UNIVERSIDAD POLITÉCNICA
TAIWAN – PARAGUAY
臺灣-巴拉圭科技大學

Boolean algebra fundamentals

UNIVERSIDAD POLITÉCNICA
TAIWAN – PARAGUAY
臺灣-巴拉圭科技大學

Fundamentals of Boolean Algebra

- Boolean algebra is defined with a set of ***elements***, a set of ***operators***, and a number of *axioms* and *postulates*.

- *A set of elements* is any collection of objects. If S is a set, and x and y are certain objects, then $x \in S$ means that x is a member of the set S and $y \notin S$ means that y is not an element of S.

Fundamentals of Boolean Algebra

- ***Basic Postulates***
- ***Postulate 1 (Definition)***: A Boolean algebra is a closed algebraic system containing a set *K* of two or more elements and the two operators · and +. For every pair of elements of K, the binary operator specifies a rule for obtaining a unique element of K.
- ***Postulate 2 (Existence of 1 and 0 element)***:

  (a) $a + 0 = a$ (identity for +), (b) $a \cdot 1 = a$ (identity for ·)
- ***Postulate 3 (Commutativity)***:

  (a) $a + b = b + a$, (b) $a \cdot b = b \cdot a$
- ***Postulate 4 (Associativity)***:

  (a) $a + (b + c) = (a + b) + c$ (b) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- ***Postulate 5 (Distributivity)***:

  (a) $a + (b \cdot c) = (a + b) \cdot (a + c)$ (b) $a \cdot (b + c) = a \cdot b + a \cdot c$
- ***Postulate 6 (Existence of complement)***:

  (a) $a + \overline{a} = 1$ (b) $a \cdot \overline{a} = 0$

<u>Precedence</u>

()

'

·

+

Note: Normally · is omitted.

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY
臺灣–巴拉圭科技大學

Fundamentals of Boolean Algebra

- ***Fundamental Theorems of Boolean Algebra***

- ***Theorem 1 (Idempotency)***:

  (a) $a + a = a$ (b) $aa = a$

- ***Theorem 2 (Null element)***:

  (a) $a + 1 = 1$ (b) $a0 = 0$

- ***Theorem 3 (Involution)***

$$\overline{\overline{a}} = a$$

- ***Properties of 0 and 1 elements***:

OR	AND	Complement
$a + 0 = a$	$a0 = 0$	$0' = 1$
$a + 1 = 1$	$a1 = a$	$1' = 0$

Fundamentals of Boolean Algebra

- ***Theorem 4 (Absorption)***

  (a) $a + ab = a$ (b) $a(a + b) = a$

- ***Examples***:
  - $(X + Y) + (X + Y)Z = X + Y$ [T4(a)]
  - $AB'(AB' + B'C) = AB'$ [T4(b)]

- ***Theorem 5***

  (a) $a + a'b = a + b$ (b) $a(a' + b) = ab$

- ***Examples***:
  - $B + AB'C'D = B + AC'D$ [T5(a)]
  - $(X + Y)((X + Y)' + Z) = (X + Y)Z$ [T5(b)]

Fundamentals of Boolean Algebra

- ***Theorem 6***

  (a) $ab + ab' = a$ (b) $(a + b)(a + b') = a$

- ***Examples***:
  - $ABC + AB'C = AC$ [T6(a)]
  - $(W' + X' + Y' + Z')(W' + X' + Y' + Z)(W' + X' + Y + Z')(W' + X' + Y + Z)$

    $= (W' + X' + Y')(W' + X' + Y + Z')(W' + X' + Y + Z)$ [T6(b)]

    $= (W' + X' + Y')(W' + X' + Y)$ [T6(b)]

    $= (W' + X')$ [T6(b)]

Fundamentals of Boolean Algebra

- ***Theorem 7***

  (a) $ab + ab'c = ab + ac$ (b) $(a + b)(a + b' + c) = (a + b)(a + c)$

- ***Examples***:
  - $wy' + wx'y + wxyz + wxz' = wy' + wx'y + wxy + wxz'$ [T7(a)]

    $= wy' + wy + wxz'$ [T7(a)]

    $= w + wxz'$ [T7(a)]

    $= w$ [T7(a)]
  - $(x'y' + z)(w + x'y' + z') = (x'y' + z)(w + x'y')$ [T7(b)]

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY 臺灣–巴拉圭科技大學

Fundamentals of Boolean Algebra

- ***Theorem 8 (DeMorgan's Theorem)***

  (a) $(a + b)' = a'b'$ (b) $(ab)' = a' + b'$

- Generalized DeMorgan's Theorem

  (a) $(a + b + \ldots z)' = a'b' \ldots z'$ (b) $(ab \ldots z)' = a' + b' + \ldots z'$

- ***Examples***:
  - $(a + bc)' = (a + (bc))'$
    $= a'(bc)'$ [T8(a)]
    $= a'(b' + c')$ [T8(b)]
    $= a'b' + a'c'$ [P5(b)]
  - Note: $(a + bc)' \neq a'b' + c'$

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY
臺灣–巴拉圭科技大學

Fundamentals of Boolean Algebra

- ***More Examples for DeMorgan's Theorem***
  - $(a(b + z(x + a')))' = a' + (b + z(x + a'))'$ [T8(b)]
    $= a' + b'\,(z(x + a'))'$ [T8(a)]
    $= a' + b'\,(z' + (x + a')')$ [T8(b)]
    $= a' + b'\,(z' + x'(a')')$ [T8(a)]
    $= a' + b'\,(z' + x'a)$ [T3]
    $= a' + b'\,(z' + x')$ [T5(a)]

  - $(a(b + c) + a'b)' = (ab + ac + a'b)'$ [P5(b)]
    $= (b + ac)'$ [T6(a)]
    $= b'(ac)'$ [T8(a)]
    $= b'(a' + c')$ [T8(b)]

Fundamentals of Boolean Algebra

Apply DeMorgan's Theorem to these expressions

- (X+Y+Z)’
- (PQ+R)’
- (M+N)’Q’

Fundamentals of Boolean Algebra

- ***Theorem 9 (Consensus)***

  (a) $ab + a'c + bc = ab + a'c$ (b) $(a + b)(a' + c)(b + c) = (a + b)(a' + c)$

- ***Examples***:
  - $AB + A'CD + BCD = AB + A'CD$ [T9(a)]
  - $(a + b')(a' + c)(b' + c) = (a + b')(a' + c)$ [T9(b)]
  - $ABC + A'D + B'D + CD$

$ABC + A'D + B'D + CD$	$= ABC + (A' + B')D + CD$	[P5(b)]
	$= ABC + (AB)'D + CD$	[T8(b)]
	$= ABC + (AB)'D$	[T9(a)]
	$= ABC + (A' + B')D$	[T8(b)]
	$= ABC + A'D + B'D$	[P5(b)]

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY 臺灣–巴拉圭科技大學

Switching Functions

- ***Switching algebra***: Boolean algebra with the set of elements $K = \{0, 1\}$

  If there are *n* variables, we can define 2^{2^n} switching functions.

- Sixteen functions of two variables:

AB	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}
00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
10	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

- A switching function can be represented by a table as above, or by a switching expression as follows:

  $f_0(A,B) = 0,\; f_6(A,B) = AB' + A'B,\; f_{11}(A,B) = AB + A'B + A'B' = A' + B, \ldots$

- Value of a function can be obtained by plugging in the values of all variables:

  The value of f_6 when A = 1 and B = 0 is: $1 \cdot 0' + 1' \cdot 0 = 0 + 1 = 1$.

Truth Tables

- Shows the value of a function for all possible input combinations.
- Truth tables for OR, AND, and NOT:

ab	$f(a,b)=a+b$	ab	$f(a,b)=ab$	a	$f(a)=a'$
00	0	00	0	0	1
01	1	01	0	1	0
10	1	10	0		
11	1	11	1		

Truth Tables

- Truth tables for $f(A,B,C) = AB + A'C + AC'$

ABC	$f(A,B,C)$	ABC	$f(A,B,C)$
000	0	FFF	F
001	1	FFT	T
010	0	FTF	F
011	1	FTT	T
100	1	TFF	T
101	0	TFT	F
110	1	TTF	T
111	1	TTT	T

Algebraic Forms of Switching Functions

- ***Literal***: A variable, complemented or uncomplemented.
- ***Product term***: A literal or literals ANDed together.
- ***Sum term***: A literal or literals ORed together.

- ***SOP (Sum of Products)***:
- ORing product terms
- $f(A, B, C) = ABC + A'C + B'C$

- ***POS (Product of Sums)***
- ANDing sum terms
- $f(A, B, C) = (A' + B' + C')(A + C')(B + C')$

Algebraic Forms of Switching Functions

- A ***minterm*** is a product term in which all the variables appear exactly once either complemented or uncomplemented.
- ***Canonical Sum of Products*** **(*canonical SOP*)**:
  - Represented as a sum of minterms only.
  - ***Example***: $f_1(A,B,C) = A'BC' + ABC' + A'BC + ABC$
- Minterms of three variables:

Minterm	Minterm Code	Minterm Number
$A'B'C'$	000	m_0
$A'B'C$	001	m_1
$A'BC'$	010	m_2
$A'BC$	011	m_3
$AB'C'$	100	m_4
$AB'C$	101	m_5
ABC'	110	m_6
ABC	111	m_7

Algebraic Forms of Switching Functions

- Compact form of canonical SOP form:

  $f_1(A,B,C) = m_2 + m_3 + m_6 + m_7$

- A further simplified form:

  $f_1(A,B,C) = \Sigma\, m\,(2,3,6,7)$ (minterm list form)

- The ***order of variables*** in the functional notation is important.
- Deriving truth table of $f_1(A,B,C)$ from minterm list:

Row No. (i)	**Inputs** ABC	**Outputs** $f_1(A,B,C)= \Sigma m(2,3,6,7)$	**Complement** $f_1'(A,B,C)= \Sigma m(0,1,4,5)$
0	000	0	1 $\leftarrow$ m_0
1	001	0	1 $\leftarrow$ m_1
2	010	1 $\leftarrow$ m_2	0
3	011	1 $\leftarrow$ m_3	0
4	100	0	1 $\leftarrow$ m_4
5	101	0	1 $\leftarrow$ m_5
6	110	1 $\leftarrow$ m_6	0
7	111	1 $\leftarrow$ m_7	0

Algebraic Forms of Switching Functions

- ***Example***: Given $f(A,B,Q,Z) = A'B'Q'Z' + A'B'Q'Z + A'BQZ' + A'BQZ$, express $f(A,B,Q,Z)$ and $f\,'(A,B,Q,Z)$ in minterm list form.

$$
\begin{aligned}
f(A,B,Q,Z) &= A'B'Q'Z' + A'B'Q'Z + A'BQZ' + A'BQZ \\
&= m_0 + m_1 + m_6 + m_7 \\
&= \Sigma\, m(0, 1, 6, 7)
\end{aligned}
$$

$$
\begin{aligned}
f\,'(A,B,Q,Z) &= m_2 + m_3 + m_4 + m_5 + m_8 + m_9 + m_{10} + m_{11} + m_{12} \\
&\quad + m_{13} + m_{14} + m_{15} \\
&= \Sigma\, m(2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15)
\end{aligned}
$$

- $\displaystyle\sum_{i=0}^{2^n-1} m_i = 1$ (2.6)
- $AB + (AB)' = 1$ and $AB + A' + B' = 1$, but $AB + A'B' \neq 1$.

Algebraic Forms of Switching Functions

- A ***maxterm*** is a sum term in which all the variables appear exactly once either complemented or uncomplemented.
- ***Canonical Product of Sums* (*canonical POS*)**:
  - Represented as a product of maxterms only.
  - ***Example***: $f_2(A,B,C) = (A+B+C)(A+B+C')(A'+B+C)(A'+B+C')$
- Maxterms of three variables:

Maxterm	Maxterm Code	Maxterm Number
$A+B+C$	000	M_0
$A+B+C'$	001	M_1
$A+B'+C$	010	M_2
$A+B'+C'$	011	M_3
$A'+B+C$	100	M_4
$A'+B+C'$	101	M_5
$A'+B'+C$	110	M_6
$A'+B'+C'$	111	M_7

Algebraic Forms of Switching Functions

- $f_2(A,B,C) = M_0 M_1 M_4 M_5$
  $= \Pi M(0,1,4,5)$ (maxterm list form)

- The truth table for $f_2(A,B,C)$:

Rwo No. (i)	**Inputs** ABC	M_0 $A+B+C$	M_1 $A+B+C'$	M_4 $A'+B+C$	M_5 $A'+B+C'$	**Outputs** $f_2(A,B,C)$
0	000	0	1	1	1	0
1	001	1	0	1	1	0
2	010	1	1	1	1	1
3	011	1	1	1	1	1
4	100	1	1	0	1	0
5	101	1	1	1	0	0
6	110	1	1	1	1	1
7	111	1	1	1	1	1

Algebraic Forms of Switching Functions

- Truth tables of $f_1(A,B,C)$ and $f_2(A,B,C)$ are identical.
- Hence, $f_1(A,B,C) = \Sigma\, m\ (2,3,6,7)$

$$= f_2(A,B,C)$$

$$= \Pi M(0,1,4,5)$$

- ***Example: Given*** $f(A,B,C) = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C')$, construct the truth table and express in both maxterm and minterm form.
  - $f(A,B,C) = M_1M_3M_5M_7 = \Pi M(1,3,5,7) = \Sigma\, m\ (0,2,4,6)$

Row No. (i)	Inputs ABC	Outputs $f(A,B,C) = \Pi\, M(1,3,5,7) = \Sigma m(0,2,4,6)$
0	000	1 m_0
1	001	0 ← M_1
2	010	1 m_2
3	011	0 ← M_3
4	100	1 m_4
5	101	0 ← M_5
6	110	1 m_6
7	111	0 ← M_7

Algebraic Forms of Switching Functions

- Relationship between minterm m_i and maxterm M_i:
  - For $f(A,B,C)$, $(m_1)' = (A'B'C)' = A + B + C' = M_1$
  - In general, $(m_i)' = M_i$

    $(Mi)' = ((m_i)')' = m_i$

Algebraic Forms of Switching Functions

- ***Example***: Relationship between the maxterms for a function and its complement.
  - For $f(A,B,C) = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C')$
  - The truth table is:

Row No. (i)	**Inputs** ABC	**Outputs** $f(A,B,C)$	**Outputs** $f'(A,B,C) = \Pi M(0,2,4,6)$
0	000	1	0 $\leftarrow$ M_0
1	001	0	1
2	010	1	0 $\leftarrow$ M_2
3	011	0	1
4	100	1	0 $\leftarrow$ M_4
5	101	0	1
6	110	1	0 $\leftarrow$ M_6
7	111	0	1

Algebraic Forms of Switching Functions

- From the truth table

  $f'(A,B,C) = \Pi M(0,2,4,6)$ and $f(A,B,C) = \Pi M(1,3,5,7)$

- Since $f(A,B,C) \cdot f'(A,B,C) = 0$,

  $(M_0 M_2 M_4 M_6)(M_1 M_3 M_5 M_7) = 0$ or $\prod_{i=0}^{2^3-1} M_i = 0$

- In general, $\prod_{i=0}^{2^n-1} M_i = 0$

- Another observation from the truth table:

  $f(A,B,C) = \Sigma\, m\,(0,2,4,6) = \Pi M(1,3,5,7)$

  $f'(A,B,C) = \Sigma\, m\,(1,3,5,7) = \Pi M(0,2,4,6)$

Derivation of Canonical Forms

- Derive canonical POS or SOP using switching algebra.
- ***Theorem 10. Shannon's expansion theorem***

  (a). $f(x_1, x_2, \ldots, x_n) = x_1 f(1, x_2, \ldots, x_n) + (x_1)' f(0, x_2, \ldots, x_n)$

  (b). $f(x_1, x_2, \ldots, x_n) = [x_1 + f(0, x_2, \ldots, x_n)]\,[(x_1)' + f(1, x_2, \ldots, x_n)]$

- ***Example***: $f(A,B,C) = AB + AC' + A'C$
  - $f(A,B,C) = AB + AC' + A'C = A\,f(1,B,C) + A'\,f(0,B,C)$
    $= A(1 \cdot B + 1 \cdot C' + 1' \cdot C) + A'(0 \cdot B + 0 \cdot C' + 0' \cdot C) = A(B + C') + A'C$
  - $f(A,B,C) = A(B + C') + A'C = B[A(1+C') + A'C] + B'[A(0 + C') + A'C]$
    $= B[A + A'C] + B'[AC' + A'C] = AB + A'BC + AB'C' + A'B'C$
  - $f(A,B,C) = AB + A'BC + AB'C' + A'B'C$
    $= C[AB + A'B \cdot 1 + AB' \cdot 1' + A'B' \cdot 1] + C'[AB + A'B \cdot 0 + AB' \cdot 0' + A'B' \cdot 0]$
    $= ABC + A'BC + A'B'C + ABC' + AB'C'$

Derivation of Canonical Forms

- ***Alternative:*** Use Theorem 6 to add missing literals.
- ***Example***: $f(A,B,C) = AB + AC' + A'C$ to canonical SOP form.
  - $AB = ABC' + ABC = m_6 + m_7$
  - $AC' = AB'C' + ABC' = m_4 + m_6$
  - $A'C = A'B'C + A'BC = m_1 + m_3$
  - Therefore,

    $f(A,B,C) = (m_6 + m_7) + (m_4 + m_6) + (m_1 + m_3) = \Sigma m(1, 3, 4, 6, 7)$

- ***Example***: $f(A,B,C) = A(A + C')$ to canonical POS form.
  - $A = (A+B')(A+B) = (A+B'+C')(A+B'+C)(A+B+C')(A+B+C)$

    $= M_3M_2M_1M_0$
  - $(A+C')= (A+B'+C')(A+B+C') = M_3M_1$
  - Therefore,

    $f(A,B,C) = (M_3M_2M_1M_0)(M_3M_1) = \Pi M(0, 1, 2, 3)$

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY
臺灣–巴拉圭科技大學

Incompletely Specified Functions

- A switching function may be incompletely specified.
- Some minterms are omitted, which are called *don't-care minterms*.
- Don't cares arise in two ways:
  - Certain input combinations never occur.
  - Output is required to be 1 or 0 only for certain combinations.
- Don't care minterms: d_i Don't care maxterms: D_i

- ***Example***: $f(A,B,C)$ *has minterms* m_0, m_3, and m_7 and don't-cares d_4 and d_5.
  - Minterm list is: $f(A,B,C) = \Sigma m(0,3,7) + d(4,5)$
  - Maxterm list is: $f(A,B,C) = \Pi M(1,2,6) \cdot D(4,5)$
  - $f'(A,B,C) = \Sigma m(1,2,6) + d(4,5) = \Pi M(0,3,7) \cdot D(4,5)$
  - $f(A,B,C) = A'B'C' + A'BC + ABC + d(AB'C' + AB'C)$

    $= B'C' + BC$ (use d_4 and omit d_5)

UNIVERSIDAD POLITÉCNICA TAIWAN – PARAGUAY
臺灣–巴拉圭科技大學

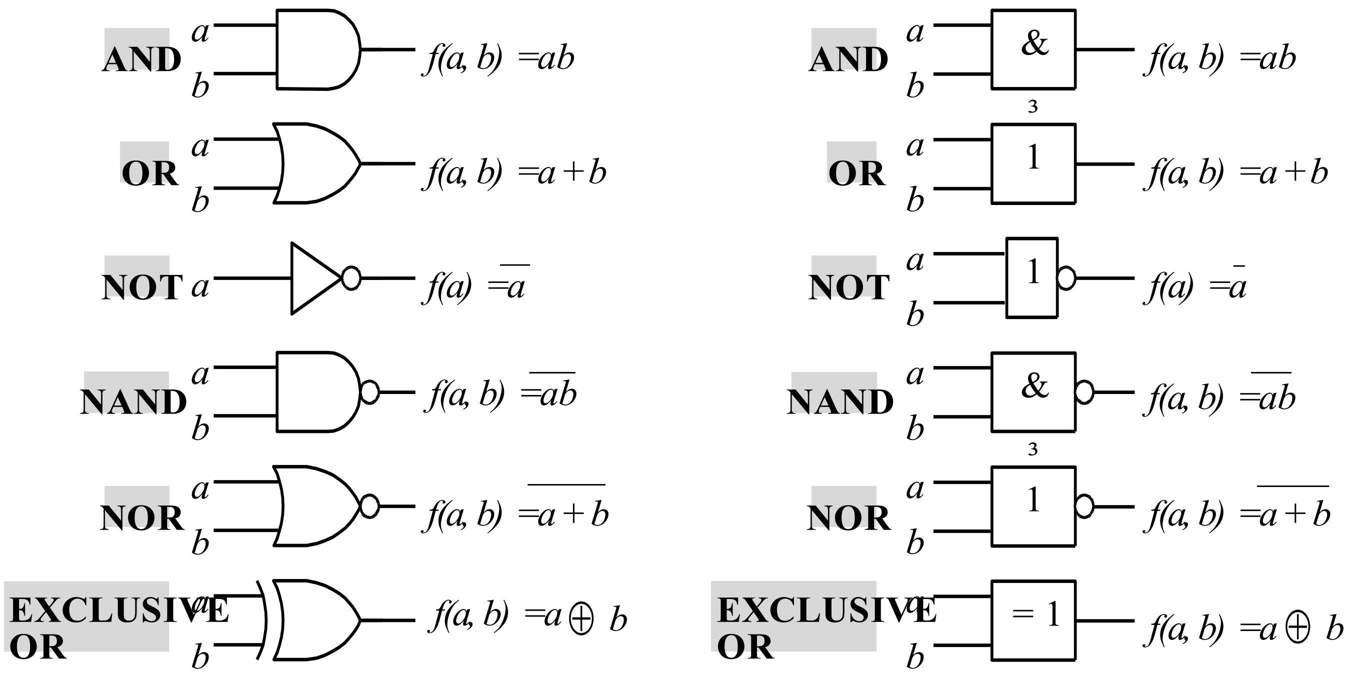
Electronic Logic Gates

- ***Electrical Signals and Logic Values***

Electric Signal	Logic Value	
	Positive Logic	Negative Logic
High Voltage (H)	1	0
Low Voltage (L)	0	1

- A signal that is set to logic 1 is said to be *asserted*, *active*, or *true*.
- An *active-high* signal is asserted when it is high (positive logic).
- An *active-low* signal is asserted when it is low (negative logic).

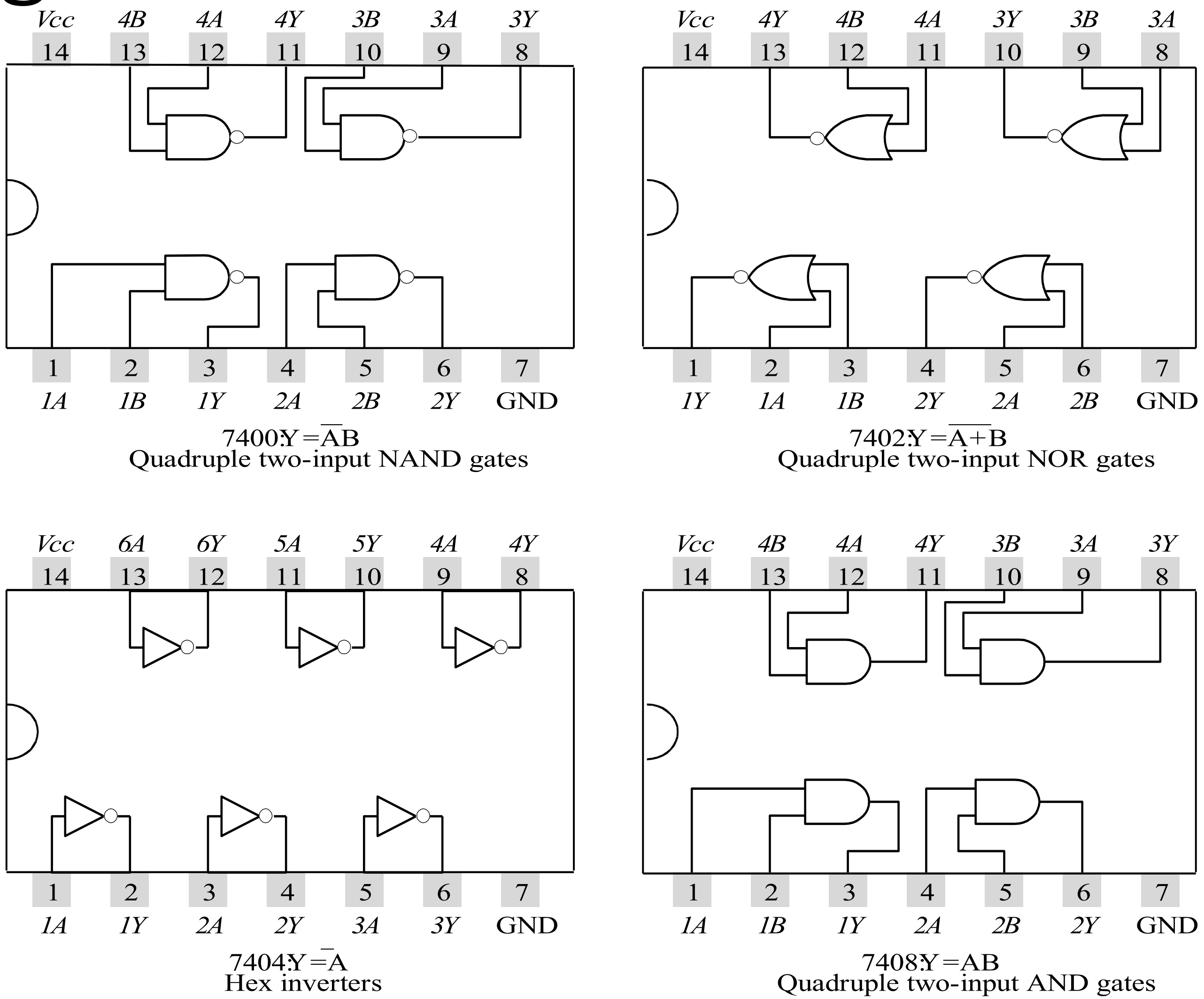
Electronic Logic Gates



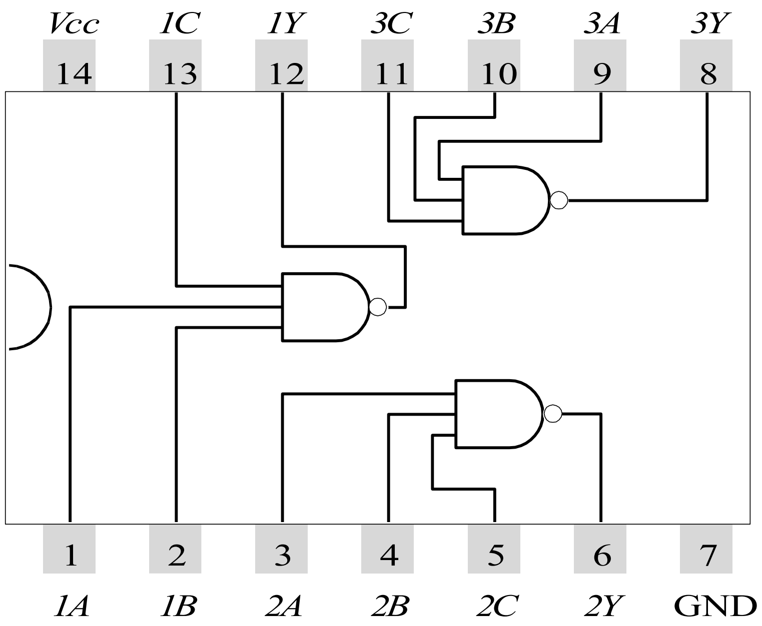
Symbol set 1

Symbol set 2
(ANSI/IEEE Standard 91-1984)

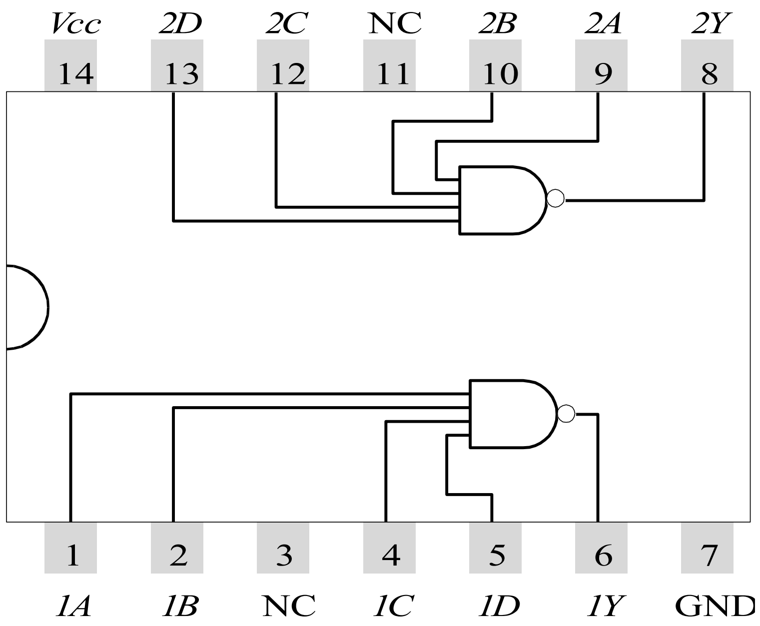
Electronic Logic Gates



Electronic Logic Gates

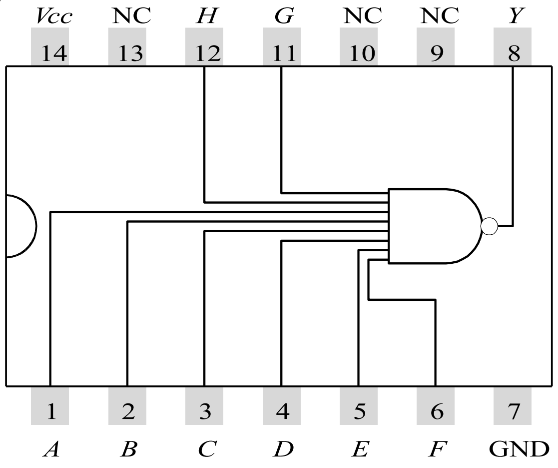


7410: $Y = \overline{ABC}$
Triple three-input NAND gates

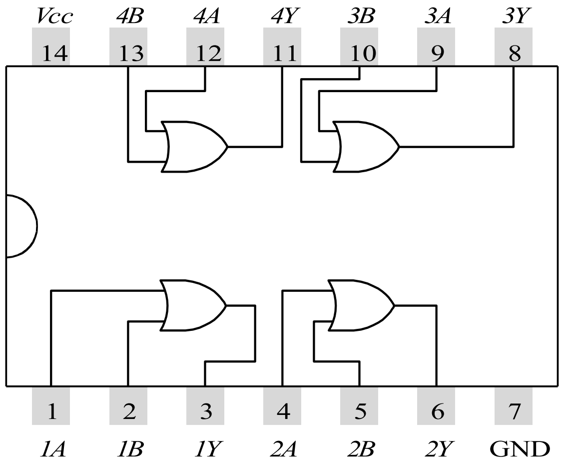


7420: $Y = \overline{ABCD}$
Dual four-input NAND gates

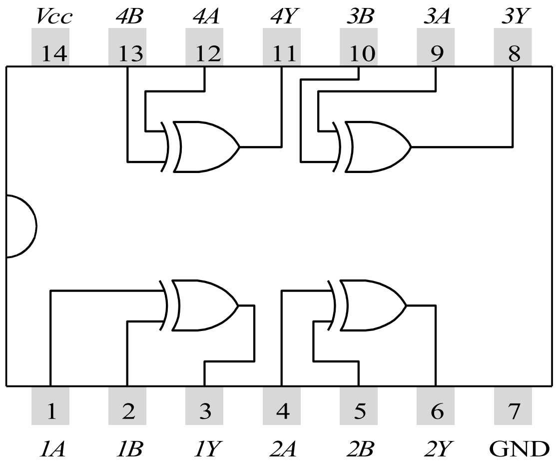
Electronic Logic Gates



7430:Y =$\overline{\text{ABCDEFGH}}$
8-input NAND gate



7432:Y =A+B
Quadruple two-input OR gates



7486:Y =AÅB
Quadruple two-input exclusive-OR gates

Basic Functional Components

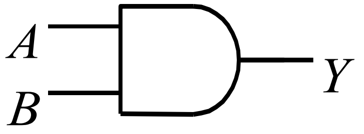
- ***AND***

a	b	$f_{AND}(a, b) = ab$
0	0	0
0	1	0
1	0	0
1	1	1

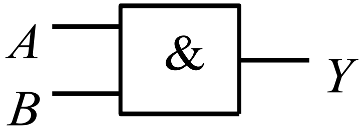
(a)

A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

(b)



(c)



(d)

(a) AND logic function.
(b) Electronic AND gate.
(c) Standard symbol.
(d) IEEE block symbol.

Basic Functional Components

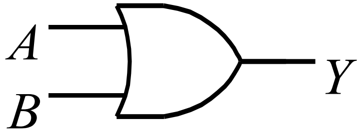
- ***OR***

a	b	$f_{OR}(a, b) = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

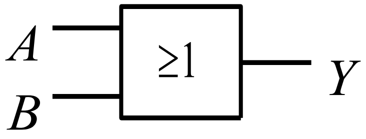
(a)

A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

(b)



(c)



(d)

(a) OR logic function.
(b) Electronic OR gate.
(c) Standard symbol.
(d) IEEE block symbol.

Basic Functional Components

- Meaning of the designation ≥ 1 in IEEE symbol:

ab	sum(a, b)	sum(a, b) ≥ 1	$f_{OR}(a, b) = a + b$
00	0	False	0
01	1	True	1
10	1	True	1
11	2	True	1

Basic Functional Components (4)

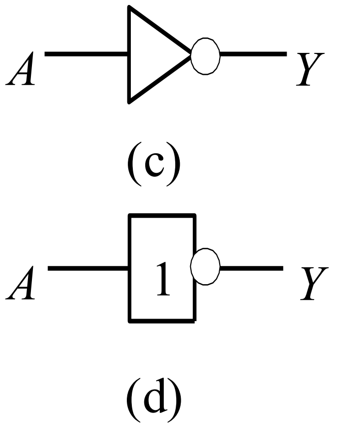
- ***NOT***

a	$f_{NOT}(a) = \bar{a}$
0	1
1	0

(a)

A	Y
L	H
H	L

(b)



(a) NOT logic function.
(b) Electronic NOT gate.
(c) Standard symbol.
(d) IEEE block symbol.

Basic Functional Components

- ***Positive Versus Negative Logic***

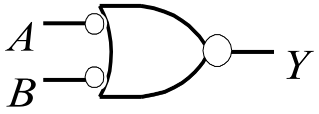
	Positive Logic	Negative Logic
1 is represented by	High Voltage	Low Voltage
0 is represented by	Low Voltage	High Voltage

Basic Functional Components

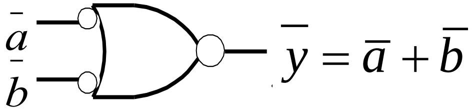
- ***AND Gate Usage in Negative Logic***

A	B	Y
1	1	1
1	0	1
0	1	1
0	0	0

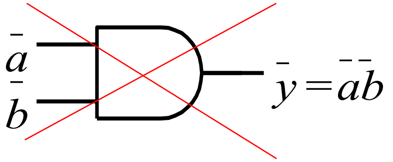
(a)



(b)



(c)



(d)

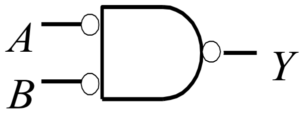
- (a) AND gate truth table ($L = 1$, $H = 0$)
- (b) Alternate AND gate symbol (in negative logic)
- (c) Preferred usage $\quad y = a \cdot b = \overline{\overline{a \cdot b}} = \overline{\bar{a} + \bar{b}} = \bar{f}_{OR}(\bar{a}, \bar{b})$
- (d) Improper usage $\quad \bar{y} = \overline{\overline{(\bar{a})} + \overline{(\bar{b})}} = \overline{a + b} = \bar{f}_{OR}(a, b)$

Basic Functional Components

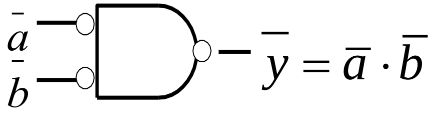
- ***OR Gate Usage in Negative Logic***

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

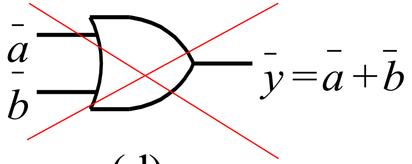
(a)



(b)



(c)



(d)

- (a) OR gate truth table(L = 1, H = 0)
- (b) Alternate OR gate symbol (in negative logic)
- (c) Preferred usage $y = a + b = \overline{\overline{a+b}} = \overline{\bar{a} \cdot \bar{b}} = \bar{f}_{AND}(\bar{a}, \bar{b})$
- (d) Improper usage $\bar{y} = \overline{\overline{(\bar{a})} \cdot \overline{(\bar{b})}} = \overline{a \cdot b} = \bar{f}_{AND}(a, b)$

Basic Functional Components

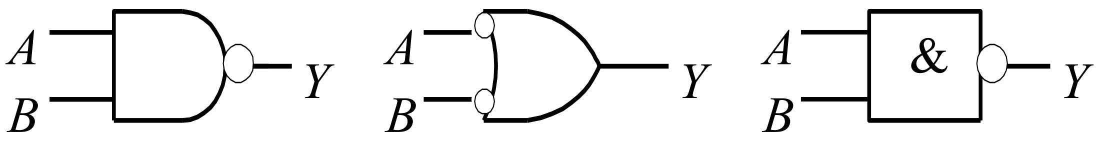
- ***NAND***

a	b	$f_{NAND}(a, b) = \overline{ab}$
0	0	1
0	1	1
1	0	1
1	1	0

(a)

A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

(b)

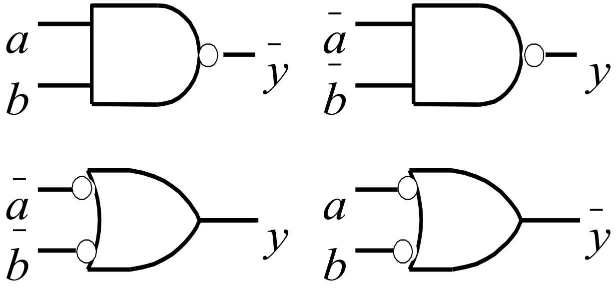


(c) (d) (e)

- (a) NAND logic function
- (b) Electronic NAND gate
- (c) Standard symbol
- (e) IEEE block symbol

Basic Functional Components

- Matching signal polarity to NAND gate inputs/outputs
  - (a) Preferred usage (b) Improper usage



(a) (b)

- Additional properties of NAND gate:

$$f_{NAND}(a,a) = \overline{a \cdot a} = \overline{a} = f_{NOT}(a)$$

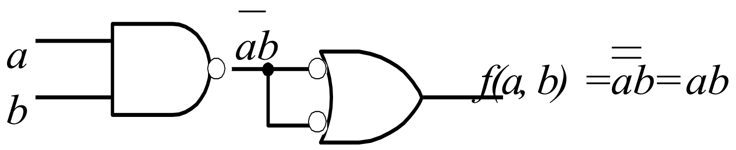
$$\overline{f}_{NAND}(a,b) = \overline{\overline{a \cdot b}} = a \cdot b = f_{AND}(a,b)$$

$$f_{NAND}(\overline{a},\overline{b}) = \overline{\overline{a} \cdot \overline{b}} = a + b = f_{OR}(a,b)$$

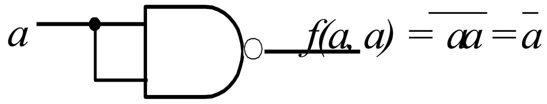
- Hence, NAND gate may be used to implement all three elementary operators.

Basic Functional Components

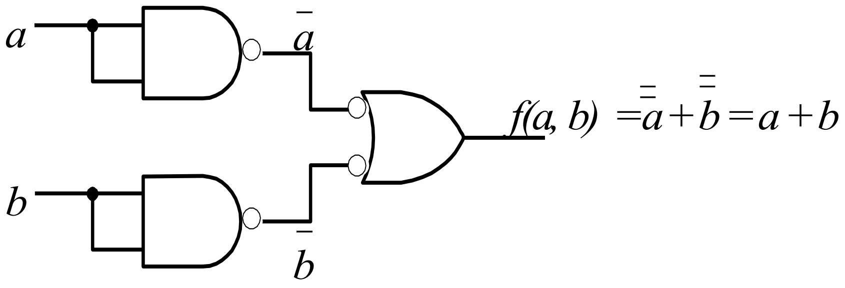
- AND, OR, and NOT gates constructed exclusively from NAND gates



AND gate



NOT gate



OR gate

Basic Functional Components

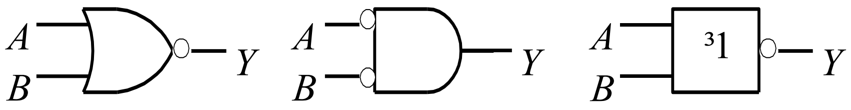
- ***NOR***

a	b	$f_{NOR}(a, b) = \overline{a + b}$
0	0	1
0	1	0
1	0	0
1	1	0

(a)

A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

(b)

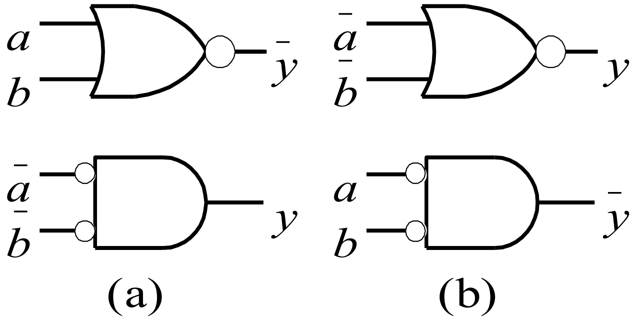


(c) (d) (e)

- (a) NAND logic function
- (b) Electronic NAND gate
- (c) Standard symbol
- (d) IEEE block symbol

Basic Functional Components

- Matching signal polarity to NOR gate inputs/outputs
  - (a) Preferred usage (b) Improper usage



- Additional properties of NOR gate:

$$f_{NOR}(a,a) = \overline{a+a} = \bar{a} = f_{NOT}(a)$$

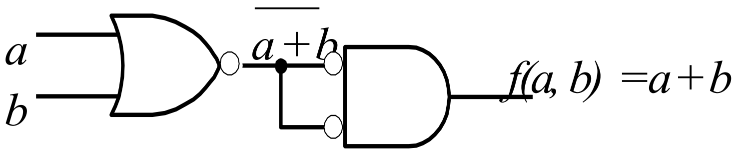
$$\bar{f}_{NOR}(a,b) = \overline{\overline{a+b}} = a+b = f_{OR}(a,b)$$

$$f_{NOR}(\bar{a},\bar{b}) = \overline{\bar{a}+\bar{b}} = a \cdot b = f_{AND}(a,b)$$

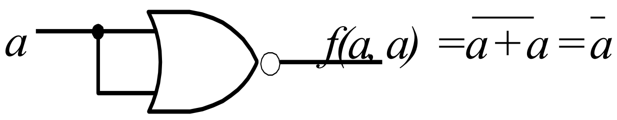
- Hence, NOR gate may be used to implement all three elementary operators.

Basic Functional Components

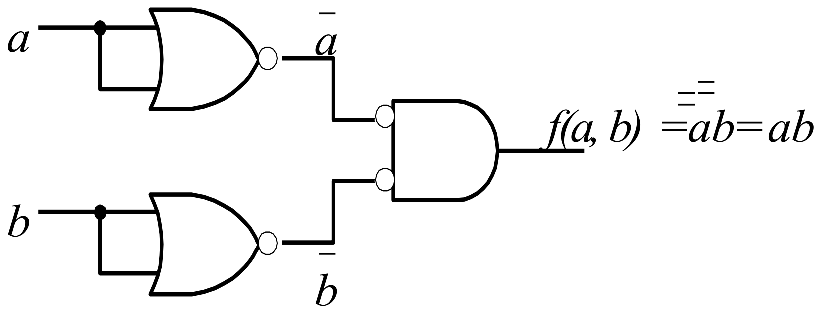
- AND, OR, and NOT gates constructed exclusively from NOR gates.



OR gate



NOT gate



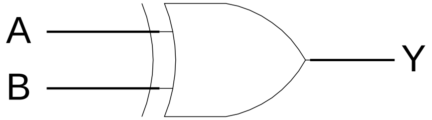
AND gate

Basic Functional Components

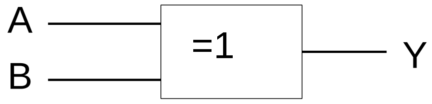
- ***Exclusive-OR (XOR)***
  - $f_{\mathrm{XOR}}(a, b) = a \oplus b = \overline{a}b + a\overline{b}$

$a\ b$	$f_{\mathrm{XOR}}(a, b) = a \oplus b$	$A\ B$	Y
0 0	0	L L	L
0 1	1	L H	H
1 0	1	H L	H
1 1	0	H H	L

(a) XOR logic function (b) Electronic XOR gate



(c) Standard symbol



(d) IEEE block symbol

Basic Functional Components

- POS of XOR

$$
\begin{aligned}
a \oplus b &= \bar{a}b + a\bar{b} \\
&= \bar{a}a + \bar{a}b + a\bar{b} + b\bar{b} && \text{[P2(a), P6(b)]} \\
&= \bar{a}(a+b) + \bar{b}(a+b) && \text{[P5(b)]} \\
&= (\bar{a}+\bar{b})(a+b) && \text{[P5(b)]}
\end{aligned}
$$

- Some other useful relationships
  - $a \oplus a = 0$ (2.25)
  - $a \oplus \bar{a} = 1$ (2.26)
  - $a \oplus 0 = a$ (2.27)
  - $a \oplus 1 = \bar{a}$ (2.28)
  - $\bar{a} \oplus \bar{b} = a \oplus b$ (2.29)
  - $a \oplus b = b \oplus a$ (2.30)
  - $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (2.31)

Basic Functional Components (17)

- Output of XOR gate is asserted when the mathematical sum of inputs is *one*:

ab	$sum(a, b)$	$sum(a, b) = 1?$	$f(a, b) = a \oplus b$
00	0	False	0
01	1	True	1
10	1	True	1
11	2	False	0

- The output of XOR is the *modulo*-2 sum of its inputs.

Basic Functional Components (18)

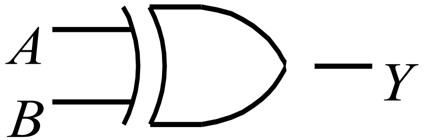
- ***Exclusive-NOR (XNOR)***
  - $f_{XNOR}(a,b) = \overline{a \oplus b} = a \odot b$

a	b	$f_{XNOR}(a,b) = a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

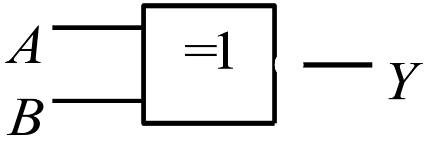
(a)

A	B	Y
L	L	H
L	H	L
H	L	L
H	H	H

(b)



(c)



(d)

- (a) XNOR logic function
- (b) Electronic XNOR gate
- (c) Standard symbol
- (d) IEEE block symbol

Analysis of Combinational Circuits (1)

- Digital Circuit ***Design***:
  - Word description of a function
    - $\Rightarrow$ a set of switching equations
    - $\Rightarrow$ hardware realization (gates, programmable logic devices, etc.)

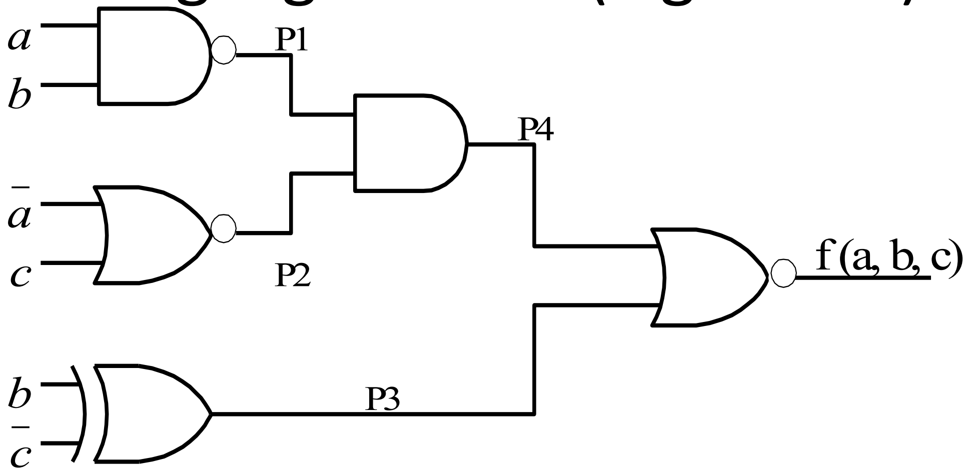
- Digital Circuit ***Analysis***:
  - Hardware realization
    - $\Rightarrow$ switching expressions, truth tables, timing diagrams, etc.

- Analysis is used
  - To determine the behavior of the circuit
  - To verify the correctness of the circuit
  - To assist in converting the circuit to a different form.

Analysis of Combinational Circuits (2)

- ***Algebraic Method***: Use switching algebra to derive a desired form.

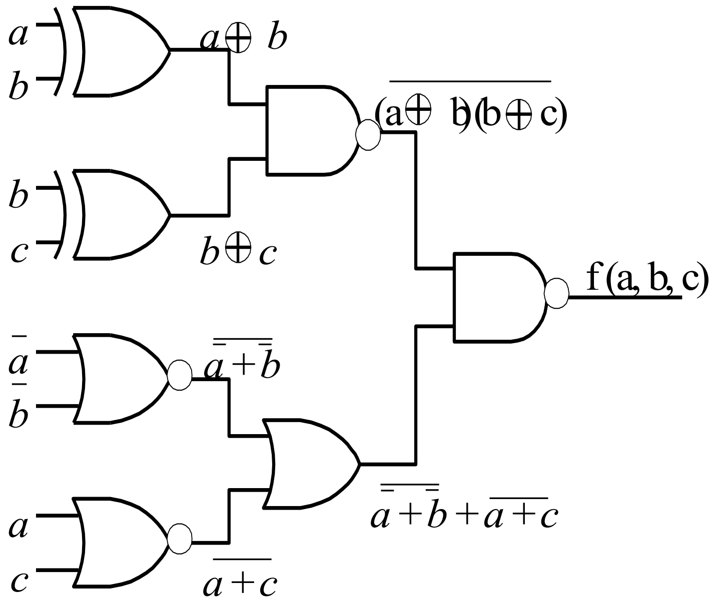
- ***Example 2.33***: Find a simplified switching expressions and logic network for the following logic circuit (Fig. 2.21a).



(a)

Analysis of Combinational Circuits (4)

- ***Example 2.34***: Find a simplified switching expressions and logic network for the following logic circuit (Fig. 2.22).

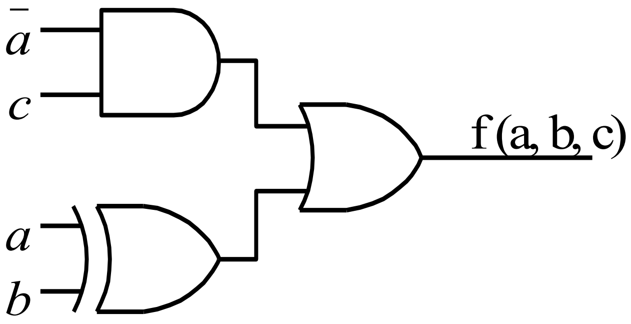


Given circuit

Analysis of Combinational Circuits (5)

- Derive the output expression:

$$
\begin{aligned}
f(a,b,c) &= \overline{\overline{(a\oplus b)(b\oplus c)}\cdot(\overline{\overline{a}+\overline{\overline{b}}}+\overline{a+c})} && \text{[T8(b)]}\\
&= \overline{\overline{(a\oplus b)(b\oplus c)}}+\overline{\overline{\overline{a}+\overline{\overline{b}}}+\overline{a+c}} && \text{[T8(a)]}\\
&= (a\oplus b)(b\oplus c)+(\overline{a}+\overline{b})(a+c) && \text{[Eq. 2.24]}\\
&= (a\overline{b}+\overline{a}b)(b\overline{c}+\overline{b}c)+(\overline{a}+\overline{b})(a+c) && \text{[P5(b)]}\\
&= a\overline{b}b\overline{c}+a\overline{b}\overline{b}c+\overline{a}bb\overline{c}+\overline{a}b\overline{b}c+\overline{a}a+\overline{a}c+a\overline{b}+\overline{b}c && \text{[P6(b), T4(a)]}\\
&= a\overline{b}c+\overline{a}b\overline{c}+\overline{a}c+a\overline{b}+\overline{b}c && \text{[T4(a)]}\\
&= \overline{a}b\overline{c}+\overline{a}c+a\overline{b}+\overline{b}c && \text{[T9(a)]}\\
&= \overline{a}b\overline{c}+\overline{a}c+a\overline{b} && \text{[T7(a)]}\\
&= \overline{a}b+\overline{a}c+a\overline{b} && \text{[Eq. 2.24]}\\
&= \overline{a}c+a\oplus b
\end{aligned}
$$



Simplified circuit

Analysis of Combinational Circuits (6)

- ***Truth Table Method***: Derive the truth table one gate at a time.

- The truth table for Example 2.34:

abc	$\bar{a}c$	$a \oplus b$	$f(a,b,c)$
000	0	0	0
001	1	0	1
010	0	1	1
011	1	1	1
100	0	1	1
101	0	1	1
110	0	0	0
111	0	0	0