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

EECS150, Spring 2013

Homework Assignment 2: Verilog Introduction, Sequential Logic Review
Due Feburary 12^{th}, 2pm

1. In homework 1, you implemented a one-hot code to binary number converter using Verilog continuous assignment. Now implement the same module using Verilog procedural assignment.

   (The converter takes in 8 bits and output 4 bits. If the input is a legal one-hot code, the lower 3 bits of the output are the corresponding binary number and the 4th bit is set to 0. If the input is not a legal one-hot code, we do not care about the lower three bits, and the 4th bit is a 1.)

2. Use behavioral Verilog to design a 3 bit up/down counter. It should have a control signal $\overline{Up}/Down$, which indicate counting up when set to 0, and counting down when set to 1.

3. The following code describes a 3-bit linear-feedback shift register (LFSR), which generates a repeating pattern of pseudo-random numbers.

```
module lfsr(R,L,Clock,Q);
        input [2:0] R;
        input L, Clock;
        output [2:0] Q;

        always@(posedge Clock)
                if(L)
                        Q <= R;
                else
                        Q <= {Q[1], Q[0]^Q[2], Q[2]};
endmodule
```

   (a) Draw the circuit that corresponds to the code.
   (b) If 001 is loaded into the LFSR initially, what is the sequence of numbers generated? List the binary numbers it generates, start from 001.
   (c) At which cycle does the sequence repeat?

4. The code below is similar to the code in question 3, but blocking assignments are used, draw the circuit. What is the pattern of numbers it generates?

```
module lfsr(R,L,Clock,Q);
        input [2:0] R;
        input L, Clock;
        output [2:0] Q;
```

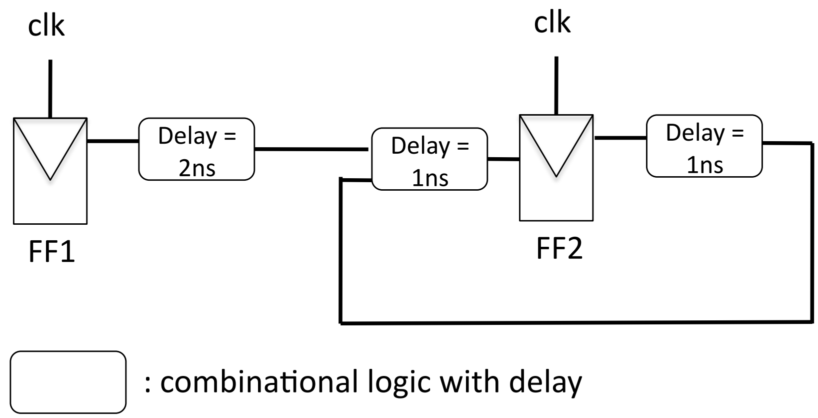
```
always@(posedge Clock)
        if(L)
                Q <= R;
        else
                begin
                        Q[0] = Q[2];
                        Q[1] = Q[0]^Q[2];
                        Q[2] = Q[1];
                end
endmodule
```

5. Given a 200 MHz clock signal, design a circuit to generate a 100 MHz clock. Draw a waveform diagram for the two clock signals, assuming a 1ns clock-to-q delay for any flip-flops, and 0.5ns delay for any gates.

6. For the circuit you have designed in question 5, assume clock-to-q delay of 1ns and setup-time of 0.9 ns, and also assume 0.5 ns of delay for any gates you have used,
   (a) What is the maximum frequency of the input clock for your circuit?
   (b) For the circuit to function properly, what's the requirement on the hold time of the FF?

7. A circuit has an input x, and an output y. The output is 1 when the input is 1 for three consecutive clock cycles.
   (a) Draw the state transition diagram for your FSM and show how do you encode each of your states.
   (b) Show the truth table showing x and the current state as inputs, and next state as output.
   (c) Draw the circuit of the FSM, using D-flip-flops and two-input gates.
   (d) Assuming clock-to-q delay of 1ns, 1ns delay for each gate, and 0.8 setup time for all D flip flops, how fast can you clock the FSM? Highlight the critical path.

8. In the circuit shown below, the worst case delay of each combinational component is shown inside the component. Assume clock-to-q of 1ns and setup time of 0.8ns.
   (a) What is the maximum clock frequency for the circuit?
   (b) If for some reason (for instance, a poor chip layout) the clock signal arrives at FF2 later than FF1 by 1.5ns, what is the maximum clock frequency?


clk
clk
Delay = 2ns
Delay = 1ns
Delay = 1ns
FF1
FF2
: combinational logic with delay