Corso di Architettura degli Elaboratori e Laboratorio (M-Z)

Esempio ALU

Nino Cauli

UNIVERSITÀ
degli STUDI
di CATANIA

Dipartimento di Matematica e Informatica

Addizionatore ad 1 bit

Per **SOMMARE** numeri binari ad 1 bit:

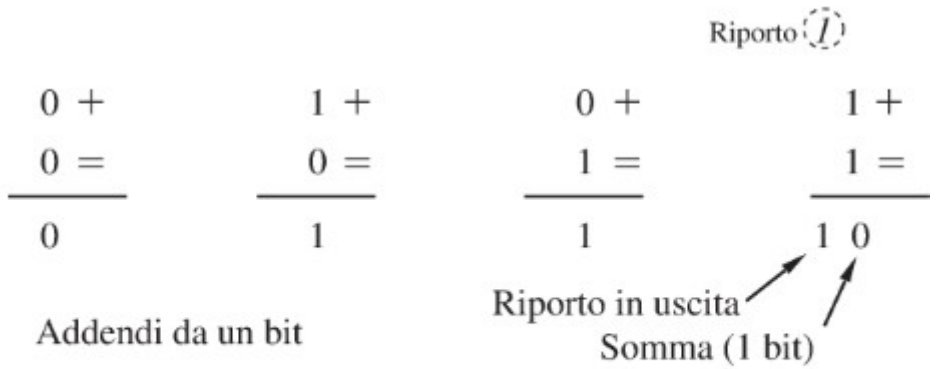


Figura 1.4 - Addizione di numeri a un bit

Il **RIPORTO IN USCITA** della cifre precedente viene assegnato come **RIPORTO IN ENTRATA** alla successiva

Addizionatore ad 1 bit

- Un addizionatore tra due singoli bit può essere espresso da 2 funzioni logiche a tre ingressi (i due bit da sommare più il riporto in ingresso):
  - La prima calcola la somma tra i bit ed il riporto in ingresso
  - La seconda calcola il riporto in uscita

- Dalla tabella di verità si ricavano le espressioni logiche per somma e riporto in uscita

x_i	y_i	Riporto in ingresso c_i	Somma s_i	Riporto in uscita c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

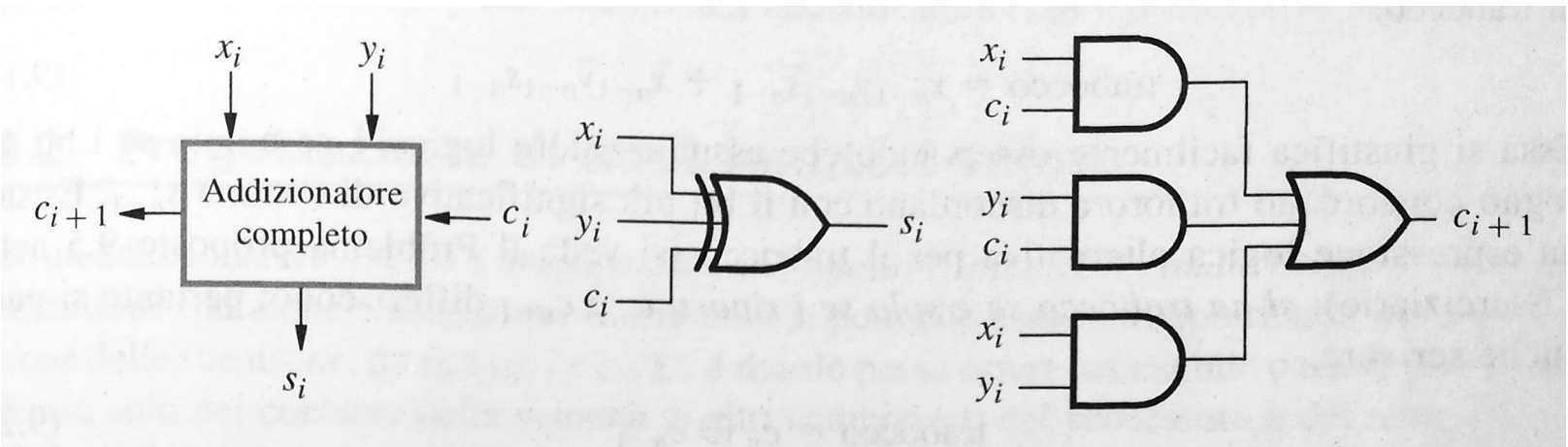
$$s_i = \bar{x}_i\bar{y}_i r_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = x_i c_i + y_i c_i + x_i y_i$$

Addizionatore completo (full adder)

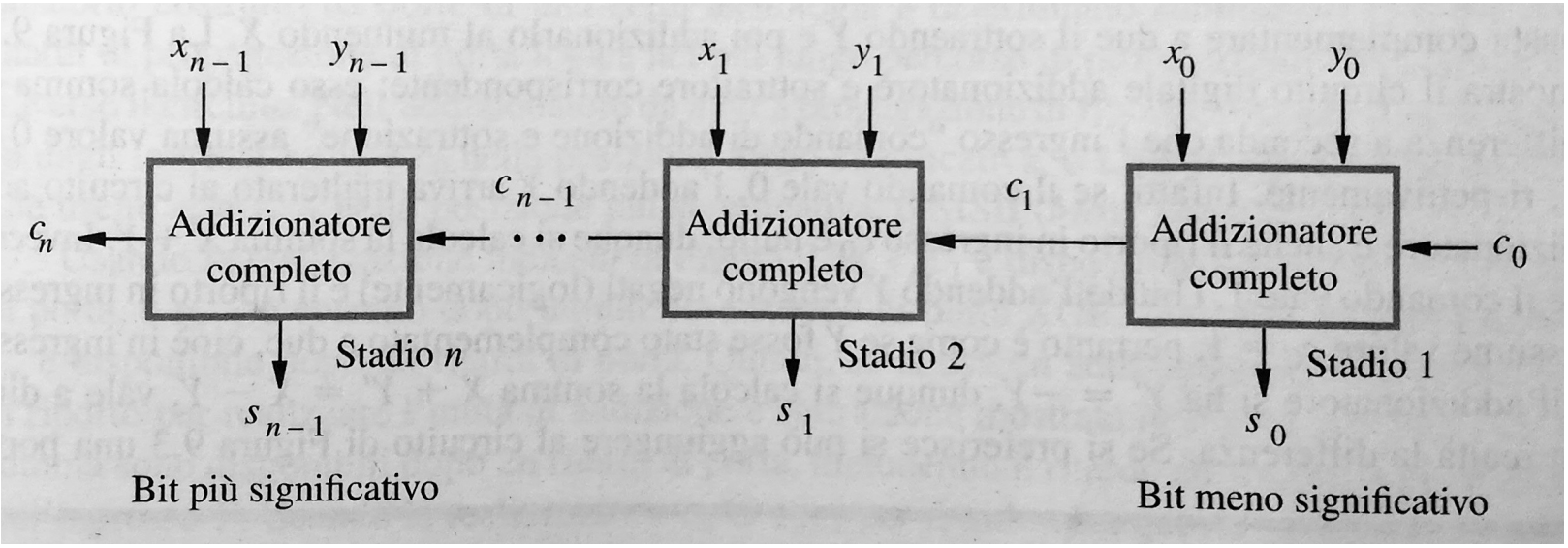
- Unendo assieme in un singolo circuito le reti logiche per le funzioni di somma e riporto in uscita si ottiene l'addizionatore completo

- L'addizionatore completo prende in ingresso i due bit da sommare e il riporto in entrata e rende in uscita somma e riporto in uscita



Addizionatore a propagazione di riporto

- Collegando una catena di n addizionatori completi in modo da propagare il riporto si ottiene un circuito in grado di sommare numeri binari di n bit
- Tale circuito è chiamato addizionatore a propagazione di riporto (ripple carry adder)



Trabocco

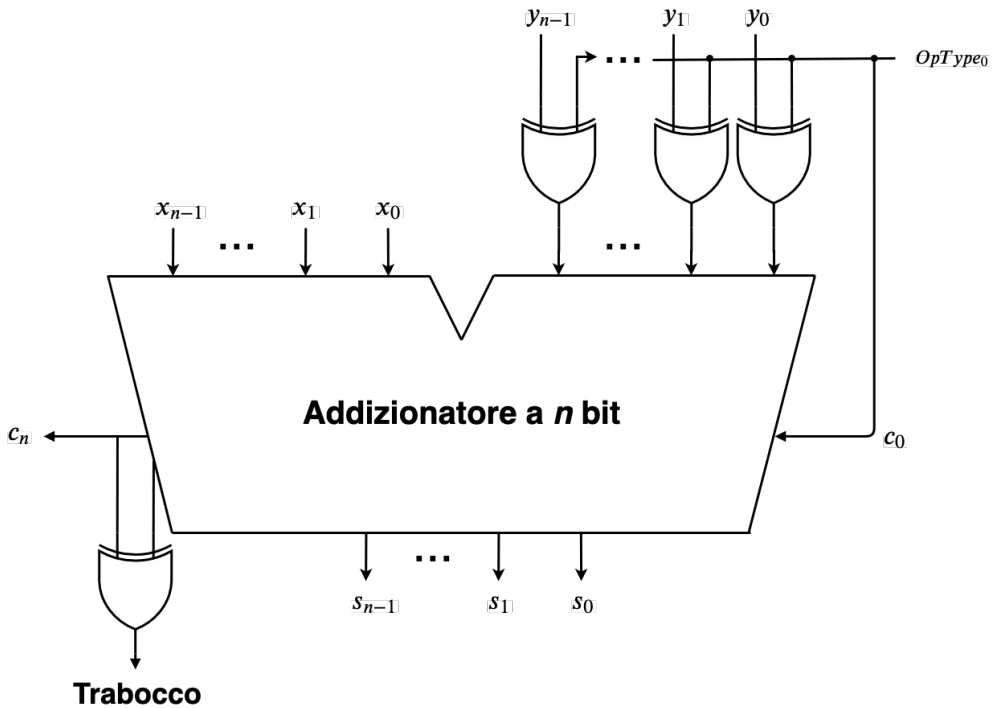
- L’addizione di due numeri in complemento a due corrisponde alla somma di due numeri binari naturali senza contare il riporto in uscita
- La sottrazione corrisponde ad un addizione complementando a due il sottraendo
- Bisogna però garantire che non avvenga trabocco
- Il calcolo del trabocco può essere espresso da una delle seguenti espressioni logiche:

$$trabocco = x_{n-1} y_{n-1} \bar{s}_{n-1} + \bar{x}_{n-1} \bar{y}_{n-1} s_{n-1}$$

$$trabocco = c_n \oplus c_{n-1}$$

Addizionatore algebrico a n bit

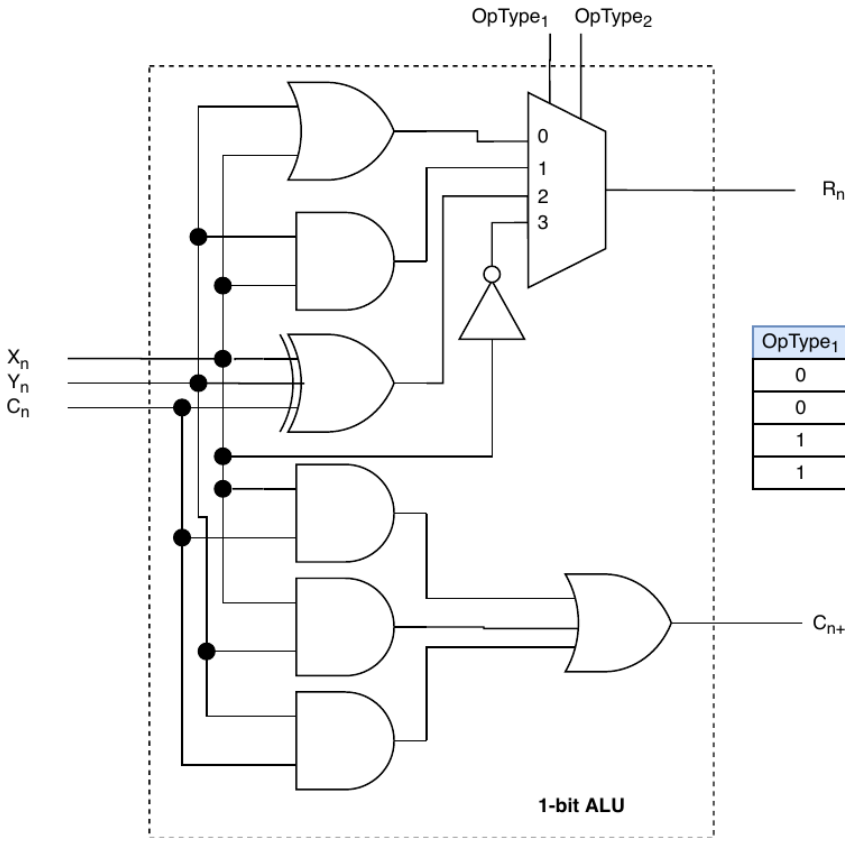
- Una unità logica per addizione e sottrazione può essere ottenuta usando un addizionatore a propagazione di riporto
- Si usa il bit $OpType_0$ per complementare a due il sottraendo in caso di sottrazione
- Nel caso $OpType_0 = 1$ si avrà un riporto in ingresso al bit meno significativo e il secondo addendo verrà complementato attraverso una catena di porte xor parallele ($y_n \oplus OpType_0$)
- Il trabocco viene calcolato come $c_n \oplus c_{n-1}$



ALU a 1 bit

- Estendiamo ora l'addizionatore completo includendo anche la possibilità di effettuare le seguenti operazioni logiche bitwise AND, OR e NOT

- Aggiungiamo dunque un multiplexer che consente di mandare in output, alternativamente, l'uscita del:
  - Sommatore: $x_i + y_i$
  - Porta AND: x_i AND y_i
  - Porta OR: x_i OR y_i
  - Negazione: $\neg x_i$



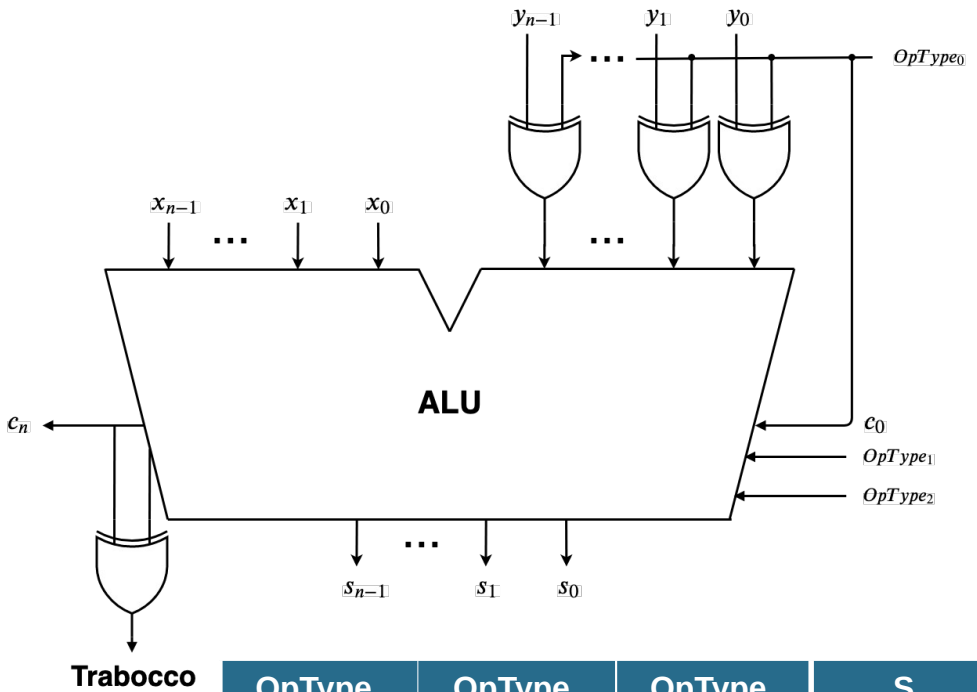
$OpType_1$	$OpType_2$	R_n
0	0	X_n or Y_n
0	1	X_n and Y_n
1	0	$X_n + Y_n$
1	1	not(X_n)

ALU a n bit

- Collegando in serie n ALU a 1 bit in un'unità logica simile all'addizionatore visto in precedenza otterremo una ALU a n bit

- I bit di controllo $OpType_1$ e $OpType_2$ servono a selezionare l'operazione da eseguire (addizione, AND, OR o NOT)

- $OpType_0$ serve a selezionare la sottrazione e complementare a 2 il sottraendo



$OpType_0$	$OpType_1$	$OpType_2$	S
0	1	0	X + Y
1	1	0	X - Y
0	0	0	X AND Y
0	0	1	X OR Y
0	1	1	NOT(X)