



I circuiti binari: definizione delle funzioni logiche

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borgnese@dsi.unimi.it

Università degli Studi di Milano



Sommario

Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.



Le operazioni logiche fondamentali



NOT

AND

OR



Circuiti Booleani



“An Investigation of the Laws of Thought on Which to Found the Mathematical Theories of Logic and Probabilities” G. Boole, 1854: approccio alla logica come algebra.

Variabili (binarie, 0 = FALSE; 1 = TRUE).

Operazioni sulle variabili (NOT, AND, OR).

Utilizzo dell'algebra Booleana per:

Analisi dei circuiti. Descrizione della funzione logica implementata dai circuiti.

Semplificazione di espressioni logiche per ottenere implementazioni efficienti.

Progettazione (sintesi) dei circuiti digitali. Data una certa funzione logica, sviluppare il circuito digitale che la implementa.

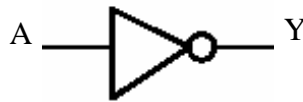


Operatore NOT



Tabella della verità

A	Y
0	1
1	0



“Inverter logico” : se **A** è vero (**TRUE=1**),
NOT A è falso (**FALSE=0**)

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

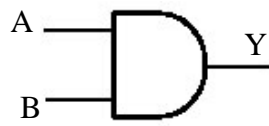


Operatore AND



Tabella della verità

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



“Prodotto logico”

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



Operatore OR



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

Tabella della verità

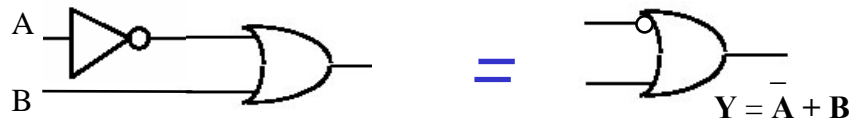
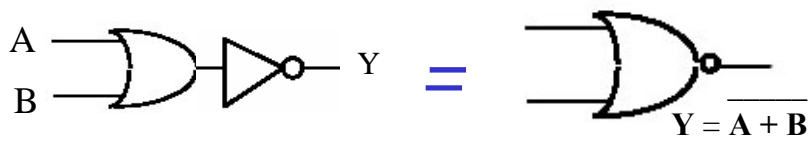


“Somma logica”

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



Concatenazione del NOT



Inserire un cerchietto all'ingresso corrisponde a negare la variabile in ingresso.
Inserire un cerchietto all'uscita corrisponde a negare (complementare) l'uscita.

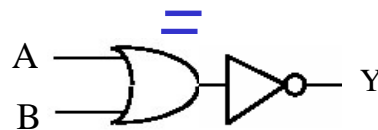
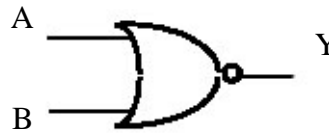


Operatore NOR



A	B	OR(A,B)	Y
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Operatore OR negato



"Not(Or(A,B))"

$$Y = \overline{A + B}$$

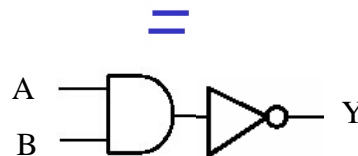
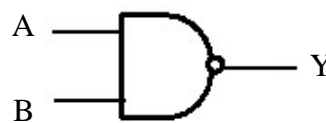


Operatore NAND



A	B	AND(A,B)	Y
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Operatore AND negato



"Not(And(A,B))"



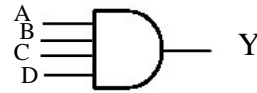
Porte logiche a più ingressi



- Rappresentano circuiti che forniscono in uscita il risultato di operazioni logiche elementari sui valori di tutte le variabili in ingresso
- Le variabili in ingresso possono essere n.

Ad esempio:

$$Y = A \text{ AND } B \text{ AND } C \text{ AND } D$$

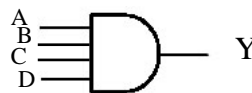


Porte logiche: tabella della verità



A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	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	0
1	1	1	1	1

$$Y = A \text{ AND } B \text{ AND } C \text{ AND } D$$





Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

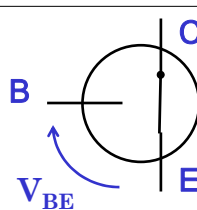
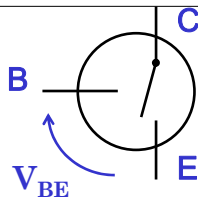
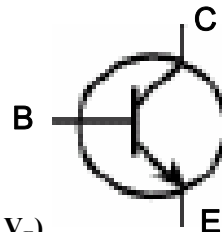
Algebra Booleana.



Il Transistor

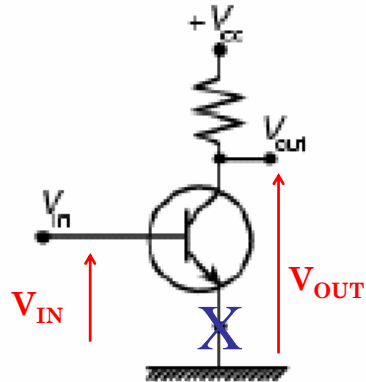


- Modello: interruttore tra **Emettitore** e **Collettore**, comandato dalla tensione sulla **Base**.
- 2 casi “estremi”:
 - Tensione V_{BE} **bassa** \rightarrow **C,E isolati**
 - Transistor in stato di **INTERDIZIONE**
 - Tensione V_{BE} **alta** \rightarrow **C,E collegati**
 - Transistor in stato di **SATURAZIONE** ($V_C = V_E$)





Inverter logico: porta NOT



$V_{in} = 0V$ è spento, $V_{out} = V_{CC}$

$V_{in} = V_{CC}$ passa corrente, la resistenza è molto bassa e $V_{out} \cong 0$

Si definisce porta logica (**gate**), un dispositivo elettronico in grado di trasformare la tensione agli ingressi secondo gli operatori fondamentali.



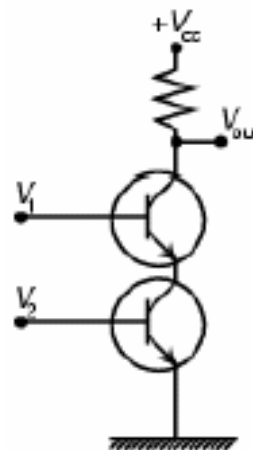
Porta NAND



- **Solo se** $V_1=V_2=V_H$ → I due transistor sono chiusi e passa corrente, $V_{OUT} = V_L$
- **Altrimenti** → $V_{OUT} = V_H$

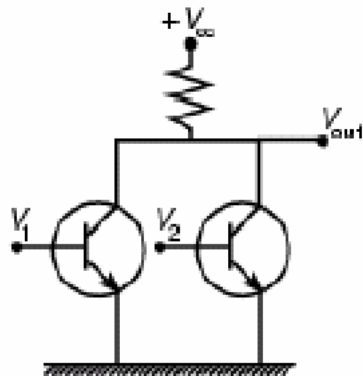
Tabella della verità

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





Porta NOR



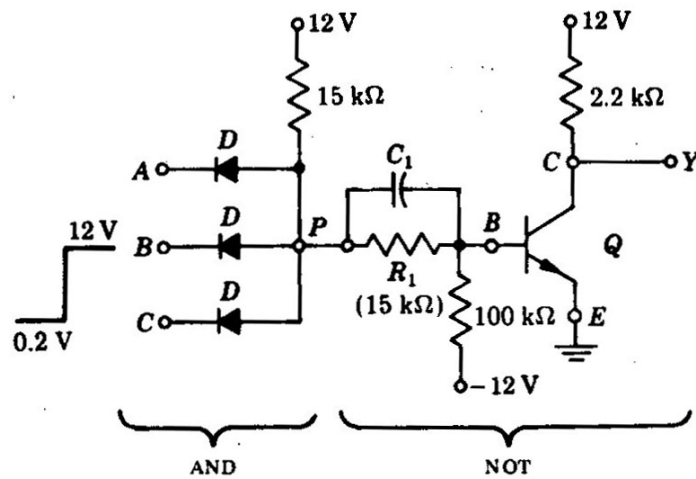
Se V_i è alto, il transistor corrispondente, conduce e la tensione V_{out} si avvicina alla massa ($V_{out} = \text{Low}$).

Se $V_1 = V_2 = 0$ nessun transistor conduce, e V_{out} viene "tirata" (pull-up) verso la tensione dell'alimentazione.

(c)
NOR



NAND in logica RTL



Porta NAND a 3 ingressi in Resistor Transistor Logic (RTL).

NAND in logica TTL

ABC = High \Rightarrow Q_1 interruttore aperto
 Q_2 chiuso $\Rightarrow P'' > 0V \Rightarrow Q_3 =$ chiuso
 $\Rightarrow Y =$ Low

$V_{CC} (5V)$

A|B|C = Low $\Rightarrow Q_1$ interruttore chiuso
 Q_2 aperto $\Rightarrow P'' = 0V \Rightarrow Q_3 =$ aperto
 $\Rightarrow Y =$ High.

Diodi di protezione

AND

NOT

A.A. 2004-2005 19/38 http://homes.dsi.unimi.it/~borgnese

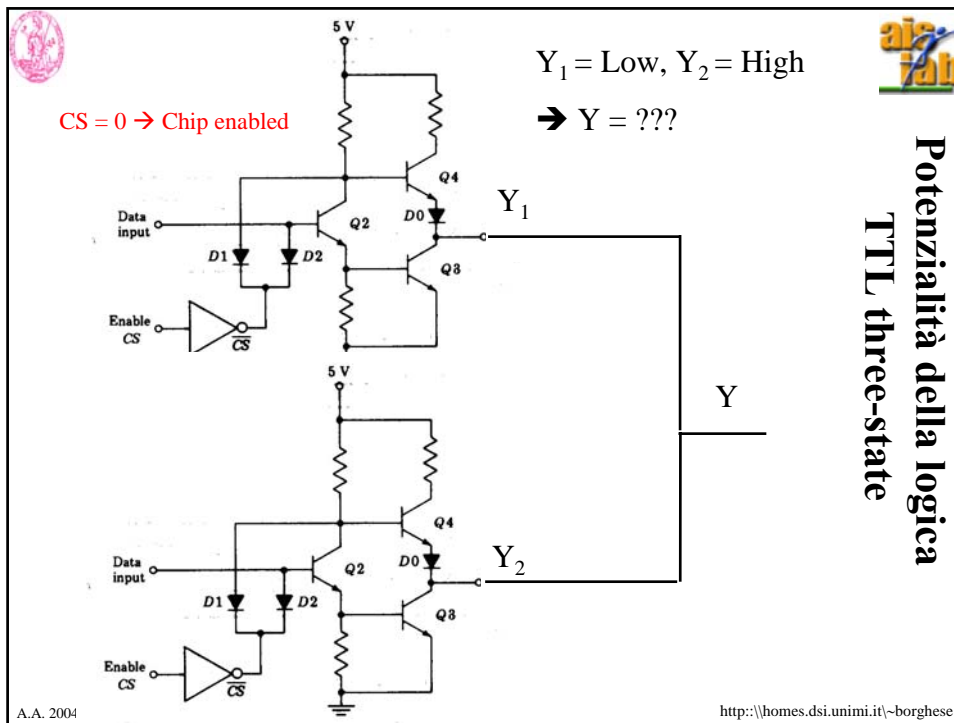
Logica TTL three-state

Stadio di output della porta NAND
 (configurazione TotemPole)
 con il segnale di abilitazione.

$CS = 1 \rightarrow D_1$ e D_2 chiusi \rightarrow
 Q_2 e Q_4 spenti \rightarrow Output
 Floating (nè collegato
 all'alimentazione, nè alla
 terra) \rightarrow Chip disabled

$CS = 0 \rightarrow D_1$ e D_2 aperti \rightarrow Chip enabled

A.A. 2004-2005 20/38 http://homes.dsi.unimi.it/~borgnese



Potenzialità della logica
 TTL three-state

Tecnologia a 3 stati

- 0 = Tensione low
- 1 = Tensione high
- Enable = Porta logica non funzionante. Uscita indifferente.
 Particolarmente utile per le memorie.

A.A. 2004-2005 http://homes.dsi.unimi.it/~borgnese

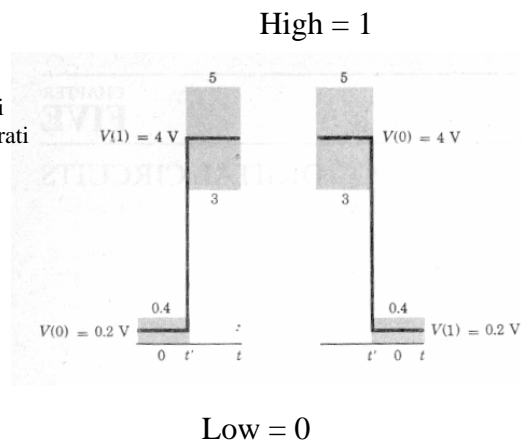


Perchè l'elettronica digitale funziona?



Perchè è progettata per essere resistente al rumore.

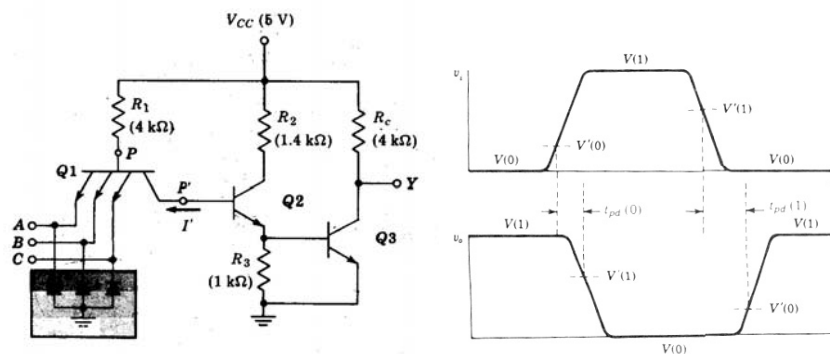
Vengono definiti 2 range di tensioni associati ai valori alto e basso, separati da un gap. Per la logica TTL:



Tempo di commutazione



La commutazione non è istantanea:





Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.



Funzioni logiche



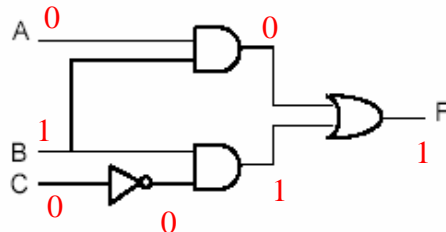
- La funzione calcolata da un circuito con n ingressi.
- Il circuito sarà costituito da un'opportuna combinazione di porte semplici (NOT, AND, OR).
- Per ciascuna delle 2^n combinazioni degli ingressi, può essere calcolata l'uscita.
- Il valore della funzione può essere rappresentato in 3 modi:
 - Circuito
 - Tabellato: tabelle di verità (Truth Table, TT).
 - Espressione simbolica



Dal circuito alla funzione logica



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



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



Dall'espressione logica alla tabella della verità



- Data l'espressione: $F = (A \text{ AND } B) \text{ OR } (B \text{ AND NOT}(C))$

Ricaviamo la tabella delle verità:

A	B	C	A and B	B and not(C)	F
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

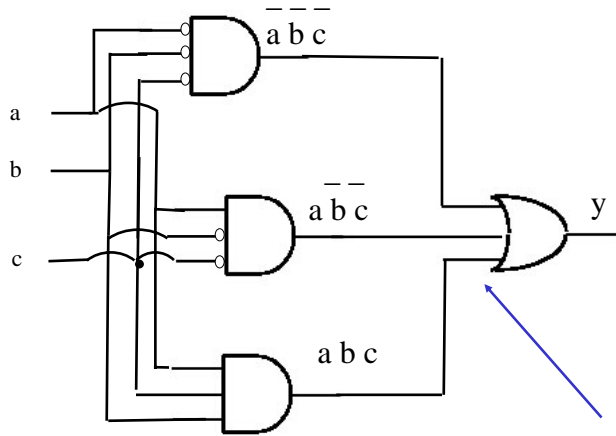


Dal circuito alla funzione logica



Esempio:

a b c	y
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1



Implementazione circuitale possibile.
Non è l'unica!

Funzione e tabella
coincidono

$$y = \bar{a} \bar{b} c + a \bar{b} c + a b \bar{c}$$



Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.



Concatenazione degli operatori



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

$$A + B \cdot C = A + (B \cdot C)$$

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



Regole algebriche



	AND	OR
Identità	$1 x = x$	$0 + x = x$
Elemento nullo	$0 x = 0$	$1 + x = 1$
Idempotenza	$x x = x$	$x + x = x$
Inverso	$x \sim x = 0$	$x + \sim x = 1$
Commutativa	$x y = y x$	$x + y = y + x$
Associativa	$(x y) z = x (y z)$	$(x + y) + z = x + (y + z)$
	AND rispetto ad OR	OR rispetto ad AND
Distributiva	$x (y + z) = x y + x z$	$x + y z = (x + y) (x + z)$
Assorbimento	$x (x + y) = x$	$x + x y = x$
De Morgan	$\sim (x y) = \sim x + \sim y$	$\sim (x + y) = \sim x \sim y$
	$\overline{xy} = \bar{x} + \bar{y}$	$\overline{x + y} = \bar{x} \bar{y}$

Si possono dimostrare sostituendo 0/1 alle variabili.



Teoremi di De Morgan

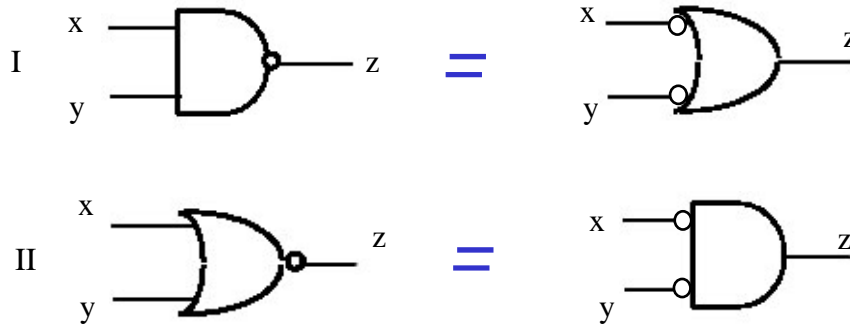


De Morgan

$$\sim (x y) = \sim x + \sim y \quad \sim (x+y) = \sim x \sim y$$

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

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



Principio di dualità



- Nell'algebra di Boole vale il principio di dualità.
- Il duale di una funzione booleana si ottiene sostituendo AND ad OR, OR ad AND, gli 0 agli 1 e gli 1 agli 0.

Esempi:

Identità	$1 x = x$	$0 + x = x$
Elemento nullo	$0 x = 0$	$1 + x = 1$

- Le proprietà commutativa, distributiva, identità, inverso sono postulati: assunti veri per definizione.
- Le altre proprietà sono teoremi dimostrabili.



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. es.: 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?

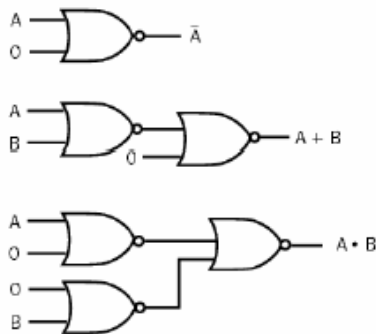
- Sì, ad esempio la porta NAND, o la NOR che sono chiamate **porte universali**.



Porta Universale NOR



- NOT A = 0 NOR A
- A OR B = (A NOR B) NOR 0
- A AND B = (A NOR 0) NOR (B NOR 0)





Esercizi



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

$$\sim(ABCD)$$

$$\sim(DA) + \sim(B + \sim C)$$



Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.