Boolean Logic & Logic Gates II

Lecture 5

Lecture 5

Boolean Logic & Logic Gates

BOOLEAN ALGEBRA

Boolean Algebra

- Logical Calculus of truth values
- Developed by George Boole in 1840s

- Algebra of only 2 values
  - 0 and 1
  - True and False

Boolean Algebra

- Resembles algebra of real number

- Instead of multiplication, addition and negation, it is replaced by logical operations of
  - conjunction $x \land y$ (**AND**),
  - disjunction $x \lor y$ (**OR**), and
  - negation $\neg x$ (**NOT**)

Algebra of Real Numbers

- Commutative Law of Addition and Commutative Law of Multiplication:
  - $A + B = B + A$
  - $AB = BA$

You should already know this from secondary school algebra.

- Associative Law of Addition:
  - $(A + B) + C = A + (B + C)$

- Distributive Law
  - $A(B + C) = AB + AC$
  - $(A + B)(C + D) = AC + AD + BC + BD$

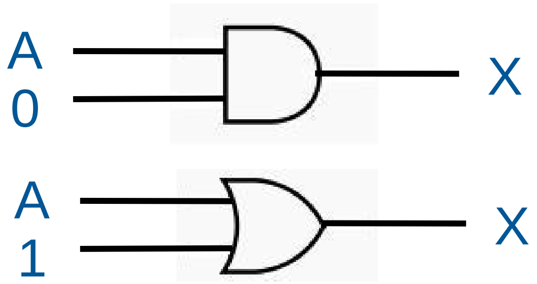
Boolean Algebra

- Commutative Law of Addition and Commutative Law of Multiplication:
  - $A \lor B = B \lor A$
  - $A \land B = B \land A$
- Associative Law of Addition:
  - $(A \lor B) \lor C = A \lor (B \lor C)$
  - $(A \land B) \land C = A \land (B \land C)$
- Distributive Law
  - $A \land (B \lor C) = (A \land B) \lor (A \land C)$
  - $A \lor (B \land C) = (A \lor B) \land (A \lor C)$
  - $(A \lor B) \land (C \lor D) = (A \land C) \lor (A \land D) \lor (B \land C) \lor (B \land D)$
  - $(A \land B) \lor (C \land D) = (A \lor C) \land (A \lor D) \land (B \lor C) \land (B \lor D)$

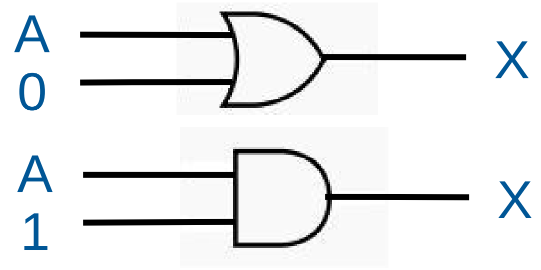
Boolean algebra is similar to real number algebra. Compare with the slide on the previous page

Other Laws

- Annulment Law
  - $A \,.\, 0 = 0$
  - $A + 1 = 1$

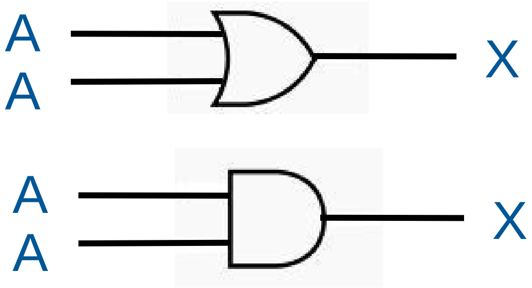


- Identity Law
  - $A + 0 = A$
  - $A \,.\, 1 = A$

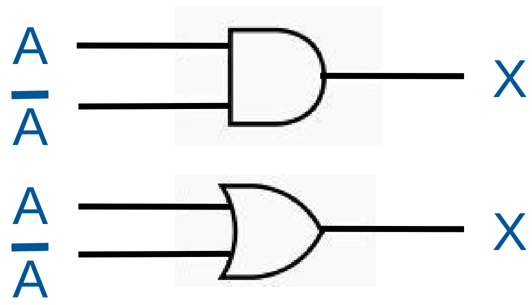


Other Laws

- Indempotent Law
  - $A + A = A$
  - $A \cdot A = A$

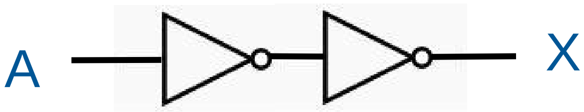


- Complement Law
  - $A \cdot \overline{A} = 0$
  - $A + \overline{A} = 1$



Other Laws

- Double Negation Law
  - $\overline{\overline{A}} = A$



- You can also write negation as
  - **¬A**
  - **~A**

Example

• Q	$= (A + B)(A + C)$	
	$= AA + AC + AB + BC$	Distributive law
	$= A + AC + AB + BC$	Identity AND law ($A.A = A$)
	$= A(1 + C) + AB + BC$	Distributive law
	$= A.1 + AB + BC$	Identity OR law ($1 + C = 1$)
	$= A(1 + B) + BC$	Distributive law
	$= A.1 + BC$	Identity OR law ($1 + B = 1$)
Q	$= A + BC$	Identity AND law ($A.1 = A$)

Example

- $Q = A + AB$

  $= A(1+B)$ Distributive law

  $= A(1)$ Identity OR law $(1 + B = 1)$

  $= A$

- $Q = A + \neg AB$

  $= A + AB + \neg AB$ See above example

  $= A + B(A + \neg A)$ Distributive law

  $= A + B(1)$ Complement law

  $= A + B$

Example

• Q	$= (A \lor B) \land (A \lor C)$	
	$= (A \land A) \lor (A \land C) \lor (A \land B) \lor (B \land C)$	Distributive law
	$= A \lor (A \land C) \lor (A \land B) \lor (B \land C)$	Identity AND law ($A \land A = A$)
	$= (A \land (1 \lor C)) \lor (A \land B) \lor (B \land C)$	Distributive law
	$= (A \land 1) \lor (A \land B) \lor (B \land C)$	Identity OR law ($1 \lor C = 1$)
	$= (A \land (1 \lor B)) \lor (B \land C)$	Distributive law
	$= (A \land 1) \lor (B \land C)$	Identity OR law ($1 \lor B = 1$)
Q	$= A \lor (B \land C)$	Identity AND law ($A \land 1 = A$)

Lecture 5

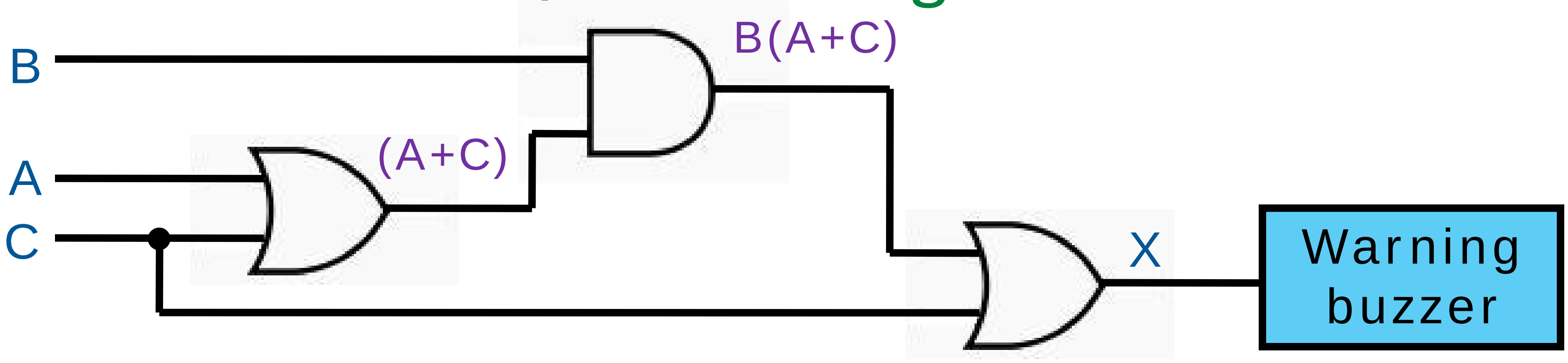
Boolean Logic & Logic Gates II

SIMPLIFICATION OF LOGIC CIRCUITS

Simplification of Logic Circuits

- Boolean Algebra is used to simplify logic circuits. Simplification is very important for reduction of circuit cost, physical size, and gate/circuit failures

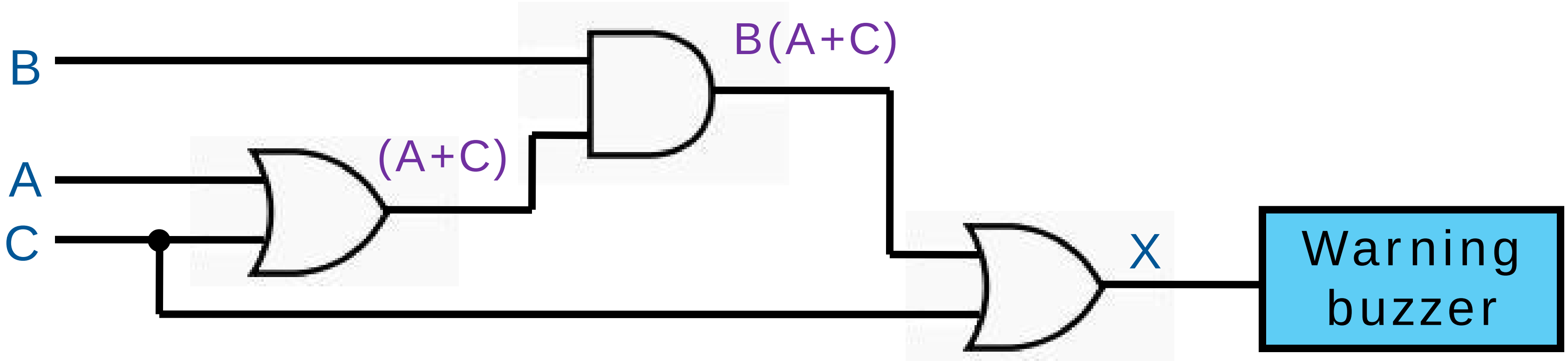
Example: Warning buzzer



Example: Warning Buzzer

- Equation:

  $X = B\,(A + C\,) + C$



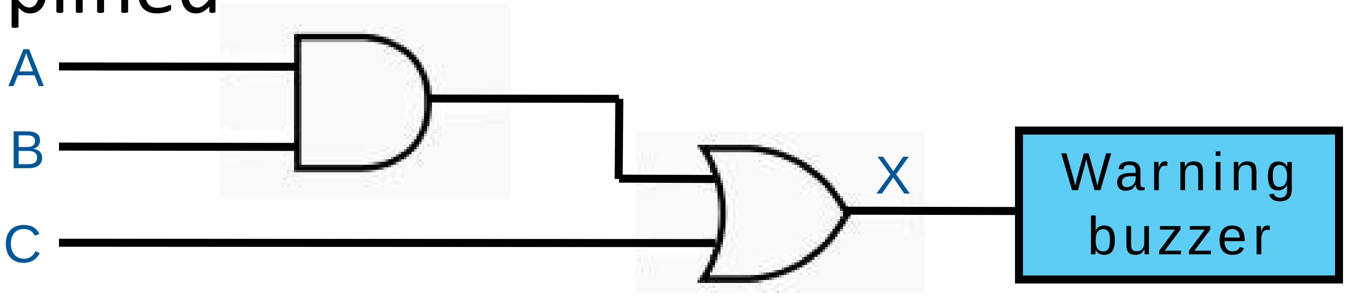
Example: Warning Buzzer

- Let's Simplify

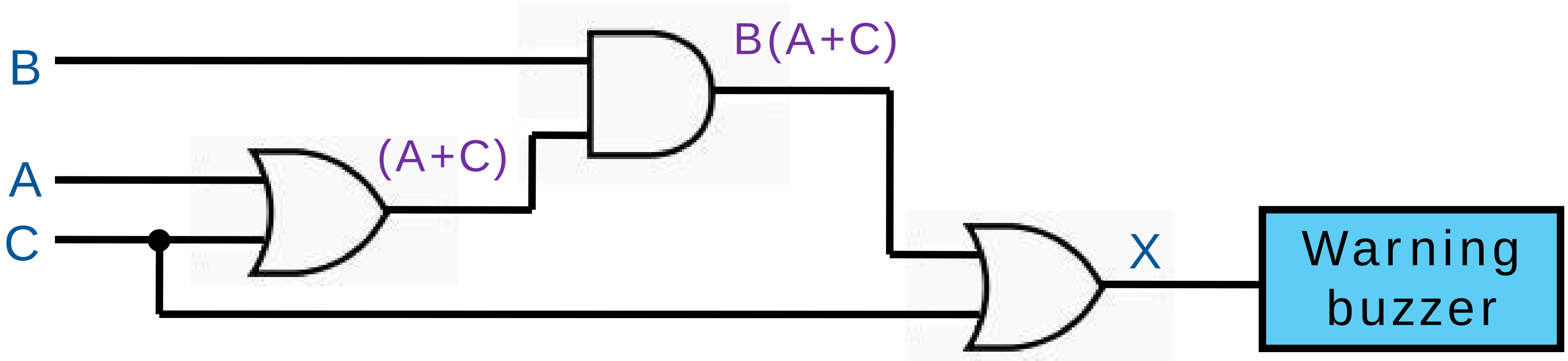
$X = B(A + C) + C$	Original Equation
$X = BA + BC + C$	**Distributive Law**
$X = BA + BC + C.1$	**Identity Law**
$X = BA + C(B + 1)$	**Distributive Law**
$X = BA + C.1$	**Identity Law**
$\mathbf{X = BA + C}$	**Identity law**
$X = AB + C$	**Commutative Law**

Example: Warning Buzzer

- Simplified

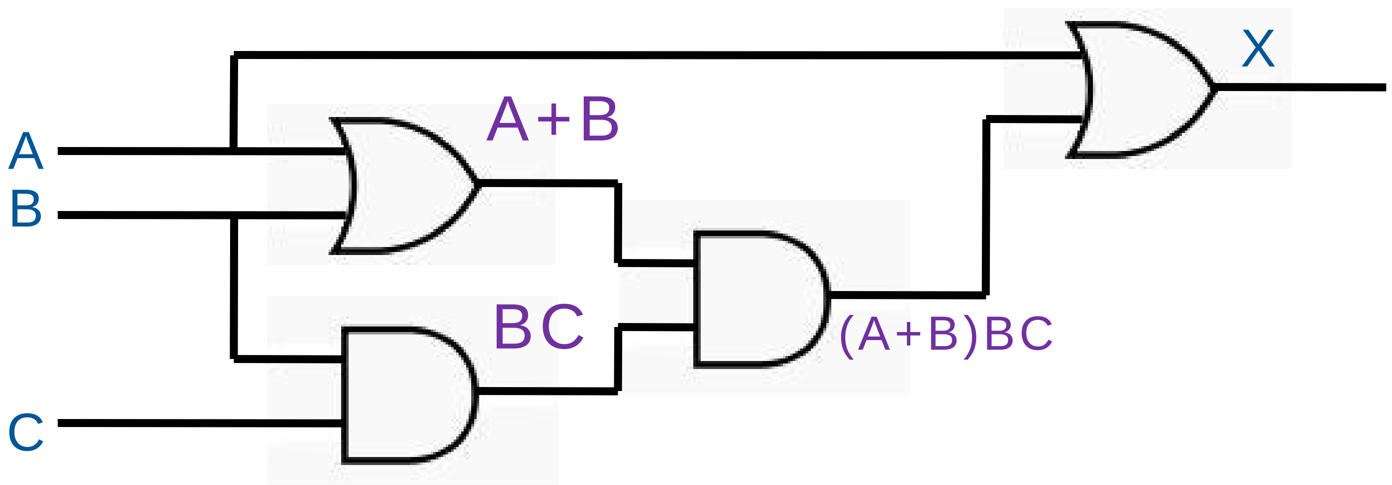


- Original



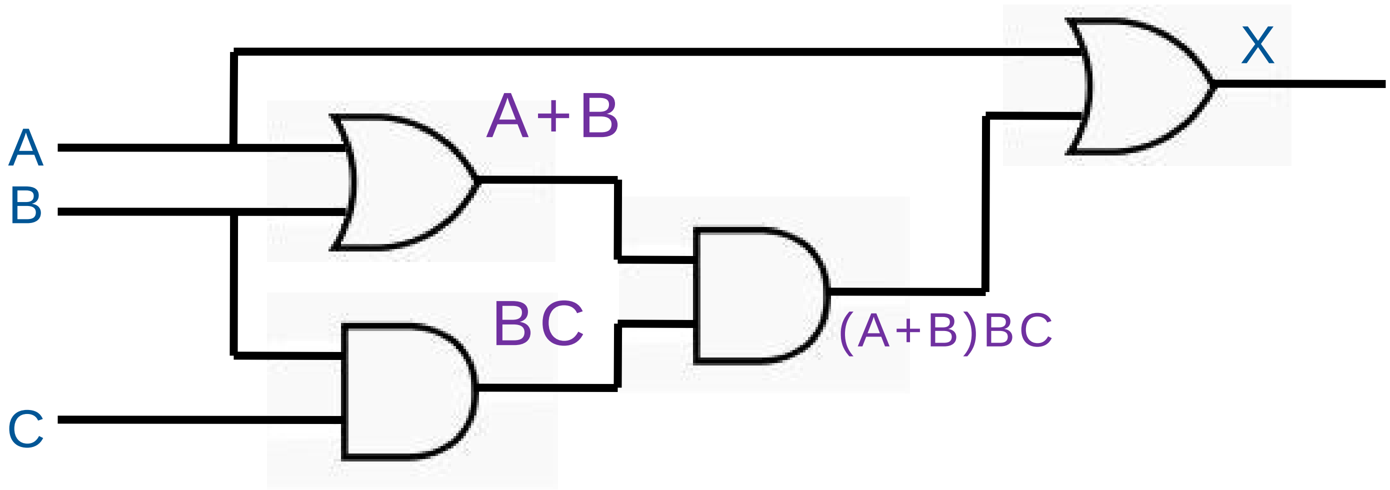
Another Example

- What is the boolean equation?



Another Example

- X = ((A + B) . (B . C)) + A = (A + B)BC + A
- Or
  - X = ((A ∨ B) ∧ (B ∧ C)) ∨ A
- In C++
  - X = ((A || B) && (B && C)) || A



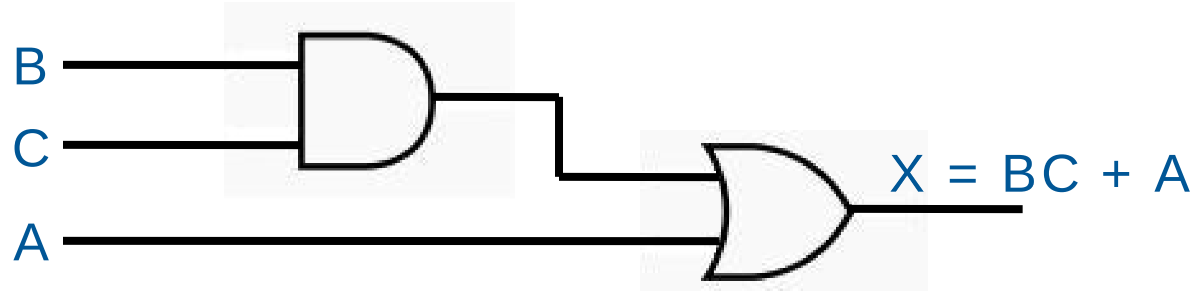
Example

- Let's Simplify

$X = (A + B)BC + A$	Original Equation
$X = \mathbf{ABC + BBC} + A$	**Distributive Law**
$X = ABC + \mathbf{BC} + A$	**Indempotent Law**
$X = ABC + \mathbf{BC.C} + A$	**Indempotent Law**
$X = \mathbf{BC\,(A + 1)} + A$	**Distributive Law**
$X = BC.1 + A$	**Identity Law**
$\mathbf{X = BC + A}$	**Identity Law**

Example

- Simplified logic circuit



- In C++, the simplified Boolean Equation is:
  - **X = (B && C) || A**

- Simplifying Boolean Equations in the program code makes it **more efficient** since there are **less conditions** for the CPU to check through

Lecture 5

Boolean Logic & Logic Gates II

DEMORGAN'S THEOREM

DeMorgan's Theorem

- Expressed in English:
  - The <u>**negation**</u> of a <u>**conjunction**</u> is the <u>**disjunction**</u> of the negations
  - The <u>**negation**</u> of a <u>**disjunction**</u> is the <u>**conjunction**</u> of the negations

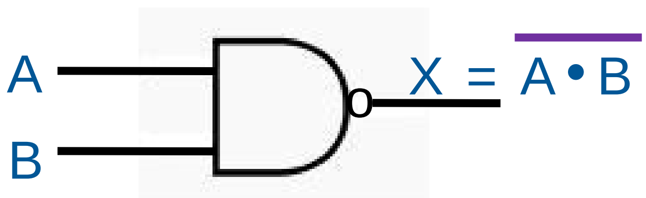
$$\neg(P \wedge Q) \iff (\neg P) \vee (\neg Q)$$
$$\neg(P \vee Q) \iff (\neg P) \wedge (\neg Q)$$

DeMorgan's Theorem

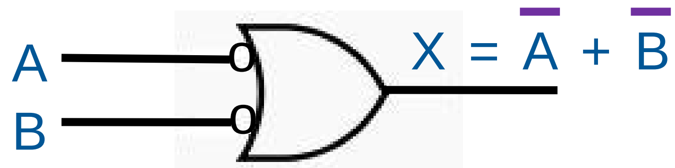
- **States that** $\boxed{\overline{A \bullet B} = \overline{A} + \overline{B}}$ and $\boxed{\overline{A + B} = \overline{A} \bullet \overline{B}}$
- **For 3 or more variables:**

$\boxed{\overline{A \bullet B \bullet C} = \overline{A} + \overline{B} + \overline{C}}$ and $\boxed{\overline{A + B + C} = \overline{A} \bullet \overline{B} \bullet \overline{C}}$

As proof for 2 variables:



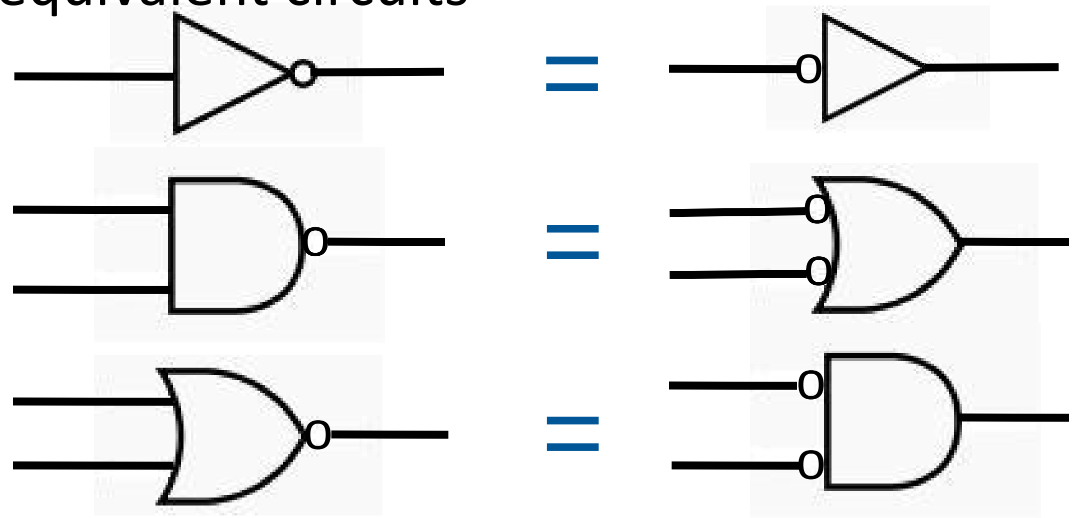
A	B	$X = \overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0



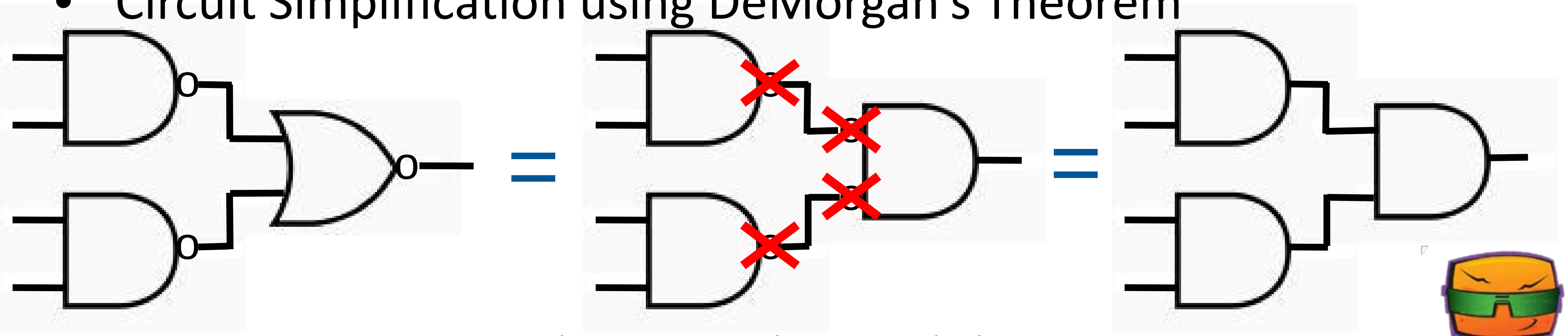
A	B	$X = \overline{A} + \overline{B}$
0	0	1
0	1	1
1	0	1
1	1	0

DeMorgan's Theorem

- Gives us equivalent circuits

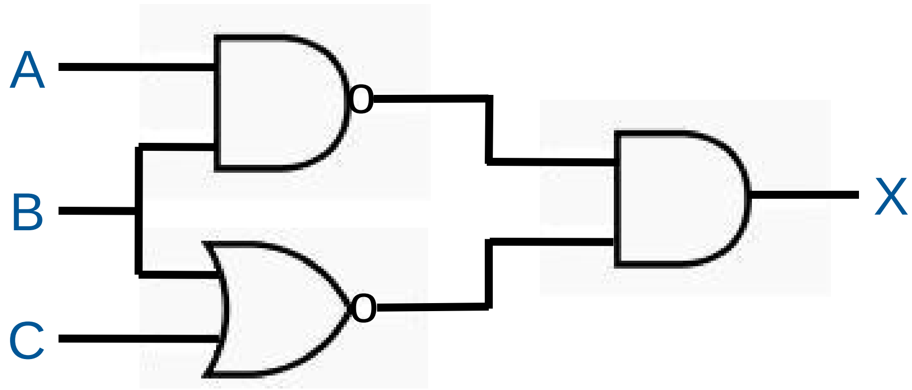


- Circuit Simplification using DeMorgan's Theorem



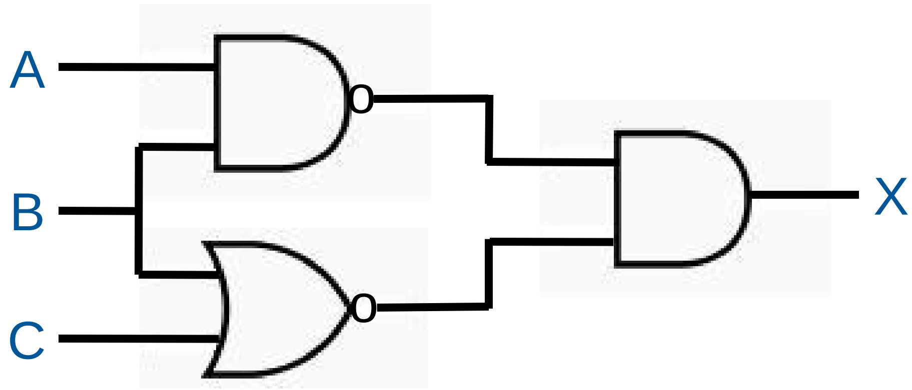
Application of DeMorgan's Theorem

- Write the equation of the following circuit



Application of DeMorgan's Theorem

- Boolean Equation at X is:
  - $X = \overline{AB} \,.\, (\overline{B + C})$
- Apply DeMorgan's Theorem
  - $X = (\,\overline{A} + \overline{B}\,) \,.\, (\overline{B}.\overline{C})$

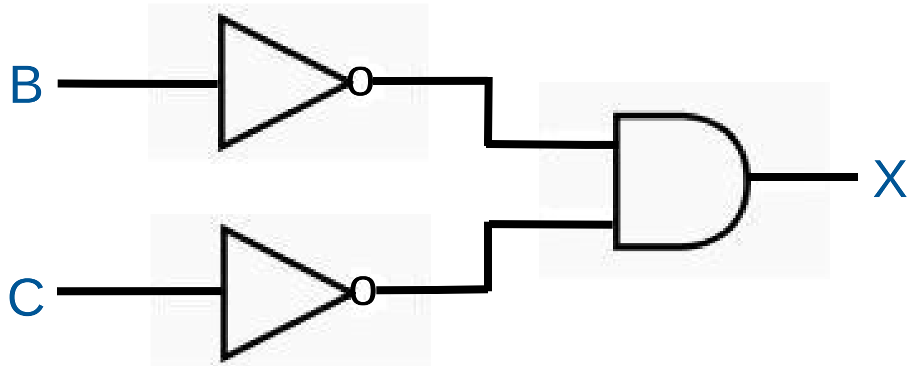


Application of DeMorgan's Theorem

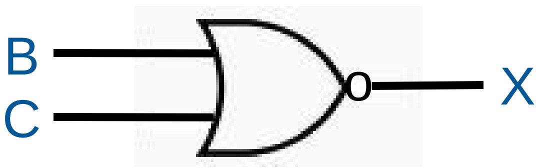
- Using Boolean Algebra
  - $X = (\overline{A} + \overline{B}).\overline{B}\,\overline{C}$ — Original Equation
  - $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{B}\,\overline{C}$ — **Distributive Law**
  - $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{C}$ — **Indempotent Law**
  - $X = \overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{C}.1$ — **Identity Law**
  - $X = \overline{B}\,\overline{C}\,\overline{A} + \overline{B}\,\overline{C}.1$ — **Commutative Law**
  - $X = \overline{B}\,\overline{C}(\overline{A} + 1)$ — **Distributive Law**
  - **$X = \overline{B}\,\overline{C}$** — **Identity Law**
  - $X = \neg\neg\overline{B}\,\overline{C}$ — **Double Negation Law**
  - $X = \neg(\overline{\overline{B}} + \overline{\overline{C}})$ — **DeMorgan's Theorem**
  - **$X = \neg(B + C)$** — **Double Negation Law**

Application of DeMorgan's Theorem

- Original ($\mathbf{X = \overline{B}\,\overline{C}}$)



- Becomes $\mathbf{X = \neg (B + C)}$



Lecture 5

Boolean Logic & Logic Gates II

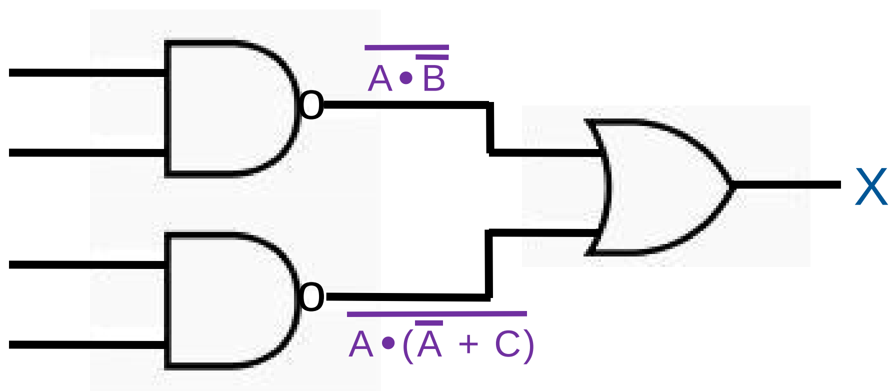
CONSTRUCTING A TRUTH TABLE

Constructing a Truth Table

- We have this equation

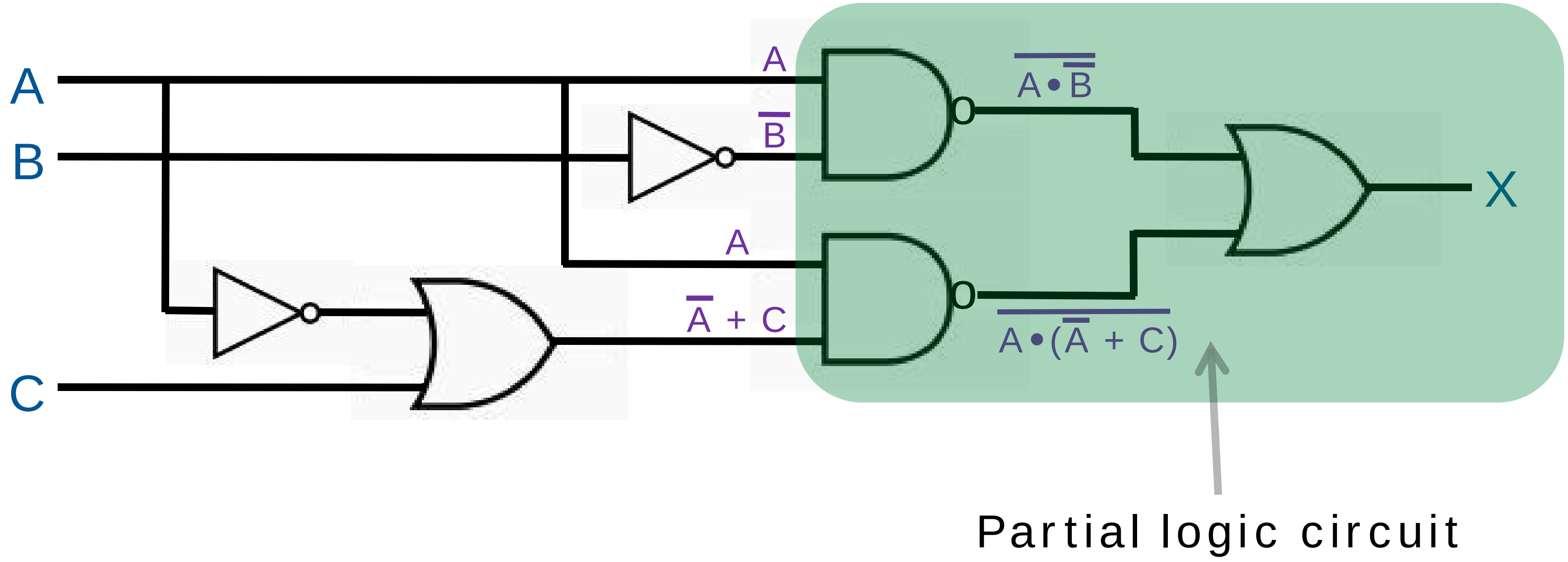
$$X = \overline{A \bullet \overline{B}} + \overline{A \bullet (\overline{A} + C)}$$

- The logic circuit



Constructing a Truth Table

- Full Logic Circuit is:



Constructing a Truth Table

- Simply

$X = \neg(A\neg B) + \neg(A (\neg A + C))$	Original Equation
$X = \neg A + \neg\neg B + \neg(A (\neg A + C))$	**DeMorgan's Theorem**
$X = \neg A + B + \neg(A(\neg A + C))$	**Double Negation Law**
$X = \neg A + B + (\neg A + \neg(\neg A + C))$	**DeMorgan's Theorem**
$X = \neg A + B + (\neg A + (\neg\neg A . \neg C))$	**DeMorgan's Theorem**
$X = \neg A + B + (\neg A + A\neg C)$	**Double Negation Law**
$X = \neg A + \neg A + A\neg C + B$	**Commutative Law**
$X = \neg A + A\neg C + B$	**Indempotent Law**

Continue next slide:

Constructing a Truth Table

- Simply

$X = \neg A + A\neg C + B$	Continued:
$X = \neg(\neg\neg A.\neg(A\neg C)) + B$	DeMorgan's Theorem
$X = \neg(A.\neg(A\neg C)) + B$	Double Negation Law
$X = \neg(A(\neg A + \neg\neg C) + B$	DeMorgan's Theorem
$X = \neg(A(\neg A + C) + B$	Double Negation Law
$X = \neg(A\neg A + AC) + B$	Distributive Law
$X = \neg(0 + AC) + B$	Complement Law
$X = \neg(AC) + B$	Identity Law
$\mathbf{X = \neg A + \neg C + B}$	DeMorgan's Theorem

Note:

.

$\boxed{A + \overline{A}B = A + B}$ and $\boxed{\overline{A} + AB = \overline{A} + B}$

Constructing the Truth Table

- Truth Table is:

A	B	C	$X = \overline{A} + \overline{C} + B$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

X = 1 whenever A = 0 OR C = 0 OR B = 1

Lecture 5

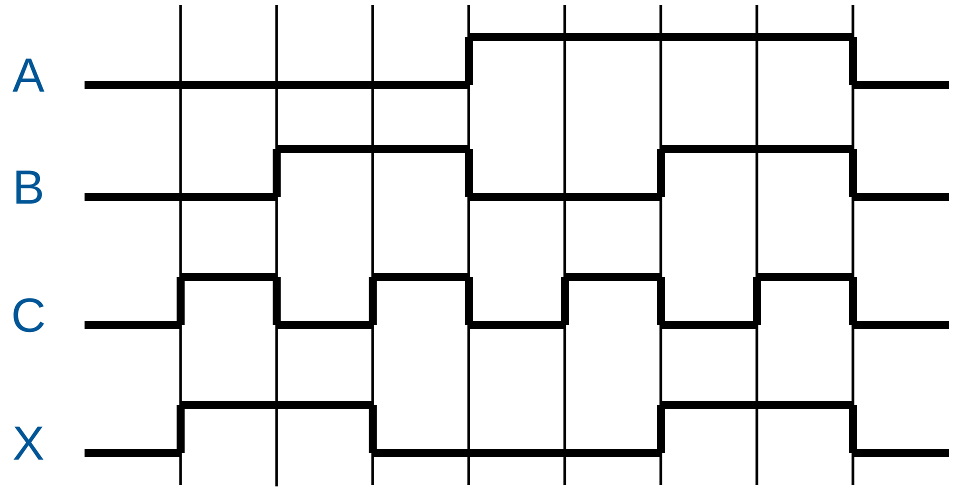
Boolean Logic & Logic Gates II

DRAWING A TIMING DIAGRAM

Drawing a Timing Diagram

- $X = AB + B\bar{C} + \bar{A}\bar{B}C$
- Truth table & Timing Diagram:

A	B	C	X	
0	0	0	0	
0	0	1	1	← $\bar{A}\bar{B}C$
0	1	0	1	← $B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	← AB, $B\bar{C}$
1	1	1	1	← AB



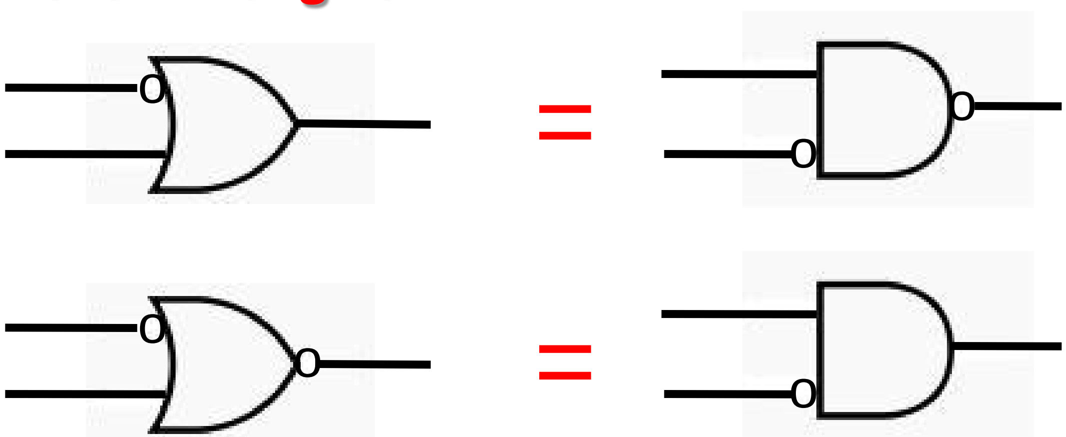
Lecture 5

Boolean Logic & Logic Gates II

BUBBLE PUSHING

Bubble Pushing

- Another trick to form an equivalent logic circuit is called bubble pushing.
- Based on DeMorgan's Theorem:
  - **Change** the logic gate between AND and OR
    - **AND to OR or**
    - **OR to AND**
  - **Add bubbles** to the input and outputs where there were **none**, and **remove** the **original** bubbles.



Lecture 5

Boolean Logic & Logic Gates II

PRODUCT-OF-SUMS & SUM-OF-PRODUCTS

Product-of-Sums & Sum-of-Products

- Most Boolean equations can be simplified into one of the following forms:
  - **Product-of-Sums** Expression (**POS**)
  - **Sum-of-Products** Expression (**SOP**)
- Example of **POS** expression
  - $X = (\overline{B} + \overline{C} + D)(BC + \overline{E})$
- Example of **SOP** expression
  - $X = AC\overline{D} + \overline{C}D + B$

Product-of-Sums & Sum-of-Products

- The **SOP** expression is **most often used** because it's **easy to figure out** truth tables, timing diagrams and Karnaugh maps
- We always strive to **simplify** a Boolean equation and then put it into the **SOP** form

Example

- Filling out the truth table for

$X = \neg(A\neg B + \neg CD)$

$= \neg(A\neg B) . \neg(\neg CD)$ **DeMorgan's Theorem**

$= (\neg A + \neg\neg B) . (\neg\neg C + \neg D)$ **DeMorgan's Theorem**

$\mathbf{= (\neg A + B) . (C + \neg D)}$ **Double Negation Law**

The above is the **POS** expression

Can you fill up the truth table from this form?

Example

- What if convert it to SOP form

$X = (\neg A + B) . (C + \neg D)$

- Using distributive Law

$X = \neg AC + \neg A\neg D + BC + B\neg D$

$= \overline{A}C + \overline{A}\,\overline{D} + BC + B\overline{D}$ ⟸ SOP

So to fill out the truth table,
X = 1 whenever A = 0 AND C = 1;
OR A = 0 AND D = 0;
OR B = 1 AND C = 1;
OR B = 1 AND D = 0.

Lecture 6

Boolean Logic & Logic Gates III

KARNAUGH MAPPING (TO BE CONTINUED)