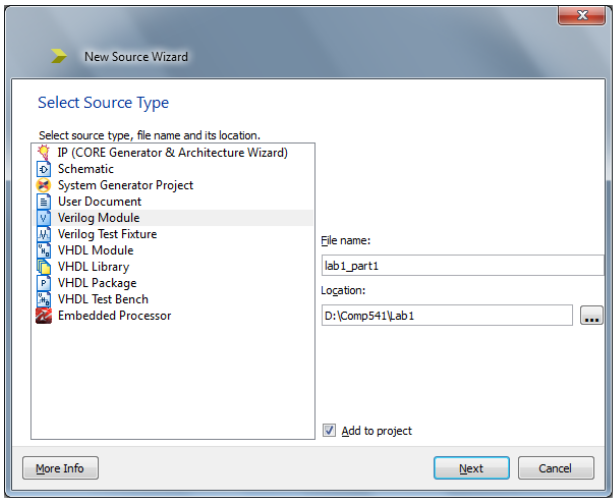


You have now created a new project, although an empty one. You should see this:

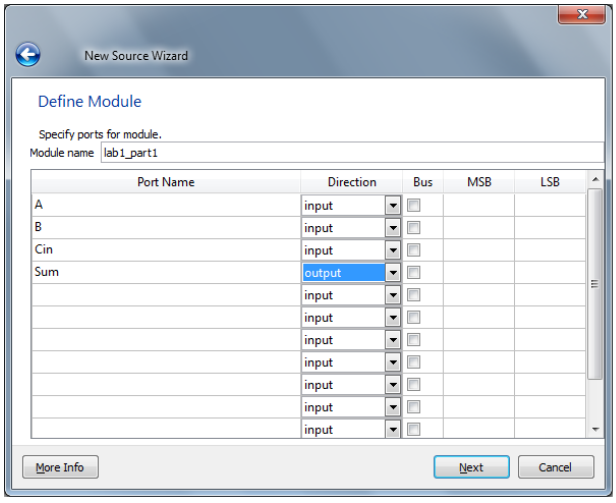


Adding a new Verilog source file.

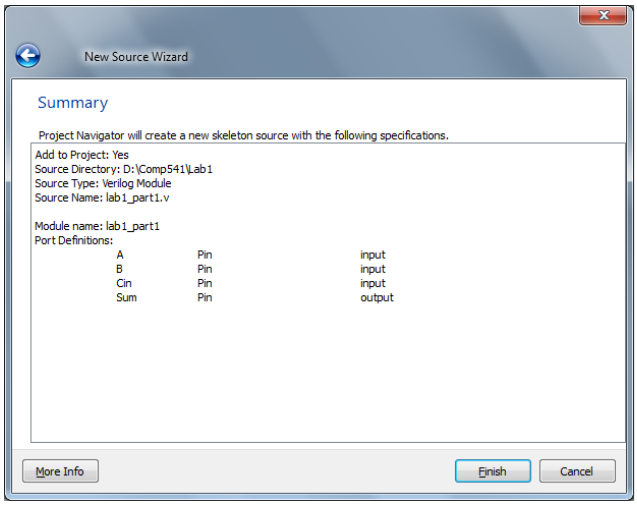
In the toolbar to the left, click the top icon (which says “New Source” if you hover over it). Alternatively, right-click on *xc3s1200e-4fg320,* and choose “New Source”).



On the left panel, select *Verilog Module,* and fill in the file name as shown. Click *Next.*



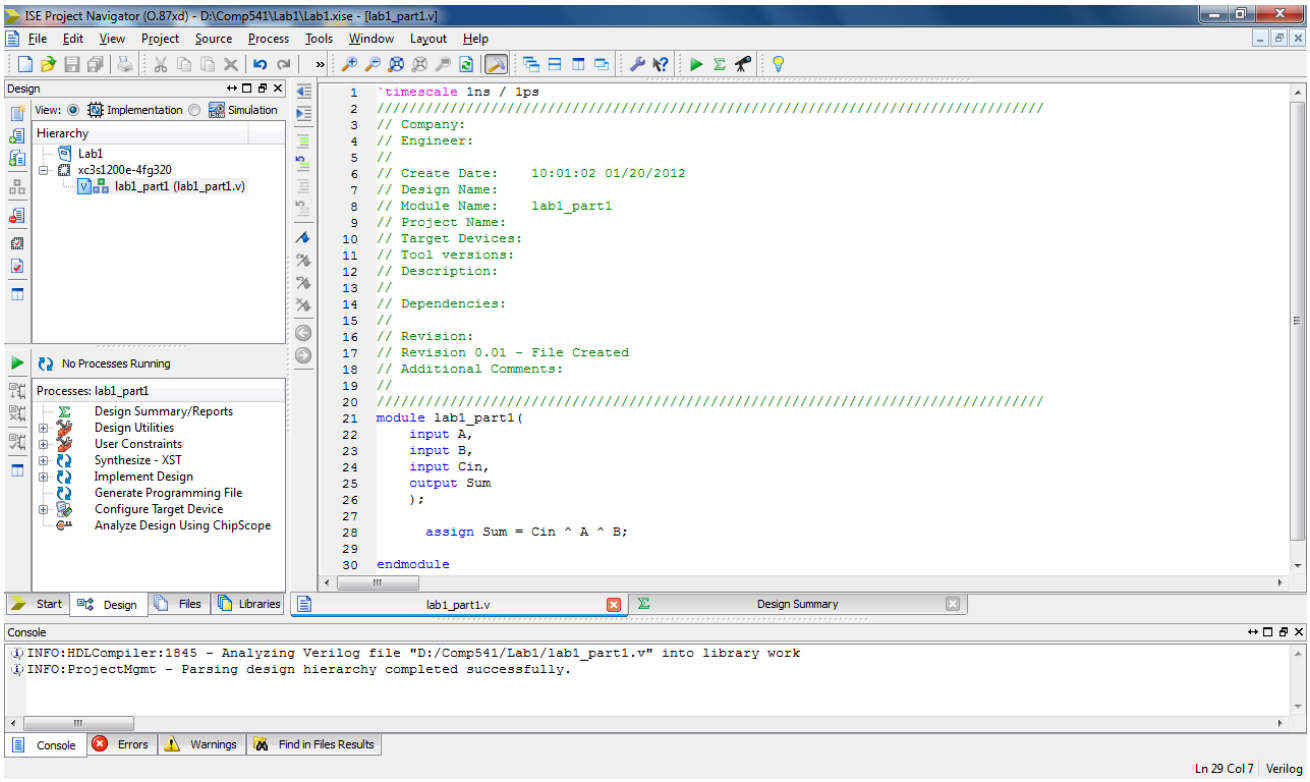
Specify the inputs and outputs and click *Next,* then *Finish.*



The editor opens and shows you a skeleton of a Verilog file. Fill in the code shown:

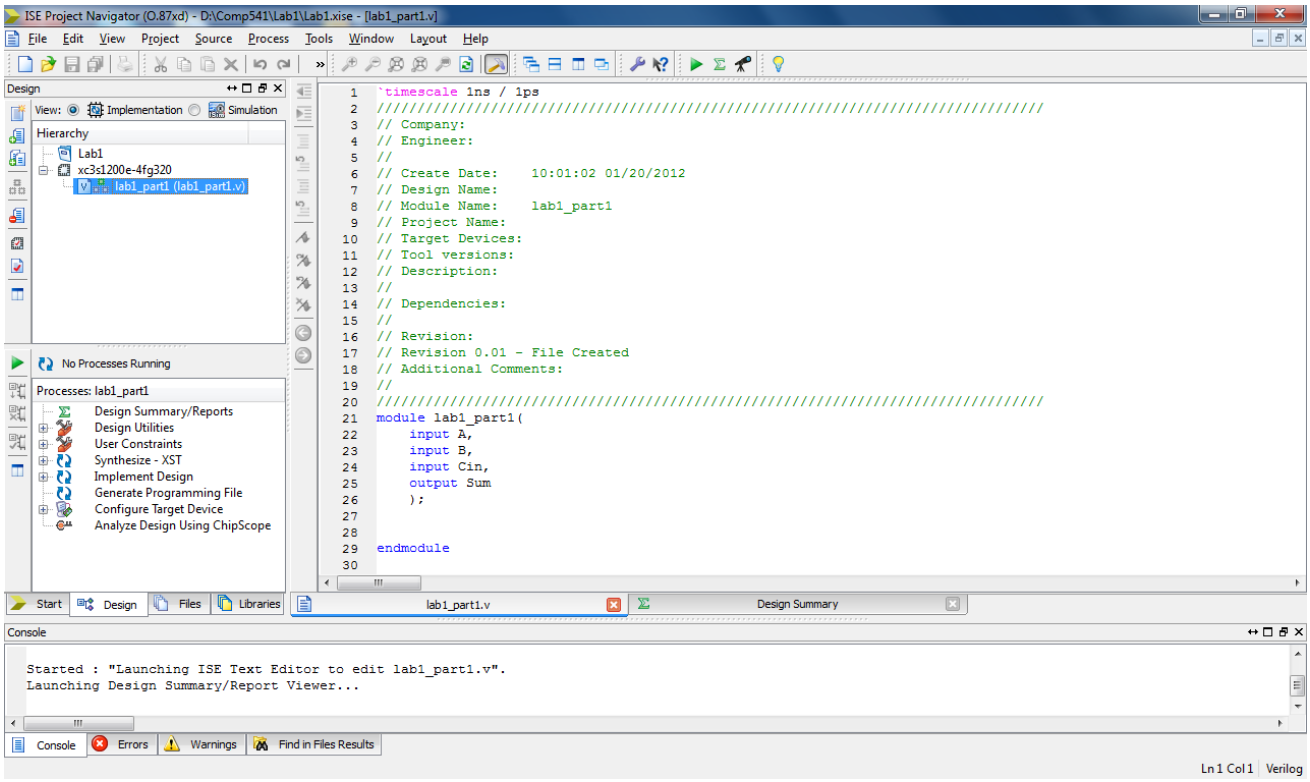
```
module lab1_part1(
    input A,
    input B,
    input Cin,
    output Sum
    );

    assign Sum = Cin ^ A ^ B;

endmodule
```

Note that '^' is the Verilog symbol for the exclusive-or (XOR) operation.




```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    10:01:02 01/20/2012
7  // Design Name:
8  // Module Name:    lab1_part1
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21 module lab1_part1(
22     input A,
23     input B,
24     input Cin,
25     output Sum
26     );
27
28       assign Sum = Cin ^ A ^ B;
29
30 endmodule
```

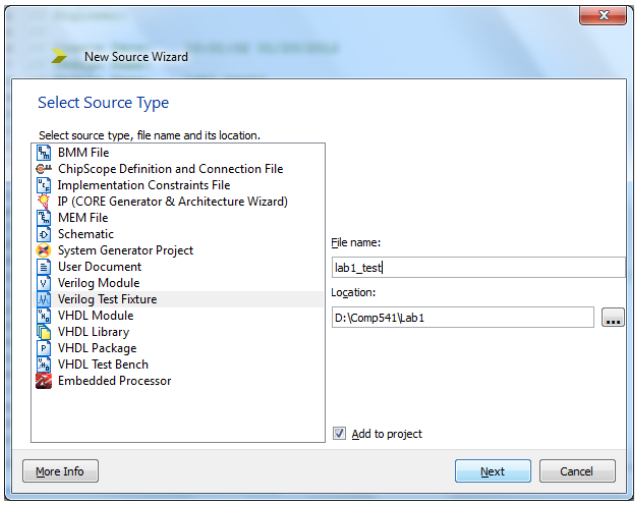
Hit *Save.*



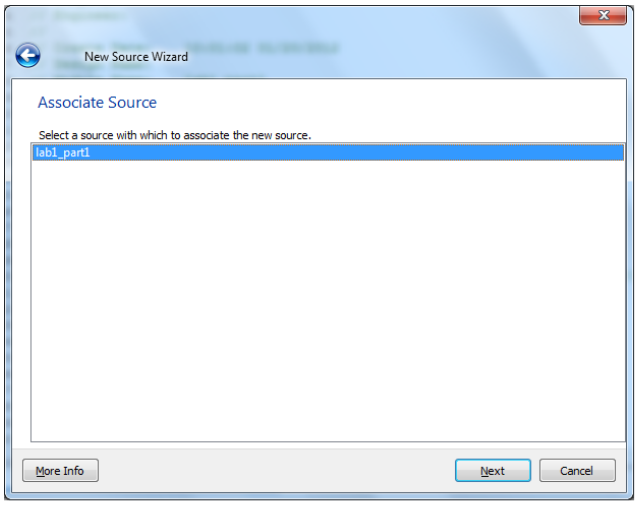

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    10:01:02 01/20/2012
7  // Design Name:
8  // Module Name:    lab1_part1
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21 module lab1_part1(
22     input A,
23     input B,
24     input Cin,
25     output Sum
26     );
27
28
29 endmodule
30
```

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 "New Source", but this time select *Verilog Test Fixture* from the left pane. Fill in the file name as shown, and click *Next.*



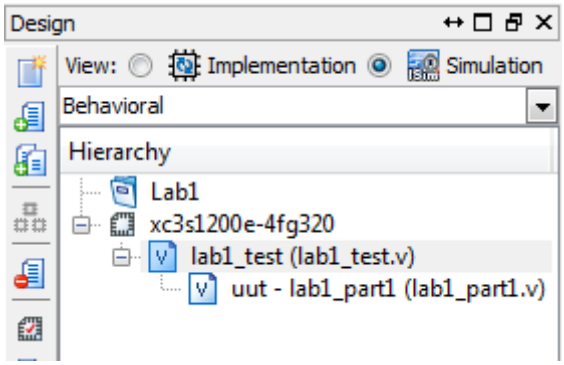
Click *Next* again.



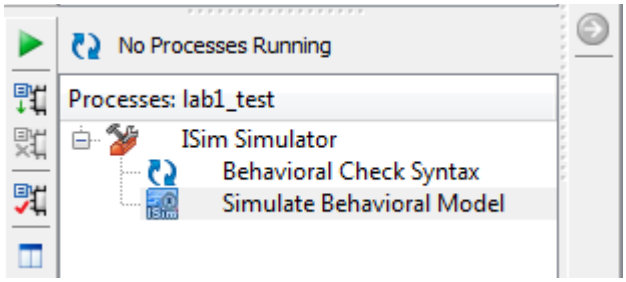
In the top left of the main window, switch to *Simulation* view. Double click on *lab1_test,* and in the editor that opens, paste in the following code within the *initial begin* and *end* block.

```
initial begin
// Initialize Inputs
A = 0;
B = 0;
Cin = 0;

// Wait, say, 10 ns before inputs start changing
#10;

// Add stimulus here
// Inputs change every 1 ns, going from 000 to 111
#1 {A, B, Cin} = 3'b001;
#1 {A, B, Cin} = 3'b010;
#1 {A, B, Cin} = 3'b011;
#1 {A, B, Cin} = 3'b100;
#1 {A, B, Cin} = 3'b101;
#1 {A, B, Cin} = 3'b110;
#1 {A, B, Cin} = 3'b111;
// Wait another 5 ns, and then finish simulation
#5 $finish;
end
```

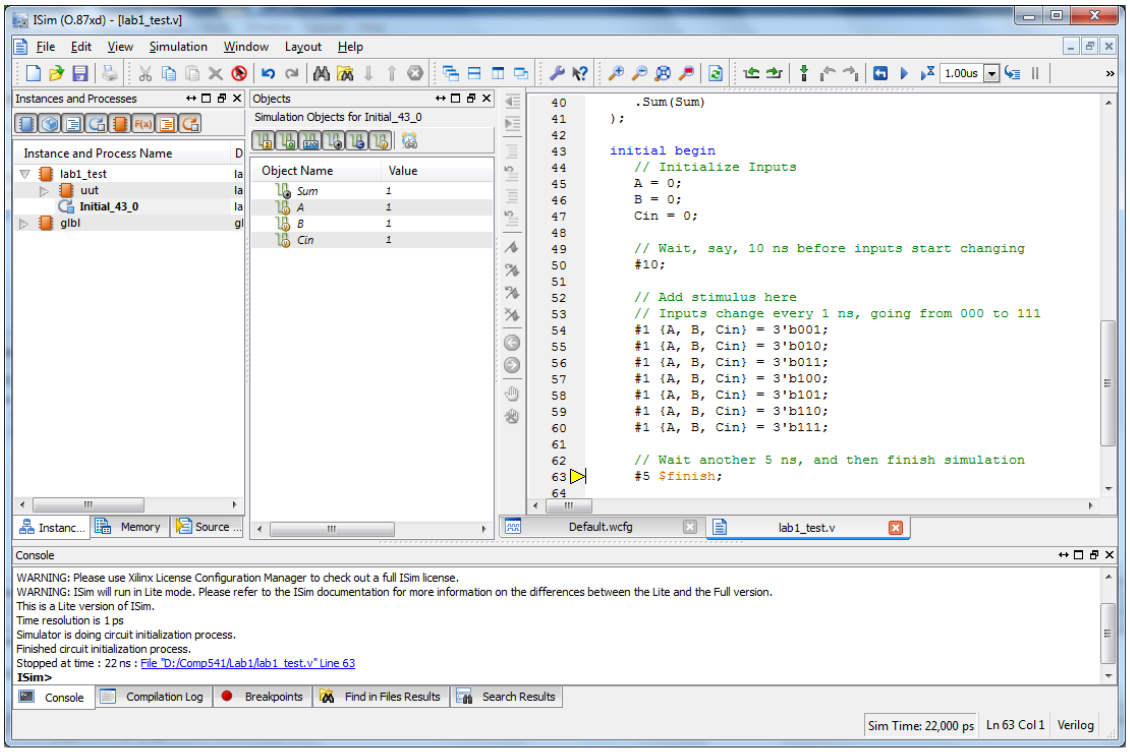
Click on *lab1_test,* make sure *Simulation* radio button is selected…



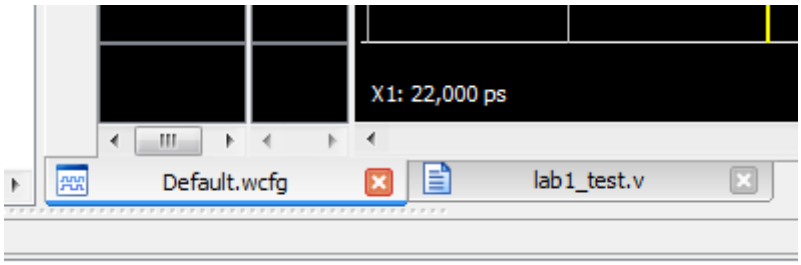
And then double-click *Simulate Behavioral Model* in the pane below.



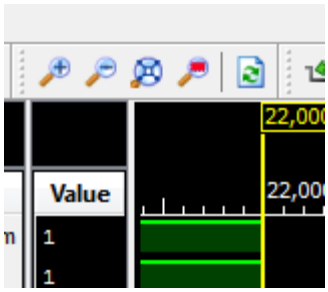
The simulator *ISim* will launch:



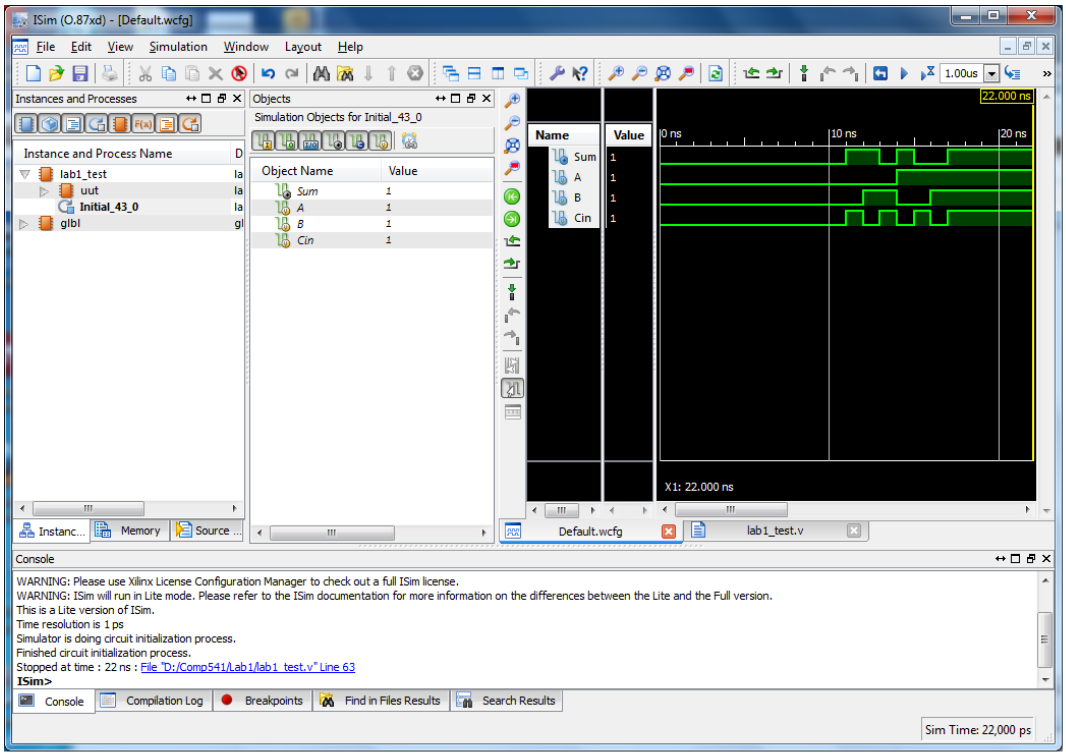
Click on the waveform window (*Default.wcfg*).

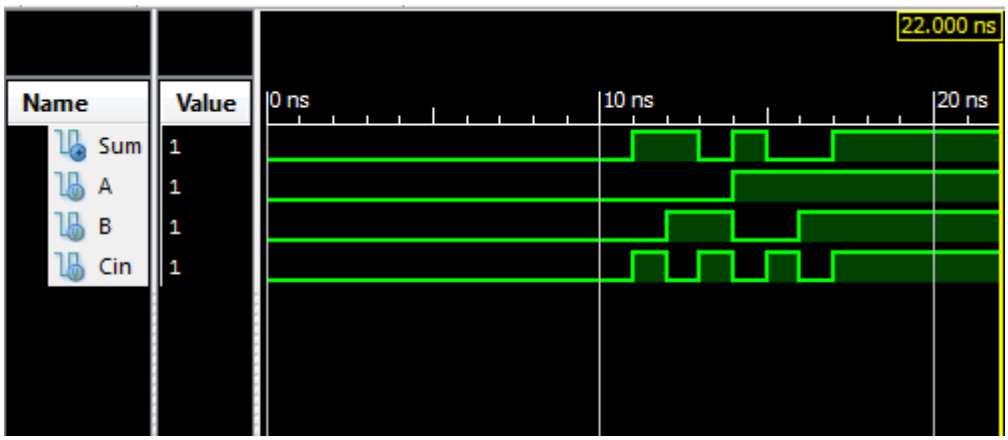


Click on the “Zoom to fit” icon.



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





This is the end of Part I!

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 $Carry_{out}$.

For a complete list of Verilog operators, please refer to the online Verilog reference (linked from the course website), Section 11.

Rename the module to **`lab1_part2.`** It will look as follows (fill in the blanks):

```
module lab1_part2(
   input A,
   input B,
   input Cin,
   output Sum,
   output …
   );

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

endmodule
```

Follow all of the steps as in Part I to perform a simulation, and observe the waveforms. Make sure the carry output is correct, along with the sum.

What to submit: **A screenshot of the *ISim* window for Part II only, clearly showing the 3 input and 2 output waveforms.**

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

- **Send email to: `comp541submit-cs@cs.unc.edu`**
- **Attach the simulator screenshot using the filename `waveforms.png` (or other appropriate extension)**
- **Submit your work by 11:59pm on Wednesday, August 27.**
