

SUPSI

Basi di Informatica

Loris Grossi, Fabio Landoni, Andrea Baldassari

Contenuto realizzato in collaborazione con: T. Leidi, A.E. Rizzoli, S. Pedrazzini

Fondamenti di Informatica

Bachelor in Ingegneria Informatica



Etimologia

Il termine informatica, contrazione di **informazione automatica**, deriva dalla lingua tedesca Informatik ed è stato coniato nel 1957 da Karl Steinbuch nel suo articolo “Informatik: Automatische Informationsverarbeitung”. Il termine è stato poi ripreso da Philippe Dreyfus nel 1962.

— Wikipedia

L'informatica non riguarda i computer più di quanto l'astronomia riguardi i telescopi.

— Edsger Wybe Dijkstra



Cos'è l'informatica?

L'informatica è la **scienza del trattamento razionale**, mediante macchine o procedure automatiche, **dell'informazione**, considerata come supporto delle conoscenze e delle comunicazioni nei campi tecnici, economici e sociali.

— *Definizione data dall'Académie française, 1966*

Informatica: scienza e tecnica della elaborazione dei dati e, in generale, del trattamento automatico delle informazioni.

— *Nuovo Zanichelli, vocabolario della lingua italiana, 1983*

L'informazione è un **insieme di fatti la cui conoscenza permette d'intraprendere un'azione** o di prendere delle decisioni.



L'elaborazione automatica

L'elaboratore elettronico gestisce grandi quantità di dati grazie ad una notevole capacità di calcolo.

L'efficacia dell'elaboratore è evidenziata soprattutto quando si devono eseguire, un gran numero di volte, operazioni di tipo ripetitivo.

Gli elaboratori sono soltanto rapidissimi e precisi esecutori d'ordini.

L'attività creativa, cioè la risoluzione del problema posto, è, almeno per il momento, compito dell'uomo.



L'elaborazione automatica

Strumenti pensati per aiutare ad eseguire i calcoli esistono da **migliaia di anni**. Il principio di funzionamento è quello delle dita delle mani.

Ad esempio, l'**abaco** è uno strumento ideato, probabilmente in Cina, per facilitare le operazioni aritmetiche. Si hanno sue tracce già nella Babilonia del 2400 a.C.





I primi computers

Vari **calcolatori meccanici** furono inventati nell'antichità e nel medioevo per eseguire **calcoli di tipo astronomico**. Fra questi, ad esempio la macchina di Anticitera dell'antica Grecia (150 – 100 a.C.).

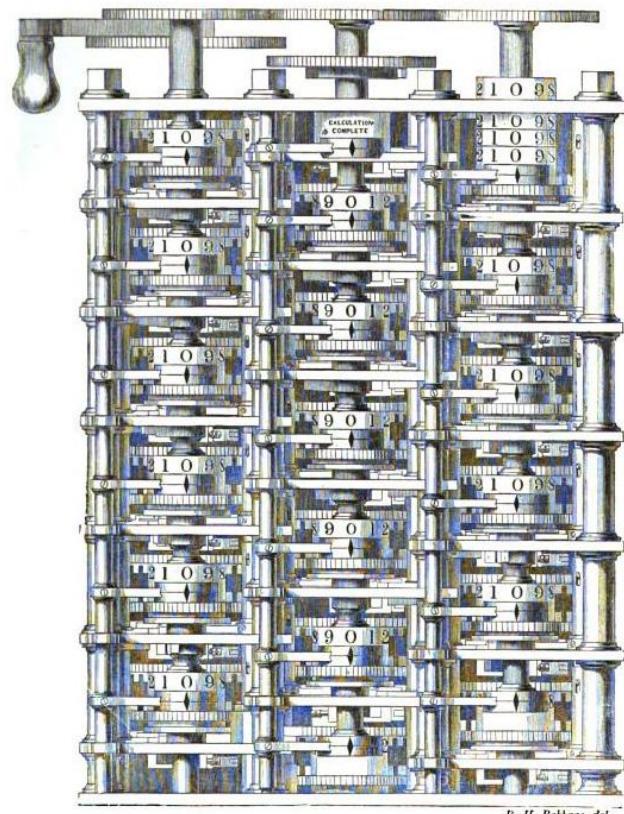




I primi computers

I primi **computers programmabili** furono inventati solo all'inizio del 19esimo secolo e riuscirono ad emergere, nella loro forma moderna, a partire dagli anni 40.

Charles Babbage, un ingegnere meccanico inglese è considerato il padre del computer. Inventò un dispositivo meccanico, capace di eseguire calcoli.





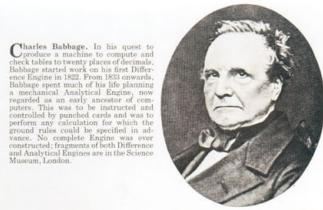
This chart has been designed to give a broad view of computer development and the connections and influences between the various manufacturers. The centre section traces the machines themselves, and panels flanking it left to right. For the sake of clarity it was not possible to include every type of computer (see various notes down the right hand side). Every effort has been made to ensure the information is accurate, but the publishers can take no responsibility for any errors in the chart or text.



Pascal's Calculating Machine. The first calculating machine was built by Blaise Pascal in 1642. It was built in Paris by Blaise Pascal, the French scientist and philosopher, invented his calculating machine at the age of 19. In just a few years he developed over fifty models capable of addition and subtraction. The machine had a wheel of wheels, each wheel of wheels carrying digits from 0 to 9, and the results window similar to the display on a modern calculator.



The Arithmometer. In 1850 the first commercially mass-produced calculating machine was devised by a financier, Thomas de Colmar, and some 1500 machines to his design were built. The Arithmometer represented a great step forward, being capable of addition, subtraction and multiplication. Many variations of the machine were sold well into this century.



Charles Babbage. In his quest to compute any check tables to twenty places of decimals, Babbage constructed the Difference Engine in 1822. From 1832 onwards, Babbage worked on the Analytical Engine, a mechanical Analytical Engine, now regarded as an early computer. It could be instructed and controlled by punched cards and was to perform any calculation that could be specified in advance and stored in memory. Babbage also constructed fragments of both Difference and Analytical Engines. These are in the Science Museum, London.



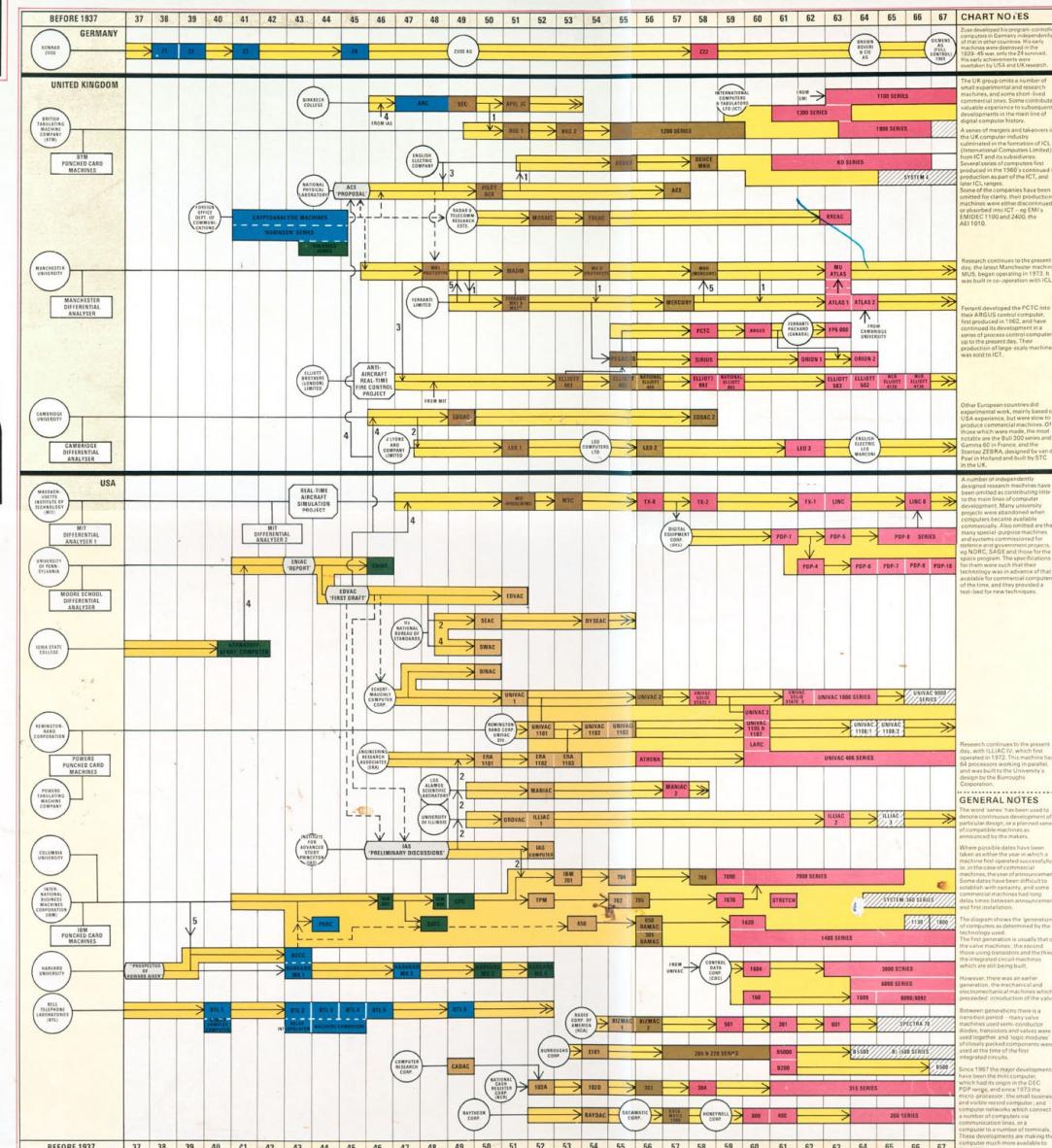
Herman Hollerith. Hollerith developed the use of punched cards for computing the 1890 U.S. Population Census in the USA. A printer, a card reader, a sorting box and a tabulator containing the card reader. This machine was the first general growth of punched card machines and made possible the processing of data carried by punched cards.



Differential Analyzers. These machines were large-scale calculators developed to handle the time-consuming task of solving differential equations. They were used in the Second World War for ballistic and trajectory calculations. The first differential analyzers were built at Manchester and elsewhere in Britain. They were the most powerful machines for calculations before the advent of electronic computers.

History of Computers

Science Museum Copyright 1975
Produced by Data Recording Products,
GM United Kingdom Ltd.



Valve Computers. The first practical stored program electronic computer to work anywhere in the world was the Cambridge University EDVAC (Electronic Delay Storage Automatic Calculator) in May 1949. It was built by the Mathematics Department of the University of Cambridge. The team at the University of Manchester had started work on the EDSVAC computer earlier than Cambridge but its completion was delayed by the war. The EDSVAC was completed in 1950. The first valve computer to work in Britain was the Ferranti Mark 1. The first machine used electronic valves and limited storage devices such as mercury delay lines, electrostatic and magnetic cores, and magnetic drums.

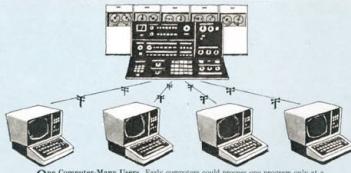
Despite the fact that a trial computer to test electronic storage was working at Manchester University in 1948 there was no real computer until the EDSVAC was completed in 1950.

Although the input for early computers came from punched cards, the output was either punched cards, paper tape, or paper prints.

Both the EDSVAC and the Ferranti Mark 1 were built in Britain, but the first computer to work in America was the UNIVAC I in the U.S.A. and the first LEO (Lyons Electronic Office) in the U.K.



Growth of Computer Applications. Parallel with the changeover to reliable transistored internal components came the improved reliability of magnetic tape drives, and magnetic drum and disc stores. The combination of these elements led to the adoption by the early 1960's of the use of computers becoming predominantly commercial, whereas before, computers had been mainly used in scientific and government projects, such as the first practical use of computers in controlling industrial processes such as chemical



Of Computer-Many Users. Early computers could process one program only at a time. In the late 1950's techniques were developed to run several programs together. This meant that the fast central processor did not remain idle for so long, while slower units were being used.

A very important advance in making computers available to many users at a time came in the mid 1960's with the introduction of time sharing. In this system whereby several users were able to use a computer simultaneously via computer terminals, the speed of the computer was increased. The number of users that could be rapidly addressed so that thousands of terminals can be serviced by one computer.

Each terminal user has the impression that the entire machine is working for him alone.

KEY TO SYMBOLS		LIST OF ABBREVIATIONS	
Yellow box	Ongoing documents	ABP	All Purpose Electronic Computer (with links to incomplete)
Blue box	Relay and Electro-mechanical machines	ADAC	Advanced Data Analysis Computer
Green box	Valve computers without stored programs	ADM	Advanced Data Management
Brown box	Transistor computers	ADCC	Advanced Data Communication Computer
Gold box	Integrated circuit computers	ADCCP	Advanced Data Communication Computer Processor
Grey box	Transistorised computers	ADCCP/C	Advanced Data Communication Computer Processor/C
White box	Integrated circuit computers	ADCCP/C/C	Advanced Data Communication Computer Processor/C/C
2	Engineering or commercial development of	ADCCP/C/C/C	Advanced Data Communication Computer Processor/C/C/C
3	Dissemination or general	ADCCP/C/C/C/C	Advanced Data Communication Computer Processor/C/C/C/C
4	Information or influence of general concepts and	ADCCP/C/C/C/C/C	Advanced Data Communication Computer Processor/C/C/C/C/C
5	Information or influence of nationally	ADCCP/C/C/C/C/C/C	Advanced Data Communication Computer Processor/C/C/C/C/C/C



Storia dei computers

Molte delle tappe dello sviluppo dei computers sono state a lungo segrete, in quanto legate a progetti militari, ma oggi è possibile ripercorrere la storia del computer in modo trasparente.



https://it.wikipedia.org/wiki/Storia_del_computer

https://en.wikipedia.org/wiki/History_of_computing_hardware

https://en.wikipedia.org/wiki/History_of_personal_computers



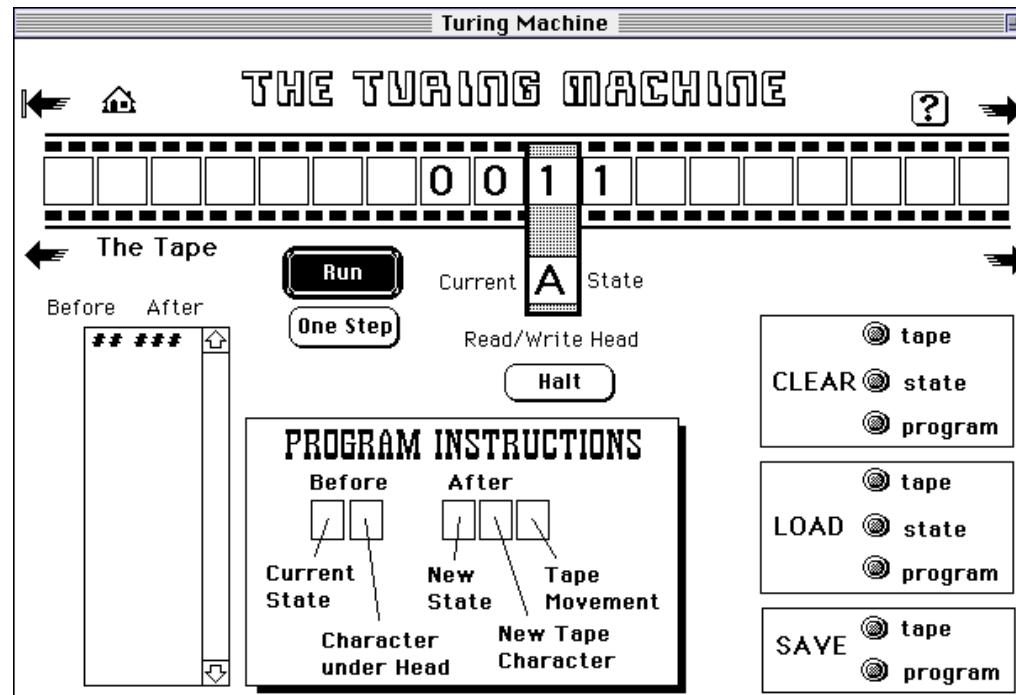
La macchina di Turing

Il principio del funzionamento dei **computers moderni** fu introdotto da **Alan Turing**, che ne descrisse l'idea in una pubblicazione del 1936.

Turing riformulò i risultati dello studio di **Kurt Gödel** del 1931 sui **limiti della calcolabilità**, sostituendo il linguaggio formale utilizzato da Gödel con una macchina ipotetica conosciuta come **macchina di Turing**, capace di eseguire qualsiasi operazione matematica rappresentabile **in forma di algoritmo**.



La macchina di Turing



Una **macchina di Turing** è una macchina ideale che manipola i dati contenuti su un nastro di lunghezza infinita, secondo un insieme prefissato di regole ben definite. In altre parole, è un modello astratto che definisce una macchina in grado di eseguire **algoritmi**.



Computers digitali

I primi computers digitali furono di tipo elettromeccanico. Circuiti elettrici pilotavano dei relais (interruttori) utilizzati per eseguire i calcoli. Questi dispositivi erano molto lenti e furono sostituiti da computers completamente elettrici che utilizzavano valvole al posto dei relais.

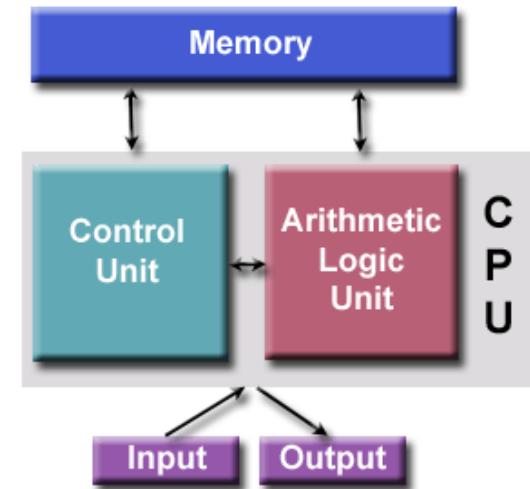




L'architettura di Von Neumann

Nel 1945, Von Neumann sviluppò **un'architettura sequenziale**, ancora in uso adesso, con le seguenti particolarità:

- le istruzioni sono caricate in memoria,
- i dati sono anch'essi in memoria,
- le istruzioni vengono eseguite una per una dall'unità centrale di calcolo (CPU).



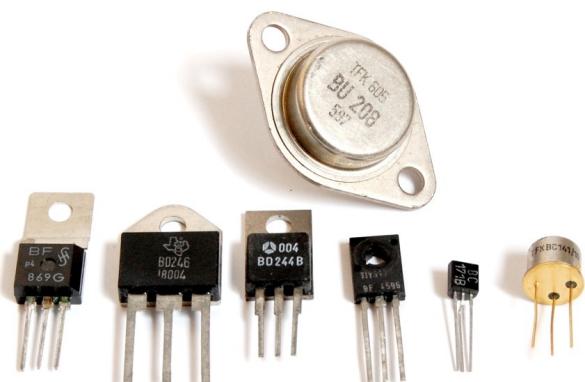


Computers a transistors

I **transistors** furono inventati nel 1947. A partire dal 1955 sostituirono le valvole all'interno dei computers, dando origine alla **seconda generazione di computers**.

I transistors sono semiconduttori capaci di **amplificare e commutare** segnali elettrici.

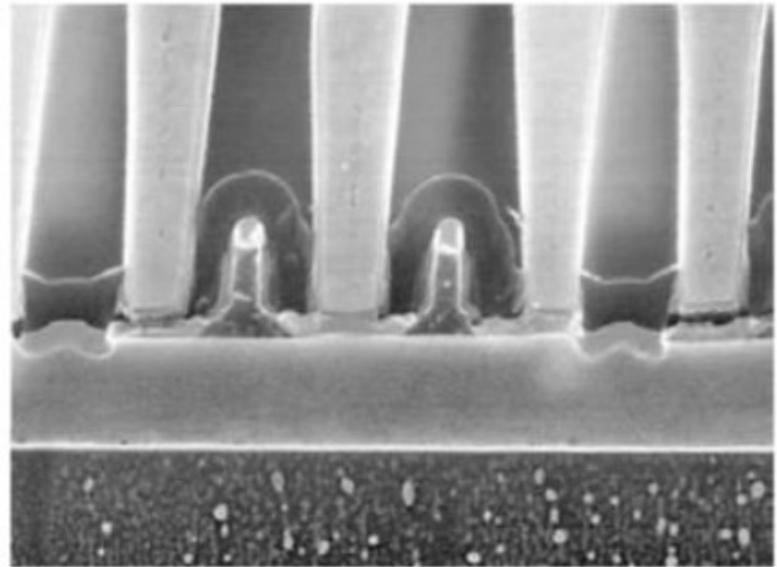
Rispetto alle valvole, i transistors sono più piccoli, consumano meno energia, dissipano meno calore e si guastano meno frequentemente.





Circuiti integrati

Il passo successivo nello sviluppo dei computers fu rappresentato dall'avvento dei **circuiti integrati**: componenti elettronici complessi capaci di contenere uno o più circuiti elettronici miniaturizzati.



Due transistors dei 790 milioni presenti all'interno del processore POWER6 dell'IBM



Microprocessori

A partire dagli anni '70 l'evoluzione dei computers fu caratterizzata dall'introduzione dei **microprocessori**.

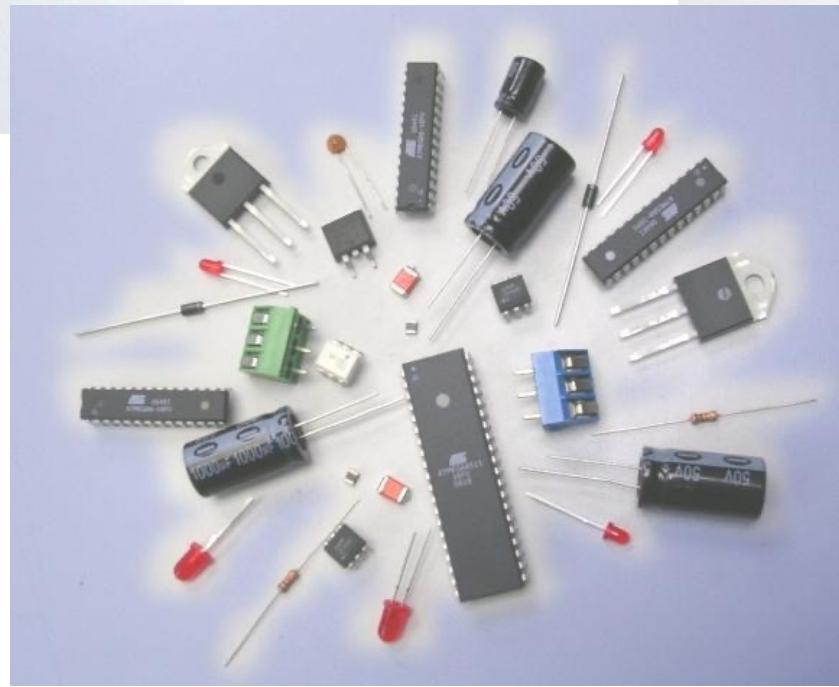
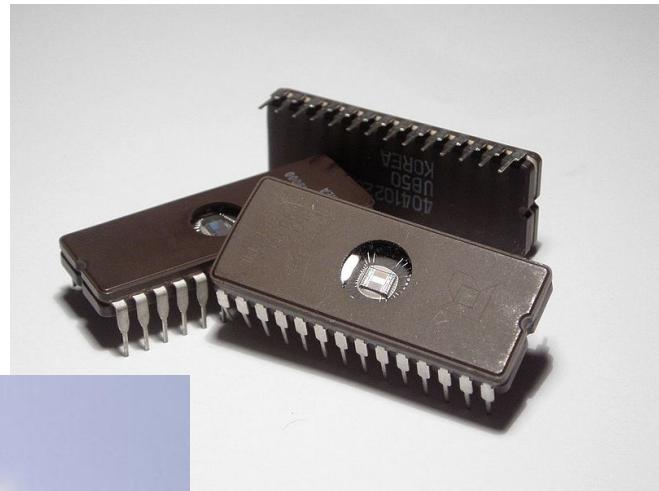
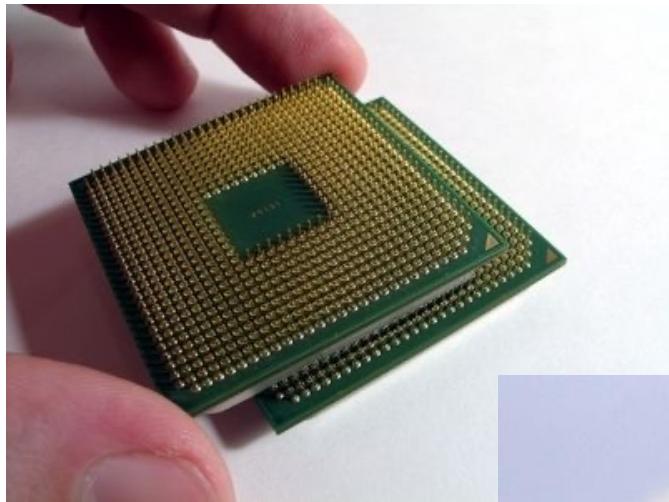
I microprocessori permisero l'integrazione delle funzionalità della CPU in un **unico circuito integrato**.

I microprocessori sono multifunzione, programmabili, accettano dati digitali che, tramite le istruzioni presenti in memoria, possono essere processati per produrre risultati.

Negli anni '70 e '80, i microprocessori permisero lo sviluppo dei primi personal computers.



Circuiti integrati e microprocessori





Vendite

Computer Sales	2011	2010	2000	1975
Number of computers sold in the U.S.	95.4 Million	93 Million	46 Million	40,000
Number of computers sold globally	355.2 Million	346.2 Million	134.7 Million	50,000
Computer Sales Revenue	2011	2010	2000	1975
U.S. computer sales revenue	\$85.5 Billion	\$83.4 Billion	\$86.9 Billion	\$50 Million
Worldwide computer sales revenue	\$329 Billion	\$321 Billion	\$251 Billion	\$60 Million
Computer Sales All-Time				
Number of computer sales all-time	3.287 Billion			
Computer sales revenue all-time	\$4.835 Trillion			
Purpose and Characteristics				
Percent of computers sold for business	74 %			
Percent of desktop computers sold	81.5 %			
Percent of laptop computers sold	16.4 %			
Percent of servers sold	2.1 %			
Region of Sales				
Percent of computers sold to the U.S.	38.8 %			
Percent of computers sold to Europe	25 %			
Percent of computers sold to Asia-Pacific	11.7 %			
<i>Figures include desktop PCs, mobile PCs, and servers using the Intel x86 processor architecture</i>				



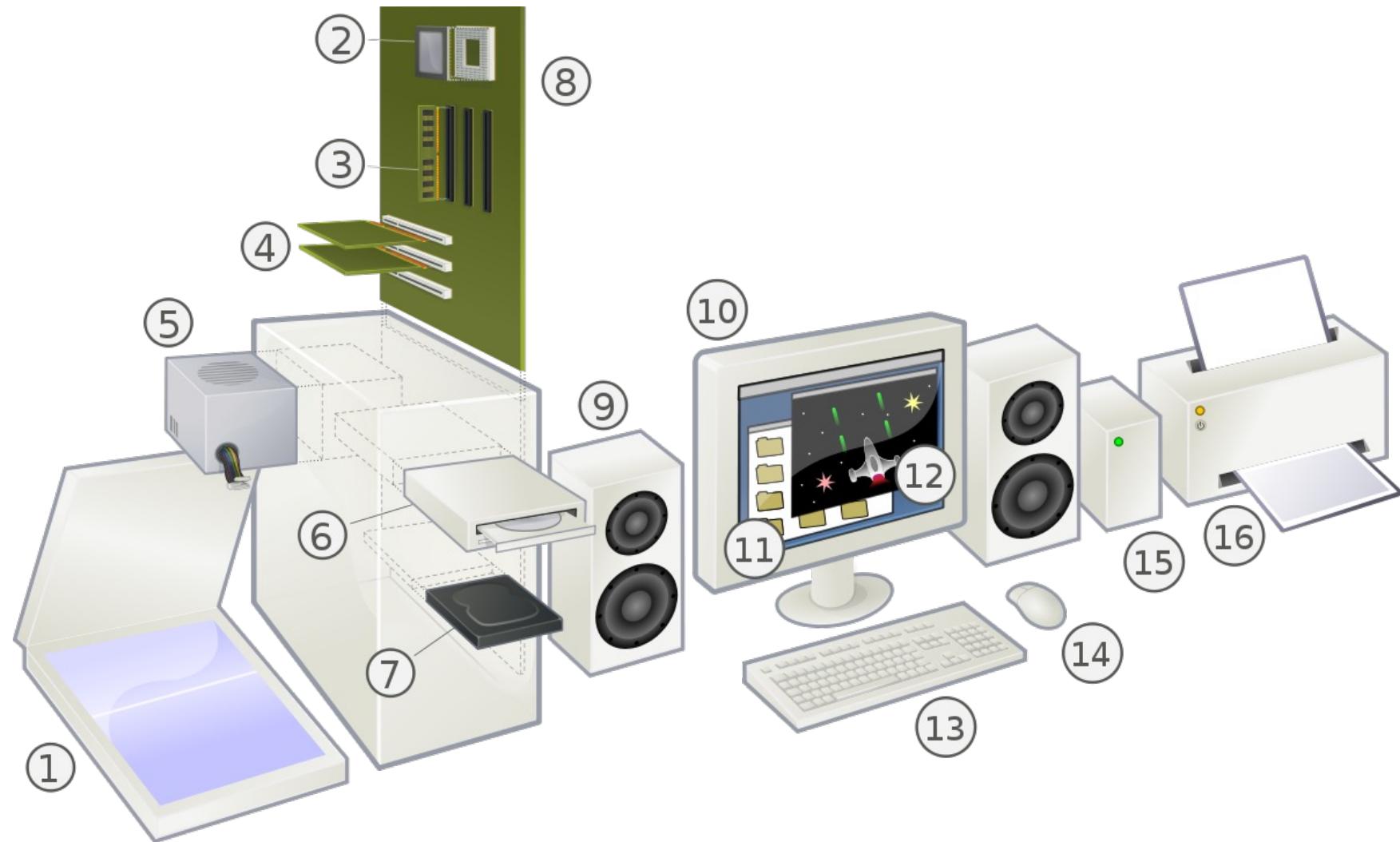
Computer

L'elaboratore è costituito da due parti:

- **Hardware**: parte fisica dell'elaboratore. Ad esempio la tastiera, il monitor, i cavi, i circuiti elettrici ed elettronici e le parti meccaniche.
- **Software**: insieme di tutti i programmi e delle funzioni che servono all'elaboratore per gestire i dati e se stesso.

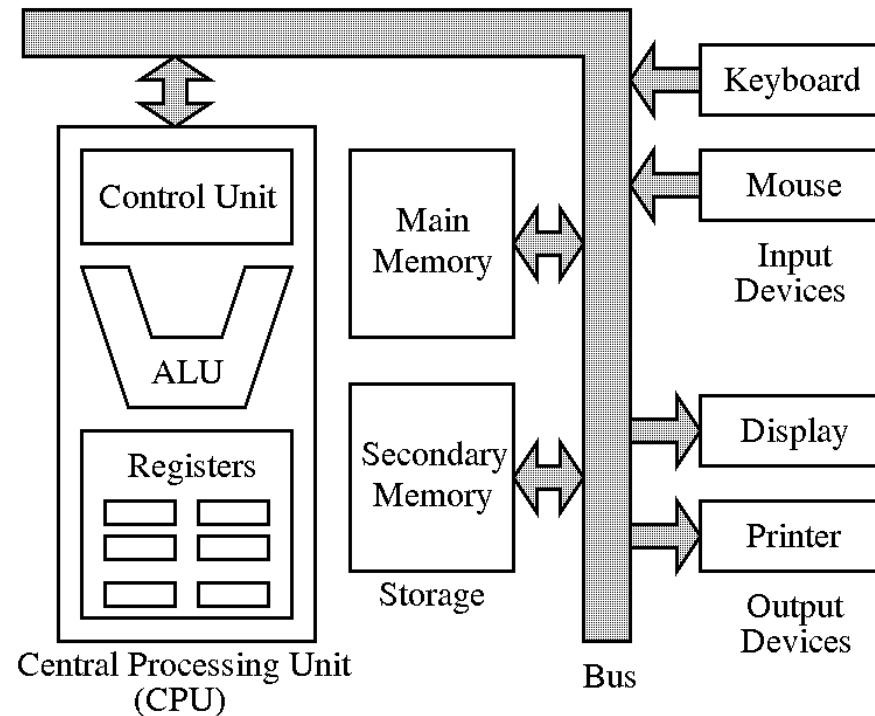


Hardware



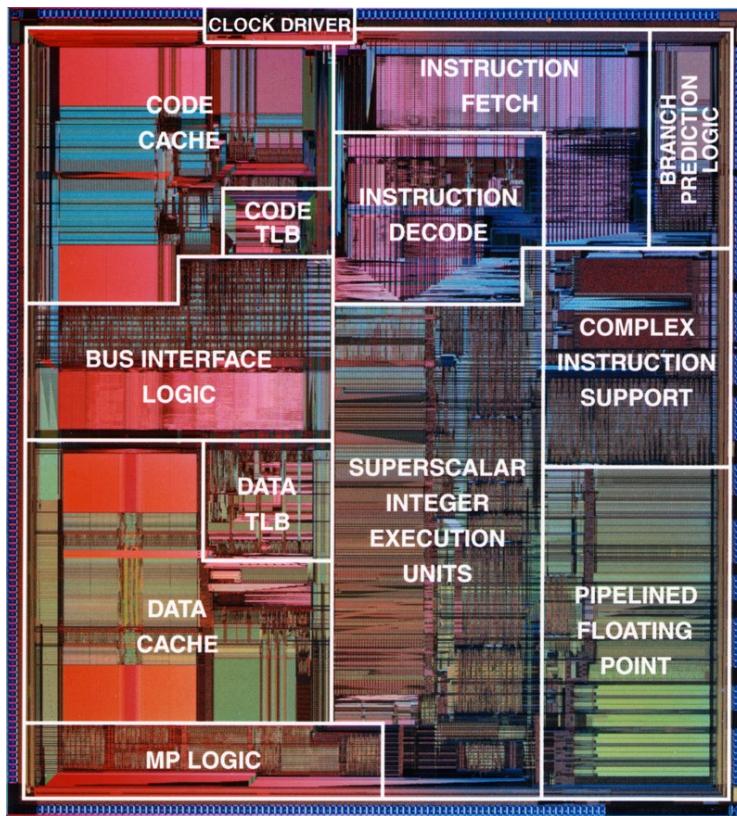


Schema concettuale di un computer





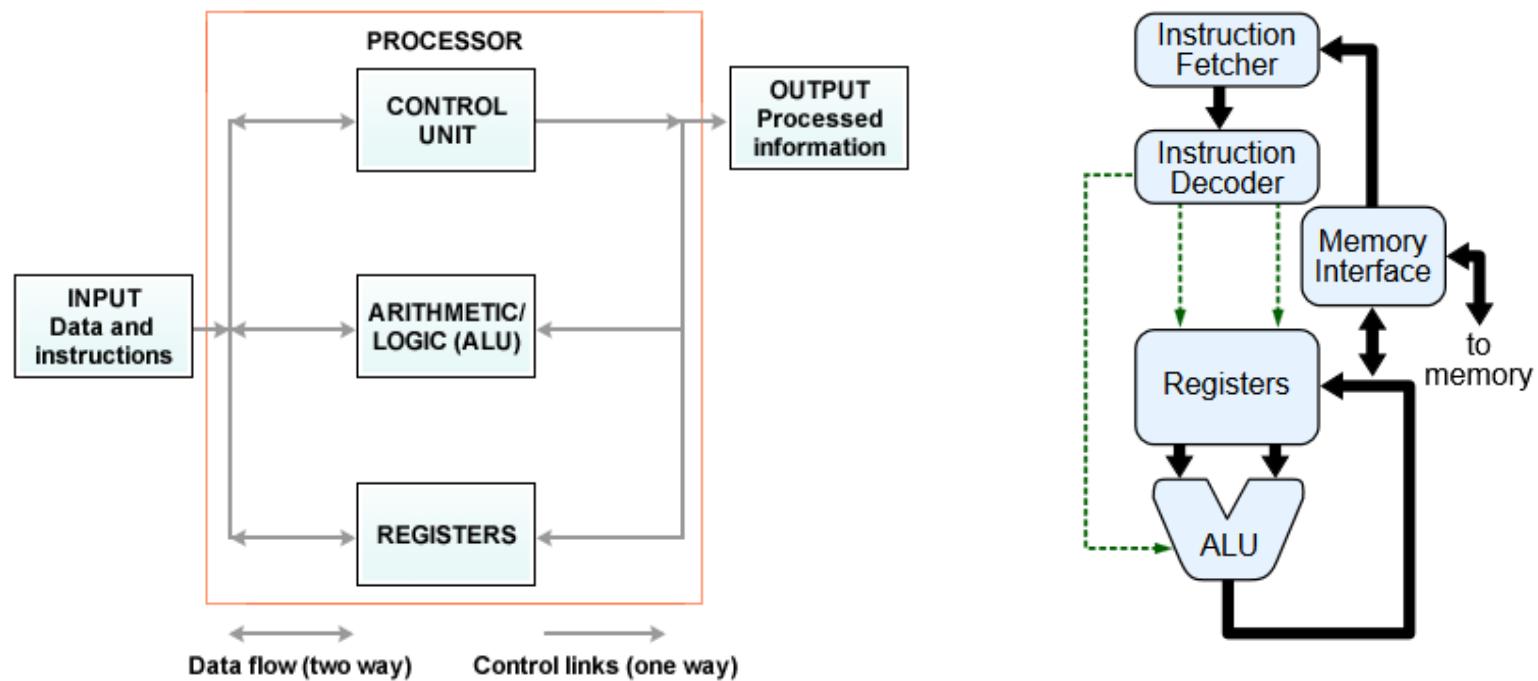
CPU



Intel Pentium



Schema concettuale di una CPU





Fabbricazione dei microprocessori

La fabbricazione dei circuiti integrati comprende una sequenza di passi di **fotolitografia e processi chimici**, in cui i circuiti vengono gradualmente creati su un **wafer** di materiale semiconduttore.

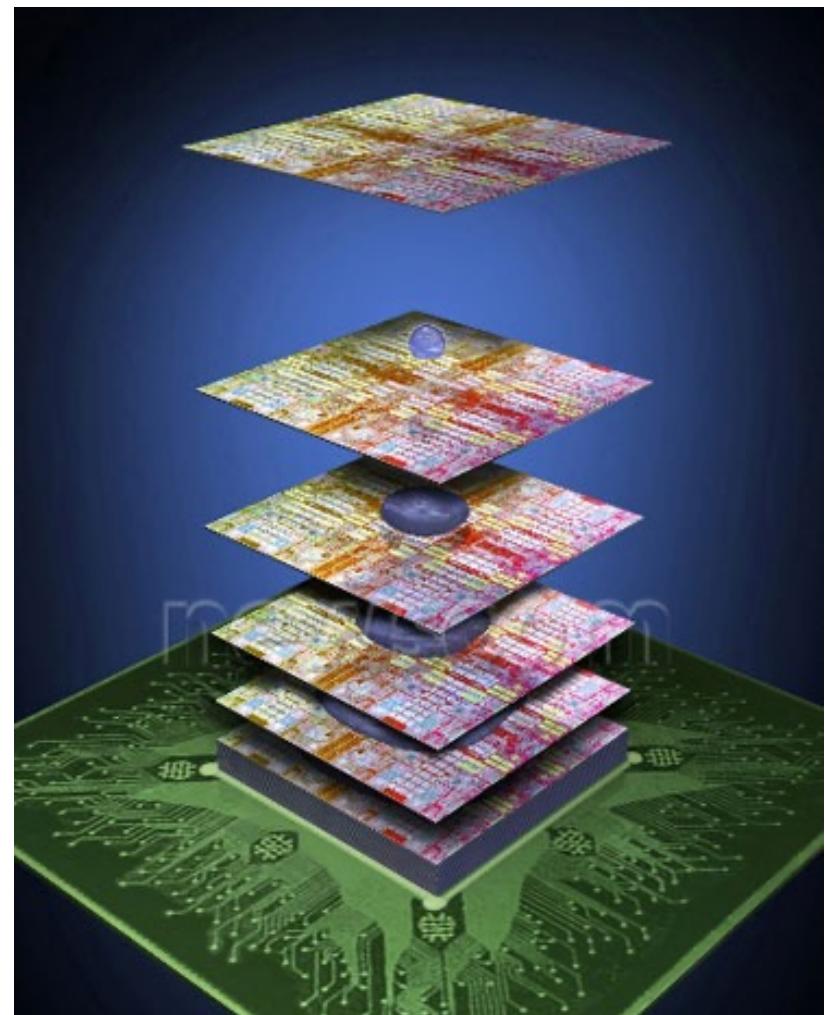
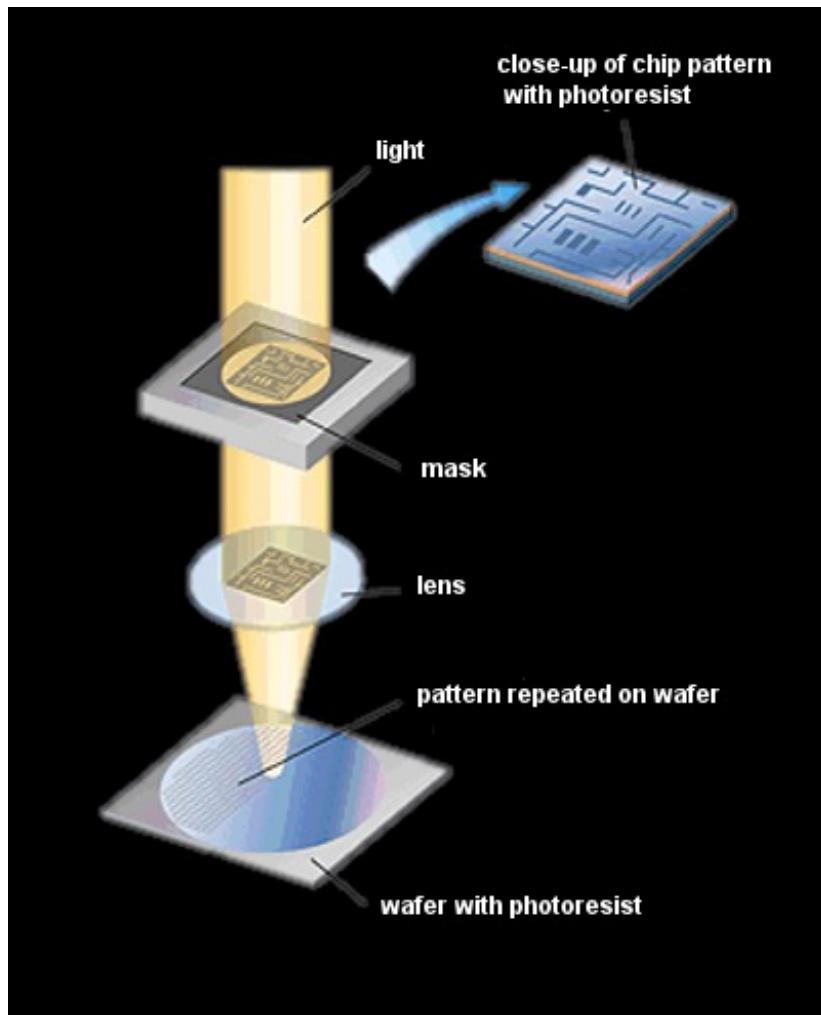
Il silicio è il materiale semiconduttore più comunemente utilizzato nell'industria dei microchips.

Introducendo sostanze chimiche in uno strato di silicio è possibile realizzare i transistors: **gli zeri e gli uni** necessari per eseguire i programmi informatici.

L'intera fabbricazione di un circuito richiede dalle 6 alle 8 settimane e viene eseguita in fabbriche altamente specializzate chiamate **fabs**.

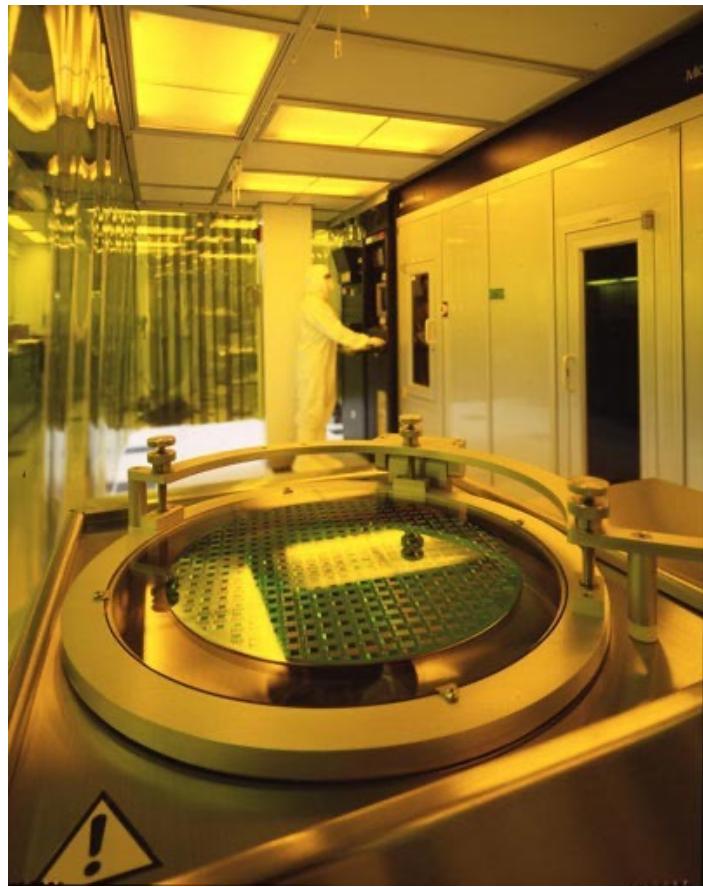


Fotolitografia



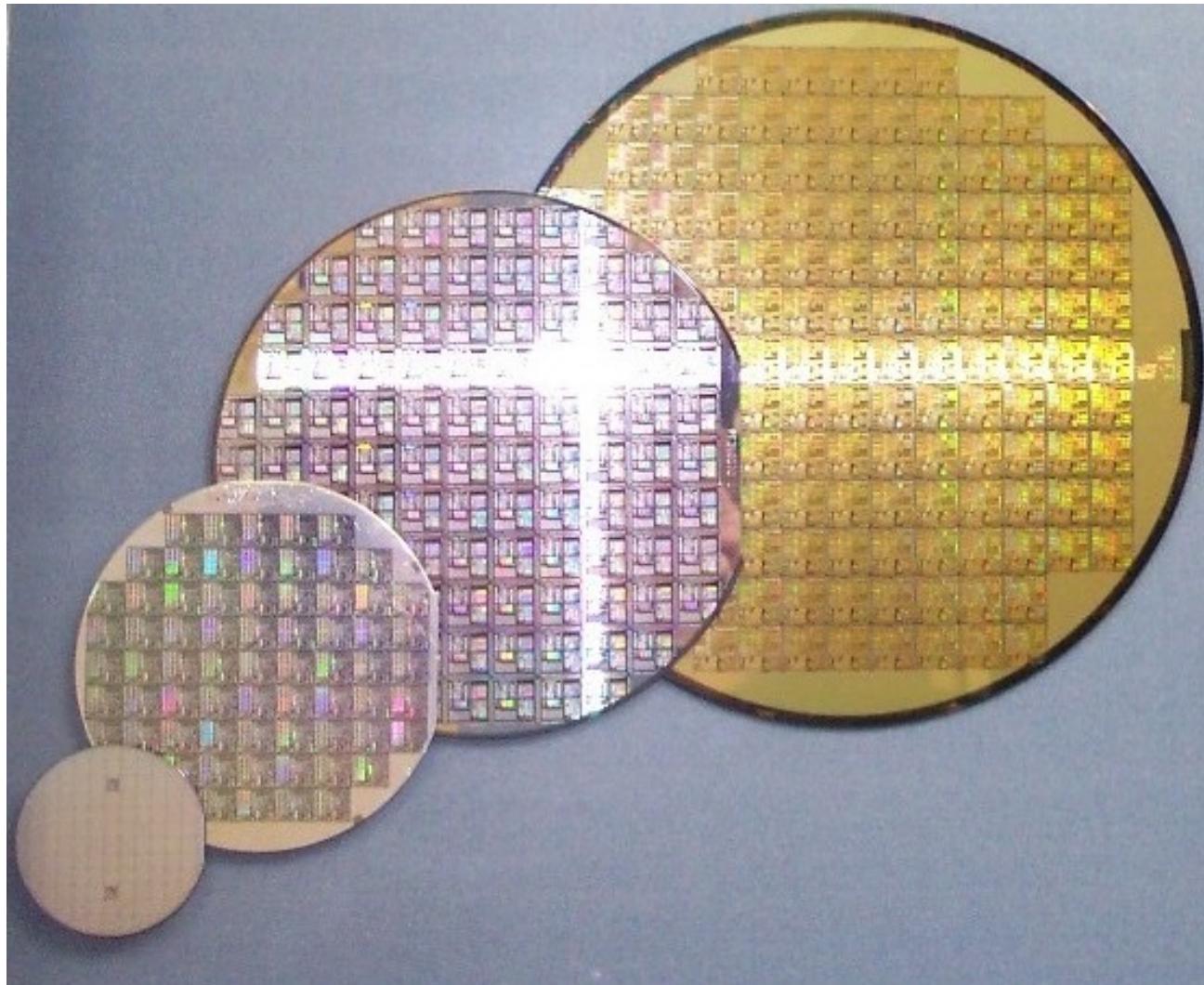


Fabs



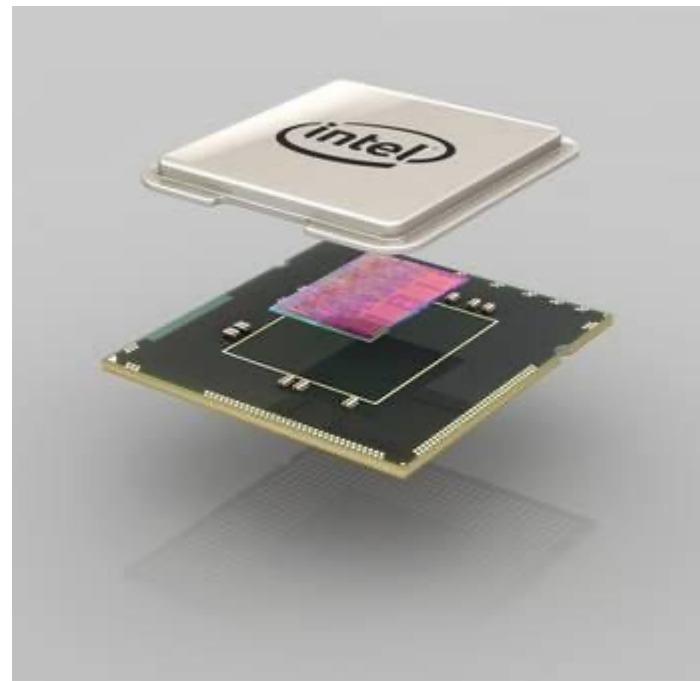
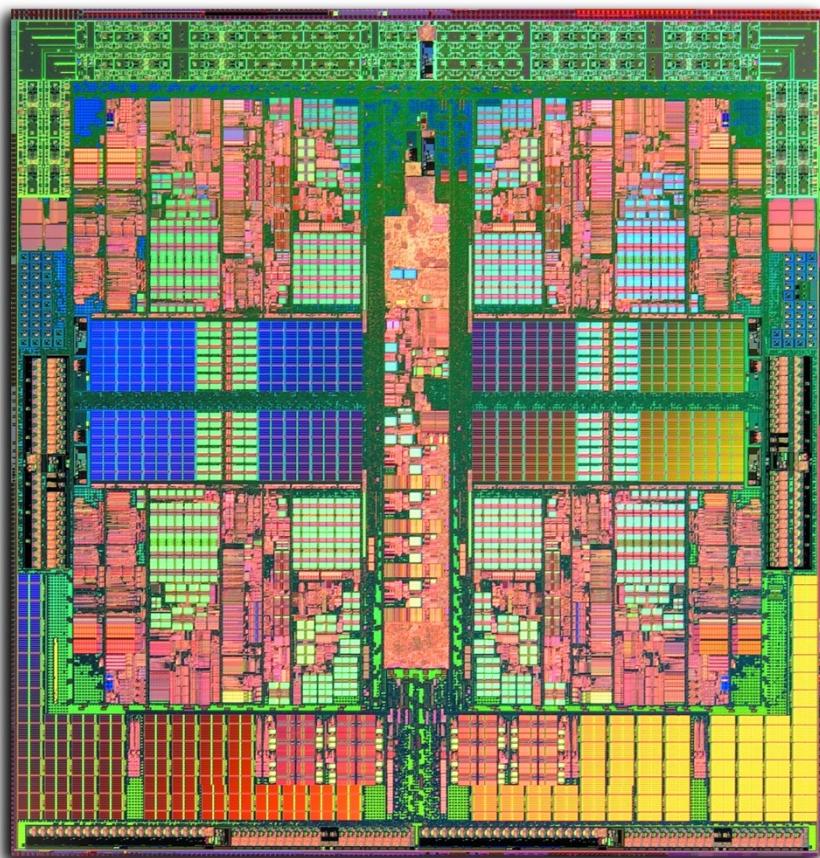


Wafer





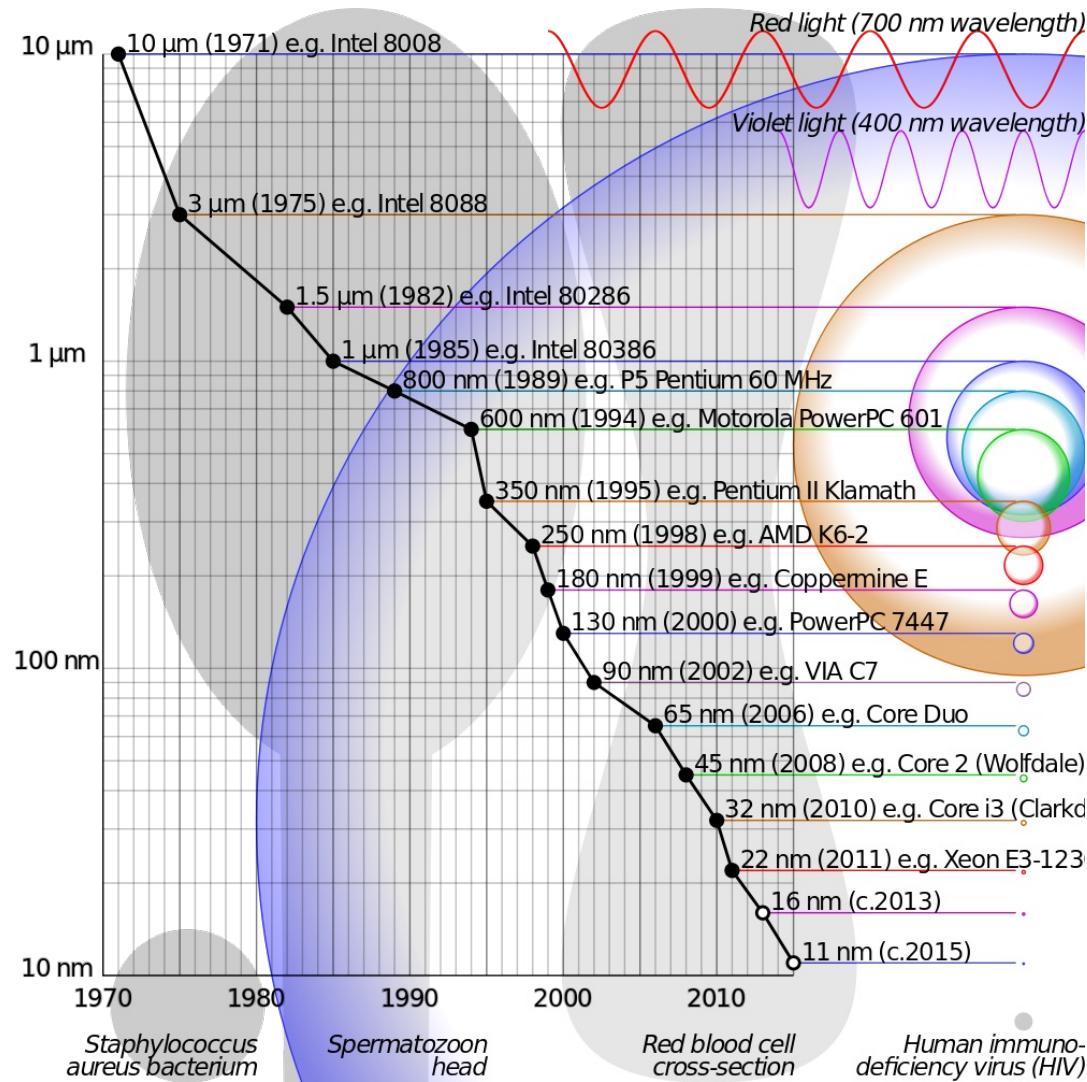
Die





Miniaturizzazione

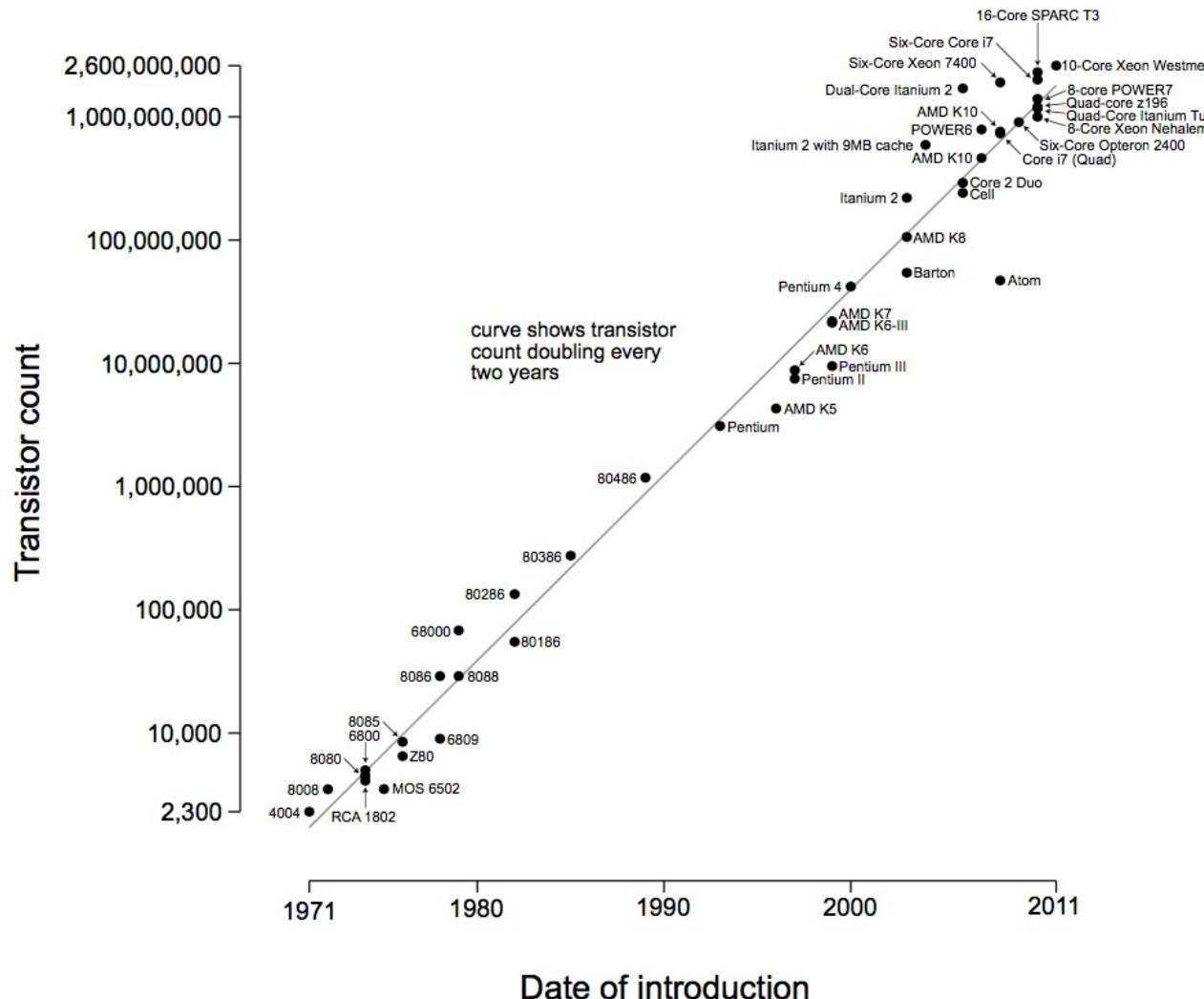
Semiconductor manufacturing processes	
10 μm	— 1971
3 μm	— 1975
1.5 μm	— 1982
1 μm	— 1985
800 nm	— 1989
600 nm	— 1994
350 nm	— 1995
250 nm	— 1997
180 nm	— 1999
130 nm	— 2002
90 nm	— 2004
65 nm	— 2006
45 nm	— 2008
32 nm	— 2010
22 nm	— 2012
14 nm	— est. 2014
10 nm	— est. 2015
7 nm	— est. 2020
5 nm	— est. 2022





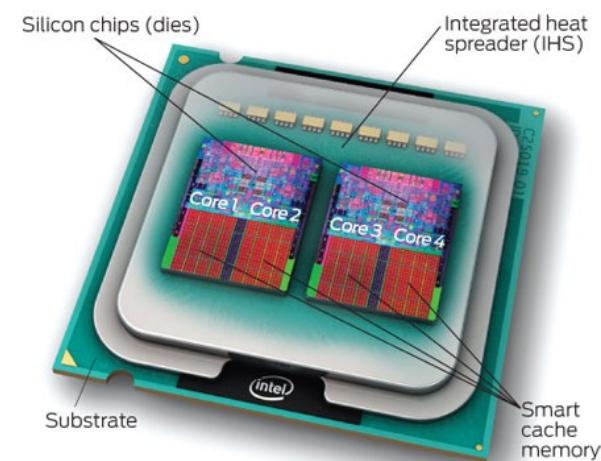
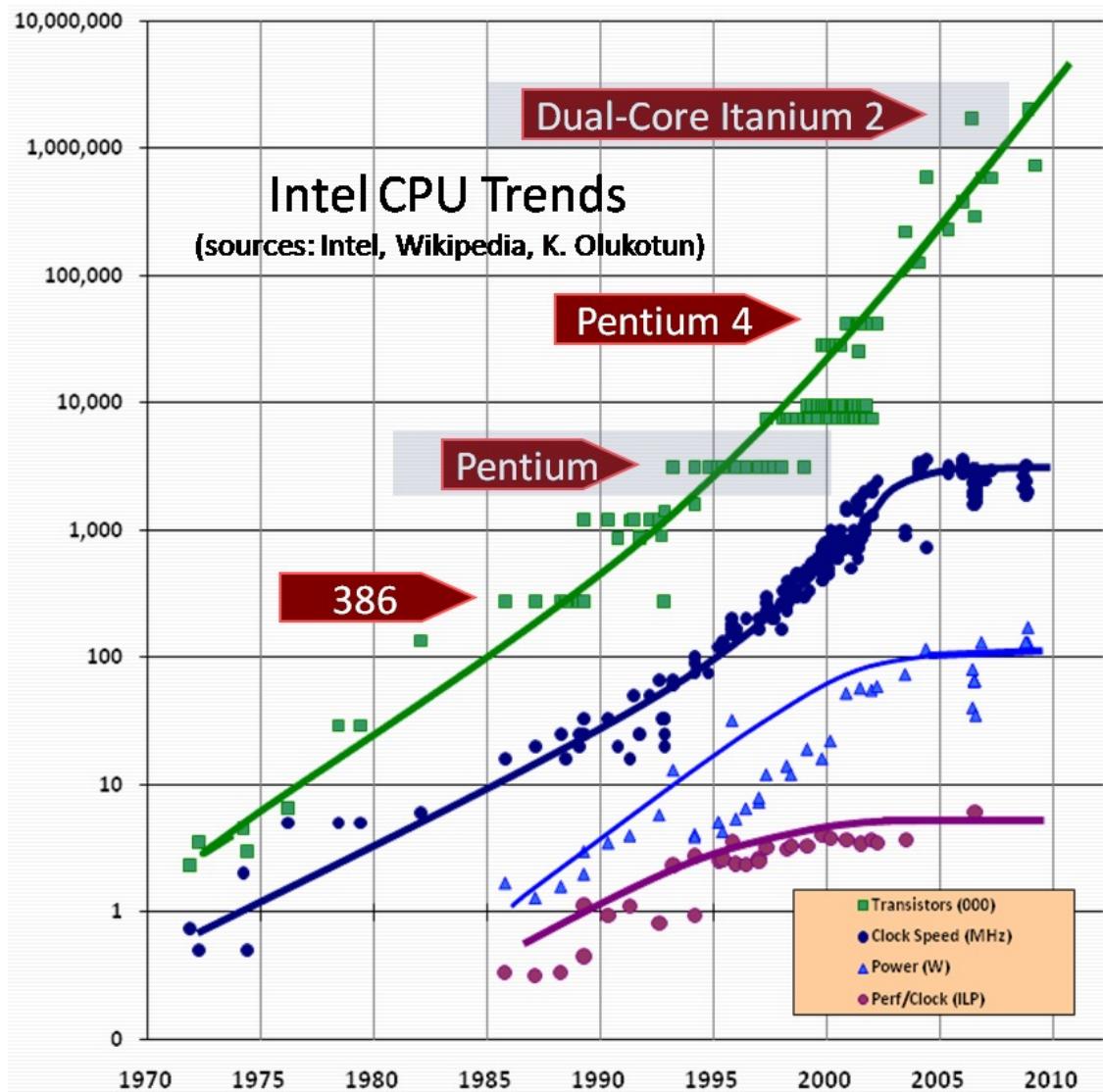
La legge di Moore

Microprocessor Transistor Counts 1971-2011 & Moore's Law





Multi-core



Fonte grafico: "The Free Lunch Is Over"
di Herb Sutter



I computers più veloci al mondo - TOP500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942
4	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
5	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438

Fonte: <https://www.top500.org>

Software

Il software si suddivide in due gruppi principali:

- **Software di base**: sistema operativo, programmi d'utilità, compilatori ed interpreti.
- **Software applicativo**: programmi orientati all'utente come videoscrittura, grafica, programmi multimediali, programmi specifici (contabilità, modelli matematici, ...), giochi, ecc.

Sistema Operativo

Un **sistema operativo** (Operating System - OS) è una **componente software** che si occupa della **gestione delle risorse hardware** mettendo a disposizione le loro **funzionalità alle applicazioni**.

Un OS è anche responsabile di **ospitare ed assistere le applicazioni** mentre eseguono.

Le principali attività di un OS sono:

- Gestione dell'esecuzione dei programmi.
- Gestione degli interrupts hardware (ad esempio la pressione di un tasto sulla tastiera).
- Gestione della memoria.
- Networking.
- Multitasking ed elaborazione parallela.
- Gestione del file system.

Macchina Virtuale

Una **macchina virtuale** (Virtual Machine - VM) è un'applicazione software che emula le funzionalità di un **computer reale** ed è quindi capace di eseguire i programmi come fosse un computer reale.

Le VMs possono essere classificate in due gruppi:

System VMs		
Appl. I	Appl. 2	Appl. 3
Sistema operativo 2		
Macchina Virtuale		
Sistema operativo		

Application VMs
Applicazione
Macchina Virtuale
Sistema operativo

Macchina Virtuale

System VMs:

- anche chiamate hypervisors,
- completa sostituzione delle funzionalità di un computer reale,
- capaci di eseguire un intero sistema operativo (VMware, Parallels Desktop, VirtualBox, ...).

Application VMs:

- offrono le funzionalità sufficienti ad eseguire un'applicazione compatibile con il sistema emulato (es. virtual machine di Java),
- comunemente utilizzate con i linguaggi di programmazione di alto livello (Java & .Net).
- Vantaggi:
 - Utilizzo di ByteCode interpretato e portabile + compilazione Just In Time (JIT)
 - Garbage Collector per la gestione automatica della memoria

Cos'è un linguaggio?

Un **linguaggio** è un **insieme di parole e di metodi di combinazione delle parole usati e compresi da una comunità di persone.**

Un linguaggio è caratterizzato da:

- Alfabeto: insieme di simboli.
- Sintassi: insieme di regole grammaticali per definire delle frasi corrette composte da parole ottenibili dall'alfabeto del linguaggio.
- Semantica: significato delle frasi del linguaggio.
- Pragmatica: quali frasi è opportuno utilizzare a seconda del contesto.



Linguaggi di programmazione

I linguaggi di programmazione vengono utilizzati per specificare il comportamento delle applicazioni.

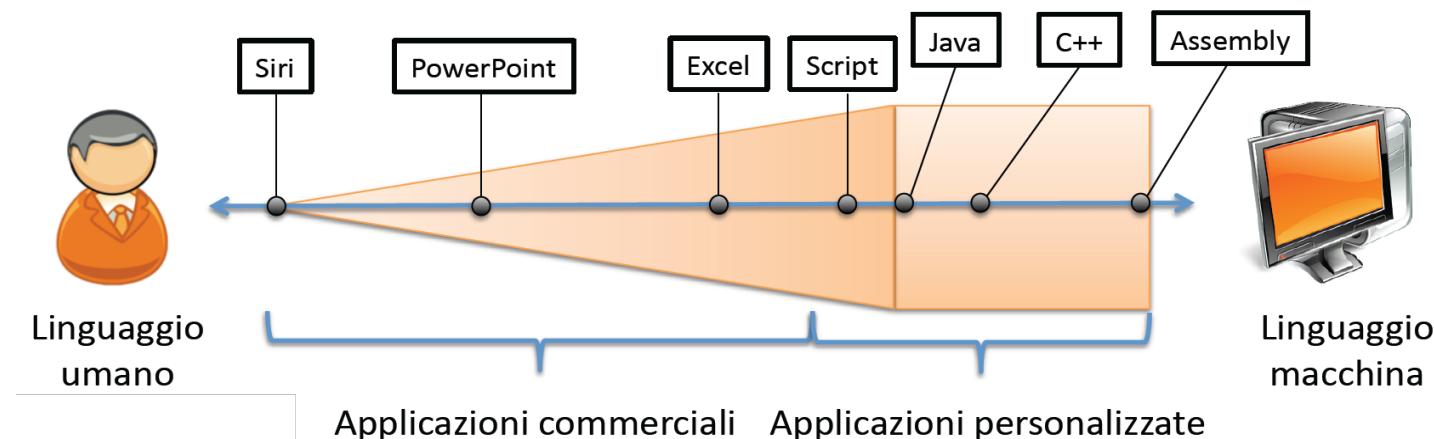
Hanno una sintassi e una semantica e si appoggiano sugli instruction sets dei microprocessori.

Prima generazione: basati sul codice macchina (in binario).

Seconda generazione: codice assembly (simbolici con utilizzo di mnemonici).

Terza generazione: strutturati ed indipendenti dall'hardware (ad es. C, Java).

Quarta generazione: sintassi semplice e specificità di applicazione (ad es. SQL).



Esempi

Linguaggio macchina (binario)

```
0000000010100010000000000011000  
00000000100011100001100000100001  
10001100011000100000000000000000  
10001100111100100000000000000000  
10101100111100100000000000000000  
10101100011000100000000000000000  
00000011110000000000000000000000
```

Assembly

swap:

muli	\$2, \$5, 4
add	\$2, \$4, \$2
lw	\$15, 0(\$2)
lw	\$16, 4(\$2)
sw	\$16, 0(\$2)
sw	\$15, 4(\$2)
jr	\$31

Linguaggio C

```
void swap(int v[ ], int k)  
{  
    int temp;  
    temp = v[k];  
    v[k] = v[k+1];  
    v[k+1] = temp;  
}
```

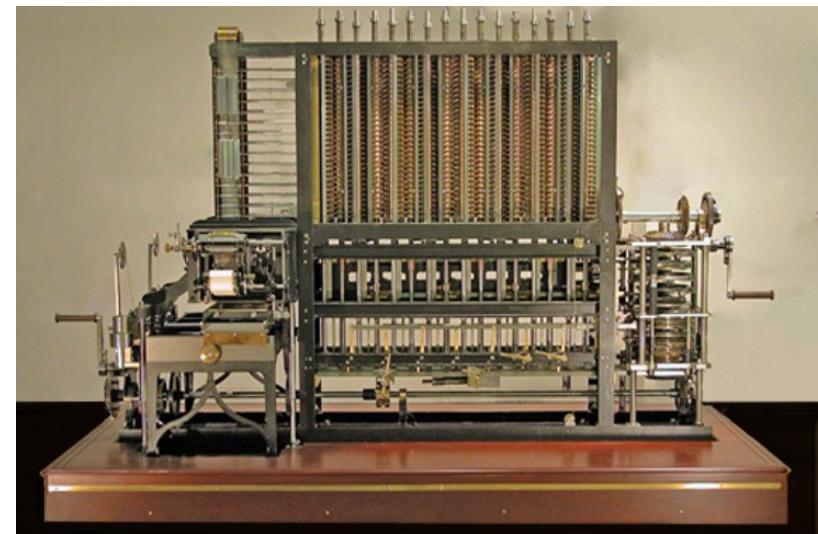


Chi inventò la programmazione

Il primo computer programmabile fu la macchina analitica di Babbage.

La prima programmatrice fu Lady Ada Byron (1815-1852), contessa di Lovelace, figlia del famoso poeta Lord Byron.

Babbage inventò la **notazione meccanica**, forse il primo “linguaggio” di programmazione.





Breve cronologia: gli anni ‘50

1950: Maurice Wilkes usa presso la Cambridge University l'Assembler (symbolic assembly language).

1954: John Backus di IBM crea il linguaggio Fortran (Formula Translation).

1959: La “conference on data system languages” (codasyl) definisce il COBOL (common business oriented language), che introduce l'uso dei blocchi di codice.



Breve cronologia: gli anni '60

- | 1960: Definizione dell'Algol 60 (algorithmic language), che introduce il concetto di ricorsione.
- | 1964: John Kemeny e Tom Kurtz creano il Basic (Beginners All-Purpose Symbolic Instruction Language).
- | 1969: Niklaus Wirth, dell'ETH, scrive il compilatore Pascal, un linguaggio con forte tipizzazione dei dati.



Breve cronologia: gli anni '70

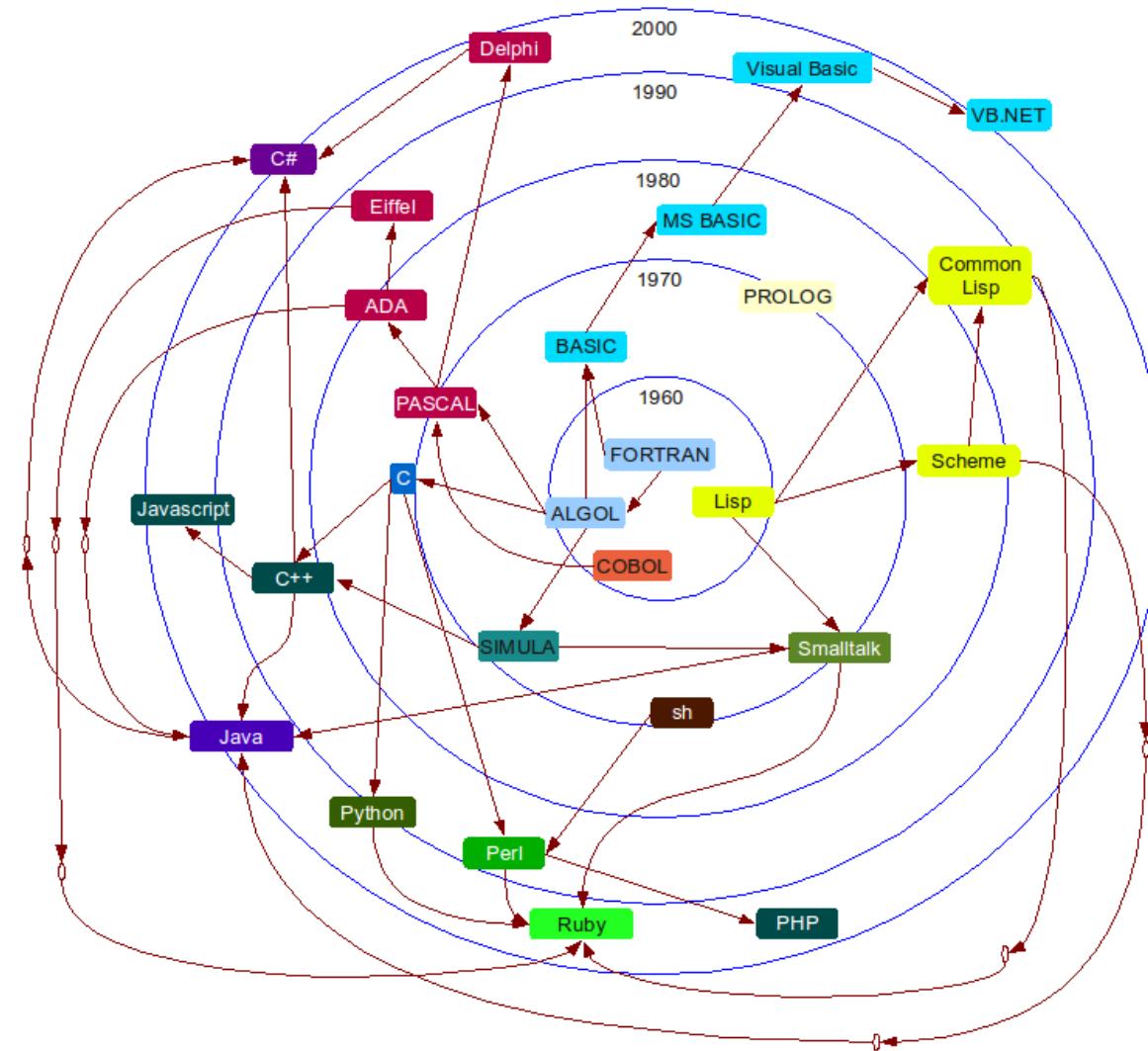
1970: Dennis Ritchie scrive il C, per scrivere UNIX.

1972: Gary Kildall (1942-1994) scrive il PL/I, il primo linguaggio per microprocessore Intel 4004.

1973: Alain Colmerauer sviluppa per l'università francese di Marseilles-Luminy il Prolog (programmation en logique).



Linguaggi di terza generazione



Programmazione procedurale

La programmazione procedurale è un paradigma di programmazione che consiste nel creare dei blocchi di codice sorgente, identificati da un nome e racchiusi da dei delimitatori. Questi sono detti anche sottoprogrammi (in inglese subroutine), procedure o funzioni.

Programmazione orientata agli oggetti

La programmazione orientata agli oggetti (OOP, Object Oriented Programming) è un paradigma di programmazione che permette di definire oggetti software in grado di interagire gli uni con gli altri.

La programmazione ad oggetto fornisce un supporto naturale alla modellazione software degli oggetti del mondo reale o del modello astratto da riprodurre.

È particolarmente adatta nei contesti in cui si possono definire delle relazioni di interdipendenza tra i concetti da modellare.



Il C e il C++

C (... perché venne dopo il linguaggio B):

- Dennis Ritchie, 1969.
- Scritto per riscrivere UNIX in C.
- Linguaggio di “basso” livello usato in applicazioni di “alto” livello.

C++:

- Bjarne Stroustrup, 1983.
- Aggiunge la programmazione ad oggetto al C.

Java

Java è un linguaggio di programmazione **orientato agli oggetti**, specificatamente progettato per essere il più possibile **general-purpose** e il più possibile **indipendente dalla piattaforma di esecuzione**.

Permette di scrivere un programma e di eseguirlo su praticamente qualsiasi tipo di computer (con qualsiasi OS): **write once, run anywhere (WORA)**.

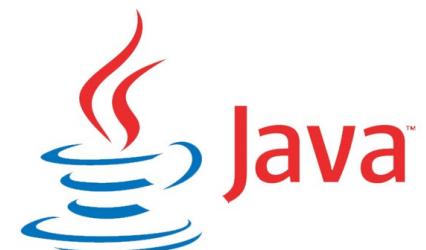
Nel 2014, Java risulta essere uno dei linguaggi di programmazione più usati al mondo, specialmente per applicazioni client-server, con un numero di sviluppatori stimato intorno ai **9 milioni**.

Il linguaggio deriva gran parte della sua sintassi dai **linguaggi C e C++**, ma ha meno costrutti a basso livello e implementa in modo più puro (rispetto al C++) il paradigma object-oriented.

Java

Java è stato originariamente sviluppato dalla Sun Microsystems, che ora è una sussidiaria della Oracle Corporation. La sua prima versione è stata rilasciata nel 1995.

Il nome Java è stato scelto a caso da una lista di suoi potenziali nomi.



Caratteristiche fondamentali di Java

Programmazione in piccolo:

- tipi primitivi, variabili e operatori,
- istruzioni di controllo e di ripetizione (if, switch, for, while, ...).

Programmazione in grande:

- procedure e funzioni,
- arrays, oggetti, classi e interfacce,
- gestione delle eccezioni,
- packages.

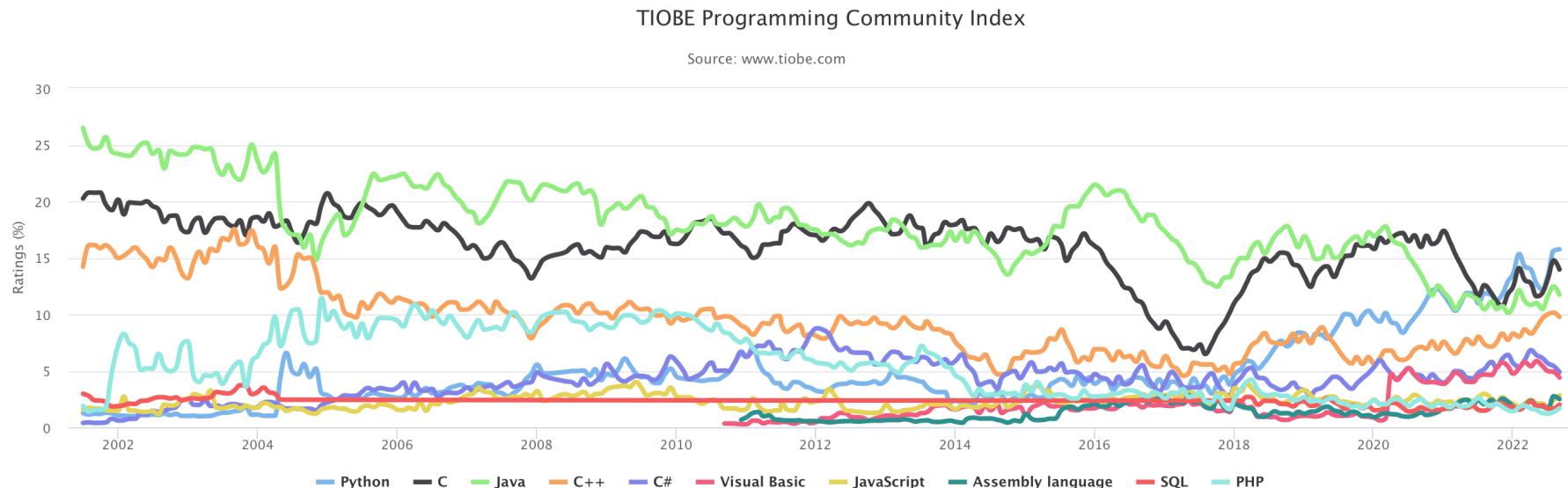
Funzionalità di input e di output

Programmazione concorrente e parallela

Programmazione di interfacce grafiche ad eventi (GUI)



Linguaggi più utilizzati



Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	▲	Python	15.74%	+4.07%
2	1	▼	C	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		C++	9.76%	+2.63%
5	5		C#	4.88%	-0.89%

Fonte: <https://www.tiobe.com/tiobe-index/>

Compilatore

Un **compilatore** è un programma che traduce istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio, normalmente di più basso livello. Questo processo di traduzione si chiama **compilazione**.

I compilatori più comuni traducono linguaggi di terza generazione in assembly specifico per una determinata architettura di processore.

Linguaggi compilati vs. interpretati

L'espressione “**linguaggio compilato**” indica un linguaggio di programmazione implementato di solito tramite un compilatore invece di un interprete.

Compilatore: converte il codice sorgente in codice macchina.

Interprete: esegue direttamente il codice sorgente.

In teoria ogni linguaggio può essere implementato sia con un compilatore, sia con un interprete.

Bytecode e compilatore JIT

Esistono e sono sempre più diffuse **soluzioni miste**, in cui un compilatore converte il codice in una forma intermedia (detta **bytecode**), che viene interpretata o compilata in codice macchina a tempo di esecuzione da un **compilatore just-in-time**.

Codice sorgente

```
public static void main(String[] args) {  
    System.out.println("Hello World!");  
}
```

Bytecode

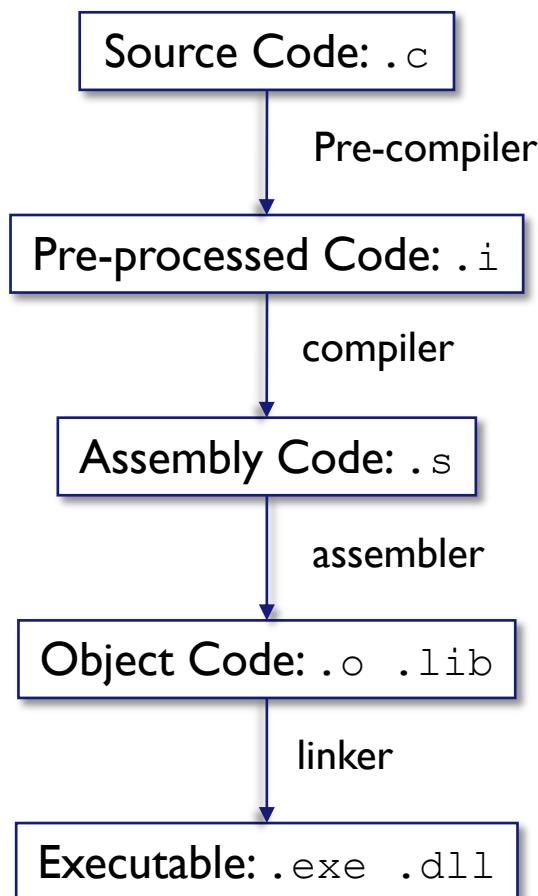
```
public static void main(java.lang.String[]);
```

Code:

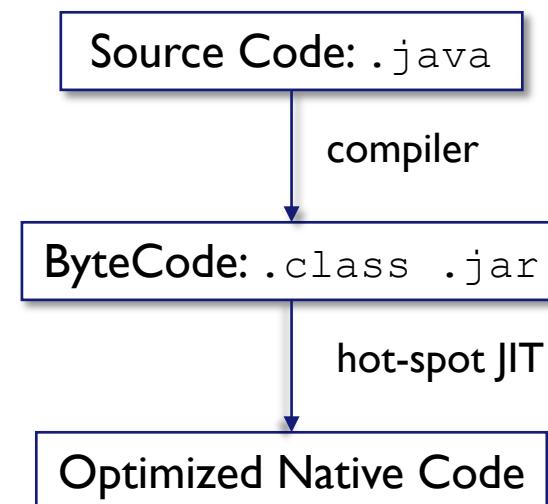
```
0: getstatic      #16 // Field java/lang/System.out:Ljava/io/PrintStream;  
3: ldc           #22 // String Hello World!  
5: invokevirtual #24 // Method java/io/PrintStream.println:(Ljava/lang/String;)V  
8: return
```

Compilatore

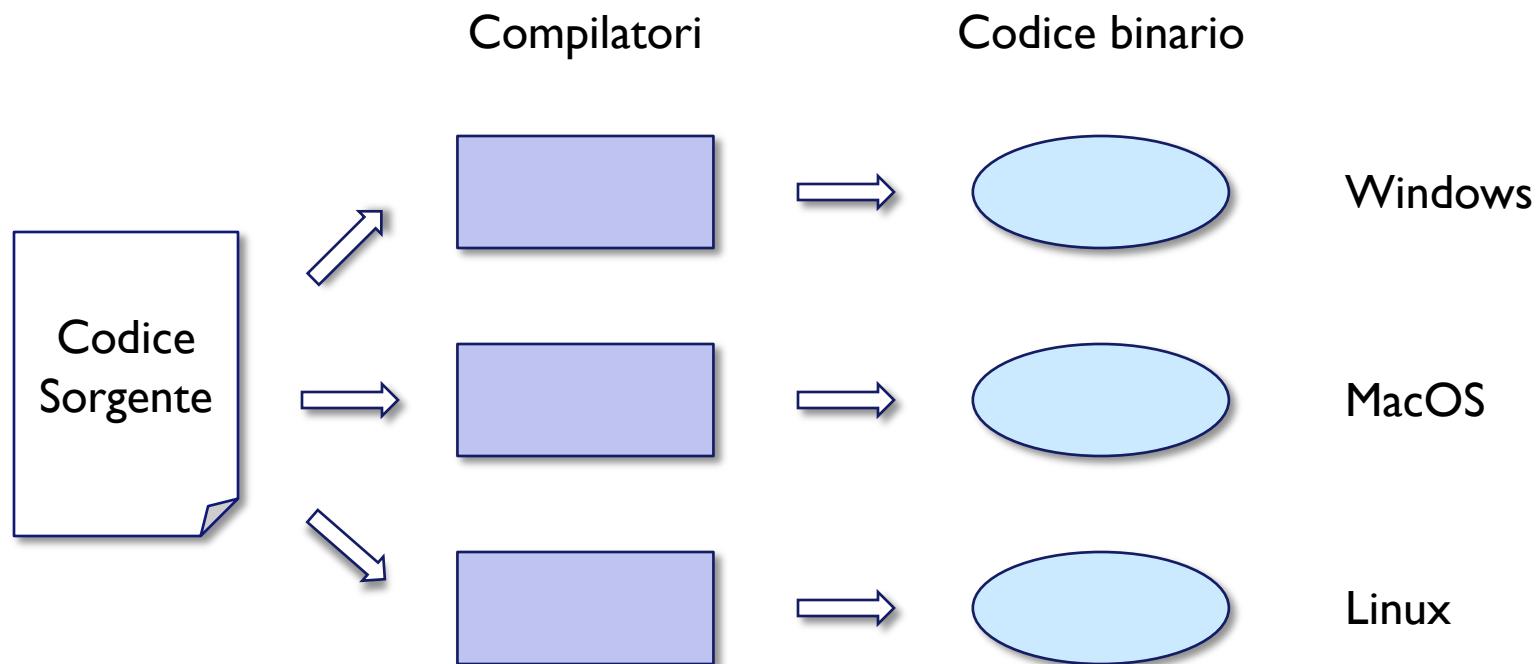
C: executed



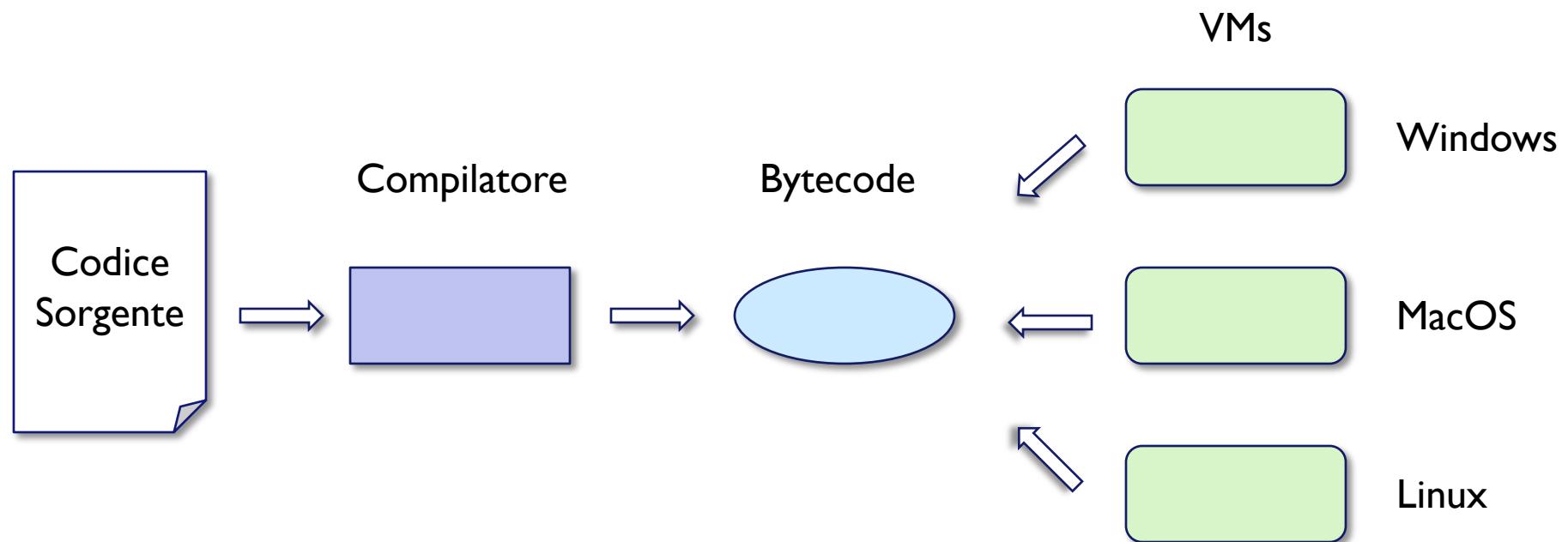
Java: interpreted



Portabilità - linguaggi tradizionali



Portabilità - Java





Ambienti di sviluppo integrato

Un ambiente di sviluppo integrato mette a disposizione un'interfaccia grafica che permette di eseguire tutte le attività necessarie allo sviluppo software.

Tra i più diffusi:

- **Eclipse:** <http://www.eclipse.org/>
- **NetBeans:** <http://www.netbeans.org>
- **IntelliJ Idea:** <http://www.jetbrains.com/idea/>

