Xilinx ISE - tutorial

February 9, 2016

Contents

Overview

Xilinx ISE is an integrated programming environment for designing, testing, and implementing designs in Xilinx FPGAs. In this tutorial, we will design and test the full adder shown in Figure 1.

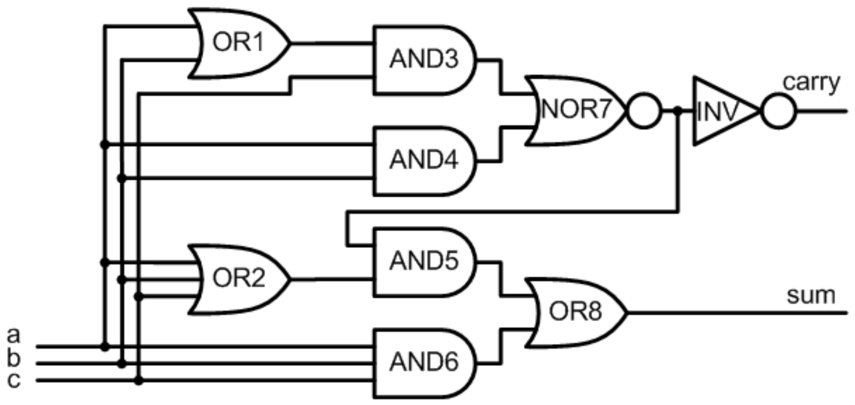


Figure 1

1 Creating a new schematic project

You can start Xilinx ISE by double-clicking on the following icon on your desktop.

Figure 2

After you have started Xilinx ISE, you should create a new project. A *project* in ISE is a collection of source files and test programs. You can start a new project by selecting the option `File/New Project...` from the menu.

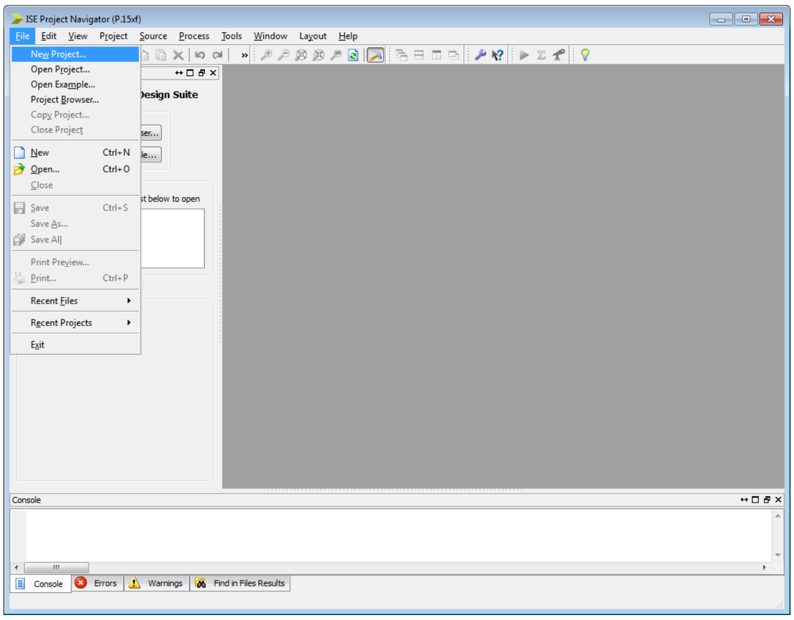
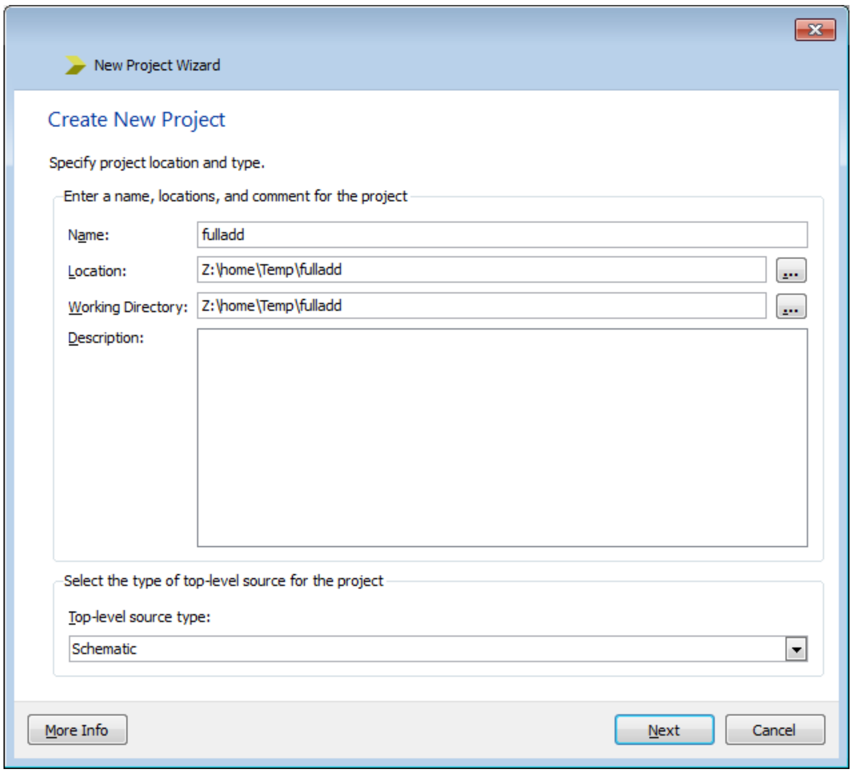


Figure 3

A new window (Figure 4) opens were you need to supply the name of the project (*fulladd*) and the location where the project files will be stored. You should also make sure that the option `Schematic` is selected as *Top-level source type*. Click *Next* to continue.



Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: fulladd

Location: Z:\home\Temp\fulladd

Working Directory: Z:\home\Temp\fulladd

Description:

Select the type of top-level source for the project

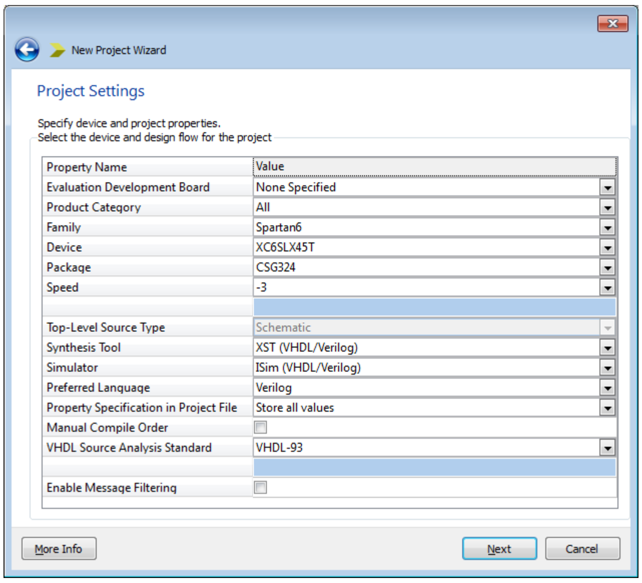
Top-level source type: Schematic

Figure 4

In the next window (Figure 5), you need to supply the details of the project. You should select the following options:

- Family: `Spartan6`
- Device: `XC6SLX45T`
- Package: `CSG324`
- Speed: *-3*
- Synthesis Tool: `XST (VHDL/Verilog)`
- Simulator: `ISim (VHDL/Verilog)`

Click *Next* to continue.



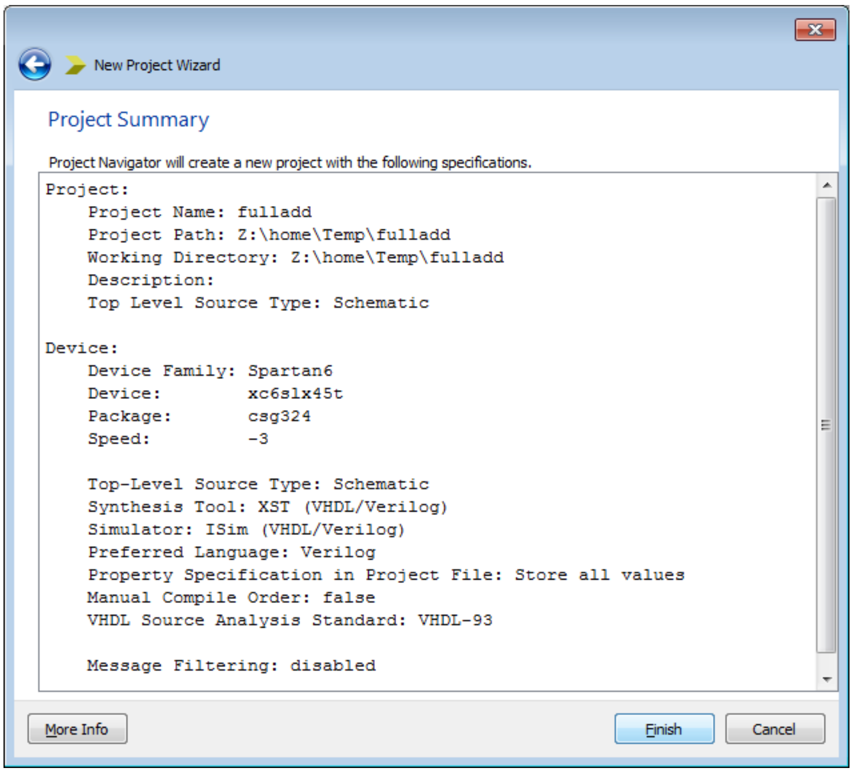
Project Settings

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	Spartan6
Device	XC6SLX45T
Package	CSG324
Speed	-3
Top-Level Source Type	Schematic
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-93
Enable Message Filtering	☐

Figure 5

Click *Finish* on the next window (Figure 6).



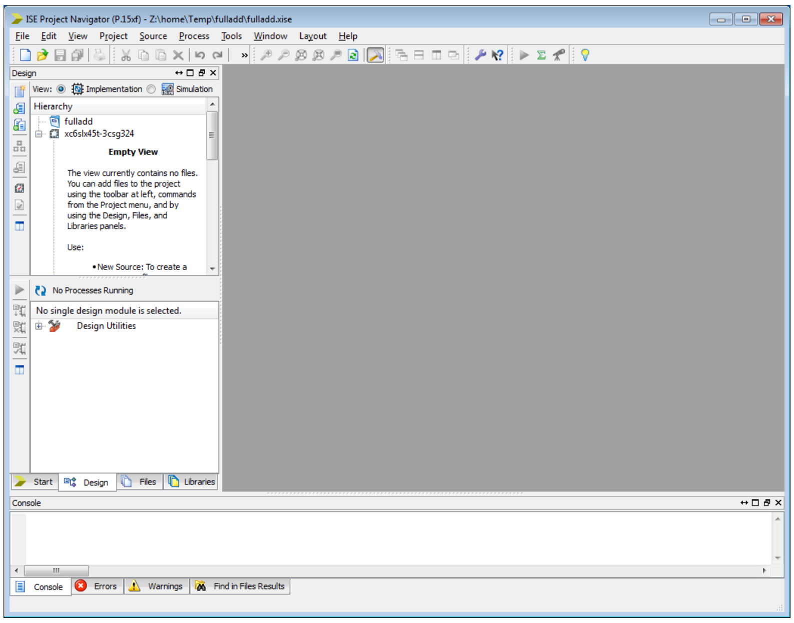
Project Summary

Project Navigator will create a new project with the following specifications.

```
Project:
    Project Name: fulladd
    Project Path: Z:\home\Temp\fulladd
    Working Directory: Z:\home\Temp\fulladd
    Description:
    Top Level Source Type: Schematic

Device:
    Device Family: Spartan6
    Device:        xc6slx45t
    Package:       csg324
    Speed:         -3

    Top-Level Source Type: Schematic
    Synthesis Tool: XST (VHDL/Verilog)
    Simulator: ISim (VHDL/Verilog)
    Preferred Language: Verilog
    Property Specification in Project File: Store all values
    Manual Compile Order: false
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

Figure 6

Once the project is created, it will be automatically opened as shown Figure 7.



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

No single design module is selected.

Figure 7

The project is initially empty. You need to add an empty schematic to the project by selecting the option `Project/New Source...` from the menu (see Figure 8).

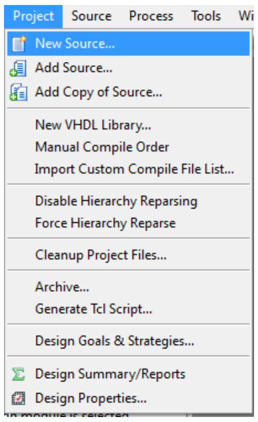
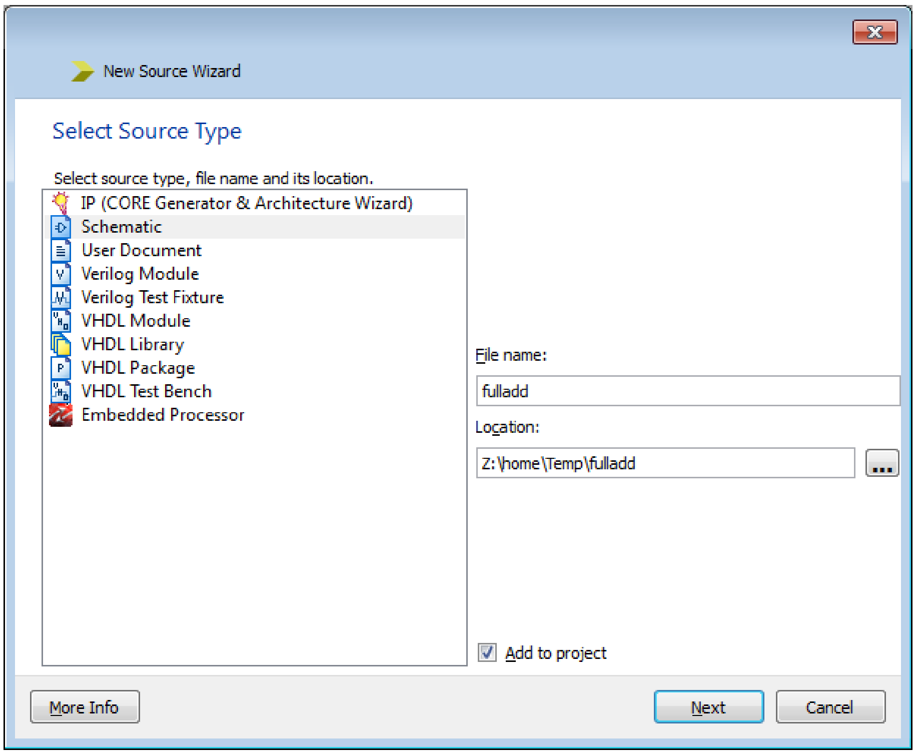


Figure 8

A new window (Figure 9) appears in which you should select `Schematic` as the source type to be created. You should also give the file a meaningful name, e.g., `fulladd`. Click *Next* to continue.


New Source Wizard

Select Source Type

Select source type, file name and its location.

- IP (CORE Generator & Architecture Wizard)
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

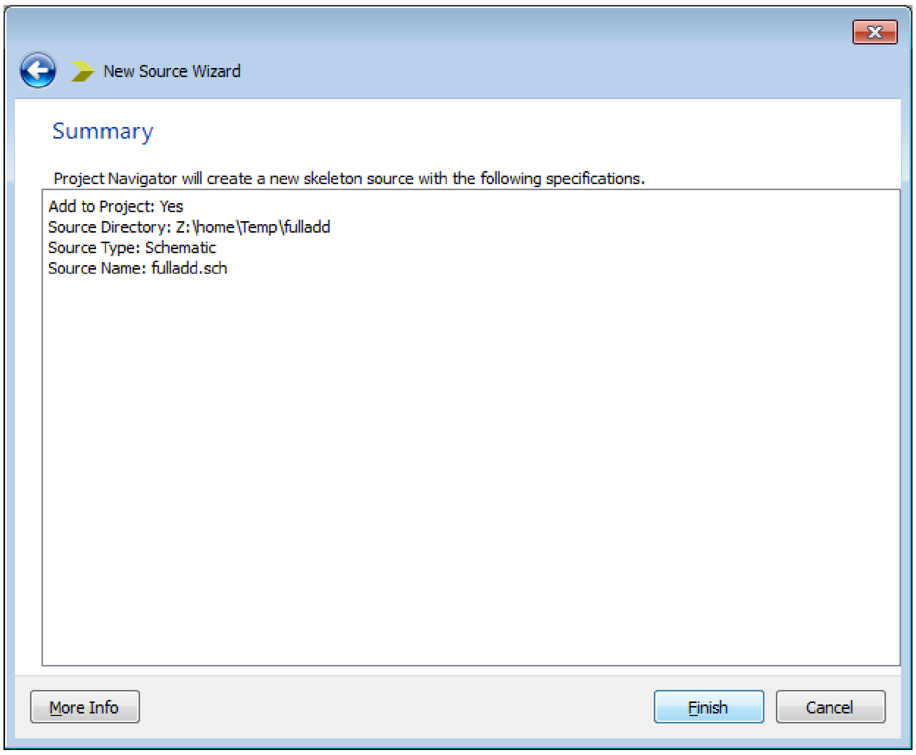
File name: fulladd

Location: Z:\home\Temp\fulladd

[✓] Add to project

Figure 9

On the next window (Figure 10), you should click *Finish* to confirm the creation of the new source file.



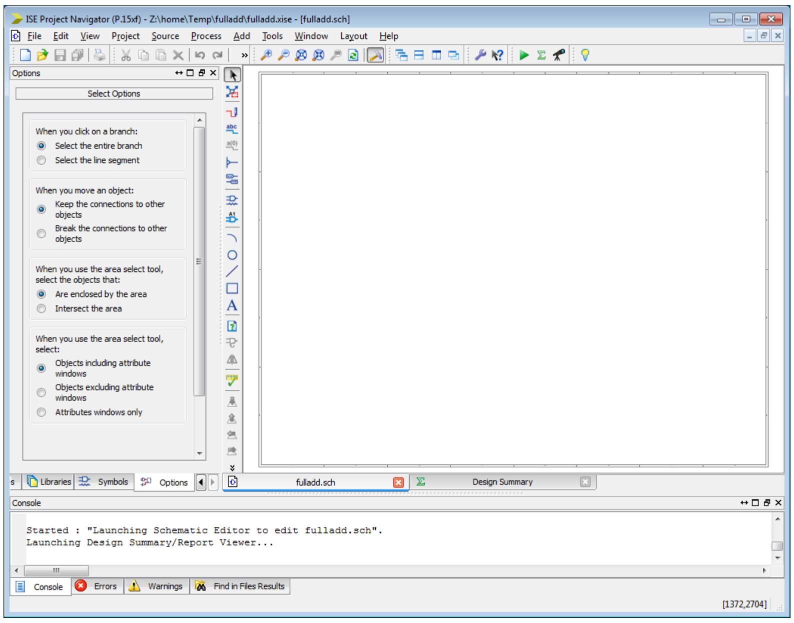
Summary

Project Navigator will create a new skeleton source with the following specifications.

Add to Project: Yes
Source Directory: Z:\home\Temp\fulladd
Source Type: Schematic
Source Name: fulladd.sch

Figure 10

After creating the source file, the view of the main window will change. It should look like the window shown in Figure 11.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
Libraries
Symbols
Options
fulladd.sch
Design Summary
Console
Errors
Warnings
Find in Files Results
[1372,2704]

When you click on a branch:
[✓ Select the entire branch]
Select the line segment

When you move an object:
[✓ Keep the connections to other objects]
Break the connections to other objects

When you use the area select tool, select the objects that:
[✓ Are enclosed by the area]
Intersect the area

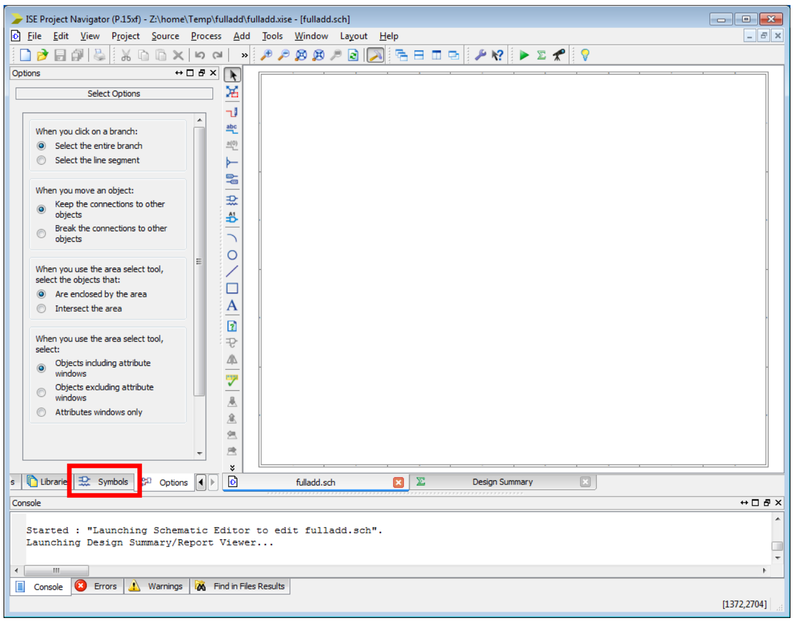
When you use the area select tool, select:
[✓ Objects including attribute windows]
Objects excluding attribute windows
Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 11

2 Schematic editor

After creating a new project, the schematic editor opens with the window shown below. As a first step, you should switch to the *Symbols* tab by clicking on the corresponding tab (red box in Figure 12).


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
Libraries
Symbols
Options
fulladd.sch
Design Summary
Console
Errors
Warnings
Find in Files Results
[1372,2704]

When you click on a branch:
- [✓ Select the entire branch]
- Select the line segment

When you move an object:
- [✓ Keep the connections to other objects]
- Break the connections to other objects

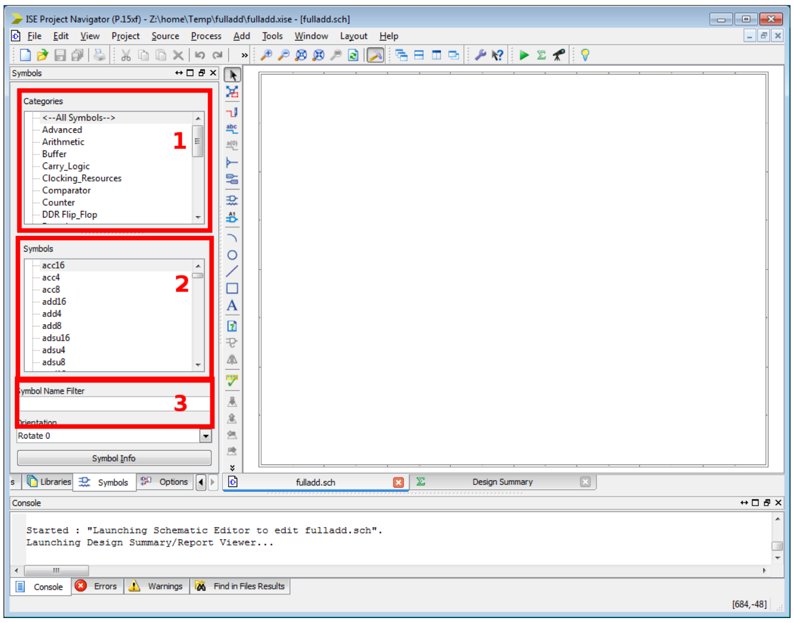
When you use the area select tool, select the objects that:
- [✓ Are enclosed by the area]
- Intersect the area

When you use the area select tool, select:
- [✓ Objects including attribute windows]
- Objects excluding attribute windows
- Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 12

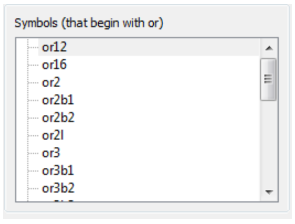
Clicking on the *Symbols* tab will open a new tab that shows all symbols which you can insert to your design. The tab contains three important elements highlighted with three red boxes in Figure 13.



```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 13

Inside box 2 you see a list of all symbols that you can use in your design. If you know that your component belongs to a certain category (e.g., logic or mux), then you can select this category in box 1. This will reduce the number of symbols from which you can select. Of course, you can always return to the complete list by selecting the category *<!–All Symbols–>* from the *Categories* list. If you know the name of a component, then there is an even easier way to locate the component. You may enter (part of) the name of a component inside box 3. This will automatically update the list inside box 2 with all components that match this name. If you for example type `or` in box 3 then you will see the following list of symbols inside box 2.

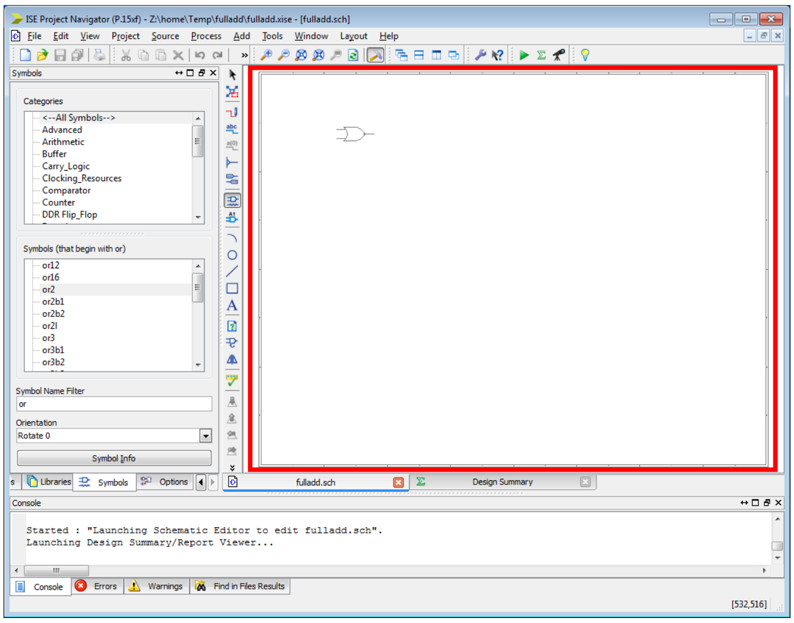


Symbols (that begin with or)

or12
or16
or2
or2b1
or2b2
or2l
or3
or3b1
or3b2

Figure 14

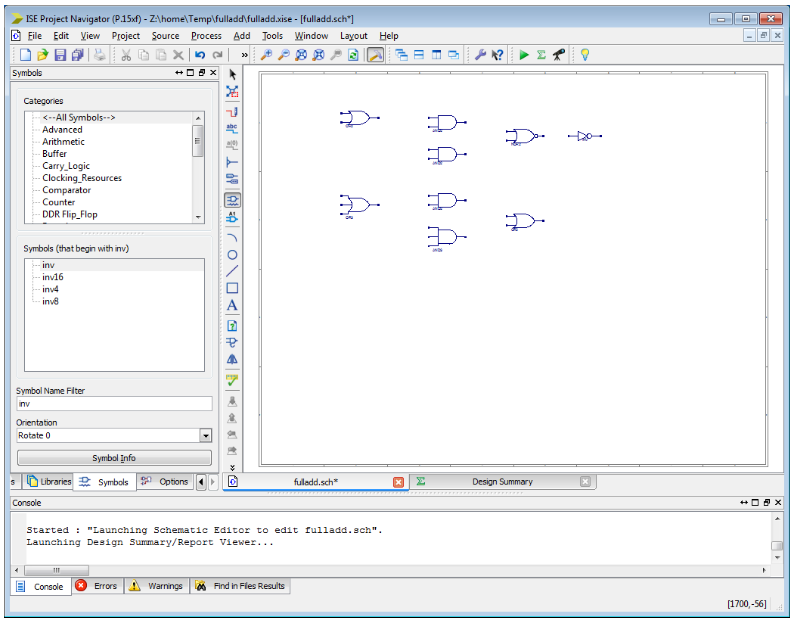
The digits behind the text `or` correspond to the number of inputs the symbols has. For example `or12` corresponds to an or-port with 12 inputs and an or-port with two inputs is called `or2`. You should now select an `or2` port by clicking on this symbol in the symbols list (box 2 in Figure 13). Next, you should move your mouse towards the schematic (red box in Figure 15). You will see that the mouse cursor changes into an *or* port. When you click on the left mouse button, the symbol will be placed on the schematic.



```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 15

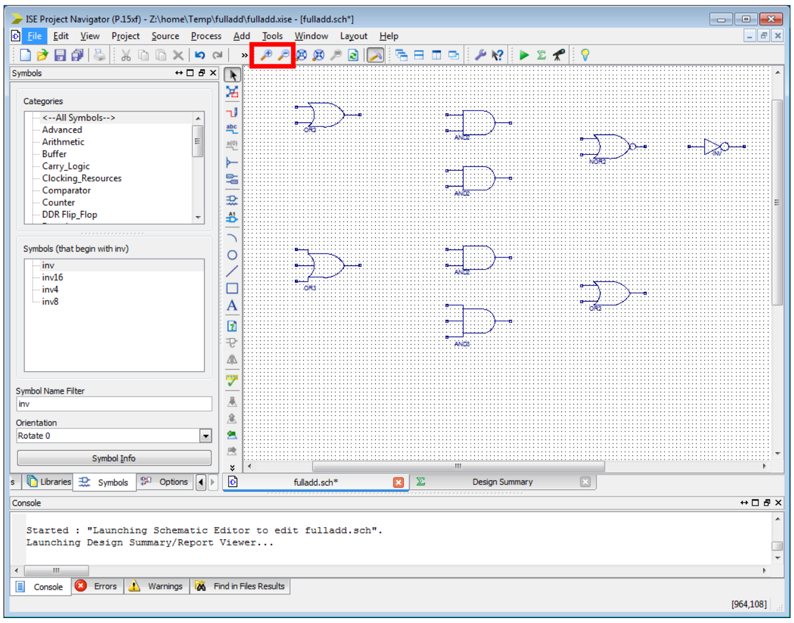
You should add all components needed to build a full adder. Figure 16 shows which components you need to insert (i.e., two or2 port, one or3 port, three and2 ports, one and3 port, one nor2 port, and one inv port).



```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 16

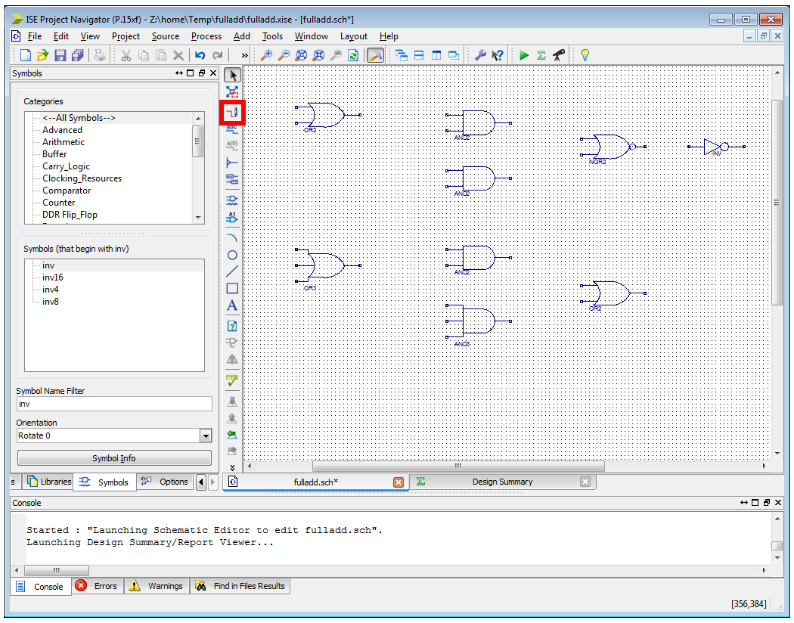
As you will see, the symbols are a bit hard to read. You can always zoom in and out of the schematic using the two zoom buttons in the red box of Figure 17. Pressing the zoom-in button a number of times will make it much easier to view the schematic.



```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 17

Now that we have inserted the symbols, it is time to connect them using wires. As a first step you should click on the *Add wire* button shown in the red box of Figure 18.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]

Categories: <--All Symbols-->, Advanced, Arithmetic, Buffer, Carry_Logic, Clocking_Resources, Comparator, Counter, DDR Flip_Flop

Symbols (that begin with inv): inv, inv16, inv4, inv8

Symbol Name Filter: inv

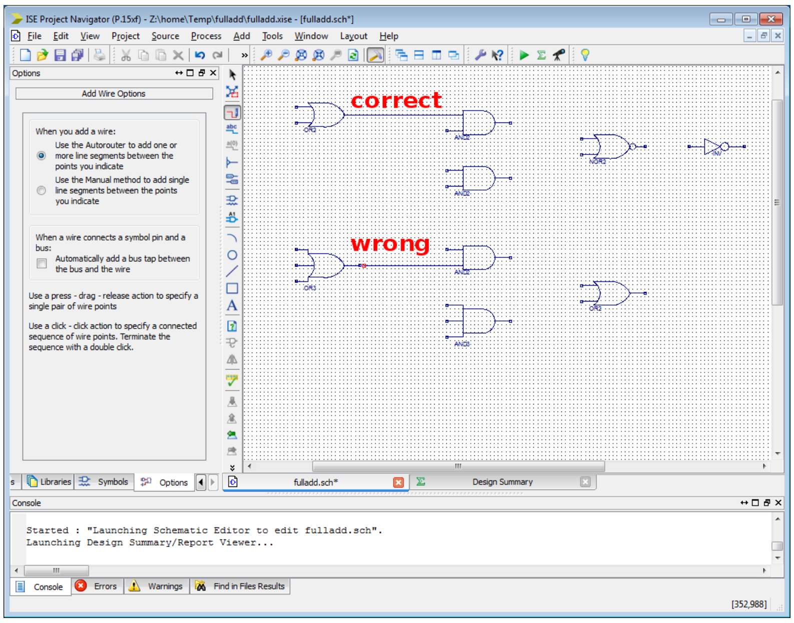
Orientation: Rotate 0

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

File Edit View Project Source Process Add Tools Window Layout Help
Symbols
Symbol Info
Libraries Symbols Options
fulladd.sch*
Design Summary
Console Errors Warnings Find in Files Results
[356,384]

Figure 18

Now you can connect an output port of one symbol with an input port of another symbol. The output ports are always shown on the right-hand side of a symbol. The input ports are shown on the left-hand side of a symbol. The *ports* are clearly marked with a small box. You should make sure that you connect the wire properly to the box, otherwise you will have a dangling wire. Figure 19 shows an example of a wire that is correctly connected and a wire that is incorrectly connected. Notice that the latter wire shows a small red box at the end where it is not connected. Since it is very difficult to see whether a wire is properly connected, you should always zoom-in sufficiently far to check that all wires are properly connected!


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]

Add Wire Options

When you add a wire:
- (●) Use the Autorouter to add one or more line segments between the points you indicate
- ( ) Use the Manual method to add single line segments between the points you indicate

When a wire connects a symbol pin and a bus:
- [ ] Automatically add a bus tap between the bus and the wire

Use a press - drag - release action to specify a single pair of wire points

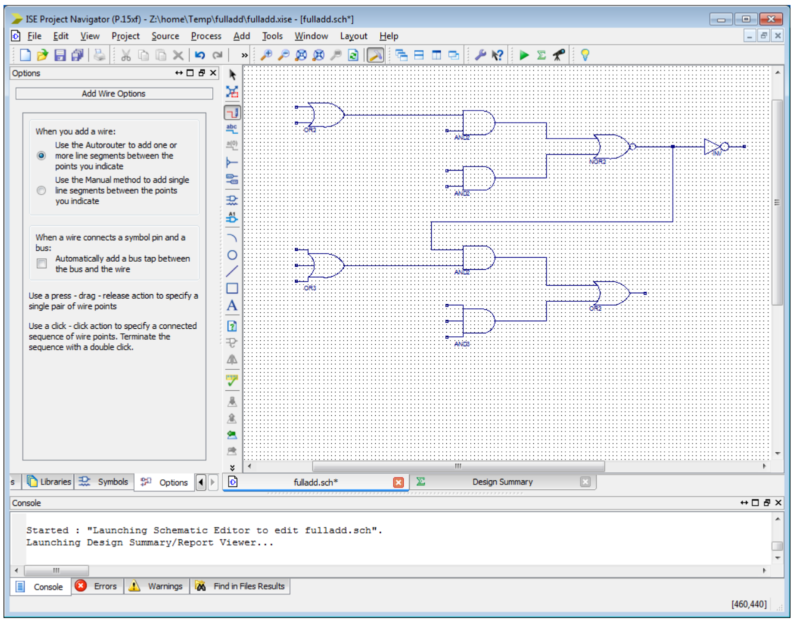
Use a click - click action to specify a connected sequence of wire points. Terminate the sequence with a double click.

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

correct
wrong
File Edit View Project Source Process Add Tools Window Layout Help
Options
Libraries Symbols Options
fulladd.sch*
Design Summary
Console Errors Warnings Find in Files Results
[352,988]

Figure 19

Now you should connect all wires in the same way as shown Figure 20. As mentioned before, you should make sure that all wires are connected properly.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Libraries
Symbols
Options
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[460,440]

Add Wire Options

When you add a wire:
[✓ Use the Autorouter to add one or more line segments between the points you indicate]
Use the Manual method to add single line segments between the points you indicate

When a wire connects a symbol pin and a bus:
Automatically add a bus tap between the bus and the wire

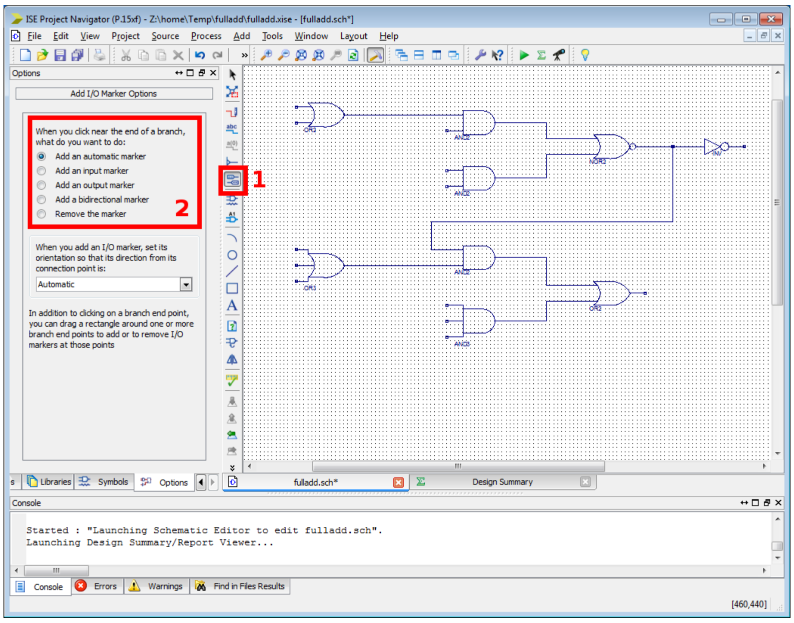
Use a press - drag - release action to specify a single pair of wire points

Use a click - click action to specify a connected sequence of wire points. Terminate the sequence with a double click.

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 20

At this moment, all inputs and outputs of our full adder are still unconnected. We need to add input and output markers to our design and we need to connect these markers to the unconnected ports of our symbols. You can do this by clicking on the *Add I/O marker* button (red box 1 in Figure 21). When you click on this button, the tab on the left-hand side will change face as you can also see in this figure. From the list shown in the red box number 2 of Figure 21 you should select the direction of the marker. In general, it is however recommendable to use the default option (i.e., add an automatic marker).


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
1
2
Libraries
Symbols
Options
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[460,440]

Add I/O Marker Options

When you click near the end of a branch, what do you want to do:
[✓ Add an automatic marker]
Add an input marker
Add an output marker
Add a bidirectional marker
Remove the marker

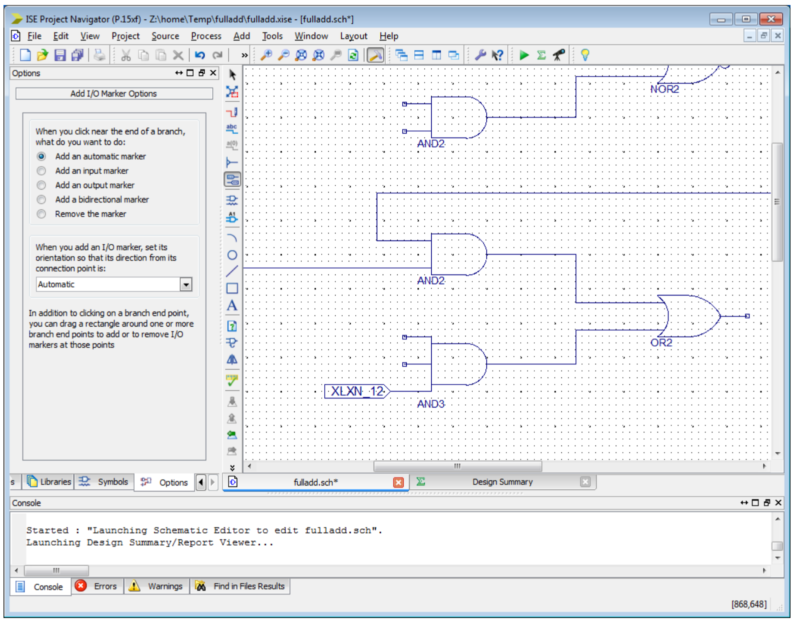
When you add an I/O marker, set its orientation so that its direction from its connection point is:
Automatic

In addition to clicking on a branch end point, you can drag a rectangle around one or more branch end points to add or to remove I/O markers at those points

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 21

We are now ready to add a marker to the design. Zoom-in on the `and3` symbol at the bottom of the design. Press the *Add I/O marker* button and connect an I/O marker to the lowest input of the `and3` symbol. This will create an I/O marker similar to the one shown in Figure 22.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Add I/O Marker Options
NOR2
AND2
AND2
OR2
XLXN_12
AND3
Libraries Symbols Options
fulladd.sch*
Design Summary
Console
Errors Warnings Find in Files Results
[868,648]

When you click near the end of a branch, what do you want to do:
- [✓ Add an automatic marker]
- Add an input marker
- Add an output marker
- Add a bidirectional marker
- Remove the marker

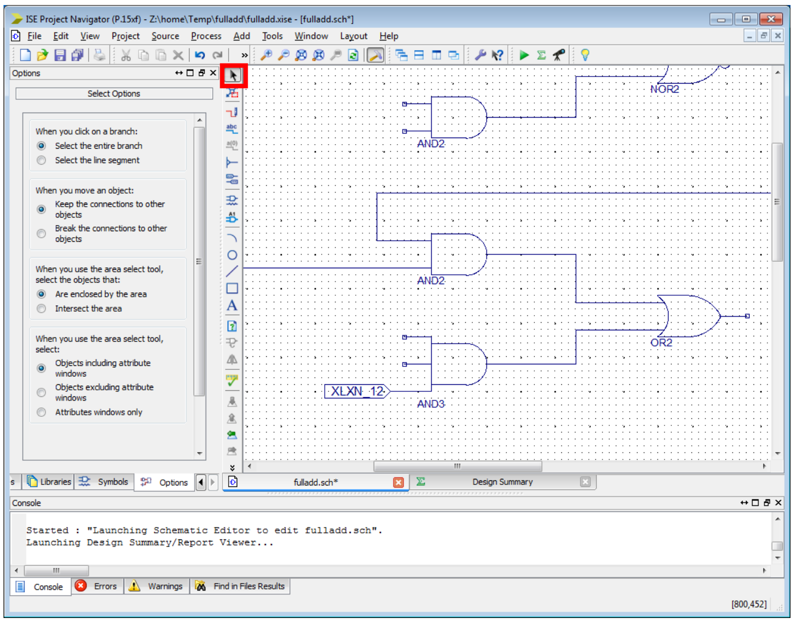
When you add an I/O marker, set its orientation so that its direction from its connection point is: Automatic

In addition to clicking on a branch end point, you can drag a rectangle around one or more branch end points to add or to remove I/O markers at those points

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 22

Xilinx assigns a "random" name to the newly created I/O marker. This is not very convenient when we would like to use our full adder in a larger design. Therefore we would like to rename the marker. This can be done by clicking on the *Select* button (red box in Figure 23).


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
NOR2
AND2
AND2
OR2
XLXN_12
AND3
Libraries Symbols Options
fulladd.sch*
Design Summary
Console
Errors Warnings Find in Files Results
[800,452]

When you click on a branch:
- [✓ Select the entire branch]
- Select the line segment

When you move an object:
- [✓ Keep the connections to other objects]
- Break the connections to other objects

When you use the area select tool, select the objects that:
- [✓ Are enclosed by the area]
- Intersect the area

When you use the area select tool, select:
- [✓ Objects including attribute windows]
- Objects excluding attribute windows
- Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 23

Next you should right-click on the marker. This will open a pull-down menu from which you need to select the option *Rename Port* (see Figure 24).

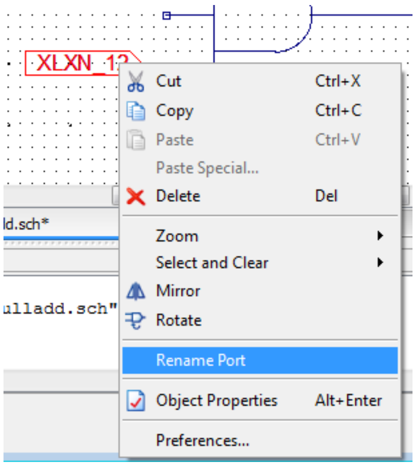


Figure 24

A new window (Figure 25) will open where you can change the name of the marker by typing a new name in the text box. You should confirm your change by clicking on the *Ok* button.

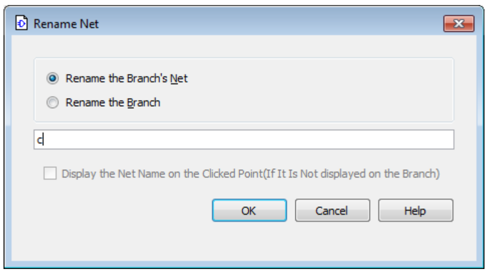
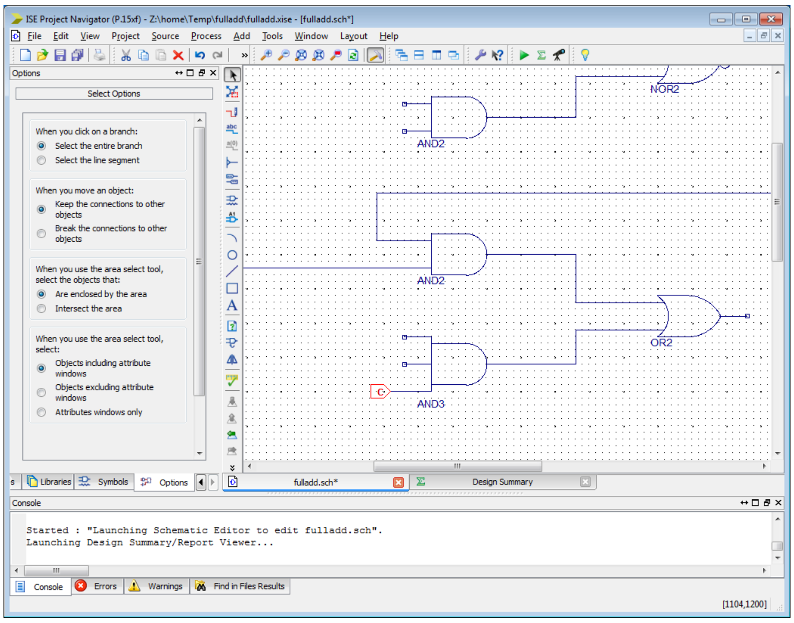

Rename Net
Rename the Branch's Net
Rename the Branch
d
Display the Net Name on the Clicked Point(If It Is Not displayed on the Branch)
OK
Cancel
Help

Figure 25

The main Xilinx ISE window will reappear and the port will now have it's new name. The port we have just added should be named c as you can see in Figure 26.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
NOR2
AND2
AND2
OR2
AND3
c
Libraries Symbols Options
fulladd.sch*
Design Summary
Console
Console Errors Warnings Find in Files Results
[1104,1200]

When you click on a branch:
- [✓ Select the entire branch]
- Select the line segment

When you move an object:
- [✓ Keep the connections to other objects]
- Break the connections to other objects

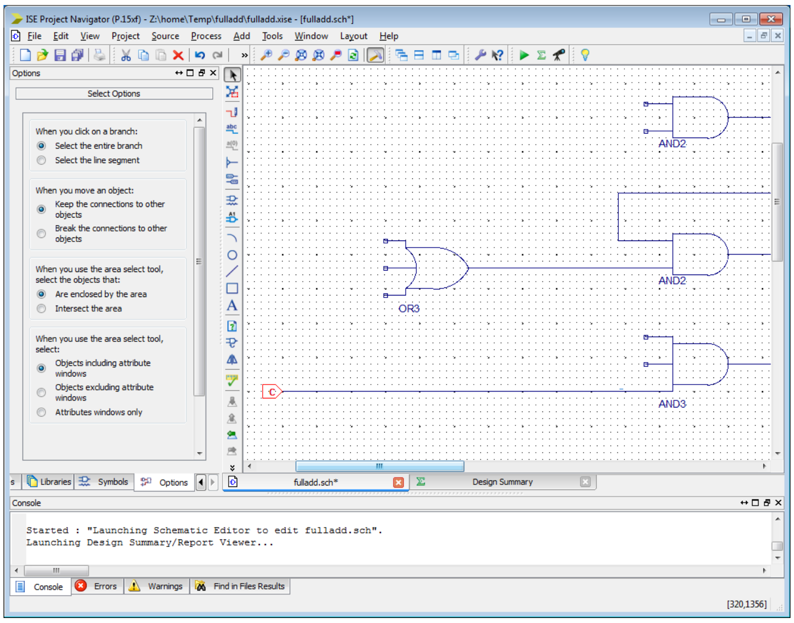
When you use the area select tool, select the objects that:
- [✓ Are enclosed by the area]
- Intersect the area

When you use the area select tool, select:
- [✓ Objects including attribute windows]
- Objects excluding attribute windows
- Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 26

In our full adder, the input c should not only be connected to this and3 port. It should also be connected to one of the or ports and one of the and ports. Therefore, it would be convenient if the marker would be positioned more towards the left of our schematic. You can move the marker by clicking on it with the left mouse button. When you keep the mouse button down, you can move the marker to a new location. (Note that moving symbols and markers only works when the *Select* button has been selected. This select button is shown in the red box on Figure 23.)


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
AND2
AND2
OR3
AND3
c
Libraries Symbols Options
fulladd.sch*
Design Summary
Console
Console Errors Warnings Find in Files Results
[320,1356]

When you click on a branch:
- [✓ Select the entire branch]
- Select the line segment

When you move an object:
- [✓ Keep the connections to other objects]
- Break the connections to other objects

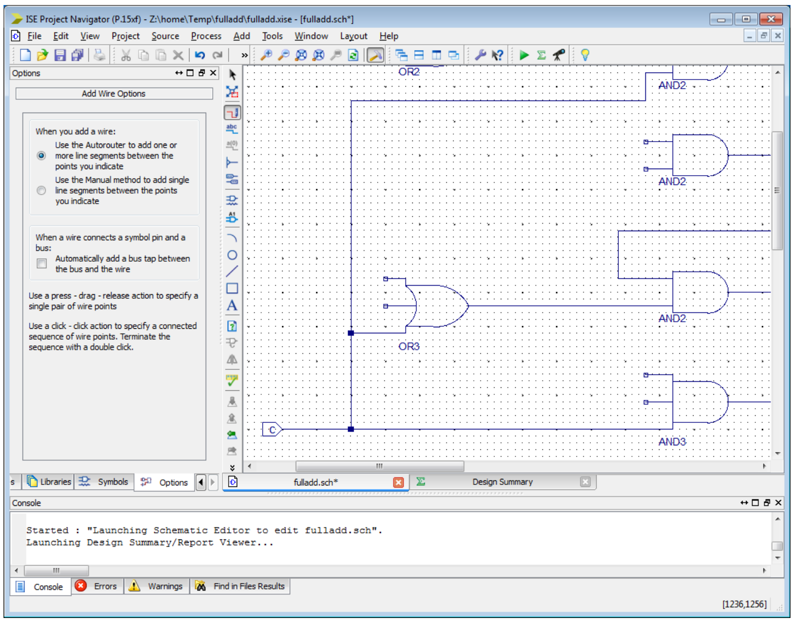
When you use the area select tool, select the objects that:
- [✓ Are enclosed by the area]
- Intersect the area

When you use the area select tool, select:
- [✓ Objects including attribute windows]
- Objects excluding attribute windows
- Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 27

Using the *Add wire* button - red box in Figure 18 - you can now connect the marker c to one of the inputs of the or3 symbol and the top and2 symbol.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
OR2
AND2
AND2
AND2
OR3
AND3
C
Libraries
Symbols
Options
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[1236,1256]

Add Wire Options

When you add a wire:
(●) Use the Autorouter to add one or more line segments between the points you indicate
( ) Use the Manual method to add single line segments between the points you indicate

When a wire connects a symbol pin and a bus:
[ ] Automatically add a bus tap between the bus and the wire

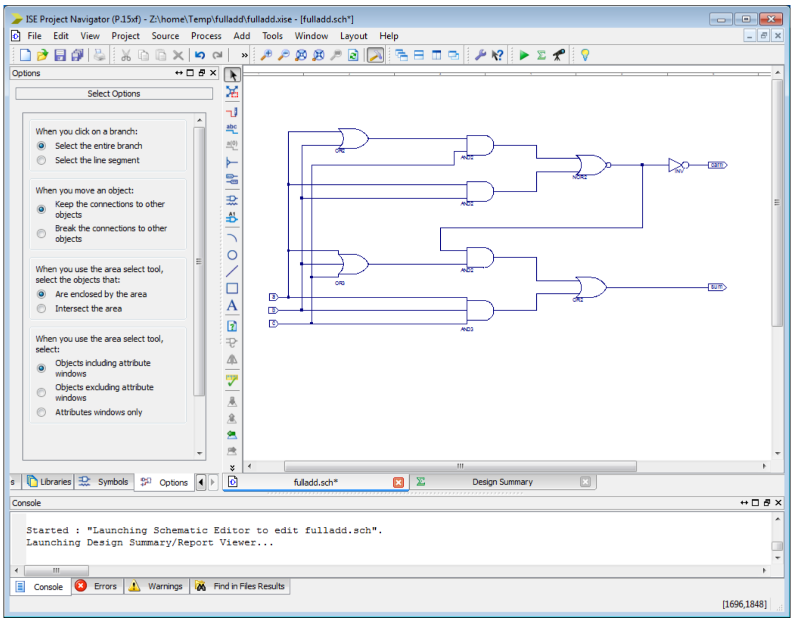
Use a press - drag - release action to specify a single pair of wire points

Use a click - click action to specify a connected sequence of wire points. Terminate the sequence with a double click.

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 28

Using the techniques you have learned in this section, you should now complete the design of the full adder. To do this you will need to add two additional input markers and two output markers. You should also give these markers meaningful names and connect them to the correct symbols. The resulting design should look as shown in Figure 29.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Libraries
Symbols
Options
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[1696,1848]

Select Options

When you click on a branch:
(●) Select the entire branch
( ) Select the line segment

When you move an object:
(●) Keep the connections to other objects
( ) Break the connections to other objects

When you use the area select tool, select the objects that:
(●) Are enclosed by the area
( ) Intersect the area

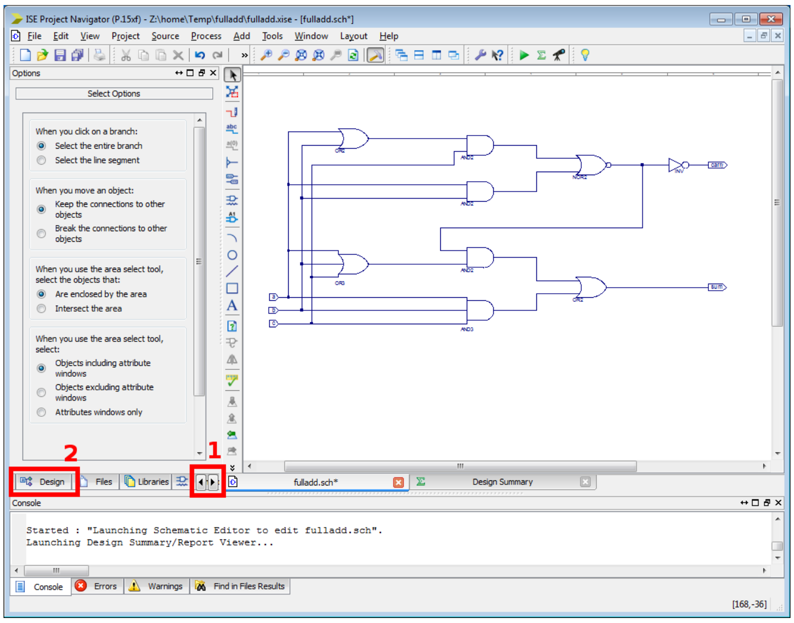
When you use the area select tool, select:
(●) Objects including attribute windows
( ) Objects excluding attribute windows
( ) Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 29

3 Simulating a design

Xilinx ISE allows you to simulate the functional behaviour of your design. In this section, we will show step-by-step how the full adder which you have build in the previous section can be simulated. As a first step, you need to switch to the *Design* tab. You can do this by clicking on the left and right arrow buttons shown in box 1 till the *Design* tab (box 2) comes into view. Next, you should click on the *Design* tab (box 2) to bring it into focus.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
2
1
Design
Files
Libraries
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[168,-36]

When you click on a branch:
[✓ Select the entire branch] Select the line segment

When you move an object:
[✓ Keep the connections to other objects] Break the connections to other objects

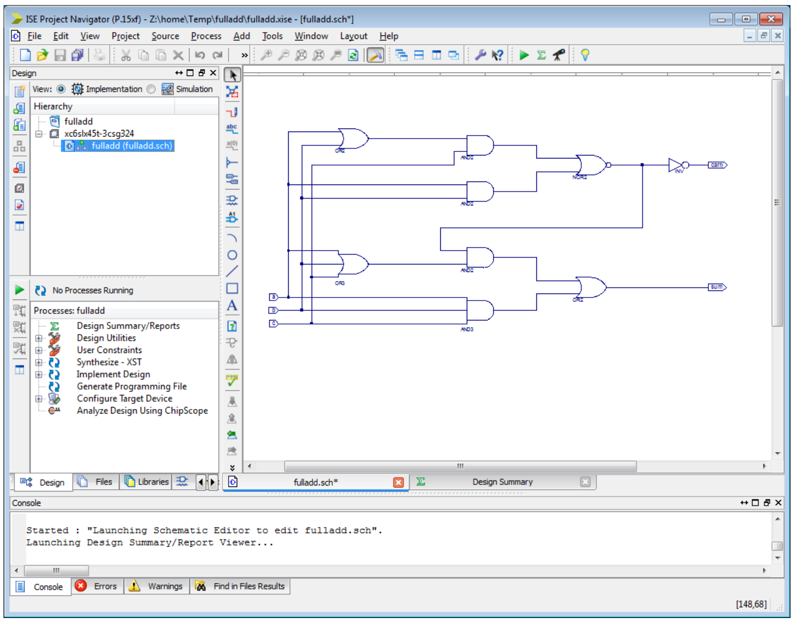
When you use the area select tool, select the objects that:
[✓ Are enclosed by the area] Intersect the area

When you use the area select tool, select:
[✓ Objects including attribute windows] Objects excluding attribute windows Attributes windows only

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 30

After clicking on the *Design* tab you will see the following window (Figure 31).


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Design
Hierarchy
No Processes Running
Design
Files
Libraries
fulladd.sch*
Design Summary
Console
Errors
Warnings
Find in Files Results
[148,68]

View: [✓ Implementation] Simulation

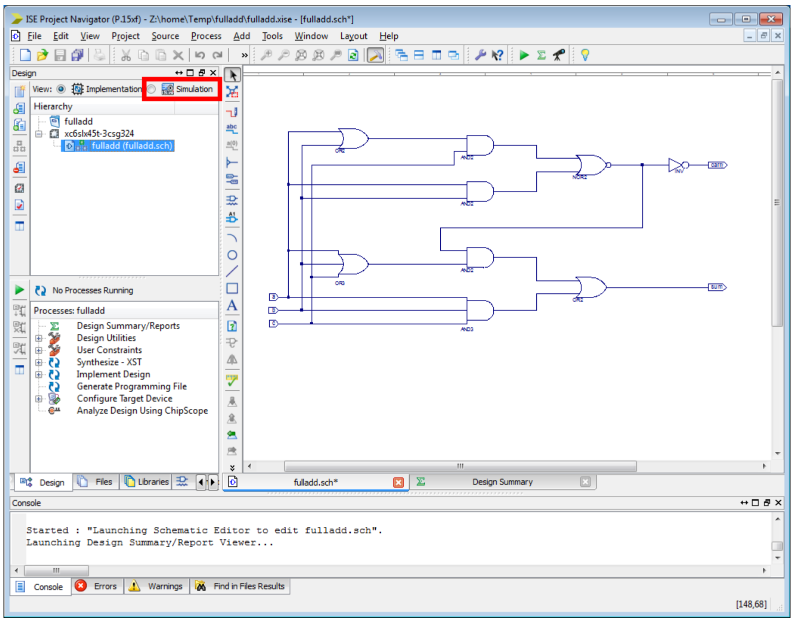
Hierarchy
- fulladd
- xc6slx45t-3csg324
  - fulladd (fulladd.sch)

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

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 31

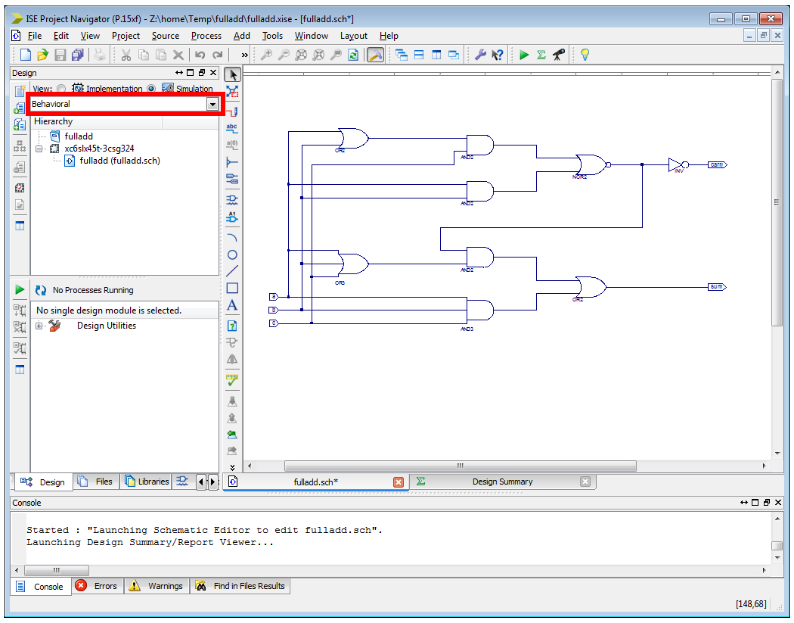
Using the the *Design* tab, you can simulate the design as well as implement the design in an FPGA. In this section, we will focus on simulating the design. The two radio buttons at the top of the tab allow you to switch between the implementation view and simulation view. You should click on the *simulation* button (red box in Figure 32) to switch to the simulation view.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
fulladd
xc6slx45t-3csg324
fulladd (fulladd.sch)
No Processes Running
Processes: fulladd
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope
Design Files Libraries
fulladd.sch*
Design Summary
Console
Console Errors Warnings Find in Files Results
[148,68]

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 32

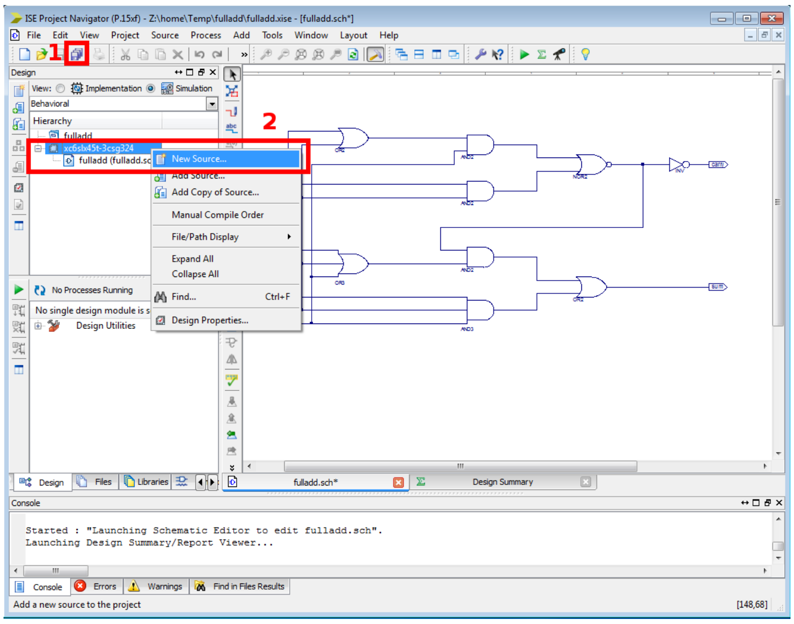
Once you have switched to the simulation view, you will see that a new pull down list (red box in Figure 33) appears just below the radio button that you have just pressed. This pull down list allows you to select different simulation modes. For this tutorial, you should make sure that it is set to `Behavioral` (which is the default option).


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [fulladd.sch*]
File Edit View Project Source Process Add Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
fulladd
xc6slx45t-3csg324
fulladd (fulladd.sch)
No Processes Running
No single design module is selected.
Design Utilities
Design Files Libraries
fulladd.sch*
Design Summary
Console
Console Errors Warnings Find in Files Results
[148,68]

```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 33

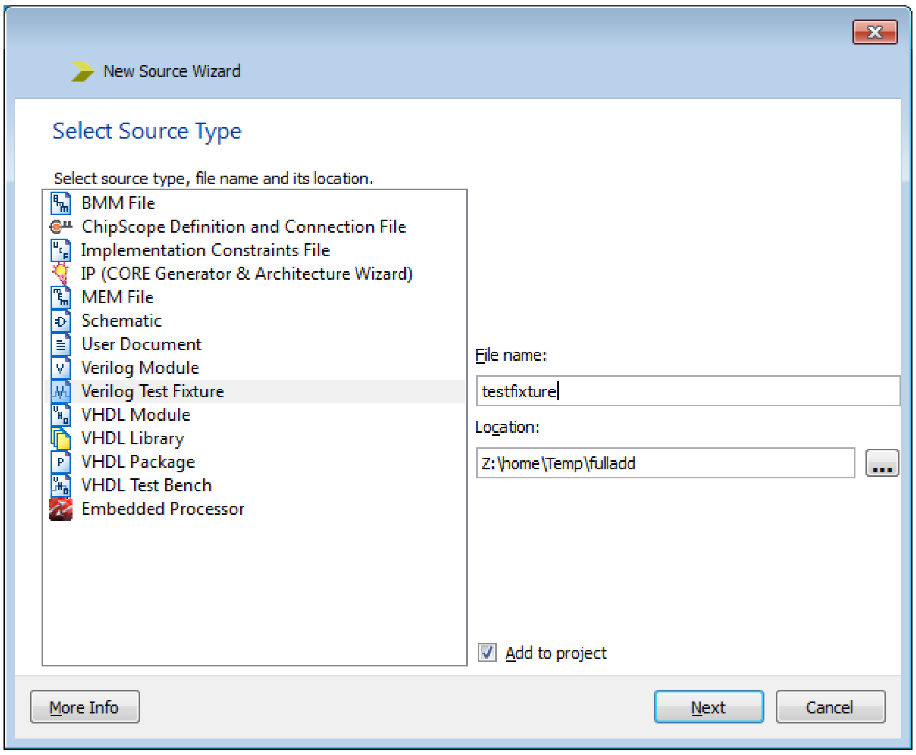
Before you can simulate the design, you need to create a *test fixture*. This test fixture specifies the values and timing of the input signals that need to be supplied to the design under test. To create this test fixture, you should first save the project by clicking on the save icon in the toolbar (box 1 in Figure 34). Next, you should right click on the element `x6slx45t-3csg324` which is visible in the *hierarchy* tree-view on the *Design* tab. You should select the option *New Source...* from the pull-down menu (box 2 in Figure 34).



```
Started : "Launching Schematic Editor to edit fulladd.sch".
Launching Design Summary/Report Viewer...
```

Figure 34

A new window (Figure 35) will open in which you need to select the *source type* of the file that you want to add. You should select `Verilog Test Fixture` as the source type. Next, you should provide a file name for this new file inside the *File name* box. In this tutorial, we will name the file `testfixture`. Once you have named the file you can click *Next* to continue.



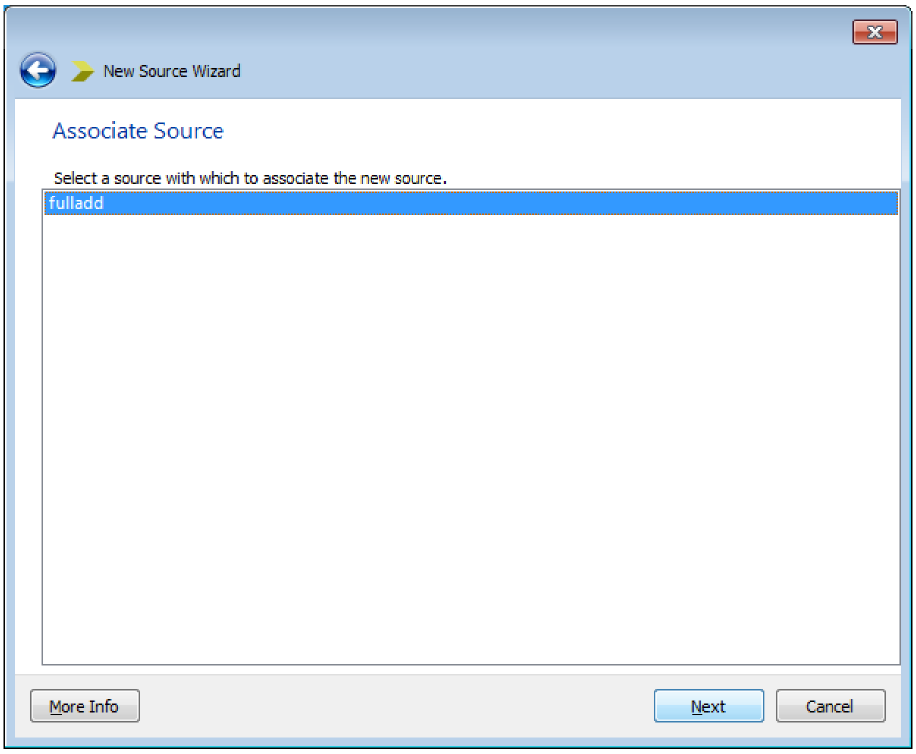
File name: testfixture

Location: Z:\home\Temp\fulladd

[✓] Add to project

Figure 35

The next window (Figure 36) shows you a list of all modules you have designed as part of this project. Since we have designed only module (i.e., fulladd), the list should only contain one module. The test fixture that we are going to create is meant for this module. Therefore you should select the module from the list by clicking on its name. The selection should be clearly visible since the module is highlighted in blue (as shown in the figure below). You can now move to the next window by clicking on the *Next* button.



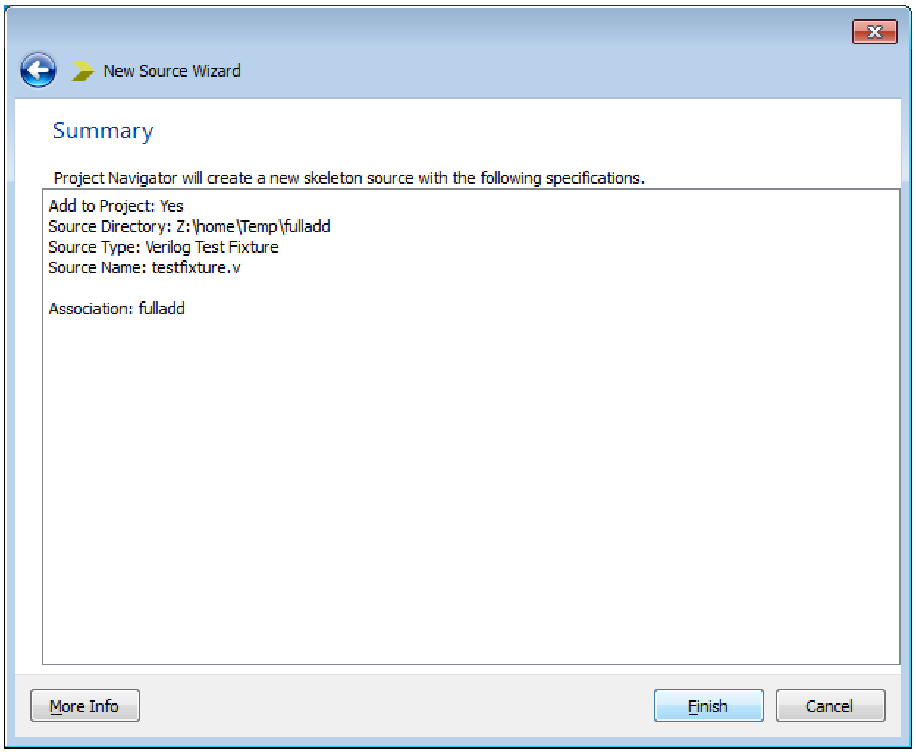
Associate Source

Select a source with which to associate the new source.

fulladd

Figure 36

This window (Figure 37) shows you a summary of the choices you have made. You should confirm these choices by clicking on the *Finish* button.

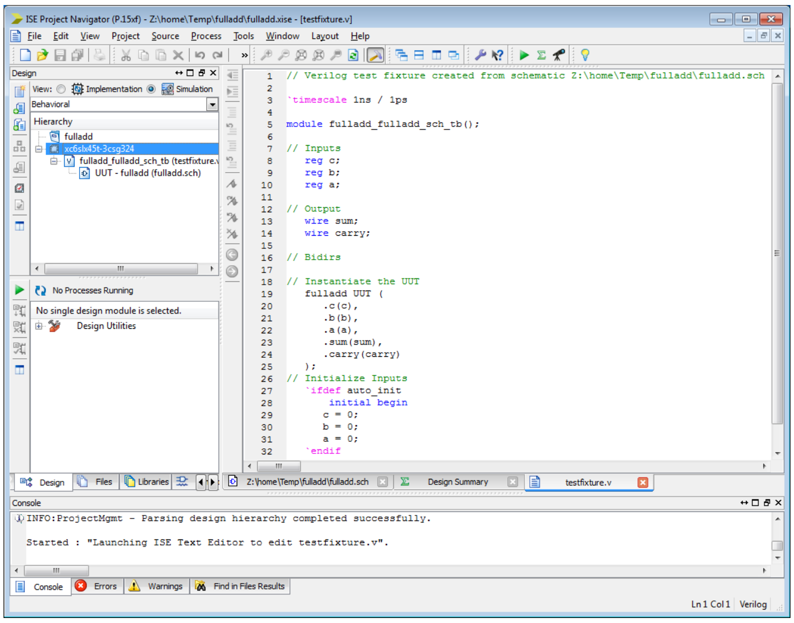


Project Navigator will create a new skeleton source with the following specifications.

```
Add to Project: Yes
Source Directory: Z:\home\Temp\fulladd
Source Type: Verilog Test Fixture
Source Name: testfixture.v

Association: fulladd
```

Figure 37

The view will return to the main window (Figure 38). However, the schematic that was previously visible in this window is now hidden behind another tab. This new tab contains an implementation of the test fixture for our *fulladd* module in the Verilog language.



```verilog
// Verilog test fixture created from schematic Z:\home\Temp\fulladd\fulladd.sch

`timescale 1ns / 1ps

module fulladd_fulladd_sch_tb();

// Inputs
   reg c;
   reg b;
   reg a;

// Output
   wire sum;
   wire carry;

// Bidirs

// Instantiate the UUT
   fulladd UUT (
      .c(c),
      .b(b),
      .a(a),
      .sum(sum),
      .carry(carry)
   );
// Initialize Inputs
   `ifdef auto_init
       initial begin
      c = 0;
      b = 0;
      a = 0;
   `endif
```

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

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

Figure 38

This test fixture has been generated automatically by Xilinx. Before we can run the simulator, we need to make some changes to the test fixture. You should remove the line containing the following statement:

```
'ifdef auto_init
```

You should also replace the line that contains the following statement:

```
'endif
```

with the following statement:

```
end
```

Now we are ready to run the simulator. However, the simulation would in this case not be very meaningful since the test fixture would initialize all input signals to zero at the start of the simulator and it would never change these input values throughout the whole simulation. We should therefore extend the test fixture to cause some signals to change at specific points in time. After the `end` statement that you have just added to the file (line 31 in our example) you should insert the following Verilog statements.

```
always #50
begin
  a = ~a;
end
always @(posedge a)
begin
  b = ~b;
end
always @(posedge b)
begin
  c = ~c;
end
```

This Verilog code uses so called *always* blocks to define when the inputs change value. The first block ('`always #50`') will be executed once every 50 time units. As you can see at the top of the file (line 3), a single time unit is equal to 1 ns. Hence, the signals mentioned in the '`always #50`' block will change value once every 50 ns. In our example, the input `a` will become equal to the inverse of its previous value. The second block ('`always @(posedge a)`') will always be executed when the input `a` goes from 0 to 1. Similarly, the third block ('`always @(posedge c)`') will be executed when the input `b` goes from 0 to 1.[1]

Instead of using the *always* blocks, Verilog provides an alternative way to specify when an input needs to change value. The following Verilog code shows how this can be done[2]. Similar to the Verilog code that you have just modified, this code starts with initializing the three inputs. The three initialization statements are followed by a line containing a '#50' statement. This statement tells the simulator to wait for 50 time units (i.e., 50 ns). Once this delay has passed, the value of

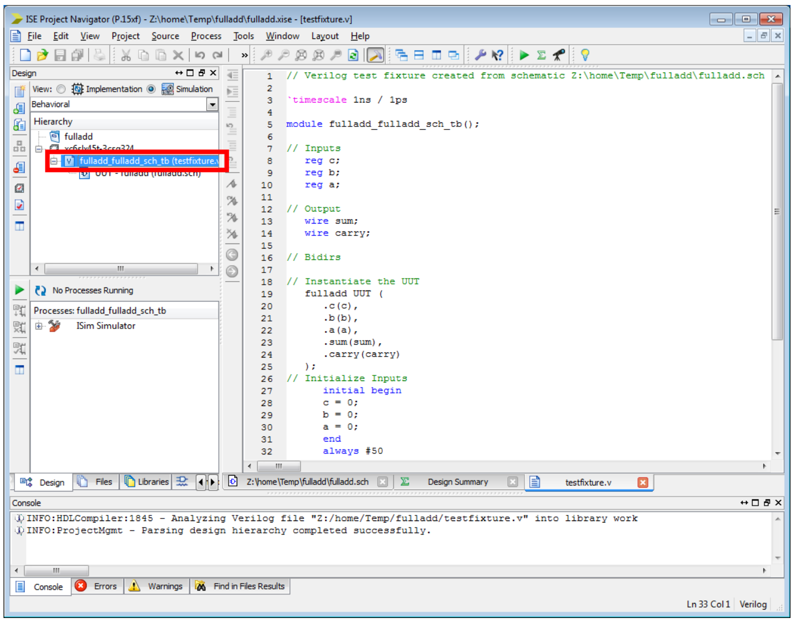
[1]Note that input `b` changes value with half the frequency of input `a` and input `c` changes value with half the frequency of input `b`. As a result, each set of 8 subsequent inputs contains all possible value combinations our three inputs may take. You can easily extend this scheme to more inputs and easily generate all possible input combinations for those circuits.

[2]You should not copy this code in your test fixture. It is only shown in this tutorial to demonstrate this alternative option. We will not use this option in this tutorial.

the input a should become the inverse of its current value. The statement '#100' then specifies that the simulator should wait for another 100 ns. So when a time period of 150 ns (= 50 ns + 100 ns) has passed, the value of b should change. Another 50 ns later, inputs a and c will change value once more. After those inputs have changed value, the inputs will never change value any more independent on how long the simulation is continued.

```
initial begin
a = 0;
b = 0;
c = 0;
#50
a = ~a;
#100
b = ~b;
#50
a = ~a;
c = ~c;
end
```

You are now ready ready to simulate the design. As a first step you should click on the module that needs to be simulated in the hierarchy tree-view (red box in Figure 39).

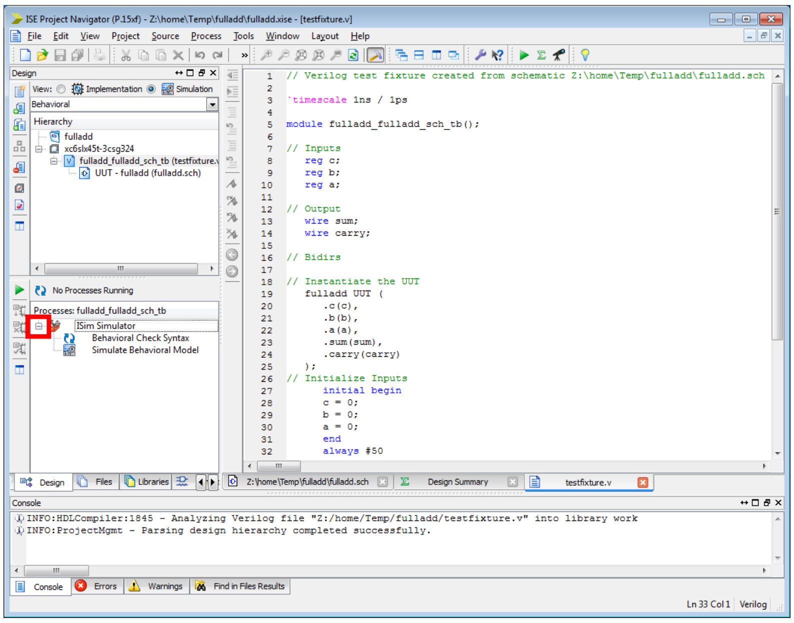

ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [testfixture.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
fulladd
xc6slx45t-3csg324
fulladd_fulladd_sch_tb (testfixture.v
UUT - fulladd (fulladd.sch)
No Processes Running
Processes: fulladd_fulladd_sch_tb
ISim Simulator
Design Files Libraries
Z:\home\Temp\fulladd\fulladd.sch
Design Summary
testfixture.v
Console
Console Errors Warnings Find in Files Results
Ln 33 Col 1 Verilog

```
// Verilog test fixture created from schematic Z:\home\Temp\fulladd\fulladd.sch

`timescale 1ns / 1ps

module fulladd_fulladd_sch_tb();

// Inputs
   reg c;
   reg b;
   reg a;

// Output
   wire sum;
   wire carry;

// Bidirs

// Instantiate the UUT
   fulladd UUT (
      .c(c),
      .b(b),
      .a(a),
      .sum(sum),
      .carry(carry)
   );
// Initialize Inputs
      initial begin
      c = 0;
      b = 0;
      a = 0;
      end
      always #50
```

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "Z:/home/Temp/fulladd/testfixture.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

Figure 39

Next you should expand the tree inside the processes tree view (red box in Figure 40).

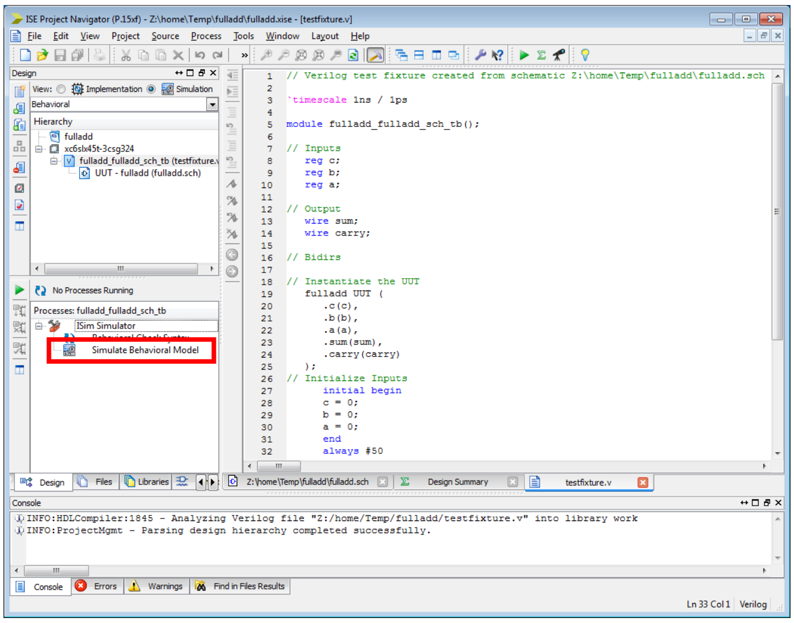

ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [testfixture.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
fulladd
xc6slx45t-3csg324
fulladd_fulladd_sch_tb (testfixture.v
UUT - fulladd (fulladd.sch)
No Processes Running
Processes: fulladd_fulladd_sch_tb
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
Design Files Libraries
Z:\home\Temp\fulladd\fulladd.sch
Design Summary
testfixture.v
Console
Console Errors Warnings Find in Files Results
Ln 33 Col 1 Verilog

```verilog
// Verilog test fixture created from schematic Z:\home\Temp\fulladd\fulladd.sch

`timescale 1ns / 1ps

module fulladd_fulladd_sch_tb();

// Inputs
   reg c;
   reg b;
   reg a;

// Output
   wire sum;
   wire carry;

// Bidirs

// Instantiate the UUT
   fulladd UUT (
      .c(c),
      .b(b),
      .a(a),
      .sum(sum),
      .carry(carry)
   );
// Initialize Inputs
      initial begin
      c = 0;
      b = 0;
      a = 0;
      end
      always #50
```

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "Z:/home/Temp/fulladd/testfixture.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

Figure 40

You can now run the simulator by double-clicking on the *Simulate Behavioral Model* text inside the processes tree view (red box in Figure 41).

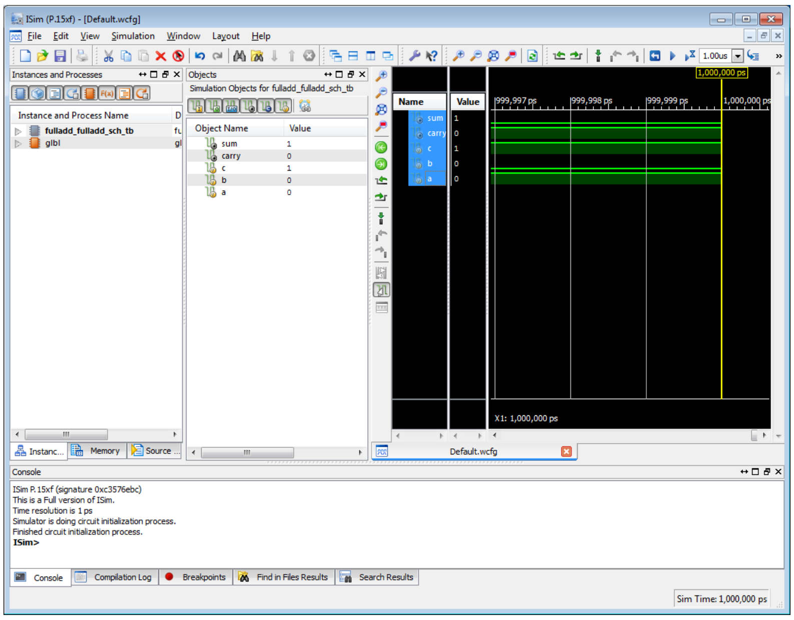

ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd\fulladd.xise - [testfixture.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Behavioral
Hierarchy
fulladd
xc6slx45t-3csg324
fulladd_fulladd_sch_tb (testfixture.v
UUT - fulladd (fulladd.sch)
No Processes Running
Processes: fulladd_fulladd_sch_tb
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
Design Files Libraries
Z:\home\Temp\fulladd\fulladd.sch
Design Summary
testfixture.v
Console
Console Errors Warnings Find in Files Results
Ln 33 Col 1 Verilog

```verilog
// Verilog test fixture created from schematic Z:\home\Temp\fulladd\fulladd.sch

`timescale 1ns / 1ps

module fulladd_fulladd_sch_tb();

// Inputs
   reg c;
   reg b;
   reg a;

// Output
   wire sum;
   wire carry;

// Bidirs

// Instantiate the UUT
   fulladd UUT (
      .c(c),
      .b(b),
      .a(a),
      .sum(sum),
      .carry(carry)
   );
// Initialize Inputs
      initial begin
      c = 0;
      b = 0;
      a = 0;
      end
      always #50
```

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "Z:/home/Temp/fulladd/testfixture.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

Figure 41

After a few second, a new window with the simulation results will open. It should look similar to the window shown in Figure 42.


ISim (P.15xf) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Instances and Processes
Instance and Process Name
fulladd_fulladd_sch_tb
glbl
Objects
Simulation Objects for fulladd_fulladd_sch_tb
Object Name Value
Name Value
X1: 1,000,000 ps
Default.wcfg
Console Compilation Log Breakpoints Find in Files Results Search Results
Sim Time: 1,000,000 ps

Object Name	Value
sum	1
carry	0
c	1
b	0
a	0

```
ISim P.15xf (signature 0xc3576ebc)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 42

The windows contains several panels. The panel which is visible in box 1 is called the *Source File* panel. It allows you to view the source code of the various modules in the design. In our example, you can use it to view the Verilog code of the test fixture. The panel which is shown in box 2 is called the *Object* panel. This panel shows all signals that have been simulated. In our example, it contains the input and output ports of our `fulladd` module. Box 3 shows the *Simulation* panel. This panel shows the value of all signals at each moment in time. We will study the content of this panel in more detail in the remainder of this section. The last panel (box 4) is called the *Console* panel. It shows some messages that are generated by the simulator. We will ignore these messages in this tutorial.

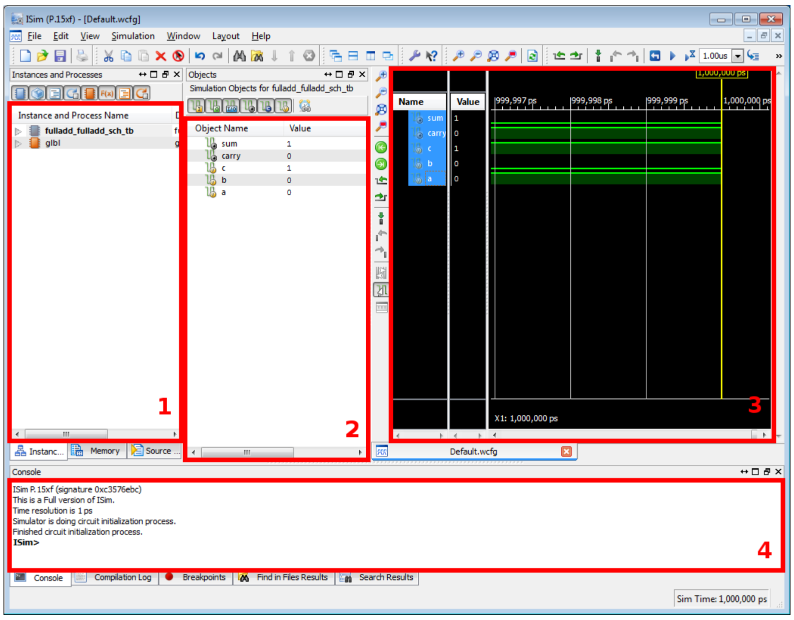
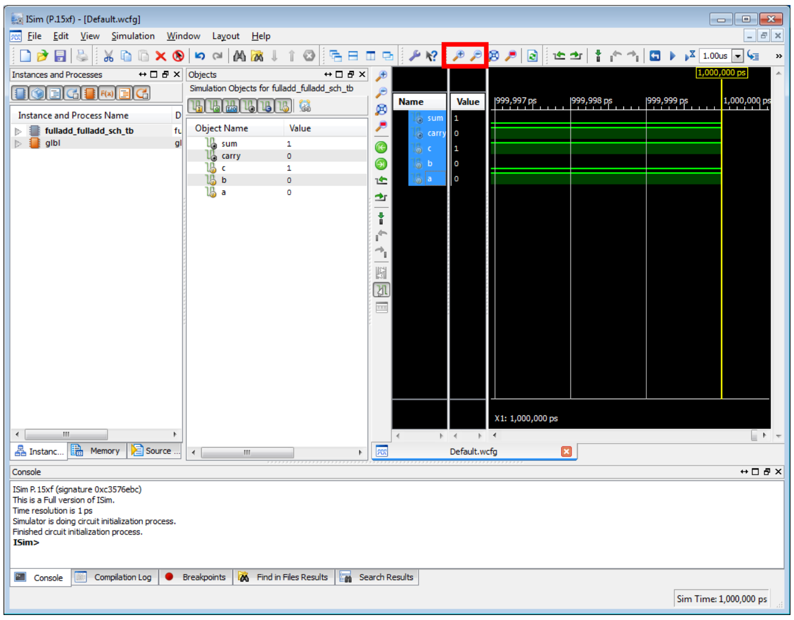


Figure 43

Let's take a more detailed look at the simulation panel (box 3). Using the zoom buttons shown in the red box of Figure 44 you can zoom in and out of the signal traces. When you zoom out a number of time you will see that the complete simulation trace (up-to 1 microsecond) becomes visible in the window. At this moment, it becomes however very hard to see the signal changes as they now happen so fast that you can only see a few blurred green bars. Using the zoom in button you can take a more detailed look at the signal and see how it changes over time.



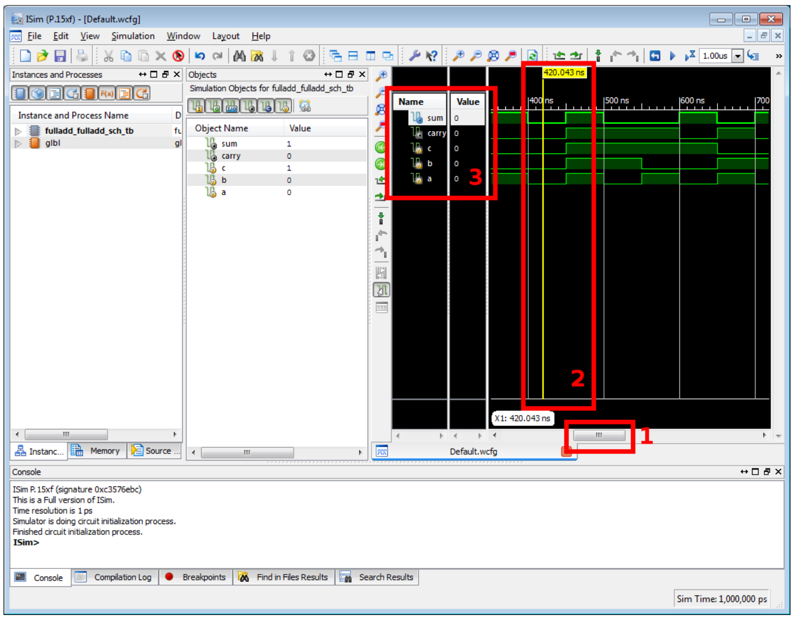
Object Name	Value
sum	1
carry	0
c	1
b	0
a	0

```
ISim P.15xf (signature 0xc3576ebc)
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

Figure 44

Using the horizontal slider (box 1 in Figure 45) you can move through time. You can also inspect the signals at a specific time instance. To do this, you should click on one of the signals. You will see a yellow line appearing at the time instance which you selected with your mouse click (see box 2 in Figure 45). In box 3 you can now read the values of the various signals at this particular time instance. In this example, all signals (both input and output) have value 0.



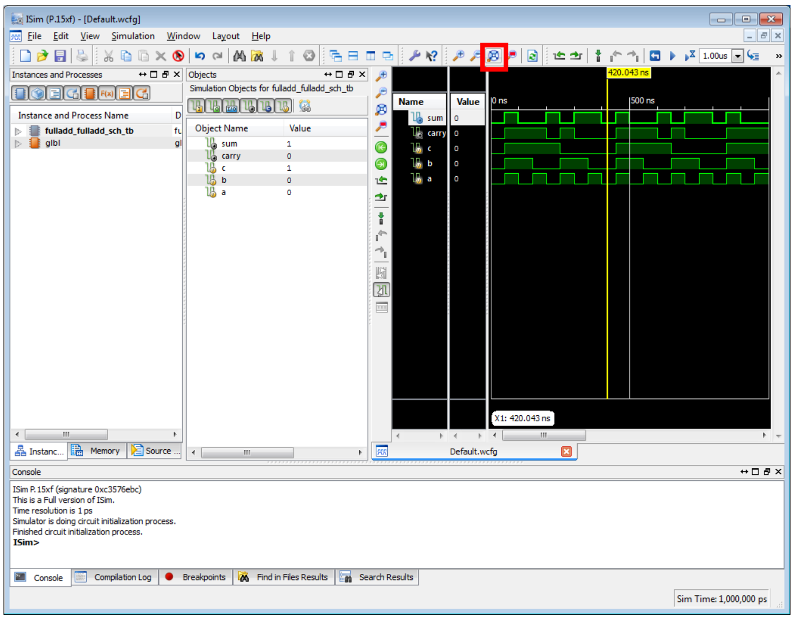
Name	Value
sum	0
carry	0
c	0
b	0
a	0

```
ISim P.15xf (signature 0xc3576ebc)
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

Figure 45

It might sometime be convenient to view the complete simulation in the panel. Using the button shown in the red box of Figure 46 you can fit the simulation trace to the panel.

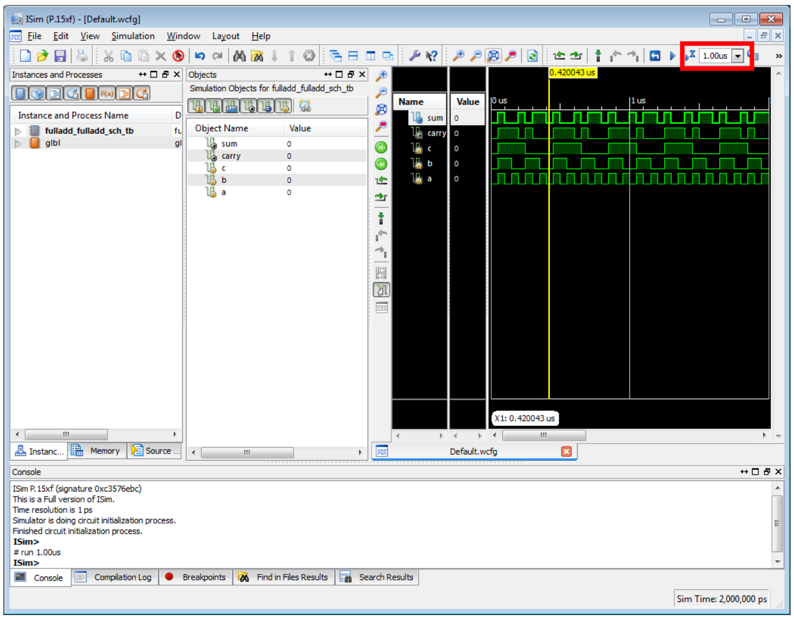


Object Name	Value
sum	1
carry	0
c	1
b	0
a	0

```
ISim P.15xf (signature 0xc3576ebc)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 46

By default, Xilinx simulates the design for 1 μs. This is sufficient for our example as all possible input combinations can be simulated within this time period. However sometimes you may need to run the simulator for a longer period of time. In that case you can continue the simulation for another 1 μs by clicking the *play* button shown in the red box of Figure 47.



Object Name	Value
sum	0
carry	0
c	0
b	0
a	0

```
ISim P.15xf (signature 0xc3576ebc)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
# run 1.00us
ISim>
```

Figure 47

4 Creating a new Verilog project

You can start a new project by selecting the option `File/New Project...` from the menu.

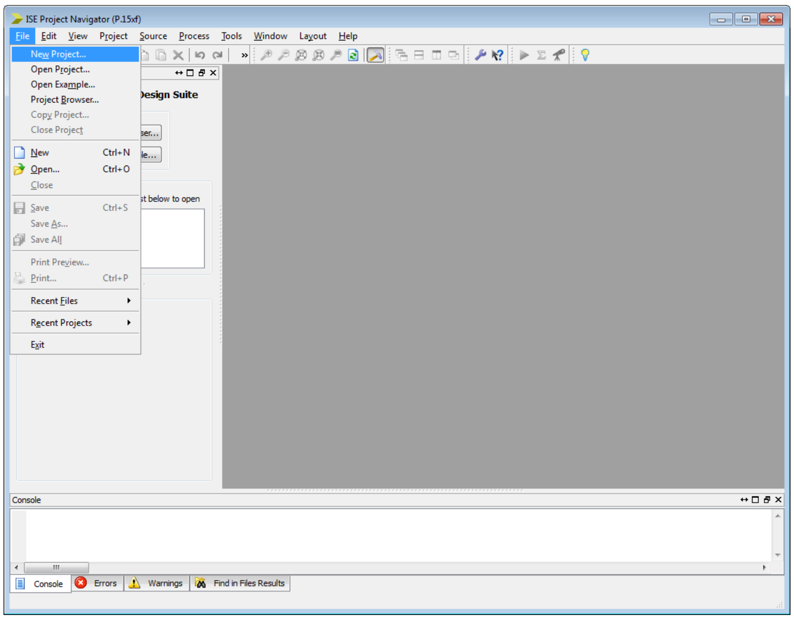
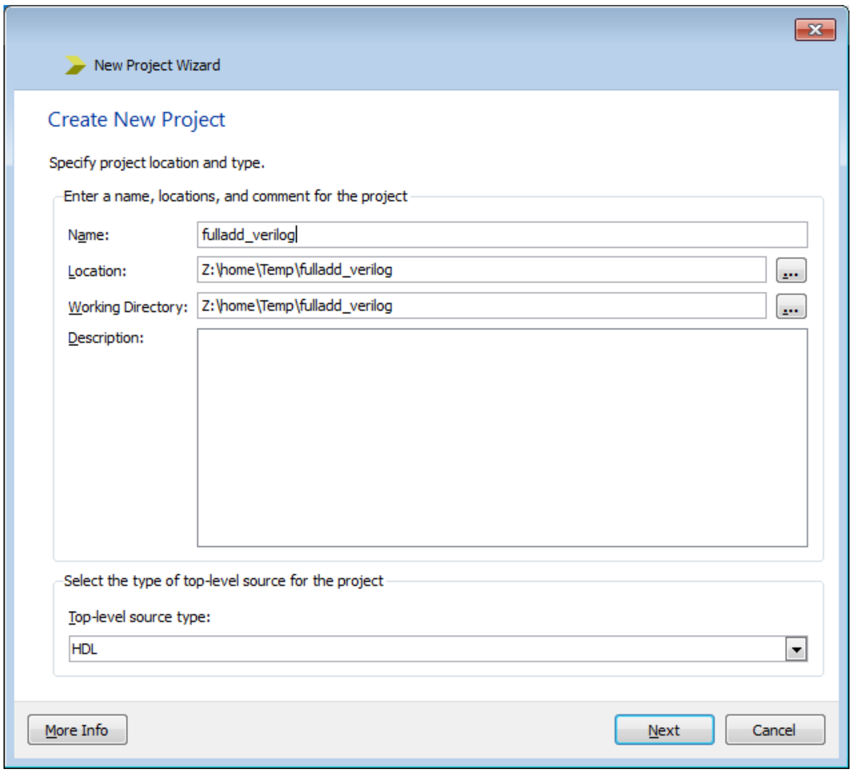


Figure 48

A new window (Figure 49) opens were you need to supply the name of the project (*fulladd_verilog*) and the location where the project files will be stored. You should also make sure that the option `HDL` is selected as *Top-level source type*. Click *Next* to continue.


New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: fulladd_verilog

Location: Z:\home\Temp\fulladd_verilog

Working Directory: Z:\home\Temp\fulladd_verilog

Description:

Select the type of top-level source for the project

Top-level source type:

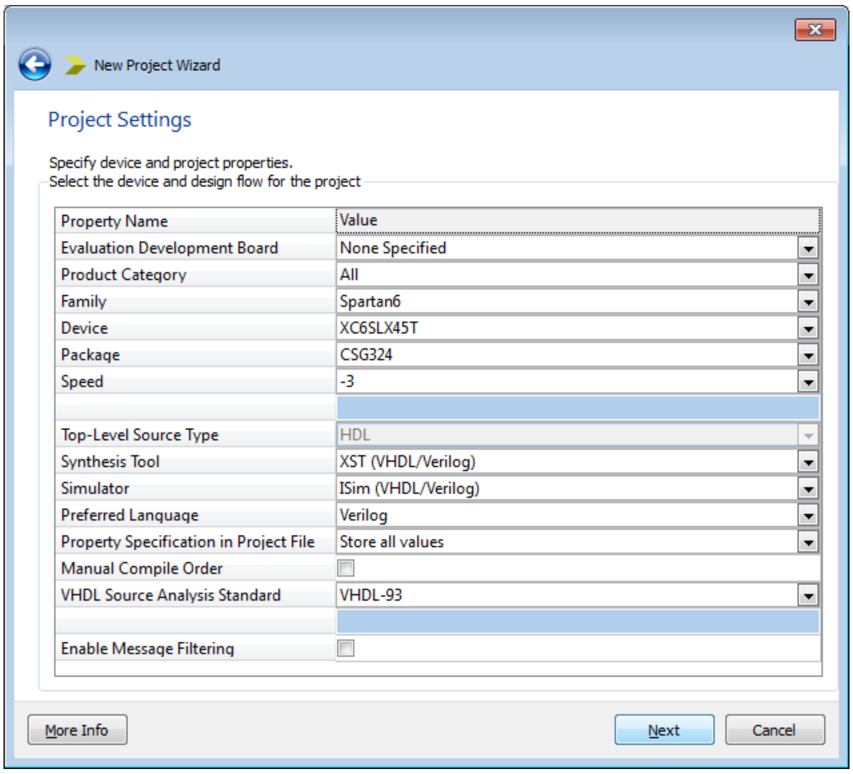
HDL

Figure 49

In the next window (Figure 50), you need to supply the details of the project. You should select the following options:

- Family: `Spartan6`
- Device: `XC6SLX45T`
- Package: `CSG324`
- Speed: *-3*
- Synthesis Tool: `XST (VHDL/Verilog)`
- Simulator: `ISim (VHDL/Verilog)`

Click *Next* to continue.



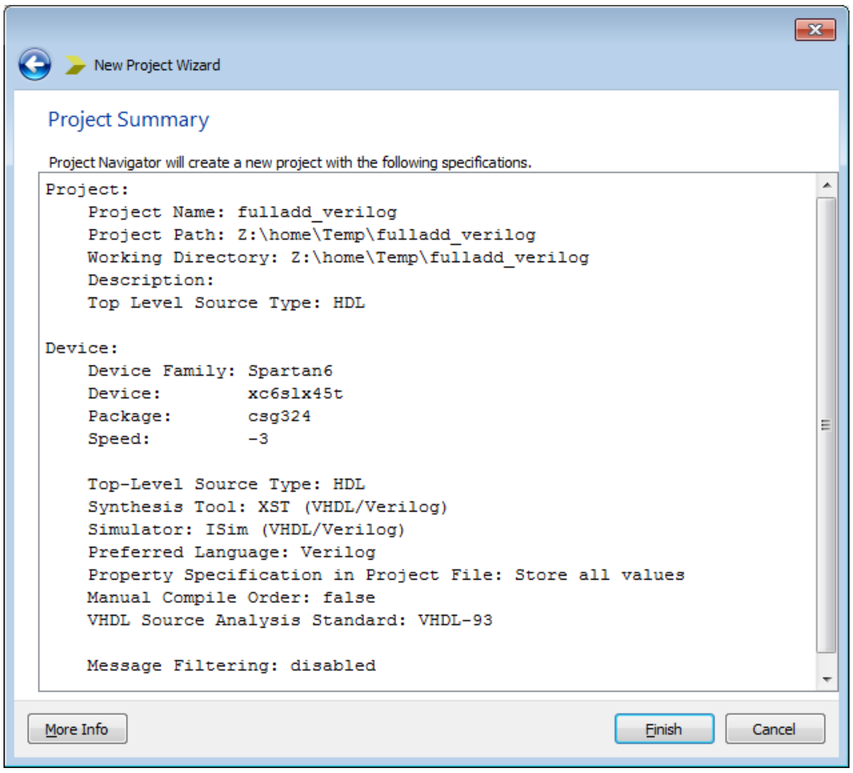
Project Settings

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	Spartan6
Device	XC6SLX45T
Package	CSG324
Speed	-3
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-93
Enable Message Filtering	☐

Figure 50

Click *Finish* on the next window (Figure 51).

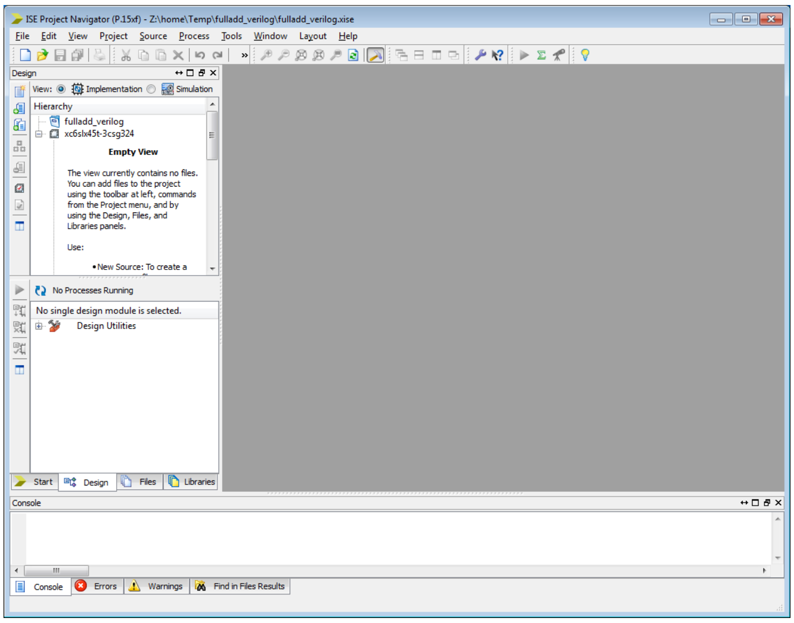


Project Navigator will create a new project with the following specifications.

```
Project:
    Project Name: fulladd_verilog
    Project Path: Z:\home\Temp\fulladd_verilog
    Working Directory: Z:\home\Temp\fulladd_verilog
    Description:
    Top Level Source Type: HDL

Device:
    Device Family: Spartan6
    Device:        xc6slx45t
    Package:       csg324
    Speed:         -3

    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: false
    VHDL Source Analysis Standard: VHDL-93

    Message Filtering: disabled
```

Figure 51

Once the project is created, it will be automatically opened as shown Figure 52.


ISE Project Navigator (P.15xf) - Z:\home\Temp\fulladd_verilog\fulladd_verilog.xise
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
fulladd_verilog
xc6slx45t-3csg324
No Processes Running
Design Utilities
Start Design Files Libraries
Console
Console Errors Warnings Find in Files Results

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

No single design module is selected.

Figure 52

The project is initially empty. You need to a add an empty Verilog file to the project by selecting the option `Project/New Source...` from the menu (see Figure 53).

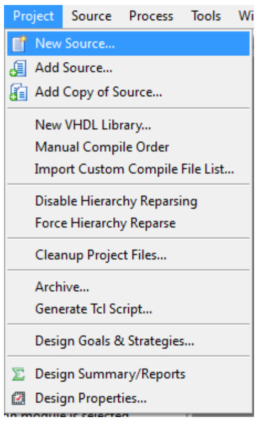
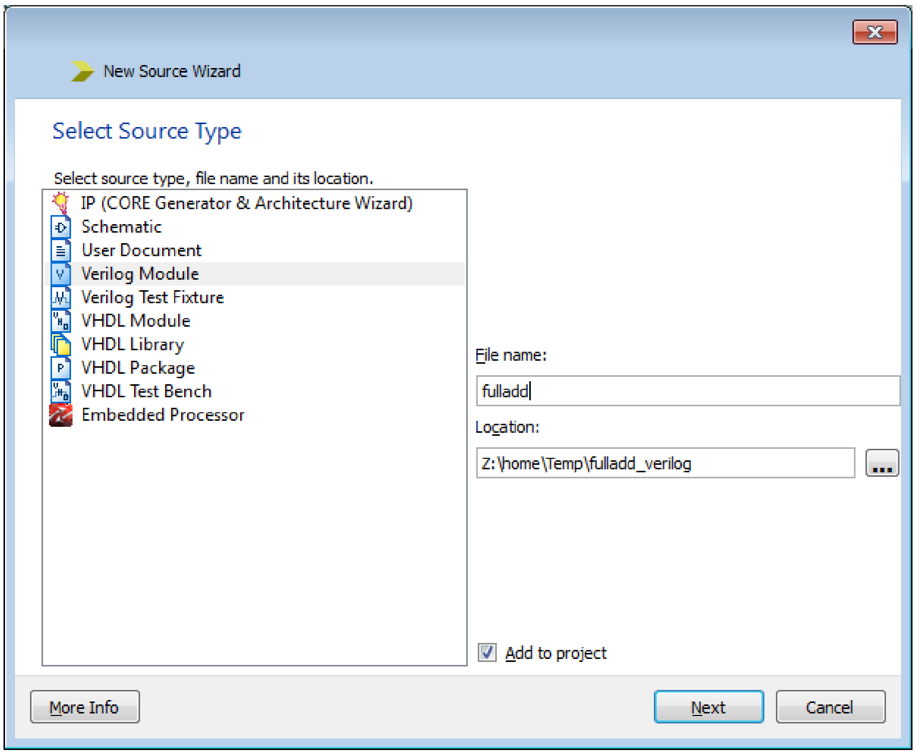


Figure 53

A new window (Figure 54) appears in which you should select `Verilog Module` as the source type to be created. You should also give the file a meaningful name, e.g., `fulladd`. Click *Next* to continue.



Select source type, file name and its location.

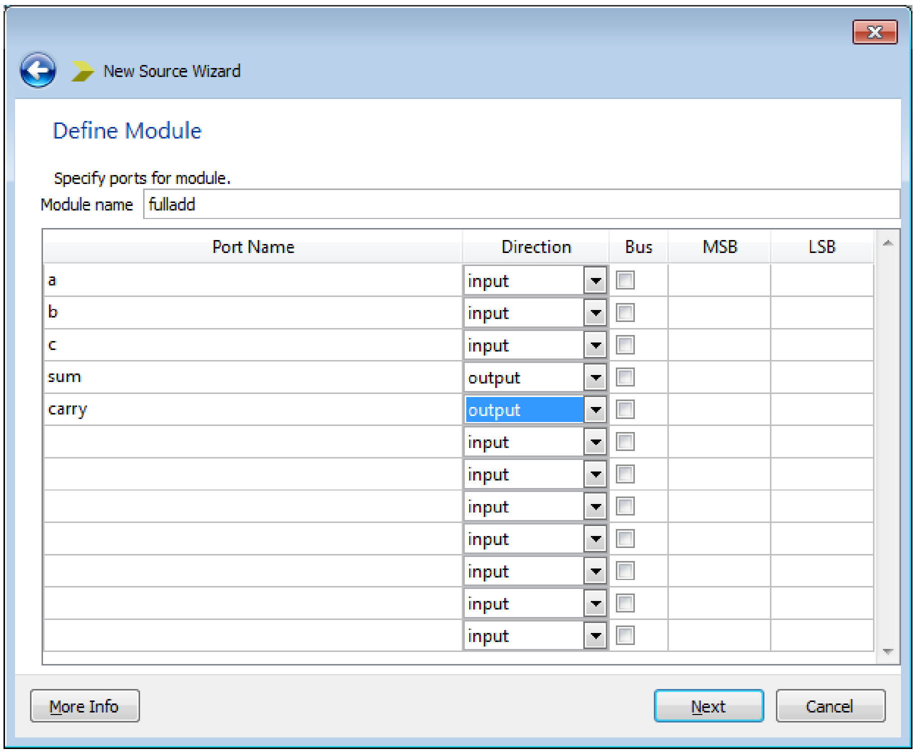
File name: fulladd

Location: Z:\home\Temp\fulladd_verilog

[✓] Add to project

Figure 54

On the next window (Figure 55) you will be asked to define all ports of the module as well as their direction (input or output). You should add all ports of the full adder and set the correct directions in this window.



Define Module

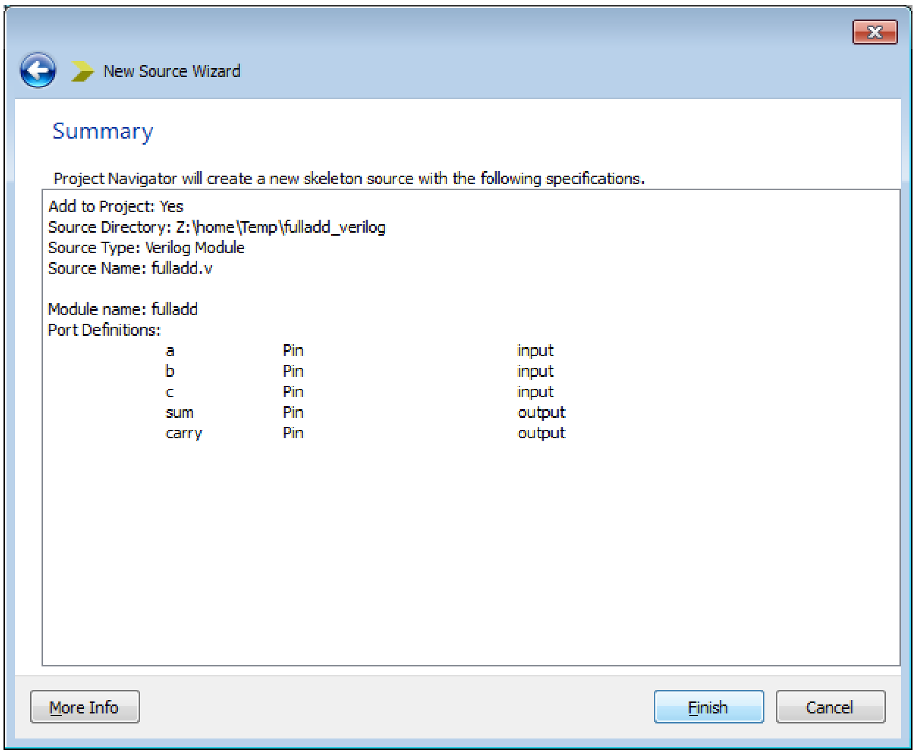
Specify ports for module.

Module name fulladd

Port Name	Direction	Bus	MSB	LSB
a	input	☐		
b	input	☐		
c	input	☐		
sum	output	☐		
carry	output	☐		
	input	☐		
	input	☐		
	input	☐		
	input	☐		
	input	☐		
	input	☐		
	input	☐		

Figure 55

On the next window (Figure 56), you should click *Finish* to confirm the creation of the new file.



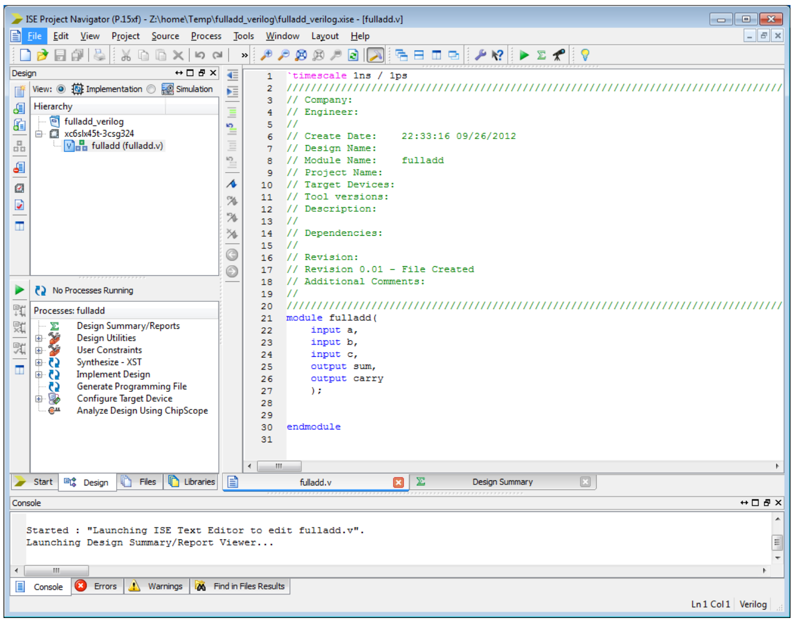
Summary

Project Navigator will create a new skeleton source with the following specifications.

```
Add to Project: Yes
Source Directory: Z:\home\Temp\fulladd_verilog
Source Type: Verilog Module
Source Name: fulladd.v

Module name: fulladd
Port Definitions:
             a         Pin                    input
             b         Pin                    input
             c         Pin                    input
             sum       Pin                    output
             carry     Pin                    output
```

Figure 56

After creating the source file, the view of the main window will change. It should look like the window shown in Figure 57.



```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    22:33:16 09/26/2012
// Design Name:
// Module Name:    fulladd
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
module fulladd(
    input a,
    input b,
    input c,
    output sum,
    output carry
    );


endmodule
```

```
Started : "Launching ISE Text Editor to edit fulladd.v".
Launching Design Summary/Report Viewer...
```

Figure 57

You can now edit the module. Using the techniques you learned in the earlier sections, you are now able to simulate and test your design.