



## Lezione 3

### *Algebra di Boole e circuiti logici*

*A. Borghese, F. Pedersini  
Dip. Scienze dell'Informazione  
Università degli Studi di Milano*

## Sommario



- ❖ **Algebra di Boole**
  - Variabili ed operatori semplici.
  
- ❖ Implementazione circuitale delle porte logiche



George Boole, 1854:

*"An Investigation of the Laws of Thought on which to found the Mathematical Theories of Logic and Probabilities"*

## Algebra Booleana

- ❖ **Variabili** binarie: FALSE(=0); TRUE(=1)
- ❖ **Operatori** logici sulle variabili: NOT, AND, OR
  
- ❖ **Applicazioni:**
  - **Analisi** dei circuiti digitali
    - ✦ Descrizione del funzionamento in modo economico.
  - **Sintesi** (progettazione) dei circuiti digitali
    - ✦ Data una certa funzione logica, svilupparne una implementazione efficiente.

## Operatore NOT



- ❖ **Operazione logica di negazione**
  - Se A è vera, NOT(A) è falsa

$$Y = \text{NOT } A = \bar{A}$$

- ❖ **Operazione definita dalla **tabella della verità****
  - Funzione definita per tutte le combinazioni di variabili

### Tabella della verità

A	Y
0	1
1	0



Negazione logica  
("Inverter")

## Operatore **AND**



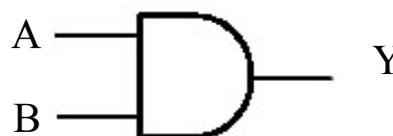
### ❖ Operazione di **prodotto logico**

- Solo se sia A che B sono veri, **A AND B** è vera.

$$Y = A \text{ AND } B = A \cdot B = AB$$

### Tabella della verità

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



*Prodotto logico  
(porta AND)*

## Operatore **OR**



### ❖ Operazione di **somma logica**

- Se A o B sono veri, che **A OR B** è vera.

$$Y = A \text{ OR } B = A + B$$

### Tabella della verità

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



*Somma logica  
(porta "OR")*



### ❖ Priorità

- In assenza di parentesi, AND ha la priorità sull'OR ed il NOT su entrambi:

**NOT → AND → OR**

### ❖ Esempi:

$$A \text{ OR } B \text{ AND } C = A + B \cdot C = A + (B \cdot C)$$

$$\text{NOT } A \text{ AND } C = \text{NOT } A \cdot C = (\text{NOT } A) \cdot C = \bar{A} \cdot C$$



### ❖ Le proprietà **commutativa**, **distributiva**, **identità**, **inverso** sono postulati: assunti veri per definizione

- Le altre proprietà sono teoremi dimostrabili

### ❖ Nell'algebra di Boole vale il: **principio di dualità**

### ❖ Il DUALE di una funzione booleana si ottiene:

- scambiando **AND** con **OR**  
(OR→AND , AND→OR)
- scambiando **TRUE (1)** con **FALSE (0)**  
(0→1 , 1→0)



- Identità
- Elemento 0
- Idempotenza
- Inverso
- Commutativa
- Associativa

## AND

$$\begin{aligned}
 1 \cdot x &= x \\
 0 \cdot x &= 0 \\
 x \cdot x &= x \\
 x \cdot \sim x &= 0 \\
 x \cdot y &= y \cdot x \\
 (x \cdot y) \cdot z &= x \cdot (y \cdot z)
 \end{aligned}$$

## OR (duale)

$$\begin{aligned}
 0 + x &= x \\
 1 + x &= 1 \\
 x + x &= x \\
 x + \sim x &= 1 \\
 x + y &= y + x \\
 (x + y) + z &= x + (y + z)
 \end{aligned}$$

- Distributiva
- Assorbimento

## AND rispetto OR

$$\begin{aligned}
 x \cdot (y + z) &= x \cdot y + x \cdot z \\
 x \cdot (x + y) &= x
 \end{aligned}$$

## OR rispetto a AND

$$\begin{aligned}
 x + y \cdot z &= (x + z) \cdot (x + y) \\
 x + x \cdot y &= x
 \end{aligned}$$

- De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

# Operatore NAND



## ❖ Operatore AND negato

$$A \text{ NAND } B = \text{NOT}(A \text{ AND } B)$$

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

operatore "NAND"



=





## ❖ Operatore **OR** negato

$$A \text{ NOR } B = \text{NOT}( A \text{ OR } B )$$

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

*operatore "NOR"*



=



# Porte universali



*Quale è il numero minimo di porte con cui è possibile implementare tutte le altre?*

- ❖ Con la legge di De-Morgan riusciamo a passare da 3 a 2:
  - con NOT e AND si ottiene OR:

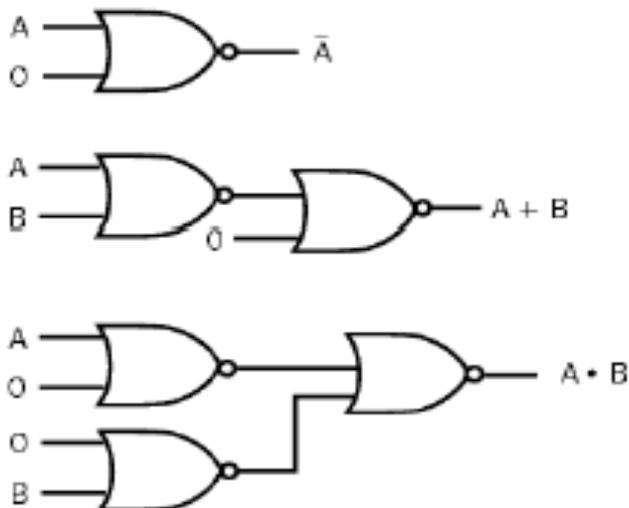
$$\text{NOT}( \text{NOT}(A) \text{ AND } \text{NOT}(B) ) = A \text{ OR } B$$

- ❖ **E' possibile usarne una sola?**
  - Sì, ad esempio la porta **NAND** e la **NOR** che sono chiamate **porte universali**



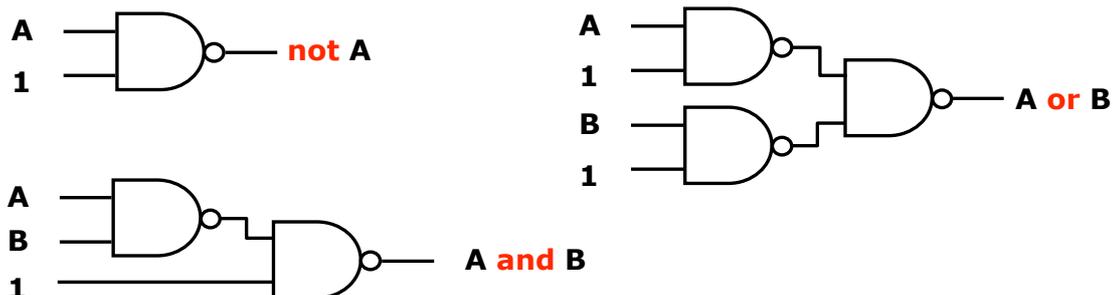
## Porta Universale NOR

$$\begin{aligned} \text{NOT } A &= 0 \text{ NOR } A = A \text{ NOR } A \\ A \text{ OR } B &= (A \text{ NOR } B) \text{ NOR } 0 \\ A \text{ AND } B &= (A \text{ NOR } 0) \text{ NOR } (B \text{ NOR } 0) \end{aligned}$$



## Porta Universale NAND

$$\begin{aligned} \text{NOT } A &= 1 \text{ NAND } A = A \text{ NAND } A \\ A \text{ AND } B &= (A \text{ NAND } B) \text{ NAND } 1 \\ A \text{ OR } B &= (A \text{ NAND } 1) \text{ NAND } (B \text{ NAND } 1) \end{aligned}$$





- ❖ *Usare la sola porta NAND per realizzare AND, OR e NOT e disegnarne gli schemi logici*
- ❖ *Dimostrare le seguenti uguaglianze:*  
$$A + \sim AB = A + B$$
$$(A + \sim B)(B + C) = AB + AC + \sim BC$$
*usando le proprietà dell'algebra di Boole.*
- ❖ *Calcolare le TT per le seguenti funzioni:*  
$$DA + AC + \sim B$$
$$A + B + C + D$$
$$\sim D \sim ABC + \sim DABC + \sim D \sim AB \sim C + \sim DAB \sim C$$
- ❖ *Trasformare in funzioni equivalenti le seguenti:*  
$$\sim(ABCD)$$
$$\sim(DA) + \sim(B + \sim C)$$



Algebra di Boole

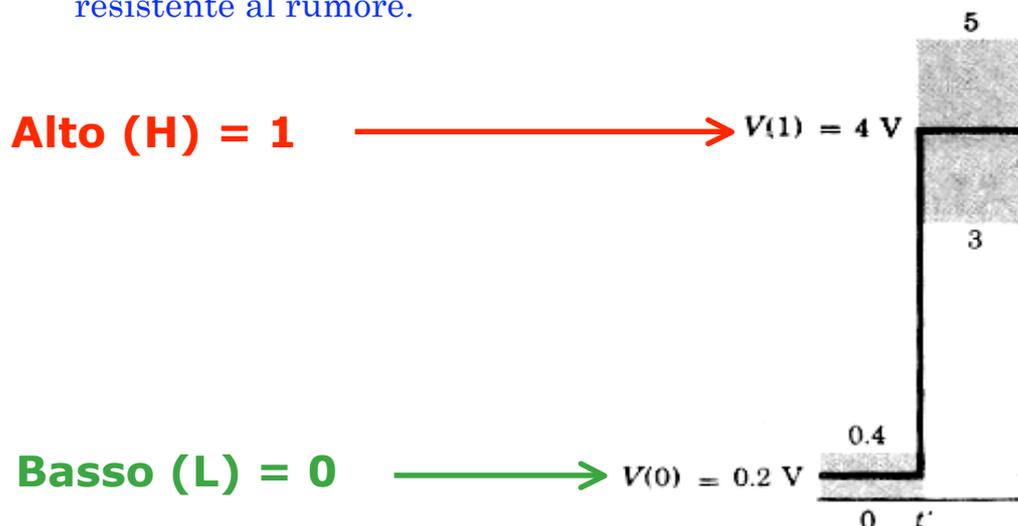
Variabili ed operatori semplici.

Implementazione circuitale delle porte logiche



❖ 2 soli stati stabili (livelli di tensione):

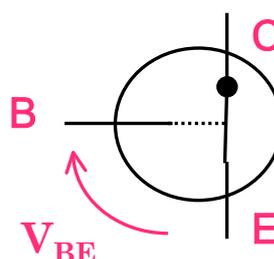
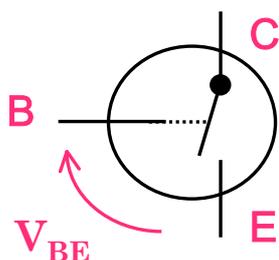
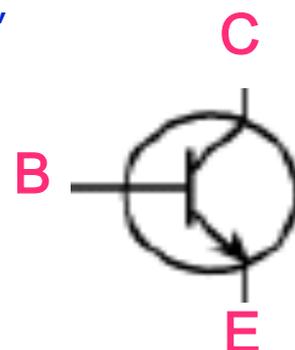
- **ALTO (H) = 1**     $3 \text{ Volt} < V_H < 5 \text{ Volt (TTL)}$
- **BASSO (L) = 0**     $0 \text{ Volt} < V_L < 0,4 \text{ Volt}$
- 2 range di tensioni, separati da un grosso gap, per essere resistente al rumore.



## Il Transistor



- ❖ Modello: interruttore tra **Emettitore** e **Collettore**, comandato dalla tensione sulla **Base**.
- ❖ 2 casi "estremi":
  - Tensione  $V_{BE}$  **bassa** → **C, E isolati**
    - ✦ Transistor in stato di **INTERDIZIONE**
  - Tensione  $V_{BE}$  **alta** → **C, E collegati**
    - ✦ Transistor in stato di **SATURAZIONE**



# INVERTER (porta NOT)

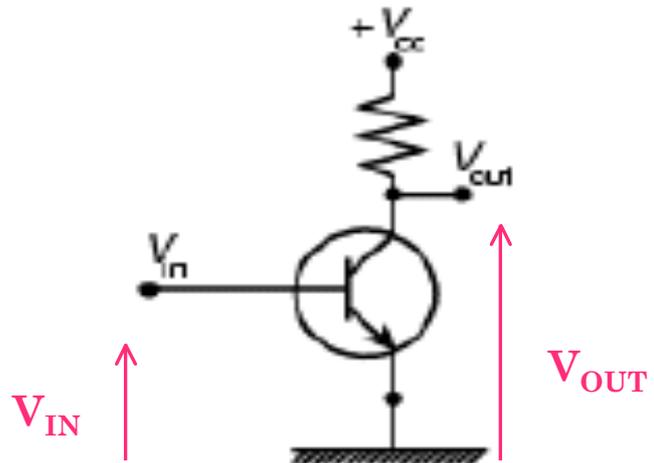


Funzionamento:

$V_{IN} = V_H \rightarrow V_{BE}$  alta  $\rightarrow$  E e C collegati  $\rightarrow V_{OUT} = V_L$   
 $V_{IN} = V_L \rightarrow V_{BE}$  bassa  $\rightarrow$  E e C scollegati  $\rightarrow V_{OUT} = V_H$

Tabella della verità

$V_{IN}$	$V_{OUT}$
$V_H = 1$	$V_L = 0$
$V_L = 0$	$V_H = 1$



# Porta NAND

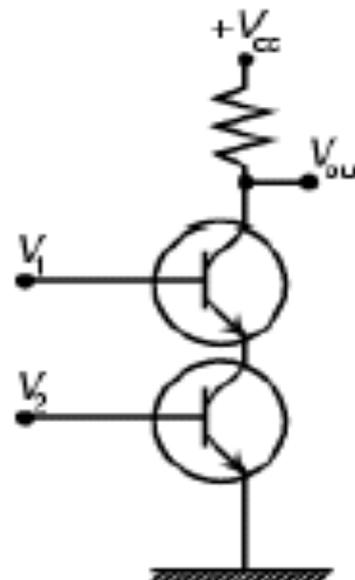


Funzionamento:

Solo se:  $V_1 = V_2 = V_H \rightarrow V_{OUT} = V_L$   
 altrimenti:  $\rightarrow V_{OUT} = V_H$

Tabella della verità

$V_1$	$V_2$	$V_{OUT}$
$V_H = 1$	$V_H = 1$	$V_L = 0$
$V_H = 1$	$V_L = 0$	$V_H = 1$
$V_L = 0$	$V_H = 1$	$V_H = 1$
$V_L = 0$	$V_L = 0$	$V_H = 1$



# Porta NOR

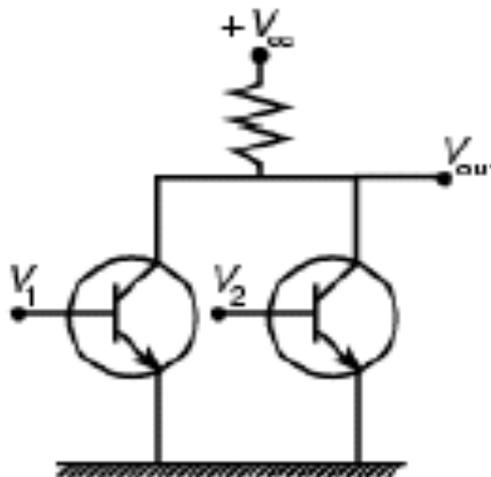


Funzionamento:

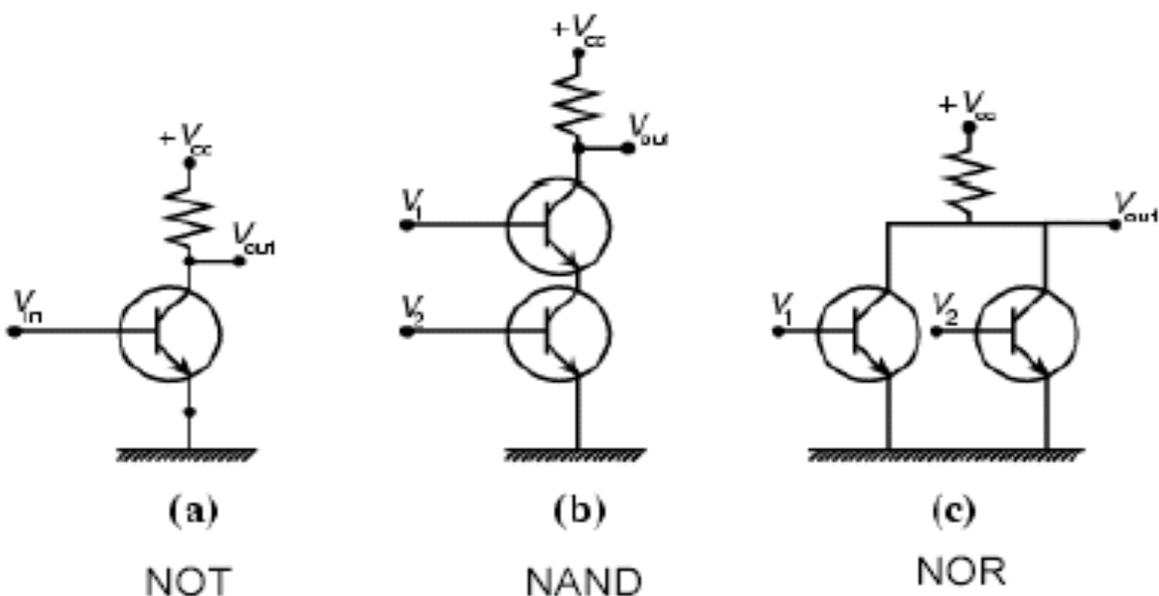
Se  $V_1$  o  $V_2 = V_H$  →  $V_{OUT} = V_L$   
altrimenti →  $V_{OUT} = V_H$

Tabella della verità

$V_1$	$V_2$	$V_{OUT}$
$V_H=1$	$V_H=1$	$V_L=0$
$V_H=1$	$V_L=0$	$V_L=0$
$V_L=0$	$V_H=1$	$V_L=0$
$V_L=0$	$V_L=0$	$V_H=1$



# Circuiti di base





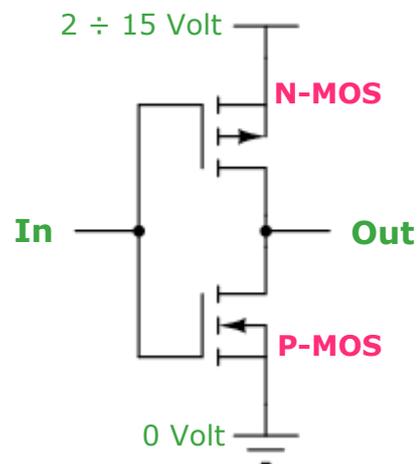
## ❖ CMOS: Complementary-MOS

- MOS: Metal – Oxide Semiconductor
- MOS complementari (N-MOS + P-MOS) che lavorano “in coppia”

## ❖ Vantaggi:

- Tensione di alimentazione “flessibile”:
  - $V_{CC} = 2 \div 15$  Volt
  - $V_{LOW} = 0 \div V_{CC}/2$
  - $V_{HIGH} = V_{CC}/2 \div V_{CC}$
- Consumo bassissimo:
  - ✦ Consuma solo nella transizione
  - ✦ In condizioni statiche, consumo nullo!

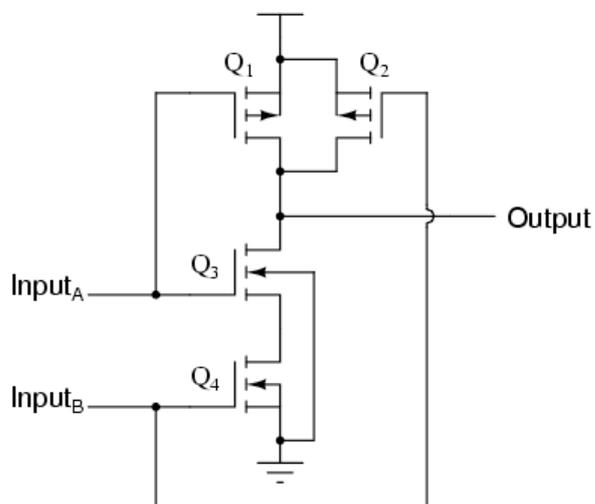
## Inverter CMOS



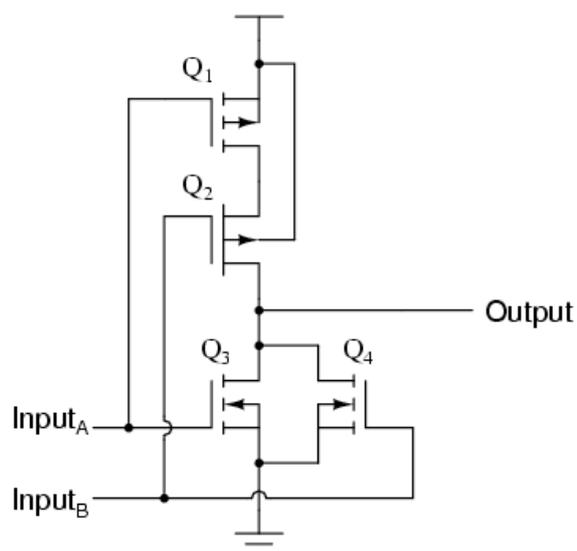
# Porte CMOS



## Porta NAND



## Porta NOR

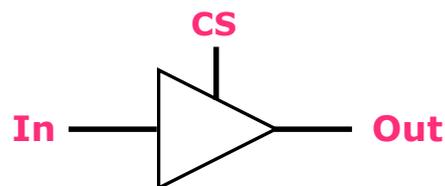




## Logica **three-state**

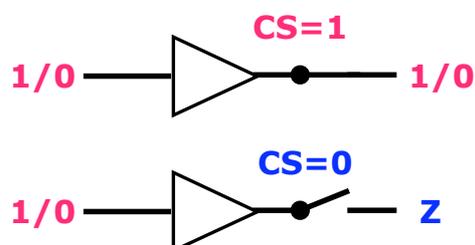
### ❖ 3 stati:

- **0** = “LOW”
- **1** = “HIGH”
- **Z** = “uscita scollegata”



### ❖ Ingresso di abilitazione: **CS**

- CS = 1 → Porta attiva
- CS = 0 → ALTA IMPEDENZA (uscita scollegata)



### ❖ È possibile scollegare l'uscita dal resto del circuito

**Vantaggio:** permette di collegare **più uscite tra loro** evitando conflitti.

CS	In	Out
0	0	<b>Z</b>
0	1	<b>Z</b>
1	0	<b>0</b>
1	1	<b>1</b>