

Architettura - termine per descrivere l'insieme degli attributi di un sistema così come appare al programmatore

Architettura dei calcolatori - scienza di selezionare e collegare componenti hardware per creare calcolatori

COMPUTER ARCHITECTURE

- **Instruction set architecture (ISA)** - come l'hardware viene visto dal software

Categorie di istruzioni:

- **load** (caricamento da memoria a registri) / **store** (memorizzazione da registri a memoria) - trasferimento dati da registri (sono una memoria secondaria + veloce ma + costosa) a memoria (è la memoria principale + lenta e - costosa)
- **Calcolo** - funzioni di base usate per costruire tutte le altre
- **jump** (salto) / **branch** (ramificazione) - realizzare programmi iterativi e procedure
- **istruzioni speciali**
- **machine organization** - descrive le funzionalità dell'implementazione fisica in termini di blocchi funzionali e logici

Esempio di una famosa architettura moderna open source

RISC-V - (= risc five) - basata sul concetto RISC (reduced instruction set computer) con poche istruzioni, semplici e veloci

Fasi per eseguire un programma RISC-V:

- Instruction fetch - la CPU prende dalla RAM l'istruzione in memoria e la copia nel registro dell'istruzione attuale
- Instruction decode e register fetch - la CPU decodifica l'istruzione, capisce che tipo di operazione deve fare e legge i registri necessari
- Execute - la CPU esegue l'operazione usando l'unità aritmetico logica
- Memory access - se l'istruzione deve leggere o scrivere nella RAM lo fa in questa fase
- Write back - il risultato finale viene scritto nel registro di destinazione e sarà disponibile per le istruzioni successive

CHIP – composto da materiale

semiconduttore con circuiti elettronici integrati – contiene milioni di componenti elettronici microscopici chiamati transistor che trasmettono segnali di dati

- **Di memoria** – conservano dati e programmi su computer e dispositivi di archiviazione
- **logici** – elaborano i dati per completare le attività assegnate
- **ASIC** - (=application specific integrated chips) - eseguire routine ripetute di elaborazione per un'applicazione specifica
- **SoC** - (=system on a chip) - tutti i componenti elettronici dell'intero sistema sono integrati in un singolo chip

CLASSIFICAZIONE DEI CIRCUITI

- **circuiti integrati analogici** – trasmettono segnali continui e variabili lungo un periodo di tempo prestabilito – il segnale di uscita è una funzione lineare dell'input con voltaggi direttamente proporzionali
- **circuiti integrati digitali** – trasmettono segnali discontinui o binari – il voltaggio di uscita può essere alto (valore booleano 1) o basso (valore booleano 0)
- **circuiti integrati a segnale misto** – chip misti che combinano gli elementi dei chip analogici con quelli dei digitali – permette la realizzazione di chip che possono operare come convertitori digitale-analogico e viceversa
- **circuiti quantistici** – routine di calcolo che definisce una serie di operazioni quantistiche logiche

STRUTTURA FISICA DEI CALCOLATORI

Principali tipi:

- **embedded** – microprocessori nelle automobili, elettrodomestici e wearables
- **pc desktop** – singolo utente
- **server** – grossi carichi di lavoro e affidabilità - realizzati con le stesse tecnologie base dei personal computer ma offrono una potenza di calcolo maggiore e una quantità di memoria maggiore
- **supercomputer** – migliaia di processori con terabyte di memoria - può usare tutto l'hardware per eseguire un singolo programma parallelo

ORGANIZZAZIONE DI UN CALCOLATORE

- **CPU** (= unità di elaborazione centrale) - cervello del computer - recupero, decodifica e esecuzione delle istruzioni del programma
- **RAM** (=memoria ad accesso casuale) - vengono caricati programmi e dati attivi in modo che ogni volta che il processore li richiede non deve recuperarli dal disco rigido - è volatile e perde il suo contenuto una volta spenta l'alimentazione
- **BIOS** (=basic input/output system) - memoria di sola lettura - è il collegamento tra l'hardware del computer e il software in un sistema - controlla la tastiera, lo schermo, le unità disco...
- **Batteria CMOS** - memorizzare informazioni di base sulla configurazione del PC
- **memoria cache** - piccolo blocco di memoria ad alta velocità che migliora le prestazioni del PC precaricando le informazioni dalla memoria principale e trasmettendole al processore su richiesta
- **chipset** - gruppo di piccoli circuiti che coordinano il flusso di dati da e verso i componenti di un PC
 - NorthBridge (controller di memoria)
 - SouthBridge (controller di input/output)
- **Bus di espansione** - percorso di ingresso/uscita dalla CPU ai dispositivi periferici
- **clock della CPU** - sincronizza il funzionamento di tutte le parti del PC

Macchina di calcolo

sistema capace di eseguire una procedura di calcolo - espressa tramite una sequenza di passi e istruzioni di calcolo comprensibili dall'essere umano

- **ALGORITMO** - sequenza di passi che rappresentano la soluzione ad un problema
- **DATI** - informazioni che la macchina deve saper gestire per eseguire un calcolo - usate codifiche per gestire i dati - es. codifica binaria

Codifica binaria

la più utilizzata - rappresenta la differenza di tensione di una componente elettronica

- **BIT**
- **BYTE** = 8 bit

Codifica binaria dei numeri naturali - linguaggio per rappresentare i dati - sequenza di 0 e 1

- 0 - 00000000
- 1 - 00000001
- 2 - 00000010
- ... 255 - 11111111

$f(x) = x^2$ $f(10) = 10^2 = 100$ (in decimale)

$f(00001010) = 01100100$ (in binario)

Macchina di turing

(matematico e padre informatica) - 1936 - modello ideale calcolatore

- **MEMORIA** - per mettere dati
- **ISTRUZIONI** - per fare calcoli e scrivere programma
- **MECCANISMO** - per eseguire calcoli e istruzioni - un solo tipo di istruzioni con varianti a seconda di come si opera sul nastro

MODELLO MEMORIA - nastro infinito diviso in celle - ognuna contiene un bit (0 / 1 / vuota) - testina che scorre le celle, legge input, produce output

HARDWARE

- Nastro
- Testina - ha uno stato (colore)
- Istruzioni - permettono alla testina di cambiare stato

COMPUTAZIONE - configurazione iniziale + esecuzione delle istruzioni applicabili nello stato attuale fino a quando possibile (non più possibile quando non ho istruzioni per la cella corrente)

racpresentazione binaria dati interi

RAPPRESENTAZIONE BINARIA DEI DATI

bit = unita minima di rappresentazione

L puo assumere due valori $\leftarrow \begin{matrix} 0 - \text{vero} \\ 1 - \text{falso} \end{matrix}$

con n bit possiamo rappresentare 2^n combinazioni diverse

byte = sequenze di bit

1 byte = 8 bit

RAPPRESENTARE NUMERI INTERI POSITIVI

numeri da binario a decimale

10100101 \rightarrow posizioni

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 128 + 32 + 4 + 1 = 165$$

$$\frac{10100101}{\text{(binario)}} = \frac{165}{\text{(decimale)}}$$

numeri da decimale a binario

- calcolare divisione intera di N per 2 (fino a quando il risultato della divisione e uguale a 0)

$$\begin{array}{r} 13 \overline{) 26} \\ \underline{2} \\ 0 \\ \underline{0} \\ 0 \\ \underline{0} \\ 0 \\ \underline{0} \\ 0 \end{array}$$

$$13 = 1101$$

RAPPRESENTARE NUMERI INTERI NEGATIVI

metodo modulo e segno

il bit piu a sinistra indica il segno $\leftarrow \begin{matrix} 0 = \text{positivo} \\ 1 = \text{negativo} \end{matrix}$

gli altri bit rappresentano il valore assoluto (senza segno)

esempio $\begin{matrix} 0111 & 2^2 + 2^1 + 2^0 = 7 \rightarrow +7 \\ 1111 & \rightarrow -7 \end{matrix}$

metodo complemento a 1

invertire tutti i bit del numero positivo

esempio $\begin{matrix} 0101 & 2^2 + 2^0 = 5 \\ 1010 & \rightarrow -5 \end{matrix}$

metodo complemento a 2

fare il complemento a 1 e aggiungere 1 al risultato

esempio $\begin{matrix} 00011011 & 2^4 + 2^3 + 2^1 + 2^0 = 27 \\ 11100100 & \\ \hline 11100101 & \rightarrow -27 \end{matrix}$

codice binario

martedì 7 ottobre 2025 12:37

tradurre da decimale a binario

$$\begin{array}{r}
 15 \mid 2 \\
 \hline
 1 \mid 7 \mid 2 \\
 \hline
 1 \mid 3 \mid 2 \\
 \hline
 1 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 15 = 1111$$

$$\begin{array}{r}
 21 \mid 2 \\
 \hline
 1 \mid 10 \mid 2 \\
 \hline
 0 \mid 5 \mid 2 \\
 \hline
 1 \mid 2 \mid 2 \\
 \hline
 0 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 21 = 10101$$

$$\begin{array}{r}
 20 \mid 2 \\
 \hline
 0 \mid 10 \mid 2 \\
 \hline
 0 \mid 5 \mid 2 \\
 \hline
 1 \mid 2 \mid 2 \\
 \hline
 0 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 20 = 10100$$

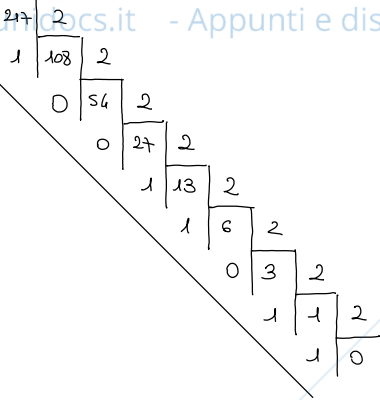
$$\begin{array}{r}
 189 \mid 2 \\
 \hline
 1 \mid 94 \mid 2 \\
 \hline
 0 \mid 47 \mid 2 \\
 \hline
 1 \mid 23 \mid 2 \\
 \hline
 1 \mid 11 \mid 2 \\
 \hline
 1 \mid 5 \mid 2 \\
 \hline
 1 \mid 2 \mid 2 \\
 \hline
 0 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 189 = 10111101$$

$$\begin{array}{r}
 130 \mid 2 \\
 \hline
 0 \mid 65 \mid 2 \\
 \hline
 1 \mid 32 \mid 2 \\
 \hline
 0 \mid 16 \mid 2 \\
 \hline
 0 \mid 8 \mid 2 \\
 \hline
 0 \mid 4 \mid 2 \\
 \hline
 0 \mid 2 \mid 2 \\
 \hline
 0 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 130 = 10000010$$

$$\begin{array}{r}
 1024 \mid 2 \\
 \hline
 0 \mid 512 \mid 2 \\
 \hline
 0 \mid 256 \mid 2 \\
 \hline
 0 \mid 128 \mid 2 \\
 \hline
 0 \mid 64 \mid 2 \\
 \hline
 0 \mid 32 \mid 2 \\
 \hline
 0 \mid 16 \mid 2 \\
 \hline
 0 \mid 8 \mid 2 \\
 \hline
 0 \mid 4 \mid 2 \\
 \hline
 0 \mid 2 \mid 2 \\
 \hline
 0 \mid 1 \mid 2 \\
 \hline
 1 \mid 0
 \end{array}
 \quad 1024 = 1000000000$$

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari



tradurre da decimale a binario

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array} = 2^3 + 2^2 + 2^1 + 2^0 = 8 + 4 + 2 + 1 = 15$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} = 2^6 + 2^1 = 64 + 2 = 66$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 8 + 2 = 90$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} = 2^6 + 2^1 + 2^0 = 64 + 2 + 1 = 67$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} = 2^1 = 2$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{array} = 2^6 + 2^4 + 2^1 + 2^0 = 64 + 16 + 2 + 1 = 83$$

$$\begin{array}{cccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} = 2^6 + 2^4 + 2^1 = 64 + 16 + 2 = 82$$

$$\begin{array}{ccccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

$$\begin{array}{ccccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array} = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 128 + 64 + 32 + 16 + 8 + 4 + 2 = 254$$

$$\begin{array}{ccccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array} = 2^6 + 2^5 + 2^3 + 2^2 + 2^1 = 64 + 32 + 8 + 4 + 2 = 110$$

$$\begin{array}{cccccccccccc} 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} = 2^{31} + 2^{23} + 2^{15} + 2^7$$

$$= 2 \cdot 167.683.648 + 8 \cdot 388.608 + 32 \cdot 768 + 128$$

$$= 2 \cdot 155.905.152$$

ARITMETICA DEI CALCOLATORI

SOMMA IN COMPLEMENTO A 2

si sommano bit a bit
 ↳ se c'è un riporto oltre il numero di bit disponibili si ignora

esempio

$$\begin{array}{r} 0101 + (+5) \\ 0010 = (+2) \\ 0111 = (+7) \end{array}$$

SOTTRAZIONE IN COMPLEMENTO A 2

per calcolare $A - B$ fare $A + (-B)$
 ↳ complemento a 2 di B
 oppure normale sottrazione

esempio

$$\begin{array}{l} A = 29 = 00011101 \\ B = 14 = 00001110 \end{array}$$

↳ $1+1=2 \rightarrow$ metto 0 e riporto 1

$$\begin{array}{r} (A) 00011101 - \\ (B) 00001110 = \\ \hline 00001111 \end{array}$$

se sottraggo 1 da 0 devo chiedere in prestito dal numero prima
 ↳ 1 vale 2 nella posizione inferiore

$$\begin{array}{r} 11110001 \\ 11110001 + \\ \hline 11110010 \quad (-B) \\ 00011101 + \\ \hline 11110010 = \\ 00001111 \quad (A + (-B)) \end{array}$$

MOLTIPLICAZIONE

mettere la cifra del moltiplicando se il moltiplicato è 1, altrimenti mettere 0

esempio

$$\begin{array}{r} 1000 \times \\ 1001 = \\ \hline 0000 \\ 0000 \\ 1000 \\ \hline 1001000 \end{array}$$

OVERFLOW

quando il risultato di un'operazione non può essere rappresentato con l'hardware a disposizione

	A	B	overflow
A + B	≥ 0	≥ 0	< 0
A + B	< 0	< 0	≥ 0

Somma → overflow quando i segni di A e B sono discordanti dal segno del risultato

	A	B	overflow
A - B	≥ 0	< 0	< 0
A - B	< 0	≥ 0	≥ 0

Sottrazione → overflow quando il segno di A è discordante dal segno del risultato

RAPPRESENTAZIONE BINARIA DEI CARATTERI

- ASCII Standard → 1 carattere rappresentato con 7 bit → caratteri ASCII
- ASCII estesa → 1 carattere rappresentato con 8 bit → caratteri speciali come è, é, à, ...
- UNICODE → 1 carattere rappresentato con 16 bit → altri alfabeti come arabo, cinese ...
↳ attualmente 21 bit

RAPPRESENTAZIONE BINARIA DEI SUONI

↳ attraverso la frequenza di campionamento e di quantizzazione

- FREQUENZA DI CAMPIONAMENTO → numero di campioni acquisiti nell'unità di tempo
- FREQUENZA DI QUANTIZZAZIONE → numero di bit necessari per codificare ogni valore

mondo reale → digitalizzazione → elaborazione → memorizzazione
calcolatore digitale → conversione digitale - analogica

formati audio

- WAVE → .wav → dimensioni elevate
- MP3 → .mp3 → grande diffusione su internet
- MIDI → .mid → non memorizzati suoni ma comandi

RAPPRESENTAZIONE BINARIA DELLE IMMAGINI (BITMAP)

1. dividere l'immagine in una griglia con righe orizzontali e verticali → campionamento
2. ogni quadratino è un pixel, con valore 0 se è bianco e 1 se è nero → quantizzazione



risoluzione = altezza · larghezza (in pixel)

informazione sul colore → 3 byte → uno per ogni colore primario (rosso, verde, blu)

formati immagini bitmap

- BMP → .bmp → formato standard non compresso per windows
- TIFF → .tif → alta qualità

formati compressi immagini bitmap

- GIF → .gif
- JPEG → .jpg
- JPEG 2000 → .j2k/.jpe → ancora più compresso

CODIFICA VETTORIALE DELLE IMMAGINI

- ↳ rappresentare le immagini tramite un insieme di elementi grafici (linee, rettangoli, ellissi, archi e curve)
- ↳ memorizzare come coordinate numeriche o formule matematiche che specificano forma e posizione
- ↳ necessaria un'operazione di rendering per produrre l'immagine

formati immagini vettoriali

- DXF
- DWG
- CDR
- AI
- WMF

RAPPRESENTARE NUMERI RAZIONALI $\left\{ \begin{array}{l} \text{in virgola mobile} \\ \text{in virgola fissa} \end{array} \right.$

VIRGOLA FISSA → virgola sempre nello stesso punto
 ↳ limita la precisione - alcune cifre non possono essere rappresentate

esempio

$$1110,101$$

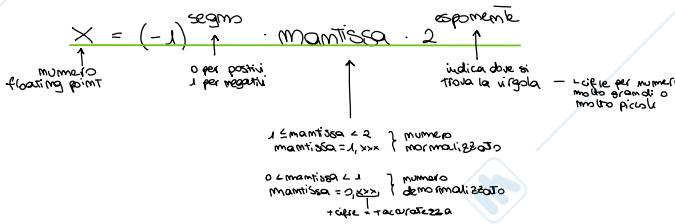
↑ potenze positive ↑ potenze negative

$$2^3 + 2^2 + 2^1 + 2^{-1} + 2^{-3} = 8 + 4 + 2 + 1/2 + 1/8 = 14,625$$

VIRGOLA MOBILE (floating point)

- rappresentare un intervallo enorme di valori con una precisione variabile $\left\{ \begin{array}{l} \text{decimale infiniti} \\ \text{molto grandi} \\ \text{molto piccoli} \end{array} \right\}$ in decimale usiamo notazione scientifica
- adattare il numero di bit di rappresentazione e la posizione della virgola a destra o sinistra a seconda esigenze di rappresentazione di valori diversi
 - ↳ per numeri grandi → + bit a sinistra della virgola
 - ↳ per numeri piccoli → - bit a sinistra della virgola
 - ↳ grazie all'esponente
 - ↳ + bit a destra della virgola
 - ↳ - bit a destra della virgola

Rappresentare in codice binario



IEEE-754 floating point standard

- singola precisione → 32 bit per la memorizzazione $\left\{ \begin{array}{l} 1 \text{ bit per il segno} \\ 8 \text{ bit per l'esponente} \\ 23 \text{ bit per la mantissa} \end{array} \right.$ → in realtà 24 per 1/0 è sempre memorizzato
- doppia precisione → 64 bit per la memorizzazione $\left\{ \begin{array}{l} 1 \text{ bit per il segno} \\ 11 \text{ bit per l'esponente} \\ 52 \text{ bit per la mantissa} \end{array} \right.$ → in realtà 53 per 1/0 è sempre memorizzato
- quadrupla precisione → 128 bit per la memorizzazione $\left\{ \begin{array}{l} 1 \text{ bit per il segno} \\ 15 \text{ bit per l'esponente} \\ 112 \text{ bit per la mantissa} \end{array} \right.$ → in realtà 113 per 1/0 è sempre memorizzato

esponente e polarizzazione (bias)

↳ valore fisso da aggiungere ad ogni esponente

- 8 bit → bias 127
- 11 bit → bias 1023

$$X = (-1)^{\text{segno}} \cdot (1 + \text{mantissa}) \cdot 2^{(\text{esponente} - \text{bias})}$$

esempio

$$\text{segno } 0 \quad \text{esponente } 10000000 \quad \text{mantissa } 10000000000000000000000000000000$$

$$+ 2^7 - 128 \quad 1 \cdot 2^0 + 2^{-1} = 1 + 0,5 = 1,5$$

$$(-1)^0 \cdot 1,5 \cdot 2^1 = 1 \cdot 1,5 \cdot 2 = 3$$