UNIVERSITY OF MICHIGAN-DEARBORN

Chapter 10 : Boolean Algebra

Introduction

Boolean Algebra

- In mathematical logic, *Boolean algebra* is the branch of algebra in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively.
- Instead of elementary algebra where the values of the variables are numbers, and the main operations are addition and multiplication, the main operations of Boolean algebra are *the conjunction and*, denoted $\land$, *the disjunction or*, denoted $\lor$, and *the negation not*, denoted $\neg$. It is thus a formalism for describing logical relations in the same way that ordinary algebra describes numeric relations.
- *Boolean algebra* has been fundamental in the development of digital electronics, and is provided for in all modern programming languages. It is also used in set theory and statistics.

Boolean Algebra

Operator	Meaning
Not	Test if value is NOT something
And	Test for more than one condition
Or	Test if the value is either OR something
Xor	Test if one and only one value is true

Boolean Algebra

- The basic operations of Boolean calculus are as follows.

  - *AND (conjunction), denoted $x \wedge y$*, satisfies

    $$\begin{cases} x \wedge y = 1 \text{ if } x = y = 1 \\ x \wedge y = 0 \text{ otherwise} \end{cases}$$

  - *OR (disjunction), denoted $x \vee y$*, satisfies

    $$\begin{cases} x \vee y = 0 \text{ if } x = y = 0 \\ x \vee y = 1 \text{ otherwise} \end{cases}$$

  - *NOT (negation), denoted $\neg x$*, satisfies

    $$\begin{cases} \neg x = 0 \text{ if } x = 1 \\ \neg x = 1 \text{ if } x = 0 \end{cases}$$

UNIVERSITY OF MICHIGAN-DEARBORN

Boolean Algebra and Combinational Circuits

Boolean Algebra

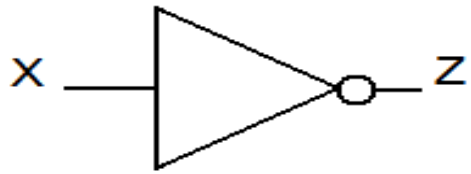
DEARBORN

	BOOLEAN EXPRESSION	TRUTH TABLE	CIRCUIT SYMBOL	KARNAUGH MAP

NOT

$Z = \overline{X}$

X	Z
0	1
1	0

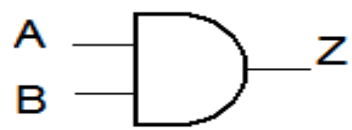


X	0	1
	1	0

AND

$Z = AB$

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



B \ A	0	1
0	0	0
1	0	1

OR

$Z = A + B$

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

B \ A	0	1
0	0	1
1	1	1

Boolean Algebra

Logical operations

NOT

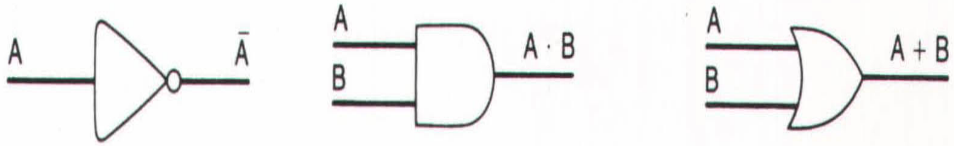
A	$\bar{A}$
0	1
1	0

AND

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR

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



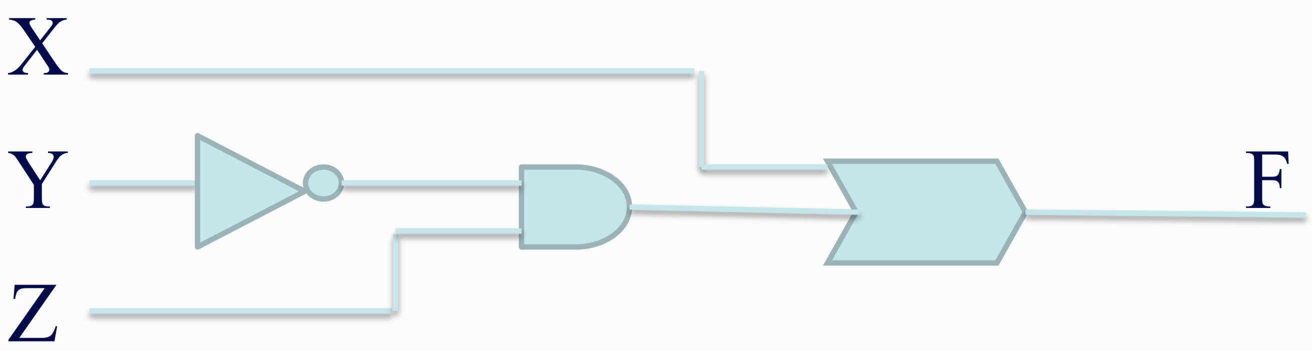
Boolean Algebra

Function	x	0	0	1	1
	y	0	1	0	1
Constant 0	0	0	0	0	0
And	$x \cdot y$	0	0	0	1
x And Not y	$x \cdot \bar{y}$	0	0	1	0
x	x	0	0	1	1
Not x And y	$\bar{x} \cdot y$	0	1	0	0
y	y	0	1	0	1
Xor	$x \cdot \bar{y} + \bar{x} \cdot y$	0	1	1	0
Or	$x + y$	0	1	1	1
Nor	$\overline{x + y}$	1	0	0	0
Equivalence	$x \cdot y + \bar{x} \cdot \bar{y}$	1	0	0	1
Not y	$\bar{y}$	1	0	1	0
If y then x	$x + \bar{y}$	1	0	1	1
Not x	$\bar{x}$	1	1	0	0
If x then y	$\bar{x} + y$	1	1	0	1
Nand	$\overline{x \cdot y}$	1	1	1	0
Constant 1	1	1	1	1	1

Figure 1.2 All the Boolean functions of two variables.

Boolean Algebra

Boolean Functions



Y'Z	Y'	X	Y	Z	F
0	1	0	0	0	0
1	1	0	0	1	1
0	0	0	1	0	0
0	0	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1

Let X,Y,Z be Boolean variables

THEN, **F = X + Y'Z**

is a Boolean function

Boolean Algebra

- *Axioms of Boolean Algebra*

AND	OR
$0 * X = 0$	$0 + X = X$
$1 * X = X$	$1 + X = 1$
$X * X = X$	$X + X = X$
$X * X' = 0$	$X' + X = 1$

Boolean Algebra

- *Theorem 1.* $x + xy = x$

  Proof.
  $$x + xy = x(1 + y) = x * 1 = x$$

- Theorem 2. $x + x'y = x + y$
  - Proof
  $$\begin{aligned} x + x'y &= (x + xy) + x'y \\ &= x + y(x + x') \\ &= x + y(1) = x + y \end{aligned}$$

- *Theorem 3.* $x + yz = (x + y)(x + z)$
  - Proof
  $$\begin{aligned} (x + y)(x + z) &= xx + xz + yx + yz \\ &= x + xz + yx + yz \\ &= x(1 + z + y) + yz \\ &= x(1 + y) + yz \\ &= x(1) + yz = x + yz \end{aligned}$$

- **Theorem De Morgan's Law**

  a) $(x + y)' = x' * y' = x'y'$

  b) $(xy)' = x' + y'$

  c) $(x')' = x$

Applications

DEARBORN

Boolean Algebra

- Simplify: $(x + y' + x'y + (x + y')\,x'y) = x + y' + x'y + xx'y + y'x'y$

  $= x + y' + x'y + (xx' = 0)y + (y'y = 0)x'$

  $= x + y' + x'y \rightarrow (x + x'y = x + y$ by theorm 2$)$

  $= (x + x')(x + y) + y'$ how?

  $= 1(x + y) + y'$

- Ex. Boolean algebra is used to simplify circuits

  $x = [(A' + B) * B']'$

Using De Morgan's Law

$x = (A' + B)' + (B')'$

$= (A')'*B' + (B')'$

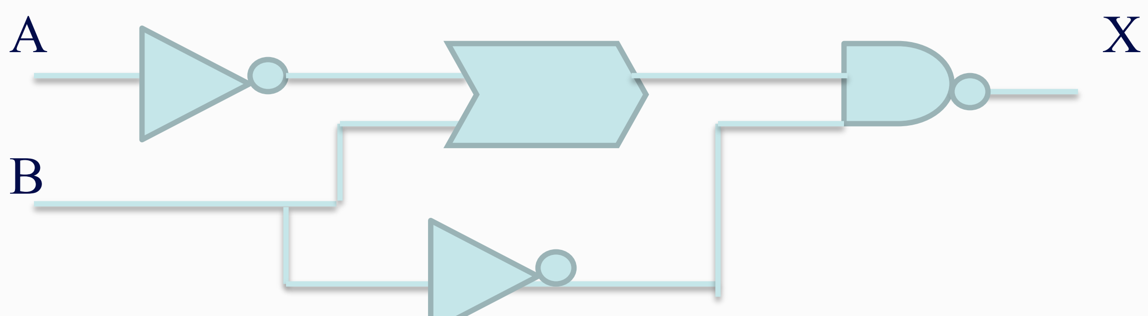
$= A * B' + B$ (thm 2)

$= (A + B)*(B' + B)$

$= (A + B)*1$

$= A + B$

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

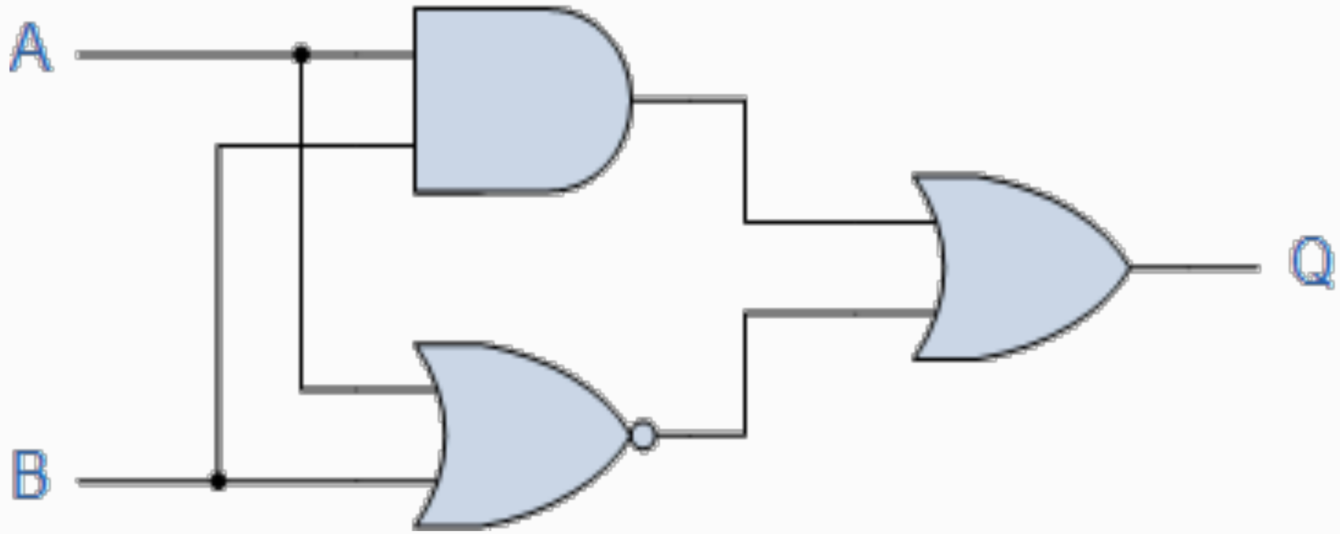


Exercises

Boolean Algebra

- **Exercise 1:**

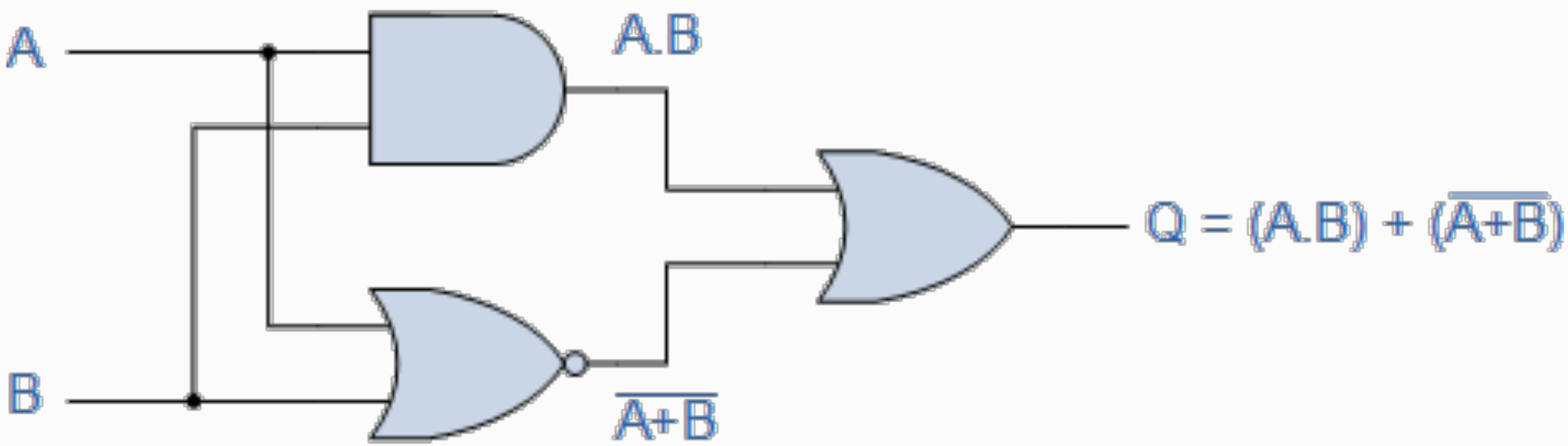
Find the Boolean algebra expression for the following system.



The system consists of an AND Gate, a NOR Gate and finally an OR Gate.

Boolean Algebra

The expression for the AND gate is A.B, and the expression for the NOR gate is $\overline{A+B}$. Both these expressions are also separate inputs to the OR gate which is defined as A+B. Thus the final output expression is given as:



Boolean Algebra

- The output of the system is given as $Q = (A.B) + (\overline{A+B})$, but the notation $\overline{A+B}$ is the same as the De Morgan´s notation $\bar{A}.\bar{B}$, Then substituting $\bar{A}.\bar{B}$ into the output expression gives us a final output notation of $Q = (A.B)+(\bar{A}.\bar{B})$, which is the Boolean notation for an Exclusive-NOR Gate as seen in the previous section.

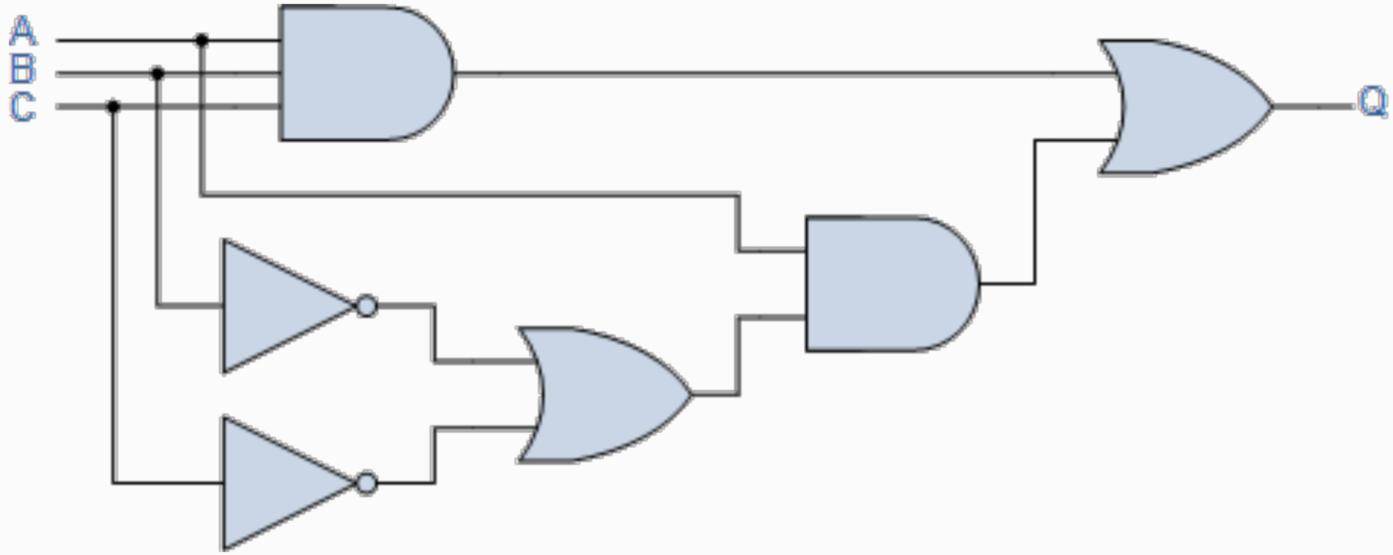
Inputs		**Intermediates**		**Output**
B	**A**	**A.B**	$\overline{\mathbf{A+B}}$	**Q**
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Then, the whole circuit above can be replaced by just one single Exclusive-NOR Gate and indeed an Exclusive-NOR Gate is made up of these individual gate functions.

Boolean Algebra

- **Exercise 2:**

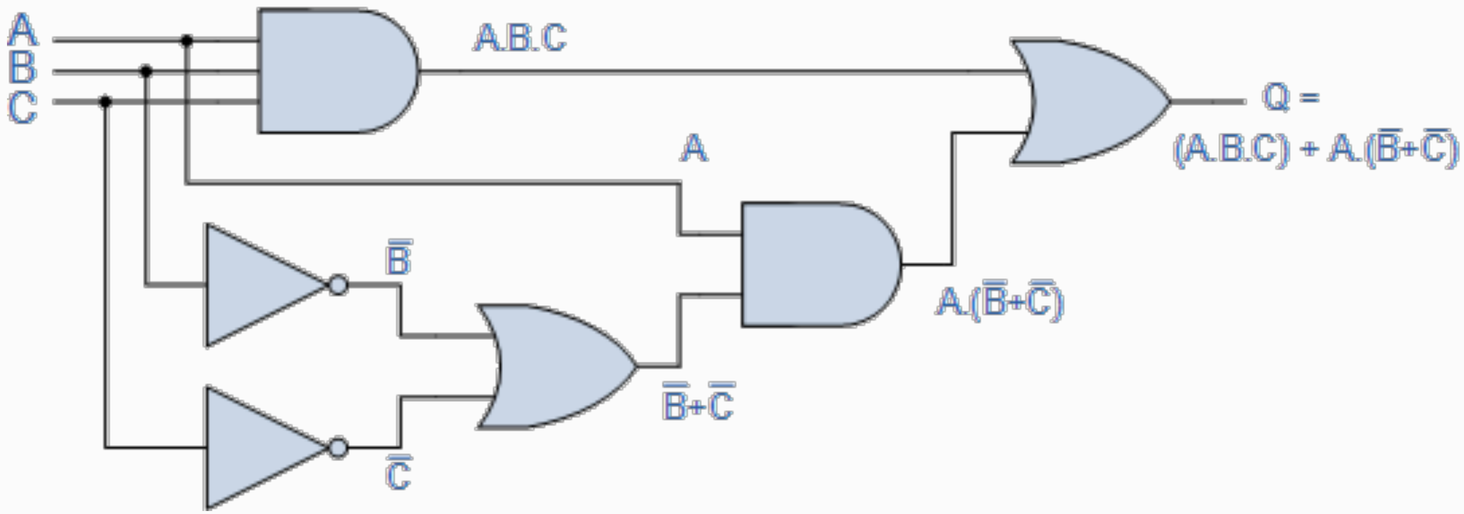
Find the Boolean algebra expression for the following system.



This system may look more complicated than the other two to analyse but again, the logic circuit just consists of simple AND, OR and NOT gates connected together.

Boolean Algebra

- we can simplify the circuit by writing down the Boolean notation for each logic gate function in turn in order to give us a final expression for the output at Q.



The output from the 3-input AND gate is only at logic "1" when **ALL** the gates inputs are HIGH at logic level "1" (A.B.C). The output from the lower OR gate is only a "1" when one or both inputs B or C are at logic level "0". The output from the 2-input AND gate is a "1" when input A is a "1" and inputs B or C are at "0". Then the output at Q is only a "1" when inputs A.B.C equal "1" or A is equal to "1" and both inputs B or C equal "0", A.($\overline{B}+\overline{C}$).

Boolean Algebra

- By using "**de Morgan's theorem**" inputs B and input C cancel out as to produce an output at Q they can be either at logic "1" or at logic "0". Then this just leaves input A as the only input needed to give an output at Q as shown in the table below.

	Inputs				Intermediates			Output
C	B	A	A.B.C	$\overline{B}$	$\overline{C}$	$\overline{B}+\overline{C}$	$A.(\overline{B}+\overline{C})$	Q
0	0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1