ARCHITETTURA DEGLI ELABORATORI I
ESERCITAZIONE 2 - ESPRESSIONI BOOLEANE
ROBERTO NAVIGLI

1 Espressioni booleane

Le espressioni booleane EB sono definite induttivamente come segue:

i) $0, 1 \in EB$

ii) $\forall v \in Var \; v \in EB$, dove Var è l'insieme delle variabili

iii) se $E_1, E_2 \in EB$ allora $E_1 + E_2$, $E_1 \cdot E_2$, $\overline{E} \in EB$.

Spesso è utile semplificare le espressioni booleane, mediante eliminazione delle parentesi superflue e applicazione degli assiomi dell'algebra di Boole:

- Commutatività:
  i) $x + y = y + x$
  ii) $x \cdot y = y \cdot x$
- Distributività:
  i) $x + (y \cdot z) = (x + y) \cdot (x + z)$
  ii) $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
- Elemento neutro:
  i) $x + 0 = x$
  ii) $x \cdot 1 = x$
- Complemento:
  i) $x + \overline{x} = 1$
  ii) $x \cdot \overline{x} = 0$

Gli esercizi seguenti richiedono di semplificare espressioni booleane, portarle in forma canonica o normale e stabilire l'equivalenza tra espressioni booleane.

1.1 *Data la seguente espressione booleana, semplificarla usando le leggi dell'algebra di Boole:* $x \cdot y + \overline{y} \cdot z + \overline{x} \cdot y$.

Applicando la proprietà distributiva e l'assioma di complementazione, $x \cdot y + \overline{y} \cdot z + \overline{x} \cdot y = (x + \overline{x}) \cdot y + \overline{y} \cdot z = y + \overline{y} \cdot z$.

1.2 *Data la seguente espressione booleana, semplificarla usando le leggi dell'algebra di Boole:* $x \cdot z + \overline{x} \cdot y \cdot z + \overline{x} \cdot \overline{y} \cdot z$.

Applicando più volte la proprietà distributiva, $x \cdot z + \overline{x} \cdot y \cdot z + \overline{x} \cdot \overline{y} \cdot z = x \cdot z + \overline{x} \cdot z \cdot (y + \overline{y}) = x \cdot z + \overline{x} \cdot z = z \cdot (x + \overline{x}) = z$.

1.3 *Data la seguente espressione booleana, semplificarla usando le leggi dell'algebra di Boole:* $\overline{x} \cdot (y \cdot z + y \cdot \overline{z}) + \overline{z} \cdot (\overline{x} \cdot y + \overline{x} \cdot \overline{y}) + z \cdot x \cdot \overline{y}$.

Applicando più volte la proprietà distributiva, $\overline{x} \cdot (y \cdot z + y \cdot \overline{z}) + \overline{z} \cdot (\overline{x} \cdot y + \overline{x} \cdot \overline{y}) + z \cdot x \cdot \overline{y} = \overline{x} \cdot y + \overline{z} \cdot \overline{x} + z \cdot x \cdot \overline{y}$.

1.4 *Data la seguente espressione booleana, semplificarla usando le leggi dell'algebra di Boole:* $\overline{x}\overline{y} + \overline{x}y + xy$.

$\overline{x}\overline{y}+\overline{x}y+xy = \overline{x}(\overline{y}+y)+xy = \overline{x}+xy = (1+y)\overline{x}+xy = \overline{x}+(\overline{x}+x)y = \overline{x}+y$.

1.5 *Data la seguente espressione booleana, semplificarla usando le leggi dell'algebra di Boole:* $\overline{x}y + \overline{y}\overline{z} + xy + \overline{y}z$.

$\overline{x}y + \overline{y}\overline{z} + xy + \overline{y}z = (\overline{x} + x)y + \overline{y}(\overline{z} + z) = y + \overline{y} = 1$

1.6 *Mostrare con un esempio che a una funzione booleana possono essere associate più espressioni booleane e che a una espressione booleana può essere associata una e una sola funzione.*

Sia $f(x, y, z)$ una funzione tale che assume valore 1 se un numero dispari di variabili in ingresso (x, y, e z) vale 1. Negli altri casi la funzione vale 0. Questa funzione può essere rappresentata mediante l'espressione $x \cdot y \cdot \overline{z} + x \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot z$, ma anche con l'espressione $x \cdot (y \cdot \overline{z} + \overline{y} \cdot z) + \overline{x} \cdot y \cdot z$ o ancora $y \cdot (x \cdot \overline{z} + \overline{x} \cdot z) + x \cdot \overline{y} \cdot z$.
Viceversa, data un'espressione booleana, la funzione deriva dalla tabella di verità associata all'espressione, poiché essa fornisce il valore della funzione per ciascuna combinazione di valori delle variabili in ingresso.

1.7 *Dimostrare la seguente uguaglianza tra espressioni booleane:* $x + y\overline{z} = xz + x\overline{z} + xyz + \overline{x}y\overline{z} \cdot (1 + x)$.

L'espressione a sinistra dell'uguaglianza non è ulteriormente semplificabile mediante le regole dell'algebra di Boole. Per l'espressione di destra, invece, si ha: $xz + x\overline{z} + xy\overline{z} + \overline{x}y\overline{z} \cdot (1 + x) = x \cdot (z + \overline{z}) + y\overline{z} \cdot (x + \overline{x} \cdot (1 + x)) = x + y\overline{z} \cdot (x + \overline{x} + x) = x + y\overline{z}$.

1.8 *Tradurre l'espressione booleana $(x + y) \cdot \overline{y}$ in un'altra espressione booleana che faccia uso esclusivamente dell'operatore NAND.*

La tabella di verità degli operatori NOT, OR, AND e NAND è la seguente:

x	y	$\overline{x}$	$x+y$	$x \cdot y$	$xNANDy$
0	0	1	0	0	1
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

Si riscrive l'operatore di complemento (o negazione) $\overline{x} = xNANDx$. Gli altri operatori si riscrivono come segue:

- $x \cdot y = \overline{\overline{(x \cdot y)}} = \overline{xNANDy} = (xNANDy)NAND(xNANDy)$.
- $x + y = \overline{\overline{x+y}} = \overline{\overline{x} \cdot \overline{y}} = \overline{(xNANDx) \cdot (yNANDy)} = (xNANDx)NAND(yNANDy)$.

Riscrivendo l'espressione data si ottiene: $(x + y) \cdot \overline{y} = (x + y) \cdot \underbrace{(yNANDy)}_{\overline{y}} = \underbrace{((xNANDx)NAND(yNANDy))}_{x+y} \cdot \underbrace{(yNANDy)}_{\overline{y}} = (((xNANDx)NAND(yNANDy))NAND(yNANDy))NAND((xNANDx)NAND(yNANDy))NAND(yNANDy))$.

1.9 *Semplificare l'espressione booleana $\overline{x} + xyz + \overline{x}z$ e implementarla con porte NAND.*

$\overline{x} + xyz + \overline{x}z = \overline{x}(1 + z) + xyz = \overline{x} + xyz = \overline{x}(1 + yz) + xyz = \overline{x} + \overline{x}yz + xyz = \overline{x} + (\overline{x} + x)yz = \overline{x} + yz$.

Ricordando le seguenti uguaglianze (cfr. esercizio precedente):

- $\overline{x} = x\ NAND\ x$.
- $x \cdot y = (x\ NAND\ y)\ NAND\ (x\ NAND\ y)$.

- $x+y=(x\ NAND\ x)\ NAND\ (y\ NAND\ y)$.

si ottiene: $\overline{x}+yz = \underbrace{(\overline{x}NAND\overline{x})}_{\overline{x}}+yz = \underbrace{(\overline{x}NAND\overline{x})}_{\overline{x}}+\underbrace{((yNANDz)NAND(yNANDz))}_{y\cdot z} =$
$((xNANDx)NAND(xNANDx))NAND$
$((((yNANDz)NAND(yNANDz))NAND((yNANDz)NAND(yNANDz)))$.

1.10 *Tradurre l'espressione booleana $(x+y)\cdot\overline{z}$ in un'altra espressione booleana che faccia uso esclusivamente dell'operatore NOR.*

La tabella di verità degli operatori NOT, OR, AND e NOR è la seguente:

x	y	$\overline{x}$	$x+y$	$x\cdot y$	$xNORy$
0	0	1	0	0	1
0	1	1	1	0	0
1	0	0	1	0	0
1	1	0	1	1	0

Si riscrive l'operatore di complemento (o negazione) $\overline{x}=xNORx$. Gli altri operatori si riscrivono come segue:

- $x+y=\overline{xNORy}=(xNORy)NOR(xNORy)$.
- $x\cdot y=\overline{\overline{(x\cdot y)}}=\overline{\overline{x}+\overline{y}}=\overline{x}NOR\overline{y}=(xNORx)NOR(yNORy)$.

Riscriviamo l'espressione booleana: $(x+y)\cdot\overline{z}=(x+y)\cdot\underbrace{(zNORz)}_{\overline{z}}=$
$\underbrace{(xNORy)NOR(xNORy)}_{(x+y)}\cdot\underbrace{(zNORz)}_{\overline{z}} = (((xNORy)NOR(xNORy))NOR$
$((xNORy)NOR(xNORy)))NOR((zNORz)NOR(zNORz))$.

2 Da tabelle di verità a forme canoniche o normali

Ricordiamo che data una riga della tabella di verità costituita da una sequenza di n valori booleani $b_1b_2\ldots b_n$, il prodotto logico (AND) dei letterali associati (x_i se b_i assume valore 1, $\overline{x_i}$ se b_i assume valore 0) prende il nome di *mintermine*. Ad esempio, il mintermine della riga 001 è $\overline{x}\cdot\overline{y}\cdot z$ (nel rappresentare un mintermine, il simbolo del prodotto tra letterali $\cdot$ può essere omesso: $\overline{x}\overline{y}z$).
Viceversa, data una riga della tabella di verità $b_1b_2\ldots b_n$, la somma logica (OR) dei letterali negati (x_i se b_i assume valore 0, $\overline{x_i}$ se b_i assume valore 1) prende il nome di *maxtermine*. Ad esempio, il maxtermine della riga 001 è

$x + y + \overline{z}$.
La somma logica (OR) dei mintermini corrispondenti alle righe della tabella di verità in cui la variabile booleana di uscita è 1 prende il nome di *forma canonica disgiuntiva* (FCD) della funzione booleana del circuito.
Il prodotto (AND) dei maxtermini corrispondenti alle righe della tabella di verità in cui la variabile booleana di uscita è 0 prende il nome di *forma canonica congiuntiva* (FCC) della funzione booleana del circuito.
La forma canonica congiuntiva è anche data dalla complementazione della forma canonica disgiuntiva della funzione complemento.
La *forma normale disgiuntiva* (FND o SOP, ovvero *sum of products*) è una generalizzazione della forma canonica disgiuntiva. In una SOP i prodotti, ottenuti applicando le regole dell'algebra booleana, *non sono necessariamente mintermini.*
La *forma normale congiuntiva* (FNC o POS, ovvero *product of sums*) è una generalizzazione della forma canonica congiuntiva. In una POS le somme *non sono necessariamente maxtermini.*

2.1 *Convertire la seguente espressione booleana in forma canonica disgiuntiva (FCD) e in forma canonica congiuntiva (FCC):* $x \cdot \overline{y} + x \cdot y \cdot \overline{z} + \overline{x} \cdot \overline{y} + z$.

La tabella di verità dell'espressione $f = x\overline{y} + xy\overline{z} + \overline{xy} + z$ è la seguente:

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Dalla tabella di verità dell'esercizio si ottiene: $FCD(f) = \overline{xy}z + \overline{x}y\overline{z} + \overline{x}yz + xy\overline{z}$. La FCC si ottiene complementando la FCD di $\overline{f}$: $FNC(\overline{f}) = \overline{xyz} + x\overline{yz} + x\overline{y}z + xyz$. $FCC(f) = \overline{FCD(\overline{f})} = (x + y + z) \cdot (\overline{x} + y + z) \cdot (\overline{x} + y + \overline{z}) \cdot (\overline{x} + \overline{y} + \overline{z})$.

2.2 *Derivare la FCD e la FCC della funzione a tre argomenti che dà 1 sse la somma degli argomenti è pari a 2.*

La tabella di verità della funzione è la seguente:

x	y	z	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	0

Da cui, secondo la definizione di FCD e FCC: $FCD(f) = \overline{x}yz+x\overline{y}z+xy\overline{z}$. $FCC(f) = (x+y+z)\cdot(x+y+\overline{z})\cdot(x+\overline{y}+z)\cdot(\overline{x}+y+z)\cdot(\overline{x}+\overline{y}+\overline{z})$. La forma canonica congiuntiva può anche essere ottenuta dal complemento di $FCD(\overline{f})$: $FCC(f) = \overline{FCD(\overline{f})}) = \overline{\overline{xyz}+\overline{xy}z+\overline{x}y\overline{z}+x\overline{yz}+xyz} = (x+y+z)\cdot(x+y+\overline{z})\cdot(x+\overline{y}+z)\cdot(\overline{x}+y+z)\cdot(\overline{x}+\overline{y}+\overline{z})$.

2.3 *Scrivere almeno due forme normali disgiuntive (FND) e due forme normali congiuntive (FNC) per l'espressione dell'esercizio 2.1 diverse dalle FCD e FCC.*

Nell'esercizio abbiamo che: $y\overline{z}+\overline{xy}z+\overline{x}yz$ e $\overline{x}y+\overline{xy}z+xy\overline{z}$ sono due SOP. $(y+z)\cdot(\overline{x}+y+\overline{z})\cdot(\overline{x}+\overline{y}+\overline{z})$ e $(x+y+z)\cdot(\overline{x}+y)\cdot(\overline{x}+\overline{y}+\overline{z})$ sono due POS.

3 Da espressioni booleane a tabelle di verità

Per ottenere la tabella di verità di un'espressione booleana è necessario trasformare l'espressione in FCD o FCC oppure nelle rispettive forme normali FND o FNC.
Data una FCD o una FND, ovvero in generale una somma di prodotti, e dato un termine t_i della somma, associamo una stringa binaria $R_i = b_1b_2\dots b_n$ costruita come segue:

- se x_i appare in t_i, allora porre $b_i = 1$.
- se $\overline{x_i}$ appare in t_i, allora porre $b_i = 0$.
- se né x_i né $\overline{x_i}$ appaiono in t_i, porre $b_i = X$ (condizione *don't care*).

Costruiamo la tabella di verità ponendo un 1 in corrispondenza di tutte le righe uguali o implicate da R_i e 0 altrimenti.

Analogamente, data una FCD o una FND, ovvero in generale un prodotto di somme, e dato un termine t_i del prodotto, associamo una stringa binaria $R_i = b_1b_2\dots b_n$ costruita come segue:

- se x_i appare in t_i, allora porre $b_i = 0$.
- se $\overline{x_i}$ appare in t_i, allora porre $b_i = 1$.
- se né x_i né $\overline{x_i}$ appaiono in t_i, porre $b_i = X$ (condizione *don't care*).

Costruiamo la tabella di verità ponendo un 0 in corrispondenza di tutte le righe uguali o implicate da R_i e 1 altrimenti.

3.1 *Derivare la tabella di verità dell'espressione booleana:* $xy(z+\overline{x})+\overline{y}(z+x)$.

$xy(z+\overline{x})+\overline{y}(z+x) = xyz+\overline{y}z+x\overline{y} = xyz+(1+x)\overline{y}z+x\overline{y} = xz+\overline{y}z+x\overline{y}$.
Abbiamo una forma normale disgiuntiva, per cui dobbiamo associare un 1 in corrispondenza delle righe $1X1$, $X01$ e $10X$:

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

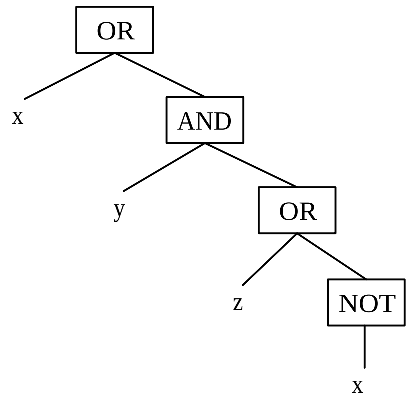
3.2 *Derivare la tabella di verità dell'espressione booleana:* $x(yz+\overline{z})+z(y+x\overline{y})+y(x+\overline{z})$.

$x(yz+\overline{z})+z(y+x\overline{y})+y(x+\overline{z}) = xyz+x\overline{z}+yz+x\overline{y}z+xy+y\overline{z} = xyz+(1+\overline{x})yz+x\overline{z}+y(z+\overline{z})+x\overline{y}z+xy = yz+xy+x\overline{z}+(1+\overline{z}+\overline{x})y+x\overline{y}z = x\overline{z}+y+x\overline{y}z = x\overline{z}+y(1+xz)+x\overline{y}z = x\overline{z}+y+xz = x+y$.
Abbiamo una forma normale disgiuntiva, per cui dobbiamo associare un 1 in corrispondenza delle righe $1XX$, $X1X$:

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

4 Da espressioni booleane a schemi circuitali

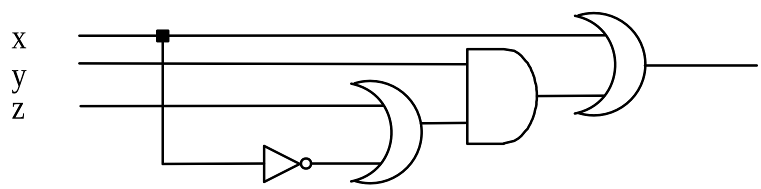
Per ricavare lo schema circuitale di una rete combinatoria a partire da una espressione booleana, conviene ancora partire dalla forma canonica congiuntiva o disgiuntiva, oppure da una sua generalizzazione FNC o FND. Questo tuttavia non è strettamente necessario, se non si desidera ottenere un circuito minimizzato, problema del quale ci occuperemo in fase di studio dei problemi di sintesi.
Assegnata dunque un'espressione booleana, costruiamo una rappresentazione gerarchica degli operatori booleani, partendo dai più esterni. Ad esempio, l'espressione $x + y(z + \overline{x})$ può essere riscritta con le versioni prefisse degli operatori AND, OR e NOT come: $OR(x, AND(y, OR(z, NOT(x))))$. Possiamo anche aiutarci con la rappresentazione ad albero:



Per passare allo schema circuitale è sufficiente considerare che:

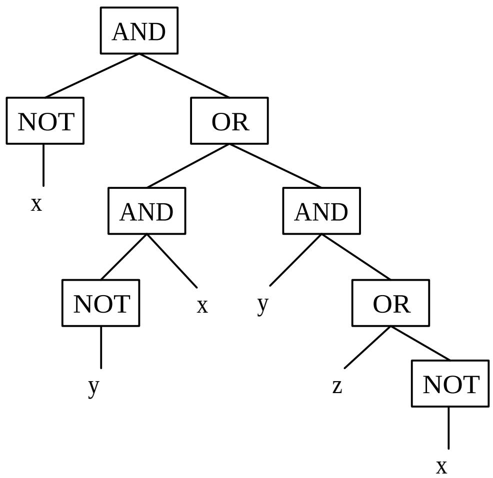
- le foglie dell'albero sono le variabili booleane di ingresso
- i nodi vengono associati alle corrispondenti porte logiche elementari (AND, OR, NOT, ecc.)
- gli archi che collegano i nodi vengono associati alle linee interne della rete.

Lo schema risultante è dunque:

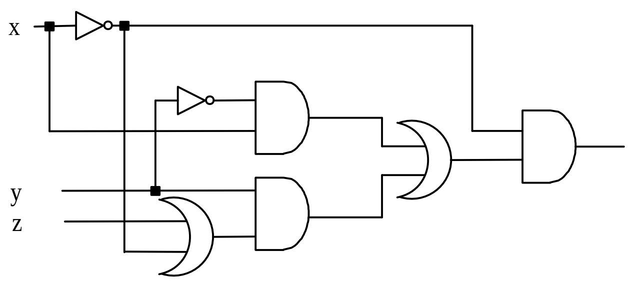


4.1 *Derivare lo schema circuitale dell'espressione booleana:* $\overline{x}(\overline{y}x + y(z + \overline{x}))$.

L'espressione si può riscrivere in forma prefissa come: $AND(NOT(x), OR(AND(NOT(y), x), AND(y, OR(z, NOT(x)))))$. La rappresentazione ad albero è la seguente:

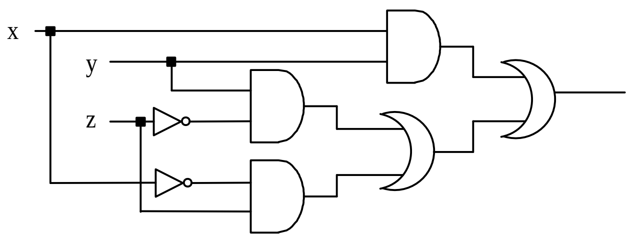


Ne risulta quindi il seguente schema circuitale:



4.2 *Derivare lo schema circuitale dell'espressione booleana:* $xy + y\overline{z} + \overline{x} + z$.

L'espressione si può riscrivere in forma prefissa come: $OR(AND(x, y), OR(AND(y, NOT(z)), AND(NOT(x), z)))$. Lo schema circuitale associato è il seguente:

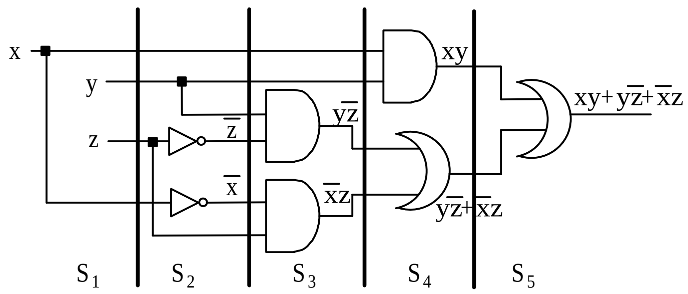


5 Da schemi circuitali a espressioni booleane

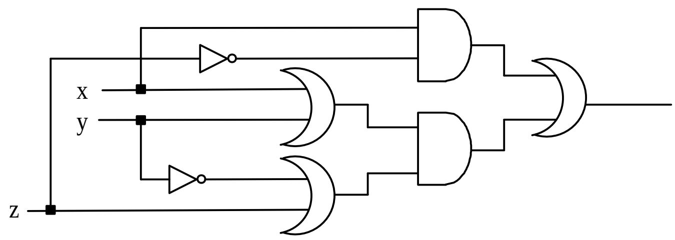
L'obiettivo è di ricavare dallo schema circuitale una rappresentazione in termini di espressione booleana o tabella di verità delle relazioni ingresso/uscita del circuito. Poiché sappiamo come ottenere una tabella di verità dall'espressione booleana (paragrafo 3), ci limiteremo ad introdurre un metodo per ottenere l'espressione booleana dallo schema circuitale. Un possibile procedimento è percorrere il cammino inverso rispetto al metodo introdotto nel paragrafo precedente per ottenere uno schema circuitale dall'espressione booleana:

- Dallo schema circuitale, per ogni linea di uscita, tracciamo il percorso inverso delle linee interne fino a raggiungere le linee di ingresso, ottenendo un albero gerarchico ai cui simboli circuitali corrispondono nodi etichettati, alle linee di collegamento corrispondono gli archi, e le cui foglie, o simboli terminali, sono i segnali di ingresso raggiunti.
- Dall'albero passiamo alla sua forma compatta parentesizzata e quindi costruiamo l'espressione booleana corrispondente, partendo dai simboli logici più interni.

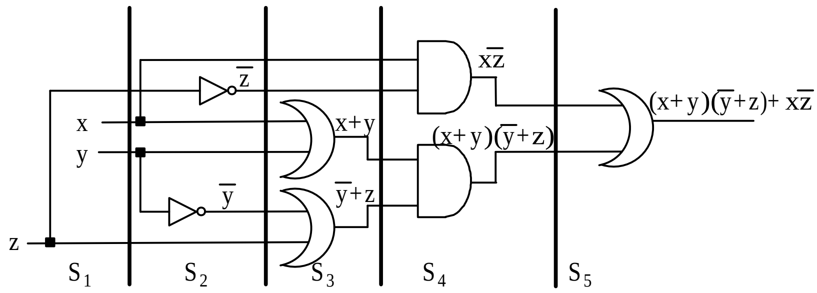
L'alternativa è di suddividere lo schema circuitale in *stadi*. Uno stadio S_i è una sezione verticale del circuito tale che ogni linea di ingresso di uno stadio attraversa al più una porta logica. Le linee di ingresso dello stadio S_1 sono i segnali di ingresso della rete, mentre le linee di uscita dello stadio S_{n-1} sono i segnali di uscita della rete. Il valore di n è pari al numero massimo di porte attraversabili quando si percorre il circuito dalle linee di ingresso fino all'uscita. Le linee di uscita del generico S_i coincidono con le linee di ingresso dello stadio successivo, S_{i+1}. Il metodo consiste, partendo dallo stadio S_1 e procedendo verso destra, nel calcolare le funzioni combinatorie delle uscite di ogni stadio S_i in funzione dei suoi ingressi. Ad esempio, analizziamo lo schema dell'esercizio 4.2:



5.1 *Derivare l'espressione booleana del seguente schema circuitale:*

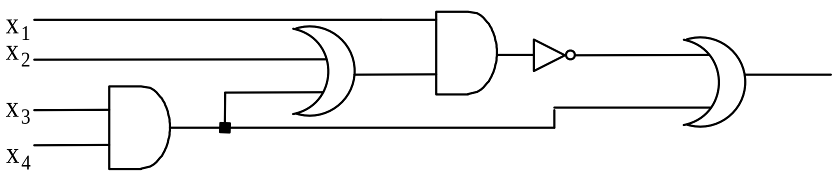


Applicando il metodo degli stadi si ottiene:

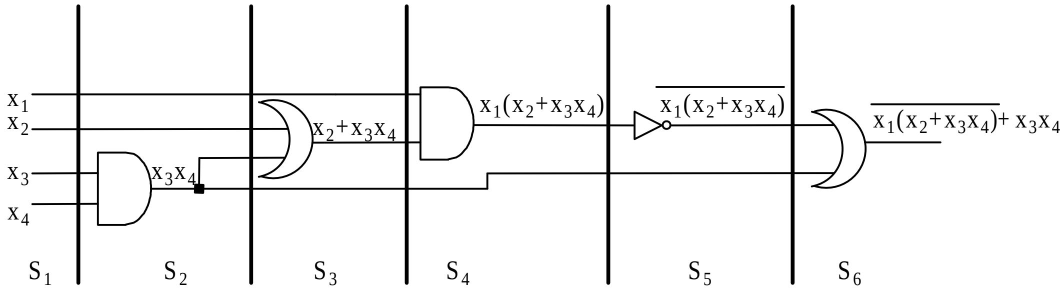


L'espressione è: $(x+y)(z+\overline{y})+x\overline{z}$.

5.2 *Derivare l'espressione booleana in FNC del seguente schema circuitale:*

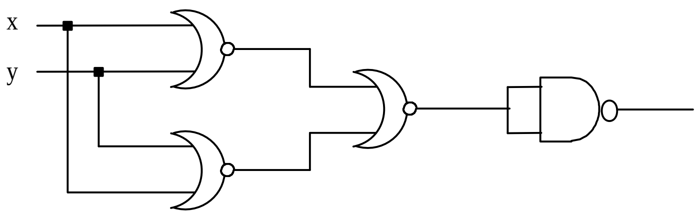


Applicando il metodo degli stadi si ottiene:

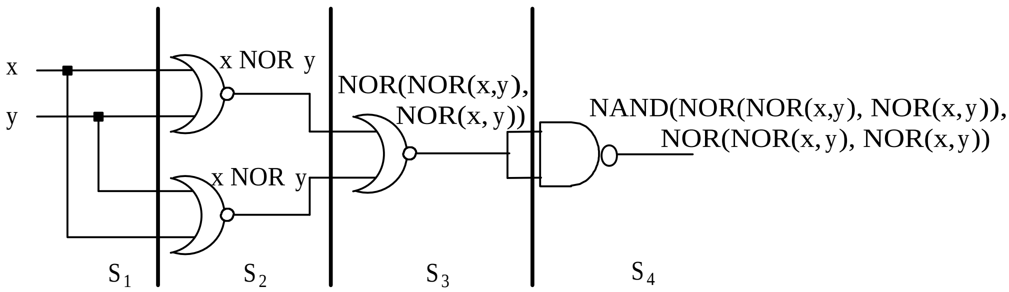


L'espressione booleana associata al circuito è: $\overline{x_1(x_2+x_3x_4)} + x_3x_4$. Portiamola a FNC: $\overline{x_1(x_2+x_3x_4)} + x_3x_4 = \overline{x_1} + (\overline{x_2+x_3x_4}) + x_3x_4 = \overline{x_1} + \overline{x_2}(\overline{x_3x_4}) + x_3x_4 = \overline{x_1} + \overline{x_2}(\overline{x_3} + \overline{x_4}) + x_3x_4 = \overline{x_1} + \overline{x_2}\overline{x_3} + \overline{x_2}\overline{x_4}) + x_3x_4$.

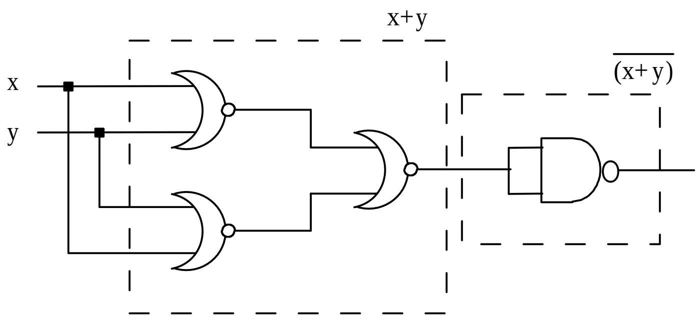
5.3 *Derivare l'espressione booleana del seguente schema circuitale:*



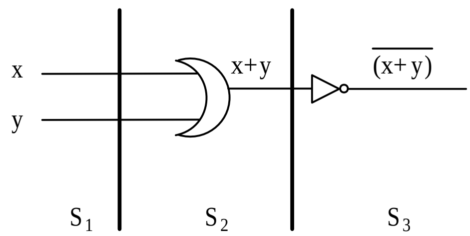
Applicando il metodo degli stadi si ottiene:



E' possibile semplificare l'espressione risultante (cfr. esercizi 1.8, 1.9 e 1.10) sostituendo le porte NOR e NAND: $NAND(\underbrace{NOR(NOR(x,y), NOR(x,y))}_{x+y}, \underbrace{NOR(NOR(x,y), NOR(x,y))}_{x+y}) =$ $NAND(x+y, x+y) = \overline{x+y}$. Si poteva però effettuare la sostituzione direttamente nello schema circuitale, in modo da evitare di determinare un'espressione booleana complicata:



Racchiusi nei rettangoli tratteggiati sono le porte NOR e NAND che realizzano rispettivamente l'OR tra x e y e la porta NOT. Al semplice circuito risultante si può quindi applicare il metodo degli stadi:



ottenendo nuovamente, ma in modo più semplice, l'espressione booleana $\overline{x+y}$.