



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,
Informatiche e Matematiche

6. Sintesi di Reti Logiche Combinatorie

Architettura dei calcolatori [MN1-1143]

Corso di Laurea in INFORMATICA
(D.M.270/04) [16-215]
Anno accademico 2022/2023

Prof. Alessandro Capotondi
a.capotondi@unimore.it

È vietata la copia e la riproduzione dei contenuti e immagini in qualsiasi forma.

È inoltre vietata la redistribuzione e la pubblicazione dei contenuti e immagini non autorizzata espressamente dall'autore o dall'Università di Modena e Reggio Emilia.

Forma canonica

- La più immediata forma di rappresentazione delle funzioni Booleane (e delle tabelle di verità) è la rappresentazione con una espressione in FORMA CANONICA

FORMA CANONICA SP (SOMMA DI PRODOTTI)

Teorema: una funzione di n variabili può essere espressa in un solo modo come somme di prodotti di n variabili (chiamati MINTERMINI)

- MINTERMINE** è il prodotto logico di n letterali ognuno dei quali compare in forma vera o complementata, ma mai in entrambe.
- Da ogni tabella si deriva deterministicamente la forma SP, prendendo in **OR** tutti i mintermini corrispondenti alle righe in cui l'uscita vale **1**, in cui ogni variabile è in forma **diretta** se nella colonna appare il valore **1** ed in forma **complementata** se appare il valore **0**.

r	a	b	S	R	
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	r'ab
1	0	0	1	0	
1	0	1	0	1	ra'b
1	1	0	0	1	rab'
1	1	1	1	1	rab

$$R = r'ab + ra'b + rab' + rab$$

Forma canonica

- La forma canonica può essere ottenuta per qualsiasi rete logica combinatoria
- Indipendentemente dalla complessità della rete logica da realizzare, la rete logica ottenuta dalla forma canonica è una rete molto veloce, in quanto composta da soli due livelli e mezzo (livello dei **not**)

Forma canonica PS

Usando il teorema di De Morgan si può provare l'esistenza di un'altra forma canonica

FORMA CANONICA PS (PRODOTTO DI SOMME)

Teor: una funzione di n variabili può essere espressa in un solo modo come prodotto di somme di n variabili (chiamate MAXTERMINI)

- **MAXTERMINE** è la somma logica di n letterali ognuno dei quali compare in forma vera o complementata, ma mai in entrambe
- Da ogni tabella si deriva deterministicamente la forma PS, prendendo in **AND** tutti i **maxtermini** corrispondenti alle righe in cui l'uscita vale **0**, in cui ogni variabile è in forma **diretta** se nella colonna appare il valore **0** ed in forma **complementata** se appare il valore **1**.

- Dalla tabella precedente:

$$R = (r+a+b)(r+a+b')(r+a'+b)(r'+a+b)$$

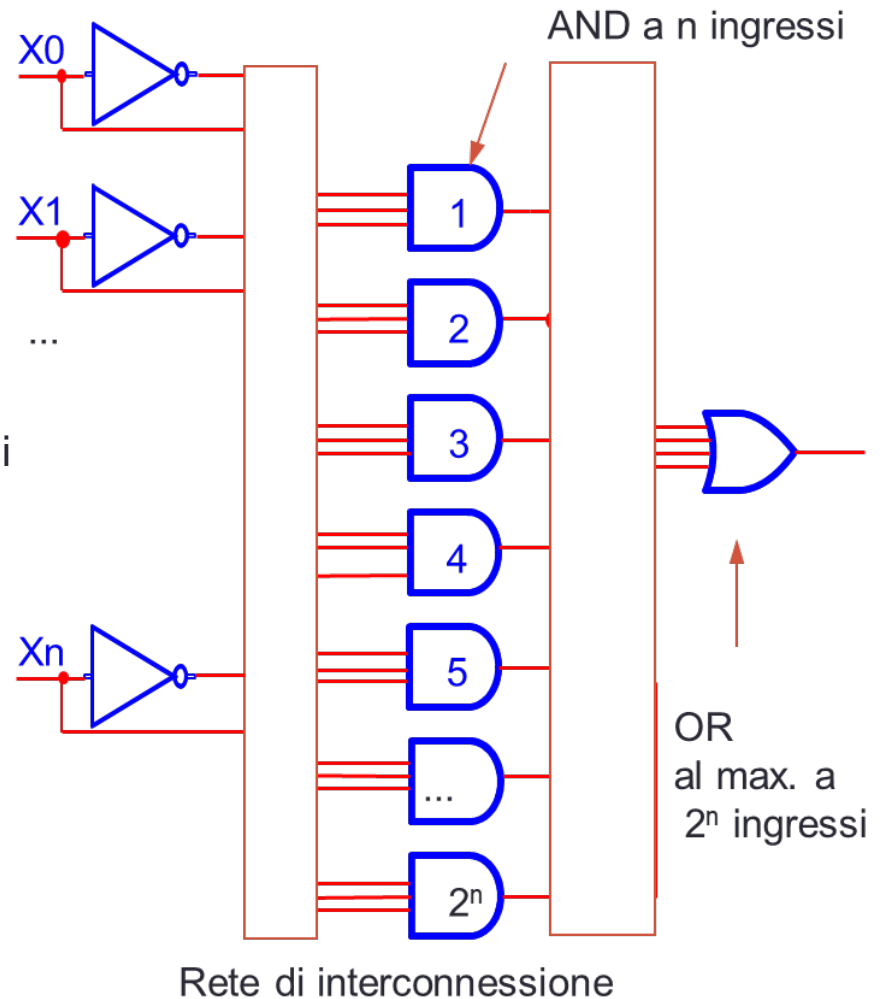
r	a	b	S	R	
0	0	0	0	0	(r+a+b)
0	0	1	1	0	(r+a+b')
0	1	0	1	0	(r+a'+b)
0	1	1	0	1	
1	0	0	1	0	(r'+a+b)
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	

- Si dimostra essere equivalente alle precedente

Sintesi di forme canoniche

- **SOMME DI PRODOTTI**

- una **funzione combinatoria** di n ingressi sintetizzata in forma canonica contiene al più n gate NOT, 2^n AND al più a n ingressi e 1 OR al più a 2^n ingressi
- Similmente per la forma canonica PS (scambiando OR e AND)



Esempio: codice Gray

Esercizio : *Progettare la rete logica di conversione di codice binario in codice GRAY a 3 ingressi*

Il **codice Gray** e' un codice a distanza di ***Hamming**** unitaria

BINARIO	GRAY
b2,b1,b0	g2,g1,g0
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

*) Hamming, Richard W. "Error detecting and error correcting codes." The Bell system technical journal 29.2 (1950): 147-160.

Esempio: codice Gray

Esercizio : *Progettare la rete logica di conversione di codice binario in codice GRAY a 3 ingressi*

Il **codice Gray** e' un codice a distanza di **Hamming*** unitaria

BINARIO	GRAY
b2,b1,b0	g2,g1,g0
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

Per la sintesi di g0

4 mintermini

$b_2'b_1'b_0$

$b_2'b_1b_0'$

$b_2b_1'b_0$

$b_2b_1b_0'$

$$g_0 = b_2'b_1'b_0 + b_2'b_1b_0' + b_2b_1'b_0 + b_2b_1b_0'$$

Esempio: codice Gray

Esercizio : *Progettare la rete logica di conversione di codice binario in codice GRAY a 3 ingressi*

Il **codice Gray** e' un codice a distanza di ***Hamming**** unitaria

BINARIO	GRAY
b2,b1,b0	g2,g1,g0
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

$$g_0 = b_2'b_1'b_0 + b_2'b_1b_0' + b_2b_1'b_0 + b_2b_1b_0'$$

$$g_1 = b_2'b_1b_0' + b_2'b_1b_0 + b_2b_1'b_0' + b_2b_1'b_0$$

$$g_2 = b_2b_1'b_0' + b_2b_1'b_0 + b_2b_1b_0' + b_2b_1b_0$$

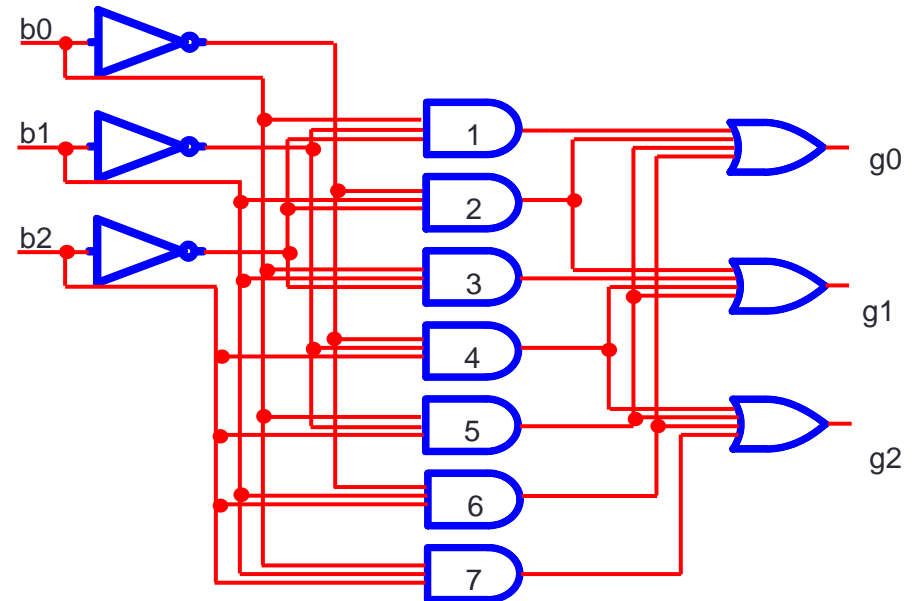
Esempio: codice Gray

Esercizio : Progettare la rete logica di conversione di codice binario in codice GRAY a 3 ingressi

Il **codice Gray** e' un codice a distanza di Hamming unitaria

BINARIO	GRAY
b2,b1,b0	g2,g1,g0
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

$$\begin{aligned}g_0 &= b_2'b_1'b_0 + b_2'b_1b_0' + b_2b_1'b_0 + b_2b_1b_0' \\g_1 &= b_2'b_1b_0' + b_2'b_1b_0 + b_2b_1'b_0' + b_2b_1'b_0 \\g_2 &= b_2b_1'b_0' + b_2b_1'b_0 + b_2b_1b_0' + b_2b_1b_0\end{aligned}$$



Funzioni non completamente specificate

- **Funzioni non completamente specificate** se le uscite hanno condizioni di INDIFFERENZA
- Esistono alcune delle 2^n configurazioni non definite: il dominio è un sottoinsieme del dominio delle 2^n configurazioni
- Una espressione definisce una funzione non completamente specificata solo limitatamente al suo dominio.
- Le espressioni canoniche SP o PS di una funzione non completamente specificata **NON SONO UNICHE**
- Gli schemi logici che rappresentano la struttura delle reti logiche devono essere completamente specificate dal progettista.

Funzioni non completamente specificate

- **Esempio:**

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	-

Forme canoniche SP equivalenti

$$Z1 = A'B + AB' \quad Z2 = A'B + AB' + AB$$

Z2 si semplifica:

per idempotenza

per distributiva

per complementarietà

per elementi nulli e commutativa

$$Z2 = A'B + AB + AB + AB'$$

$$Z2 = (A' + A)B + A(B + B')$$

$$Z2 = 1B + 1A$$

$$Z2 = A + B$$

Sintesi e minimizzazione

Sintesi di reti logiche combinatorie:

- 1) descrizione mediante tabella della verità
- 2) sintesi della espressione canonica SP o PS
- 3) corrispondenza 1 a 1 con uno schema logico

Tale sintesi non è minimizzata: può esserlo in termini di:

- area minima, costo minimo
- minimo n. di gate
- minimo n. di livelli
- minimo n. di interconnessioni
- fan in e fan out limitato

Sintesi e minimizzazione

Normalmente una rete logica si dice in forma minima per indicare il minor numero di livelli e, a parità di livelli, il minor numero di gate e di ingressi dei gate

Tecniche di minimizzazione:

- minimizzazione con manipolazione algebrica
- minimizzazione manuale (k-mappe)
- minimizzazione con algoritmi CAD o software appositi (es. Logisim)

Mappe

- **Mappa:** Rappresentazione più compatta della tabella di verità
- E' una rappresentazione matriciale della tabella in cui le righe indicano tutte le possibili configurazioni di un sottoinsieme delle variabili di ingresso e le colonne tutte le configurazioni delle variabili rimanenti, il valore nelle celle indica il valore dell'uscita nella configurazione corrispondente

x_3x_2		x_1x_0			
		00	01	10	11
00		1	0	1	-
01		0	1	1	-
10		1	1	1	-
11		1	1	-	-

- **OGNI CELLA CORRISPONDE AD UNA CONFIGURAZIONE DELLE VARIABILI**

Mappe di Karnaugh: Mappe in cui le configurazioni successive in ogni lato sono ADIACENTI

- due configurazioni sono adiacenti (*logicamente*) se differiscono di un solo bit
- due celle sono adiacenti (*geometricamente*) se corrispondono a configurazioni adiacenti
- **Nelle Mappe di Karnaugh adiacenza geometrica e logica COINCIDONO**



Mappe di Karnaugh

K-mappe a 2 variabili

		x0	
		0	1
x1	0	0	0
	1	1	1

K-mappe a 3 variabili

		x1,x0			
		00	01	11	10
x2	0	0	0	-	1
	1	1	1	1	1

K-mappe a 4 variabili

		x1x0			
		00	01	11	10
x3x2	00	1	0	-	1
	01	0	1	-	1
	11	1	1	-	1
	10	1	1	-	-

Criteri geometrici di adiacenza:

- 1 lato in comune
- un'estremità di colonna
- un'estremità di riga
- stessa posizione in sottomatrici adiacenti

K-mappe a 5 variabili

		x1x0			
		00	01	11	10
x3x2	00	1	0	-	1
	01	0	1	-	1
	11	1	1	-	1
	10	1	1	-	-

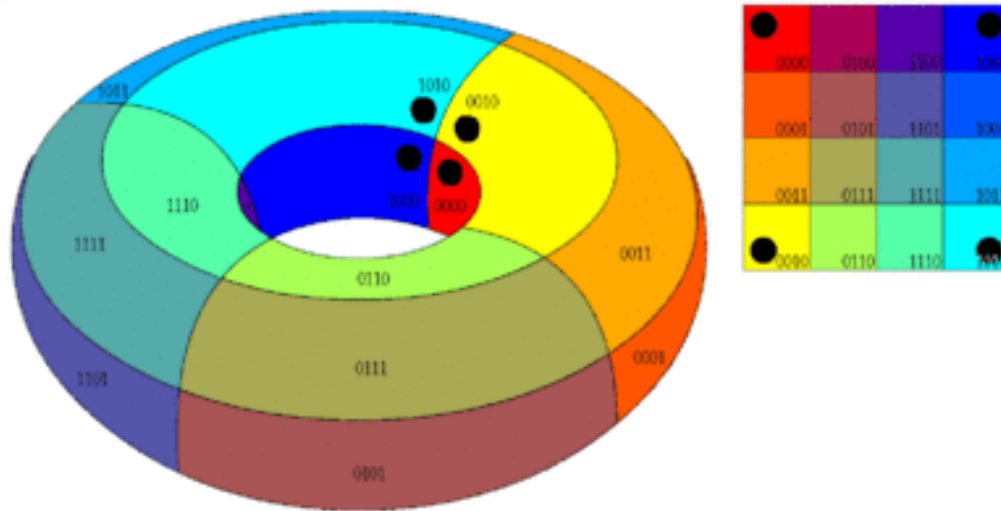
X4=0

		x1x0			
		00	01	11	10
x3x2	00	1	0	-	1
	01	0	1	0	1
	11	1	0	1	1
	10	1	1	1	-

X4=1

Visualizzazione 3D delle mappe

- Le mappe vanno viste come «arrotolate» su se stesse.
- La prima riga risulta «adiacente» all'ultima riga. Stessa cosa per le colonne.
- Una visualizzazione 3d delle mappe che mette in risalto tale adiacenza è quella rappresentata in figura

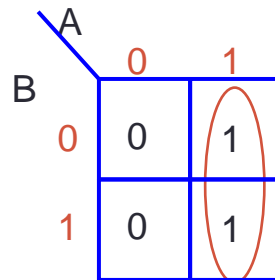


Minimizzazione con mappe di Karnaugh

- ogni casella della mappa è **adiacente** a caselle corrispondenti a mintermini (maxtermini) aventi distanza di Hamming unitaria dal mintermine (maxtermine) corrispondente alla casella considerata.

$$F = AB' + AB = A(B+B') = A \text{ proprietà distributiva}$$

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



la funzione vale 1 quando A vale 1
indipendentemente dal valore di B

Minimizzazione con mappe di Karnaugh

- Nelle mappe due mintermini (o maxtermini) di distanza 1 sono adiacenti

		BC			
		00	01	11	10
A	0	1	1	1	0
	1	0	1	0	0

$$Z = A'B'C' + A'B'C + A'BC + AB'C$$

idempotenza

$$\begin{aligned} Z &= A'B'C' + A'B'C + \\ &\quad A'BC + A'B'C + \\ &\quad AB'C + A'B'C \end{aligned}$$

$$\begin{aligned} Z &= A'B'(C'+C) + \\ &\quad A'C(B'+B) + \\ &\quad B'C(A'+A) \end{aligned}$$

$$Z = A'B' + A'C + B'C$$

Raggruppamenti rettangolari (1/2)

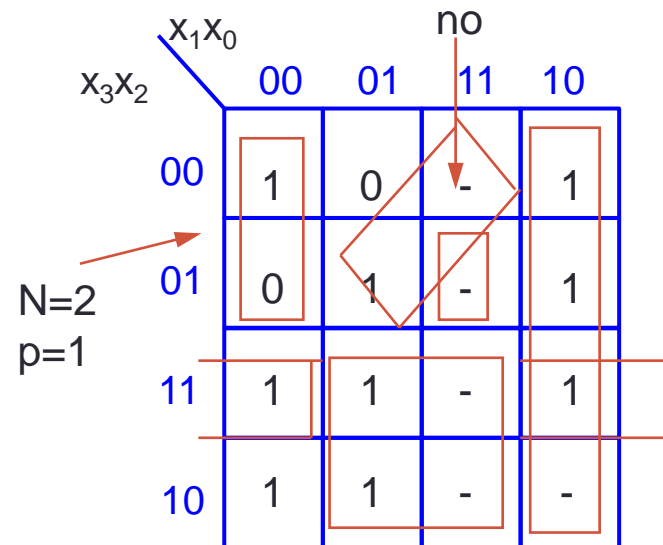
- Si dice **raggruppamento rettangolare di ordine p** una parte di una mappa a n variabili costituita da 2^p elementi (con $p \leq n$) tali da avere $n-p$ coordinate uguali fra loro, e di far assumere alle restanti p coordinate tutte le possibili configurazioni.

- Ogni cella ha all'interno p celle adiacenti

ordine 0 1 cella
ordine 1 2 celle
ordine 2 4 celle

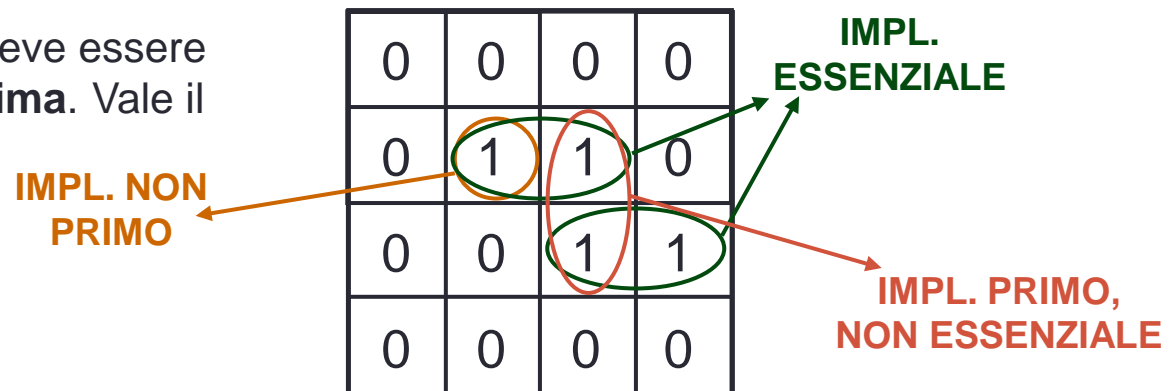
- Un Raggruppamento Rettangolare (RR) nel quale la funzione assume sempre valore 1 si dice **implicante** della funzione. In modo duale, un RR nel quale la funzione assume sempre valore 0 si dice **implicato** della funzione.

- Un **implicante** (*implicato*) corrisponde a un **prodotto logico** (somma logica) dei letterali *delle sole variabili di ingresso che non cambiano valore*, presi negati se la corrispondente variabile di ingresso vale 0 (1), non negati se tale variabile vale 1 (0).



Raggruppamenti rettangolari (2/2)

- Un implicante non contenuto in nessun implicante di dimensioni maggiori prende il nome di **implicante primo**.
- Si dice **copertura degli 1** un insieme di implicanti che contengono tutti gli 1 della funzione ed eventualmente indifferenze
(**copertura di 0** un insieme di implicati che contenga tutti gli 0 e al più indifferenze)
- **implicanti essenziale**: un implicante primo contenente almeno un mintermine non contenuto in nessun altro implicante primo (cioè un implicante primo che “copra” almeno un mintermine non coperto da altri).
- Ogni implicante essenziale deve essere contenuto nella **somma minima**. Vale il duale per gli implicati



Copertura e forma canonica

- **Una copertura di 1** individua una forma SP

- $Z = A'BC'D + A'BCD +$

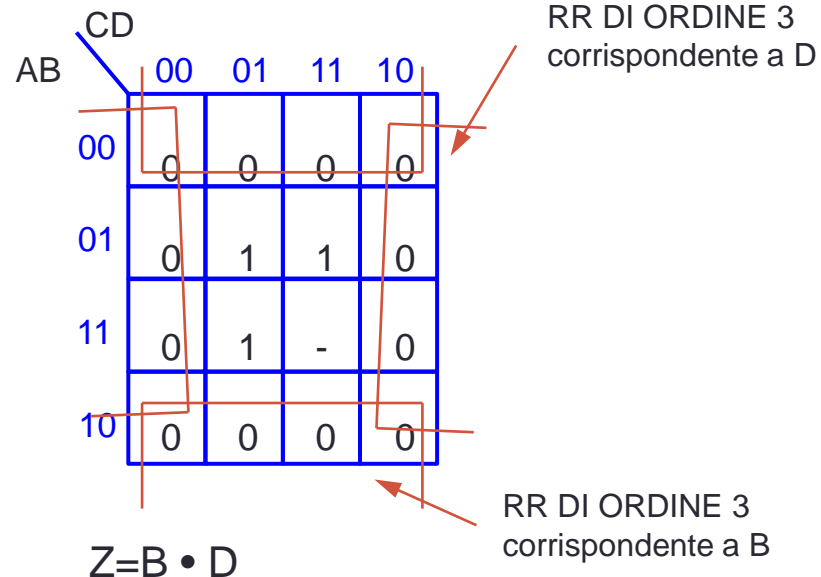
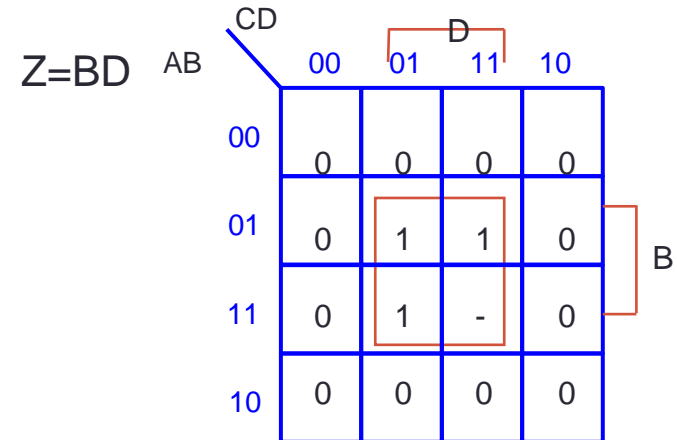
$$ABC'D + ABCD$$

$$Z = A'BD(C+C') + ABD(C+C')$$

$$Z = (A'+A)BD$$

$$Z = BD$$

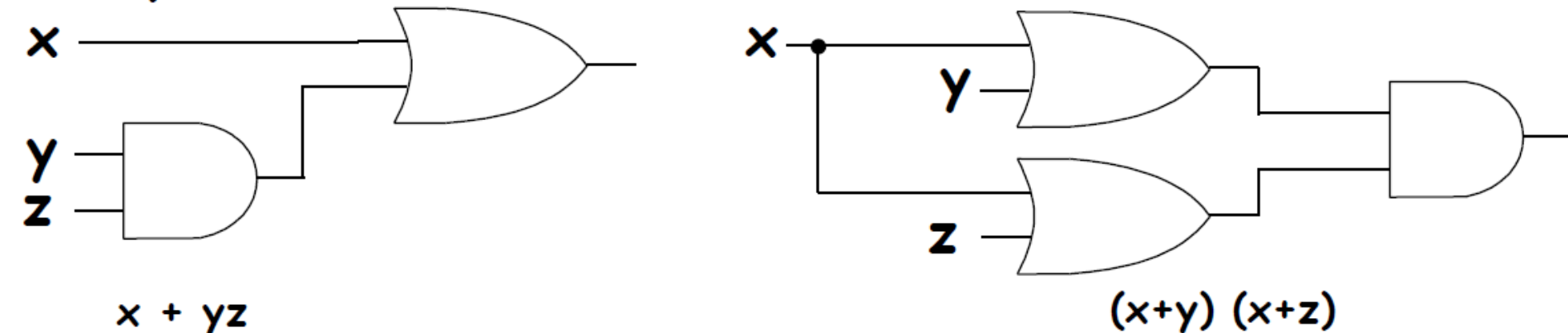
- **Una copertura di 0** individua una forma PS



Complessità - Velocità

- Per valutare la complessità di una rete logica in termini di complessità e velocità si utilizzano 3 indicatori :
 - N_{gate} = numero di gate,
 - N_{conn} = numero di connessioni
 - N_{casc} = max numero di gate disposti in cascata
- **Complessità** \Rightarrow funzione crescente di N_{gate} , N_{conn}
- **Velocità di elaborazione** \Rightarrow funzione decrescente di N_{casc}

Esempio:



- Le due reti sono equivalenti (prop. distrib.).
- Hanno la stessa velocità di elaborazione.
- La rete di sinistra è meno complessa.

Forme normali e minime (1/2)

Una espressione

- si dice **normale SP** se è data dalla somma di prodotti non necessariamente di n variabili
- si dice **normale PS** se è data dal prodotto di somme non necessariamente di n variabili

Una espressione normale è equivalente alla forma canonica ma minimizzata

SINTESI MINIMA (di costo minimo)

- con il minor numero di livelli
- minimo numero di gate (ad es. di prodotti da sommare) --- forma normale
- minimo numero di connessioni
- **l'espressione minima normale** e non ridondante si ottiene con una **copertura** usando il **numero minimo di RR di ordine massimo** (implicanti primi)
 - **ordine massimo** : minor numero di ingressi
 - **minimo numero di RR**: minimo numero di gate
 - forma normale **irridondante**: solo implicanti essenziali

Forme minime (2/2)

Forma minima PS ed SP

- sono diverse
- Potrebbe valer la pena valutarle entrambe specialmente con indifferenze

Reti a più uscite

- a volte è conveniente scegliere degli implicant comuni anche se non primi o essenziali

		x_3x_2			
		00	01	11	10
x_1x_0	00	0	0	1	1
	01	0	1	1	0
	11	1	1	0	0
	10	1	1	0	0

$$Z = x_3x_1'x_0' + x_3x_2x_1' + x_2x_1'x_0 + x_3'x_2x_0 + x_3'x_1$$

		x_3x_2			
		00	01	11	10
x_1x_0	00	0	0	1	1
	01	0	1	1	0
	11	1	1	0	0
	10	1	1	0	0

$$Z = x_3x_1'x_0' + x_2x_1'x_0 + x_3'x_1$$

Sintesi di reti combinatorie complesse

La mappa di Karnaugh si può usare come metodo manuale perfino a 5 o 6 variabili

- Può essere impiegata efficientemente per sfruttare le indifferenze

Esistono tecniche ed algoritmi per la sintesi automatica a più livelli

1. *Mediante manipolazione algebrica, ad esempio usando sistematicamente la proprietà distributiva*
2. *Algoritmi di sintesi logica (e.g., Quine Mc Cluskey)*
3. *CAD tools (Berkeley Octtools Package, MisII, Espresso logic minimizer)*
4. *Metodi empirici*

Perchè minimizzare?

- Perche' le forme canoniche richiedono un fan in troppo alto, troppi gate (consumo di area)

Evaluation (it is your moment)

- Oppure usa il QR code

Collegati

<https://menti.com>

Inserisci il codice

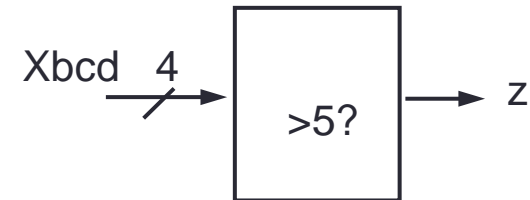
8489 1658



<https://www.menti.com/al6irk6p9qua>

Esercizi (1/3)

Esercizio 1: Dato un numero in BCD progettare la rete logica che indichi se è maggiore di 5; quante sono le forme canoniche equivalenti? Perché sono più di 1? Quale è la forma minima PS quale la SP?

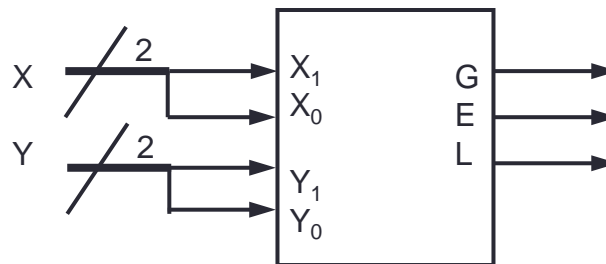


Esercizio 2: Progettare un comparatore a 2 bit, ossia una rete logica che indichi quale dei due operandi a 2 bit è maggiore uguale o minore dell'altro.

Descrizione a parole:

```
if (X>Y) {G=1,E=0,L=0;}
else if (X<Y){G=0,E=0,L=1;}
else {G=0,E=1,L=0;}
```

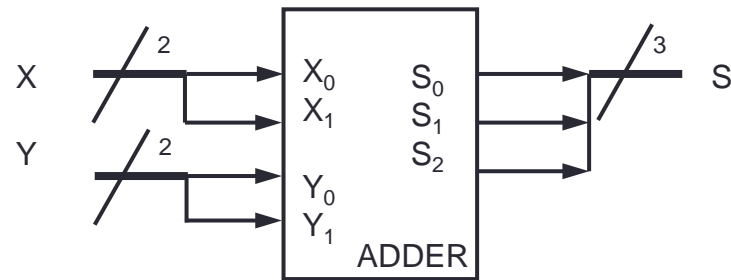
- 1) tabella della verità
- 2) mappe di Karnaugh
- 3) minimizzazione



Esercizi (2/3)

Esercizio 3: progettare il FULL ADDER ossia il sommatore ad 1 bit con riporto di ingresso e di uscita (nella forma SP).

Esercizio 4: Progettare un sommatore a 2 bit di ingresso e 3 di uscita



Esercizio 5: Modificare l'esercizio 4 con anche il riporto di ingresso. Che differenza c'è in termini di gate e di ritardo rispetto a quello dell'esercizio 3? Come progettare un sommatore modulare usando un FULL ADDER?

Esercizio 6: trovare la forma canonica e minima SP e PS

		AB			
		00	01	11	10
CD	00	0	0	1	1
	01	0	1	1	1
	11	0	1	1	1
	10	-	1	1	1

Esercizi (3/3)

Esercizio 7: Una lampadina può essere accesa o spenta da 3 interruttori X_1 , X_2 e X_3 : però viene accesa solo se sono ON un numero dispari di interruttori, oppure se X_2 e X_3 sono contemporaneamente ON. Progettare la rete logica corrispondente.

Esercizio 8: Tre interruzioni possono arrivare alla CPU anche contemporaneamente ma devono essere servite con una priorità: IR_1 , IR_2 e IR_3 in ordine di priorità decrescente, nel senso che IR_1 è la più prioritaria. Progettare la rete che dà la richiesta di interruzione INT alla CPU e abilita le tre interruzioni IS_1 , IS_2 , IS_3 .