

# Calcolatori Elettronici by Palopoli

angeloperotti7

February 2025

## 0.1 turbo introduzione epica

### 0.1.1 struttura del corso

- prevalentemente lezioni teoriche
- qualche esercitazione su assembly :(
- suddivisioni del corso
  1. aritmetica dei calcolatori, cenni reti logiche
  2. Assembly
  3. elementi di architettura dei calcolatori

### 0.1.2 modalita' d'esame

- previsto parziale -> 6 domande in 30 min
- composizione esame:
  - prova scritta (gran parte scelta multipla)
  - prova orale in cui si discute lo scritto

### 0.1.3 altre info

- libro di riferimento: Struttura e progetto dei calcolatori Progettare con RISC-V David A Patterson, John L Hennessy
- mail: luigi.palopoli@unitn.it

# Chapter 1

## La rivoluzione dei calcolatori

### 1.1 la rivoluzione

- i calcolatori hanno creato la terza rivoluzione della società, portandoci in un mondo post industriale
- calcolatore o calcolatori?
  - calcolatori personali
    - \* buone prestazioni a costo ridotto
    - \* sw di terze parti (architetture aperte)
  - Server
    - \* poche applicazioni molto complesse
    - \* tantissime applicazioni molto complesse
  - embedded
    - \* vasto spettro di applicazioni
    - \* applicazioni spesso "dedicate"
    - \* requisiti non funzionali essenziali
      - consumi
      - rispetto di vincoli temporali
      - costi

Capire le prestazioni: per scrivere un programma con buone prestazioni il programmatore moderno deve:

- comprendere la gerarchia di memoria
- fare uso efficiente del parallelismo (multi-threading, GPU, distributed computing)
- TL;DR – > comprendere e conoscere l'organizzazione del calcolatore

Algoritmi	determina il numero di istruzioni di alto livello e di operazioni I/O
Linguaggi di programmazione, compilatori e architetture	Determina il numero di istruzioni macchina per ogni istruzione di basso livello
Processore e sistema di memoria	Determinano quanto velocemente e' possibile eseguire ciascuna istruzione
sistema di I/O (HW e sistema operativo)	Determina quanto velocemente possono essere eseguite le istruzioni

Software di sistema

- Sistema operativo
  - gestisce le operazioni di I/O
  - alloca la memoria
  - consente il multitasking
- Compilatore

- traduce da linguaggio di alto livello a *linguaggio macchina*

### Linguaggio macchina

- il componente base di un calcolatore sono le porte logiche, interruttori elettrici
- unità di base di informazione è il *bit*; un interruttore vale 1 se acceso, 0 se spento

programmare in Assembly

- estremamente difficile
- Assembly è un linguaggio mnemonico che viene tradotto in stringhe di bit da un traduttore (assembler)

## 1.1.1 Componenti del Calcolatore

struttura di Von Neumann

### Esempi di I/O

- mouse – inventato nel 1967 da Doug Engelbart
  - schermo LCD (Liquid Crystal Display)
    - i cristalli galleggiano in un fluido
    - ciascun cristallo è associato ad un punto (pixel)
    - tramite un campo elettrico (applicato a ciascun cristallo) si gestisce il colore e la luminosità
- come funziona
- l'immagine si compone di una matrice di pixel
  - ciascun pixel è associato a tre byte, ciascun associato a RGB
  - l'immagine viene memorizzata in una matrice (*frame buffer*) che è una memoria veloce che si aggiorna molte volte al secondo

### Dentro al PC

#### Il Processore

- Datapath: esegue le operazioni aritmetiche sui dati
- parte di controllo: indica al datapath alla memoria, e alle componenti di I/O cosa fare sulla base di quanto stabilito nel programma

#### LA memoria

- due tipi principali:
  - volatile (DRAM)
  - non volatile (dischi rigidi e memorie a stato solido)
- la memoria volatile viene usata per memorizzare dati e programmi mentre questi vengono eseguiti. viene definita memoria principale
  - allo spegnimento vengono persi i dati
- memoria non volatile o persistente è lo spazio di immagazzinamento dei dati (memorie di massa)
  - attualmente tre tipi principali:
    - \* memorie flash
    - \* dischi rigidi
    - \* CD/DVD
  - le memorie flash sono molto simili alle memorie RAM. L'idea è di memorizzare i dati intrappolando una carica elettrica

### 1.1.2 Astrazione

processo di omissione di dettagli che permette di gestire un progetto di grande complessita'

- il processore viene nascosto nei suoi dettagli, esportando come interfaccia l'insieme delle istruzioni macchina che il processore offre (*instruction set architecture, ISA*)
- insieme all'interfaccia del sistema operativo, l'ISA costituisce l'interfaccia binaria delle applicazioni (*Application Binary Interface*)
- una volta definita una ABI lo sviluppatore e' svincolato da come i dettagli HW sottostanti l'applicazione sono implementati

### 1.1.3 interconnessione

- i calcolatori esprimono maggior parte del loro potenziale quando interconnessi
  - LAN: Local Area Network, condivisione di risorse
  - WAN; Wide Area Network

### 1.1.4 Prestazioni

La complessita' di un moderno calcolatore rende la valutazione delle prestazioni complessa

Quando si parla di prestazioni e' importante capire di cosa si parla

Le prestazioni variano in base alle necessita' ed al contesto:

- Per il singolo utente interessa sapere quanto e' il tempo medio di una risposta
- Per il gestore di un centro di calcolo, interessa di piu' il *throughput*, cioe' quante task sono in gradi di completare nell'unita' di tempo

#### Tempo di esecuzione

- definiremo le prestazioni in termini di tempo di esecuzione
- per un calcolatore X

$$Prestazioni_X = \frac{1}{Tempo\ di\ esecuzione_X}$$

- Quindi dati due calcolatori diremmo:

$$Prestazioni_X \geq Prestazioni_Y \longleftrightarrow Tempo\ di\ esecuzione_X \leq Tempo\ di\ esecuzione_Y$$

- in una macchina multi-programmata il tempo di esecuzione dipende anche dagli altri task attivi e dalle loro prioritaa
- il tempo di esecuzione della CPU tiene conto solo del tempo effettivamente speso per il task
- Tale tempo e' in parte dedicato al programma utente e in parte al sistema operativo

#### capire le prestazioni

- i moderni processori sono costruiti usando un segnale periodico che ne sincronizza le operazioni
- il ciclo di clock e' l'intervallo di tempo che intercorre tra due colpi di clock
- il ciclo di clock e' misurato in secondi, la frequenza in Hertz
  - esempio: un clock che va ad un Giga Hertz (10<sup>9</sup>Hz) equivale ad un periodo di clock pari a 10<sup>-9</sup>secondi
- il tempo di calcolo di un programma puo' essere espresso nella seguente maniera:

$$Tempo\ CPU\ Programma = Cicli\ di\ clock\ della\ CPU \times Periodo\ del\ clock = \frac{Cicli\ di\ clock\ della\ CPU}{Frequenza\ del\ clock}$$

- equazione classica delle prestazioni

$$TEMPO\ CPU = Numero\ istruzioni \times CPI \times Periodo\ Clock = \frac{Numero\ Istruzioni \times CPI}{Frequenza\ Clock}$$