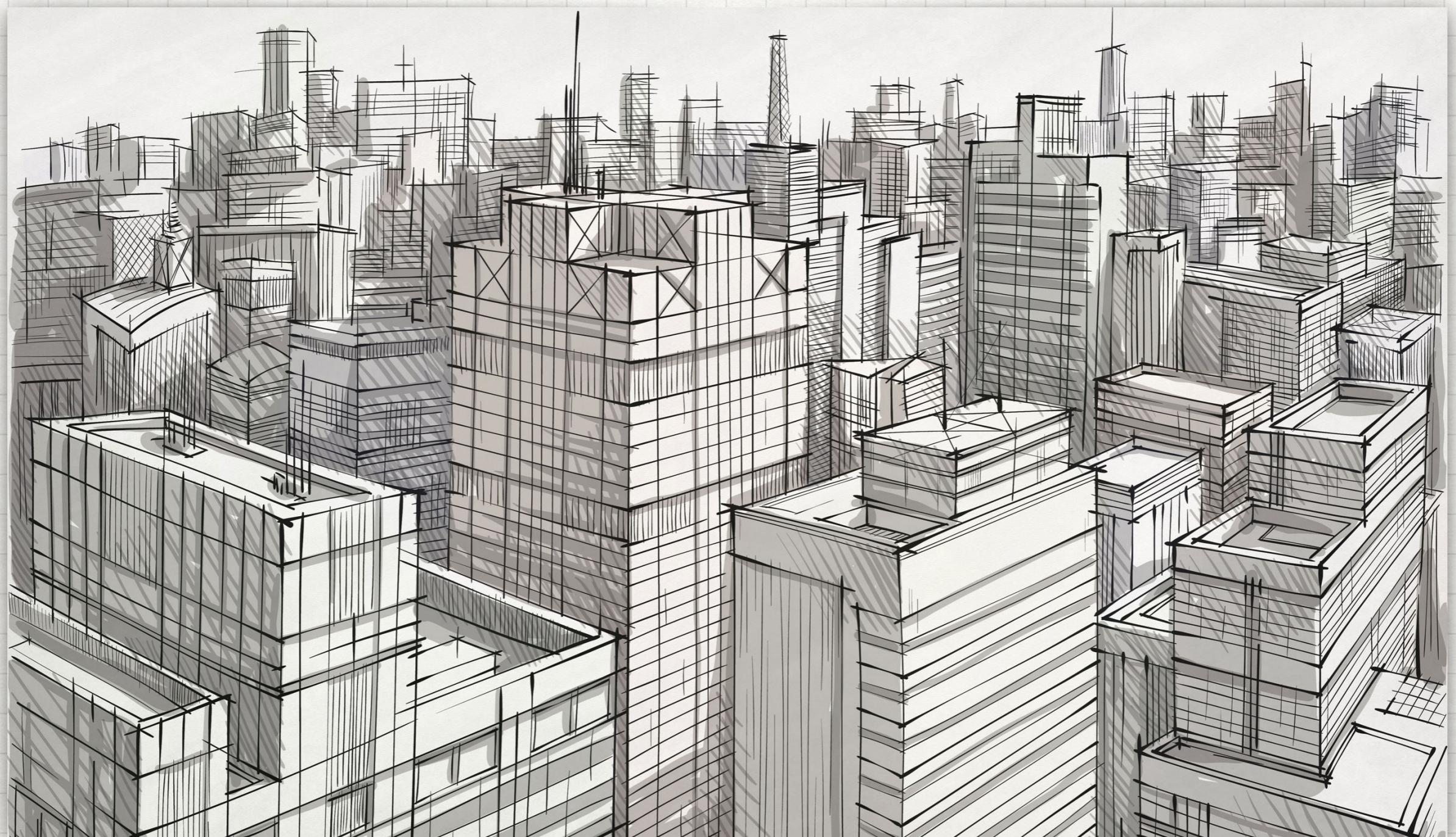


ELEMENTI DI INFORMATICA CDL ING. CIVILE A.A.2016/2017

# ARCHITETTURA DEL CALCOLATORE (2)

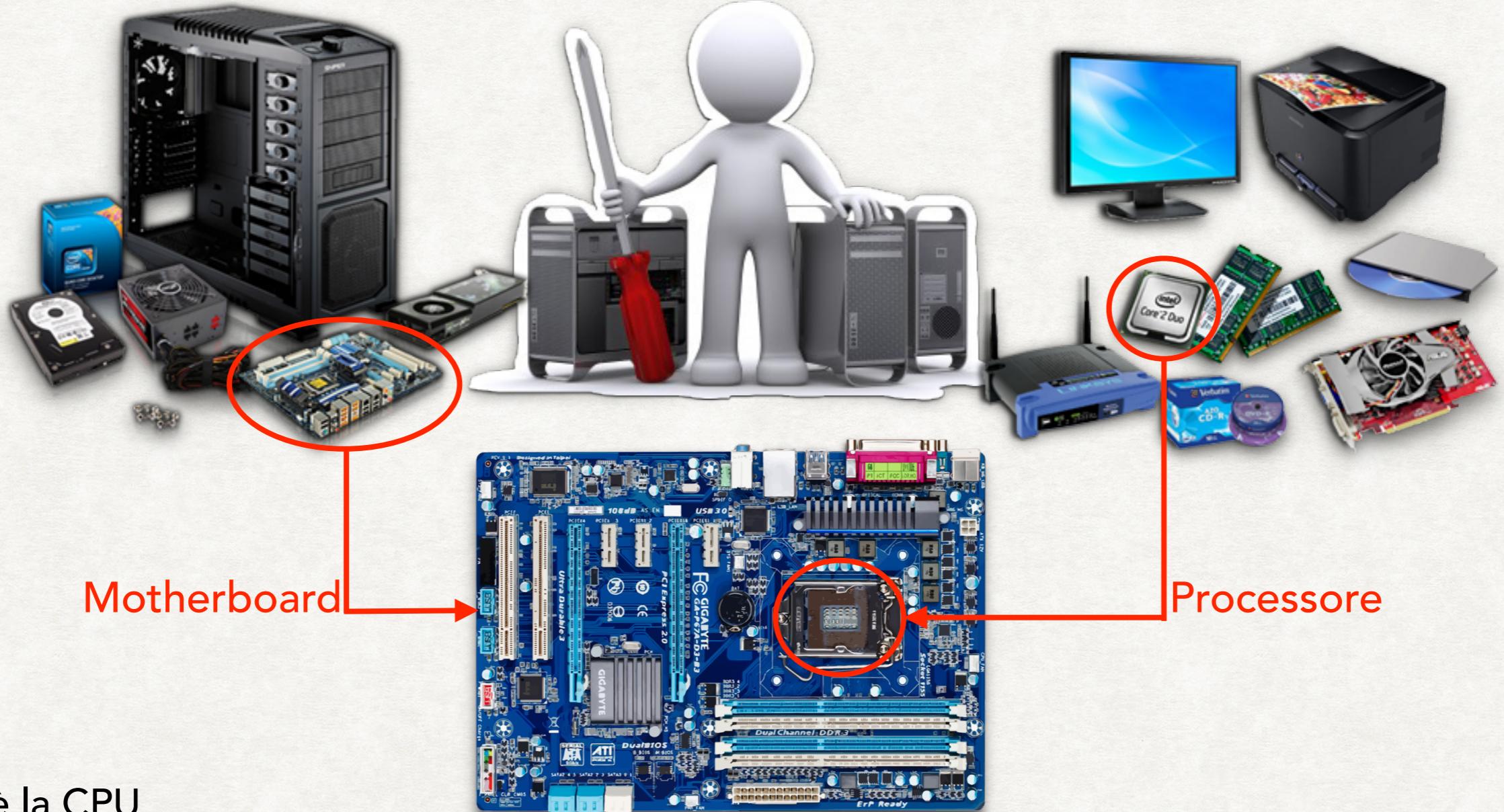


Vittoria Nardone,

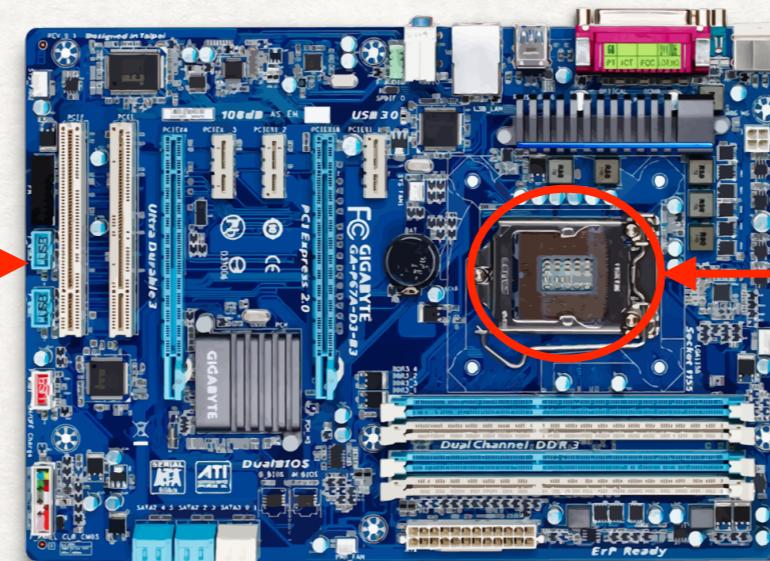
[vnardone@unisannio.it](mailto:vnardone@unisannio.it)

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

# RELOAD!



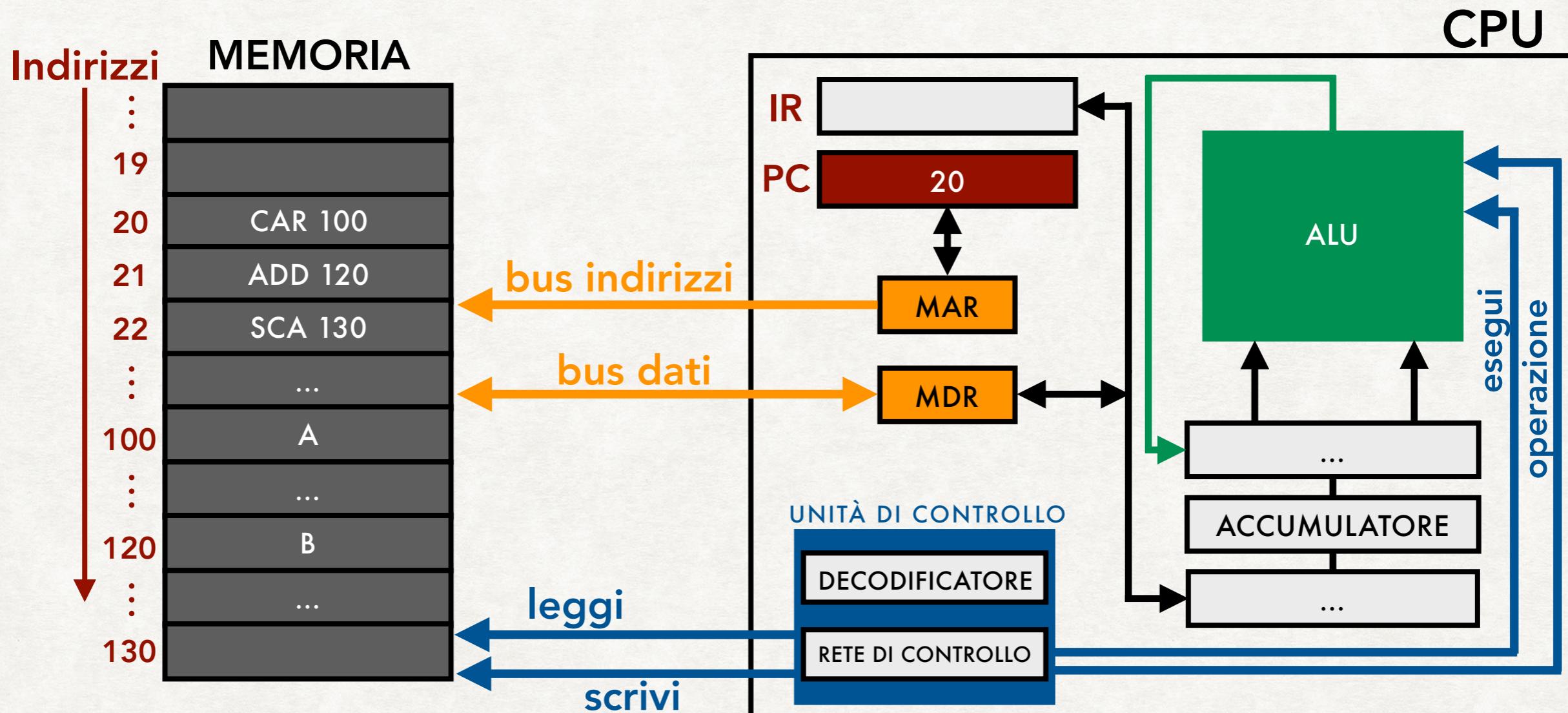
Motherboard



Processore

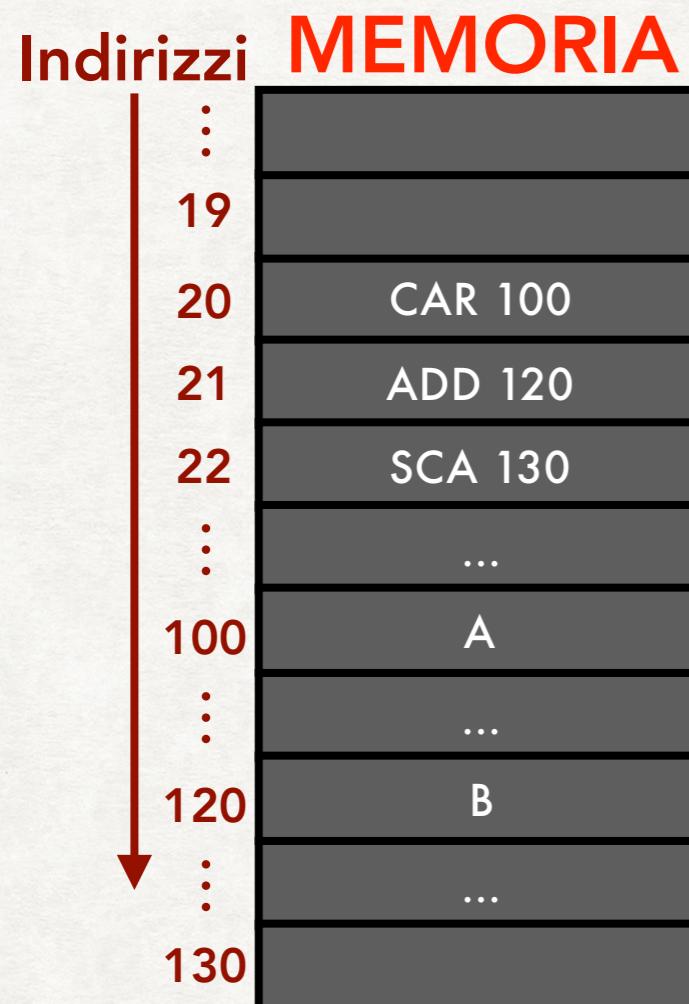
- Cos'è la CPU
- Elementi che compongono la CPU
- Le istruzioni
- Le fasi in cui è suddivisa l'esecuzione di un'istruzione macchina (ciclo di esecuzione)
- Il clock
- I bus
- i registri

# RELOAD!



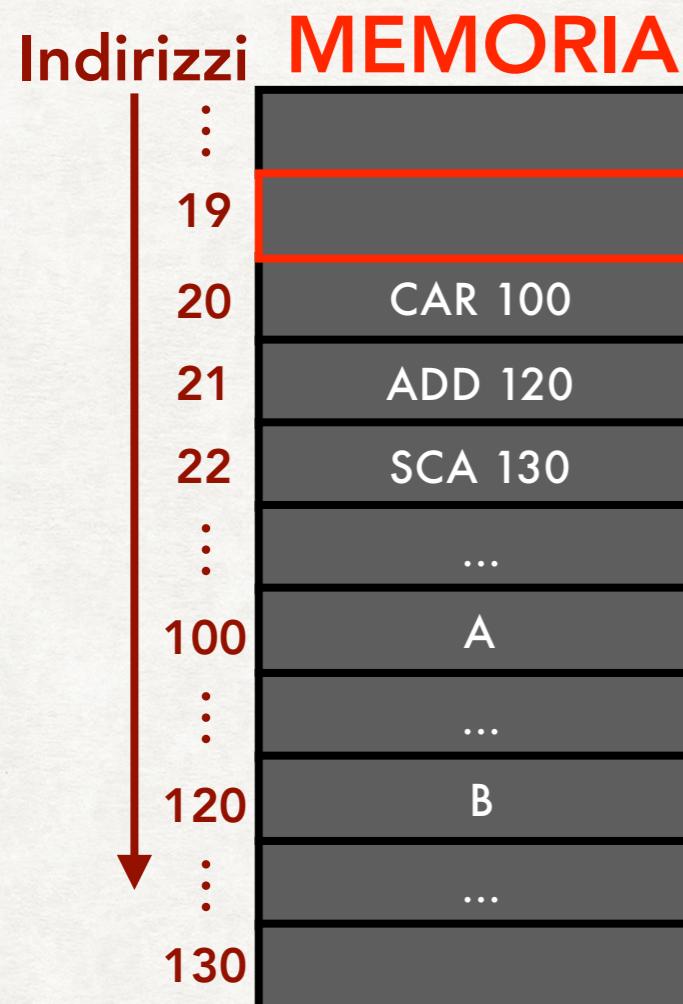
# MEMORIZZAZIONE

L'unità responsabile della memorizzazione è la **MEMORIA**



# MEMORIZZAZIONE

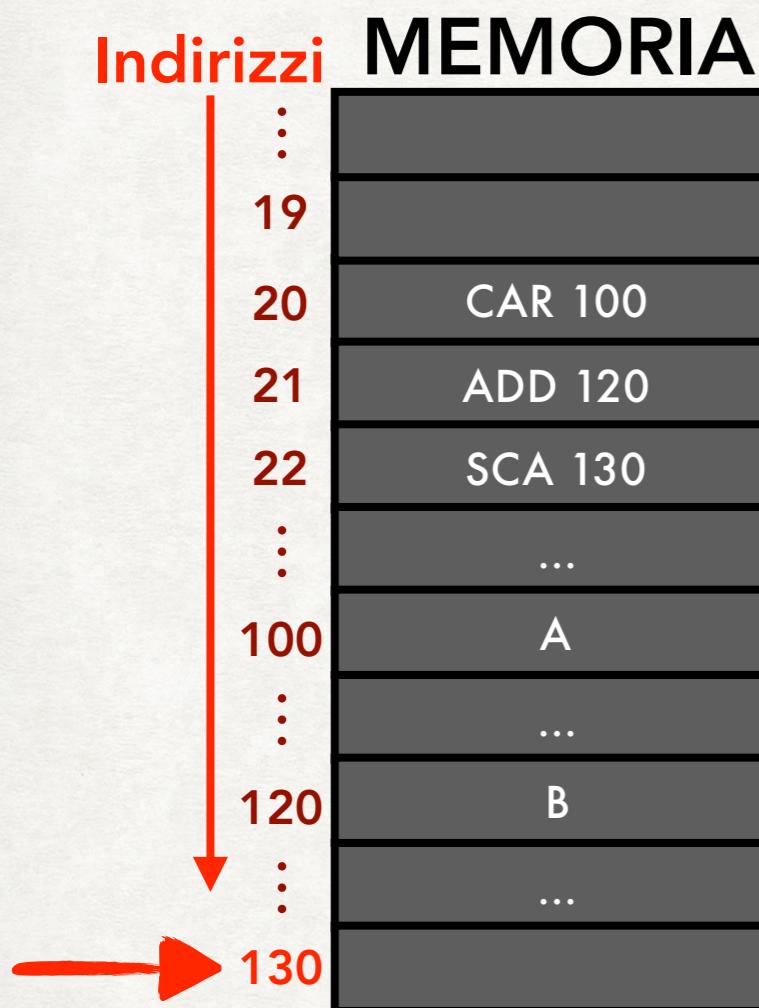
L'unità responsabile della memorizzazione è la **MEMORIA**



- L'unità di memoria è organizzata in **CELLE**;

# MEMORIZZAZIONE

L'unità responsabile della memorizzazione è la **MEMORIA**

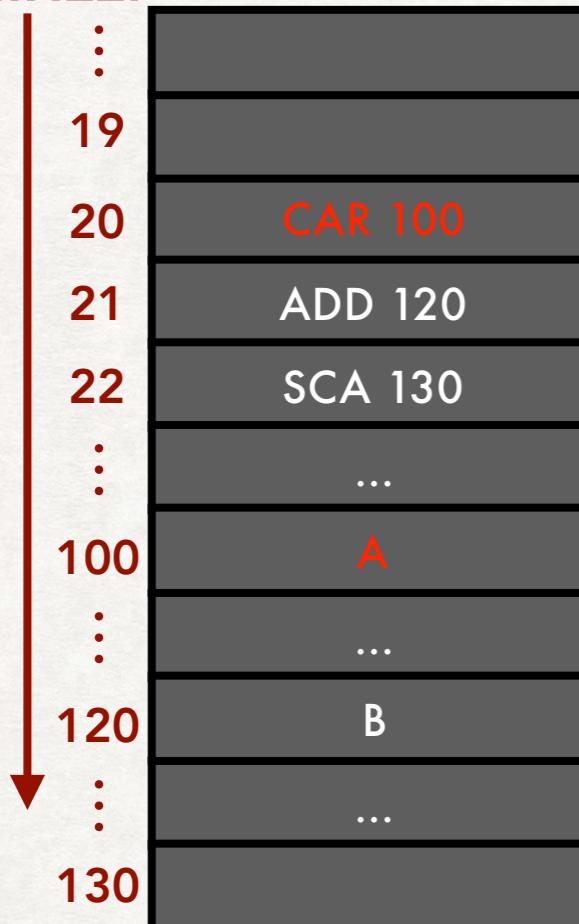


- L'unità di memoria è organizzata in **CELLE**;
- A ciascuna cella è associato un **INDIRIZZO** che la identifica;

# MEMORIZZAZIONE

L'unità responsabile della memorizzazione è la **MEMORIA**

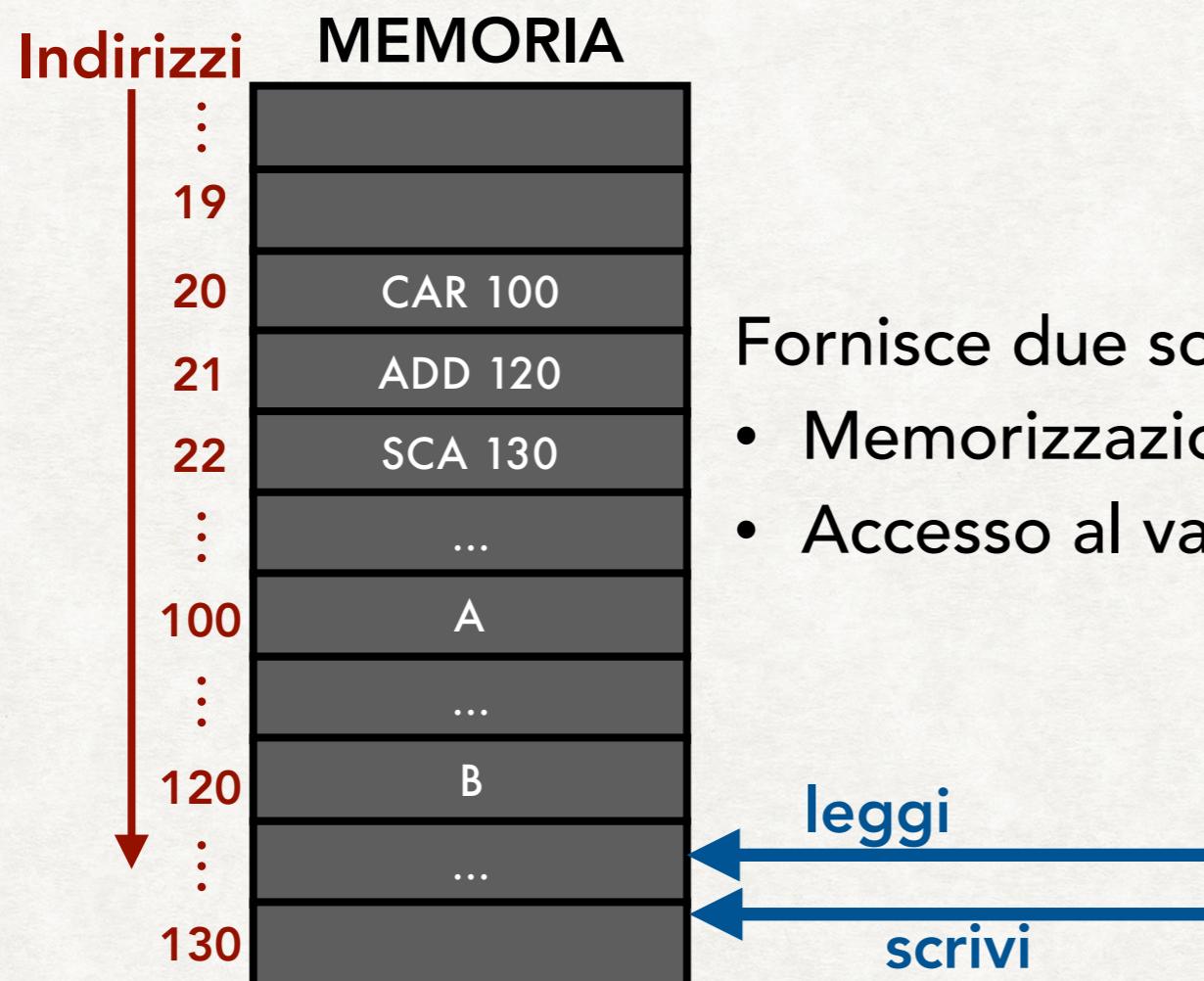
## Indirizzi **MEMORIA**



- L'unità di memoria è organizzata in **CELLE**;
- A ciascuna cella è associato un **INDIRIZZO** che la identifica;
- Ciascuna cella è in grado di memorizzare un **singolo dato o una singola istruzione**.

# MEMORIZZAZIONE

Un calcolatore memorizza i **DATI**, che rappresentano le informazioni di interesse, e i **PROGRAMMI** per l'elaborazione dei dati.



Fornisce due sole operazioni:

- Memorizzazione di un valore in una cella (**Scrittura**)
- Accesso al valore memorizzato in una cella (**Lettura**)

# ORGANIZZAZIONE DELLE MEMORIE

**La memoria ideale dovrebbe essere a capacita' infinita, tempo di accesso nullo e costo e consumo nullo.**

Nei casi reali l'organizzazione del calcolatore comprende diversi tipi di memorie con caratteristiche diverse:

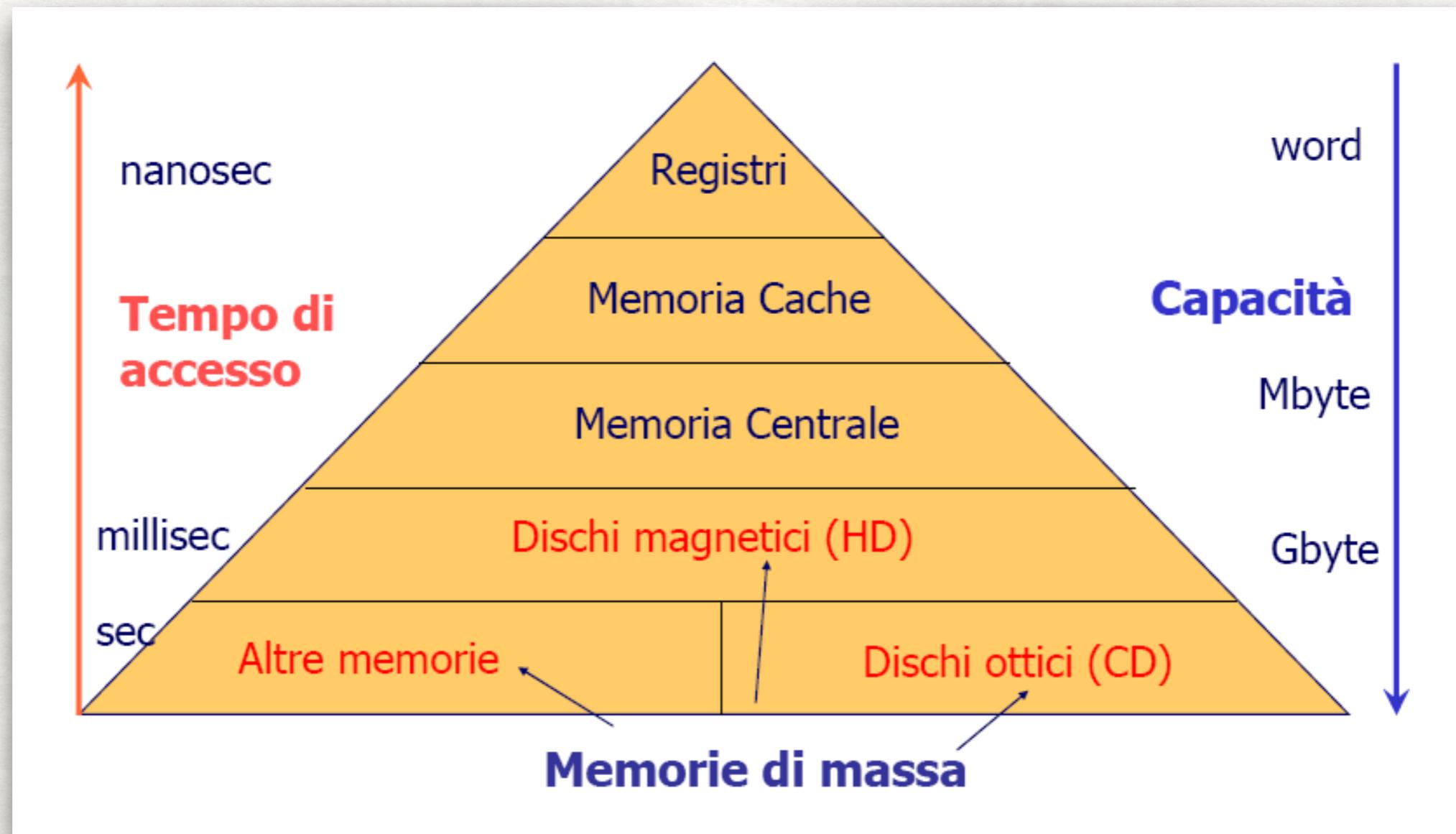
- Alcune memorie sono dedicate, ossia sono progettate per contenere specifici tipi di dati (es la EPROM o flash di bootstrap)
- Altre memorie sono general-purpose e possono essere accedute mediante specifici metodi di indirizzamento per leggere o scrivere dati di qualsiasi tipo. Il loro impiego dipende dalle loro caratteristiche e da come il dato viene impiegato durante il funzionamento del calcolatore

**Gerarchia di memorie:** La memoria del calcolatore e' organizzata gerarchicamente in modo da comprendere poche memorie a bassa capacità ed alti costi ma molto veloci e molta memoria più lenta e di capacità maggiore

# GERARCHIA DELLE MEMORIE

Nella gerarchia di memoria i dati sono temporaneamente trasferiti in memorie sempre più veloci.

L'obiettivo e' di organizzare le politiche di piazzamento ed accesso dei dati in modo tale da avere i dati più frequentemente usati, o in generale più "utili" virtualmente sempre nelle memorie più veloci.



# LOCALITÀ SPAZIALE E TEMPORALE

Regola base dell'efficienza è rendere di massima velocità il **caso più frequente**, per raggiungere questo obiettivo si sfrutta la **località spaziale e temporale**.

Principio di località: un programma in ogni istante utilizza una porzione limitata dello spazio di indirizzamento.

- **Località spaziale**: accedendo ad un dato è assai probabile che si debba accedere ad altri dati localizzati “vicino” nello spazio di indirizzamento.
- **Località temporale**: accedendo ad un dato e' assai probabile che si debba riaccedere ad esso in un tempo “vicino”.

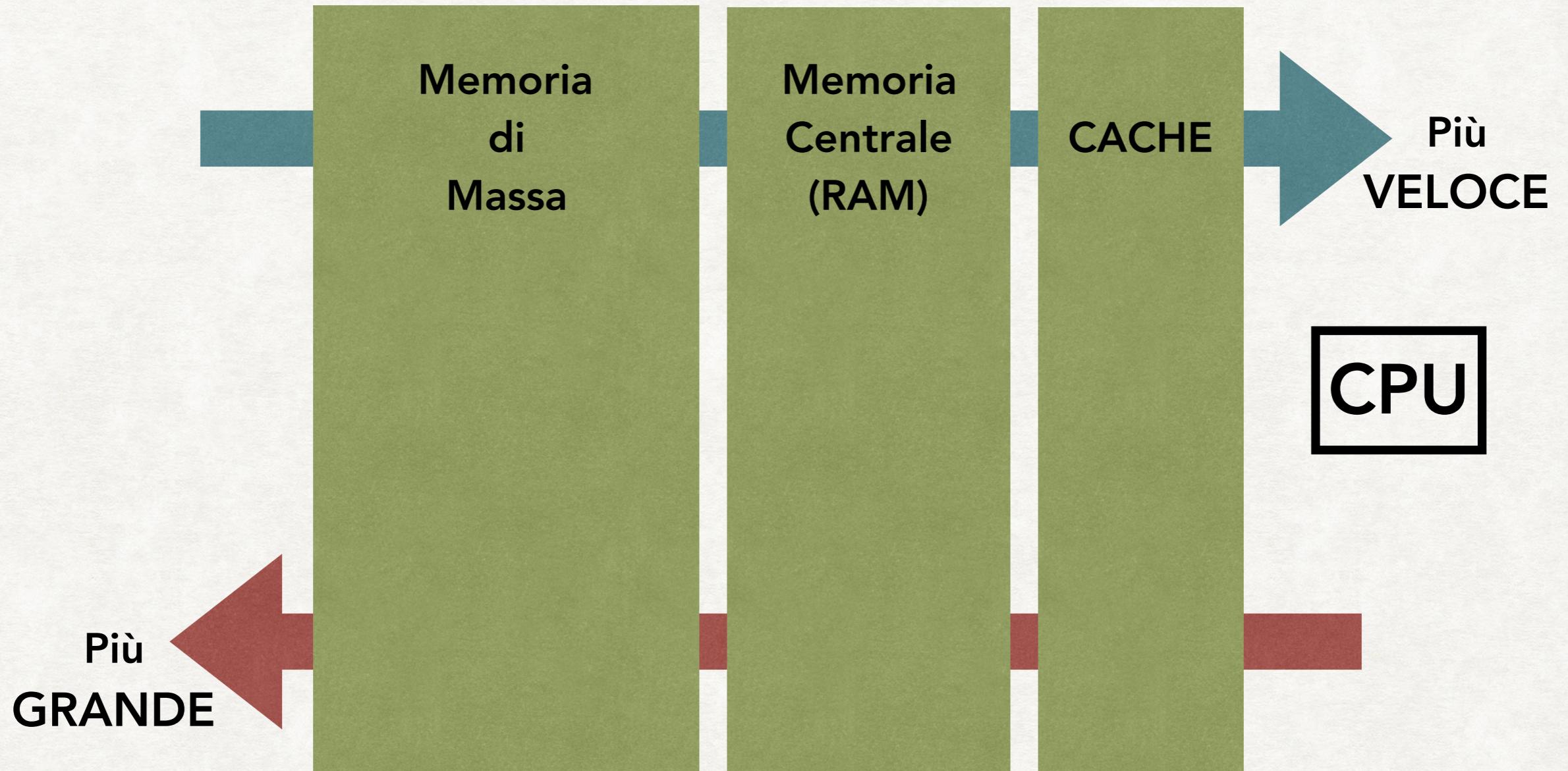
I dati che inizialmente si trovano ad un livello più basso:

- conviene spostarli in memorie più veloci (loc. temporale);
- conviene spostare anche i dati vicini (loc. spaziale).

I programmi NON vedono la gerarchia ma referenziano i dati come se fossero sempre in memoria centrale (a parte per i registri che sono nominati esplicitamente) perciò bisogna mantenere il più vicino possibile alla CPU dati utilizzati più recentemente.

La gerarchia delle memorie deve prevedere in successione memorie sempre più larghe e più lente per mantenere i dati nei livelli più alti proporzionalmente alla previsione della frequenza d'uso.

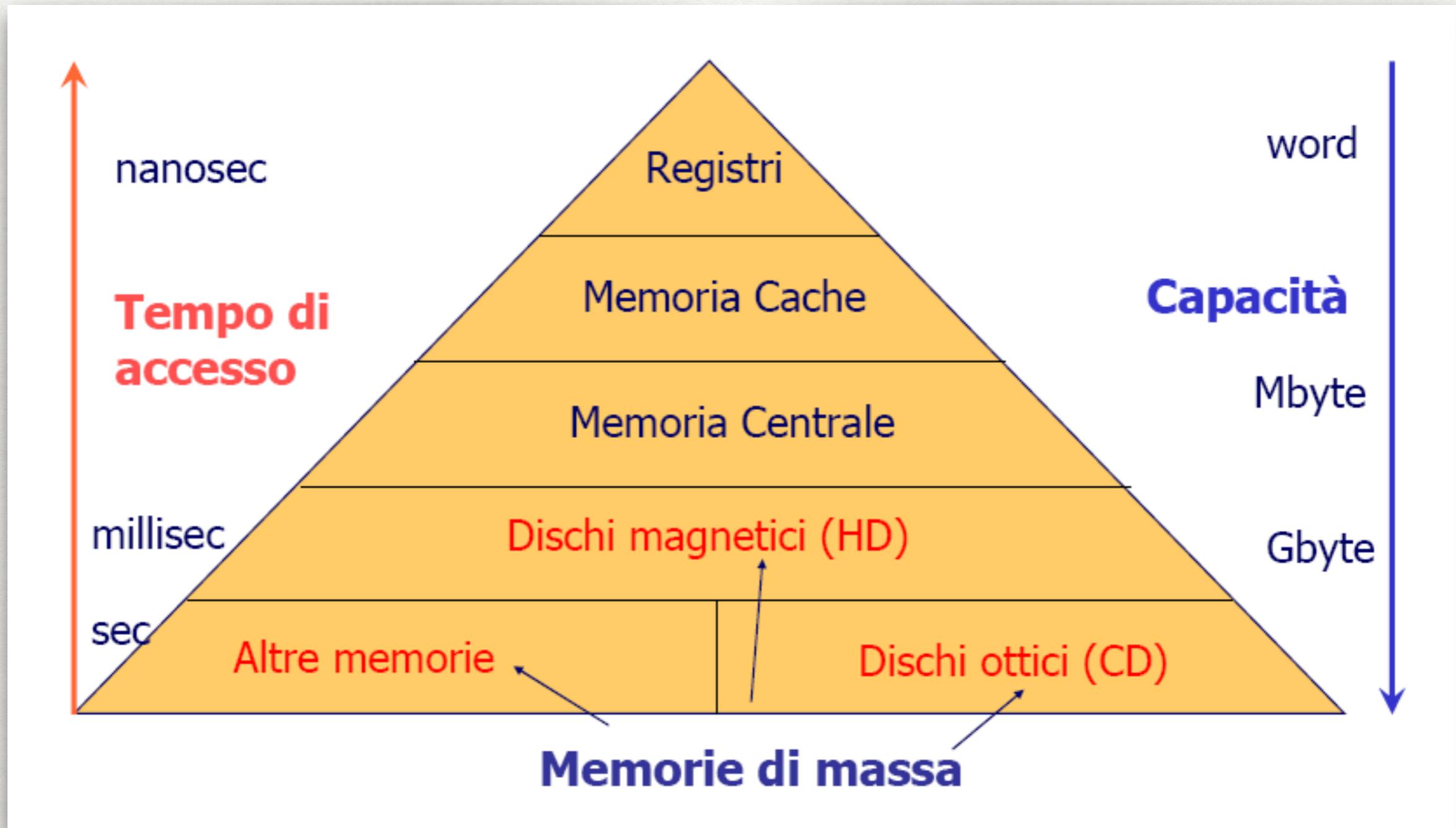
# LOCALITÀ SPAZIALE E TEMPORALE (2)



# GERARCHIA DI MEMORIA: PIÙ LIVELLI

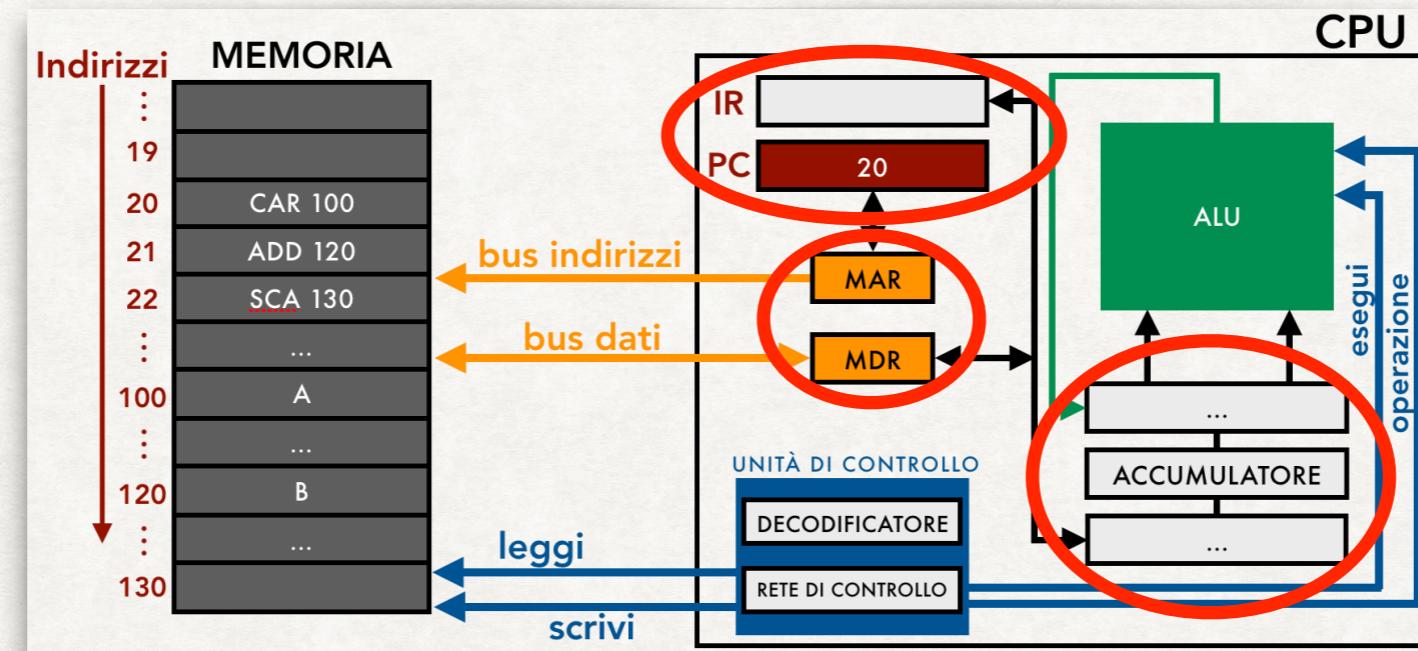
- **Blocco:** unità di informazione minima scambiata fra livelli (ex: 32 bytes)
- **Hit:** un accesso alla memoria che trova l'informazione cercata nel livello superiore
- **Hit Rate:** frequenza di accessi trovati nel livello superiore ( $h$ )
- **Miss Rate:**  $(1 - h)$  caso di insuccesso (non trovo il dato nella memoria adiacente)
- **Hit Time:** tempo di accesso al livello superiore  $T_h$
- **Miss Penalty:** tempo per rimpiazzare un blocco nel livello superiore più il tempo per fornirlo alla CPU. Il "miss penalty", può anche essere scomposto nei tempi per reperire il primo dato di un blocco più il tempo per trasferire gli altri. Comprende certamente un  $T_h$  e un "overhead" nel caso di miss ( $T_{miss}$ ).
- Il  $T_{miss}$  comprende il tempo per reperire il blocco nella gerarchia di livello più basso più (eventualmente trascurabile) il tempo per rimpiazzare i dati dal livello alto al livello più basso (se il livello più alto deve essere "svuotato", cioè se devo salvare i dati che sono in esso presenti, operazione di write-back)

# GERARCHIA DELLE MEMORIE



# 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.).

# MEMORIE CACHE

Il termine "cache" deriva dal francese (pronuncia in francese "cascé", in inglese "kesh"), e significa "nascosto".

La memoria cache è una memoria veloce relativamente piccola, non visibile al software, cioè completamente gestita dall'hardware, che memorizza i dati più recentemente usati della memoria principale.

Il meccanismo è semplice: Il processore interagisce direttamente SOLO con la cache. Quando il processore richiede una cella non presente nella cache (miss) la cella viene trasferita dalla memoria nella cache.

Occorre definire dei meccanismi per:

1. Conoscere se un dato è nella cache
2. Nel caso in cui il dato sia presente, conoscere la sua posizione ed accedervi.

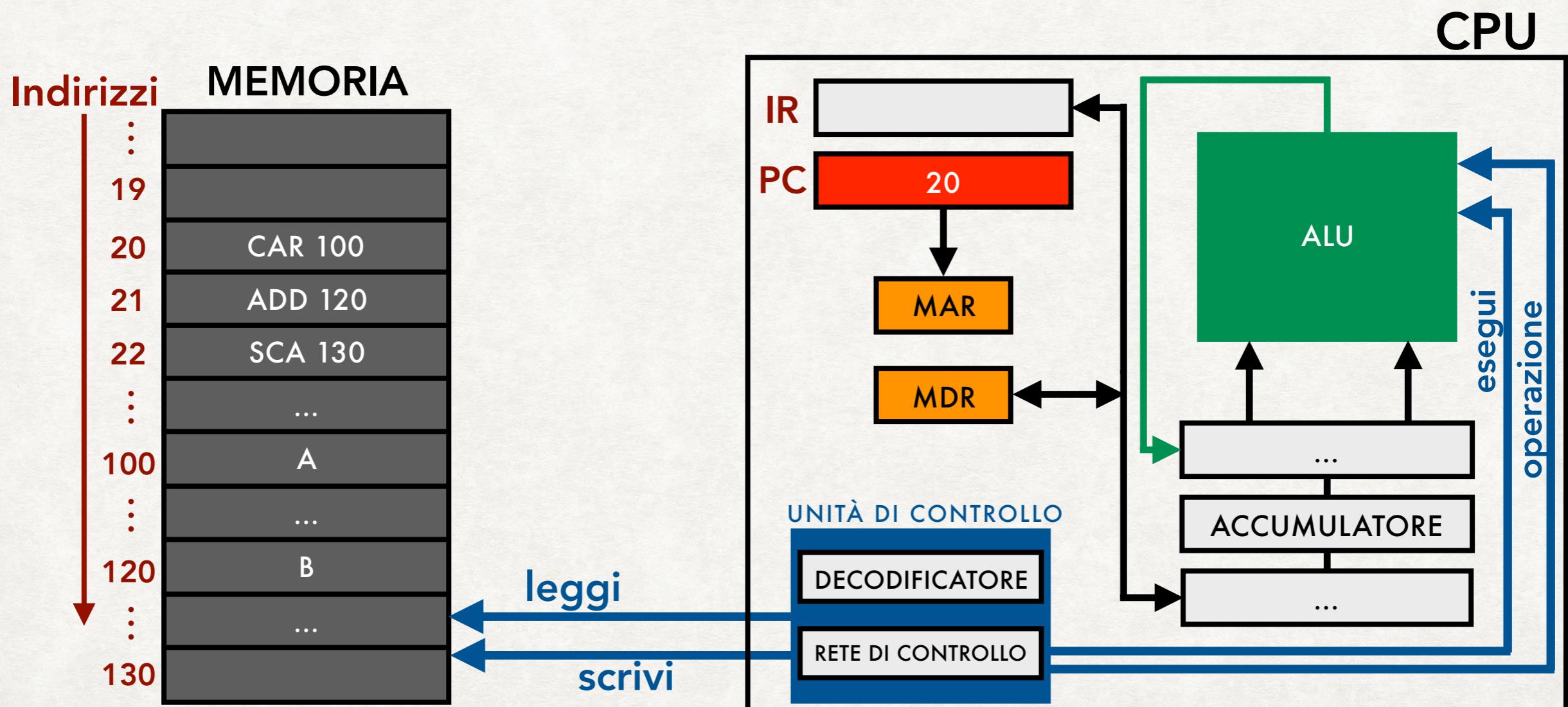
Queste due operazioni devono essere eseguite nel minor tempo possibile, poiché la velocità con cui si riesce ad accedere ai dati nella cache influisce drasticamente sulle prestazioni dell'intero sistema di memoria.

Il termine cache è stato usato per la prima volta per indicare il livello della gerarchia tra la CPU e la memoria principale, ma è oggi utilizzato per indicare qualsiasi tipo di gestione della memoria che tragga vantaggio dalla località degli accessi.

# MEMORIA CENTRALE

La **MEMORIA CENTRALE** (o **PRINCIPALE** o **RAM**) è la componente in cui si immagazzinano e da cui si accedono dati e programmi.

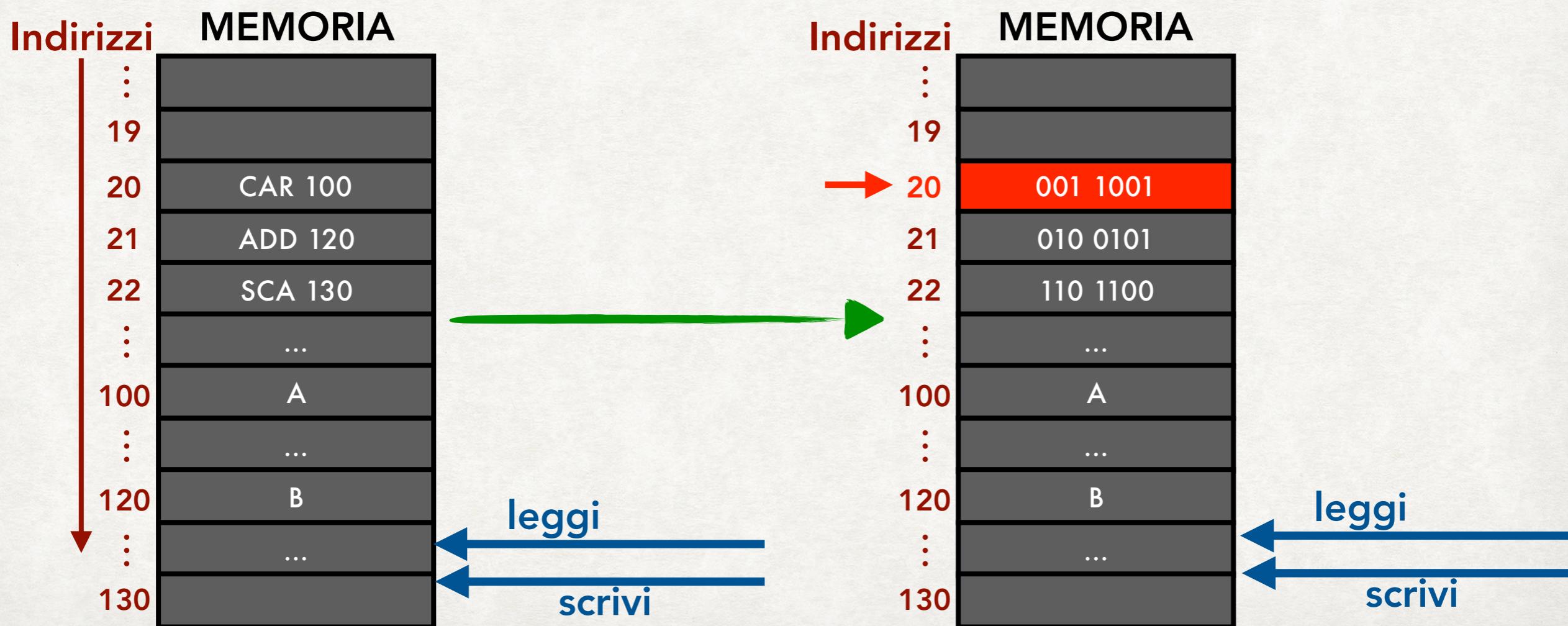
È costituita da celle (o locazioni) ed ogni cella può contenere una quantità fissata di memoria (**NUMERO di BIT**), detta **parola**.



# MEMORIA CENTRALE

Ogni cella è caratterizzata da un **INDIRIZZO** e da un **VALORE**. L'indirizzo è un numero che identifica la cella e ne consente l'accesso, il valore è la sequenza di bit memorizzata nella cella.

La memoria fornisce operazioni di Lettura e Scrittura: la prima permette la consultazione del valore di una cella a partire da un dato indirizzo, la seconda permette la modifica del valore di una cella a partire da un dato indirizzo.



# MEMORIE RAM E MEMORIE ROM

- ♦ Le memorie **RAM** (random access memory):
  - possono essere accedute sia in lettura che in scrittura e sono volatili
  - i dati memorizzati vengono persi allo spegnimento del calcolatore
- ♦ La memoria **ROM** (read only memory)
  - sono persistenti
  - mantengono il contenuto anche senza alimentazione
  - permettono solo la lettura dei dati (o programmi)
  - memorizzano alcuni programmi di sistema (firmware)
- ♦ Evoluzioni delle memorie ROM:
  - PROM (scritte una sola volta) EPROM (scritte più volte)

RAM è la memoria in cui sono caricati i dati che devono essere utilizzati dal calcolatore per elaborare (Memoria Principale). Il processore identifica le celle della RAM tramite indirizzi assegnati che ne specificano la posizione (la memoria si rappresenta come una stringa ordinata di elementi detti celle o parole). Il tempo di accesso alle celle della memoria è costante.

ROM è una memoria permanente presente sulla scheda madre, che contiene il BIOS (Basic I/O System), cioè le istruzioni che la CPU deve caricare per consentire l'avvio del sistema.

# MEMORIA SECONDARIA

In un calcolatore esistono diverse memorie secondarie (o memorie di massa) che hanno le seguenti caratteristiche:

- alta capacità
- bassi costi
- non volatili



# MEMORIA SECONDARIA

Nastri magnetici (lettura e scrittura sequenziale)

Dischi magnetici (accesso diretto ai dati)

Floppy disk



Dischi ottici (CD-ROM, CDW, DVD)



USB FLASH DISK (pen drive, hd...)

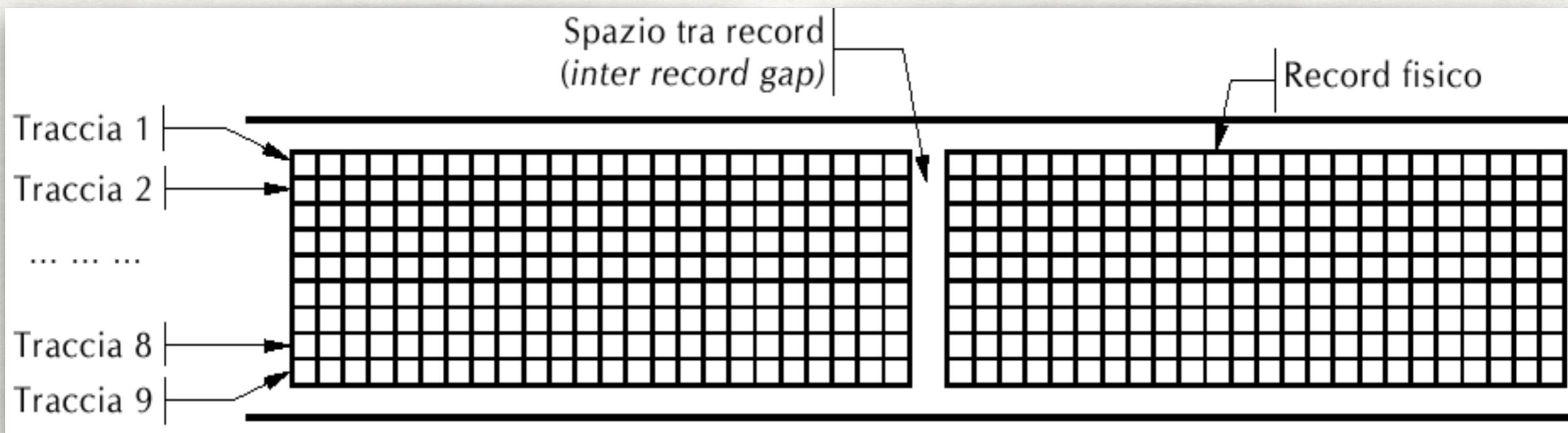


SSD (Solid State Drives)



# NASTRI MAGNETICI

- Primi dispositivi di memoria di massa impiegati storicamente
- Notevoli analogie con i comuni nastri impiegati nelle audio cassette (ma informazioni registrate in formato digitale)
- Accesso sequenziale
- Molto lenti
- Utili per operazioni di backup

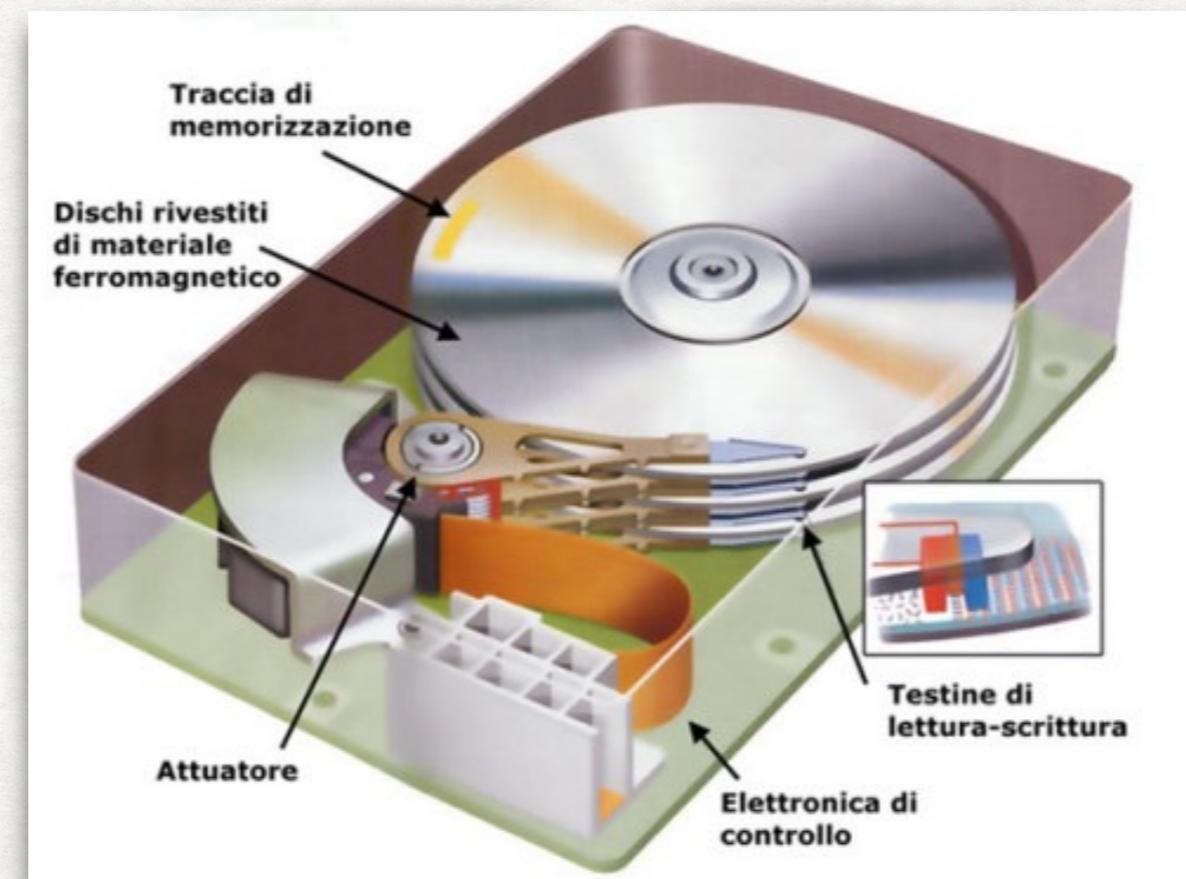


# DISCHI MAGNETICI

Piatti d'alluminio ricoperti di materiale ferromagnetico

Testina di un disco

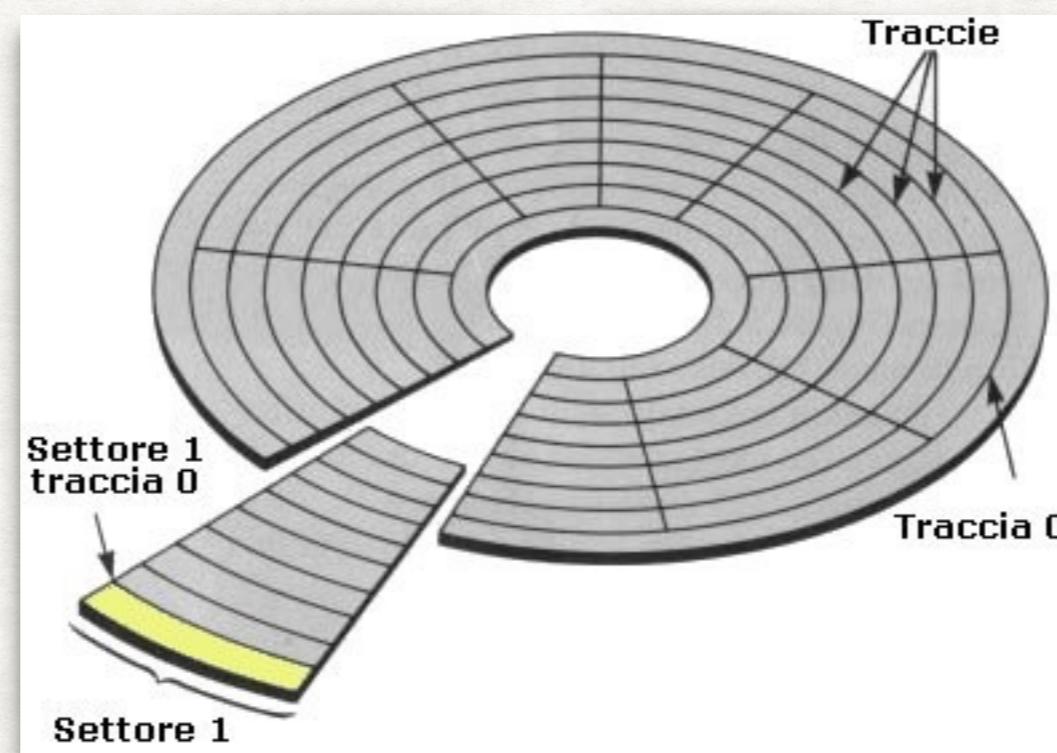
- sospesa appena sopra la superficie magnetica
- scrittura: il passaggio di corrente positiva o negativa attraverso la testina magnetizza la superficie
- lettura: il passaggio sopra un'area magnetizzata induce una corrente positiva o negativa nella testina



# TRACCE E SETTORI

Traccia (track): sequenza circolare di bit scritta mentre il disco compie una rotazione completa. Tra una traccia e l'altra c'è un piccolo spazio di separazione (gap) .

Settore (sector): parte di una traccia corrispondente a un settore circolare del disco. Tra settori consecutivi si trova un piccolo spazio (intersector gap).



Formattazione: operazione che predisponde tracce e settori per la lettura/scrittura.

# FLOPPY DISK

Storicamente, i più diffusi avevano un diametro di 3,5 pollici, una capacità di 1,44 Mbyte e formattati con 80 tracce e 18 settori.

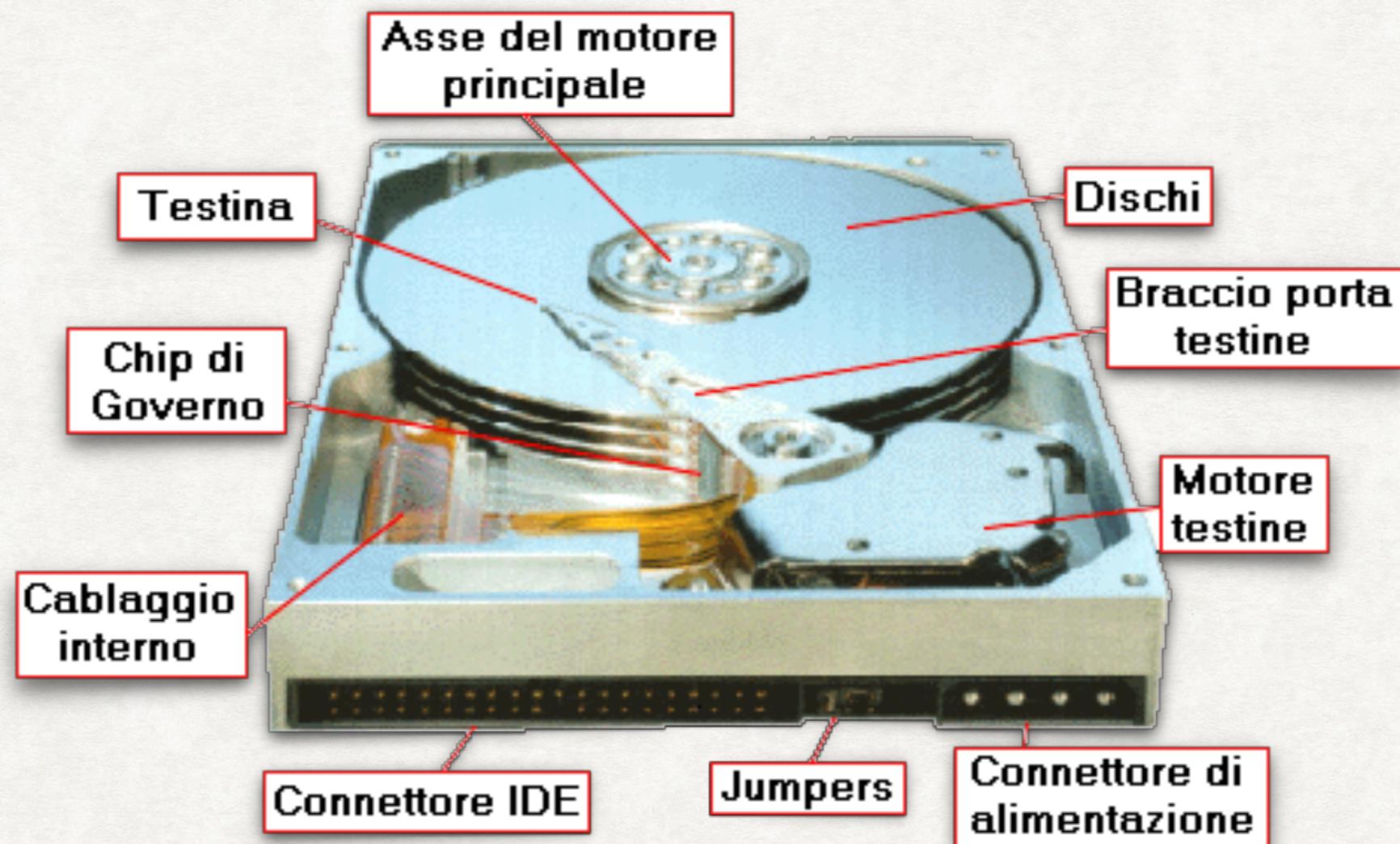
## Limite

Capacità ormai inadeguata per la dimensione media dei dati e dei programmi che vengono utilizzati oggi. Piuttosto deteriorabili

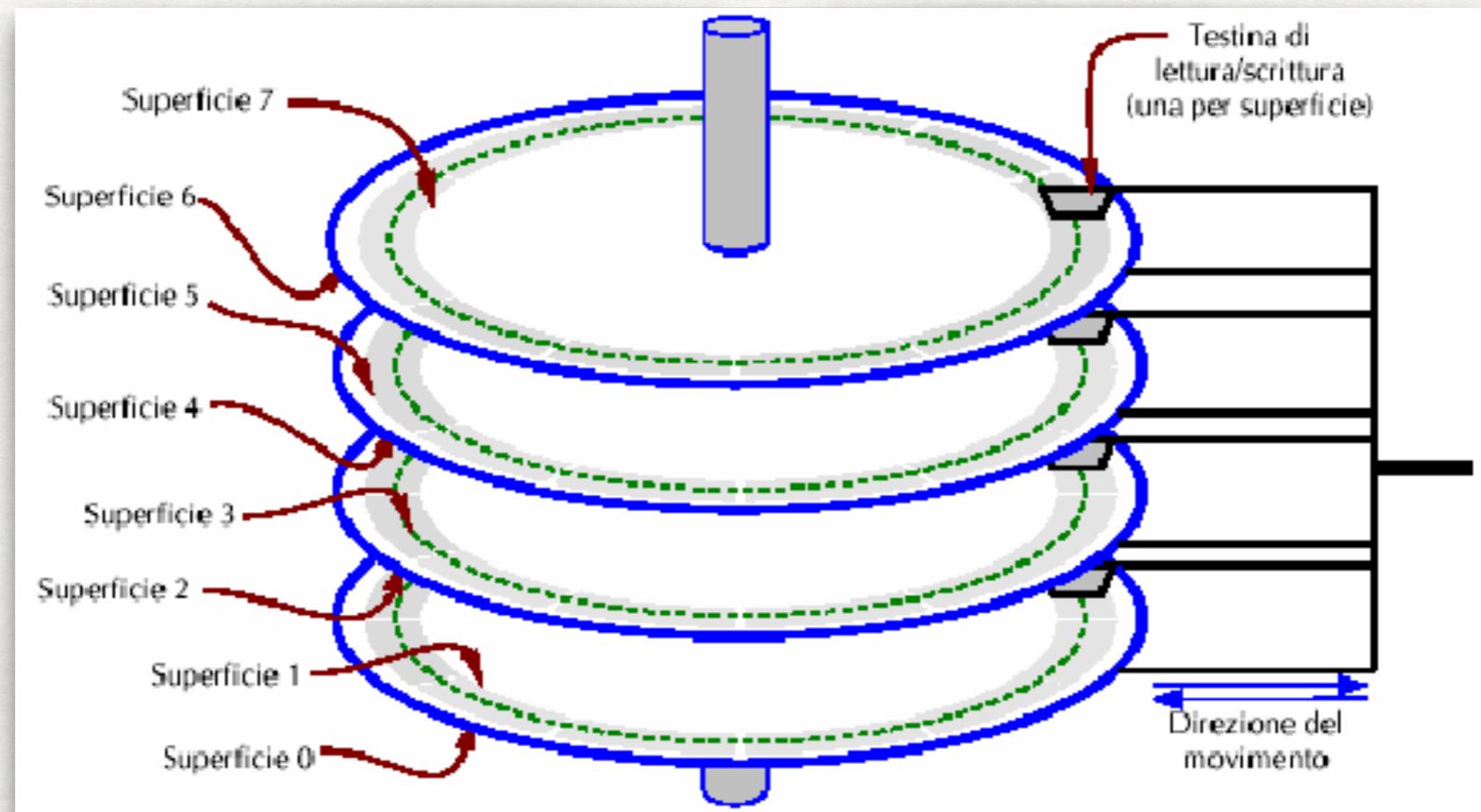


# HARD DISK

In un disco rigido i dati sono memorizzati in diversi platters, ognuno dei quali è organizzato in tracce e settori. Entrambe le facce sono utilizzate



# HARD DISK (2)



**Tempo di accesso:** intervallo tra la richiesta di un dato e il completamento dell'operazione. E' la somma dei seguenti tempi:

**Tempo di ricerca:** tempo necessario a posizionare la testina sulla traccia

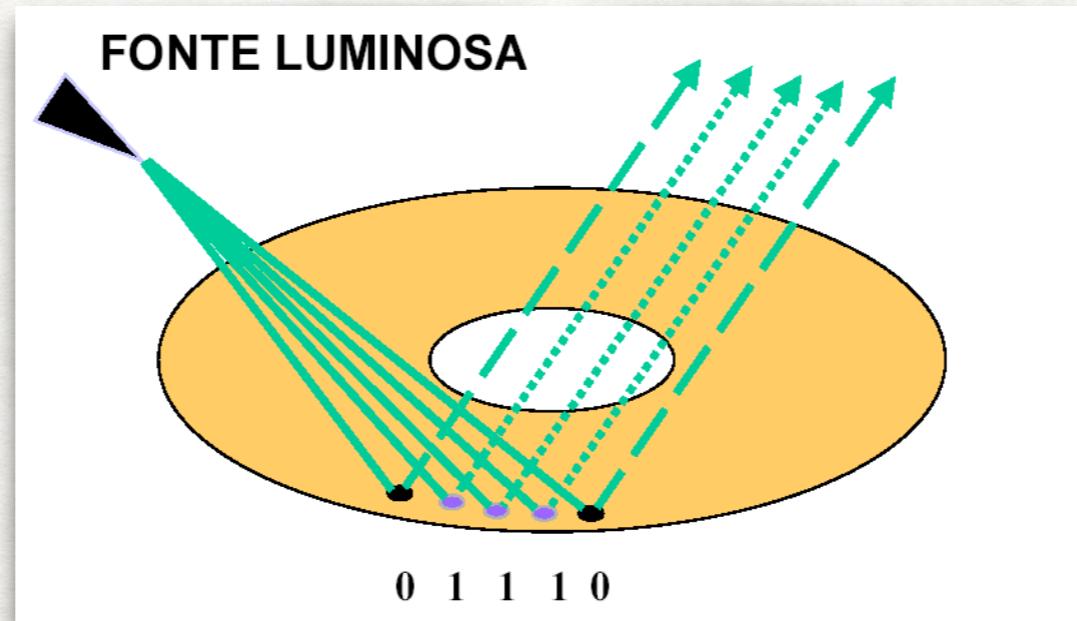
**Tempo di latenza:** tempo necessario affinché il settore desiderato si posizioni in corrispondenza della testina.

**Tempo di trasferimento:** tempo necessario per il trasferimento dei dati dal disco alla memoria o viceversa

# DISCHI OTTICI

Le tecnologie dei dischi ottici sono basate sull'uso di raggi laser  
Il raggio laser è un particolare tipo di raggio luminoso estremamente focalizzato che può essere emesso in fasci di dimensioni molto ridotte

Il raggio laser viene riflesso in modo diverso da superfici diverse e si può pensare di utilizzare delle superfici con dei piccolissimi forellini



Ogni unità di superficie può essere forata o non forata e questo corrisponde ai due diversi tipi di informazione elementare (bit)

L'informazione contenuta su un'unità di superficie può essere letta guardando la riflessione del raggio laser proiettato sulla superficie stessa

Aggregazioni di informazioni possono essere ottenute dividendo una superficie di grandi dimensioni in molte unità elementari, ognuna delle quali rappresenta un singolo bit

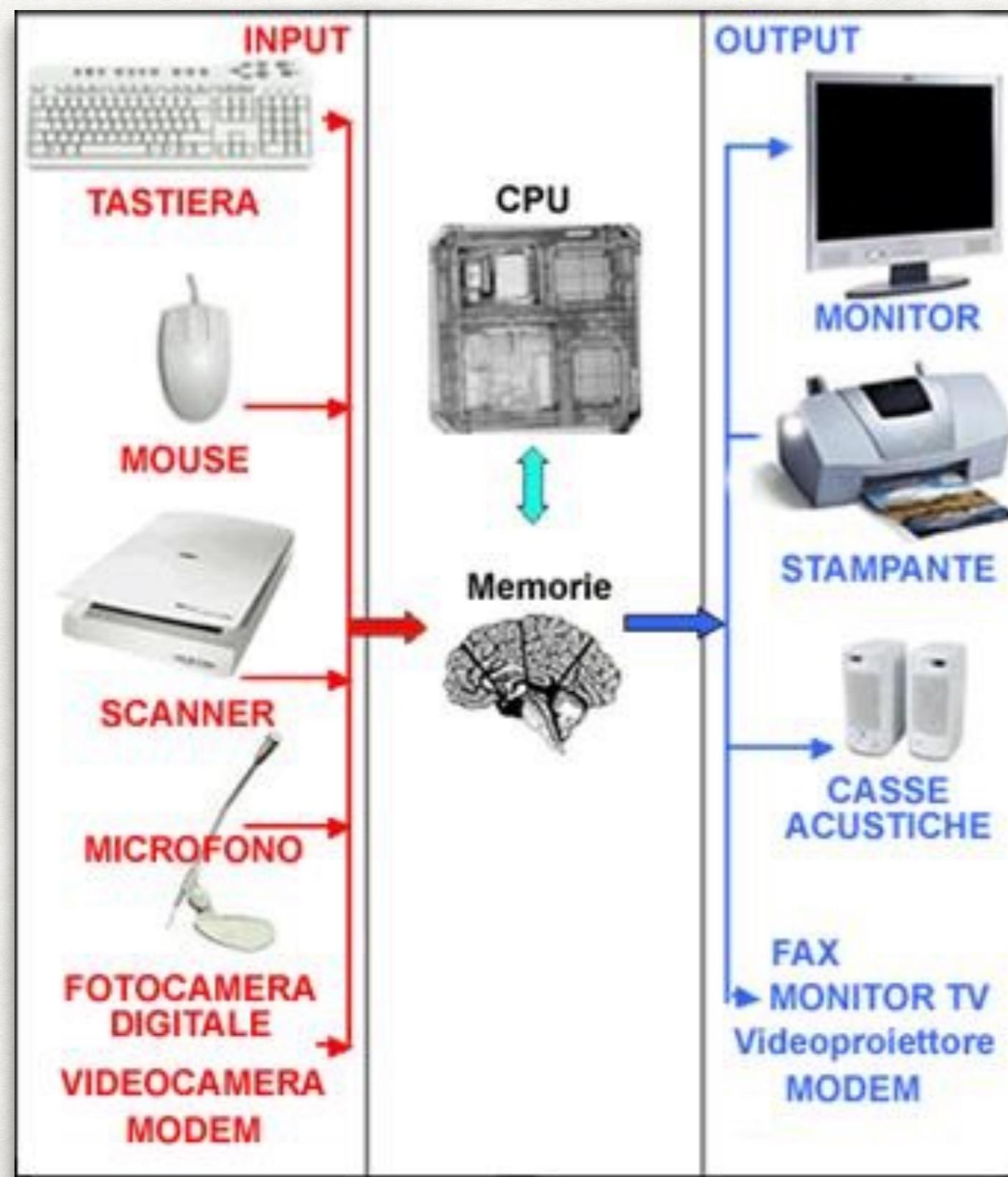
# UNITÀ PERIFERICHE

**Sottosistema di I/O:** Permette di interfacciare il calcolatore verso il mondo esterno:

- Dispositivi di INPUT
- Dispositivi di OUTPUT



# DISPOSITIVI DI INPUT/OUTPUT



# CELLE DI MEMORIA

Un'istruzione o i dati presenti in memoria sono espressi utilizzando il **codice binario**.

Indirizzo della cella

codice binario

Indirizzo della cella

codice operativo

indirizzo - operando

20

CAR

100

20

001

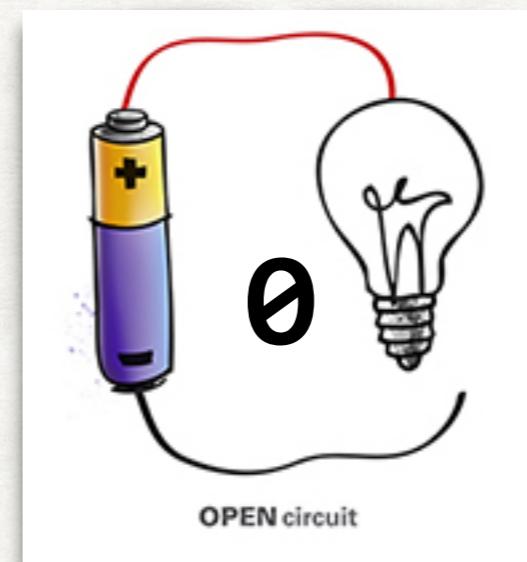
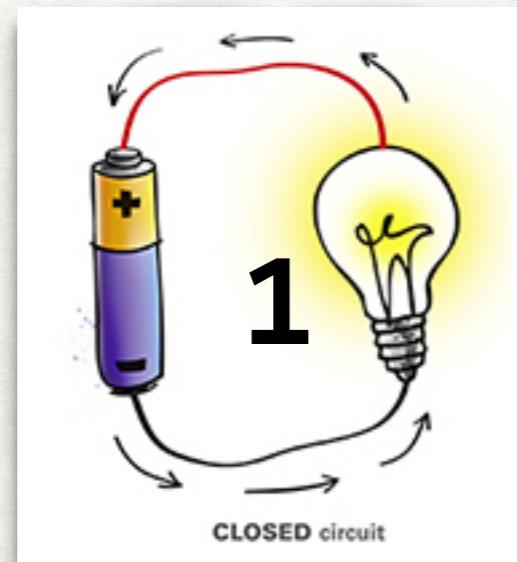
1001

# LA CODIFICA BINARIA

All'interno di un computer i dati non sono composti da numeri o lettere così come li digitiamo da tastiera. Il computer "comprende" solo impulsi elettrici, per cui tutti i dati che inseriamo (numeri, lettere, segni di interpunkzione, etc.) devono essere convertiti in impulsi elettrici affinché possano essere memorizzati.

Ogni impulso elettrico si definisce **Bit** (*binary digit* = cifra binaria).

Il bit può essere immaginato come una lampadina che ha solo due stati: accesa e spenta.



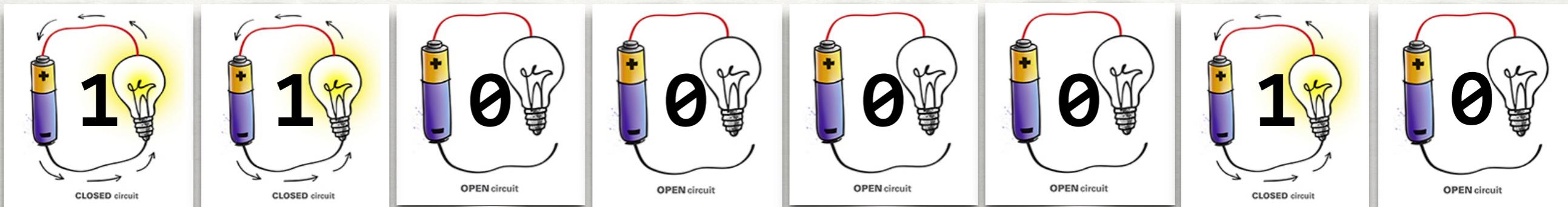
Nel primo caso il bit si trova nello stato **1** (che significa passaggio di corrente), mentre nel secondo caso si trova nello stato **0** (che significa non passaggio di corrente).

Pertanto gli unici simboli che il computer riconosce sono, quindi, le cifre **0** ed **1**.

# LA CODIFICA BINARIA (2)

Per rappresentare un qualsiasi simbolo, però, un singolo bit non basta; per questo solitamente si usano raggruppamenti di più bit.

Un raggruppamento di **8 bit** prende il nome di **byte**. Un byte, quindi, corrisponde ad una "parola" composta da otto bit.



Con un byte, quindi, è possibile codificare  $2^8 = 256$  diverse alternative; può quindi rappresentare un carattere scelto da un alfabeto di 256 simboli, un numero compreso tra 0 e 255, un colore scelto da una "tavolozza" di 256 colori diversi, e così via.

Bit e Byte sono le unità di misura di base per esprimere la capacità (la "dimensione") di una memoria. Man mano che le dimensioni delle memorie crescono si deve far riferimento ai multipli del byte.

# CAPACITÀ DI MEMORIA

Per rappresentare un qualsiasi simbolo, però, un singolo bit non basta; per questo solitamente si usano raggruppamenti di più bit.

Un raggruppamento di 8 bit prende il nome di byte. Un byte, quindi, corrisponde ad una "parola" composta da otto bit.

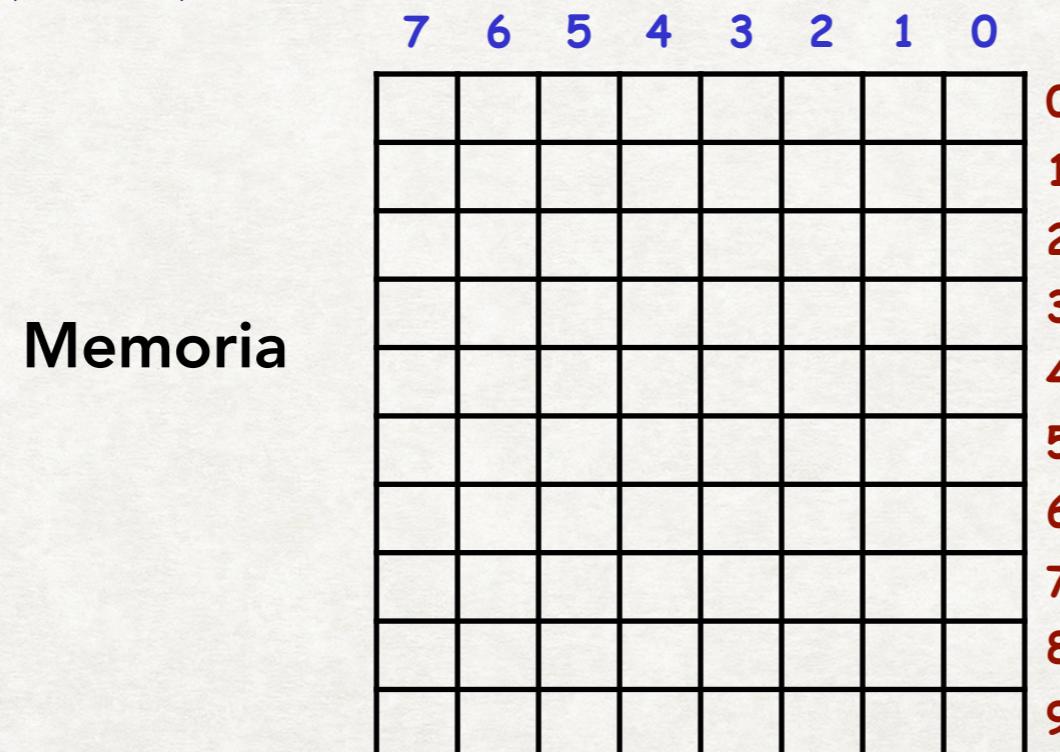
Unità di misura	Simbolo	Equivale a ...
Byte	B	8 bit
KiloByte	KB	$2^{10}$ byte = 1024 byte
MegaByte	MB	$2^{10}$ KB = 1048576 byte = $2^{20}$ byte
GigaByte	GB	$2^{10}$ MB = $2^{30}$ byte
TeraByte	TB	$2^{10}$ GB = $2^{40}$ byte
PetaByte	PB	$2^{10}$ TB = $2^{50}$ byte

n bit posso rappresentare  $2^n$  "parole"

Per rappresentare m parole ho bisogno di  $\log_2 m$  bit

# INDIRIZZI DI MEMORIA

- ❖ I bit nelle memorie sono raggruppati in celle:
  - tutte le celle sono formate dallo stesso numero di bit, una cella composta da  $k$  bit è in grado di contenere una qualunque tra  $2^k$  combinazioni diverse di bit
- ❖ Ogni cella ha un indirizzo:
  - serve come accesso all'informazione in una memoria con  $N$  celle gli indirizzi vanno da 0 a  $N-1$
- ❖ La cella è l'unità indirizzabile più piccola:
  - costituita da 8, 16, 32 o 64 bit
- ❖ Una successione di bit avente lunghezza pari al numero di elementi binari di un cella è chiamata **parola (word)**



# ORGANIZZAZIONE DELLA MEMORIA

Anche gli indirizzi della memoria sono rappresentati come numeri binari

**Il numero di bit nell'indirizzo determina il numero massimo di celle indirizzabili nella memoria ed è indipendente dal numero di bit per cella** (una memoria con 212 celle richiede sempre 12 bit di indirizzo, quale che sia la dimensione della cella)

# ORGANIZZAZIONE DELLA MEMORIA

Anche gli indirizzi della memoria sono rappresentati come numeri binari

**Il numero di bit nell'indirizzo determina il numero massimo di celle indirizzabili nella memoria ed è indipendente dal numero di bit per cella** (una memoria con 212 celle richiede sempre 12 bit di indirizzo, quale che sia la dimensione della cella)

## ESEMPIO:

Con 96 bit possiamo avere: **6 parole (6 celle) di 16 bit ognuna ( $6 \times 16 = 96$ )**

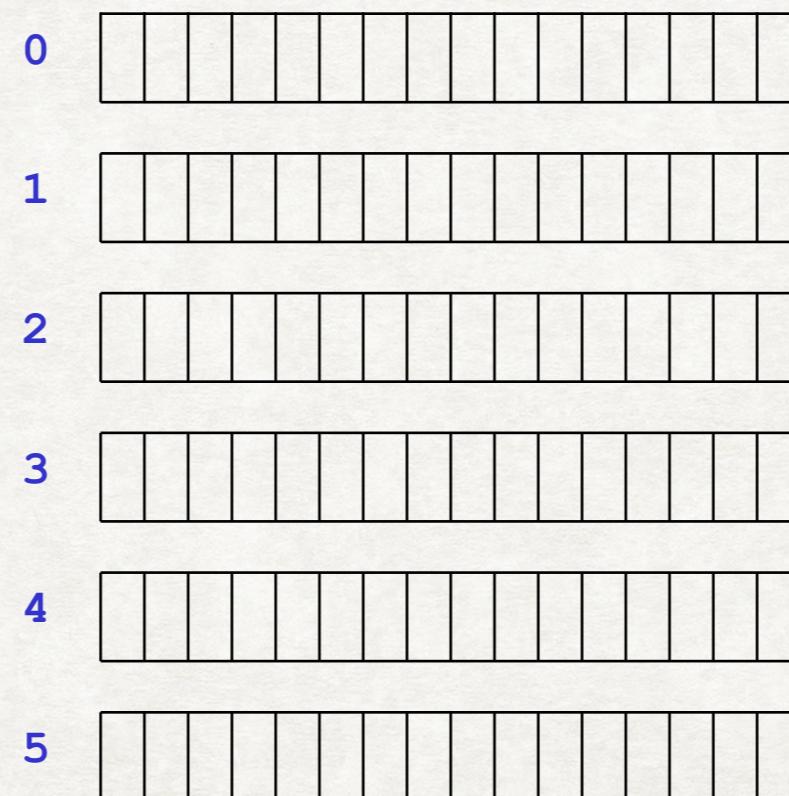
# ORGANIZZAZIONE DELLA MEMORIA

Anche gli indirizzi della memoria sono rappresentati come numeri binari

**Il numero di bit nell'indirizzo determina il numero massimo di celle indirizzabili nella memoria ed è indipendente dal numero di bit per cella** (una memoria con 212 celle richiede sempre 12 bit di indirizzo, quale che sia la dimensione della cella)

## ESEMPIO:

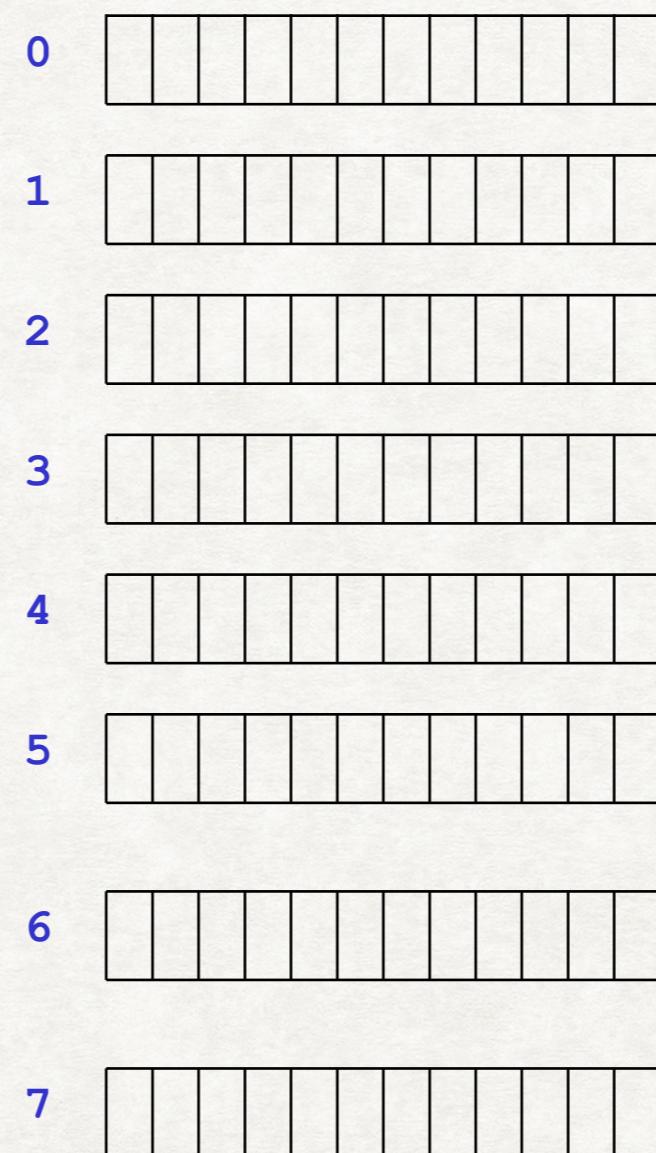
Con 96 bit possiamo avere: **6 parole (6 celle) di 16 bit ognuna ( $6 \times 16 = 96$ )**



# ORGANIZZAZIONE DELLA MEMORIA (2)

## ESEMPIO 2:

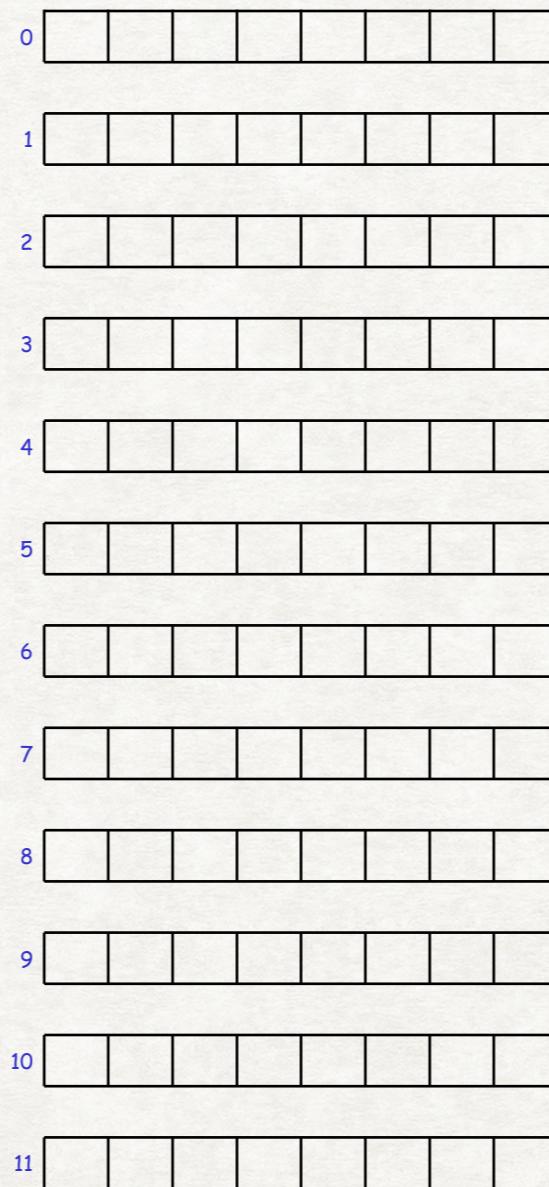
Con 96 bit possiamo avere: **8 parole (8 celle) di 12 bit ognuna ( $8 \times 12 = 96$ )**



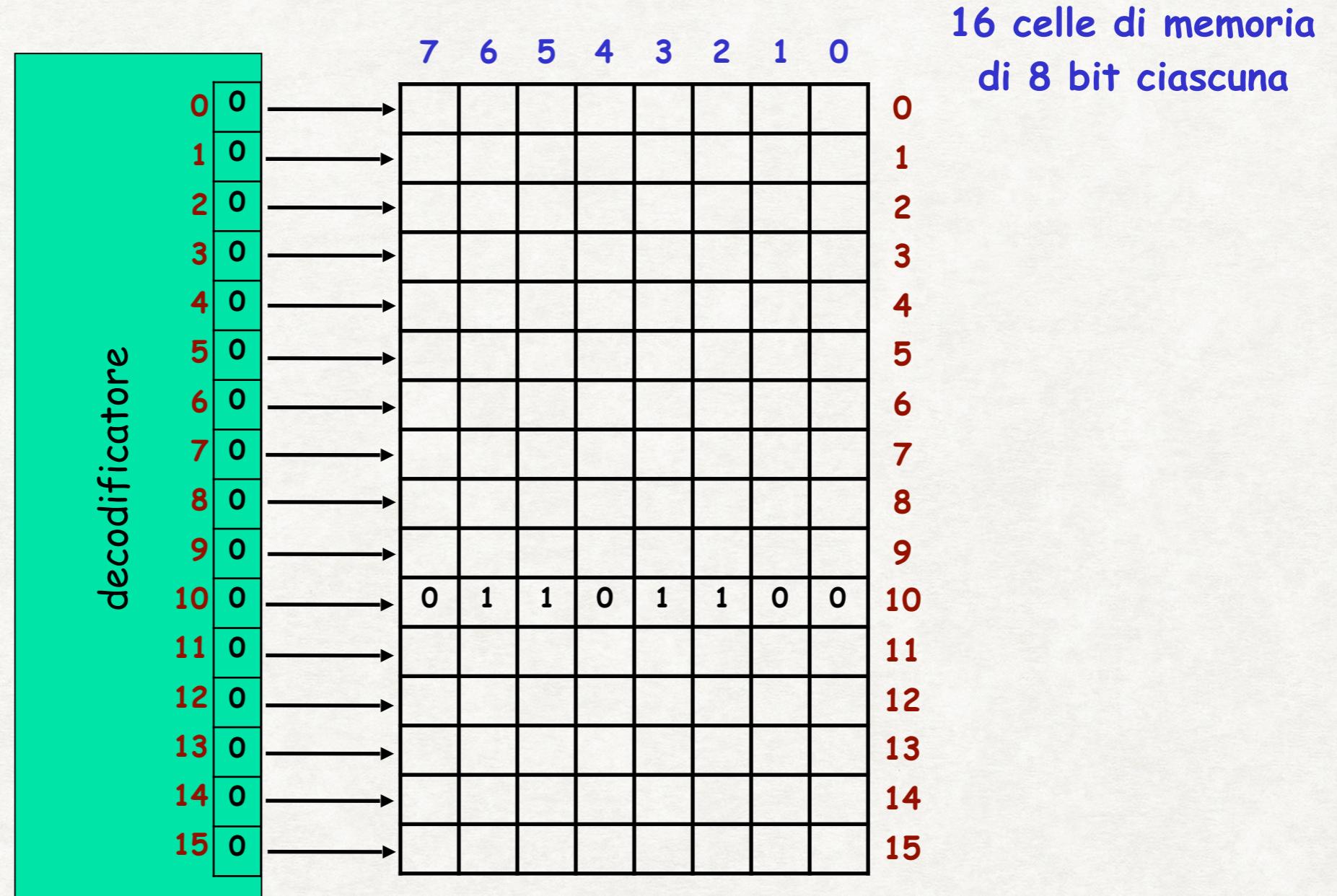
# ORGANIZZAZIONE DELLA MEMORIA (3)

## ESEMPIO 3:

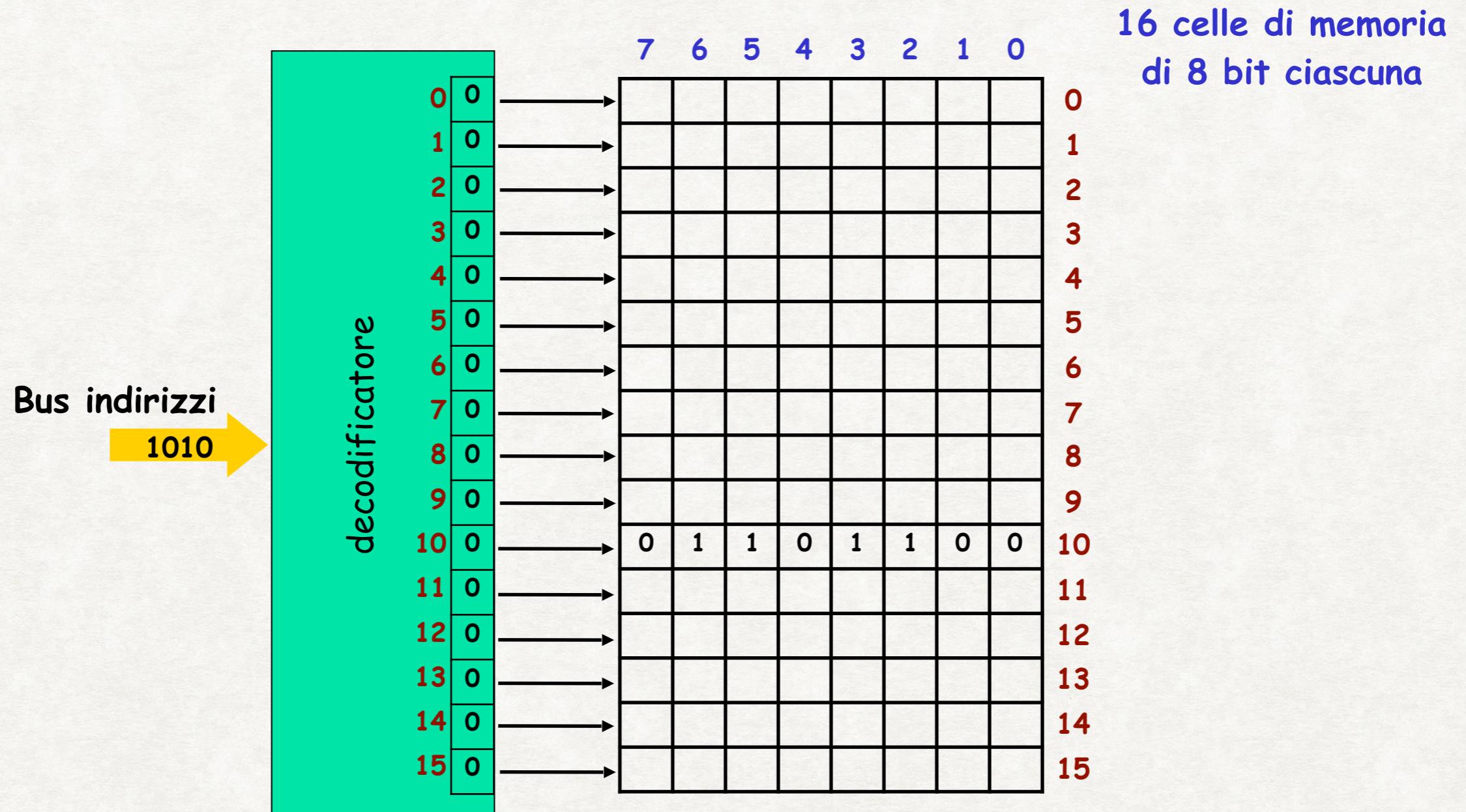
Con 96 bit possiamo avere: **12 parole (12 celle) di 8 bit ognuna ( $12 \times 8 = 96$ )**



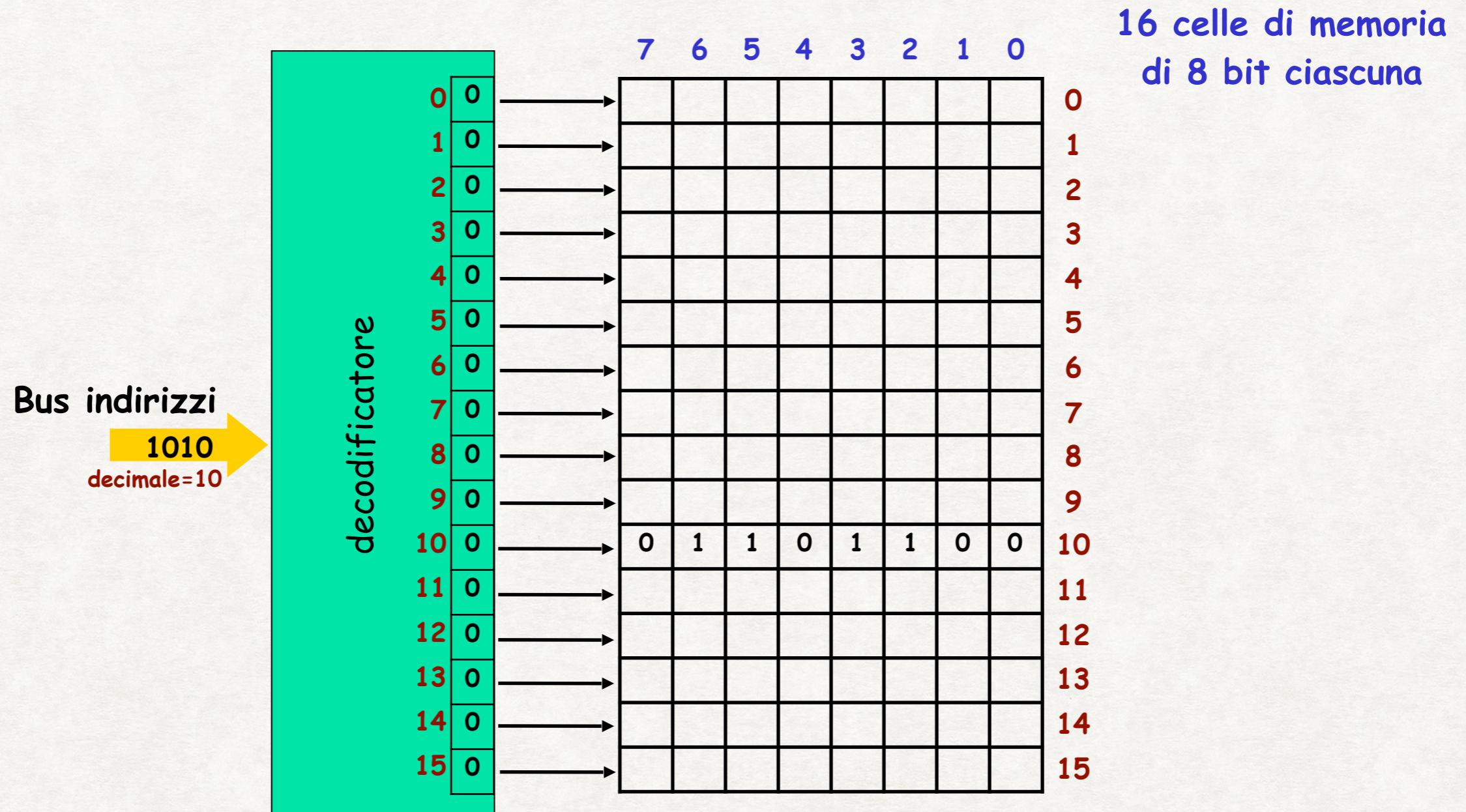
# SCHEMA DELLA MEMORIA



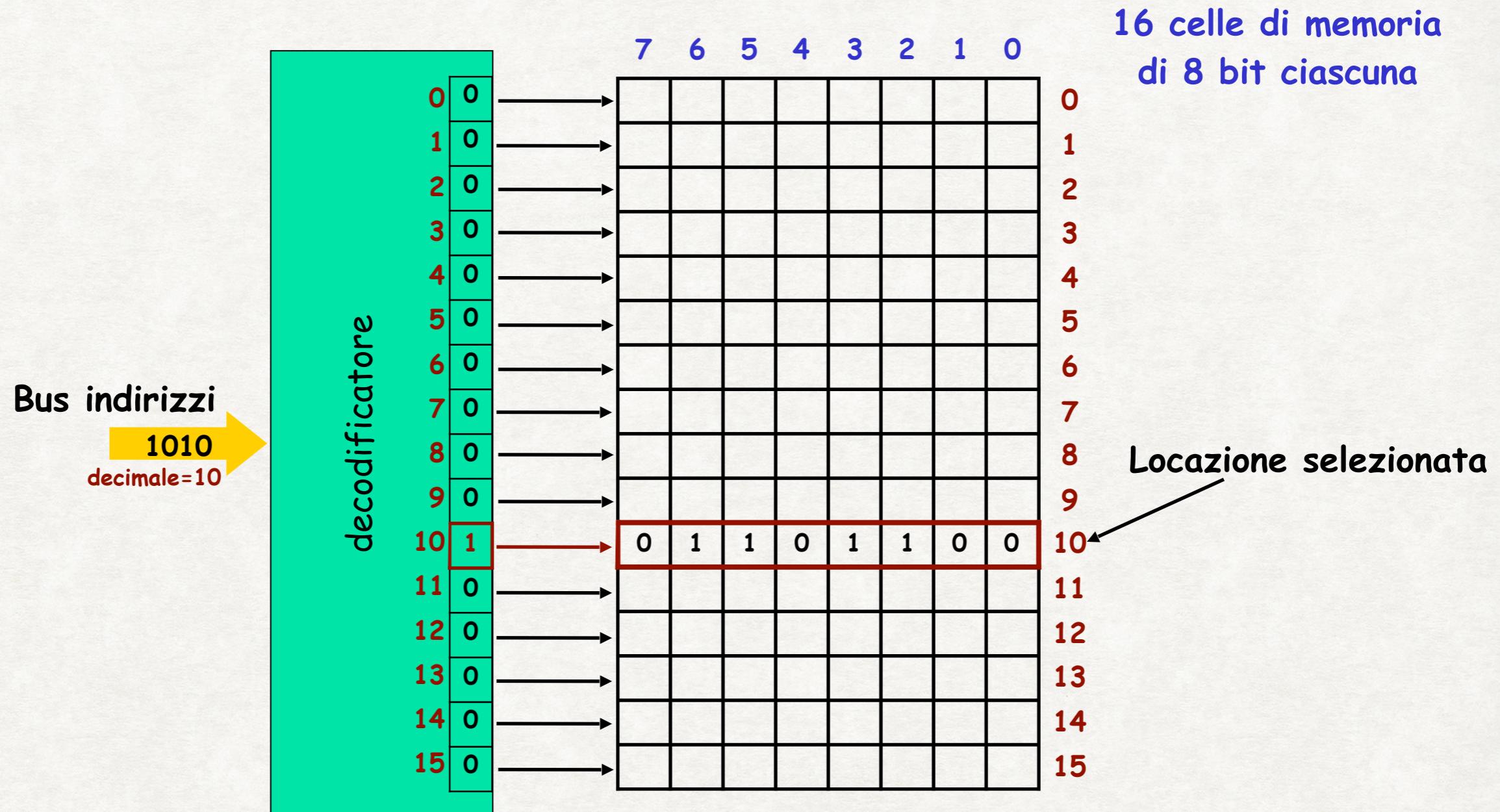
# SCHEMA DELLA MEMORIA



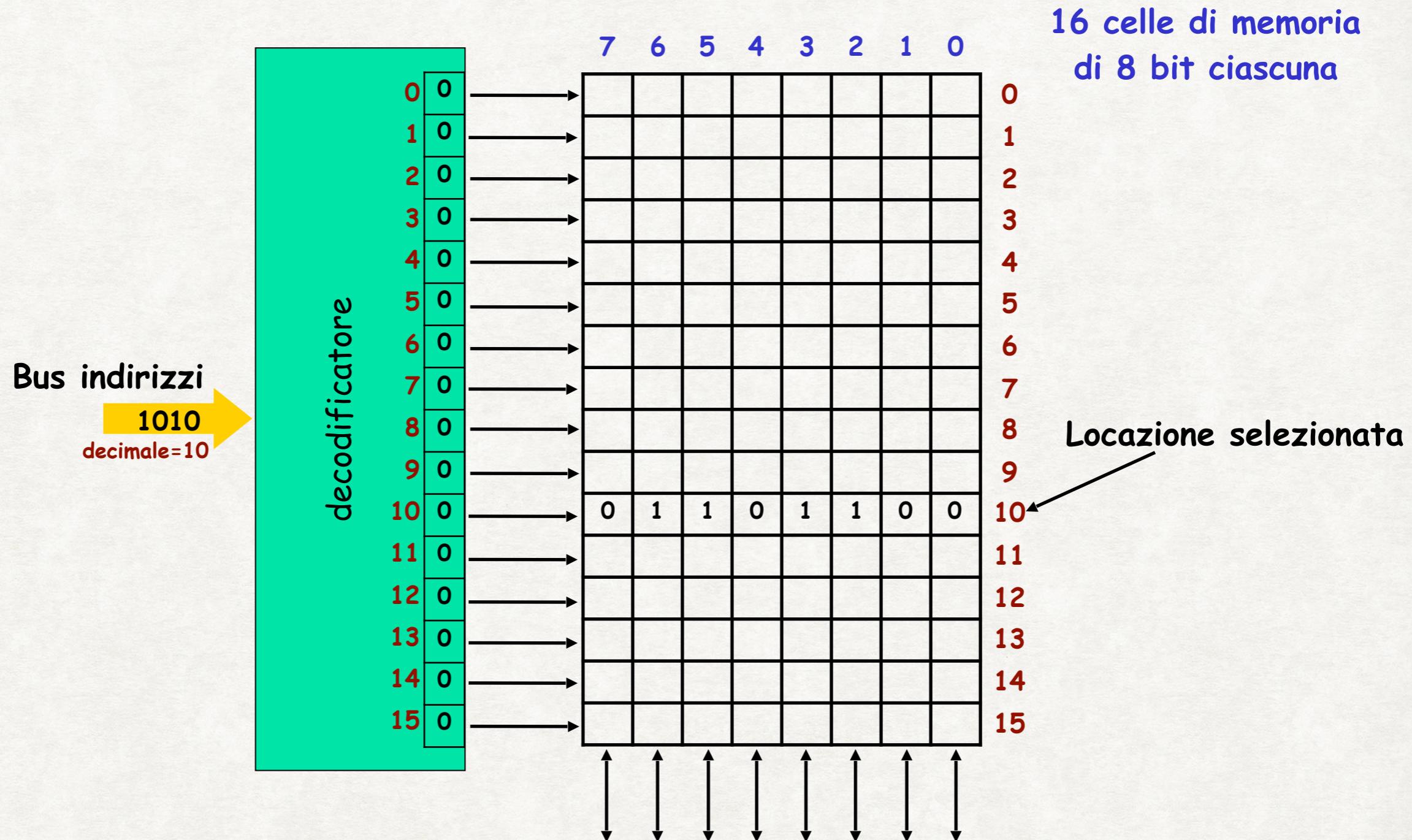
# SCHEMA DELLA MEMORIA



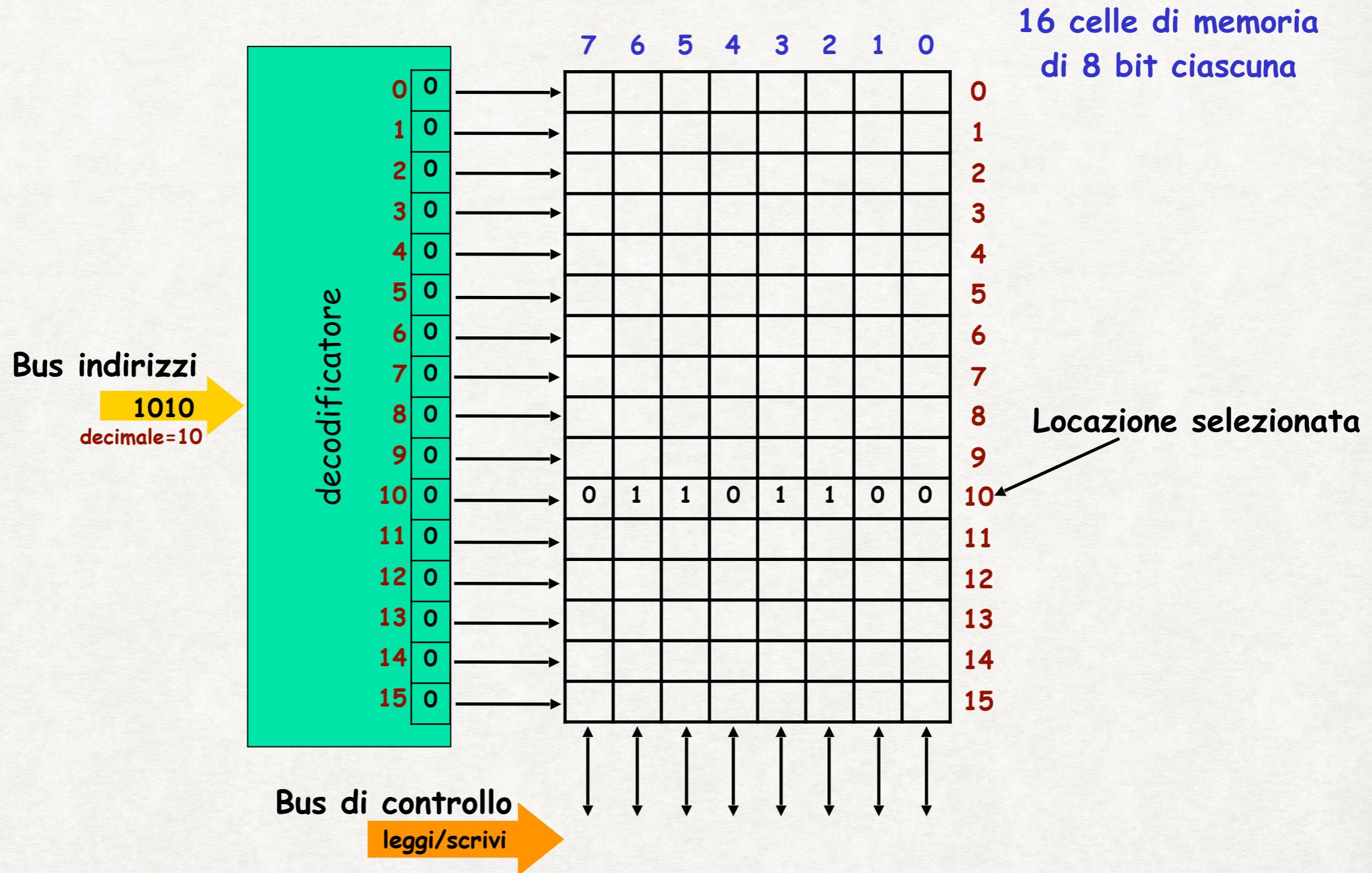
# SCHEMA DELLA MEMORIA



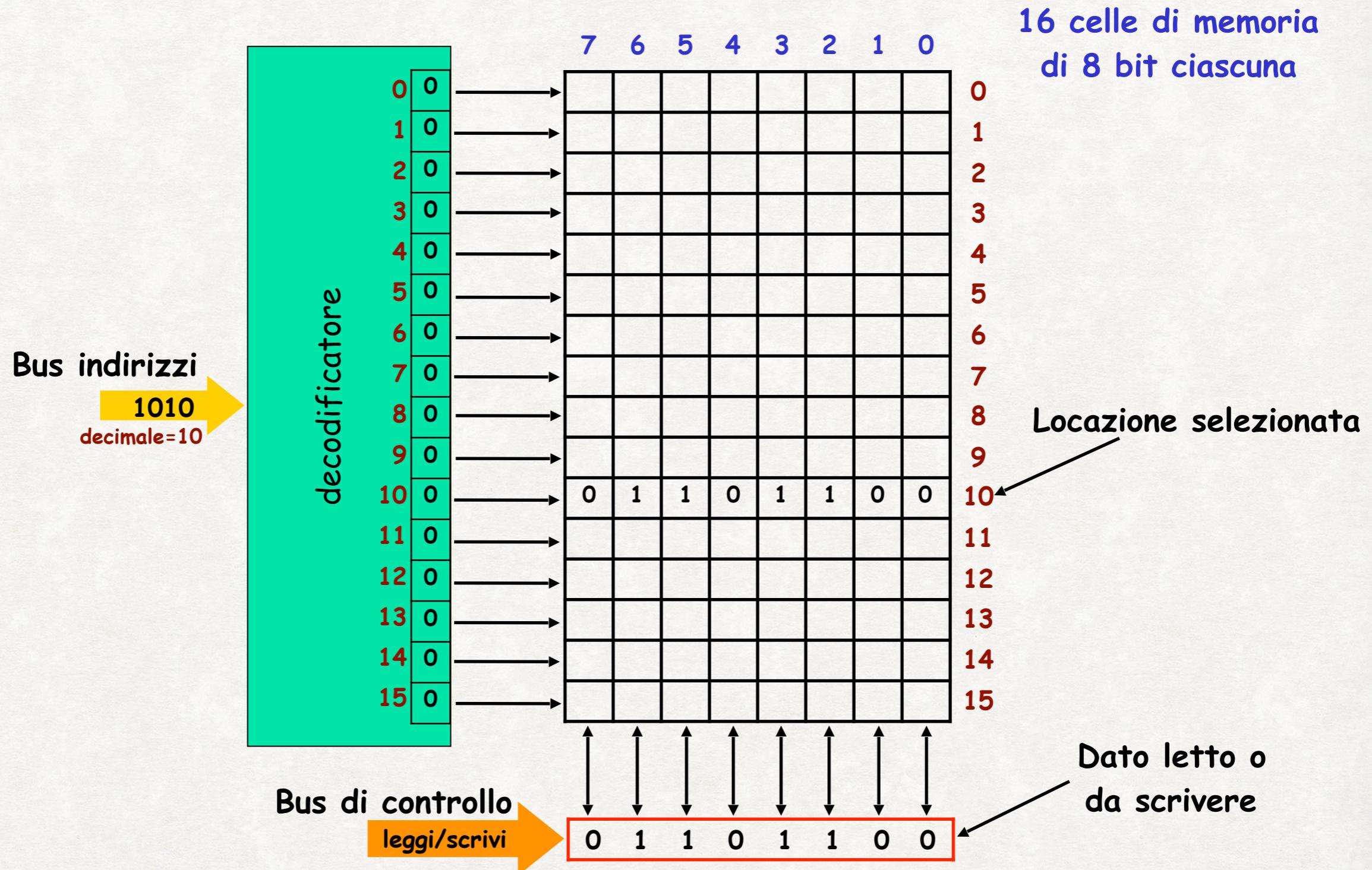
# SCHEMA DELLA MEMORIA



# SCHEMA DELLA MEMORIA



# SCHEMA DELLA MEMORIA



# SCHEMA DELLA MEMORIA

