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

EECS150, Spring 2012

Homework Assignment 9: FSM, Counter and Adders
Due April 5^{th}, 2pm

1. Design a 4-bit synchronous down counter which counts in 2's complement representation, using simple gates and flip flops. Your counter should have a CE (count enable) signal and when it reaches the minimum number, it wraps around and start from the maximum number again. Show the steps you have taken to derive the circuit(Truth table for next state logic, generalization to multiple stages etc.)

2. You are given a counter, design a controller for the shift-and-add multiplier you have seen in class (lecture 20). Use mealy FSM & one-hot encoding. You should first list out the necessary control signals to be generated by your controller and briefly explain what each does to the datapath. Then show your state transition diagram and the gate level circuit for your FSM as well as the output logic.

3. You are to design an FSM circuit that has three inputs, clk, reset, and in, and an output, out. Your FSM would detect the sequences 0110 and 0101. The FSM continuously inspects its input and assert out for one cycle when either input sequence has been detected,otherwise it outputs a 0. Sequences can overlap. For instance, the input sequence 010110 will output a pulse for one cycle, and another pulse one cycle after that.

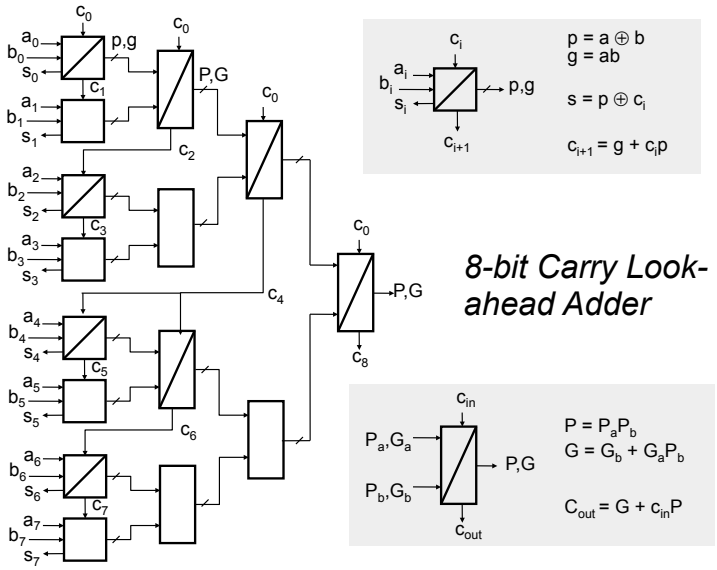
   (a) Show the state transition diagrams for both the Moore and Mealy implementations of this circuit.

   (b) For each implementation, show the gate level diagrams of the FSM using one-hot encoding, make sure you include the logic for generating the output as well.

4. The diagram below shows an 8-bit carry-look ahead adder.

   (a) Highlight the path with the longest delay, circle the starting signal and the ending signal.

   (b) If you are to implement this circuit with 6LUT, how many LUTs would you need. Assume each 6 LUT has a delay of 1ns, what is the delay of your circuit?



5. Carry-select adders.

   (a) Assuming that the select groups are all of the same number of bits, and the carry delay through a full-adder cell is equal to the delay of a 2-to-1 mux, what is the optimal size select group for a 32-bit adder?

   (b) Consider the possibility of applying the carry-select idea hierarchically. The idea is that a ripple adder of n-bits can be split and implemented as a carry-select adder with group size n/2 (implemented as three ripple adders of size n/2 along with muxes). Then each of the three adders of size n/2 could be split and implemented as a carry-select adder with group size n/4, etc. By continuing this process, eventually the adder size would be 1-bit and the process ends. Discuss the worst-case delay through this adder. Again using big O notation, how does the delay scale with n? How does the cost (amount of hardware - number of gates or transistors) scale with n?