LAB 3

Xilinx Intro, Half Adder, Full Adder, and Test Fixtures

**Due:**

**PROJECT DESCRIPTION**

This lab consists of three parts:

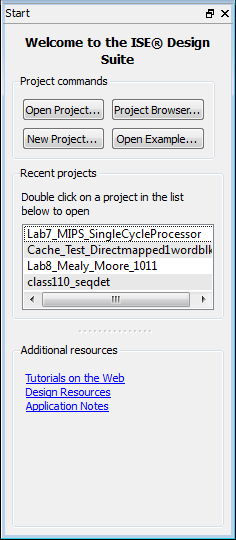
1. Xilinx Project Creation Guide-----(page 1)
2. Define and test a Half Adder-----(page 5)
3. Define and test a Full Adder-----(page 13)

# Part 1: Xilinx Project Creation Guide

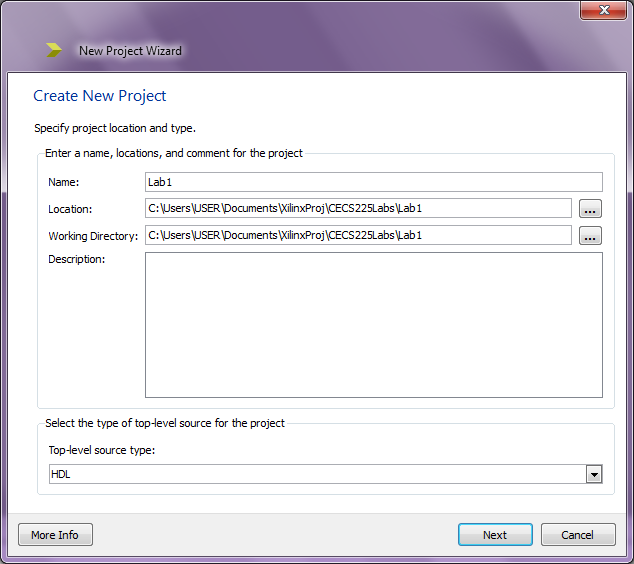
**Step 1:** **Open Xilinx 14.7!** Go to the **start** menu and find then Xilinx Design Tools folder. Then select one of the Project Navigator executables as shown on the right:

Once you click on the Project Navigator, a splash screen will appear for some time while Xilinx starts up.

**Step 2: Create a new project.** If the project creation wizard doesn’t appear automatically then click the New Project button shown in the image below:



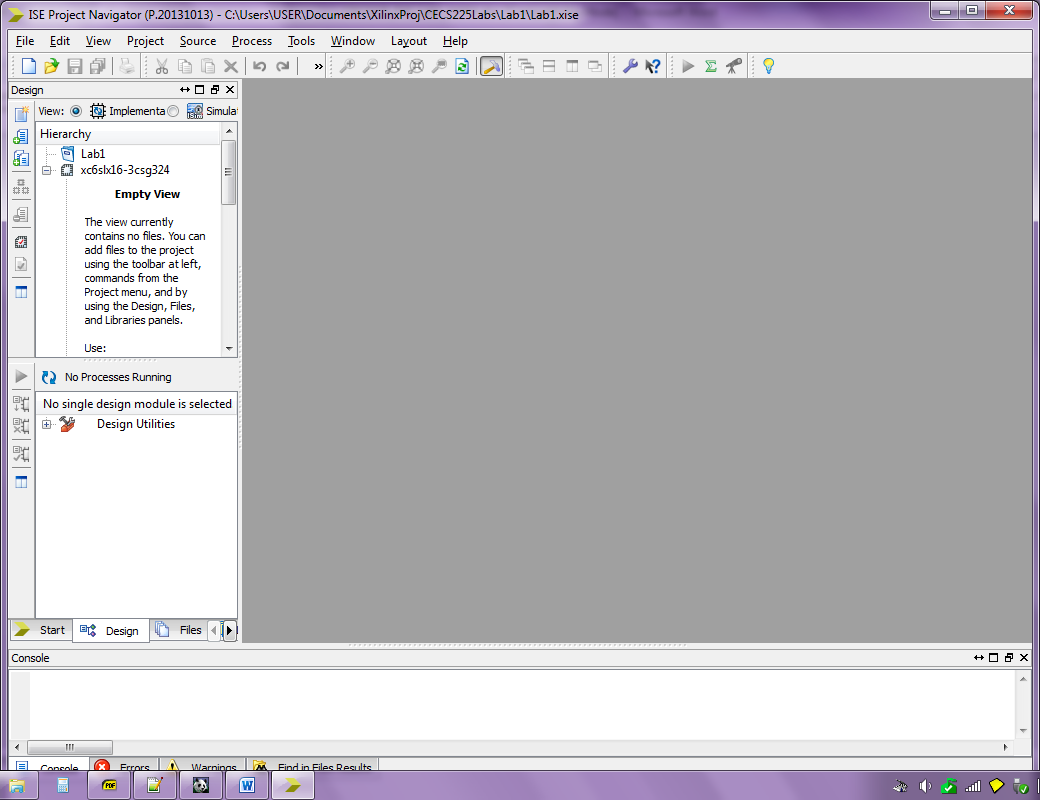
If you are working on a CECS Lab computer, you must create your project in an unrestricted directory. The Documents folder works well for this purpose. Make a sub folder for your project and give it a meaningful name like shown below and then click the **Next** button:

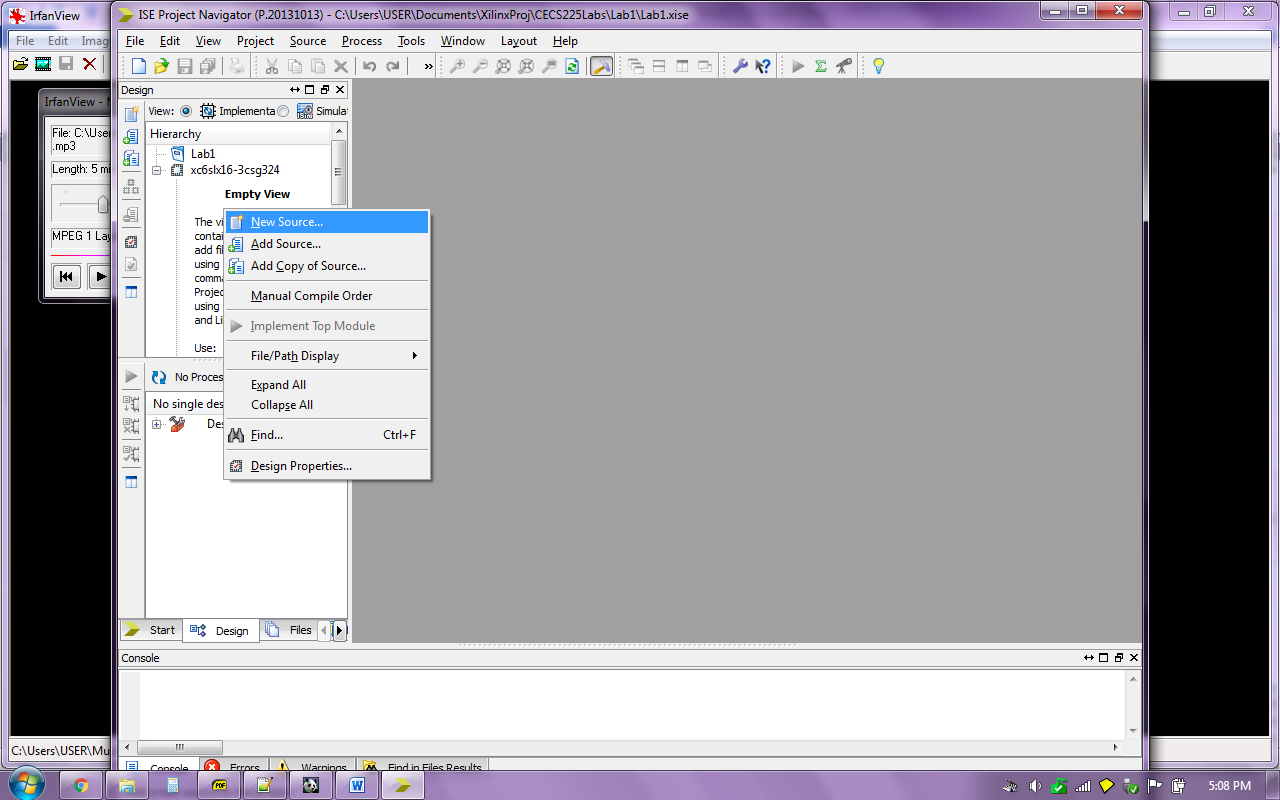


A window entitled Project Settings will appear with configurations relevant for loading a project onto a hardware test platform. All projects in this course are simulation only so click the Next button.

Finally a window entitled Project Summary will appear. Click the **Finish** button.

If all went well you will have created an empty project like shown in the image below:



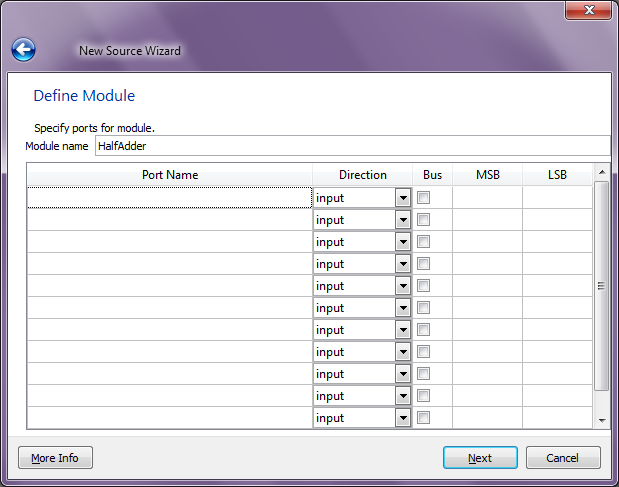


**Step 3: Create a Half Adder Verilog module.** Right click in the background of the Hierarchy window pane and select the option named **New Source…**

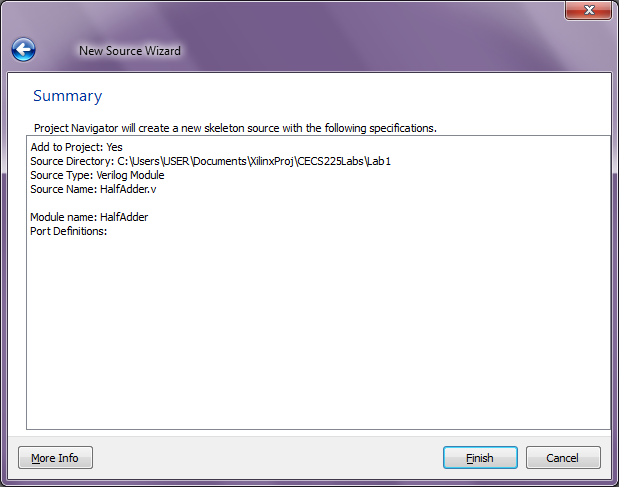
A **New Source Wizard** window will appear. In **Select Source Type** choose **Verilog Module**, enter **HalfAdder** as the File Name, and click the **Next** button



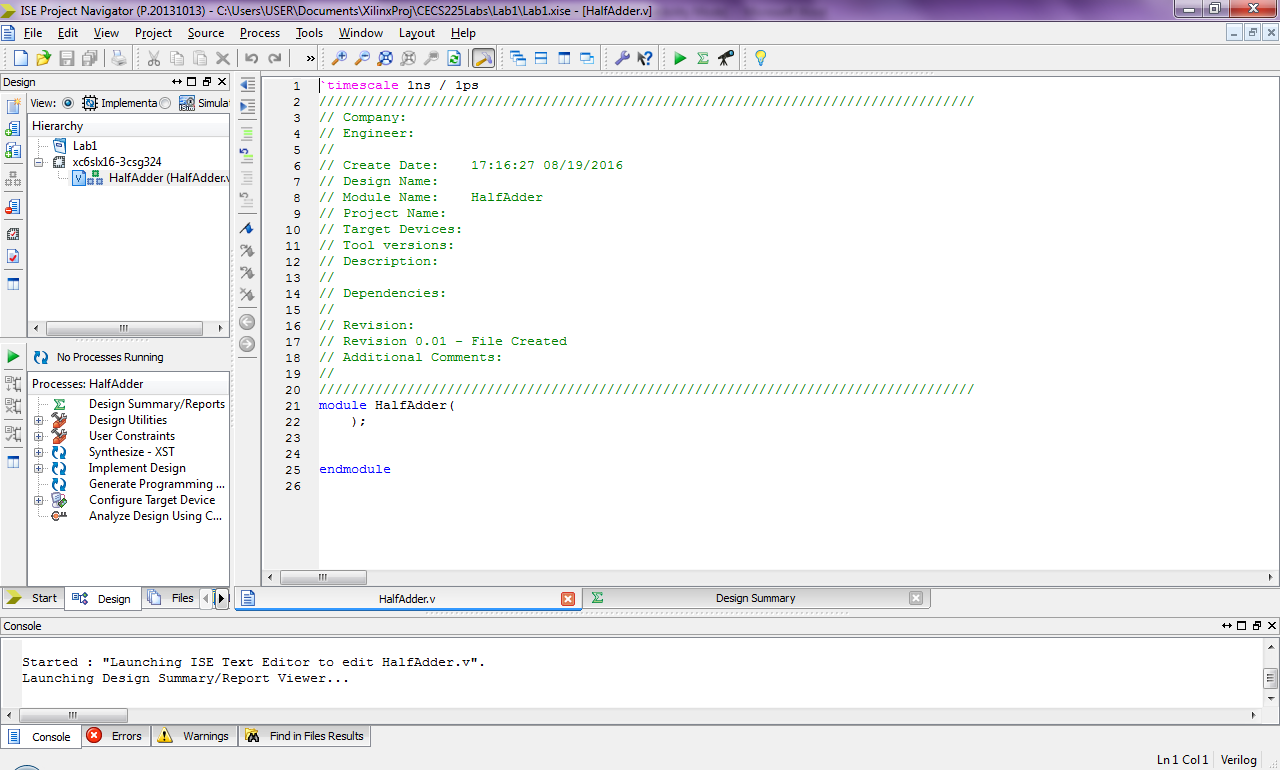
On the next screen, skip the option to “Define Module” clicking the **Next** button



A Summary window will appear, click the **Finish** button:



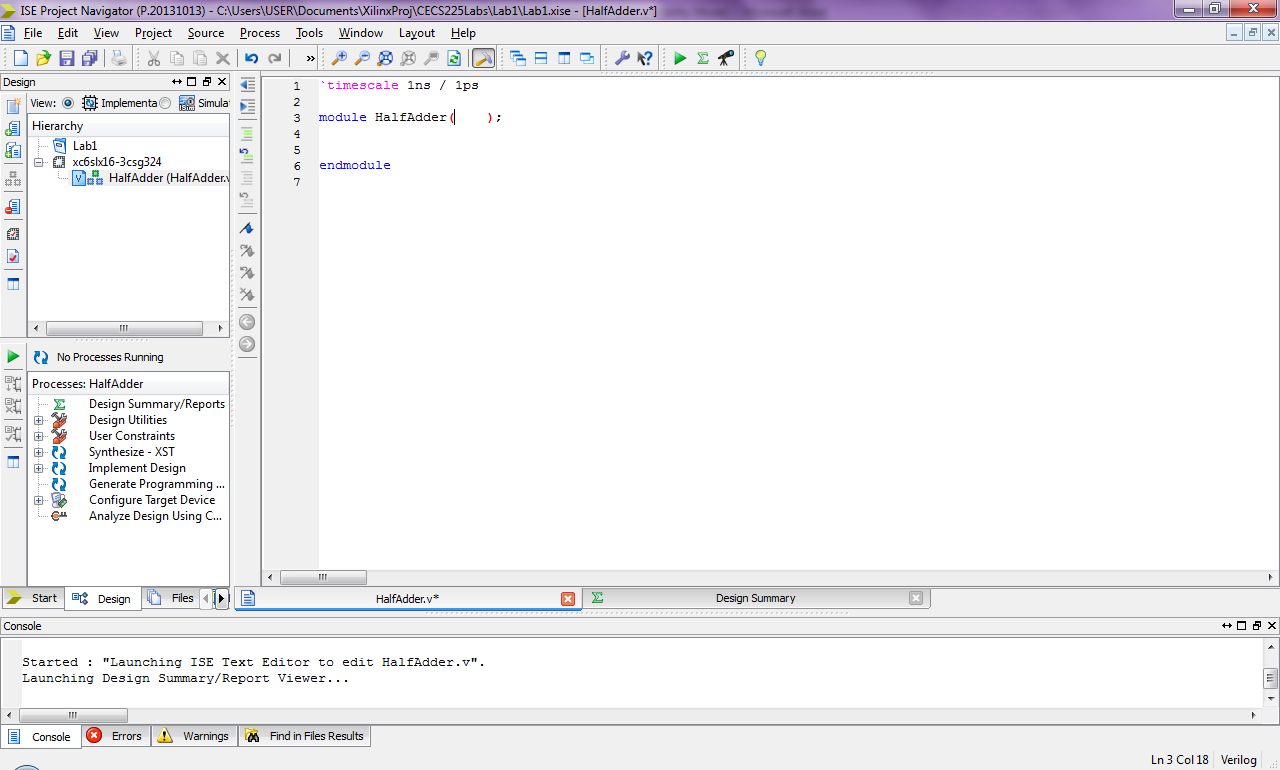
An empty source file will be created and you are ready to fill it in to make the Half Adder. Welcome to the world of Xilinx!



# Part 2: Define and Test a Half Adder

*(Read Chapter 5 pages 239 and 240 for an explanation of the half adder.)*

**Step 1:** Delete all the automatically generated comments as shown below:



**Step 2:** The Half Adder is a digital building block with 2 inputs (A,B) and 2 outputs (S,Cout). This information will be included first in the Half Adder definition:

Port list: All inputs and outputs go here separated by commas



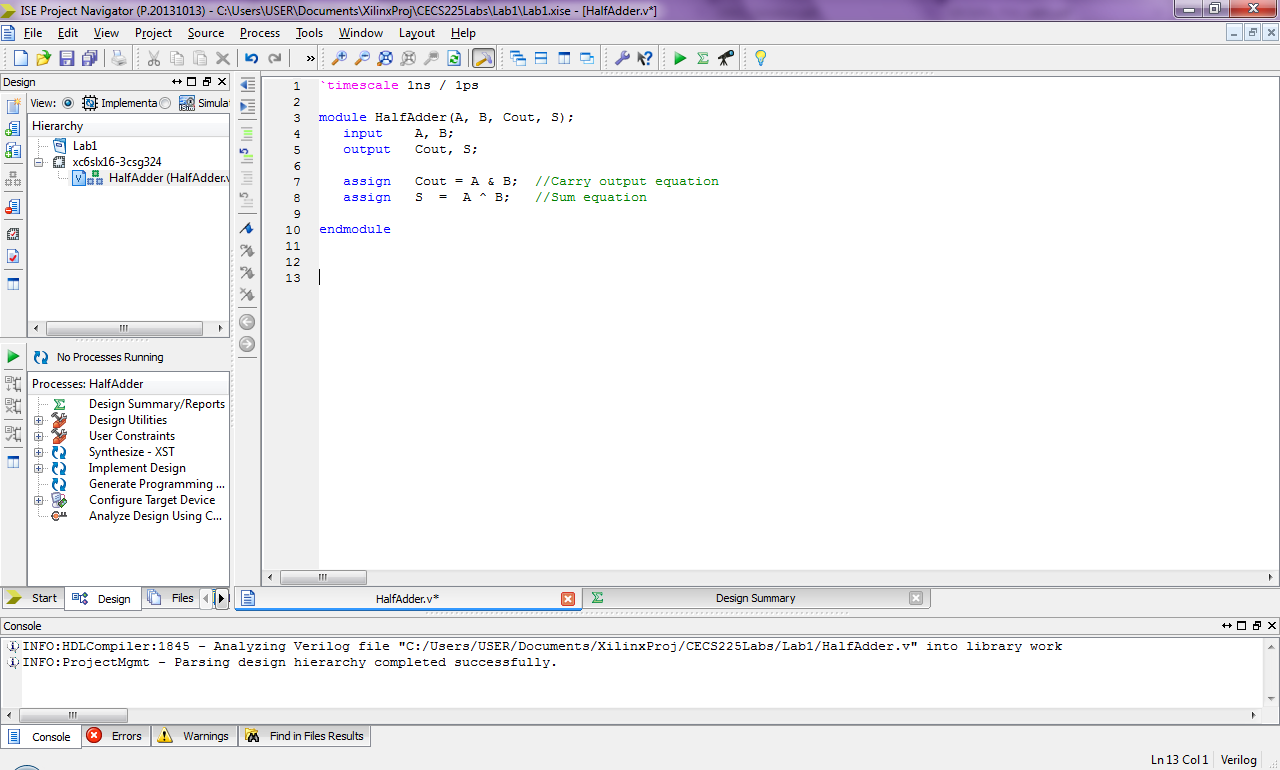
Port direction:

Signal variables must be declared as input or output

**Step 3:** The Half Adder logic must be modeled next. The circuit below shows the Half Adder logic circuit and the equivalent boolean equations:

|  |  |
| --- | --- |
| out | **S = A xor B**  **Cout = A and B** |

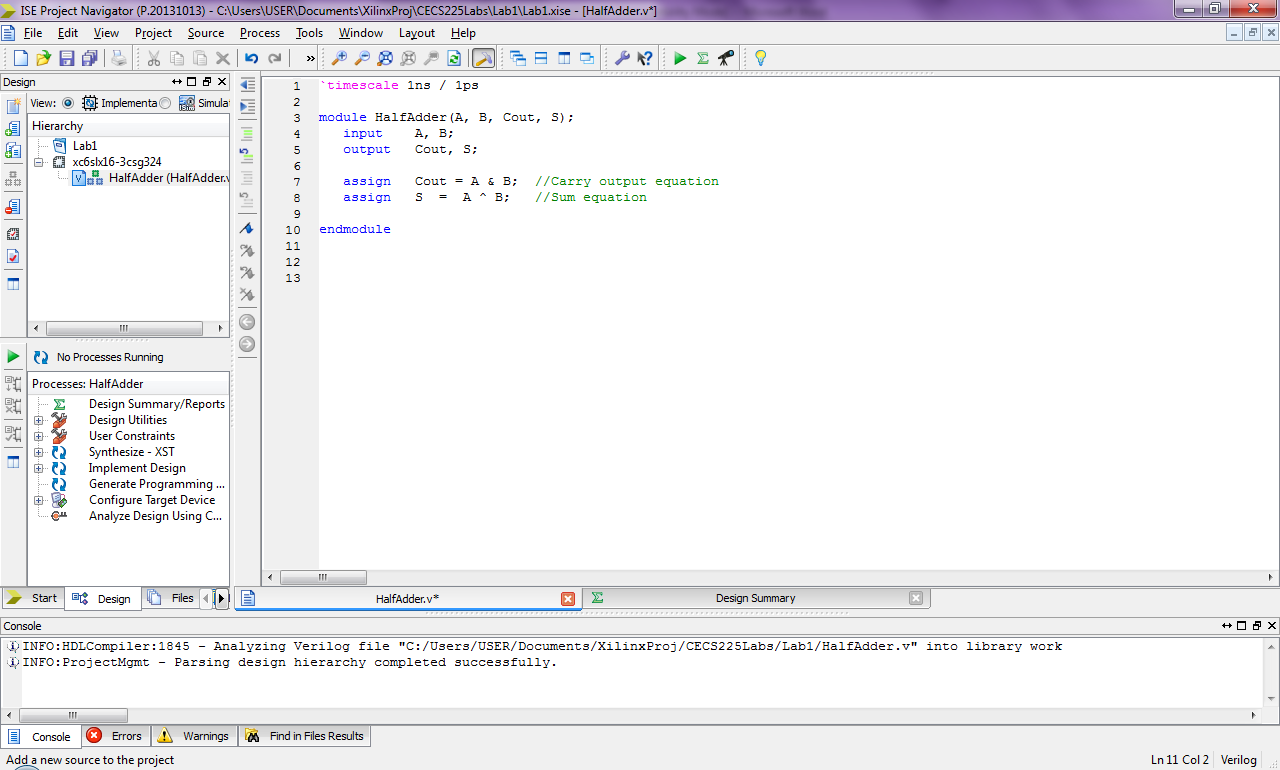
The two logic equations must be modeled in Verilog syntax as shown below:

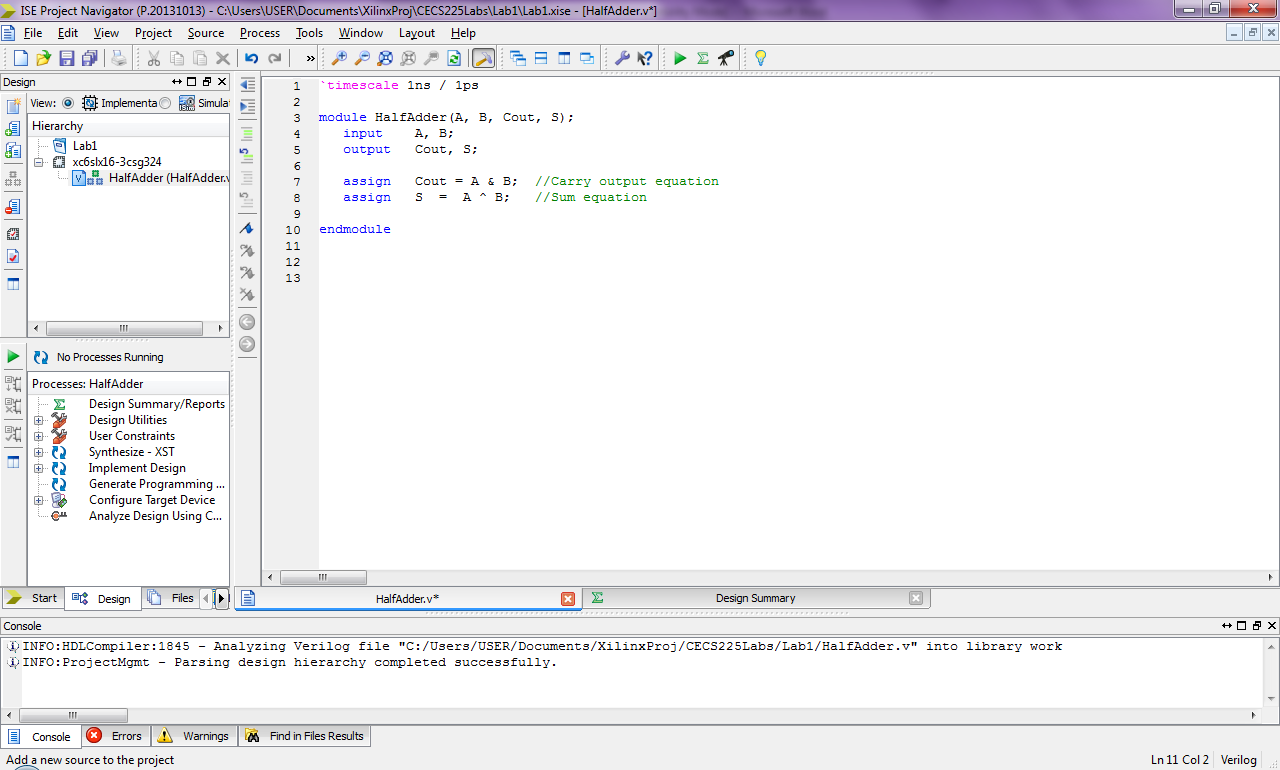


This completes the verilog module definition of the half adder. Next it must be tested to ensure it works correctly.

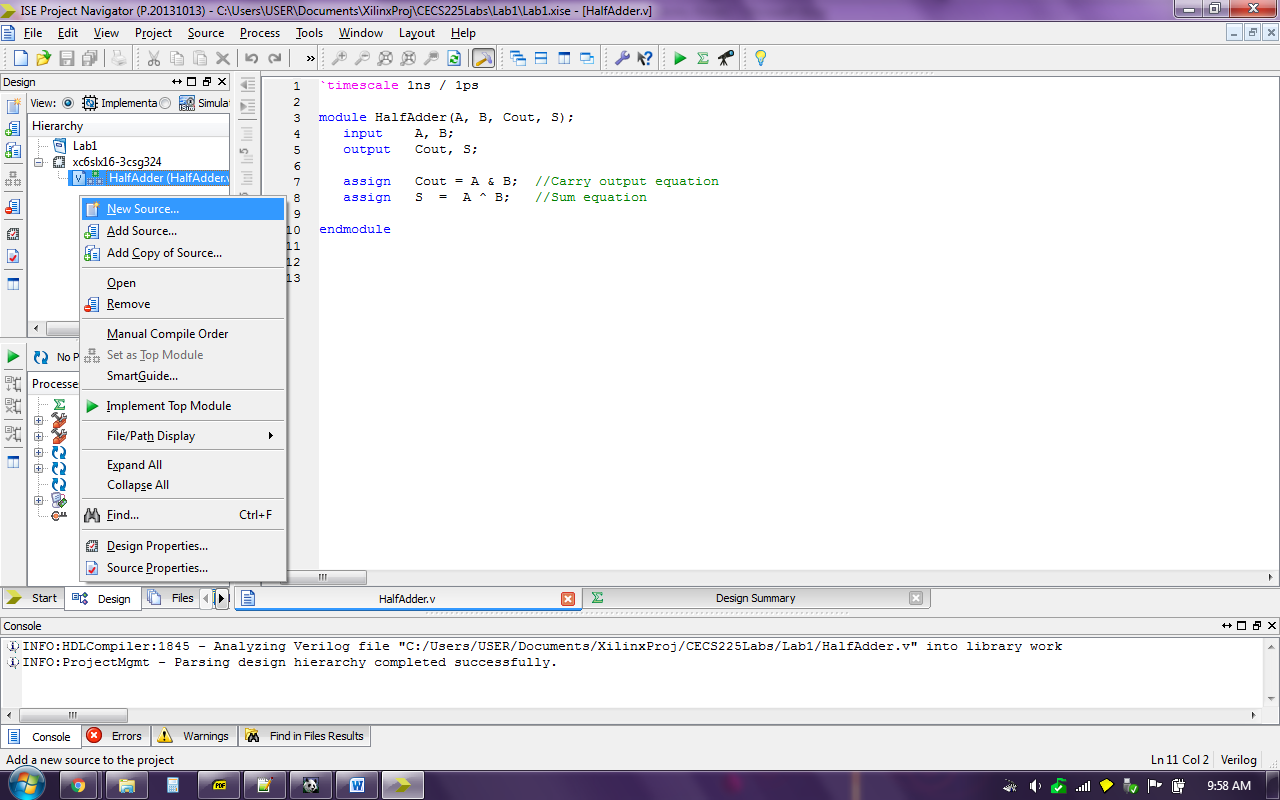
**Step 4:** **Create a Half Adder Verilog Test Fixture.** To test a module for correct functionality, a set of inputs will be provided to produce an expected set of outputs. A **Verilog Test Fixture** is used to test a Verilog source module.

**NOTE:** *Before making the Verilog test fixture, ensure you have saved your Verilog module. An unsaved Verilog module will show an asterisk next to the file name in the window title bar and in a tab at the bottom of the file definition as shown in the images below.*

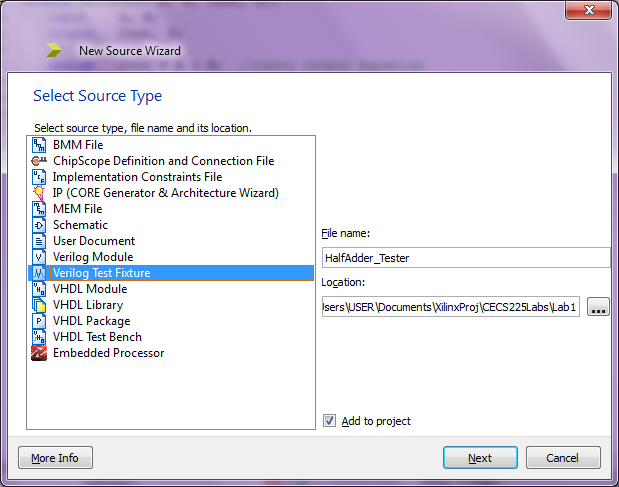


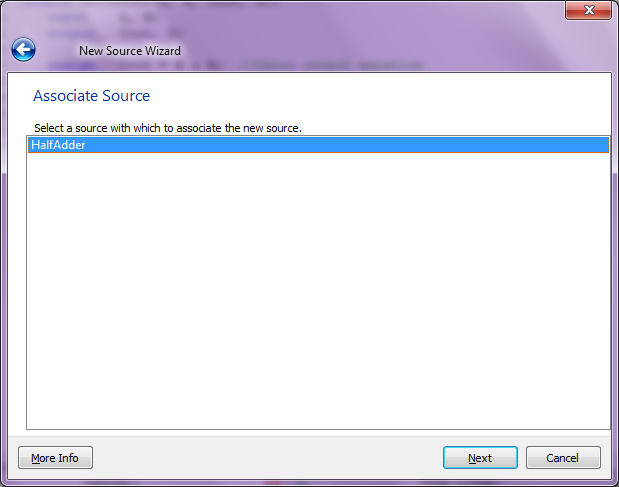


Once you have saved your Verilog source file, right click on the file name in the window pane on the left and select **New Source**:



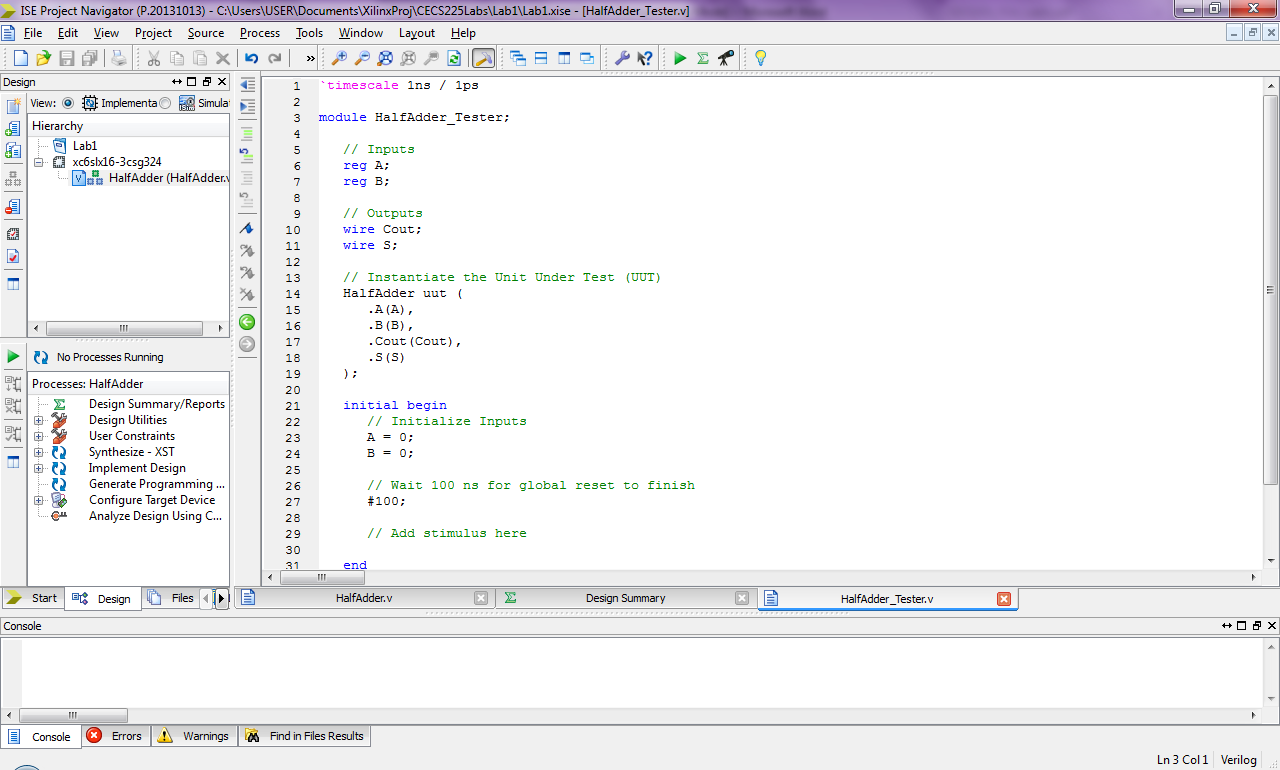
The New Source Wizard will appear again. Select **Verilog Test Fixture** in the Select Source Type box and name it **HalfAdder\_Tester** as shown in the image below. Then click the **Next** button.



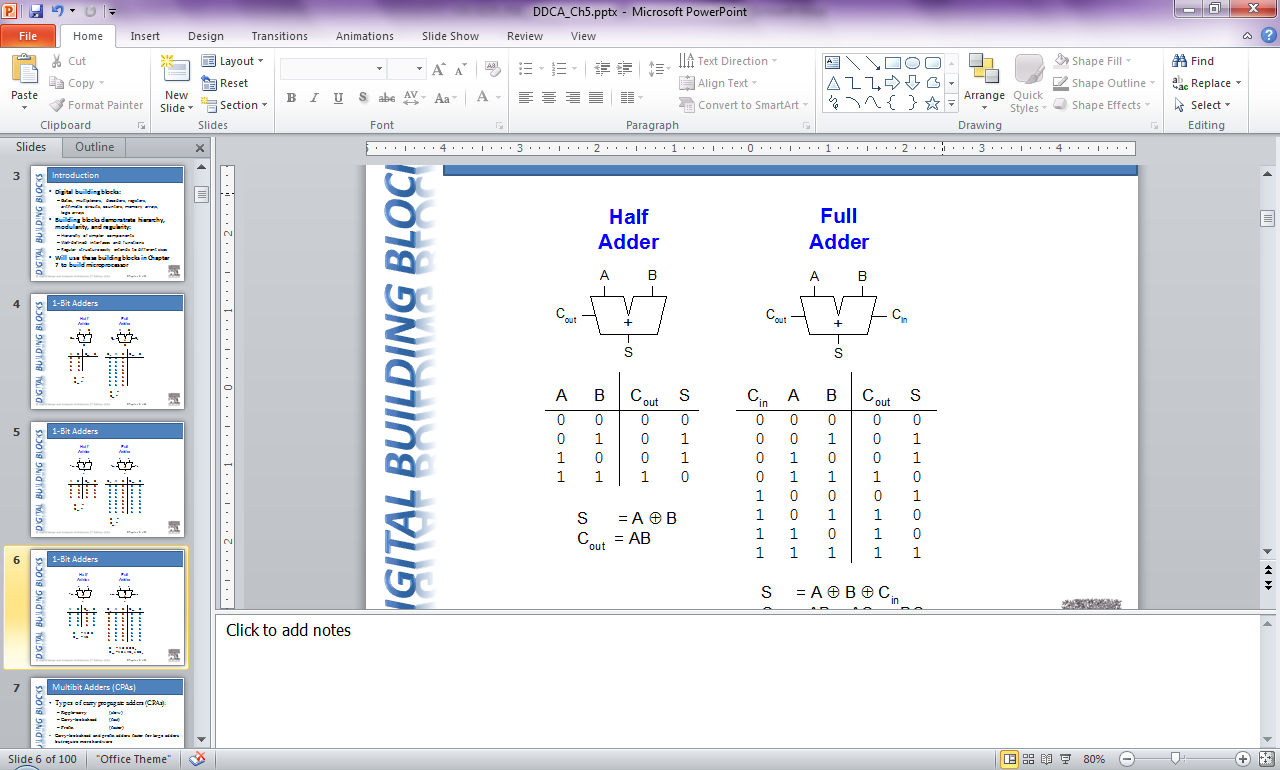
A window named **Associate Source** will appear where you will select a Verilog source module to be tested. HalfAdder is the only choice for now. Once we have a project with multiple Verilog source modules it will be important to select the correct source. Click the **Next** button to proceed.

A **Summary** window will appear. Click the **Finish** button.

A Verilog Test Fixture skeleton will be made. Delete the auto-generated comments section at the beginning of the file and your source should look like what is shown below:



**Step 5:** **Create the Half Adder test script.** To test a module for correct functionality, a set of inputs will be provided to produce an expected set of outputs. For simple modules like the half adder a truth table is used to show the outputs that can be expected from a set of inputs.

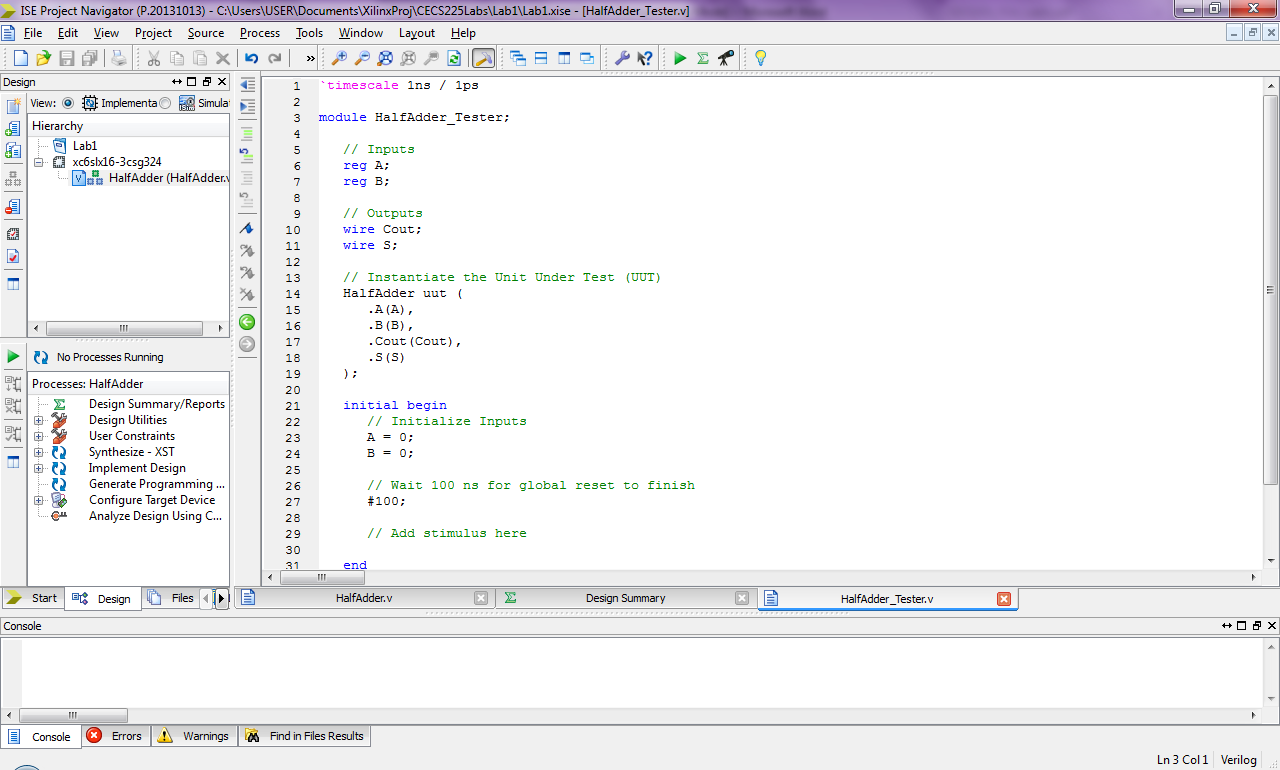


The Cout output column shows that Cout equals 1 only when A equals 1 and B equals 1.

The S output column shows that S equals 1 when the value of A is not equal to the value of B.

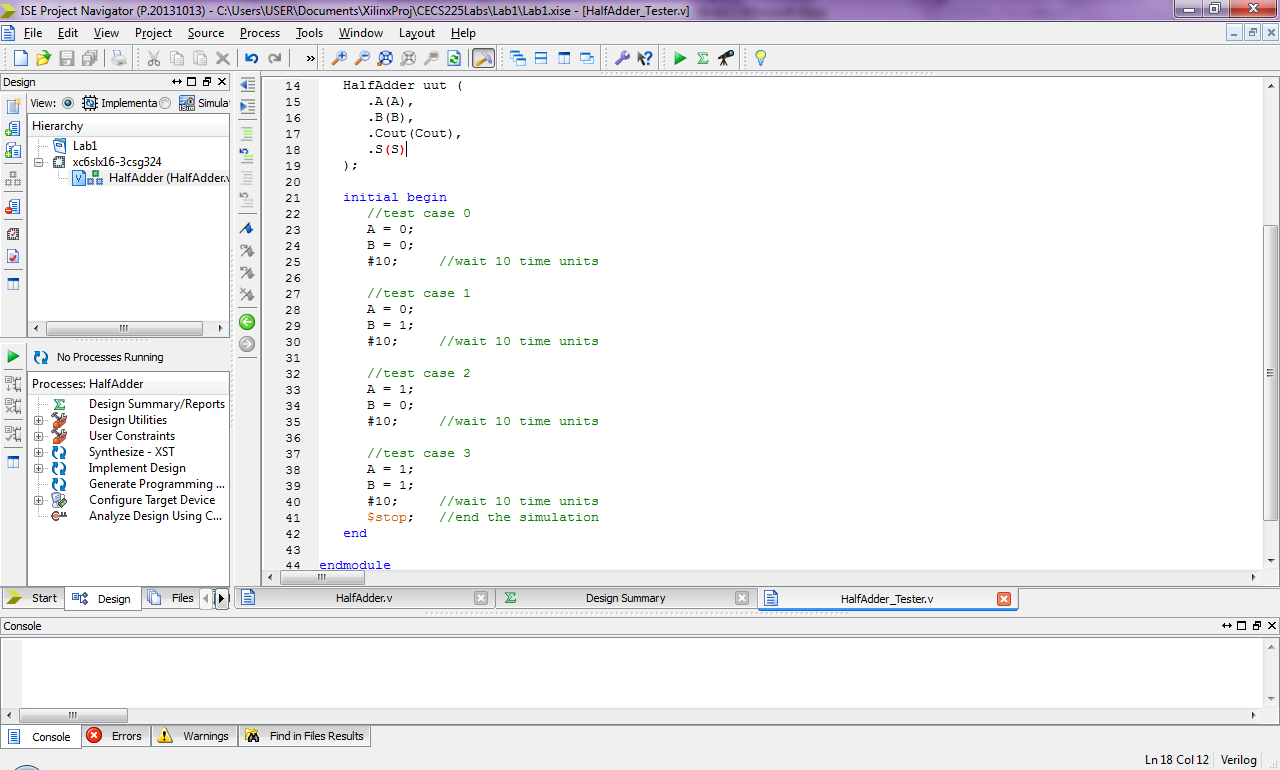
Each row of the truth table can be thought of as a test case. Four test cases are needed to completely test the half adder. These test cases are defined in Verilog on the next page.

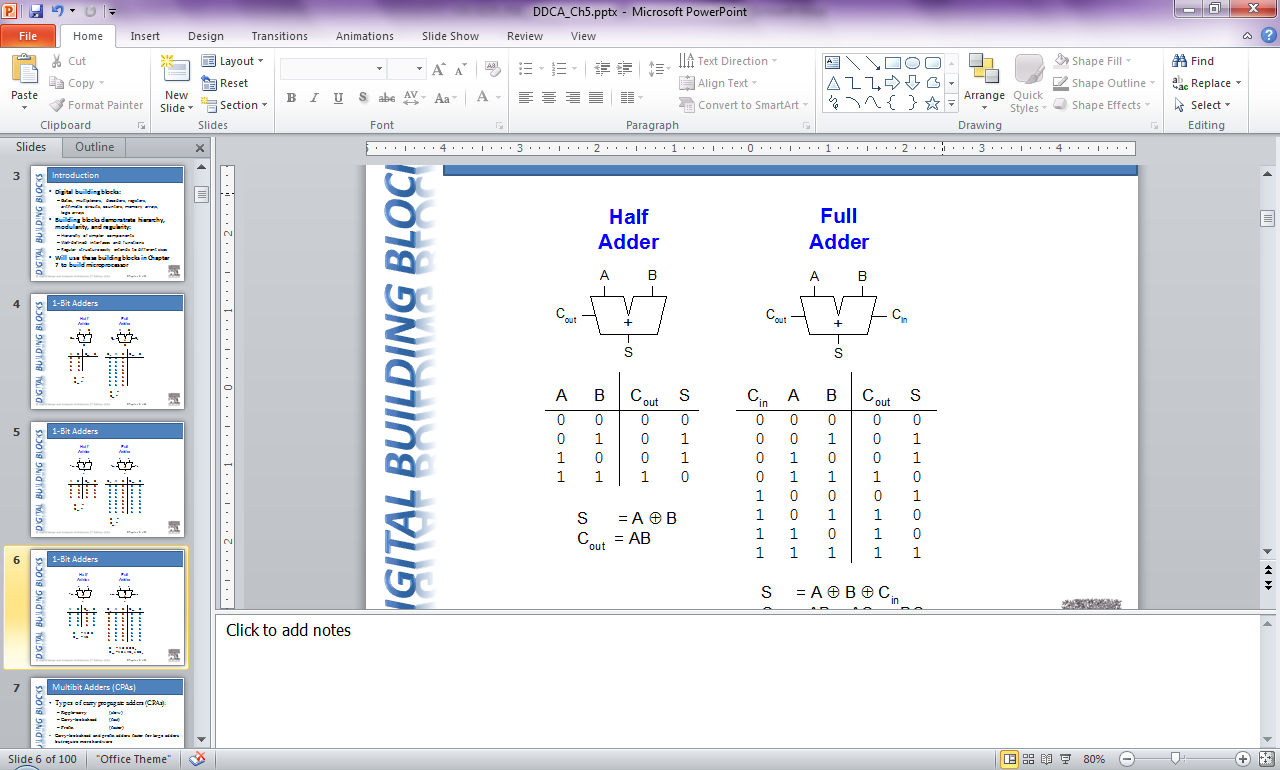
The section of the Verilog Test Fixture called the **Initial Code Block** shown below will contain the test cases. The keyword **initial** identifies code that will run one time in the very beginning of the Test Fixture’s execution. The **begin** keyword and **end** keyword encapsulates a block of code.



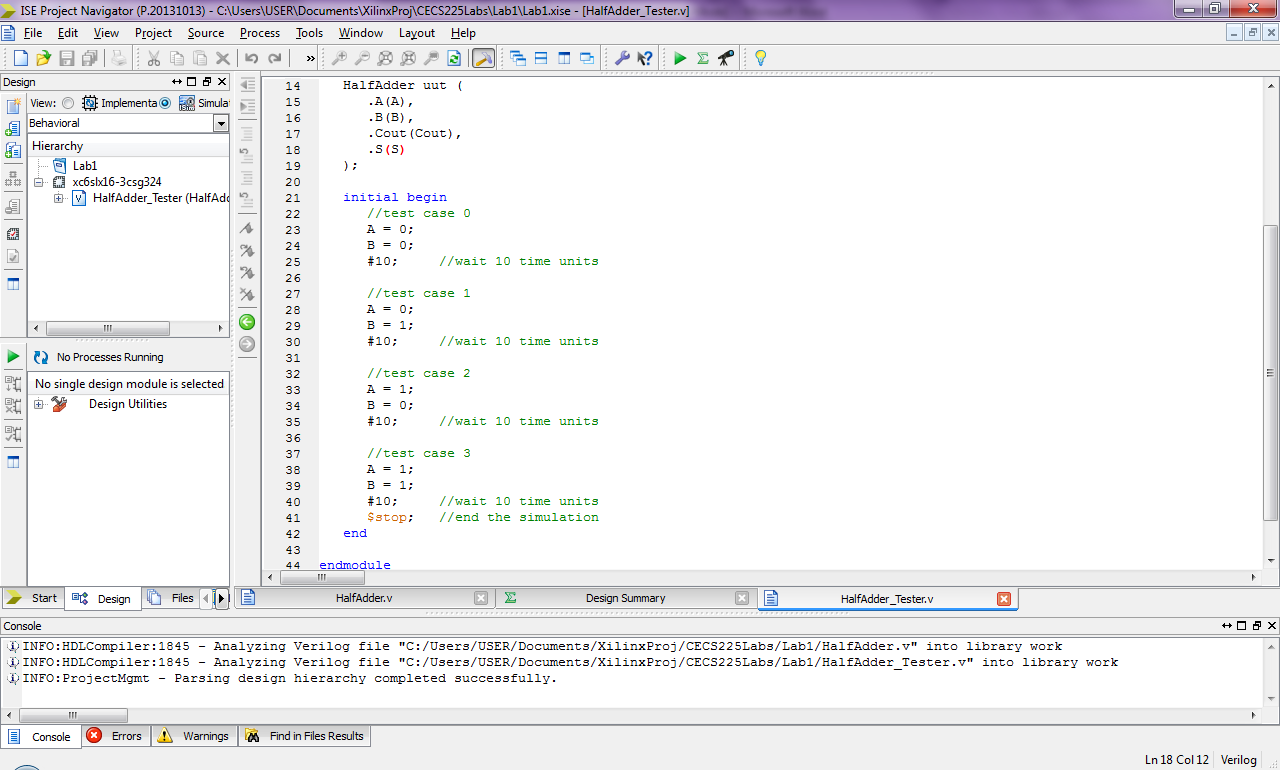
Skeleton code to be replaced with Half Adder test cases

Translate the truth table into test cases in verilog according to the diagram below:

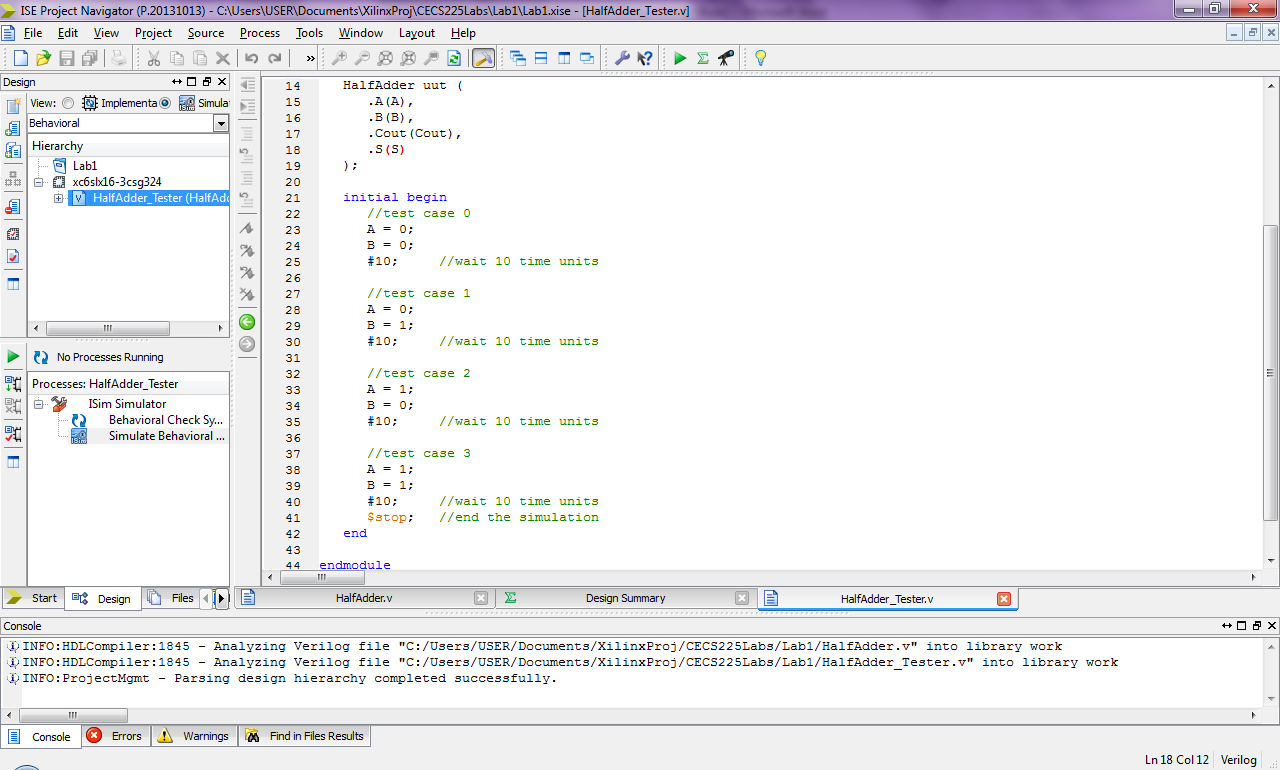




***Once the above code has been entered, save your Verilog test fixture!***



**Step 6: Run a simulation to verify correct operation of Half Adder.** Click on **Simulation** radio button to show Verilog test fixture source file in the Design window pane.

Click on the **HalfAdder\_Tester** source file within the Design window pane.

In the window pane below an option will appear named **ISim Simulator**.

Expand the **ISim Simulator** item to display the options **Behavioral Check Syntax** and **Simulate Behavioral Model**.

Double click **Simulate Behavioral Model** to launch the simulator.

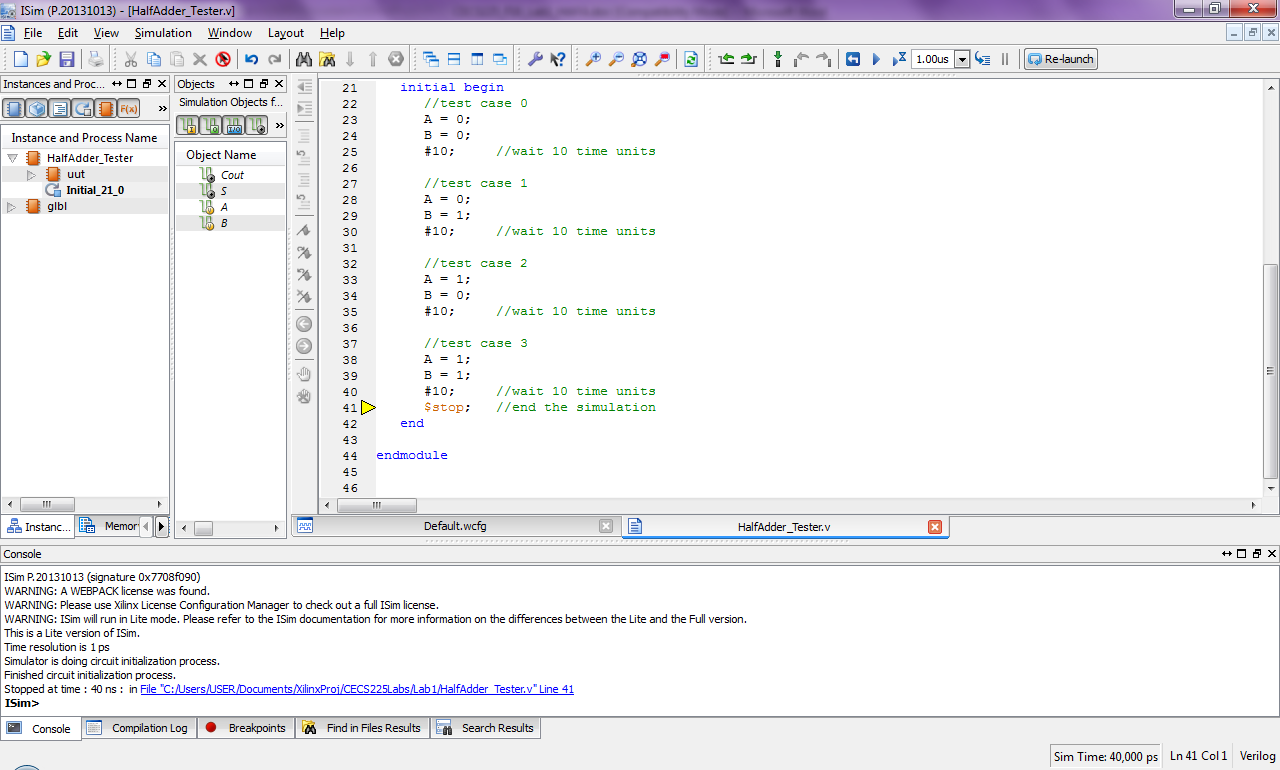
If you project contains no errors,

a simulation interface window will

appear as shown below with a yellow arrow

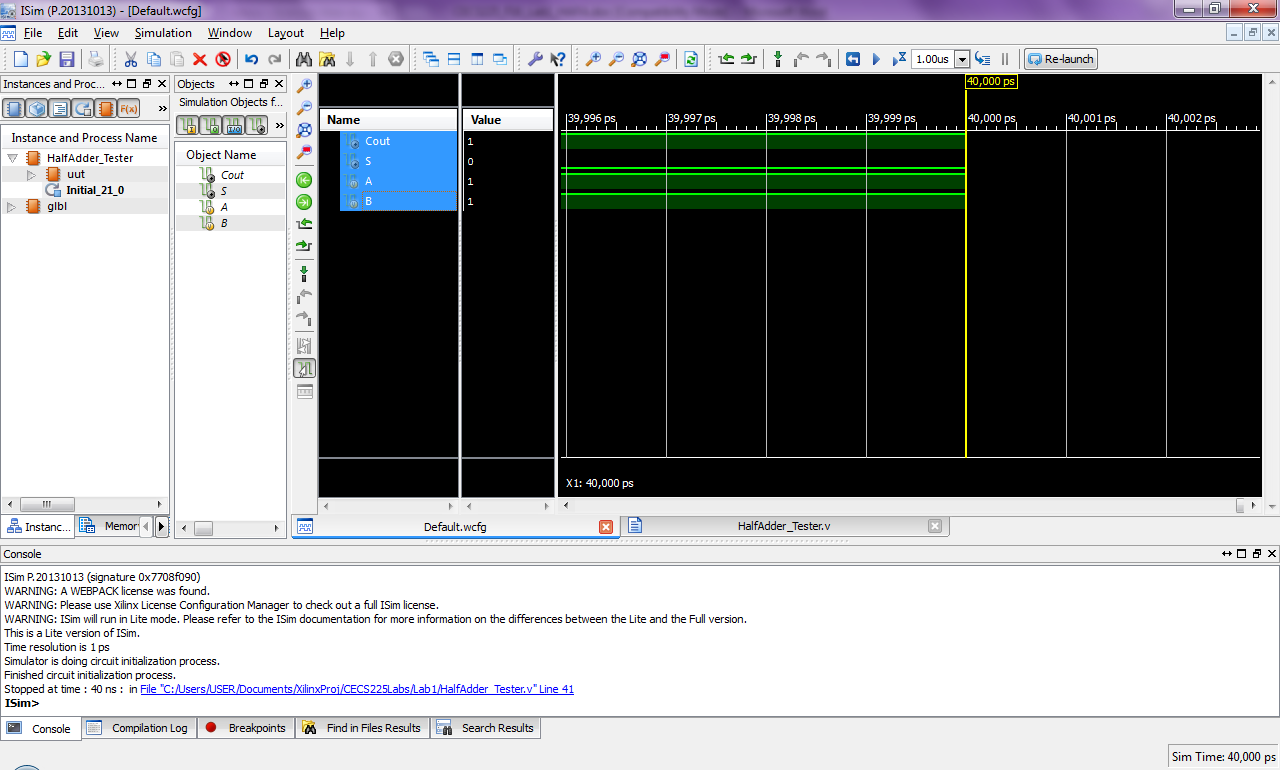
pointing to the line of Verilog code where the

simulation has been stopped.

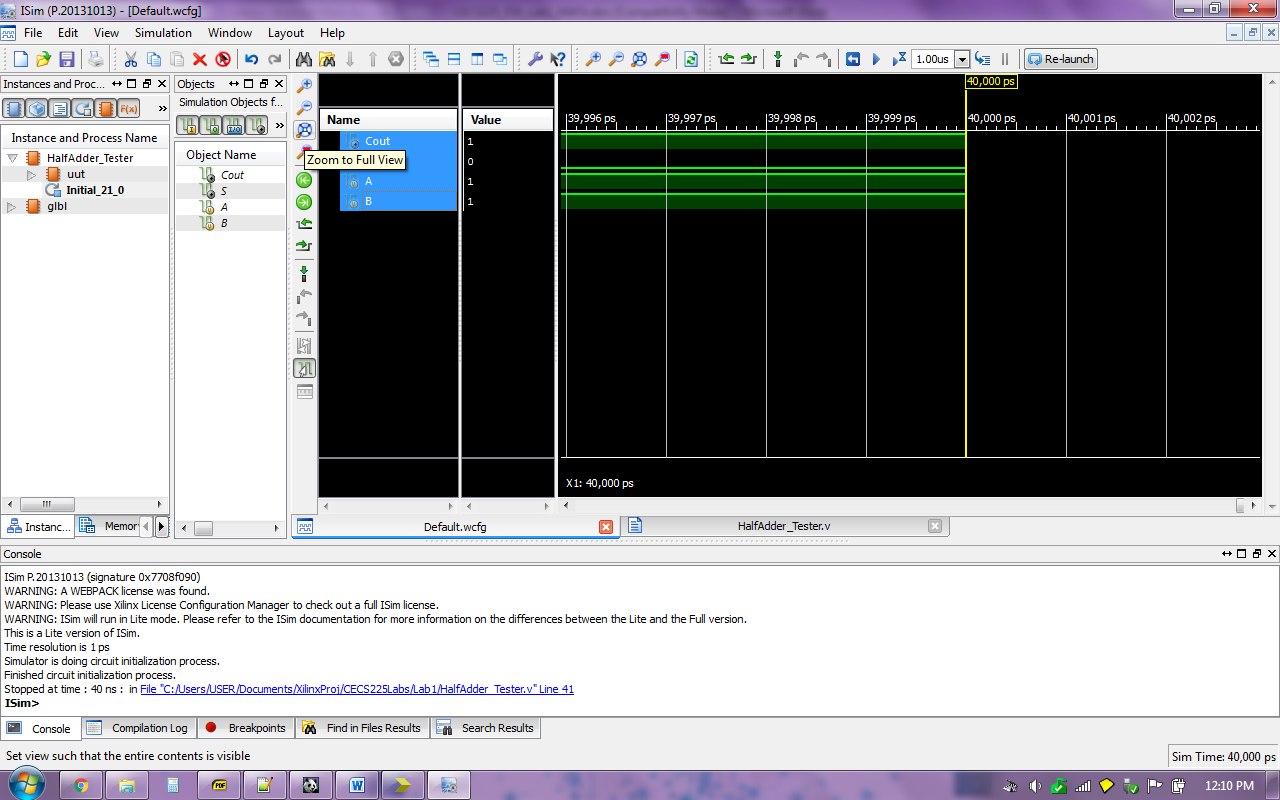


Simulation results are shown as waveforms. Steps to view and interpret simulation results are outlined on the next page.

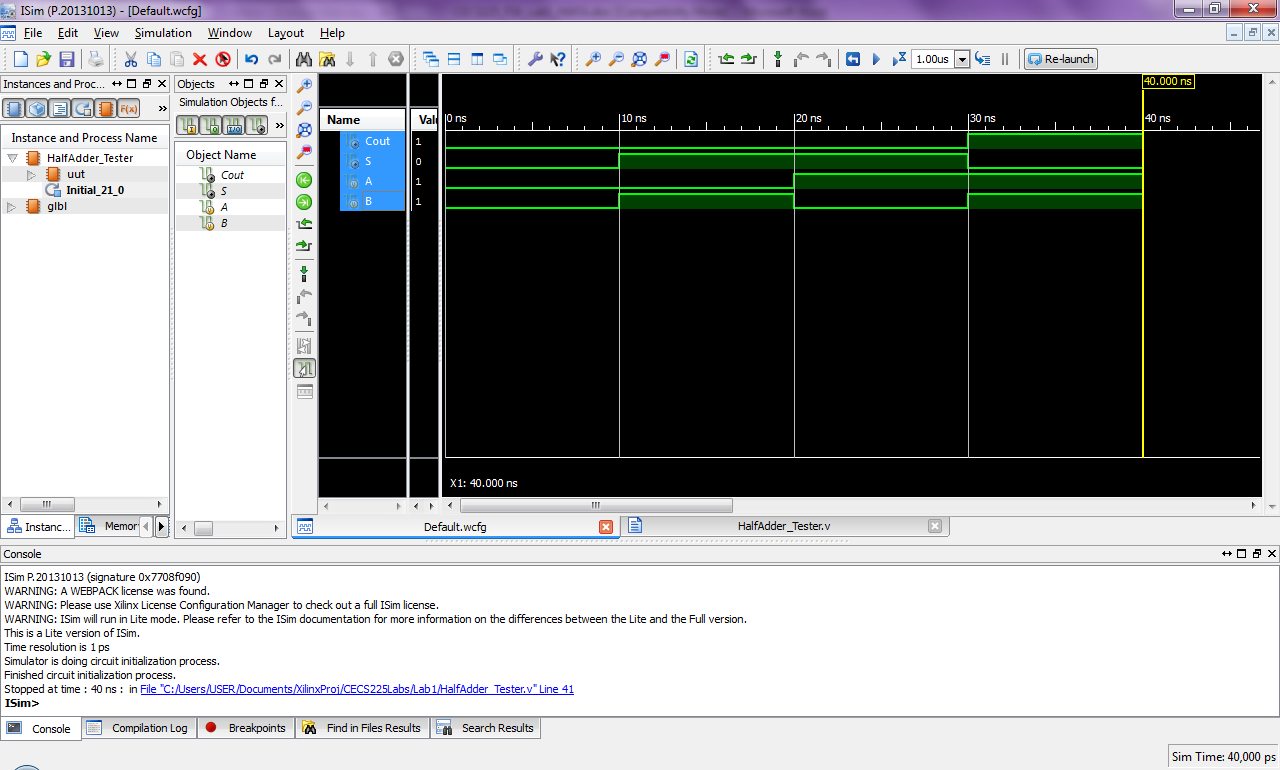
Click on the tab labeled **Default.wcfg** and simulation waveforms will be shown.



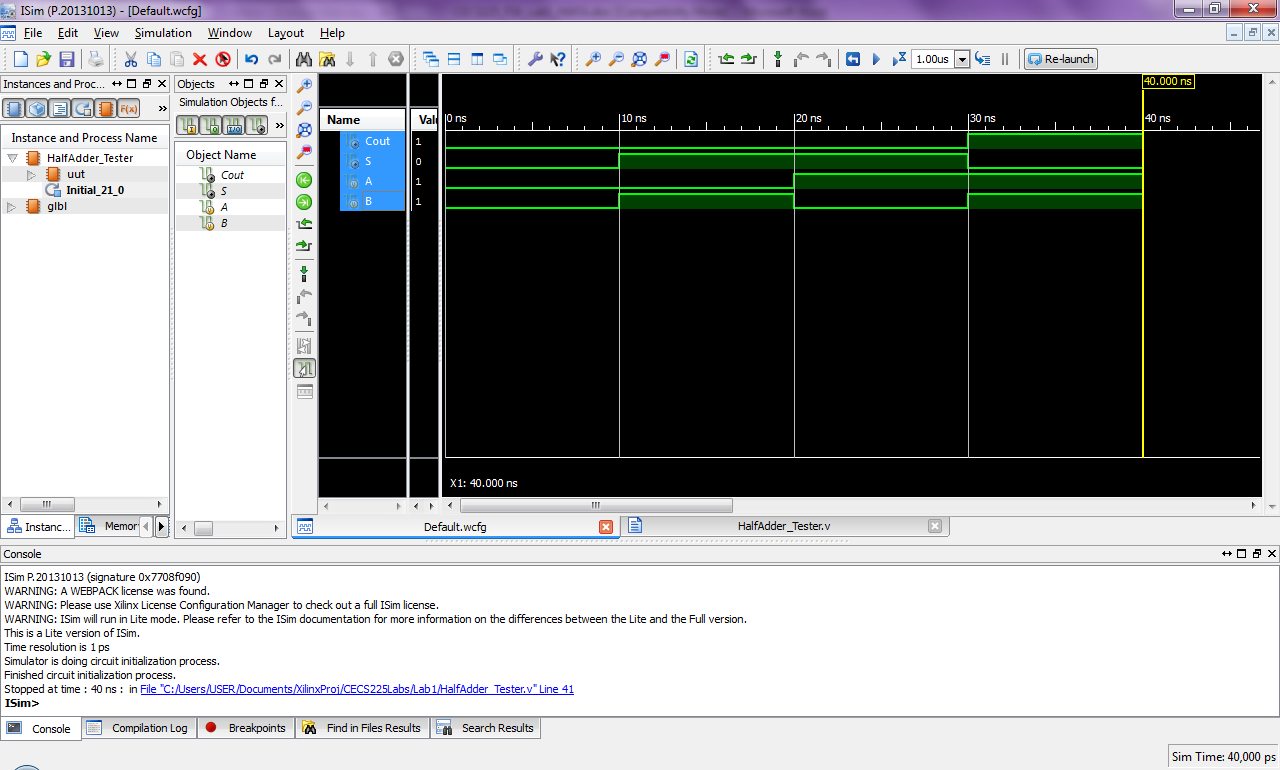
Not much information can be gathered from this section of the waveform because only a very small window of time is being shown. To view the entire simulation, click the icon of a magnifying glass with the four arrows.



Now simulation waveforms for all of the test cases will be shown.



Next the waveforms must be interpreted, each input and output signal value is shown as a voltage level with respect to time. A green line with black space above it represents logic level 0. A green line with green shading beneath it represents logic level 1. For this test script, each test case is displayed for 10ns (10 nanoseconds) of simulation time. Correlation between the truth table and waveforms is shown below.



# Part 3: Define and test a Full Adder

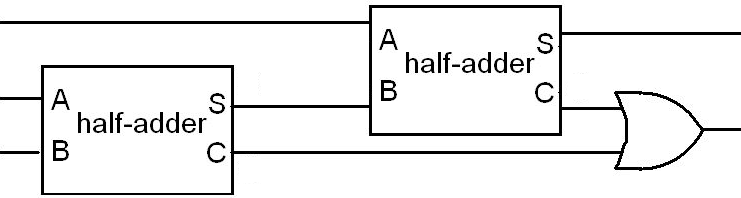
*(Read Chapter 5 page 240 of the textbook for an explanation of the Full Adder.)*

**Step 1: The plan!** A Full Adder will be created using **hierarchical design**. A Full Adder can be made by using two instances of the **Half Adder** and an **OR** gate as shown below.

**FullAdder**

**ha1**

**Cin**



**ha1\_C**

**ha0\_S**

**ha0\_C**

**ha0**

**Cout**

**FA\_S**

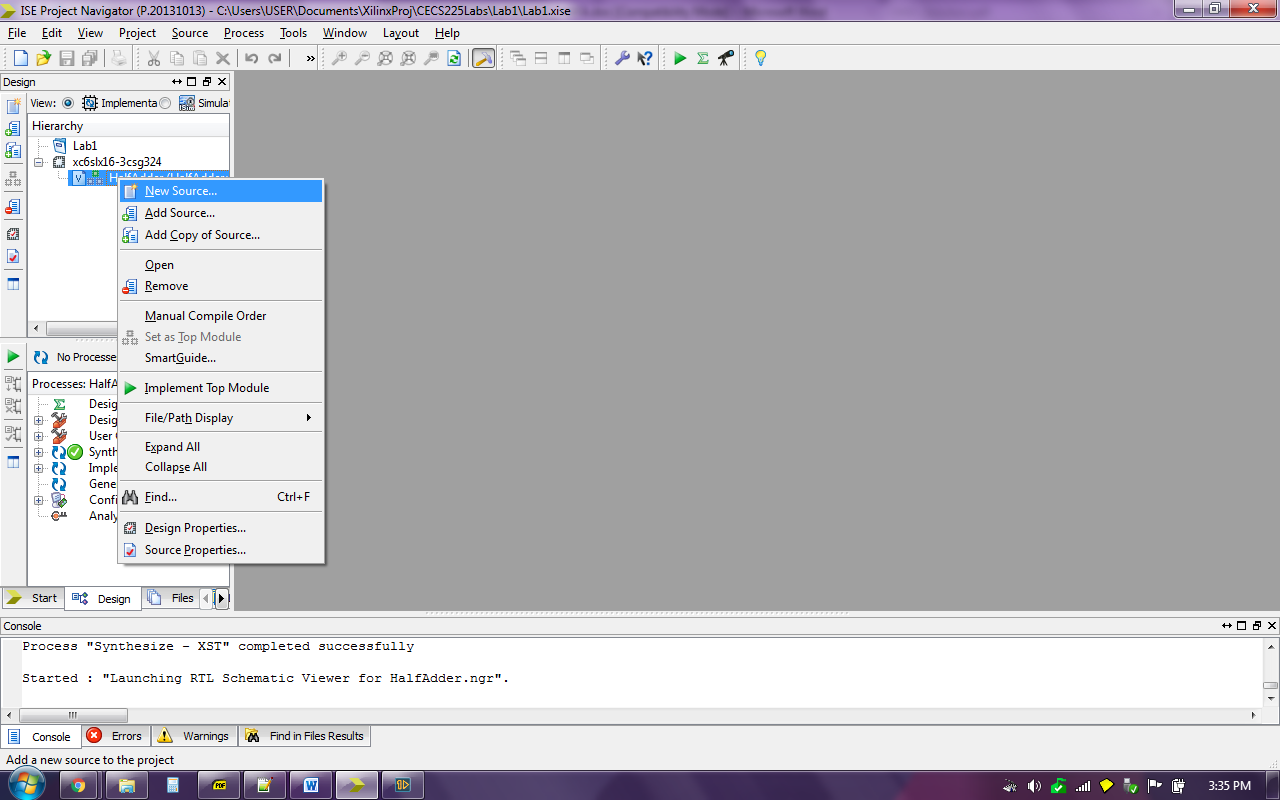
**FA\_B**

**FA\_A**

The circuit has been annotated with extra labels for easy translation into Verilog. Below is the Verilog module to model everything within the dotted box above.



Take note of how the labels in the diagram correlate to the labels in the Verilog module. There are two instances of the HalfAdder module created in Part 2 of this lab. The instances have instance labels **ha0** for HalfAdder zero and **ha1** for HalfAdder one. Variables in Verilog are referred to as *signals*. A Verilog convention known as named port mapping is used to connect inputs and outputs of the HalfAdder to signals within the FullAdder. The port signals from the HalfAdder are preceded with the dot operator and signals from the FullAdder go in the following parenthesis. *Local* signals that are not inputs or outputs within the FullAdder must be declared using the wire keyword. Finally the OR gate is represented by the assignment statement where the OR logic operation is represented by the pipe operator

**Step 2: Make the Full Adder Verilog module.** To make the Full Adder in Xilinx, click on the **Implementation** radio button.

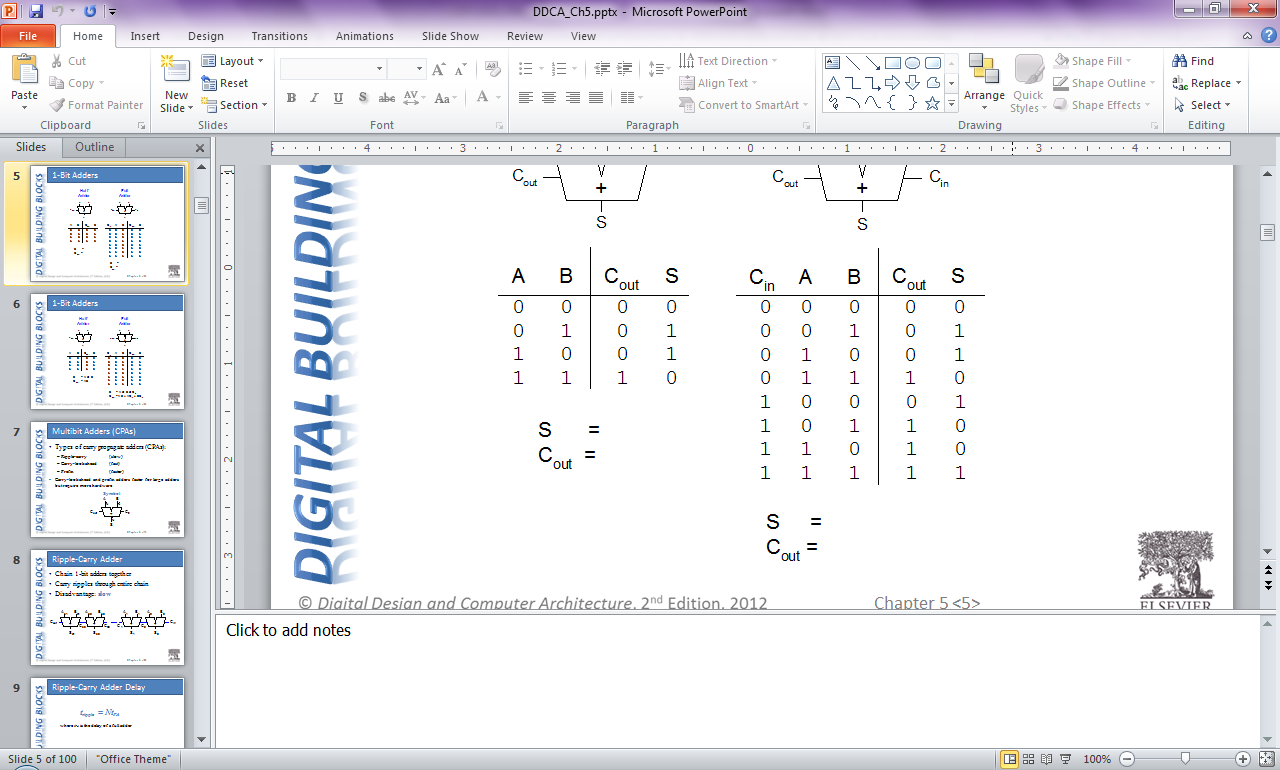
Then right click in the **Hierarchy** window and select **New Source**.

Choose **Verilog Module** within the **New Source Wizard** and enter the name **FullAdder**. Then click the **Next** button.

Skip the **Define Module** window by clicking the **Next** button.

Click the **Finish** button in the **Summary** window and a source file skeleton will be created.

Then enter the Verilog code shown on page 13 of this lab.

**Step 3: Make a Verilog test fixture for the FullAdder.** Use the given truth table on the left to make your test cases. Follow the procedure outlined in Part 2 of this lab to create your FullAdder Verilog test fixture except use the File name: **FullAdder\_Tester** and be sure select FullAdder in the **Associate Source** window of the New Source Wizard.

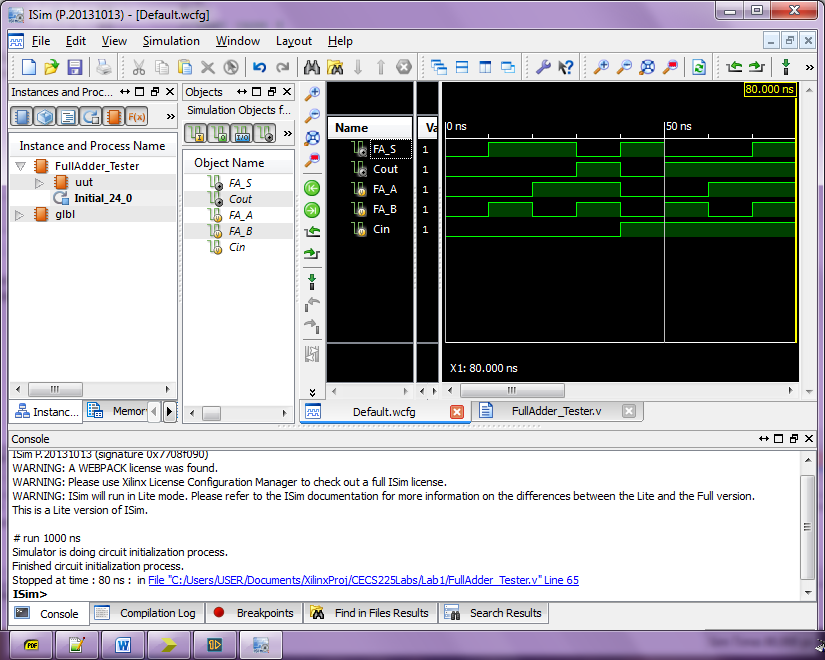
Cin FA\_A FA\_B

Cout FA\_S

Since a module with 3 inputs is being tested, 8 test cases will be needed for an exhaustive test. Create test cases according to the process outlined in Part 2 of this lab.

Once your FullAdder Verilog Test Fixture is complete, save the FullAdder\_Tester, click the Implementation radio button, select the Full\_Adder\_Tester source file, then double click Simulate Behavioral Model in the lower window pane to start a simulation. Refer to Part 2 of this lab for further details regarding this process.

Correct simulation results are shown in the image below.



Save the Verilog Source Files named **HalfAdder.v** and **FullAdder.v** as they will be used in future projects.

**Lab 3 Report:** Submit a single PDF to beachboard with the following contents.

* **Title Page**
  + CECS 225
  + Lab 3
  + Your Name
* **Section 1:** HalfAdder Verilog module source
* **Section 2:** HalfAdder Verilog Test Fixture
* **Section 3:** HalfAdder Simulation Screenshot showing correct results (make sure the order of input/output variables match the truth table)
* **Section 4:** FullAdder Verilog module source
* **Section 5:** FullAdder Verilog Test Fixture
* **Section 6:** FullAdder Simulation Screenshot showing correct results (make sure the order of input/output variables match the truth table)

**Put each section at the beginning of a new page.**