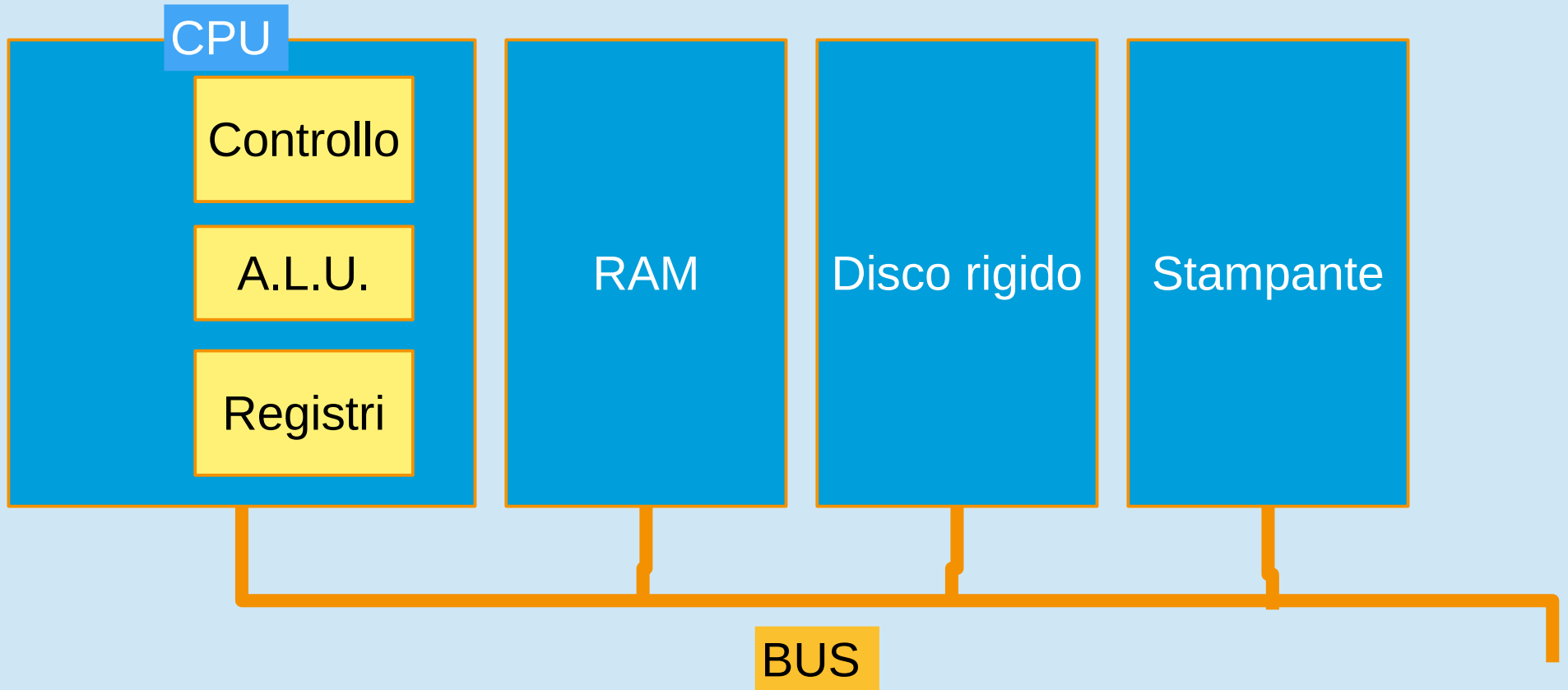


Architetture di calcolo

01 Processori

Organizzazione del Computer

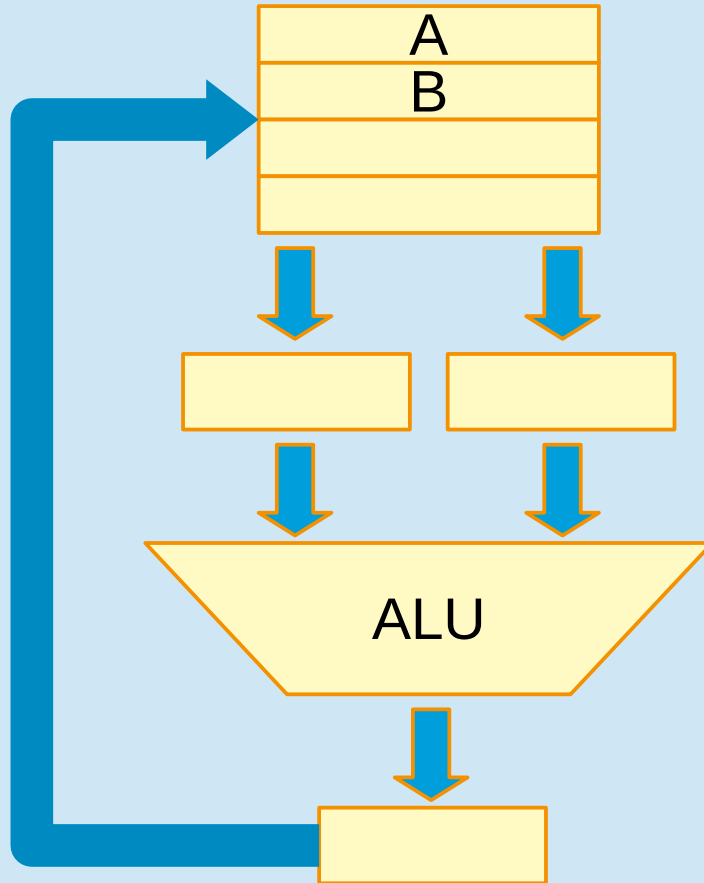


Organizzazione del Computer

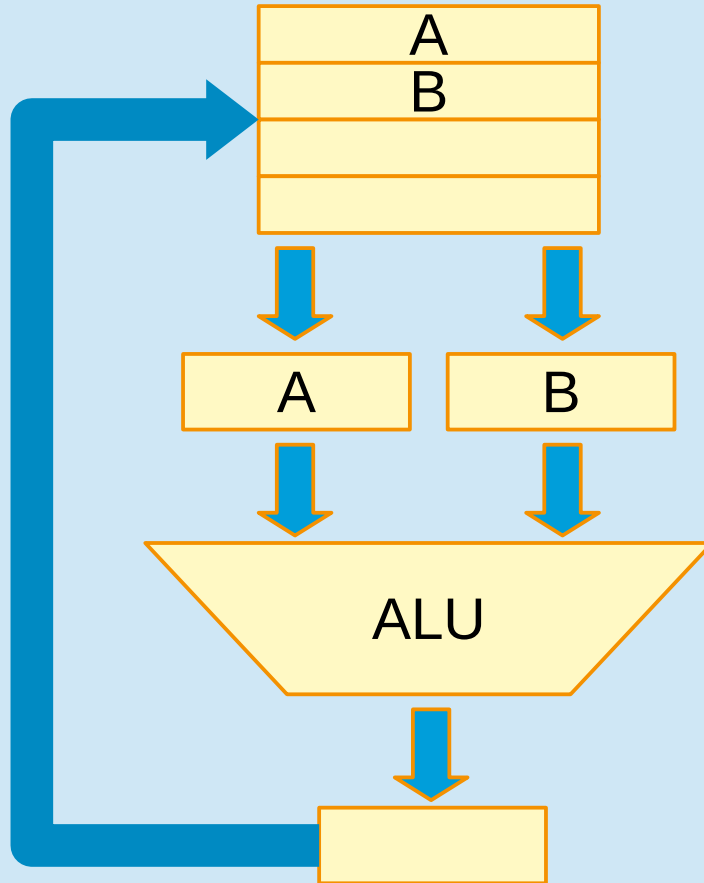
Instruction SET

E' l'insieme (set) di istruzioni supportate dalla CPU.

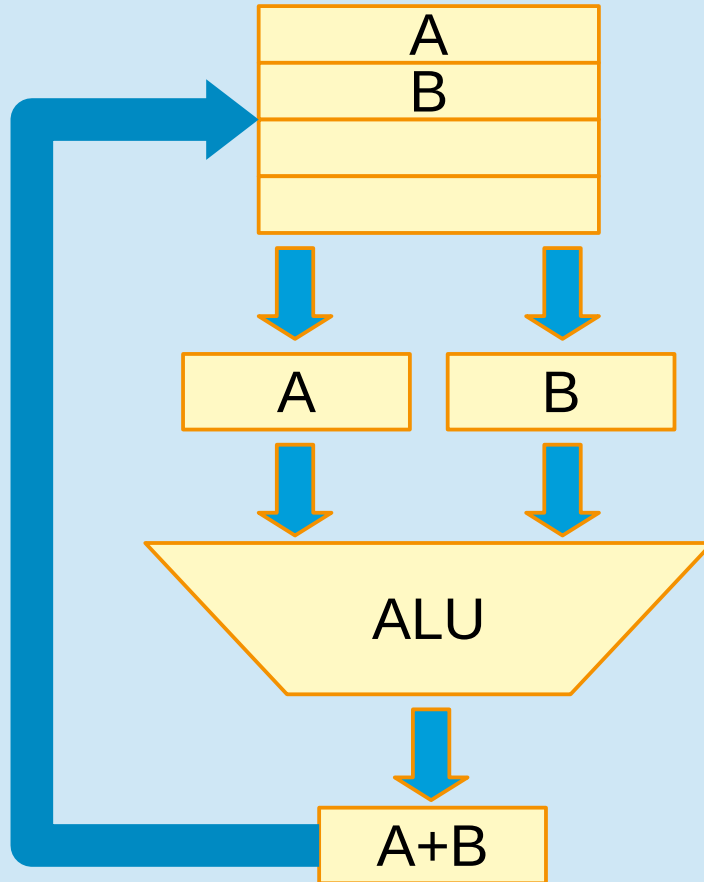
Organizzazione della CPU - 1



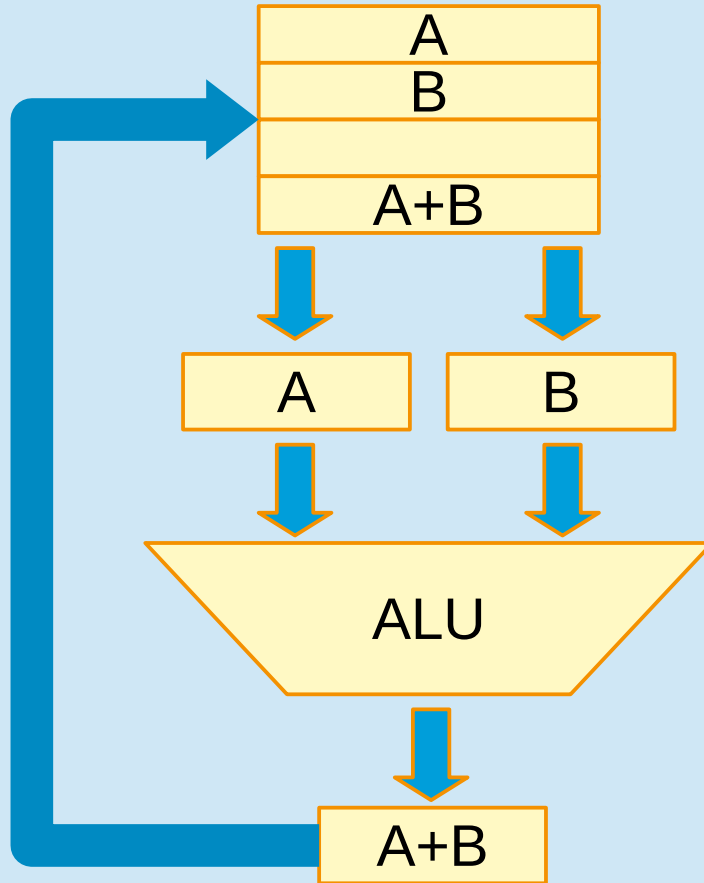
Organizzazione della CPU - 1



Organizzazione della CPU - 2



Organizzazione della CPU - 3



Esecuzione istruzioni

Fasi di esecuzione

- 1.FETCH : Legge istruzione
- 2.DECODE : Decodifica istruzione
- 3.MEMORY : Legge da memoria
- 4.EXECUTE : Esegue istruzione
- 5.WRITE BACK : Scrive su memoria

Interprete in versione Software

L'esecuzione istruzioni può essere eseguita anche da un **programma software**, invece che da strutture hardware.

In questo caso il programma si chiama **'INTERPRETE'** perché interpreta i comandi

Esempio di interprete software

```
Int PC = 0 // La prima istruzione numero 0
Int AC // Registro accumulazione
Int INSTR
Int INSTR_TYPE
Int DATA_ADDRESS
Int DATA

While True:
    INSTR = Memory(PC)
    PC = PC + 1
    INSTR_TYPE = Decode(INSTR)
    DATA_ADDRESS = Fetch(INSTR_TYPE)
    if DATA_ADDRESS > 0:
        data = Memory(DATA_ADDRESS)
    Execute(INSTR_TYPE, DATA)
```

Fase Execute: Scrive internamente sul registro AC

Fase Write Back: Assente perche non scrive su memoria

Motivi di un interprete software

L'interprete dei comandi, realizzato in software, e' utile perche consente di ampliare l'INSTRUCTION SET senza realizzare nuovo hardware (sarebbe molto costoso) .

L'interprete si comporta come una CPU simulata, che viene chiamata 'Virtual Machine'.

Questa tendenza negli anni 80 ha causato la proliferazione di sistemi con moltissime istruzioni complesse chiamate **CISC**:
Complex Instruction Set Computer

Motivi di un interprete software

TODO ESEMPIO INTERPRETE SU CPU A LAYER

CISC e RISC

Durante la fase storica in cui i sistemi CISC si stavano diffondendo, altri sperimentavano sistemi con meno istruzioni, più semplici, il cui hardware però era molto più veloce: I sistemi **RISC**
Reduced Instruction Set Computer.

CISC e RISC

TODO: INTEL -> Ibrido: CISC lente interpretate per istruzioni rare, RISC non interpretate -> + velocità per istruzioni frequenti
Reduced Instruction Set Computer.

Pipeline

La pipeline e' un sistema per permettere di aumentare le istruzioni eseguite per ciclo di clock, senza pero' velocizzare la singola istruzione.

TODO: esempio pipeline

Pipeline

PIPELINING

Storia: memoria e prefetch buffer

Pipeline: fetch, decode, read registers, exec, write back

Superscalare: Piu ALU in stadio 3

TODO

Aumento di prestazioni CPU

Concetti importanti per aumentare prestazioni (MIPS) :
Istruzioni direttamente eseguite da hw sono piu veloci

Ottimizzare la velocita di INIZIO delle istruzioni
(Massimizzare i MIPS e nn la singola istruzione)

Istruzioni facili da decodificare

Solo load e store devono accedere alla memoria (sono lente)

Molti registri evitano continui accessi lenti alla cache o alla RAM

