

La memoria

Liceo G.B. Brocchi

Classi prime Scientifico - opzione scienze applicate

Bassano del Grappa, Novembre 2022

Prof. Giovanni Mazzocchin

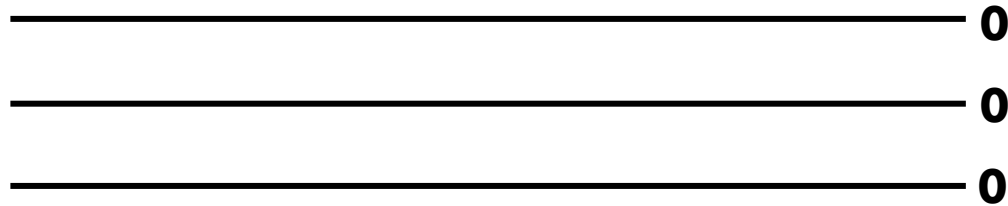
La memoria principale

- La memoria principale di un computer è un contenitore di celle ordinate
- Ogni cella è ampia **1 byte (8 bit)** e può contenere istruzioni o dati
- Ogni cella è localizzata da un indirizzo
- Se la memoria ha 512 celle:
 - la prima ha indirizzo **0x000 (0_{dec})**
 - l'ultima ha indirizzo **0x1FF (511_{dec})**
 - se le celle sono n e vengono indirizzate partendo da 0, l'ultima cella ha indirizzo n - 1
- L'insieme di tutte le celle di una memoria è detto **spazio di indirizzamento** della memoria

La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **n** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^n**

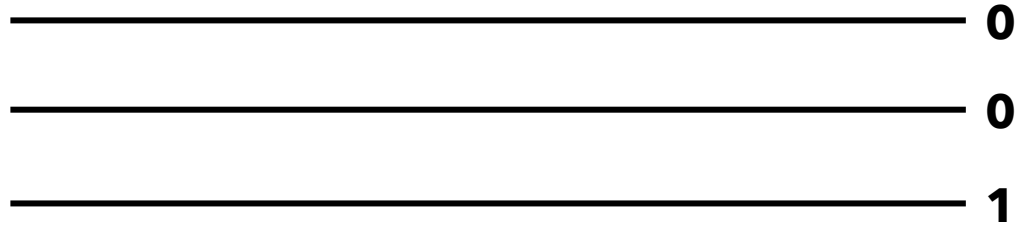
bus indirizzi
a 3 linee



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

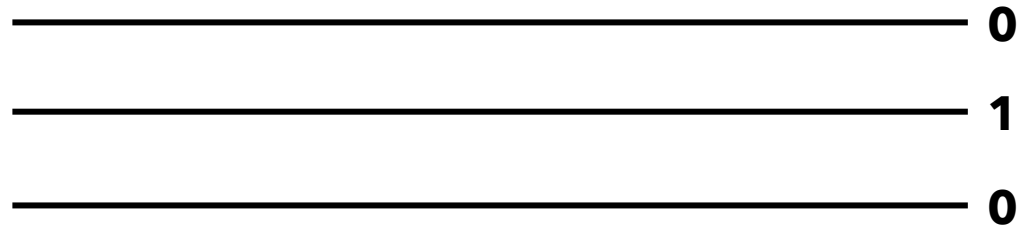
**bus indirizzi
a 3 linee**



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

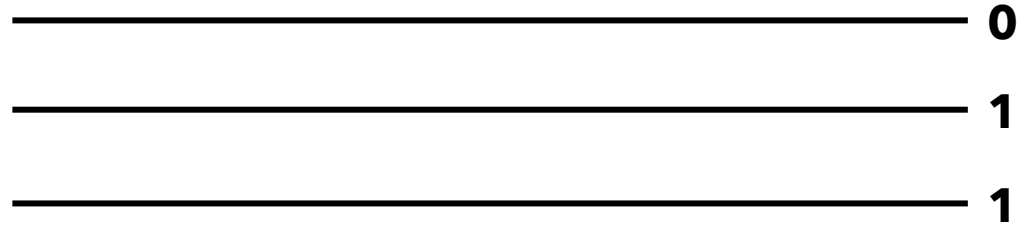
**bus indirizzi
a 3 linee**



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

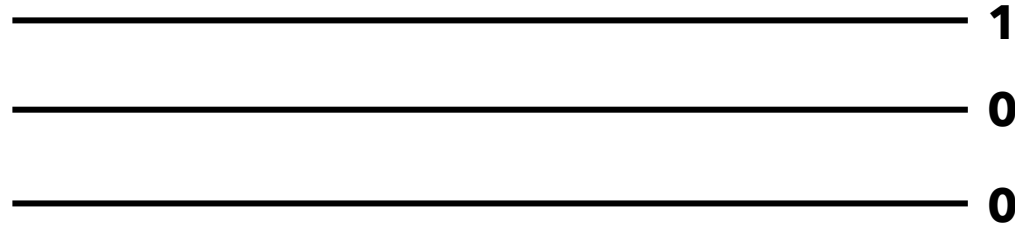
bus indirizzi
a 3 linee



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

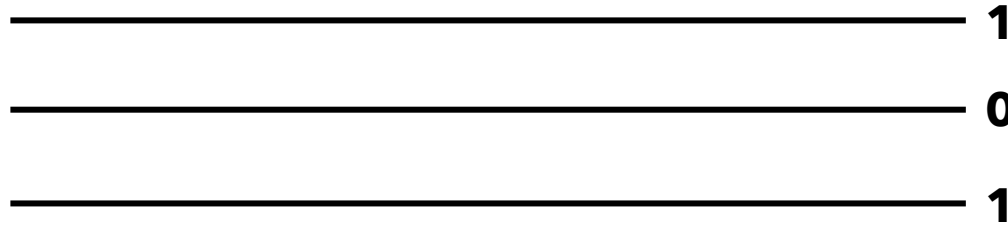
**bus indirizzi
a 3 linee**



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

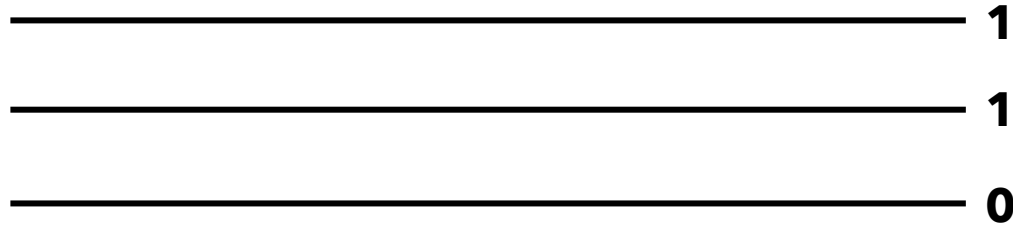
**bus indirizzi
a 3 linee**



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

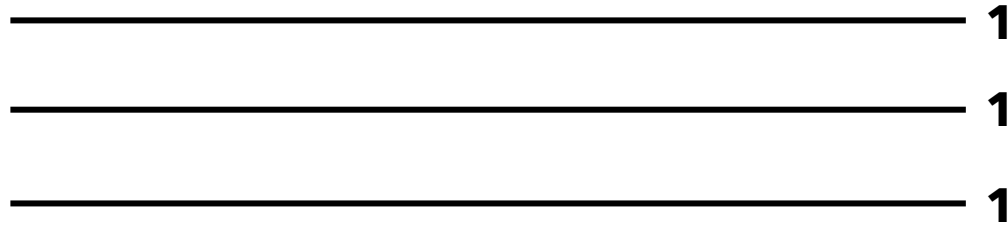
**bus indirizzi
a 3 linee**



La memoria principale

- L'ampiezza dello **spazio di indirizzamento** dipende dall'ampiezza del **bus indirizzi**
- Se il bus indirizzi è composto da **3** linee, ciascuna delle quali trasferisce **1 bit**, lo spazio di indirizzamento è **2^3** . E' solo un esempio, ovviamente con solo 3 linee la memoria indirizzabile sarebbe piccolissima

**bus indirizzi
a 3 linee**



La memoria principale

- La memoria principale è **volatile**, ossia le sue celle perdono i contenuti quando la macchina non è più alimentata da corrente elettrica (ovviamente vedremo anche altre memorie **non volatili**... del resto se siete in grado di salvare un file qualche supporto sicuramente li memorizza anche in assenza di alimentazione elettrica. Stiamo parlando di memorie di massa, o secondarie, basate su *dischi, nastri, CD, DVD, SSD*)
- La memoria principale è di tipo **RAM** (**Random Access Memory**), la cui traduzione letterale in italiano è *memoria ad accesso casuale*

ATTENZIONE MASSIMA:
cosa significa che un supporto di memoria è ad accesso casuale?

Sicuramente non significa che funziona a caso...
La comprensione di questo concetto è fondamentale per lo studio dell'Informatica

Accesso casuale (*random access*)

ACCESSO CASUALE o DIRETTO

il tempo che la CPU impiega ad accedere ad una locazione di memoria è indipendente dalla posizione della cella

memory address	memory content, 1 byte cells			
0x00	0xAA	0xB1	0x45	0x56
0x04	0x5A	0xB2	0x33	0x96
0x08	0x99	0x43	0x45	0x56
0x0C	0xAA	0x01	0x45	0xFE
0x10	0xFE	0xEF	0xEF	0xBE
0x14	0xAA	0xB6	0x02	0x04
.....			
0xFA	0xAA	0xB8	0xCA	0xFE
0xFE	0xCA	0xEE	0x45	0x56

Accesso casuale (*random access*)

memory address	memory content, 1 byte cells			
0x00	0xAA	0xB1	0x45	0x56
0x04	0x5A	0xB2	0x33	0x96
0x08	0x99	0x43	0x45	0x56
0x0C	0xAA	0x01	0x45	0xFE
0x10	0xFE	0xEF	0xEF	0xBE
0x14	0xAA	0xB6	0x02	0x04
.....			
0xF6	0xAA	0xB8	0xCA	0xFE
0xFA	0xCA	0xEE	0x45	0x56

I valori scritti nelle celle sono byte casuali. Non confondere il contenuto delle celle con il loro indirizzo

Per la CPU, il tempo di accesso alla cella di indirizzo 0x00 non differisce dal tempo di accesso alla cella 0xFD, o dal tempo di accesso a qualsiasi altra cella. Si dice anche che l'accesso avviene in tempo costante. Il tempo di accesso alle celle della RAM del vostro computer è probabilmente dell'ordine dei nanosecondi.

La cosa non è scontata: ci sono supporti di memoria nei quali per accedere ad una locazione specifica bisogna scorrere tutte le locazioni precedenti in ordine (**accesso sequenziale**).

Tecnologia di realizzazione della RAM

- La memoria principale è realizzata in gran parte con tecnologia **DRAM** (**D**ynamic **R**andom **A**ccess **M**emory)
 - a livello fisico-elettronico, una DRAM è un *chip* di semiconduttore in cui i dati, ossia i bit, sono memorizzati su *condensatori*, quindi sotto forma di carica elettrica
 - allo stato logico **1** è associata la presenza di carica, allo stato logico **0** l'assenza di carica
 - **NB**: questi condensatori si scaricano velocemente per non perdere il contenuto, questi microcondensatori devono essere *refreshati* frequentemente, ossia letti e ricaricati
 - l'operazione di **refresh** viene eseguita anche migliaia di volte al secondo.
Chiaramente il refresh rallenta l'operazione di accesso ad una cella di memoria

Tecnologia qui significa «componenti elettroniche». Un **condensatore** può essere visto come un piccolo secchio che immagazzina elettroni. Se vi interessano questi dettagli tecnologici potremmo organizzare un minicorso pomeridiano di elettronica digitale

Tecnologia di realizzazione della RAM

- Un'altra parte, più piccola, della memoria principale è realizzata con tecnologia **SRAM** (**S**tatic **R**andom **A**ccess **M**emory)
- La singola cella di memoria non è realizzata come condensatore, ma come **circuito flip-flop**
- Questi circuiti flip-flop non necessitano del refresh per memorizzare i bit

Tecnologia di realizzazione della RAM

- Un'altra parte, più piccola, della memoria principale è realizzata con tecnologia **SRAM** (**S**tatic **R**andom **A**ccess **M**emory)
- La singola cella di memoria non è realizzata come condensatore, ma come **circuito flip-flop**
- Queste celle di memoria realizzate con circuiti flip-flop non necessitano del refresh per memorizzare i bit
- Una memoria **SRAM** è **più veloce, più costosa e meno capiente** di una memoria **DRAM**

Capacità di memoria e unità di misura

Capacità di memoria:
la quantità di *byte (binary octet)* che un supporto di memoria è in grado di memorizzare

Avete sicuramente sentito parlare di *4-8 giga di RAM*, di *SSD da 1 tera* etc...

A scuola abbiamo imparato che esistono anche memorie molto piccole all'interno della CPU, con capacità di pochi byte

Premessa: sicuramente la quantità di memoria fisica è sempre finita. **L'infinito in informatica non esiste.**

Cosa significano esattamente queste cose?

Capacità di memoria e unità di misura

In informatica e in elettronica digitale, tutte le informazioni sono memorizzate come sequenze di bit (da *binary digit*)

Esempi di informazione: testo, numeri, suoni, immagini etc...

```
cyofanni@LAPTOP-I0S1RKRC:~/Desktop$ xxd -b bassano-del-grappa.jpg | head -5
00000000: 11111111 11011000 11111111 11100000 00000000 00010000  ....
00000006: 01001010 01000110 01001001 01000110 00000000 00000001  JFIF..
0000000c: 00000001 00000001 00000000 01001000 00000000 01001000  ...H.H
00000012: 00000000 00000000 11111111 11011011 00000000 01000011  ....C
00000018: 00000000 00000011 00000010 00000010 00000011 00000010  ....
cyofanni@LAPTOP-I0S1RKRC:~/Desktop$ hexdump bassano-del-grappa.jpg | head -5
00000000 d8ff e0ff 1000 464a 4649 0100 0101 4800
00000010 4800 0000 dbff 4300 0300 0202 0203 0302
00000020 0303 0403 0303 0504 0508 0405 0504 070a
00000030 0607 0c08 0c0a 0b0c 0b0a 0d0b 120e 0d10
00000040 110e 0b0e 100b 1016 1311 1514 1515 0f0c
cyofanni@LAPTOP-I0S1RKRC:~/Desktop$
```

Capacità di memoria e unità di misura

name	symbol	equal to	real-world analogy	power of 10 (approximation)
bit	b	1 bit		
byte	B	8 bit	an alphanumeric character	
kilobyte	KB	$2^{10} = 1024$ bytes	a short text written on a text editor	1.000 B
megabyte	MB	2^{20} B = 1024 KB = 1.048.576 bytes	a 500-page book	1.000.000 B

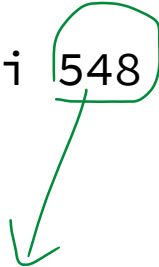
Capacità di memoria e unità di misura

name	symbol	equal to	real-world analogy	power of 10 (approximation)
gigabyte	GB	2^{30} B = 1024 MB	1000 500-page books	1.000.000.000 B
terabyte	TB	2^{40} B = 1024 GB	many big libraries	10^{12} B
petabyte	KB	2^{50} = 1024 TB	hundreds of big libraries	10^{15} B
exabyte	MB	2^{60} B = 1024 TB	billions of libraries	10^{18} B

Alcuni comandi della shell Bash

- Ecco alcuni comandi che potete lanciare dalla shell per ottenere le dimensioni di un file:
 - **ls -l example**

```
-rw-r--r-- 1 cyofanni cyofanni 548 Dec 11 17:02 example
```

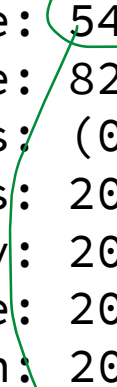


l'unità di misura è il byte
Significa che il file di nome example pesa **548 byte**

Alcuni comandi della shell Bash

- **stat example**

```
File: example
  Size: 548          Blocks: 8          IO Block: 4096   regular file
Device: 820h/2080d  Inode: 83310       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/cyofanni)   Gid: ( 1000/cyofanni)
Access: 2022-12-11 17:02:07.825242553 +0100
Modify: 2022-12-11 17:02:15.735242796 +0100
Change: 2022-12-11 17:02:15.735242796 +0100
Birth:  2022-12-11 15:41:23.825229709 +0100
```

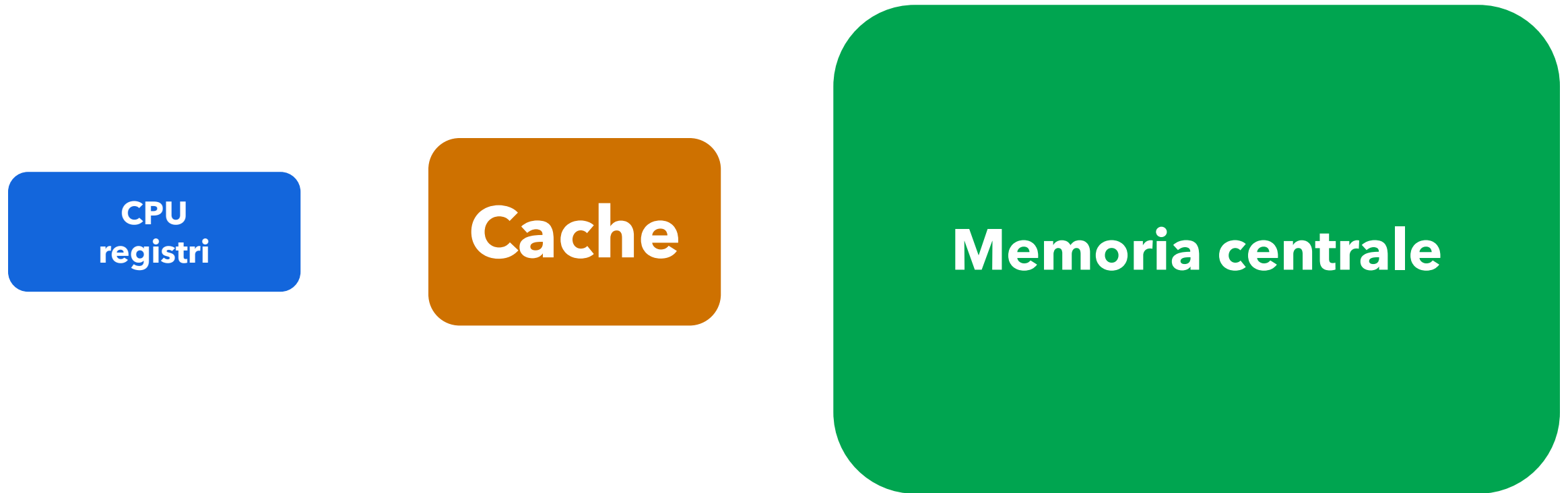


l'unità di misura è il byte

Significa che il file di nome example pesa **548 byte**

La memoria cache

- La **memoria cache** è una memoria (volatile) più veloce della RAM, ma con minore capacità
- È posta **tra i registri della CPU** e la **memoria centrale**



La memoria cache

- Quando la CPU ha bisogno di un'informazione (istruzione/dato)
 - prima la cerca nella cache. Se è presente, la utilizza (**best case, cache hit**)
 - Se non è presente, la legge dalla RAM (**worst case, cache miss**)
 - Le informazioni prelevate dalla RAM vengono memorizzate anche nella cache per i prossimi accessi

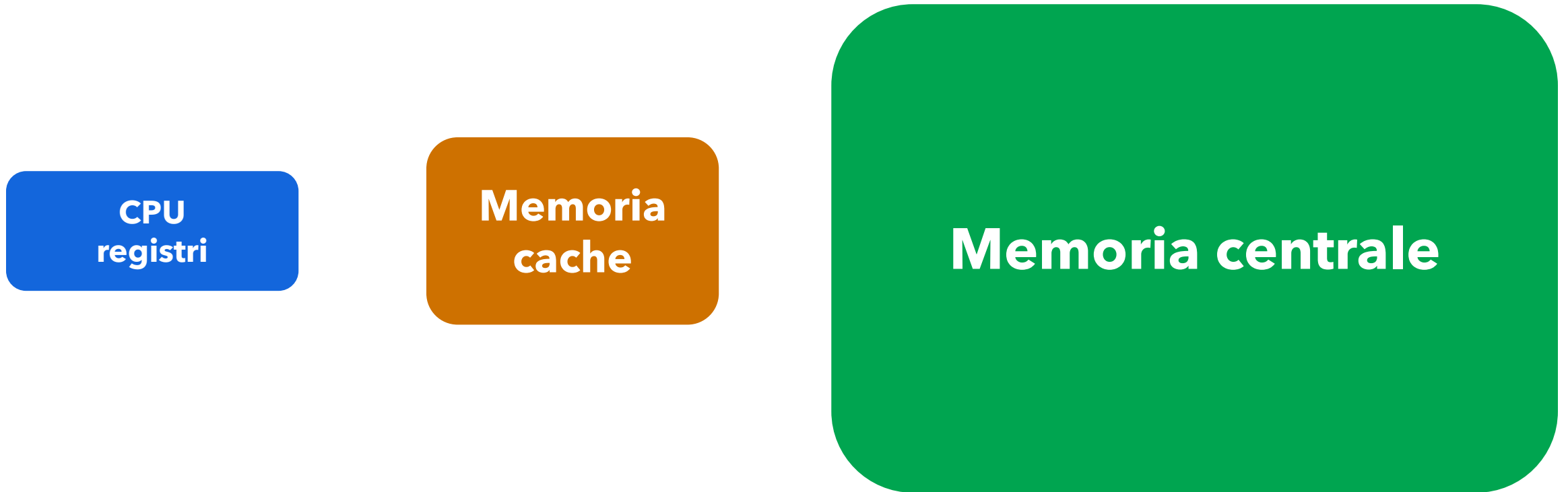
CPU
registri

Memoria
cache

Memoria centrale

La memoria cache

- La cache è realizzata normalmente con tecnologia **SRAM**
- E' molto meno capiente, più veloce e più costosa della memoria principale, realizzata con tecnologia **DRAM**



La memoria cache

- La cache, essendo molto più piccola, non può contenere in ogni momento una copia dell'intera memoria principale
- Contiene solo le informazioni **statisticamente più richieste**, eliminandole quando probabilmente non servono più

CPU
registri

Memoria
cache

Memoria centrale

La memoria cache

- Si possono avere vari livelli di cache:
 - cache di primo livello, **L1**: integrata direttamente nel processore
 - cache di secondo livello, **L2**, esterna al processore ma collegata direttamente ad esso
 - cache di terzo livello, **L3**, sulla motherboard (scheda madre), accessibile tramite un bus specializzato
- Su un sistema Linux, potete ottenere informazioni sulle cache lanciando il comando:

lscpu

Informazioni fornite da *lscpu*

Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 39 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Vendor ID: GenuineIntel
Model name: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
CPU family: 6
Model: 142
Thread(s) per core: 2
Core(s) per socket: 4
Socket(s): 1
Stepping: 10

Caches (sum of all):

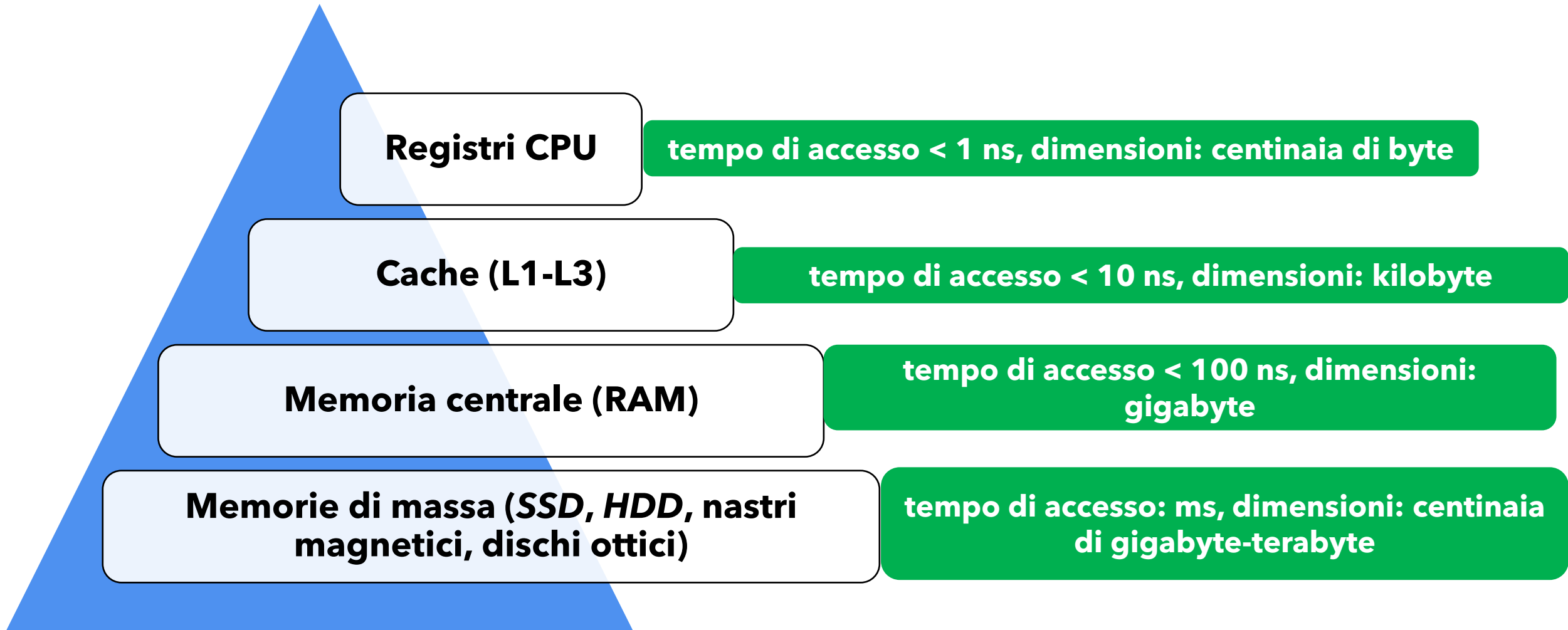
L1d: 128 KiB (4 instances)
L1i: 128 KiB (4 instances)
L2: 1 MiB (4 instances)
L3: 8 MiB (1 instance)



La gerarchia di memoria

- Ogni memoria interagisce con 1 o 2 altri tipi di memoria
- Si forma una gerarchia, in cui le informazioni vengono copiate tra livelli adiacenti
- **Livelli superiori:** più vicini al processore, ad accesso più veloce, con capienza più piccola e più costose
- **Livelli inferiori:** più lontani dal processore, ad accesso più lento, con capienza più grande e più economiche

La gerarchia di memoria



La gerarchia di memoria – perché?

- Quando la CPU richiede un'informazione:
 1. la cerca nella cache, se c'è (cache hit) ok, altrimenti la cerca nella RAM
 2. se l'informazione c'è nella RAM, l'informazione viene copiata nella cache
 3. se l'informazione non c'è nella RAM, viene cercata nel disco (memoria di massa). Una volta trovata nel disco, viene copiata in RAM.
- Se le informazioni non venissero portate ai livelli superiori e rimanessero tutte nel disco, l'esecuzione dei programmi sarebbe di una lentezza inaccettabile. Per questo i livelli superiori devono memorizzare quante più informazioni possibili
- **Ma come vengono scelte le informazioni da copiare nei livelli superiori?**

Un'analogia domestica

- Immaginate di avere un cestino di bottiglie d'acqua in cucina (*registri, cache*)
- Quando avete bisogno di una bottiglia, cercate nel cestino. *L'accesso è immediato*
- Se il cestino è vuoto, andate in cantina (*RAM*) a fare rifornimento
 - riempite tutto il cestino, per evitare di tornarci troppo presto!
 - per arrivare alla cantina bisogna fare le scale... l'accesso non è immediato
- Se non ci sono le bottiglie cercate neanche in cantina allora bisogna uscire di casa a comprarle (*accesso al disco, molto lento e faticoso*)

Principi di località

- Quando si accede ad un livello inferiore della gerarchia, non viene trasferita soltanto l'informazione necessaria in quel momento, ma un **insieme di informazioni**, che ha nomi diversi:
 - dalla RAM alla cache vengono trasferiti **blocchi**
 - dal disco alla RAM vengono trasferite **pagine**



Principi di località

- **Principio di località temporale:** se un dato viene referenziato («richiesto») in un certo istante, probabilmente questo dato verrà referenziato di nuovo entro breve
- **Principio di località spaziale:** se un dato viene referenziato in un certo istante, probabilmente i dati memorizzati nelle celle adiacenti verranno richiesti a breve

Endianness

```
void show_endianness() {  
    unsigned int bytes = BYTES;  
    unsigned char *bytes_ptr = (unsigned char*)&bytes;  
    printf("bytes sequence is 0x%04X\n", bytes);  
    putchar('\n');  
    puts("bytes in memory, ordered by increasing addresses:");  
    for (int i = 0; i < BYTES_SEQ_LENGTH; i++) {  
        printf("\tbyte 0x%02X at address %p\n", *(bytes_ptr + i) & 0xFF, bytes_ptr +  
i);  
    }  
}
```