Mehran University of Engineering & Technology Jamshoro

**Institute of Information and Communication Technologies**

**Lab # 02**

**Design Combinational Logic Full Adder Circuit, Simulate & Synthesis Using Xilinx ISE**

|  |
| --- |
| Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ I.D No\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Score\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date\_\_\_\_\_\_\_\_\_\_ |

**OBJECTIVE:**

* Design, Simulation, Synthesis and Implementation of the full Adder using Xilinx ISE Design Suite 12.3 Software and develop concepts with design entry using VHDL.
* Simulation of Combinational Logic Full Adder using ISIM Simulator.
* Implementation of Combinational Logic Full Adder with **NEXYS2** Spartan 3E Kit using Xilinx ISE & Adept Software.

**REQUIREMENTS:**

* PC with Windows XP/2007 Operating System.
* Xilinx ISE Design Suite 12.3 Software installed.
* Digilent **Adept** Software.
* **NEXYS2** Spartan 3E Kit

**Digital Circuits:**

Digital circuits are divided into two parts one is called combinational logic and another is called sequential logic circuits. Combinational logic circuits implement Boolean functions. Boolean functions are mappings of input bit-strings to output bit-strings. These circuits are functions of input only. Sequential circuits are basically combinational circuits with the additional properties of storage (to remember past inputs) and feedback figure-1 represents the functional block diagram of sequential logic.

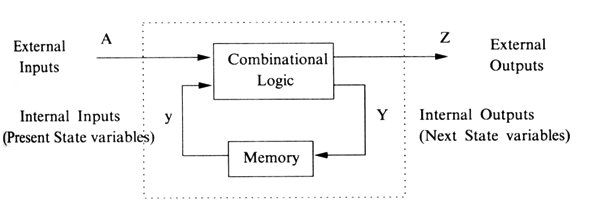


Figure-1: Block Diagram of Sequential Logic

We're going to construct combinational logic circuits that perform binary addition. We need a circuit that can add three bits. That circuit is called a *full adder*. In full adder circuit having three inputs a, b, cin and two outputs sum and cout. The circuit diagram of full adder is shown in figure-2 and complete operation is available in truth-table.





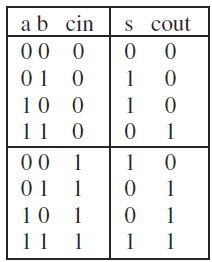
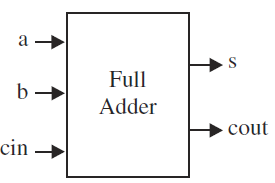


Table-1 Truth-Table of Full Adder

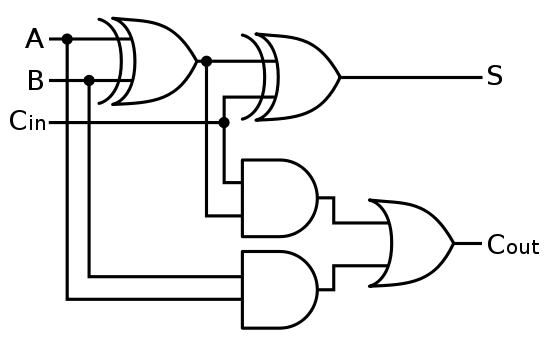
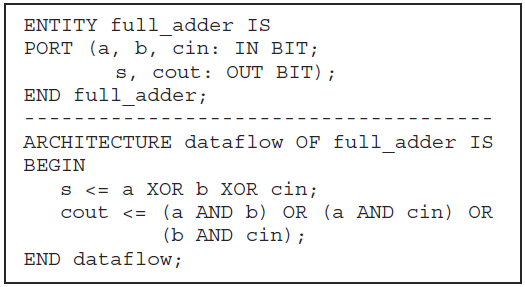


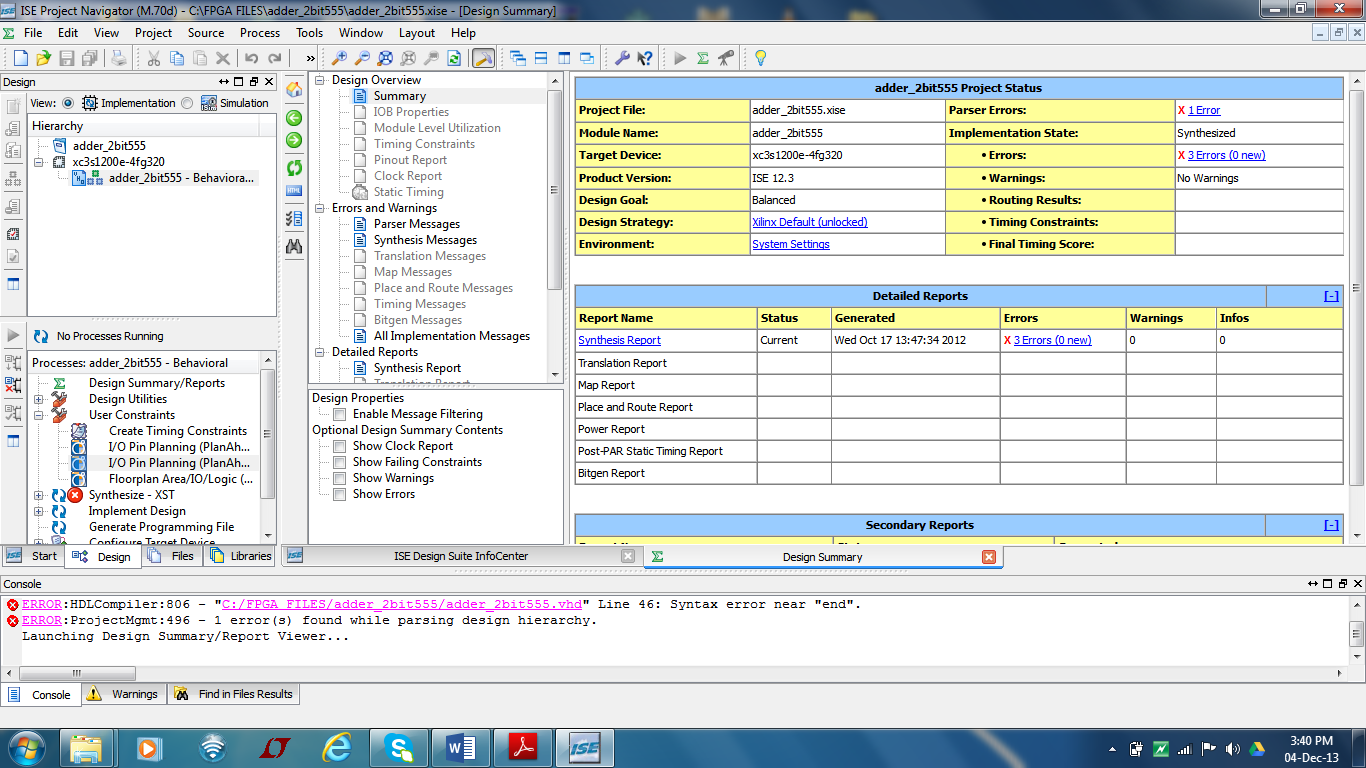
Figure-2 Circuit Diagram of Full Adder.



**Designing Procedure:**

**STEP1: DESIGN ENTRY**

1. Invoke Xilinx ISE Design Suite 12.3 Software. Double click the Xilinx ISE Design Suite icon present on your desktop, Check with YOUR instructor if not available. A screen like that of **figure 3** will be displayed.



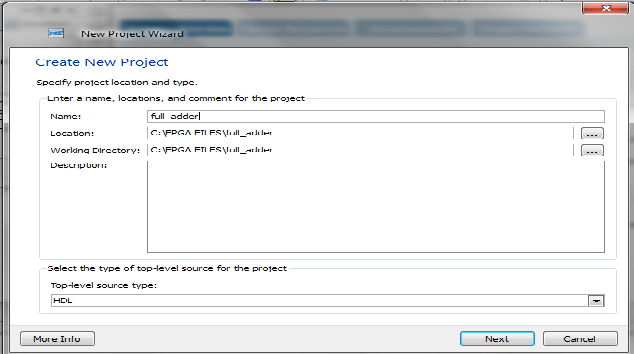
**Figure-3:** Xilinx ISE Design Suite **Project Navigator**

1. **Creating a New Project**

To create a new project using the New Project Wizard, do the following:

From Project Navigator, select **File > New Project**.

The New Project Wizard appears, as shown in **figure-4**

****

**Figure-4*:*** New Project Wizard**—Create New Project**

**3.** Enter your desired Project name in the Name field.

**4.** In the Location field, browse to the directory in which you installed the project.

**5.** Verify that **HDL** is selected as the Top-Level Source Type, and click **Next**.

Choose the following settings Figure-5 as shown below:

Product Category: **All**

Family: **Spartan3E**

Device: **XC3S500E**

Package: **FG320**

Speed Grade: **-4**

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**

VHDL Source Analysis Standard: **VHDL-93**

These settings can be done any time, but if settled once will not change automatically until user disturbances.

After Fulfill the Setting Click **Next,** the following Project summary window (Fig 2.5) will appear.

**6.** Click **Finish** Figure-6 to complete the project creation and begin the design process.

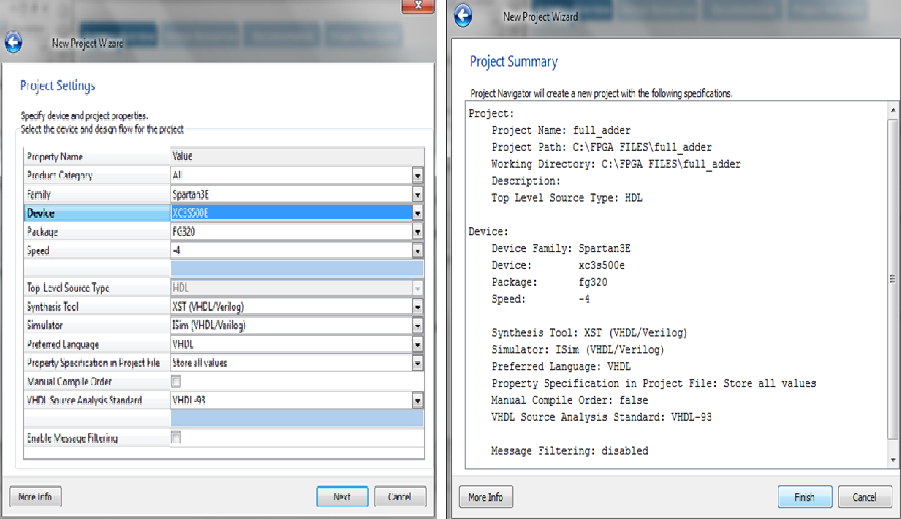


Figure-5-6 New Project Wizard Setting/Summary

**7. Creating a New Source file**; Go to **Project** \_ **New Source**, select **VHDL Module** and Enter the name “**full adder**”.

**8**. Enter the ports for your module. This is strictly optional, and saves you the typing of creating the VHDL entity manually. Whether you fill this out or not, the resultant text file will be totally under your control for later editing.

**9**. After enter the ports click **Next,** the following summary window Figure-7 will appear.

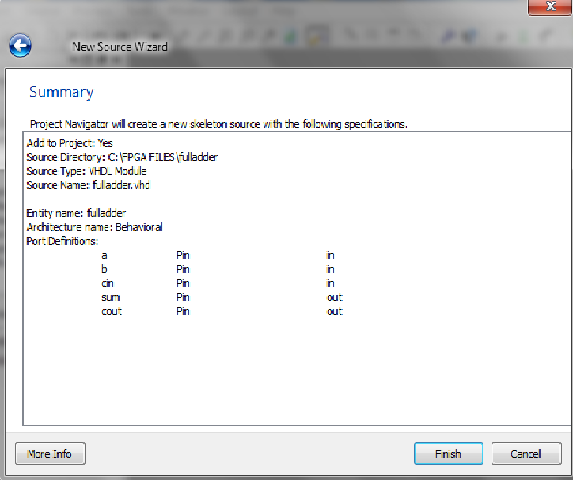


Figure-7 New Source Wizard- Summary

**10.** Click **Finish** Figure-8 to open the empty VHDL file in the ISE Text Editor Figure-8. In the ISE Text Editor, the ports are already declared in the VHDL file, and some of the basic file structure is already in place. Keywords are displayed in blue, comments in green, and values are black. The file is color-coded to enhance readability and help you recognize typographical errors.

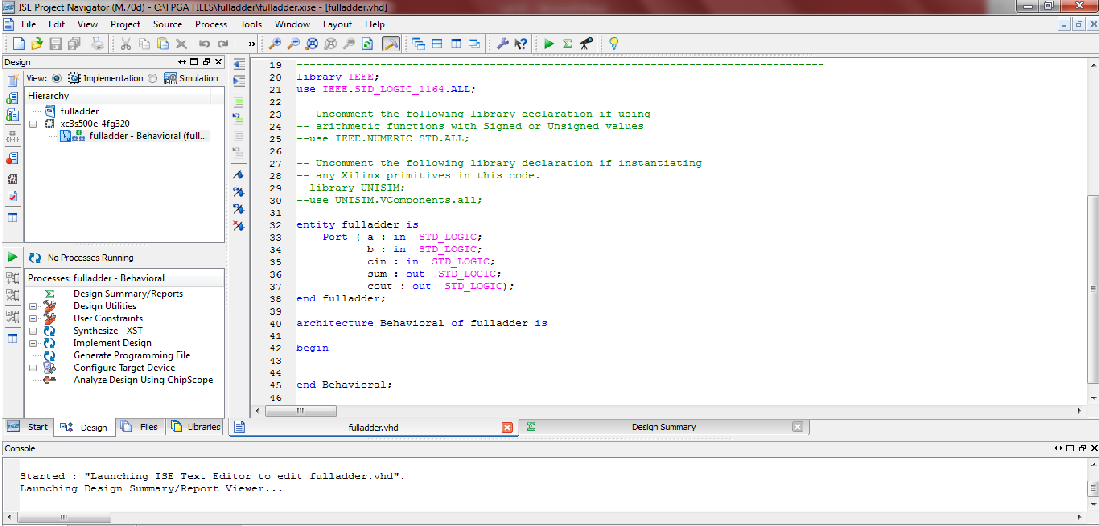


Figure-8 **VHDL File in ISE Text Editor**

**Step No.2 Simulation:**

Simulation is the process of testing the logical or processing functionality of designed logic. Several kinds of Software and hardware Tools to provide this functionality. With Xilinx ISim Simulator you can simulate your design by writing “Test benches” and/or by assigning Manual wave flows to inputs and check functionality of outputs after simulation.

The Xilinx® ISE® Design Suite provides an integrated flow with the Mentor ModelSim simulator and the Xilinx ISim simulator that allows simulations to be run from the Xilinx Project Navigator.

**Behavioral Simulation:**

After Creating New project and New source File now it is ready to simulate our design.to simulate the Design follow the following steps.

In the Design Panel, select the **Simulation** radio button. The Simulation type drop-down list appears.

Select gate\_logic\_Behavioral file **figure-9** and double click on Simulator Behavioral Model in the Process window, following ISim window **Figure-10** will appear.

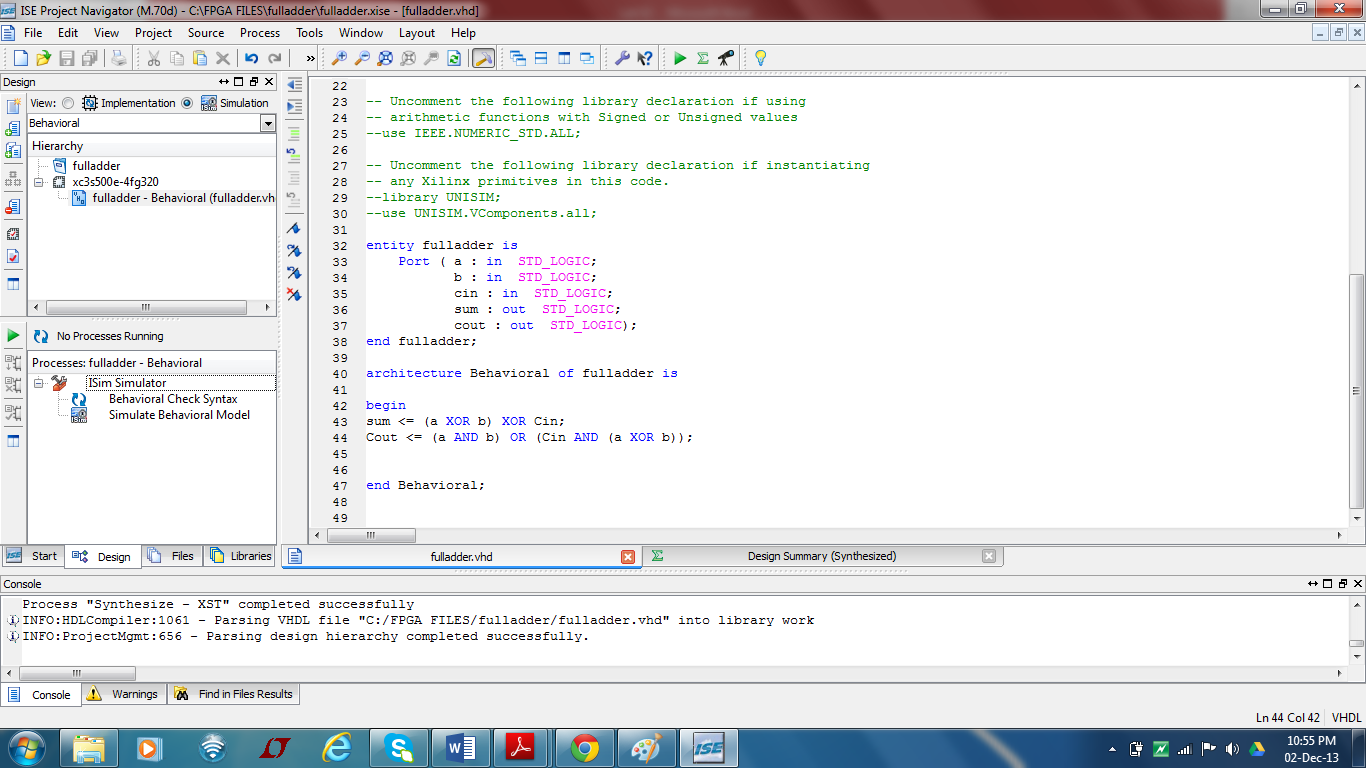


Figure-9 Behavioral Simulation Process

Some features of this window include:

1. A Source Files panel where source files to be viewed can be selected

2. An Objects panel where different signals can be added to the simulation

3. A simulation panel where the state of signals can be observed

4. A Console panel

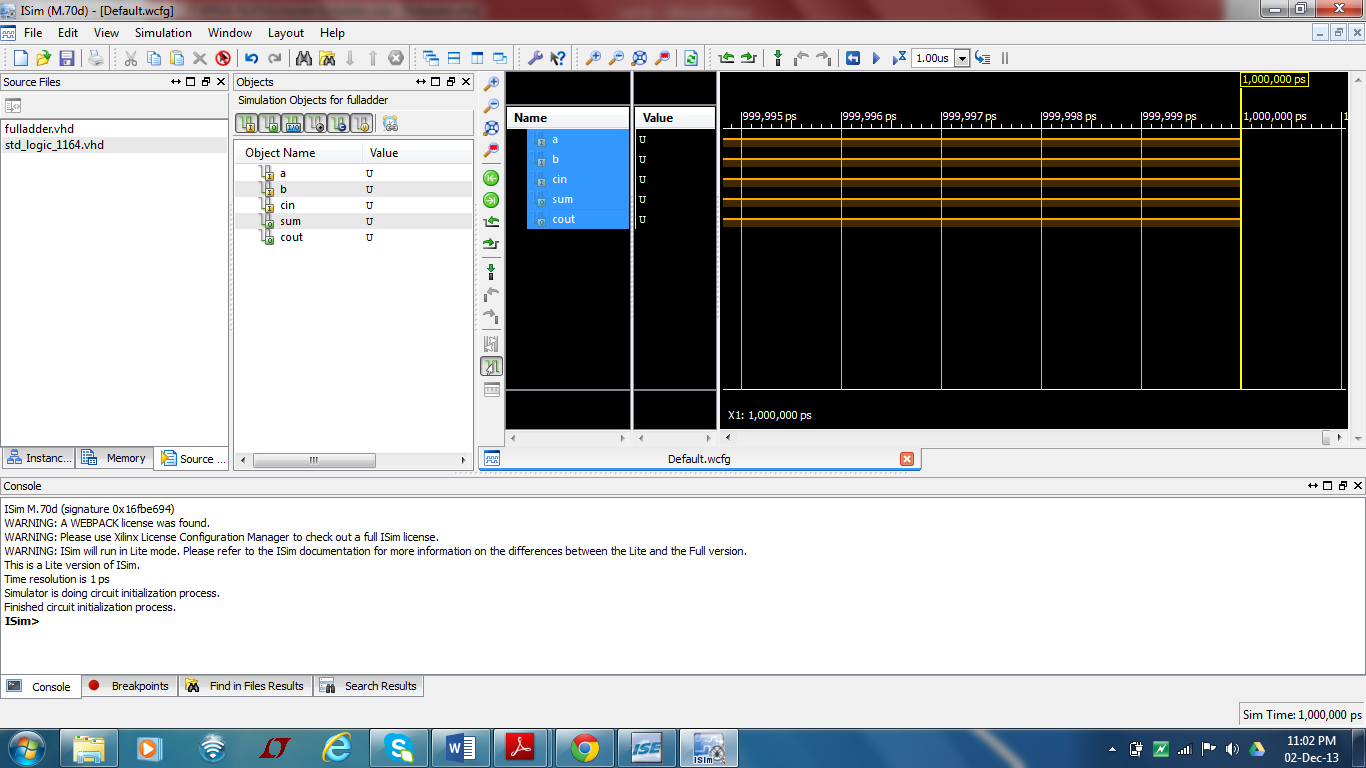


Figure-10 ISim Window

Select the Simulation from the menu bar and click on Restart as shown in following **figure-11**. After click on Restart window.

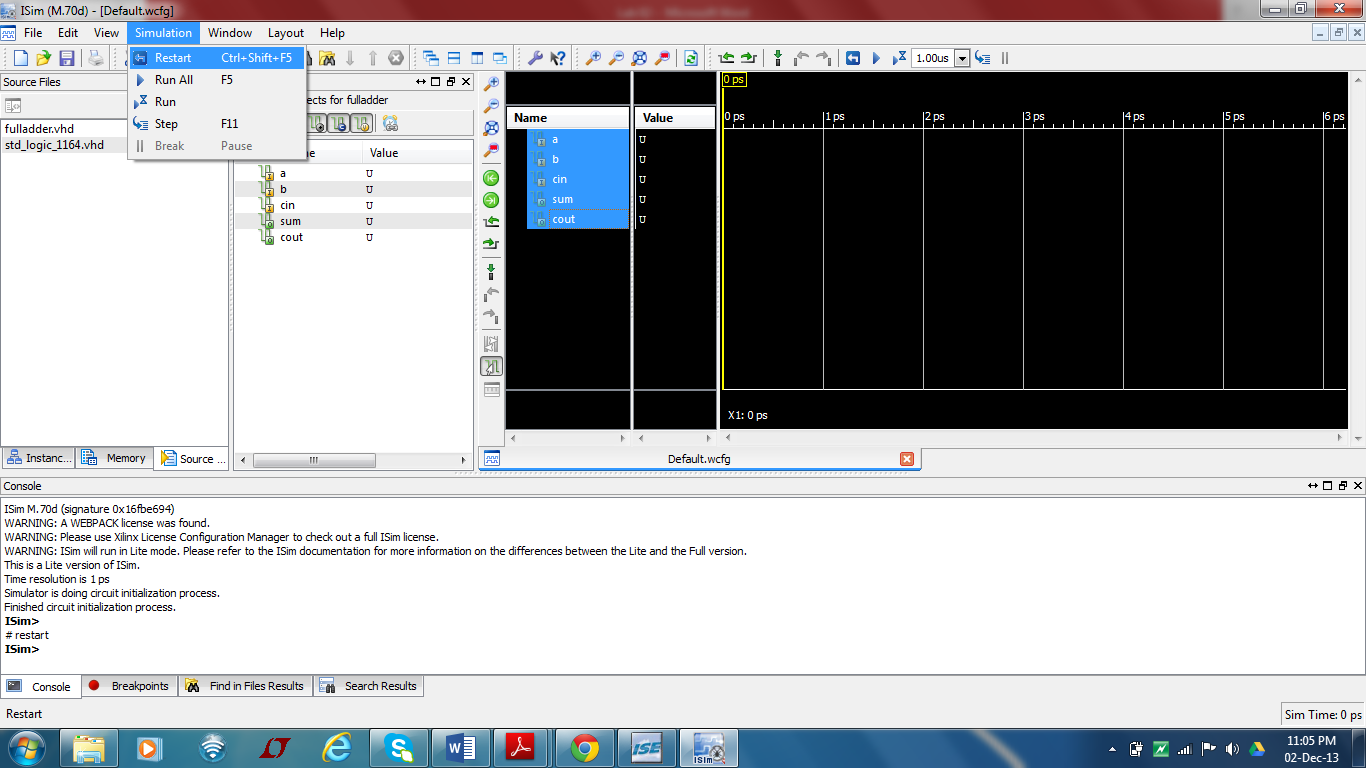
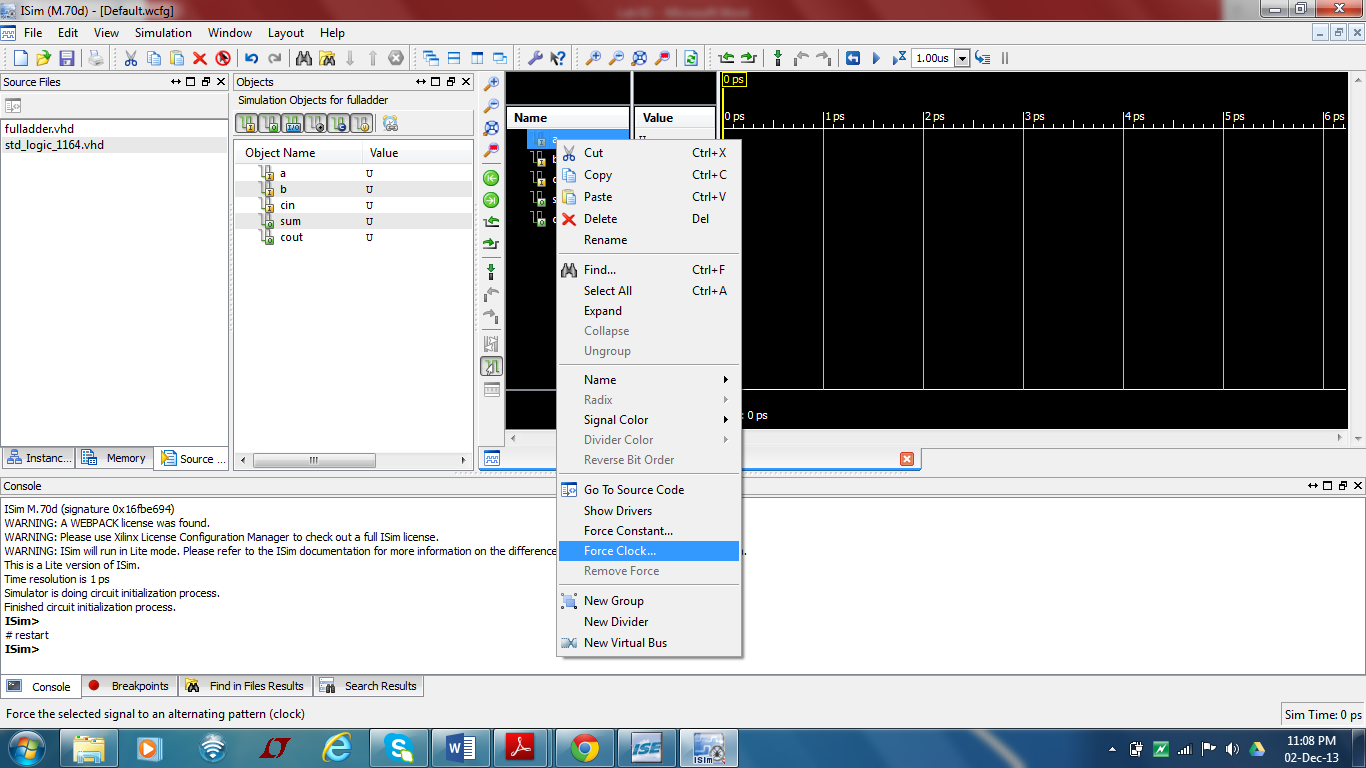


Figure-11 Simulation Process-**After Restart**

To force the input signal (a, b & Cin) to an alternating pattern (clock), Right click on the input port “**a**” **figure-12** and select the Force clock.



**Figure-12** Simulation Process-**Assigning clock input**

Enter the following parameter **figure-13** . click on apply and OK.

Leading Edge Value: **1/0**

Trailing Edge Value: **1/0**

Starting at the time offset: 0

Cancel after the time offset: **100, 200, …**

Duty cycle (%): **10, 20, 30 ,……**

Period: **100**

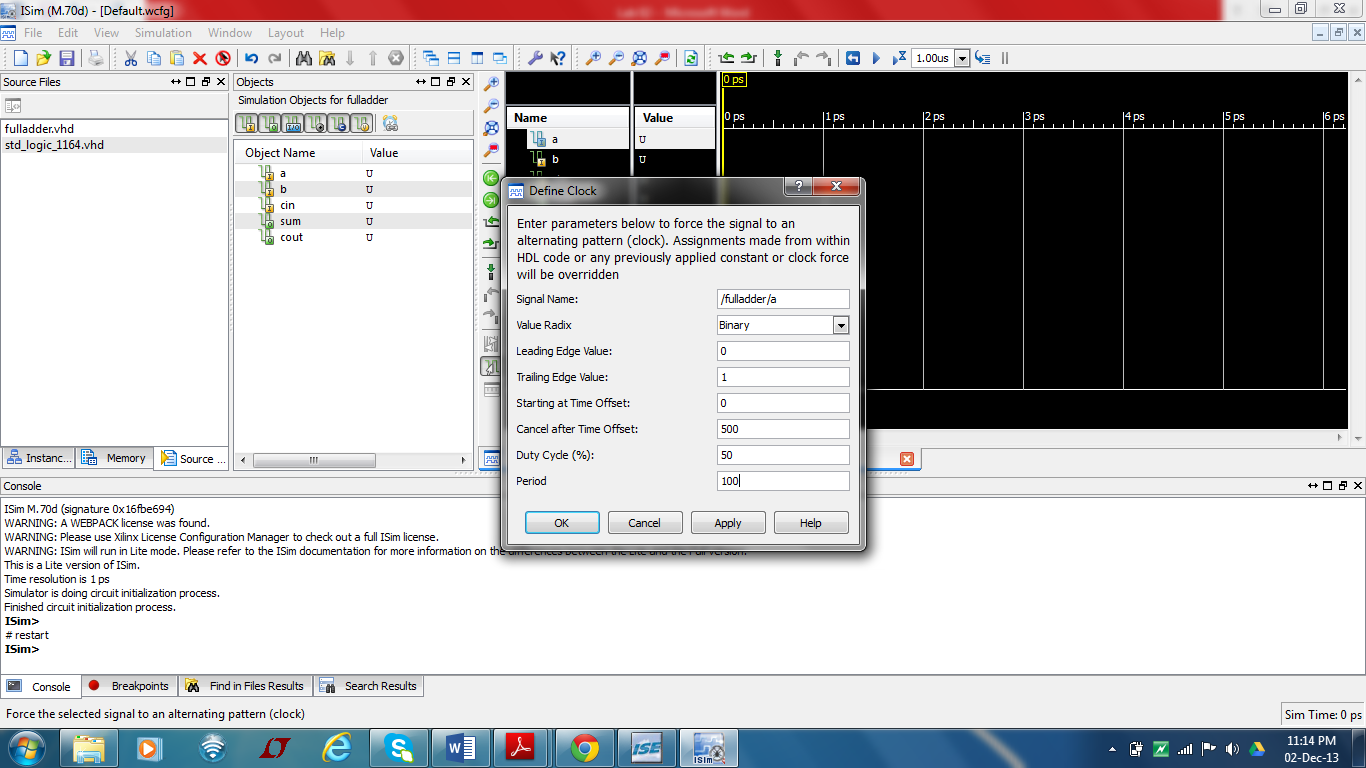
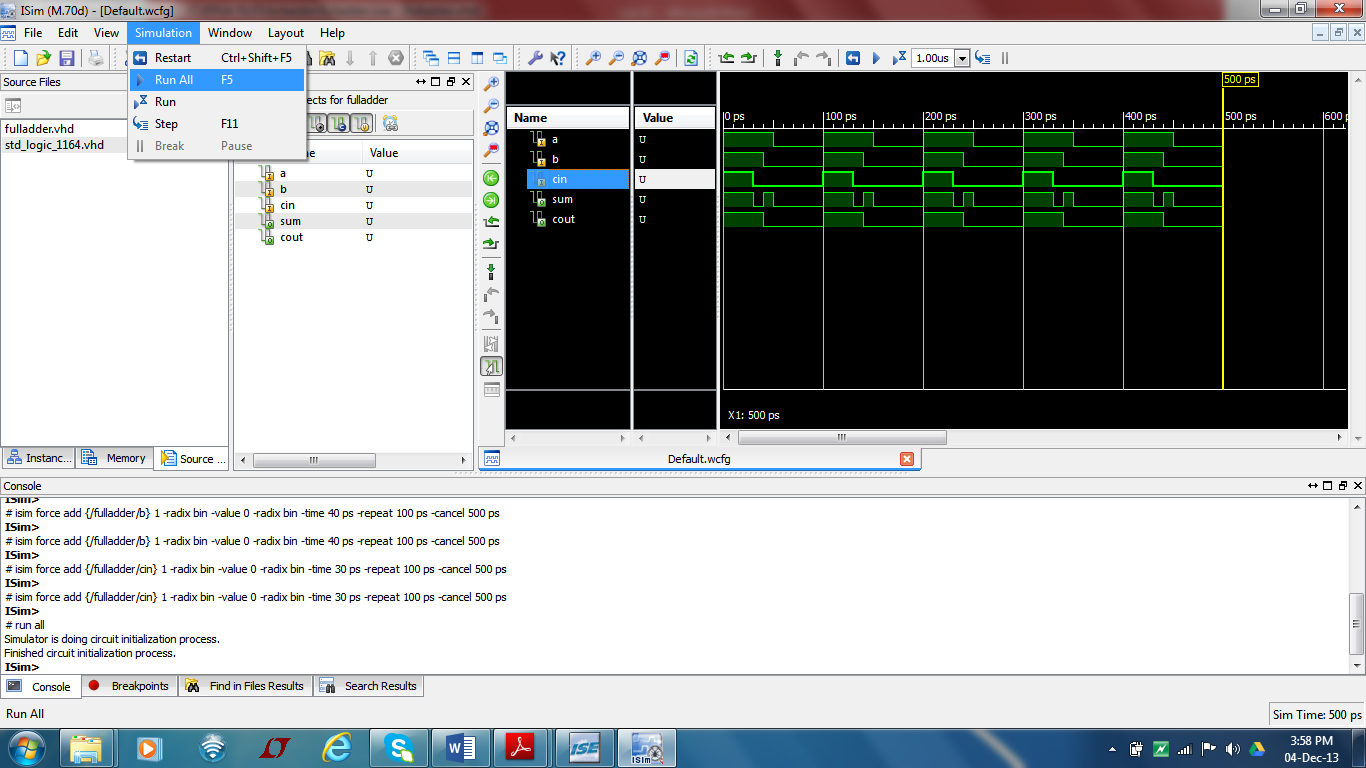


Figure-13 Force the Signal- **Enter Parameter**

Similarly Right click on the input port “**b**” and “**Cin”**

Select the Simulation from the menu bar and click on Run as shown in following **figure 14**. After click on Run following waveform window will appear.



**Figure-14** Simulation Result for Full adder

**Synthesis and implementation:**

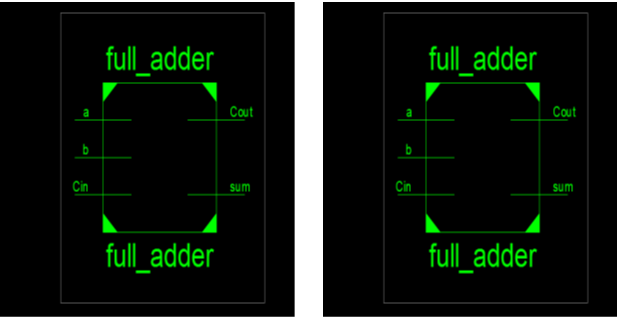
The design has to be synthesized and implemented before it can be checked for correctness, by running functional simulation or downloaded onto the prototyping board. With the top-level VHDL file opened (can be done by double-clicking that file) in the HDL editor window in the right half of the Project Navigator, and the view of the project being in the **Module view** , the **implement design** option can be seen in the **process view**. **Design entry utilities** and **Generate Programming File** options can also be seen in the process view. The former can be used to include user constraints, if any.

**Step 3: Synthesis**

**1.** To synthesize the design, double click on the **Synthesize Design** option in the **Processes window.**

2. Now the schematic diagram of the Full adder can be viewed by double clicking View RTL Schematic under Synthesize-XST menu in the Process Window, **(Figure 15).**

**3.** Similarly you can view the Technology Schematic Symbol by double clicking **View Technology Schematic** under Synthesize-XST menu in the Process Window, **(Figure 16).**

****

**Fig: 16** Schematic diagram for Full adder **Fig: 17** Technology Schematic Symbol for Full adder

**Step 4: Implementation**

**1.** Before implement the design you must create the User Constraint File **UCF**.

**2.** After creating the UCF file Single Click on full\_adder.ucf file from within Project Navigator, and then Select “Edit Constraints (Text)” from the Process window.

**3.** Assigning the pins to inputs and outputs of Full adder as following and save Ucf file.

**NET "a" LOC = "G18";**

**NET "b" LOC = "H18";**

**NET "Cin" LOC = "K18";**

**NET "sum” LOC = "J14";**

**NET "Cout" LOC = "J15";**

**4.** After assigning the Pins double click on “Implement Design” option in the **Processes window**. It will go through steps like **Translate, Map and Place & Route.**

**5.** Now create a programming file (bit stream file) of the design. This is done by clicking once on your top-level design in the Sources Pane, followed by a double click on “**Generate Programming File**” in the process window.

**6.** Once the programming file (bit stream file) is generated, the file has to be downloaded to the **NEXYS2 Spartan3E** device, using Digilent Adept Software.

**Review Question**

1. Select NEXYS2 Spartan3E FPGA design the 4-Bit Full adder.
2. Select NEXYS2 Spartan3E FPGA design the 4-Bit Magnitude Comparator. Attach simulation result and UCF coding for NEXYS 2 Board.