



# I circuiti digitali: dalle funzioni logiche ai circuiti

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

Università degli Studi di Milano

Riferimento al testo: Sezione B.3; Approfondimento sulle forme canoniche:  
Fummi et al., Progettazione Digitale, McGrawHill, capitolo 3.



## Sommario

### I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

Criteri di ottimalità.

Implementazione circuitale mediante PLA o ROM.



## 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



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

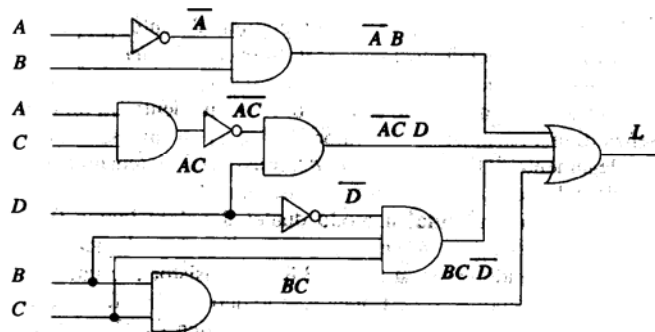
A	B	C	A and B	B and not(C)	F
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1



## Esempio – rappresentazione 1



$$L = \overline{A}B + \overline{AC}D + BC + BC\overline{D}$$

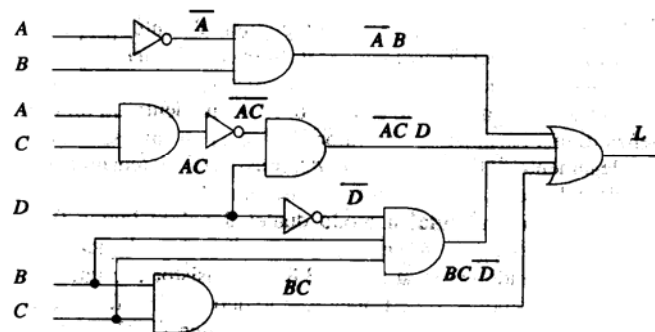


## Esempio – tabella verità



A	B	C	D	L
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$$L = \overline{A}B + \overline{AC}D + BC + BC\overline{D}$$





## Sommario



I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

Criteri di ottimalità.

Implementazione circuitale mediante PLA o ROM.



## Dall'espressione logica al circuito



Ad ogni espressione logica corrisponde un circuito, ad ogni circuito corrisponde una tabella delle verità, ad ogni tabella della verità, in generale, **non corrisponde** un unico circuito possibile.

- Esistono più espressioni tra loro equivalenti: 2 espressioni sono equivalenti se hanno la stessa tabella di verità.
- Quale è la "migliore"?
- È possibile trovare un metodo di semplificazione sfruttando le proprietà dell'algebra booleana.
- Esistono tecniche automatiche o semi-automatiche di semplificazione.



# Le espressioni logiche



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

Si possono dimostrare sostituendo 0/1 alle variabili.



# XOR – Rappresentazioni equivalenti

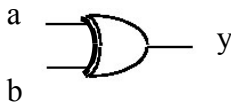


a	b	y	
0	0	0	SOP: $y = \overline{a}b + a\overline{b}$ POS: $y = (a+b)(\overline{a}+\overline{b})$
0	1	1	
1	0	1	
1	1	0	

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

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

$$y = a \oplus b$$





## Manipolazione algebrica



Applichiamo De Morgan.

$$L = \overline{A} \overline{B} + \overline{AC} \overline{D} + \overline{BC} \overline{D} + \overline{BC} + =$$

$$= \overline{A+B} + \overline{AC+D} + \overline{BC+D} + \overline{B+C} + = \quad \overline{xy} = \overline{x+y}$$

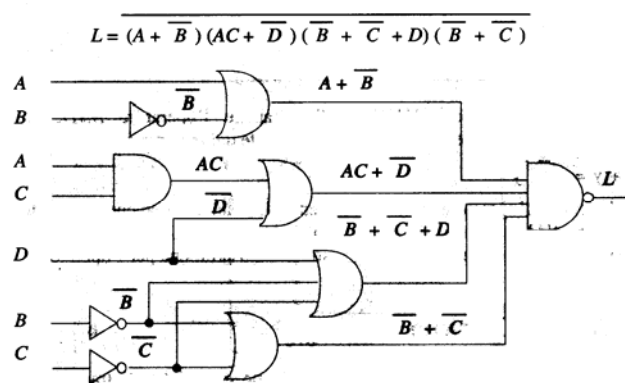
$$= (\overline{A+B})(\overline{AC+D})(\overline{BC+D})(\overline{B+C}) \quad \overline{x+y} = \overline{xy}$$



## Esempio – rappresentazione 2



A	B	C	D	L
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1





## Sommario



I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

Criteri di ottimalità.

Implementazione circuitale mediante PLA o ROM.



## Funzione come espressione logica o come tabella delle verità



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

$$F = 1$$

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

$$A = 0 \text{ B} = 1 \text{ C} = 0$$

OR

$$A = 1 \text{ B} = 1 \text{ C} = 0$$

OR

$$A = 1 \text{ B} = 1 \text{ C} = 1$$



## La prima forma canonica



$F = (A \text{ AND } B) \text{ OR } (B \text{ AND NOT}(C)) =$  **Implicante:** prodotto delle variabili (in forma naturale o negata) per le quali la funzione vale 1

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

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

Mintermine,  $m_j$ : un implicante che contiene tutte le N variabili della funzione (e.g. ABC).

j indica il numero progressivo in base 10.

$$\text{Prima forma canonica: } F = \sum_{i=1}^Q m_i$$

$$Q \leq 2^N$$

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



## La SOP è la prima forma canonica



- La forma canonica di una funzione è la somma dei suoi mintermini.
- Qualunque funzione è esprimibile in forma canonica.

Esempio:  $Z(A,B,C,D) = AC + BC + \sim A \sim B \sim C$

$$= A(B + \sim B)C(D + \sim D) + (A + \sim A)BC(D + \sim D) + \sim A \sim B \sim C(D + \sim D)$$

$$= \sim A \sim B \sim C \sim D + \sim A \sim B \sim C D + \sim A B C \sim D + \sim A B C D + A \sim B C \sim D + A \sim B C D + A B C \sim D + A B C D$$

La stessa espressione si ricaverebbe dalla tabella della verità:

$$Z = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD =$$

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

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





## Perchè SOP è una forma canonica



- Forma universale mediante la quale è possibile rappresentare qualunque funzione booleana.
- In generale una forma canonica non è una forma ottima, ma un punto di partenza per l'ottimizzazione.
- Si basa su componenti caratterizzanti la struttura della funzione (mintermine), che traducono le condizioni logiche espresse dalla funzione.

Mintermine,  $m_i$ :

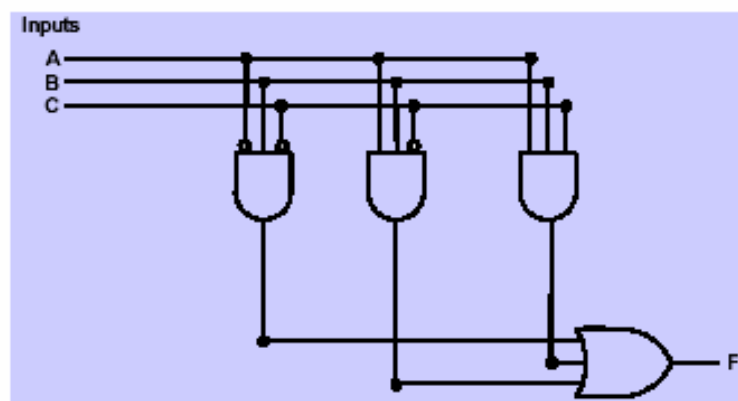
- E' una funzione booleana a n ingressi che vale 1 in corrispondenza della sola i-esima configurazione di ingresso.
- Al massimo,  $2^n$  mintermini per ogni n variabili.
- ogni mintermine è rappresentabile con un AND con n ingressi.



## Il circuito della prima forma canonica: SOP

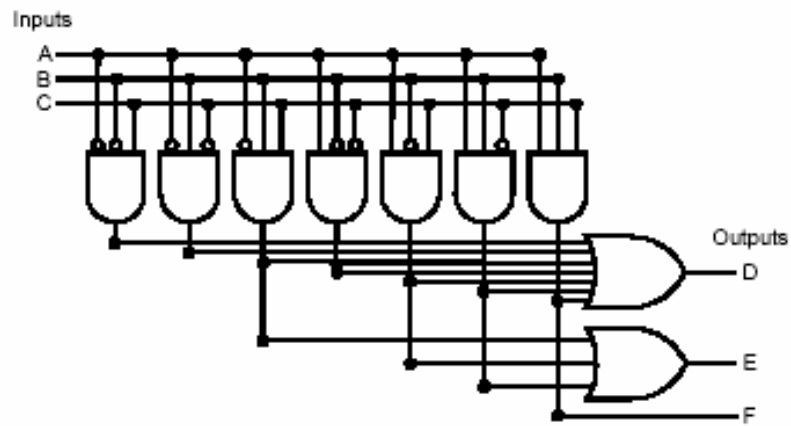


$$F = \sim AB\sim C + AB\sim C + ABC$$





## SOP a più uscite



Ricavare la funzione in forma di tabella della verità?



## Dalla SOP al circuito: osservazioni



- Dalla forma canonica (somma di mintermini) è facile passare al circuito:
  - Ogni mintermine è un AND
  - Tutti gli AND entrano in un OR
- Implementazione regolare
- Solo due livelli di porte
- Blocchi generali personalizzabili purché ci sia un numero sufficiente di componenti elementari.



## Sommario



I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

**Criteri di ottimalità.**

Implementazione circuitale mediante PLA o ROM.



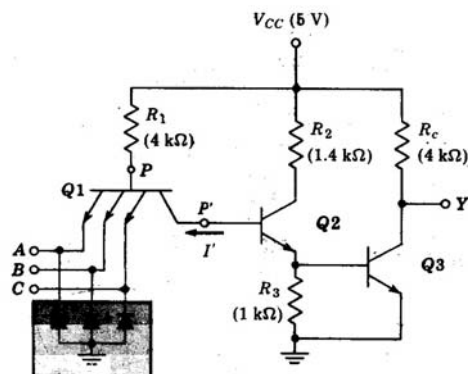
## Valutazione della semplicità di un circuito



Area (numero di porte)

Tempo di commutazione (numero di transistor attraversati)

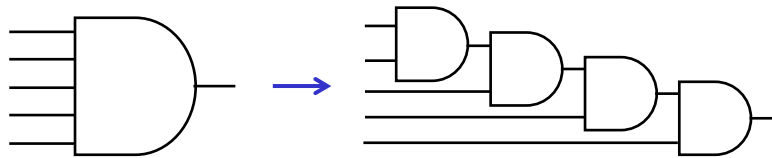
Soddisfazione di vincoli, potenza dissipata, facilità di debug...





## Implementazione con porte a 2 ingressi

- Gli elementi costruttivi di base tipici sono porte a 2 ingressi
  - Porta a N ingressi  $\rightarrow$  N-1 porte a 2 ingressi

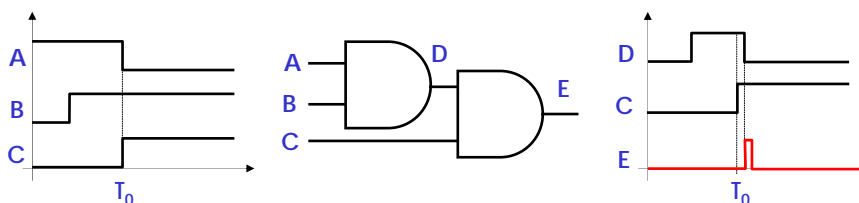


Cammino Critico: N-1



## Cammino critico

- Ogni circuito logico è caratterizzato da un **tempo di commutazione**
  - Più porte devo attraversare, più è lungo il tempo della **transizione del circuito nel suo complesso**.
- **CAMMINO CRITICO**
  - max numero di porte da attraversare da ingresso a uscita
  - Non si contano gli inverters (inclusi nelle porte)



A e C commutano contemporaneamente in  $T_0$ , E raggiunge il valore corretto dopo un tempo  $2 \Delta T$  (la commutazione di D segue la commutazione di B con un ritardo  $\Delta T$ ).



## Semplificazioni notevoli



Dimostrare che:  $A + \sim AB = A + B$

*Proprietà distributiva di OR rispetto ad AND:*

$$A + \sim AB = (A + B)(A + \sim A)$$

*Sviluppando il prodotto:*

$$(A + B)(A + \sim A) = AA + A \sim A + BA + B \sim A = A + AB + \sim AB$$

Raccogliendo A:

$$A + AB + \sim AB = A + (A + \sim A)B = A + B$$

Dimostrare che:  $(A + \sim B)(B + C) = AB + AC + \sim BC$



## Esempio di semplificazione algebrica



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

*Raccogliendo B AND NOT(C):*

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

*Proprietà dell'inverso: "NOT(A) OR A = 1"*

$$= 1 \text{ AND } B \text{ AND NOT}(C) \text{ OR } A \text{ AND } B \text{ AND } C =$$

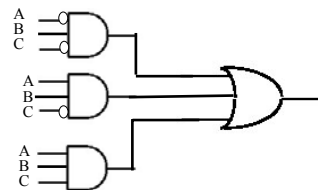
*Proprietà dell'identità: "1 AND B = B"*

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

*Dalla slide precedente:*

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

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



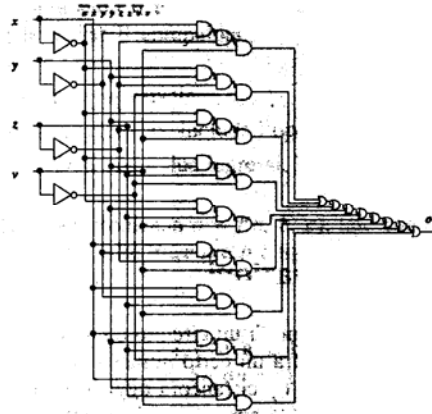
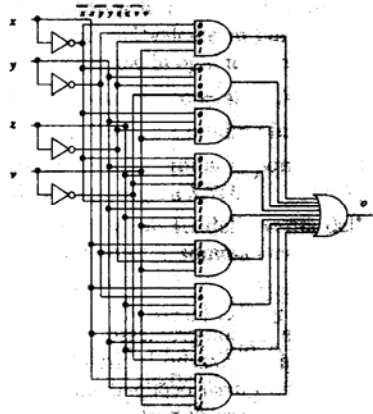


## Riduzione del cammino critico



$$o = \bar{x}y\bar{z}v + \bar{x}yz\bar{v} + \bar{x}y\bar{z}\bar{v} + \bar{x}yzv + \bar{x}y\bar{z}v + \bar{x}yzv + \bar{x}y\bar{z}v +$$

$$\bar{x}y\bar{z}v + \bar{x}yz\bar{v} + \bar{x}yzv$$



**Cammino critico** pari a 11: cammino più lungo in circuiti con porte a 2 ingressi.  
Numero di porte: 35.

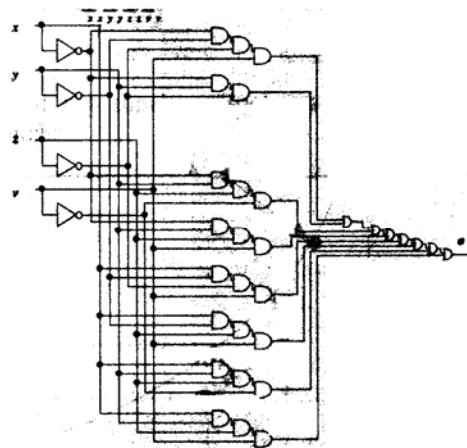
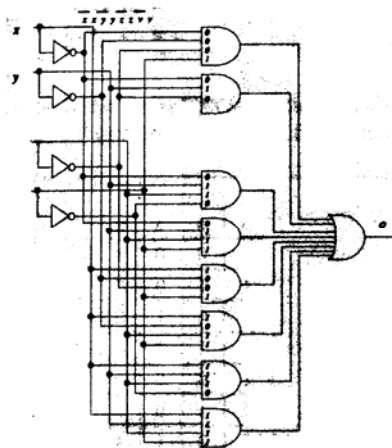


## Semplificazione



2° e 3° AND

$$o = \bar{x}y\bar{z}v + \bar{x}yz\bar{v} = \bar{x}y\bar{z}(v + \bar{v}) = \bar{x}y\bar{z}$$



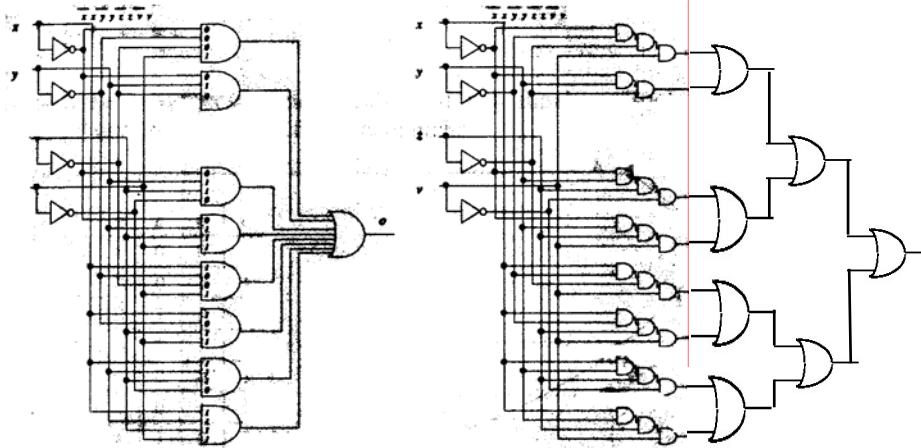
**Cammino critico** pari a 10. Numero di porte pari a 30.



## Ottimizzazione del cammino critico



Riorganizzando gli OR



Cammino critico pari a 6. Numero di porte pari a 30.



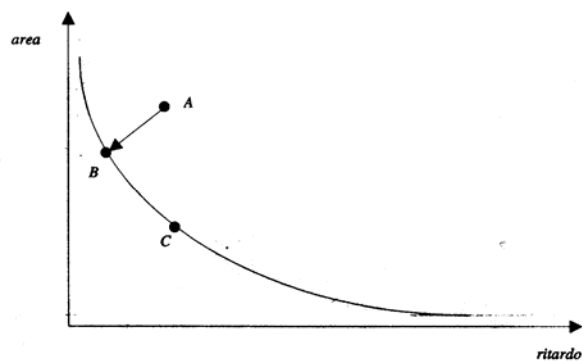
## Criteri di progetto



I circuiti a 2 livelli sono difficili da realizzare.

Si utilizzano insiemi di porte logiche (FPGA), la cui cablatura viene ricavata mediante complesse procedure di ottimizzazione che tengono conto del cammino critico, numero di porte, ....

Questa procedura si chiama "fitting".





## Sommario



I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

Criteri di ottimalità.

**Implementazione circuitale mediante PLA o ROM.**



## 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.





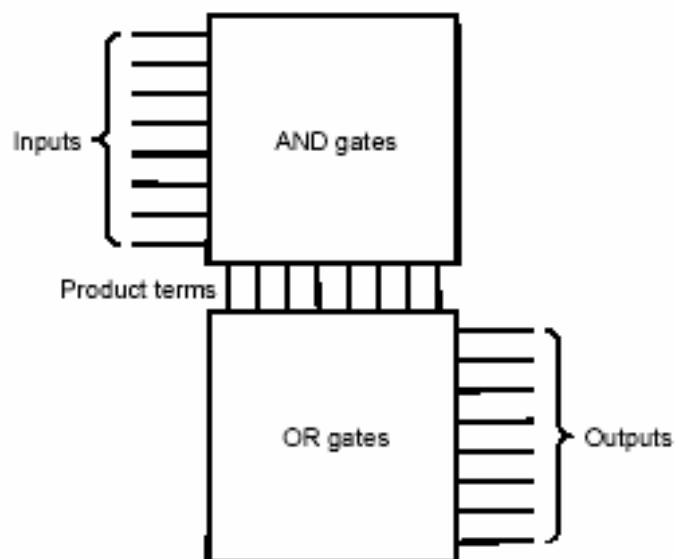
## PLA (Programmable Logica Array)



- 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



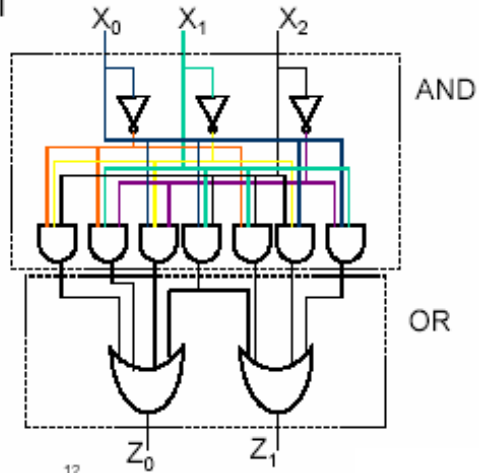


## Esempio di PLA



- 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



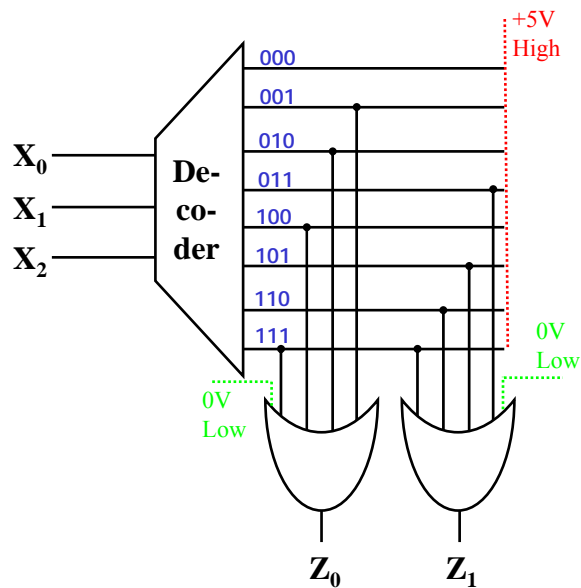
- Read Only Memory, memoria di sola lettura.  
Funge anche da modulo combinatorio a uscita multipla.
- n linee di ingresso, m linee di uscita (ampiezza)  
a ciascuna delle  $2^n$  (altezza) configurazioni di ingresso  
(parole di memoria) è associata permanentemente una  
combinazione delle m linee di 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

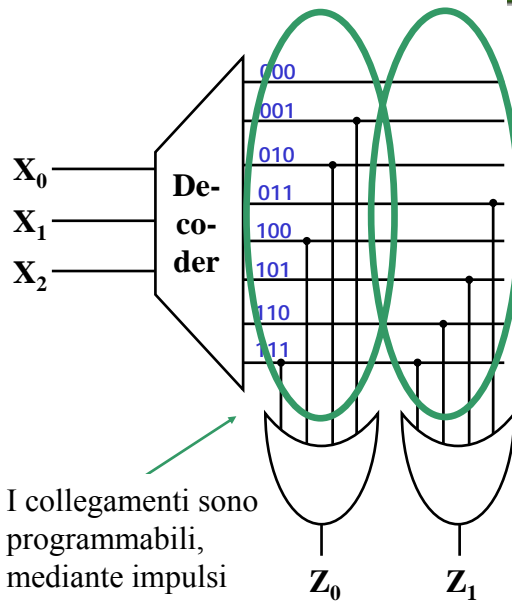




## EEPROM



$X_0$	$X_1$	$X_2$	$Z_0$	$Z_1$
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



## 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



I circuiti combinatori.

Dall'espressione logica al circuito. Semplificazione algebrica.

Dalla tabella della verità al circuito: la prima forma canonica: SOP.

Criteri di ottimalità.

Implementazione circuitale mediante PLA o ROM.