Chapter 2

Boolean Algebra and Logic Gates

2-1 Basic definitions

- Boolean Algebra or Symbolic Logic: Boolean algebra may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.
- It defines rules for manipulating symbolic binary logic expressions.
  - a symbolic binary logic expression consists of binary variables and the operators AND, OR and NOT (e.g. A+B.C')
- The possible values for any Boolean expression can be tabulated in a truth table.
- Boolean algebra can define circuit for expression by combining gates
- N.B.: In Boolean algebra there are only *three* basic operations: OR, AND, and NOT.

Algebras

- What is an algebra?
  - Mathematical system consisting of
    - Set of elements
    - Set of operators
    - Axioms or Postulates
- Why is this important?
  - Defines rules of "Calculations".
- Example: Arithmetic on natural numbers
  - Set of elements: N={1, 2, 3,...}
  - Operator: +, -, *, /, %
  - Axioms: Associativity, Distributivity, Closure etc.
- N.B.: Operators with two inputs are called binary operators, does not mean they restricted to binary Numbers.

Common Axioms or Postulates

- The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the systems.
  - If S is a set, and x and y are certain objects. A ***binary operator*** defined on a set S of elements is ***a rule*** that assigns to each pair of elements from S a unique element from S.
  - As an example, consider the relation a*b=c. We say that * is ***a binary operator*** if it satisfies ***a rule*** for finding c from the pair (a, b) and also if a, b, c ∈ S. However, * is not a binary operator if a, b ∈ S, while the rule finds c ∉ S.

Axioms or Postulates Cont…

- 1. ***Closure***: A set S is closed with respect to a binary operator if, for every pair of elements of S, the binary operator specifies ***a rule*** for obtaining an element of S. For example, the set of natural numbers is closed w.r.t binary operator (+) but not closed w.r.t. binary operator (-).
- 2. ***Associative law***: Associative: A binary operator * on a set S is said to be associative if
  $(x * y) * z = x * (y * z)$ for all $x, y, z \in S$.
- 3. ***Commutative Law***: A binary operator * on a set S is said to be commutative if $x * y = y * x$ for all $x, y \in S$.
- 4. ***Identity element***: Set S is said to have an identity element with ***respect to binary operation**** on S if there exists an element e member of S such that
  $e * x = x * e = x$ for every x member of S.

Axioms or Postulates Cont…

- Example: The element ***0*** is an identity element w.r.t. operation + on the set of integers I since $x + 0 = 0 + x = x$, for any $x \in I$. The element ***1*** is an identity element w.r.t. operation · on the set of integers I since $x \cdot 1 = 1 \cdot x = x$.
- 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 member*** *of S*, there ***exists an element y*** member of S such that $x * y = e$.
  - Example: In the set of *integers I with e = 0* the inverse of an element a is (-a) such that $a + (-a) = 0$.
- 6. ***Distributive Law***: If * and · are two binary operators on a set S, * is said to be distributive over · whenever $x * (y \cdot z) = (x * y) \cdot (x * z)$.
- N.B.: Identity elements depend on operator.

Axioms or Postulates Cont…

- The operator and postulates have the following meaning:
  - The binary operator + defines addition.
  - The additive identity is 0.
  - The additive inverse defines subtraction.
  - The binary operator · defines multiplication.
  - The multiplicative identity is 1.
  - The multiplicative inverse of a=1/a defines division, that is a.1/a=1.
  - The only distributive law is applicable is that of · over +: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Axiomatic definition of Boolean Algebra

- Boolean algebra is an algebraic structure defined on a set of elements B together with two binary operators + and · provided the following (***Huntington***) postulates are satisfied:
  - 1. (a) Closure with respect to the operator +.
    (b) Closure with respect to the operator · .
  - 2. (a) An identity element with respect to +, designed by
    0: $x+0= 0+x =x$.
    (b) An identity element with respect to ·, designated by
    1: $x \cdot 1 = 1 \cdot x = x$.
  - 3.(a) Commutative w.r.t +: $x + y= y + x$.
    (b) Commutative w.r.t · : $x \cdot y = y \cdot x$.

Axiomatic definition Cont...

- 4. (a) $\cdot$ is distributed over $+$: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
  (b) $+$ is distributed over $\cdot$: $x + (y \cdot z) = (x + y) \cdot (x + z)$
- 5. For every element $x \in B$, there exist an element $x' \in B$ (called the complement of x) such that : a) $x + x' = 1$ and b) $x \cdot x' = 0$.
- 6. There exists at least two elements $x, y \in B$ such that $x \neq y$.

☐ ***Boolean vs. Ordinary Algebra***

- Postulates do not include associative law.
- Distributive law of $+$ over $\cdot$ holds for Boolean, not for ordinary algebra i.e., $x+(y \cdot z) = (x + y)\cdot(x + z)$
- Boolean does not have inverse elements. Thus, no subtraction or division operations

Axiomatic definition Cont...

- Complement is not defined in ordinary algebra.
- Ordinary deals with the real numbers, which constitute an infinite set of elements. Set of elements in Boolean algebra not yet defined. But there must be at least two elements. Boolean 0 and 1 represent the state of a voltage variable (logic level) and is used to express the effects that various digital circuits have on logic inputs.

☐ ***Two-Valued Boolean Algebra***

☐ A two-valued Boolean algebra is defined on a set of two elements, $B = \{0, 1\}$, with rules for the two binary operators $+$ and $\cdot$ as shown in the operator tables:

Two-Valued Boolean Algebra Cont...

x	y	x·y
0	0	0
0	1	0
1	0	0
1	1	1

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

x	x'
0	1
1	0

Observations: OR is +, AND is ·, and NOT is '.

☐ Here we show that the Huntington postulates are valid for the set $B = \{0, 1\}$ and two binary operators $+$ and $\cdot$.

☐ *Closure?*

- Yes, Because the result of each operation is either 1 or 0 and $0, 1 \in B$.

Two-Valued Boolean Algebra Cont...

☐ *Identity ?*

- Yes, $0+0=0$, $0+1=1+0=1$, $1\cdot1=1$, $1\cdot0=0\cdot1=0$

☐ *Commutative Law?*

- Yes, for $+$: $0+0=0$, $1+1=1$, $0+1=1+0=1$
- Yes, for $\cdot$: $0\cdot0=0$, $1\cdot1=1$, $0\cdot1=1\cdot0=0$

☐ Distributive Law?

- Yes, $x \cdot (y + z) = (x \cdot y)+(x \cdot z)$ can be shown to hold true from the operator tables by forming a truth table of all possible values of x, y, and z.

x	y	z	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Two-Valued Boolean Algebra Cont…

- *Complement?*
  - Yes, $x + x' = 1$; $0+0' = 0+1 = 1$ and $1+1' = 1+0 = 1$
  - Yes, $x \cdot x' = 0$; $0 \cdot 0' = 0 \cdot 1 = 0$ and $1 \cdot 1' = 1 \cdot 0 = 0$
- *Postulate 6?*
  - Yes, because the two-valued Boolean algebra has two distinct elements 1 and 0 with $1 \neq 0$

2-3 Basic Theorems and Properties of Boolean Algebra

The Duality Principle

- The dual of a Boolean expression is obtained by interchanging all ANDs and ORs, and all 0s and 1s. example: the dual of $A+(B \cdot C')+0$ is $A \cdot (B+C') \cdot 1$
- The duality principle states that if E_1 and E_2 are Boolean expressions then
  $E_1 = E_2 \Leftrightarrow \text{dual}(E_1) = \text{dual}(E_2)$
  where dual(E) is the dual of E. For example,
  $A+(B \cdot C')+0 = (B' \cdot C)+D \Leftrightarrow A \cdot (B+C') \cdot 1 = (B'+C) \cdot D$

Basic Theorems Cont…

Basic Theorems

- Table 2-1 lists six theorems of Boolean Algebra and four of its postulates. The theorems must be proven from the postulates.

TABLE 2-1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Postulate 5	(a) $x + x' = 1$	(b) $x \cdot x' = 0$
Theorem 1	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3, involution	$(x')' = x$	
Postulate 3, commutative	(a) $x + y = y + x$	(b) $xy = yx$
Theorem 4, associative	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Postulate 4, distributive	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Theorem 6, absorption	(a) $x + xy = x$	(b) $x(x + y) = x$

Basic Theorems Cont…

- Theorem 1(a): $x + x = x$

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

Note: Theorem 1(b) is the dual of Theorem 1(a)

- Theorem 1(b): $x \cdot x = x$

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

Basic Theorems Cont...

- Theorem 2(a): $x + 1 = 1$

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

- Theorem 2(b): $x \cdot 0 = 0$

$$x \cdot 0 = 0 + x \cdot 0$$
$$= x \cdot x' + x \cdot 0$$
$$= x \cdot (x' + 0)$$
$$= x \cdot x' = 0$$

Basic Theorems Cont...

- Theorem 3: $(x')'$

$$(x')' = (x')' + 0$$
$$= (x')' + x \cdot x'$$
$$= [(x')' + x] \cdot [(x')' + x']$$
$$= [x + (x')'] \cdot 1$$
$$= [x + (x')'] \cdot [x + x']$$
$$= x + (x')' \cdot x'$$
$$= x + 0$$
$$= x$$

Basic Theorems Cont...

- Theorem 5(a): $(x + y)' = x' \cdot y'$
- We show that $x + y$ and $x' \cdot y'$ are complementary.

Let $L = x + y$ and $L' = x' \cdot y'$

To be complement $L \cdot L' = 0$ and $L + L' = 1$

$$L \cdot L' = (x + y) \cdot (x' \cdot y')$$
$$= (x' \cdot y') \cdot x + (x' \cdot y') \cdot y$$
$$= (x' \cdot x) \cdot y' + x' \cdot (y' \cdot y)$$
$$= 0 \cdot y' + x' \cdot 0$$
$$= 0 + 0$$
$$= 0$$

Basic Theorems Cont...

- $L + L' = (x + y) + (x' \cdot y')$

$$= (x + y + x') \cdot (x + y + y')$$
$$= (1 + y) \cdot (x + 1)$$
$$= 1 \cdot 1$$
$$= 1$$

So it is proved that $(x + y)' = x' \cdot y'$.

- Theorem 5(b): $(x \cdot y)' = x' + y'$

Let, $L = x \cdot y$ and $L' = x' + y'$

$$L \cdot L' = (x \cdot y) \cdot (x' + y')$$
$$= (x \cdot (x' + y')) \cdot (y \cdot (x' + y'))$$
$$= (x \cdot x' + x \cdot y') \cdot (y \cdot x' + y \cdot y')$$

Basic Theorems Cont…

- $= (0 + x \cdot y') \cdot (y \cdot x' + 0)$

  $= (x \cdot x') \cdot (y \cdot y')$

  $= 0 \cdot 0$

  $= 0$

  $L + L' = (x \cdot y) + (x' + y')$

  $= (x + (x' + y')) \cdot (y + x' + y')$

  $= (1 + y') \cdot (1 + x')$

  $= 1 \cdot 1$

  $=1$

So it is proved that $(x \cdot y)' = x' + y'$

Basic Theorems Cont…

- Theorem 6(a): $x + x \cdot y = x$

  $x + x \cdot y = x \cdot 1 + x \cdot y$

  $= x \cdot (1 + y)$

  $= x \cdot 1$

  $= x$

- Theorem 6(b): $x \cdot (x + y) = x$

$X \cdot (x + y) + 0$

$= x \cdot (x + y) + x. x'$

$= x \cdot (x + y + x')$

$= x \cdot (1 + y)$

$= x$

Basic Theorems Cont…

- ***Operator Precedence***
- The operator precedence for evaluating Boolean expressions is
  1. Parentheses
  2. NOT
  3. AND
  4. OR

2-4 Boolean Functions

- A ***Boolean function*** is an expression formed with binary variables, two binary operators OR and AND, the unary operator NOT, parentheses, and equal sign. For a given value of the variables, the function can be either 0 or 1.
  - For example, $F_1 = xyz'$, $F_2 = x + y'z$, $F_3 = x'y'z + x'yz + xy'$ $F_4 = xy' + x'z$
- Boolean function may be represented in a Truth table. These four functions are shown in Table 2-2.
  - From table, we find that F_4 is same as F_3. Why? Two functions of n binary variables are said to be equal if they have same value for all possible 2^n combinations of the n variables.

Boolean Functions Cont...

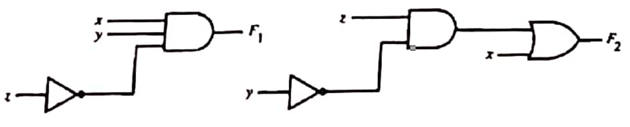
- A Boolean function may be transformed from an algebraic expression into a logic diagram composed of AND, OR, and NOT gates.

TABLE 2-2
Truth Tables for $F_1 = xyz'$, $F_2 = x + y'z$, $F_3 = x'y'z + x'yz + xy'$, and $F_4 = xy' + x'z$

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

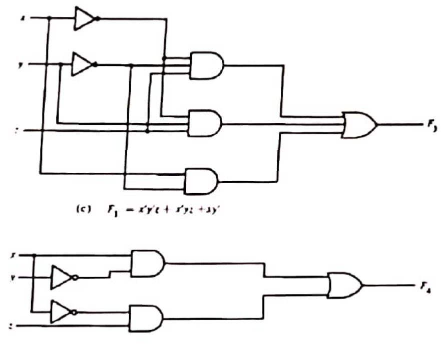
Boolean Functions Cont...

- The implementation of the four functions is shown in Figure 2-4.
- It is obvious that implementation of F_4 requires fewer gates and fewer inputs than F_3, since F_3 and F_4 are equal Boolean functions.



(a) $F_1 = xyz'$ (b) $F_2 = x + y'z$

Boolean Functions Cont...



(c) $F_3 = x'y'z + x'yz + xy'$

(d) $F_4 = xy' + x'z$

FIGURE 2-4
Implementation of Boolean functions with gates

Boolean Functions Cont...

- ***Algebraic Manipulation***
- A ***literal*** is a primed or unprimed variable. When a Boolean function is implemented with logic gates, each literal in the function designates an input to a gate, and each term is implemented with a gate. *The minimization of the number of literals and the number of terms results in a circuit with less equipment.*
- It is not always possible to minimize both simultaneously, usually further criteria must be available.

Algebraic Manipulation Cont...

- Example 2-1: Simplify the following Boolean functions to a minimum number of literals.

1. $x + x'y = (x + x')(x + y) = 1 \cdot (x + y) = x + y$
2. $x(x' + y) = xx' + xy = 0 + xy = xy$
3. $x'y'z + x'yz + xy' = x'z(y' + y) + xy' = x'z + xy'$
4. $xy + x'z + yz = xy + x'z + yz(x + x')$
   $= xy + x'z + xyz + x'yz$
   $= xy(1 + z) + x'z(1 + y)$
   $= xy + x'z$
5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$ by duality from function 4.

   - Fourth illustrates that an increase in the number of literals leads to a final simpler expression.

Complement of a Function

- The complement of a function F is F′ and is obtained from an interchange of 0′s for 1′s and 1′s for 0′s in the value of F. The complement of a function may be derived algebraically through De Morgan′s theorem.
  - Let F = (A + B + C)′. Find complement of F, that is F′. So for a function of n number of variables, the complement is?
- Example 2-2

Find the complement of the functions $F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$. By applying DeMorgan's theorem as many times as necessary, the complements are obtained as follows:

Complement of a Function Cont...

- Solution:

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$
$$F_2' = [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')' \cdot (yz)'$$
$$= x' + (y + z)(y' + z')$$

2-5 Canonical and Standard Forms

- ***Minterms and Maxterms***
- A binary variable may appear either in its normal form(x), or in its complement form (x′).
- Consider two binary variables x and y combined with an AND operation. So there are four possible combinations: x′y′, x′y, xy′, and xy. Each of these four AND terms is called a ***minterm*** or a ***standard product***. In a similar manner, n variables can be combined to form 2^n minterms.

Canonical and Standard Forms Cont…

- Consider two binary variables x and y combined with an OR operation. So there are four possible combinations: $x'+y'$, $x'+y$, $x+y'$, and $x+y$. Each of these four OR terms is called a ***maxterm*** or a ***standard sums***. In a similar manner, n variables can be combined to form 2^n maxterms.
- A Boolean function may be expressed algebraically from a given truth table by forming a minterm for each combination of the variables which produces a 1 in the function, and taking the OR of all those terms. Consider Table 2-4.

Canonical and Standard Forms Cont…

TABLE 2-3
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

Canonical and Standard Forms Cont…

- For example, the function f_1 in Table 2-4 is determined by expressing the combinations 001, 100, and 111 as $x'y'z$, $xy'z'$, and xyz respectively.

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

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

  N.B.: Any Boolean function can be expressed as a sum of minterms.

TABLE 2-4
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

Canonical and Standard Forms Cont…

- The complement of f_1 is read as:

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

  If we take the complement of f_1', we obtain the function f1 :

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

  Similarly it is possible to read the expression for f_2 from the table:

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

  Boolean functions expressed as a sum of minterms or product of maxterms are said to be in *canonical form*.

Canonical and Standard Forms Cont...

- ***Sum of Minterms***

Express the Boolean function $F = A + B'C$ in a sum of minterms. The function has three variables, A, B, and C. The first term A is missing two variables; therefore:

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

This is still missing one variable:

$$A = AB(C + C') + AB'(C + C')$$
$$= ABC + ABC' + AB'C + AB'C'$$

The second term $B'C$ is missing one variable:

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

Combining all terms, we have

$$F = A + B'C$$
$$= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$$

But $AB'C$ appears twice, and according to theorem 1 $(x + x = x)$, it is possible to remove one of them. Rearranging the minterms in ascending order, we finally obtain

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

It is sometimes convenient to express the Boolean function, when in its sum of minterms, in the following short notation:

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

Canonical and Standard Forms Cont...

- ***Product of Maxterms***

Express the Boolean function $F = xy + x'z$ in a product of maxterm form. First, convert the function into OR terms using the distributive law:

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

The function has three variables: x, y, and z. Each OR term is missing one variable; therefore:

$$x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$
$$x + z = x + z + yy' = (x + y + z)(x + y' + z)$$
$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

Combining all the terms and removing those that appear more than once, we finally obtain:

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

Canonical and Standard Forms Cont...

A convenient way to express this function is as follows:

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

The product symbol, Π, denotes the ANDing of maxterms; the numbers are the maxterms of the function.

- ***Conversions between Canonical Forms***
- The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function.

  As an example, consider the function: $F(A, B, C) = \sum(1, 4, 5, 6, 7)$

  This has a complement that can be expressed as: $F'(A, B, C) = \sum(0, 2, 3) = m_0 + m_2 + m_3$

Conversions between Canonical Forms Cont...

- If we take complement of F' by De Morgan's theorem, we obtain F in a different form:

  $F = (m_0 + m_2 + m_3)' = m'_0 \cdot m'_2 \cdot m'_3 = M_0M_2M_3 = \prod(0, 2, 3)$

  The last conversion follows from the definition of maxterms and minterms. It is clear that following relation holds true:

  $$m'_j = M_j$$

  That is, the maxterm with subscript j is a complement of the minterm with same subscript j, and vice versa.