Dept. of CSE, Bennett University

Digital Design – ECSE 108L

Lab Assignment – 4

In this Lab, we shall learn 'Module Instantiation'. You will write the code for one module; the same module will be used in other module of your Verilog Code.

A module provides a template from which you can create actual objects. When a module is invoked, Verilog creates a unique object from the template. Each object has its own name, variables, parameters, and I/O interface. The process of creating objects from a module template is called instantiation, and the objects are called instances.

Previously, we have used these instances in the Testbench code. Now we shall use it in Verilog Code also.

1. Write down three modules in a single Verilog code to design AND, OR and NOT gate. Now use those modules to design the XOR gate. Use AND, OR and NOT gate as the instances to implement the XOR gate. Write the corresponding Testbench code for the verification of your XOR gate.

   Truth Table of XOR Gate:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. Consider the following expression:

   Y = A'.B'.C' + A'.B.C' + A.B'.C' + A.B'.C

   (i) Generate the Truth Table manually for the same.

   (ii) Design one three input 'And' gate module and one four input 'OR' gate module using Verilog. Instantiate those two modules to design the above-mentioned expression. Design the corresponding TestBench code for the verification purpose.

   (iii) Now Minimize the given expression.

   (iv) Design again 'AND' gate and 'OR' gate module with required number of inputs and instantiate them to implement the minimized expression. Design the corresponding TestBench code for the verification purpose.

3. Consider the following expression:

Y = A.B + B.C

(i) Express this function using Universal NAND gate representation.

(ii) Design one NAND Gate module using Verilog and instantiate that to implement the NAND Gate representation of the above-mentioned equation. Only NAND Gate module need to be instantiated.

(iii) Write the Truth Table of the above-mentioned expression.

(iv) Verify the same with corresponding Testbench Code.

4. Consider the following expression:

Y = (A + B) . (A + B')

(i) Express this function using Universal NOR gate representation.

(ii) Design one NOR Gate module using Verilog and instantiate that to implement the NOR Gate representation of the above-mentioned equation. Only NOR Gate module need to be instantiated.

(iii) Write the Truth Table of the above-mentioned expression.

(iv) Verify the same with corresponding Testbench Code.

- **Sample Verilog Design Code for Question 1:**

```
module not_gate (input e, output f);
  assign f = ~e;
endmodule

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

module or_gate (input p, q, output r);
  assign r = p | q;
endmodule

module build_xor (input m, n, output o);
  wire x, y, a_not, b_not;
  not_gate n1 (.e(m),.f(a_not));
  not_gate n2 (.e(n),.f(b_not));
  and_gate a1 (.a(a_not),.b(n),.c(x));
  and_gate a2 (.a(m),.b(b_not),.c(y));
```

```
  or_gate o1 (.p(x),.q(y),.r(o));

endmodule
```

- **Sample TestBench Code for Question 1:**

```
module tb_xor_str;
   reg A,B;
   wire C;
  build_xor x1 (.m(A), .n(B), .o(C));
  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
```

Submission Instructions:

- Prepare the submission file according to the following process:
    1. Copy the Verilog code, the Test Bench Code in a Word File.
    2. Take the ScreenShot of Waveform and paste into the same word file.
    3. Repeat Step 1 and 2 for all the programs.
    4. Copy and Paste all the Verilog code, Testbench Code and Waveform into a single word file as 1_verilog, 1_TestBench, 1_Waveform, 2_verilog, 2_TestBench, 2_Waveform… etc.
    5. Convert it into pdf file, name it as **RollNo_Assignment# (Example: E20CSE001_ Assignment3.pdf)**.
    6. Submit your file on LMS **within the deadline**.

- Write your **Name and Roll No. as comment before starting of each program**. Keep in mind this is **Mandatory**. Failing which you may lose your marks.
- Make it sure that in each program, **you have mentioned enough comments** regarding the explanation of program instructions.
- **Each student will submit their assignment on their corresponding group slot only.**
- Late submission will lead to penalty.
- Any form of plagiarism/copying from peer or internet sources will lead penalty.
- Following of all instructions at submission time is mandatory. Missing of any instructions at submission time will lead penalty.