Corso di Laurea in Informatica
Architettura degli elaboratori
a.a. 2025-2026

Architettura degli Elaboratori 2025-2026

ALU

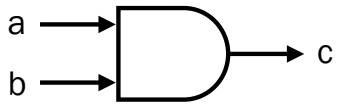
Prof. Elisabetta Fersini
elisabetta.fersini@unimib.it

ALU (Arithmetic Logic Unit)

- L'Aritmethic 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

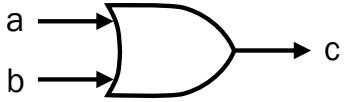
Blocchi di base per costruire l'ALU

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



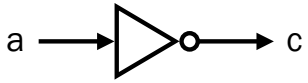
a	b	$c = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

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



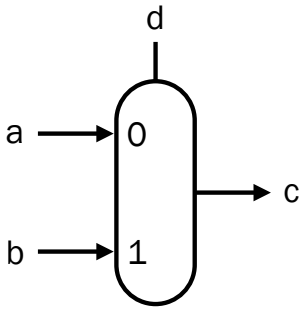
a	b	$c = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

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



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

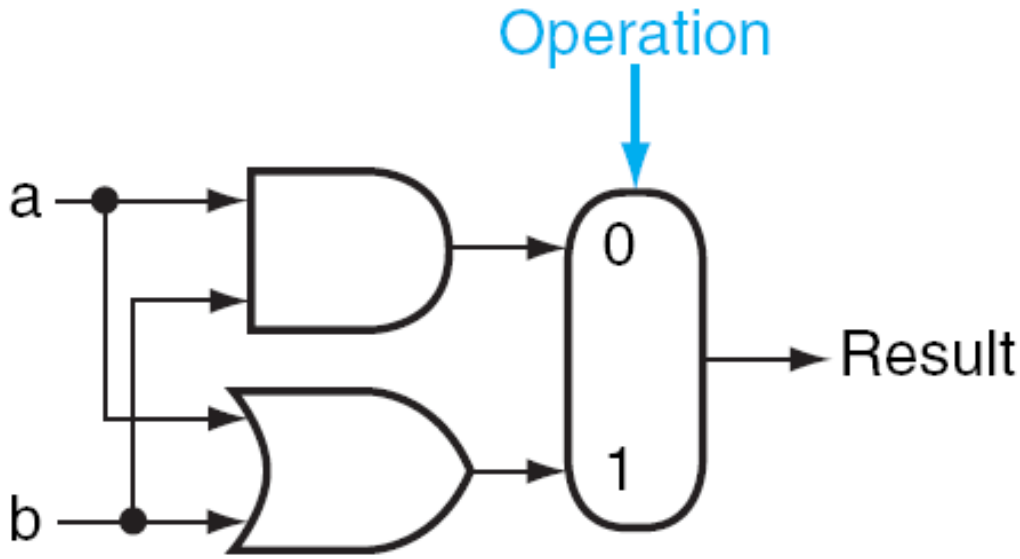
4. Multiplexor
   (if d = = 0, c = a;
   else c = b)



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

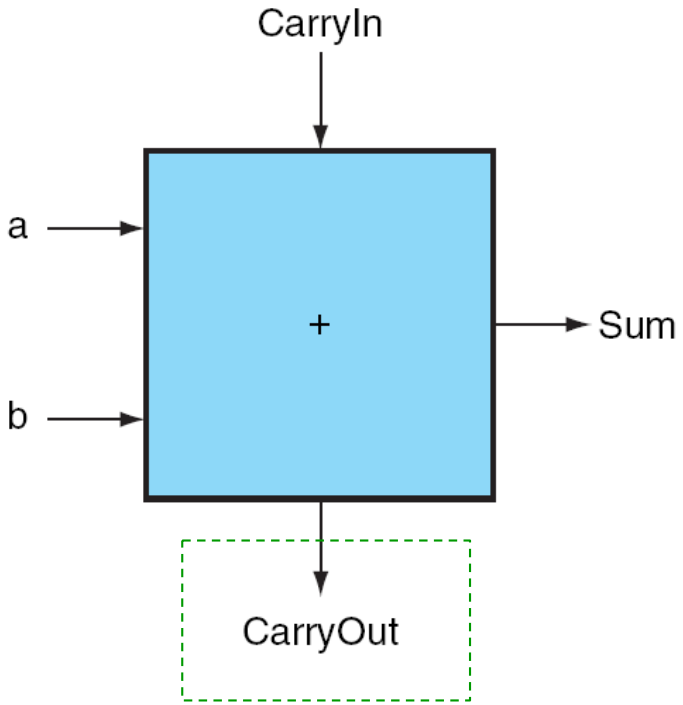
0000000000000000 0001 1 1 1 1 0 0 1 1 0 1 0 0 — carry

0000000000000000 0000 1 0 1 1 1 1 0 0 1 0 1 1 +
0000000000000000 0000 0 1 1 0 1 0 0 1 1 0 1 0 =

0000000000000000 0001 0 0 1 0 0 1 1 0 0 1 0 1

ALU ad 1 bit – operazioni aritmetiche

- 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**

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

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

ALU ad 1 bit – operazioni aritmetiche

- Il carry out in un full-adder è generato quando almeno due tra i tre ingressi (a, b, CarryIn) sono **1**. La formula logica

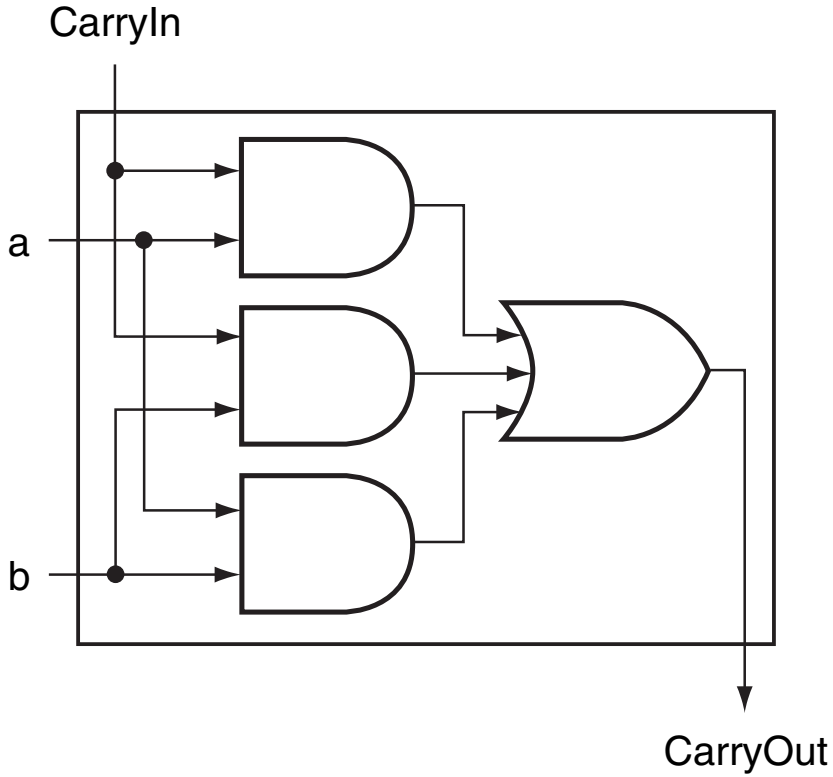
$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b)$$

già copre tutti questi casi:

- **a·b**: CarryIn=1 se sia a che b sono 1 (indipendentemente da CarryIn)
- **a·Cin**: CarryIn=1 se a e Cin sono 1 (indipendentemente da b)
- **b·Cin**: CarryIn=1 se b e Cin sono 1 (indipendentemente da a)

ALU ad 1 bit – operazioni aritmetiche

- Addizione



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

ALU ad 1 bit – operazioni aritmetiche

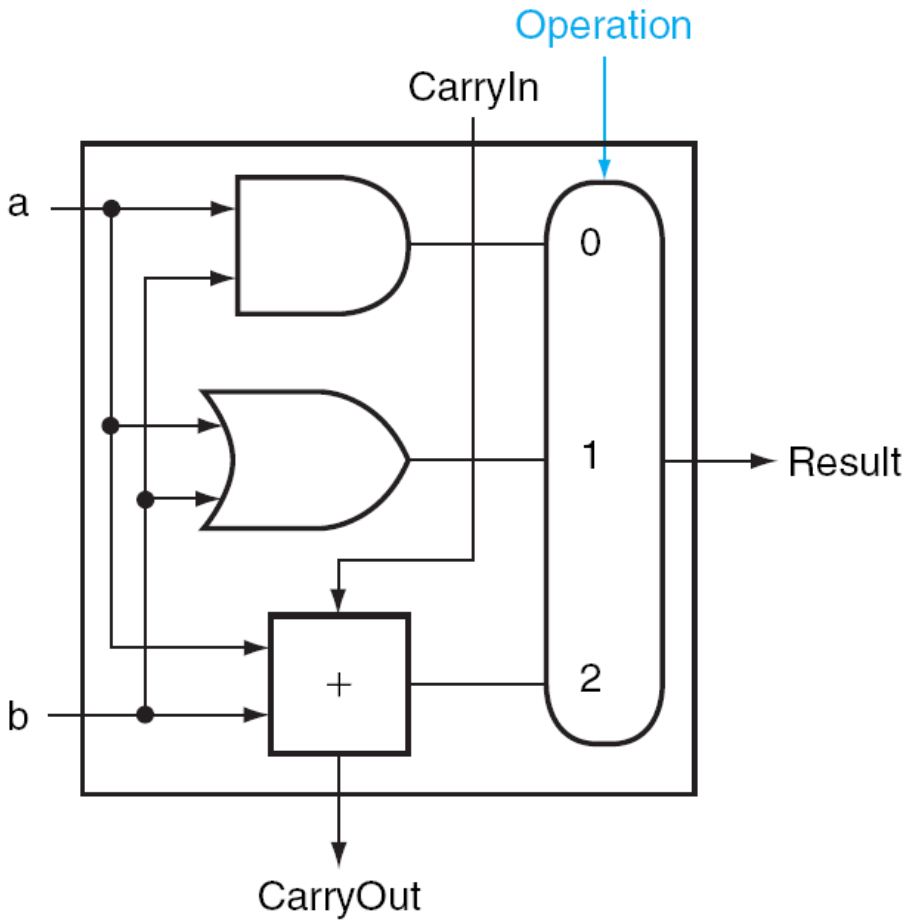
- Addizione

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**

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

ALU ad 1 bit

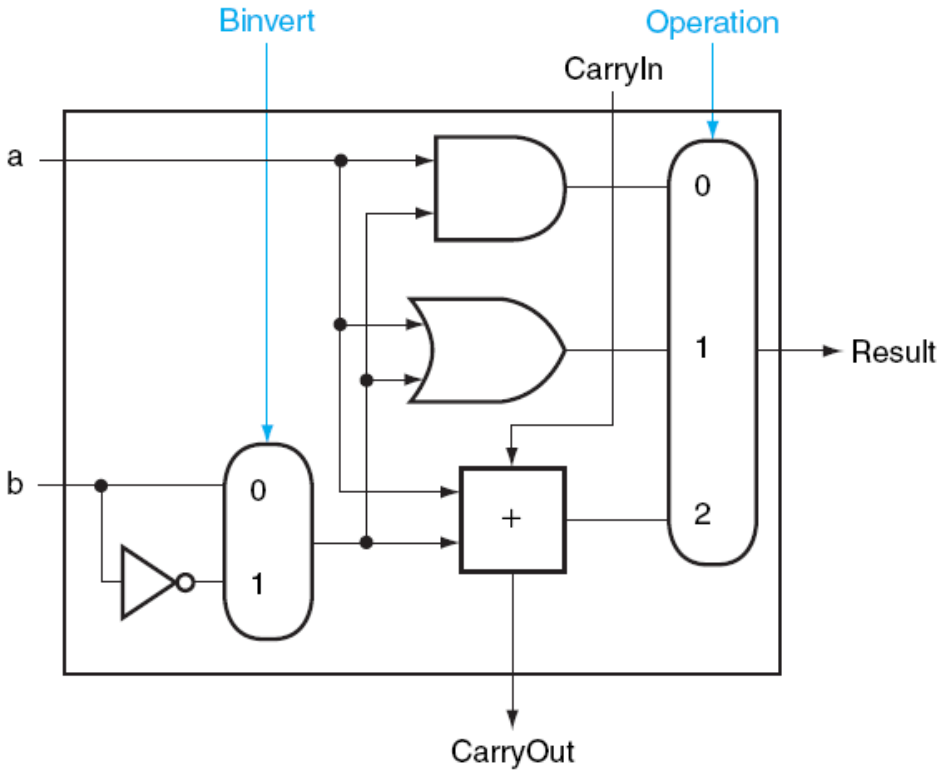
- ALU su 1 bit che esegue somma, AND, OR



ALU ad 1 bit

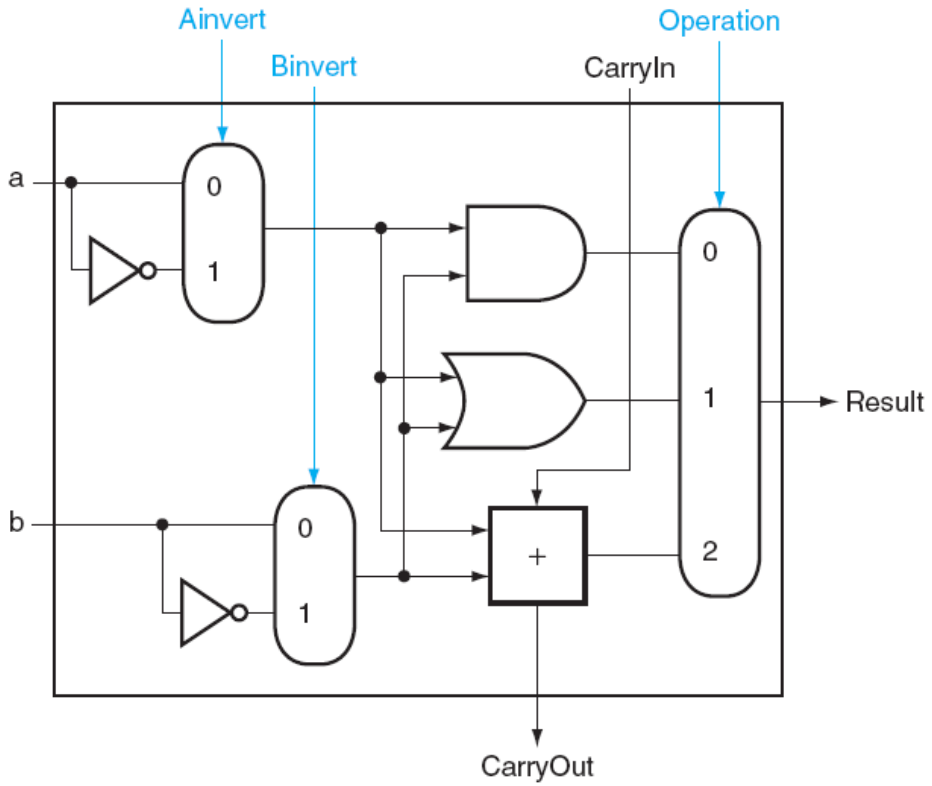
- Cosa succede se settiamo il CarryIn a 1?
  - Possiamo sommare a+b+1
- Se neghiamo b, possiamo ottenere una **sottrazione** (in CA2)

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



ALU ad 1 bit

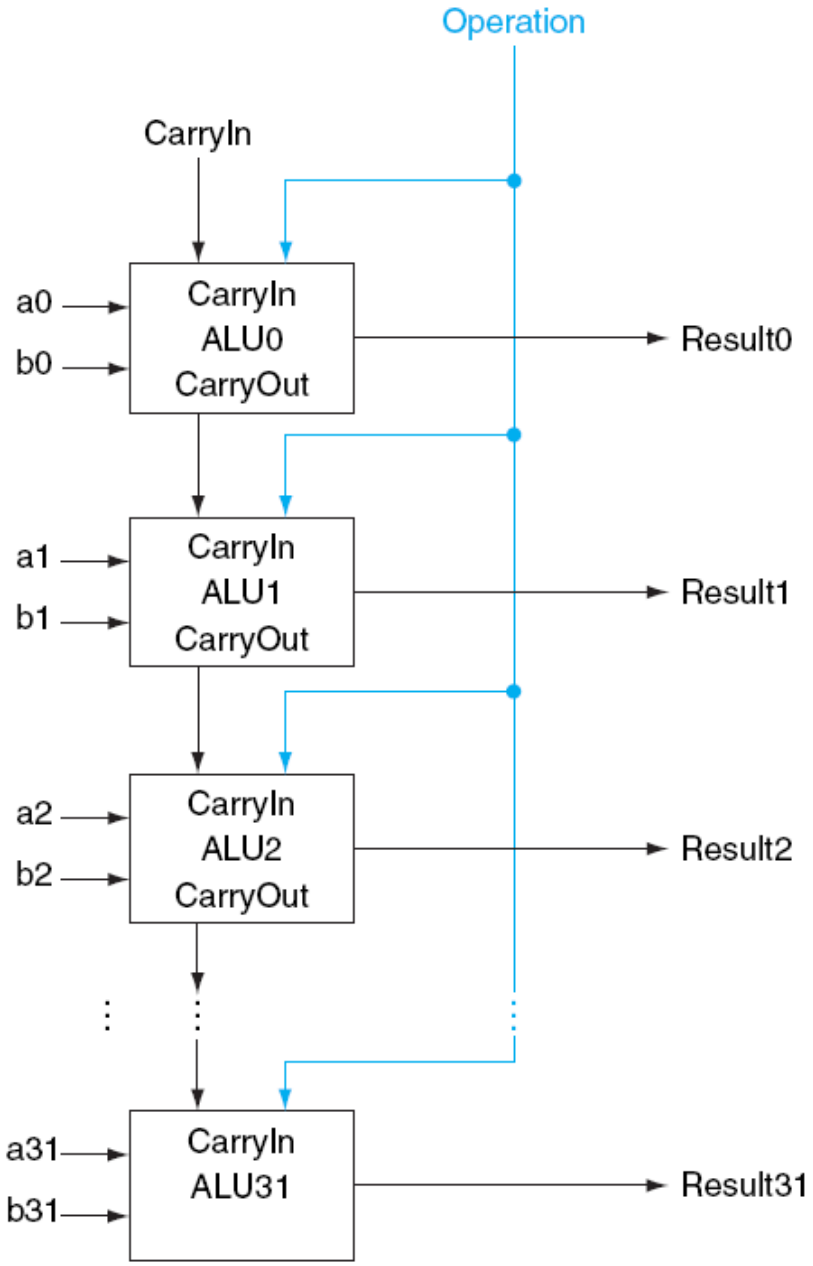
- Come si implementa anche NOR? NOT(a+b)
- De Morgan:
  - NOT (a OR b) = NOT a AND NOT b ➔ $\overline{(a+b)} = \bar{a} \cdot \bar{b}$
  - NOT (a AND b) = NOT a or NOT B ➔ $\overline{(a \cdot b)} = \bar{a} + \bar{b}$



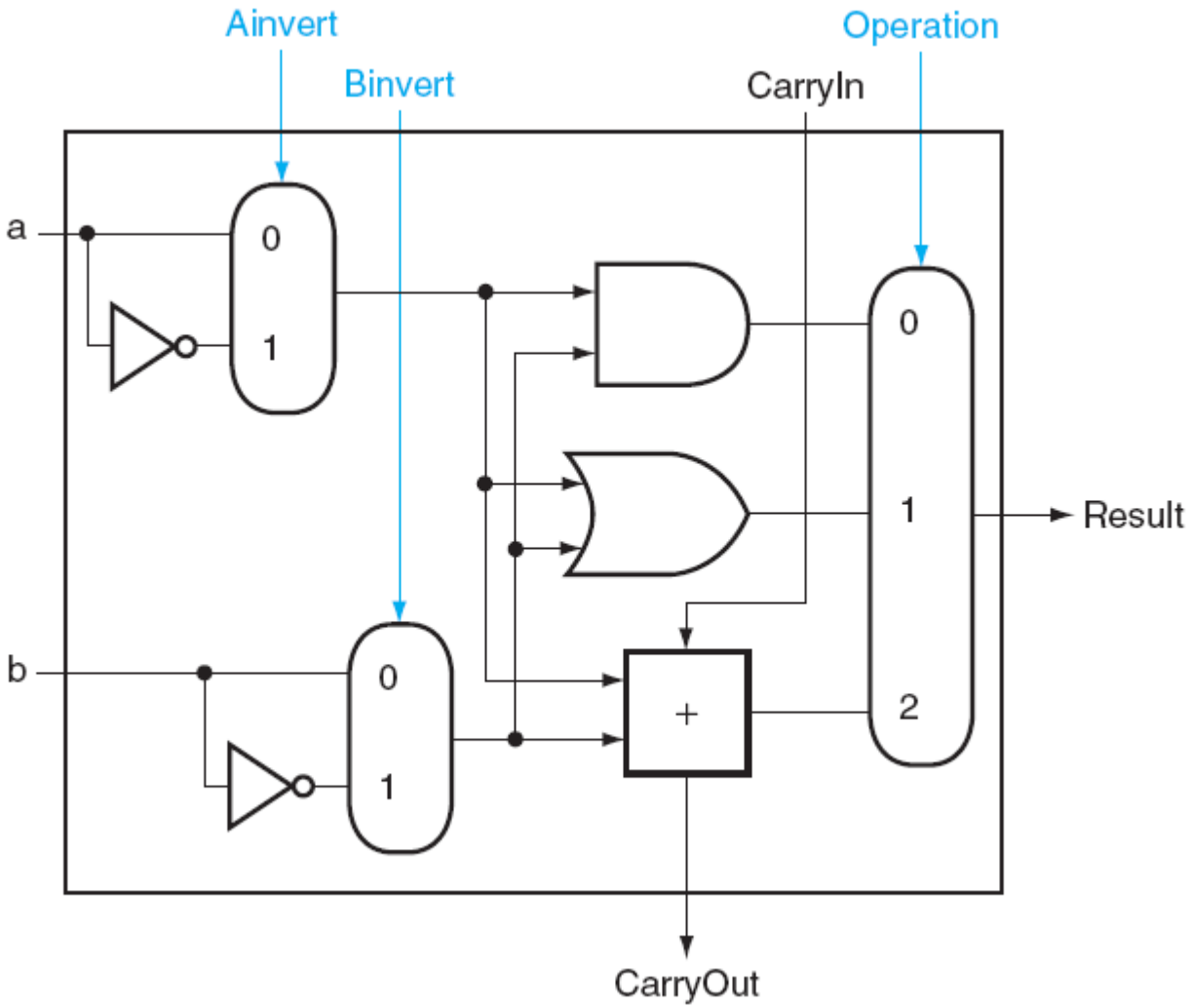
ALU su 32 bit

- Connessione di 1-bit ALU adiacenti

RIPPLE CARRY



ALU ad 1 bit



Operazioni di **confronto**:

- SLT, set-on-less-then
- BEQ, branch-on-equal

ALU su 1 bit - SLT

- SLT – Set on Less Then
  - slt registro1, registro2, registro3
  - Poni registro1=1 se il valore contenuto in registro2 è minore del valore contenuto in registro3
    - Altrimenti se il valore contenuto in registro2 non è minore del valore contenuto in registro3, allora poni registro1=0
- Risultato: 1 se a<b, 0 altrimenti

ALU su 1 bit - SLT

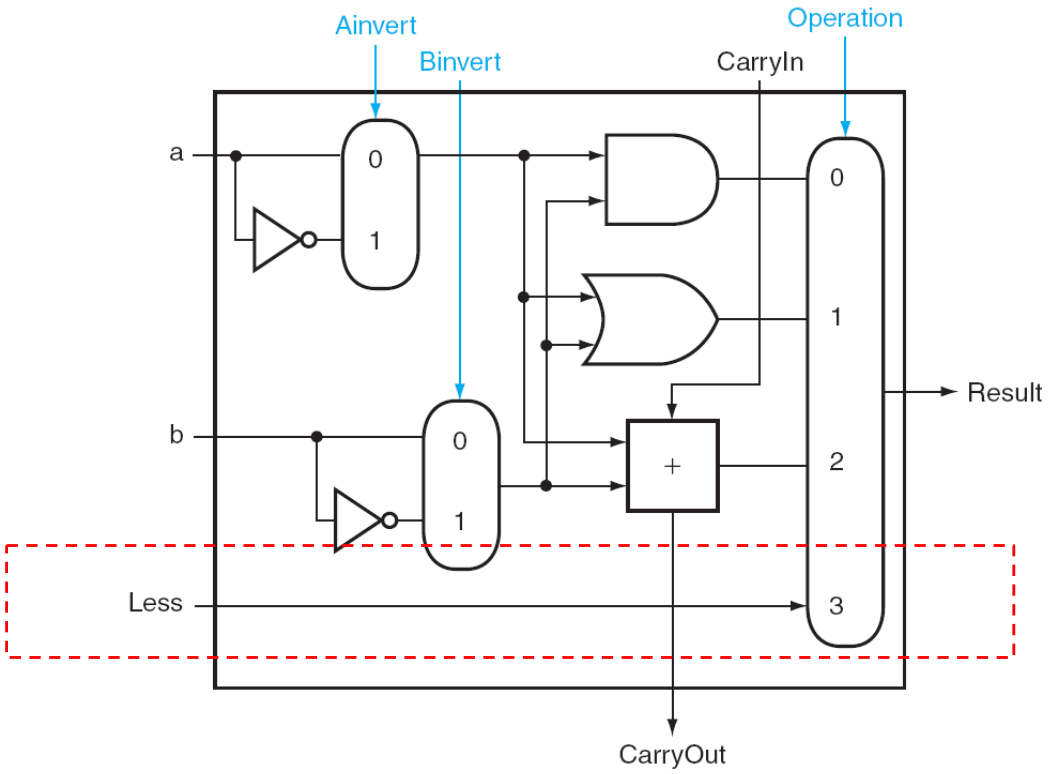
- SLT – Set on Less Then
- Risultato: uscita=1 se a<b, 0 altrimenti
- Per eseguire questa istruzione si devono poter azzerare tutti i bit dal bit-1 al bit-31 ed assegnare al bit-0 il valore il risultato dell'operazione di slt.

a<b ⇔ a-b<0
⇔ la sottrazione (a - b) è negativa
⇔ bit-31 della sottrazione (a - b)=1

- Per poter realizzare il confronto si effettua la sottrazione tra a e b
  - Se a-b è minore di 0, allora a<b (per l'istruzione slt)
    - Il risultato sarà 00...01
  - Se a-b è maggiore di 0, allora a>b (per l'istruzione slt)
    - Il risultato sarà 00...00

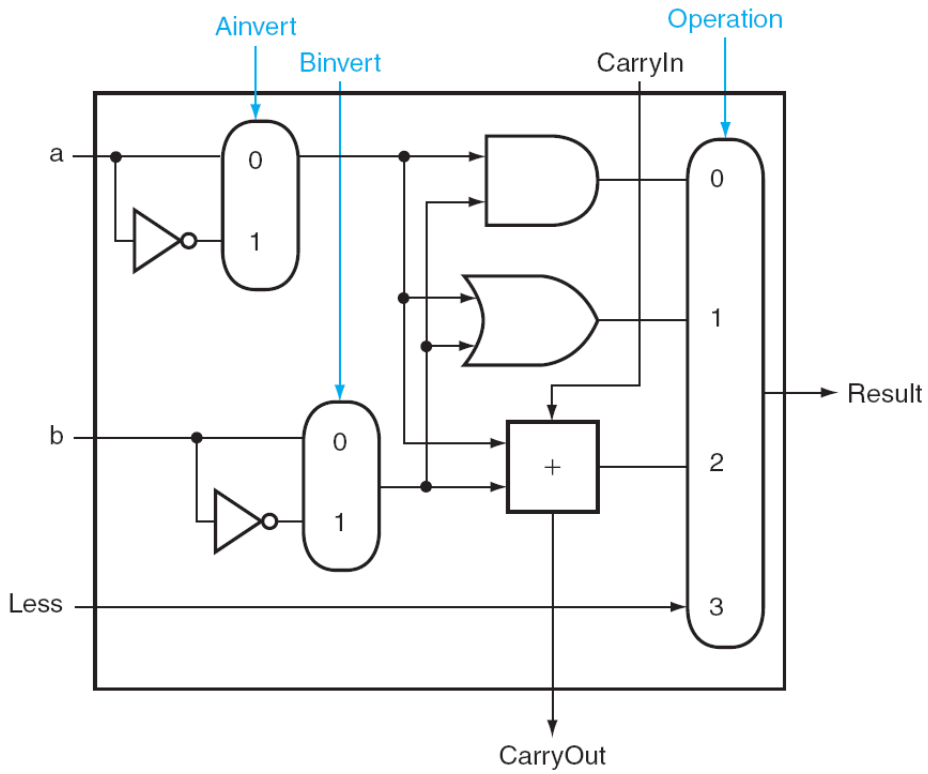
ALU su 1 bit - SLT

- ALU su 1 bit – si aggiunge Less



ALU su 32 bit - SLT

- ALU0 – ALU30



ALU1-ALU30:

$CarryIn_i = CarryOut_{i-1}$

Less = 0

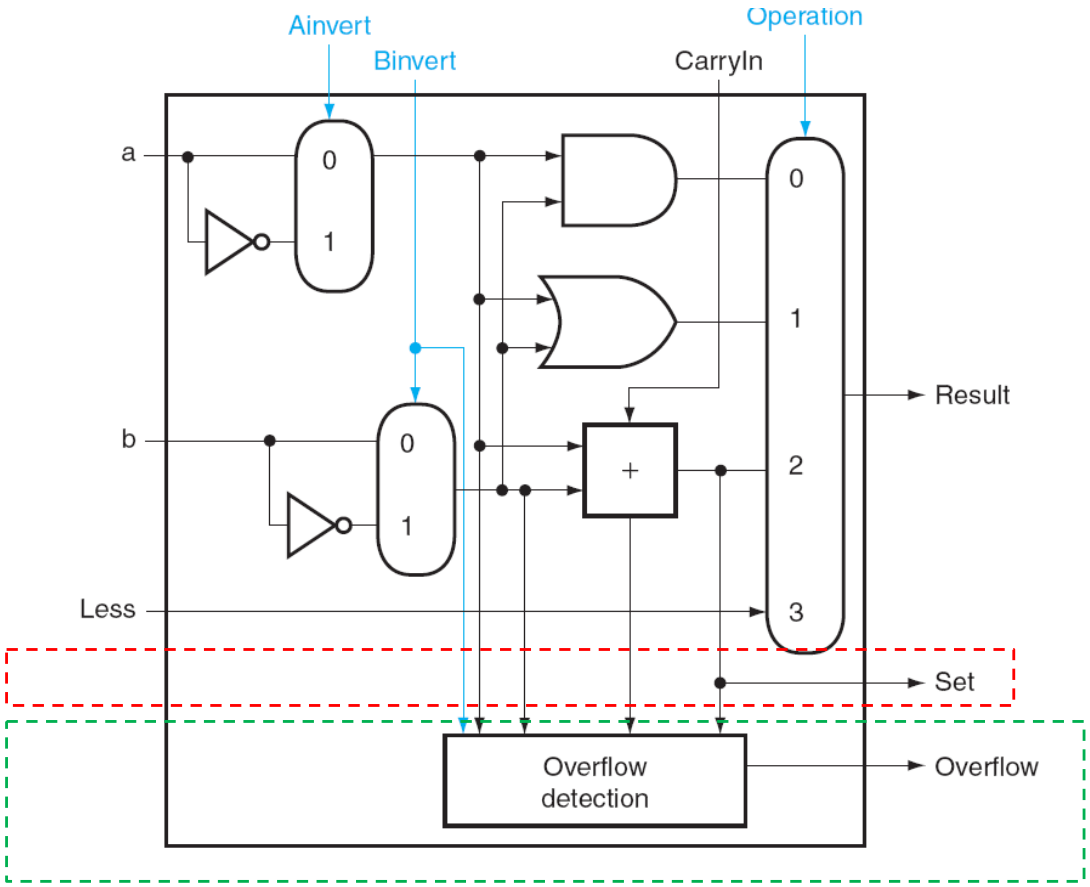
ALU0:

CarryIn = 1 (per sottrazione)

Less: vedi lucido successivo

ALU su 32 bit - SLT

- ALU31



ALU31:

$CarryIn_{31} = CarryOut_{30}$

Less = 0

Test di overflow

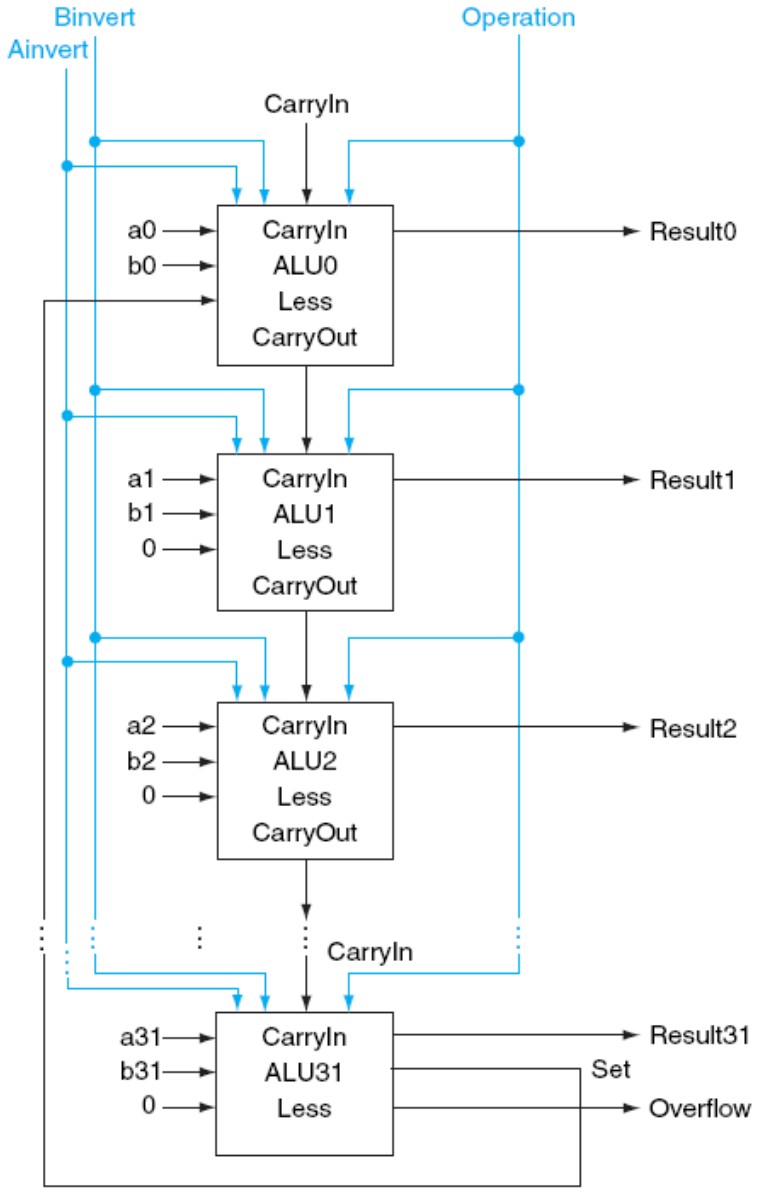
- Casi di overflow:
  - (A - B)> 0 e bit-31 di (A - B)=1 oppure
  - (A - B)< 0 e bit-31 di (A - B)=0

$$\begin{array}{lll} \text{bit-31(A)} = 0, \text{ bit-31(B)} = 1 & & \text{bit-31(A - B)} = 1 \\ \text{oppure} & & \\ \text{bit-31(A)} = 1, \text{ bit-31(B)} = 0 & & \text{bit-31(A - B)} = 0 \end{array}$$

- Controllo di overflow:

$$overflow = \bar{a} \cdot b \cdot Result + a \cdot \bar{b} \cdot \overline{Result}$$

ALU su 32 bit – con SLT + overflow



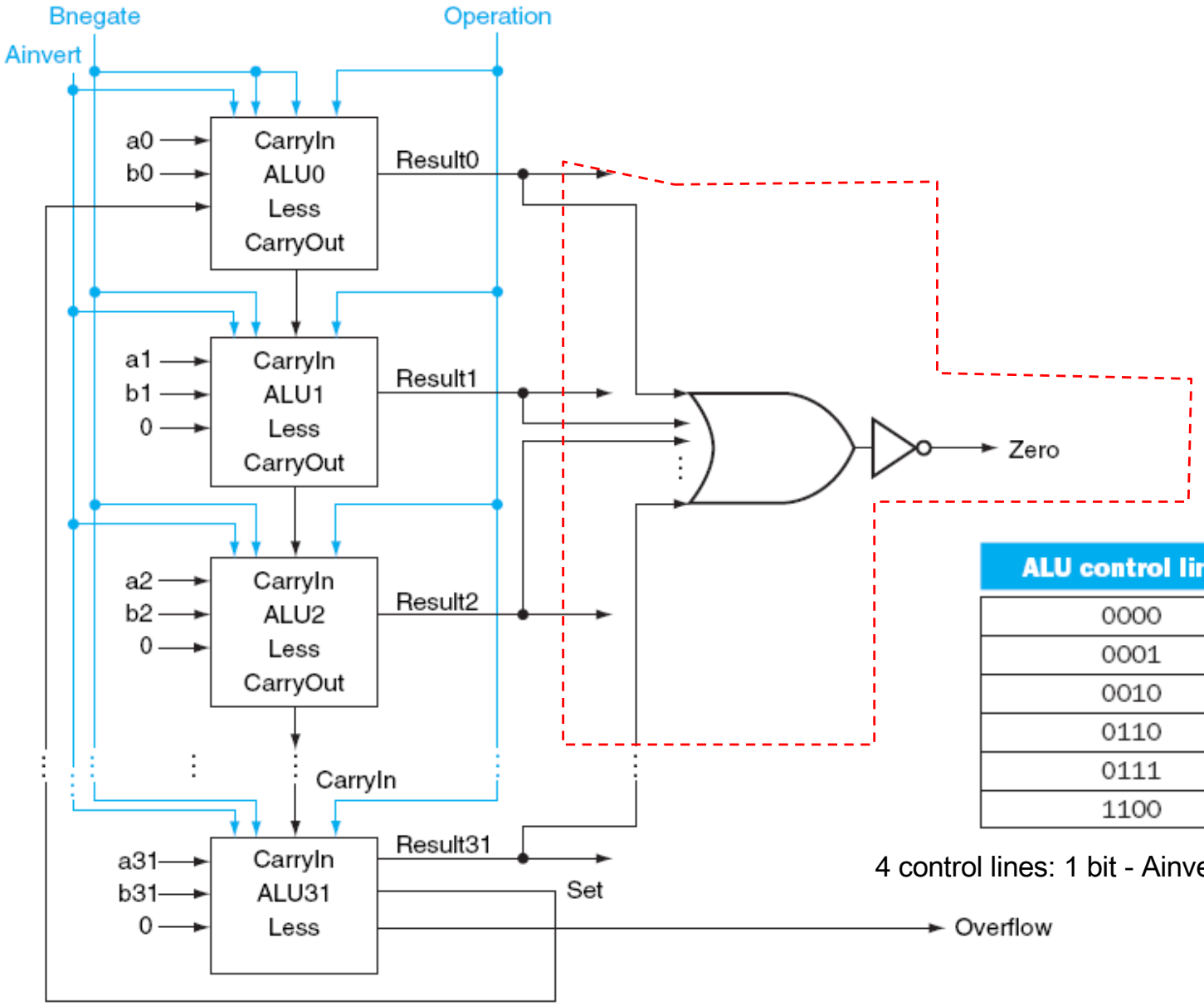
- Per sottrazioni: Binvert e CarryIn = 1
- Per addizioni e operazioni logiche: Binvert e CarryIn = 0
- Unificare i due segnali in uno: Bnegate

ALU su 32 bit – BEQ

- BEQ: branch-on-equal
  - beq registro1, registro2, L1
  - Vai all'istruzione etichettata con L1 se il valore contenuto in registro1 è uguale al valore contenuto in registro2
- Per verificare l'uguaglianza di a e b: sottrazione
  - a-b = 0 ↔ a = b
- Se a=b allora dobbiamo mettere a 1 la linea di output usata per l'uguaglianza
  - Soluzione: OR di tutte le uscite e poi negare il risultato

$$Zero = \overline{(Result31 + Result30 + \cdots + Result0)}$$

ALU su 32 bit – Branch



ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

4 control lines: 1 bit - Ainvert, 1 bit - Binvert, 2 bits - operation

ALU – rappresentazione simbolica