

Architettura degli Elaboratori - Porte logiche e circuiti combinatori

Andrea Malvezzi

26 Settembre, 2024

Contents

1	Algebra di Boole	3
1.1	Espressioni booleane	3
1.2	Proprietà dell'algebra di Boole	3
1.2.1	La legge di De Morgan	4
1.2.2	Esempio di applicazione della legge di De Morgan . . .	4
1.3	Formula canonica	4
1.3.1	Esempio di formula canonica	5
2	I transistor	6
2.1	La porta NOT realizzata tramite transistor	7
2.2	La porta NAND realizzata tramite transistor	7

1 Algebra di Boole

1.1 Espressioni booleane

Un'espressione booleana si costruisce usando:

- 0 e 1 (False e True);
- gli operatori booleani (o logici);
- delle variabili sempre con valore 0 oppure 1.

1.2 Proprietà dell'algebra di Boole

Nella tabella seguente sono presenti delle equivalenze per descrivere le proprietà dell'algebra di Boole.

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Figure 1: Una visualizzazione delle proprietà dell'algebra di Boole.

1.2.1 La legge di De Morgan

La più importante tra le leggi presentate è sicuramente quella di De Morgan, in quanto permette di passare da una colonna della tabella all'altra in modo semplice e veloce.

1.2.2 Esempio di applicazione della legge di De Morgan

- Per cominciare, scriviamo la OR form della legge inversa: $A + \bar{A} = 1$;
- Seguentemente occorre pensare a com'è scritta l'espressione: siamo davanti ad una OR tra due variabili A, di cui una negata, il tutto pari ad 1;
- Ora osserviamo la legge di De Morgan nella forma OR. Questa afferma quanto segue: La negazione di una OR equivale ad una AND con entrambi gli input negati;
- Appliciamo quindi De Morgan: $\overline{A + A} = 1$ diventerà $\bar{A}\bar{A} = \bar{1}$, ovvero $\bar{A}\bar{A} = 0$.

Ed ecco mostrato come passare da un lato all'altro della tabella tramite la formula di De Morgan.

1.3 Formula canonica

Una funzione booleana si può definire attraverso un'espressione basata solamente sulla AND, la OR e la NOT.

Inoltre, una funzione booleana è esprimibile in una forma detta "canonica". Per ricavarla occorre:

- identificare tutte le combinazioni per cui la funzione in esame è vera (queste son dette **mintermini**);
- fare la OR dei mintermini trovati;

1.3.1 Esempio di formula canonica

Analizziamo la seguente tabella:

A	B	C	F	
0	0	0	0	$\overline{A} \overline{B} \overline{C}$
0	0	1	0	$\overline{A} \overline{B} C$
0	1	0	1	$\overline{A} B \overline{C}$
0	1	1	0	$\overline{A} B C$
1	0	0	0	$A \overline{B} \overline{C}$
1	0	1	0	$A \overline{B} C$
1	1	0	1	$A B \overline{C}$
1	1	1	1	$A B C$

Figure 2: Nella tabella presentata si hanno 3 mintermini: $\overline{A}B\overline{C}$, $AB\overline{C}$, ABC .

Ora dobbiamo fare la OR tra i 3 mintermini trovati precedentemente:

$$\overline{A}B\overline{C} + AB\overline{C} + ABC$$

Questa espressione equivale alla forma canonica della funzione studiata.

2 I transistor

Un transistor è un dispositivo a 3 connettori: **collettore**, **emettitore** e **base**.

Quando non c'è tensione sulla base, il componente agisce come una resistenza infinita tra emettitore e collettore.

In caso contrario, si comporta da conduttore ideale.

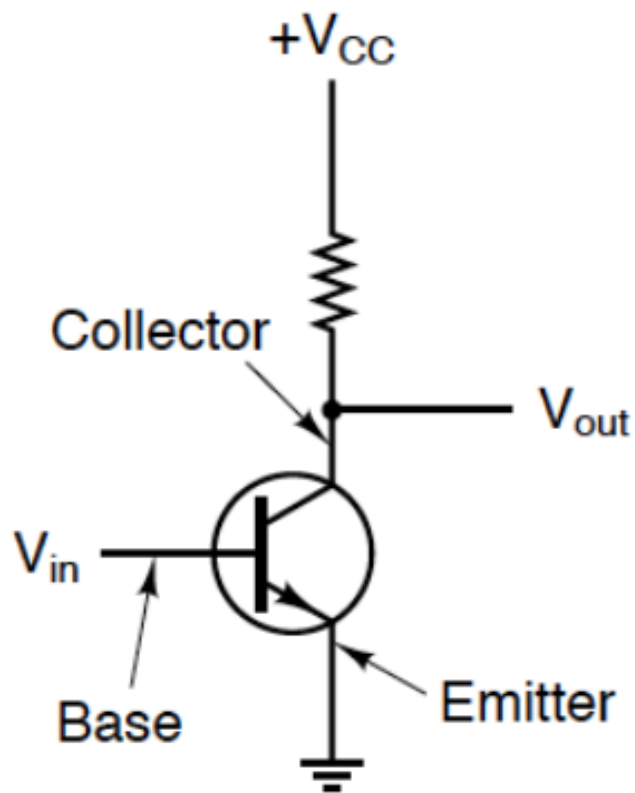


Figure 3: Esempio di porta NOT realizzata con transistor.

2.1 La porta NOT realizzata tramite transistor

La porta NOT della figura (3) funziona nella seguente maniera:

- quando l'ingresso V_{in} è *alto*, la corrente V_{cc} attraversa il transistor e lascia V_{out} basso (Input 1, output 0);
- quando l'ingresso V_{in} è *basso*, la corrente V_{cc} non riesce ad attraversare il transistor, il quale agisce da resistenza infinita, passando quindi per V_{out} , il quale diventa alto (Input 0, output 1).

2.2 La porta NAND realizzata tramite transistor