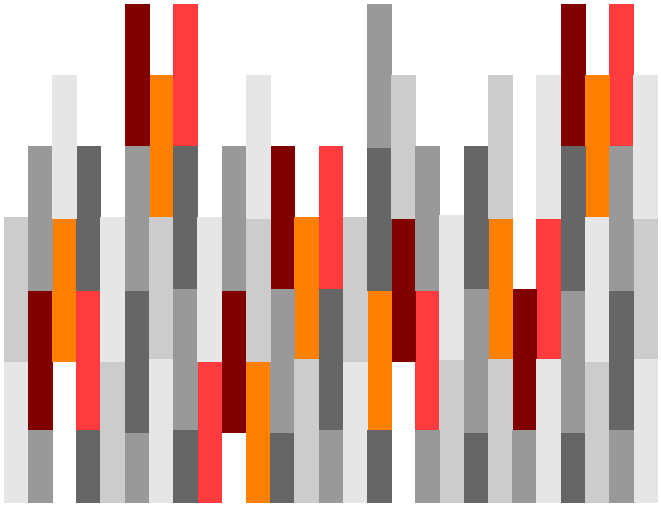
Fluency



Come funzionano i computer

Capitolo 6

Cosa sanno fare

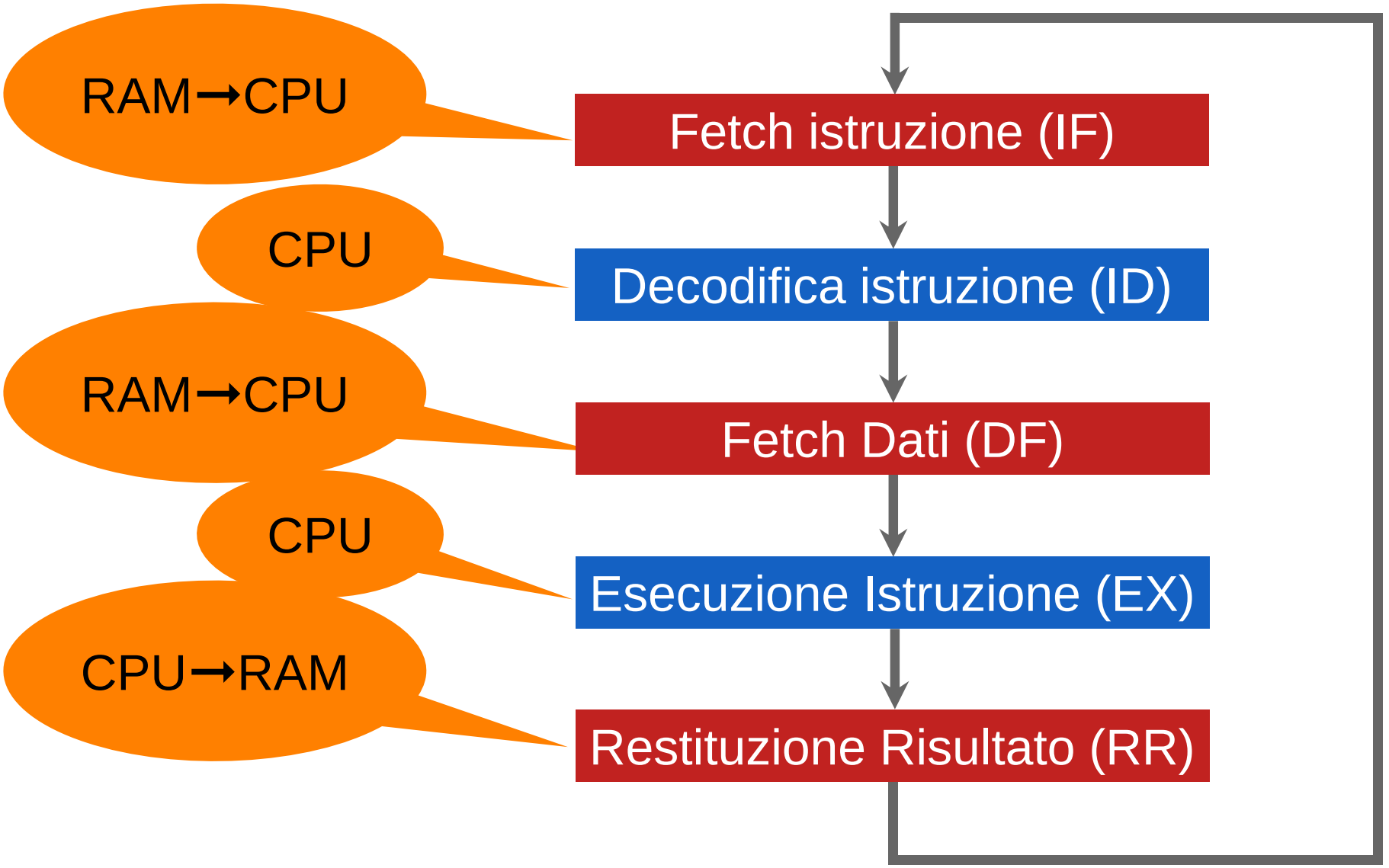
- ***Esecuzione deterministica***
  - istruzioni per elaborare dati
- Devono ricevere una ***serie di istruzioni*** da seguire

Cosa non sanno fare

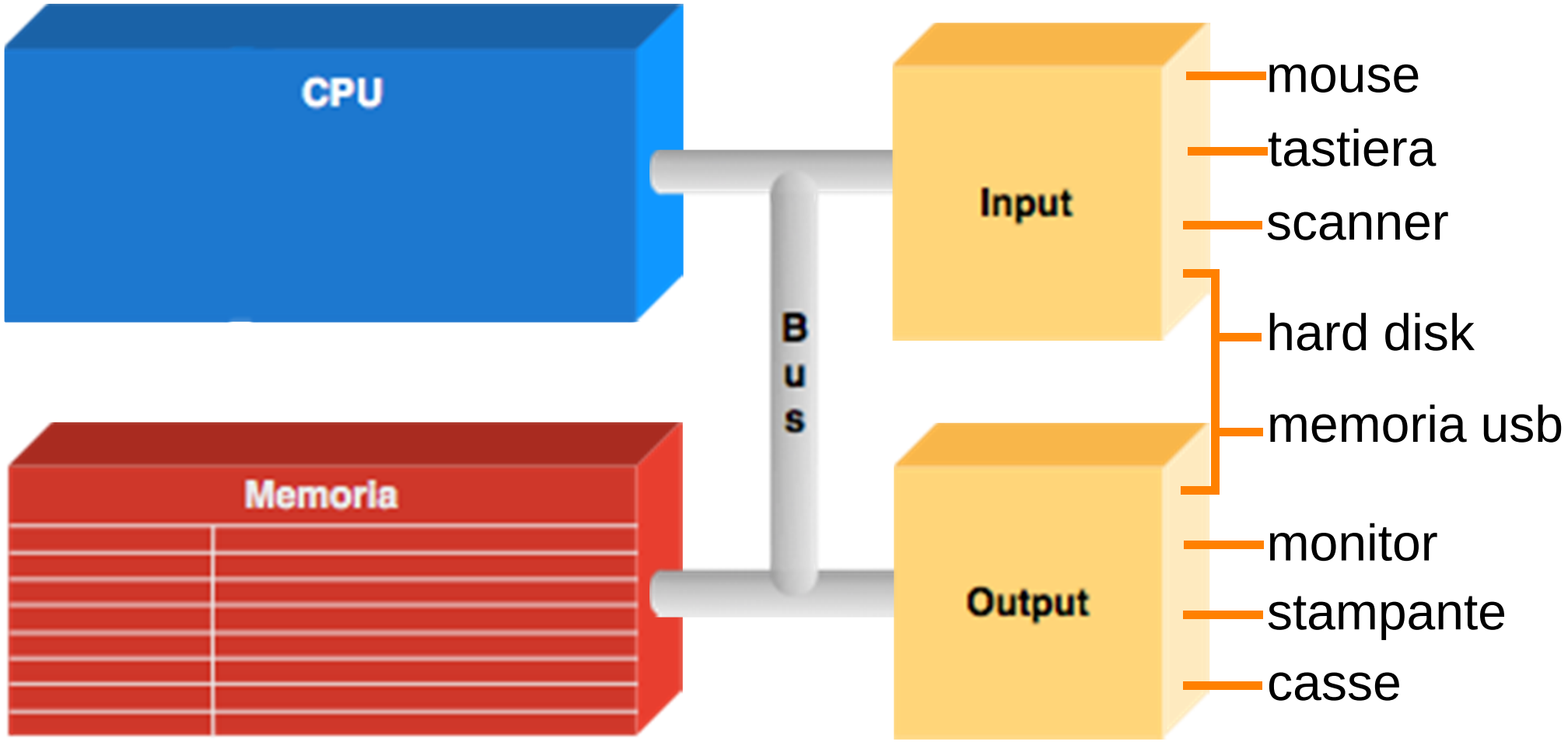
- Non hanno ***immaginazione*** né ***creatività***
- Non hanno alcun ***intuito***
- Non hanno ***senso dell'humor***, ***ironia***, ***decoro***, …
- Non **scherzano**
- Non sono ***crudeli*** né ***vendicativi***
- Non hanno ***libero arbitrio***

Cosa hanno in comune?

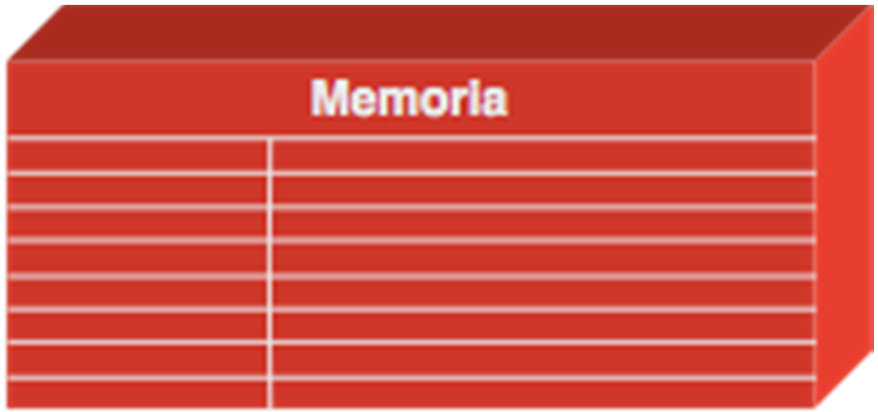
Ciclo Macchina



Architettura generale



RAM



RAM

- Contiene
  - il programma in esecuzione
  - i dati su cui il programma stesso opera

Caratteristiche della RAM

- **Locazioni discrete**
  - ogni locazione è di 1 byte
- **Indirizzi**
  - univoco per ogni locazione (intero a partire da 0)

Caratteristiche della RAM

- **Valori**
  - le locazioni di memoria memorizzano valori
- **Capacità finita**
  - sia nel numero di locazioni
  - sia nella capacità di ogni locazione

Locazioni da un byte

Locazioni da un byte

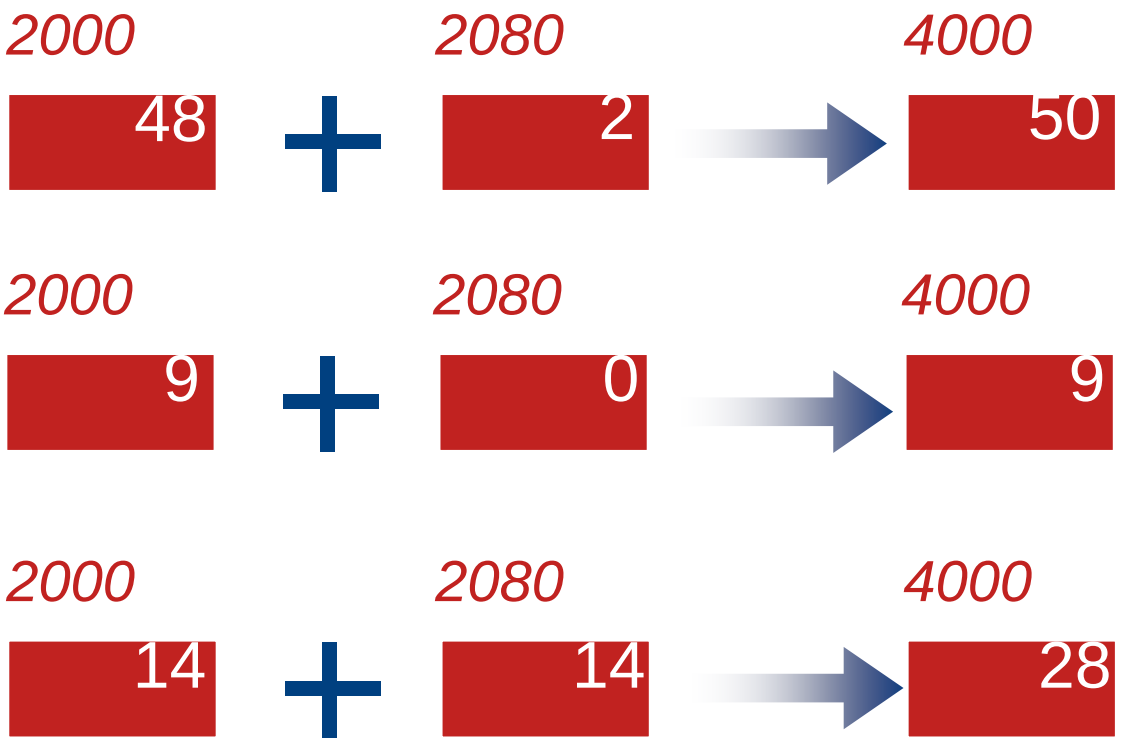
- Ogni locazione contiene
  - un carattere ASCII
  - oppure un numero in [0 , 255]
- Parola di memoria
  - blocchi di 4 byte usati come singola unità

RAM

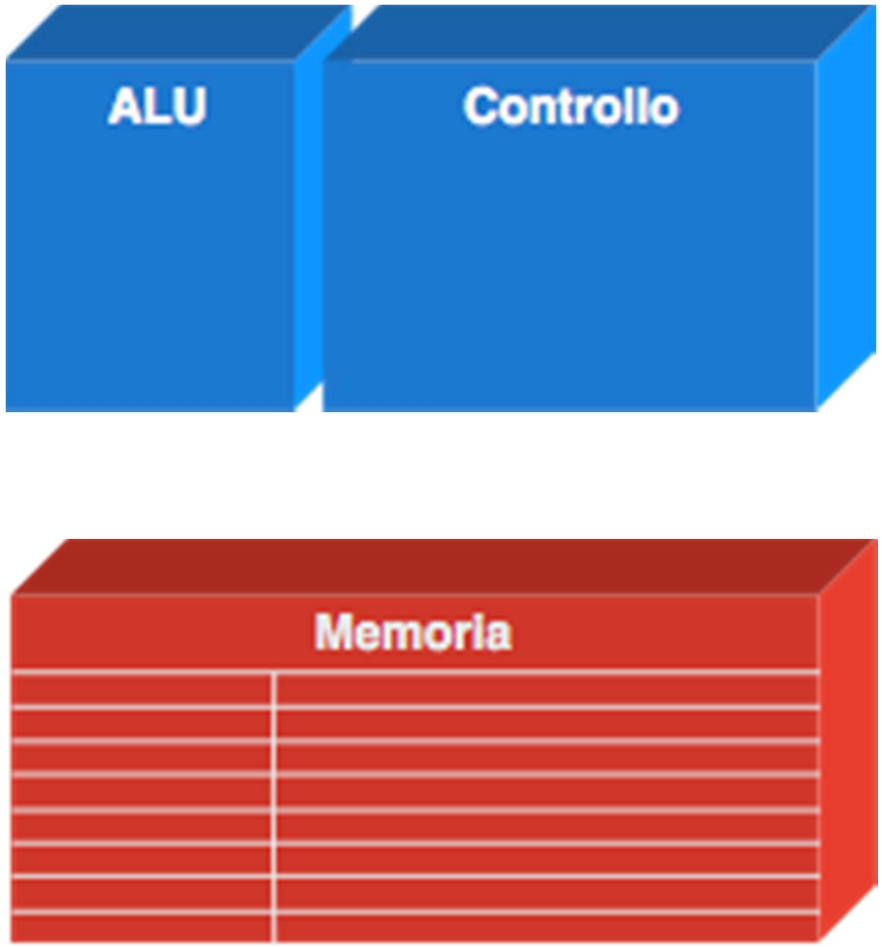
- Accesso *Casuale* o *diretto*
  - il computer può accedere direttamente a qualsiasi locazione di memoria
- Ordine di grandezza: **gigabyte (GB)**
- Avere molta memoria è preferibile
  - evita problemi di spazio per i programmi e i dati

Risultati operazione

ADD 4000, 2000, 2080



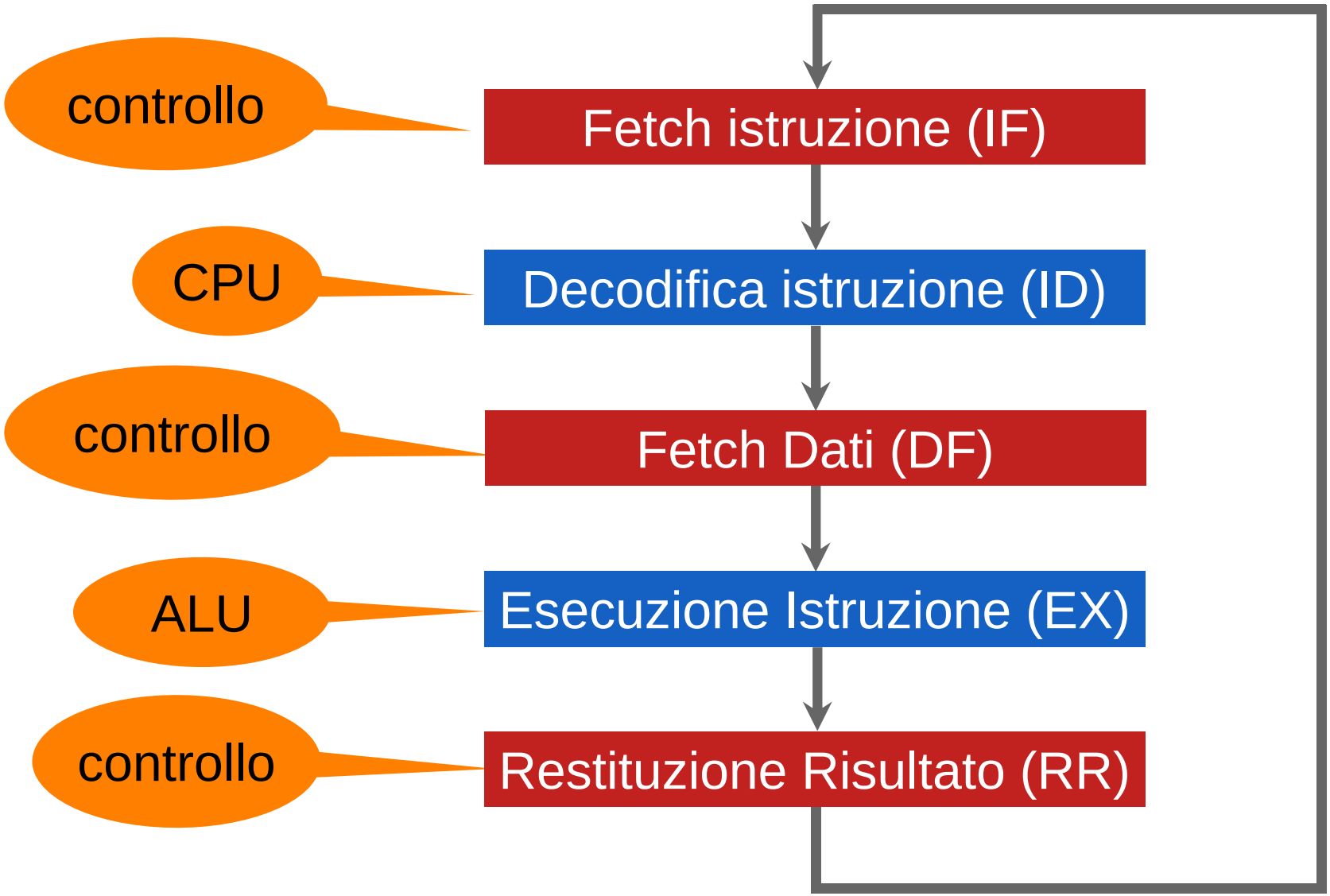
CPU



CPU

- Implementa il ciclo macchina
  - direttamente via hardware

Ciclo macchina



Istruzione esempio

- Ha la forma: ***ADD 4000, 2000, 2080***
  - somma i numeri nelle locazioni ***2080*** e ***2000***
  - scrive il risultato nella locazione di memoria ***4000***

Istruzione esempio

- Il passo di ***Fetch Dati***
  - deve estrarre i due valori
- Il passo ***Restituzione Risultato***
  - inserirà la somma nella locazione 4000

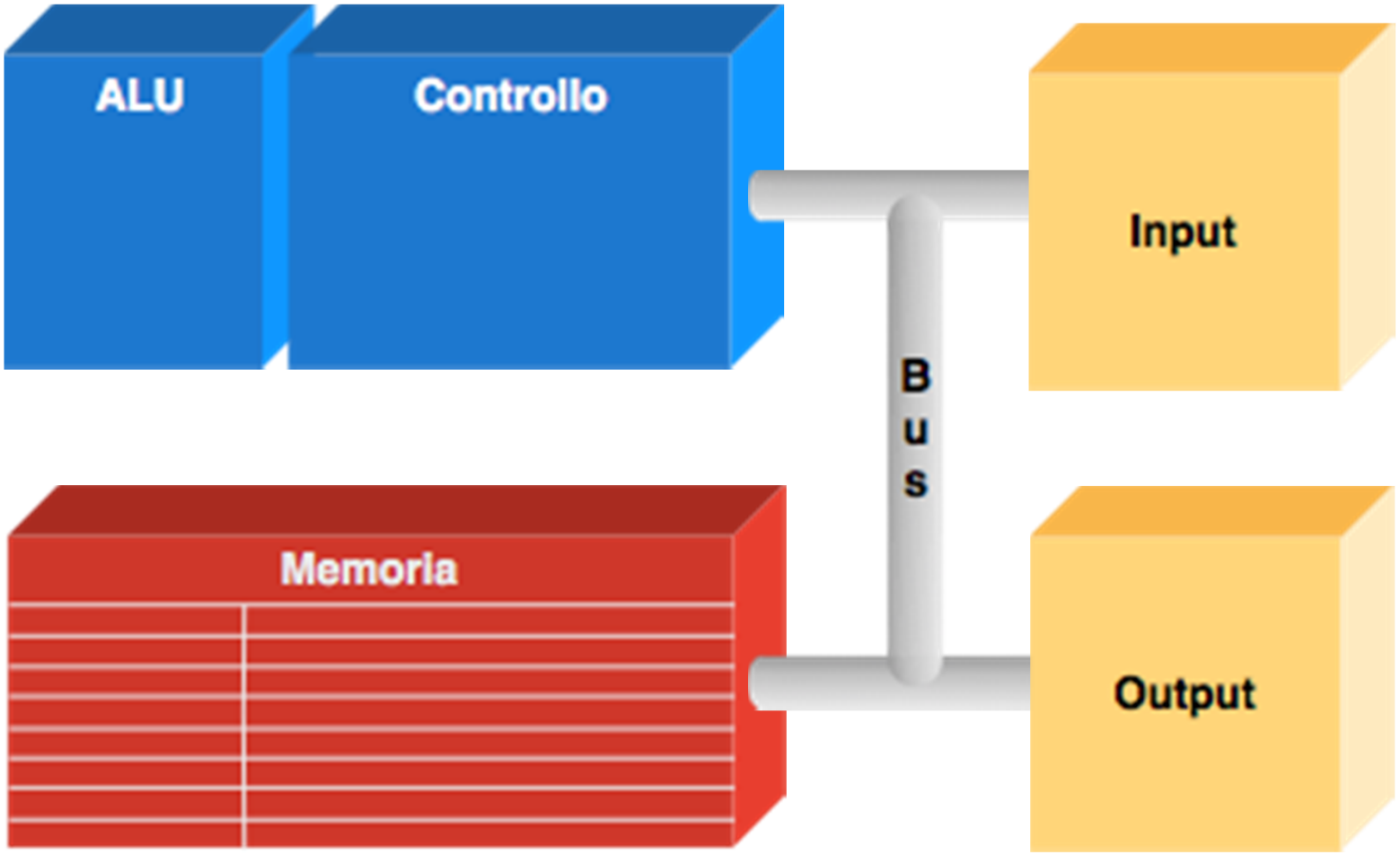
ALU

- *Arithmetic Logic Unit*
- Esegue tutti i calcoli
  - il passo ***esecuzione istruzione***
- Circuito della ALU per la somma
- Si sono altri circuiti
  - dedicati alla moltiplicazione, al confronto, …

ALU

- Le istruzioni di trasferimento dei dati **non usano** la ALU
  - il passo ***Fetch Dati***
  - il passo ***Restituzione Risultato***

Input e output



Input e output

- Le ***periferiche***
  - *si collegano* alle porte di input/output
  - *non sono parti* vere e proprie del computer
  - *specializzate* per codificare o decodificare i dati scambiati col mondo esterno

Hard disk

- *Periferica alfa*
- *Chiavi USB* e *h*
  - hanno entram *tput*
  - archivio *“permanente”* di dati
  - vita “attesa”

Obsolescenza della tecnologia

Le periferiche

- La ***tastiera***
  - trasforma le battute sui tasti in formato binario
- Il ***monitor***
  - rappresenta i dati contenuti nella memoria video

Driver delle periferiche

- Le periferiche sono “stupide”
- Driver
  - ***traduzione*** tra fenomeno fisico e segnale binario
- Il computer fa tutto il resto
  - *interpreta* il segnale binario riportato dalla periferica
  - *prepara* l’output

Un “PC” nel PC

- *Program Counter*
- Qual’è la *prossima l’istruzione* da eseguire?
  - l’indirizzo è memorizzato nell’unità di controllo

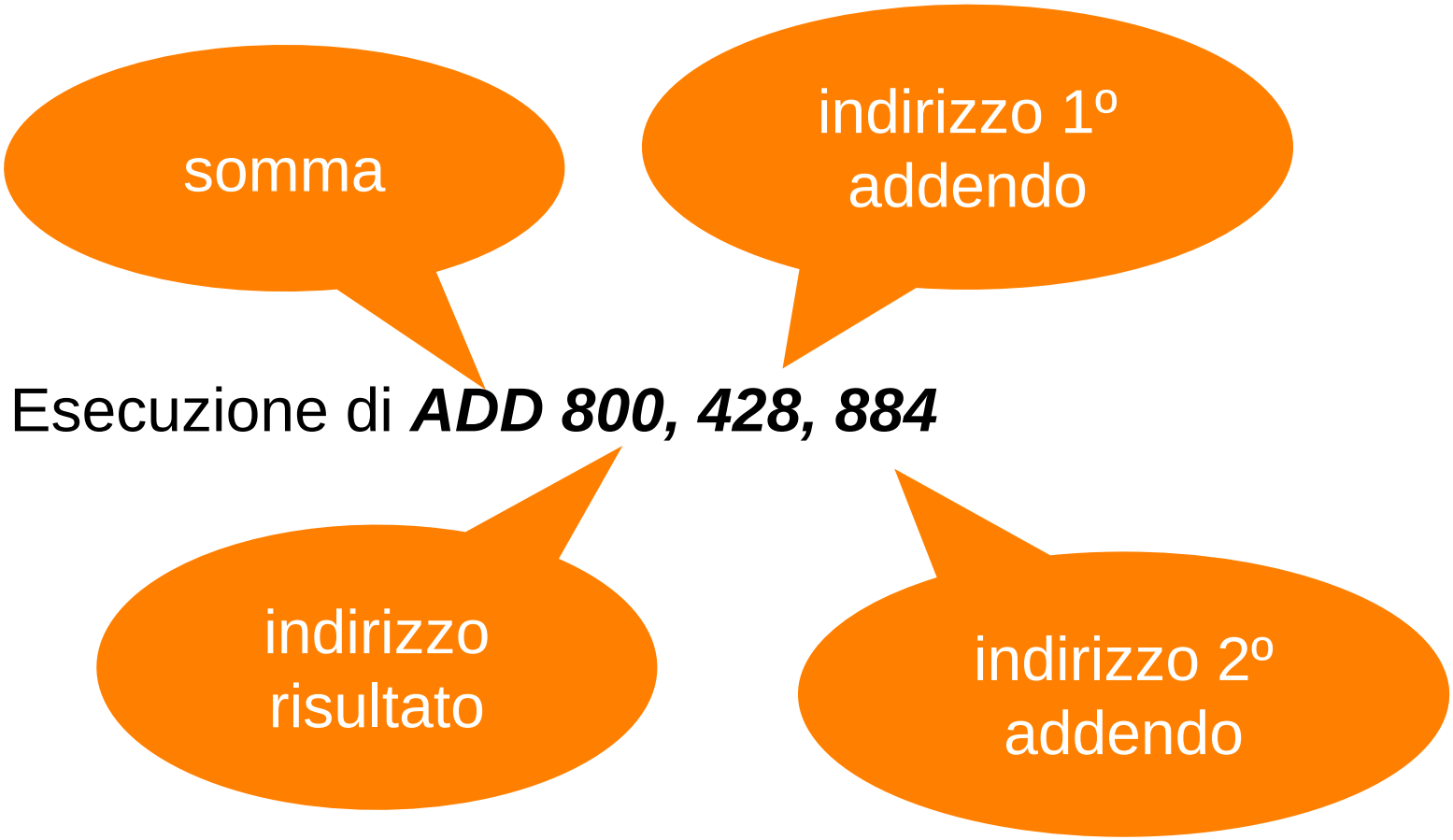
Aggiornamento del PC

- Al fetch di una nuova istruzione
  - il PC è incrementato di 4
- Al prossimo fetch
  - il PC “punta” all’istruzione giusta

Decodifica delle istruzioni

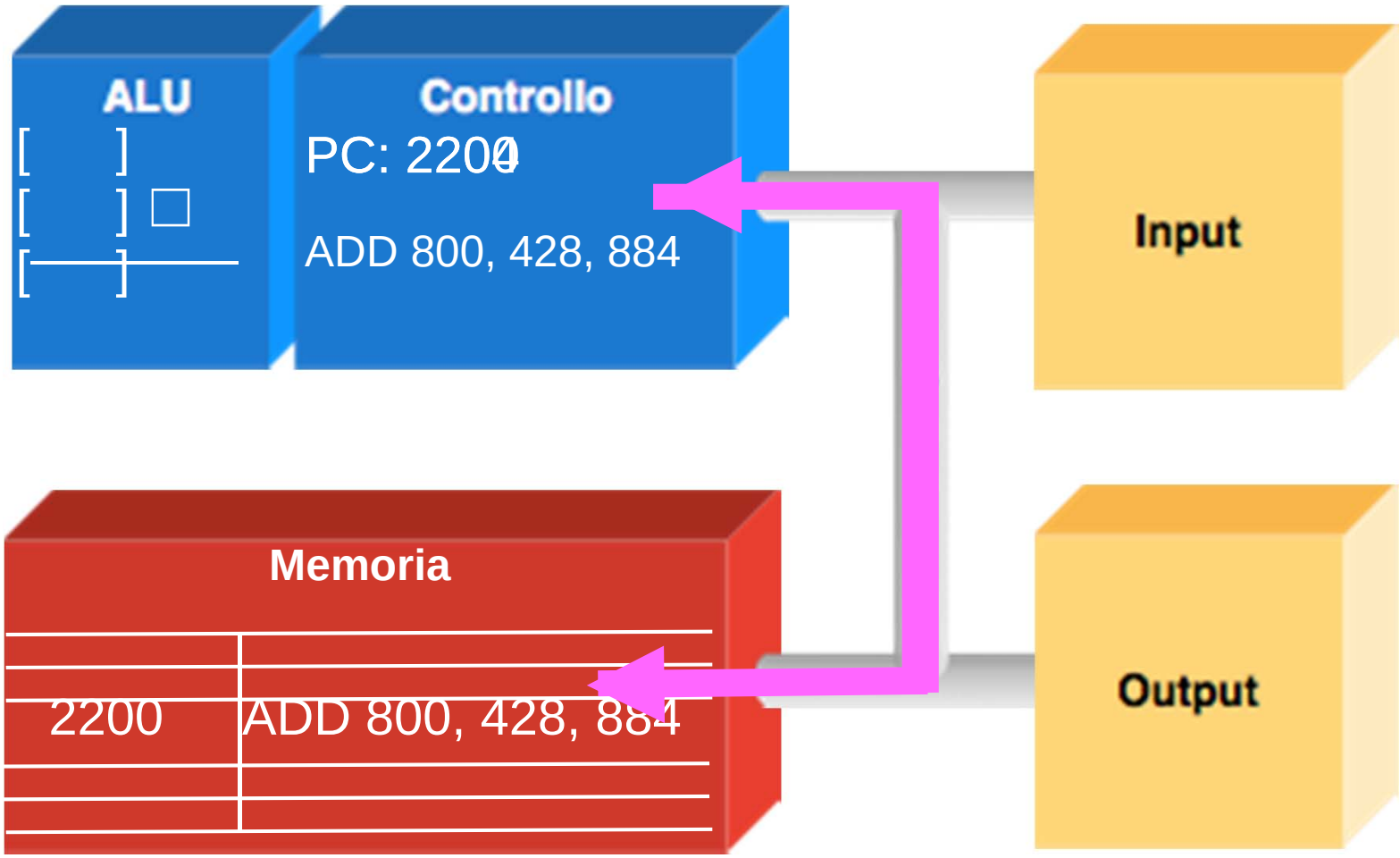
- Esecuzione di un programma
  - il computer interpreta i nostri comandi
  - espressi nel suo *proprio linguaggio*

ADD 800, 428, 884

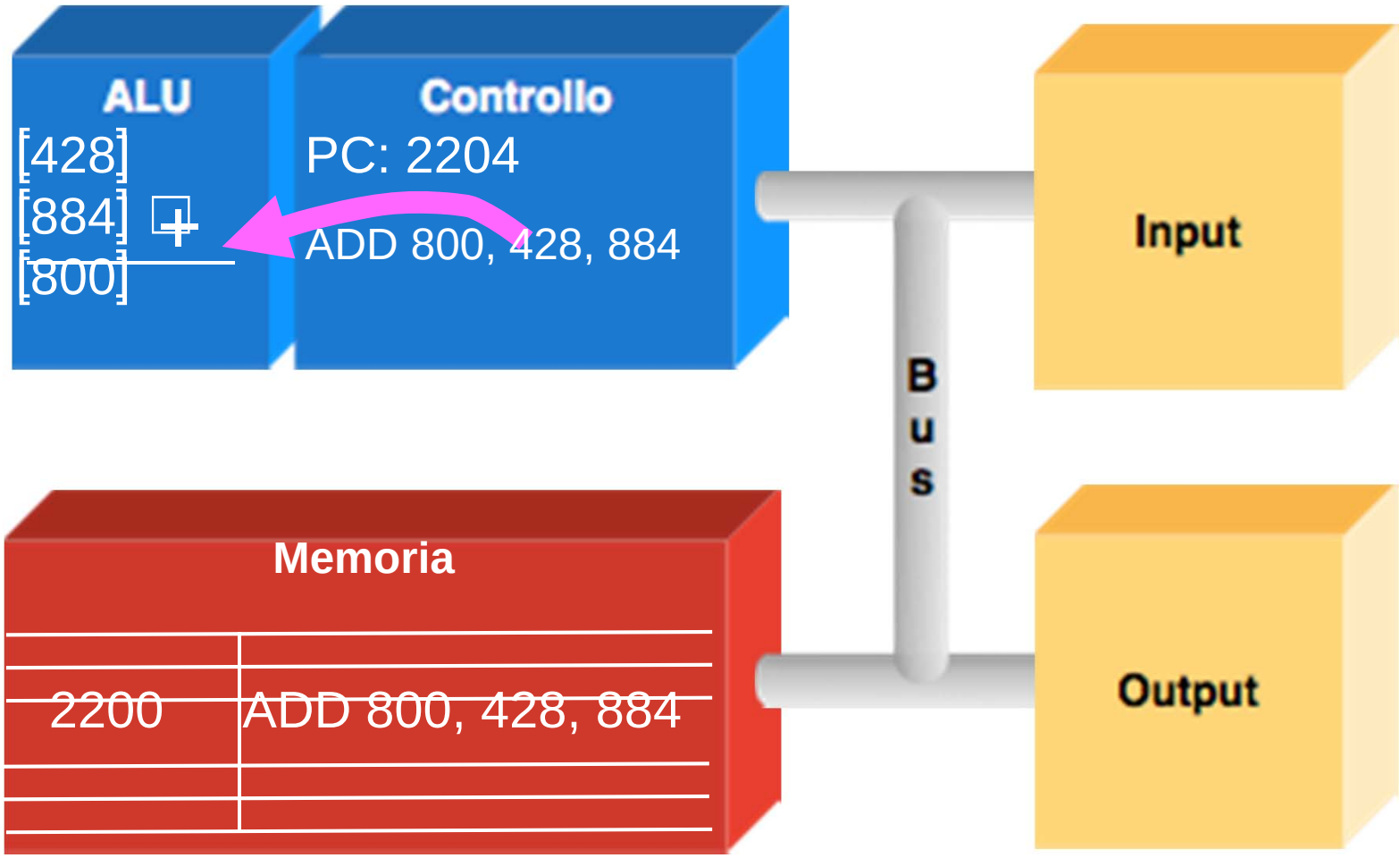


- Esecuzione di ***ADD 800, 428, 884***

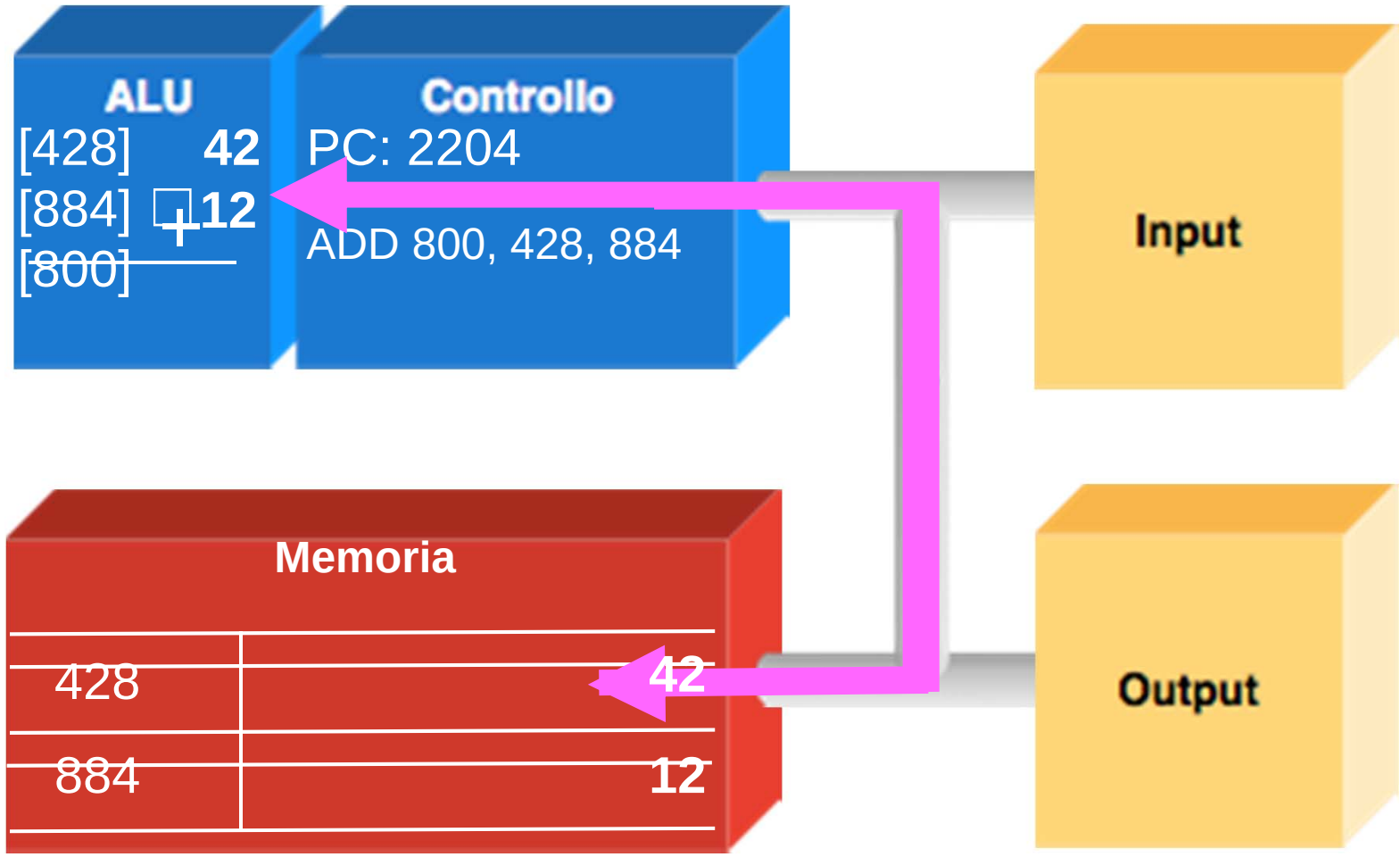
Fetch istruzione



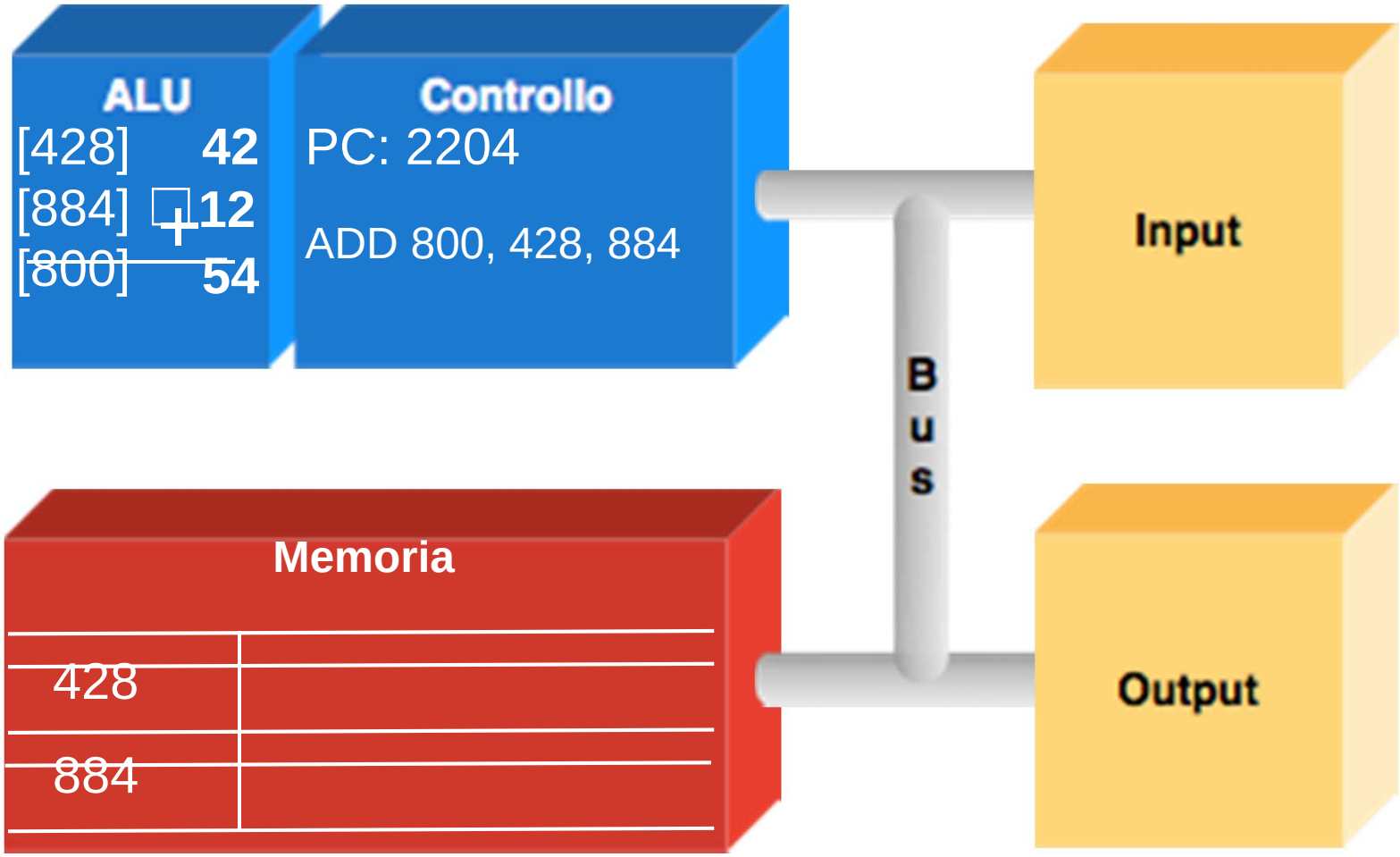
Decodifica


ALU
[428]
[884]
[800]
Controllo
PC: 2204
ADD 800, 428, 884
Input
Bus
Memoria
2200
ADD 800, 428, 884
Output

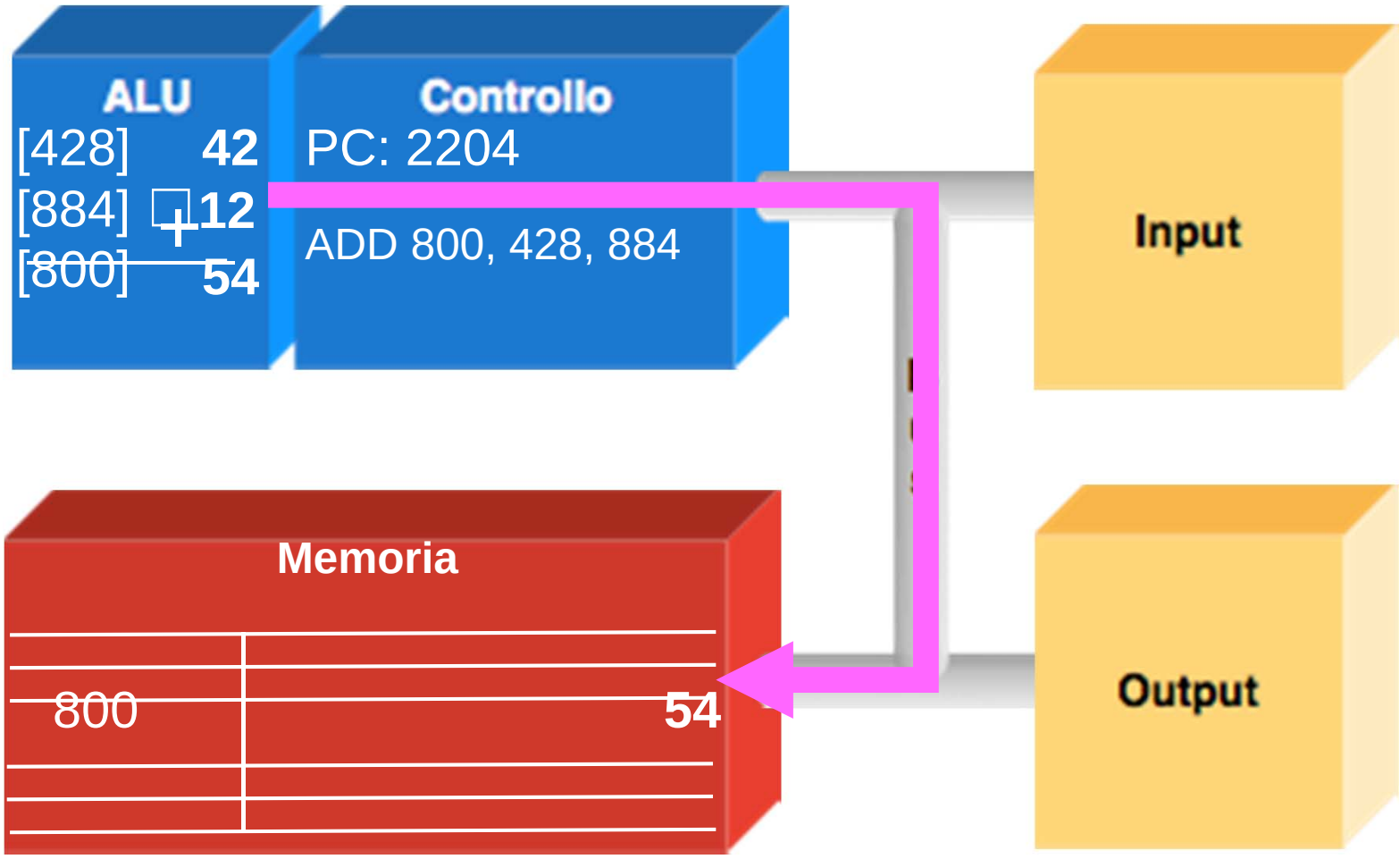
Fetch degli operandi



Esecuzione istruzione



Restituzione risultato



Velocità del ciclo macchina

- In un secondo esegue un enorme numero di istruzioni
- Il clock del computer
  - determina la velocità del ciclo macchina
  - misurato in Hertz (Hz)
- ODG del **miliardo** di cicli al secondo (GHz)

La frequenza è importante?

- I computer moderni
  - cominciano un'istruzione a ogni ***tick*** del clock
- I circuiti si passano l'istruzione l'un l'altro (pipelining)
  - possono essere processate 5 istruzioni simultaneamente

Software visto dal computer

- Vede un *oggetto binario* (**codice macchina**)
  - una sequenza di *parole* (*word*, gruppi di 4 byte)
  - l'unico che comprende e sa eseguire

```
. . .
10001111 10010100 00000011 01110100
10001111 10011000 00000001 10101100
00000010 10011000 10100000 00100000
10101111 10010100 00000001 10010000
. . .
```

Il linguaggio assembly

- Alternativa al linguaggio macchina
  - usa lettere e numeri
- Di più facile comprensione per le persone
- Traducibile automaticamente in linguaggio macchina

Assemblare

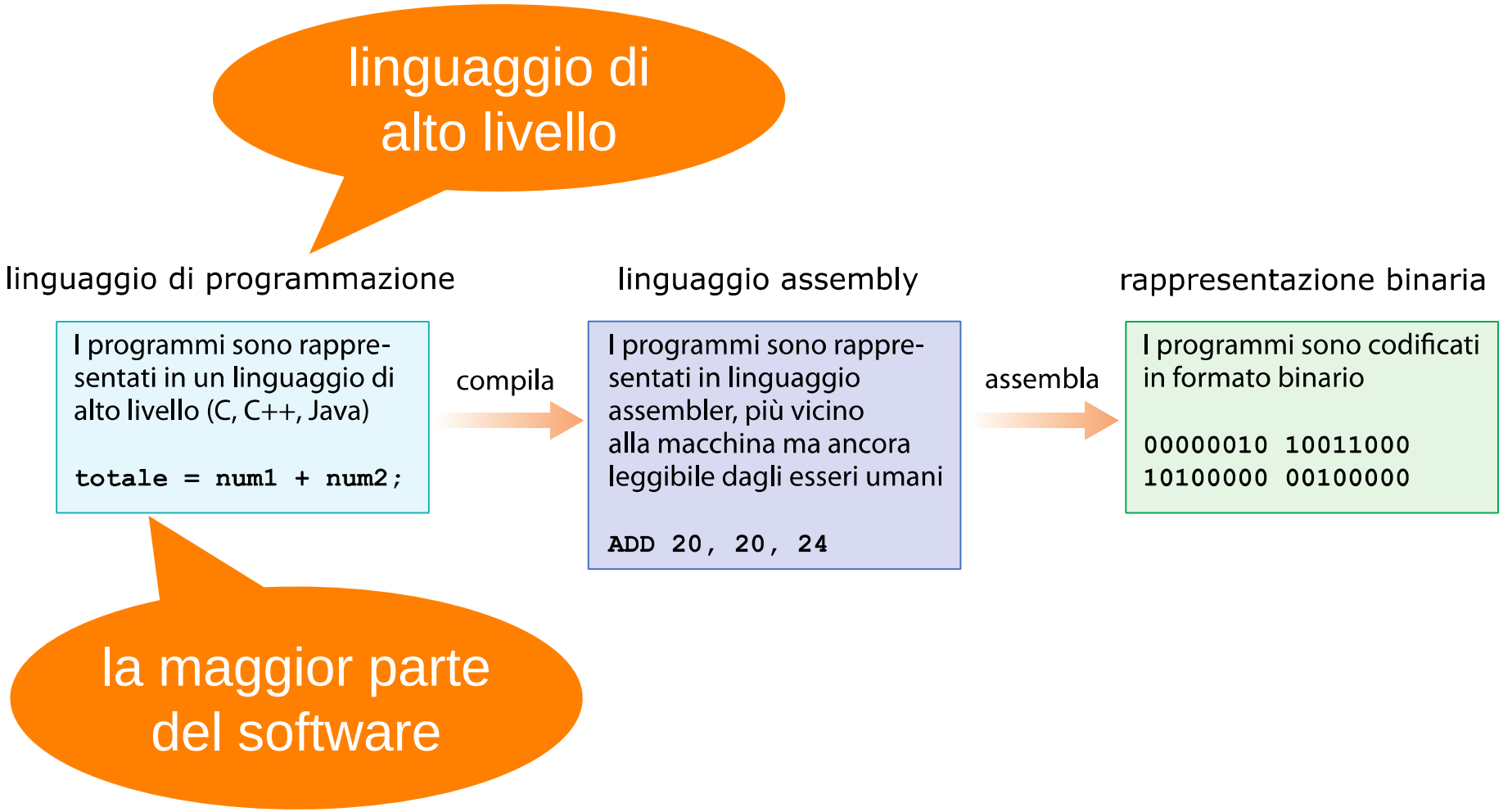
- Il computer scandisce il programma assembly
  - quando incontra una parole chiave
    - cerca in una tabella la corrispondente sequenza binaria
    - *assembla* le varie parti dell'istruzione
      - "costruisce" l'istruzione macchina

Linguaggi di alto livello

- La maggior parte del software è scritta così
- Prima **compilato** in linguaggio assembly
- Poi **assemblato**
  - per ottenere un *file binario*

Compilare

- Il processo è svolto da un’applicazione
- Dai costrutti del linguaggio di alto livello all’assembly
- Scritto in un **linguaggio di alto livello**
  - p.e. C o Java


linguaggio di alto livello
linguaggio di programmazione
I programmi sono rappre-sentati in un linguaggio di alto livello (C, C++, Java)
totale = num1 + num2;
compila
linguaggio assembly
I programmi sono rappre-sentati in linguaggio assembler, più vicino alla macchina ma ancora leggibile dagli esseri umani
ADD 20, 20, 24
assembla
rappresentazione binaria
I programmi sono codificati in formato binario
00000010 10011000
10100000 00100000
la maggior parte del software

Eseguire un’applicazione

- Le istruzioni macchina vengono **trasferite** dal disco nella memoria RAM
- Il ciclo macchina **esegue** le istruzioni
- Tutte le istruzioni del computer sono eseguite dai circuiti contenuti nell’ALU

Sistemi operativi

- Offrono le operazioni base per l’uso del computer
  - non supportate direttamente dall’hardware
- I tre più usati per personal computer:
  - Microsoft Windows
  - Apple MacOS X
  - Linux (Unix)

GUI

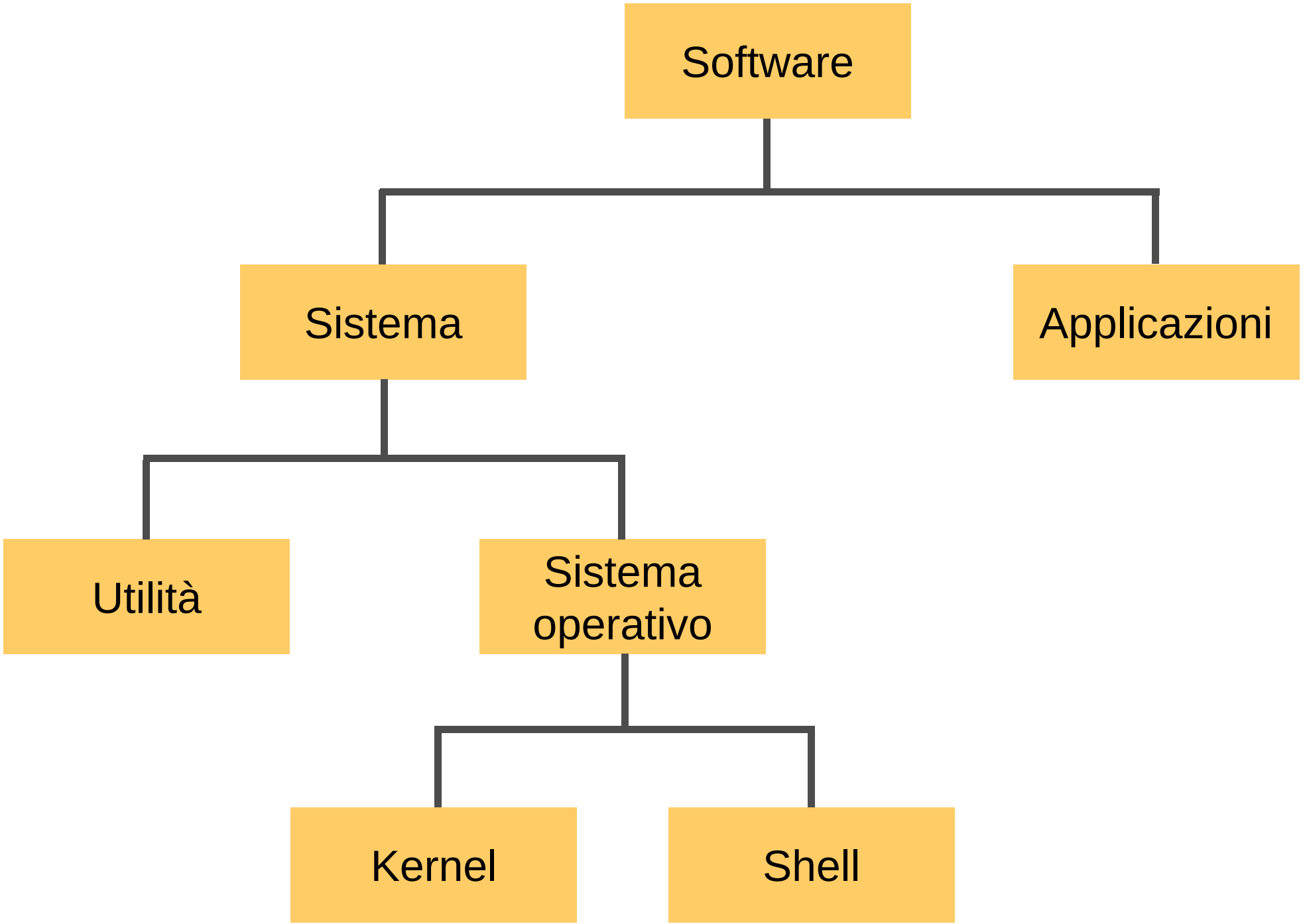
- Sono “impacchettati” e forniti con l’OS
  - bordi delle finestre,
  - le barre di scorrimento,
  - i pulsanti,
  - i puntatori, ecc.

Caratteristiche

- ***Multitasking***
  - può eseguire più processi per volta
  - gestione dei programmi in esecuzione
- ***Multiuser***
  - più utenti in “contemporanea”

Time sharing

- Suddivide il tempo in intervalli
  - frazioni di secondo
- Ad ogni intervallo:
  - un solo programma è in esecuzione
  - ciclicamente esegue tutti i programmi
- L’utente ha l’impressione di contemporaneità


Software
Sistema
Applicazioni
Utilità
Sistema operativo
Kernel
Shell

Shell

- Interfaccia con l’utente
  - contiene anche le GUI
- Speso confuso con il sistema operativo

Kernel

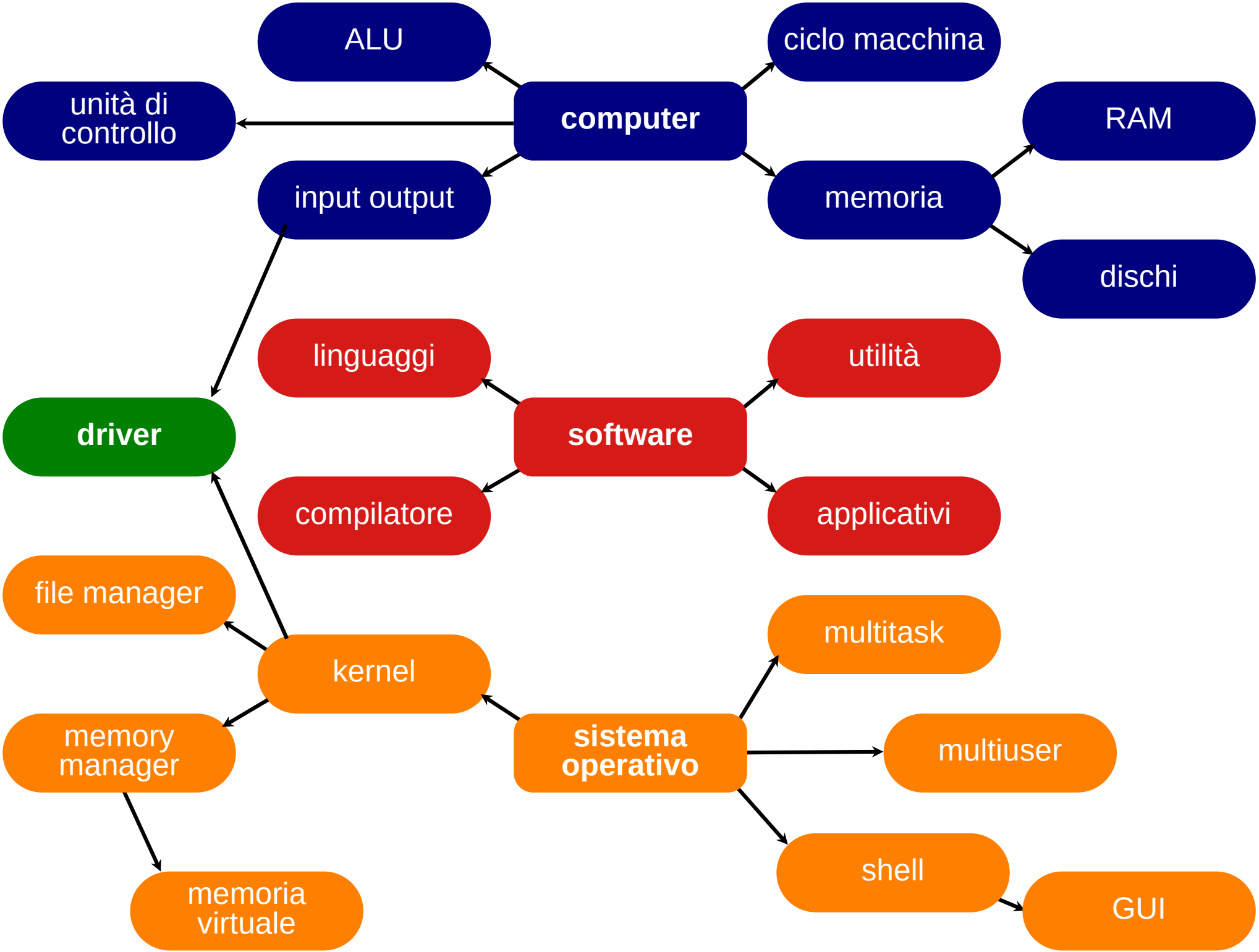
- Parte interna del sistema operativo
  - driver
  - file manager
  - memory manager

File manager

- Gestisce i file nella memoria di massa
  - permessi di accesso
  - allocazione
  - icone

Memory manager

- Gestisce la RAM e i processi in esecuzione
- Paginazione
- Memoria virtuale


ALU
ciclo macchina
unità di controllo
computer
RAM
input output
memoria
dischi
linguaggi
utilità
driver
software
compilatore
applicativi
file manager
multitask
kernel
memory manager
sistema operativo
multiuser
shell
memoria virtuale
GUI