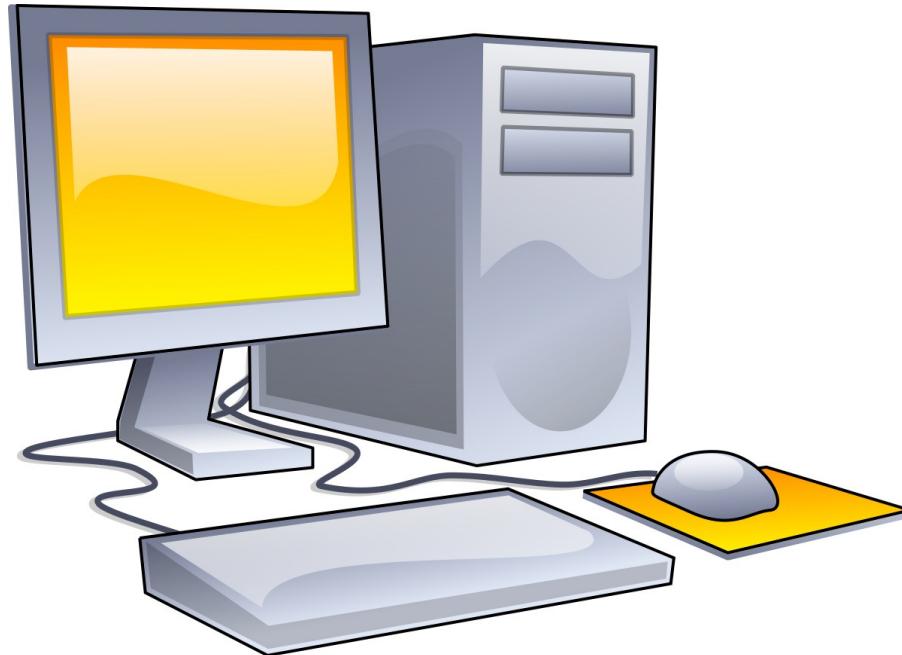
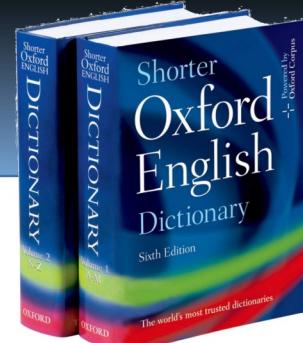


- Cosa significa computer
- Un po' di storia
- Il modello di Von Neumann

Computer





- Computer

Noun, Brit. /kəm'pjutə/, U.S. /kəm'pjudər/

1. A person who makes calculations or computations; a calculator, a reckoner; spec. a person employed to make calculations in an observatory, in surveying, etc. Now chiefly *historical*.
2. A device or machine for performing or facilitating calculation.
3. An electronic device (or system of devices) which is used to store, manipulate, and communicate information, perform complex calculations, or control or regulate other devices or machines, and is capable of receiving information (data) and of processing it in accordance with variable procedural instructions (programs or software); esp. a small, self-contained one for individual use in the home or workplace, used esp. for handling text, images, music, and video, accessing and using the internet, communicating with other people (e.g. by means of email), and playing games

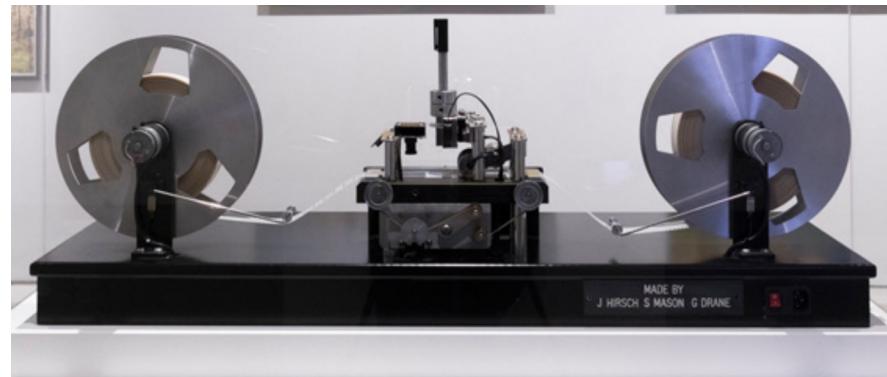


- In Italiano
 - Elaboratore
 - Calcolatore (elettronico)
- Dispositivo *elettronico* e digitale
 - **programmabile**
 - **per uso generale**

- Esempi di dispositivi programmabili
 - Sveglia
 - Lavatrice
 - Microonde
 - Ecc.
- Programmabile significa che posso impostare le operazioni da eseguire

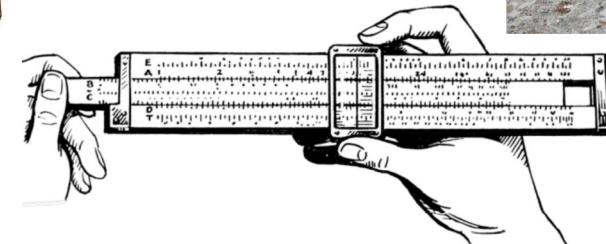
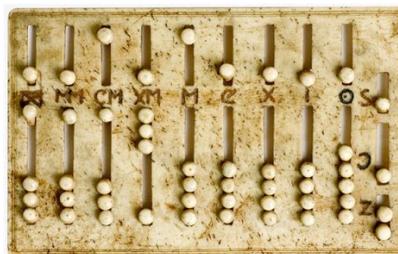


- General Purpose
 - Non specializzato
 - Adattabile per ogni utilizzo
- Da un punto di vista teorico non ci sono limiti





- La maggior esigenza di automazione che l'uomo ha avuto per lungo tempo è stata quella dei calcoli
- Nella storia sono apparsi differenti soluzioni per “agevolare” tali operazioni





- Appaiono in Francia a fine 700
 - Inventore italiano
- Telaio Jacquard (1804)
 - Nastro con fori che viene fatto avanzare durante la lavorazione
 - Contrappesi che controllano lo schema di tessitura
 - Quando il contrappeso incontra il foro agisce sull'ordito



Linguaggio di Programmazione → schema di fori

Primo “computer”



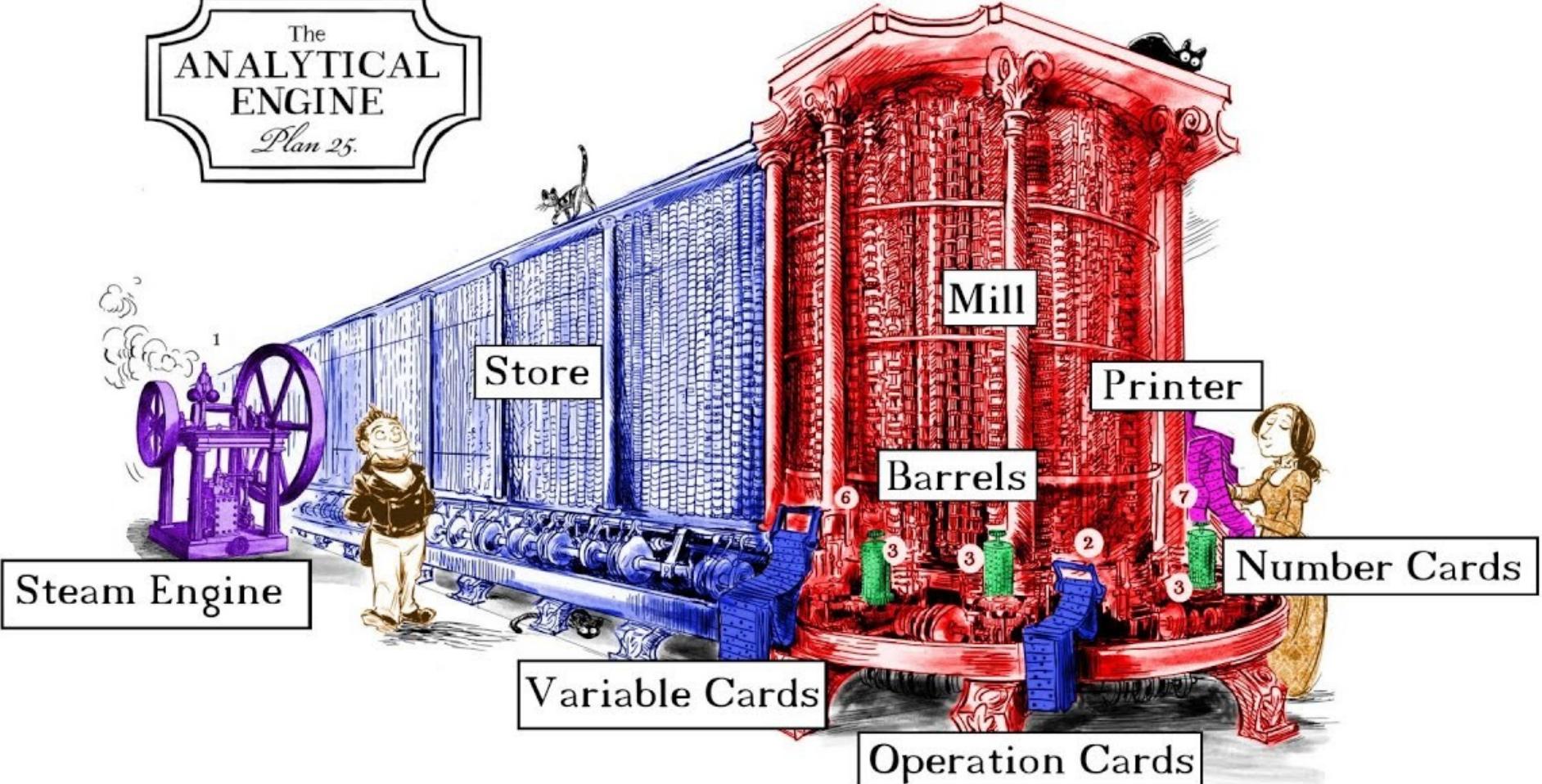
- Macchina analitica di Babbage (1833)
 - *On the Mathematical Power of the Calculating Engine*
- Mai costruita
 - 30 x 25 metri gestita da motore a vapore
- Schede perforate
 - Piccole → operazioni
 - Grandi → dati (variabili)

Linguaggio di Programmazione → schema di fori

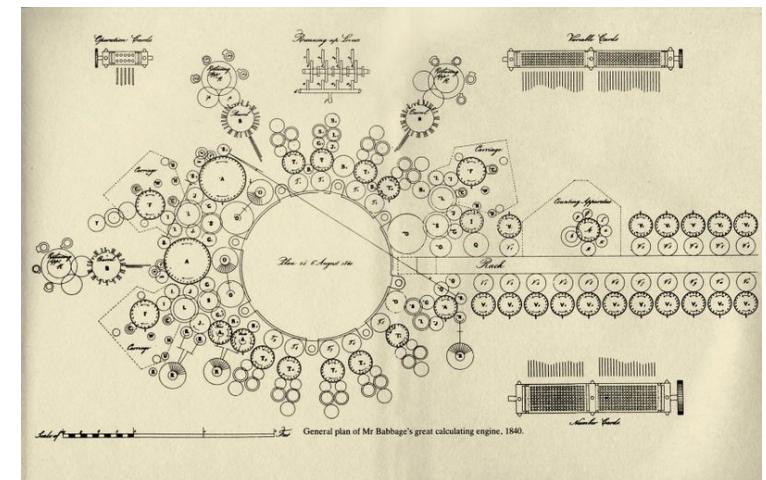
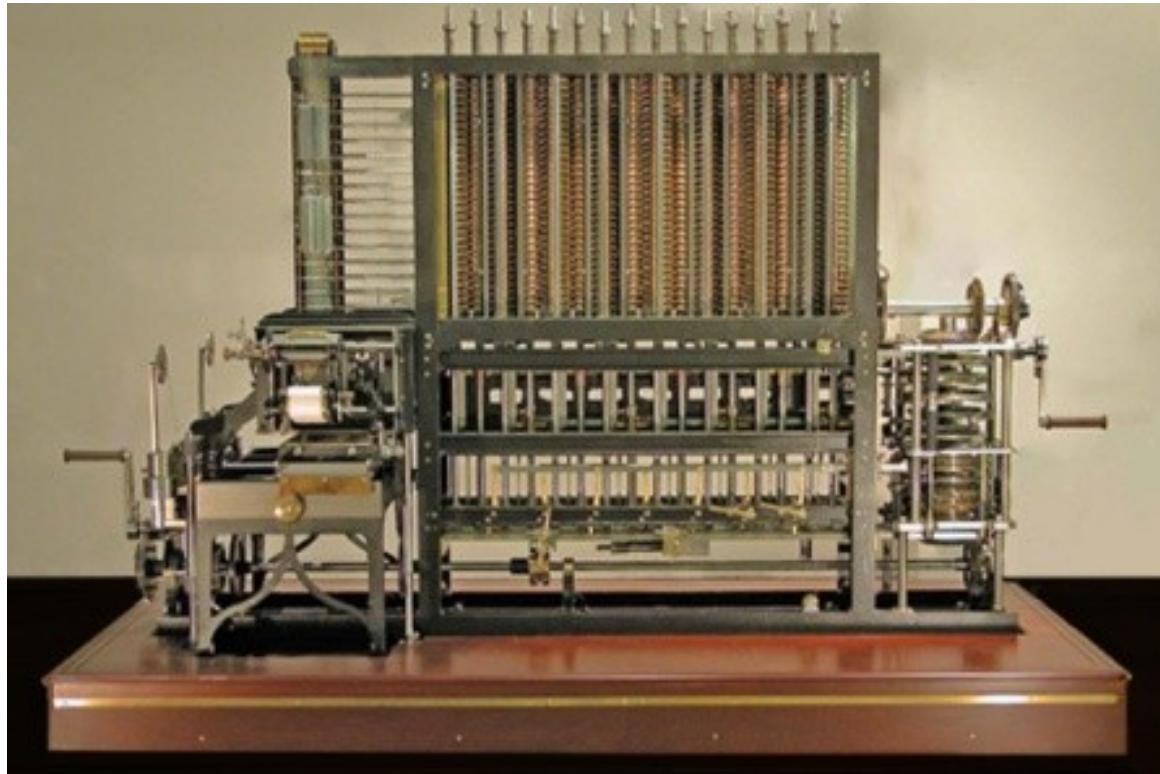


- La macchina di Babbage contiene già tutti gli elementi di un moderno computer
 - Controllo → lettura e interpretazione schede di istruzioni
 - Elaborazione → il *Mill* in grado di eseguire le 4 operazioni principali
 - Opzionalmente anche la radice quadrata
 - Procedure “barrel”
 - Memoria → registri interni fino a 1000 numeri da 40 cifre a virgola fissa (~17 kByte)
 - Input → le schede
 - Output → sistema di perforazione schede (stampa)
- Linguaggio di programmazione *Turing completo*
 - Equivalente ad una macchina di Turing universale

The
**ANALYTICAL
ENGINE**
Plan 25.



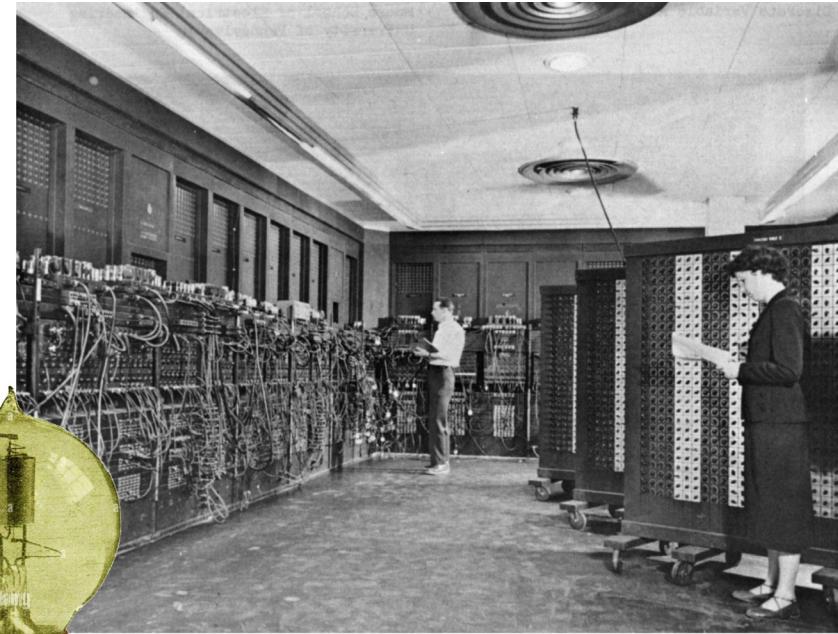
Analytical Engine



- IBM + Università di Harvard
- Elettromeccanico
- Sempre schede perforate
- 3 addizioni o sottrazioni al secondo
- 1 moltiplicazione in 6 secondi
- 1 divisione in oltre 15 secondi
- Logaritmi e operazioni trigonometriche > 1 minuto



- Esclusivamente elettronico
- Valvole termoioniche
 - Problema affidabilità (2 minuti)
 - I transistor furono inventati nel 1947
- Circa 1000 volte più veloce del Mark I
- Sempre schede perforate e sistema decimale
- Turing Completo

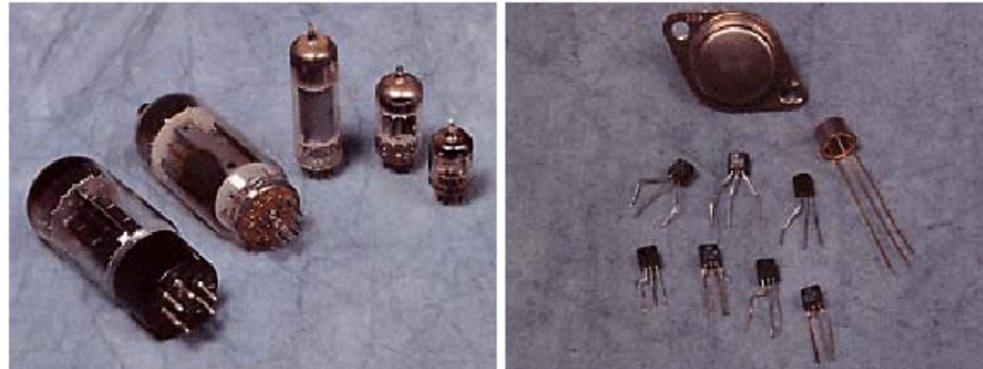


Zuse Z4 (1939-1945)



- Non governativo
 - Primo computer commerciale
- Ancora elettromeccanico
- Numeri a virgola mobile **binari**

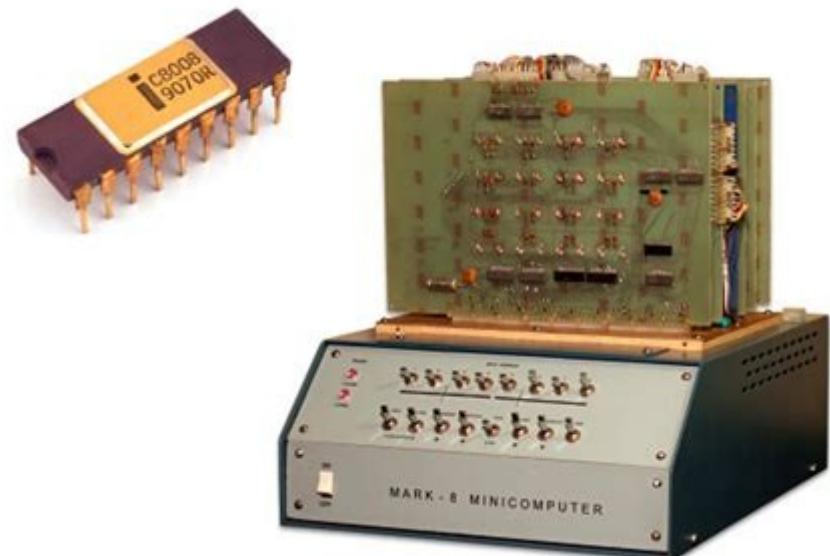




The miniaturization of computers began with the invention of the transistor on December 23rd 1947 by William Shockley, John Bardeen and Walter Brattain at Bell Laboratories. Early computers like the ENIAC contained thousands of vacuum tubes, millions of soldered joints and consumed 100's of kilowatts of electricity. Transistors would replace the tubes, greatly reduce the heat output and space requirements.



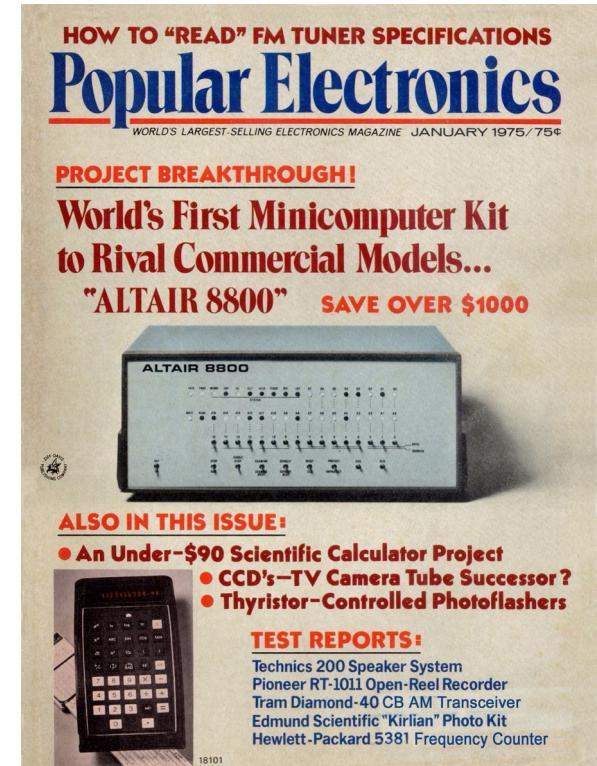
- Dai transistor ai microprocessori
- Intel 4004
 - ~2300 transistor
 - 10.000 nm
 - Fino a 740 kHz
- Stessa potenza dell'ENIAC
 - 84 m³
- Calcolatrici...



IMIT Altair 8800 (1975)



- Primo vero personal computer (Intel 8080)
 - 397 USD in kit o 495 USD già montato
 - ~ 1700 EUR di oggi
- 256 byte di RAM
- No ROM ma avvio manuale
- I/O nastro magnetico (input) e led (output)
 - No tastiera di serie
 - No display
 - No storage interno
- Altair BASIC → MicroSoft



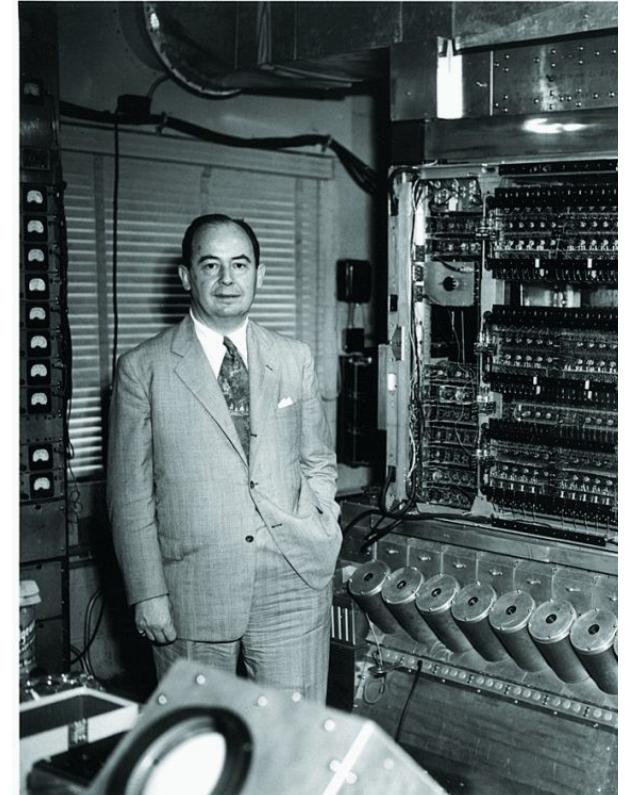
- In pochi anni diventano disponibili altri elementi
 - Tastiera
 - Video
 - Storage
- Dopo l'Altair nascono i Personal Computer
 - IBM PC (1981)
 - Olivetti M24 (1983)
 - Macintosh (1984)
- Dal punto di vista organizzativo dal 1946 è comunque cambiato poco



Modello di Von Neumann (1946)

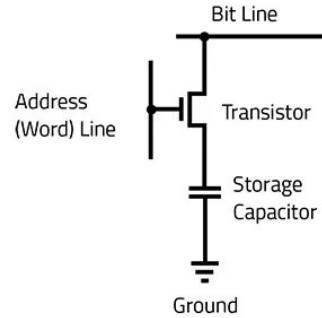


- Modello di organizzazione **logica** di un computer
- Tutt'ora attuale!
- 5 parti fondamentali
 - Memoria (unica!)
 - Unità di elaborazione aritmetica e logica
 - Unità di controllo
 - Input/Output

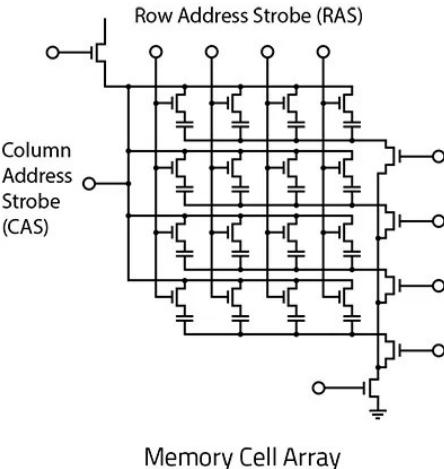




- RAM → Random Access Memory
 - Diverse unità di stoccaggio “dati”
 - Celle elementari con valore fittizio
 - Bit organizzati in unità di accesso (**Byte**)
 - Ogni cella ha un ben preciso **indirizzo**
- Cosa contiene?
 - Dati
 - Ma anche istruzioni!



Single Memory Cell

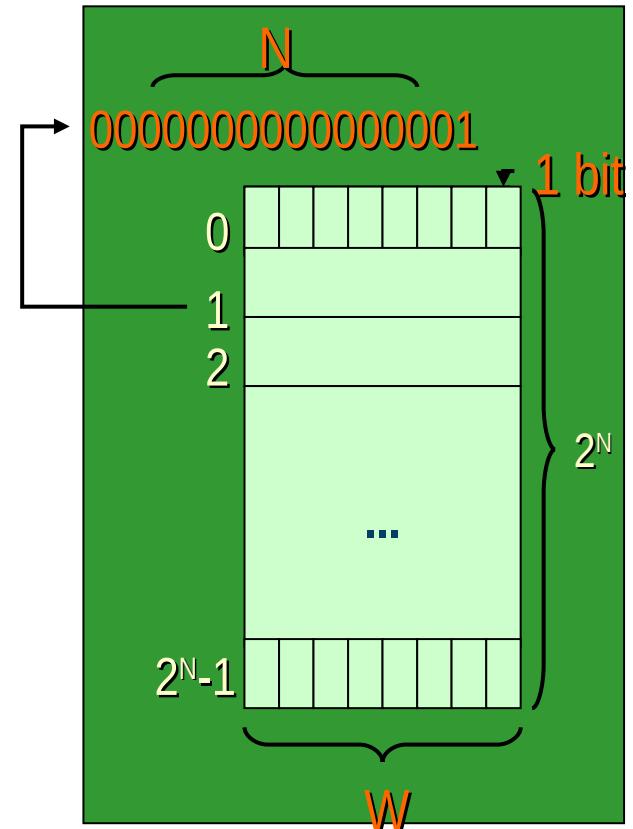


Memory Cell Array

Numeri → convenzionalmente composti da 0 e 1

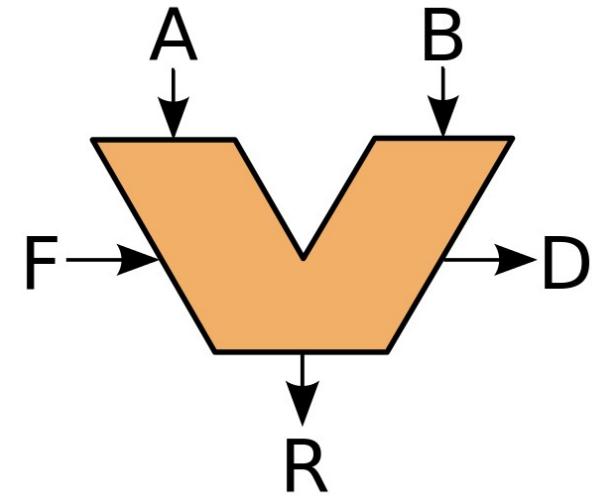


- RAM
 - Distinguiamo indirizzo e contenuto
- Memory Width (W)
 - Quanti bit per cella?
 - Tipicamente 8 bit → 1 Byte
- Address width (N)
 - Quanti bit per indirizzo?
 - Limita il quantitativo massimo di memoria
 - N bit → 2^N possibili indirizzi





- Componenti
 - Aritmetic Logic Unit - ALU
 - Operazioni matematiche (+, -, /, *...)
 - Interi o virgola mobile
 - Operazioni logiche (>, <, =, and, or, not...)
 - Circuiti che eseguono le relative operazioni
 - Registri ovvero memorie rapide per funzionamento interno
 - Registri ad uso “generale”
 - Registri per accesso memoria
 - Registro di stato o flag
 - ...



A, B	→	operandi
R	→	risultato
F	→	operazione
D	→	status

- Dove si trova il programma da eseguire?
 - In memoria → “numeri”
- La control unit esegue i programmi ovvero:
 - Legge da memoria la relativa istruzione da eseguire → **Fetch**
 - Decodifica il suo significato ovvero cosa deve fare → **Decode**
 - Esegue quanto richiesto pilotando gli altri sistemi → **Execute**
 - Scrive eventuali risultati elaborazione → **Store**
 - Ripete...
- Registri per lettura istruzione (IR) e posizione lettura (PC)

- Come è fatta una istruzione decodificata dall'unità di controllo?
- Tipicamente più “parti”
 - Operation code → mi dice che istruzione è
 - Operando/i → mi fa capire su cosa si opera
- Instruction Set → dipende da processore!
- Esempio: ADD X, Y
 - Somma il contenuto della memoria all'indirizzo X a quello in Y e scrivi il risultato in X

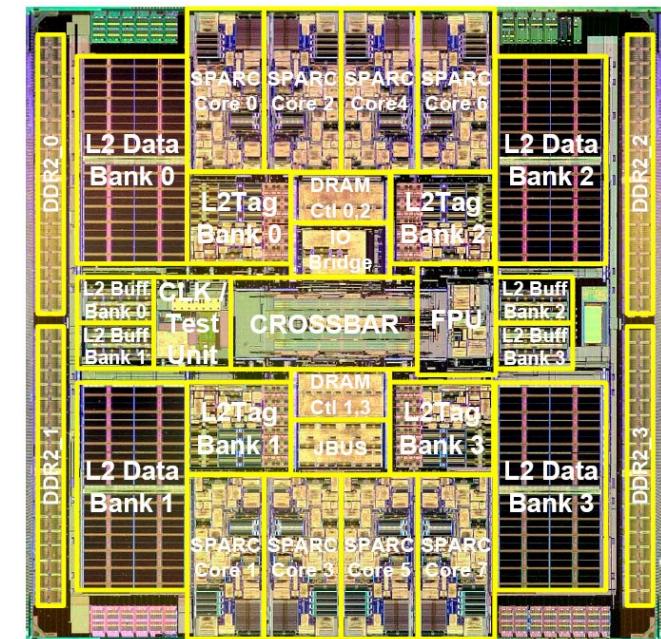
Opcode (8 bit) Indirizzo X (16 bit) Indirizzo Y (16 bit)

00001001

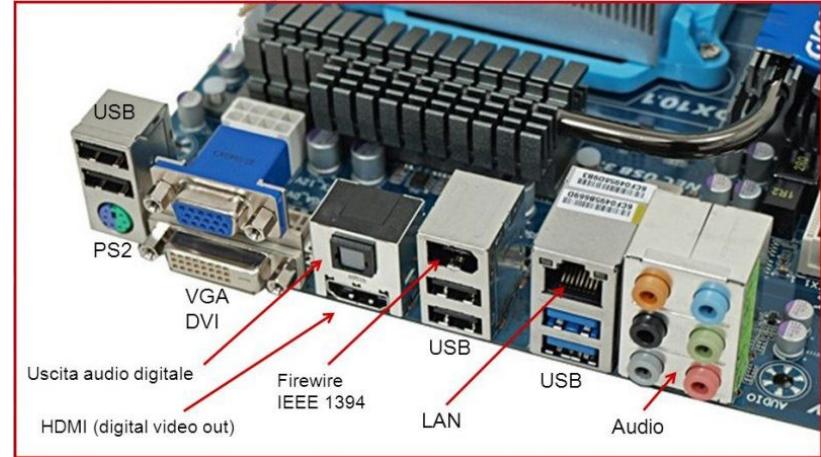
0000000001100011

0000000001100100

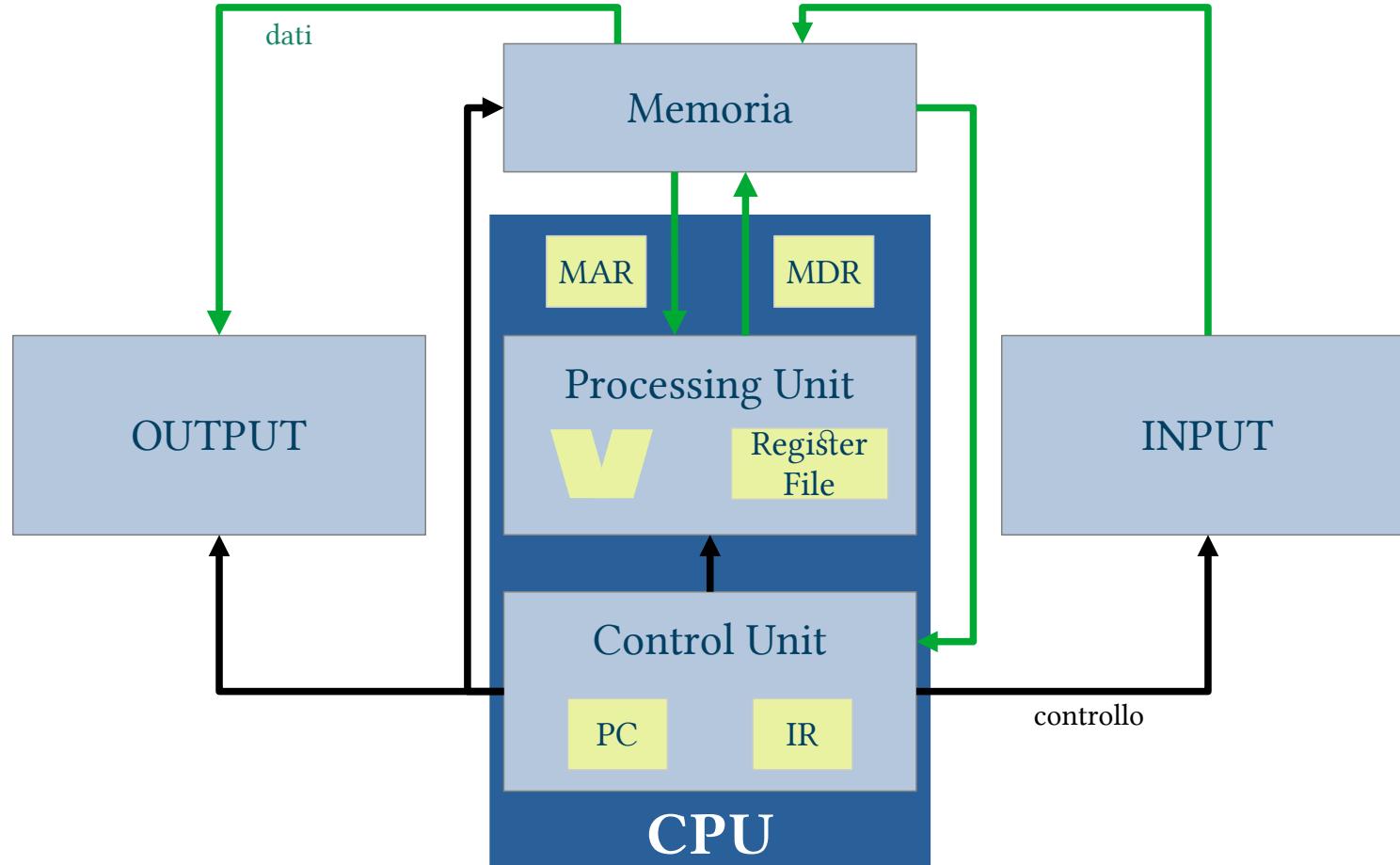
- Distinzione tra unità di elaborazione e controllo di fatto non c'è più fisicamente
 - CPU → Central Processing Unit
- Caratteristiche evolute:
 - Multiprocessing (superscalari) → ALU multiple
 - Multicore → anche più unità di controllo



- Gestisce device “esterni”
 - Tastiera/Video
 - Hard disk, hardware vario...
- Quasi mai connessione sincrona
 - Velocità differenti
 - Buffer sia in ingresso che in uscita
 - Memoria per i dati in ingresso/uscita
 - Trasferimento da o verso RAM



Von Neuman model



- Modello ancora attuale
- Il modello si riflette in quanto dovremo fare per programmare
 - CPU esegue sequenze di istruzioni
 - Il programmatore individua la sequenza di istruzioni appropriata
 - Primi linguaggi di programmazione → linguaggi imperativi

