



ANADOLU UNIVERSITY

**DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING**

EEM 334 - Digital Systems II

LAB 1 - INTRODUCTION TO XILINX ISE
SOFTWARE AND FPGA

1. PURPOSE

In this lab, after you learn to use the basic functionality of ISE platform you will learn to simulate your design on ISE.

2. BACKGROUND

2.1. FPGA Architecture

FPGAs are different than traditional integrated circuits, because interconnects between the gates are programmable, which means the **hardware** itself is configurable. FPGAs are an array semiconductor modules of Programmable Logic Blocks (PLB) interconnected with programmable interconnections. These PLBs can be simple standard logic gates, like AND, OR gates; as well as more complex blocks like decoders and memory blocks. These PLBs can be even more complex blocks like microprocessors.

The users that want to build large scale digital circuits need to program only the interconnections to build their circuitry. There exists programs to simplify and automatize the procedure to embed the designs onto the FPGAs. FPGA systems include hardware and software components to provide the users with a flexible and easy to use environment to implement mid-scale to very large scale digital systems.

Throughout this lab, we will use the hardware and software provided by the Xilinx company, which is the leader in FPGA industry.

!!! Please avoid putting your fingers on the chips, as they are extremely electrostatic discharge sensitive. !!!

2.2. Xilinx ISE Design Tools

Xilinx ISETM is the design tool provided by Xilinx. Xilinx provides a free version of this tool called WebpackTM which would be virtually identical for our purposes.

There are four fundamental steps in all digital logic design. These consist of:

1. Design – The schematic or code that describes the circuit.
2. Synthesis – The intermediate conversion of human readable circuit description to FPGA code (EDIF) format. It involves syntax checking and combining of all the separate design files into a single file.
3. Place & Route – Where the layout of the circuit is finalized. This is the translation of the EDIF into logic gates on the FPGA.
4. Program – The FPGA is updated to reflect the design through the use of programming (.bit) files.

Testbench simulation is in the second step. As its name implies, it is used for testing the design by simulating the result of driving the inputs and observing the outputs to verify your design.

ISE has the capability to do a variety of different design methodologies including: Schematic Capture, Finite State Machine and Hardware Descriptive Language (VHDL or Verilog).

2.3. 4 bit Adder Design

In this part we will outline how to build a digital full adder. It is called “full” because it will include a “carry-in” bit and a “carry-out” bit. The carry bits will allow a succession of 1-bit full adders to be used to add binary numbers of arbitrary length. (Half adder includes only one carry bit.)

After preparing half adder with using gate level design, we will build full adder with using two half adder and we will build 4 bit adder with using 4 full adder.

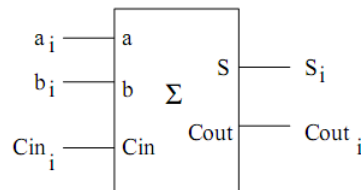


Figure (1) - Block schematic of full adder

Cin_i	a_i	b_i	S_i	$Cout_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure (2) - Truth table of full adder

We can design full adder according to truth table, gate by gate or by combining two half adder. To learn modular design procedure, we build the circuit by using half adders.

Inputs		Outputs	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure (3) - Truth table of half adder

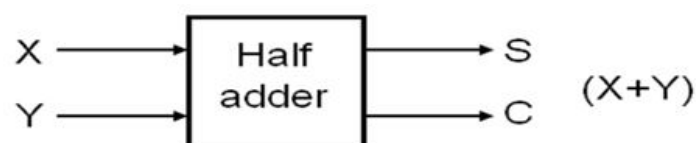


Figure (4) – Half adder block diagram

With using the truth table given above, we can obtain a logic gate equivalent of “sum” and “carry” output in the karnough map minimization like below.

$$C = X \cdot Y$$

$$S = X \oplus Y$$

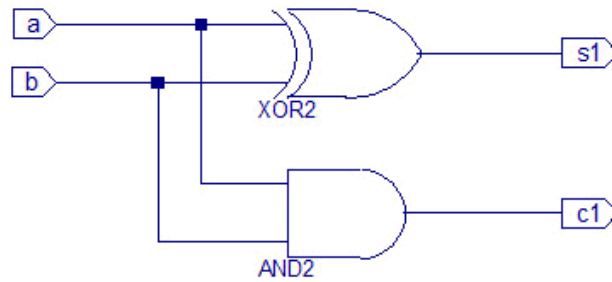


Figure (5) - Gate equivalent of half adder

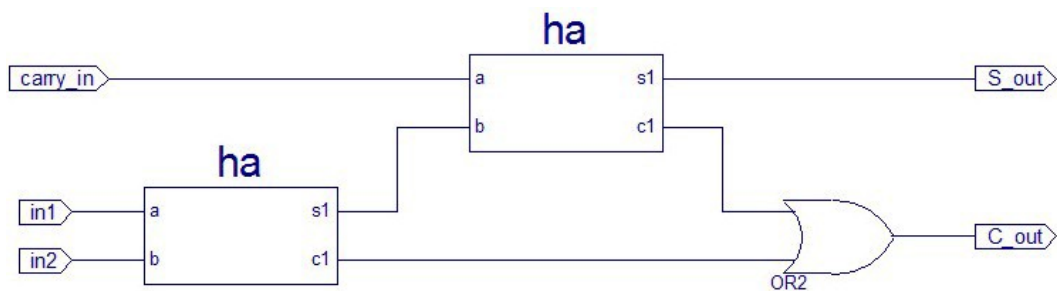


Figure (6) - Full adder circuit

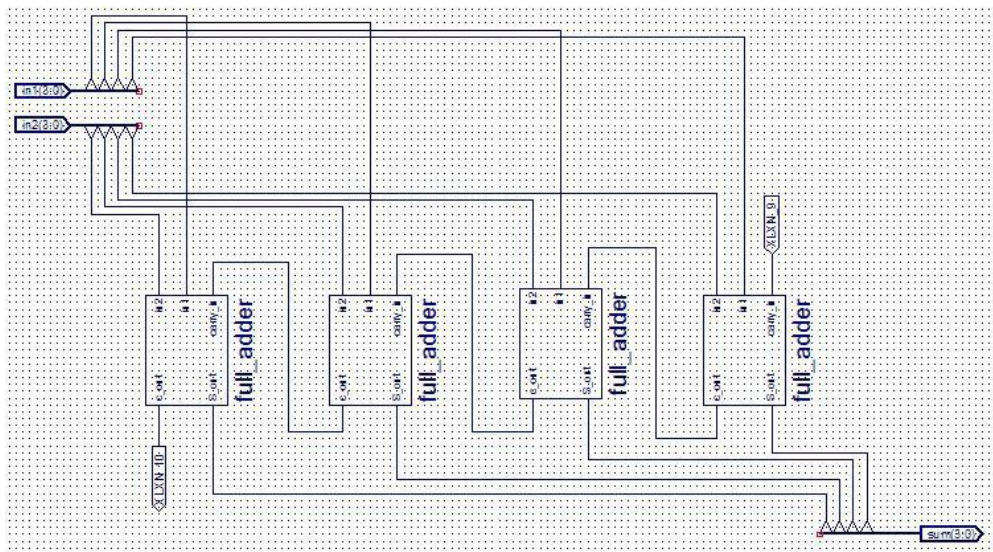


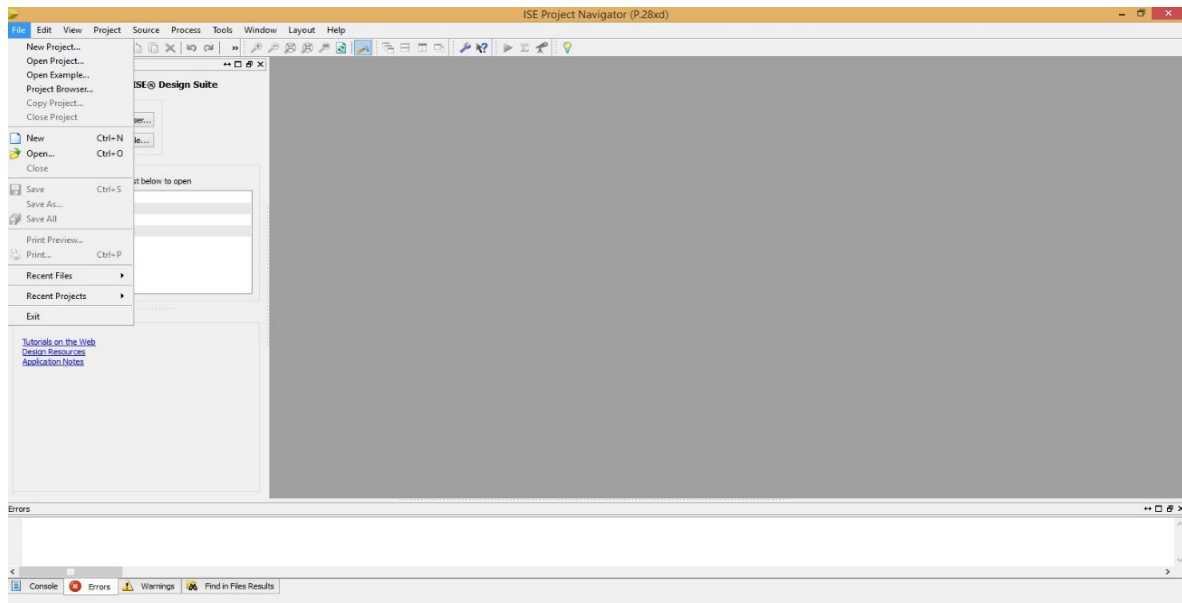
Figure (7) - Four Bit Ad

3. PROCEDURE

In this first lab, you will learn to use ISE software to build digital designs. First two weeks, you do not need to design anything for prelab, but you must learn the procedure for upcoming labs. You should follow the instructions step by step. Try to get used to the procedure since you will be using the same procedure in the upcoming labs as well.

3.1. Schematic Design

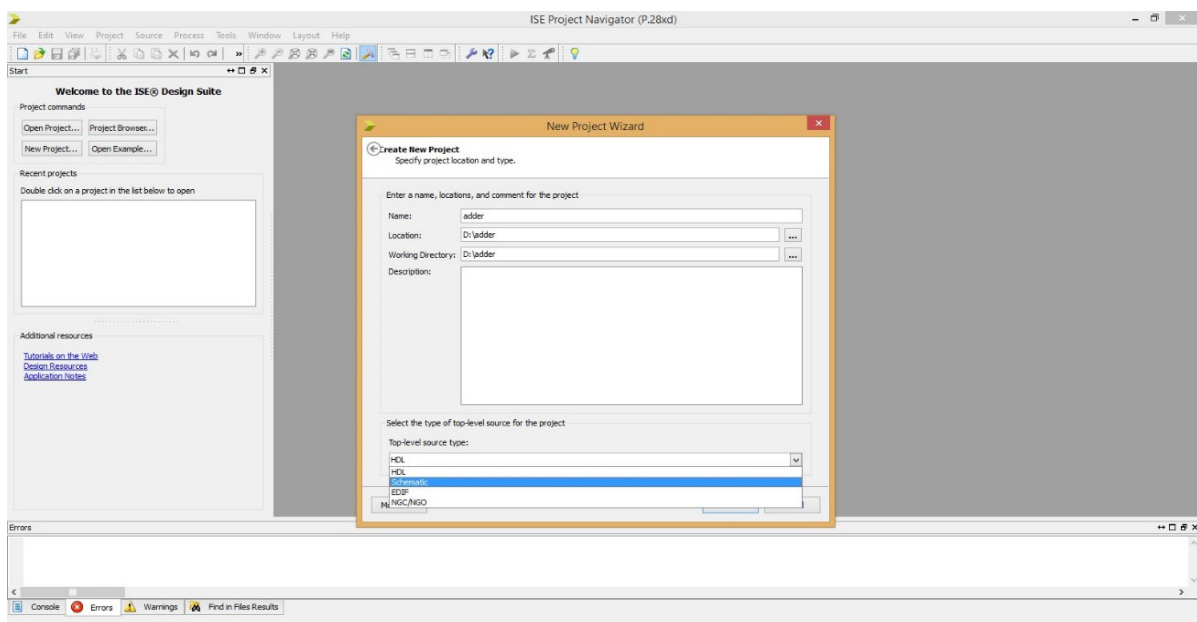
1. Open the Xilinx ISE Project Navigator by selecting it from the start menu or from the desktop icon. Xilinx ISE working place looks like below.



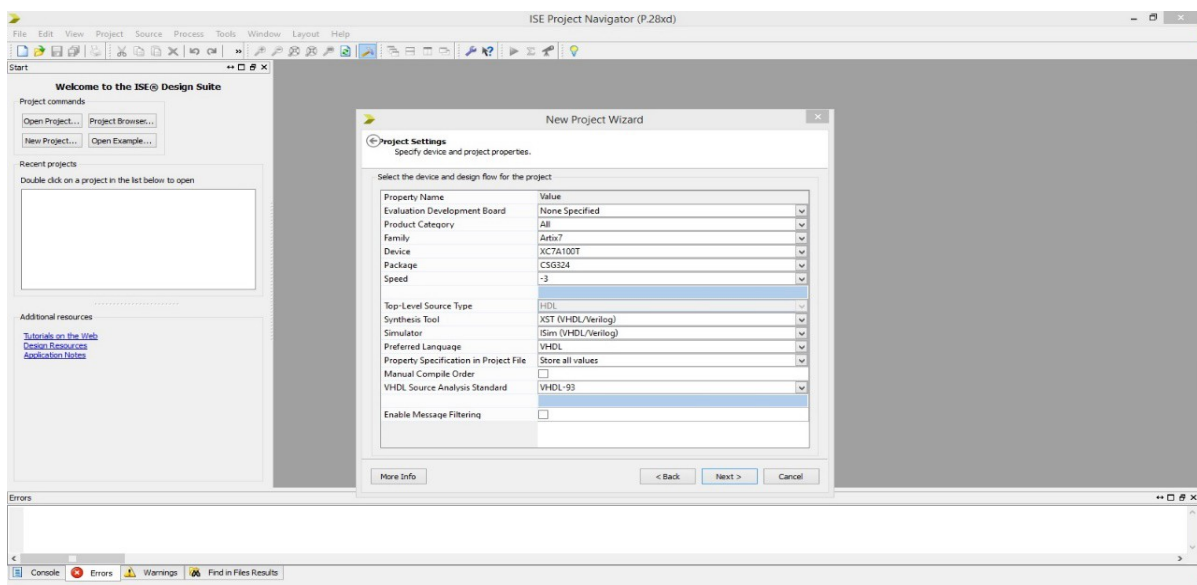
2.

- Within the project navigator, select from the File Menu, New Project like the figure above. Name the project “adder”.
- To put the project in a different location, either directly type the file path into Project Location or search by clicking the Three-dotted Browse button.
- The location of the project can always be seen on the top of the Project Navigator screen. Project name and location can't contain any empty spaces. It is also important for you for the future projects.
- In the “top level source type” menu you should select schematic for the first part of experiment.

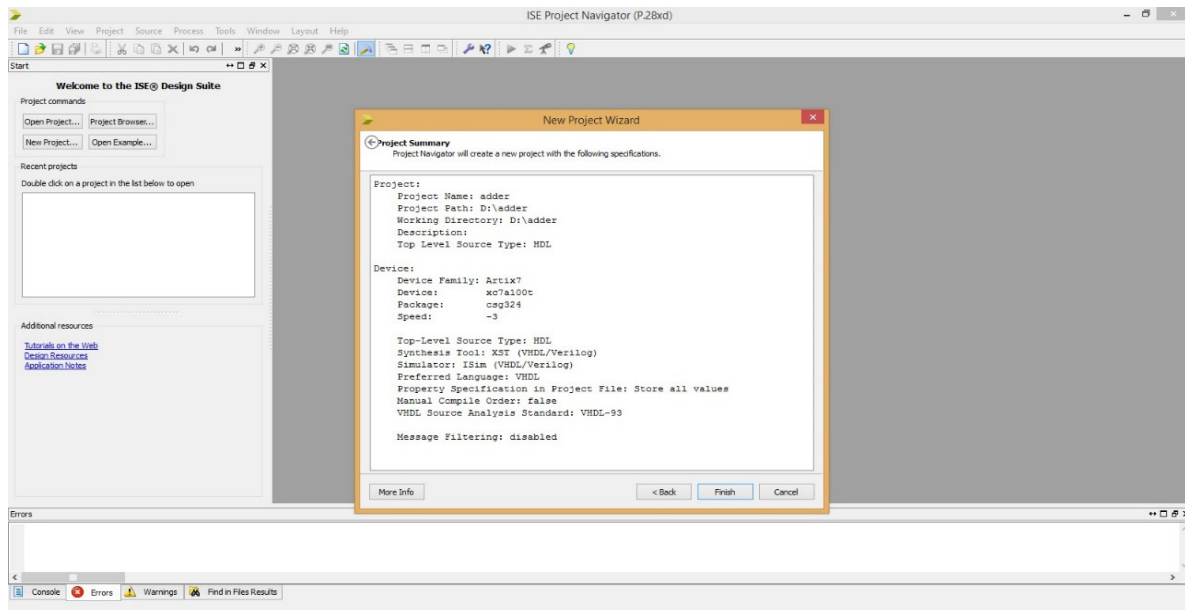
In the second part, we will build a new project and we use VHDL language, so in second part you should select “HDL” for the top level source type. Click next and it will show the device properties window.



3. Set the device properties exactly as shown in the following picture and continue to select next until the project is created.

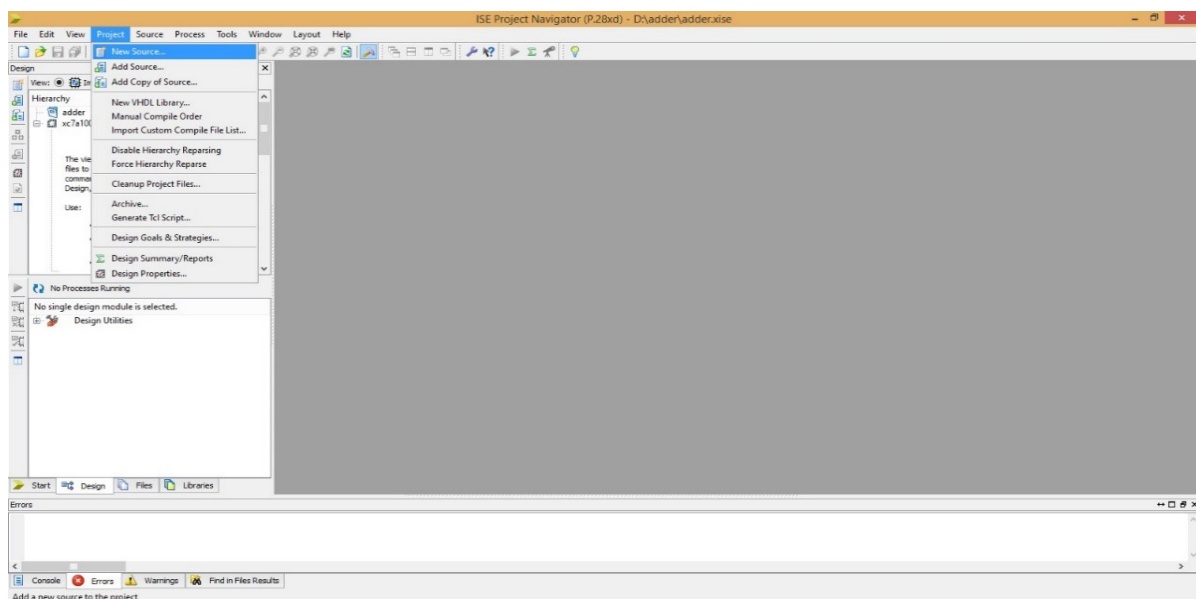


4. After device properties set correctly you should skip screen with using next and Finish button. Following screen show the summary screen. We will add files to the project after we build it.

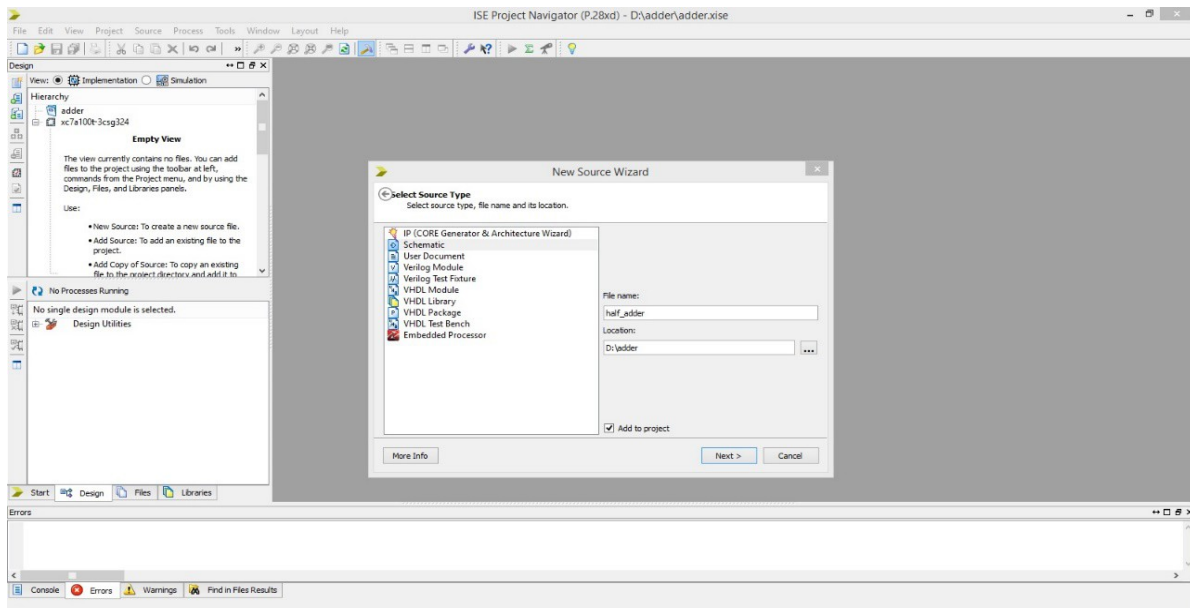


5. The project will be a blank project. For this project, there are two source files. In schematic design you will make design like shown below. In second part you will add copy of source files into your project.

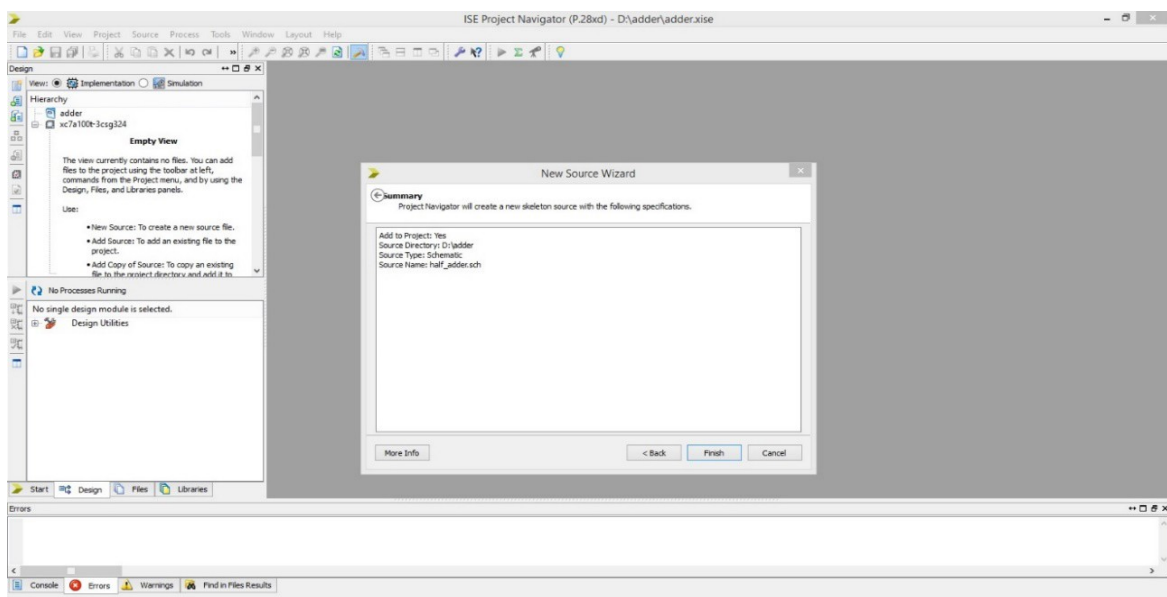
6. First of all, you must add new source to your project by using “Project” menu or by right clicking on project name.



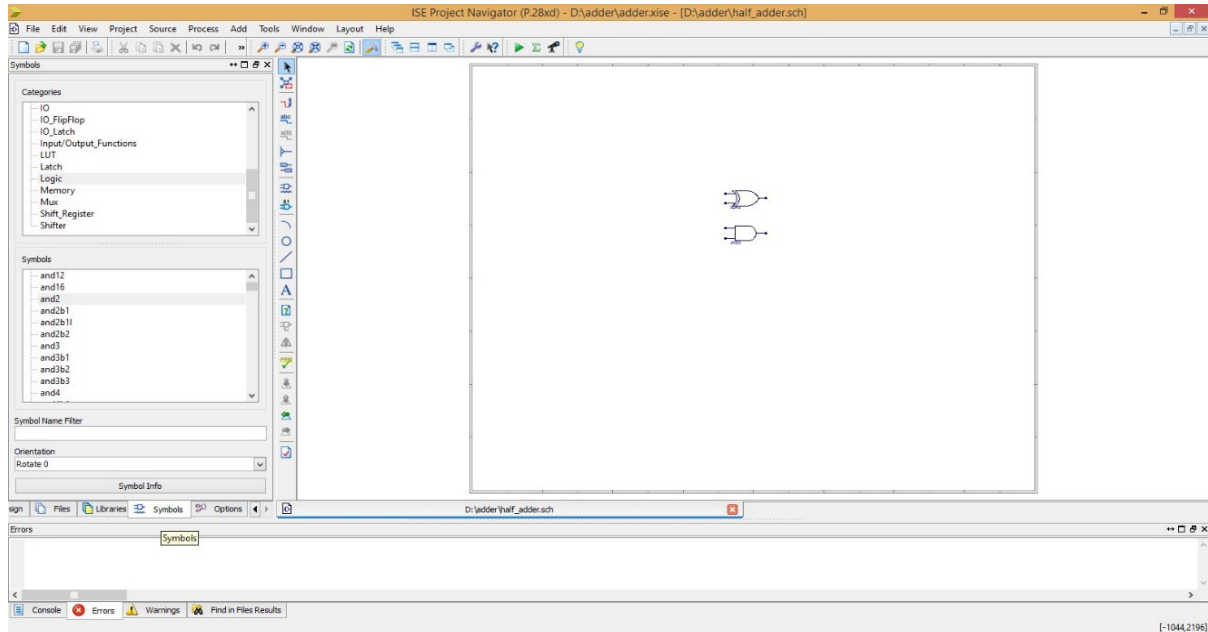
7. You should select “Schematic” on left panel. Name the file “half_adder”. Press next button to go summary screen. You can skip summary screens by clicking finish. Summary screens only show where you are.



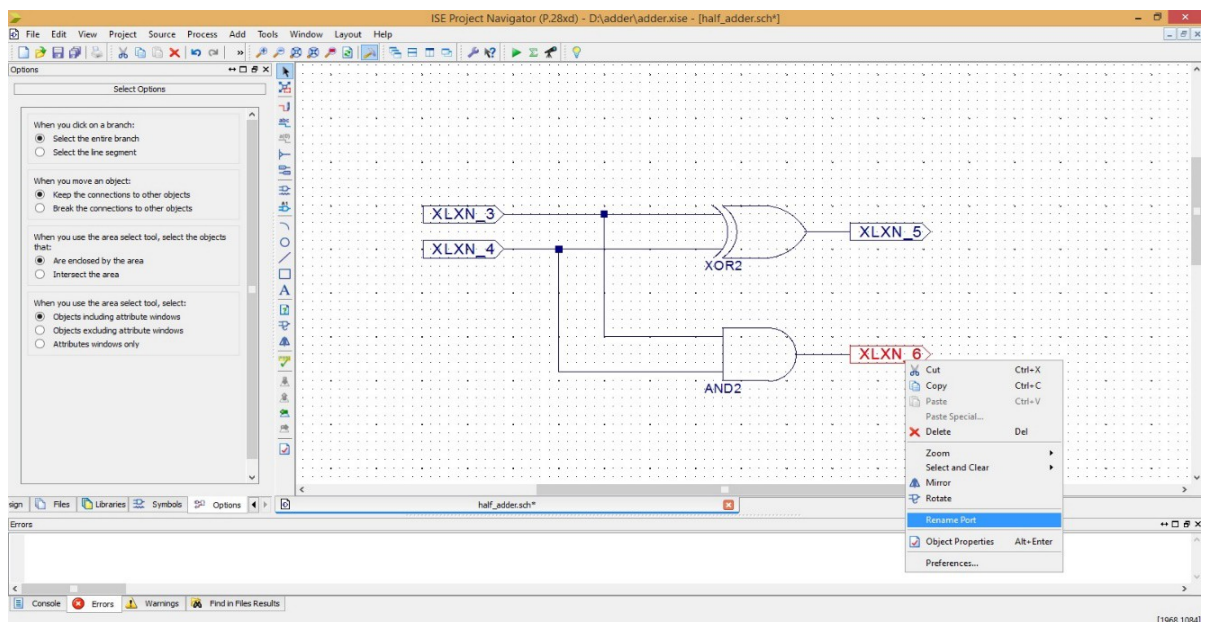
8. Schematic working place must be opened like the following figure and “Creating Schematic completed successfully” message seen on transcript screen.



9. You can use “Symbols” tab on the left-hand side. In this tab you can add some logic element into your workspace. For half adder circuit you should add “xor2” and “and2” gates into your design. They are in “Logic” categories.

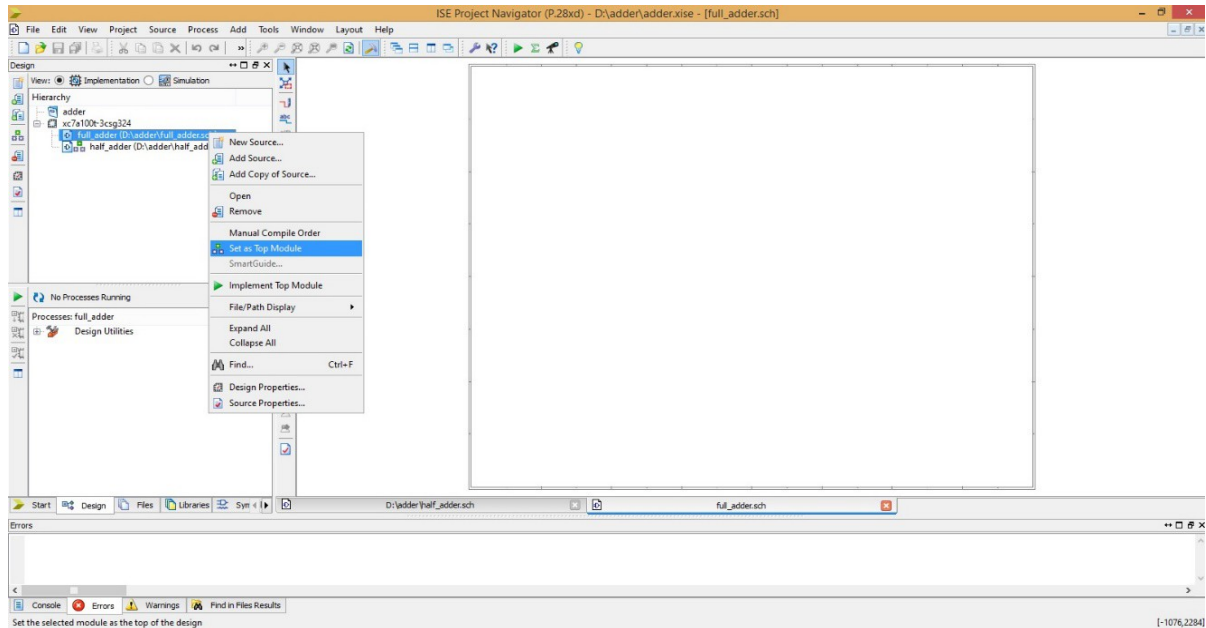


10. You can use zoom buttons to see more clearly. After adding gates to your project you should connect them by using wires and you should add some I/O markers. You can use “Add wire” button or “Ctrl+W” shortcut to go adding wire mode. After that, you should add I/O marker to show the compiler input and output ports. You can use “Add I/O marker” button or “Ctrl+G” shortcut. After adding them your circuit looks like following figure. You can also rename the I/O markers according to your input and outputs not to be confused in the following designs.

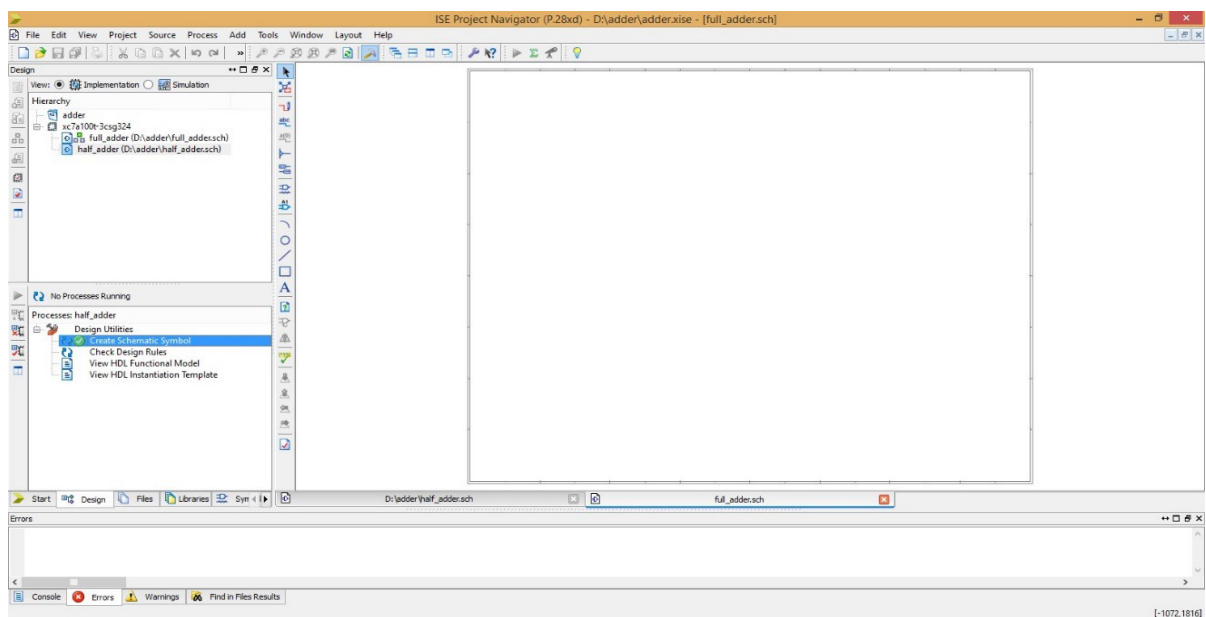


11. Add one more schematic module to your project and name it “full_adder”. You should change “symbol” tab to “design” tab.

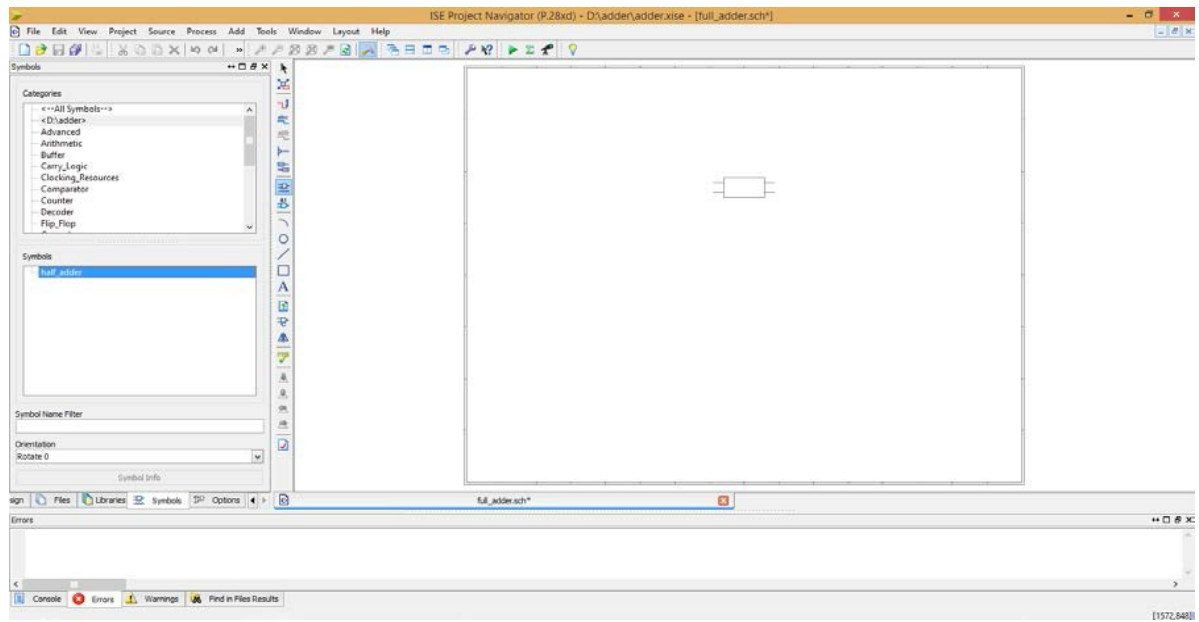
12. Right click “full_adder” and select “Set as a top module” like below.



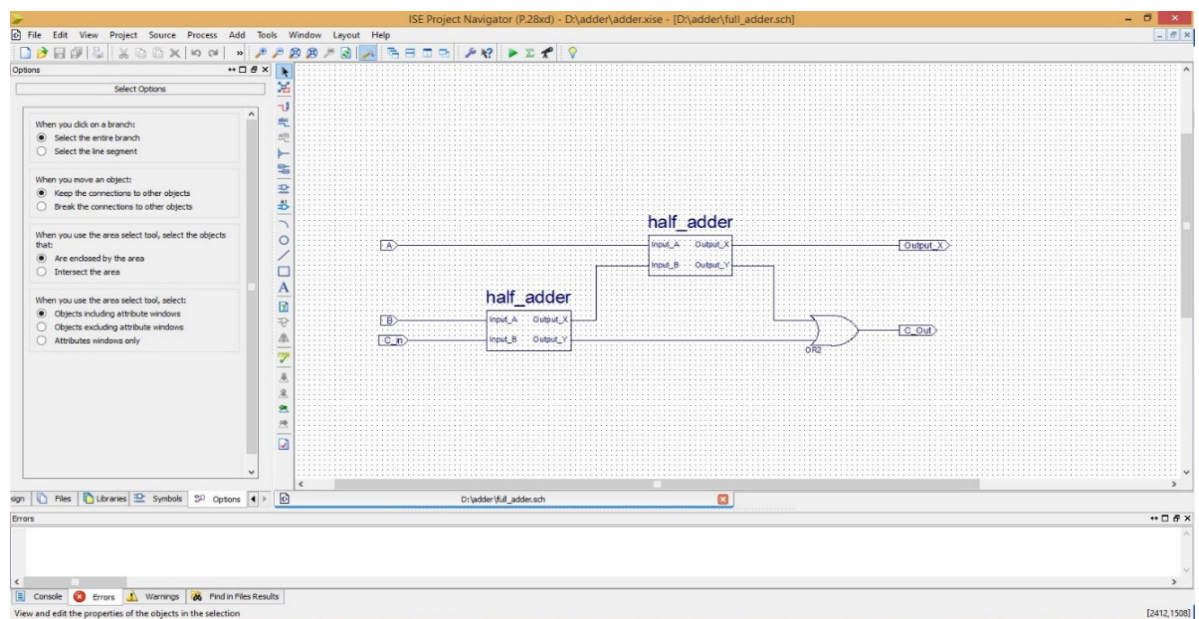
13. Select “half_adder” on source screen and extend the “Design Utilities” below, then double click “Create schematic symbol”. So we can use half adder design on full adder design.



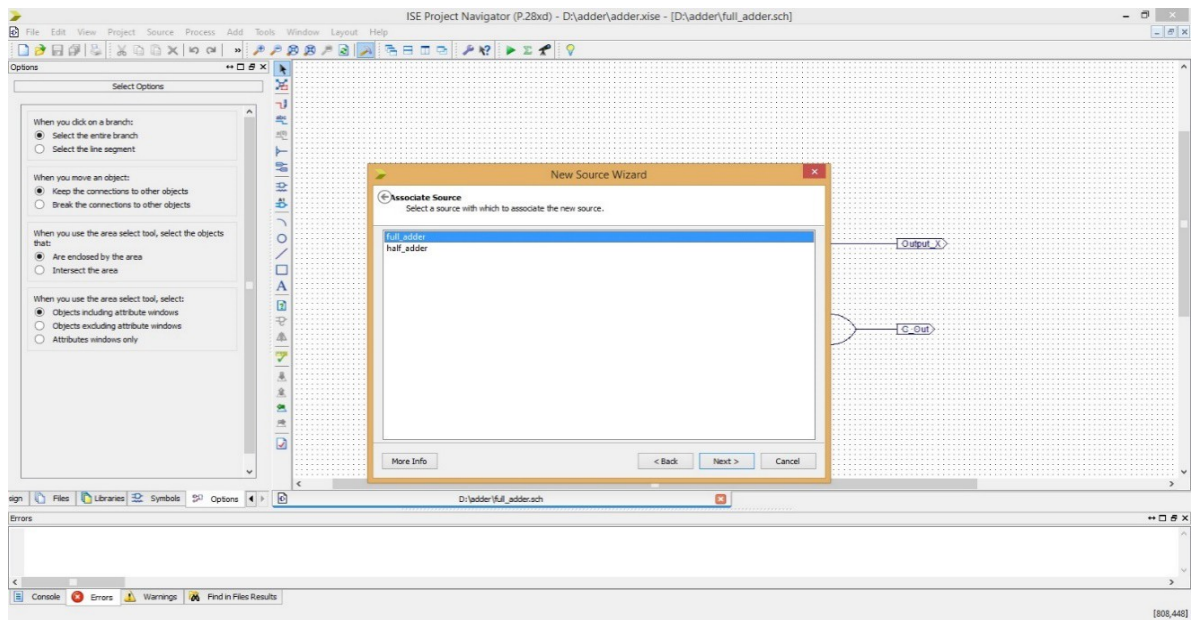
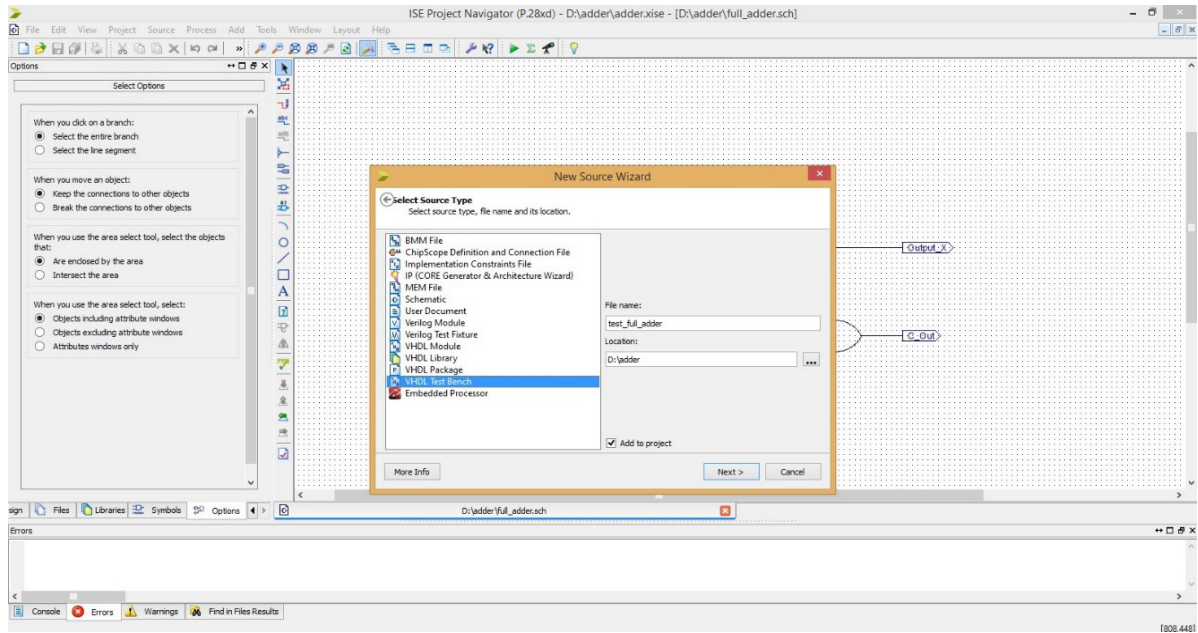
14. In “symbols” tab you can see created objects in (“project directory”) in “categories” tab.



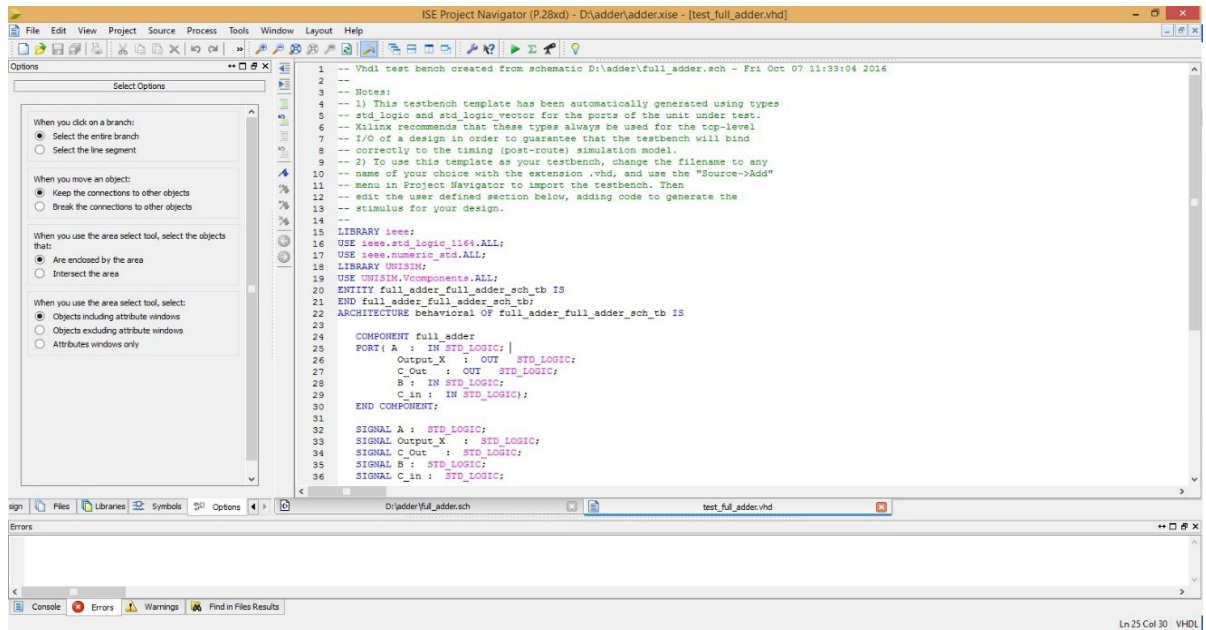
15. Build full adder like the image below



16. After complete the design we must test it by using simulator to understand and confirm its behavior. Add new “VHDL Test Bench” and name it “test_full_adder” like below. Next screen will ask you which component to be tested. Click next and finish button to complete.



17. The test bench should look like the image below.



18. You should enter all input combinations and see the results to verify the design. After doing that to start simulation you should save it and go to the design tab. Then On the top of the tab you should see View section and click on the simulation checkbox. Double click on the “Simulate Behavioral Model” and the simulation screen will appear.

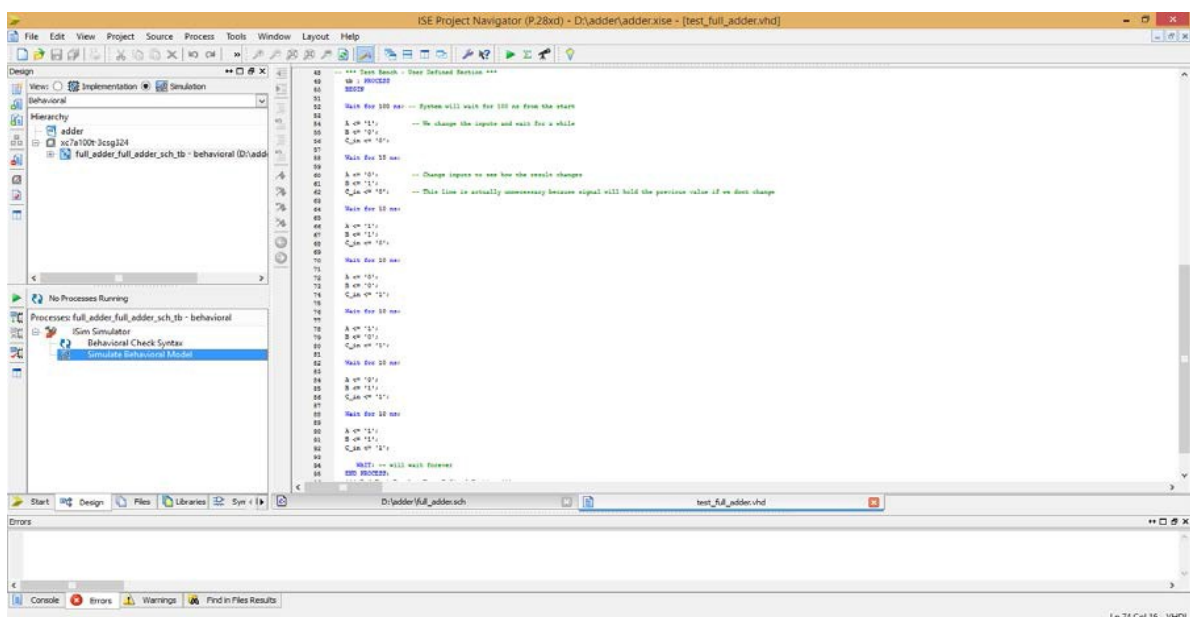
Example input set;

A<= '1';

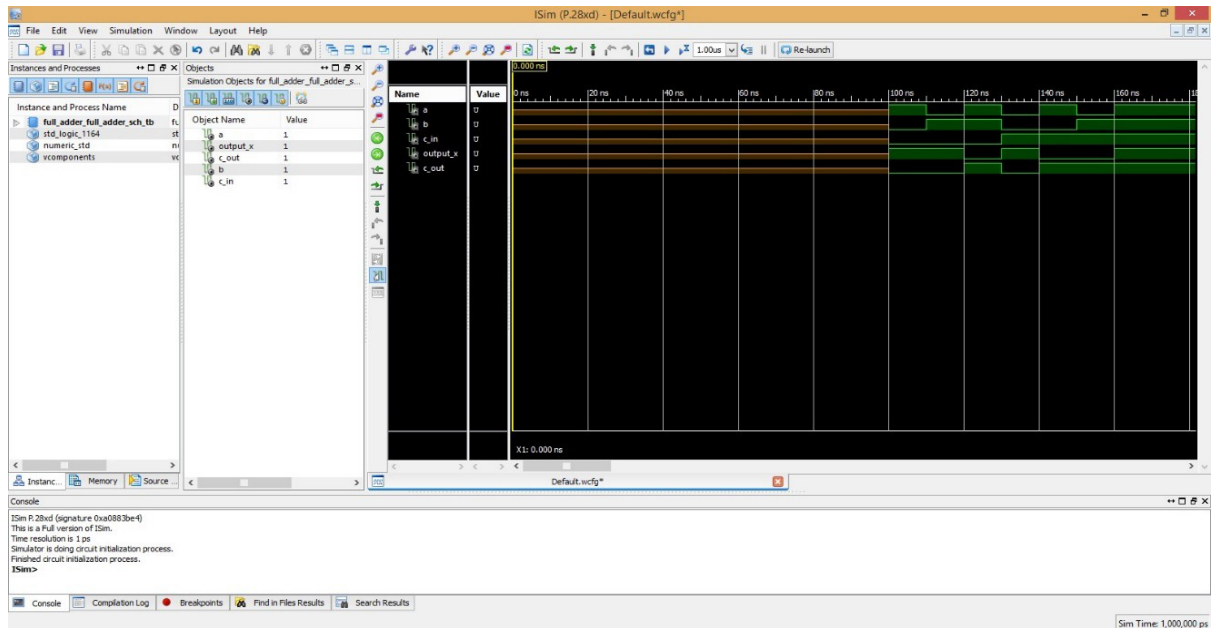
B<= '0';

C_in<= '1'; Wait

for 10 ns;



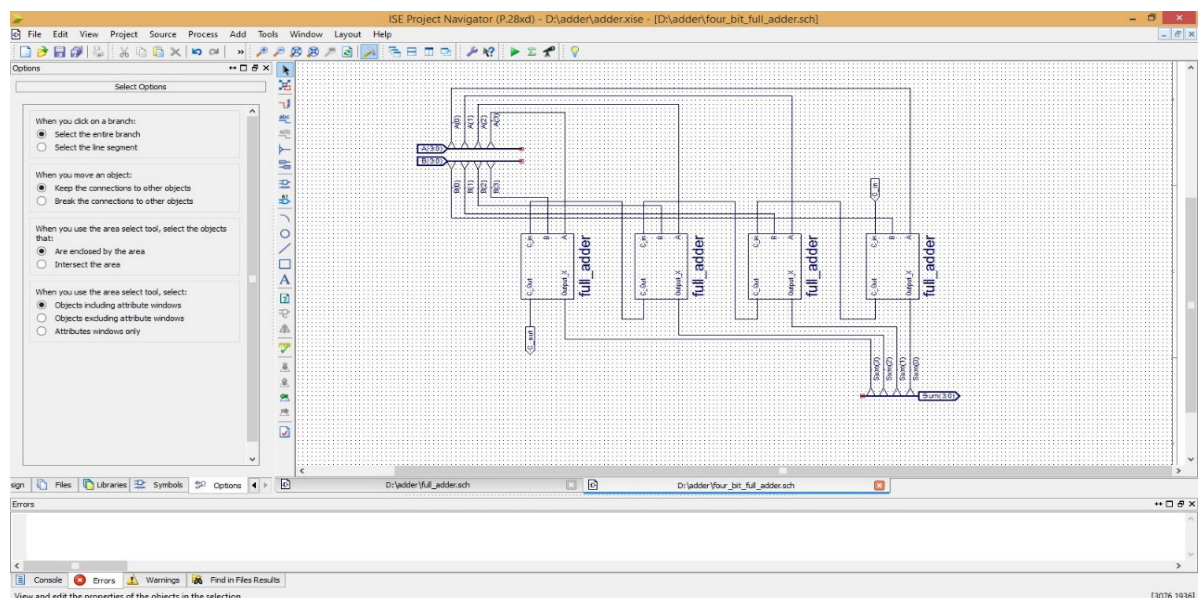
19. You can zoom in to see inputs and outputs clearly



20. Like full adder design we can build 4 bit adder with combining full adders. To do that you should create new schematic file and then you should make it top module by right clicking it and selecting “Set as Top Module”. To use full adder design modularly in 4 bit adder design, you should create schematic symbol of full adder.

21.

- When you try to connect wires, like the picture below, you will see that you cannot connect wires with buses. Bus tap is used to separate and combine buses.
- After creating a single wire, you should name it like “Sum(3:0)” to build 4 bit bus. ISE change wire to bus when you named it like a bus.
- After that, you should add “bus tap” to access wires of bus.
 1. First, click on the bus tap icon
 2. Click on the bus that you want to add bus tap
 3. Rotate the bus tap as needed.
 4. Finally you can drop the bus tap near the bus and it will attach to the bus. (Not directly on the bus, click on a blank space near the bus)



22. You should test 4 bit adder. Test procedure is the same (Don't forget to convert the design to symbol). You should obtain results like below when you prepare test to add (5 and 3).

