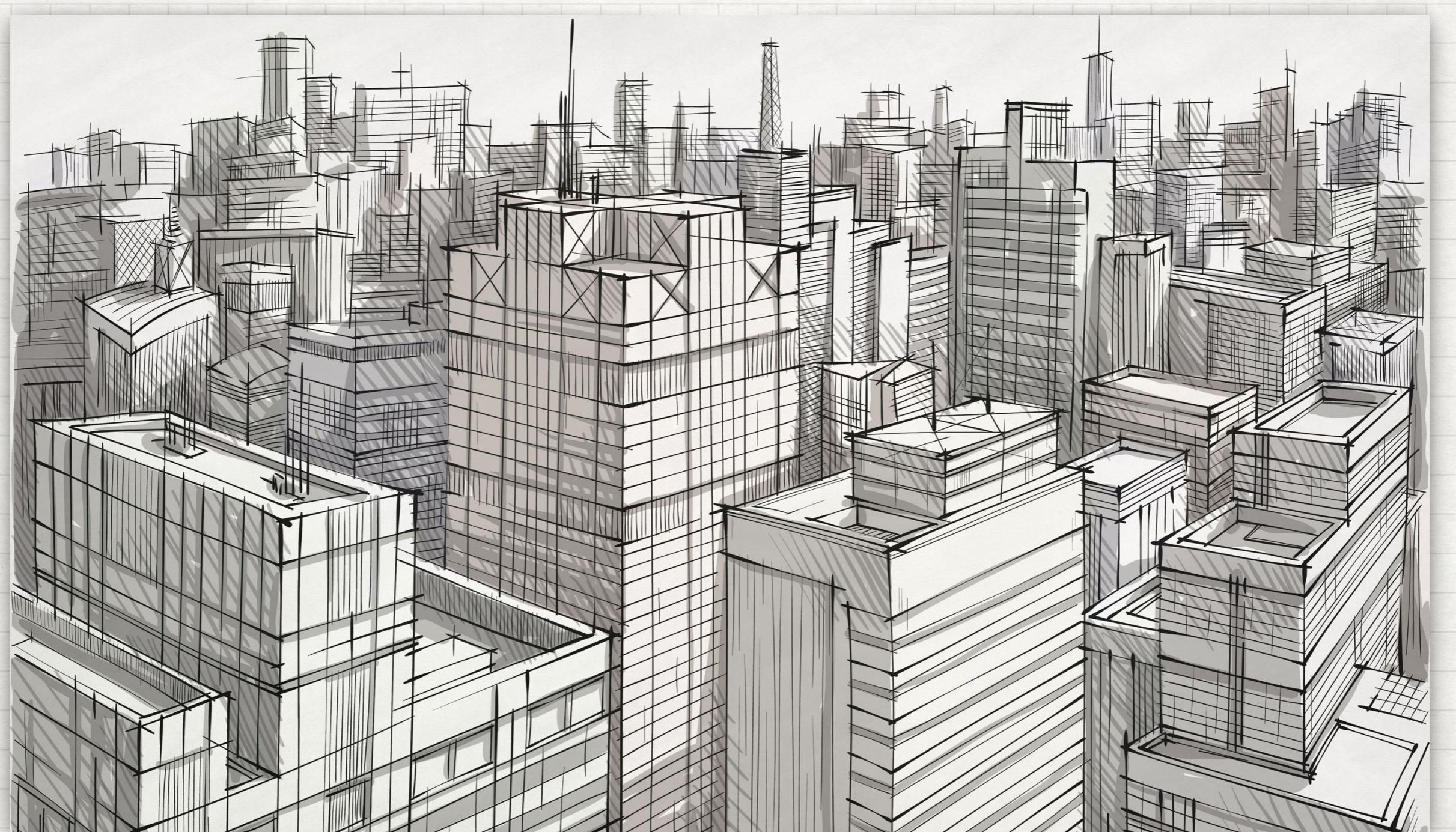


ELEMENTI DI INFORMATICA CDL ING. CIVILE A.A.2016/2017

# ARCHITETTURA DEL CALCOLATORE



Vittoria Nardone,

[vnardone@unisannio.it](mailto:vnardone@unisannio.it)

[vittoria.nardone@gmail.com](mailto:vittoria.nardone@gmail.com)

# ARCHITETTURA DEL CALCOLATORE

- Cosa è un Calcolatore?



- ❖ Strumento **programmabile** per rappresentare, memorizzare ed elaborare informazioni
- ❖ Un **sistema** costituito da molte componenti

- Cosa significa studiare l'architettura di un sistema?



- (1) Individuare ciascun componente del sistema
- (2) Comprendere i principi generali di funzionamento di ciascun componente
- (3) Comprendere come interagiscono le varie componenti

# MACRO-COMPONENTI

La prima decomposizione di un calcolatore è relativa a due macro componenti:

♦ **HARDWARE** →

LA STRUTTURA FISICA DI UN CALCOLATORE, FORMATA DA TUTTO CIÒ CHE PUÒ ESSERE CONSIDERATO COME PARTE MECCANICA, ELETTRICA ED ELETTRONICA DEL SISTEMA

♦ **SOFTWARE** →

PARTE "MORBIDA" (NON FISICA) DI UN CALCOLATORE CHE COMPRENDE I PROGRAMMI CHE CONSENTONO LA GESTIONE DEL CALCOLATORE E LA RISOLUZIONE DEI PROBLEMI PRESENTATI DALL'UTENTE

♦ A cosa serve un calcolatore? Un calcolatore effettua l'elaborazione automatica delle informazioni.

Per capire meglio la struttura fisica di un calcolatore è bene chiarire il concetto di Elaborazione Automatica delle Informazioni

# PRINCIPI DELL'ELABORAZIONE AUTOMATICA DELLE INFORMAZIONI

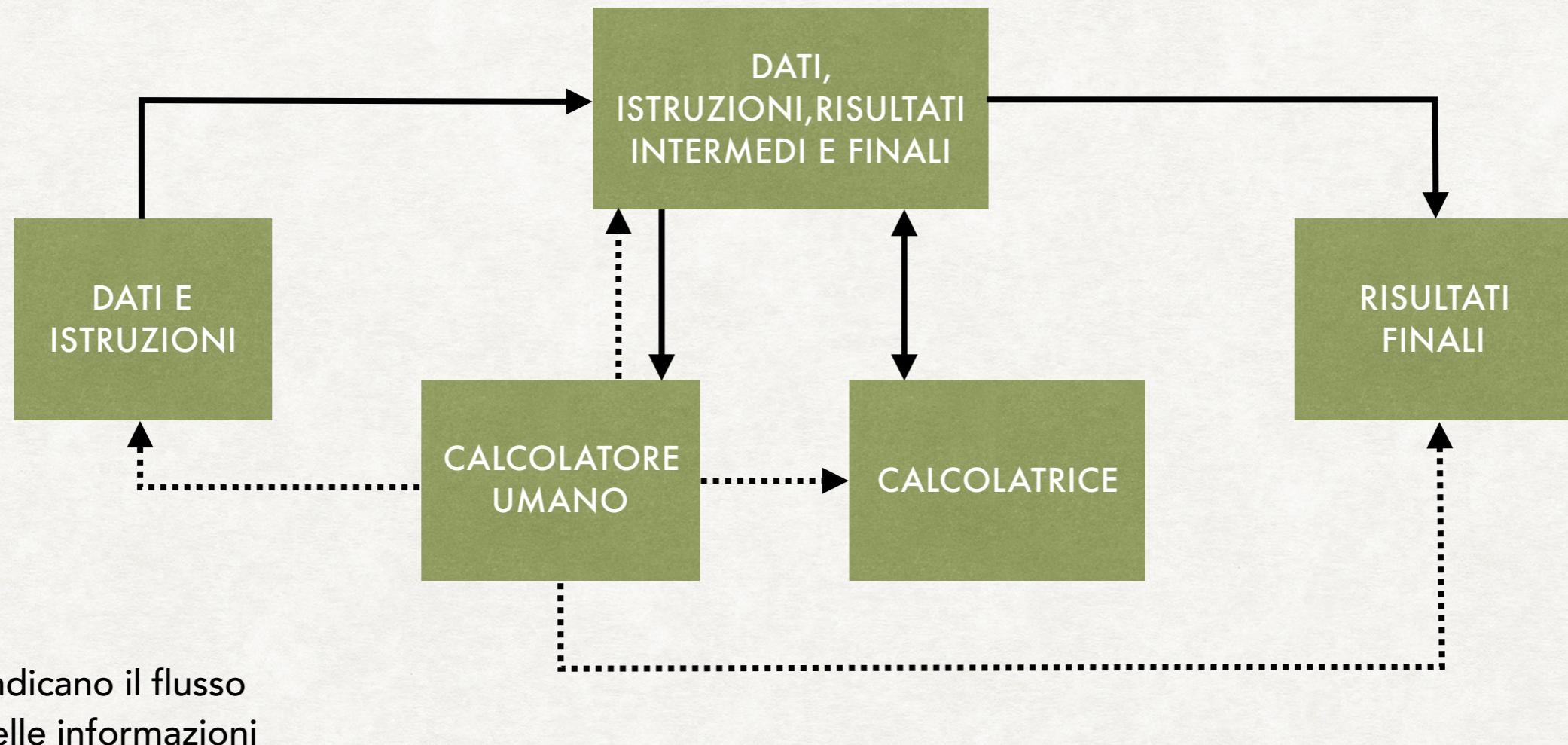
I Calcolatori sono macchine che simulano **comportamenti intelligenti** e sono utilizzati per risolvere la più ampia gamma di problemi

❖ **Esempio:** Risolvere un problema di matematica



- 1) Ricezione dei dati —> L'insegnante consegna allo studente (calcolatore umano) il compito con la traccia da svolgere
- 2) Trascrizione i dati e scrittura dei passi dell'algoritmo per svolgere l'esercizio (Memorizzazione)
- 3) Esecuzione dei passi dell'algoritmo nel giusto ordine per risolvere il problema, esecuzione di calcoli e possibile utilizzo di una calcolatrice. I risultati intermedi e quelli finali possono essere "memorizzati" sul foglio di carta
- 4) I risultati finali sono trascritti e consegnati all'insegnante (l'output è il risultato del problema risolto)

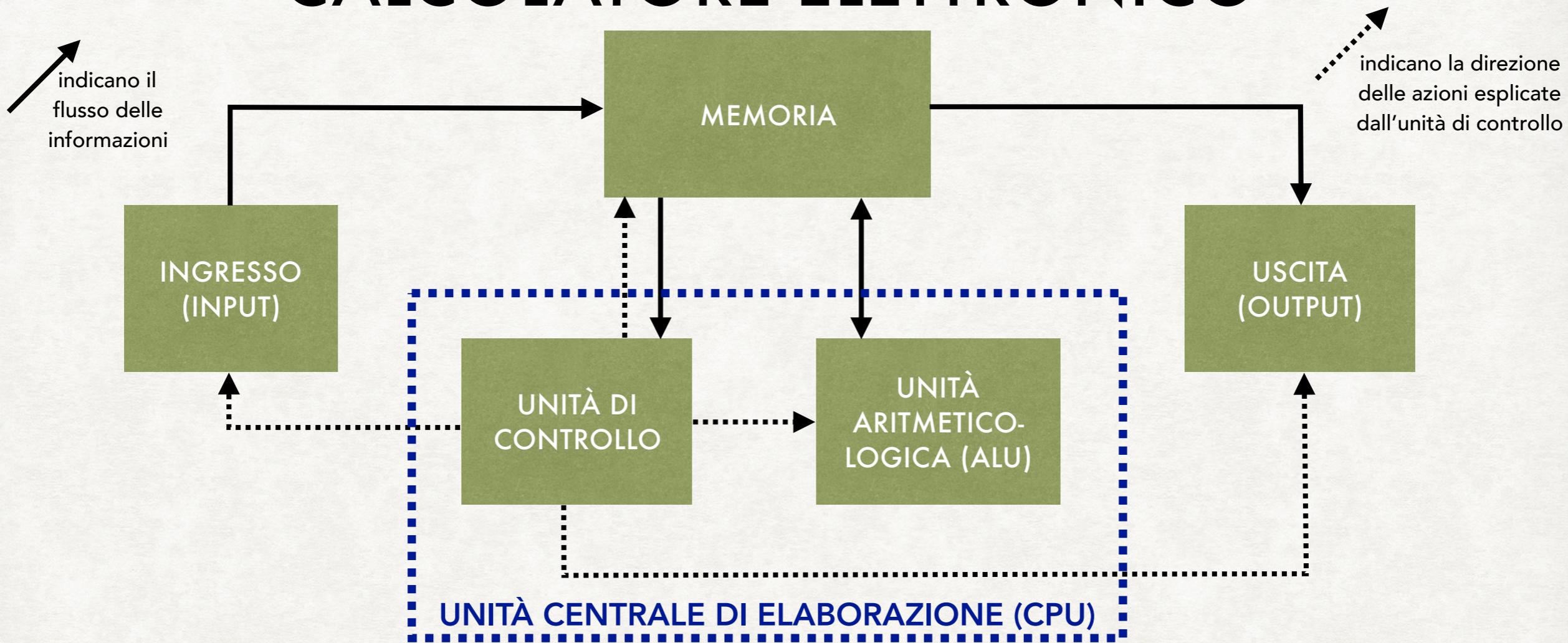
# SCHEMA DEL PROCESSO ORGANIZZATIVO DEL CALCOLATORE UMANO



indcano il flusso  
delle informazioni

indcano le azioni  
del calcolatore  
umano

# SCHEMA A BLOCCHI DELLE UNITÀ DI UN CALCOLATORE ELETTRONICO



- ❖ L'unità di controllo espleta le stesse funzioni del calcolatore umano:
  - ◆ Attiva l'unità di ingresso (o unità di input) attraverso la quale vengono introdotti i dati e le istruzioni dell'algoritmo.
    - \* La sequenza delle istruzioni prende il nome di **programma**.
  - ◆ Provvede alla memorizzazione dei dati e del programma in una unità detta **memoria**. Quest'ultima ha la stessa funzione del foglio di carta usato dal calcolatore umano.
  - ◆ Interpreta ogni istruzione, trasferisce i dati nell'**unità aritmetico-logica**, che ha lo stesso ruolo della calcolatrice.
  - ◆ Ordina a tale unità di eseguire sui dati le operazioni aritmetiche e logiche indicate da ogni istruzione presa in esame.
  - ◆ Provvede a memorizzare i risultati intermedi nella memoria per poi estrarli e trasferirli, al momento opportuno, nell'ALU per ulteriori elaborazioni.
  - ◆ Trasferisce i risultati finali prima nella memoria e poi nell'unità di uscita (o unità di output).

# SVILUPPO STORICO DELL'ELABORAZIONE DEI DATI

Abaco



Stchoty



Soroban

Telaio di Jacquard



- Il pallottoliere può essere considerato il più elementare calcolatore numerico. Utilizzato fin dall'antichità aiutava l'uomo nei suoi conti.
- Blaise Pascal nel 1642 costruì la prima macchina per il calcolo. La macchina poteva eseguire l'addizione e la sottrazione.
- Gottfried Leibniz nel 1671 costruì una macchina per il calcolo che era in grado di eseguire anche moltiplicazioni e divisioni.
- Joseph-Marie Jacquard nel 1800 inventò (in realtà rielaborò e perfezionò un'idea del tessitore Giovanni il Calabrese di metà '400) un metodo per l'automatizzazione dei telai per la tessitura, scrivendo un insieme di passi da far eseguire ai telai (programma) su nastri opportunamente perforati.
- Charles Babbage nel 1835 creò la macchina analitica. Per la prima volta vengono usate insieme macchine a ruote a riporto automatico (pascalina) e le schede come quelle di Jacquard.
- In questo periodo però la tecnologia non era in grado di supportare adeguatamente le idee degli inventori. Bisogna aspettare i primi del '900 per vedere diffuse le prime macchine calcolatrici.

Pascalina



Macchina Analitica

# SVILUPPO STORICO DELL'ELABORAZIONE DEI DATI (2)



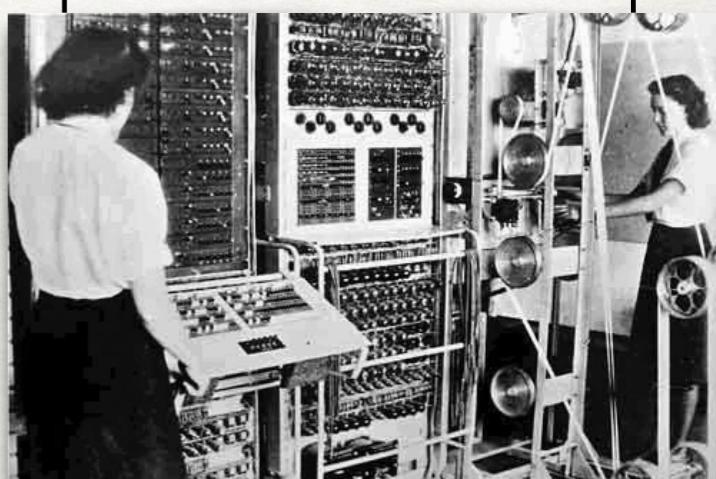
Machina Tabulatrice

- USA 1880 censimento, risultati del censimento ottenuti dopo 7 anni. Per il censimento del 1890 **Herman Hollerith** costruì la Macchina Tabulatrice, che aveva alla base l'idea delle schede perforate di Babbage. Hollerith, infatti, si rese conto che perforando in modo opportuno delle schede avrebbe potuto memorizzare più facilmente i dati relativi alle persone e, con opportune macchine, anche contarli. Grazie a questa intuizione i dati del censimento vennero elaborati in solo due anni e mezzo.
- Nel 1895 Hollerith fondò una società in grado di produrre calcolatori anche per applicazioni di tipo contabile. La società si sviluppò fino a prendere il nome di **IBM** (International Business Machines) nel 1924.
- Le macchine allora erano di tipo meccanico, o al più elettromagnetico, e venivano chiamati **calcolatori meccanografici**. Questi calcolatori funzionavano egregiamente per l'elaborazione dei dati contabili mentre per i calcoli scientifici vi era la necessità di sistemi più potenti. I primi prototipi di calcolatori elettronici, capaci di eseguire operazioni complicate in pochi secondi, compariranno solo alla fine degli anni trenta, sulla spinta delle necessità della seconda guerra mondiale.

## Konrad Zuse



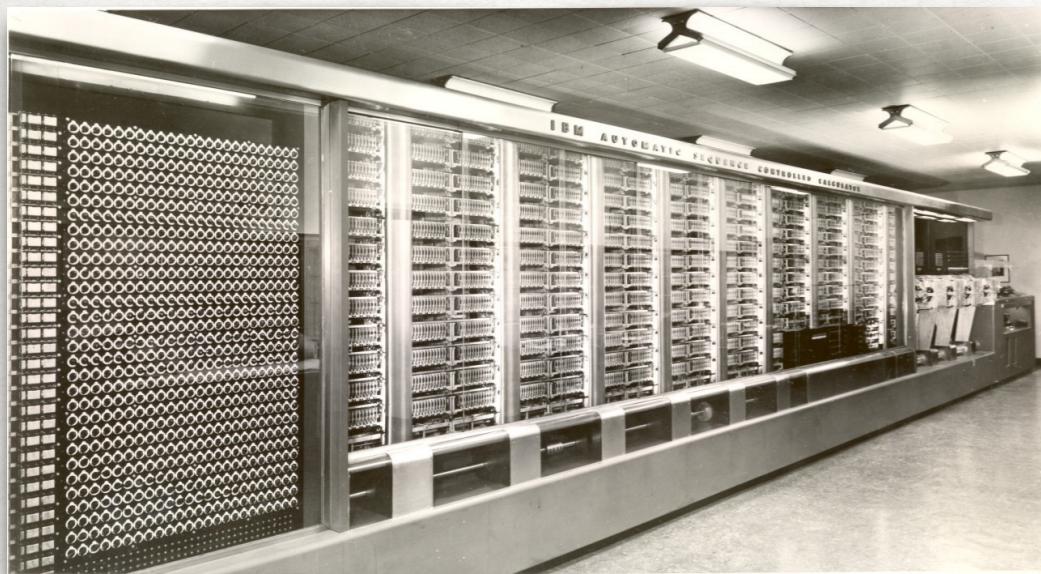
- Il tedesco **Konrad Zuse** nel 1938 realizza lo **Z1** e nel 1941 lo **Z3** per poter effettuare calcoli per il progetto di aerei usando alcuni relè.
- Questi calcolatori vennero distrutti nei bombardamenti di Berlino.
- In Gran Bretagna nel 1943 si realizza **Colossus** (di cui si scoprirà l'esistenza solo nel 1975) per decrittografare i messaggi in codice dei tedeschi.



Colossus

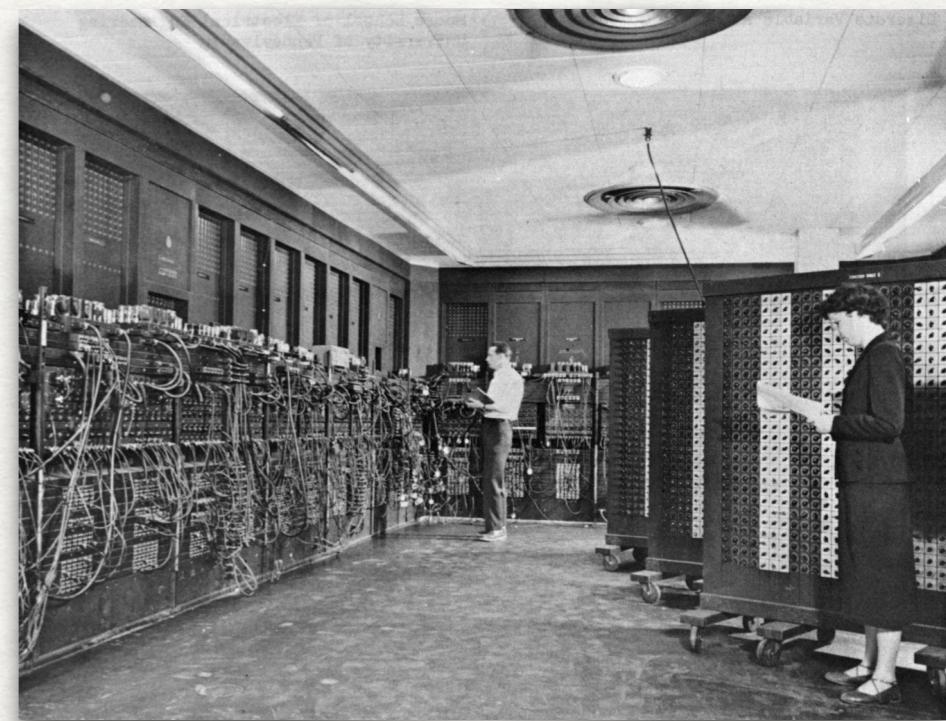
# SVILUPPO STORICO DELL'ELABORAZIONE DEI DATI (3)

Harvard Mark I



- All'università di Harvard nel 1944 viene realizzato Mark I che riprende le idee di Babbage. Era composto da 78 calcolatrici, 800 km di cavi, 3300 relè, era controllato da un programma perforato su un nastro di carta e forniva i risultati utilizzando di macchine per scrivere elettriche. Era in grado di sommare due numeri di 23 cifre in tre decimi di secondo e di moltiplicarli in 6 secondi.
- Viene considerato il capostipite dei moderni calcolatori poiché fu la prima macchina in grado di effettuare da sola calcoli una volta fornita dell'opportuno programma e dei dati.

- Come primo vero elaboratore elettronico si considera l'**ENIAC** (Electronic Numerical Integrator And Computer), nato nel 1946 per risolvere problemi di calcolo balistico e usato in seguito per altri scopi. Un computer quasi completamente elettronico che però era molto grande (30 tonnellate di peso, lungo 30 metri, alto 3 e profondo 1, occupava una superficie di 180 metri quadri), e le componenti (fili, valvole, spinotti) andavano distanziate tra loro per permettere la dissipazione del calore. Era molto rudimentale anche se molto veloce.

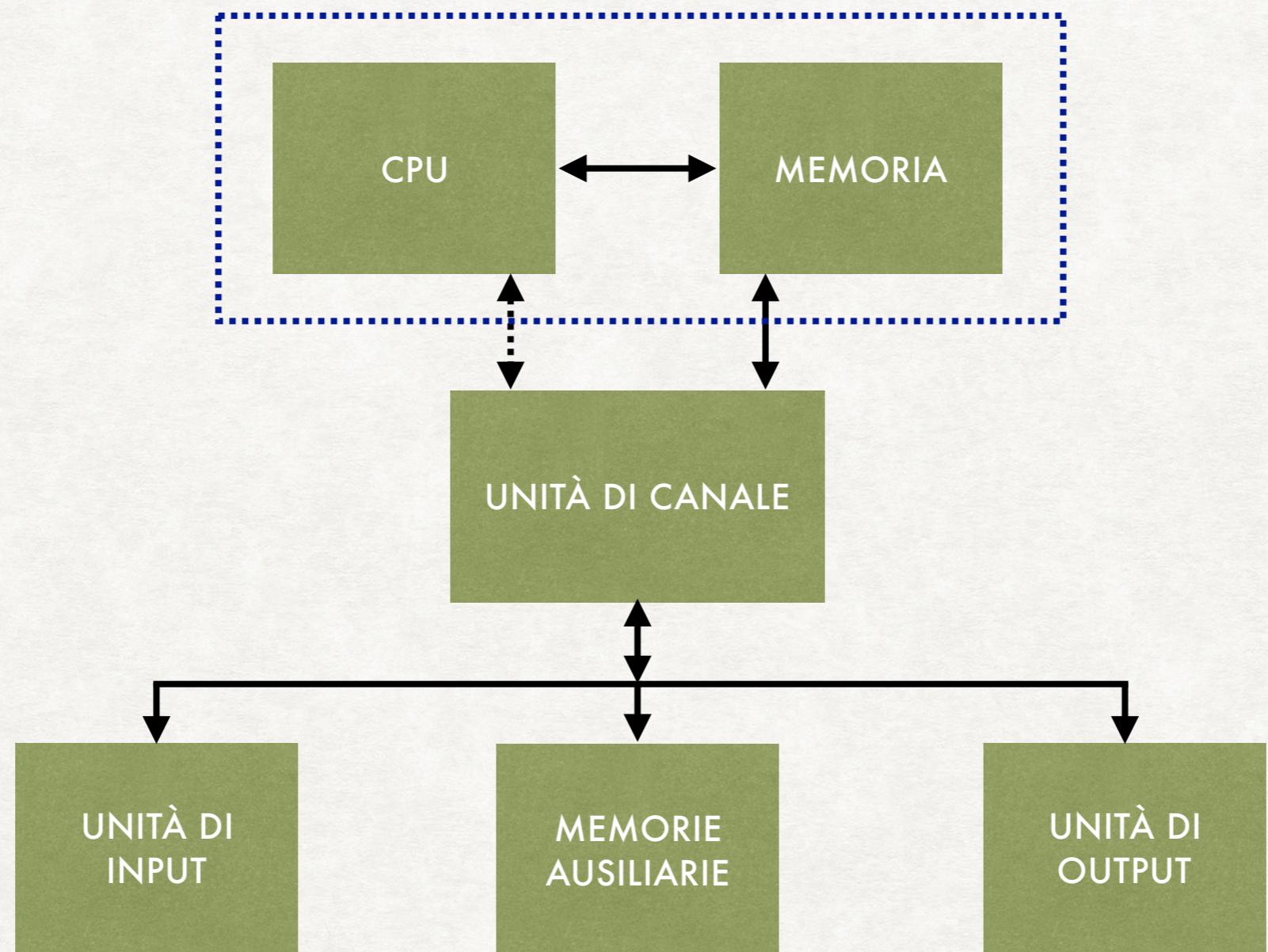


ENIAC

- Solo con l'**EDVAC** (Electronic Discrete Variable Automatic Computer) si riuscì a memorizzare sia dati sia programmi sulle schede per fornirli in input ai calcolatori. Progettato nel 1945 da **Von Neumann**, entra in funzione solo nel 1952, ma tutti i successivi elaboratori, compresi quelli attuali, si baseranno su questo tipo di architettura.

# ARCHITETTURA DI VON NEUMANN

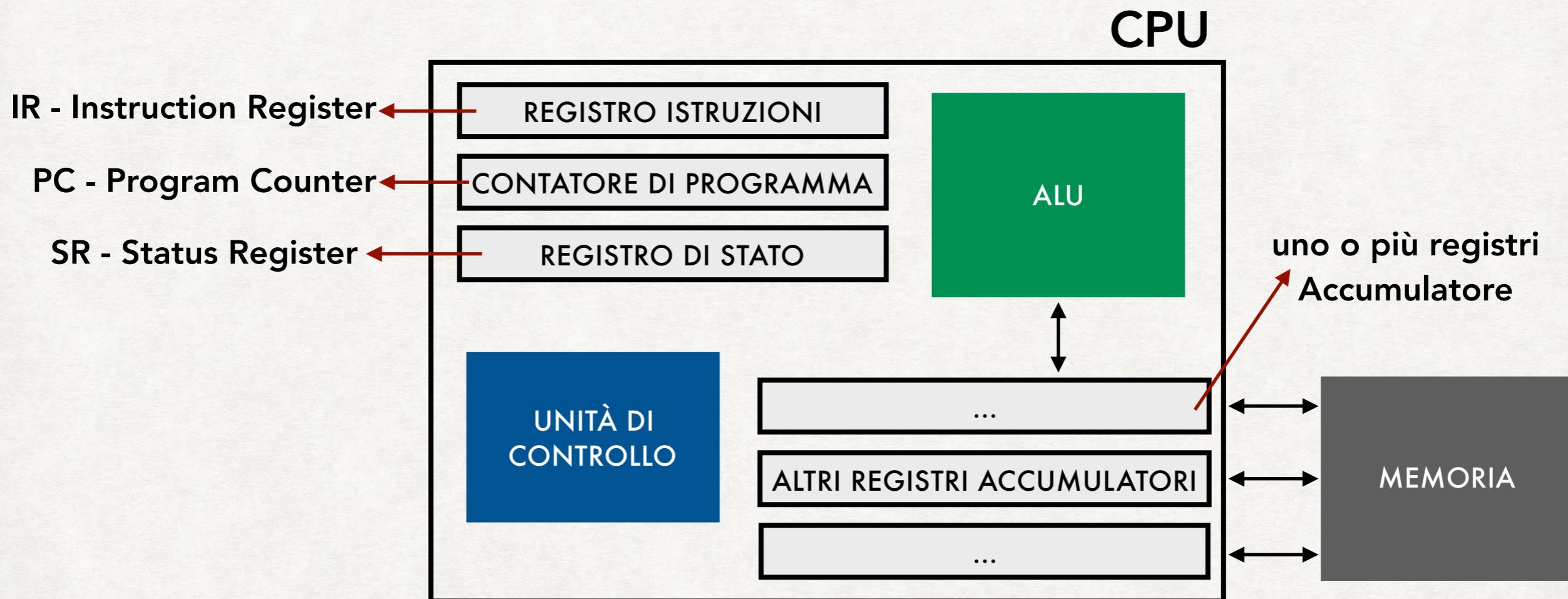
## LOGICA CABLATA



# UNITÀ CENTRALE DI ELABORAZIONE

## CENTRAL PROCESSING UNIT (CPU)

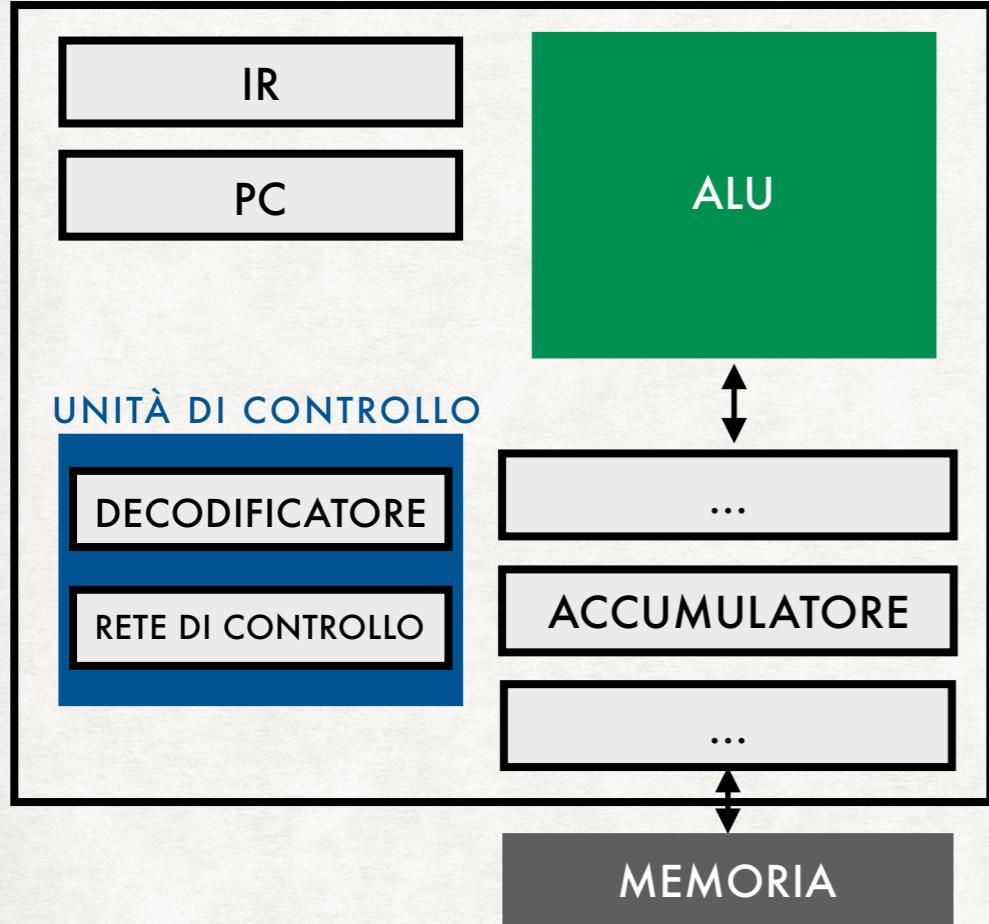
- L'**unità centrale di elaborazione** (CPU) o **processore** presiede l'esecuzione di un programma
  - \* Esegue istruzioni per l'elaborazione dei dati
- Svolge anche funzioni di controllo



# UNITÀ CENTRALE DI ELABORAZIONE (2)

## CENTRAL PROCESSING UNIT (CPU)

CPU



- La CPU è composta da:

- ◆ unità di controllo
- ◆ unità aritmetico-logica (ALU)
- ◆ alcuni registri (IR,PC,SR,Accumulatore)

- Registro Istruzioni (IR - Instruction Register), che contiene l'istruzione da eseguire.
- Contatore di Programma (PC - Program Counter), che punta alla prossima istruzione da eseguire, ovvero l'istruzione da eseguire subito dopo quella in atto.
- Registro di Stato (ST - Status Register), che contiene le informazioni relative agli stati che la CPU attraversa per eseguire ogni istruzione.

- Uno o più registri operativi (Accumulatori), che supportano il flusso delle informazioni da e verso la memoria e sono il riferimento per tutte le operazioni eseguite dall'ALU.
- L'Unità Aritmetico-Logica (Arithmetic Logic Unit, ALU) è un dispositivo capace di eseguire le operazioni aritmetiche, logiche e di confronto su dati prelevati dalla memoria. L'ALU è una calcolatrice automatica a cui, di volta in volta, vanno forniti gli operandi e il codice dell'operazione da effettuare.
- L'Unità di Controllo (opp. Unità di Governo) è un dispositivo che controlla e gestisce tutte le altre unità del calcolatore durante l'esecuzione di un programma e più precisamente è formata da:
  - ❖ un decodificatore, in grado di interpretare il codice dell'istruzione;
  - ❖ una rete di controllo, che genera una successione di comandi relativi all'istruzione considerata, in base all'uscita del decodificatore.

# LE ISTRUZIONI

Per comprendere il funzionamento della CPU è necessario dapprima conoscere la struttura delle **istruzioni**.

Ogni volta che un calcolatore deve svolgere un algoritmo è necessario preparare un **programma**, cioè una successione di operazioni elementari dette istruzioni che descrivono l'algoritmo stesso.

Ogni istruzione si compone di due parti:

- 1) **parte funzione** che stabilisce quale operazione deve essere eseguita dal calcolatore
- 2) **parte indirizzi** che contiene gli indirizzi dei dati (operandi) su cui opera l'istruzione stessa

Esempio di <u>Istruzione</u> :	<b>codice operativo</b>	<b>indirizzo - operando</b>	<b>indirizzo - operando</b>
--------------------------------	-------------------------	-----------------------------	-----------------------------

Nel dettaglio i campi di un'istruzione sono:

- un campo **codice operativo**, che identifica il tipo di operazione che deve essere eseguita
- un campo **indirizzo/operando**, che contiene i dati o gli indirizzi dei dati o, ancora, i registri sui quali sarà eseguita l'operazione descritta dal codice operativo o, ancora, riferimenti a registri. Il numero di campi indirizzo/operando è variabile, ma in generale non supera mai tre. In assenza di operandi, l'istruzione sarà composta, naturalmente, dal solo codice operativo. Nel caso di tre operandi, invece, il terzo indica, normalmente, dove porre il risultato.

# LE ISTRUZIONI (2)

Analizziamo un esempio di istruzione a tre operandi:

codice operativo	indirizzo - operando	indirizzo - operando	indirizzo - operando
------------------	----------------------	----------------------	----------------------

Supponiamo che l'istruzione da eseguire sia quella di sommare i valori A e B e di memorizzare il risultato nella variabile C:

SOMMA	INDIRIZZO di A	INDIRIZZO di B	INDIRIZZO di C
-------	----------------	----------------	----------------

I dati relativi alle variabili A e B sono contenuti in memoria centrale in apposite celle (INDIRIZZO di A ed INDIRIZZO di B)

SOMMA	1010	1011	1101
-------	------	------	------

L'istruzione di sopra può essere letta nel seguente modo: *somma il dato contenuto nella cella di memoria di indirizzo 1010 al dato contenuto nella cella di indirizzo 1011 e poni il risultato nella cella di memoria 1101.* Se immaginiamo che il **codice operativo** della somma sia **0101**, l'istruzione macchina diventa la seguente:

0101	1010	1011	1101
------	------	------	------

L'istruzione è espressa utilizzando il codice binario, ovvero è scritta in **linguaggio macchina**.

# CICLO DI ESECUZIONE

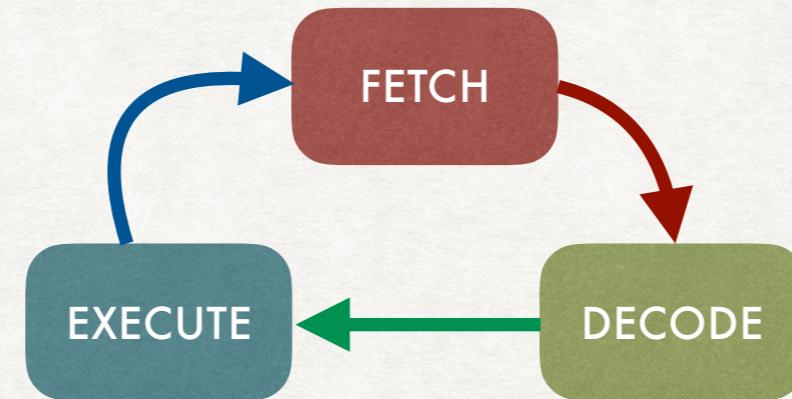
Per eseguire un'istruzione, scritta in linguaggio macchina, la CPU esegue un insieme di operazioni.

Più in dettaglio, l'elaborazione delle istruzioni di un programma consiste nell'esecuzione, da parte dell'unità di controllo, di una precisa successione di passi elementari. Precisamente:

- ❖ prelevamento dalla memoria centrale dell'istruzione da eseguire (fase di **FETCH**)
- ❖ interpretazione (decodifica) dell'istruzione (fase di **DECODE**)
- ❖ esecuzione dell'istruzione (fase di **EXECUTE**)

L'intero procedimento viene chiamato **CICLO di ESECUZIONE delle ISTRUZIONI**, ed è costituito da più microoperazioni.

Un programma, quindi, è eseguito applicando ad ogni istruzione la sequenza di **fetch-decode-execute**.



L'esecuzione delle microoperazioni avviene per passi detti *cicli macchina*.

Il ciclo macchina è una sequenza di operazioni elementari che l'unità di controllo deve eseguire ogni volta che deve accedere alla memoria o alle unità di I/O (input/output).

# CICLO DI ESECUZIONE

## UN ESEMPIO

Per semplicità si suppone che il calcolatore sia ad un indirizzo e quindi le funzioni ad accumulatore e debba eseguire la somma di due dati  $a$  e  $b$ .

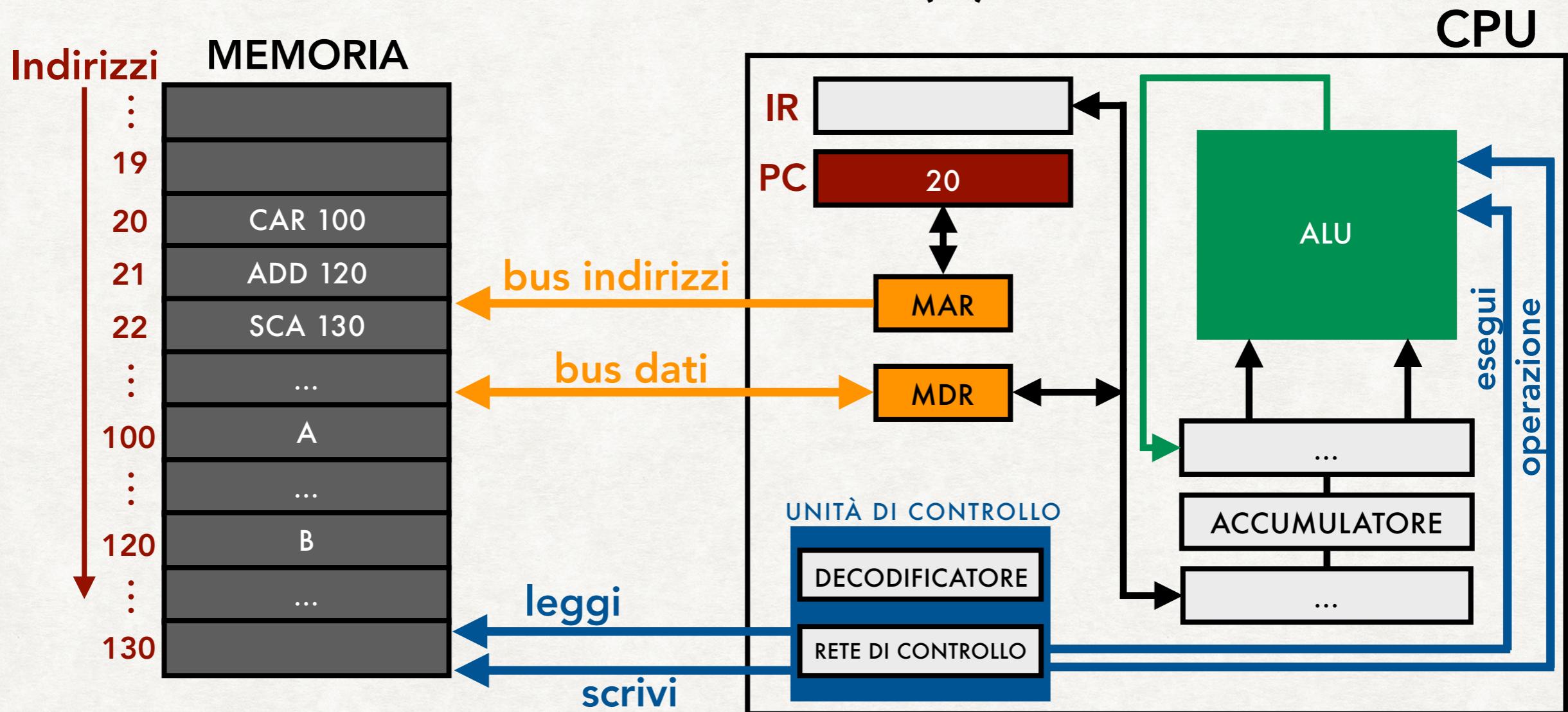
La macchina deve eseguire il seguente programma:

Indirizzo istruzione	Codice operativo	Indirizzo operandi	Descrizione
20	<b>CAR</b>	100	Preleva dalla locazione di memoria 100 il primo operando e caricalo nell'accumulatore
21	<b>ADD</b>	120	Addiziona il secondo operando, memorizzato nella posizione 120, con il contenuto dell'accumulatore e conserva il risultato in accumulatore
22	<b>SCA</b>	130	Trasferisci il risultato dell'accumulatore nella posizione di memoria 130

Quando si dà il segnale di inizio di programma l'indirizzo 20 della prima istruzione viene memorizzato nel registro contatore di programma (PC - Program Counter) della CPU. A questo punto ha inizio l'esecuzione della prima istruzione. Tale esecuzione avverrà in tre cicli.

# CICLO DI ESECUZIONE

## UN ESEMPIO (2)

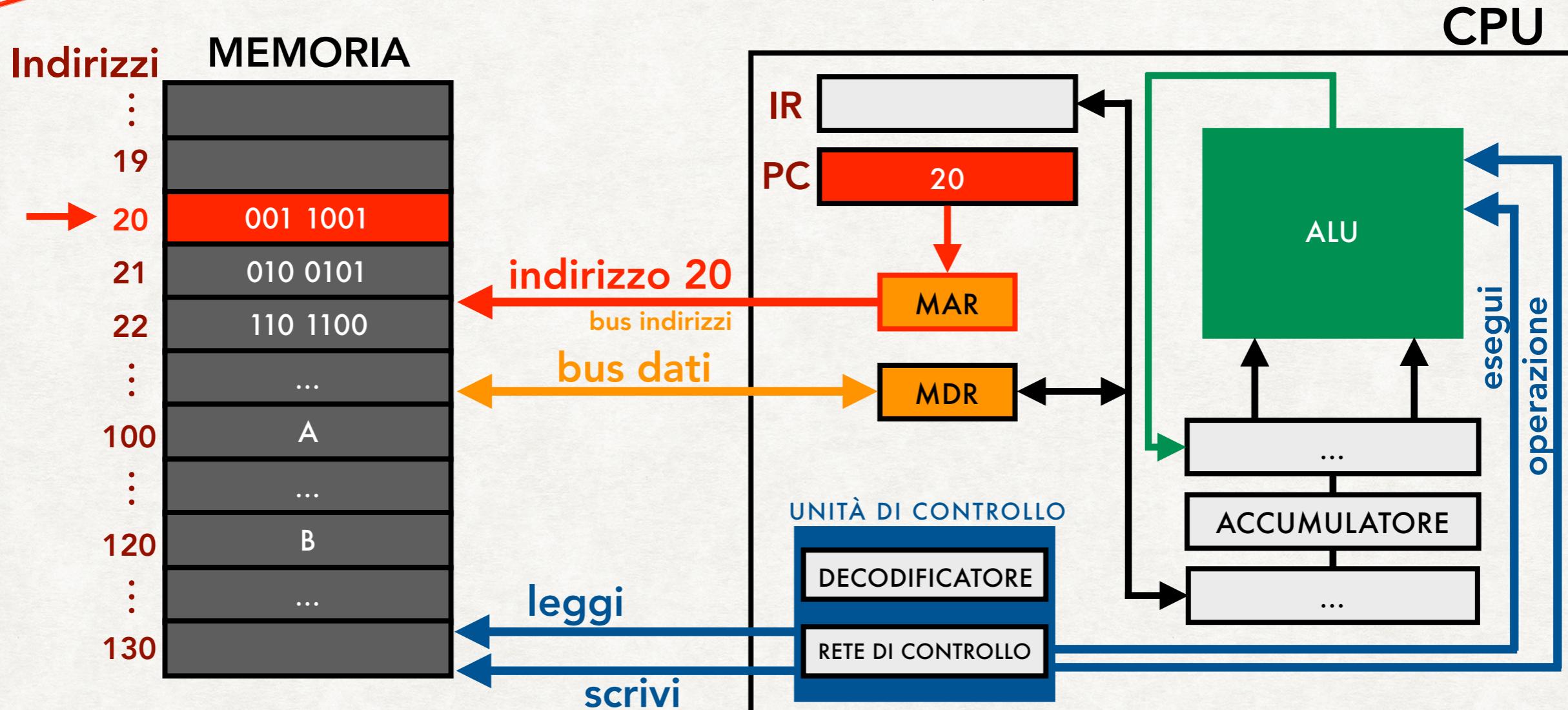


Nel registro PC (Program Counter) c'è l'indirizzo di memoria 20, ovvero l'indirizzo della prossima istruzione che dovrà essere eseguita.

- ❖ **MAR** - Registro indirizzi della memoria (Memory Address Register) contiene l'indirizzo della memoria che deve essere acceduta o memorizzata.
- ❖ **MDR** - Registro dati della memoria (Memory Data Register) contiene il dato che è stato acceduto oppure che deve essere memorizzato.

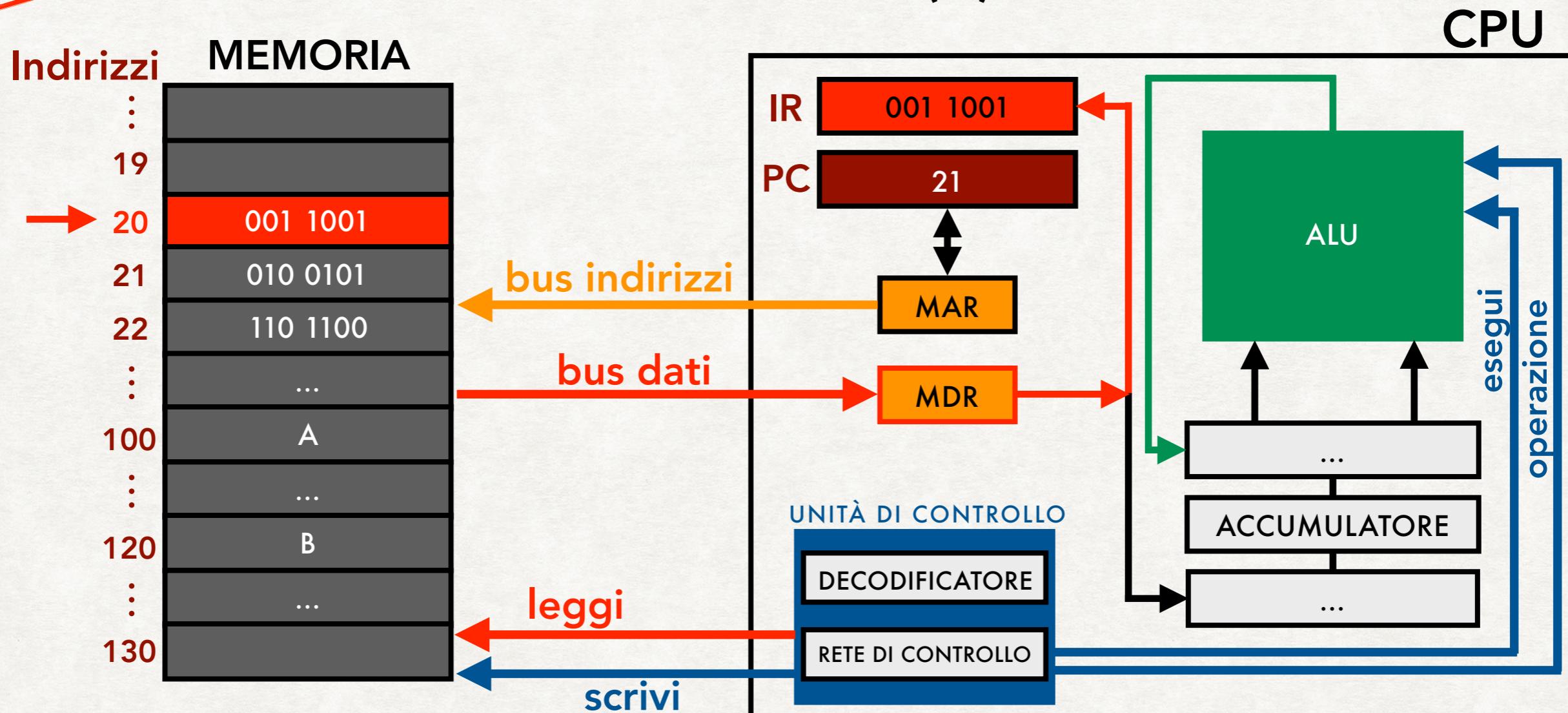
**FASE FETCH**

# CICLO DI ESECUZIONE UN ESEMPIO (3)



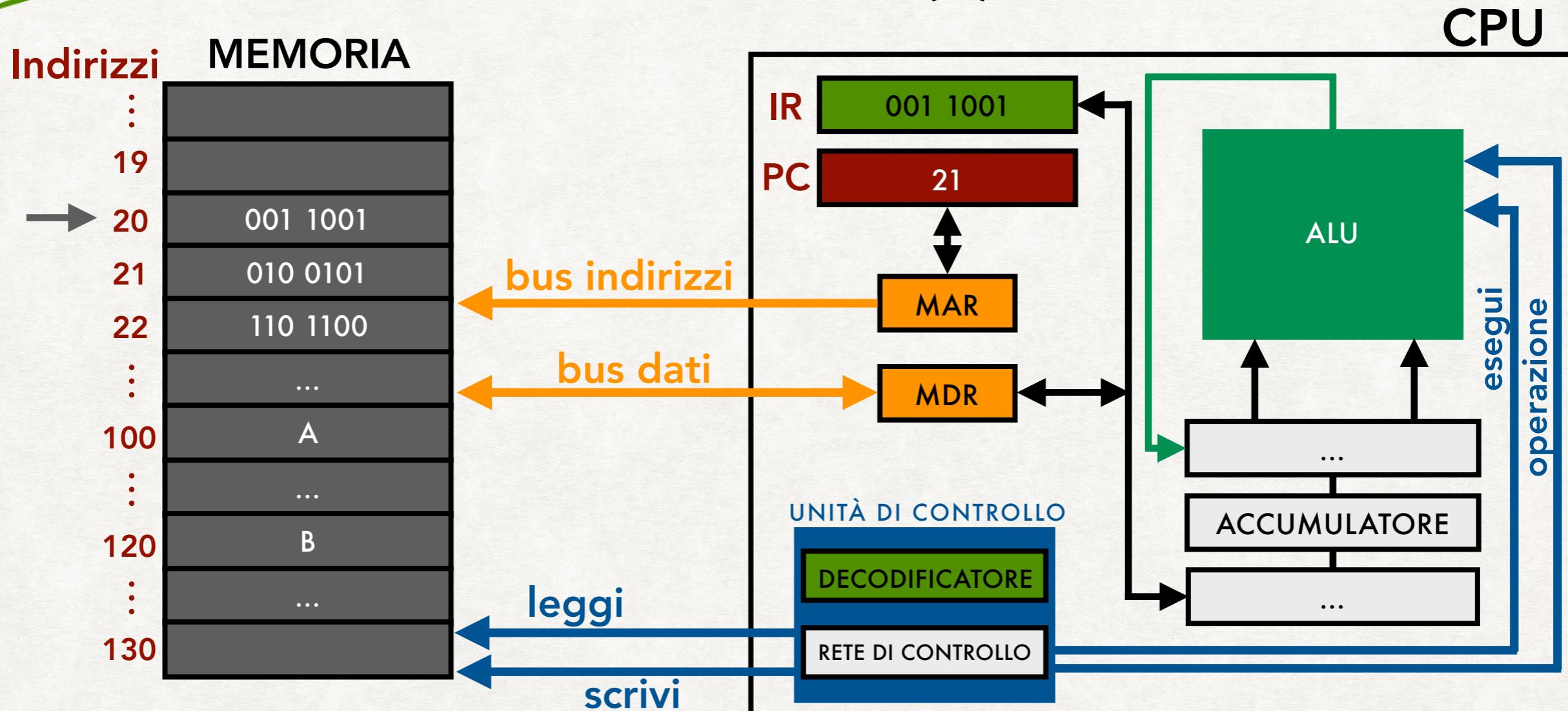
**FASE FETCH**

# CICLO DI ESECUZIONE UN ESEMPIO (4)



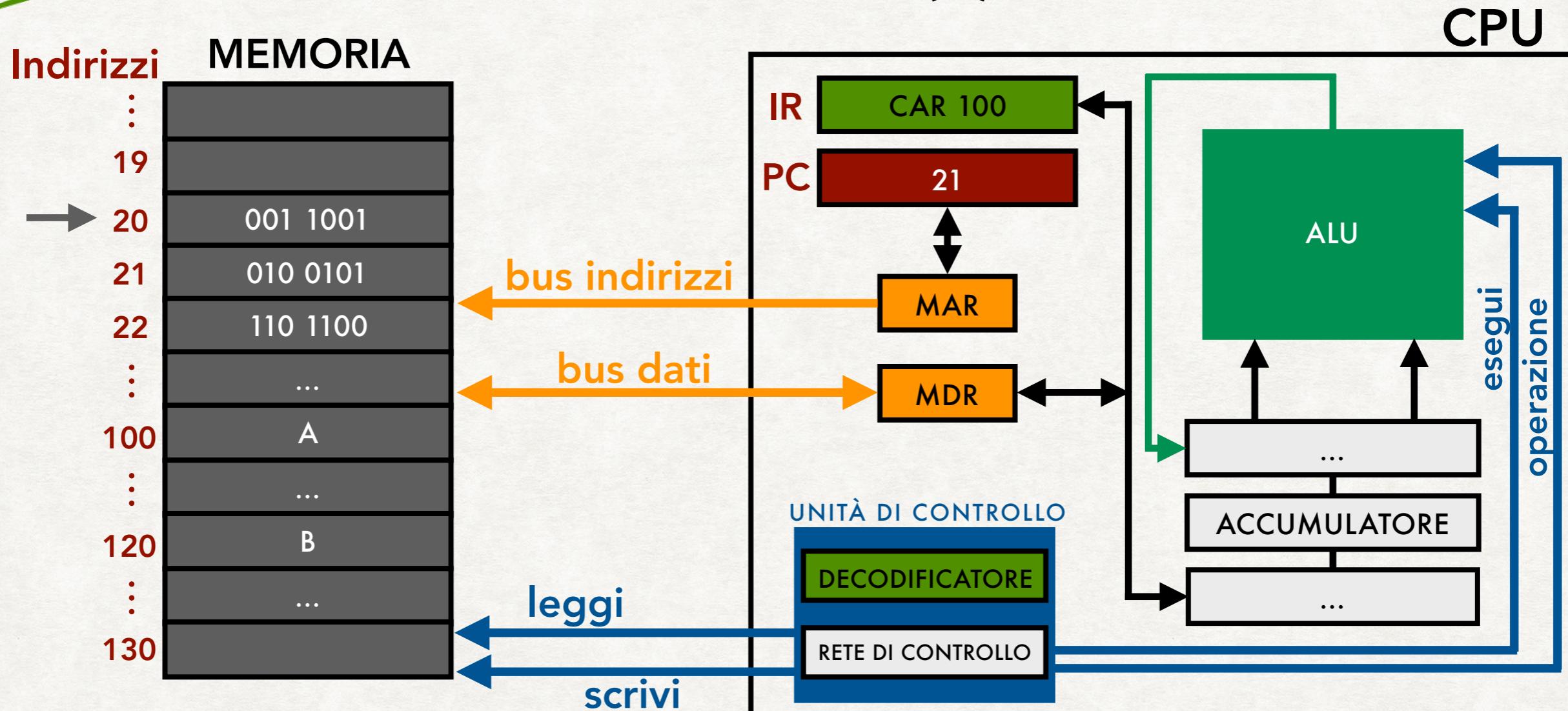
**FASE DECODE**

# CICLO DI ESECUZIONE UN ESEMPIO (5)



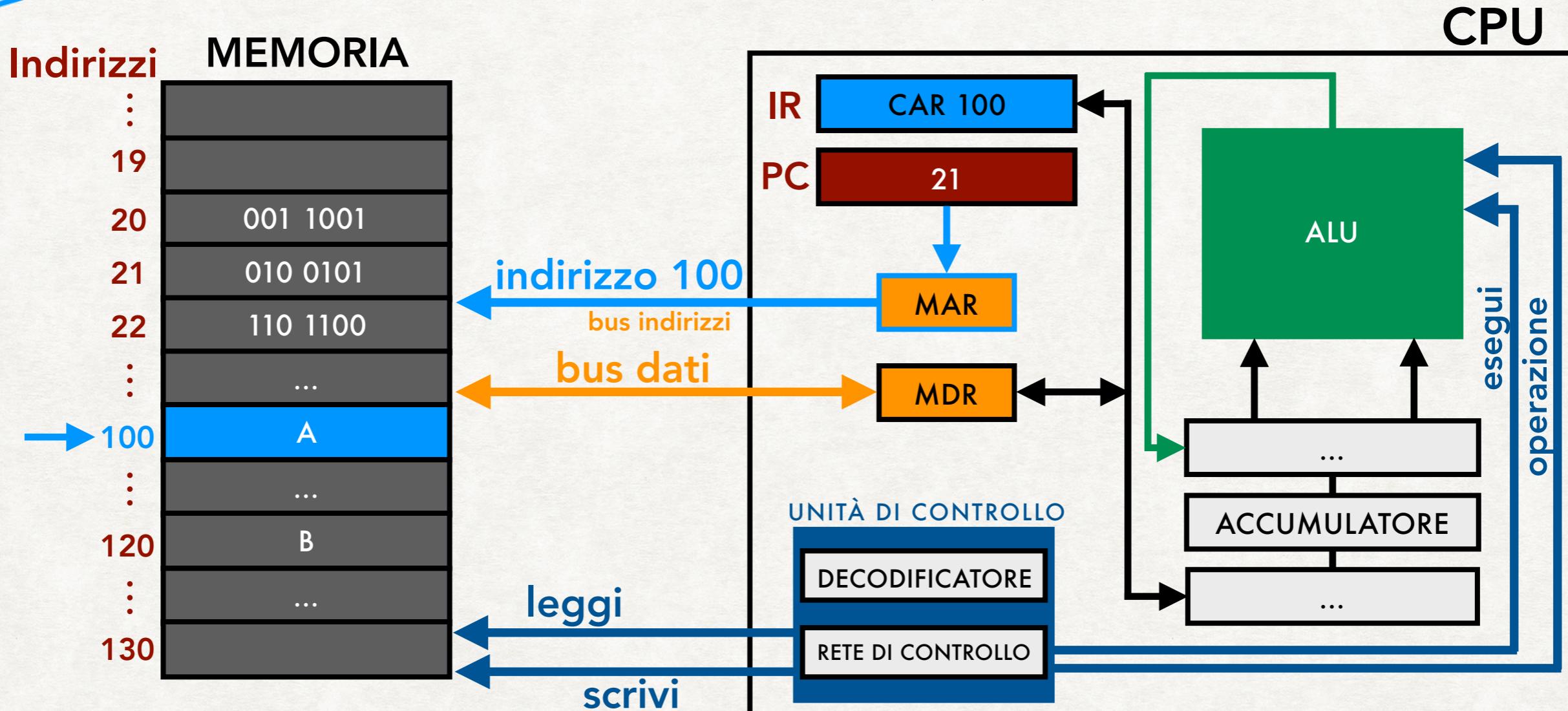
**FASE DECODE**

# CICLO DI ESECUZIONE UN ESEMPIO (6)



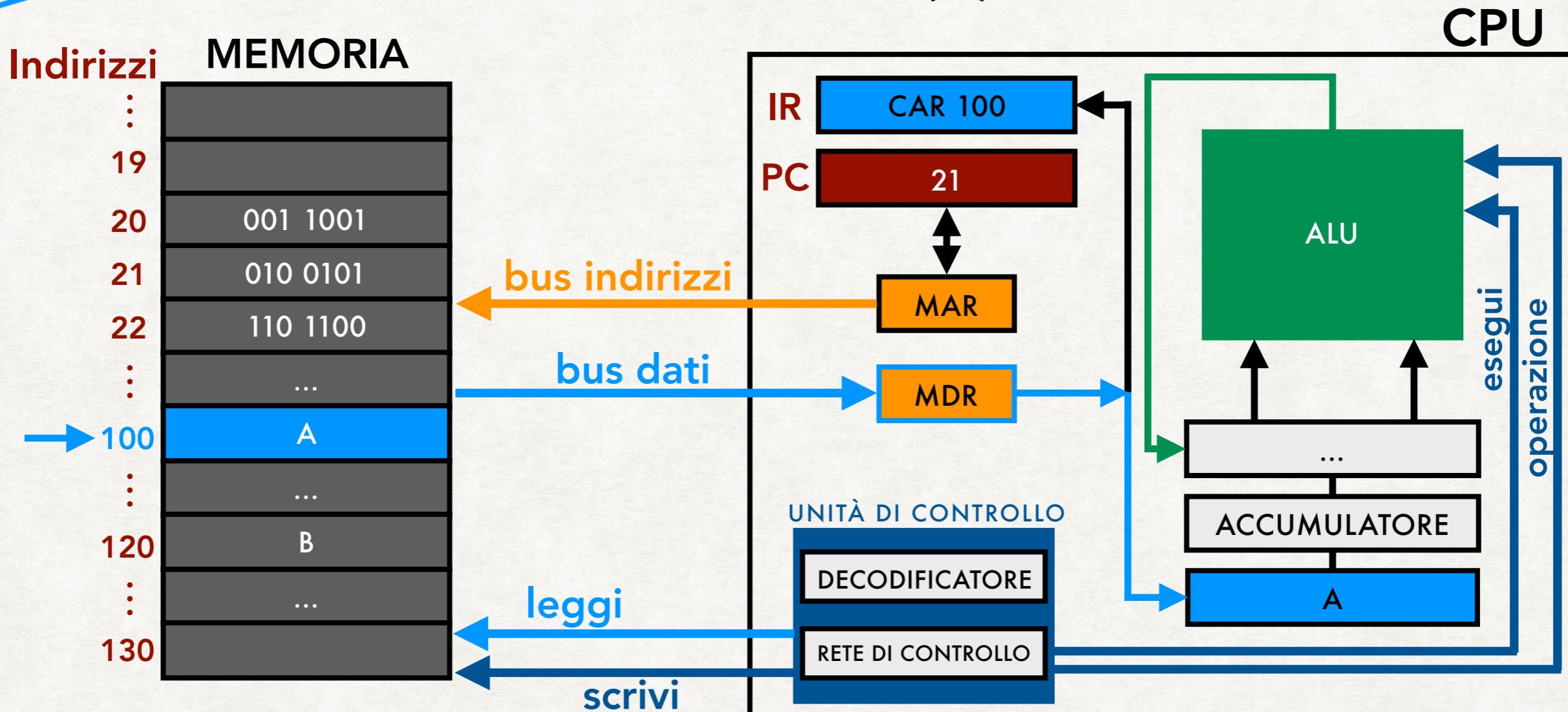
FASE EXECUTE

# CICLO DI ESECUZIONE UN ESEMPIO (7)



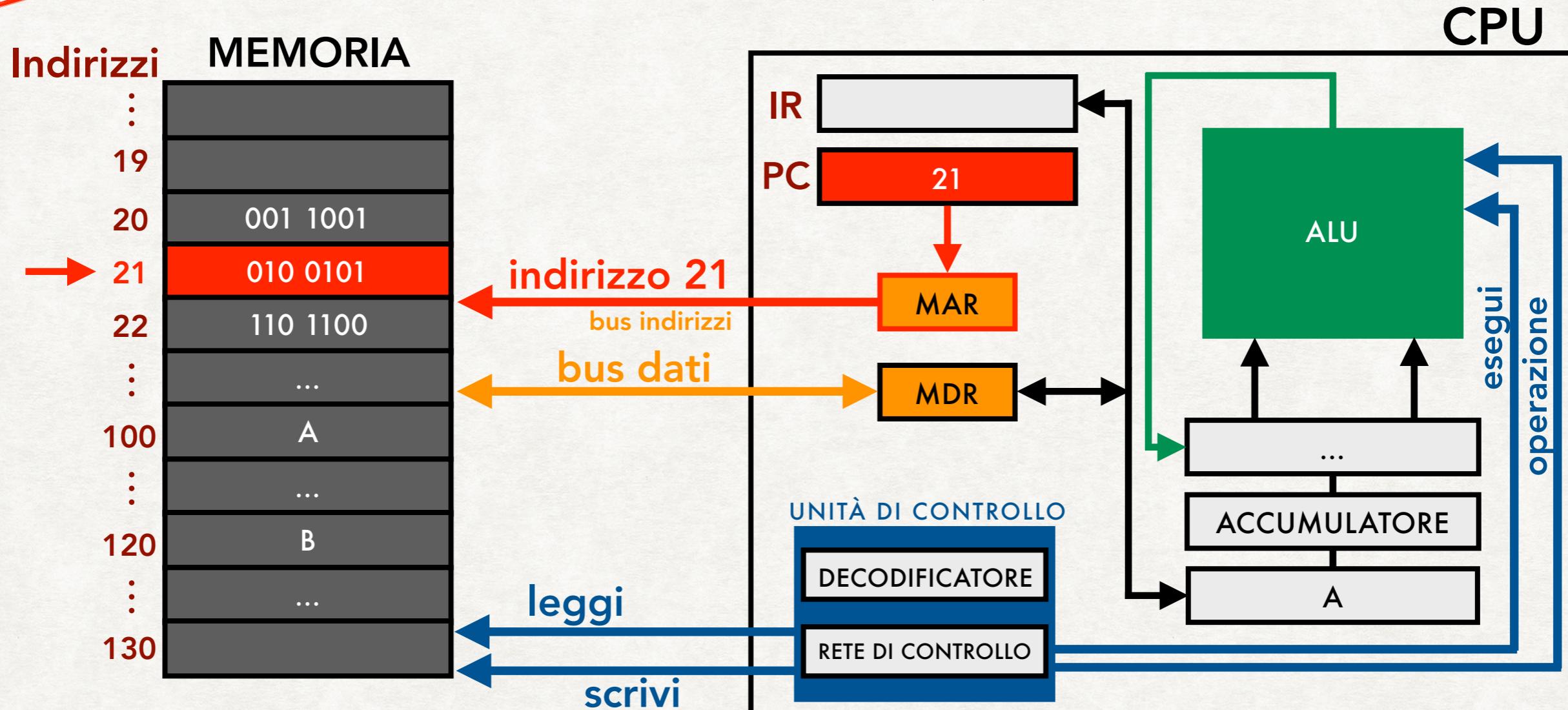
FASE EXECUTE

# CICLO DI ESECUZIONE UN ESEMPIO (8)



**FASE FETCH**

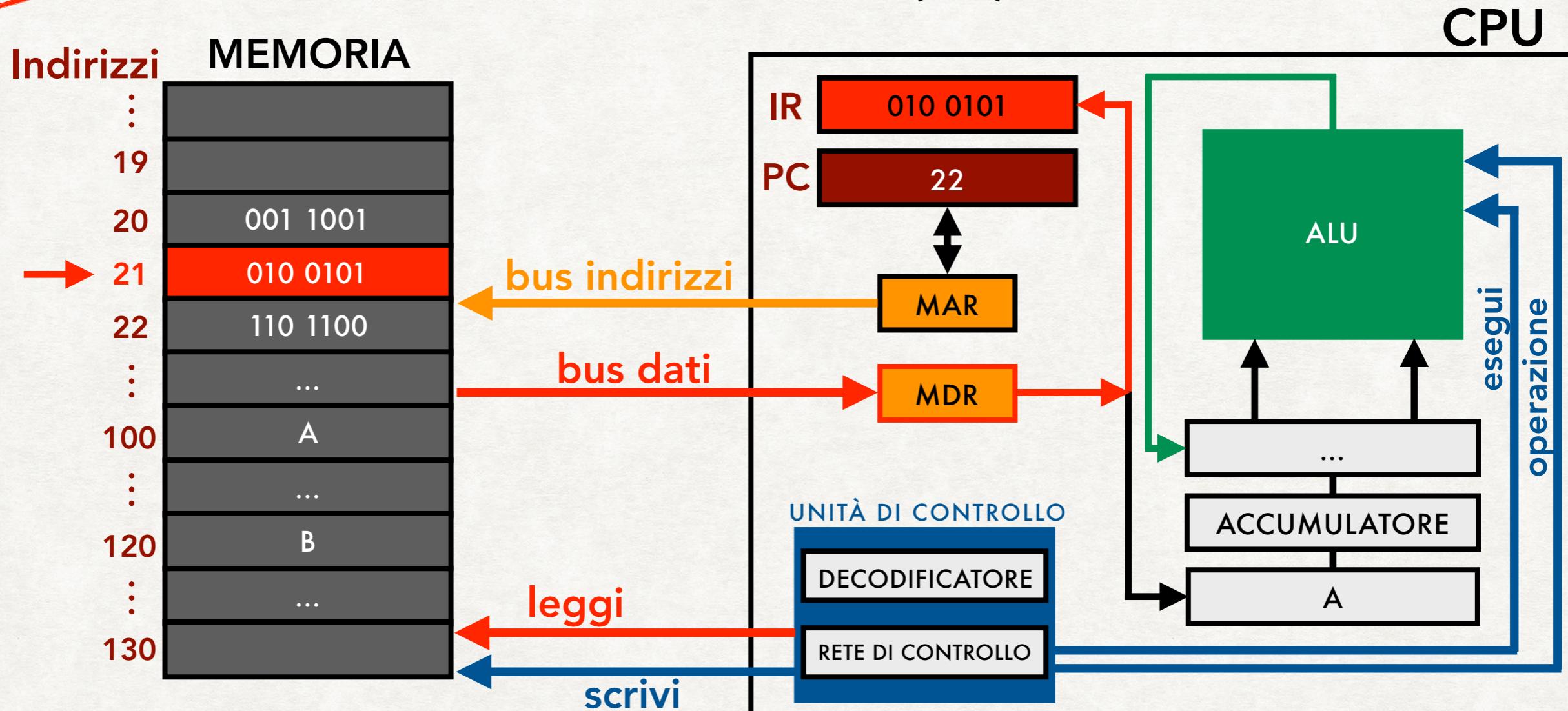
# CICLO DI ESECUZIONE UN ESEMPIO (9)



**FASE FETCH**

# CICLO DI ESECUZIONE

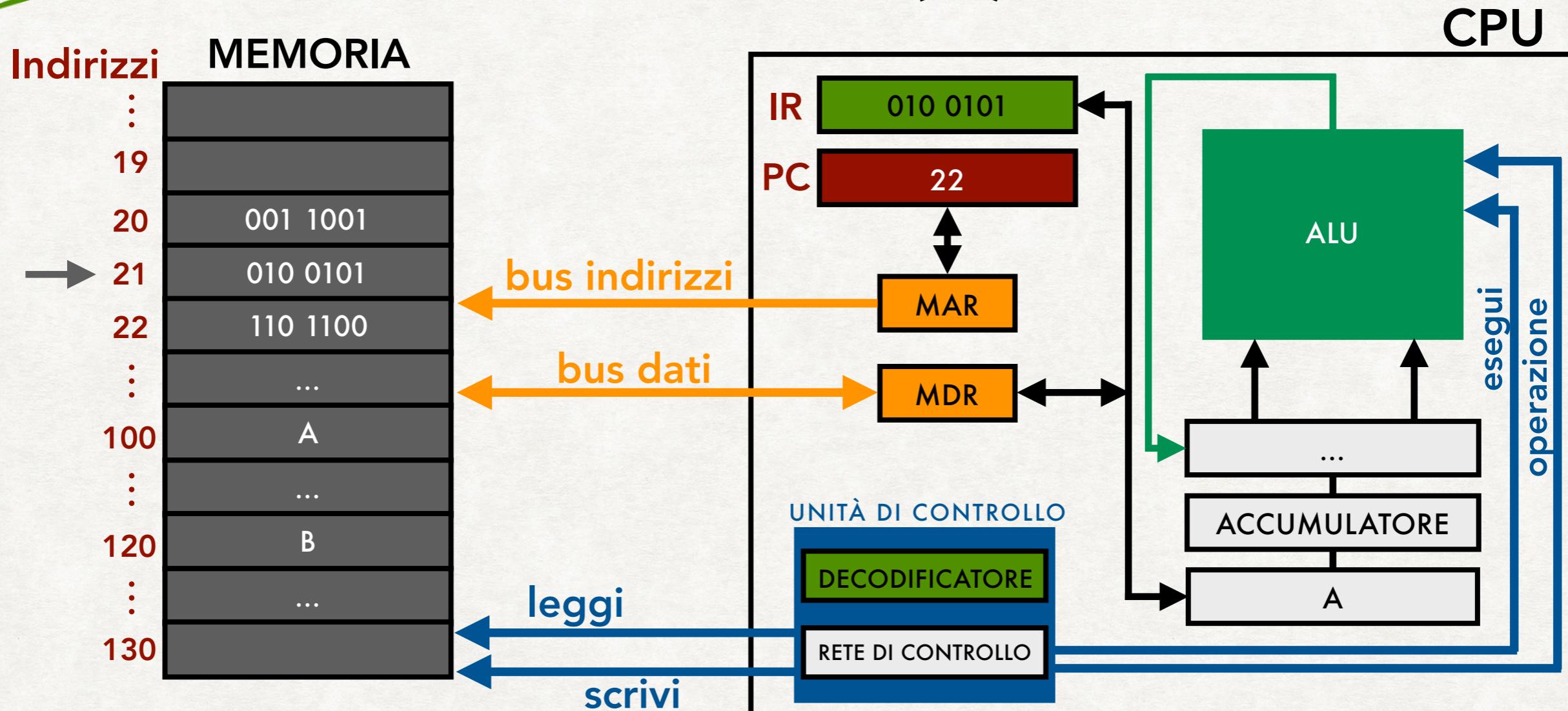
## UN ESEMPIO (10)



**FASE DECODE**

# CICLO DI ESECUZIONE

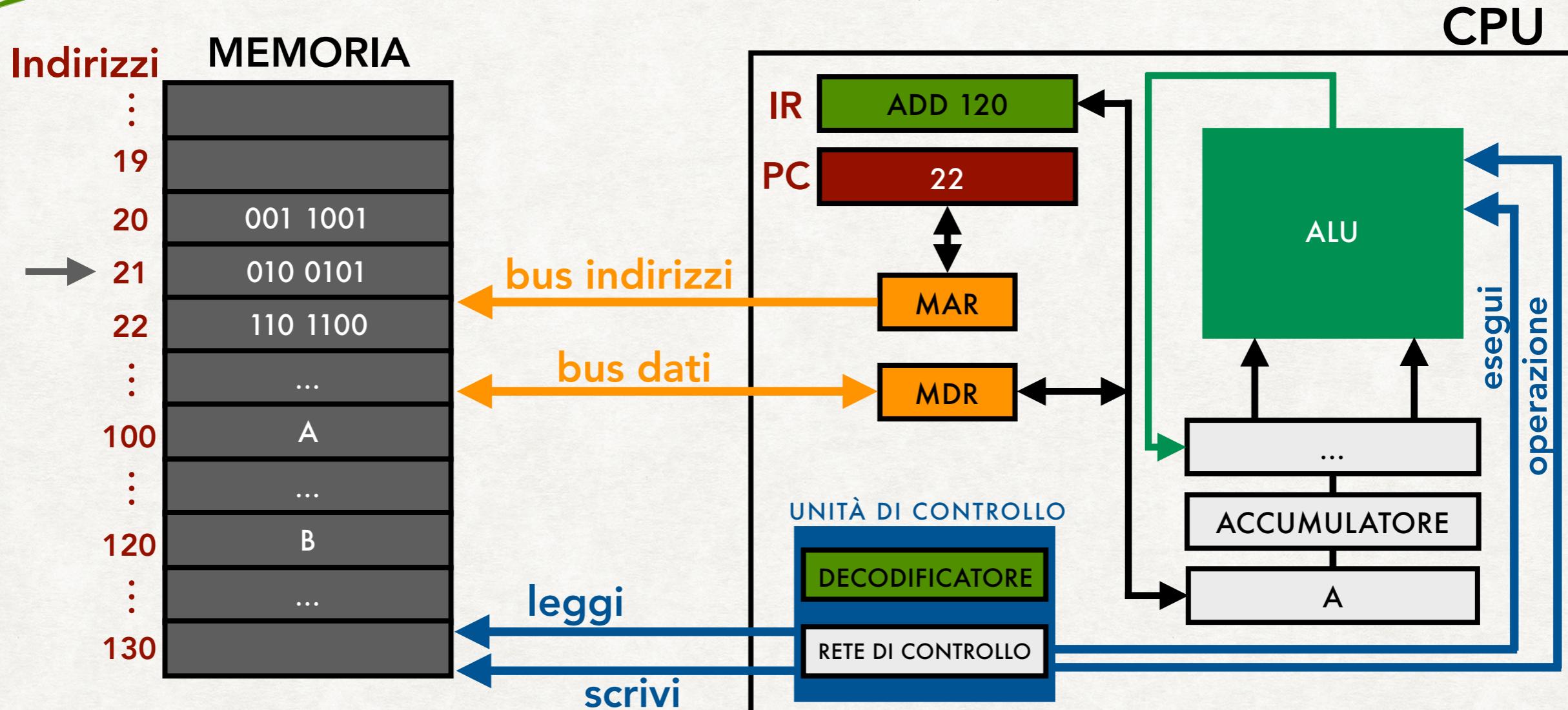
## UN ESEMPIO (11)



FASE DECODE

# CICLO DI ESECUZIONE

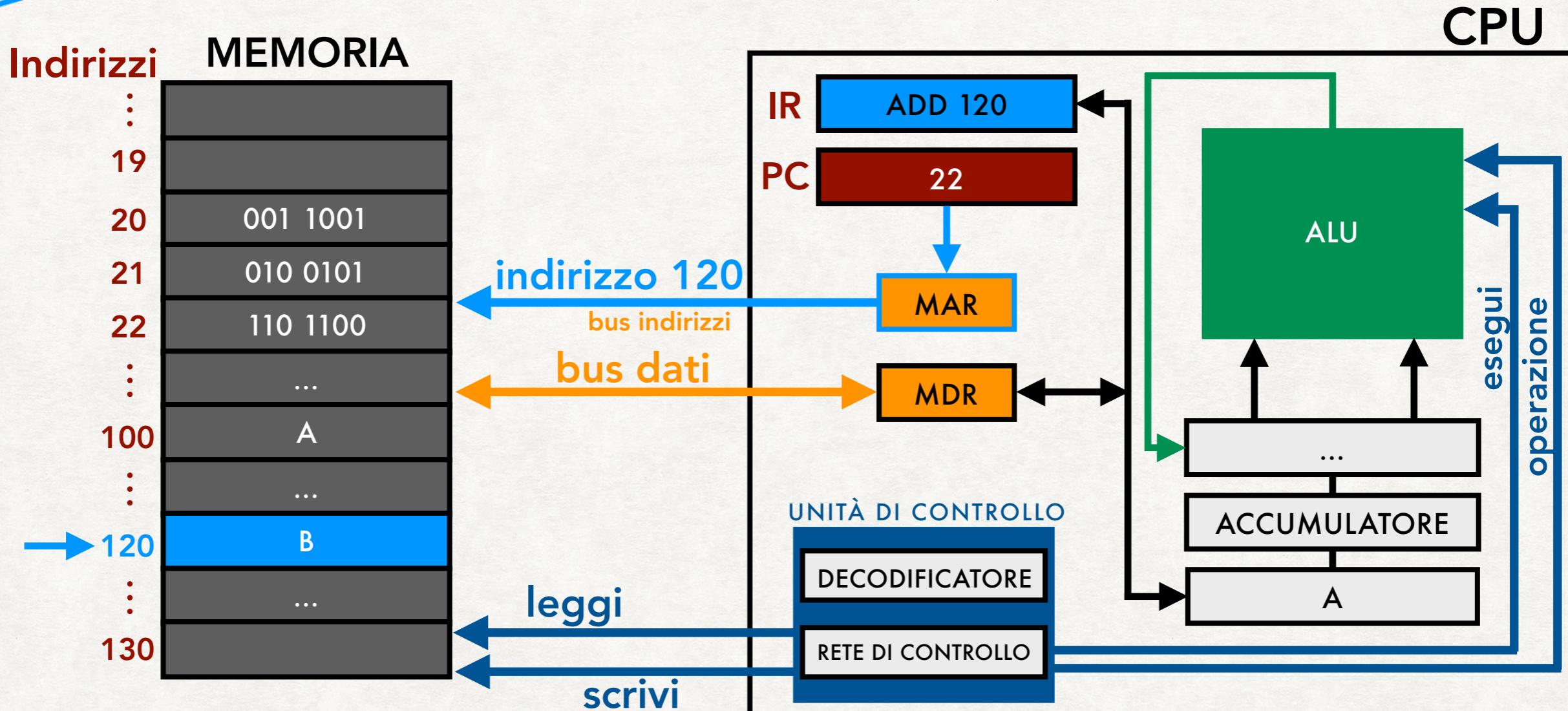
## UN ESEMPIO (12)



FASE EXECUTE

# CICLO DI ESECUZIONE

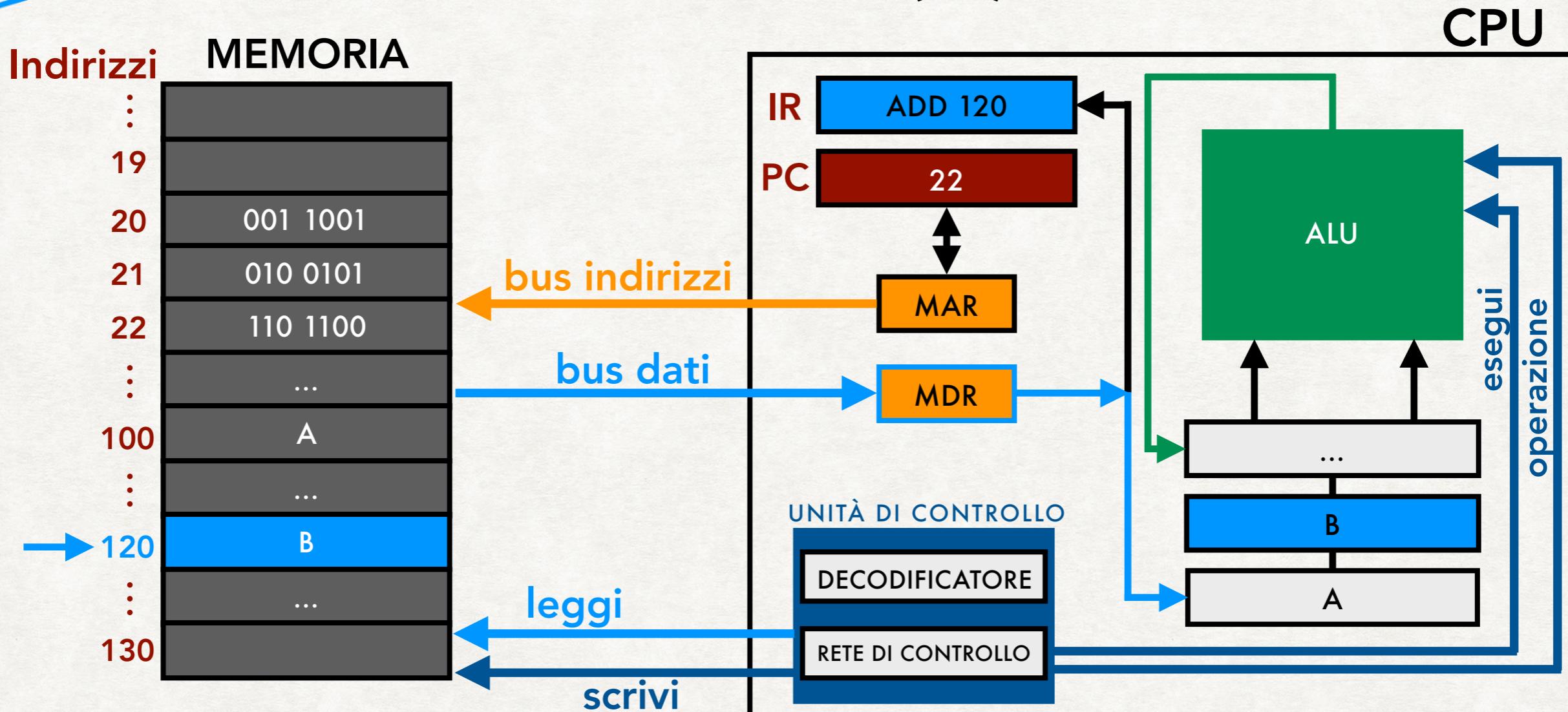
## UN ESEMPIO (13)



FASE EXECUTE

# CICLO DI ESECUZIONE

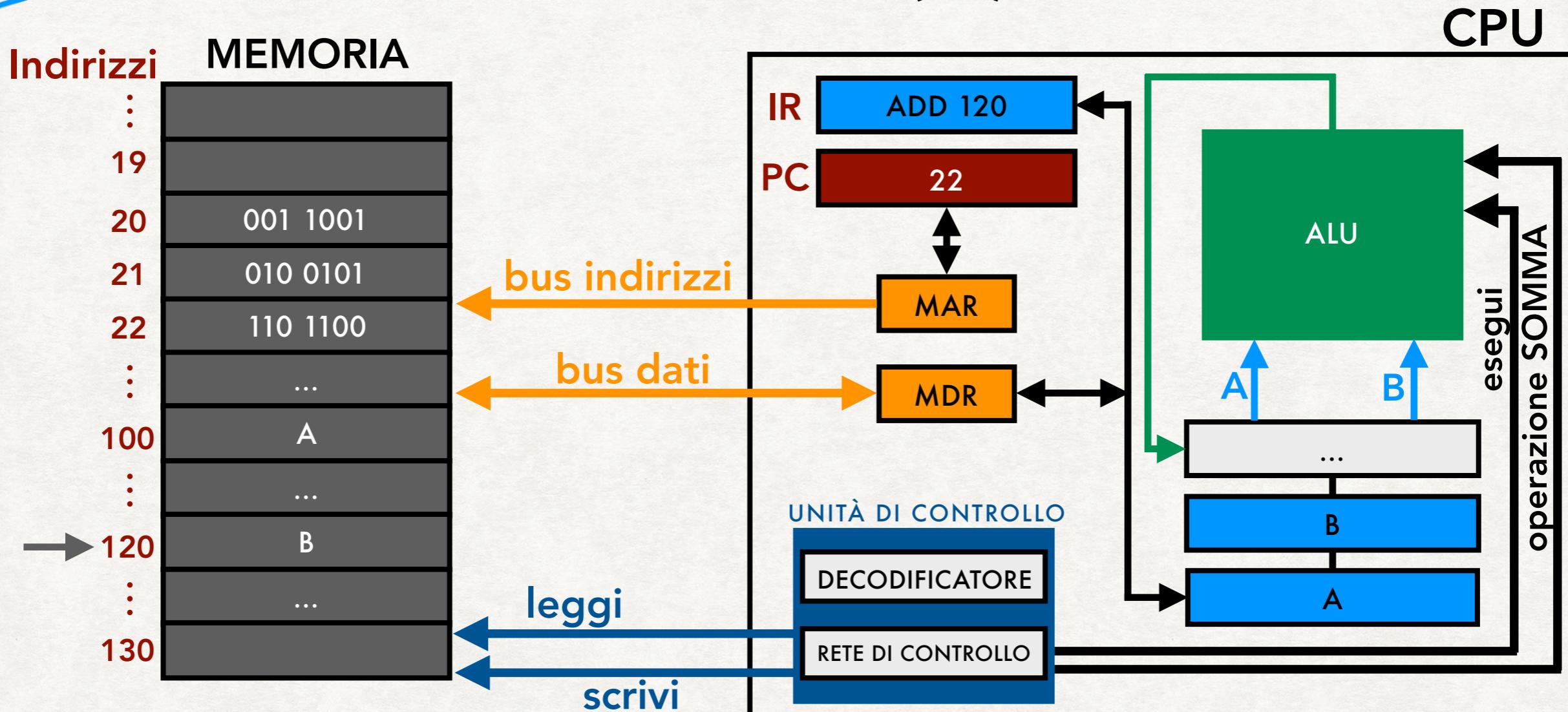
## UN ESEMPIO (14)



FASE EXECUTE

# CICLO DI ESECUZIONE

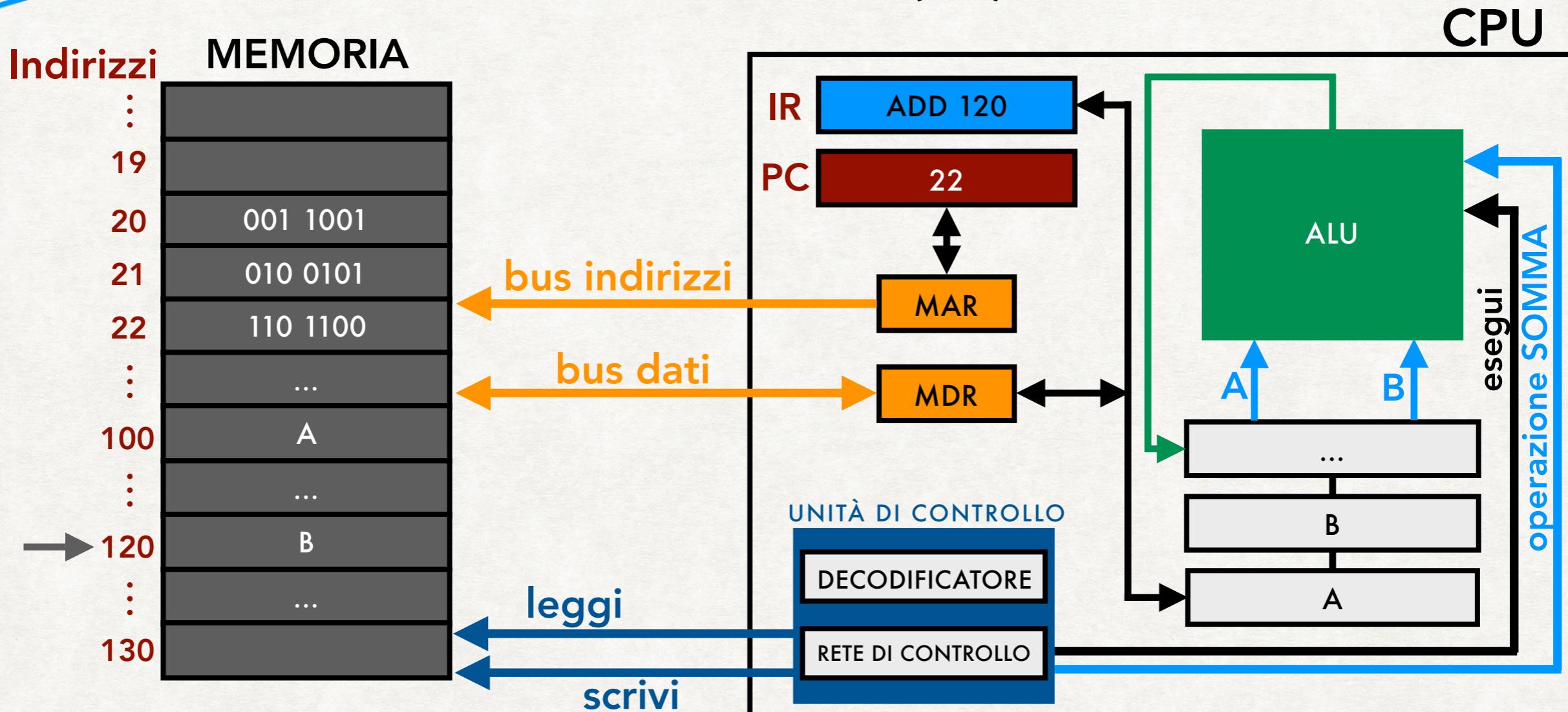
## UN ESEMPIO (15)



FASE EXECUTE

# CICLO DI ESECUZIONE

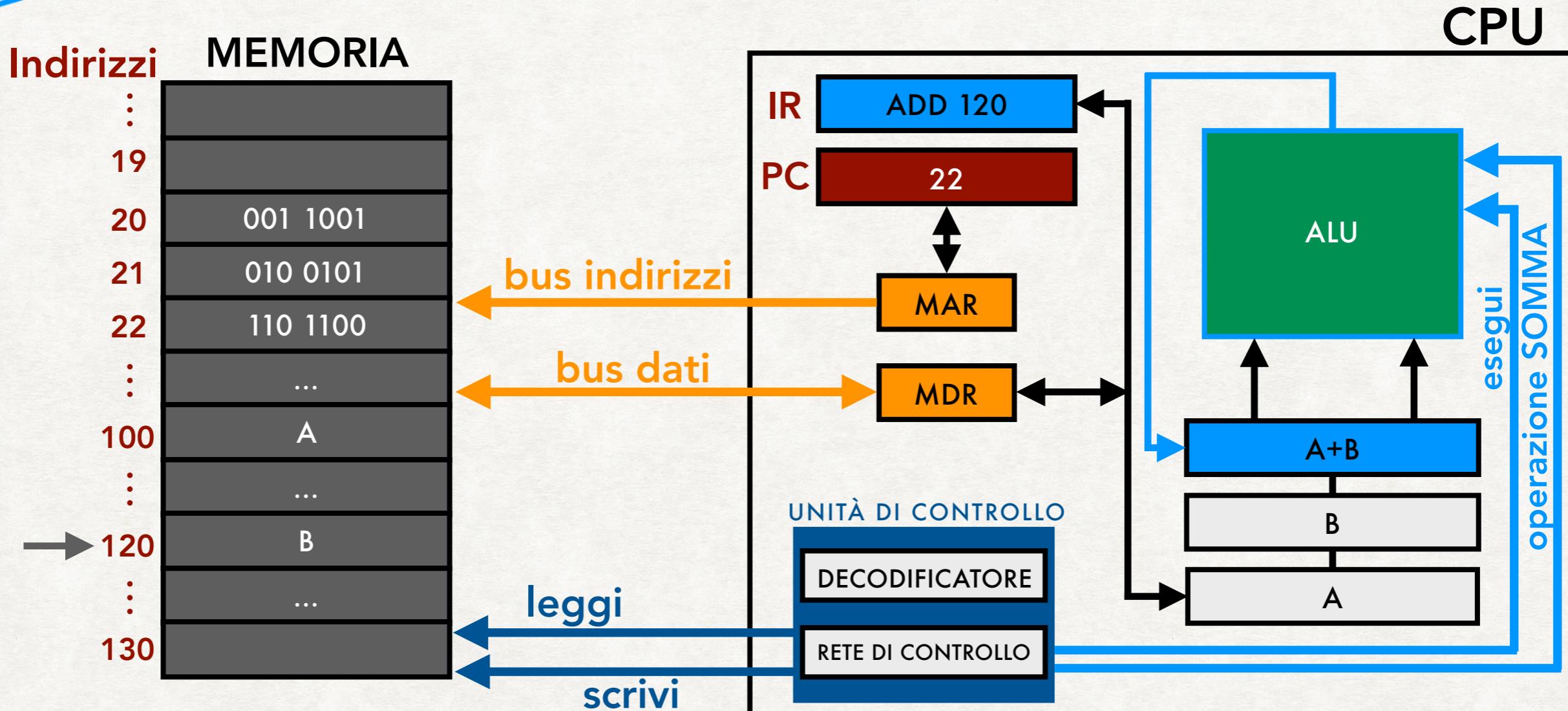
## UN ESEMPIO (16)



FASE EXECUTE

# CICLO DI ESECUZIONE

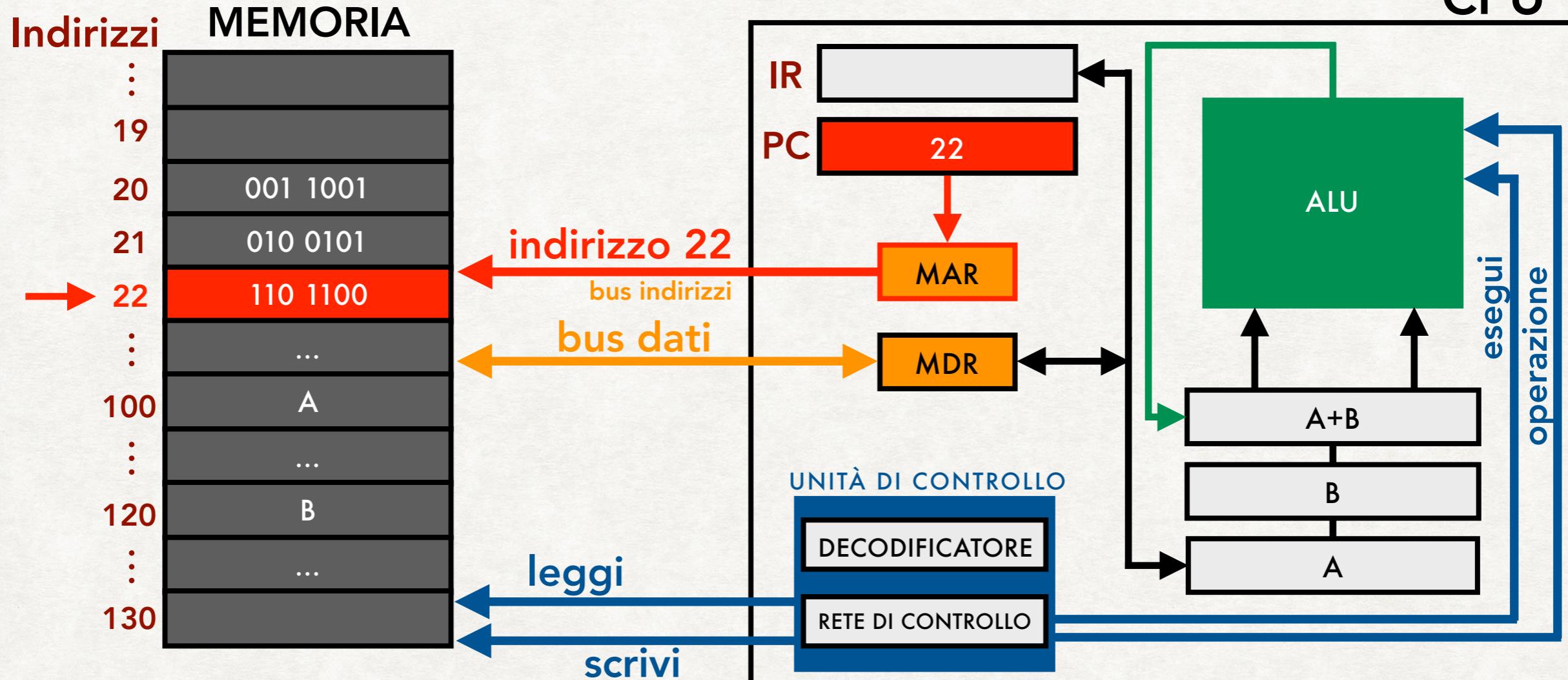
## UN ESEMPIO (17)



**FASE FETCH**

# CICLO DI ESECUZIONE

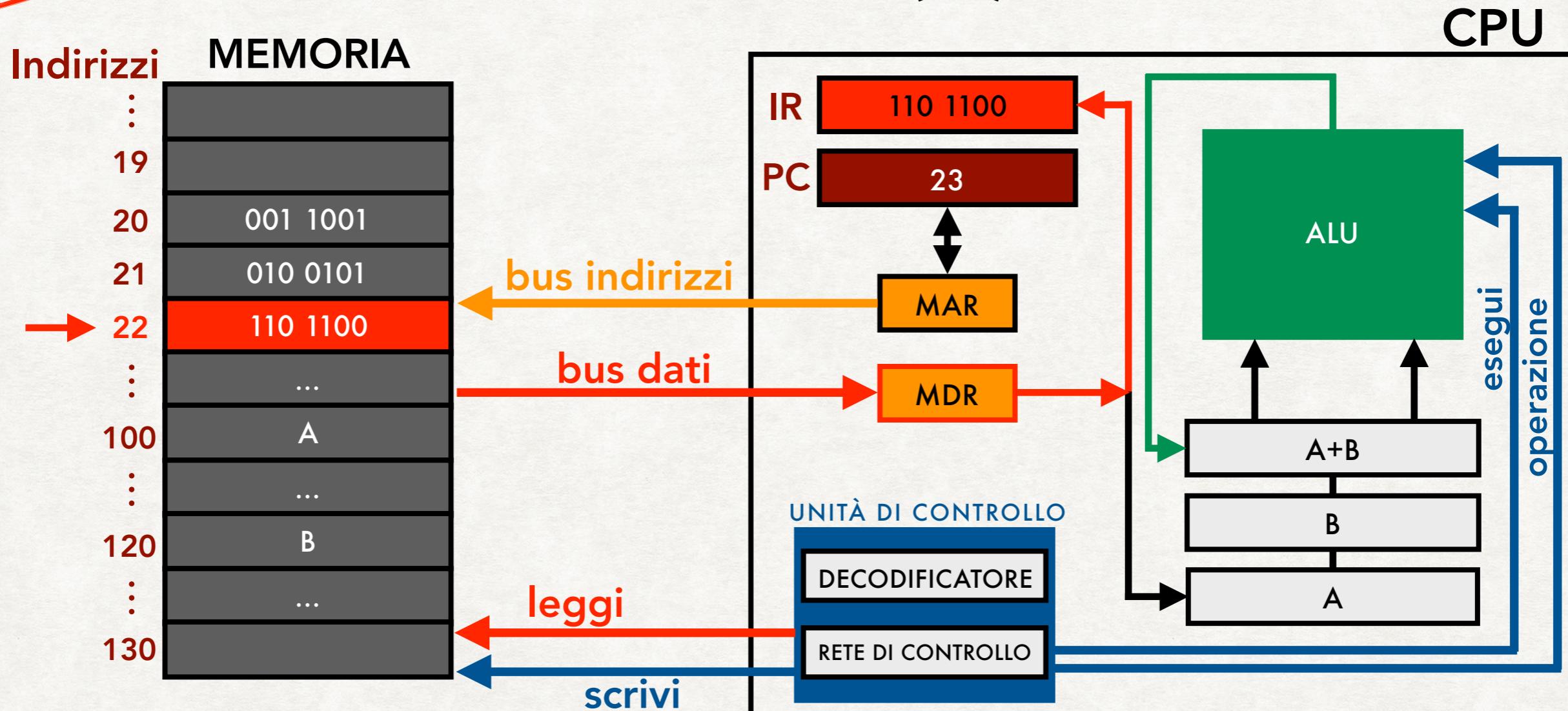
## UN ESEMPIO (18)



**FASE FETCH**

# CICLO DI ESECUZIONE

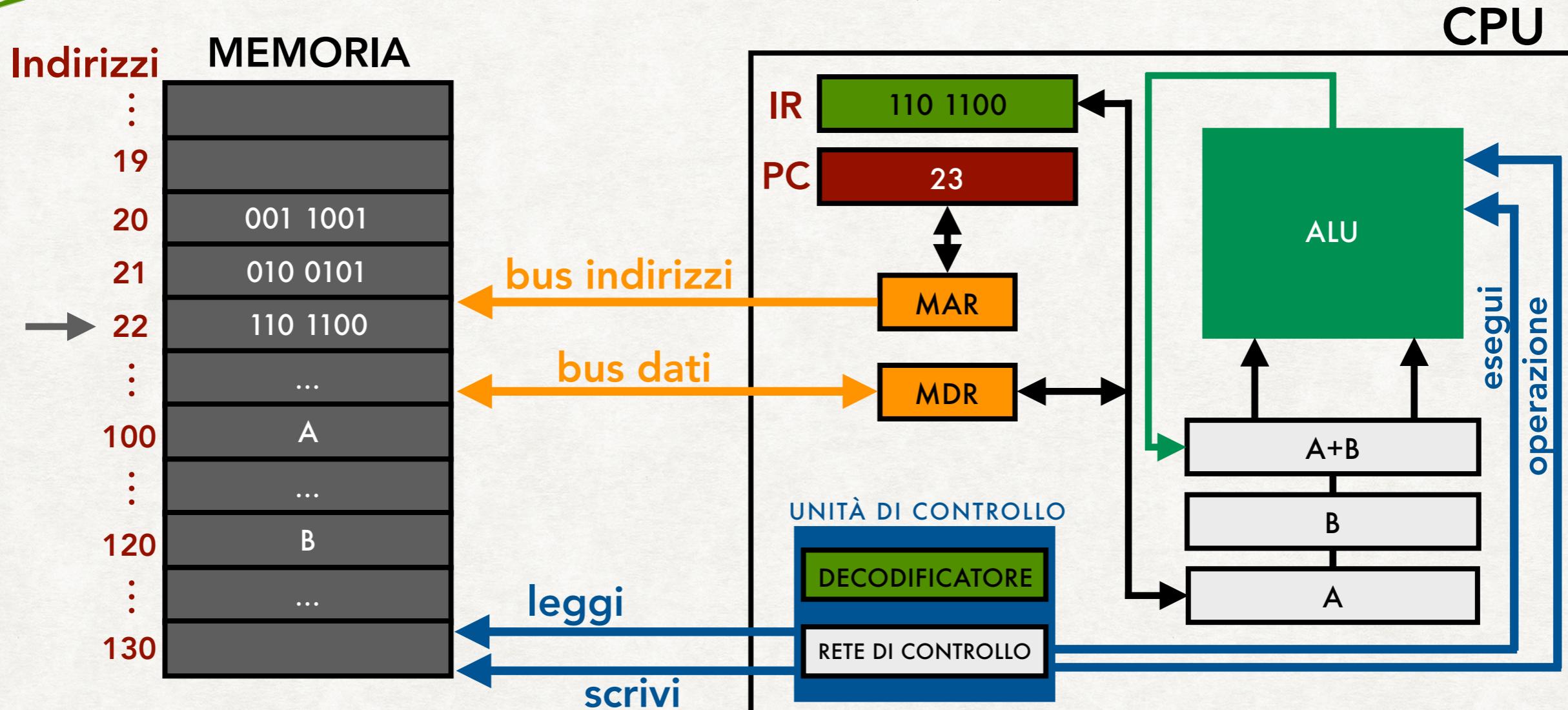
## UN ESEMPIO (19)



**FASE DECODE**

# CICLO DI ESECUZIONE

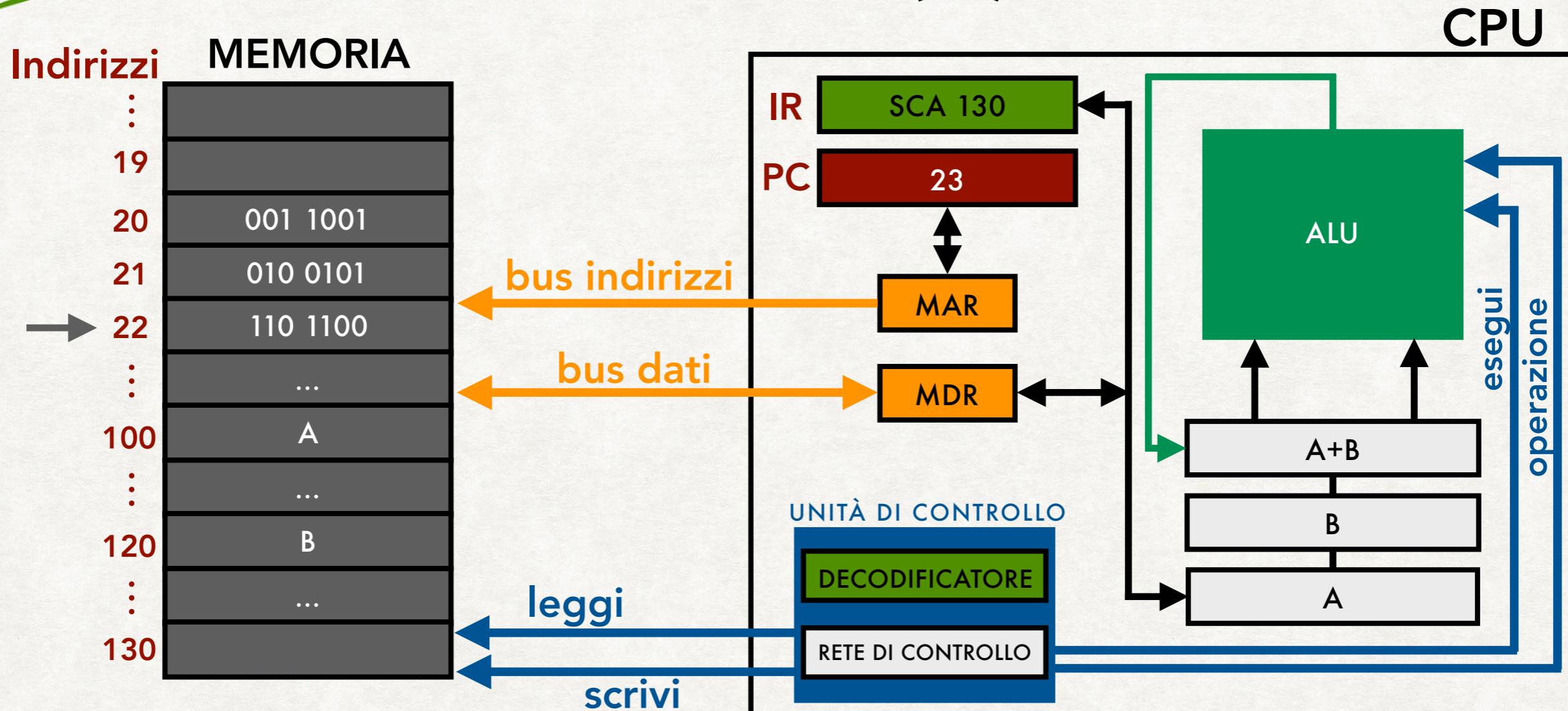
## UN ESEMPIO (20)



**FASE DECODE**

# CICLO DI ESECUZIONE

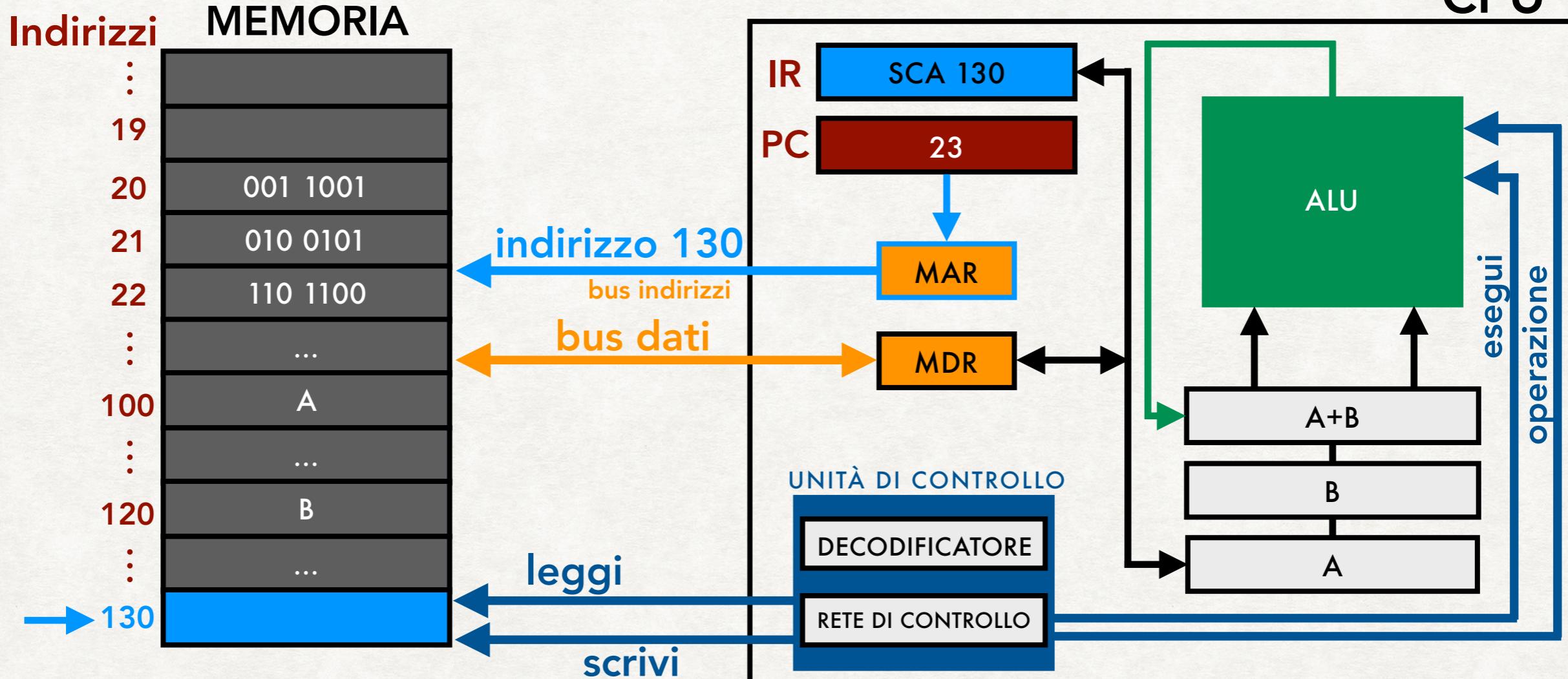
## UN ESEMPIO (21)



FASE EXECUTE

# CICLO DI ESECUZIONE

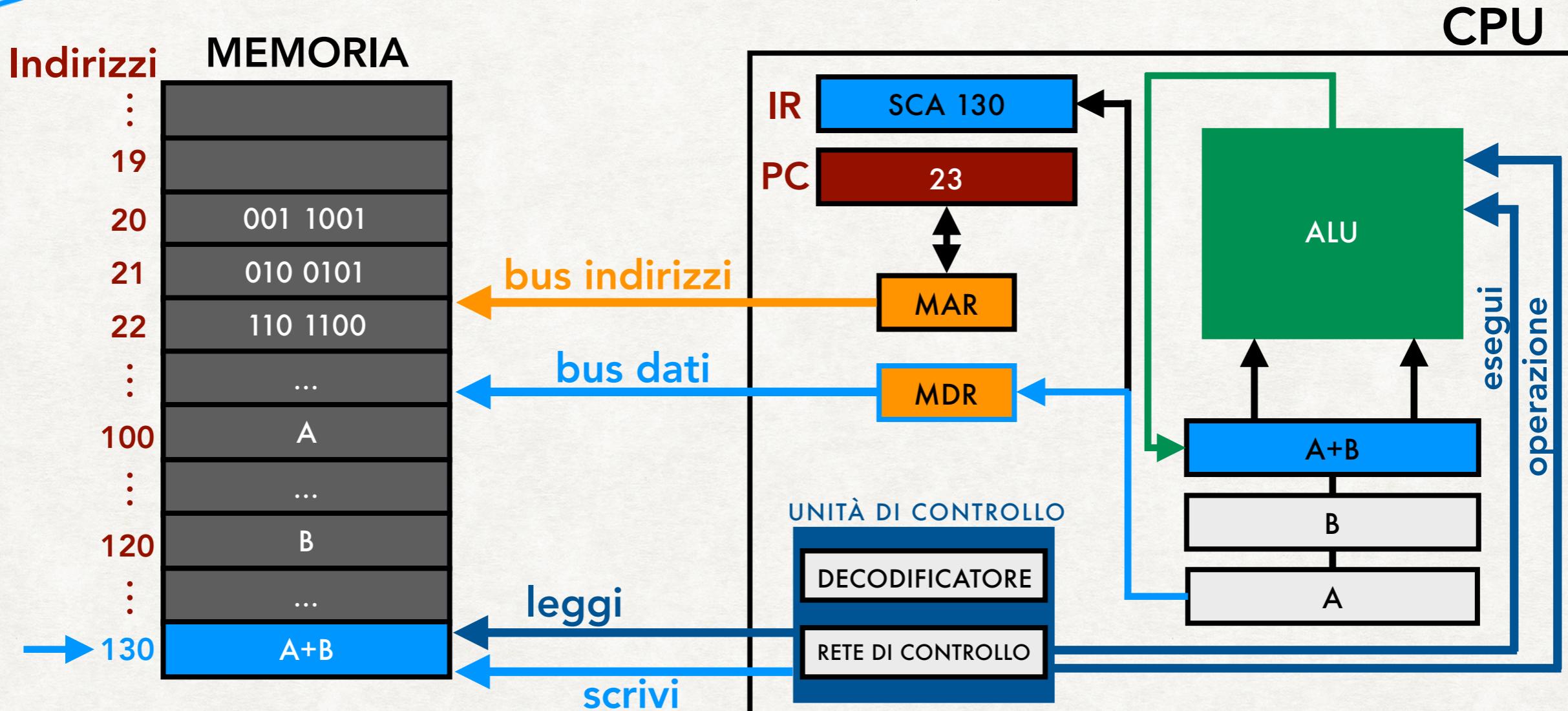
## UN ESEMPIO (22)



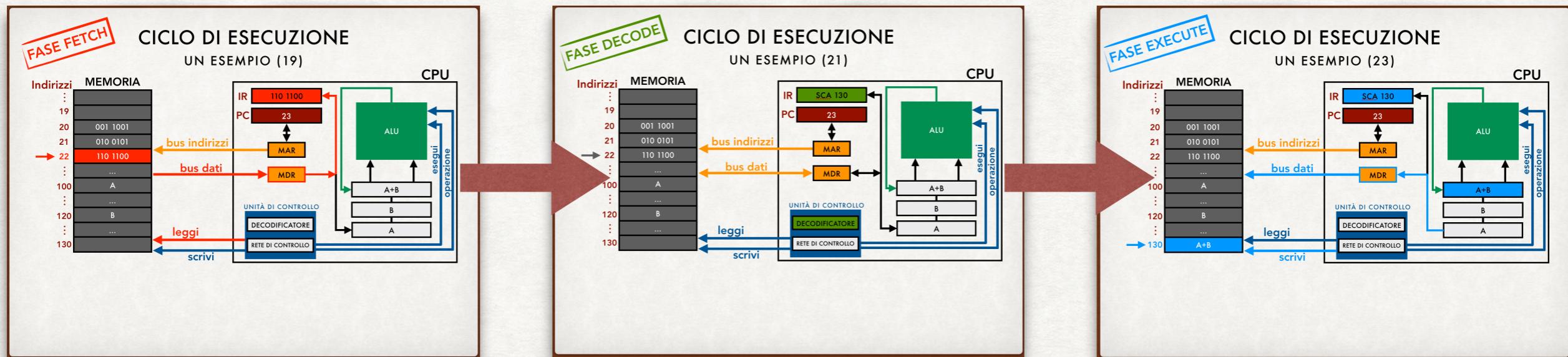
FASE EXECUTE

# CICLO DI ESECUZIONE

## UN ESEMPIO (23)



# QUANTO TEMPO CI VUOLE PER ESEGUIRE UN CICLO DI ESECUZIONE DELLE ISTRUZIONI?



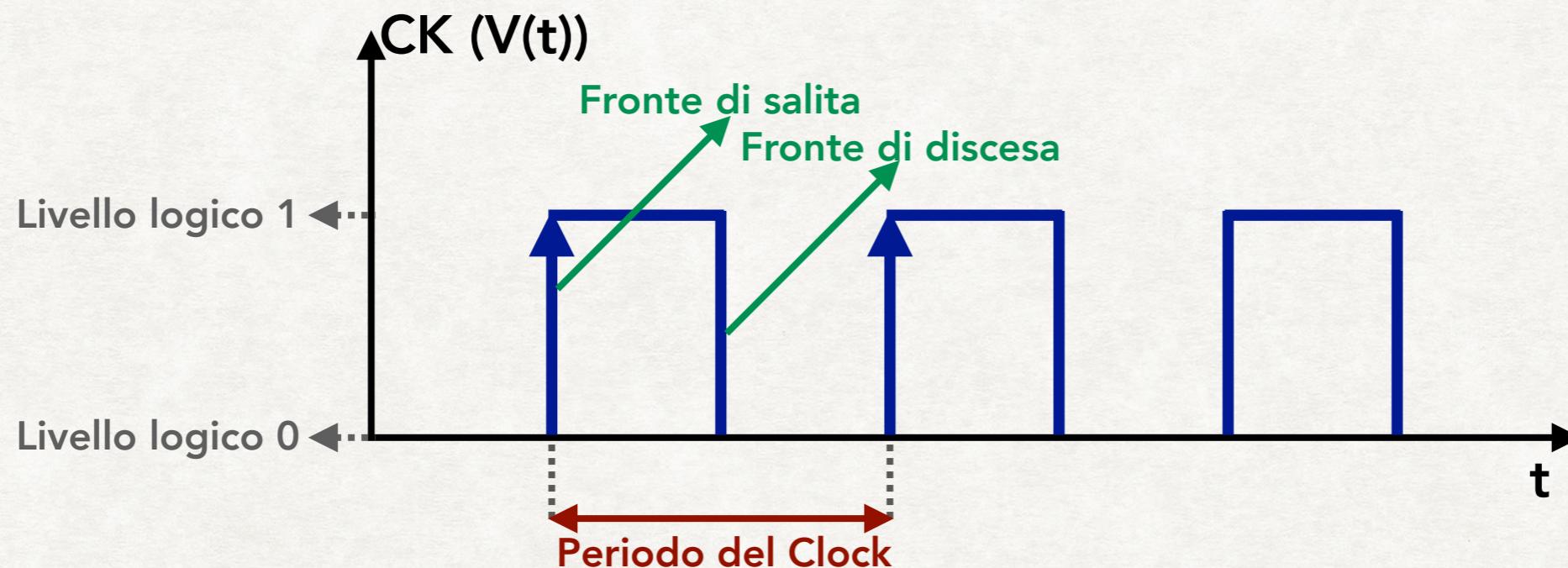
Un ciclo di esecuzione delle istruzioni, ovvero una sequenza di fase fetch, decode ed execute, è eseguito in un **CICLO di CLOCK**.

La frequenza del clock misura, se così si può dire, i “battiti del cuore” del computer.

# IL CLOCK

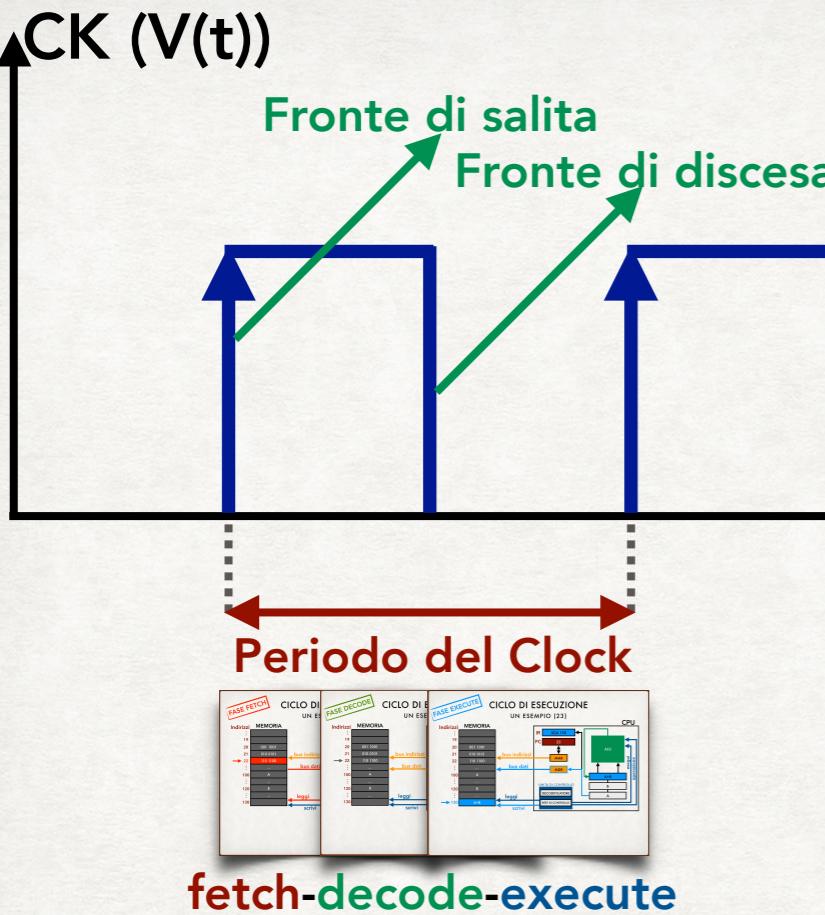
Il clock è un **segnale periodico**, generalmente un'onda quadra, ed è utilizzato per sincronizzare il funzionamento dei dispositivi elettronici digitali.

Il clock è utilizzato come orologio interno del microprocessore e il suo segnale sincronizza tutto il sistema.



L'unità di misura del clock è cicli al secondo (Hertz, Hz).

La **velocità o frequenza di clock** è il numero di commutazioni tra i due livelli logici "0" e "1" che circuiti logici interni ad un'unità di calcolo o di un microprocessore sono in grado di eseguire nell'unità di tempo di un secondo. Attualmente la frequenza dei processori è dell'ordine dei Gigahertz (GHz), ossia miliardi di computazioni al secondo ( $1\text{GHz} = 1000\text{MHz}$ ).



## IL CLOCK (2)

Per esempio, un computer che lavora a 2,6 GHz (o 2600MHz) esegue due milioni e seicento milioni di cicli di istruzione al secondo.



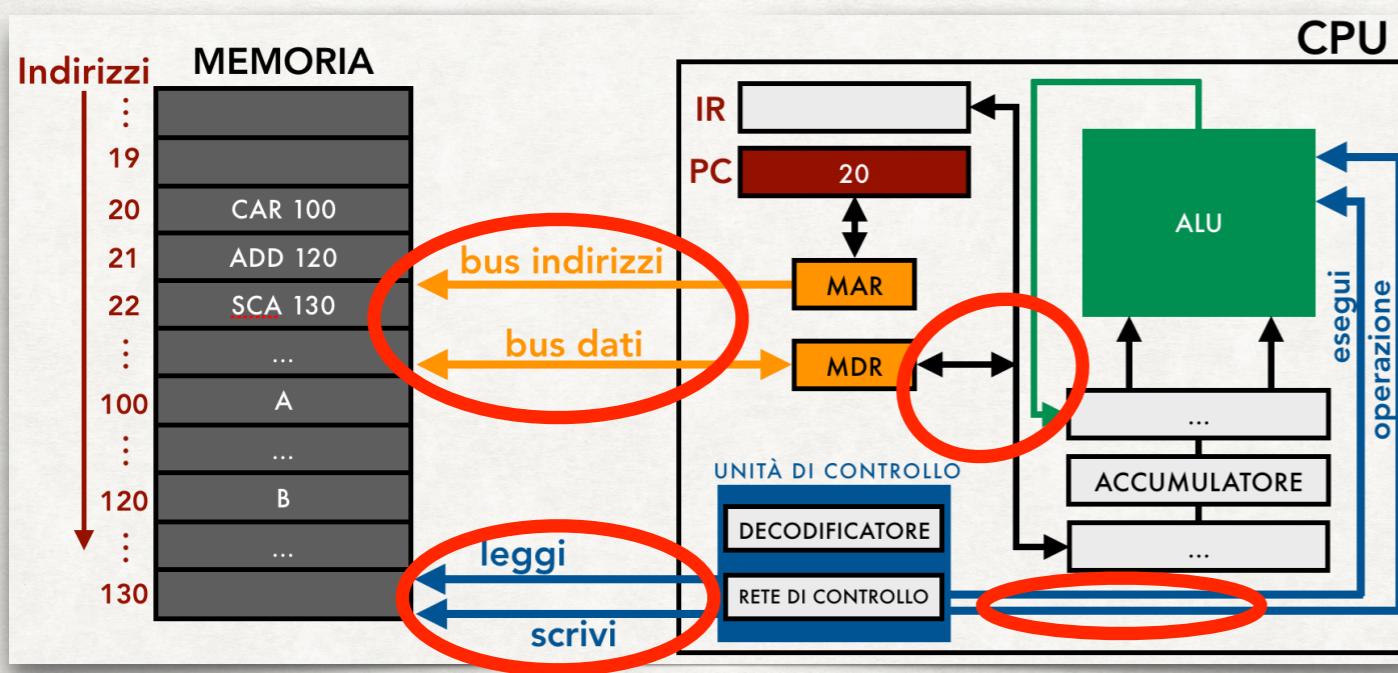
Per i più curiosi

I microprocessori delle classi più recenti sono solitamente più performanti rispetto a quelli delle classi più vecchie. Un processore che costituisce la punta avanzata della tecnologia informatica può diventare già entro pochi mesi abbastanza superato. Ogni 18 mesi la velocità dei processori e il numero di componenti interni della CPU raddoppia (legge di Moore). In base a questa osservazione un processore acquistato oggi diventa vecchio in meno di un anno e decisamente obsoleto in un anno e mezzo.



# I BUS

Un **bus** è un insieme di connessioni elettriche parallele (fili) utilizzato per trasportare tutte le informazioni da un componente all'altro.



È un canale che mette in comunicazione tutte le unità di un computer. Il bus può trasportare *dati*, *indirizzi* e *comandi*.

I bus sono di tre tipi:

- ❖ **bus dati (Data Bus)**
- ❖ **bus indirizzi (Address Bus)**
- ❖ **bus comandi (Command Bus)**

- ❖ Il **bus dati** serve per trasferire i dati tra la memoria centrale e il registro dati MDR della CPU, e per trasferire dati tra le periferiche e la CPU e/o la memoria centrale. Questo bus è bidirezionale.
- ❖ Il **bus indirizzi** serve per trasmettere il contenuto del registro indirizzi MAR alla memoria (o a una periferica). Serve a selezionare una cella per le successive operazioni di lettura o scrittura. Questo bus è unidirezionale.
- ❖ Il **bus comandi** serve per inviare comandi:
  - verso la memoria, ad esempio lettura o scrittura;
  - verso una periferica o un'altra unità, ad esempio esegui inviato all'ALU.Anche questo bus è unidirezionale.

# CARATTERISTICHE DEL COLLEGAMENTO A BUS

Il bus interconnette le componenti interne del calcolatore (CPU, memorie ed interfacce a periferiche, quali ad esempio I/O, memorie di massa, etc.).

Connette due unità funzionali alla volta, nella logica che una riceve e l'altra trasmette, per questi motivi è:

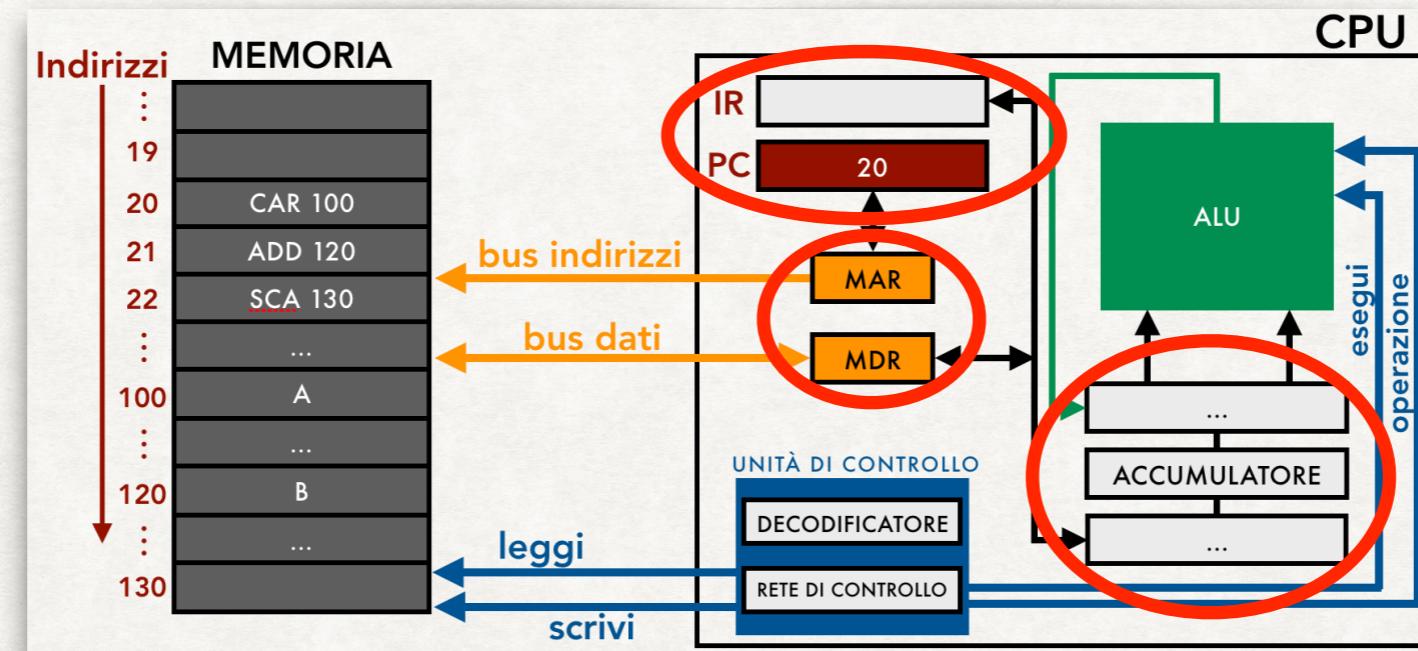
- ✓ molto **SEMPLICE** da realizzare: essendoci un'unica linea di connessione si riducono i costi di produzione;
- ✓ inoltre è facilmente **ESTENDIBILE**, ovvero è molto semplice aggiungere nuovi dispositivi;
- ✗ purtroppo pecca in **LENTEZZA**, poiché vi è l'utilizzo in mutua esclusione del bus.

Il funzionamento implementato è quello di **Master/Slave**: la CPU (Master) seleziona la connessione da attivare e *ordina* il trasferimento dei dati. Le caratteristiche derivanti sono:

- ✓ **STANDARDIZZABILITÀ**: ci sono regole ben precise per la comunicazione da parte di dispositivi diversi;
- ✗ **LIMITATA CAPACITÀ** al crescere del numero dispositivi collegati dallo stesso bus;
- ✗ **SOVRACCARICO** del **PROCESSORE** perché funge lui solo da master sul controllo del bus.

# I REGISTRI

I registri sono *piccole ma veloci* unità di memoria interne alla CPU, in grado di memorizzare temporaneamente un'informazione utile e significativa in determinate fasi dell'elaborazione dei dati.

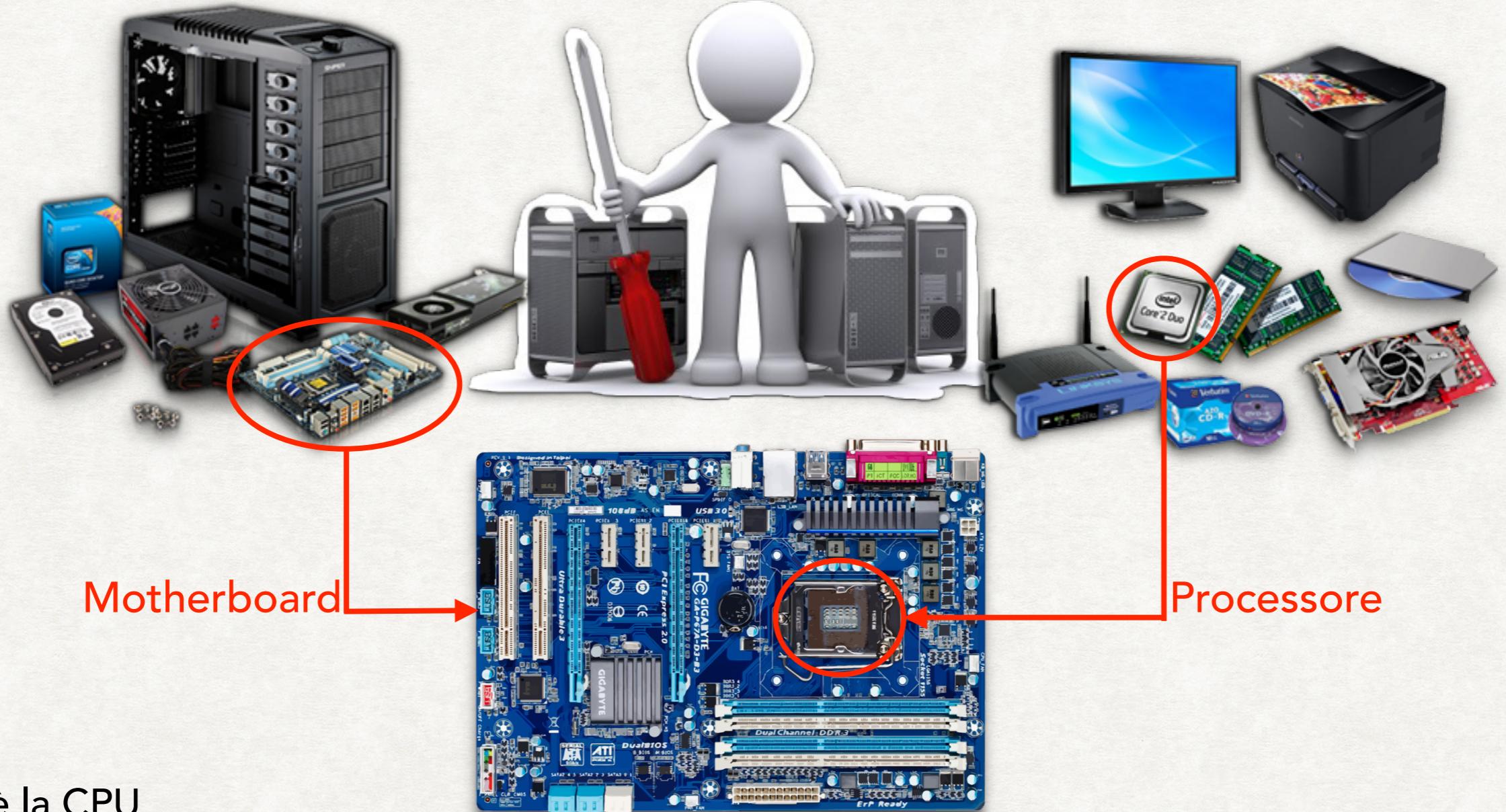


I registri supportano operazioni di lettura e scrittura.

**Le caratteristiche principali sono:**

- **elevata velocità**: sono memorie molto veloci, ovvero la scrittura e la lettura di queste unità avviene molto velocemente;
  - ne esistono di **varie tipologie**: quelli di uso generico (ACCUMULATORI) e quelli di uso specifico (Program Counter, Instruction Register, etc.).

# RELOAD!



- Cos'è la CPU
- Elementi che compongono la CPU
- Le istruzioni
- Le fasi in cui è suddivisa l'esecuzione di un'istruzione macchina (ciclo di esecuzione)
- Il clock
- I bus
- i registri