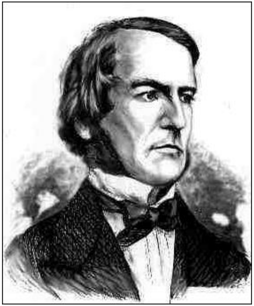
Aritmetica dei Calcolatori 2

Architettura degli Elaboratori e Laboratorio

6 Marzo 2013

Gottfried Wilhelm Leibniz

George Boole



Proprietá dell'algebra di Boole

Identità:	$A + 0 = A$	$A \cdot 1 = A$
Nullo:	$A + 1 = 1$	$A \cdot 0 = 0$
Idempotente:	$A + A = A$	$A \cdot A = A$
Inverso:	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
Commutativa:	$A + B = B + A$	$A \cdot B = B \cdot A$
Associativa:	$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributiva:	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
DeMorgan:	$\overline{A + B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

Forme Canoniche e Semplificazione di una funzione

- Data una funzione booleana, le forme canoniche sono due:
  - Prima forma canonica - Somma di Prodotti (SOP)
    - Mintermini
    - La funzione vale 1 nella tabella di veritá
  - Seconda forma canonica - Prodotto di Somme (POS)
    - Maxtermini
    - La funzione vale 0 nella tabella di veritá
- Data una funzione booleana esistono una ed una sola forma canonica SOP ed una e una sola forma POS che la rappresenta.

- Usando le proprietá dell'algebra di Boole, é possibile semplificare una funzione espressa in forma canonica in una funzione equivalente.

Operazioni logiche bit a bit

NOT

IN	OUT
A	NOT *A*
0	1
1	0

AND

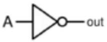
IN		OUT
A	*B*	*A* AND *B*
0	0	0
0	1	0
1	0	0
1	1	1

OR

IN		OUT
A	*B*	*A* OR *B*
0	0	0
0	1	1
1	0	1
1	1	1

XOR

IN		OUT
A	*B*	*A* XOR *B*
0	0	0
0	1	1
1	0	1
1	1	0

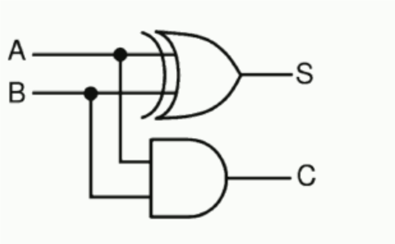


In C: $\sim$ & $|$ $\wedge$ operano bit a bit!

Half Adder

Implementiamo la somma bit a bit:

IN		OUT	
A	*B*	*C*	*S*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

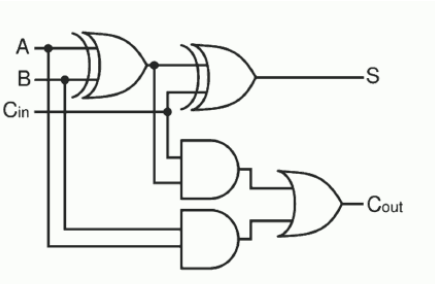


Half Adder riceve due bit A e B in input, calcola il bit di somma S e il bit di riporto C.

Full Adder

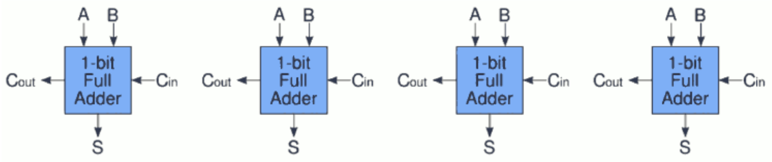
Ora, consideriamo il bit di riporto C_{in}, calcolato da una somma precedente:

IN			OUT	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Full Adder riceve tre bit A, B e C_{in} in input, calcola il bit di somma S e il bit di riporto C_{out}.

Somma binaria



Per ottenere un sommatore a N bit, basta collegare N Full Adder: in questo modo si esegue la classica somma in "colonna".

Esempio: Funzione di maggioranza

Si consideri una funzione combinatoria dotata di 3 ingressi A , B e C , e di un'uscita F , funzionante come segue:

- Se la maggioranza degli ingressi vale 0 , l'uscita vale 0
- Se la maggioranza degli ingressi vale 1 , l'uscita vale 1

A	B	C	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	1

$F(A, B, C) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$ in Mintermini
e in Maxtermini?

Esempio: Semplificazione della funzione di maggioranza

$F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$
Per la proprietà dell'idempotenza:
$F = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$
Per la proprietà distributiva: $F = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$
Per la proprietà dell'inverso: $F = BC1 + AC1 + AB1$
Per la proprietà dell'identità: $F = BC + AC + AB$