

L'ALGEBRA BOOLEANA

L'Algebra Booleana è un sistema algebrico formato da "variabili logiche" (variabili booleane) ed "operatori logici" (operatori booleani).

Le *variabili logiche* possono assumere i valori **true** (vero, 1) o **false** (falso, 0) e cambiare il loro stato (ossia il valore che hanno in un certo tempo: da true->>false o da false->>true) nel tempo.

Gli *operatori logici* sono operatori che agiscono sulle variabili logiche come gli operatori aritmetici (+, -, x, /) agiscono sui numeri. Infatti, come un operatore aritmetico che agisce su due numeri produce ancora un numero (4 + 2 = 6, 6 è un numero), così gli operatori logici quando agiscono sulle variabili logiche, producono un valore logico (true o false).

I principali operatori logici sono gli operatori **AND** (detto prodotto logico ed indicato anche con il simbolo '*'), **OR** (detto somma logica ed indicato anche con il simbolo '+') e **NOT** (negazione - indicato anche con il simbolo ! o con un trattino sulla variabile logica).

Siano X1 e X2 due variabili logiche vediamo come agiscono su di esse gli operatori logici e quali risultati (R) restituiscono.

AND		
X1	X2	R
1	1	1
1	0	0
0	1	0
0	0	0

OR		
X1	X2	R
1	1	1
1	0	1
0	1	1
0	0	0

NOT	
X1	R
1	0
0	1

1=true, 0=false.

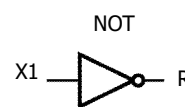
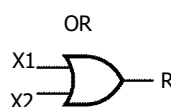
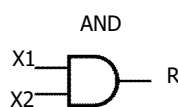
Le Tabelle sopra indicate prendono il nome di **Tabelle di Verità**.

L'operatore NOT, come si può notare, è un *operatore a singolo elemento* (agisce su una singola variabile logica).

Le variabili logiche vengono combinate in espressioni logiche del tipo: X1 AND X2 OR X3.

Per negare una variabile si usa anche porre sulla variabile una linea : es. $\overline{X1}$

In elettronica gli operatori logici vengono implementati in circuiti e prendono il nome di porte logiche:



Le porte logiche ricevono in input X1 ed X2 e restituiscono in output il risultato R in accordo con le tavole di verità sopra indicate.

Combinando gli operatori logici si ottengono gli operatori NAND, NOR e XOR.

NAND (NOT AND) altro non è che la tabella AND negata, così come NOR (NOT OR) è la tabella OR negata. XOR viene definito OR esclusivo: il risultato è vero se i valori delle variabili logiche X1 e X2 sono complementari tra loro (se X1 è vero X2 deve essere falso e viceversa).

Le tavole di verità di NAND, NOR ed XOR:

NAND		
X1	X2	R
1	1	0
1	0	1
0	1	1
0	0	1

NOR		
X1	X2	R
1	1	0
1	0	0
0	1	0
0	0	1

XOR		
X1	X2	R
1	1	0
1	0	1
0	1	1
0	0	0

Abbiamo visto che l'operatore NOT è l'operatore di negazione che mi cambia il valore di una variabile booleana: se la variabile è vera, negandola diventa falsa, se falsa diventa vera.

Ma cosa succede se neghiamo un'espressione logica in cui compaiono variabili booleane ed operatori logici?

TEOREMA DI DE MORGAN

$$1. \overline{(X1 \text{ AND } X2)} = \overline{X1} \text{ OR } \overline{X2}$$

$$2. \overline{(X1 \text{ OR } X2)} = \overline{X1} \text{ AND } \overline{X2}$$

$$3. \overline{\overline{X1}} = X1$$

Le equivalenze dettate dal teorema di De Morgan sono utili per effettuare eventuali semplificazioni nelle espressioni logiche.

FUNZIONI LOGICHE

Sono descrivibili con le espressioni logiche. Facciamo un esempio.

Per effettuare la somma tra due bit sappiamo che si usano le regole:

A B R S

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = \mathbf{1}0 \text{ (}\mathbf{1} \text{ è il riporto che chiameremo } \mathbf{R}\text{)}$$

Bene questa somma può essere fatta utilizzando le espressioni logiche che restituiscono un risultato (funzione logica), per semplificare la scrittura, indichiamo l'operatore **AND** con il simbolo *, per indicare **OR** usiamo il simbolo + e per indicare NOT antepoiamo alla variabile logica il punto esclamativo !

Come funzione logica, abbiamo che la somma tra due bit (A e B) sarà:

$$\mathbf{S} = A * !B + !A * B$$

per il riporto:

$$\mathbf{R} = A * B$$

Infatti se $A = 1$ e $B = 1$ ci aspettiamo che la somma sia **10** ove **1** è il riporto **R**, 0 è **S**. Vediamo se è vero.

$$\mathbf{S} = 1 * !1 + !1 * 1 = 1 * 0 + 0 * 1 = 0 + 0 = \mathbf{0}$$

$$\mathbf{R} = 1 * 1 = \mathbf{1}$$

Se volessimo costruire il sommatore per eseguire la somma di due bit con le porte logiche:

