CMPE12L Lab Report 1

Intro to Digital Logic

Ryan Coley

1332490

Lab Section 2

MW 2:00pm - 4:00pm

10/11/14

Description:

The purpose of this lab was to get familiar with digital logic by making a few simple schematics and learn how to use digital logic that follows a truth table. This was accomplished by following a tutorial on how to use the program Multimedia Logic, then after wards schematics were drawn to illustrate De Morgan's Law, implement Sum of Products using only AND and OR gates in the first design and then using only NAND gates in the other, minimize the logic from the previous schematic using Boolean Algebra, and finally making a guessing game using all of the information that was learned from the previous parts.

Part A: Using Multimedia Logic

In this part of the lab, there were two parts to it. The first part was to follow the tutorial from the help section to first learn how to use the product. The point of going through the tutorial was to learn how to use the program and make simple circuits such as a switch connected to an LED and a switch connected to a not gate then connected to an LED. These circuits were used to show how simple it is to make a logic circuit. The next part was to show how De Morgan's law works. This was done by setting up two switches to a NOR gate then the OR gate to an LED. Then the same switches were setup to NOT gates that were then hooked up to an AND gate which was hooked up to the led. The LED lit up exactly the same time as when it would have lit up from just the first circuit. This shows that the truth table (Table 1) is the same and that a circuit can have multiple different configurations that have the same output.

Table 1

Input A	Input B	Result
0	0	1
0	1	0
1	0	0
1	1	0

Part B: Implementing Functions as Sum of Products (SOP)

In this part of the lab, the truth table (Table 2) had to be implemented using AND and OR gates and then implement the same truth table using only NAND gates. In order to do this, the rows with one as the output were used to get the correct scheme. The equation that was implemented was: A'BC'+A'BC+AB'C+ABC' where ' is a NOT gate, + means an OR gate and if the letters are next to each other they are connected by an AND gate. The first implementation used 52 transistors. But for only the NAND gate part, the NOT, AND, and OR gates were converted to only NAND gates. This was done by connecting both inputs together of a NAND gate to create a NOT gate (Figure 1), to create an AND gate, a NOT gate was connected to another NAND gate to make an NOT NAND gate (Figure 2), and finally an OR gate was created by connecting the output of a NAND to the input of another and the first NAND gate has the inputs connected together (Figure 3). The same equation was used to create the circuit but it just used different gates. This implementation used 58 transistors.

Table 2

Input A	Input B	Input C	Result
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Figure 1

Figure 2

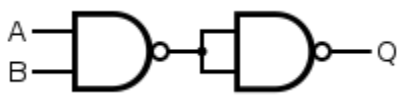
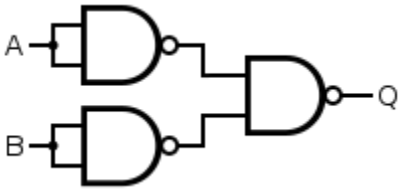


Figure 3



Part C: Logic Minimization

For this part of the lab, Boolean Algebra had to be used to reduce the size of the circuit that was made in Part B. From the first equation A'BC'+A'BC+AB'C+ABC', A and A' was factored out to get A'(BC'+BC) + A(B'C+BC'). Then using a Boolean Algebra identity C was removed from the first part of the equation to get A'B + A(B'C+BC'). The (B'C+BC') can be recognized as an XOR gate because only one can be active at a time to get A'B+A(B⊕C) so instead of ten gates, it was minimized down to using only 5 gates using 26 transistors.

Part D: Guessing Game

The point of this part was to design a circuit that makes a guessing game using a random number generator, user input and an LED to tell if the user got it correct. The truth table in table 3 shows when the user will be correct. The random number generator generates a number by using an algorithm and inputting a random seed such as the date and outputs the number through the output signals. The user then selects their guess and flips another switch to see if they are correct.

Table 3

User Input A	User Input B	Random A	Random B	Check Guess Input	Result
0	0	0	0	1	1
0	1	0	1	1	1
1	0	1	0	1	1
1	1	1	1	1	1

Conclusion:

This lab was used to make the understanding of logic diagrams an easy thing to do by using Boolean Algebra to minimize a schematic, make a guessing game, and make the same truth tables two different ways but have the same output.