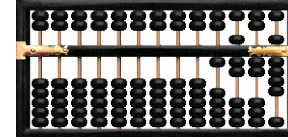


Breve storia dei calcolatori elettronici (architettura di von Neumann)

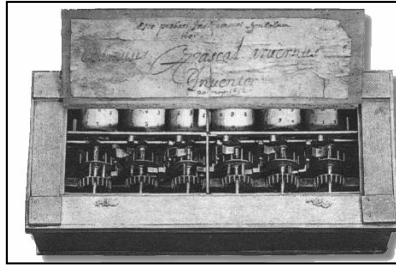
A. Marchetti Spaccamela, Domenico Lembo,
Paolo Liberatore, Giuseppe Santucci, Marco Schaerf

Gli inizi

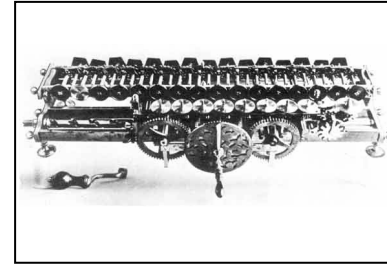
- I calcolatori elettronici sono recenti
- Le macchine di calcolo sono antiche: L'abaco è stato inventato circa 4000 anni fa
- Nel XVII secolo sono state costruite diverse macchine meccaniche per il calcolo



Ricostruzione della macchina calcolatrice inventata da Wilhelm Schickard – 1623 (Deutsches Museum, Monaco)



La "Pascalina" di Blaise Pascal – 1642 (Musée des Arts et Metiers, Parigi)



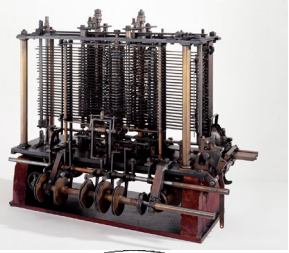
La calcolatrice (Stepped Reckoner) di Gottfried Wilhelm von Leibniz' s – 1671 ca.

1800: Babbage

Charles Babbage è una delle figure più affascinanti dell'informatica. Spinto dall'idea di poter costruire una macchina per la produzione di tavole matematiche, Babbage progettò due macchine, la “differential machine” e la “analytic machine”, PROGRAMMABILE, che ha anticipato molte delle caratteristiche che si trovano nei computer moderni.

Babbage non è riuscito a completare nessuna delle macchine durante la sua vita. Lo Science Museum di Londra è stato in grado di realizzare nel 1910 una versione funzionante della “differential machine” in occasione del 200° anniversario della sua nascita.

Schede perforate per l'input,
ispirate a quelle usate ai tempi per i telai tessili di Jacquard



Charles Babbage (1791-1871)



1910



Poteva succedere

- Nel 1835 Camillo Cavour incontra Babbage a Londra, sente parlare della macchina analitica che lo colpisce (Cavour era un ingegnere)
- Nel 1840 Charles Babbage è invitato e partecipa al primo Congresso degli scienziati italiani a Torino
- Babbage tiene una serie di seminari a Torino e regala i disegni della sua macchina all'Accademia delle Scienze
- Nel 1842 Luigi Menabrea pubblica (in francese) la descrizione dell'architettura della macchina di Babbage
- Solo successivamente il lavoro di Menabrea è tradotto in inglese da Ada Byron Lovelace (detta da Babbage *the Enchantress of numbers*)
- Il Piemonte pensa di finanziare la costruzione *dell' analytical engine*
- Poi, come al solito, non se ne fece nulla ...
- Così Moncalieri (Torino) è rimasto Moncalieri e non è diventato Cupertino (California)

Ada Byron, la prima programmatrice



**Augusta Ada Byron,
Lady Lovelace (1815–1852)**

Ada Byron, figlia del poeta inglese Lord Byron, fu incoraggiata a perseguire i suoi interessi nel campo della scienza e della matematica in un momento in cui pochissime donne potevano studiare questi argomenti.

All'età di 17 anni, Ada incontrò Charles Babbage e rimase affascinata dalle sue macchine. Ada era convinta delle potenzialità della “analytic machine” di Babbage e scrisse ampie note sul suo progetto, insieme a diversi programmi matematici complessi che hanno portato molte persone a caratterizzare la sua figura come quella della prima programmatrice della storia.

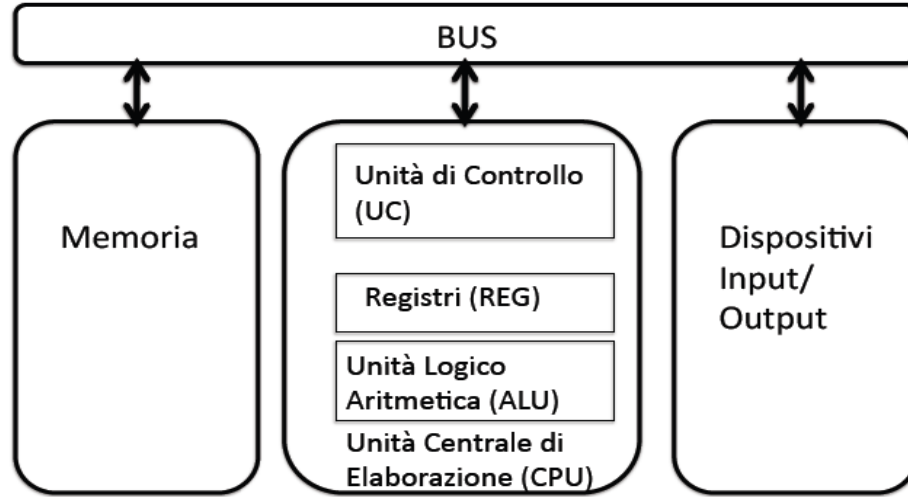
Nel 1980, il linguaggio di programmazione per calcolo parallelo Ada è stato così chiamato in suo onore.

1940: La nascita dell'informatica

Non è chiaro chi ha inventato i computer moderni: ci sono molti candidati

- Nel 1939, [John Atanasoff](#) e [Clifford Barry](#) (Iowa State University) costruirono un prototipo per risolvere equazioni lineari (L'Atanasoff-Berry Computer, spesso chiamato *ABC*), e successivamente, nel 1942, una macchina più grande.
- [Konrad Zuse](#) in Germania verso la fine degli anni '30, ha realizzato la macchina programmabile Z1, poi evoluta nelle macchine Z2 e Z3.
- Il Colossus è una macchina elettronica programmabile realizzata nel Regno Unito (1943-44) per decifrare i messaggi codificati dell'esercito nazista. Al suo progetto ha contribuito anche [Alan Turing](#) (a capo del gruppo crittografico inglese durante la seconda guerra mondiale)
- Il primo computer su larga scala costruito è l'*Electronic Numerical Integrator and Computer* (ENIAC), completato nel 1946 sotto la direzione di [J. Presper Eckert](#) e [John Mauchly](#) (University of Pennsylvania).
- Altri importanti contributi durante i primi anni comprendono il concetto di programmazione memorizzata, generalmente attribuita a [John von Neumann](#), e l'uso di circuiti di commutazione per implementare aritmetica binaria proposto da [Claude Shannon](#).

Architettura CPU: modello di von Neumann



Modello: una rappresentazione concettuale (spesso una semplificazione) del mondo reale o di una sua parte, o di un manufatto capace di spiegarne il funzionamento.

Nota: la descrizione originale dell'architettura di von Neumann non prevedeva un bus ma collegamenti diretti fra le diverse componenti.

Il modello di von Neumann

Il Modello di Von Neumann comprende

1. L'**Unità di controllo** si occupa di controllare tutte le operazioni del calcolatore, interpretare le istruzioni prelevate dalla memoria e inviare alle altre unità i segnali per l'esecuzione delle operazioni

2. L'**Unità aritmetico-logica**, detta **ALU** (Arithmetic & Logic Unit), permette di effettuare operazioni aritmetiche di base (somme, sottrazioni, ecc.) e di prendere decisioni

*Queste due unità sono spesso integrate in una **CPU**, Central Processing Unit – Unità di Elaborazione Centrale (che contiene anche una serie di registri, simili alla memoria)*

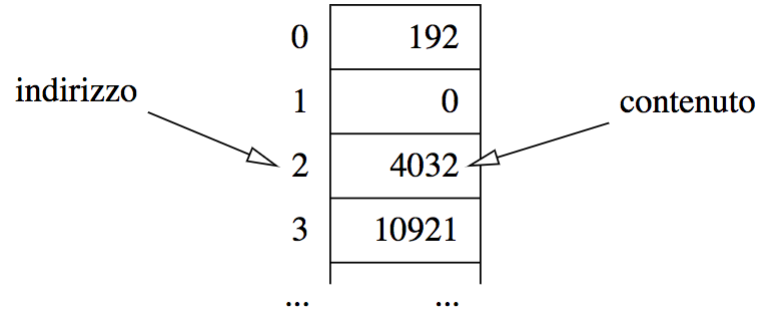
Il modello di von Neumann: cont.

3. La **Memoria** conserva *le istruzioni e i dati* da elaborare e i risultati ottenuti dalle elaborazioni*;
4. L'**Unità di ingresso (Input)** immette le informazioni nel calcolatore per farle elaborare;
5. L'**Unità di uscita (Output)** riceve le informazioni dalla memoria del calcolatore per renderle pronte all'uso; *le unità di ingresso e uscita sono anche dette periferiche*
6. Il **Bus**, vero e proprio canale di comunicazione che consente ai dati di transitare fra diversi componenti del calcolatore.

* Il modello alternativo chiamato Berkley prevede spazi di memoria diversi per le istruzioni ed i dati.

Memoria

- una serie di celle (locazioni di memoria)
- ogni cella può contenere un numero (contenuto)
- le celle sono numerate (indirizzo di una locazione)



Memoria

Lettura da memoria

0	192
1	0
2	4032
3	10921
...	...

Se la CPU vuole leggere il contenuto della locazione di indirizzo 2:

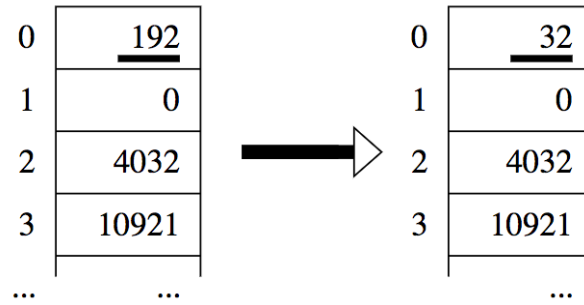
- La CPU manda alla memoria:
 - l'indirizzo 2
 - l'indicazione che vuole leggere
- la memoria risponde con il valore 4032

Scrittura in memoria

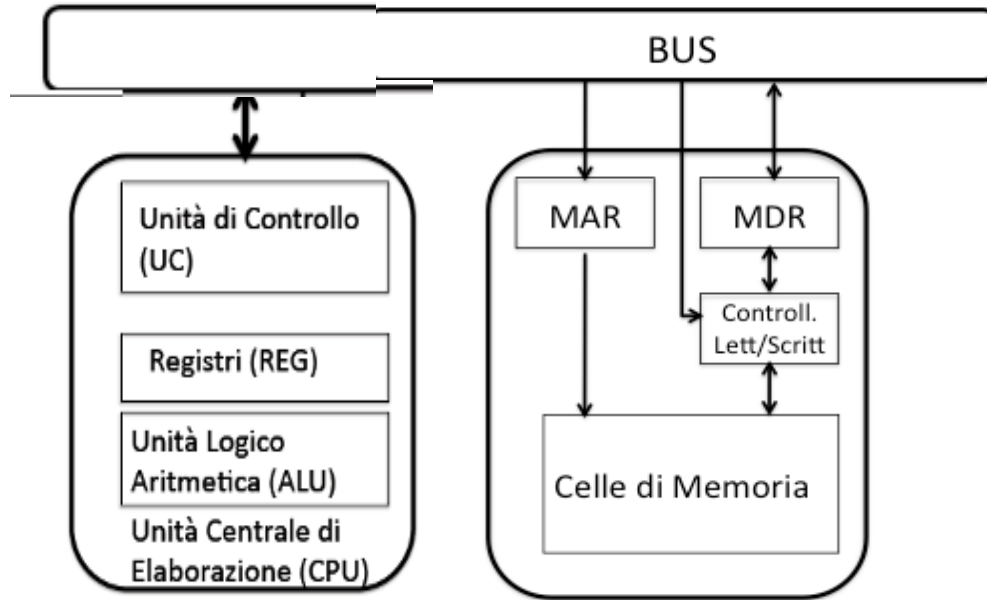
se la CPU vuole scrivere 32 nella locazione di indirizzo 0:

1. La CPU manda alla memoria
 - l'indirizzo 0
 - il valore 32
 - l'indicazione che si tratta di una scrittura

la memoria non manda indietro nessun valore



Sottosistema memoria



- insieme di celle
- MAR = registro per l'indirizzo
- MDR = registro per il dato
- Controll. Lett./Scritt.: selettore che decide se il contenuto della cella indirizzata dal MAR va alla CPU (se la CPU ha chiesto una lettura) o se deve essere sovrascritto con il contenuto dell'MDR (se la CPU ha chiesto una scrittura)

Nota: un registro è come una singola cella di memoria

Registri della CPU

La CPU contiene diversi registri

- Registro Istruzione (IR): memorizza l'istruzione da eseguire
- Contatore di Programma (PC): memorizza l'indirizzo in memoria della prossima istruzione da eseguire
- Registri Dati (in numero variabile): contengono dati utilizzati per i calcoli

I registri della CPU sono aggiornati dall'esecuzione dell'istruzione

Il ciclo Fetch-Decode-Execute

Passi dell'esecuzione di ciascuna istruzione nella CPU:

1. Carica l'istruzione da memoria in **IR** (*Instruction Register*) (**Fetch**)
2. Incrementa **PC** (*Program Counter*)
3. Decodifica l'istruzione (**Decode**)
4. Se l'istruzione usa un dato in memoria ne calcola l'indirizzo
5. Carica l'operando in un registro
6. Esegue l'istruzione (**Execute**)
7. Torna al passo 1. per l'esecuzione dell'istruzione successiva

*Accessi a memoria sono effettuati sempre al passo 1,
e non sempre ai passi 4, 5 e 6*

Tipi di istruzioni

- aritmetiche e logiche
 - es. somma di due registri, risultato in un terzo
- lettura e scrittura da memoria:
 - mem \rightarrow reg
 - reg \rightarrow mem
- input/output:
 - dispositivo \rightarrow reg
 - reg \rightarrow dispositivo
- salti (condizionati e non)
 - alterano l'esecuzione lineare del programma
 - permettono di implementare condizioni, cicli, ecc.

Istruzioni di controllo

di norma, dopo una istruzione si esegue la successiva

es. dopo l'istruzione alla locazione 133 si esegue quella alla locazione 134

esistono solo due tipi di istruzioni per alterare quest'ordine:

- salti incondizionati
- salti condizionati

entrambi scrivono un valore in PC

Il Programma

- nell'architettura di Von Neumann, si trova in memoria assieme ai dati
- è rappresentato da una sequenza di codici numerici (e.g., 56 = somma)
- vengono letti ed eseguiti uno per volta

Esecuzione del programma (esempio):

la sequenza 1921 - 992 - 10063 - 56 rappresenta un programma di quattro istruzioni (codificate, per comodità, in base 10)

l'unità centrale legge 1921 e la esegue

poi legge 992 e la esegue

poi legge 10063 e la esegue

poi legge 56 e la esegue

PC = l'indirizzo (132, 133, 134,...)

IR = l'istruzione (1921, 992, 10063, ...)

...	...
132	1921
133	992
134	10063
135	56
...	...

- *Nota: se l'istruzione da eseguire è una istruzione di "salto", il PC viene aggiornato all'indirizzo di memoria che contiene l'istruzione a cui saltare*

Esecuzione del programma

sequenza di passi:

PC=132 IR=...

lettura da memoria della locazione 132

incremento PC

PC=133 IR=1921

controllo esegue l'istruzione 1921

lettura da memoria della locazione 133

incremento PC

PC=134 IR=992

controllo esegue l'istruzione 992

...

...	...
132	1921
133	992
134	10063
135	56
...	...

Nota: se l'istruzione da eseguire è una istruzione di “salto”, il PC viene aggiornato all'indirizzo di memoria che contiene l'istruzione a cui saltare

Il linguaggio della macchina

- La CPU è in grado di eseguire solo istruzioni molto semplici.
- Queste sono espresse in linguaggio macchina (o codice binario), dipendente dall'elaboratore.
- Il linguaggio assemblativo consente di esprimere istruzioni in forma simbolica (comprensibile da un umano). Queste sono in corrispondenza 1-1 con le istruzioni in linguaggio macchina.
- Nel seguito consideriamo un esempio di processore semplice (non esistente nella realtà), con quattro registri (R1, R2, R3, R4) ed un insieme semplificato di istruzioni, espresse in forma simbolica.

Nota: Linguaggi come Python, Java, C, C++,.. sono linguaggi di alto livello: si basano su costrutti non elementari, comprensibili da un umano. Ogni istruzione corrisponde in genere a molte istruzioni in linguaggio macchina. In larga misura questi linguaggi sono indipendenti dallo specifico elaboratore.

Un semplice linguaggio

HALT	(arresto del programma)
LOAD X IND	(Leggi la cella con indir. IND e trasferisci il dato al reg. X)
STORE X IND	(Scrivi il valore del registro X nella cella di indirizzo IND)
ADD X,Y,Z	(Esegui $X=Y+Z$, dove X, Y e Z sono registri)
READ IND	(Leggi un valore in ingresso e ponilo in cella di indir. IND)
WRITE IND	(Scrivi in uscita il valore della cella di indir. IND)

Programma per la somma tre numeri

```
READ 1000
READ 1001
READ 1002
LOAD R1 1000
LOAD R2 1001
LOAD R3 1002
ADD R4,R1,R2
ADD R4,R3,R4
STORE R4 1003
WRITE 1003
HALT
```

```
a=int(input())
b=int(input())
c=int(input())
d=a+b+c
print(d)
```

Codice Python

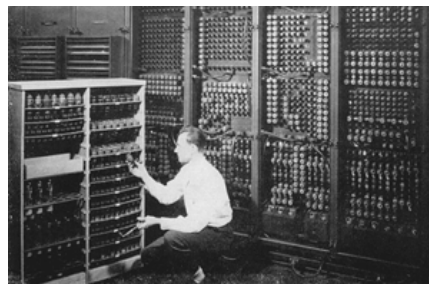
Calcolatori e mente umana

- Perché i calcolatori sono molto più veloci degli essere umani nei calcoli anche se eseguono operazioni molto semplici?
- Velocità : 1 operazione elementare richiede
 - Neuroni: circa 5 millisecondi (200 oper. al sec.)
 - Computer: circa 1 nanosecondo (1000 milioni al sec.)
- Memoria: difficile da valutare negli umani
 - Essere umani (difficile valutazione: circa 400 milioni di bit)
 - Computer (1 tera byte : 8 milioni di milioni di bit)
- e gli esseri umani si stancano, sbagliano, hanno dubbi ...

La 2^a guerra mondiale (1930-1955)

Tecnologia elettromeccanica (relè)

- **Konrad Zuse** (~1930 Germania)
 - Macchina a relè (distrutta nella guerra)
- **John Atanasoff** (~1940 USA)
 - Aritmetica binaria
 - Memoria a condensatori
- **Howard Aiken** (~1940 USA)
 - *MARK 1*: versione a relè della macchina di Babbage
 - Memoria: 72 x 23 cifre decimali
 - tempo di ciclo: 6 sec.
 - I/O su nastro perforato
 - Mark 2
 - Nasce il bug



Bug ? 9 settembre 1947, una falena incastrata in un relè

9/9

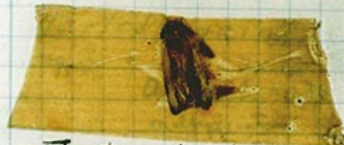
0800 Antam started
1000 " stopped - antam ✓
1300 (033) MP-MC ~~1.58264000~~ 1.2700 9.037847025
(033) PRO 2 ~~2.130476415~~ 2.130476415 9.037846995 correct
correct 2.130676415

Relays 6-2 in 033 failed special speed test
in relay " 11.000 test.

Relay
3145
Relay 3376

1100 Started Cosine Tape (Sine check)
1525 Started Multi-Adder Test.

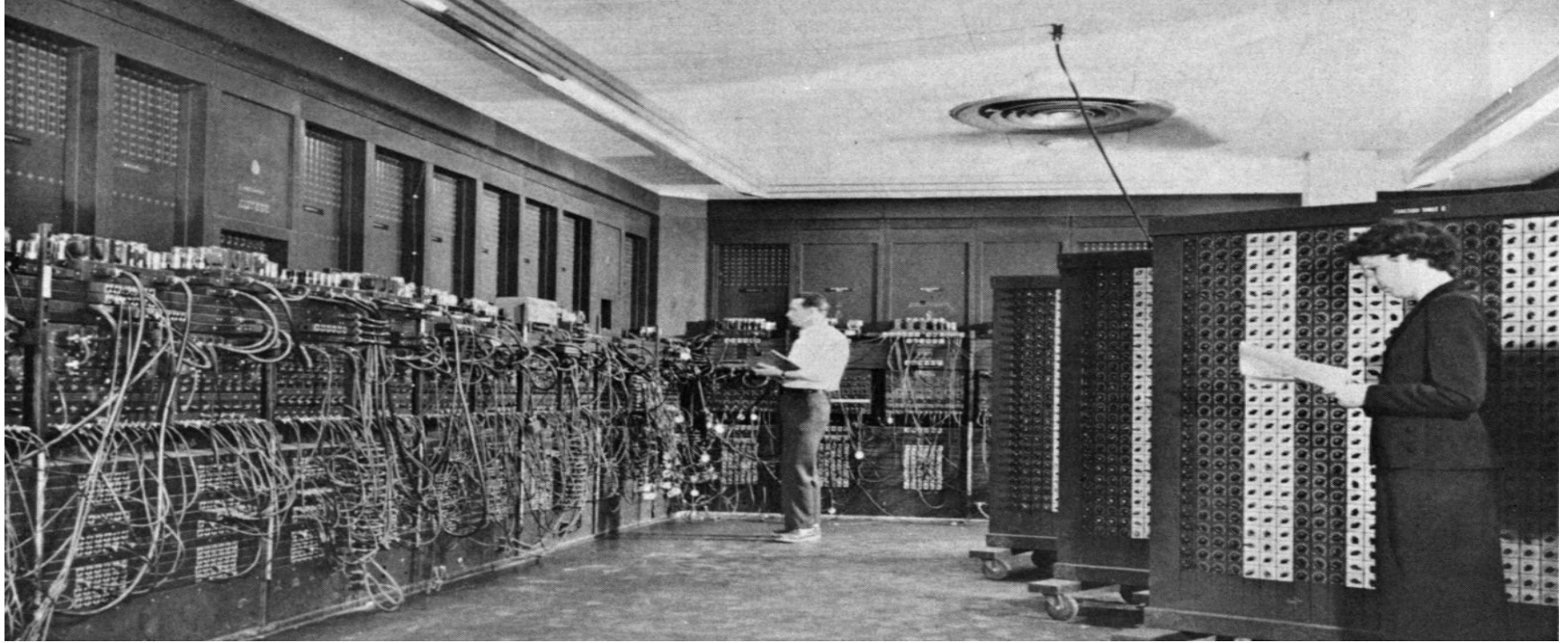
1545



Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.
1630 Antam started.
1700 closed down.

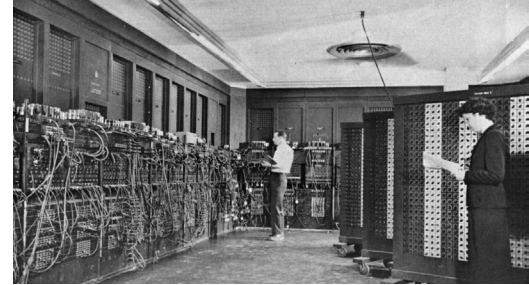
Un progresso incredibile: ENIAC 1946



Un progresso incredibile: ENIAC

1946

- Velocità : 5000 operazioni al sec.
(5 K Hz)
- Memoria: 20 registri da 10 cifre
- Costo: circa 5 Milioni di euro (di oggi)
- Peso e dimensioni : 27 tonnellate e richiedeva uno spazio di circa 160 metri quadri
- Programmabile tramite 6000 interruttori e pannelli cablati
- 18.000 valvole
- Consumo : 150 K Watt
- Affidabilità: all'inizio un disastro, poi migliore: si rompeva una volta ogni due giorni ...



Computer missione Apollo, 1969



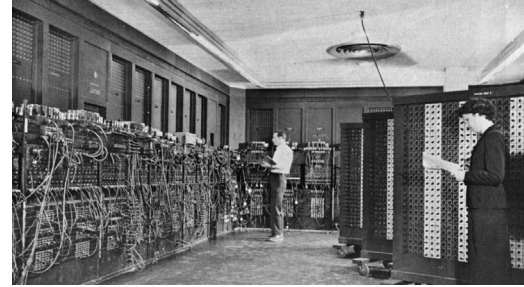
61 × 32 × 17 cm 32Kg
30720 bits di memoria (RAM)

Un progresso incredibile

50 anni dopo

Notebook

- Velocità: 3-4 GHz
- Memoria: disco 1 teraB
- Costo 500-1000 euro
- Peso: meno di 1 kilo
- Consumo : 50 Watt (CPU)
- Affidabilità enorme



1 K (kilo) = mille

1 M (Mega) = 1 milione

1G (Giga) = 1 miliardo

1 T (Tera) = 1000 miliardi

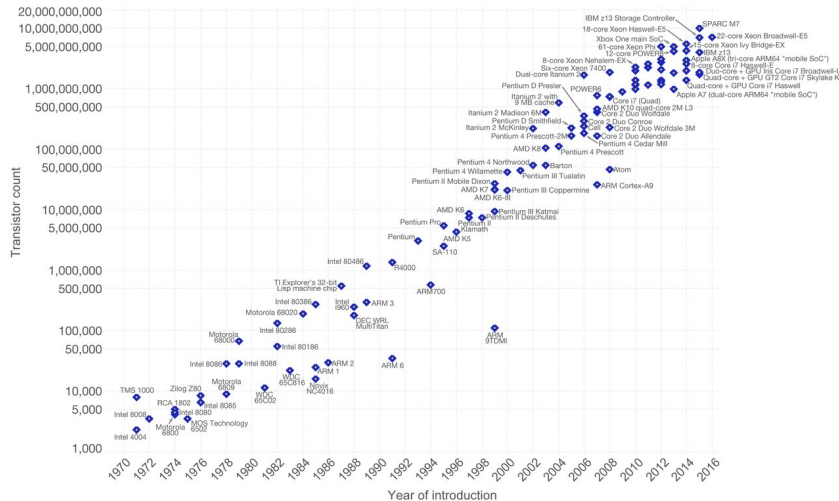
1 P (Peta) = 1 milione di miliardi

Legge di Moore per le CPU

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)



Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



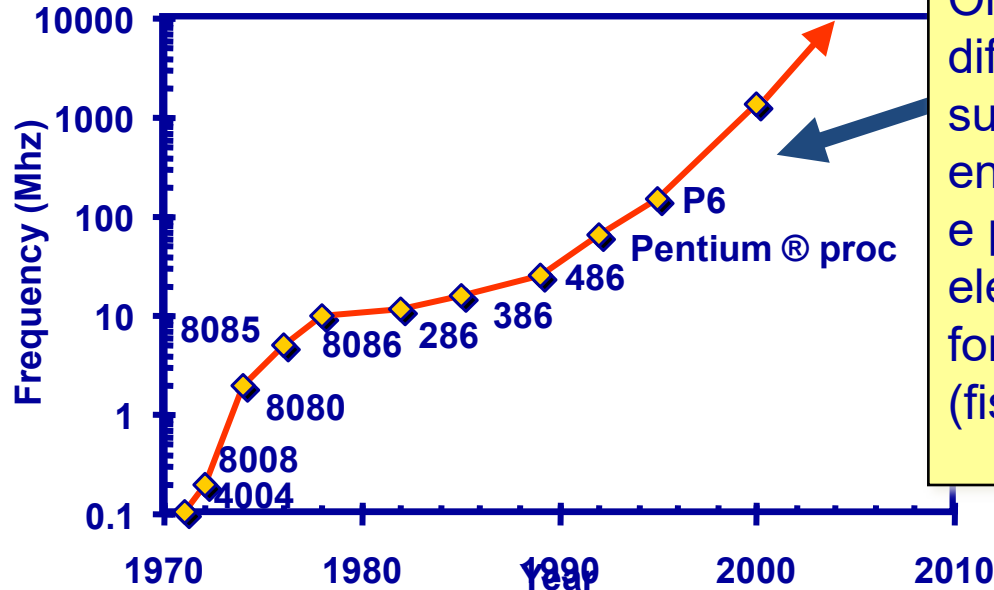
Il numero di transistor per mm² raddoppia ogni 24 mesi

- Più transistor in una CPU significano:
 - Eseguire direttamente istruzioni più complesse
 - Maggiore memoria sul chip (cache)
 - Maggiore parallelismo interno (+ core)
- Altro fattore importante è la *frequenza di funzionamento* ma è *intrinsecamente limitata*

Frequenza di clock (orologio)

La frequenza di Clock indica quante operazioni elementari si eseguono
1 Giga Hertz = 1 miliardo al secondo

La frequenza di clock raddoppia ogni due anni circa
Oltre 3GHz ci sono difficoltà da superare:
energia temperat.
e problemi elettrici
fondamentali
(fisica quantistica)



Legge di Moore

Miglioramenti tecnologici simili si sono avuti anche in altri aspetti

- Immagini (costo pixel)
- Memoria (costo e dimensioni)
- Velocità e costo delle comunicazioni (Internet)

Seconda legge di Moore

- I costi di progettazione e realizzazione delle fabbriche aumentano (es. costo progetto e realizzazione fabbrica per nuovo chip costa qualche miliardo di euro)
- Il numero di produttori si riduce (infatti oggi ci sono pochi produttori di chip)

Quanto possiamo progredire?

Siamo vicini a limiti fisici attuale tecnologia

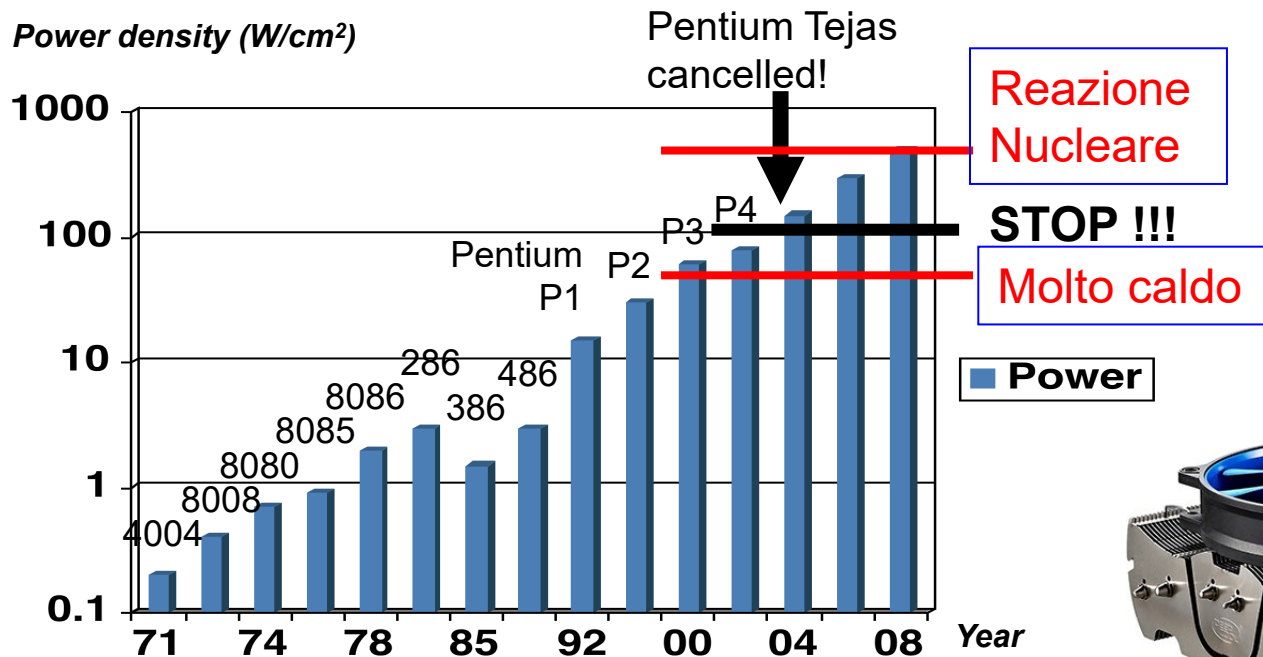
- Dimensione di un gate di un transistor (elemento base dei circuiti elettronici) 2014: 14 nm
- 2018-19: in commercio sono disponibili dispositivi mobili (TSMC) e processori desktop (AMD) con tecnologia a 7nm
- 1 nanometro = 1 milionesimo di un millimetro
- La dimensione di un atomo circa 0,1 nanometri;
una molecola di silicio circa 0,2 nanometri;
- Piccole dimensioni → aumento difetti fabbricazione → aumento dei costi (pezzi scartati)
- Circuiti oggi bidimensionali; uso chip tridimensionali (più compatti):
ci sono problemi tra cui il surriscaldamento

Quanto possiamo progredire?

Siamo vicini a limiti fisici attuale tecnologia

- Velocità della luce (non superabile secondo Einstein) = 300.000 Km al secondo
 - La luce in un nanosecondo percorre 30 cm
 - Se la frequenza di clock è 3 GHz, la luce in un ciclo di clock percorre 10 cm
 - I chip hanno dimensioni di 3-4 centimetri di lato
 - Non possiamo con questa tecnologia superare i 30-40 GHz con chip di circa 1 cm di lato (perché il tempo che la luce impiega a percorrere 1 cm è di 0,034 ns, che corrisponde quindi a circa 30GHz).
 - Inoltre maggiore è la frequenza maggiore è il **calore** sviluppato

Il calore sviluppato è molto alto



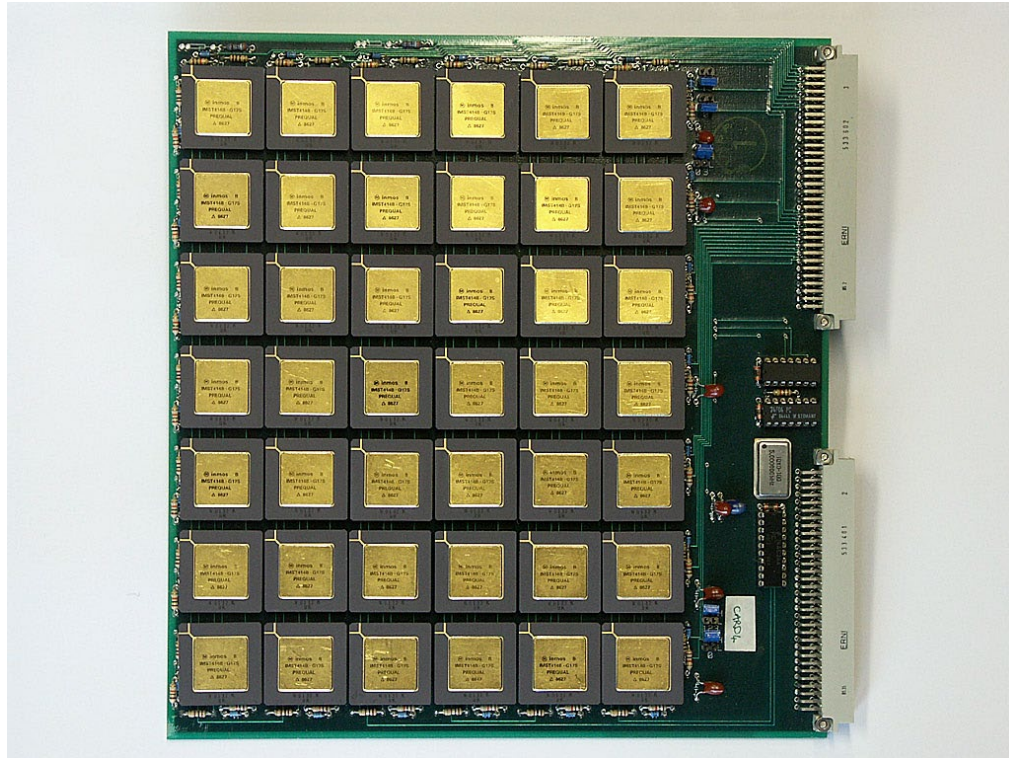
Mantenere la legge di Moore

- Stiamo raggiungendo i limiti fisici in termini di velocità e dimensione dei dispositivi (e calore sviluppato)
- Spesso annunci di nuove tecnologie sono realizzati con ritardo; gli annunci non sono sempre veritieri in assoluto
- Per migliorare è necessario usare le possibilità di avere circuiti più densi in modo diverso

Molte CPU (core) su un unico chip!

Cosa è questo?

1979 - David May's B0042 board - 42 Transputers

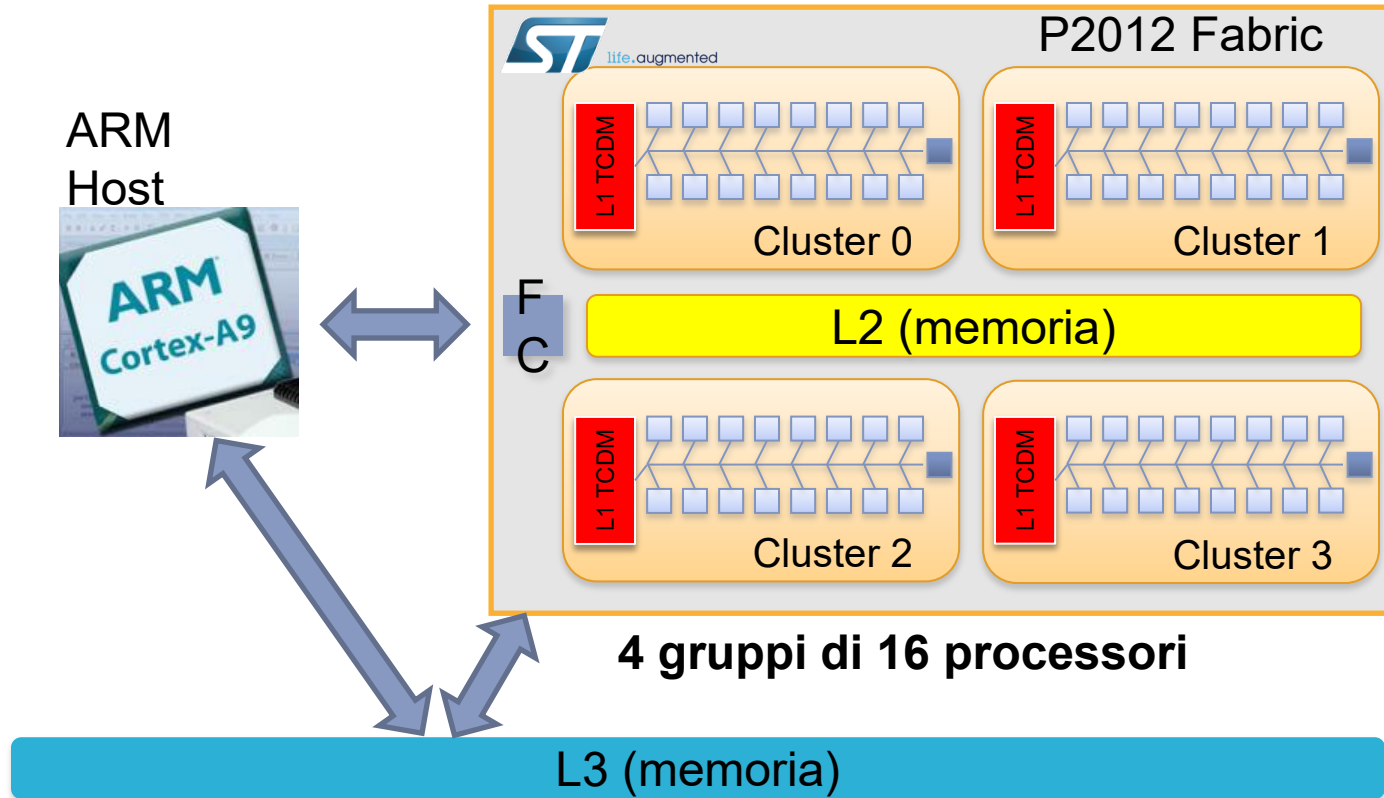


L'invasione multicore

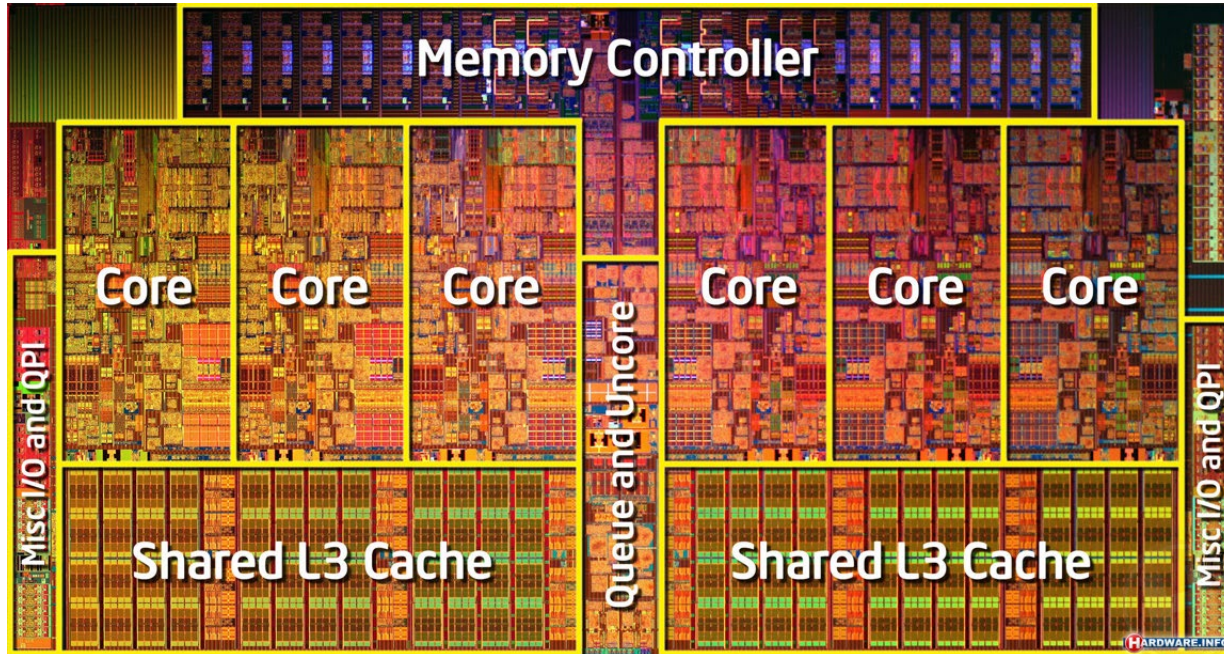
- Intel' s Core 2, Itanium, Xeon: 2, 4 cores
- AMD' s Opteron, Athlon 64 X2, Phenom: 2, 4 cores
- IBM' s POWER7: 8 cores
- IBM-Toshiba-Sony Cell processor: 8 cores (PSX3)
- Sun' s Niagara UltraSPARC: 8 cores
- Microsoft' s Xenon: 3 cores (Xbox 360)
- Tiler a' s TILE64: 64-core
- Others (network processors, DSP, GPU,...)

P2012 SoC: 64 core su un chip

Progetto architettura ST Microelectronics 2009-2012



Architetture multi core: il chip

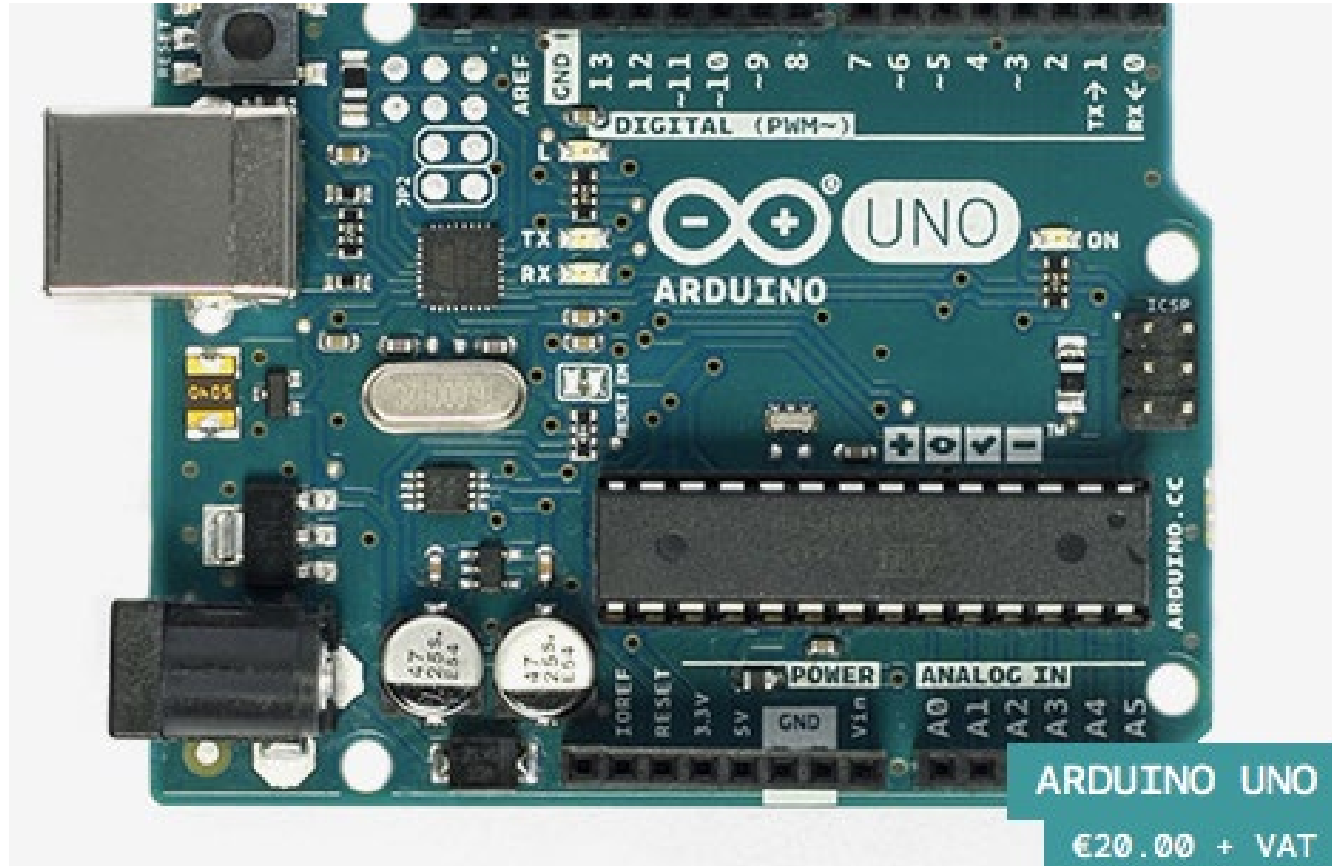


Layout del chip Core i7 Extreme Edition

Sistemi embedded

- Automobili: controllo sicurezza, condizioni di guida, ecc.
- Elettrodomestici
- Domotica
- Internet of things: l'internet degli oggetti

Arduino



Arduino

Scheda elettronica di **piccole dimensioni e basso costo** con un microcontrollore e circuiteria di contorno, utile per creare rapidamente prototipi e per scopi hobbistici e didattici.

Nota: Il nome della scheda deriva da quello di un bar di Ivrea dove si incontravano gli inventori di Arduino

Arduino

Con Arduino si possono realizzare rapidamente piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, temperatura e umidità e molti altri progetti che utilizzano sensori comunicando con altri dispositivi.

È fornito di un semplice ambiente di sviluppo integrato per la programmazione. Tutto il software a corredo è libero, e gli schemi circuitali sono distribuiti come hardware libero.