COS/ELE 375 Verilog & Design Tools Tutorial

In this tutorial, you will walk through a tutorial using the Xilinx ISE design software with a Digilent Nexys4 DDR FPGA board. In this tutorial, you will learn how to build a 1-bit full adder using Verilog, and how to build Verilog test bench to test out the full adder design in simulation. Software tools required to complete this tutorial are the Xilinx ISE design tools.

1. Install Xilinx ISE Webpack on your machine

Xilinx ISE Webpack is a free version of Xilinx ISE design software with limited functionalities, but is enough for the class projects. The install package for Xilinx ISE Webpack can be found by going to the following link:
http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html

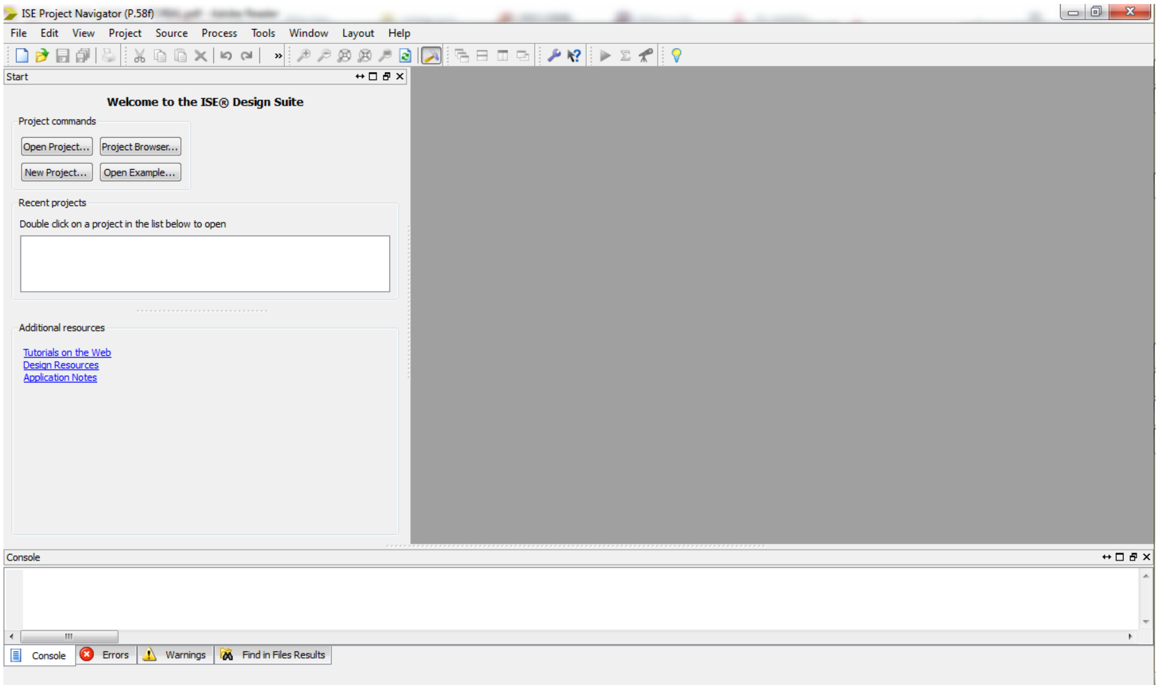
Currently Xilinx ISE Webpack supports Windows and Linux platforms, download the right installer for your machine and complete the installation process. You could use the following article for reference:

http://www.cosmiac.org/wp-content/uploads/ISE-Webpack-download-install-instructions.pdf

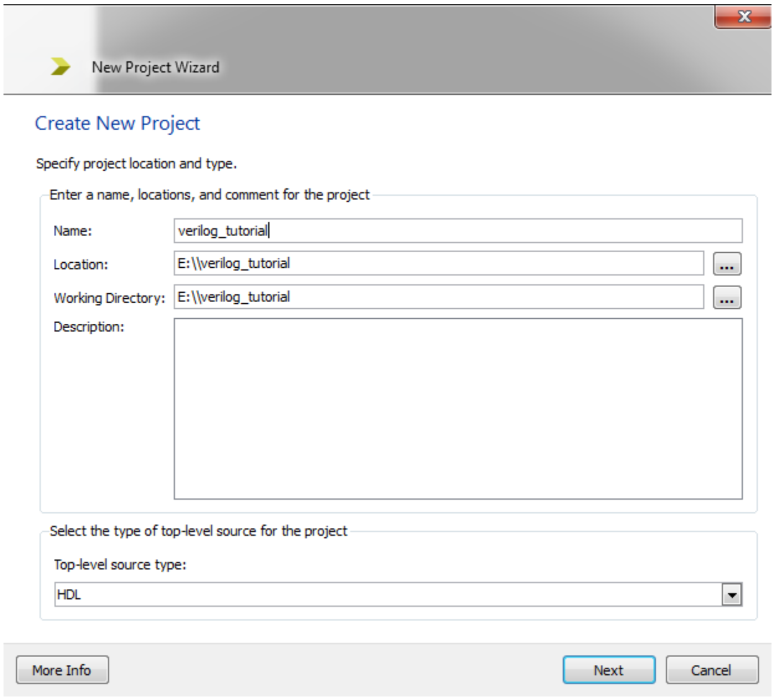
2. Create a New Project with Xilinx ISE

The tutorial below shows the **Step by Step Instructions** on how to set up a new project and run simulations with Xilinx ISE. It will be based on the Windows version of Xilinx ISE 14.7, the Linux version will be identical.

(1) Double click on the **Xilinx ISE 14.7** icon on your desktop or go **to Start->All Programs->Xilinx ISE Design Suite 14.7->ISE Design Tools->64-bit Project Navigator**. The following window should appear.



(2) Select **File->New Project** to create a new project. Give your project a name (i.e. verilog_tutorial). Set your project location. Leave “Top-Level Source Type” as HDL (the default value). Click on **Next** to continue.



Create New Project

Specify project location and type.

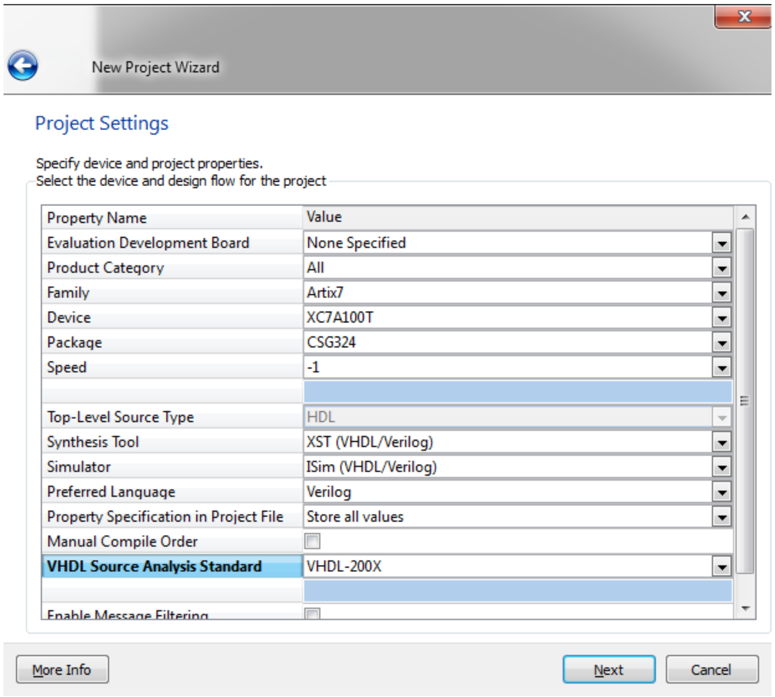
Enter a name, locations, and comment for the project

Name:	verilog_tutorial
Location:	E:\\verilog_tutorial
Working Directory:	E:\\verilog_tutorial
Description:	

Select the type of top-level source for the project

Top-level source type: HDL

(3) Setup the device properties as shown below. Set the Evaluation Development Board as “None Specified”, set Family as “Atrix 7”, set Package as “csg324”, set Speed as “-1” and set VHDL Source Analysis Standard as “VHDL-200X”. This corresponds to the specifications of the Nexys 4 FPGA board in Project 2. Click **Next**, then click **Finish**.

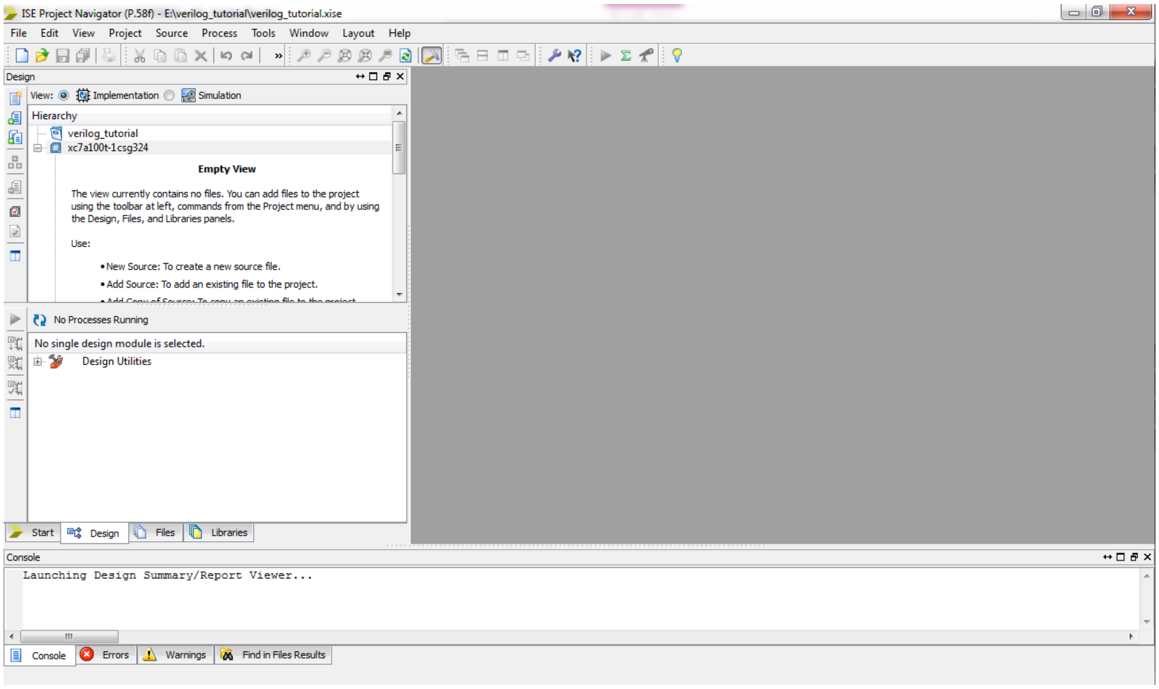


Project Settings

Specify device and project properties.
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
Device	XC7A100T
Package	CSG324
Speed	-1
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	□
VHDL Source Analysis Standard	VHDL-200X
Enable Message Filtering	□

(4) You will see the project view shown below.



Empty View

The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.

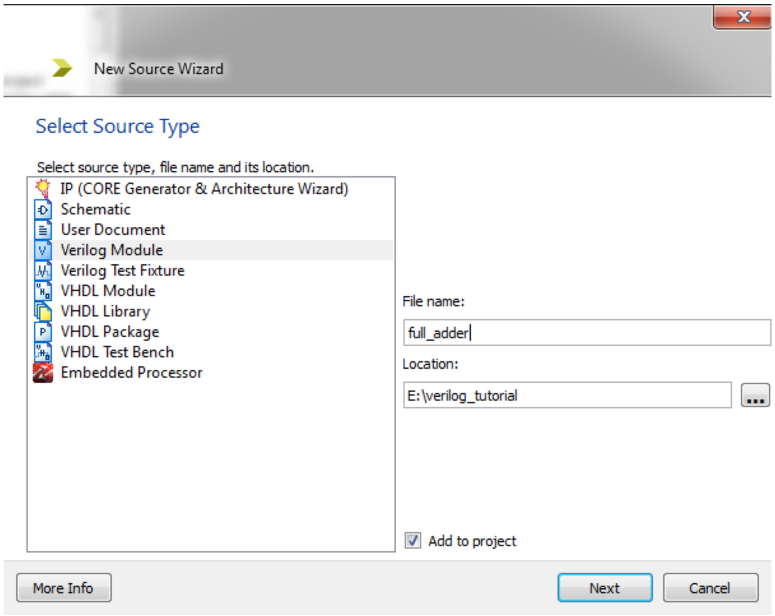
Use:

- New Source: To create a new source file.
- Add Source: To add an existing file to the project.
- Add Copy of Source: To copy an existing file to the project.

No single design module is selected.

```
Launching Design Summary/Report Viewer...
```

(5) Now we are going to add a 1-bit full adder Verilog module to the design. Go to **Project->New Source.** A new source wizard window will pop up. Choose **Verilog Module** as your source type. Setup your file name (i.e. full_adder). Click **Next**.



Select Source Type

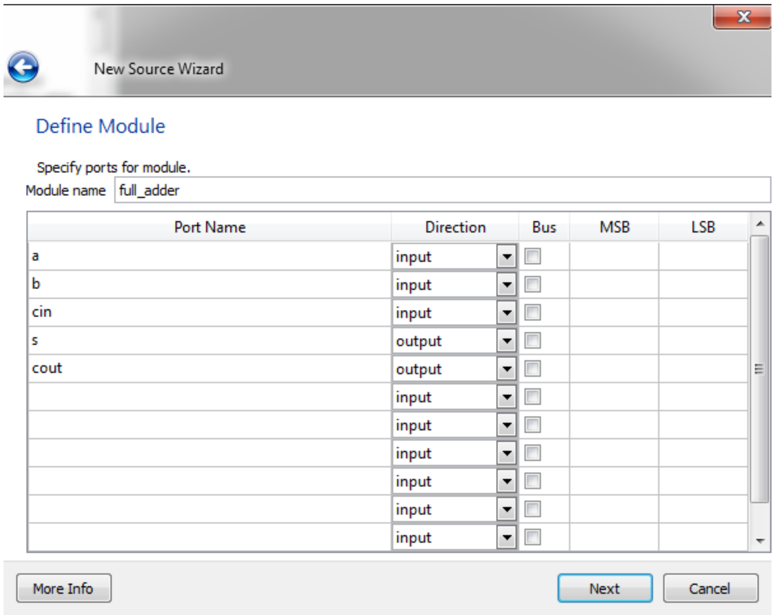
Select source type, file name and its location.

File name: full_adder

Location: E:\verilog_tutorial

☑ Add to project

You will arrive at the second stage of the new source wizard, here you will be asked to specify the input/output ports of the full adder module. You could choose to skip this part and directly edit the source file later. But here we specify ‘a’, ‘b’, ‘cin’ as input ports and ‘s’, ‘cout’ as output ports. Click **Next**, then click **Finish**.



New Source Wizard

Define Module

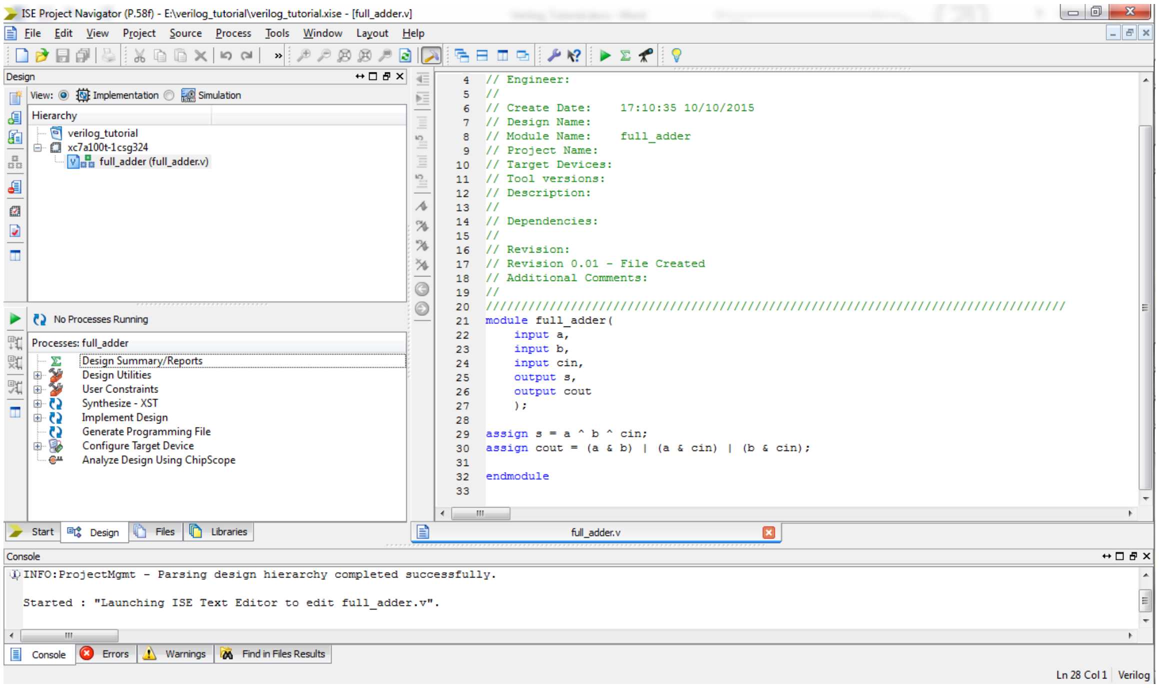
Specify ports for module.

Module name: full_adder

Port Name	Direction	Bus	MSB	LSB
a	input	□		
b	input	□		
cin	input	□		
s	output	□		
cout	output	□		
	input	□		
	input	□		
	input	□		
	input	□		
	input	□		
	input	□		

(6) You will now arrive at project view and see content of the full adder source file (full_adder.v) on the right. You should add the following two lines of Verilog code as show below:

```
assign s = a ^ b ^ cin;

assign cout = (a & b) | (a & cin) | (b & cin);
```



ISE Project Navigator (P.58f) - E:\verilog_tutorial\verilog_tutorial.xise - [full_adder.v]

- verilog_tutorial
- xc7a100t-1csg324
  - full_adder (full_adder.v)

Processes: full_adder

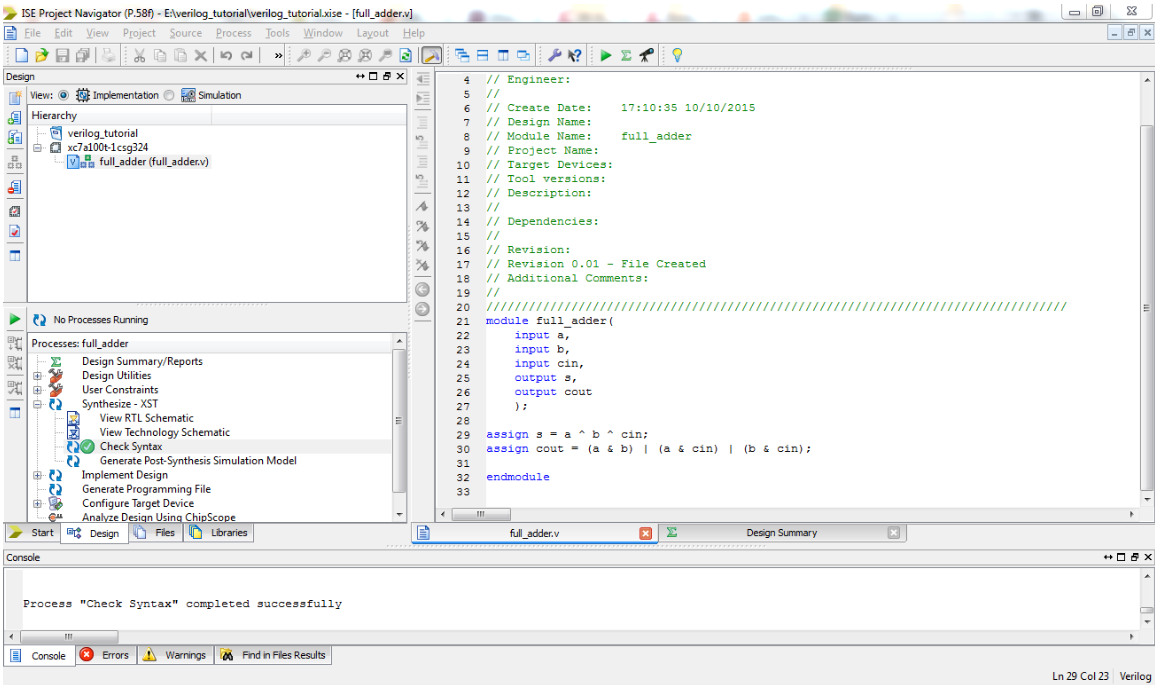
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Configure Target Device
- Analyze Design Using ChipScope

```
// Engineer:
//
// Create Date:    17:10:35 10/10/2015
// Design Name:
// Module Name:    full_adder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module full_adder(
    input a,
    input b,
    input cin,
    output s,
    output cout
    );

assign s = a ^ b ^ cin;
assign cout = (a & b) | (a & cin) | (b & cin);

endmodule

```

Console

```
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.

Started : "Launching ISE Text Editor to edit full_adder.v".
```

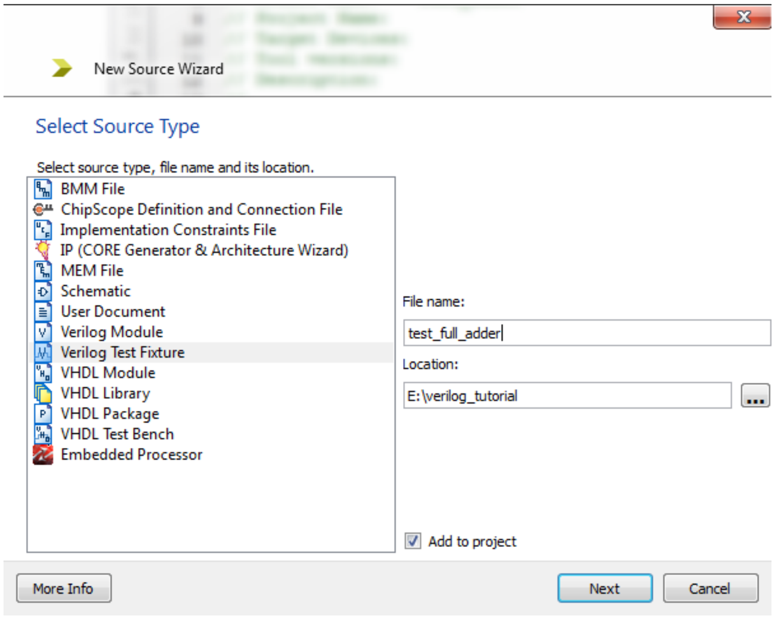
At this point you could run a preliminary check on your design by going to the **Synthesis** drop down menu on the left and double click **Check Syntax.** If your design has no syntax errors you will see a green check, as shown below.



```
// Engineer:
//
// Create Date:    17:10:35 10/10/2015
// Design Name:
// Module Name:    full_adder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module full_adder(
    input a,
    input b,
    input cin,
    output s,
    output cout
    );

assign s = a ^ b ^ cin;
assign cout = (a & b) | (a & cin) | (b & cin);

endmodule
```

```
Process "Check Syntax" completed successfully
```

(7) Now you can proceed to create a test bench for your Verilog design. Go to **Project->New Source.** A new source wizard window will pop up. Choose **Verilog Test Fixture** as your source type. Setup your test bench name (i.e. test_full_adder). Click **Next**.

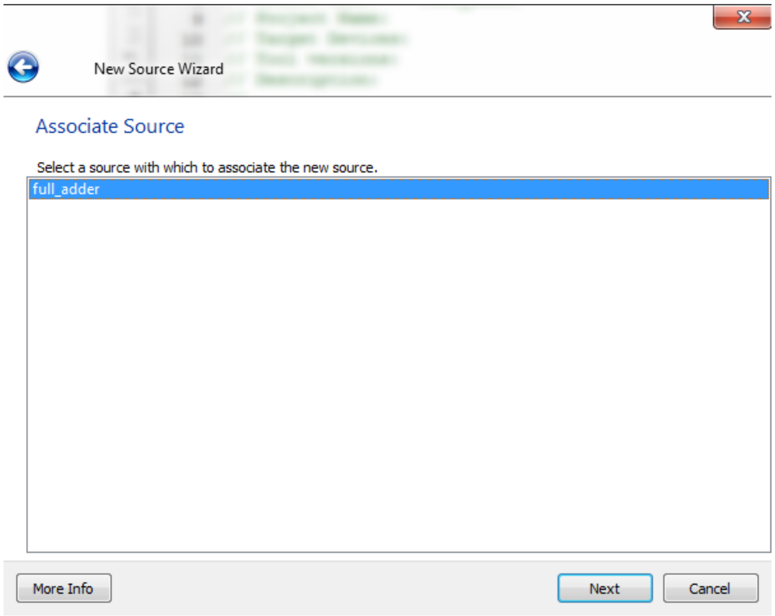


File name: test_full_adder

Location: E:\verilog_tutorial

☑ Add to project

In the associate source stage of the new source wizard, choose full_adder. Click **Next**, then click **Finish**.



New Source Wizard

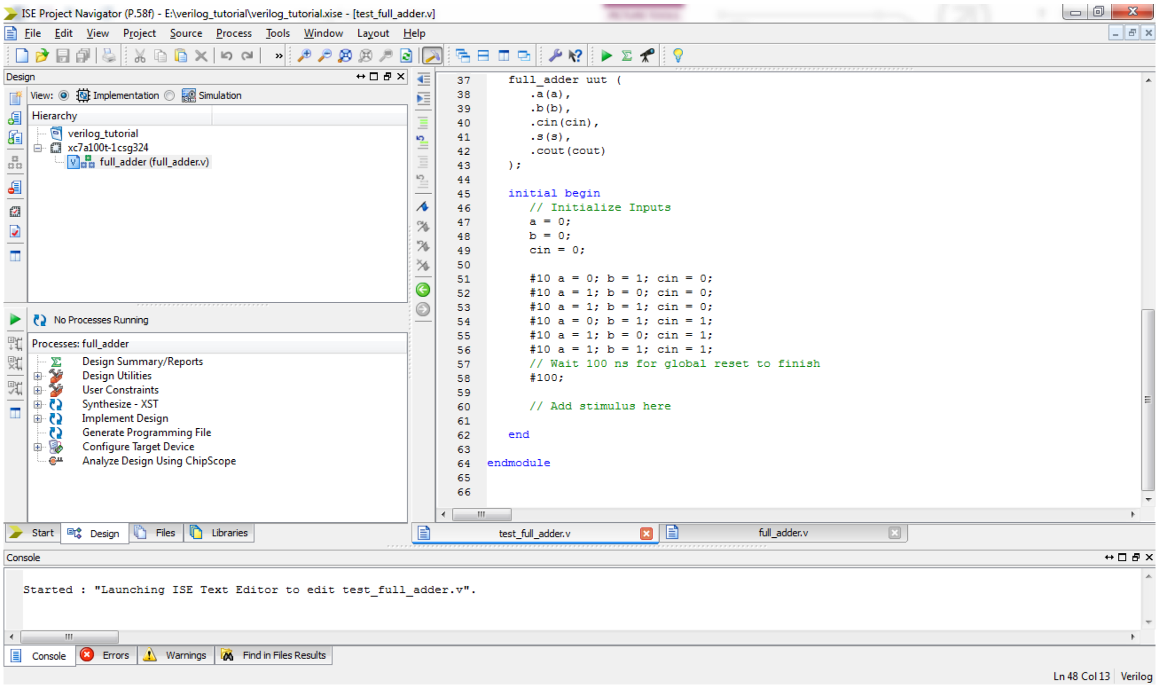
Associate Source

Select a source with which to associate the new source.

full_adder

(8) You will now arrive at project view and see content of the test bench file (test_full_adder.v) on the right. You should add the following two lines of Verilog code as show below:

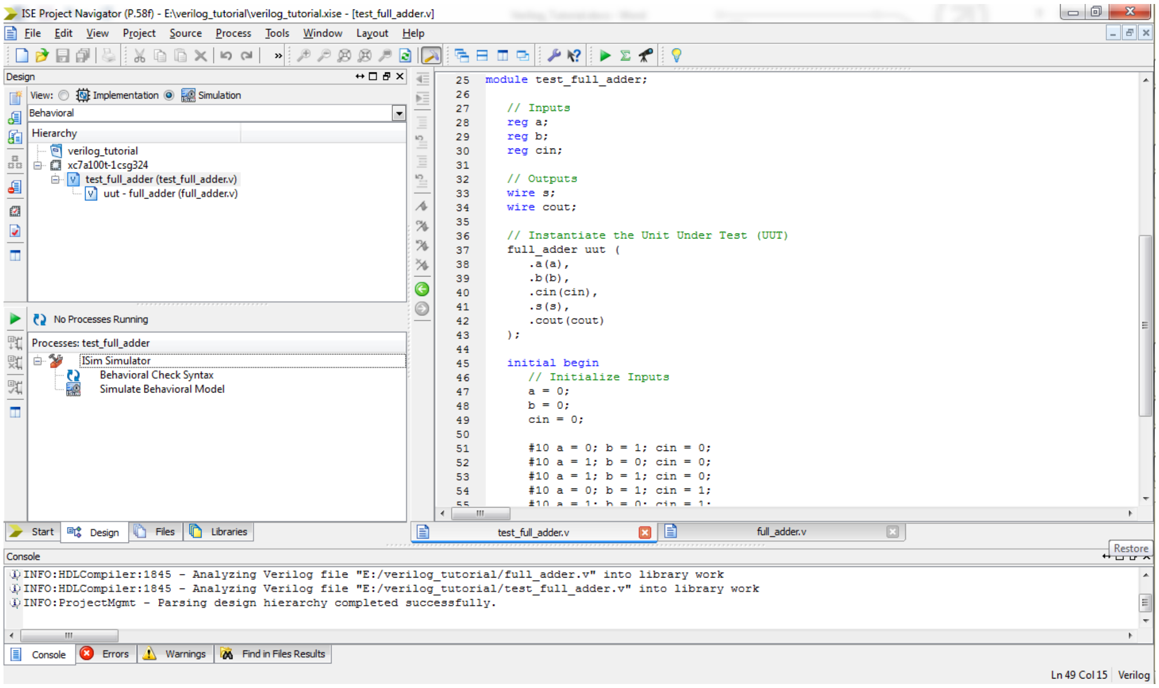
```
#10 a = 0; b = 1; cin = 0;

#10 a = 1; b = 0; cin = 0;

#10 a = 1; b = 1; cin = 0;

#10 a = 0; b = 1; cin = 1;

#10 a = 1; b = 0; cin = 1;

#10 a = 1; b = 1; cin = 1;
```



```
   full_adder uut (
      .a(a),
      .b(b),
      .cin(cin),
      .s(s),
      .cout(cout)
   );

   initial begin
      // Initialize Inputs
      a = 0;
      b = 0;
      cin = 0;

      #10 a = 0; b = 1; cin = 0;
      #10 a = 1; b = 0; cin = 0;
      #10 a = 1; b = 1; cin = 0;
      #10 a = 0; b = 1; cin = 1;
      #10 a = 1; b = 0; cin = 1;
      #10 a = 1; b = 1; cin = 1;
      // Wait 100 ns for global reset to finish
      #100;

      // Add stimulus here

   end

endmodule
```

```
Started : "Launching ISE Text Editor to edit test_full_adder.v".
```

(9) Go to the **Design** section, choose the **Simulation** view instead of the **Implementation** view, and you will see the test bench you just wrote. Go to the **Process** section below the **Design** section, double click "Simulate Behavioral Model".

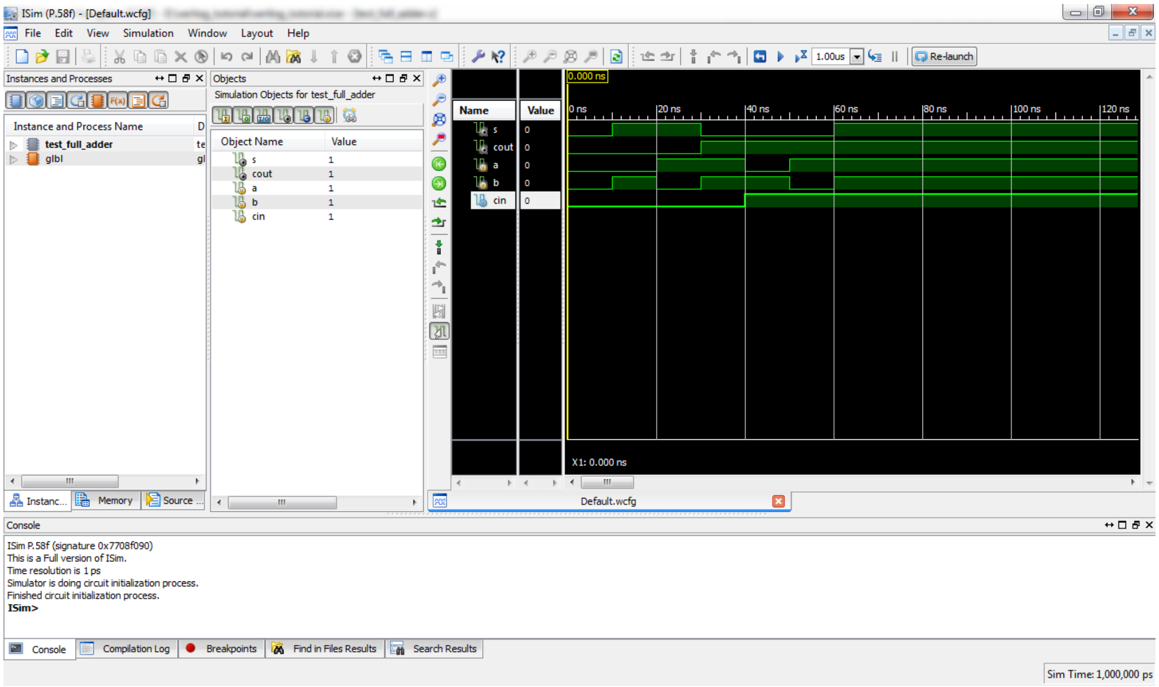


ISE Project Navigator (P.58f) - E:\verilog_tutorial\verilog_tutorial.xise - [test_full_adder.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: ○ Implementation ● Simulation
Behavioral
Hierarchy
verilog_tutorial
xc7a100t-1csg324
test_full_adder (test_full_adder.v)
uut - full_adder (full_adder.v)
No Processes Running
Processes: test_full_adder
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
Start
Design
Files
Libraries
test_full_adder.v
full_adder.v
Console
Restore
Errors
Warnings
Find in Files Results
Ln 49 Col 15
Verilog

```
module test_full_adder;

   // Inputs
   reg a;
   reg b;
   reg cin;

   // Outputs
   wire s;
   wire cout;

   // Instantiate the Unit Under Test (UUT)
   full_adder uut (
      .a(a),
      .b(b),
      .cin(cin),
      .s(s),
      .cout(cout)
   );

   initial begin
      // Initialize Inputs
      a = 0;
      b = 0;
      cin = 0;

      #10 a = 0; b = 1; cin = 0;
      #10 a = 1; b = 0; cin = 0;
      #10 a = 1; b = 1; cin = 0;
      #10 a = 0; b = 1; cin = 1;
      #10 a = 1; b = 0; cin = 1;
```

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "E:/verilog_tutorial/full_adder.v" into library work
INFO:HDLCompiler:1845 - Analyzing Verilog file "E:/verilog_tutorial/test_full_adder.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

(10) You will arrive at the ISim simulator interface shown below. Here you can play with the waveform on the right (i.e. zoom in/out) and verify if your design is behaving correctly.



Object Name	Value
s	1
cout	1
a	1
b	1
cin	1

Name	Value
s	0
cout	0
a	0
b	0
cin	0

```
ISim P.58f (signature 0x7708f090)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Sim Time: 1,000,000 ps

(11) This is the end of tutorial on working with Verilog design and simulation using Xilinx ISE.