UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Digital System I, 894, Fall 1403
Computer Assignment 3
Basic flip-flops and memory elements

Name: **Date:**
Username:

1. Start with a cross-coupled NAND structure and design an SR latch with active low S and R inputs. Using this structure and OR gates, generate a clocked SR-latch with activity levels that are opposite of those discussed in class based on cross-coupled NOR gates.

2. Use Verilog for describing the circuit of Part 1. Use approximate gate delays that are based on gate level delays of 7 NS. In a testbench, test your circuit, and, among other inputs, apply simultaneous active inputs (S and R) and see the loss of memory.

3. Use an inverter to convert the SR latch of Part 1 to a clocked D-latch. The inverter delay is the same as a gate delay. By adding inverters and NAND or NOR gates, change your circuit to one with an active high clock input.

4. Simulate your circuit of Part 3 to verify its operation. In the testbench that you develop, apply the clock to various D input values, change the clock while the D input is 1 and when it is zero, try various input combinations and sequences, and report your results in a timing diagram.

5. To the D latch of Part 4, add a reset input, R, that forces a 0 into the Q output when clock is active. Generate an 8-bit register using eight latches that now have a reset input.

6. Take an 8-bit adder with 8-bit *A* and *B* inputs. Adjust the worst-case delay of this circuit according to transistor delays of CA1. Use an **assign** statement for the description of the adder. Using this adder, design a circuity that on each clock, it takes it's *A* input adds it to a register contents and saves the added result until the next clock. This can be achieved by taking the output of the adder, feeding it to the register, and then feeding back the register output to the *B* input of the adder. We refer to this circuit as a Sequence Adder. Use the register of Part 5 for building this the Sequence Adder.

7. Generate a testbench for the Sequence Adder of Part 6, apply data and clock, observe the worst-case delay in timing your clock, and report the results.

8. Use two D-latches of Part 3 to build a master-slave D-type flip-flop. Create a synchronous reset for this flip-flop such that when this input becomes 1, the output becomes 0 with the clock. Simulate this

circuit to verify its operation. When not being reset, apply test data of Part 4 to the clock and D input of this flip-flop. Report the results and compare the waveforms with those of Part 4.

9. Use the flip-flop of Part 8 to build the Sequence Adder circuit. Write a testbench for the Sequence Adder and test its operations. Always reset the circuit before starting a new operation.

10. In a testbench, instantiate Sequence Adder circuits of Part 6 and Part 9, and compare the results.

Deliverables:

Generate a report that includes all the items below:

a. For Parts 1,3 show your design on paper.

b. For other parts, show simulation results. Your simulation run and the project built for this purpose must be demonstrated to the TA.

c. For all parts, project files, and Waveforms must be demonstrated to the TA. Using waveforms, circuit diagrams, and other circuit representations justify your answers.

Make a PDF file of your report and name it with the format shown below:

FirstinitialLastnameStudentnumber-CAnn-ECEmmm

Where *nn* is a two-digit number for the Computer Assignment, *mmm* is the three-digit course number under which you are registered, and hopefully you know the rest. For the *Firstinitial* use only one character. For *Lastname* and for the multi-part last names use the part you are most identified with. Use the last five digits of your student id (exclude 8101) for the *Studentnumber* field of the report file name.