Circuits Lab

Names:_____

Objectives
Learn how circuits work, different properties of circuits, and how to derive truth tables, circuits from functions and simply circuits using Boolean circuit equivalence..

Introduction
This lab will concentrate on building and understanding circuits. It will use a tool developed by two former computer science majors here at Gordon College called *Circuit Sandbox*.

The course web site's notes entitled "*Switches, Gates and Circuits*" which can be found at will be useful to you throughout the lab.

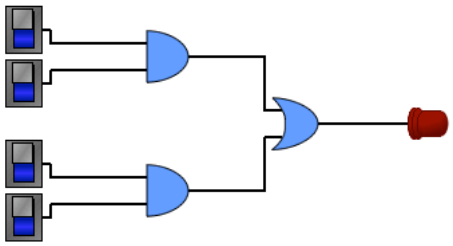
We will start off by creating a simple circuit to help you get familiar with how Circuit Sandbox works. Then we will move into more depth in exploring properties of circuits and then finally look at a couple of mystery circuits that are essential to computing.

Note: In order to print from circuit sandbox - you use the *File/Export Model as Image* command. Save the image file in a known location and then open it and print. (*Print and attach all the circuits you create.*)

Note: TA will initial at various places in lab – (TA_____)

Part 1: Create a simple circuit in Circuit Sandbox

1. Start Circuit Sandbox. (Click on the beach ball)
2. Build a circuit that looks like the one in the diagram below.



3. Run and play with the switches and answer the questions below.

Observations

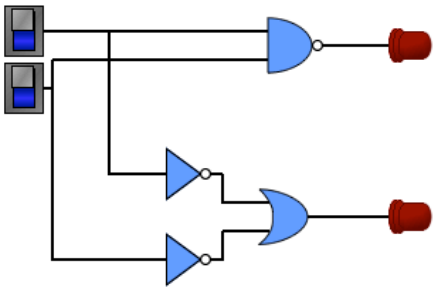
Record the truth table that the above circuit produces. The inputs are numbered from top to bottom so input A is at the top etc.

Input A	Input B	Input C	Input D	Output

What is the Boolean expression that the truth table and circuit represents? Let output be M. (TA____)

Part 2: Circuit Properties

1. In Circuit Sandbox draw a circuit containing two switches.
2. Draw a NAND-gate and an OR-gate each with their own LED output.
3. Draw two inverters (NOT-gates) hooked up to the inputs of the OR-gate.
4. Hook the inputs on the NAND-gate and the inverters to the switches.
5. The diagram should look like the one below.



6. Now run the circuit and try all the different switch positions.

Observations
What do you observe? Record the results in the truth table below.

A	B	Output 1	Output 2

Which of the Boolean laws does this demonstrate? (see appendix A) (TA_____)

Part 3: Create a Boolean expression and circuit from a truth table

1. Use the "sum-of-products" algorithm to convert the following truth table into a Boolean expression.

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

Boolean Expression:

2. Now put your circuit into the sandbox, verify that it works correctly, and print. (TA_____)

Part 4: Create a truth table, expression and circuit from a circuit description.

1. An odd parity bit is an extra bit, attached to the end of a string of bits, which is set to 1 or 0 in such a way that the entire string (including the parity bit) contains an odd number of 1's. For example, if the original string of bits is 01101010, then the parity bit show be 1, yielding the result 011010101 with an odd number of 1's. If the original string is 100110, the parity bit is 0 since the original string of bits already had an odd number of 1's.

Create a circuit that produces as its output the correct odd parity bit to accompany 3 one-bit input values. In other words, the circuits job is to decide what the parity should be for a 3 bit input string. (ie. If the input bits are 111 then the parity bit will be 0).

Create a neat truth table, Boolean expression, and the working circuit in Circuit Sandbox. (Print and attach the circuit)

Odd Parity Truth Table:

Odd Parity Boolean Expression:

Have a TA initial that the circuit works correctly_____

Part 5: Create a truth table and circuit from a function

1. Given the following function create a truth table.
   M = A'BC+A ($M = \overline{A}BC + A$)

A	B	C	A'BC+A

2. From this truth table or function construct a circuit. Print out the circuit and attach it.
3. NAND gates can be arranged to build every other type of gate. Convert the circuit diagram from the previous step to a diagram containing only NAND gates. Try to use as *few gates* as possible. (Hint: look at the demonstrated law in part 2 - this law is the basis for transforming all circuits into a circuit of just NAND gates.)
4. Use Circuit Sandbox to create your circuit of just NAND gates and verify that it performs as expected. Print out this circuit and attach it as well. (TA_____)

Part 6: Simplification

Circuit#1

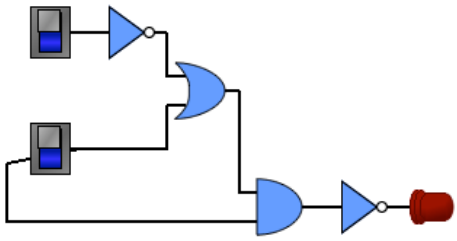
1. Given this function: **(QP(R + P))P' + (QR)' -** create its circuit and diagram below.

2. Now using the Boolean laws discussed in class and given in appendix A - simplify it as much as possible and give the new simplified function:

3. Now use Circuit Sandbox to demonstrate that these circuits are the same. (In other words create both of the circuits and see that they function the same.)

Circuit#2

1. Give the Boolean function for this circuit (label the input: P and Q).



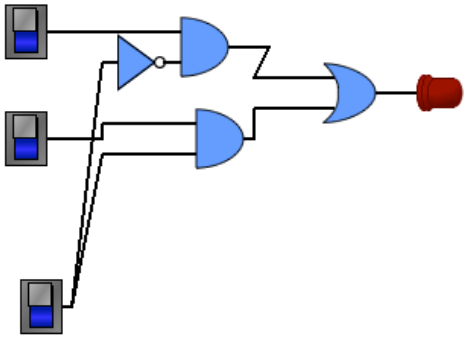
2. Now using the Boolean laws discussed in class and given in appendix A - simplify your function as much as possible:

3. Now use Circuit Sandbox to demonstrate that these circuits are the same. (In other words, create both of the circuits and see that they function the same.)

Part 7: Primary Circuits

2-to-1 Multiplexer

1. Create this circuit: (switches from top to bottom: A, B, C)



2. Create a truth table that represents this circuit. (be neat in creating your table)

3. Run the circuit and set the switches to different states and observe how it changes the output.

2-to-4 Decoder

1. Using Circuit Sandbox, create the circuit that matches the truth table below.

A	B	Out1	Out2	Out3	Out4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

2. Run the circuit and set the switches to different states and observe how it changes the output.

3. What is a possible use of this device?

Appendix A: Boolean Equivalence Laws

Circuit Equivalence - each law has 2 forms that are duals of each other.

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$