Programmi e la macchina di Von Neumann

Informatica A, AA 2023/2024

Giacomo Boracchi

giacomo.boracchi@polimi.it

19 Settembre 2023

I Programmi

Dall'algoritmo al codice

Cos'è l'Informatica?

Scienza della rappresentazione e dell'elaborazione dell'informazione.

Scienza: ovvero una **conoscenza sistematica** e **rigorosa** di tecniche e metodi.

Informazione: l'oggetto dell'investigazione scientifica (informazione intesa come entità astratta e come tecnologie per la sua gestione)

Rappresentazione: il modo in cui l'informazione viene strutturata e trasformata in dati fruibili da macchine

Elaborazione: uso e trasformazione dell'informazione per un dato scopo. L'elaborazione deve poter essere eseguita da **macchine** che processano dati.

[da «Informatica Arte e Mestiere»]

Sistema Informatico

Sistemi informatico: L'esecutore dei programmi

PC, laptop, telefoni, smartphones, server con migliaia di utenti etc… tutti sistemi informatici

Sono oggetti complessi, costruiti da diverse parti che interagiscono tra loro.

I sistemi informatici sono composti da

- **Hardware**: i componenti fisici del sistema
- **Software**: i programmi eseguiti dal sistema (es. web browser, mail, suite office, sistema operativo, compilatori, IDE… i programmi che scriveremo)

Consideriamo inizialmente l'hardware dei sistemi informatici.

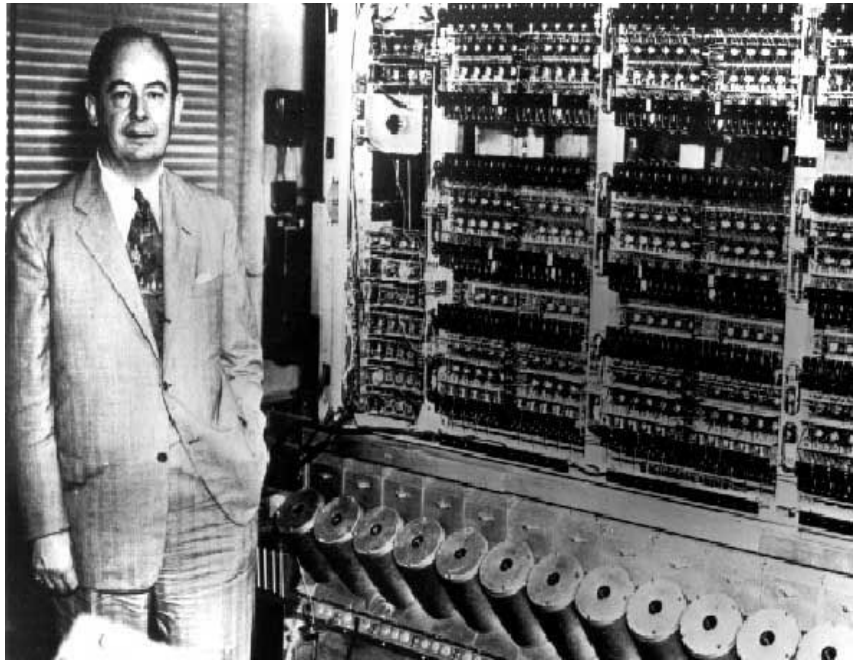
I Modelli dei Calcolatori

Macchina di Turing (1936):

- Modello teorico legato alla logica matematica

Macchina di von Neumann (1943→46)

- Calcolatore reale (ENIAC) dotato di capacità di scelta, di funzionalità programmata e digitale (ma non binario bensì decimale)
- Calcola esattamente le stesse funzioni della MdT
  - (purché dotato di memoria sufficientemente grande)



La Macchina di Von Neumann

Un modello dell'architettura dei calcolatori

La Macchina di Von Neumann

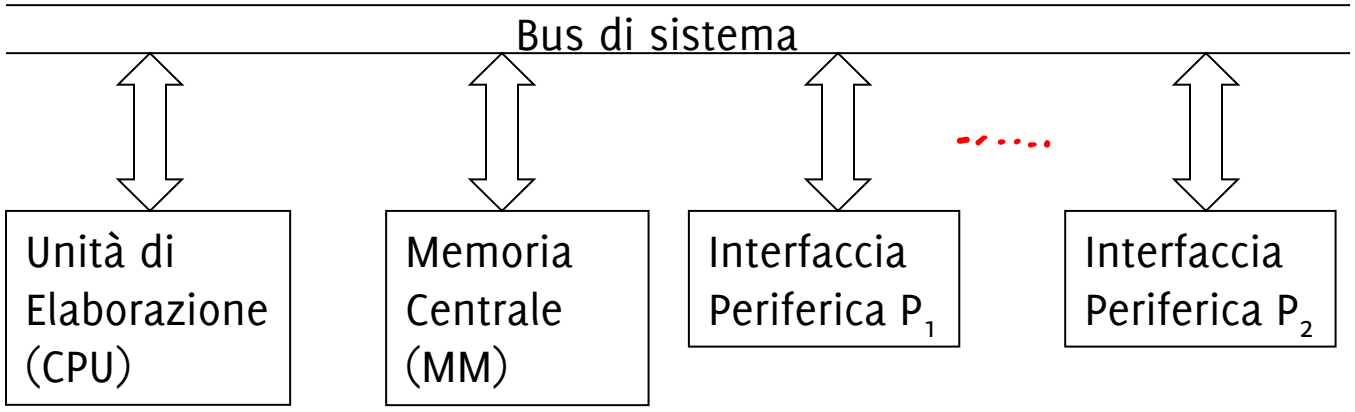
Modello composto da **quattro elementi funzionali**

- **Unità di elaborazione (CPU):** interpretazione ed esecuzione dei programmi, coordinamento macchina
- **Memoria centrale:** contiene dati ed istruzioni
- **Interfacce delle periferiche:** scambio informazioni con mondo esterno (e.g, stampante, tastiera, mouse, rete, schermo, HDD ..)
- **Bus di sistema:** collega gli altri elementi funzionali

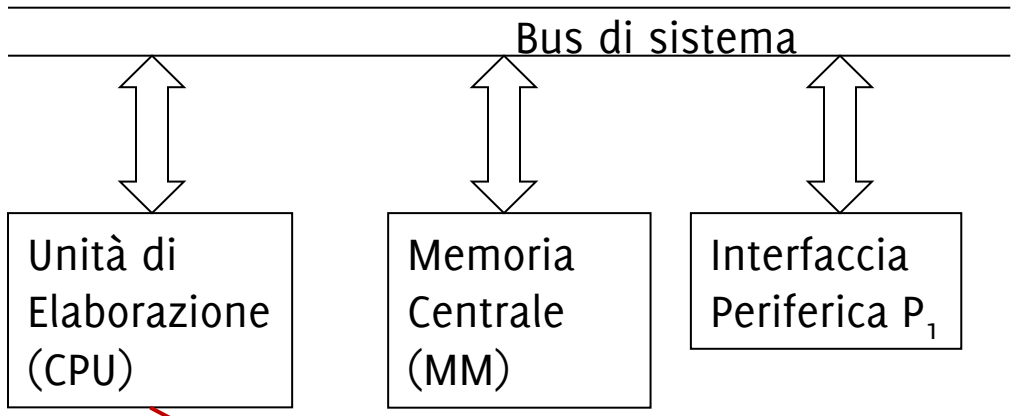
La Macchina di Von Neumann

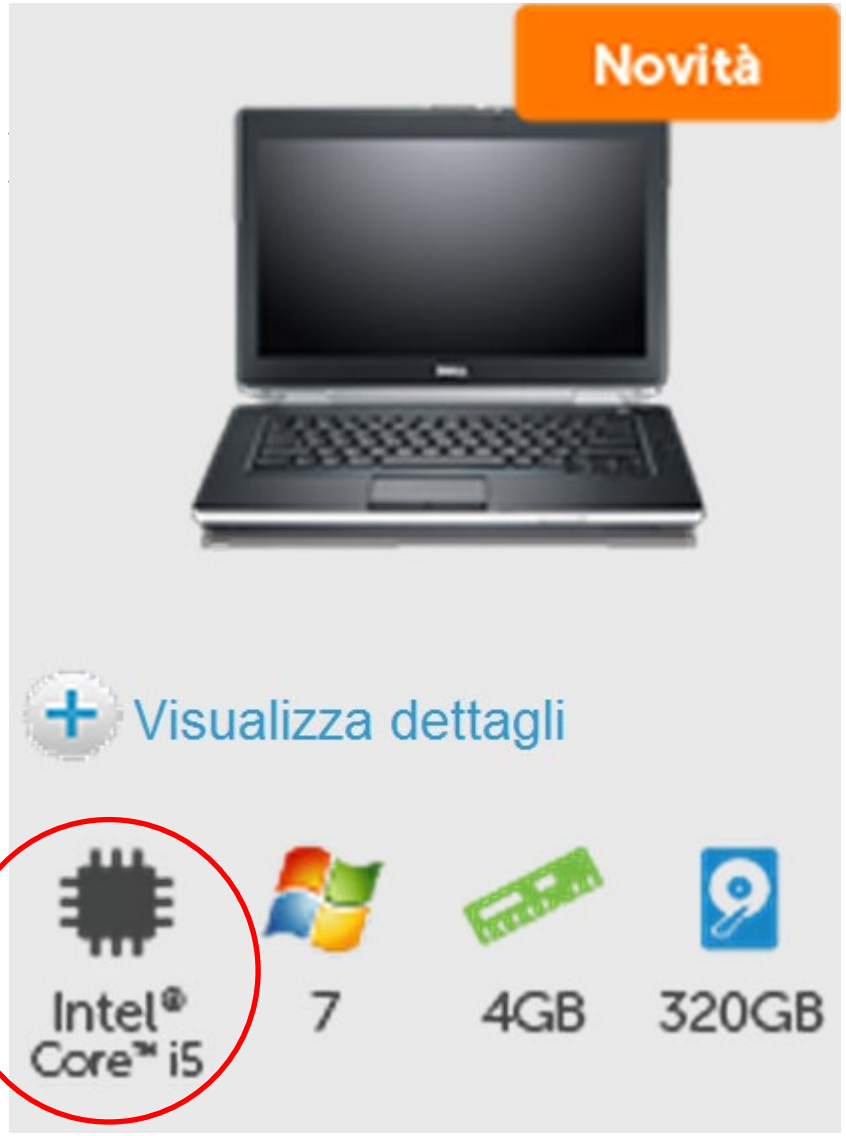
Modello composto da **quattro elementi funzionali**

- **Unità di elaborazione (CPU):** interpretazione ed esecuzione dei programmi, coordinamento macchina
- **Memoria centrale:** contiene dati ed istruzioni
- **Interfacce delle periferiche:** scambio informazioni con mondo esterno (e.g, stampante, tastiera, mouse, rete, schermo, HDD ..)
- **Bus di sistema:** collega gli altri elementi funzionali

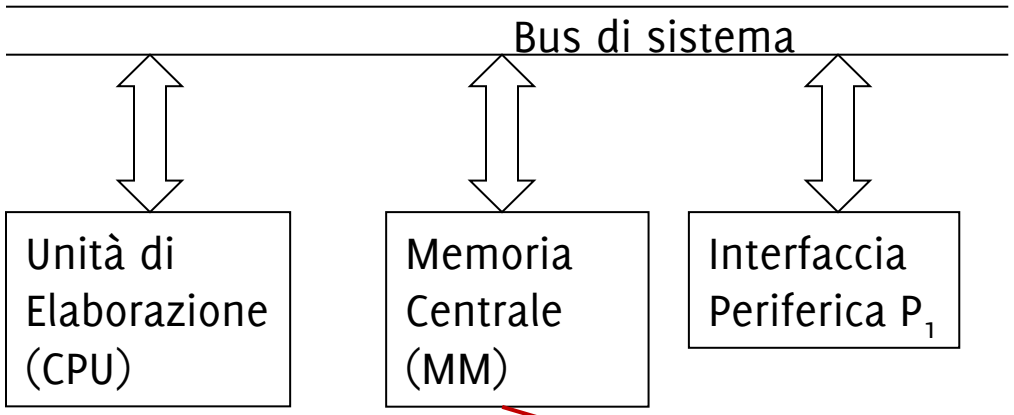


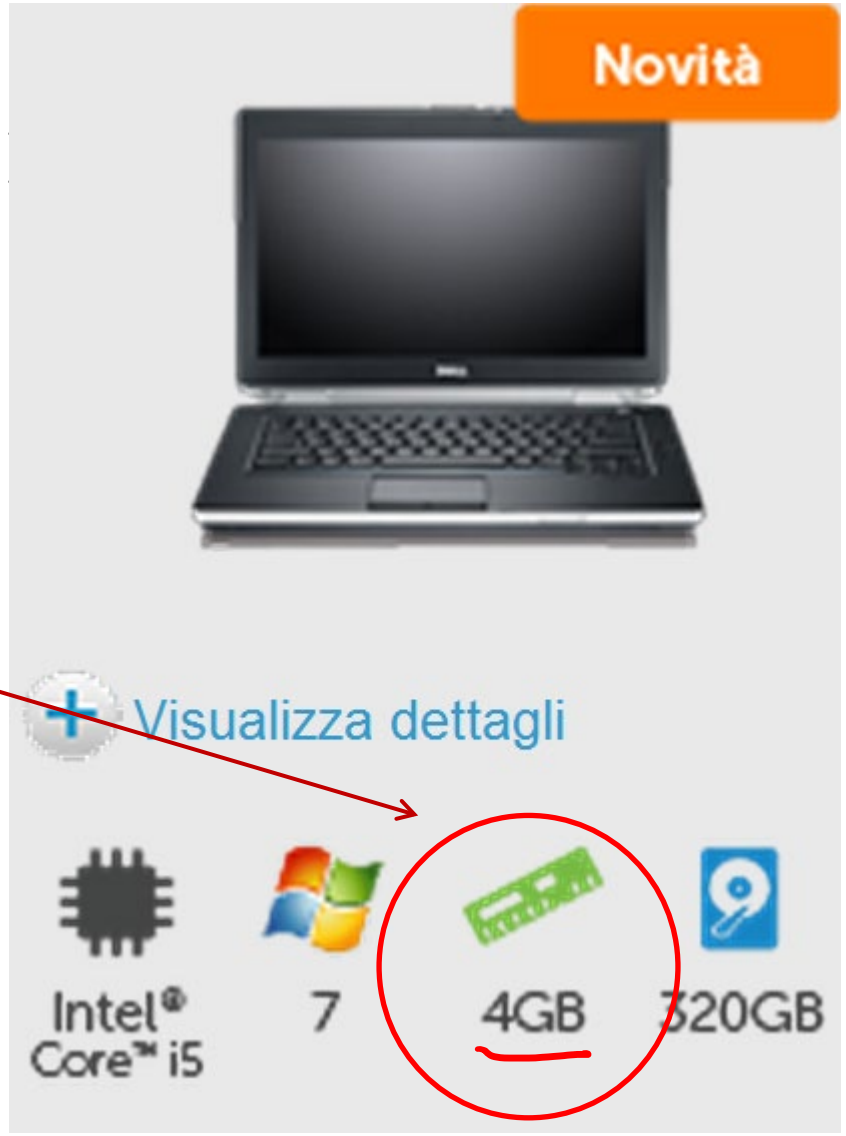
La Macchina di Von Neumann



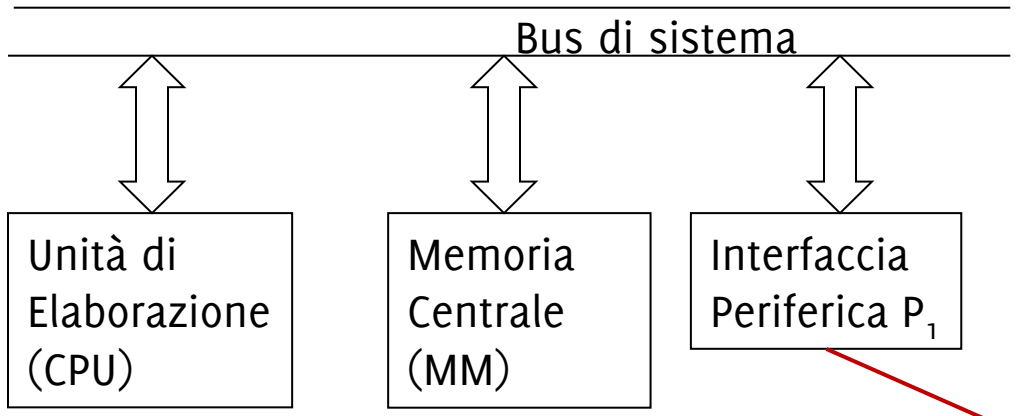


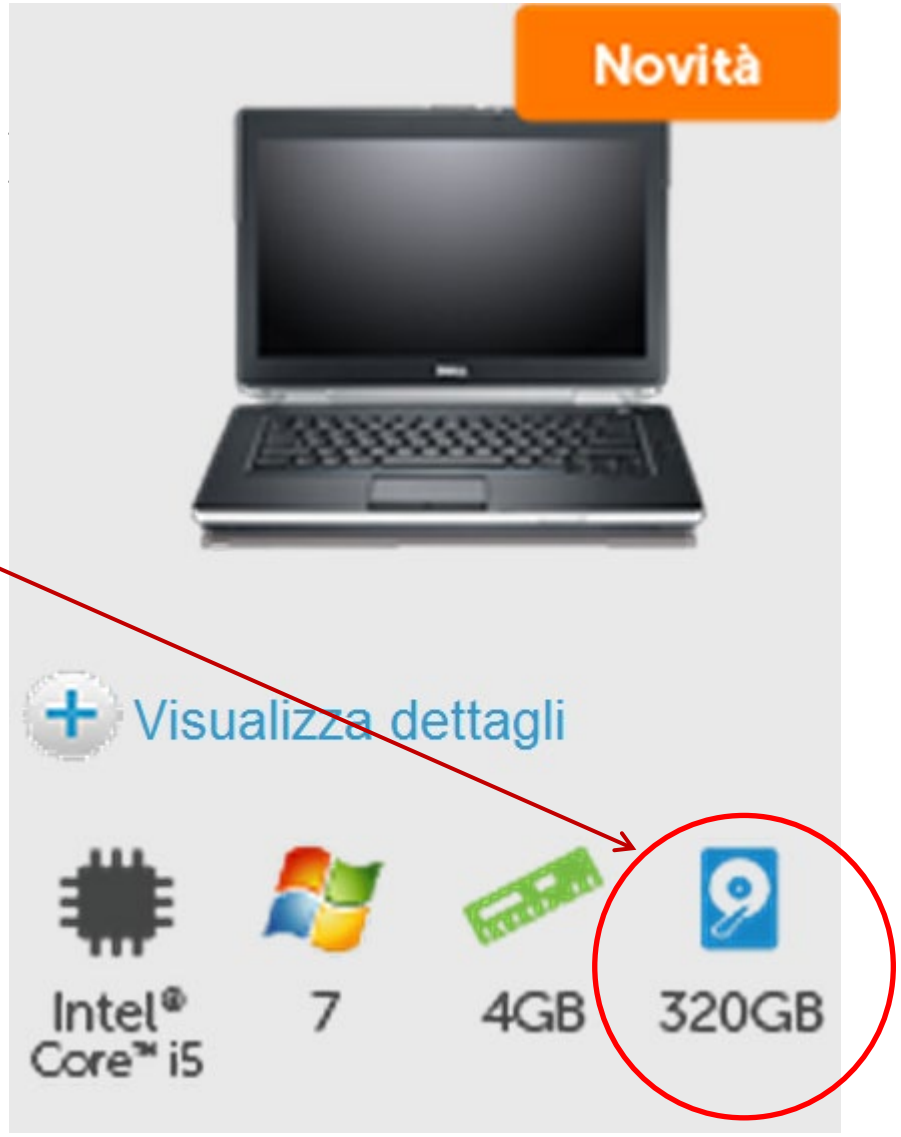
La Macchina di Von Neumann



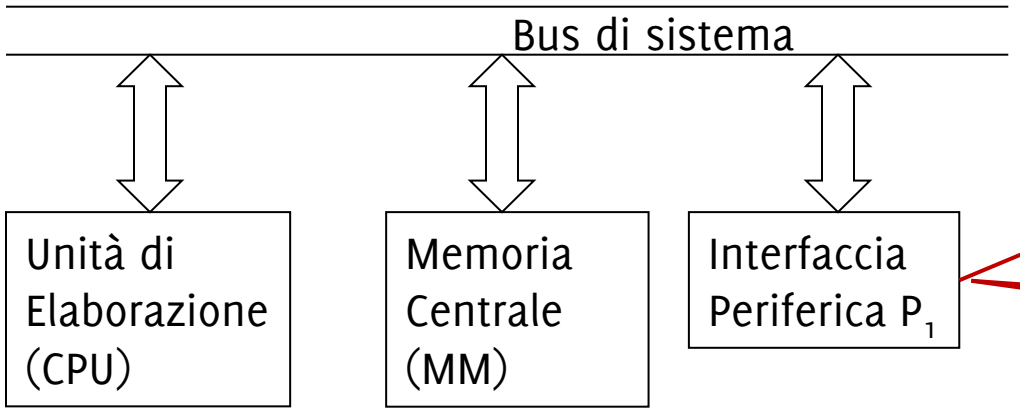


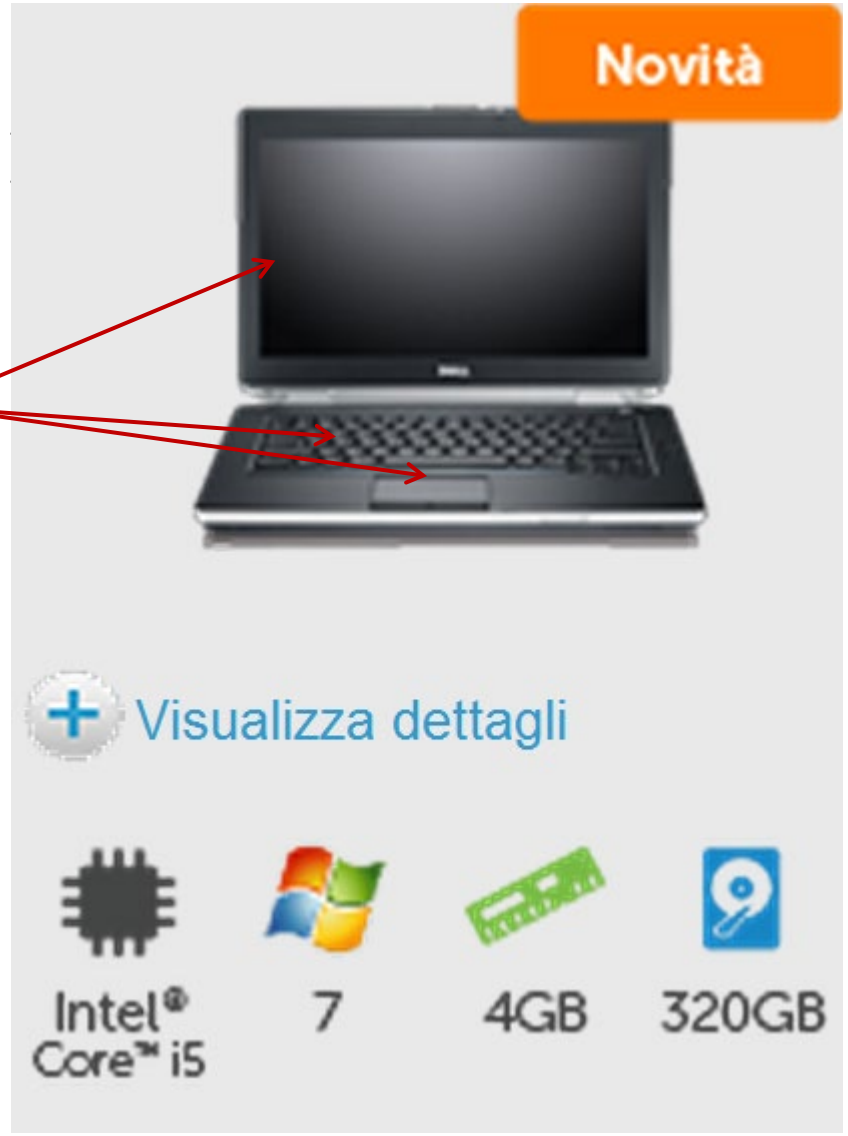
La Macchina di Von Neumann





La Macchina di Von Neumann

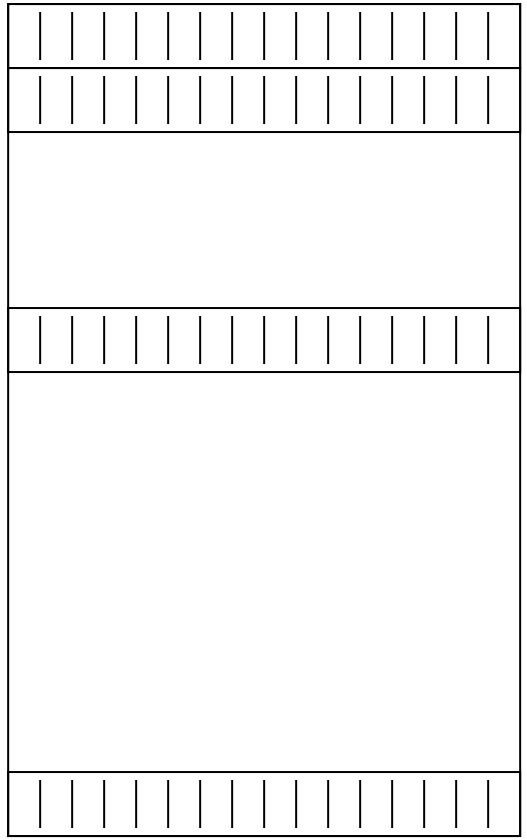




Main Memory

La Memoria Centrale (MM)

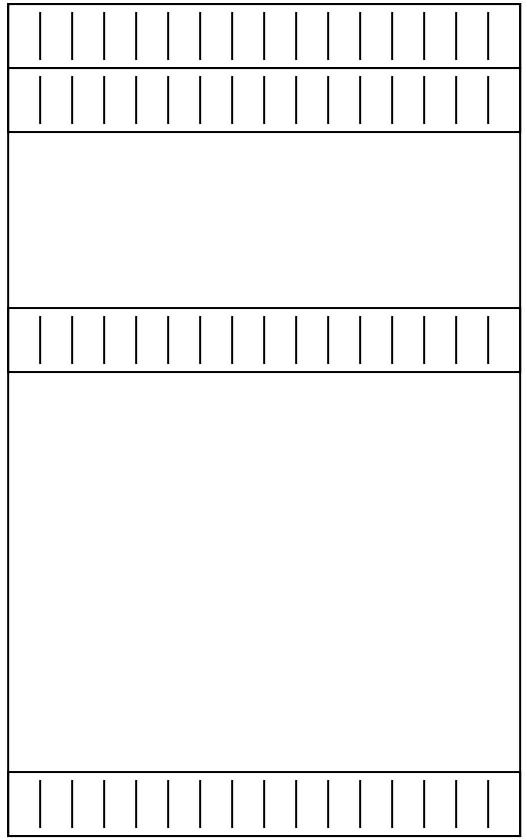
Contiene i **programmi (sequenza di istruzioni)**
in **esecuzione** ed i relativi **dati**



La Memoria Centrale (MM)

Contiene i **programmi (sequenza di istruzioni)** in **esecuzione** ed i relativi **dati**

È schematizzata come una sequenza di celle.

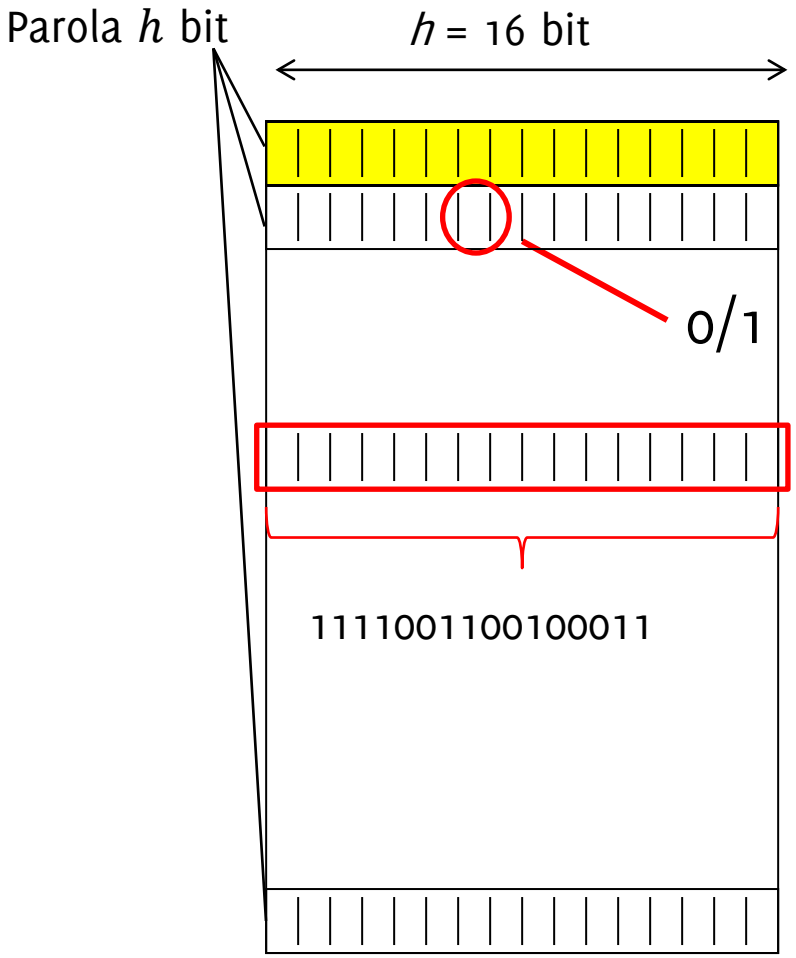


La Memoria Centrale (MM)

Contiene i **programmi (sequenza di istruzioni)** in **esecuzione** ed i relativi **dati**

È schematizzata come una sequenza di celle.

Ogni cella contiene *h* bit, i.e., una Parola (*word*)



La Memoria Centrale (MM)

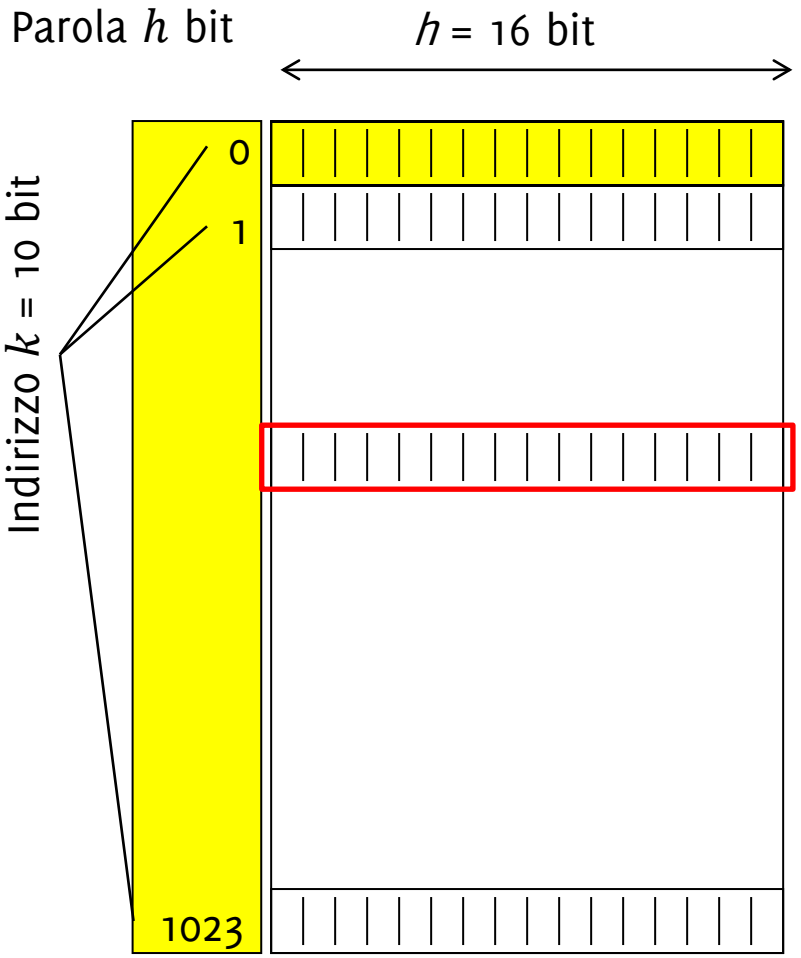
Contiene i **programmi (sequenza di istruzioni)** in **esecuzione** ed i relativi **dati**

È schematizzata come una sequenza di celle.

Ogni cella contiene *h* bit, i.e., una Parola (*word*)

Ogni cella ha un indirizzo e deve essere scritto in binario

Se ho a disposizione *k* bit per scrivere l'indirizzo, lo spazio di indirizzamento è 2^k celle



La Memoria Centrale (MM)

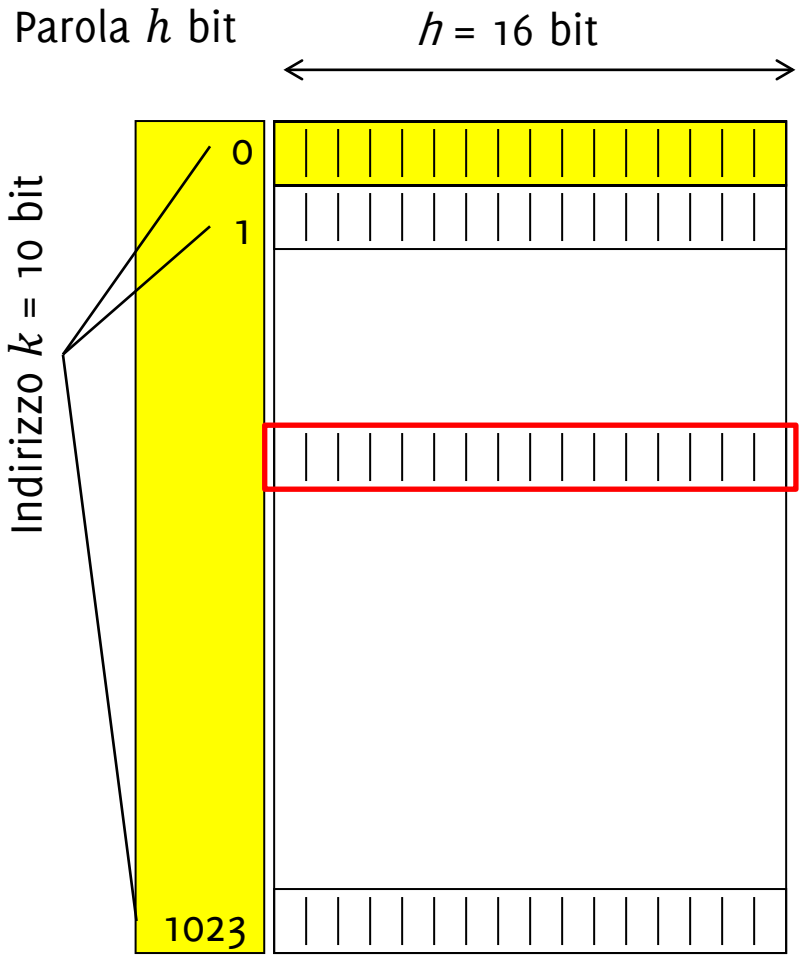
Contiene i **programmi (sequenza di istruzioni)** in **esecuzione** ed i relativi **dati**

È schematizzata come una sequenza di celle.

Ogni cella contiene *h* bit, i.e., una Parola (*word*)

Ogni cella ha un indirizzo e deve essere scritto in binario

Se ho a disposizione *k* bit per scrivere l'indirizzo, lo spazio di indirizzamento è 2^k celle



Oss: ***Avere più celle dello spazio di indirizzamento è come un palazzo (senza scale) dove l'ascensore ha meno pulsanti del numero di piani***
Anonimo studente di Informatica A

La Memoria Centrale (MM)

La MM contiene i **programmi in esecuzione**: ogni **dato e ogni istruzione**, prima di essere elaborato, viene copiato in memoria centrale.

Diversi tipi di Memorie

- RAM (*Random Access Memory)* memoria volatile. Celle indirizzabili in qualunque ordine (accesso random = diretto)
- ROM *(Read Only Memory)* memoria permanente. Il BIOS (Basic I/O System) che carica in memoria il sistema operativo quando la macchina viene accesa
- EPROM *(Erasable Programmable ROM)* riprogrammabile

L'HDD è memoria permanente ma **non è memoria centrale** ed in riferimento alla macchina di Von Neumann è una periferica.

Tecnologie per la memoria centrale

La memoria RAM

- È realizzata mediante circuiti a transistori
- È modificabile (leggibile e scrivibile)
- deve essere continuamente alimentata per mantenere le informazioni (volatile)
- All'accensione contiene una sequenza casuale di 0 e 1

La memoria ROM

- È solo leggibile: le informazioni sono di solito scritte in modo permanente dal costruttore
- È caricata al momento della produzione del calcolatore
- Vi si accede ogni qualvolta questo viene acceso
- Contiene il bootstrap, un programma contenente le prime istruzioni che la CPU deve eseguire

Central Processing Unit

L'Unità di Elaborazione (CPU)

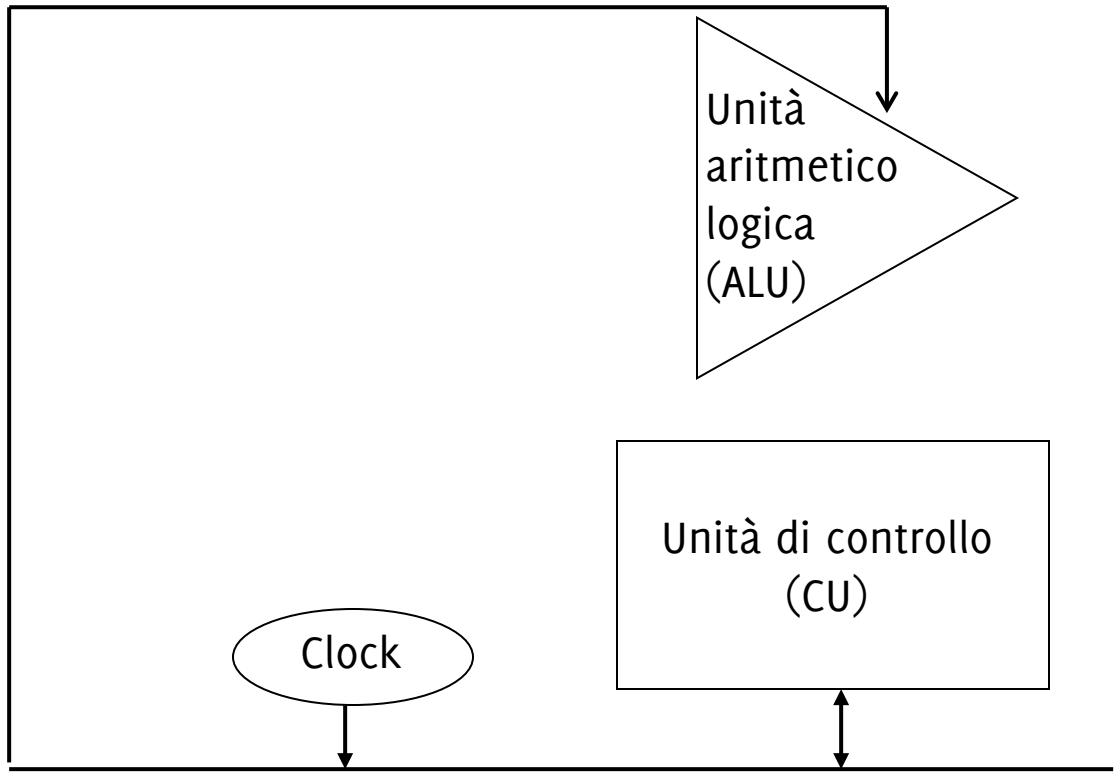
La *Central Processing Unit* (CPU) **coordina** il funzionamento del **calcolatore** ed **esegue** i **programmi**:
estrae, decodifica ed esegue le istruzioni in memoria.

Le istruzioni possono comportare **elaborazione** o **trasferimento** dell'informazione

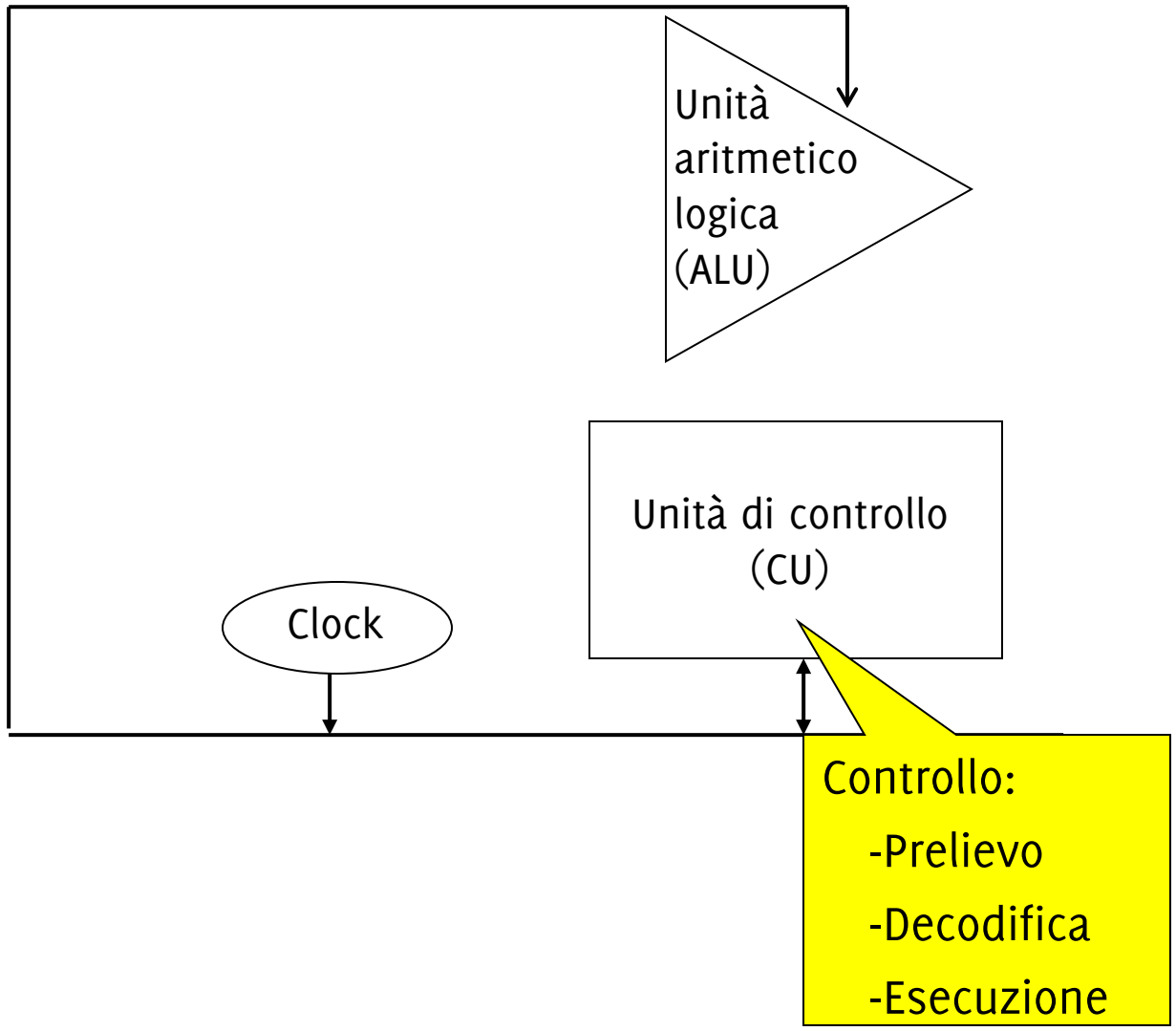
La CPU contiene a sua volta:

- l'**Unità di Controllo** che preleva e decodifica istruzioni dalla MM, invia segnali per eseguire le istruzioni
- Il **Clock di sistema**, opera come un metronomo per la CPU
- L'**Unità Aritmetico Logica:** operazioni aritmetiche e logiche

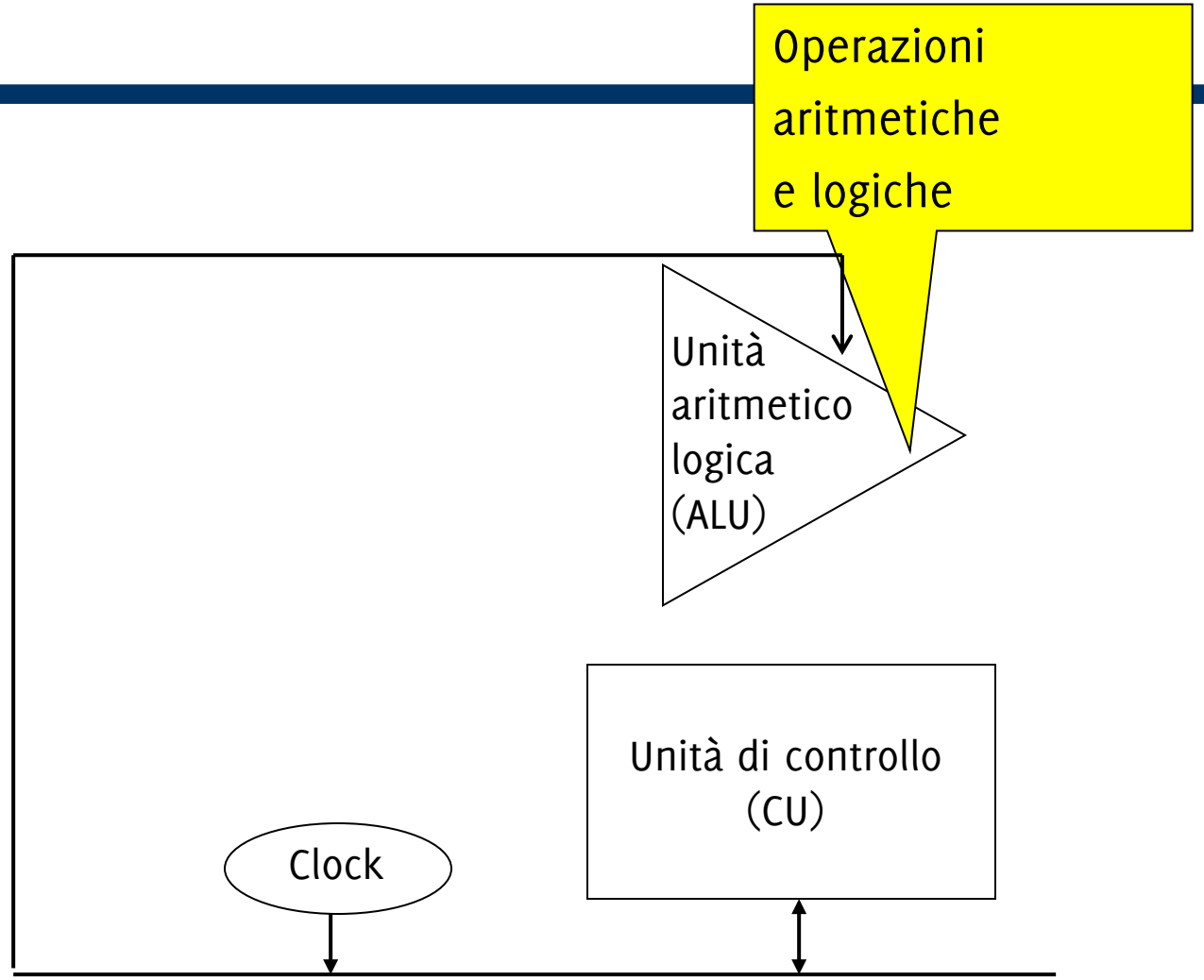
L'Unità di Elaborazione (CPU)



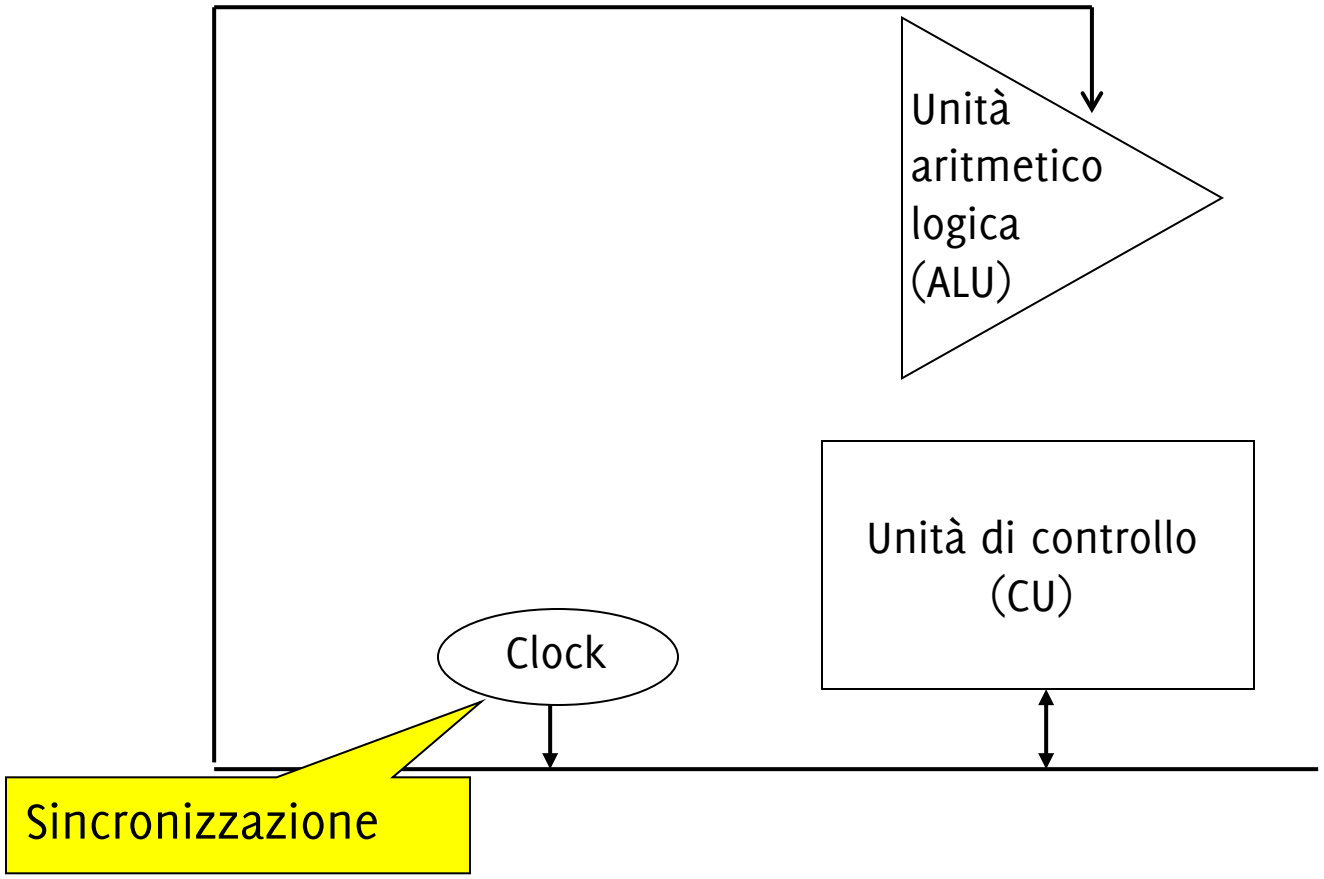
L'Unità di Elaborazione (CPU)



L'Unità di Elaborazione (CPU)



L'Unità di Elaborazione (CPU)



L'Unità di Elaborazione (CPU)

La *Central Processing Unit* (CPU) **coordina** il funzionamento del **calcolatore** ed **esegue** i **programmi**:
estrae, decodifica ed esegue le istruzioni in memoria.

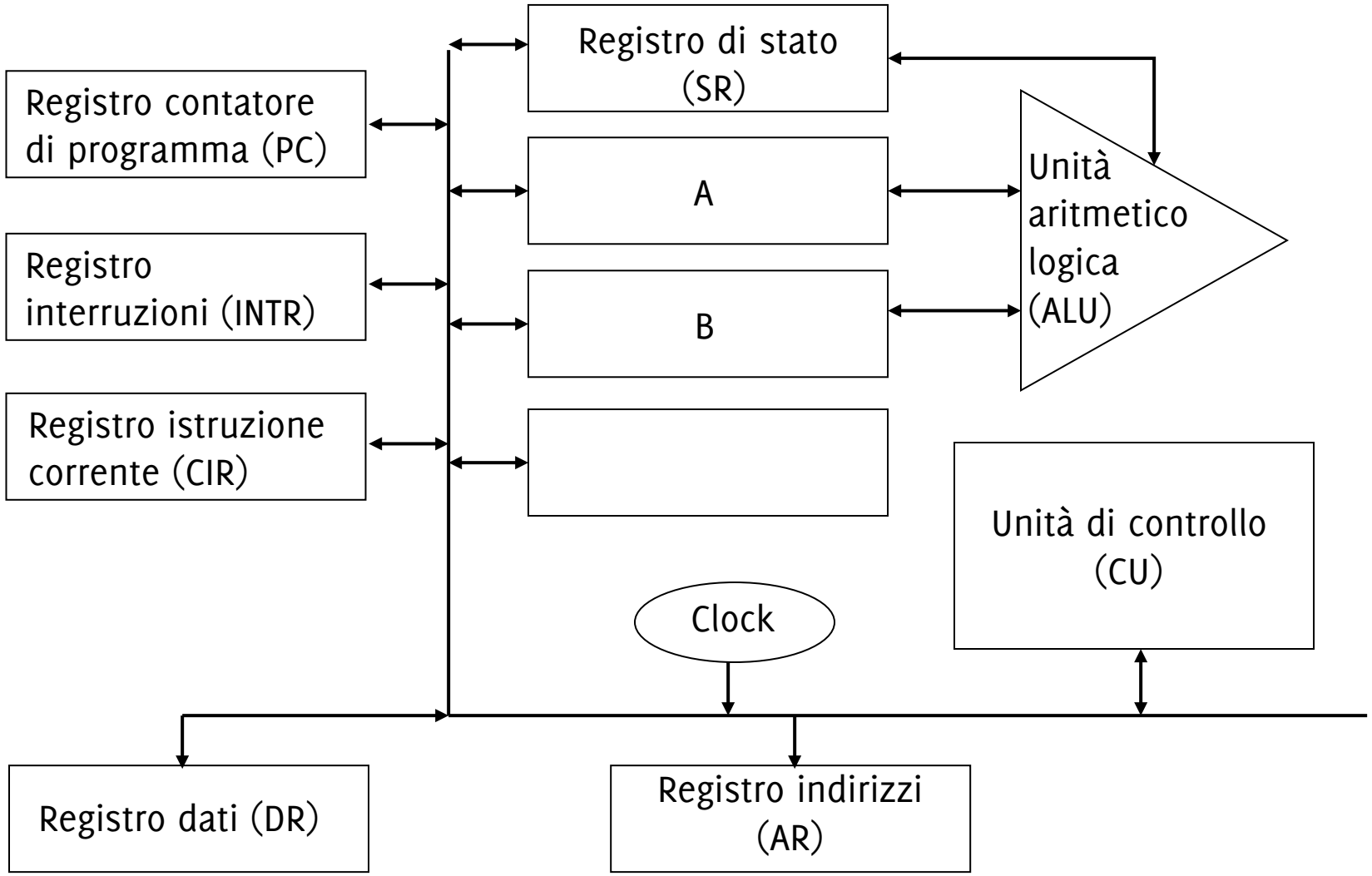
Le istruzioni possono comportare **elaborazione** o **trasferimento** dell'informazione

La CPU contiene a sua volta:

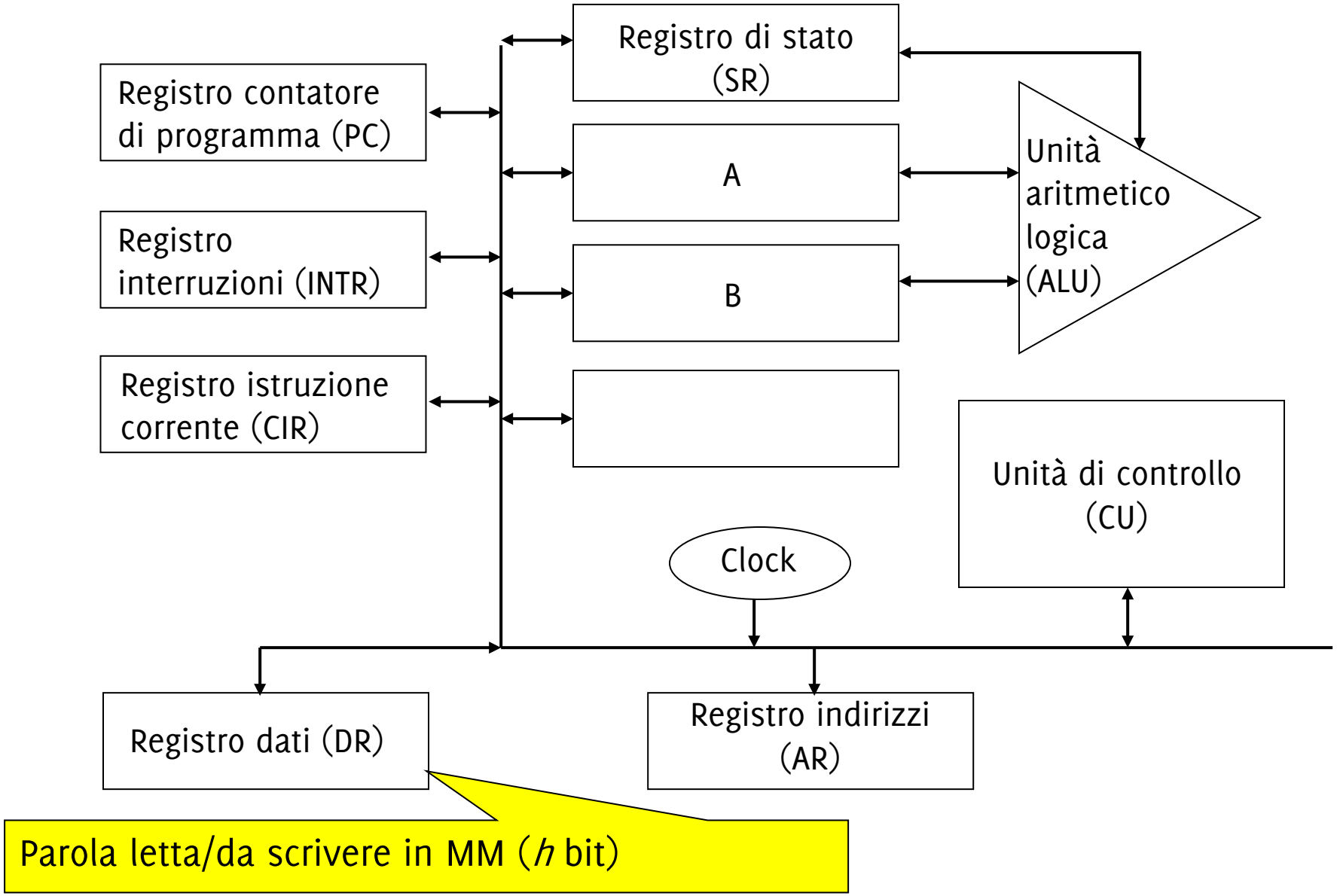
- l'**Unità di Controllo** che preleva e decodifica istruzioni dalla MM, invia segnali per eseguire le istruzioni
- Il **Clock di sistema**, opera come un metronomo per la CPU
- L'**Unità Aritmetico Logica:** operazioni aritmetiche e logiche

La CPU contiene inoltre molti **registri**: memorie «rapide» per informazioni richieste dalla CU (es due numeri da sommare, il loro risultato)

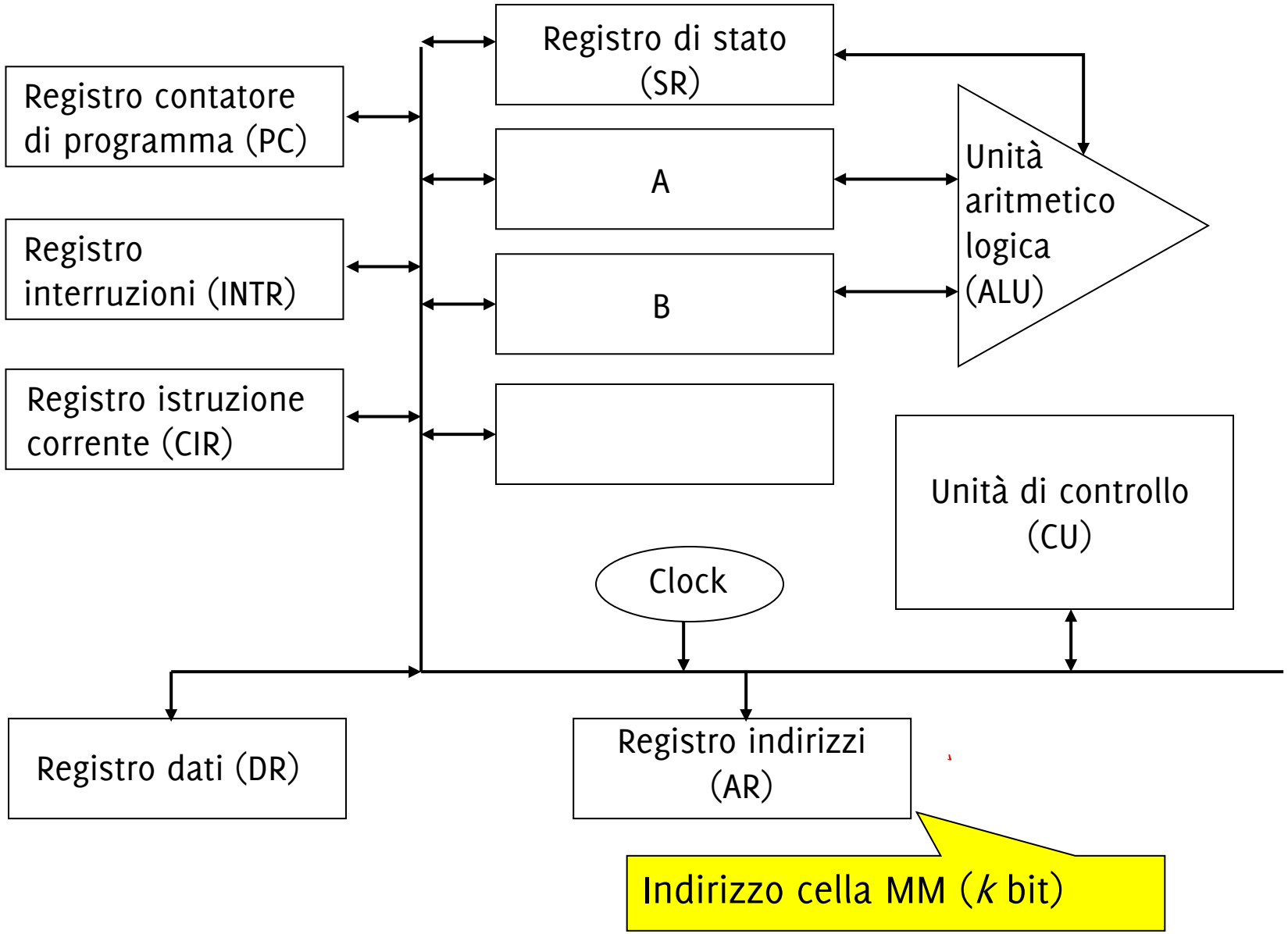
L'Unità di Elaborazione (CPU)



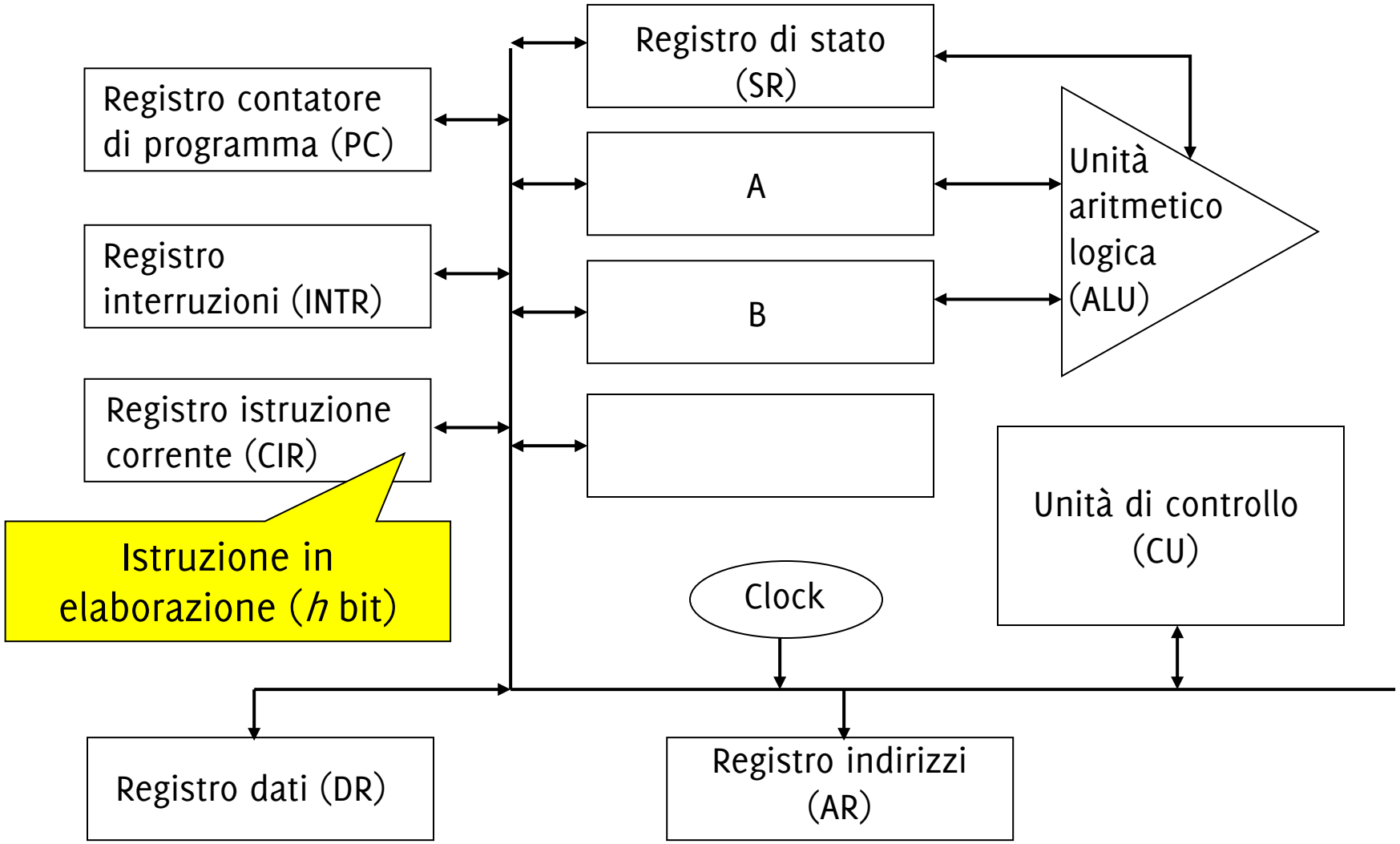
L'Unità di Elaborazione (CPU)



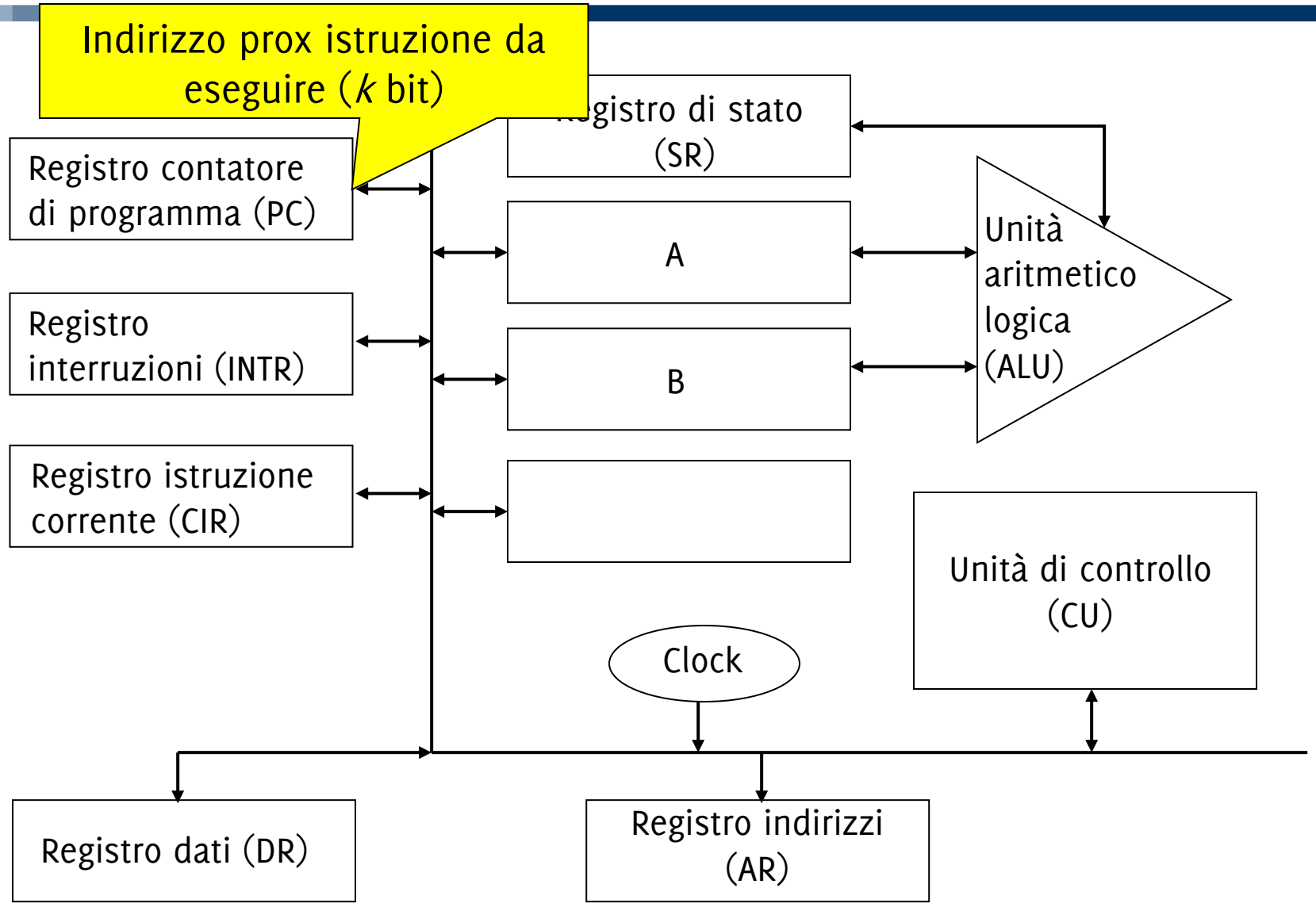
L'Unità di Elaborazione (CPU)



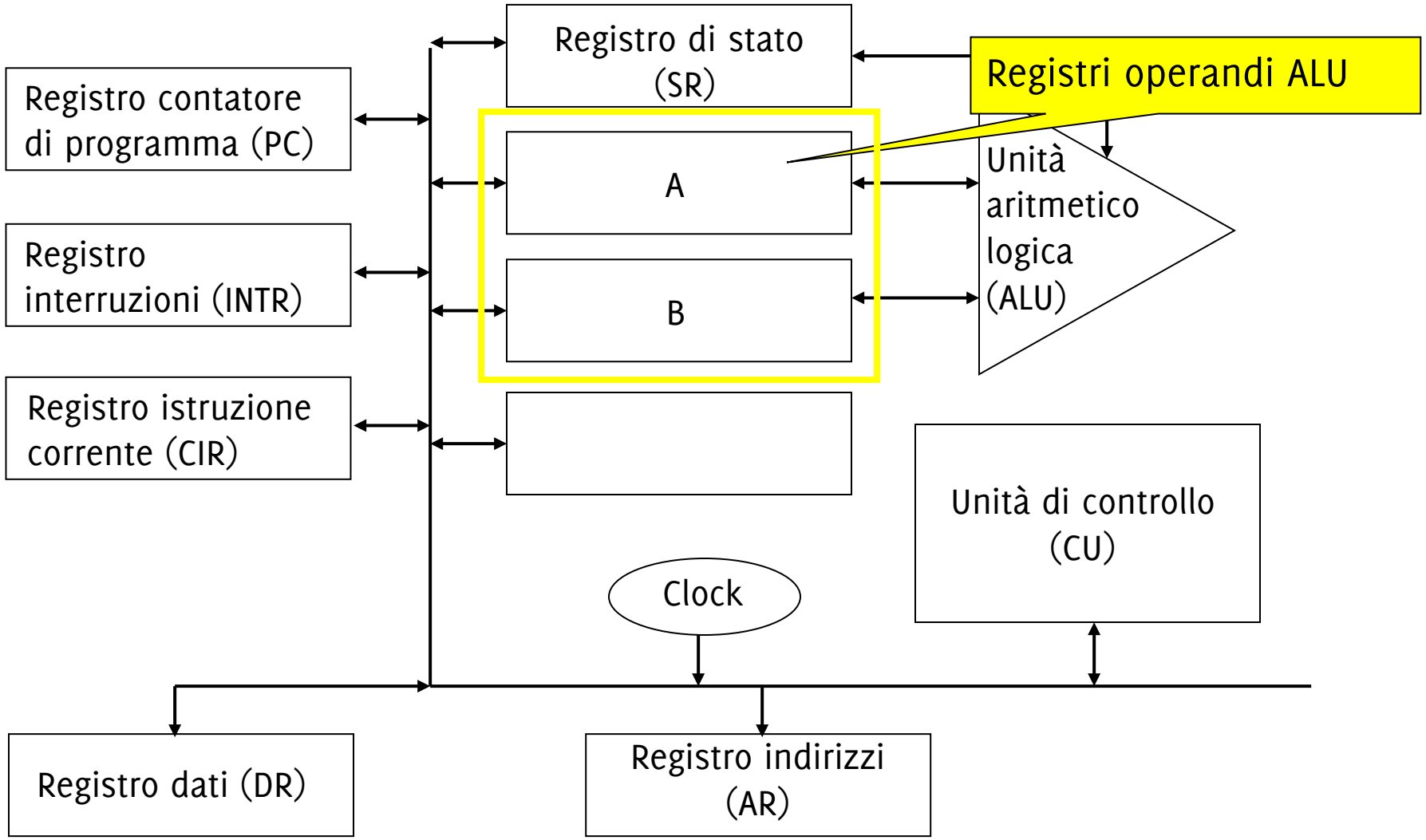
L'Unità di Elaborazione (CPU)



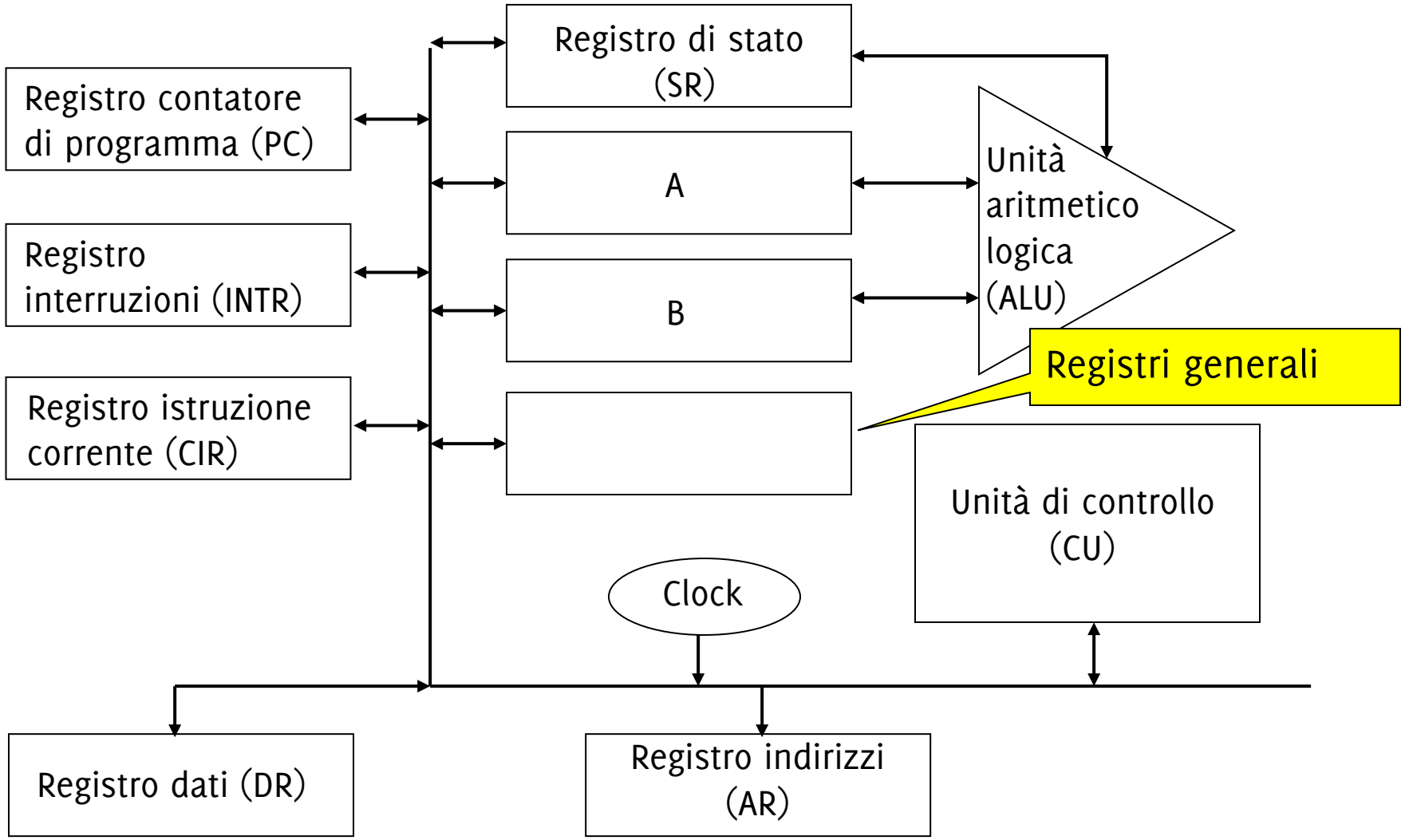
L'Unità di Elaborazione (CPU)



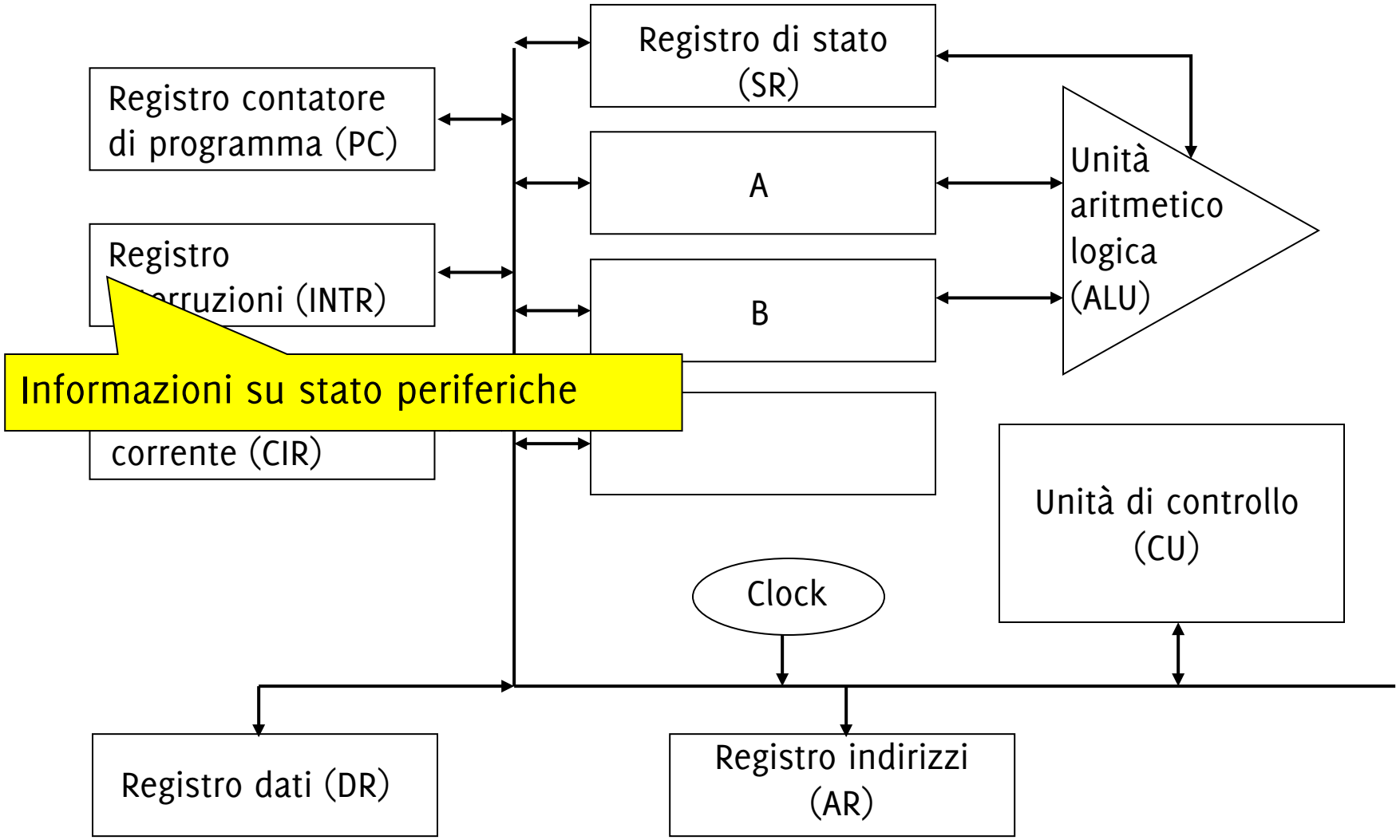
L'Unità di Elaborazione (CPU)



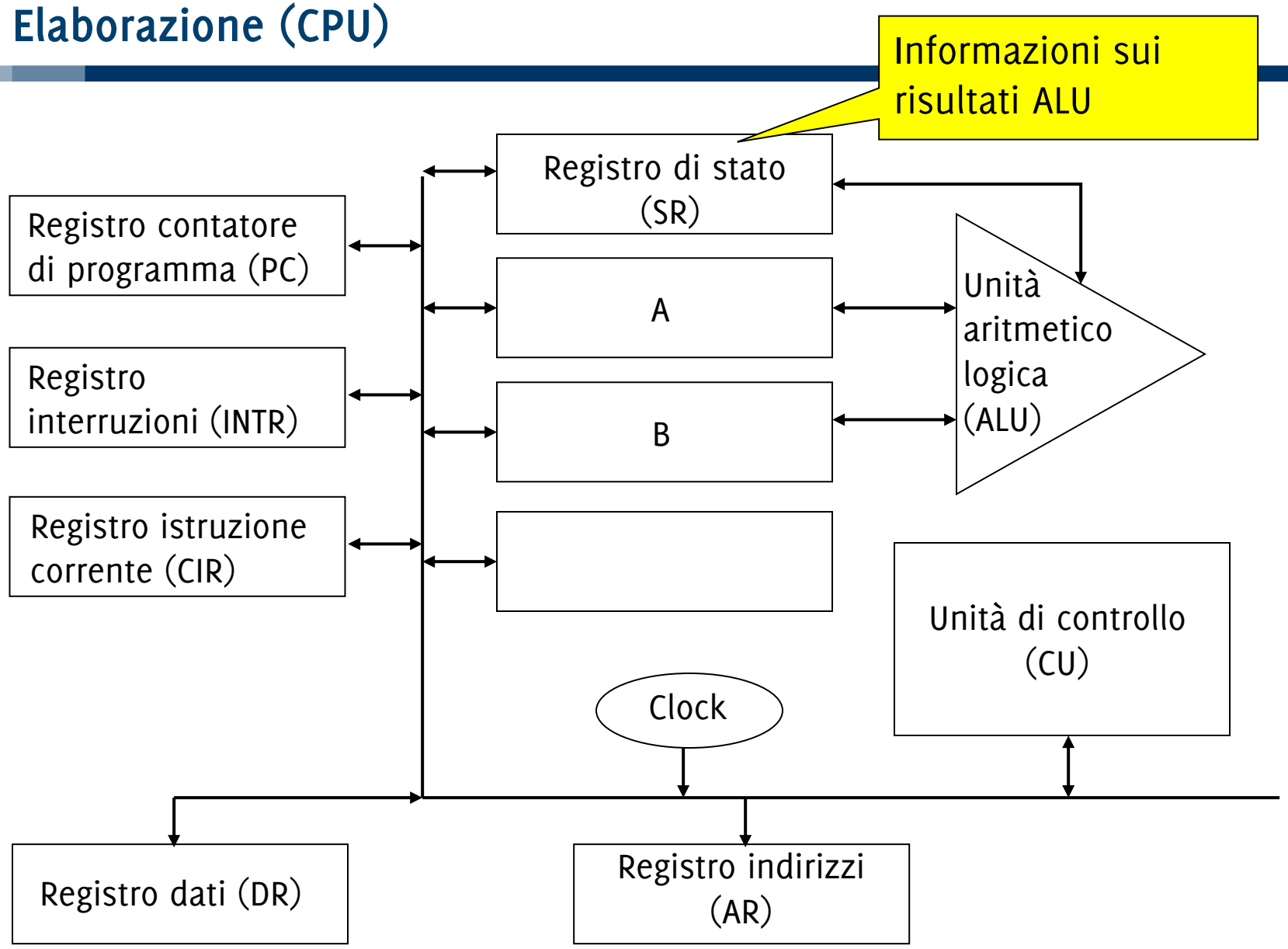
L'Unità di Elaborazione (CPU)



L'Unità di Elaborazione (CPU)



L'Unità di Elaborazione (CPU)



Velocità della CPU

Soluzioni di diversi problemi importanti richiedono una grande potenza di calcolo:

I progettisti tentano di rendere le macchine più veloci. Fino ad un certo punto, **le macchine possono essere accelerate velocizzando l'hardware**.

Limiti fisici di vario tipo si profilano all'orizzonte.

- **Niente può andare più veloce della luce**, che è di circa trenta centimetri per nanosecondo nel vuoto e di venti centimetri per nanosecondo in un filo di rame.
- Per costruire un computer con un tempo di un nanosecondo per istruzione, la distanza totale su cui il segnale elettrico può viaggiare, all'interno della CPU, verso la memoria e ritorno, non può essere maggiore di venti centimetri.
- **Perciò computer molto veloci devono essere molto piccoli**. Sfortunatamente i computer veloci producono più calore di quelli lenti, e comprimere i computer in uno spazio ristretto rende **difficile dissipare il calore**.
- **I super computer sono immersi nel freon**, come liquido di raffreddamento, per trasferire all'esterno il calore il più velocemente possibile.

Parallelismo

Tuttavia è possibile velocizzare le operazioni il problema da un'altra prospettiva.

Invece di un'unica CPU ad alta velocità, **è possibile costruire una macchina con molte ALU più lente (ed economiche) o perfino con diverse CPU complete**, per ottenere la stessa potenza di calcolo ad un costo minore.

Molta ricerca è dedicata alla costruzione di macchine parallele.

Bus di Sistema

Bus di Sistema

È un insieme di connessioni che permettono di trasferire l'informazione tra **due** entità funzionali

- Un'entità trasmette, l'altra riceve

Due soli tipi di connessioni logiche, **stabilite** dalla **CPU**:

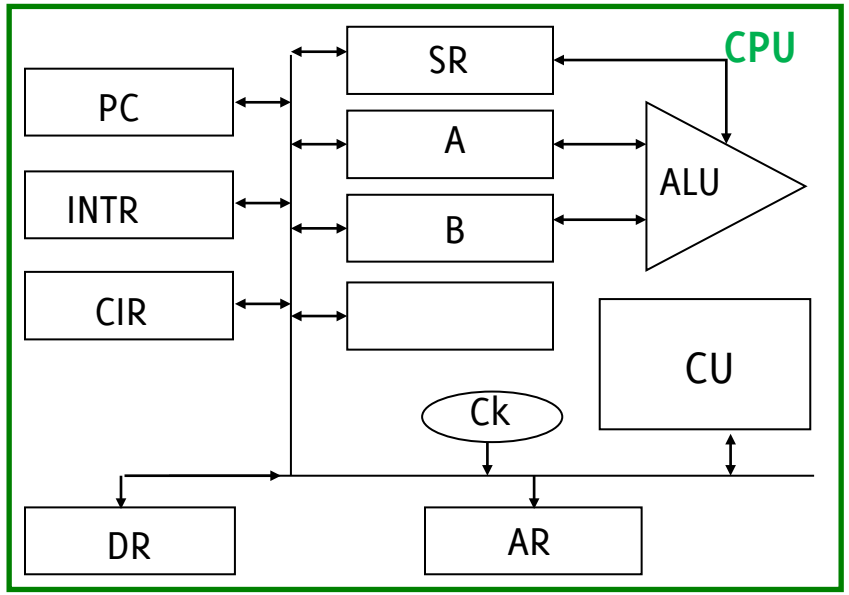
- CPU (**master**) - memoria (**slave**)
- CPU (**master**) - interfaccia periferica (**slave**)

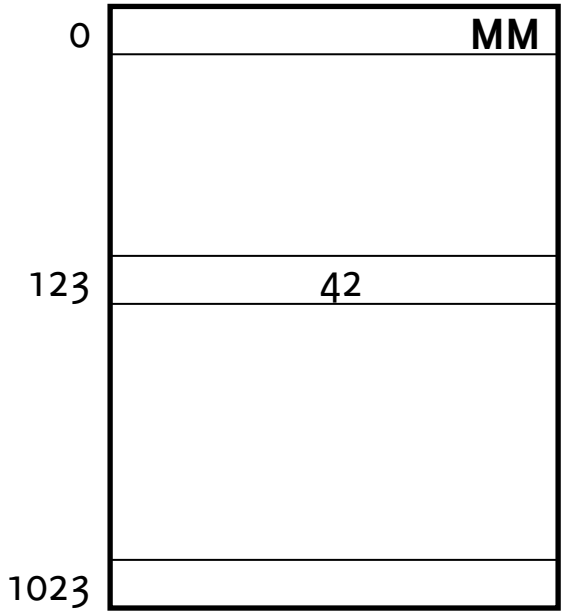
le connessioni fisiche sono sempre presenti.

Ci sono **tre tipi di linee**, con tre funzionalità diverse

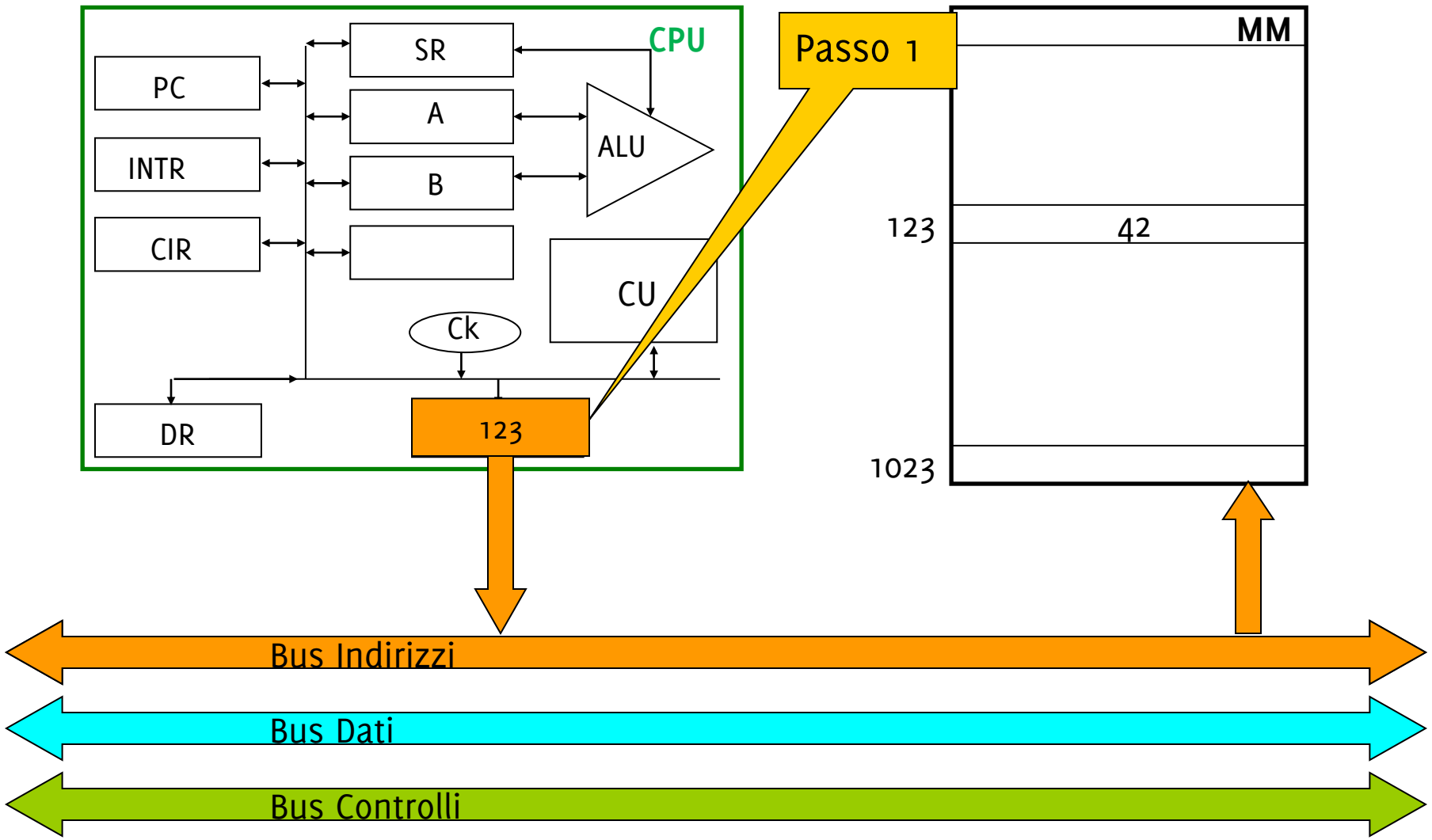
- Bus **dati**
- Bus **indirizzi**
- Bus **controlli**: usato dal master per **trasmettere le istruzioni** da eseguire allo slave, e dallo slave per dare **feedback**

Sequenza di Lettura

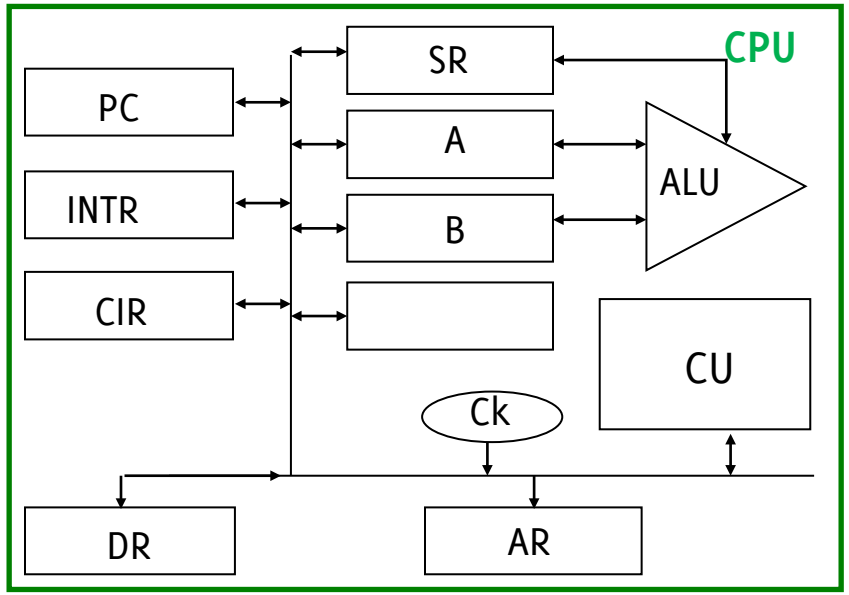


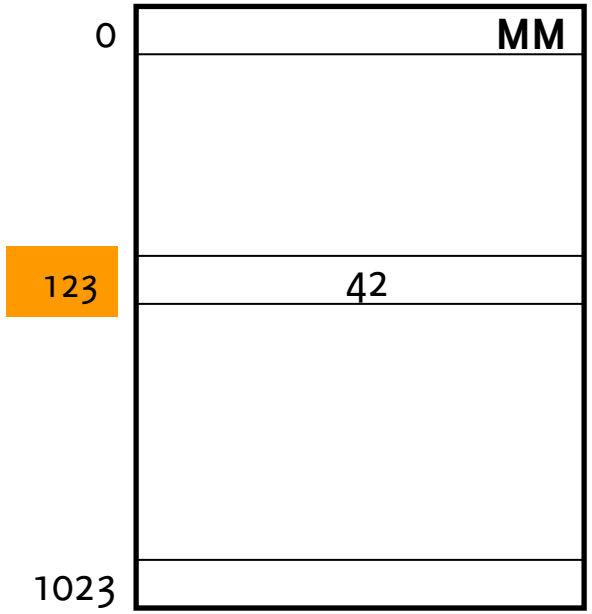


Sequenza di Lettura

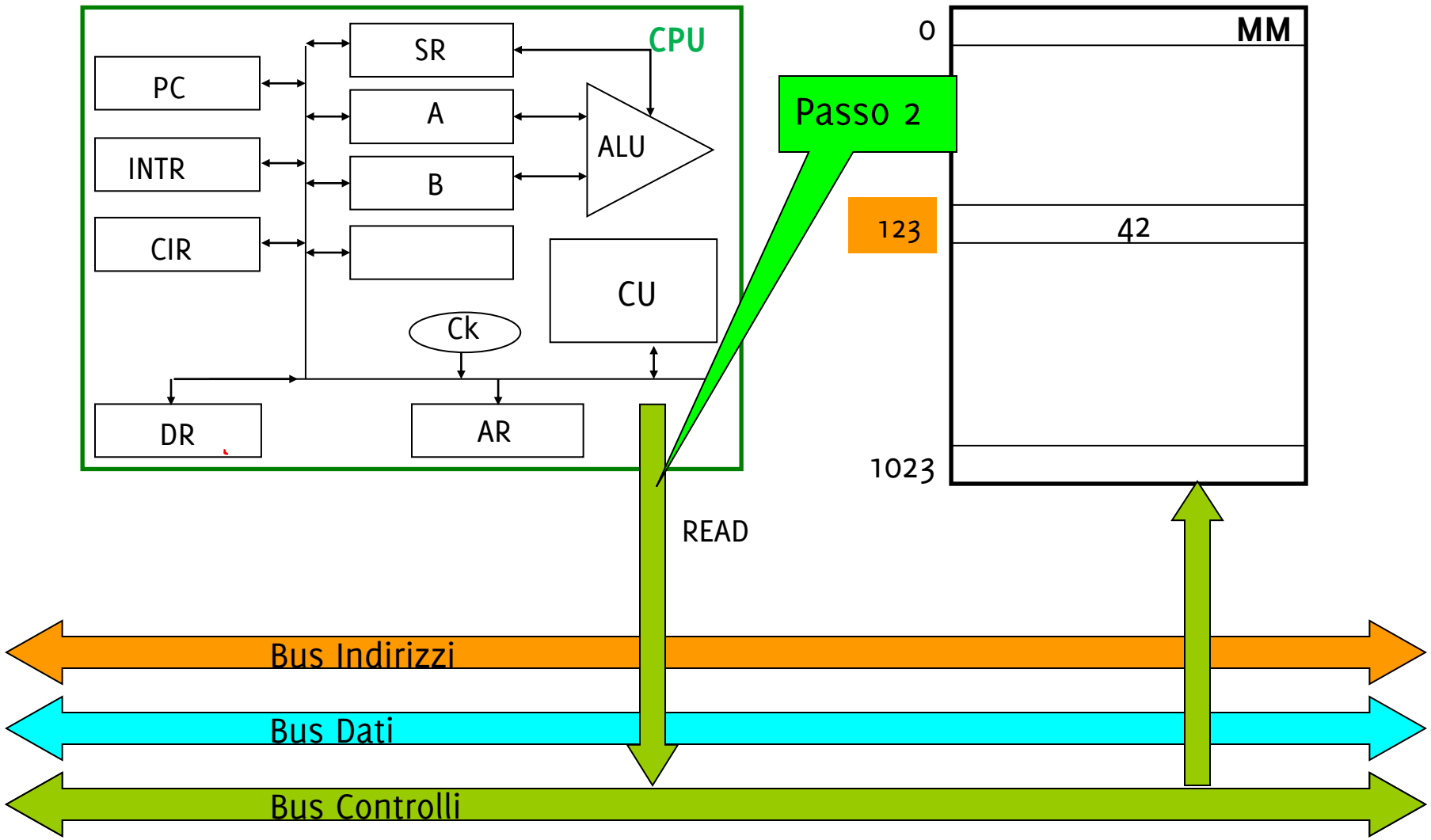


Sequenza di Lettura

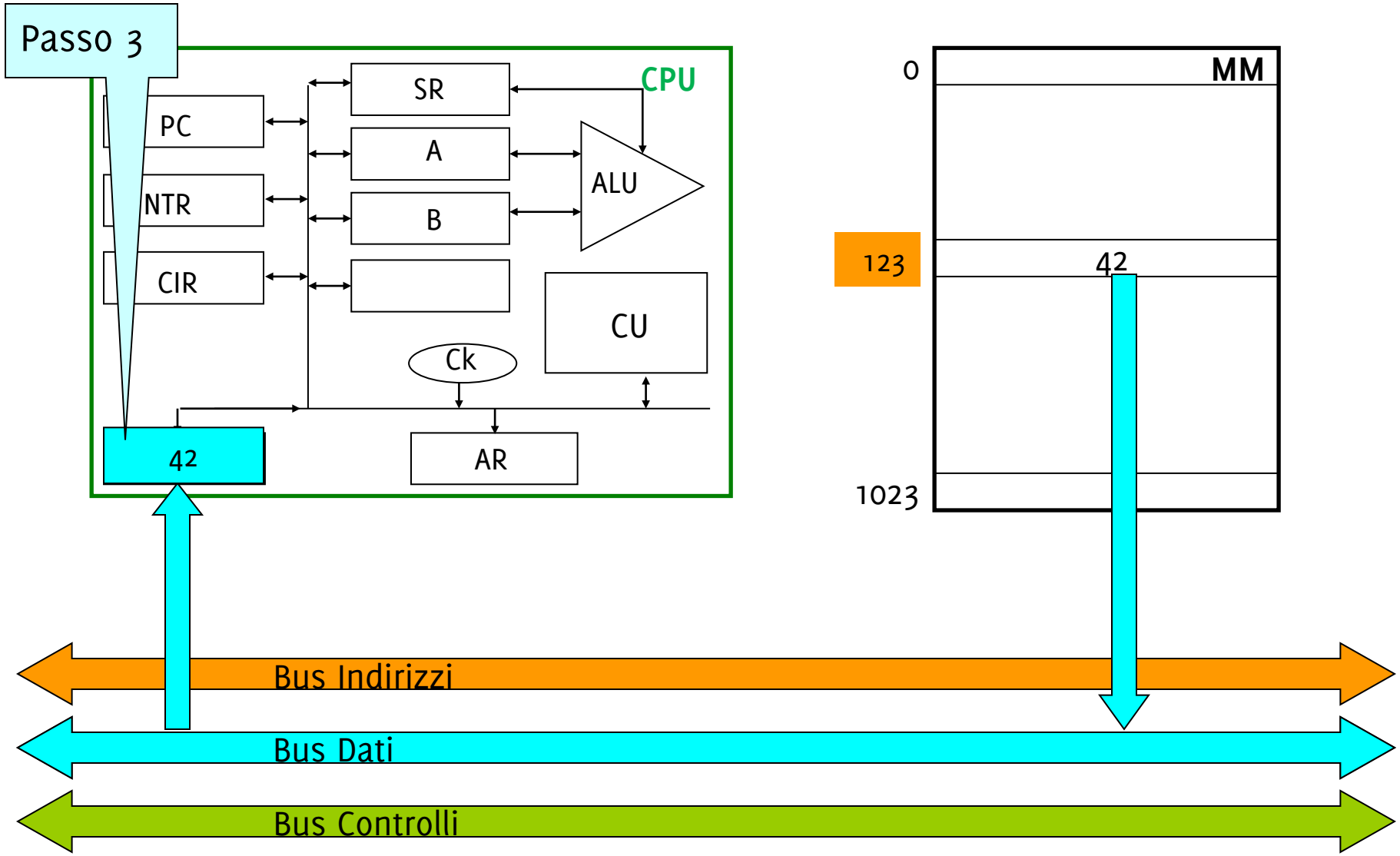




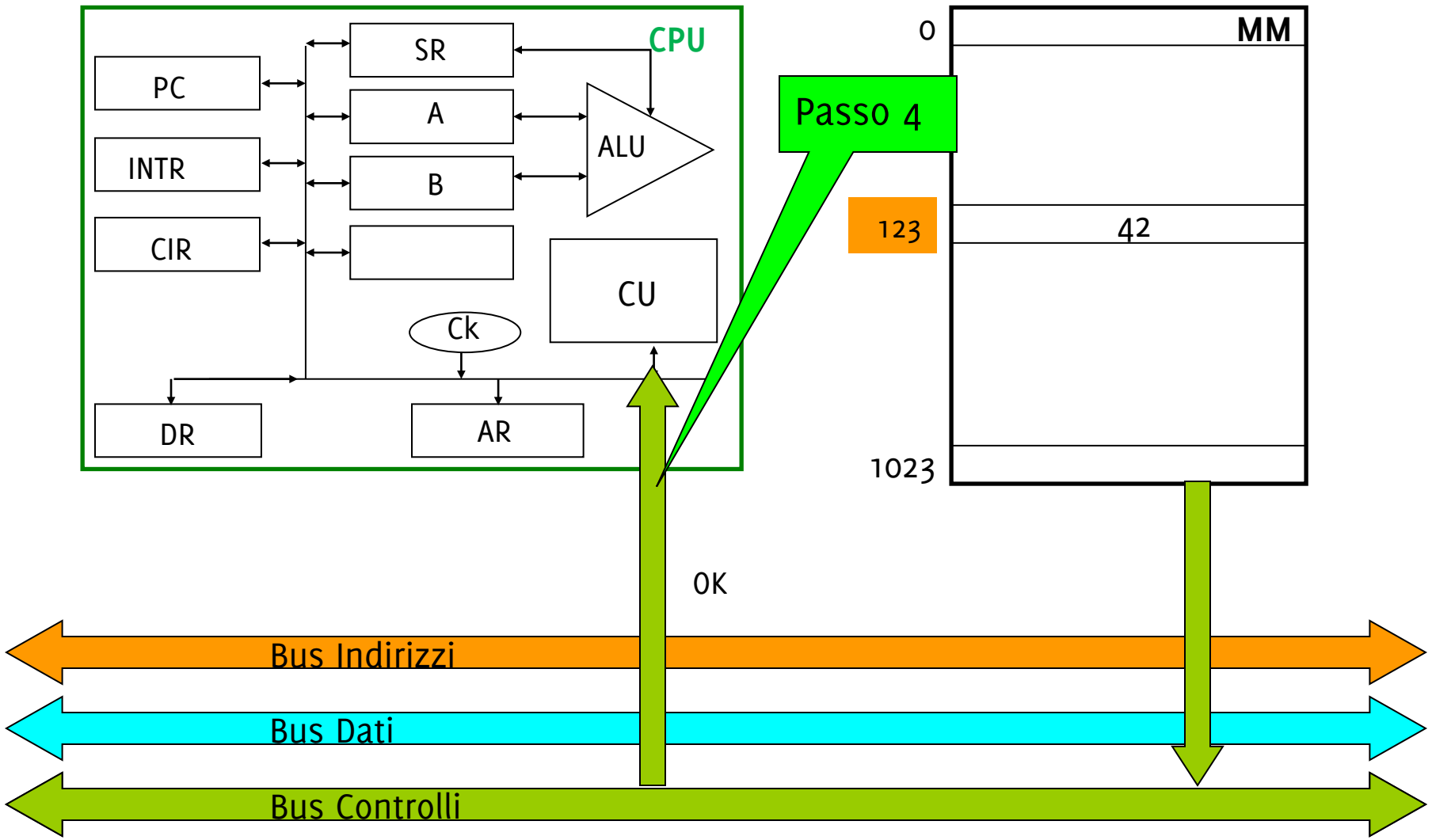
Sequenza di Lettura



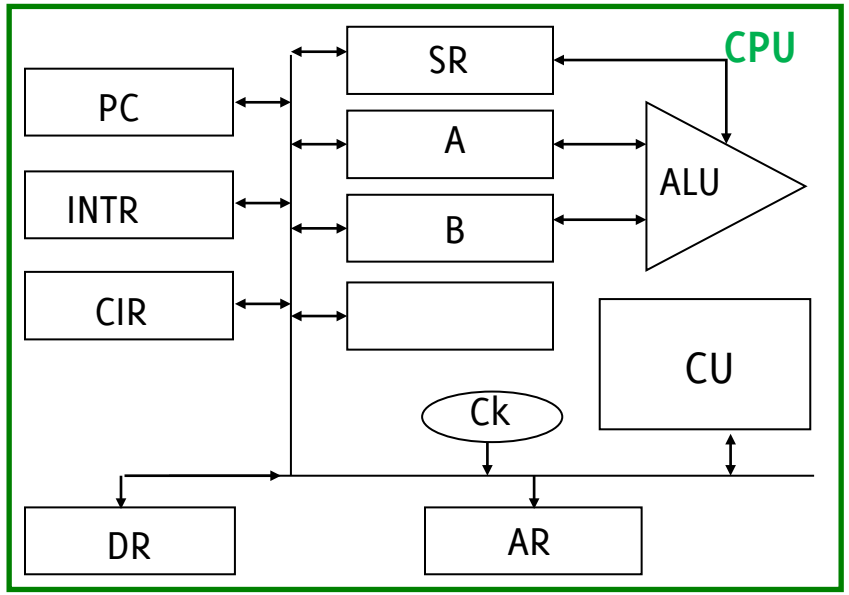
Sequenza di Lettura

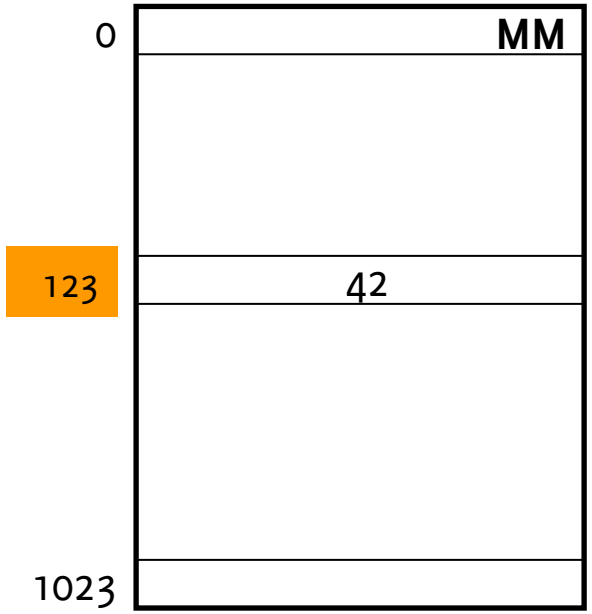


Sequenza di Lettura

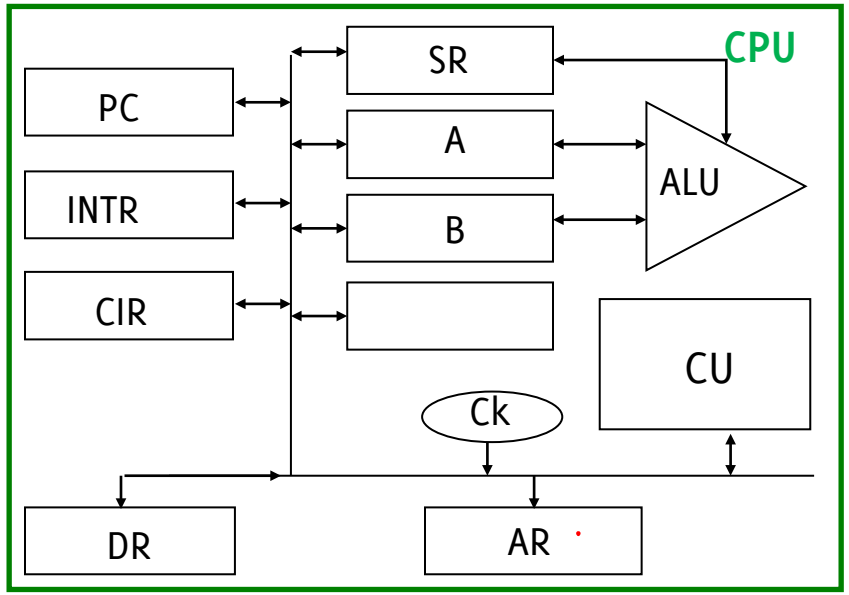


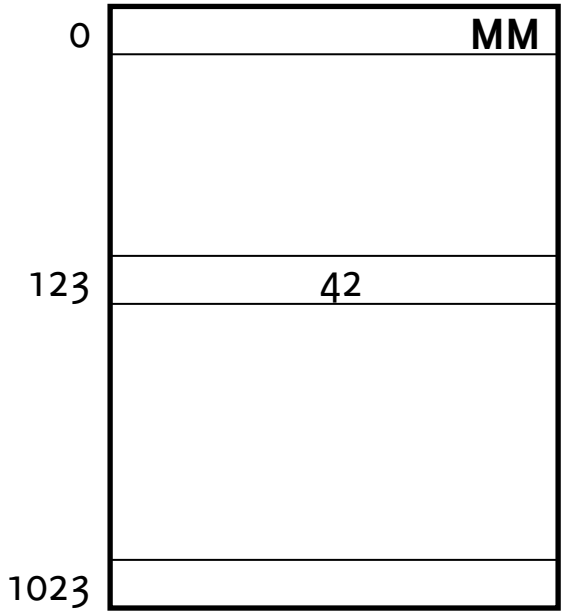
Sequenza di Lettura



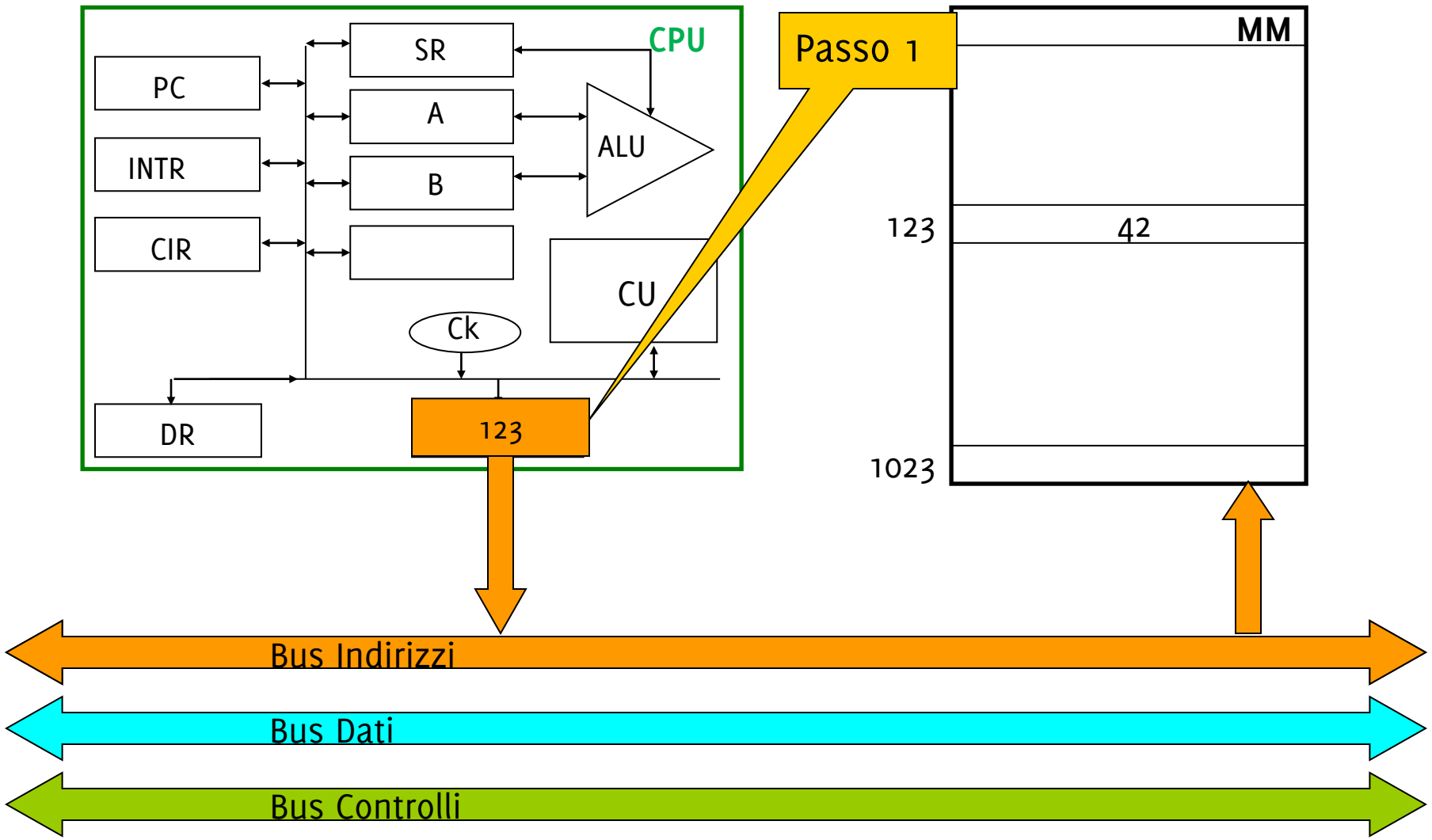


Sequenza di Scrittura

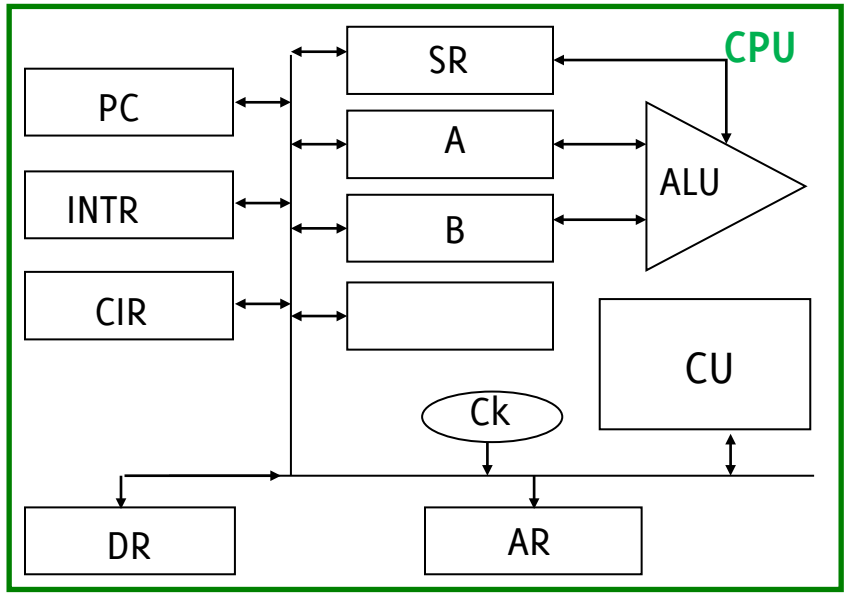


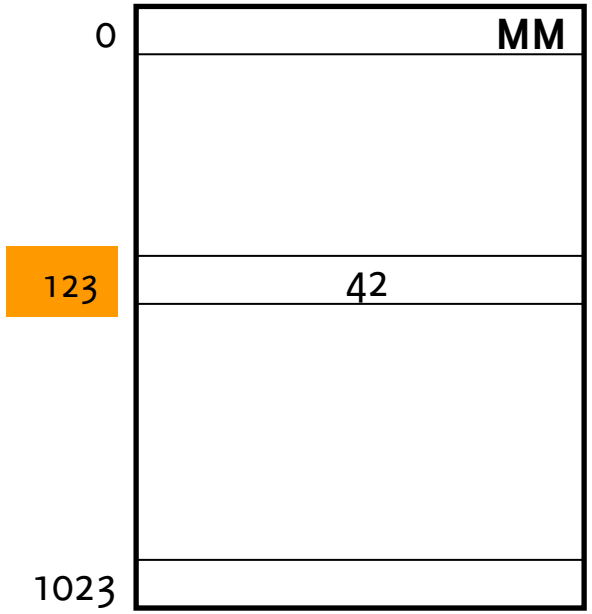


Sequenza di Scrittura

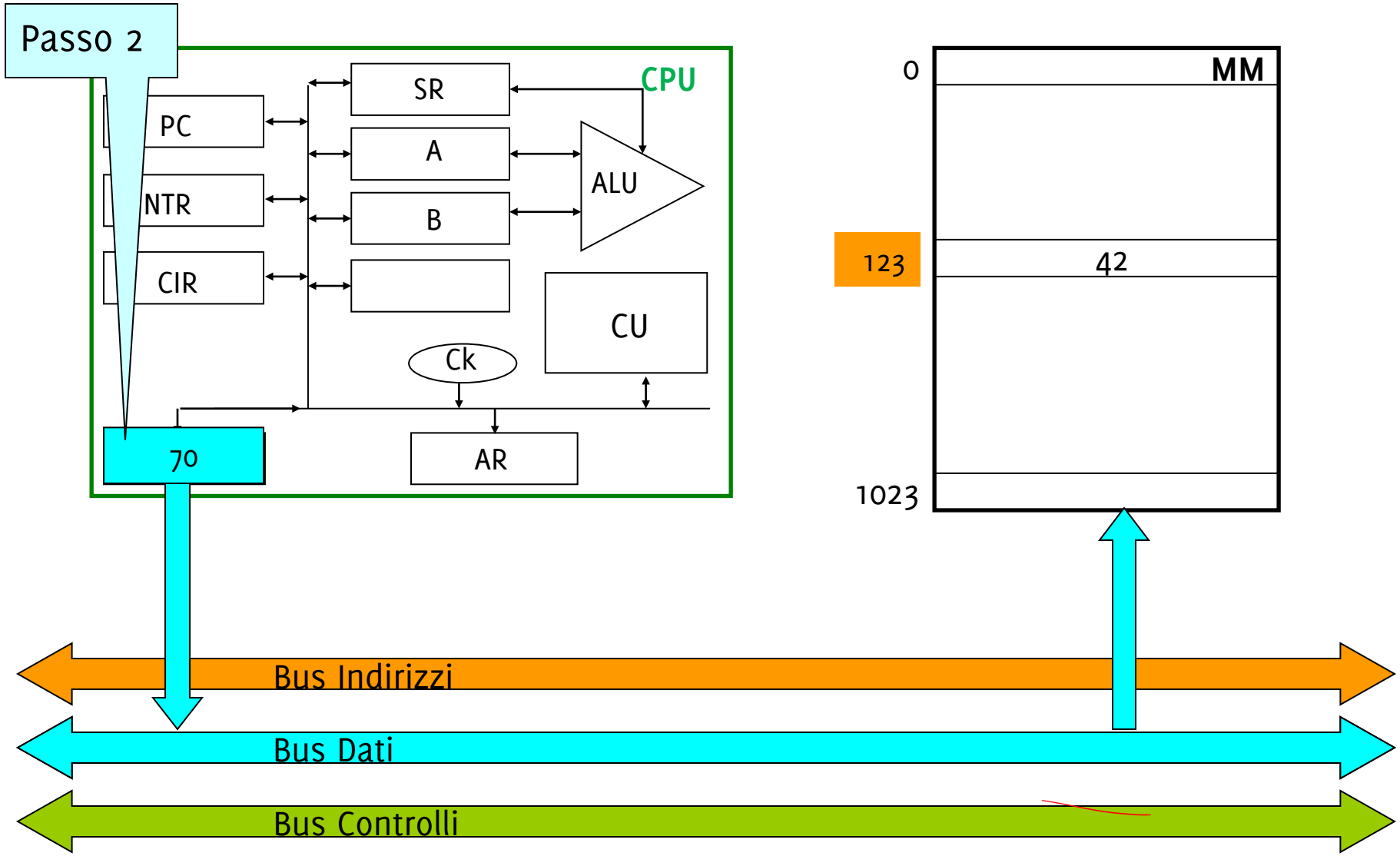


Sequenza di Scrittura

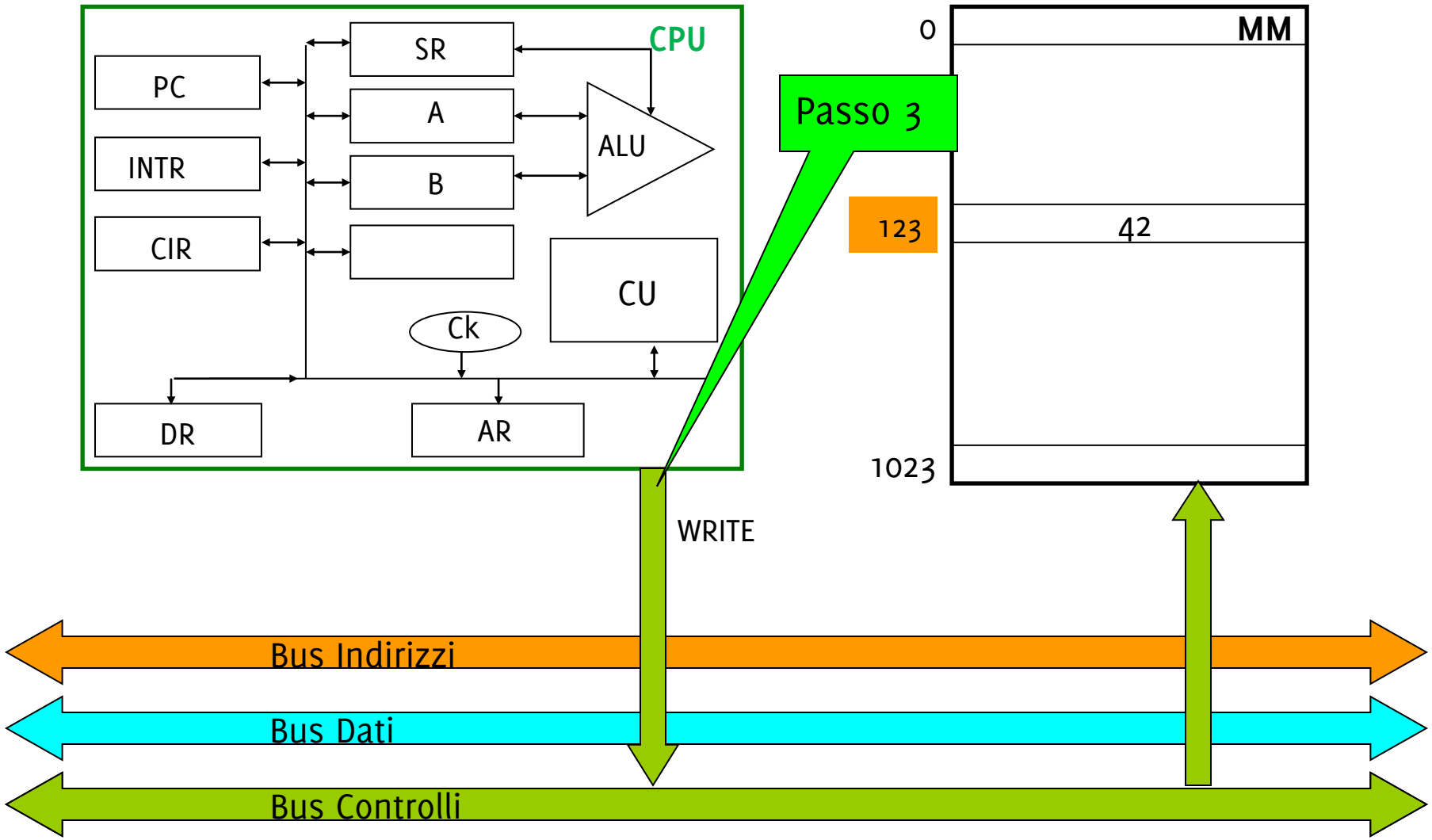




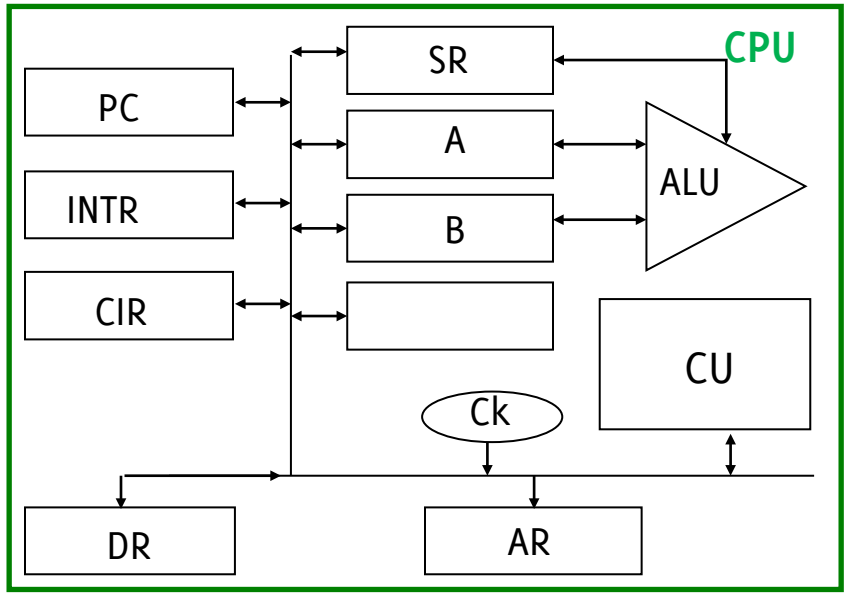
Sequenza di Scrittura

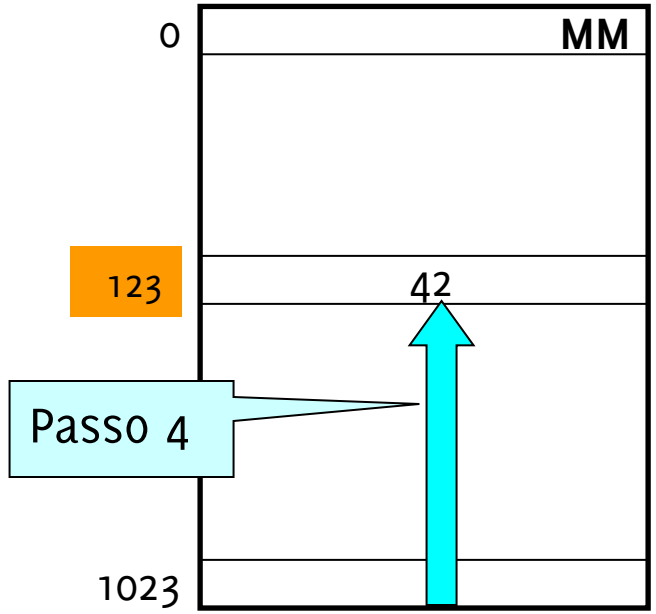


Sequenza di Scrittura

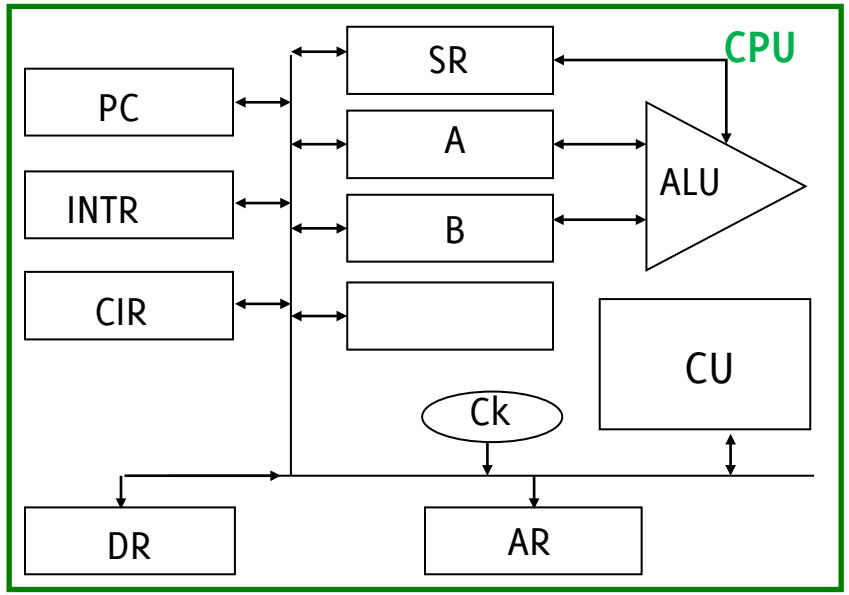


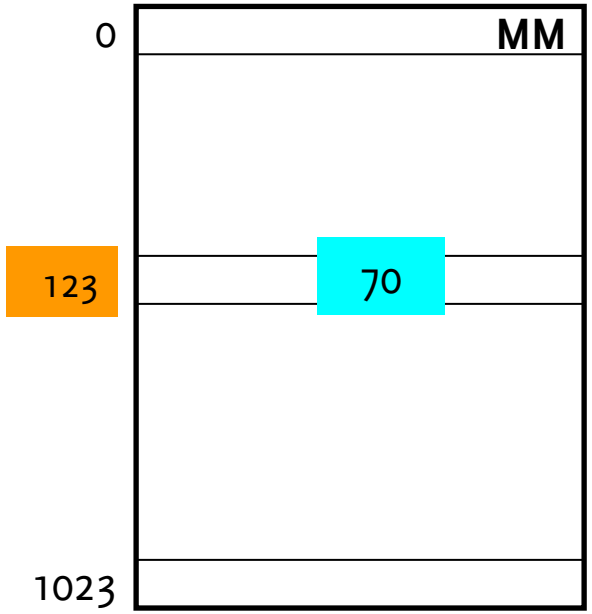
Sequenza di Scrittura



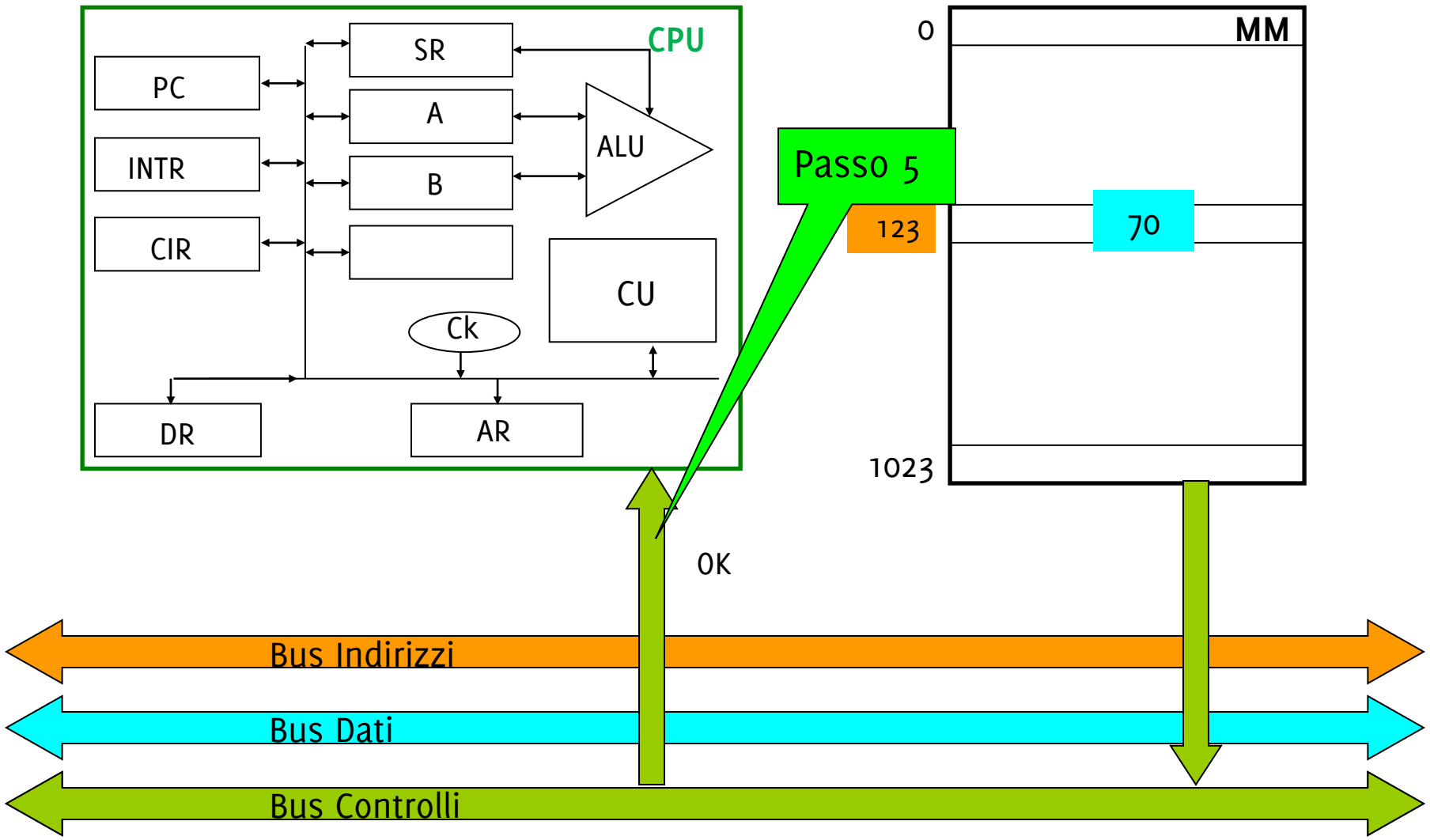


Sequenza di Scrittura

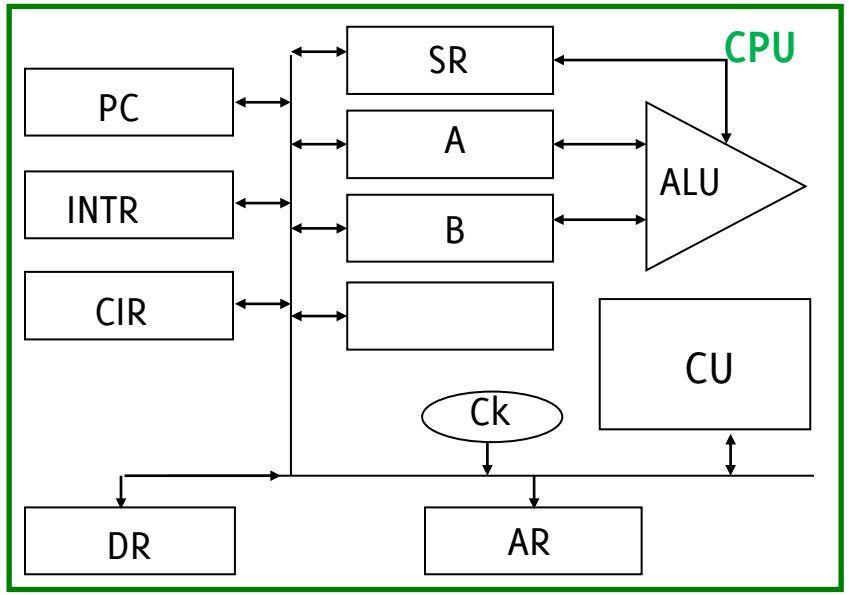


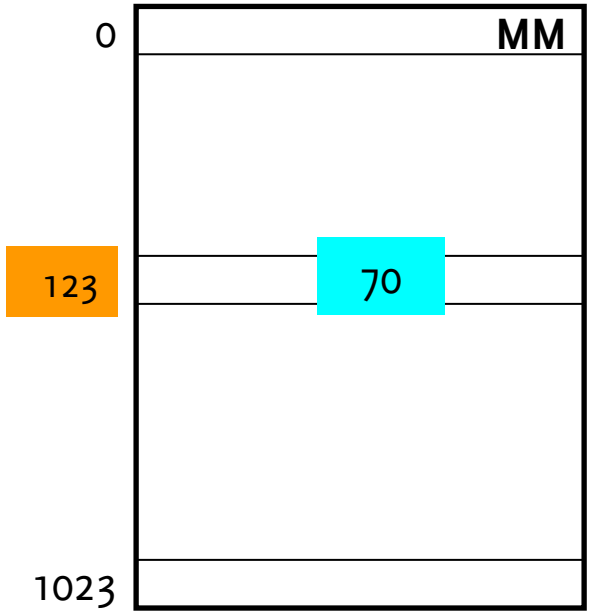


Sequenza di Scrittura



Sequenza di Scrittura





Interfacce delle Periferiche

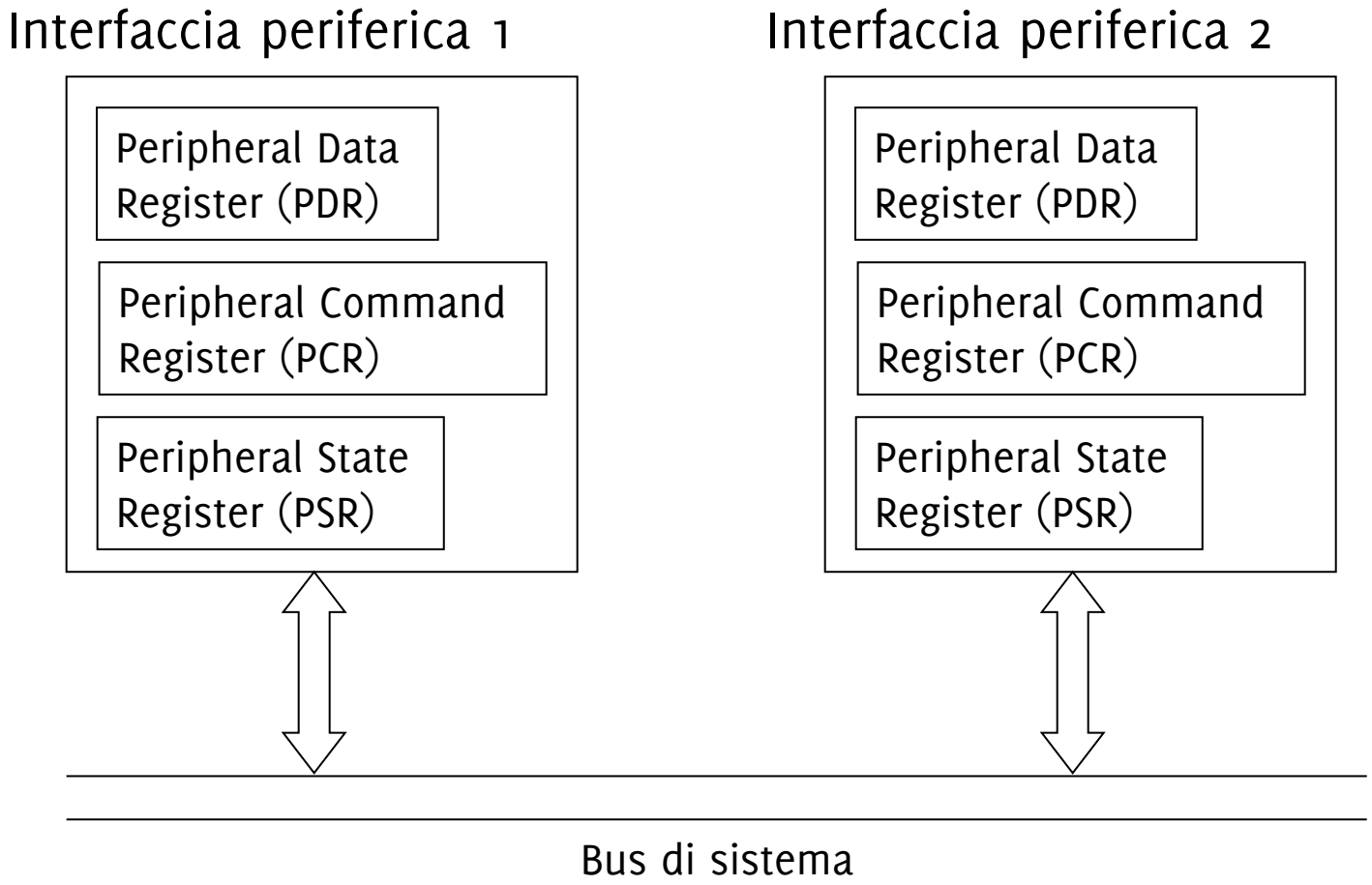
Interfacce alle Periferiche

Le interfacce collegano il calcolatore a periferiche esterne

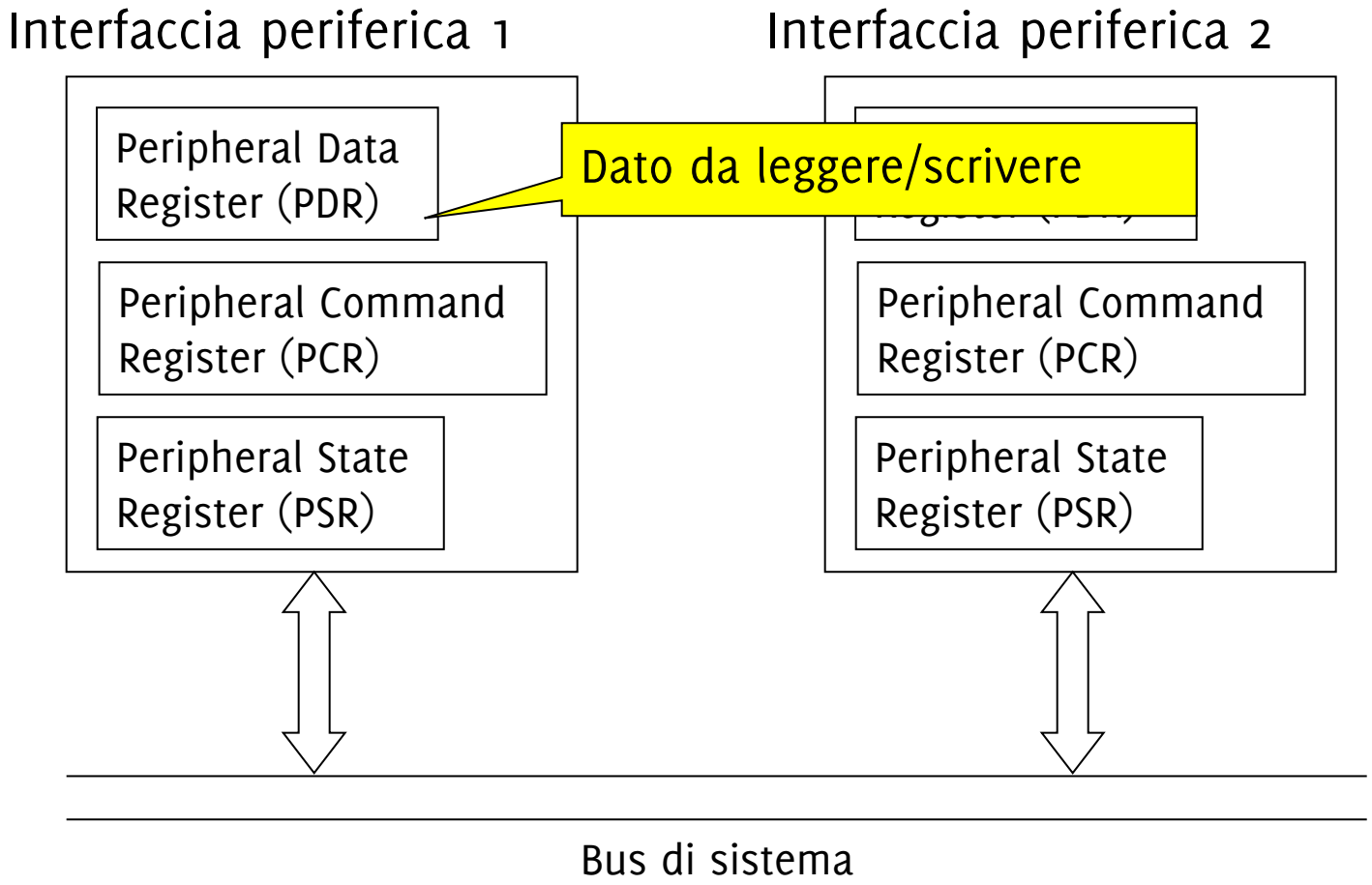
Ogni interfaccia contiene dei registri per lo scambio dei dati con la periferica

- Registro dati
- Registro comando della periferica
- Registro di stato

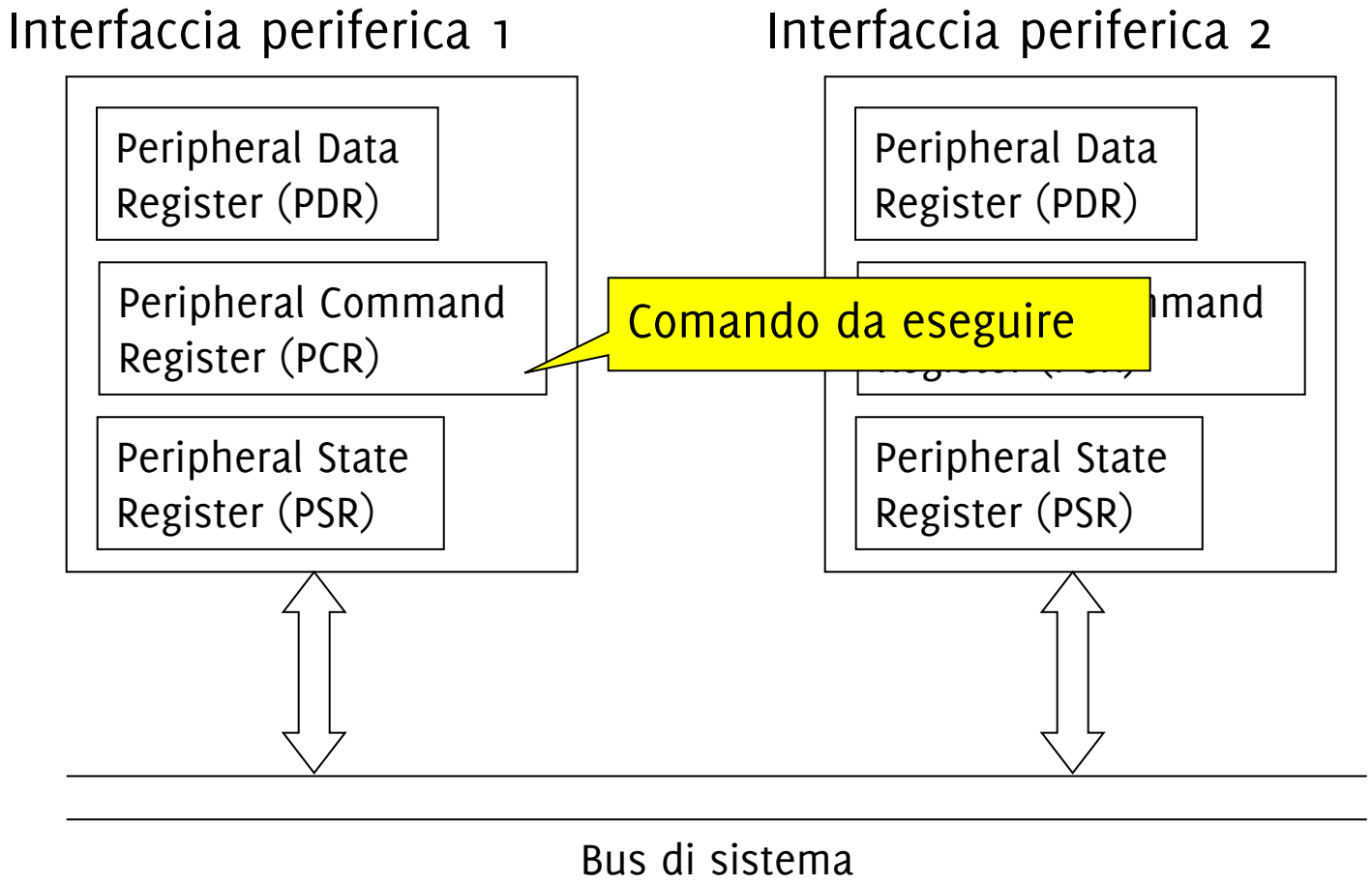
Interfacce alle Periferiche



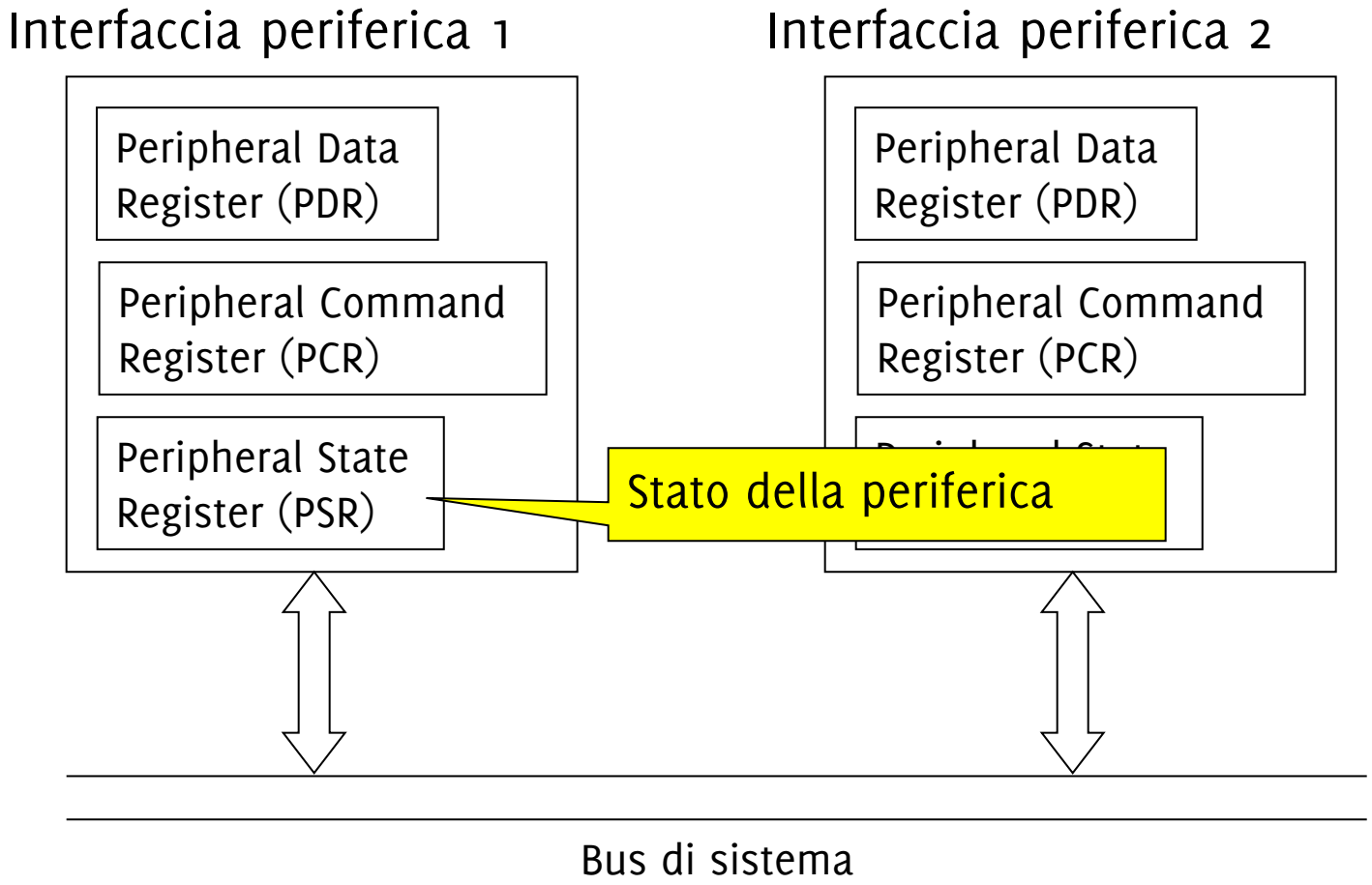
Interfacce alle Periferiche



Interfacce alle Periferiche



Interfacce alle Periferiche



I Programmi Nella Macchina di Von Neumann

I Programmi nella Macchina di Von Neumann

Le **istruzioni** sono (necessariamente) codificate in **binario** e, **come i dati,** sono salvate in **parole nella MM**

I Programmi nella Macchina di Von Neumann

Le **istruzioni** sono (necessariamente) codificate in **binario** e, **come i dati,** sono salvate in **parole nella MM**

Supponiamo una MM con parole da $h = 16$ bit ed indirizzi da $k = 10$ bit, con istruzioni così codificate:

Codice operativo (4bit) oo Indirizzo Operando (10bit)

ad esempio, 0100oo0000010000

Cod. operativo	Operandi

Costituite (ovviamente) da sequenze di 0 e 1

- campo ***codice operativo*** *(obbligatorio)* specifica l'operazione da eseguire
- campo ***operandi*** *(facoltativo) indica i dati da utilizzare.* Può contenere direttamente il valore o l'indirizzo della cella che contiene il valore (riferimento a una variabile)

I Programmi nella Macchina di Von Neumann

Consideriamo le seguenti istruzioni eseguibili dalla CPU

- Lettura da periferica, scrittura su periferica
- Caricare un dato da MM in un registro della CPU (*load*)
- Salvare in MM un dato di un registro della CPU (*store*)
- Operazioni aritmetiche (le gestisce la ALU)
- **Istruzioni di salto per cambiare il flusso di esecuzione**

I Programmi nella Macchina di Von Neumann

Le **variabili** sono (necessariamente) codificate in **binario** e salvate in **parole nella MM**

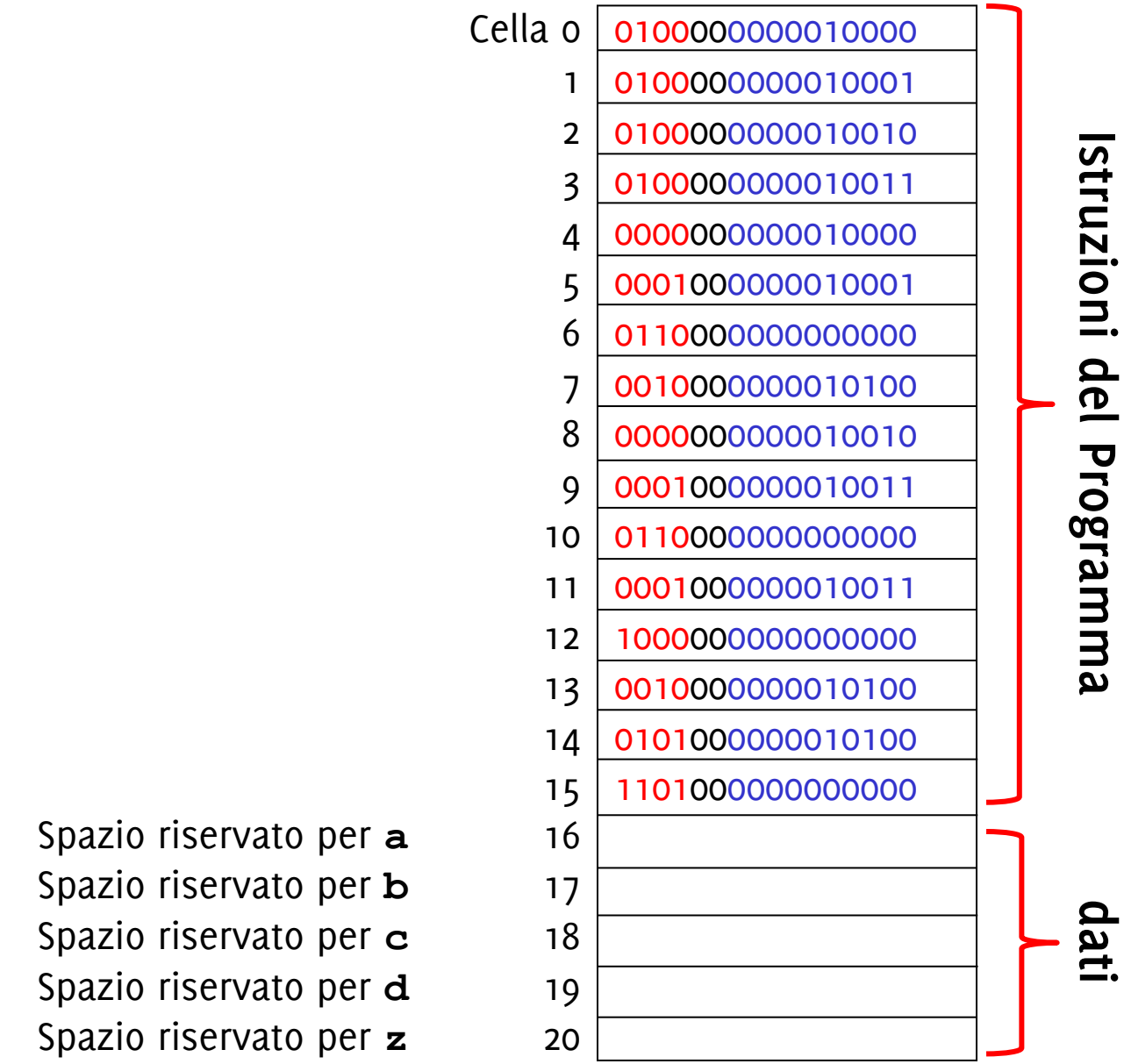
Lo **spazio riservato alle variabili** deve essere:

- **ben definito**
- **Allocato** tipicamente «**sotto**» le **istruzioni del programma**

Alle variabili si **accederà specificando l'indirizzo della parola** di memoria in cui risiedono (niente riferimento simbolico!).

Il **programma** deve in qualche modo «**conoscere**» **gli indirizzi delle variabili!**

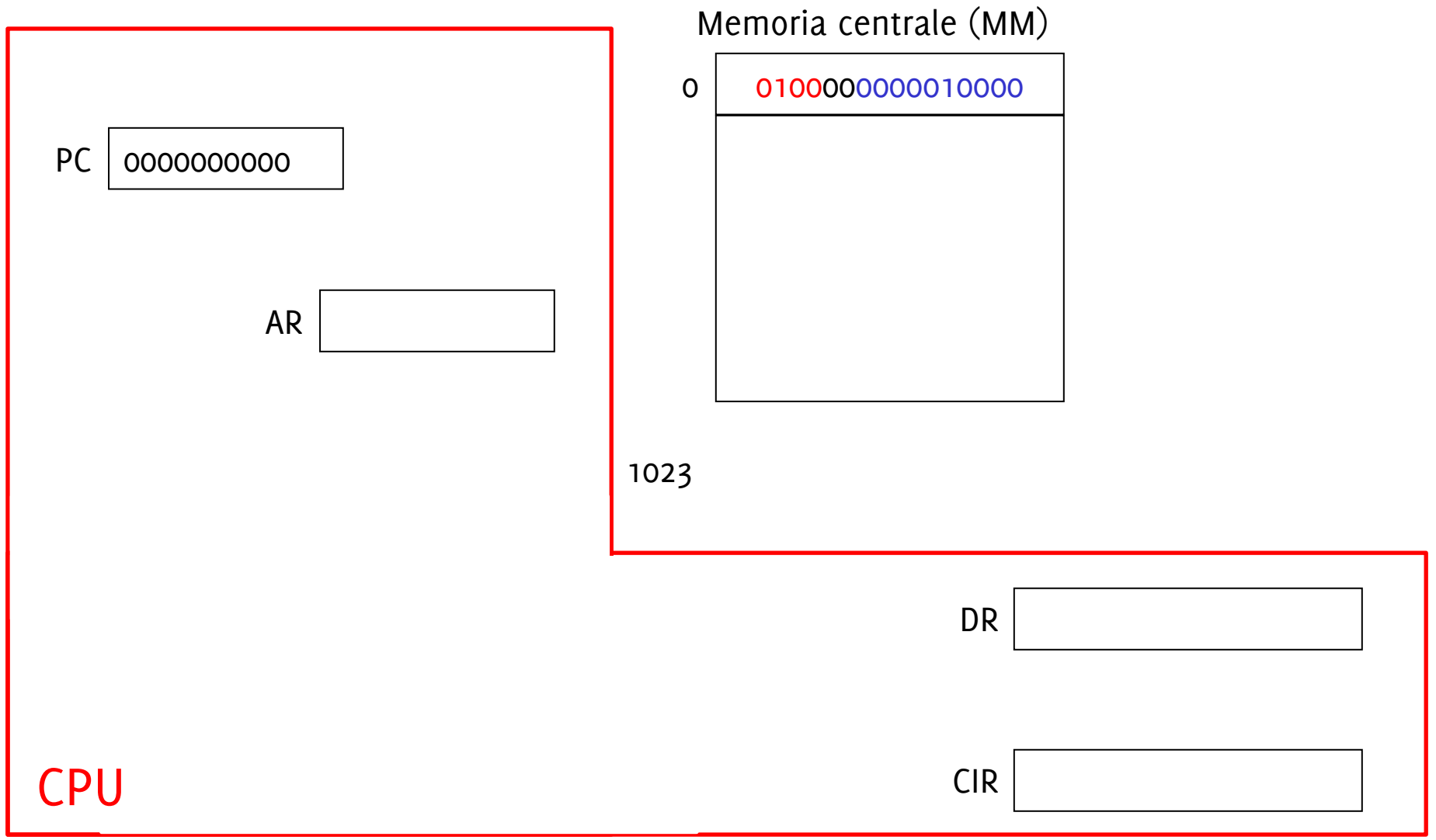
Programma in Memoria Centrale



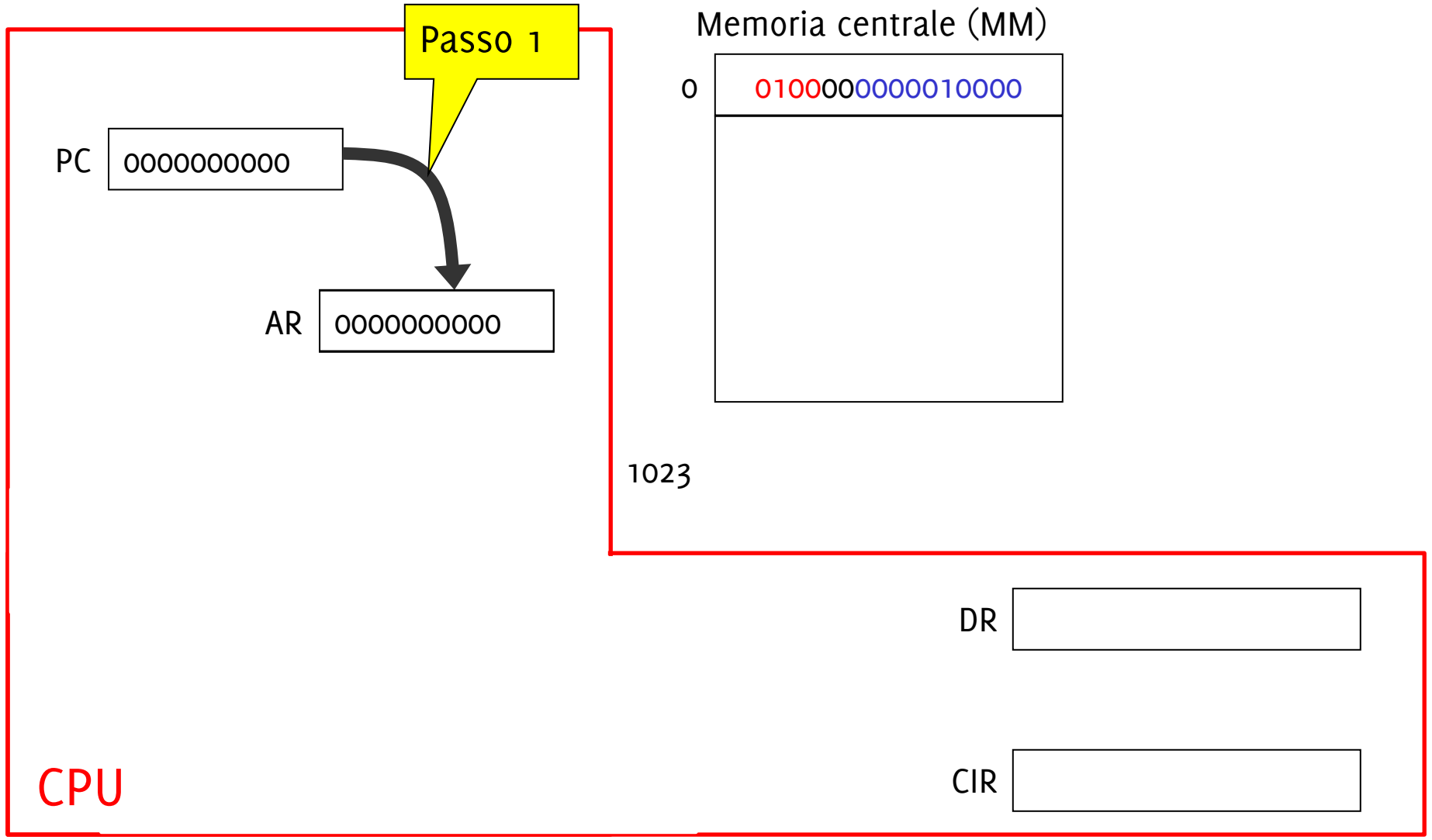
Le Tre Fasi Per Eseguire un'Istruzioni

1. **Fetch:** Acquisizione dell'istruzione dalla MM
    1. Trasferimento da PC a AR dell' indirizzo cella contenente l'istruzione da eseguire.
    2. Lettura dalla MM della cella all'indirizzo in AR, contenuto trasferito sul DR (l'istruzione è un dato)
    3. Sposta da DR a CIR (riferimento istr. in esecuzione)
    4. Incrementa PC (definisce la prossima istruzione: incremento di 1 = sequenzialità)
2. **Decodifica**: riguarda il codice operativo che viene letto dal CIR
3. **Esecuzione**: dipende dall'istruzione specifica.

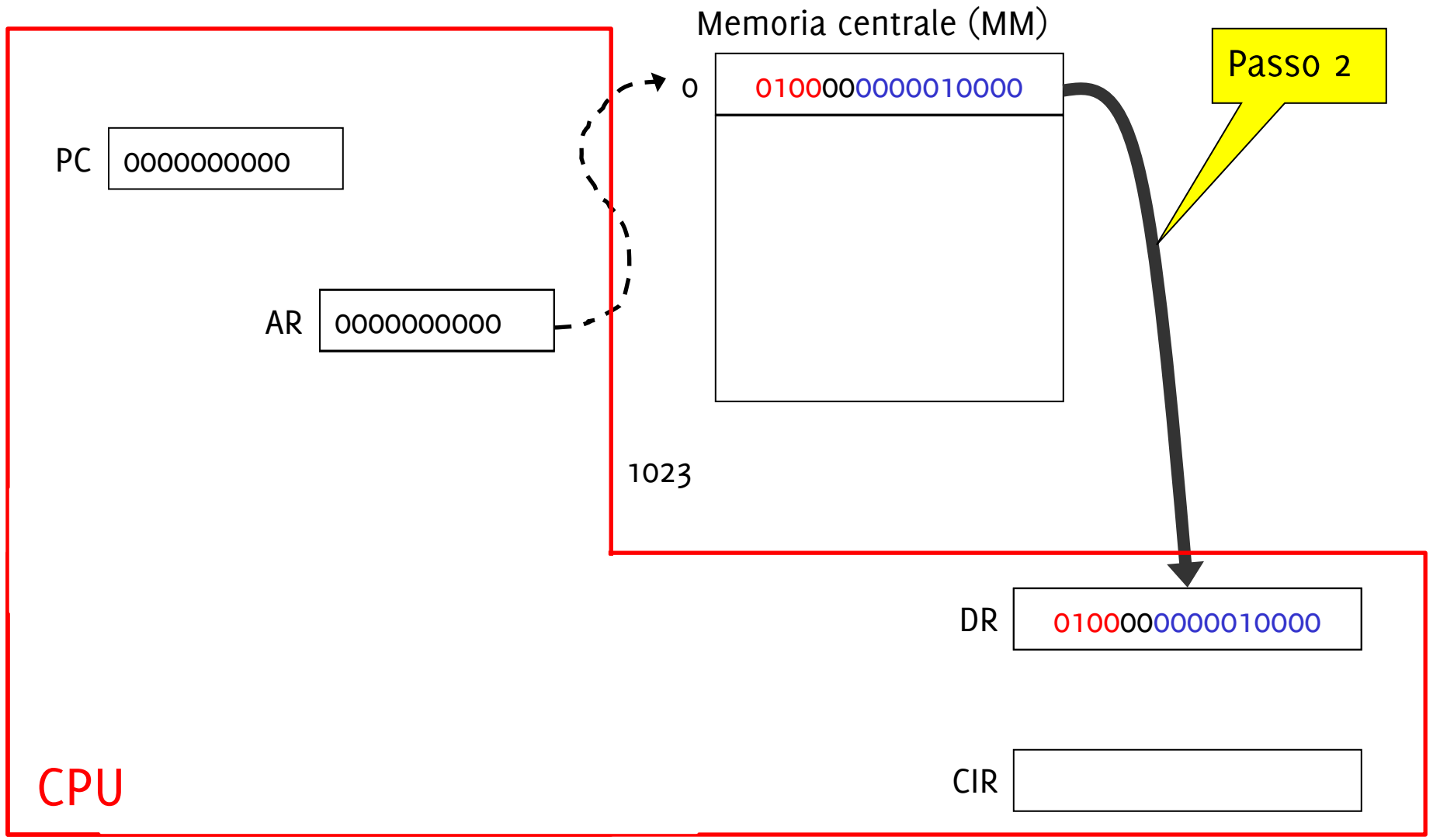
Fase di Fetch



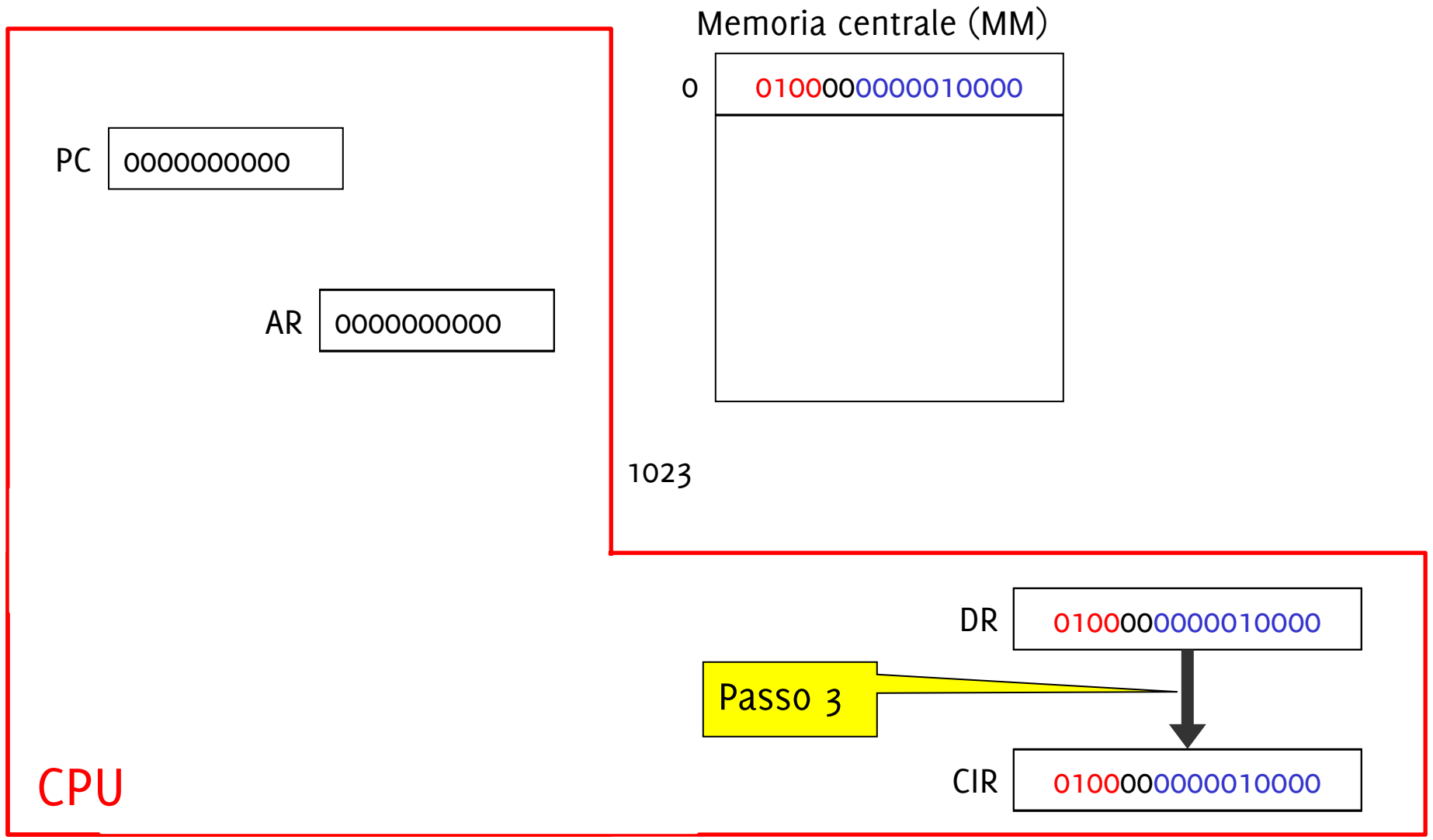
Fase di Fetch



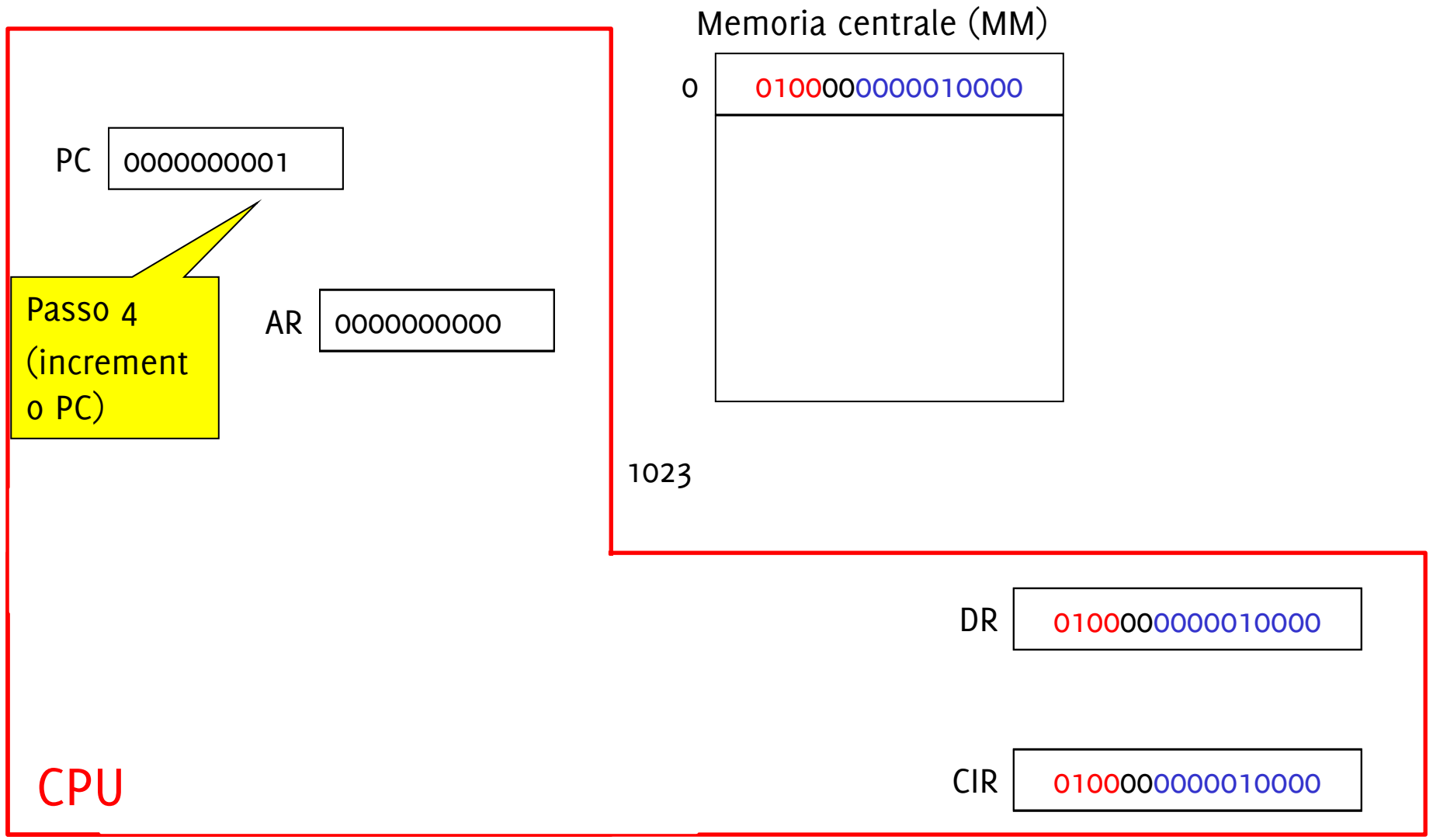
Fase di Fetch



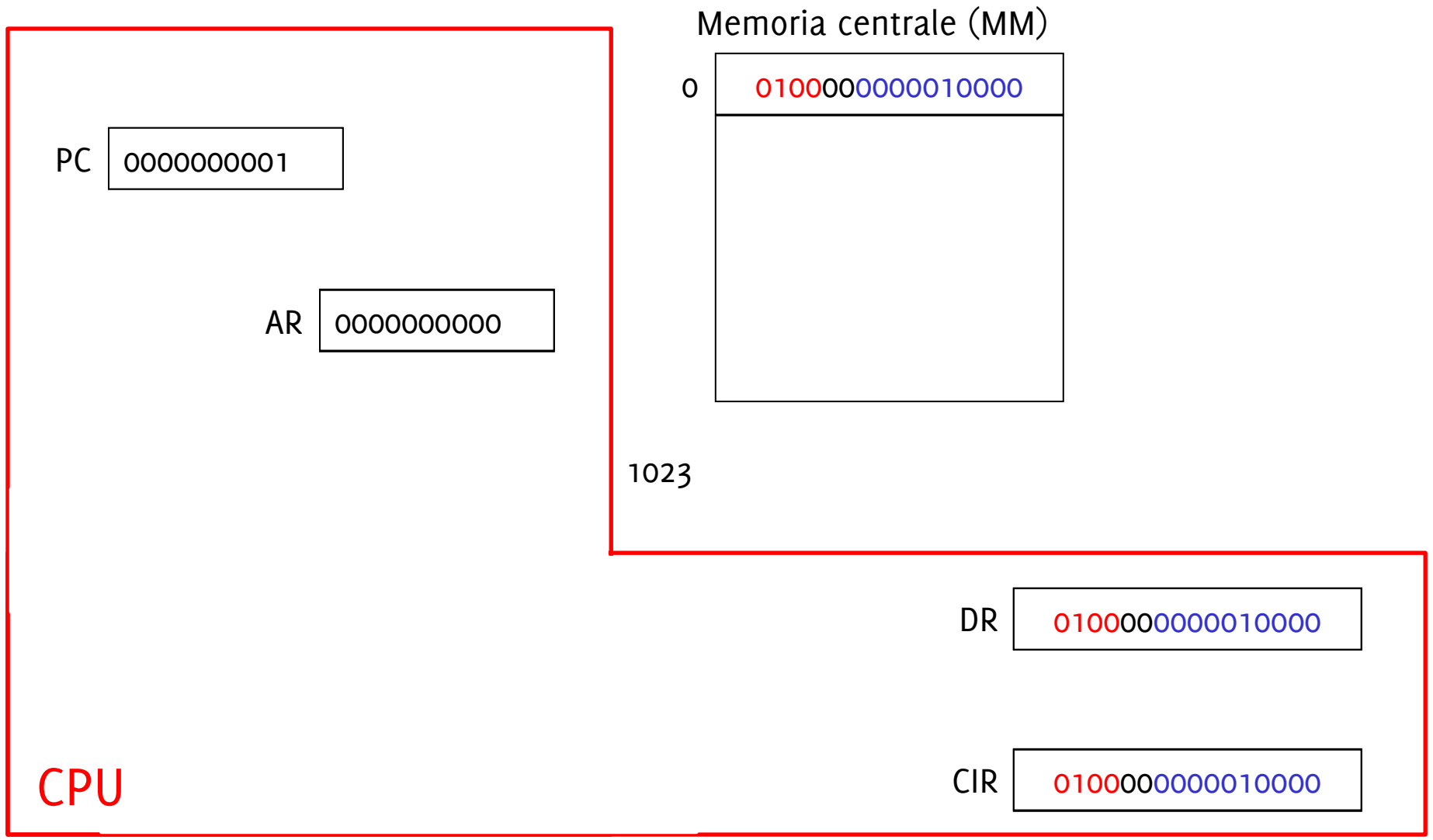
Fase di Fetch



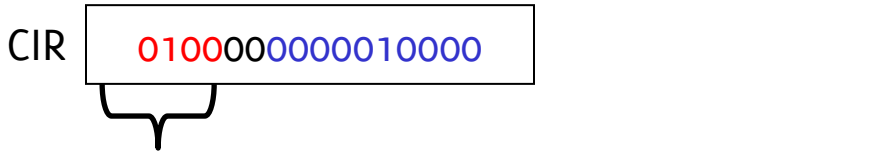
Fase di Fetch



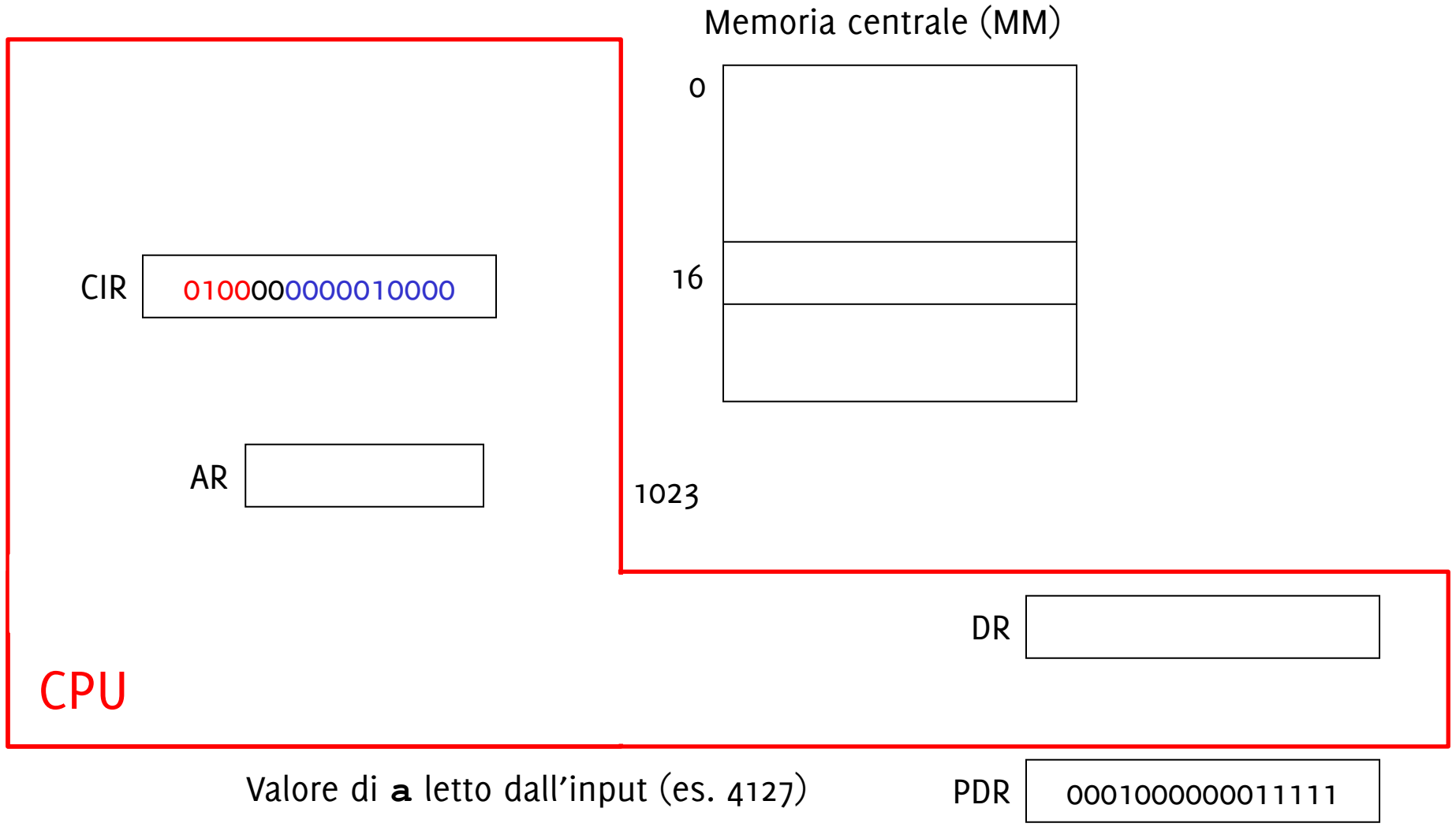
Fase di Fetch



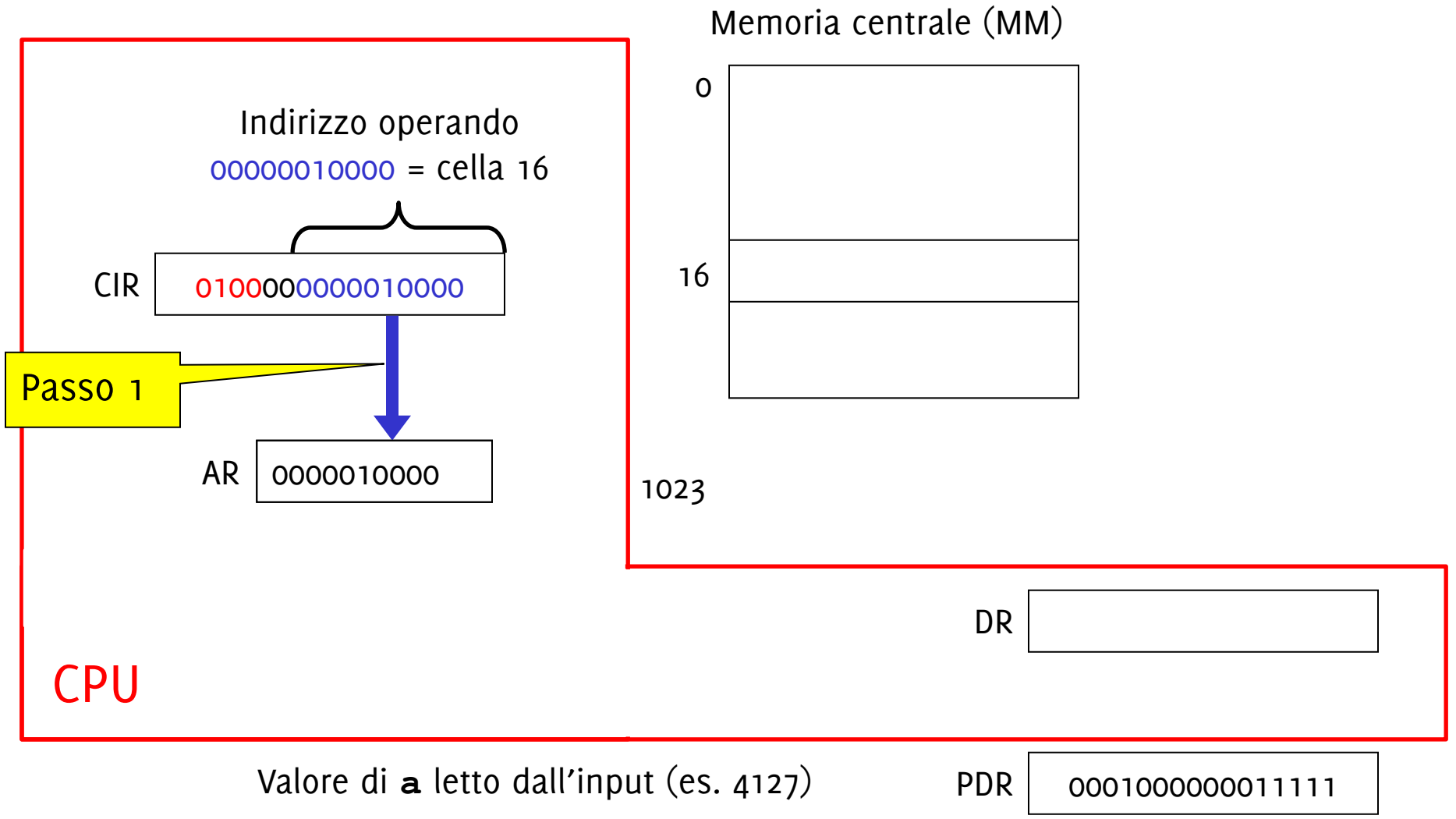
Fase di Decodifica 1ª istruzione



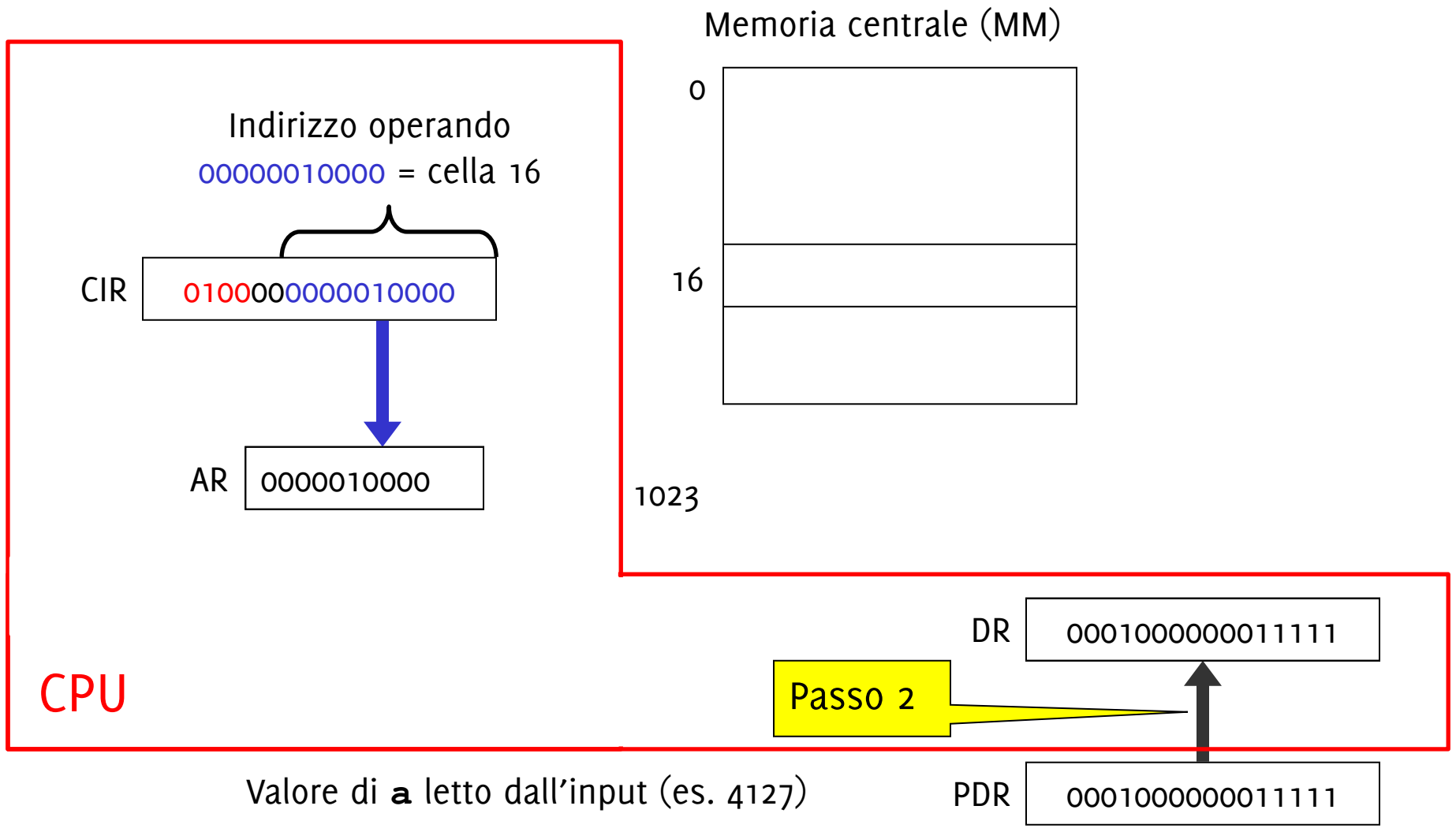
Fase di esecuzione: esempio lettura da input



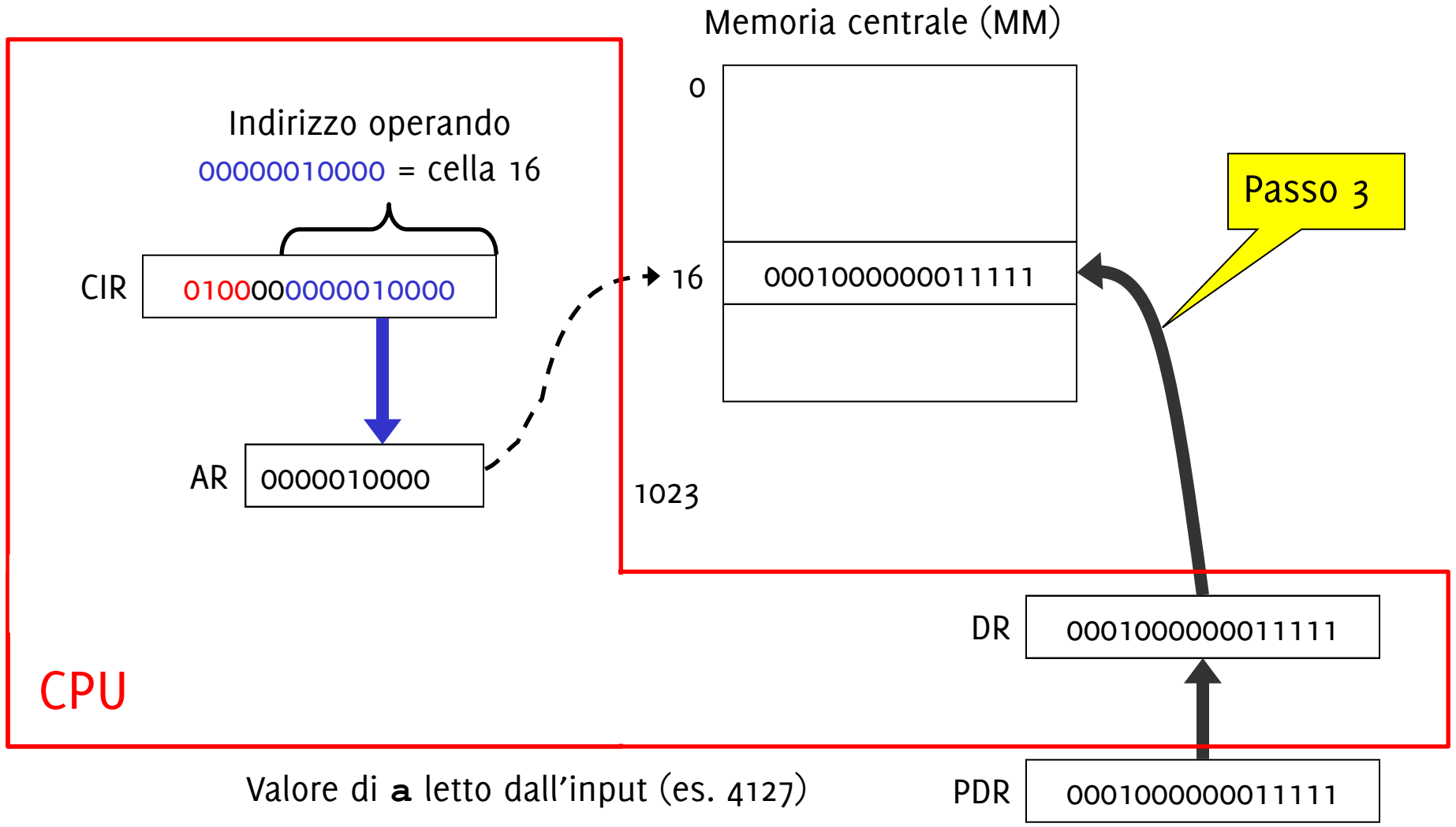
Fase di esecuzione: esempio lettura da input



Fase di esecuzione: esempio lettura da input



Fase di esecuzione: esempio lettura da input



Esempio

Si consideri una Macchina di Von Neumann con Parola a 16 bit, indirizzi a 10 bit e codice operativo 4 bit

Vogliamo calcolare il valore dell'espressione:

$$(a+b) \cdot (c+d)$$

leggendo i valori delle variabili **a**, **b**, **c**, **d** dal dispositivo di ingresso e scrivendo il risultato della valutazione sul dispositivo di uscita.

Esempio

1. Leggi dal dispositivo di ingresso il valore delle variabili **`a`**, **`b`**, **`c`**, **`d`**
2. Somma il valore di **`a`** al valore di **`b`**
3. Salva il risultato parziale ottenuto
4. Somma il valore di **`c`** al valore di **`d`**
5. Moltiplica il risultato parziale appena ottenuto con quello precedentemente salvato
6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva
7. Termina l'esecuzione del programma.

Esempio

1. Leggi dal dispositivo di ingresso il valore delle variabili **`a`**, **`b`**, **`c`**, **`d`**
2. Somma il valore di **`a`** al valore di **`b`**
3. Salva il risultato parziale ottenuto
4. Somma il valore di **`c`** al valore di **`d`**
5. Moltiplica il risultato parziale appena ottenuto con quello precedentemente salvato
6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva
7. Termina l'esecuzione del programma.

Il programma deve essere tradotto in opportuni codici operativi e indirizzi di celle di memoria!

Esempio: più nel dettaglio

1. Leggi dal dispositivo di ingresso il valore delle variabili **`a,b,c,d`**

1. Scrivi nella cella di memoria centrale riservata al valore di **`a`** il valore letto dal registro dati della periferica.
   - Occorre la posizione di **`a,b,c,d`**

Esempio: più nel dettaglio

1. Leggi dal dispositivo di ingresso il valore delle variabili **`a,b,c,d`**
2. Somma il valore di **`a`** al valore di **`b`**

1. Scrivi nella cella di memoria centrale riservata al valore di **`a`** il valore letto dal registro dati della periferica.
   - Occorre la posizione di **`a,b,c,d`**
2. Somma il valore di **`a`** al valore di **`b`**

   2.1 Copia il contenuto della cella di memoria riservata ad **`a`** nel registro A

   2.2 Copia il contenuto della cella di memoria riservata a **`b`** nel registro B

   2.3 Somma contenuto dei registri A e B

Esempio: più nel dettaglio

1. Leggi dal dispositivo di ingresso il valore delle variabili **a,b,c,d**
2. Somma il valore di **a** al valore di **b**
3. Salva il risultato parziale ottenuto

1. Scrivi nella cella di memoria centrale riservata al valore di **a** il valore letto dal registro dati della periferica.
   - Occorre la posizione di **a,b,c,d**
2. Somma il valore di **a** al valore di **b**
   2.1 Copia il contenuto della cella di memoria riservata ad **a** nel registro A
   2.2 Copia il contenuto della cella di memoria riservata a **b** nel registro B
   2.3 Somma contenuto dei registri A e B
3. Salva il risultato parziale, contenuto nel registro A, in una cella di memoria predisposta per il risultato (**z**).

Esempio: più nel dettaglio

1. Leggi dal dispositivo di ingresso il valore delle variabili **`a,b,c,d`**
2. Somma il valore di **`a`** al valore di **`b`**
3. Salva il risultato parziale ottenuto
4. Somma il valore di **`c`** al valore di **`d`**

4. Somma il valore di **`c`** al valore di **`d`**
   - 4.1 Copia il contenuto della cella di memoria riservata a **`c`** nel registro A
   - 4.2 Copia il contenuto della cella di memoria riservata a **`d`** nel registro B
   - 4.3 Somma il contenuto dei registri A e B

Esempio: più nel dettaglio

1. Leggi dal dispositivo di ingresso il valore delle variabili **a,b,c,d**
2. Somma il valore di **a** al valore di **b**
3. Salva il risultato parziale ottenuto
4. Somma il valore di **c** al valore di **d**
5. Moltiplica il risultato parziale appena ottenuto con quello precedentemente salvato

4. Somma il valore di **c** al valore di **d**
   - 4.1 Copia il contenuto della cella di memoria riservata a **c** nel registro A
   - 4.2 Copia il contenuto della cella di memoria riservata a **d** nel registro B
   - 4.3 Somma il contenuto dei registri A e B
5. Moltiplica
   - 5.1 Copia il contenuto della cella **z** nel registro B (**z** e B contengono ora **a + b**, mentre A contiene **c + d**)
   - 5.2 Moltiplica contenuto dei registri A e B

Esempio: più nel dettaglio

6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva

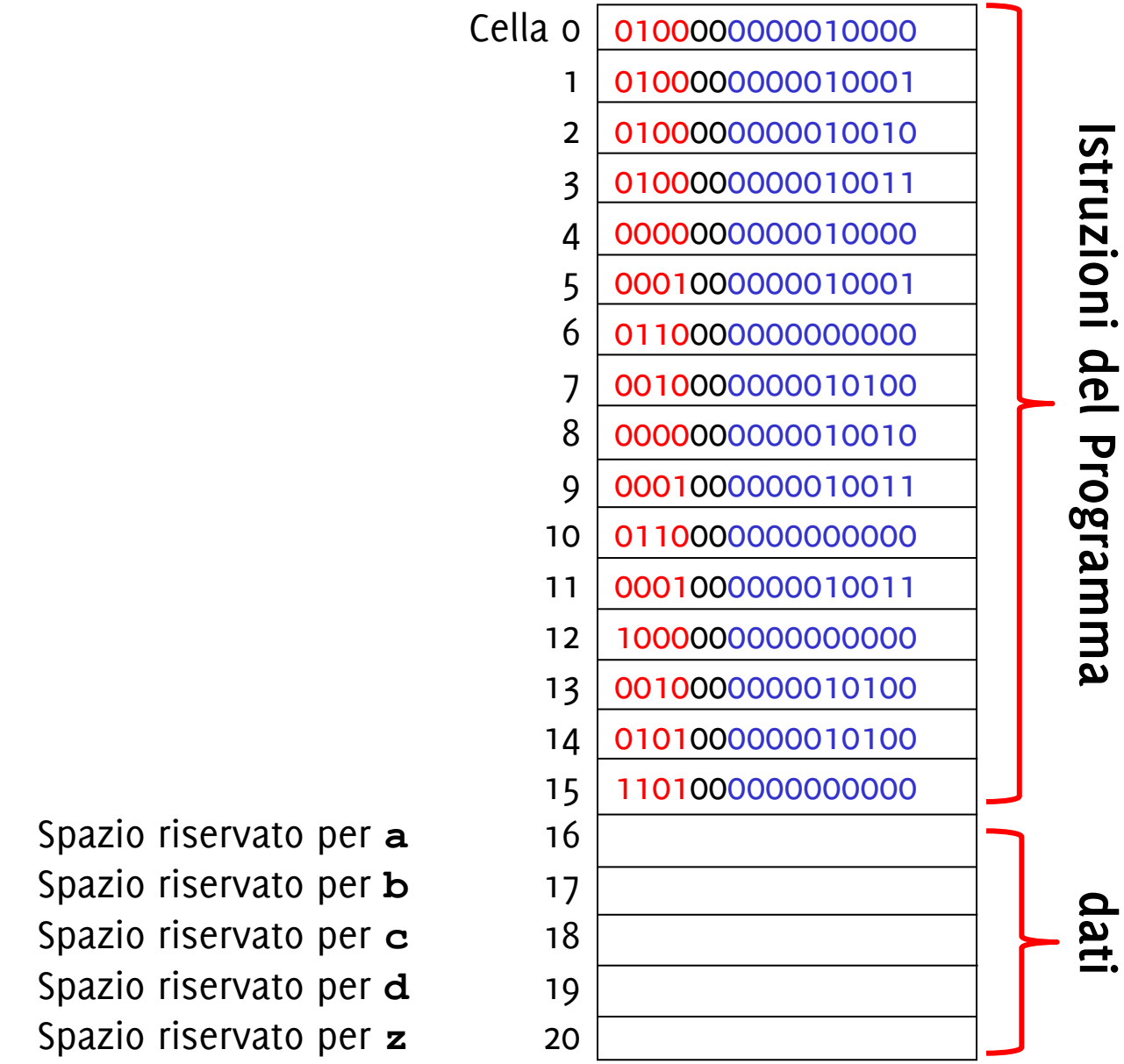
6. Scrivi sul dispositivo in uscita
   6.1 Memorizza il risultato calcolato (disponibile nel registro A) nella cella di memoria riservata a **z**
   6.2 Copia il contenuto della cella di memoria riservata a **z** nel registro dati della periferica di uscita

Esempio: più nel dettaglio

6. Scrivi sul dispositivo di uscita il risultato della valutazione complessiva
7. Termina l'esecuzione del programma

6. Scrivi sul dispositivo in uscita
   - 6.1 Memorizza il risultato calcolato (disponibile nel registro A) nella cella di memoria riservata a **z**
   - 6.2 Copia il contenuto della cella di memoria riservata a **z** nel registro dati della periferica di uscita
7. Manda il comando di Halt

Programma in Memoria Centrale



Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Forma Binaria del Programma

0100000000010000	**Leggi** un valore dall'input e mettilo nella cella 16 (**a**)
0100000000010001	**Leggi** un valore dall'input e mettilo nella cella 17 (**b**)
0100000000010010	**Leggi** un valore dall'input e mettilo nella cella 18 (**c**)
0100000000010011	**Leggi** un valore dall'input e mettilo nella cella 19 (**d**)
0000000000010000	**Carica** il contenuto della cella 16 (**a**) **nel registro A**
0001000000010001	**Carica** il contenuto della cella 17 (**b**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris.parziale)
0000000000010010	**Carica** il contenito della cella 18 (**c**) **nel registro A**
0001000000010011	**Carica** il contenito della cella 19 (**d**) **nel registro B**
0110000000000000	**Somma** i registri A e B
0001000000010100	**Carica** il contenuto della cella 20 (**z**) (ris. parziale) **in B**
1000000000000000	**Moltiplica** i registri A e B
0010000000010100	**Scarica** il contenuto di A nella cella 20 (**z**) (ris. totale)
0101000000010100	**Scrivi** il contenuto della cella 20 (**z**) (ris. totale) output
1101000000000000	**Halt**

Il programma in C

```
int main() {
      int x, y, z, w, ris;
      scanf("%d%d%d%d", &x, &y, &z, &w);
      ris = (x+y)-(z+w);
      printf("\n\n Il risultato e' : %d", ris);
      return 0;
}
```

Il Calcolatore

Il **calcolatore** è un potente **esecutore di algoritmi**

- ↑ È rapido: permette di gestire quantità di informazioni altrimenti intrattabili
- ↑ È preciso: non commette mai errori
- ↓ Non ha spirito critico

I **programmi** sono **algoritmi codificati** in **linguaggi** comprensibili dal calcolatore.