Esercitazione 03 - Soluzioni

Esercizio 1: Si consideri la porta XNOR. Si ricavi la SOP per la XNOR e si simuli in Gatesim il circuito equivalente. Si dica se la forma circuitale derivata è "ottima" e si spieghi in che senso.

Soluzione: Il circuito SOP della XNOR ha cammino critico pari a 2 e complessità pari a 3 porte. Migliorare il cammino critico significherebbe usare una sola porta AND o OR ma si può verificare esaustivamente che qualunque sia la configurazione di negazioni applicate agli ingressi ed all'uscita, AND e OR non possono generare una tabella di verità con esattamente due 1 e due 0. Migliorare la complessità significherebbe usare solo due porte in cascata: due AND, due OR o una AND e una OR. Si può verificare esaustivamente che nessuna di queste configurazioni può generare la tabella di verità della XNOR.

Esercizio 2: Sia Y=A(A + ~B)(B + C)+ ~BD una funzione logica. Si ricavi la tabella di verità e la SOP. Si implementino in Gatesim il circuito associato alla formula originale ed il circuito associato alla SOP e li si confrontino. Si proceda poi alla semplificazione algebrica della SOP, si implementi il circuito corrispondente e lo si confronti con gli altri due circuiti implementati.

Soluzione:

La tabella di verità:

A	B	C	D	~B	A+~B	B+C	~BD	A(A+~B)(B+C)	Y
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

Nota: Esaminando la tabella di verità generalmente si riescono a ricavare indicazioni utili per la semplificazione della funzione di partenza. Guardando la tabella si identificano a colpo d'occhio alcuni implicanti:

AB, ~A~BD

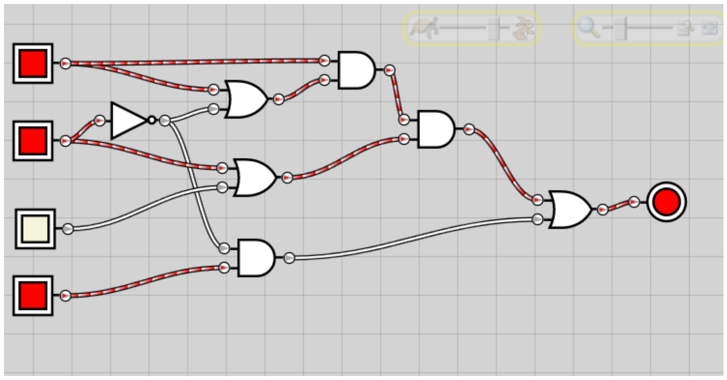
In aggiunta la parte inferiore della tabella è la funzione B+C+D. Ne segue che una forma ridotta (non è detto che sia la migliore) della funzione è la seguente:

Y=~A~BD + A(B+C+D)

In questo caso purtroppo la soluzione trovata non è migliore rispetto a quella originale.

Massimo Marchi – marchi@dsi.unimi.it

Il circuito della funzione originaria è il seguente:

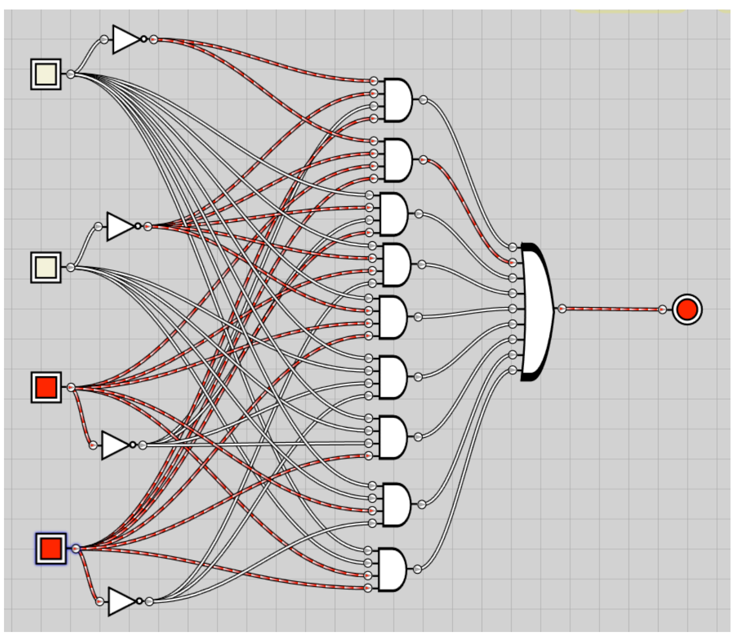


Il circuito ha cammino critico 4 e costo 6.

La forma SOP è la seguente:

Y = (~A~B~CD)+(~A~BCD)
+(A~B~CD)+(A~BC~D)+(A~BCD)+(AB~C~D) +(AB~CD)+(ABC~D) +(ABCD)

Il circuito risultante è il seguente (*Gatesim ha difficoltà a maneggiare circiti così complessi, ne segue una certa lentezza nella commutazione delle porte*):



Semplifichiamo la funzione originaria partendo dalla SOP:

Y = (~A~B~CD)+(~A~BCD)
+(A~B~CD)+(A~BC~D)+(A~BCD)+(AB~C~D) +(AB~CD)+(ABC~D) +(ABCD)

applico varie volte **xy+x~y=x**

Y = ~A~BD +A~B~CD + A~BC~D + A~BCD + AB~C~D + AB~CD + ABC~D + ABCD

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

Y = ~A~BD + A~B~CD + A~BC +AB

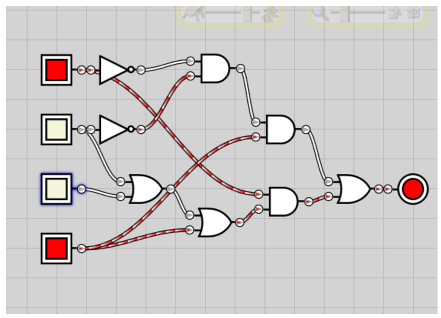
Y = ~A~BD + A(~B~CD + ~BC +B)

Y = ~A~BD + A(~B(~CD + C) +B)

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

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

Y = ~A~BD + A(B + C +D)



Il circuito così ha costo 6 e cammino critico 4, provo a estrarre l'implicante AB per vedere se trovo una forma migliore:

Y = ~A~BD + A(B + C +D)

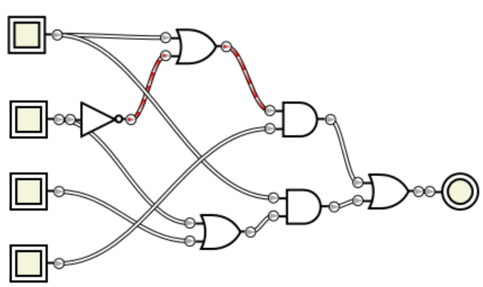
Y = ~A~BD + AB + AC +AD

Y = ~A~BD +AD + AB + AC

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

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

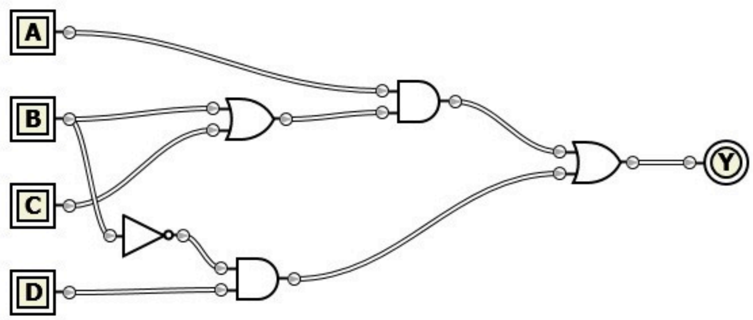
Il circuito così ha costo 5 e cammino critico 3:



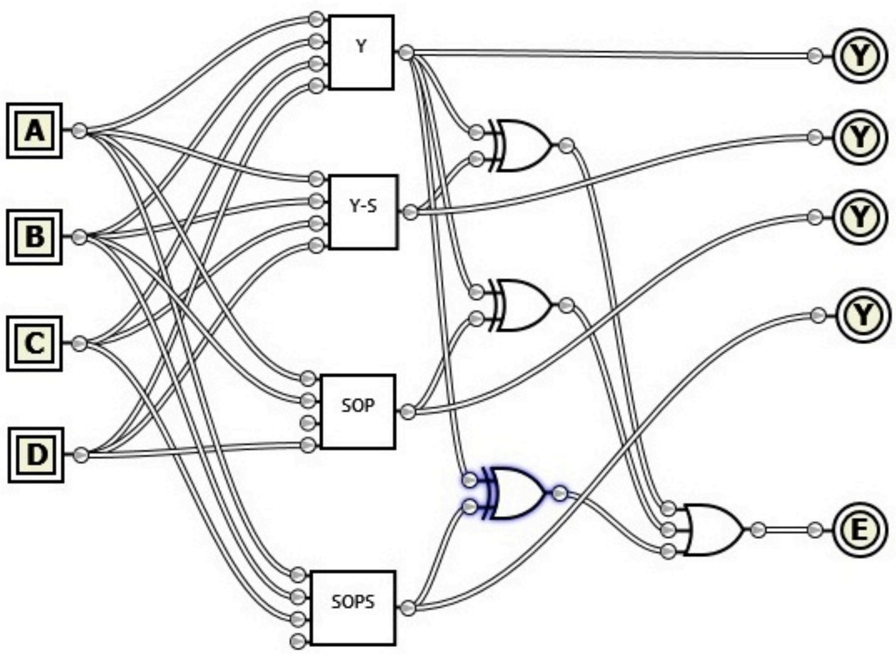
Partendo dalla funzione originale Y=A(A + ~B)(B + C)+ ~BD possiamo semplificare in questo modo:

Y	=	A(A+~B)(B+C) + ~BD	**x(x+y)=x, x=A, y=B**
Y	=	A(B+C) + ~BD	

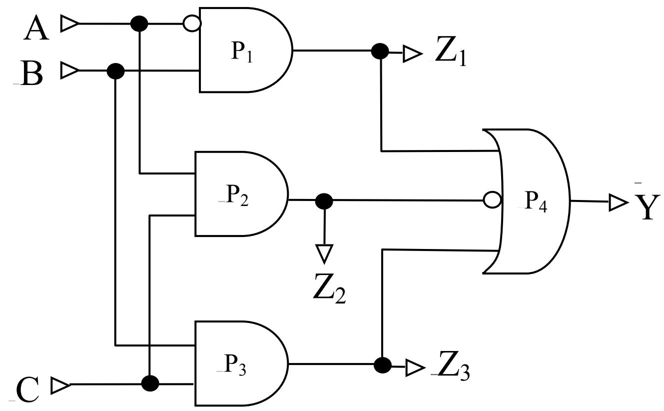
Il circuito cosi semplificato ha cammino critico 3 e costo 4:



Per confrontare i vari circuiti si può realizzare un circuito come segue:



Esercizio 3: 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 utilizzabili

Soluzione:

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

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

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)

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

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

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

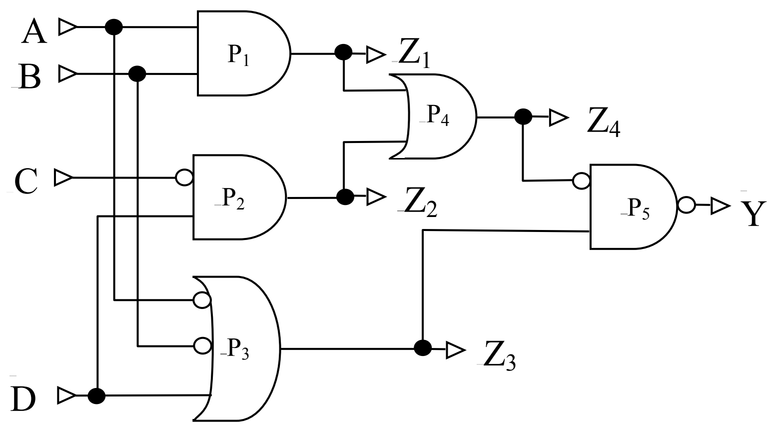
Y = ~A + A~C + ABC ← *6b Associativa:* **A(w)+A(z) = A[(w) +(z)]**

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*

Esercizio 4: Ricavare la forma tabellare , la prima forma canonica e la forma algebrica del seguente circuito semplificando dove possibile.



Soluzione:

Forma tabellare:

A	**B**	**C**	**D**	~A	~B	~C	Z_1= AB	Z_2=~CD	Z_3=~A~BD	Z_4= Z_1+ Z_2	~Z_4	Z_3~Z_4	**Y**
0	0	0	0	1	1	1	0	0	1	0	1	1	0
0	0	0	1	1	1	1	0	1	1	1	0	0	1
0	0	1	0	1	1	0	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0	0	1	0	1	1	0
0	1	0	0	1	0	1	0	0	1	0	1	1	0
0	1	0	1	1	0	1	0	1	1	1	0	0	1
0	1	1	0	1	0	0	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	0	1	1	0
1	0	0	0	0	1	1	0	0	1	0	1	1	0
1	0	0	1	0	1	1	0	1	1	1	0	0	1
1	0	1	0	0	1	0	0	0	1	0	1	1	0
1	0	1	1	0	1	0	0	0	1	0	1	1	0
1	1	0	0	0	0	1	1	0	0	1	0	0	1
1	1	0	1	0	0	1	1	1	1	1	0	0	1
1	1	1	0	0	0	0	1	0	0	1	0	0	1
1	1	1	1	0	0	0	1	0	1	1	0	0	1

Nota: dalla tabella si possono ricavare suggerimenti sul come semplificare la formula data.
Ex: il gruppo di **1** *corrispondenti alle configurazioni per cui A e B valgono* **1** *suggerisce di cercare di derivare l'implicante* **AB** *durante la semplificazione della SOP.*

Stesso discorso vale per gli **1** *presenti quando* **CD = 01** *che suggerisce di cercare di derivare l'implicante* ~**CD.**

Calcoliamo la forma algebrica partendo dal circuito logico:

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

$$Y = \overline{(\overline{Z_4}+Z_3)} = \overline{(\overline{Z_1+Z_2})\ (\overline{A}+\overline{B}+D)} \qquad \leftarrow \textit{Sviluppo le porte del circuito}$$

$$= \overline{\overline{(AB+\overline{C}D)}\ (\overline{A}+\overline{B}+D)} \qquad \leftarrow \textit{9a De Morgan}$$

$$= \overline{\overline{(AB+\overline{C}D)}} + \overline{(\overline{A}+\overline{B}+D)} \qquad \leftarrow \textit{0 Doppia Negazione}$$

$$= (AB+\overline{C}D) + \overline{(\overline{A}+\overline{B}+D)} \qquad \leftarrow \textit{9b De Morgan}$$

$$= (AB+\overline{C}D) + (\ \overline{(\overline{A}+\overline{B})}\ \overline{D})\ ) \qquad \leftarrow \textit{9b De Morgan}$$

$$= (AB+\overline{C}D) + (\overline{\overline{A}}\ \overline{\overline{B}}\ \overline{D}) \qquad \leftarrow \textit{0 Doppia Negazione}$$

$$= (AB+\overline{C}D + (AB\overline{D}) \qquad \leftarrow \textit{5b Commutativa}$$

$$= AB + AB\overline{D} + \overline{C}D \qquad \leftarrow \textit{8b Assorbimento}$$

$$= \mathbf{AB + \overline{C}D}$$

Nota: Questa è la somma dei due implicanti indviduati in tabella!!! Ci sono solo loro perché insieme coprono tutti gli **1** *della tabella (vedi mappe di karnaugh)*

Calcoliamo la forma canonica partendo dalla tabella e semplifichiamo:

Y = ~A~B~CD + ~AB~CD + A~B~CD + AB~C~D + AB~CD + ABC~D + ABCD

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

Y = ~A~B~CD + ~AB~CD + A~B~CD + AB~C~D + AB~CD + ABC~D + ABCD
= ~A~B~CD + ~AB~CD + A~B~CD + AB~C~D + AB~CD + ABC
= ~A~B~CD + ~AB~CD + A~B~CD + AB~C + ABC
= ~A~B~CD + ~AB~CD + A~B~CD + AB
= ~B~CD + ~AB~CD + AB

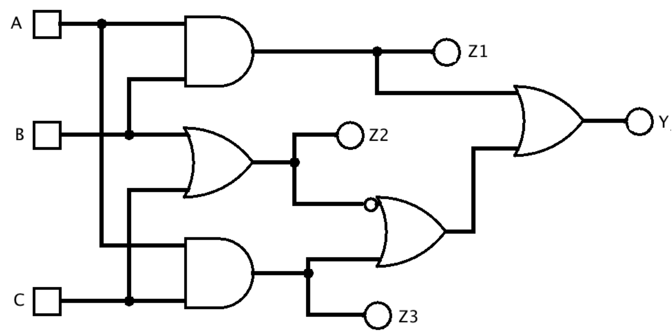
Nello sviluppo e semplificazione tramite le SOP conviene come primo passaggio individuare degli implicanti della funzione data. Questo si può ottenere individuando iterativamente coppie di termini uguali eccetto che per un letterale che compare sia negato che non negato.

Applico sul termine **AB** la regola di 8b, ***x* = *x* + *x y*,**
in modo da creare un termine **AB~CD** da utilizzare nella semplificazione successiva:

= ~B~CD + ~AB~CD + AB
= ~B~CD + ~AB~CD + AB + AB~CD ← 7a e 4b: ***x y*** + ~***x y*** = ***y***
= ~B~CD + B~CD + AB ← idem
= ~CD + AB ← *5b Commutativa*
= **AB + ~CD**

Massimo Marchi – marchi@dsi.unimi.it

Esercizio 5: Ricavare la forma tabellare , la prima forma canonica e la forma algebrica del seguente circuito semplificando dove possibile.



Soluzione:

A	B	C	Z_1=AB	Z_2=B+C	Z_3=AC	$\sim Z_2$	$Z_4=\sim Z_2+Z_3$	**Y=Z_1+Z_4**
0	0	0	0	0	0	1	1	1
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	1	0	1	0	0	0	0
1	0	0	0	0	0	1	1	1
1	0	1	0	1	1	0	1	1
1	1	0	1	1	0	0	0	1
1	1	1	1	1	1	0	1	1

Calcoliamo la forma algebrica partendo dal circuito logico:

Y =	← *Sviluppo la porta P_5*
= (Z_1+Z_4)	← *Sviluppo le porte P_1 e P_4*
= (AB) + ($\sim Z_2$+Z_3)	← *Sviluppo le porte P_2 e P_3*
= AB + ~(B+C) + (AC)	← *5b Commutativa:* **x + y = y + x**
= AB + (AC) + ~(B+C)	← *7a Distributiva* **zx + zy = z(x + y)**
= A(B +C) + ~(B+C)	← *vedi sopra:* **x + ~xy = x + y** ***(*** **x**=*~(B+C),* y=*A* ***)***
= A + ~(B+C)	←*9bDe Morgan* **~(x + y)** = **~x~y**
= A + ~B~C	

Questi sono i due *mintermini della prima forma canonica della funzione ottenuta.*

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

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

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

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

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

= ~A~B~C + A~C + AC

= ~A~B~C + A

= ~A~B~C + A~B ~C + A

= ~B ~C + A

Calcoliamo la seconda forma canonica POS partendo dal risultato precendente:

Y = A + ~B~C

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

= A+A~B +AB+A~C+~B~C

= AA+A~B +BA+B~B +~CA+~C ~B

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

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

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

Esercizio 6: Calcolare una forma algebrica semplificata della seguente tabella. Si ricavi la SOP. Si determinino il cammino critico del circuito corrsipondente alla SOP e quello del circuito semplificato. Avrebbe senso in questo caso utilizzare la POS invece della SOP? Perchè? Come sarebbe possibile utilizzare la porta XNOR per semplificare il circuito derivato dalla POS?

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
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

Soluzione:

La SOP è ~A~B~C~D+~A~BCD+~ABC~D+A~B~CD+AB~C~D+ABCD.
Semplificando si ottiene:

SOP = ~A~B(~C~D+CD)+AB(~C~D+CD)+ ~ABC~D+A~B~CD=
= (~A~B+AB)(~C~D+CD)+(~ABC~D+A~B~CD)

Cammino critico: nel caso della SOP originale, dobbiamo eseguire in parallelo 6 AND a 4 entrate (cammino critico 3), quindi 5 OR (ovvero un OR a 6 entrate, cammino critico 5), quindi in tutto il cammino critico è pari a 3+5=8.
Nel caso della formulazione semplificata, abbiamo un OR che agisce su (~A~B+AB)(~C~D+CD) e su (~ABC~D+A~B~CD). Il primo di questi due termini può essere calcolato con 4 AND paralleli (c.c. 1) + 2 OR paralleli (c.c. 1) + 1 AND, quindi con un c. c. di 3. Il secondo membro richiede due AND a 4 ingressi (c.c. 3) ed un OR (c.c. 1), per un cammino critico totale pari a 4. Il c.c. totale sarà quindi pari a 4+1=5.
In questo caso l'utilizzo della POS sembra poco sensato, in quanto il numero di **0** che compaiono nella Y è maggiore del numero di **1** (quindi la POS avrebbe più termini della SOP).
Le porte XNOR possono essere utilizzate per semplficare il circuito, infatti ~A~B+AB=A XNOR B (considerazioni analoghe per ~C~D+CD e ~ABC~D+A~B~CD).

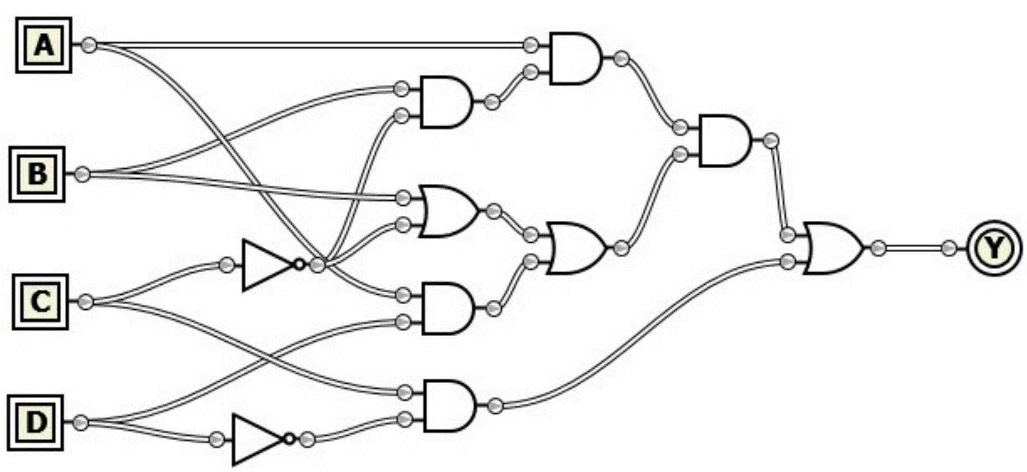
Esercizio 7: Sia Y=AB~C [(B +~C)+(AD)] + C~D una funzione logica. Si ricavi la tabella di verità e la SOP. Si implementino in Gatesim il circuito associato alla formula originale ed il circuito associato alla SOP e li si confrontino. Si proceda poi alla semplificazione algebrica della Y, si implementi il circuito corrispondente e lo si confronti con gli altri due circuiti implementati.

Soluzione:

La tabella di verità:

A	B	C	D	~C	AB~C	B+~C	AD	(B+C)+AD	AB~C[(B+~C)+AD]	~D	C~D	Y
0	0	0	0	1	0	1	0	1	0	1	0	0
0	0	0	1	1	0	1	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	1	1
0	0	1	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	1	0	1	0	0
0	1	0	1	1	0	1	0	1	0	0	0	0
0	1	1	0	0	0	1	0	1	0	1	1	1
0	1	1	1	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	1	0	1	0	1	0	0
1	0	0	1	1	0	1	1	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1	1	1
1	0	1	1	0	0	0	1	1	0	0	0	0
1	1	0	0	1	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	0	1
1	1	1	0	0	0	1	0	1	0	1	1	1
1	1	1	1	0	0	1	1	1	0	0	0	0

Il circuito della funzione originaria è il seguente:

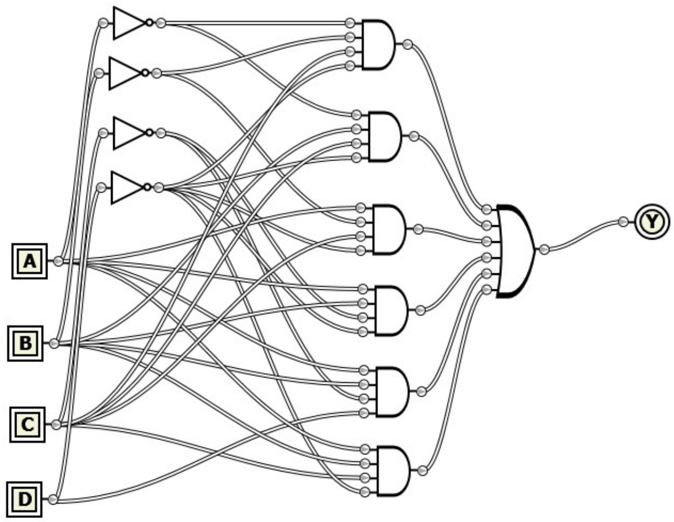


Il cammino critico è 4 e la complessita 8.

La forma SOP è la seguente:

Y = (~A~BC~D)+(~ABC~D)+(A~BC~D)+(AB~C~D)+(AB~CD) +(ABC~D)

Il circuito risultante è il seguente (*Gatesim ha difficoltà a maneggiare circiti cosi complessi, ne segue una certa lentezza nella commutazione delle porte*):

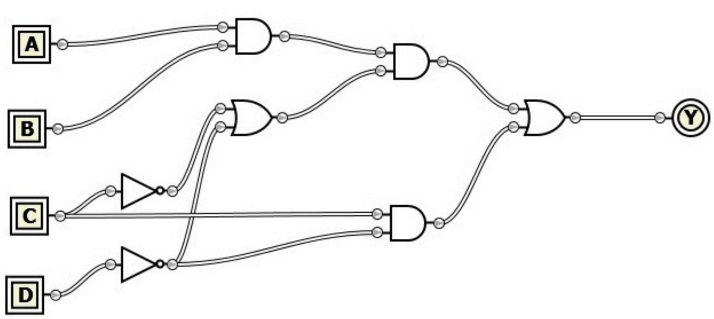


Semplifichiamo la funzione originaria partendo dalla SOP:

Y = (~A~BC~D)+(~ABC~D)+(A~BC~D)+(AB~C~D)+(AB~CD) +(ABC~D)

applico varie volte

Y	=	(~AC~D) +(A~BC~D)+(AB~C~D)+(AB~CD) +(ABC~D)	**xy+x~y=x**
Y	=	(~AC~D) +(A~BC~D)+(ABC~D)+(AB~C~D)+(AB~CD) +(ABC~D)	**x=x+x**
Y	=	(~AC~D) +(AC~D)+(AB~C~D)+(AB~CD) +(ABC~D)	**xy+x~y=x**
Y	=	C~D +AB(~C~D+~CD+C~D)	**xy+x~y=x, xy+xz=x(y+z)**
Y	=	C~D +AB(~C+~D)	**xy+x~y+~xy=x+y**



Il cammino critico è 3 e la complessità 5. Vedi esercizio 3 per la verifica di correttezza.

Massimo Marchi – marchi@dsi.unimi.it

Esercizio 8: Si determini la forma algebrica più semplice per la rappresentazione circuitale della seguente tabella della verità, avendo cura di scegliere il valore delle X in modo ottimale.

A	**B**	**C**	**Y**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	X
1	0	1	0
1	1	0	X
1	1	1	1

La seconda X può essere messa ad 1 – in questo modo le ultime due righe possono essere accorpate nella SOP, dal momento che si genera l'implicante AB (a partire dai due mintermini ABC'+ABC). La prima X può invece essere messa a 0.
La SOP che si ottiene è A'BC+ABC'+ABC, semplificabile in A'BC+AB = B(A'C+A)=B(A+C).
Ma si può fare di meglio…
Mettiamo le due X a 0 ed otteniamo la SOP=A'BC+ABC
Che si può semplificare in
Y=(A'+A)BC=BC.

Esercizio 9: Si determini la forma algebrica più semplice per la rappresentazione circuitale della seguente tabella della verità, avendo cura di scegliere il valore delle X in modo ottimale.

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

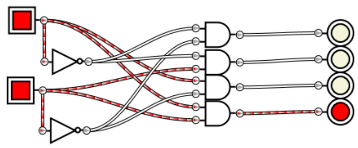
Soluzione → non riportata.

Esercizio 10: Si progetti e si implementi in Gatesim il circuito di un decodificatore a 2 bit (Hint => il decodificatore riceve in ingresso una sequenza di 2 bit e attiva in uscita una delle 4 linee, in particolare quella identificata dalla sequenza di bit in ingresso).
Si utilizzi il decodificatore così creato per creare un multiplexer a 4 vie. => implementare in Gatesim (Hint -> il multiplexer seleziona una delle quattro linee in ingresso e la lascia passare in uscita).

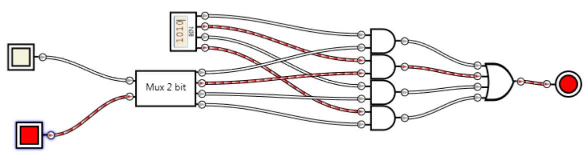
Per il decodificatore, disegniamo la tabella della verità e ricaviamo la SOP.

b1	b2	y1	y2	y3	y4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Quindi: y1=b1’b2’, y2=b1’b2, y3=b1b2’, y4=b1b2.
In Gatesim, il circuito (che salviamo come Mux 4 vie) è:



Utilizziamo ora questo circuito per fare passare uno solo tra quattro segnali definit dall'utente, in modo da realizzare un multiplexer a 4 vie. In particolare, dovremo fare in modo ciascuna uscita del decoder entri in una porta AND insieme al segnale che deve bloccare o lasciare passare (ricordiamo che la porta AND può essere vista come una porta che lascia passare il segnale A quando il bit di controllo B è a 1, non lascia passare B altrimenti). Abbiamo quindi:



Il multiplexer a 2 bit esce con un solo filo attivato, quello corrispondente ai due bit di ingresso (nell'esempio, la configurazione 01 attiva la seconda uscita del mux). Abbiamo poi una stringa di 4 bit, il Mux deve selezionare in questo caso il solo valore del secondo bit... Ogni bit della stringa va in un and insieme alla linea di uscita corrispondente del Mux. Nel caso rappresentato, è attiva la sola seconda linea del mux, quindi passa il secondo bit della stringa. Le uscite delle And vanno in un or che lascia passare il segnale, sia esso zero oppure uno.

Si noti che quella realizzata è in pratica una ROM programmabile. A seconda della configurazione di bit in ingresso, possiamo scegliere se il valore in uscita del circuito è zero oppure uno!!! Nel caso rappresentato in figura, alle configurazioni di bit di ingresso pari a 00, 01, 10 e 11 corrispondono le uscite 0, 1, 0 e 1.

- **Esercizio 11:** Si definisca lo schema di una PLA (in Gatesim) per l'implementazione delle funzioni:

  X=[(not A) or B] and {A or [B and not(C)]}

  Y=[A and B or C] and [not(B)]

  Soluzione → non riportata.