**INTRODUCTION TO VHDL:**

VHDL (Very High Speed IC Hardware description Language) is one of the standard hardware description language used to design digital systems. VHDL can be used to design the lowest level (gate level) of a digital system to the highest level (VLSI module). VHDL though being a rigid language with a standard set of rules allows the designer to use different methods of design giving different perspectives to the digital system.

Other than VHDL there are many hardware description languages available in the market for the digital designers such as Verilog, ABEL, PALASM, CUPL, and etc but VHDL and Verilog are the most widely used HDLs. The major difference between hardware description programming languages and others is the integration of time. Timing specifications are used to incorporate propagation delays present in the system.

**TYPES OF REPRESENTATION:**

VHDL representation can be seen as text file describing a digital system. The digital system can be represented in different forms such as a behavioral model or a structural model. Most commonly known as levels of abstraction, these levels help the designer to develop complex systems efficiently.

**DATAFLOW MODEL:**

Dataflow Model is convenient for illustrating asynchronous and concurrent events, where the delays represent actual hardware component delays.

**BEHAVIORAL MODEL:**

Behavioral level describes the system the way it behaves instead of a lower abstraction of its connections. Behavioral model describes the relationship between the input and output signals. The description can be a Register Transfer Level (RTL) or simple Boolean equations.

**STRUCTURAL MODEL:**

Structural level describes the systems as gates or component block interconnected to perform the desired operations. Structural level is primarily the graphical representation of the digital system and so it is closer to the actual physical representation of the system.

**VHDL PROGRAMMING**

One can design hardware in a VHDL IDE (for FPGA implementation such as Xilinx ISE, Altera Quartus, Synopsys Synplify or Mentor Graphics HDL Designer) to produce the [RTL](http://en.wikipedia.org/wiki/Register-transfer_level" \o "Register-transfer level)schematic of the desired circuit. After that, the generated schematic can be verified using simulation software which shows the waveforms of inputs and outputs of the circuit after generating the appropriate testbench.

**INTODUCTION TO XILINX**

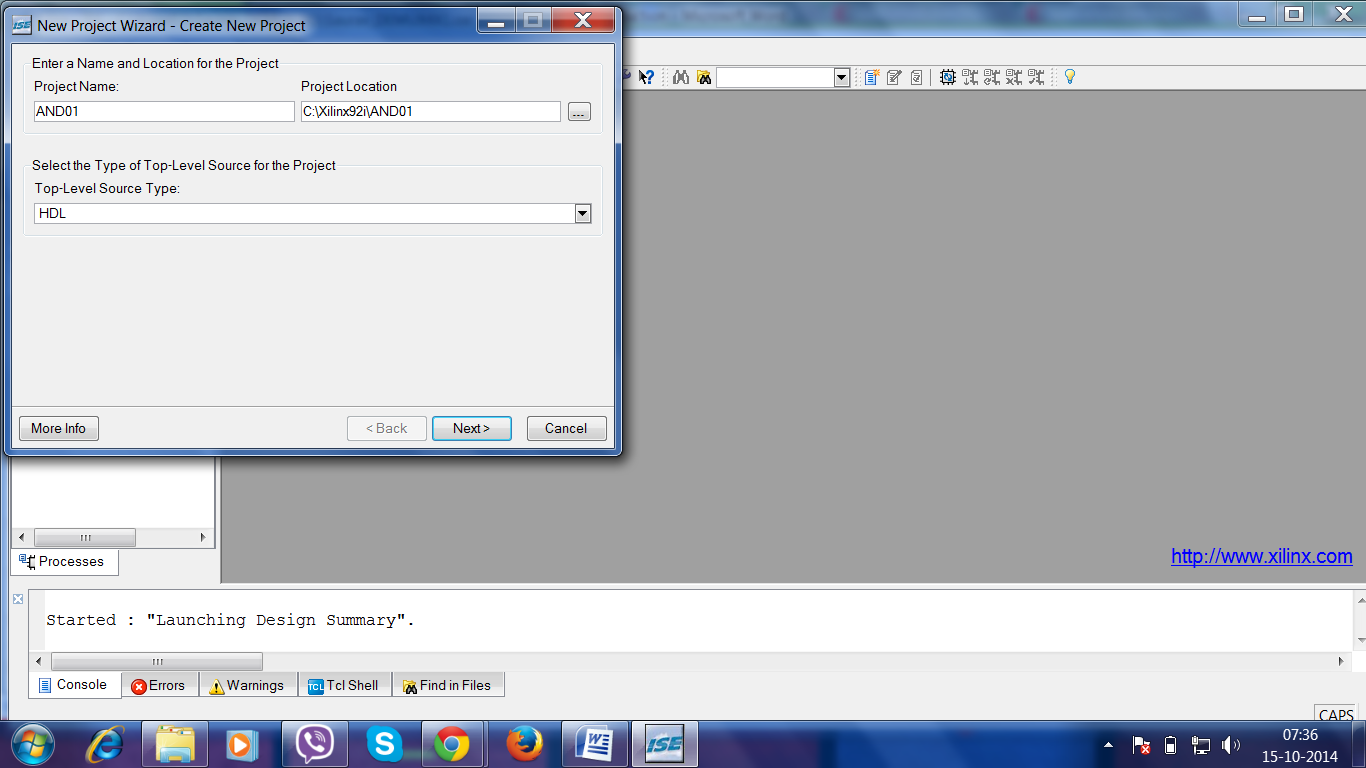
Xilinx Integrated Software Environment (ISE) is a design software suite that allows you to take your design from design entry through Xilinx device programming. The ISE Project Navigator manages and processes your design through several steps in the ISE design flow. These steps are Design Entry, Synthesis, Implementation, Simulation/Verification, and Device Configuration.

Let us see, a program to implement AND GATE in the Xilinx ISE and how to simulate it with ISE Simulator.

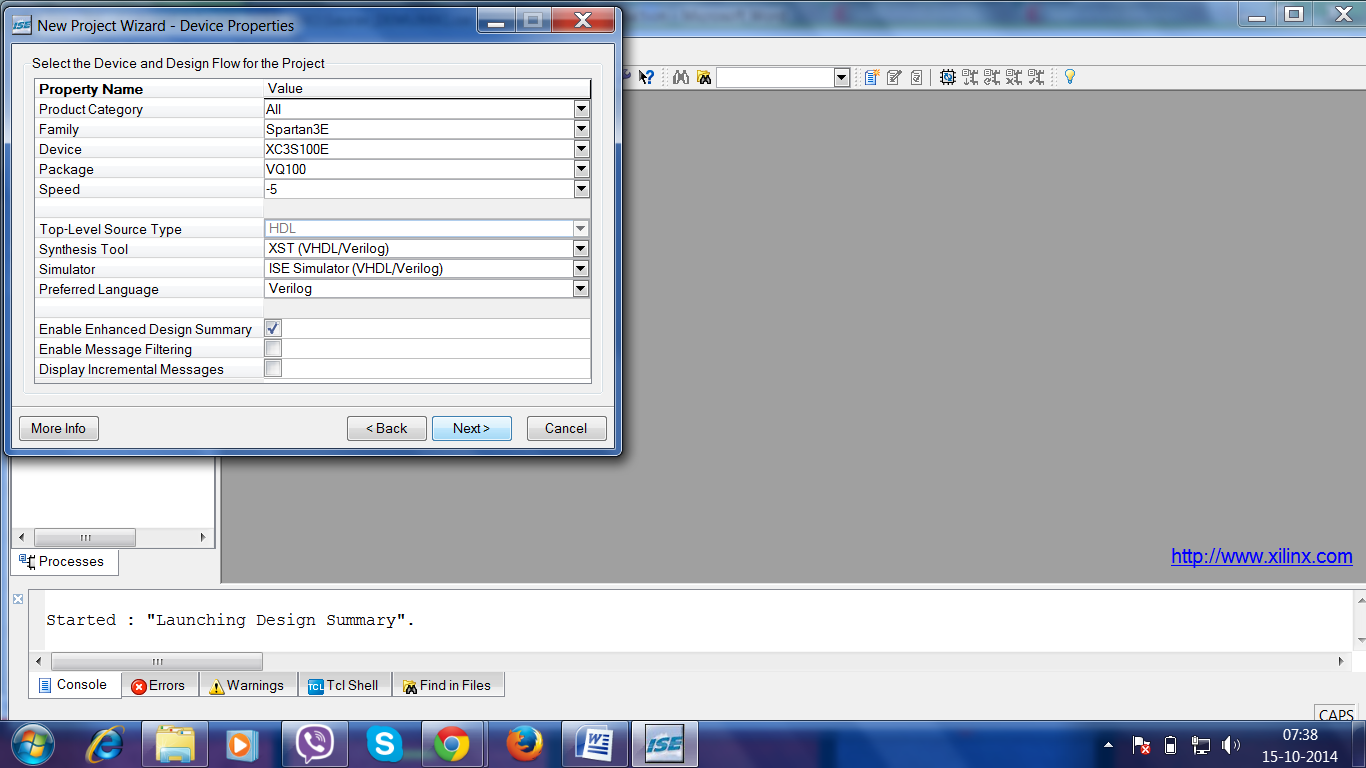
## Step-1- Create a project

In this section, you will create a new ISE project. A project is a collection of all files necessary to create and to download a design to a selected FPGA or CPLD device.

1. Select **File > New Project**. The New Project Wizard appears.
2. First, enter a location (directory path) for the new project, and then give a name for the project. For example, we name it **AND01**.



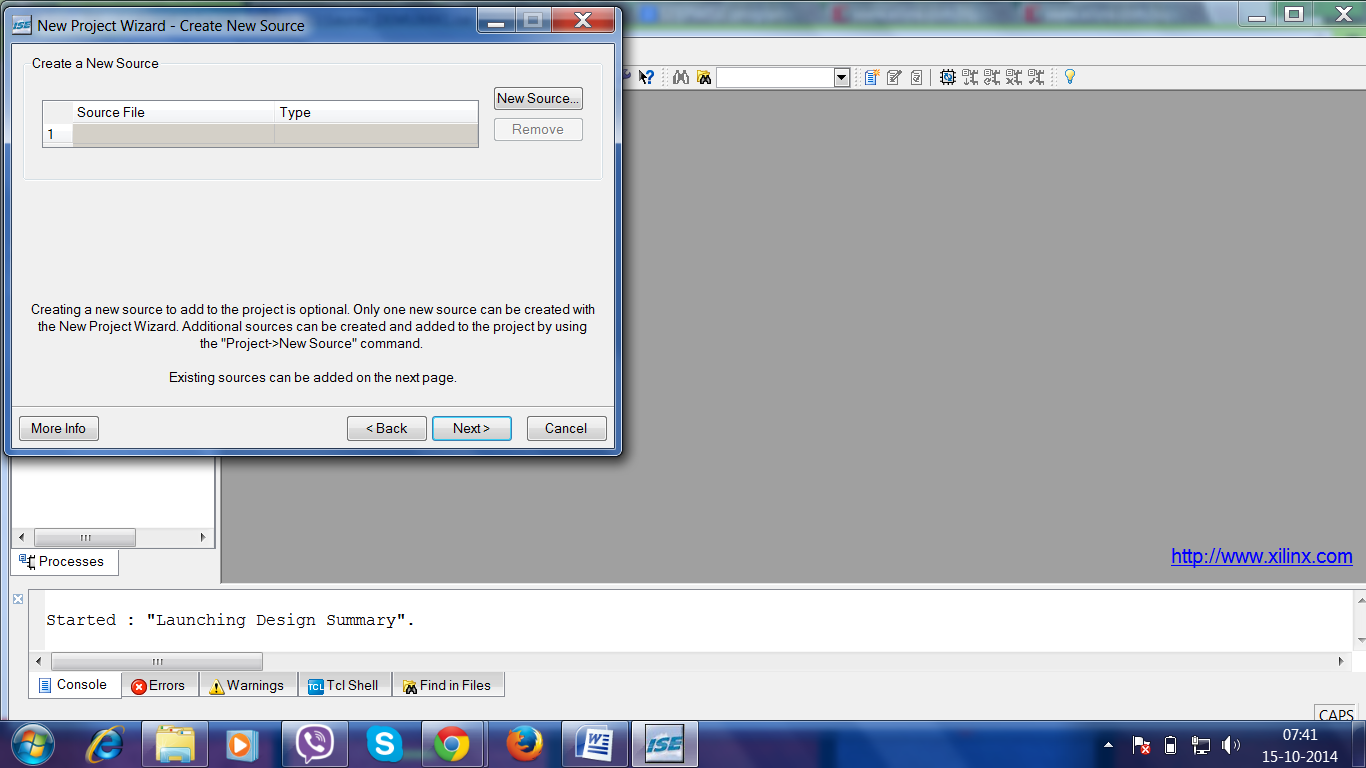
1. Select **HDL** from the Top-Level Module Type list, indicating that the top-level file in your project will be HDL, rather than Schematic or other stuffs.
2. Click on **Next** to move to the project properties page.
3. Fill in the properties in the table as shown below:

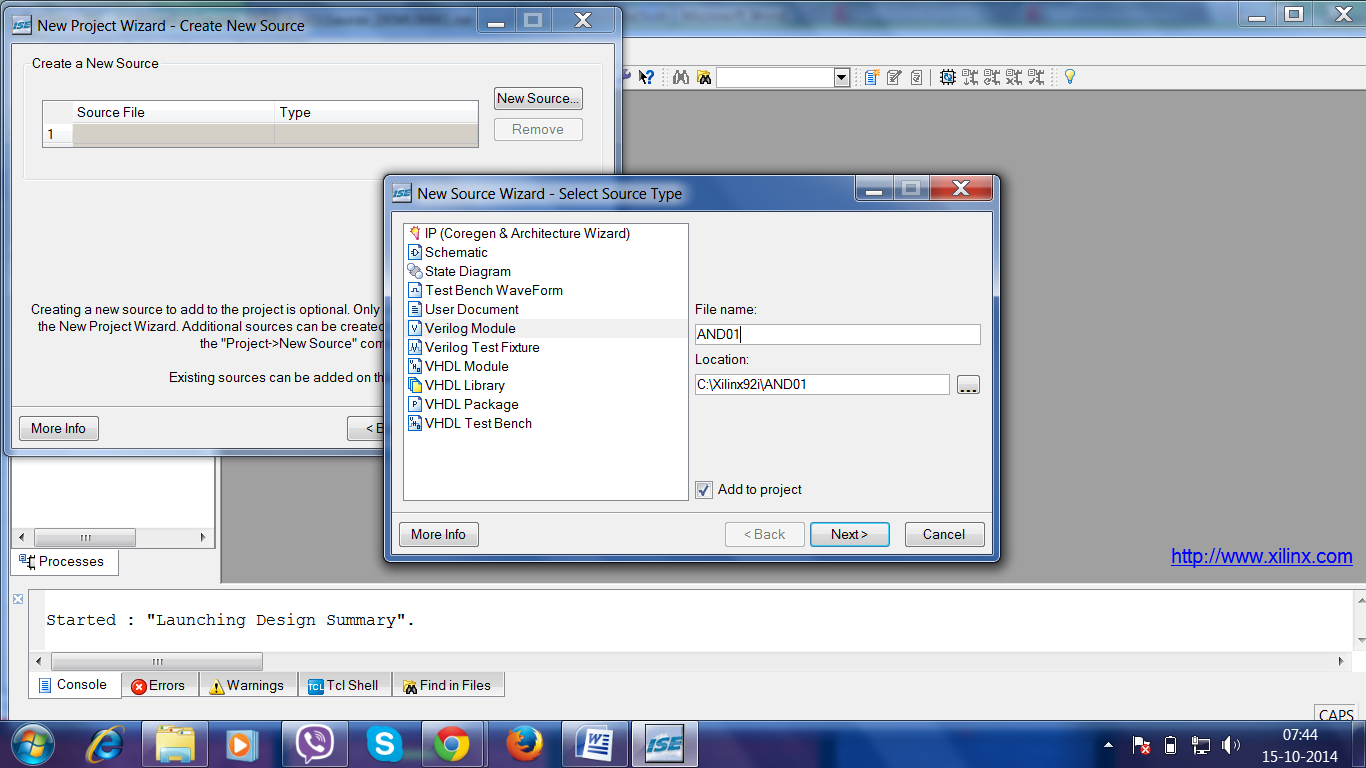
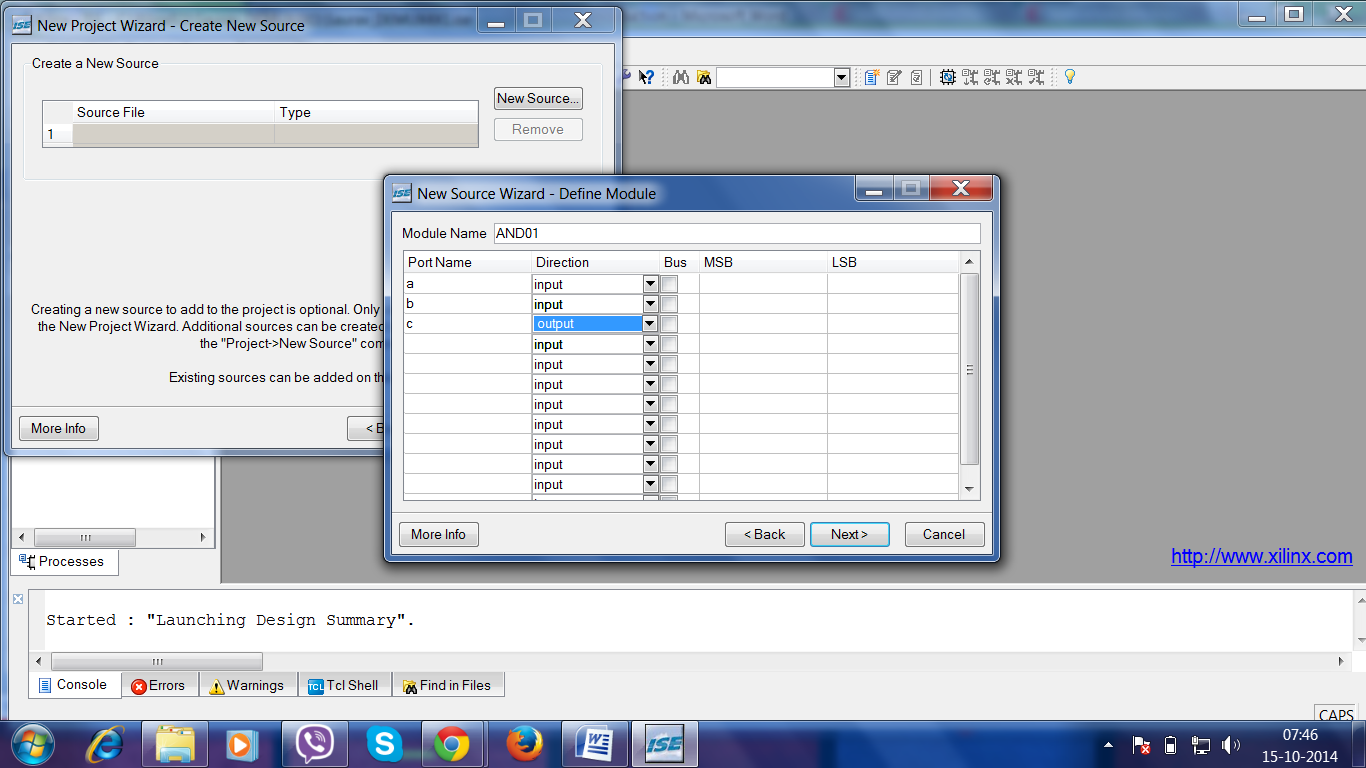


1. Click **Next** to proceed to the Create New Source window in the New Project Wizard. At the end of the next section, your new project will be created.

**Step-2-Create VHDL Source**

In this section, you will create a top-level HDL file for your design. You are going to design an up-down counter which is the same as what you did in the previous lab.

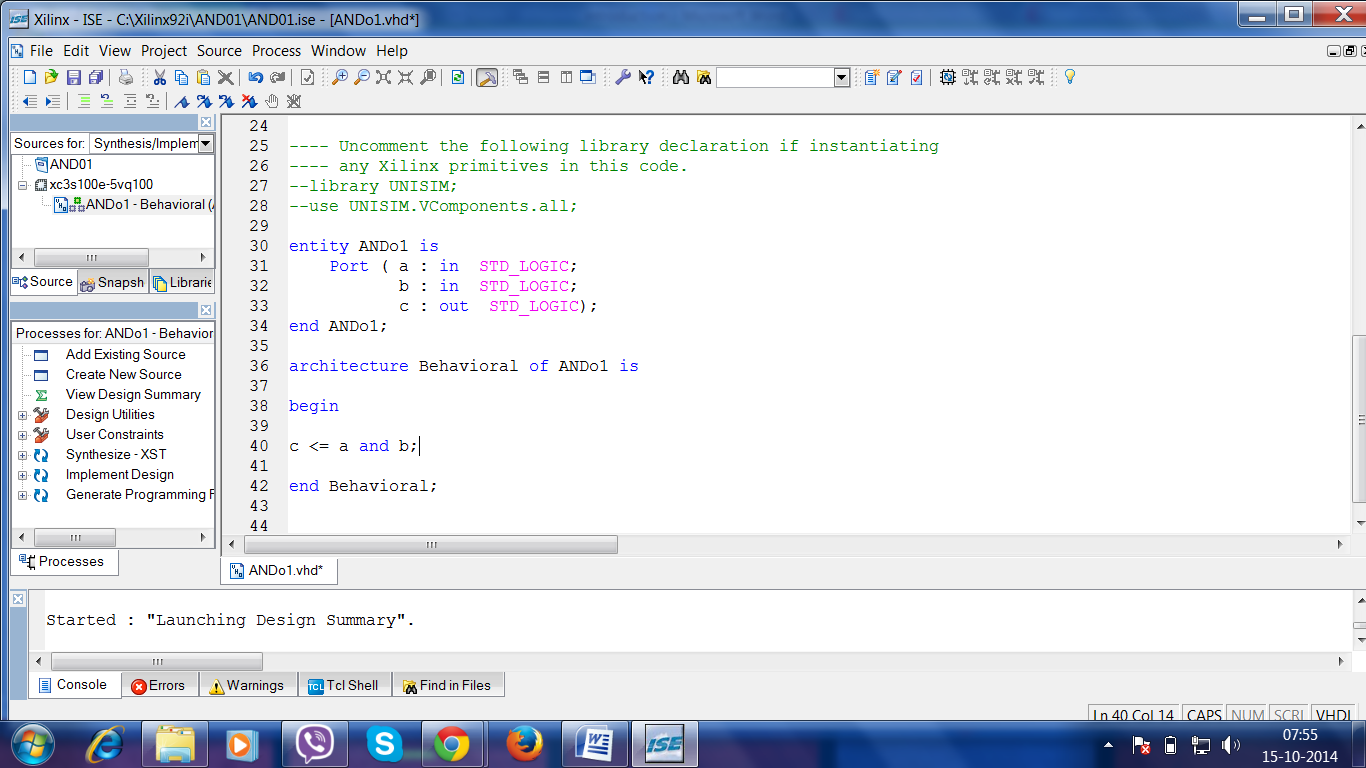
1. Click **New Source** in the New Project Wizard to add to one new source to your project.
2. Type in the file name **AND01**.
3. Select **VHDL Module** as the source type in the New Source Dialog box.
4. Verify that the **Add to Project** checkbox is selected.
5. Click **Next**.
6. Define the ports for your VHDL source.



1. Click **Next** in the Define Module dialog box.
2. Click **Finish** in the New Project Wizard - Summary dialog box to complete the new source file template.
3. If the following window appears, click on "yes".
4. Click **Next** in the New Project Wizard.
5. Click **Next** again.
6. Click **Finish** in the New Project Wizard - Project Summary dialog box.   
   ISE creates and displays the new project in the Source in Project window and adds the **AND01.vhd** file to the project.

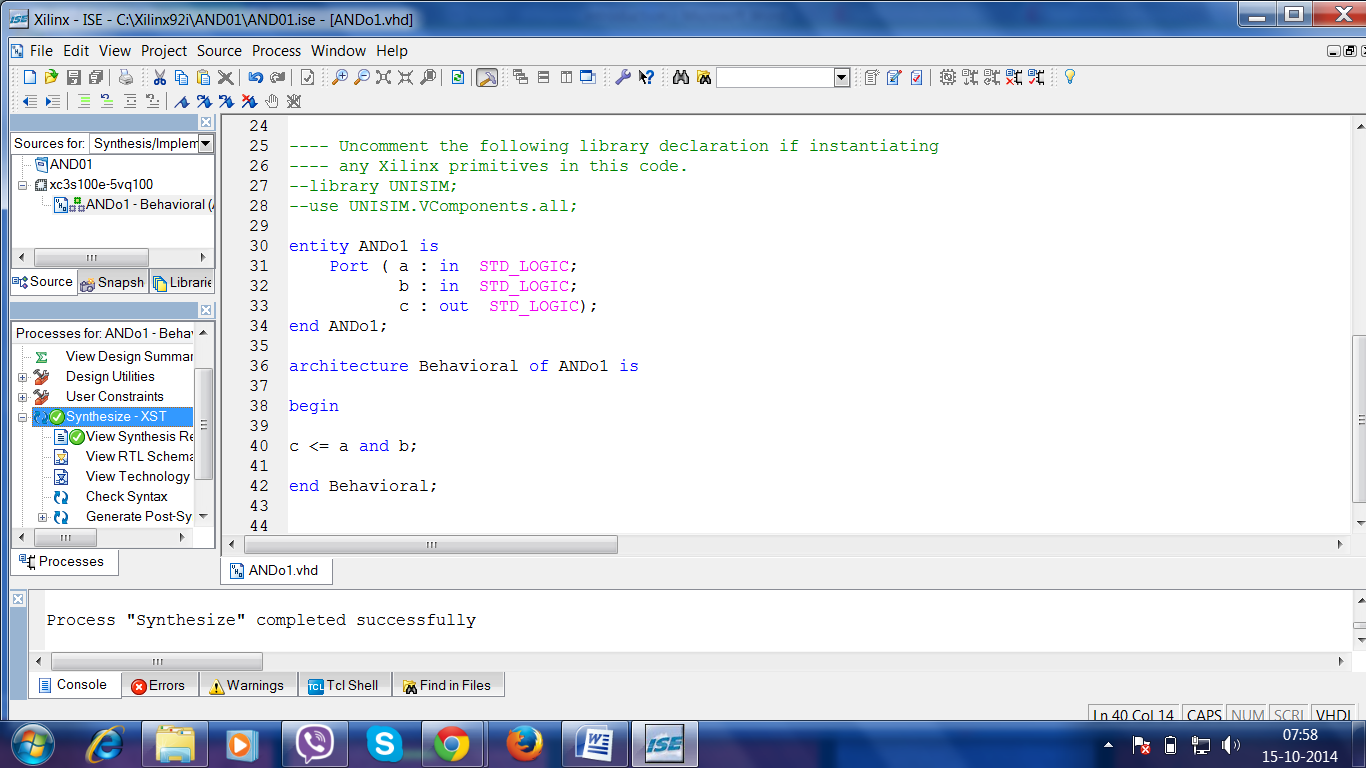
**Step-3-Enter and Edit VHDL Code**

1. Save the file by selecting **File -> Save**. When you are finished, the code for the AND01 should look like the following.



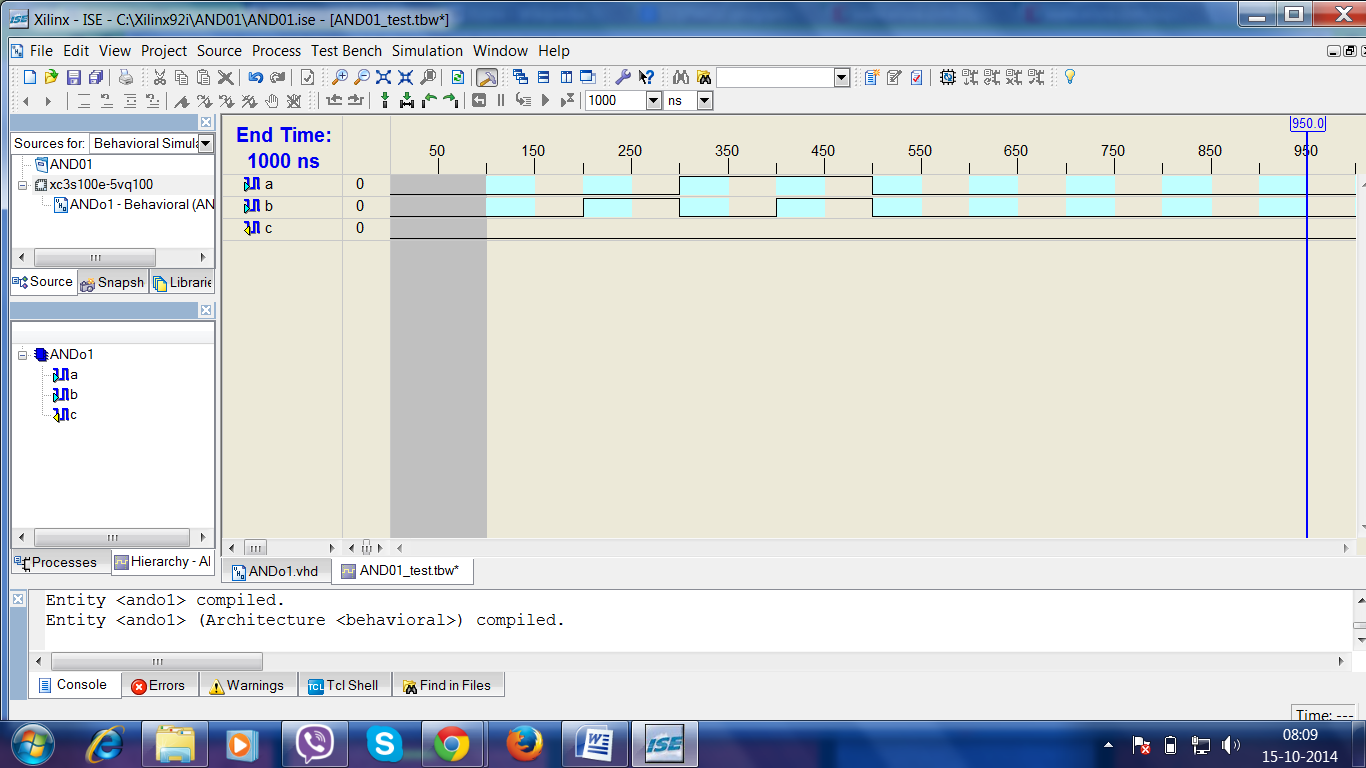
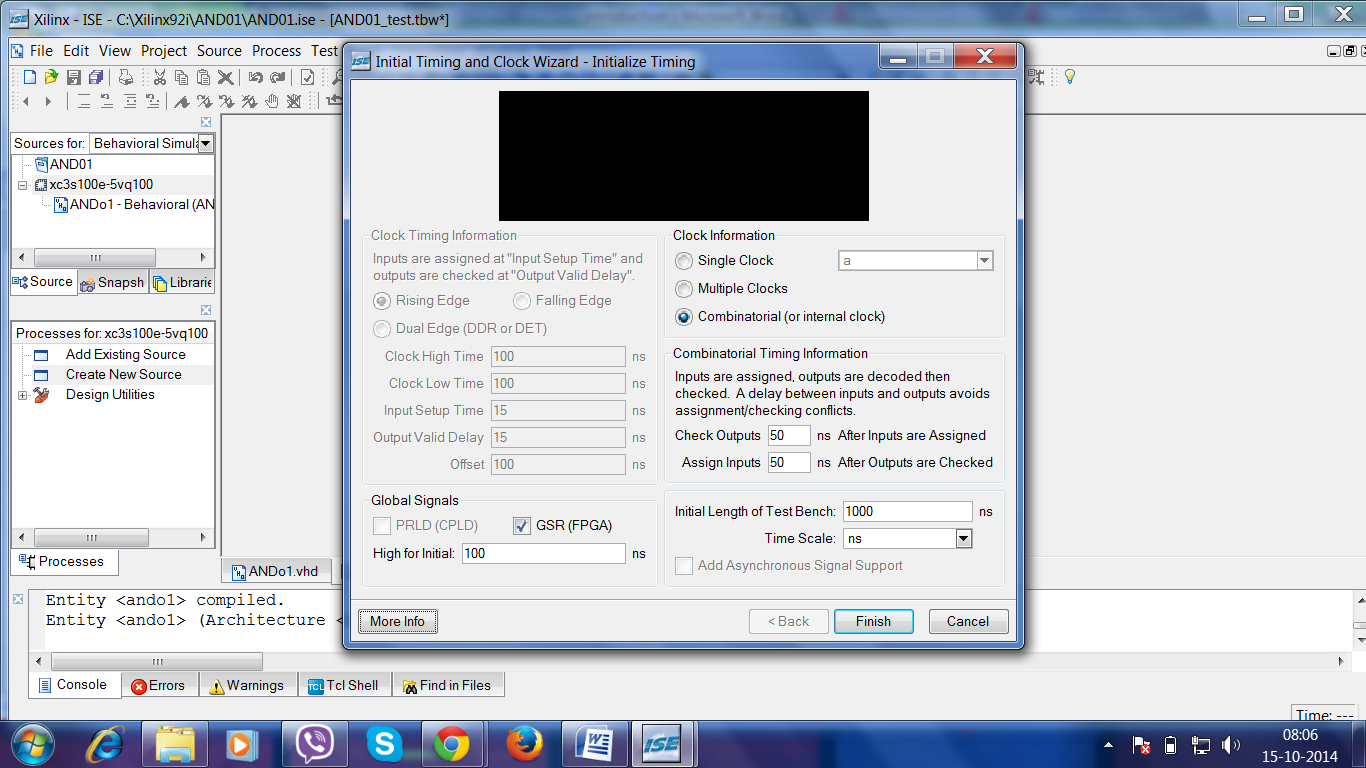
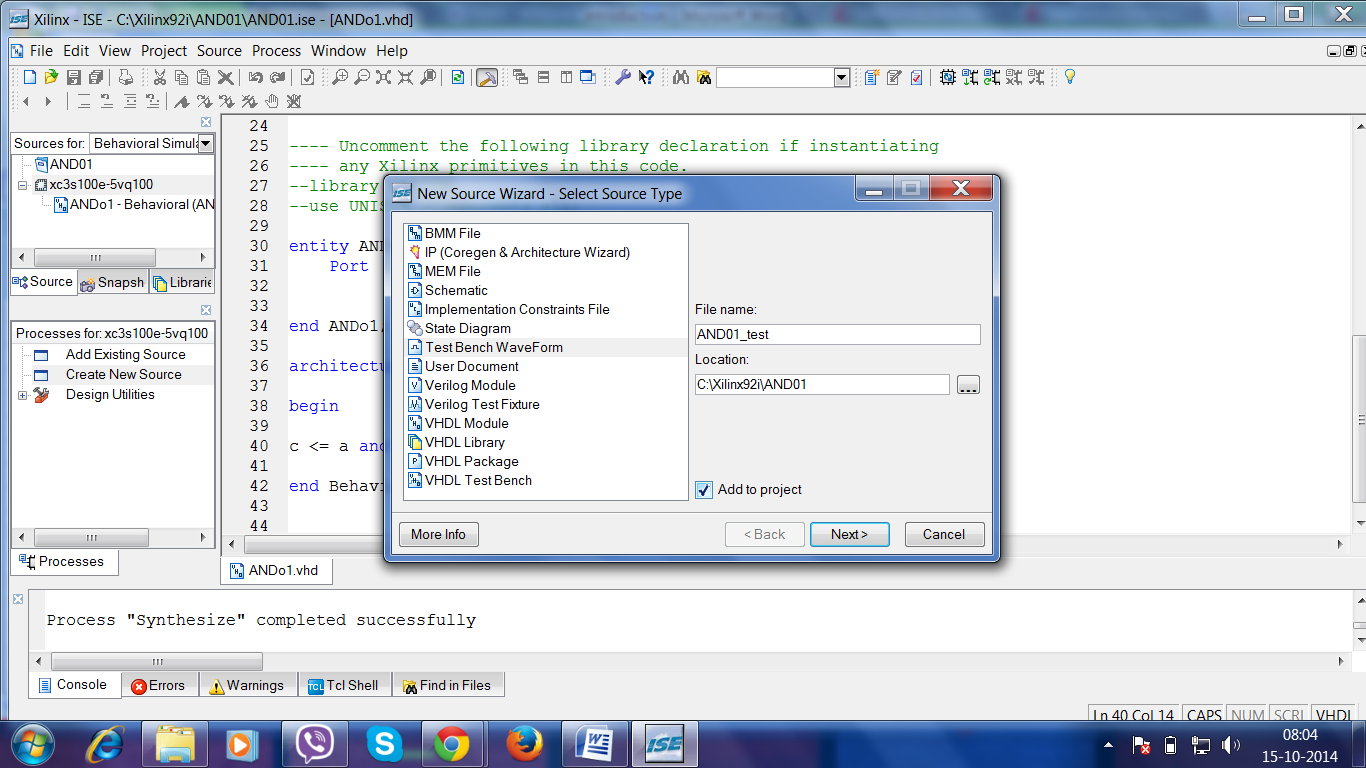
**Step-4-Check the Syntax of your VHDL source - Synthesize Your Code**

When source files are complete, the next step is to check the syntax of the design. Syntax errors and typos can be found using this step.

1. Click **+** next to the Synthesize-XST process to expand the hierarchy.
2. Double-click on the **Synthesize -XST** process.

**Step-5-Create a Test Bench for Simulation**

In this section, you will create a test bench waveform containing input stimulus you can use to simulate the counter module. This test bench waveform is a graphical view of a test bench. It is used with simulator to verify that the counter design meets both behavioral and timing design requirements. You will use the waveform editor to create a test bench waveform (TBW) file.

1. Select the **counter** HDL file in the Sources in Project window.
2. Create a new source by selecting **project -> New Source**.
3. In the New Source window, select **Test Bench Waveform** as the source type, and type **AND01\_test** in the File Name field.
4. Make sure the box for Add to Project is checked.
5. Click **Next**.
6. The source File dialog box shows that you are associating the test bench with the source file: counter. Click **Next**.
7. Click **Finish**.   
   You need to set the initial values for test bench waveform in the Initialize Timing dialog box before the test bench waveform editing window opens.
8.  Select **File -> Save** to save the waveform.
9. Select the **Behavioral Simulation**in the Source window.
10. On the Sources in Project Window, the TBW file AND01\_test**.tbw** is automatically added to your project.

### Step-6-Simulating Behavioral Model (ISE Simulator)

To run the integrated simulation process in ISE:

1. Select the **AND01\_test** waveform in the Sources in Project window. You can see Xilinx ISE Simulator processes in the Processes for Source window.
2. Double-click on the **Simulate Behavioral Model**process in the Project window. The ISE Simulator opens and run the simulation to the end of the test bench.

