



# La ALU

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione

[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)

Università degli Studi di Milano



## Sommario

ALU ad 1 bit

ALU a 32 bit

Comparazione

Overflow

Test di uguaglianza



## Funzione della ALU



E' integrata nel processore, all'inizio degli anni 90 è stata rivoluzionaria la sua introduzione con il nome di co-processore matematico.

Esegue le operazioni aritmetico-logiche.

E' costituita da circuiti combinatori. Utilizza i blocchi di base già visti.

Opera su parole (MIPS 32 bit).

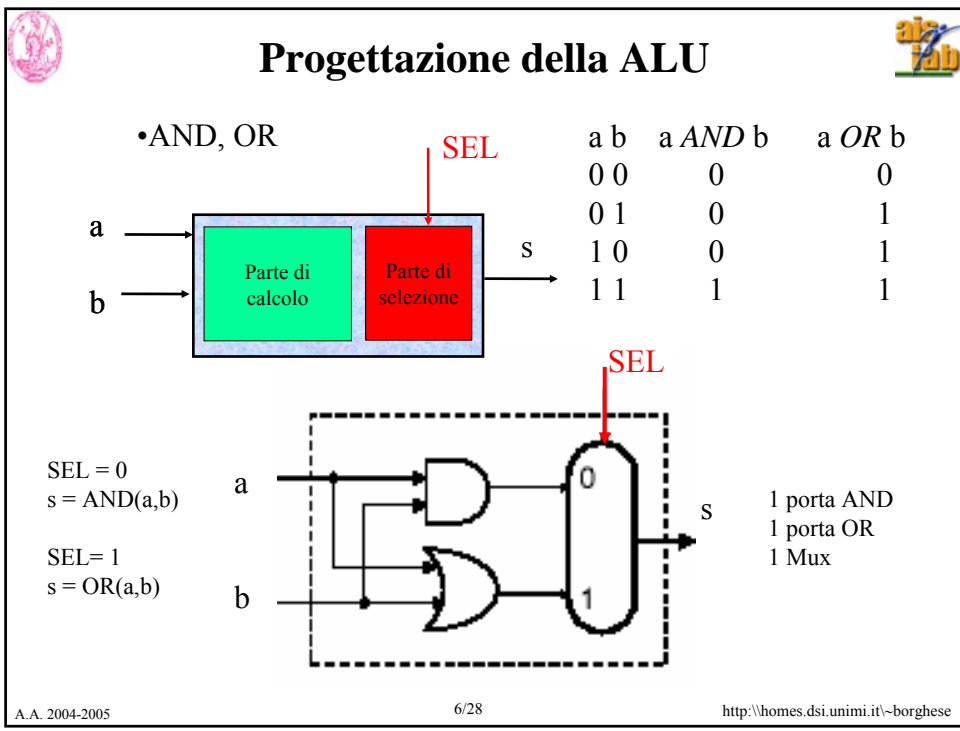
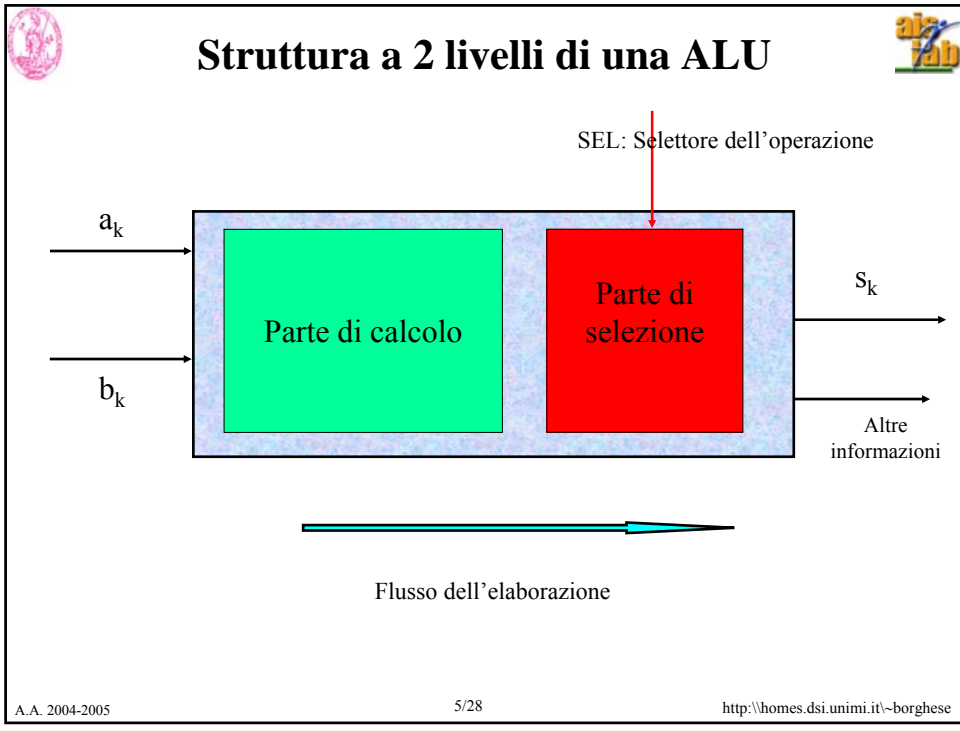
Le ALU non compaiono solamente nei micro-processori.

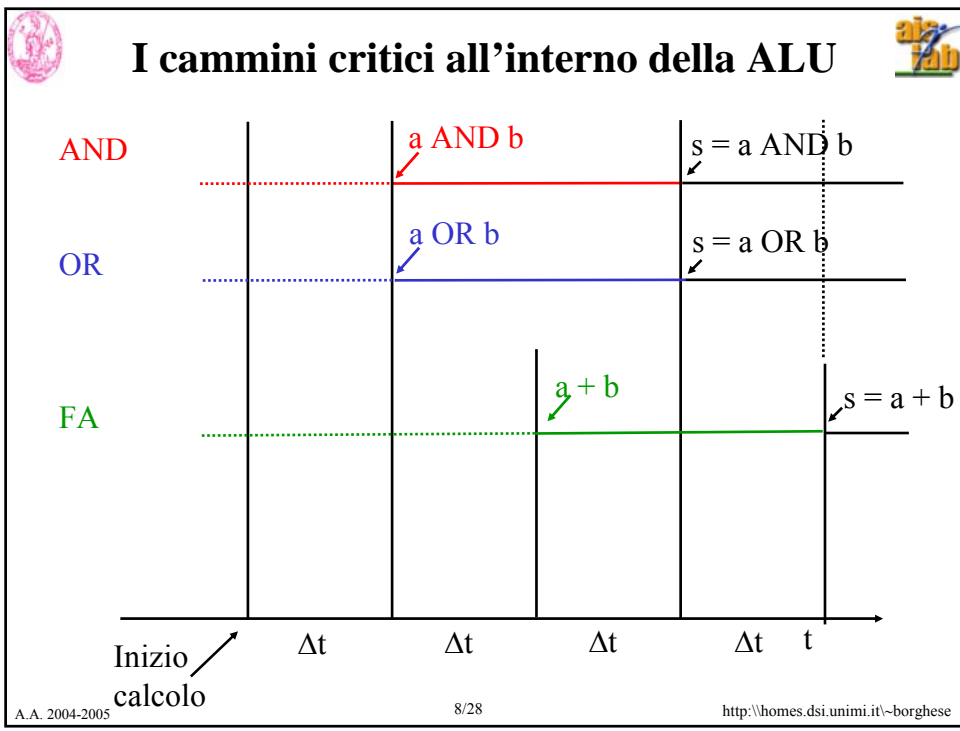
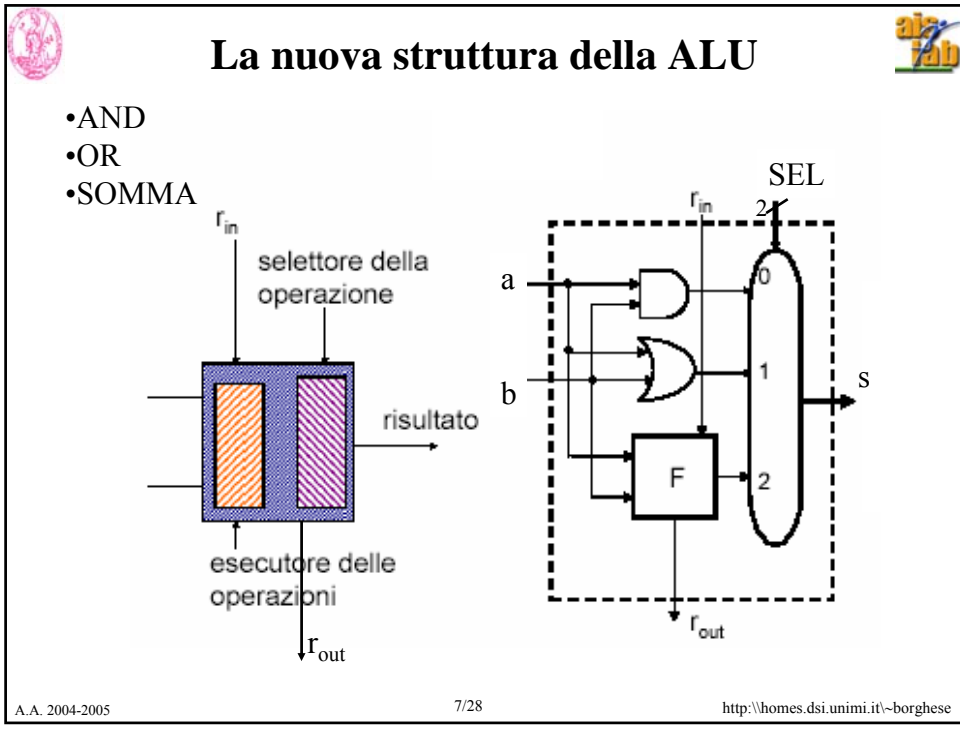


## Problematiche di progetto



- Velocità (Riporto).
- Costo.
- Precisione.
- Affidabilità
- Consumo.







## Sommario



ALU ad 1 bit

ALU a 32 bit

Comparazione

Overflow

Test di uguaglianza



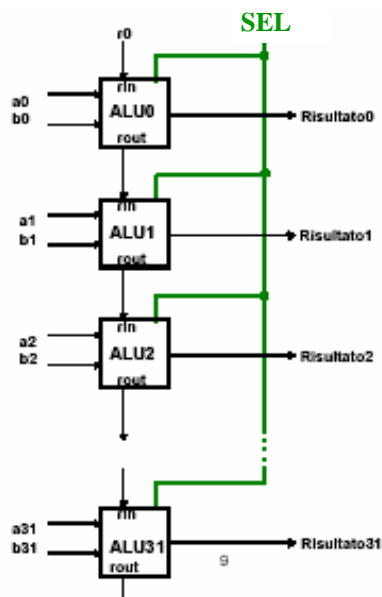
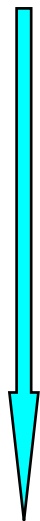
## ALU a 32 bit



Come collegare le  
ALU ad 1 bit?

Flusso di calcolo

Perchè non si può  
parallelizzare?





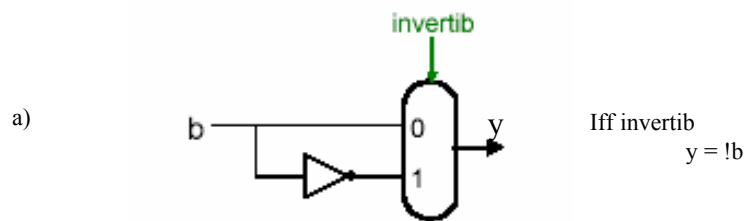
## Sottrazione



In complemento a 2 diventa un'addizione:  $a - b = a + !b + 1$

Serve:

- a) un inverter (NOT).
- b) la costante 1



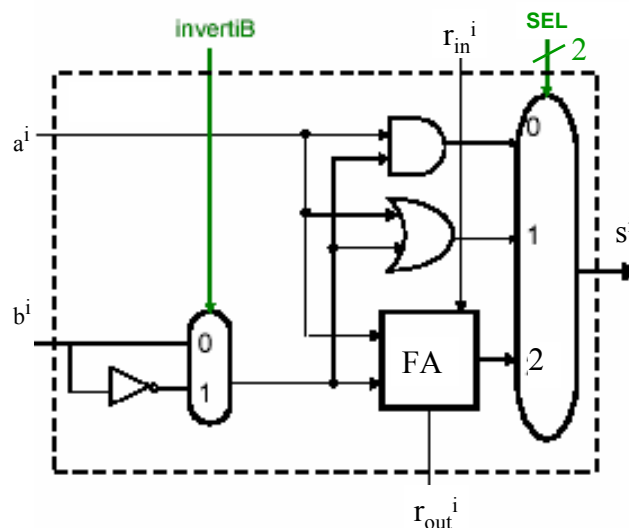
b) Da dove prendo la costante 1?



## Sottrazione - ALU<sub>i</sub>

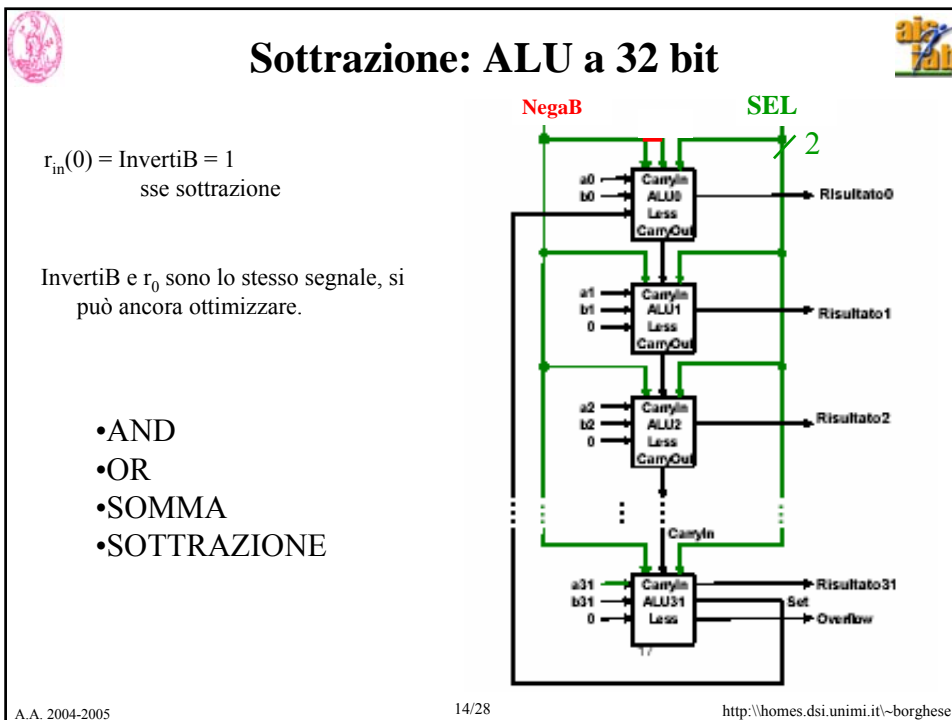
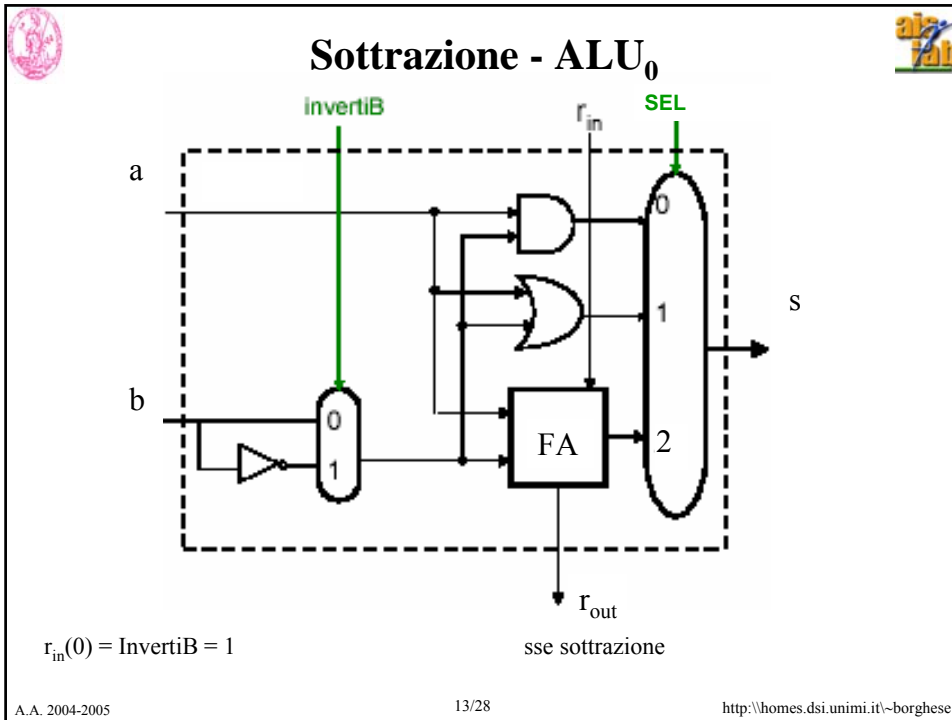


- AND
- OR
- SOMMA
- SOTTRAZIONE



$r_{in}(i) = r_{out}(i-1) \quad i = 1, 2, 3, \dots, 31$   
 InvertiB = 1

$i \neq 0$   
 sse sottrazione





## Sommario



ALU ad 1 bit

ALU a 32 bit

Comparazione

Overflow

Test di uguaglianza



## Confronto



Fondamentale per dirigere il flusso di esecuzione (test, cicli...)

if a **less\_than** b then  
    s = 1

if a < b then  
    s = 1

if (a - b) < 0 then  
    s = 1





# Come sviluppare la comparazione?



if  $(a - b) < 0$  then  
 $s = 1$   
 else  
 $s = 0$

Risultato somma



MSB

LSB

Risultato verso l'esterno



MSB

LSB

Si controlla che il primo bit della somma (bit di segno) sia = 1.

Occorre quindi:

- Impostare una differenza.
- Inviare l'uscita del sommatore del MSB all'input LESS di  $ALU_0$ .  
 (L'uscita del MSB viene anche chiamata segnale di set).



# Comparatore - $ALU^0 : ALU^{31}$

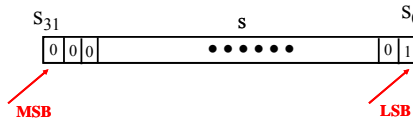
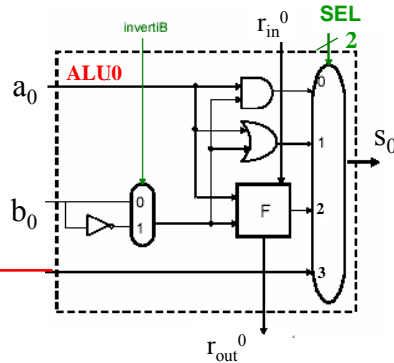
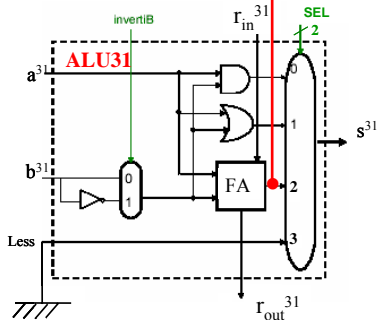


InvertiB =  $r_{in}^0 = 1$

SEL = 11

if  $(a - b) < 0$  then  
 $s = 1$   
 else  
 $s = 0$

Less(0) = MSB



MSB

LSB





## Sommario



ALU ad 1 bit

ALU a 32 bit

Comparazione

**Overflow**

Test di uguaglianza



## Overflow



Esempio decimale:

$a + b = c$  - codifica su 2 cifre,  $a = 19$ ,  $b = 83 \Rightarrow$  Overflow.

$$19 + 83 = (1)02$$

Supponiamo sia definito il bit di segno possiamo riscrivere:

$$019 + 083 = 102.$$

Quindi si ha overflow nella somma quando:

$a + b = s$ ,  $a > 0$ ,  $b > 0$  MSB di  $a$  e  $b = 0$ , MSB di  $s = 1$ .

$a + b = s$ ,  $a < 0$ ,  $b < 0$  MSB di  $a$  e  $b = 1$ , MSB di  $s = 0$ .

Si può avere overflow con la differenza?

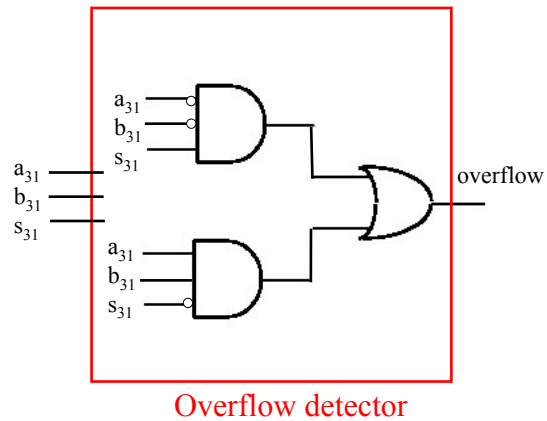


## Circuito di riconoscimento dell'overflow

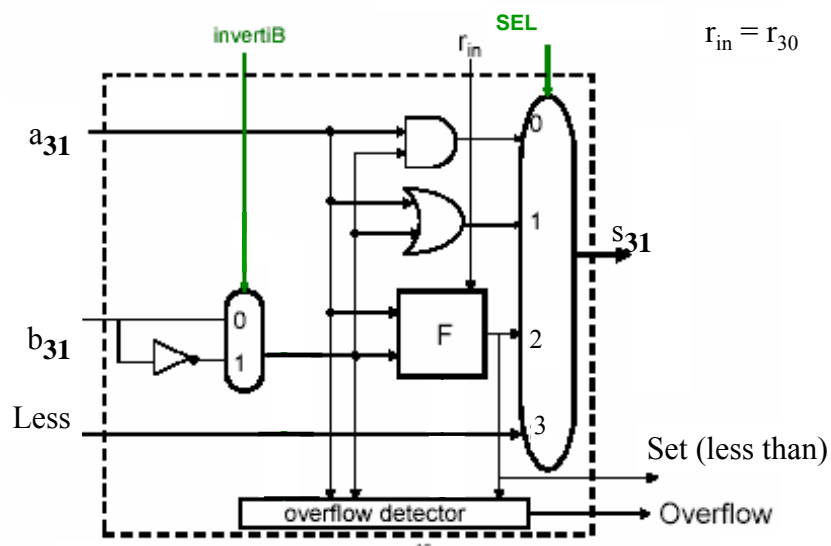


Lavora sui MSB

$a_{31}$	$b_{31}$	$s_{31}$	overflow
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



## ALU<sub>31</sub>





## Sommario



ALU ad 1 bit

ALU a 32 bit

Comparazione

Overflow

**Test di uguaglianza**



## Operazione di uguaglianza



beq rs, rt label                      iff  $(rs - rt) = 0$  , salta.

*Occorre quindi:*

- Impostare una differenza.
- Effettuare l'OR logico di tutti i bit somma.
- L'uscita dell'OR logico = 0 i due numeri sono uguali.

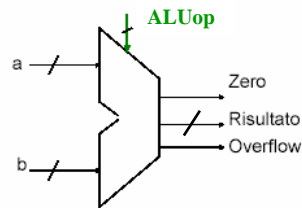


## ALU a 32 bit: struttura finale

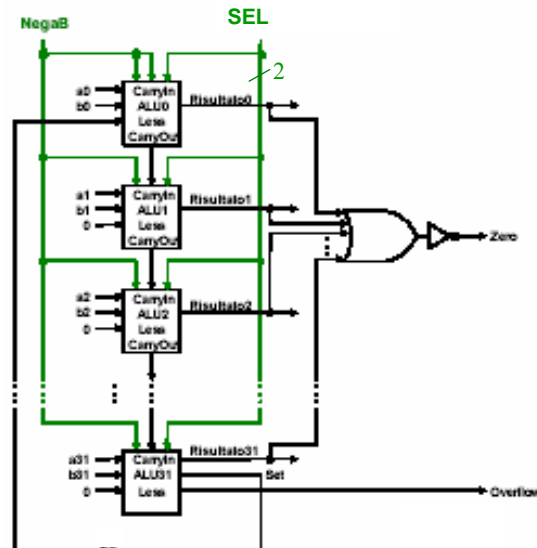


Operazioni possibili:

- AND
- OR
- Somma / Sottrazione
- Comparazione
- Test di uguaglianza



Sono evidenziate  
solamente le variabili  
visibili all'esterno.



## Sommario



ALU ad 1 bit

ALU a 32 bit

Comparazione

Overflow

Test di uguaglianza