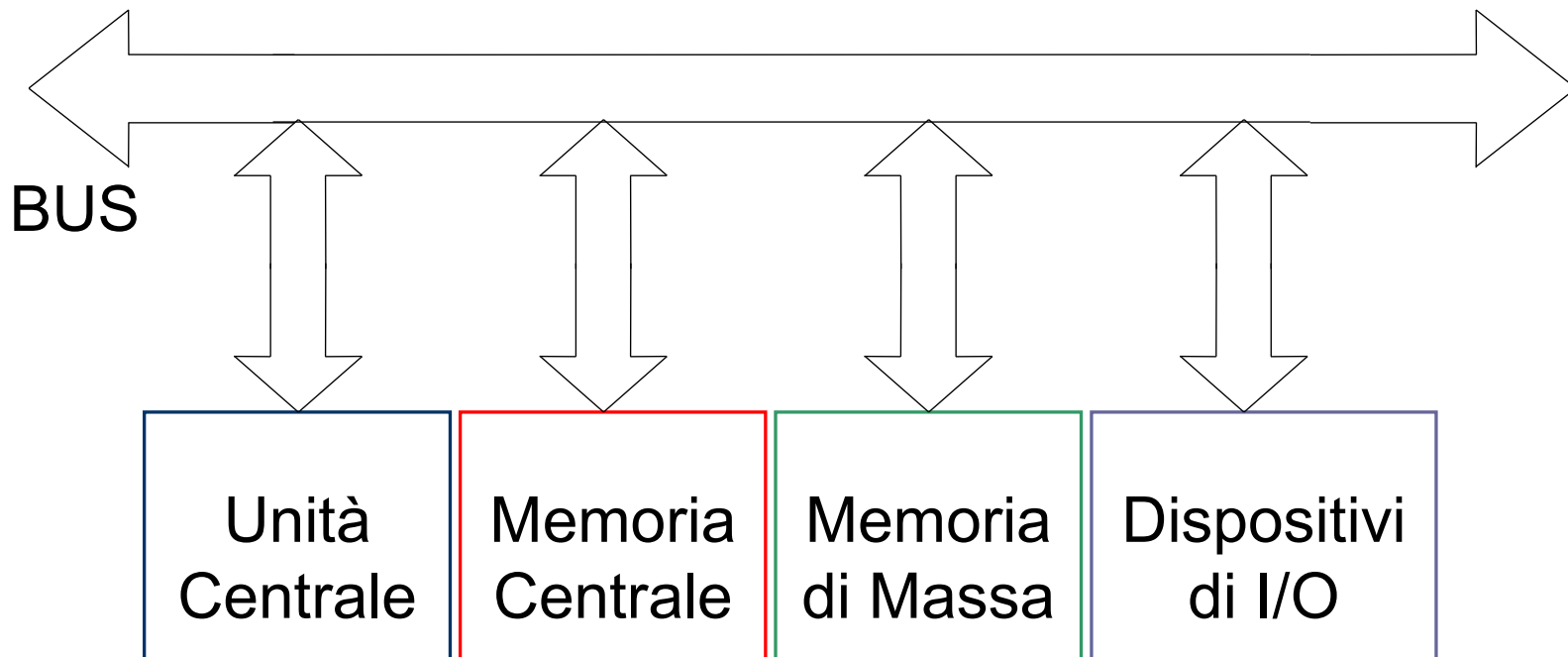


# Architettura di un elaboratore

# Modello architetturale di un computer

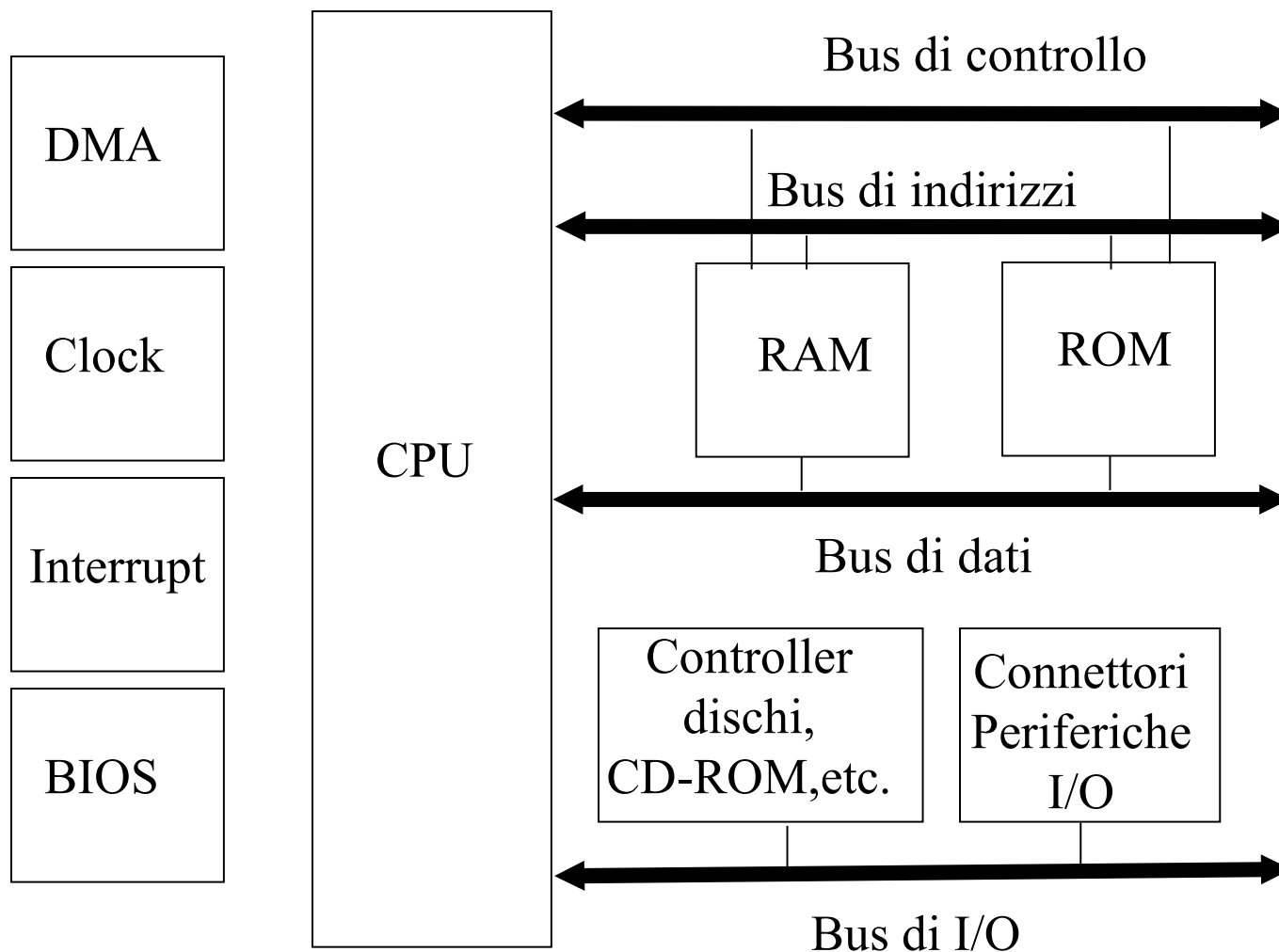
Prof. G. Ascia

Ogni computer è costituito da un insieme di blocchi funzionali tra loro interconnessi da sistemi di comunicazioni, per assolvere al ruolo di “esecutore” di programmi



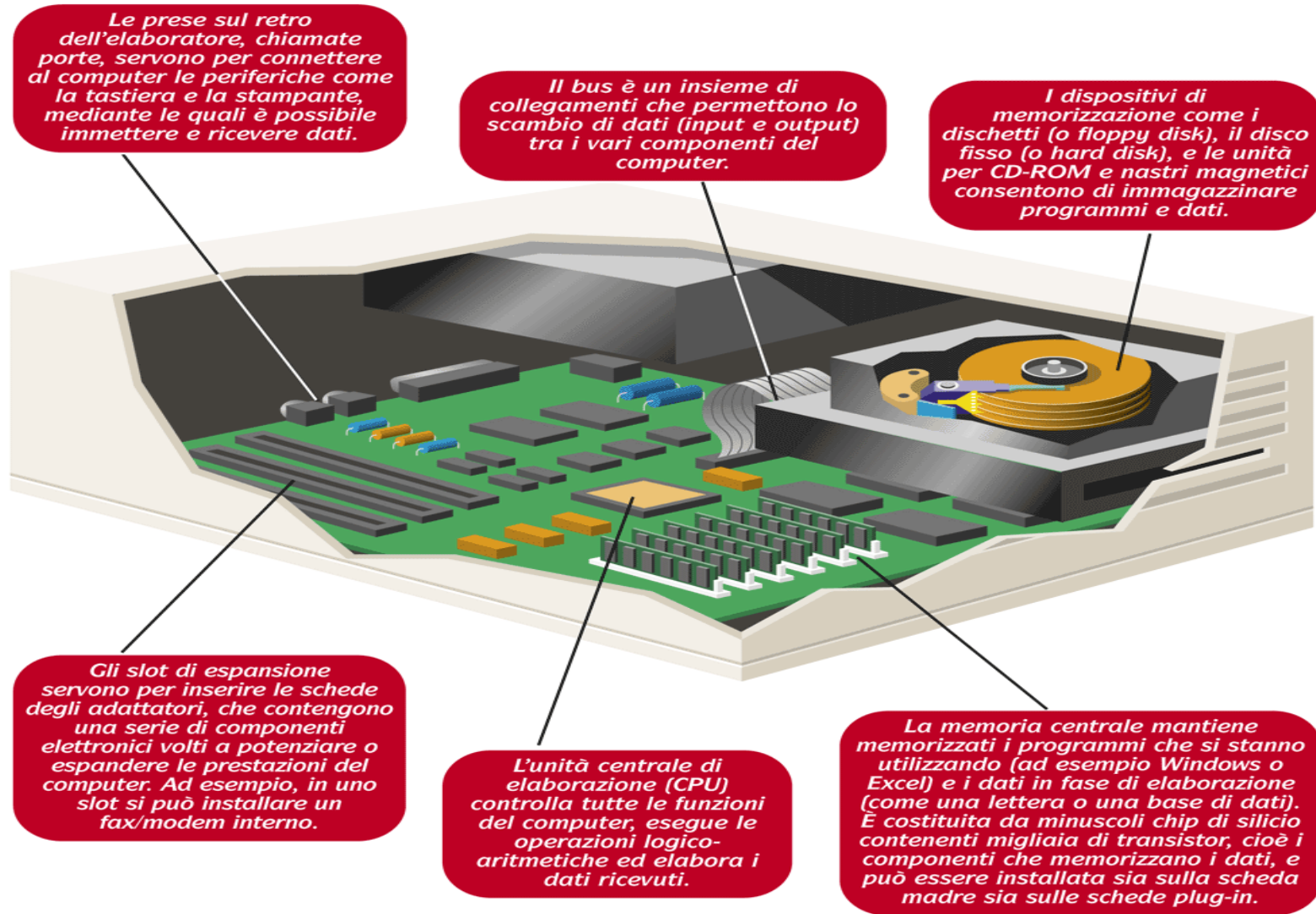
# Architettura di un computer

Prof. G. Ascia



# Dentro il Computer

Prof. G. Ascia



# Memorie di un elaboratore

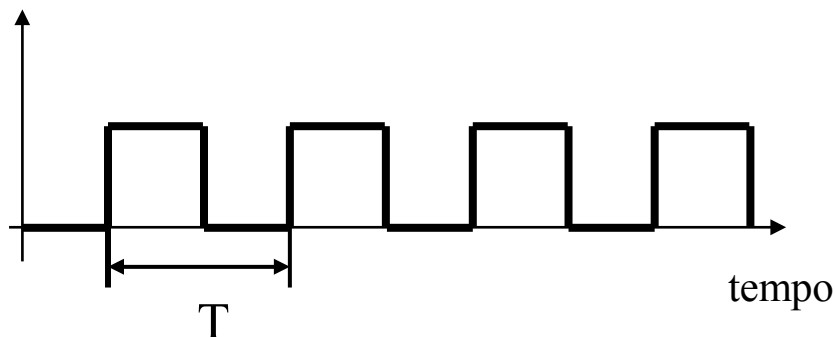
---

Prof. G. Ascia

- **MEMORIA PRINCIPALE**
  - Memorizzano i dati e i programmi su cui lavora direttamente la CPU
  - Tempi di accesso < 50-60 ns.
  - Si accede a byte o a word
  - Dimensioni tipiche 1-4 Gbytes
  - Realizzata con dispositivi a semiconduttore (Transistor in tecnologia CMOS)
- **MEMORIA SECONDARIA**
  - Memorizzano dati e programmi sotto forma di file
  - una quantità assai elevata di dati (fino a diverse centinaia di Gbytes).
  - Tempi di accesso elevati (msec)
  - Si accede a blocchi
  - Poco costosa
  - Realizzata con dispositivi magnetici (HD), ottici (CD,DVD) o elettronici (Flash memory)

# Clock

Prof. G. Ascia



Frequenza=1/T

T=10 ns f=100 MHz

T=1 ns f=1 GHz

T=0,5 ns f=2 GHz

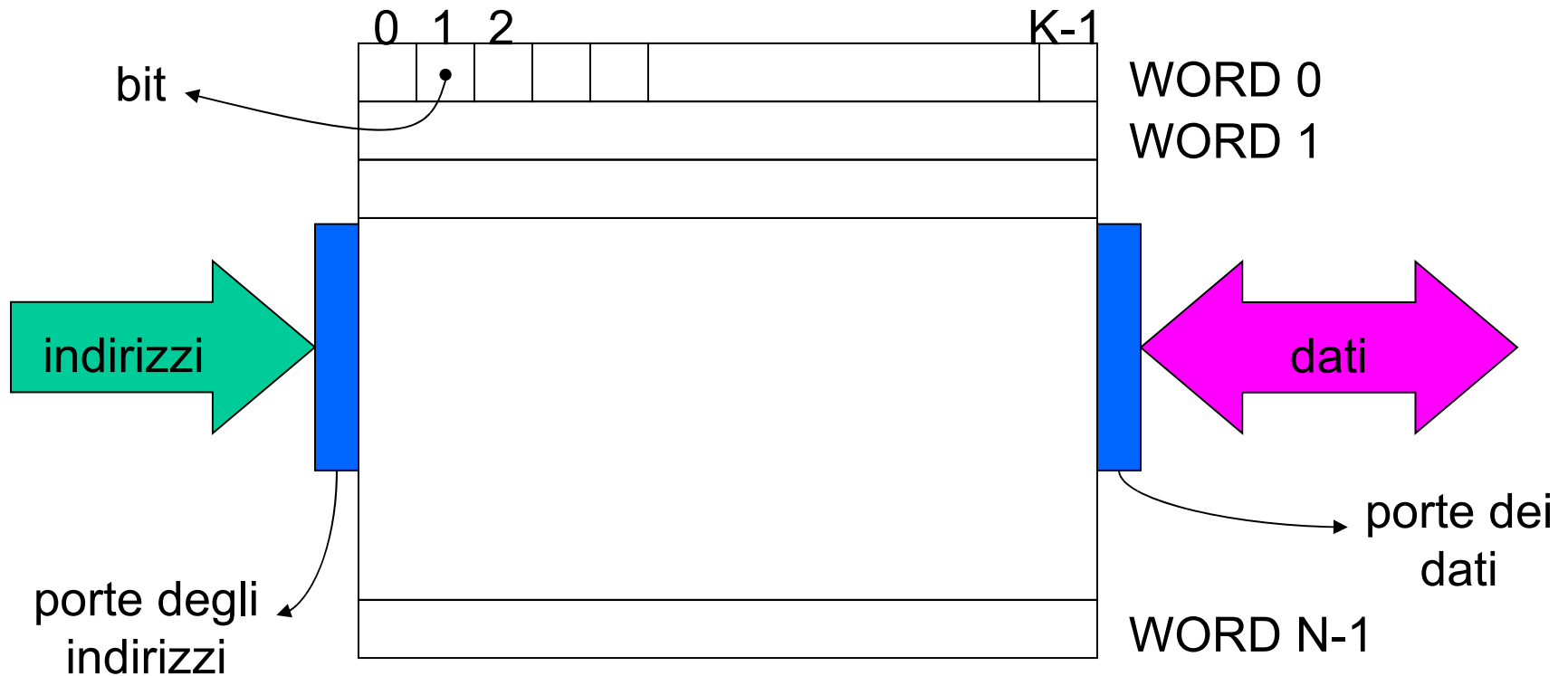
- Le operazioni all'interno dell'elaboratore vengono sincronizzate mediante un segnale detto clock
- In generale il clock é prodotto da un apposito circuito integrato, la cui oscillazione é controllata da un cristallo di quarzo.

# Memoria Centrale

Prof. G. Ascia

Consente di immagazzinare informazioni in formato digitale

## ORGANIZZAZIONE



# Memoria Centrale

---

Prof. G. Ascia

E' costituita da un insieme di locazioni, ciascuna in grado di memorizzare una parola (word) di **K** bit {K = 1, 8, 16, 32, 64}

La **capacità di memoria** esprime la quantità di informazioni massima in essa memorizzabile.

Ad es. 1 Mbyte =  $2^{20}$  byte, ovvero  $2^{20}$  word di 1 byte



# Memoria Centrale

---

Prof. G. Ascia

## COSA CONSENTE:

LA SCRITTURA: consente di memorizzare una parola in una determinata posizione (indirizzo)

LA LETTURA: consente di leggere una parola memorizzata in una determinata posizione

La scrittura di una parola sovrascrive il contenuto precedente.

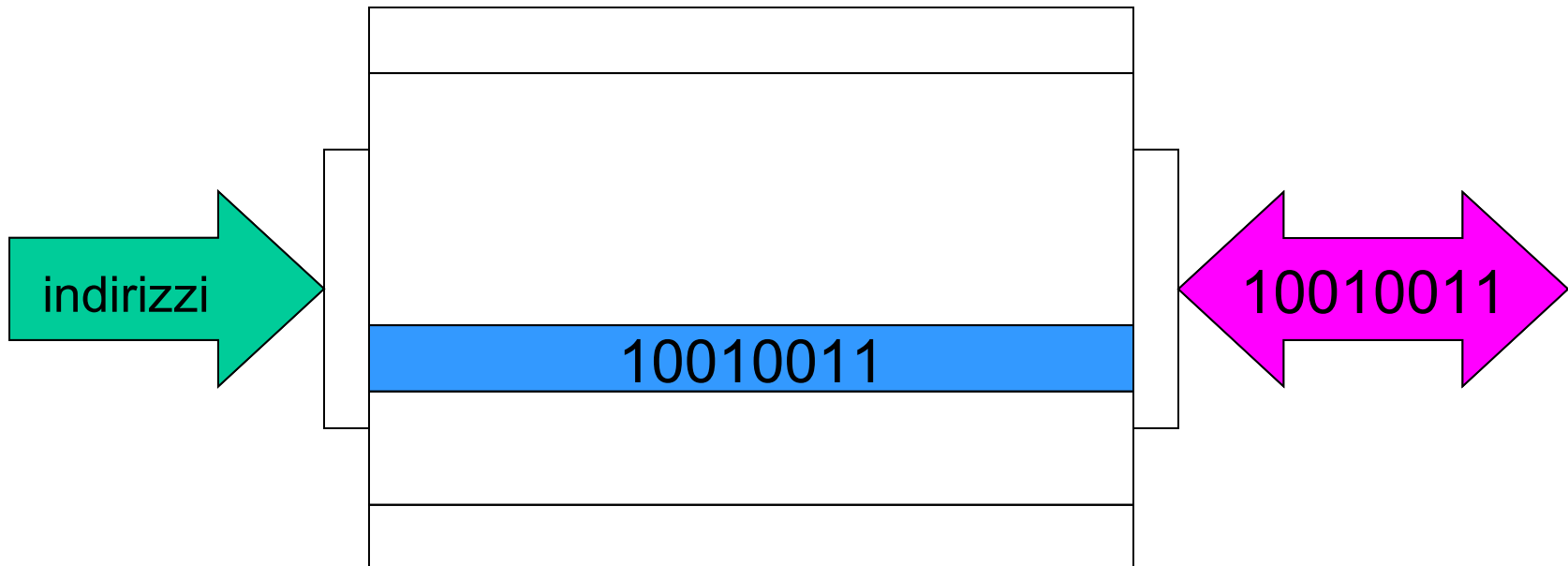
La lettura di una parola non altera il contenuto memorizzato.

# Come si usa

Prof. G. Ascia

## IN SCRITTURA:

- sulla porta degli indirizzi si deve fornire l'indirizzo (la posizione) della locazione su cui memorizzare la parola
- sulla porta dei dati si deve fornire la parola (configurazione di bit) da scrivere

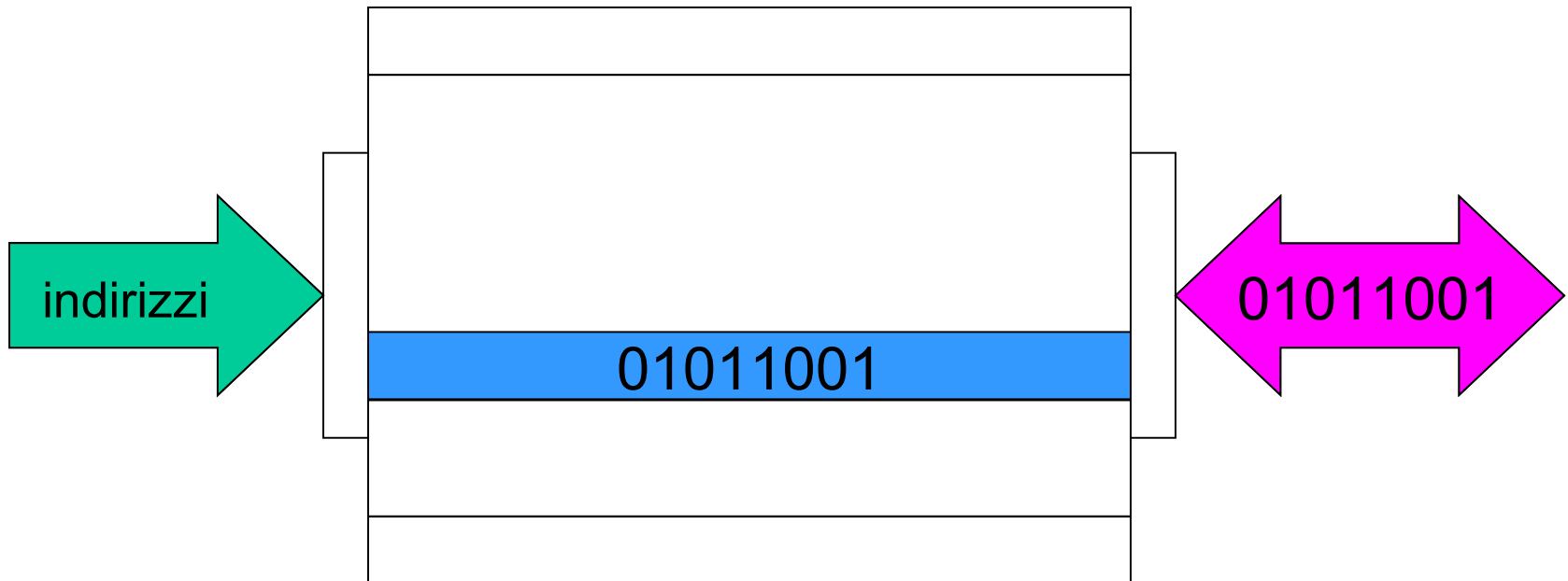


# Come si usa

Prof. G. Ascia

## IN LETTURA:

- sulla porta degli indirizzi si deve fornire l'indirizzo (la posizione) della locazione da leggere
- sulla porta dei dati si ottiene la parola (configurazione di bit) memorizzata in quell'indirizzo



# Memoria Centrale

---

Prof. G. Ascia

Un indirizzo è costituito da un insieme di bit che codifica la posizione di una parola

Ad es.: Se la memoria è di 1 Kbyte ( $2^{10}$  byte) occorrono 10 bit ( $\lg_2 2^{10} = 10$ ) per specificare un indirizzo.

# Memoria principale: parametri caratteristici

---

Prof. G. Ascia

- **Capacità della memoria:** Indica il numero  $N$  di word da  $W$  bit presenti nella memoria
- **Tempo di accesso:** Indica il tempo che intercorre tra l'istante in cui è richiesta l'informazione e l'istante in cui è disponibile, espresso in ns
- **Tempo di ciclo della memoria:** il minimo intervallo tra due successivi accessi per lettura/scrittura in memoria.
- **Modificabilità:** le memorie possono essere riscrivibili o a sola lettura
- **Persistenza:** volatili (RAM) o permanenti
- **Tipo di accesso:** diretto o sequenziale

# Memorie

---

Prof. G. Ascia

- Possiamo identificare tre tipi di memorie
  - RAM (Random Access Memory)
  - ROM (Read Only Memory)
  - Cache Memory

# RAM - Random Access Memory

---

Prof. G. Ascia

## ✓ Memoria Volatile

- La mancanza di tensione provoca la perdita di tutte le informazioni contenute.
- In presenza di tensione, ciascuna informazione memorizzata rimane fino alla successiva scrittura.

## ✓ Memoria ad accesso casuale

- ✓ Attualmente assumono valori tipici di centinaia/migliaia di MBytes.

# RAM - Random Access Memory

---

Prof. G. Ascia

- **RAM STATICA (SRAM)**
  - Conservano le informazioni fino a quando è mantenuta la tensione di alimentazione.
  - Tempo di accesso: <10 ns
  - Molto costosa
- **RAM DINAMICA(DRAM)**
  - Conservano le informazioni solo per un breve periodo anche in presenza di alimentazione.
  - E' necessaria un dispositivo (di refresh) che ripristina il contenuto della memoria prima che venga perduto.
  - Tempo di accesso: <70 ns
  - Poco costosa



# ROM - Read Only Memory

---

Prof. G. Ascia

- ✓ Memoria Permanente
  - Nella ROM l'informazione in essa contenuta rimane anche quando manca la corrente.
  - Nella ROM l'informazione rimane fino alla successiva scrittura.
- ✓ Memoria ad accesso casuale
- ✓ Le ROM vengono in genere utilizzate per memorizzare programmi e dati di configurazione essenziali per il funzionamento del computer che devono essere memorizzati anche quando il computer è spento.
  - Bios

# ROM - Read Only Memory

---

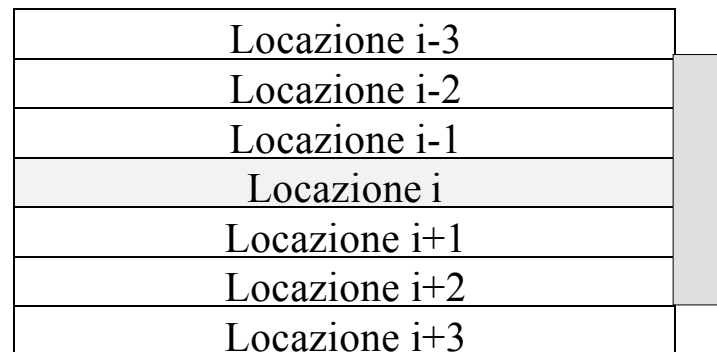
Prof. G. Ascia

- ROM (Read Only Memory)
  - Memoria a sola lettura. Scrittura ad opera del costruttore.
- PROM( Programmable ROM)
  - Possono essere scritte una sola volta con un apposito dispositivo detto Programmatore di PROM
- EPROM(Erasable Programmable ROM)
  - Sono cancellabili mediante raggi ultravioletti e programmabili più volte.
- EEPROM (Electrically EPROM)
  - Sono cancellabili elettricamente
- Flash Memory:
  - come le EPROM, ma la scrittura avviene per blocchi e non per byte, molto velocemente.

# Memoria cache

Prof. G. Ascia

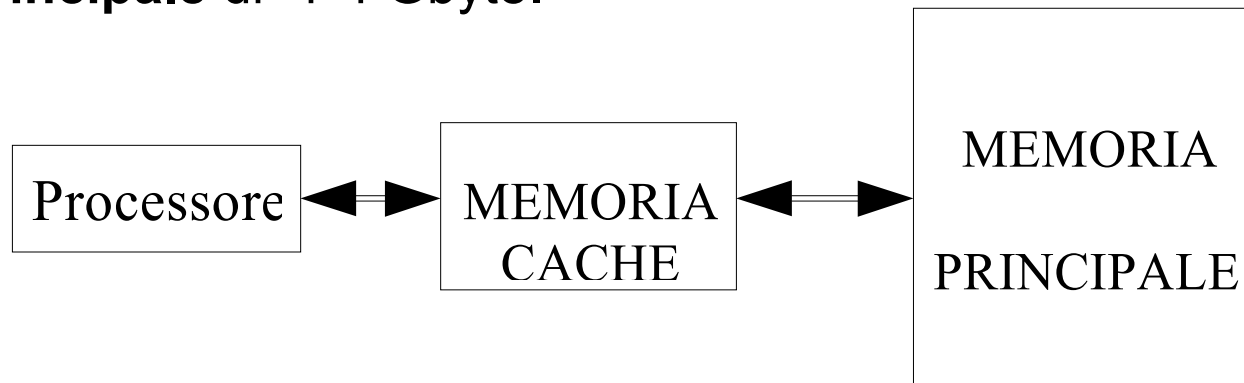
- La **Memoria CACHE** contiene quella parte di programma e dati di più frequente uso.
- Essa è realizzata utilizzando RAM statica e di conseguenza l'accesso è molto più veloce rispetto alla **memoria principale**.
- Il suo utilizzo si spiega con il seguente principio di località:
  - Durante l'esecuzione dei programmi c'è la tendenza entro breve tempo ad accedere nuovamente alla stessa locazione (località temporale)
  - ad accedere a locazioni vicine (località spaziale)



# Memoria Cache

Prof. G. Ascia

- L'uso della memoria cache dà quasi l'illusione al processore che tutta la memoria sia veloce come la cache.
- Durante l'esecuzione di un programma solo una parte del programma e dei dati è presente nella CACHE.
- Se l'unità centrale deve accedere ad una parte di programma che non è presente nella Cache, essa viene copiata dalla **Memoria Principale** alla **Memoria Cache**.
- La **Cache** ha una dimensione di 128-1024 Kbyte, la **Memoria Principale** di 1-4 Gbyte.



# Memoria cache

Prof. G. Ascia

- Nel caso in cui un'istruzione si trova nella cache si dice che c'è stato un **HIT**.
- Nel caso in cui un'istruzione non è presente nella cache si dice che c'è stato un **MISS**.
- Il tempo richiesto per copiare una parte del programma contenente l'istruzione da eseguire è detto **MISS PENALTY**.
- Il tempo di accesso medio in memoria è dato da:

$$t_{ACC} = t_{HIT} + f_{MISS} * t_{MISS PENALTY}$$

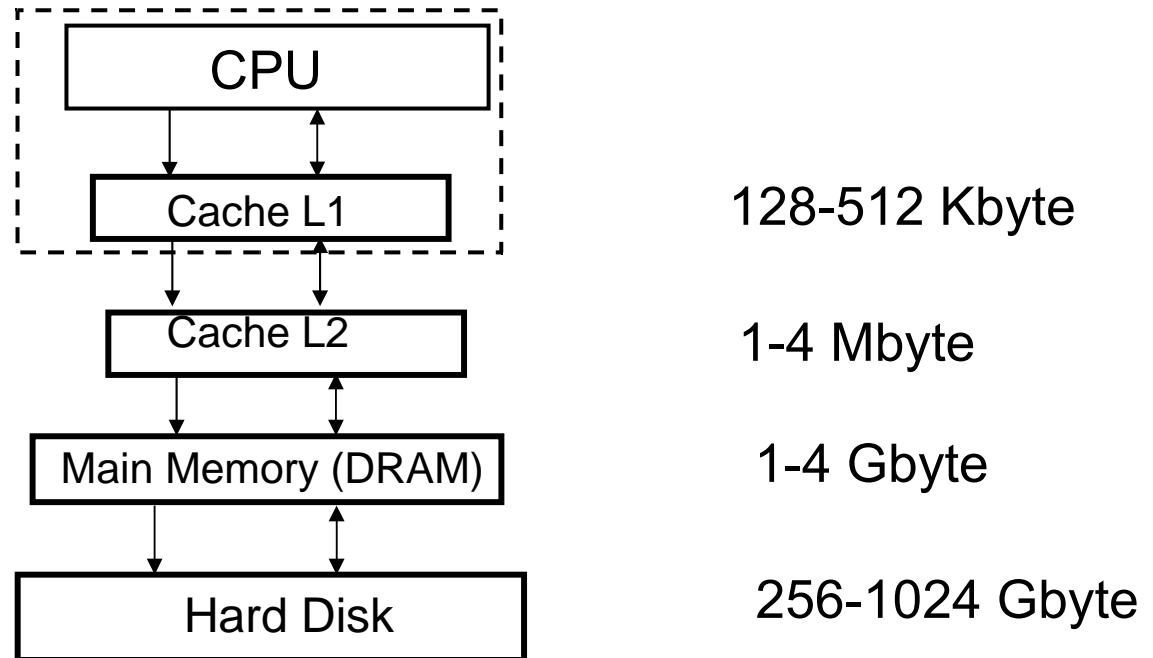
- Es. Cache di 128 Kbyte

$$t_{HIT} = 1 \text{ ciclo di clock} = 10 \text{ ns}, \quad f_{MISS} = 1 \% \quad t_{MISS PENALTY} = 50 \text{ cicli di clock}$$

- $t_{ACC} = 1 + 0.01 * 50 = 1,5 \text{ cicli di clock} = 24 \text{ ns}$ .
- Se la cache non è presente il tempo di accesso è quello della memoria principale.
- Es.  $t_{ACC} = t_{MEM} = 60 \text{ ns}$

# Gerarchia di memoria

Prof. G. Ascia



- I programmi e i dati vengono conservati nella MEMORIA DI MASSA.
- Al momento dell'esecuzione dei programmi, essi vengono copiati nella MEMORIA CENTRALE.
- Nella CACHE vengono copiate le parti di programmi di più frequente uso.

# Unità Centrale di Processo (CPU)

---

Prof. G. Ascia

**Sovrintende** all'esecuzione di un programma immagazzinato nella memoria centrale e **coordina** il funzionamento dei vari blocchi costituenti il computer.

Una data CPU è caratterizzata da un proprio set di istruzioni, ovvero dall'insieme di istruzioni che essa è in grado di eseguire

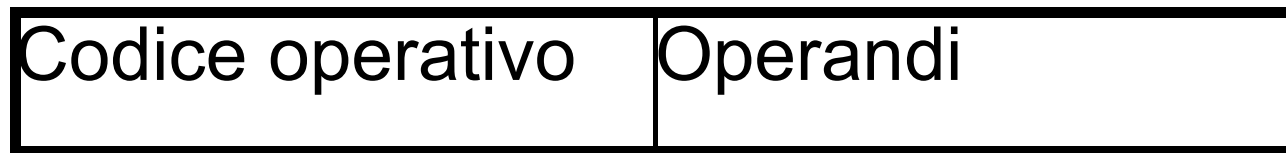
Ciascuna istruzione è codificata in binario, in modo che sia riconoscibile dalla CPU.

# Unità Centrale di Processo (CPU)

---

Prof. G. Ascia

## Formato di una istruzione



- Il codice operativo indica il tipo di istruzione.
- Il campo operandi contiene o l'indirizzo di un operando o l'operando stesso.

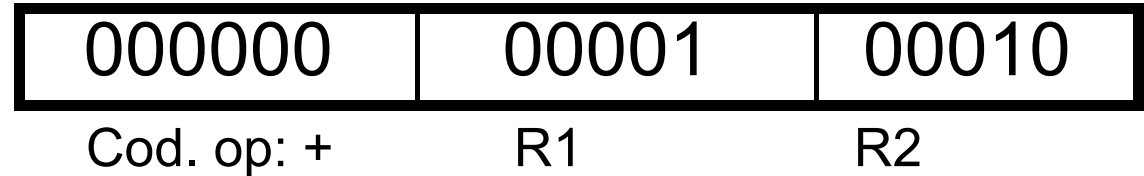


# Istruzioni

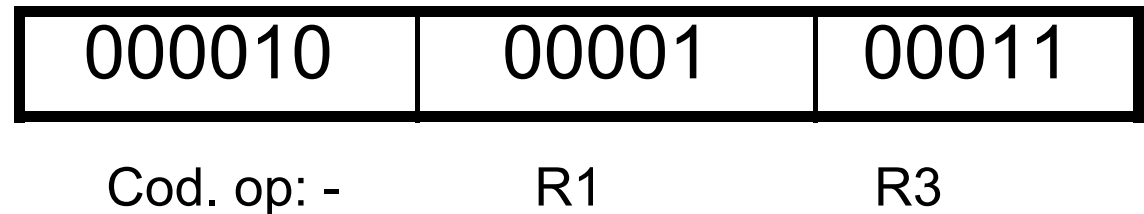
Prof. G. Ascia

- *Esempi*

$R1=R1+R2$



$R1=R1-R3$



# Unità Centrale di Processo (CPU)

---

Prof. G. Ascia

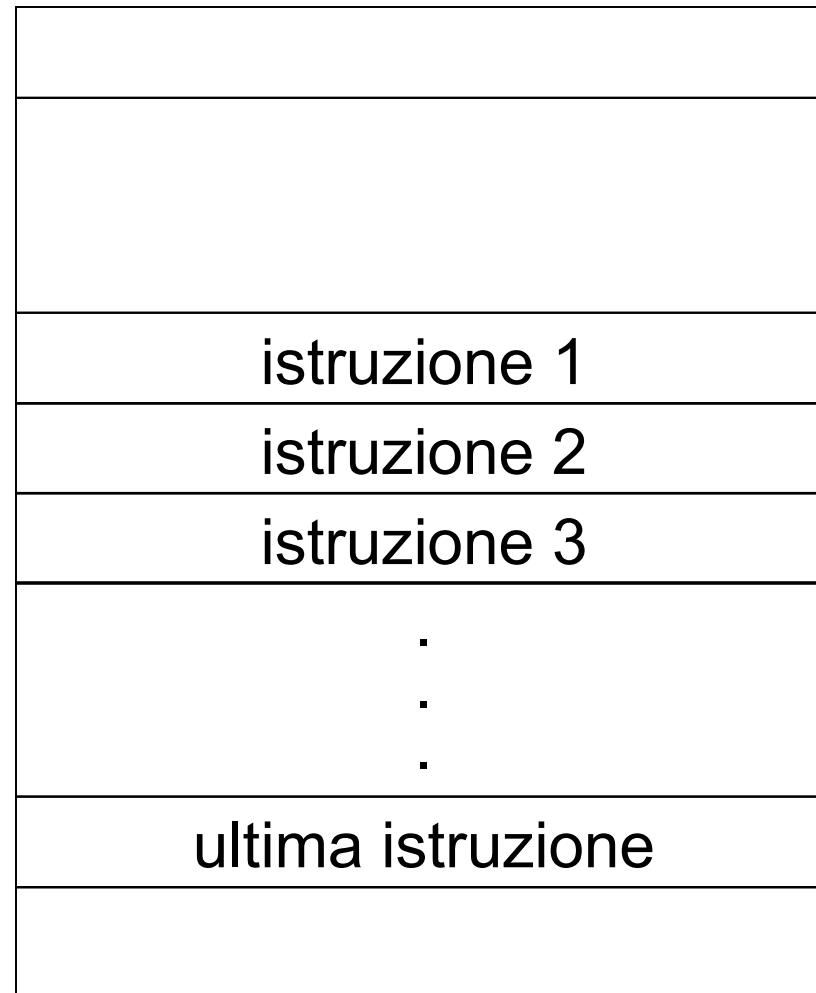
Un programma è eseguibile dalla CPU se:

- È costituito da istruzioni appartenenti al set di istruzioni della CPU
- Tutte le istruzioni sono codificate in binario e memorizzate sequenzialmente nella memoria centrale

# Unità Centrale di Processo (CPU)

---

Prof. G. Ascia



# Unità Centrale di Processo (CPU)

---

Prof. G. Ascia

Una CPU è costituita da due componenti:

## UNITÀ DI CONTROLLO:

esegue le istruzioni dei programmi, coordina le attività dei vari blocchi che costituiscono il computer, controlla il flusso delle istruzioni tra CPU e Memoria.

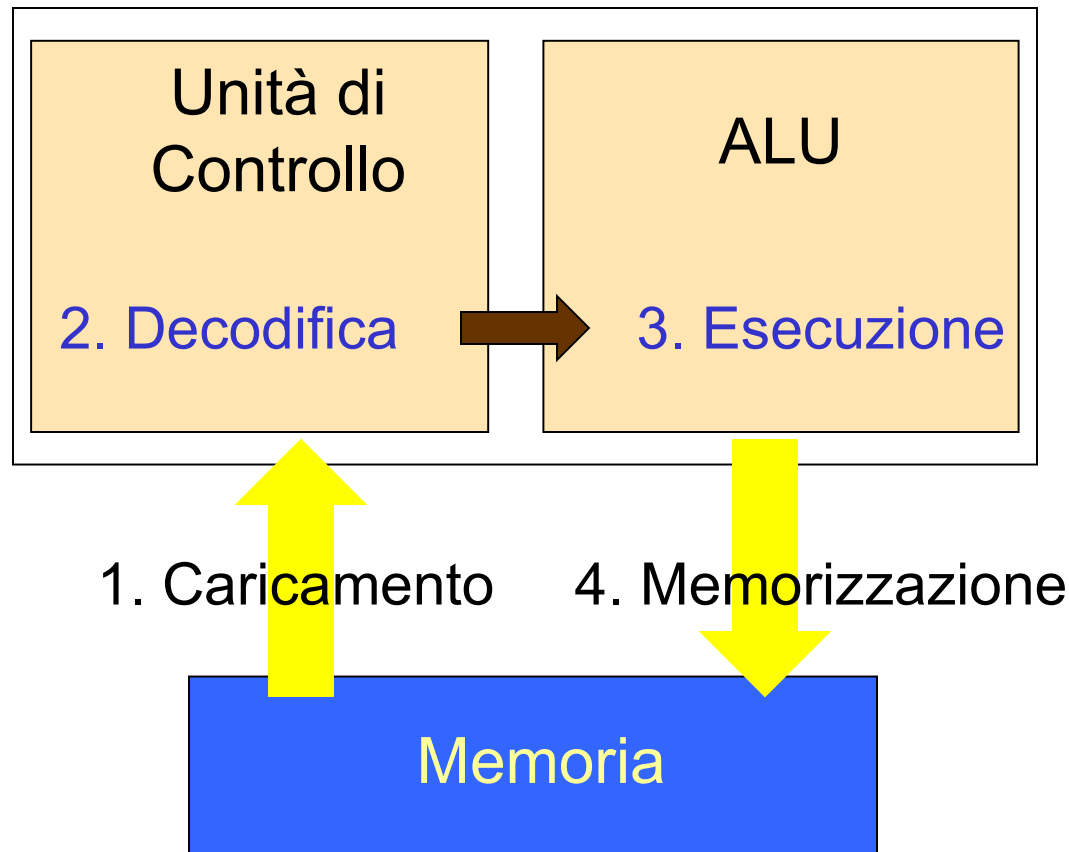
## UNITÀ LOGICO-ARITMETICA:

esegue le operazioni aritmetiche (somma, sottrazione, prodotto, divisione) e le operazioni logiche (ad es. verifica se un valore è maggiore, minore o uguale ad un altro valore)

# Meccanismo di Esecuzione

Prof. G. Ascia

Unità di controllo e ALU (Arithmetic Logic Unit) collaborano con la memoria in un processo distinto in quattro fasi che porta al completamento di un CICLO MACCHINA



# Meccanismo di Esecuzione

---

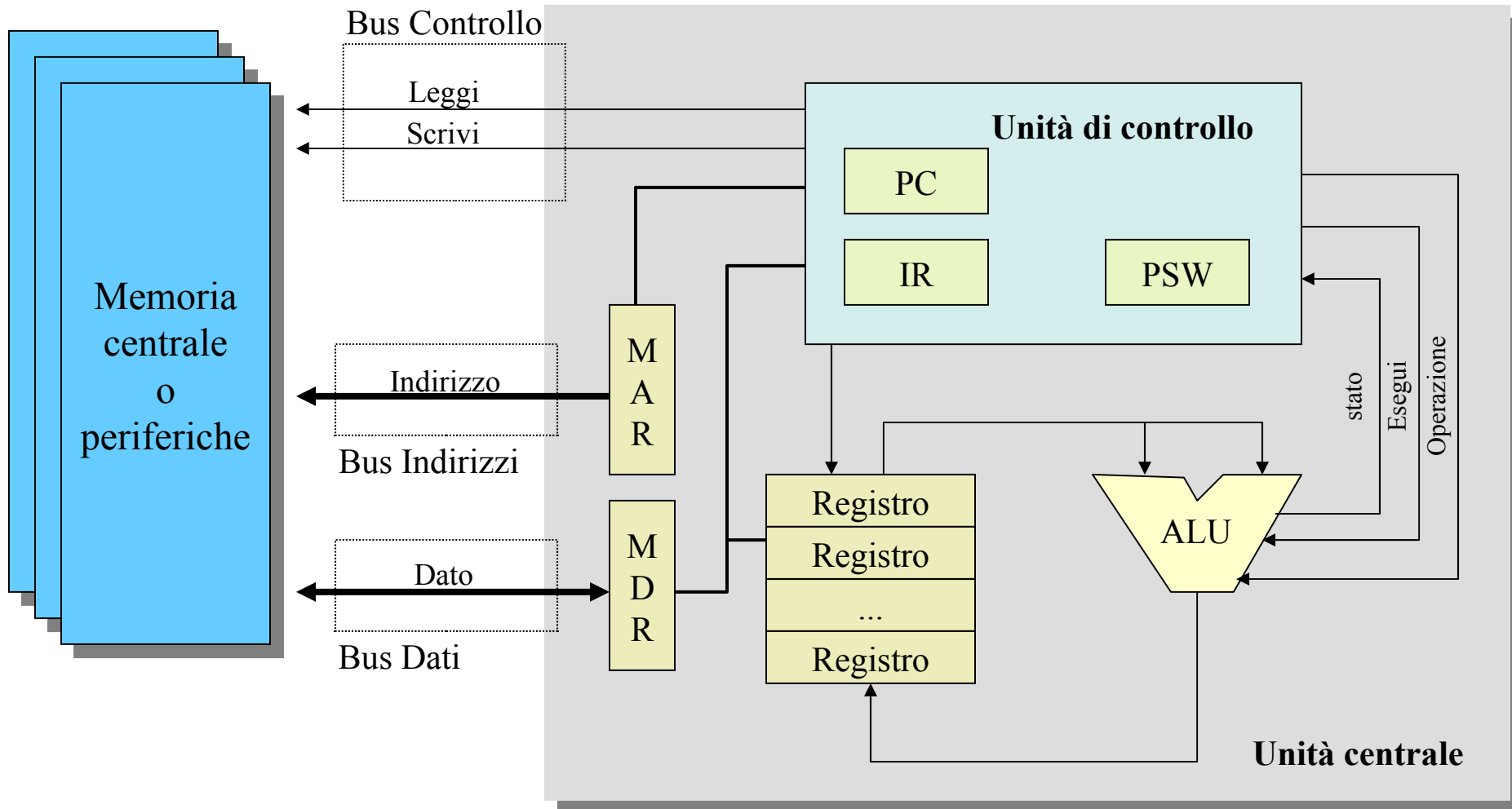
Prof. G. Ascia

All'interno della CPU vi sono delle unità di memorizzazione, i REGISTRI, su ciascuno dei quali è possibile memorizzare una word.

La dimensione di tale word (in bit) determina il parallelismo della CPU ovvero l'unità di informazione che la CPU può processare in parallelo durante la propria attività

# Registri della CPU

Prof. G. Ascia



# Registri della CPU

---

Prof. G. Ascia

## **PC** (Program Counter):

è utilizzato dalla CPU per memorizzare l'indirizzo della memoria dove risiede la prossima istruzione da eseguire. Ogni qualvolta la CPU termina l'esecuzione di una istruzione usa il PC per accedere alla prossima istruzione da eseguire.

## **IR** (Instruction Register):

viene usato dalla CPU per memorizzare l'istruzione appena letta dalla memoria



# Registri della CPU

---

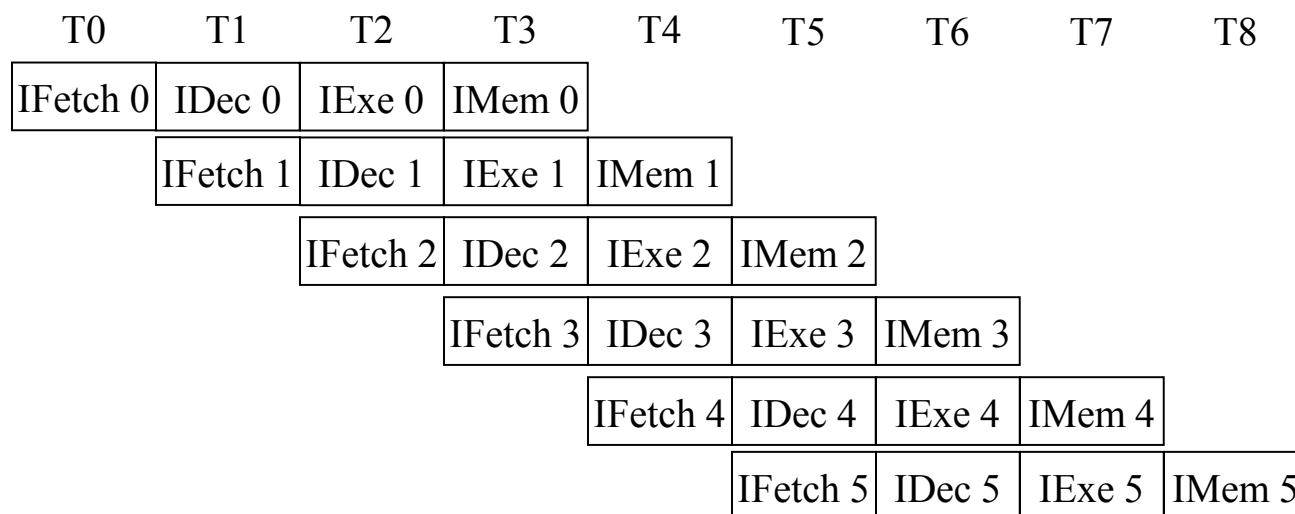
Prof. G. Ascia

- **MAR**: contiene l'indirizzo della locazione di memoria da leggere o scrivere
  - La dimensione di MAR determina l'ampiezza dello spazio di memoria fisica
  - Dalla fine degli anni '80 vengono prodotti microprocessori con bus indirizzi a 32 bit
- **MDR**: Registro attraverso il quale viene scambiata l'informazione tra la memoria e la CPU
  - Tradizionalmente la dimensione di MDR dà la misura del grado di parallelismo della macchina (8, 16, 32, 64 bit)
- **R0, R1,...Rn**: Registri di uso generale

# Incremento delle prestazioni: pipeline

Prof. G. Ascia

- Nella macchina con pipeline, come in una catena di montaggio tutte le attività (lettura, decodifica, esecuzione, memorizzazione) vengono svolte contemporaneamente, ma su istruzioni diverse.

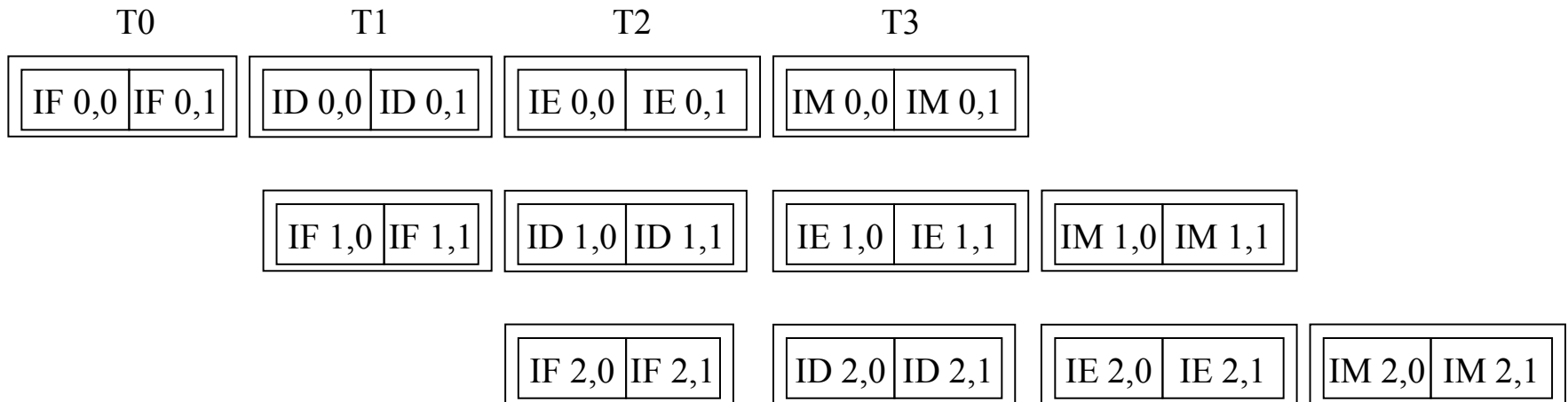


Con il pipeline non viene ridotto il tempo di esecuzione della singola istruzione, ma il tempo totale di esecuzione del programma

# Incremento delle prestazioni

Prof. G. Ascia

- Architetture superscalari o VLIW
  - In un processore più istruzioni sono eseguite in parallelo



# Incremento delle prestazioni

---

Prof. G. Ascia

- Multicore: nello stesso chip sono presenti più processori (core)
  - Vengono aumentate le prestazioni senza aumentare la frequenza del clock
  - Minore consumo di energia a parità di prestazioni