

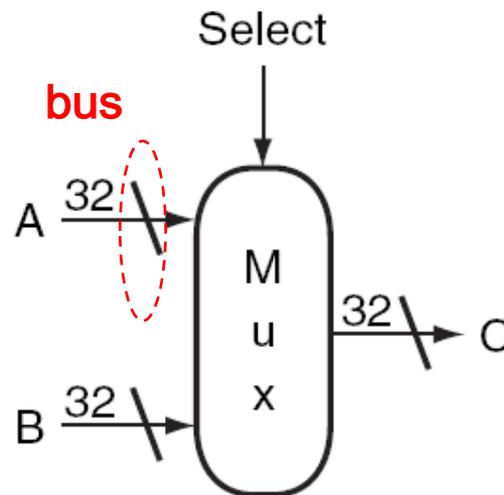
Architettura degli Elaboratori 2021-2022

Circuiti digitali
Logica combinatoria

Prof. Elisabetta Fersini
elisabetta.fersini@unimib.it

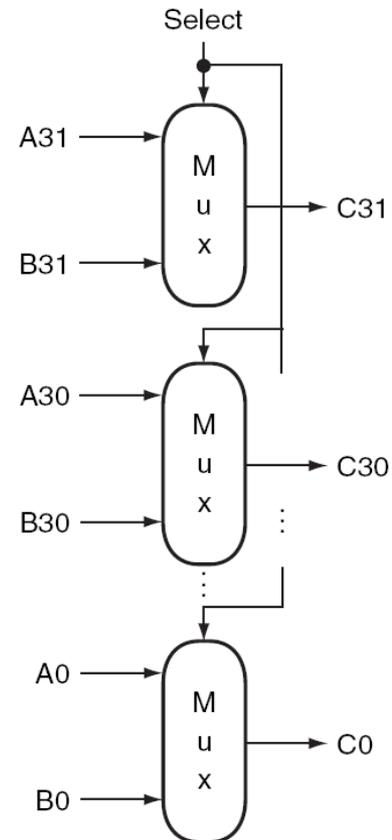
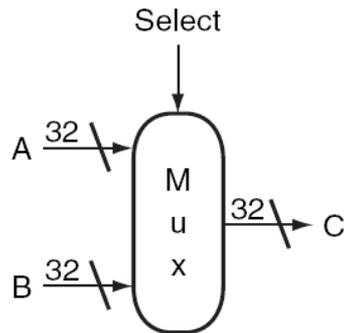
Array di elementi logici

- La maggior parte delle operazioni vengono svolte su 32 bit, mettendo in luce la necessità di creare **array di elementi logici**.
- Un **bus** una collezione di linee di input che verranno trattate come un singolo segnale



Array di elementi logici

- Un multiplexor con un bus a 32-bit corrisponde ad un array di 32 multiplexor ad 1-bit.



ALU (Arithmetic Logic Unit)

- L'Arithmetic Logic Unit:
 - E' la parte del processore che svolge le operazioni aritmetico-logiche
 - E' un insieme di circuiti combinatori che implementa:
 - Operazioni aritmetiche: es somma e sottrazione
 - Operazioni logiche: es AND e OR

Blocchi di base per costruire l'ALU

1. AND gate ($c = a \cdot b$)



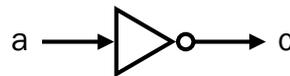
a	b	$c = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

2. OR gate ($c = a + b$)



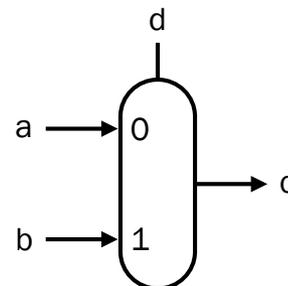
a	b	$c = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

3. Inverter ($c = \bar{a}$)



a	$c = \bar{a}$
0	1
1	0

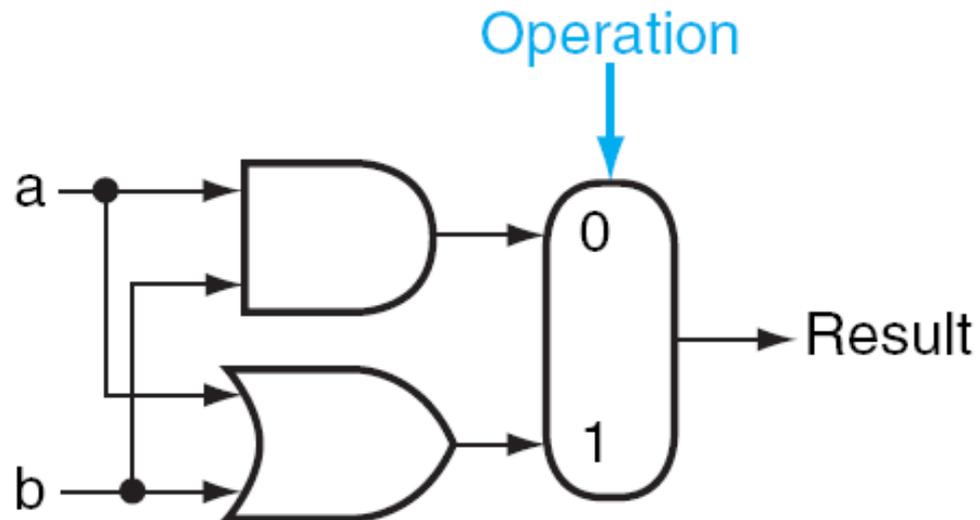
4. Multiplexor
(if $d = 0$, $c = a$;
else $c = b$)



d	c
0	a
1	b

ALU ad 1 bit – operazioni logiche

- ALU su 1 bit che implementa AND e OR



ALU ad 1 bit – operazioni aritmetiche

- **Addizione**

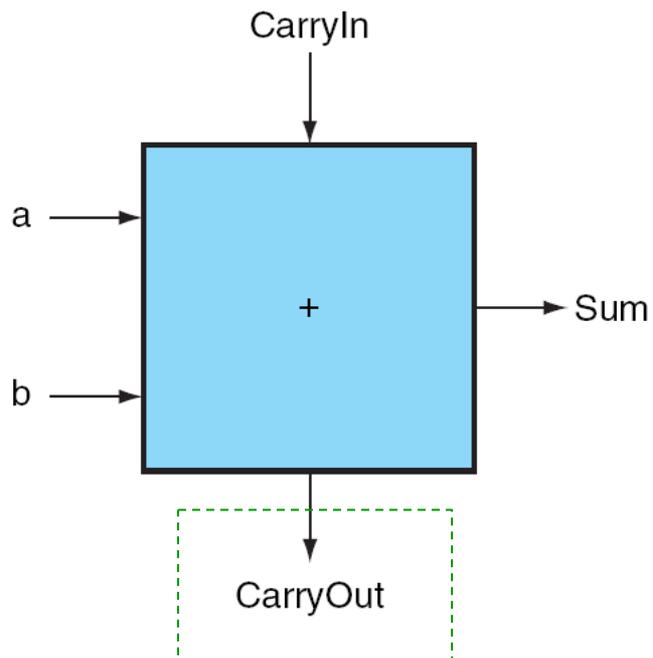
00000000000000000000 0001 1 1 1 1 0 0 1 1 0 1 0 0

carry

00000000000000000000 0000 1 0 1 1 1 1 0 0 1 0 1 1 +
 00000000000000000000 0000 0 1 1 0 1 0 0 1 1 0 1 0 =

00000000000000000000 0001 0 0 1 0 0 1 1 0 0 1 0 1

- **Addizione**



ALU ad 1 bit – operazioni aritmetiche

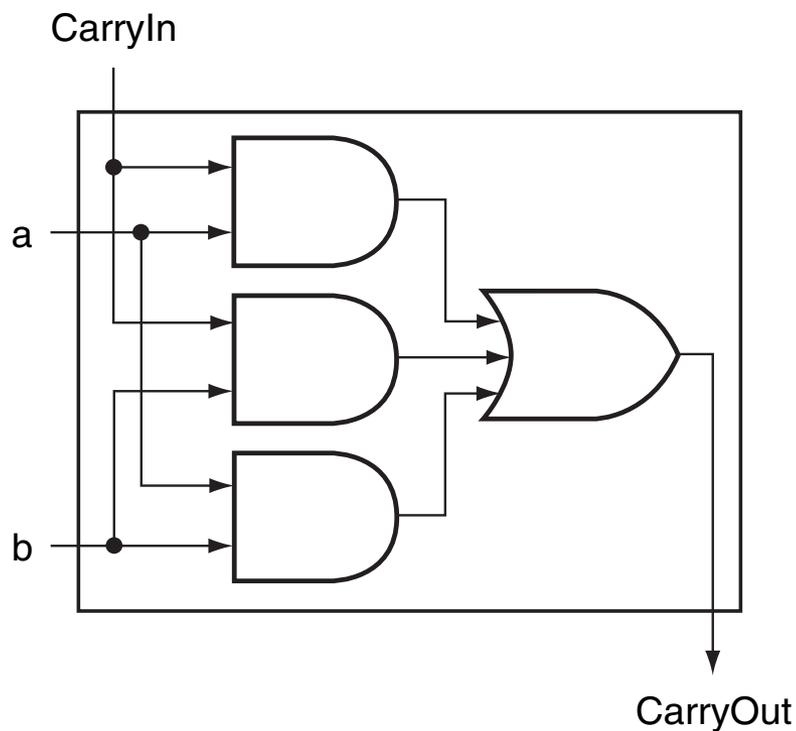
- **Addizione**

a	b	CarryIn	CarryOut	Somma
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

$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b) + (a \cdot b \cdot \text{CarryIn})$$

$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b)$$

- **Addizione**



$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b)$$

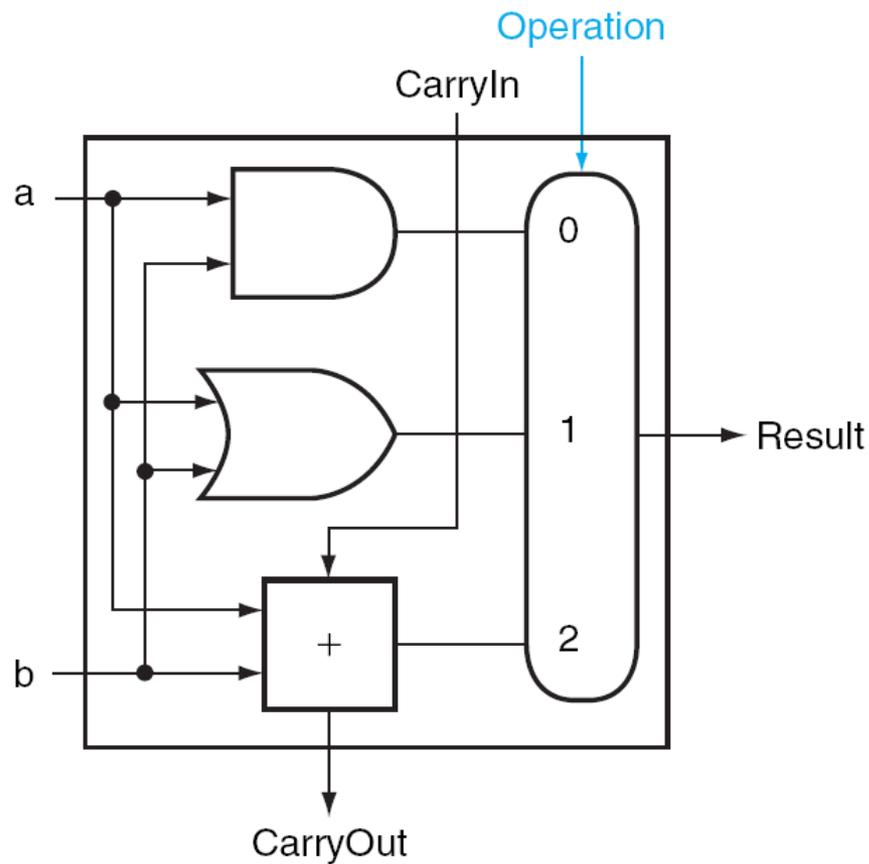
- **Addizione**

a	b	CarryIn	CarryOut	Somma
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

$$\text{Sum} = (a \cdot \bar{b} \cdot \overline{\text{CarryIn}}) + (\bar{a} \cdot b \cdot \overline{\text{CarryIn}}) + (\bar{a} \cdot \bar{b} \cdot \text{CarryIn}) + (a \cdot b \cdot \text{CarryIn})$$

ALU ad 1 bit

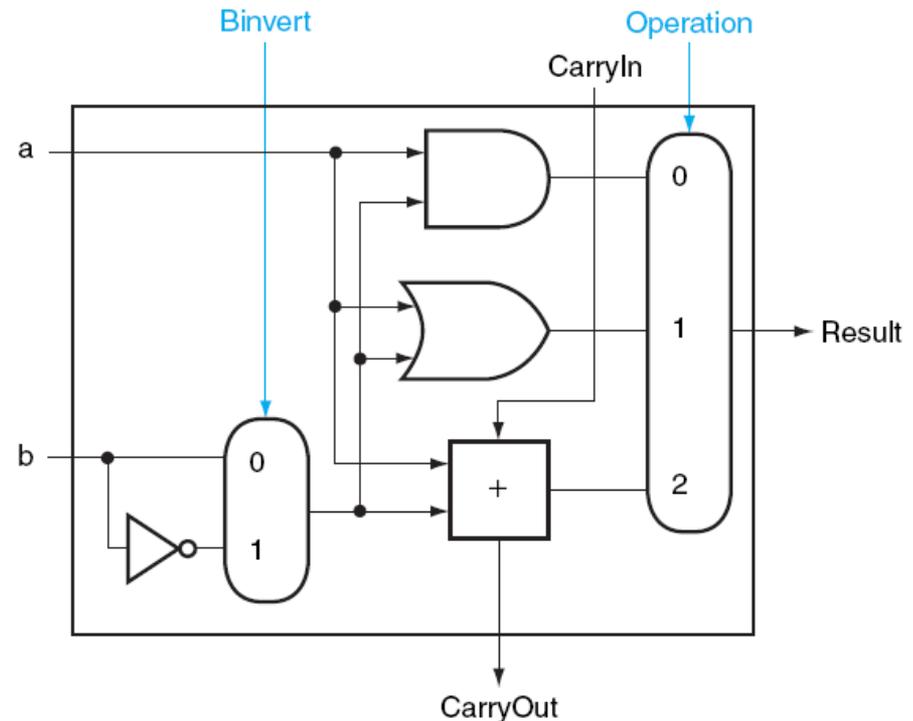
- ALU su 1 bit che esegue somma, AND, OR



ALU ad 1 bit

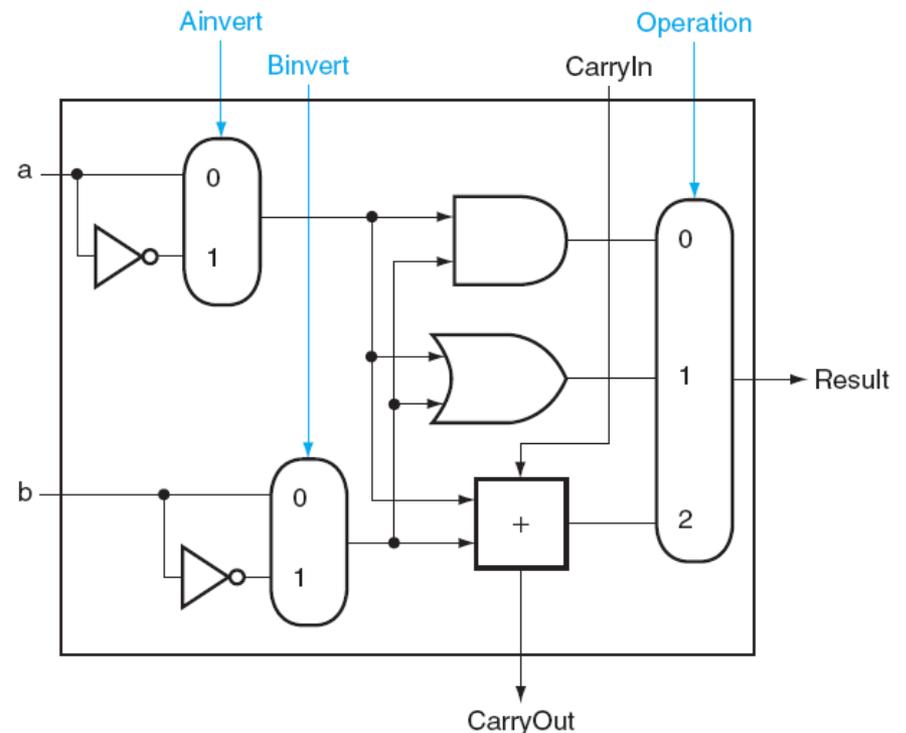
- Cosa succede se settiamo il CarryIn a 1?
 - Possiamo sommare $a+b+1$
- Se neghiamo b , possiamo ottenere una **sottrazione** (in CA2)

$$a - b = a + (\bar{b} + 1)$$



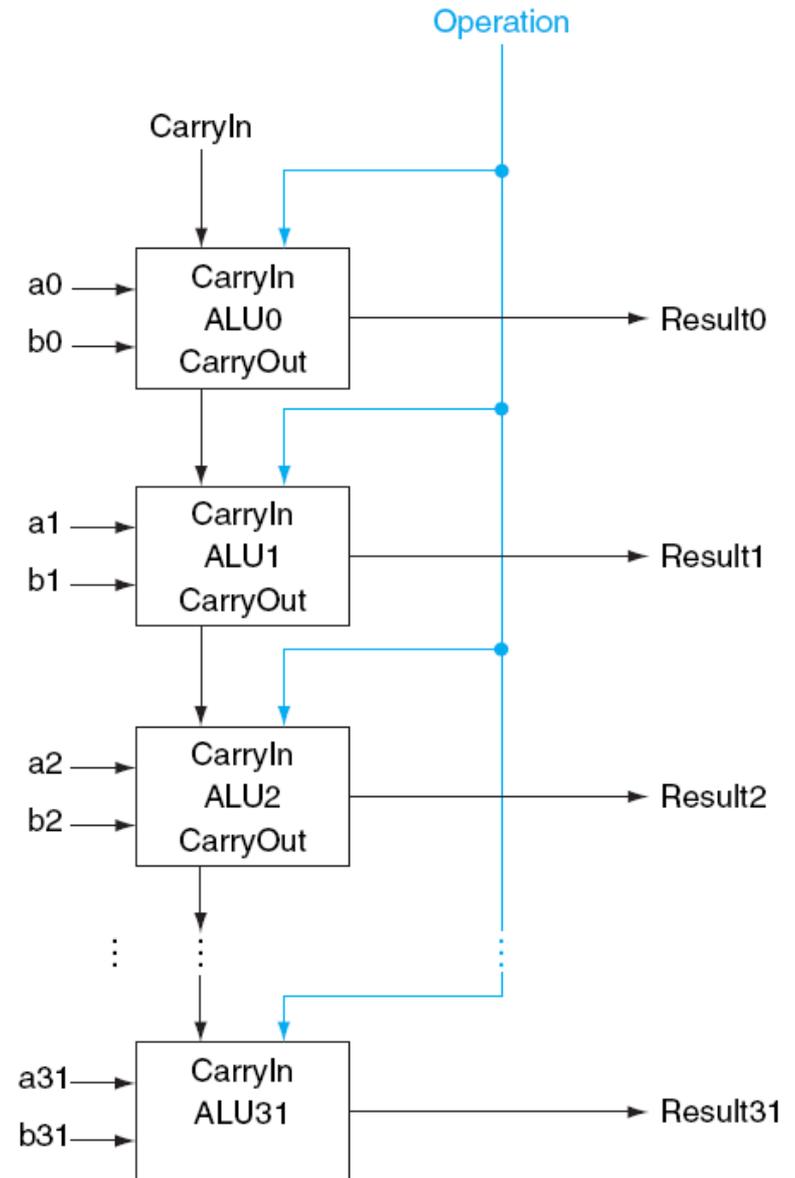
ALU ad 1 bit

- Come si implementa anche NOR? NOT(a+b)
- De Morgan:
 - NOT (a OR b) = NOT a AND NOT b $\rightarrow (\overline{a + b}) = \bar{a} \cdot \bar{b}$
 - NOT (a AND b) = NOT a or NOT B $\rightarrow (\overline{a \cdot b}) = \bar{a} + \bar{b}$



ALU su 32 bit

- Connessione di 1-bit ALU adiacenti



Da leggere

- "Basics of Logic Design", appendice C del testo 4th ed. (nel testo 5th ed. questo argomento è trattato nella appendice B)
pag. C-4 – pag. C-31