CprE 281: Digital Logic
Extra Credit Homework

Due: During your last lab for this semester.

What to submit: Submit your solutions on paper to your lab TAs either before or along with your final project. You will also have to demonstrate your designs to your lab TAs on the boards in the lab. In other words, all problems have a paper and a hardware component.

For all problems please use positive-edge triggered memory elements.

1. Design and implement a finite state machine that detects the pattern 101 in its 1-bit serial input stream. Please provide the state diagram, state table, the state-assigned table, and the final implementation of the sequential circuit. Also, please provide the K-maps and the logic expressions for the next state logic and for the output logic. Please use the same naming conventions that were used in the class slides.

2. Design and implement a finite state machine that detects the pattern 1001 in its 1-bit serial input stream. Please provide the state diagram, state table, the state-assigned table, and the final implementation of the sequential circuit. Also, please provide the K-maps and the logic expressions for the next state logic and for the output logic. Please use the same naming conventions that were used in the class slides.

3. Design and implement a finite state machine that computes the Hamming distance between the 3-bit histories of two serial input data streams.

The Hamming distance between two strings of equal size is defined as the number of positions at which the strings differ. For example, the Hamming distance between the strings 000 and 010 is 1 because the strings differ in exactly one position, that is, in the second position.

Note that the possible range of Hamming distances between two 3-bit numbers is 0 (meaning the strings are identical) to 3 (meaning the strings differ at all positions).

Your circuit must accept two serial input data streams. This means that the machine is receiving one bit of input from each of the two streams at the beginning of each clock cycle.
The finite state machine must keep track of a history of the last three inputs from each of the streams so that it can calculate the Hamming distance between them.

Note that the FSM should initially be outputting a Hamming distance of 0.

4. Design and implement an Up/Down counter using a FSM model. This counter will count up (0, 1, 2, 3) if the input $w$ is 0, and count down (0, 3, 2, 1) if the input is 1. In both cases, the counter will continuously cycle through the count without resetting it. Suppose that the counter is currently 2 and the counter was counting up and then switched to count down. Then after the next clock the count should be equal to 1.

Fill in the state-assigned table on the answer sheet and use it to design the circuit.

Use D flip-flops for the device’s memory. The $w$ input will need to be assigned to a toggle switch. The output needs to be represented on a seven-segment display. Use the DE2 board to verify your design. When you are convinced your design is working correctly, demonstrate it to the TA.

The weight of each problem is as follows: 1(1%), 2(1%), 3(2%), and 4(2%).

Good Luck!