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

EECS 150 R. H. Katz
Fall 2005

Problem Set # 10 (Assigned 14 November, Due 2 December)

1. Consider the design of two communicating Finite State Machines MASTER and SLAVE. The MASTER FSM asserts its output GO when it wants the SLAVE FSM to begin processing. The SLAVE FSM asserts its output DONE when it has completed processing. The MASTER keeps asserting GO until it detects that DONE is asserted. The SLAVE keeps asserting DONE until it detects that GO is no longer true, and returns to a configuration to await the next asserted GO signal.
   (a) Draw a state diagram for a MEALY MACHINE implementation that captures the signaling protocol between MASTER and SLAVE FSMs. Don't worry about exactly what the SLAVE does between GO and DONE; just assume that there are several states there eventually leading to the sequence of states that generate the DONE handshake.
   (b) Draw the timing diagram for a complete GO/DONE sequence. Assume a common clock for the MASTER and SLAVE, and include this signal in your timing diagrams.
   (c) Will your solution still work if the MASTER and SLAVE operate with different clocks? Illustrate your answer with a modified timing diagram with independent clocks and explain why or why not.

2. A barrel shifter is circuit that allows its input to be shifted any number of positions. For example, a 4-bit rotating barrel shifter can shift its inputs I_3, I_2, I_1, I_0, zero, one, two, or three positions to the right as indicated by the shift control inputs S_1, S_0. The outputs become:

S1	S0	O3	O2	O1	O3
0	0	I3	I2	I1	I0
0	1	I0	I3	I2	I1
1	0	I1	I0	I3	I2
1	1	I2	I1	I0	I3

   (a) Using conventional multiplexers, decoders, and logic gates, show how to implement the barrel shifter as a purely combinational logic circuit.
   (b) Take a different approach by implementing the barrel shifter as a Finite State Machine plus simple datapath. Define the control signals between your controller and datapath, and show your detailed state machine for implementing the correct behavior of the barrel shifter as indicated above.
   (c) Compare and contrast your answers to 2(a) and 2(b). Which is faster? Which utilizes more hardware? Which is the better design and under what circumstances?

3. The traditional binary adder operates on all of its input bits at the same time, calculating the sum output bits in parallel. Consider an alternative way to implement a binary adder using a so-called bit serial approach. The two numbers to be added are presented to a Finite State Machine one bit at a time, with the lowest order bits presented first. The Finite State Machine produces at its output the lowest order bit of the sum, then the next higher order sum bit, and so on, until all of the input bits have been processed and all of the output sum bits generated.
   (a) Design a simple datapath for the bit-serial adder down to the gate level, and identify the interface between your control finite state machine and the datapath. Consider how to deal with carry-in and carry-out in your design.
   (b) Show your state diagram for a 4-bit bit-serial adder, where the outputs of the state machine are the control signals of the datapath you designed in 3(a).
   (c) Demonstrate how your subsystem works by showing step-by-step how it executes the summation of 0110 and 1100. The carry-in to the low order bit is initially zero.