

Computer System Design Lab

Tutorial 1

Xilinx Schematic Editor

Xilinx ISim

1-bit and 4-bit Full Adder

Visual Walkthrough

Create New Project

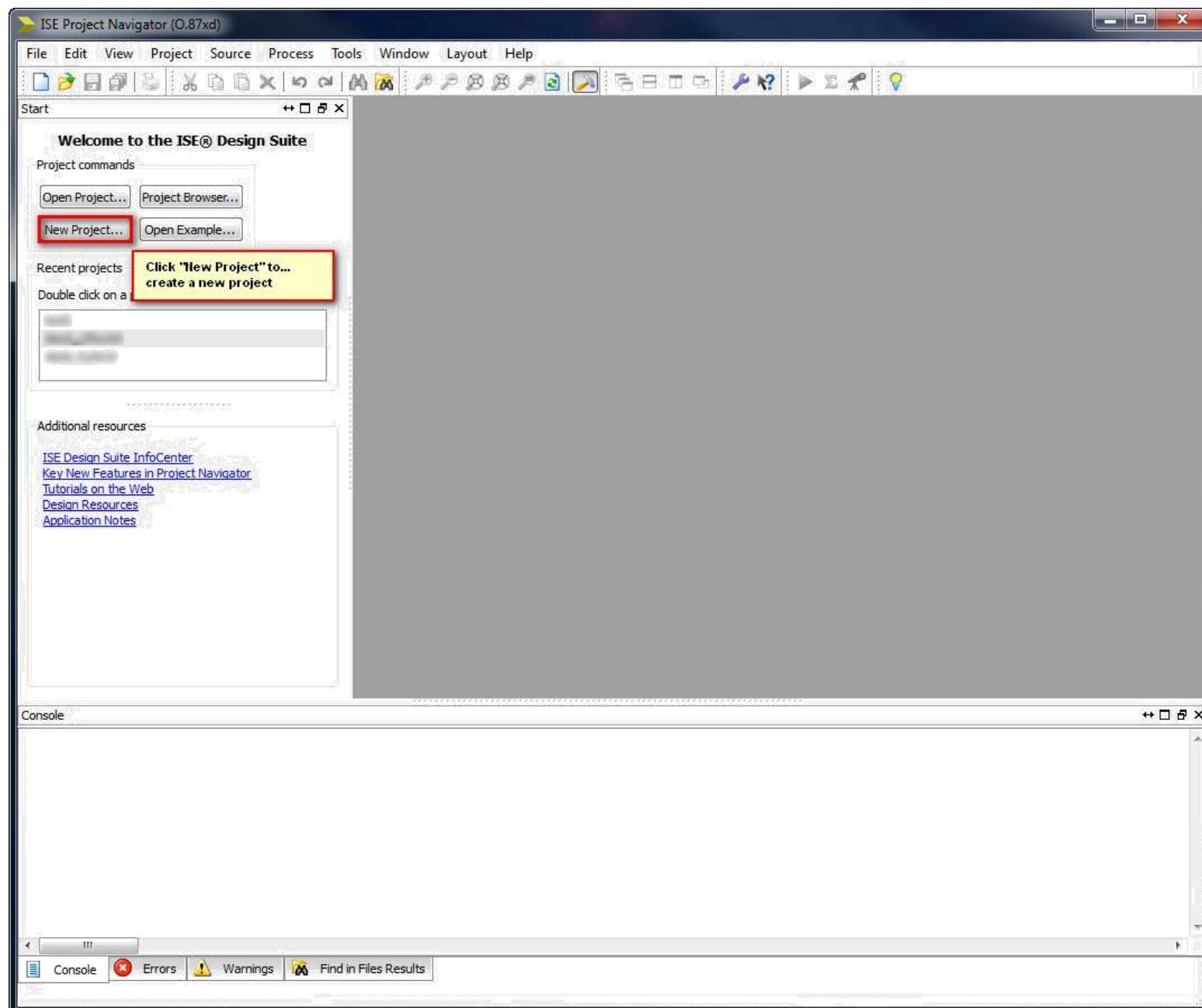
Create Schematic

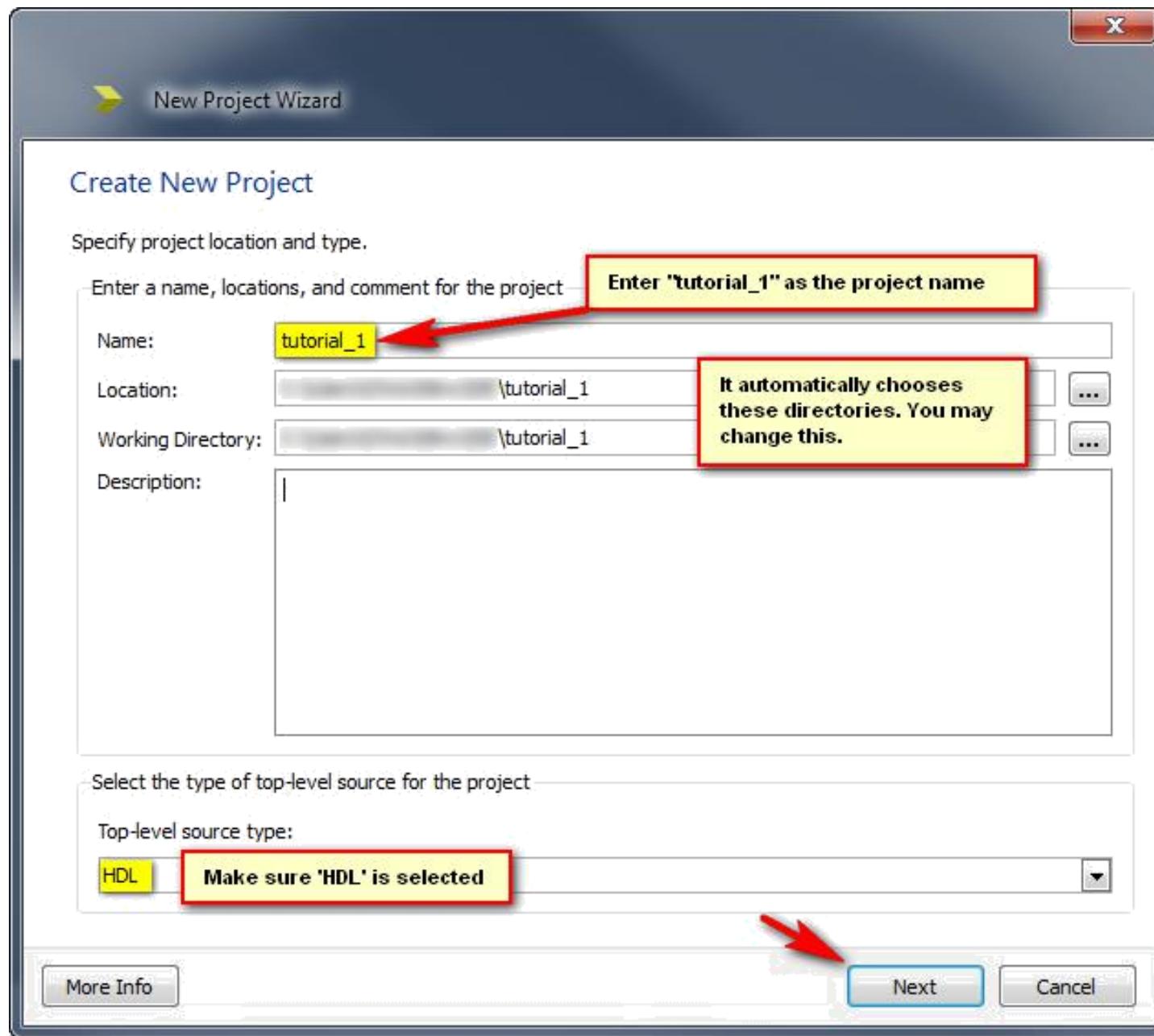
Simulate

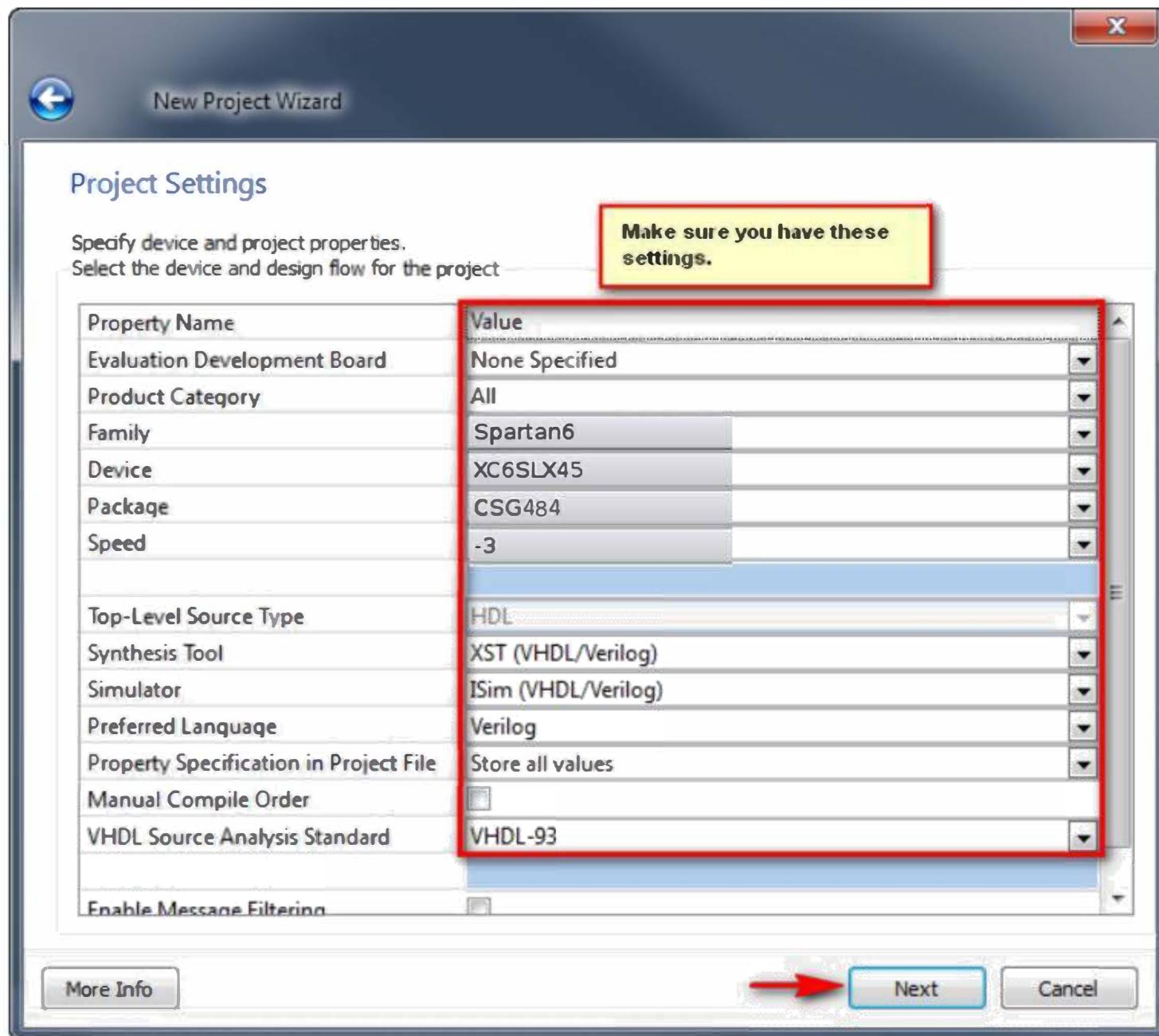
Create Schematic Symbol

Our Anvyl FPGA boards include:

Xilinx Spartan-6 LX45 FPGA (XC6SLX45-CSG484-3)









New Project Wizard

Project Summary

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

Project:

Project Name: tutorial_1

Project Path: \Xilinx\ISE\tutorial_1

Working Directory: \Xilinx\ISE\tutorial_1

Description:

Top Level Source Type: HDL

Check to make sure
everything looks right. (Or
just click 'Finish').

Device:

Device Family: Spartan6

Device: xc6slx45

Package: csg484

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

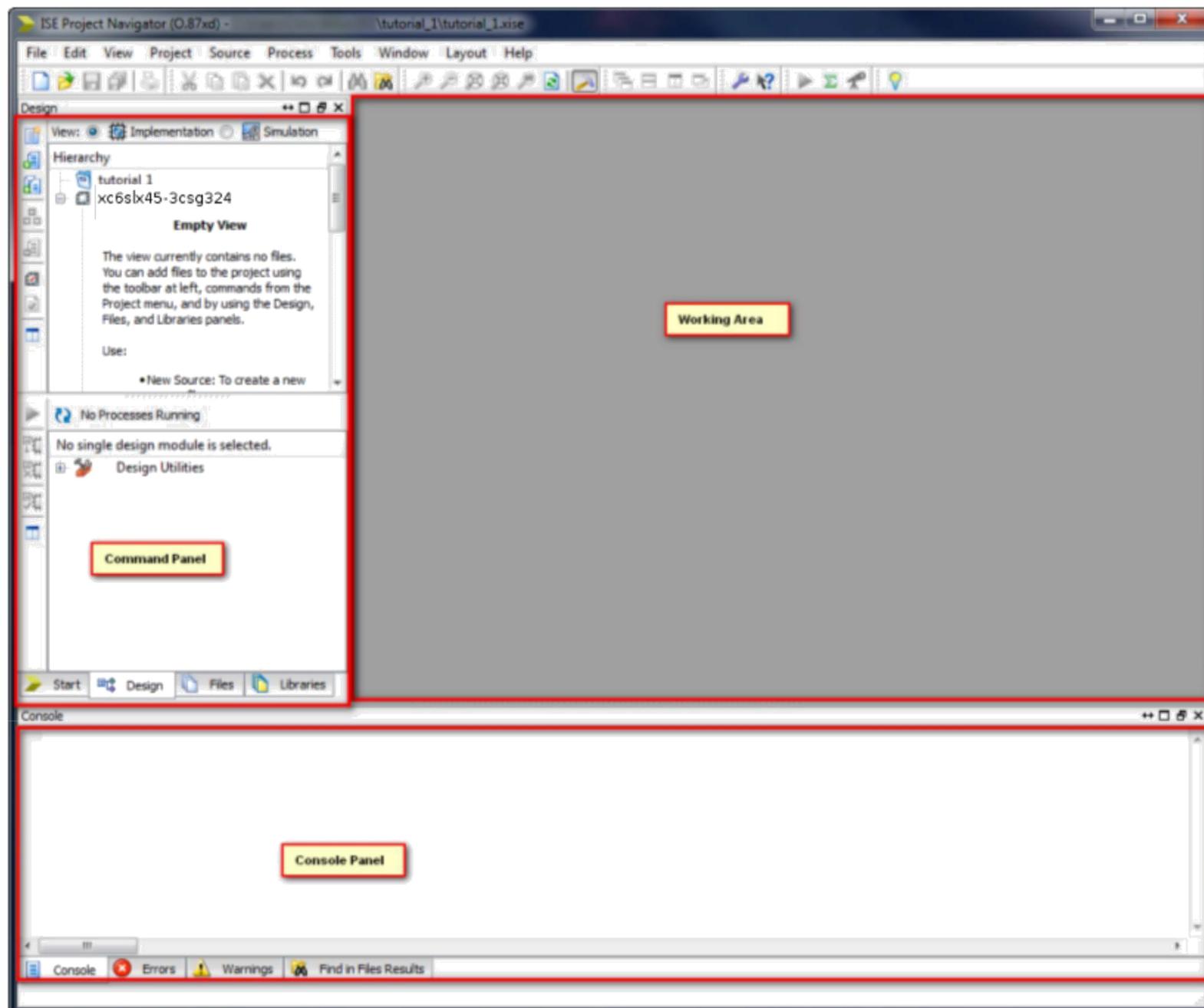
VHDL Source Analysis Standard: VHDL-93

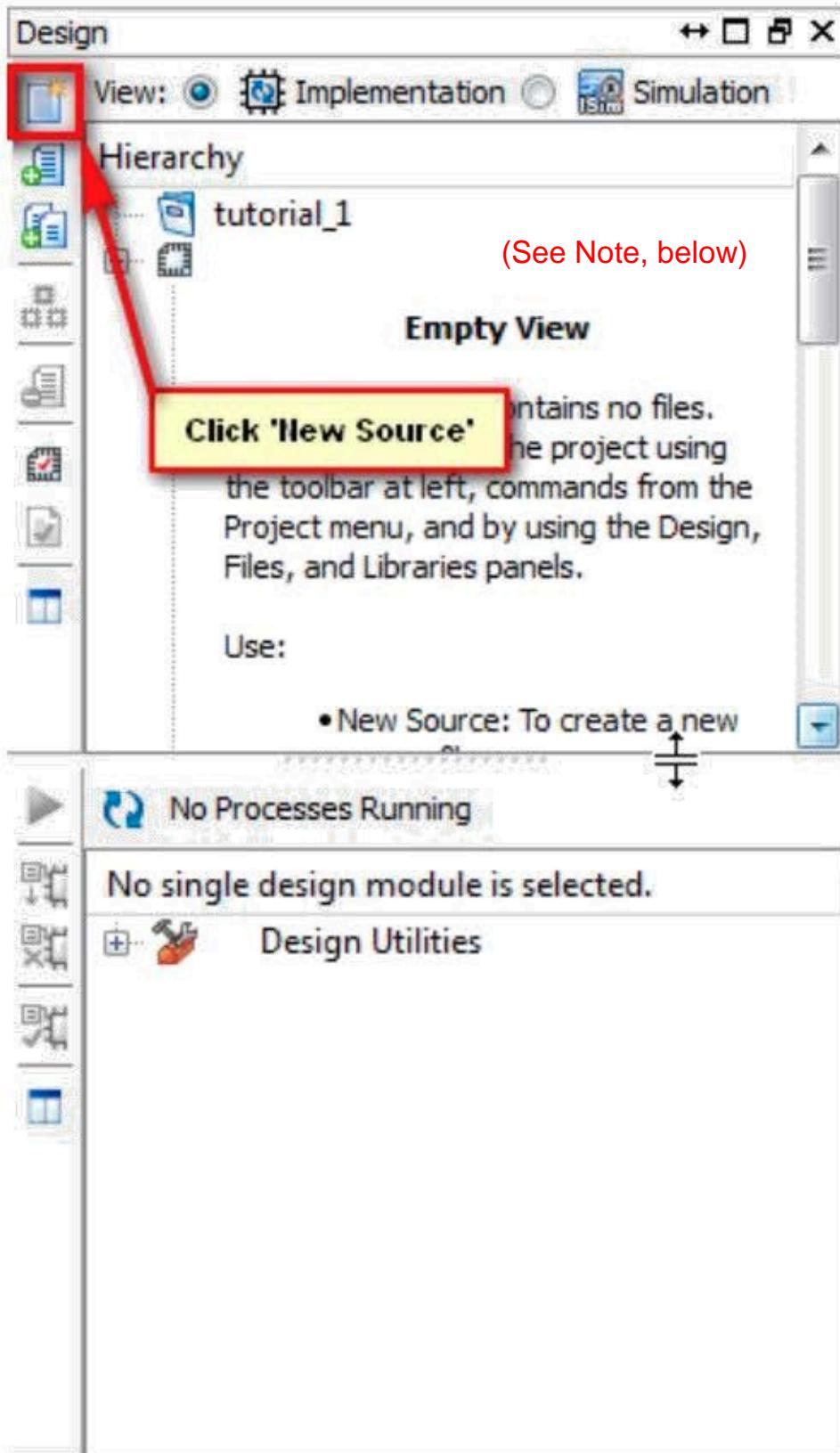
Message Filtering: disabled

More Info

Finish

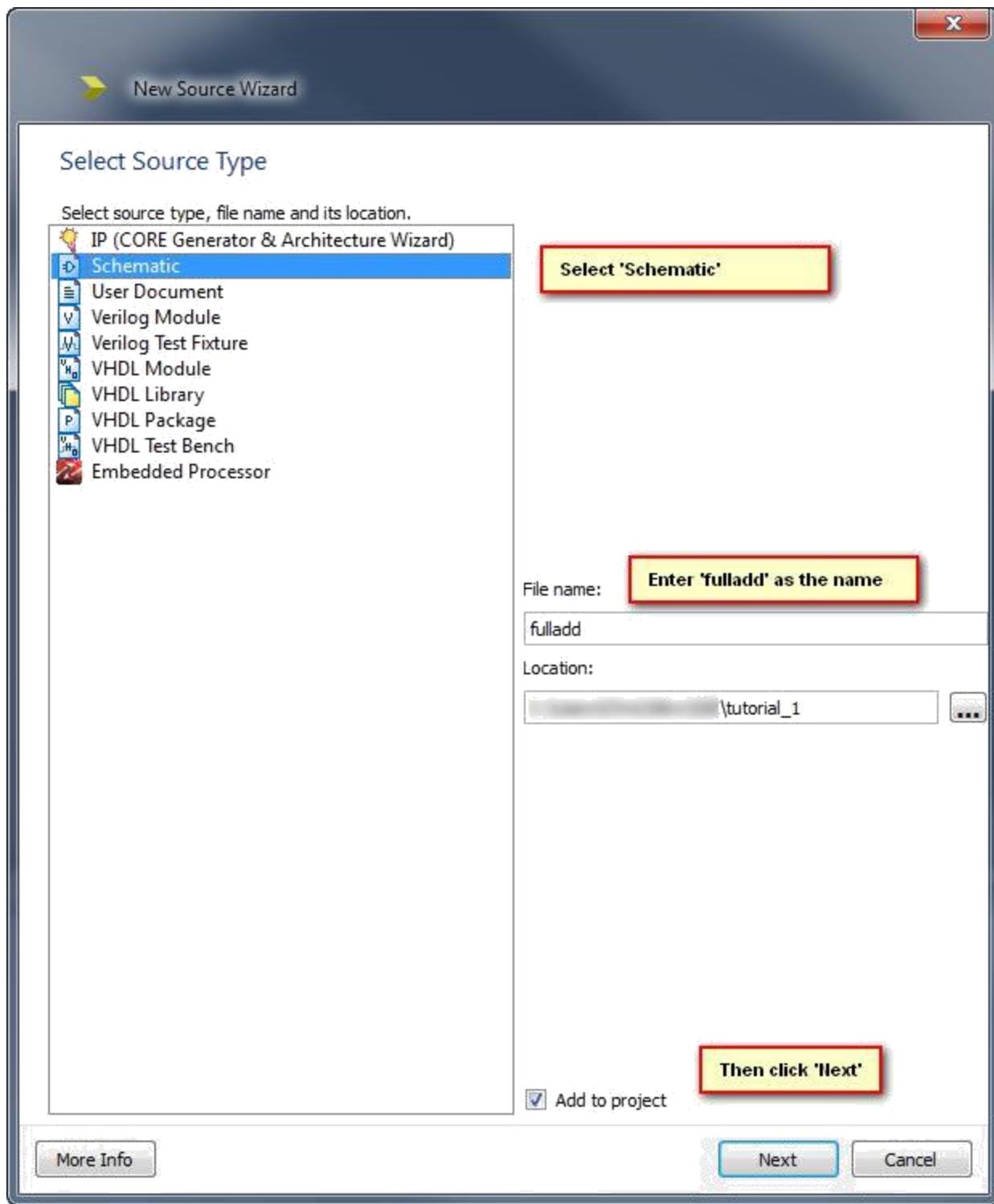
Cancel

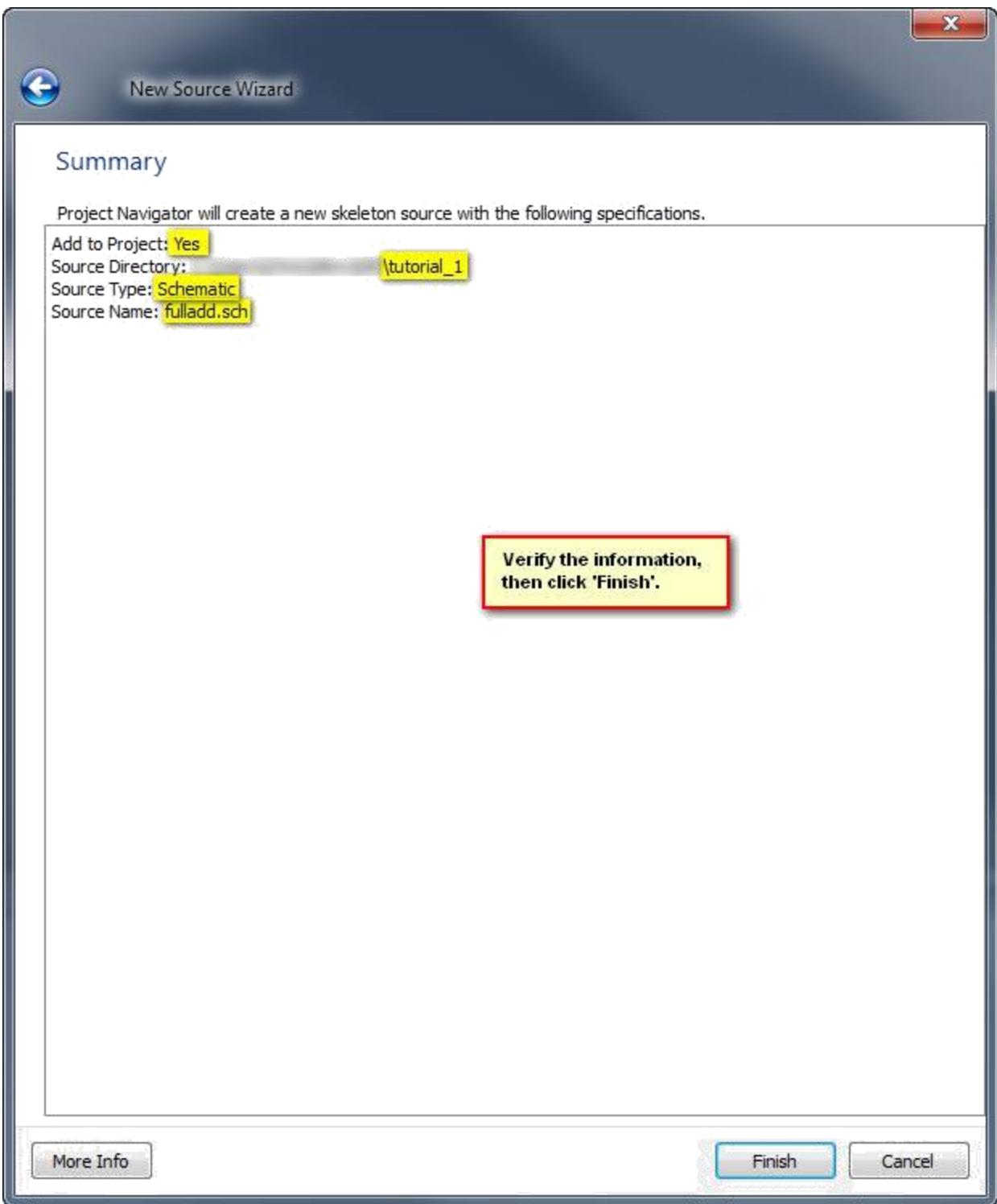


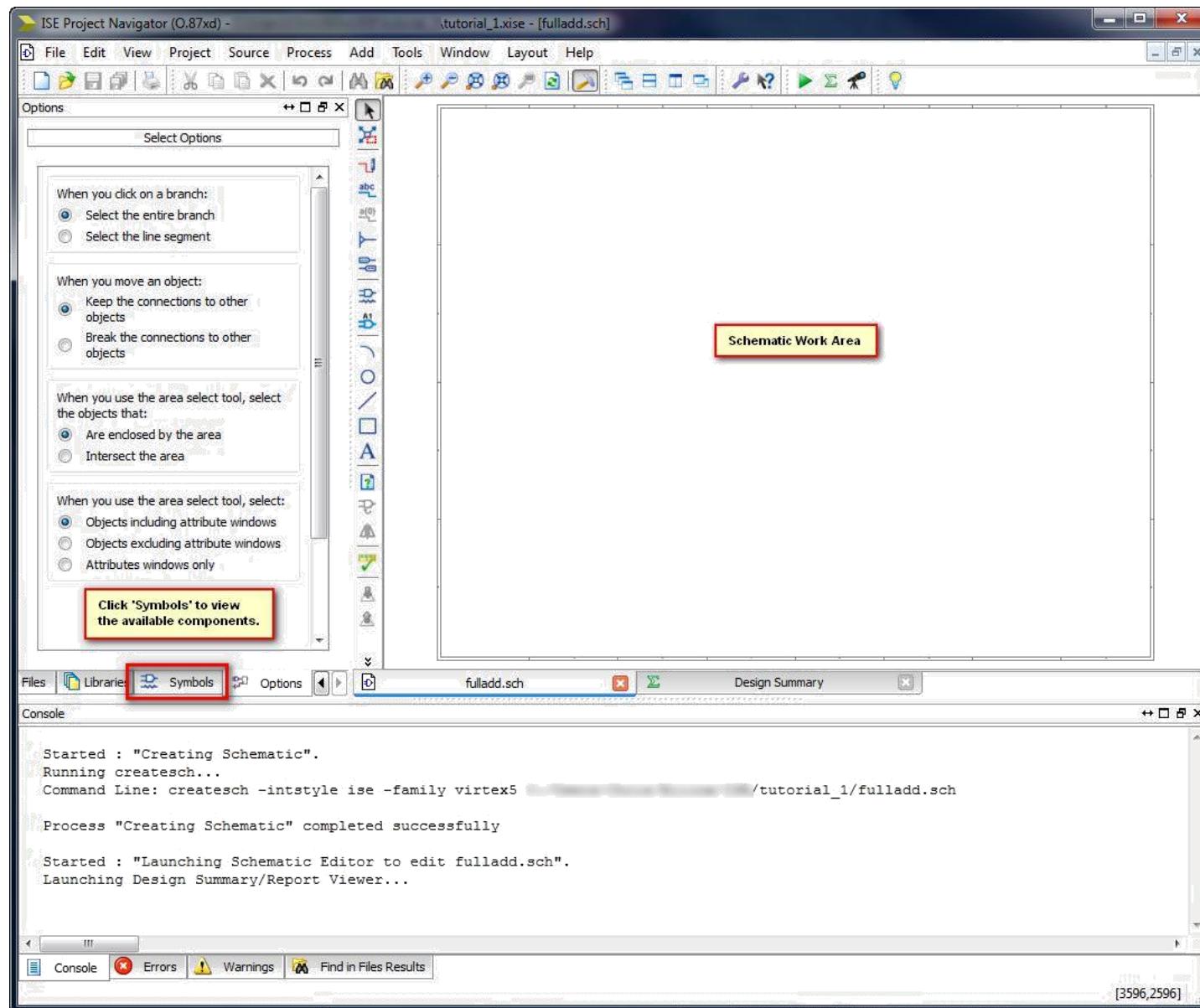


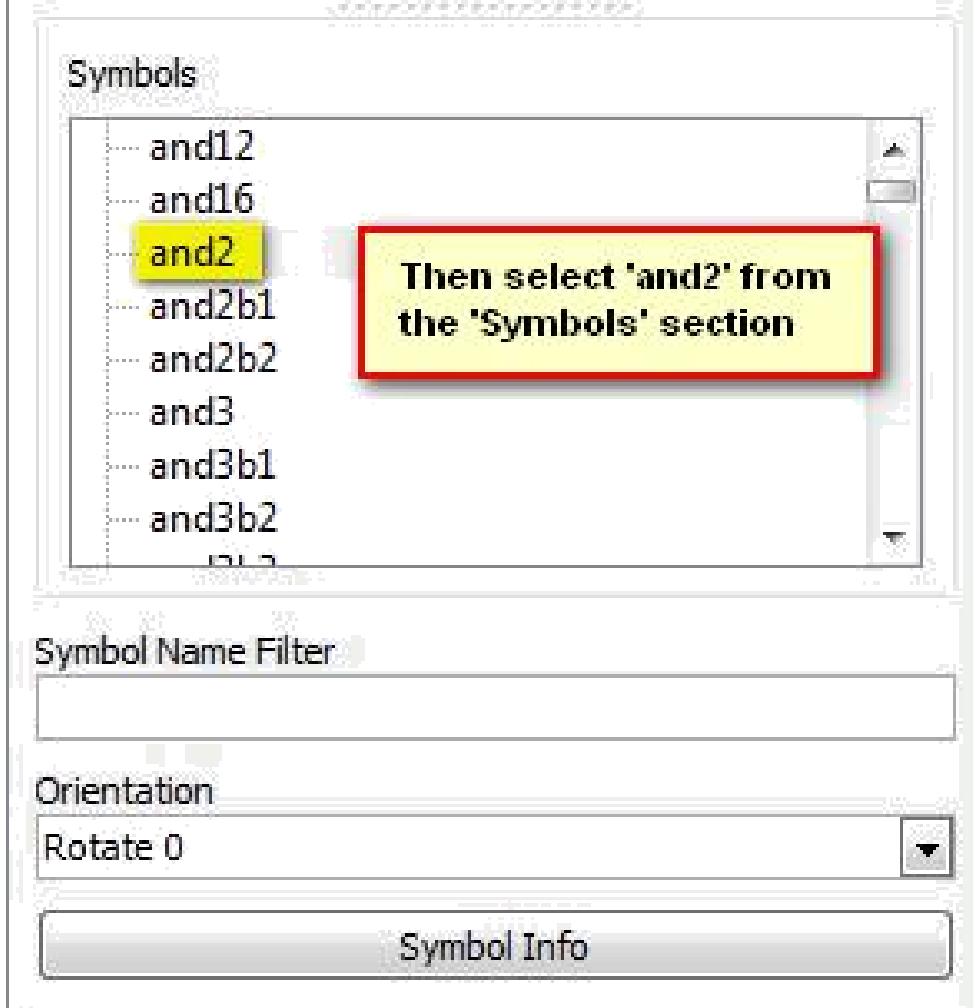
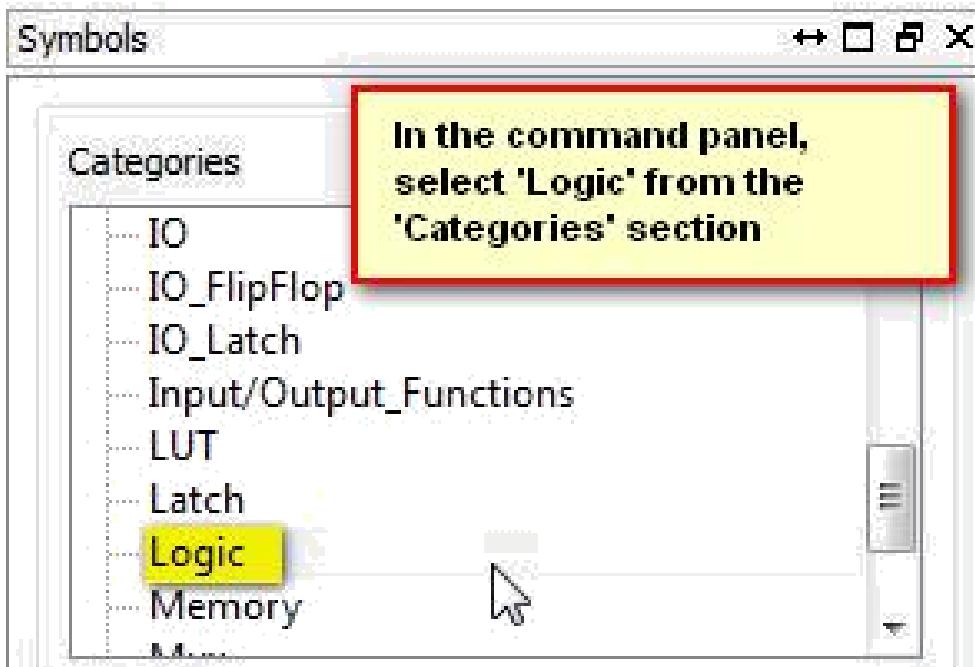
Our Anvil FPGA boards include:

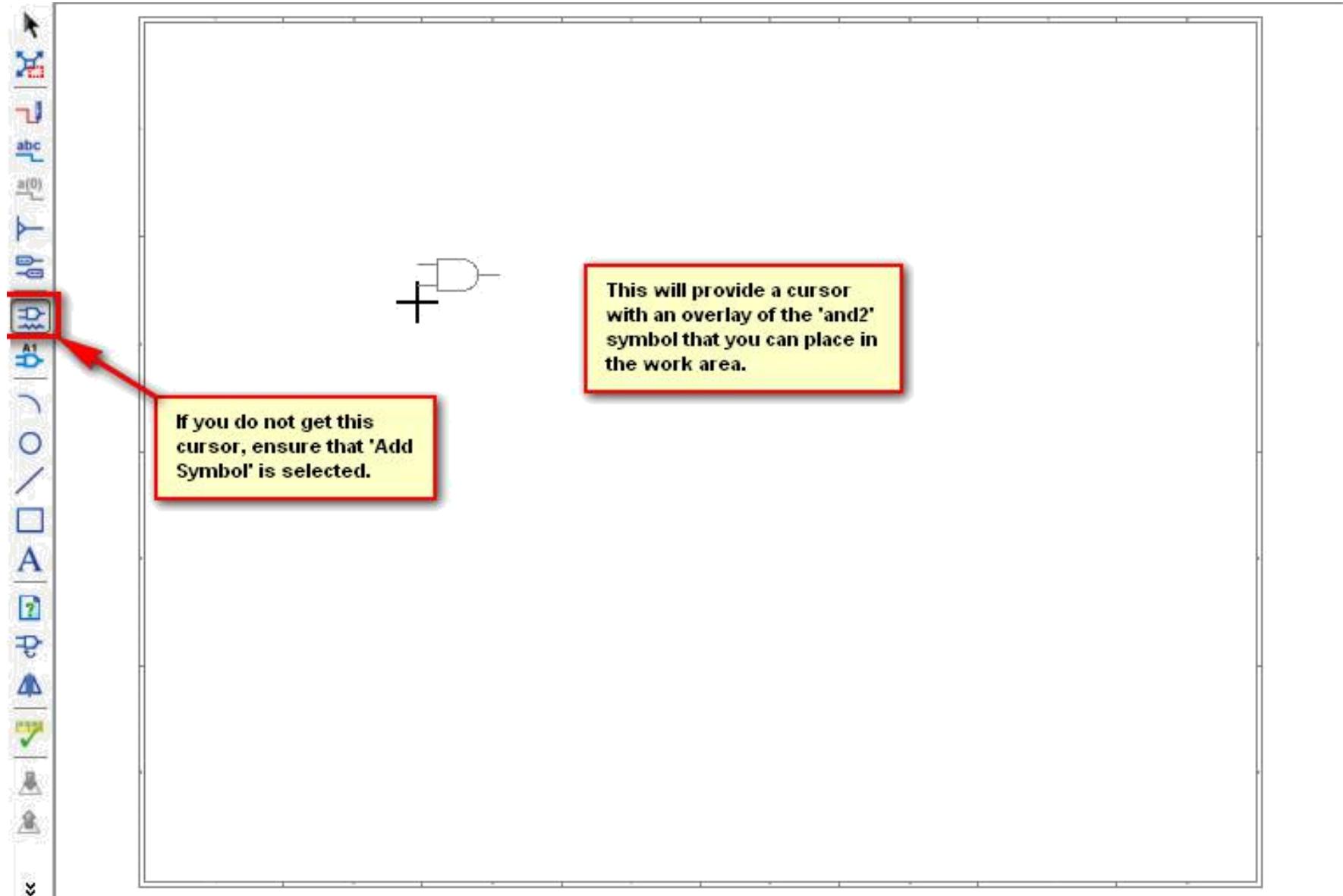
Xilinx Spartan-6 LX45 FPGA (XC6SLX45-CSG484-3)

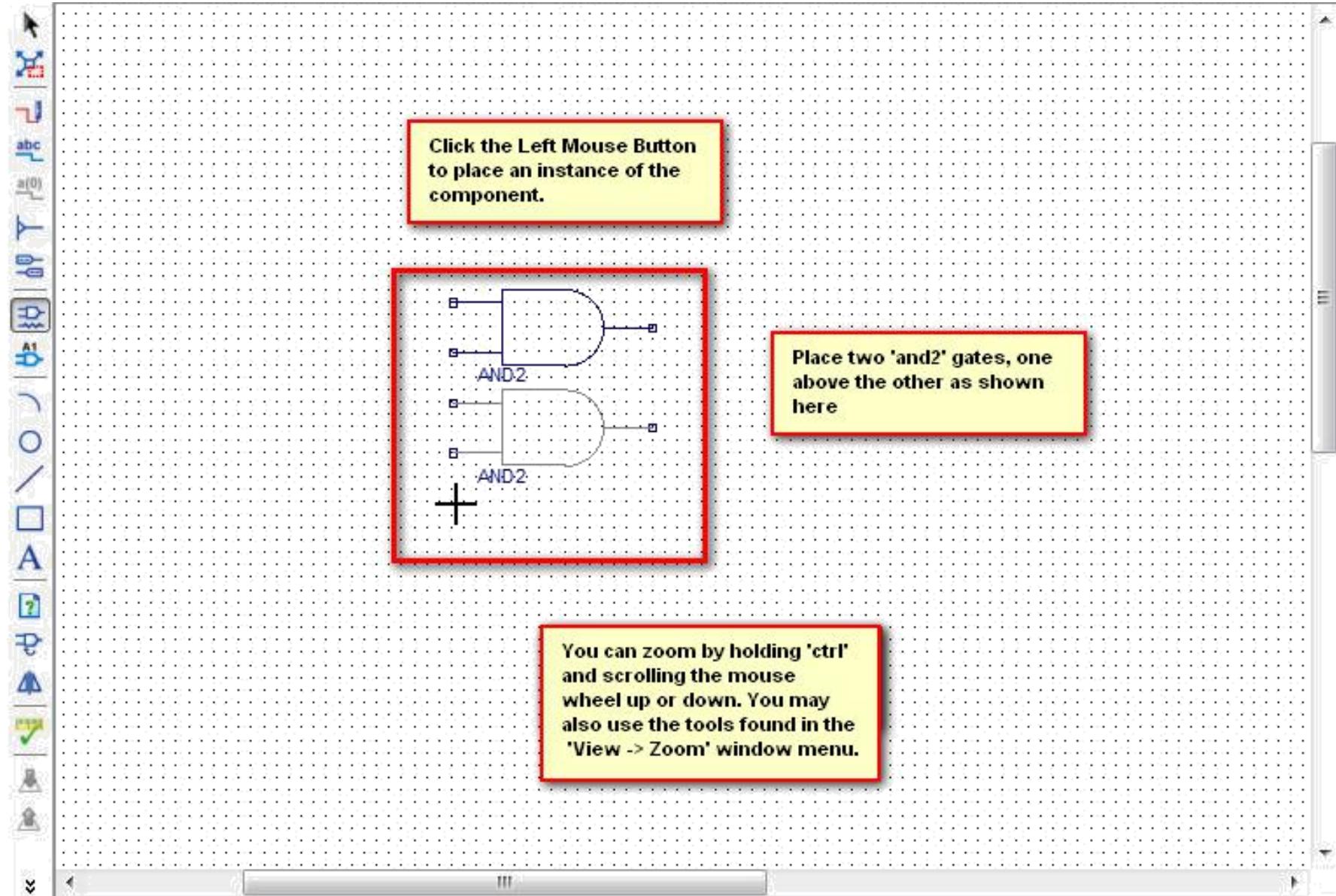


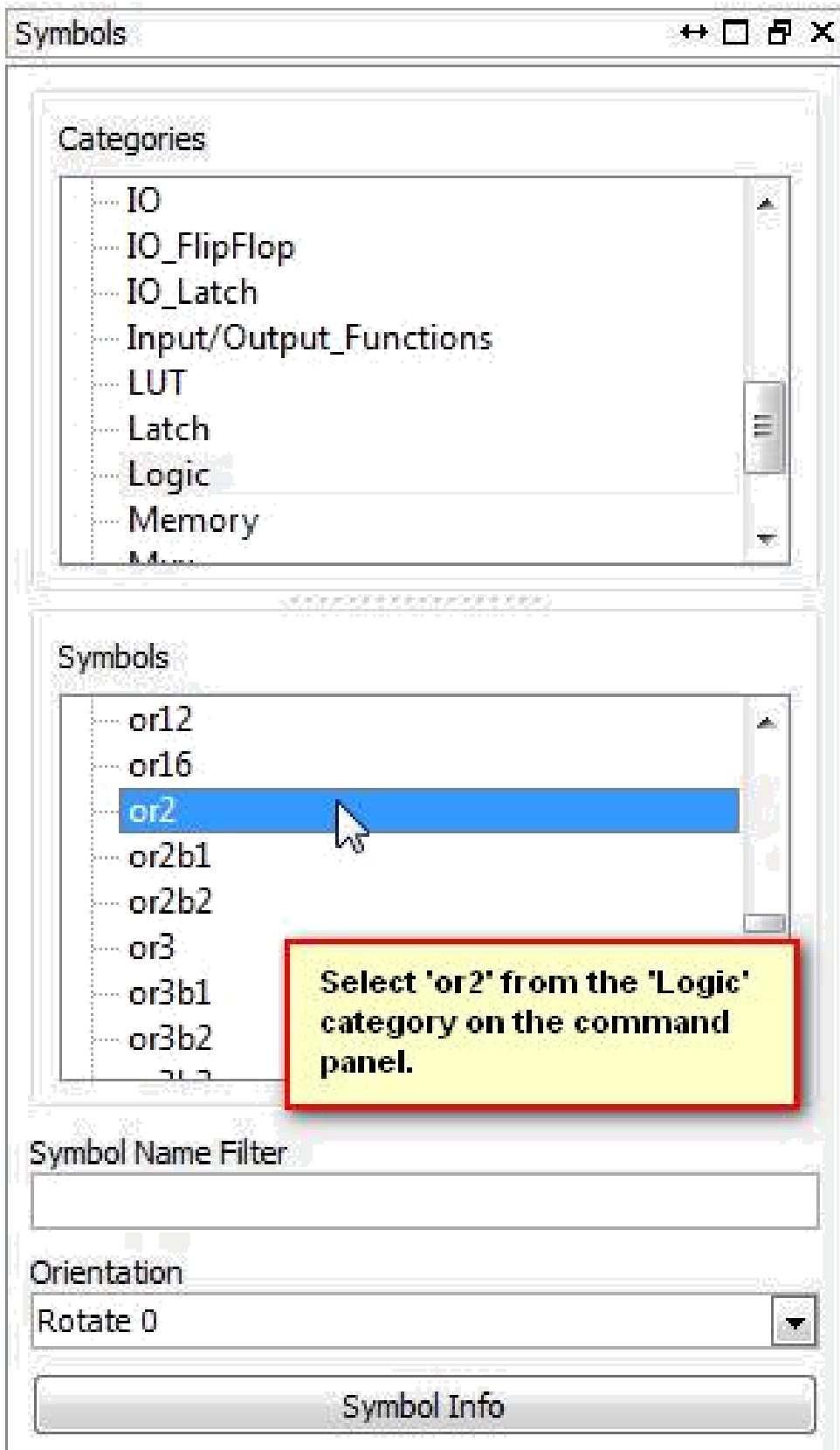


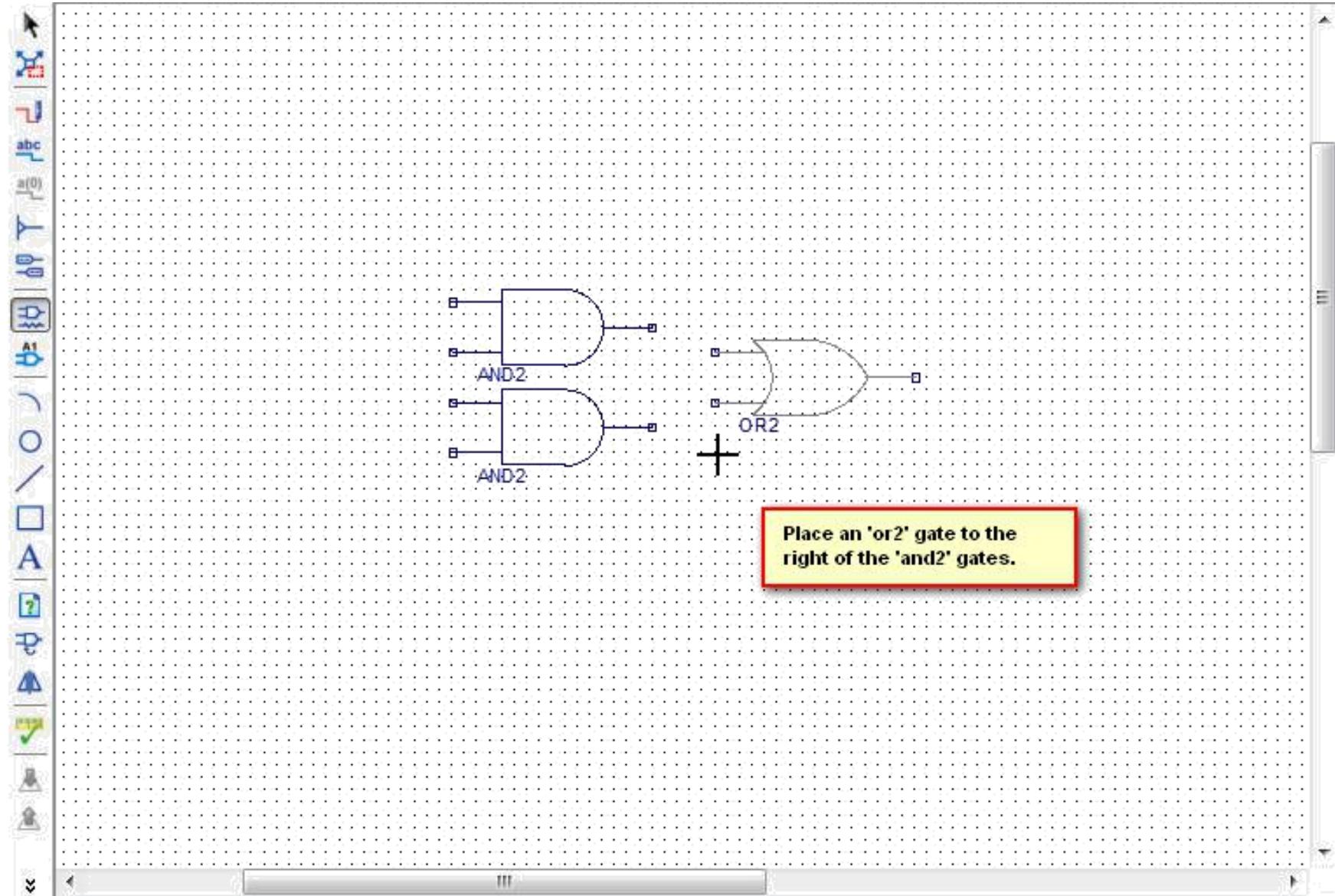


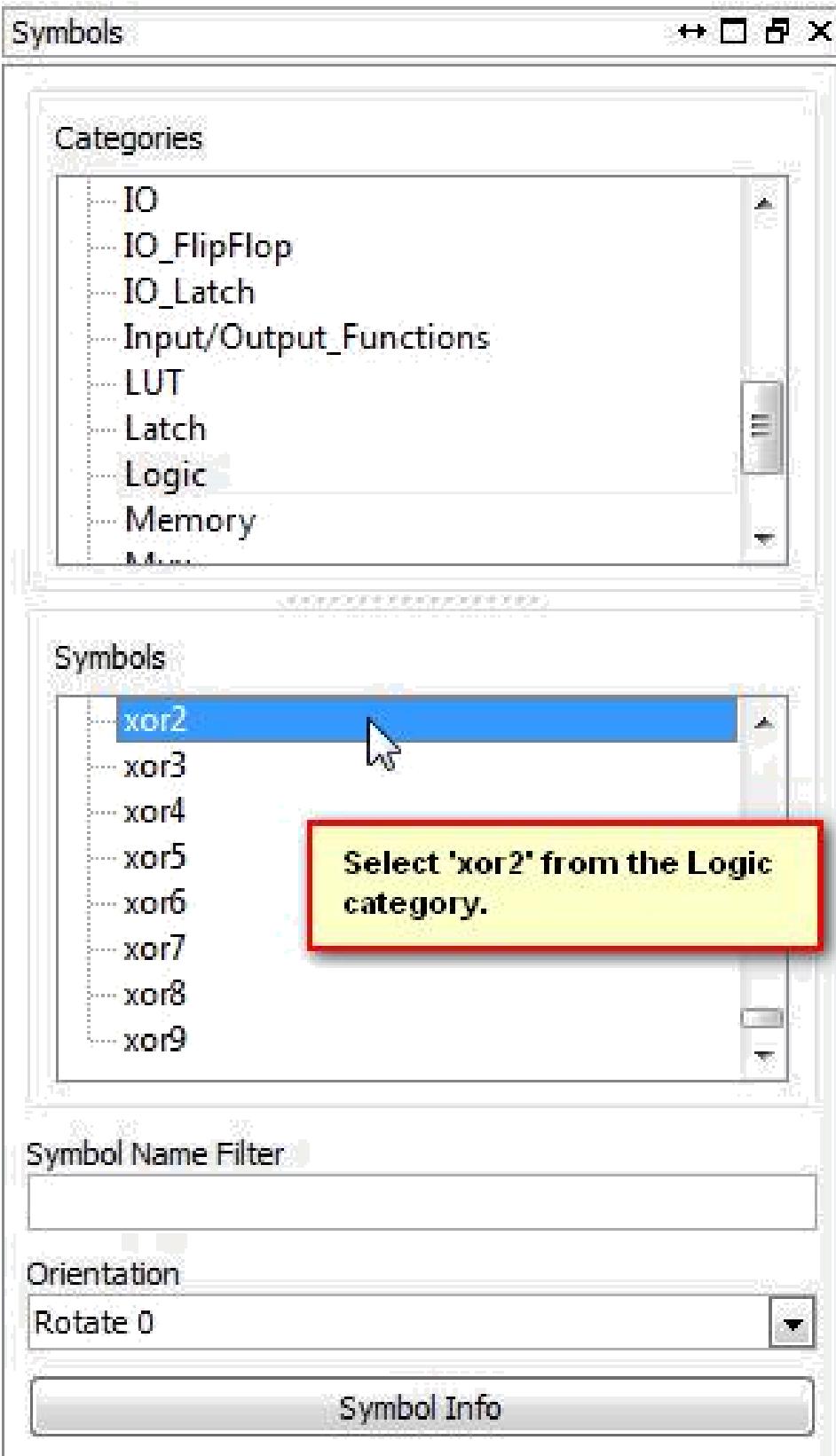


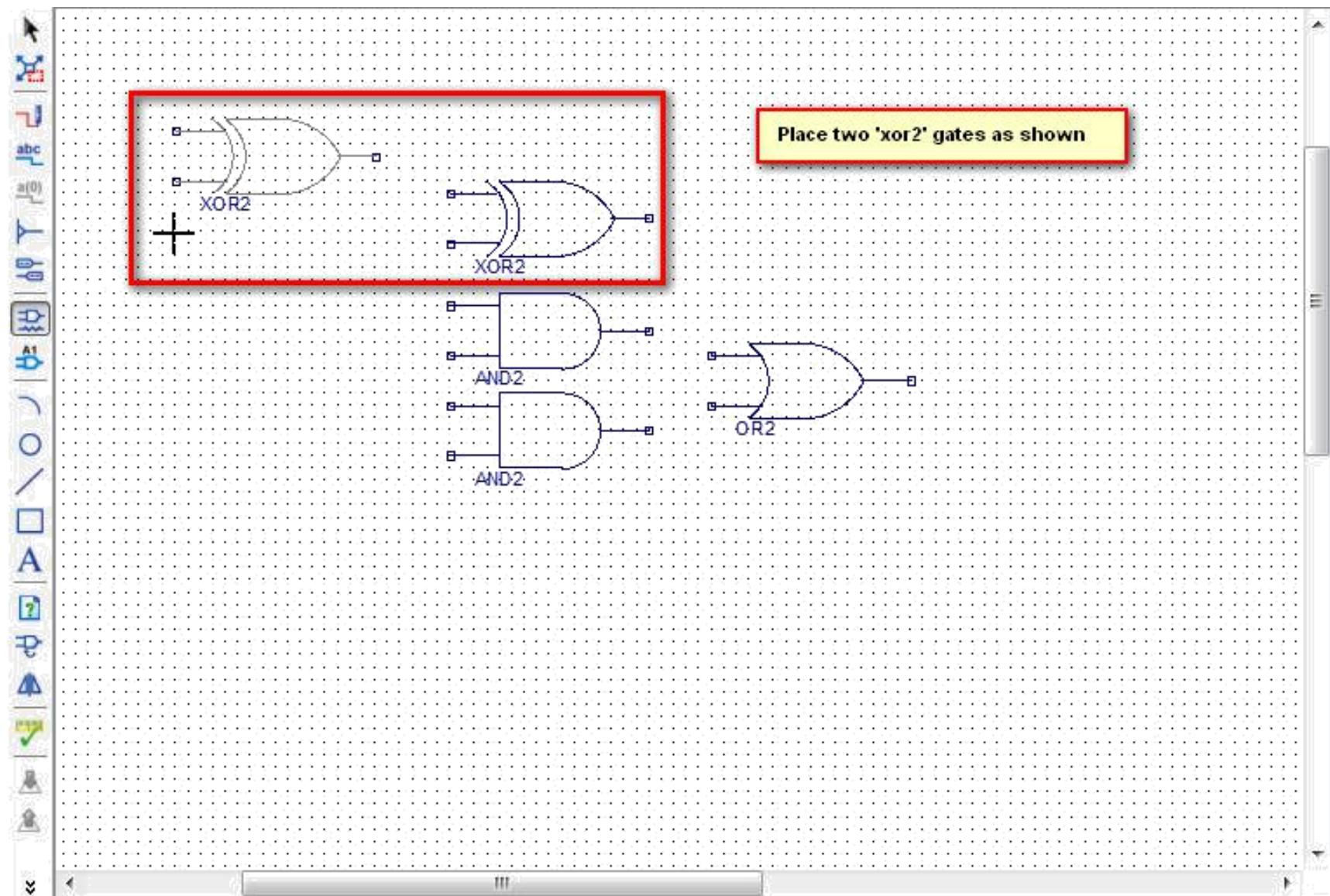


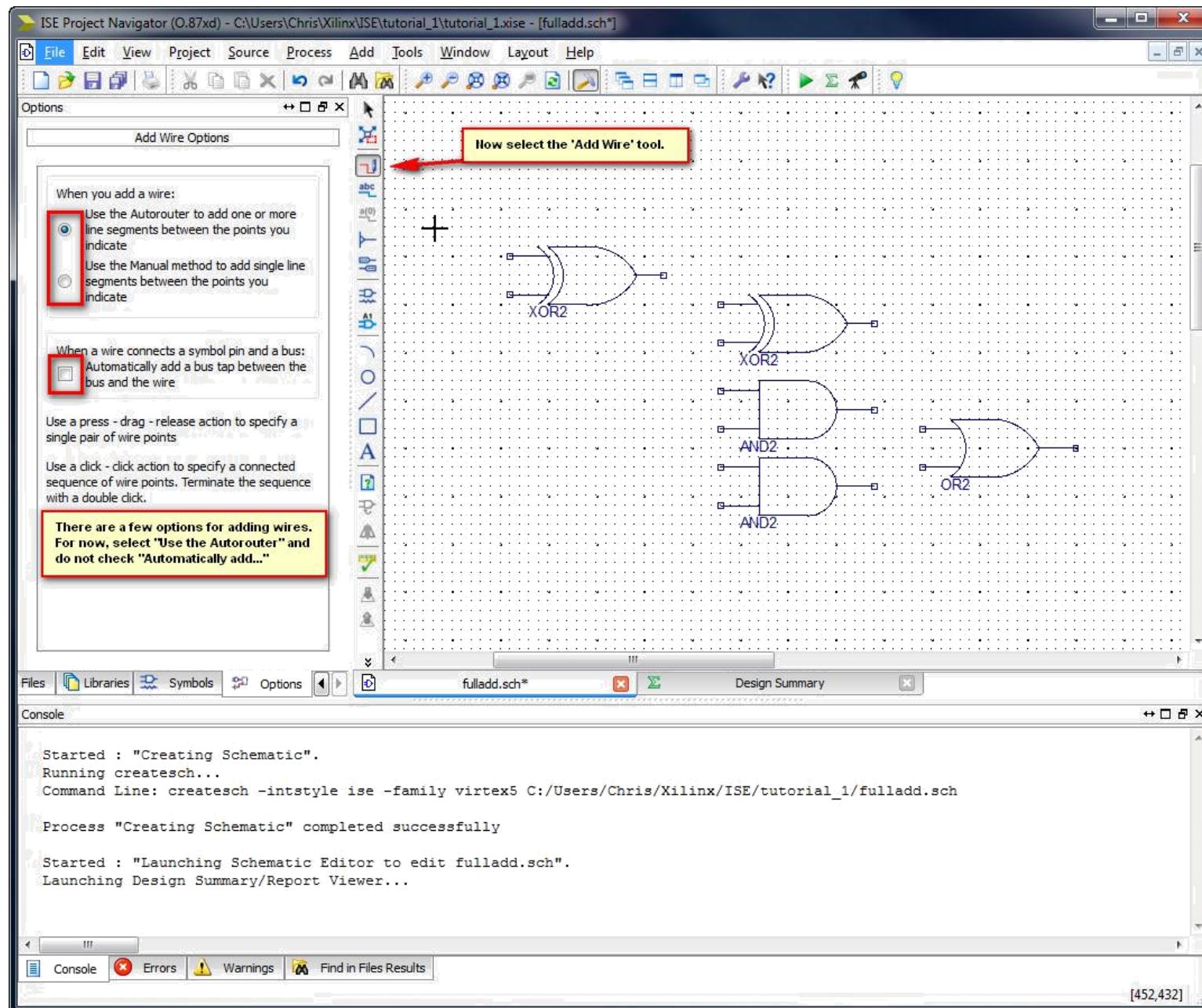


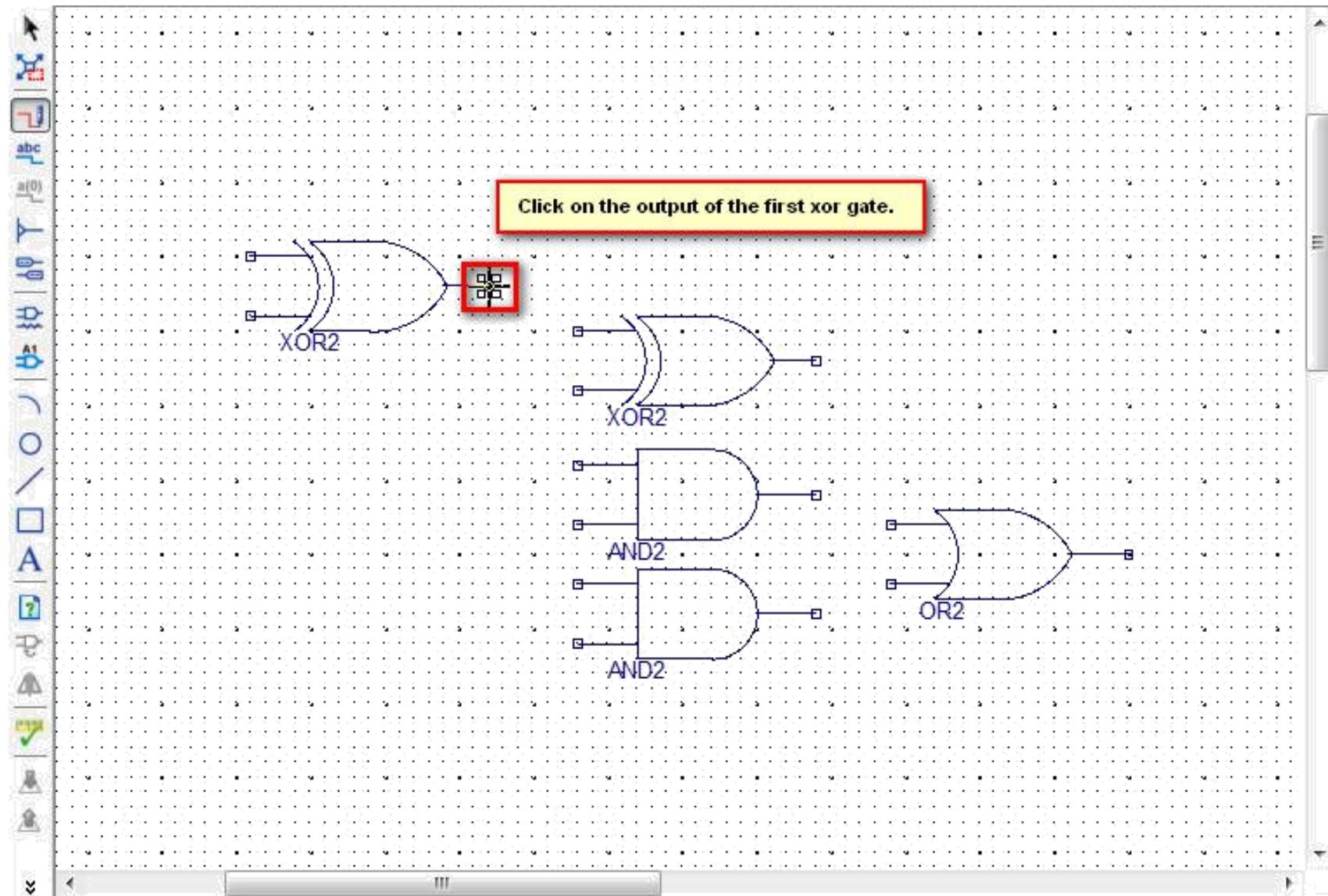


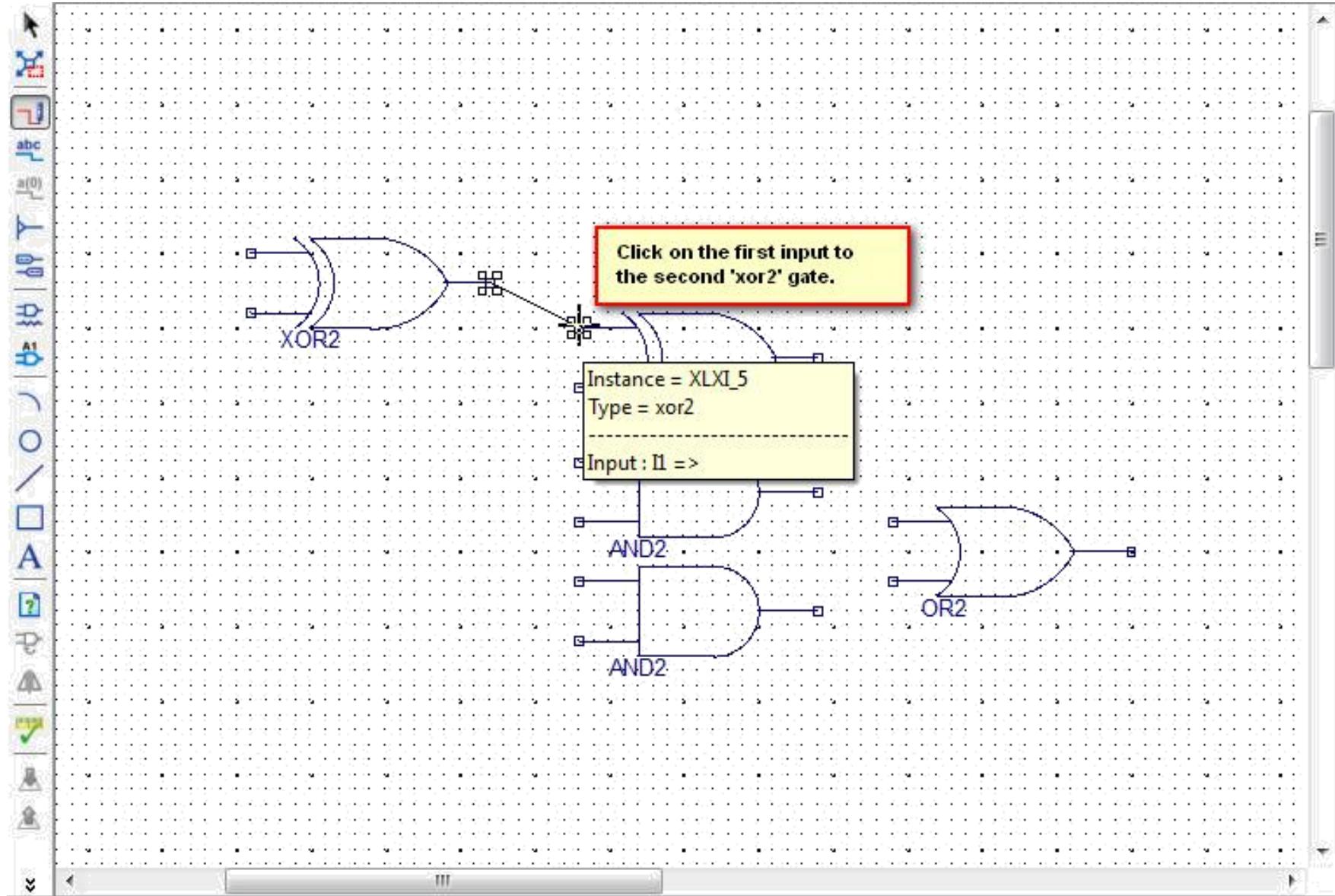


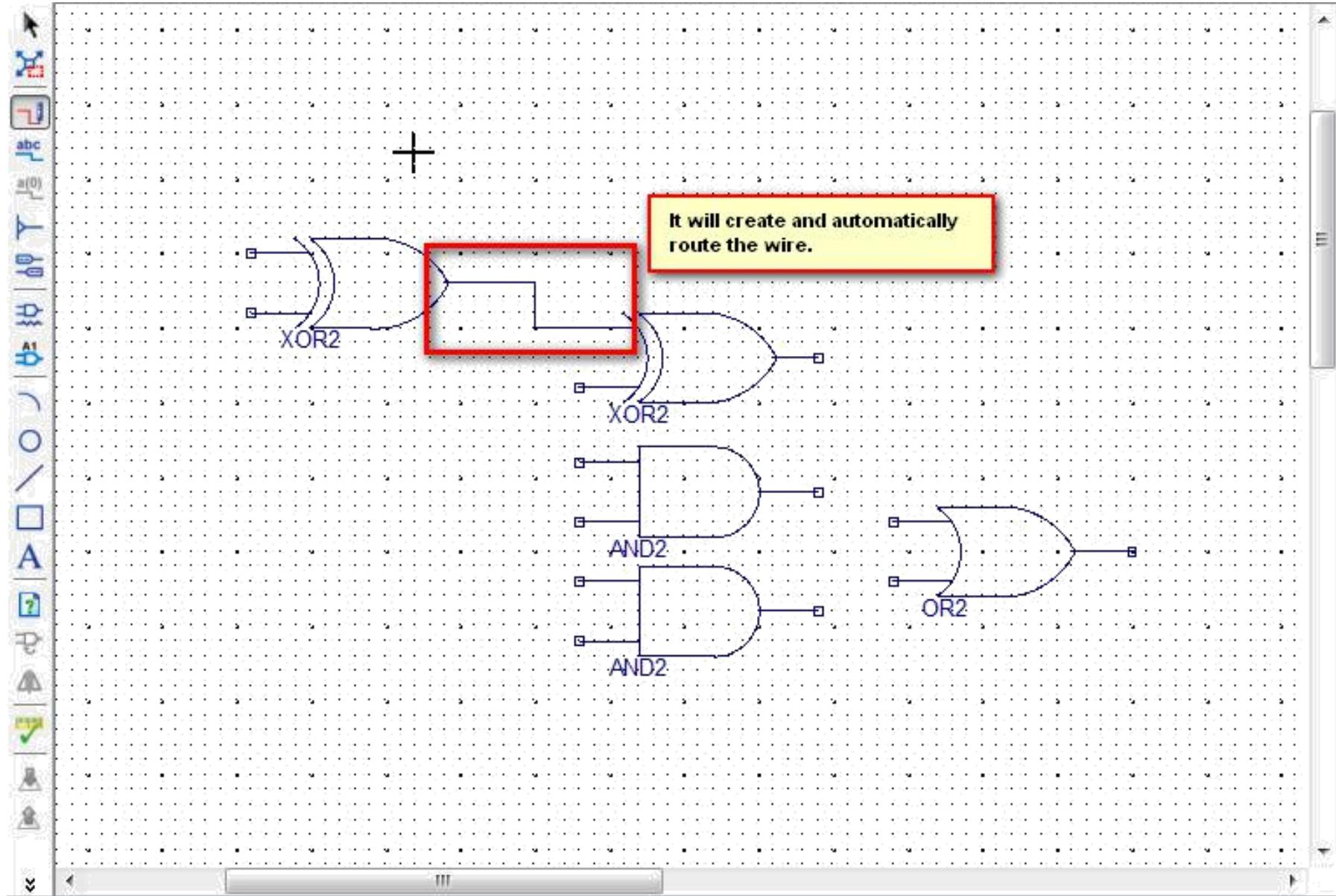


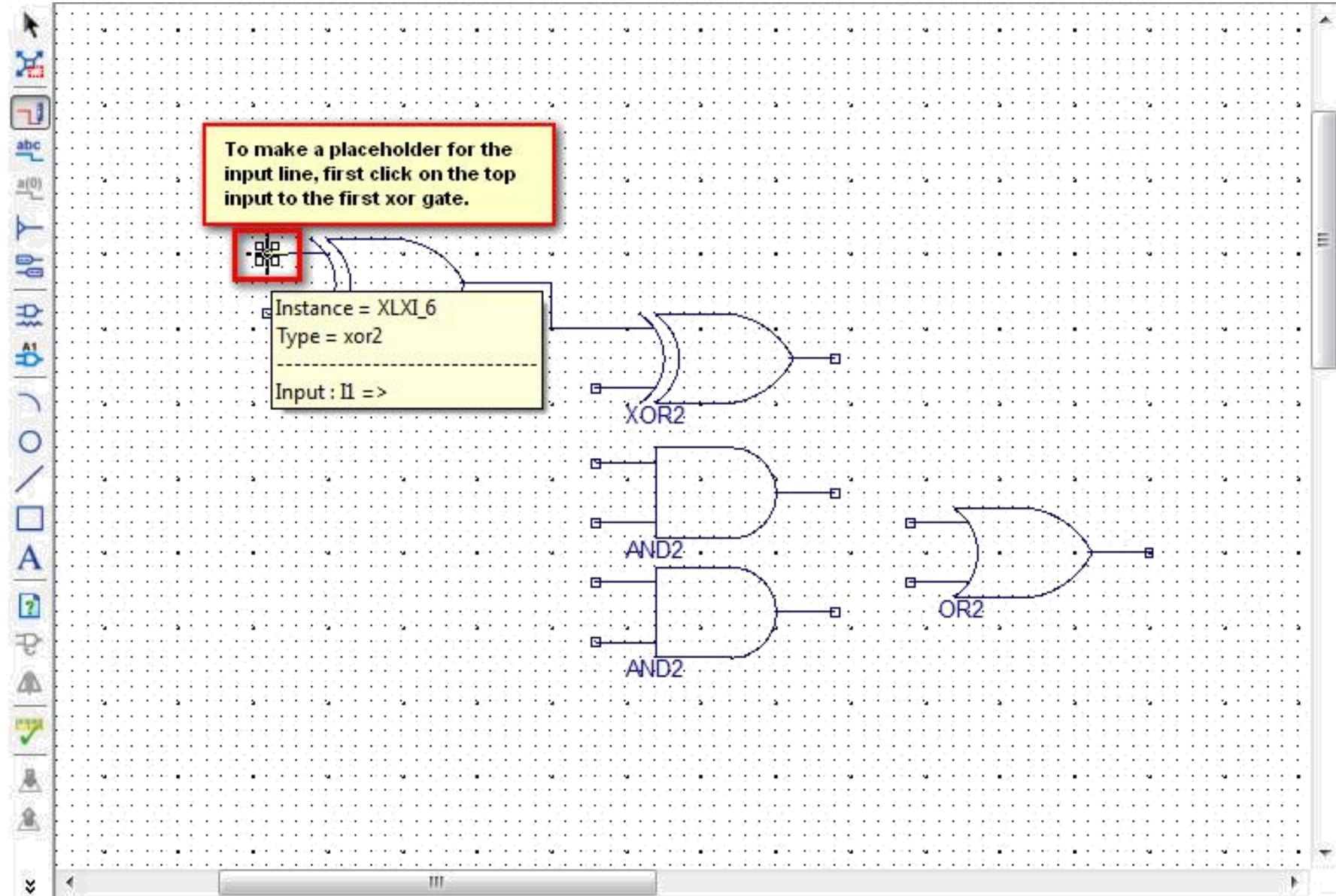


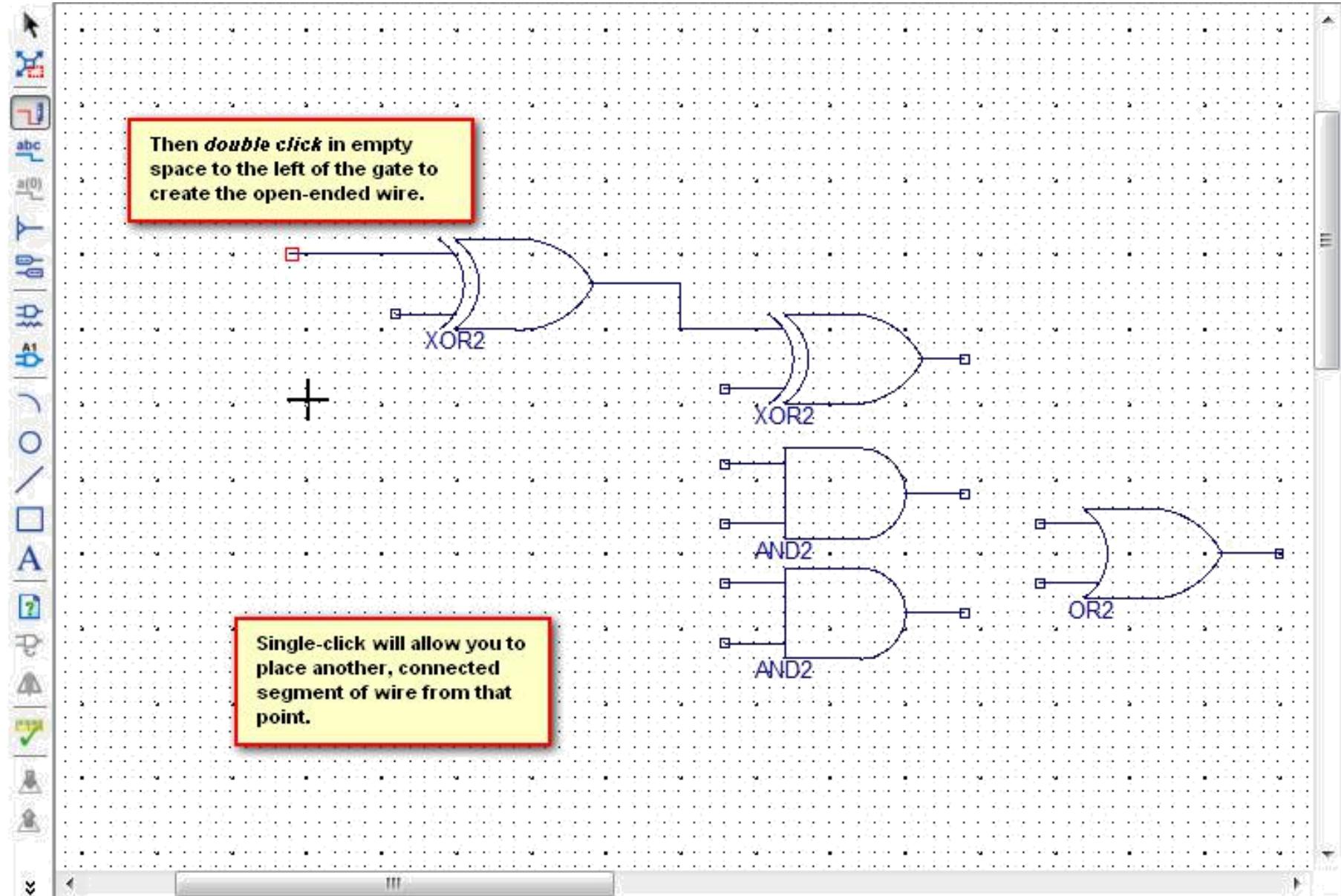


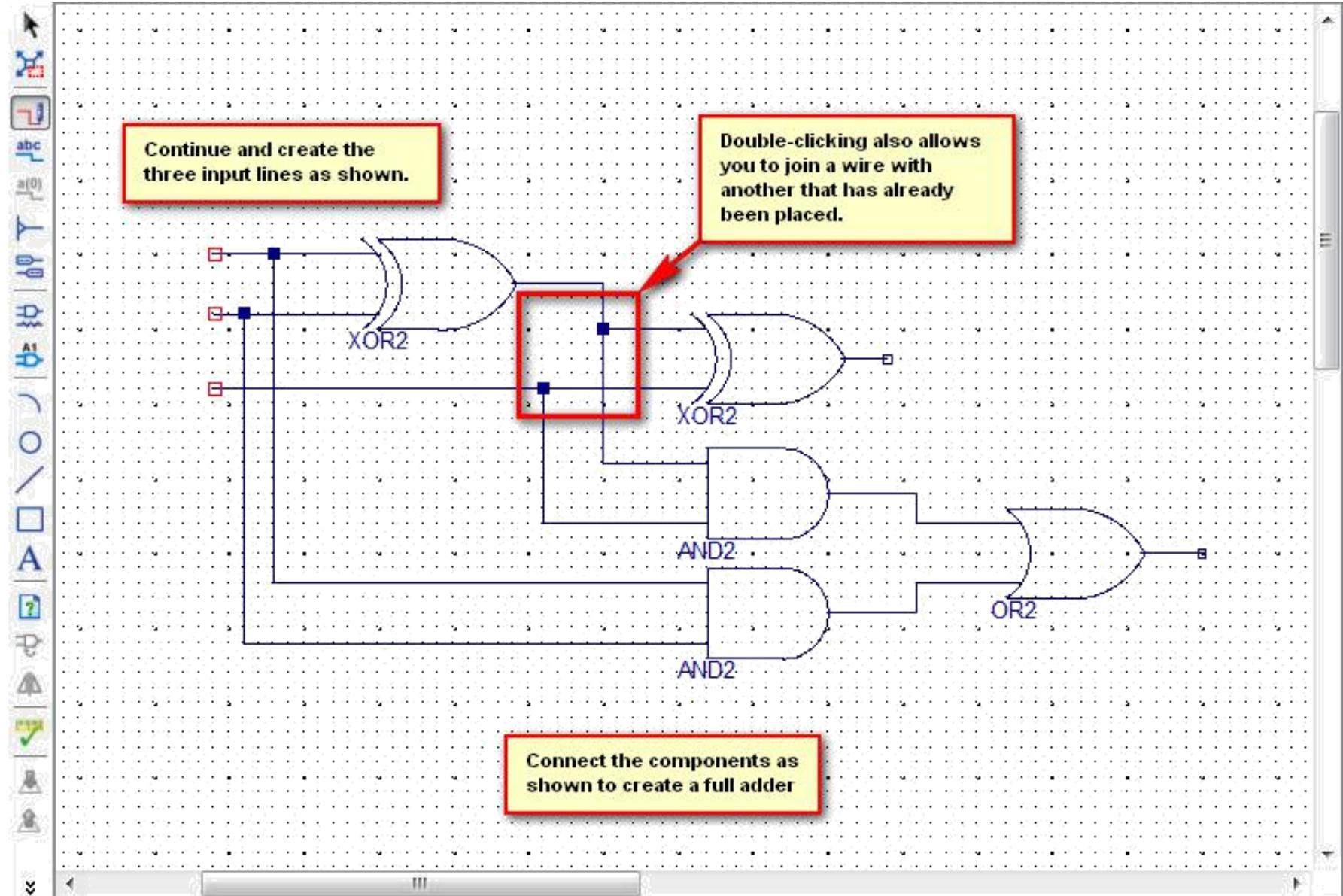


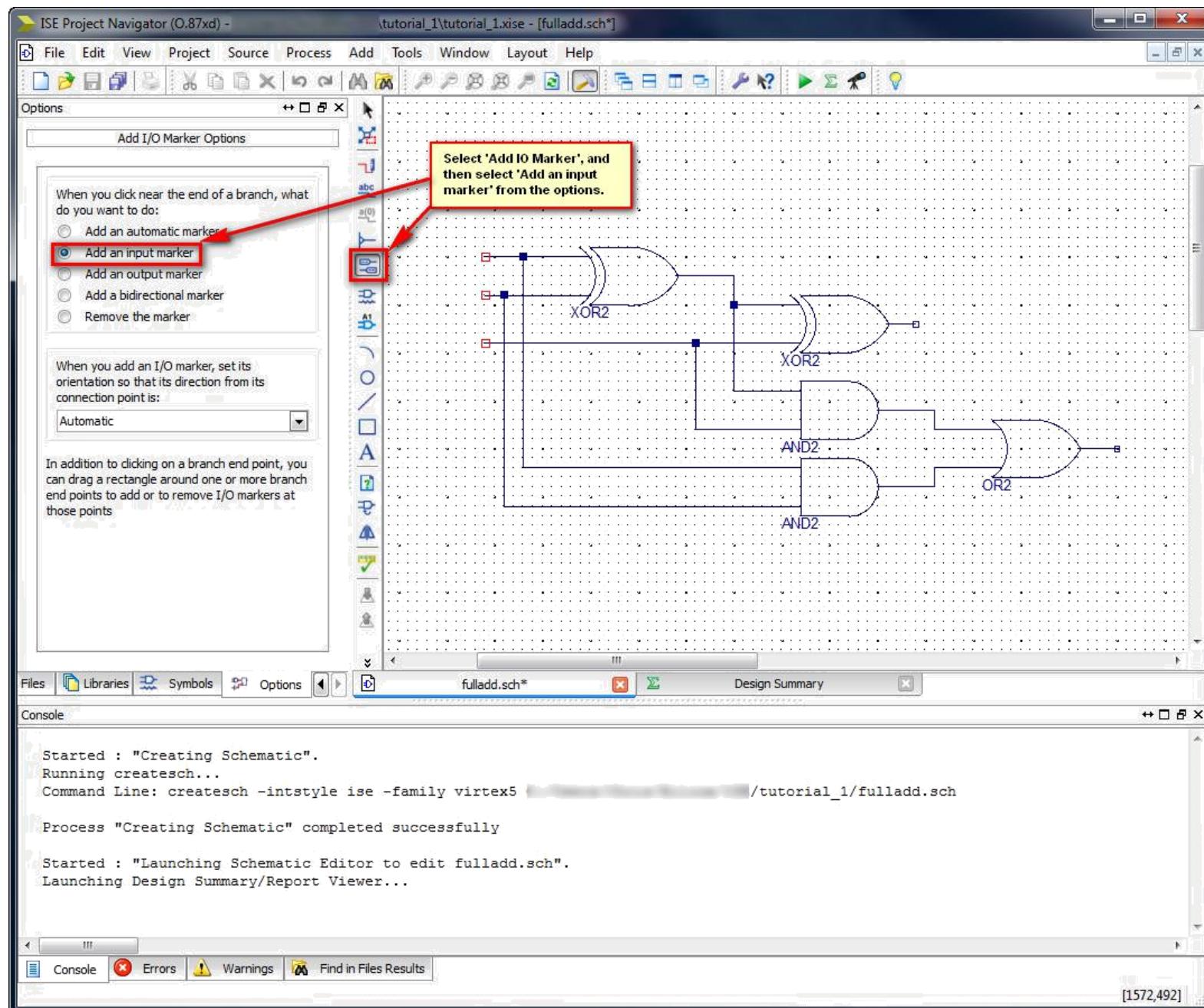


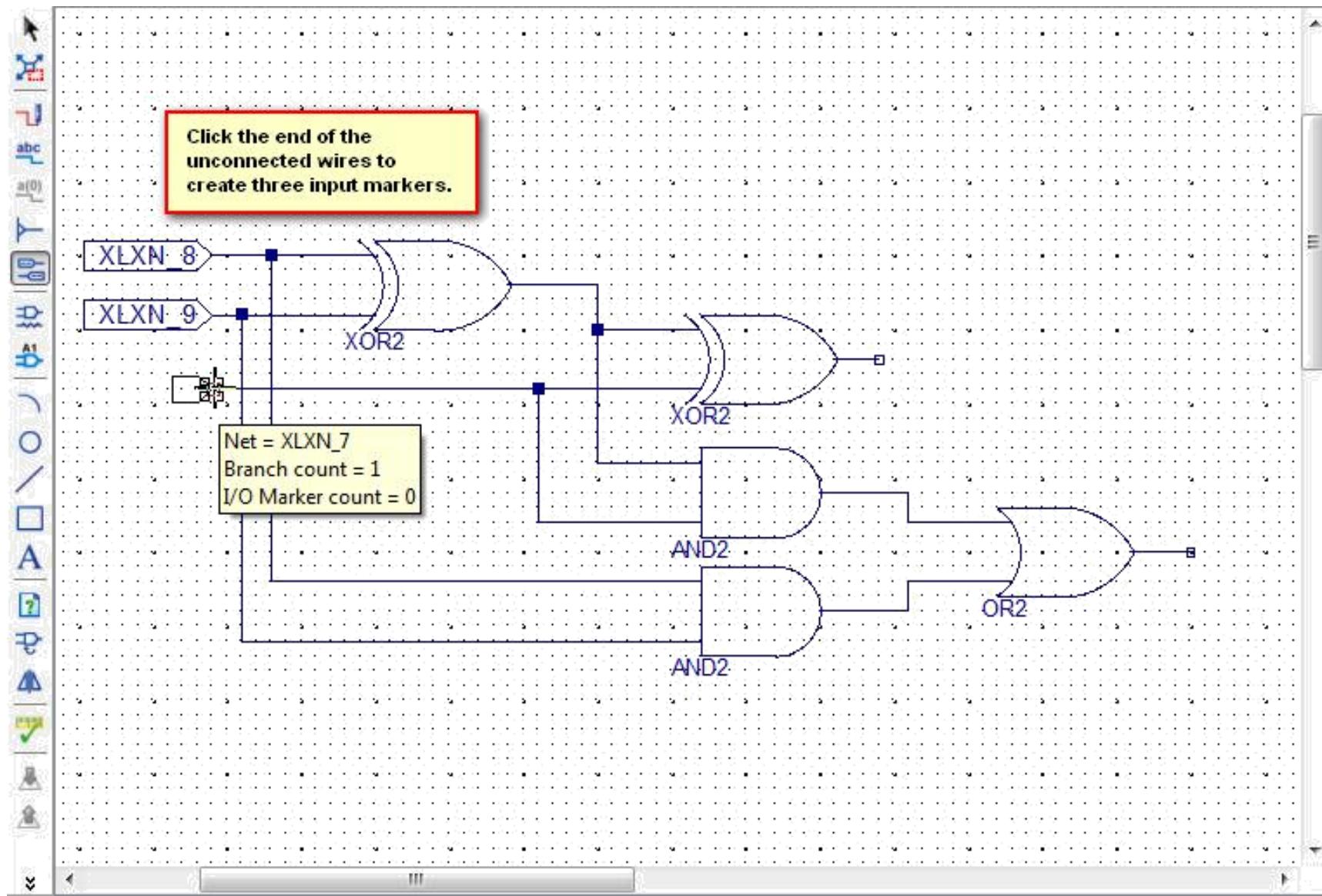


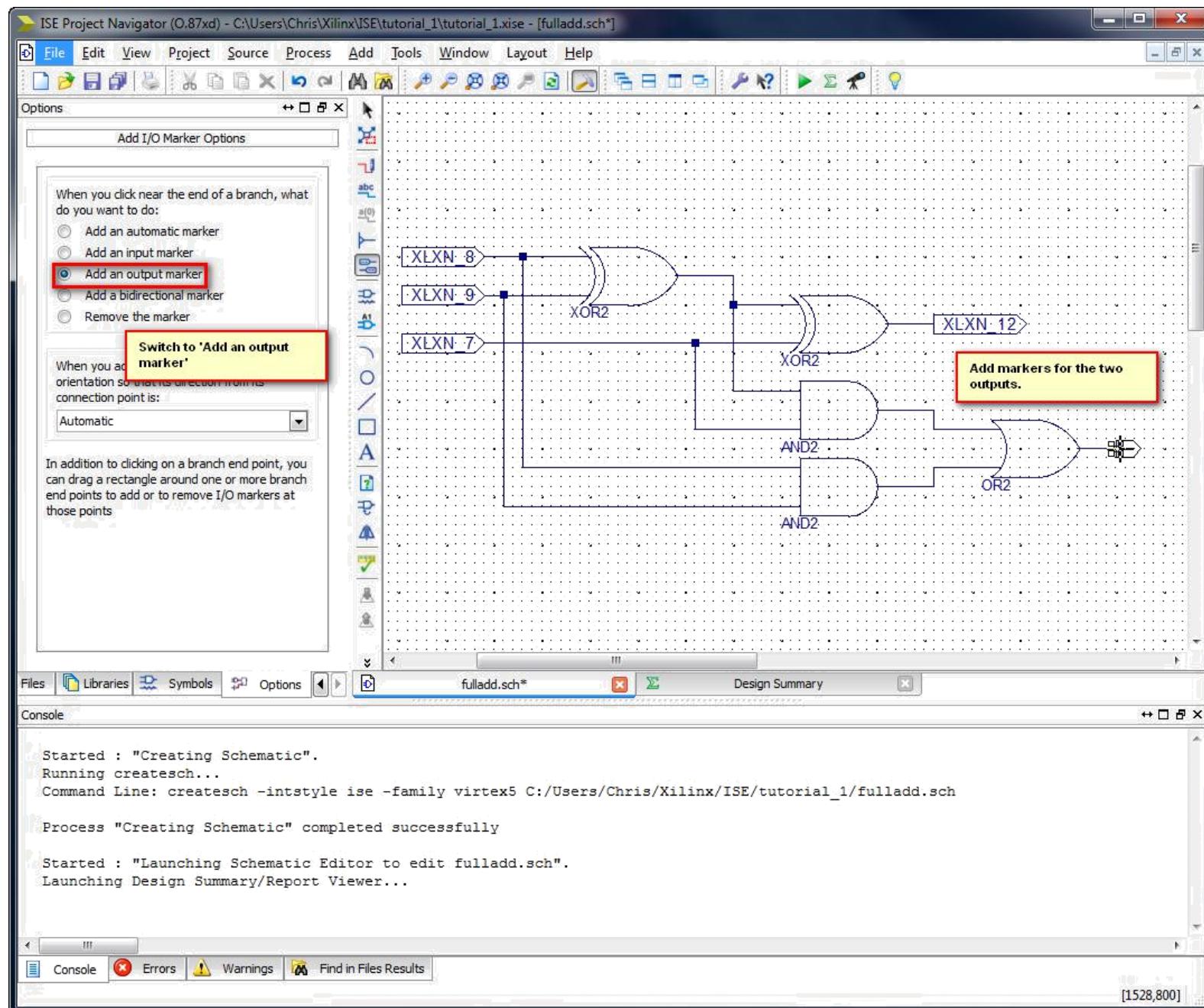


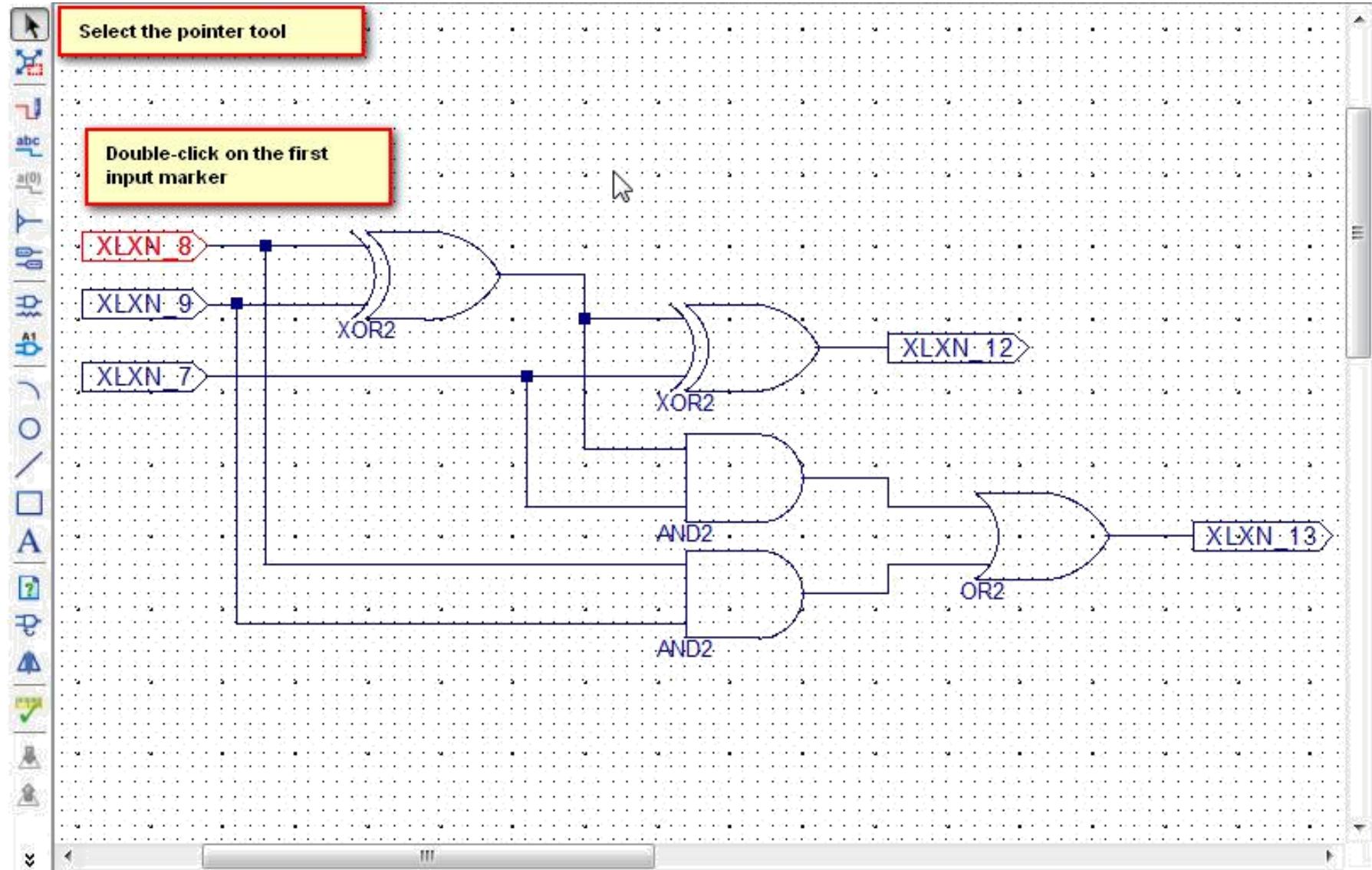


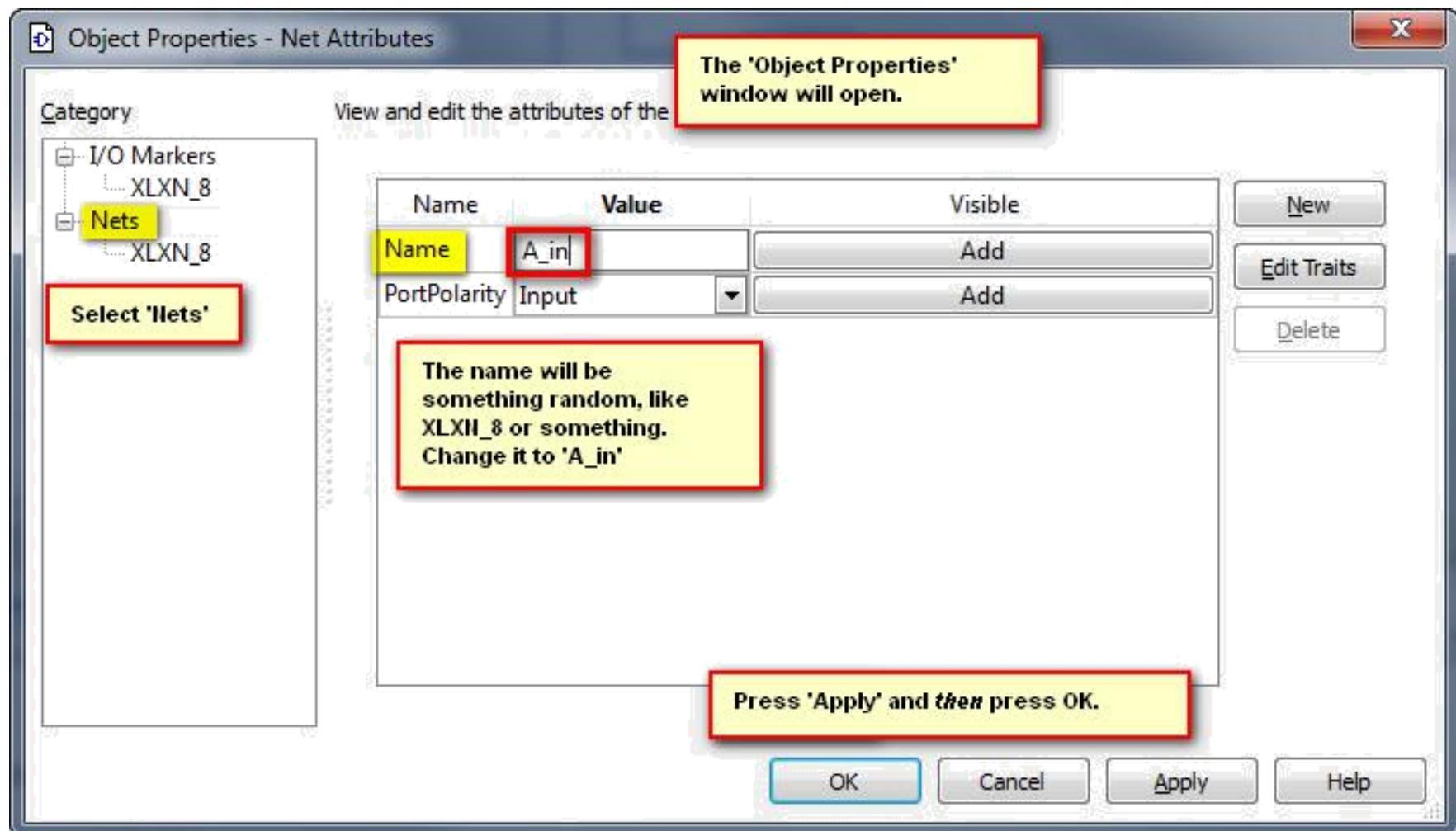


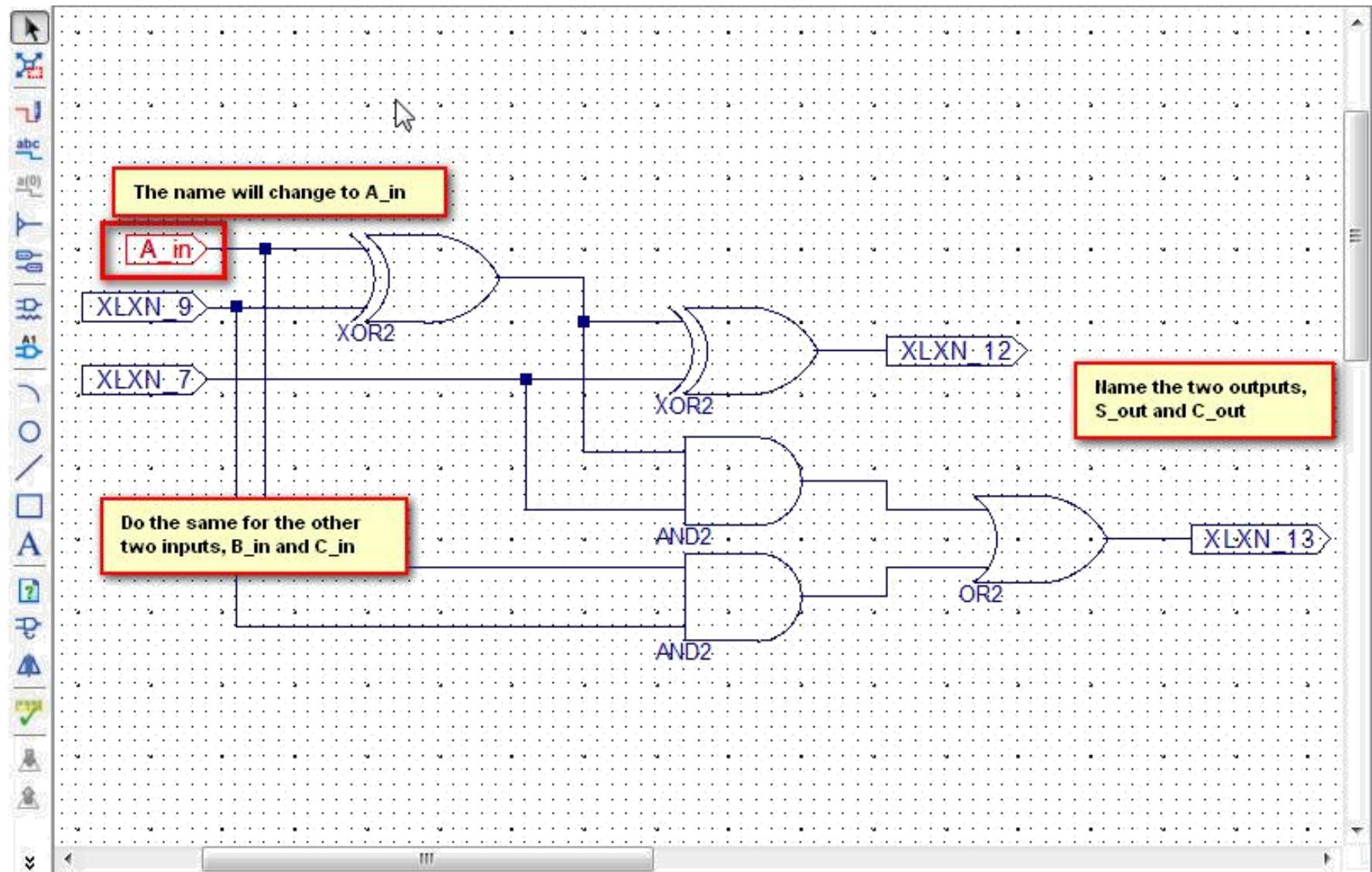


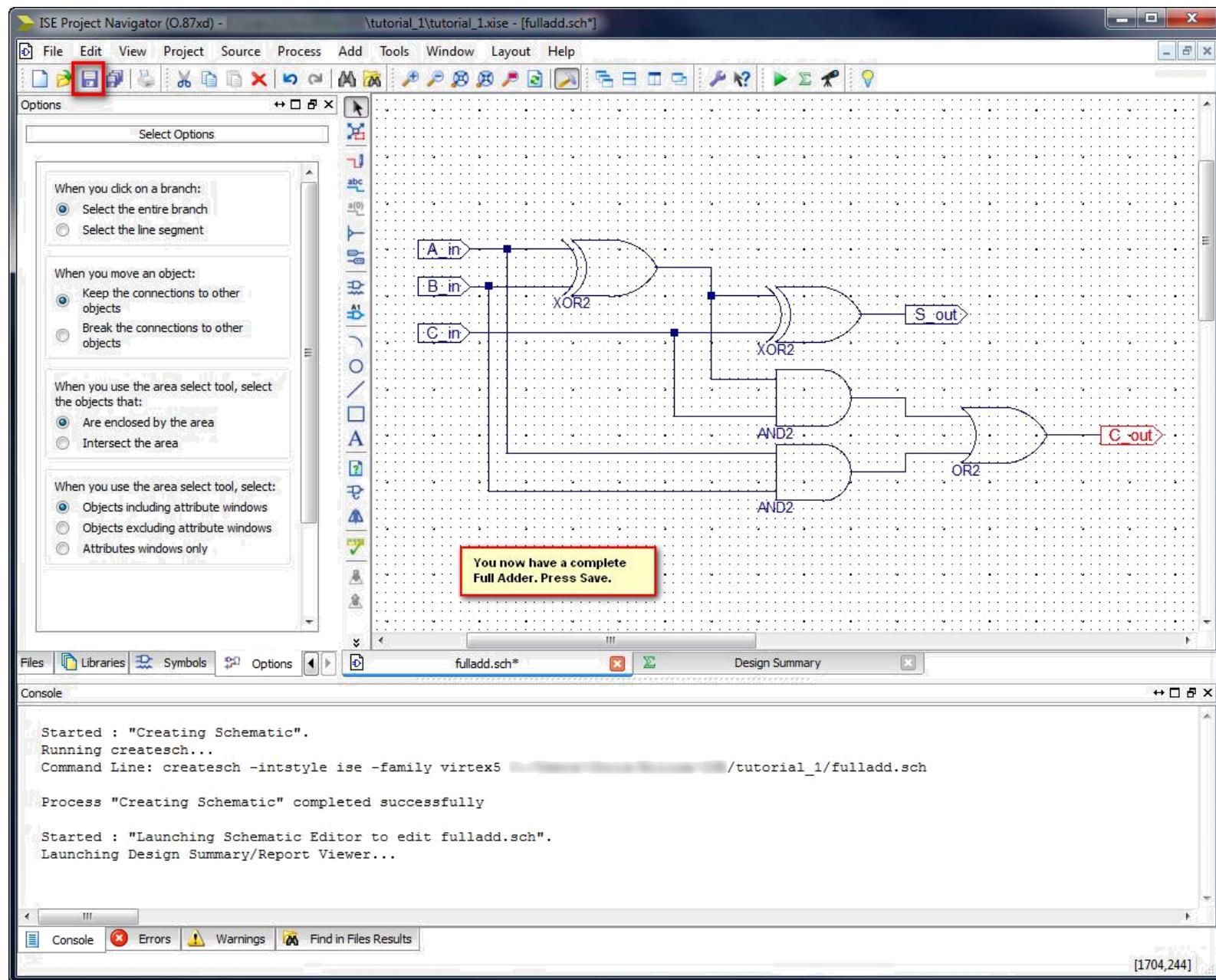


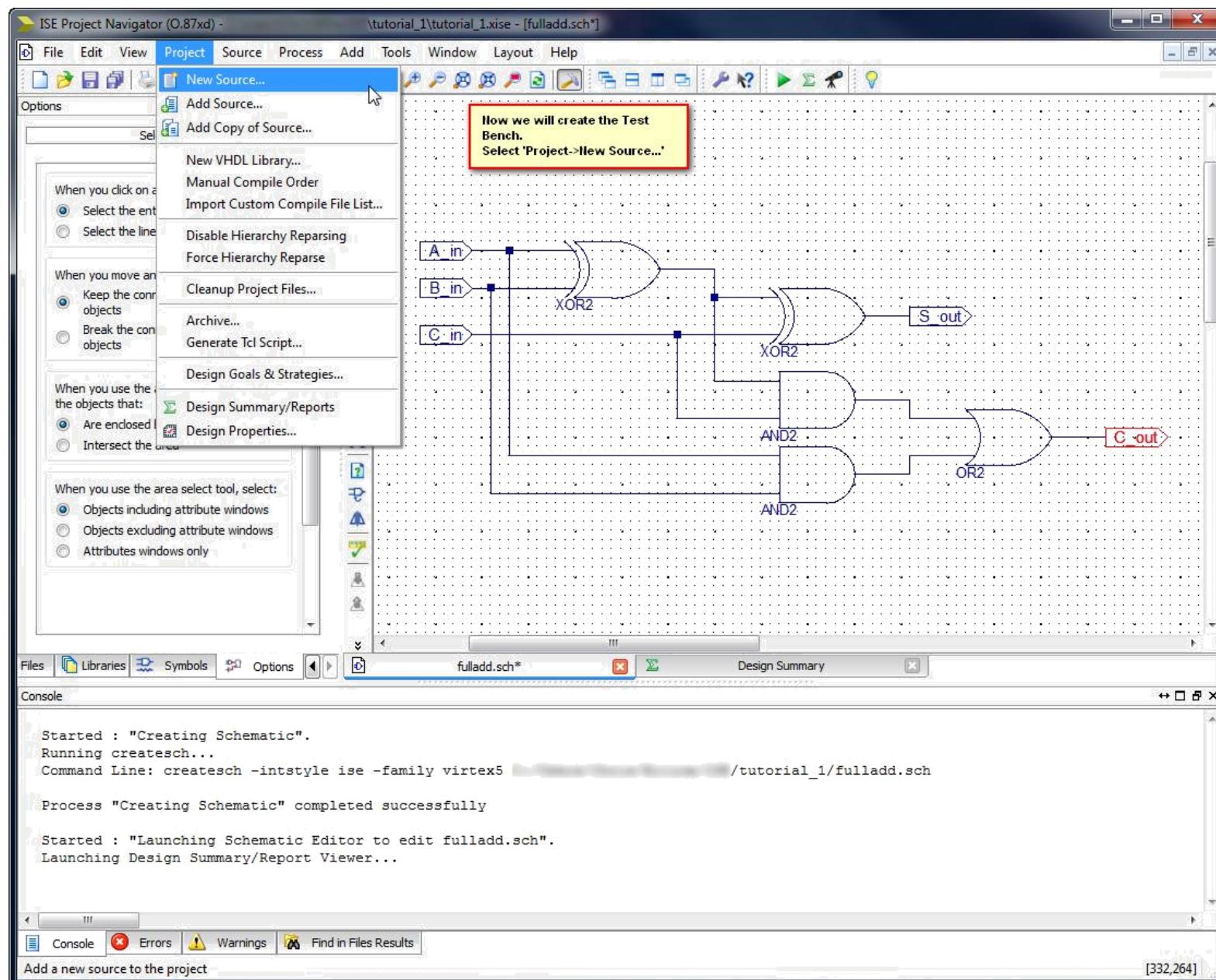












➤ New Source Wizard

Select Source Type

Select source type, file name and its location.

- BMM File
- ChipScope Definition and Connection File
- Implementation Constraints File
- IP (CORE Generator & Architecture Wizard)
- MEM File
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

Select 'Verilog Test Fixture'
and name it fulladd_testbench

File name:

fulladd_testbench

Location:

\tutorial_1

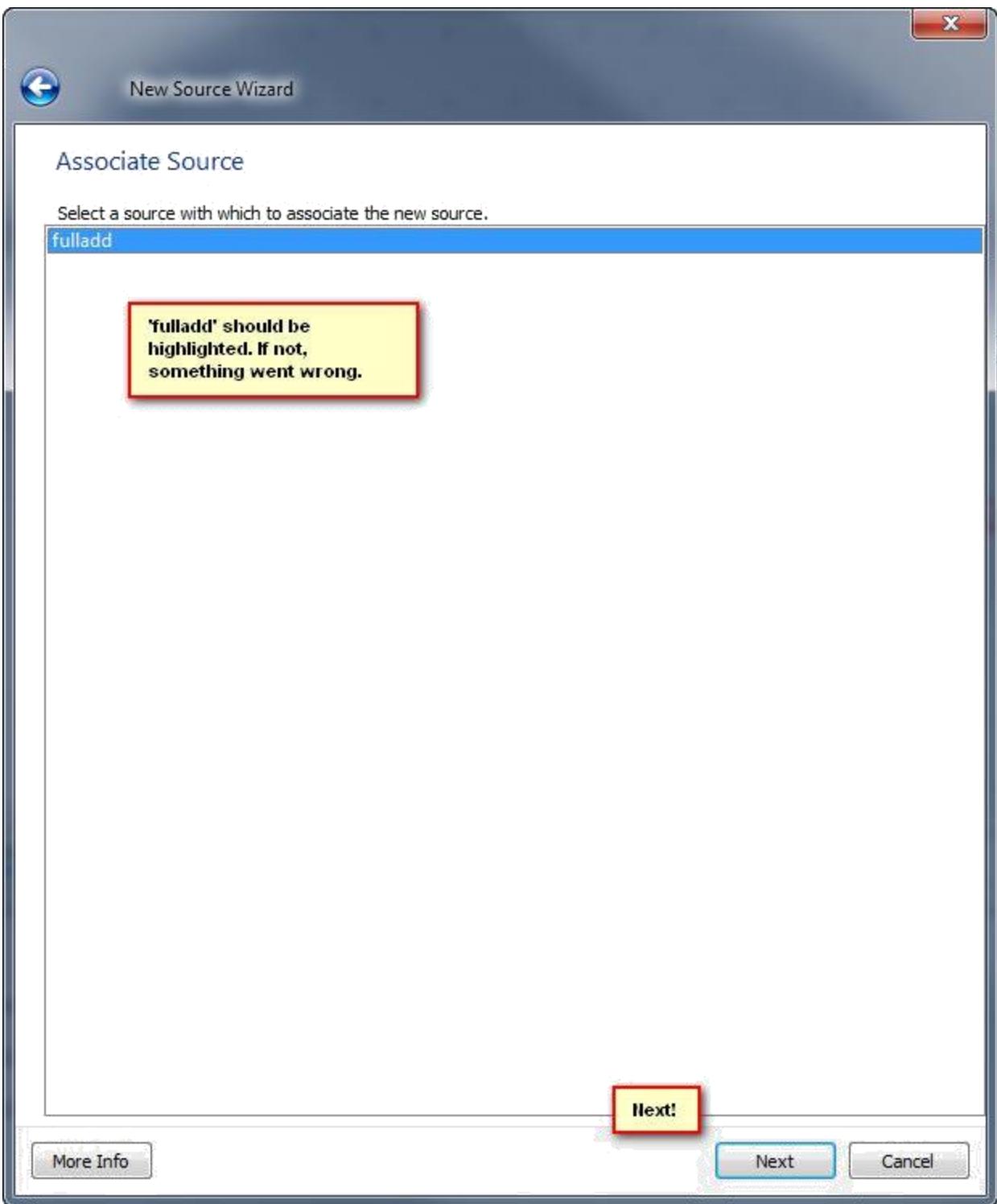
Add to project

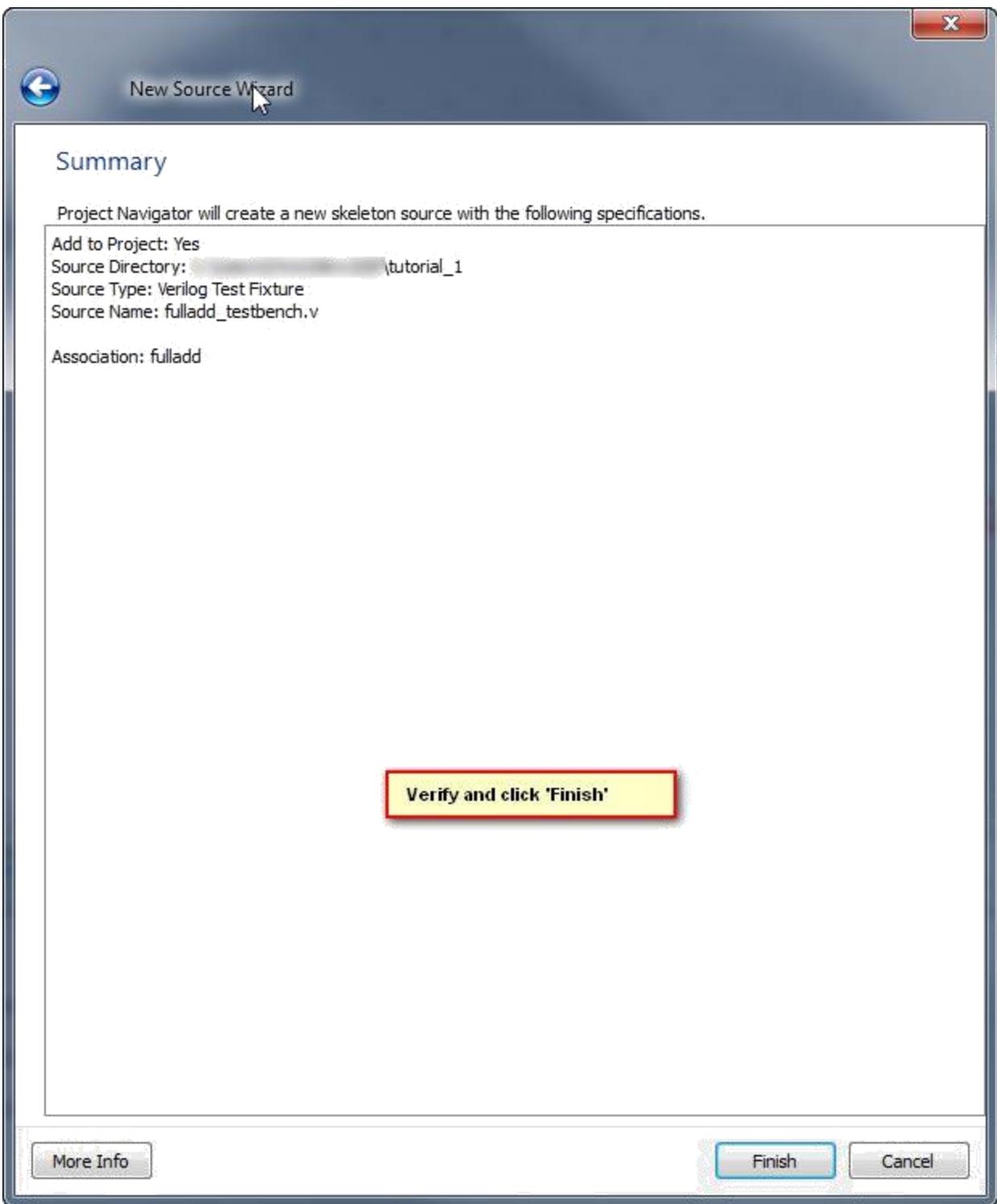
Click Next...

More Info

Next

Cancel





```
1 // Verilog test fixture created from schematic C:\Users\Chris\Xilinx\ISE\tutorial_1\fulladd_tb.v
2
3 `timescale 1ns / 1ps
4
5 module fulladd_fulladd_sch_tb();
6
7 // Inputs
8
9 // Output
10
11 // Bidirs
12
13 // Instantiate the UUT
14     fulladd UUT (
15
16 );
17 // Initialize Inputs
18     `ifdef auto_init
19         initial begin
20             `endif
21     endmodule
22
```

It will open a new tab in the work area, which will have the generated Verilog code. We will modify this for our tests.

```
1 // Verilog test fixture created from schematic C:\Users\Chris\Xilinx\ISE\tutorial_1\full
2
3 `timescale 1ns / 1ps
4
5 module fulladd_fulladd_sch_tb();
6
7 // Inputs
8 reg C_in;
9 reg A_in;
10 reg B_in;
11
12 // Output
13 wire S_out;
14 wire C_out;
15
16 // Bidirs
17
18 // Instantiate the UUT
19 fulladd UUT (
20     .C_in(C_in),
21     .A_in(A_in),
22     .B_in(B_in),
23     .S_out(S_out),
24     .C_out(C_out)
25 );
26 // Initialize Inputs
27 `ifdef auto_init
28     initial begin
29         C_in = 0;
30         A_in = 0;
31         B_in = 0;
32     `endif
33 endmodule
```

Declares the inputs and outputs for the testbench. These do not have to match the inputs/outputs for your design, but these are the names the simulator will use.

Inputs are type 'reg' and outputs are type 'wire'

This maps the registers and wires in the testbench with the inputs/outputs of the fulladd schematic. The name after the dot is the name of the fulladd input/output, and the name in parentheses is the name of the wire/reg in the testbench that it connects to.

This sets the initial values of the inputs to zero.

A new tab will open in the Work Area which contains the Verilog code for the testbench.

Xilinx automatically generates some of the testbench design.

```
15
16 // Bidirs
17
18 // Instantiate the UUT
19     fulladd UUT (
20         .C_in(C_in),
21         .A_in(A_in),
22         .B_in(B_in),
23         .S_out(S_out),
24         .C_out(C_out)
25     );
26 // Initialize Inputs
27 initial begin
28     A_in=0;
29     B_in=0;
30     C_in=0;
31
32     #100;           Wait 100 ns
33
34     C_in=1;          Set C_in to 1
35
36     #100;           Wait 100 ns
37
38     A_in=1;          Set C_in to 1
39
40     #100;           Wait 100 ns
41
42     B_in=1;          Set C_in to 1
43
44     #100;           Wait 100 ns
45 end
46
47 endmodule
```

Remove the 'ifdef auto_init' and 'endif' lines. We will not use them, since we initialize manually.

Set A_in, B_in and C_in to 0.

Add the code as shown here between 'initial begin' and 'endmodule'

Wait 100 ns

Set C_in to 1

Wait 100 ns

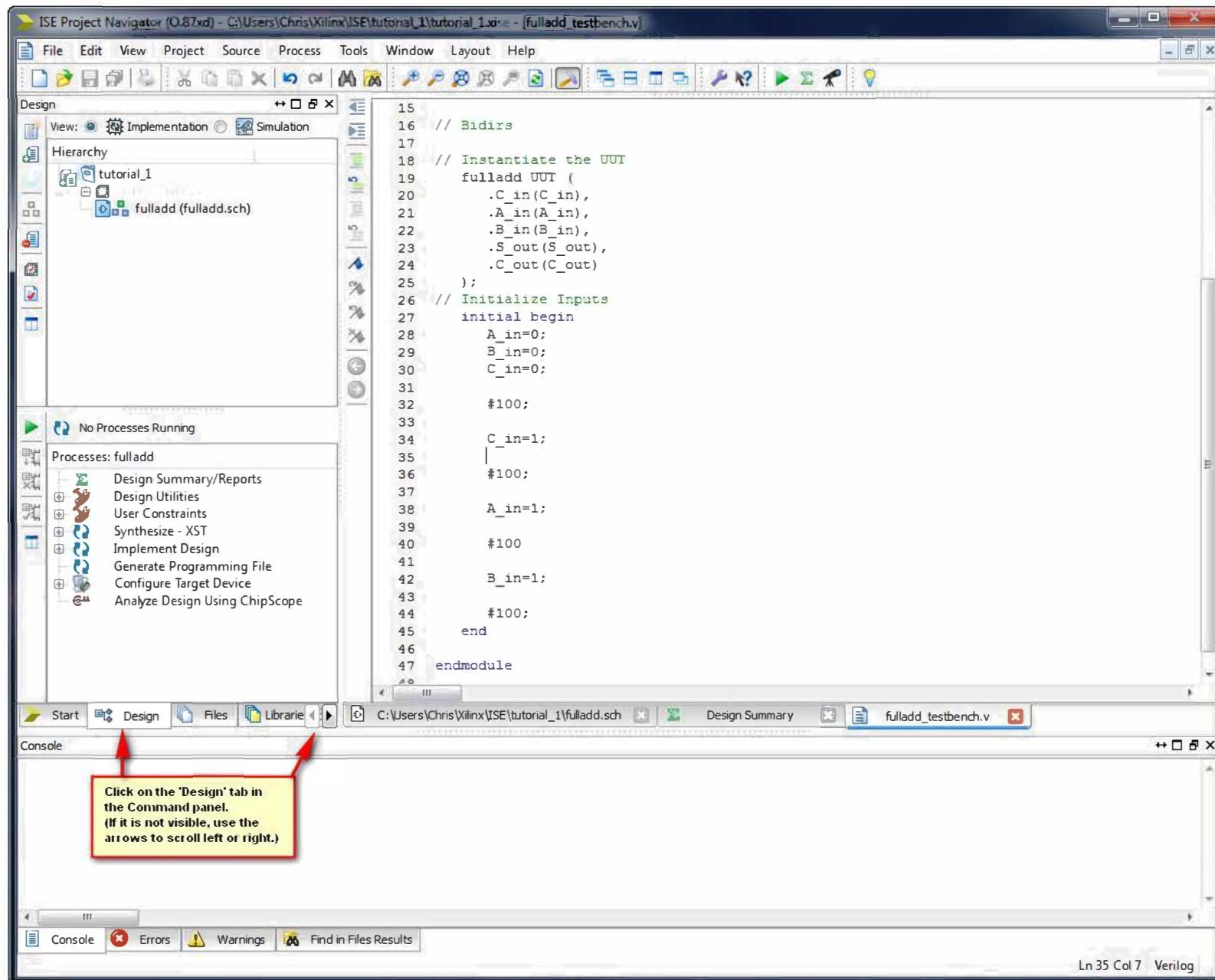
Set C_in to 1

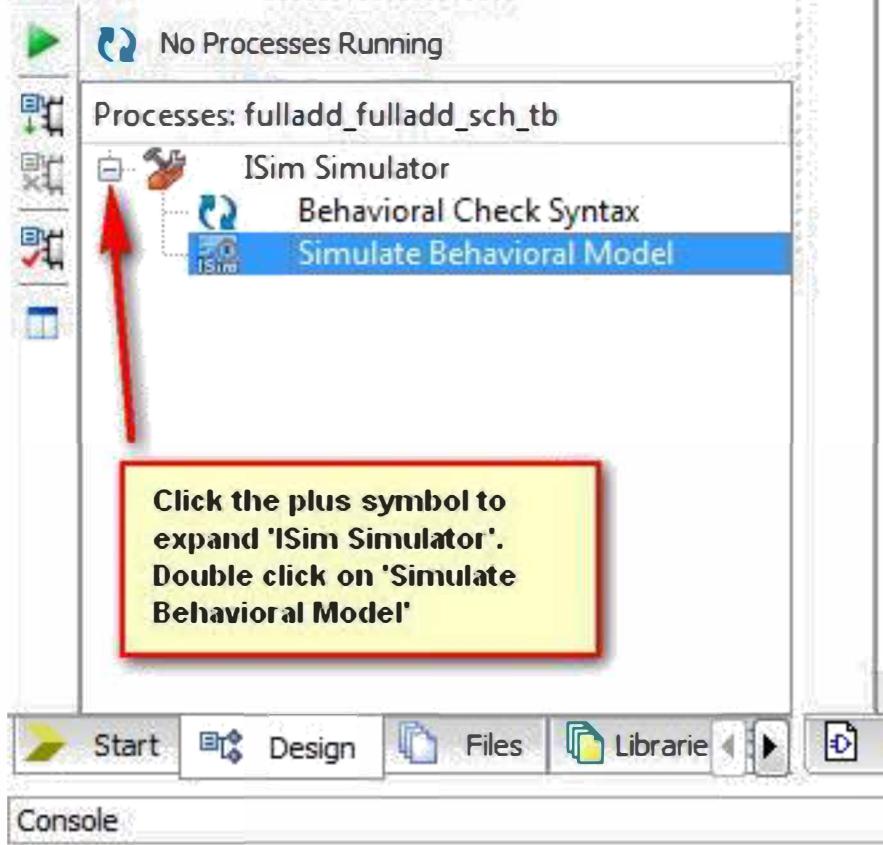
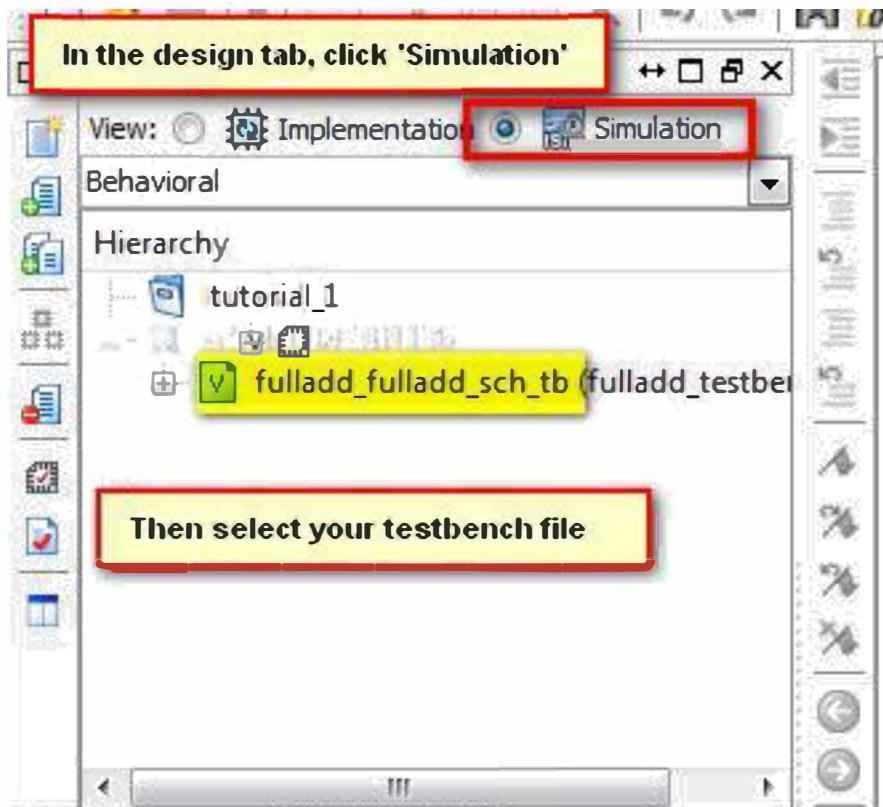
Wait 100 ns

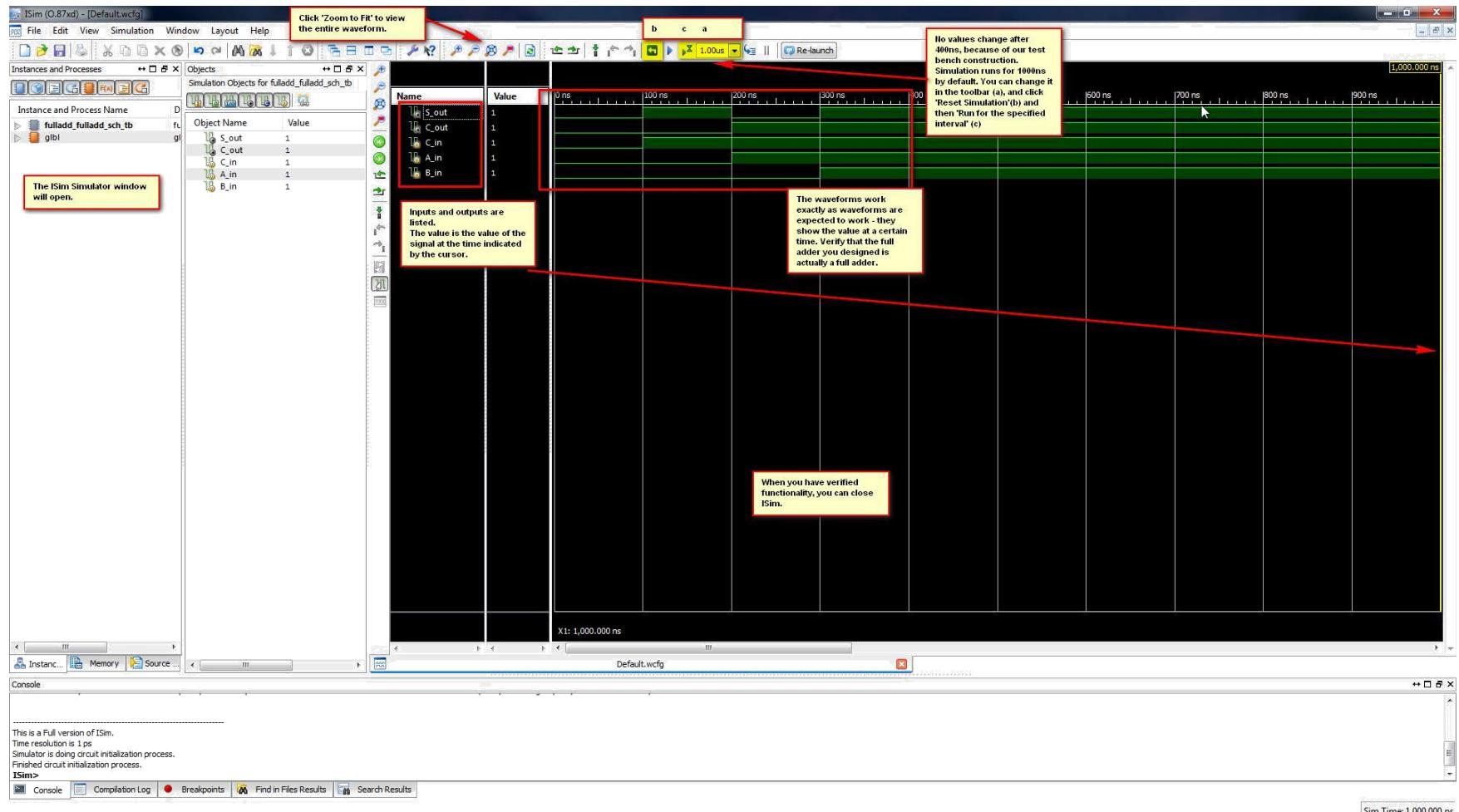
Set C_in to 1

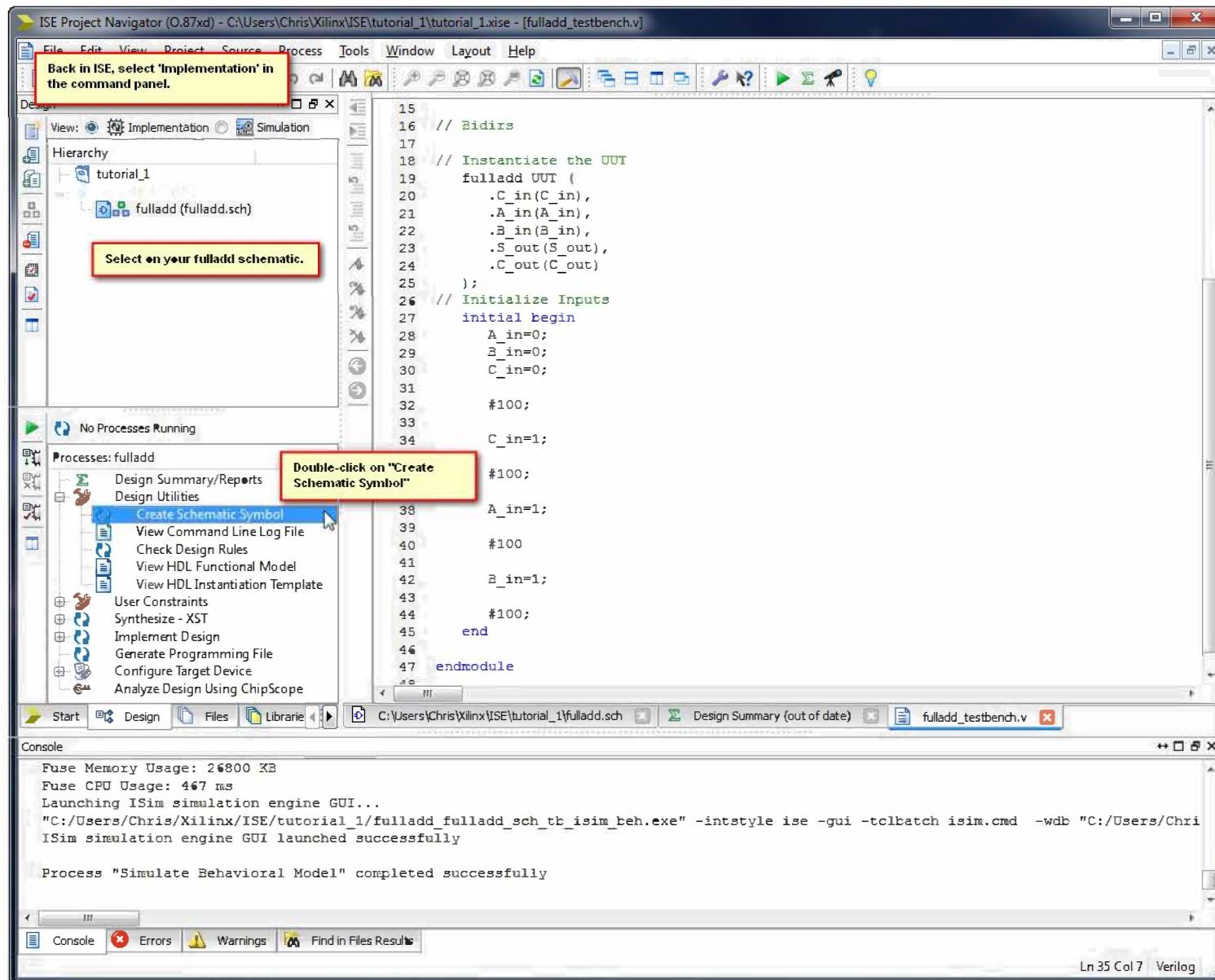
Wait 100 ns

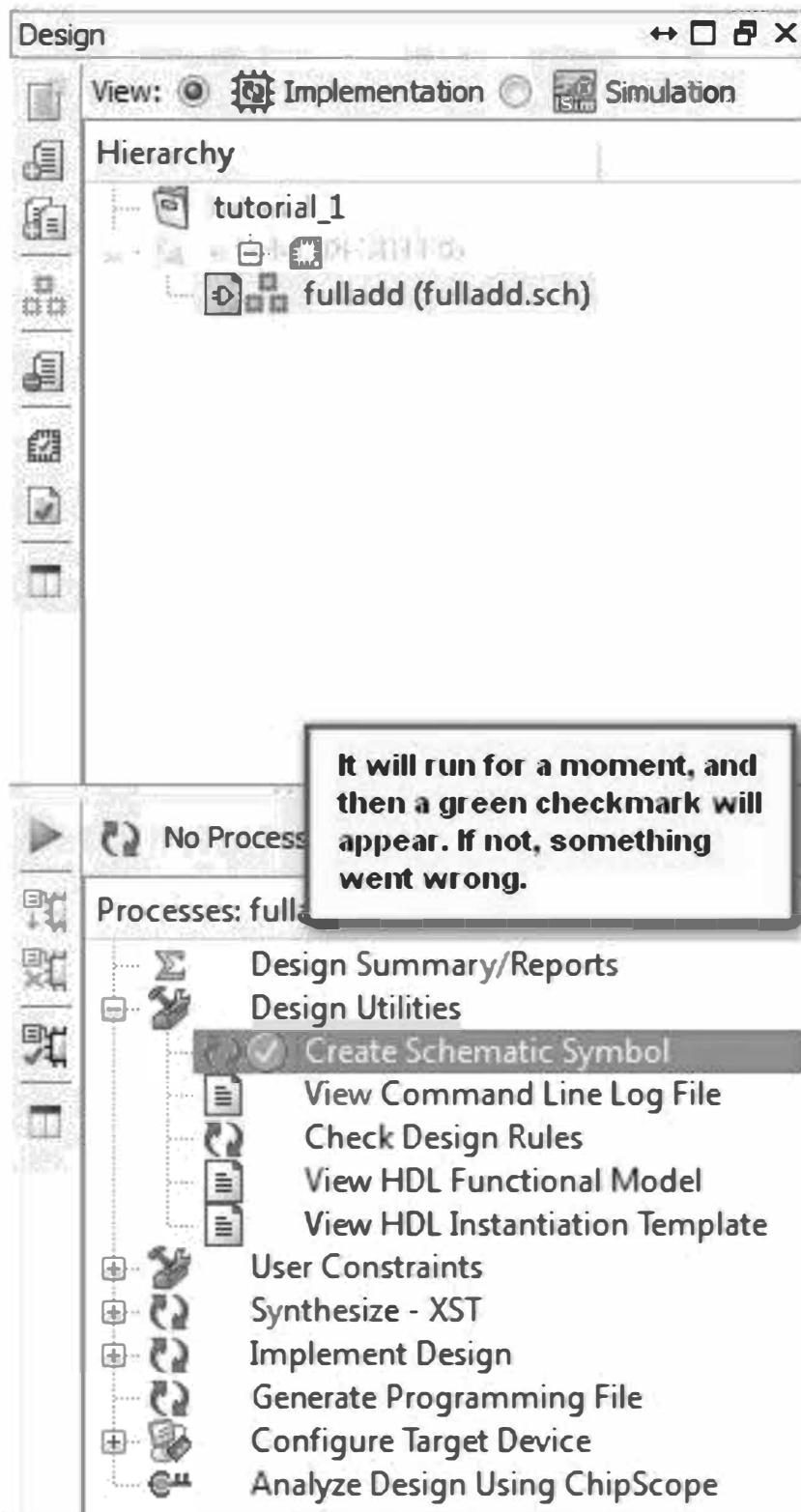
Done

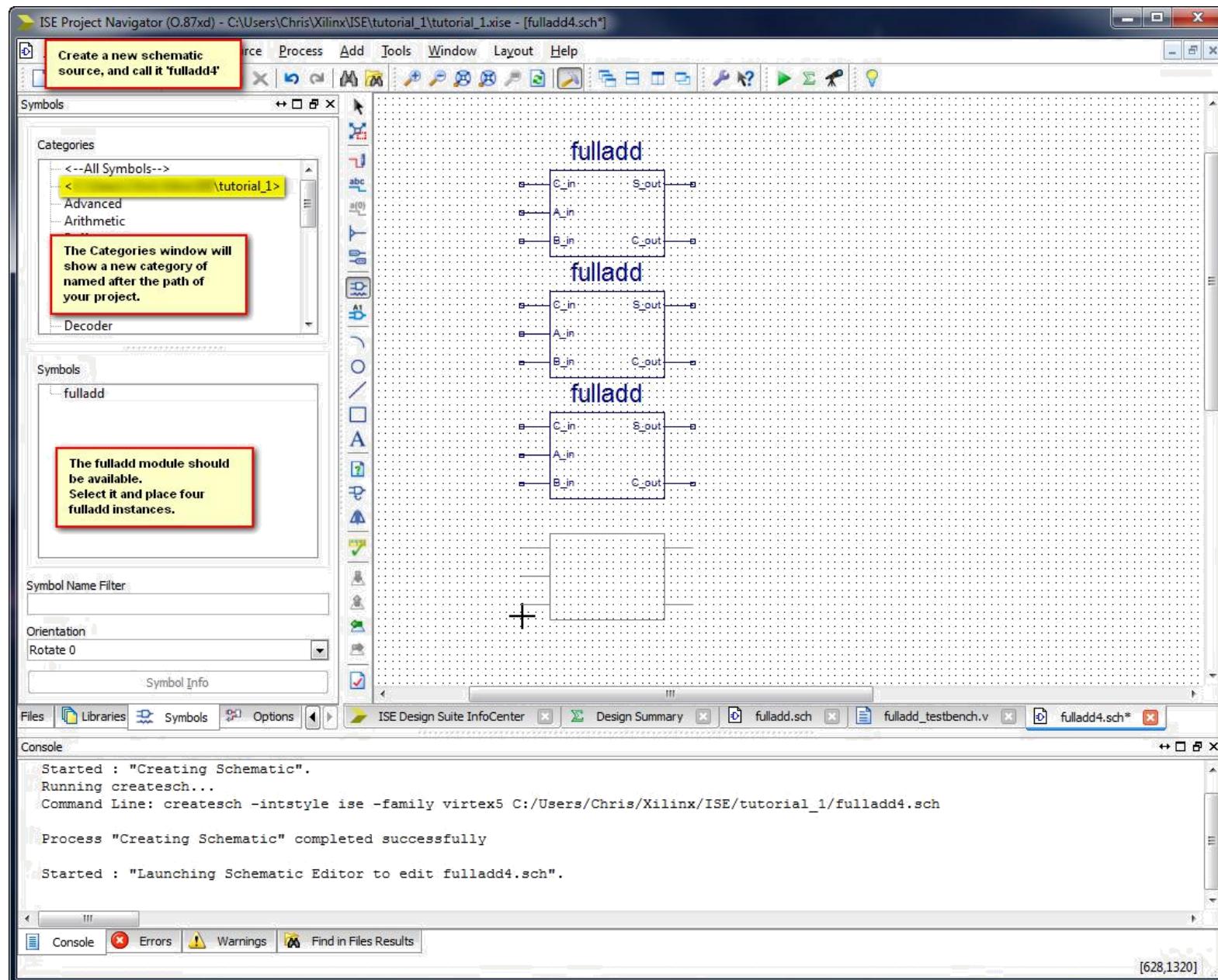


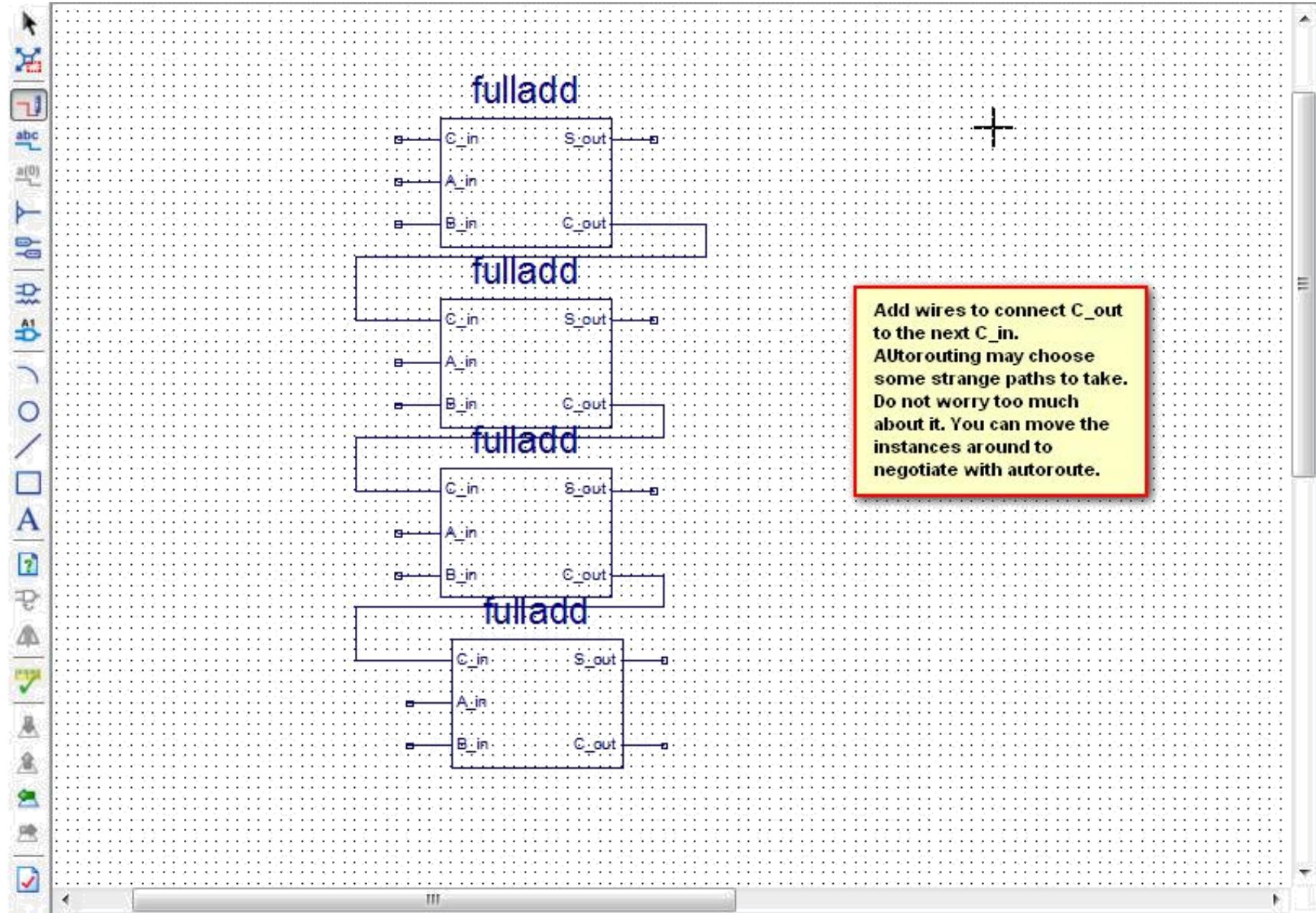


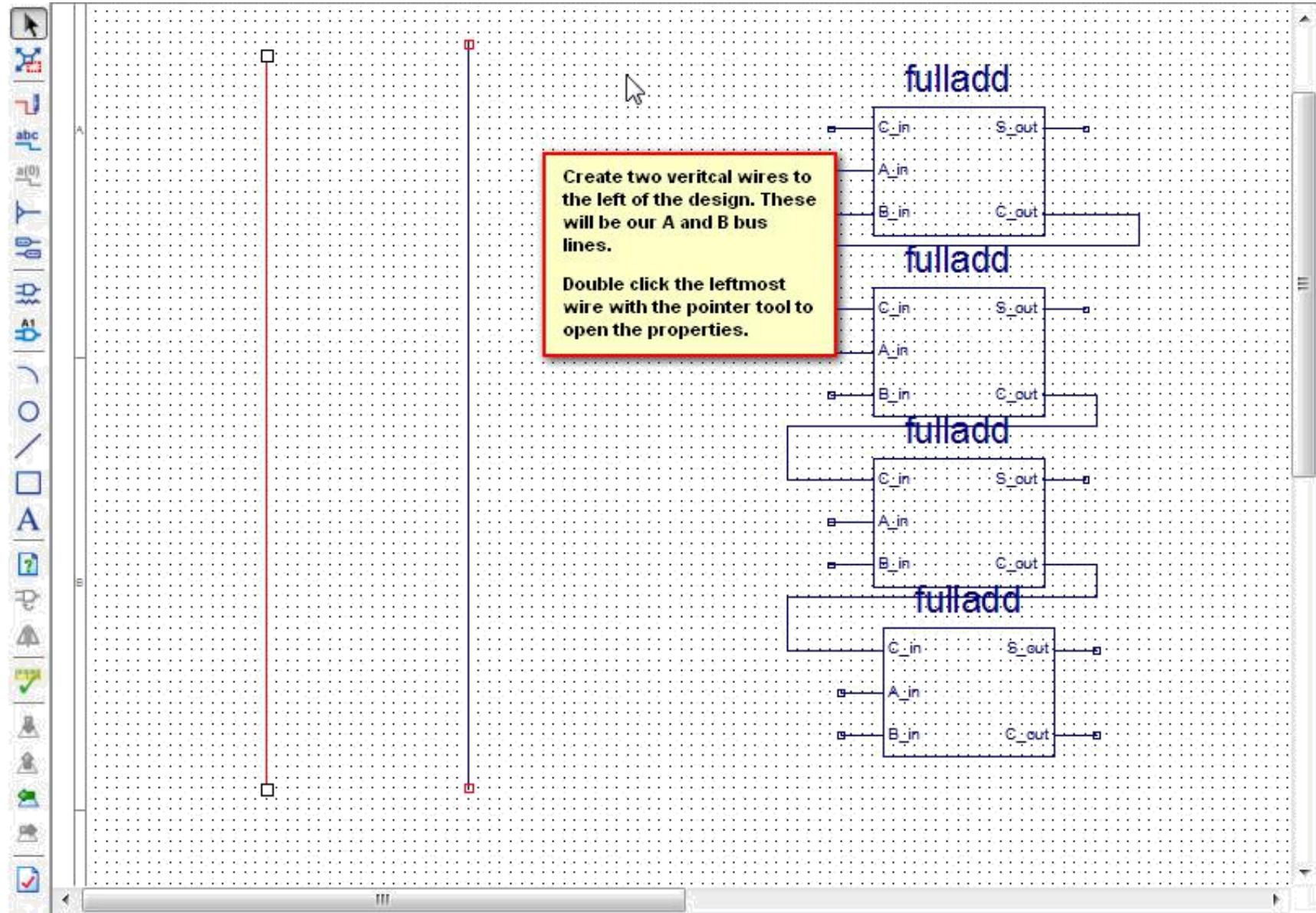


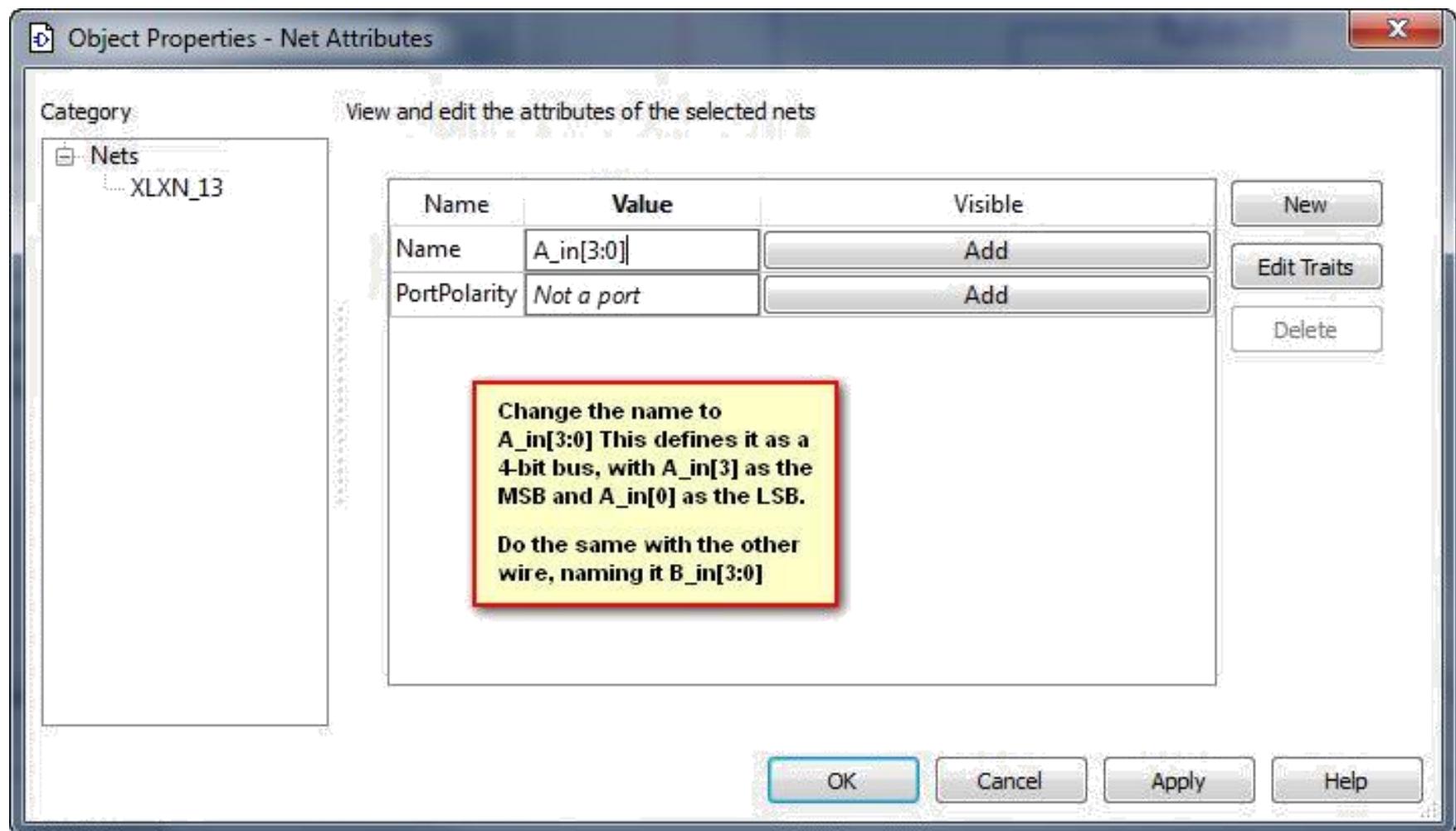


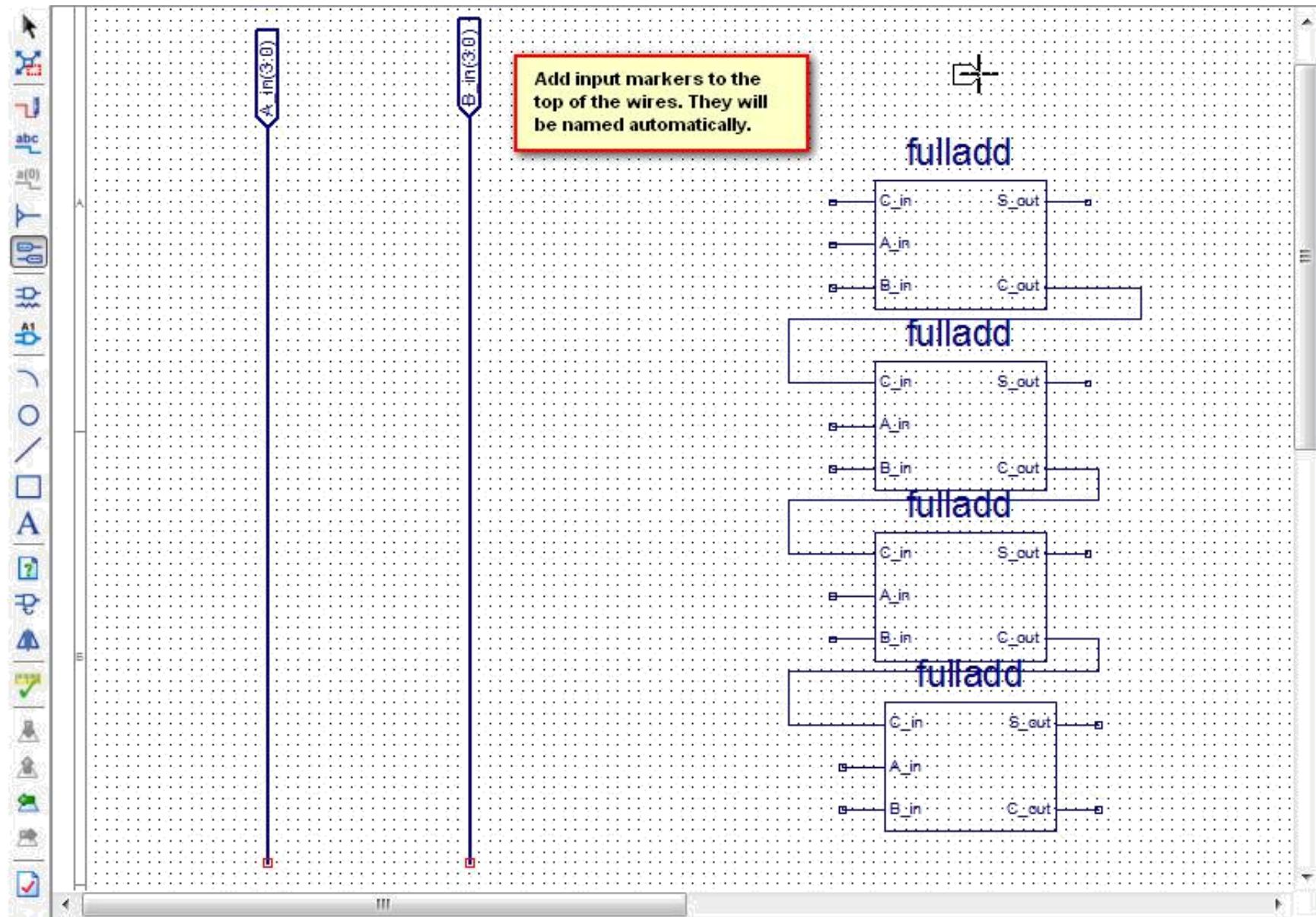


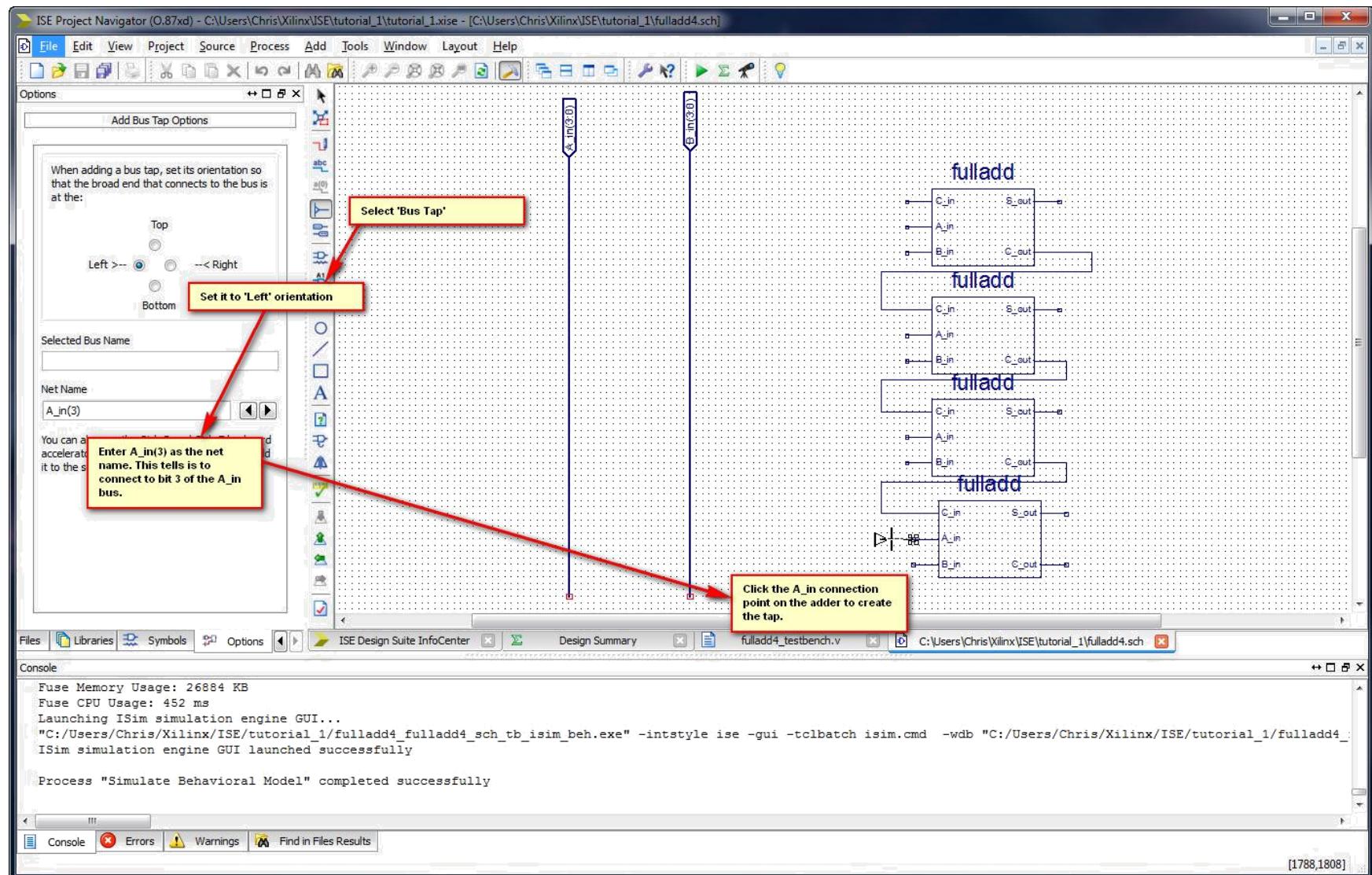


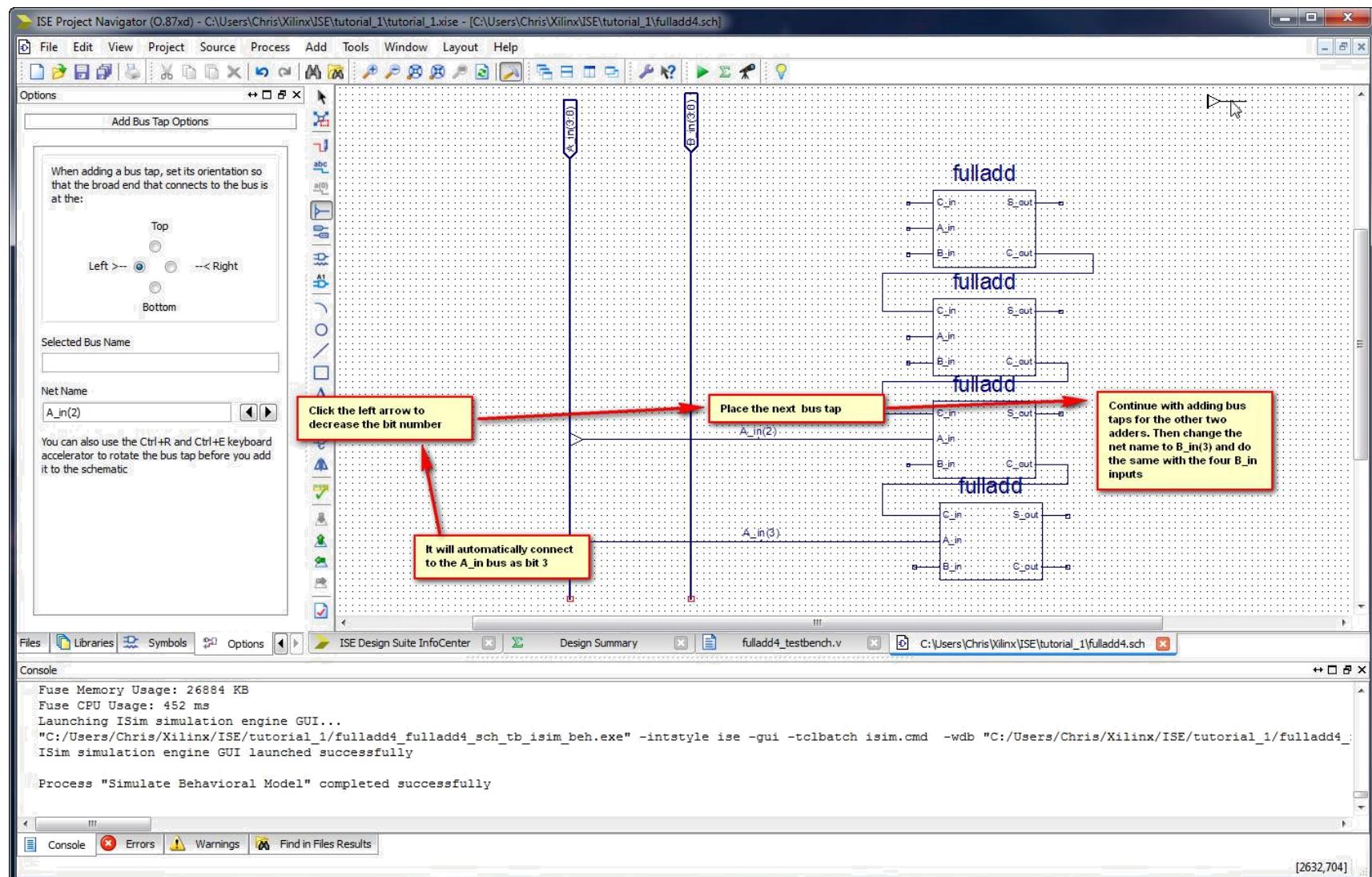


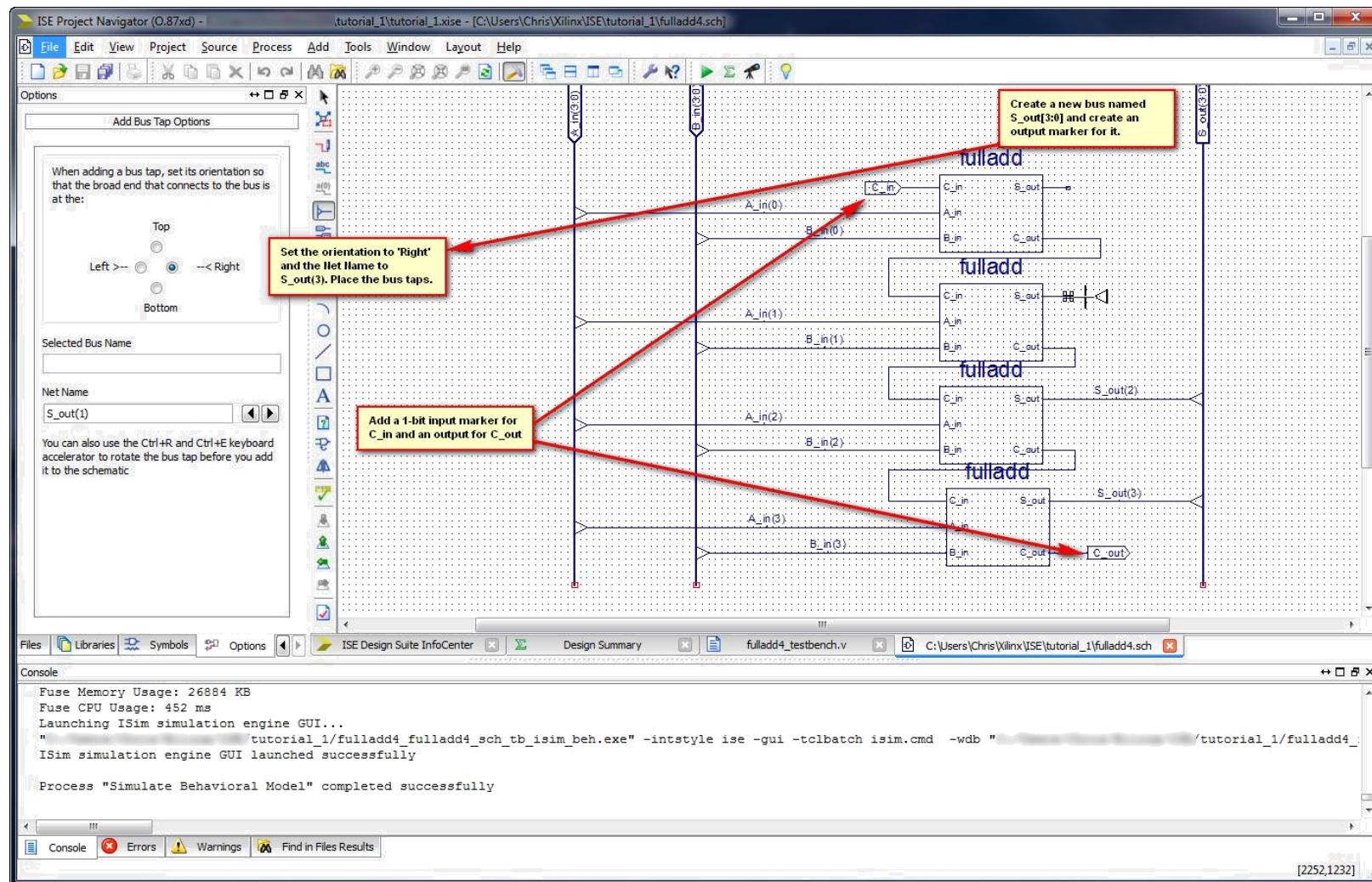












New Source Wizard

Select Source Type

Select source type, file name and its location.

- BMM File
- ChipScope Definition and Connection File
- Implementation Constraints File
- IP (CORE Generator & Architecture Wizard)
- MEM File
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

Create a new Verilog Test
Fixture and name it
fulladd4_testbench

File name:

fulladd4_testbench

Location:

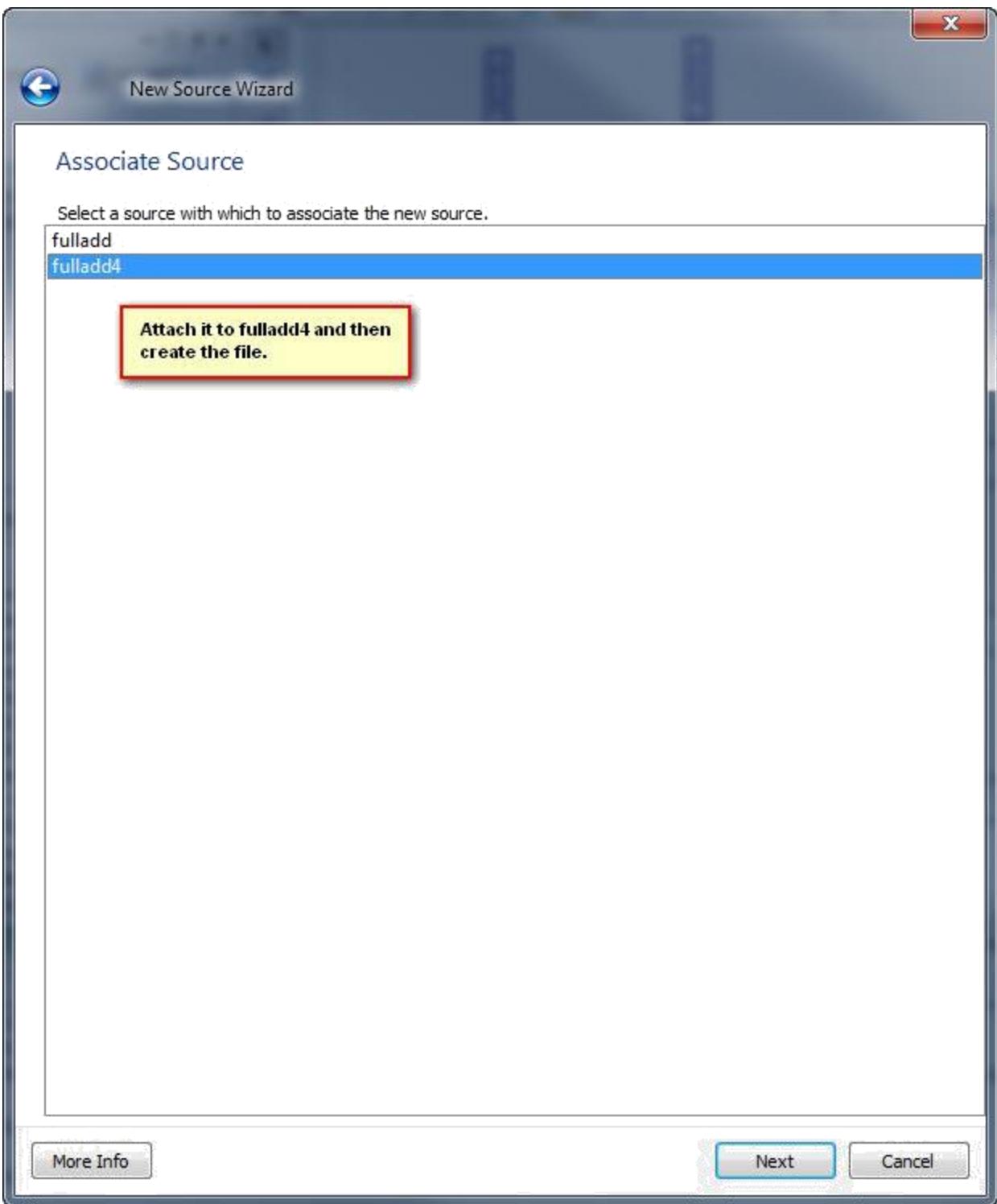
C:\Users\...\tutorial_1

Add to project

More Info

Next

Cancel



ISE Project Navigator (O.87xd) - C:\Users\challa1\Xilinx\tutorial_10\tutorial_10.xise - [fulladd4_testbench.v]

File Edit View Project Source Process Tools Window Layout Help

Design Implementation Simulation

Hierarchy

View: Implementation Simulation

Behavioral

tutorial_10
 xc6sx5-3csg24
 fulladd4_fulladd4_sch_tb (fulladd4_testbench.v)
 UUT - fulladd4 (fulladd4.sch)
 XLXI_1 - fulladd (fulladd.sch)
 XLXI_2 - fulladd (fulladd.sch)
 XLXI_3 - fulladd (fulladd.sch)
 XLXI_4 - fulladd (fulladd.sch)
 fulladd4_fulladd_sch_tb (fulladd_testbench.v)
 UUT - fulladd (fulladd.sch)

4
5 module fulladd4_fulladd4_sch_tb();
6
7 // Inputs
8 reg [3:0] A_in;
9 reg [3:0] B_in;
10 reg C_in;
11
12 // Output
13 wire [3:0] S_out;
14 wire C_out;
15
16 // Bidirs
17
18 // Instantiate the UUT
19 fulladd4 UUT (
20 .A_in(A_in[3:0]),
21 .B_in(B_in[3:0]),
22 .C_in(C_in),
23 .S_out(S_out[3:0]),
24 .C_out(C_out)
25);
26 // Initialize Inputs
27
28 initial begin
29 A_in = 0;
30 B_in = 0;
31 C_in = 0;
32
33 #100;
34
35 A_in = 4;
36 B_in = 3;
37
38 #100;
39
40 A_in = 2;
41 B_in = 3;
42 C_in = 1;
43
44 #100;
45
46 A_in = 7;
47
48 end
49
50 endmodule
51

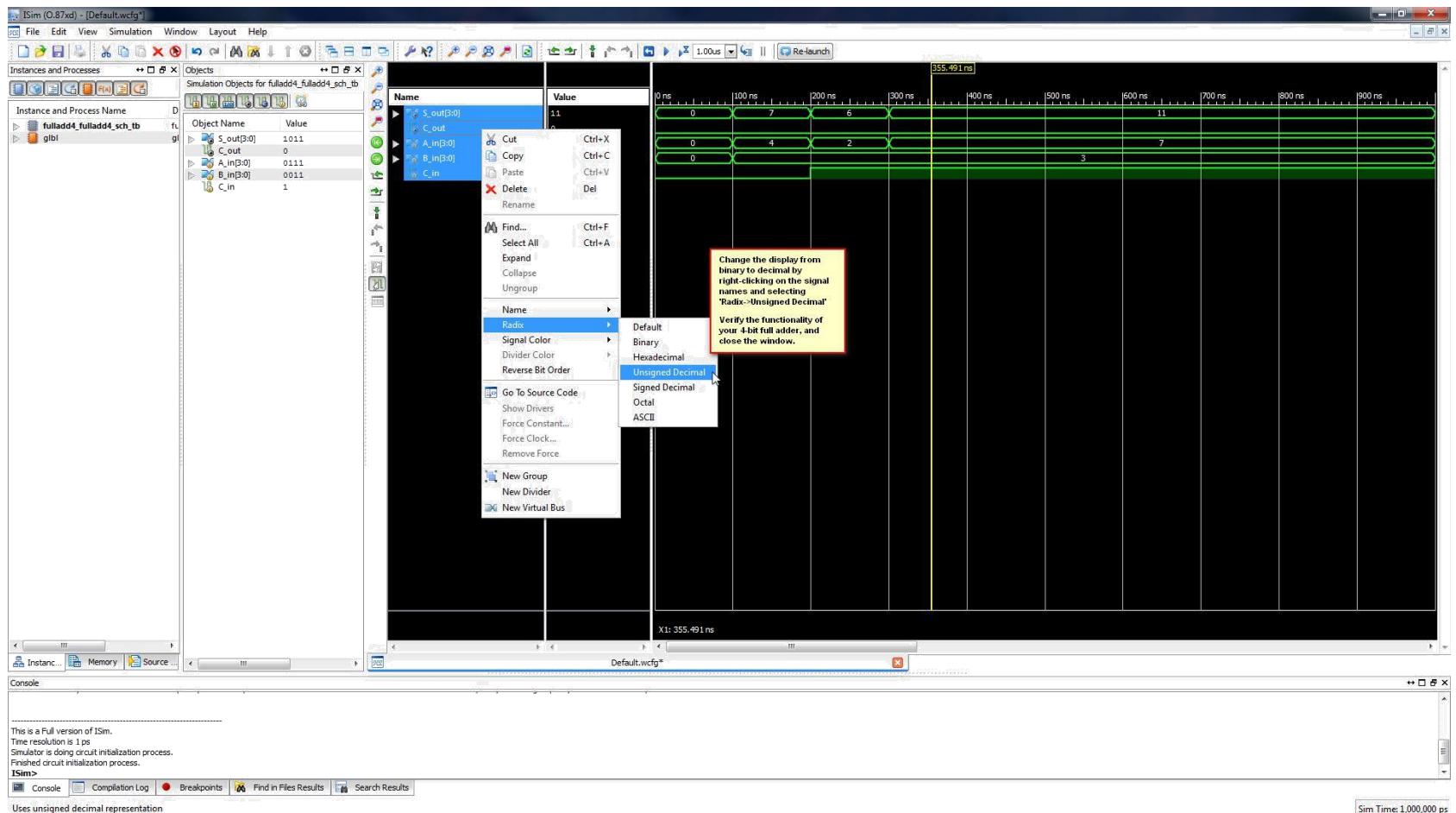
No Processes Running

Processes: fulladd4_fulladd4_sch_tb
 iSim Simulator
 Behavioral Check Syntax
 Simulate Behavioral Model

Design Files Libraries Symbols Options ISE Design Suite InfoCenter Design Summary fulladd_testbench.v C:\Users\challa1\Xilinx\tutorial_10\fulladd4.sch fulladd4_testbench.v Support Keyword Search fulladd.sch

Errors

Ln 22 Col 20 | Verilog



On your own:

- Create a schematic symbol for the 4-bit adder.
- Create an 8-bit adder from two 4-bit adders. (You do not need to make a symbol)

If you get the error: failed to link the design while performing tutorial 1, please perform the following steps to remove this error.

On windows 10, Find the "installation directory(Windows(C:) \ Xilinx \ 14.x \ ISE_DS \ ISE \ gnu \ MinGW \ 5.0.0 \ nt \ libexec \ gcc \ mingw32 \ 3.4.2" and delete collect2.exe and re-run the emulator.