1. Design a combinational logic subsystem with three inputs, $I_3$, $I_2$, $I_1$, and two outputs, $O_1$, $O_0$, that behaves as follows. The outputs indicate the highest index of the inputs that is driven high. For example, if $I_3$ is 0, $I_2$ is 1, $I_1$ is 1, then $O_1$, $O_0$ would be 10 (i.e., $I_2$ is the highest input set to 1). If none of the inputs is 1, then the outputs would be 00.
   a. Specify the subsystem by filling out a complete truth table for the two outputs.
   b. Write a specification of this function in Verilog.
   c. Find the minimized Sum of Products description using K-maps.
   d. Implement the subsystem using 2 x 4:1 multiplexers.
   e. If implemented using a ROM, what size ROM is required? Why?
   f. Compare your solutions in parts (c) and (d). Which is simpler and why (i.e., what criteria are you using to measure complexity)?

2. Scientists have discovered that the Venusians use a base 16 number system. However, the digits are quite different than the ones to which we are accustomed. The first row below is 0 through 7, and the second row is 8 through F. Your task is to design a combinational logic subsystem to decode a hexadecimal digit in the range of 0 (0000) through F (1111) to drive a seven-segment display for the Martian version of the hexadecimal digits (0-7 in the top row, 8-F in the bottom row). The LED segments are numbered counter clockwise starting at the horizontal LED at the bottom (LED0). The middle LED is LED6.

   ![Seven-segment display diagram]

   a. Specify the function by filling out a complete truth table for each of the seven segment drivers.
b. Write each as a Verilog specification.
c. Develop the minimized gate-level implementation using the K-map method, minimizing each K-map independently.
d. Repeat (c), but this time, minimize so as to exploit shared product terms wherever possible, as though the implementation target is a PLA.
e. How does the complexity of your answers to (c) and (d) compare? Which is simpler and why? What criteria are you using to measure the complexity of these two different implementations?

3. Your textbook presents the concept of a Master-Slave flip-flop, constructed from two cascaded stages of R-S flip-flops (see Figure 6.18 on page 269).
   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.

4. Your textbook presents a negative edge-triggered D flip-flop using NOR gates (see Figure 6.24 on page 272).
   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.

5. Design a 3-bit counter that implements the following sequence: 000, 010, 100, 110, 111, 101, 011, 001, and repeat. Design the counter with a reset input that causes the counter to enter the 000 state.
   a. Write a Verilog specification for this counter.
   b. Design the next state functions and minimize for PLA-based implementation.
   c. Show how to implement this counter using 2:1 multiplexers and D flip-flops only. You may assume that the D FFs have a reset input.