Boolean Algebra

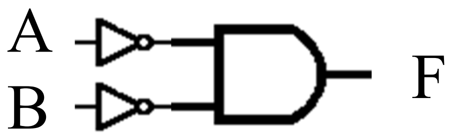
- Boolean Algebra
- Basic Axioms (Huntington's Postulates)
- Duality Principle
- Fundamental Theorems

Boolean Algebra

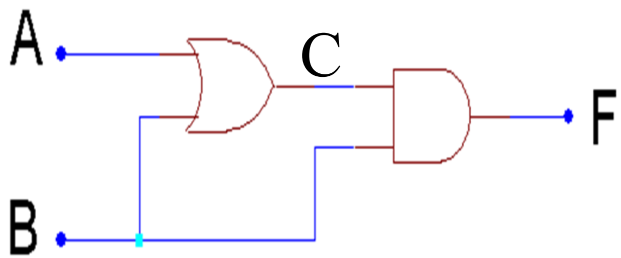
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	
0	1	
1	0	
1	1	



A	B	$\overline{A}$	$\overline{B}$	F
0	0			
0	1			
1	0			
1	1			



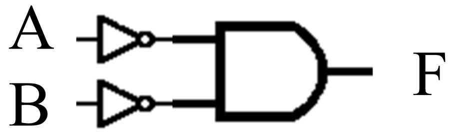
A	B	C	F
0	0		
0	1		
1	0		
1	1		

Boolean Algebra

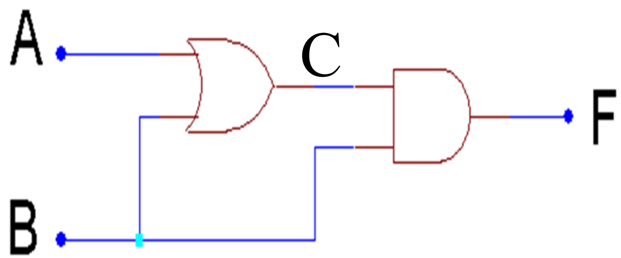
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0			
0	1			
1	0			
1	1			



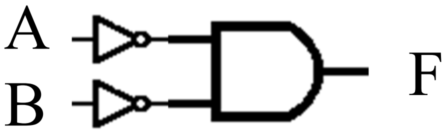
A	B	C	F
0	0		
0	1		
1	0		
1	1		

Boolean Algebra

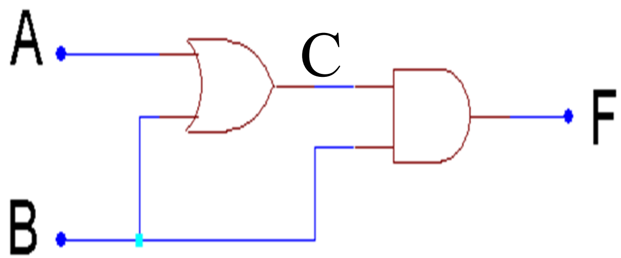
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0	1		
0	1	1		
1	0	0		
1	1	0		



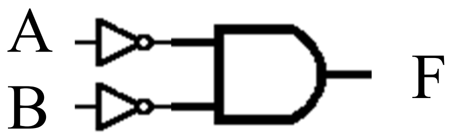
A	B	C	F
0	0		
0	1		
1	0		
1	1		

Boolean Algebra

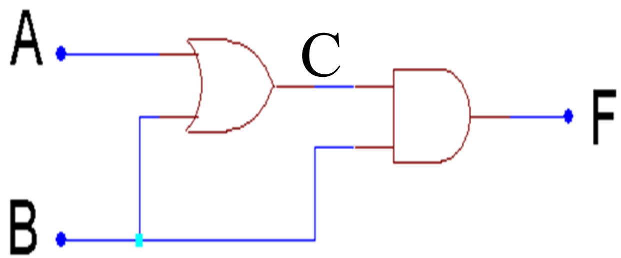
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0	1	1	
0	1	1	0	
1	0	0	1	
1	1	0	0	



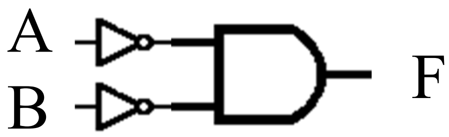
A	B	C	F
0	0		
0	1		
1	0		
1	1		

Boolean Algebra

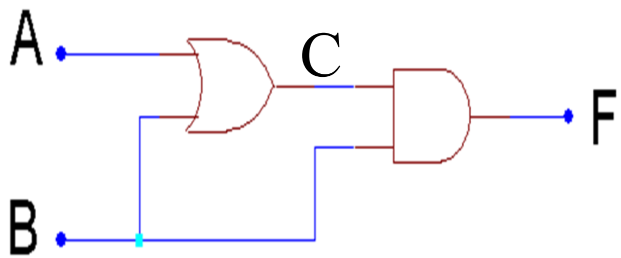
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0



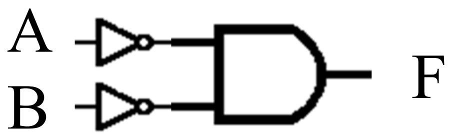
A	B	C	F
0	0		
0	1		
1	0		
1	1		

Boolean Algebra

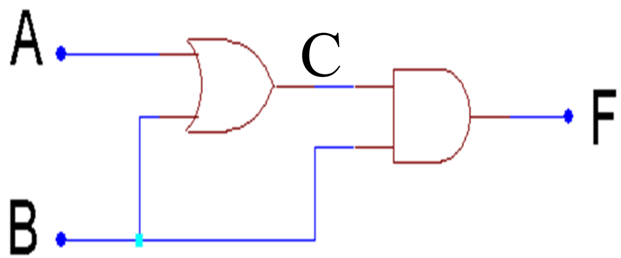
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0



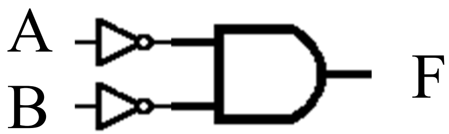
A	B	C	F
0	0	0	
0	1	1	
1	0	1	
1	1	1	

Boolean Algebra

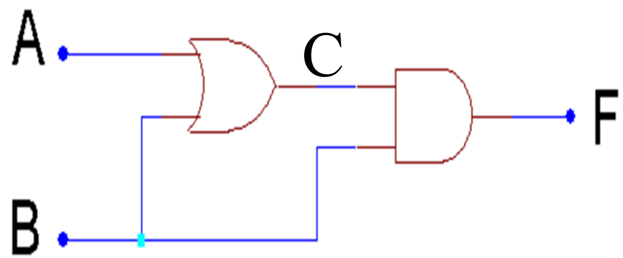
Boolean Algebra is the mathematical basis for digital systems. It allows us to manipulate logic expressions resulting in simpler circuits and to produce circuits with alternative gate representations.

Complete the truth tables for the following:

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



A	B	$\overline{A}$	$\overline{B}$	F
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0



A	B	C	F
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1

Boolean Algebra

Originates from George Boole, 1854, "An investigation of the laws of thought". A formal way to describe logic statements.

Based on propositions that can be **TRUE** or **FALSE**.

Switching Algebra

Switching algebra, based on the more general Boolean Algebra, is the arithmetic of a two state system. Developed in 1938 by Claude Shannon.

The condition of a switch which can be either open or closed is represented by a variable, say X, and can take one of two values **0** or **1**.

The state of a logic signal is represented by a variable that can be in one of two conditions, **0** or **1**.

The algebra consists of a set of postulates and derived theorems.

Axioms - Huntington's Postulates

- From work of E.V.Huntington (1904)

1. Closure.

If X and Y take only the values {0,1},

then (X + Y) takes only the values {0,1}.

If X and Y take only the values {0,1},

then (X.Y) takes only the values {0,1}.

Axioms and Duality Principle

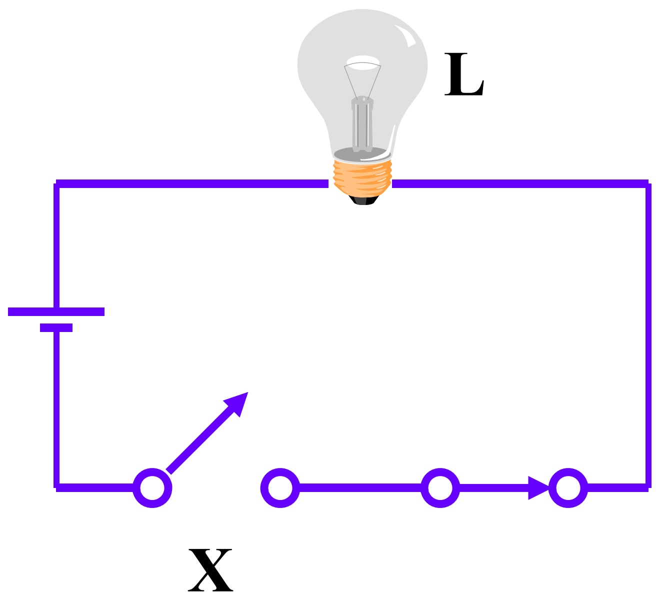
2. Identity Properties	3. Commutative Properties
$X + 0 = X$	$X + Y = Y + X$
$X.1 = X$	$X.Y = Y.X$

4. Distributive Properties	5. Complement Properties
$X + (Y.Z) = (X + Y).(X + Z)$	$X + \overline{X} = 1$
$X.(Y + Z) = X.Y + X.Z$	$X.\overline{X} = 0$

These postulates provide the basis for the entire switching algebra.

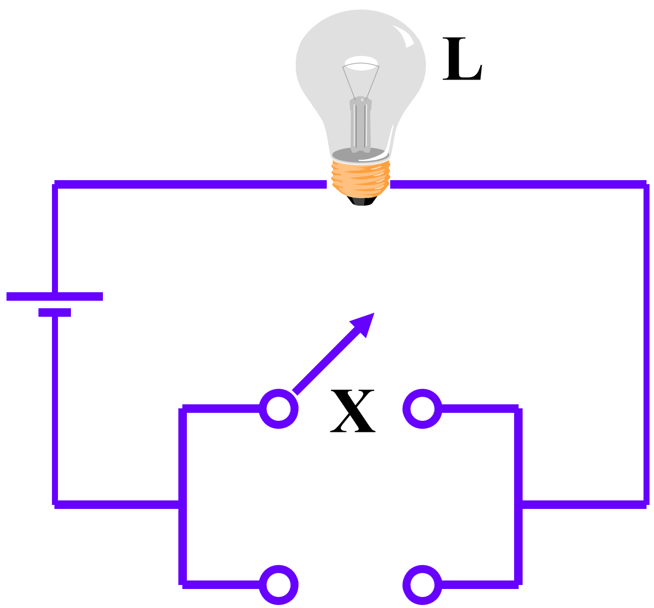
Switch Representations

Switch circuits can be used to visualize some of the axioms. Switches in series represent an **AND** function. So **X.1** can be represented with variable **X** indicating the state of the first switch and the second switch in the **1** or **CLOSED** state.



The output of the circuit depends only upon the state of **X** illustrating that **X.1 = X**

Switches in parallel represent an OR function. So **X + 0** can be represented with variable **X** indicating the state of the top switch and the bottom switch in the **0** or **OPEN** state.



The output of the circuit depends only upon the state of X illustrating that X + 0 = X

It is not really necessary to draw in the entire circuit, but just the switches to illustrate if a circuit path is being completed.

Duality Principle

Each of the axioms is presented in pairs. Each pair has the following property:

One of the postulates in each pair can be obtained by interchanging the (+) and (.) operators and interchanging the identity elements (0) and (1).

Principle of Duality: If a Boolean statement is true then the dual of the statement is true.

Because this principle is true for all axioms, it applicable to all theorems.

Fundamental Theorems 1

Theorems can be derived from the postulates and duality principle

Theorem1. Null Law	Theorem2. Involution
$X + 1 = 1$	$\overline{\overline{X}} = X$
$X.0 = 0$	

Fundamental Theorems 2

Theorem3. Idempotency	Theorem4. Absorption
$X + X = X$	$X + X.Y = X$
$X.X = X$	$X.(X + Y) = X$

Fundamental Theorems 3

Theorem 5. Simplification

$X + \overline{X}.Y = X + Y$

$X.(\overline{X} + Y) = X.Y$

Theorem 6. Associativity

$X + (Y + Z) = (X + Y) + Z$

$X.(Y.Z) = (X.Y).Z$

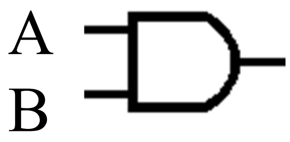
Theorem 7. Consensus

$X.Y + \overline{X}.Z + Y.Z = X.Y + \overline{X}.Z$

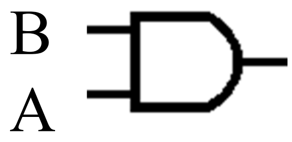
$(X + Y).(\overline{X} + Z).(Y + Z) = (X + Y).(\overline{X} + Z)$

Logic Gates

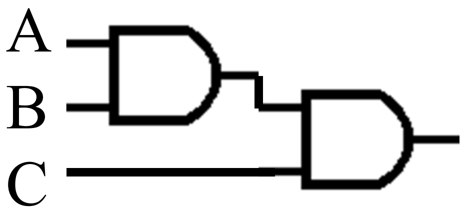
In terms of logic gates, the **commutative** law is telling us that

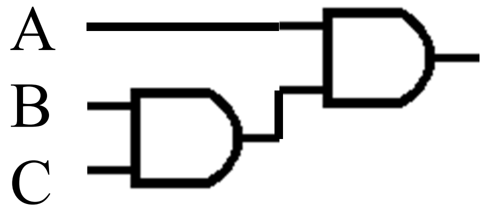


is equivalent to

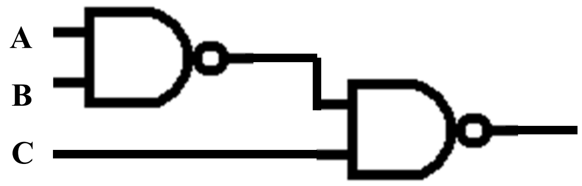


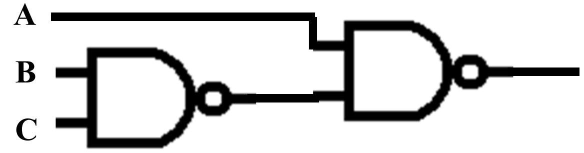
The **associative** law is telling us that the circuits below are equivalent.



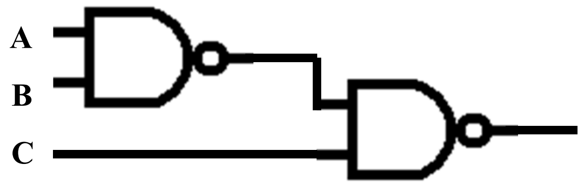


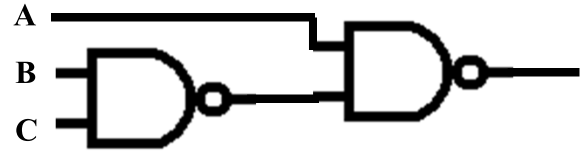
Is NAND associative ?





Is NAND associative ?

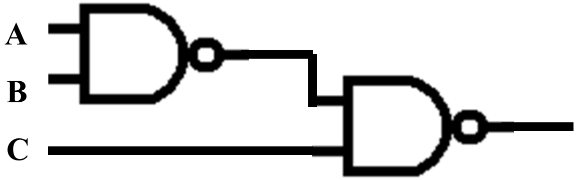


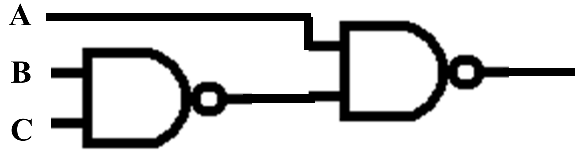


A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$

Is NAND associative ?

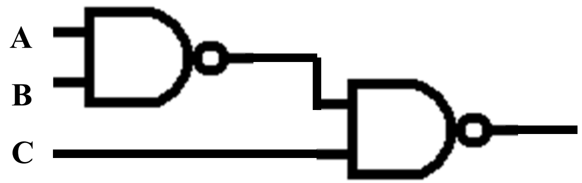
Remember : 0 drives a NAND gate to 1

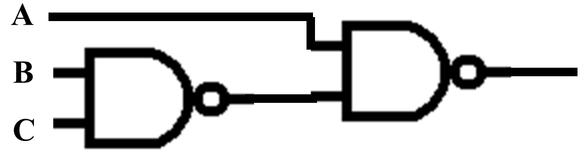




A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$
0 0 0				
0 0 1				
0 1 0				
0 1 1				
1 0 0				
1 0 1				
1 1 0				
1 1 1				

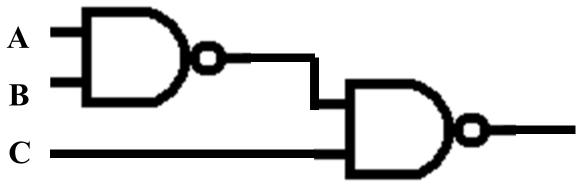
Is NAND associative ?

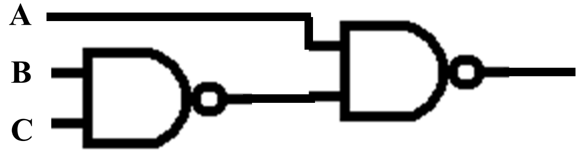




A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$
0 0 0	1			
0 0 1	1			
0 1 0	1			
0 1 1	1			
1 0 0	1			
1 0 1	1			
1 1 0	0			
1 1 1	0			

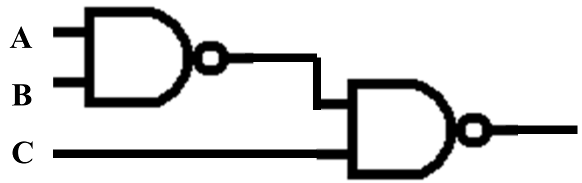
Is NAND associative ?

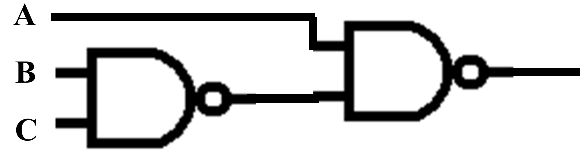




A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$
0 0 0	1	1		
0 0 1	1	0		
0 1 0	1	1		
0 1 1	1	0		
1 0 0	1	1		
1 0 1	1	0		
1 1 0	0	1		
1 1 1	0	1		

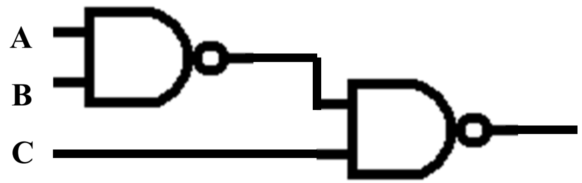
Is NAND associative ?

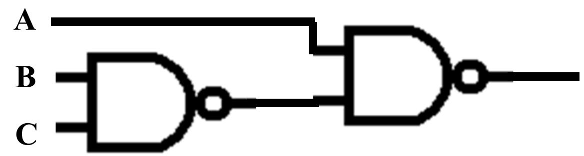




A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$
0 0 0	1	1	1	
0 0 1	1	0	1	
0 1 0	1	1	1	
0 1 1	1	0	0	
1 0 0	1	1	1	
1 0 1	1	0	1	
1 1 0	0	1	1	
1 1 1	0	1	0	

Is NAND associative ?



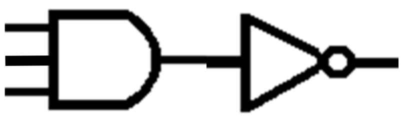


A B C	$\overline{A.B}$	$\overline{\overline{A.B}.C}$	$\overline{B.C}$	$\overline{A.\overline{B.C}}$
0 0 0	1	1	1	1
0 0 1	1	0	1	1
0 1 0	1	1	1	1
0 1 1	1	0	0	1
1 0 0	1	1	1	0
1 0 1	1	0	1	0
1 1 0	0	1	1	0
1 1 1	0	1	0	1

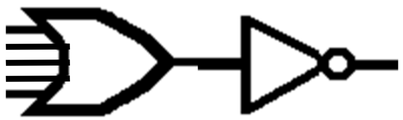
Multiple Input Gates

The **AND** and **OR** functions can be extended to three or more inputs as they are commutative and associative.

The **NAND** and **NOR** functions are commutative but not associative. The gates themselves can be extended to have more than two inputs by defining them as the complemented form of **AND** and **OR** respectively.



NAND



NOR

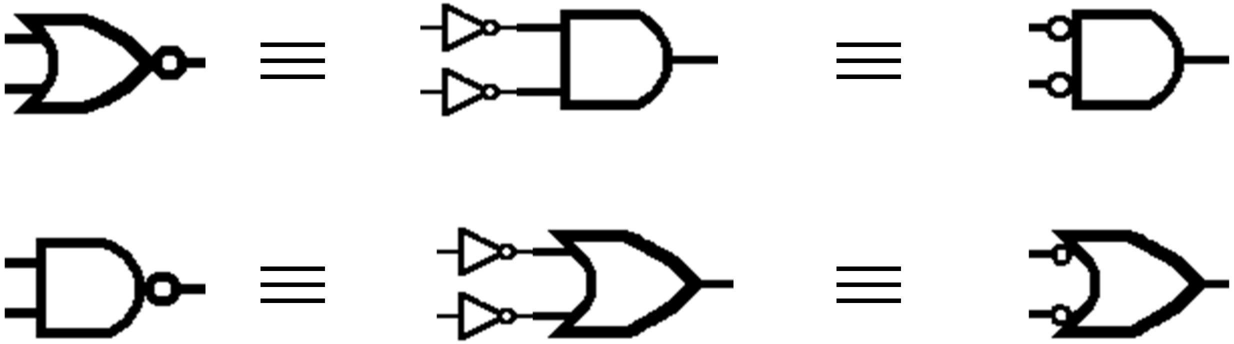
De Morgan’s Law

Theorem 8. De Morgan’s Laws

$$\overline{(X + Y)} = \overline{X} . \overline{Y}$$

$$\overline{(X . Y)} = \overline{X} + \overline{Y}$$

Used to rewrite expressions to give alternative gate implementations.



Generalisation of De Morgan's Law

Does De Morgan hold for more than two variables ?

Consider the expression $\overline{A + B + C}$

Rename A + B as D, which gives $\overline{D + C}$

De Morgan for two variables gives $\overline{D}.\overline{C}$

But $D = A + B$

By De Morgan again $\overline{D} = \overline{A + B} = \overline{A}.\overline{B}$

Hence $\overline{A + B + C} = \overline{A}.\overline{B}.\overline{C}$ and by duality $\overline{A.B.C} = \overline{A} + \overline{B} + \overline{C}$

The same principle can be applied for any number of variables.

Examples of De Morgan

Simplify the expression:

$$\overline{\overline{A + B.\overline{C}} + \overline{D.(E + \overline{F})}} \quad = \quad \overline{\overline{A + B.\overline{C}}} \;\mathbf{.}\; \overline{D.\overline{(E + \overline{F})}}$$

$$= \quad (A + B.\overline{C}) \;\mathbf{.}\; \overline{D.\overline{(E + \overline{F})}}$$

$$= \quad (A + B.\overline{C}) \;\mathbf{.}\; (\overline{D} + \overline{\overline{(E + \overline{F})}})$$

$$= \quad (A + B.\overline{C}) \;\mathbf{.}\; (\overline{D} + (E + \overline{F}))$$

$$= \quad (A + B.\overline{C}) \;\mathbf{.}\; (\overline{D} + E + \overline{F})$$

Examples of De Morgan

Simplify the expression:

$$\overline{(\overline{A + B.\overline{C}}) + (D.\overline{(E + \overline{F})})} = \overline{\overline{A + B.\overline{C}}} \,.\, \overline{D.\overline{(E + \overline{F})}}$$

$$= (A + B.\overline{C}) \,.\, \overline{D.\overline{(E + \overline{F})}}$$

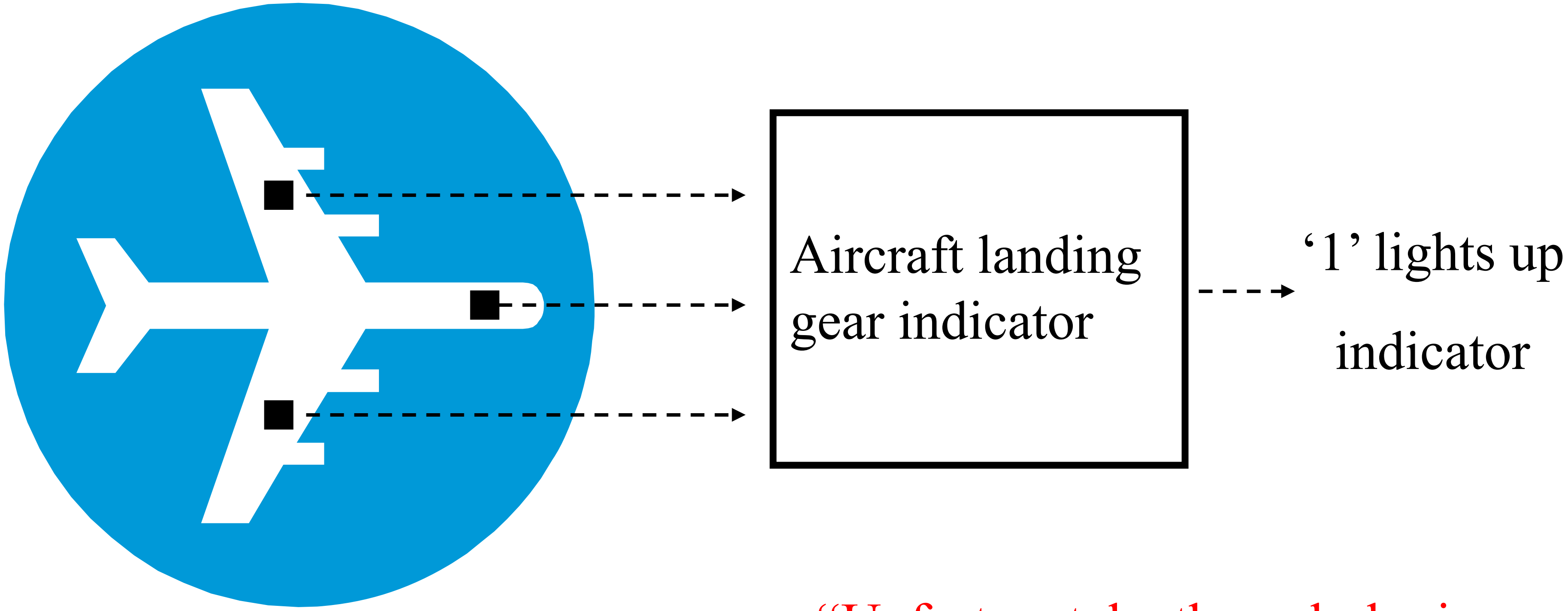
$$= (A + B.\overline{C}) \,.\, (\overline{D} + \overline{\overline{(E + \overline{F})}})$$

$$= (A + B.\overline{C}) \,.\, (\overline{D} + (E + \overline{F}))$$

$$= (A + B.\overline{C}) \,.\, (\overline{D} + E + \overline{F})$$

Design Problem

"An indicator is required for an aircraft landing system that lights up when all three sets of landing wheels are fully down."



Sensors on wheels give a '1' when wheels are fully down.

"Unfortunately, the only logic devices available to you are OR gates and INVERTERS."

Design Solution

Assign variables for each set of wheels.

Wheels1 = A, '1' = fully down

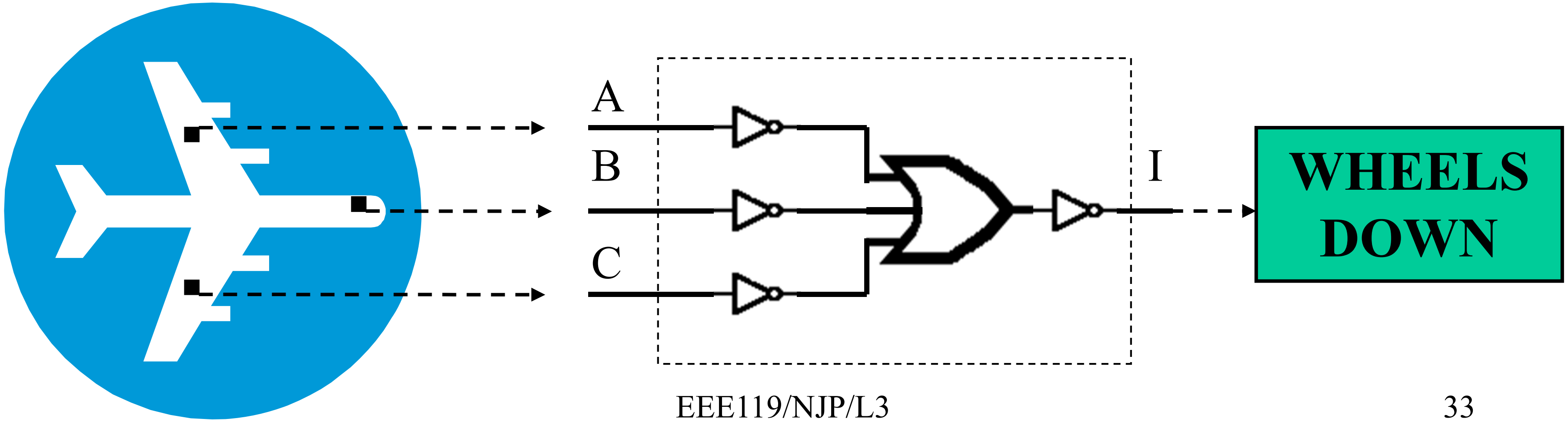
indicator I, '1' = lit

Wheels2 = B, '1' = fully down

Wheels3 = C, '1' = fully down

$I = A.B.C = \overline{\overline{A.B.C}}$ (involution theorem $\overline{\overline{X}} = X$)

$I = \overline{\overline{A} + \overline{B} + \overline{C}}$ (De Morgan for 3 variables)



Summary

- The theorems of Boolean Algebra are derived from a set of basic postulates
- Boolean Algebra can be used to manipulate logic expressions
- De Morgan's Laws enable us to break inversions and simplify logic expressions