

# Architettura degli Elaboratori

*Università degli Studi di Padova*

*Scuola di Scienze*


**Corso di Laurea in Informatica**

*docenti: Nicolò Navarin (lab), **Alessandro Sperduti***

**Nicolò Navarin**



## Informazioni Generali

- Lucidi ed esercizi disponibili in formato elettronico sulla piattaforma Moodle del Dipartimento di Matematica  
<https://elearning.unipd.it/math/>
- Modalità d'esame : 
  - di norma: scritto *obbligatorio* con orale *opzionale* (in presenza o online)
  - in casi eccezionali (**es. studente che copia durante la prova d'esame scritta**): scritto ed orale *obbligatori*
- Compitino intermedio 1<sup>a</sup> parte: 12 Novembre (da confermare)
- Appelli d'esame scritti (iscrizione tramite **UNIWEB**):
  - fine Gennaio, metà Febbraio, due date a Giugno/Luglio, una data a Settembre;
  - chi supera il compitino (voto  $\geq 17$ ) ed è soddisfatto del voto, agli appelli d'esame (uno qualunque) deve SOLO sostenere l'esame sulla 2<sup>a</sup> parte; voto finale dato dalla media dei voti sulle singole parti, ma bisogna avere la sufficienza in ogni parte.

## Organizzazione dell'insegnamento

- <https://gestionedidattica.unipd.it/PortaleStudenti/>
- Circa 60 ore di lezione
  - in aula 14D (Fiera) e *online* su Zoom (link su Moodle):
    - dalle 10:30 alle 12:30 il Lunedì
    - dalle 8:30 alle 10:30 il Martedì, Giovedì
  - **solo *online* dal 15 al 30 Ottobre**
- 10 ore di laboratorio (in aula 14D e *online*)
  - date:
    - Novembre: 2, 30
    - Dicembre: 14, 21
    - Gennaio: 11

## Materiale utile

### ■ Libro di testo

- Titolo: Computer Organization And Architecture
- Tenth Edition
- Autore: William Stallings
- Casa editrice: Pearson, 2018

### ■ Sito web del libro:

- <http://williamstallings.com/ComputerOrganization/>

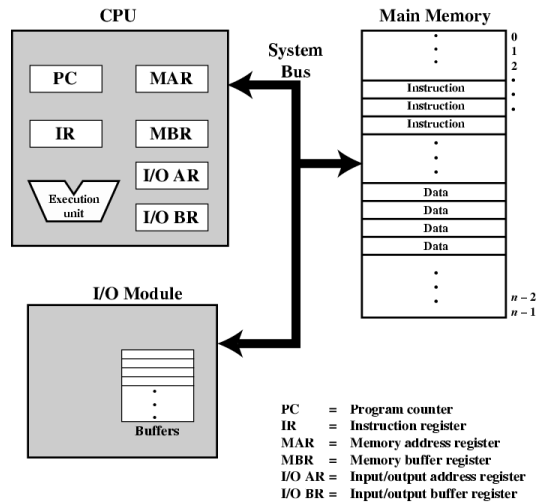
## Sommario degli argomenti di lezione

- Introduzione (capitolo 1)
- Evoluzione dei calcolatori (capitolo 1)
- Rappresentazione binaria, ottale, esadecimale; algebra di Boole
- [Struttura del Calcolatore](#) (capitolo 3)
- [Memoria Cache](#) (capitoli 4)
- [Memoria Interna](#) (capitolo 5)
- [Memoria Esterna](#) (capitolo 6)
- [Input/Output](#) (capitolo 7)
- [Sistemi di Numerazione](#) (capitolo 9)  
(<http://www.box.net/shared/4zcr479igj#/shared/4zcr479igj/1/26116716/270740128/1>)
- [Aritmetica del Calcolatore](#) (capitolo 10)
- [Cenni di Reti Combinatorie e Sequenziali, Microprogrammazione](#)  
(<http://www.box.net/shared/4zcr479igj#/shared/4zcr479igj/1/26116716/270739814/1>)
- [Linguaggi Macchina](#) (capitoli 12-13)
- [Struttura e Funzione della CPU, Pipelining](#) (capitolo 14)
- [Processori RISC](#) (capitolo 15)
- 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 ?



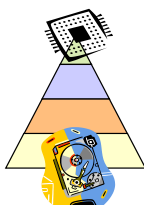
# Memoria Cache



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)**



# 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 funzionano i dischi SSD (*Solid State Disk*) ?
- 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:

- 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* ?

Simbolo circuitale	Simbolo algebrico	Tabella verità															
<b>AND</b> 	$A * B = C$	<table><tr><th>A</th><th>B</th><th>C</th></tr><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></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															
<b>OR</b> 	$A + B = C$	<table><tr><th>A</th><th>B</th><th>C</th></tr><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></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															
<b>NOT</b> 	$A = \bar{B}$	<table><tr><th>A</th><th>B</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	B	0	1	1	0									
A	B																
0	1																
1	0																



# 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)

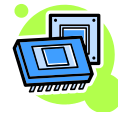
**pipelining:**

- 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