Logic Design

Dr. Yosry A. Azzam

Boolean Algebra & Logic Gates

Chapter 2

- Agenda
  - Basic definitions
  - Proprieties of Boolean Algebra
  - Boolean Functions
  - Canonical and standard Forms
- Reading
  - Mano: Ch 2

- Objectives
  - Understanding the canonical forms
  - Maxterms and minterms
  - Simplifying Boolean expression and functions

Boolean Algebra

- The mathematical system that operate with binary values (George Boole (1854))
- Is the algebraic structure defined on a set of elements with two binary operators + (OR) and . (AND) provided that the following Huntington postulates are satisfied:

- Reading :

Mano Chapter 2

Huntington Postulates

1. (a) closures w.r.t operator +

   (b) closures w.r.t operator .

2. (a) An identity element w.r.t + (0): $x + 0 = 0 + x = x$

   (b) An identity element w.r.t . (1) : $x \cdot 1 = 1 \cdot x = x$

3. (a) Commutative w.r.t +: $x + y = y + x$

   (b) Commutative w.r.t . : $x \cdot y = y \cdot x$

4. (a) . is distributive over +: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

   (b) + is distributive over .: $x + (y \cdot z) = (x + y) \cdot (x + z)$

5. The complement of x is such that (a) $x + x' = 1$ and

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

6. There exist at least 2 elements x, y such that $x \neq y$

Two-Valued Boolean Algebra

- Is defined on a set of 2 elements, $B = \{0,1\}$ with rules for the two binary operators + and . as shown in the following tables:
- These rules are exactly the same as the AND, OR, and Not operations.

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

Huntington Postulates on the Two-Valued Set and the two Binary Operators

1. closures is obvious from the tables as the result of each operation is either 0 or 1
2. (a) An identity element w.r.t + (0): $0 + 0 = 0$, $0+1 = 1+0 =1$

   (b) An identity element w.r.t . (1) : $1 . 1 = 1$, $1 . 0 = 0 . 1 =0$
3. The Commutative laws are obvious
4. (a) the distributive law : $x . (y + z) = (x . y) + (x . z)$ can be verified by the truth table of all possible values of x, y and z

   (b) the distributive law: $x + (y . z) = (x + y) . (x + z)$ can also be verified by the truth table of all possible values of x, y and z
5. The complement of x (a) $x+x' =1$: $0+0'=0+1=1$ and $1+1' =1+0 =1$

   (b) $x . x' =0$: $0.0'=0.1=0$ and $1.1' =1.0 =0$
6. The two valued Boolean algebra has two distinct elements 1 and 0 with $1 \neq 0$

Truth Tables Verification

x	y	z	$y+z$	$x.(y+z)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

x	y	z	$x.y$	$x.z$	$(x.y)+(x.z)$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Property of Boolean Algebra 1

- Closure
  - Obtaining a unique elements (which are the members of Boolean set)
- Associative Law
  - (X*Y)*Z = X*(Y*Z)
  - (X+Y)+Z = X+(Y+Z)
- Commutative Law
  - X*Y = Y*X
  - X+Y = Y+X

Property of Boolean Algebra 2

- Identity Element
  - $1.X=X.1=X$
  - $0+X=X+0=X$
- Inverse
  - $X.\ X'=0$
  - $X+X'=1$
- Distributive Law

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

Basic Theorems

- Duality
  - Huntington (1904) postulate that of an algebraic expression, we can simply interchange OR and AND operator and replace 1 by 0 and 0 by 1

  - [*We will discuss this more in future sessions when you enter the realm of digital design*]

Postulates and Theorems of Boolean Algebra

Postulate 2		(a) x + 0=x	(b) x .1=x
Postulate 5		(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	
Postulate 3	Commutative	(a) x +y=y +x	(b) x y=y x
Theorem 4	Associative	(a) x +(y +z)=(x +y) +z	(b) x (y z)=(x y) z
Postulate 4	Distributive	(a) x (y +z)= x y+ x z	(b) x +y z=(x +y) (x +z)
Theorem 5	DeMorgan	(a) (x +y)’=x’ y’	(b) (x y)’=x’ + y’
Theorem 6	Absorption	(a) x + x y= x	(b) X (x +y)=x

Proof of some basic theorems

- **Theorem 1(a):** $x+x=x$.
- Proof:
- $x + x = (x+x).1$ (postulate 2b)=

  $= (x + x) . (x + x')$ (postulate 5a

  $= x + xx' = x$ (postulate 4b)
- **Theorem 1(b):** $x . x=x$
- Proof:
- $x . x = xx +0 = x x + x x' = x . (x+x')= x . 1 = x$

Proof of some basic theorems

- **Theorem 2(a):):** $x + 1 = 1$.
- Proof:
- $x+1 = 1 . (x+1) = (x+x')(x+1) = x+x' . 1 = x+x' = 1$
- **Theorem 2(b):** $x . 0 = 0$**.**
- Proof: by duality $x.0=0$

Proof of some basic theorems

- **Theorem 6(a):** $x+xy=x$.
- Proof:
- $x + x y = x.1 + xy = x(1+y) = x (y+1) = x.1 = x$

- **Theorem 6(b):** $x(x+y)=x$.
- Proof: By duality

x	y	$x\ y$	$x + x\ y$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

DeMorgan's Theorem

(a) $(x + y)' = x'y'$ (b) $(xy)' = x' + y'$

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

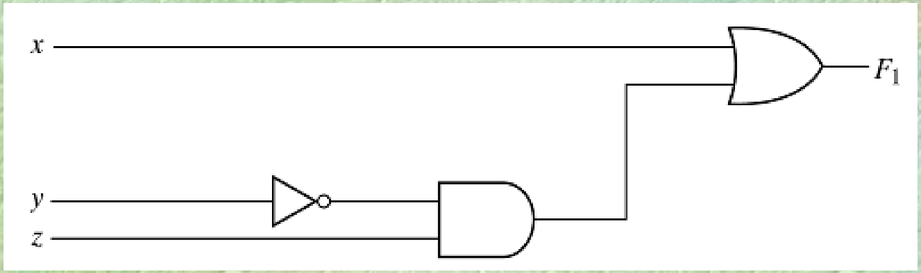
x'	y'	$x'y'$
1	1	1
1	0	0
0	1	0
0	0	0

Operator precedence

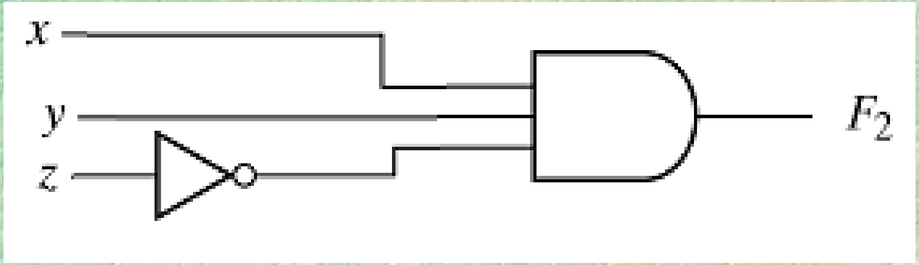
- Parenthesis
- NOT
- AND
- OR

Boolean Functions 1

- $F_1=x+y'.z$

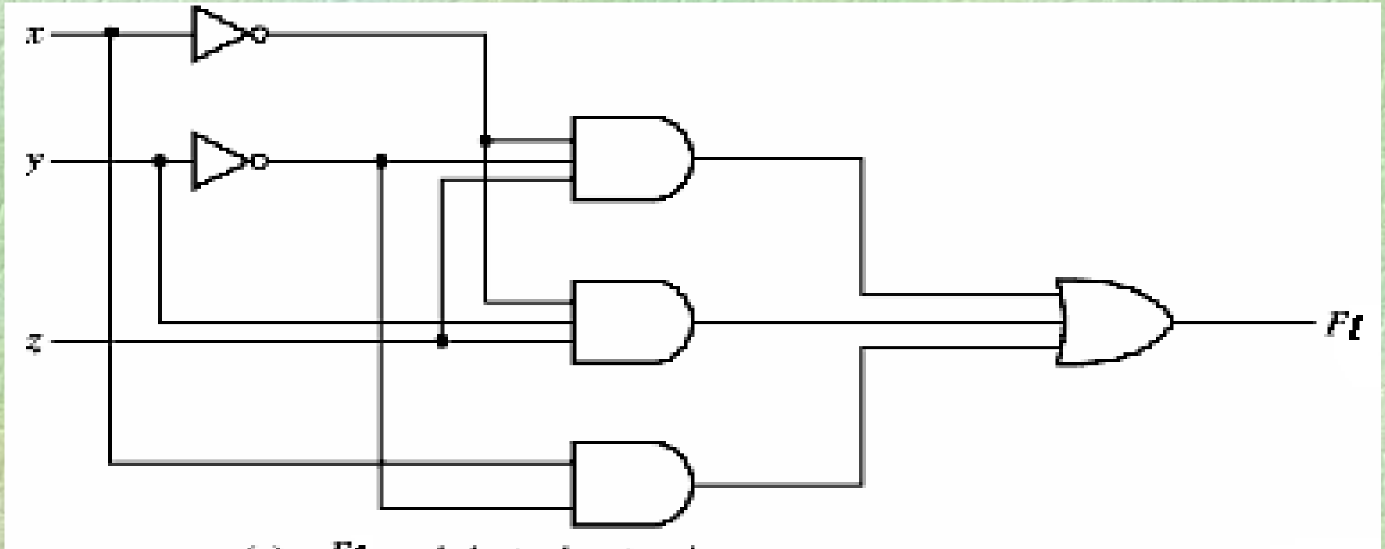


- $F_2=x.y.z'$

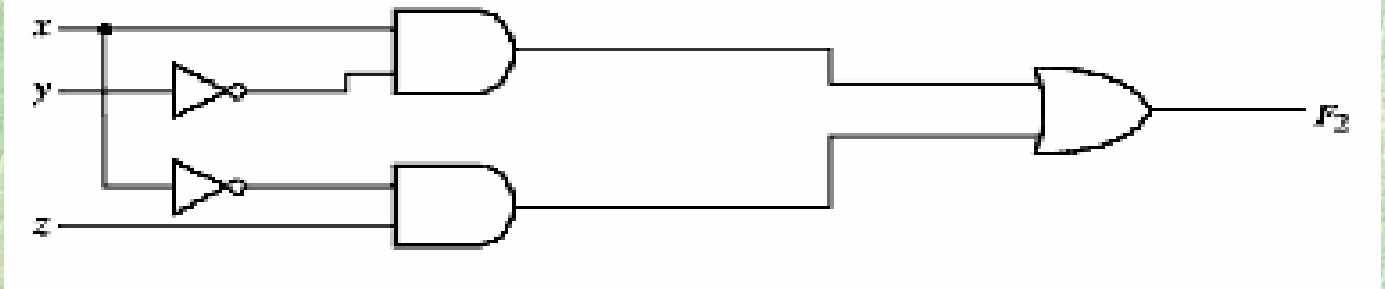


X	**Y**	**Z**	$\mathbf{F_1}$	$\mathbf{F_2}$
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

Boolean Functions 2



(a) $F1 = x'y'z + x'yz + xy'$



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

HW : give the truth table of Functions F1 and F2

Algebraic Manipulation

- **Simplify the following Boolean function to a minimum number of literals:**

(1) $F(x, y) = x(x' + y)$

(2) $F(x, y) = x + x'y$

(3) $F(x, y) = (x + y)(x + y')$

(4) $F(x, y, z) = xy + x'z + yz$

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

- ***Solution:***

(1) $F(x, y) = x x' + x y = 0 + x y = x y$

(2) $F(x, y) = (x + x')(x + y) = 1.(x + y) = x + y$

(3) $F(x, y) = x + y y' = x + 0 = x$

or

$= x + x y' + y x + y y' = x(1 + y' + y) = x . 1 = x$

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

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

(5) $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$ **by duality from 4**

Complement of a Function

- De Morgan
  - $(X+Y)' = X'.Y'$
- $(A+B+C)'$
  - $= (A+X)'$ with $X=B+C$
  - $= A'X'$ (De Morgan)
  - $= A'.(B+C)'$
  - $= A'.(B'C')$ (De Morgan)
  - $= A'.B'.C'$ (Associative)

Complement of a Function

- Example:

Find the complement of the functions:

(1) $F=A+B+C$.

(2) $F1=x' y z' + x' y' z$

(3) $F2= x (y' z' +y z)$

- Solution:

(1) $F'=(A+B+C)'=A'B'C'$

(2) $F1'=(x' y z' + x' y' z)' = (x' y z')' (x' y' z)'$

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

(3) $F2'=x' + (y + z) (y' + z')$

Canonical Forms

- Binary Variable can either be:
  - Normal Form (x)
  - Complement Form (x′)
  - For 3 binary variables (x, y, z) there are 8 possibility of response for AND operation (minterms) and 8 possibility of OR operation (maxterms)

Minterms #1

- "Minterms" or "Standard Product"
- 2 Binary Variable (X & Y) will form $2^{n=2}$ Minterms
  - $X'Y'$, $X'Y$, XY', XY
  - AND Terms ("product")
- Any Boolean function can be expressed as a **sum of Minterms**

Minterms and Maxterms for three binary variables:

			Minterms		Maxterm	
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

Function of three variables

x	*y*	*Z*	*Function f1*	*Function 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

F1=x’ y’ z + x y’ z’ + x y z = m1+m4+m7
F2=x′ y z + x y′ z + x y z′ + x y z = m3 + m5 + m6 + m7

F'_1= x′ y′ z′ + x′ y z′ + x′ y z + x y′ z + x y z′
The complement of F'_1=(x+y+z)(x+y′+z)(x+y′+z′)(x′+y+z′) (x′ +y′+z)

$=F_1= M_0.M_2.M_3.M_5.M_6$

Minterms #2

- Example 2-4 (page 46)

  Sum of Minterms

  - $F = A + B'C$
  - $F = A (B + B') + B'C$
  - $F = AB + AB' + B'C$
  - $F = AB (C + C') + AB' (C + C') + B'C$
  - $F = ABC + ABC' + AB'C + AB'C' + B'C$
  - $F = ABC + ABC' + AB'C + AB'C' + (A + A') B'C$
  - $F = ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C$
  - $F = ABC + ABC' + AB'C + AB'C' + A'B'C$
  - $F = m7 + m6 + m5 + m4 + m1$

    (Table 2-5, page 47)
  - $F(A, B, C) = \sum(1, 4, 5, 6, 7)$

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

Maxterms

- “Maxterms” or “Standard Sums”
- 2 Binary Variables (X & Y) will form $2^{n=2}$ Maxterms
  - $X'+Y'$, $X'+Y$, $X+Y'$, $X+Y$
  - OR Terms (“sum”)
- Any Boolean function can be expressed as a **product of Maxterms**
  - $F2=(x+y+z)(x+y+z')(x+y'+z)(x'+y+z)$
  - $F2=M_0 . M_1 . M_2 . M_4=\Pi(0,1,2,4)$

Minterms & Maxterms #1

- Minterms
  - $X'Y'$, $X'Y$, XY', XY
- Maxterms
  - $X'+Y'$, $X'+Y$, $X+Y'$, $X+Y$
- Each Maxterms is the complement of its corresponding Minterms & vice versa
  - Remember **De Morgan**?
  - Take a look at the next slide

Minterms & Maxterms #2

			Minterms		Maxterms	
x	y	z	Term	Desig.	Term	Desig.
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	$x' y z'$	m_2	$x + y' + z$	M_2
0	1	1	$x' y z$	m_3	$x + y' + z'$	M_3
1	0	0	$x y' z'$	m_4	$x' + y + z$	M_4
1	0	1	$x y' z$	m_5	$x' + y + z'$	M_5
1	1	0	$x y z'$	m_6	$x' + y' + z$	M_6
1	1	1	$x y z$	m_7	$x' + y' + z'$	M_7

Conversion between canonical forms

F=xy + x'z

The function F expressed in sum of minterms is: $F(x,y,z)= \sum(1,3,6,7)$

The missing terms are 0,2,4,5

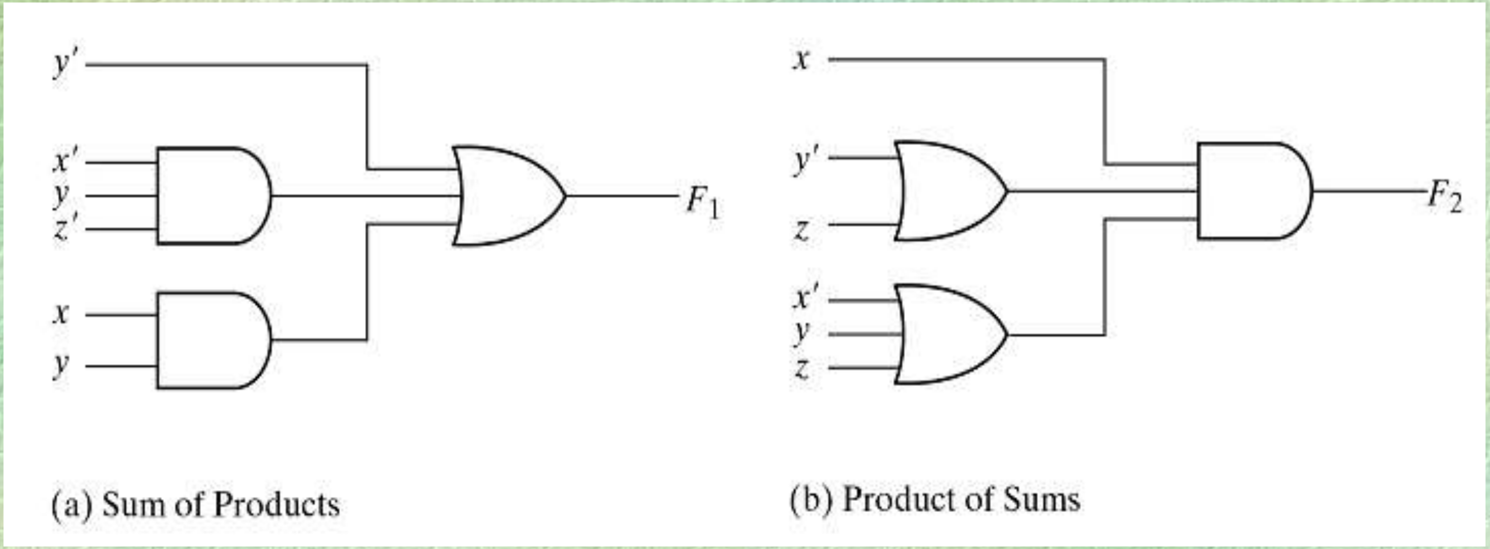
Then, The function F expressed in product of maxterm is:

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

X	Y	Z	F_1
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

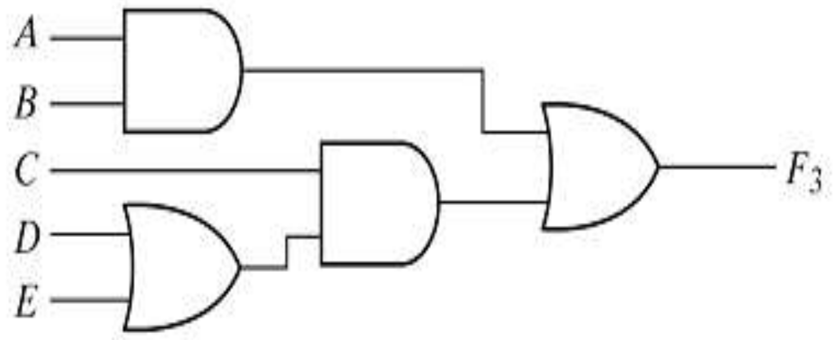
Standard Forms

- Doesn’t have to consists of all variables
  - Sum of Products : $F_1 = y' + xy + x'yz'$
  - Products of Sum : $F_2 = x\,(y' + z)\,(x' + y + z')$

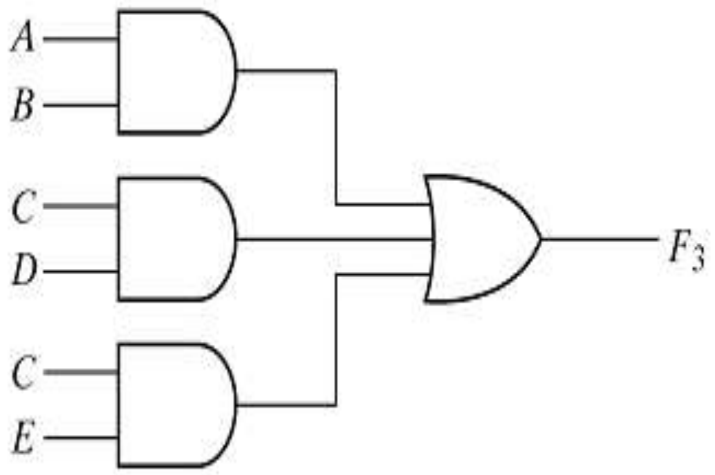


(a) Sum of Products
(b) Product of Sums

Implementation with two and three levels



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



(b) $AB + CD + CE$

Truth table for 16 functions of 2 binary variables

X	Y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

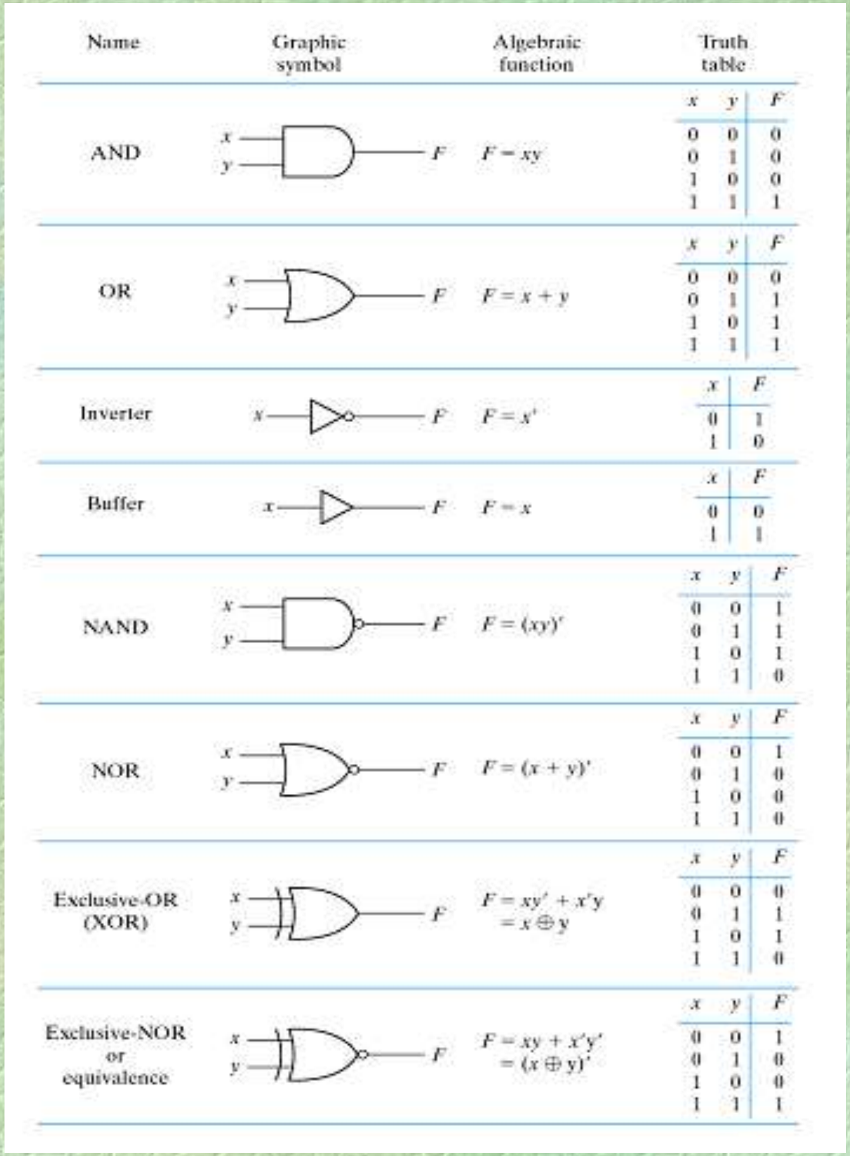
Other Logic Operations

Boolean Operator	Operator Symbol	Name	Comments
F0 = 0		Null	Binary Constant 0
F1 = xy	x.y	AND	x and y
F2 = xy′	x/y	Inhibition	x but not y
F3 = x		Transfer	x
F4 = x′y	y/x	Inhibition	y but not x
F5 = y		Transfer	y
F6 = xy′ + x′y	x⊕y	Exclusive OR	x or y but not both
F7 = x + y	x+y	OR	x or y
F8 = (x + y)′	x↓y	NOR	Not OR
F9 = xy + x′y′	(x⊕y)′	Equivalence	x equals y
F10 = y′	y′	Complement	Not y
F11 = x + y′	X ⊂ y	Implication	If y then x
F12 = x′	x′	Complement	Not x
F13 = x′ + y	X ⊃ y	Implication	If x then y
F14 = (xy)′	x↑y	NAND	Not AND
F15 = 1		Identity	Binary Constant 1

Common Logic Gates

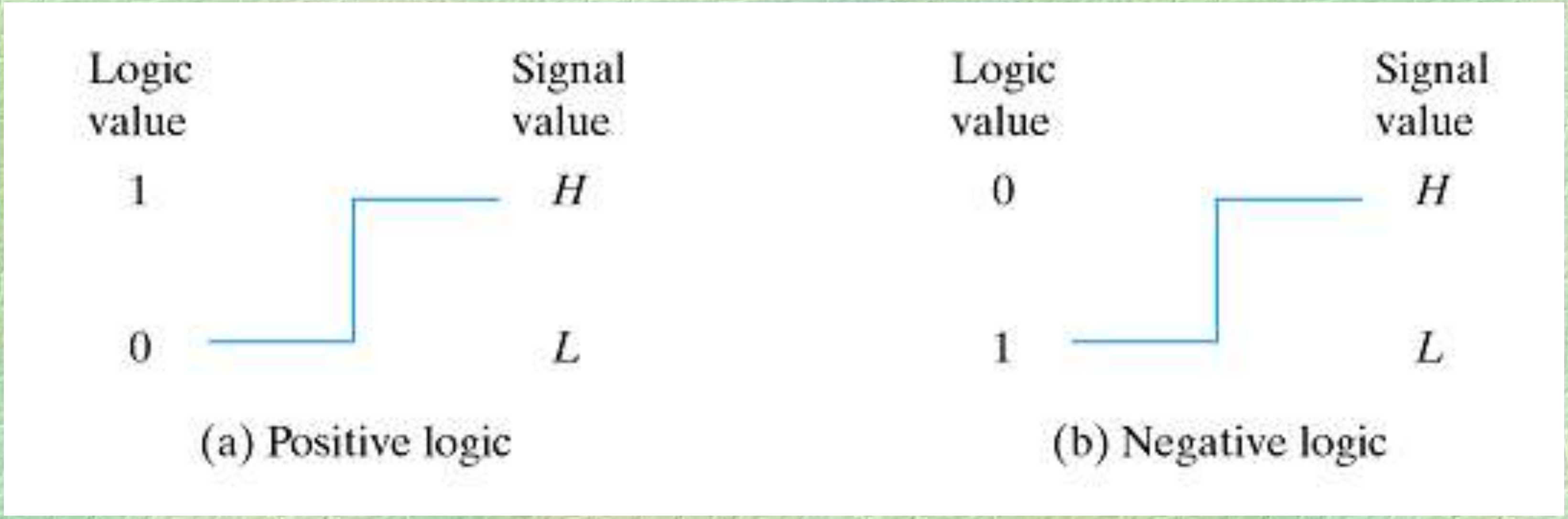
- AND
- OR
- Inverter
- Buffer
- NAND
- NOR
- XOR
- XNOR

Figure 2-5 (page 54)



Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
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

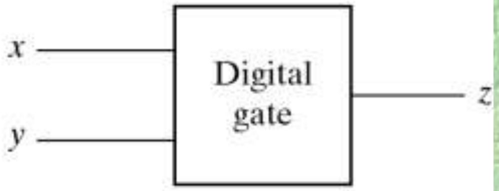
Positive and negative logic



Positive and negative logic (*Contd.*)

x	y	F
L	L	L
L	H	L
H	L	L
H	H	H

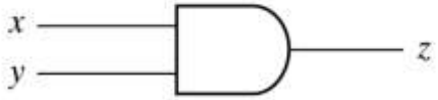
(a) Truth table with H and L



(b) Gate block diagram

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

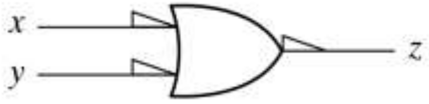
(c) Truth table for positive logic



(d) Positive logic AND gate

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

(e) Truth table for negative logic



(f) Negative logic OR gate

Fig. 2-10 Demonstration of positive and negative logic

H

Integrated Circuits (ICs)

- Levels of Integration

1- Small-Scale Integration (SSI)

Number of gates are < 10

2- Medium-Scale Integration (MSI)

Number of gates are from 10 to 1000

3- Large-Scale Integration (LSI)

contains thousands of gates in a single package

4- Very Large-Scale Integration (VLSI)

contains hundred of thousands of gates in a single package

Digital Logic families

- Digital ICs are classified not only by their complexity or logical operations but also by the specific circuit technology to which they belong which is referred to as digital logic family.
- The basic circuits in each technology is a NAND, NOR, or inverter gate.
- The electronic components employed in the construction of the basic circuit are usually used to name the technology.

Digital Logic families (*Contd.*)

- The most popular logic families are:

1- **TTL** **Transistor-Transistor Logic**

is being in operation for a long time and is a standard

2- **ECL** **Emitter-Coupled Logic**

has an advantages in systems requiring high speed operation

3- **MOS** **Metal-oxide Semiconductor**

suitable for circuits that need high component density

4- **CMOS** **Complementary Metal-oxide Semiconductor**

preferable in systems requiring low power consumption as in VLSI

Digital Logic families parameters

- These are the parameters that are evaluated and compared for different families:

1- Fan-out: the number of standard loads that the output of a gate can drive.

2- Fan-in: the number of inputs available in the gate.

3- Power dissipation: power consumed by the gate that must be available from the power supply.

4- Propagation delay: average transition delay time for the signal to propagate from input to output.

5- Noise margin: is the maximum external noise voltage added to an input signal that does not cause an undesirable change in the circuit output

Computer-Aided Design (CAD)

- Software programs that support computer-based representation and aid in the development of digital hardware by automating the design process.

Exercises

- HW :

  Problems 2-1 until 2-23 Mano