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

EECS150 J. Wawrzynek
Fall 2002

Homework #4

This homework is due on **Friday September 27th by 2pm**. Homework will be accepted in the EECS150 box outside of 125 Cory Hall. Late homework will be penalized by 50%. No late homework will be accepted after the solution is posted. **Make sure to put your *lab* section number on your homework.**

1) Based on the design of the combination lock presented in class, consider the design of a lock that accepts a three digit binary-coded-decimal (BCD) string for the combination, one digit at a time. As with the lock in class, assume that we use a separate circuit for "decoding" the combination.
    a) Draw the state transition diagram for the new lock.
    b) Show the circuit for decoding the combination (using ANDs, ORs, and Inverters) for the combination 5,7,1.
    c) Draw the circuit diagram for a one-hot encoded implementation of your lock. Assume that the flip-flops have reset and preset inputs.

2) Design a bit serial 2's complementer as a Moore-style FSM with one input and one output. The circuit accepts a string of bits from the input (least significant bit first) and generates the 2's complement at the output. The circuit can be reset to start and end the operation. Show your work by following the standard steps for FSM design: state transition diagram, symbolic table, encoded table, Boolean equations, and circuit diagram. You may assume that the flip-flops have reset and preset inputs.

Hint: The method for generating 2's complement is not the usual method of inverting all the bits and adding one. You can discover an alternate method by looking at several simple examples of binary numbers and their respective 2's complement representation. Note where the bits are the same and where they are different.

    a) use as few flip-flops as possible.
    b) use one-hot encoding with one flip-flop per state. For this part just show the circuit diagram.

3) From Mano: Problems 5-16.

4) Consider the design of a Moore-style FSM with the following description: The FSM has 2 inputs, IN and RESET, and four outputs X0, X1, X2, and X3. After the RESET input is asserted the FSM accepts a stream of bits applied to the input IN. Bit patterns of the form 1**1 are "recognized" by the FSM, where each "*" may be either a 0 or a 1. After the rising edge of the clock on the cycle when the final 1 is passed in, the FSM asserts one of X0, … , X3 to indicate 00, 01, 10, or 11 for "**". All other bit patterns result in no outputs being asserted. The RESET signal can come at any time and initializes the FSM to again look for bit patterns. Recognizable bit patterns do not overlap. For example, the bit stream (with time moving left to right through this bit string) 0101100100111010 contains only three recognized bit patterns, 1011, 1001, and 1101.

   Draw the state transition diagram and fill in a symbolic state transition table. "Symbolic" means that you do not need to encode the state – just make up stat names.

5) From Mano: Problems 3-34, 3-35, 3-36, 3-37, 4-38, 4-43, 4-44, 5-21, 5-22, 5-25, 5-30.

6) Using the World Wide Web assemble a list of software tools related to Verilog (a google search would be a good way to start). For each tool briefly describe its function and include a URL. Try to list at least 10 tools.