Algebra di commutazione

Algebra booleana: introduzione

- Per descrivere i dispositivi digitali è necessario avere:
  - Un modello che permette di rappresentare insiemi di numeri binari
  - Le funzioni che li mettono in relazione

Algebra booleana

- **Operazione**:

  una operazione ***op*** sull'insieme S={s1,s2,...} è una funzione

  op : SxS→S

  che da SxS (S cartesiano S) porta in S.

Algebra booleana

- E' una quintupla *<B, op1, op2, a, b>*
  - *B*: Insieme in cui vengono eseguite le operazioni
  - *op1*, *op2*: Operazioni a due elementi che agiscono sugli elementi di *B*
  - *a*, *b*: 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*: **AND**
  - Vale 1 solamente se applicata a due valori uguali a 1 altrimenti vale 0
- *op2*: **OR**
  - Vale 0 solamente 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:
  - $f(B^n) \rightarrow B^m$
  - $f_i(B^n) \rightarrow B$, $i = 1,2,\ldots,m$
  - Tabella della verità
- Esempi
  - **AND**: $B \times B \rightarrow B$
  - **OR**: $B \times B \rightarrow B$
  - **NOT**: $B \rightarrow B$

Algebra di commutazione

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
  - $z = \mathrm{AND}(x, y) \rightarrow z = x \bullet y \rightarrow z = xy$
  - $z = \mathrm{OR}(x, y) \rightarrow z = x + y$
  - $z = \mathrm{NOT}(x) \rightarrow z = x'$

Proprietà dell'algebra di commutazione

1) **Elemento neutro**

$x+1=1$ $\qquad$ $x \bullet 0=0$

$x+0=x$ $\qquad$ $x \bullet 1=x$

2) **Idempotenza**

$x+x=x$ $\qquad$ $x \bullet x=x$

3) **Complementazione**

$x+x'=1$ $\qquad$ $x \bullet x'=0$

Proprietà dell'algebra di commutazione

4) **Commutatività**

$x+y=y+x$ $\qquad$ $x \bullet y=y \bullet x$

5) **Associatività**

$(x+y)+z=x+(y+z)$ $\qquad$ $(x \bullet y) \bullet z=x \bullet(y \bullet z)$

6) **Assorbimento**

$x+xy=x$ $\qquad$ $x \bullet(x+y)=x$

Proprietà dell'algebra di commutazione

7) **Distribuitività**

$x+y \bullet z= (x+y) \bullet(x+z)$ $\quad$ $x \bullet(y+z)=x \bullet y+x \bullet z$

8) **Involuzione**

$(x')'=x$

9) **Dualità (De Morgan)**

$(x_1+x_2+....+x_n)' = x'_1 \bullet x'_2 \bullet .... \bullet x'_n$

$(x_1 \bullet x_2 \bullet .... \bullet x_n)' = x'_1+x'_2+....+x'_n$

Teorema di Shannon

- Data una funzione Booleana $f(B^n)=B$ è sempre vero che:
  - $f(x_1,x_2,...,x_n) = x_1 f(1,x_2,...,x_n) + x_1' f(0,x_2,...,x_n)$
  - $f(x_1,x_2,...,x_n) = (x_1+f(0,x_2,...,x_n)) \bullet (x_1'+f(1,x_2,...,x_n))$

Funzione completamente specificata

- Una funzione **f** è detta *completamente specificata* se il suo valore è specificato in corrispondenza di tutto il dominio.

- Se il valore della funzione non è specificato in corrispondenza di alcune assegnazioni, la funzione è detta *non completamente specificata.*

- Tali assegnazioni individuano delle *condizioni di indifferenza* e il valore di f in corrispondenza di esse è denotato con il simbolo "-".

- Per tali configurazioni la funzione può assumere *indifferentemente* il valore 0 od il valore 1.

Funzione non completamente specificata

x	y	z	f(x,y,z)
0	0	0	-
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	-

Operatori funzionalmente completi

- La terna di operatori AND, OR e NOT sono funzionalmente completi ovvero permettono di rappresentare qualsiasi funzione di commutazione.

- La coppia (AND, NOT) è funzionalmente completa.
  -Dimostrazione: NOT(OR(x,y)) = AND(NOT(x),NOT(y))
  OR(x,y) = NOT(AND(NOT(x),NOT(y)))

- La coppia (OR, NOT) è funzionalmente completa.
  -Dimostrazione: NOT(AND(x,y)) = OR(NOT(x), NOT(y))
  AND(x,y) = NOT(OR(NOT(x), NOT(y)))

Operatori funzionalmente completi

- Esistono altri due operatori che sono funzionalmente completi:
  - NAND: NAND (x,y) = NOT (AND (x,y))
  - NOR: NOR (x,y) = NOT (OR (x,y))

Dimostrazione:

A partire dal NAND possiamo ottenere la coppia (AND,NOT):

- NOT(x) =NAND(x,x);
- AND(x,y) = NOT(NAND(x,y))

A partire dal NOR possiamo ottenere la coppia (OR,NOT):

- NOT(x) =NOR(x,x);
- OR(x,y) = NOT(NOR(x,y))

Espressioni booleane

- Si definisce ***espressione booleana*** una combinazione di variabili booleane e costanti (0,1) attuata mediante gli operatori +, •, ‘.

Es. $e = x \bullet y + x + z' + y \bullet w$

- Si definisce ***letterale*** ogni presenza in forma diretta o negata di una variabile in una espressione e ***numero di letterali*** il loro numero.

letterale

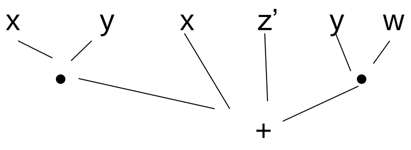
- Es. $e4 = x \bullet y' + y \bullet z$

L’espressione e4 ha 4 letterali

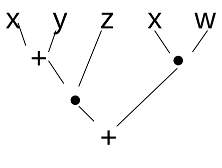
Espressioni booleane: numero di livelli

- Si definisce ***numero di livelli*** di una espressione il massimo tra i numeri di operazioni che agiscono in cascata sui letterali.

$e = x \bullet y + x + z' + y \bullet w$ è un’espressione a due livelli con 6 letterali



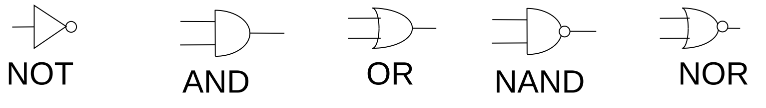
$e = (x+y) \bullet z + x \bullet w$ è un’espressione a tre livelli con 5 letterali



Porte Logiche

- L'algebra di commutazione può essere utilizzata per descrivere sistemi caratterizzati da grandezze fisiche che possono assumere 2 soli livelli logici.

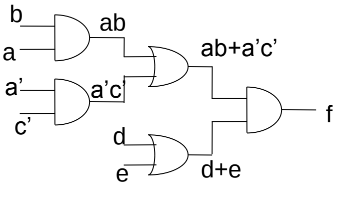
- Sono stati individuati sistemi molto semplici che realizzano le operazioni OR, AND, NOT, NAND, NOR a cui viene dato il nome di porte logiche



Porte Logiche

- Utilizzando le porte, si può far corrispondere ad una espressione booleana un insieme interconnesso di porte, detto rete logica

- F=(ab+a'c')(d+e)



Espressioni booleane

- Una espressione **e** può essere utilizzata per rappresentare quella funzione booleana f che assume
  - valore 1 in corrispondenza delle assegnazioni per le quali e=1
  - valore 0 in corrispondenza delle assegnazioni per le quali e=0.
- Due espressioni e1, e2 nelle stesse variabili sono equivalenti se e1=e2 per tutte le assegnazioni delle variabili.

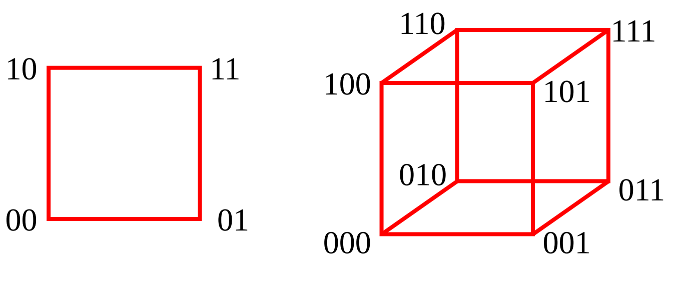
Es. Le espressioni

e1=xy'+xy'z+xz   e2=xy'+xz

sono equivalenti.

Forme canoniche

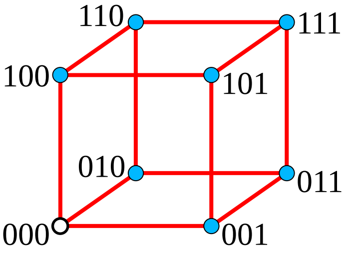
- Sia $f(B^n)=B$ una funzione Booleana ad una sola uscita completamente specificata
- Le 2^n configurazioni degli ingressi possono essere mappate sui vertici di un *n*-cubo in modo tale che due punti adiacenti siano a distanza di Hamming pari a 1



Forme canoniche

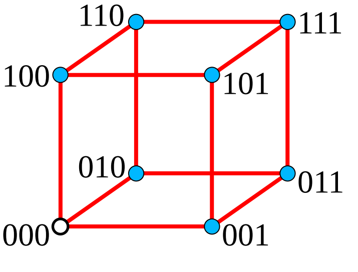
- Si consideri per esempio n=3 (variabili a, b, c)
- Gli spigoli del cubo saranno indicati con un pallino pieno se in corrispondenza di quel valore di ingresso la funzione vale 1 e con un pallino vuoto altrimenti

OR(a, b, c)



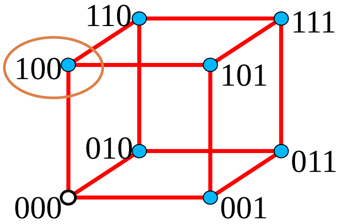
Forme canoniche: definizioni

- **Letterale:** E' una 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, ab'c, … sono implicanti per questa funzione

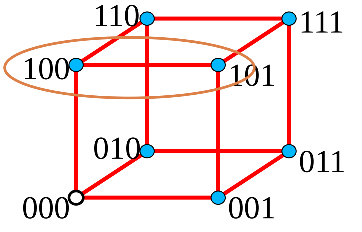


Forme Canoniche (Definizioni)

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



Implicante ab’c’



Implicante ab’

Forme Canoniche (Definizioni)

- **Mintermine:** un implicante in cui compaiono tutte le variabili di ingresso
  - a’b’c, a’bc’, a’bc, ab’c’, ab’c, abc’, abc
  - NOTA: I mintermini di una funzione vengono spesso identificati con il numero in base 10 corrispondente al valore binario del mintermine. (Es., ab’c’ = 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 di una funzione
- ***Off-set***: Insieme dei maxtermini di una funzione

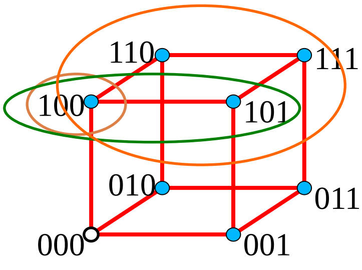
Forme Canoniche (Definizioni)

- **Implicante primo:** implicante tale che non esiste nessun altro implicante di dimensioni maggiori che lo contenga interamente
- **Es.:** L'implicante ab'c' 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

Implicante ab'c'

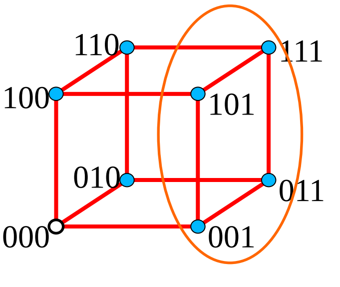
Implicante ab'

Implicante **a**



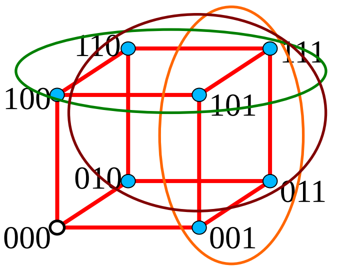
Forme Canoniche (Definizioni)

- **Implicante essenziale:** implicante primo che copre almeno un mintermine non coperto da altri implicanti primi
- **Es.:** L'implicante primo c è essenziale perchè copre il mintermine a'b'c 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**, a'bc'} è una copertura.



Prima forma canonica

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

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

Seconda forma canonica

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

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

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= a'b'c'	quando a=0,b=0,c=0 $m_0=1$
M_1= a+b+c'	quando a=0,b=0,c=1 $M_1=0$
M_2= a+b'+c	quando a=0,b=1,c=0 $M_2=0$
m_3= a'bc	quando a=0,b=1,c=1 $m_3=1$
m_4= ab'c'	quando a=1,b=0,c=0 $m_4=1$
M_5= a'+b+c'	quando a=1,b=0,c=1 $M_5=0$
M_6= a'+b'+c	quando a=1,b=1,c=0 $M_6=0$
m_7=abc	quando a=1,b=1,c=1 $m_7=1$

On-set = {m0, m3, m4, m7}

PFC: m0 + m3 + m4 + m7

PFC: a'b'c' + a'bc + ab'c' + abc

Off-set = {M1, M2, M5, M6}

SFC: M1• M2 • M5 • M6

SFC: (a+b+c') (a+b'+c) (a'+b+c') (a'+b'+c)

Notazione contratta di forma canonica SP

Data una funzione booleana f, se consideriamo i numeri decimali p_i corrispondenti alle configurazioni P_i delle variabili in cui è presente un 1, la f può essere rappresentata come sommatoria dei p_i.

$$f(x_1,x_2,..,x_n)= \Sigma_i(p_i)$$

Per la funzione

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$f(x,y,z)= \Sigma(1,4,5,7)$

Notazione contratta di forma canonica PS

Data una funzione booleana f, se consideriamo i numeri decimali s_i corrispondenti alle configurazioni S_i delle variabili in cui è presente uno 0, la f può essere rappresentata come produttoria dei s_i.

$$f(x_1,x_2,..,x_n)= \Pi_i(s_i)$$

Per la funzione

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$f(x,y,z)= \Pi(0,2,6)$

Notazione contratta per funzioni non completamente specificate

Data una funzione booleana f non completamente specificata, le configurazioni S_i relative alle condizioni di indifferenza vengono aggiunte a quelle in cui la funzione è specificata mediante un ulteriore sommatoria o produttoria relativa solo a tali configurazioni.

$f(x_1,x_2,..,x_n)= \Sigma_i(p_i)+ d_\Sigma (p_i)$

$f(x,y,z)= \Sigma(1,4,5,7)+d_\Sigma (2,3)$

$f(x_1,x_2,..,x_n)= \Pi_i(s_i)+ d_{\Pi i}(s_i)$

$f(x,y,z)= \Pi(0,2,6)+d_{\Pi}(1,3)$