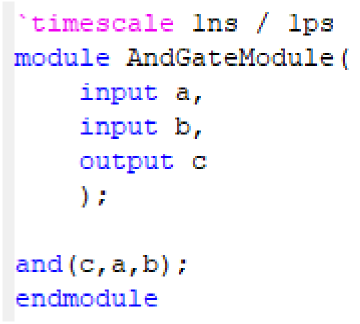
LAB#01

Introduction to Verilog Language, Gate Level Modelling and Example.

OBJECTIVE: Introduction to Verilog Language, Gate Level Modelling and Example

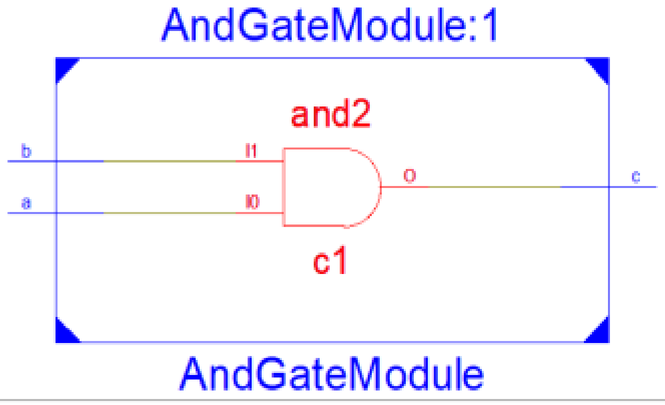
Lab Tasks: • Develop and implement different logic gates (AND, OR, NOT, XOR) using Gate Level Modeling in Verilog.

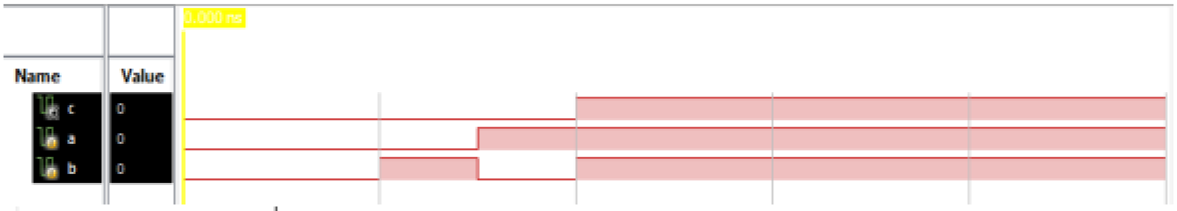
1. To Develop, Implement and Simulate AND gate using Verilog Gate Level Modelling.

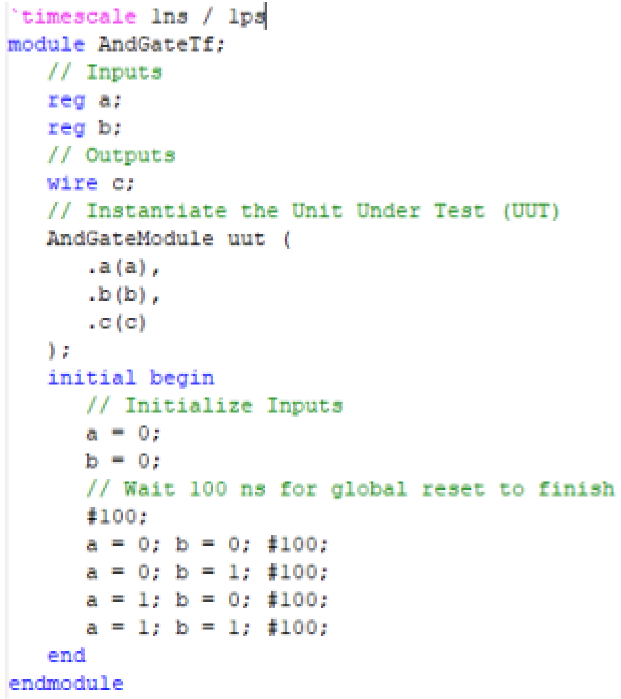


```
`timescale 1ns / 1ps
module AndGateModule(
     input a,
     input b,
     output c
     );

and(c,a,b);
endmodule
```

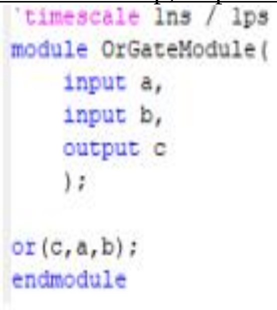






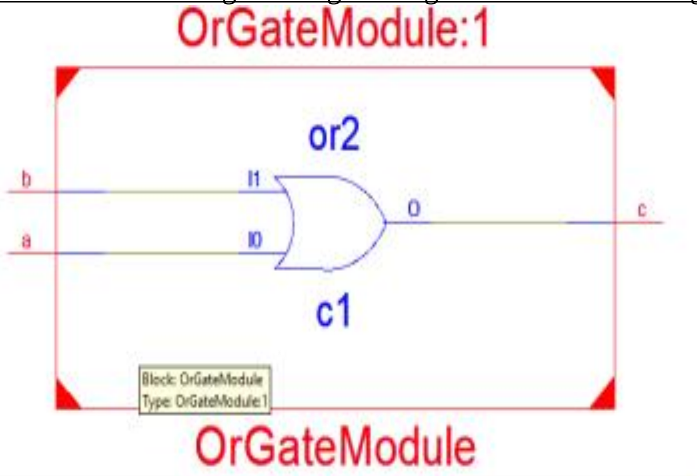
```
`timescale 1ns / 1ps
module AndGateTf;
   // Inputs
   reg a;
   reg b;
   // Outputs
   wire c;
   // Instantiate the Unit Under Test (UUT)
   AndGateModule uut (
      .a(a),
      .b(b),
      .c(c)
   );
   initial begin
      // Initialize Inputs
      a = 0;
      b = 0;
      // Wait 100 ns for global reset to finish
      #100;
      a = 0; b = 0; #100;
      a = 0; b = 1; #100;
      a = 1; b = 0; #100;
      a = 1; b = 1; #100;
   end
endmodule
```

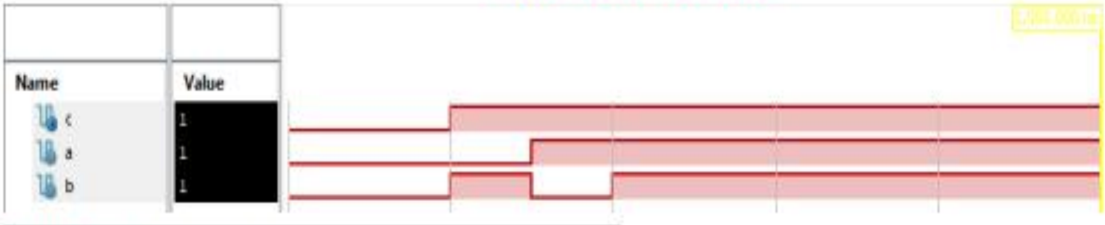
2. To Develop, Implement and Simulate OR gate using Verilog Gate Level Modelling.

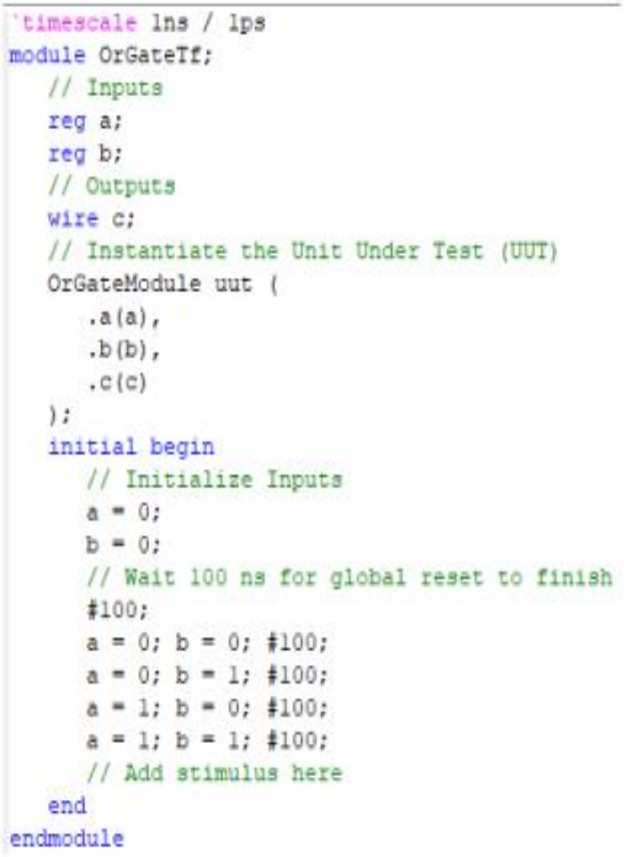


```
`timescale 1ns / 1ps
module OrGateModule(
    input a,
    input b,
    output c
    );

or(c,a,b);
endmodule
```

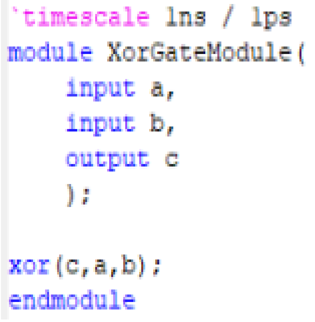






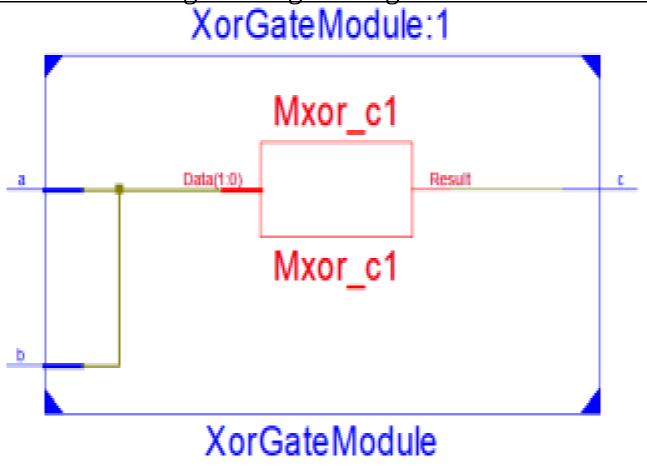
```
`timescale 1ns / 1ps
module OrGateTf;
   // Inputs
   reg a;
   reg b;
   // Outputs
   wire c;
   // Instantiate the Unit Under Test (UUT)
   OrGateModule uut (
      .a(a),
      .b(b),
      .c(c)
   );
   initial begin
      // Initialize Inputs
      a = 0;
      b = 0;
      // Wait 100 ns for global reset to finish
      #100;
      a = 0; b = 0; #100;
      a = 0; b = 1; #100;
      a = 1; b = 0; #100;
      a = 1; b = 1; #100;
      // Add stimulus here
   end
endmodule
```

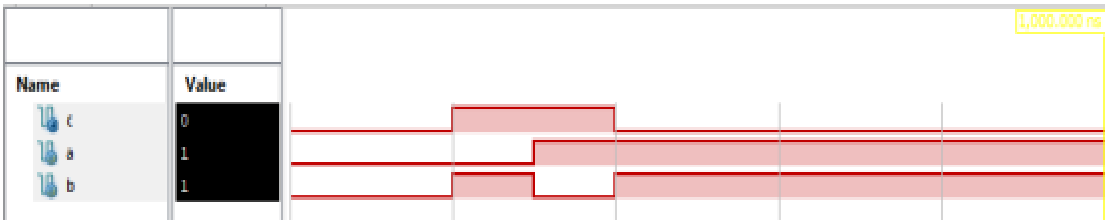
3. To Develop, Implement and Simulate XOR gate using Verilog Gate Level Modelling

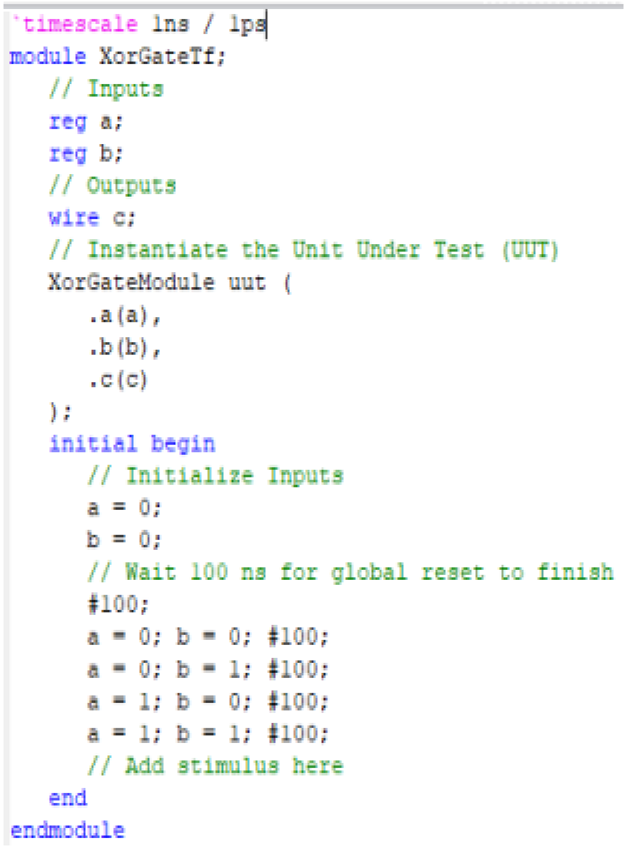


```
`timescale 1ns / 1ps
module XorGateModule(
    input a,
    input b,
    output c
    );

xor(c,a,b);
endmodule
```

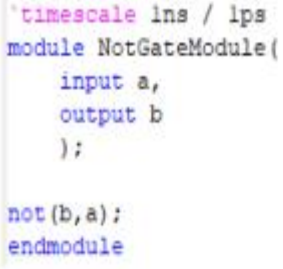






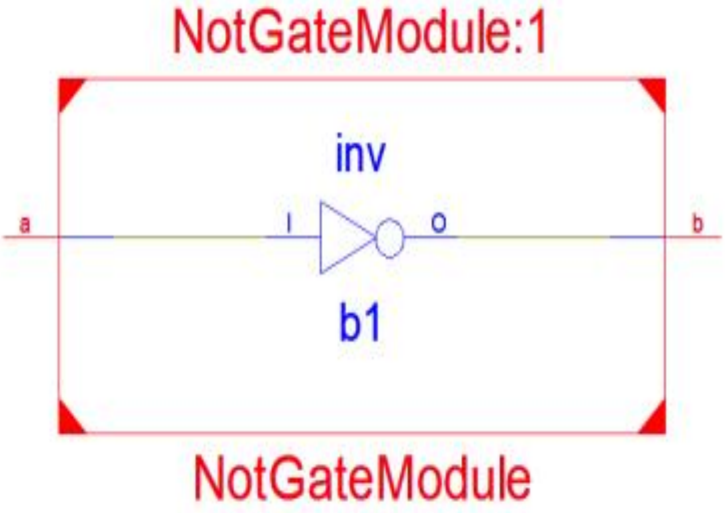
```
`timescale 1ns / 1ps
module XorGateTf;
   // Inputs
   reg a;
   reg b;
   // Outputs
   wire c;
   // Instantiate the Unit Under Test (UUT)
   XorGateModule uut (
      .a(a),
      .b(b),
      .c(c)
   );
   initial begin
      // Initialize Inputs
      a = 0;
      b = 0;
      // Wait 100 ns for global reset to finish
      #100;
      a = 0; b = 0; #100;
      a = 0; b = 1; #100;
      a = 1; b = 0; #100;
      a = 1; b = 1; #100;
      // Add stimulus here
   end
endmodule
```

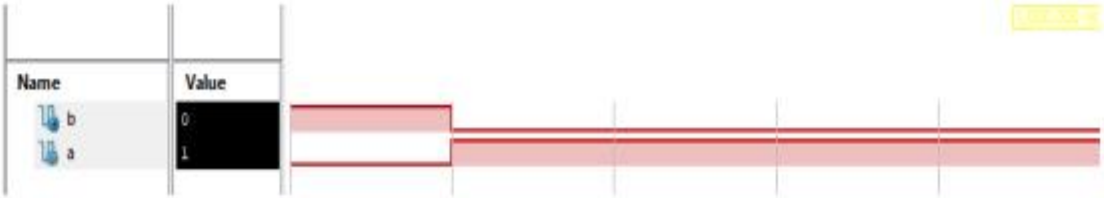
4. To Develop, Implement and Simulate NOT gate using Verilog Gate Level Modelling

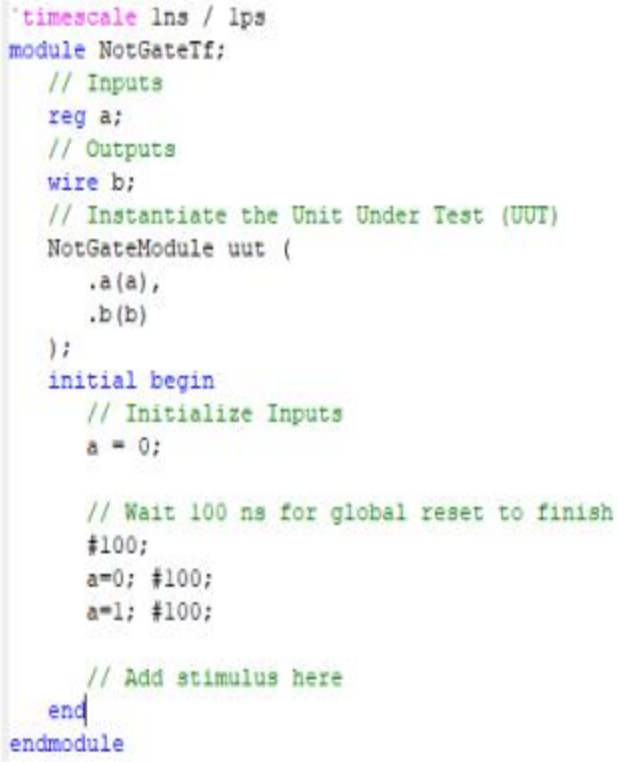


```
`timescale 1ns / 1ps
module NotGateModule(
    input a,
    output b
    );

not(b,a);
endmodule
```







```
`timescale 1ns / 1ps
module NotGateTf;
   // Inputs
   reg a;
   // Outputs
   wire b;
   // Instantiate the Unit Under Test (UUT)
   NotGateModule uut (
      .a(a),
      .b(b)
   );
   initial begin
      // Initialize Inputs
      a = 0;

      // Wait 100 ns for global reset to finish
      #100;
      a=0; #100;
      a=1; #100;

      // Add stimulus here
   end
endmodule
```