2^ Esercitazione : Algebra Booleana & Porte Logiche

A) ALGEBRA di BOOLE

(A , + , ·)   dove   + , · : A × A ¾→ A

tali che :

1. *(commutatività)*
   - ♦ $x + y = y + x$
   - ♦ $x \cdot y = y \cdot x$
2. *(distributività)*
   - ♦ $x + (y \cdot z) = (x + y) \cdot (x + z)$
   - ♦ $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
3. *(elemento neutro)*
   - ♦ $x + 0 = x$
   - ♦ $x \cdot 1 = x$
4. *(elemento complementare)*
   - ♦ $x + \overline{x} = 1$
   - ♦ $x \cdot \overline{x} = 0$

Esistono molte leggi derivate da tali assiomi; molto importanti sono le leggi di **De Morgan** :

- ♦ $\overline{(x + y)} = \overline{x} \cdot \overline{y}$
- ♦ $\overline{(x \cdot y)} = \overline{x} + \overline{y}$

Considereremo come insieme di supporto A = { 0 , 1 }

A.1) ESPRESSIONI BOOLEANE

Dato un insieme numerabile di variabili *Var* a valori binari, le espressioni booleane EB su di esso sono definite induttivamente da :

- ♦ *0 , 1* EB
- ♦ *v* *Var* *v* EB
- ♦ se E_1 , E_2 EB allora $E_1 + E_2$, $E_1 \cdot E_2$, $\overline{E_1}$ EB

dove $\overline{E}$ è ottenuta applicando opportunemente le leggi di De Morgan e ricordando che:

- $\overline{0} = 1$ e $\overline{1} = 0$
- ¾ v = v

Esempio 1 : il complemento di $(\overline{x_1} + (\overline{x_2} \cdot x_3)) + 1$ è $(x_1 \cdot (x_2 + \overline{x_3})) \cdot 0$

A.2) FUNZIONI BOOLEANE

f : $\{ 0 , 1 \}^n$ ¾→ $\{ 0 , 1 \}$

rappresentate tramite tabelle (per ognuno dei 2^n possibili input binari esprime il corrispondente output di f).

Teor.: *- per ogni espressione booleana esiste un'unica funzione booleana equivalente*
- per ogni funzione booleana esistono infinite espressioni booleane equivalenti

Esempio 2 : l'espressione booleana $\overline{x_1} + (x_2 \cdot \overline{x_3})$ ha come unica funzione equivalente

x_1	x_2	x_3	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	0

Esempio 3 : consideriamo la funzione booleana che realizza l'AND di due variabili

$f (x , y) \; = \; x \quad y$

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

Tra le altre, essa ammette come espressioni equivalenti le seguenti :
$x \cdot y$, $x \cdot y + 0$, $x \cdot y \cdot 1$, $x \cdot y \cdot (z + \overline{z})$

A.3) FORME CANONICHE

Servono per associare ad ogni funzione un'unica espressione booleana. Sia

$$f \; : \; \{ 0 , 1 \}^n \longrightarrow \{ 0 , 1 \}$$

e si chiami *letterale* una variabile affermata o negata. Due tipi :

(i) Forma Canonica Disgiuntiva : Somma di prodotti di n letterali

Algoritmo :
– rappresenta f in forma tabellare

– per ogni riga della tabella in cui f vale 1 deriva il prodotto di n letterali corrispondente. Esso sarà $z_1 \ldots z_n$ dove

$$z_i = \begin{cases} x_i & \text{se valore di } x_i \text{ nella riga considerata è 1} \\ \overline{x_i} & \text{altrimenti} \end{cases}$$

– somma tutti i prodotti così ottenuti

Esempio 4 : derivare la FCD della funzione che dà 1 *sse* riceve un numero pari di 1 in input (si consideri una funzione booleana triargomentale)

La forma tabellare di f è :

x_1	x_2	x_3	f
0	0	0	1
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 $\quad \mathrm{FCD}(f) = \overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3 + \overline{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x}_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x}_3$

(ii) Forma Canonica Congiuntiva : Prodotto di somme di n letterali

Algoritmo :
- calcola $\mathrm{FCD}(\overline{f})$
  (come l'algoritmo precedente, ma considera le righe con valore 0)
- $\mathrm{FCC}(f) = \overline{\mathrm{FCD}(\overline{f})}$

Esempio 5 : si calcoli la FCC della funzione dell'esercizio precedente

$$\mathrm{FCD}(\overline{f}) = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 + \overline{x}_1 \cdot x_2 \cdot \overline{x}_3 + x_1 \cdot \overline{x}_2 \cdot \overline{x}_3 + x_1 \cdot x_2 \cdot x_3$$

$$\mathrm{FCC}(f) = (x_1 + x_2 + \overline{x}_3) \cdot (x_1 + \overline{x}_2 + x_3) \cdot (\overline{x}_1 + x_2 + x_3) \cdot (\overline{x}_1 + \overline{x}_2 + \overline{x}_3)$$

A.4) FORME NORMALI

Simili alle forme canoniche; sono meno complesse ma non sono uniche.

(i) Forma Normale Disgiuntiva : Somma di prodotti di letterali

(ii) Forma Normale Congiuntiva : Prodotto di somme di letterali

La differenza è che i prodotti o le somme non devono essere costituiti per forza da n letterali ma possono contenerne meno.

Esempio 6 : derivare delle FND della funzione f t.c.

$$f(x_1, x_2, x_3) = 1 \quad sse \quad x_1 = x_2 = 1$$

La forma tabellare di f è :

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Le seguenti espressioni sono FND di f

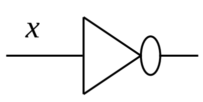
- $x_1 \cdot x_2$
- $x_1 \cdot x_2 \ + \ x_1 \cdot x_2 \cdot \overline{x_3}$
- $x_1 \cdot x_2 \ + \ x_1 \cdot x_2 \cdot x_3$
- $x_1 \cdot x_2 \cdot x_3 \ + \ x_1 \cdot x_2 \cdot \overline{x_3}$

N.B.: l’ultima è la FCD(f)

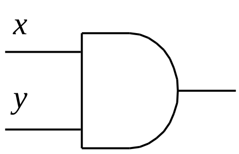
B) PORTE LOGICHE

x	***NOT***
0	1
1	0

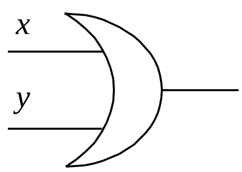
x	y	***AND***	***OR***	***NAND***	***NOR***	***XOR***
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0



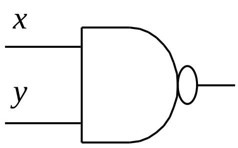
NOT



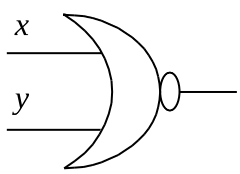
AND



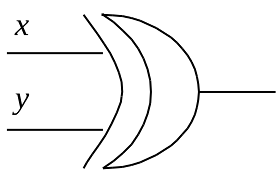
OR



NAND



NOR



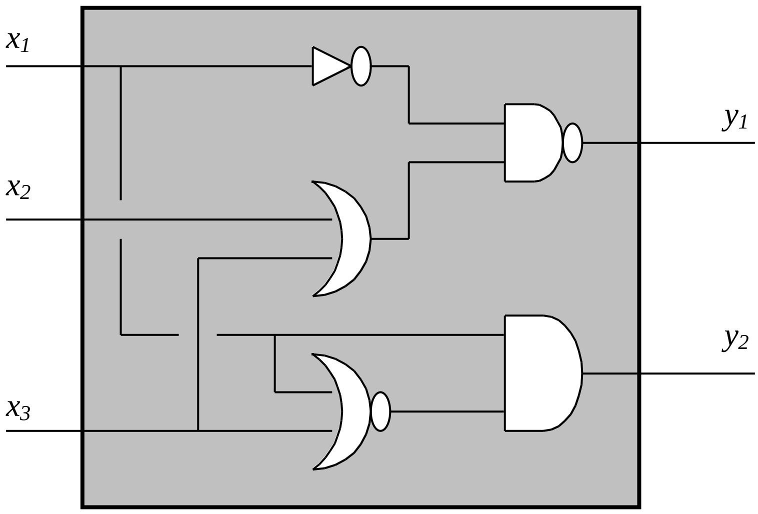
XOR

Una rete logica è un insieme di porte logiche interconnesse in maniera aciclica.

Teor.: *- per ogni rete logica ad m uscite esiste un'unica m-pla di espressioni booleane equivalenti*
- per ogni m-pla di espressioni booleane esiste un'unica rete logica ad m uscite equivalente

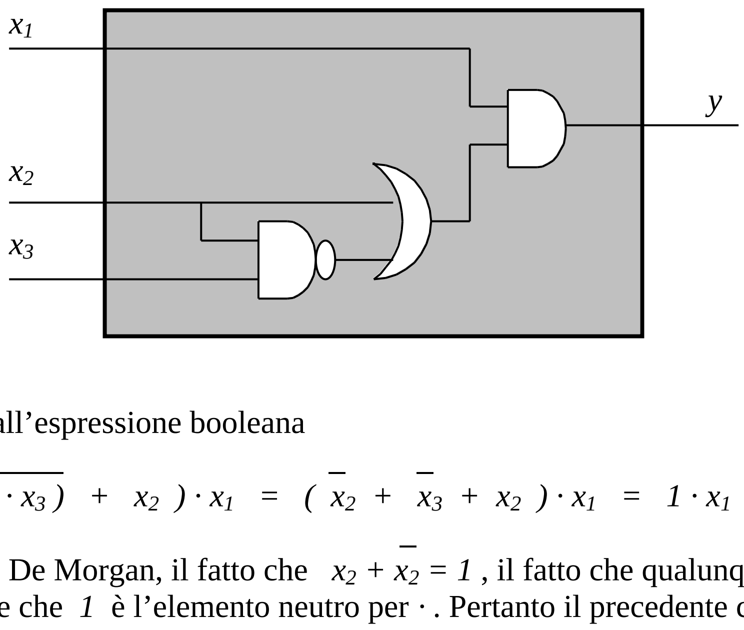
Esempio 7 : la coppia di espressioni $< \overline{\bar{x}_1 \cdot (x_2 + x_3)}\ ,\ \overline{(x_3 + x_1)} \cdot x_1 >$

ha come circuito equivalente



Tramite regole dell'algebra booleana è possibile semplificare reti combinatorie.

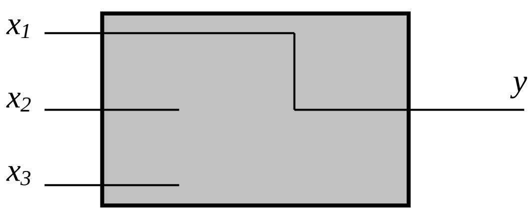
Esempio 8 : il circuito



corrisponde all'espressione booleana

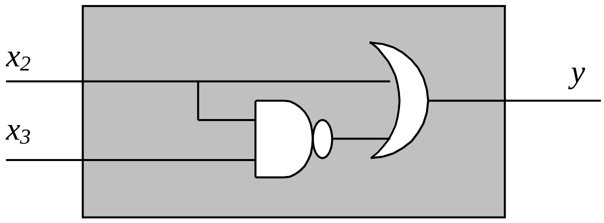
$$(\overline{(x_2 \cdot x_3)} + x_2) \cdot x_1 = (\bar{x}_2 + \bar{x}_3 + x_2) \cdot x_1 = 1 \cdot x_1 = x_1$$

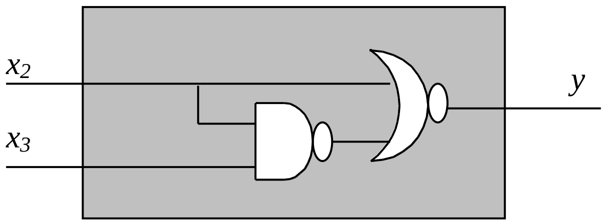
avendo usato De Morgan, il fatto che $x_2 + \bar{x}_2 = 1$, il fatto che qualunque numero sommato ad 1 dà 1 e che 1 è l'elemento neutro per $\cdot$. Pertanto il precedente circuito equivale a



OSS. : è possibile avere reti tautologiche (che ad ogni assegnazione delle variabili di input danno sempre 1) che reti contraddittorie (che ad ogni assegnazione danno sempre 0).

Esempio 9 : le reti seguenti sono rispettivamente tautologica e contraddittoria





B.1) CIRCUITI ALL – NAND E ALL – NAND

Le porte NAND e NOR sono chiamate universali perché, tramite esse, si possono simulare tutte le altre porte.

Esempio 10 :

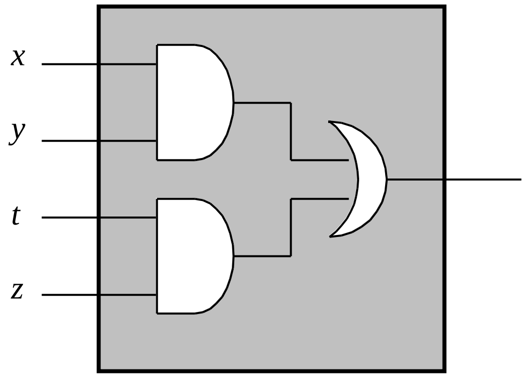
- ***NOT*** *(x)* = ***NAND*** *(x , x)*
  Infatti $x = x \wedge x$ da cui $\neg x = \neg (x \wedge x)$

- ***AND*** *(x , y)* = ***NAND*** *(* ***NAND*** *(x , y)* , ***NAND*** *(x , y))*
  Infatti $x \wedge y = \neg (\neg (x \wedge y)) = \neg (\neg (x \wedge y) \wedge \neg (x \wedge y))$

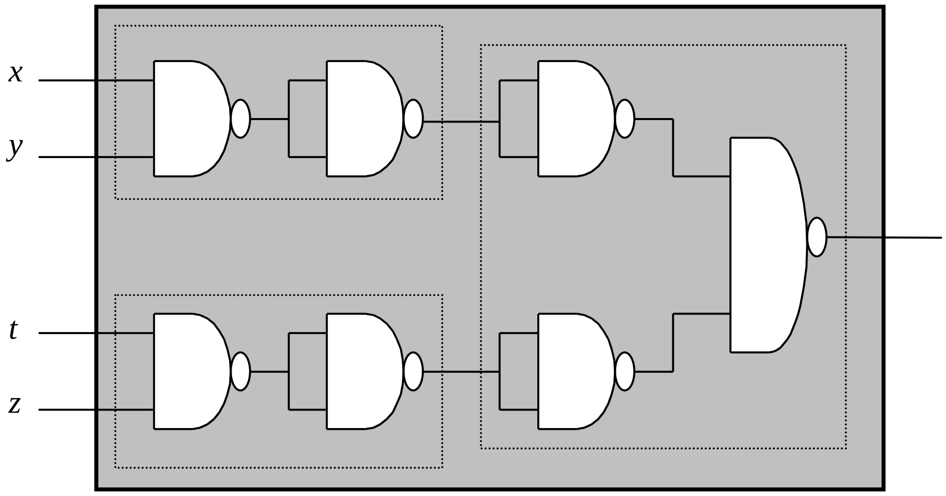
- ***OR*** *(x , y)* = ***NAND*** *(* ***NAND*** *(x , x)* , ***NAND*** *(y , y))*
  Infatti $x \vee y = \neg (\neg x \wedge \neg y) = \neg (\neg (x \wedge x) \wedge \neg (y \wedge y))$

Quindi un qualsiasi circuito può essere realizzato usando solo porte ***NAND*** o porte ***NOR*** .

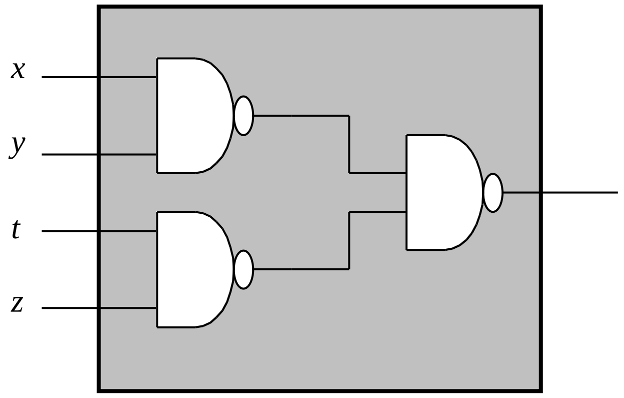
Esempio 11 : trasformare in ALL – NAND il seguente circuito



Sostituendo porta a porta si ottiene il circuito



Osservando però che $(x \textbf{ NAND } x) \textbf{ NAND } (x \textbf{ NAND } x) = x$ [1] si ottiene



[1] Poiché $\neg(\neg(x \wedge x) \wedge \neg(x \wedge x)) = (x \wedge x) \vee (x \wedge x) = x \vee x = x$.

C) ESERCIZI DA SVOLGERE

1 Si dimostrino, facendo uso esclusivamente degli assiomi dell'algebra di Boole, le seguenti leggi:

- *a)* $\overline{x + y} = \bar{x} \cdot \bar{y}$ e $\overline{x \cdot y} = \bar{x} + \bar{y}$
- *b)* $1 + x = 1$
- *c)* *unicità dell'elemento complementare*
- *d)* $x + x = x$ e $x \cdot x = x$ (proprietà di idempotenza)

2 Scrivere la FCD e FCC delle seguenti funzione booleane :

- a) la funzione che dà 1 se la concatenazione dei quattro input x_1 , x_2 , x_3 , x_4 è la rappresentazione binaria di un numero naturale multiplo di 3.
- b) la funzione che ha come input quattro valori binari e dà 1 se riceve almeno tre 1
- c) la funzione che, presi 3 booleani, dà 1 se il numero di zeri è dispari

3 Per ognuna delle funzioni dell'esercizio precedente scrivere almeno due FND e due FNC diverse dalle FCD e FCC.

4 Scrivere i circuiti combinatori per le seguenti espressioni booleane e, se possibile, semplificarli usando regole dell'algebra di Boole:

- *a)* $\overline{(\, x_1 + (\, x_2 \cdot x_3 \,)\,)} + x_2 \cdot (\, \overline{(\, \bar{x}_4 \cdot x_2 \,)} + \bar{x}_3 \,)$
- *b)* $x_1 \cdot \bar{x}_2 \cdot (\, x_1 + \overline{(\, \overline{(\, x_3 \cdot x_2 \,)} + (\, x_1 \cdot x_2 \,)\,)}\,)$
- *c)* $\bar{x}_2 + (\, \overline{(\, x_1 \cdot x_3 \,)} + (\, x_3 \cdot \bar{x}_2 \,)\,) \cdot x_2$

5 Dimostrare come gli operatori ***NOT*** , ***AND*** , ***OR*** , ***NAND*** e ***XOR*** possano venir simulati usando esclusivamente porte ***NOR*** . Dimostrare poi come gli operatori ***NOR*** e ***XOR*** possano venir simulati usando esclusivamente porte ***NAND*** .

6 La regola *(x **NAND** x) **NAND** (x **NAND** x) = x* (usata nell'Esercizio 11) serve per semplificare un circuito ALL – NAND e corrisponde alla regola $\neg (\neg x) = x$. Utilizzando nozioni del corso di Logica Matematica, trovare ulteriori regole di semplificazione per circuiti ALL – NAND e ALL – NOR .

7 Tradurre le seguenti espressioni booleane in circuiti ALL – NAND e in circuiti ALL – NOR ed eventualmente semplificare i risultati ottenuti :

- *a)* $(\, x_1 + x_2 \,) \cdot \bar{x}_2 + x_1$
- *b)* $x_2 \cdot \overline{(\, x_1 + x_3 \,)} \cdot (\, x_2 + \bar{x}_3 \,)$