UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital Logic Design, ECE 367 / Digital Systems I, ECE 894

Fall 1400

Computer Assignment 4

Latches, flip-flops, and a little beyond

Week 20

Name:

Date:

1. In the previous assignments you have seen that you can build a 2-to-1 multiplexer using tri-state inverters. You have also seen that you can 2-state inverters in part of this multiplexer to save a few transistors. In the mid-term exam you saw that you can build any logic using a multiplexer. Considering the transistors or Computer Assignment 1, built a 2-to-1 multiplexer with tri-state and 2-state inverters. Using this multiplexer, by feeding back the multiplexer output back to the input selected by select=0, build a D-latch. Now you see that you can even build sequential circuits using a multiplexer.
   a. Write SystemVerilog description of the multiplexer using **not** and **notif1** primitives. Use delay values bases on the transistors of Computer Assignment 1.
   b. Use the multiplexer of Part a to build a positive level sensitive D latch.
   c. Write a testbench to test the latch of Part b. For the latch to work, you should first clock it to force a good D value in the latch.
   d. In the testbench, examine cases that the clock duration is too short, the D input is too close to the clock, the D input is a short pulse while the clock is active, and other timing violations that you think the latch may not operate properly.
2. Form an 8-bit shift register using the above latch. From left to right (*i* to 0), the output of latch *i* connects to the D input of latch *i-1*. The D input of latch *i* is the *serIn* (serial input) of the shift register.
   a. Generate a testbench and test your shift register.
   b. Simulate the shift register and explain why or why-not this circuit works as expected.
3. Use two D-latches of Problem 1 to build a master-slave D-type flip-flop. Create a synchronous reset (*syncReset*) for the flip-flop such that when this input becomes 1, the output becomes 0 with the clock.
   a. Write a testbench for testing your flip-flop and verifying its load and resetting capabilities
   b. Examine the timing of the flip-flop as in Problem 1.
4. Build an 8-bit shift-register using the flip-flop of Problem 3 in a **generate** statement.
   a. Using a testbench, simulate this register and verify its clocking, shifting, and resetting operations. Explain why this circuit works and that of Problem 2 does not.

b. Write an 8-bit shift-register using an **always** statement. Provide a synchronous reset. Adjust the delays so that it performs as close as possible to the shift register that uses the **generate** statement.

5. Using the shift register of Problem 4 design an LFSR with $x^8+x^6+x^5+x^2+1$ polynomial. For this, number shift-register flip-flops from 7 to 0 from left to right. A feedback from output of bit 0 (x^0 in the polynomial) connects to the input of bit 7 (x^8 in the polynomial). In the polynomial, the coefficient of term (x^i) is 1 where flip-flop *i* output connects to the feedback through an XOR gate. Coefficient of 0 for a term means that the corresponding output does not contribute to the feedback. In addition to the *syncReset*, provide an input that synchronously *syncInit* forces the register into the 8’b10000000 state.

a. Write a testbench to test the LFSR.

b. Start with all 0’s with the *syncReset* and see what happens when you clock the LFSR

c. Load 10000000 with *syncInit* and see what happens when you clock the LFSR.

d. Study what happens with the sequences of the LFSR.

e. (Bonus) What is the period of this polynomial, i.e., how many clock cycles you have to give it for it to repeat itself. What is a maximal length polynomial? Learn about the period of this polynomial and answer questions when asked by the TA.

Deliverables:

Generate a report that includes all the items below:

A. Prior coming to the lab, for all the above problems hand-drawn schematic diagrams and partial timing diagrams are required. 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.

B. For all problems, be prepared to answer questions asked about the timings, minimum pulse widths, and time distance between various signals.

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.