

Algebra di commutazione

Algebra Booleana - Introduzione

Per descrivere i dispositivi digitali è necessario avere

- Un modello che permetta di rappresentare insiemi di numeri binari;
- Le funzioni che li mettano in relazione.

Algebra Booleana

È una quintupla $\langle B, op1, op2, a, b \rangle$ dove:

- B è l'insieme in cui vengono eseguite le operazioni;
- $op1$ e $op2$ sono operazioni a due elementi che agiscono sugli elementi di B ;
- a e b sono elementi neutri di B per le operazioni $op1$ e $op2$.

Tra le possibili algebre Booleane quella a 2 valori è detta *Algebra di Commutazione*

Algebra di Commutazione

$B: \{0, 1\}$

$op1: \text{AND}$

- ▣ Vale 1 solo se applicata a due valori uguali a 1 altrimenti vale 0

$op2: \text{OR}$

- ▣ Vale 0 solo se applicata a due valori uguali a 0 altrimenti vale 1

$a, b: 1, 0$

Dalla presenza di due soli valori in B è direttamente derivabile la seguente operazione a un valore:

NOT

- ▣ Vale 1 se applicata al valore 0 e 0 se è applicata al valore 1

Descrizione delle Funzioni

Una generica funzione dell'algebra di commutazione può essere descritta in **3 modi**

1. $f(B^n) \rightarrow B^m$
2. $f_i(B^n) \rightarrow B, i = 1, 2, \dots, m$
3. Tabella della verità

Esempi

AND: $B \times B \rightarrow B$

OR: $B \times B \rightarrow B$

NOT: $B \rightarrow B$

Tabelle della Verità

AND		
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

OR		
x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
x	z
0	1
1	0

Simbologia

Utilizzando le tre operazioni elementari si possono scrivere delle espressioni tra variabili di commutazione che descrivono in forma compatta il comportamento delle funzioni

Simbologia

$$z = \text{AND}(x, y) \quad \rightarrow \quad z = x \cdot y \quad \rightarrow \quad z = xy$$

$$z = \text{OR}(x, y) \quad \rightarrow \quad z = x + y$$

$$z = \text{NOT}(x) \quad \rightarrow \quad z = \underline{x}$$

Identità e Teoremi (1 di 2)

Identità

$$1 \cdot x = x$$

$$0 + x = x$$

Elemento nullo

$$0 \cdot x = 0$$

$$1 + x = 1$$

Idempotenza

$$x \cdot x = x$$

$$x + x = x$$

Inverso

$$x \cdot \underline{x} = 0$$

$$x + \underline{x} = 1$$

Commutativa

$$x \cdot y = y \cdot x$$

$$x + y = y + x$$

Associativa

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

$$(x + y) + z = x + (y + z)$$

Identità e Teoremi (2 di 2)

Assorbimento

$$x \cdot (x + y) = x$$

$$x + x \cdot y = x$$

Distributiva

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + y \cdot z = (x + y) \cdot (x + z)$$

De Morgan

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

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

Teorema di Shannon

Data una funzione Booleana $f(B^n)=B$ è sempre vero che:

$$f(x_1, x_2, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) + \underline{x}_1 f(0, x_2, \dots, x_n)$$

$$f(x_1, x_2, \dots, x_n) = (x_1 + f(0, x_2, \dots, x_n)) \cdot (\underline{x}_1 + f(1, x_2, \dots, x_n))$$

Principio di Dualità

Un'equazione booleana rimane valida se si considera il duale di entrambi i lati dell'uguaglianza

Il **duale** di un'espressione si ottiene

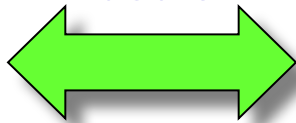
- Cambiando gli **AND** con gli **OR** e viceversa
- Cambiando gli **1** con **0** e viceversa

1. $x + xy = x(1 + y) = x$

2. $xy + x\bar{y} = x(y + \bar{y}) = x$

3. $x + \bar{x}y = (x + \bar{x})(x + y) = x + y$

duale



1d. $x(x + y) = x + (0 \cdot y) = x$

2d. $(x + y)(x + \bar{y}) = x + (y\bar{y}) = x$

3d. $x(\bar{x} + y) = x\bar{x} + xy = xy$

Teorema del Consenso

$$xy + \underline{x}z + yz = xy + \underline{x}z$$

Dimostrazione

$$\begin{aligned} xy + \underline{x}z + yz &= xy + \underline{x}z + yz(\underline{x} + x) = xy + \underline{x}z + xyz + \underline{x}yz = \\ &= xy + xyz + \underline{x}z + \underline{x}yz = xy(1 + z) + \underline{x}z(1 + y) = \\ &= xy + \underline{x}z \end{aligned}$$

□

Per il *principio di dualità* si ha

$$(x + y)(\underline{x} + z)(y + z) = (x + y)(\underline{x} + z)$$

Esercizio

Semplificare l'espressione booleana $(a + b)(\underline{a} + c)$

$$(a + b)(\underline{a} + c) = a\underline{a} + ac + \underline{a}b + bc =$$

$$= \boxed{ac + \underline{a}b} + \boxed{bc} =$$

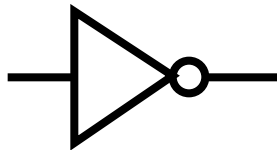
$$= ac + \underline{a}b$$

Dalla Funzione al Circuito

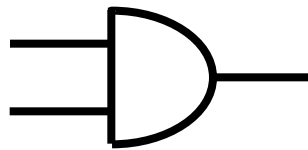
Per costruire un circuito digitale che si comporti come una espressione è necessario trovare degli equivalenti elettrici per le variabili e per gli operatori

Variabile → Conduttore

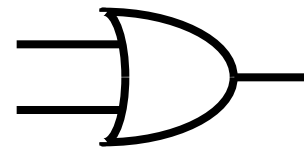
Operatori → Porte logiche



NOT

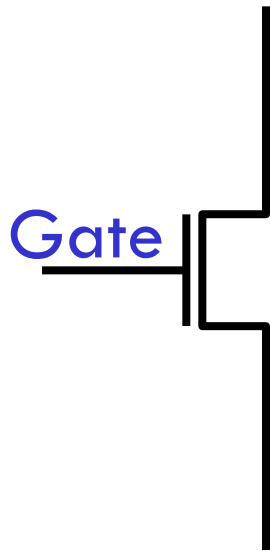


AND



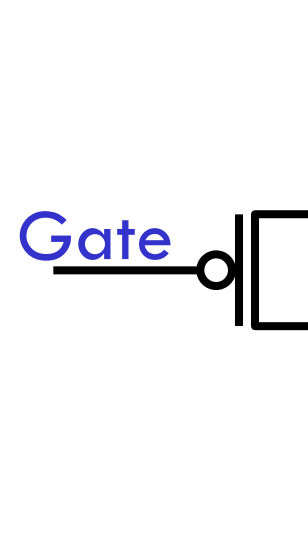
OR

Transistori di tipo N e P



Tipo N

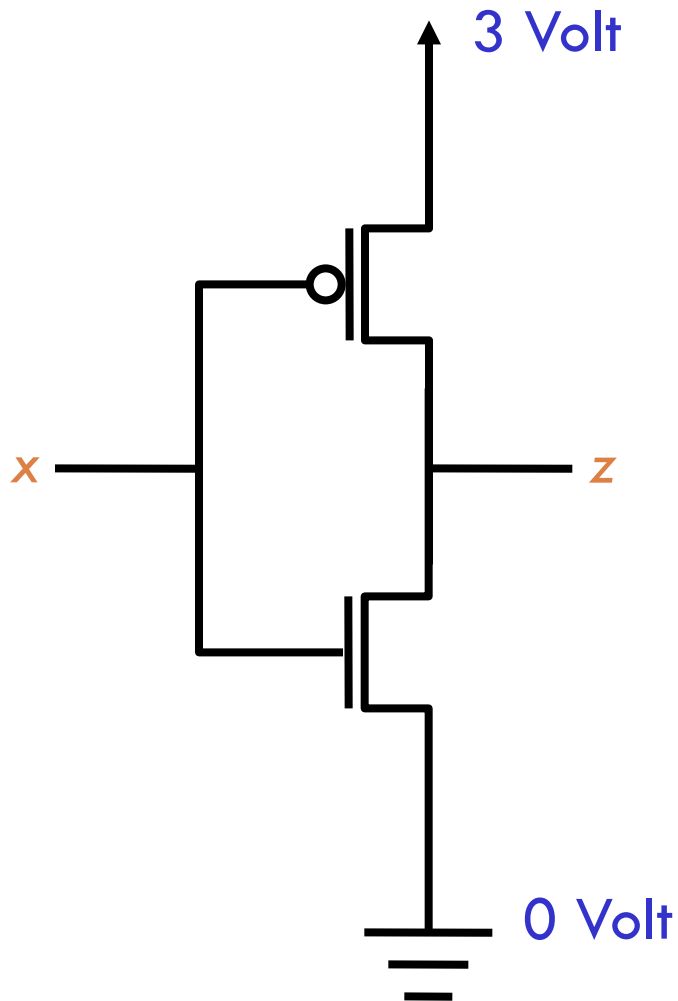
Gate = 1, conduzione



Tipo P

Gate = 0, conduzione

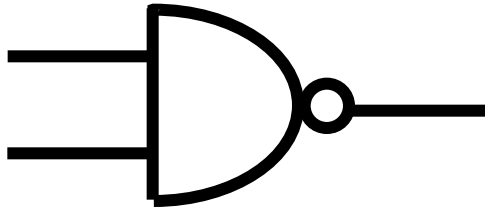
Porta logica NOT



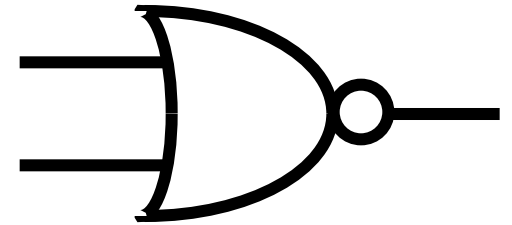
NOT	
<i>x</i>	<i>z</i>
0	1
1	0

Altre Porte Logiche

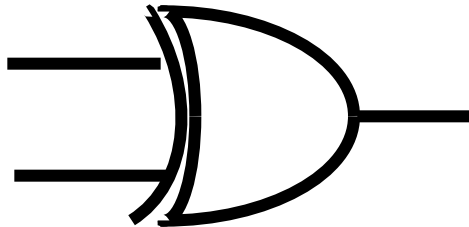
NAND		
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0



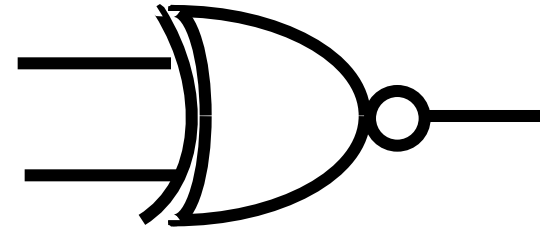
NOR		
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0



XOR		
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0



XNOR		
x	y	z
0	0	1
0	1	0
1	0	0
1	1	1



Operatori Universali (1 di 2)

Gli **operatori universali** (o funzionalmente completi) consentono di rappresentare una qualsiasi funzione di commutazione.

AND, NOT sono operatori universali

$$X \text{ or } Y = (X \text{ or } Y)'' = (X' \text{ and } Y')'$$

OR e NOT sono operatori universali

$$X \text{ and } Y = (X \text{ and } Y)'' = (X' \text{ or } Y')'$$

Operatori Universali (2 di 2)

NAND è un operatore universale

$$X' = X \text{ nand } X$$

$$X \text{ and } Y = (X \text{ nand } Y)'$$

$$X \text{ or } Y = (X \text{ or } Y)'' = (X' \text{ and } Y')' = X' \text{ nand } Y'$$

NOR è un operatore universale

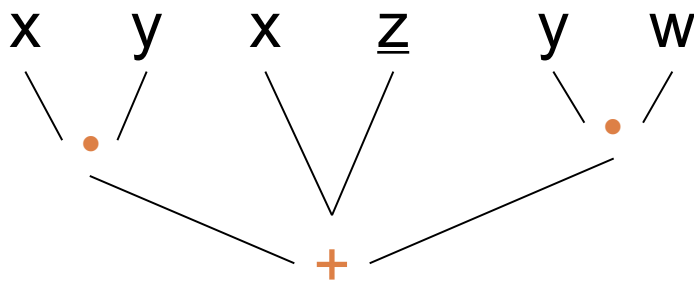
$$X' = X \text{ nor } X$$

$$X \text{ or } Y = (X \text{ nor } Y)'$$

$$X \text{ and } Y = (X \text{ and } Y)'' = (X' \text{ or } Y')' = X' \text{ nor } Y'$$

Espressioni Booleane

- Si definisce espressione booleana una combinazione di variabili booleane e costanti (0,1) attuata mediante gli operatori AND, OR e NOT
 - Si definisce *numero di livelli* di una espressione il massimo tra i numeri di operazioni che agiscono in cascata sui *letterali*
- $x \cdot y + x + \underline{z} + y \cdot w$



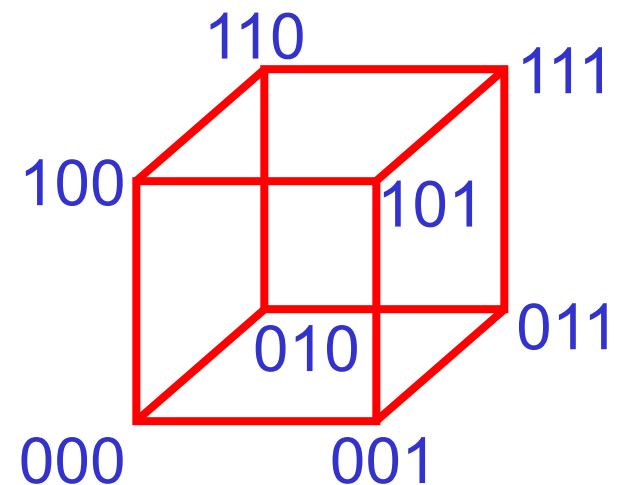
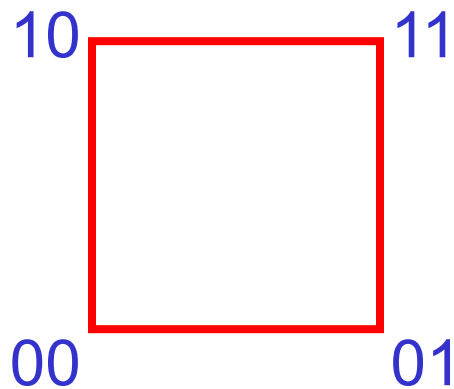
Primo livello

Secondo livello

Forme Canoniche

Sia $f(B^n)=B$ una funzione Booleana ad una sola uscita
completamente specificata

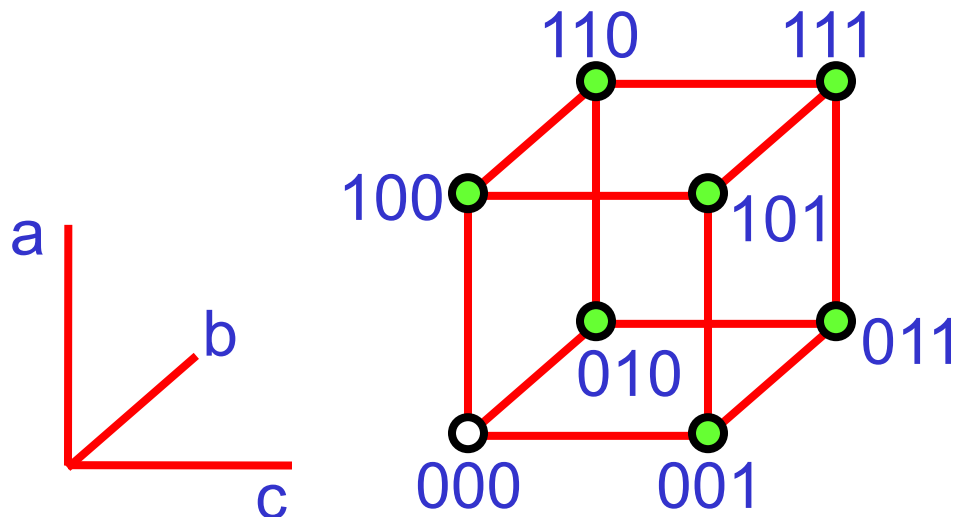
Le 2^n configurazioni degli ingressi possono essere mappate sui vertici di un n -cubo in modo tale che due punti adiacenti siano a **distanza di Hamming** pari a 1



Forme Canoniche

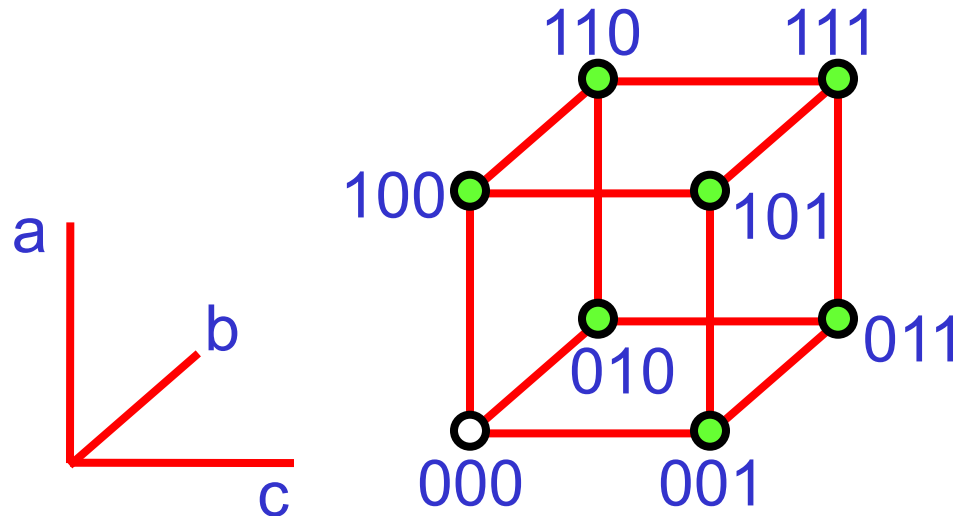
Si consideri per esempio $n=3$ (variabili a, b, c)

Gli spigoli del cubo saranno indicati con un pallino pieno se in corrispondenza di quel valore di ingresso la funzione vale 1 e con un pallino vuoto altrimenti



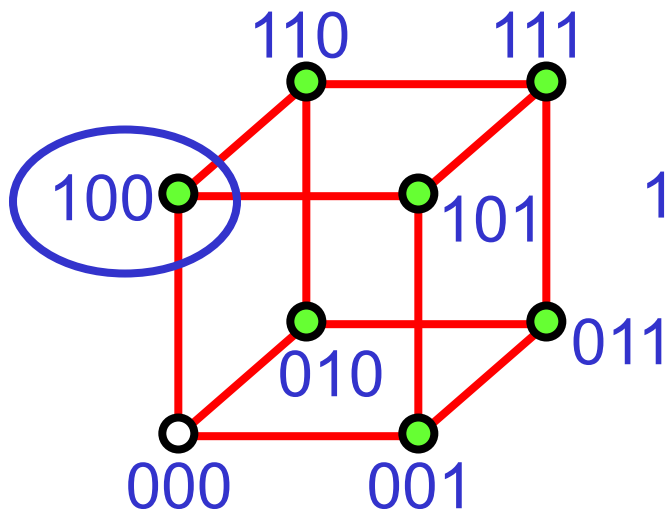
Forme Canoniche (Definizioni)

- **Letterale:** coppia variabile valore
 - ▣ $(a,0), (a,1), (b,0), (b,1), (c,0), (c,1)$
 - ▣ Per brevità i suddetti saranno indicati rispettivamente come: a, a, b, b, c, c
- **Implicante:** prodotto di letterali tale che se tale prodotto vale 1 anche f vale 1
 - ▣ Esempio: $a, b, c, ab, abc, a\underline{b}c, \dots$ sono implicanti per questa funzione

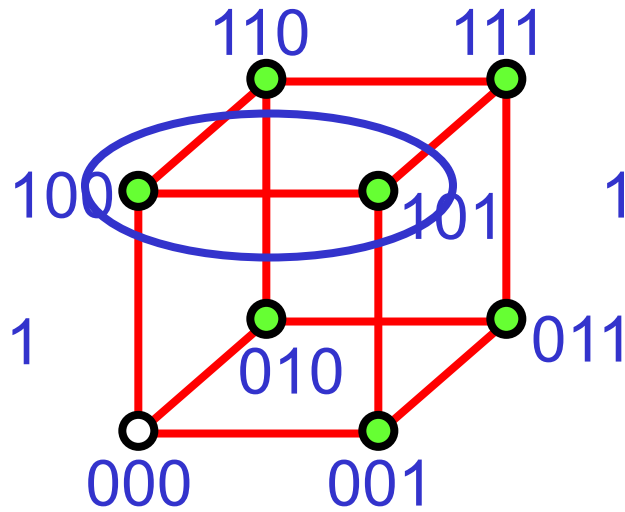


Forme Canoniche (Definizioni)

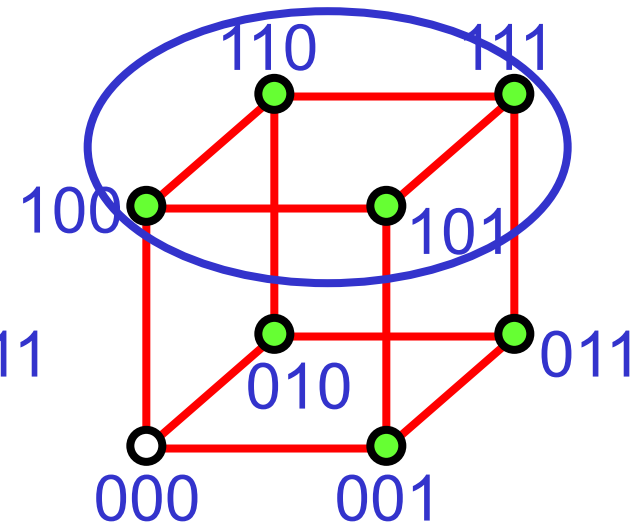
- **Implicante**: può essere visto come un sottocubo di soli 1 della funzione data. Cioè come un insieme di 2^k configurazioni di ingresso a *distanza di Hamming* unitaria



Implicante abc
(dimensione 1)



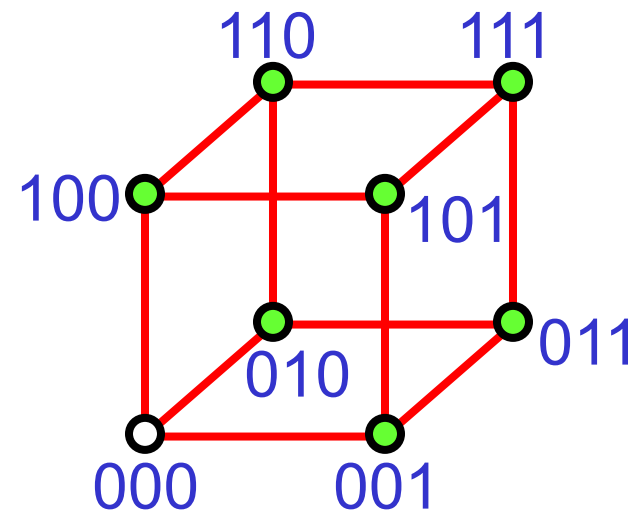
Implicante ab
(dimensione 2)



Implicante a
(dimensione 4)

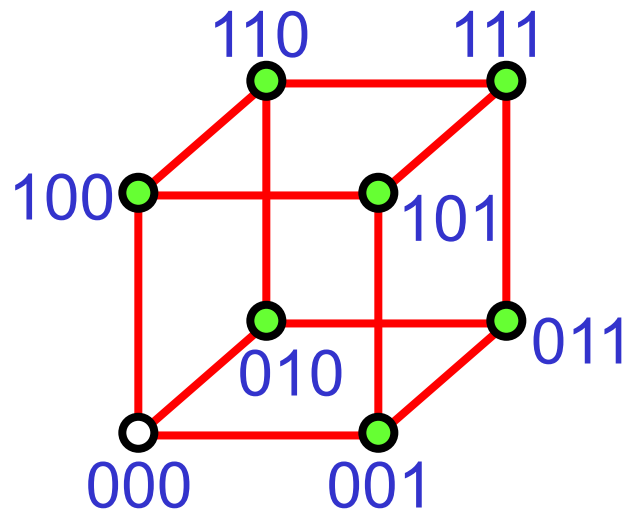
Forme Canoniche (Definizioni)

- **Mintermine**: un implicante in cui compaiono tutte le variabili di ingresso
 - ▣ $\underline{a}bc, a\underline{b}c, abc, \underline{a}\underline{b}\underline{c}, \underline{a}bc, a\underline{b}\underline{c}, \underline{a}bc$
 - ▣ **NOTA**: I mintermini di una funzione vengono spesso identificati con il numero in base 10 corrispondente al valore binario del mintermine.
(Es., $\underline{a}bc = m_4$)
- **Maxtermine**: è un punto dello spazio B^n tale per cui la funzione calcolata in quel punto vale 0
- **On-set**: Insieme dei mintermini
- **Off-set**: Insieme dei maxtermini



Forme Canoniche (Definizioni)

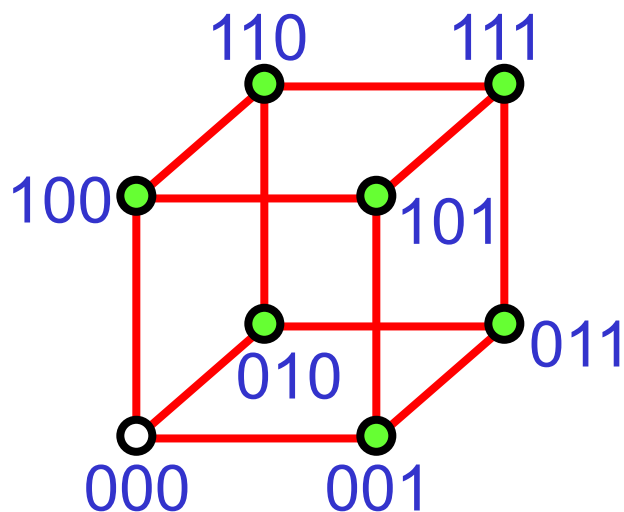
- **Implicante primo**: implicante tale che non esiste nessun altro implicante di dimensioni maggiori che lo contenga interamente



Es.: L'implicante abc non è primo perché è contenuto nell'implicante ab che è a sua volta contenuto nell'implicante a che è primo perché non è contenuto in nessun altro implicante

Forme Canoniche (Definizioni)

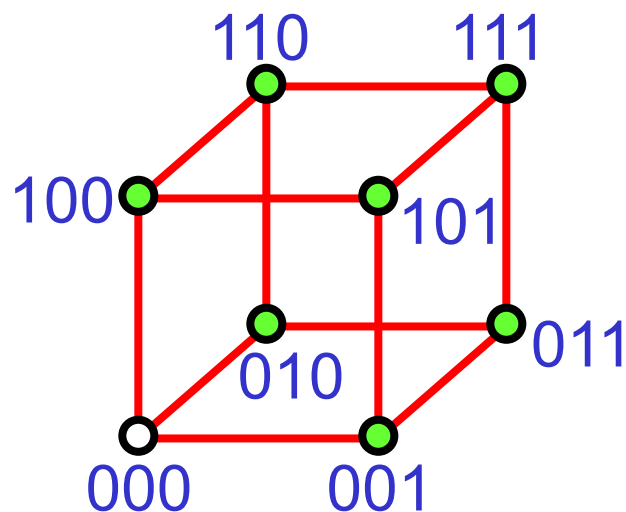
- **Implicante essenziale**: implicante primo che copre almeno un mintermine non coperto da altri implicanti primi



Es.: L'implicante primo c è **essenziale** perché copre il mintermine abc che non è coperto da nessun altro implicante primo.

Forme Canoniche (Definizioni)

- **Copertura di una funzione:** è un insieme di implicanti che coprono tutti i mintermini della funzione.



Es.: $\{a, b, c\}$ è una copertura. $\{a, b, \underline{a}bc\}$ è una copertura.

Prima Forma Canonica

- Sia $f(x_1, \dots, x_n)$ una funzione Booleana
- Sia $\{m_1, m_2, \dots, m_k\}$ l'*On-set* della funzione
- La **prima forma canonica di copertura** della funzione (detta **somma di prodotti**) è:

$$f(x_1, \dots, x_n) = m_1 + m_2 + \dots + m_k$$

Seconda Forma Canonica

- Sia $f(x_1, \dots, x_n)$ una funzione Booleana
- Sia $\{M_1, M_2, \dots, M_k\}$ l'*Off-set* della funzione
- La **seconda forma canonica di copertura** della funzione (detta **prodotto di somme**) è:

$$f(x_1, \dots, x_n) = M_1 \cdot M_2 \cdot \dots \cdot M_k$$

Esempio

o = f(a,b,c)			
a	b	c	o
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$m_0 = \underline{a}\underline{b}\underline{c}$$

$$M_1 = \underline{a} + \underline{b} + \underline{c}$$

$$M_2 = \underline{a} + \underline{b} + c$$

$$m_3 = \underline{a}\underline{b}c$$

$$m_4 = \underline{a}\underline{b}c$$

$$M_5 = \underline{a} + \underline{b} + \underline{c}$$

$$M_6 = \underline{a} + \underline{b} + c$$

$$m_7 = abc$$

quando $a=0, b=0, c=0 \rightarrow m_0=1$

quando $a=0, b=0, c=1 \rightarrow M_1=0$

quando $a=0, b=1, c=0 \rightarrow M_2=0$

quando $a=0, b=1, c=1 \rightarrow m_3=1$

quando $a=1, b=0, c=0 \rightarrow m_4=1$

quando $a=1, b=0, c=1 \rightarrow M_5=0$

quando $a=1, b=1, c=0 \rightarrow M_6=0$

quando $a=1, b=1, c=1 \rightarrow m_7=1$

$$\text{On-set} = \{m_0, m_3, m_4, m_7\}$$

$$\text{PFC: } m_0 + m_3 + m_4 + m_7$$

$$\text{PFC: } \underline{a}\underline{b}\underline{c} + \underline{a}\underline{b}c + \underline{a}\underline{b}c + abc$$

$$\text{Off-set} = \{M_1, M_2, M_5, M_6\}$$

$$\text{SFC: } M_1 M_2 M_5 M_6$$

$$\text{SFC: } (a+b+\underline{c}) (a+\underline{b}+c) (\underline{a}+b+\underline{c}) (\underline{a}+\underline{b}+c)$$

Forme Negate

Un mintermine ed un maxtermine identificati dallo stesso pedice sono l'uno il complemento dell'altro: $M_i = \underline{m}_i$

Es: $\underline{m}_3 = \text{NOT}(\underline{x}yz) = x + \underline{y} + \underline{z} = M_3$

È possibile ricavare semplicemente le forme negate delle funzioni espresse in prima e seconda forma canonica

Es:

$$f = \sum_i m_i \rightarrow \underline{f} = \text{NOT}(\sum_i m_i) \rightarrow \text{De Morgan} \rightarrow \underline{f} = \prod_i \underline{m}_i \rightarrow \underline{f} = \prod_i M_i$$

Es:

$$f = \prod_i M_i \rightarrow \underline{f} = \text{NOT}(\prod_i M_i) \rightarrow \text{De Morgan} \rightarrow \underline{f} = \sum_i \underline{M}_i \rightarrow \underline{f} = \sum_i m_i$$

Forme Negate

Esempio

$$\textbf{PFC: } f = m_0 + m_3 + m_4 + m_7$$

$$\underline{f} = M_0 M_3 M_4 M_7$$

$$\underline{f} = (a+b+c)(a+\underline{b}+\underline{c})(\underline{a}+b+c)(\underline{a}+\underline{b}+\underline{c})$$

$$\textbf{SFC: } f = M_1 M_2 M_5 M_6$$

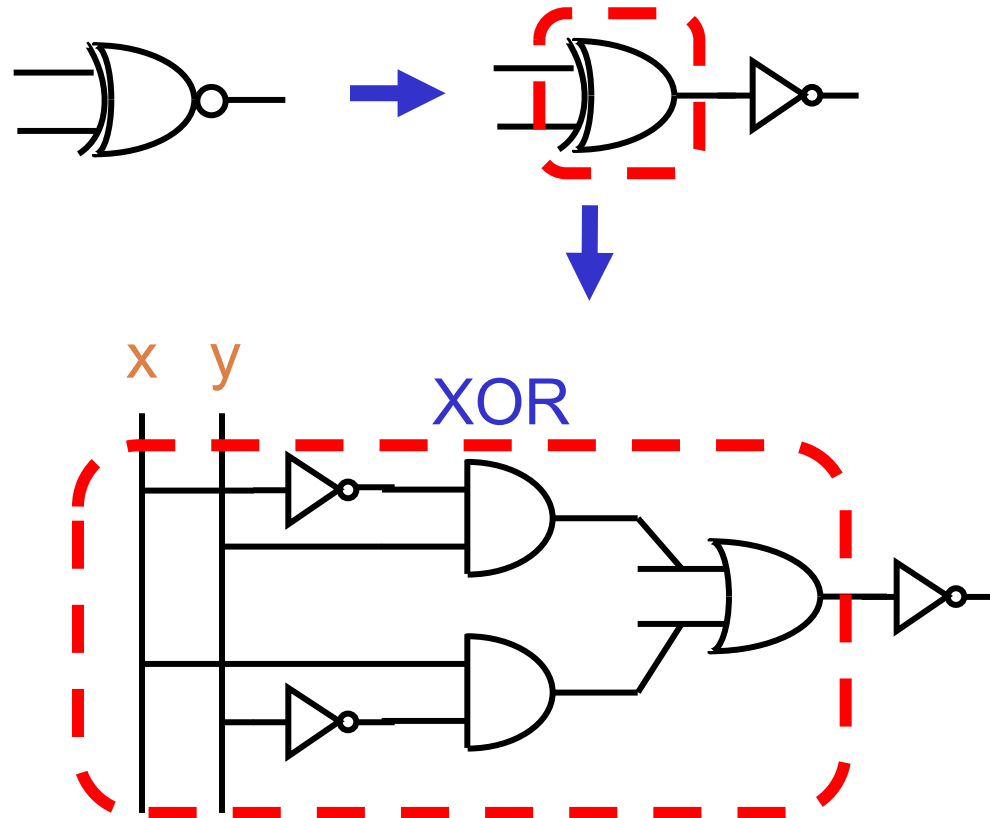
$$\underline{f} = m_1 + m_2 + m_5 + m_6$$

$$\underline{f} = \underline{a}bc + \underline{a}b\underline{c} + a\underline{b}c + ab\underline{c}$$

Esercizio 1

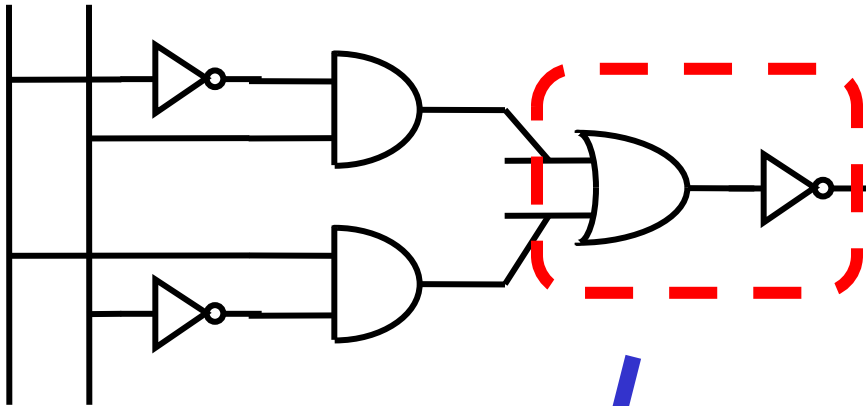
Si descriva in termini di porte logiche **NAND**, a uno e due ingressi, la funzionalità della porta logica **XNOR**.

XNOR		
x	y	z
0	0	1
0	1	0
1	0	0
1	1	1



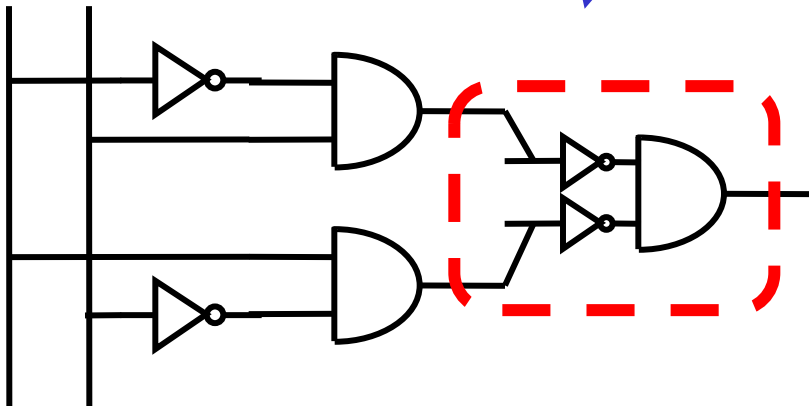
Esercizio 1 (cont.)

x y

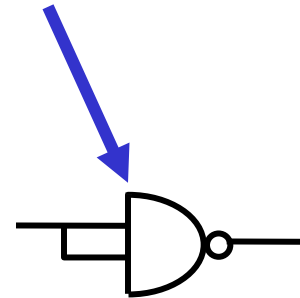
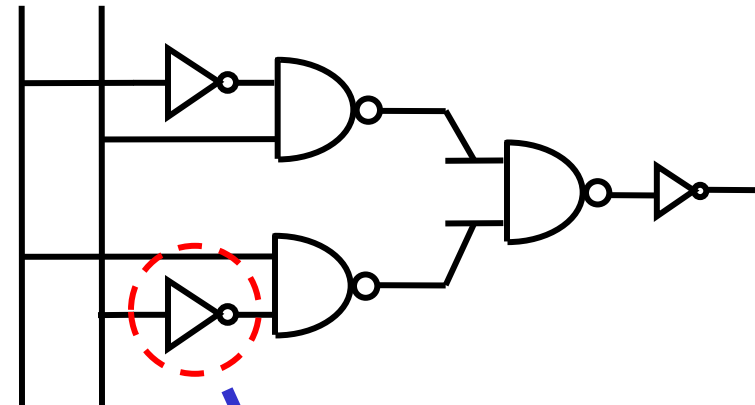


De Morgan

x y



x y



Esercizio 2

Si scriva la tabella della verità della funzione Booleana $o=f(a,b,c)$, tale che o vale 1 solo se la somma (senza riporto) dei bit sulle linee a e b è uguale al valore della linea c e si ricavi l'*On-set*.

$o = f(a,b,c)$			
a	b	c	o
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

On-set

$m_0 = \underline{a}bc$

$m_3 = \underline{a}bc$

$m_5 = a\underline{b}c$

$m_6 = ab\underline{c}$

Esercizio 3

Si consideri un circuito combinatorio che ha due ingressi a 2 bit: $A(a_1, a_0)$ e $B(b_1, b_0)$, un ingresso op a 1 bit e un'uscita out a 1 bit. Se $op=0$, out vale 1 se $A > B$. Se $op=1$, out vale 1 se $A = B$.

Definire l'*On-set* della funzione $out(op, a_1, a_0, b_1, b_0)$.

Esercizio 3 (cont.)

Soluzione

Non è agevole procedere come prima costruendo la tabella della verità che in questo caso conterrebbe un numero di righe pari a $2^5=32$.

Generiamo quindi direttamente i mintermini

out vale 1 se $op=0$ (cioè $\underline{op}$) **E** se $A>B$ [cioè se $a_1=b_1$ (a_1b_1 **O** $\underline{a_1b_1}$) **E** $a_0=1$ (a_0) **E** $b_0=0$ ($\underline{b_0}$) **OPPURE** se $a_1=1$ (a_1) **E** $b_1=0$ ($\underline{b_1}$)]

Quindi la precedente diventa (gli **E** diventano AND ($\bullet$) e gli **O** diventano OR ($+$)): $\underline{op}((\underline{a_1b_1} + \underline{a_1b_1}) \bullet a_0 \bullet \underline{b_0} + a_1 \bullet \underline{b_1})$

Allo stesso modo si ha per la seconda: $op((\underline{a_0b_0} + a_0b_0)(\underline{a_1b_1} + a_1b_1))$

$\underline{op}a_1a_0b_1\underline{b_0}=m_{14}$, $\underline{op}\underline{a_1}a_0\underline{b_1}\underline{b_0}=m_4$, etc.

Esercizio 4

Si scriva l'espressione secondo la prima forma canonica della funzione $f(a,b,c)$ che ha On-set = $\{m_2, m_4, m_5, m_7\}$

Esercizio 4 (cont.)

Soluzione

$$m_2 = m_{010} = \underline{a}b\underline{c}$$

$$m_4 = m_{100} = a\underline{b}\underline{c}$$

$$m_5 = m_{101} = a\underline{b}c$$

$$m_7 = m_{111} = abc$$

$$f(a,b,c) = \underline{a}b\underline{c} + a\underline{b}\underline{c} + a\underline{b}c + abc$$

Esercizio 5

Si scriva l'espressione secondo la seconda forma canonica della funzione $f(a,b,c)$ che ha

$$\text{On-set} = \{m_2, m_4, m_5, m_7\}$$

Esercizio 5 (cont.)

Soluzione

$$f(a,b,c) = M_0 M_1 M_3 M_6$$

$$M_0 = M_{000} = a+b+c$$

$$M_1 = M_{001} = a+b+\underline{c}$$

$$M_3 = M_{011} = a+\underline{b}+\underline{c}$$

$$M_6 = M_{110} = \underline{a}+\underline{b}+c$$

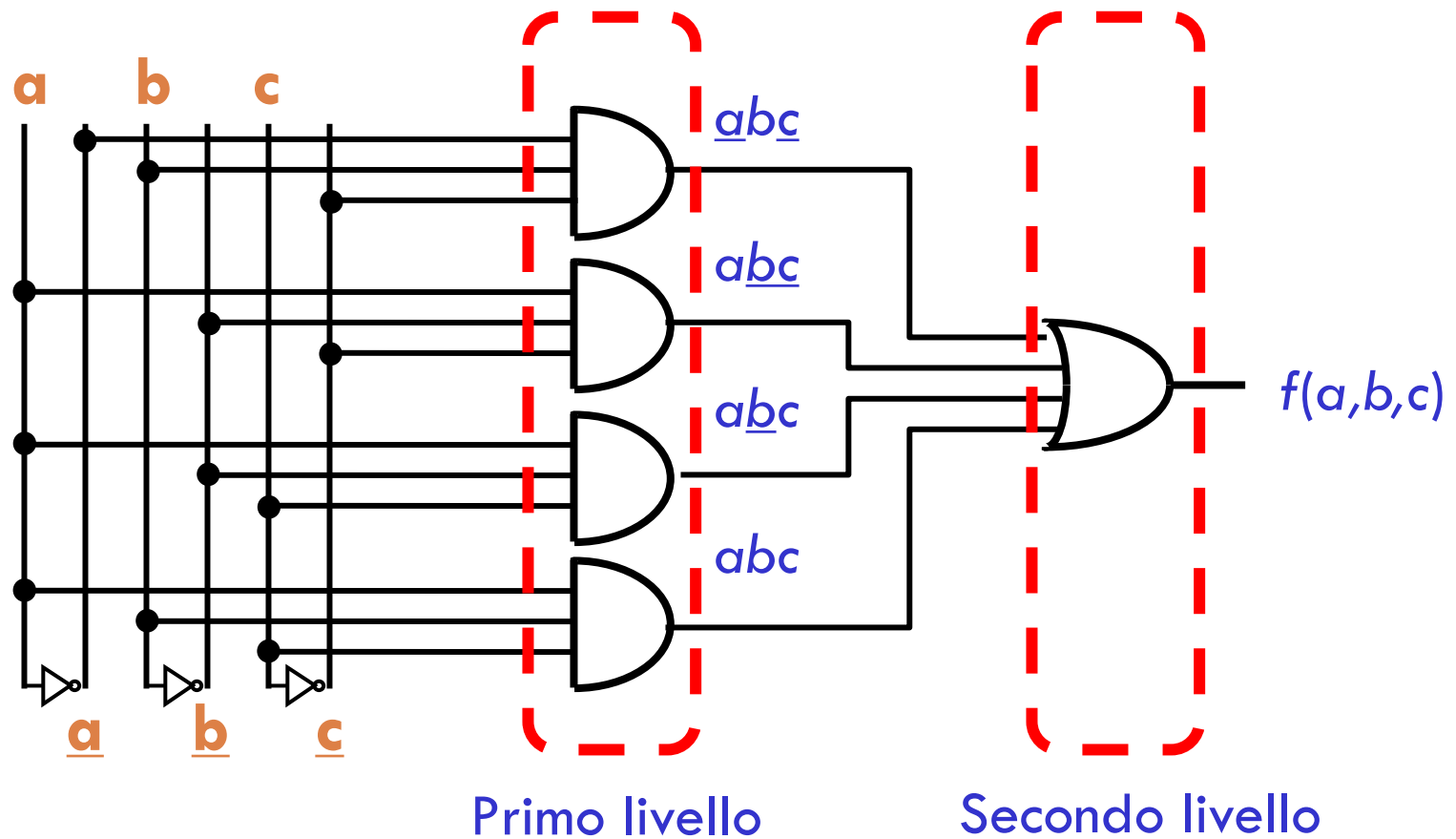
$$f(a,b,c) = (a+b+c) \cdot (a+b+\underline{c}) \cdot (a+\underline{b}+\underline{c}) \cdot (\underline{a}+\underline{b}+c)$$

Esercizio 6

Si disegni il circuito a due livelli corrispondente all'espressione in somma di prodotti della funzione dell'Esercizio 4

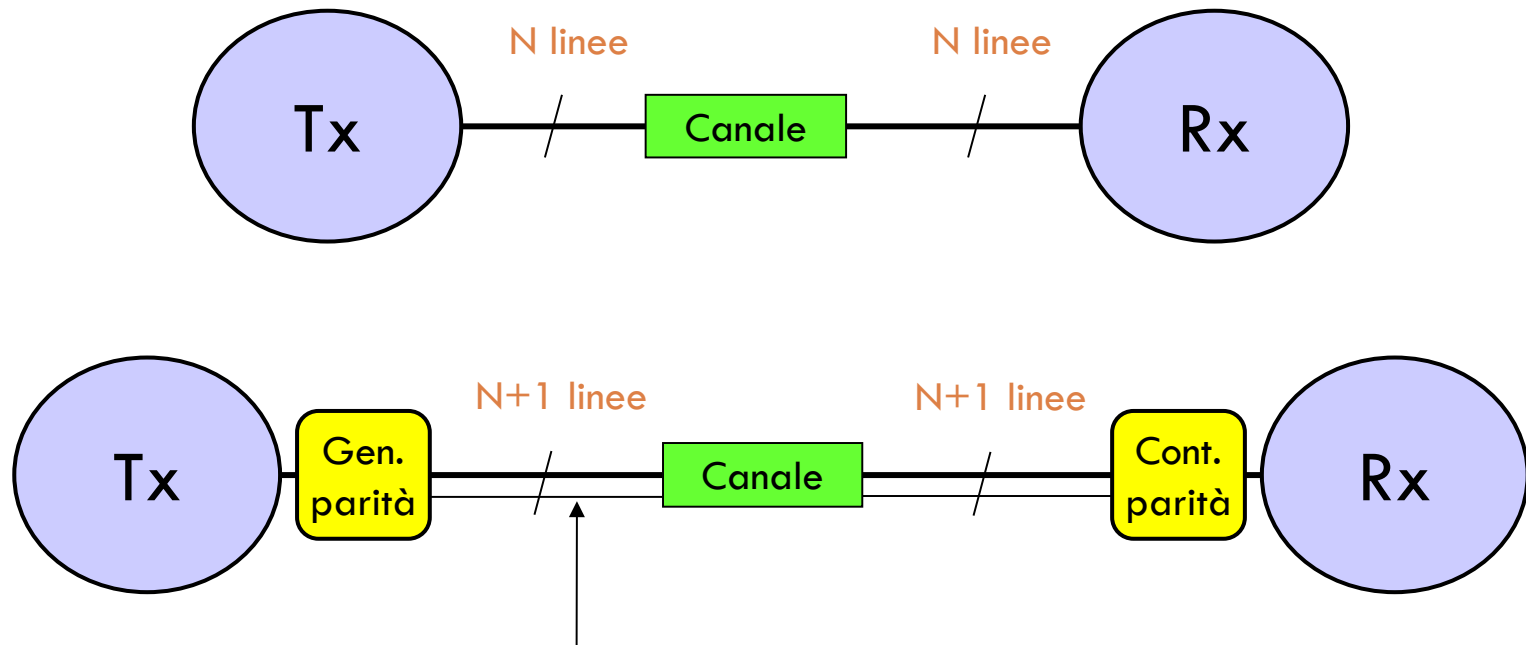
$$f(a,b,c) = \underline{a}\underline{b}\underline{c} + a\underline{b}\underline{c} + a\underline{b}c + abc$$

Esercizio 6 (cont.)



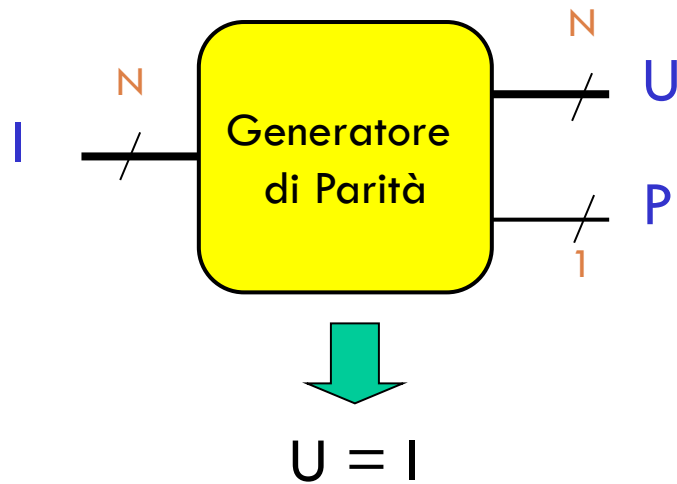
Esercizio 7

Progettare un generatore ed un controllore di parità

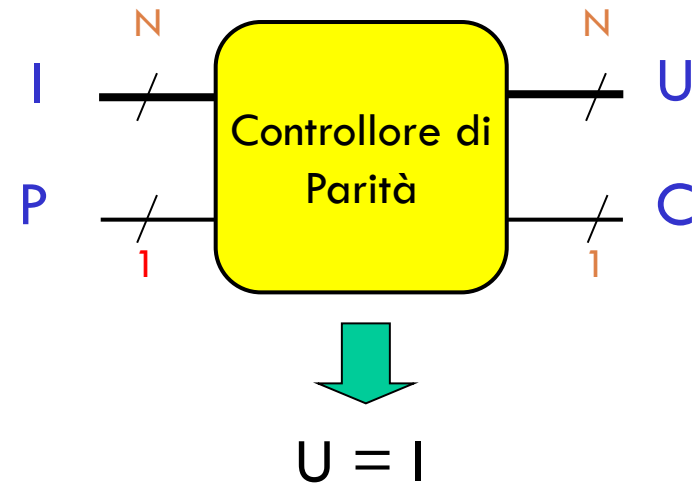


L'informazione è formata
da un numero di 1 pari

Esercizio 7 (cont.)



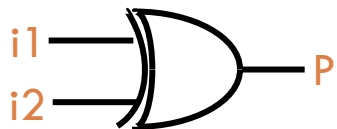
$$P = \begin{cases} 1 & \text{se il numero di } 1 \\ & \text{in } I \text{ è dispari} \\ 0 & \text{altrimenti} \end{cases}$$



$$C = \begin{cases} 1 & \text{se il numero di } 1 \\ & \text{in } I \cup P \text{ è dispari} \\ 0 & \text{altrimenti} \end{cases}$$

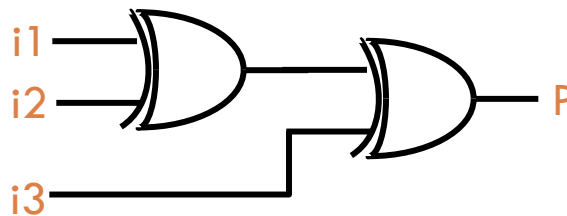
Esercizio 7 (cont.)

XOR		
i1	i2	P
0	0	0
0	1	1
1	0	1
1	1	0



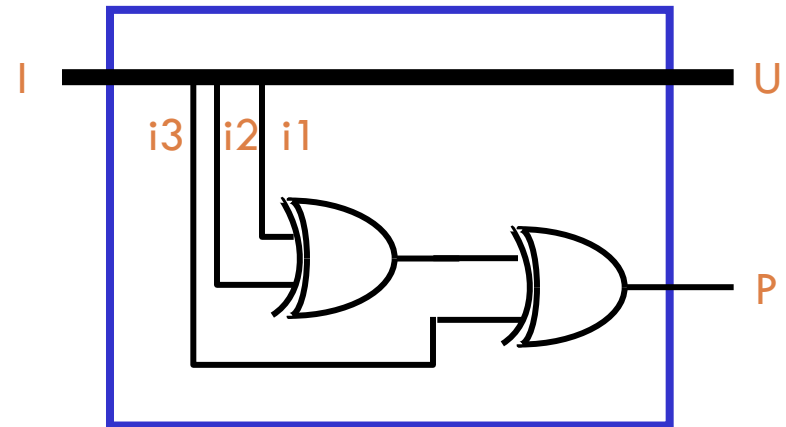
$$P = i1 \oplus i2$$

XOR			
i1	i2	i3	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



$$P = i1 \oplus i2 \oplus i3$$

Generatore di Parità



Controllore di Parità

