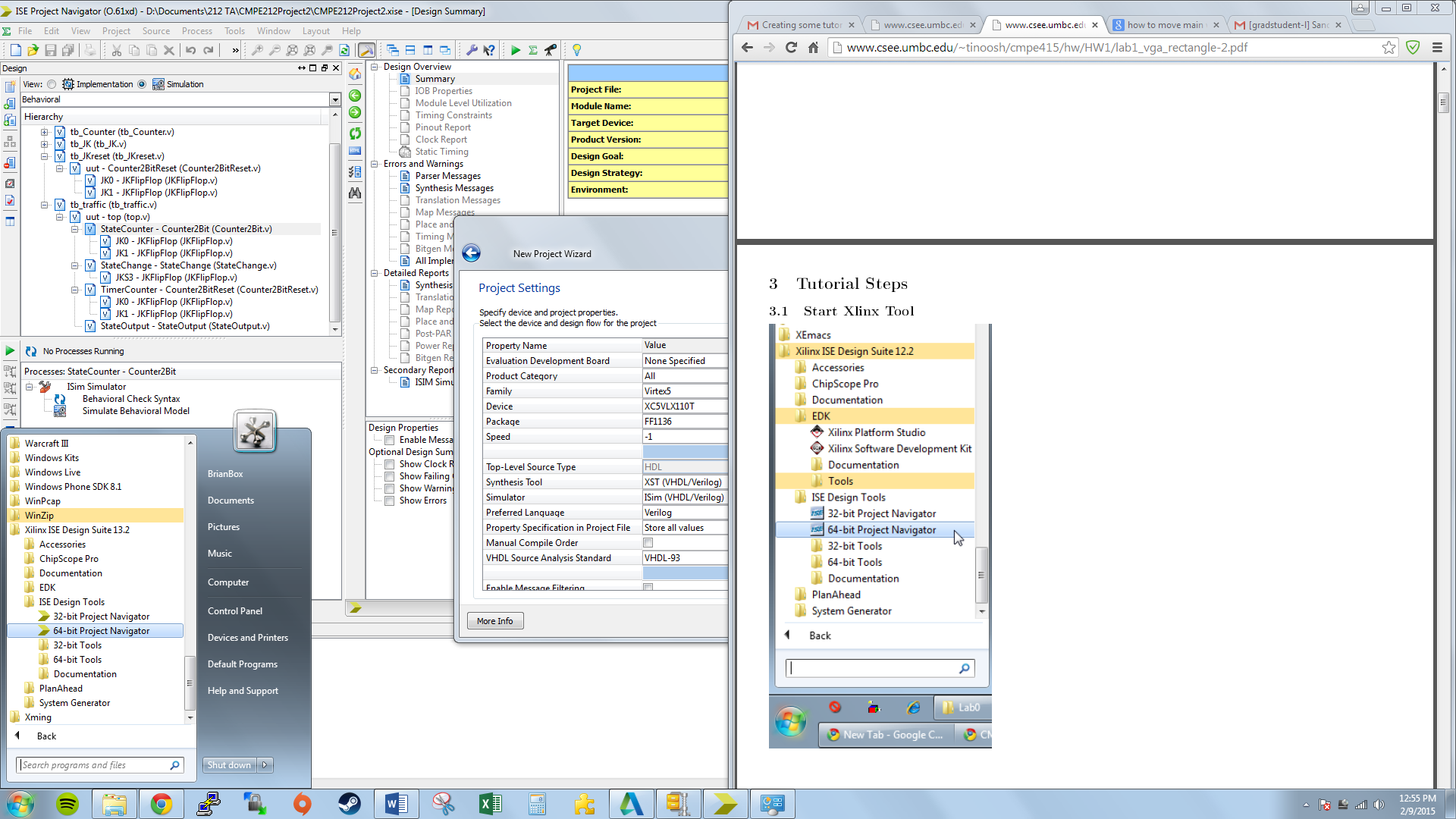
**Verilog Module Tutorial**

By TA Brian W. Stevens – CMPE415 – UMBC Spring 2015 – Dr. Tinoosh Mohsenin

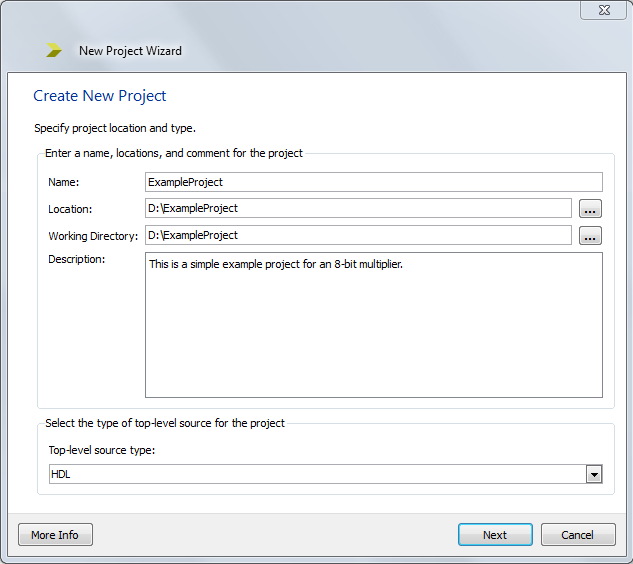
**What will this guide teach you?**

This guide will go through how to use Xilinx 13.2 to create a Verilog module for a simple 8 bit multiplier. It will show you how to add files to Xilinx projects and how to incorporate a testbench for your Verilog module. There are also some other helpful tips as well.

1. **Open up Xilinx ISE Design Suite 13.2**

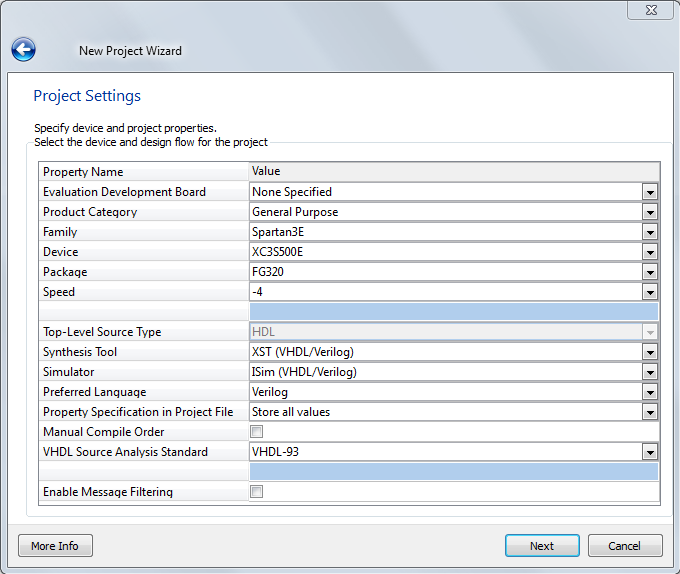


1. **Create New Project**
   1. On the top toolbar go to File > New Project
   2. Name your project and select the location for the project

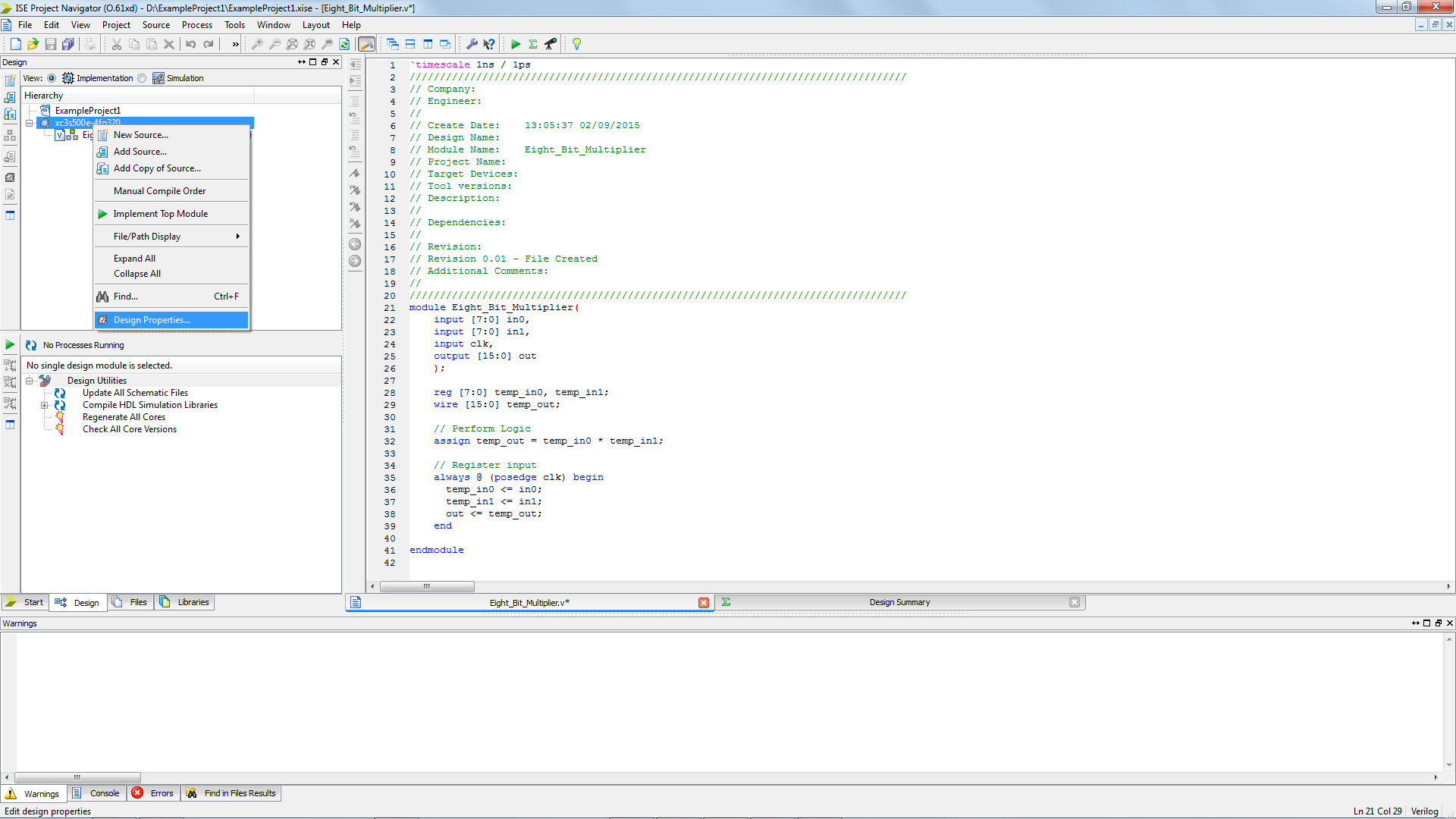


* 1. Click Next

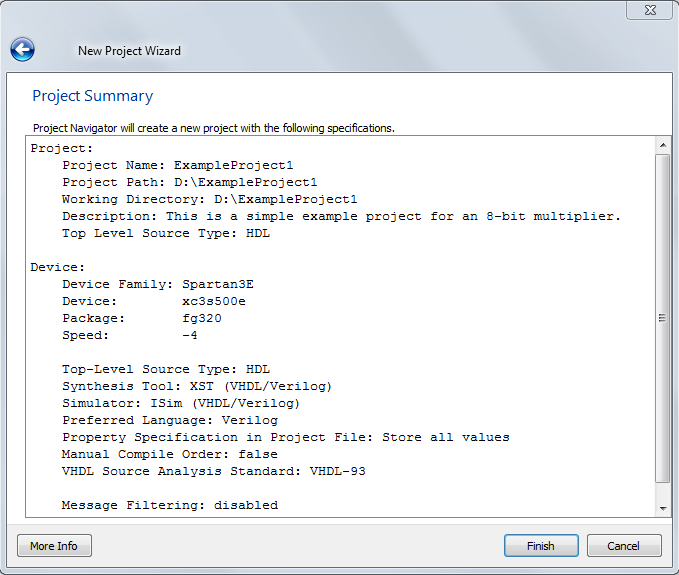
1. **Project Settings**
   1. Enter the following information about your FPGA. This information is also under “Spartan-3E FPGA Family: Data Sheet” under Package Marking (page 6). http://www.xilinx.com/support/documentation/data\_sheets/ds312.pdf



* 1. Click Next
* Helpful Hint #2 – you can change the settings of your FPGA Board(Speed, Family, Package… ect) by right clicking your project tree and selecting “Design Properties”

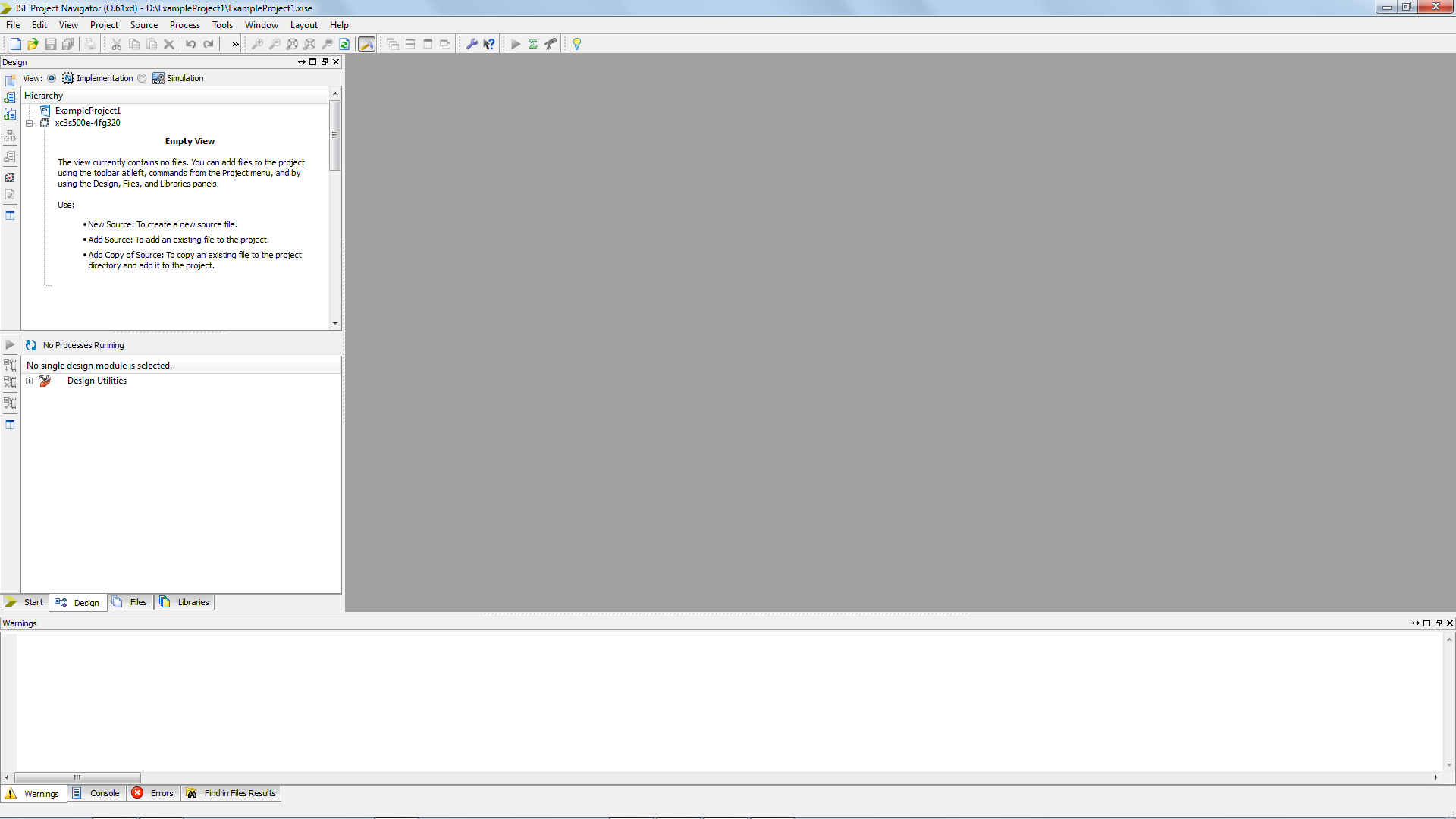


1. **Project Summary Displayed**



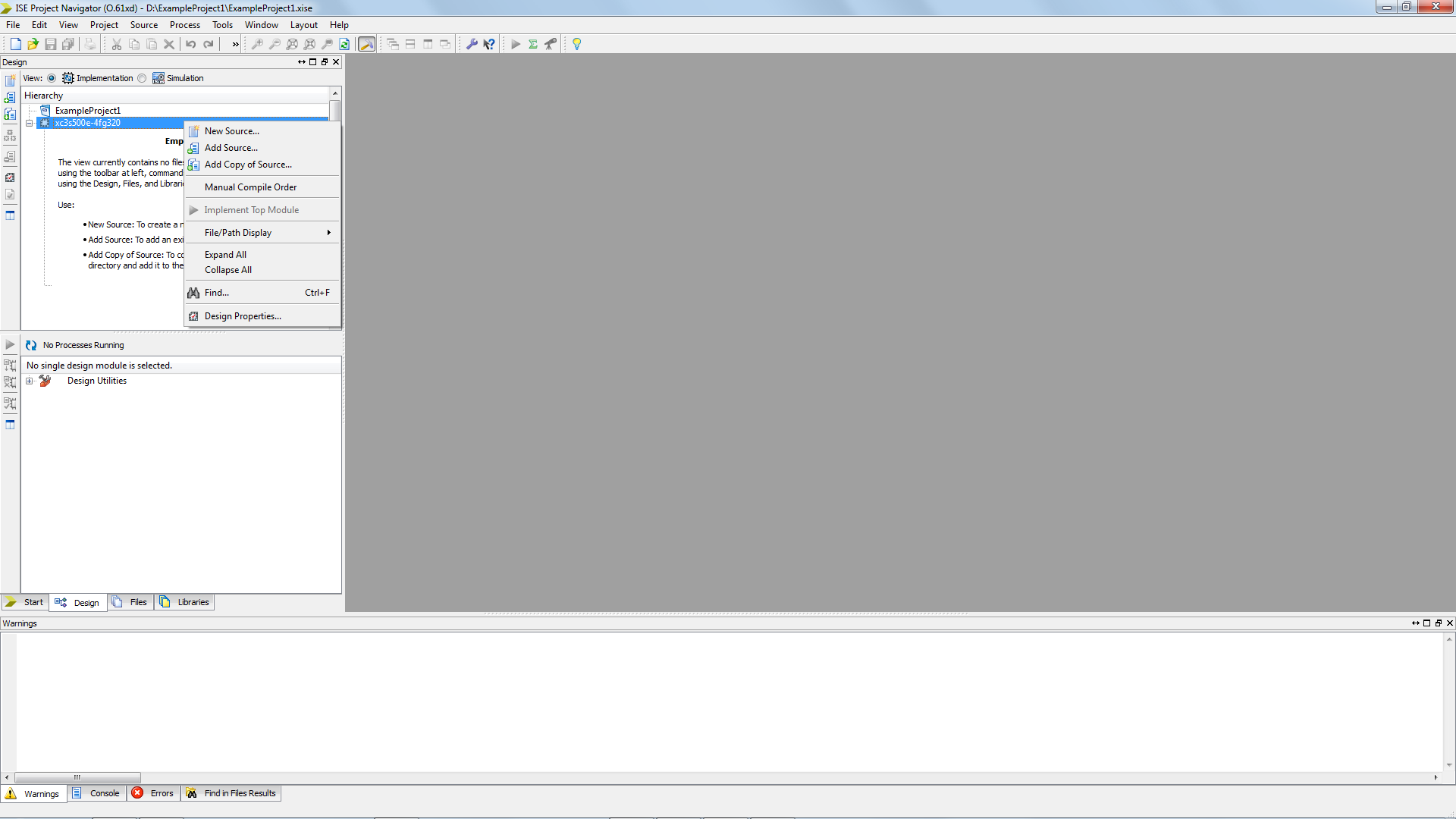
* 1. Click Finish

1. **Project ISE Display**
   1. Your project environment will be displayed as seen below.



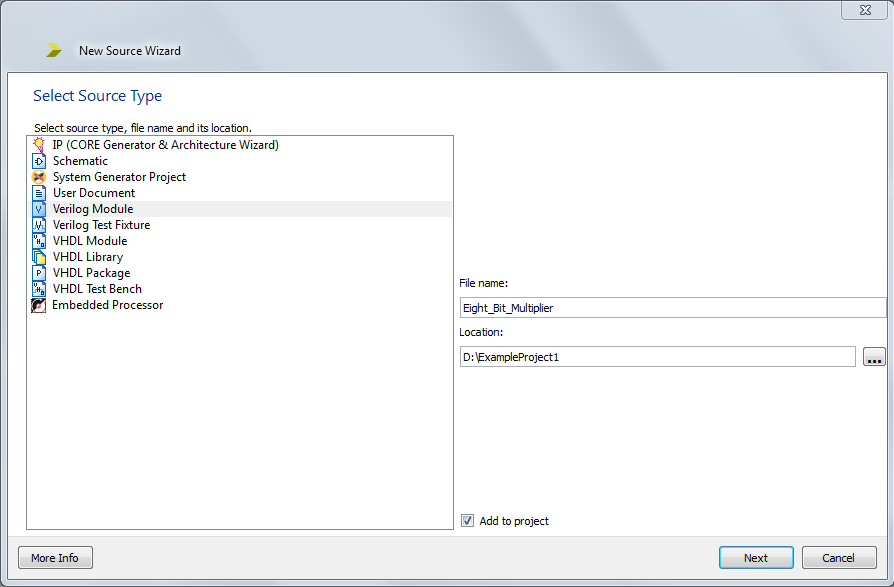
1. **Create a New Source**
   1. Right click on the project and select “New Source” as seen below **OR** go to

Project > New Source **OR** you can use the buttons to the left of the hierarchy tree

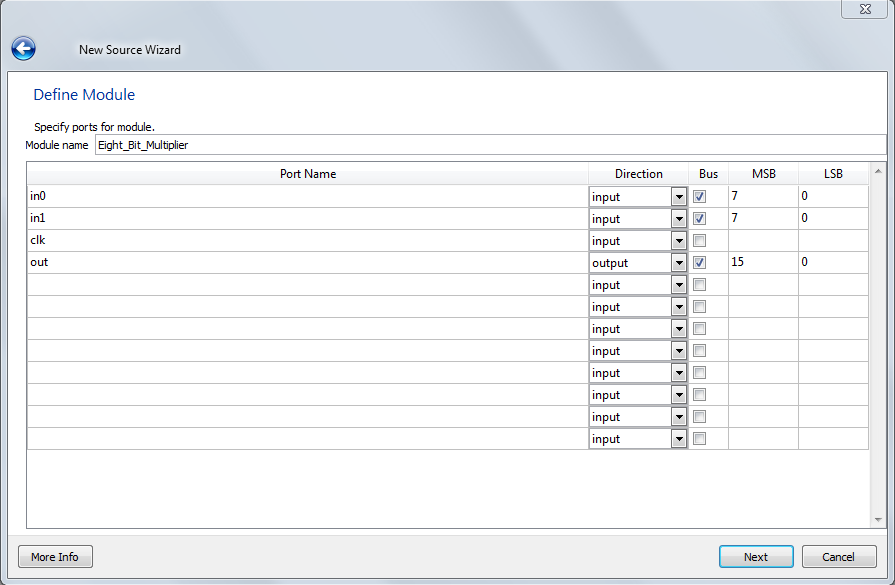


* Helpful Tip #1 – here you can also add existing Verilog and other source files to your design, here are the options
  1. New Source - starts wizard to create a new source \_le (Verilog, VHDL, Schematic, etc...), creating a shell if desired, and adds it for use in the project.
  2. Add Source - Allows you to point to a source \_le in the project directory or anywhere else and include it for use in the project
  3. Add Copy of Source - Creates a copy of the source \_le in the project directory and adds the copy for use in the project

1. **New Source Type and File Name**
   1. After clicking “New Source”, select “Verilog Module” as the file type and enter the file name(here as “Eight\_Bit\_Multipler”) as seen below.

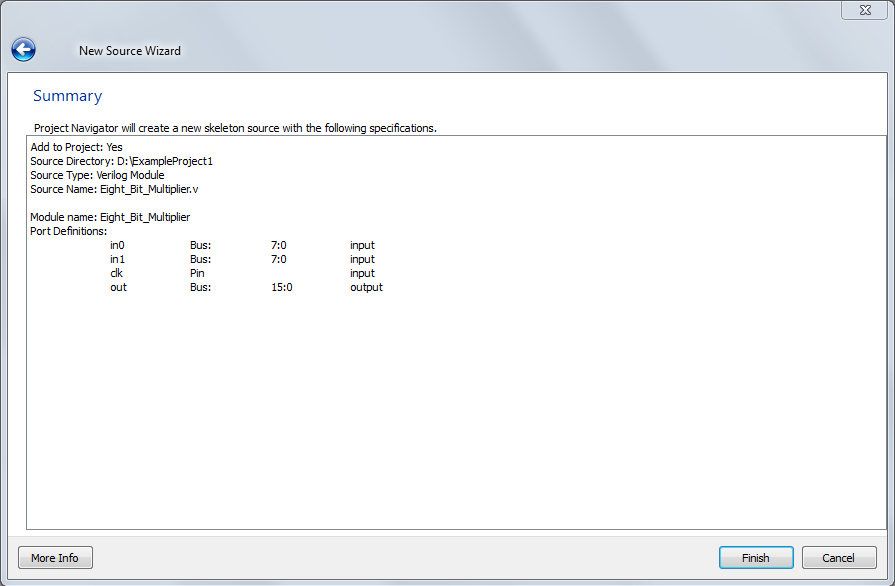


1. **Pre-define Inputs and Outputs**
   1. This window allows you to enter in your pre-defined inputs and outputs. This step will automatically create a template module for you with the inputs and outputs you type below. However, this is not required and you can always just select “Next” and change the inputs and outputs later.



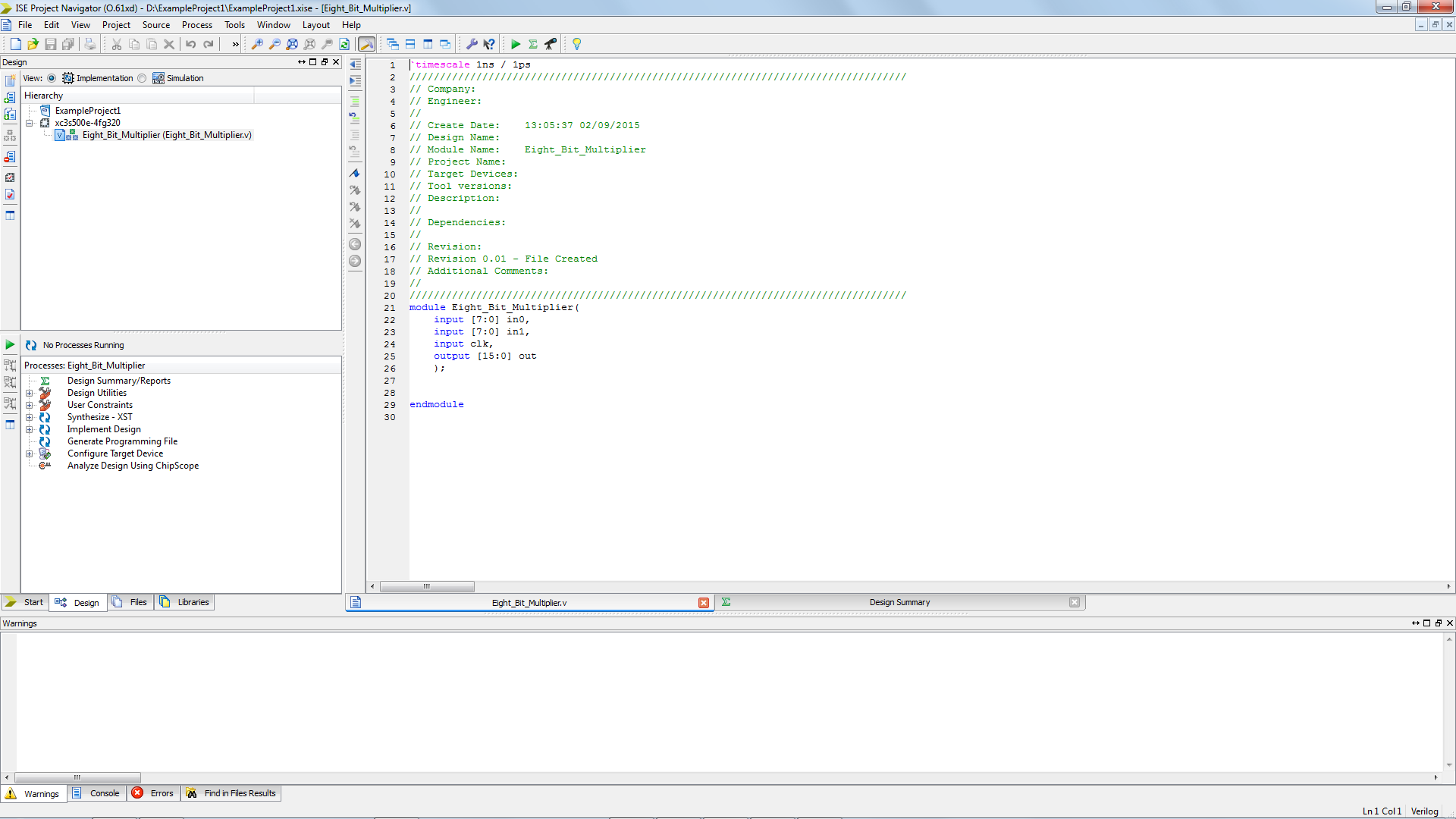
* 1. Click Next

1. **Summary Displayed**
   1. Displays a summary of the file you are creating, lists inputs/outputs ect.

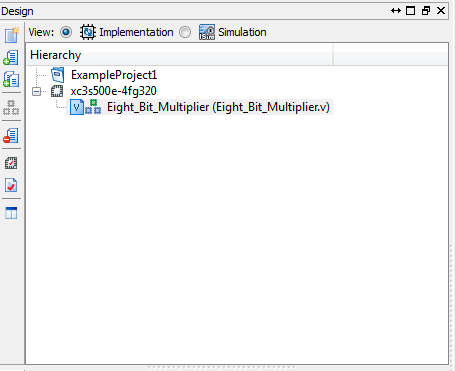


* 1. Click Finish

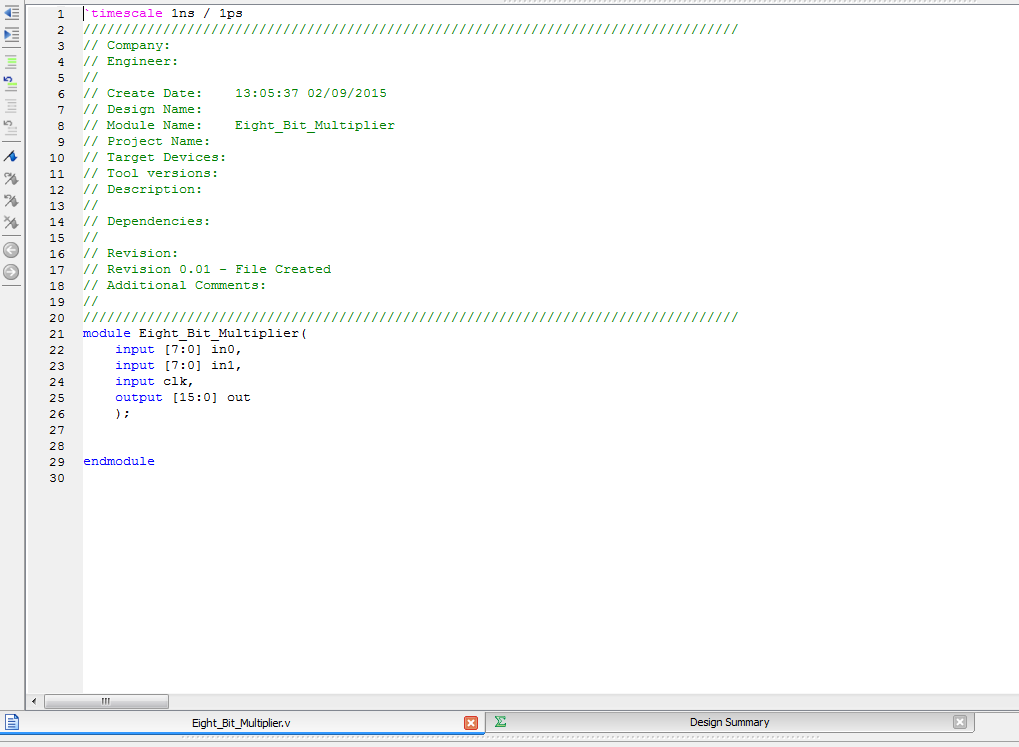
1. **New Verilog File Displayed**
   1. This will now show your added file in the ISE environment.



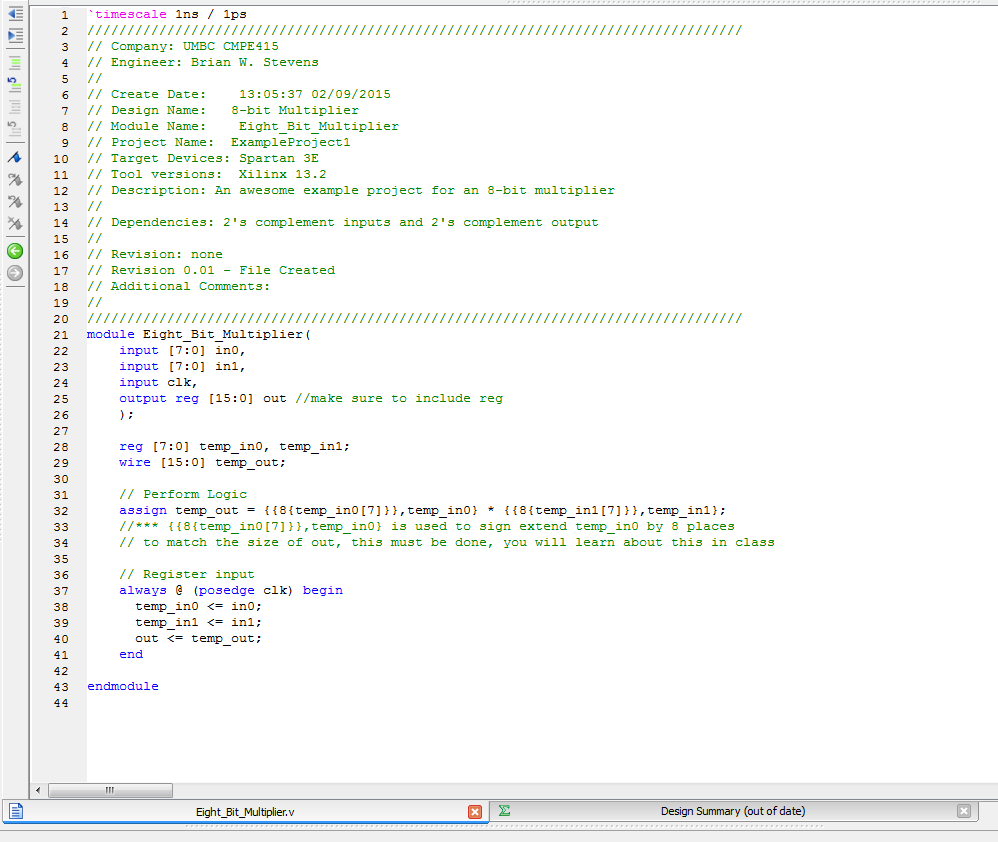
* 1. You will notice that the Verilog module has been added under your project tree(hierarchy)



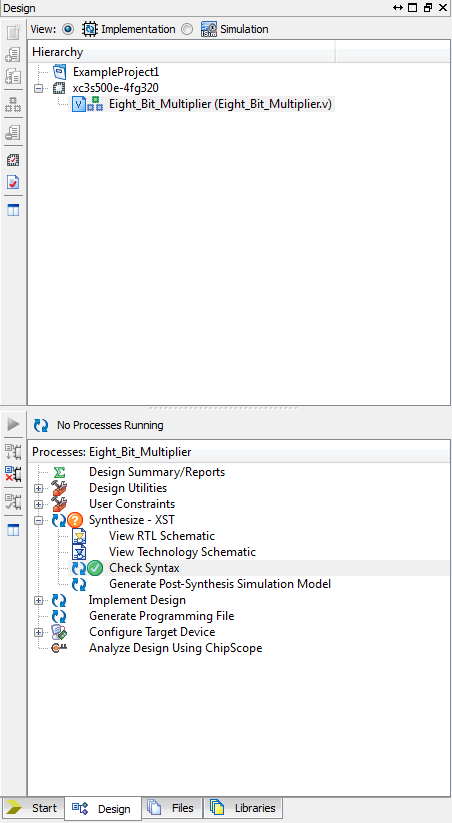
* 1. Your new file will show your auto-generated Verilog module with any inputs and outputs already previously defined.



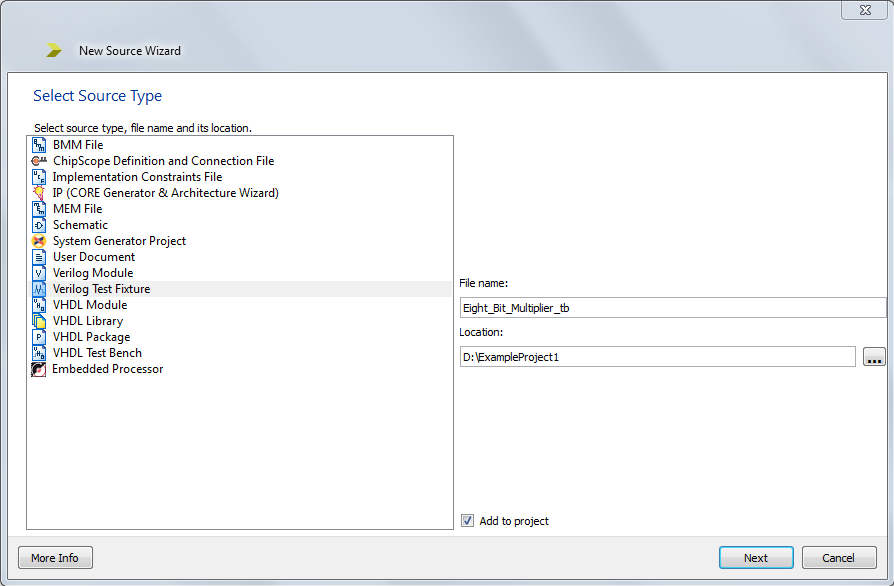
* 1. Completed code for the 8-bit multiplier is below. You will learn about this syntax in class.



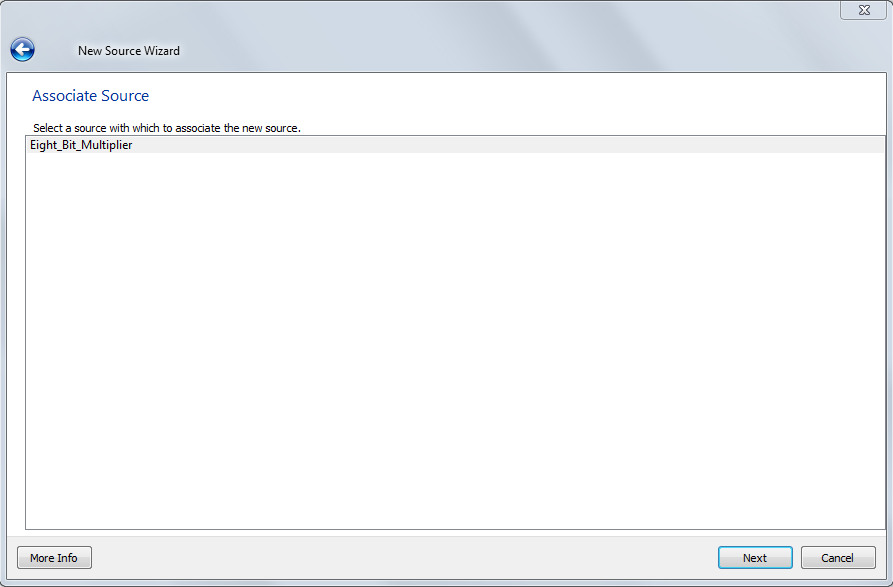
* 1. Make sure to check syntax before running simulations of your Verilog modules



1. **Creating a Testbench**
   1. Now we will create a testbench based off our Verilog module
   2. Go to Project > New Source
   3. Select Verilog Test Fixture and name the file with an extension such as “\_tb” or “\_test”

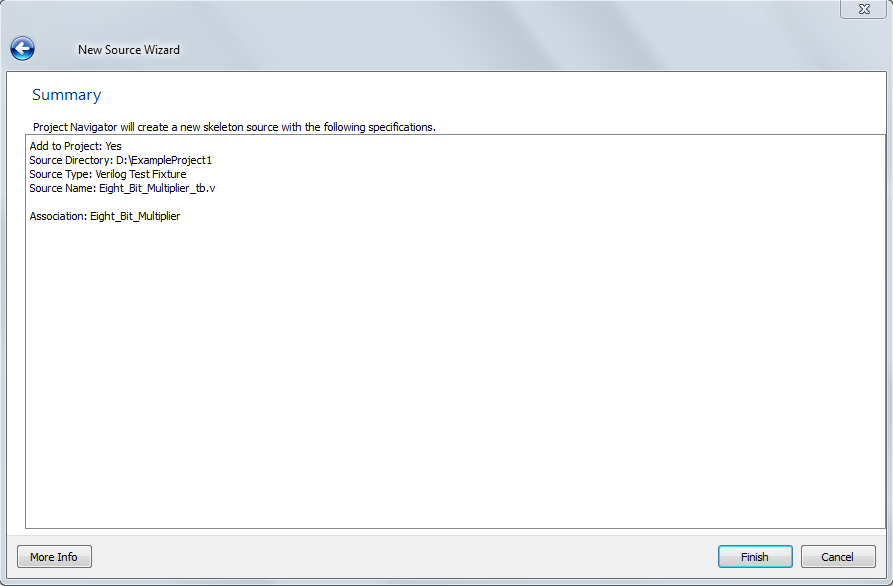


* 1. Choose the associate source or the Verilog file you wish to make a testbech for.

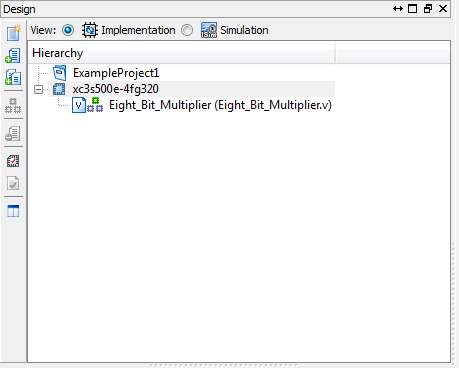


* 1. Click Next

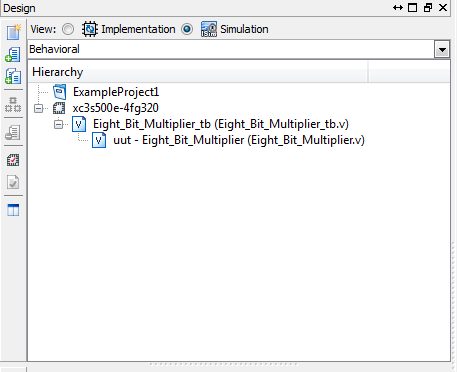
1. **Summary of Testbench**



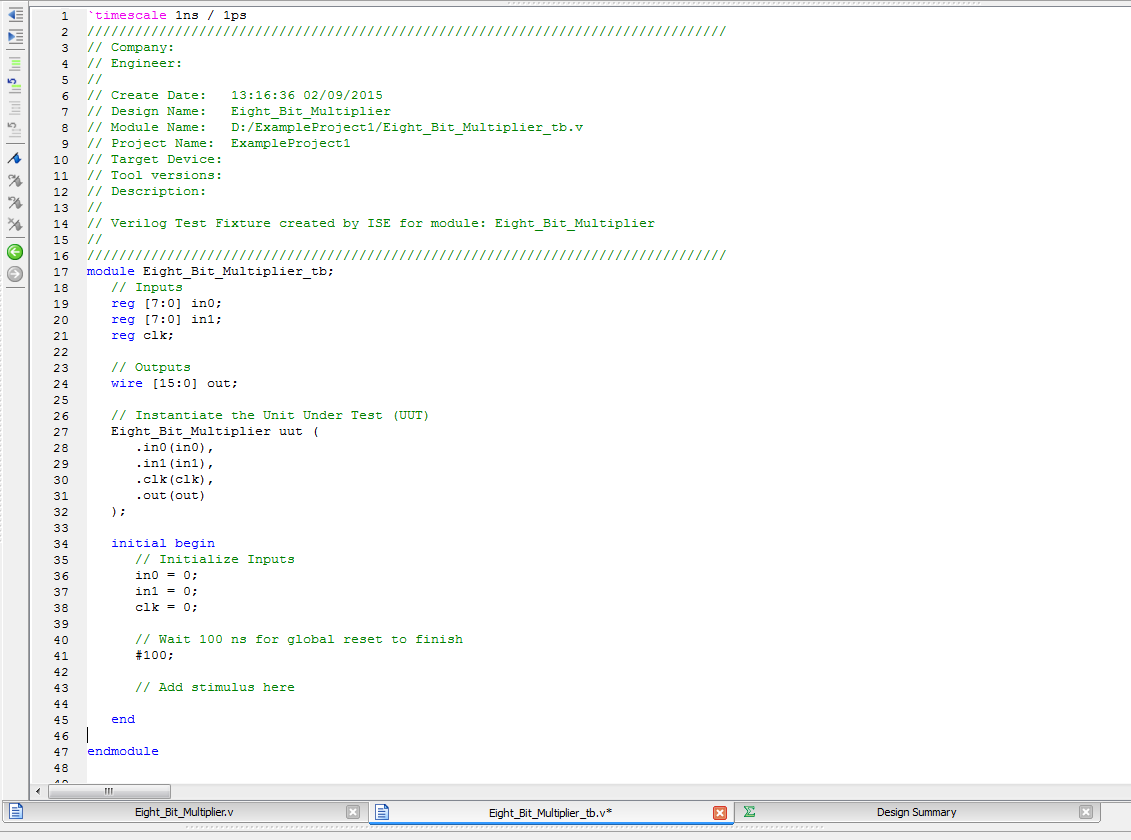
* 1. Click Finish
  2. Earlier in your Design Hierarchy in the top left of the ISE, you saw that your Verilog module was under the “Implementation” Section. This views your non-test fixtures.



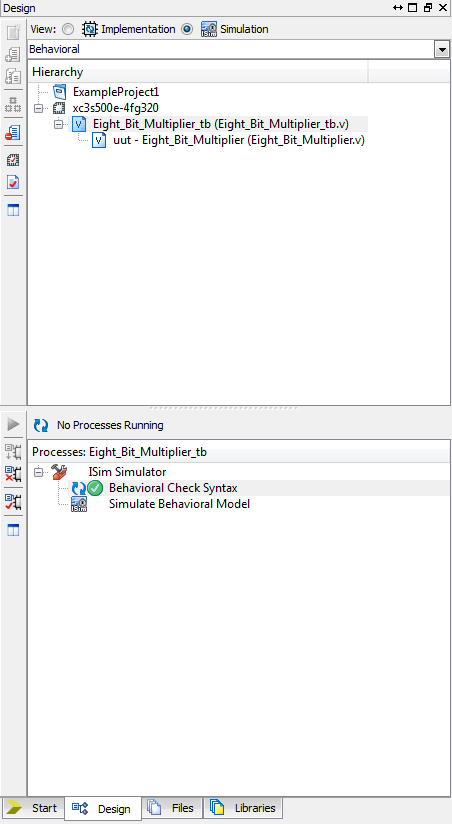
* 1. Click the “Simulation” circle to switch to view your projects simulations files



* 1. This opens the file seen below. As you can see this testbench has auto generated inputs and outputs based on your Verilog module. It also creates an instantiation of the Verilog module that you are testing, in this case called “uut”, so you can test the module versus different inputs. It also initializes the inputs to the module you are testing.



* 1. Make sure to run Behavioral Check Syntax to check your testbench code before simulating the behavioral model



1. **Complete the Test Bench**
   1. Now it is time for you to write your own testbench, try this link for a tutorial about test benches and using ISim.

<http://www.csee.umbc.edu/~tinoosh/cmpe415/tutorials/ISimTestbenchTutorial.pdf>

* 1. A sample testbench file is listed here, provided by Dr. Tinoosh Mohsenin

<http://www.csee.umbc.edu/~tinoosh/cmpe691/hw/HW1/tbench_template.v>