CDA 3201L Thursday Section
Lab number 02 -- Combinational Logic Circuits II
05/31/14 , John Gangemi & Kyle Peck

PURPOSE AND OBJECTIVES

Lab number 2 furthers our understanding combinational logic and design by introducing another universal gate and introducing a new concept in logic design, that is to build a circuit given a list of specifications.

The first part of lab number 2 demonstrates the "universal" feature of the NOR gate by implementing a given Boolean expression only using NOR gates.

The second part of lab number 2 requires that a logic circuit be built to satisfy the given specifications for a home alarm system, implying that the inputs and outputs of the system must be identified by the group accordingly.

COMPONENTS USED

- Integrated Circuits
    - . 74LS04 HEX Inverter
    - . 74LS11 3-Input AND gate
    - . 74LS08 2-Input AND gate
    - . 74LS32 2-Input OR gate
- 4 x 5mm Red LEDs
- 4 x 470 ohm Resistors (provides the right amount of dimming)
- 5 Volt Regulated DC Power Supply
- Assortment of 22 AWG Jumper Wires
- 2250 point Breadboard
- Logisim Program version 2.7.1

DESIGN DESCRIPTION

Part A

To implement the expression, Z = XY' + X'Y, using a maximum of five NOR gates, the expression must first be simplified using Boolean algebra and laws.

Z = XY' + X'Y
Z = XY' + X'Y + XX' + YY' *(Inverse Law)*
Z = X(X' + Y') + Y(X' + Y') *(Distributive Law)*
let A = X' + Y'
Z = X(A) + Y(A)

Z = A(X + Y) *(Distributive Law)*
let B = X + Y
Z = AB
Z = (A' + B')' *(De Morgan's Law)*
Z = ((X' + Y')' + (X + Y)')'

In order to construct a circuit with only NOR gates for the expression, Z = ((X' + Y')' + (X + Y)')', it is necessary to understand the Boolean expression for a NOR gate and the relevant Boolean logic for the NOT gate using the universal NOR gate.

NOR Truth Table:

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

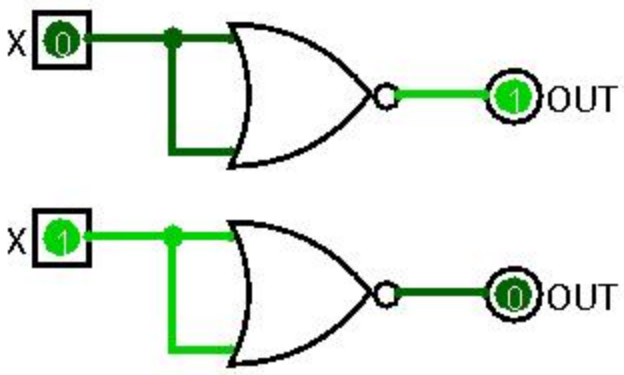
Thus the Boolean expression for a NOR gate is (X + Y)'

NOT using NOR:

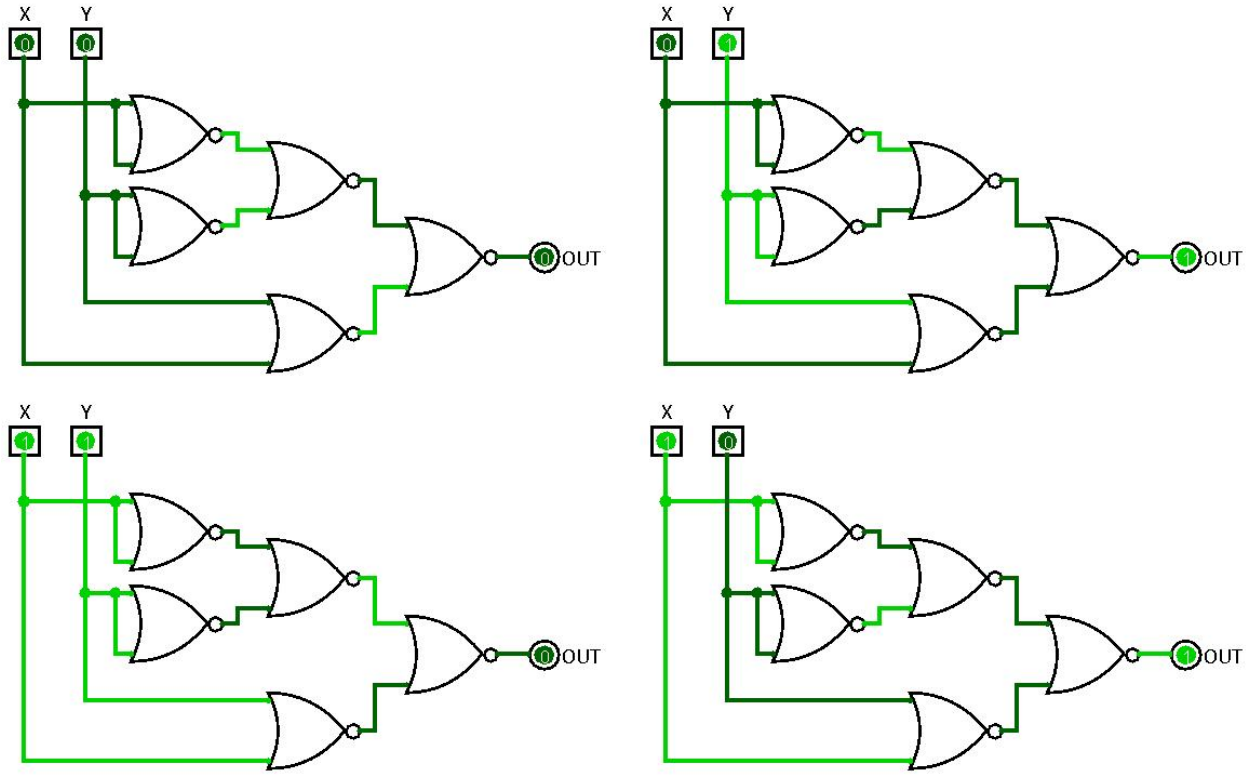
Show that X' = (X + Y)', however since the NOT gate only accepts one input let X and Y for the NOR gate be equivalent inputs.

X' = (X + X)'
X' = (X)' *(Idempotent Law)*

Therefore the NOT gate can be modeled as follows…



Examining the expression, Z = ((X' + Y')' + (X + Y)')', using the knowledge of a NOR gate's Boolean expression and the circuit for a NOT gate using a NOR gate it is possible to derive a logic diagram of the following…



Checking the Inputs/Outputs with a truth table of the original expression shows the correct logic circuit has been designed.

Logic Circuit Truth Table:

X	Y	Z = XY' + X'Y
0	0	0
0	1	1
1	0	1
1	1	0

Part B

In planning the design of a logic circuit that behaves like a home alarm system it was required that the following features be implemented…

- If the front door is open, the FRONT light should illuminate.
- If the back door is open, the BACK light should illuminate.
- If both the front and back doors are open, the ALL_OPEN light should illuminate and the FRONT / BACK lights should turn off.
- No lights will be illuminated unless the alarm system is turned on.
- If the alarm is triggered, the ALARM light and FRONT / BACK / ALL_OPEN lights should illuminate.

Accordingly the number of inputs and outputs was decided based on the specifications provided. The inputs for the logic circuit are Front Door (A), Back Door (B), Trigger (C), and System (D). The outputs for the logic circuit are Front Light (x), Back Light (y), All Open light (w), and Alarm Light (z).

To derive the desired emission of light from each of the individual LED's connected to their respective output a truth table was created with the appropriate designation of "1s" in each column of the outputs.

Home Alarm Truth Table:

A	B	C	D	x	y	w	z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	1	1	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	1	1	1

1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1

Resulting equations for each the outputs are…

x = A'B'CD + A'BCD + AB'C'D + AB'CD + ABCD
y = A'B'CD + A'BC'D + A'BCD + AB'CD + ABCD
w = A'B'CD + A'BCD + AB'CD + ABC'D + ABCD
z = A'B'CD + A'BCD + AB'CD + ABCD

Applying a Karnaugh map to each equation for the individual outputs shown above will give their simplest form.

Karnaugh Map for X:

AB	**00**	**01**	**11**	**10**
CD				
00	0	0	0	0
01	0	0	0	1
11	1	1	1	1
10	0	0	0	0

x = AB'D + CD

Karnaugh Map for Y:

AB	**00**	**01**	**11**	**10**
CD				
00	0	0	0	0
01	0	1	0	0

11	1	1	1	1
10	0	0	0	0

$y = A'BD + CD$

Karnaugh Map for W:

AB	**00**	**01**	**11**	**10**
CD				
00	0	0	0	0
01	0	0	1	0
11	1	1	1	1
10	0	0	0	0

$w = ABD + CD$

Karnaugh Map for Z:

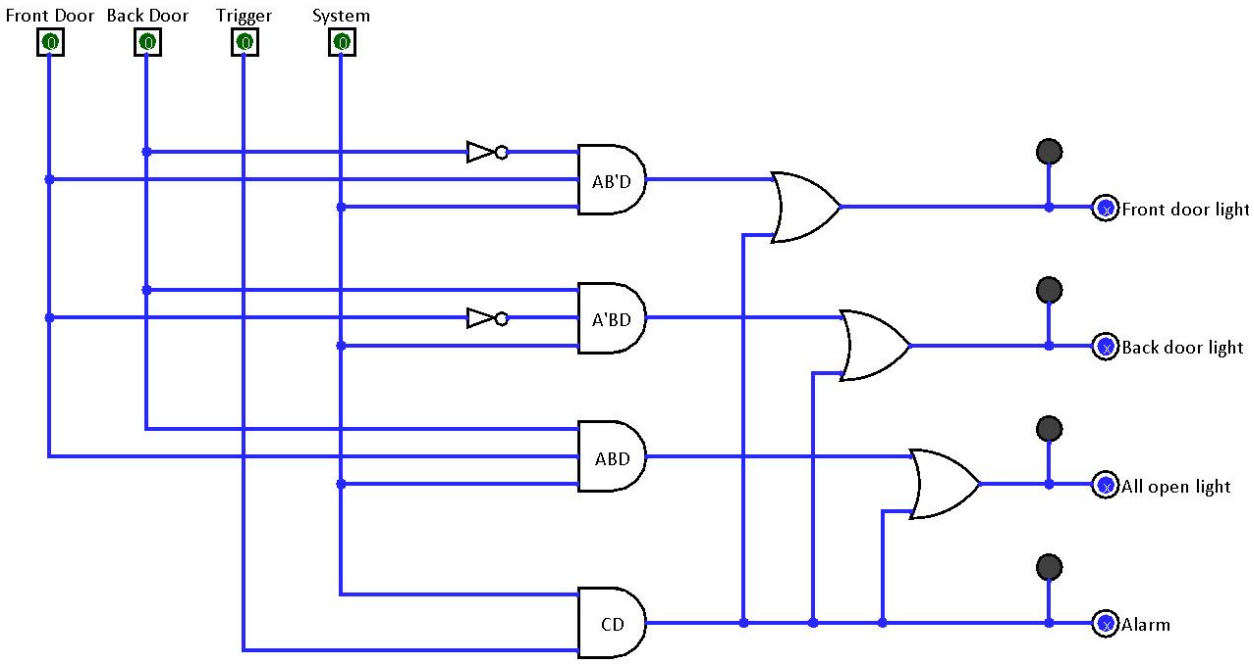
AB	**00**	**01**	**11**	**10**
CD				
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

$z = CD$

OBSERVATIONS AND DATA ANALYSIS

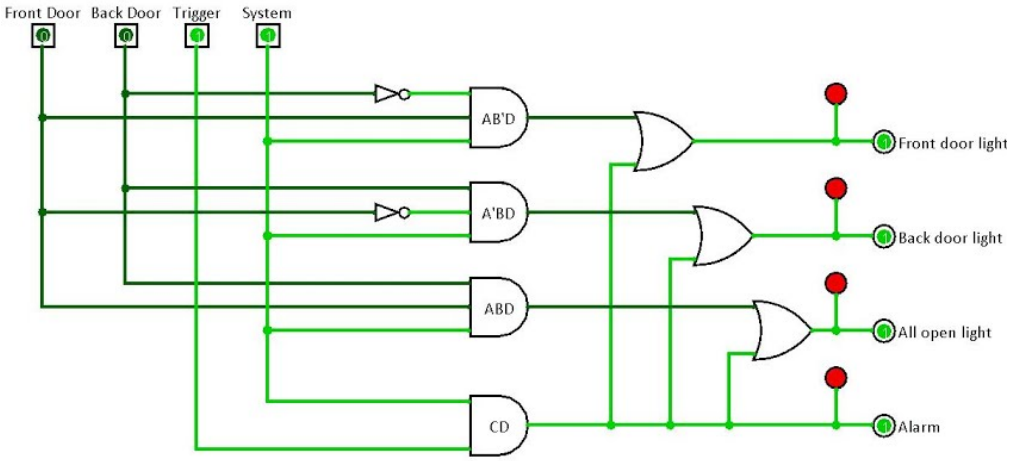
Part B

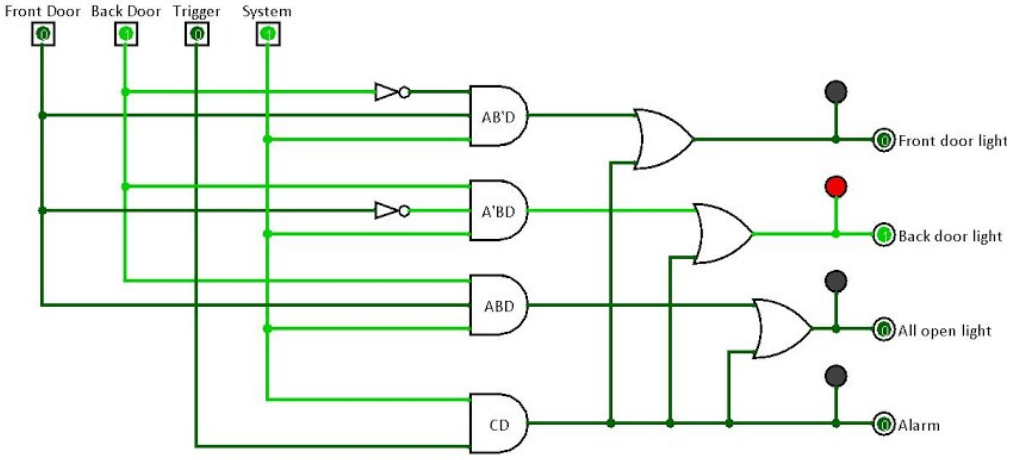
Analyzing the final simplified Boolean expression for each output corresponding to the logic circuit implies that the logic diagram should be constructed as follows...

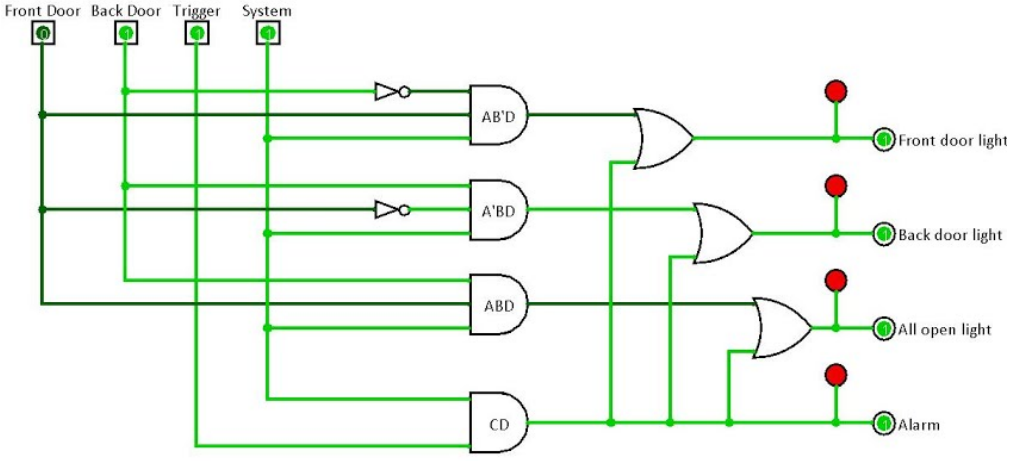


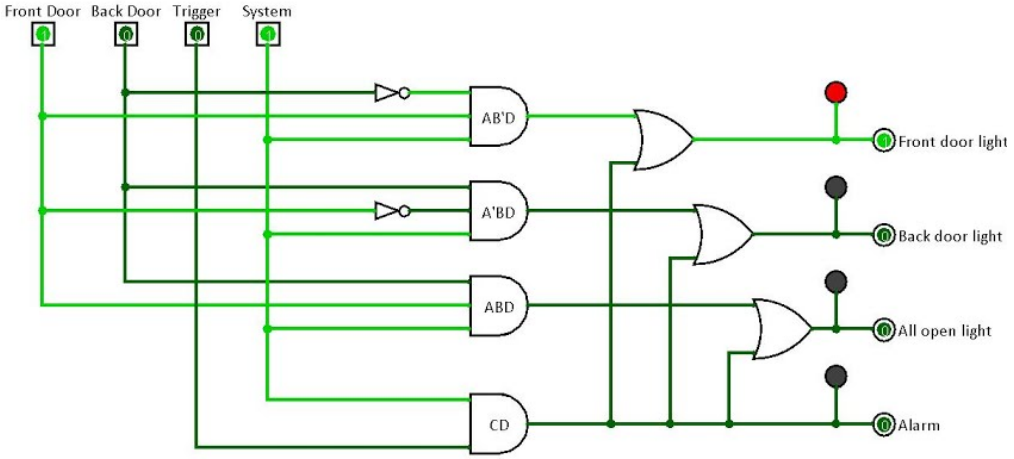
It can be observed that the ANDing of the Trigger (C) and the System (D) is common to all Boolean expressions for the outputs and thus activates all other outputs if the resulting logic from CD is true.

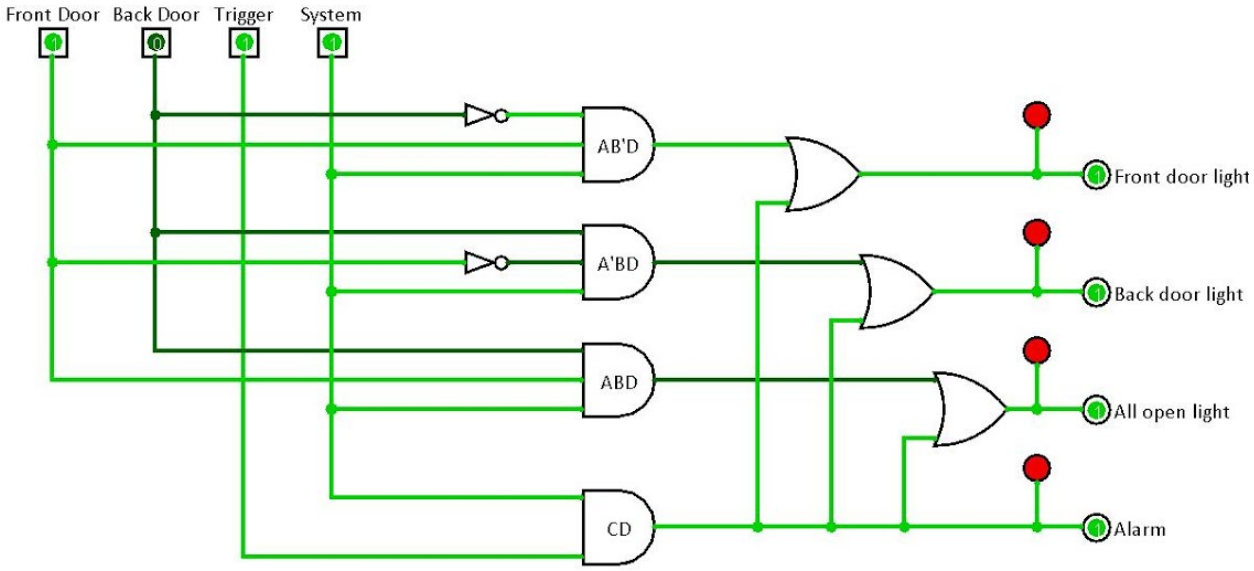
Instead of displaying all 16 possible configurations of the logic diagram, only the configurations that result in either or all outputs being “active” are shown below. Accordingly the Inputs to Outputs match the “Home Alarm Truth Table” given earlier.

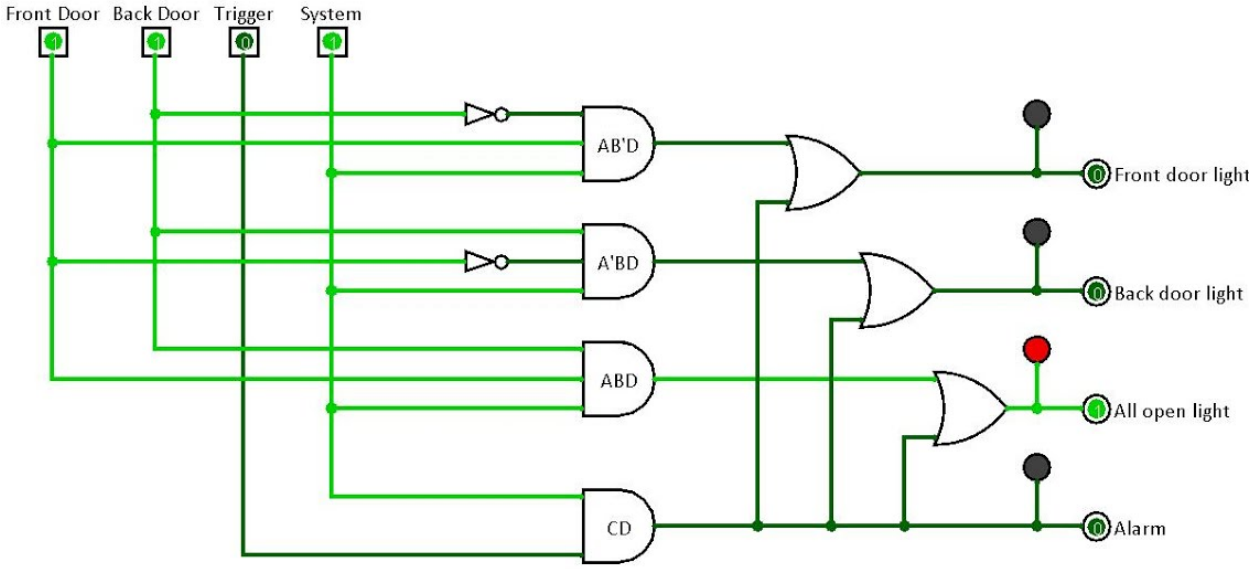

Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm

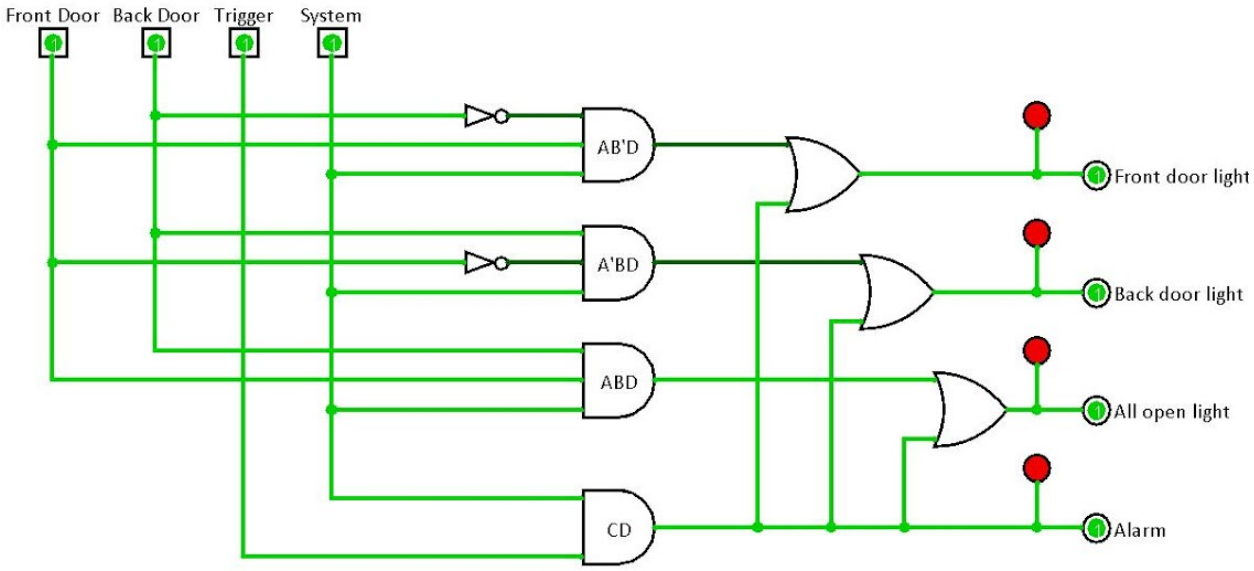

Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm


Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm


Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm


Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm


Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm


Front Door
Back Door
Trigger
System
AB'D
A'BD
ABD
CD
Front door light
Back door light
All open light
Alarm

DISCUSSION AND CONCLUSION

From the first section of the lab (part A) the group learned how a NOR gate can be used to model the functionality of a Boolean expression but more importantly how the NOR can be used to model the functionality of all other logic gates, thus making the NOR gate a universal gate.

The second section of the lab (part B) demonstrated how logic can be applied to word problems where certain specifications must be accomplished in order to receive the desired operation. This was the first experience with selecting inputs and outputs for a logic circuit and justifying a Boolean expression for each of the outputs. It was overall an intriguing lab, the most satisfaction was obtained from the design method of part B. Applying the basic rules and techniques learned from lecture the design was finished before the end of the lab, however the most frustrating process noted in lab has been the physical implementation of the circuits. It is very easy to misplace a wire and thus have an incorrect circuit requiring a complete rebuild for which this was the exact dilemma our group experienced.