ECE/CS 250 – Recitation #5 – Prof. Sorin Logic Design with Logisim

Objective: In this recitation, you will learn how to design digital logic and use Logisim for the design and simulation of digital circuits.

Complete as much of this as you can during recitation. If you run out of time, please complete the rest at home.

1. Task 1: Download Logisim and Work Through Tutorial

In this course, you will design and test logic circuits with Logisim. Please download Logisim and work through (at least part of) the tutorial to help you understand how to use this tool.

http://www.cburch.com/logisim/index.html

2. Task 2: Design a Small Combinational Circuit

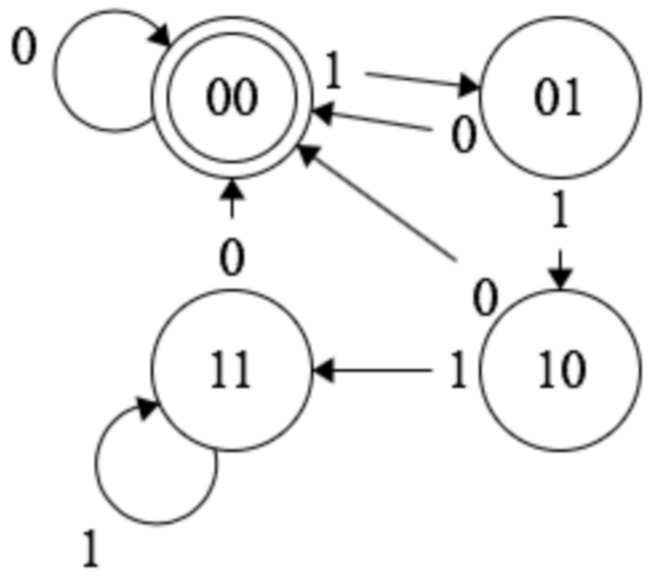
In logisim, make and simulate a circuit that implements the following truth table (3 inputs, 1 output). Just do a simple product-of-sums – no need to optimize. Use only basic logic gates (NOT, AND, OR, NAND, NOR, XOR) from the logisim library. You must simulate your circuit to test whether it behaves as expected in all cases.

in1	in2	in3	OUTPUT
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

3. Task 3: Design a Small Sequential Circuit

Design and simulate a sequential circuit with one input (and a clock input) and the following behavior. If the input has been equal to 1 for the three previous cycles, then the output is 1. Otherwise, the output is 0. Use D flip-flops from the logisim library. You must simulate your circuit to test that it behaves as expected.

Please use the systematic methodology for developing the FSM. Start with a state transition diagram, write out the truth table, then implement it.



Current	State	Output	Input (A)	Next State	
Q1	Q0			D1	D0
0	**0**	0	0	0	0
0	**0**	0	1	0	1
0	**1**	0	0	0	0
0	**1**	0	1	1	0
1	**0**	0	0	0	0
1	**0**	0	1	1	1
1	**1**	0	0	0	0
1	**1**	1	1	1	1

Product of Sums:

Out: $Q1 \cdot Q0$

D1: $A \cdot !Q1 \cdot Q0 + A \cdot Q1 \cdot !Q0 + A \cdot Q1 \cdot Q0 = A \cdot (Q1 + Q0)$

D0: $A \cdot !Q1 \cdot !Q0 + A \cdot Q1 \cdot !Q0 + A \cdot Q1 \cdot Q0 = A \cdot ((Q1 \text{ xnor } Q0) + (Q1 \cdot !Q0))$

4. Task 4: A Little Bit of Logic Optimization

Time permitting, optimize the logic from Task #2 and implement the optimized circuit.