



# La seconda forma canonica Circuiti notevoli

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)

Università degli Studi di Milano

Riferimenti: Sezione B3.



## Sommario

Implementazione circuitale mediante PLA o ROM.

La seconda forma canonica.

Circuiti combinatori notevoli



## Circuiti combinatori



- Circuiti logici digitali in cui le operazioni (logiche) dipendono solo da una combinazione degli input.
- Circuiti senza memoria. Ogni volta che si inseriscono in ingresso gli stessi valori, si ottengono le stesse uscite. Il risultato non dipende dallo stato del circuito.
- I circuiti combinatori descrivono delle funzioni Booleane. Queste funzioni si ottengono combinando tra loro (in parallelo o in cascata) gli operatori logici: **NOT, AND, OR**.
- Il loro funzionamento può essere descritto come **tabella della verità**.
- Come nelle funzioni algebriche, il risultato è aggiornato immediatamente dopo il cambiamento dell'input (si suppone il tempo di commutazione trascurabile, tempo di attesa prima di guardare l'output sufficientemente ampio per permettere a tutti i circuiti la commutazione).



## Funzione come espressione logica o come tabella delle verità



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

A B C	A and B	B and $\bar{C}$	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



## Razionale della prima forma canonica



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

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

$$F = 1$$

iif

$$A = 0 \ B = 1 \ C = 0$$

OR

$$A = 1 \ B = 1 \ C = 0$$

OR

$$A = 1 \ B = 1 \ C = 1$$



## Forme canoniche



- Esiste un metodo per ricavare automaticamente un circuito che implementi una tabella di verità?
- Esistono 2 forme canoniche (equivalenti) che garantiscono di poter realizzare una qualunque tabella di verità con solo due livelli di porte OR, AND e NOT:

1) Somme di Prodotti (SOP)

$$F = \sum_{i=1}^Q m_i$$

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

2) Prodotti di Somme (POS)



## Tipi di circuiti che implementano le SOP



In generale abbiamo funzioni logiche booleane multi-input / multi-output.

- Logica distribuita.
- PLA: Programmable Logic Array: matrici regolari AND e OR in successione, personalizzabili dall'utente.
- ROM: Read Only Memory circuiti ad hoc che implementano una particolare funzione in modo irreversibile.



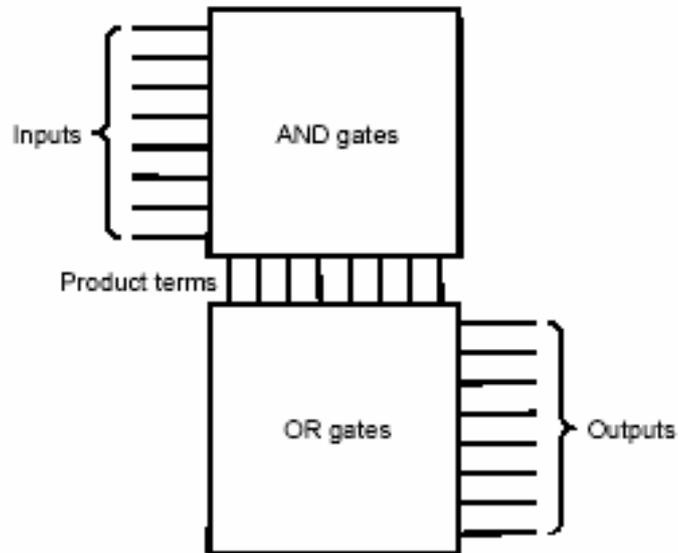
## PLA (Programmable Logica Array)



- La matrice degli AND ha  $n$  linee di ingresso: ciascuna porta ha in ingresso le  $n$  linee e il loro complemento.
- L'utente fornisce la matrice che dice quale linea entra (e come) in quale porta AND:  
Crea la matrice dei mintermini, bruciando in ingresso alle porte AND le linee che non servono.
- Le uscite della matrice AND entrano nella matrice OR programmata come la precedente in base ad un'altra matrice fornita dall'utente  
Si utilizza una porta OR per ogni funzione calcolata.



## Struttura di una PLA

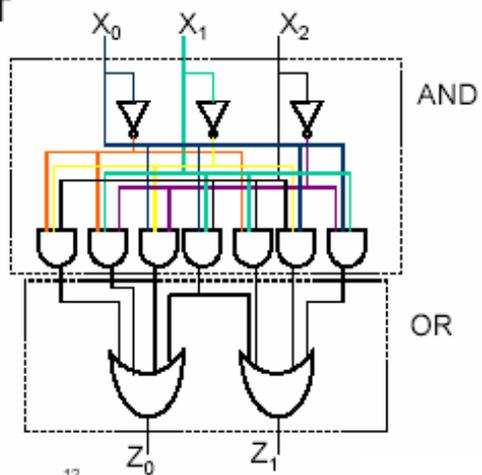


## Esempio di PLA



- Realizzare con un PLA la funzione descritta dalla seguente TT

$X_0$	$X_1$	$X_2$	$Z_0$	$Z_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





## Esercizi sulla PLA



Realizzare mediante PLA con 3 ingressi:

- la funzione maggioranza.
- la funzione che vale 1 se e solo se 1 solo bit di ingresso vale 1
- un decoder
- la funzione che vale 0 se l'input è pari, 1 se dispari
- la funzione che calcola i multipli di 3 (con 4 ingressi)



## Rappresentazione circuitale mediante ROM



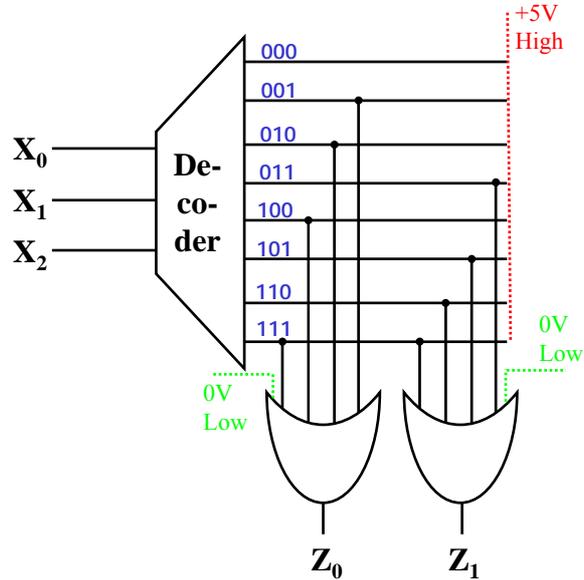
- Read Only Memory, memoria di sola lettura.  
Funge anche da modulo combinatorio a uscita multipla.
- $n$  linee di ingresso,  $m$  linee di uscita (ampiezza)  
a ciascuna delle  $2^n$  (altezza) configurazioni di ingresso (parole di memoria) è associata permanentemente una combinazione delle  $m$  linee di uscita.
- l'input seleziona la parola da leggere di  $m$  bit, che appare in uscita
- realizzato con un decoder  $n-a-2^n$  seguito da una matrice di  $m$  porte OR.



## ROM - esempio



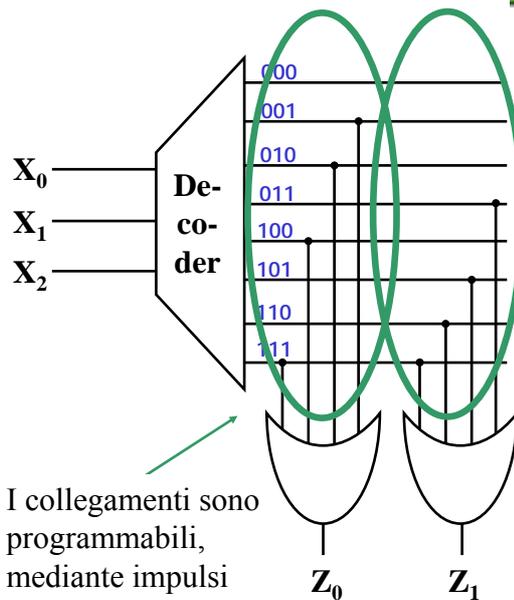
$X_0$	$X_1$	$X_2$	$Z_0$	$Z_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



## EEPROM



$X_0$	$X_1$	$X_2$	$Z_0$	$Z_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



I collegamenti sono programmabili, mediante impulsi elettronici.



## Confronto PLA - ROM



ROM – fornisce un'uscita per ognuna delle combinazioni degli ingressi. Decoder con  $2^n$  uscite, dove  $n$  è il numero di variabili in ingresso alla ROM. Crescita esponenziale delle uscite.

- approccio più generale. Può implementare una qualsiasi funzione, dato un certo numero di input e output.

PLA – contiene solamente i mintermini in uscita al primo stadio. Il loro numero cresce meno che esponenzialmente.

FPGA – connettività libera. Non è una struttura a 2 livelli.



## Sommario



Implementazione circuitale mediante PLA o ROM.

**La seconda forma canonica.**

Circuiti combinatori notevoli.



## Razionale della seconda forma canonica



$$F = \text{NOT} (\bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B C + A \bar{B} \bar{C} + A \bar{B} C)$$

$$F = 0$$

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

iif

$$(A = 0 \ B = 0 \ C = 0)$$

OR

$$(A = 0 \ B = 0 \ C = 1)$$

OR

$$(A = 0 \ B = 1 \ C = 1)$$

OR

$$(A = 1 \ B = 0 \ C = 0)$$

OR

$$(A = 1 \ B = 0 \ C = 1)$$



## Verso la seconda forma canonica



**Maxtermine,  $M_j$ :** e' un prodotto di tutte le variabili di ingresso al quale corrisponde un valore 0 per la funzione. (e.g.  $A \bar{B} \bar{C}$ ).

j indica il numero progressivo in base 10.

Possibile espressione della seconda forma canonica:

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

$$W \leq 2^N$$

$$Q + W = 2^N$$

$$F = \sum_{i=1}^W M_i$$

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



## Verso la seconda forma canonica: manipolazioni algebriche



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

$$F = \sum_{i=1}^W M_i$$

Applico De Morgan

$$F = \prod_{i=1}^W \overline{M_i}$$

$$F = (\overline{A} \overline{B} \overline{C})(\overline{A} \overline{B} C)(\overline{A} B C)(A \overline{B} \overline{C})(A \overline{B} C)$$

F = 1 quando nessun fattore si annulla



## Razionale della seconda forma canonica



$$F = (\overline{A} \overline{B} \overline{C})(\overline{A} \overline{B} C)(\overline{A} B C)(A \overline{B} \overline{C})(A \overline{B} C)$$

**F = 1**

iif

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

!(A = 0 B = 0 C = 0)

AND

!(A = 0 B = 0 C = 1)

AND

!(A = 0 B = 1 C = 1)

AND

!(A = 1 B = 0 C = 0)

AND

!(A = 1 B = 0 C = 1)



## La seconda forma canonica: prodotto di somme



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

$$F = \prod_{i=1}^w \overline{M_i} \quad M_0 = \overline{A} \overline{B} \overline{C} \Rightarrow \overline{M_0} = A + B + C$$

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

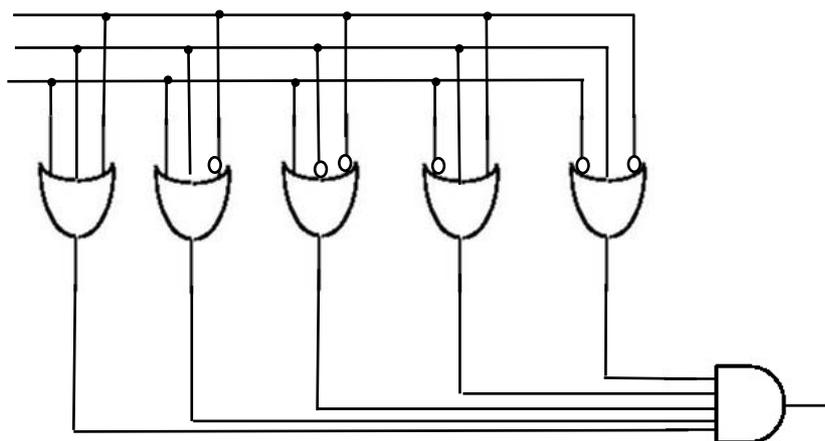
Applicando il secondo teorema di De Morgan:

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

F = 1 quando nessun fattore si annulla



## Il circuito della seconda forma canonica: POS



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



## Esempio 2



$$F = \overline{\overline{A}BC} + \overline{A\overline{B}C} = \overline{A}C$$

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$$F = \overline{M_1} \overline{M_3} \overline{M_4} \overline{M_5} \overline{M_6} \overline{M_7}$$

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



## Sommario



Implementazione circuitale mediante PLA o ROM.

La seconda forma canonica.

Circuiti combinatori notevoli.



# XOR – POS & SOP



a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

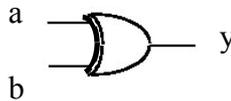
SOP:  $y = \bar{a}b + a\bar{b}$

POS:  $y = (a+b)(\bar{a}+\bar{b})$

$$y = (\bar{a}+\bar{b})(a+b) = (\bar{a}a) + (\bar{a}b) + (\bar{b}a) + (\bar{b}b) =$$

$$= \bar{a}b + a\bar{b} \quad \text{c.v.d.}$$

$y = a \oplus b$



# Uscite indifferenti di un tabella delle verità

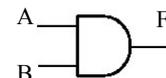


A	B	F
0	0	0
0	1	X
1	0	0
1	1	1

Ho 2 possibilità:

$X = 0 \quad F = A B$

$X = 1 \quad F = \bar{A} B + A B = B$



Diminuisce il numero di porte e si accorcia il cammino critico.



## Decodificatore (decoder)



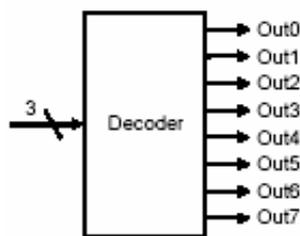
- E' caratterizzato da n linee di input e  $2^n$  linee di output
- il numero binario espresso dalla configurazione delle linee di input è usato per asserire la sola linea di output di ugual indice.
- es.: con 4 linee di input e 16 di output (da 0 a 15), se in ingresso arriva il valore 0110, in uscita si alza la linea di indice 5 (la sesta!).
- utilizzato per indirizzare la memoria (cf. ROM).



## La funzione decoder



### Decoder



a. A 3-bit decoder

A	B	C	$U_0$	$U_1$	$U_2$	$U_3$	$U_4$	$U_5$	$U_6$	$U_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Le funzioni di uscita sono  $2^n$  per n input:

$$U_0 = \sim A \sim B \sim C$$

...

$$U_7 = A B C$$

$$U_j = m_j$$



## Codificatore (encoder)



- E' caratterizzato da n linee di input e  $\text{ceil}(\log_2 n)$  linee di output
- Una sola linea di ingresso può essere attiva.
- il numero binario espresso dalla configurazione delle linee di output rappresenta la linea di ingresso attiva.
- es.: con 16 linee di input e 4 di output, se in ingresso arriva il valore 0000 0100 0000 0000, in uscita leggiamo il numero 10.



## La funzione encoder



ABCD	$Y_1 Y_2$
0000	X X
0001	0 0
0010	0 1
0011	X X
0100	1 0
0101	X X
0110	X X
0111	X X
1000	1 1
1001	X X
1010	X X
1011	X X
1100	X X
1101	X X
1110	X X
1111	X X

Codifica quattro linee in ingresso: 0,1,2,3

$$Y_1 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} = \bar{C}\bar{D} (A \oplus B)$$

$$Y_2 = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} = \bar{B}\bar{D} (A \oplus C)$$

Come è conveniente impostare le X?



## La funzione encoder: ottimizzazione



ABCD	$Y_1 Y_2$
0000	0 X
0001	0 0
0010	0 1
0011	0 X
0100	1 0
0101	1 X
0110	1 X
0111	1 X
1000	1 1
1001	1 X
1010	1 X
1011	1 X
1100	0 X
1101	0 X
1110	0 X
1111	0 X

Consideriamo la prima funzione,  $Y_1$

$$Y_1 = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} = \bar{C}\bar{D} (A \oplus B)$$

Specifichiamo in modo opportuno i valori di uscita indifferenti, X

$$Y_1 = \bar{A} B \bar{C} \bar{D} + A \bar{B} \bar{C} \bar{D} + \bar{A} B C \bar{D} + A \bar{B} C \bar{D} + \bar{A} B C D + A \bar{B} C D + A \bar{B} \bar{C} D + A B \bar{C} D = (A \oplus B)$$



## Multiplexer



- E' caratterizzato da n linee di input (data),
- k linee di controllo (**selezione**).
- In base alla linea di controllo viene connessa all'uscita la linea di ingresso selezionata (cf. ROM).
- Quante linee di controllo, k, servono?

$$k = \text{ceil}(\log_2 n)$$

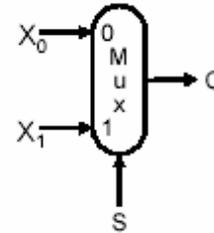
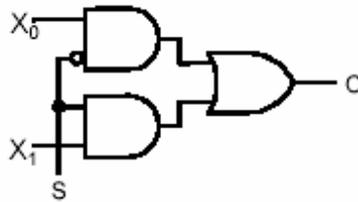
Esempio: con 4 linee di input (da 0 a 3), se sulle linee di controllo c'è 11, in uscita si avrà il valore presente sulla linea 3



## Multiplexer



S	x <sub>0</sub>	x <sub>1</sub>	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Il segnale di selezione S, “apre” la porta opportuna, cioè chiude il cammino opportuno.



## Sintesi della funzione Mux



S	x <sub>0</sub>	x <sub>1</sub>	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$\begin{aligned} \text{SOP: } C &= \bar{S} x_0 \bar{x}_1 + \bar{S} x_0 x_1 + S x_0 \bar{x}_1 + S x_0 x_1 \\ &= \bar{S} x_0 + S x_1 \quad \text{cvd} \end{aligned}$$

Il mux si comporta come interruttore



## Sintesi della funzione Mux nella forma POS



S	x <sub>0</sub>	x <sub>1</sub>	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$\text{POS: } C = (S+x_0+x_1)(S+x_0+\bar{x}_1)(\bar{S}+x_0+x_1)(\bar{S}+\bar{x}_0+x_1) =$$

$$\text{Definisco: } a = \bar{S}+x_1 \\ b = S+x_0$$

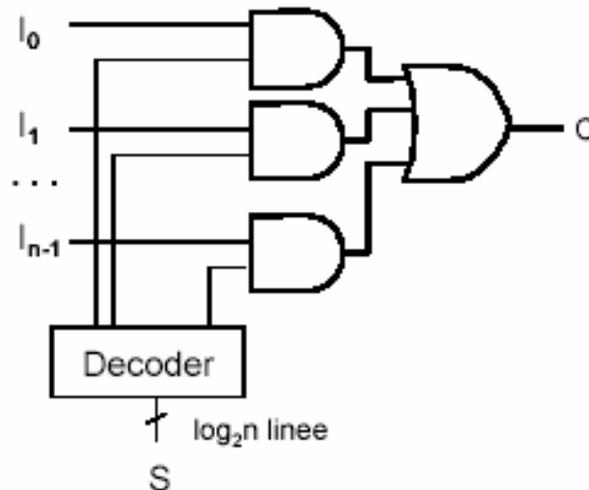
$$[(b+x_1)(\bar{b}+\bar{x}_1)] [(a+x_0)(\bar{a}+\bar{x}_0)] =$$

$$[b+x_1\bar{x}_1] [a+x_0\bar{x}_0] = ba = (\bar{S}+x_1)(S+x_0) =$$

$$\bar{S}S + \bar{S}x_0 + Sx_1 + x_0x_1 = \bar{S}x_0 + Sx_1 \quad \text{cvd}$$



## Mux a più vie.



Una sola porta alla volta viene aperta dal segnale S. Le porte sono mutuamente esclusive.



## Comparatore



- E' caratterizzato da 2 insiemi di n linee di ingresso ciascuna e un output.
- L'output vale 1 se i due insiemi di bit hanno uguale valore, 0 se sono diversi.

$A_0$	$B_0$	$C_0$	$A_1$	$B_1$	$C_1$	
0	0	1	0	0	1	$C = C_0 C_1 \dots C_{n-1}$
0	1	0	0	1	0	
1	0	0	1	0	0	
1	1	1	1	1	1	

...

$$C_k = \overline{a_k \oplus b_k}$$


## Sommario



Implementazione circuitale mediante PLA o ROM.

La seconda forma canonica.

Circuiti combinatori notevoli.