Boolean Algebra

- Basic definitions (2.2, 2.3)
- Basic theorems (2.4)
- Boolean functions (2.5)
- Other logic operations (2.7)
- Gates (2.8)
- Integrated circuits (2.9)
- Minterms/maxterms (2.6)

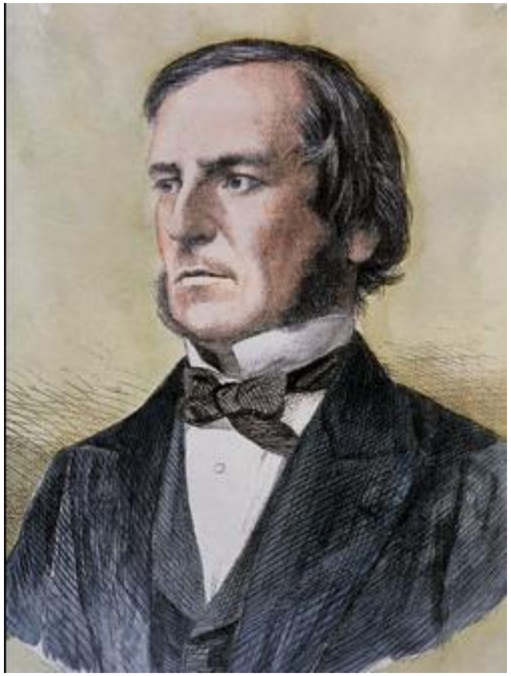
2.2 Basic definitions

- Boolean algebra is named after George Boole, English mathematician, educator, philosopher and logician.

- Best known as the author of *The Laws of Thought* (1854) which contains Boolean algebra.

- Shannon (1937) discovered how to optimize digital systems using Boole's work.

- There is a crater on the moon named after him, *Boole.*



2.2 Basic definitions

A **variable, or literal** is a symbol used to represent a condition, or data. Each variable has a value of 1 or 0.

The **complement** represents the inverse of a variable; indicated by an over-bar. Thus, the complement of *A* is $\overline{A}$ *or A'*.

Closure. The result of any operation is a 1 or 0.

2.4 Basic definitions: Commutative Laws

The **commutative laws** apply to both OR and AND. For OR, the commutative law states:

> The order in which variables in the OR operations makes no difference.

$$\boxed{A + B = B + A}$$

For AND, the commutative law states:

> The order in which variables in the AND operations makes no difference.

$$\boxed{AB = BA}$$

2.4 Basic definitions: Associative Laws

The **associative laws** also apply to both OR and AND. For OR, the associative law states:

When forming the OR of more than two variables, the result is the same regardless of the grouping of the variables.

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

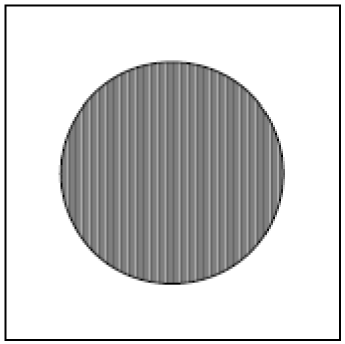
For AND, the associative law states:

When forming the AND of more than two variables, the result is the same regardless of the grouping of the variables.

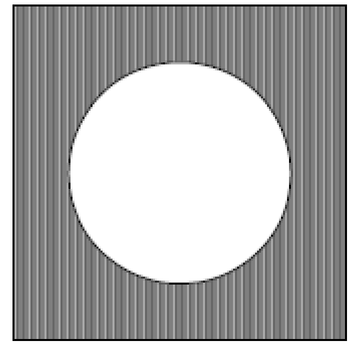
$$A(BC) = (AB)C$$

2.4 Basic definitions:

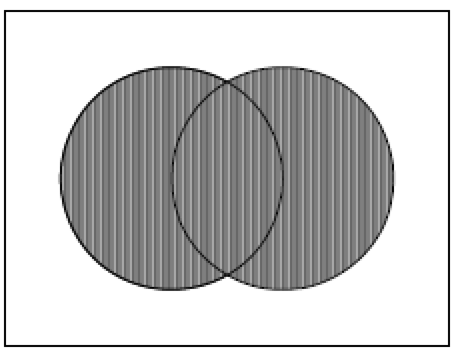
Venn Diagrams for Boolean Algebra



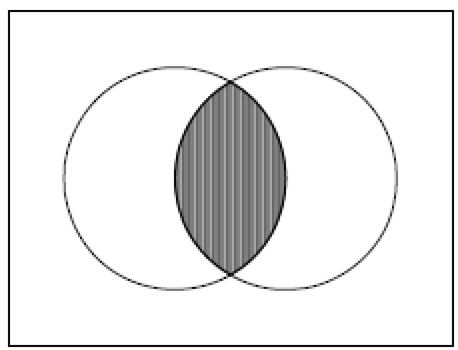
A



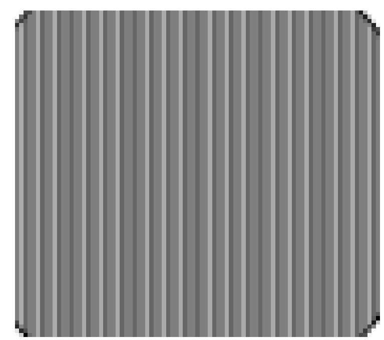
$\overline{A}$



A + B



A . B



1

2.4 Basic definitions: Order preference

1 Parenthesis

2 AND

3 OR

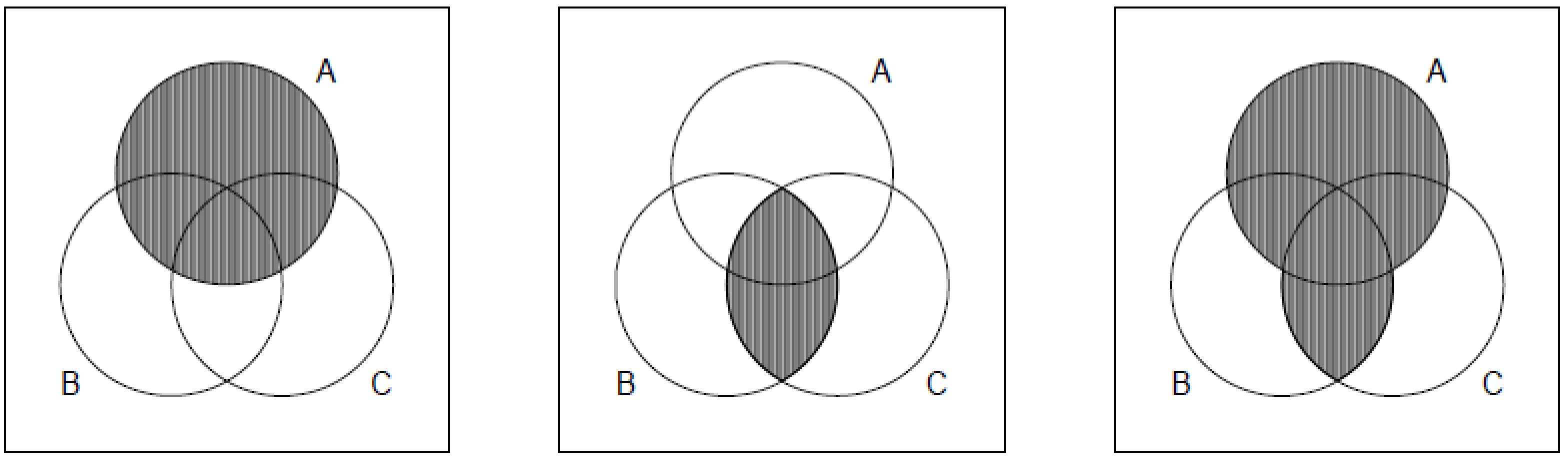
A OR B AND C OR D

$= A + B \cdot C + D$

$= A + BC + D = A + (BC) + D$

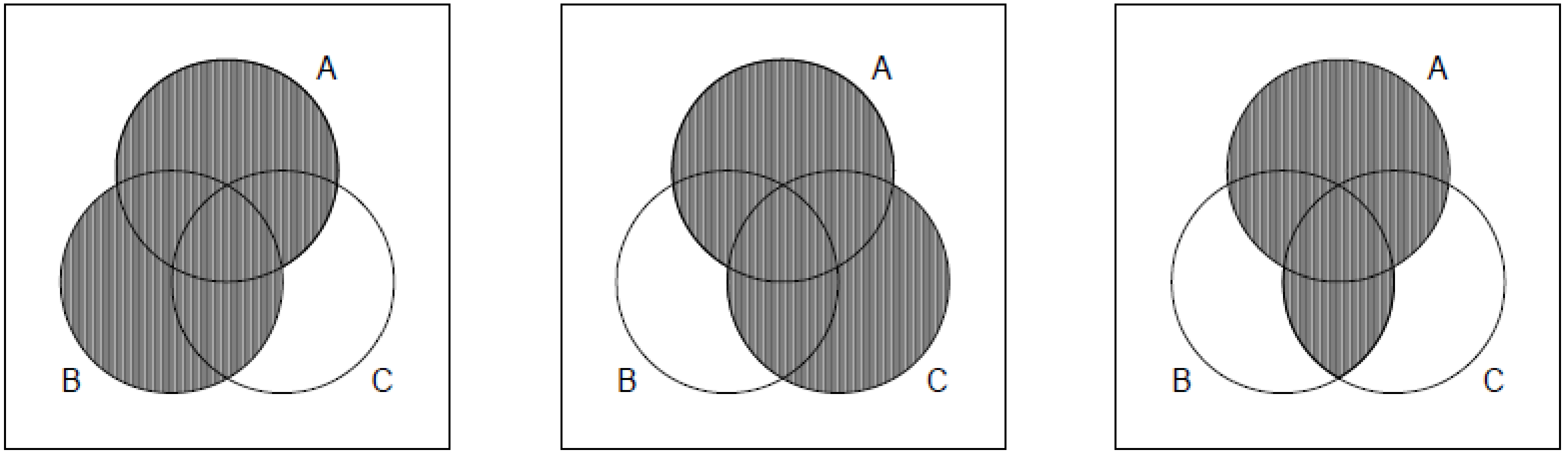
2.4 Basic definitions:

Venn Diagrams for Boolean Algebra



A B . C A + B . C

2.4 Basic definitions:
Venn Diagrams for Boolean Algebra



A + B   A + C   (A + B) . (A + C)

Can describe the same set (function) in different ways

2.4 Basic functions

AND

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

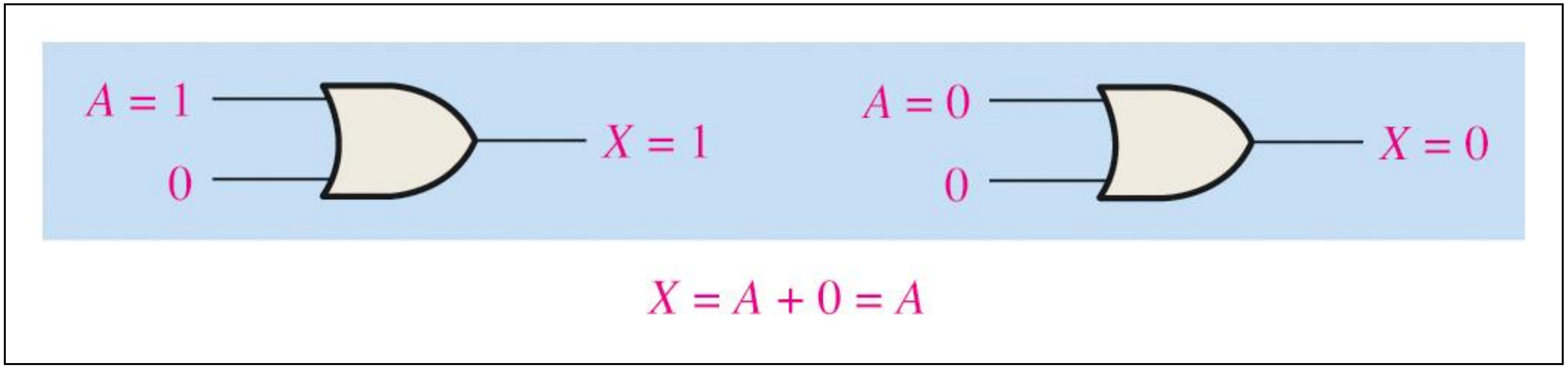
2.4 Basic Theorems

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \overline{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

2.4 Basic Theorems

Rule 1: $A + 0 = A$

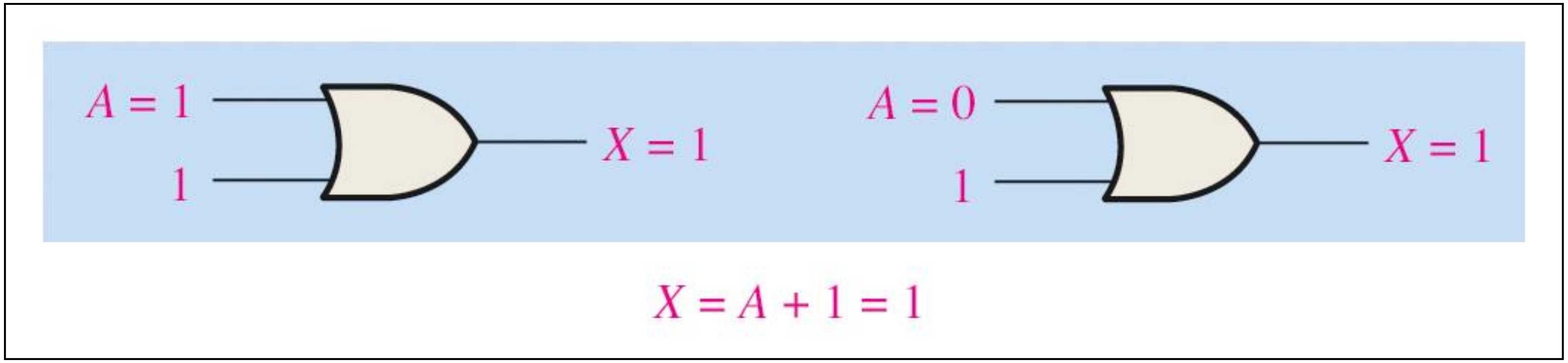
When $A = 1$, the input causes the output to go to $X = 1$.
When $A = 0$, the 0 inputs cause the output to go to $X = 0$.
In either case, the value of X equals the value of A.



2.4 Basic Theorems

Rule 2: $A + 1 = 1$

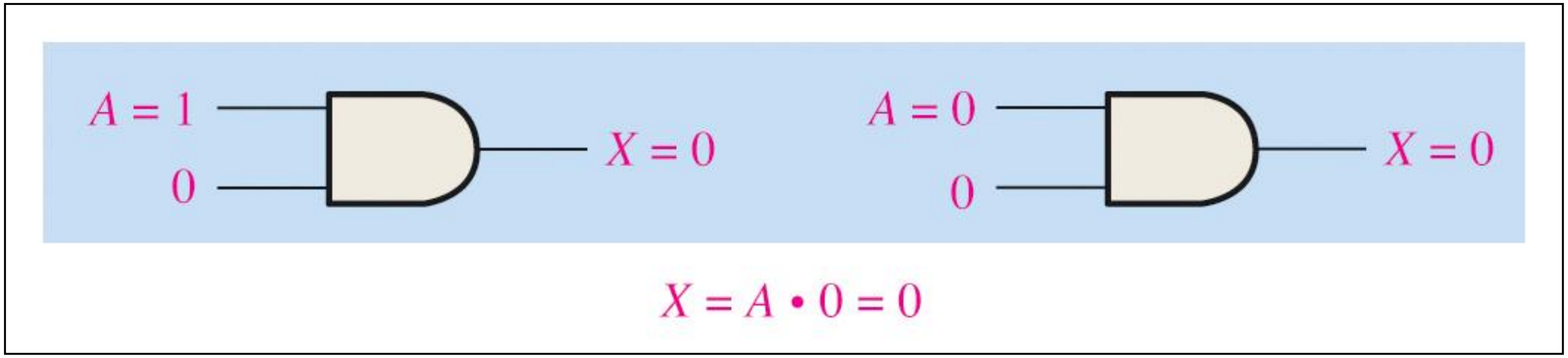
When $A = 1$, the inputs cause the output to go to $X = 1$.
When $A = 0$, the 1 input caused the output to go to $X = 1$.
In either case, the value of X equals one (1).



2.4 Basic Theorems

Rule 3: $\boldsymbol{A \cdot 0 = 0}$

When either input to an AND gate equals 0, the output from the gate has a value of $X = 0$, regardless of the value at the other input.



2.4 Basic Theorems

Rule 4: $A \bullet 1 = A$

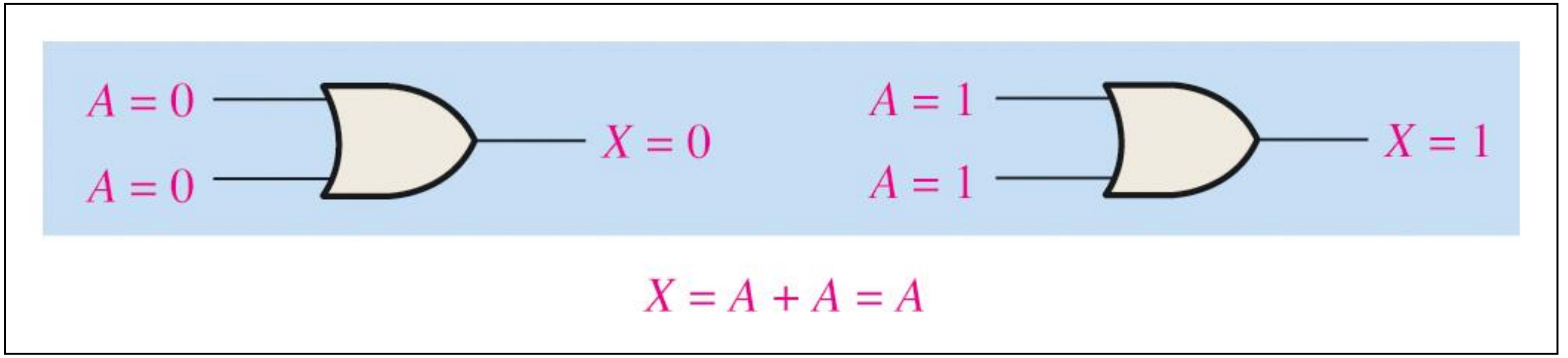
When one input to an AND gate equals 1, the output from the gate has a value of $X = A$. As shown, $X = 1$ when $A = 1$ and $X = 0$ when $A = 0$.

$X = A \bullet 1 = A$

2.4 Basic Theorems

Rule 5: $A + A = A$

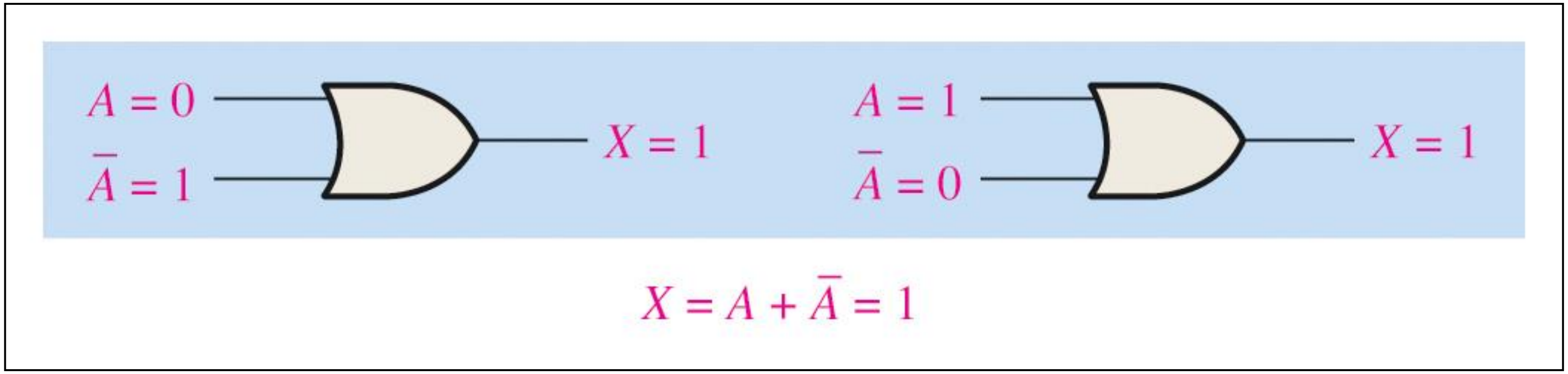
When the inputs to an OR gate are equal, the output equals the value at the inputs. When both inputs equal 1, the gate output is $X = 1$. When both inputs equal 0, the gate output is $X = 0$.



2.4 Basic Theorems

Rule 6: $\boldsymbol{A + \overline{A} = 1}$

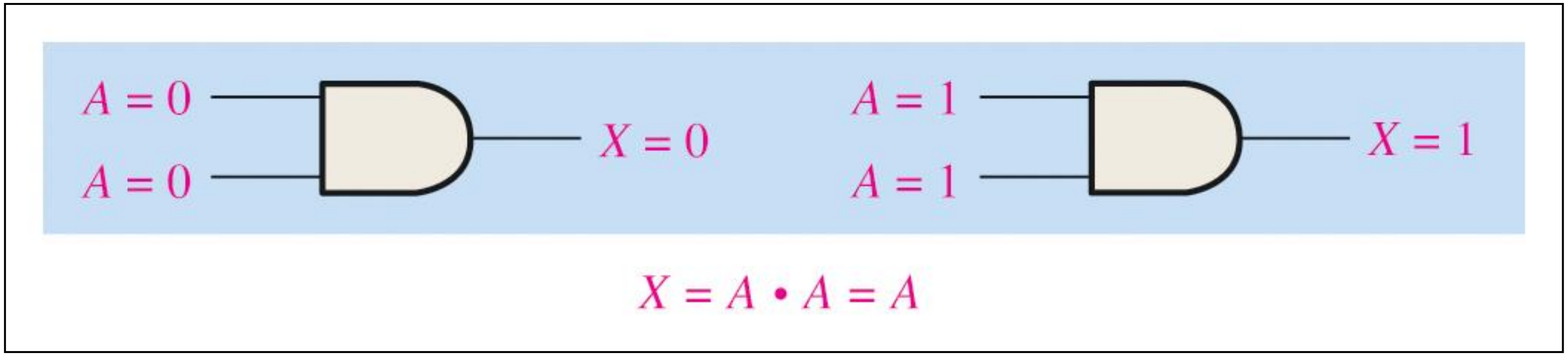
When the inputs to an OR gate are unequal (complements), one of the two always equals 1. When either input equals 1, the gate output is X = 1. Therefore, the output from the OR gate equals 1 whenever the inputs are unequal (complementary).



2.4 Basic Theorems

Rule 7: $\boldsymbol{A \cdot A = A}$

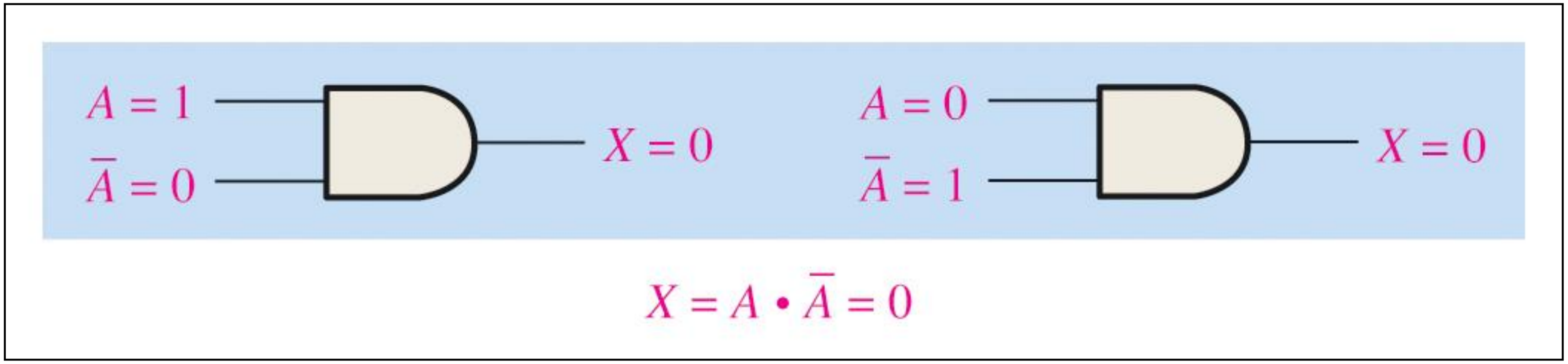
When the inputs to an AND gate are equal, the gate output also equals that value. Thus, $X = 1$ when both inputs equal 1 and $X = 0$ when both inputs equal 0.



2.4 Basic Theorems

Rule 8: $A \bullet \overline{A} = 0$

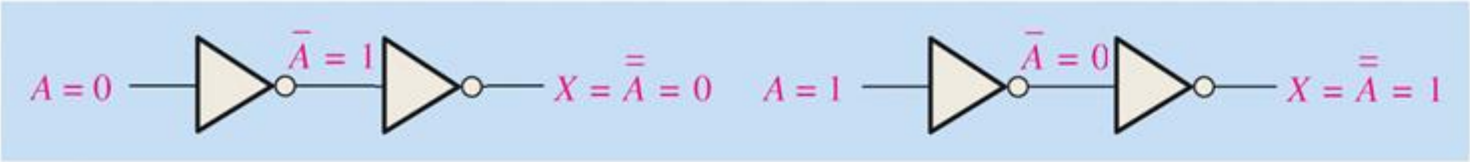
When the inputs to an AND gate are unequal (complements), one of the two always equals 0. When either input equals 0, the gate output is $X = 0$. Therefore, the output from the AND gate equals 0 whenever the inputs are unequal (complementary).



2.4 Basic Theorems

Rule 9: $\overline{\overline{A}} = A$

When a value is inverted, it is the complement of the original value. When inverted a second time, it returns to its original value. Thus, A = 0 inverted twice equals 0 and A = 1 inverted twice equals 1.



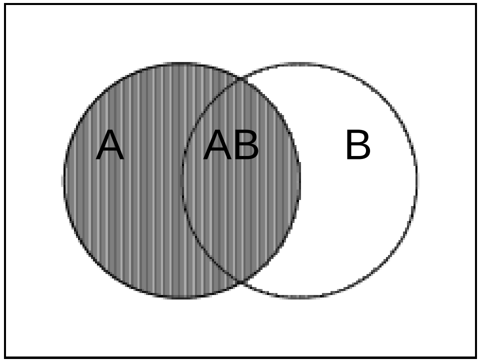
$X = \overline{\overline{A}} = A$

2.4 Basic Theorems

Rule 10: A + AB = *A*

A(1 + B) = *A*

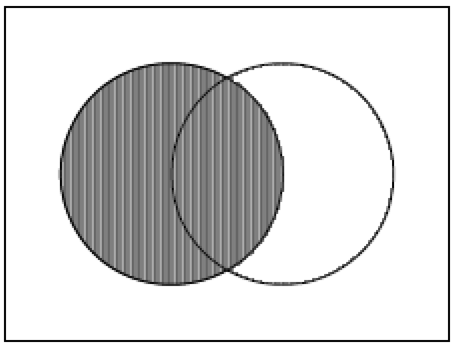
A · 1 = A



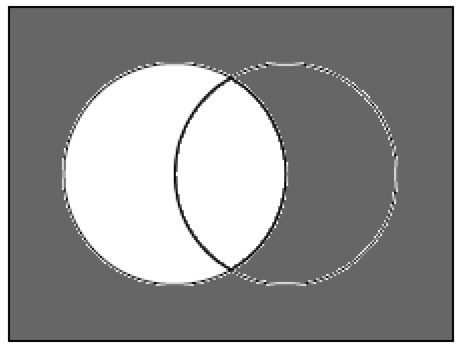
A + AB = A

2.4 Basic Theorems

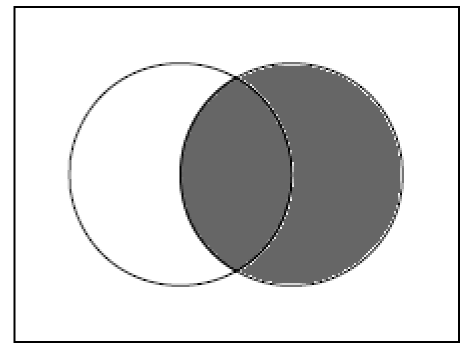
Rule 11: $A + \bar{A}B = A + B$



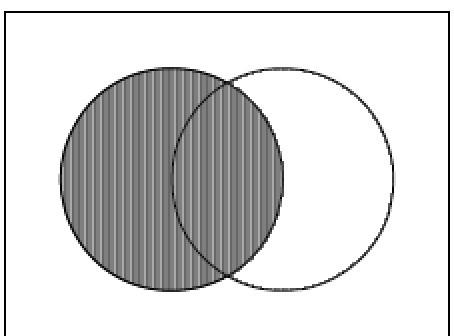
A



$\bar{A}$

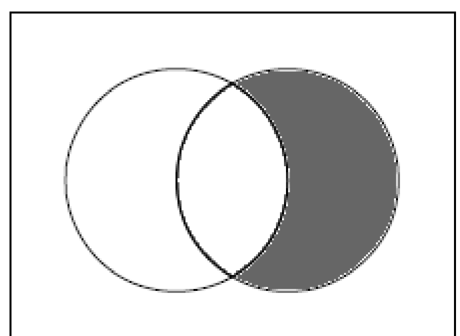


B



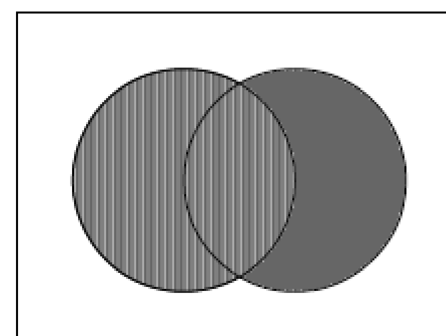
A

+



$\bar{A}B$

=



A + B

2.4 Basic Theorems

Rule 12: $(A + B)(A + C) = A + BC$

$AA + AC + AB + BC$

$A(1 + C + B) + BC$

$A + BC$

2.5 Boolean Functions

2.2 Simplify

(a) $xy + x\bar{y}$

In general, group similar terms

$= x(y + \bar{y})$

$= x \cdot 1 = x$

$AA = A$

$A + A = A$

$A \cdot 0 = 0$

$A \cdot 1 = A$

$A\bar{A} = 0$

$A + \bar{A} = 1$

$1 + A = 1$

2.5 Boolean Functions

(b) $(x+y)(x+\bar{y})$

In general, factor terms into a sum-of-product (SOP) expression

$$= \cancelto{x}{xx} + x\bar{y} + xy + \cancelto{0}{y\bar{y}}$$

$$= x + x\bar{y} + xy$$

$$= x(1 + \bar{y} + y)$$

$$= x$$

2.5 Boolean Functions

(c) $xyz + \bar{x}y + xy\bar{z}$

$= xy\cancel{(z+\bar{z})} + \bar{x}y$

$= xy + \bar{x}y$

$= y\cancel{(x+\bar{x})}$

$= y$

2.5 Boolean Functions

2.2(e)

$$(A+B+\bar{C})(\bar{A}\bar{B}+C)$$

$$= \cancelto{0}{A\bar{A}\bar{B}} + AC + \cancelto{0}{BA\bar{B}} + CB + \bar{A}\bar{B}\bar{C} + \cancelto{0}{\bar{C}C}$$

$$= AC + BC + \bar{A}\,\bar{B}\,\bar{C}$$ $\Rightarrow$ sum-of-products form

2.5 Boolean Functions

(F) $\bar{A}BC + AB\bar{C} + ABC + \bar{A}B\bar{C}$

Best to combine that change by one variable

$= BC\,\cancel{(A+\bar{A})}^{1} + B\bar{C}\,\cancel{(A+\bar{A})}^{1}$

$= BC + B\bar{C}$

$= B\,\cancel{(C+\bar{C})}^{1}$

$= B$

2. 4(e) Reduce to two literals

$$ABC'D + A'BD + ABCD$$

$$ABD(C' + C) + A'BD$$

$$ABD + A'BD$$

$$BD(A + A')$$

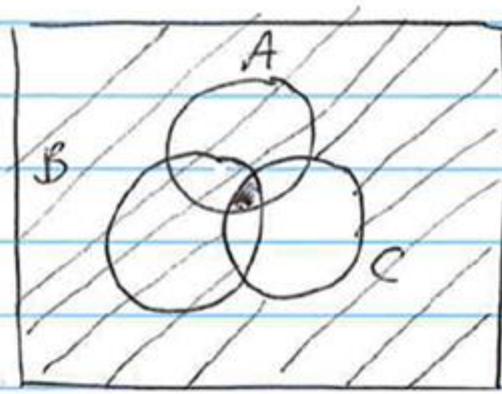
$$BD$$

2.4 Reduce the Boolean expressions to:

(a) three literals $\bar{A}\bar{C} + ABC + A\bar{C}$

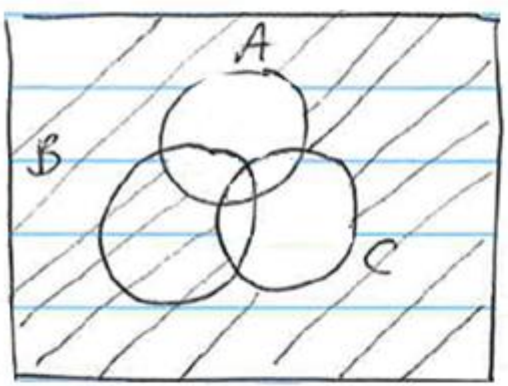
$= \bar{C}(\bar{A}+A) + ABC$

$= \bar{C} + ABC$ ✓



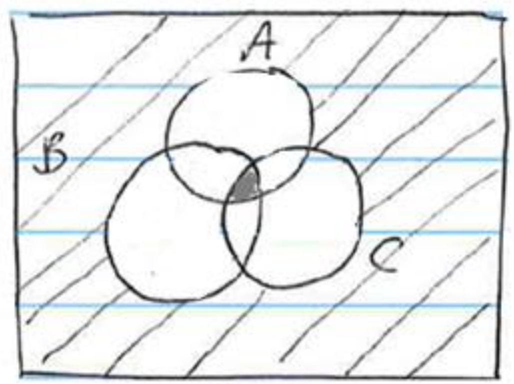
$= C' + AB$

$=$



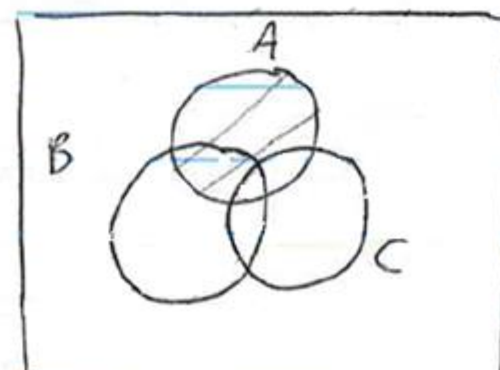
$\bar{A}\bar{C}$

$+$



ABC

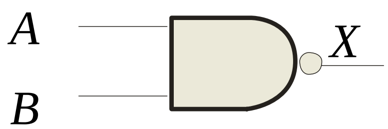
$+$



$A\bar{C}$

2.5 Boolean Functions
The NAND function
(complement of AND function)

Inputs		Output
A	*B*	*X*
0	0	1
0	1	1
1	0	1
1	1	0

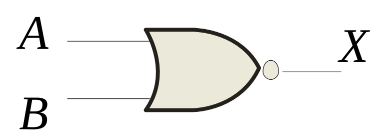


The **NAND** operation is shown with a dot between the variables and a bar covering them. Thus, the NAND operation is written as $X = \overline{AB}$.

2.5 Boolean Functions
The NOR function

(complement of OR function)

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	0

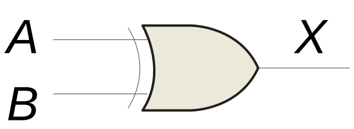


The **NOR** operation is shown with a OR sign between the variables and an over-bar covering them. Thus, the NOR operation is written as $X = \overline{A + B}$.

2.5 Boolean Functions
The XOR function

The **exclusive-OR gate** produces a HIGH output only when both inputs are at opposite logic levels. The truth table is

Inputs		Output
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0



The **XOR** operation is written as $X = A\overline{B} + \overline{A}B$. Alternatively, it can be written with a circled OR sign between the variables as

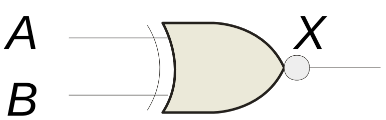
$X = A \oplus B$

2.5 Boolean Functions
The XNOR function

(complement of XOR function)

The **exclusive-NOR gate** produces a HIGH output only when both inputs are at opposite logic levels. The truth table is

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

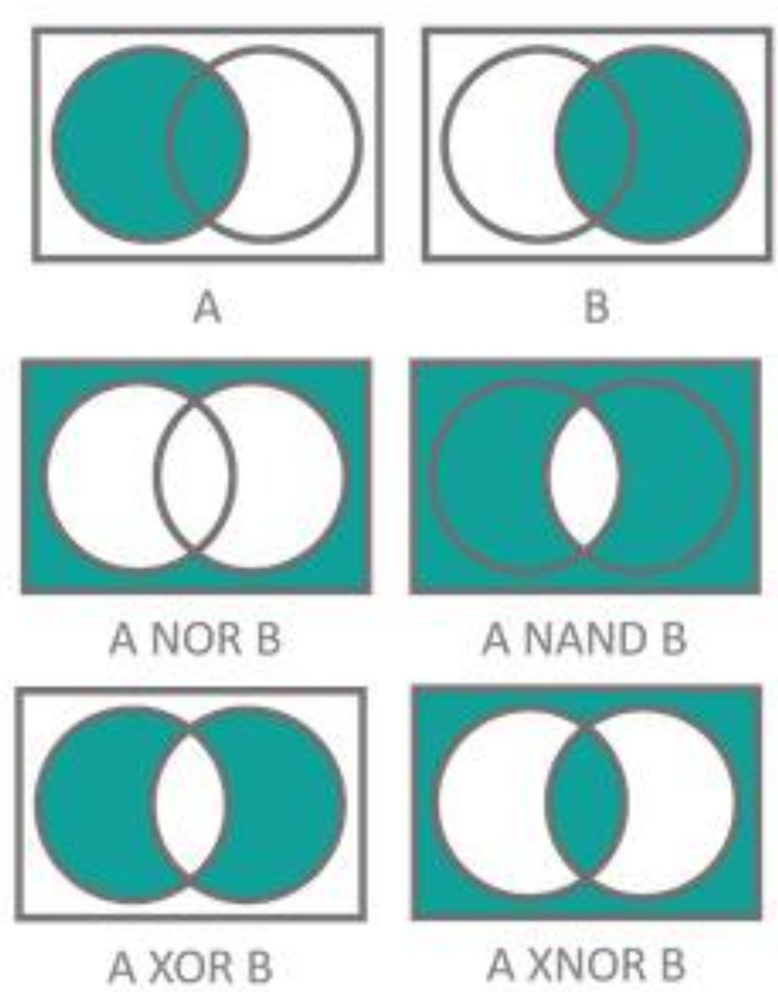


The **XNOR** operation is written as $X = \overline{A}\,\overline{B} + AB$. Alternatively, it can be written with a circled OR sign between the variables as

$$X = A \odot B$$

2.5 Boolean Functions

Venn Diagrams of Some Basic Functions



4-Input AND Functions