

# Architettura del calcolatore

## Calcolatori (1)

- Introdotti all'inizio degli anni 1940
- Costruiti assemblando **componenti elettronici elementari** per memorizzare informazioni ed eseguire programmi
- **informazioni** manipolate per ottenere i risultati desiderati
- informazioni inserite tramite sequenze di cifre **0,1**
  - 0: assenza di **tensione elettrica**
  - 1: presenza di tensione elettrica
- Si parla di **informazione digitale**

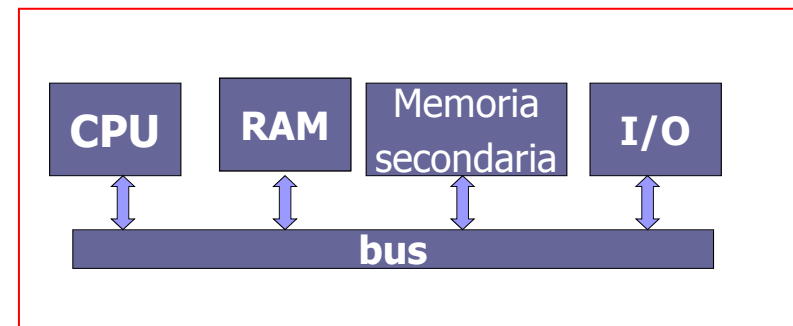
2

## Calcolatori (2)

- Programmi e dati dei programmi come sequenze di cifre binarie → lavoro arduo per l'utente
- **Linguaggi di programmazione** per sollevare l'utente dalla scrittura dei programmi in binario
- L'architettura di base del calcolatore e' rimasta fondamentale la stessa: si tratta della cosiddetta **architettura di Von Neumann**

3

## L'architettura di Von Neumann



4

## RAM = Random Access Memory (memoria ad accesso casuale)

Nella RAM, come in ogni altra componente di un computer, le **informazioni** sono sempre rappresentate **digitalmente** tramite **sequenze di 0 e di 1**.

La RAM quindi memorizza numeri binari:

- un **bit** (=binary digit) può contenere o **0** o **1**
- un **byte** è una **sequenza di 8 bit** (es. 11001010)
- una **parola** è una **sequenza di 4 byte** = 32 bit

5

## Rappresentare gli interi in cifre binarie

- **Intero** → **binario** = in base 2
- Esempio  $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0$ 

$$= 1 \ 1 \ 0 \ 0 \ 1$$

$$\uparrow \ \uparrow \ \uparrow \ \uparrow \ \uparrow$$

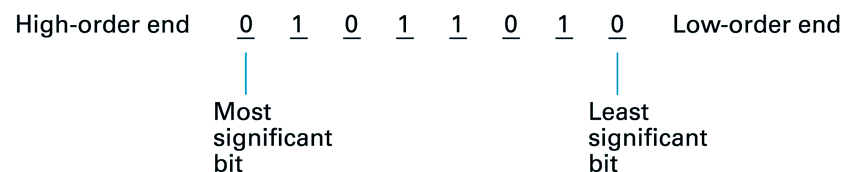
$$2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$$

- **Binario** → **Intero**
- Esempio:  $1 \ 1 \ 0 \ 1 = 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 = 8 + 4 + 1 = 13$

6

## Memoria principale

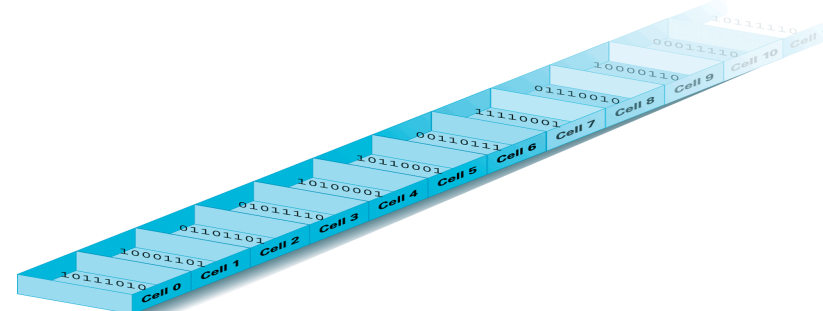
- Celle di memoria: di solito 8 bit (1 byte)



7

## Indirizzi di memoria

- Per identificare ogni cella
- Parte da 0
- Segue secondo la numerazione binaria



8

# Numerazione binaria degli indirizzi

Indirizzo parola	Indirizzo byte		
	Indirizzo decimale	Indirizzo binario	
0	0	000000 00	
	1	000000 01	
	2	000000 10	
	3	000000 11	
1	4	000001 00	
	5	000001 01	
	6	000001 10	
	7	000001 11	
2	8	000010 00	
	9	000010 01	
	10	000010 10	
	11	000010 11	

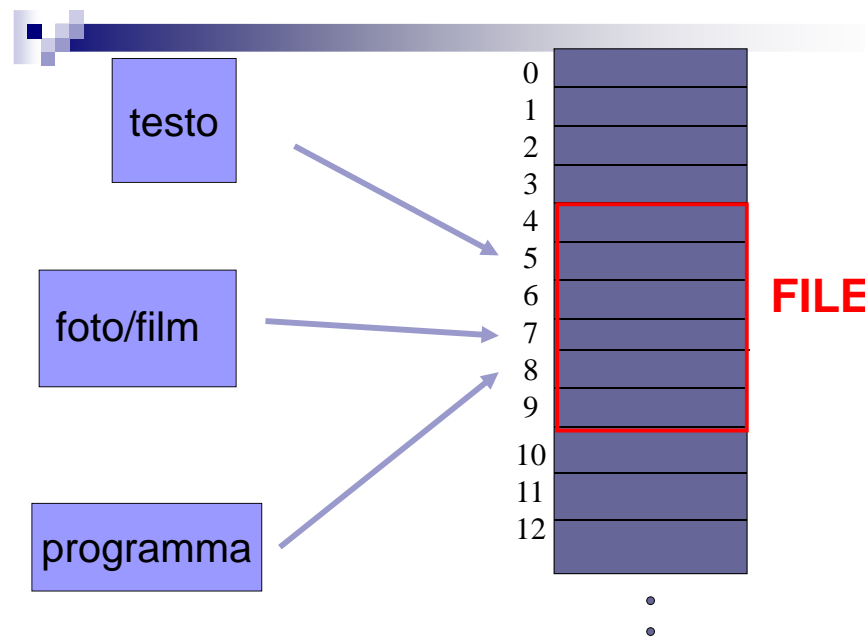
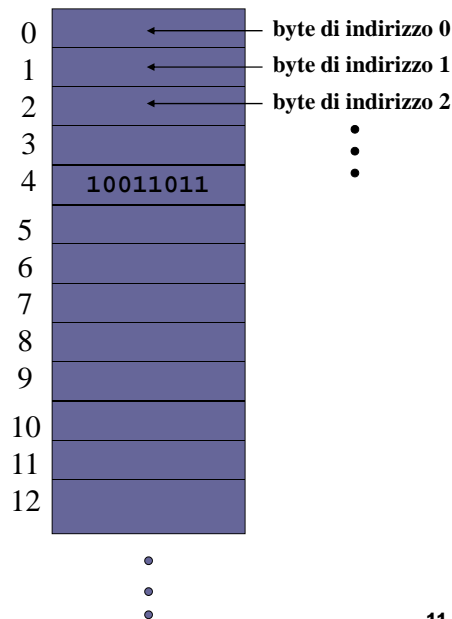
# Accesso alla memoria

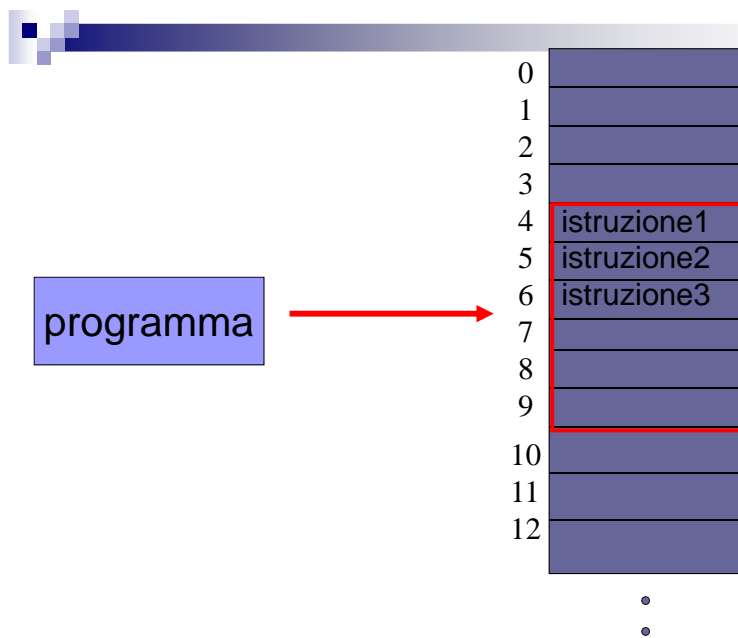
- **Accesso diretto** ad ogni cella, senza dover iniziare dalla cella 0
- **Accesso a piccole unita'**: 8 bit (a differenza delle memorie secondarie)
- Tipi di accesso: **lettura e scrittura**

Quindi la **RAM** puo' **concettualmente** essere vista come una **sequenza di byte**

**indirizzi**

da cui si **leggono** ed in cui si **scrivono** **blocchi di byte consecutivi**





13

## Unità di misura della RAM (e della memoria in generale)

1 **KiloByte** (KB) =  $2^{10}$  byte = 1.024 byte;  
circa 1.000 byte

1 **MegaByte** (MB) =  $2^{20}$  byte = 1.024 KB;  
circa 1.000.000 byte

1 **GigaByte** (GB) =  $2^{30}$  byte = 1.024 MB;  
circa 1.000.000.000 byte

1 **TeraByte** (TB) =  $2^{40}$  byte = 1.024 GB;  
circa 1.000.000.000.000 byte

14

## Proprietà della RAM

- RAM => **accedere ad ogni byte ha la stessa durata** ( $10^{-7}$  sec): non dipende da quale byte è stato acceduto prima
- è **volatile**: se tolgo la spina l'informazione è persa (c'è anche la **ROM**)
- ogni byte ha un **indirizzo** 0,1,2,.....
- il **byte** è la minima quantità accessibile (attraverso il suo indirizzo)

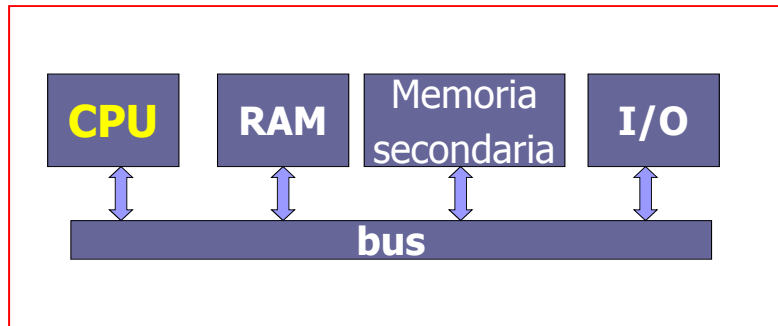
15

## Vari tipi di memoria ...

- CACHE** (magazzino temporaneo) è una sottoparte della RAM disponibile al processore per un **accesso più veloce**, ha **dimensione minori** della RAM
- ROM (Read Only Memory)**: una parte di memoria su cui si possono fare solo accessi e non scritture. **Contiene le istruzioni dei programmi** di bootstrap dei sistemi operativi che permettono di avviarli ad ogni **accensione**. Viene mantenuta da una piccola batteria interna.

16

## L'architettura di Von Neumann



17

## CPU

- La CPU (Central Processing Unit) è in grado di **eseguire dei programmi**, cioè sequenze di istruzioni elementari
- Idea fondamentale dell'architettura di Von Neumann: **programmi e dati** risiedono entrambi in memoria RAM
- Per poter essere eseguiti i programmi devono risiedere nella RAM, e quindi sono **codificati digitalmente**

18

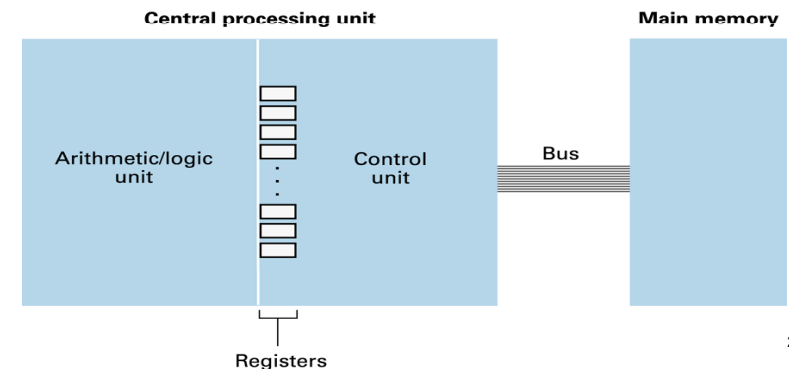
## Elementi della CPU

- **Central Processing Unit**, processore
  - **Unità aritmetico-logica**: elaborazione dati
  - **Unità di controllo**: coordina le attività
  - **Registri**: memoria temporanea, simili a celle di memoria principale
    - **Generici**: per gli operandi di un'operazione logica/aritmetica, e il risultato
    - **Speciali**: per operazioni particolari

19

## CPU e memoria principale

- Trasferimento dati in entrambe le direzioni (lettura e scrittura): bus



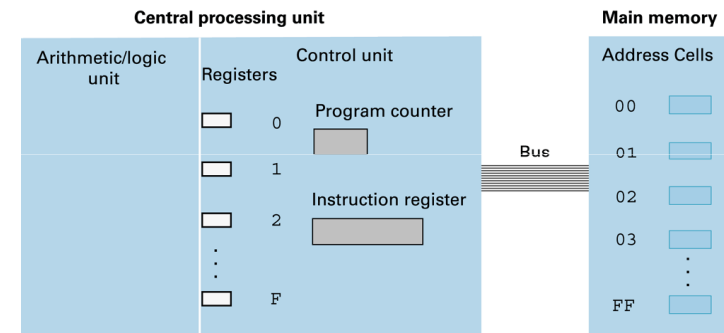
20

## ALU e Registri della CPU

- L'**ALU** e' l'**unità aritmetico-logica** (ALU e' un acronimo dall'inglese) che **esegue** le istruzioni e **gestisce** i registri della CPU
- I **registri** servono per **memorizzare** gli operandi per le istruzioni di calcolo dell'ALU
- **Registri particolari**
  - **PC (program counter)**: contiene l'**indirizzo RAM della prossima istruzione** da eseguire
  - **IR (instruction register)**: contiene l'**istruzione** da eseguire

21

## Esempio di architettura



- 16 registri, 256 celle di memoria
- **Program counter**: indirizzo della prossima istruzione da eseguire
- **Instruction register**: istruzione da eseguire

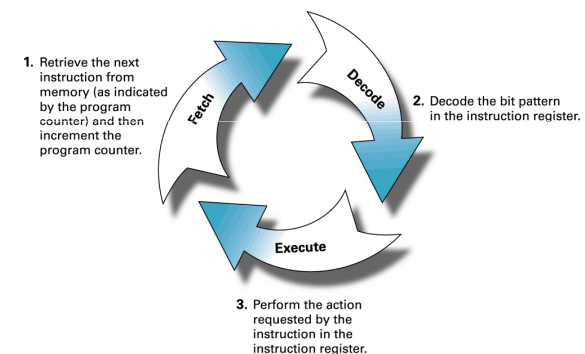
22

## Ciclo FDE della CPU ( $10^{-9}$ sec)

- **Reperimento dell'istruzione (fetch)**:
  - **lettura della cella** di RAM il cui indirizzo e' contenuto nel **contatore** di programma
  - **caricamento del registro istruzione** con l'istruzione
  - **Incremento del contatore** programma
- **Decodifica dell'istruzione (decode)**:
  - **Trova gli operandi** a seconda del codice operativo
  - **Modifica contatore** programma se istruzione di salto
- **Esecuzione dell'istruzione (execute)**:
  - **Attiva i circuiti** necessari

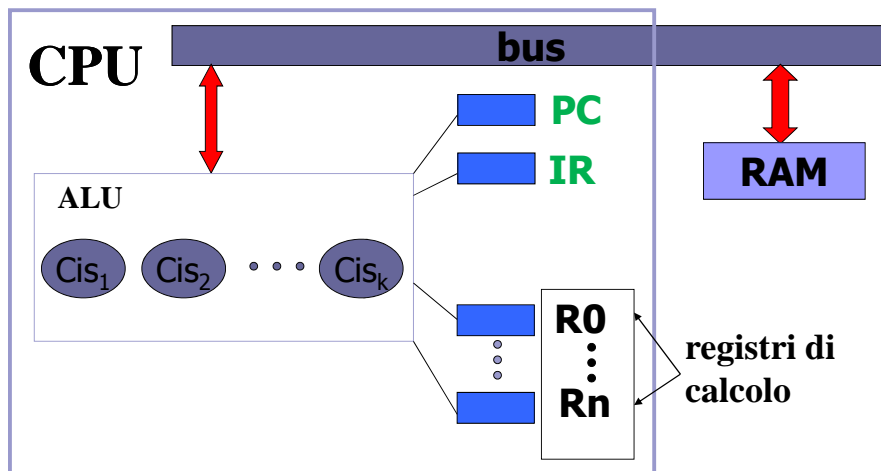
23

## Ciclo della CPU



24

## Modello concettuale della CPU



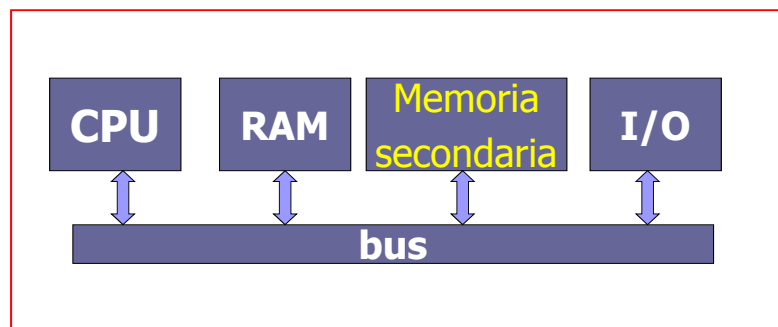
25

## Esecuzione delle seguenti istruzioni

1. **Carica** nel **registro R3** il contenuto della cella di memoria 10
2. **Carica** nel **registro R4** il contenuto della cella di memoria 11
3. **Somma di interi** sul contenuto dei registri R3 e R4, **risultato** nel **registro R0**
4. **Trasferisci** il contenuto del **registro R0** nella cella 12
5. **STOP**

26

## L'architettura di Von Neumann



27

## Memoria principale e secondaria

- **Volatilità'** della **memoria principale** (senza tensione perde il suo contenuto) e **dimensione limitata**
- → **memoria secondaria** (dischi magnetici e CD)
  - permanente**
  - contiene tutto quello che si vuole salvare **anche dopo lo spegnimento**
  - memoria **sequenziale** (il tempo varia a seconda dell'accesso precedente)
  - adatta** per **leggere/scrivere** grandi quantità di dati (in **posizioni contigue**)

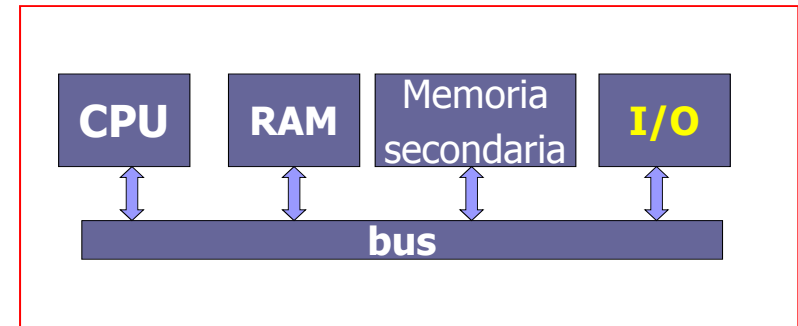
28

## Dati e tipi di memoria

- **Registri e cache** ← dati **in elaborazione**
- **RAM** ← dati elaborati in un **futuro immediato**
- **ROM** (Read Only Memory) ← dati **permanenti**
- **CD e DVD** ← dati di **utilizzo non immediato**

29

## L'architettura di Von Neumann



30

## Dispositivi di I/O (Input/Output)

Sono i dispositivi di comunicazione ed **interazione** tra **utente** e **computer**.

In un moderno PC:

- **input:** tastiera, mouse, touchpad, microfono, videocamera, scanner, connessione di rete, etc
- **output:** video, stampanti, audio, etc
- **velocità diverse** e molto maggiori delle altre componenti di un computer
  - **sec** per l'**input**
  - **decimi di sec** per l'**output**

31