03. Architettura dei Calcolatori

Corso di Informatica

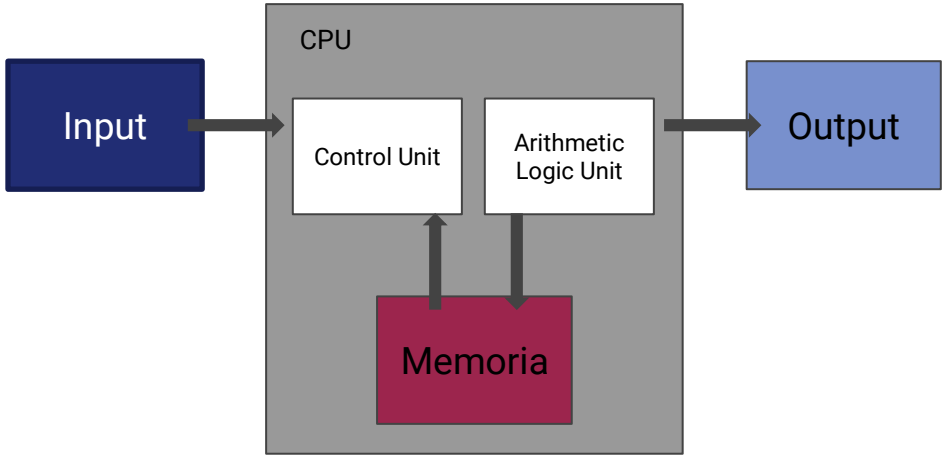
Corso di Laurea in Matematica (D.M. 270/04) - A.A. 2020/2021
Angelo Cardellicchio
angelo.cardellicchio@uniba.it
Ultimo aggiornamento: 26/10/2020

Outline

- La macchina di von Neumann
  - Componenti fondamentali
  - Esecuzione dei programmi
  - Il collo di bottiglia di von Neumann
- L’architettura Harvard
- Von Neumann vs Harvard
- L’architettura Harvard modificata

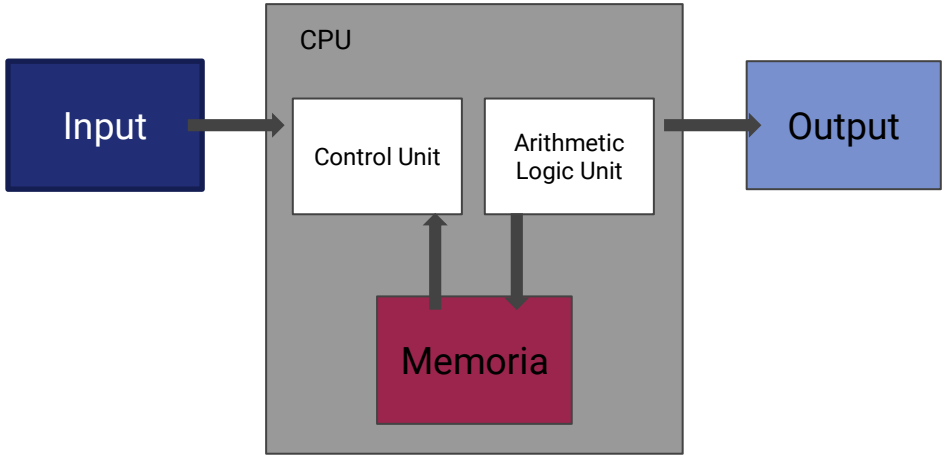
La macchina di von Neumann

- Introdotta per superare i limiti delle architetture non programmabili
- Definisce il paradigma di *stored-program computer*
- Rappresenta la base delle architetture moderne



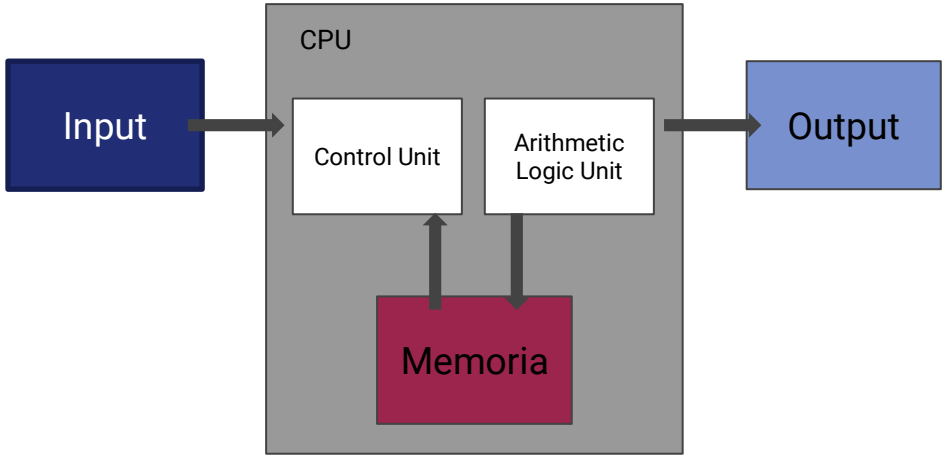
Componenti fondamentali

- ***Central Processing Unit***
  - ***Control Unit***
    - ***Instruction Register***
    - ***Program Counter***
  - ***Arithmetic Logic Unit***
- ***Memoria***
- ***Input***
- ***Output***
- ***Bus***



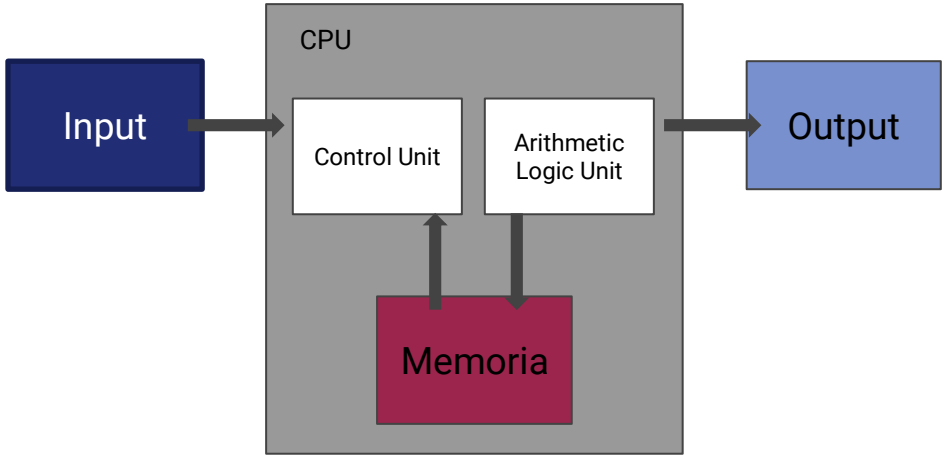
Esecuzione dei programmi (1)

1. *Il programma è diviso in **dati** ed **istruzioni***
2. *Il programma viene caricato sulla memoria*
3. *L'esecuzione avviene seguendo il ciclo:*
   a. *FETCH: estrazione dell'istruzione*
   b. *DECODE: interpretazione dell'istruzione*
   c. *EXECUTE: esecuzione dell'istruzione*



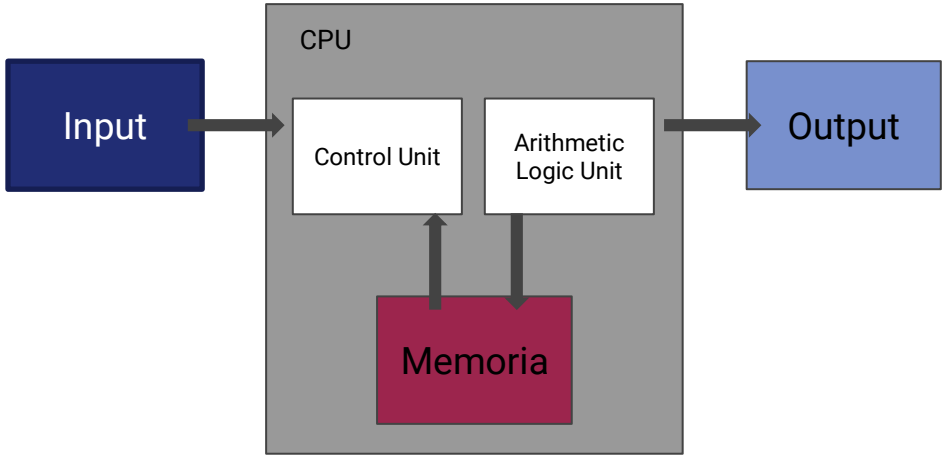
Esecuzione dei programmi (2)

1. ***FETCH***
    a. *Il contenuto cui punta il PC viene caricato nel CIR.*
2. ***DECODE***
    a. *L'istruzione viene decodificata dalla CU.*
    b. *Il PC viene aggiornato per puntare all'istruzione successiva.*
3. ***EXECUTE***
    a. *La CU esegue l'istruzione.*
    b. *Se è necessario manipolare uno dei registri dell'ALU, questa viene interpellata.*



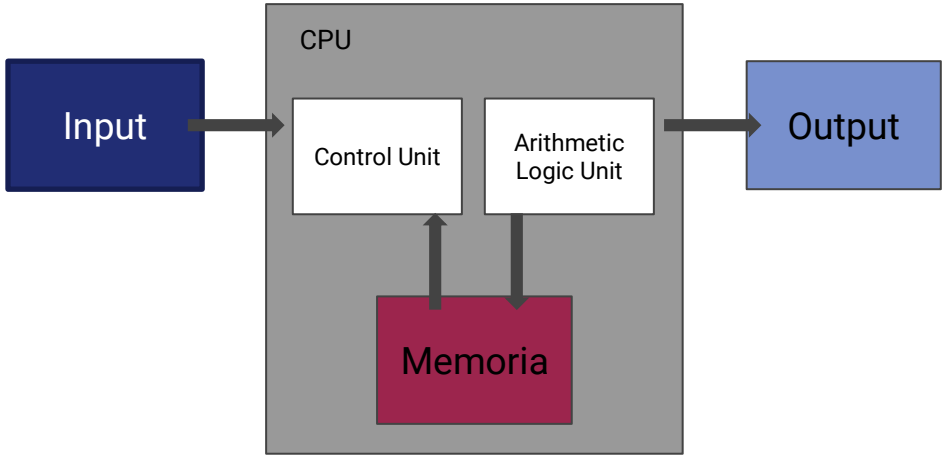
Il collo di bottiglia di von Neumann (1)

- Nel tempo, le CPU sono diventate molto più veloci...
- ...mentre le memorie molto più dense!
- Il bus per dati ed istruzioni è condiviso, e spesso *congestionato*.
- Il tempo di elaborazione di un'istruzione è spesso *inferiore* a quello di estrazione della stessa dalla memoria.



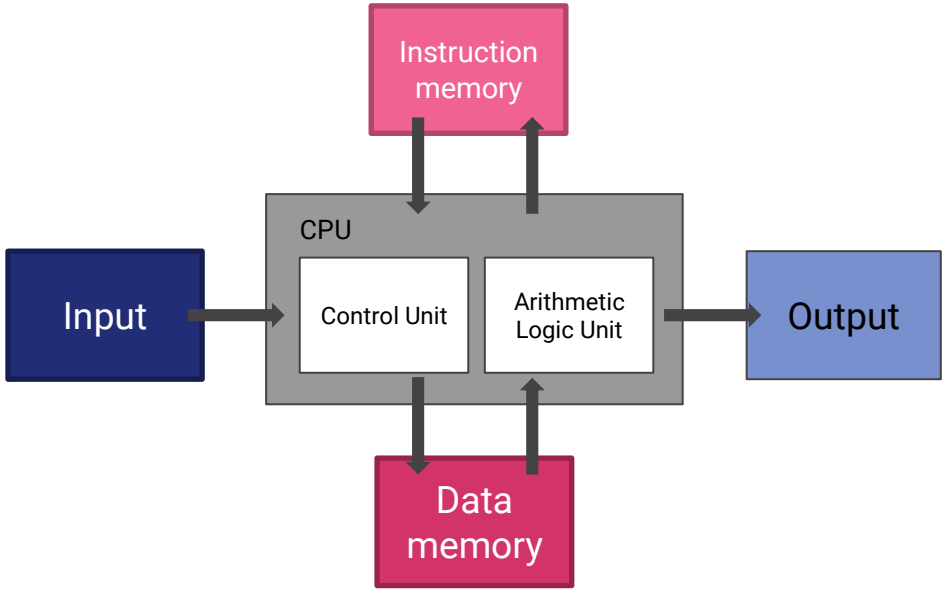
Il collo di bottiglia di von Neumann (2)

- Per mitigare questo fenomeno, si sono adottati diversi stratagemmi:
  - utilizzo di memorie *cache*, di dimensioni ridotte ma molto veloci;
  - algoritmi di *branch prediction*;
  - etc.
- La soluzione vera e propria è però arrivata grazie all'*architettura Harvard*.



L’architettura Harvard

- L’architettura Harvard è nata per sopperire ai limiti dell’architettura di von Neumann
- Prevede due aree distinte di memoria:
  - una *data memory*...
  - ...ed una *instruction memory*.
- Ognuna delle due aree ha un proprio bus.
- ***Domanda: come influisce la presenza di due aree di memoria sul FETCH?***
  - ***Risposta: la CPU può contemporaneamente estrarre un’istruzione e leggere/scrivere un dato.***



Von Neumann vs Harvard

Architettura di Von Neumann	Architettura Harvard
Utilizza lo stesso percorso fisico per accedere a dati ed istruzioni.	Ha percorsi fisici differenti per l'accesso a dati ed istruzioni.
Usa lo stesso bus per le leggere/scrivere dati ed estrarre istruzioni.	Ha un insieme dedicato di indirizzi e bus per leggere/scrivere dati, ed un altro insieme di indirizzi e dati per estrarre istruzioni.
La CPU non può contemporaneamente leggere/scrivere un dato ed estrarre un'istruzione.	La CPU può contemporaneamente leggere/scrivere un dato ed estrarre un'istruzione.
Gli indirizzi a disposizione di dati ed istruzioni sono gli stessi.	Dati ed istruzioni hanno indirizzi separati.

L’architettura Harvard modificata

- Le CPU moderne sono basate su *architetture Harvard modificate*.
- Le architetture Harvard:
  - *sono maggiormente performanti delle architetture di von Neumann…*
  - *…ma sono anche più complesse da gestire e realizzare!*
  - *Sono quindi usate prevalentemente da hardware specializzato.*
- Le architetture di von Neumann:
  - *sono meno performanti, ma più semplici da realizzare;*
  - *possono essere meglio adattate a scopi generici.*
- Soluzione: ***architetture di Harvard modificate***.
  - *Dal punto di vista del programmatore, sono architetture di von Neumann.*
  - *Dal punto di vista del funzionamento interno, sono architetture Harvard.*

L’architettura Harvard modificata

- Le architetture Harvard modificate sono alla base della maggior parte delle CPU moderne.
- Le modifiche si concentrano su un *rilassamento* della suddivisione tra data memory ed instruction memory.
- ***Split-cache***
  - *Implementa delle memorie cache separate per dati ed istruzioni.*
- ***Access-Instruction-Memory-as-Data***
  - *Accesso diretto a determinate parti della instruction memory (es. costanti, funzioni) senza copiarle nella data memory, migliorando performance ed efficienza.*
- ***Read-Instructions-from-Data-Memory***
  - *Permette l’esecuzione di codice direttamente dalla data memory.*

Domande?

42