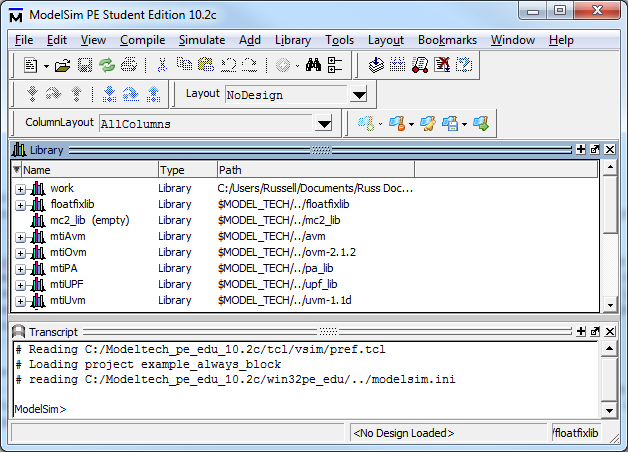
# LAB # 03: Simulation Process

## Lab Objective:

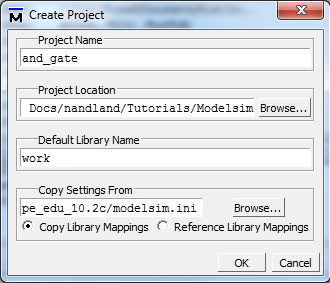
To understand and practice VERILOG syntax and simulation process.

## Lab Description:

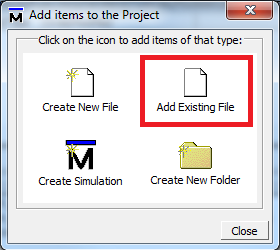
Modelsim is a program created by Mentor Graphics used for simulating your VHDL and Verilog designs. It is the most widely use simulation program in business and education. This tutorial explains first why simulation is important, then shows how you can acquire Modelsim Student Edition for free for your personal use.Let's open Modelsim. You are greeted with a window that looks like this



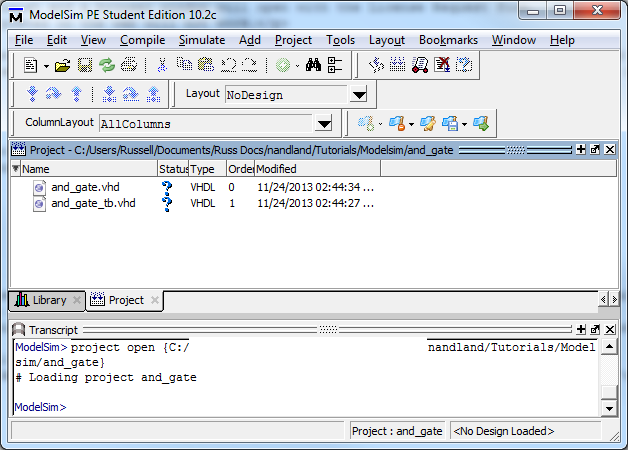
**Modelsim Main Window**



In order to run your simulation, you need to create a project. Click File -> New -> Project. You will see the window presented on the left. Choose a location for your new project and give it the name and\_gate. Projects in Modelsim have the file extension .prj. Leave the other settings to their default. This just says that all code will be compiled into the library "work".

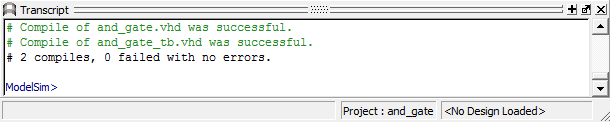


Click on Add Existing File as shown in the picture to the right. Navigate to the location where you downloaded and\_gate.vhd and and\_gate\_tb.vhd and add both of those to your project. Keep other settings at their default. Click OK when done.

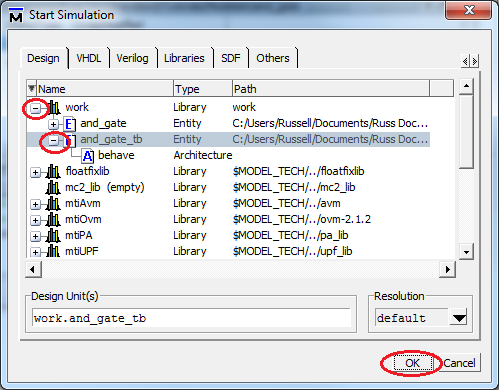


**Modelsim Project Window - Files Added to Project**

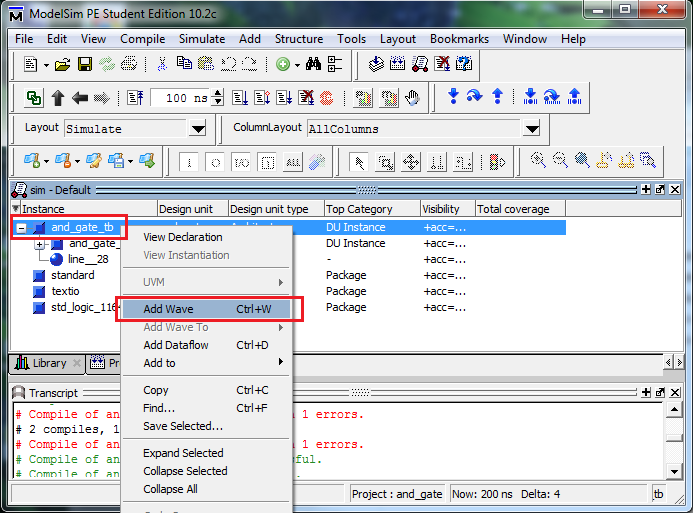
Notice now that the files have been added successfully to your project. See those two blue question marks in the Modelsim Project Window Figure above? That means that Modelsim has not compiled the files yet. You will need to compile the source files. To do this, right click on and\_gate.vhd, click on Compile, then click on Compile All. You should see messages in the Console window appear in green that the compile was successful as shown in the screenshot below.



**Results of a Successful Compile**

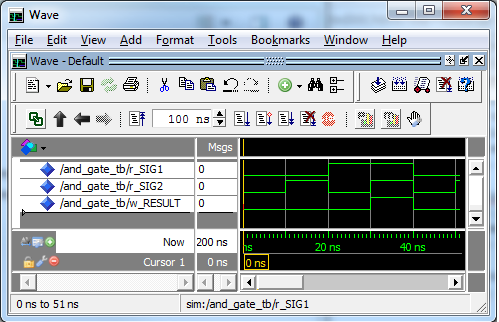


To start your simulation, click on Simulate in the Menu Bar, then click Start Simulation. This opens the Start Simulation Window. Click on the plus sign next to work, then click on the plus sign next to and\_gate\_tb. Make sure you select and\_gate\_tb and not and\_gate as we want to simulate the design at the test bench level. Once and\_gate\_tb is highlighted, click OK.



**Modelsim Simulation Window - Simulation Ready**

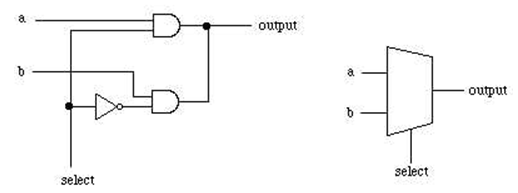
Almost there! The simulation is ready and waiting. Now, the majority of the time that you use Modelsim will be spent looking at the waveform view. The waveform view contains waves (binary 0's and 1's, hexadecimal digits, binary digits, enumerated types, etc) for all of the signals in your design. It shows how your module reacts to different stimulus. The next figure shows you what your waveform view looks like, but first you need to add some signals to monitor. In this example, we will monitor all of the signals in the test bench. To do this, **right click** on and\_gate\_tb in the sim window and click **Add Wave**. You can also click and drag signals to the waveform window from other windows in Modelsim.



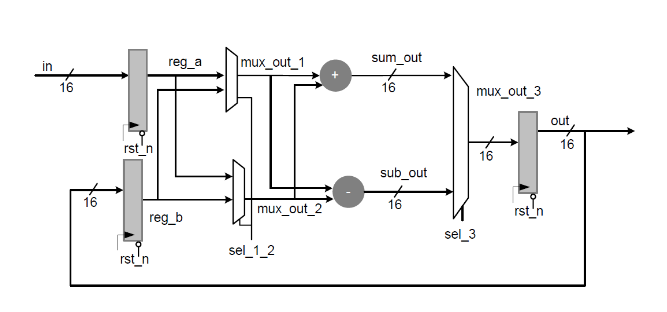
Here is your waveform window. All of the test bench signals have been added as signals you can monitor. To run the simulation, click the Icon with a little piece of paper and a down arrow next to the 100 ns time. This will run your simulation for 100 nanoseconds. Watch how the signals change!

## Lab Tasks:

1. For the following



1. Design a 2-to-1 Mux using as shown in above diagram
2. Using the Design Module of 2-to-1 Mux created in (a) design a 4-to-1 mux.
3. Again Using the Design Module of 2-to-1 Mux and 4-to-1 mux created in (a), (b) and Design 8-to-1 mux.

2. Write RTL VERILOG code for the following circuit. Test your design for all values of in. 

## Test Plans:

1. **DESIGN QUESTION:**

**You are required to design a system which counts the number of cycles required by the system until all (3 i.e. x, y, z) inputs of the system become equal. The system has 3 N bit wide inputs and a single M bit wide output which shows the number of cycles required by the system. The inputs must be stored in the registers before start of the process.**

1. **Write A Verilog Code To Swap Contents Of Two Registers With And Without A Temporary Register?**

## THINK!!

1. **Which of the following loops are supported by Verilog?**

* if-else loop
* for loop
* while loop
* All of the above

1. **What Is Meant By Inferring Latches,how To Avoid It?**
2. **In A Pure Combinational Circuit Is It Necessary To Mention All The Inputs In Sensitivity Disk? If Yes, Why?**