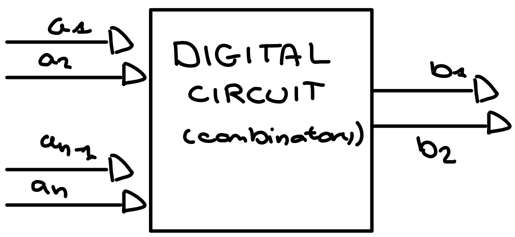
→ Basic Logical Operator

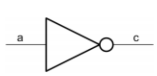
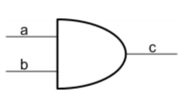
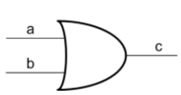
→ I circuiti digitali sono basati su
- **Logica Combinatoria**
  → L'output dipende solo dall'input.
- **Logica Sequenziale**
  → viene usato un clock

LOGICA COMBINATORIA (Algebra di Bode)
→ si basa su logica booleana
→ NOT, AND OR
→ operatori logici

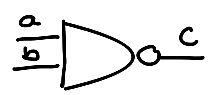
Bit
→ 0 = Low → no corrente
→ 1 = High → c'è corrente



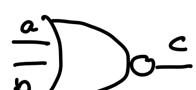
$$b_1 = f(a_1, a_2, \dots, a_n)$$
$$b_2 = g(a_1, a_2, \dots, a_n)$$

Circuital Symbol	Logical Operator	Truth Table	Logical Symbol
	c = **NOT** a	a c 0 1 1 0	c=ā
	c = a **AND** b (·)	a b c 0 0 0 0 1 0 1 0 0 1 1 1	c=a·b
	c = a **OR** b (+)	a b c 0 0 0 0 1 1 1 0 1 1 1 1	c= a+b

\+



c = a NAND b



c = a NOR b

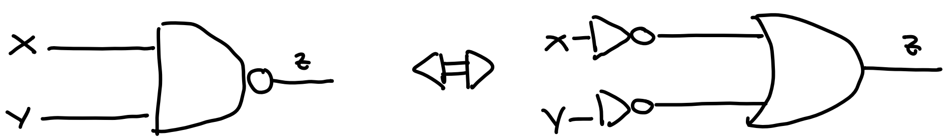
c = a XOR b

→ Leggi di De Morgan

NAND e OR possono essere scombiati

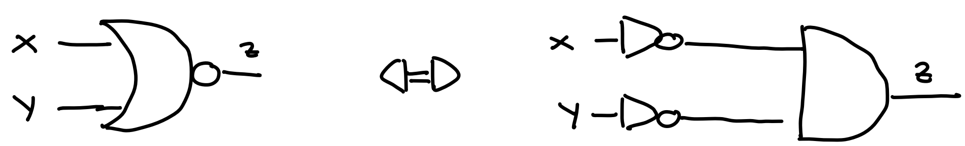
NOT (X AND Y)

$\overline{X \cdot Y} = \overline{X} + \overline{Y}$ X NAND Y = (NOT X) OR (NOT Y)



NOR e AND possono essere scombiati

$\overline{X + Y} = \overline{X} \cdot \overline{Y}$



→ Teorema del Consenso

1) $W = X \cdot Y + \overline{X} \cdot Z + Y \cdot Z = X \cdot Y + \overline{X} \cdot Z$

Quando 2 segnali binari sono in AND con X e $\overline{X}$, l'ultimo termine $Y \cdot Z$ si riduce.

2) $W = (X+Y) \cdot (\overline{X} + Z) \cdot (Y+Z) = (X+Y) \cdot (\overline{X} + Z)$

Quando 2 segnali binari sono in OR con X e $\overline{X}$, l'ultimo temine é $(Y+Z)$ si riduce

1) Se $x = 0$ abbiamo

$W = Z + Y \cdot Z = Z$ in entrambi i casi

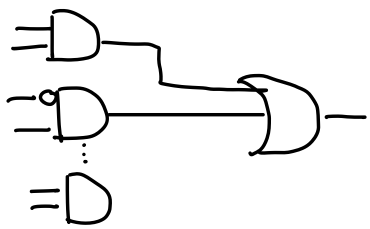
else $Y \cdot Z$ si riduce

$W = Y + YZ = Y$

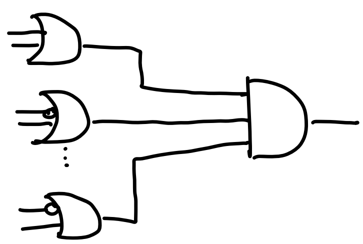
→ Sintesi di somma dei prodotti (SoP) e prodotto delle somme (PoS).

Ogni funzione logica con N input e 1 output può essere espressa in 2 forme canoniche:

- Somma dei prodotti $X \cdot Y \cdot Z + \overline{X} \cdot \overline{Y} \cdot Z + X \cdot \overline{Y} \cdot Z$



- Prodotto delle some $(X+Y+Z) \cdot (\overline{X}+\overline{Y}+Z) \cdot (X+\overline{Y}+\overline{Z})$



Esempio

"Design un circuito logico che compara due numeri A e B binari, $Y = A > B$ $W = A = B$ "

1 Faccio tabella della verità

A_1	A_0	B_1	B_0	Y	W
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	0
⋮		⋮		⋮	⋮
0	1	0	0	1	0

Otteniamo SoP: (Sommo dove c'é 1)

$$Y = \overline{A1} \cdot A0 \cdot \overline{B1} \cdot \overline{B0} + A1 \cdot \overline{A0} \cdot \overline{B1} \cdot \overline{B0} + A1 \cdot \overline{A0} \cdot \overline{B1} \cdot B0$$
$$+ A1 \cdot A0 \cdot \overline{B1} \cdot \overline{B0} + A1 \cdot A0 \cdot \overline{B1} \cdot B0 + A1 \cdot A0 \cdot B1 \cdot \overline{B0}$$

"dalla tabella della veritá tiro fuori i dati"

quando $\overline{A1} = 0 \quad A1 = 1$

PoS (Faccio i prodotti dove c'é 0)

$$Y = (A1 + A0 + B1 + B0) \ldots$$

→ Mappa di Karnaugh

Funzioni Logiche possono essere minimizzate con le proprietá logiche.

esercizi

1) $A \cdot \overline{B} \cdot C + A \cdot B \cdot C$

Distributiva

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

Complemento

$$A \cdot C \cdot (B + \overline{B}) = A \cdot C$$

2) $A + \overline{B} + C + \overline{A} \cdot B \cdot \overline{C}$

De Morgan (1)

$$(A + B + C) + \overline{A} \cdot B \cdot \overline{C} = \overline{\overline{A} \cdot B \cdot \overline{C}} + \overline{A} \cdot B \cdot \overline{C}$$

Complemento

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

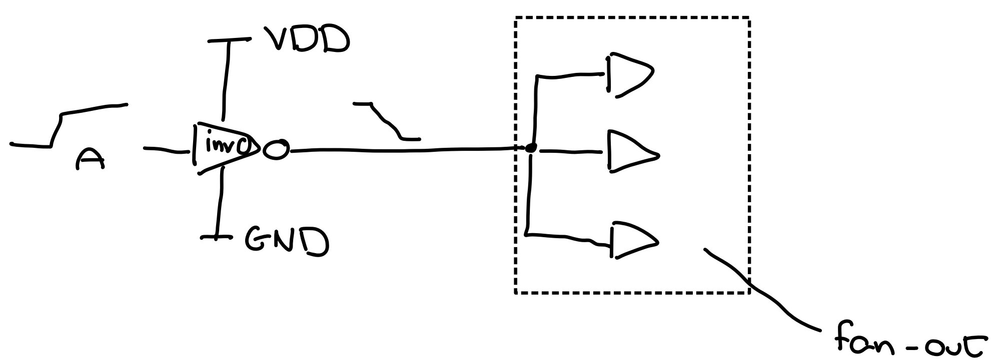
K-MAP (con esempio di prima)

Y		A1A0			
		00	01	11	10
B1B0	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

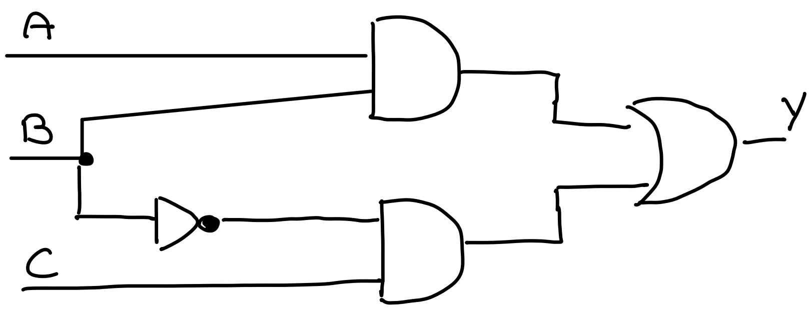
⇒ é una tabella della verità ristretta!

→ Pericolo Statico e Dinamico

- Le porte logiche non hanno un comportamento ideale
- Non commutano istantaneamente quando cambio il segnale di ingresso.
- Il tempo di salita / discesa di uno specifico processo logico dipende da quante porte sono connesse.
- Il numero di LP collegati all'uscita di INVO é chiamato Fan-Out
- Più é grande FO, più lento é il tempo di risposta



Consideriamo:



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

- facciamo la tabella di verità

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

KMAP

C \ AB	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$A \cdot C$ mi ridà sempre 1

→ KMAP con mintermini

Y	A1A0: 00	01	11	10
B1B0: 00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

→ analizzo questi 4 bit.

quando passo da 11→10 in A1A0 l'uscita vale sempre 1

Ho minimizzato →

$$Y = \underbrace{A1 \cdot \overline{B1}}_{\text{perché}} + A0 \cdot \overline{B1} \cdot \overline{B0} + A0 \cdot A1 \cdot \overline{B0}$$

se $A1 = 1$ e $B1 = 0$ accetto qualsiasi $A0$ e $B0$