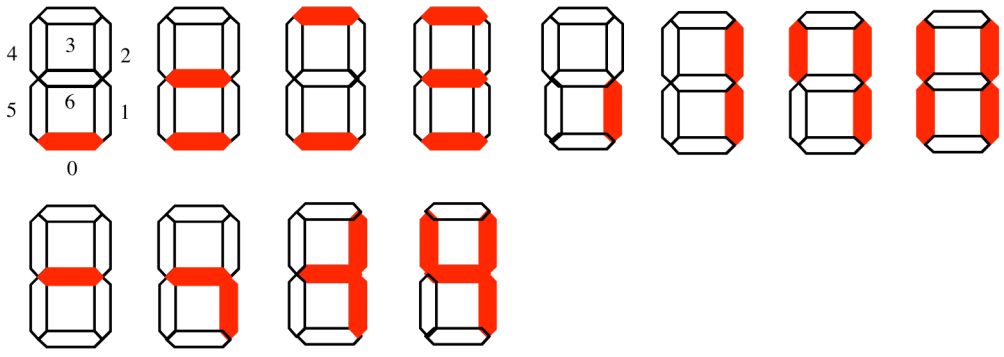
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150 R. H. Katz
Spring 2007

Problem Set #3: Combinational and Sequential Logic (REVISED)
Assigned 30 January 2007, Due 9 February at 2 PM

1. Design a combinational logic subsystem with five inputs, I_4, I_3, I_2, I_1, I_0, and three outputs, O_2, O_1, O_0, which behaves as follows. The outputs indicate the count of the inputs that are currently true. For example, if I_4 and I_3 are 0, and I_2, I_1, and I_0 are 1, then O_2, O_1, O_0 would be 011 (i.e., three of the five inputs are 1).
    a. Specify the subsystem by filling out a complete truth table for the three outputs.
    b. Write a specification of this function in Verilog. HINT: Think hard about how this function actually behaves, rather than thinking in terms of Boolean equations. Can you make use of the "+" operator in Verilog to describe this subsystem?
    c. Find the minimized Sum of Products description using K-maps (yes, a 5 variable K-map! You can do it!).
    d. Implement the subsystem using 3 x 16:1 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 Plutonians use a base 12 number system. Furthermore, the digits are quite different than the ones to which we are accustomed. The first row below is 0_{10} through 7_{10}, and the second row is 8_{10} through 11_{10} (there are no digits for the base 10 numbers 12 through 15). Your task is to design a combinational logic subsystem to decode a hexadecimal digit in the range of 0_{16} (0000) through B_{16} (1011) to drive a seven-segment display for the Plutonian version of the hexadecimal digits (hex 0-7 in the top row, hex 8-B in the bottom row; the rest of the hex digits are don't cares). The LED segments are numbered counter clockwise starting at the horizontal LED at the bottom (LED_0). The middle LED is LED_6.



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. Consider a new kind of flip-flop with inputs X,Y (and the clock of course) and output Q. The behavior of the X-Y flip-flop is as follows: When X and Y are both 0, the flip-flop holds its current state. When X is 0 and Y is 1, the flip-flop toggles its state (a state of 0 becomes 1 and vice versa). If X is 1 and Y is 0, the flip-flop state is made zero. If X and Y are both 1, the flip-flop state becomes 1.
    a. Show how to implement the X-Y FF as a combinational logic block with inputs X, Y, Q, Q' and single output D that is connected to a second stage that is a standard D FF. Use Nand, Nor, and Inverter logic.
    b. Repeat your design for the combinational logic block of part (a), but this time using a 4:1 multiplexer with selection inputs X and Y and the data inputs wired to 0, 1, Q or Q'.

4. Using the X-Y FF of Problem 3, show how to wire up X and Y to implement the following other kinds of FFs. Your solution should use minimum logic – the fewest possible gates and literals.
    a. Implement the same function as a D FF by designing a combinational logic block whose inputs are drawn from the values constant 0, constant 1, D, D', Q, Q', and whose outputs are X and Y.
    b. Implement the same function as a T FF (a toggle flip-flop inverts its state whenever the T input is true; it holds its current state otherwise) by designing a combinational logic block whose inputs are drawn from the values constant 0, constant 1, D, D', Q, Q', and whose outputs are X and Y.

5. Design a 3-bit counter that implements the following sequence: 000, 111, 001, 110, 010, 101, 011, 100, 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.