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

EECS 150   D. E. Culler
Fall 2007

Problem Set #2: Programmable Logic and HDLs
**Assigned 9/6/2007, Due 9/14/2007 at 2 PM**

Problem 1. You are given a single 2-input MUX. How many of the 2-input boolean functions can you obtain by wiring the three inputs of this device to some combination of A, B, Logical 0, and Logical 1. You do not have the complements of A and B. (Show how.) How many more can you get with 2 MUXes? 3 Muxes?

Problem 2. Based on the information in lecture and the data sheet on the Xilinx Vertex family, how does the number of User I/O pins grow with the number of LUTs?   How do the BlockRAM Bits scale with LUTs?

Problem 3: You are synthesizing logic (by hand) for and idealized FPGA consisting of many CLBs as shown in the following figure and interconnect that ties them together. Show how to use a collection of these blocks to implement the 4-bit Parallel-to-Serial converter discussed in lecture (slide 10 of Lecture 3). For each block, specify the contents of the 4-LUT and the latch. Show how the blocks are wired together.
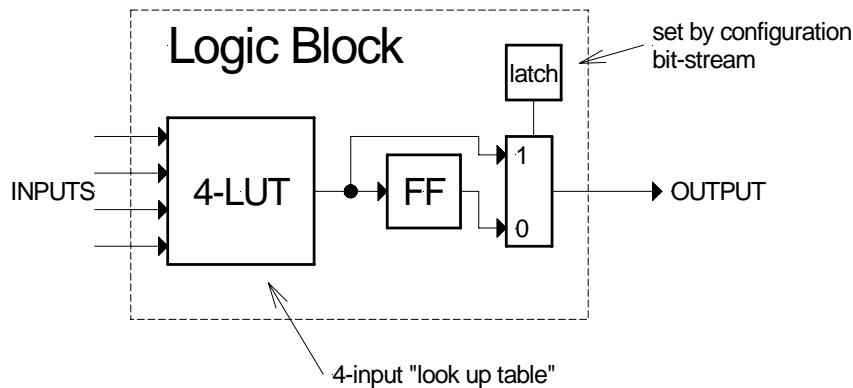
Problem 4: In lab 0 you implemented an XOR using 4 nand gates. Write the structural verilog module corresponding to the schematic below.
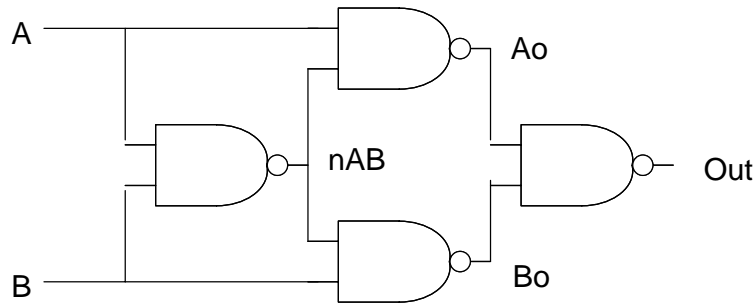

Write the same behavior as a behavioral module using continuous assignment.

Write the same behavior as a behavioral module using Blocking assignment.

Write the same behavior as a behavioral module using non-Blocking assignment.

Problem 5. Some aspects of digital design are deeply rooted in the mathematical elegance of Boolean algebra. Others are motivated by clean geometrical structure that yields simple physical layouts. One example of the latter in the Johnson counter shown in schematic below. With n bits it is possible to represent 2^n values. This counter only uses 2n of those possible values, so it has a few more flip-flops, but almost no logic and no carry propagation, so it is fast and easy to route. The clr input resets all the flip-flops to zero.
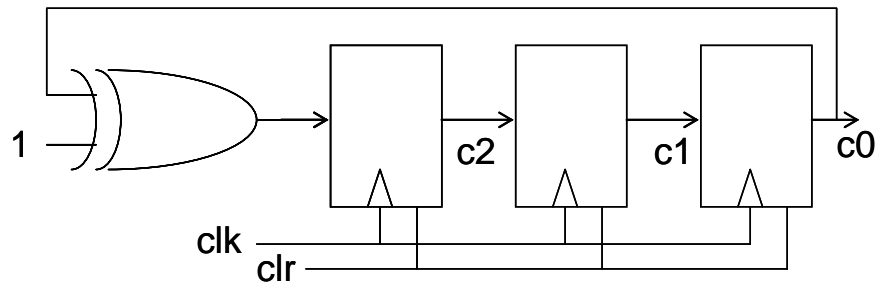
Generate the truth table how this counter "increments". This has four input columns (clr, c2, c1, and c0) and three output columns (the new c2, c1, and c0).

Show how this logical would be implemented on a 4-input, 3-output PLA.

Write a behavioral Verilog module for this counter using a bit vector for the counter value.

Problem 6: Draw the a schematic that would implement the behavior described be each of the follow verilog modules.

```verilog
module sifter1 (in, A,B,C, D, clk);
      input in, clk;
      output A,B,C;
      reg A, B, C;
      always @ (posedge clk) begin
          A <= in;
          B <= A;
          D <= C
          C <= B;
      end
endmodule

module sifter2 (in, A,B,C, D, clk);
      input in, clk;
      output A,B,C;
      reg A, B, C;
      always @ (posedge clk) begin
          A = in;
          B = A;
          D = C
          C = B;
      end
endmodule

module sifter2 (in, A,B,C, D, clk);
      input in, clk;
      output A,B,C;
      reg A, B, C;
      always @ (posedge clk OR reset) begin
          A = in;
          B = A;
          C = B;
          D = C;
          if (reset) begin A = 0; B = 0; C=0; D=0; end
      end
endmodule
```