

**University of California at Berkeley**  
**College of Engineering**  
**Department of Electrical Engineering and Computer Science**

EECS 150  
Spring 2004

R. H. Katz

**Problem Set # 4 (Assigned 19 February, Due 27 February)**

1. In lecture, we presented a negative edge-triggered D flip-flop using NOR gates (Lecture #6, Slide 18).
  - (a) Implement a *positive-edge* triggered D flip-flop function using *NAND* gates only.
  - (b) Explain the timing behavior of this circuit. Label your internal circuit nodes, show their waveforms on the timing chart, and use these waveforms to briefly explain why the triggering works.
2. Your textbook presents the concept of a Master-Slave flip-flop, constructed from two cascaded stages of R-S flip-flops (Lecture #6, Slide 15).
  - (a) Master-Slave flip-flops implemented in this way exhibit the phenomena of “ones catching.” Explain what it is and why it happens. Is it possible for a Master-Slave flip-flop to “catch zeros”? Justify your answer!
  - (b) Suppose that you implement a Master-Slave flip-flop as two cascaded D flip-flops, the first stage a positive edge triggered D FF and the second stage a negative edge triggered D FF. Draw a timing chart with a clock waveform and a D input that oscillates from 0 to 1 or 1 to 0 each time the clock is low. Make sure you show how the two flip-flops’ outputs change in response to input and clock changes.
  - (c) Can the configuration in part (b) catch ones? Explain your answer in terms of this timing chart.
3. Design a 3-bit counter that implements the following sequence: 000, 111, 010, 101, 001, 110, 100, 011, and repeat. Design the counter with a reset input that causes the counter to enter the 000 state.
4. Consider the design of a simple elevator controller. The building has two floors, an up button on the first floor, a down button on the second floor, and two buttons inside the elevator indicating a desire to go up or go down. While you can make assumptions, the behavior of the system has to be reasonable. For example, pressing the “Up” button with the elevator on the second floor causes the elevator to remain there with its door open.
  - (a) Identify your inputs, outputs, and name and describe your states.
  - (b) Draw a symbolic state diagram for your design, labeling all state transitions.
  - (c) Write the Verilog code for a Moore Machine implementation of this state diagram.
5. Consider the following variation on the classical traffic light controller problem. A North-South road intersects an East-West road. In addition to the Red/Yellow/Green traffic lights, each of the four corners of the intersection has *Walk/Don’t Walk* signs facing N-S and E-W. The *Walk/Don’t Walk* signs cycle for the N-S road as follows. When the traffic lights turn green in the N-S direction, *Walk* is illuminated. Half way through the green light time, the *Don’t Walk* sign becomes illuminated and starts to blink (*Walk* is no longer illuminated). When the traffic light turns yellow, the *Don’t Walk* sign remains solidly illuminated (i.e., no blinking). It stays this way throughout the red light cycle. The behavior is the same in the E-W direction.
  - (a) Assume that the standard Green light time is 56 seconds, the Yellow time is 8 seconds, and the Red time is 64 seconds. Draw a simple timing chart that shows the behavior of the N-S and E-W traffic lights and *Walk/Don’t Walk* lights.
  - (b) Identify your inputs and outputs. What additional circuitry, like timers and flipflops, do you need outside of the state machine?
  - (c) Draw a symbolic state diagram. Make clear all of your assumptions about the problem consistent with the specification above.
  - (d) Write the Verilog code for a Moore Machine implementation of this state diagram.