Name: Gavin Mack – gmmack

Lab section: 4 TT TA: Ehsan Hemmati

Due 1/19/2014

Lab Partner: None

Title:

Lab 1: Intro to Digital Logic

Purpose:

The purpose of this lab is to use MML to demonstrate DeMorgan's law by showing how an inverter and an AND gate produce the same output, to use sum of products to implement a truth table, and to use logic minimization to create equivalent logic structures that are simpler.

Procedure:

For part A I connected two switches to an AND gate and routed the output of the AND gate to an LED, then added an inverter between the inputs and the AND gate. I then routed the same inputs through a NOR gate and saw that the outputs were the same, proving DeMorgan's law. For part B I first took the sum of products to create our logic equation, then used the equation to create a circuit with three inverters, three AND gates, and an OR gate. I then applied DeMorgan's law to my logic equation to change it to use only NAND gates, and created a circuit with three inverters and four NAND gates. For part C I used Boolean algebra to minimize the logic equation, then based off the minimized equation created a circuit with two inverters, two AND gates, and an OR gate.

Algorithm and Other Data:

In part B I used sum of products to create the logic equation A'B'C' + A'B'C + A'BC = OUT. This equation was used to create a circuit with three inverters, three AND gates, and an OR gate. I then double NOT'ed the left side of the equation, and applied DeMorgan's law (A'B') = (A+B)' to change my logic equation to [(A'B'C')' * (A'B'C)' * (A'BC)'] = OUT. Based off of this new equation I created a circuit with three inverters and four NAND gates. This new implementation only uses 22 transistors, while the first implementation uses 30 transistors. In part C I minimized my logic equation by factoring out A'B' so that my equation became A'B' (C'+C) + A'BC = OUT. This simplifies down to A'B' + A'BC = OUT, which can then be changed to A' (B' + BC) = OUT. I used this minimized equation to implement a circuit with two inverters, two AND gates, and an OR gate.

What Went Wrong or What Were the Challenges:

My only real challenge in completing this lab was figuring out the correct use of DeMorgan's law in order to change my original logic equation from part B to use only NAND gates. I originally tried NOTing only the left side of the equation once, because I did not realize that doing that unbalances the

equation. You must either NOT both sides, or double NOT a single side so that the equation remains balanced.

Other Information:

How many transistors does each implementation take?

The AND and OR gate implementation in part B takes 30 transistors, but the only NAND gate implementation only takes 22 transistors. The final minimized implementation also takes 22 transistors.

Conclusion:

An inverter and an AND gate will produce the same output as a NOR gate because of DeMorgan's law. I was surprised that my second implementation in part B which used only NAND gates used the same number of transistors as the minimized implementation. I expected the minimized implementation to use fewer transistors, although this discrepancy may be because I did not correctly fully minimize my logic equation. I discovered that minimizing a logic equation is the easiest way to minimize the number of transistors in one's implementation.

Extra:

The directions in number three of part A seemed vague to me. It took reading through it a few times for me to understand that we wanted the same inputs but a different output for our NOR gate.