



- Il laboratorio di venerdì 2 Dicembre (domani) è da svolgere **in gruppi di 5 studenti**
- L'esercitazione di martedì 6 Dicembre è **spostata** a:
 - Lunedì 5 Dicembre ore 11:15 – 13:15 in aula Osvaldo De Donato



Comunicazioni di servizio

- Com'è fatto l'esame?
 - Vari esercizi (a volte 4/5, più spesso 6/7) su tutto il programma:
 - CP2, e/o IEEE 754
 - Array/Matrici
 - Strutture
 - File
 - Liste
 - Durata: fra 1h e 2h; spesso 1h e 40'
 - Esame da svolgere su carta
 - Potete scrivere in qualsiasi colore (anche a matita)...magari non in rosso
 - No libri/appunti/quaderni/calcolatrici/telefoni/computer
 - No orale



Architettura del calcolatore

Fondamenti di informatica, AA 2022/23

Luca Cassano

luca.cassano@polimi.it



Architettura di un Sistema Informatico



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»]



Un Sistema Informatico

- Hardware
- Software



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, CAD... i programmi che scriveremo)
- Consideriamo inizialmente l'hardware dei sistemi informatici.



La Macchina di Von Neumann

Un modello dell'architettura dei calcolatori





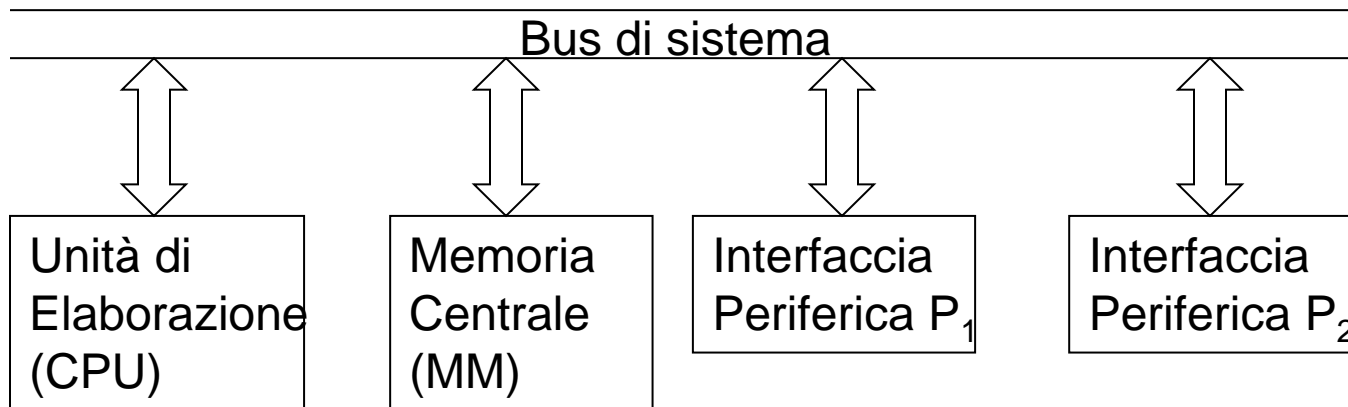
La Macchina di Von Neumann

- 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



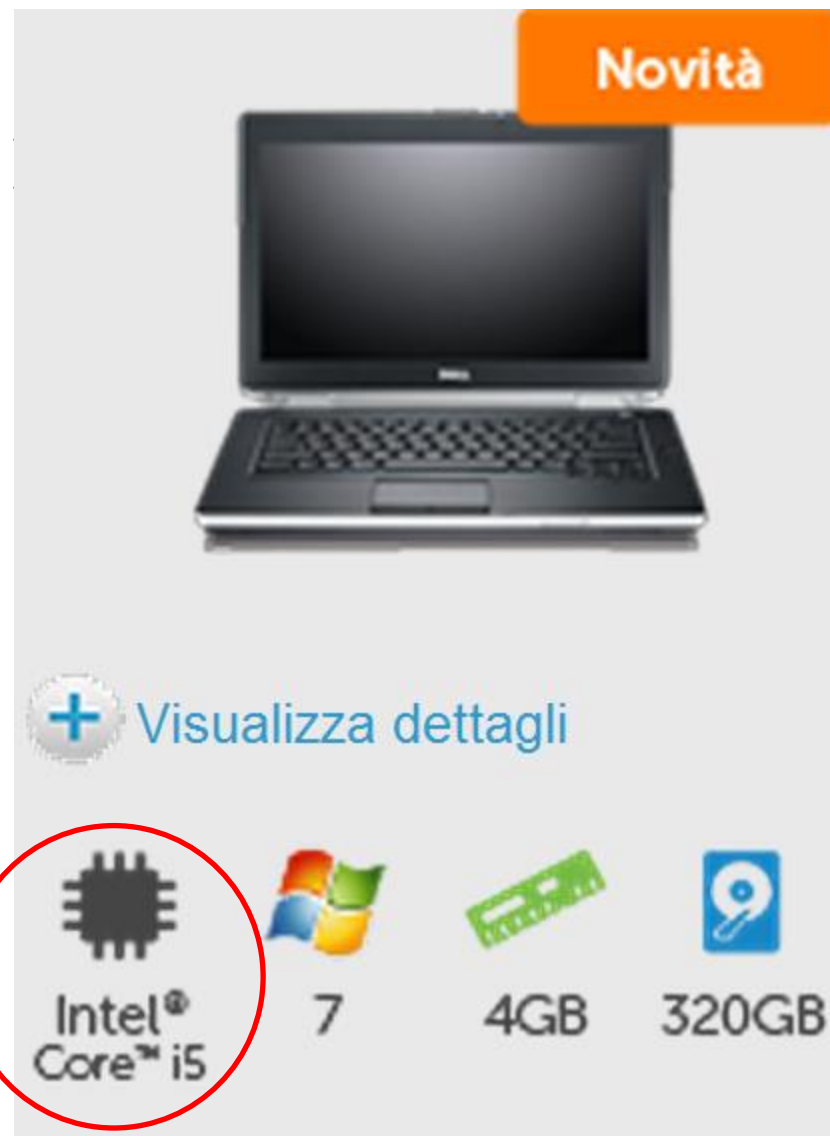
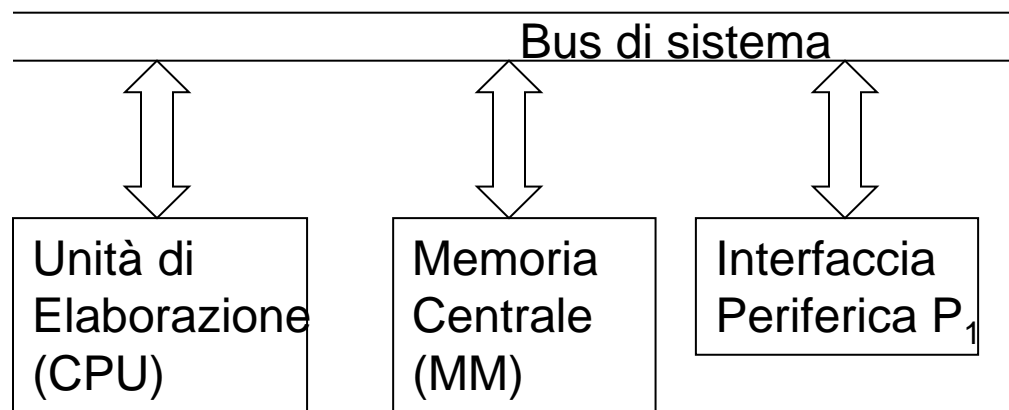
La Macchina di Von Neumann

- 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



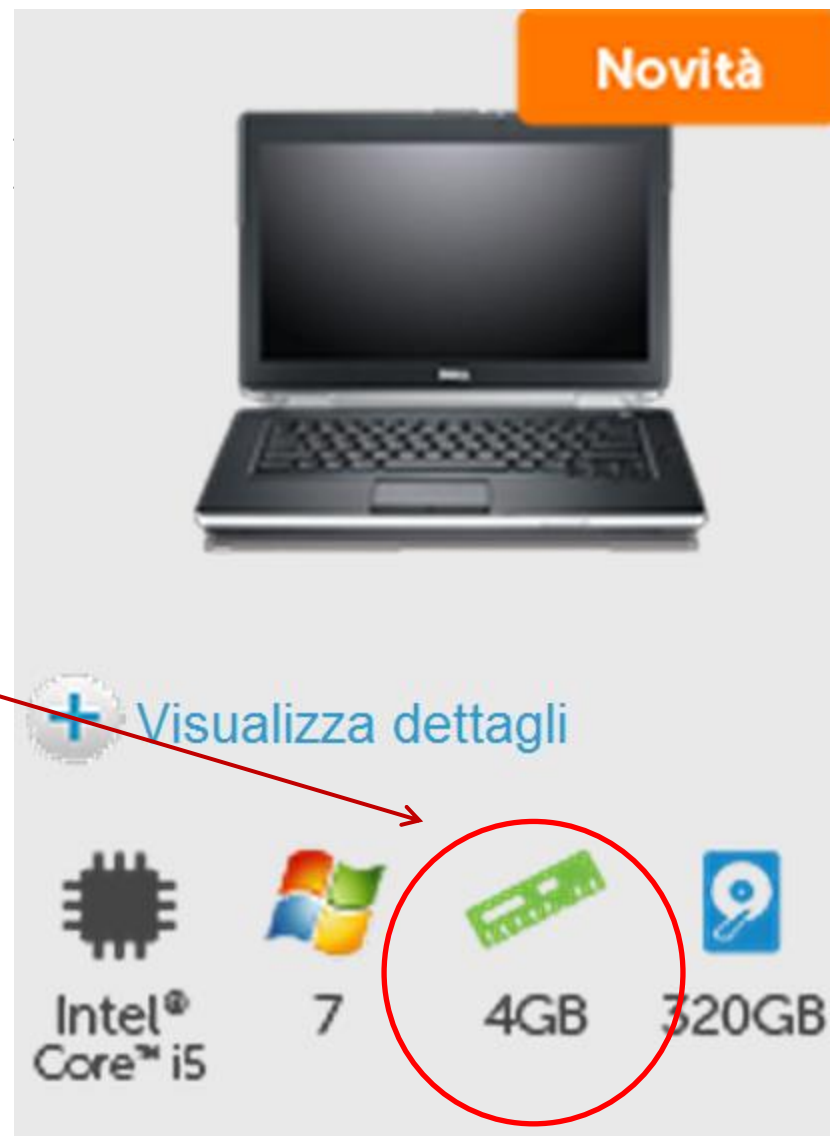
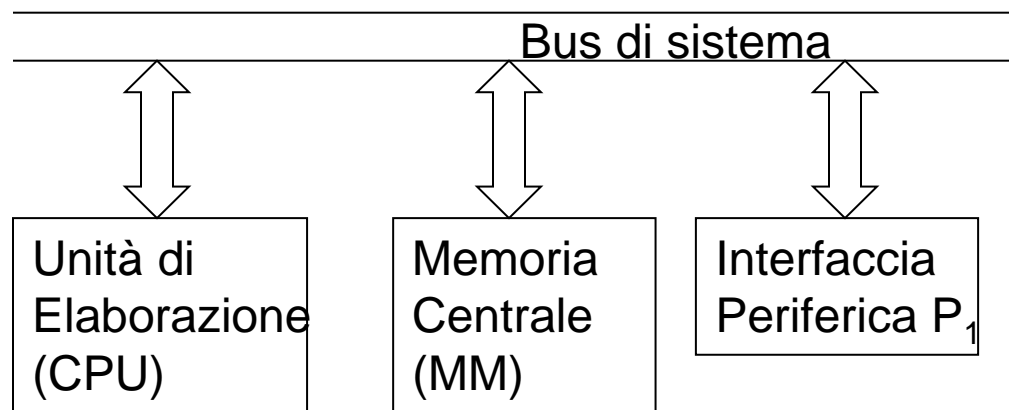


La Macchina di Von Neumann



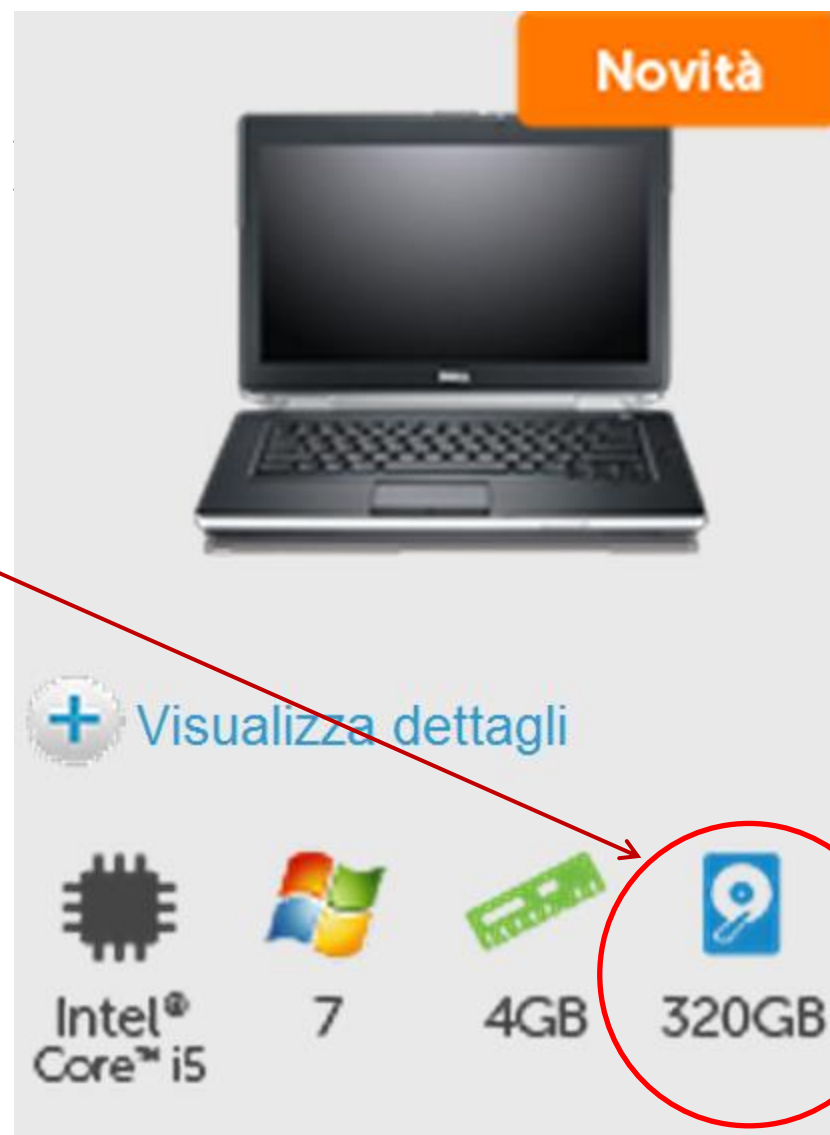
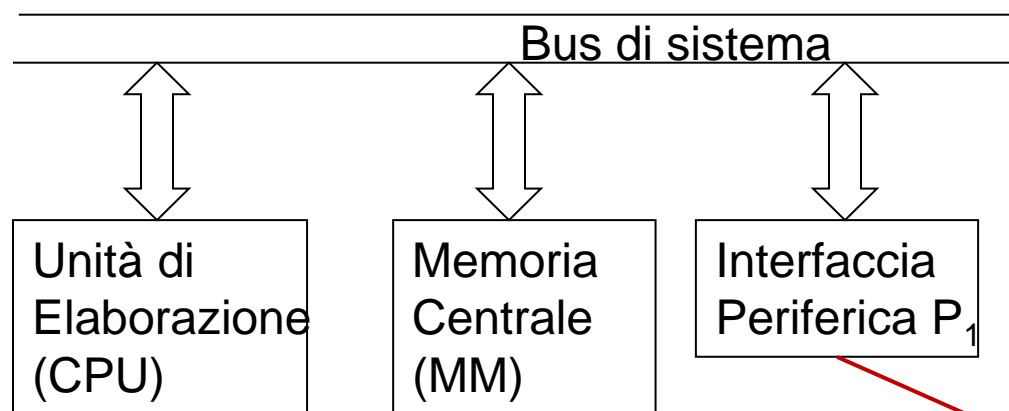


La Macchina di Von Neumann



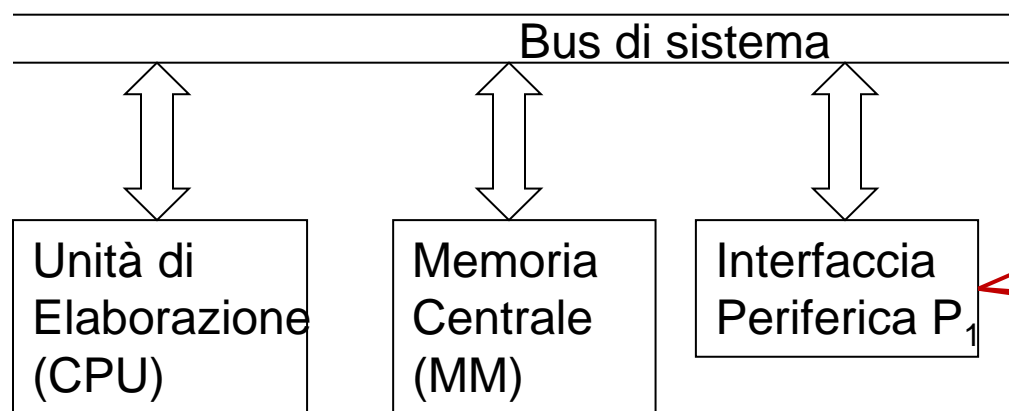


La Macchina di Von Neumann





La Macchina di Von Neumann



Novità

Visualizza dettagli

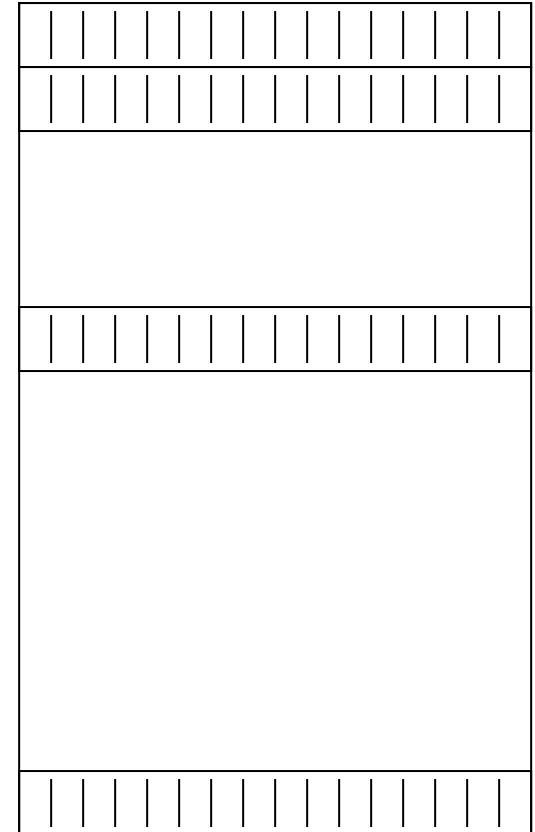
Intel® Core™ i5 7 4GB 320GB

The screenshot shows a Dell laptop with a black screen and keyboard. Three red arrows point from the 'Interfaccia Periferica P_1 ' box in the Von Neumann diagram to the laptop's screen, keyboard, and touchpad. Below the laptop, there is a section with icons and specifications: a CPU icon for 'Intel® Core™ i5', a Windows logo for '7', a RAM icon for '4GB', and a hard drive icon for '320GB'.



La Memoria Centrale (MM)

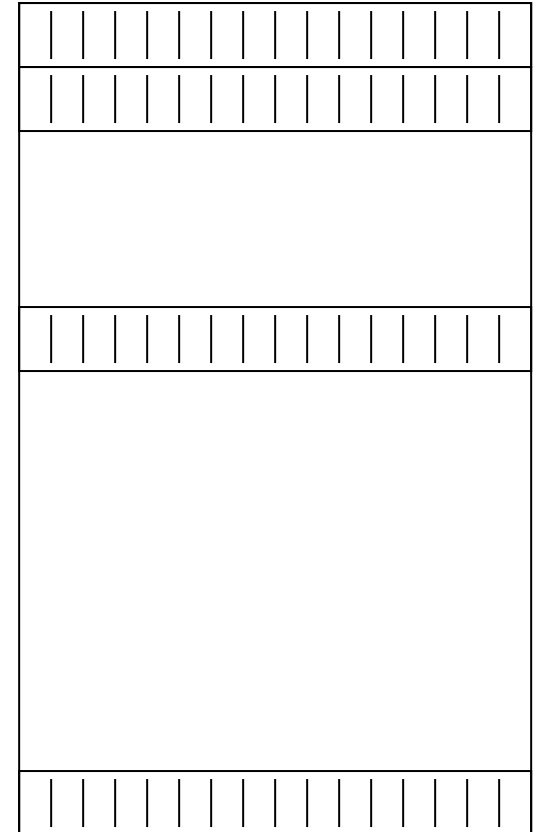
- Contiene i **programmi** (**sequenza di istruzioni**) in **esecuzione** ed i relativi **dati**





La Memoria Centrale (MM)

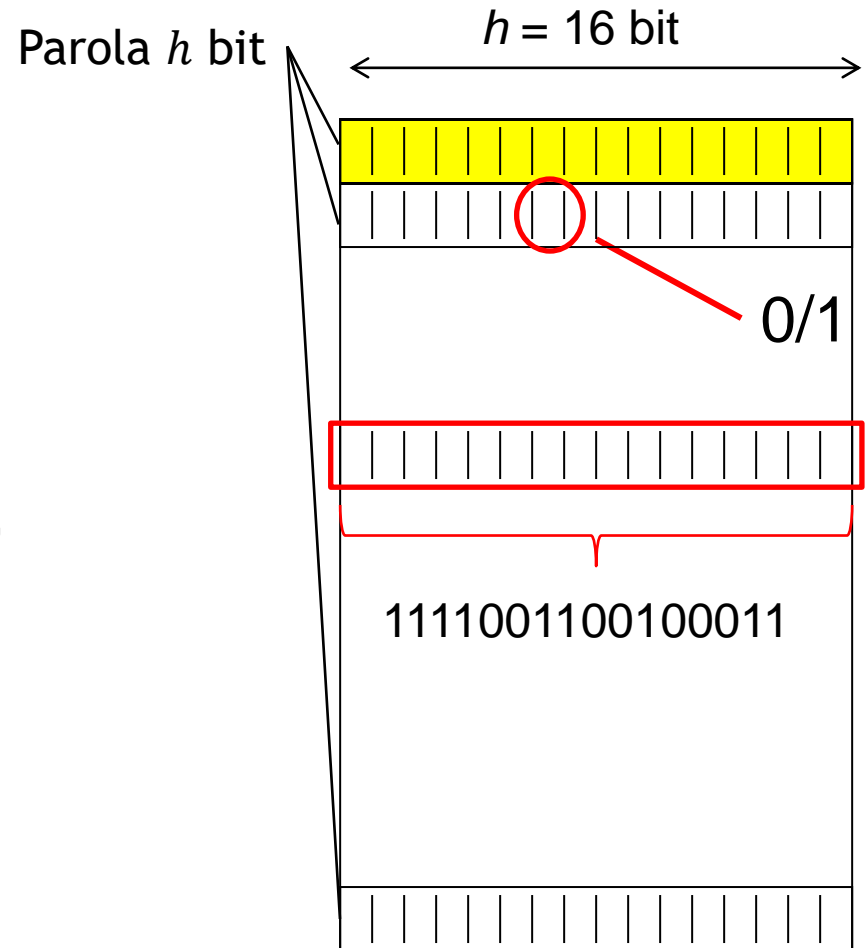
- Contiene i **programmi (sequenza di istruzioni)** in **esecuzione** ed i relativi **dati**
- È schematizzata come una sequenza di celle.





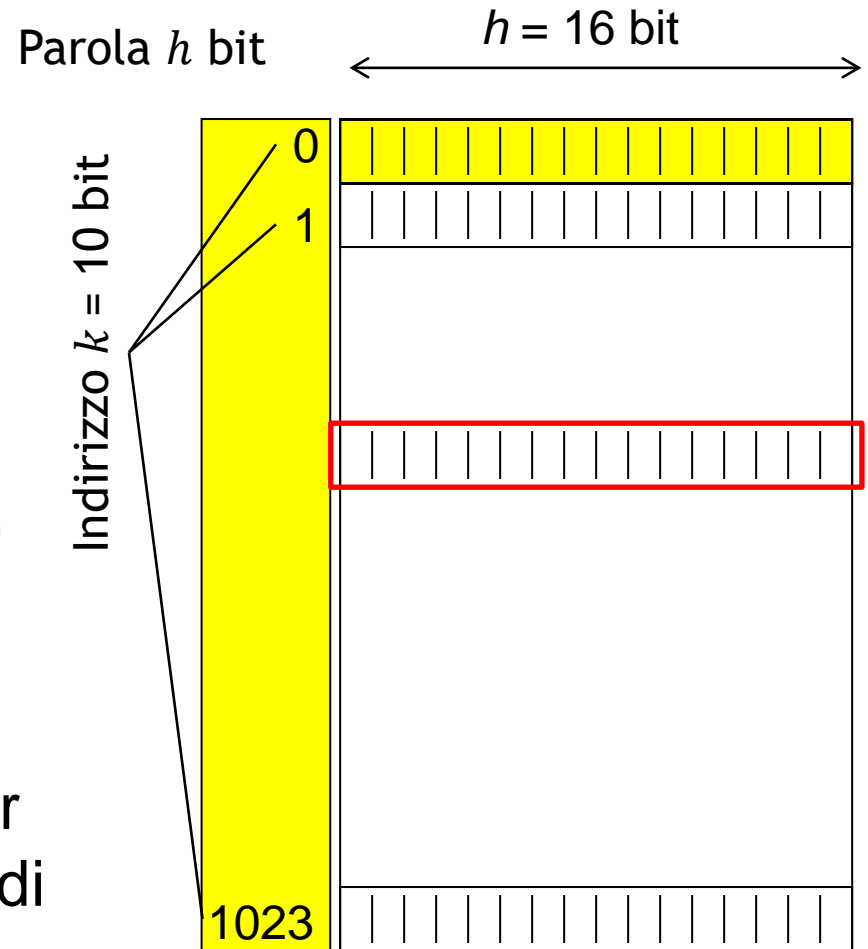
La Memoria Centrale (MM)

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





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.
 - ROM (*Read Only Memory*) memoria permanente.
 - EPROM (*Erasable Programmable ROM*)
riprogrammabile
- **L'HD è memoria permanente ma non è memoria centrale** ed in riferimento alla macchina di Von Neumann è una periferica.

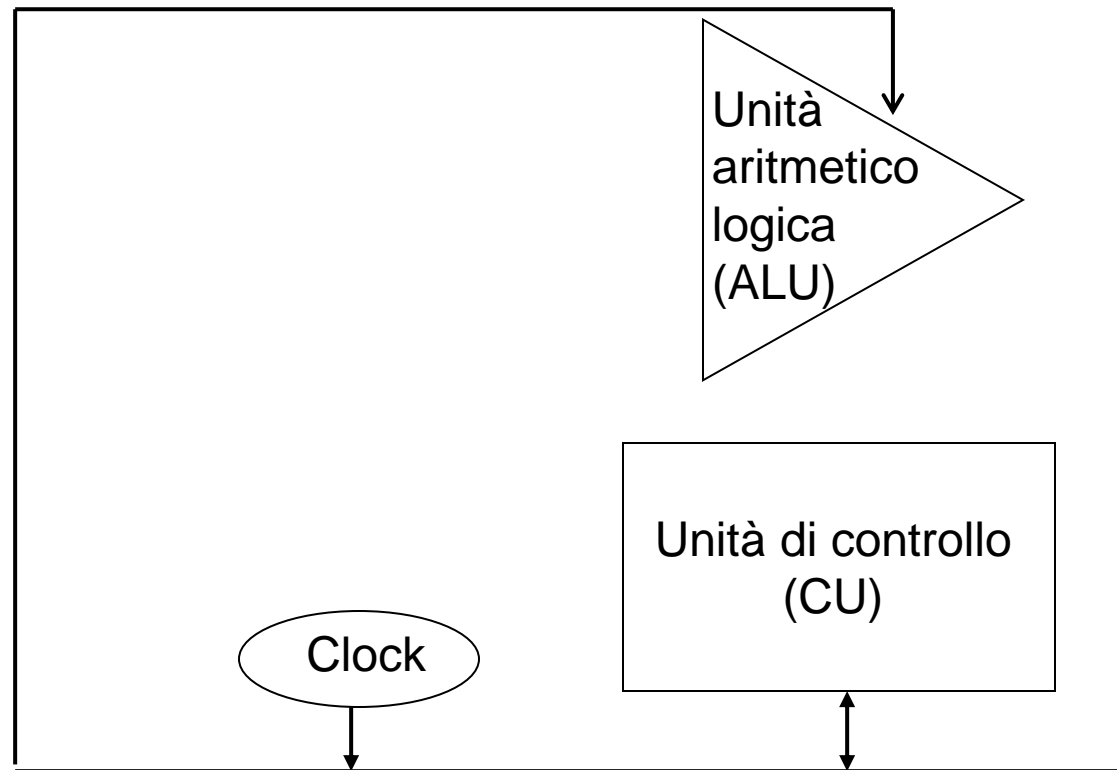


L'Unità di Elaborazione (CPU)

- 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**,
 - L'**Unità Aritmetico Logica**

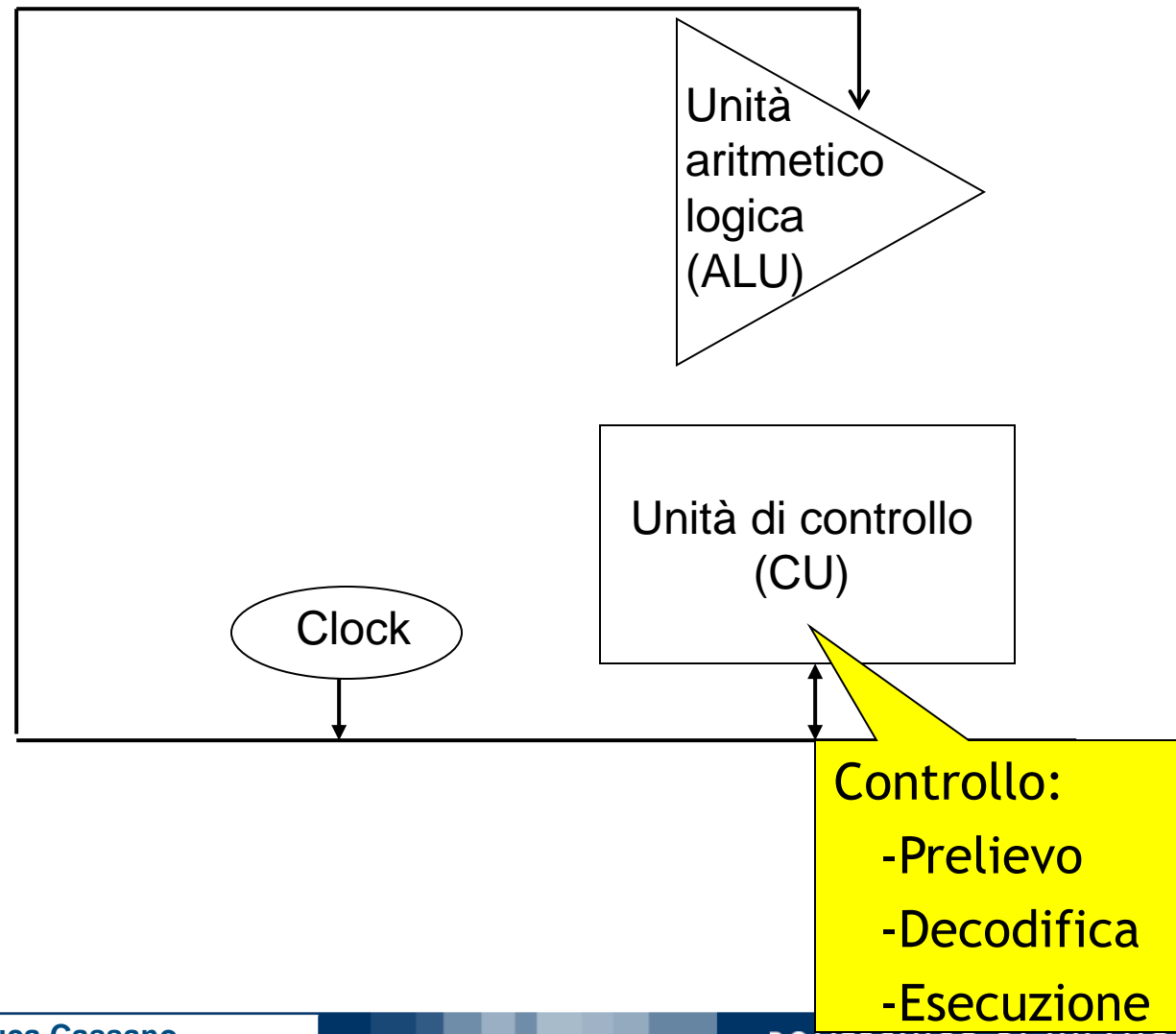


L'Unità di Elaborazione (CPU)





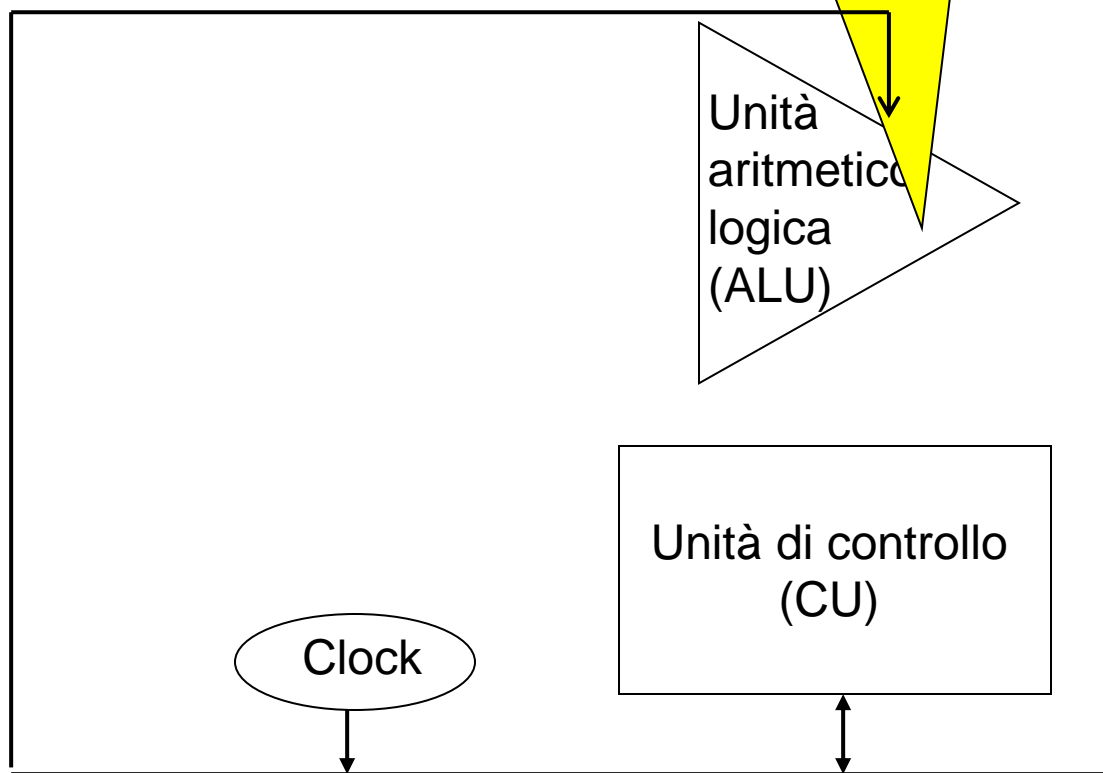
L'Unità di Elaborazione (CPU)





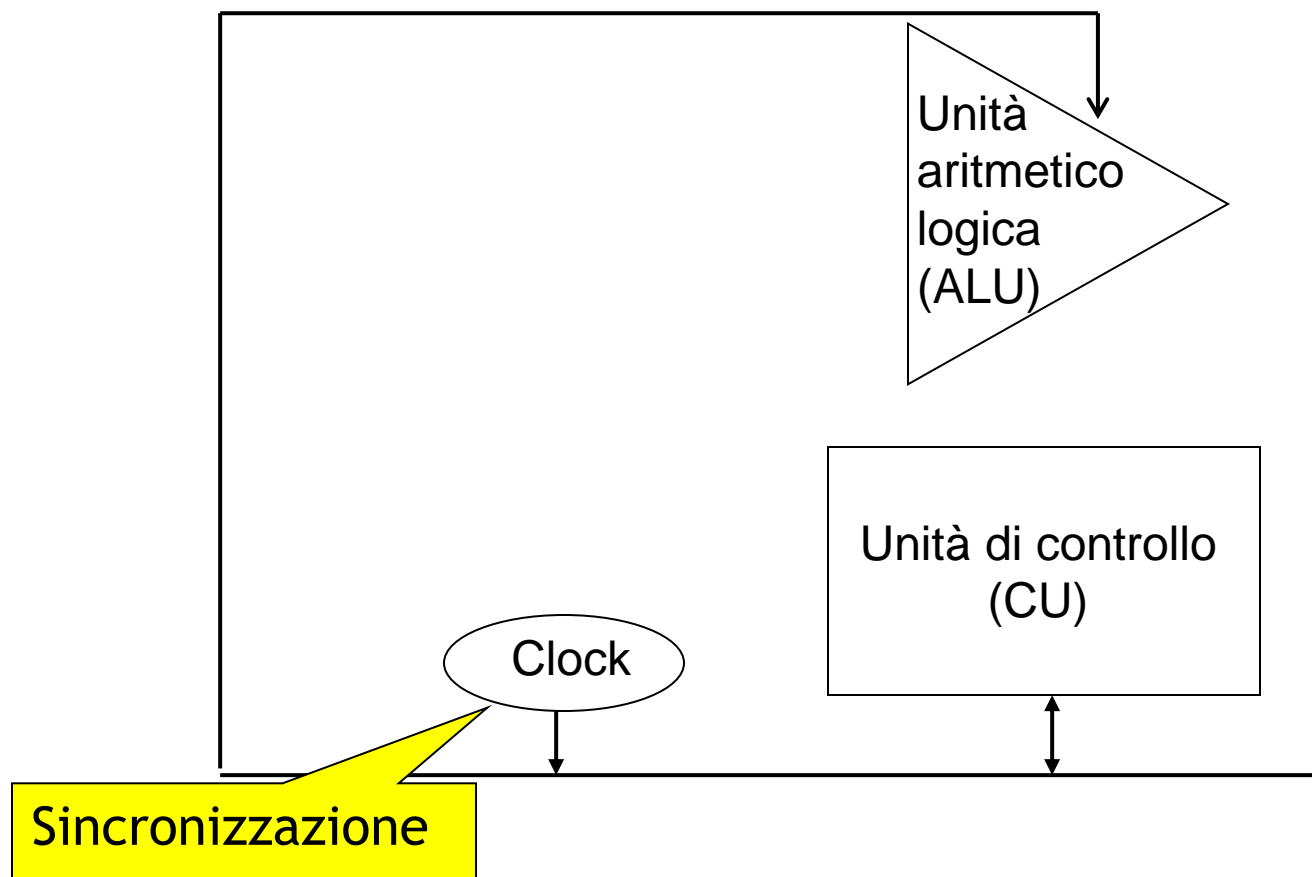
L'Unità di Elaborazione (CPU)

Operazioni
aritmetiche
e logiche





L'Unità di Elaborazione (CPU)



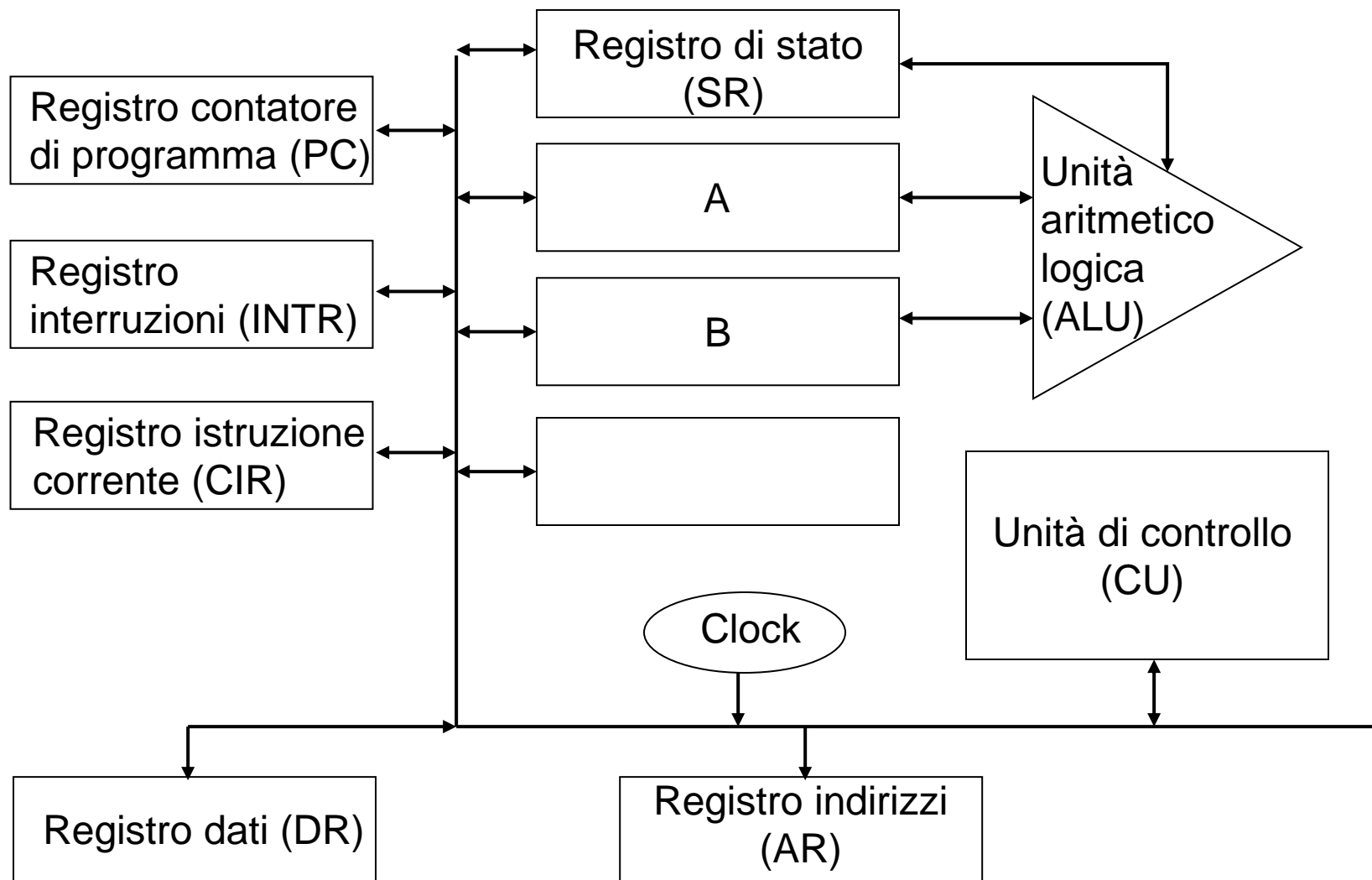


L'Unità di Elaborazione (CPU)

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

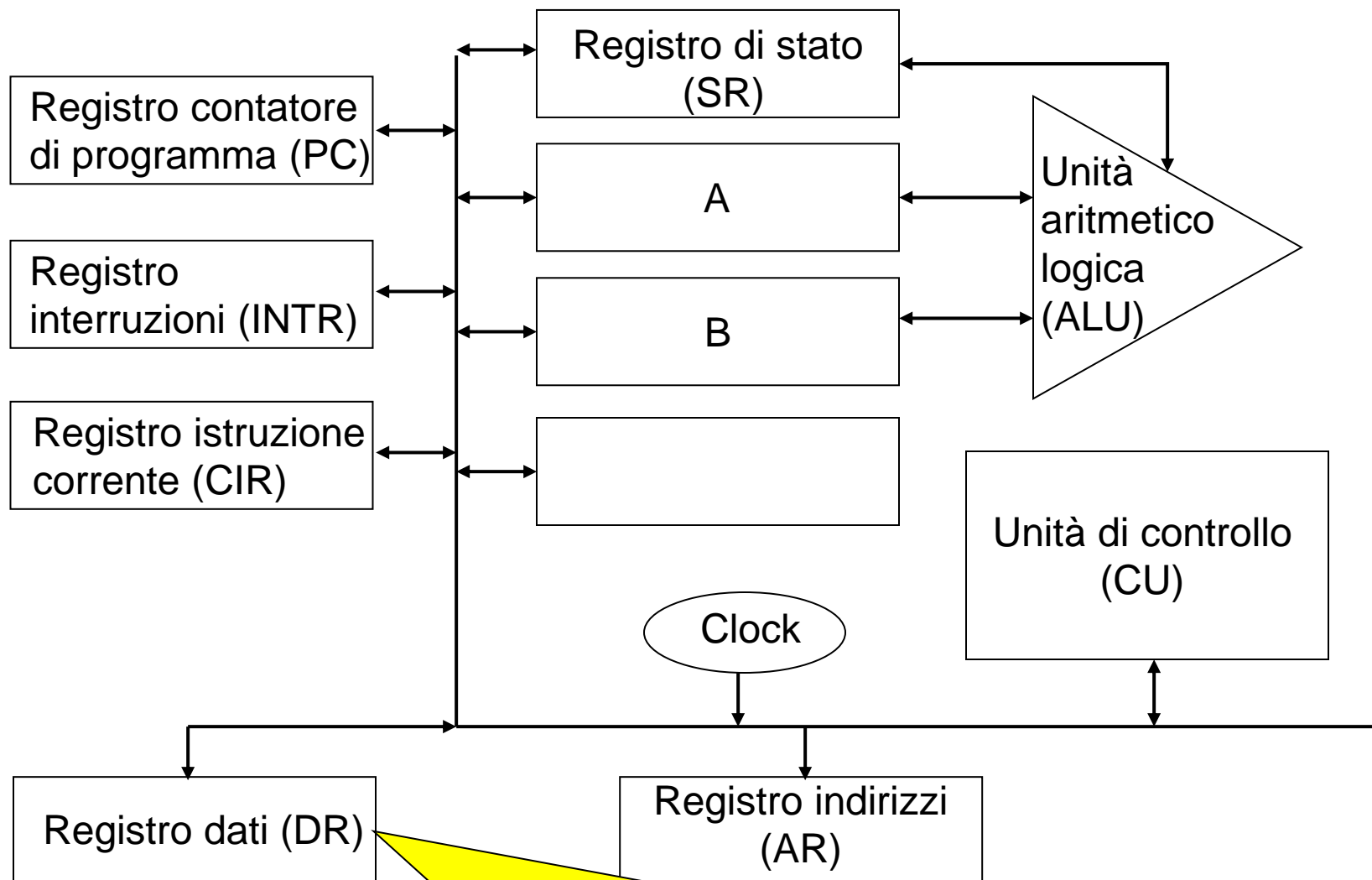


L'Unità di Elaborazione (CPU)





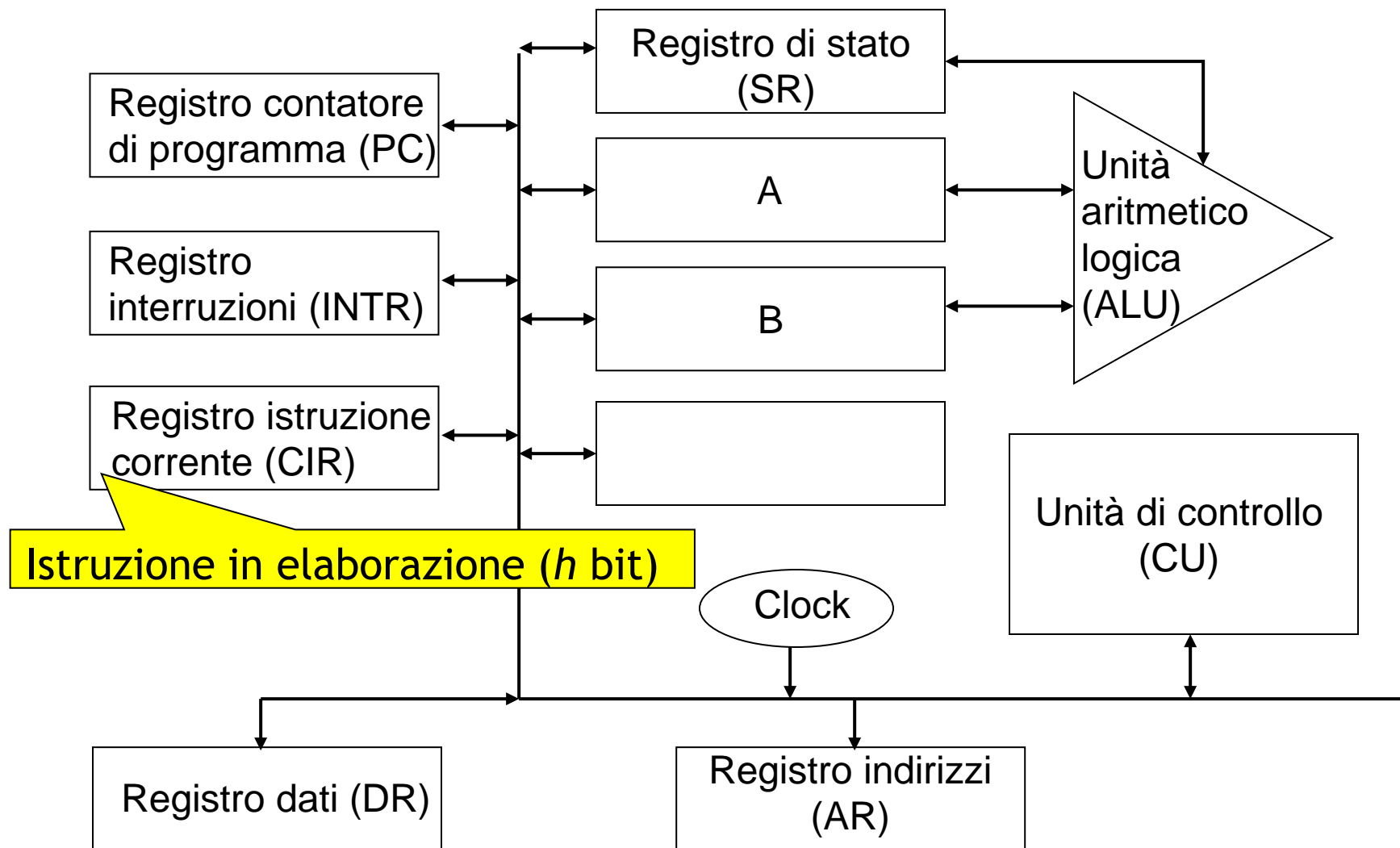
L'Unità di Elaborazione (CPU)



Parola letta/da scrivere in MM (*h* bit)

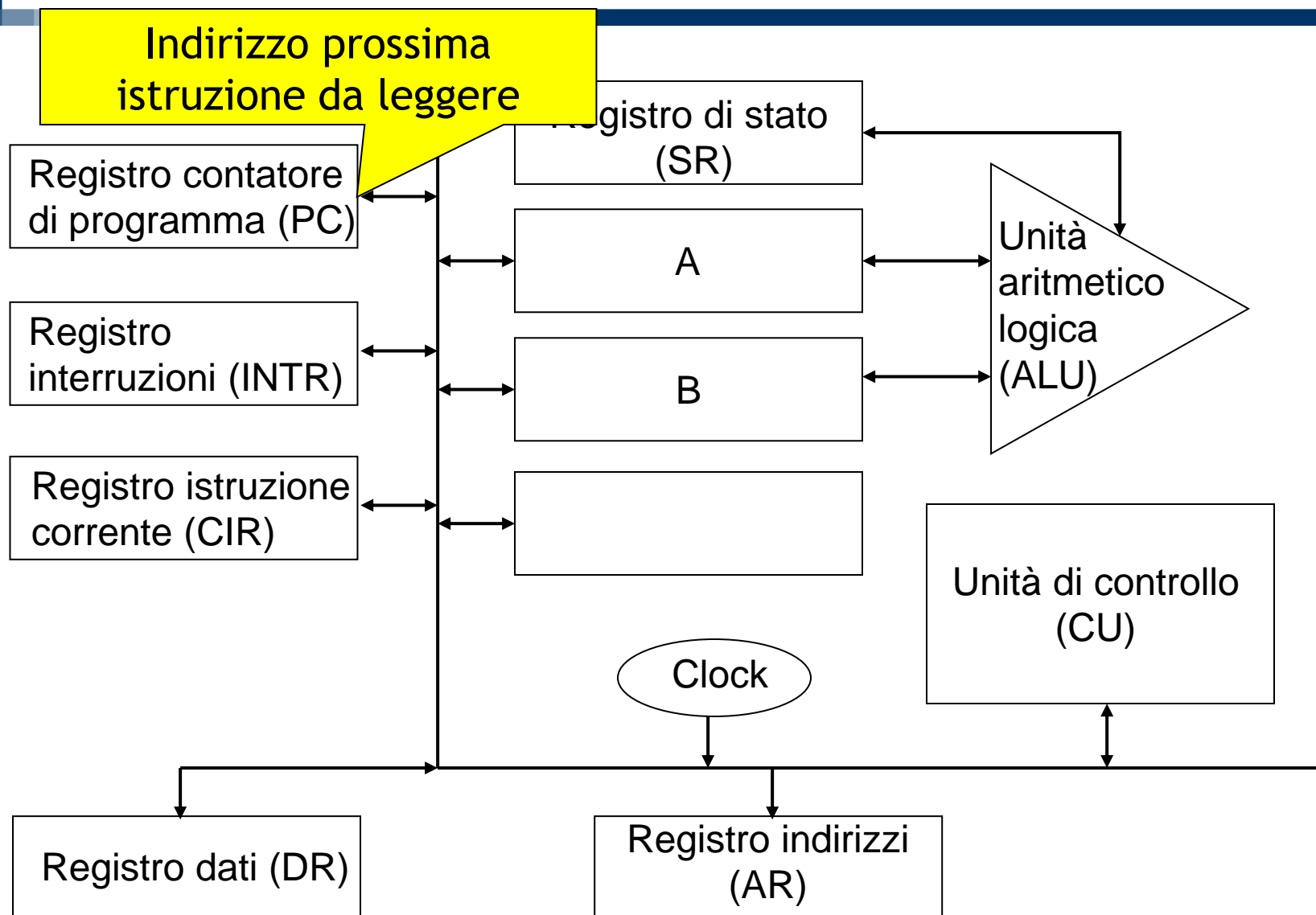


L'Unità di Elaborazione (CPU)



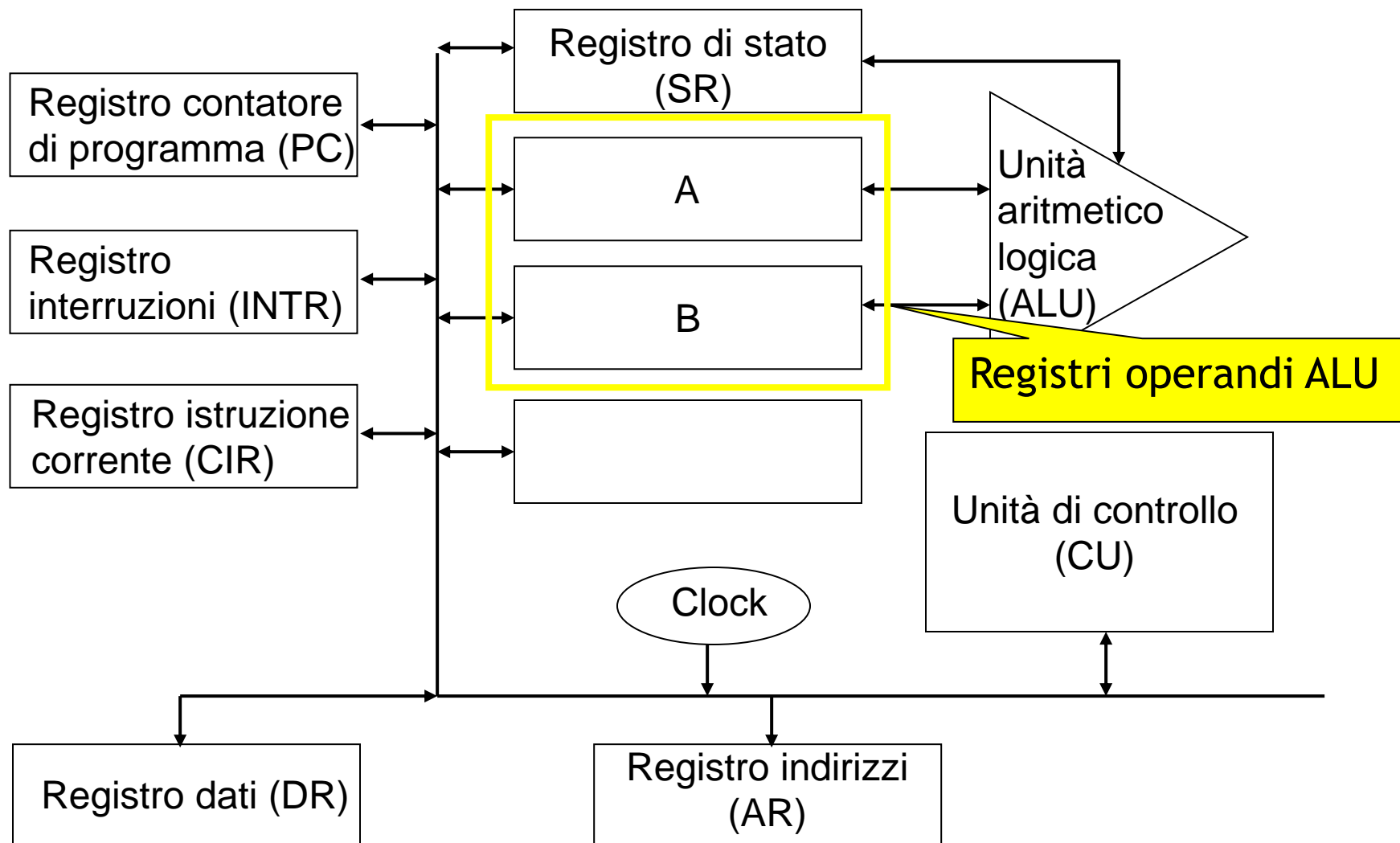


L'Unità di Elaborazione (CPU)



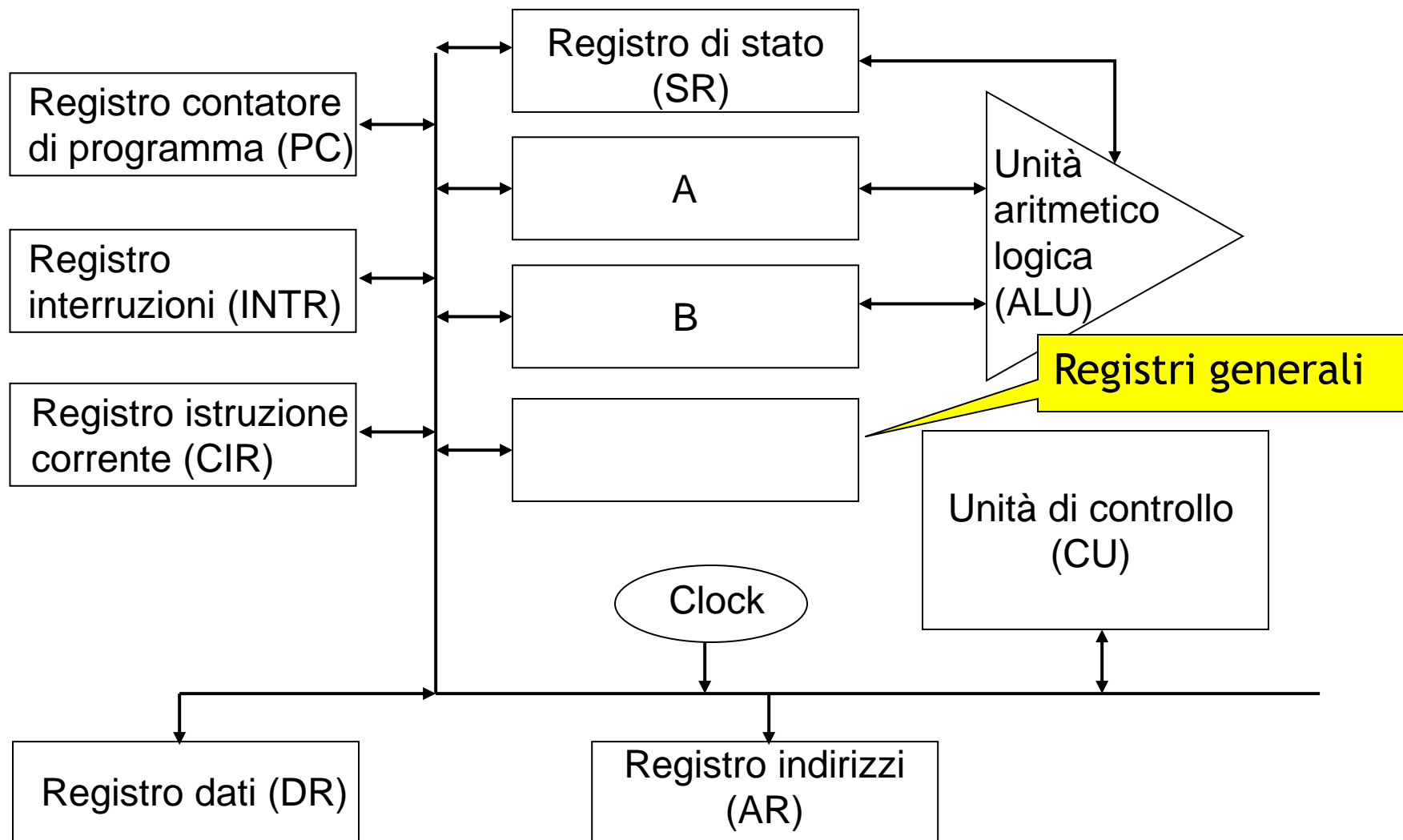


L'Unità di Elaborazione (CPU)



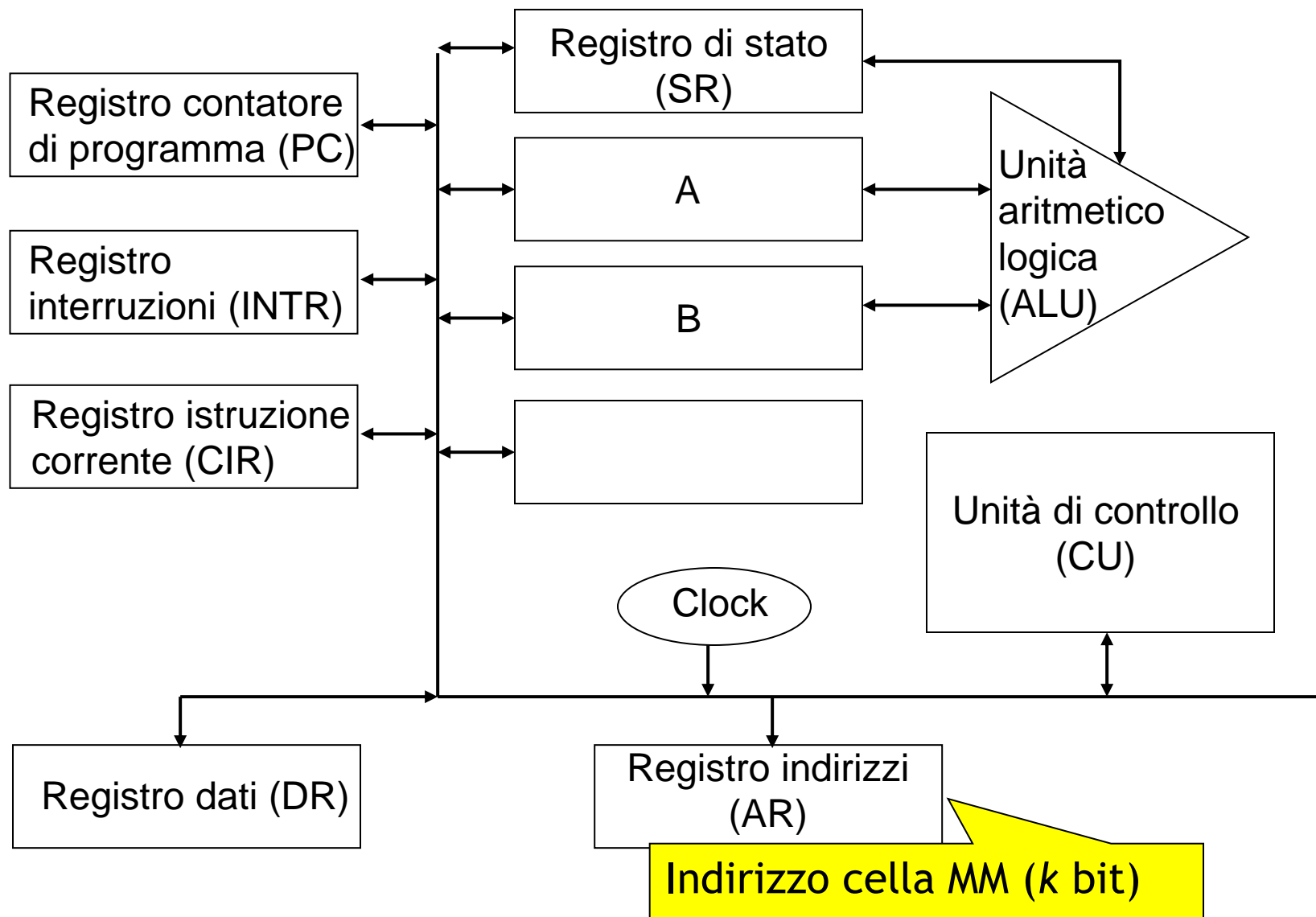


L'Unità di Elaborazione (CPU)



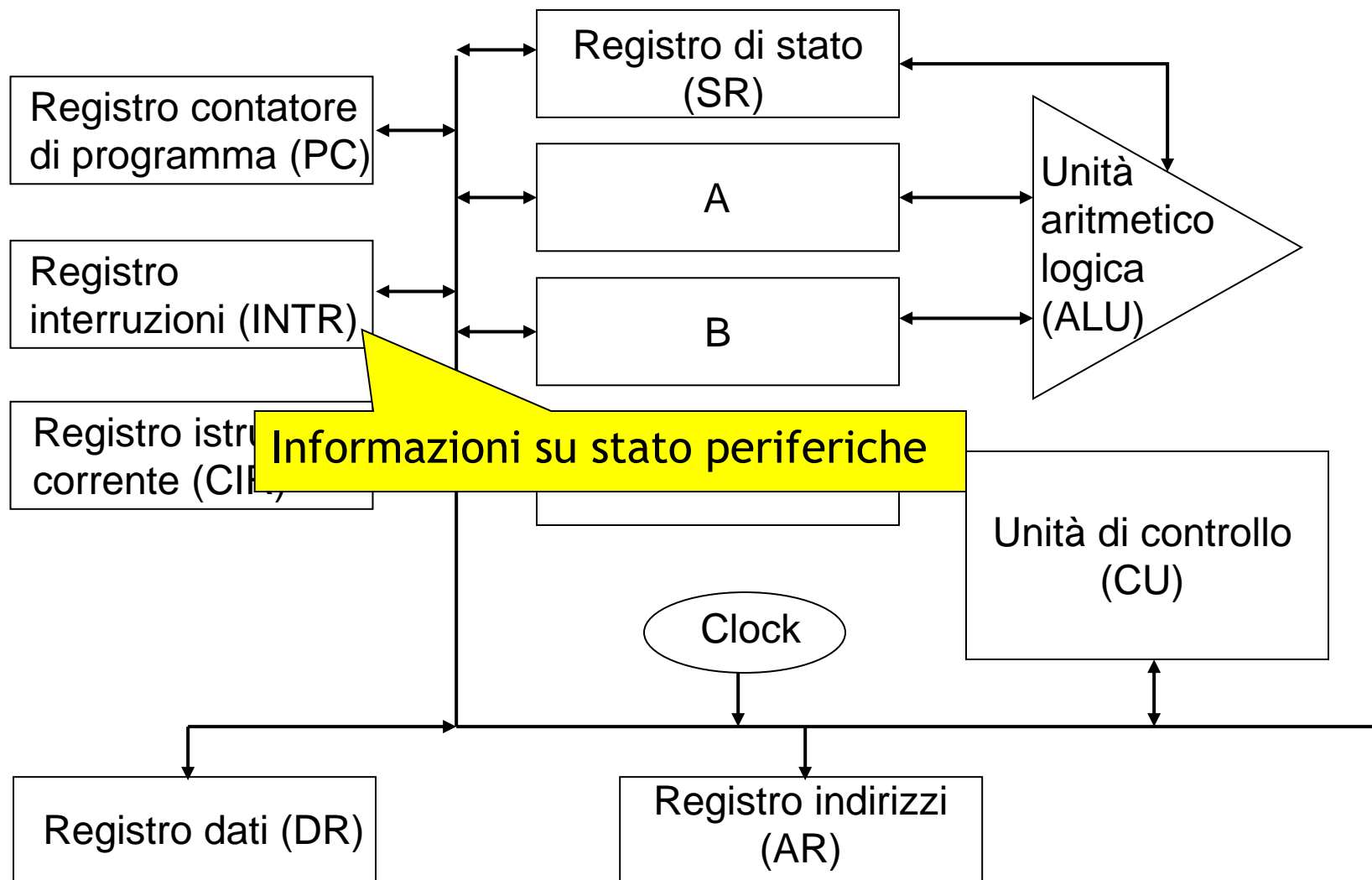


L'Unità di Elaborazione (CPU)





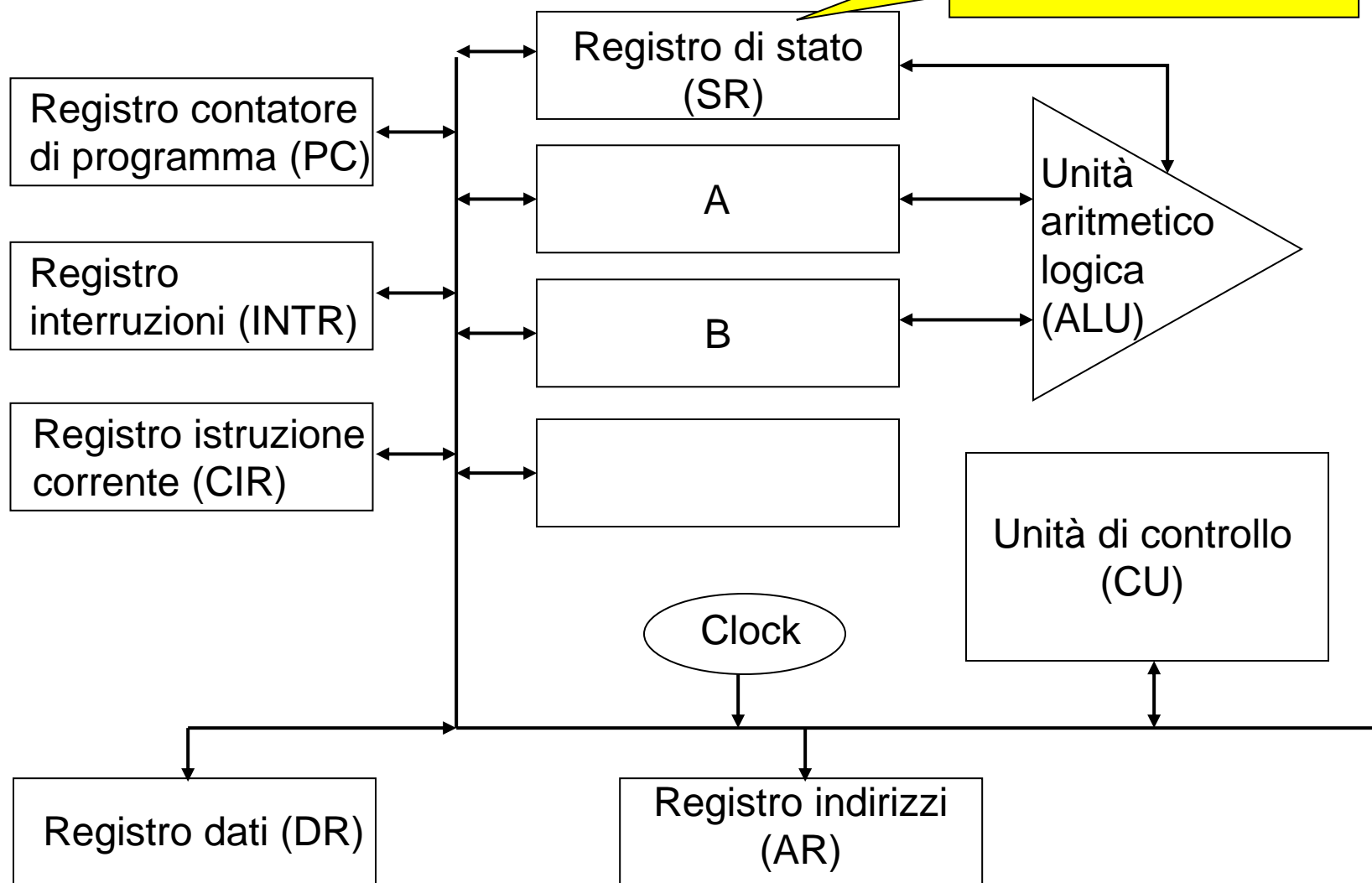
L'Unità di Elaborazione (CPU)





L'Unità di Elaborazione (CPU)

Informazioni sui
risultati ALU



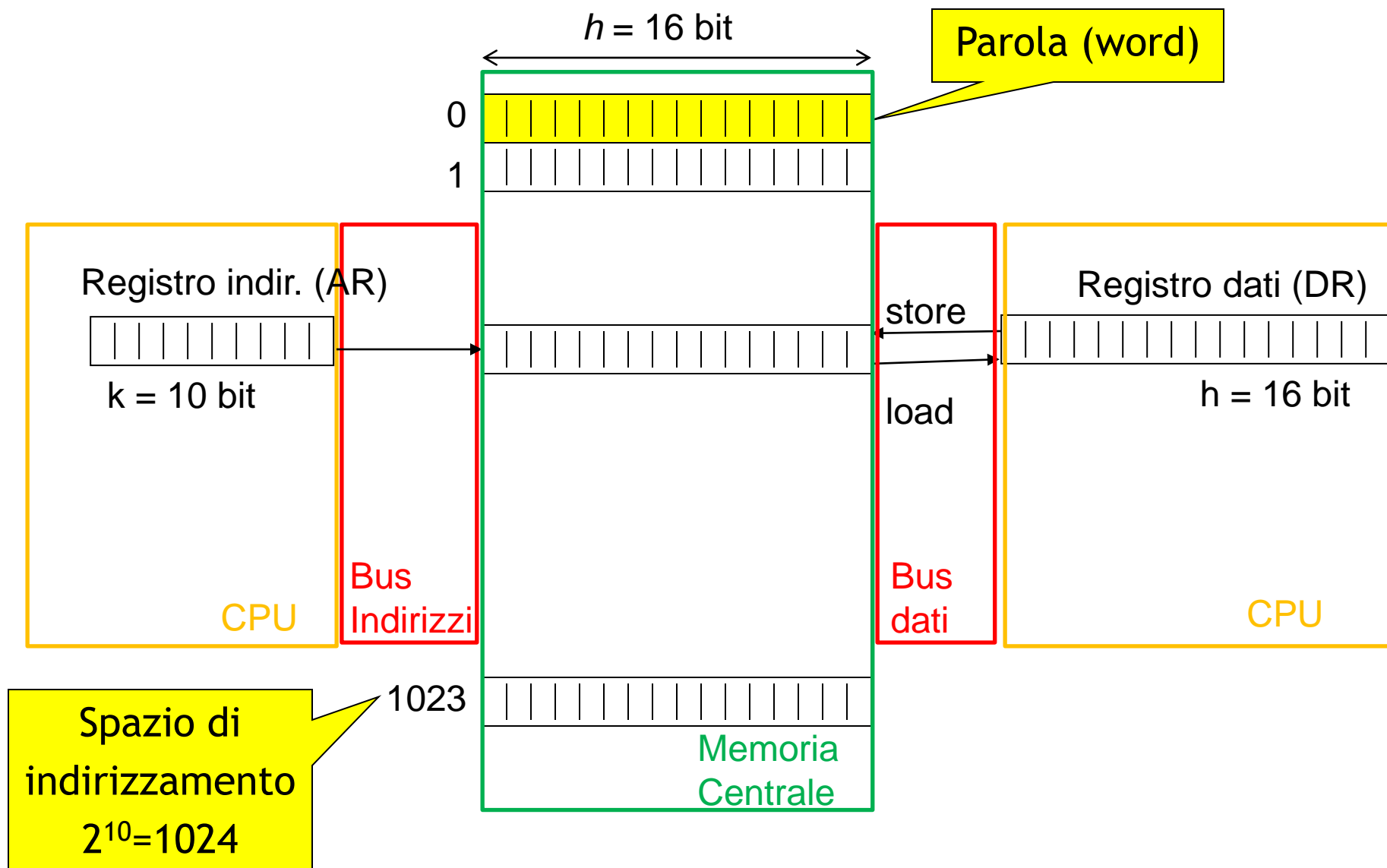


Bus di Sistema

- È un insieme di connessioni che permettono di trasferire l'informazione tra **due** entità funzionali (una 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 (il master lo usa per trasmettere allo slave i codici relativi alle istruzioni da eseguire, lo slave per dare feedback sull'esecuzione)

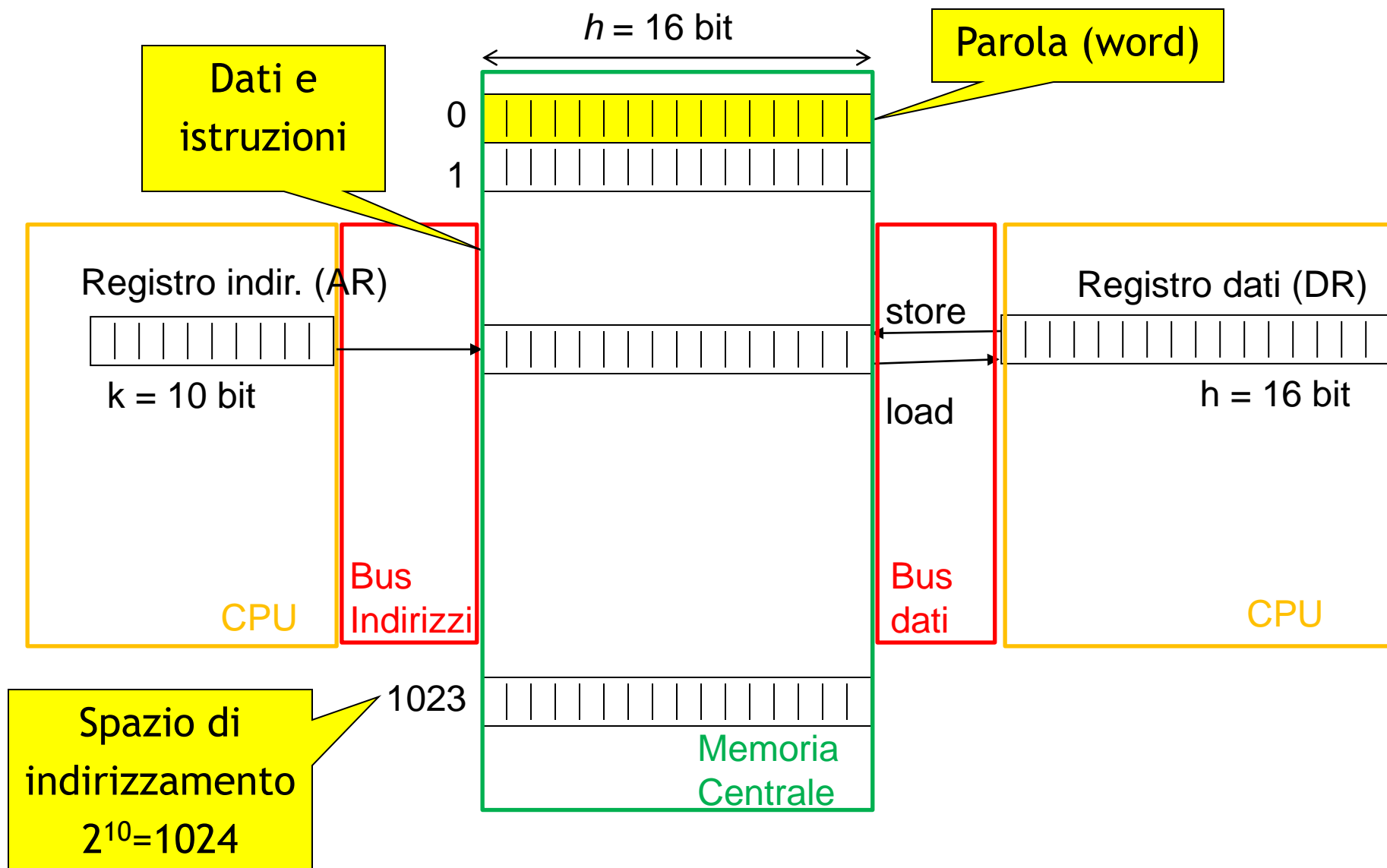


Lettura e Scrittura dalla Memoria Centrale



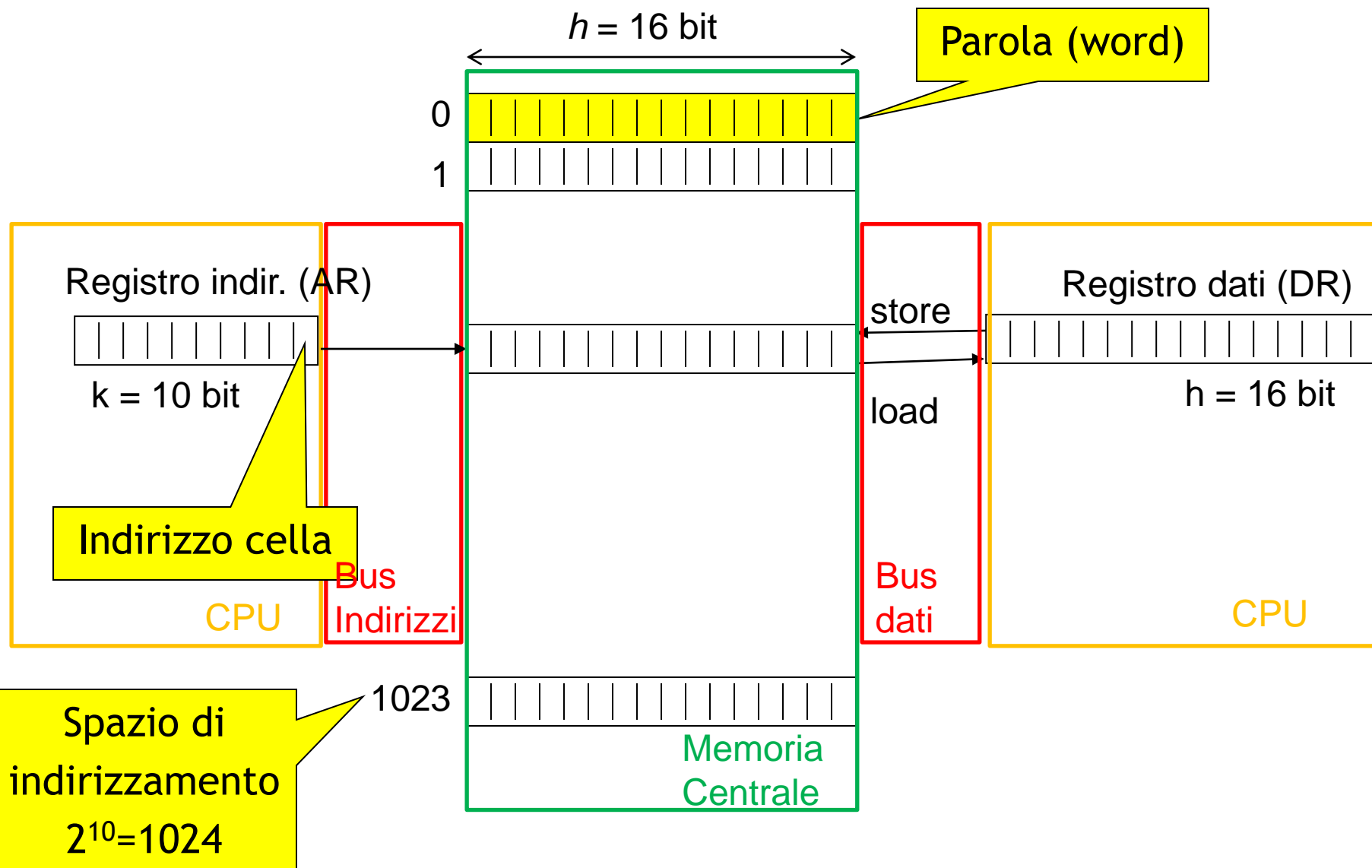


Lettura e Scrittura dalla Memoria Centrale



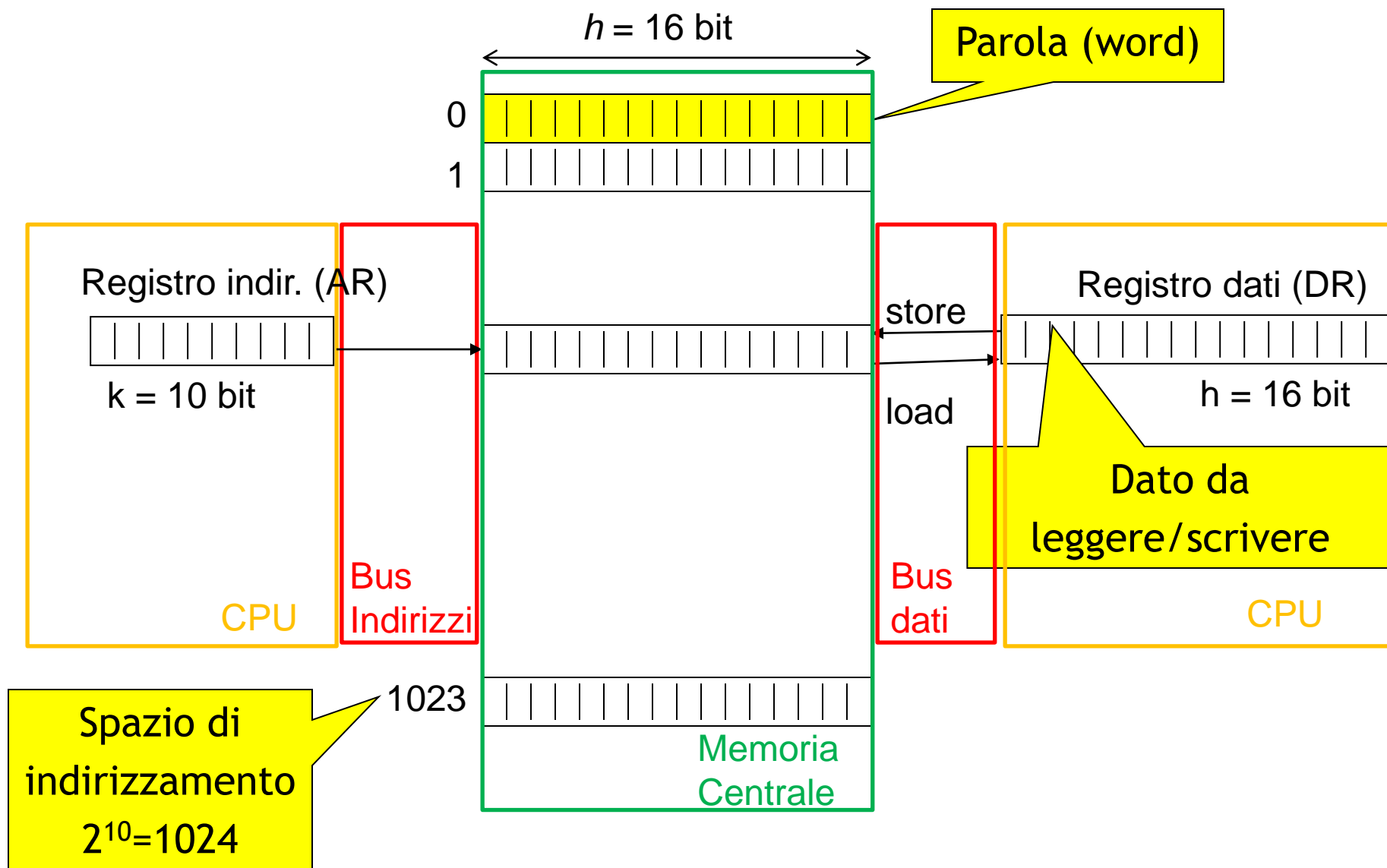


Lettura e Scrittura dalla Memoria Centrale



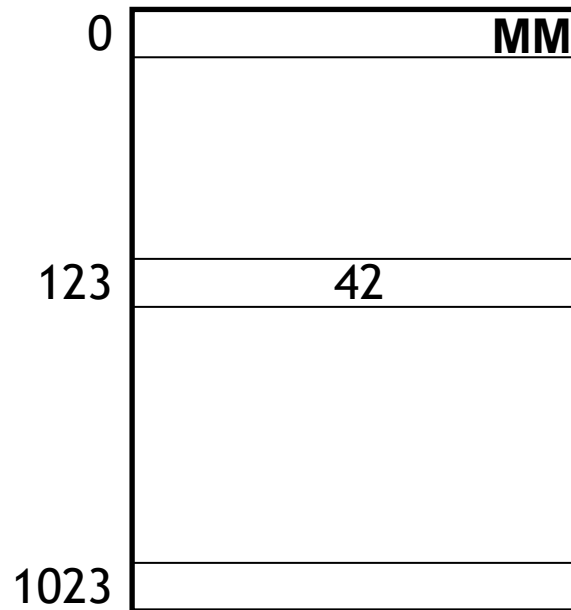
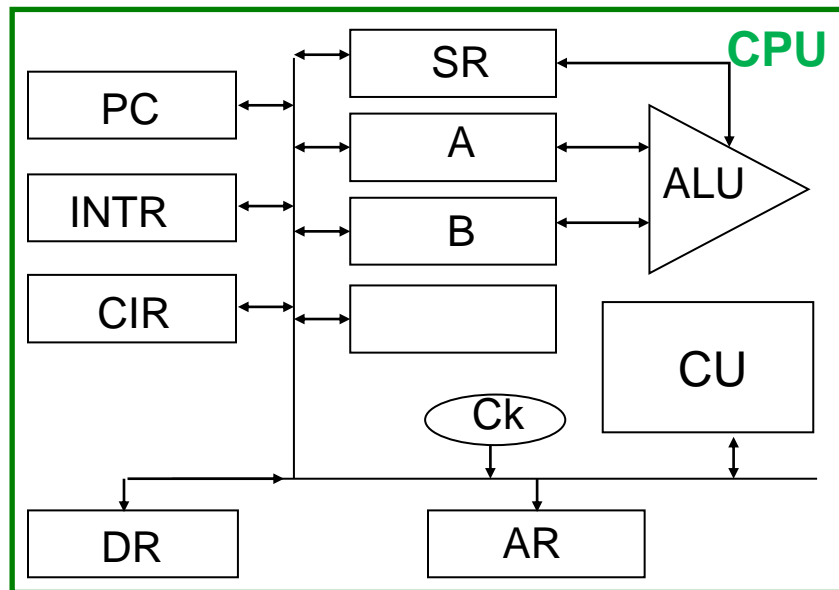


Lettura e Scrittura dalla Memoria Centrale





Sequenza di Lettura



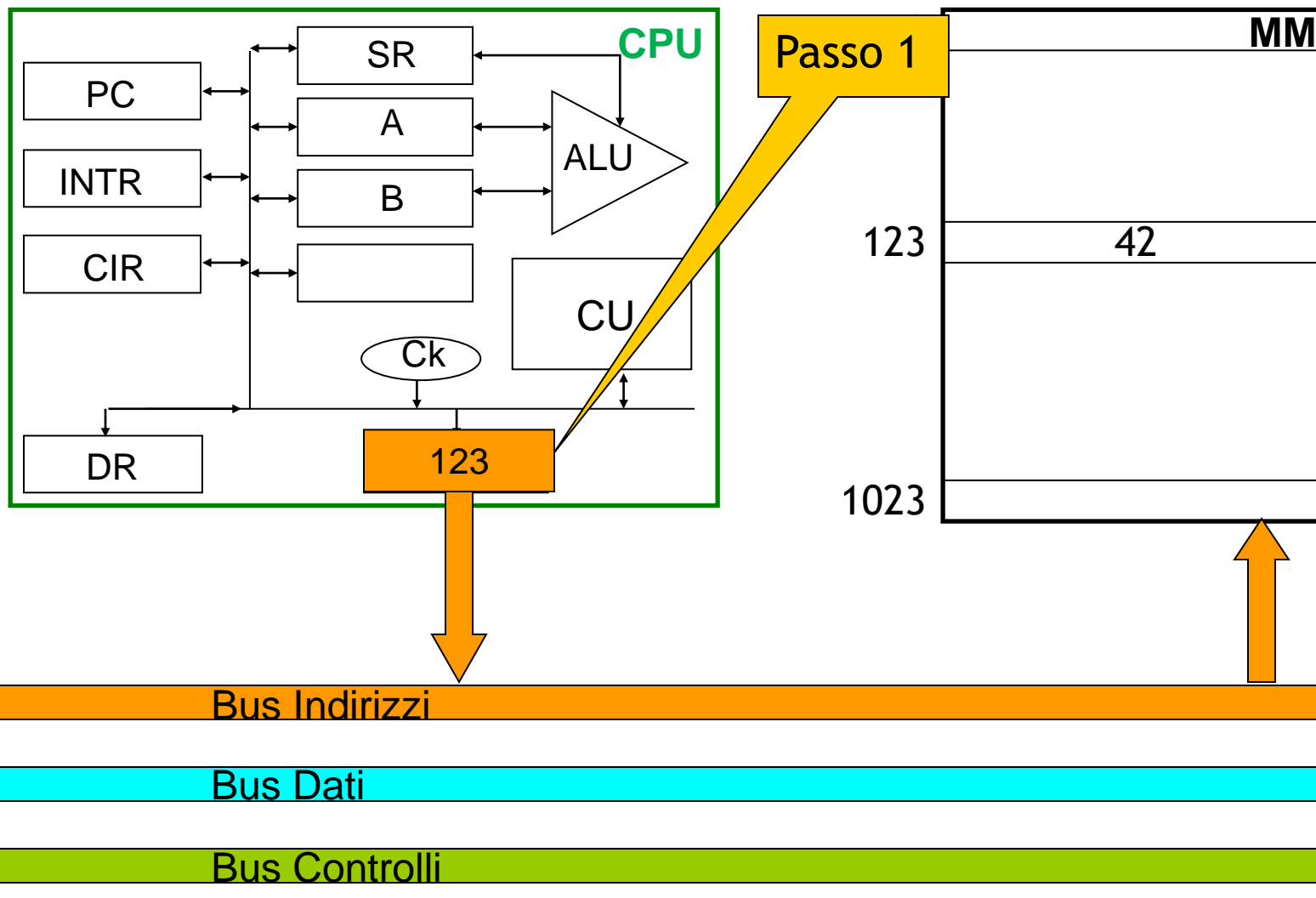
Bus Indirizzi

Bus Dati

Bus Controlli

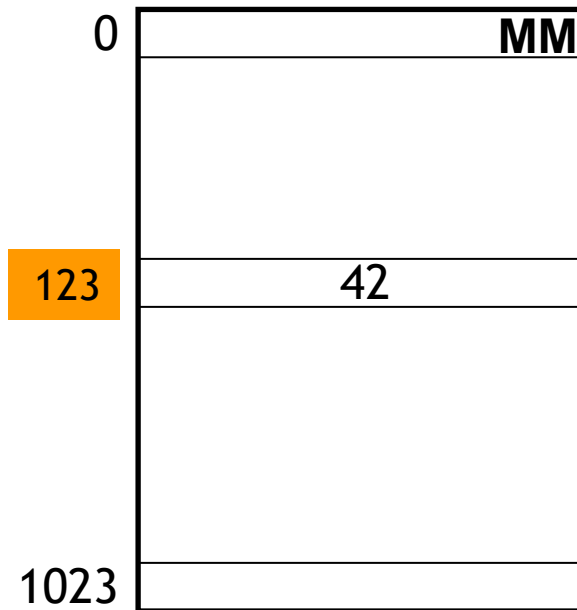
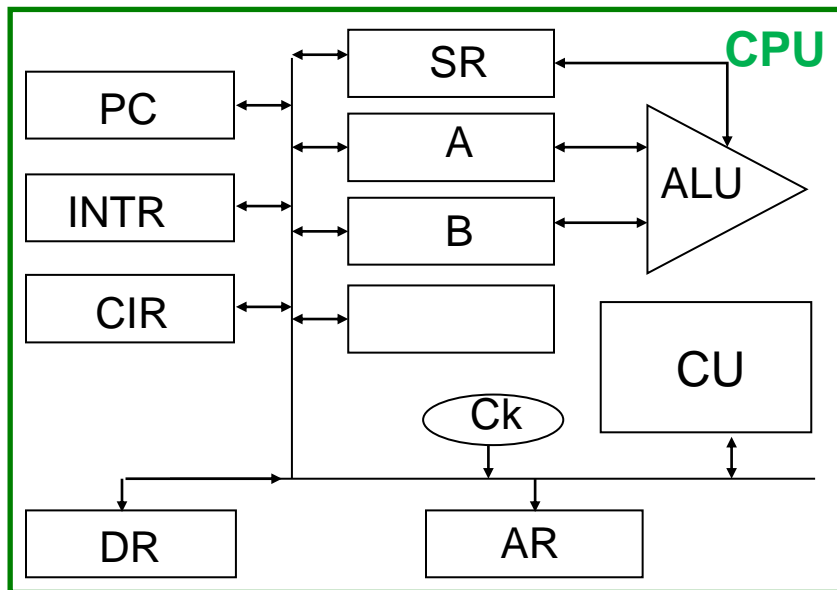


Sequenza di Lettura





Sequenza di Lettura



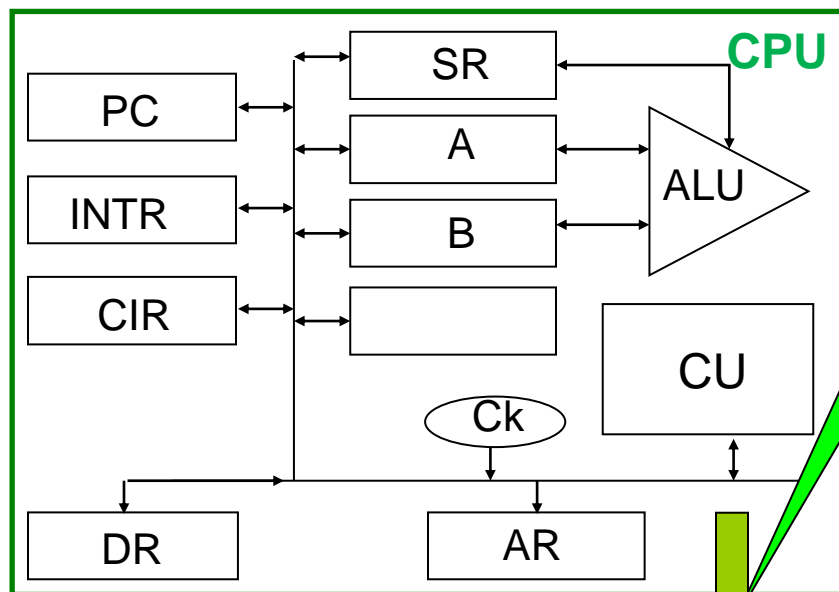
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Lettura



Passo 2

123

1023

MM

42

READ

Bus Indirizzi

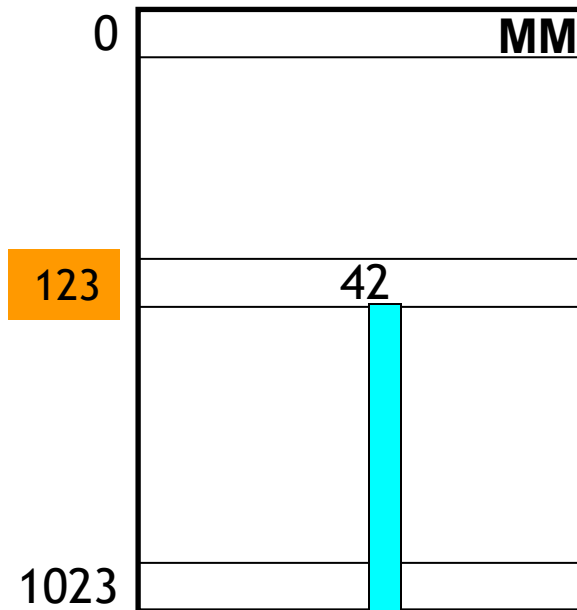
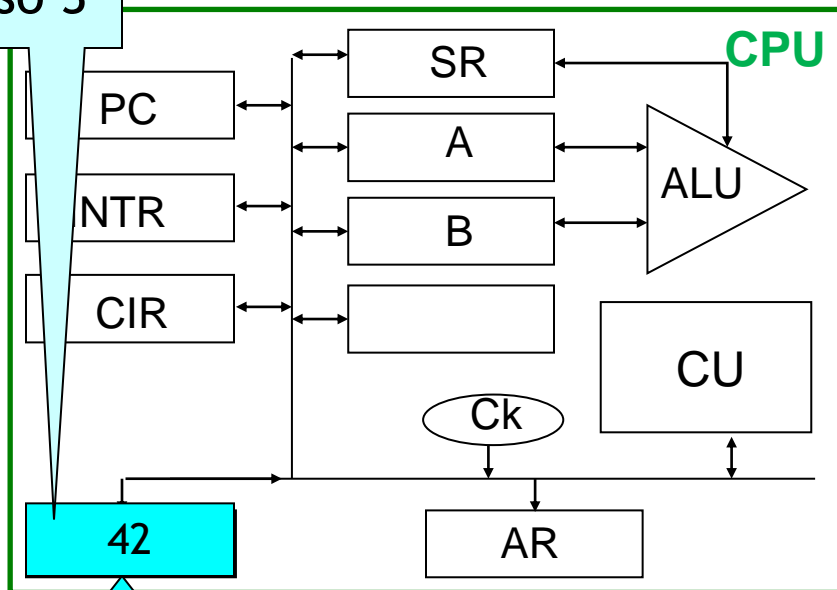
Bus Dati

Bus Controlli



Sequenza di Lettura

Passo 3



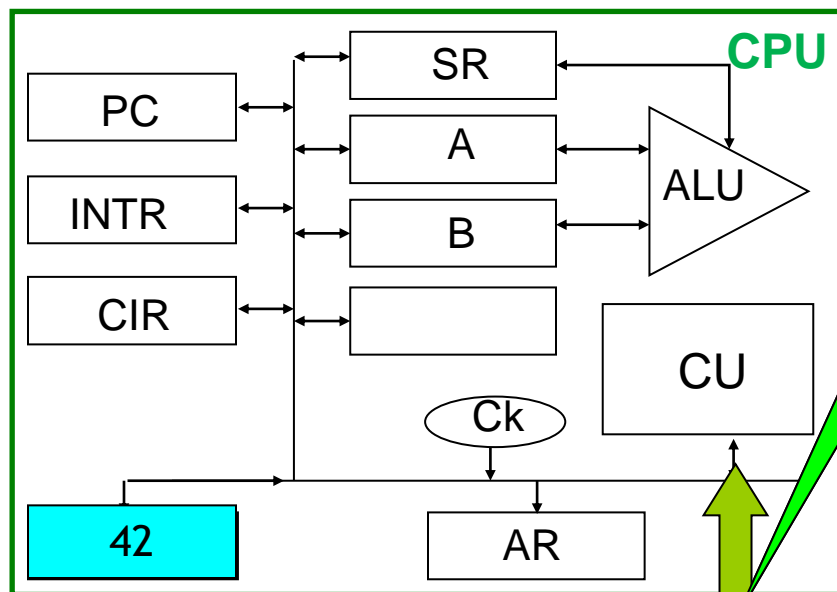
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Lettura



Passo 4

123

1023

MM

42

OK

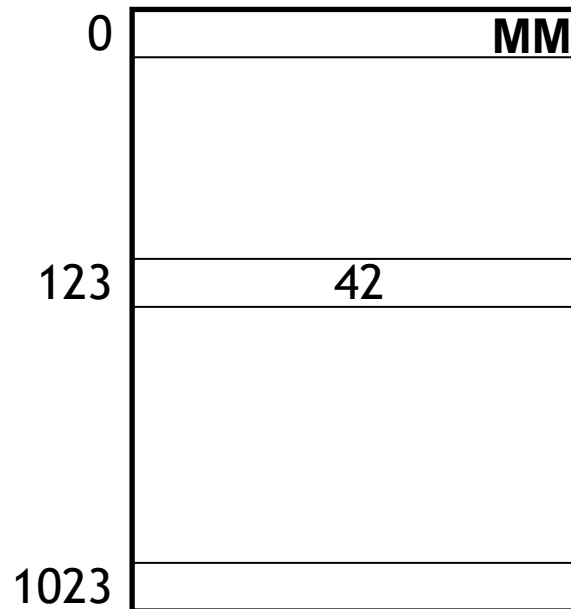
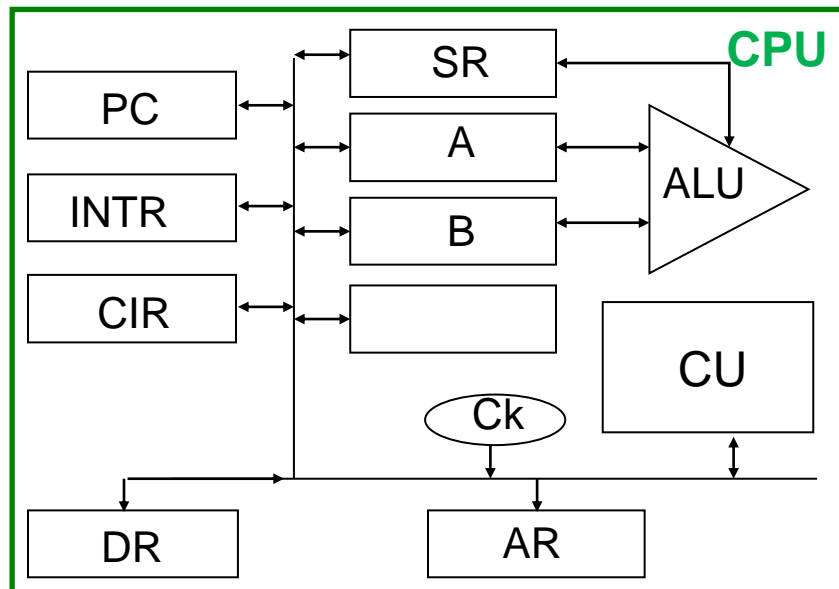
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



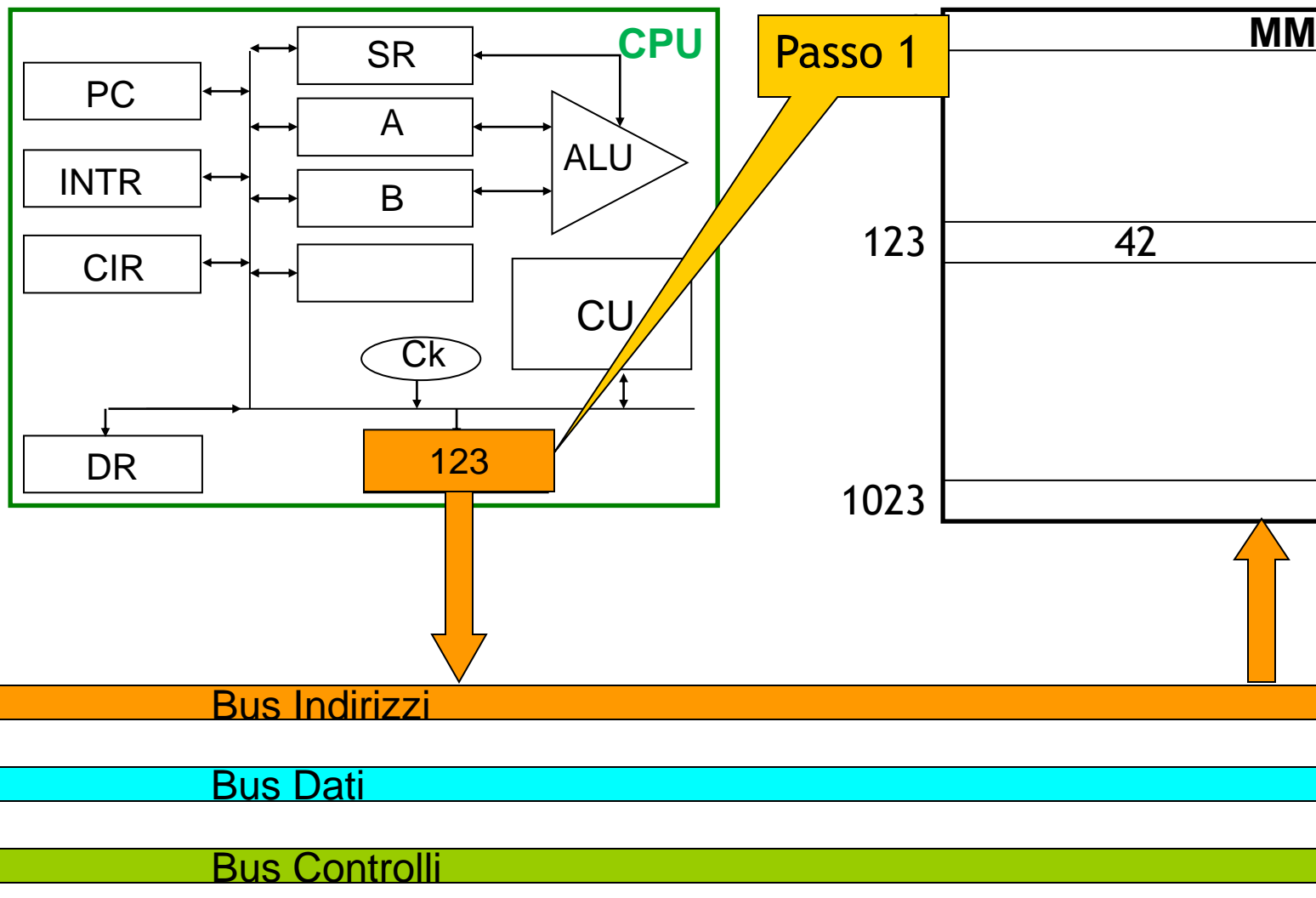
Bus Indirizzi

Bus Dati

Bus Controlli

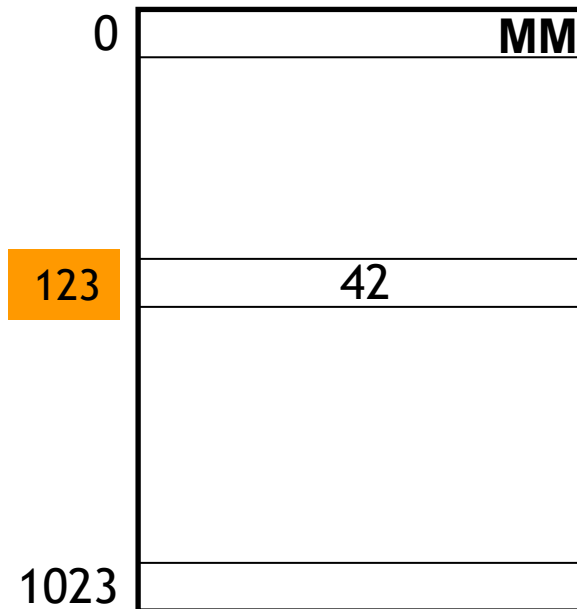
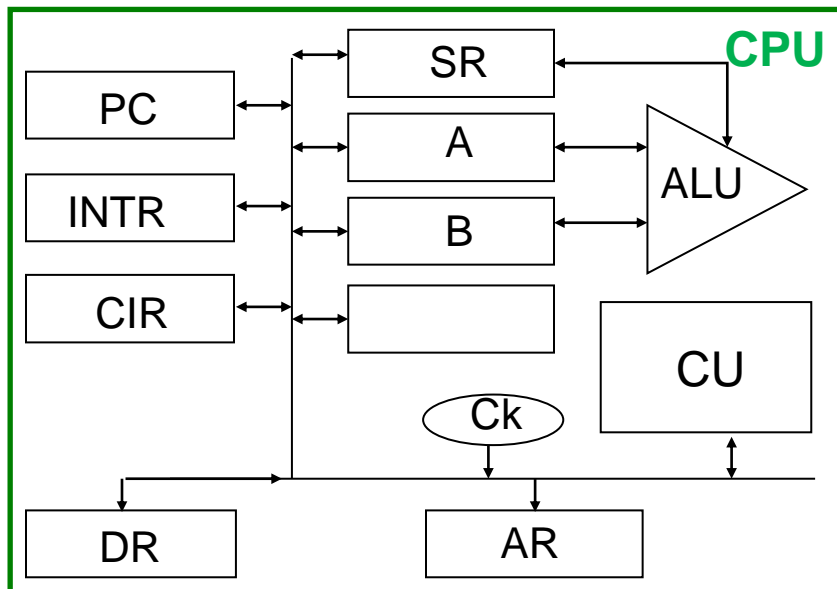


Sequenza di Scrittura





Sequenza di Scrittura



Bus Indirizzi

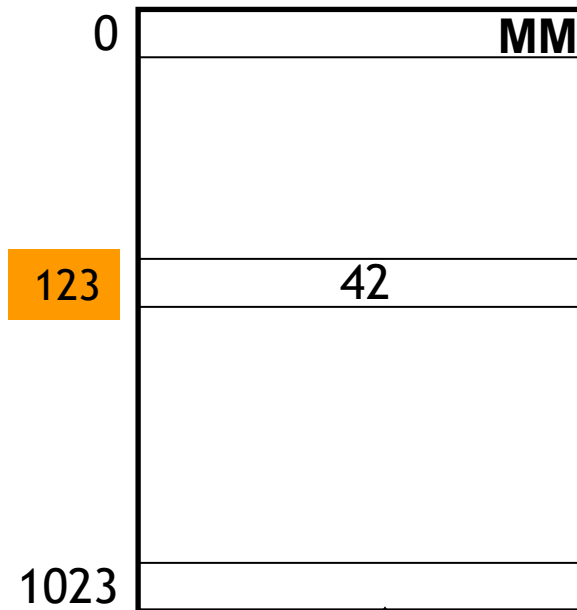
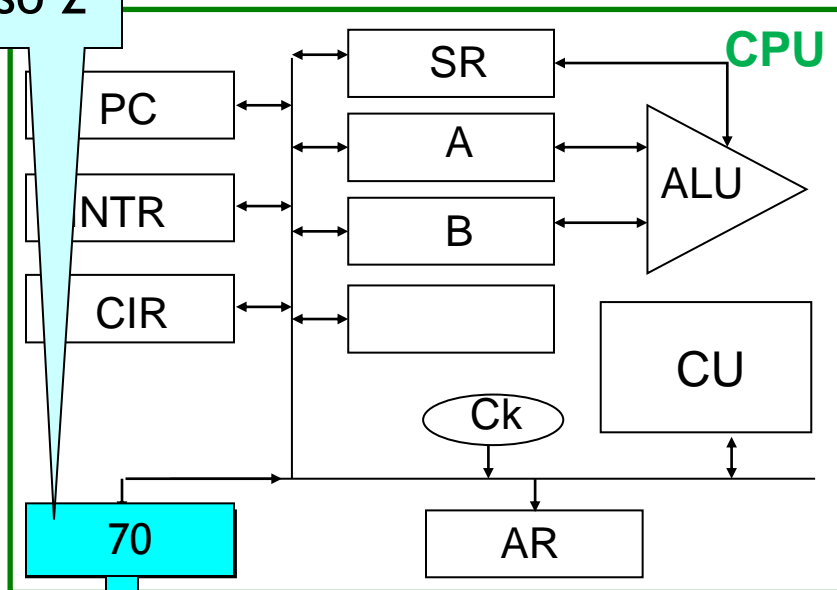
Bus Dati

Bus Controlli



Sequenza di Scrittura

Passo 2



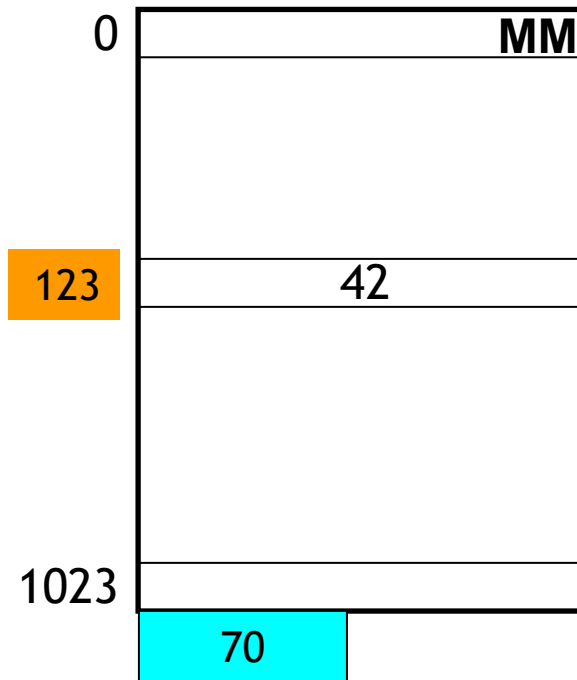
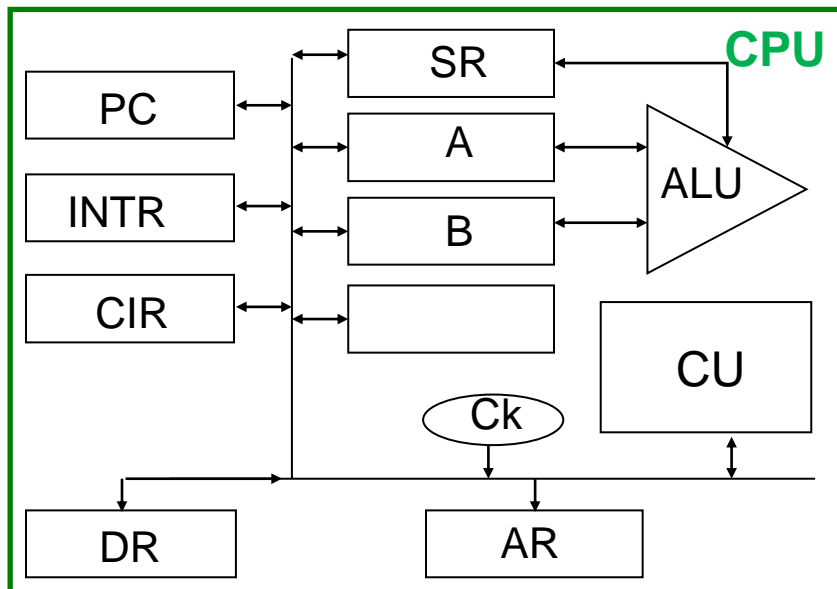
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



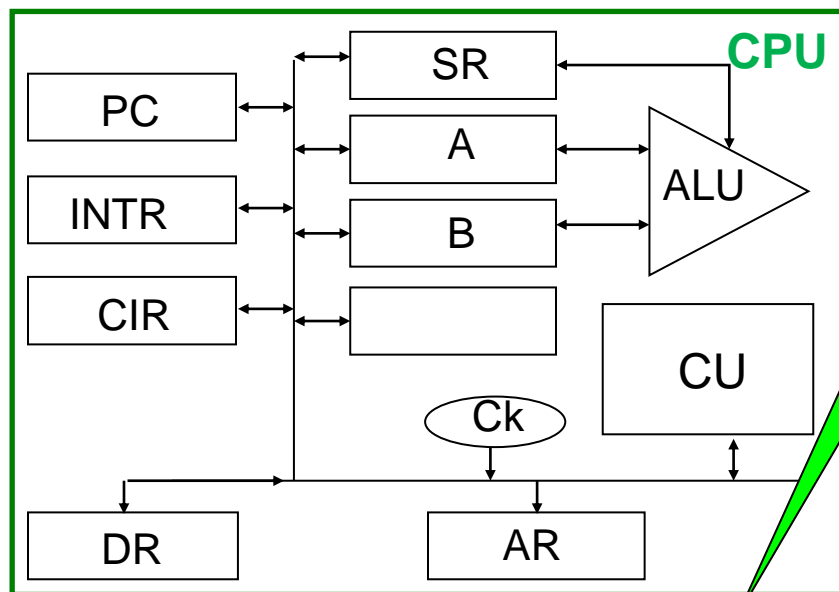
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



Passo 3

123

1023

MM

42

70

WRITE

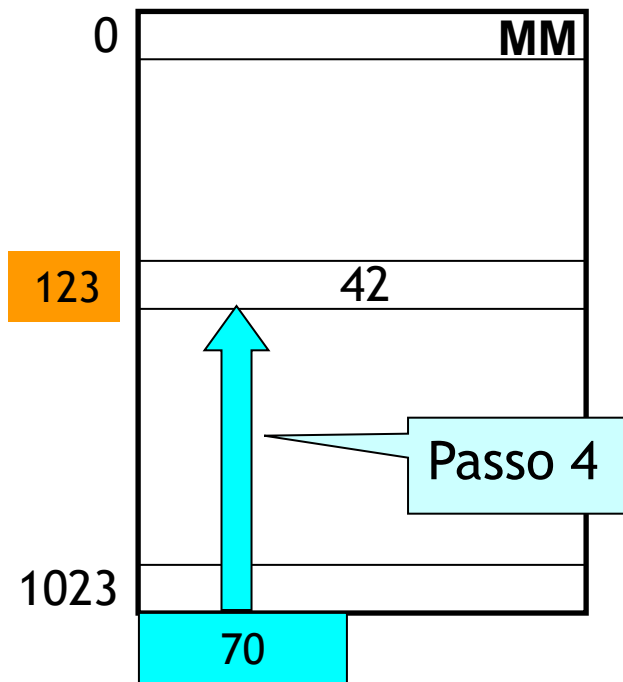
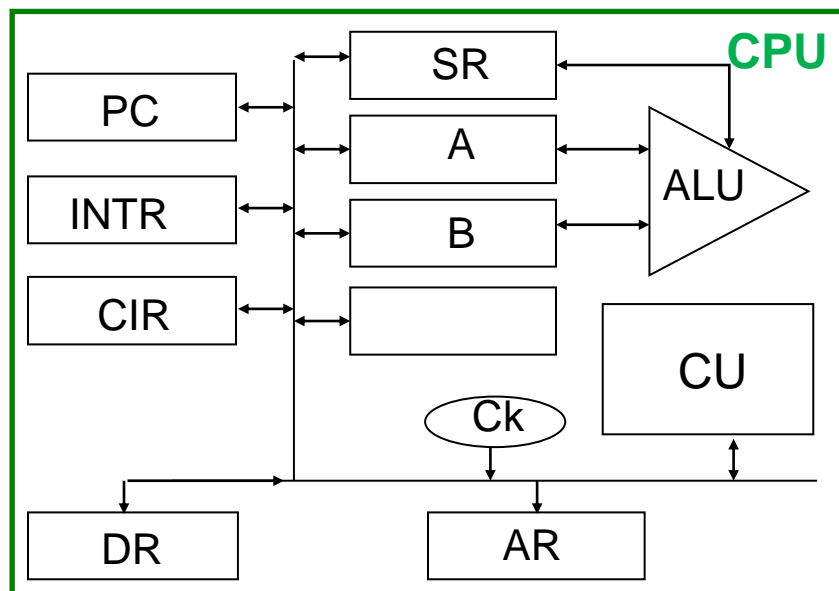
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



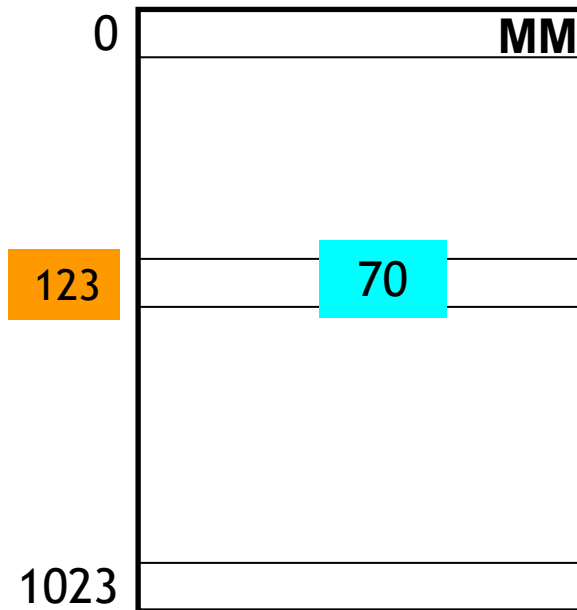
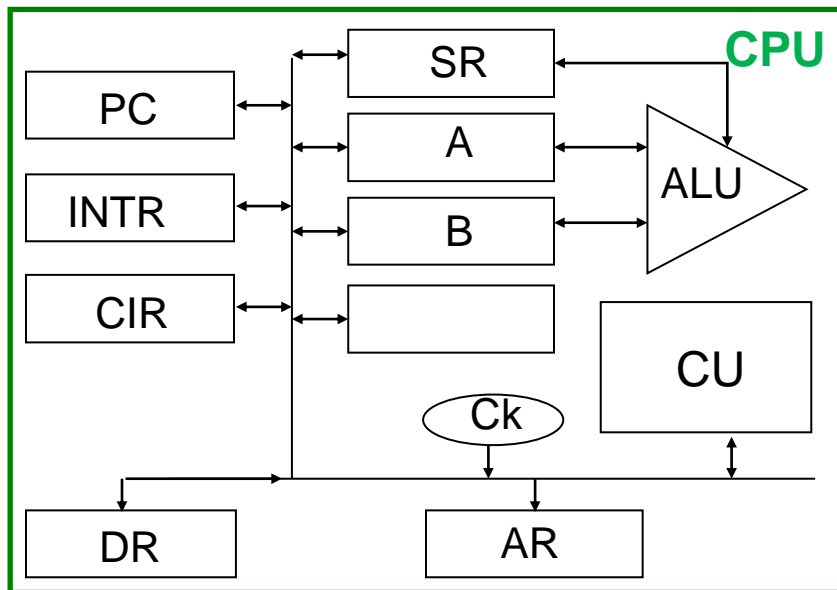
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



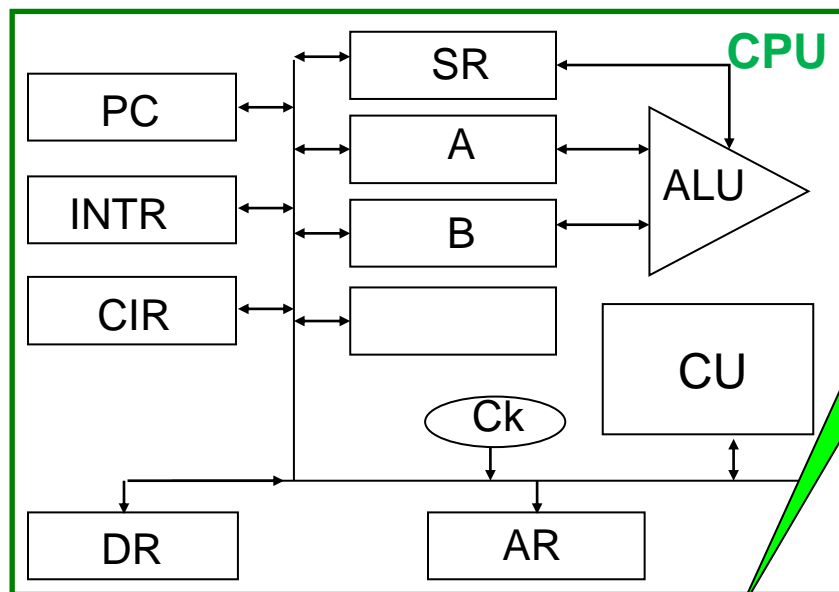
Bus Indirizzi

Bus Dati

Bus Controlli



Sequenza di Scrittura



Passo 5

123

70

1023

OK

Bus Indirizzi

Bus Dati

Bus Controlli



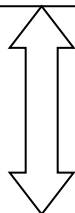
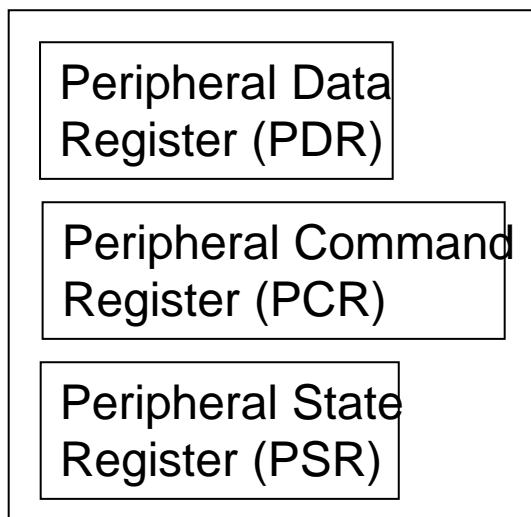
Interfacce alle Periferiche

- 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

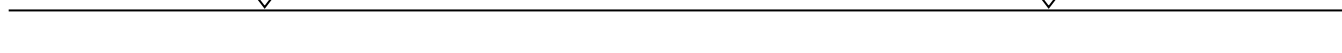
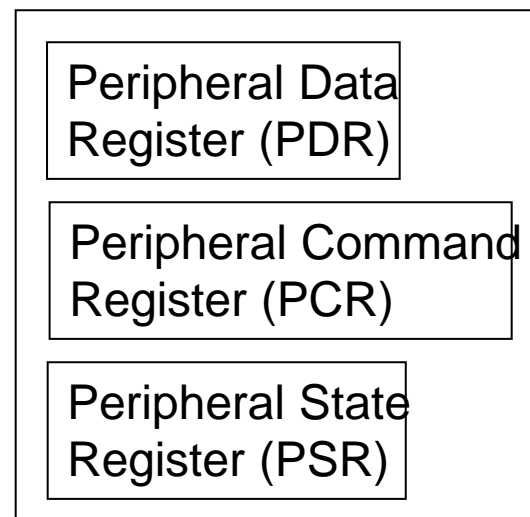


Interfacce alle Periferiche

Interfaccia periferica 1



Interfaccia periferica 2

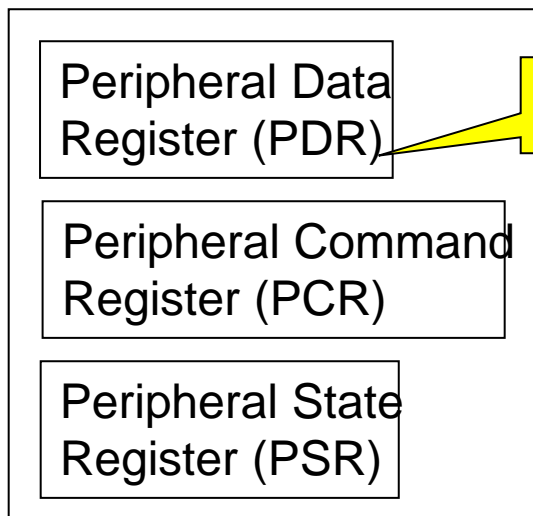


Bus di sistema

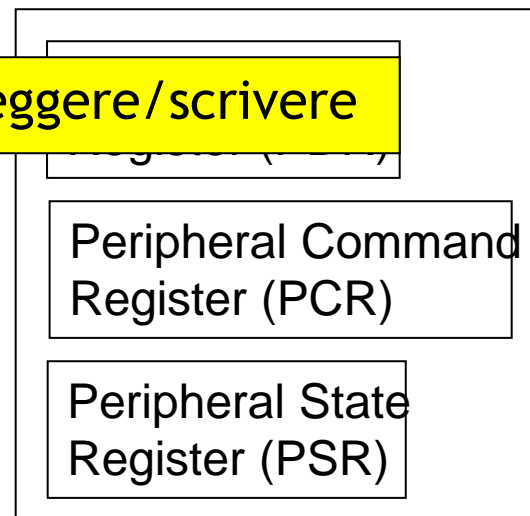


Interfacce alle Periferiche

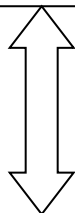
Interfaccia periferica 1



Interfaccia periferica 2



Dato da leggere/scrivere

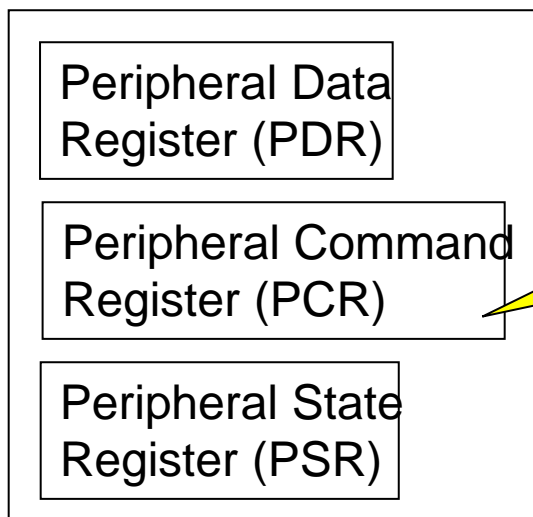


Bus di sistema

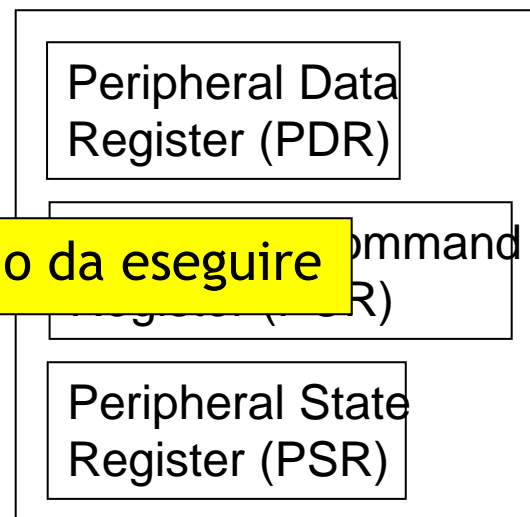


Interfacce alle Periferiche

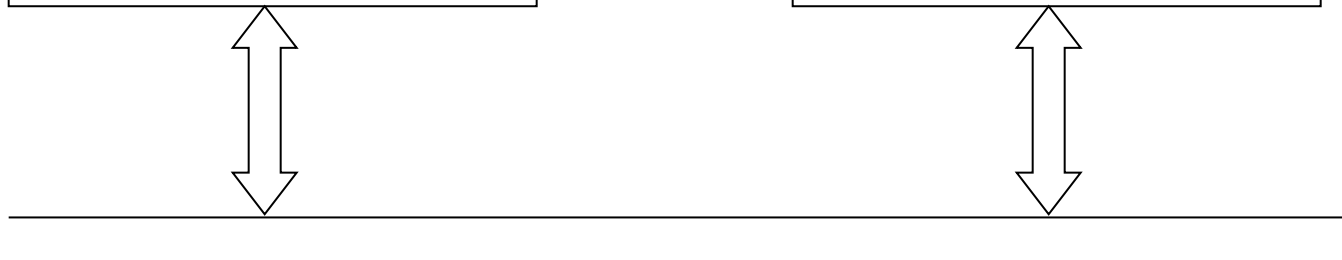
Interfaccia periferica 1



Interfaccia periferica 2



Comando da eseguire

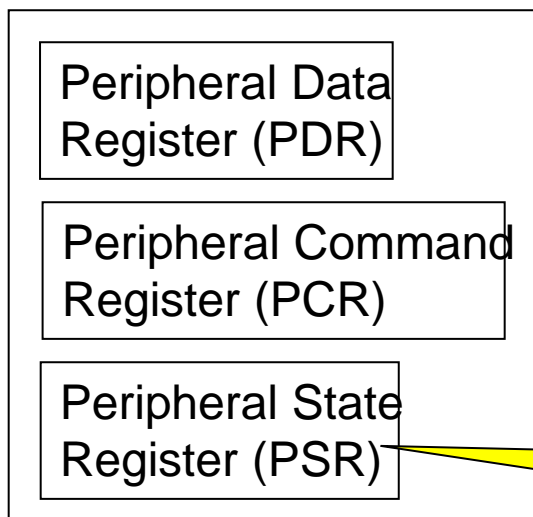


Bus di sistema

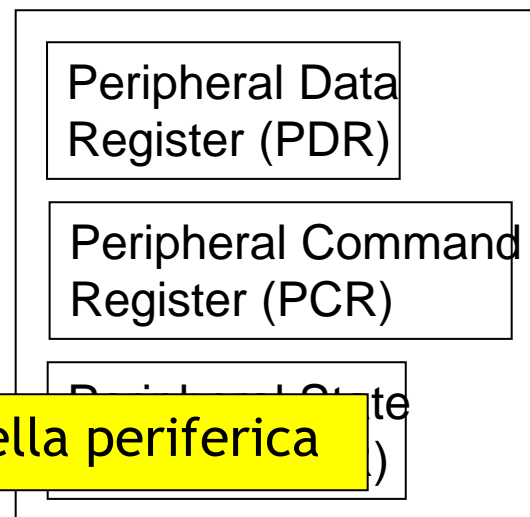


Interfacce alle Periferiche

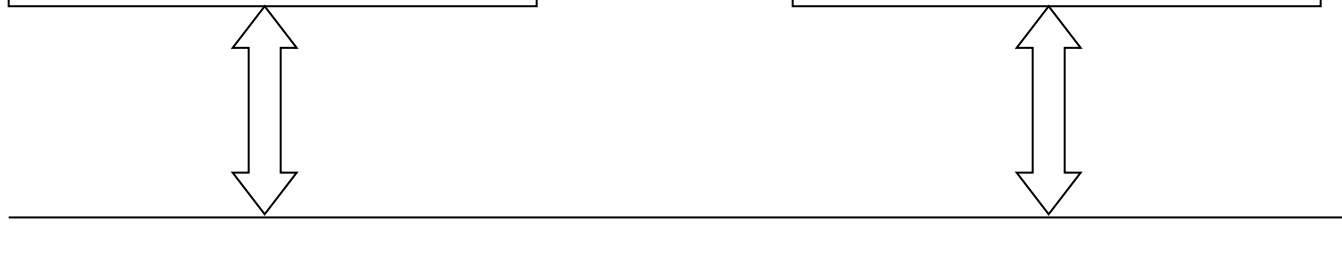
Interfaccia periferica 1



Interfaccia periferica 2



Stato della periferica



Bus di sistema



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**

- **Codice operativo:** indica quale istruzione si deve eseguire
- **Indirizzo operando:** indica in quale punto della memoria si trova l'operando



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) 00 **Indirizzo Operando (10bit)**
ad esempio, **0100**00**0000010000**



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) 00 **Indirizzo Operando (10bit)**
ad esempio, **0100**00**0000010000**

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)

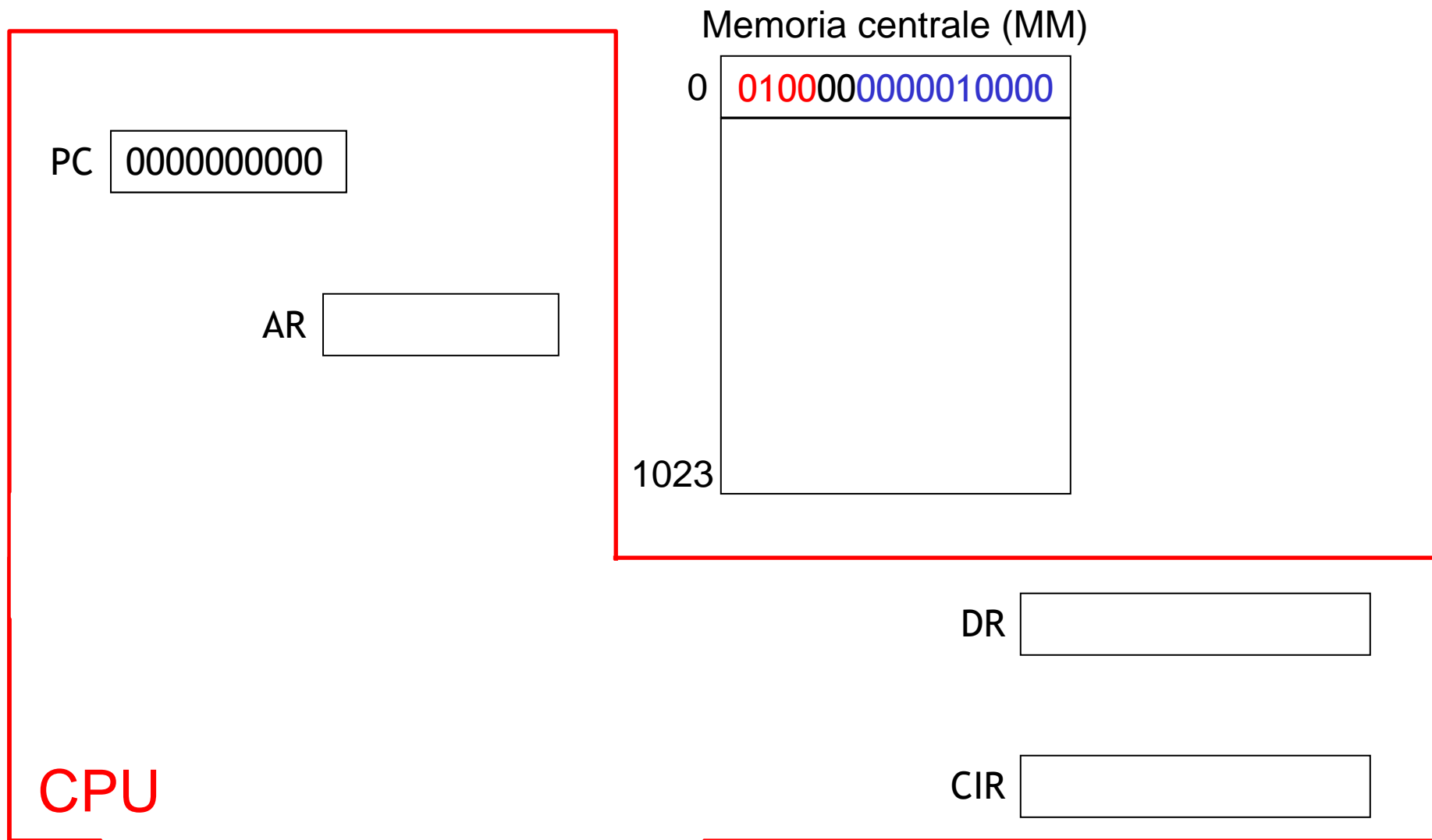


Le Tre Fasi Per Eseguire un'Istruzioni

1. **Fetch:** Acquisizione dell'istruzione dalla MM
 1. Trasferimento da PC a AR dell' indirizzo della 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, legge dal CIR
3. **Esecuzione:** dipende dall'istruzione specifica.



Fase di Fetch





Fase di Fetch

Passo 1

PC 0000000000

AR 0000000000

Memoria centrale (MM)

0 0100000000010000

1023

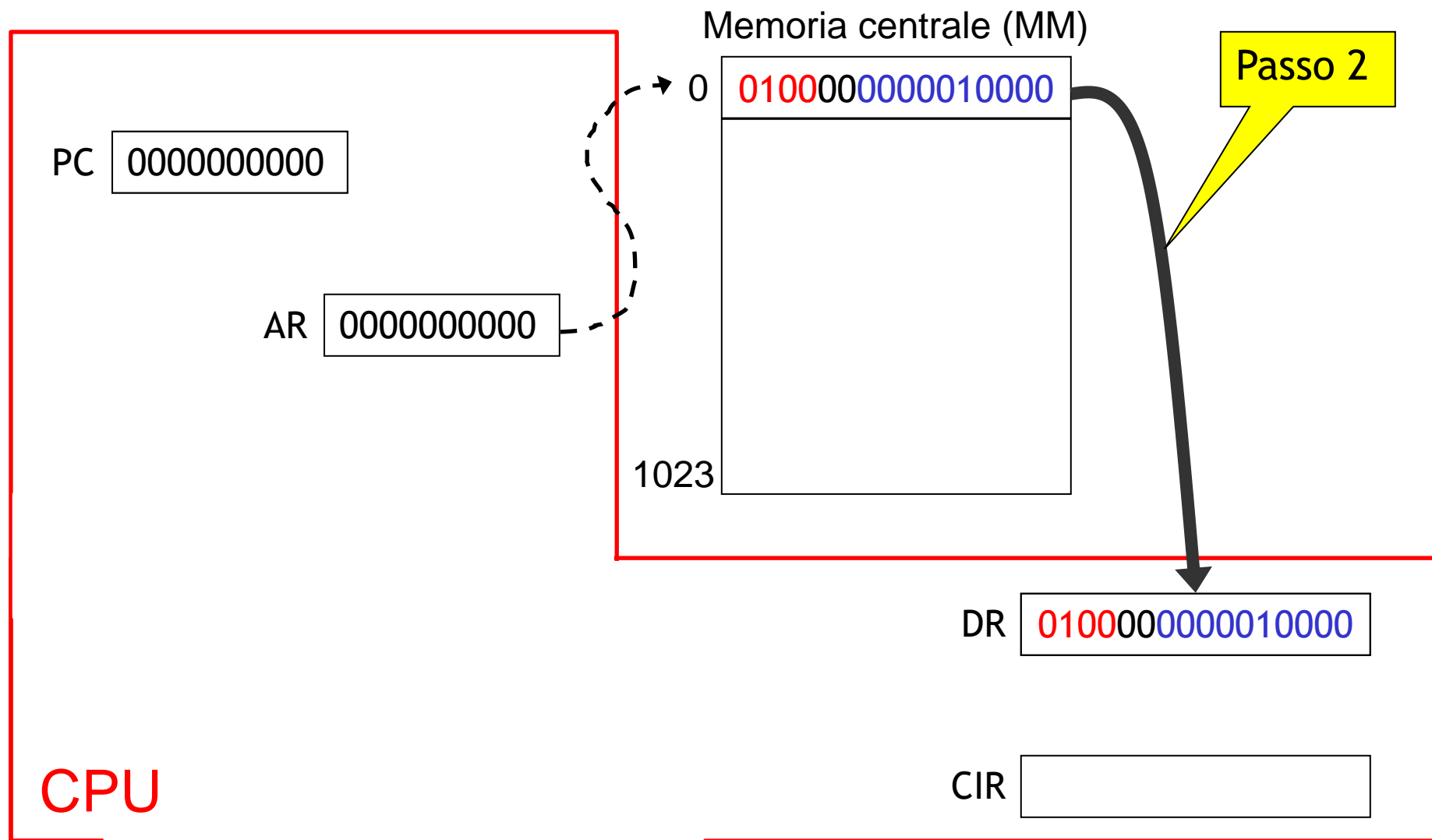
DR

CIR

CPU

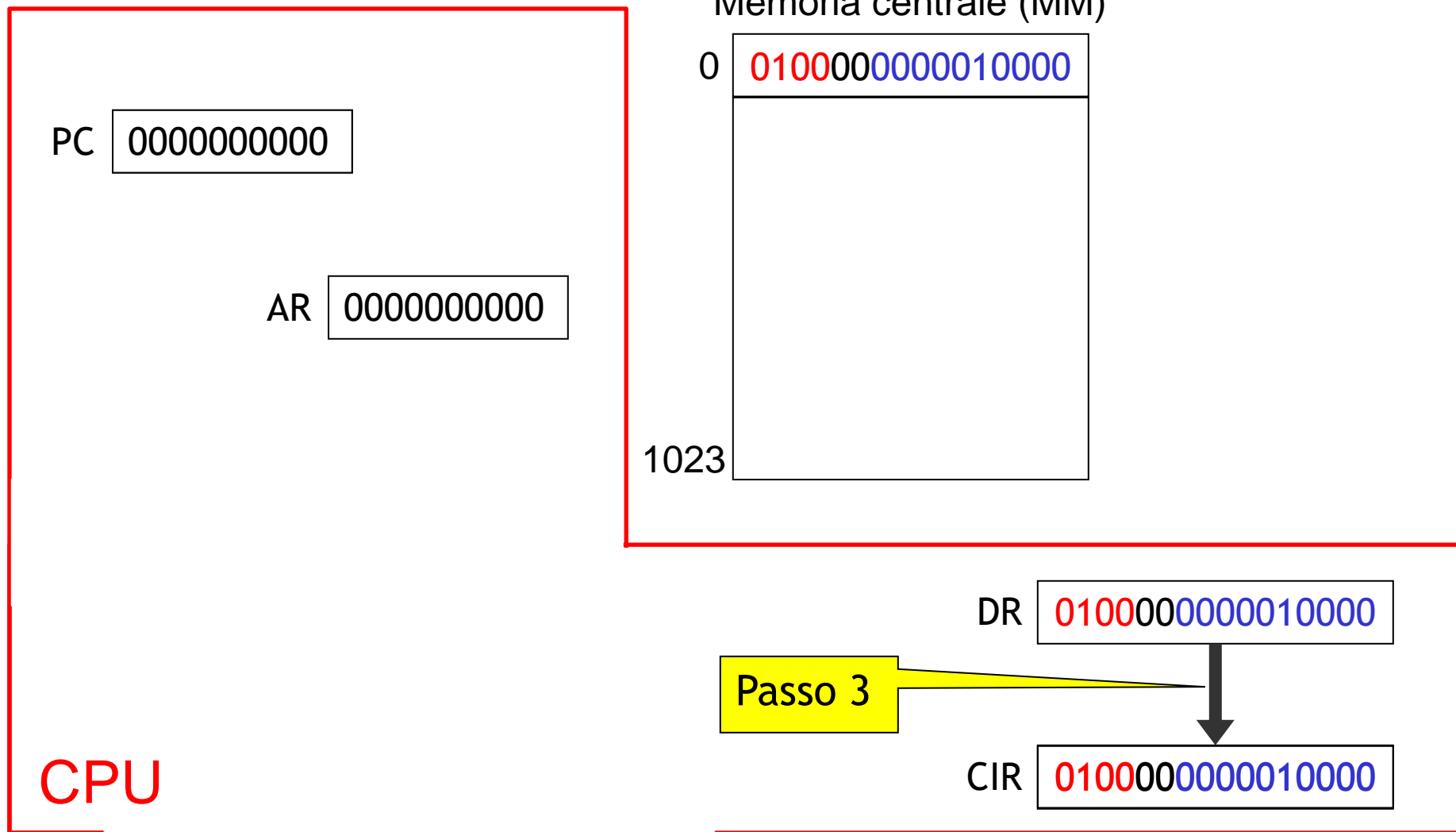


Fase di Fetch



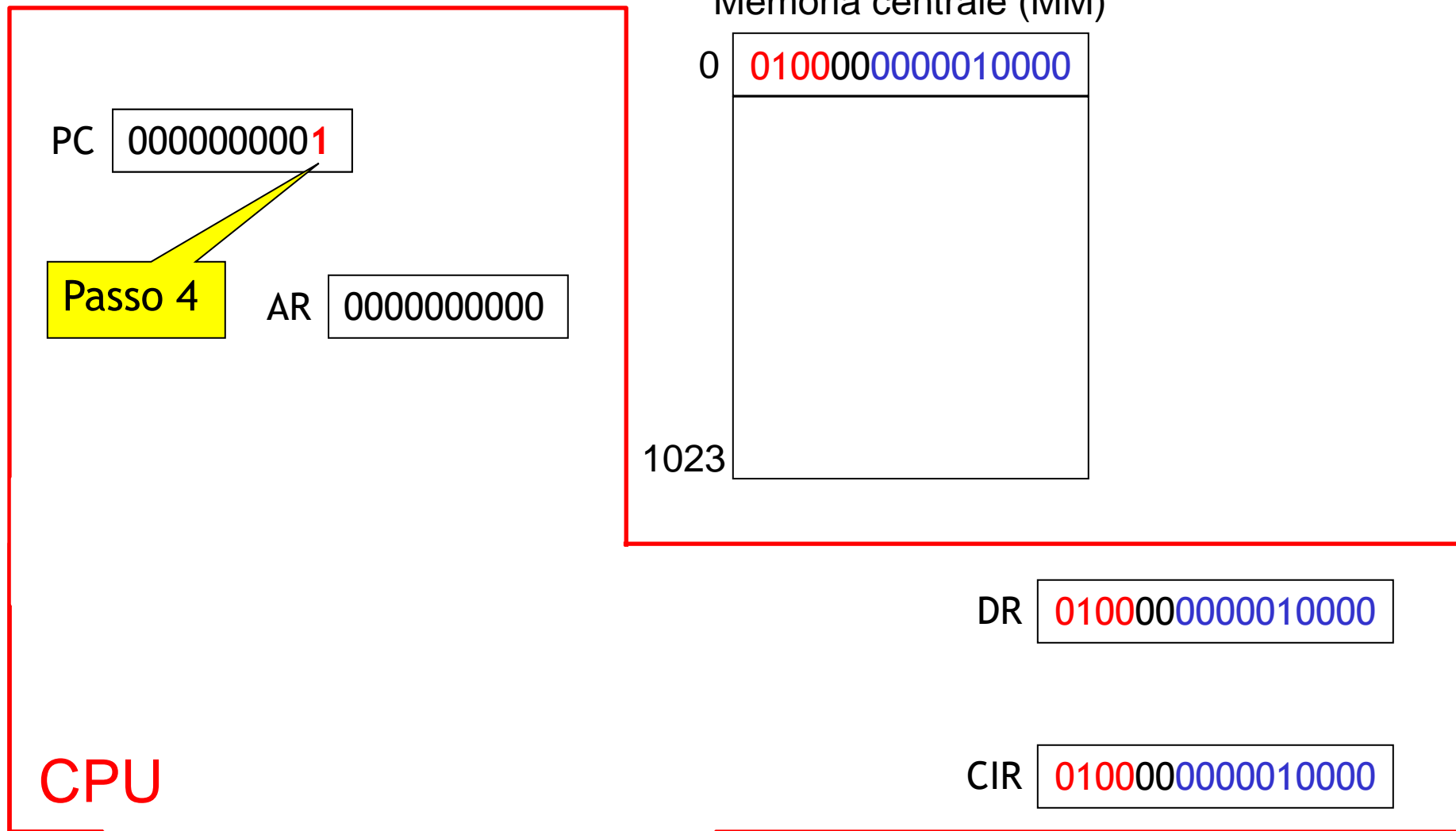


Fase di Fetch






Fase di Fetch





Fase di Decodifica 1^a istruzione

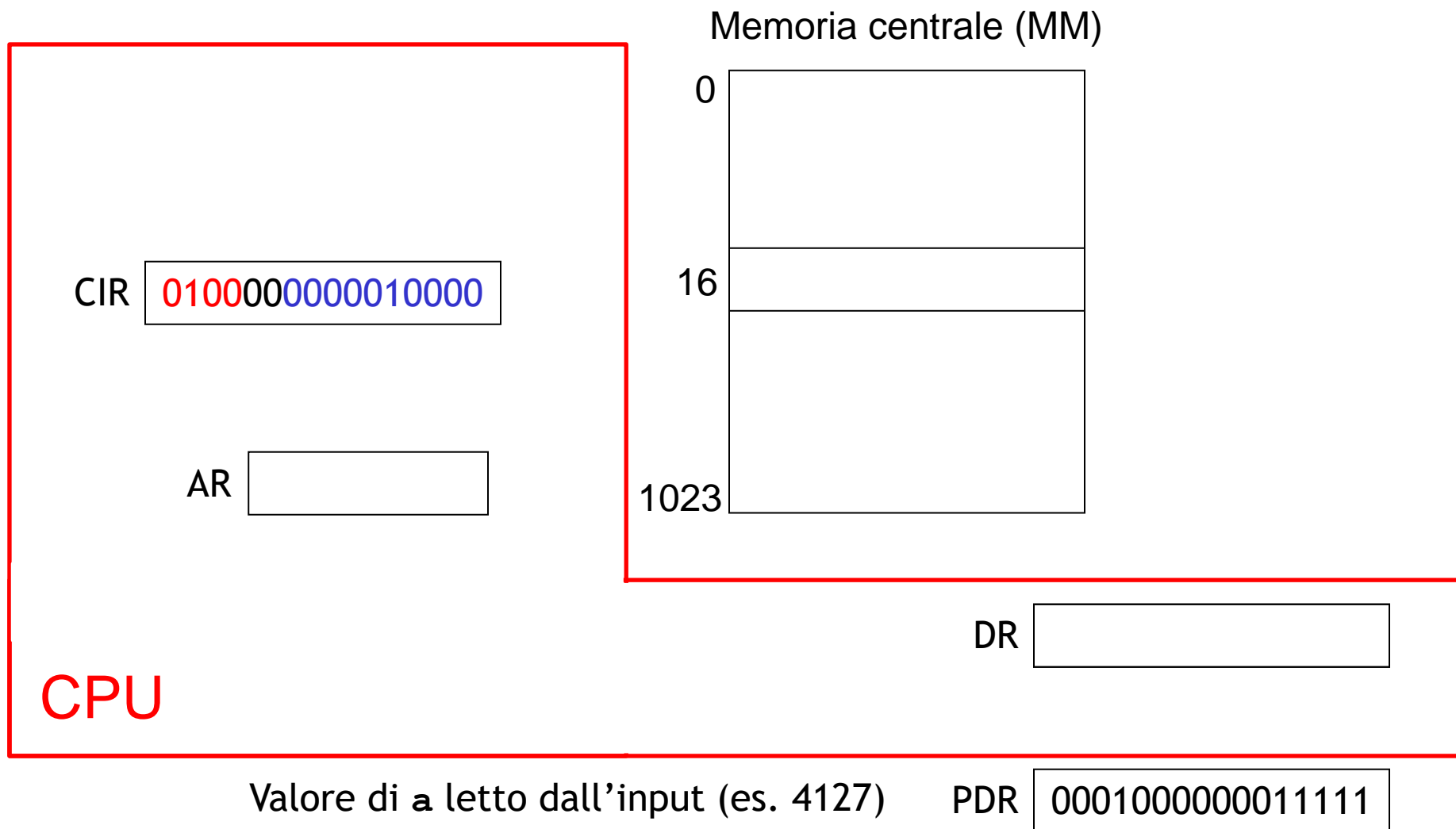
CIR 010000000010000



Codice operativo **0100** = leggi da input e salva il dato letto all'indirizzo specificato come operando (00**0000010000**)

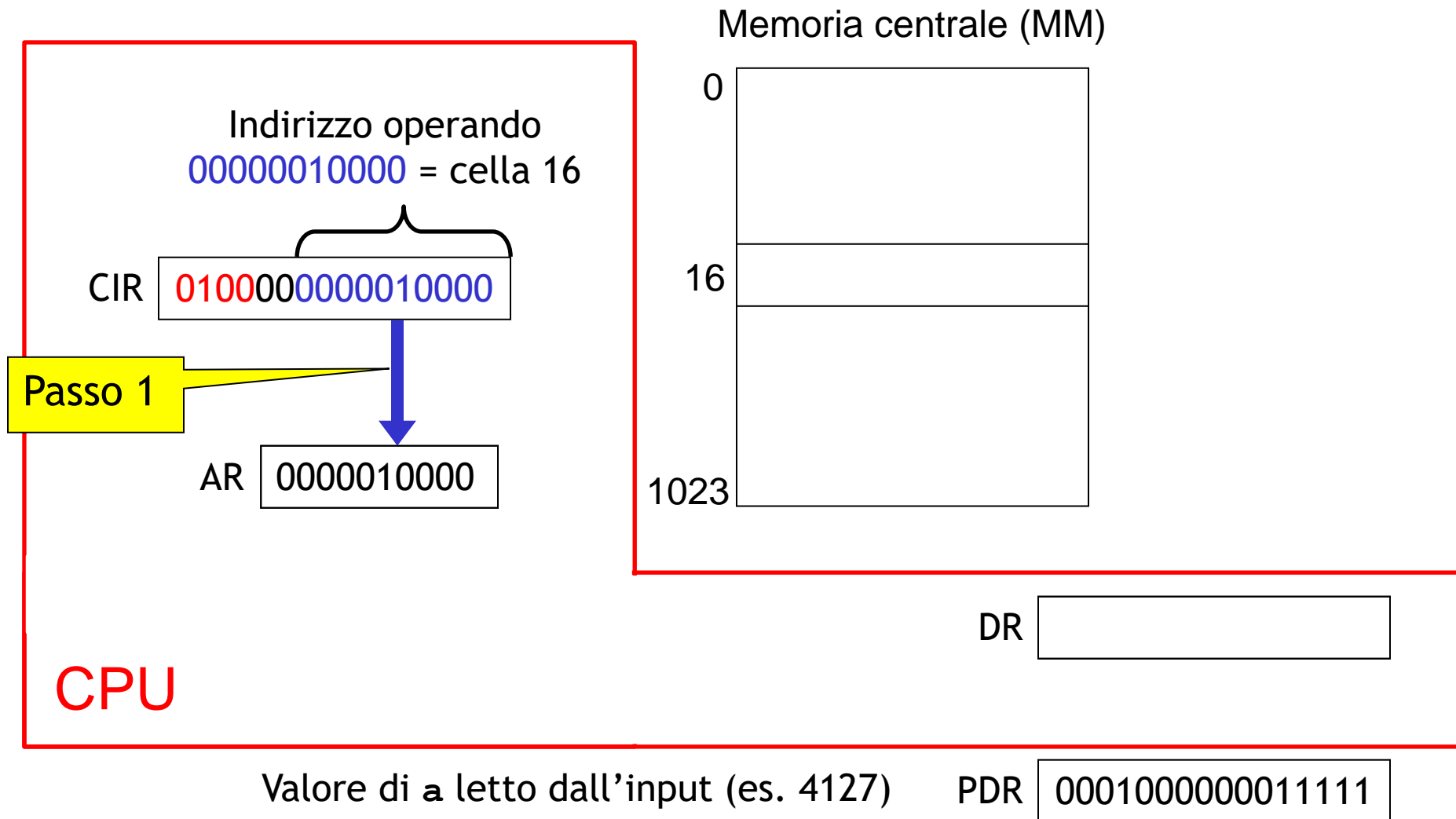


Fase di esecuzione: esempio lettura da input



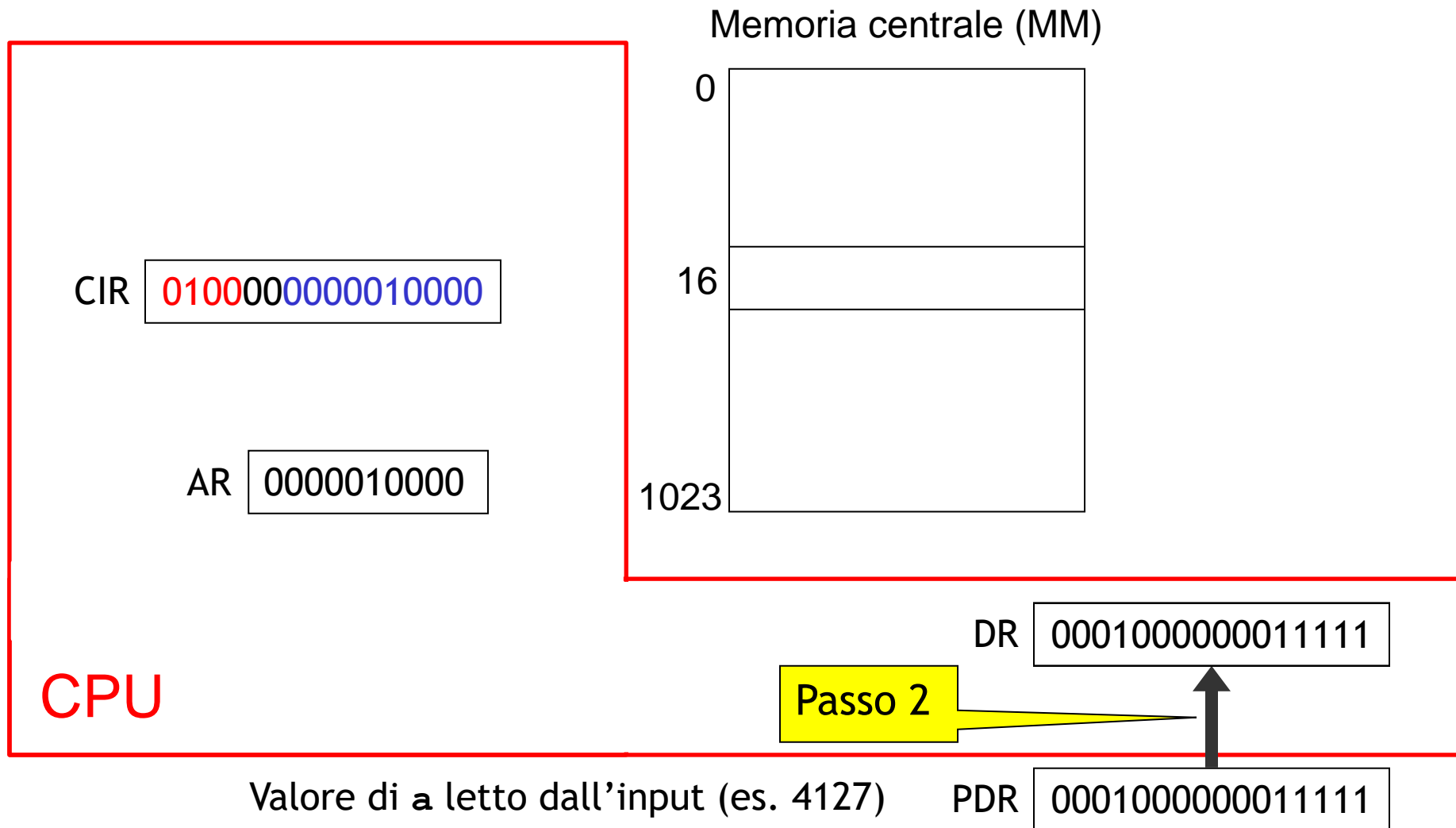


Fase di esecuzione: esempio lettura da input



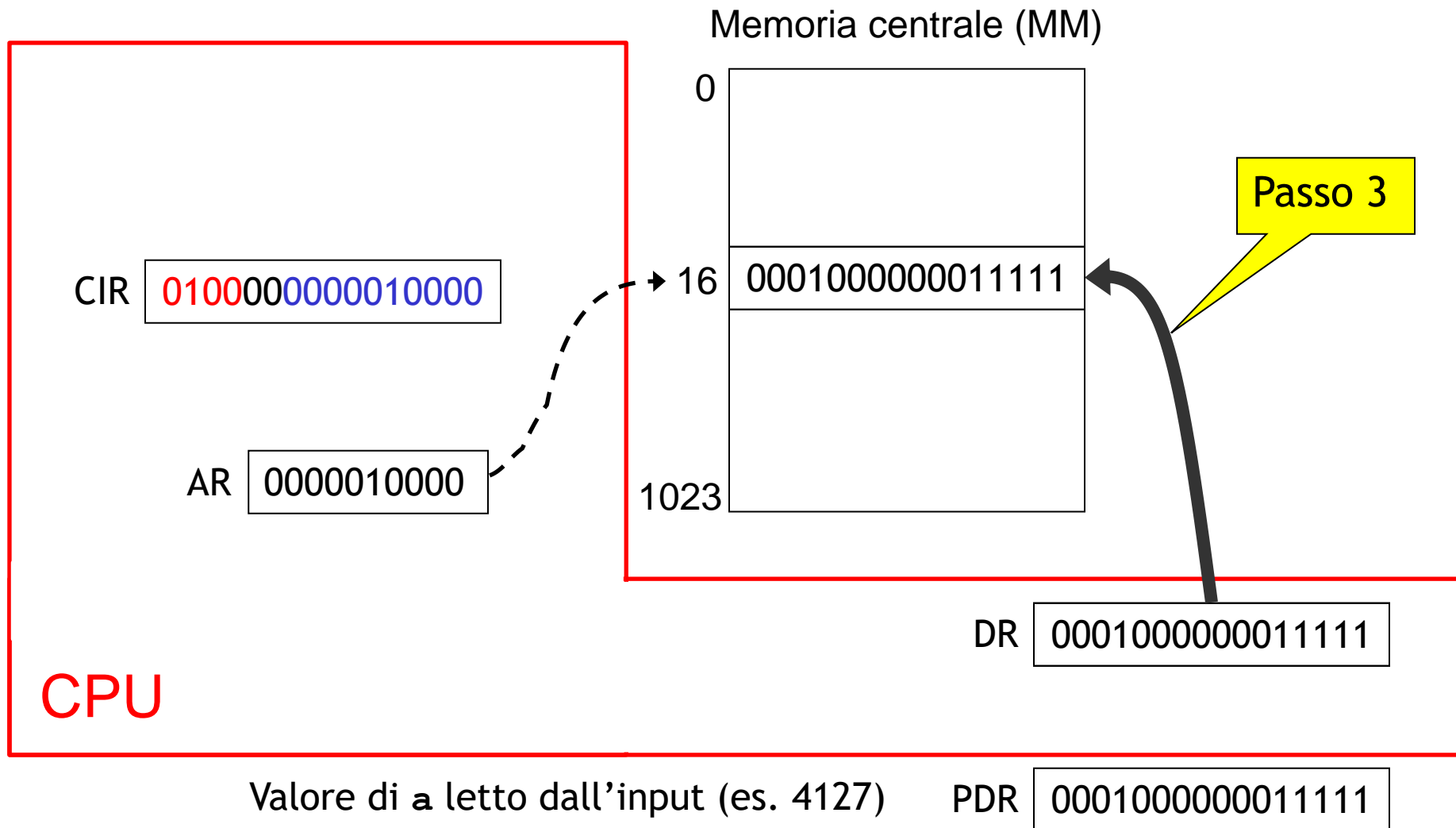


Fase di esecuzione: esempio lettura da input





Fase di esecuzione: esempio lettura da input





Programma in Memoria Centrale

	Cella 0	010000000010000	Istruzioni del Programma
	1	010000000010001	
	2	000000000010000	
	3	000100000010001	
	4	011000000000000	
	5	010000000010010	
	6	010000000010011	
	7	001000000010100	
	8	000000000010010	
	9	000100000010011	
	10	011000000000000	
	11	000100000010011	
	12	100000000000000	
	13	001000000010100	
	14	010100000010100	
	15	110100000000000	
Spazio riservato per a	16		dati
Spazio riservato per b	17		
Spazio riservato per c	18		
Spazio riservato per d	19		
Spazio riservato per z	20		



Forma Binaria del Programma

010000000010000	Leggi un valore dall'input e mettilo nella cella 16 (a)
010000000010001	Leggi un valore dall'input e mettilo nella cella 17 (b)
000000000010000	Carica il contenuto della cella 16 (a) nel registro A
000100000010001	Carica il contenuto della cella 17 (b) nel registro B
011000000000000	Somma i registri A e B
001000000010100	Scarica il contenuto di A nella cella 20 (z) (ris.parziale)
010000000010010	Leggi un valore dall'input e mettilo nella cella 18 (c)
010000000010011	Leggi un valore dall'input e mettilo nella cella 19 (d)
000000000010010	Carica il contenuto della cella 18 (c) nel registro A
000100000010011	Carica il contenuto della cella 19 (d) nel registro B
011000000000000	Somma i registri A e B
000100000010100	Carica il contenuto della cella 20 (z) (ris. parziale) in B
100000000000000	Moltiplica i registri A e B
001000000010100	Scarica il contenuto di A nella cella 20 (z) (ris. totale)
010100000010100	Scrivi il contenuto della cella 20 (z) (ris. totale) output
110100000000000	Halt



Forma Binaria del Programma

```
010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) **output**

Halt



Forma Binaria del Programma

```
010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) output

Halt



Forma Binaria del Programma

```
010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) output

Halt



Forma Binaria del Programma

```
01000000000010000
010000000000010001
00000000000010000
00010000000010001
01100000000000000
00100000000010100
01000000000010010
01000000000010011
00000000000010010
00010000000010011
01100000000000000
00010000000010100
10000000000000000
00100000000010100
01010000000010100
11010000000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) **output**

Halt



Forma Binaria del Programma

```
010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)
Leggi un valore dall'input e mettilo nella cella 17 (b)
Carica il contenuto della cella 16 (a) **nel registro A**
Carica il contenuto della cella 17 (b) **nel registro B**
Somma i registri A e B
Scarica il contenuto di A nella cella 20 (z) (ris.parziale)
Leggi un valore dall'input e mettilo nella cella 18 (c)
Leggi un valore dall'input e mettilo nella cella 19 (d)
Carica il contenuto della cella 18 (c) **nel registro A**
Carica il contenuto della cella 19 (d) **nel registro B**
Somma i registri A e B
Carica il contenuto della cella 20 (z) (ris. parziale) **in B**
Moltiplica i registri A e B
Scarica il contenuto di A nella cella 20 (z) (ris. totale)
Scrivi il contenuto della cella 20 (z) (ris. totale) output
Halt



Forma Binaria del Programma

010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) **output**

Halt



Forma Binaria del Programma

```
010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000
```

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) **output**

Halt



Forma Binaria del Programma

010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
01010000000010100
110100000000000

Leggi un valore dall'input e mettilo nella cella 16 (a)
Leggi un valore dall'input e mettilo nella cella 17 (b)
Carica il contenuto della cella 16 (a) **nel registro A**
Carica il contenuto della cella 17 (b) **nel registro B**
Somma i registri A e B
Scarica il contenuto di A nella cella 20 (z) (ris.parziale)
Leggi un valore dall'input e mettilo nella cella 18 (c)
Leggi un valore dall'input e mettilo nella cella 19 (d)
Carica il contenuto della cella 18 (c) **nel registro A**
Carica il contenuto della cella 19 (d) **nel registro B**
Somma i registri A e B
Carica il contenuto della cella 20 (z) (ris. parziale) **in B**
Moltiplica i registri A e B
Scarica il contenuto di A nella cella 20 (z) (ris. totale)
Scrivi il contenuto della cella 20 (z) (ris. totale) output
Halt



Forma Binaria del Programma

010000000010000
010000000010001
000000000010000
000100000010001
011000000000000
001000000010100
010000000010010
010000000010011
000000000010010
000100000010011
011000000000000
000100000010100
100000000000000
001000000010100
010100000010100
110100000000000

Leggi un valore dall'input e mettilo nella cella 16 (a)

Leggi un valore dall'input e mettilo nella cella 17 (b)

Carica il contenuto della cella 16 (a) **nel registro A**

Carica il contenuto della cella 17 (b) **nel registro B**

Somma i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris.parziale)

Leggi un valore dall'input e mettilo nella cella 18 (c)

Leggi un valore dall'input e mettilo nella cella 19 (d)

Carica il contenuto della cella 18 (c) **nel registro A**

Carica il contenuto della cella 19 (d) **nel registro B**

Somma i registri A e B

Carica il contenuto della cella 20 (z) (ris. parziale) **in B**

Moltiplica i registri A e B

Scarica il contenuto di A nella cella 20 (z) (ris. totale)

Scrivi il contenuto della cella 20 (z) (ris. totale) **output**

Halt