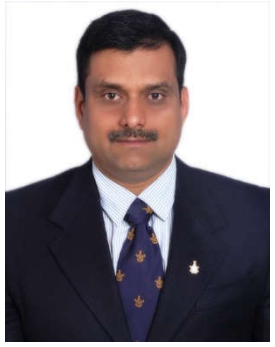
BITS Pilani
Hyderabad Campus
Department of Electrical Engineering

Digital Design : 2020-21
Lab 4
Introduction to Verilog and Implementation of Majority Circuit in Xilinx ISE

By Dr. Sanjay Vidhyadharan



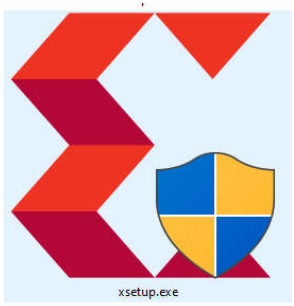
(Go to Slide no. 3 for Windows Home Edition)

For Window Professional

1. Download
Xilinx_ISE_For_Windows_Professional_S6_Win10_14.7_ISE_VMs_0206_1.zip

from http://sanjayvidhyadharan.in/Downloads/

2. Install xsetup.exe

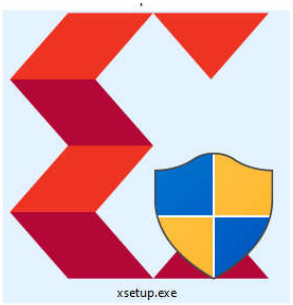


For Window Home

1. Download
   Xilinx_ISE_For_Window_Home_DS_Win_14.1_P.15xf.0.1.rar

   from http://sanjayvidhyadharan.in/Downloads/

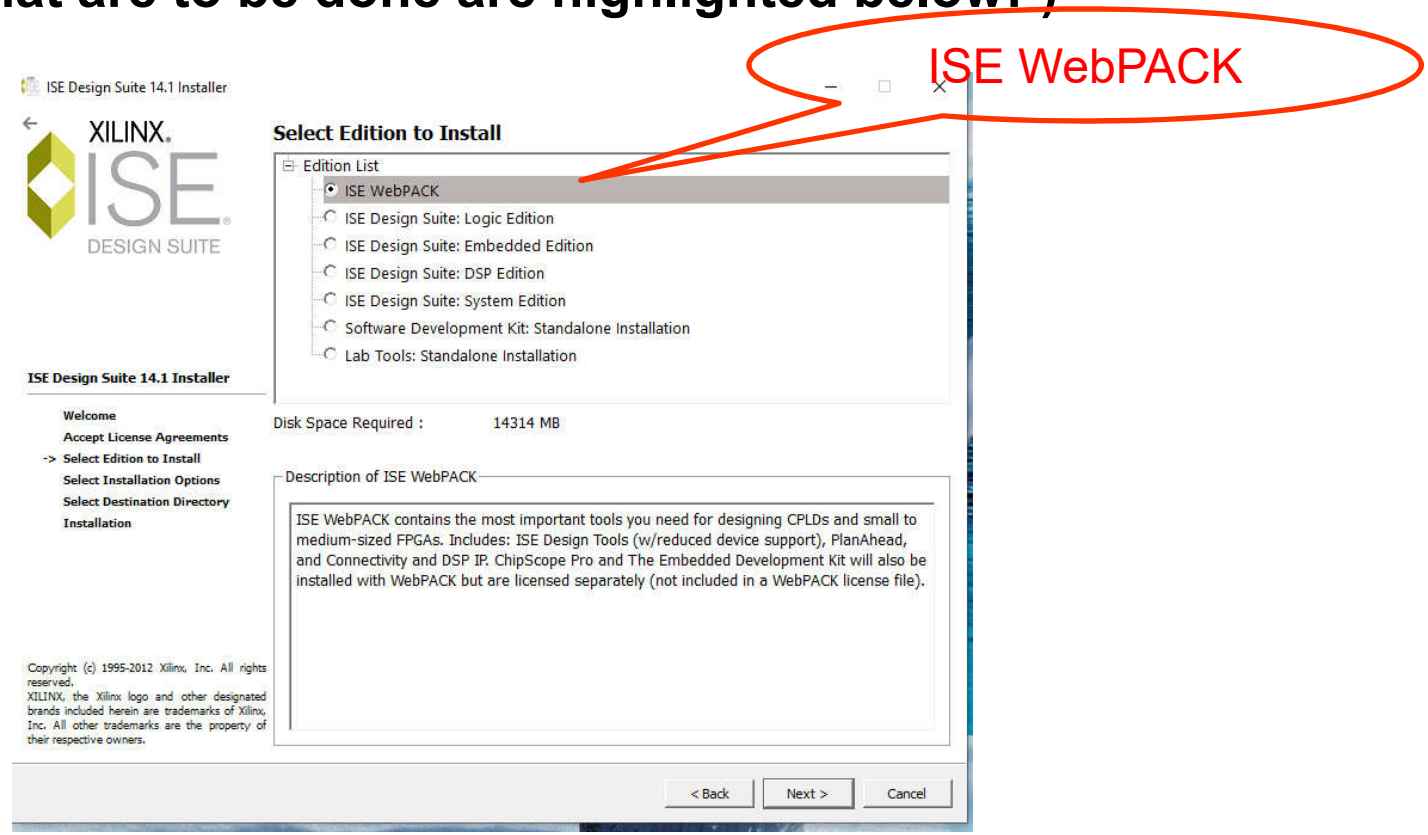
2. Install xsetup.exe



3. Make the following choices during installation process

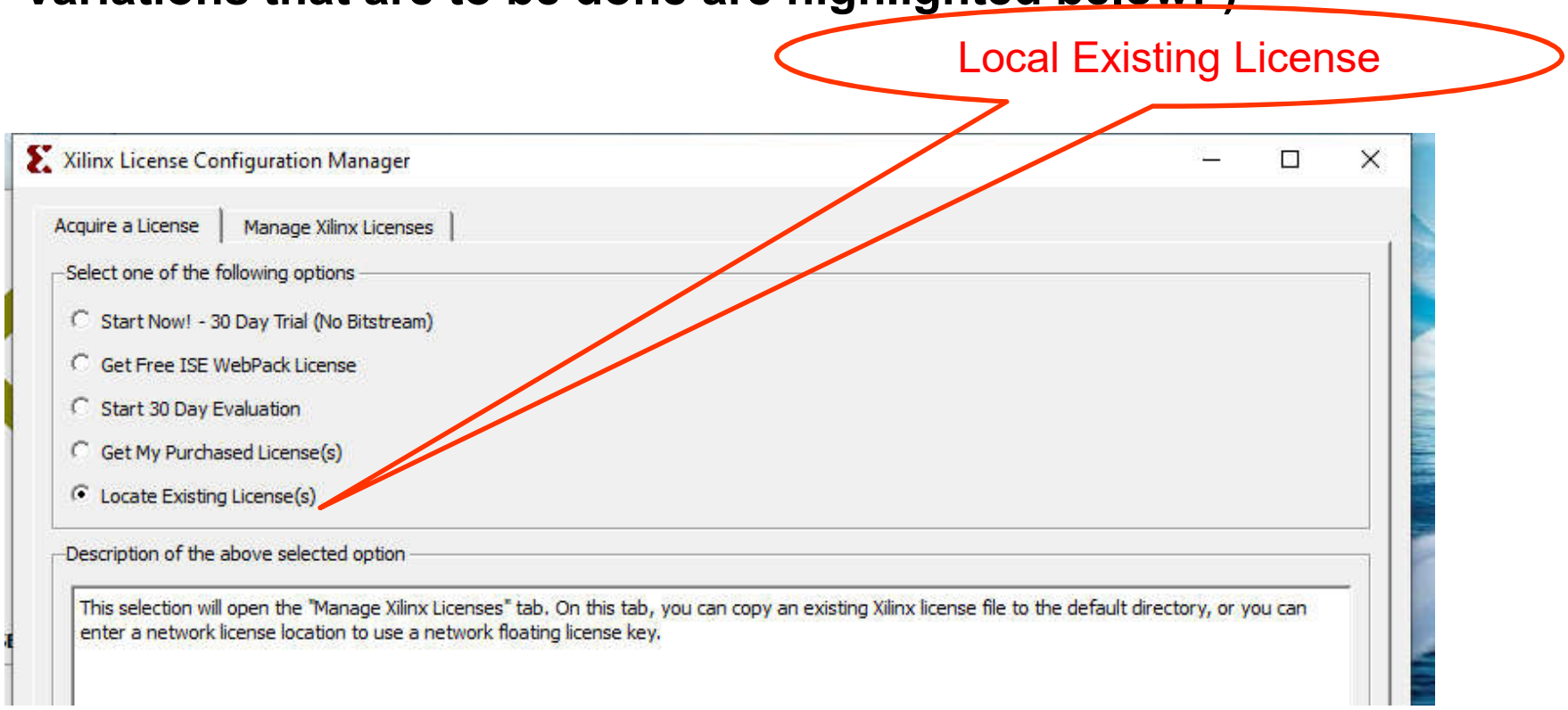
For Window Home

3. Make the following specific choices during installation process (Most of them are default and not shown below. Only Variations that are to be done are highlighted below:-)



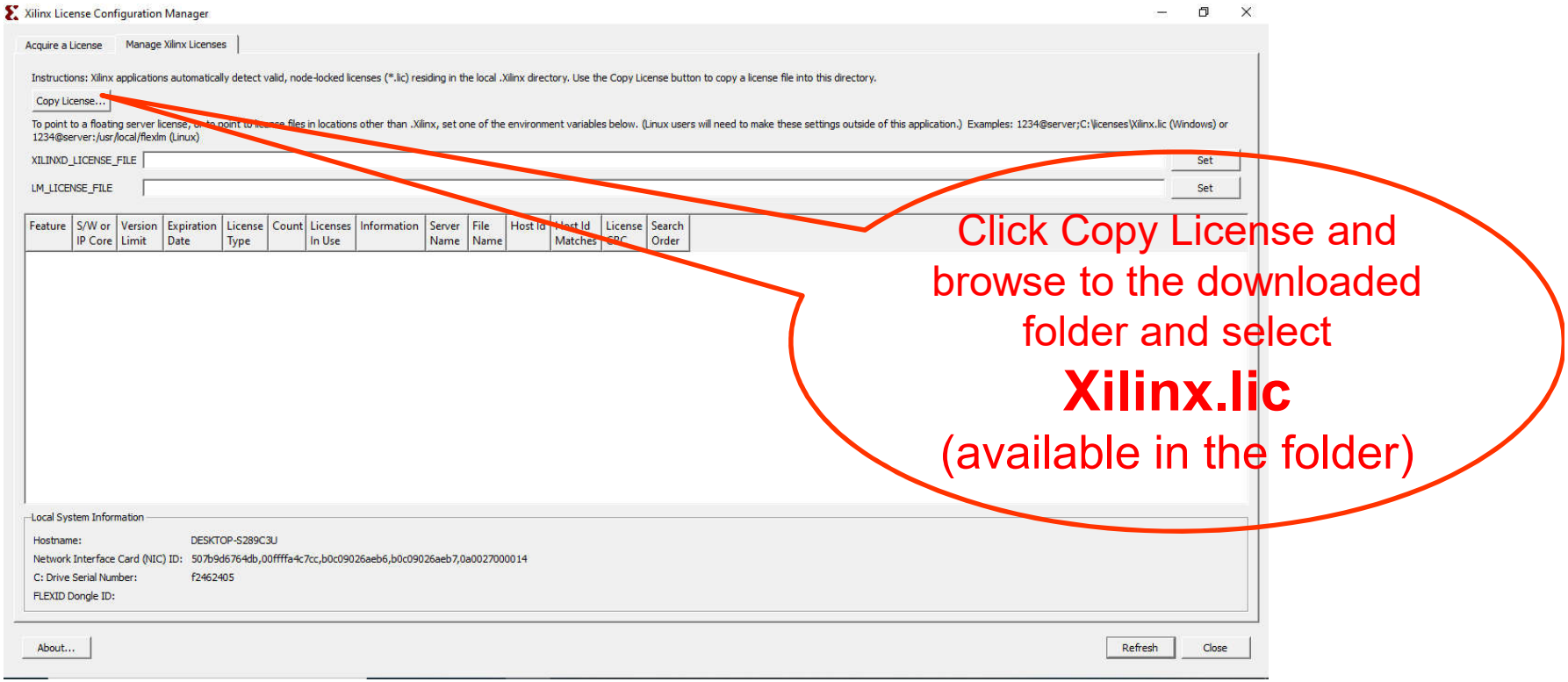
For Window Home

3. Make the following specific choices during installation process (Most of them are default and not shown below. Only Variations that are to be done are highlighted below:-)



For Window Home

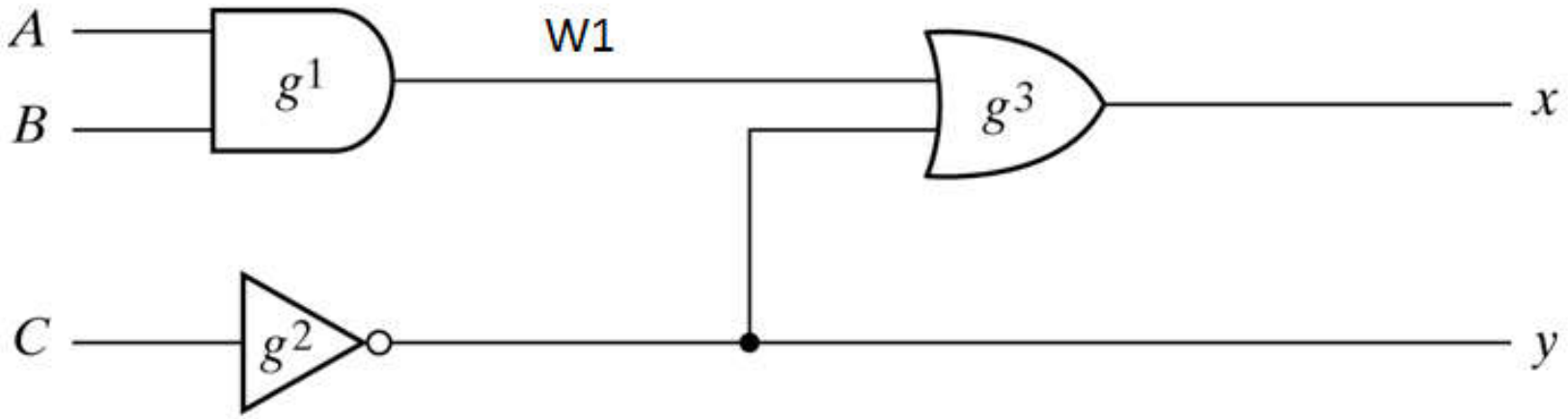
3. Make the following specific choices during installation process (Most of them are default and not shown below. Only Variations that are to be done are highlighted below:-)



Simple Circuit for Demonstration

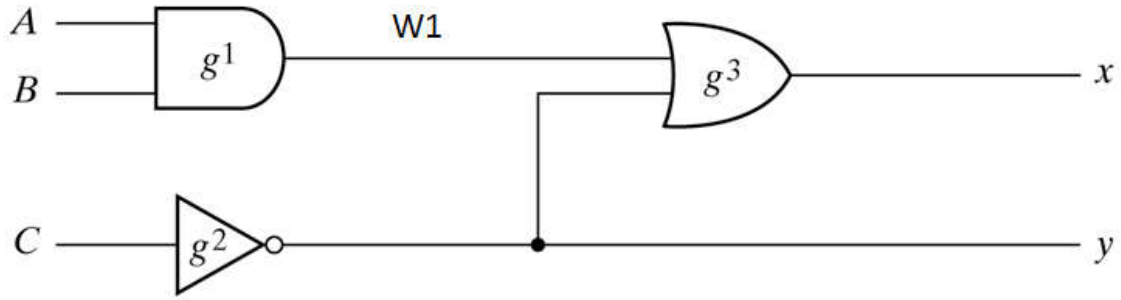
Gate Level Modelling

Simple Circuit to demonstrate HDL



Simple Circuit for Demonstration

Verilog Code




```
module     Simple_circuit (input A, input B, input C,
output x, output y);
wire       w1;

and        g1 (w1,A,B); // and gate instance
not        g2 (y,C);
or         g3 (x,w1,y);

endmodule
```

Problem Definition

1. Three-bit Majority Circuit
2. Parity Encoder

Additional Problems for Practice

- Gray to Binary code

Learning Objectives

(1) Implement and verify the following modules using Verilog gate level modelling

3-input majority function using AND – OR gates

(2) Implementing the Verilog designs on to FPGA

Exercise 1: Design and Implementation of simplified SOP form of a Boolean 3-input majority function using AND – OR gates

Design & Implementation for simplified SOP form of Majority Function: A three-input majority circuit having A(MSB), B and C(LSB) as inputs with output F is shown in the figure below.

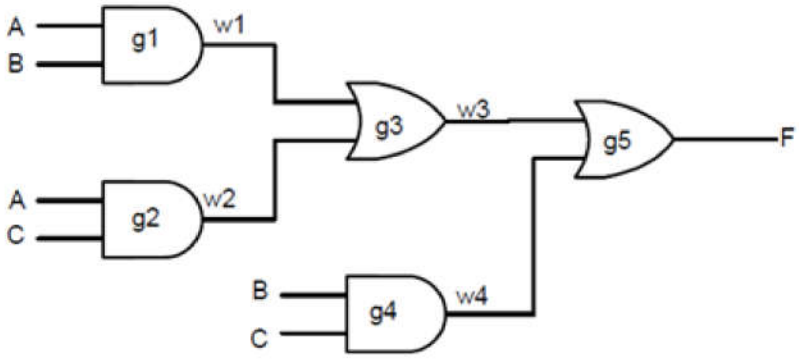


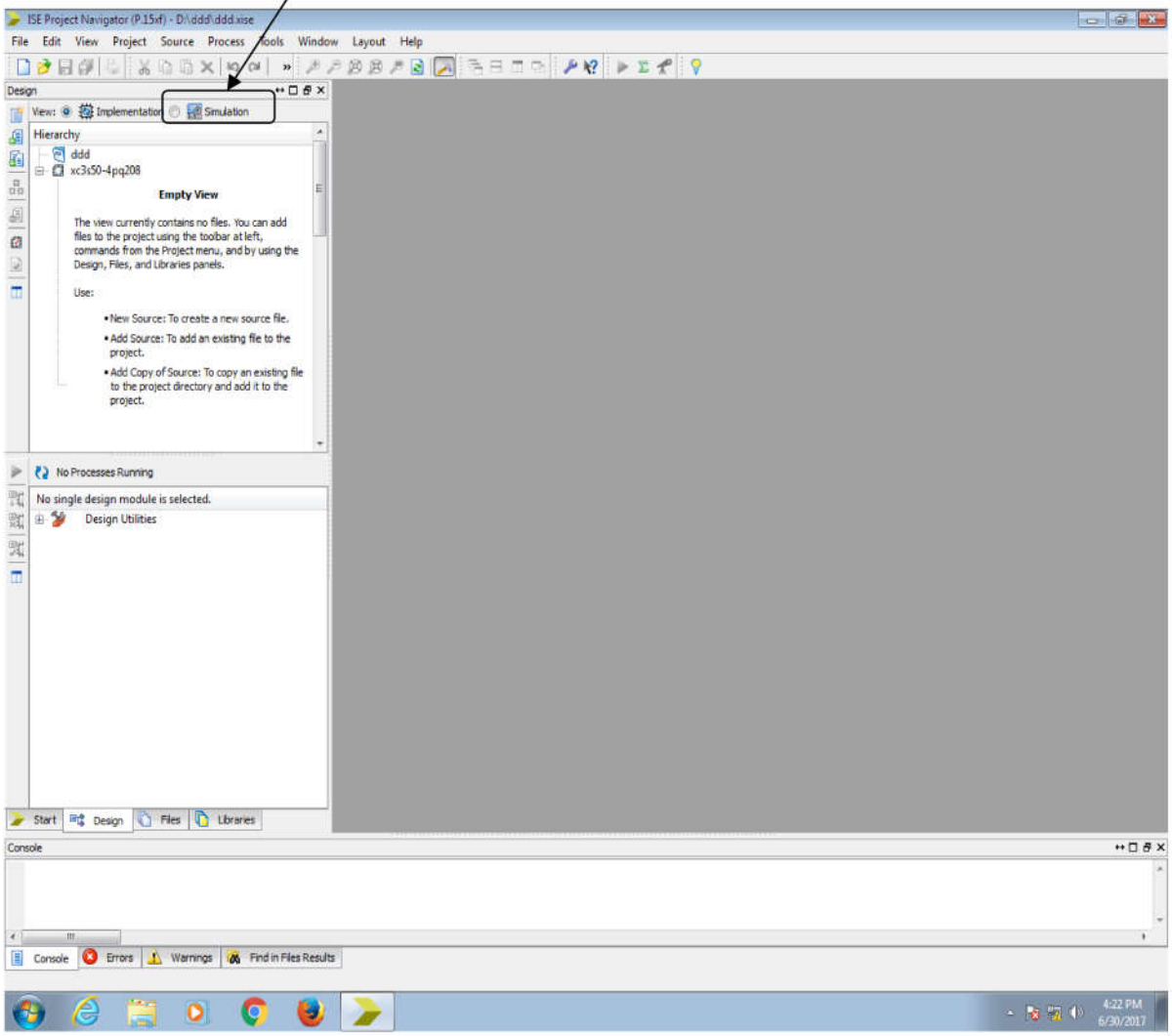
Figure 3.1 : Three-input Majority Circuit

STEPS FOR IMPLEMENTING THE VERILOG CODE

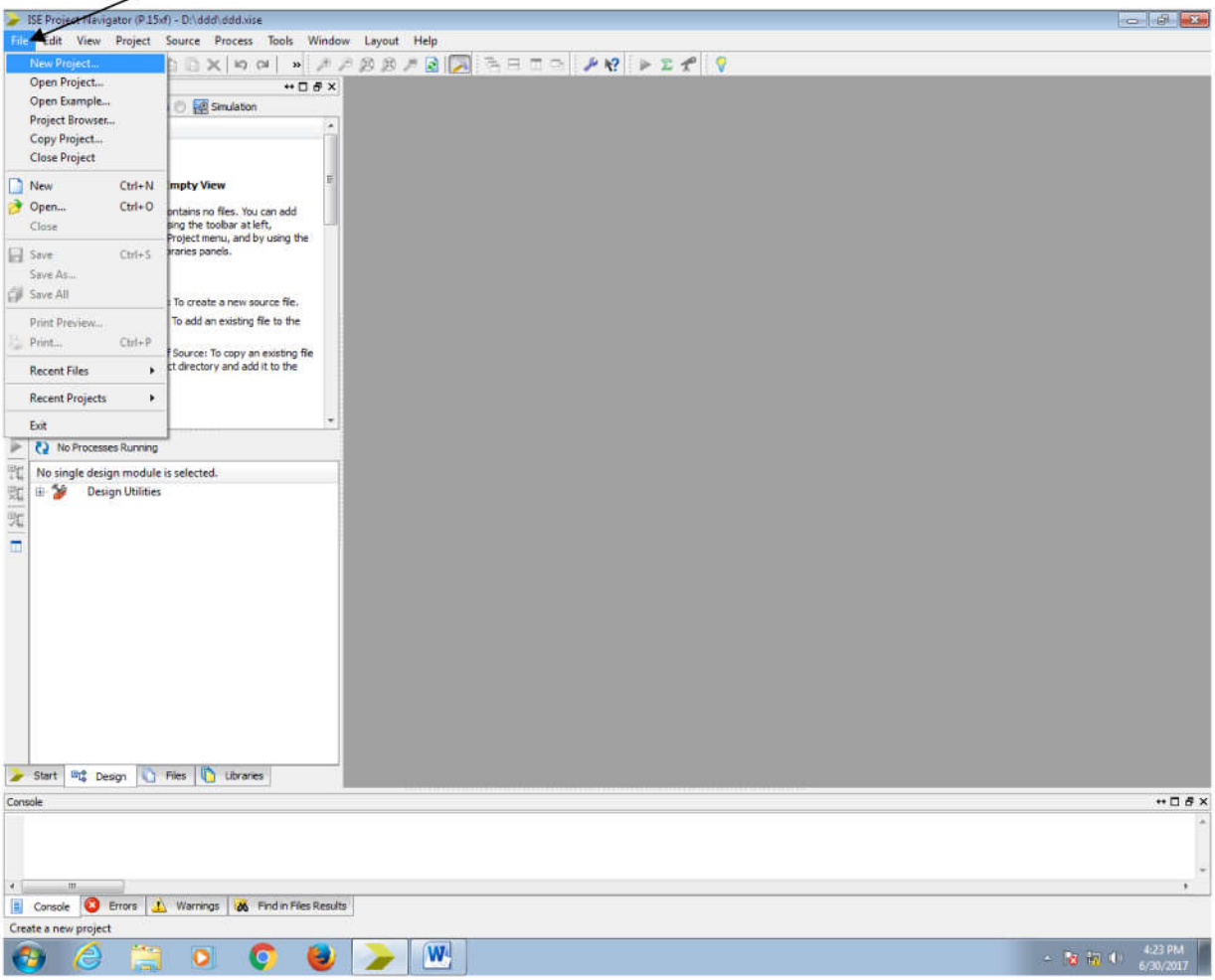
Click on ISE Design 14.1 

in desktop

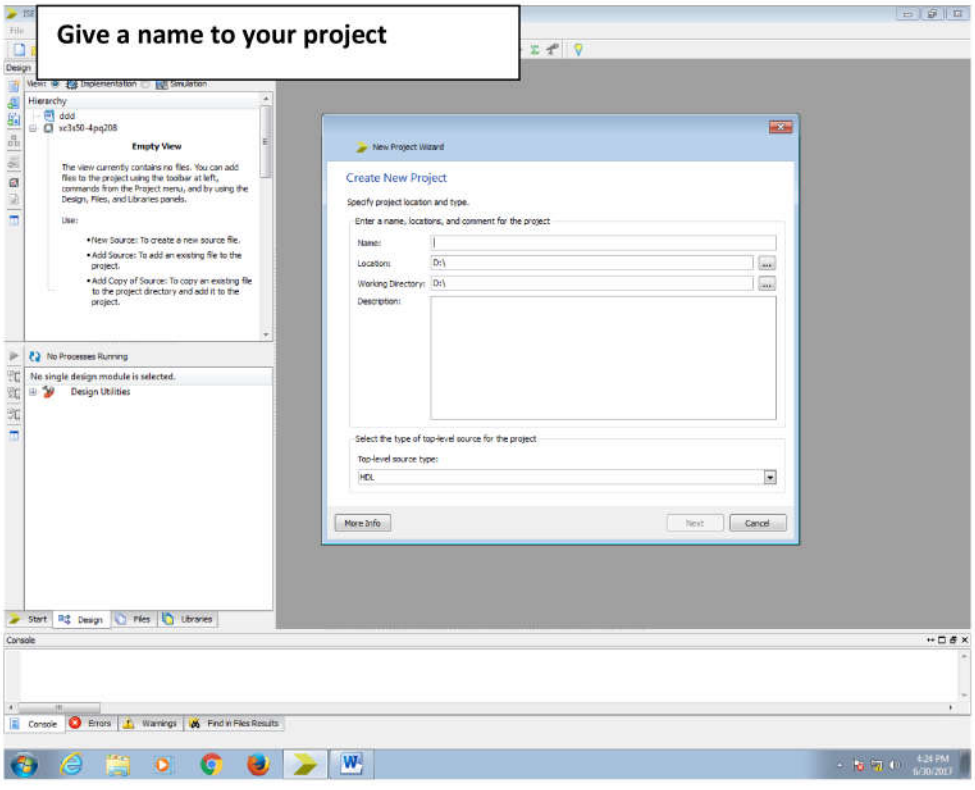
Select the Simulation option when you are implementing the Verilog code and Testing

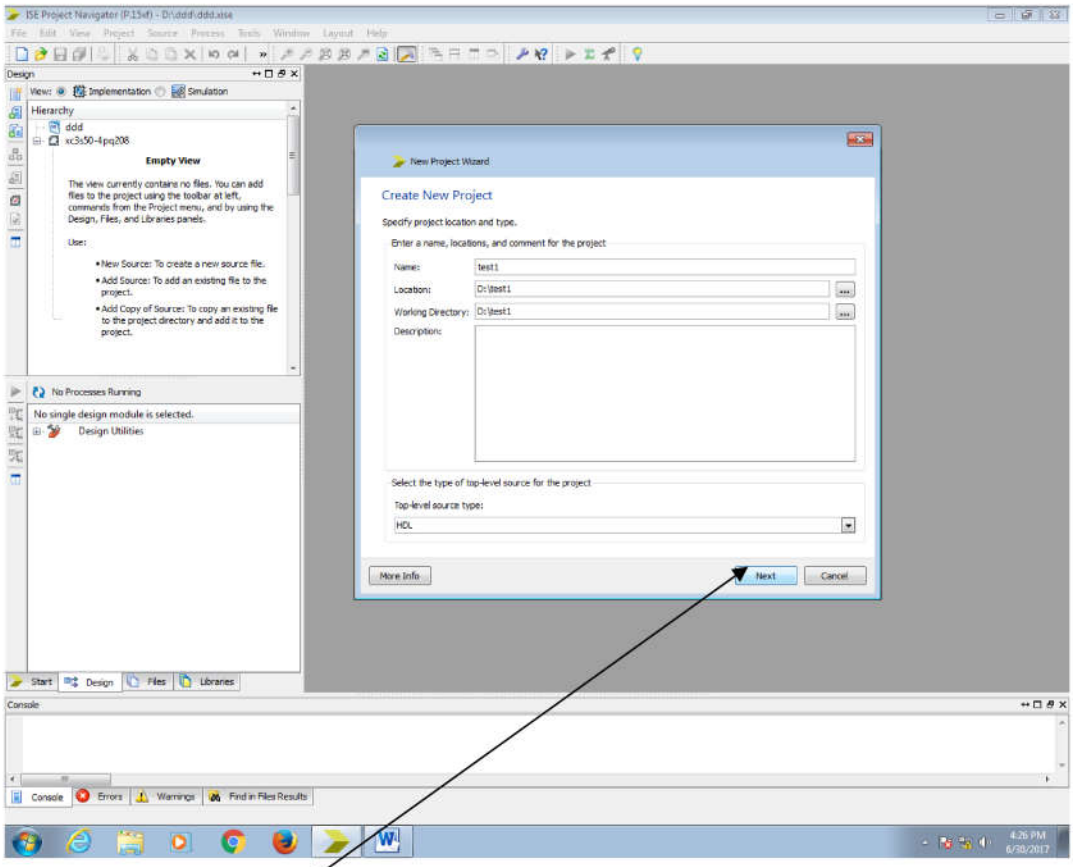


Go to File and click on New Project

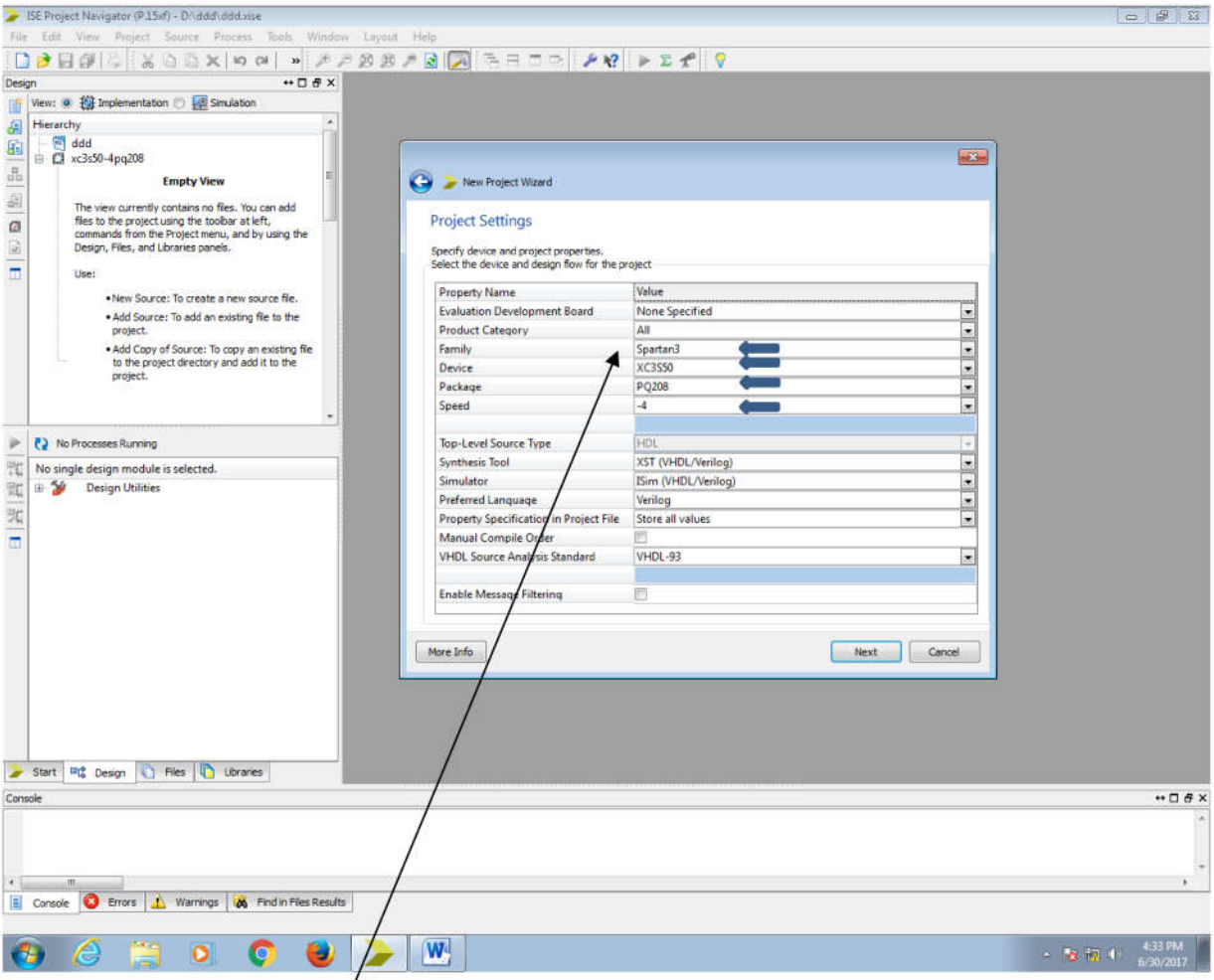


Give a name to your project

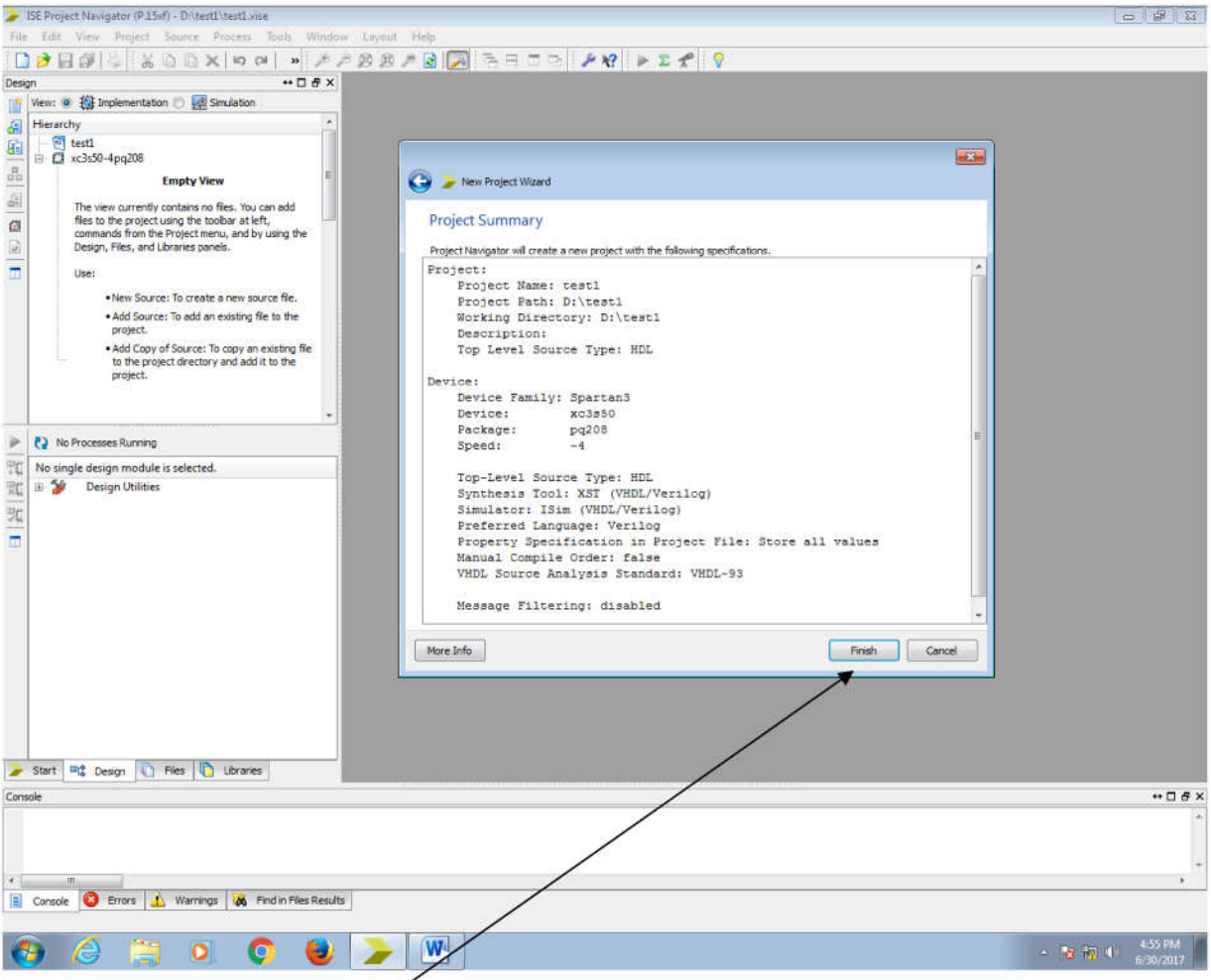




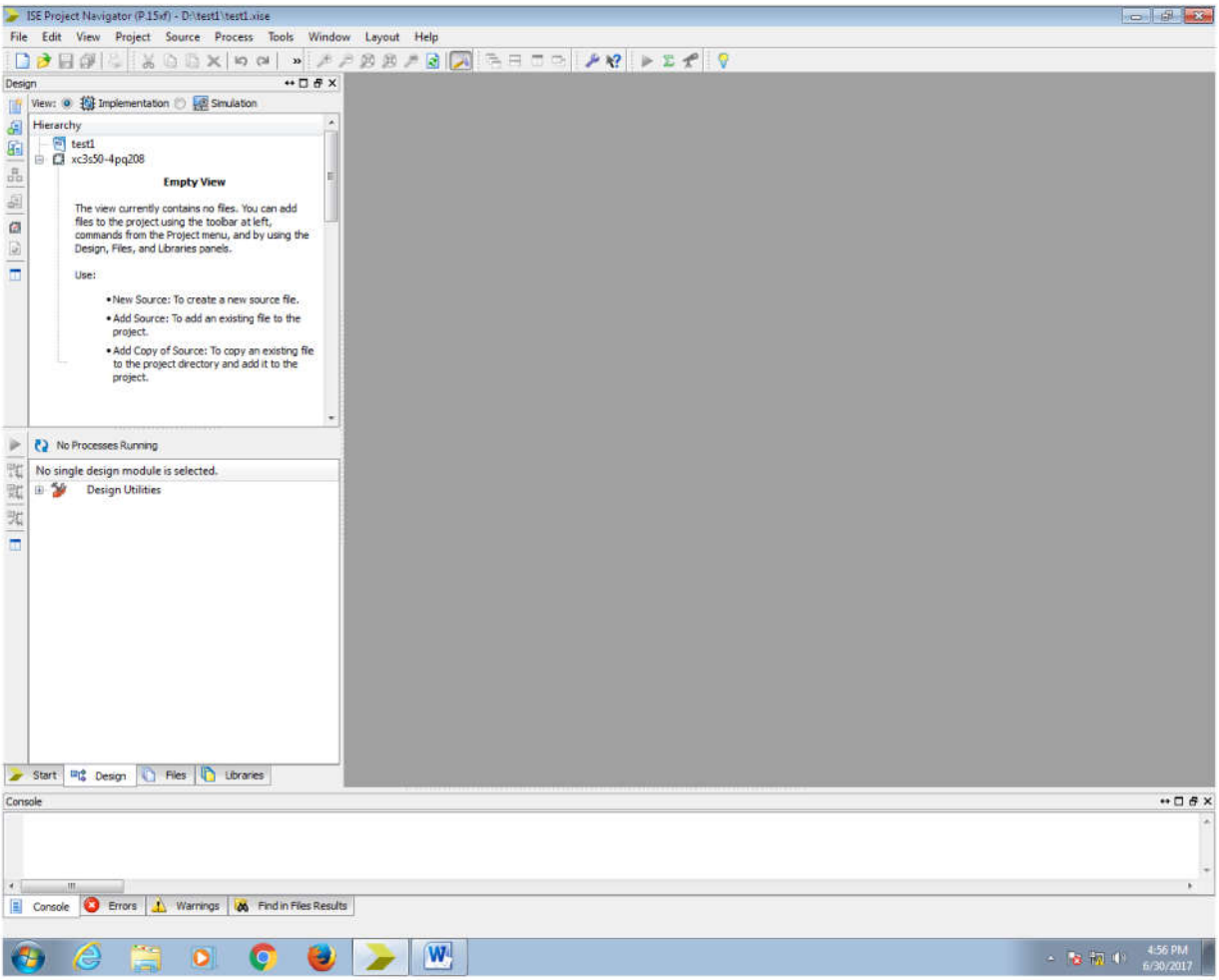
Click on <u>Next</u> after giving the name

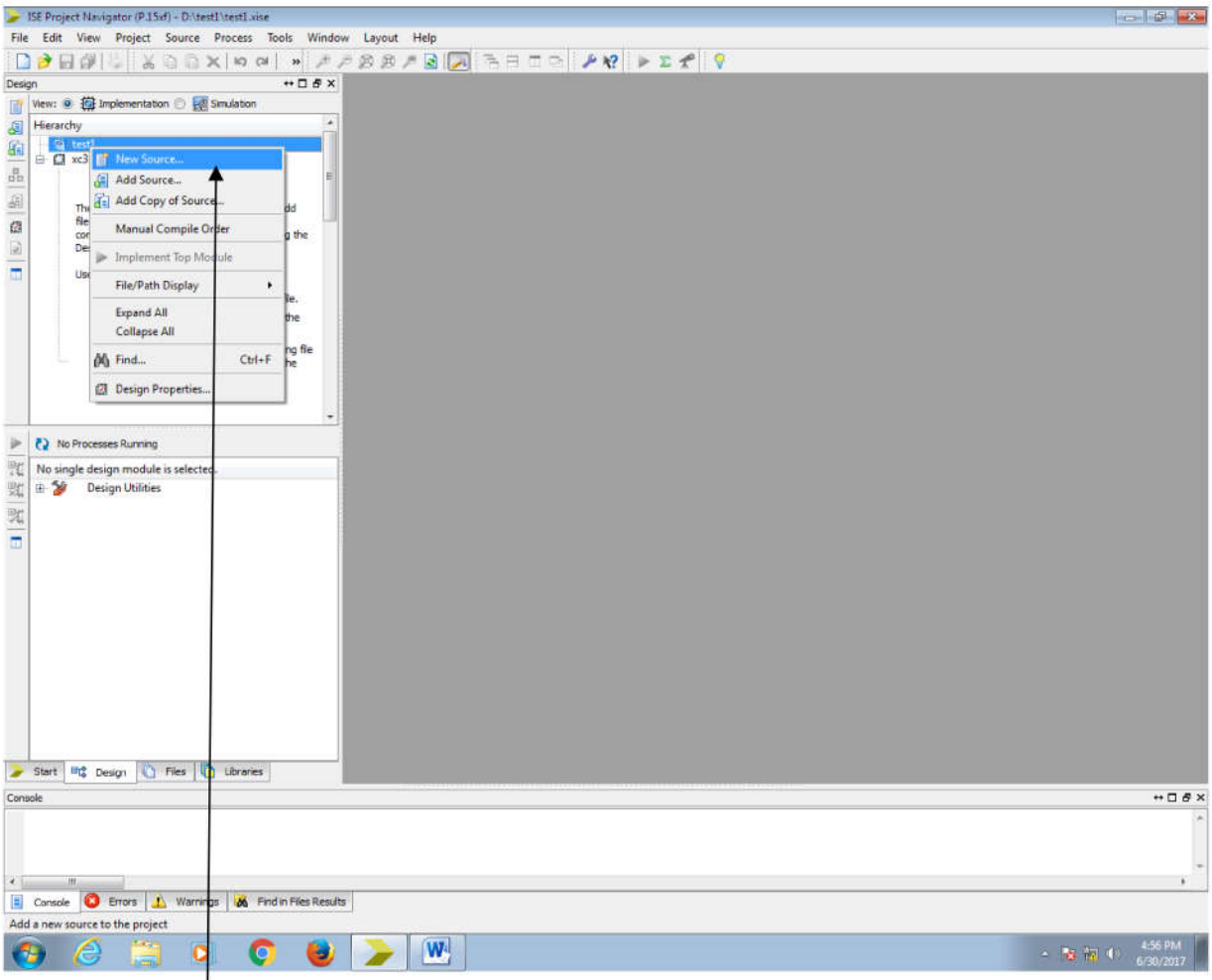


Select the <u>Family, Device and Package</u> as shown in the figure.

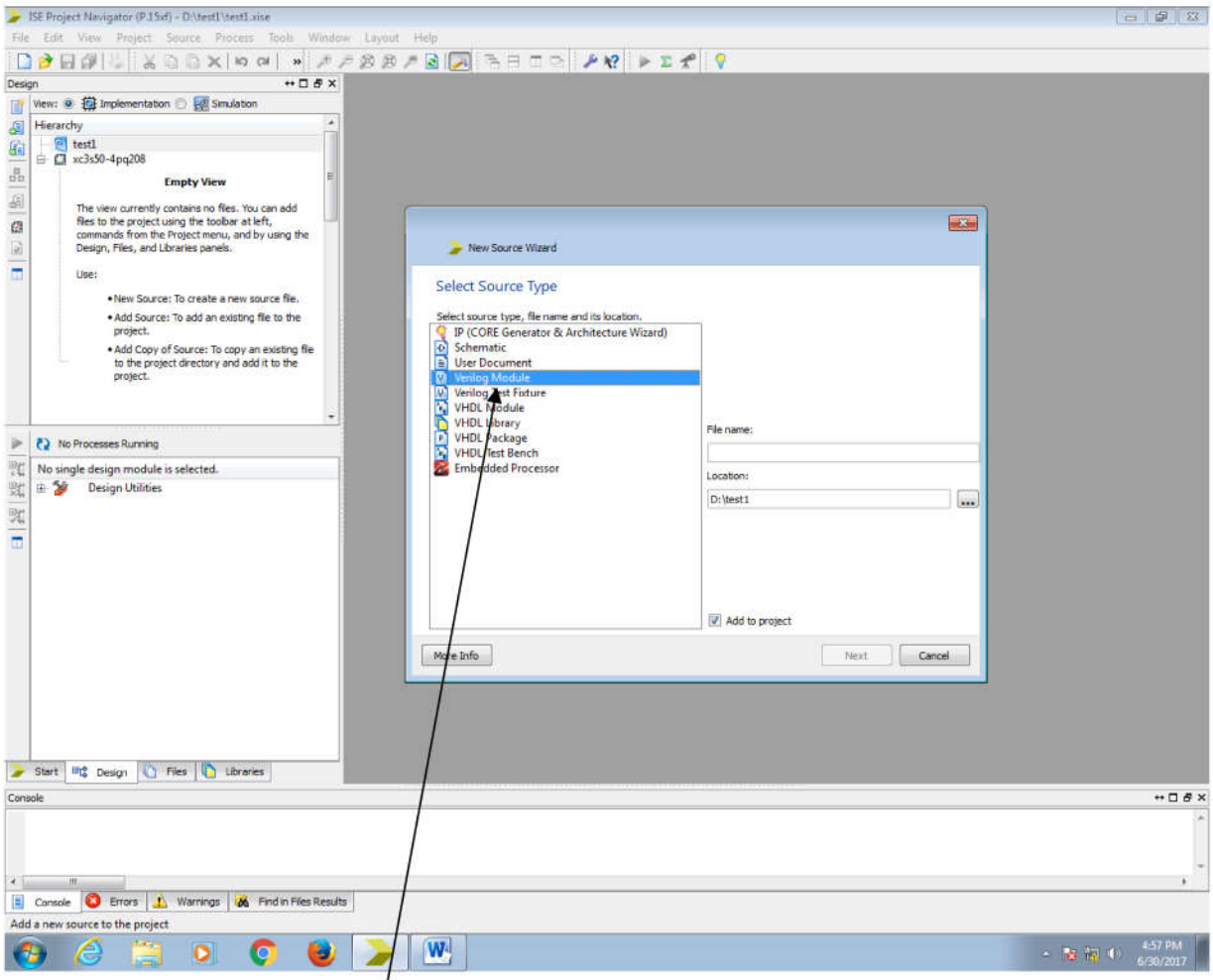


Click on <u>FINISH</u> after the above steps

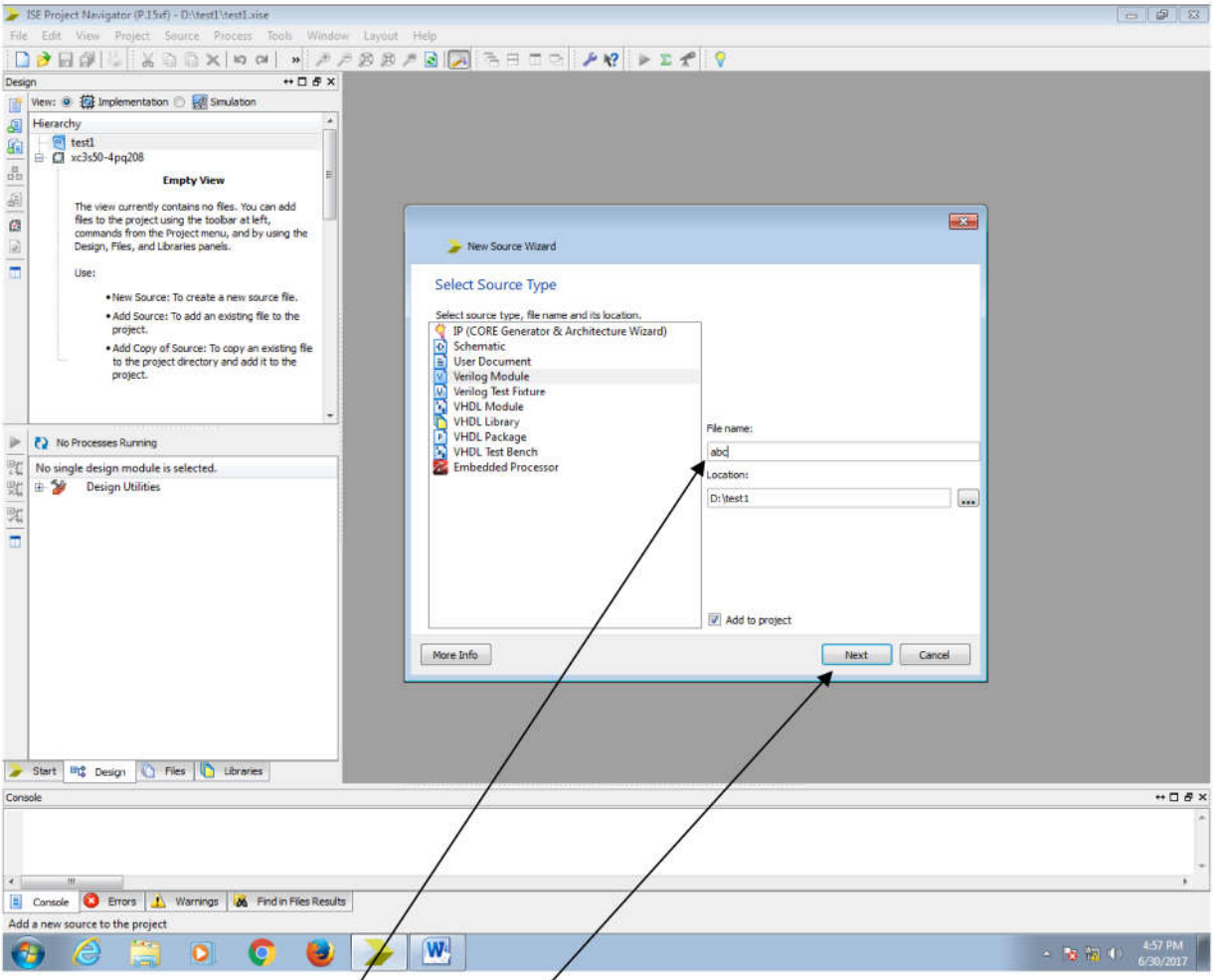

ISE Project Navigator (P.15xf) - D:\test1\test1.xise
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
test1
xc3s50-4pq208
Empty View
The view currently contains no files. You can add files to the project using the toolbar at left, commands from the Project menu, and by using the Design, Files, and Libraries panels.
Use:
• New Source: To create a new source file.
• Add Source: To add an existing file to the project.
• Add Copy of Source: To copy an existing file to the project directory and add it to the project.
No Processes Running
No single design module is selected.
Design Utilities
Start Design Files Libraries
Console
Console Errors Warnings Find in Files Results
4:56 PM
6/30/2017



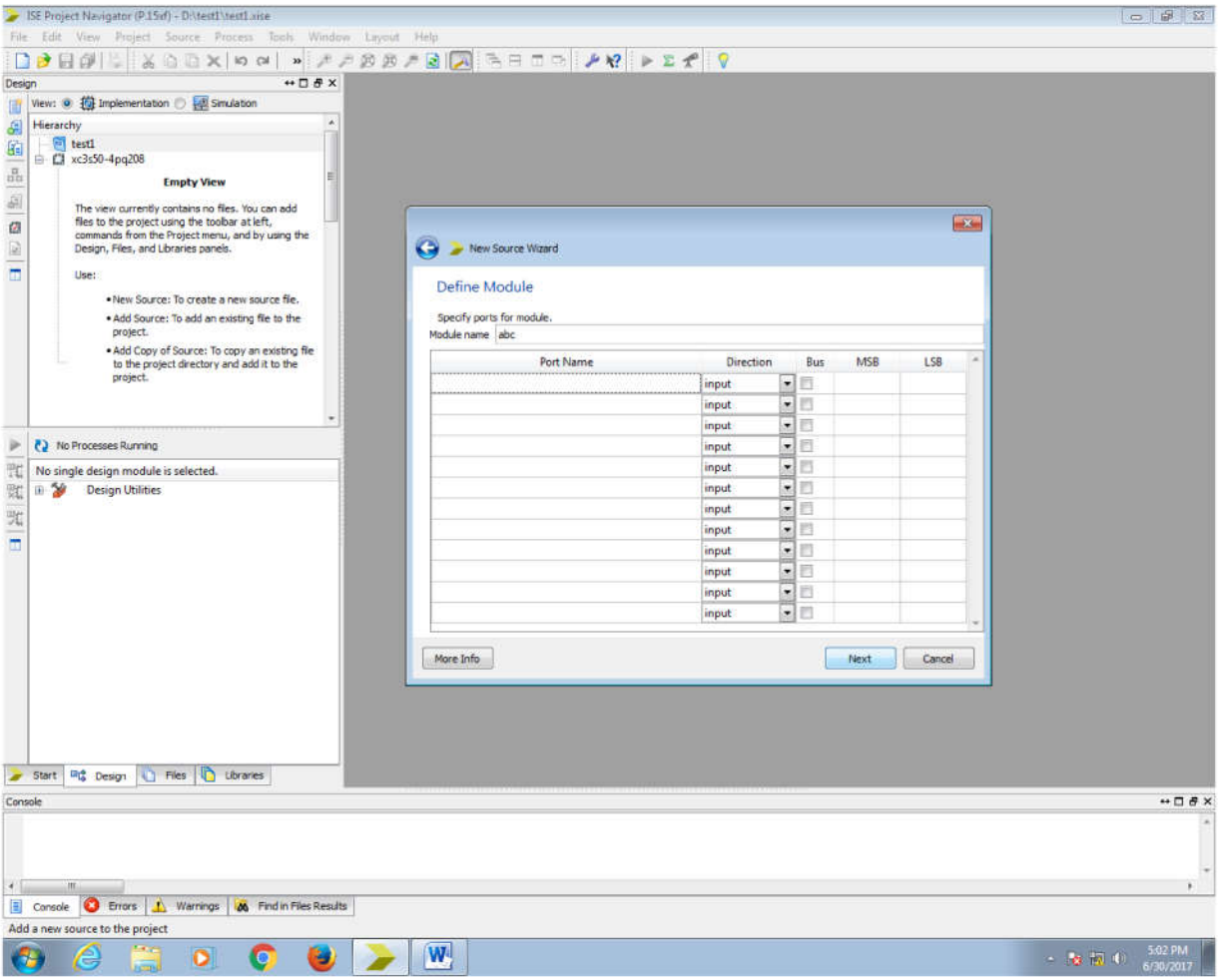
Right Click on the project you have created and then click on New Source



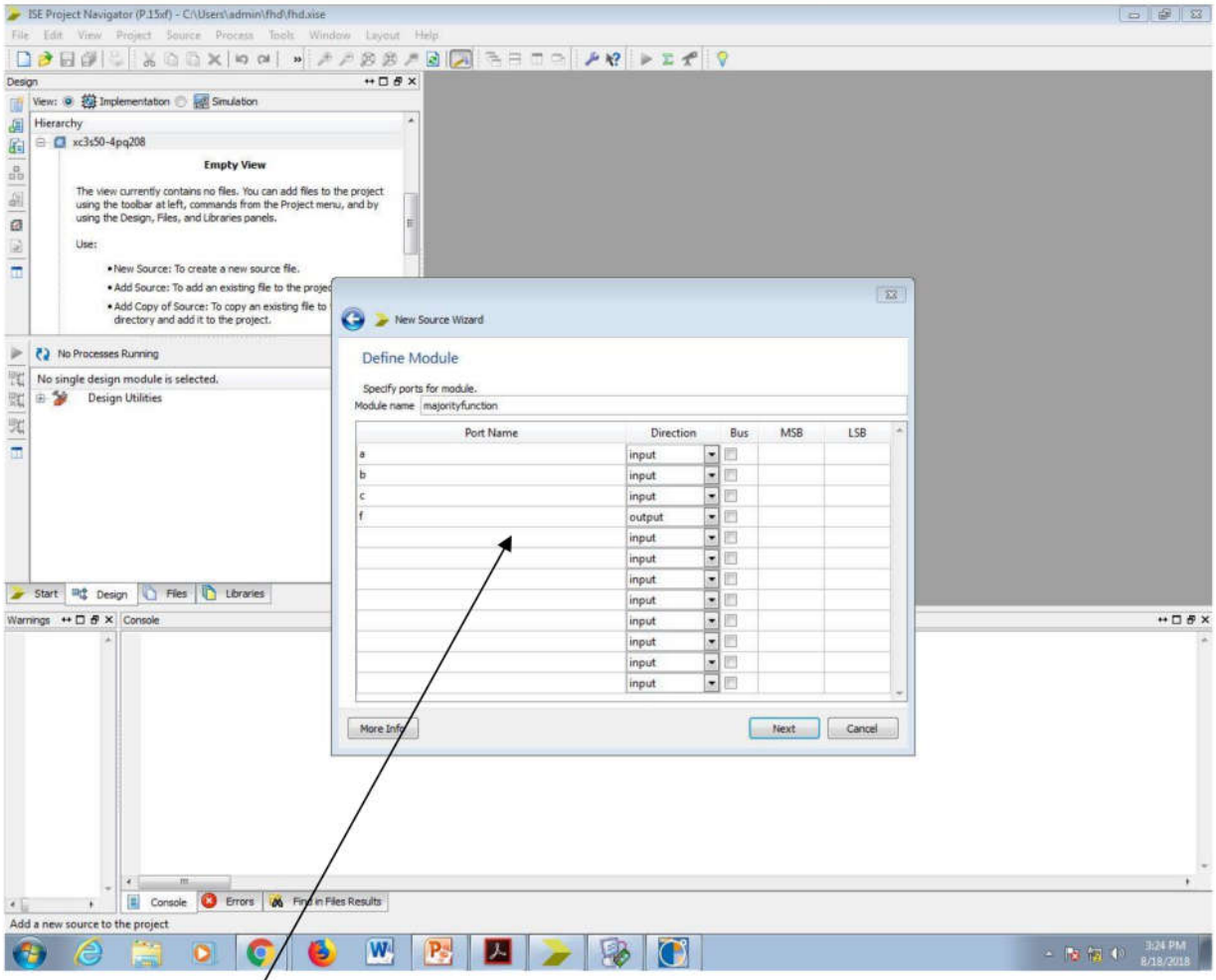
Select **VERILOG MODULE**



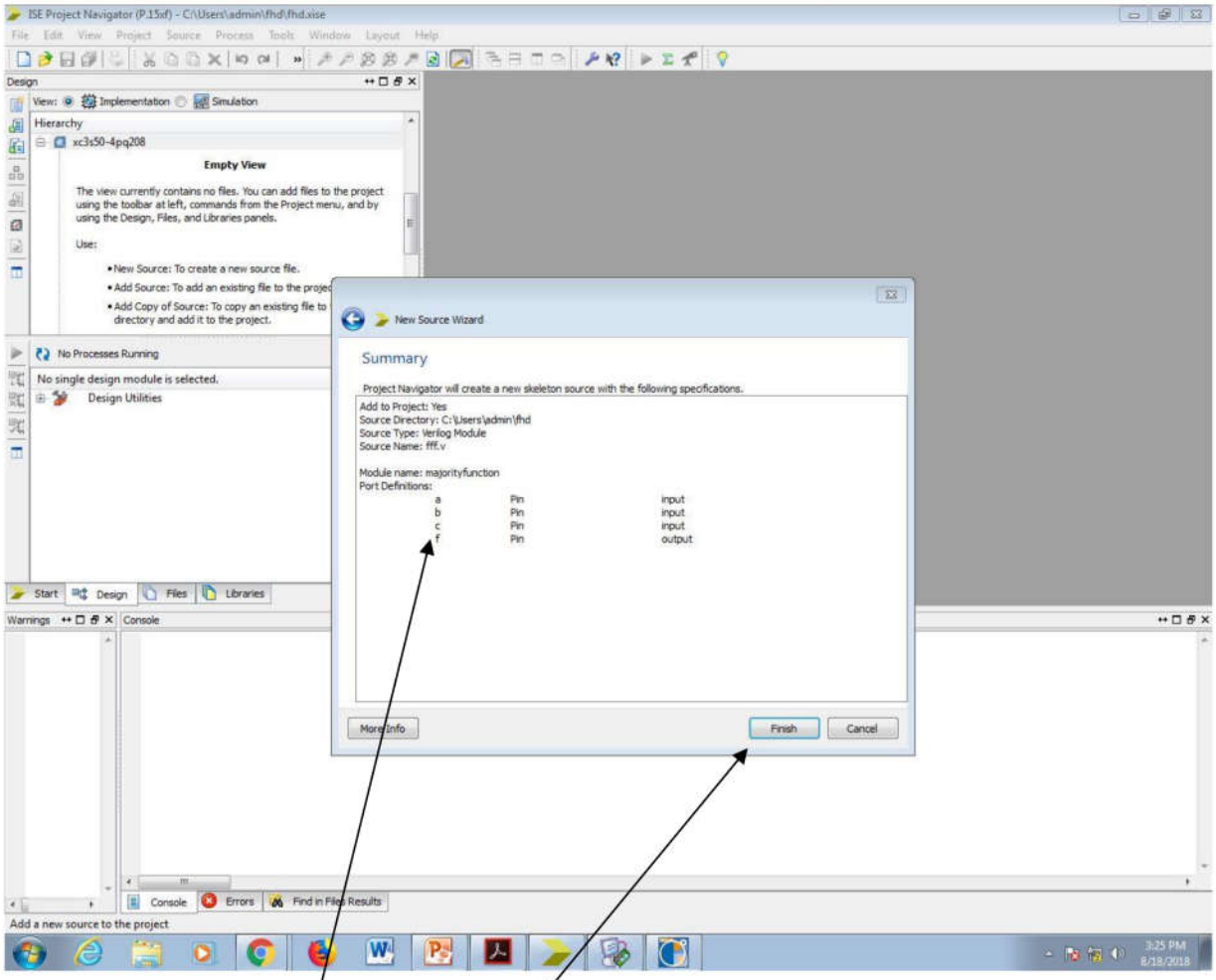
Give a **file name** and then click on **NEXT**.



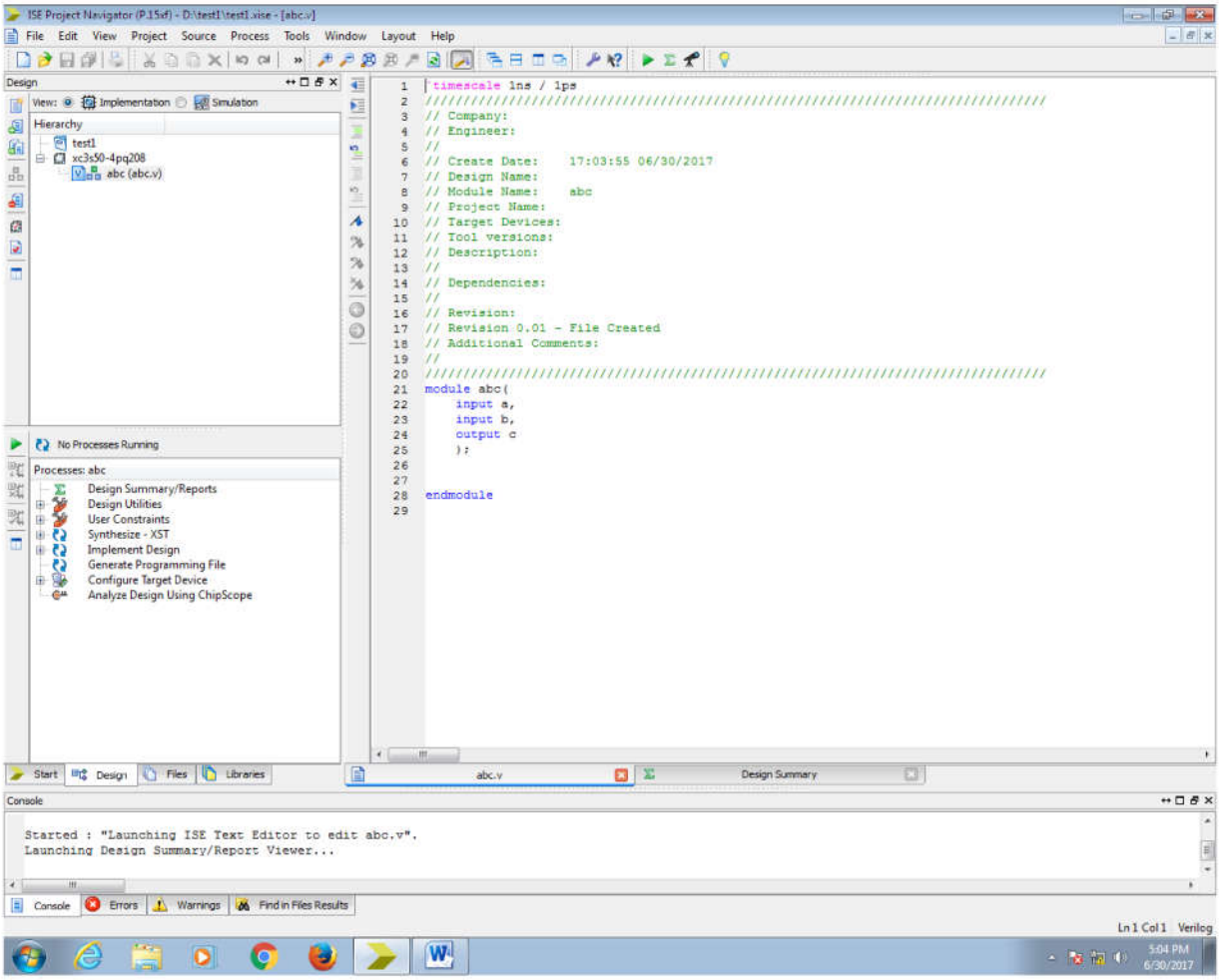
Specify **INPUTS** and **OUPUTS** for your design.



After specifying INPUTS and OUTPUTS, click on <u>NEXT</u>.



Summary of the given inputs and outputs is shown. Now click on <u>FINISH</u>.




```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    17:03:55 06/30/2017
7  // Design Name:
8  // Module Name:    abc
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21 module abc(
22     input a,
23     input b,
24     output c
25     );
26
27
28 endmodule
29
```

A module will appear with the given inputs and outputs.

Write Verilog code for gate-level design of Majority Function using logic gates. In this design you require three 2-input AND gates and two 2-input OR gates.

Partial Verilog Code for Majority Function: major_func.v

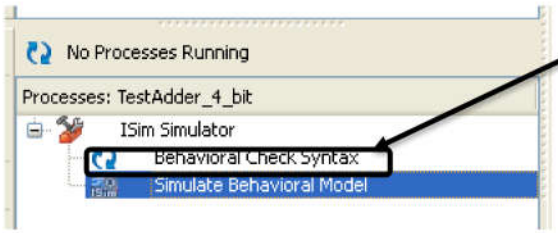
```
module major_func (
        Input A,
        Input B,
        Input C,
        Output F
        );
```

```
Wire w1, w2, w3, w4;
and g1(w1, A, B); // AND gate instance for g1
// with A & B are as inputs and w1 as output.

// similarly write the AND gate instances
// for g2 & g4

//w1 & w2 are now inputs to OR gate g3.
or g3(w3, w1, w2);

// similarly write the OR gate instance for g5.

endmodule
```

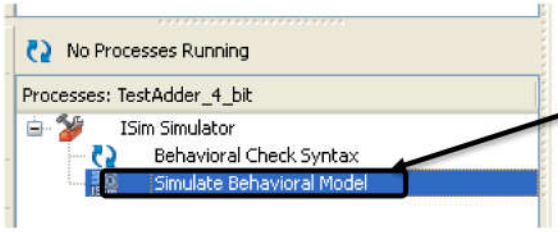
STEPS FOR TESTING THE VERILOG CODE

For testing make sure the view is selected as **Simulation**

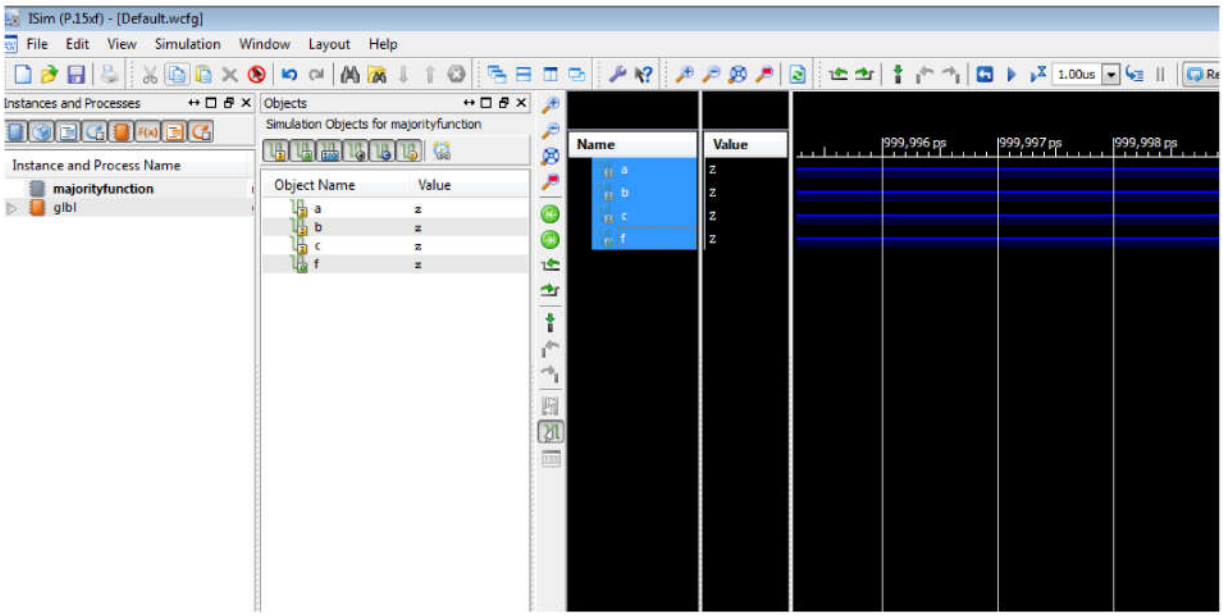
Check the syntax of the Verilog code by double clicking on **Behavioral Check syntax** in the Process window **(Under ISim Simulator).**



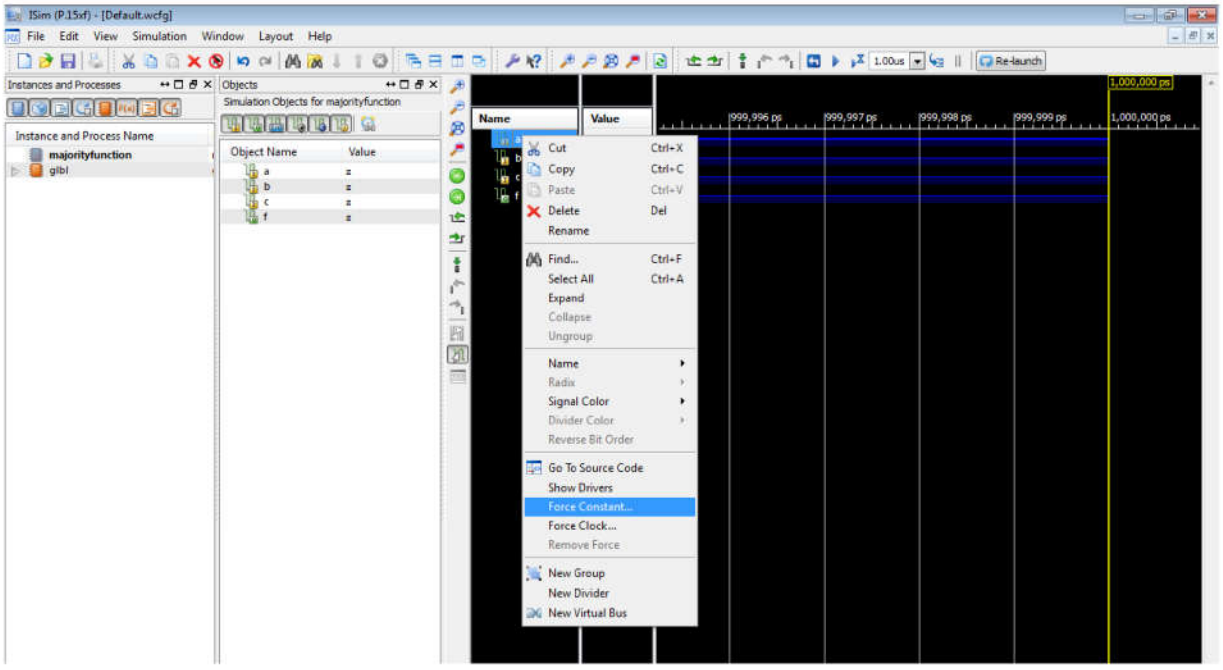
After the syntax is checked simulate your Verilog code by double clicking on **Simulate Behavioral Model**. **(Under ISim Simulator)**



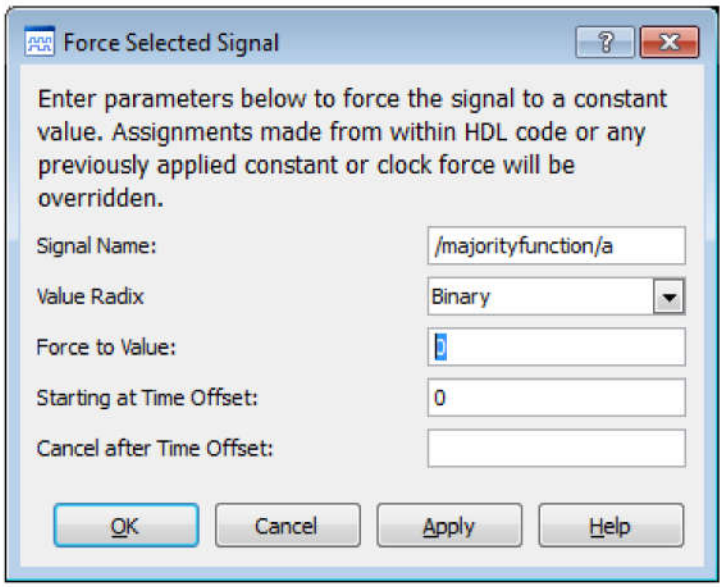
A simulation window will open.



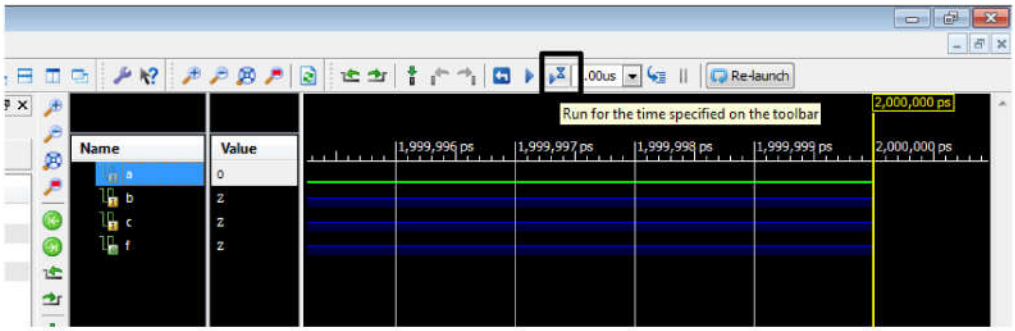
Here you have to give different patterns to inputs by right clicking on the input select **<u>"Force Constant"</u>** .



Specify the value of the input. Then Click on **Ok.**



Repeat this for all inputs and then click to run the simulation for specified amount of time.



You can test your design by 8 input combinations below. Also follow these steps

	0	1	2	3	4	5	6	7
A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
F	0	0	0	1	0	1	1	1

Output

Inputs

Decimal value

Step1: : After forcing the input 000, click button to run for 1us specified on the tool bar, then you can see output F getting updated in the Value column to "0".

Step2: For 2nd combination, just change C to "1" and click to run for next step of 1us, you will notice output remains same.

Step3: Repeat for 3 - 8 combinations and click Zoom to Full View button to see the final **Expected Output.**

Expected Output: