EECS 151/251A Homework 3

Due Friday, September 27th, 2019

Problem 1: Logic Simplification [4 pts]

Refer to this truth table containing 4 inputs (A, B, C, D) and 1 output (Q):

A	B	C	D	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

1. Write a sum-of-products directly from the truth table.
2. Use a Karnaugh Map to simplify the logic and write the simplified sum-of-products and product-of-sums representations.
3. Using the sum-of-products representation, draw the circuit that implements this function. Transform this circuit such that it is made up only of inverters and NAND gates.

Problem 2: K-Map [5 pts]

Refer to this truth table containing 6 inputs (A, B, C, D, E, F) and 1 output (Q):

A	B	C	D	E	F	Q	A	B	C	D	E	F	Q
0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0	0	0	1	0	1
0	0	0	0	1	1	0	1	0	0	0	1	1	X
0	0	0	1	0	0	1	1	0	0	1	0	0	1
0	0	0	1	0	1	1	1	0	0	1	0	1	1
0	0	0	1	1	0	1	1	0	0	1	1	0	1
0	0	0	1	1	1	1	1	0	0	1	1	1	1
0	0	1	0	0	0	0	1	0	1	0	0	0	1
0	0	1	0	0	1	0	1	0	1	0	0	1	1
0	0	1	0	1	0	X	1	0	1	0	1	0	0
0	0	1	0	1	1	0	1	0	1	0	1	1	0
0	0	1	1	0	0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	0	1	0	1	1	0	1	0
0	0	1	1	1	0	0	1	0	1	1	1	0	0
0	0	1	1	1	1	0	1	0	1	1	1	1	0
0	1	0	0	0	0	0	1	1	0	0	0	0	1
0	1	0	0	0	1	0	1	1	0	0	0	1	1
0	1	0	0	1	0	1	1	1	0	0	1	0	1
0	1	0	0	1	1	0	1	1	0	0	1	1	0
0	1	0	1	0	0	1	1	1	0	1	0	0	1
0	1	0	1	0	1	X	1	1	0	1	0	1	1
0	1	0	1	1	0	1	1	1	0	1	1	0	1
0	1	0	1	1	1	0	1	1	0	1	1	1	0
0	1	1	0	0	0	X	1	1	1	0	0	0	X
0	1	1	0	0	1	X	1	1	1	0	0	1	1
0	1	1	0	1	0	0	1	1	1	0	1	0	0
0	1	1	0	1	1	0	1	1	1	0	1	1	0
0	1	1	1	0	0	1	1	1	1	1	0	0	1
0	1	1	1	0	1	1	1	1	1	1	0	1	1
0	1	1	1	1	0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	0	1	1	1	1	1	1	0

1. Use a 6-variable Karnaugh Map to derive a simplified boolean function from this table using sum-of-products method.
2. Use a 6-variable Karnaugh Map to derive a simplified boolean function from this table using product-of-sums method.
3. Are the two functions you derived equivalent? If not, why?

Problem 3: FSMs - Pattern Detection [6 pts]

In this problem, you are asked to design a pattern detector circuit that aims to extract the pattern "10010" from an input serial bitstream. The circuit receives a new bit every clock cycle from its input "*in*" and has an output "*out*" used to indicate a pattern has been detected. *out* is pulled high for 1 clock cycle to indicate a pattern is detected, and then pulled low to prepare for the next match.

1. Draw the state diagram of this circuit, marking the transition conditions and output values. The detected patterns should not overlap, i.e. it should take at least 5 more cycles after a pattern is detected to detect another pattern.
2. Write the Verilog that corresponds to your circuit. Simulate your circuit and show the waveform with two pattern detection events.
3. Draw an updated state diagram of your circuit if we now want overlapping patterns to be detected as two patterns, i.e. "10010010" should have two pattern detection events in it. Use a Moore state machine for your implementation.
4. Modify your answer to the previous part to implement a Mealy state machine.
5. Make the necessary modifications to your Verilog and show the updated simulation results showing an overlapping detection event (2 detections seperated by less than 5 cycles).