

## Capitolo 2.3

domenica 24 dicembre 2023 13:22

Home

# CODICE GREY, KARNEAUGH

## CODICE GREY

Il codice grey è un codice utilissimo per ottimizzare le semplificazioni delle formule.

Noi normalmente scriviamo la tabella della verità così

0	0
0	1
1	0
1	1

Con il codice grey invece arriviamo alla metà e la invertiamo ovvero...

00  
01  
11  
10

Un **letterale** si dice se c'è una lettera che è positiva o negativa.

Un **implicante** quando hai un and tra più letterali.

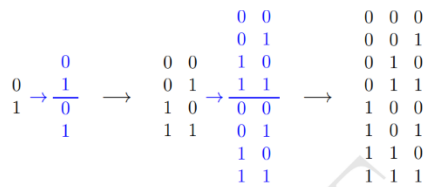
**Mintermine** è quando abbiamo un implicante tra tutte le lettere possibili

$$(x_1 \overline{x_2} x_3) + (x_1 x_2)$$

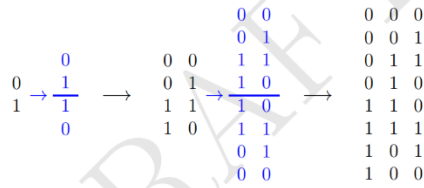
Ecco lo screen della lezione del pasqualone sul codice grey

Per evidenziare direttamente dalla tabella quali semplificazioni si possono fare, sarebbe utile scrivere le righe con un ordinamento in cui ogni riga differisca dalla precedente e dalla successiva per un unico bit. È possibile farlo? Vediamo.

L'ordinamento usuale con cui scriviamo le  $2^n$  righe di una tabella con  $n$  variabili si può generare ricorsivamente in questo modo: per costruire le righe della tabella con  $n+1$  variabili, partiamo dalla righe della tabella con  $n$  variabili, la "duplichiamo" e aggiungiamo 0 davanti a tutte le righe dell'"originale" e 1 davanti a tutte le righe della "copia".



Se nella costruzione precedente, quando “duplichiamo” la sequenza, la “ribaltiamo” anche, otteniamo un ordinamento chiamato *codice Gray*.



**Esercizio 3.** Osservare che, per costruzione, in un codice Gray a  $n$  bit

- Ogni sequenza differisce dalla successiva per un unico bit;
- L'ultima sequenza differisce dalla prima per un unico bit.

# Karnaugh

Serve per semplificare le formule al toppingz.

Ti scrivi una tabella tipo

così  $x_0 x_1 \overline{x_2} + x_0 x_1 x_2$

Si potrebbe semplificare con l'algebra booleana però la facciamo con

karnaugh

$x_0$	$x_1$	$x_2$	$y$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array}$$

$x_2$		0	1
$x_0$	$x_1$		
0	0	0	0
0	1	0	0
1	1	1	1
1	0	0	0

Raccogliamo gli uni in  
potenza di 2 alla n quindi  
in 1 in 2 in 4 in 8 ecc...

Vediamo dove differisce il  
bit, quello che differisce  
non lo scriviamo,  
In questo caso  $x_0$  e  $x_1$  non  
cambiano e li lasciamo

invece  $x_2$  cambia tra 0 e 1  
e non lo consideriamo  
perchè si annulla con sé  
stesso

$x_0 x_1$

La formula semplificata  
sarà questa e ora  
possiamo passare al  
circuito

## CNF E DNF

Formula normale

(Congiuntiva e disgiuntiva)

1. **Prodotti di somma(0)**
2. **somma di prodotti(1)**

