



**UNIVERSITY OF TEHRAN**  
**Electrical and Computer Engineering Department**  
**Digital Logic Design, ECE 367, ECE 894, Fall 1402**  
**Computer Assignment 1-2**  
**Basic Switch and Gate Structures in Verilog**

**Name:**

**Date:**

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 cascable to build any circuit with  $2^n-1$  inputs. Assume complement of all inputs are available and use Verilog  $\sim$  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 a 3-input One's Counter, i.e., 3-input OC. 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  $\sim$  when complements of  $a$ ,  $b$ , and  $c$  are needed. The  $\sim$  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 3-input OC circuit of Part 1. Use delay values given above. Generate a testbench for this circuit, apply test data and justify the results you are getting. Explain X and Z values.
3. Describe the circuit of Part 2 using Verilog gate primitives (start from the circuit diagram). Adjust primitive gate delays to make the delay of the gate level circuit of this part as close as possible to the transistor-based circuit of Part 2. Generate a testbench that instantiates the transistor-level circuit of Part 2 and the gate-level circuit of Part 3. Test the two descriptions simultaneously and explain the differences.
4. 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. In a testbench, instantiate the descriptions of Parts 2, 3, and 4 and test the three circuits simultaneously and explain the differences.

---

Thus far, you have built a  $(2^n-1)$ -input OC circuit where  $n$  is 2. In the continuation, you are to build larger OC circuits.

5. Using several copies of 3-input OC circuit of Part 2 in a structural fashion, build a 7-input OC circuit.
6. Repeat the above, but use the 3-input OC circuit that you used **assign** statements to describe (circuit of Part 4). In a testbench, instantiate the circuits of Part 5 and Part 6 and compare the timings.

7. Now design a 15-input OC circuit ( $n=4$ ), using 7-input and 3-input OC circuits of Part 5 and Part 2. Test this circuit and calculate its worst-case delay.

**Deliverables:**

Generate a report that includes all the items below:

- A. Do Parts 1 and 3, do the circuit diagram on paper and calculate the delay values manually before running the simulation.
- B. For Parts 3 and 5, show your extracted delay values from the waveforms and compare them with your hand calculated values.
- C. For all parts, in your submitted report, show hand calculations and waveforms. Your SystemVerilog descriptions must correspond to the circuit diagrams. Your simulation run and the project built for this purpose must be demonstrated to the TA.

In your hand-written report, write your name, student number, course name and course number and the assignment number. Hand-in your completed report to the TA.