Lab no 01: Simulate Logic Gates

The purpose of this Lab is to learn how to simulate simple logic gates on ModelSim. You will install ModelSim Software and write AND and OR gates in Verilog Hardware Description Language (HDL). Then you will write a testbench to verify the functionality of the gates and check the output on Wave Window.

Parts: -

1. ModelSim Installation and New Project Steps
2. And Gate Verilog Code & Testbench.
3. OR Gate Verilog Code & Testbench.

Part 1. ModelSim Installation & New Project Steps

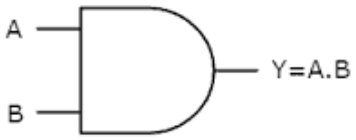
a. Download ModelSim (Here)

b. Install ModelSim (follow the steps Here)

c. Create a new project

Step	Description
1	**Open ModelSim**
2	**Create a new project** File----->>> New ------->>>> Project
3	**Give a title to the project**
4	**Add a new file** create new file ---->> New file To add existing file ----->> Existing file
5	**Define the name and the type of the file.** Create a filename with extension (.v) ex: and.v and select Verilog type in file list
6	**Edit the file to include your code.** In the project area, right-click on the file (and.v) and edit, now you can edit your program in the editor space
7	**Compile your code.** From ModelSim tabs, select "Compile" → "compile all"
8	**Simulate your design** From ModelSim tabs, select "Simulate" → "start Simulation"
9	**Verify design function using wave window** From ModelSim tabs, select "Add" → "To Wave" → "All items in Region"
10	**Run simulation** From ModelSim tabs, select "Simulate" → "Run 100"

Part 2. AND Gate



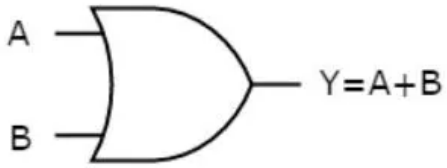
Verilog code for AND gate

```
module andGate (a,b,c);
input a, b;
output c;
assign c = a & b;
endmodule
```

Testbench of the AND gate using Verilog

```
module andGate_tb;
reg A, B;
wire C;
andGate andGate_dut (A, B ,C);
initial
begin
 #10 A = 0; B = 0;
 #10 A = 0; B = 1;
 #10 A = 1; B = 0;
 #10 A = 1; B = 1;
end
endmodule
```

Part 3. OR Gate



Verilog code for OR gate

```
module orGate (a,b,c);
input a, b;
output c;
assign c = a | b;
endmodule
```

Testbench of the OR gate using Verilog

```
module orGate_tb;
reg A, B;
wire C;
orGate orGate_dut (A, B ,C);
initial begin
 #10 A = 0; B = 0;
 #10 A = 0; B = 1;
 #10 A = 1; B = 0;
 #10 A = 1; B = 1;
end
endmodule
```