

# Architettura degli Elaboratori - 1

Università degli Studi di Padova  
Facoltà di Scienze MM.FF.NN.  
**Corso di Laurea in Informatica**

docente: Alessandro Sperduti

## Contenuto del Corso

- 1. Struttura di un elaboratore**
  - Macchina astratta, livelli di astrazione
- 2. La CPU (*Central Processing Unit*)**
  - Parte operativa, parte controllo, microprogrammazione
- 3. Evoluzione delle architetture**
  - *Pipelining*, gestione della memoria, cache, CISC e RISC, la famiglia Intel
- 4. Gestione dell'I/O (*cenni*)**
  - Tastiera, mouse, dischi magnetici, dischi ottici, *display*, la rete

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 3

## Obiettivi del Corso

- Descrizione dell'architettura degli elaboratori *in ambito locale*  
Presentazione delle componenti principali
- Analisi dell'architettura  
Caratteristiche funzionali e software di gestione
- Principi basilari dei Sistemi Operativi

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 2

## Contenuto del Corso (*segue*)

- 6. Sistema Operativo**
  - Classificazioni e definizioni, struttura generale semplificata
  - La gestione dei processi, concorrenza, IPC
  - La gestione del processore, *kernel*, *scheduling*
  - La gestione della memoria, il file system (*cenni*)

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 4

## Testi di Riferimento

- J L Hennessy & D A Patterson  
**Computer Architecture A Quantitative Approach**  
ISBN 1-55860-329-8
- A S Tanenbaum  
**Modern Operating Systems**  
ISBN 0-13-031358-0 [anche per Architetture 2]  
(in italiano: **I Moderni Sistemi Operativi**,  
Jakson; disponibile presso Libreria Progetto)
- Lucidi disponibili in formato elettronico  
<http://www.math.unipd.it/~sperduti/architettura1.html>

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 5

## Struttura di un Elaboratore

*Indice (segue)*

- 1.2 L'architettura di un elaboratore
  - 1.2.1 Funzioni della RAM
  - 1.2.2 Funzioni della ROM
  - 1.2.3 Funzioni dell'I/O
  - 1.2.4 Funzioni dei Bus
  - 1.2.5 Funzioni della CPU

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 7

## Struttura di un Elaboratore

*Indice*

- 1 Struttura di un elaboratore
  - 1.1 Macchina astratta – livelli di astrazione
    - 1.1.1 Livello 0: Macchina digitale
    - 1.1.2 Livello 1: Macchina microprogrammata
    - 1.1.3 Livello 2: Macchina hardware
    - 1.1.4 Livello 3: Macchina virtuale (Sistema Operativo)
    - 1.1.5 Livello 4: Macchina programmatore
    - 1.1.6 Livello 5: Macchina utente finale

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 6

## 1. Struttura di un Elaboratore

- Macchina astratta - livelli di astrazione
- Modello di riferimento
  - ✓ l'interfaccia di I/O e le periferiche
  - ✓ la memoria (dati e programmi)
  - ✓ i bus (indirizzi e dati)
  - ✓ l'unità centrale

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 8

## 1.1 Macchina Astratta

- L'elaboratore è una macchina programmabile che utilizza un linguaggio molto rudimentale e poco espressivo (**linguaggio macchina**)
- Per renderne l'uso più facile ed efficace si introducono vari **livelli di astrazione** gerarchici
  - ✓ Perciò si parla di 'macchina astratta'

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 9

## 1.1 Macchina Astratta (segue)

- Punto di vista dell'**utente finale** `File Apri`
- Punto di vista del **programmatore in linguaggio evoluto** `i++;  
cout << i;`
- Punto di vista del **programmatore in linguaggio Assembler** `MOV AL,00  
INC [013F]`
- Punto di vista della **CPU** `3E 02 5A 25  
FF BA 08 3F`

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 11

## 1.1 Macchina Astratta (segue)

- Ogni livello è caratterizzato da un proprio linguaggio che utilizza i comandi (i servizi) messi a disposizione dal livello inferiore
- Ad ogni livello superiore
  - ✓ Aumenta l'astrazione e la facilità di utilizzo
  - ✓ Diminuisce la velocità di esecuzione

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 10

## 1.1.1 Livello 0: Macchina digitale

- Livello 5 : Macchina utente
  - Livello 4 : Macchina programmatore
  - Livello 3 : Macchina virtuale
  - Livello 2 : Macchina hardware
  - Livello 1 : Macchina microprogrammata (*gate*). Viene progettata dal costruttore dei vari componenti ed è puramente hardware.
  - Livello 0 : Macchina digitale**
- E' regolata dalle leggi dell'algebra booleana e dell'elettronica digitale. I suoi componenti sono le porte logiche**

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 12

### 1.1.2 Livello 1: Macchina microprogrammata

Livello 5 : Macchina utente  
 Livello 4 : Macchina programmatore  
 Livello 3 : Macchina virtuale  
 Livello 2 : Macchina hardware  
**Livello 1 : Macchina microprogrammata**  
 Livello 0 : Macchina digitale

**L'insieme delle regole logiche che governano il livello digitale e gli fanno svolgere le funzioni previste. Viene vista solo dal costruttore della CPU. Anche se caratterizzata da istruzioni e microprogrammi, è essenzialmente hardware. Può non esserci.**

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 13

### 1.1.4 Livello 3: Macchina virtuale (Sistema Operativo)

Livello 5 : Macchina utente  
 Livello 4 : Macchina programmatore  
**Livello 3 : Macchina virtuale**  
 Livello 2 : Macchina hardware  
 Livello 1 : Macchina microprogrammata  
 Livello 0 : Macchina digitale

**Offre un insieme di istruzioni più strutturate che agevolano l'accesso e l'uso delle risorse fisiche. Spesso oscura la natura ed il funzionamento dei livelli inferiori. Offre un linguaggio a 'chiamate di sistema'. E' prettamente software ed è indispensabile.**

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 15

### 1.1.3 Livello 2: Macchina hardware

Livello 5 : Macchina utente  
 Livello 4 : Macchina programmatore  
 Livello 3 : Macchina virtuale  
**Livello 2 : Macchina hardware**  
 Livello 1 : Macchina microprogrammata  
 Livello 0 : Macchina digitale

**E' governata dal linguaggio macchina. Offre visione ed accesso diretto a tutte le risorse fisiche, tramite una specifica interfaccia di livello (funzionalmente simile alle altre implementazioni). Fornisce un'interfaccia prettamente software (i.e. programmabile) ai livelli superiori.**

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 14

### 1.1.5 Livello 4: Macchina programmatore

Livello 5 : Macchina utente  
**Livello 4 : Macchina programmatore**  
 Livello 3 : Macchina virtuale  
 Livello 2 : Macchina hardware  
 Livello 1 : Macchina microprogrammata  
 Livello 0 : Macchina digitale

**La macchina vista dal programmatore a livello evoluto. In funzione del linguaggio e degli scopi può anche accedere direttamente al livello 2. I servizi di questo livello possono essere interpretati o compilati. Lo sviluppatore è ancora uno specialista.**

Struttura di un elaboratore      Architettura degli elaboratori -1      Pagina 16

### 1.1.6 Livello 5: Macchina utente

#### Livello 5 : Macchina utente

- Livello 4 : Macchina programmatore
- Livello 3 : Macchina virtuale
- Livello 2 : Macchina hardware
- Livello 1 : Macchina microprogram
- Livello 0 : Macchina digitale

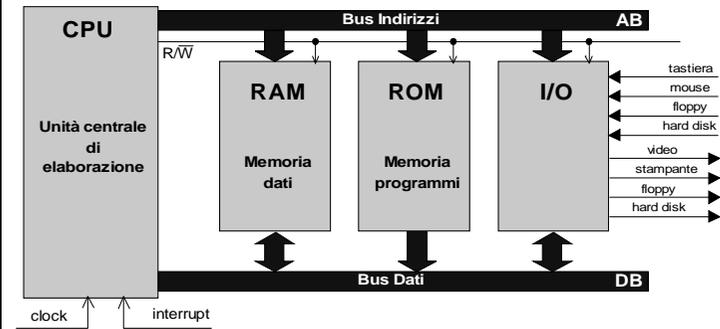
E' la macchina vista da chi la utilizza seguendo schemi software predisposti ai livelli 4 e 3. Il livello di astrazione è **massimo** e le conoscenze d'uso richieste sono **minime**. L'obiettivo è **massimizzare il numero e la capacita' delle funzioni d'uso sicuro disponibili a questo livello.**

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 17

### Architettura Concettuale



Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 19

### 1.2 L'architettura di un Elaboratore

Due modelli originari (~1944):

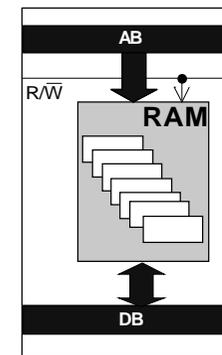
- **computer di von Neumann** (*stored-program computer*), nel quale dati e istruzioni risiedono in una memoria **comune**
  - ✓ istruzioni viste come dati
- **architettura Harvard**, nella quale dati e istruzioni risiedono in memorie **distinte**
  - ✓ più efficiente ed ottimizzabile

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 18

### 1.2.1 Funzioni della RAM



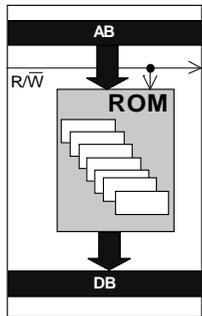
- Memoria ad accesso casuale (=programmato), **non permanente e riscrivibile**
- Destinata a memorizzare dati parziali e programmi temporanei

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 20

### 1.2.2 Funzioni della ROM



- Memoria a sola lettura, **permanente**
- Viene scritta dal produttore o mediante scrittura assai più lenta e costosa della lettura
- Serve per programmi non modificabili e per dati di avviamento

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 21

### 1.2.4 Funzioni dei bus

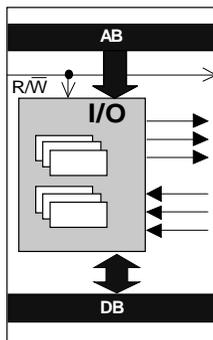
- Trasferiscono le informazioni in parallelo in maniera sincrona
- Di dimensioni legate alla CPU
- Vari standard per i bus di espansione
- Analizzati in sezione 4.7 del corso

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 23

### 1.2.3 Funzioni dell'I/O



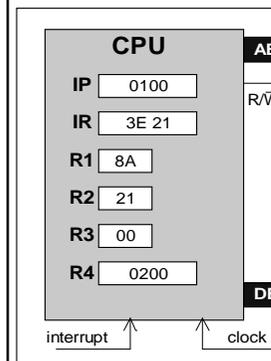
- Gestisce i flussi di ingresso ed uscita
- Deve interfacciare grandezze fisiche e velocità di flusso assai diversificate

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 22

### 1.2.5 Funzioni della CPU



- Dispositivo a comportamento funzionale variabile
- Contiene, tra l'altro, l'equivalente di:
  - ✓ Registro istruzione successiva (Instruction Pointer)
  - ✓ Registri generali (General Purpose)
  - ✓ Registro istruzione corrente (Instruction Register)

Struttura di un elaboratore

Architettura degli elaboratori -1

Pagina 24

### 1.2.5 Funzioni della CPU *(segue)*

La CPU esegue ciclicamente, senza interruzione, le seguenti fasi operative:

- **FETCH**, o *caricamento dell'istruzione*
  1. Emette il contenuto di **IP** nell'**AB**
  2. Legge il codice ritornato dal **DB** e lo pone in **IR**
  3. Incrementa **IP**

### 1.2.5 Funzioni della CPU *(segue)*

- **EXECUTE**, o *esecuzione dell'istruzione*
  1. Decodifica l'istruzione in **IR**
  2. Se necessario, legge i dati richiesti (dalla memoria, dall'I/O, dai registri interni)
  3. Esegue l'operazione codificata dall'istruzione
  4. Se necessario, salva i risultati ottenuti (in memoria, nell'I/O, nei registri interni)