CS 250 Spring 2017 - Lab 01 Digital Logic and Gates

The policy for all lab assignments this semester is as follows. Please sign, which you may do by typing in your name.

*In the following I have not represented the work of another person as my own nor have I knowingly or actively assist another person in violating this standard.*

(Signed) Craig Ruble

## In-Lab Assignment

1. [50 points] Show your TA the working circuit that you have you build in lab today. After demonstrating your circuit, disassemble it except for the 10 ohm current-limiting resistor. Keep the various pieces of hookup wire to reuse in your next circuit.

## Take-Home Lab Assignment (due at the beginning of your lab session next week)

Build a combinatorial digital logic circuit using components in your lab kit to implement the following Boolean function.

- The circuit should take in three one-bit inputs called A, B, and C, in the form of three voltage divider circuits built from a 10,000 ohm resistor and a push button switch. These three inputs should be built as active-high, that is, pushing a switch button should create a high voltage to send to the combinatorial logic portion of the circuit. NOTE: Keep in mind that the inputs you built in lab were active low (1 when not pressed/ 0 when pressed).
- The combinatorial circuit to be built treats the input bits A, B, and C as representing an unsigned decimal integer from zero (inputs ABC = 000) to seven (inputs ABC = 111). When the number of letters in the name of the decimal number is even, e.g., ABC represent an element of {zero, four, five}, then the circuit output is to be a logic 1 represented by a high voltage. When the number of letters is odd, e.g., {one, two, three, six, seven}, the circuit output should be a logic 0 and a low voltage.
- The combinatorial logic circuit must be built using only two-input NAND gates.
- The circuit should light an LED when the logic function value is 1 and the LED should be unlit when the logic function value is 0.

2. [10 points] Write out the truth table for the logic function to be implemented by the circuit. [insert a table here in this Word document.]

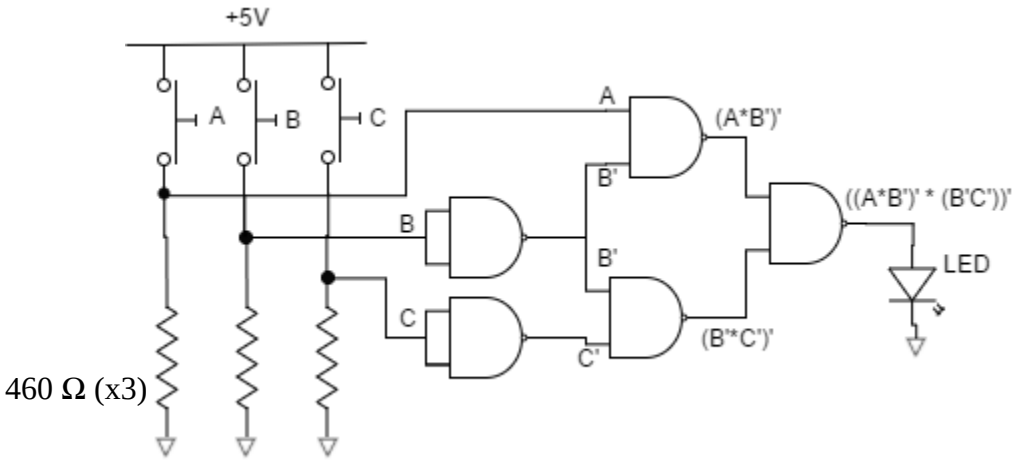
| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

3. [10 points] Use a Karnaugh Map to create a Sum of Products (SOP) Boolean expression for the logic circuit in terms of two-input AND and two-input OR operations. Then use DeMorgan's Law to convert your SOP expression into an expression using only 2-input NAND gates. Write both expressions here.

| c \ ab | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |

SOP you want to circle the 1's in the most optimized form (i.e. groups of 4 first, then 2, then 1). The expression as AND and OR is AB' + B'C'. Using DeMorgans (double negation), we can turn this into a expression of NAND. [((AB') + (B'C'))']'. Distributing the one negation, you end up with the NAND realization. ((AB') * (B'C'))' this is realized with 5 NAND gates as two inputs need to negated with a NAND as an inverter.

4. [10 points] Draw the circuit schematic for the NAND implementation and be sure to include the input voltage dividers and the output LED. Your diagram will have much in common with the diagram for the circuit that you build during lab.




5. [30 points] Implement circuit on your breadboard and bring it to lab next week to be checked by your TA at the start of lab. **You will not get to work on this during next week's lab; lab take-home work will be checked at the beginning of lab.**