Jason Ou (jaou 1385128)
CMPE 12L
20 Jan 2015
TA: Daphne Gorman
Section: 6

Lab Write-Up #1

OVERVIEW

The first lab is for learning how to use MultiMedia Logic (MML) application in Windows to simulate a logic schematic. The MML program is used to build some simple circuits to help build a better intuition on how logic works.

PROCEDURES

PART A:

Do the MultiMedia Logic (MML) tutorial. After doing the tutorial, use text tools and create a logic gate that demonstrates De Morgan's Law.

PART B:

Implement the Sum of Products (SOP) and solve for the truth table in the PDF. In the first implementation, use only OR, AND, and NOT Gates. Implement it again using NAND Gates.

PART C:

Simplify the Sum of Products expression in part B by using Boolean algebra to minimize it. Afterwards, draw the final circuit.

PART D:

Create a guessing game where the user has to guess what the two random numbers generated by the push button and random number generator creates. If the user guesses the two numbers correctly, make it so the LED switches ON.

RESULTS

PART A:

The tutorial taught the very basics of MML such as using tools, buttons, and running the circuit simulator. De Morgan's Law is ‘(AB) = (‘A + ‘B).

PART B:

Implementation one of part B was made using OR, NOT, and AND Gates by converting the Sum of Products expression into circuits. Implementation two of part B was made using NAND and NOT Gates, converting OR and AND Gates into NAND Gates.

PART C:

The Sum of Products expression could be reduced using Boolean algebra. After reducing the expression, we created a simplified version of the original circuit which utilized only 16 transistors.

PART D:

By using two XNOR Gates, we could compare the user input (two on/off switches) with the random number generator making it so that when the user input matches random number generator, it will return 1. When both of the user inputs matches 1, we use an AND gate to confirm and turn on the LED to show that the user got the answer correct.

Jason Ou (jaou 1385128)
CMPE 12L
20 Jan 2015
TA: Daphne Gorman
Section: 6

DISCUSSION

#1) **Discuss the original design. How did you get to the logic design from the truth table?**

- The original design was made using a mixture of NOT, AND, and OR gates. The logic of this design was made by using the Sum of Products where we first find the inputs with the output of 1 and by using those inputs, we derive equations for each of the inputs and combine them all together with OR gates. If the input was [0, 0, 0] then we get the equation (A'B'C'). If there was another input such as [1, 1, 1] then the equation would be (ABC) and combined with the first input, it would be (A'B'C' + ABC).

#2) **How many transistors in the original design from part B?**

- 3 NOT Gates, 1 Input ; 4 AND Gates, 3 Inputs ; 1 OR Gate, 4 Inputs
- #Transistors = 3(2*1) + 4(2*3 + 2) + 1(2*4 + 2) = 6 + 32 + 10 = 48.
- The original design from part B uses **48** transistors.

#3) **Discuss the changes you made to your design.**

- Changes that I made to my design were changing all of my AND and OR logic gates into NAND gates.
- A AND B turns to NOT[NOT(A AND B) AND NOT(A AND B)]
- A OR B turns to NOT[NOT(A AND A) AND NOT (B AND B)]

#4) **How many transistors in the improved design from part B?**

- 3 NOT Gates, 1 Inputs ; 4 NAND Gates, 3 Inputs ; 8 NAND Gates, 2 Inputs ; 1 NAND Gate, 4 Inputs
- #Transistors = 3(2*1) + 4(2*3) + 8(2*2) + 1(2*4) = 6 + 24 + 32 + 8 = 70.
- The improved design from part B uses **70** transistors.

#5) **Why do AND and OR gates have more transistors than NAND and NOR?**

- A NAND gate is equivalent to inverter gates and an OR gate, so by DeMorgan's law, '(AB) = (A' + B'). So it takes 2 less transistors than using AND and OR.

#6) **Discuss how you reduced the circuit in part B to the final circuit in part C. How many transistors in the final circuit?**

- I reduced the circuit in part B using **Boolean algebra**.
- **Original Equation**: A'B'C' + A'B'C + A'BC + ABC
- **Negation Law**: A'B' + A'B' + BC + BC
- **Idempotent Law**: A'B' + BC
- **DeMorgan's Law**: '(A + B) + BC ← Final reduced equation using Boolean algebra.
- 1 NOR Gate, 2 Inputs ; 1 AND Gate, 2 Inputs ; 1 OR Gate, 2 Inputs
- #Transistors = 1(2*2) + 1(2*2 + 2) + 1(2*2 + 2) = 16.
- The final circuit has **16** transistors.

#7) **Make some sort of guess on how that random number generator works? How can things be really random in a computer with logic gates being deterministic?**

- The random number generator may be using a series of logic gates that are constantly changing through some sort of logic implementation, making it not so random, but at the same time, always changing due to some algorithm that causes the gates to change such that even when you input the same values over and over, since the logic gates change, the output comes out differently.

Jason Ou (jaou 1385128)
CMPE 12L
20 Jan 2015
TA: Daphne Gorman
Section: 6

CONCLUSION

This lab was used to learn the basics to MML and converting truth tables into Boolean expressions into circuits. We also learned the advantages of using NAND over AND and OR Gates. Also, we learned to minimalize Boolean expressions.