

Informatica Grafica  
Corso di Laurea in Ingegneria Edile – Architettura

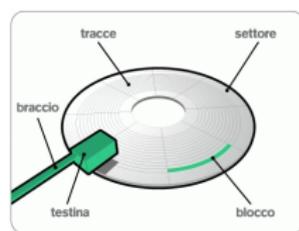
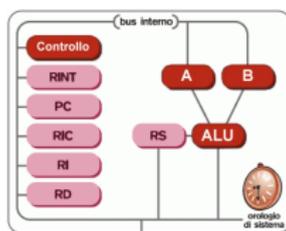
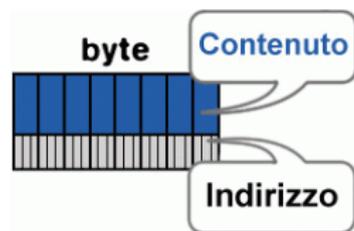
## **Architetture degli elaboratori**

Paolo Torroni

Dipartimento di Elettronica, Informatica e Sistemistica (DEIS)  
Università degli Studi di Bologna

Anno Accademico 2009/2010

# Architetture degli elaboratori



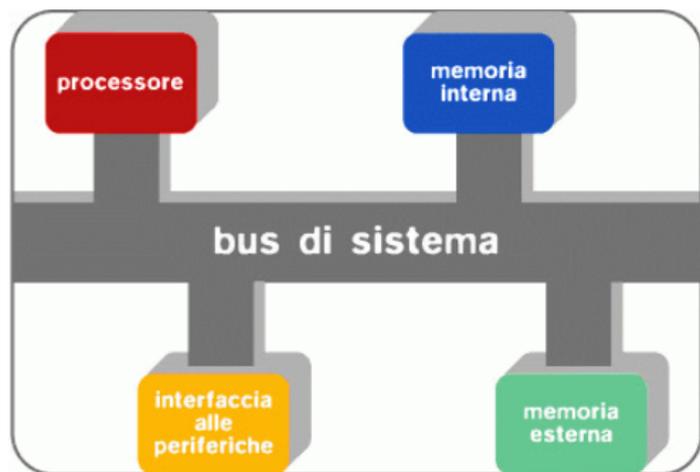
## ► Architetture degli elaboratori

- Com'è strutturato un computer? Quali sono le principali unità funzionali al suo interno?
- Hardware, software, sistema di numerazione binario
- Un'architettura di riferimento: la macchina di Von Neumann
- I linguaggi più vicini alla macchina
- CPU, RAM, memoria di massa, bus, USB, periferiche
- Come si misura un computer?
- La Legge di Moore
- Architetture parallele

# Parte I

## Un'architettura di riferimento

# La macchina di Von Neumann



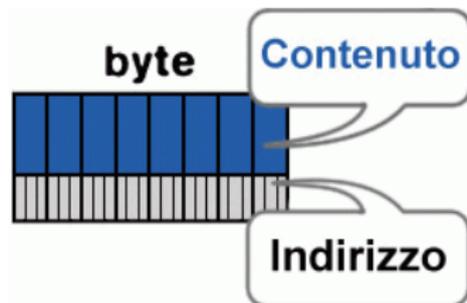
- ▶ **5 moduli funzionali**
- ▶ Nel computer: motherboard, circuiti integrati, porte (interfacce), collegamenti anche wireless.

# Funzionamento della CPU



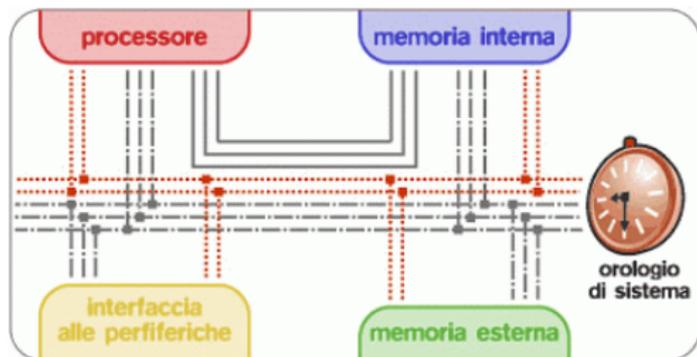
- ▶ Modello sequenziale.
- ▶ Sequenza dei passi scandita dal **clock** di sistema.

# Funzionamento della RAM



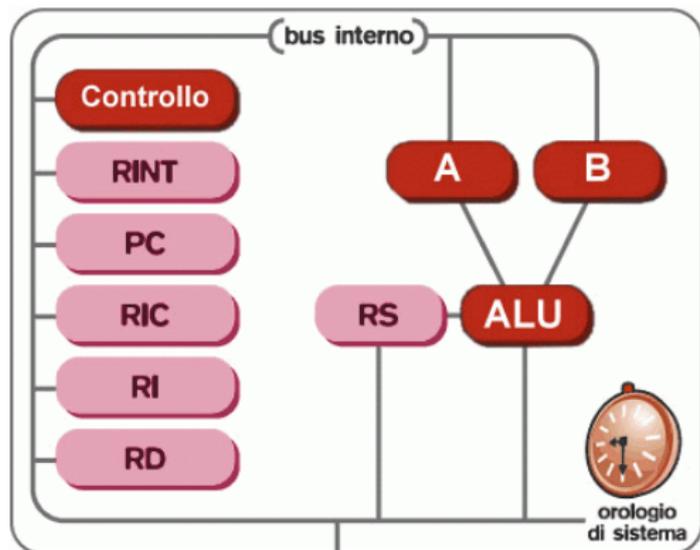
- ▶ Suddivisione in **parole** (es: 4 byte, 8 byte)
- ▶ identificate univocamente da un **indirizzo** (numero intero)
- ▶ Larghezza della parola (numero di byte) dipende dall'architettura del computer
- ▶ **Volatile**, e ad **accesso diretto** (Random Access Memory)
- ▶ Due operazioni possibili:
  - ▶ **LOAD** (lettura da RAM a CPU)
  - ▶ **STORE** (scrittura da CPU a RAM)

# Funzionamento del BUS di sistema



- ▶ Esistono **3 tipologie di connessione**
  - ⇔ **bus dati**: fa transitare il **contenuto** del registro dati dalla CPU a una delle altre unità e viceversa;
  - ⇒ **bus indirizzi**: fa transitare dalla CPU verso la memoria l'**indirizzo della parola** a cui si accede;
  - ⇔ **bus controllo**: fa transitare
    - ▶ dalla CPU il codice della operazione da eseguire,
    - ▶ dalla unità funzionale il codice risultato dell'operazione
- ▶ Larghezza del bus dipende dall'architettura

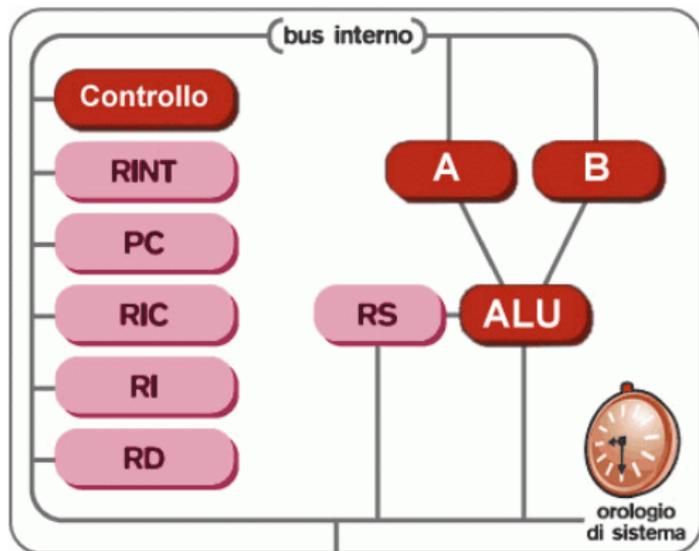
# La CPU



## ► Control Unit:

- regola l'esecuzione del ciclo della CPU,
- invia i segnali di controllo per i trasferimenti sul bus
- decodifica le istruzioni
- gestisce sia i trasferimenti all'interno della CPU sia quelli con il mondo esterno.

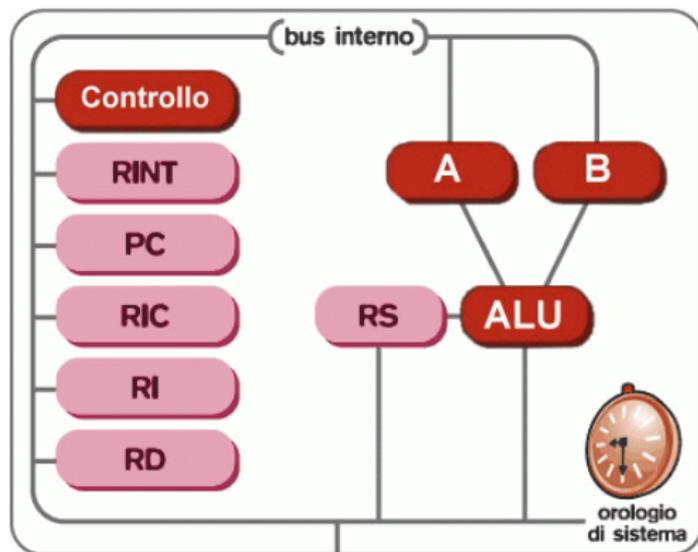
# La CPU



## ► Clock

- determina la frequenza delle operazioni da eseguire

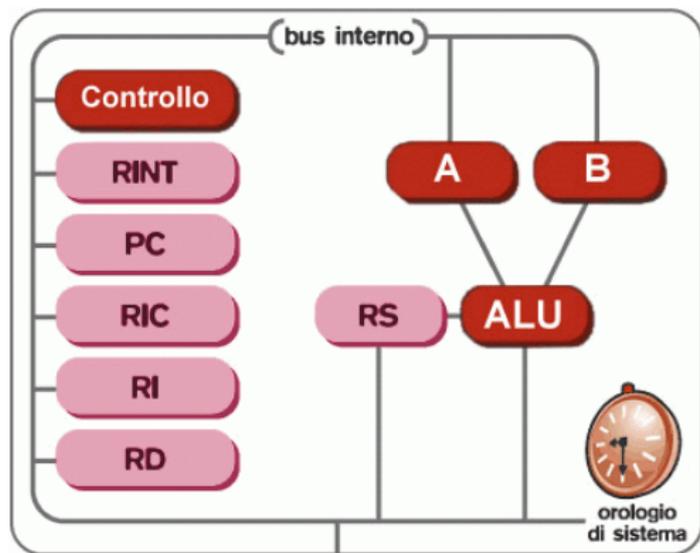
# La CPU



## ► Arithmetic Logic Unit

- realizza le operazioni aritmetiche e quelle logiche
- due operandi: A e B
- risultato memorizzato in A

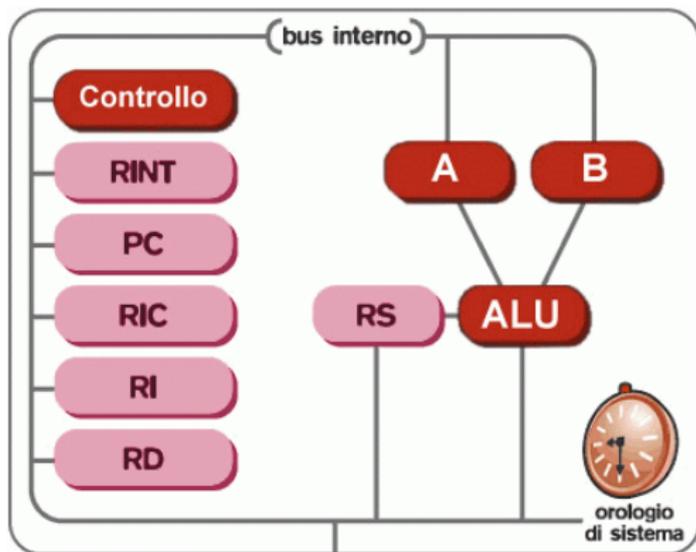
# La CPU



## ► Registri interni

- memorie locali che possono essere accedute alla velocità del clock della CPU
- molto più veloci della RAM (clock del bus  $\ll$  clock CPU)

# Registri della CPU



- ▶ **Dati (RD)**
- ▶ **Indirizzi (RI)**
- ▶ **Istruzione corrente (RIC)**
- ▶ **Contatore di programma (PC)**
- ▶ **Interruzioni (RINT)** (stato delle periferiche)
- ▶ **Operandi** (es: A e B)
- ▶ **Stato (RS)** (a volte: **flag**)

## Ricordate CR?

Tabella: Possibile codifica delle istruzioni di CR

INC	001	<b>somma</b> 1 alla cella $C_n$
RESET	000	<b>memorizza</b> 0 nella cella $C_n$
STORE	010	<b>leggi</b> dall'esterno un numero, memorizzalo in $C_n$
WRITE	011	<b>mostra</b> il contenuto della cella $C_n$
HALT	111	<b>stop</b>
JMPEQ	100	se $C_n = C_m$ vai all'istruzione ( $X$ )

## Codifica degli operandi per CR

- ▶ Gli operandi si riferiscono a celle di memoria
  - ▶ Esempio: memoria a disposizione: 4GByte, suddivisa in parole di 64 bit
  - ▶ Indirizzi distinti:  $4GB/8B = 2^{32}/2^3 = 2^{32-3} = 2^{29}$
- ⇒ servono 29 bit per gli indirizzi
- ▶ Bastano invece 3 bit per codificare 6 istruzioni distinte

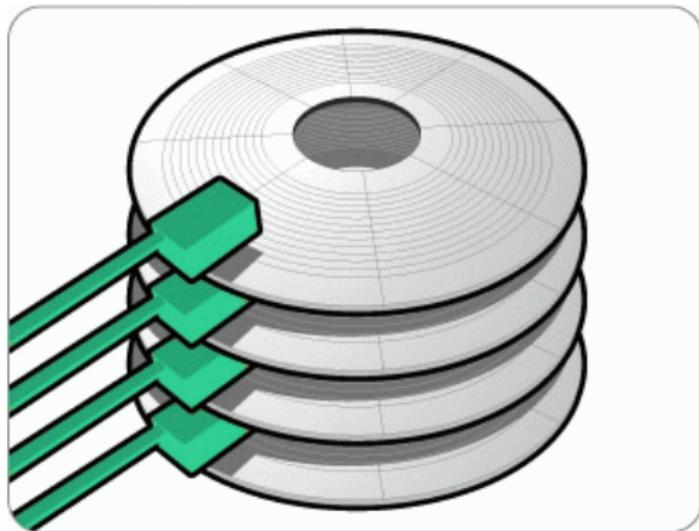
INC <i>cell_num</i>	001 xx...x
RESET <i>cell_num</i>	000 xx...x
STORE <i>cell_num</i>	010 xx...x
WRITE <i>cell_num</i>	011 xx...x
HALT	111
JMPEQ <i>c_1 c_2 (i)</i>	100 xx...x xx...x xx...x

- ▶ Lunghezza delle istruzioni:
  - ▶ 32 bit per INC, RESET, STORE, WRITE;
  - ▶ 3 bit per HALT,
  - ▶ 90 bit per JMPEQ

## Parte II

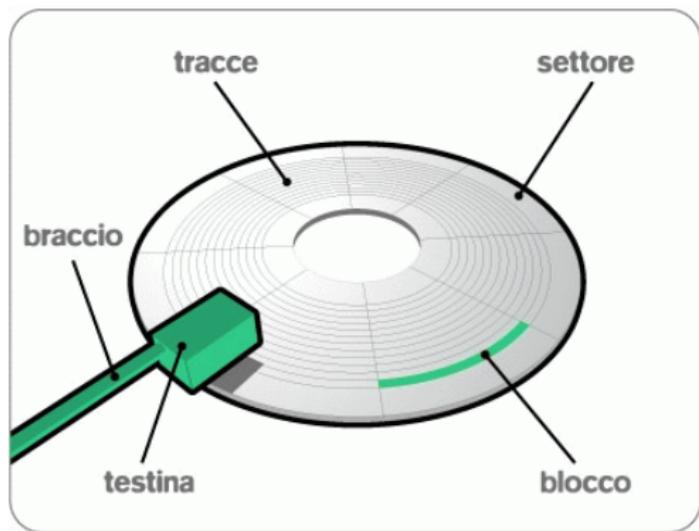
### Dispositivi di memoria e di I/O

## Memoria di massa



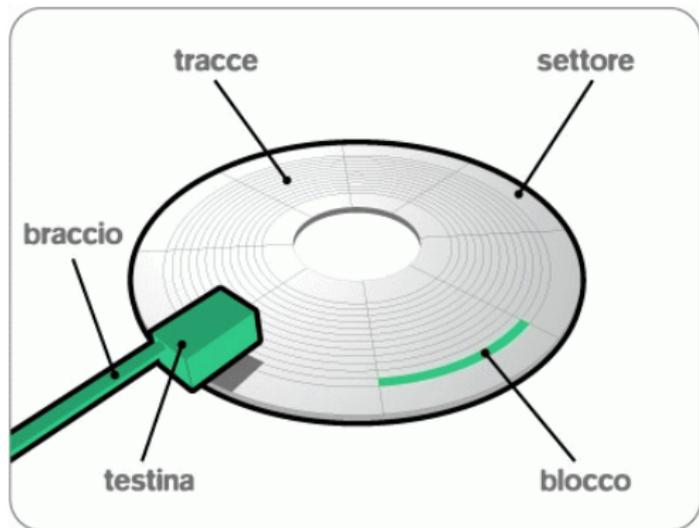
- ▶ **Hard disk:** dispositivo **magnetico**
  - ▶ Pila di dischi, ciascuno organizzato in **tracce** e **settori**

# Memoria di massa



- ▶ **Hard disk**: dispositivo **magnetico**
  - ▶ Pila di dischi, ciascuno organizzato in **tracce** e **settori**

# Memoria di massa



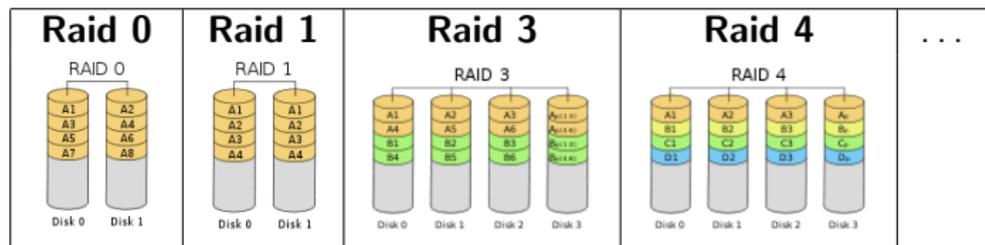
## ► **Hard disk:** dispositivo **magnetico**

- Pila di dischi, ciascuno organizzato in **tracce** e **settori**
- ciascun file occupa un numero intero di **blocchi**
- **allocazione** dei file: tramite FAT (**File Allocation Table**)
- operazioni di lettura/scrittura gestite da un **controller**

# Altri dispositivi di memoria

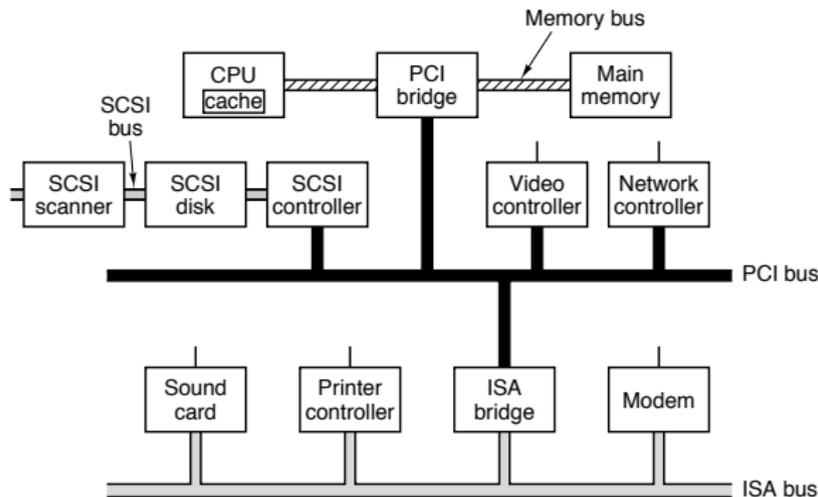
## ▶ RAID

- ▶ Gruppi di dischi rigidi che funzionano in parallelo
- ▶ Varie modalità, per aumentare prestazioni e/o robustezza
  - ▶ Raid 1 (*striping*): migliori prestazioni
  - ▶ Raid 2 (*mirroring*): migliore robustezza, metà spazio
  - ▶ Raid 3, 4: almeno 3 dischi; 1 usato per garantire robustezza; migliori prestazioni
  - ▶ ...



- ▶ Dispositivi **ottici**: **CD, DVD, Blue Ray**
  - ▶ utili come sistemi di archiviazione
- ▶ Dispositivi **a stato solido**: **USB stick, Flash cards**
  - ▶ utili per trasporto e assenza di testine di lettura

# Canali di comunicazione

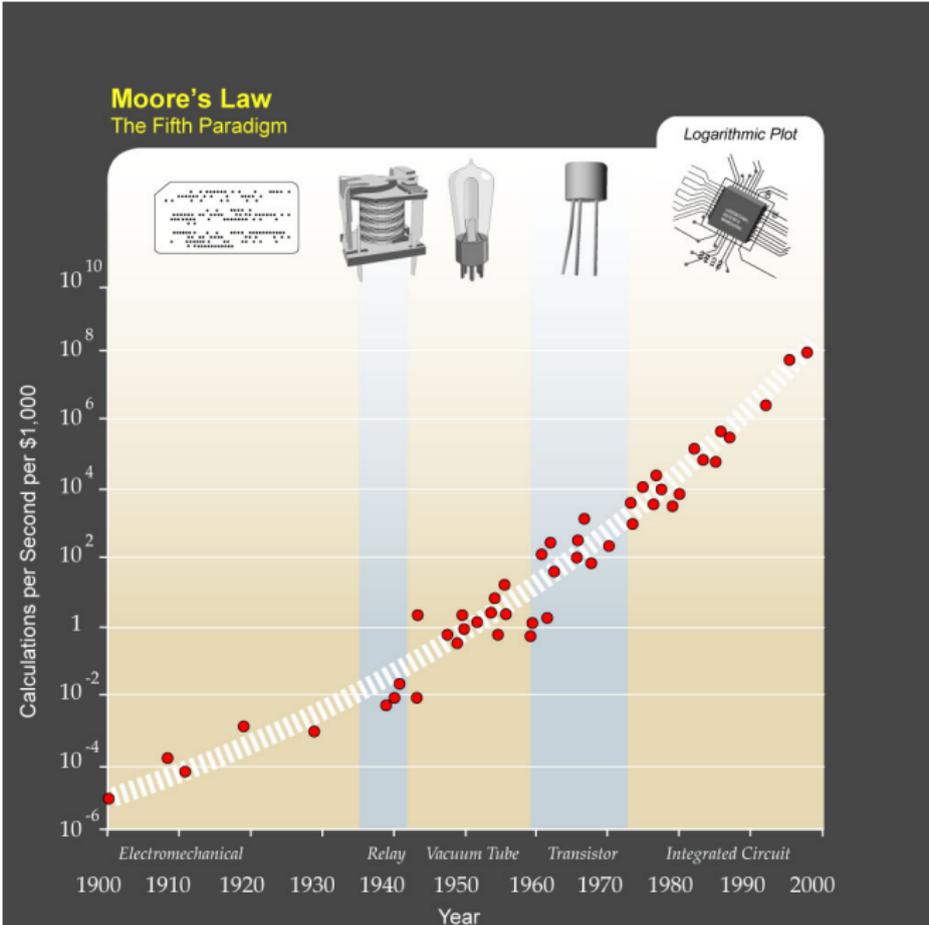


- ▶ **BUS** interni al sistema, di vario tipo: **Memory bus**, **PCI**, **ISA**
- ▶ Canali di comunicazione esterna, tramite interfacce:
  - ▶ **Porte seriali e parallele**
  - ▶ **USB Firewire** (IEEE 1394)
  - ▶ **Wi-Fi e Bluetooth**

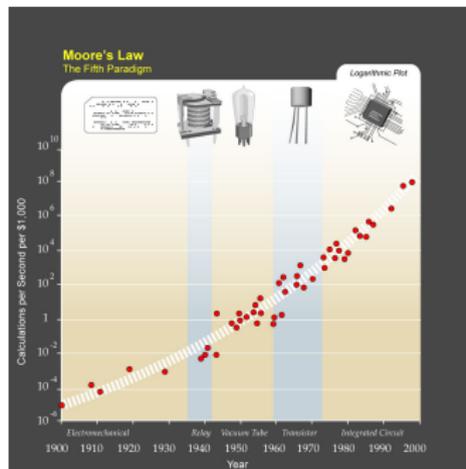
## Parte III

### Evoluzione delle architetture

# La Legge di Moore (Gordon Moore, 1965, Intel)



# La Legge di Moore (Gordon Moore, 1965, Intel)



- ▶ Crescita esponenziale dovuta a miglioramenti tecnologici
- ▶ Limiti fisici
- ▶ Per continuare a crescere: necessarie modifiche all'architettura (sequenziale → parallela)

# Migliorie alla macchina di Von Neumann

- ▶ Co-processori
- ▶ **Pipeline** della CPU
- ▶ Gerarchie di memorie (**cache**)
- ▶ Processori di canali (Direct Memory Access, **DMA**)
- ▶ Architetture **multi processore**

# Supercomputer



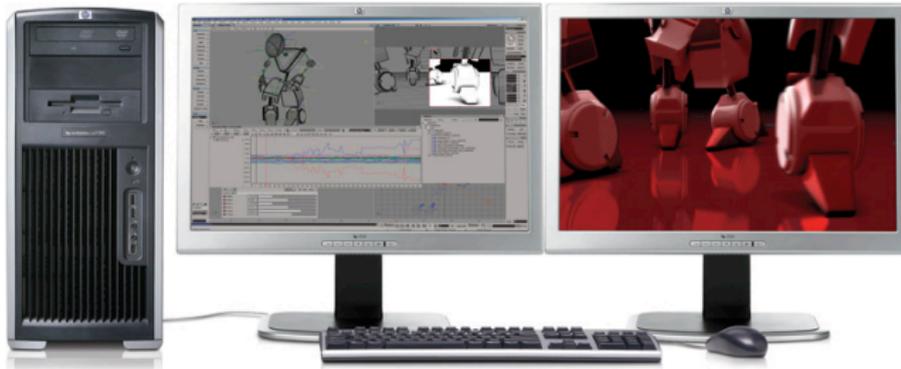
- ▶ Enti governativi, grossi progetti scientifici
  - ▶ IBM Roadrunner, **129.600 CPU**, > 1 Petaflop  
<http://www.lanl.gov/>  
[http://www.youtube.com/v/s\\_4rIQm0w28](http://www.youtube.com/v/s_4rIQm0w28)
    - ▶ SGI Pleiades, Sun Ranger, Bull JUROPA, Dawning Magic Cube, 30/60.000 CPU,  $2/6 \times 10^5$  GFlops
- (fonte: <http://www.top500.org>)

# Mainframe



- ▶ Grandi imprese, milioni di euro
- ▶ IBM System z10
  - <http://www.ibm.com/systems/z/>
    - ▶ B2B, cloud computing, data warehousing, virtualization, enterprise content management, transaction management, customer relationship management, ...
    - ▶ scalability, availability, security

# Server e Workstation



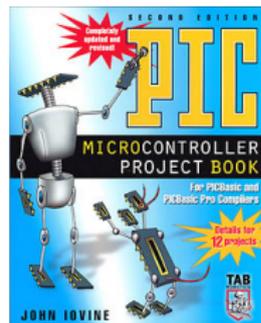
- ▶ PMI, migliaia/decine di migliaia di euro
- ▶ **Server:** per mettere risorse a disposizione di molti utenti contemporaneamente
  - ▶ servizi relativamente semplici
  - ▶ mainframe su piccola scala
- ▶ **Workstation:** orientata al singolo utente, per compiti che richiedono molte risorse
  - ▶ ambito grafico, progettazione CAD/CAM

# Personal Computer



- ▶ Uso personale sia domestico che lavorativo
- ▶ Costo: centinaia/migliaia di euro
- ▶ Decine di miliardi di personal computer prodotti nel mondo

# Microcontrollori



- ▶ Elettrodomestici, automobili, robotica, etc.
- ▶ Ovunque
- ▶ Euro/decine di euro



Handouts and all other material for **Informatica Informatica Grafica per Ingegneria Edile-Architettura**, Università di Bologna - A.A. 2009/2010 by Paolo Torroni is licensed under a **Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License**.

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/>

Based on a work at University of Bologna, Italy. <http://www.unibo.it/>

Paolo Torroni's Web site: <http://lia.deis.unibo.it/~pt/>

Composed using the  $\LaTeX$  Beamer Class, <http://latex-beamer.sourceforge.net/>