UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Fall 1395
Computer Assignment 1 and 2
Switch, Gate, and Expressions in Verilog - Week 4

Name:	**Date:**
Username:	

The problem for this assignment is a circuit that calculates the number of 1's on its data inputs. The circuit that you start with has three inputs (n=2) and it is cascadable to build any circuit with 2^n-1 inputs. Assume complement of all inputs are available and use Verilog ~ operation for getting the complement of circuit primary inputs.

1. Show switch level circuit diagram for a Complex gate CMOS structure for a circuit with *a*, *b*, and *c* inputs and y_0 and y_1 outputs. The outputs (y_1, y_0) form a two-bit number that represent the total; number of 1's on inputs *a*, *b*, and *c*. Where possible, factor out a variable (using Boolean Distributive law) to save nMOS and pMOS transistors. We refer to this circuit as One's Count with 3 inputs (OC3). Use a minimum number of nMOS and pMOS transistors. Use #(3, 4, 5) delay values for the nMOS transistors and #(5, 6, 7) for the pMOS transistors. As indicated above use ~ when complements of *a*, *b*, and *c* are needed. The ~ operation will not be counted as using transistors in your design. However, complementing an output will require two MOS transistors.
2. Show Verilog description for OC3 circuits of Part 1. Use delay values given above.
3. Generate a testbench for the OC3 circuit of Part 2, apply test data and justify the results you are getting. Explain X and Z values.
4. Describe the circuit of OC3 circuit of Part 2 using Verilog gate primitives. Adjust primitive gate delays to make the delay of the gate level OC3 circuit of this part as close as possible to the transistor-based circuit of Part 2.
5. Generate a testbench that instantiates the transistor-level OC3 of Part 2 and the gate-level OC3 of Part 4. Test the two OC3 descriptions simultaneously and explain the differences.
6. Describe the circuit of Part 2 using two **assign** statements. Adjust **assign** statement delays to make the delay of this description as close as possible to the gate-level circuit.
7. Generate a testbench that instantiates the descriptions of Parts 2, 4, and 6. Test the three circuits simultaneously and explain the differences.

---

8. Using several copies of OC3 of Part 2 in a structural fashion, build a circuit for counting the number of ones on its seven data inputs (n is 3). We refer to this circuit as an OC7 circuit.
9. Repeat Part 8, but use the OC3 circuit described using **assign** statements (circuit of Part 6).
10. Generate a testbench for the circuits of Part 8 and 9, and compare the timings.
11. Now design an OC15 circuit (n=4), using OC7 and OC3 circuits of Part 8 and Part 2. Test this circuit and calculate its worst case delay.