

Università degli Studi di Bologna

Area Medica

Insegnamento **INFORMATICA**

Lezione 01

Bit (binary digit)

In informatica il bit è l'unità di misura elementare dell'informazione.

Il bit può assumere solo due valori "0" o "1"

A ciascuno dei due suoi stati possiamo associare una informazione.

Ad esempio consideriamo un interruttore collegato ad una lampadina:

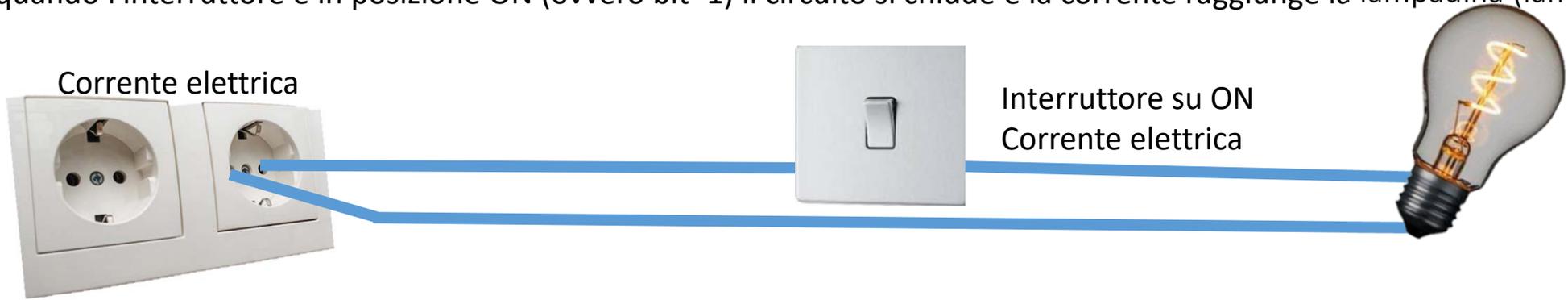
- se posizionato su "ON" (bit con valore="1") determina l'accensione della lampada
- Se posizionato su "OFF" (bit con valore="0") determina lo spegnimento della lampada

Così facendo associamo ad un singolo bit una informazione precisa:

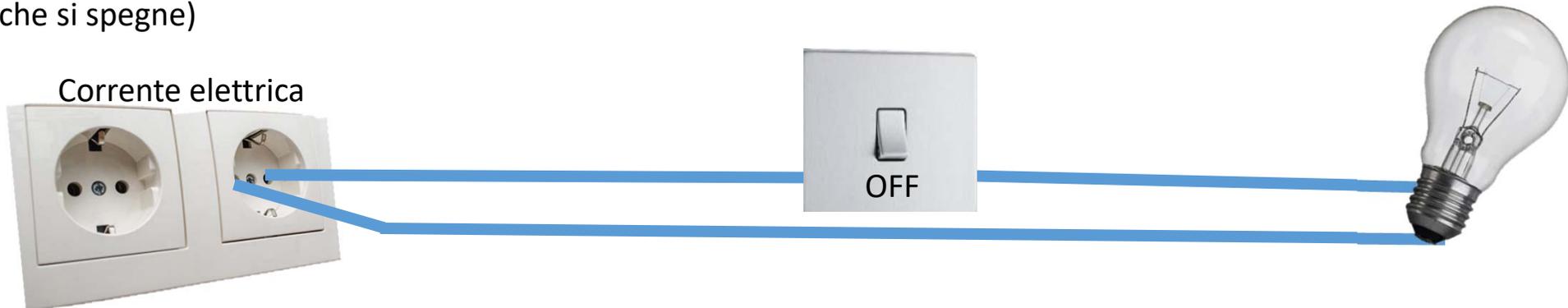
- bit ad "1" (cioè su ON) --> "luce accesa"
- bit a "0" (cioè su OFF) --> "luce spenta"

Bit (binary digit)

quando l'interruttore è in posizione ON (ovvero bit=1) il circuito si chiude e la corrente raggiunge la lampadina (lamp. accesa)



quando l'interruttore è in posizione OFF (ovvero bit=0) il circuito si apre e la corrente non arriva al terminale della lampadina (che si spegne)



Bit (binary digit)

Così facendo si associa allo stato di un singolo bit un'informazione:

Codifica: "1" = lampadina accesa" e "0" = lampadina spenta.

Questo concetto elementare è alla base dell'elettronica digitale che ha permesso la nascita dei computer: attraverso la **combinazione** di molti binary digit (cifre binarie) si possono denotare miliardi di informazioni differenti.

Il computer lavora con circuiti digitali ovvero linee elettriche che possono, in un certo istante, essere percorse da corrente oppure no.

Bit (binary digit)

Ed è per questo motivo che nell'ambito informatico l'unità di misura è il "bit".

Il sistema di numerazione binario è una sequenza di una o più cifre ciascuna delle quali può solo assumere solo uno dei due valori: **uno** o **zero**,
Con i bit viene codificata tutta l'informazione all'interno di un calcolatore.

Sistema di numerazione

Un sistema di numerazione è un modo di esprimere e rappresentare i numeri attraverso un insieme finito di simboli.

Esistono due tipologie di sistemi di numerazione:

- Sistema di numerazione non posizionale
- Sistema di numerazione posizionale

Posizionale e Non posizionale

In un sistema di numerazione posizionale i simboli assumono valori diversi a seconda della posizione che occupano in una stringa.

Viceversa, in un sistema non posizionale il valore attribuito a ciascun simbolo è sempre lo stesso indipendentemente dalla posizione che occupa.

N.B. con stringa indichiamo una sequenza di simboli giustapposti

Sistema di numerazione non posizionale

Sistema di numerazione unario

Il più banale è il sistema di numerazione unario che ha come alfabeto (alfabeto = l'insieme dei simboli ammessi in un determinato sistema di numerazione) un unico simbolo (ad esempio "I") che varrà sempre 1 qualunque sia la sua posizione all'interno della stringa.

Con questo semplice sistema è possibile creare una numerazione solo giustappponendo più simboli "I" uno dopo l'altro:

$$I_1 = 1_{10}; II_1 = 2_{10}; III_1 = 3_{10}; \dots; I_1, II_1, III_1, \dots = n_{10}$$

In questo caso la posizione del simbolo "I", essendo unico, dovunque lo si sposti varrà sempre 1.

Nota bene: il pedice 10 indica che sono i simboli del sistema decimale e il pedice 1 indica che sono simboli del sistema unario.

Sistemi di numerazione non posizionale

Il sistema di numerazione Romano è un sistema **additivo/sottrattivo** non posizionale.

In questo sistema l'insieme dei simboli ammessi (alfabeto) è:

$$\text{Simboli}_{\text{RM}} = \{I, V, X, L, C, D, M\}$$

Con le seguenti equivalenze nel sistema decimale:

$$I = 1; V = 5; X = 10; L = 50; C = 100; D = 500; M = 1\ 000$$

Quando si giustappongono più simboli consecutivi, il simbolo che precede è maggiore del successivo il valore è ottenuto dalla somma (additivo), quando il successivo è maggiore del precedente il valore è ottenuto dalla differenza (sottrattivo).

Sistemi di numerazione non posizionale

Sistema di numerazione Romano

Anche in questo sistema ciascun simbolo assume lo stesso valore in qualunque posizione si trovi all'interno di una stringa.

Ad esempio considerando due numeri romani IV_{RM} e VI_{RM} questi, pur avendo un valore numerico in decimale differente ($IV_{RM} = 5 - 1 = 4_{10}$ e $VI_{RM} = 5 + 1 = 6_{10}$) valgono, in entrambi i casi: il simbolo $I_{RM} = 1_{10}$ e il simbolo $V_{RM} = 5_{10}$.

Sistemi di numerazione posizionale

Di contro in un sistema posizionale (come quello decimale) ad esempio i numeri 15_{10} e 51_{10} il potere numerico del simbolo **1** di 15_{10} è differente del potere numerico del simbolo **1** di 51_{10} e lo stesso vale anche per il **5**. In 15_{10} il simbolo **1** vale 10 (decine) e in 51_{10} il simbolo **1** vale 1 (unità) e allo stesso modo il **5** del primo vale 50_{10} (decine) e il **5** del secondo vale 5 (unità)

Sistema di numerazione posizionale

Consideriamo il numero 354 ricordiamo che è costituito da:

300 = centinaia

50 = decine

4 = unità

La cui somma $300+50+4$ ci riporta al numero 354

Il "peso" di ciascun numero è determinato dalla posizione in cui si trova ed è per questo che il nostro è considerato un sistema di numerazione **posizionale**

Sistema di numerazione posizionale

Il sistema di numerazione posizionale necessita dello zero per occupare con valore nullo la posizione della cifra che occupa all'interno della stringa

Ciascun alfabeto è composto da una quantità finita di simboli.

Ad esempio:

- Il sistema decimale è costruito su 10 simboli
- Il sistema binario è costruito su 2 simboli
- Il sistema esadecimale è costruito su 16 simboli

Sistema di numerazione decimale

Sistema Posizionale

È un sistema in **base 10**.

L'insieme dei simboli ammessi al sistema sono: 0,1,2, ..., 9

Le potenze sono espresse nella forma 10^n con $n \in \mathbb{N}$ (ma vanno da 0 a n-1)

$$10^0 = 1$$

$$10^1 = 10$$

$$10^2 = 10 * 10$$

....

$$10^n = 10 * 10 * \dots * 10 \quad (\text{per } n\text{-volte})$$

Sistemi di numerazione posizionale

Definizione formale di sistema di numerazione posizionale

1. si sceglie un qualsiasi numero naturale b (diverso da zero e da uno), chiamato base;
2. si scelgono m simboli diversi (l'alfabeto del sistema ovvero l'insieme dei simboli ammessi)
3. Si sceglie il numero di cifre n
4. si compongono i numeri tenendo presente che **il valore di ogni cifra va moltiplicato per:**
 b^0 cioè 1 (unità) se è l'ultima cifra alla destra del numero che stiamo considerando
 b^1 cioè b se è la seconda cifra
 b^2 se è la terza cifra
e così via fino a $b^{(n-1)}$ se n è la n -esima cifra

N.B. le cifre sono n ma le potenze vanno da 0 a $n-1$

Esempio se $n=4$ le potenze saranno 0,1,2,3

La somma di tutti i valori così ottenuti è il numero che stiamo considerando

Il massimo numero rappresentabile con n cifre è pari a b^n-1

Sistema di numerazione decimale

Sistema Posizionale

Esempio di un numero decimale a **3 cifre**: dove $b=10$; $m=\{0,1,2,..,9\}$; $n=3$
 Essendo $n=3$ le potenze vanno da 0 a $n-1$ ovvero da 0 a $(3-1)$ cioè da 0 a 2

$$675 = \boxed{(6 \cdot 10^2)}_{\substack{\text{centinaia} \\ 1^\circ \text{ cifra}}} + \boxed{(7 \cdot 10^1)}_{\substack{\text{decine} \\ 2^\circ \text{ cifra}}} + \boxed{(5 \cdot 10^0)}_{\substack{\text{unità} \\ 3^\circ \text{ cifra}}} = 600 + 70 + 5 = 675$$

$m=\{0,1,2,\dots,9\}$ $b=10$ $n=3$

Atro Esempio

$$14510 = \underbrace{1}_{1} \text{ decine di migliaia} \quad \underbrace{4}_{4} \text{ unita' di migliaia} \quad \underbrace{5}_{5} \text{ centinaia} \quad \underbrace{1}_{1} \text{ decine} \quad \underbrace{0}_{9} \text{ unita'}$$

Sistema di numerazione decimale

10^2	10^1	10^0
5	6	4

$$564 = (5 * 10^2) + (6 * 10^1) + (4 * 10^0)$$

Simboli dell'alfabeto $m = \{0, 1, 2, \dots, 9\}$

Base $b = 10$

Cifre $n = 3$

Ciascun numero va moltiplicato, a seconda della sua posizione nella stringa, per b^0, b^1, b^{n-1} ovvero $10^0, 10^1, 10^{3-1}$ ovvero $10^0, 10^1, 10^2$. (Le potenze della base vanno da 0 a $n-1$ partendo da sinistra ovvero dalla cifra meno significativa)

Sistema di numerazione decimale

10^3	10^2	10^1	10^0
9	9	9	9

$$9999_{10} = (9 \cdot 10^3) + (9 \cdot 10^2) + (9 \cdot 10^1) + (9 \cdot 10^0) = 9999_{10}$$

n = numero di cifre

Il massimo numero che si può ottenere con n cifre è: $10^n - 1$

Quindi con 4 cifre si ha = $10^4 - 1 = 10000 - 1 = 9999$



Sistema di numerazione decimale

$$32 = 30 + 2 = (10^1 * 3) + (10^0 * 2)$$

$$642 = 600 + 40 + 2 = 10^2 * 6 + 10^1 * 4 + 2 * 10^0$$

Sistema di numerazione binario

b è la Base: $b=2$;

m sono i simboli ammessi al sistema (alfabeto): $m \in \{0, 1\}$

n è Il numero di cifre

Le potenze sono espresse nella forma 2^n con $n \in \mathbb{N}$ (\mathbb{N} =insieme dei numeri naturali)

N:B: Le potenze vanno da 0 a n-1

Ogni singola cifra è un bit (**binary digit**)

Esempio:

In base **$b=2$** , con simboli **$m=2$** e un numero di cifre **$n=4$**

posso rappresentare numeri nell'intervallo CHIUSO

$$[0, 2^n - 1] = [0, 2^4 - 1] = [0_{10}, 15_{10}] = [0000_2, 1111_2]$$

Conversione binario - decimale

Esempio: $1010_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 0 + 2 + 0 = 10_{10}$

2^3	2^2	2^1	2^0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



Decimale
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Equivalenza tra
sistema binario e decimale



Sistema Esadecimale

Definizione formale

Base $b = 16$;

Alfabeto $m_{16} = \{0,1,2,\dots,9,A,B,C,D,E,F\}$; $m_{10} = \{0,1,2,\dots,9,10,11,12,13,14,15\}$; **16 SIMBOLI**

Cifre $n = 1,2,\dots$

Conversione Esadecimale – decimale: Esempio $B2_{16}$

Con $b=16$, $n=2$, $m=2 \rightarrow B2_{16} = (16^1 * B) + (16^0 * 2) = 16 * 11 + 1 * 2 = 176 + 2 = 178_{10}$

Conversione Esadecimale - decimale - binario: Esempio $B2_{16}$

Ciascuna cifra esadecimale equivale a 4 bit pertanto è possibile convertire ciascuna cifra separatamente con 4 bit: convertiamo separatamente in decimale $B_{16} = 11_{10}$ e $2_{16} = 2_{10}$

$B_{16} = 11_{10} = 11:2=5$ resto **1**; $5:2=2$ resto **1**; $2:2=1$ resto **0**; $1:2=0$ resto **1** = 1011_2

$2_{16} = 2_{10} = 2:2=1$ resto **0**; $1:2=0$ resto **1** = $10_2 = 0010_2$

$B2_{16} = 11_{10} = 10110010_2$

Conversione Hex - Dec e Hex - Bin

$$B2_{\text{hex}} = (11 * 16^1) + (2 * 16^0) = 176_{10} + 2_{10} = 178_{10}$$

$$178_{10} = (1 * 10^2) + (7 * 10^1) + (8 * 10^0) = 178_{10}$$

Converto 178₁₀ in binario : (R=resto) Risultato = 10110010

178:2=89 R=0; 89:2=44 R=1; 44:2=22 R=0; 22:2=11 R=0; 11:2=5 R=1; 5:2=2 R=1
2:2 =1 R=0; 1:2=0 R=1

Converto B2_{hex} in binario

Converto separatamente $B_{\text{hex}} = 11_{10} = (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 1011$

Converto separatamente $2_{\text{hex}} = 2_{10} = (0 * 2^3) + (0 * 2^2) + (0 * 2^1) + (0 * 2^0) = 0010$

Giustappongo i risultati: 10110010

N.B. ricorda ciascuna cifra esadecimale è costituita da 4 cifre binarie

Sistema di numerazione Hex vs Bin

16^3	16^2	16^1	16^0
F	F	F	F

$$FFFF_{16} = (F \cdot 16^3) + (F \cdot 16^2) + (F \cdot 16^1) + (F \cdot 16^0) = 15_{10}$$

n= numero di cifre=4

$$16^n - 1 = 16^4 - 1 = 65536 - 1 = 65535$$

$$1111111111111111_2 \quad n=\text{numero di cifre}=16$$

$$2^n - 1 = 2^{16} - 1 = 65536 - 1 = 65535$$

Sistema di numerazione Esadecimale

Da binario a esadecimale considerando separatamente 4 bit alla volta

$$01010110 = 56_{16}$$

convertito separatamente $0101 = 5$

convertito separatamente $0110 = 6$

Conversione decimale - binario

Per convertire un numero da decimale a binario basta fare la divisione del numero per due, dividere il quoziente ottenuto ancora per due, e così via fino ad ottenere quoziente zero.

Il numero binario corrispondente si ottiene prendendo i resti a partire dall'ultimo.

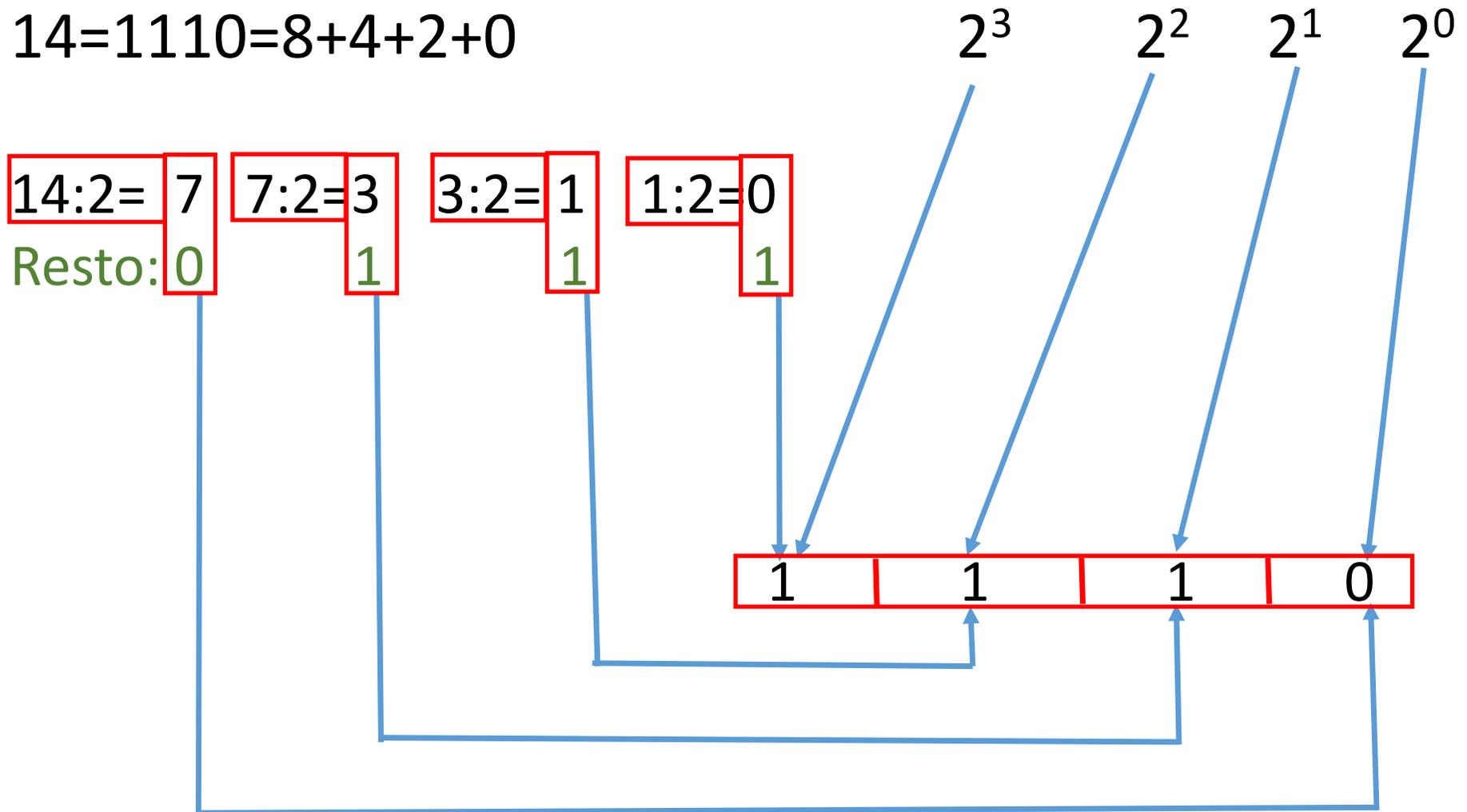
Esempio 14_{10}

$$\begin{array}{cccc}
 14:2=7 \text{ resto } 0 & 7:2=3 \text{ resto } 1 & 3:2=1 \text{ resto } 1 & 1:2=0 \text{ resto } 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 2^0 & 2^1 & 2^2 & 2^3
 \end{array}$$

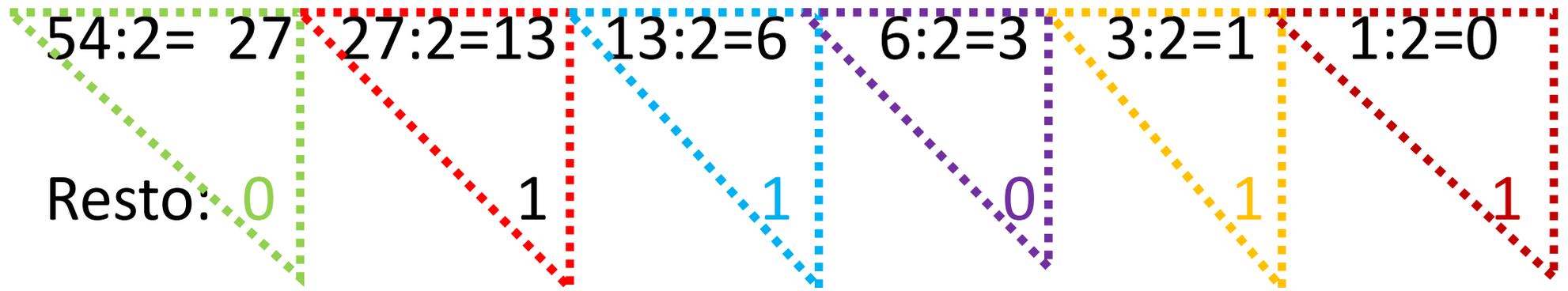
$$14_{10} = 0 \quad 1 \quad 1 \quad 1 = 1110_2$$

Esempio conversione DEC to BIN e BIN to DEC

$$14 = 1110 = 8 + 4 + 2 + 0$$



Esempio di conversione DEC a BIN



Risultato: $54_{10} = 110110_2$

110110_2

ESEMPI

Conversione da decimale a binario

$$12_{10} = 12:2=6; 6:2=3; 3:2=1; 1:2=0$$

$$\text{Resto} \quad 0 \quad 0 \quad 1 \quad 1 \quad \rightarrow 1100_2$$

Conversione da binario a decimale

$$1 \quad 1 \quad 0 \quad 0$$

$$2^3 + 2^2 + 0 + 0 = 8 + 4 + 0 + 0 = 12_{10}$$

Esempio: conversione Binario a Decimale e Decimale a Binario

$$101101_2 = 32 + 8 + 4 + 1 = 45_{10}$$

$$45:2=22 \quad 22:2=11 \quad 11:2=5 \quad 5:2=2 \quad 2:2=1 \quad 1:2=0$$

$$\text{resto: } 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1$$

$$\text{Risultato: } 45_{10} = 101101_2$$

$$89:2=44 \quad \text{Resto}=1$$

$$44:2=22 \quad \text{Resto}=0$$

$$22:2=11 \quad \text{Resto}=0$$

$$11:2=5 \quad \text{Resto}=1$$

$$5:2=2 \quad \text{Resto}=1$$

$$2:2=1 \quad \text{Resto}=0$$

$$1:2=0 \quad \text{Resto}=1$$

$$1011001=$$

$$64+16+8+1=89$$

Esempi

Da binario a decimale: $00101101 = 45_{10}$

Da esadecimale a decimale = $2D = 16^1 \times 2 + 16^0 \times 13 = 45_{10}$

Per la conversione binario a decimale basta convertire separatamente blocchi da 4 bit (se il numero di blocchi di bit è inferiore a 4 occorre anteporre degli zero alla cifra più a sinistra)

Esempio: $10 \rightarrow 0010$; $11011 \rightarrow 00011011$

Esempi

Conversione binario esadecimale

$$00101101_2 = 2D_{16} \quad \text{calcolo separatamente}$$

$$0010_2 = 2_{10} \quad \text{e} \quad 1101 = 8+4+1 = 13_{10} = D_{16}$$

Conversione binario a decimale

$$00101101_2 = 32+8+4+1 = 45_{10}$$

Conversione da esadecimale a decimale

$$2D = 16^1 \times 2 + 16^0 \times D = 32 + 13 = 45_{10}$$

Sistema di numerazione ottale

Sistema ottale

Alfabeto = {0,1,2,3,4,5,6,7} ,

b=8

Esempio: Conversione da ottale a decimale con n=2

$$57_8 = 8^1 \times 5 + 8^0 \times 7 = 40 + 7 = 47_{10}$$

$$17 = 8^1 \times 1 + 8^0 \times 7 = 15_{10}$$

$$16 = 8^1 \times 1 + 8^0 \times 7 = 14_{10}$$

Esempi di Conversioni

da esadec. a decimale

$$56_{16} = 16^1 \times 5 + 16^0 \times 6 = 80 + 6 = 86_{10}$$

da decimale a binario:

$$86_{10} = 86:2=43 \quad 43:2=21 \quad 21:2=10 \quad 10:2=5 \quad 5:2=2 \quad 2:2=1 \quad 1:2=0$$

$$\text{Resto} \quad \quad \quad 0 \quad \quad \quad 1 \quad \quad \quad 1 \quad \quad \quad 0 \quad \quad \quad 1 \quad \quad \quad 0 \quad \quad \quad 1$$

Da binario a decimale:

$$01010110 = 6 + 16 + 64 = 86_{10}$$

Da binario a esadecimale considerando separatamente 4 bit alla volta

$$01010110 = 56_{16}$$

$$\text{converto separatamente } 0101 = 5$$

$$\text{converto separatamente } 0110 = 6$$

Esempi

Da binario a decimale

$$1 \ 1 \ 1 \ 0 \ 1_2 =$$

$$2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 16 \times 1 + 8 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 16 + 8 + 4 + 0 + 1 = 29_{10}$$

Da decimale a binario

$$14_{10} = 14:2=7 \quad 7:2=3 \quad 3:2=1 \quad 1:2=0$$

$$\text{Resto:} \quad \quad 0 \quad \quad 1 \quad \quad 1 \quad \quad 1 \quad = 1110_2$$

Da binario a decimale

$$1 \ 1 \ 1 \ 0_2 = 8 + 4 + 2 + 0 = 14_{10}$$

Sistemi a confronto

Sistema binario

Alfabeto₂ $\in\{0, 1\}$

con $m=2$

Sistema decimale

Alfabeto₁₀ $\in\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

con $m=10$

Sistema esadecimale

Alfabeto₁₆ $\in\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

con $m=16$

Decimale	2^3	2^2	2^1	2^0	EsaDecimale
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	A
11	1	0	1	1	B
12	1	1	0	0	C
13	1	1	0	1	D
14	1	1	1	0	E
15	1	1	1	1	F



Equivalenza tra
 Sistema decimale,
 binario e esadecimale

Alcuni Sistemi di misura

- Il sistema metrico decimale (S.M.D.) è un sistema in base 10 e il rapporto tra multipli e sottomultipli di ogni sua unità di misura è sempre espresso con 10 o una potenza di 10.

Esempio: il peso medio di una persona è circa $62\text{Kg} = 62 * 10^3 = 62 * 1000 = 62000\text{g}$

- L'unità di misura della potenza della corrente elettrica è espressa in Watt e il rapporto tra i suoi multipli è espresso con 10 o con una potenza di 10.

Esempio $3\text{KW} = 3 * 10^3 = 3 * 1000 = 3000\text{W}$; e $3\text{GW} = 3 * 10^9 = 3 * 1000000000\text{W}$.

Bit, byte, Megabyte, Gigabyte, Terabyte..

Definizione di byte: è l'unità di misura della capacità di memoria di un calcolatore è costituita da una sequenza di 8 bit dove il primo bit a sinistra è chiamato **Least significant bit** e il primo bit a destra è chiamato **Most significant bit**. I multipli del Byte sono: **kbyte Mbyte, Gbyte, Terabyte**

Nel sistema di numerazione binaria abbiamo potenze di 2 (differente dal sistema decimale dove il K corrisponde a 10^3)

Nel sistema binario **1Kbyte** equivale a **2^{10}** cioè **1024byte** che corrisponde al numero binario **0100 0000 0000**

Questo comporta che tutti i multipli del byte cioè kbyte Mbyte, Gbyte, Terabyte sono espressi in potenze in base 2.

Bit, byte, Mbyte, Gbyte, Tbyte, Pbyte

Reminder: 1byte= 8 bit = $2^8 = 256$ informazioni diverse

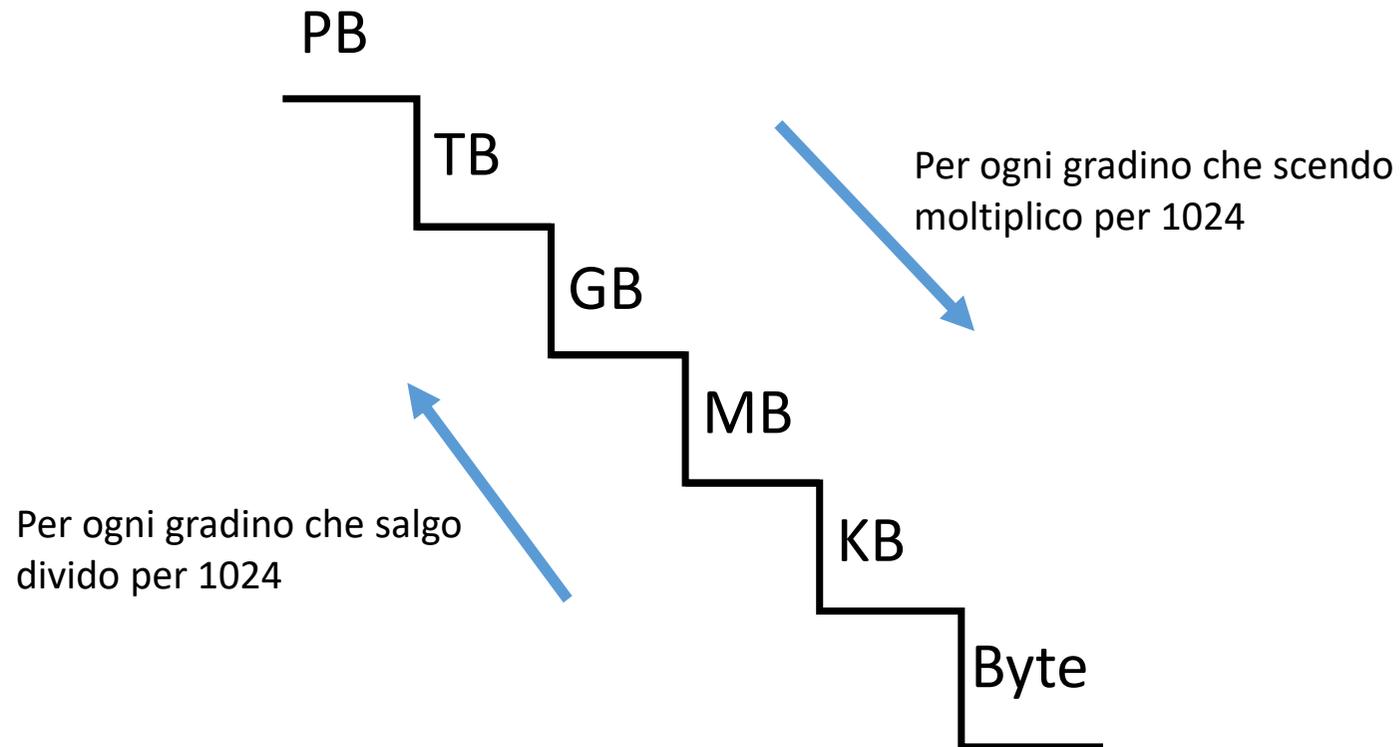
- kilobyte (KB) = $2^{10} = 1024$ byte
- megabyte (MB) = $2^{20} = 1.048.576$ (circa un milione di) byte = 1024 kilobyte (KB)
- gigabyte (GB) = $2^{30} = 1.073.741.824$ (circa un miliardo di) byte = 1024 megabyte (MB)
- terabyte (TB) = $2^{40} = 1.099.511.627.776$ (circa mille miliardi di) byte = 1024 gigabyte (GB)
- petabyte (PB) = $2^{50} = 1.125.899.906.842.624$ (circa un milione di miliardi di) byte = 1024 terabyte

Metodo per la conversione

Per passare da byte a kbyte basta dividere per 1024, per passare da Mbyte a Kbyte basta moltiplicare per 1024

Per passare da kbyte a byte basta moltiplicare per 1024

Scala del Byte



Esempi di conversione

- A quanti KByte corrisponde un file da: 1,5Mbyte?
Risposta: $1,5\text{Mbyte} \times 1024 = 1536 \text{ Kbyte}$
- A quanti MByte corrisponde un file da 0.9 Tbyte?
Risposta: $0,9\text{Tb} \times 1024 = 921,6 \text{ Gbyte} ;$
 $921,6 \text{ Gbyte} \times 1024 = 943.718,4 \text{ Mbyte}$
- A quanti Mbyte corrisponde un file da 665,6 Kbyte?
Risposta: $665,6 : 1024 = 0,65\text{Mbyte}$

Conversione da Bit a Byte e viceversa

Reminder: $1 \text{ byte} = 8 \text{ Bit}$ e quindi $1 \text{ Bit} = 1:8 = 0,125 \text{ Byte}$

Esempio:

- Una rete in fibra ottica ha una connessione ad una velocità di 1GBit al secondo, a quanti Byte al secondo corrisponde?

Soluzione: $1 \text{ Mbit} = 1.000.000 \text{ bit} = 1.000.000 \text{ bit} : 8 = 125000 \text{ Byte/s}$;
e $125000 \text{ Byte} = 125000 \text{ Byte} : 1024 = 122,07 \text{ Kbyte/s}$

- Un Computer trasferisce un file su un HD ad una velocità di 5KByte/s a quanti Bit/s trasferisce i dati?

Soluzione: $5 \text{ Kbyte} = 5 \times 1024 = 5120 \text{ byte}$; $5120 \times 8 = 40960 \text{ bit/s}$

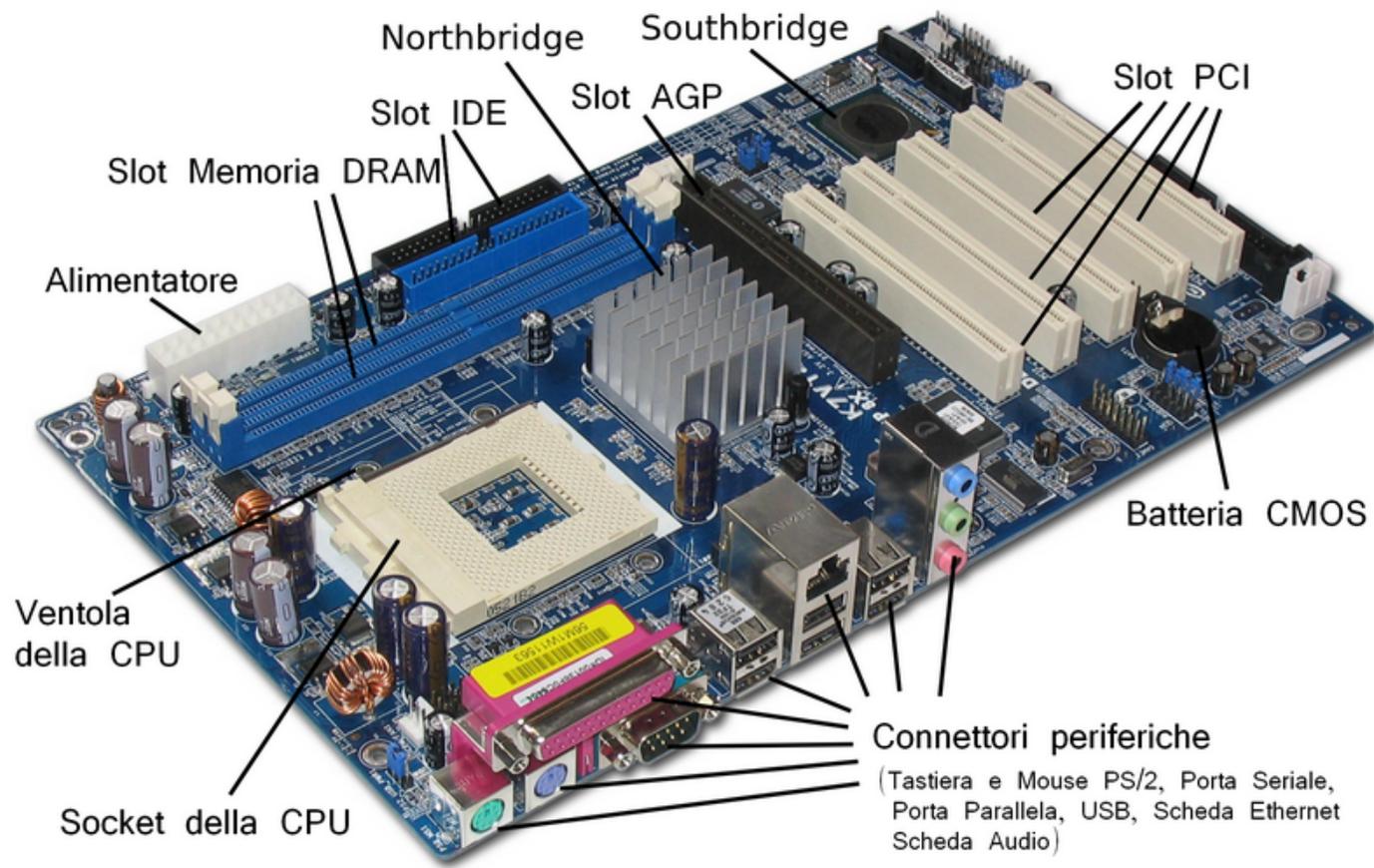
Informatica (da Treccani)

informàtica *s. f.* [dal fr. *informatique*, comp. di **informat(ion)** e **(automat)ique** «**informazione automatica**», termine coniato da Ph. Dreyfus (1962)]. – L'insieme dei vari aspetti scientifici e tecnici che sono specificamente applicati alla raccolta e al trattamento dell'informazione e in particolare all'elaborazione automatica dei dati, come sussidio e supporto alla documentazione, alla ricerca e allo studio nei vari settori della scienza, della tecnica, delle attività economiche, sociali, e anche pratiche: l'informatica applicata alle scienze, al diritto (e alla documentazione giuridica), alla medicina, alla linguistica, alla gestione aziendale, ecc.

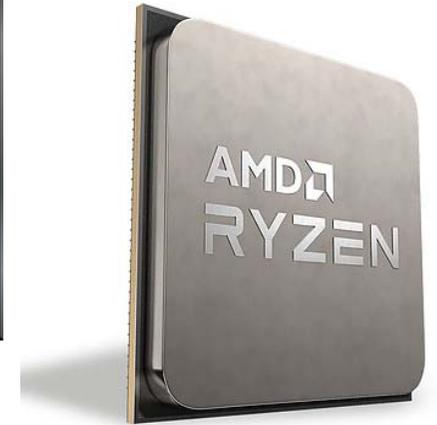
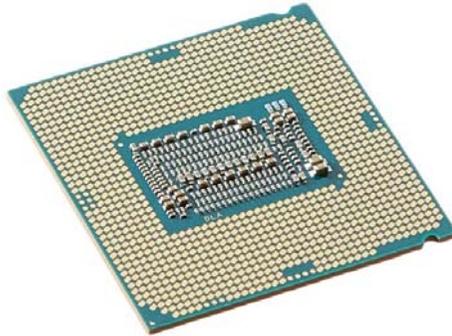
Hardware in un calcolatore.



Scheda madre



CPU



Caratteristiche della CPU

Marca del processore: Intel, AMD

Tipo di processore:

Numero di Core: 1 – 2 – 4,- 6, ...

Numero di thred: 2 – 4

Numero di bit: 8 – 16 -23 -64 ..

Dimensione della Cache: (in Mega byte) 4 – 6 ...

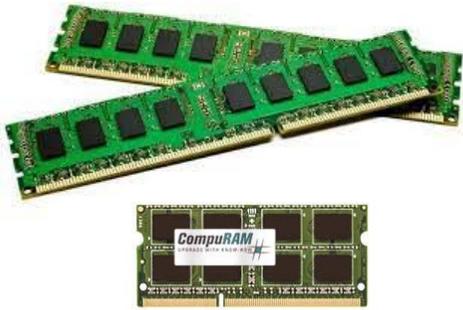
Frequenza di clock (in GHz): 2,4 – 3,4 – 3,7 - 4,6 ...

Periferiche



Memorie

Volatili: RAM



B.I.O.S.



Non volatili: Hard disk , SSD



Caratteristiche della memoria non volatile:

Tecnologia: SSD (stato solido)- HD (magnetico) – DVD (ottico)

Dimensione: 256Gbyte – 500Gbyte- 1T.byte ..

Velocità di scrittura/lettura SSD: circa 550Mbyte al secondo

Velocità di scrittura lettura HD: max 125Mbyte

Caratteristiche memoria volatile (RAM) Random Access Memory):

Dimensione (Gbyte): 8, 16, 32, 64,

Frequenza di lettura/scrittura: 1600 MHz, 2933 MHz ..

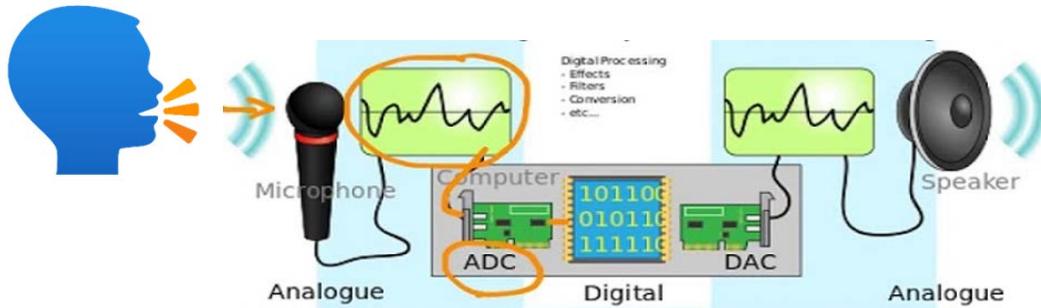
Dispositivi di INPUT OUTPUT



Rappresentazione dell'informazione - Sistemi di codifica A/D e D/A - (convergenza digitale).



→ 1 01001011011101...



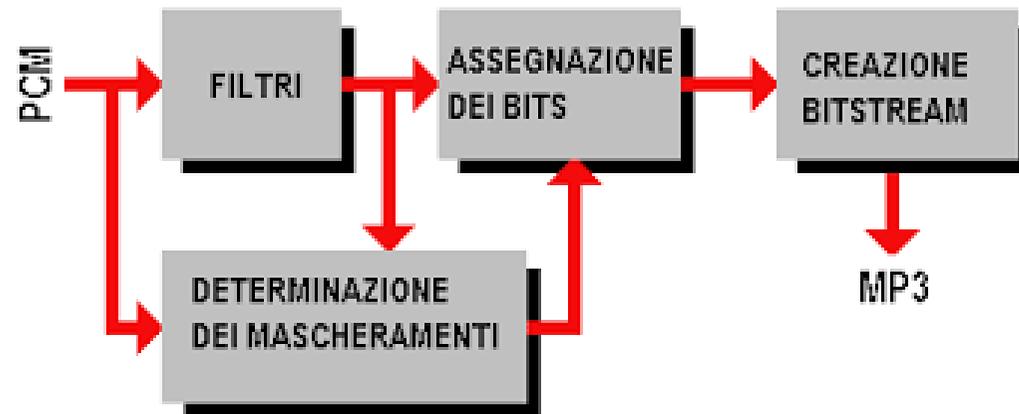
Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00000000	0	Null	00100000	32	SpC	01000000	64	@	01100000	96	.
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00101000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00100001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00100010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00100011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00101000	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00101001	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00101010	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00101011	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00101100	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00101101	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00101110	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00101111	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111000	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111001	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111010	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111011	63	?	01011111	95	_	01111111	127	~



ADC 1010010110111010010001... DAC



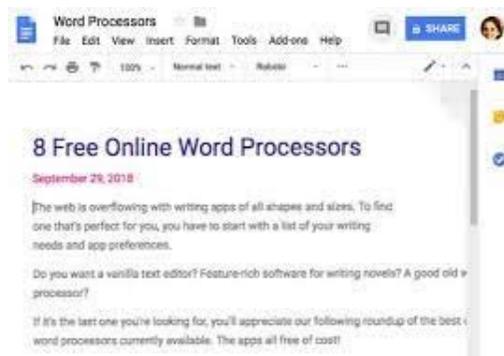
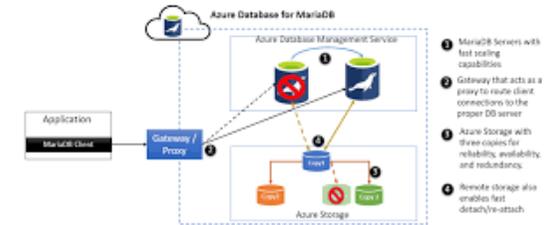
Sistemi di compressione loss e lossless



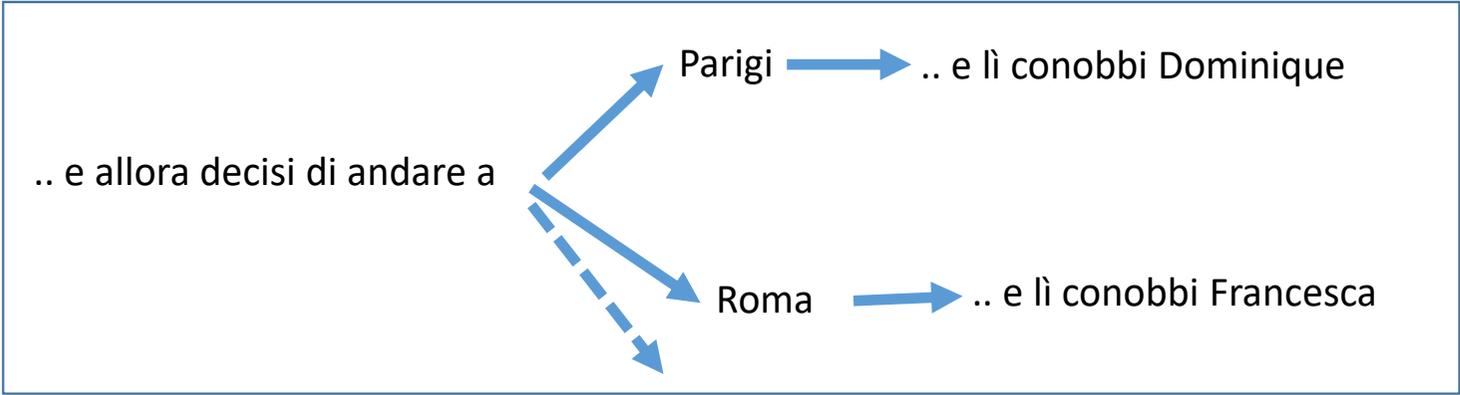
Editor di testi, Spreadsheet, Power Point ...



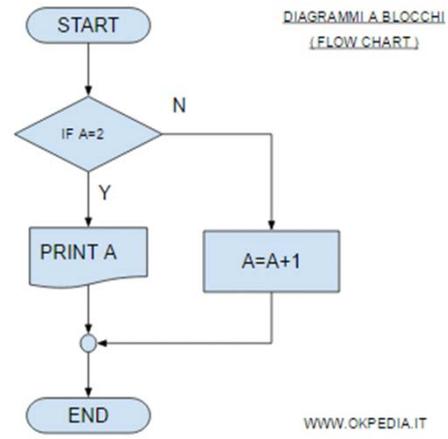
Built-in High Availability
99.99% Uptime SLA



Sistema Multitasking, Multimediale, Iperestuale/Ipermediale, Multimodale.



Sistemi operativi, algoritmi e cenni di programmazione



```
int input (int m, int n) {
n = int(input('Inserire n: '))

c = s = 0

for i in range(1, min(m, n)+1):
    if m%i == 0 and n%i == 0:
        print(i)
        c+=1
        s+=i
```

Internet, reti informatiche, protocolli di trasmissione http, p2p e applicativi.

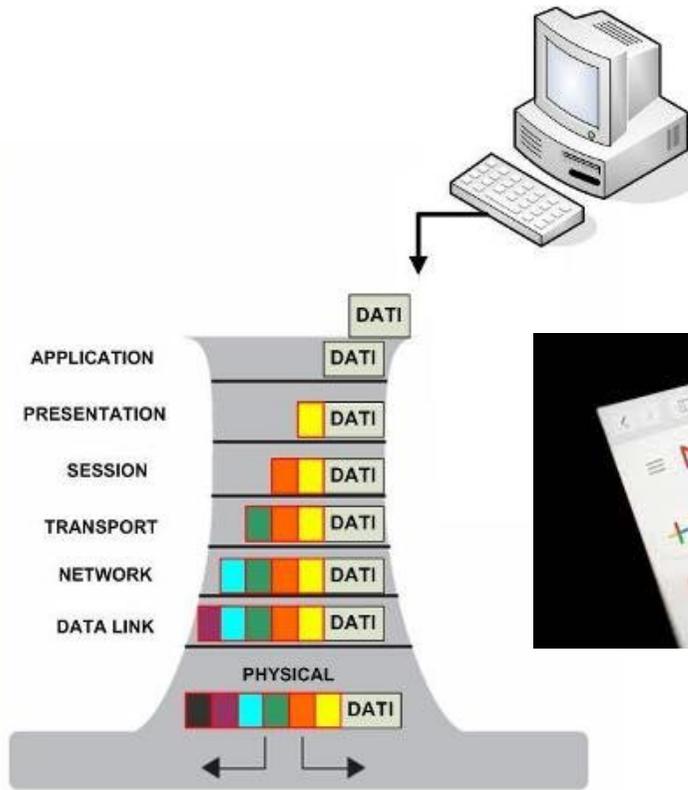
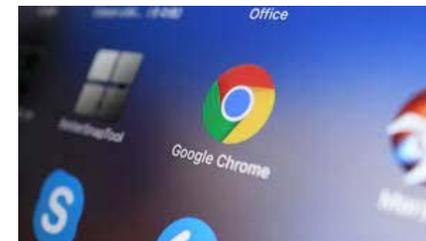


Figura 9



Firma digitale, hash (crittografia) , marca temporale, PEC, PCT.



HASH
L'IMPRONTA
DIGITALE DI
UN FILE

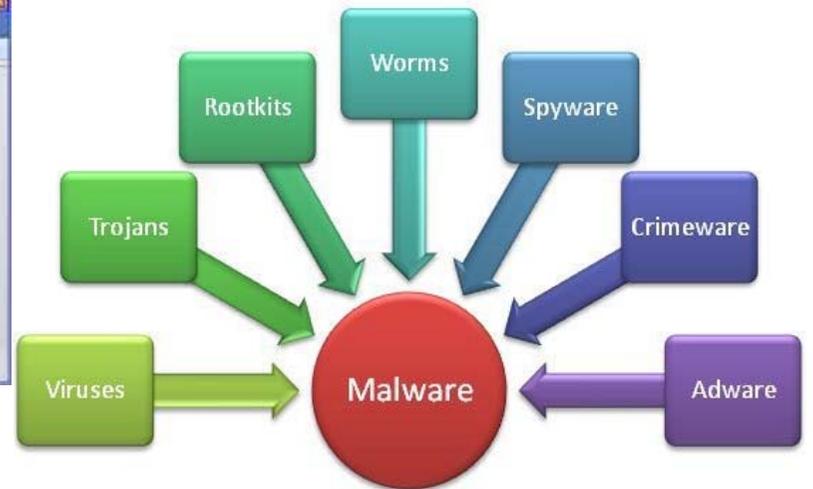


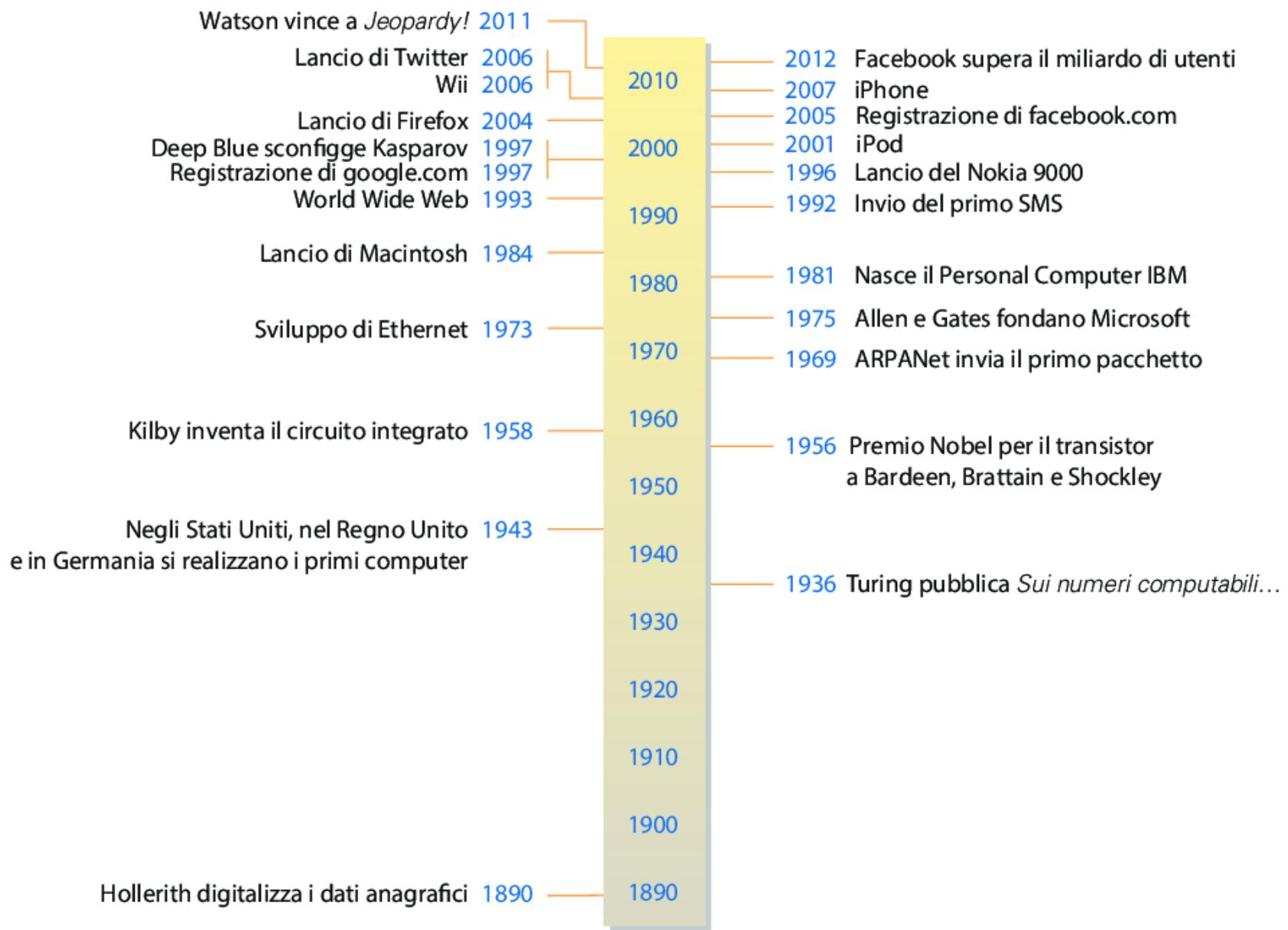
ARUBA PEC



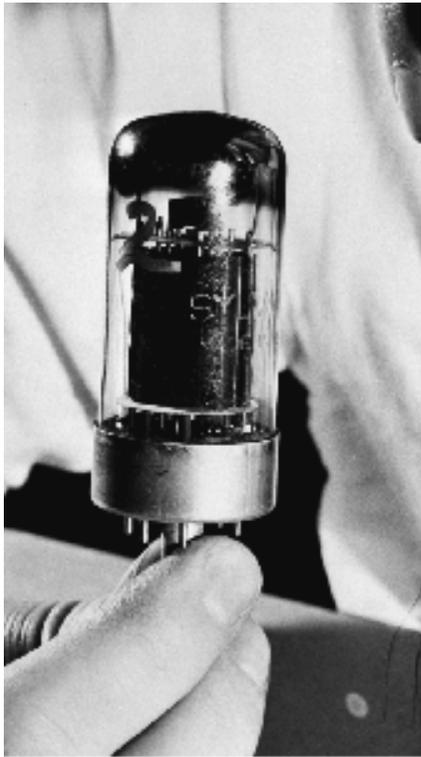
Marca
Temporale

Virus e antivirus e firewall

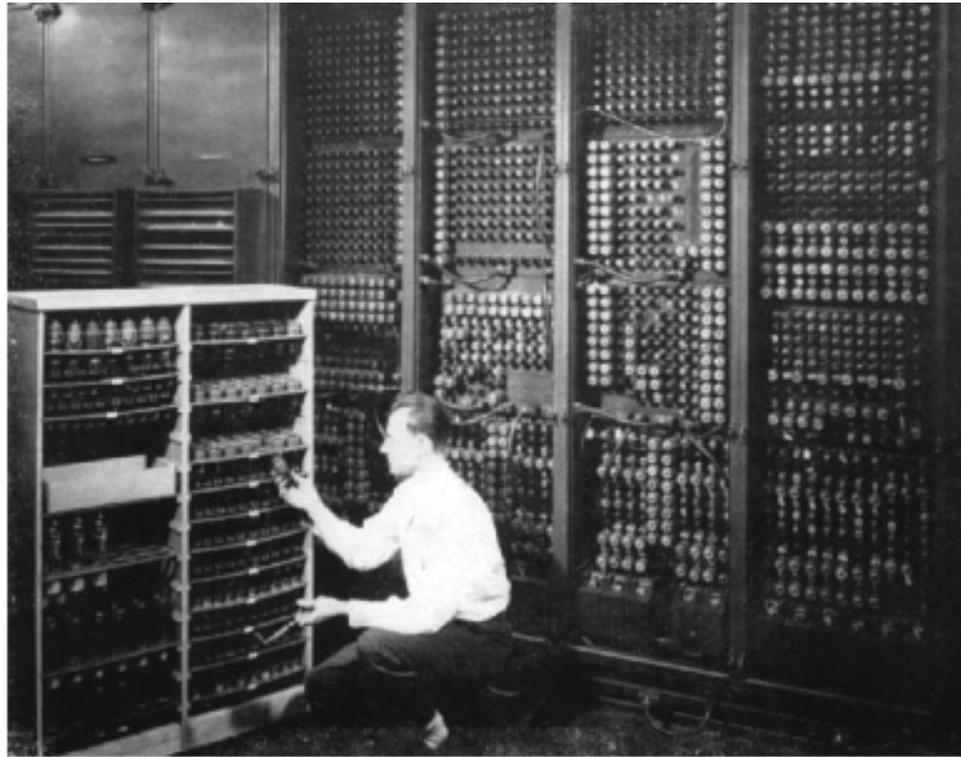




Primo Computer elettronico pesava 30 tonnellate
costruito da più di 17000 valvole



(a)



(b)

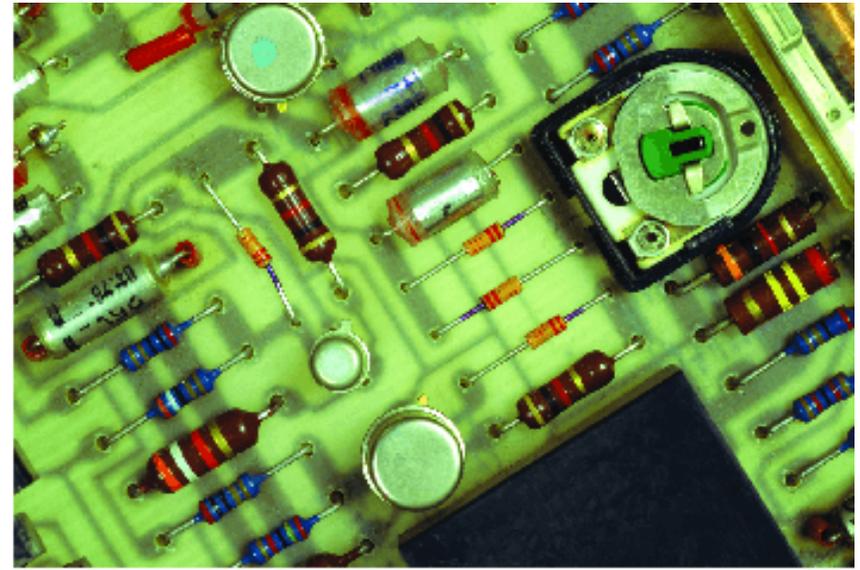
Passaggio ai transistor a semiconduttore



(a)

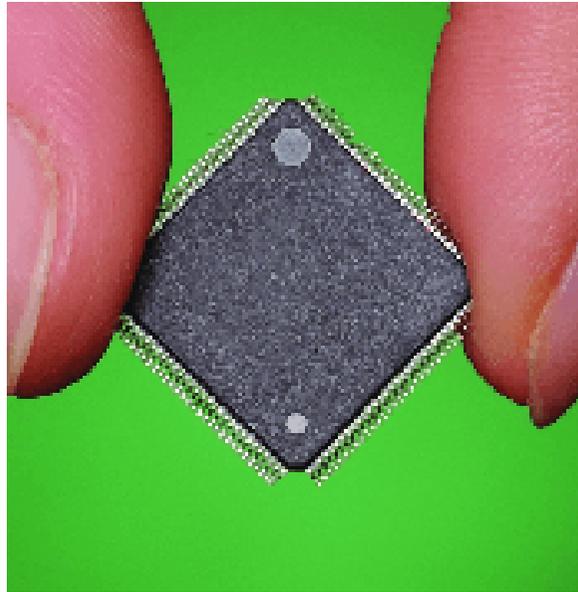


(b)



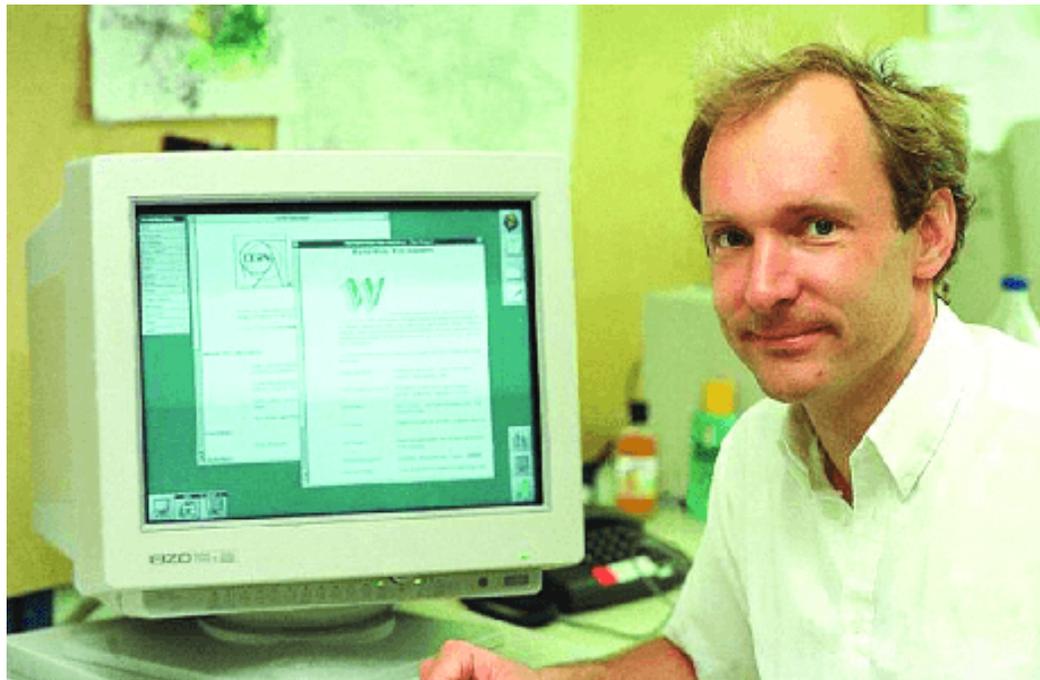
(c)

I circuiti integrati



La legge di Moore (ex presidente della Intel) si accorse che il numero di transistor all'interno dei circuiti integrati ogni anno raddoppiava (anche se oggi un po' meno velocemente)

Il protocollo HTTP e il World Wide Web



(a)

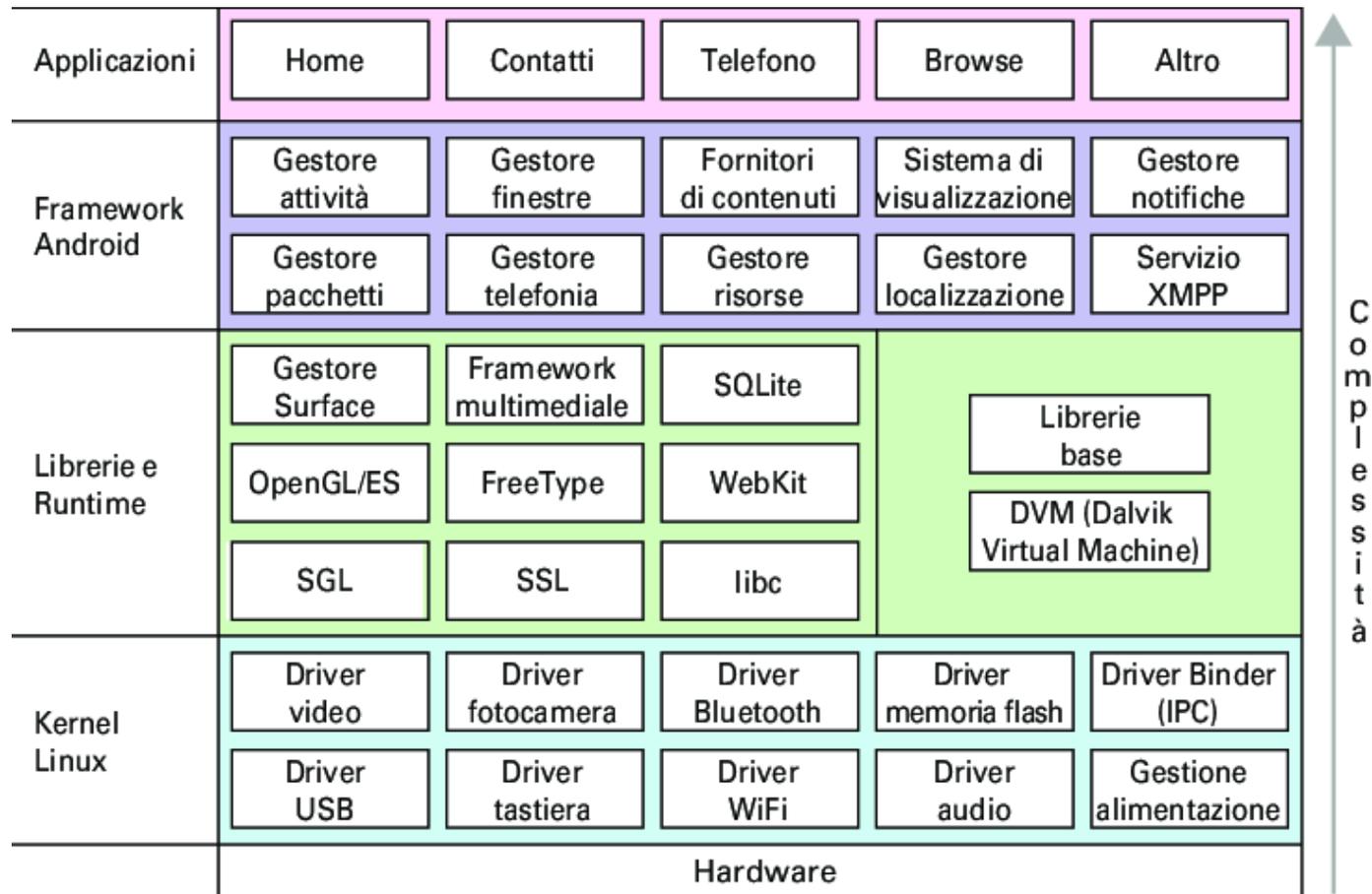


(b)

Tim Berners Lee 1994 (CERN) mostra una pagina web

Sviluppo del software a più livelli

"la pila software di uno smartphone"



Codifica e rappresentazione dell'informazione

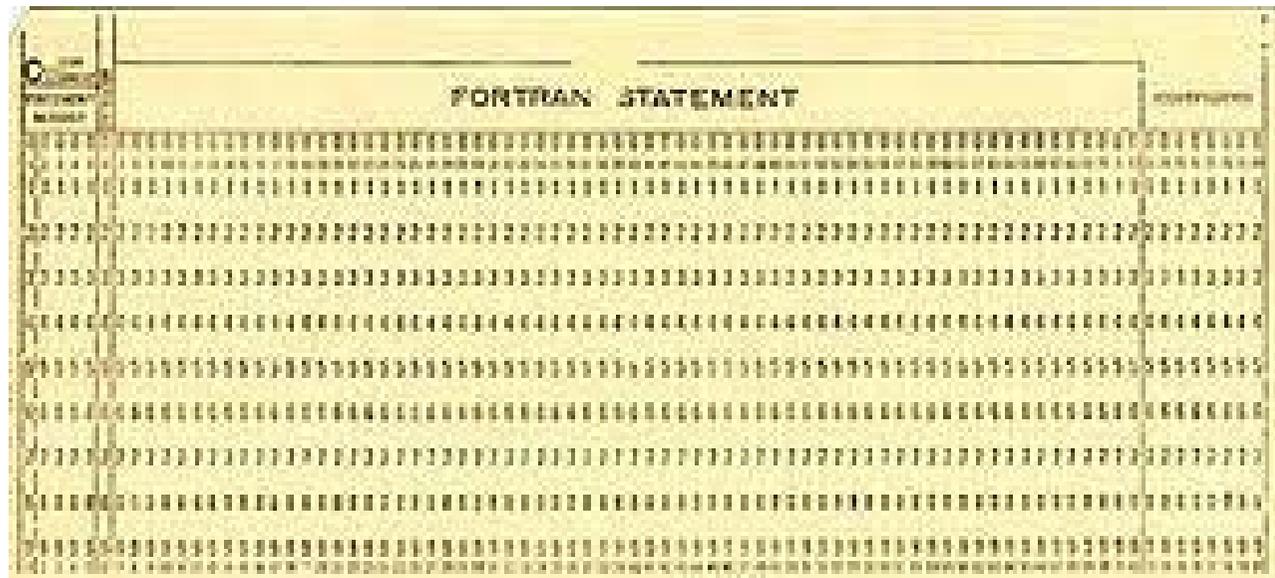
L'informazione è rappresentata dai suoi dati che a loro volta sono espressi sotto forma di simboli.

Un computer memorizza ed elabora informazioni.

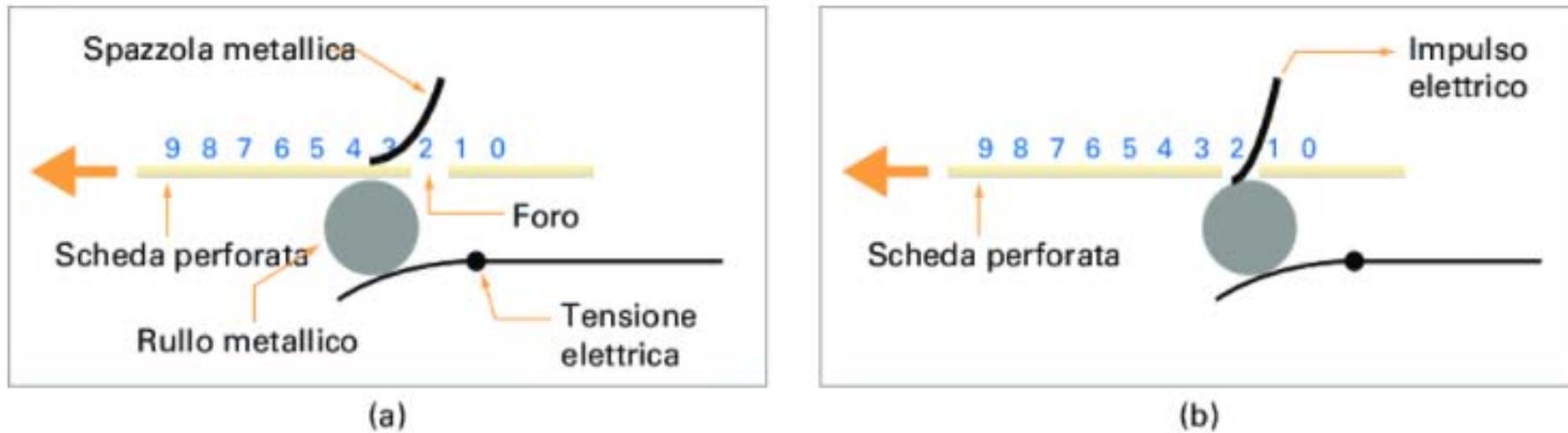
Le informazioni sul calcolatore devono essere rappresentate in una "forma" tale che siano "adatte" ad un certo tipo di memorizzazione e manipolazione

Bit e memorizzazione

- I computer rappresentano le informazioni attraverso bit (binary digit) che può assumere solo 2 valori 0 e 1.
- Una sequenza di bit può rappresentare numeri, caratteri, immagini ecc.
- Quindi necessariamente per memorizzare un bit occorre un dispositivo che può assumere solo uno dei 2 stati.



Bit e memorizzazione



Funzionamento di un lettore di schede perforate. (a) La scheda scorre verso sinistra, mossa da un rullo metallico. (b) Quando un foro passa sopra il rullo, la spazzola lo attraversa e chiude un contatto elettrico.

John von Neumann



John von Neumann (Budapest 1903 – Washington 1957)
matematico – fisico -informatico

È considerato uno dei più grandi matematici della storia moderna e una delle personalità scientifiche preminenti del XX secolo

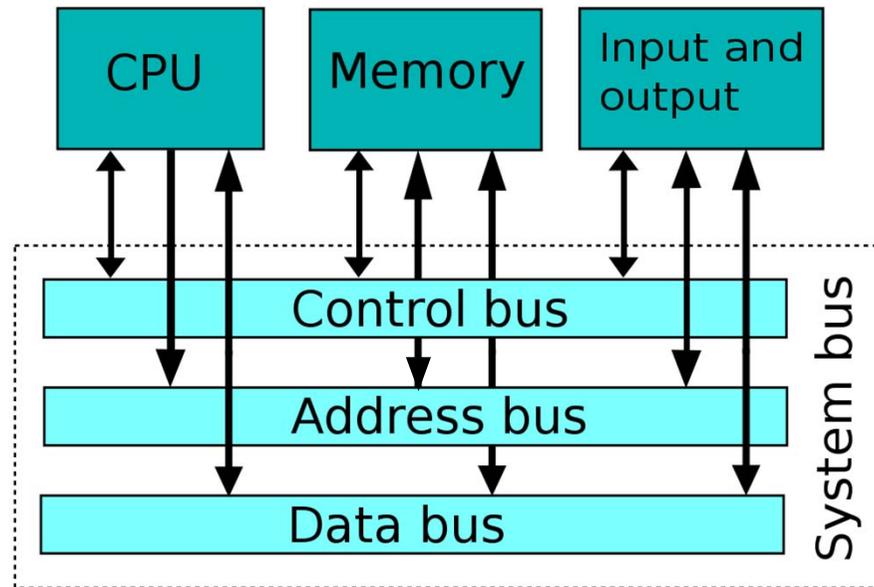
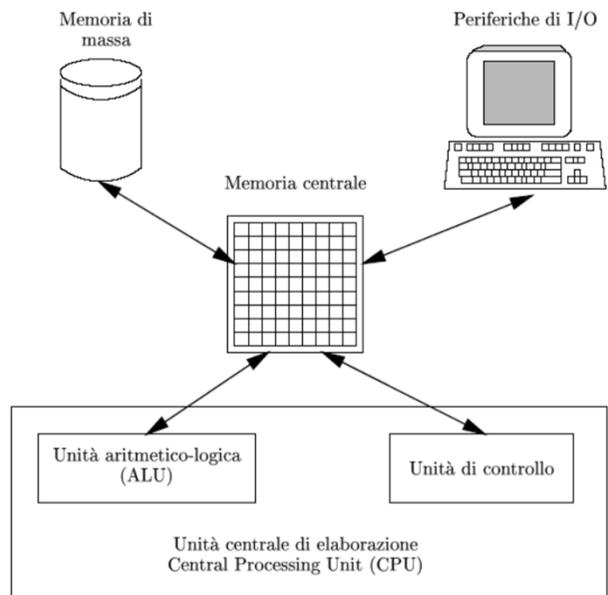
Architettura di Von Neumann

Il modello attuale dei calcolatori deriva dal modello definito da Von Neumann negli anni '40. Il modello o architettura di von Neumann è stata una delle prime descrizioni strutturali del calcolatore.

Lo schema si basa su 4 componenti fondamentali:

1. Unità centrale di elaborazione (CPU), che si divide a sua volta in unità aritmetica e logica (ALU o unità di calcolo) e unità di controllo;
2. Unità di memoria, intesa come memoria di lavoro o memoria principale (RAM, Random Access Memory);
3. Unità di input e output, con la prima vengono inseriti i dati nel calcolatore e con la seconda restituisce i dati elaborati all'operatore;
4. Bus, un canale che collega tutti i componenti fra loro.

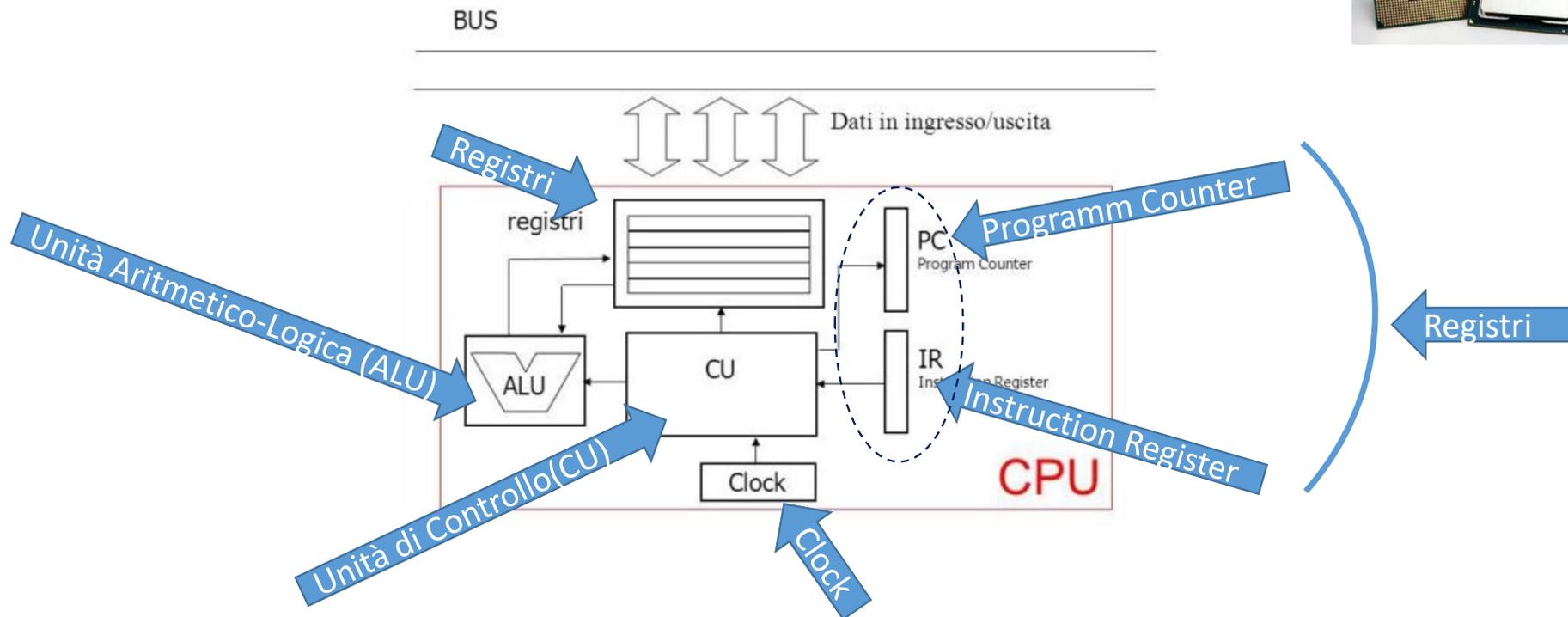
Architettura di Von Neumann



1. CPU Unità centrale di elaborazione (con unità ALU e unità di controllo)
2. Unità di memoria RAM
3. Unità di Input e Output
4. BUS

CPU (Central Processing Unit)

Il **processore o CPU (Central Processing Unit)** esegue le istruzioni, coordina il flusso e lo spostamento dei dati all'interno del computer e esegue le **operazioni logico-aritmetiche**.



Registri della CPU (Central Processing Unit)

La **ALU (Arithmetic and Logic Unit)** è l'unità di elaborazione in grado di compiere le operazioni di calcolo richieste ed eseguire le istruzioni del programma (confronto, trasformazione delle informazioni e operazioni aritmetiche e logiche).

CU (Control Unit) è l'unità di controllo collegata a tutte le parti della CPU e coordina l'esecuzione delle operazioni.

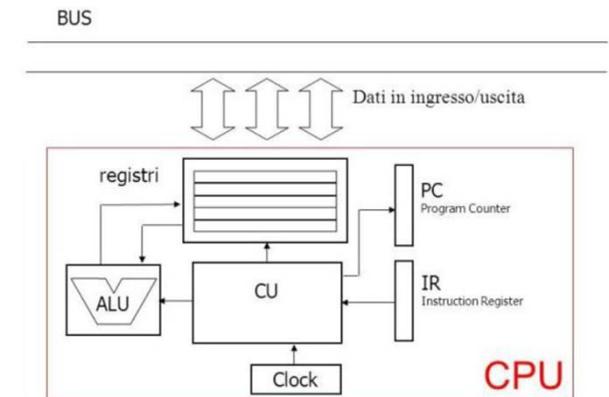
- Cura l'acquisizione e l'interpretazione delle istruzioni che si trovano in memoria.
- Reperisce i dati dall'input e coordina il lavoro della unità aritmetico

I **registri** sono delle piccole celle di memoria di pochi byte interne alla CPU utilizzate per memorizzare temporaneamente istruzioni e risultati parziali delle operazioni, nonché informazioni di controllo.

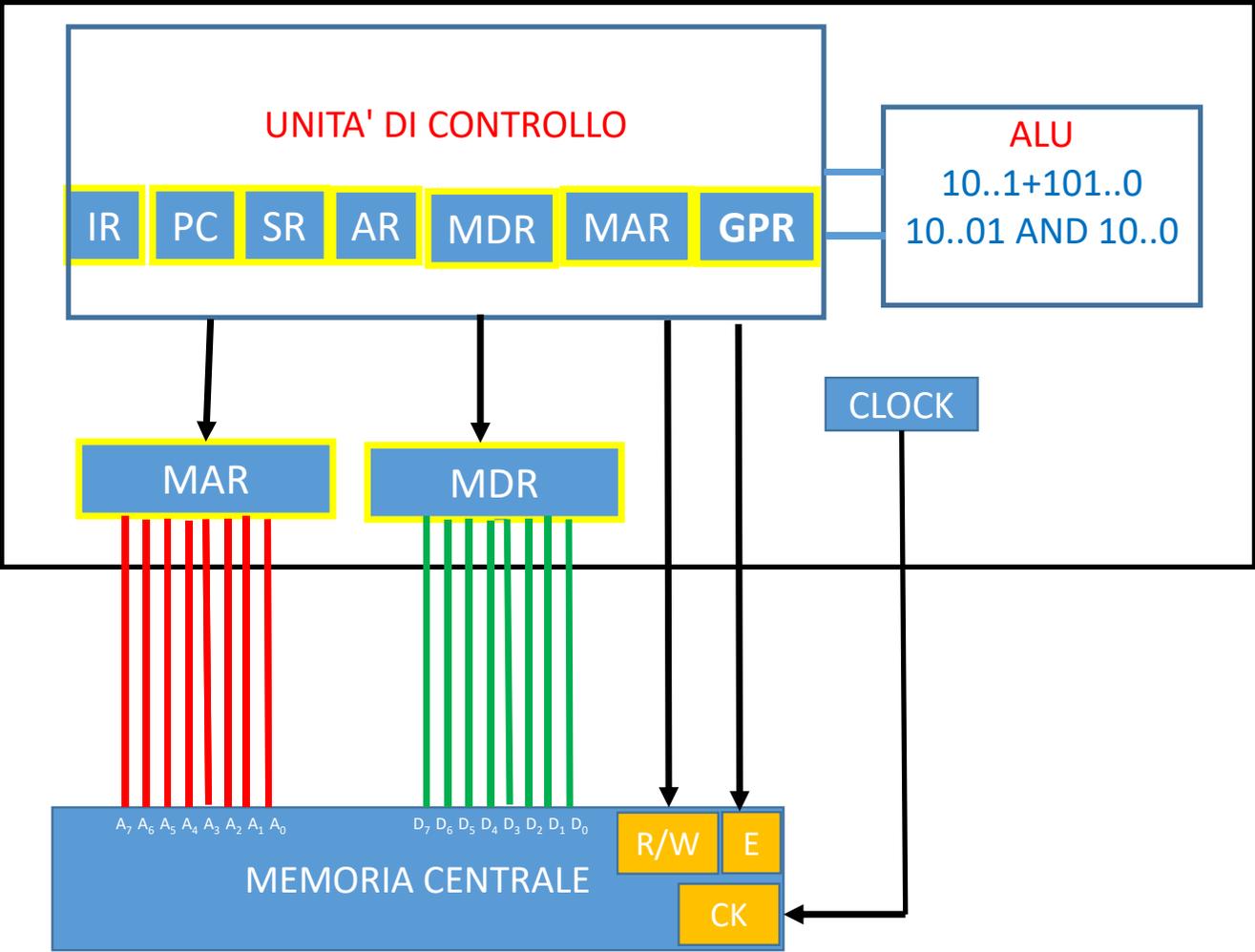
Il **clock** è l'orologio interno alla CPU che scandisce l'esecuzione delle operazioni.

- Ad ogni segnale di clock, viene inviato un segnale alla CU.

La frequenza operativa della CPU, ossia la velocità di lavoro, è determinata dalla frequenza del clock.



Registri della CPU

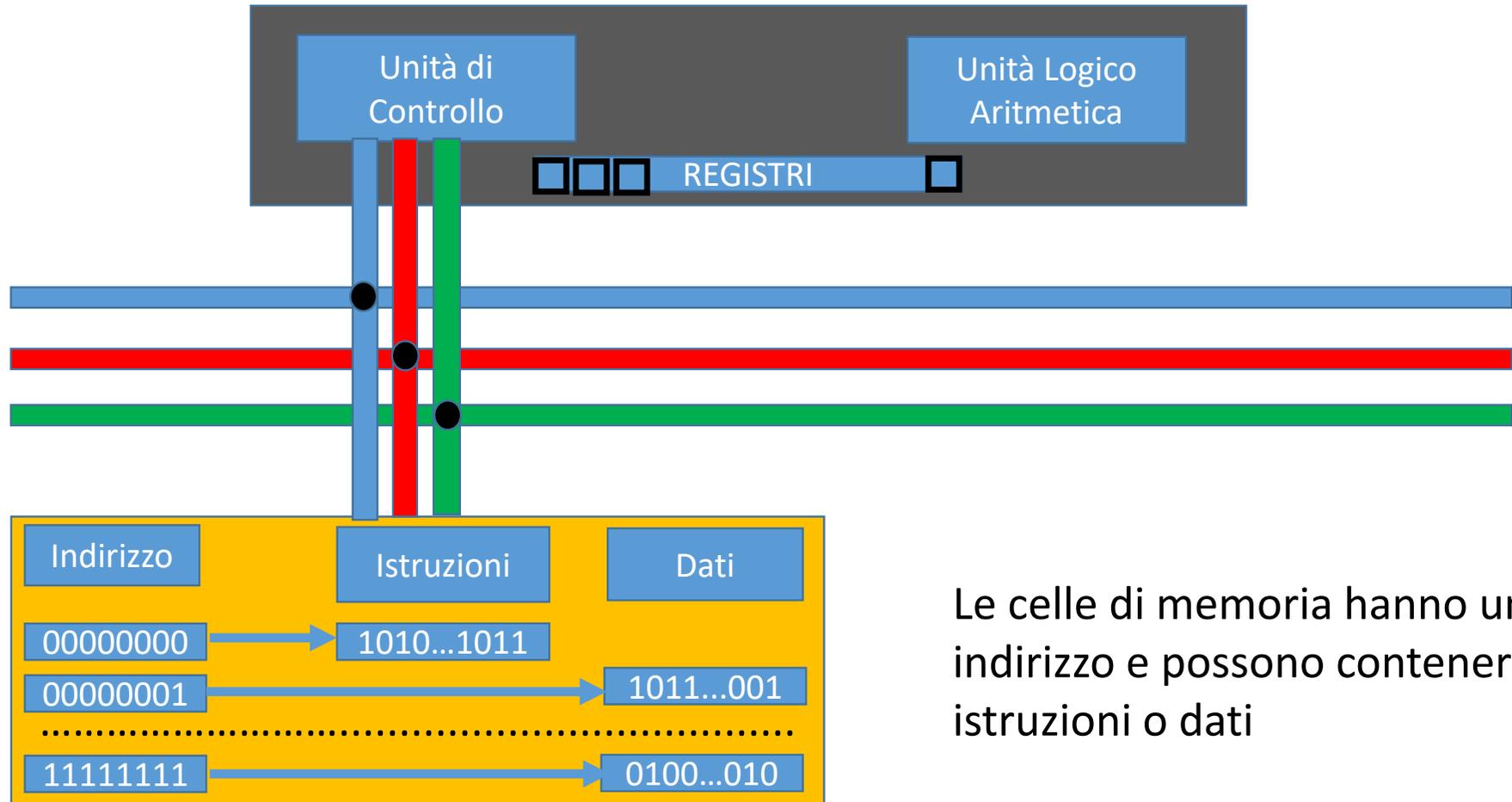


- PC = Program Counter
- IR = Instruction Register
- SR = Status Register
- AR = Accumulator Register
- MDR = Memory Data Register
- MAR = Memory Address Register
- GPR = General Purpose Register

Indirizzamento della memoria

- Il Memory Address Register (MAR) contiene l'indirizzo della cella di memoria che deve essere selezionata.
- La lunghezza in bit (n) del MAR determina la quantità massima di celle di memoria indirizzabili. Queste vanno da "0" a 2^n-1 .
Es per 16 bit $\rightarrow 2^{16}-1 = 65535$ ovvero sono indirizzabili 65536 celle.

MEMORIA

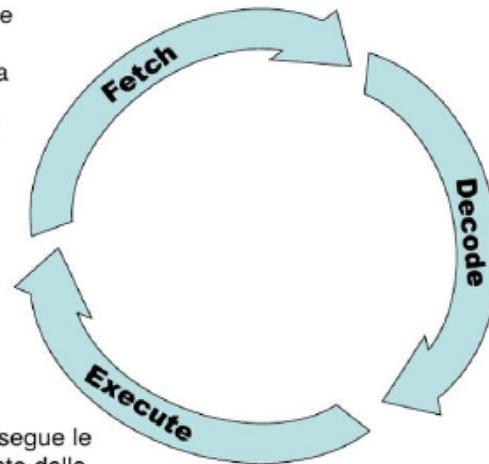


Le celle di memoria hanno un indirizzo e possono contenere istruzioni o dati

Unità di Controllo

- Questa unità controlla e determina il funzionamento del computer ed esegue ciclicamente tre FASI in maniera consecutiva:
- Fase di FETCH (prelievo)
- Fase di DECODE (decodifica)
- Fase di EXECUTE (esecuzione)

Fetch = Prende la prossima istruzione dalla memoria e incrementa il Contatore di Programma



Decode = Decodifica i bit nel registro delle istruzioni

Execute = Esegue le azioni richieste dalla istruzione nel registro delle istruzioni

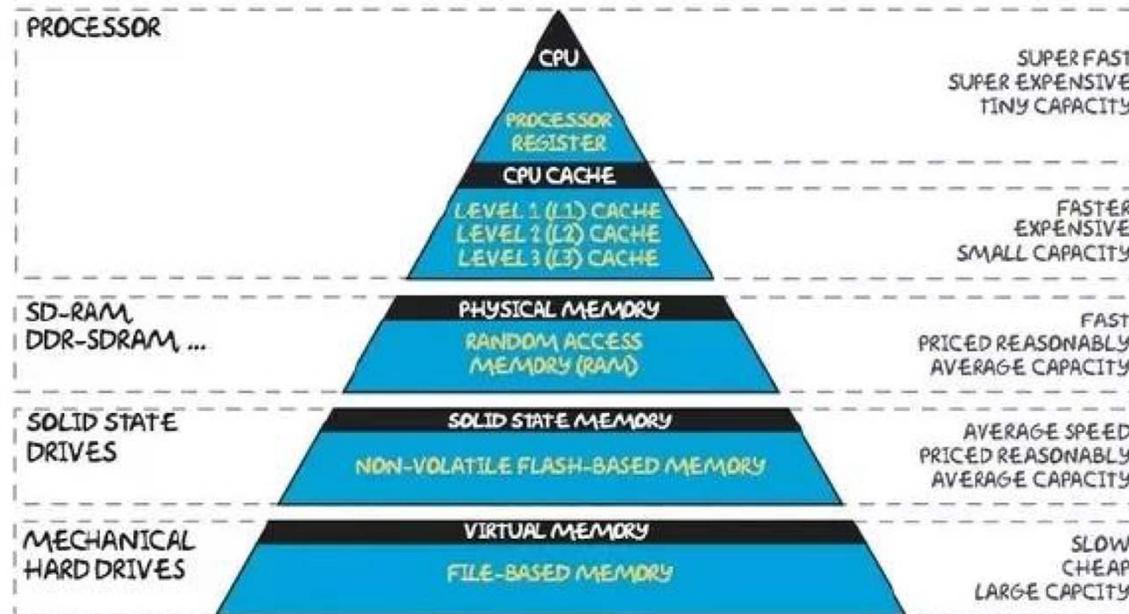
CICLO di Fetch-Decode-Execute

- **Fetch** (prelievo)
 1. accedi alla prossima istruzione (vai all'indirizzo della memoria in cui è contenuta la nuova istruzione)
 2. Memorizza l'indirizzo nel registro IR (registro delle istruzioni)
- **Decode** (decodifica)
 1. Decodifica l'istruzione in base al suo codice operativo (che può essere una operazione di somma ... ecc)
- **Execute** (esecuzione)
 1. Si individuano i dati usati nell'istruzione
 2. Si trasferiscono i dati nei registri opportuni
 3. Si esegue l'istruzione

Parametri di classificazione della memoria

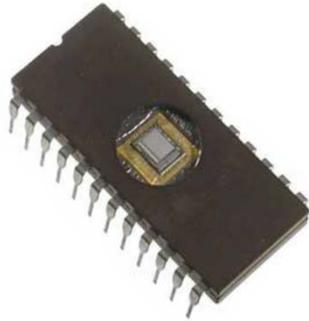
- **Volatilità:** le memorie volatili mantengono l'informazione solo finché vengono alimentate, in caso di spegnimento i dati sono persi. Le memorie non volatili sono quelle memorie dove è possibile ritrovare i dati memorizzati anche dopo aver spento il calcolatore.
- **Capacità:** numero di bit/byte memorizzabili (espressa in multipli di byte).
- **Velocità** di trasferimento dati: quantità di dati trasferiti dalla memoria in un'unità di tempo.
- **Tecnologia:** memorie elettroniche (RAM, ROM, FLASH, SSD ecc.), memorie magnetiche, memorie ottiche...

THE MEMORY HIERARCHY



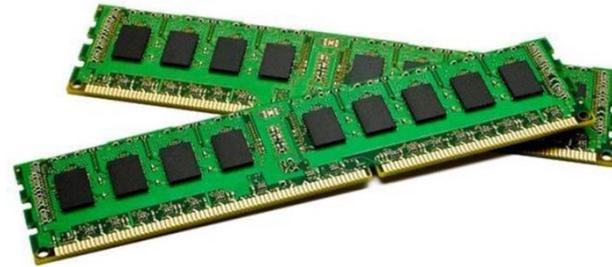
- 1 **Registri (interni alla CPU):** memorie volatili ad altissima velocità, di piccole dimensioni e locali al processore.
- 2 **Memoria Cache:** memoria piccola e veloce usata come interfaccia tra CPU e memoria centrale.
- 3 **Memoria RAM:** (Random Access Memory) mantiene dati e istruzioni dei programmi in esecuzione (volatile)
- 4 **Memorie di Massa:** memorie non volatili contenenti dati “salvati” dall’utente, rispetto alla memoria centrale hanno un tempo di accesso più elevato ma una capacità maggiore e un costo più ridotto.

ROM

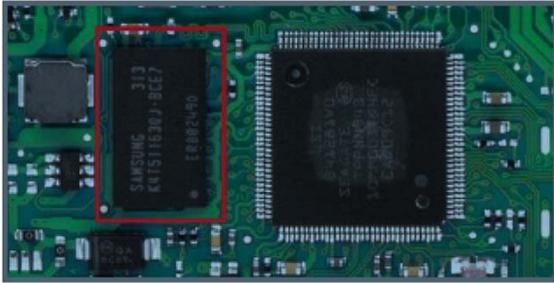


- ROM (Read Only Memory): è una memoria persistente e non volatile che possiamo solo leggere.
- Le memorie ROM vengono utilizzate per memorizzare i firmware, cioè quei software integrati direttamente all'interno di un componente elettronico programmato.
- Nella ROM della scheda madre è memorizzato il BIOS (Basic Input Output System) ovvero il software che permette di gestire l'hardware della scheda madre e interfacciarsi con il sistema operativo.

RAM



- La **RAM**(Random Access Memory) è una memoria volatile ad accesso casuale(non è a lettura/scrittura sequenziale)
- In questa memoria vengono caricati i dati e le istruzioni di un programma sotto forma di processi.
- È una memoria volatile



La cache

La memoria cache è la memoria più veloce nel calcolatore e molto costosa. Le istruzioni e i dati dalla RAM vengono caricati anche sulla memoria cache.

La memorizzazione avviene secondo due principi:

- **Località spaziale:** la lettura di una cella indica che probabilmente quelle vicine saranno presto necessarie.
- **Località temporale:** quando una cella è stata letta o modificata, è probabile che venga di nuovo utilizzata eliminando il dato precedente.

Memorie di Massa

Le **memorie di massa** sono tutti i dispositivi **magnetici**(dischi), **ottici**(CD, DVD, BD), ed **elettronici**(SSD, chiavette USB, memory card ..) che permettono di memorizzare file in modo permanente, per poterli riutilizzare e conservare finché l'utente non decide di eliminarli.



Collegamento con le periferiche

Collegamento delle Periferiche

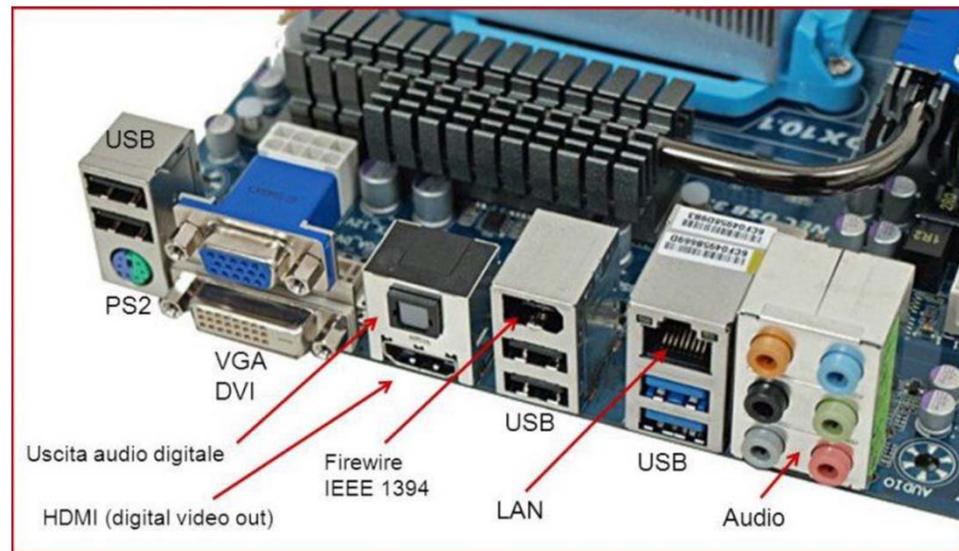
Il collegamento delle periferiche avviene attraverso **porte**(o interfacce):

Le porte è dove si collegano i dispositivi, si trovano nel retro del case del computer

Le varie porte si differenziano per: quantità e velocità di trasferimento delle informazioni.

Le periferiche possono essere:

- Periferiche esterne: Es.: tastiera, mouse, stampate, webcam, ecc.
- Periferiche interne: schede audio, video ecc.



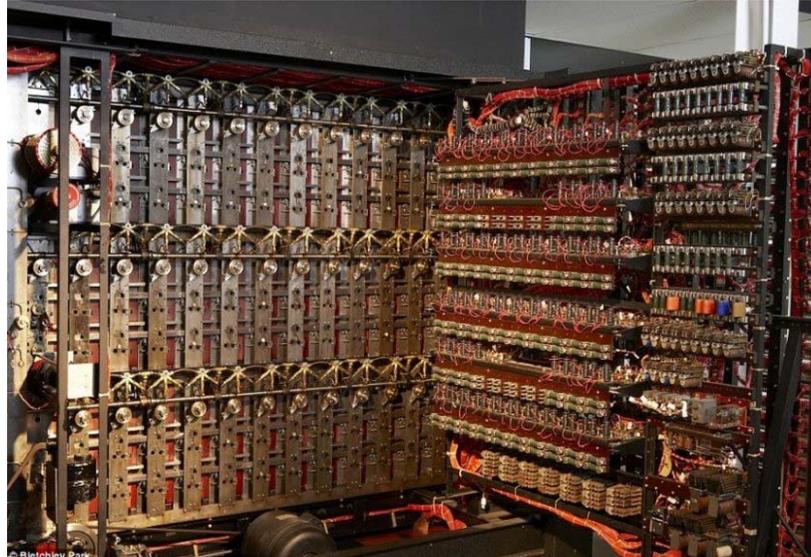
Alan Mathison Turing



(Londra nel 1912 Regno Unito 1954)

Alan Mathison Turing è stato un matematico, logico, crittografo e filosofo britannico, considerato uno dei padri dell'informatica e uno dei più grandi matematici del XX secolo

Macchina di Turing



Modello di agente di calcolo adatto a simulare la logica di qualsiasi algoritmo computazionale. Un sistema astratto che, opportunamente programmato, era capace di eseguire ogni tipo di operazione

Macchina di Turing



la macchina di Turing è formata da un'unità di controllo contenente un programma con un numero finito di istruzioni, da un nastro di lunghezza illimitata suddiviso in celle e da un'unità di lettura e scrittura sul nastro in grado di spostarsi avanti e indietro di un numero qualsiasi di celle e di leggere e scrivere in una qualsiasi delle celle un simbolo di un alfabeto prefissato.

La macchina di Turing

Il modello definito da Turing permette con un'unica macchina eseguire task differenti attraverso l'esecuzione di algoritmi.

Questa macchina è dotata di un nastro potenzialmente infinito su cui può leggere e/o scrivere dei simboli

è un potente strumento teorico che viene largamente usato nella teoria della calcolabilità e nello studio della complessità degli algoritmi: permette di comprendere i limiti del calcolo e proprio per questo ancora oggi è utilizzata per definire in modo formalmente preciso la nozione di algoritmo viene ricondotto alle elaborazioni effettuabili con macchine di Turing.

I problemi che non sono risolvibili con un procedimento algoritmico si dicono **non computabili**.

Per poter affermare che un problema è non computabile, e quindi insolubile, bisogna “dimostrare” che non esiste una soluzione algoritmica.

La macchina di Turing

- La soluzione di Turing consiste nel simulare il processo di calcolo umano, scomponendolo nei suoi passi.

La macchina è formata da una testina di lettura e scrittura con cui è in grado di leggere e scrivere su un nastro potenzialmente infinito suddiviso in celle.

- Ad ogni istante di tempo t_1 , la macchina si trova in uno stato ben determinato che è il risultato dell'elaborazione compiuta sui dati letti.
- Lo stato interno è la condizione in cui si trovano le componenti della macchina ad un determinato istante di tempo t . Le componenti da in gioco sono:
 - il numero della cella osservata
 - il suo contenuto e l'istruzione da eseguire

Algoritmo

- L'algoritmo è un metodo preciso e sistematico per produrre un determinato risultato.
- Un processo può essere definito algoritmo se ha 5 proprietà:
 1. Input specificato → quali sono i dati forniti all'elaborazione
 2. Output specificato → quali i dati che questo produce
 3. Determinatezza → tutto l'intero processo non deve essere non ambiguo
 4. Efficacia → l'agente deve essere in grado di eseguire le istruzioni fornite
 5. Terminazione → l'agente prima o poi dovrà alla fine fornire una risposta corretta

Esempio di algoritmo di ricerca lineare

Problema: vogliamo cercare un nome di uno studente in una classe.

Dati di INPUT: un elenco degli studenti della classe

Vogliamo che il risultato di OUPUT sia:

A) "Si, ho trovato il nome oppure" B)"No, non ho trovato il nome"

1. Colloca un contrassegno sul primo elemento
 2. Se l'elemento su cui si trova il contrassegno non è quello cercato invia il messaggio B
 3. Se l'elemento su cui si trova il contrassegno è quello cercato invia il messaggio A
 4. Avanza il contrassegno alla posizione successiva
 5. Torna al passo 2
- Algoritmi e processi euristici