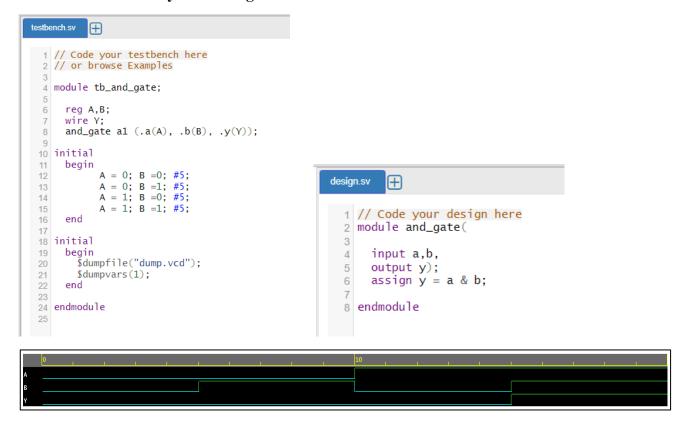
16/08/2022 21BCP359

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

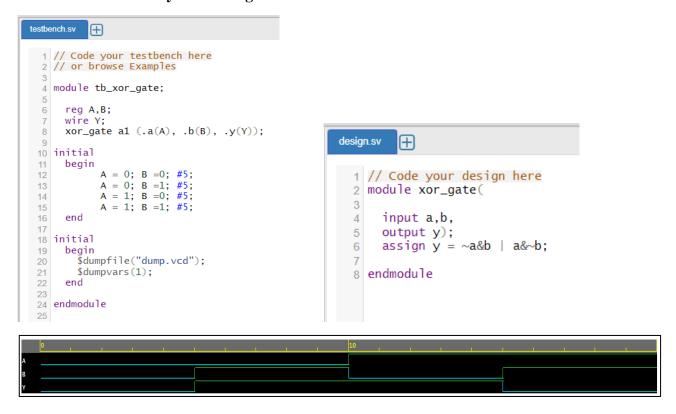


Question 2: Write a Verilog code to implement OR gate. Write the corresponding Testbench code for the verification of your Verilog code.

```
testbench.sv 📳
     / Code your testbench here
  2 // or browse Examples
  4 module tb_or_gate;
      reg A,B;
      or_gate a1 (.a(A), .b(B), .y(Y));
 10 initial
 11
      begin
            A = 0; B = 0; #5;
 13
14
            A = 0; B = 1; #5; A = 1; B = 0; #5;
                                                          // Code your design here
 16
      end
                                                       2 module or_gate(
 18 initial
                                                            input a,b,
 19
        $dumpfile("dump.vcd");
                                                            output y);
 20
        $dumpvars(1);
                                                            assign y = a \mid b;
                                                        8 endmodule
 24 endmodule
```

16/08/2022 21BCP359

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

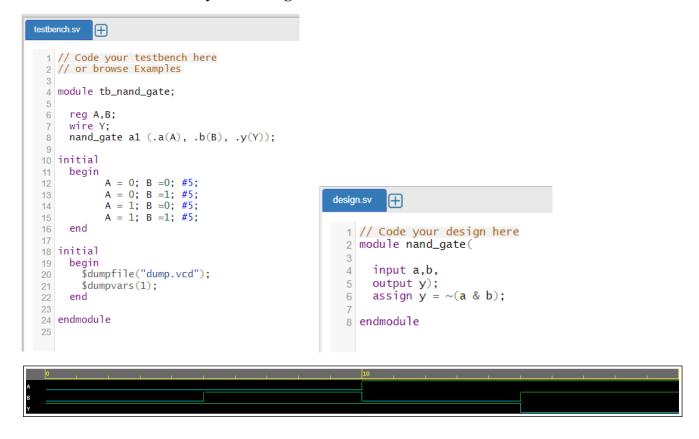


Question 4: Write a Verilog code to implement NOR gate. Write the corresponding Testbench code for the verification of your Verilog code.

```
testbench.sv
          \blacksquare
   // Code your testbench here
  2 // or browse Examples
  4 module tb_nor_gate;
  6
      reg A,B;
  8
      nor_gate a1 (.a(A), .b(B), .y(Y));
 10 initial
      begin
 11
            A = 0; B = 0; #5;
 12
                                                    design.sv
                                                               \oplus
            A = 0; B = 1; #5;
 13
            A = 1; B = 0; #5;
 14
            A = 1; B = 1; #5;
                                                       1 // Code your design here
 16
      end
                                                       2 module nor_gate(
 17
                                                       3
 18 initial
                                                            input a,b,
      begin
 19
        $dumpfile("dump.vcd");
 20
                                                            output y);
        $dumpvars(1);
 21
                                                       6
                                                            assign y = \sim (a \mid b);
      end
 22
                                                       8 endmodule
 24 endmodule
```

16/08/2022 21BCP359

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



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

```
testbench.sv
  1 // Code your testbench here
  2 // or browse Examples
  4 module tb_xnor_gate;
  6
      reg A,B;
      wire Y:
      xnor_gate al (.a(A), .b(B), .y(Y));
  8
 10 initial
 11
 12
            A = 0; B = 0; #5;
                                                     design.sv
            A = 0; B = 1; #5;
                                                               \oplus
 13
            A = 1; B = 0; #5;
 14
            A = 1; B = 1; #5;
 15
                                                          // Code your design here
      end
 16
                                                          module xnor_gate(
                                                       2
 17
 18 initial
                                                       3
 19
                                                             input a,b,
                                                       4
        $dumpfile("dump.vcd");
 20
                                                            output y);
        $dumpvars(1);
                                                            assign y = \sim (\sim a\&b \mid a\&\sim b);
                                                       6
    endmodule
 24
                                                       8 endmodule
 25
```