

# Architettura degli elaboratori

Matteo

February 1, 2026

## 1 Introduction

Un calcolatore è un sistema composto da processori, memorie, e dispositivi di input/output. un **Bus** è un insieme di connessioni elettriche parallele che trasportano dati da un componente all'altro.

Nell'architettura di Von Neumann viene introdotta l'idea di usare la memoria non solo per i dati ma anche per il programma.

Collega la cpu e la memoria con un bus indirizzi, indicando la posizione dei dati. Mentre su un altro bus dati memoria e cpu si scambiano i dati veri e propri.

CPU composta da:

- Unità di controllo: legge e interpreta le istruzioni.
- Unità aritmetico logica (ALU): esegue operazioni aritmetiche e logiche.
- registri: memorie veloci interne alla cpu.

ci sono diversi registri specializzati:

- Program counter (PC): indica la prossima istruzione da eseguire.
- Memory address register (MAR): contiene l'indirizzo di memoria da cui leggere o scrivere dati.
- Memory data register (MDR): registro che accede ai bus dei dati.
- Instruction register (IR): contiene l'istruzione che si sta per eseguire.
- Program status word (PSW): contiene informazioni sullo stato della CPU.

in pratica il ciclo di esecuzione di un'istruzione è:

1. il contenuto di PC viene copiato in MAR e inviato alla memoria attraverso il bus indirizzi.
2. la memoria risponde inviando il dato all'MDR attraverso il bus dati.
3. il contenuto di MDR viene copiato in IR e decodificato.
4. l'istruzione passa alla ALU per essere eseguita.
5. se servono altri dati vengono letti dalla memoria nello stesso modo.

6. se serve il risultato e copiato in memoria attraverso MDR
7. il PC viene aggiornato per puntare alla prossima istruzione.

Questo ciclo è chiamato **fetch-decode-execute cycle**.

**data path**: insieme di registri e ALU che eseguono operazioni sui dati.

il percorso dei dati da memoria ad ALU, la loro esecuzione e il ritorno in memoria viene chiamato **ciclo di data path**, ed è governato da un clock.

durata ciclo di data path o ciclo di clock =  $1/F$ , dove F è la frequenza di lavoro della CPU (cicli al secondo), misurata in hertz.

la velocità di esecuzione delle istruzioni ISA (instruction set architecture), è misurata con la durata di un ciclo di clock per i cicli necessari.

fissato il ciclo di clock, la velocità può essere aumentata con il parallelismo.

negli anni 70 si sviluppò la differenza tra CISC e RISC. CISC: complex instruction set computer, set di istruzioni complesso, con istruzioni che fanno operazioni complesse in un singolo ciclo. RISC: reduced instruction set computer, set di istruzioni ridotto, con istruzioni semplici che richiedono più cicli per operazioni complesse.

Risc usa la microprogrammazione, in modo da far fare diverse operazioni a pochi componenti.

più cicli FDE possono essere eseguiti insieme con il **pipelining**

si sviluppano anche architetture che permettono il parallelismo. , ad esempio cpu con più ALU e control units.

oppure molte cpu possono lavorare in modo coordinato sulla stessa istruzione su dati diversi, creando un **array computer**

se invece eseguono istruzioni diverse su dati, con una memoria condivisa, diversi si parla di **multiprocessor systems**.

la forma più complessa sono i **multicomputer systems**, con più cpu, ognuna con memoria propria, collegate in rete.

## 1.1 memoria

vari tipi di memoria

- memoria volatile: perde il contenuto quando viene spenta (es: ram).
- memoria non volatile: mantiene il contenuto quando spenta (es: rom, hard disk).
- memoria online: sempre accessibile
- memoria offline: il supporto deve essere montato

la memoria si organizza in celle, ogni cella è una sequenza di bit con il proprio indirizzo. cella di 8 bit = byte.

molti calcolatori lavorano su blocchi da 32 o 64 bit, questi blocchi si chiamano word.

le word possono essere memorizzate in celle standard, occupando più celle consecutive. si può fare in 2 modi:

- big-endian: byte più significativo all'indirizzo più basso.
- little-endian: byte meno significativo all'indirizzo più basso.

la **cache** è una memoria poco capiente ma veloce, che viene usata per contenere le word più usate.

la cpu prima cerca i dati nella cache, se non li trova li va a prendere in memoria principale.

**principio di località:** dati usati recentemente gli uni dagli altri sono spesso in locazioni vicine.

tipi di supporti:

- hard disk: memoria non volatile, lenta, grande capacità, usa piatti magnetici rotanti. (può essere resa più veloce con la tecnica RAID, che usa più dischi in parallelo).
- SSD: memoria non volatile, veloce, grande capacità, usa memoria flash, più veloce ma meno capiente.
- CD e DVD: memoria non volatile, usano supporti ottici, capacità limitata.

## 2 porte logiche e circuiti combinatori

Porte logiche sono basate sull'algebra di Boole, un'espressione booleana si costruisce con: le costanti di boole(0,1), gli operatori booleani, e variabili(x,y, ecc.).

**mintermine:** un prodotto (AND) di tutte le variabili della funzione, ognuna presente una sola volta, negata o non negata, che vale 1 per una sola combinazione di valori delle variabili.

**forma canonica:** l'OR di tutti mintermini veri

porta nand è una porta logica utile per costruire circuiti combinatori, falsa solo quando  $a=1$  e  $b=1$ . è universale e facile da costruire con transistor

array logici programmabili: collegando dei fusibili è possibile creare porte logiche personalizzate.

### 2.1 mappe di karnaugh

**mappe di karnaugh:** modo di rappresentare funzioni booleane. permettono di costruire un circuito combinatorio minimale.

tabella bidimensionale in cui ogni mintermine ha una cella, e la cella ha un valore, 1 se vale, se 0 se no.

puoi creare raggruppamenti che comprendono gruppi di certi letterali =1.

all'interno di una mappa di karnaugh una copertura si dice minimale quando:

- i suoi raggruppamenti non è contenuto in raggruppamenti più grandi.
- i suoi raggruppamenti contengono almeno una cella che non appare in altri raggruppamenti della copertura

### 2.2 bus multi bit

circuiti combinatori collegati a connessioni da più bit sono detti **bus**

sono praticamente un raggruppamento di segnali e ogni coppia lavora separatamente come in una porta normale, per poi far uscire i segnali uno dopo l'altro.

### **3   codice binario**

codifiche posizionali principali:

- binario
- ottale
- esadecimale