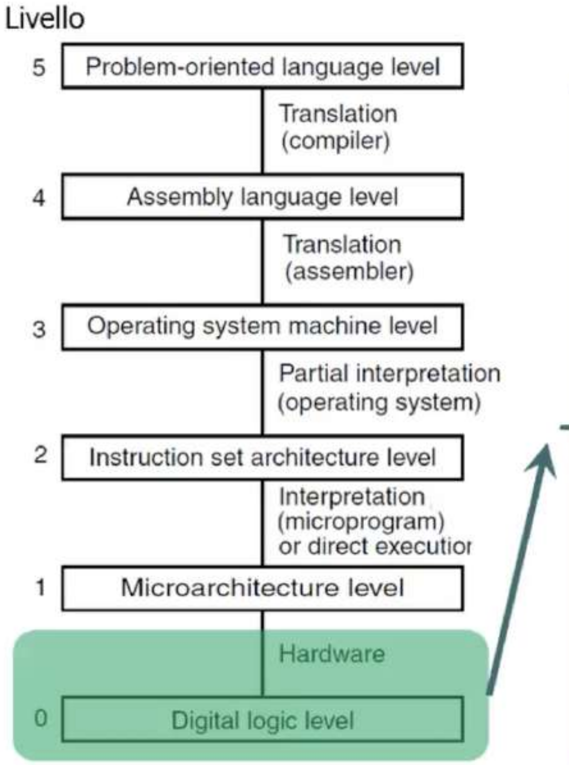
Argomenti

- Il livello logico digitale:
  - algebra di Boole;
  - le trasformazioni nel dominio di Boole.
- Circuiti logici digitali elementari:
  - circuiti Integrati;
  - circuiti combinatori e sequenziali;
  - circuiti combinatori:
    - Multiplexer;
    - Decoder;
    - Comparatori;
    - Programmable Logic Arrays.
  - Circuiti Aritmetici:
    - Shifter;
    - Adder;
    - Arithmetic Logic Units.
  - Clock.

Obiettivi

- Conoscere il dominio di Boole e le operazioni sul dominio.
- Comprendere il funzionamento dei circuiti logici digitali.
- Analizzare il comportamento dei circuiti combinatori elementari.
- Studiare i circuiti che permettono di eseguire calcoli aritmetici ed operazioni logiche.

Il livello logico digitale



- Il livello logico digitale è l'hardware del calcolatore.
- È fatto di piccoli dispositivi elettronici chiamati porte logiche che lavorano con l'albegra di Boole.
- Utilizzando questi piccoli mattoncini si possono comporre funzioni complesse come sommatori, shifter, memorie,...

Algebra di Boole

- George Boole (1815–1864) è il matematico che ha ideato quest'algebra basata su un insieme (B) di due valori: true e false.
- Per convenzione al valore logico true corrisponde il simbolo 1 e a false il simbolo 0.

- Sul dominio B, si possono definire delle funzioni booleane:

funzioni unarie: $B \longrightarrow B$

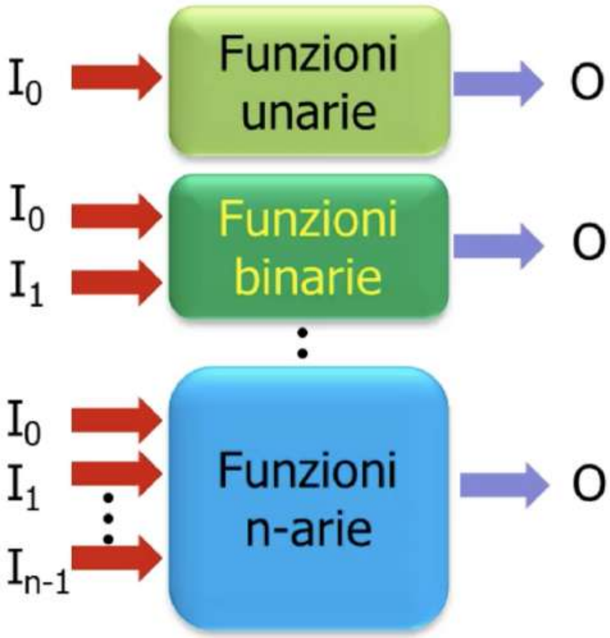
funzioni binarie: $B \times B \longrightarrow B$

…

funzioni n-arie: $\underbrace{B \times \ldots \times B}_{n} \longrightarrow B$

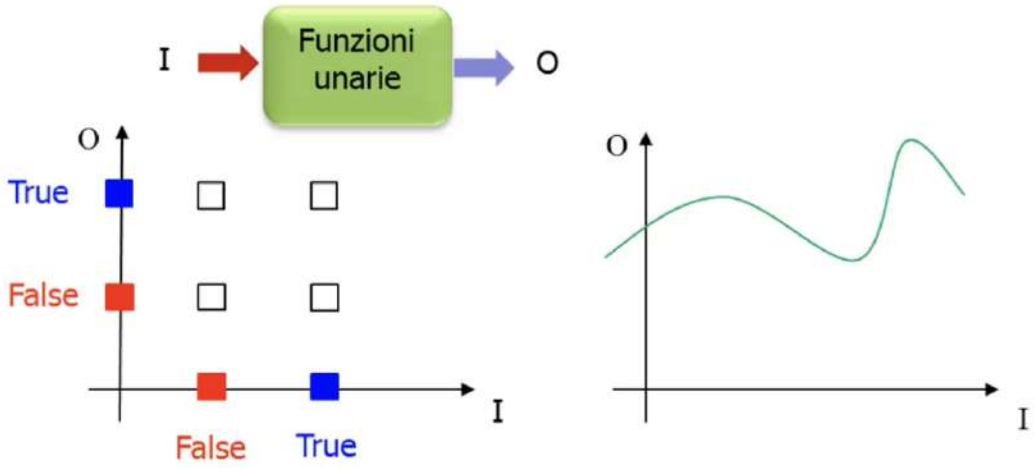
Funzioni Booleane come scatole nere

- Si possono immaginare le funzioni logiche come delle scatole nere che ricevono in ingresso e restituiscono in uscita variabili logiche.



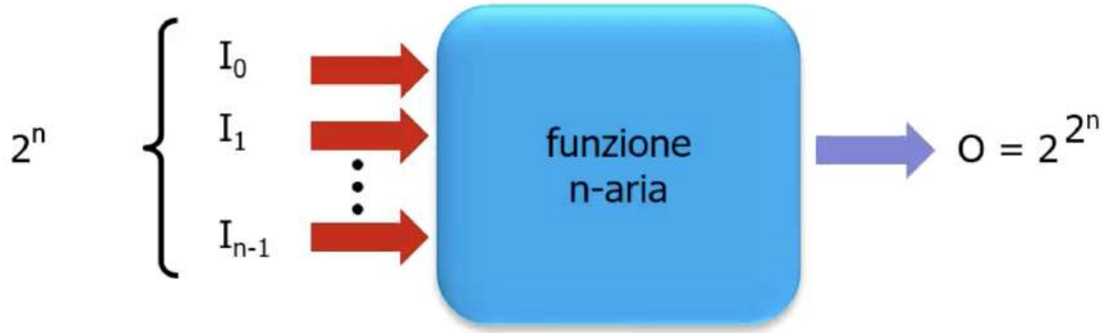
Confronto tra funzioni booleane e funzioni reali

- Il comportamento di una funzione booleana è semplice da descrivere, perché gli ingressi/uscite variano in un insieme finito di valori.
- La **tavola di verità** è lo strumento che permette di esplorare esaustivamente tutte le possibili combinazioni.



L'insieme delle possibili funzioni booleane

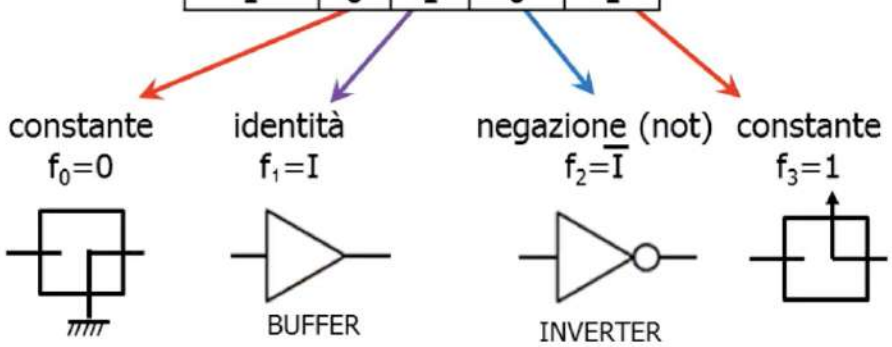
- Con n ingressi si hanno 2^n combinazioni che originano 2^{2^n} possibili funzioni.



Unary Boolean functions

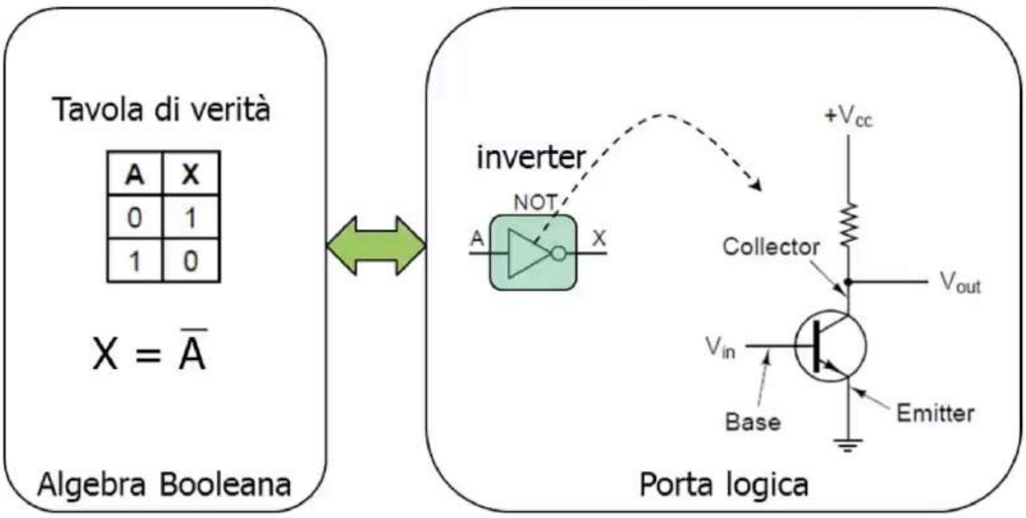
- Le funzioni ad un solo operando sono $2^2=4$.

input	funzioni			
I	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1



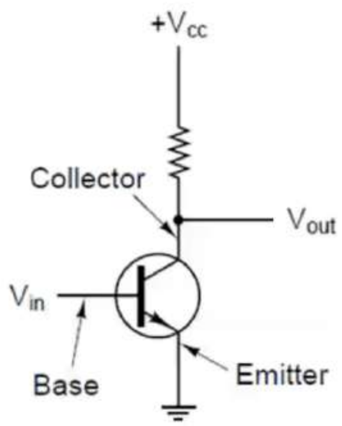
Negation (NOT operator)

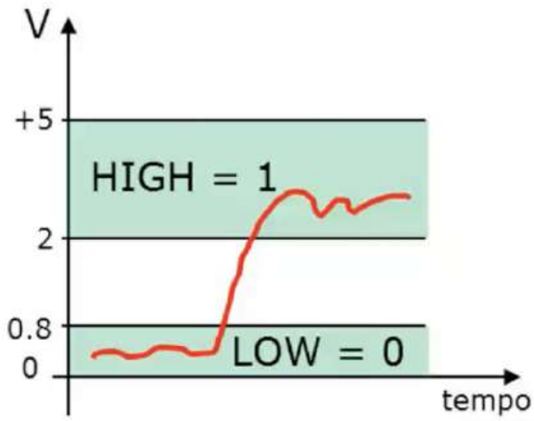
- Esiste una stretta correlazione tra algebra booleana e circuiti digitali.
- La negazione è una funzione che inverte il valore della variabile in ingresso.



Segnali digitali

- La relazione che sussiste tra livelli logici digitali e reali segnali analogici:

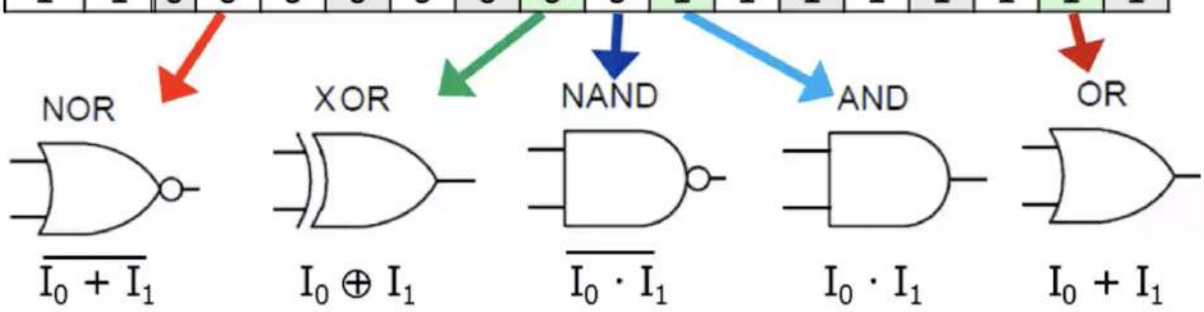




Livelli logici digitali dei TTL

L'insieme delle possibili funzioni booleane binarie

ingressi		funzioni															
I_0	I_1	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1



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	$A\overline{A} = 0$	$A + \overline{A} = 1$
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	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Come verificare la validità di un'equivalenza?

$$A + A\,B \stackrel{?}{=} A$$

- Utilizzando il metodo algebrico attraverso le proprietà:

$$A + A\,B = A\,(\,1 + B\,) = A$$

- Utilizzando la tavola di verità con il metodo esaustivo:

A	**B**	AB	A + AB	A
0	0	0	0	0
0	1	0	0	0
1	0	0	1	1
1	1	1	1	1

Le due espressioni sono identiche per ogni combinazione degli ingressi

- Utilizzando la logica e le definizioni: la variabile A è già presente a sinistra dell'operatore OR perciò la sua presenza nel gruppo di AND ha lo scopo di ridurre B ad A (quindi non aggiunge nulla a B).

Come verificare la validità di un'equivalenza?

$$A + \overline{A}\,B \stackrel{?}{=} A + B$$

- Utilizzando il metodo algebrico attraverso le proprietà (distributiva):

$$A + \overline{A}\,B = (A + \overline{A})(A + B) = A + B$$

- Utilizzando la tavola di verità con il metodo esaustivo:

A	**B**	$\overline{A}$	$\overline{A}\,B$	$A + \overline{A}B$	$A+B$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Le due espressioni sono identiche per ogni combinazione degli ingressi

- Utilizzando la logica e le definizioni: la variabile A è già presente a sinistra dell'operatore OR perciò la sua presenza nel gruppo di AND in forma negata ha lo scopo di ridurre B ad A negato (quindi se A è true l'espressione è true mentre se A è falsa il risultato è B).

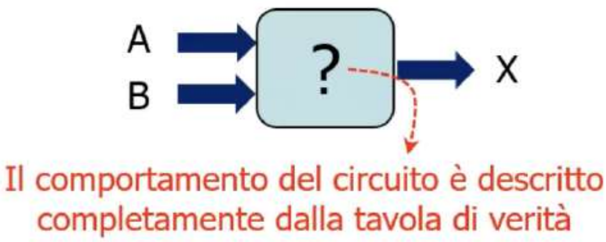
Le trasformazioni nel dominio di Boole

- Per ogni espressione logica c'è un circuito digitale equivalente e una colonna della tavola di verità.
- Per ogni colonna della tavola di verità c'è una espressione che la rappresenta e un corrispondente circuito digitale.
- Per ogni circuito digitale c'è una espressione che lo descrive ed una corrispondente colonna della tavola di verità.
- Ora si analizzeranno la trasformazione di una colonna della tavola di verità nella corrispondente espressione logica.

Tavola di verità → espressione logica

- Si vuole tradurre una colonna (X) della tavola di verità in funzione delle variabili A e B, cioè X = F (A , B).

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



- Si considerano il minor numero di valori 0 oppure 1:

True 1: X è composta da un insieme di gruppi AND che contengono tutte le variabili (negate quelle che hanno il valore 0 nella riga corrispondente), legati con OR.

False 0: X è composta da un insieme di gruppi OR che contengono tutte le variabili (negate quelle che hanno il valore 1 nella riga corrispondente), legati con AND.

Tavola di verità → espressione logica (1)

- X è composto da tanti gruppi AND quanti sono i valori **true** nella colonna della tavola di verità.
- Ogni gruppo di AND contiene tutte le variabili in ingresso, sono negate quelle che sono **false** nella riga corrispondente.

A	**B**	X
0	0	0
0	1	1
1	0	1
1	1	0

$$\overline{A} \cdot B$$

$$A \cdot \overline{B}$$

$$X = \overline{A} \cdot B + A \cdot \overline{B}$$

Tavola di verità → espressione logica (0)

- X è composto da tanti gruppi OR quanti sono i valori **false** nella colonna della tavola di verità.
- Ogni gruppo di OR contiene tutte le variabili in ingresso, sono negate quelle che sono **true** nella riga corrispondente.

A	**B**	X
0	0	0
0	1	1
1	0	1
1	1	0

$$A + B$$

$$\overline{A} + \overline{B}$$

$$X = (A + B) \cdot (\overline{A} + \overline{B})$$

- ...so any Boolean function can be implemented using NOT, AND, and OR gates.

Gli operatori universali

- Abbiamo visto come ogni funzione logica booleana (una colonna della tavola di verità) si possa realizzare con gli operatori AND, OR e NOT
- È facile dimostrare che questi tre operatori possano essere realizzati con una sola porta NAND o NOR.
- Quindi ogni funzione logica booleana può essere realizzata con un solo operatore detto **universale**.

	AND	OR	NOT
NAND			
NOR			

Gli operatori universali

- Tutte le porte logiche possono essere realizzate utilizzando dolo porte NAND o NOR.
- La presenza di operatori universali è molto importante per la costruzione dei circuiti digitali.

	AND	OR	NOT
NAND	$\overline{\overline{A\cdot B}\cdot\overline{A\cdot B}} = \overline{\overline{A\cdot B}} = A\cdot B$	$\overline{\overline{A\cdot A}\cdot\overline{B\cdot B}} = \overline{\overline{A}\cdot\overline{B}} = A+B$	$\overline{A\cdot A} = \overline{A}$
NOR	$\overline{\overline{A+A}+\overline{B+B}} = \overline{\overline{A}+\overline{B}} = A\cdot B$	$\overline{\overline{A+B}+\overline{A+B}} = \overline{\overline{A+B}} = A+B$	$\overline{A+A} = \overline{A}$