



I circuiti logici: definizione delle funzioni logiche

Prof. Alberto Borghese
Dipartimento di Informatica
borgnese@di.unimi.it

Università degli Studi di Milano

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



Sommario

Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.



Le operazioni logiche fondamentali



NOT

AND

OR

QUALUNQUE funzione booleana (logica) può essere espressa combinando opportunamente tre funzioni booleane elementari.

Si dice anche che AND, OR, NOT formano un set completo.



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).

Equivalenza tra operazioni logiche su proposizioni vere/false e operazioni algebriche su variabili binarie.

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



A	Y
0	1
1	0

Tabella della verità



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

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

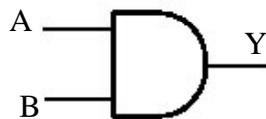


Operatore AND



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

Tabella della verità



“Prodotto logico”

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



Operatore OR



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

Tabella della verità

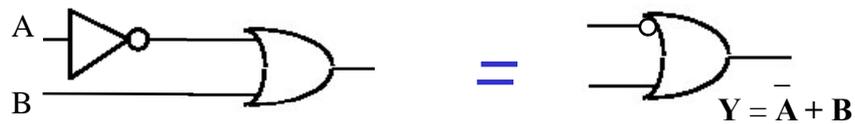
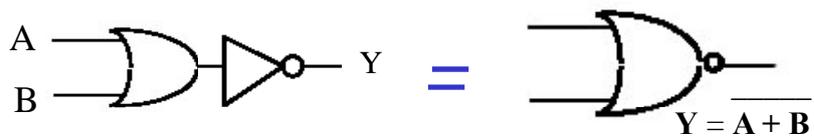


“Somma logica”

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



Concatenazione del NOT



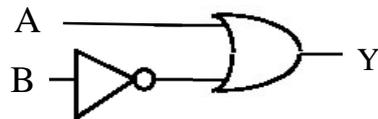
Inserire un cerchietto all'ingresso corrisponde a negare la variabile in ingresso.
Inserire un cerchietto all'uscita corrisponde a negare (complementare) l'uscita.



Funzione composta



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



“Or(A,!B)”

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

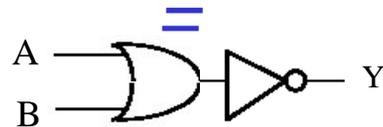


Operatore NOR



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))”

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



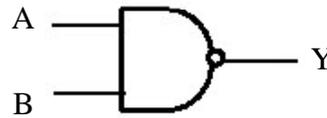
Operatore NAND



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

"Not(And(A,B))"

Operatore AND negato



=



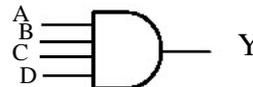
Porte logiche a più ingressi



- 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:

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



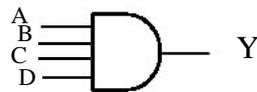


Porte logiche: tabella della verità



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.

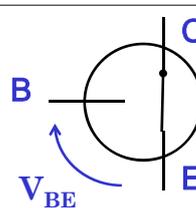
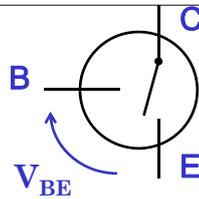
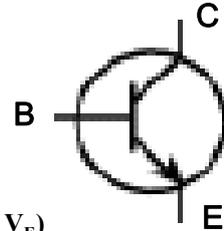
Algebra Booleana.



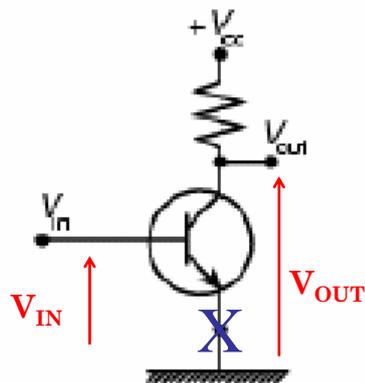
Il Transistor



- 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.



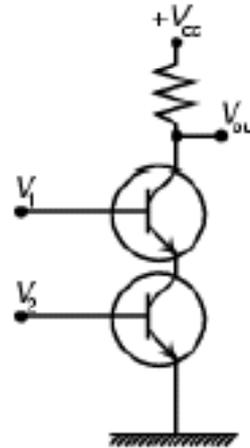
Porta NAND



- 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}
$V_H=1$	$V_H=1$	$V_L=0$
$V_H=1$	$V_L=0$	$V_H=1$
$V_L=0$	$V_H=1$	$V_H=1$
$V_L=0$	$V_L=0$	$V_H=1$

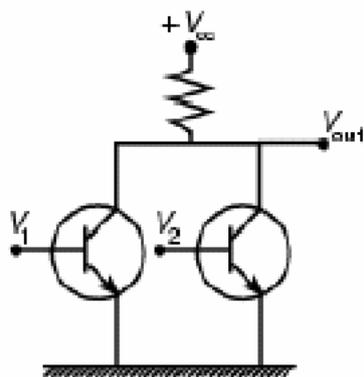


Porta NOR



Se V_1 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.



(c)
NOR

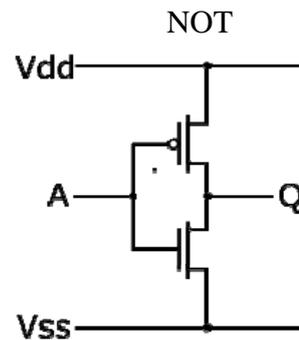
V_1	V_2	V_{OUT}
$V_H=1$	$V_H=1$	$V_L=0$
$V_H=1$	$V_L=0$	$V_L=0$
$V_L=0$	$V_H=1$	$V_L=0$
$V_L=0$	$V_L=0$	$V_H=1$



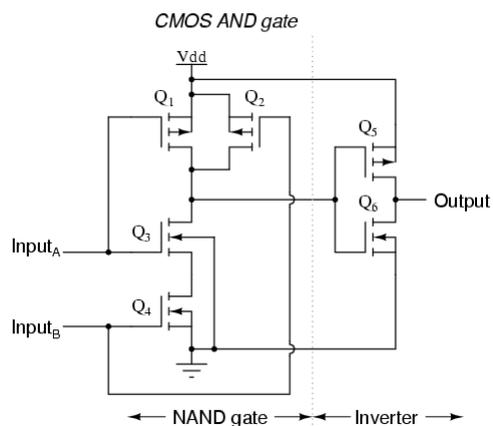
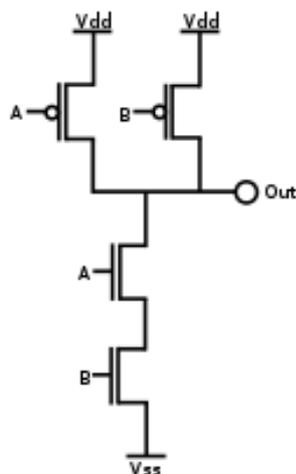
La tecnologia CMOS (1980 – oggi)



- **CMOS**: Complementary–MOS
 - MOS: **M**etal – **O**xide **S**emiconductor
 - MOS complementari (**N**-MOS + **P**-MOS) che lavorano “in coppia”: substrati comuni.
- Vantaggi:
 - Tensione di alimentazione “flessibile”:
 - $V_{CC} = 2 \div 15$ Volt
 - $V_{LOW} = 0 \div V_{SS}$
 - $V_{HIGH} = 1 \div V_{dd}$
 - Consumo bassissimo:
 - Consuma solo nella transizione
 - In condizioni statiche, consumo praticamente nullo!

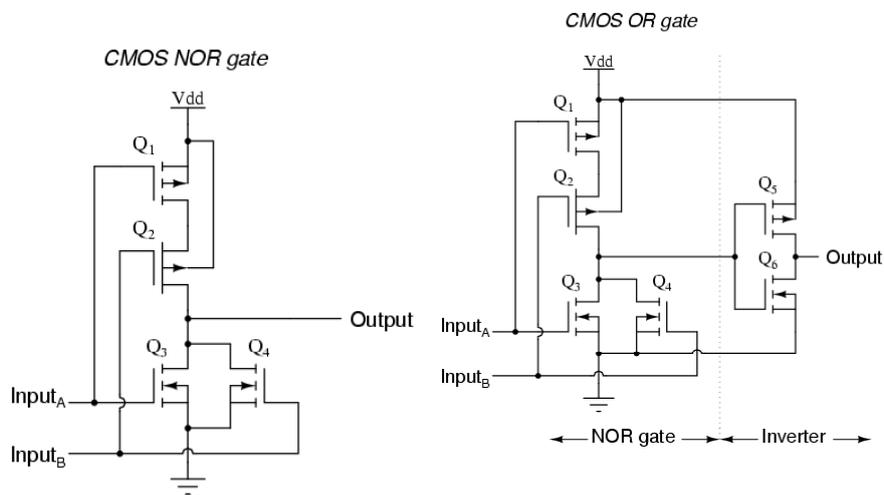


Porta NAND e AND in C-MOS





PORTA NOR e OR IN CMOS

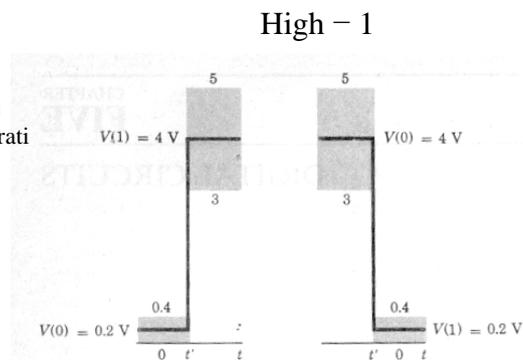


Perchè l'elettronica digitale funziona?



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:



Low = 0

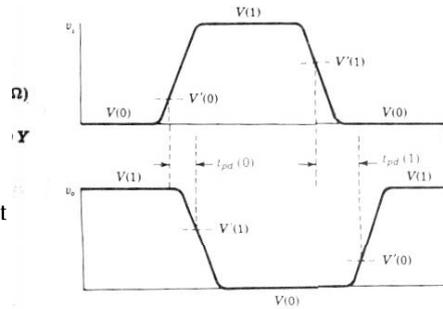
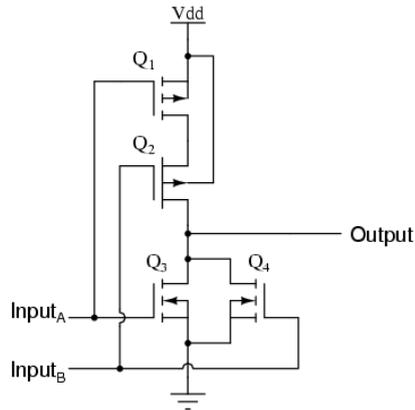


Tempo di commutazione



La commutazione non è istantanea:

CMOS NOR gate



Definizione del cammino critico nei circuiti combinatori.



Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.



Funzione



- Una relazione che associa ad ogni elemento dell'insieme $X(x)$, detto dominio, associa uno ed un solo elemento dell'insieme $Y(y)$, detto codominio, indicandola con $y = f(x)$ (Wikipedia).
Sinonimi: mappa, trasformazione.
- Nulla è detto sulla forma di questa relazione.
 - Funzione analitica (e.g. $Y = \sin(x) \log(\cos(x/2))$)...
 - Funzione a più valori di input e più valori di output (x ed y vettori)
 - Tabella di corrispondenza.
 -



Funzioni logiche



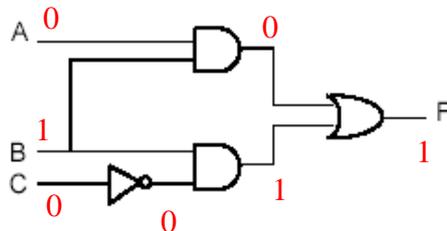
- La funzione calcolata da un circuito con n ingressi.
- Il circuito sarà costituito da un'opportuna combinazione di porte semplici (NOT, AND, OR).
- Per ciascuna delle 2^n combinazioni degli ingressi, può essere calcolata l'uscita.
- Il valore della funzione può essere rappresentato in 3 modi:
 - Circuito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

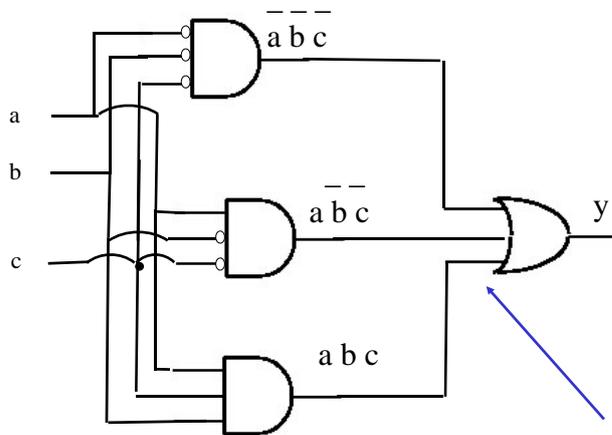


Dal circuito alla funzione logica



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

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



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:

$$A + B \cdot C = A + (B \cdot C) \quad (A+B) \cdot C \text{ per eseguire prima OR}$$

$A B C$ Eseguo prima AND di A e B, nego poi il risultato ed infine eseguo l'AND con C

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

In assenza di parentesi, la negazione ha la priorità sugli altri operatori. Anche sulle negazioni esiste una gerarchia:

$$\bar{\bar{A}} B = (\bar{A}) B \quad \bar{\bar{A}} B C = [(\bar{A}) B] C$$



Regole algebriche



Doppia Inversione	$\bar{\bar{x}} = x$	
	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	$x \bar{x} = 0$	$x + \bar{x} = 1$
Commutativa	$x y = y x$	$x + y = y + x$
Associativa	$(x y) z = x (y z)$	$(x + y) + z = x + (y + z)$
Distributiva	AND rispetto ad OR	OR rispetto ad AND
Assorbimento	$x (y + z) = x y + x z$	$x + y z = (x + y) (x + z)$
De Morgan	$\overline{xy} = \bar{x} + \bar{y}$	$\overline{x + y} = \bar{x} \bar{y}$

Si possono dimostrare sostituendo 0/1 alle variabili.



Teoremi di De Morgan

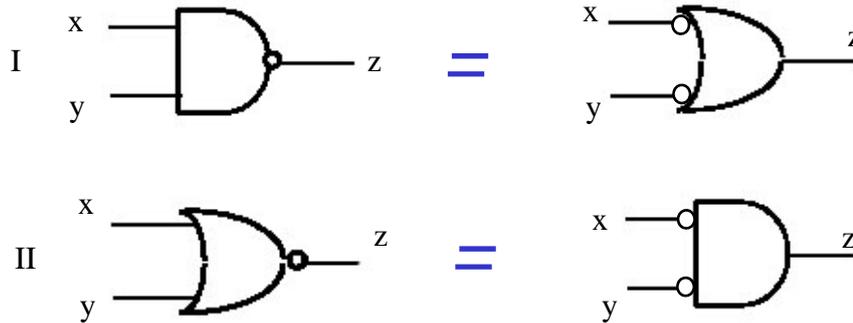


De Morgan

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

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

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



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.

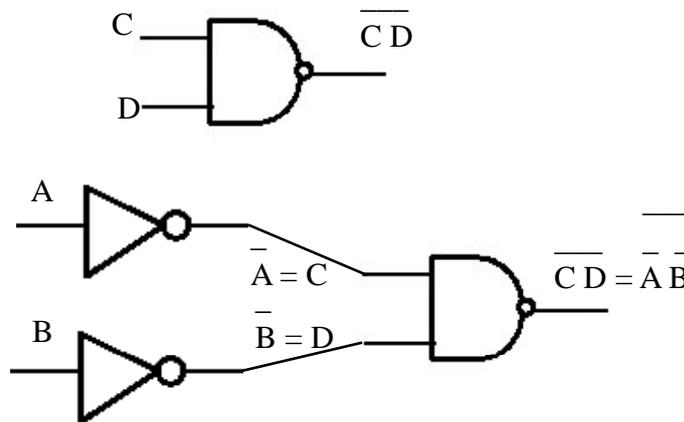
Esempi:

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.



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:

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

E' possibile usarne una sola?

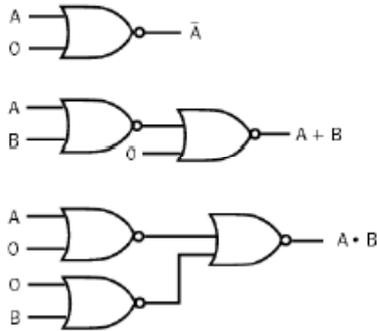
- Sì, ad esempio la porta NAND, o la NOR che sono chiamate **porte universali**.



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)

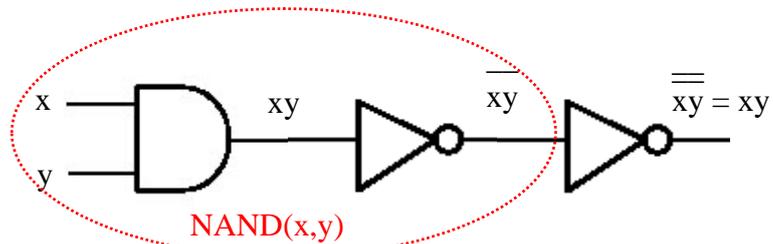


OR mediante porta NAND

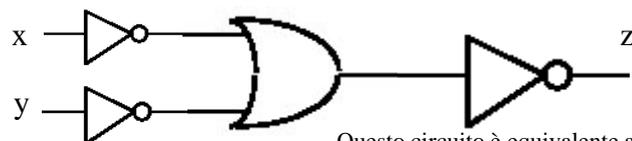


De Morgan: $\overline{xy} = \overline{x} + \overline{y}$

Consideriamo $z = xy$ (porta AND)



= (per De Morgan)



Questo circuito è equivalente a AND(x,y)

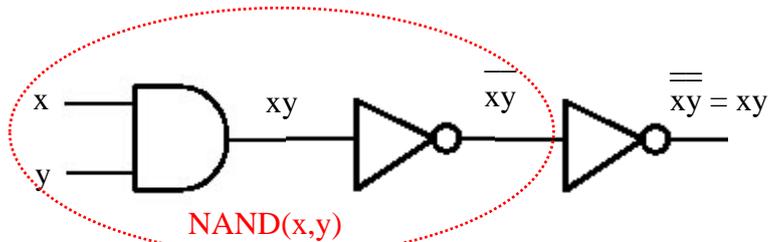


Regole di manipolazioni algebriche

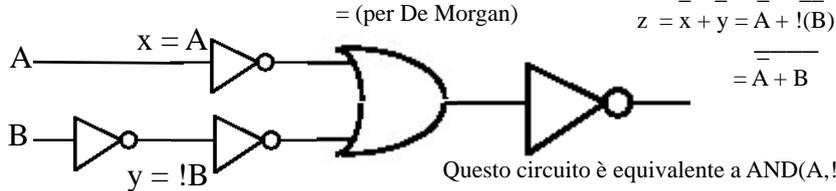


Consideriamo $L = \overline{AB}$

Vogliamo rappresentarlo con porta NOR



$x = A$
 $y = !B$



Semplificazioni notevoli



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

Proprietà distributiva di OR rispetto ad AND:

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

Sviluppando il prodotto:

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

Raccogliendo A:

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

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

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



Esempio di semplificazione algebrica (esercizio)



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

Raccogliendo $\bar{B}C$:

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

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

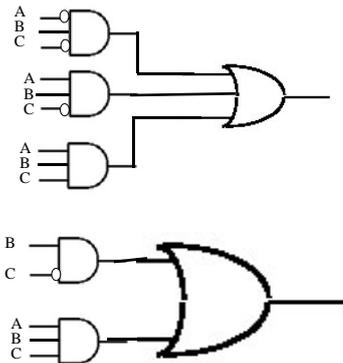
$$= 1\bar{B}C + A\bar{B}\bar{C} =$$

Proprietà dell'identità: " $1B = B$ "

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

Dalla slide precedente:

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



Esempi di manipolazione algebrica



$$F = !xyv + yz + !y!zv + !xy!v + x!yv =$$

$$F = A !B !C + A B C + A B !C + A !B C =$$

$F = A = ?$ Somma di prodotti di 3 variabili: A, B, C (inverso dell'esercizio precedente):



Esercizi



Usare la sola porta NAND per realizzare AND, OR e NOT e disegnarne gli schemi logici

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



Esercizio



Data la funzione booleana:

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

Esprimere la funzione F con il solo connettivo logico NOR e disegnare il circuito.

Esprimere la funzione F con il solo connettivo logico NAND e disegnare il circuito.



Esercizio



Data la funzione booleana:

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

Esprimere la funzione F con il solo connettivo logico NOR e disegnare il circuito.

Esprimere la funzione F con il solo connettivo logico NAND e disegnare il circuito.

Costruire le porte logiche: AND, OR, NOT utilizzando solo la porta NAND.



Sommario



Variabili ed operatori semplici.

Implementazione circuitale (porte logiche).

Dal circuito alla funzione.

Algebra Booleana.