EGEC 180 – Digital Logic and Computer Structures

Spring 2024

Lecture 3: Logic Gates (1.5)

Rakesh Mahto, Ph.D.
Office: *E 314, California State University, Fullerton*
Office Hour: *Monday and Wednesday 2:00 - 3:30 pm*
Or by appointment
Office Hour Zoom Meeting ID: *891 2907 5346*
Email: *ramahto@fullerton.edu*
Phone No: *657-278-7274*

Logical Gates

Basic Operations

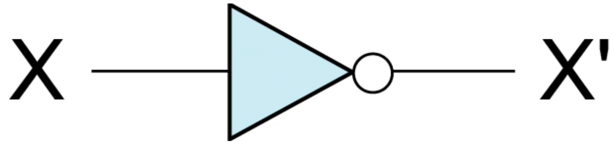
- The basic operations of Boolean Algebra are AND, OR, and complement (also known as the Inverse or NOT operations)
- Operator precedence: Parentheses, NOT, AND, OR
- Inverse or NOT or Negation or Complement
  - So here, we formally state that the complement of “0” is “1”, complement of “1” is “0”.

Example: Ones Complement arithmetic we flipped the 1s and 0s.

- Symbol a tick mark (′) is often used to denote complementation.

Example: if X = 0 then X’ = 1

if X = 1 then X’ = 0



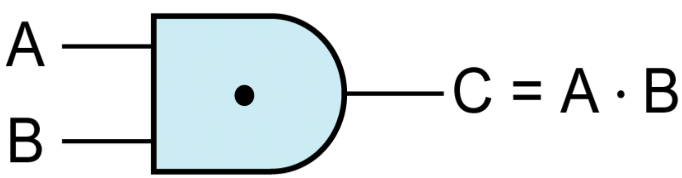
Basic Operations (AND)

- AND operation -> Z=1 iff X=1 and Y=1

  if X=1 AND Y=1 then Z=1 else Z=0;

  When two or more variables are ANDed together, the value of the result will be 1 if and only if (iff) all the variables have the value 1. If any of the variables have the value 0, the result of the AND operation will be 0.

- Representation: XY=Z, X•Y=Z, X.Y=Z



Truth Table

A B	C = A • B
0 0	0
0 1	0
1 0	0
1 1	1

Input **Output**

Basic Operations (OR)

- OR operation -> Z = 0, iff X=0 and Y=0

  if X=1 OR Y=1 then Z=1 else Z=0;

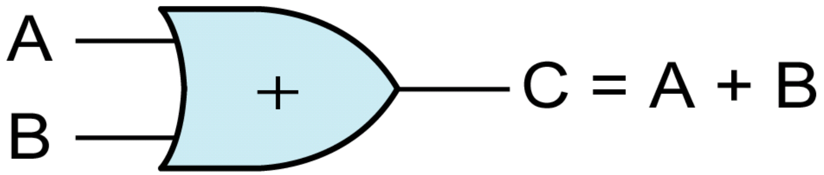
  When two or more variables are ORed together, the value of the result will be 1 if and only if (iff) one or both of the variables have the value 1. The result of the OR operation will be a 0 iff all of the variables have the value 0.

X+Y=Z, X v Y=Z

Truth Table

A B	C = A + B
0 0	0
0 1	1
1 0	1
1 1	1
Input	**Output**

NOTE: Binary Logic (Boolean Algebra) ≠ Binary Arithmetic



Example Problem #1

- Lets evaluate XY+Z

X	Y	Z	XY	XY+Z
0	0	0	0.0 = 0	0+0 = 0
0	0	1	0.0 = 0	0+1 = 1
0	1	0	0.1 = 0	0+0 = 0
0	1	1	0.1 = 0	0+1 = 1
1	0	0	1.0 = 0	0+0 = 0
1	0	1	1.0 = 0	0+1 = 1
1	1	0	1.1 = 1	1+0 = 1
1	1	1	1.1 =1	1+1 = 1

Practice Problem #1

- Lets evaluate X+YZ'

X	Y	Z	Z'	YZ'	X+YZ'
0	0	0	1	0.1 = 0	0+0 = 0
0	0	1	0	0.0 = 0	0+0 = 0
0	1	0	1	1.1 = 1	0+1 = 1
0	1	1	0	1.0 = 0	0+0 = 0
1	0	0	1	0.1 = 0	1+0 = 1
1	0	1	0	0.0 = 0	1+0 = 1
1	1	0	1	1.1 = 1	1+1 = 1
1	1	1	0	1.0 = 0	1+0 = 1

Practice Problem #2

- Is XY+Z = X+YZ'

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

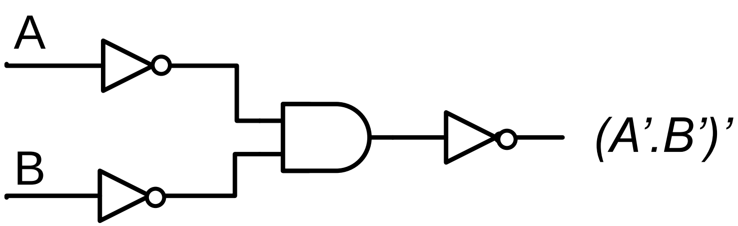
So the answer is NO

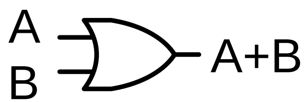
Example Problem #2

- Is A+B = (A’.B’)’?
- Step 1 Complete the truth table for A+B
- St Ta B’)’

A	B	A’	B’	A’.B’	(A’.B’)’
0	0	1	1	1.1 = 1	0
0	1	1	0	1.0 = 0	1
1	0	0	1	0.1 = 0	1
1	1	0	0	0.0 = 0	1

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



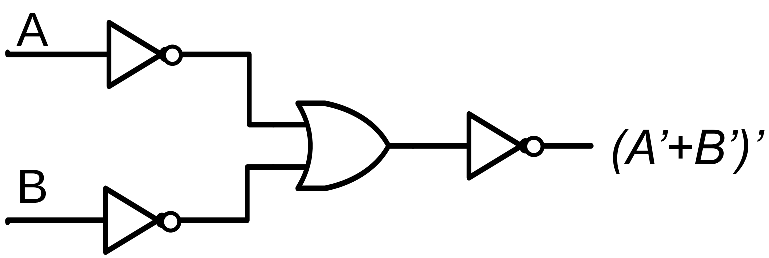


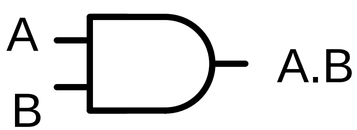
Practice Problem # 3

- Is AB = (A'+B')'

A	B	A'	B'	A'+B'	(A'+B')'
0	0	1	1	1+1 = 1	0
0	1	1	0	1+0 = 1	0
1	0	0	1	0+1 = 1	0
1	1	0	0	0+0 = 0	1

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1





Practice Problem # 4

- What is the output for the function (X+X').Y

X	Y	X'	X+X'	(X+X').Y
0	0	1	0+1 = 1	1.0=0
0	1	1	0+1 = 1	1.1=1
1	0	0	1+0 = 1	1.0=0
1	1	0	1+0 = 1	1.1=1

Why implement the circuit It does not change Y?

TABLE 1.1 Major binary operators with variables (or signals) in a Boolean expression

Boolean expression	Type of operation	VHDL equivalent
$\overline{X}$	**Complement** of X	**NOT** X
$X \cdot Y$	**Intersection** of X, Y	X **AND** Y
$X + Y$	**Union** of X, Y	X **OR** Y

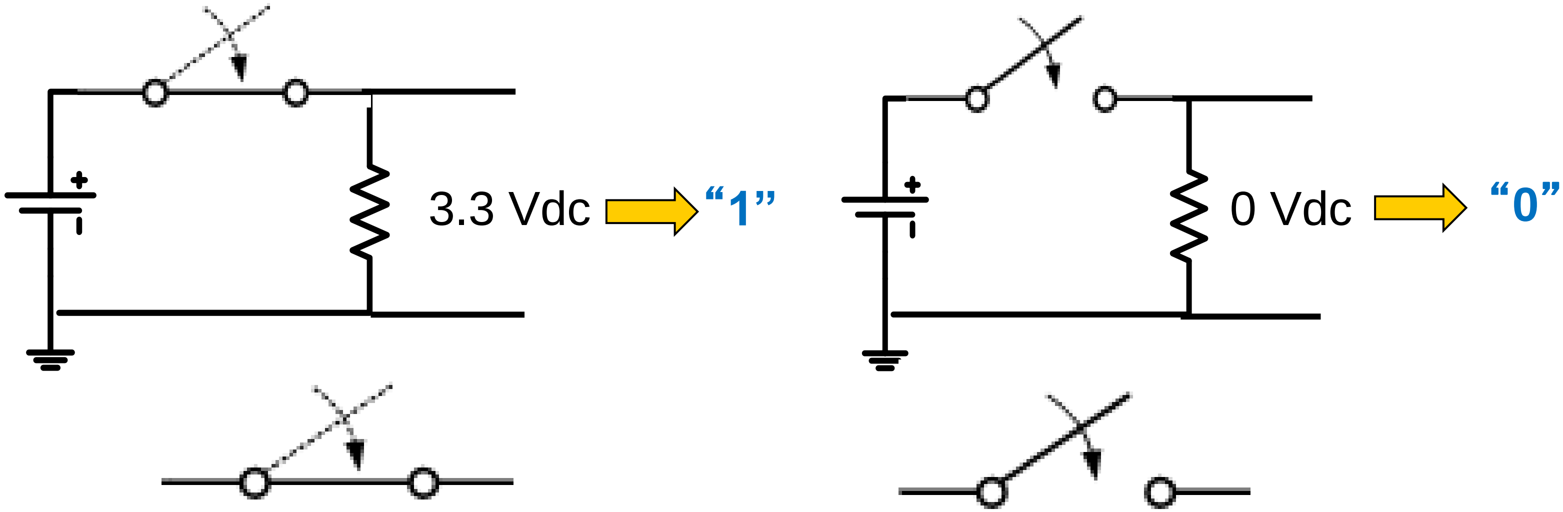
Operator precedence: parentheses, NOT, AND, OR

Switching Algebra (Switches)

If switch *X* is open, then we will define the value of *X* to be 0; if switch *X* is closed, then we will define the value of *X* to be 1.

Used with Factory floor automation systems and VHDL Labs.

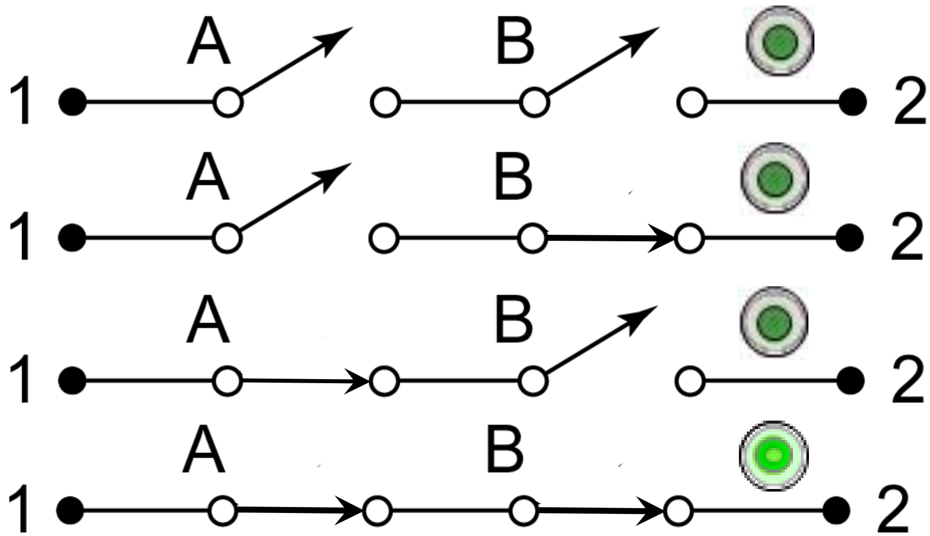
Seventh way of playing with Boolean Functions!!!!



Boolean Algebra or Switching Algebra

Switching Algebra (AND)

$T = AB$



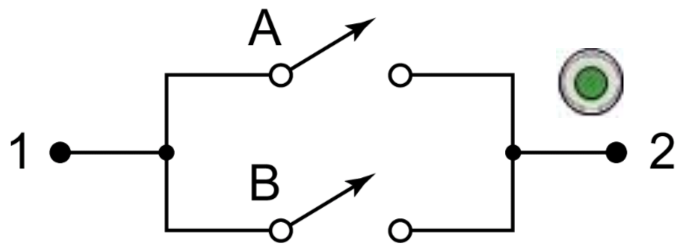
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

$T = 0$ open circuit between terminals 1 and 2

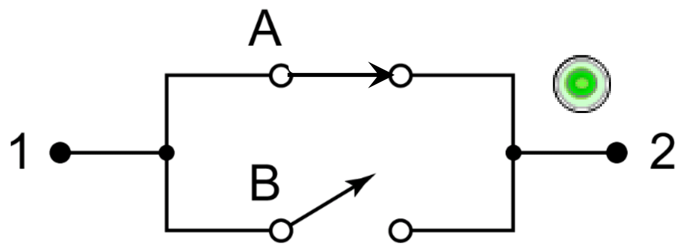
$T = 1$ closed circuit between terminals 1 and 2

Switching Algebra (OR)

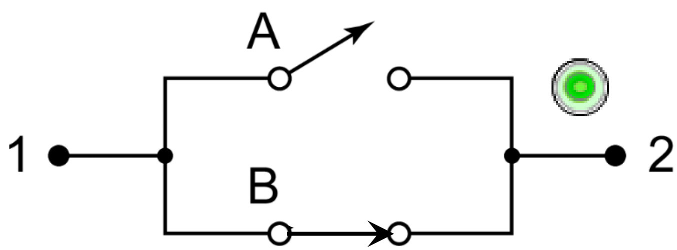
Note: Parallel Structure T = A+B



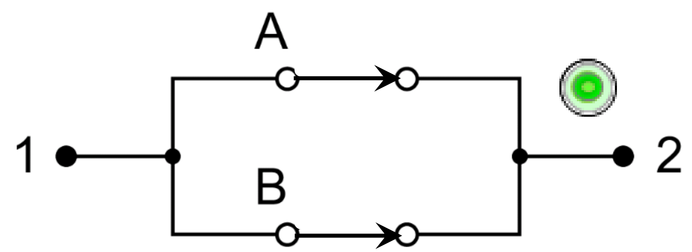
A	B	A+B
0	0	0



A	B	A+B
1	0	1



A	B	A+B
0	1	1

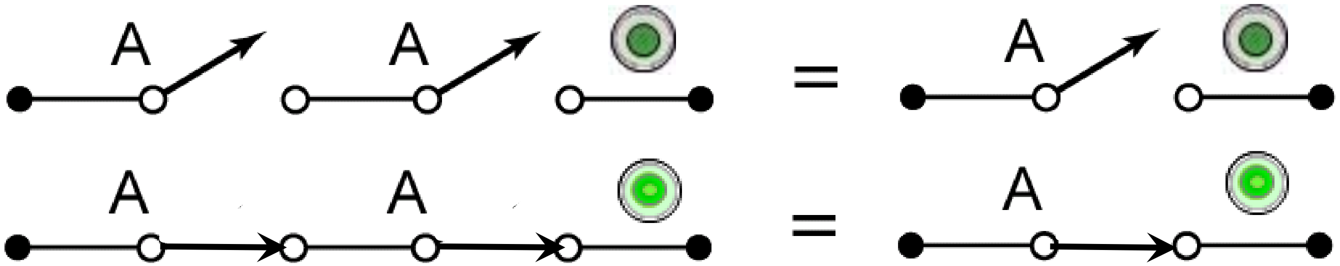


A	B	A+B
1	1	1

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

More on Switching Algebra

If two switches are both labeled with the variable A, this means that both switches are open when A = 0 and both are closed when A = 1, thus the following circuits are equivalent:

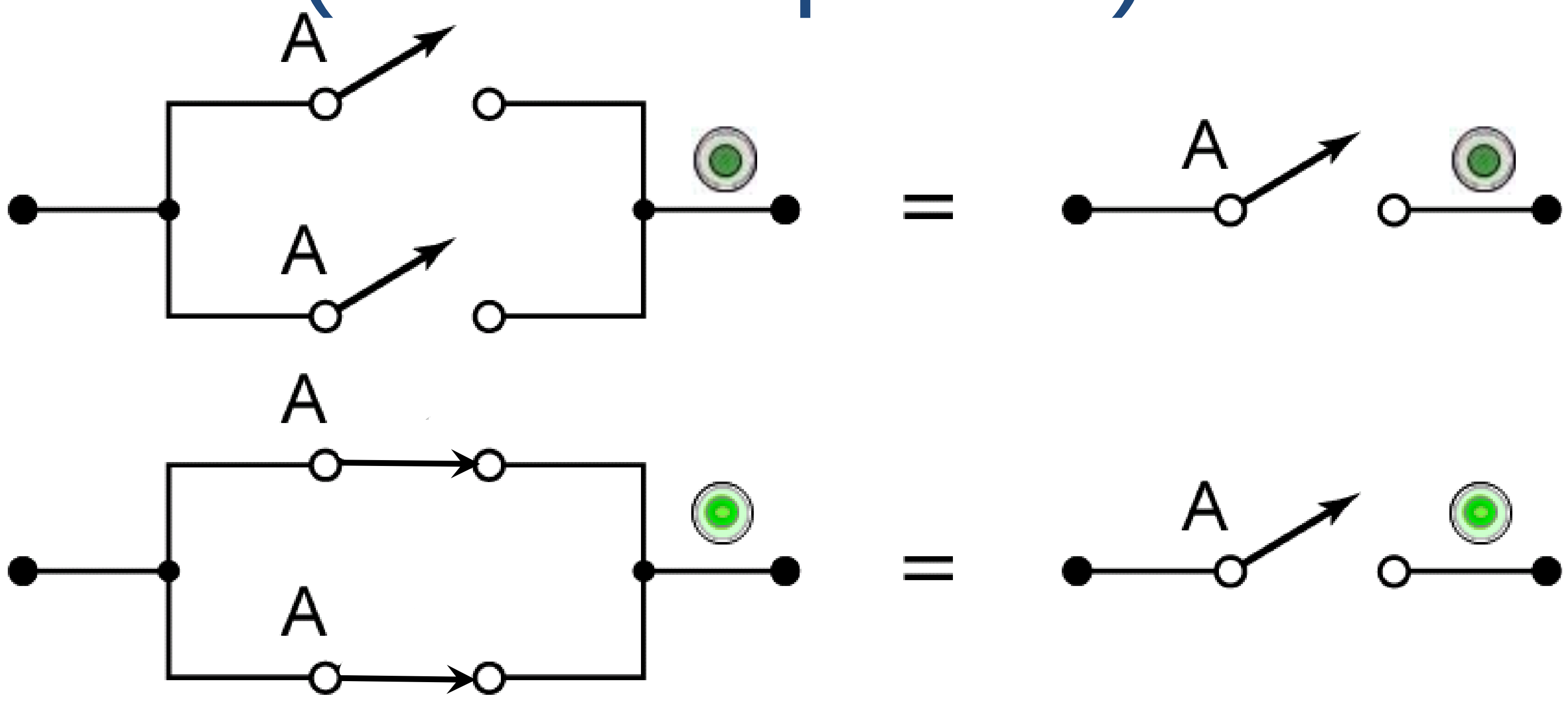


$(A \cdot A = A)$

A	A	A.A
0	0	0.0=0
1	1	1.1=1

$\}$ A

Switching Algebra (OR Idempotent)



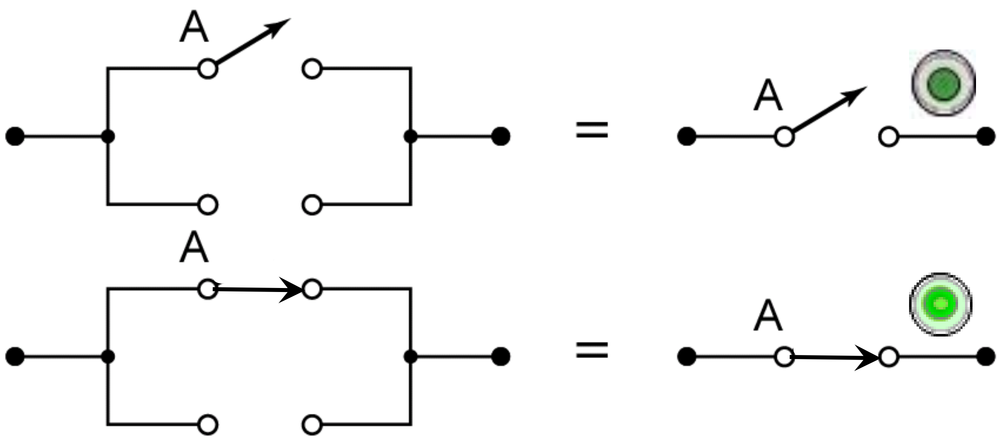
Note: By similar reasoning
A.A = A

A	A	A+A
0	0	0+0=0
1	1	1+1=1

A

Switching Algebra (Identities)

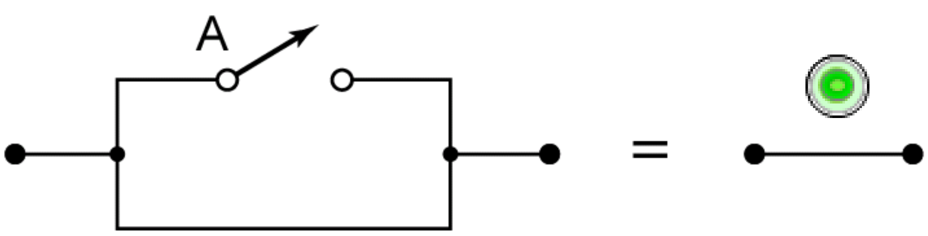
$(A + 0 = A)$



A	0	A+0
0	0	0+0 =0
1	0	1+0 = 1

A

$(A + 1 = 1)$



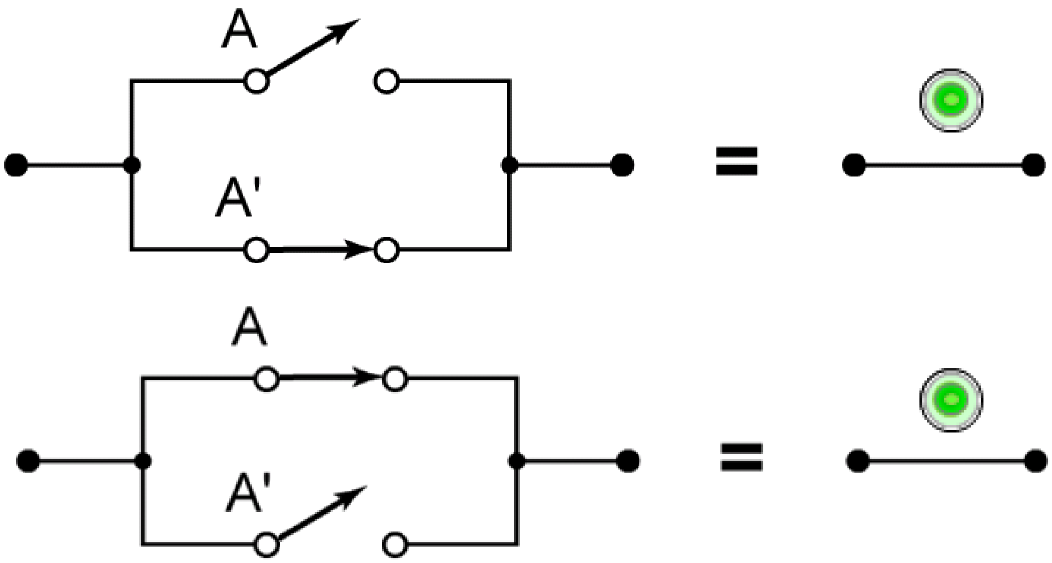
A	1	A+1
0	1	0+1 =1
1	1	1+1 = 1

A

Switching Algebra (Tautology)

A in parallel with A’ can be replaced with a closed circuit because one or the other of the two switches is always closed.

$$(A + A' = 1)$$

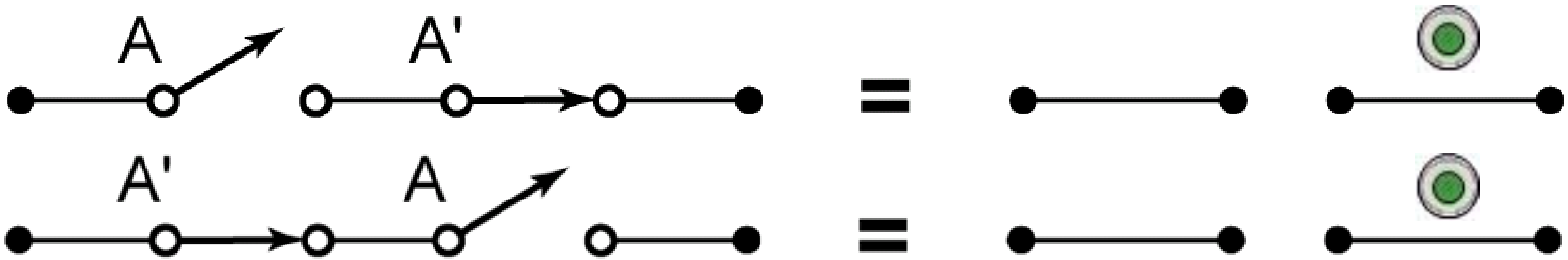


A	A’	A+A’
0	1	0+1=1
1	0	1+0=1

1

Switching Algebra (Fallacy)

Similarly, switch A in series with A’ can be replaced with an open circuit because one or the other of the two switches is always open.



$$(A \cdot A' = 0)$$

A	A’	A.A’
0	1	0.1=0
1	0	1.0=0

0

Boolean Algebra Theorems

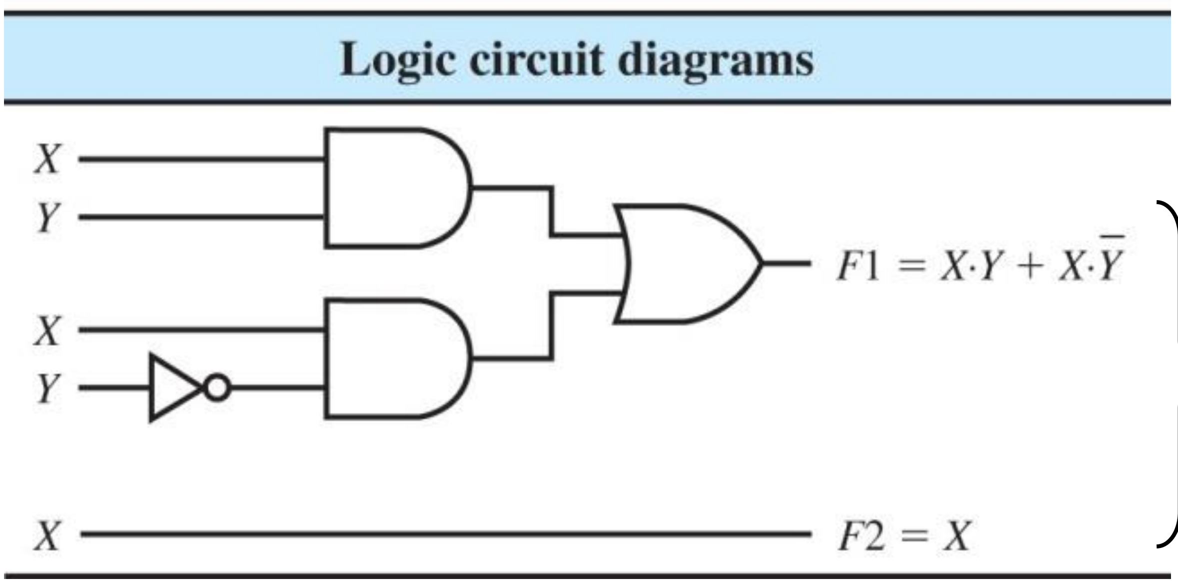
Absorption Theorem	T1a:	$X \cdot (X + Y) = X$
	T1b:	$X + X \cdot Y = X$
Adjacency Theorem	T2a:	$X \cdot Y + X \cdot \overline{Y} = X$
	T2b:	$(X + Y) \cdot (X + \overline{Y}) = X$
Associative Theorem	T3a:	$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$
	T3b:	$X + (Y + Z) = (X + Y) + Z$
Consensus Theorem	T4a:	$X \cdot Y + \overline{X} \cdot Z + Y \cdot Z = X \cdot Y + \overline{X} \cdot Z$
	T4b:	$(X + Y) \cdot (\overline{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\overline{X} + Z)$

Double Complementation or Double Negation Theorem	T7:	$\overline{\overline{X}} = X$
Idempotency Theorem	T8a:	$X \cdot X = X$
	T8b:	$X + X = X$
Identity Element Theorem	T9a:	$X \cdot 0 = 0$
	T9b:	$X + 1 = 1$
Simplification Theorem	T10a:	$X \cdot (\overline{X} + Y) = X \cdot Y$
	T10b:	$X + \overline{X} \cdot Y = X + Y$

DeMorgan's Theorem (with two variables)	T5a:	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$
	T5b:	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$
DeMorgan's Theorem (with multiple variables)	T6a:	$\overline{X \cdot Y \cdot Z \ldots} = \overline{X} + \overline{Y} + \overline{Z} + \ldots$
	T6b:	$\overline{X + Y + Z + \ldots} = \overline{X} \cdot \overline{Y} \cdot \overline{Z} \cdot \ldots$

Comparing circuit complexity for the Boolean functions F1 and F2

In general, it is better to use fewer logic gates to minimize complexity, cost, and power requirements.



Application of T2a

Adjacency Theorem

$$T2a: \quad X \cdot Y + X \cdot \overline{Y} = X$$

$$T2b: \quad (X + Y) \cdot (X + \overline{Y}) = X$$

Proving Boolean Algebra Theorems

Simplification Theorem T10b: $X + \overline{X}\cdot Y = X + Y$

Proof by perfect induction method

Step 1: Make the truth table

X	Y	$\overline{X}$	$\overline{X}\cdot Y$	$X + \overline{X}\cdot Y$	$X + Y$
0	0				
0	1				
1	0				
1	1				

Step 2: Fill in each column in the table using the operator definitions

X	Y	$\overline{X}$	$\overline{X}\cdot Y$	$X + \overline{X}\cdot Y$	$X + Y$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

EOL = EOR

Step 3: Verify that EOL = EOR

Proof by mathematical manipulation method (proving that EOR = EOL)

Substitute postulates and/or theorems to show that EOR = EOL, which can be tricky because it involves trial and error

	Using the following postulates and/or theorems:	
$X + Y$		
$X + Y\cdot 1$	Variable dominate rule	P1a: $X\cdot 1 = X$
$X + Y\cdot(X + \overline{X})$	Complement rule	P4b: $X + \overline{X} = 1$
$X + Y\cdot X + Y\cdot\overline{X}$	Distributive rule	P3a: $X\cdot(Y + Z) = X\cdot Y + X\cdot Z$
$X\cdot 1 + Y\cdot X + Y\cdot\overline{X}$	Variable dominate rule	P1a: $X\cdot 1 = X$
$X\cdot 1 + X\cdot Y + \overline{X}\cdot Y$	Commutative rule	P2a: $X\cdot Y = Y\cdot X$
$X\cdot(1 + Y) + \overline{X}\cdot Y$	Distributive rule	P3a: $X\cdot(Y + Z) = X\cdot Y + X\cdot Z$
$X\cdot 1 + \overline{X}\cdot Y$	Identity element theorem	T9b: $X + 1 = 1$
$X + \overline{X}\cdot Y$	Variable dominate rule	P1a: $X\cdot 1 = X$

Note that EOR = EOL

Proof of Adjacency Theorem

Adjacency Theorem T2a: $X \cdot Y + X \cdot \overline{Y} = X$

Proof by perfect induction method

X	Y	$\overline{Y}$	$X \cdot Y$	$X \cdot \overline{Y}$	$X \cdot Y + X \cdot \overline{Y}$
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
EOR			=		EOL

Proof by mathematical manipulation method (proving that EOL = EOR)

$X \cdot Y + X \cdot \overline{Y}$	Using the following postulates and/or theorems:	
$X \cdot (Y + \overline{Y})$	Distributive rule	P3a: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
$X \cdot 1$	Complement rule	P4b: $X + \overline{X} = 1$
X	Variable dominate rule	P1a: $X \cdot 1 = X$
Note that EOL = EOR		

Boolean Functions and Truth Tables

Truth Table

We can use a Truth Table to specify the output values of a circuit of logic gates in terms of the values of the input variables.

If an expression has n variables and each variable can have the value 0 or 1, the number of different combinations of values of the variables is 2^n.

Thus, a Truth Table for an n-variable expression will have n columns and 2^n rows.

Two expressions are equal if they have the same value for every possible combination of the variables.

Truth Table for 3 variables

Since the expression (A + C)(B' + C) has the same value as AB' + C for all eight combinations of values of the variables A, B, and C, we conclude that:

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

A B C	B'	AB'	AB'+C		A+C	B'+C	(A+C)(B'+C)
0 0 0	1	0.1=0	0+0=0		0+0=0	1+0=1	0.1=0
0 0 1	1	0.1=0	0+1=1		0+1=1	1+1=1	1.1=1
0 1 0	0	0.0=0	0+0=0		0+0=0	0+0=0	0.0=0
0 1 1	0	0.0=0	0+1=1		0+1=1	0+1=1	1.1=1
1 0 0	1	1.1=1	0+1=1		1+0=1	0+1=1	1.1=1
1 0 1	1	1.1=1	1+1=1		1+1=1	1+1=1	1.1=1
1 1 0	0	1.0=0	0+0=0		1+0=1	0+0=0	1.0=0
1 1 1	0	1.0=0	1+0=1		1+1=1	0+1=1	1.1=1

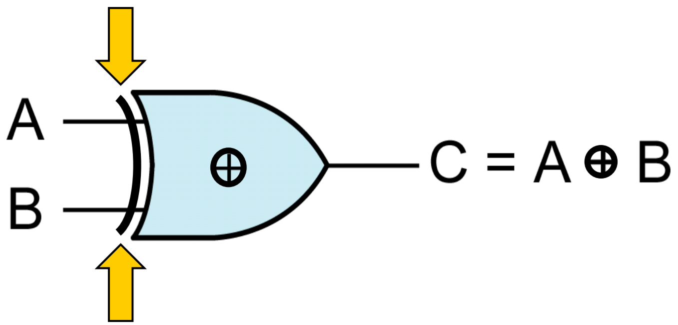
Other Operations (exclusive OR)

- **Exclusive OR (XOR) operation -> Z = 1, iff $X \neq Y$**

  if X != Y then Z=1 else Z=0;

  When two or more variables are XORed together, the value of the result will be 1 if and only if (iff) <u>they are not the same</u> of the. The result of the XOR operation will be a 0 iff both have the same value.

X xor Y=Z, $X \oplus Y = Z$, $XY' + X'Y = Z$, $X\bar{Y} + \bar{X}Y = Z$



Note: the Extra line

Truth Table

Boolean	X	Y	X⊕Y
X' Y'	0	0	0
X' Y	0	1	1
X Y'	1	0	1
X Y	1	1	0

Input **Output**

Truth Function

$Z = X.Y' + X'.Y$

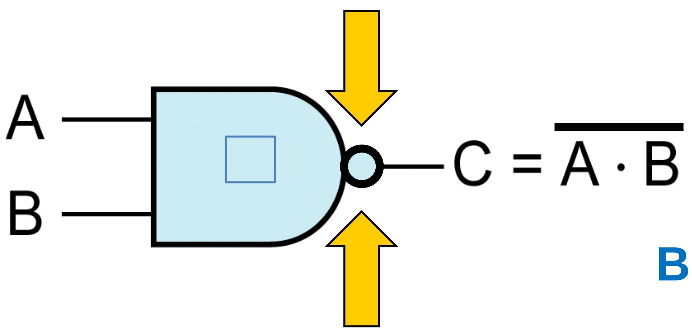
Basic Operations (Not AND)

- Not AND (NAND) operation -> Z=0 iff X=1 and Y=1

  if X=0 AND Y=0 then Z=1 else Z=0;

  When two or more variables are NANDed together, the value of the result will be 1 if and only if (iff) all the variables have the value *0*. If any of the variables have the value 1, the result of the NAND operation will be 0.

- Representation: $\overline{XY}=Z$, $\overline{X\bullet Y}=Z$, $(X.Y)'=Z$



Note: the Negation or Inversion Bubble

Truth Table

Boolean	X	Y	$\overline{XY}$
X' Y'	0	0	1
X' Y	0	1	1
X Y'	1	0	1
X Y	1	1	0

Input **Output**

Truth Function

$$Z = X'Y' + X'Y + XY'$$

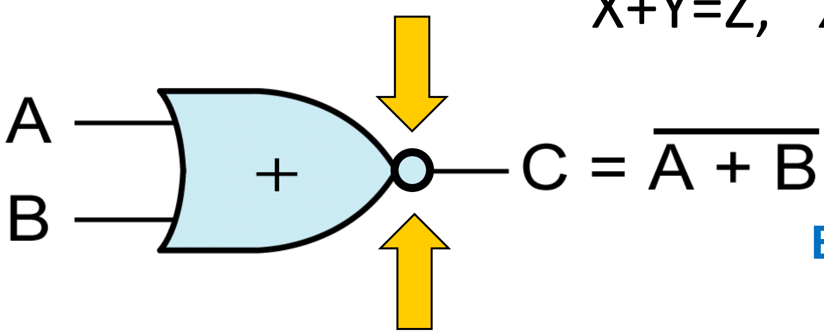
Basic Operations (Not OR)

- Not OR or NOR operation -> Z = 1, iff X=0 and Y=0

  if X=1 OR Y=1 then Z=0 else Z=1;

  When two or more variables are NORed together, the value of the result will be 1 if and only if (iff) both of the variables have the value *0*. The result of the NOR operation will be a 1 iff any of the variables have the value 1.

$\overline{X+Y}=Z$, $\overline{X \vee Y}=Z$



Note: the Negation or Inversion Bubble

Truth Table

Boolean	X	Y	$\overline{X+Y}$
X' Y'	0	0	1
X' Y	0	1	0
X Y'	1	0	0
X Y	1	1	0

Input **Output**

Truth Function

Z=X' Y'

Q&A