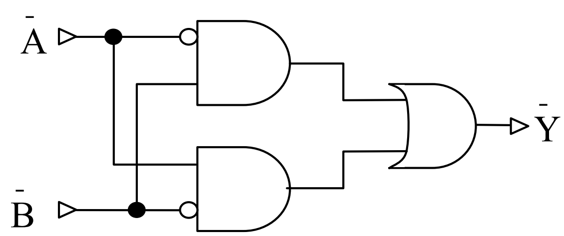
Laboratorio del 10/11/2010 - Soluzioni

Rappresentazioni possibili per una funzione logica:

- **circuito logico:**



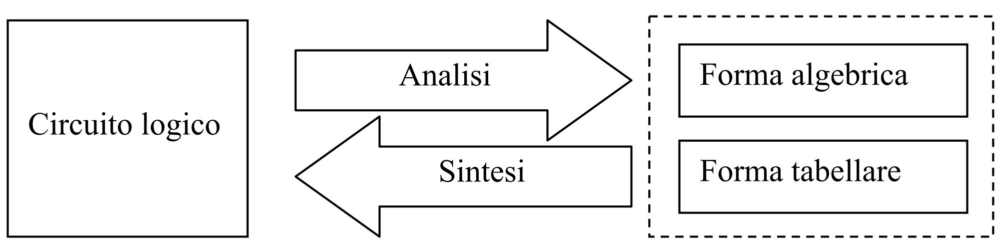
- **forma tabellare** (tabella lookup):

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

- **formula algebrica:**

Y= (NOT A)B OR NOT(B) A

Sintesi e analisi di circuiti logici:



Massimo Marchi – marchi@dsi.unimi.it

Precedenza degli operatori logici:

In assenza di parentesi, AND ha la priorità sull'OR ed il NOT su entrambi:

NOT > AND > OR

Principio di dualità:

Il duale di una funzione si ottiene sostituendo:
AND con OR,OR con AND, 0 con 1 ed 1 con 0.

Proprietà generali dell'Algebra Booleana:

0. **Doppia Inversione**	$\sim(\sim \boldsymbol{x}) = \boldsymbol{x}$	
1. **Identità**:	$\mathbf{1}\,\boldsymbol{x} = \boldsymbol{x}$	$\mathbf{0} + \boldsymbol{x} = \boldsymbol{x}$
2. **Elemento nullo**:	$\mathbf{0}\,\boldsymbol{x} = \mathbf{0}$	$\mathbf{1} + \boldsymbol{x} = \mathbf{1}$
3. **Idempotenza**:	$\boldsymbol{x}\,\boldsymbol{x} = \boldsymbol{x}$	$\boldsymbol{x} + \boldsymbol{x} = \boldsymbol{x}$
4. **Inverso:**	$\boldsymbol{x} \sim \boldsymbol{x} = \boldsymbol{0}$	$\boldsymbol{x} + \sim \boldsymbol{x} = \mathbf{1}$
5. **Commutativa**:	$\boldsymbol{x}\,\boldsymbol{y} = \boldsymbol{y}\,\boldsymbol{x}$	$\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$
6. **Associativa:**	$(\boldsymbol{x}\,\boldsymbol{y})\,\boldsymbol{z} = \boldsymbol{x}\,(\boldsymbol{y}\,\boldsymbol{z})$	$(\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} = \boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z})$
7. **Distributiva:**	$\boldsymbol{x}\,(\boldsymbol{y} + \boldsymbol{z}) = \boldsymbol{x}\,\boldsymbol{y} + \boldsymbol{x}\,\boldsymbol{z}$	$\boldsymbol{x} + \boldsymbol{y}\,\boldsymbol{z} = (\boldsymbol{x} + \boldsymbol{y})\,(\boldsymbol{x} + \boldsymbol{z})$
8. **Assorbimento:**	$\boldsymbol{x}\,(\boldsymbol{x} + \boldsymbol{y}) = \boldsymbol{x}$	$\boldsymbol{x} + \boldsymbol{x}\,\boldsymbol{y} = \boldsymbol{x}$
9. **De Morgan**:	$\sim(\boldsymbol{x}\,\boldsymbol{y}) = \sim\boldsymbol{x} + \sim\boldsymbol{y}$	$\sim(\boldsymbol{x} + \boldsymbol{y}) = \sim\boldsymbol{x}\ \sim\boldsymbol{y}$

Forma Canonica SOP *(Sum Of Product)* :

Implicante → Prodotto di variabili, semplici o negate, che vale 1 per combinazioni di valori per cui la funzione data vale 1 (*se implicante vale 1 allora la funzione vale 1, il contrario può non valere*)

Mintermine → Implicante contenente tutte le variabili della funzione data

Forma **SOP** di F → $F = \sum_{j=1}^{Q} m_j$ dove m_j è il j-esimo mintermine della funzione F

Ex: **A** XOR **B** = $m_1 + m_2$ = ~**A B** + **A ~B**

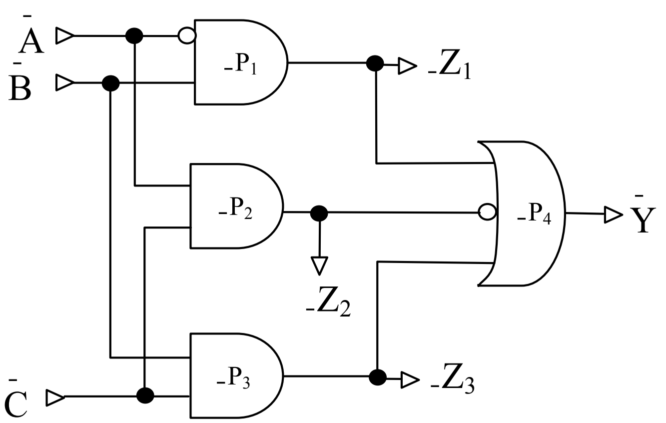
Forma Canonica POS *(Product Of Sum)* :

Maxtermine → Somma contenente tutte le variabili eventualmente negate, che vale 0 quando la funzione data vale 0 (se il maxtermine vale 0 allora la funzione vale 0, il contrario può non valere)

Forma **POS** di F → $F = \prod_{i=1}^{Q} M_i$ dove Mi è il i-esimo Maxtermine della funzione F

Ex: **A** XOR **B** = $M_0\,M_3$ = **(A + B)** (~**A+ ~B)**

1. **Ricavare la forma tabellare , la prima forma canonica e la forma algebrica del seguente circuito semplificando dove possibile.**



Nel ricavare la forma tabellare dal circuito logico conviene procedere dagli ingressi verso le uscite,individuando ogni punto di calcolo intermedio e l'ordine in cui questi calcoli saranno disponibili e quindi

Costruite una tabella con una riga per ogni possibile combinazione degli ingressi ed una colonna con i risultati intermedi per l'uscita di ogni porta logica, ev. aggiungendo ulteriori colonne per il risultato negato (come $\sim Z_2$ nell'esempio).

AND OR

B	**C**	~A	Z_1=~AB	Z_2=AC	Z_3=BC	~Z_2	**Y=Z_1+~Z_2+Z_3**
0	0	1	0	0	0	1	1
0	1	1	0	0	0	1	1
1	0	1	1	0	0	1	1
1	1	1	1	0	1	1	1
0	0	0	0	0	0	1	1
0	1	0	0	1	0	0	0
1	0	0	0	0	0	1	1
1	1	0	0	1	1	0	1

Nota: *una porta AND dà come risultato 0 quando uno dei suoi ingressi è 0. Per calcolare una colonna risultato di una AND allora è comodo procedere nel seguente modo: per ogni termine si identificano le celle a 0 e si pone a 0 la cella risultato corrispondente. Alla fine le celle ancora vuote si pongono ad 1. Analogamente il metodo duale può essere applicato alle porte OR: prima si identificano le celle risultato a 1 corrispondenti ai termini posti uguale a 1 quindi si completano le celle ancora vuote con 0.*

Es.: Si consideri il calcolo di Z_1. Il termine ~A va a zero per le ultime quattro configurazioni. Ne segue che Z_1 può essere posto a zero per le corrispondenti celle. Analogamente B è uguale a zero per le prime due configurazioni e per la 5^{ta} e la 6^{ta} . Ne segue che Z_1 può essere messo a 0 anche per le prime due celle. Le restanti celle saranno obbligatoriamente uguali ad 1.

Nota: Il numero di mintermini nella forma canonica SOP è pari al numero di 1 nella colonna risultato. Viceversa il numero di maxtermini presenti nella seconda forma canonica POS è uguale al numero di 0 presenti. In questo caso quindi sarebbe più conveniente in termini di compattezza di descrizione usare la seconda forma canonica POS al posto della forma SOP.

Calcoliamo una forma algebrica semplificata partendo dal circuito logico:

Nel ricavare la formula dal circuito logico è comodo ricostruire la formula partendo dalle uscite e risalendo poi il circuito fino agli ingressi.

NOTA: Per ogni passaggio di semplificazione è indicato il punto di lavoro con un evidenziazione gialla e il risultato ottenuto con una zona sottolineata nella riga successiva.

Y = ← *Sviluppo la porta P_4*

= (Z_1 + ~Z_2 + Z_3) ← *Sviluppo le porte P_1, P_2 e P_3*

= (~AB + ~(AC) + BC) ← *9a De Morgan:* ~(**xy**) = ~**x** + ~**y**

= (~AB + ~A + ~C + BC) ← *8b Assorbimento:* **xy** + **x**= ~**x** *(**x**=~A,**y**=B)*

= (~A + ~C + BC)

Vale la seguente proprietà: **x + ~xy = x + y**

Dim: **x + ~xy**	= (**x + xy**) + ~**xy**	←*8b*
	= ***x*** + (***xy*** + ~***xy***)	←*6b*
	= ***x*** + ***y(x*** + ~***x)***	←*7a*
	= ***x*** + ***y1***	←*4b*
	= ***x*** + ***y***	←*1a*

Questo *non è altro che l'unico maxtermine della seconda forma canonica.*

= (~A + ~C + BC) ← *Applico:* **x + ~xy = x + y**

= (~A + ~C + B) ← *8b De Morgan*

= ~ (A~BC)

Questo *non è altro che l'unico mintermine della prima forma canonica della funzione ottenuta negando quella data.*

Calcoliamo lo stesso risultato partendo dalla tabella la forma SOP e semplificando:

La prima forma canonica si ottiene sommando i mintermini della funzione risultato. Ad ogni combinazione degli ingressi per cui la funzione vale 1 corrisponde un mintermine.

Y = ~A~B~C + ~A~BC + ~AB~C + ~ABC + A~B~C + AB~C + ABC

Semplifichiamo applicando varie volte le regole 7a e 4b:

Questo corrisponde ad individuare di volta in volta degli *implicanti* (**y**) sempre più piccoli della funzione data

x y + ~***x y*** = (***x*** + ~***x***) ***y*** = ***1 y*** = ***y***

Y = ~A~B~C + ~A~BC + ~AB~C + ~ABC + A~B~C + AB~C + ABC

Y = ~A~B + ~AB~C + ~ABC + A~B~C + AB~C + ABC

Y = ~A~B + ~AB + A~B~C + AB~C + ABC

Y = ~A~B + ~AB + A~C + ABC

Y = ~A + A~C + ABC ← *6b Associativa:* **A**(**w**)+**A**(**z**) = **A**[(**w**) +(**z**)]

Y = ~A + A(~C + BC) ← **x + ~xy = x + y**: **x**=~*C* , ~***x***=~*(~C)=C*

Y = ~A + A(~C + B) ← **x + ~xy = x + y**: **x**=~*A* , ***y***=*(~C+B)*

Y = ~A + ~C + B = ~(A~BC) ← *Vedi sopra*

Applichiamo Karnaugh:

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

~A

B

~C

Gli indici di tabella vanno riempiti in modo che la variazione sia limitata ad un solo bit per volta. Nel caso a tre variabili, due variabili andranno sulle colonne (A e B) ed una sulle righe (C). Gli indici di colonna saranno:

$$\mathbf{00 \rightarrow 01 \rightarrow 11 \rightarrow 10}$$

Gli indici di riga:

$$\mathbf{0 \rightarrow 1}$$

Per identificare gli implicanti massimi bisogna individuare i box 1x1, 1x2, 1x4, 2x1, 2x2, 2x4, 4x1, 4x2 (4x4) che siano "giustamente allineati" alla tabella e che coprano tutti e soli gli **1.**

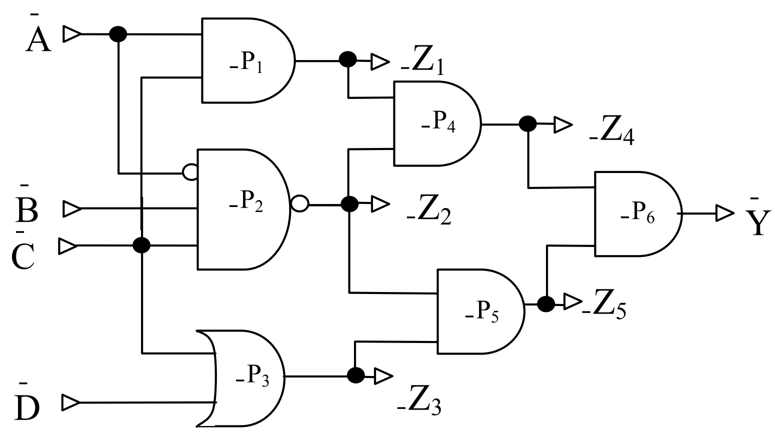
La definizione formale di "giustamente allineati" è alquanto complicata da scrivere per esteso. Più semplice è darne una definizione informale: un box è giustamente allineato se le configurazioni degli ingressi che copre esauriscono tutte le combinazioni possibili delle variabili che cambiano.

> *Es: il box che copre tutta la prima riga della tabella copre le configurazioni 000,010,110,100; in queste configurazioni C vale sempre* ***0*** *mentre le altre variabili A e B assumono tutte le configurazioni possibili 00,01,11,10.*

Dai box individuati si deducono gli implicanti della funzione considerando per ogni box le variabili che restano fisse:

$$\mathbf{Y = \sim A + \sim C + B}$$

2. **Ricavare la forma tabellare , la prima forma canonica e la forma algebrica del seguente circuito semplificando dove possibile.**



Forma tabellare:

A	**B**	**C**	**D**	~A	Z_1= AC	~Z_2=~ABC	Z_3=C+D	Z_2	Z_4=Z_1Z_2	Z_5=Z_2Z_3	**Y=Z_4Z_5**
0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	1	0	0	1	1	0	1	0
0	0	1	0	1	0	0	1	1	0	1	0
0	0	1	1	1	0	0	1	1	0	1	0
0	1	0	0	1	0	0	0	1	0	0	0
0	1	0	1	1	0	0	1	1	0	1	0
0	1	1	0	1	0	1	1	0	0	0	0
0	1	1	1	1	0	1	1	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	1	1	0	1	0
1	1	1	0	0	1	0	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1

Nota: gli "1" presenti nella tabella appartengono tutti al solo implicante AC poichè la funzione è pari ad 1 per ogni configurazione possibile tale che A=1 e C=1. Ne segue che la forma minima è Y=AC

Calcoliamo la forma algebrica partendo dal circuito logico:

$$Y = (Z_4\ Z_5) = ((Z_1Z_2)\,(Z_2Z_3))$$

$$= (((AC)\overline{(\overline{A}BC)})\,(\overline{(\overline{A}BC)}(C+D))) \quad \leftarrow \textit{Sviluppo le porte del circuito}$$

Quando compaiono diversi livelli di negazione conviene applicare ripetutamente Demorgan e Doppia Negazione

$$= (((AC)(A+\overline{B}+\overline{C}))\,((A+\overline{B}+\overline{C})(C+D))) \quad \leftarrow \textit{9a De Morgan, 0 Doppia Negazione}$$

$$= (ACA+AC\overline{B}+AC\overline{C})\,((A+\overline{B}+\overline{C})C+(A+B+C)D) \quad \leftarrow \textit{7a Distributiva}$$

$$= (AAC+AC\overline{B}+A0)\,((AC+\overline{B}C+\overline{C}C)+(AD+\overline{B}D+\overline{C}D)) \quad \leftarrow \textit{5a Commutativa, 7a Distributiva, 4a Inverso}$$

$$= (AC+AC\overline{B})\,(AC+\overline{B}C+0+AD+\overline{B}D+\overline{C}D) \quad \leftarrow \textit{3a Idempotenza, 2a Elemento nullo}$$

$$= (AC)\,(AC+\overline{B}C+AD+\overline{B}D+C\overline{D}) \quad \leftarrow \textit{8b Assorbimento}$$

$$= (ACAC+AC\overline{B}C+ACAD+AC\overline{B}D+ACC\overline{D}) \quad \leftarrow \textit{7a Distributiva}$$

$$= AC+AC\overline{B}+ACD+AC\overline{B}D+AC\overline{D} \quad \leftarrow \textit{3a Idempotenza, 5a Commutativa}$$

$$= AC+AC\overline{B}+AC+AC\overline{B}D \quad \leftarrow xy+x\overline{y} = x(y+\overline{y}) = x1 = x$$

$$= AC+AC\overline{B}D \quad \leftarrow \textit{8b Assorbimento (2 volte)}$$

$$= \mathbf{\underline{AC}}$$

Nota: Questo è l' implicanti indviduato in tabella! C'è solo lui perché copre tutti gli **1** *della tabella (vedi mappe di karnaugh)*

Calcoliamo la forma canonica partendo dalla tabella e semplifichiamo:

Y = A~BC~D + A~BCD + ABC~D + ABCD

Semplifichiamo applicando varie volte: ***x y* + ~*x y* = *y***

Y = A~BC~D + A~BCD + ABC~D + ABCD

= A~BC~D + A~BCD + ABC

= A~BC + ABC

= AC

Nello sviluppo e semplificazione tramite le SOP conviene (quasi sempre) come primo passaggio individuare degli implicanti della funzione data. Questo si può ottenere individuando iterativamente coppie di termini uguali eccetto che per un letterale che compare sia negato che non negato ed applicando la regola

xy +xy = x (y+y)=x1=x

Applichiamo Karnaugh:

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

AC

Dai box individuati si deduce che l'unico implicante della funzione è:

Y = AC

3. **Data la seguente tabella di verità, ricavare la forma tabellare , il circuito equivalente, la prima e la seconda forma canonica semplificando dove possibile.**

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

Calcoliamo la forma canonica SOP partendo dalla tabella e semplifichiamo:

La prima forma canonica si ottiene sommando i mintermini della funzione risultato. Ad ogni combinazione degli ingressi per cui la funzione vale 1 corrisponde un mintermine.

Y = ~A~BC + ~AB~C + A~B~C + ABC

Proviamo a semplificare. Si nota rapidamente che i minterm non possono essere accorpati quindi la forma SOP è già minima se si utilizzano solo AND e OR. Se dividiamo la tabella in due metà possiamo notare che la parte superiore corrisponde alla funzione **B xor C** mentre la parte sotto corrisponde a ~(B XOR C). Esiste quindi una forma più compatta che usi porte XOR:

Y=~A(B XOR C) + A~(B XOR C)

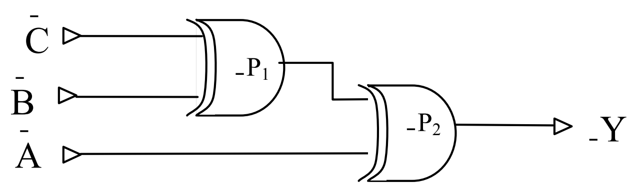
Deriviamo la funzione dalla SOP:

Y	= ~A~BC + ~AB~C + A~B~C + ABC	
	= ~A(~BC + B~C) + A(~B~C + BC)	→ xy=xz=x(y+z)
	= ~A(B⊕C) + A~ ~ (~B~C + BC)	→ x= ~~x
	= ~A(B⊕C) + A~ (~(~B~C) ~(BC))	→ ~(x+y)=(~x~y)
	= ~A(B⊕C) + A~ ((~~B+~~C) (~B+~C))	→ ~(xy)=(~x+~y)
	= ~A(B⊕C) + A~ ((B+C) (~B+~C))	→ x= ~~x
	= ~A(B⊕C) + A~ (B~B + B~C +C~B+C~C)	→ (x+y)z = xz +yz
	= ~A(B⊕C) + A~ (0 + B~C +C~B+0)	→ x~x = 0
	= ~A(B⊕C) + A~ (B~C +C~B)	→ x+0 = x
	= ~A(B⊕C) + A~(B⊕C)	

Esaminando la formula si nota che i termini A e (B XOR C) sono sistemati in modo da formare un'ulteriore XOR. Ne segue che la formula può essere ulteriormente semplificata::

$$Y = A \oplus (B \oplus C)$$

Disegnamo il circuito:



Calcoliamo la seconda forma canonica POS partendo dal risultato precendente:

La seconda forma canonica si ottiene considerando tutte le configurazioni in cui la funzione risulta uguale a 0: il maxtermine corrispondente si ottiene riportando la somma di ogni letterale della configurazione, eventualmente negato se uguale a 1:

Es.: se F=1 per A=1,B=0,C=1 allora il MaxTermine corrispondente è ~A+B+~C

In questo caso la POS è uguale a:

$$\mathbf{Y = (A + B + C)(A + \sim B + \sim C)(\sim A + B + C)(\sim A + \sim B + \sim C)}$$

E' possibile ricavare lo stesso risultato precedente ache partendo dalla POS:

$$\begin{aligned}
Y &= (A + B + C)(A + \sim B + \sim C)(\sim A + \sim B + C)(\sim A + B + \sim C) \\
&= [A + (B + C)(\sim B + \sim C)][\sim A + (\sim B + C)(B + \sim C)] \qquad \rightarrow \boldsymbol{x+yz=(x+y)(x+z)} \\
&= [A + (B + C)\sim B + (B + C)\sim C)][\sim A + (\sim B + C)B + (\sim B + C)\sim C)] \\
&= [A + B\sim B + C\sim B + B\sim C + C\sim C][\sim A + \sim B B + CB + \sim B\sim C + C\sim C)] \\
&= [A + C\sim B + B\sim C][\sim A + CB + \sim B\sim C] \\
&= [A + C\oplus B][\sim A + \sim(C\oplus B)] \\
&= [A + C\oplus B] \sim [A (C\oplus B)]\ ) \\
&= \sim(\sim[A + C\oplus B] + [A (C\oplus B)]) \\
&= \sim[\sim A\sim(C\oplus B) + A(C\oplus B)] \\
&= \sim\sim[A\oplus(C\oplus B)] \\
&= A\oplus(C\oplus B)
\end{aligned}$$

Massimo Marchi – marchi@dsi.unimi.it