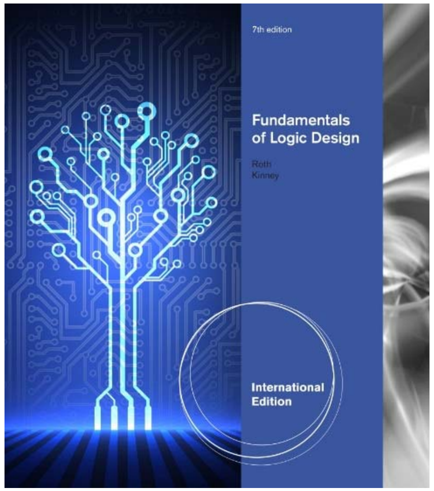
CHAPTER 2

Boolean Algebra



This chapter in the book includes:

Objectives

Topics introduced in this chapter:

- Understand the basic operations and laws of Boolean algebra
- Relate these operations and laws to AND, OR, NOT gates and switches
- Prove these laws using a truth table
- Manipulation of algebraic expression using
  - Multiplying out
  - Factoring
  - Simplifying
  - Finding the complement of an expression

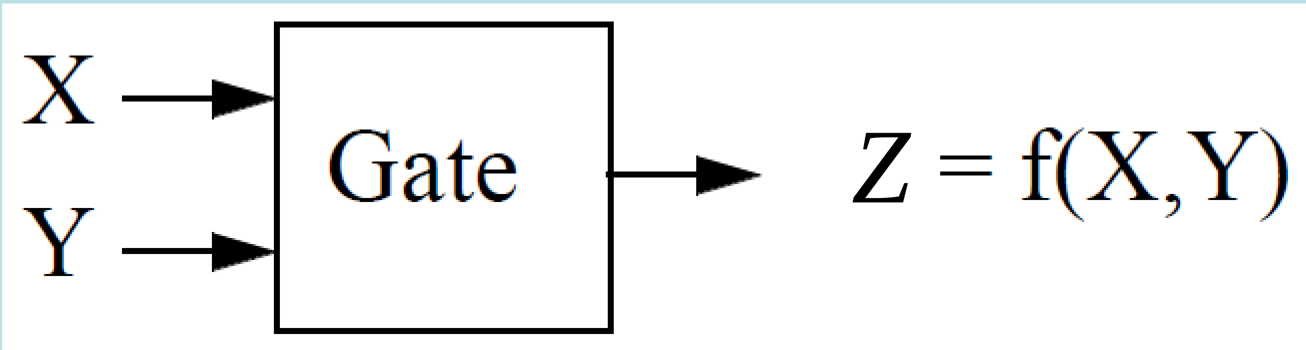
2.1 Introduction

- Basic mathematics for logic design: Boolean algebra
- Restrict to switching circuits(Two state values 0, 1) – Switching algebra
- Boolean Variable : X, Y, … can only have two state values (0, 1)

  – representing True(1) False (0)

Gate

- **Gate** : A simple electronic circuit (a system) that realizes a logical operation.

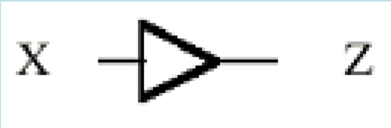


Truth Table

- **Truth Table** : listing all of it possible input configurations
  and the corresponding output signal
  - » The use of the symbols L and H usually correlates
    with the **high** and **low** voltages.
  - » The use of 0 (F) and 1(T) must be associated with
    the voltages. It does not matter which way it is done.
    - * If 1 is assigned to H and 0 to L ====> ***positive logic***
    - * If 0 is assigned to H and 1 to L ====> ***negative logic***
- We will use the positive logic convention
  unless explicitly indicated otherwise.

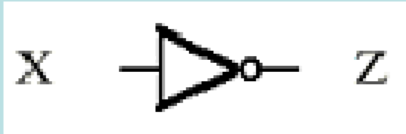
Standard Gates & Symbols

- Buffer



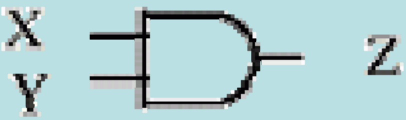
X	Z
0	0
1	1

- Not (Inverter or Complement)



X	Z
0	1
1	0

- AND



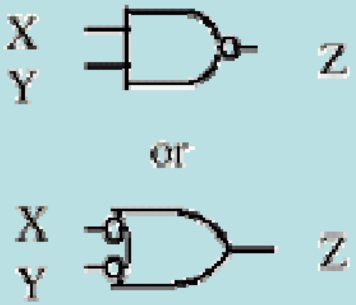
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

- OR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

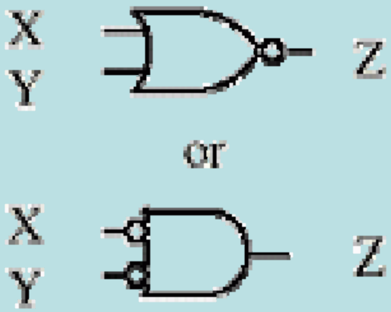
Standard Gates & Symbols

- NAND



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

- NOR

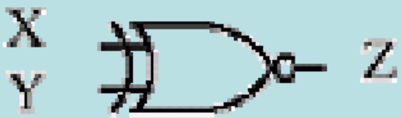


X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

- XOR (exclusive OR)

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

- Equivalence



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

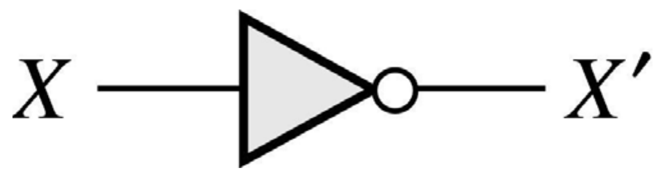
2.2 Basic Operations

NOT(Inverter)

$$0' = 1 \qquad \text{and} \qquad 1' = 0$$

$$X' = 1 \quad \text{if} \quad X = 0 \qquad \text{and} \qquad X' = 0 \quad \text{if} \quad X = 1$$

Gate Symbol



2.2 Basic Operations

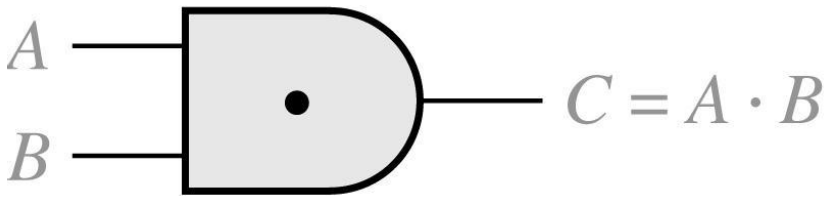
AND

$$0 \cdot 0 = 0 \qquad 0 \cdot 1 = 0 \qquad 1 \cdot 0 = 0 \qquad 1 \cdot 1 = 1$$

Truth Table

A B	C = A · B
0 0	0
0 1	0
1 0	0
1 1	1

Gate Symbol



2.2 Basic Operations

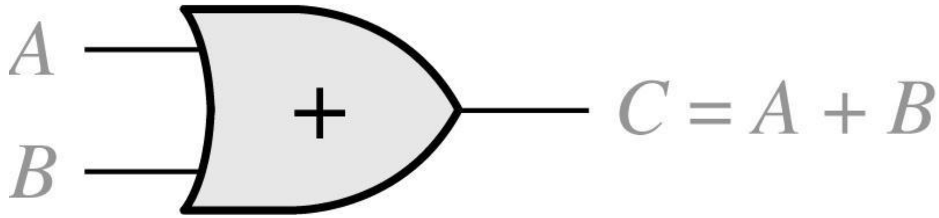
OR

$$0+0=0 \qquad 0+1=1 \qquad 1+0=1 \qquad 1+1=1$$

Truth Table

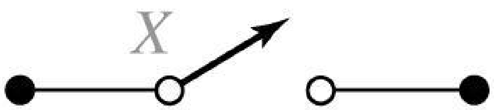
A B	C = A + B
0 0	0
0 1	1
1 0	1
1 1	1

Gate Symbol



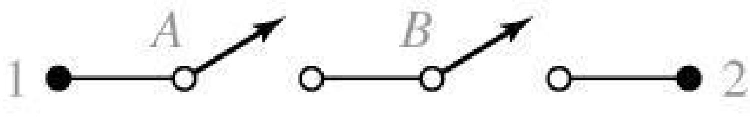
2.2 Basic Operations

Apply to Switch



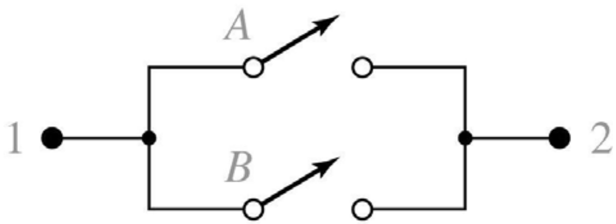
$X = 0 \rightarrow$ switch open
$X = 1 \rightarrow$ switch closed

AND $T = A \cdot B$



$T = 0 \rightarrow$ open circuit between terminals 1 and 2
$T = 1 \rightarrow$ closed circuit between terminals 1 and 2

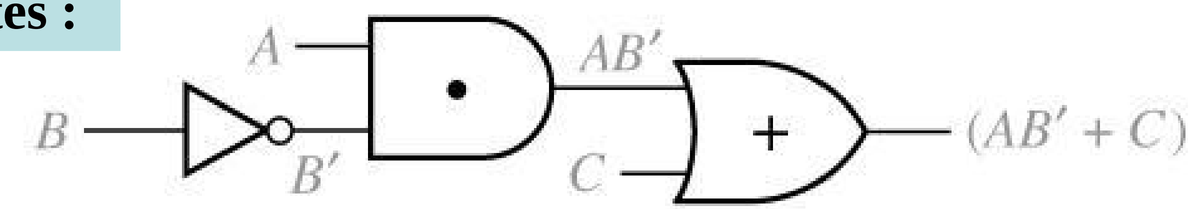
OR $T = A + B$



2.3 Boolean Expressions and Truth Tables

Logic Expression : $AB'+C$

Circuit of logic gates :

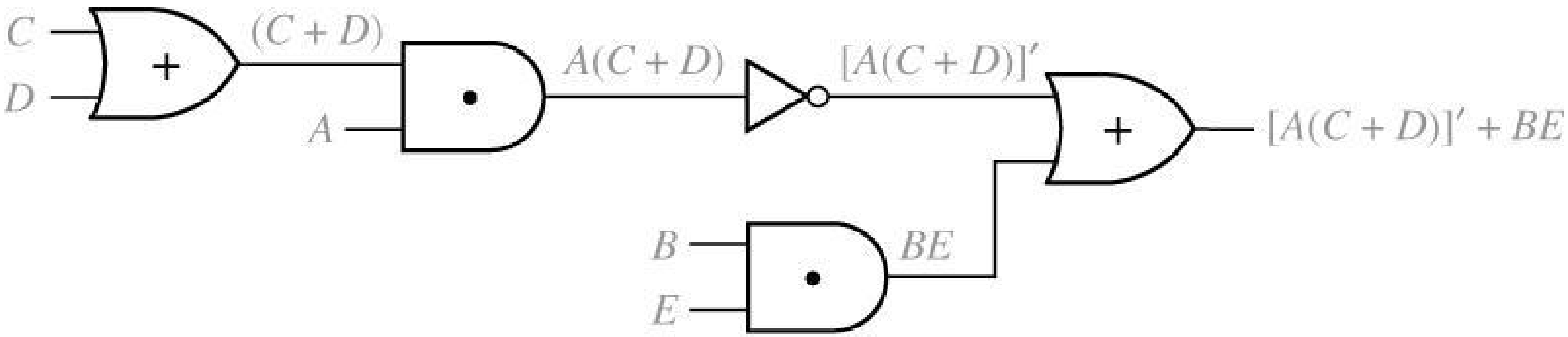


(a)

2.3 Boolean Expressions and Truth Tables

Logic Expression : $[A(C+D)]'+BE$

Circuit of logic gates :



Logic Evaluation : A=B=C=1, D=E=0

$$[A(C+D)]'+BE = [1(1+0)]'+1\cdot 0 = [1(1)]'+0 = 0+0 = 0$$

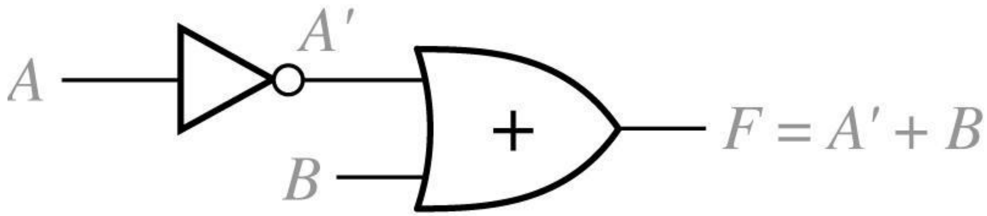
Literal : a variable or its complement in a logic expression

$$ab'c+a'b+a'bc'+b'c'$$

10 literals

2.3 Boolean Expressions and Truth Tables

2-Input Circuit and Truth Table



(a)

A B		A'	F = A' + B
0 0		1	1
0 1		1	1
1 0		0	0
1 1		0	1

2.3 Boolean Expressions and Truth Tables

Proof using Truth Table $AB'+C=(A+C)(B'+C)$

n variable needs $\underbrace{2\times 2\times 2\times \ldots}_{\text{n times}} = 2^n$ rows

TABLE 2.1

A B C		B'	AB'	AB' + C		A + C	B' + C	(A + C) (B' + C)
0 0 0		1	0	0		0	1	0
0 0 1		1	0	1		1	1	1
0 1 0		0	0	0		0	0	0
0 1 1		0	0	1		1	1	1
1 0 0		1	1	1		1	1	1
1 0 1		1	1	1		1	1	1
1 1 0		0	0	0		1	0	0
1 1 1		0	0	1		1	1	1

2.3 Boolean Expressions and Truth Tables

- So there are three ways of defining a switching (Boolean) function:
  - (1) Logical expression
  - (2) Truth table
  - (3) Logic Network (circuit)

  ===> Three representations all describe the same function

- Precedence(우선순위) in algebraic expressions:

  NOT AND OR except for brackets

2.4 Basic Theorems

Operations with 0, 1	$X+0=X$	$X\cdot 1=X$
	$X+1=1$	$X\cdot 0=0$
Idempotent Laws	$X+X=X$	$X\cdot X=X$
Involution Laws	$(X')'=X$	
Complementary Laws	$X+X'=1$	$X\cdot X'=0$

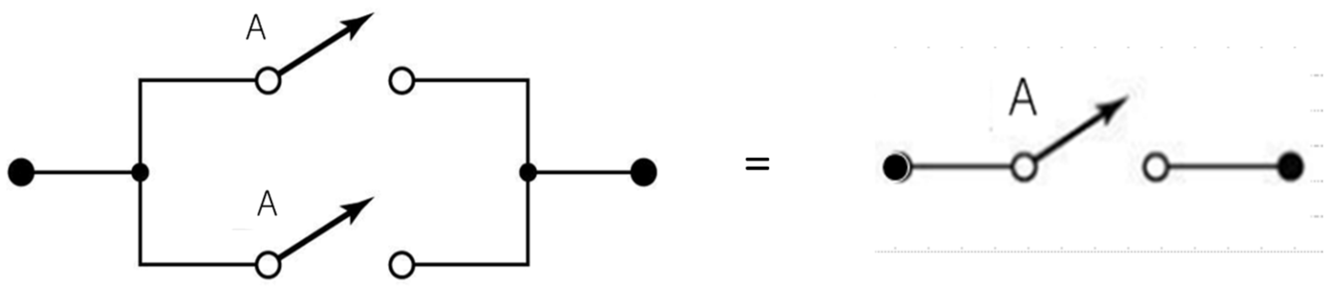
Proof $X=0,\quad 0+0'=0+1=1,\quad$ and if $X=1,\quad 1+1'=1+0=1$

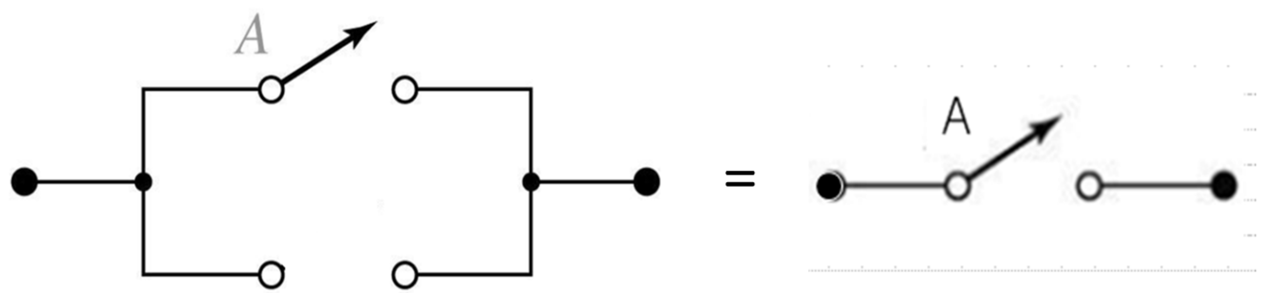
Example

$$(AB'+D)E+1=1$$

$$(AB'+D)(AB'+D)'=0$$

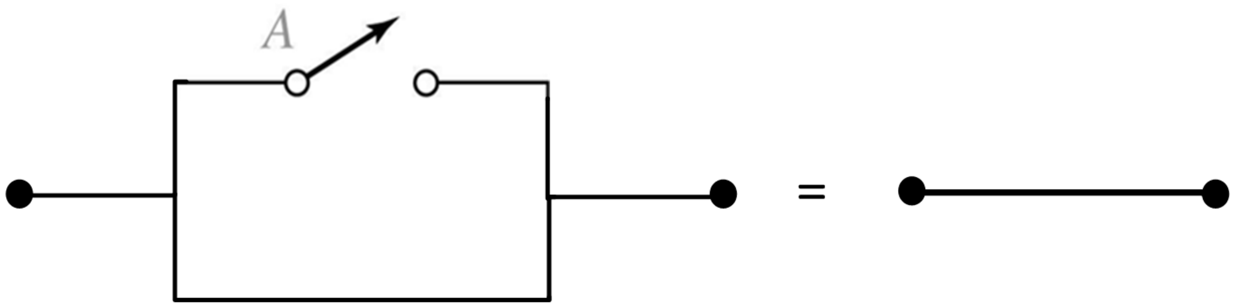
2.4 Basic Theorems with Switch Circuits



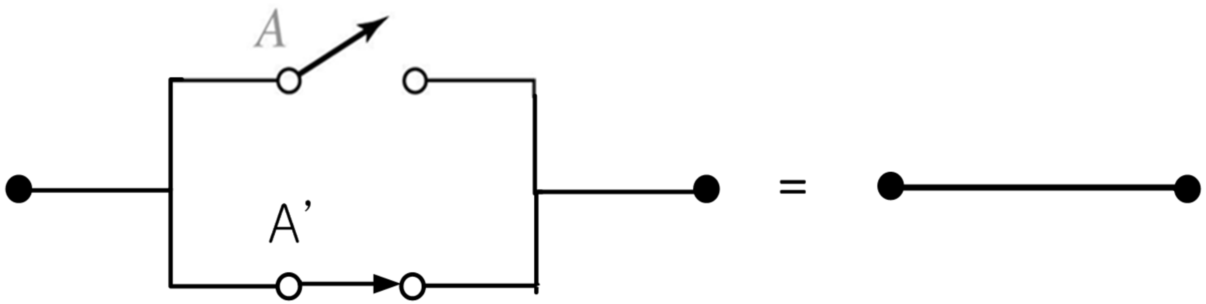


(A+0=A)

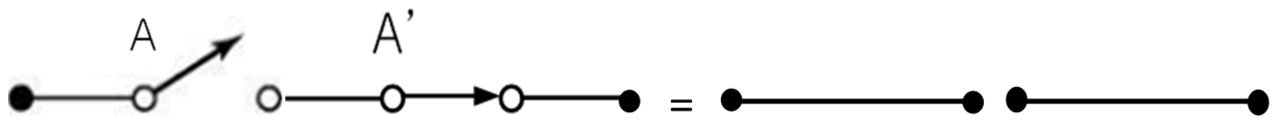
2.4 Basic Theorems with Switch Circuits



(A+1=1)



(A+A’=1)



(A A’=0)

2.5 Commutative, Associative, and Distributive Laws

Commutative Laws:

$$XY = YX \qquad\qquad X + Y = Y + X$$

Associative Laws:

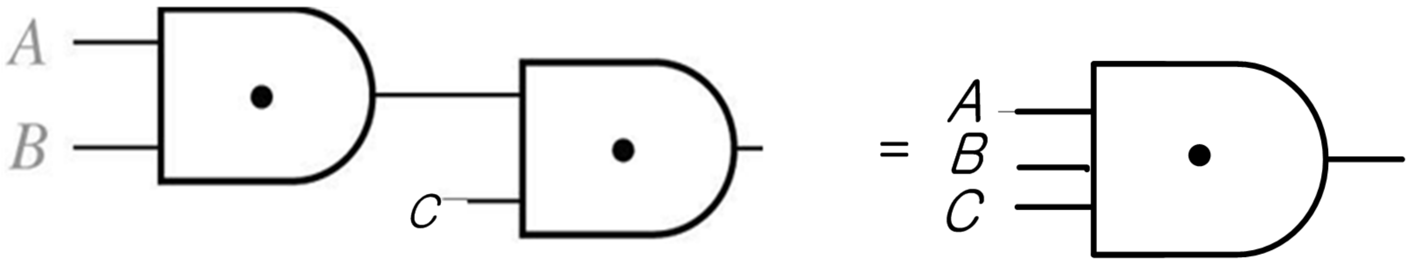
$$(XY)Z = X(YZ) = XYZ$$

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

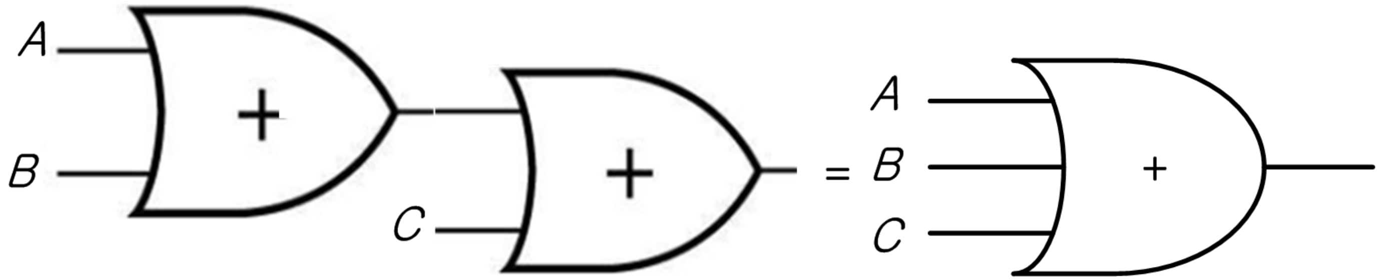
Proof of Associate Law for AND

X Y Z	XY	YZ	(XY)Z	X(YZ)
0 0 0	0	0	0	0
0 0 1	0	0	0	0
0 1 0	0	0	0	0
0 1 1	0	1	0	0
1 0 0	0	0	0	0
1 0 1	0	0	0	0
1 1 0	1	0	0	0
1 1 1	1	1	1	1

Associative Laws for AND and OR



(AB)C=ABC



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

2.5 Commutative, Associative, and Distributive Laws

AND $XYZ = 1 \quad \text{iff} \quad X = Y = Z = 1$

OR $X + Y + Z = 0 \quad \text{iff} \quad X = Y = Z = 0$

Distributive Laws:

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

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

Valid only Boolean algebra not for ordinary algebra

자주 활용됨

Proof

$$(X + Y)(X + Z) = X(X + Z) + Y(X + Z) = XX + XZ + YX + YZ$$
$$= X + XZ + XY + YZ = X \cdot 1 + XZ + XY + YZ$$
$$= X(1 + Z + Y) + YZ = X \cdot 1 + YZ = X + YZ$$

2.6 Simplification Theorems

Useful Theorems for Simplification

$$XY + XY' = X \qquad (X+Y)(X+Y') = X$$

$$X + XY = X \qquad X(X+Y) = X$$

$$(X+Y')Y = XY \qquad XY' + Y = X + Y$$

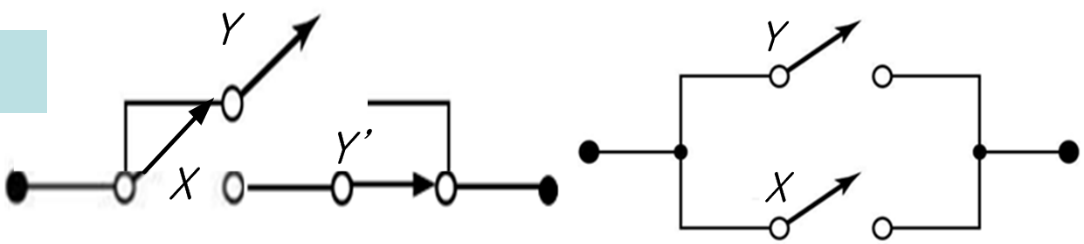
Proof

$$X + XY = X \cdot 1 + XY = X(1+Y) = X \cdot 1 = X$$

$$X(X+Y) = XX + XY = X + XY = X$$

$$Y + XY' = (Y+X)(Y+Y') = (Y+X)1 = Y + X$$

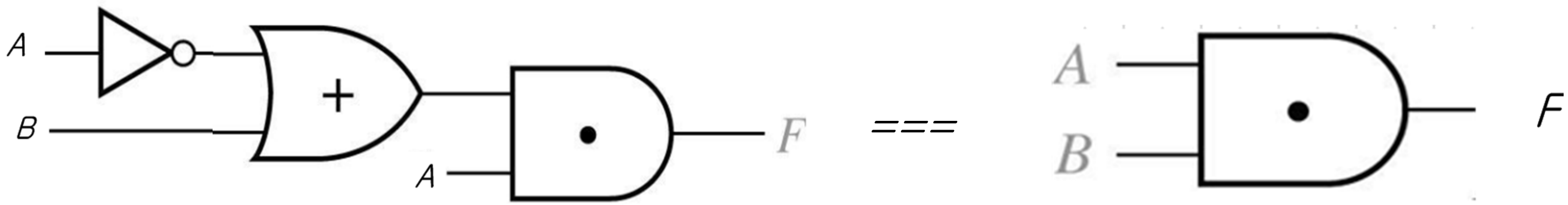
Proof with Switch



2.6 Simplification Theorems

Equivalent Gate Circuits

$$F = A(A' + B) = AB$$



2.7 Multiplying Out and Factoring

To obtain a sum-of-product form ➔ Multiplying out using distributive laws

Sum of product form: $AB' + CD'E + AC'E$

Still considered to be in sum of product form:

$ABC' + DEFG + H$

$A + B' + C + D'E$

Not in Sum of product form: $(A + B)CD + EF$

Multiplying out and eliminating redundant terms

$$
\begin{aligned}
(A + BC)(A + D + E) &= A + AD + AE + ABC + BCD + BCE \\
&= A(1 + D + E + BC) + BCD + BCE \\
&= A + BCD + BCE
\end{aligned}
$$

2.7 Multiplying Out and Factoring

To obtain a product of sum form → all sums are the sum of single variable

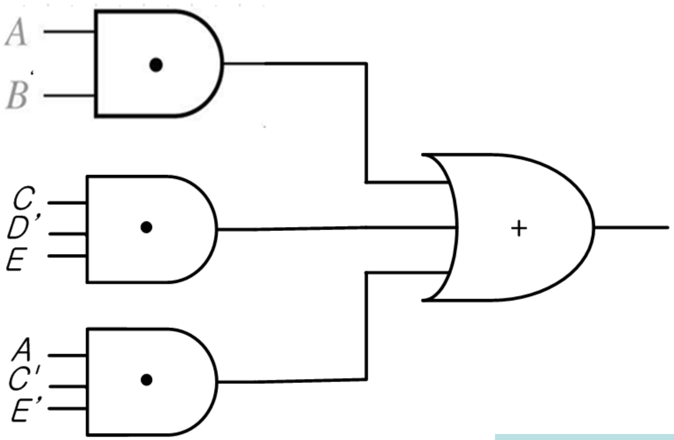
Product of sum form: $(A+B')(C+D'+E)(A+C'+E')$

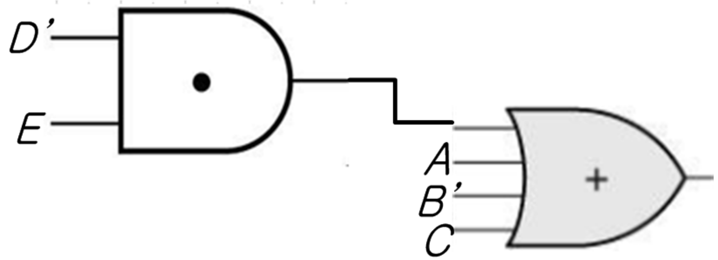
Still considered to be in product of sum form:

$$(A+B)(C+D+E)F$$

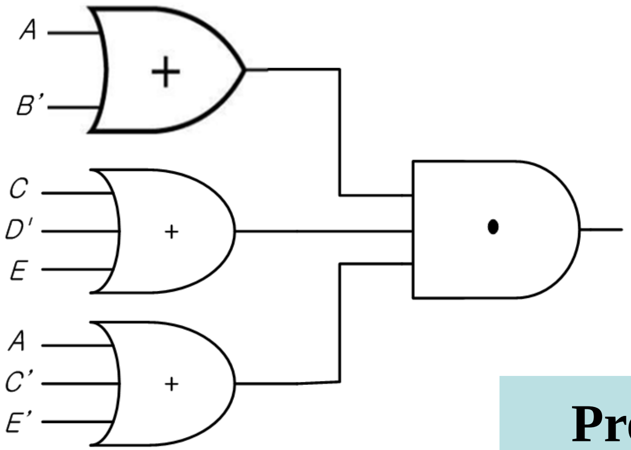
$$AB'C(D'+E)$$

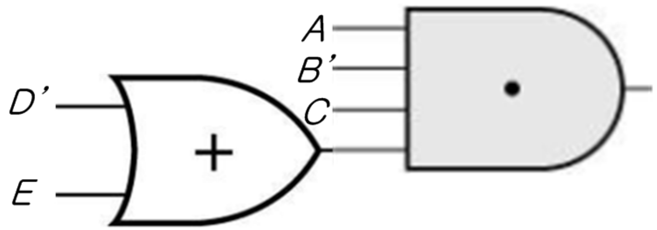
Circuits for SOP and POS form





Sum of product form:





Product of sum form:

2.8 DeMorgan's Laws

DeMorgan's Laws

$$(X+Y)' = X'Y'$$

$$(XY)' = X' + Y'$$

Proof

X Y	X' Y'	X + Y	(X + Y)'	X' Y'	XY	(XY)'	X' + Y'
0 0	1 1	0	1	1	0	1	1
0 1	1 0	1	0	0	0	1	1
1 0	0 1	1	0	0	0	1	1
1 1	0 0	1	0	0	1	0	0

DeMorgan's Laws for *n* variables

$$(X_1 + X_2 + X_3 + \dots + X_n)' = X_1' X_2' X_3' \dots X_n'$$

$$(X_1 X_2 X_3 \dots X_n)' = X_1' + X_2' + X_3' + \dots + X_n'$$

Example

$$(X_1 + X_2 + X_3)' = (X_1 + X_2)' X_3' = X_1' X_2' X_3'$$

2.8 DeMorgan's Laws

Inverse of
F=A'B+AB'

$$F' = (A'B + AB')' = (A'B)'(AB')' = (A + B')(A' + B)$$
$$= AA' + AB + B'A' + BB' = A'B' + AB$$

A B		A' B	A B'	F = A'B+AB'		A' B'	A B	F' = A'B' + AB
0 0		0	0	0		1	0	1
0 1		1	0	1		0	0	0
1 0		0	1	1		0	0	0
1 1		0	0	0		0	1	1

Dual: 'dual' is formed by replacing AND with OR, OR with AND, 0 with 1, 1 with 0

$$(XYZ\ldots)^D = X + Y + Z + \ldots \qquad (X + Y + Z + \ldots)^D = XYZ\ldots$$

The dual of an expression may be found by complementing the entire expression and then complementing each individual variable

$$(AB' + C)' = (AB')'C' = (A' + B)C', \quad \text{so} \quad (AB' + C)^D = (A + B')C$$