I circuiti binari: definizione delle funzioni logiche

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

Università degli Studi di Milano

Riferimenti al testo: Appendice B, sezioni B.1 e B.2

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 Thorught 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).

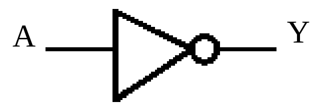
Utilizzo dell'algebra Booleana per:

- 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



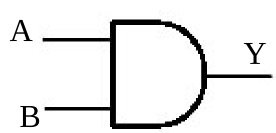
"Inverter logico" : se **A** *è vero (**TRUE**=1),*
NOT **A** *è falso (**FALSE**=0)*

NOT **A** = $\overline{\mathbf{A}}$

Operatore AND

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

Tabella della verità



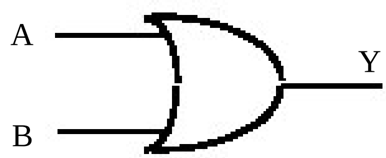
"Prodotto logico"

Y = A ***AND*** B = A · B = AB

Operatore OR

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

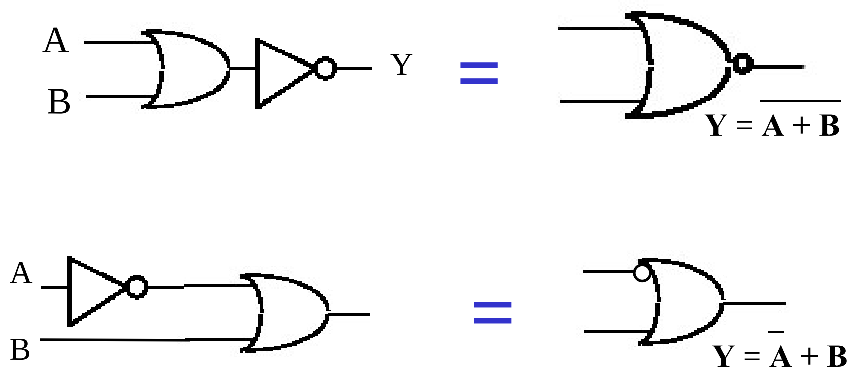
Tabella della verità



"Somma logica"

Y = A ***OR*** B = A + B

Concatenazione del NOT

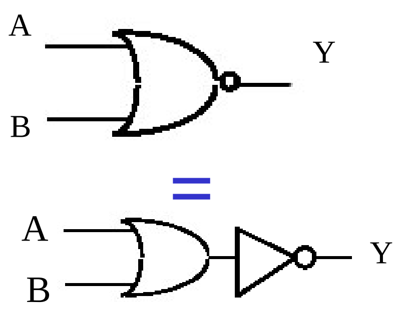


Inserire un cerchietto all'ingresso corrisponde a negare la variabile in ingresso.
Inserire un cerchietto all'uscita corrisponde a negare (complementare) l'uscita.

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



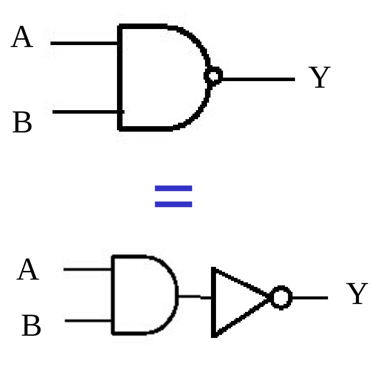
"Not(Or(A,B))"

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

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



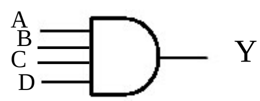
"Not(And(A,B))"

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 ingesso possono essere n.

Ad esempio:

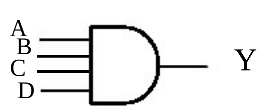
Y = A AND B AND C AND D



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 AND B AND C AND D



Sommario

Variabili ed operatori semplici.

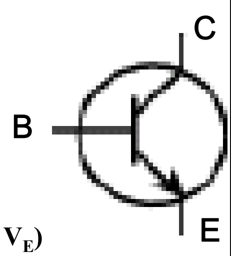
Implementazione circuitale (porte logiche).

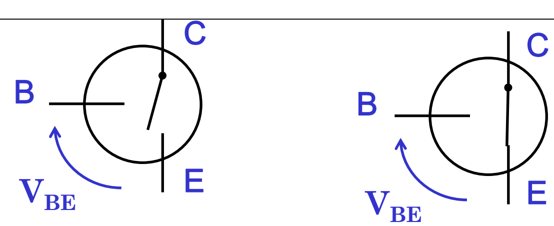
Dal circuito alla funzione.

Algebra Booleana.

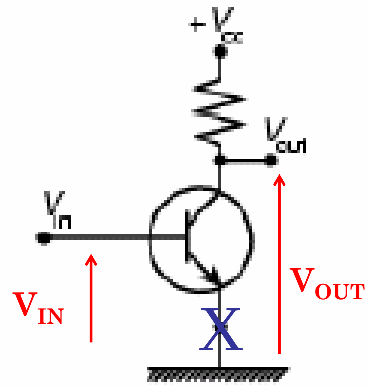
Il Transistor

- Modello: interruttore tra **Emettitore** e **Collettore**, comandato dalla tensione sulla **Base.**
- 2 casi "estremi":
  - Tensione V_{BE} **bassa → C,E isolati**
    - Transistor in stato di **INTERDIZIONE**
  - Tensione V_{BE} **alta → 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 $V_{out} \cong 0$

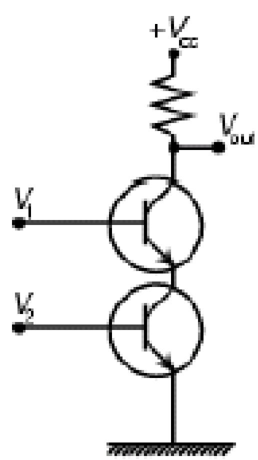
Si definisce porta logica (**gate**), un dispositivo elettronico in grado di trasformare la tensione agli ingressi secondo gli operatori fondamentali.

Porta NAND

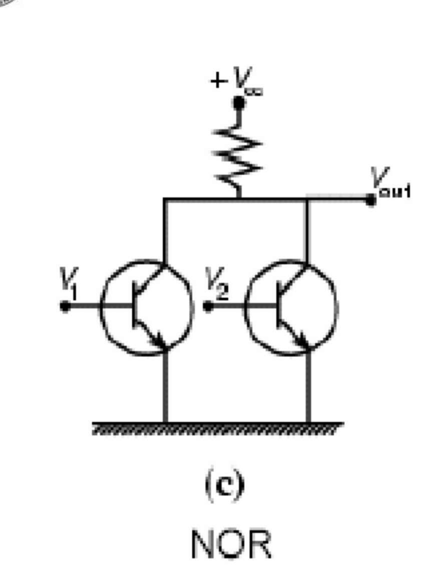
- **Solo se V_1=V_2= V_H** → I due transistor sono chiusi e passa corrente, $V_{OUT} = V_L$
- **Altrimenti** → $V_{OUT} = V_H$

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} = Low).

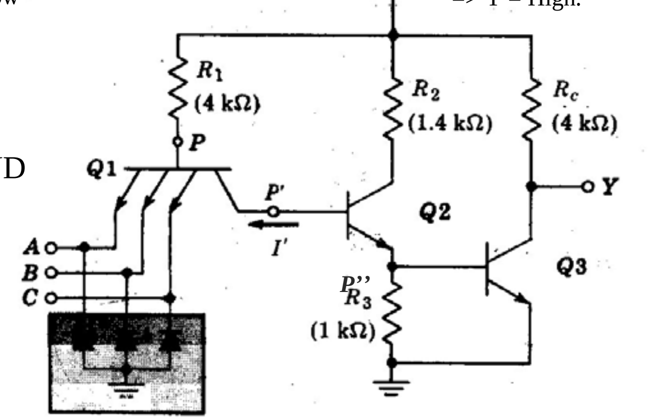
Se $V_1 = V_2 = 0$ nessun transistor conduce, e V_{out} viene "tirata" (pull-up) verso la tensione dell'alimentazione.

NAND in logica TTL

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

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

AND



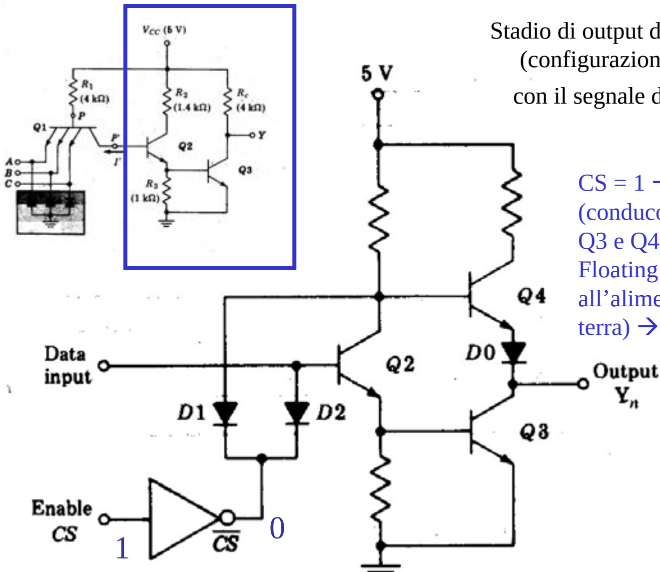
NOT

Diodi di protezione

Se A, B e C sono alti, Q_1 è spento e Q_2 acceso.
Se A, B o C sono bassi, Q_1 è acceso e Q_2 spento.

Logica TTL three-state

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



CS = 1 → D_1 e D_2 chiusi (conducono) → Q2 spento → Q3 e Q4 spenti → Output Floating (nè collegato all'alimentazione, nè alla terra) → Chip disabled

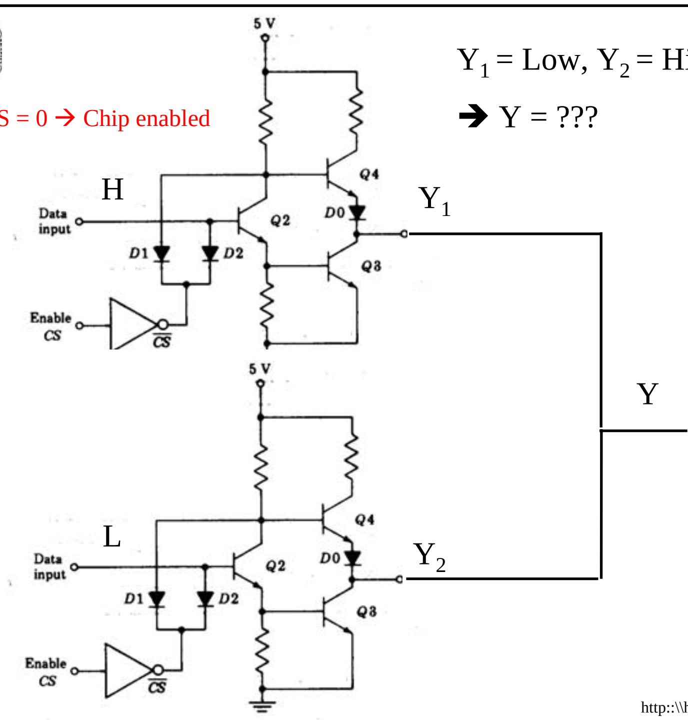
CS = 0 → D_1 e D_2 aperti → Chip enabled

Potenzialità della logica TTL three-state

CS = 0 → Chip enabled

Y_1 = Low, Y_2 = High

➔ Y = ???



Tecnologia a 3 stati

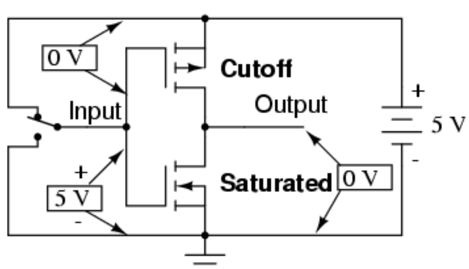
0 = Tensione low

1 = Tensione high

Enable = Porta logica non funzionante. Uscita indifferente. Particolarmente utile per le memorie.

Porte CMOS

- **CMOS**: Complementary–MOS
  - MOS: Metal – Oxide Semiconductor
  - MOS complementari (N-MOS + P-MOS) che lavorano "in coppia"
- Vantaggi:
  - Tensione di alimentazione "flessibile":
  - V_{CC} = 3 ÷ 15 Volt
  - V_{LOW} = 0 ÷ $V_{CC}/2$
  - V_{HIGH} = $V_{CC}/2$ ÷ V_{CC}
  - Consumo bassissimo:
    - Consuma solo nella transizione
    - In condizioni statiche, consumo nullo!

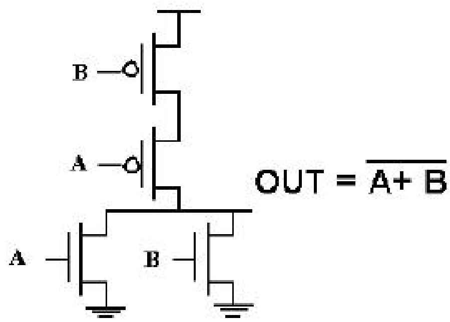


Input = "high" (1)

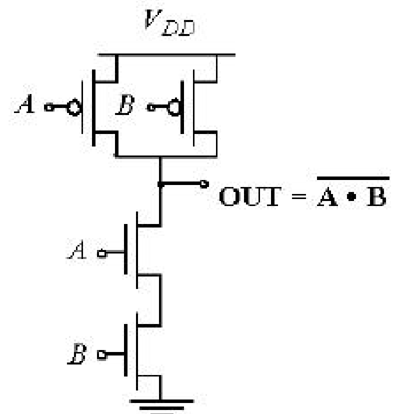
Output = "low" (0)

Porte CMOS

NOR



NAND

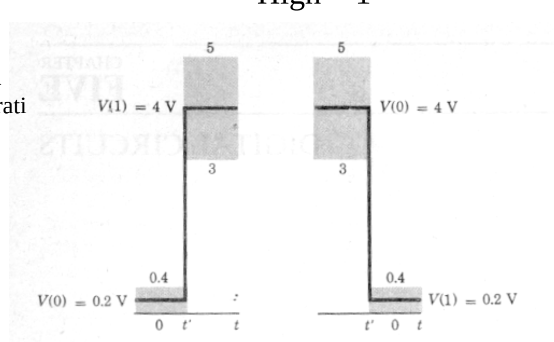


Perchè l’elettronica digitale funziona?

Perchè è progettata per essere resistente al rumore.

High = 1

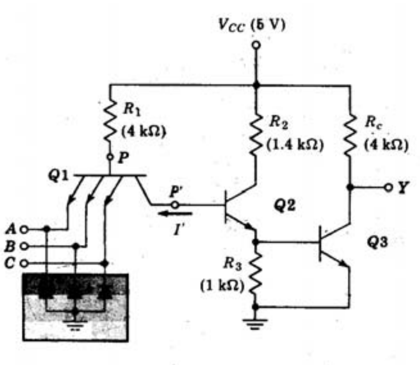
Vengono definiti 2 range di tensioni associati ai valori alto e basso, separati da un gap. Per la logica TTL:

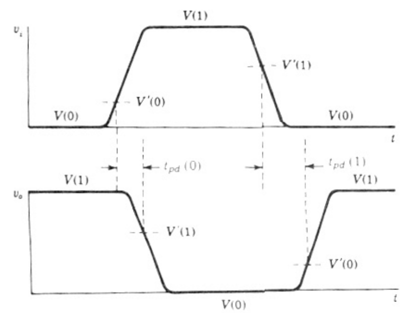


Low = 0

Tempo di commutazione

La commutazione non è istantanea:





Definizione del cammino critico nei circuiti combinatori.

Sommario

Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

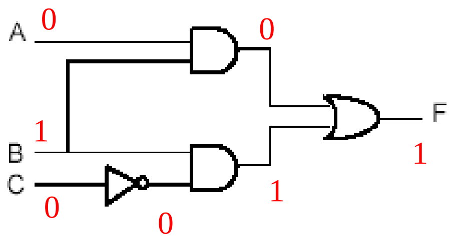
Algebra Booleana.

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 AND B) OR (B 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 AND B) OR (B AND NOT(C))

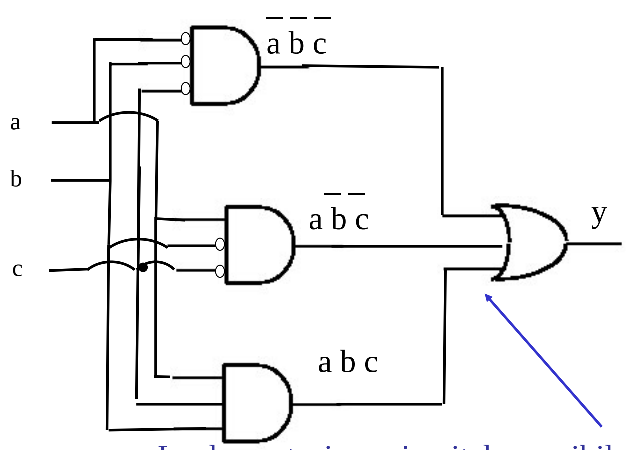
Ricaviamo la tabella delle verità:

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

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

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:

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

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

Regole algebriche

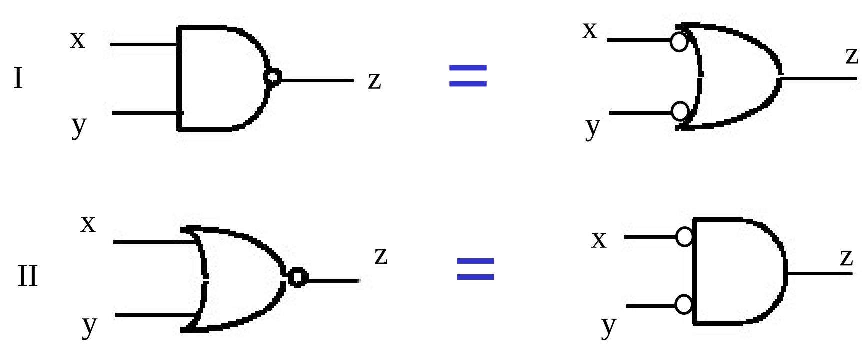
	AND	OR
Doppia Inversione	$\overline{\overline{x}} = x$	
Identità	$1\,x = x$	$0 + x = x$
Elemento nullo	$0\,x = 0$	$1 + x = 1$
Idempotenza	$x\,x = x$	$x + x = x$
Inverso	$x\,\overline{x} = 0$	$x + \overline{x} = 1$
Commutativa	$x\,y = y\,x$	$x + y = y + x$
Associativa	$(x\,y)\,z = x\,(y\,z)$	$(x + y) + z = x + (y + z)$
	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	$\overline{xy} = \overline{x} + \overline{y}$	$\overline{x + y} = \overline{x}\,\overline{y}$

Si possono dimostrare sostituendo 0/1 alle variabili.

Teoremi di De Morgan

De Morgan $\sim(x\,y) = \sim x + \sim y$ $\sim(x+y) = \sim x \sim y$

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



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.

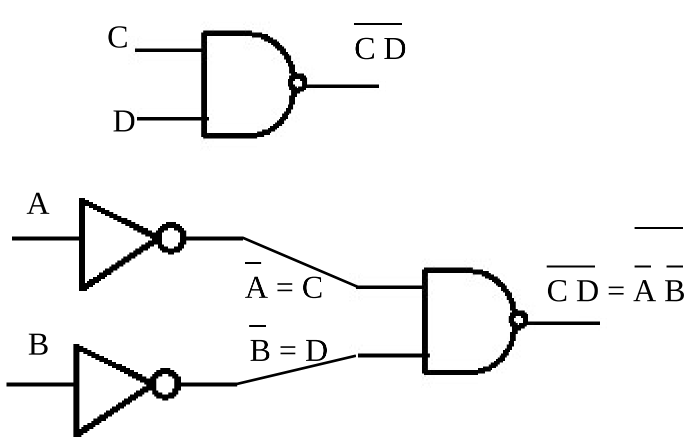
Esempi:

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.

Verso le porte universali



Porte Universali

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

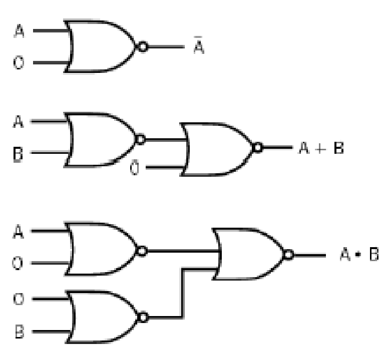
• Con la legge di De-Morgan riusciamo a passare da 3 a 2. es.: con NOT e AND (NAND) si ottiene OR:

NOT(NOT(A)AND(NOT(B))) = A 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 + ~AB = A + B
(A + ~B)(B + C) = AB + AC + ~BC
usando le proprietà dell'algebra di Boole.

• Calcolare le TT per le seguenti funzioni
DA + AC + ~B
A + B + C + D
~D~ABC + ~DABC + ~D~AB~C + ~DAB~C

• Trasformare in funzioni equivalenti le seguenti
~(ABCD)
~(DA) + ~(B + ~C)

Semplificazioni notevoli

Dimostrare che: $A + \bar{A}B = A + B$

Proprietà distributiva di OR rispetto ad AND:

$A + \bar{A}B = (A + B)(A + \bar{A})$

Sviluppando il prodotto:

$(A + B)(A + \bar{A}) = AA + A\bar{A} + BA + B\bar{A} = A + AB + \bar{A}B$

Raccogliendo A:

$A + AB + \bar{A}B = A + (A + \bar{A})B = A + B$

Dimostrare che: $(A + \bar{B})(B + C) = AB + AC + \bar{B}C$

Dimostrare che: $\bar{A} + AB = \bar{A} + B$

Esempio di semplificazione algebrica (esercizio)

$F = \bar{A}B\bar{C} + AB\bar{C} + ABC =$

Raccogliendo $B\bar{C}$:

$(\bar{A} + A)B\bar{C} + ABC =$

Proprietà dell'inverso: "$\bar{A} + A = 1$"

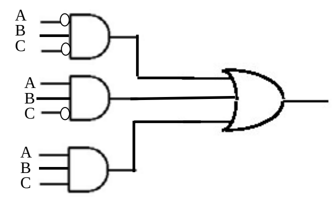
$= 1B\bar{C} + ABC =$

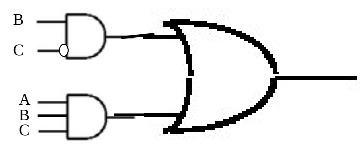
Proprietà dell'identità: "$1B = B$"

$= B\bar{C} + ABC =$

Dalla slide precedente:

$= B(\bar{C} + AC) = B(\bar{C} + A)$





Sommario

Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.