ESKİŞEHİR TECHNICAL 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.

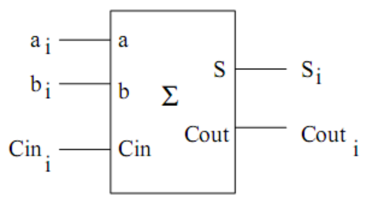


Figure (1) - Block schematic of full adder

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

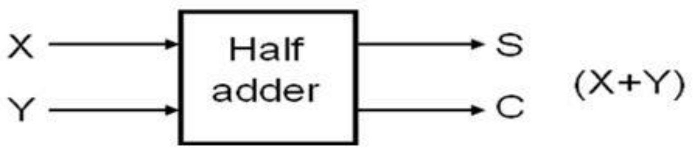


Figure (4) – Half adder block diagram

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.

$$\mathbf{C = X \cdot Y}$$

$$\mathbf{S = X \oplus Y}$$

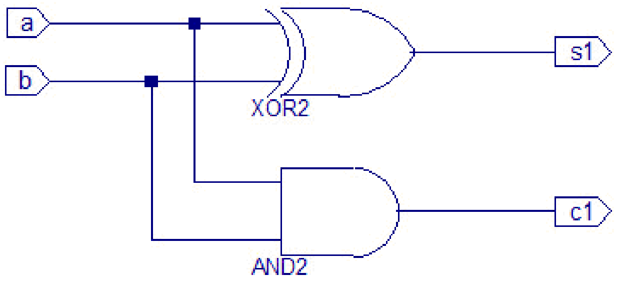


Figure (5) - Gate equivalent of half adder

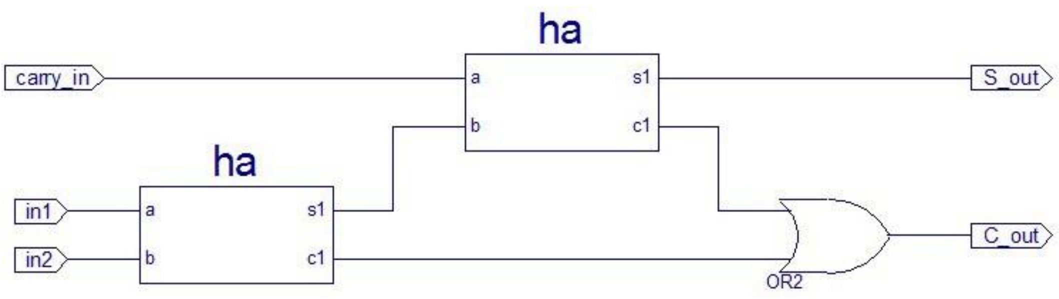


Figure (6) - Full adder circuit

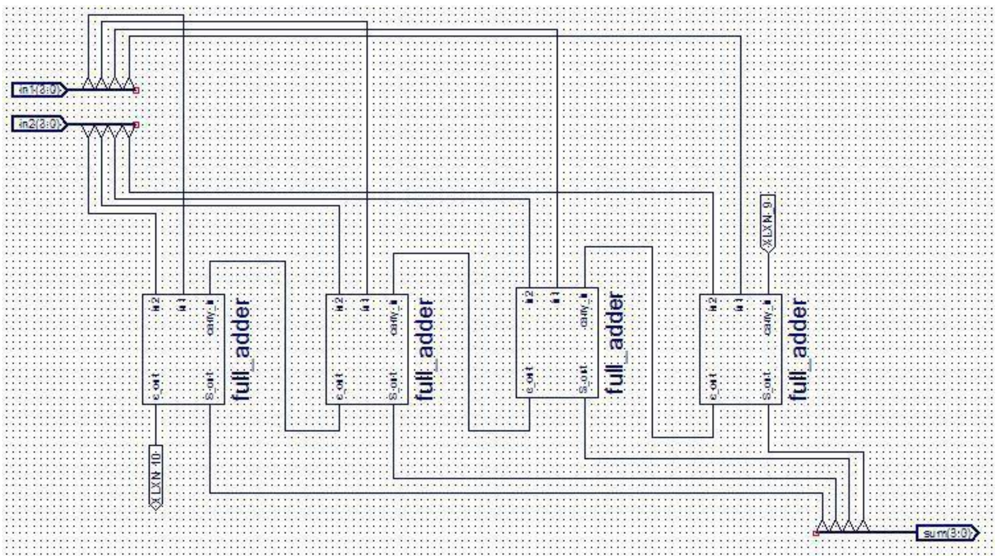


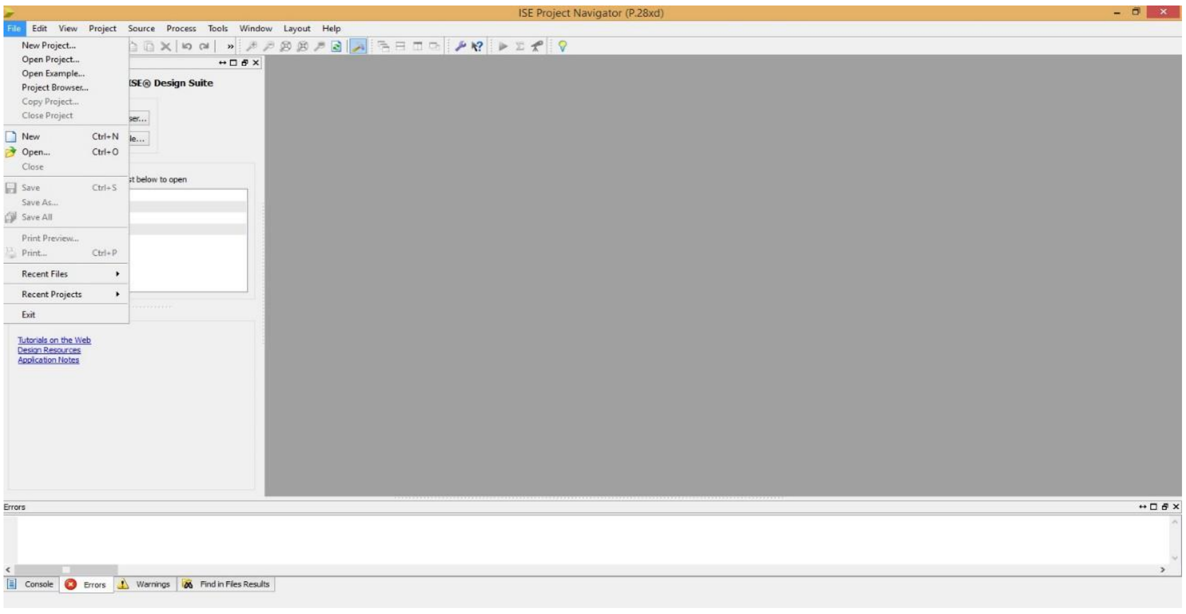
Figure (7) - Four Bit Ad

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.

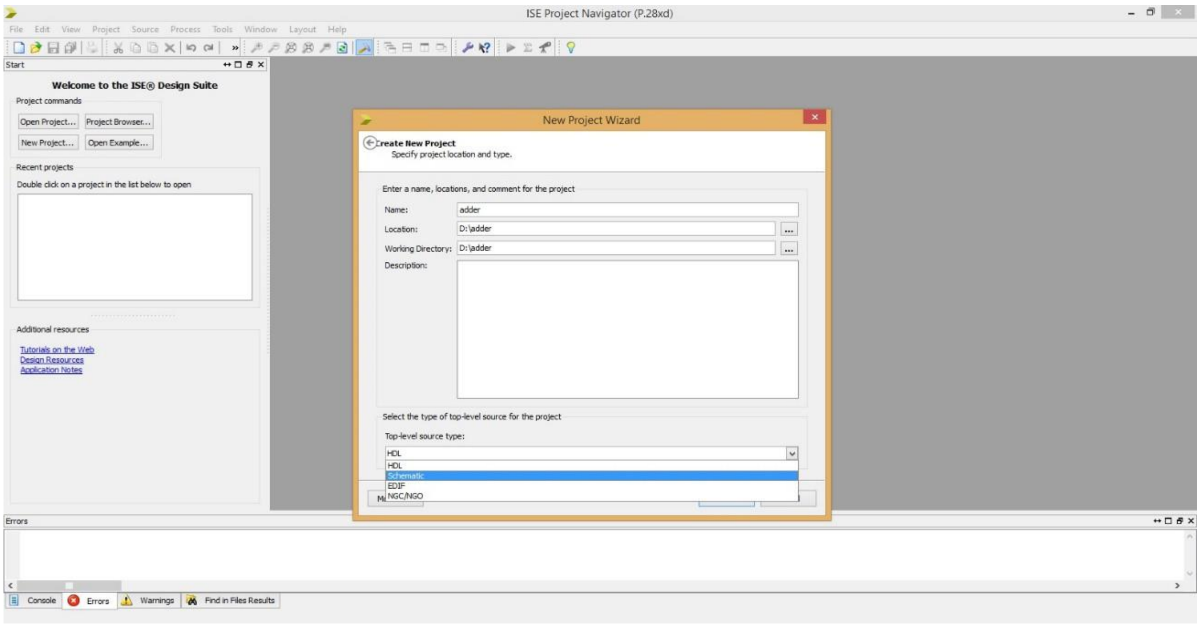
3.1. Schematic Design

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.



New Project Wizard

Create New Project
Specify project location and type.

Enter a name, locations, and comment for the project

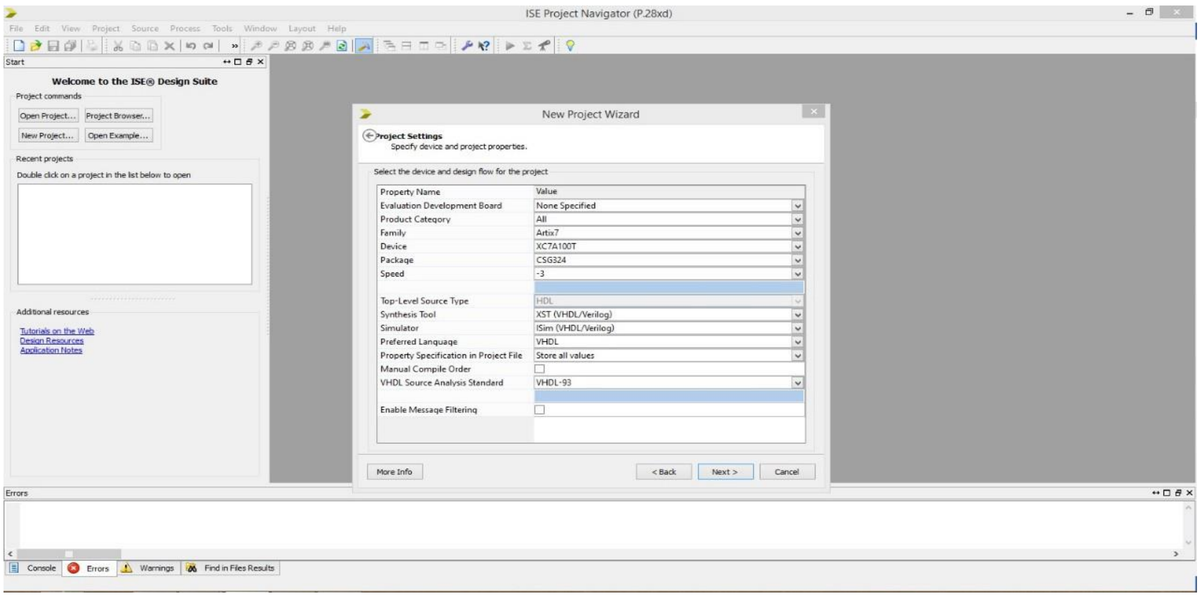
Name:	adder
Location:	D:\adder
Working Directory:	D:\adder
Description:	

Select the type of top-level source for the project

Top-level source type: HDL

HDL
Schematic
EDIF
NGC/NGO

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



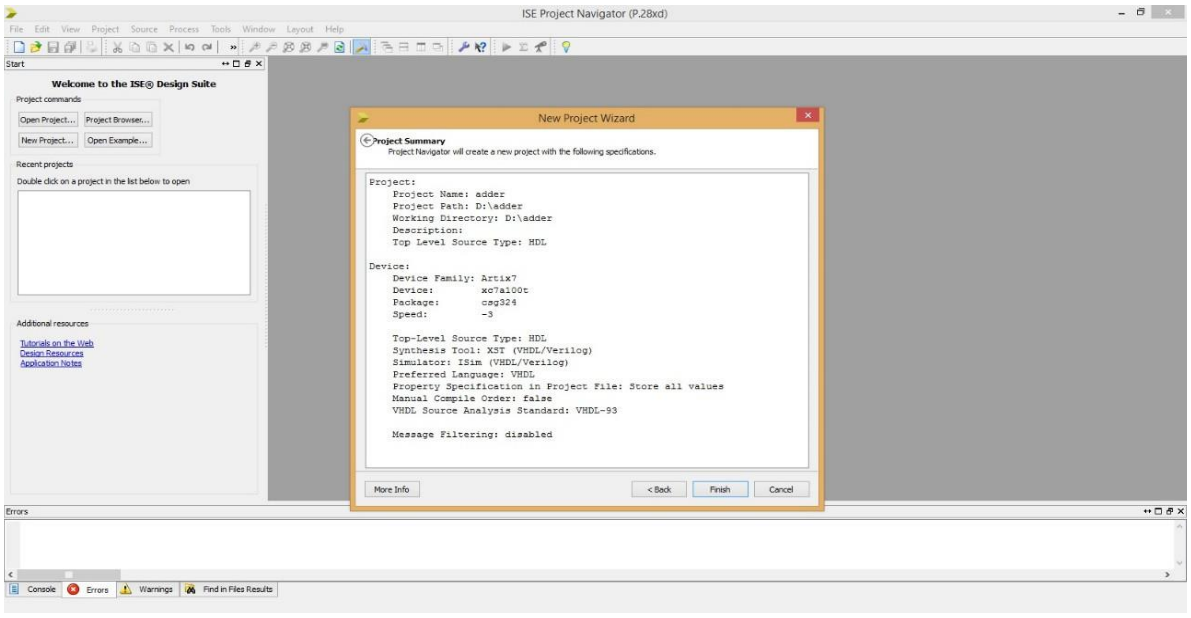
New Project Wizard

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	-3
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	☐
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	☐

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.

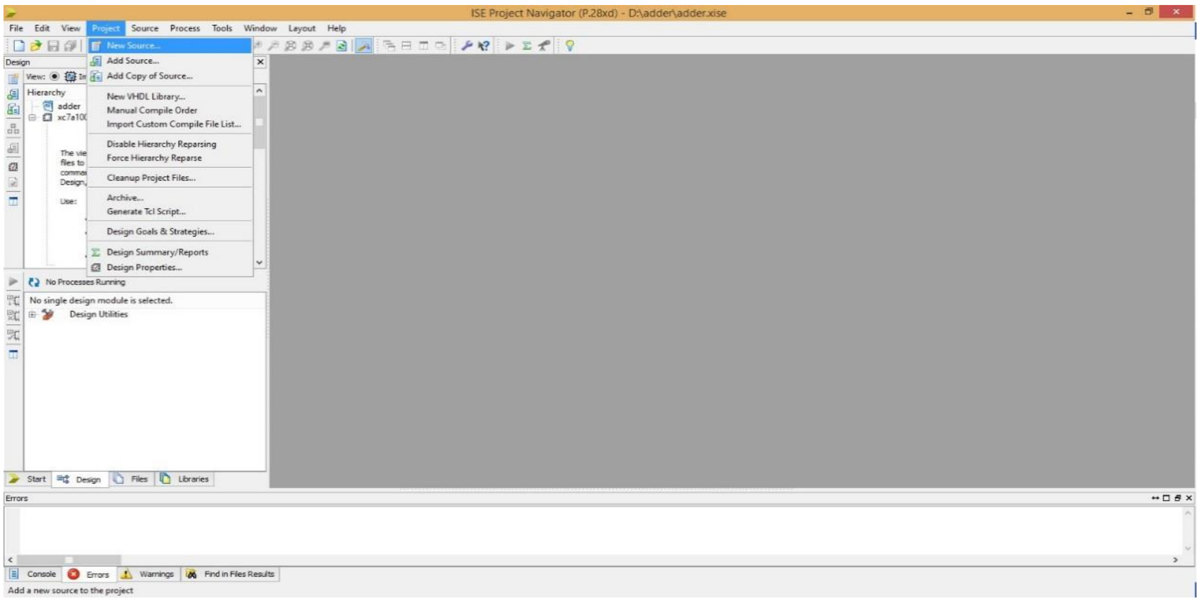

ISE Project Navigator (P.28xd)
File Edit View Project Source Process Tools Window Layout Help
Welcome to the ISE® Design Suite
Open Project...
Project Browser...
New Project...
Open Example...
New Project Wizard
More Info
< Back
Finish
Cancel

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

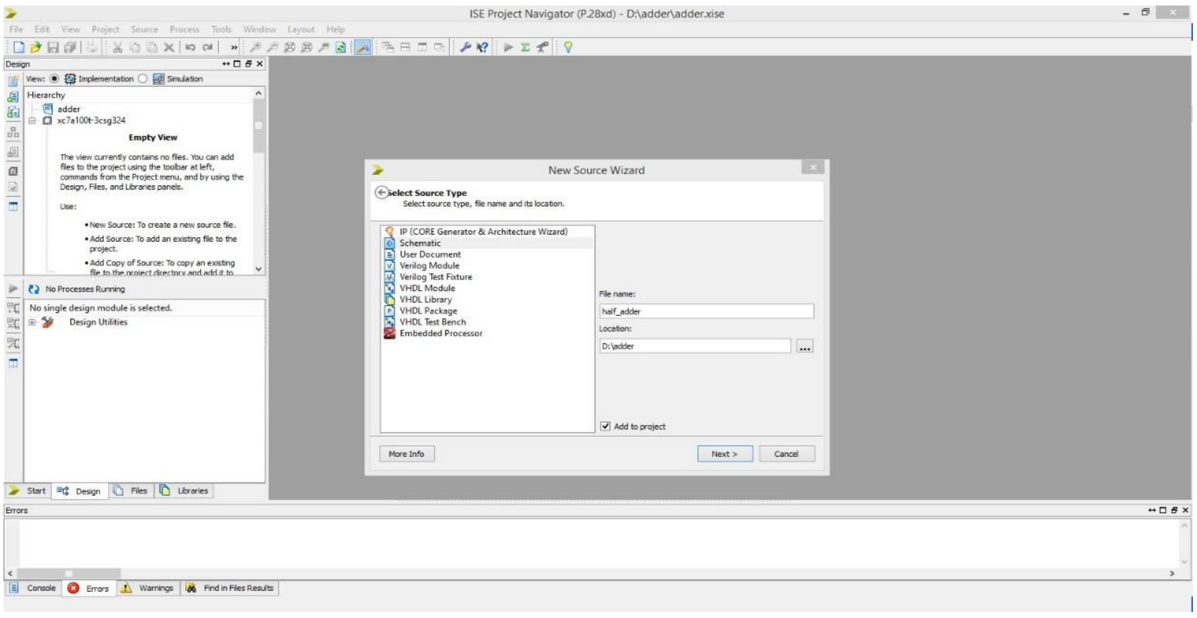
```
Project:
    Project Name: adder
    Project Path: D:\adder
    Working Directory: D:\adder
    Description:
    Top Level Source Type: HDL

Device:
    Device Family: Artix7
    Device:        xc7a100t
    Package:       csg324
    Speed:         -3

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

    Message Filtering: disabled
```

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.

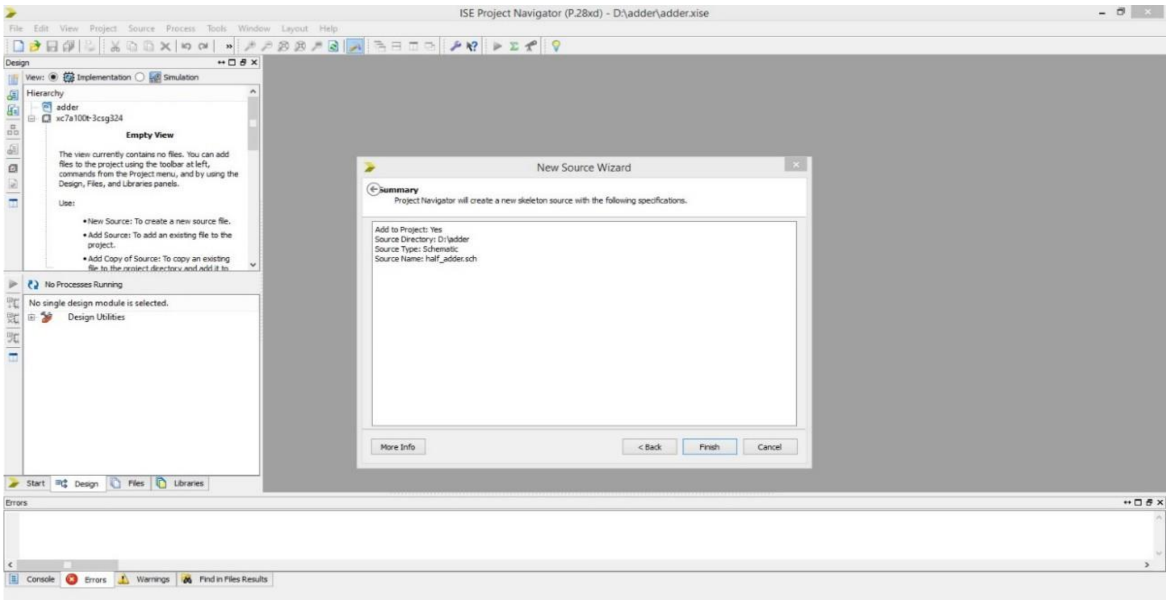
6. First of all, you must add new source to your project by using “Project” menu or by right clicking on project name.


ISE Project Navigator (P.28xd) - D:\adder\adder.xise
File Edit View Project Source Process Tools Window Layout Help
New Source...
Add Source...
Add Copy of Source...
New VHDL Library...
Manual Compile Order
Import Custom Compile File List...
Disable Hierarchy Reparsing
Force Hierarchy Reparse
Cleanup Project Files...
Archive...
Generate Tcl Script...
Design Goals & Strategies...
Design Summary/Reports
Design Properties...
No Processes Running
No single design module is selected.
Design Utilities
Add a new source to the project

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.

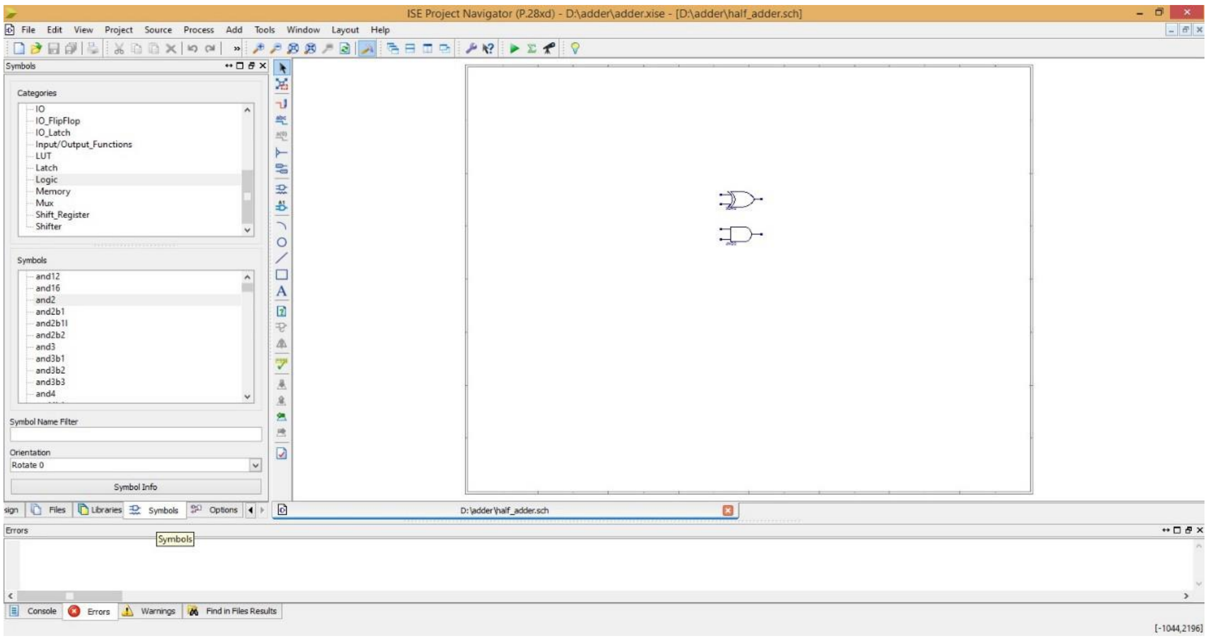


8. Schematic working place must be opened like the following figure and “Creating Schematic completed successfully” message seen on transcript screen.

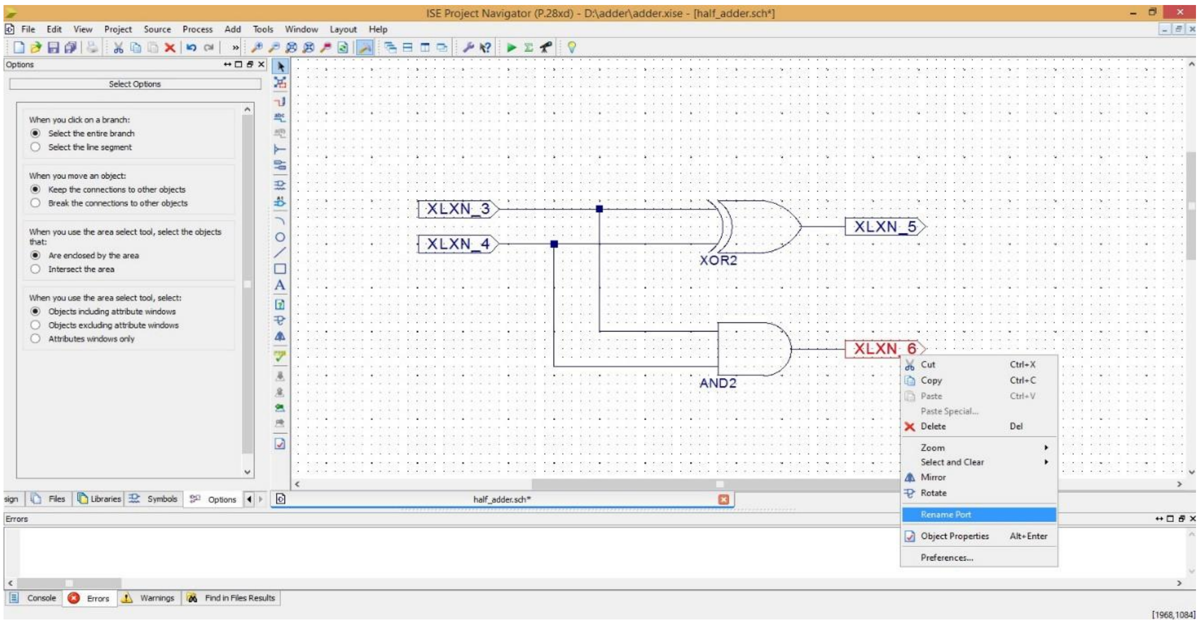

ISE Project Navigator (P.28xd) - D:\adder\adder.xise
New Source Wizard
Summary
Project Navigator will create a new skeleton source with the following specifications.
More Info
< Back
Finish
Cancel

Add to Project: Yes
Source Directory: D:\adder
Source Type: Schematic
Source Name: half_adder.sch

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.



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

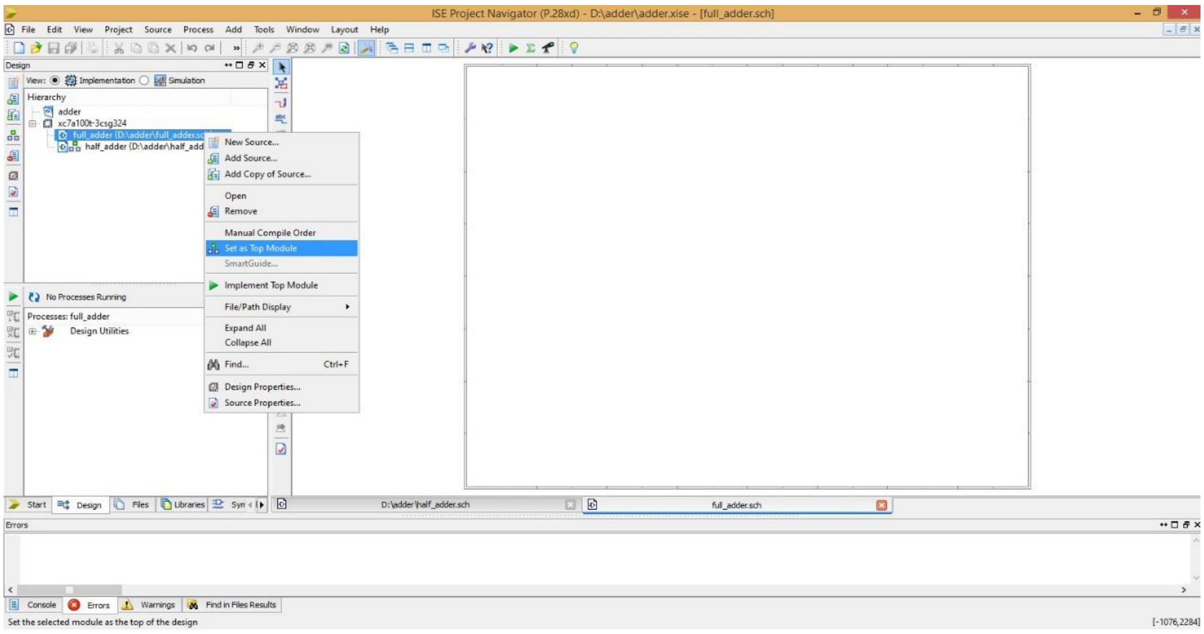
When you move an object:
- [✓ Keep the connections to other objects]
- Break the connections to other objects

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

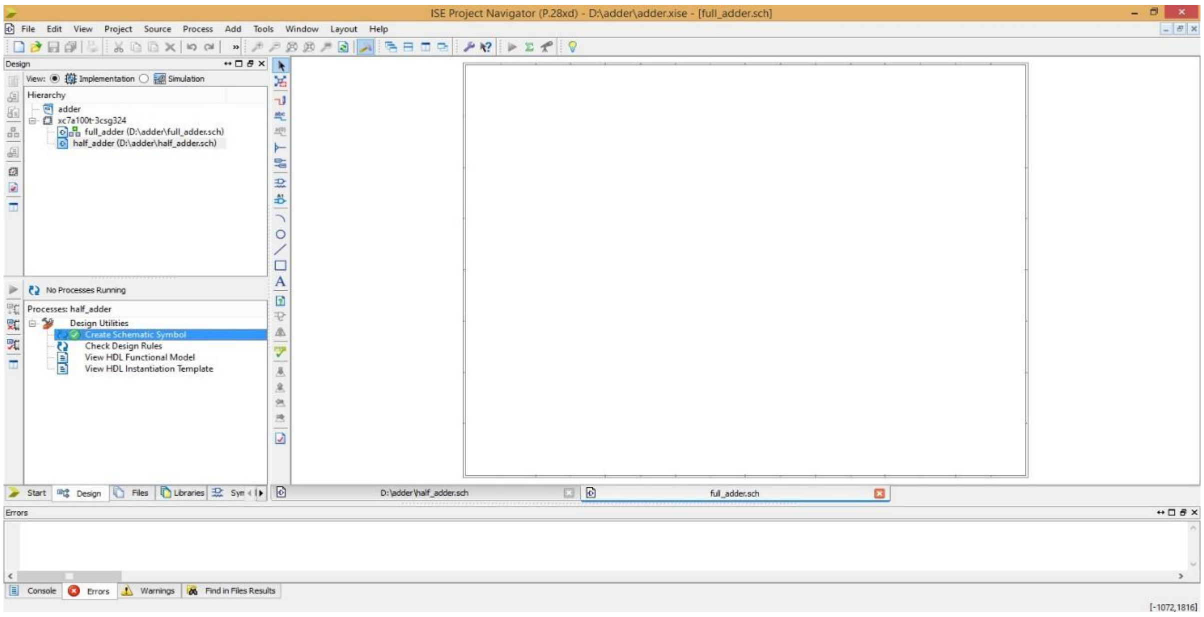
11. Add one more schematic module to your project and name it “full_adder”. You should change “symbol” tab to “design”tab.

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

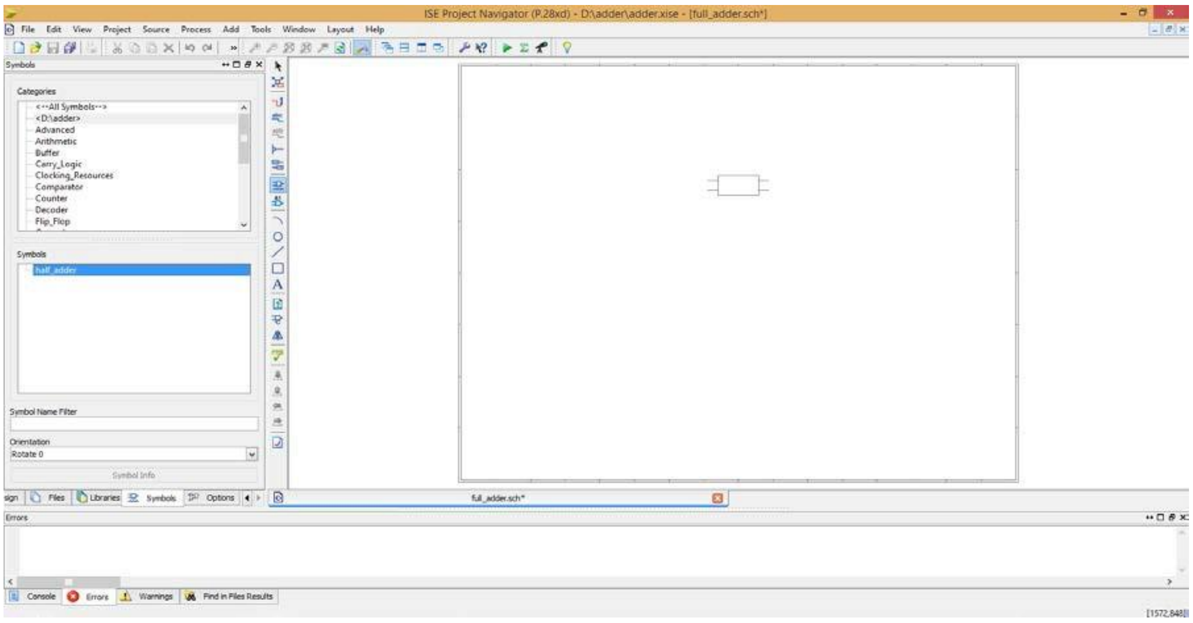


Set the selected module as the top of the design

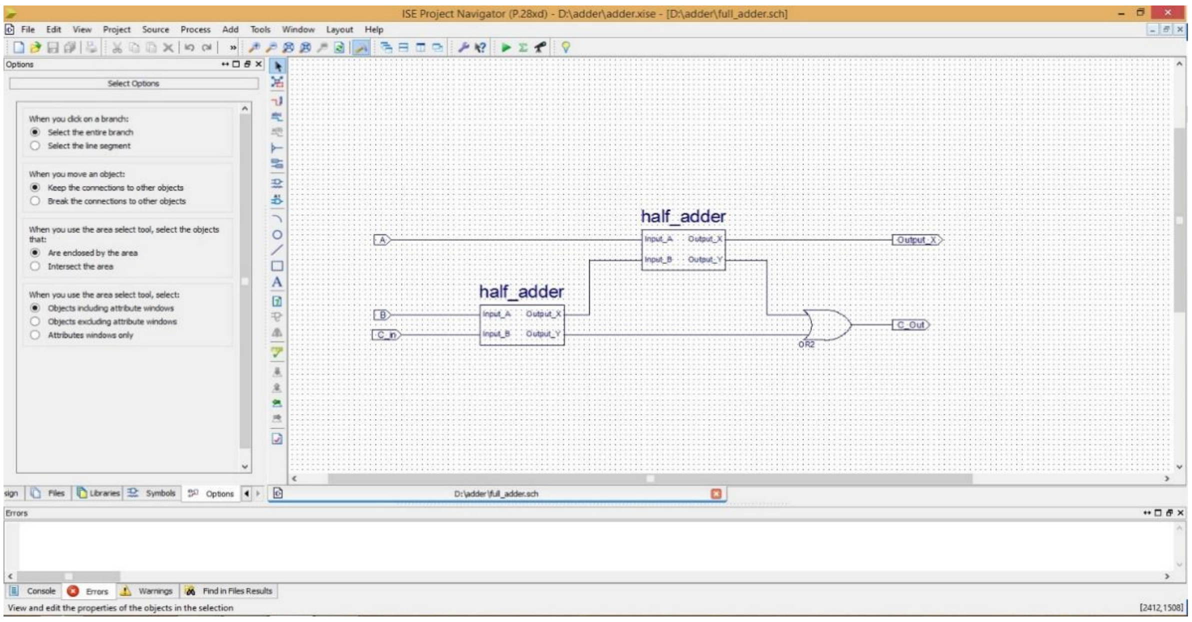
13. Sellect “half_adder” on source screen and extend the “Design Utilities” below, then double click “Create schematic symbol”. So we can use half adder design on full adder design.


ISE Project Navigator (P.28xd) - D:\adder\adder.xise - [full_adder.sch]
File Edit View Project Source Process Add Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
adder
xc7a100t-3csg324
full_adder (D:\adder\full_adder.sch)
half_adder (D:\adder\half_adder.sch)
No Processes Running
Processes: half_adder
Design Utilities
Create Schematic Symbol
Check Design Rules
View HDL Functional Model
View HDL Instantiation Template
Start Design Files Libraries Sym
D:\adder\half_adder.sch
full_adder.sch
Errors
Console Errors Warnings Find in Files Results
[-1072,1816]

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



15. Build full adder like the image below



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

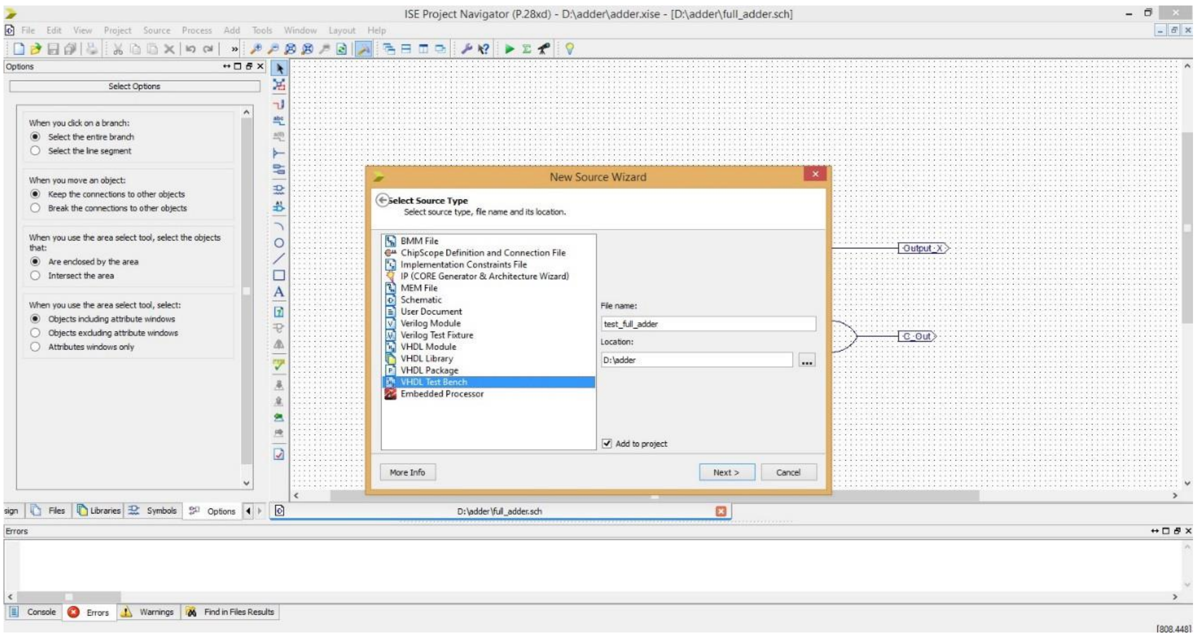
When you move an object:
[✓ Keep the connections to other objects] Break the connections to other objects

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

View and edit the properties of the objects in the selection

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.

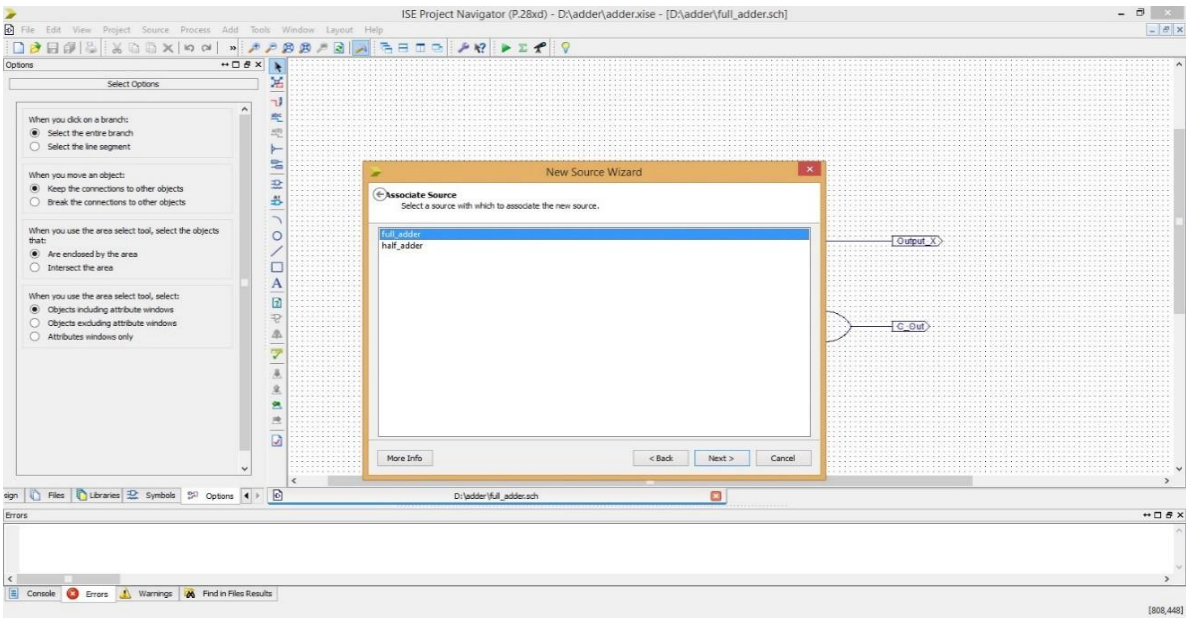

ISE Project Navigator (P.28xd) - D:\adder\adder.xise - [D:\adder\full_adder.sch]

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

File name: test_full_adder
Location: D:\adder
☑ Add to project

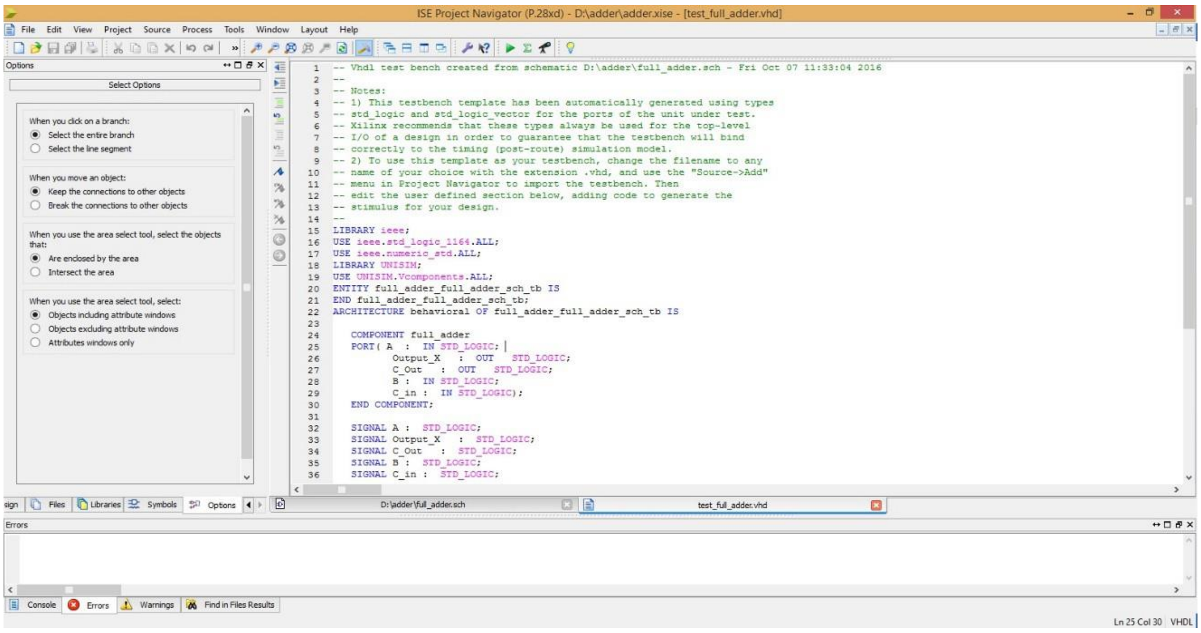

ISE Project Navigator (P.28xd) - D:\adder\adder.xise - [D:\adder\full_adder.sch]

New Source Wizard

Associate Source
Select a source with which to associate the new source.

full_adder
half_adder

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



```
-- Vhdl test bench created from schematic D:\adder\full_adder.sch - Fri Oct 07 11:33:04 2016
--
-- Notes:
-- 1) This testbench template has been automatically generated using types
-- std_logic and std_logic_vector for the ports of the unit under test.
-- Xilinx recommends that these types always be used for the top-level
-- I/O of a design in order to guarantee that the testbench will bind
-- correctly to the timing (post-route) simulation model.
-- 2) To use this template as your testbench, change the filename to any
-- name of your choice with the extension .vhd, and use the "Source->Add"
-- menu in Project Navigator to import the testbench. Then
-- edit the user defined section below, adding code to generate the
-- stimulus for your design.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY full_adder_full_adder_sch_tb IS
END full_adder_full_adder_sch_tb;
ARCHITECTURE behavioral OF full_adder_full_adder_sch_tb IS

   COMPONENT full_adder
   PORT( A  :  IN STD_LOGIC;
         Output_X   :  OUT   STD_LOGIC;
         C_Out   :  OUT   STD_LOGIC;
         B :  IN STD_LOGIC;
         C_in :  IN STD_LOGIC);
   END COMPONENT;

   SIGNAL A :  STD_LOGIC;
   SIGNAL Output_X   :  STD_LOGIC;
   SIGNAL C_Out   :  STD_LOGIC;
   SIGNAL B :  STD_LOGIC;
   SIGNAL C_in :  STD_LOGIC;
```

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.

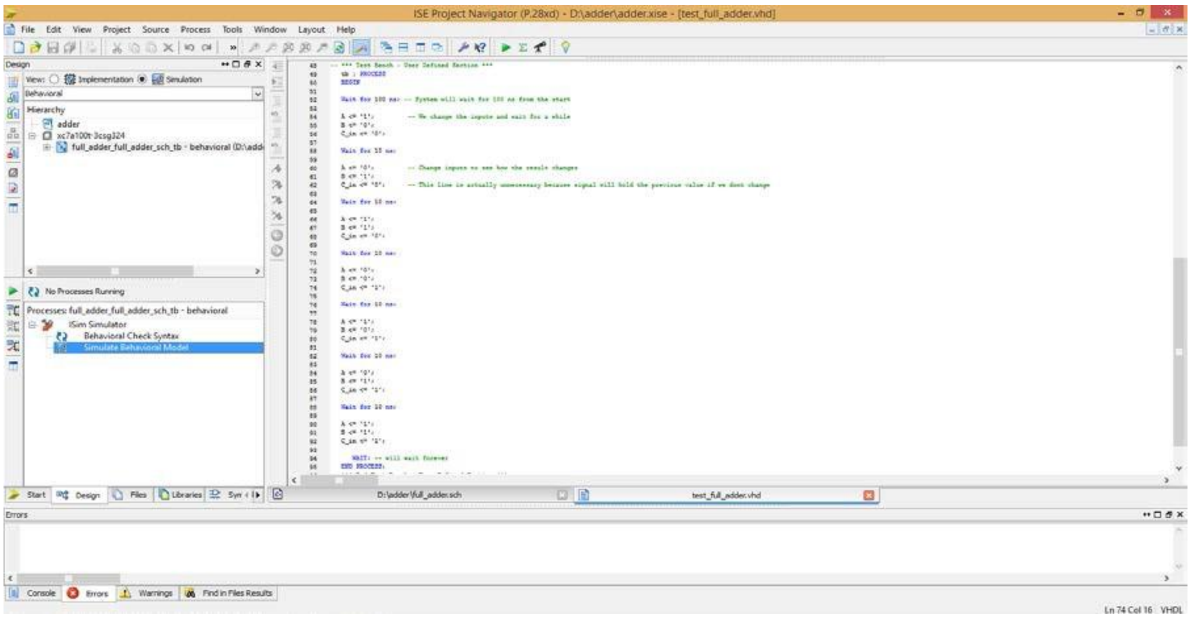
Example input set;

A<= '1';

B<= '0';

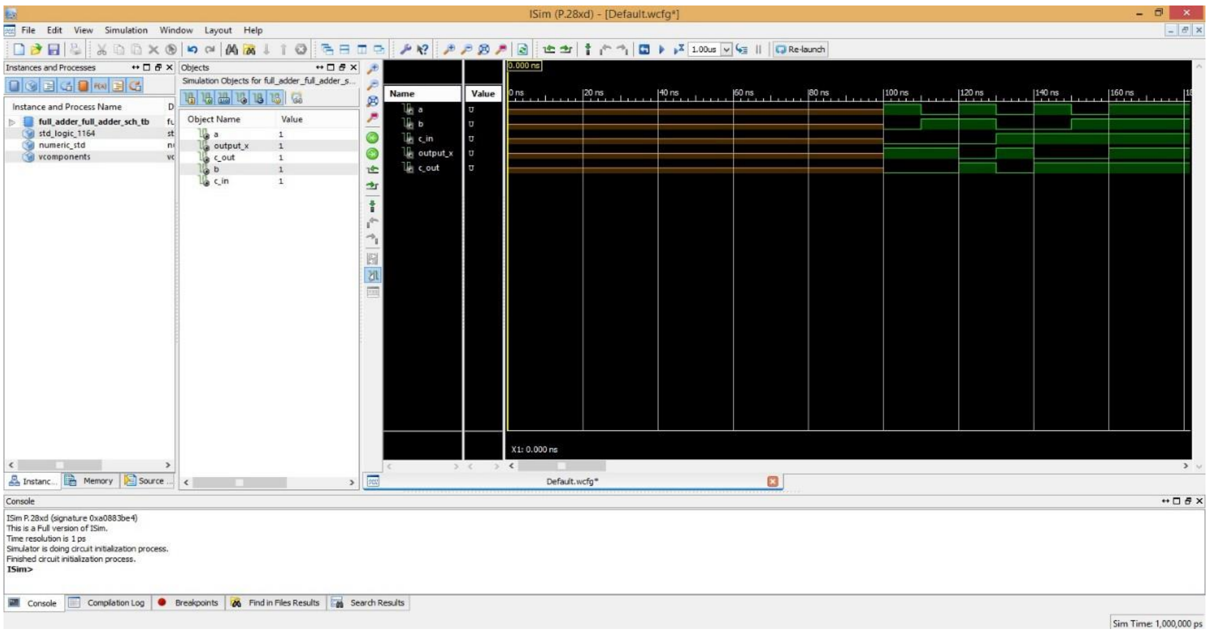
C_in<= '1'; Wait

for 10 ns;



```
-- *** Test Bench - User Defined Section ***
tb : PROCESS
BEGIN

Wait for 100 ns; -- System will wait for 100 ns from the start

A <= '1';        -- We change the inputs and wait for a while
B <= '0';
C_in <= '0';

Wait for 10 ns;

A <= '0';        -- Change inputs to see how the result changes
B <= '1';
C_in <= '0';     -- This line is actually unnecessary because signal will hold the previous value if we dont change

Wait for 10 ns;

A <= '1';
B <= '1';
C_in <= '0';

Wait for 10 ns;

A <= '0';
B <= '0';
C_in <= '1';

Wait for 10 ns;

A <= '1';
B <= '0';
C_in <= '1';

Wait for 10 ns;

A <= '0';
B <= '1';
C_in <= '1';

Wait for 10 ns;

A <= '1';
B <= '1';
C_in <= '1';

   WAIT; -- will wait forever
END PROCESS;
```

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

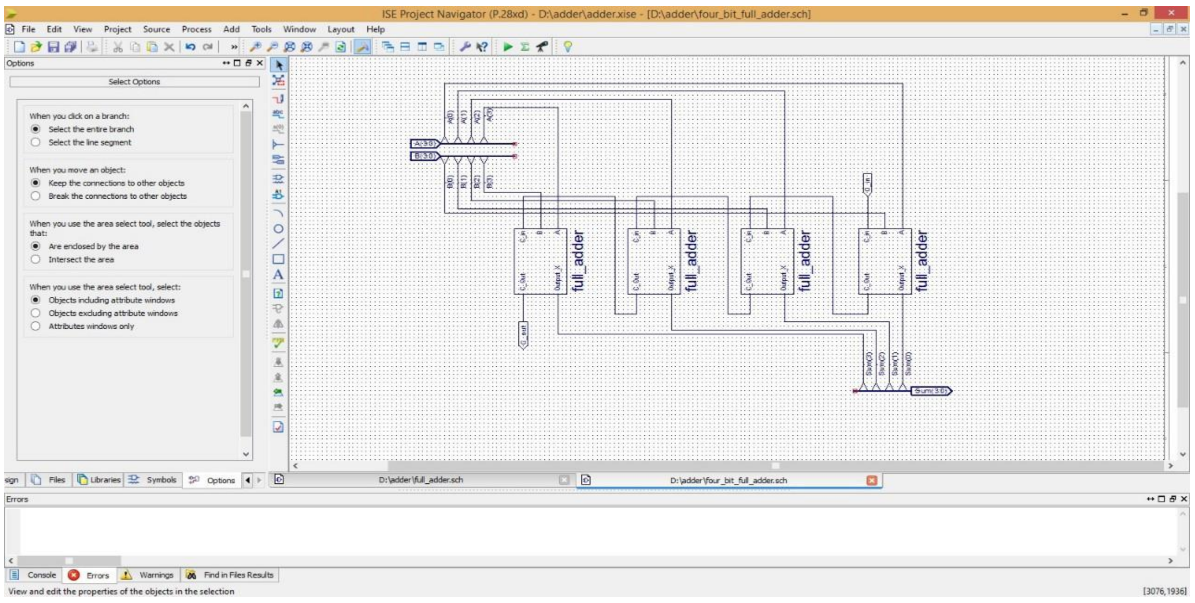


```
ISim P.28xd (signature 0xa0883be4)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

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 tab” 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)


ISE Project Navigator (P.28xd) - D:\adder\adder.xise - [D:\adder\four_bit_full_adder.sch]
File Edit View Project Source Process Add Tools Window Layout Help
Options
Select Options
A(3:0)
B(3:0)
full_adder
full_adder
full_adder
full_adder
Sum(3:0)
Files
Libraries
Symbols
Options
D:\adder\full_adder.sch
D:\adder\four_bit_full_adder.sch
Errors
Console
Errors
Warnings
Find in Files Results
View and edit the properties of the objects in the selection
[3076,1936]

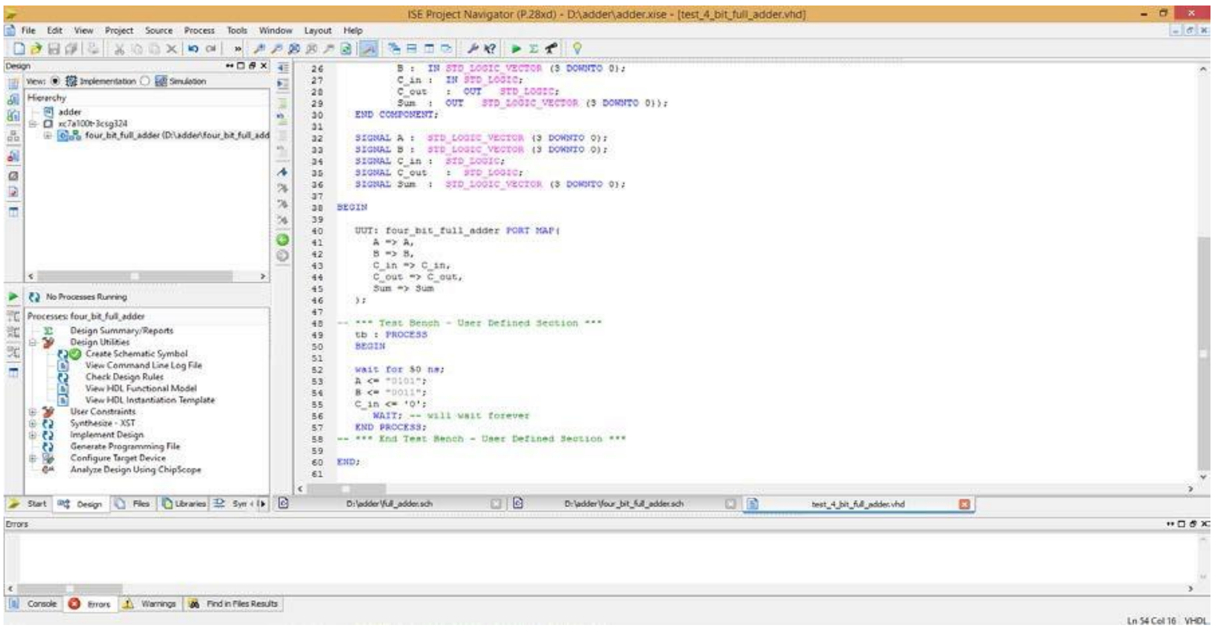
When you click on a branch:
[✓ Select the entire branch]
Select the line segment

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

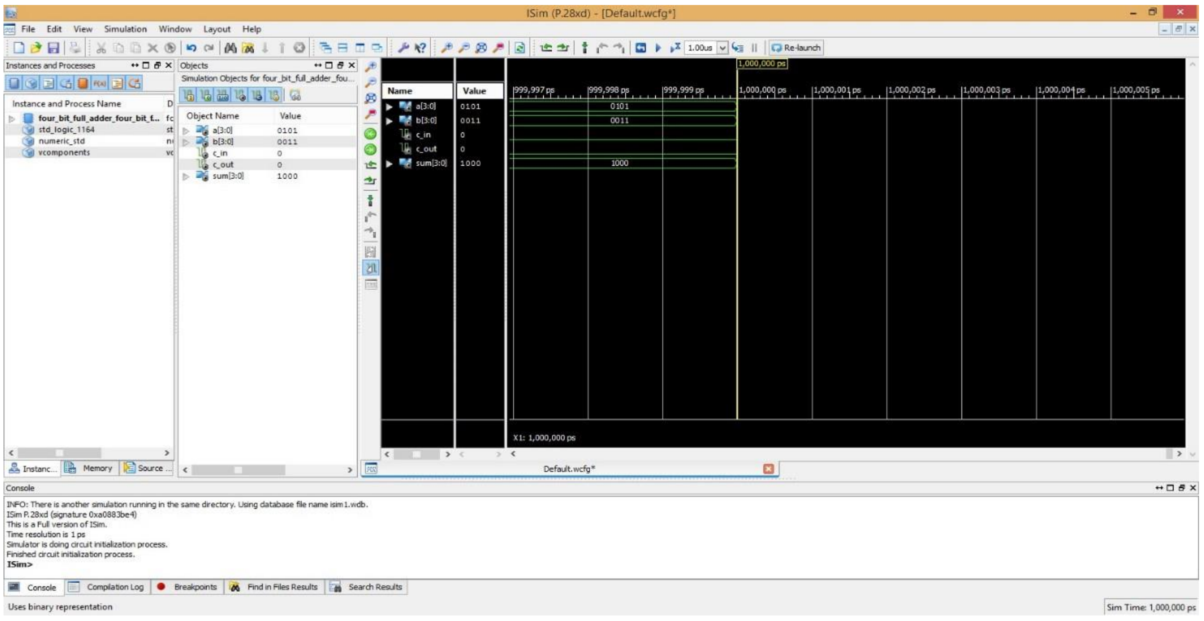
When you use the area select tool, select the objects that:
[✓ Are enclosed by the area]
Intersect the area

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

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).


ISE Project Navigator (P.28xd) - D:\adder\adder.xise - [test_4_bit_full_adder.vhd]
Processes: four_bit_full_adder
Create Schematic Symbol

```
        B :  IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        C_in :  IN STD_LOGIC;
        C_out   :  OUT   STD_LOGIC;
        Sum  :  OUT   STD_LOGIC_VECTOR (3 DOWNTO 0));
   END COMPONENT;

   SIGNAL A :  STD_LOGIC_VECTOR (3 DOWNTO 0);
   SIGNAL B :  STD_LOGIC_VECTOR (3 DOWNTO 0);
   SIGNAL C_in :  STD_LOGIC;
   SIGNAL C_out   :  STD_LOGIC;
   SIGNAL Sum  :  STD_LOGIC_VECTOR (3 DOWNTO 0);

BEGIN

   UUT: four_bit_full_adder PORT MAP(
      A => A,
      B => B,
      C_in => C_in,
      C_out => C_out,
      Sum => Sum
   );

-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN

   wait for 50 ns;
   A <= "0101";
   B <= "0011";
   C_in <= '0';
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```



Object Name	Value
a[3:0]	0101
b[3:0]	0011
c_in	0
c_out	0
sum[3:0]	1000

```
INFO: There is another simulation running in the same directory. Using database file name isim1.wdb.
ISim P.28xd (signature 0xa0883be4)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Uses binary representation