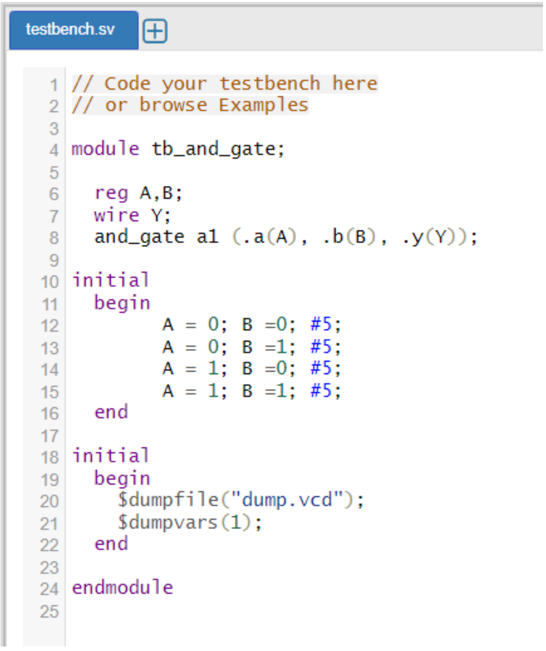
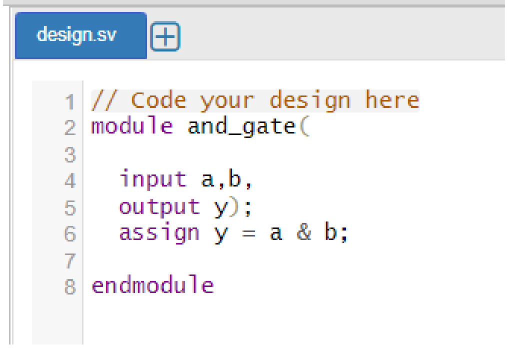
# Lab 2: Introduction To Verilog

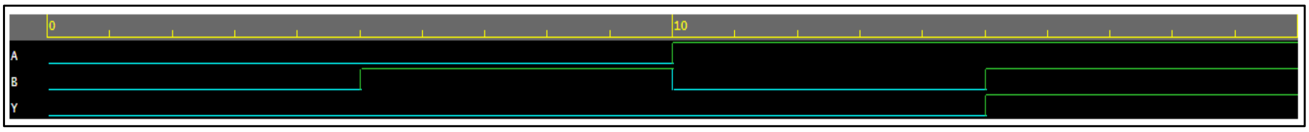
**Question 1: Write a Verilog code to implement AND gate. Write the corresponding Testbench code for the verification of your Verilog code**



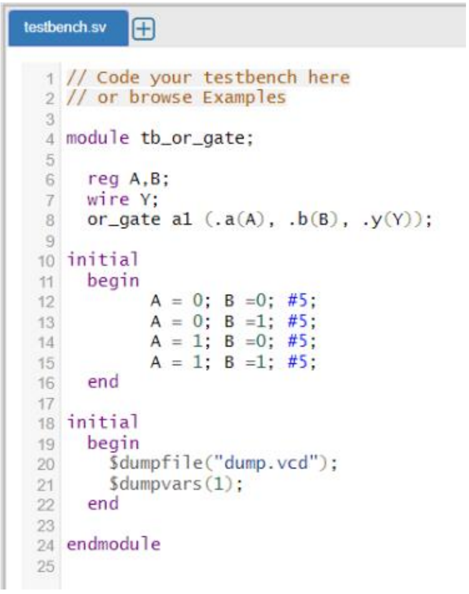
```
// Code your testbench here
// or browse Examples

module tb_and_gate;

  reg A,B;
  wire Y;
  and_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



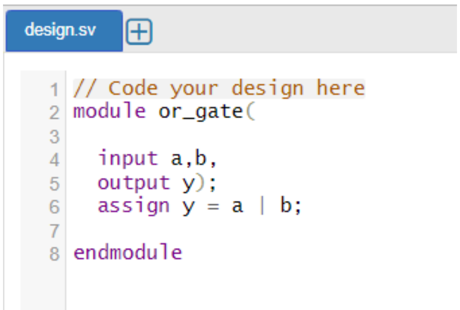
```
// Code your design here
module and_gate(

  input a,b,
  output y);
  assign y = a & b;

endmodule
```



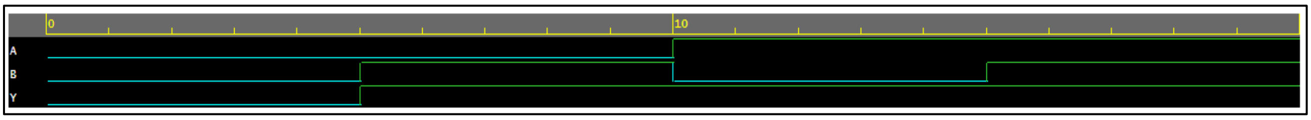
**Question 2: Write a Verilog code to implement OR gate. Write the corresponding Testbench code for the verification of your Verilog code.**



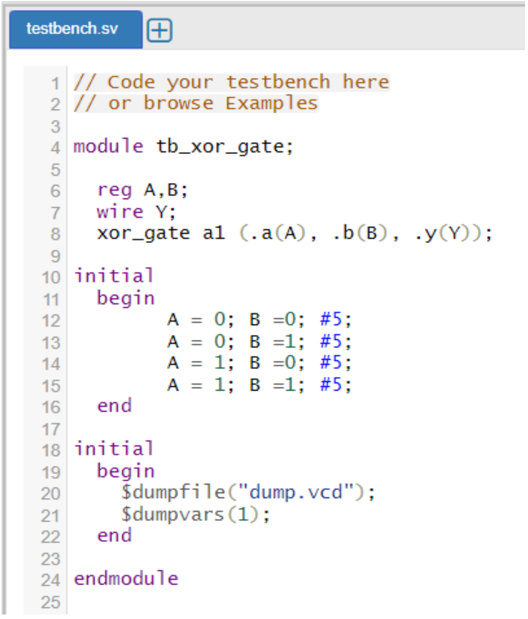
```
// Code your testbench here
// or browse Examples

module tb_or_gate;

  reg A,B;
  wire Y;
  or_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



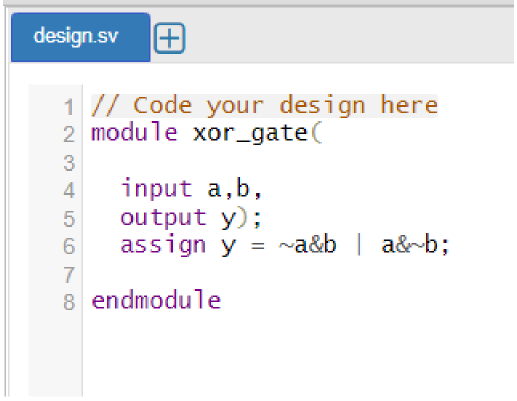
```
// Code your design here
module or_gate(

  input a,b,
  output y);
  assign y = a | b;

endmodule
```



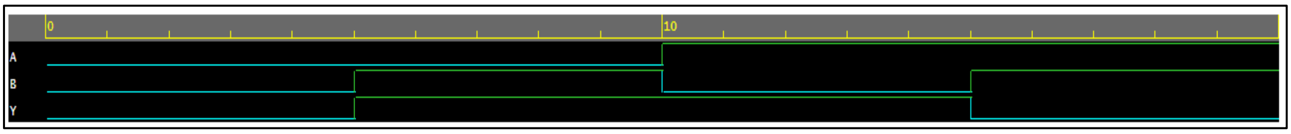
**Question 3: Write a Verilog code to implement XOR gate. Write the corresponding Testbench code for the verification of your Verilog code.**



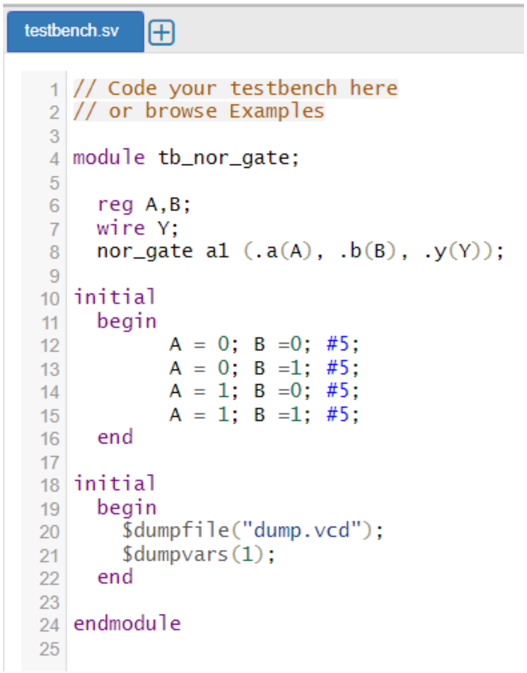
```
// Code your testbench here
// or browse Examples

module tb_xor_gate;

  reg A,B;
  wire Y;
  xor_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



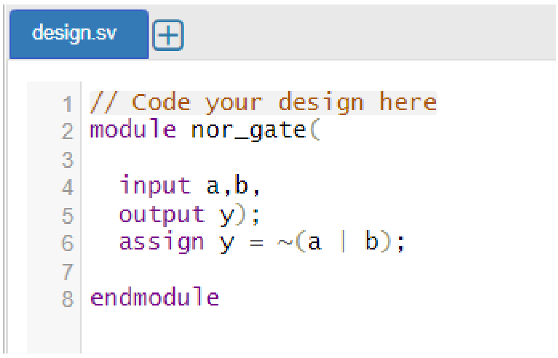
```
// Code your design here
module xor_gate(

  input a,b,
  output y);
  assign y = ~a&b | a&~b;

endmodule
```



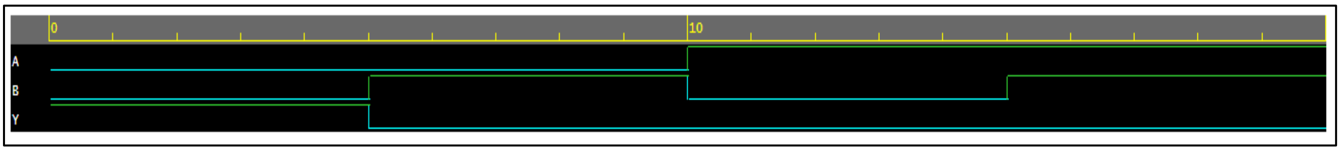
**Question 4: Write a Verilog code to implement NOR gate. Write the corresponding Testbench code for the verification of your Verilog code.**



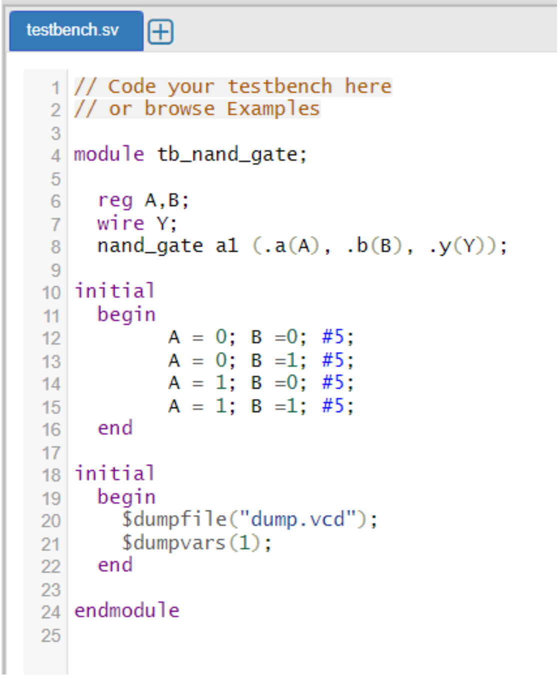
```
// Code your testbench here
// or browse Examples

module tb_nor_gate;

  reg A,B;
  wire Y;
  nor_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



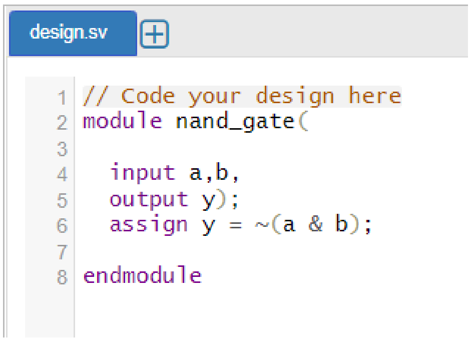
```
// Code your design here
module nor_gate(

  input a,b,
  output y);
  assign y = ~(a | b);

endmodule
```



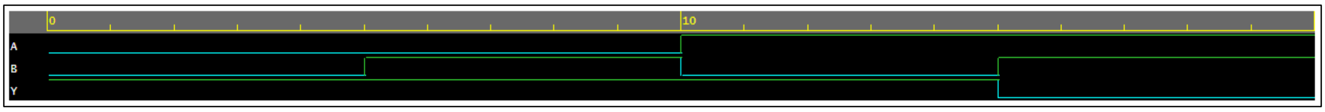
**Question 5: Write a Verilog code to implement NAND gate. Write the corresponding Testbench code for the verification of your Verilog code.**



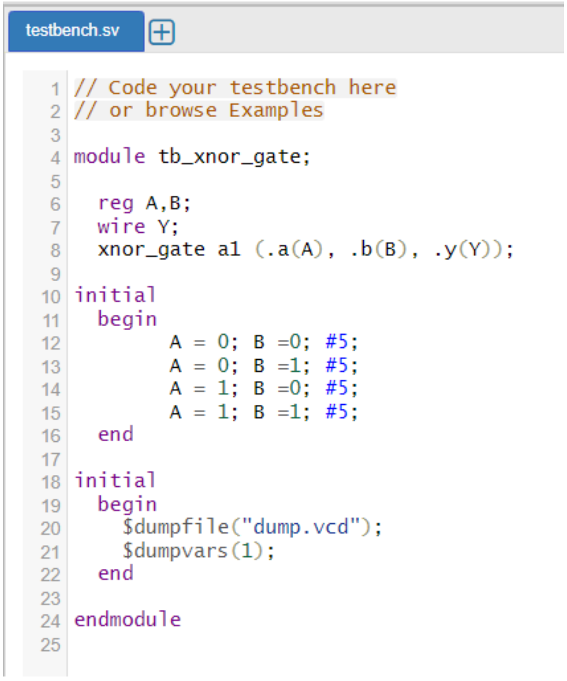
```
// Code your testbench here
// or browse Examples

module tb_nand_gate;

  reg A,B;
  wire Y;
  nand_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



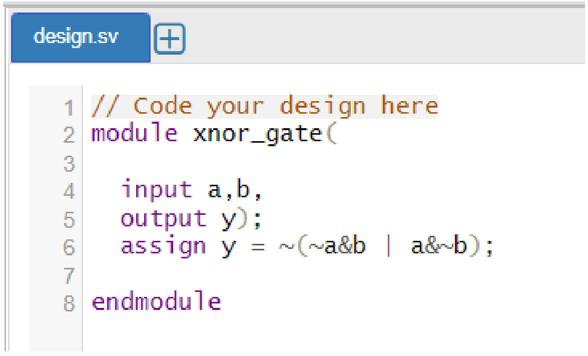
```
// Code your design here
module nand_gate(

  input a,b,
  output y);
  assign y = ~(a & b);

endmodule
```



**Question 6: Write a Verilog code to implement XNOR gate. Write the corresponding Testbench code for the verification of your Verilog code**



```
// Code your testbench here
// or browse Examples

module tb_xnor_gate;

  reg A,B;
  wire Y;
  xnor_gate a1 (.a(A), .b(B), .y(Y));

initial
  begin
        A = 0; B =0; #5;
        A = 0; B =1; #5;
        A = 1; B =0; #5;
        A = 1; B =1; #5;
  end

initial
  begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end

endmodule
```



```
// Code your design here
module xnor_gate(

  input a,b,
  output y);
  assign y = ~(~a&b | a&~b);

endmodule
```