

Introduzione

Programma del corso

Scopo del corso: Fornire allo studente le nozioni necessarie alla comprensione delle architetture e del funzionamento degli elaboratori.

Programma del corso:

- Introduzione e storia dei calcolatori
- Rappresentazione digitale dell'informazione
- Architettura del calcolatore e delle periferiche
- Il livello ISA e l'assembly language
- Algebra di Boole
- Il livello della logica digitale
- Architetture a confronto
- Architetture parallele

Esercitazioni e laboratorio

- Laboratorio di assembler (assistito)
- Esercitazioni in aula
- Prove di autovalutazione in aula

Modalità di esame:

- 1) Presentazione di **elaborati assegnati** dal docente. La consegna di elaborati "validi" (verranno assegnati a partire dal mese di Aprile) è obbligatoria e deve avvenire prima dell'iscrizione alla prova orale. *Meccanismo di registrazione su web.*
- 2) **Due prove scritte** nello stesso giorno di cui la prima consiste in una serie di esercizi, mentre la seconda è articolata in una serie di domande (generalmente a risposta aperta) su tutte le tematiche del corso.

Testi e materiale didattico:

- A. S. Tanenbaum, T. Austin. **Architettura dei calcolatori**. Pearson Italia, 6a Edizione, 2013.
- *altri testi consigliati, vedi pagina Web.*
- **Dispense a cura del docente** (*pagina Web*)
- **Video consigliati** (*pagina Web*)

Pagina Web del corso: <http://bias.csr.unibo.it/arc>

Il calcolatore

“Un computer digitale è una macchina in grado di risolvere problemi eseguendo istruzioni appositamente specificate”

Questa semplice definizione pone l'accento su due concetti fondamentali del mondo dell'informatica:

Hardware: indica l'insieme dei dispositivi che compongono il calcolatore. L'hardware si compone di oggetti tangibili: circuiti integrati, memorie, stampanti, ecc.

Software: indica l'insieme delle istruzioni e delle informazioni necessarie per risolvere i problemi a cui il sistema è preposto. Un insieme di istruzioni, **codificate in termini comprensibili a un calcolatore**, sufficienti a risolvere un problema viene detto **programma**.

Il software richiede un apposito hardware per essere eseguito, viceversa l'hardware è pressoché inutile se non si dispone di un apposito software che ne sfrutti le potenzialità per risolvere problemi.

Programmare un calcolatore significa scrivere la sequenza di istruzioni (**algoritmo**) necessaria a risolvere un problema.

Firmware: software integrato in un componente elettronico (es. chip) o dispositivo (es. smartphone), in grado (almeno) di avviare il componente stesso e di farlo interagire con altri componenti.

Macchine a più livelli (1)

L'insieme di istruzioni eseguite direttamente dall'hardware di un calcolatore è detto **linguaggio macchina (assembly)** e normalmente è molto limitato, ad esempio:

- somma due numeri binari
- verifica se un numero è uguale a 0
- copia una informazione da una parte all'altra della memoria

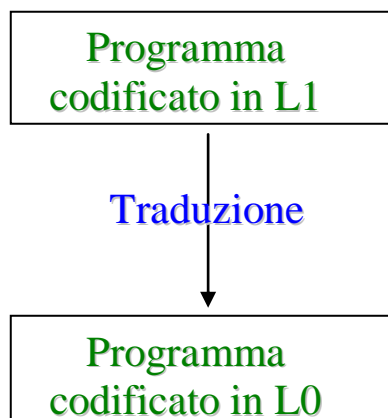
Combinando queste semplici operazioni è possibile risolvere problemi molto complessi.

La limitatezza del linguaggio è dovuta principalmente a due fattori:

- **Costo:** realizzare l'hardware che esegua operazioni complesse è oggi fattibile ma estremamente costoso.
- **Complessità:** la complessità di un computer cresce in modo più che proporzionale rispetto alla complessità del suo linguaggio macchina.

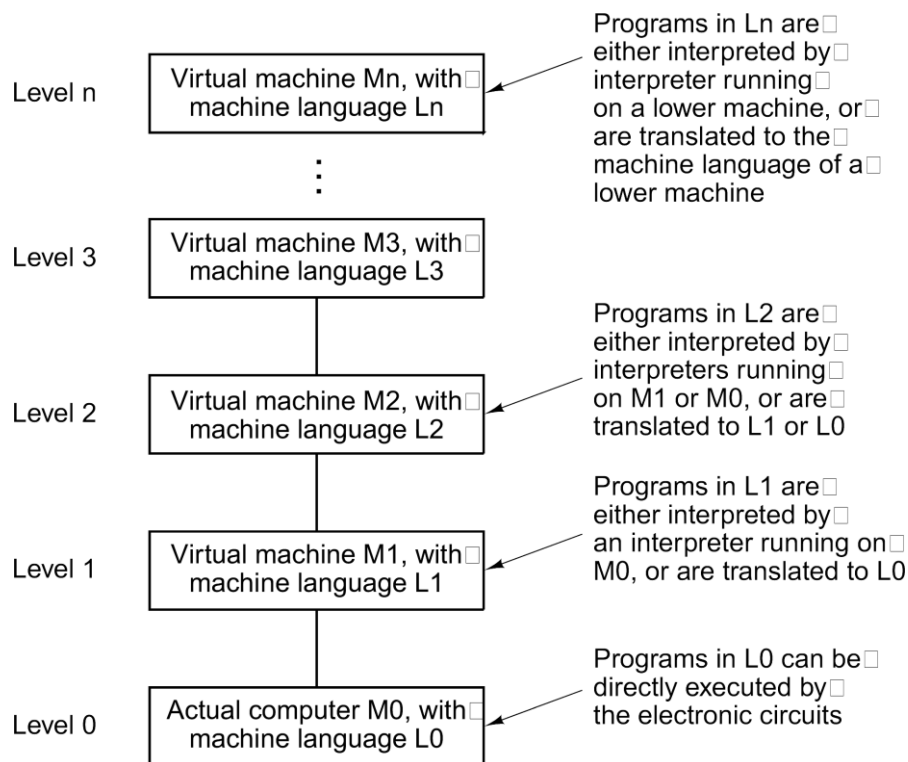
Tuttavia, il linguaggio macchina (L0) non è adatto a un programmatore umano che lo troverebbe noioso e difficile. Per semplificare il processo di programmazione sono quindi nate le macchine a più livelli:

- Viene creato un nuovo linguaggio L1, più facile da utilizzare, che opera su un insieme di istruzioni eseguibili da un **computer virtuale**.
- Nella pratica l'hardware del computer esegue solo istruzioni di L0, quindi è necessario tradurre le istruzioni scritte per il computer virtuale (con linguaggio L1) in istruzioni eseguibili dall'hardware (e quindi in linguaggio L0).



Macchine a più livelli (2)

Per non rendere troppo complessa la traduzione tra i linguaggi utilizzati dai programmatori e il linguaggio macchina l'idea di base delle macchine virtuali può essere replicata più volte.



A ogni livello i è necessario un traduttore, codificato in linguaggio $i-1$, che traduca le istruzioni in linguaggio macchina i in istruzioni in linguaggio macchina $i-1$.

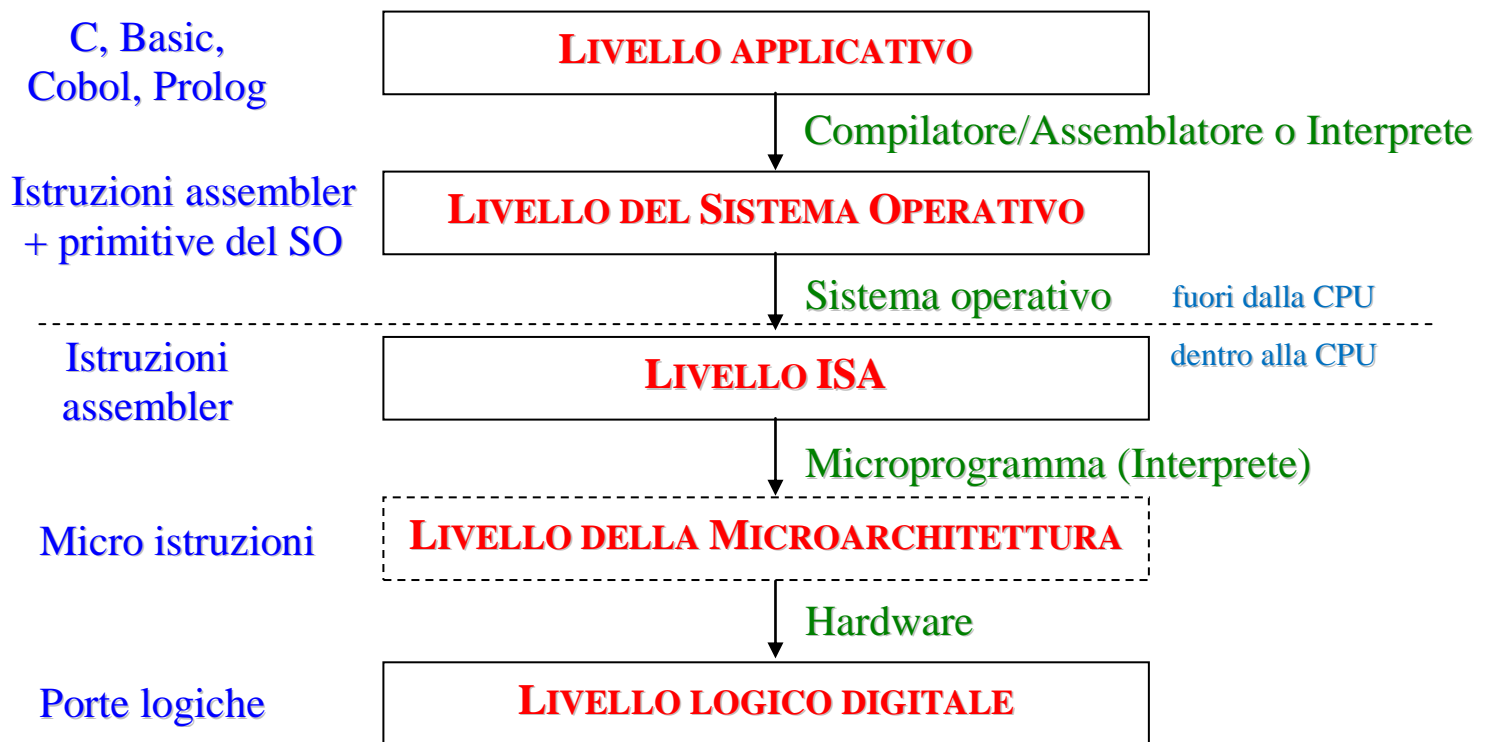
Esistono due tecniche per la traduzione:

Compilazione: il traduttore (**compilatore/assemblatore**) legge il programma in linguaggio macchina i e produce un nuovo programma in linguaggio $i-1$.

Interpretazione: il traduttore (**interprete**) legge il programma in linguaggio macchina i ed esegue direttamente la traduzione sulla macchina virtuale a livello $i-1$.

Macchine a più livelli (3)

I calcolatori moderni si basano su architetture a più livelli (normalmente 5).



Livello logico digitale: rappresenta l'hardware del calcolatore, i suoi componenti sono detti porte o circuiti elementari.

Livello della microarchitettura: la macchina virtuale di questo livello è in grado di effettuare semplici operazioni (microistruzioni) che, ad esempio, prelevano due numeri da **registri** o dalla memoria e li sommano. *Nella pratica implementata tramite l'unità di controllo della CPU.*

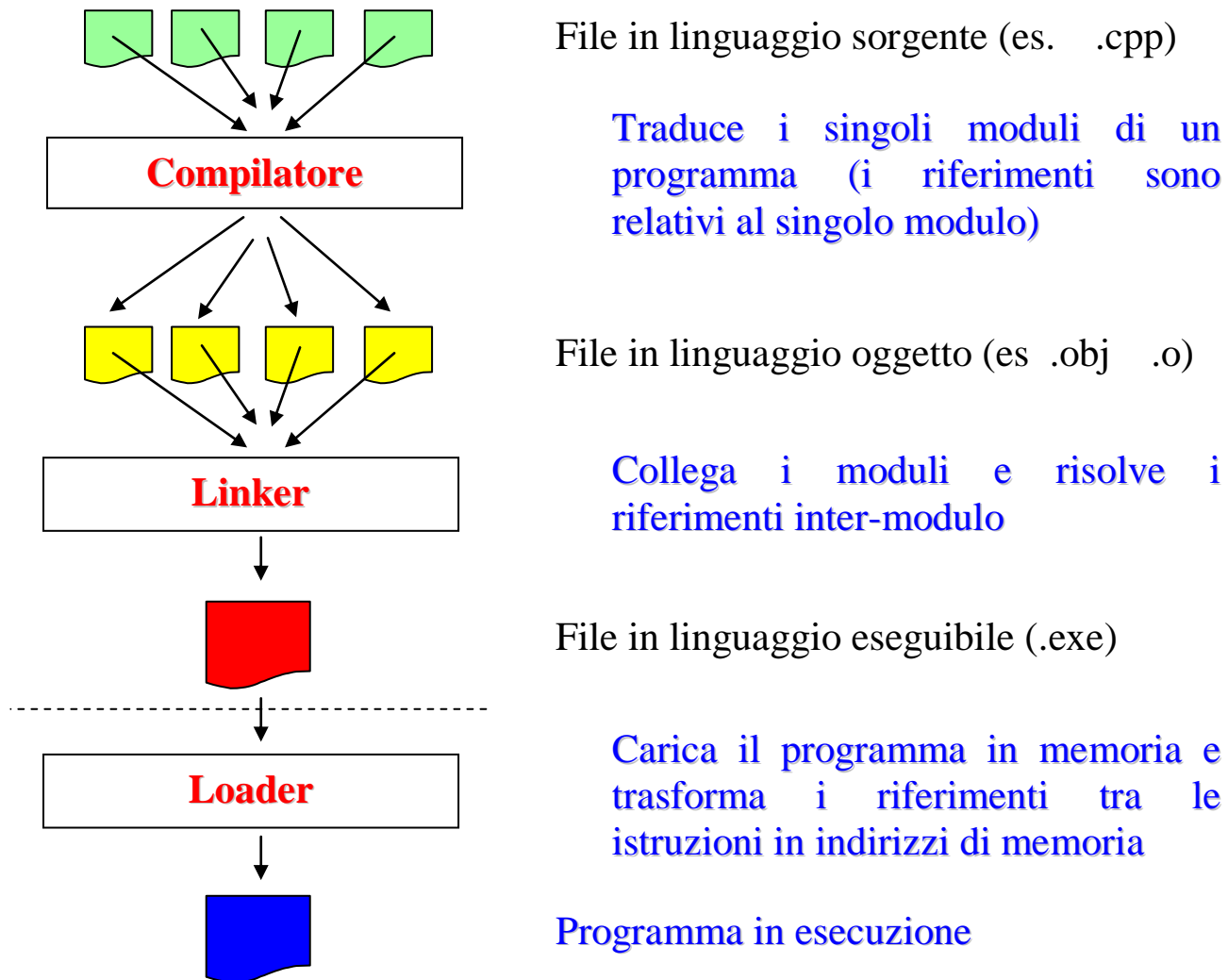
Livello ISA (Istruzioni Assembler): la macchina virtuale di questo livello esegue il linguaggio assembler del calcolatore. La complessità e il numero delle istruzioni presenti a questo livello (**instruction set**) variano da calcolatore a calcolatore e rappresentano una scelta del progettista del sistema. La traduzione verso il livello della microarchitettura viene effettuata da un apposito interprete detto microprogramma.

Macchine a più livelli (4)

Livello del sistema operativo: la macchina virtuale di questo livello ha un livello di astrazione simile a quella sottostante. Tuttavia, sebbene molte delle istruzioni eseguibili a questo livello siano le stesse del livello ISA, vengono fornite un insieme funzionalità più sofisticate quali, ad esempio, una diversa organizzazione della memoria e la possibilità di far funzionare più programmi contemporaneamente.

Livello applicativo: consiste in linguaggi che possono essere utilizzati dai programmatori (Basic, Pascal, C, C++, Java, Lisp, Prolog, ecc.).

Traduzione di programmi C, C++



Il **traduttore** è detto **compilatore** quando il linguaggio sorgente è un **linguaggio di alto livello**. È detto **assemblatore (assembler)** quando il linguaggio sorgente è il **linguaggio assembly**. Nel primo caso il rapporto tra istruzione sorgente e istruzione macchina è **1 a molti**, nel secondo è **1 a 1** infatti l'assembly è una **rappresentazione simbolica del linguaggio macchina** (usa istruzioni simboliche invece che codici numerici per facilitare la programmazione).

Evoluzione delle macchine a più livelli

Il **livello della microprogrammazione** è stato introdotto negli anni '60 per supportare un livello ISA complesso utilizzando un hardware semplice.

- Nel corso degli anni '70 la dimensione del microprogramma andò via via aumentando a causa dell'inserimento di istruzioni sempre più complesse e non supportabili direttamente dall'hardware.
- Alla fine degli anni '70 la dimensione dei microprogrammi era tale da rallentare significativamente il funzionamento dei calcolatori. **Fu quindi rimessa in discussione la reale utilità del livello della microarchitettura.** Nelle **moderne architetture RISC** microprogrammi praticamente assenti (modellazione con circuiti sequenziali).

L'utilità del livello della microprogrammazione è strettamente correlato al livello della tecnologia hardware. Il limite tra hardware e software è in continuo cambiamento e un'istruzione deve essere realizzata in hardware/software in base a considerazioni prestazionali ed economiche.

Il livello del **Sistema Operativo** è stato introdotto negli anni '50 per eliminare le fasi di inattività della CPU dovute all'input/output dei dati (a quei tempi eseguito tramite schede perforate). Le funzioni del SO sono aumentate nel tempo:

- **Gestione della CPU:** più processi possono essere eseguiti contemporaneamente sullo stesso sistema. È il SO che gestisce questa concorrenza (**time sharing**) in base alla priorità dei processi e al loro stato.
- **Gestione della memoria:** permette ai diversi processi di utilizzarla in modo dinamico.
- **Gestione delle periferiche:** permette ai diversi processi di utilizzare le risorse per l'I/O dei dati.

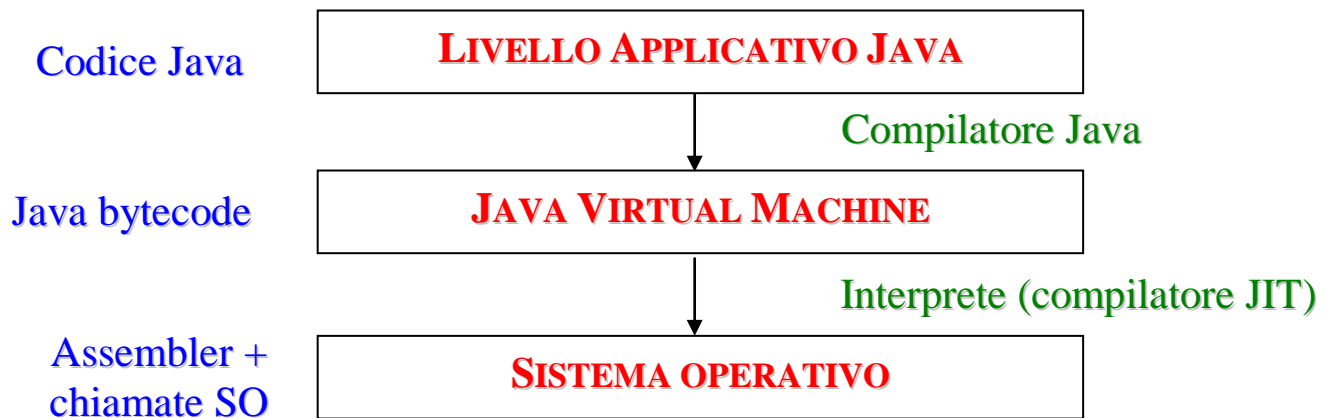
La virtualizzazione oggi

A partire dagli anni '90 iniziano a diffondersi tecnologie di virtualizzazione dell'hardware con l'obiettivo di rendere sistemi operativi e software applicativo “**portabile**”: poter eseguire lo stesso software su calcolatori con hardware, periferiche e sistemi operativi diversi (es. un PC e un Mac). Esempi:

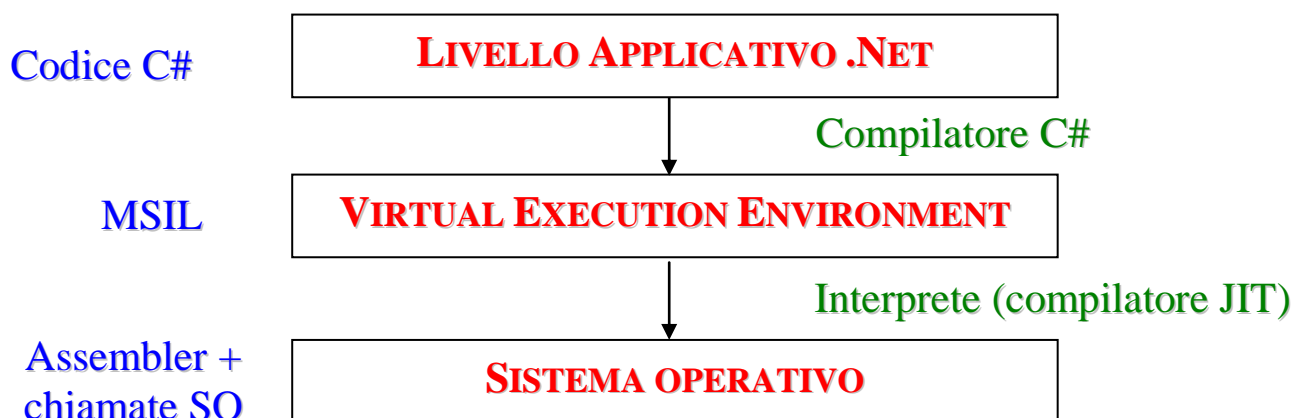
- Virtualizzazione dell'hardware per poter **ospitare sistemi operativi** diversi: es. **VMware**.
 - Il sistema operativo ospite (Guest) può essere installato sopra il sistema operativo ospitante (Host), oppure a partire da “Bare-metal” (nessun sistema operativo Host).
 - Il sistema operativo ospite (Windows, Linux, etc.) vede l'hardware sottostante sempre nello stesso modo.
 - La virtualizzazione della CPU può essere **software-based** o **hardware-assisted**.
 - Si possono lanciare più sistemi operativi contemporaneamente e suddividere tra questi le risorse.
- Macchine virtuali per rendere il software **portabile**: **Java** e **Microsoft .Net**.
 - Il codice sorgente non è compilato direttamente in codice binario (composto da istruzioni ISA), ma in codice **intermedio** indipendente dall'hardware (**Java bytecode**, **MSIL** = Microsoft Intermediate Language).
 - Questo codice intermedio è compreso da una macchina virtuale (**JVM** = Java Virtual Machine, **VES** = Virtual Execution System) installata tra l'hardware e il sistema operativo.
 - Durante l'esecuzione la macchina virtuale traduce il codice intermedio in codice ISA per l'hardware sottostante.
 - La macchina virtuale opera logicamente come un **interprete** (traduzione run-time) ma adotta tecniche di compilazione just-in-time (subito prima dell'esecuzione) e caching (non si ritraduce codice già tradotto) per massimizzare efficienza.

Macchine Virtuali Java e .Net

Java:



.Net (C#):



Unità di misura e prefissi

In ambito informatico si fa spesso riferimento a diverse grandezze (es. tempo di esecuzione, spazio occupato). È quindi necessario avere ben chiare le unità di misura e i prefissi utilizzati per identificarne i multipli.

Prefissi metrici				
Yotta	10^{24}		Milli	10^{-3}
Zetta	10^{21}		Micro	10^{-6}
Exa	10^{18}		Nano	10^{-9}
Peta	10^{15}		Pico	10^{-12}
Tera	10^{12}		Femto	10^{-15}
Giga	10^9		Atto	10^{-18}
Mega	10^6		Zepto	10^{-21}
Kilo	10^3		Yocto	10^{-24}

Prefissi binari	
Yotta byte YB	$2^{80}=1.208.925.819.614.629.174.706.176$
Zetta byte ZB	$2^{70}=1.180.591.620.717.411.303.424$
Exa byte EB	$2^{60}=1.152.921.504.606.846.976$
Peta byte PB	$2^{50}=1.125.899.906.842.624$
Tera byte TB	$2^{40}=1.099.511.627.776$
Giga byte GB	$2^{30}=1.073.741.824$
Mega byte MB	$2^{20}=1.048.576$
Kilo byte KB	$2^{10}=1.024$

ATTENZIONE: i prefissi del sistema binario non corrispondono ai prefissi del sistema decimale. Secondo lo standard IEEE 1541, ai prefissi binari andrebbe aggiunta una ‘i’ per evitare confusione; ad esempio 1024 Bytes andrebbero indicati come 1 KiB (nome esteso “kibibyte”), ma nella pratica pochi hanno seguito tale raccomandazione.

Storia dei calcolatori

Anno	Nome	Costruttore	Commenti
1642	Macchina calcolatrice	Blaise Pascal	Dispositivo meccanico in grado di eseguire addizioni e sottrazioni
1672	Macchina calcolatrice	Von Leibniz	Dispositivo meccanico in grado di eseguire anche moltiplicazioni e divisioni
1834	Analytical Engine	Babbage	1° tentativo di costruire un calcolatore programmabile
1936	Z1	Zuse	1° calcolatore funzionante a relè
1943	COLOSSUS	Gov. Britannico	Primo calcolatore elettronico
1944	Mark I	Aiken	1° calcolatore americano general purpose
1946	ENIAC	Eckert/Mauchley	La storia dei calcolatori moderni comincia da questa macchina
1949	EDSAC	Wilkes	1° calcolatore con programma in memoria
1951	Whirlwind I	MIT	1° calcolatore in tempo reale
1952	IAS	Von Neumann	I calcolatori attuali usano la stessa struttura
1953	IBM 701	IBM	1° calcolatore IBM
1960	PDP-1	DEC	1° minicalcolatore (50 esemplari)
1961	1401	IBM	Calcolatore di grande successo per applicazioni commerciali in piccole aziende
1962	7094	IBM	La macchina più usata per il calcolo scientifico all'inizio degli anni '60
1963	B5000	Burroughs	1° calcolatore progettato per supportare linguaggi ad alto livello
1964	360	IBM	1° famiglia di calcolatori
1964	6600	CDC	1° calcolatore scientifico
1965	PDP-8	DEC	1° calcolatore largamente diffuso (50.000 esemplari)
1970	PDP-11	DEC	Domina il mercato dei minicalcolatori negli anni '70
1974	8080	Intel	1° calcolatore general purpose a 8 bit su un solo chip
1974	CRAY-1	Cray	1° supercalcolatore vettoriale
1978	VAX	DEC	1° super-mini a 32 bit
1981	IBM PC	IBM	Inizia l'era dei PC
1985	MIPS	MIPS	Prima macchina commerciale RISC
1987	SPARC	Sun	1° workstation RISC basata sull'architettura SPARC
1990	RS6000	IBM	1° calcolatore superscalare

Generazione 0: I calcolatori meccanici

- Il primo a costruire macchine calcolatrici automatiche fu lo scienziato francese **Blaise Pascal** che, nel **1642 (a 19 anni)**, realizzò un dispositivo (*Pascalina*) a ingranaggi azionati per mezzo di una manovella e in grado di effettuare addizioni e sottrazioni.
 - Rispetto al pallottoliere implementava meccanicamente meccanismi di riporto.
 - L'obiettivo era aiutare il padre nel suo lavoro di esattore delle tasse.

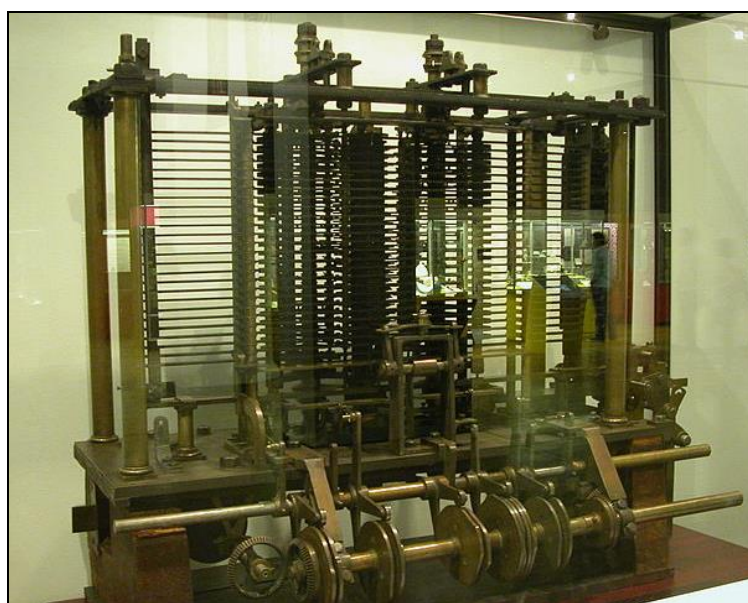


- In seguito, nel **1672**, **Gottfried Von Leibniz** (tedesco), inventò una macchina in grado di fare anche moltiplicazioni e divisioni.
 - La novità, rispetto alla pascalina, era costituita da un traspositore che permetteva di memorizzare un numero per sommarlo ripetutamente.



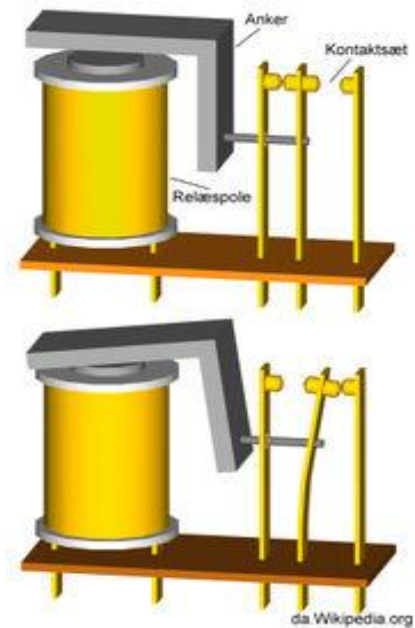
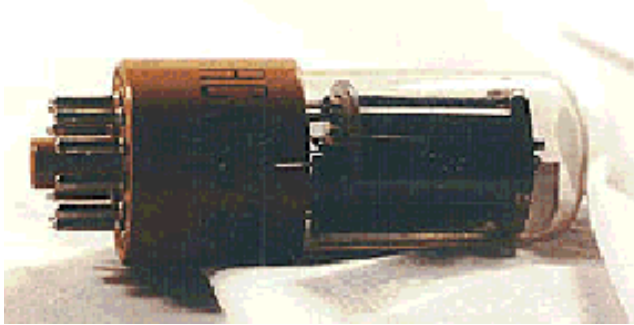
I calcolatori meccanici (2)

- Nel 1834, **Charles Babbage** (britannico), progettò l'**analytical engine** il primo dispositivo in grado di calcolare funzioni diverse in base al tipo di programma che veniva caricato (**Macchina di Turing** completa).
 - Il sistema, completamente meccanico, consisteva di quattro parti, la memoria (store), l'unità di calcolo (mill), l'input (tramite schede perforate) e l'output (tramite schede perforate).
 - Per una serie di problemi realizzativi e di finanziamento la realizzazione non venne mai completata. Recentemente è stato dimostrato attraverso prototipi che i dispositivi progettati da Babbage potevano realmente funzionare.
 - Per produrre il software necessario Babbage collaborò con **Ada Augusta Lovelace**: la **prima programmatrice** della storia. Tuttavia questi programmi rimasero teorici e non furono mai eseguiti.



Generazione I: Valvole termoioniche e relè

Sono i primi calcolatori elettrici e digitali. Valvole termoioniche, relè e transistor si basano sullo stesso principio: permettono di memorizzare un'informazione binaria (0 o 1).



COLOSSUS (1943) nato in UK per l'esigenza di decodificare i messaggi criptati (tramite la macchina Lorenz SZ40/42) tra Hitler e i capi di stato maggiore durante la seconda guerra mondiale.

Rappresenta un'evoluzione della macchina "Bomba" a cui lavorò Alan Turing (a Bletchley Park) per decifrare i messaggi di Enigma. Bomba era infatti una macchina elettromeccanica progettata per un compito specifico, mentre Colossus un calcolatore programmabile basato su valvole termoioniche.

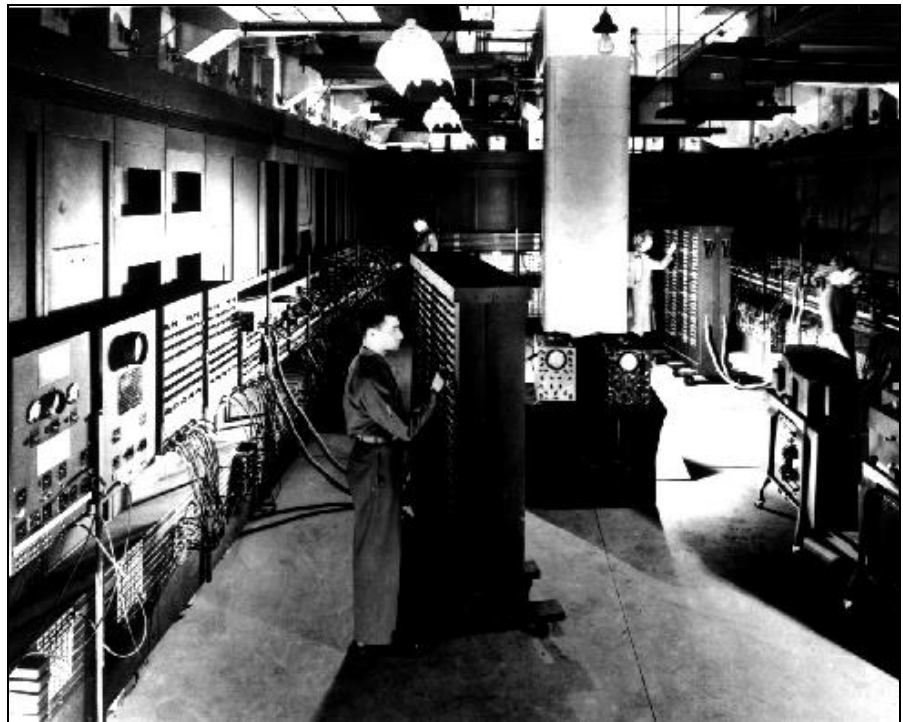


Valvole termoioniche e relè (2)

ENIAC (Electronic Numerical Integrator and Computer - 1946) costituito da 18000 valvole e 3000 relè, nasce negli Stati Uniti per calcolare le tabelle per il puntamento dell'artiglieria pesante.

Pesava 30 tonnellate, era basato su aritmetica decimale, possedeva 20 registri ed era programmabile tramite 6000 interruttori multi-posizione e da una moltitudine di cavi.

L'input/output era possibile attraverso schede perforate.



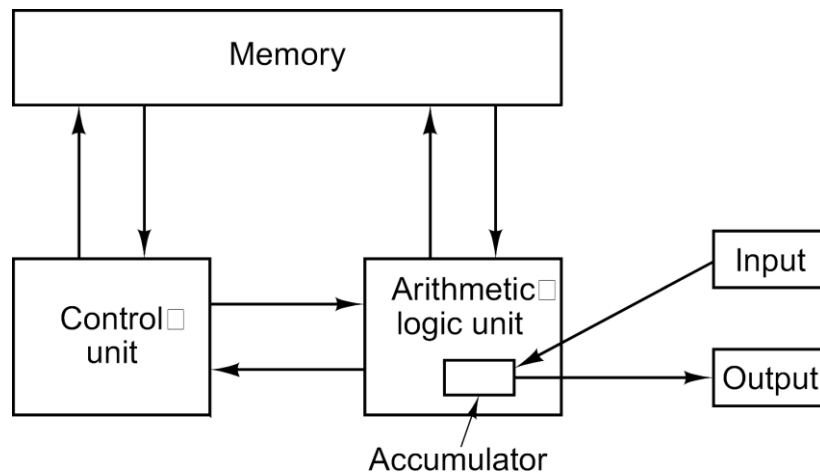
Valvole termoioniche e relè (3)

Calcolatori in tempo di Guerra

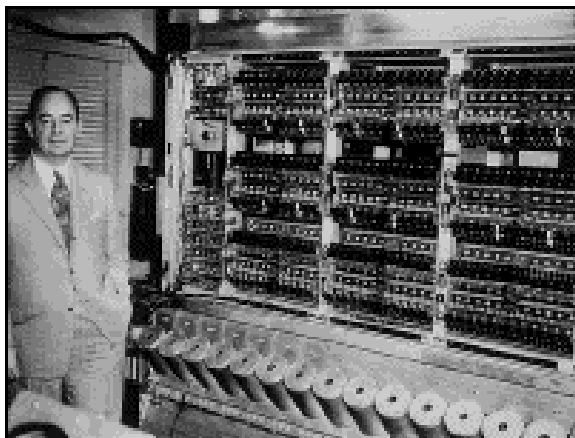
Name	Date	Numeral system	Computing mechanism	Programming	Turing complete
Zuse Z3 (Germany)	1941	Binary floating point	Electro-mechanical	Program-controlled by punched 35 mm film stock (but no conditional branch)	In theory (1998)
Atanasoff–Berry Computer (US)	1942	Binary	Electronic	Not programmable—single purpose	No
Colossus Mark1 (UK)	1944	Binary	Electronic	Program-controlled by patch cables and switches	No
Harvard Mark I – IBM SCC (US)	1944	Decimal	Electro-mechanical	Program-controlled by 24-channel punched paper tape (but no conditional branch)	Debatable
Colossus Mark2 (UK)	1944	Binary	Electronic	Program-controlled by patch cables and switches	In theory (2011)
Zuse Z4 (Germany)	1945	Binary floating point	Electro-mechanical	Program-controlled by punched 35 mm film stock	Yes
ENIAC (US)	1946	Decimal	Electronic	Program-controlled by patch cables and switches	Yes

Valvole termoioniche e relè (4)

- **IAS (1952)**, denominata **macchina di Von Neumann**, rappresenta uno dei più importanti punti di riferimento dell'ingegneria informatica poiché la sua architettura è tuttora alla base della maggior parte dei calcolatori digitali.



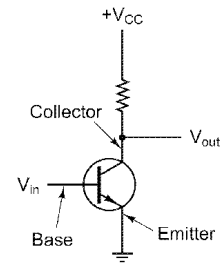
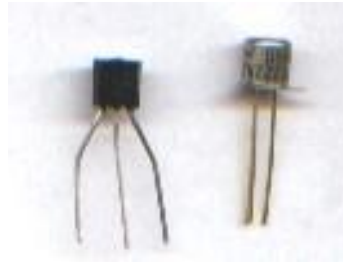
Idea fondamentale di Von Neumann è quella di memorizzare **anche i programmi** (e non solo i dati) **in forma numerica** nella memoria riducendo drasticamente i tempi di programmazione tramite interruttori.



- **IBM 701 (1953)** è il primo computer (2048 parole da 36 bit) prodotto da **IBM**. Nei dieci anni seguenti IBM divenne leader del mercato con una serie di calcolatori a valvole termoioniche via via più potenti (IBM 704, IBM 709).

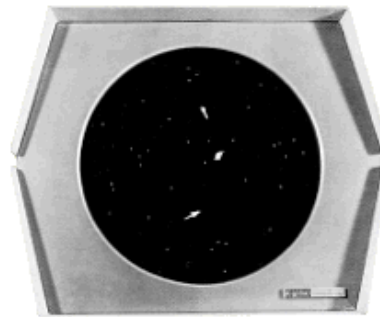
Generazione II: I transistor

Il transistor fu inventato nel 1948 ai Bell Labs e fruttò ai suoi inventori il premio Nobel per la fisica.

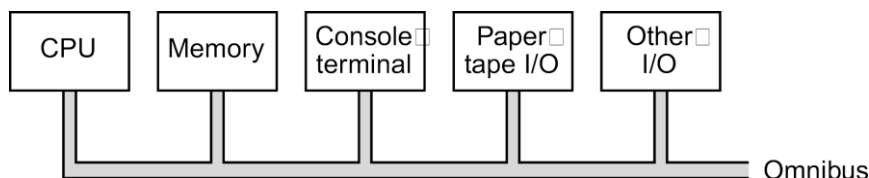


- **PDP-1** (1960), prodotto dalla DEC, rappresenta il primo **minicalcolatore**. Il progetto basa il suo punto di forza non tanto sulla potenza ma sul prezzo. Il PDP-1 era infatti oltre 10 volte meno costoso (120K\$) della macchina più potente di quel periodo (IBM 7090, alcuni M\$), ma solo due volte più lento. Ne furono vendute decine.

Sul PDP-1, che era dotato di display grafico, alcuni studenti dell'MIT programmarono il primo videogioco della storia: **Spacewar**



- **PDP-8** (1965). Molto più economico del PDP-1: 16K\$. Presenta una grande innovazione: il **bus**. Un **bus** è un insieme di fili usato per collegare i componenti di un calcolatore. Ne furono venduti 50000.

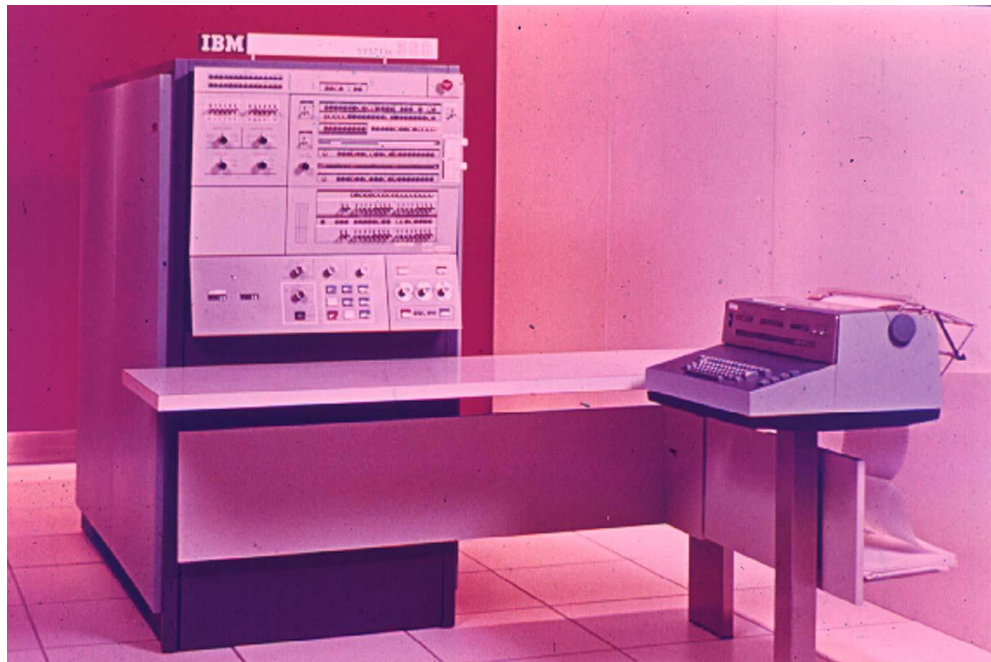


- **CDC 6600** (1964) realizzato dalla CDC (Control Data Corporation) rappresenta la prima **macchina parallela** della storia, destinata al calcolo. Al suo interno coesistevano diverse unità funzionali preposte a compiti diversi (addizioni, sottrazioni, divisioni, ecc.). Al momento della sua uscita sul mercato CDC 6600 era dieci volte più veloce della macchina più potente di quei tempi.

Generazione III: I circuiti integrati

I circuiti integrati, inventati nel 1958 da Robert Noyce, permisero di inserire dozzine di transistor su una singola piastrina di silicio e quindi favorirono la costruzione di calcolatori più piccoli, più veloci e meno costosi.

- **IBM 360** (1964): rappresenta la prima famiglia di calcolatori, esistevano infatti 4 modelli con prestazioni via via crescenti. Un'innovazione molto importante introdotta con questo computer è la possibilità di avere più programmi in memoria (**multiprogrammazione**) cosicché, mentre si aspettava il completamento dell'input/output, si poteva eseguire un altro programma. Inoltre, la famiglia IBM 360 permetteva l'emulazione dei modelli precedenti IBM (1401 e 7094), *grazie alla microprogrammazione*, in modo eliminare il problema di riscrittura del codice nel passare ai nuovi calcolatori. Già allora esisteva il **problema della compatibilità**.



Generazione IV: Very Large Scale Integration

Dagli inizi degli anni '80 i progressi della tecnologia hanno permesso di integrare su un singolo chip un numero sempre crescente di transistor (attualmente milioni) riducendo enormemente il costo dei calcolatori fino al punto in cui anche un singolo individuo poté possederne uno. Comincia così l'era dei **personal computer** (PC).

IBM 5150 (IBM PC - 1981): è il primo personal computer. Dotato del sistema operativo MS-DOS raggiungerà velocemente una diffusione enorme e sarà seguito da un'ampia famiglia di processori. IBM decide di pubblicare schemi e dettagli dell'architettura per semplificare il progetto di schede di espansione: questo *da il via all'industria dei cloni compatibili IBM*.



Il sistema operativo MS-DOS fu fornito da una (allora *piccola*) azienda: la **Microsoft Corporation** di proprietà di Bill Gates.



Very Large Scale Integration (2)

Altre società tra cui **Commodore**, **Atari**, **Sinclair** ed **Apple** realizzarono personal computer non basati su chip Intel. L'unico oggi sopravvissuto è Apple grazie al successo del Macintosh il primo computer con una GUI (Graphical User Interface) simile a quelle odierne.



Il Commodore 64 è il modello di computer più venduto al mondo: nel 1986 furono venduti più di 10 milioni di esemplari in tutto il mondo. In totale, ne sono stati venduti nel mondo oltre **30 milioni di esemplari**: record che, con tutta probabilità non verrà mai più superato.

- Il **primo portatile fu l'Osborne-I** con un peso (non trascurabile) di 11 Kg. A breve **Compaq** produsse il primo clone di PC IBM portatile e divenne leader di tale mercato.
- Intel continuò a produrre CPU sempre più potenti: degno di nota il **386 del 1985**. Gli attuali microprocessori sono ancora fortemente basati su questa architettura.
- Un passo in avanti, dovuto principalmente al passaggio **dell'architettura da 32 a 64 bit** fu fatto nel 1992 quando DEC introdusse **Alpha**. Solo recentemente però si sta assistendo al passaggio 32 → 64 bit.



Generazione V:

Computer invisibili e computazione onnipresente (ubiquitous computing)

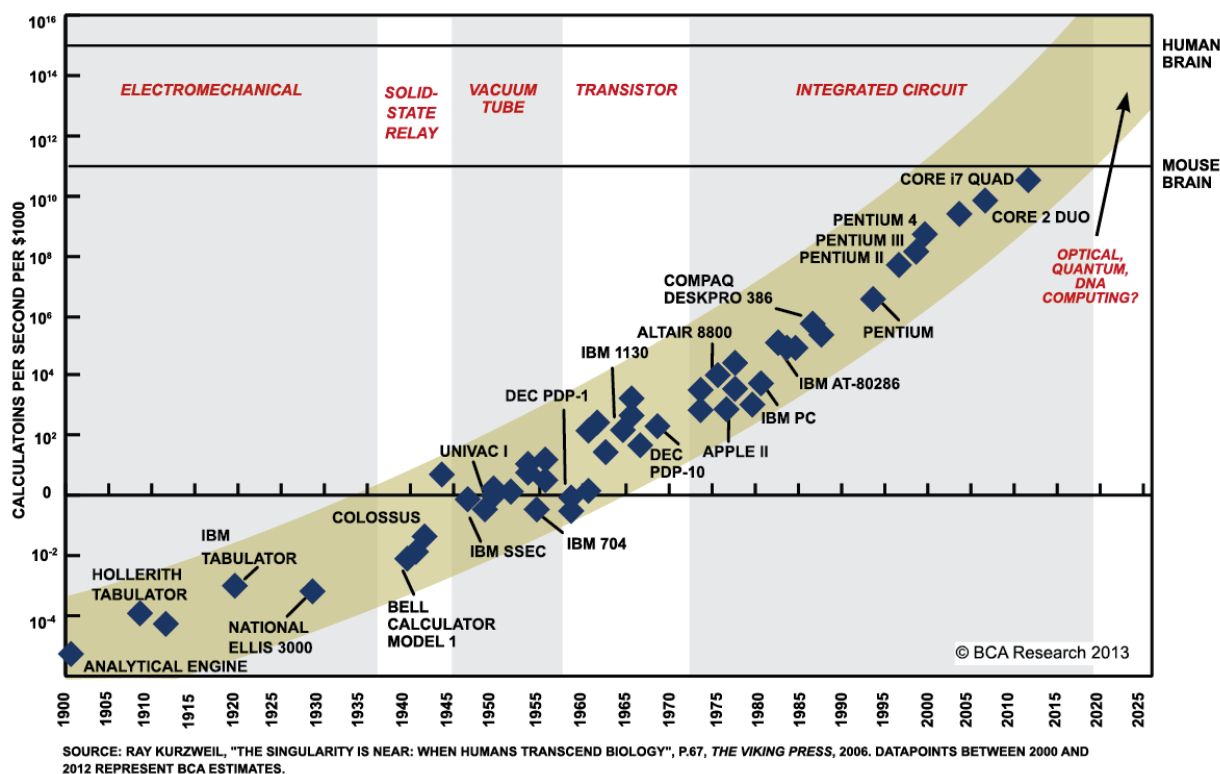
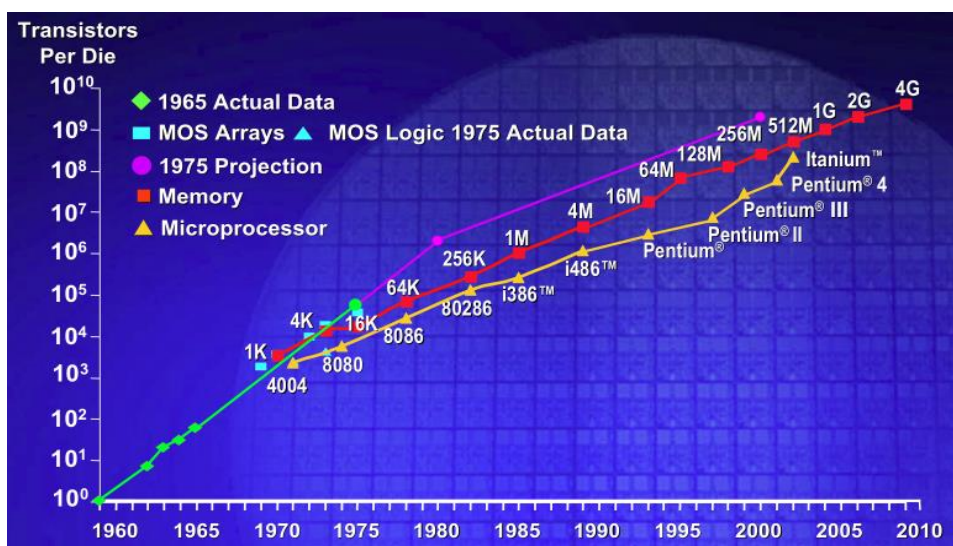
Nel 1981 il **governo giapponese** annuncio di voler stanziare 500 M\$ per aiutare società locali nella realizzazione della V generazione di computer basati su intelligenza artificiale. Il progetto però fallì essendo probabilmente un'idea visionaria non possibile con la tecnologia esistente.

Quella che possiamo oggi definire la V generazione di computer è legata soprattutto alle **forti riduzioni di dimensioni** e **integrazione in altri oggetti** (sistemi embedded).

- Nel 1993 **Apple introduce Newton** (con dispositivo di scrittura a mano libera), aprendo la strada agli odierni **PDA** (Personal Digital Assistants).
- Molto più importanti sono i cosiddetti “**computer invisibili**” in quanto integrati (embedded) all'interno di altri apparati:
 - elettrodomestici
 - giocattoli
 - orologi
 - carte di credito
 - RFID
- Oggigiorno **Smartphone** e **Tablet** dispongono di capacità di calcolo, periferiche e sensori che rendono la computazione onnipresente.

Presente e futuro (1)

- Il livello di evoluzione dei calcolatori è tale da rendere difficile qualsiasi previsione. Tra le poche previsioni che non sono state disattese vi è la **legge di Moore** (cofondatore ed ex-presidente di Intel) che nel 1965 notò la regola che descriveva l'aumento del numero di transistor su un singolo chip che **raddoppiano ogni 18 mesi** (circa il 60% in più ogni anno).



Presente e futuro (2)

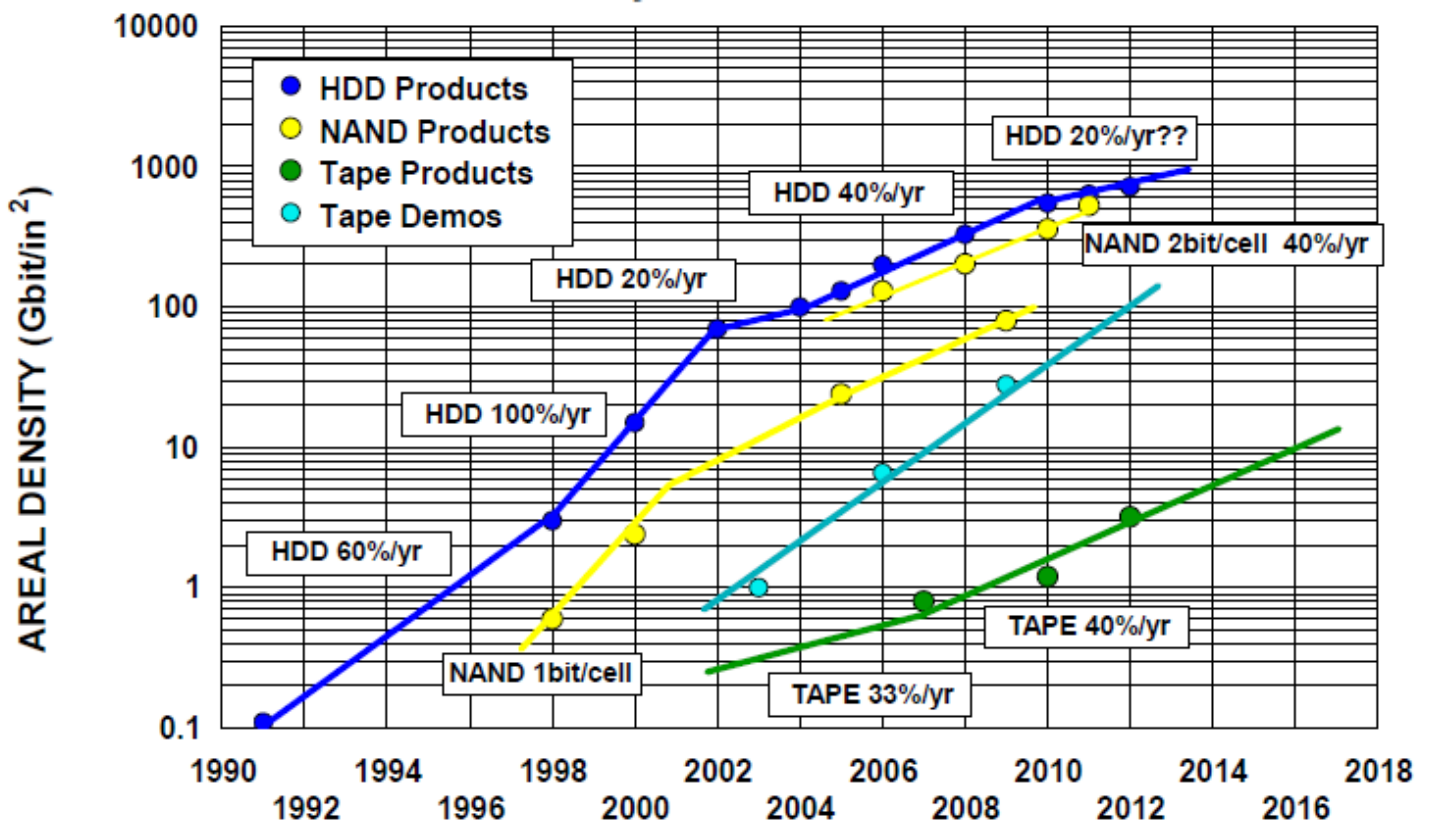
Da Top500: classifica permanente dei 500 calcolatori più potenti



Se il settore automobilistico si fosse sviluppato quanto quello dell'informatica, oggi una Rolls Royce costerebbe 2,50 euro, farebbe più di un milione di chilometri con un litro e andrebbe da Bologna a Milano in un secondo!

Presente e futuro (3)

- Questa stupefacente crescita ha riguardato molti dei dispositivi connessi ai calcolatori:
 - **Capacità Dischi rigidi:** 10 MB nel 1980 – 12 TB nel 2017 $\approx 1.200.000$ volte
 - **Storage:** densità di memorizzazione in Gbit per pollice quadrato (fonte IBM, 2012)



- **Telecomunicazioni e reti:** in 20 anni da 300 bit/sec a 200 Mbit/sec per VDSL

- **Validità Legge di Moore: per quanto ancora?**

Presente e futuro (4)

L'enorme potenza di calcolo e di memorizzazione che si è avuta via via a disposizione ha determinato un rapido incremento della complessità e della dimensione dei prodotti software

“Il software è come un gas che si espande per riempire il contenitore che lo contiene”

Questa regola trova conferma nella tendenza attuale:

- Troppo spesso si trascura l'ottimizzazione del codice affidandosi alla potenza dell'hardware.
- Prolifera il numero di livelli e della virtualizzazione (nate per consentire indipendenza dall'hardware oggi sono meccanismi troppo spesso sovrautilizzati).
- Aggiunta di nuove funzionalità ai prodotti già esistenti fino alla saturazione delle capacità del processore.
- Continua evoluzione dei linguaggi di programmazione che possono diventare obsoleti in pochi anni.

Il forte trend di sviluppo che caratterizza oggi l'informatica rappresenta anche un rischio per l'intero settore:

- Carenza nella qualità e rapida obsolescenza dei prodotti.
- Difficoltà per i programmatori di mantenersi aggiornati.

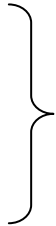
Difficile fare previsioni: alcune frasi celebri

- **1943**: *“I think there is a world market for about five computers”* – T.J.Watson Jr., in seguito diventato presidente dell'IBM.
- **1957**: *“I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year”* - il capo redattore del settore libri per le aziende della Prentice Hall.
- **1970**: *“In from three to eight years we will have a machine with the general intelligence of an average human being”* - Marvin Minsky, uno dei pionieri dell'Intelligenza Artificiale.
- **1977**: *“There is no reason for any individual to have a computer in their home”* - Kenneth Olson, fondatore della DEC.
- **1981**: *“640 K ought to be enough memory for anybody”* - Bill Gates, fondatore e presidente della Microsoft.
- **1988**: Mr. Bill Gates's own estimate is that by 1990, 75-80% of IBM compatible computers will be sold with OS/2.

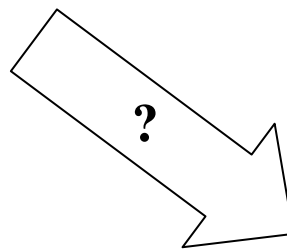
Alcune previsioni per il futuro di Rodney Brooks (uno dei maggiori esperti di robotica):

- <https://rodneybrooks.com/my-dated-predictions>

Trend attuali e poi ...

- Multicore & GPU programming
 - Mobile computing
 - Cloud and Big Data
 - Internet delle cose (sistemi embedded)
 - Realtà aumentata
- 
- trend tecnologici a breve**

- Visione artificiale e Robotica
- Neuroscienze computazionali
- Machine Learning
- Intelligenza artificiale



**Singularità
tecnologica**

- una **singularità tecnologica** è un punto, previsto nello sviluppo di una civilizzazione, dove il progresso tecnologico accelera oltre la capacità di comprendere e prevedere degli esseri umani moderni.
 - dal momento in cui sarà realizzato il primo calcolatore più intelligente dell'uomo si assisterà a uno sviluppo esponenziale, potendo delegare la progettazione di nuovi sistemi ai sistemi stessi (che operano, con feedback positivo, senza fatica 24 ore su 24!). *“la prima macchina ultraintelligente sarà l'ultima invenzione che l'uomo avrà la necessità di fare”* I.J. Good, 1965.
- **Fantascienza o realtà?**
 - La legge di Moore e l'evoluzione tecnologica
 - Già disponibili super-computer con “raw computing power” superiore a quello del cervello umano (stimato in 10-100 PetaFlops).
 - Raw computing power **non significa però intelligenza!**
 - Reverse-engineering del cervello umano e neuroscienze computazionali. L'algoritmo della corteccia celebrale ancora sconosciuto (lettura consigliata: “**On Intelligence**”, **J. Hawking & S. Blakeslee**).
 - **Ray Kurzweil** e altri ipotizzano singolarità attorno al 2040.