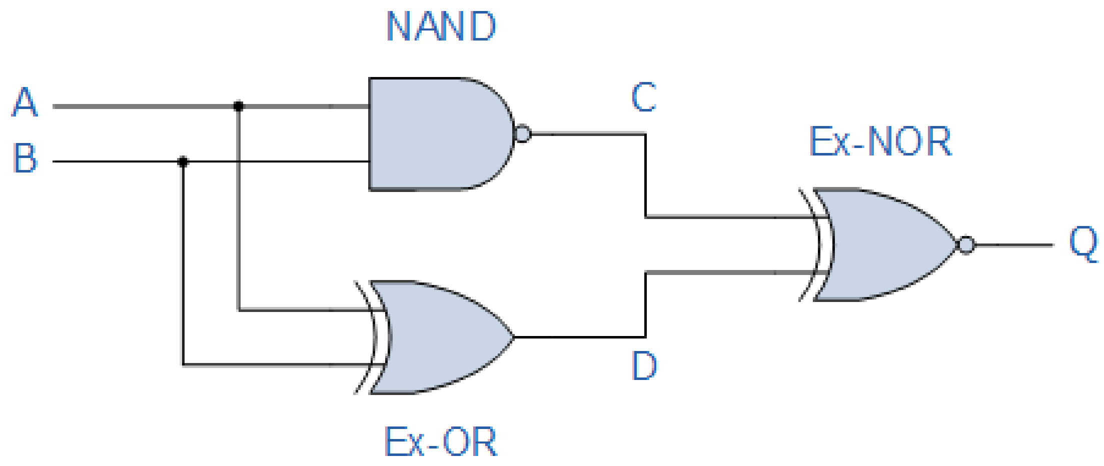
Discrete Mathematics 5
Boolean Algebra II

Dr. Maged Kassab

Boolean Algebra Simplification (Examples)

Boolean Algebra Examples No1

Construct a Truth Table for the logical functions at points C, D and Q in the following circuit and identify a single logic gate that can be used to replace the whole circuit.



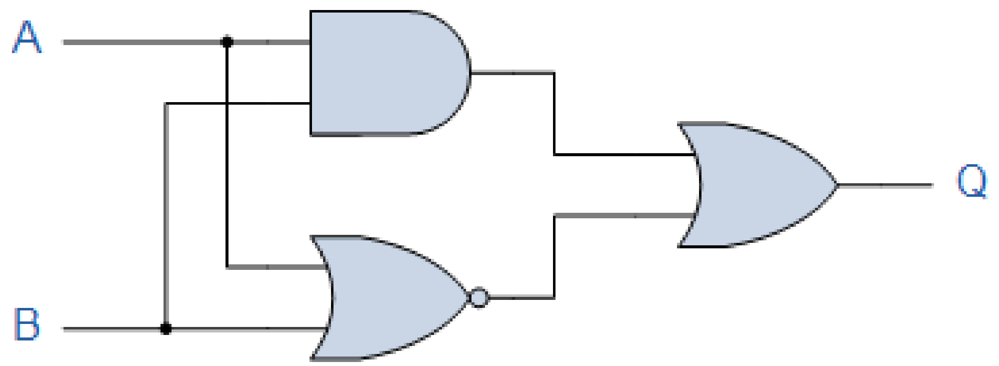
Boolean Algebra Simplification (Examples)

Inputs		Output at		
A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

Boolean Algebra Simplification (Examples)

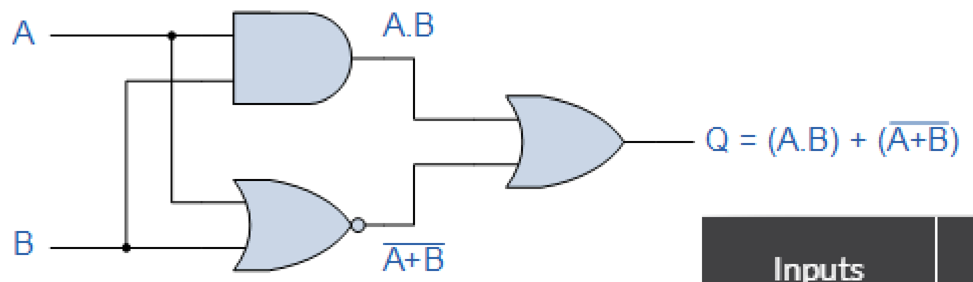
Boolean Algebra Examples No2

Find the Boolean algebra expression for the following system.



Boolean Algebra Simplification (Examples)

output expression is given as:

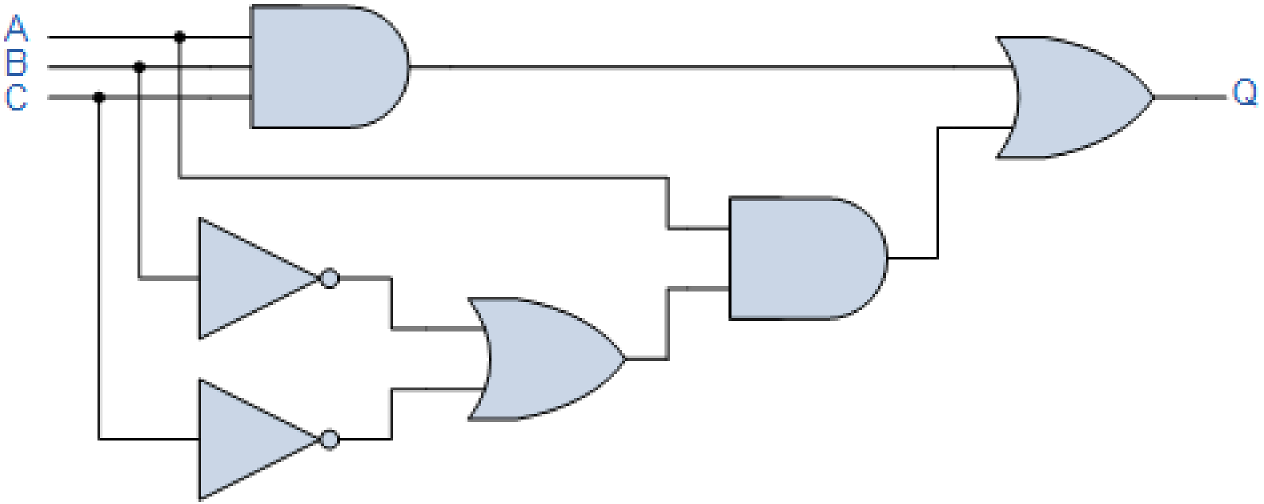


Inputs		Intermediates		Output
B	A	A.B	$\overline{A+B}$	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

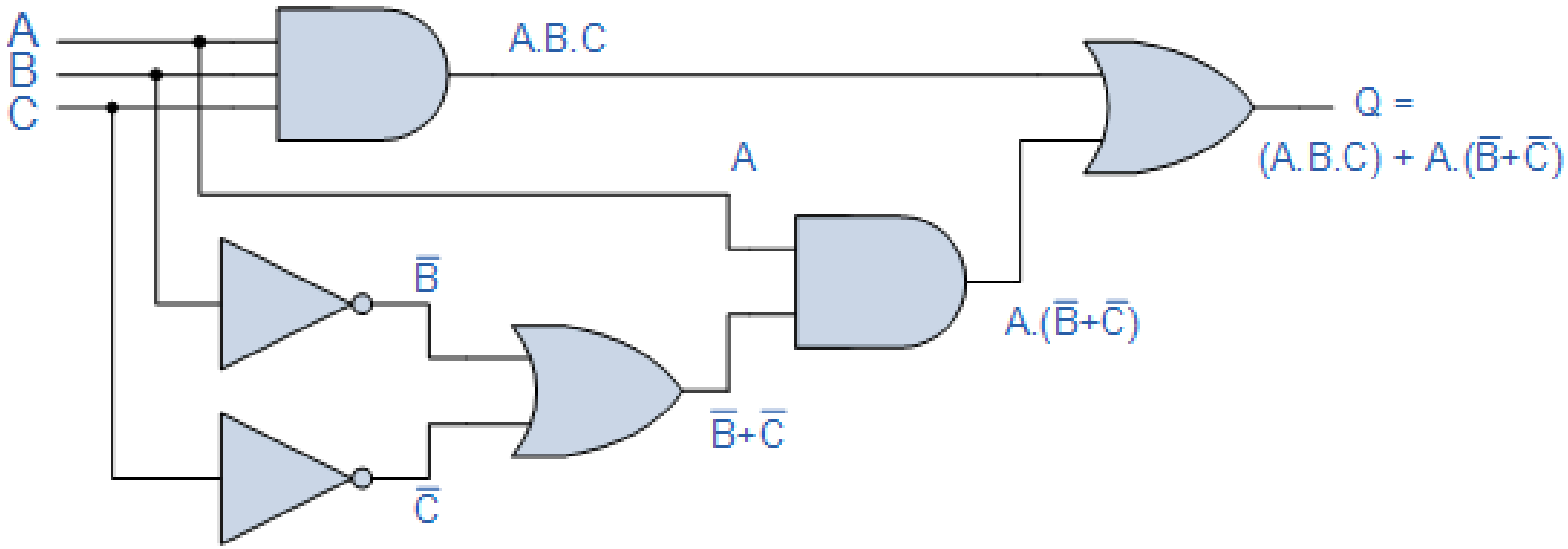
Boolean Algebra Simplification (Examples)

Example No3

Find the Boolean algebra expression for the following system.



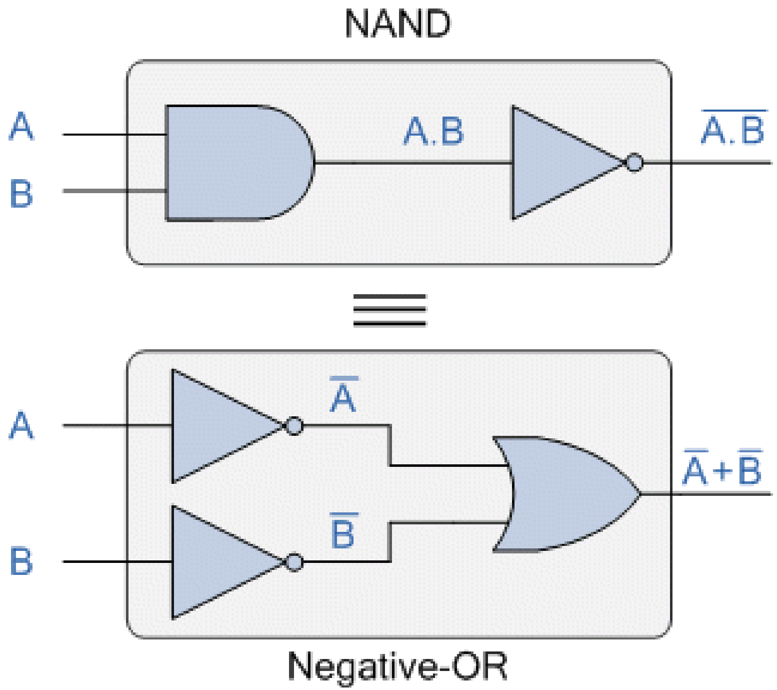
Boolean Algebra Simplification (Examples)



Boolean Algebra Simplification (Examples)

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

DeMorgan’s Theorem



DeMorgan’s Theorem

DeMorgan´s Theorem and Laws can be used to to find the equivalency of the NAND and NOR gates

DeMorgan’s Theorem

Truth Table for Each Logical Operation

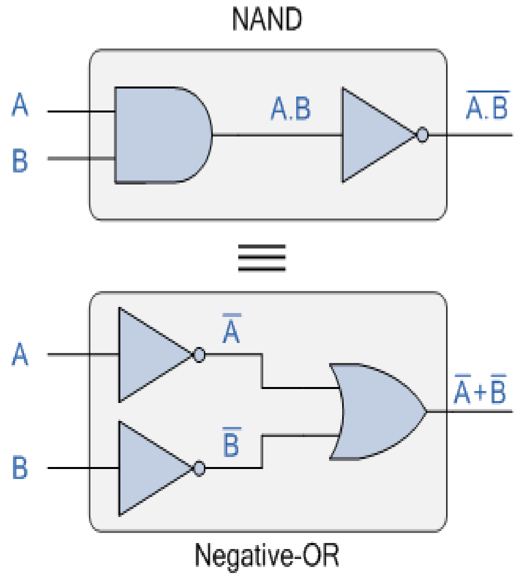
Logic Function	Boolean Notation
AND	A.B
OR	A+B
NOT	$\overline{A}$
NAND	$\overline{A.B}$
NOR	$\overline{A+B}$

Input Variable		Output Conditions			
A	B	AND	NAND	OR	NOR
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	1	0

DeMorgan's Theorem

DeMorgan's First Theorem

DeMorgan's First theorem proves that when two (or more) input variables are AND'ed and negated, they are equivalent to the OR of the complements of the individual variables. Thus the equivalent of the NAND function will be a negative-OR function, proving that $\boxed{\overline{A.B} = \overline{A}+\overline{B}}$. We can show this operation using the following table.



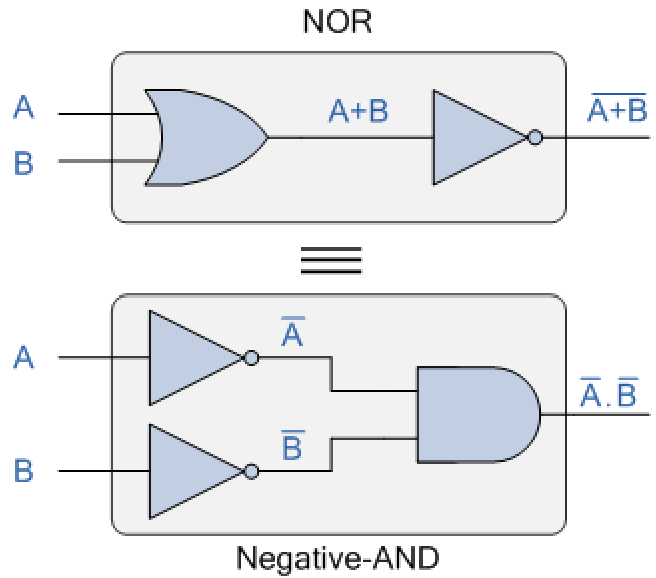
DeMorgan’s Theorem

Inputs		Truth Table Outputs For Each Term				
B	A	A.B	$\overline{A.B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	1	1	0	1
1	1	1	0	0	0	0

DeMorgan’s Theorem

DeMorgan's Second Theorem

DeMorgan's Second theorem proves that when two (or more) input variables are OR'ed and negated, they are equivalent to the AND of the complements of the individual variables. Thus the equivalent of the NOR function is a negative-AND function proving that $\boxed{\overline{A+B} = \overline{A}.\overline{B}}$, and again we can show operation this using the following truth table.



DeMorgan’s Theorem

Inputs		Truth Table Outputs For Each Term				
B	A	A+B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}\,.\,\overline{B}$
0	0	0	1	1	1	1
0	1	1	0	0	1	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

DeMorgan’s Theorem

For a 3-variable input

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

and also

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

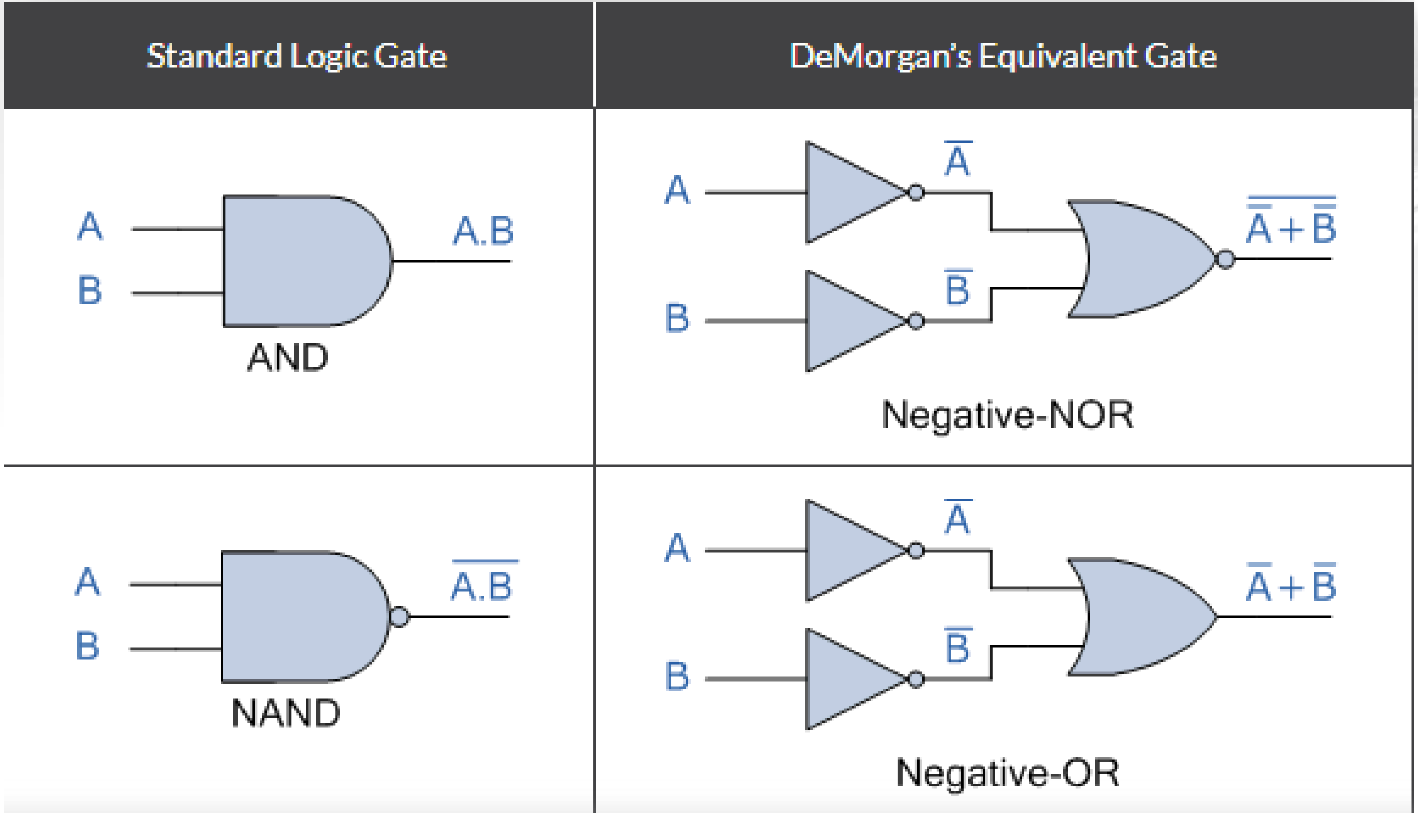
For a 4-variable input

$$\overline{A.B.C.D} = \overline{A}+\overline{B}+\overline{C}+\overline{D}$$

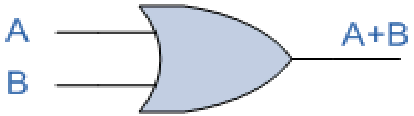
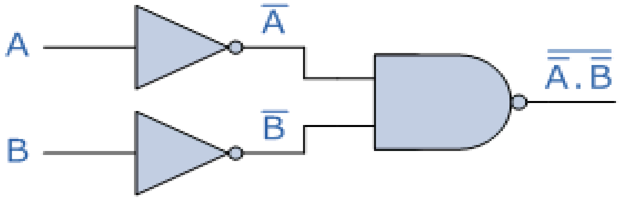
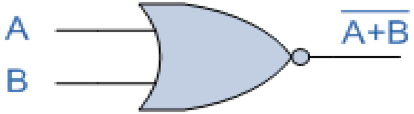
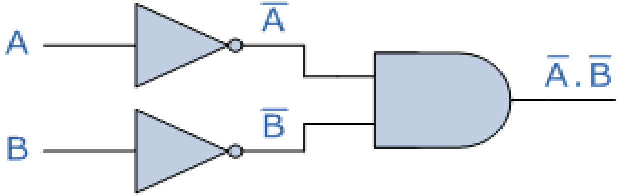
and also

$$\overline{A+B+C+D} = \overline{A}.\overline{B}.\overline{C}.\overline{D}$$

DeMorgan’s Equivalent Gates



DeMorgan’s Equivalent Gates

Standard Logic Gate	DeMorgan’s Equivalent Gate
 OR	 Negative-NAND
 NOR	 Negative-AND

Boolean Algebra Simplification

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Boolean Algebra Simplification

Boolean Algebra Simplification and how to simplify Boolean algebra expressions using some basic rules applied to their variables, literals and terms

Boolean Algebra Simplification

Boolean Algebra Simplification Table

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\overline{B}$

Boolean Algebra Simplification

Boolean Algebra Simplification

Having established the switching operation of the **AND**, **OR**, and **NOT** functions. We can now look at simplifying some basic Boolean Algebra expressions to obtain a final expression that has the minimum number of terms.

First let us start with something simple such as:

The Boolean Expression: $A.(A + B)$

Multiplying out the brackets gives us:

Boolean Algebra Simplification

The Boolean Expression: A.(A + B)

Multiplying out the brackets gives us:

	A.(A+B)	Start
multiply:	A.A + A.B	Distributive Law
but:	A.A = A	Idempotent Law
then:	A + A.B	Reduction
thus:	A.(1 + B)	Annulment Law
equals to:	A	Absorption Law

Then we can see that the Boolean expression of A.(A + B) can be reduced to just "A" which follows Boole's *Absorption Law*.

Boolean Algebra Simplification

Example No2

This time we will use three Boolean terms, A, B, and C and use the same Boolean algebra simplification rules as before.

Boolean Expression: $(A + B)(A + C)$

Again, multiplying out the brackets gives us:

Boolean Algebra Simplification

	(A + B)(A + C)	Start
multiply:	A.A + A.C + A.B + B.C	Distributive law
but:	A.A = A	Idempotent Law
then:	A + A.C + A.B + B.C	Reduction
however:	A + A.C = A	Absorption Law
thus:	A + A.B + B.C	Distributive Law
again:	A + A.B = A	Absorption Law
thus:	A + B.C	Result

Then the Boolean expression of (A + B)(A + C) can be reduced to just "A + B.C" using the various Boolean algebra laws.

Boolean Algebra Simplification

Boolean Algebra Simplification Example No3

This time we will again use the same three Boolean terms of A, B and C but introduce a NOT function to one of the terms.

Boolean Expression: $AB(\overline{B}C + AC)$

Boolean Algebra Simplification

	$AB(\overline{B}C+AC)$	Start
multiply	$A.B.\overline{B}.C + A.B.A.C$	Distributive Law
again:	$A.A = A$	Idempotent Law
then:	$A.B.\overline{B}.C + A.B.C$	Reduction
but:	$B.\overline{B} = 0$	Complement Law
so:	$A.0.C + A.B.C$	Reduction
becomes:	$0 + A.B.C$	Reduction
as:	$0 + A.B.C = A.B.C$	Identity Law
thus:	ABC	Result

Then the Boolean expression of $AB(\overline{B}C+AC)$ is reduced to "ABC".

Boolean Algebra Simplification

Simplify the Boolean function:
$(A + B)(A + \bar{B})(\bar{A} + C)$

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

After simplification we get

$Y = [(A)(A) + (A)(\bar{B}) + (A)(B) + (B)(\bar{B})](\bar{A} + C)$

$Y = [A + (A)(\bar{B}) + (A)(B) + 0](\bar{A} + C)$

$Y = [A(1 + \bar{B}) + (A)(B)](\bar{A} + C)$

$Y = [A + (A)(B)](\bar{A} + C)$

$Y = [A(1 + B)](\bar{A} + C)$

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

$Y = A\bar{A} + AC$

$\mathbf{Y = AC}$

Boolean Algebra Simplification

- Simplify: $C + \overline{BC}$:

Expression	Rule(s) Used
$C + \overline{BC}$	Original Expression
$C + (\overline{B} + \overline{C})$	DeMorgan's Law.
$(C + \overline{C}) + \overline{B}$	Commutative, Associative Laws.
$T + \overline{B}$	Complement Law.
T	Identity Law.

Boolean Algebra Simplification

Simplify: $\overline{AB}(\overline{A}+B)(\overline{B}+B)$:

Expression	Rule(s) Used
$\overline{AB}(\overline{A}+B)(\overline{B}+B)$	Original Expression
$\overline{AB}(\overline{A}+B)$	Complement law, Identity law.
$(\overline{A}+\overline{B})(\overline{A}+B)$	DeMorgan's Law
$\overline{A}+\overline{B}B$	Distributive law. This step uses
$\overline{A}$	Complement, Identity.

Boolean Algebra Simplification

Simplify: $(A + C)(AD + A\overline{D}) + AC + C$:

Expression	Rule(s) Used
$(A + C)(AD + A\overline{D}) + AC + C$	Original Expression
$(A + C)A(D + \overline{D}) + AC + C$	Distributive.
$(A + C)A + AC + C$	Complement, Identity.
$A((A + C) + C) + C$	Commutative, Distributive.
$A(A + C) + C$	Associative, Idempotent.
$AA + AC + C$	Distributive.
$A + (A + T)C$	Idempotent, Identity, Distributive.
$A + C$	Identity, twice.

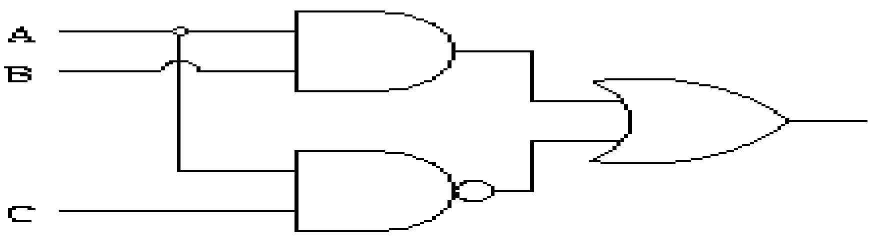
Boolean Algebra Simplification

Simplify: $\overline{A}(A+B)+(B+AA)(A+\overline{B})$:

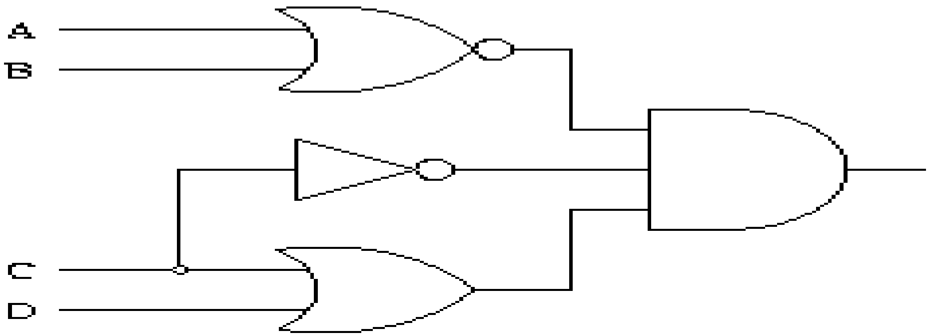
Expression	Rule(s) Used
$\overline{A}(A+B)+(B+AA)(A+\overline{B})$	Original Expression
$\overline{A}A+\overline{A}B+(B+A)A+(B+A)\overline{B}$	Idempotent (AA to A), then Distributive, used twice.
$\overline{A}B+(B+A)A+(B+A)\overline{B}$	Complement, then Identity. (Strictly speaking, we also
$\overline{A}B+BA+AA+B\overline{B}+A\overline{B}$	Distributive, two places.
$\overline{A}B+BA+A+A\overline{B}$	Idempotent (for the A's), then Complement and Identit
$\overline{A}B+AB+AT+A\overline{B}$	Commutative, Identity; setting up for the next step.
$\overline{A}B+A(B+T+\overline{B})$	Distributive.
$\overline{A}B+A$	Identity, twice (depending how you count it).
$A+\overline{A}B$	Commutative.
$(A+\overline{A})(A+B)$	Distributive.
$A+B$	Complement, Identity.

Boolean Algebra Simplification

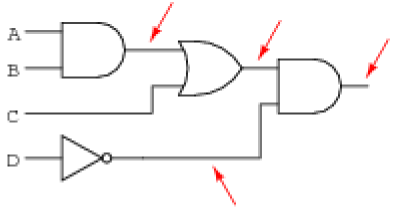
- Draw a logic circuit for $AB + \overline{AC}$.

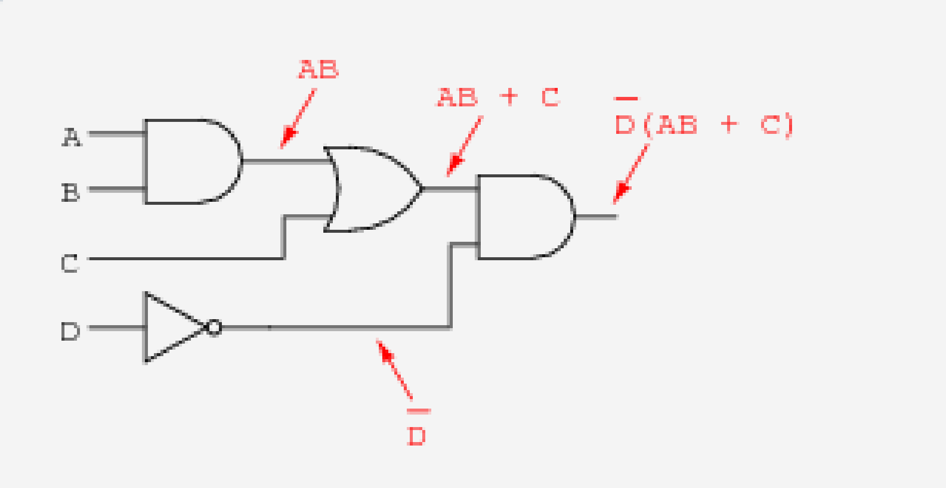


- Draw a logic circuit for $\overline{(A + B)}(C + D)\overline{C}$.



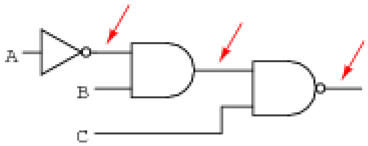
Convert the following logic gate circuit into a Boolean expression, writing Boolean sub-expressions next to each gate output in the diagram:

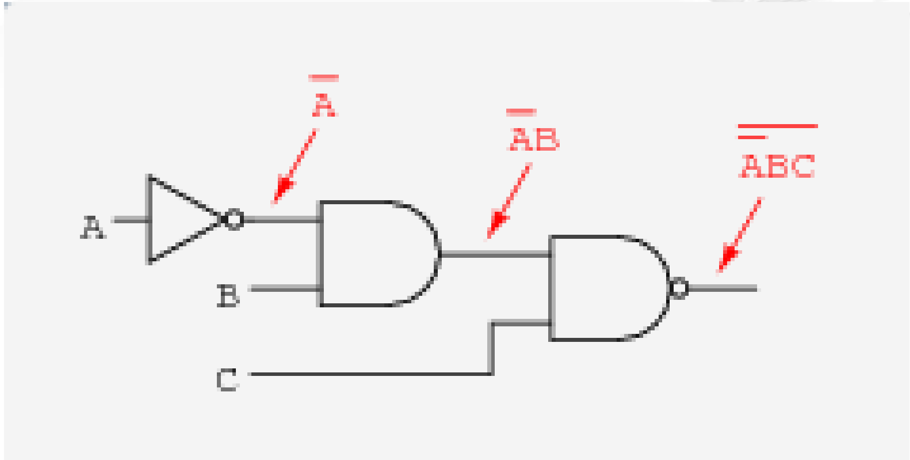




Boolean Algebra Simplification

Convert the following logic gate circuit into a Boolean expression, writing Boolean sub-expressions next to each gate output in the diagram:



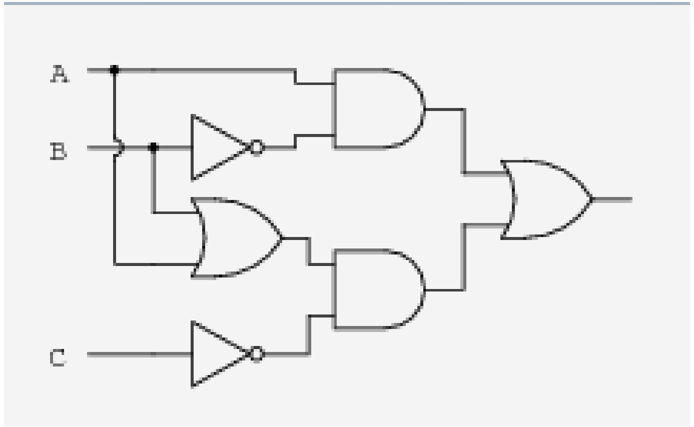


Boolean Algebra Simplification

An engineer hands you a piece of paper with the following Boolean expression on it, and tells you to build a gate circuit to perform that function:

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

Draw a logic gate circuit for this function.

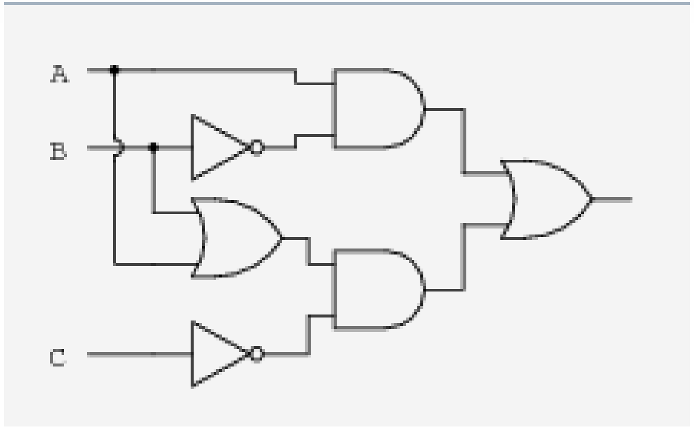


Boolean Algebra Simplification

An engineer hands you a piece of paper with the following Boolean expression on it, and tells you to build a gate circuit to perform that function:

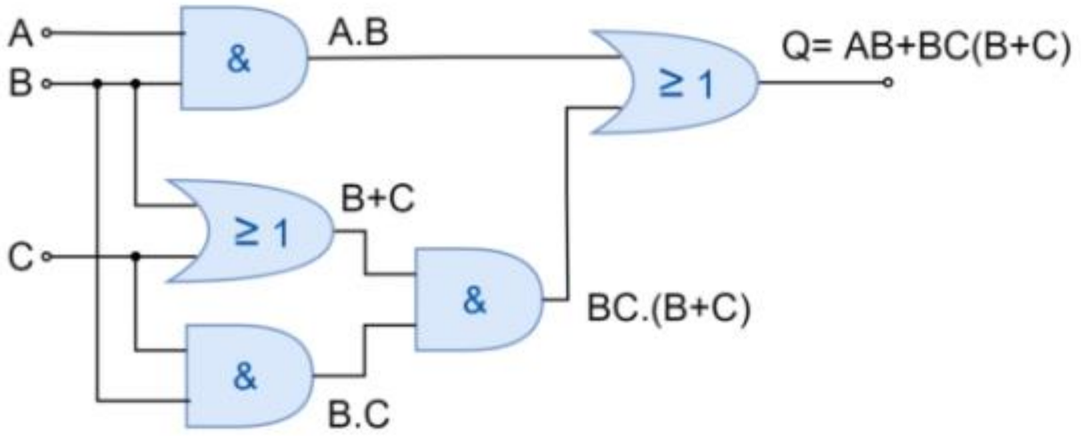
$$A\overline{B} + \overline{C}(A + B)$$

Draw a logic gate circuit for this function.



Boolean Algebra Simplification

In the following figure, a digital logic circuit is shown. The circuit consists of three AND gates and two OR gates. A total of five gates are required to produce the desired logic function. In this example, the desired logic function is reduced by applying Boolean Algebra Laws and Theorems.



Boolean Algebra Simplification

The steps involved in the reduction of Boolean expression are as follow:

AB + BC(B+C)

Applying Distributive Law

AB + BBC + BCC

Applying Identity Law (AA=A) to 2nd and 3rd terms

AB + BC + BC

Applying Identity Law (A + A = A) to 2nd and 3rd terms

AB + BC

Taking out common B

B(A+C)

Boolean Algebra Simplification

The steps involved in the reduction of Boolean expression are as follow:

AB + BC(B+C)

Applying Distributive Law

AB + BBC + BCC

Applying Identity Law (AA=A) to 2nd and 3rd terms

AB + BC + BC

Applying Identity Law (A + A = A) to 2nd and 3rd terms

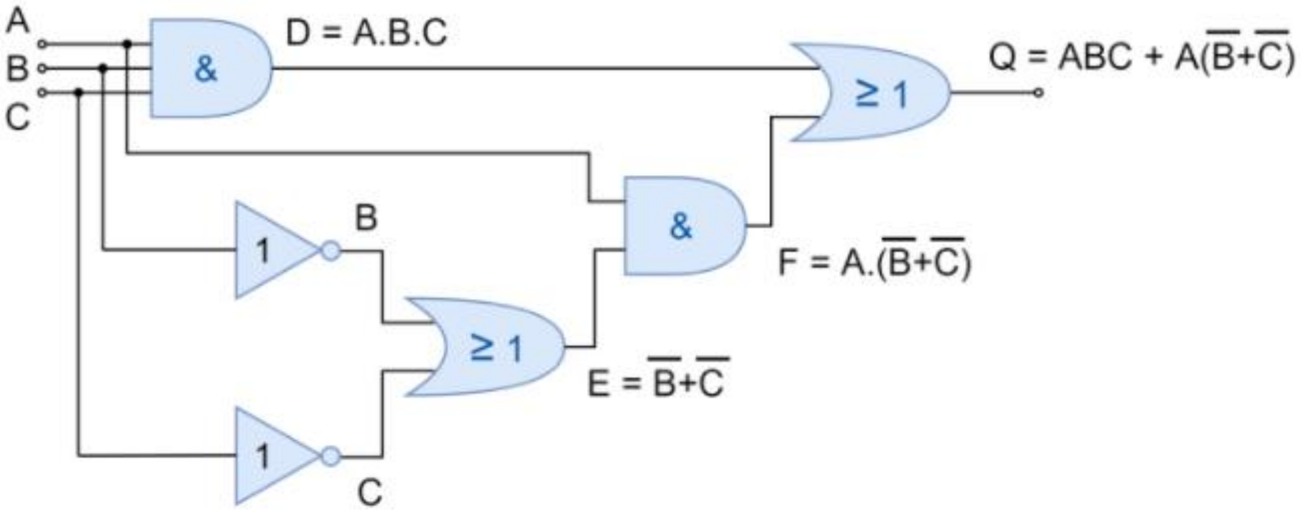
AB + BC

Taking out common B

B(A+C)

Boolean Algebra Simplification

In the final example, a more complex digital logic circuit has simplified which consists of six logic gates. The logic circuit to be reduced has been shown below.



Boolean Algebra Simplification

Equation to be reduced:

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

Applying De Morgan's Law: $\bar{B} + \bar{C} = \overline{BC}$

$$= ABC + A(\overline{BC})$$

Factoring (A) out:

$$= A(BC + \overline{BC})$$

Applying Complement Law: $BC + \overline{BC} = 1$

$$= A.1$$

Applying Identity Law: $A.1 = A$

$$= A$$

Boolean Algebra Simplification

https://www.electronics-tutorials.ws/boolean/boolean-algebra-simplification.html

شكراً لحسن المتابعة