CO
Unit-1

Part-2

Boolean Algebra And Logic Gates

Syllabus

- Basic Definitions
- Axiomatic definition of Boolean Algebra
- Basic theorems and properties of Boolean algebra
- Boolean functions - canonical and standard forms

Basic Definitions

Boolean Algebra:

- A deductive mathematical system
- Defined with
  - A set of elements
    - ex: B = {0, 1}
  - A set of operators
    - ex: +, *, …
  - A number of unproved axioms or postulates

Basic Definitions

- The most common postulates used to formulate various algebraic structures are:

1. Closure. N={1,2,3,4…}, for any a,b $\in$N we obtain a unique c $\in$N by the operation a+b=c. Ex:2−3= −1 and 2,3 $\in$N, while (−1) $\notin$ N.
2. Associative law. A binary operator $*$ on a set S is said to be associative whenever

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

3. Commutative law.

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

Basic Definitions

4. Identity element. e is identity element which belongs to S.

   $e * x = x * e = x$ for every $x \in S$

   Ex: $x + \boxed{0} = 0 + x = x$ for any $x \in$ I={…,−2, −1, 0, 1, 2,…}

   $x * \boxed{1} = 1 * x = x$

5. Inverse. In the set of integers, I, with e = 0

   $x * \boxed{y} = e\,; \quad a + (\boxed{-a}) = 0$

   −a and y are inverse elements

6. Distributive law. If $*$ and $\cdot$ are two binary operators on a set S, $*$ is said to be distributive over $\cdot$ Whenever

   $$x * (y \cdot z) = (x * y) \cdot (x * z)$$

Basic Definitions

- The operators and postulates have the following meanings:

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, i.e.,

$$a \cdot 1/a = 1$$

The only distributive law applicable is that of · over +:

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

AXIOMATIC DEFINITION OF BOOLEAN ALGEBRA

- In 1854, George Boole
  - Introduced a systematic treatment of logic
  - Developed an algebraic system called ***Boolean algebra***
- In 1938, C. E. Shannon
  - Introduced a two-valued Boolean algebra called ***switching algebra***
  - The properties of bistable electrical switching circuits (digital circuits) can be represented by it
- In 1904, E. V. Huntington
  - Formulate the postulates as the formal definitions

AXIOMATIC DEFINITION OF BOOLEAN ALGEBRA

- Boolean algebra is defined by a set of elements, B, provided following postulates with two binary operators, + and ·, are satisfied:

1. Closure with respect to the operators + and ·.
2. An identity element with respect to + and · is 0 and 1,respectively.
3. Commutative with respect to + and ·. Ex: $x + y = y + x$
4. + is distributive over · : $x + (y \cdot z)=(x + y) \cdot (x + z)$
   . is distributive over + : $x \cdot (y + z)=(x \cdot y) + (x \cdot z)$
5. Complement elements: $x + x' = 1$ and $x \cdot x' = 0$.
6. There exists at least two elements $x,y \in B$ such that $x \neq y$.

AXIOMATIC DEFINITION OF BOOLEAN ALGEBRA

1. Huntington postulates don't include the associative law, however, this holds for Boolean algebra.
2. The distributive law of + over · is valid for Boolean algebra, but not for ordinary algebra.
3. Boolean algebra doesn't have additive and multiplicative inverses; therefore, no subtraction or division operations.
4. Postulate 5 defines an operator called complement that is not available in ordinary algebra.
5. Ordinary algebra deals with the real numbers. Boolean algebra deals with the as yet undefined set of elements, B, in two-valued Boolean algebra, the B have two elements, 0 and 1.

Two-Valued Boolean Algebra

A two-valued Boolean algebra is

- Defined on a set of two elements B = {0, 1}
- With rules for the binary operators + and *

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

- Satisfying the six Huntington postulates
- Called "switching algebra", "***binary logic***"

Two-Valued Boolean Algebra

- With rules for the two binary operators + and . as shown in the following table, exactly the same as AND, OR , and NOT operations, respectively.
- From the tables as defined by postulate 2.

x	y	$x \cdot y$
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

Two-Valued Boolean Algebra

x	y	z	$y + z$	$x \cdot (y + z)$	$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
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

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

- To emphasize the similarities between two-valued Boolean algebra and other binary systems, this algebra was called "binary logic". We shall drop the adjective "two-valued" from Boolean algebra in subsequent discussions.

Basic Theorems and Properties

- If the binary operators and the identity elements are interchanged, it is called the duality principle. We simply interchange OR and AND operators and replace 1's by 0's and 0's by 1's.

- The theorem 1(b) is the dual of theorem 1(a) and that each step of the proof in part (b) is the dual of part (a). Show at the slice after next slice.

Basic Theorems and Properties

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 and Properties

- Theorem 1(a)

  $X + X = X$

<proof>

$X + X = (X + X) * 1$ .... p2(b)

$= (X + X)(X + X')$ ... p5(a)

$= X + XX'$ .... p4(b)

$= X + 0$ .... P5(b)

$= X$ .... P2(a)

- Theorem 1(b)

  $X * X = X$

<proof>

$X * X = XX + 0$ .... p2(a)

$= XX + XX'$ .... p5(b)

$= X(X + X')$ .... p4(a)

$= X * 1$ .... p5(a)

$= X$ .... p2(a)

Basic Theorems and Properties

- Theorem 2(a): X + 1 = 1

<proof>

X + 1 = 1 * (X + 1) .... p2(b)

= (X + X‘) (X + 1) .... p5(a)

= X + X‘ * 1 .... p4(b)

= X + X‘ .... p2(b)

= 1 .... p5(a)

- Theorem 2(b): X * 0 = 0

<proof>

by duality of theorem 2(a)

Basic Theorems and Properties

- Theorem 3: $(X')' = X$ (involution, 乘方)

<proof>

$X + X' = 1$ and $X * X' = 0$ (from p5)

$\Rightarrow$ X and X' are complement to each other

$\therefore$ the complement of $X' = (X')' = X$

Basic Theorems and Properties

- Theorem 4: (associative)

  (a) X + (Y + Z) = (X + Y) + Z

  (b) X (YZ) = (XY) Z

<proof for (a)>

X + (Y + Z) = X*1 + (Y + Z)*1 .... p2(b)

= X*1 + Y*1 + Z*1 .... p4(a)

= (X + Y)*1 + Z*1 .... p4(a)

= (X + Y) + Z .... p2(b)

<proof for (b)>

can be obtained by the duality of theorem 4(a)

Basic Theorems and Properties

- Theorem 5: (DeMorgan)

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

<proof>

(a) by truth table

x y	x+y	(x+y)'
0 0	0	1
0 1	1	0
1 0	1	0
1 1	1	0

x'	y'	x'y'
1	1	1
1	0	0
0	1	0
0	0	0

(b) can be proofed by similar way

Basic Theorems and Properties

- Theorem 6: (absorption, 合併)

  (a) $X + XY = X$ (b) $X (X + Y) = X$

<proof for (a)>

$X + XY = X * 1 + XY$ .... p2(b)

$= X (1 + Y)$ .... p4(a)

$= X (Y + 1)$ .... p3(a)

$= X * 1$ .... p2(a)

$= X$ .... p2(b)

<proof for (b)>

can be obtained by the duality of theorem 6(a)

Basic Theorems and Properties

THEOREM 6(b): $x(x + y) = x$ by duality.

The theorems of Boolean algebra can be proven by means of truth tables. In truth tables, both sides of the relation are checked to see whether they yield identical results for all possible combinations of the variables involved. The following truth table verifies the first absorption theorem:

x	y
0	0
0	1
1	0
1	1

xy	$x + xy$
0	0
0	0
0	1
1	1

Basic Theorems and Properties

The algebraic proofs of the associative law and DeMorgan's theorem are long and will not be shown here. However, their validity is easily shown with truth tables. For example, the truth table for the first DeMorgan's theorem, $(x + y)' = x'y'$, is as follows:

x	y	$x + y$	$(x + y)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

x'	y'	$x'y'$
1	1	1
1	0	0
0	1	0
0	0	0

Operator Precedence

(1) Parentheses (括號)

(2) NOT .... similar to the sign of numbers

(3) AND .... similar to multiplication

(4) OR .... similar to addition

Ex: $(X + Y)' + Z$

step 1: $X + Y$ .... inside the parentheses

step 2: $(X + Y)'$

step 3: $(X + Y)' + Z$

BOOLEAN FUNCTIONS

- A Boolean function is described by an algebraic expression that consists of
  - Binary variables
  - The constant 0 and 1
  - The logic operation symbols
- For a given value of the binary variables, the function can be equal to either 1 or 0
- Ex: $F1 = X + Y' Z$

  $X = 1 \rightarrow F1 = 1$

  $Y = 0$ and $Z = 1 \rightarrow F1 = 1$

  otherwise $\rightarrow F1 = 0$

BOOLEAN FUNCTIONS

- A Boolean function can be represented in a ***truth table***
  - An unique representation
- A truth table includes
  - A list of combinations of 1's and 0's assigned to the binary variables
  - A column that shows the value of the function for each binary combination
- Ex: F1 = X +Y' Z
  - No. of binary variables = 3
  - No. of rows = 2^3 = 8

x	y	z	F1
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

BOOLEAN FUNCTIONS

- A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

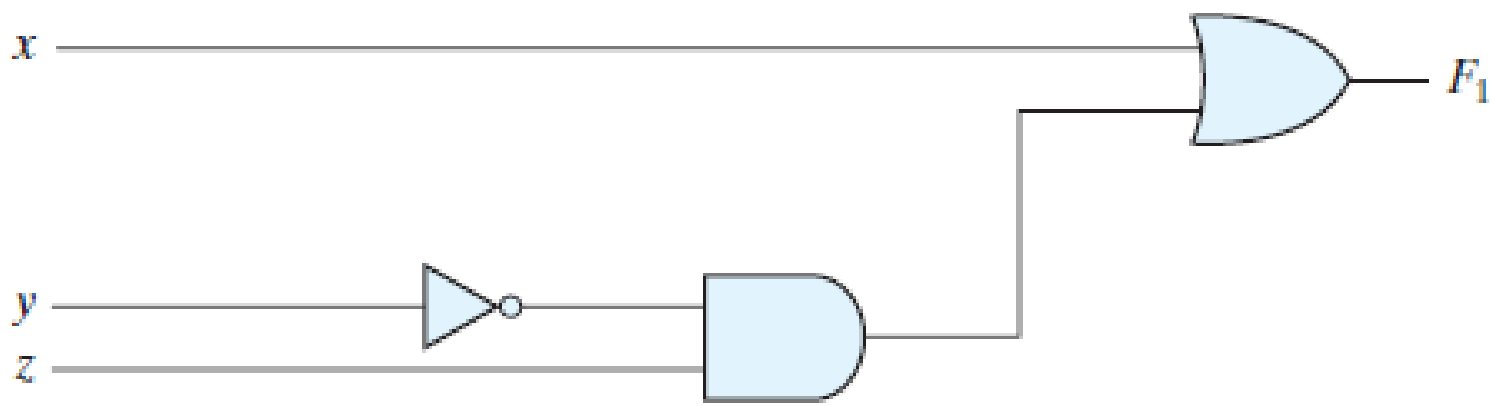
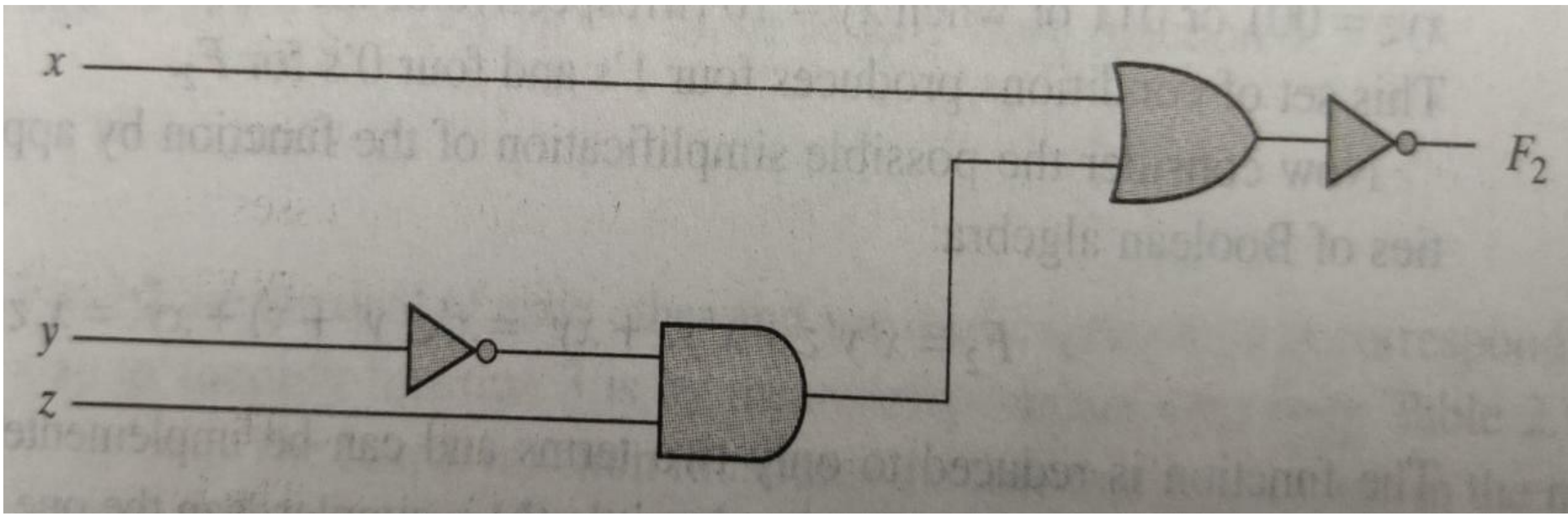


FIGURE 2.1
Gate implementation of $F_1 = x + y'z$

BOOLEAN FUNCTIONS

- There is only one way to represent Boolean function in a truth table.
- In algebraic form, it can be expressed in a variety of ways.
- By simplifying Boolean algebra, we can reduce the number of gates in the circuit and the number of inputs to the gate.

BOOLEAN FUNCTIONS



Before simplification of Boolean function

- Consider the following Boolean function:

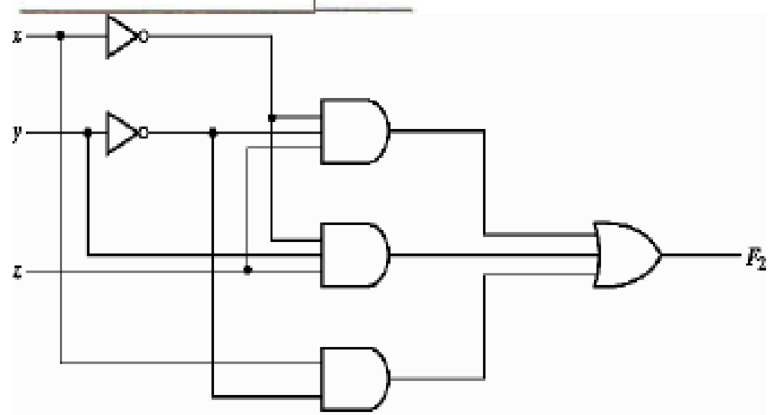
  $F_2 = x'y'z + x'yz + xy'$

This function with logic gates is shown in Fig. 2-2(a)

- The function is equal to 1 when xyz = 001 or 011 or when xyz = 10x .

Table 2-2
Truth Tables for F_1 and F_2

x	y	z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



(a) $F_2 = x'y'z + x'yz + xy'$

After simplification of Boolean function

- Simplify the following Boolean function:

  $F_2 = x'y'z + x'yz + xy'$

  $= x'z\,(y' + y) + xy'$

  $= x'z + xy'$

- In 2-2 (b), would be preferable because it requires less wires and components.

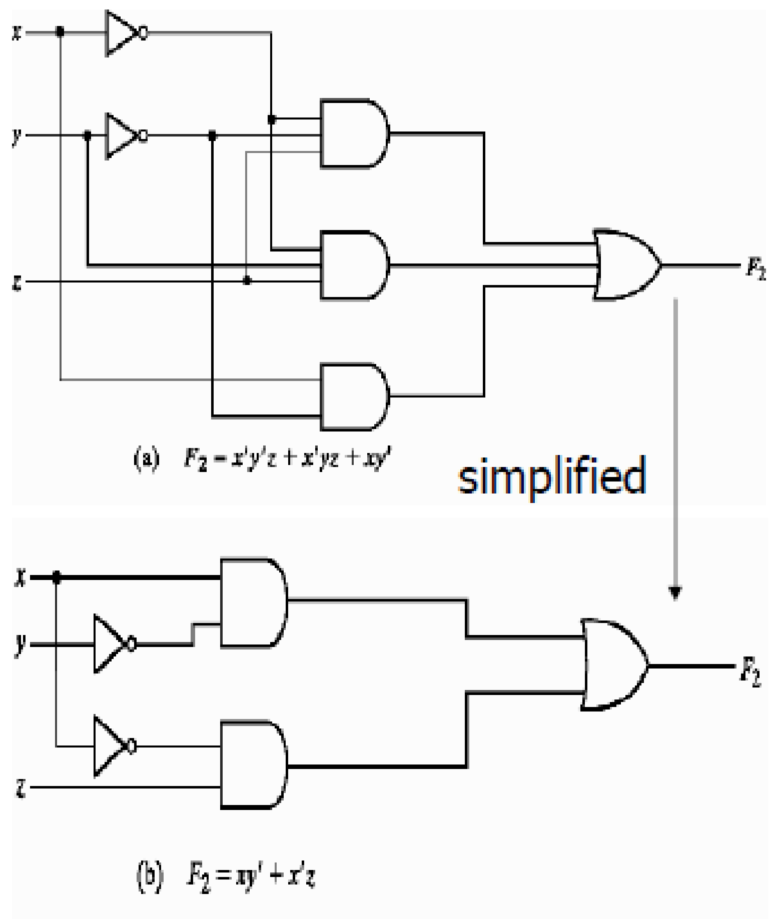


Fig. 2-2 Implementation of Boolean function F_2 with gates

Equivalent

$F_2 = x'y'z + x'yz + xy'$ (primitive)

F_2=1 when xyz=001 or 011 or

when xy=10x

$F_2 = x'z + xy'$ (simplified)

F=1 when xz=01 or

when xy=10

- Since both expression produce the same truth table, they are said to be equivalent.

Table 2-2
Truth Tables for F_1 and F_2

x	y	z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Algebraic Manipulation

- When a Boolean expression is implemented with logic gates, each term requires a gate and each variable within the term designates an input to the gate.
- Literal: a single variable within a term that may be complement or not
  - F2 = x'y'z + x'yz + xy' (3 terms, 8 literals)
  - F2 = xy' + x'z (2 terms, 4 literals)
- Reducing the **number of terms** and the **number of literals** can often lead to a simpler circuit

Algebraic Manipulation

counteracted

Ex 2-1: 4. $xy + x'z + yz$ → Consensus term

Merged

$$= xy + x'z + yz(x + x')$$

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

$$= xy(1 + z) + x'z(1 + y)$$

$$= xy + x'z$$

Function 5 can be derived from the dual of the steps used to derive function 4.

- Functions 4 and 5 are known as the consensus theorem.

Example 2-1

- Simplify the following Boolean functions to a minimum number of literals

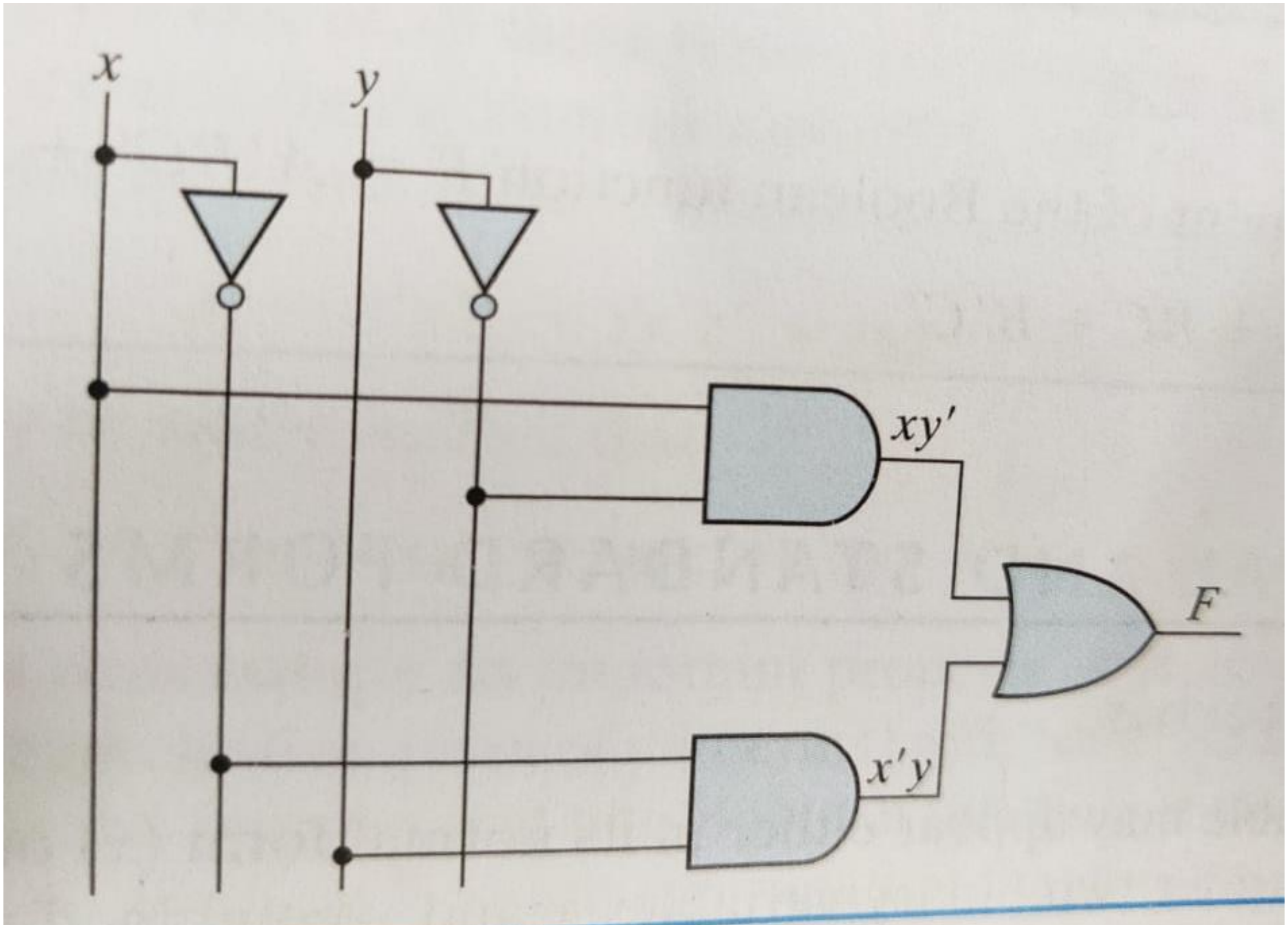
1. $x(x' + y) = xx' + xy = 0 + xy = xy$
2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y$
3. $(x + y)(x + y') = x + xy + xy' + yy'$
   $= x(1 + y + y') + 0 = x$
4. $xy + x'z + yz = xy + x'z + yz(x + x')$
   $= xy + x'z + xyz + x'yz$
   $= xy(1 + z) + x'z(1 + y) = xy + x'z$
5. $(x + y)(x' + z)(y + z)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

Example

- Draw a logic diagram for the Boolean function

$F=x'y+xy'$

Answer



Example

What boolean Expression is implemented by the following logic diagram?

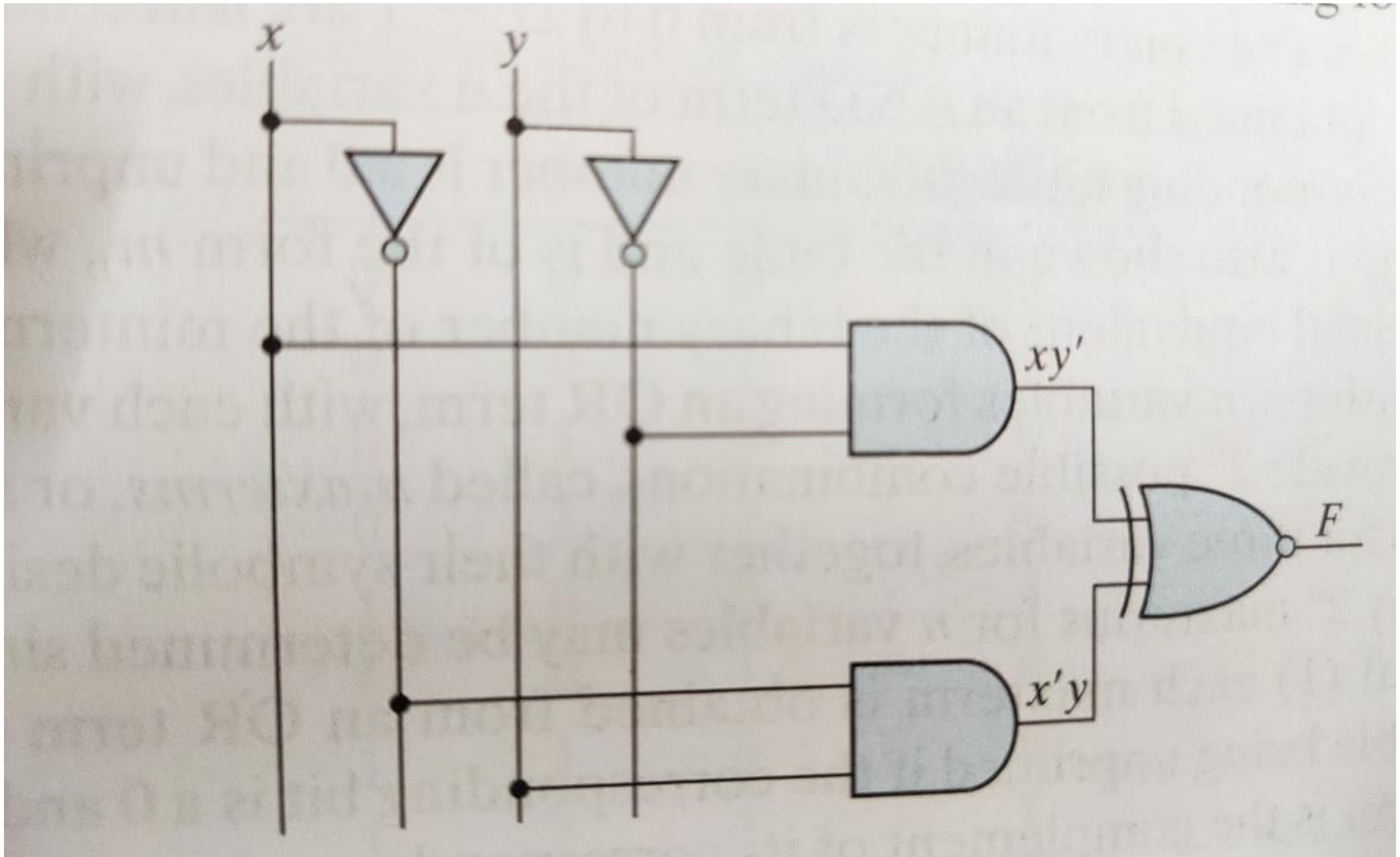
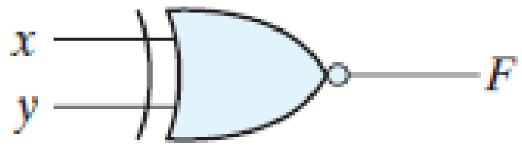


FIGURE PE2.4

Ex-NOR

Exclusive-NOR
or
equivalence



$$F = xy + x'y' = (x \oplus y)'$$

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

Answer

- (xy’)(x’y)+(xy’)’(x’y)’
- 0+(x’+y)(x+y’)
- xx’+x’y’+xy+yy’
- xy+x’y’
- Draw truth table for xy+x’y’

X	Y	X’	Y’	XY	X’Y’	XY+X’Y’

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 DeMorgan's theorems
- DeMorgan's theorems can be extended to three or more variables. The three-variable form of the first DeMorgan's theorem is derived as follows, from postulates and theorems.

Complement of a Function

$$
\begin{aligned}
(A + B + C)' &= (A + x)' && \text{let } B + C = x \\
&= A'x' && \text{by theorem 5(a) (DeMorgan)} \\
&= A'(B + C)' && \text{substitute } B + C = x \\
&= A'(B'C') && \text{by theorem 5(a) (DeMorgan)} \\
&= A'B'C' && \text{by theorem 4(b) (associative)}
\end{aligned}
$$

DeMorgan's theorems for any number of variables resemble the two-variable case in form and can be derived by successive substitutions similar to the method used in the preceding derivation. These theorems can be generalized as follows:

$$(A + B + C + D + \cdots + F)' = A'B'C'D' \ldots F'$$

$$(ABCD \ldots F)' = A' + B' + C' + D' + \cdots + F'$$

The generalized form of DeMorgan's theorems states that the complement of a function is obtained by interchanging AND and OR operators and complementing each literal.

Complement of a Function

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 theorems as many times as necessary, the complements are obtained as follows:

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$

$$\begin{aligned} F_2' = [x(y'z' + yz)]' &= x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\ &= x' + (y + z)(y' + z') \\ &= x' + yz' + y'z \end{aligned}$$

Complement of a Function

EXAMPLE 2.3

Find the complement of the functions F_1 and F_2 of Example 2.2 by taking their duals and complementing each literal.

1. $F_1 = x'yz' + x'y'z$.
   The dual of F_1 is $(x' + y + z')(x' + y' + z)$.
   Complement each literal: $(x + y' + z)(x + y + z') = F_1'$.
2. $F_2 = x(y'z' + yz)$.
   The dual of F_2 is $x + (y' + z')(y + z)$.
   Complement each literal: $x' + (y + z)(y' + z') = F_2'$.

Example

- Find the complement of the boolean function
- $F=A'BC'+A'B'C$

Answer

- (A’BC’+A’B’C)’
- (A+B’+C).(A+B+C’)
- AA+AB+AC’+B’A+B’B+B’C’+CA+CB+CC’
- A+AB+AC’+B’A+0+B’C’+CA+CB+0
- A+AB+AC’+B’A+B’C’+CA+CB
- A+A(B+B’)+A(C+C’)+B’C’+BC
- A+A+A+B’C’+BC
- A+B’C’+BC

CANONICAL AND STANDARD FORMS

- n variables can form 2^n ($0 \sim 2^n-1$) Minterms, so does Maxterms (Table 2-3).
- Minterms and Maxterms

  Minterms: obtain from an AND term of the n variables, or called standard product.

  Maxterms: n variables form an OR term, or called standard sum.
- Each Maxterm is the complement of its corresponding Minterm, and vice versa.
- A sum of minterms or product of maxterms are said to be in canonical form.

CANONICAL AND STANDARD FORMS

			Minterms		Maxterms	
x	y	z	Term	Name	Term	Name
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

Minterm = standard product

Maxterm = standard sum

CANONICAL AND STANDARD FORMS

- $F1 = x'y'z+xy'z'+xyz$
  $= m_1+m_4+m_7$
- $F1' = x'y'z'+x'yz'+x'yz+xy'z+xyz'$
  → $F1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$
  $= M_0M_2M_3M_5M_6$
- Similarly:
  $F2 = x'yz+xy'z+xyz'+xyz = m_3+m_5+m_6+m_7$
  $= (x+y+z)(x+y+z')(x+y'+z)(x'+y+z) = M_0M_1M_2M_4$
- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in ***canonical form***

x	y	z	F1	F2
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 Form

- From a truth table can express a minterm for each combination of the variables that produces a 1 in a Boolean function, and then taking the OR of all those terms.

<EX.> Upon the table 2-4 that produces 1 in $f_1=1$:

- $F1 = x'y'z+xy'z'+xyz$
  $= m_1+m_4+m_7$
- $F1' = x'y'z'+x'yz'+x'yz+xy'z+xyz'$
  $\rightarrow F1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$
  $= M_0M_2M_3M_5M_6$

x	y	z	F1	F2
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

- Similarly:
  $F2 = x'yz+xy'z+xyz'+xyz = m_3+m_5+m_6+m_7$
  $= (x+y+z)(x+y+z')(x+y'+z)(x'+y+z) = M_0M_1M_2M_4$
- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in ***canonical form***

Sum of Minterms

- Each term must contain all the variables
  - If x is missing, ANDed with $(x + x')$

Ex.2-4 Express the Boolean function $F = A + B'C$ in a sum of minterms.

A→ lost two variables (B, C)

$A = A(B+B') = AB + AB'$ → still missing one variable

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

$=ABC + ABC' + AB'C + AB'C'$

B'C→ lost one variable (A)

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

Combining all terms

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

Convenient expression

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

Notation for Sum of Minterms

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

  $= m_1+m_4+m_5+m_6+m_7$

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

  - $\sum$: ORing of terms

- Can be derived directly from the truth table

A	B	C	F1	
0	0	0	0	
0	0	1	1	$\rightarrow m_1$
0	1	0	0	
0	1	1	0	
1	0	0	1	$\rightarrow m_4$
1	0	1	1	$\rightarrow m_5$
1	1	0	1	$\rightarrow m_6$
1	1	1	1	$\rightarrow m_7$

$m_1, m_4, m_5, m_6, m_7 \rightarrow \sum(1,4,5,6,7)$

Product of Maxterms

- Each term must contain all the variables
  - If x is missing, ORed with xx' and apply distributive law

Ex.2-5 Express the Boolean function F= xy + x'z in a product of maxterm form.

using distributive law→ $F = xy + x'z = (xy+x')(xy+z)$

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

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

Each OR term missing one variable

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

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

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

Combining all the terms

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

$$= M_0 M_2 M_4 M_5$$

A convenient way to express this function

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

Conversion between Canonical Forms

- The complement of a function = the sum of minterms missing from the original function
  - $F(A,B,C) = \sum(1,4,5,6,7)$
  - $F'(A,B,C) = \sum(0,2,3) = m_0+m_2+m_3$
- From DeMorgan's theorem:
  - $F = (m_0+m_2+m_3)' = m_0'm_2'm_3' = M_0M_2M_3 = \prod(0,2,3)$
    - $m_j' = M_j$ are shown in Table 2-3
- To convert from one canonical from to another:
  - Interchange the symbol $\sum$ and $\prod$
  - List those numbers missing from the original form

Example 2-5 by Conversion

- $F = xy + x'z = x'y'z + x'yz + xyz' + xyz$

x	y	z	F	
0	0	0	0	
0	0	1	1	→ m_1
0	1	0	0	
0	1	1	1	→ m_3
1	0	0	0	
1	0	1	0	
1	1	0	1	→ m_6
1	1	1	1	→ m_7

$m_1, m_3, m_6, m_7 \rightarrow \Sigma(1,3,6,7)$

- The missing numbers are 0, 2, 4, 5
  - $F = \prod(0,2,4,5)$

Standard forms

- Another way to express Boolean functions is in standard form.

1. Sum of products(SOP): $F_1 = y' + xy + x'yz'$
2. Product of sums(POS): $F_2 = x(y' + z)(x' + y + z')$

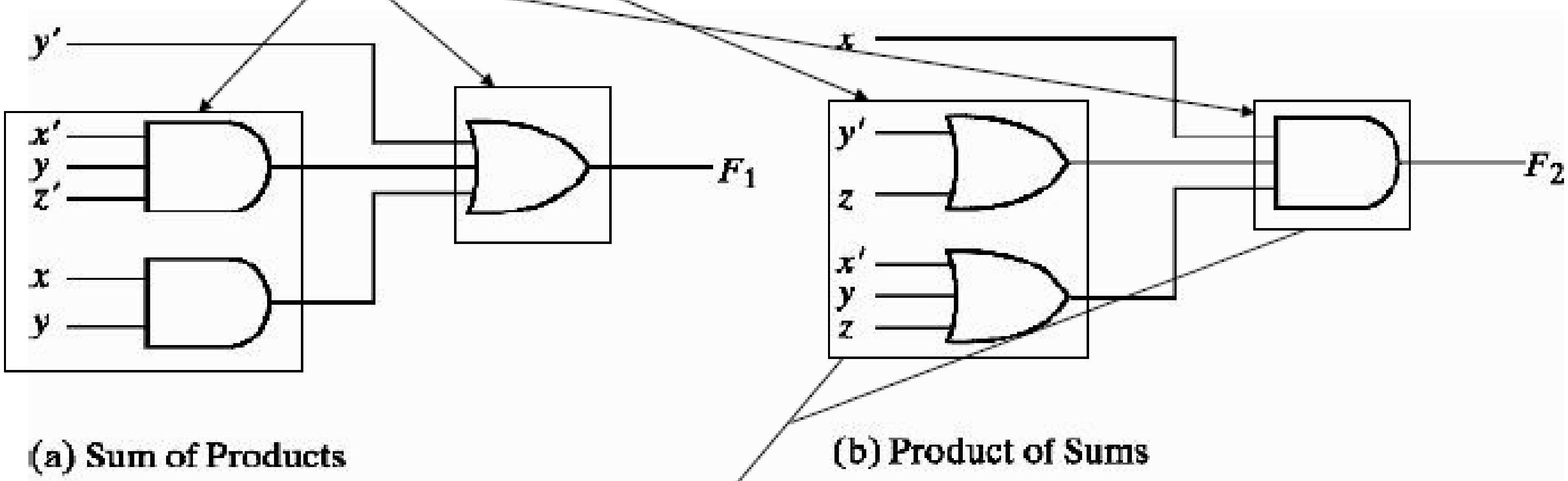
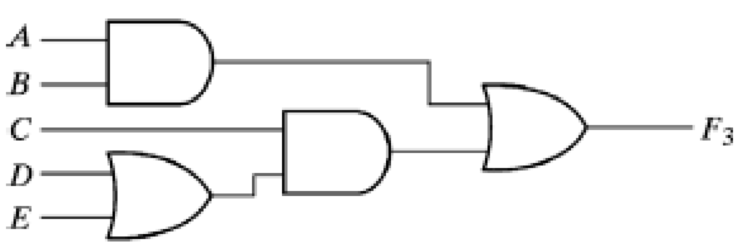


Fig. 2-3 Two-level implementation

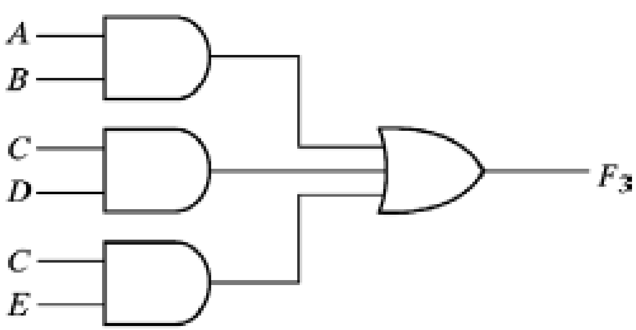
- Standard forms : not required to have all variables in each term

Non-Standard Forms

- Neither in sum of products nor in product of sums
  - F3 = AB + C(D+E) ………… non-standard form
  - F3 = AB + CD + CE ……….. standard form
- Results in a multi-level gating structure
- In general, two-level implementations are preferred
  - Produce the least amount of delay from inputs to outputs



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



(b) $AB + CD + CE$

Fig. 2-4 Three- and Two-Level implementation