

Architettura degli Elaboratori

1 - Introduzione

Ugo Dal Lago

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna

Anno Accademico 2007/2008

Sommario

Informazioni Utili sul Corso

L'Approccio Strutturale

Pietre Miliari nell'Architettura dei Computer

Tipologie di Computer

Unità di Misura per l'Informazione

Due Famiglie di Computer

Cosa?

- ▶ Vengono studiati i **principi** sottostanti il funzionamento dei calcolatori.
- ▶ In particolare, si analizzano i loro **componenti base** e come questi interagiscono tra loro.
- ▶ Nella tradizionale distinzione tra hardware e software: questo corso esplora l'**interfaccia** tra hardware e software.
 - ▶ Procedendo, ci occuperemo sempre meno di hardware e sempre più di software.
- ▶ Si utilizzerà un approccio strutturato, partendo dal **livello logico digitale** e arrivando al **livello del linguaggio assemblativo**.
- ▶ Verranno considerate alcune architetture di esempio (Pentium, ULTRASparc).

Cosa?

- ▶ Corso in buona parte **descrittivo**:
 - ▶ Si descrivono le diverse parti del calcolatore e la loro interazione.
 - ▶ Vengono presentati i principi e le idee base.
 - ▶ Non si va troppo nel dettaglio, pochi aspetti tecnici-progettuali.
- ▶ Vengono però trattati quattro aspetti più **metodologici**:
 - ▶ Algebra booleana e progettazione di circuiti logici.
 - ▶ Rappresentazione binaria dei numeri.
 - ▶ Microprogrammazione (MAL).
 - ▶ Programmazione assembly (8088).

Perché?

- ▶ L'**informatica** è la scienza che si occupa della conservazione, dell'elaborazione e della rappresentazione dell'informazione.
- ▶ Gli strumenti concreti che l'informatico ha a disposizione sono proprio i calcolatori.
- ▶ Conoscere bene il funzionamento dei calcolatori aiuta l'informatico a svolgere meglio il proprio lavoro.
- ▶ Un corso come questo, però, **non** caratterizza il curriculum dell'Informatico.

"Computer Science is no more about computers than astronomy is about telescopes."

– Edsger Dijkstra

Perché?

- ▶ L'informatica si occupa essenzialmente del **software**:
 - ▶ È possibile interagire con un calcolatore senza sapere come questo è fatto fisicamente.
 - ▶ Per analogia, è possibile guidare un'auto senza conoscere la termodinamica o senza sapere come è fatto un pistone.
- ▶ Ma una conoscenza (non troppo approfondita) dell'**hardware** è utile:
 - ▶ Valutare, scegliere o gestire l'hardware.
 - ▶ Considerare i fattori che determinano le prestazioni,
 - ▶ Gestire i malfunzionamenti.

Chi?

- ▶ Tutti gli studenti iscritti al primo anno del Corso di Laurea in Informatica devono seguire questo corso e superare il relativo Esame.
- ▶ Chi tiene questo corso?
- ▶ Lezioni e esercitazioni: **Ugo Dal Lago**.
 - ▶ Email: `dallago@cs.unibo.it`
 - ▶ Ufficio: **via Malaguti, 1**
 - ▶ Orario di Ricevimento:
 - ▶ **Lunedì**: 14.00 - 16.00
 - ▶ **Martedì**: 14.00 - 16.00
- ▶ Assembly: **Marco Di Felice**.
 - ▶ Email: `difelice@cs.unibo.it`
 - ▶ Ufficio: **mura Anteo Zamboni, 7**

Come?

- ▶ Il **corso** prevede 60 ore di lezione suddivise in
 - ▶ 30-35 ore di lezioni teoriche (Dal Lago)
 - ▶ 10-15 ore di esercitazioni (Dal Lago)
 - ▶ 10-15 ore di lezioni ed esercitazioni sulla programmazione Assembly (Di Felice).
- ▶ Le modalità d'**esame** sono le seguenti:
 - ▶ **Esame scritto**: alcuni esercizi ed alcune domande di teoria.
 - ▶ **due compitiini**, a metà e a fine semestre.
 - ▶ **un compito unico**.
 - ▶ **Progetto**: uno o più programmi Assembly.
 - ▶ **Orale**: a discrezione della commissione.
- ▶ Alcuni consigli:
 - ▶ È **sufficiente** studiare e svolgere alcuni esercizi.
 - ▶ È **indispensabile** capire quello che si è studiato,
 - ▶ È **utile** cercare di distribuire il carico di lavoro (e non rimandare tutto alla settimana precedente l'esame).
 - ▶ È **dannoso** concentrarsi sugli esercizi, perdendo il contatto con la teoria.

Come?

- ▶ Il libro di testo è: *Andrew S. Tanenbaum, Architettura dei Calcolatori. Un Approccio Strutturale*, Quinta Edizione. Pearson Prentice Hall, 2006.
- ▶ Un testo classico. È chiaro e piacevole da leggere.
- ▶ L'autore è un'autorità nella didattica dell'informatica e nel campo dei sistemi operativi.
- ▶ È piuttosto aggiornato.
- ▶ Molto discorsivo. Argomenti difficili, tecnici, quantitativi vengono evitati.
- ▶ In una parte del corso, quella riguardante **le rappresentazioni numeriche, l'algebra booleana e le reti logiche**, il libro di testo andrà integrato con un altro testo: *Morris M. Mano, Charles R. Kime, Reti Logiche*, Pearson Addison Wesley, 2002.
- ▶ I lucidi **non bastano**.

Come?

- ▶ Altri riferimenti interessanti e utili per il corso sono i seguenti:
 - ▶ *Morris M. Mano*. **Computer system architecture**, 3rd edition Prentice Hall, 1993.
 - ▶ *D. Patterson, J. Hennessy*. **Computer organization and design**, 3rd Edition. Morgan Kaufmann, 2004. Edizione italiana della Zanichelli.
 - ▶ *F. Luccio, L. Pagli*. **Reti Logiche e Calcolatore**. Bollati Boringhieri, 1991.

- ▶ La pagina web del corso:

`http://www.cs.unibo.it/~dallago/AE0708/`

si può raggiungere dalla pagina web del docente

`http://www.cs.unibo.it/~dallago/`

- ▶ Le regole da seguire per superare l'esame sono descritte minuziosamente in:

`http://www.cs.unibo.it/~dallago/AE0708/REGOLE.pdf`

- ▶ Il newsgroup del corso è:

`unibo.cs.informatica.architettura`

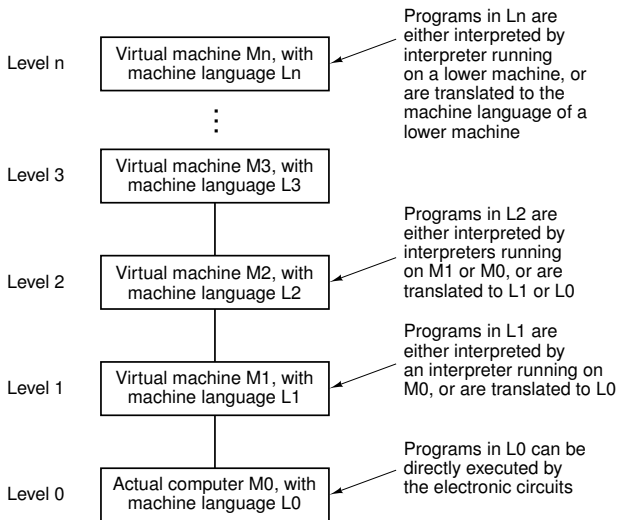
Macchine Virtuali

- ▶ Con **programma** si intende una sequenza di istruzioni che il calcolatore può eseguire e che descrive come risolvere un problema computazionale.
- ▶ Il **linguaggio macchina** descrive l'insieme delle istruzioni primitive che il calcolatore può eseguire.
 - ▶ Di solito il linguaggio macchina risulta difficile da utilizzare.
- ▶ Si definisce quindi una **macchina virtuale** che sfrutta la funzionalità del calcolatore, ma che può essere programmata in un linguaggio diverso, spesso più semplice da utilizzare.
- ▶ Il calcolatore può essere visto come una **macchina a livelli**: un insieme di **macchine virtuali**.

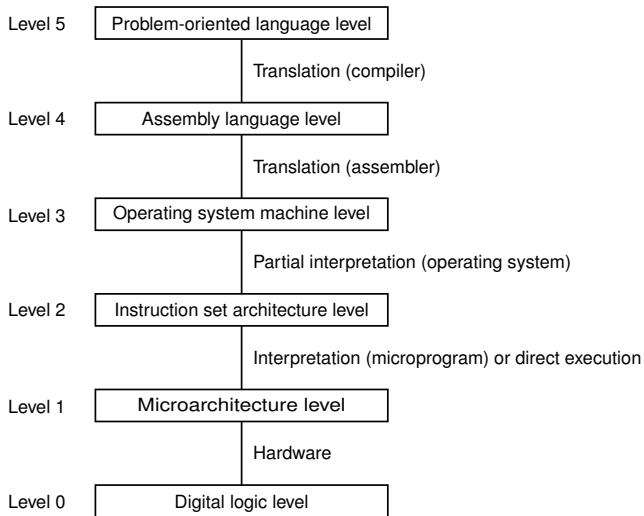
Traduzione e Interpretazione

- ▶ Chiamiamo M_0 l'elaboratore che abbiamo a disposizione e L_0 il relativo linguaggio.
- ▶ Possiamo pensare di costruire una macchina virtuale M_1 con un relativo linguaggio macchina L_1 .
- ▶ Un modo per eseguire un programma scritto in L_1 nella macchina M_0 consiste nel **tradurre** ogni istruzione di L_1 in una sequenza di istruzioni di L_0 ad essa equivalente.
- ▶ Un altro modo per eseguire programmi scritti in L_1 tramite le funzionalità di M_0 consiste nello scrivere un programma in L_0 , detto **interprete**, che sia in grado di eseguire tutti i programmi in L_1 .
- ▶ Per rendere traduzione e interpretazione utilizzabili, i linguaggi L_0 e L_1 non devono essere troppo diversi tra loro.
- ▶ La stessa idea si può applicare a L_2 e M_2 , L_3 e M_3 , etc.

Macchina Multilivello



Un Computer a Sei Livelli



Livello Logico Digitale

- ▶ Non ci interessiamo del livello dei dispositivi analogici.
- ▶ Gli oggetti in questione sono le **porte logiche**.
- ▶ Ogni porta è costituita da al più una manciata di transistor.
- ▶ Ogni porta calcola in output una semplice funzione dei valori in input.
- ▶ Un piccolo numero di porte possono essere combinate per formare una memoria a 1 bit.
- ▶ Combinando le memorie in gruppi di 16,32 o 64 unità si possono comporre i cosiddetti **registri**.

Livello di Microarchitettura

- ▶ Vi è una **memoria locale**, formata da un gruppo di registri.
- ▶ Vi è un circuito, chiamato **ALU** (*Arithmetic Logic Unit*), capace di effettuare semplici operazioni aritmetiche.
- ▶ I registri sono connessi alla ALU tramite un **percorso dati**.
- ▶ Il percorso dati :
 - ▶ Può essere controllato da un programma che l'utente può modificare, detto **microprogramma**.
 - ▶ Può essere controllato da circuiti che non sono modificabili dall'utente (ma anche in questo caso parleremo di microprogramma).

Livello ISA e Livello del Sistema Operativo

- ▶ **ISA** significa *Instruction Set Architecture Level*.
- ▶ Il Livello ISA è il primo livello pubblico.
- ▶ Le istruzioni del Livello ISA vengono eseguite direttamente dal microprogramma.
- ▶ Il Livello del Sistema Operativo è un'estensione del Livello ISA, ottenuta aggiungendo alcuni nuovi servizi.
- ▶ I nuovi servizi messi a disposizione nel Livello del Sistema Operativo vengono eseguiti (interpretati) da un programma del livello ISA, detto appunto **Sistema Operativo**.

Livello del Linguaggio Assemblativo

- ▶ Tutti i livelli inferiori non sono utilizzati dal programmatore medio.
- ▶ I livelli inferiori sono concepiti per eseguire interpreti e traduttori necessari ai livelli più alti. Tali programmi sono scritti dai cosiddetti **programmatore di sistema**.
- ▶ I programmi di questo livello e dei livelli superiori sono invece dei programmi che risolvono **problemi applicativi**.
- ▶ I programmi in linguaggio assemblativo, a differenza di quelli dei livelli precedenti, sono sequenze di caratteri alfanumerici pensate per essere comprensibili.
- ▶ Il programma che traduce i programmi in linguaggio assemblativo in programmi del livello ISA è detto **assemblatore**.

Evoluzione delle Macchine Multilivello

- ▶ La **distinzione** tra hardware e software è diventata sempre meno importante.
- ▶ Nel 1951, Maurice Wilkes propone la **microprogrammazione**, che si inserisce tra il livello logico digitale e il livello ISA.
- ▶ A partire dagli anni '60, le operazioni (ripetitive) necessarie a far funzionare un calcolatore vengono automatizzate e ciò porta alla nascita del **sistema operativo**.
- ▶ Negli anni '60 e '70, un numero sempre maggiore di funzionalità che prima venivano implementate al livello ISA migrano verso il livello del microcodice.
- ▶ Più recentemente (dagli anni '80) si è capito che eliminando il microprogramma e riducendo il numero delle istruzioni a livello ISA si può ottenere un sensibile miglioramento delle prestazioni.

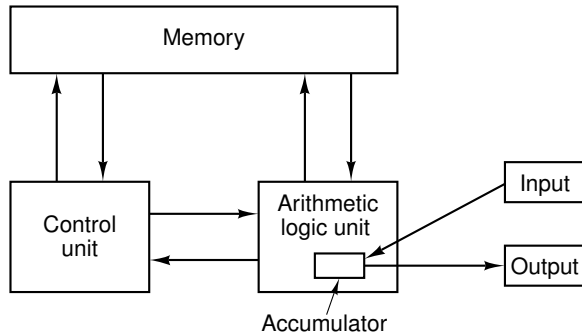
Calcolatori Meccanici e Elettromeccanici (1642-1945)

- ▶ *Blaise Pascal* (1623-1662) e *Gottfried Leibniz* (1646-1716)
 - ▶ Potevano eseguire solo **operazioni aritmetiche**.
- ▶ *Charles Babbage* (1792-1871).
 - ▶ **Difference Engine**. Un singolo algoritmo.
 - ▶ **Analytical Engine**. Qualunque algoritmo (parte dell'input).
- ▶ *Konrad Zuse* (1910-1995).
 - ▶ **Z1** (fine degli anni '30). Relé elettromagnetici.
- ▶ *John Atanasoff* (1903-1995).
 - ▶ **ABC** (Iowa State College, 1939). Aritmetica binaria e logica booleana. Solo un prototipo.
- ▶ *George Stibitz* (1904-1995).
 - ▶ **Complex Number Calculator** (1940). Poteva calcolare nel dominio dei numeri complessi.
- ▶ *Howard Aiken* (1900-1973).
 - ▶ **Mark I** (Harvard University, 1944). Analytical Engine con relé elettromagnetici.

Valvole (1945-1955)

- ▶ **COLOSSUS** (1943).
 - ▶ Scopi militari. Restò segreto per decenni. *Alan Turing* partecipò al suo progetto.
- ▶ **ENIAC** (University of Pennsylvania, 1946).
 - ▶ 18.000 valvole termoioniche, 1500 relé, 140 KW.
 - ▶ 20 registri da 10 cifre decimali.
 - ▶ Si programmava regolando 6000 interruttori multi-posizione.
- ▶ **IAS** (Institute for Advanced Study, 1952)
 - ▶ Si passò ad una rappresentazione binaria dei dati.
 - ▶ Non solo i dati, ma anche i programmi risiedevano in memoria.

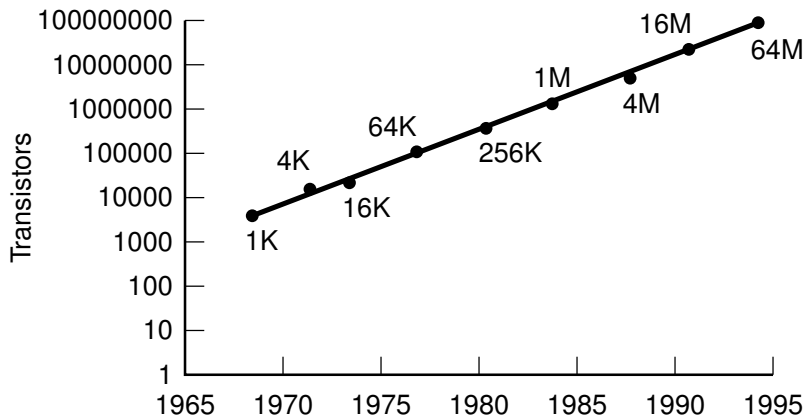
Macchina di Von Neumann



Fasi Successive

- ▶ **Transistor (1955-1965).**
 - ▶ Permettono di rappresentare l'informazione in uno spazio più piccolo, consumando meno energia e processando l'informazione più velocemente.
 - ▶ Memorie nell'ordine delle migliaia di caratteri, migliaia di istruzioni in un secondo.
 - ▶ Esempi: **PDP-1, PDP-8, IBM 7094, Cray**, etc.
- ▶ **Circuiti Integrati (1965-1980).**
 - ▶ Decine di transistor su un unico chip.
 - ▶ Di conseguenza, macchine più piccole, veloci ed economiche.
 - ▶ Esempi: **IBM System/360, PDP-11**, etc.
- ▶ **Integrazione a Larghissima Scala (VLSI) (1980-?).**
 - ▶ Fino a milioni di transistor su un unico chip.
 - ▶ Esempi: **PC IBM, Apple**, etc.

Legge di Moore



Spettro di Computer

Tipo <i>Esempio di applicazione</i>	Prezzo (dollari)
Computer usa e getta <i>Cartoline d'auguri</i>	0,5
Microcontrollore <i>Orologi, automobili, elettrodomestici</i>	5
Computer da gioco <i>Videogiochi</i>	50
Personal computer <i>Desktop o Notebook</i>	500
Server <i>Server di rete</i>	5K
Raggruppamento di workstation <i>Minisupercomputer dipartimentali</i>	50K-500K
Mainframe <i>Elaborazione dati in una banca</i>	5M

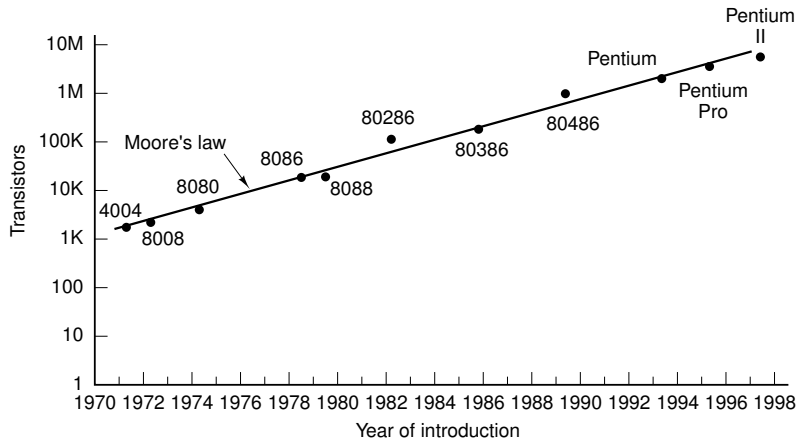
Bit, Byte, etc.

Nome	Significato
Bit	cifra binaria
Byte	8 bit
KByte (KB)	2^{10} ($\sim 10^3$) byte
MByte (MB)	2^{20} ($\sim 10^6$) byte
GByte (GB)	2^{30} ($\sim 10^9$) byte
TByte (TB)	2^{40} ($\sim 10^{12}$) byte

Intel Pentium

- ▶ È prodotto dalla Intel.
- ▶ Il processore Pentium (o suoi cloni e derivati) equipaggia buona parte dei personal computers odierni.
- ▶ Le sue radici, però, risalgono all'inizio degli anni Settanta.
- ▶ Dall'Intel **4004**...
 - ▶ 2300 transistor.
 - ▶ 640 bytes di memoria indirizzabile....si passa al **Pentium 4**.
 - ▶ 42 milioni di transistor.
 - ▶ 4GB bytes di memoria indirizzabile.
- ▶ Si è passati attraverso numerosi cambiamenti nell'architettura, volti a migliorarne le prestazioni.
- ▶ Si è mantenuta la cosiddetta **compatibilità all'indietro**.
- ▶ Tra i vari processori della famiglia, uno dei primi è **Intel 8088**, che verrà studiato approfonditamente in questo corso.

Legge di Moore per la Famiglia Intel Pentium



Sun ULTRASparc

- ▶ Introdotto all'inizio nel 1995.
- ▶ Appartenente alla famiglia **SPARC**, nata nel 1987.
- ▶ Prodotto da Sun Microsystems.
- ▶ **SPARC** sta per "Scalable Processor ARChitecture".
- ▶ ULTRASparc è un'architettura a 64 bit, capace quindi di indirizzare 2^{64} parole di memoria.
- ▶ I processori con architettura SPARC hanno una potenza in genere maggiore dei contemporanei processori della famiglia Pentium.
- ▶ I processori con architettura SPARC sono stati pensati fin dall'inizio come workstations grafiche connesse tra loro in rete.