



## FONDAMENTI DI INFORMATICA

### Lezione n. 4

- **MINIMIZZAZIONE LOGICA**
- **MAPPE DI KARNAUGH**
- **ESERCIZI**

**In questa lezione verranno considerate alcune tecniche di minimizzazione di circuiti logici combinatori.**



## MINIMIZZAZIONE LOGICA

**Per semplificare un circuito occorre:**

- **Esprimere la funzione realizzata dal circuito.**
- **Semplificare la espressione utilizzando le proprietà dell'algebra booleana.**

**Questa procedura non garantisce di ottenere il circuito ottimo perché le proprietà applicate dipendono dall'intuizione del progettista.**

**Esistono, in casi semplici, algoritmi che consentono la progettazione del circuito ottimo per realizzare una funzione data.**

**In generale il problema è molto complesso.**



## MINIMIZZAZIONE LOGICA

La traduzione in circuito delle forme canoniche non genera un progetto ottimo (a minimo costo).

Costo minimo se si minimizza:

- Numero di porte.
- Numero di ingressi (o fan-in) delle porte.

Una somma di prodotti *minima*:

- ha il numero minimo di termini prodotto.
- non è possibile eliminare variabili da alcun termine prodotto.

Una *SdP minima* corrisponde a un circuito a minimo costo: numero minimo di porte con fan-in minimo.



## MINIMIZZAZIONE LOGICA

- Un termine prodotto è un *implicante* se vale 1 per configurazioni di valori delle variabili per cui la funzione non vale 0.
- Un implicante è *principale* quando contiene il minore numero di variabili possibili, cioè eliminando una variabile non è più un implicante.
- Due implicanti si dicono *adiacenti* se differiscono in una variabile.
- Es:

$$f = AB + AC\bar{D}$$

$ABC$  è un implicante,  $A$  non è un implicante.

$AB$  è un implicante principale.

$ABC$  e  $ABC\bar{D}$  sono implicanti adiacenti.



## MINIMIZZAZIONE LOGICA

Per *minimizzazione logica* si intende:

- Il calcolo di tutti gli implicanti principali di una funzione.
- La scelta dell'insieme minimo di implicanti principali la cui somma logica corrisponde alla funzione da minimizzare.

La minimizzazione logica è un problema *intrattabile*:

- Fino a 4 o 5 variabili esistono tecniche manuali.
- Fino a 10 variabili tecniche esatte su calcolatore.
- Al crescere del numero delle variabili vengono utilizzate tecniche euristiche su calcolatore.



## MAPPE DI KARNAUGH

- Permettono di identificare *ad occhio* gli implicanti adiacenti.
- Permettono di selezionare gli implicanti *essenziali*(\*) e l'insieme minimo di implicanti principali.
- Sono utilizzabili con funzioni con un massimo di 4 variabili.

(\*) Un implicante si dice essenziale se è l'unico a coprire un determinato minterm.



## MAPPE DI KARNAUGH

		<i>CD</i>			
<i>AB</i>	00	01	11	10	
00	0	0	0	0	
01	0	0	0	1	
11	0	0	0	1	
10	0	0	0	0	

Ogni casella rappresenta un valore della funzione.

Questa ha sempre valore 0 tranne che per  $[A=0, B=1, C=1, D=0]$  e  $[A=1, B=1, C=1, D=0]$ .

La funzione è descritta da:

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

I due termini sono implicanti adiacenti e gli 1 nella mappa risultano fisicamente adiacenti. Si raggruppano *ad occhio* i due implicanti in uno unico e pertanto:

$$f = B C \bar{D}$$



## MAPPE DI KARNAUGH

		<i>CD</i>			
<i>AB</i>	00	01	11	10	
00	0	0	1	1	
01	0	0	0	0	
11	0	0	0	0	
10	0	0	1	1	

La struttura della mappa è toroidale.

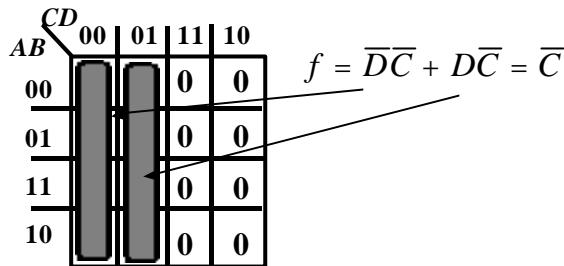
$$f = \bar{A} \bar{B} C + A \bar{B} C$$

$$= \bar{B} C$$

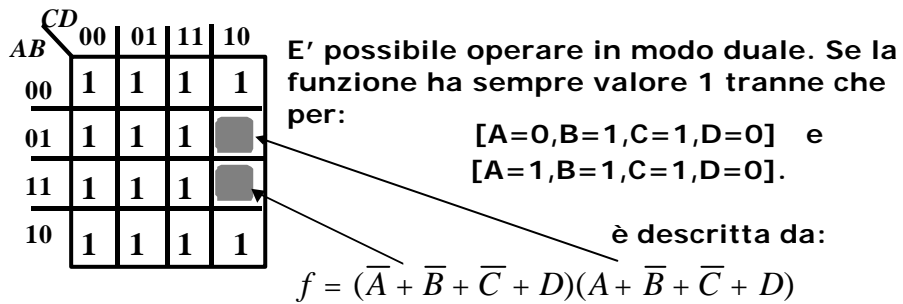
I due termini sono implicanti adiacenti nella visione toroidale della mappa.



### MAPPE DI KARNAUGH



### MAPPE DI KARNAUGH



I due termini sono implicanti (duali) adiacenti e gli 0 risultano fisicamente adiacenti. Si raggruppano *ad occhio* i due implicanti in uno unico e pertanto:

$$f = \overline{B} + \overline{C} + D$$



## ESERCIZIO 1A

Progettare un circuito logico che abbia come ingresso una cifra binaria (4 bit) che rappresenti i numeri da 0 a 9 e fornisca in uscita il numero incrementato di 1.

- Occorre progettare quattro circuiti combinatori a 4 variabili in ingresso.



## ESERCIZIO 1B

INPUT					OUTPUT			
Nm.	X1	X2	X3	X4	Z1	Z2	Z3	Z4
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	0	0	0	0
10	1	0	1	0	d	d	d	d
11	1	0	1	1	d	d	d	d
12	1	1	0	0	d	d	d	d
13	1	1	0	1	d	d	d	d
14	1	1	1	0	d	d	d	d
15	1	1	1	1	d	d	d	d

- La funzione è definita solo per i primi 10 valori.
- Per i dati successivi il valore delle uscite è indifferente.
- Nella tabella si inserisce il valore d (do not care)

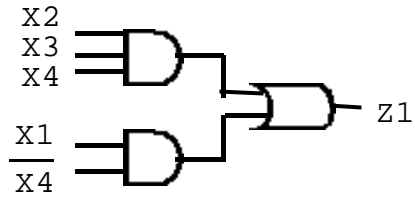


### ESERCIZIO 1C

Mappa per Z1:

		$X_3, X_4$			
		00	01	11	10
$X_1, X_2$	00	0	0	0	0
	01	0	0	1	0
	11	d	d	d	d
	10	1	0	d	d

$$Z1 = X_1 X_4 + X_2 X_3 X_4$$

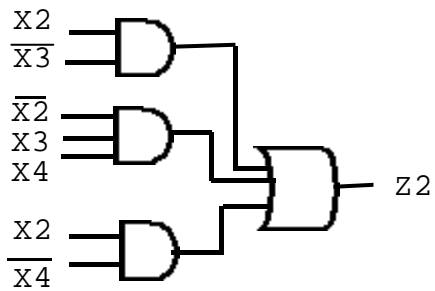


### ESERCIZIO 1D

Mappa per Z2:

		$X_3, X_4$			
		00	01	11	10
$X_1, X_2$	00	0	0	1	0
	01	1	1	0	1
	11	d	d	d	d
	10	0	0	d	d

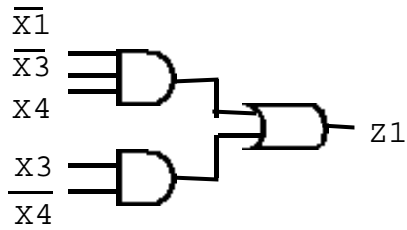
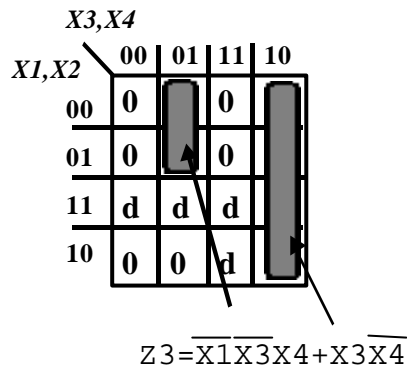
$$Z2 = X_2 X_4 + X_2 X_3 + X_2 X_3 X_4$$





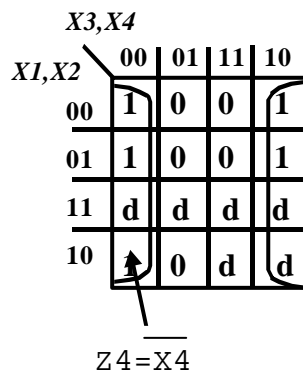
### ESERCIZIO 1E

Mappa per Z3:



### ESERCIZIO 1F

Mappa per Z4:



Il circuito che realizza il sistema è dato dalla sovrapposizione dei circuiti che realizzano le singole uscite.

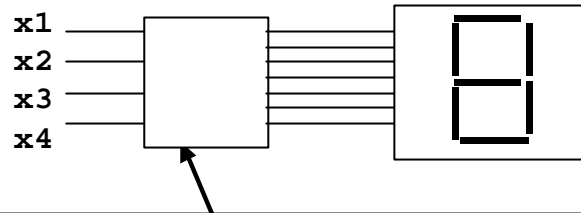
Non è detto però che questa sia la soluzione ottima.

Possono esistere termini comuni alle varie funzioni.



## ESERCIZIO 2A

Progettare un circuito logico che generi i comandi ad un display a 7 segmenti.



Progettare un circuito combinatorio a 4 variabili in ingresso e 7 uscite.

