ECE 281

Lesson 4 and 6 Notes

Objectives:

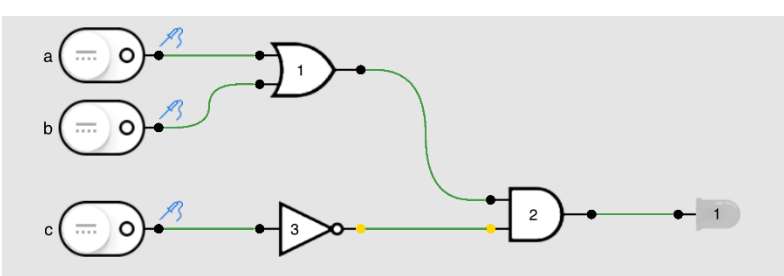
- Based on a problem description, design a solution using Boolean equations to solve the problem
- Utilize Boolean algebra to simplify Boolean equations or to derive a form that yields the most desirable hardware configuration
- Demonstrate the ability to describe a combinational digital system by a truth table, Boolean equation, Sigma/Pi notation, and schematic
- Given any one form, be able to produce the others

Review and Example #1: Lets create a logic circuit that will turn on an LED to see if you are eligible to apply for USAF Test Pilot School as a Flight Test Engineer. The key criteria are degree and time in service.

A – signals applicant has engineering degree
B – signals applicant has basic science degree
C – signals greater than 8 years time in service

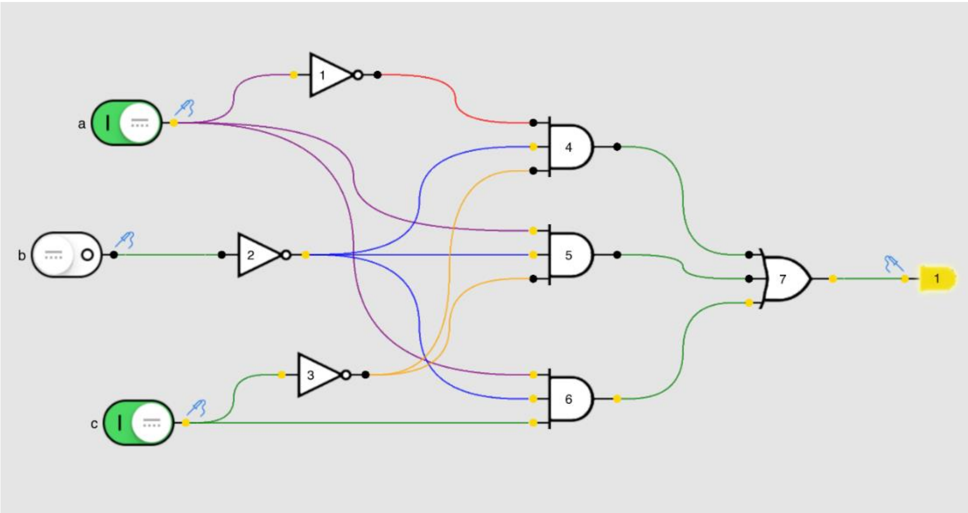
A	B	C	Y

$$Y = (A + B)\overline{C}$$



A	B	C	Y

Example #2) Y = A'B'C'+AB'C'+AB'C



Circuit Schematic Best Practices:

A. **Inputs on the left (or Top)**

B. **Outputs on the right (or bottom)**

C. **Gates flow from left to right**

D. **Straight wires are best**

Boolean Algebra Main Principles:

In Boolean Algebra, variables can take on either a true or false value. Using the rules we will discuss in this lesson, expressions can often be reduced to result in more efficient circuit implementation. While the second example above was already rather simple, we will prove an even simpler implementation through the course of today's lesson.

<u>Terminology</u>

Equivalent Representations: A and B $\Longleftrightarrow$ AB
A or B $\Longleftrightarrow$ A + B
Not A $\Longleftrightarrow$ A' $\Longleftrightarrow$ $\bar{A}$

Truth Table:

Minterm:

Maxterm:

Order of Operations:

Sum of Products:

Input A	Input B	Output Y	Minterm (Form 1)	Minterm (Form 2)
0	0		$\bar{A}\bar{B}$	A'B'
0	1		$\bar{A}B$	A'B
1	0		$A\bar{B}$	AB'
1	1		AB	AB

Product of Sums:

Input A	Input B	Output Y	Maxterm (Form 1)	Maxterm (Form 2)
0	0		$A+B$	A+B
0	1		$A+\bar{B}$	A+B'
1	0		$\bar{A}+B$	A'+B
1	1		$\bar{A}+\bar{B}$	A'+B'

NOTE: $\bar{A}\bar{B}$ is not the same as $\overline{AB}$ … more on this shortly

Commonly Used Boolean Algebra Properties (Review from zyBooks)

Property	Name	Description
A(B+C) = AB+AC A+(BC) = (A+B)(A+C)	Distributive (AND) Distributive (OR)	(AND) Same as multiplication in regular algebra (OR) Not at all like regular algebra
AB=BA A+B=B+A	Commutative	Variable order does not matter. Good practice is to sort variables alphabetically
(AB)C=A(BC) (A+B)+C = A+(B+C)	Associative	Same as regular algebra
AA' = 0 A+A' = 1	Complement (AND) Complement (OR)	(AND) Clearly one of A, A' must be 0. 1·0 = 0·1=0 (OR) Clearly one of A,A' must be 1. 1+0 = 0+1 =**1**
A·1=A A+0=A	Identity (AND) Identity (OR)	(AND) Result of A·1 is always A's value **0**·1=**0** **1**·1=**1** (OR) Result of A+0 is always A's value **0**+0=**0** **1**+0=**1**
A·0=0 A+1=1	Null elements	Result doesn't depend on the value of A
A·A=A A+A=A	Idempotent	Duplicate values can be removed
(A')' = A	Involution	(0')'=(1)'=0 (1')'=(0)'=1
A(A+B) = A A+(AB) = A	Covering	In either case, the value of B is irrelevent
(AB)+(AB') = A (A+B)(A+B')= A	Combining	This is an application of the complement and identity properties listed above
AB+B'C= AC (A+B)(B'+C)=(A+C)	Consensus	The conjunction of all unique terms
(AB)'=A'+B'	DeMorgan's Law (for AND)	Each literal complemented ANDs become ORs
(A+B)' = A'B'	DeMorgan's Law (for OR)	Each literal complemented, ORs become ANDs

Visualization of Demorgan's Theorem:

Y=(AB)' =

Y=(A+B)'

Simplification of Example #2 Using Boolean Algebra:

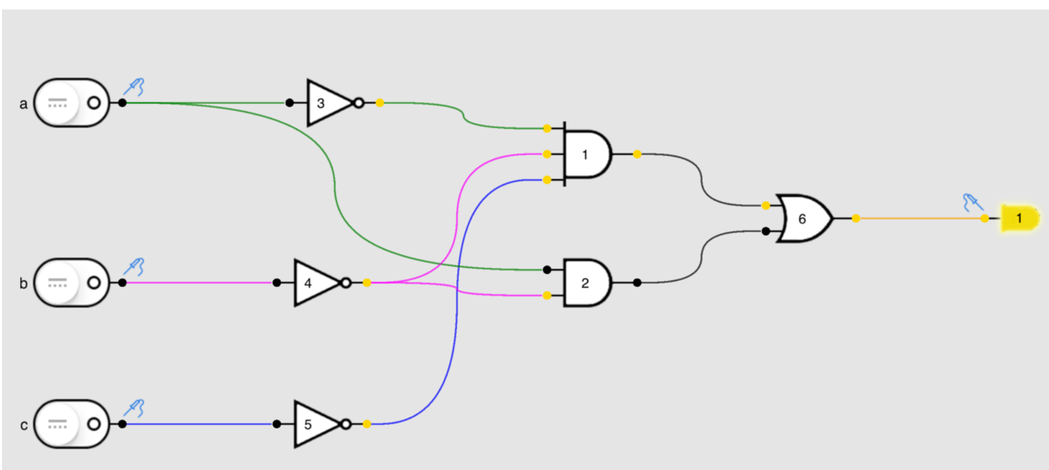
A	B	C	Y

$$Y = A'B'C' + AB'C' + AB'C$$

$$Y = A'B'C' + AB'(C + C')$$

$$Y = A'B'C' + AB'(1)$$

$$Y = A'B'C' + AB'$$



Revisiting the concept that $\overline{A}\overline{B}$ is not the same as $\overline{AB}$

$\overline{A}\overline{B}$

A	B	Y

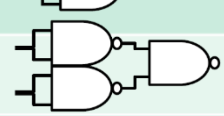
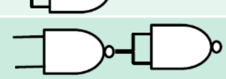
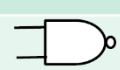
$\overline{AB}$

A	B	Y

Which logic gate represents $\overline{AB}$?

Through application of DeMorgan's law, which single logic gate represents $\overline{A}\overline{B}$?

And for the final topic today, any logic operation can be represented solely with NAND gates.

	Standard	NAND	Schematic
NOT	$\bar{A}$	$\overline{AA}$	
OR	$A + B$	$\overline{\overline{A}\,\overline{B}}$	
AND	AB	$\overline{\overline{AB}}$	
NAND	$\overline{AB}$	$\overline{AB}$	
NOR	$\overline{A + B}$	$\overline{\overline{\overline{A}\,\overline{B}}}$