//
Seongwoo Choi
scho29@ucsc.edu
CMPE - 012
T/Th 11:00AM - 1:00PM
//

Title:
Lab 1: Intro to Logic with Multimedia Logic

Purpose:
The purpose of this lab is to learn how to build a logic circuit using a Windows software, MultiMedia Logic.

Procedure:
The methodology of using MultiMedia Logic software to build a virtual logic circuit was the main goal of this lab assignment. First, scheme through the truth table that was provided on the lab 1 document and then find out if the truth table results are correct or not. Second, translation of the truth table into a MML (MultiMedia Logic) layout. Third, on MultiMedia Logic software, test the logic circuit and make sure the result works with corresponding to the truth table that was provided by the lab 1 assignment PDF.

Algorithms and other data:
Using the truth table and translating from the truth table to MultiMedia Logic layout is the mission of this lab. For Part B, translation of the truth table was necessary. So, translating the truth table and put it as a logic circuit on a piece of paper were helpful. Three inputs were placed in the front and then six inverters were placed and then three AND gates and one OR gate were placed.

For Part D, the random button was placed behind the random number generator. This generator then was connected to two LED flashes. For example, one string from the random number generator was connected to LED A and then the same string from the random generator was connected to XOR gate. This XOR is connected with Switch A and another XOR gate. For another string that is connected to another LED flash (LED B from the random number generator) is also connected to another XOR gate. This time, this XOR gate is connected to Switch B and the same XOR gate that is connected with Switch A.

Afterward, these two XOR gates are connected with another XOR gate. So this means that there are two truth table to figure out whether the data of the user inputs and the random data are same or not. If both are same, then third LED flashes.

What went wrong or what were the challenges?
Getting familiar with a software was a challenge since it is required to work on the MultiMedia Logic, and also the translation of the truth onto the MultiMedia Logic layout was very challenging in the beginning.

Other information:
Discuss the original design. How did you get the logic design from the truth table?
During the lab session, one of TA's showed us how to get the logic design from the truth table. Basically, I looked through the truth table and found which section has one. There was one section that shows one as a result when A, B, and C are zero. I placed three switches that represent A, B, and C. When A indicates zero, based on the truth table, I connected it to inverter. Afterward, I connected switch B to other inverter since it shows another zero value and also I connected switch C to another inverter. For the next section where the result shows one, there were A and B that indicated to zero. So, I connected to two other inverters and I connected switch C to AND gate because it was showing one. I connected switch A to the last inverter and other two switches to I placed six inverters eventually. This is how I got the logic design from the truth table.
How many transistors from in the original design from part B?
In total there are 44 transistors in the original design from part B.

Discuss the changes you made in the design
From the truth table, finding which combinations of A,B,C shows the result as one was important. There were in total of three cases where the results show as one. One means the LED light flashes. There were ABC, ABC', and AB'C' showing one as results. So, translation of those three cases into a drawing was necessary to clarify how to draw a logic circuit on MultiMedia Logic software. On the drawing, it was possible to connect inputs A,B,C on three inverters, and then connect those inverters to one AND gate. And then, those same inputs A, and B were connected to two other inverters and these were connected to AND gate, but the input C was connected to AND gate directly. There was one last inverter and it was connected to input A and then inputs B, and C were connected directed to the other AND gate. In total there were six inverters and three AND gates. Lastly, those three AND gates were connected one OR gate and this OR gate was connected to LED flash.

On the second part of the PART B, the replacement of inverters to NAND gates was necessary. Those three inputs remained same, but each time each input was connected two transistors since

there were required to connect more than one more transistor to correct the phases from NAND output into AND output.

How many transistors in the improved design from part B?
There are 62 transistors in the improved design from part B.

Why do AND and OR gates have more transistors than NAND and NOR?
Because it requires at least one more transistor to correct its phase from NAND output into AND output.

Discuss how you reduced the circuits in part B to the final circuit in part C. How many transistors in the final circuit?
To do this part of the assignment, using the sum of product was necessary. Since the sum of product is ~a(~b +c), translation of sum of product into MultiMedia Logic was required. Input A was connected to an inverter and then it was connected an AND gate. Input B was connected to an OR gate and it was then connected to the same AND gate that was connected with Input A's inverter. Input C was connected to another inverter and then it was connected to the same OR gate was connected with input B. This OR gate was connected to AND gate and then the AND gate was connected to an LED.

Make some sort of guess on how that random number generator works? How can thing really be random in a computer with logic gates being so, well, logical?
Not sure how a computer would use random number generator, but a computer might need for user input on a computer game, which an user needs to manipulate the game scenario.