

Computer Science 141

Weekly Assignment #4

Due Wed Oct 14 @ 1pm

You may hand in your solution via email (cs141-2009-staff@eecs.harvard.edu) or in person at the beginning of lecture.

1 Required Reading

Sections 3.1-3.5, 4.4, 5.4 (ignore sections on VHDL)

2 Problems (60 total points)

Please show your work in your solutions.

Optional midterm preparation questions will be *marked like this*. You don't need to turn them in, but they will help prepare for the midterm.

If you need any help, please contact the TFs (Jian and Svilen) with questions or meeting requests at cs141-staff@eecs.harvard.edu.

1. (10 points) It is possible to complement a twos-complement number using the following algorithm: Proceed right to left. Complement each bit after the first 1. Design a 1-bit combinational complement circuit that can be cascaded (like the 1-bit full-adder circuit shown in class) to create a complement circuit for any input word width. Show how to cascade these 1-bit circuits to create a 4-bit complement circuit.

(Optional) It can be proven that this way of complementing works for an arbitrary number of bits. Do it.

2. (12 points) Write `multiplication8(a, b, c)`, a `verilog` module that multiplies two 2s-complement 8-bit numbers. Your module should take two numbers, `a[7:0]` and `b[7:0]`, and output one number, `c[15:0]`. You should assume you have a black-box module called `mul_unsigned8(a, b, c)`, which multiplies two unsigned 8-bit numbers, as well as modules `complement8(in, out)` and `complement16(in, out)`, which calculate the 2s-complement for 8-bit and 16-bit numbers, respectively. Think about how you would handle negative

numbers.

This problem requires you to instance several Verilog modules within your own module. Consult this week's lab handout and the Verilog tutorial in the Resources section of the class website for syntax and useful examples.

3. (8 points) Show how to implement a J-K flip-flop using a D flip-flop and some simple gates (AND, OR, NOT gates). Please draw your D flip-flop as we did in class (so that D is the data input, not the clock input).

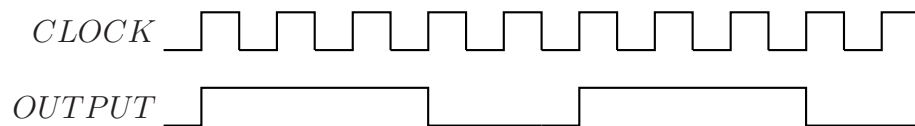
(Optional) A finite state machine has one input and one output. The output becomes 1 and remains 1 thereafter when the number of 1's on input exceeds the number of 0's on input by 2, or vice versa. Assuming this is to be implemented as a Moore machine, draw a state diagram for the machine. Do not implement further.

4. (14 points) Design a finite state machine for a sequential adder. The inputs for your FSM are A and B , the bits for the two numbers that you are adding. The output for your FSM should be S – the respective bit of the sum. Note that the inputs start with least significant bits of each number to add and may loop forever (*i.e.* there is no upper bound on the size of numbers that we're adding). For example, if $A = 111010011100000\dots$ and $B = 001110110100000\dots$, then the output should be $S = 110011100110000\dots$, which represents the addition of

$$1110010111 + 1011011100 = 11001110011 \quad (1)$$

Write out the state table for your FSM and minimized equations using D flip-flops.

5. (16 points) Design a circuit that will accept an oscillating clock signal as input and will produce an output that is 1 for three clock cycles and 0 for two cycles, repeated over and over. The timing diagram for this circuit is shown below. Draw a state diagram and a next state table, then implement your design using J-K flip-flops.



Since your circuit must “remember” where in the sequence of 5 zeros and ones it is, it must have at least 5 states, and therefore 3 or more flip-flops will be required. Make sure that the circuit functions properly by initializing it using the clear inputs on the FFs. By clever assignment of outputs to states (values latched in the flip-flops) you can minimize the gates

required to generate the state transitions and the output. For example, the output should be simply the value of one of the flip-flops. Think about the problem for a while before you settle on one design philosophy or another.