



# I circuiti digitali: dalle funzioni logiche ai circuiti (le SOP)

Prof. Alberto Borghese  
Dipartimento di Informatica  
[borgnese@di.unimi.it](mailto:borgnese@di.unimi.it)

Università degli Studi di Milano

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



## Sommario

### I circuiti combinatori.

Semplificazione algebrica.

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

Criteri di ottimalità.



## Circuiti combinatori



- Circuiti logici digitali in cui le operazioni (logiche) dipendono solo da una combinazione degli input. 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).
- 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**.
- Dato un circuito è univoca l'espressione algebrica che ne rappresenta il funzionamento.



## Un po' di tassonomia



- Funzione logica. Corrispondenza tra un insieme di ingresso (valori possibili di A, B, C) e insieme di uscita (valori possibili di F)

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

- Espressione logica. Combinazione di operatori logici che implementa la funzione logica. Ad ogni espressione logica è associato un ben preciso circuito.

$$\overline{A}B + BC$$



## Regole manipolazione algebrica



Doppia Inversione	$\overline{\overline{x}} = x$	
	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	$x \overline{x} = 0$	$x + \overline{x} = 1$
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	$x (y + z) = x y + x z$	$x + y z = (x + y) (x + z)$
Assorbimento	$x (x + y) = x$	$x + x y = x$
De Morgan	$\overline{xy} = \overline{x} + \overline{y}$	$\overline{x + y} = \overline{x} \overline{y}$

Si possono dimostrare sostituendo 0/1 alle variabili.



## Regole algebriche su più variabili



Commutativa	$x y z = y x z = z x y$	$x + y + z = y + x + z = z + x + y$
	AND rispetto ad OR	OR rispetto ad AND
Distributiva	$x (y h + z) = x y h + x z$	$x h + y z = (x h + y) (x h + z)$
De Morgan	$\overline{xyz} = \overline{x} + \overline{y} + \overline{z}$	$\overline{x + y + z} = \overline{x} \overline{y} \overline{z}$

Si possono dimostrare sostituendo 0/1 alle variabili.



## Funzione come espressione logica o come tabella delle verità



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

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



## Una seconda rappresentazione



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

Applico De Morgan ai prodotti logici:

$$F = \overline{\overline{A + B} + \overline{B + C}} \quad \text{NB } !B \text{ e } !B \text{ non si sommano!!}$$

Applico ancora De Morgan e ottengo:

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

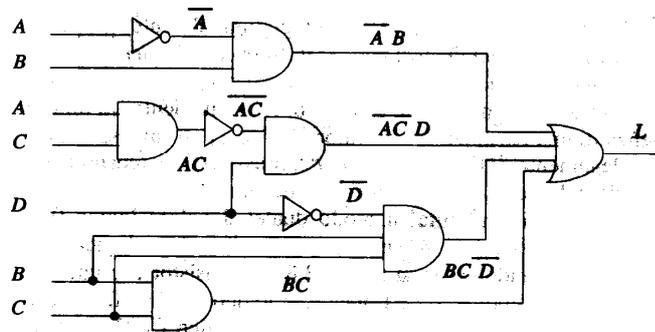
A B C	F
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1



## Esempio – rappresentazione 1



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

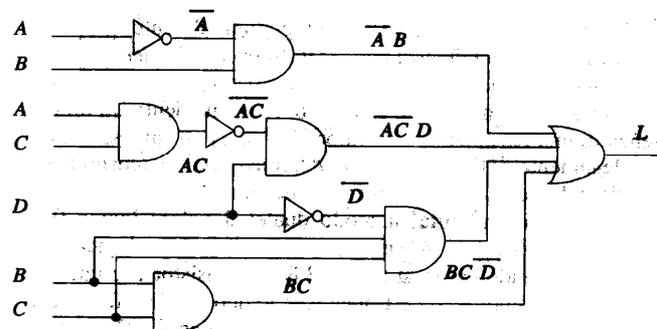


## Esempio – tabella verità



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

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





## Manipolazione algebrica



Applichiamo De Morgan.

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

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

$$\overline{xy} = \overline{x+y}$$

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

$$\overline{x+y} = \overline{xy}$$

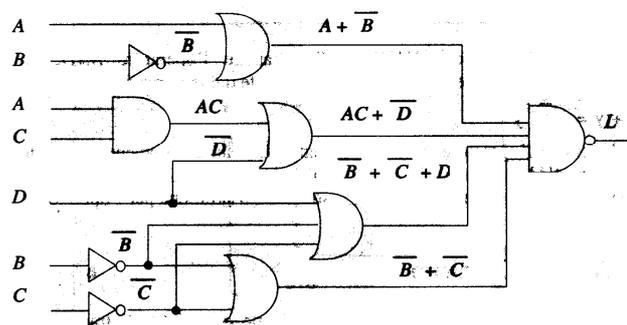


## Esempio – rappresentazione 2



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

$$L = \overline{(A+B)(AC+D)(B+C+D)(B+C)}$$





## Sommario



I circuiti combinatori.

**Semplificazione algebrica.**

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

Criteri di ottimalità.



## Semplificazioni notevoli



Dimostrare che:  $A + \overline{A}B = A + B$

*Proprietà distributiva di OR rispetto ad AND:*

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

*Sviluppando il prodotto:*

$$(A + \overline{A})(A + B) = AA + A\overline{A} + \overline{A}A + \overline{A}B = A + \overline{A}B$$

Raccogliendo A:

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

Dimostrare che:  $(A + \overline{B})(B + C) = \overline{A}B + AC + \overline{B}C$

Dimostrare che:  $\overline{A} + \overline{A}B = \overline{A} + B$



## Esempio di semplificazione algebrica (esercizio)



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

Raccogliendo  $\bar{B}\bar{C}$ :

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

Proprietà dell'inverso: " $\bar{A} + A = 1$ "

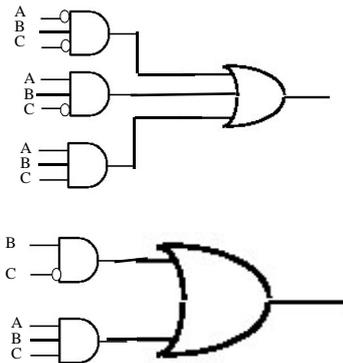
$$= 1\bar{B}\bar{C} + A\bar{B}\bar{C} =$$

Proprietà dell'identità: " $1B = B$ "

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

Dalla slide precedente:

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



## Esempi di manipolazione algebrica



$$F = !xyv + yz + !y!zv + !xy!v + x!yv =$$

$$F = A !B !C + A B C + A B !C + A !B C =$$

$F = A = ?$  Somma di prodotti di 3 variabili: A, B, C (inverso dell'esercizio precedente):



## Esercizi



Usare la sola porta NAND per realizzare AND, OR e NOT e disegnarne gli schemi logici

- Calcolare le TT per le seguenti funzioni  
 $DA + AC + \sim B$   
 $A + B + C + D$   
 $\sim D \sim ABC + \sim DABC + \sim D \sim AB \sim C + \sim DAB \sim C$
- Trasformare in funzioni equivalenti le seguenti  
 $\sim(ABCD)$   
 $\sim(DA) + \sim(B + \sim C)$



## 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.



## Funzione come espressione logica o come tabella delle verità



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

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*

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



## Circuito associato



$$F = 1$$

iif

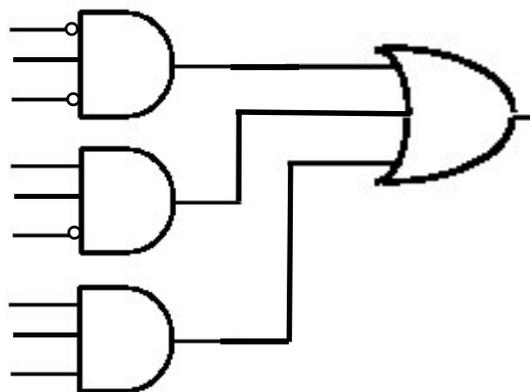
$$A = 0 \ B = 1 \ C = 0$$

*OR*

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

*OR*

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





## La prima forma canonica



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

$$(A \text{ AND } B) =$$

$$F = \bar{A}BC + AB$$

**Implicante:** prodotto delle variabili (in forma asserita o negata) per le quali la funzione vale 1

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) associato agli 1 della funzione.

j indica il numero progressivo in base 10.

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

$$Q \leq 2^N$$

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



## Mintermini e Maxtermini



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

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) associato agli 1 della funzione.

j indica il numero progressivo in base 10.

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

**Maxtermine,  $M_k$ :** un implicante che contiene tutte le N variabili della funzione associato agli 0 della funzione.



## Dall'espressione algebrica alla SOP



- Passare attraverso la tabella della verità
- $F = (\text{NOT}(A) \text{ AND } B \text{ AND NOT}(C)) \text{ OR } (A \text{ AND } B) =$

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

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



## 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:  $Z(A,B,C,D) = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} C \bar{D} + \bar{A} B \bar{C} \bar{D} + \bar{A} B C \bar{D} + \bar{A} \bar{B} \bar{C} D + \bar{A} \bar{B} C D + \bar{A} B \bar{C} D + \bar{A} B C D$

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

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

La stessa espressione si ricaverebbe dalla tabella della verità:

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

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

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



## 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,  $m_i$ :

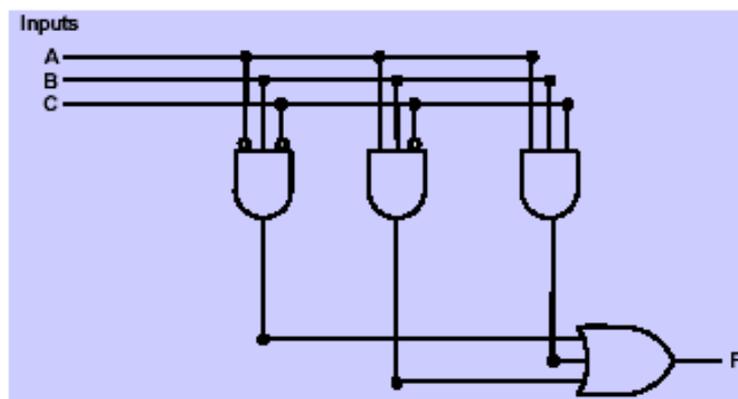
- E' una funzione booleana a n ingressi che vale 1 in corrispondenza della sola  $i$ -esima configurazione di ingresso.
- Al massimo,  $2^n$  mintermini per ogni n variabili.
- ogni mintermine è rappresentabile con un AND con n ingressi.



## Il circuito della prima forma canonica: SOP

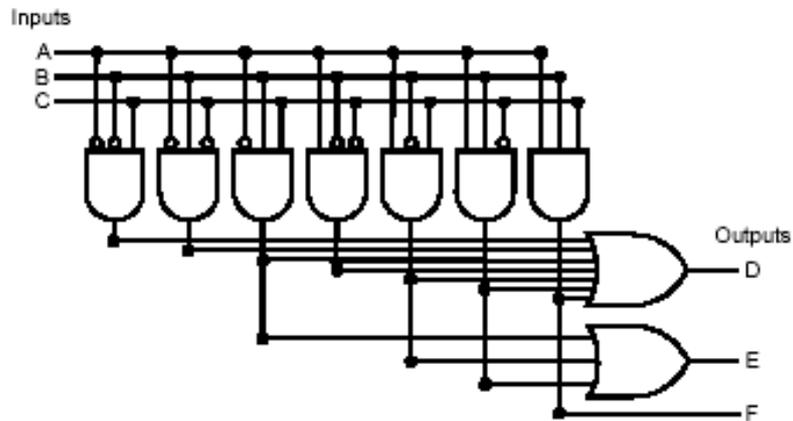


$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$$





## SOP a più uscite



Riutilizzo alcune “parti”, in questo caso alcuni mintermini.

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.**



## 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.



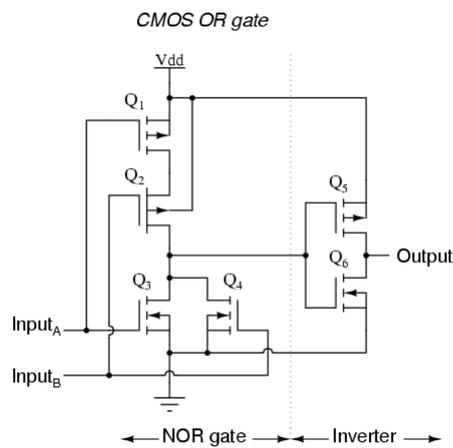
## Valutazione della semplicità di un circuito



Area (numero di porte) = “ampiezza”

Tempo di commutazione (numero di transistor attraversati = “profondità”)

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



A.A. 2015-2016

31/40

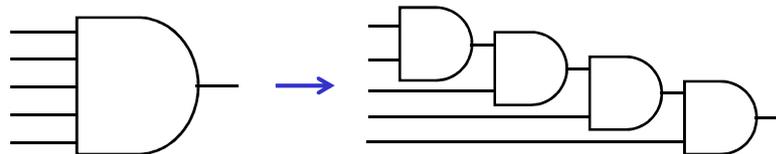
<http://borghese.di.unimi.it/>



## Esempio di trasformazione in un'implementazione con porte a 2 ingressi



- Gli elementi costruttivi di base tipici sono porte a 2 ingressi
  - Porta a N ingressi → N-1 porte a 2 ingressi



Numero di porte:  $N-1 = 4$

Cammino Critico:  $N-1 = 4$



Non è efficiente

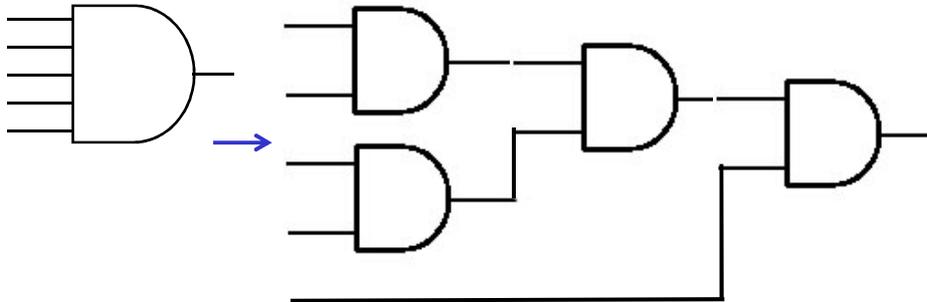
A.A. 2015-2016

32/40

<http://borghese.di.unimi.it/>



## Esempio di trasformazione in un'implementazione efficiente con porte a 2 ingressi



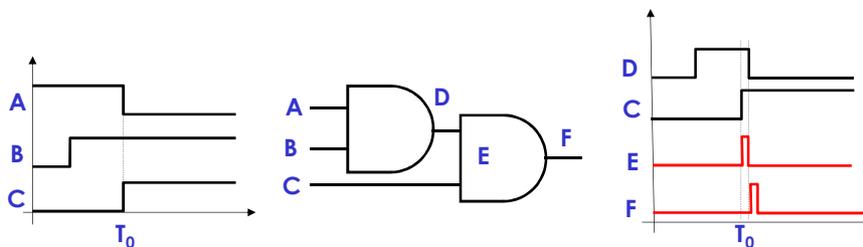
Numero di porte: 4  
Cammino Critico: 3



## 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  $\Delta T$ ).

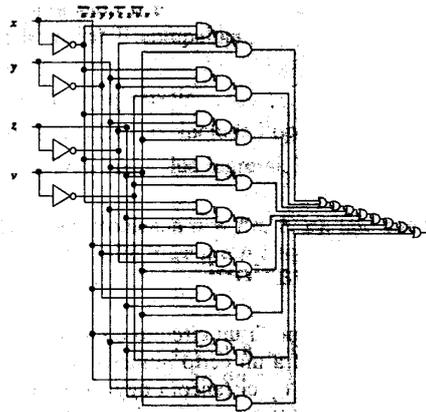
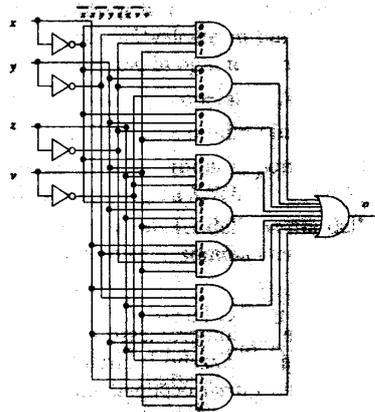


## Riduzione del cammino critico



$$o = x y z v + x y z v + x y z v + x y z v + x y z v + x y z v +$$

$$x y z v + x y z v + x y z v$$



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

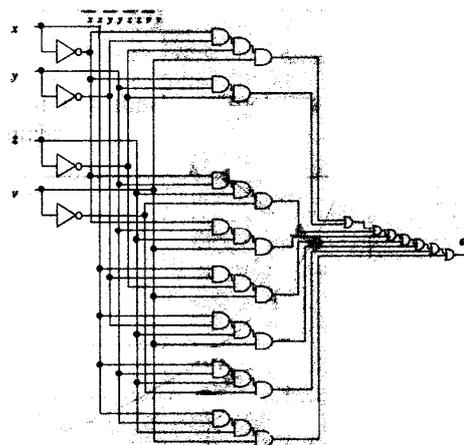
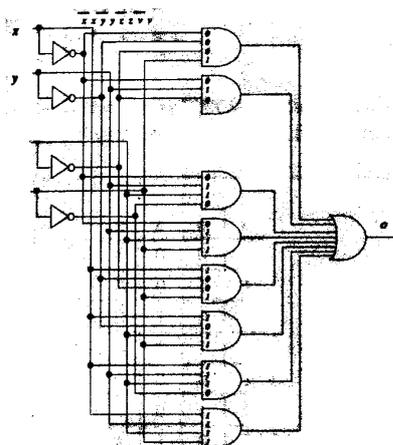


## Semplificazione



2° e 3° AND

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



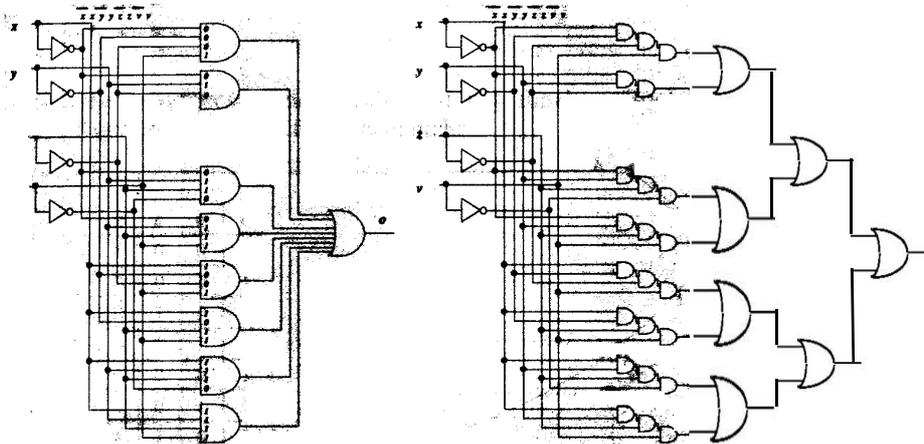
**Cammino critico** pari a 10. Numero di porte pari a 30.



## Ottimizzazione del cammino critico



Riorganizzando gli OR



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



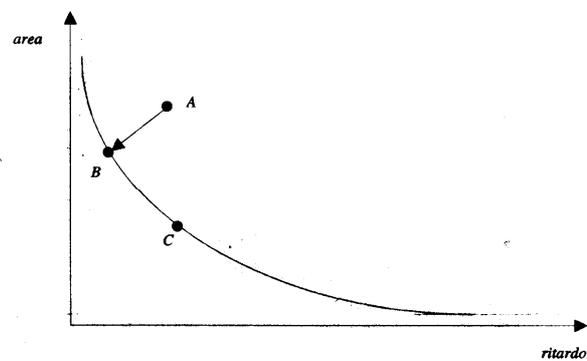
## Criteri di progetto



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.