Chapter 2
Boolean Algebra and Logic Gates

Definitions

Theorems

Functions

Canonical and Standard Forms

Operations

Gates

Integrated Circuits

Duality

- **The** duality principle **states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged.**
  - **The Huntington postulates have been listed in pairs and designed as part (a) and part (b).**
  - **If the dual of an algebraic equation is required, we interchange the OR and AND operators and replace 1's by 0's and 0's by 1's.**
  - **Example:**

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

Duality (More Examples)

P2	$x+0 = x$	$x\,1 = x$
p5	$x+x' = 1$	$x\,x' = 0$
T1	$x + x = x$	$x\,x = x$
T2	$x + 1 = 1$	$x\,0 = 0$
T3, involution	$(x')' = x$	
p3	$x+y = y+x$	$xy = yx$
T4	$x+(y+z)=(x+y)+z$	$x(yz) =(xy)z$
P4	$x(y+z)=xy+xz$	$x+yz=(x+y)(x+z)$
T5, DeMorgan	$(x+y)' =x'y'$	$(xy)'=x'+y'$
T6, absorption	$x+xy = x$	$x(x+y) =x$

Basic Theorems

- **The following six (6) theorems can be deduced from the Huntington postulates:**
  - **Theorem 1(a):** $x + x = x$
  - **Theorem 1(b):** $x \cdot x = x$
  - **Theorem 2(a):** $x + 1 = 1$
  - **Theorem 2(b):** $x \cdot 0 = 0$
  - **Theorem 3 (involution):** $(x')' = x$
  - **Theorem 4(a) (associative):** $x + (y + z) = (x + y) + z$
  - **Theorem 4(b) (associative):** $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
  - **Theorem 5(a) (DeMorgan):** $(x + y)' = (x' \cdot y')$
  - **Theorem 5(b) (DeMorgan):** $(x \cdot y)' = (x' + y')$
  - **Theorem 6(a) (absorption):** $x + x \cdot y = x$
  - **Theorem 6(b) (absorption):** $x \cdot (x + y) = x$

Proving Theorem 1(a)

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

Proving Theorem 1(b)

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

Proving Theorem 2(a)

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

- Theorem 2(b) can be proved by duality:

$x \cdot 0 = 0$

Proving Theorem 6(a)

$x + x \cdot y = x$		
$= x \cdot 1 + x \cdot y$	by postulate:	2(b)
$= x \cdot (1 + y)$		4(a)
$= x \cdot (y + 1)$		3(a)
$= x \cdot 1$	by theorem:	2(a)
$= x$	by postulate:	2(b)

Proving Theorem 6(b)

$x \cdot (x + y) =$		
$= (x + 0) \cdot (x + y)$	by postulate:	2(a)
$= x + 0 \cdot y$		4(b)
$= x + y \cdot 0$		3(b)
$= x + 0$	by theorem:	2(b)
$= x$	by postulate:	2(a)

DeMorgan’s Theorem

- **With truth table the validity of first DeMorgan’s Theorem can be shown**
- **(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

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

Boolean Functions

- **A** Boolean function **is an expression described by:**
  - **binary variables**
  - **constants 0 and 1**
  - **logic operation symbols**
- **For a given value of the binary variables the result of the function can either be 0 or 1.**
- **An example function:**
  - $F_1 = x + y'z$
  - **F_1 is equal to 1 if x is equal to 1 or if both y' and z equal to 1. F_1 is equal to 0 otherwise**

Function as a Truth Table

- A Boolean function can be represented in a truth table.
  - A truth table is a list of combinations of 1's and 0's assigned to the binary variables and and a column that shows the value of the function for each binary combination

x	y	z		$F_1 = x + y'z$
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

Function as a Gate Implementation

- **A** Boolean function **can be transformed from an algebraic expression into** circuit diagram **composed of** logic gates**.**
  - **F_1 = x + y'z**
  - **The logic-circuit diagram for this function is shown below:**

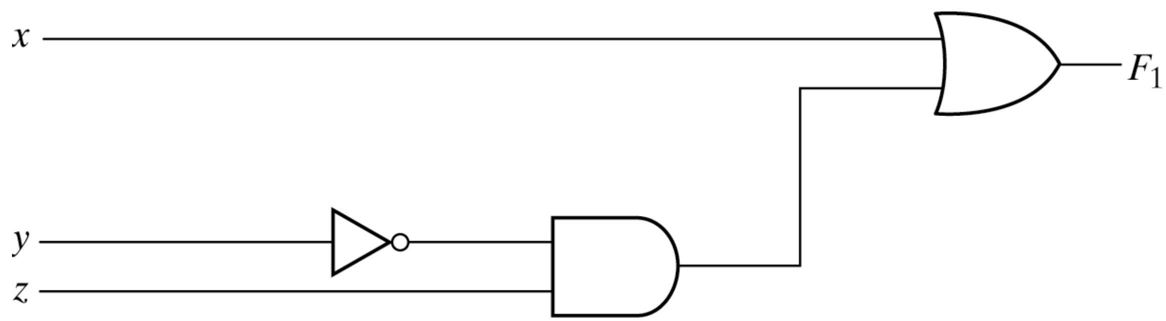
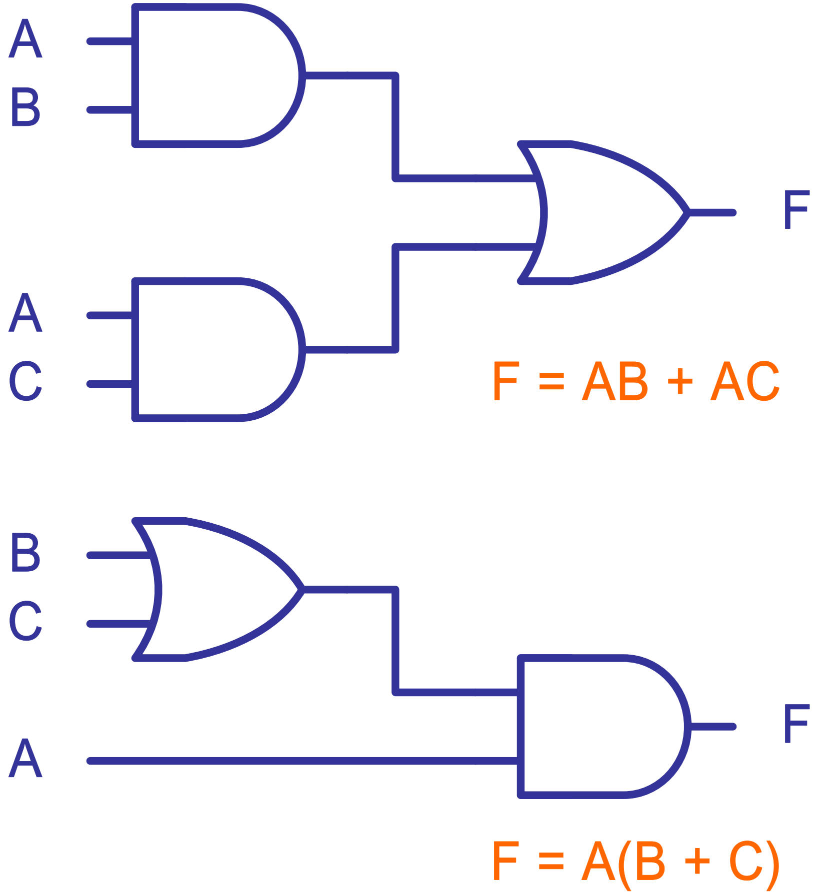


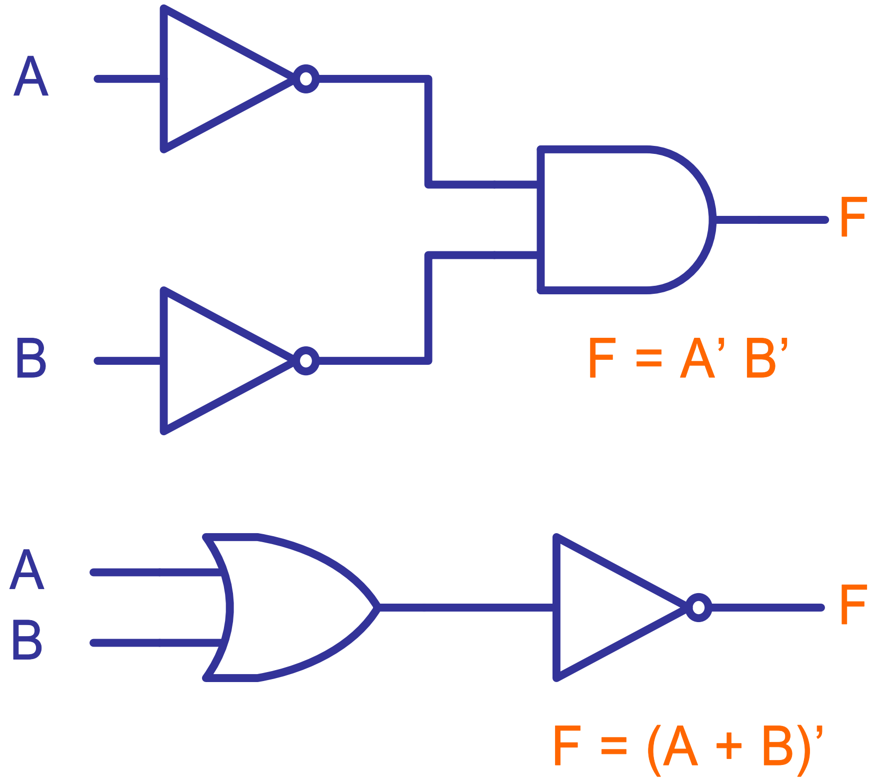
Fig. 2-1 Gate implementation of $F_1 = x + y'z$

Gate Implementation (Examples)



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

Gate Implementation (Examples)



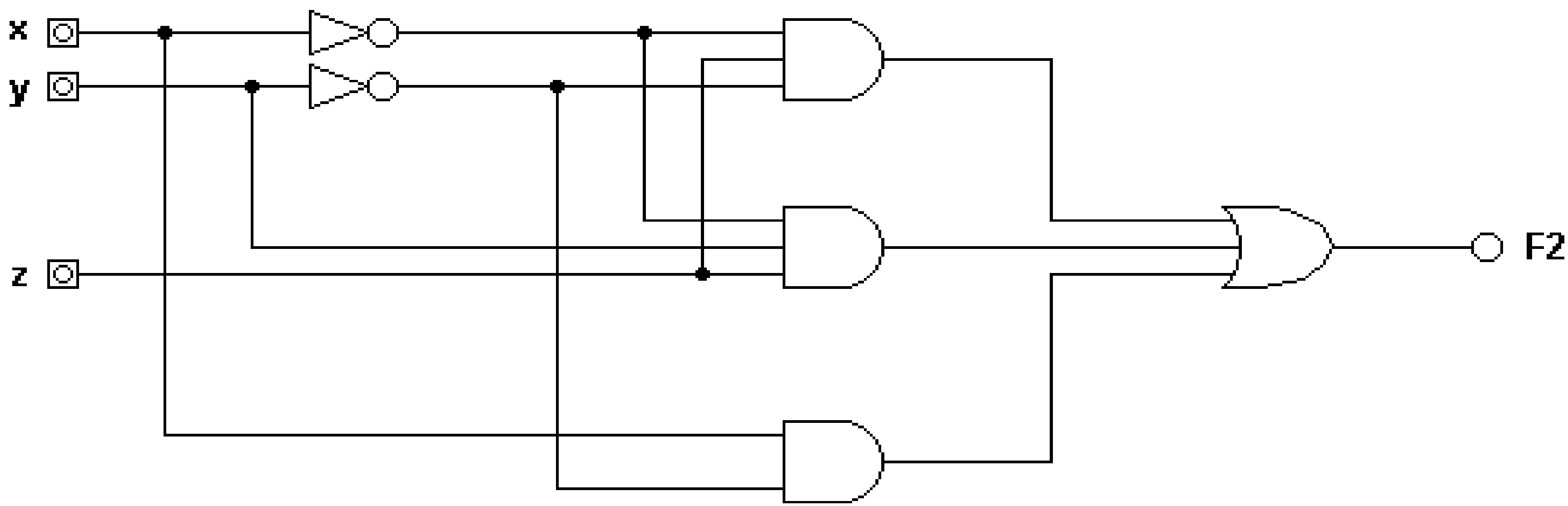
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Minimization

- **Functions in algebraic form can be represented in various ways.**
  - **Remember the postulates and theorems that allows us to represent a function in various ways.**
- **We must keep in mind that the algebraic expression is representative of the gates and circuitry used in a hardware piece.**
  - **We want to be able to minimize circuit design to reduce cost, power consumption, and package count, and to increase speed.**
- **By manipulating a function using the postulates and theorems, we may be able to minimize an expression.**

Non-Minimized Function

- The following is an example of a non-minimized function:
  - $F_2 = x'y'z + x'yz + xy'$

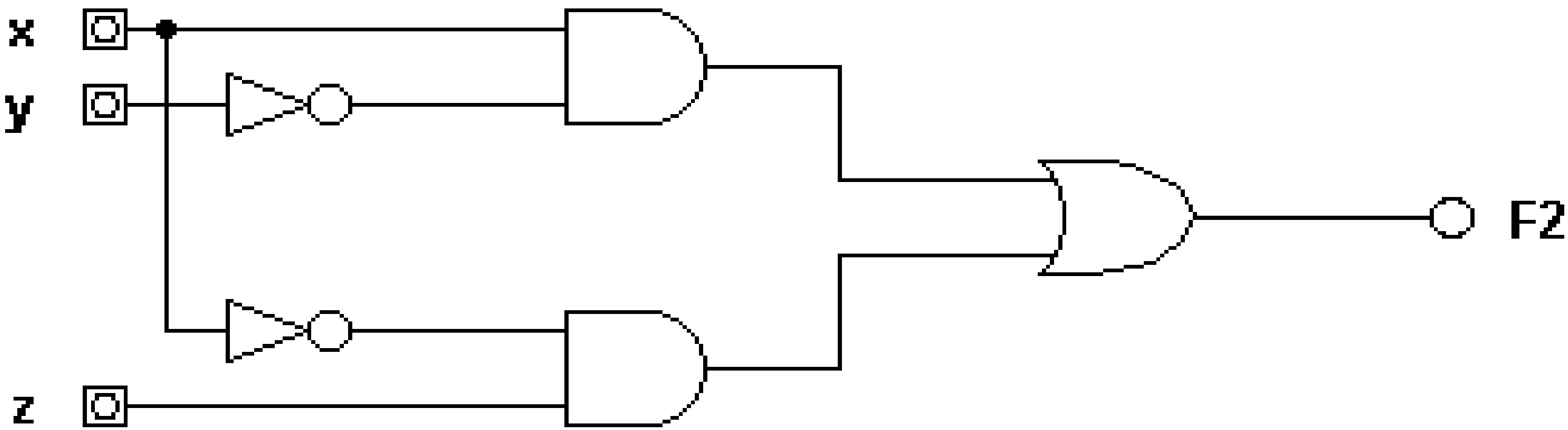


Minimization of the F_2

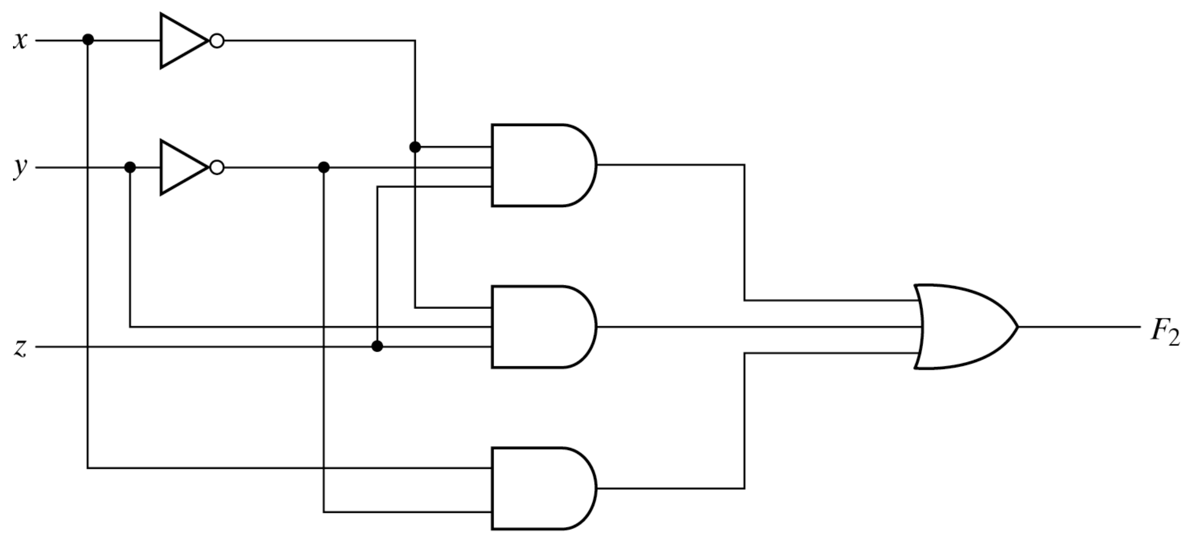
- **The function can be minimized as follows:**

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

= x'z · (y' + y) + xy'	**by postulate:**	**4(a)**
= x'z · 1 + xy'		**5(a)**
= x'z + xy'		**2(b)**

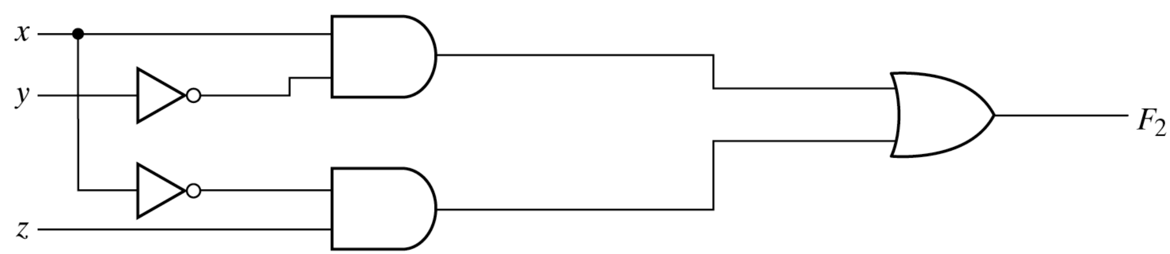


Implementation of Boolean Function



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

- Minimized Function



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

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