Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2023-2024
Università degli Studi di Milano

SOP, POS, cammino critico
Semplificazione: mappe di Karnaugh

Forme canoniche e cammino critico

- Prima forma canonica (SOP: Sum Of Products)

$$\sum_{i=1}^{Q} m_i,\ Q \leq 2^n$$

- m_i mintermine
- Q numero di mintermini

- Seconda forma canonica (POS: Product Of Sums)

$$\prod_{i=1}^{W} M_i,\ W \leq 2^n$$

- M_i maxtermine
- W numero di maxtermini

- Cammino critico: massimo numero di porte <u>a due ingressi</u> da attraversare (escluso quindi l'inverter) da un qualsiasi ingresso a una qualsiasi uscita

Esercizio 1

1. Si ricavi la SOP per la porta XNOR e si simuli in Logisim il circuito equivalente

2. Se ne derivi il cammino critico

3. Si dica se ricavandone la POS cambia il cammino critico

Esercizio 1

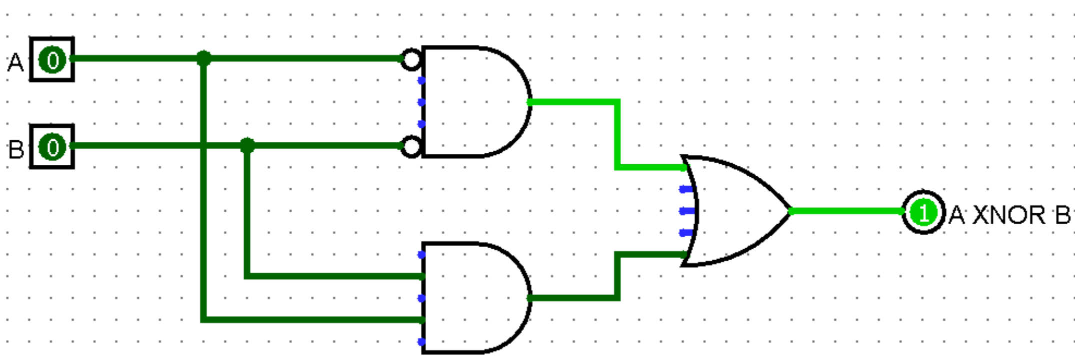
Tabella di verità

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Mintermini $(\neg A \neg B), (AB)$

SOP $(\neg A \neg B) + (AB)$

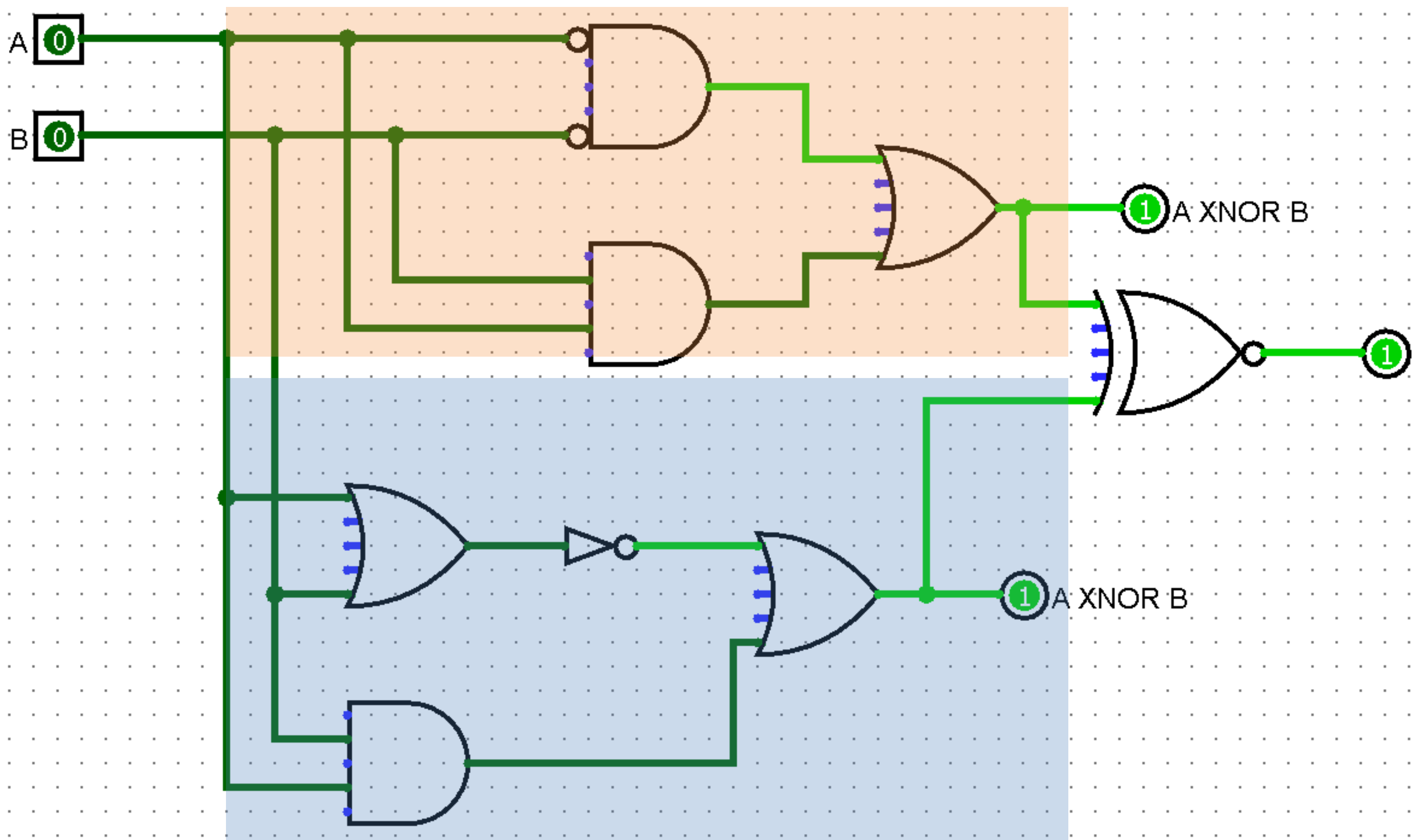
Circuito in Logisim



Il **cammino critico** è pari a 2

Esercizio 1

SOP Porta XNOR



XNOR - esercizio 3, laboratorio 2

Ci sono diversi modi per implementare la stessa espressione logica

Esercizio 1

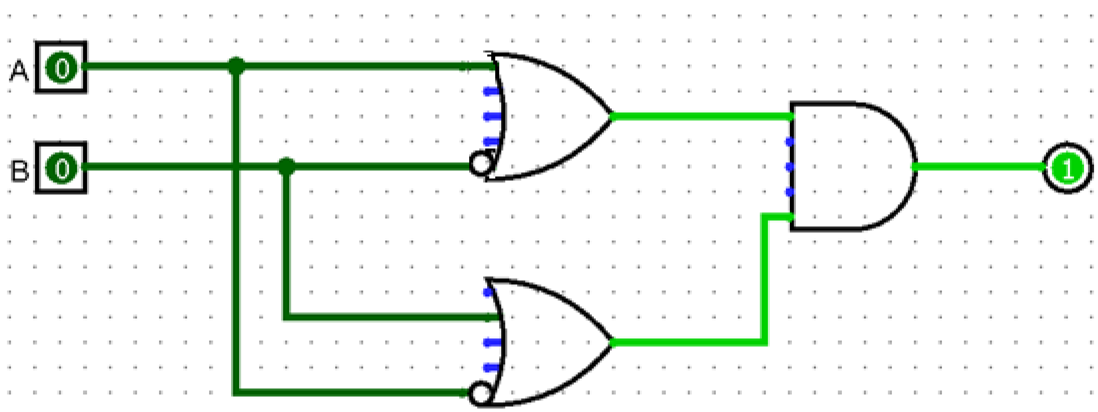
Tabella di verità

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Maxtermini $(A + \neg B), (\neg A + B)$

POS $(A + \neg B)(\neg A + B)$

Circuito in Logisim



Il **cammino critico** anche in questo caso è pari a 2

Esercizio 2

Sia data la seguente espressione logica

$$X = A(A + \neg B)(B + C) + \neg BD$$

1. Si derivi la tabella di verità (si indichino anche alcune sotto-espressioni)

2. Si derivi la SOP

3. Si implementino in Logisim il circuito associato alla formula originale ed il circuito associato alla SOP e li si confrontino

4. Si proceda poi alla semplificazione algebrica della SOP, si implementi il circuito corrispondente e lo si confronti con gli altri due circuiti implementati

Esercizio 2

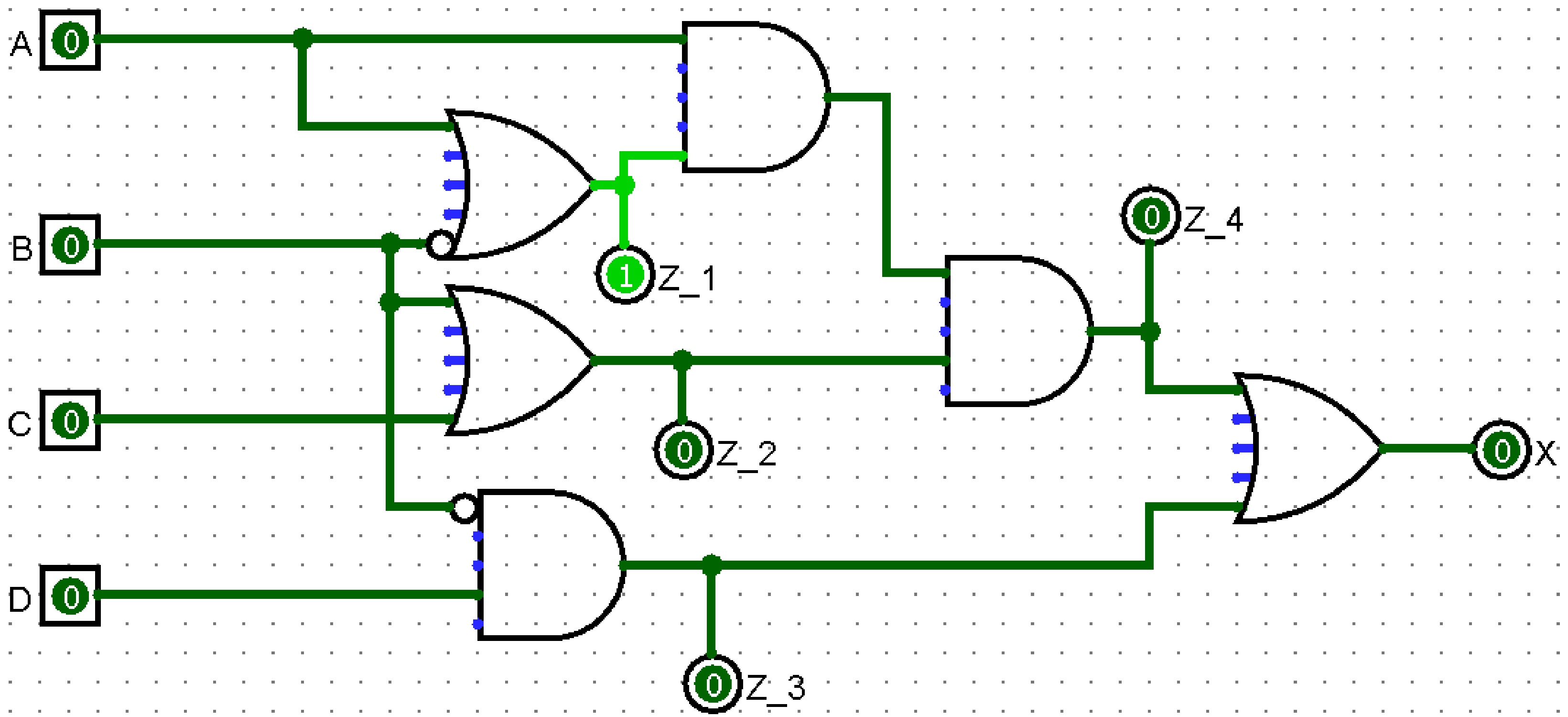
Tabella di verità

A	B	C	D	$\neg B$	$Z_1 = A + \neg B$	$Z_2 = B + C$	$Z_3 = \neg BD$	$Z_4 = AZ_1Z_2$	$X = Z_4 + Z_3$
0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	1	0	1
0	0	1	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	1
0	1	0	0	0	0	1	0	0	0
0	1	0	1	0	0	1	0	0	0
0	1	1	0	0	0	1	0	0	0
0	1	1	1	0	0	1	0	0	0
1	0	0	0	1	1	0	0	0	0
1	0	0	1	1	1	0	1	0	1
1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	0	1	1
1	1	0	1	0	1	1	0	1	1
1	1	1	0	0	1	1	0	1	1
1	1	1	1	0	1	1	0	1	1

$$X = (\neg A \neg B \neg C D) + (\neg A \neg B C D) + (A \neg B \neg C D) + (A \neg B C \neg D) + (A \neg B C D) + (A B \neg C \neg D) + (A B \neg C D) + (A B C \neg D) + (A B C D)$$

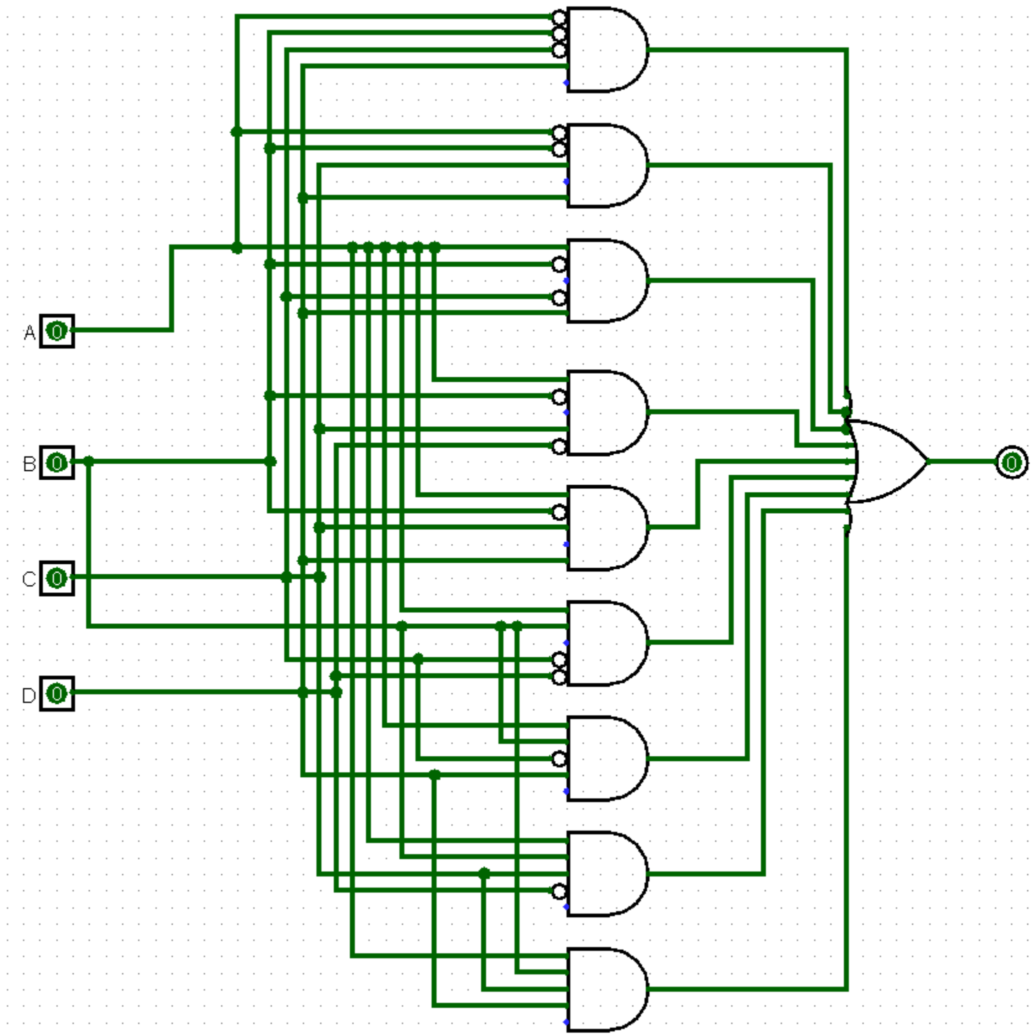
Esercizio 2

Circuito originale



Esercizio 2

Circuito SOP



$$X = (\neg A \neg B \neg C D) + (\neg A \neg B C D) + (A \neg B \neg C D) + (A \neg B C \neg D) + (A \neg B C D) + (A B \neg C \neg D) + (A B \neg C D) + (A B C \neg D) + (A B C D)$$

Esercizio 2

$$
\begin{aligned}
X = \quad & (\neg A \neg B \neg C D) + (\neg A \neg B C D) + (A \neg B \neg C D) + \\
& (A \neg B C \neg D) + (A \neg B C D) + (A B \neg C \neg D) + \\
& (A B \neg C D) + (A B C \neg D) + (A B C D) \\
= \quad & (\neg B \neg C D)(\neg A + A) + (\neg B C D)(A + \neg A) + & \text{(Raccoglimento)} \\
& (A C \neg D)(\neg B + B) + (A B \neg C)(D + \neg D) + \\
& (A B C D) \\
= \quad & (\neg B \neg C D) + (\neg B C D) + & \text{(Inverso)} \\
& (A C \neg D) + (A B \neg C) + (A B C D) \\
= \quad & (\neg B D)(\neg C + C) + & \text{(Raccoglimento)} \\
& (A C \neg D) + (A B)(\neg C + (C D)) \\
= \quad & (\neg B D) + & \text{(Inverso)} \\
& (A C \neg D) + (A B)(\neg C + (C D)) \\
= \quad & (\neg B D) + & \text{(Assorbimento II)} \\
& (A C \neg D) + (A B)(\neg C + D) \\
= \quad & (\neg B D) + (A C \neg D) + & \text{(Distributiva)} \\
& (A B \neg C) + (A B D) \\
= \quad & D(\neg B + (A B)) + & \text{(Raccoglimento)} \\
& (A C \neg D) + (A B \neg C) \\
= \quad & D(\neg B + A) + & \text{(Assorbimento II)} \\
& (A C \neg D) + (A B \neg C)
\end{aligned}
$$

Esercizio 2

$$
\begin{aligned}
&= D(\neg B + A) + (AC\neg D) + (AB\neg C) && \\
&= (D\neg B) + (DA) + (AC\neg D) + (AB\neg C) && \text{(Distributiva)} \\
&= (D\neg B) + A(D + (C\neg D)) + (AB\neg C) && \text{(Raccoglimento)} \\
&= (D\neg B) + A(D + C) + (AB\neg C) && \text{(Assorbimento II)} \\
&= (D\neg B) + (AD) + (AC) + (AB\neg C) && \text{(Distributiva)} \\
&= D(A + \neg B) + A(C + (B\neg C)) && \text{(Raccoglimento)} \\
&= D(A + \neg B) + A(C + B) && \text{(Assorbimento II)}
\end{aligned}
$$

Mintermini adiacenti

- Si parte dall'espressione in forma canonica SOP

- Due **mintermini** sono detti **adiacenti** se differiscono su una sola variabile

$$x_4\overline{x_3}x_2\overline{x_1} \qquad\qquad x_4\overline{x_3}x_2x_1$$

- Un'espressione che contiene due mintermini adiacenti può essere minimizzata nel seguente modo

$$x_4\overline{x_3}x_2x_1 + x_4\overline{x_3}x_2\overline{x_1} = x_4\overline{x_3}x_2(x_1 + \overline{x_1}) = x_4\overline{x_3}x_2$$

  - Da 2 prodotti e 8 occorrenze di letterali siamo passati a 1 prodotto e 3 letterali

- Abbiamo bisogno di un metodo che ci permetta di individuare velocemente i **mintermini adiacenti**

Mintermini adiacenti

L'adiacenza fisica non è rispettata se scriviamo le tabelle come abbiamo fatto finora

- Ogni riga riguarda una configurazione di bit anche molto diversa dalla riga precedente
- Qui:
  in rosso i bit di input che cambiano valore rispetto alla riga precedente

a	b	c	F(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Mintermini adiacenti

a	b	F(a,b)
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

a	b	F(a,b)
0	0	**0**
0	1	**0**
1	1	**1**
1	0	**0**

Mappe di Karnaugh

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Tabella di verità classica

nota l'ordine

B C \ A	0	1
0 0	0	0
0 1	1	0
1 1	0	0
1 0	1	1

Mappa di Karnaugh (corrispondente)

- Sono tabelle a due dimensioni che rappresentano la tavola di verità della funzione in un modo diverso.
- Hanno 2^n quadrati se n è il numero di variabili
- Ogni quadrato contiene il valore f_i di un mintermine m_i

Mappe di Karnaugh: copertura

- Data la rappresentazione di una funzione sulla Mappa di Karnaugh possiamo “ricoprirla” con una serie di cubi tali che
  - Ogni cubo contenga solo celle con valore 1
  - Ogni 1 della tavola di verità sia ricoperto da almeno un cubo

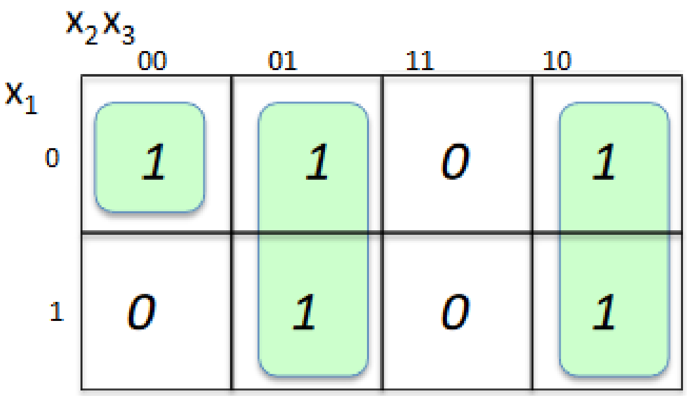
x_1 \ $x_2 x_3$	00	01	11	10
0	1	1	0	1
1	0	1	0	1

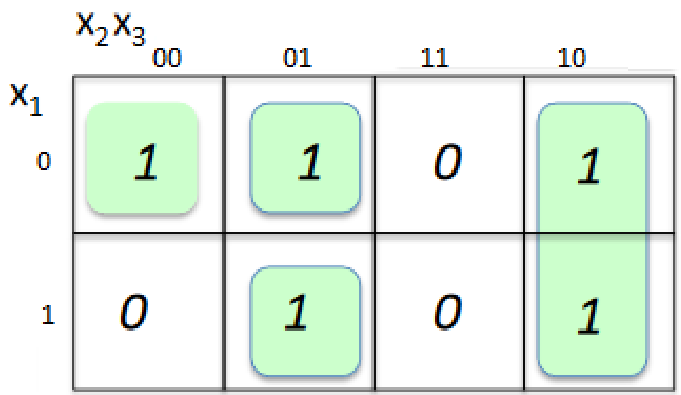
- Ogni cubo rappresenta un prodotto di letterali
- L’unione dei cubi è un’espressione normale in forma SOP

$$f = \overline{x_3}\,\overline{x_2}\,\overline{x_1} + x_3\overline{x_2} + \overline{x_3}x_2$$

Copertura minimale

- Tra tutti gli insiemi di cubi che ricoprono una funzione quelli minimali sono gli insiemi che:
  - Contengono il minor numero possibile di cubi/termini prodotto e, a parità di numero, quelli più grandi





Insieme di cubi minimale

$$f = \overline{x_3}\,\overline{x_2}\,\overline{x_1} + x_3\overline{x_2} + \overline{x_3}x_2$$

Insieme di cubi **non** minimale

$$f = \overline{x_3}\,\overline{x_2}\,\overline{x_1} + \overline{x_3}x_2\overline{x_1} + \overline{x_3}x_2x_1 + x_3\overline{x_2}$$

Copertura minimale

1. Per ogni 1 nella mappa determiniamo i cubi massimali, cioè non contenuti in altri cubi, che lo contengono e che ricoprono solo celle contenenti 1

2. Individuiamo i cubi massimali essenziali
   - Cubi che ricoprono 1 coperti solo da quel cubo massimale

3. Scegliamo un insieme minimale di cubi massimali che ricoprono gli 1 lasciati scoperti dai cubi essenziali

Passo 1

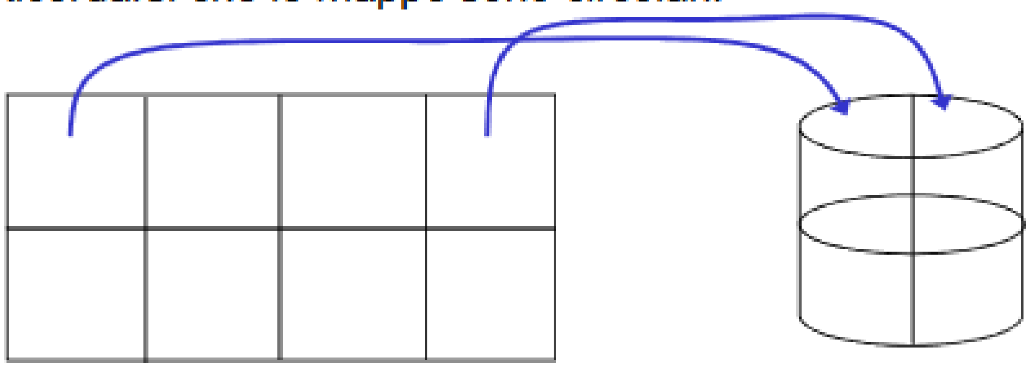
Riscrivere tabella delle verità data come mappa di Karnaugh

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

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

Passo 2

- Identificare un insieme di gruppi adiacenti ***di 2^n celle***
  di tutti 1 in modo che tutti gli 1 appartengano ad almeno un gruppo (SP)
- oppure, di tutti 0 in modo che tutti gli 0 appartengano… (PS)
- Criteri per trovare i gruppi:
  - I gruppi devono essere rettangolari (o quadrati)
  - Più i gruppi sono grandi e più letterali verranno eliminati
  - Meno gruppi danno luogo a meno termini
  - Lo stesso 1 o 0 può essere incluso in più gruppi
  - Ricordarsi che le mappe sono circolari:



Passo 3

- Ogni gruppo corrisponde a un mintermine contenente solo le variabili che non cambiano.

A \ B C	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

- BC + AC + AB

Mappe di Karnaugh: esempi

indipendente da a:
b~c

indipendente da c:
ab

AB C	00	01	11	10
0	000	010	110	100
1	001	011	111	101

AB C	00	01	11	10
0	0	1	1	0
1	0	0	1	0

F = ab + b~c

Esercizio Excel: costruire la mappa di K. con Excel, indicando i cubi di copertura. Usare lo strumento Inserisci – Forme per disegnare il bordo dei cubi. Colorare il bordo di ogni cubo con colore diverso.

Mappe di Karnaugh: esempi

- Rappresentazione piana, utilizzabile per: **2 ≤ N ≤ 4**

B \ A	0	1
0	1	0
1	1	0

F = ~a

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

F = ab + cd + b~c~d

Mappe di Karnaugh: esempi

Mappa di Karnaugh: rappresentazione piana e ciclica

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

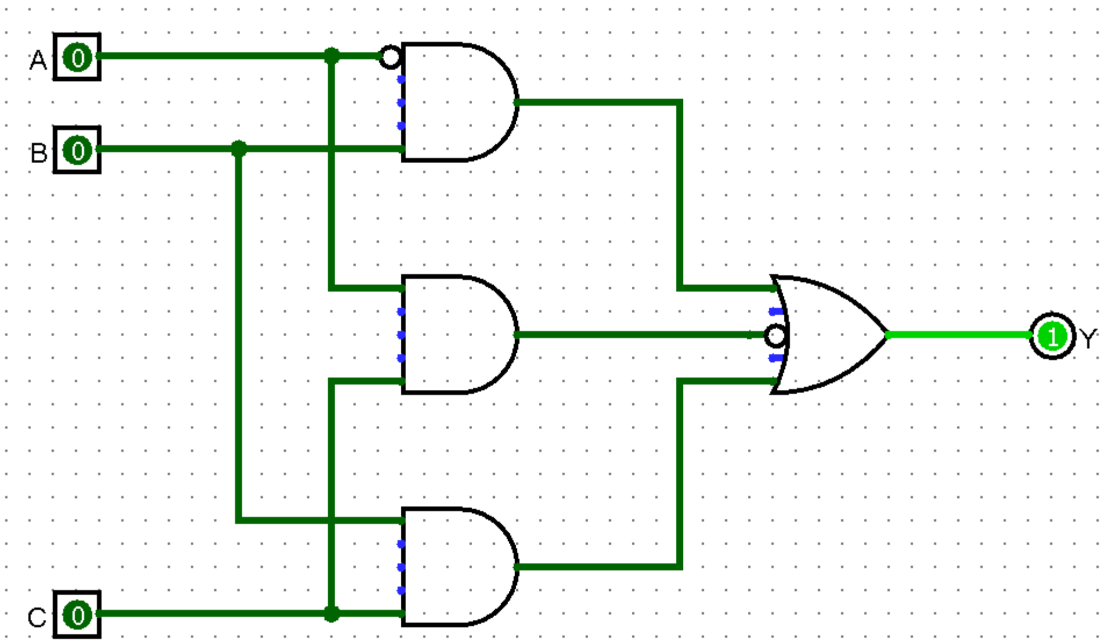
F = ab + b~c~d + ~bcd

Mappe di Karnaugh: esercizio

Minimizzare il circuito dell’esercizio 2 usando le mappe di Karnaugh.

Esercizio 3

Sia dato il seguente circuito



Si determinino:

- La tabella di verità
- La forma canonica più conveniente

Esercizio 3

Tabella di verità

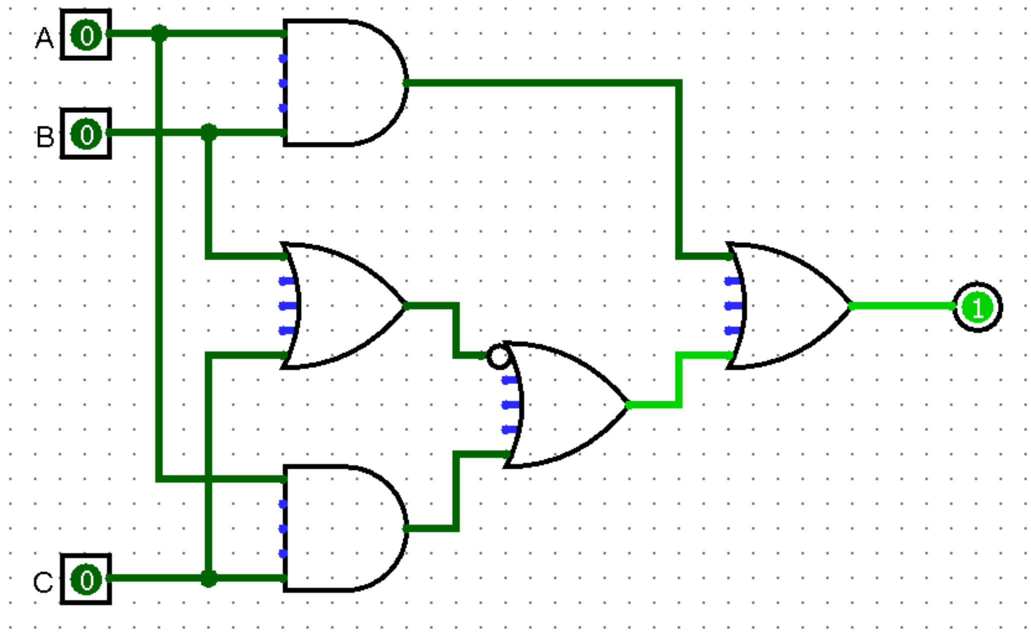
A	B	C	$\neg A$	$Z_1 = \neg AB$	AC	BC	$\neg Z_2$	$Z = Z_1 + \neg Z_2 + Z_3$
0	0	0	1	0	0	0	1	1
0	0	1	1	0	0	0	1	1
0	1	0	1	1	0	0	1	1
0	1	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1	1
1	0	1	0	0	1	0	0	0
1	1	0	0	0	0	0	1	1
1	1	1	0	0	1	1	0	1

Forma canonica POS (un solo maxtermine)

$$Z = \neg A \vee B \vee \neg C = \neg(A \wedge \neg B \wedge C)$$

Esercizio 4

Sia dato il seguente circuito



Si determinino:

- La tabella di verità
- La forma canonica SOP e POS
- La forma algebrica del circuito, semplificando a partire dalla SOP (usare sia manipolazioni algebriche che mappe di K.)

Esercizio 4

Tabella di verità

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

SOP

$$Y = (\neg A \neg B \neg C) + (A \neg B \neg C) + (A \neg B C) + (A B \neg C) + (ABC)$$

POS

$$Y = (A + B + \neg C)(A + \neg B + C)(A + \neg B + \neg C)$$

Forma algebrica semplificata

$$Y = \neg B \neg C + A$$

Esercizio 5

Sia data la seguente tabella di verità

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Si determinino:

- La forma canonica SOP
- La forma algebrica, semplificando a partire dalla SOP
- I cammini critici dei circuiti corrispondenti alle due forme
- Avrebbe senso utilizzare la POS invece della SOP? Perché?

Esercizio 5

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

A	B	C	D	Y
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

SOP

$$(\neg A\neg B\neg C\neg D)+(\neg A\neg BCD)+(\neg ABC\neg D)+(A\neg B\neg CD)+(AB\neg C\neg D)+(ABCD)$$

Cammino critico: (4-1) + (6-1) = 8

Semplificando

$$(\neg A\neg B + AB)(\neg C\neg D + CD) + (\neg ABC\neg D + A\neg B\neg CD)$$

Cammino critico: 1 + 1 + (4-1) = 5

L’uso della POS non è ottimale (ci sono più 0 che 1)

Esercizio 6

Progettare un circuito combinatorio per convertire un codice BCD in un codice Excess-3.

- I calcolatori hanno bisogno di memorizzare un numero decimale per convertirlo nel sistema binario.
- Poiché i calcolatori possono gestire soltanto 0 e 1, bisogna codificare le cifre decimali soltanto con i simboli 0 e 1.
- Il codice binario decimale (BCD) è un sistema pe codificare i numeri decimali:
  - associa ogni cifra decimale 0,1,2,..9 a un numero binario di 4 bit.
- Un numero con n cifre decimali codificato in BCD richiede 4n bit:
  - ciascun gruppo di quattro bit rappresenta una cifra decimale.
- $(185)_{10} = (\underbrace{0001}_{1}\ \underbrace{1000}_{8}\ \underbrace{0101}_{5})_{BCD} \neq (10111001)_2$

Binary-Coded Decimal (BCD)

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Excess-3 è un sistema per codificare i numeri decimali.
Esso associa **ogni cifra decimale** al numero binario di BCD **aumentato di 3 unità**.
Vediamo un esempio pratico di codifica:

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (0100\ 1011\ 1000)_{E\text{-}3}$$

$$0100 = 0001 + 0011$$

$$1011 = 1000 + 0011$$

$$1000 = 0101 + 0011$$

Costruire un convertitore di numero decimale espresso in codifica BCD nel corrispondente numero espresso in codifica Excess-3.
Numero di ingressi e di uscite: entrambi i codici utilizzano **4 bit**.
In tutto, 4 variabili **ABCD** in ingresso e 4 variabili **WXYZ** in uscita.

a) Compilare la tabella di verità del convertitore, tenendo presente che:
- il codice E-3 è ottenuto aggiungendo $(0011)_2 = (3)_{10}$ al codice BCD
- consideriamo solo 10 righe: le altre 6 combinazioni dei valori non sono utilizzate (possono essere trattate come don't care nelle mappe)

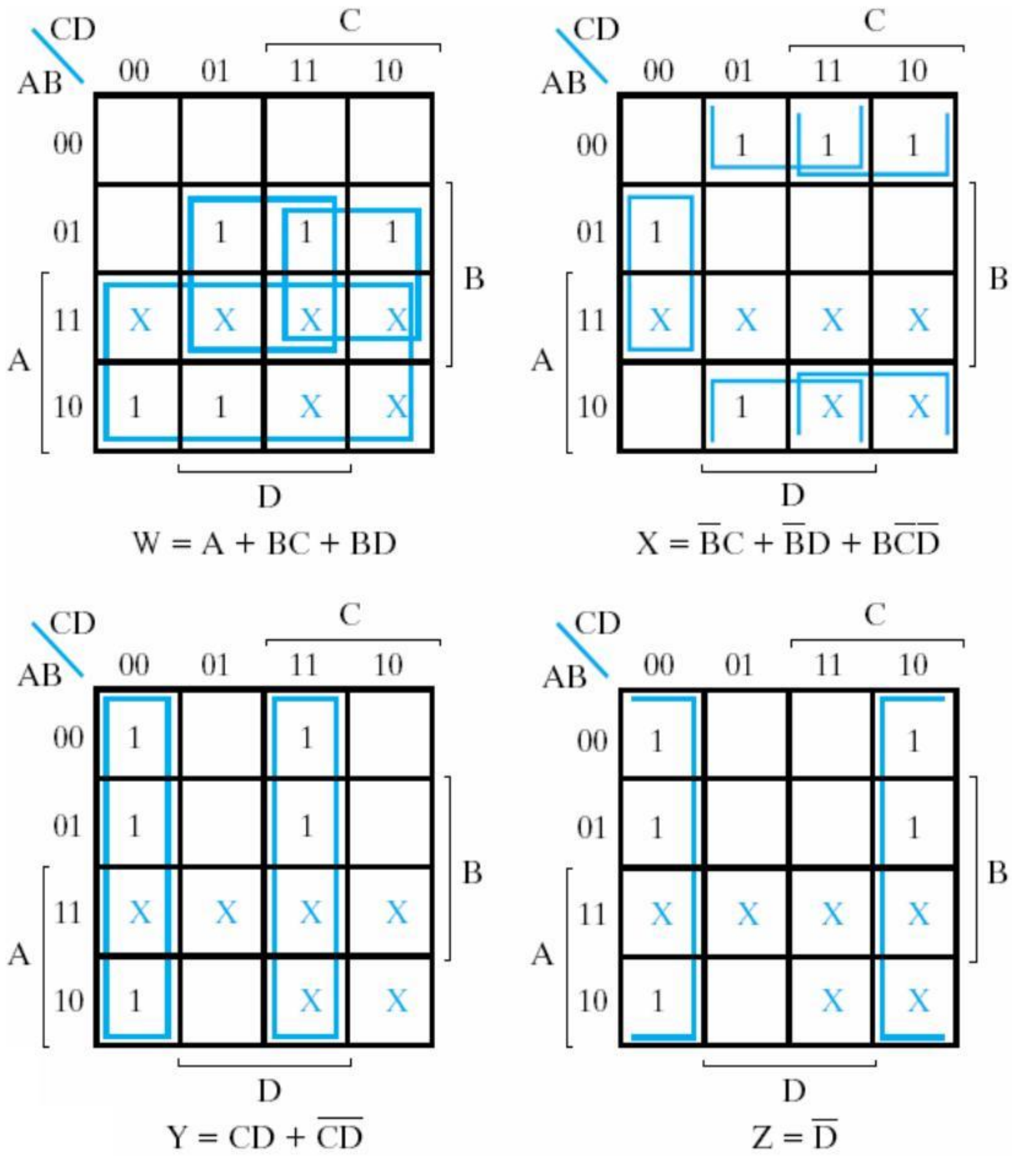
b)

b) Scrivere le funzioni booleane di ogni uscita in forma minima utilizzando le mappe di Karnaugh per l'ottimizzazione.

Soluzione: tabella di verità

Decimal Digit	Input BCD				Output Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Soluzione: mappe di K.



$W = A + BC + BD$

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

$Y = CD + \overline{C}\overline{D}$

$Z = \overline{D}$