Boolean Algebra

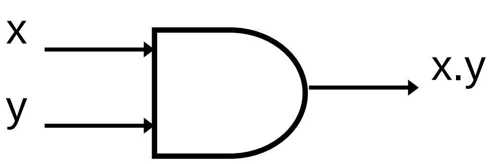
Two-valued Boolean Algebra

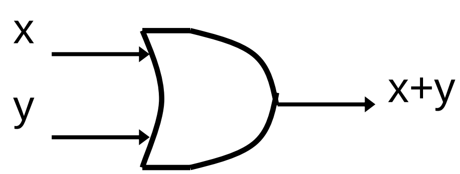
- Set of elements: {0,1}
- Set of operations: { ., + , ‘ }

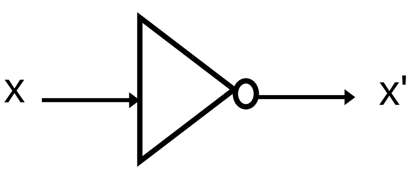
x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1

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

x	x'
0	1
1	0







Signals: High = 5V = 1; Low = 0V = 0

Boolean Functions

- **Boolean function** is an expression formed with binary variables, the two binary operators, OR and AND, and the unary operator, NOT, parenthesis and the equal sign.
- Its result is also a binary value.
- We usually use **.** for AND, **+** for OR, and **'** for NOT. Sometimes, we may omit the **.** if there is no ambiguity.

Complement of Boolean Functions

- Given a function, F, the **complement** of this function, F', is obtained by interchanging 1 with 0 in the function's output values.

x	y	z	**F1**	**F1'**
0	0	0	**0**	**1**
0	0	1	**0**	**1**
0	1	0	**0**	**1**
0	1	1	**0**	**1**
1	0	0	**0**	**1**
1	0	1	**0**	**1**
1	1	0	**1**	**0**
1	1	1	**0**	**1**

Example: F1 = x. y. z'

Complement: F1' = (x.y.z')'
= x' + y' + (z')' DeMorgan
= x' + y' + z Involution

Complement of Functions

- More general DeMorgan's theorems useful for obtaining complement functions:

$$(A + B + C + ... + Z)' = A' \cdot B' \cdot C' \cdot \ldots \cdot Z'$$

$$(A \cdot B \cdot C \ldots \cdot Z)' = A' + B' + C' + \ldots + Z'$$

Standard Forms

- Certain types of Boolean expressions lead to gating networks which are desirable from implementation viewpoint.
- Two Standard Forms: ***Sum-of-Products*** and ***Product-of-Sums***
- **Literals**: a variable on its own or in its complemented form. Examples: x, x' , y, y'
- **Product Term**: a single literal or a logical product (AND) of several literals.

  Examples: x, x.y.z', A'.B, A.B

- **Sum Term**: a single literal or a logical sum (OR) of several literals.

  Examples: x, x+y+z', A'+B, A+B, c+d+h'+j

- **Sum-of-Products (SOP) Expression**: a product term or a logical sum (OR) of several product terms.

  Examples: x, x+y.z', x.y‘ + x'.y.z, A.B + A'.B',
  A + B'.C + A.C' + C.D

- **Product-of-Sums (POS) Expression**: a sum term or a logical product (AND) of several sum terms.

  Examples: x, x.(y+z'), (x+y').(x'+y+z),
  (A+B).(A'+B'), (A+B+C).D'.(B'+D+E')

Standard Forms

- Every Boolean expression can either be expressed as sum-of-products or product-of-sums expression.

Examples:

SOP: $x'.y + x.y' + x.y.z$

POS: $(x + y').(x' + y).(x' + z')$

both: $x' + y + z$ or $x.y.z'$

neither: $x.(w' + y.z)$ or $z' + w.x'.y + v.(x.z + w')$

Minterm & Maxterm

- Consider two binary variables x, y.
- Each variable may appear as itself or in complemented form as literals (i.e. x, x' & y, y')
- For two variables, there are four possible combinations with the AND operator, namely:

  x'.y', x'.y, x.y', x.y

- These product terms are called the *minterms*.
- A **minterm** of *n* variables is the product of *n* literals from the different variables.

Minterm & Maxterm

- **In general, *n* variables can give 2^n minterms**.

- In a similar fashion, a **maxterm** of *n* variables is the sum of *n* literals from the different variables.

  Examples: x'+y', x'+y, x+y',x+y

- In general, *n* variables can also give 2^n maxterms.

Minterm & Maxterm

- The minterms and maxterms of 2 variables are denoted by m0 to m3 and M0 to M3 respectively:

Minterms				Maxterms	
x	y	term	notation	term	notation
0	0	x'.y'	m_0	x+y	M_0
0	1	x'.y	m_1	x+y'	M_1
1	0	x.y'	m_2	x'+y	M_2
1	1	x.y	m_3	x'+y'	M_3

Each minterm is the complement of the corresponding maxterm:

Example: $m_2 = x.y'$

$$m_2' = (x.y')' = x' + (y')' = x'+y = M_2$$

Canonical Form: Sum of Minterms

- What is a canonical/normal form?
  - A unique form for representing something.
- Minterms are product terms.
  - Can express Boolean functions using Sum-of-Minterms form.

Canonical Form: Sum of Minterms

a) Obtain the truth table. Example:

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Canonical Form: Sum of Minterms

b) Obtain Sum-of-Minterms by gathering/summing the minterms of the function (where result is a 1)

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

F1 = x.y.z' = Σm(6)

F2 = x'.y'.z + x.y'.z' + x.y'.z + x.y.z' + x.y.z
= Σm(1,4,5,6,7)

F3 = x'.y'.z + x'.y.z + x.y'.z' +x.y'.z
= Σm(1,3,4,5)

Canonical Form: Product of Maxterms

- Maxterms are sum terms.
- For Boolean functions, the maxterms of a function are the terms for which the result is 0.
- Boolean functions can be expressed as Products-of-Maxterms.

Canonical Form: Product of Maxterms

E.g.: $F2 = \Pi M(0,2,3) = (x+y+z).(x+y'+z).(x+y'+z')$

$F3 = \Pi M(0,2,6,7)$

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

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Canonical Form: Product of Maxterms

- Why is this so? Take F2 as an example.

  $F2 = \Sigma m(1,4,5,6,7)$

- The complement function of F2 is:

  $F2' = \Sigma m(0,2,3)$

  $= m0 + m2 + m3$

(Complement functions' minterms are the opposite of their original functions, i.e. when original function = 0)

x	y	z	F2	F2'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Canonical Form: Product of Maxterms

From previous slide, F2' = m0 + m2 + m3

Therefore:

$$
\begin{aligned}
F2 &= (m0 + m2 + m3)' & \\
&= m0' \,.\, m2' \,.\, m3' & \text{DeMorgan} \\
&= M0 \,.\, M2 \,.\, M3 & m\mathit{x}' = M\mathit{x} \\
&= \Pi M(0,2,3) &
\end{aligned}
$$

- Every Boolean function can be expressed as either Sum-of-Minterms or Product-of-Maxterms.

Conversion of Canonical Forms

- **Sum-of-Minterms $\Rightarrow$ Product-of-Maxterms**
  - Rewrite minterm shorthand using maxterm shorthand.
  - Replace minterm indices with indices not already used.

    Eg: $F1(A,B,C) = \sum m(3,4,5,6,7) = \prod M(0,1,2)$

- **Product-of-Maxterms $\Rightarrow$ Sum-of-Minterms**
  - Rewrite maxterm shorthand using minterm shorthand.
  - Replace maxterm indices with indices not already used.

    Eg: $F2(A,B,C) = \prod M(0,3,5,6) = \sum m(1,2,4,7)$

Conversion of Canonical Forms

- **Sum-of-Minterms of F $\Rightarrow$ Sum-of-Minterms of F'**
  - In minterm shorthand form, list the indices not already used in F.

    Eg: $F1(A,B,C) = \sum m(3,4,5,6,7)$

    $F1'(A,B,C) = \sum m(0,1,2)$

- **Product-of-Maxterms of F $\Rightarrow$ Prod-of-Maxterms of F'**
  - In maxterm shorthand form, list the indices not already used in F.

    Eg: $F1(A,B,C) = \prod M(0,1,2)$

    $F1'(A,B,C) = \prod M(3,4,5,6,7)$

Conversion of Canonical Forms

- **Sum-of-Minterms of F $\Rightarrow$ Product-of-Maxterms of F'**
  - Rewrite in maxterm shorthand form, using the same indices as in F.

    Eg: F1(A,B,C) = $\sum m(3,4,5,6,7)$

    F1'(A,B,C) = $\prod M(3,4,5,6,7)$

- **Product-of-Maxterms of F $\Rightarrow$ Sum-of-Minterms of F'**
  - Rewrite in minterm shorthand form, using the same indices as in F.

    Eg: F1(A,B,C) = $\prod M(0,1,2)$

    F1'(A,B,C) = $\sum m(0,1,2)$

Binary Functions

- Given *n* variables, there are 2^n possible minterms.
- As each function can be expressed as sum-of-minterms, there could be 2^{2^n} different functions.
- In the case of two variables, there are $2^2 = 4$ possible minterms; and $2^4 = 16$ different possible binary functions.
- The 16 possible binary functions are shown in the next slide.

Binary Functions

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Symbol			.	/		/		$\oplus$	+
Name			AND	x, but not y		y, but not x		XOR	OR

x	y	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
Symbol		$\downarrow$	$\odot$	'	$\subset$	'	$\supset$	$\uparrow$	
Name		NOR	XNOR					NAND	

Logic Gates

Introduction

- A Logic Gate is an electronic circuit capable of making logical decisions.
- It has one output and one or more inputs.
- Basic building blocks of digital systems.
- 0 and 1.
- 0 means 0 to 3 V.
- 1 means 3.5 V to 5 V.
- 3 V to 3.5 V is undefined.

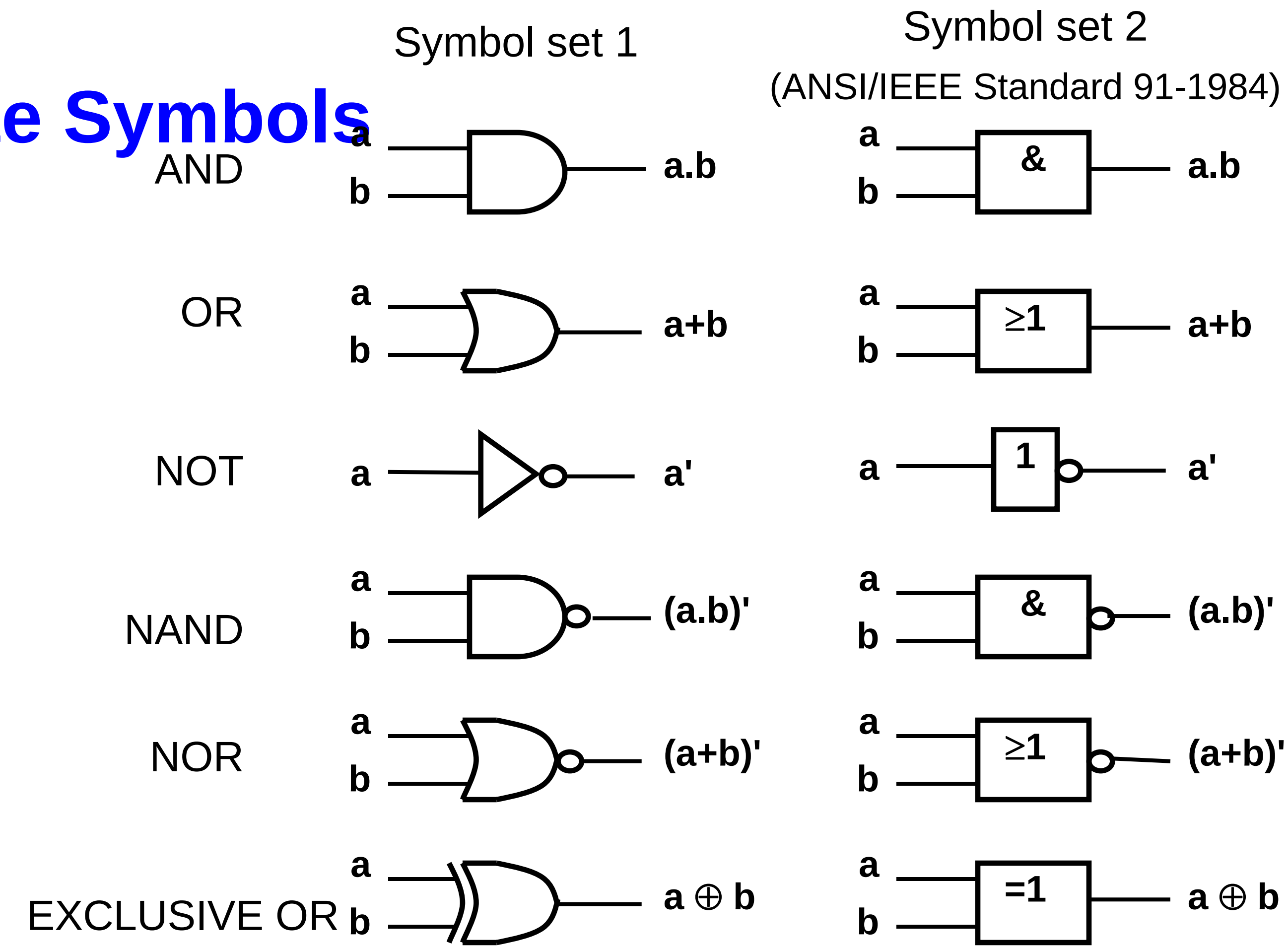
Logic Gates and Circuits

- Logic Gates
  - The Inverter
  - The AND Gate
  - The OR Gate
  - The NAND Gate
  - The NOR Gate
  - The XOR Gate
  - The XNOR Gate
- Drawing Logic Circuit
- Analysing Logic Circuit

- Universal Gates: NAND and NOR
  - ❖ NAND Gate
  - ❖ NOR Gate
- Implementation using NAND Gates
- Implementation using NOR Gates
- Implementation of SOP Expressions
- Implementation of POS Expressions
- Positive and Negative Logic
- Integrated Circuit Logic Families

Logic Gates

- **Gate Symbols**

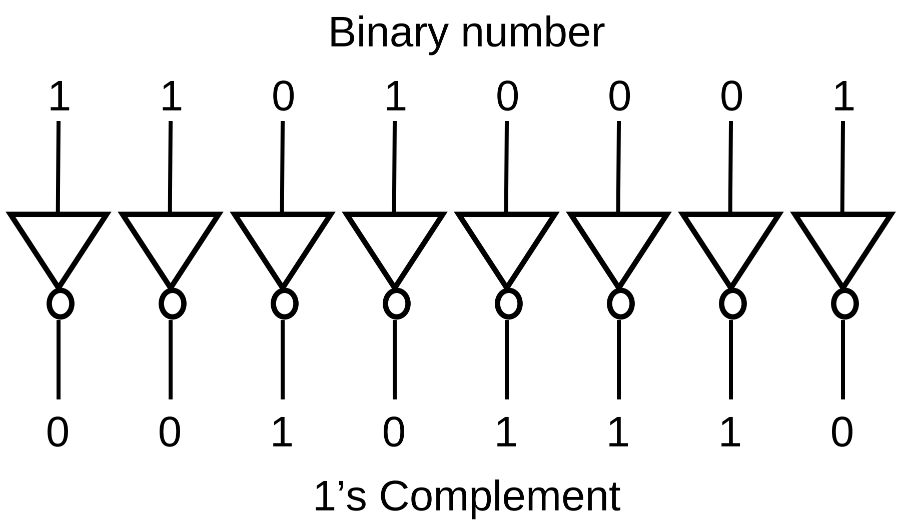


Logic Gates: The Inverter

- The **Inverter**

A	A'
0	1
1	0

- Application of the inverter: complement.



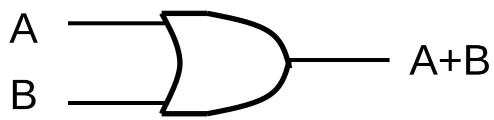
Logic Gates: The AND Gate (1/2)

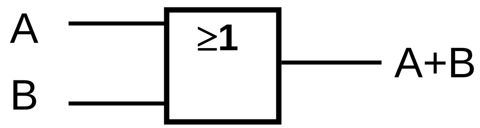
- The **AND** Gate

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gates: The OR Gate

- The **OR** Gate

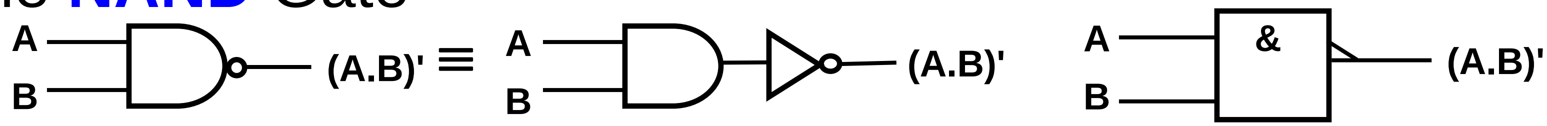




A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates: The NAND Gate

- The **NAND** Gate

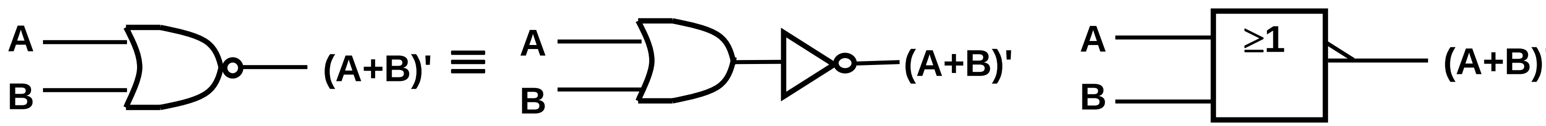


A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

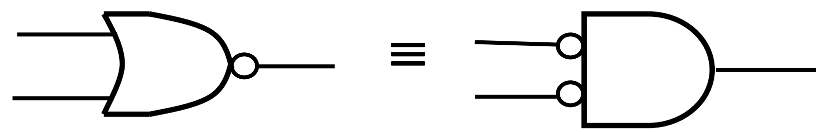
NAND　　Negative-OR

Logic Gates: The NOR Gate

- The **NOR** Gate



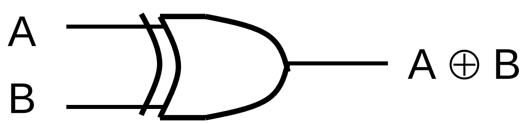
A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

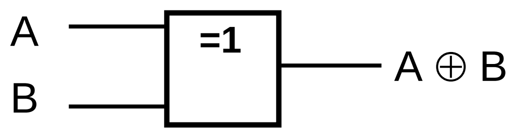


NOR Negative-AND

Logic Gates: The XOR Gate

- The **XOR** Gate

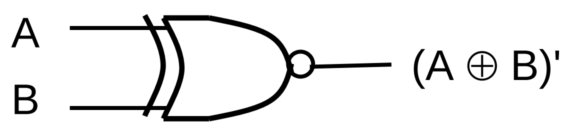


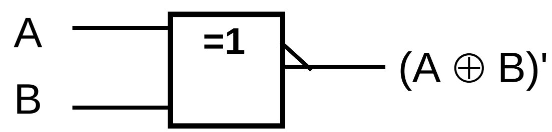


A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Logic Gates: The XNOR Gate

- The **XNOR** Gate





A	B	$(A \oplus B)'$
0	0	1
0	1	0
1	0	0
1	1	1