CSE 4205
Digital Logic Design

Boolean Algebra & Logic Gates

Course Teacher: Md. Hamjajul Ashmafee

Lecturer, CSE, IUT

Email: ashmafee@iut-dhaka.edu

Basic Definitions

- **Boolean algebra** – like others deductive mathematical systems – defined with a set of elements, a set of operators, and a number of unproved axioms or postulates
- **A set of elements, S** – any collections of objects having a common property – {a,b,c,…x,y,z}
- **A set of operators** – a set of rules that defined on the set **S** – {+,*}
- **Postulates** – form the basic assumptions – other rules, theorems, properties of the system are deduced from them

Common Postulates of a Mathematical System

1. **Closure:** A Set that is closed under an operation or collection of operations is said to satisfy a closure property. Or a *set* has closure under an operation if performance of that operation on members of the set always produce a member of the same set: the set is closed under the operation.
   - ***Example:*** The set of natural number, N ={1,2,3,…} is closed with respect to the binary operator plus (+) by the rules of arithmetic addition but not closed with respect to binary operator minus (-).
   - a,b ∈ **N** and we obtain a unique c ∈ N by the operation a+b=c but 2-3=-1 where 2,3 ∈ **N** but -1 ∉ **N**
2. **Associative Law:** A binary operator + on a set S is said to be associative whenever

   $(x+y)+z = x+(y+z)$, *for all* $x,y,z \in S$

Common Postulates…

3. **Commutative Law:** A binary operator + on a set S is said to be commutative whenever:

   $$x+y = y+x, \text{ for all } x,y \in S$$

4. **Identity element:** A set **S** is said to have an identity element with respect to a binary operation + on **S** if there exists an element e ∈ **S** with the property:

   $$x+e = e+x = x, \text{ for all } x,e \in S$$

   - *Example:* The element 0 is an identity element with respect to the operation + on the set of integers **I** = {…-3,-2,-1,0,1,2,3…} but not on the set **N.**
   - $x+0 = 0+x = x$, for any $x \in \mathbf{I}$

Common Postulates…

5. **Inverse:** A set **S** having the identity element e with respect to a binary operator + is said to have an inverse whenever, for every x ∈ *S* there exists an element y ∈ *S* such that

   $$x+y=e, \text{ for all } x,y,e \in S$$

   - ***Example:*** In the set I, e=0 and the inverse of every element a is (-a) since

     $a+(-a) = 0$

6. **Distributive Law:** If * and + are two binary operators on a set S, * is said to be distributive over + whenever:

   $$x*(y+z) = (x*y) + (x*z), \text{ for all } x,y,z \in S$$

Basic Definition

- **Field**: An algebraic structure that has a set of elements, together with two binary operators, each having properties 1 to 5 and both operators combined to give property 6.

  Example: Set of real number, **R**, with two binary operators + and * form the field of real numbers. (*verification ?*)

Boolean Algebra

- In 1854, George Boole introduced systematic treatment of logic and developed Boolean algebra
- In 1938, C. E. Shannon introduced a two level Boolean algebra called switching circuit/algebra
- For formal definition of Boolean algebra, we follow the postulates by E. V. Huntington (1904).
- Boolean algebra is a field with set of elements ***B,*** together with two binary operators + and **.** that follows **Huntington** postulates.

Huntington Postulates

1. **a**. Closure with respect to the operator +

   b. Closure with respect to the operator **.**
2. **a**. An identity element with respect to +, is designated by 0 :

   $x+0 = 0+x = x$

   b. An identity element with respect to **.**, is designated by 1 :

   $x.1 = 1.x = x$
3. **a**. Commutative with respect to + : $x+y = y+x$

   b. Commutative with respect to **.** : $x.y = y.x$
4. **a**. . is distributed over + : $x.(y+z) = (x.y) + (x.z)$

   b. + is distributed over . : $x+(y.z) = (x+y) . (x+z)$

Huntington Postulates

5. For every element $x \in B$ there exists an element $x' \in B$ (complement of x) such that
   a. $x + x' = 1$
   b. $x.x' = 0$
6. There exists at least two elements $x,y \in B$ such that $x \neq y$

Comparison between Boolean &Ordinary algebra

- Huntington postulates do not mention associative law but it holds here.
- The distributive law of + over . is only valid for Boolean algebra. *Example:* $x+(y.z) = (x+y).(x+z)$
- Boolean algebra does not have any additive or multiplicative inverse. So there is no subtraction or division operations in Boolean algebra
- Complement is only available in Boolean algebra
- Boolean algebra deals with set B having two elements 0 and 1 but real numbers deals with infinite set of elements

Basic Theorems and Postulates

- **Duality:** Every algebraic expression deducible from the postulates of the Boolean algebra remains valid if the operators and identity elements are interchanged. For this reason, Huntington postulates are listed in pairs. For dual of any algebraic expression, we simply interchange OR and AND operators and replace 1s by 0s and 0s by 1s.
- **Examlpe:** Postulate – 5 from Huntington postulates:
    a. $x + x' = 1$
    b. $x.x' = 0$

Basic Theorems

Postulate 2 (identity)	a. x+0 = x	b. x.1 = x
Postulate 3 (commutative)	a. x+y = y+x	b. x.y = y.x
Postulate 4 (distributive)	a. x(y+z) = xy+xz	b. x+yz = (x+y)(x+z)
Postulate 5 (complement)	a. x+x’ = 1	b. x.x’ = 0
Theorem 1	a. x+x = x	b. x.x = x
Theorem 2	a. x+1 = 1	b. x.0 = 0
Theorem 3 (involution)	(x’)’ = x	
Theorem 4 (associative)	a. x+(y+z) = (x+y)+z	b. x(yz) = (xy)z
Theorem 5 (De Morgan)	a. (x+y)’ = x’y’	b. (xy)’ = x’+y’
Theorem 6 (absorption)	a. x+xy = x	b. x(x+y) = x

Proof of Basic Theorems

THEOREM 1(a): $x + x = x$.

Proof:		
	$x + x = (x + x).1$	by postulate: 2(b)
	$= (x + x)(x + x')$	by postulate: 5(a)
	$= x + x.x'$	by postulate: 4(b)
	$= x + 0$	by postulate: 5(b)
	$= x$	by postulate: 2(a)

THEOREM 1(b): $x.x = x$.

Proof:		
	$x.x = xx + 0$	by postulate: 2(a)
	$= xx + xx'$	by postulate: 5(b)
	$= x(x + x')$	by postulate: 4(a)
	$= x.1$	by postulate: 5(a)
	$= x$	by postulate: 2(b)

THEOREM 2(a): $x + 1 = 1$.

Proof:		
	$x + 1 = 1.(x + 1)$	by postulate: 2(b)
	$= (x + x')(x + 1)$	by postulate: 5(a)
	$= x + x'.1$	by postulate: 4(b)
	$= x + x'$	by postulate: 2(b)
	$= 1$	by postulate: 5(a)

THEOREM 2(b): $x.0 = 0$ — by Duality Principle

THEOREM 3: $(x')' = x$.

Proof: we have $x + x' = 1$ and $x.x' = 0$, which defines the complement of x. The complement of x' is x and is also $(x')'$. Therefore, since the complement is unique, we have that $(x')' = x$.

Proof of Basic Theorems…

THEOREM 6(a): $x + xy = x$

Proof:	$x + xy = x.1 + xy$	by postulate 2(b)
	$= x(1 + y)$	by postulate 4(a)
	$= x(y + 1)$	by postulate 3(a)
	$= x \,.\, 1$	by postulate 2(a)
	$= x$	by postulate 2(b)

THEOREM 6(b): $x(x + y) = x$

Proof:	$x.(x + y) = x.x + x.y$	by postulate 4(a)
	$= x + x.y$	by theorem 1(b)
	$= x$	by theorem 6(a)

- Theorems of Boolean algebra can be proven using truth table.

Proof using Truth Table

- Absorption Rule (x + xy = x):

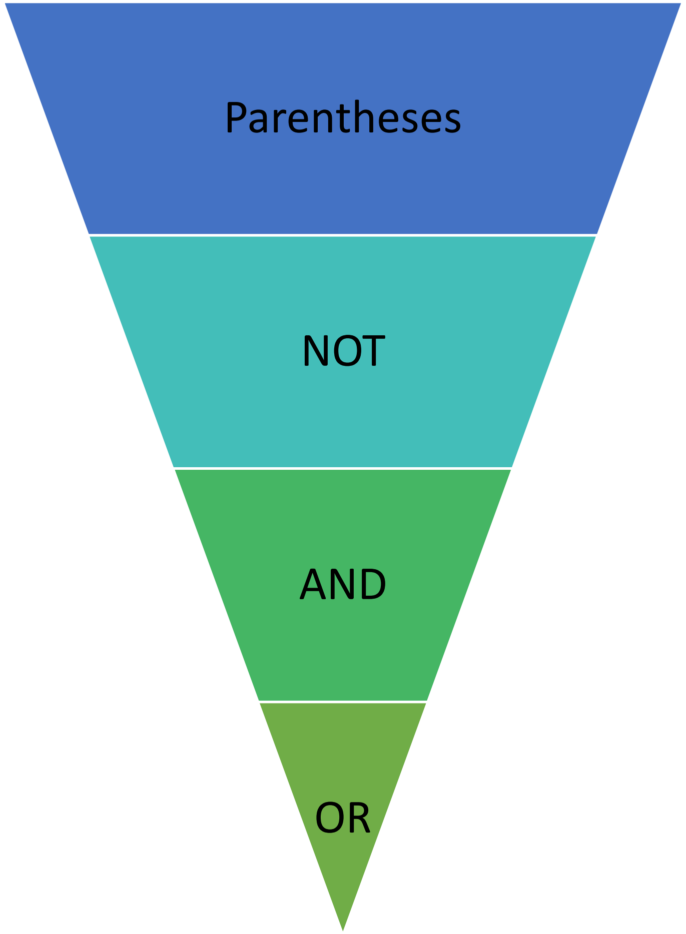
X	Y	XY	X+XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

- De Morgan's Law ((x+y)’ = x’+y’):

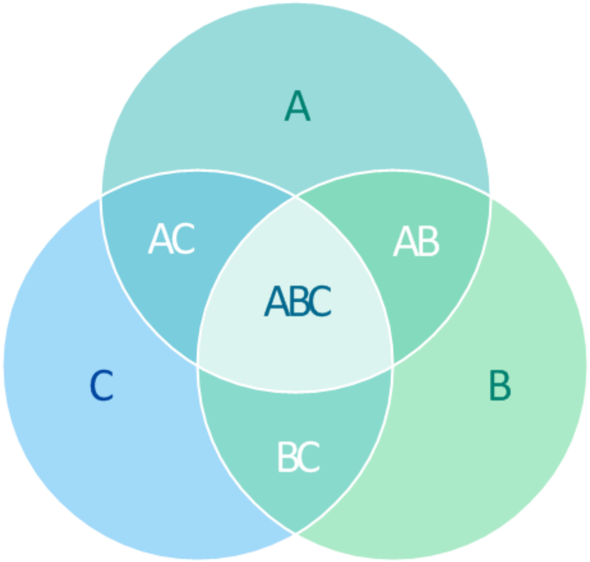
X	Y	X+Y	(X+Y)’	X’	Y’	X’Y’
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Operator Precedence

-



Venn Diagram



Boolean Function

- **Algebraic Expression**
- **Literals**

Boolean Function…

- **Truth Table**

Boolean Function…

- **Logic diagram**

Complement of a function

- De morgen's law
- Using Duality principle

Canonical & Standard Form

- Binary Variable – appeared as x or x'
- **Minterm:** A product that contains all variables of a particular function in either complemented (0/absence) or non-complemented (1/present) form. **(Standard Product)**
  - n variables can be combined with **AND** to form 2^n minterms. (counting $0 \sim 2^n$)
  - Symbol of minterm – m_j (Where j = decimal equivalent)
- **Maxterm:** A sum that contains all variables of a particular function in either non-complemented (0/absence) or complemented (1/present) form. **(Standard Sum)**
  - n variables can be combined with **OR** to form 2^n maxterms. (counting $0 \sim 2^n$)
  - Symbol of maxterm – M_j (Where j = decimal equivalent)
- *Maxterm and minterm are complements each other ($m_j = M_j'$).*

Minterms & Maxterms

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

Boolean Functions Using Minterms & Maxterms

- **Boolean expression** – from a given *truth table*, form a minterm for each combination of the variables which produces a **1** in the function and then the **OR** of all of those terms.

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

Boolean Functions Using Minterms & Maxterms…

Functions of Three Variables

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Functions Using Minterms & Maxterms…

- **Complement:** From the truth table, taking each combinations that produces 0in the function and then OR all of those terms.

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$

$f_2' =$?

Boolean Functions Using Minterms & Maxterms…

- **Boolean function using Maxterms**: Taking the complement of complemented function (f_1').

$$f_1 = (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z)$$
$$= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

$$f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z)$$
$$= M_0 M_1 M_2 M_4$$

- Any Boolean function can be written as a product (**AND**) of maxterms.
- *Boolean function written as a sum of minterms or product of maxterms* – **Canonical form**

Boolean Function in Canonical Form (Minterms)

- It is convenient to express the Boolean function in **canonical form** using minterms. (**Sum of minterms - SOP**)
  - Each term is inspected if it contains all the variables.
  - If any variable (x) is missed, **AND**ed that term with (x+x').

$$F = A + B'C$$

$$F = A'B'C + AB'C + AB'C + ABC' + ABC$$
$$= m_1 + m_4 + m_5 + m_6 + m_7$$

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Truth Table for $F = A + B'C$

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

Alternative method

Boolean Function in Canonical Form (Maxterms)

- Boolean function is also expressed in **canonical form** using maxterms (**Product of maxterms – POS**)
  - Each term is inspected if it contains all the variables.
  - If any variable (x) is missed, **OR**ed that term with (x.x’).
  - Use distributive law to express. (**x+yz=(x+y)(x+z)**)

$$F = xy + x'z$$

$$F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$$
$$= M_0M_2M_4M_5$$

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Conversions between Canonical Forms

- $m_j = M_j'$
- Example:

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

Conversions between Canonical Forms…

- Another Example:

$$F = xy + x'z$$

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Truth Table for $F = xy + x'z$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Minterms

Maxterms

Standard Forms

- Canonical form:
  - Formed from truth table
  - Rarely having least number of literals
- Standard form: Having one, two, three or any number of literals.
  - Two types: SOP and POS

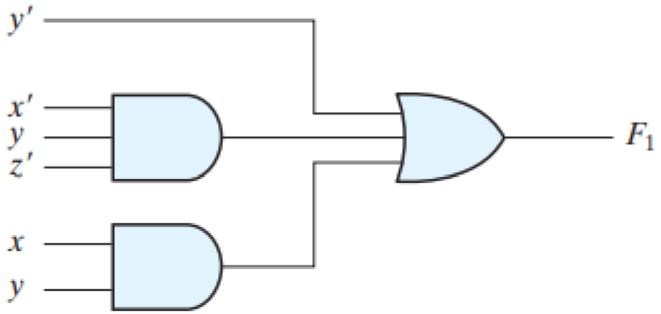
SOP: $F_1 = y' + xy + x'yz'$

POS: $F_2 = x(y' + z)(x' + y + z')$

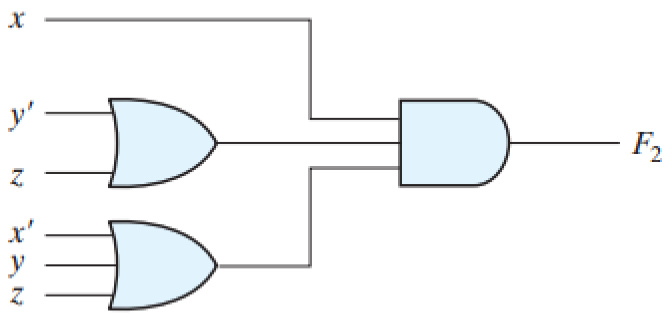
Non standard Form: $F_3 = AB + C(D + E)$

Standard Form: $F_3 = AB + C(D + E) = AB + CD + CE$

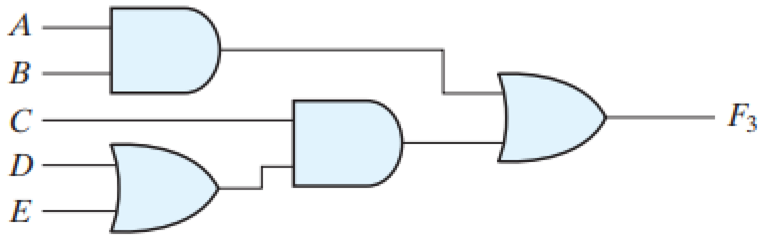
Standard Forms – 2 level Implementation



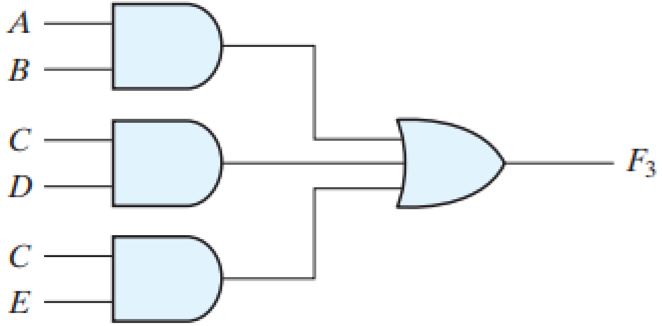
(a) Sum of Products



(b) Product of Sums



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



(b) $AB + CD + CE$

Other Logic Operations (Home task)

- Boolean expression and symbols for all possible logic operators
- Extension of multiple inputs