

# Architettura degli Elaboratori


Università degli Studi di Padova

Facoltà di Scienze MM.FF.NN.

**Corso di Laurea in Informatica**

*docente: Alessandro Sperduti*


## Informazioni Generali

- Lucidi ed esercizi disponibili in formato elettronico  
<http://www.math.unipd.it/~sperduti/architettura1.html>
- Modalità d'esame : 
  - di norma: scritto *obbligatorio* con orale *opzionale*
  - in casi eccezionali (**es. studente che copia durante la prova d'esame scritta**): scritto ed orale *obbligatori*
- Compitino intermedio sulla 1<sup>a</sup> parte: 15 Novembre
- Appelli d'esame (iscrizione tramite **UNIWEB**):
  - 13 Dicembre, 10 Gennaio, Marzo, Luglio, Settembre
  - chi supera il compitino (voto  $\geq 18$ ) ed è soddisfatto del voto, agli appelli d'esame deve solo sostenere l'esame sulla 2<sup>a</sup> parte; voto finale dato dalla media dei voti sulle singole parti

## Informazioni Generali

**ATTENZIONE !**



- Registrazione dell'esame: 
  - **SOLO** nelle date deputate alla registrazione (una data per appello d'esame)
  - entro l'anno accademico (30 Settembre 2011), altrimenti si **PERDE** il voto



## Organizzazione dell'insegnamento

- Circa 54 ore di lezione in aula LUM250
  - da Lunedì a Giovedì  
9:30-11:30
- 10 ore in aula informatica LabP140
  - dalle 14:30 alle 16:30
    - Novembre: 8, 15, 22, 29
    - Dicembre: 6
  - LabP140: Via Paolotti, III piano

## Materiale utile

- Libro di testo
  - Titolo: Architettura e Organizzazione dei Calcolatori – Progetto e Prestazioni
  - ottava edizione
  - Autore: William Stallings
  - Casa editrice: Pearson Prentice Hall, 2010
  
- Sito web del libro:
  - [WilliamStallings.com/COA/COA8e.html](http://WilliamStallings.com/COA/COA8e.html)

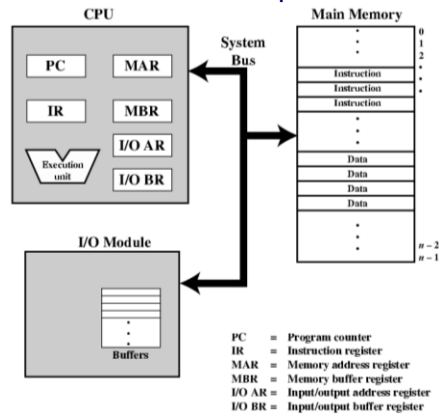
## Sommario degli argomenti di lezione

- Introduzione (capitolo 1)
- Evoluzione dei calcolatori (capitolo 2)
- [Struttura del Calcolatore](#) (capitolo 3)
- [Memoria Cache](#) (capitolo 4)
- [Memoria Interna](#) (capitolo 5)
- [Memoria Esterna](#) (capitolo 6)
- [Input/Output](#) (capitolo 7)
- [Cenni Logica Booleana, Reti Combinatorie e Sequenziali, Microprogrammazione](#)  
(<http://www.box.net/shared/4zcr479iqj#/shared/4zcr479iqj/1/26116716/270739814/1>, capitolo 16)
- [Sistemi di Numerazione](#)  
(<http://www.box.net/shared/4zcr479iqj#/shared/4zcr479iqj/1/26116716/270740128/1>)
- [Aritmetica del Calcolatore](#) (capitolo 9)
- [Linguaggi Macchina](#) (capitoli 10 e 11)
- [Struttura e Funzione della CPU, Pipelining](#) (capitoli 12 e 14)
- [Processori RISC](#) (capitolo 13)
- [Cenni sui Calcolatori Multicore](#) (capitolo 18)

# Struttura del Calcolatore

Cercheremo di rispondere alle seguenti domande:

- Qual'è la funzione di un calcolatore ?
- Eseguire un programma
- Quali sono le sue componenti ?



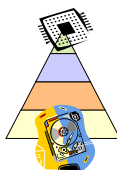
# Memoria Cache



Cercheremo di rispondere alla seguente domanda:

- Memorie RAM:
  - quelle veloci sono molto costose e di capacità limitata;
  - quelle di grossa capacità sono lente ma economiche;
- come è possibile combinare l'uso di questi due tipi di memorie in modo da avere una memoria RAM:
  - di grossa capacità
  - abbastanza veloce
  - economica ?

Gerarchia di memoria:



- la CPU richiede i dati necessari alla memoria veloce (Cache)
- se la Cache non li contiene, li richiede alla memoria più lenta che contiene tutti i dati
- quindi dati riferiti frequentemente si troveranno in Cache



## Memoria Interna



Cercheremo di rispondere alle seguenti domande:

- Che tipo di memoria, e quale tecnologia, viene usata per la memoria Cache ?

**SRAM (static RAM)**

- Che tipo di memoria, e quale tecnologia, viene usata per la RAM più lenta ma più capiente ?

**DRAM (dynamic RAM)**

- Come si possono organizzare più moduli di DRAM per avere un accesso più veloce ed efficiente ?

**DRAM sincrona (DRAM RamBus)**



## Memoria Esterna



Cercheremo di rispondere alle seguenti domande:

- Come viene memorizzata l'informazione in un disco rigido ?
- Come si accede all'informazione in un disco rigido ?
- Come si possono usare più dischi rigidi in parallelo in modo da recuperare più velocemente le informazioni e preservarla da guasti ?

**Redundant Array of Independent Disk (RAID)**


- Come viene memorizzata l'informazione in un CD-ROM o DVD ?
- Con quale tecnologia ?



## Input/Output

Cercheremo di rispondere alle seguenti domande:

- Come vengono organizzate e gestite le operazioni di input/output fra la CPU ed i dispositivi esterni (disco rigido, tastiera, video, ...) ?
- Cosa è un modulo di I/O ? Quali funzionalità svolge ? Quale è la sua struttura ?
- Quali sono le alternative possibili di gestione dell' I/O ?
  - I/O da programma
  - I/O driven (guidato da interrupt)
  - accesso diretto alla memoria (DMA)
- Cosa è un canale o processore di I/O ?




## Porte Logiche e Circuiti

Cercheremo di rispondere alle seguenti domande:

Simbolo circuitale	Simbolo algebrico	Tabella verità															
AND	$A \cdot B = C$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	0	0	0	0	1	0	1	0	0	1	1	1
A	B	C															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
OR	$A + B = C$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	C	0	0	0	0	1	1	1	0	1	1	1	1
A	B	C															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
NOT	$A = \bar{B}$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	B	0	1	1	0									
A	B																
0	1																
1	0																

- Come vengono realizzate le funzioni logiche in hardware ?
- Come si realizza la CPU in hardware ? C'è differenza fra i circuiti di una unità esecutiva (ALU) rispetto ai circuiti che implementano una memoria (registro) ?
  - Reti Combinatorie
  - Reti Sequenziali
- In cosa consiste la *Microprogrammazione* ?



## Sistemi di Numerazione



Cercheremo di rispondere alle seguenti domande:

- Come vengono rappresentati i numeri decimali in un calcolatore ?
- Come si converte un numero in rappresentazione decimale in rappresentazione binaria, e viceversa ?
- Cosa è la rappresentazione esadecimale ? Perché è utile ?



## Aritmetica del Calcolatore



Cercheremo di rispondere alle seguenti domande:

- Come vengono rappresentati i numeri interi in un calcolatore ?
- ...e come vengono realizzate le relative operazioni aritmetiche ?
- Come vengono rappresentati i numeri reali in un calcolatore ? virgola fissa, virgola mobile
- ...e come vengono realizzate le relative operazioni aritmetiche ?



## Linguaggi Macchina

Cercheremo di rispondere alle seguenti domande:



- Che tipo di istruzioni sono eseguite a livello macchina ?
- Come vengono specificati eventuali operandi ?
- Dove risiedono gli operandi delle istruzioni ? Come si indica la loro locazione ?
- Perché esistono vari modi di *indirizzamento* per gli operandi ?
- Come si rappresentano i vari modi di indirizzamento per una istruzione ?
- Cosa è un *set di istruzioni* ?



## Struttura e funzione della CPU

Cercheremo di rispondere alle seguenti domande:

- Come si differenziano le CPU dal punto di vista dei registri e del loro uso ?
- Come può una CPU organizzare i calcoli relativi alla esecuzione delle istruzioni in modo da essere più efficiente ?

efficiente:

utilizzare al meglio tutte le componenti della CPU in modo da non avere componenti inattive (inefficienza)

**pipelining:**

- ogni istruzione, quando è eseguita, "passa" attraverso stadi separati di esecuzione (ad es.: fetch, decodifica, calcolo indirizzi,...);



- **idea:** mantenere tutti gli stadi attivi eseguendo più istruzioni, come in una catena di montaggio (ogni stadio lavora contemporaneamente su una istruzione diversa)





## Processori RISC



Cercheremo di rispondere alle seguenti domande:

- Quali sono le istruzioni più frequentemente utilizzate dai programmi scritti nei linguaggi ad alto livello ?
- Si può utilizzare la risposta alla prima domanda per decidere quale deve essere il set delle istruzioni da implementare per una CPU ?

### Reduced Instruction Set Computer (RISC):

- un ridotto insieme di istruzioni, tutte dello stesso formato
- un elevato numero di registri o l'impiego di un compilatore che ne ottimizza l'uso
- particolare attenzione all'ottimizzazione della pipeline




## In laboratorio:

- Simulatore di Cache
- Simulatore di CPU
  - Linguaggio macchina
  - Memoria
  - Registri
  - Esecuzione delle istruzioni
- Simulatore di Pipeline



# Introduzione (cap.1)



## Architettura e organizzazione

- Architettura: caratteristiche visibili al programmatore
  - Istruzioni
  - Spazio (numero bit) usato per rappresentare i dati
  - Tecniche di indirizzamento della memoria
- Organizzazione: unità operative e loro connessioni
  - Interfacce tra calcolatore e periferiche
  - Tecnologia per le memorie

## Esempio

- Istruzione per la moltiplicazione:
  - Decidere se è disponibile, è una decisione architeturale
  - Come implementarla (circuito per la moltiplicazione o somme ripetute) è una decisione di organizzazione (costo, velocità, ...)
- Modelli diversi della stessa marca: stessa architettura, organizzazione diversa
- Esempio: architettura dell'IBM 370 (dal 1970)
  - Fino ad oggi per calcolatori mainframe
  - Varie organizzazioni con costo e prestazioni diverse

## Struttura e funzione

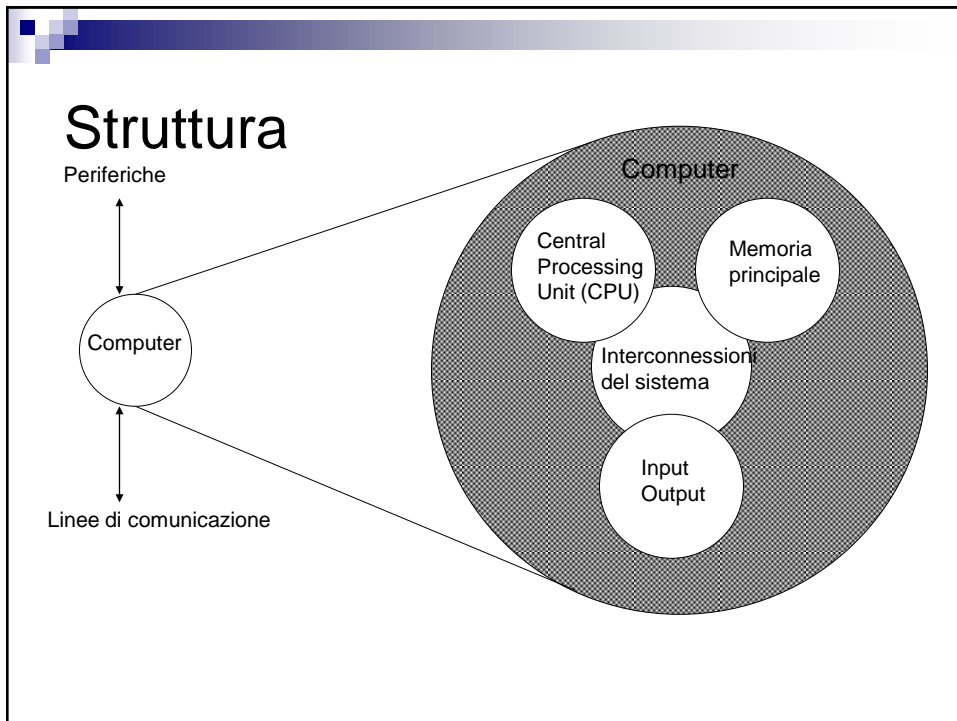
- Calcolatore:
  - Insieme di componenti connesse tra loro
- Visione gerarchica
  - Insieme di sottosistemi correlati
  - Ogni sistema ad un livello si basa sulla descrizione astratta del livello successivo
- Ad ogni livello
  - Struttura: come sono correlati i componenti
  - Funzione: cosa fa ciascun componente
- Descrizione top-down:
  - da componenti principali a sottocomponenti, fino a una descrizione completa dei dettagli

## Funzioni basilari di un calcolatore (livello più alto della gerarchia)

- Elaborazione dati
- Memorizzazione dati
- Trasmissione dati
  - Input/output o verso un dispositivo remoto
- Controllo
  - Delle tre funzioni sopra

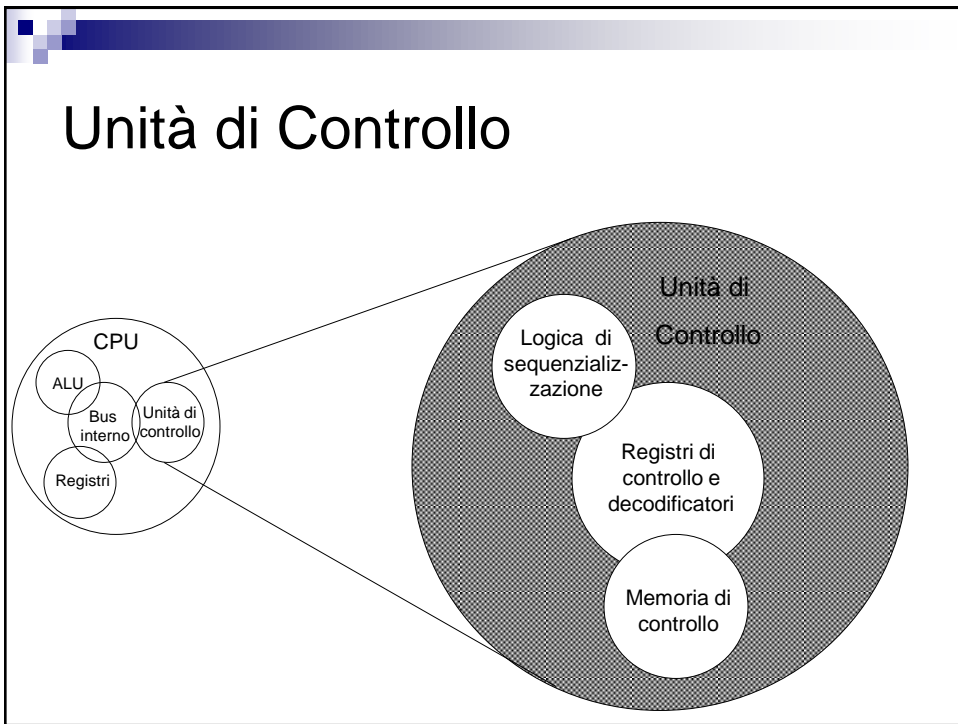
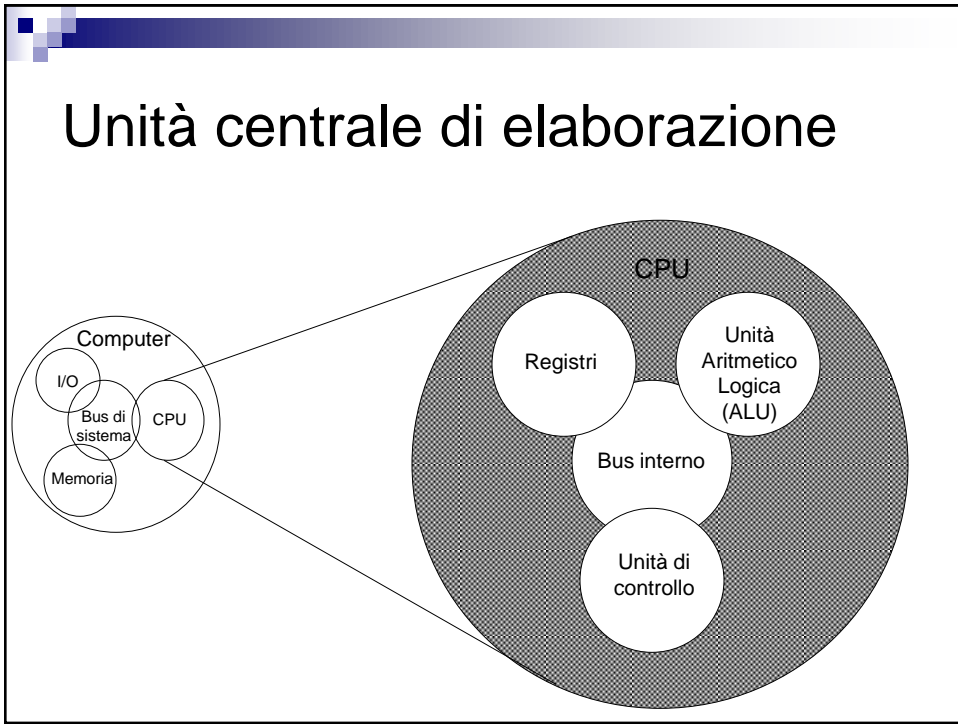
## Struttura (livello più alto della gerarchia)

- Quattro componenti principali:
  - Unità centrale di elaborazione (CPU)
    - Esegue le funzioni di elaborazione dati
  - Memoria centrale
    - Per immagazzinare i dati
  - I/O (input/output)
    - Per trasferire i dati tra calcolatore ed esterno
  - Interconnessioni
    - Per far comunicare CPU, memoria centrale, e I/O



## Central Processing Unit (Unità Centrale di Elaborazione)

- Unità di controllo
  - Controlla la sequenza di operazioni
- Unità aritmetico-logica (ALU)
  - Elaborazione dati
- Registri
  - Memoria interna della CPU
- Interconnessioni
  - Comunicazione tra unità di controllo, ALU e registri



## Perché studiare l'architettura dei calcolatori?

- Capire i compromessi costo-prestazioni
  - Esempio: scegliere il calcolatore migliore a parità di costo
    - spesa maggiore ma memoria più grande o frequenza di clock più alta e quindi maggiore velocità
- Supporto ai linguaggi di programmazione
  - Diverso a seconda delle architetture