



# Università degli Studi di Cassino e del Lazio Meridionale

## Corso di Calcolatori Elettronici *Gerarchia di Memoria*

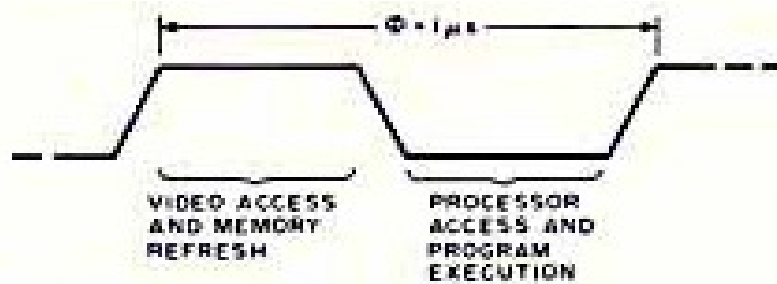
Anno Accademico 2015/2016

Francesco Tortorella

# 1977: DRAM più veloce del processore (© prof. Patterson)

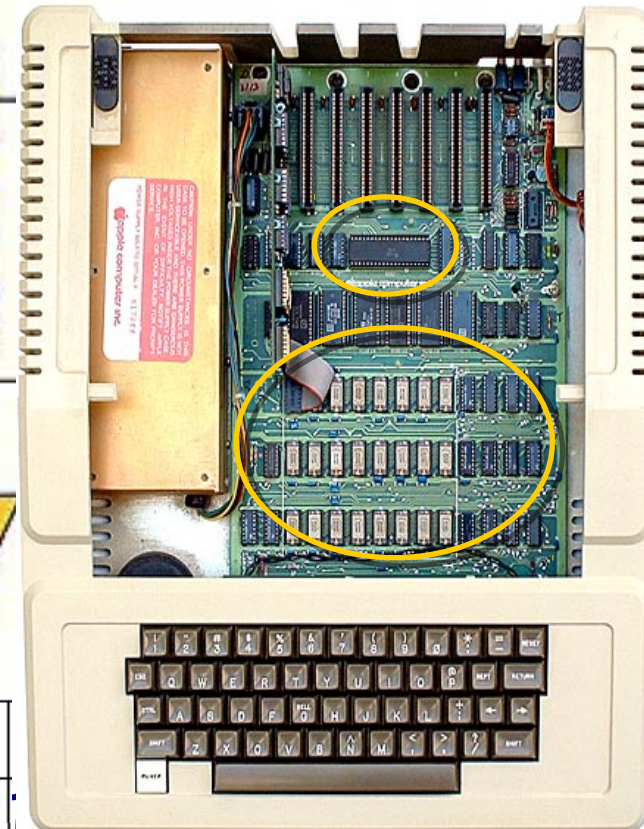
## TIMING:

6502 PROCESSOR'S  
 $\Phi_1$  CLOCK SHOWING  
WHEN AND BY WHOM  
MEMORY IS ACCESSED



Apple II (1977)

CPU: 1000 ns  
DRAM: 400 ns

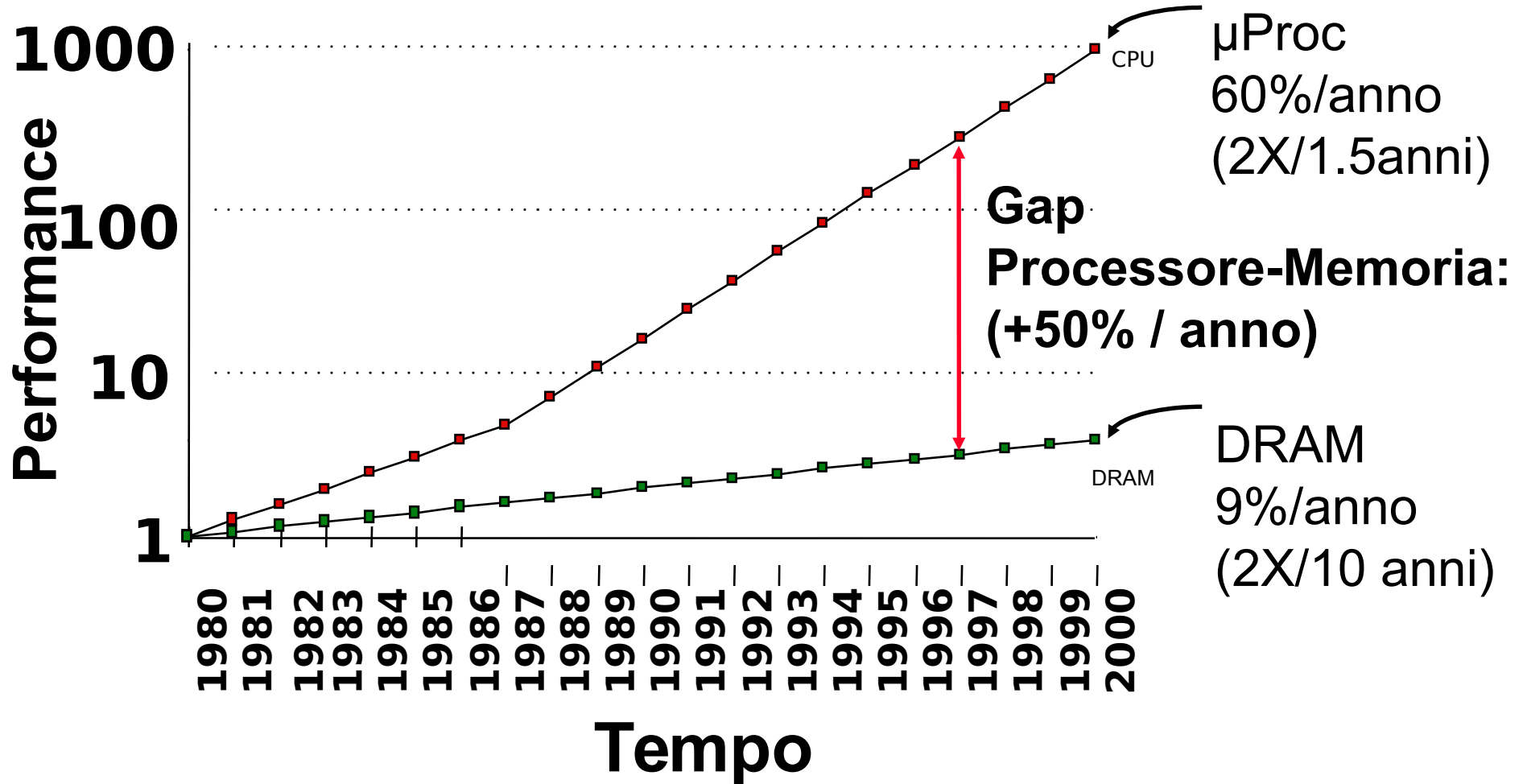


Steve Jobs

Steve Wozniak

RAM Complement	Apple II System
4K	\$ 1,298.00
48K	2,638.00

## II Performance Gap processore/memoria



# Gerarchia di memoria

*“Ideally one would desire an indefinitely large memory capacity such that any particular ... word would be immediately available ... We are forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding but which is less quickly accessible.”*

**Burks, Goldstine, von Neumann, 1946**

Requisiti teorici di un sistema di memoria:

**capacità infinita**

**velocità infinita**

Evidenza:

→ le memorie capienti sono lente

→ le memorie veloci sono costose

**Come realizzare un sistema di memoria che sia capiente, economico e veloce ?**

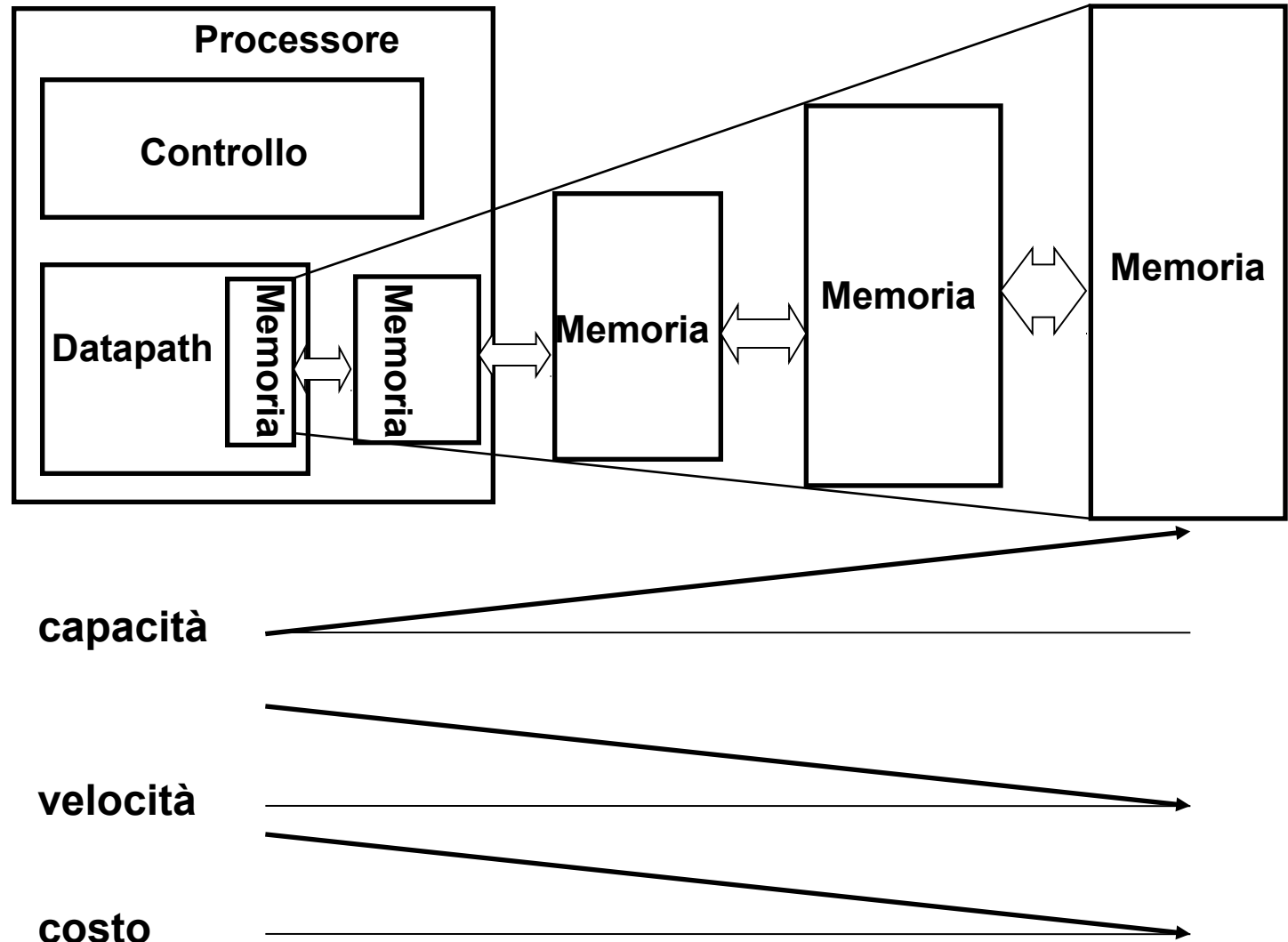
 **Un sistema basato su una gerarchia di memoria**

# Sistema di memoria gerarchico

Il sistema di memoria è composto da moduli di memoria aventi caratteristiche diverse e organizzati a livelli.

I dati memorizzati sono distribuiti sui vari moduli e possono essere trasferiti tra moduli adiacenti.

La distribuzione dei dati è realizzata in base al **principio di località**.



# Principio di località

**Un programma accede solo una parte (relativamente piccola) dello spazio di indirizzi in ogni istante.**

## Località temporale

Se una posizione di memoria viene acceduta allora tenderà presto ad essere acceduta di nuovo

## Località spaziale

Se una posizione di memoria viene acceduta allora le posizioni vicine tenderanno presto a essere accedute

# Principio di località: conseguenze

**Un programma accede solo una parte (relativamente piccola) dello spazio di indirizzi in ogni istante.**

## Località temporale

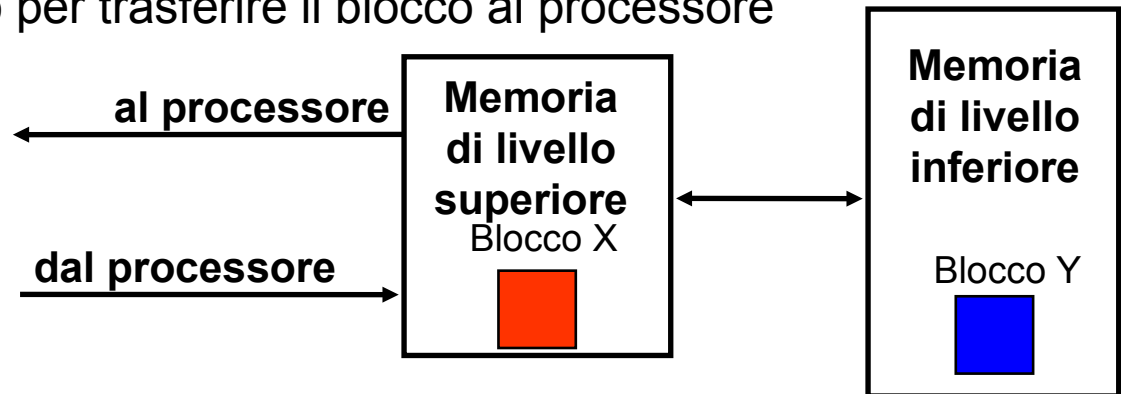
⇒ Mantenere vicino al processore i dati più recentemente acceduti

## Località spaziale

⇒ Avvicinare al processore blocchi di dati contigui

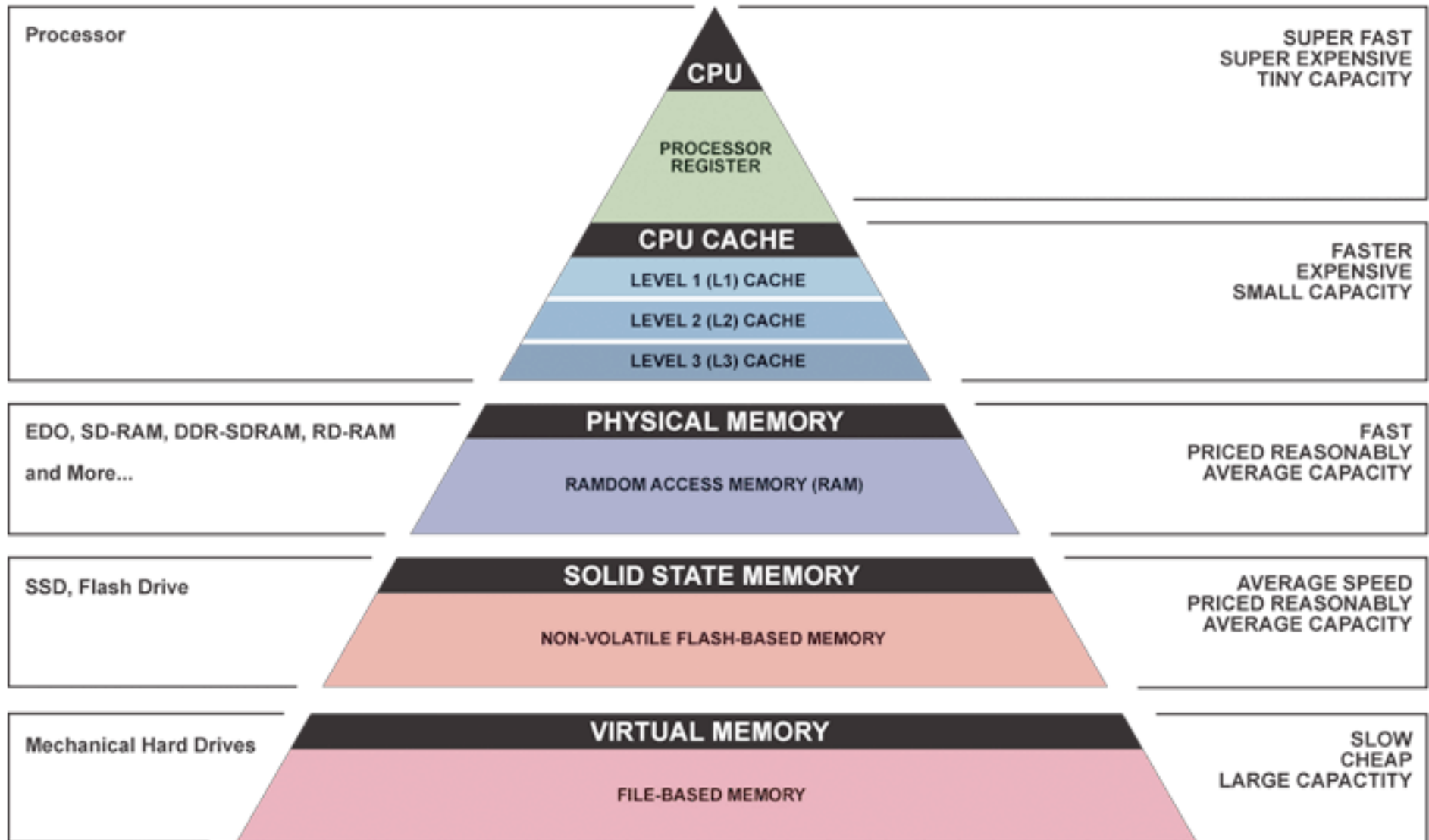
# Hit or Miss ?

- **Hit**: i dati richiesti sono presenti in un blocco della memoria di livello superiore (es.: blocco X)
  - **Hit Rate**: la frazione di accessi di memoria risolti nella memoria di livello superiore
  - **Hit Time**: tempo necessario per accedere ai dati presenti nella memoria di livello superiore. E' formato da:  
tempo per determinare hit/miss + tempo trasferimento
- **Miss**: i dati richiesti non sono presenti nella memoria di livello superiore e devono essere trasferiti dalla memoria di livello inferiore (es.: blocco Y)
  - **Miss Rate** =  $1 - (\text{Hit Rate})$
  - **Miss Penalty**: tempo necessario per rimpiazzare un blocco nella memoria di livello superiore + tempo per trasferire il blocco al processore
- **Hit Time << Miss Penalty**

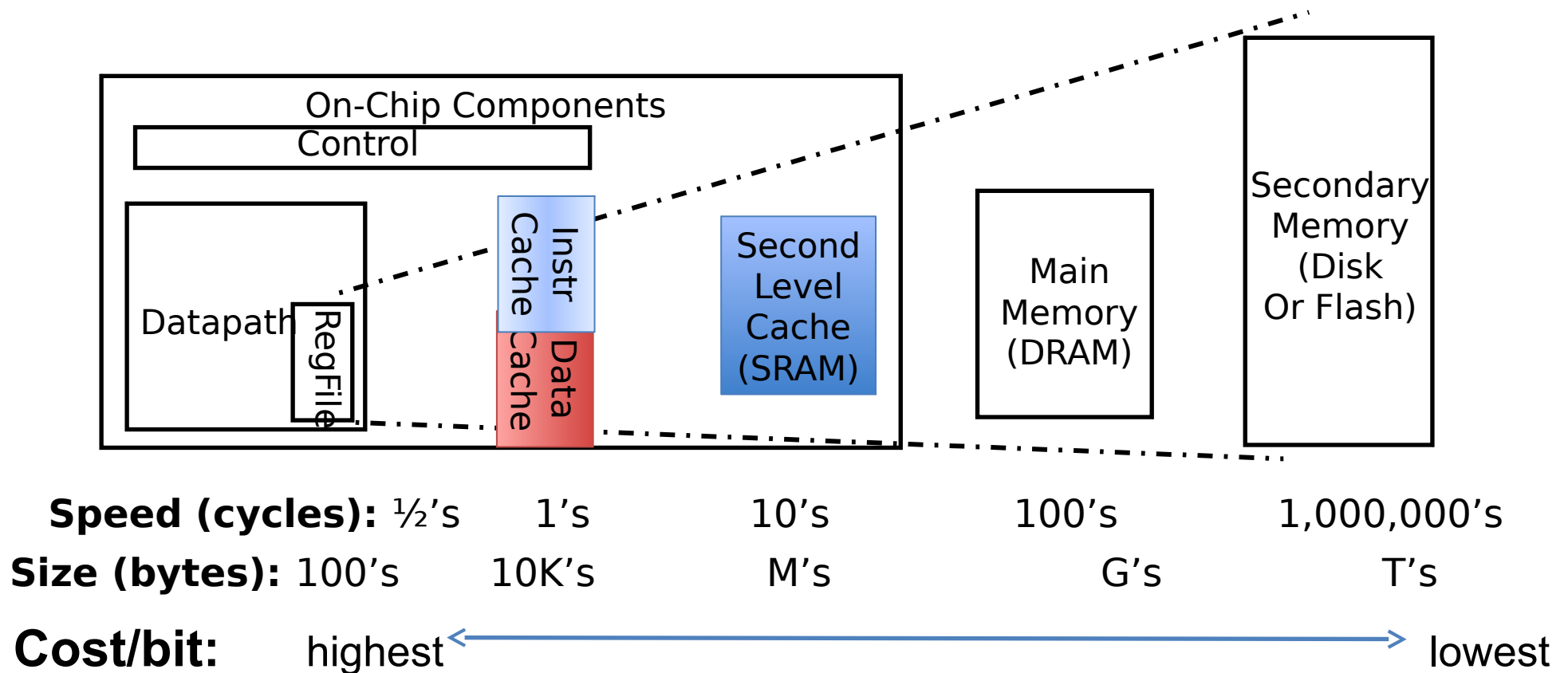




# Gerarchia di memoria: vista complessiva



# Gerarchia di memoria in un sistema attuale



**Principio di località + gerarchia di memoria** permettone di disporre di una grossa quantità di memoria alla tecnologia più economica e alla velocità offerta dalla tecnologia più veloce

# Chi gestisce la gerarchia di memoria ?

