Architettura di un elaboratore

Prof. Giulio Maraldi

Istituto di Istruzione Superiore *Marie Curie*
Savignano sul Rubicone
A.S. 2017/2018

Argomenti

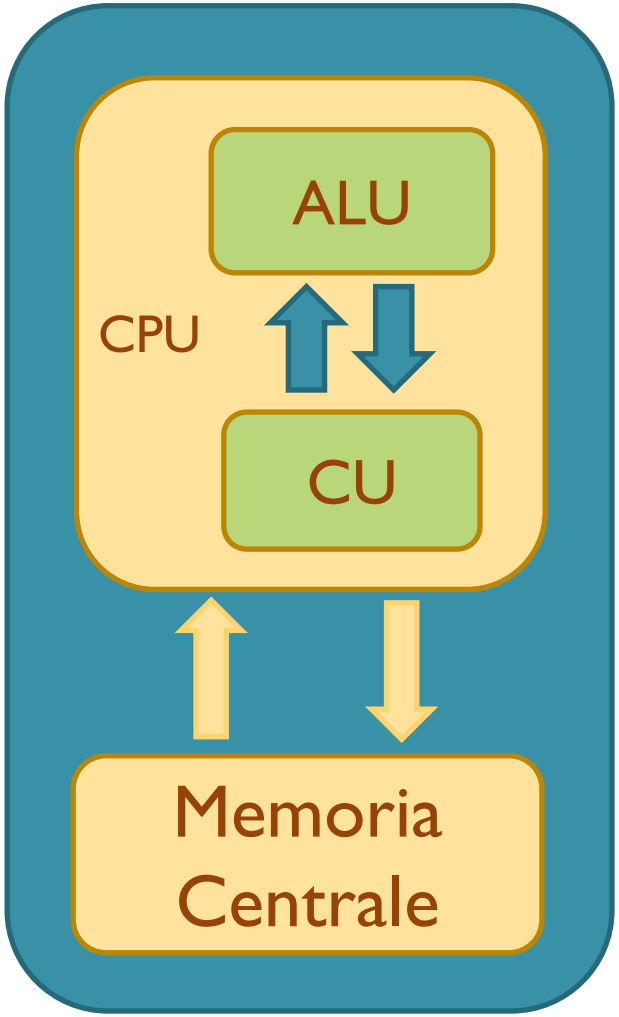
1. Componenti e struttura
2. CPU
3. Memorie
4. Periferiche di input/output

Argomenti

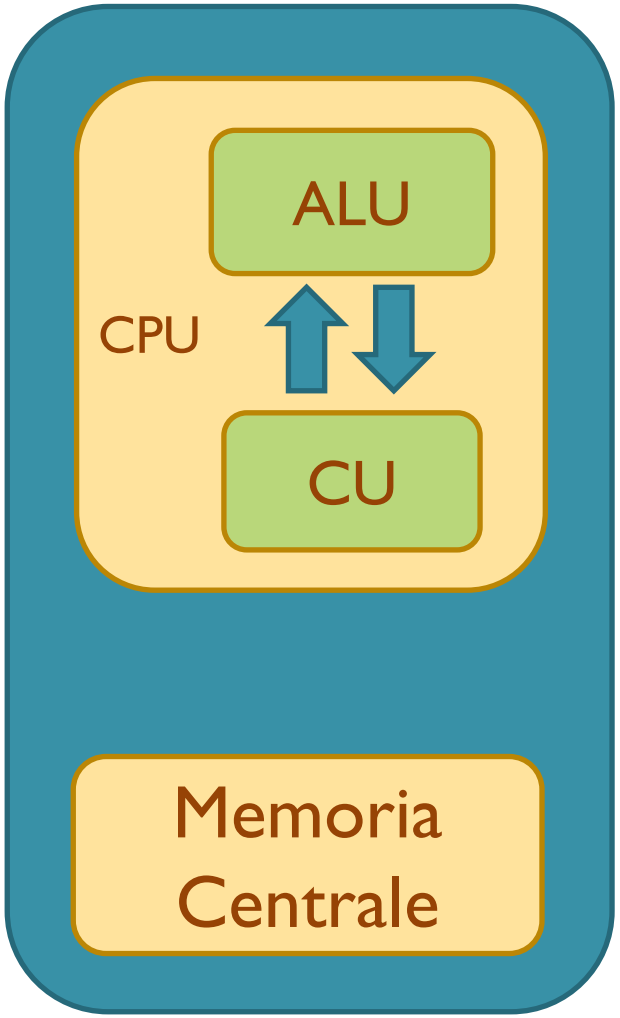
1. Componenti e struttura
2. CPU
3. Memorie
4. Periferiche di input/output

La macchina di Von Neumann

La macchina di Von Neumann

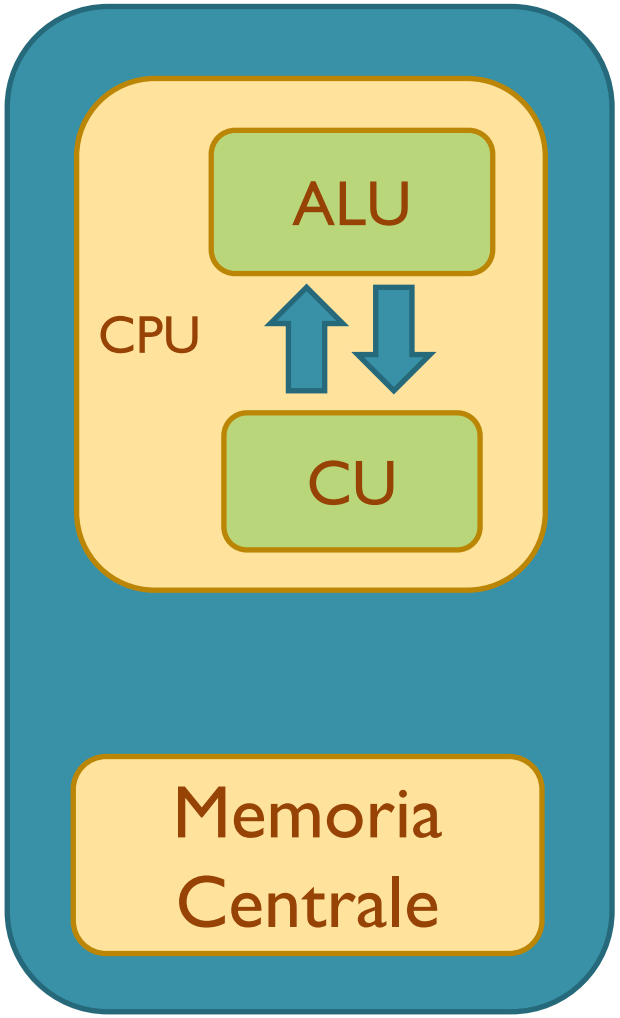


La macchina di Von Neumann



È un *modello architetturale:*

La macchina di Von Neumann



È un *modello architetturale:*

- Descrive le parti del sistema e la loro collocazione
- Descrive le interazioni tra le parti
- Definisce i compiti di ogni parte

La macchina di Von Neumann

È un *modello architetturale:*

STRUTTURA

- Descrive le parti del sistema e la loro collocazione
- Descrive le interazioni tra le parti
- Definisce i compiti di ogni parte

La macchina di Von Neumann

È un *modello architetturale:*

STRUTTURA

INTERAZIONE

- Descrive le parti del sistema e la loro collocazione
- Descrive le interazioni tra le parti
- Definisce i compiti di ogni parte

La macchina di Von Neumann

È un *modello architetturale:*

STRUTTURA

INTERAZIONE

COMPORTAMENTO

SONO LE 3 COMPONENTI **FONDAMENTALI** CHE USIAMO PER DESCRIVERE UN SISTEMA

STRUTTURA

INTERAZIONE

COMPORTAMENTO

Nell'informatica quando dobbiamo progettare qualcosa partiamo sempre dalla definizione di questi 3 aspetti

STRUTTURA

INTERAZIONE

COMPORTAMENTO

Quali sono i componenti del sistema, come sono fatti (descrive qualcosa di *statico*)

STRUTTURA

INTERAZIONE

COMPORTAMENTO

Come interagiscono i componenti tra di loro, come si scambiano le informazioni

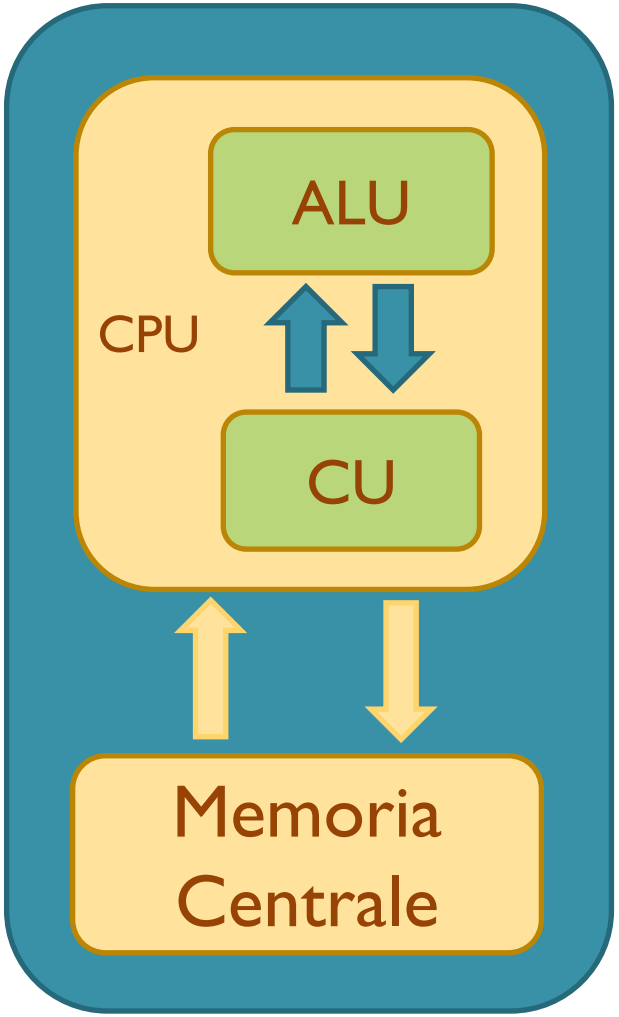
STRUTTURA

INTERAZIONE

COMPORTAMENTO

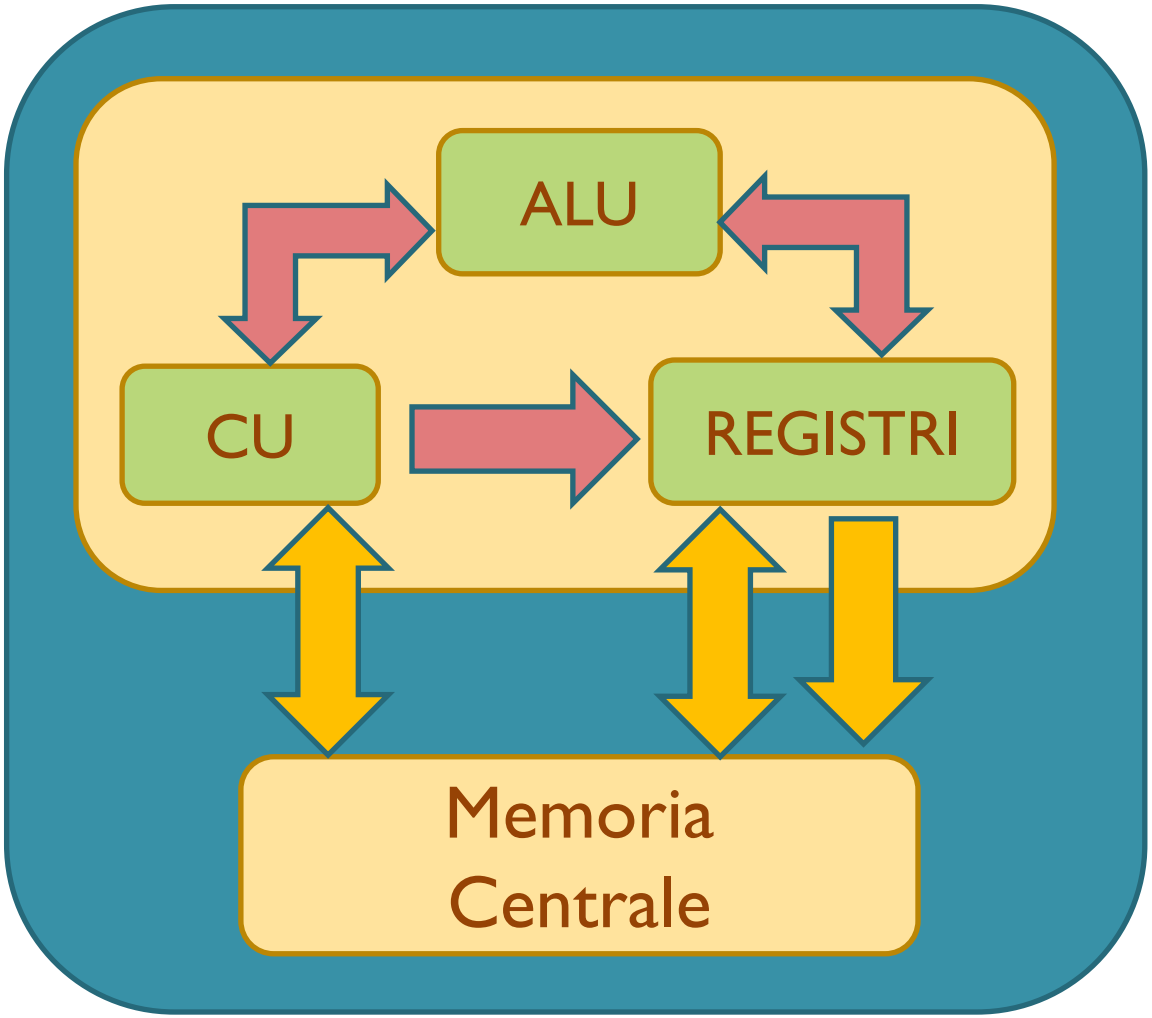
Quali azioni compiono i componenti e come reagiscono ai segnali esterni (descrive qualcosa di *dinamico*)

La CPU: struttura

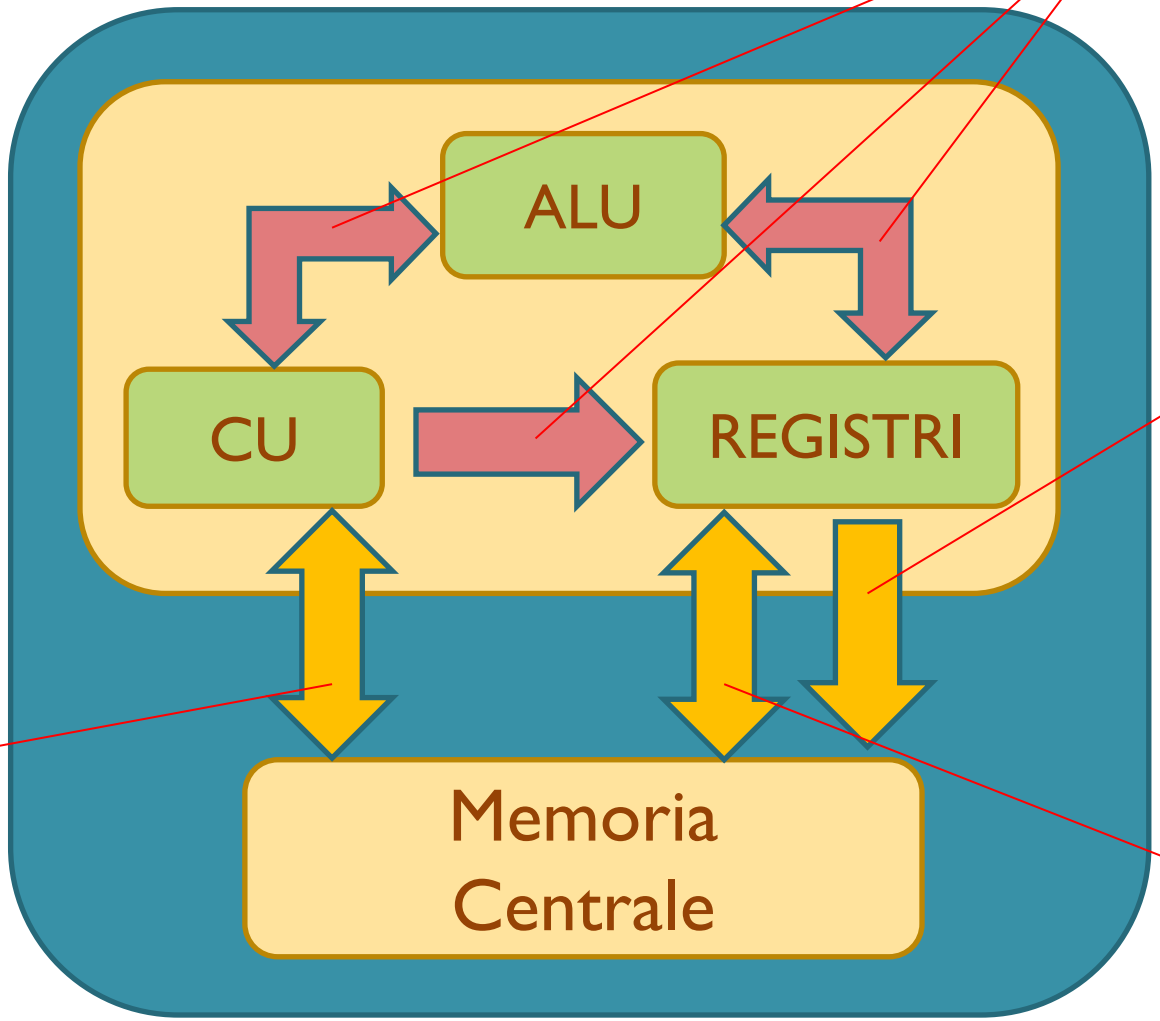


Un po' più in dettaglio...

La CPU: struttura



La CPU: interazione



BUS INTERNI

BUS INDIRIZZI

BUS DI CONTROLLO

BUS DATI

La CPU: i bus

BUS INTERNI

BUS

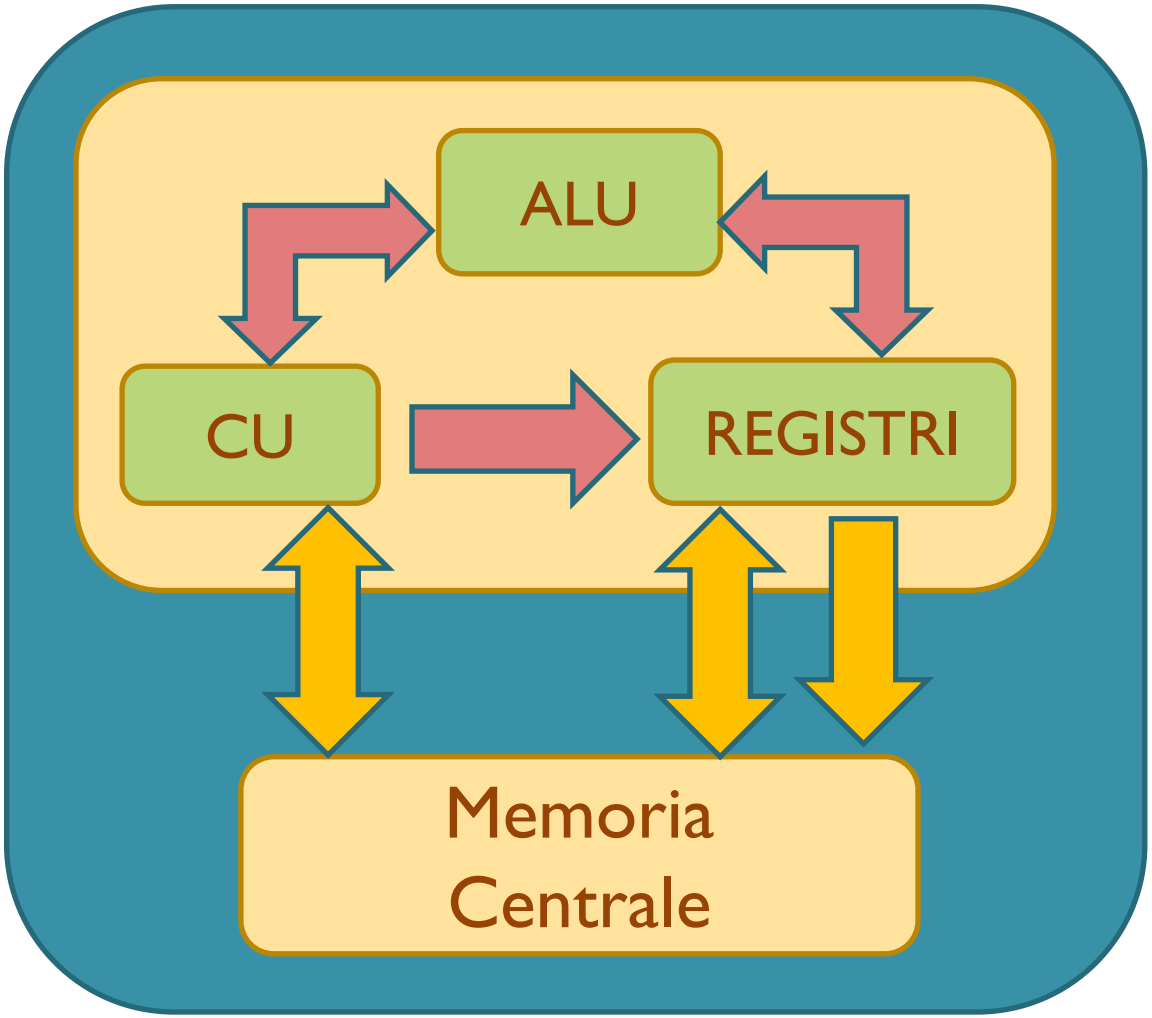
Sono i canali di comunicazioni che i componenti usano per scambiarsi i dati.

BUS INDIRIZZI

BUS DI CONTROLLO

BUS DATI

La CPU: struttura



La CPU: struttura

REGISTRI

La CPU: i registri

Sono le memorie interne della CPU. La ALU esegue le sue operazioni sui valori contenuti in queste memorie.

La CPU: i registri

È invece la CU a decidere quali valori spostare nei registri in base alle istruzioni del programma.

La CPU: i registri

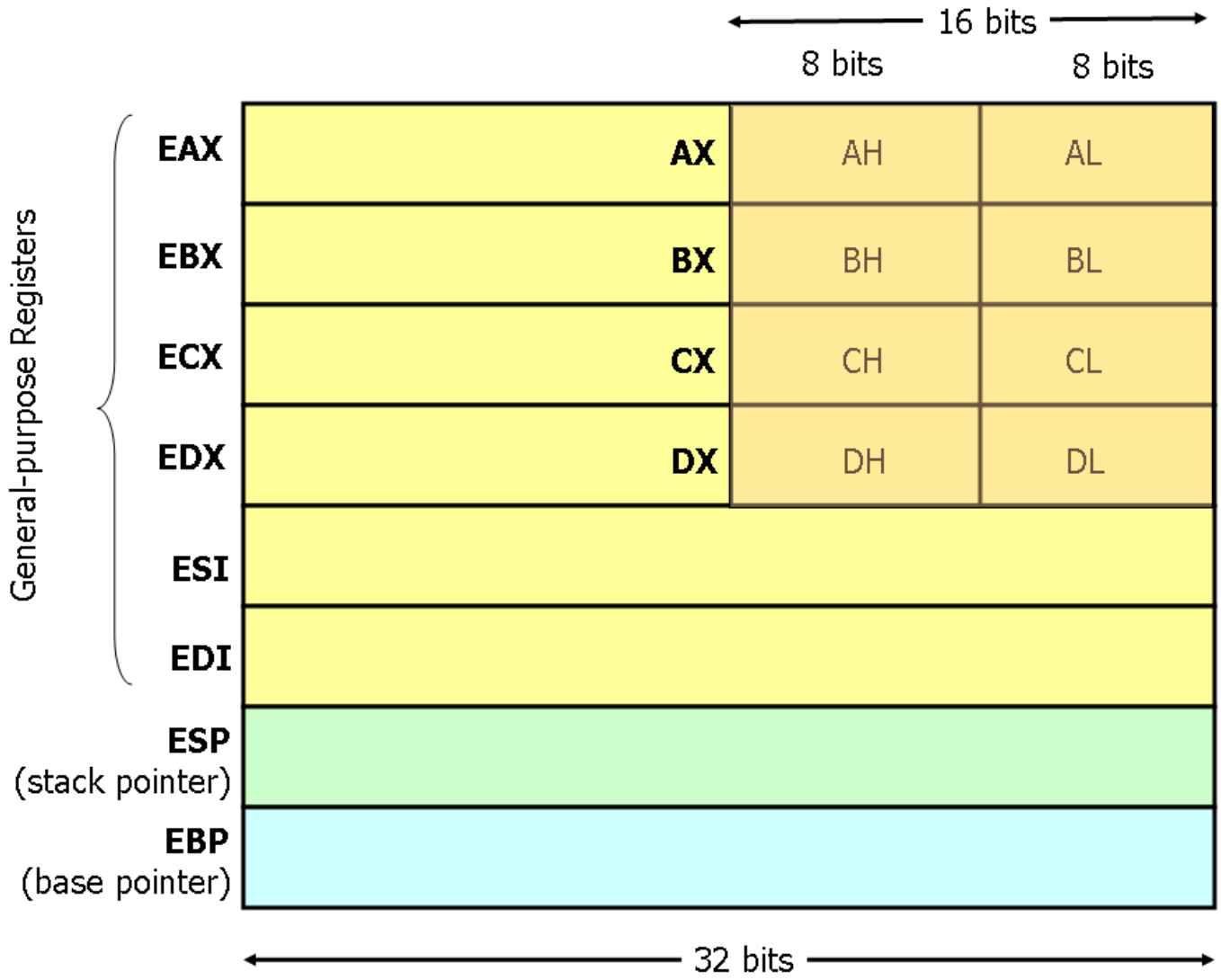
Sono la memorie più veloci di tutto il PC, ma sono di dimensioni molto ridotte:

- 32 bit (architettura x86)
- 64 bit (architettura x64)

La CPU: i registri

...e sono anche molto pochi!
A seconda dell'architettura un processore può avere in genere **16** o **32** registri general purpose (ovvero usati per eseguire dei calcoli) più qualche decina di registri specifici (che non vedremo).

La CPU: i registri



La CPU: comportamento

ALU

Svolge un numero limitato di operazioni logiche

ALU: comportamento

Operazione **AND**

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1

ALU: comportamento

Esempio:

1 0 1 0 0 1 0 1 **AND**
0 1 0 1 1 1 0 1 =
---
0 0 0 0 0 1 0 1

ALU: comportamento

Operazione **OR**

0 OR 0 = 0
0 OR 1 = 1
1 OR 0 = 1
1 OR 1 = 1

ALU: comportamento

Esempio:

1 0 1 0 0 1 0 1 **OR**
0 1 0 1 1 1 0 1 =
---
1 1 1 1 1 1 0 1

ALU: comportamento

Esempio:

1 0 1 0 0 1 0 1 **OR**
0 1 0 1 1 1 0 1 =
---
1 1 1 1 1 1 0 1

È un BYTE!

La CPU

Facciamo un passo indietro...

Cosa sappiamo finora:

- La CPU nelle macchine moderne è composta da CU, ALU e registri
- La ALU ha il compito di fare dei calcoli (operazioni elementari tra bit come AND e OR)
- La CU dice alla ALU quali operazioni svolgere

La CPU: comportamento

Control Unit

Coordina tutte le azioni necessarie ad eseguire delle istruzioni

La CPU: comportamento

Control Unit

Coordina tutte le azioni necessarie ad eseguire delle istruzioni

Che tipo di istruzioni?

I programmi

Un programma è una **sequenza di istruzioni**

I programmi

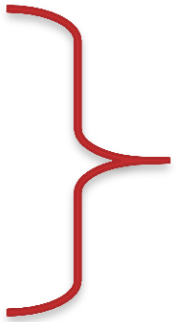
Esempio: programma SOMMA

1. Prendi A
2. Prendi B
3. Somma A e B

I programmi

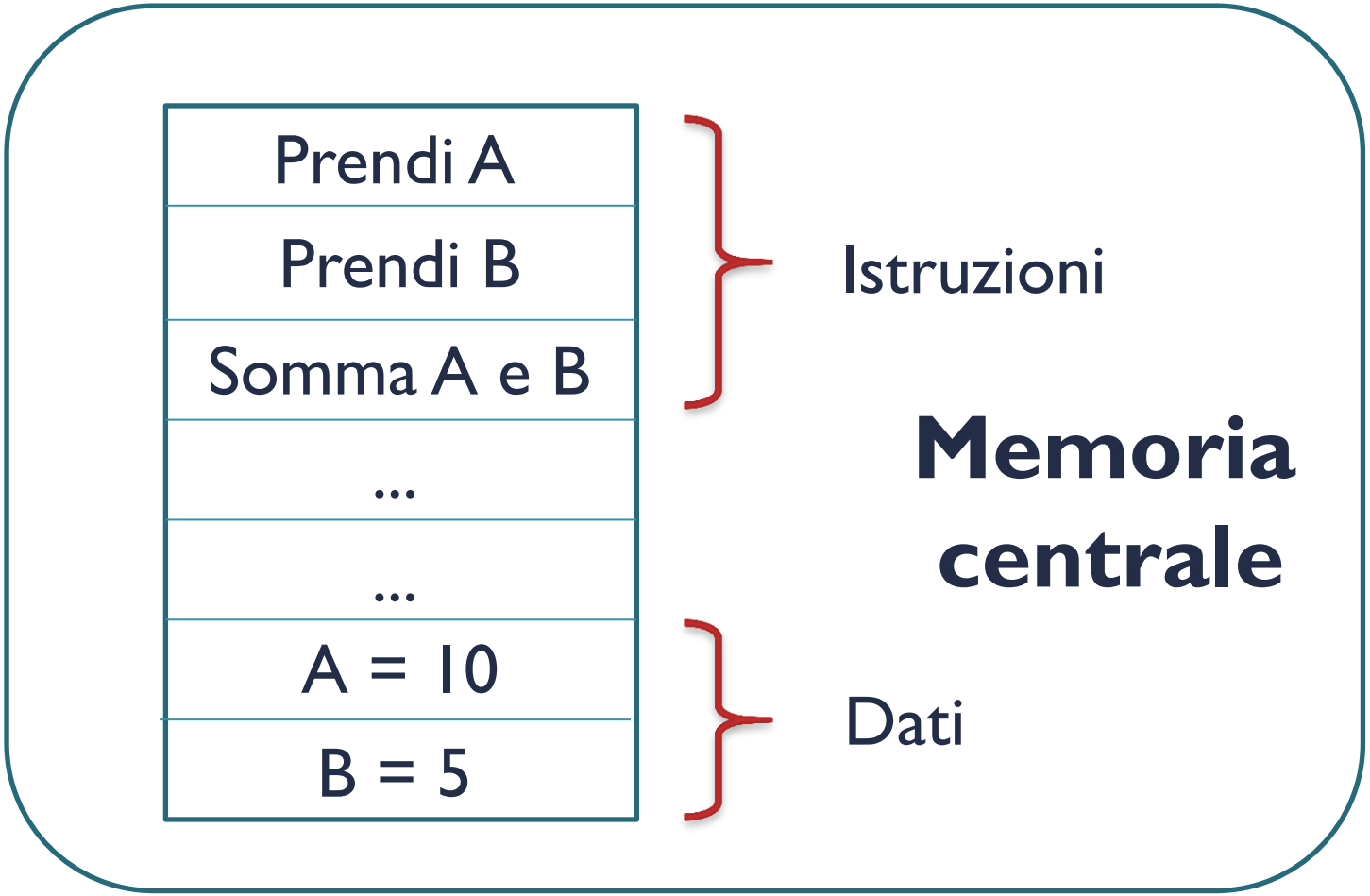
Esempio: programma SOMMA

1. Prendi A
2. Prendi B
3. Somma A e B

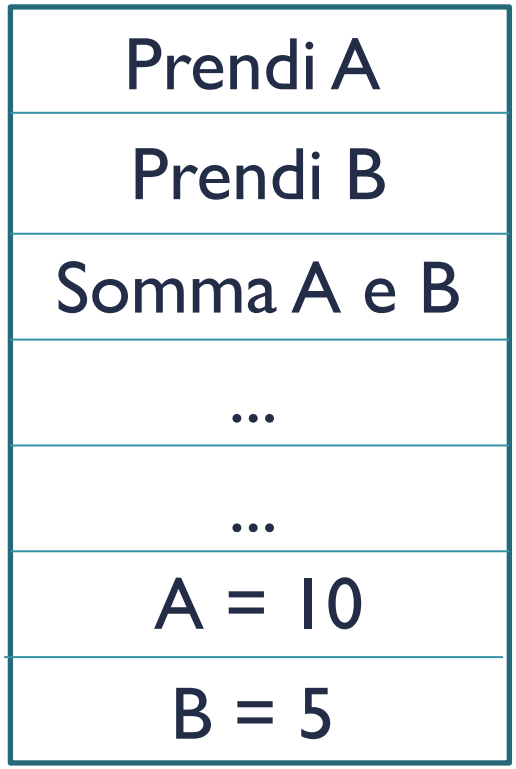


Sono 3 istruzioni

I programmi


Prendi A
Prendi B
Somma A e B
...
...
A = 10
B = 5
Istruzioni
Dati
Memoria centrale

La CPU: comportamento



Il nostro programma appare così in memoria centrale. Il compito della CU è quello di andare a prendere ognuna di queste istruzioni, interpretarle ed eseguirle.

Cosa fa la CU

1. Va a prendere le istruzioni
2. Le interpreta
3. Le esegue

Cosa fa la CU

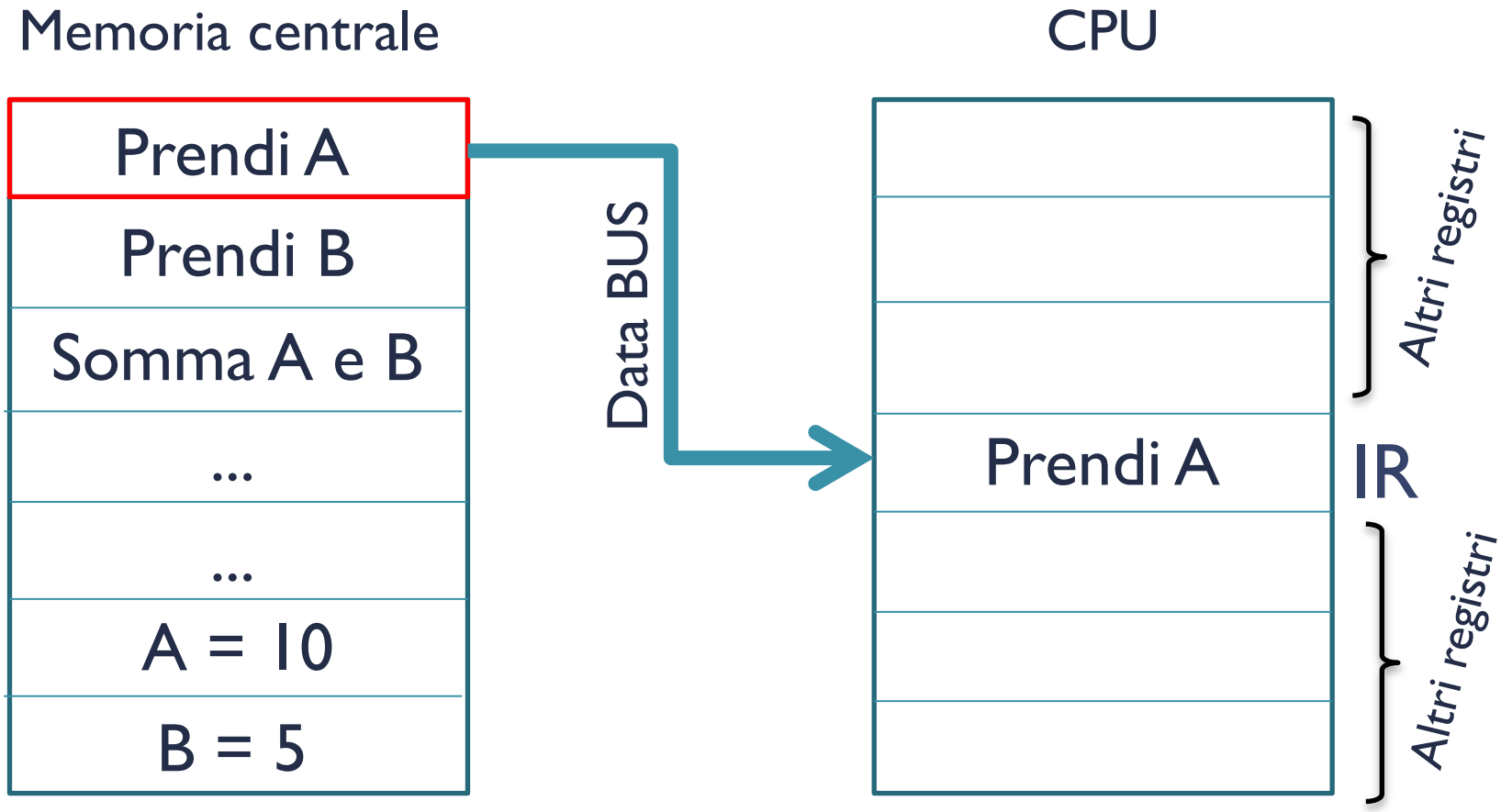
1. Va a prendere le istruzioni **FETCH**
2. Le interpreta **DECODE**
3. Le esegue **EXECUTE**

Il ciclo fetch-decode-execute

Fetch

È l'operazione con cui la CU va a prelevare un'istruzione della memoria centrale e la sposta all'interno della CPU, dentro un registro specifico che si chiama instruction register (IR)

Il ciclo fetch-decode-execute



Il ciclo fetch-decode-execute

Decode

È l'operazione con cui la CU interpreta l'istruzione, ovvero capisce cosa deve fare e si prepara ad agire di conseguenza.

Il ciclo fetch-decode-execute

Decode

Esempio:
«Prendi A» = leggi il valore contenuto in A e copialo nel registro R1

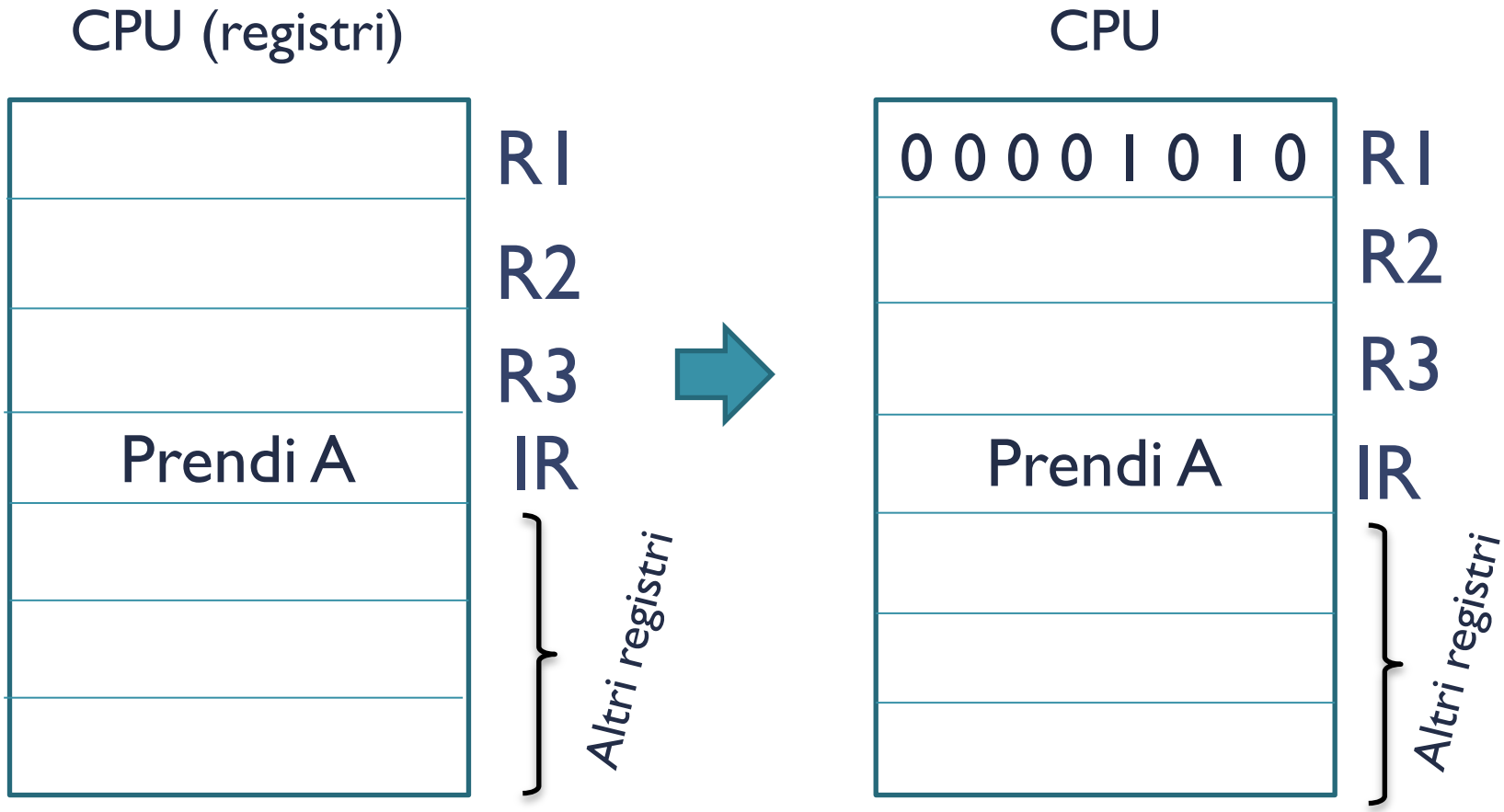
Nota:
A = 10 = 0 0 0 0 10 1 0

Il ciclo fetch-decode-execute

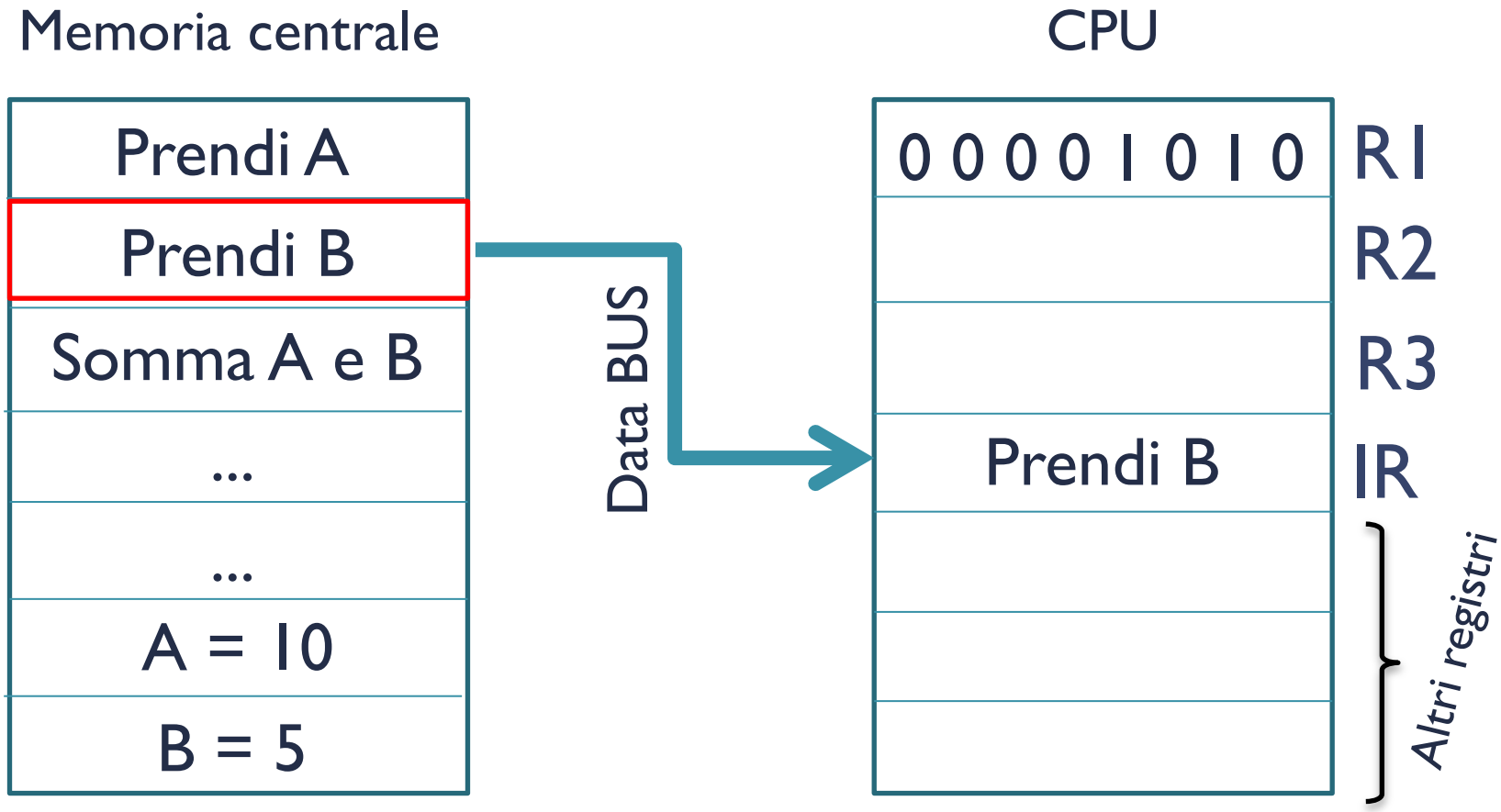
Execute

La CU esegue l'istruzione chiamando in causa ALU e registri se necessario.

Il ciclo fetch-decode-execute



Fetch...



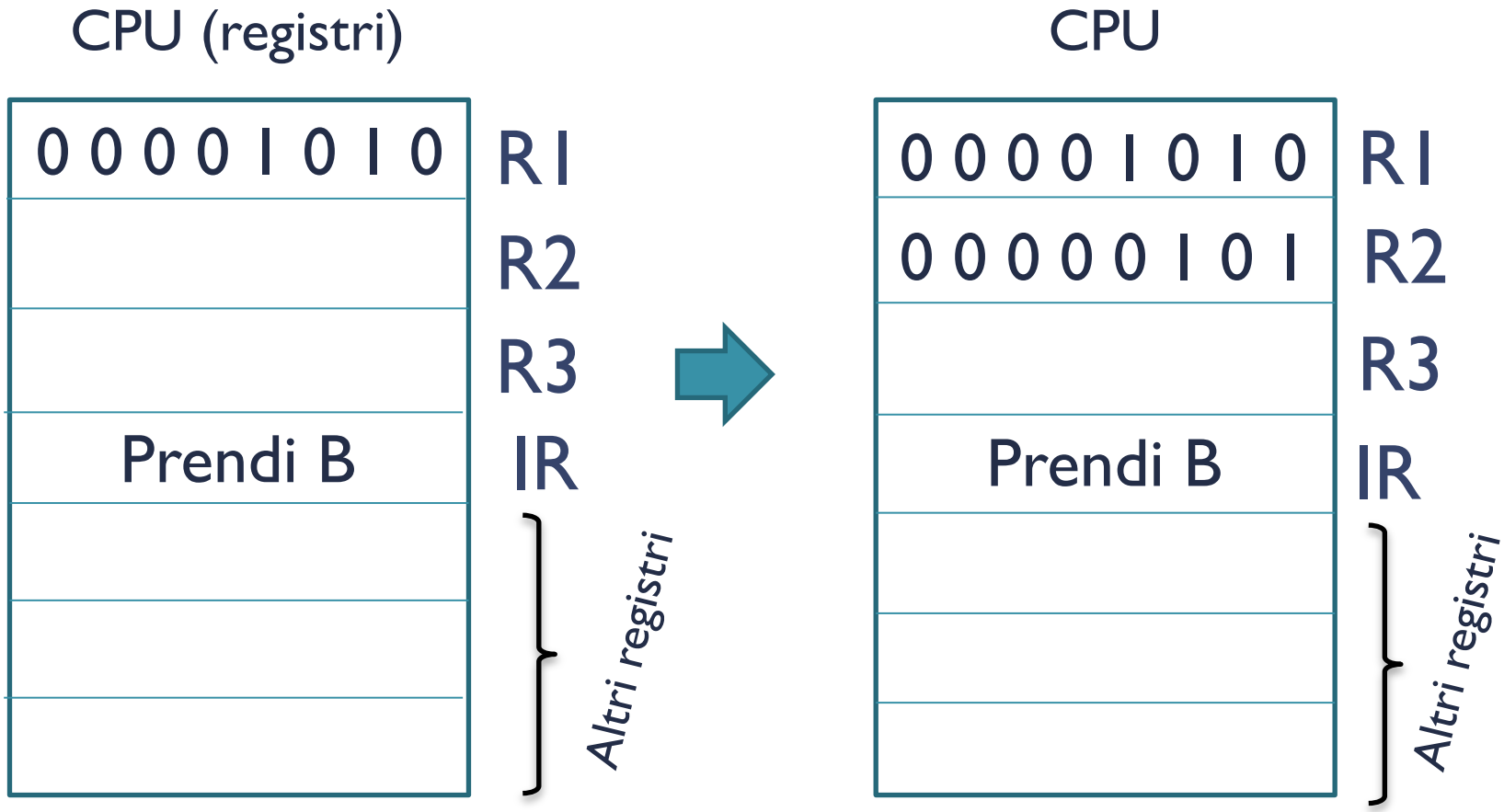
Il ciclo continua...

Decode...

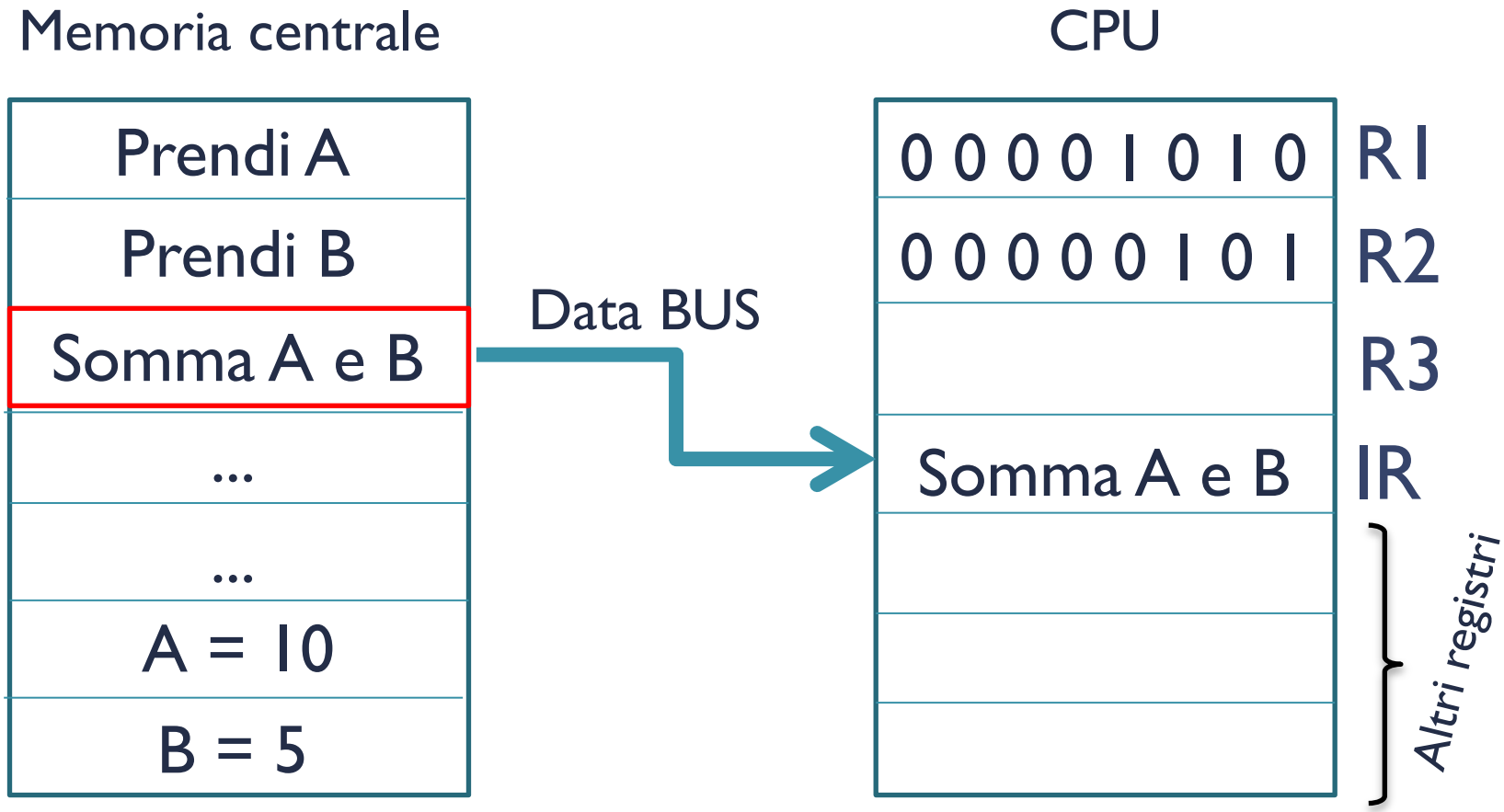
«Prendi B» = leggi il valore contenuto in B e copialo nel registro R2

Nota:
B = 5 = 0 0 0 0 0 1 0 1

Execute...


CPU (registri)
0 0 0 0 1 0 1 0
Prendi B
R1
R2
R3
IR
Altri registri
CPU
0 0 0 0 1 0 1 0
0 0 0 0 0 1 0 1
Prendi B
R1
R2
R3
IR
Altri registri

Fetch...



Il ciclo continua...

Decode...

«Somma A e B» = Somma il contenuto dei registri R1 e R2 e scrivi il risultato in R3.

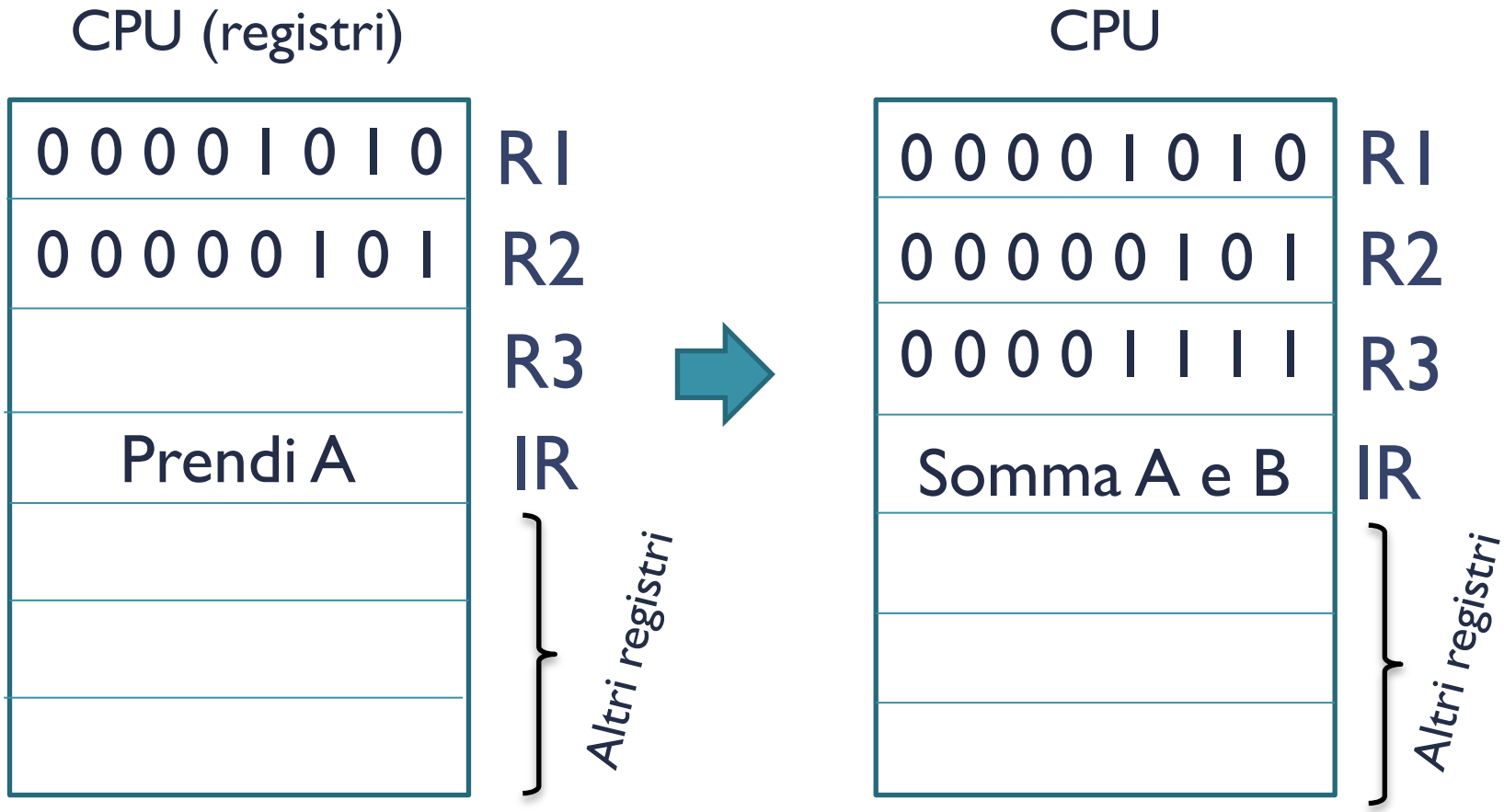
Nota:

0 0 0 0 1 0 1 0 +

0 0 0 0 0 1 0 1 =

???

Execute...


CPU (registri)
0 0 0 0 1 0 1 0
0 0 0 0 0 1 0 1
Prendi A
R1
R2
R3
IR
Altri registri
CPU
0 0 0 0 1 0 1 0
0 0 0 0 0 1 0 1
0 0 0 0 1 1 1 1
Somma A e B
R1
R2
R3
IR
Altri registri

Un momento...

A cosa serve la fase di decode?

L'nstruction set

Il processore parla un linguaggio complesso, fatto di valori binari che vengono scambiati dai registri.
Per rendere la programmazione più agevole ogni processore è dotato di un **instruction set** (insieme di istruzioni).

L'instruction set

L'instruction set è l'insieme di operazioni che può svolgere la CPU. Viene espresso in un linguaggio semplificato in modo che i programmatori possano impartire comandi al processore senza dover scrivere manualmente complicate operazioni tra i registri.

L'instruction set

Ad esempio l'istruzione ADD(R1, R2) permette di sommare il contenuto dei due registri senza specificare quali operazioni logiche deve eseguire la ALU per produrre il risultato.

Il linguaggio Assembly

Le istruzioni presenti nell'instruction set permettono di scrivere software usando un linguaggio di programmazione chiamato ASSEMBLY.

Il linguaggio Assembly

Assembly è un linguaggio di basso livello, perché permette di usare solo istruzioni molto semplici.
A dire il vero è il linguaggio di livello più basso che ci sia, ovvero il più vicino al linguaggio macchina usato dal processore.

RISC vs CISC

Il tipo di instruction set di un processore permette di distinguere due diverse architetture.

RISC

RISC (Reduced Instruction Set Computer): l'instruction set è ridotto e composto solo da istruzioni semplici. Predilige la velocità.

CISC

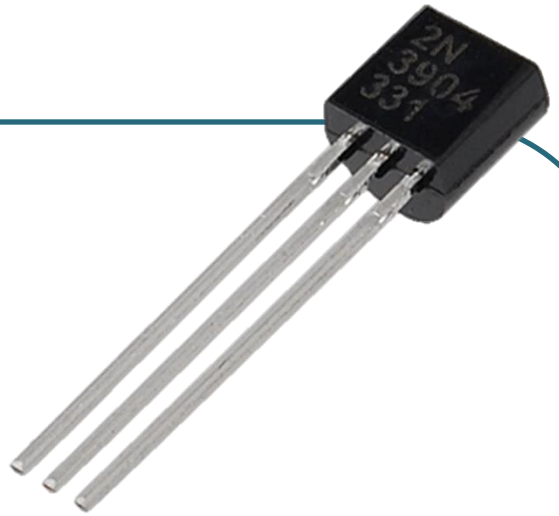
CISC (Complex Instruction Set Computer): l'instruction set è più ampio e contiene istruzioni complesse. Tra le due è l'architettura più complicata da realizzare, ma fornisce ai programmatori potenzialità maggiori.

Struttura fisica del processore

Zero e uno

All'interno del computer, a livello fisico, una corrente positiva (ad esempio da 1 volt) viene letta come 1, mentre una corrente neutra (da 0 volt) viene letta come 0.

Transistor



Un transistor è un dispositivo elettrico che funziona come un interruttore: può lasciare passare la corrente oppure bloccarla.

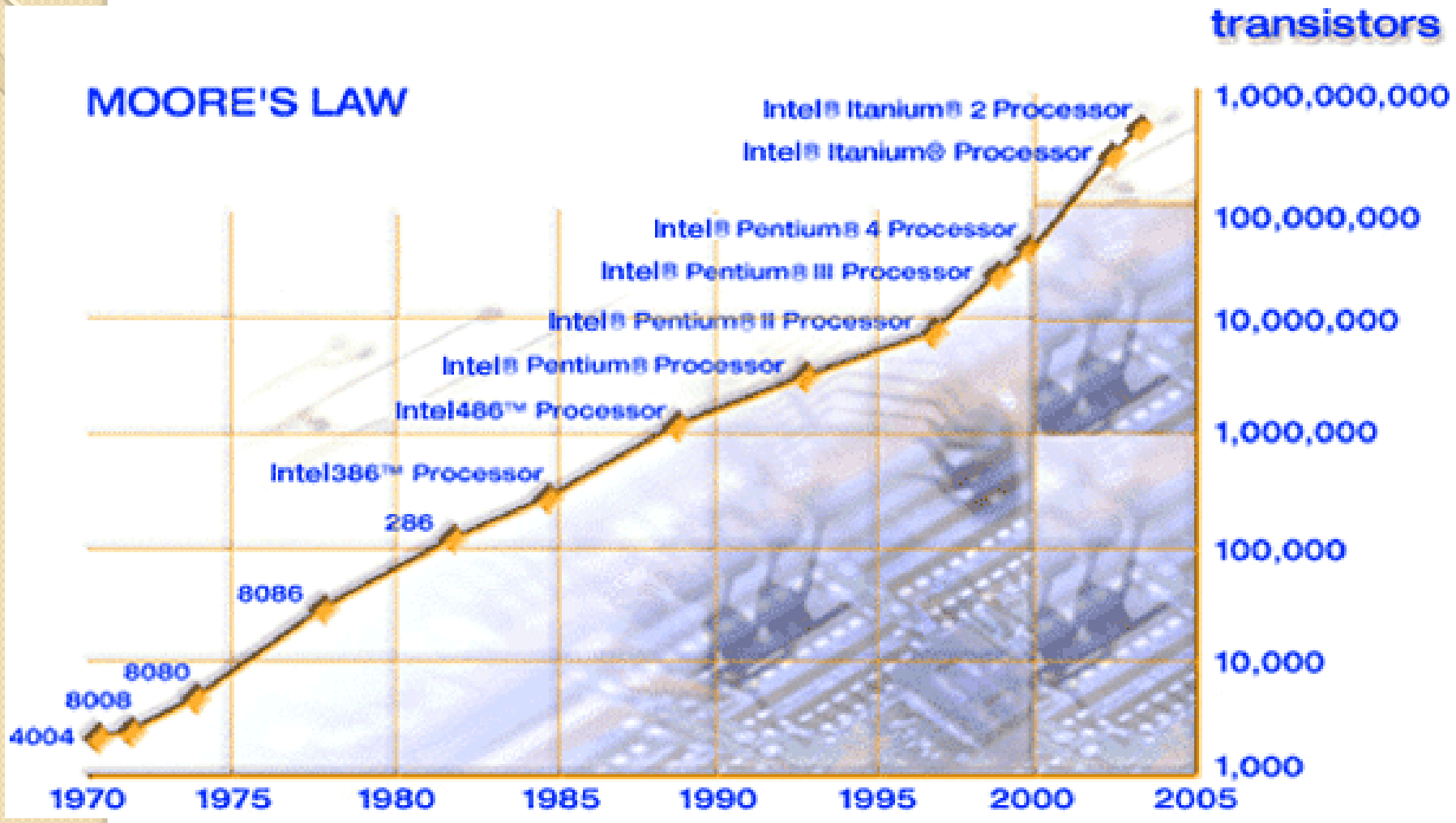
Transistor

Utilizzando i transistor è possibile creare dei circuiti elettrici che eseguano le operazioni logiche che abbiamo visto poco fa (AND e OR, ma non solo).

Transistor

Nel corso degli anni la ricerca scientifica è riuscita a miniaturizzare sempre di più i transistor, permettendo di creare circuiti sempre più piccoli.

Transistor


MOORE'S LAW
transistors
Intel® Itanium® 2 Processor
Intel® Itanium® Processor
Intel® Pentium® 4 Processor
Intel® Pentium® III Processor
Intel® Pentium® II Processor
Intel® Pentium® Processor
Intel486™ Processor
Intel386™ Processor
286
8086
8080
8008
4004
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
1970
1975
1980
1985
1990
1995
2000
2005

La legge di Moore

« La complessità di un microcircuito, misurata ad esempio tramite il numero di transistori per chip, raddoppia ogni 18 mesi (e quadruplica quindi ogni 3 anni). »

La legge di Moore

Enunciata nel 1965 dal chimico Gordon Moore, questa legge si è rivelata nel tempo sorprendentemente accurata. L'evoluzione di tutti i settori tecnologici ha seguito questo andamento negli ultimi cinquant'anni.

La legge di Moore

Se negli anni 70 un processore poteva contare poco più di 1000 transistor, oltre 10 anni fa è stato superato il miliardo. Nel 2018 Intel lancerà sul mercato la prima architettura a 10 nano-metri, che significa 100 milioni di transistor per millimetro quadrato.

Il Clock

Un processore è un circuito elettrico miniaturizzato composto da miliardi di transistor. Per funzionare, quindi, è necessario che una corrente elettrica passi all'interno del circuito.

Il Clock

La velocità di un processore normalmente si misura in GigaHertz (GHz), e viene anche chiamata frequenza di clock.

Questo valore esprime il numero di operazioni che la CPU può eseguire in un secondo.

Il Clock

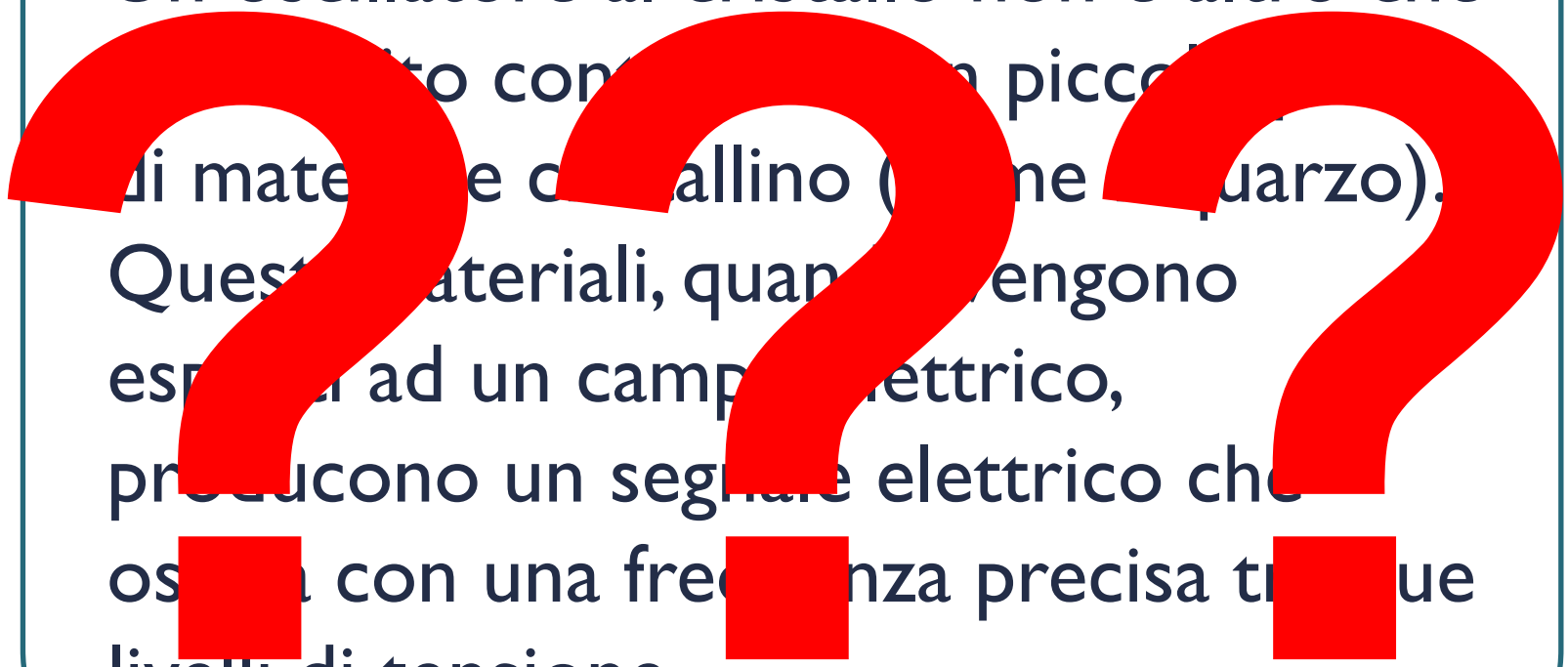
La sincronizzazione del lavoro di tutti i transistor all'interno della CPU avviene infatti per mezzo di un oscillatore al quarzo, come quelli usati all'interno degli orologi.

Il Clock

Un oscillatore al cristallo non è altro che un circuito contenente un piccolo pezzo di materiale cristallino (come il quarzo). Questi materiali, quando vengono esposti ad un campo elettrico, producono un segnale elettrico che oscilla con una frequenza precisa tra due livelli di tensione.

Il Clock

Un oscillatore al cristallo non è altro che
to con picco
di mate e c allino (me quarzo).
Ques ateriali, quan engono
esp ad un camp ettrico,
pr ucono un segn elettrico che
os a con una fre nza precisa t ue
livelli di tensione.



Il Clock

In pratica, se un processore ha una frequenza di 3.5 GHz, significa che 3,5 miliardi di volte al secondo il suo stato interno può cambiare (ovvero i transistor possono passare dal valore 0 al valore 1 e viceversa).

Possiamo dire che il processore esegue 3,5 miliardi di cicli di clock al secondo.

Il Clock

I cicli di clock scandiscono il tempo all'interno del PC, facendo in modo che tutte le operazioni siano sincronizzate. La durata di un'operazione eseguita dal processore non si misura in secondi ma in cicli di clock.