



# LABORATORIO DI INFORMATICA

Lez. 1 – Introduzione all'informatica



Sito web: [www.docenti.unina.it/mettivier](http://www.docenti.unina.it/mettivier)

Ci si può registrare come studenti inserendo matricola/pin.

Sarà possibile scaricare tutto il materiale del corso



## Introduzione all'informatica

- Definizione di Informatica
- Sistemi di Elaborazione
- Applicazioni dell'Informatica

## L'elaboratore

- Tipi di elaboratore
- Rappresentazione dell'informazione
- Codifica dell'informazione

## HW dell'elaboratore

- L'unità centrale di elaborazione
- Le memorie secondarie
- Le periferiche di I/O

## SW dell'elaboratore

- La programmazione
- Il Sistema Operativo (SO)
- I programmi applicativi



## INFORMATICA

- Acronimo di INFORMAzione automaTICA
- Insieme dei processi e delle tecnologie che rendono possibile la creazione, la raccolta, l'elaborazione, l'immagazzinamento e la trasmissione dell'informazione con metodi automatici.
- Scienza preesistente all'elaboratore

L'informatica è lo studio degli algoritmi:

- delle loro proprietà formali e matematiche
- delle loro realizzazioni hardware
- delle loro realizzazioni linguistiche
- delle loro applicazioni



Dati: simboli con cui si rappresenta la realtà

Informazione: prodotta attraverso l'elaborazione dei dati

L'elaborazione trasforma dati di INPUT in dati di OUTPUT





## Procedura di elaborazione (Algoritmo)

Sequenza finita di azioni che risolve in un tempo finito un problema

## Programma

Codifica della procedura di elaborazione che consente la risoluzione di un problema, in un linguaggio comprensibile dall'esecutore.

## Esecutore

Una macchina capace di eseguire i programmi (le azioni specificate dall'algoritmo codificate in un determinato linguaggio di programmazione)



## Che cos'è un algoritmo?

Un insieme ben ordinato e finito di operazioni non ambigue ed effettivamente calcolabili che, applicate ad un insieme di condizioni iniziali, produce un risultato e termina in una quantità di tempo finita.



La ricetta per cuocere 100g di pasta:

1. Metti 1 L di acqua in una pentola
2. Poni la pentola sul fornello
3. Accendi la fiamma del fornello
4. Finché l'acqua non bolle ripeti il passo 5.
5. Aspetta un minuto
6. Aggiungi 10 g di sale grosso
7. Leggi sulla confezione della pasta la cottura prevista
8. Versa la pasta nell'acqua bollente
9. Aspetta il tempo di cui al punto 7
10. Scola la pasta
11. Fine



## L'elaboratore

Mezzo strumentale (**macchina, hardware**) per la rappresentazione, la memorizzazione e l'elaborazione delle informazioni (**dati, software**).

Elaboratore elettronico digitale

Elaboratore

Elettronico = utilizza componenti elettronici per elaborare le informazioni.

Digitale = elabora informazioni mediante numerazione binaria



Il computer ha due macrocomponenti:

## 1. Hardware

La struttura fisica dell'elaboratore, costituita da componenti elettronici, elettromeccanici ed ottici

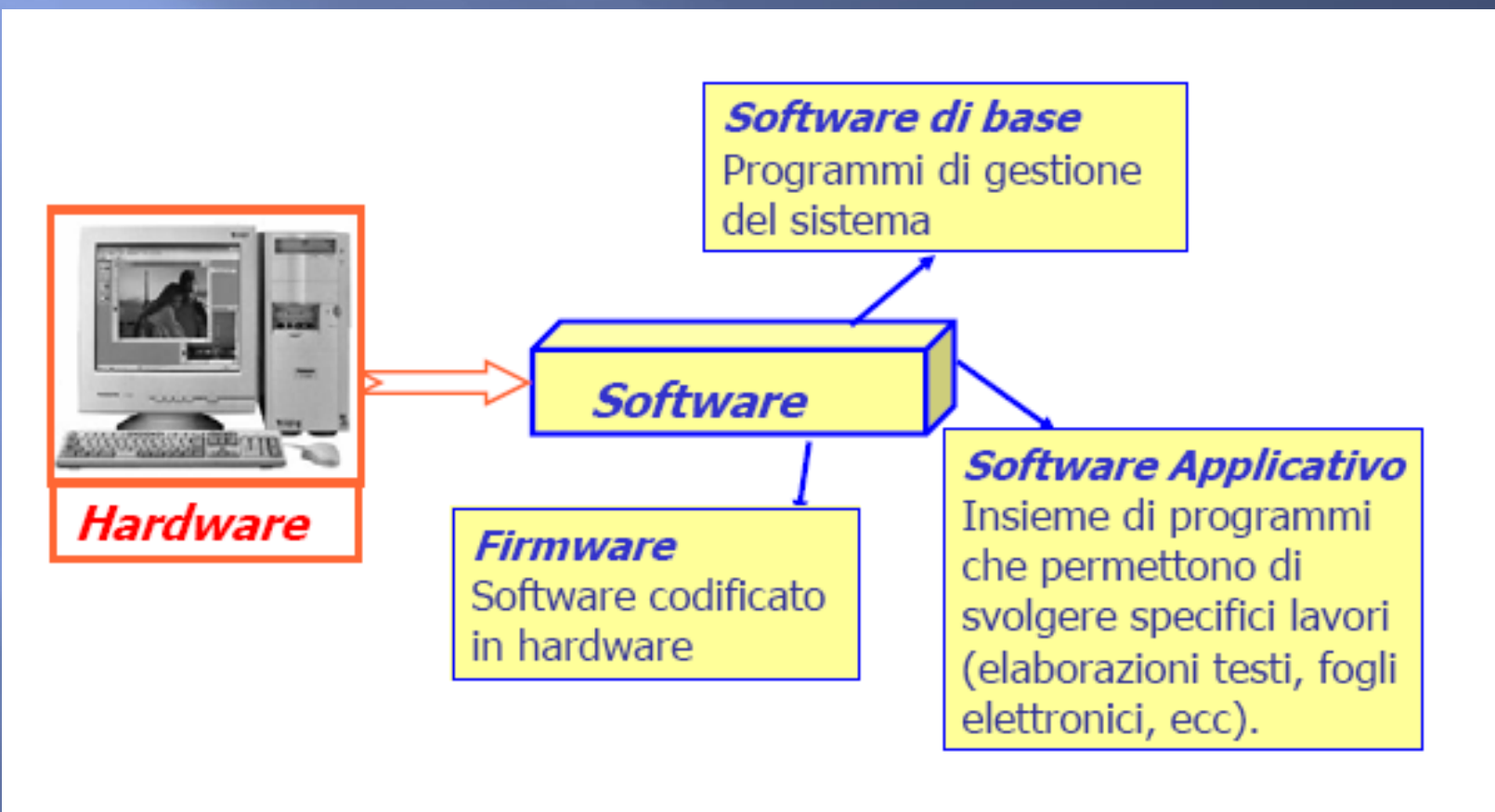
## 1. Software

L'insieme dei programmi che consentono all'hardware di svolgere compiti specifici.

La presenza di una componente hardware ed una software determina la capacità della macchina di eseguire elaborazioni di tipo differente senza modificare la struttura fisica.



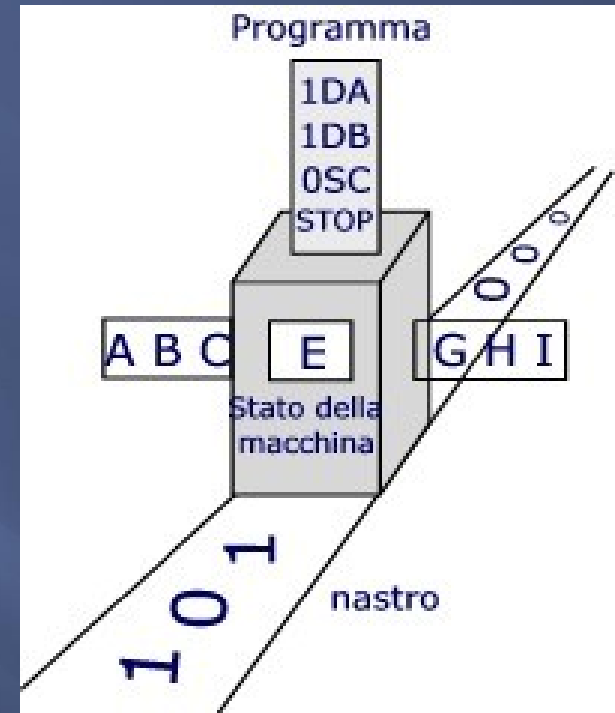
# Sistemi di Elaborazione





# La macchina Turing

Turing ideò una macchina teorica, in grado di simulare il comportamento di un uomo intento a eseguire un calcolo aritmetico. La macchina di Turing è costituita da un nastro di lunghezza potenzialmente infinita, suddiviso in sezioni o celle (simile, quindi, a una pellicola cinematografica), che la macchina può far scorrere avanti e indietro, e da una unità di lettura e scrittura in grado di leggere, scrivere o cancellare simboli (uno alla volta) nelle celle del nastro. Tali simboli sono elementi di un insieme finito di simboli distinti, l'alfabeto della macchina. Inoltre, la macchina è dotata di una memoria interna (la memoria esterna è il nastro) che può assumere un insieme finito di stati. Il comportamento della macchina, che funziona eseguendo una successione di passi discreti, è determinato da un insieme finito di regole che, dato il simbolo letto e lo stato interno della macchina, prescrivono il simbolo da scrivere, lo stato da assumere e il verso di spostamento del nastro nel passo successivo.





Lo schema della macchina di Von Neumann, modello sul quale si basano praticamente tutti i moderni computer è basato su quattro elementi fondamentali:

- La memoria
- L'unità aritmetico logica (ALU)
- L'unità di controllo
- Le unità di Ingresso/uscita (I/O)



Processore  
(CPU)



Nella macchina di von Neumann sia i dati reali sia le istruzioni primitive sono codificate mediante i due simboli della notazione numerica binaria.

Questa codifica viene detta **linguaggio macchina**.

Un aspetto interessante della notazione binaria è che le operazioni fondamentali sui numeri in cifre binarie equivalgono a operazioni logiche del calcolo proposizionale, come mostrò nella metà del'800 il matematico inglese George Boole. Una macchina che calcola funzioni sui numeri naturali in notazione binaria compie operazioni logiche, e viceversa.



Negli anno '50 Jack Kilby (1923-2005), inventa i **circuiti integrati**, facendo entrare la miniaturizzazione nella storia del computer.

Nel 1969 M.E. Hoff, ingegnere di una piccola azienda, la Intel, riuscì a ridurre le dimensioni delle imponenti CPU a quelle di un chip della grandezza di un'unghia. Nasce il **microprocessore**.

Ormai si trovano non solo all'interno degli elaboratori, dove possono essere più di uno, ma anche in posti assolutamente impensabili.

Oggi un qualsiasi PC, con un chip di silicio della grandezza di un fagiolo e dal costo di appena 100 dollari è più potente dell'ENIAC.

Se il settore automobilistico si fosse sviluppato quanto quello dell'informatica, oggi una Rolls Royce costerebbe 2,75 dollari e con circa un litro di benzina riuscirebbe a fare più di un milione di chilometri.





Gli elaboratori si differenziano in base alla **velocità di elaborazione**, alla **capacità di memoria**, alla **tipologia dei processori**, al **costo** e agli **impieghi tipici**.

Supercomputer, mainframe, minicomputer, workstation, personal computer

Miglioramento rapidissimo delle prestazioni degli elaboratori (1945-2005), nonostante il modello base dell'elaboratore sia pressoché invariato.

Tasso di crescita elevato determinato da due fattori:

- Miglioramento della architettura (organizzazione funzionale delle parti costituenti l'elaboratore);
- Miglioramento della tecnologia di realizzazione.



# Rappresentazione dell'informazione



## ANALOGICO

Ragiona utilizzando grandezze che variano con continuità (analogiche)

conversione D/A  
Digitale/Analogico



conversione A/D  
Analogico/Digitale



## DIGITALE

Elabora solamente informazioni espresse come sequenze di 1 e 0 (bit)



## Rappresentazione digitale

- Ogni quantità discreta è rappresentata da un simbolo
- L'insieme dei simboli costituisce un Alfabeto
- Le informazioni sono rappresentate, trasmesse ed elaborate usando **livelli discreti di una grandezza fisica.**
- I valori della grandezza fisica sono interpretati come valori discreti, anziché come quantità analogiche.
- Se ad ogni livello della grandezza fisica associamo un simbolo, si ottiene un alfabeto.

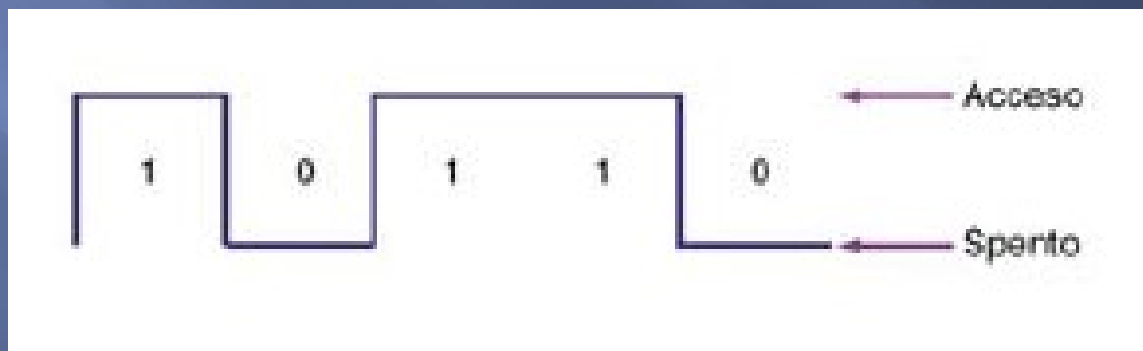


## Rappresentazione binaria digitale

La rappresentazione delle informazioni all'interno dell'elaboratore si basa su un alfabeto binario {0, 1}.

Ogni segnale elettrico trasporta uno dei due possibili valori, a seconda del livello di tensione.

- Le tensioni di **basso livello** sono interpretate come **0**.
- Le tensioni di **alto livello** sono interpretate come **1**.





## Perché la rappresentazione binaria?

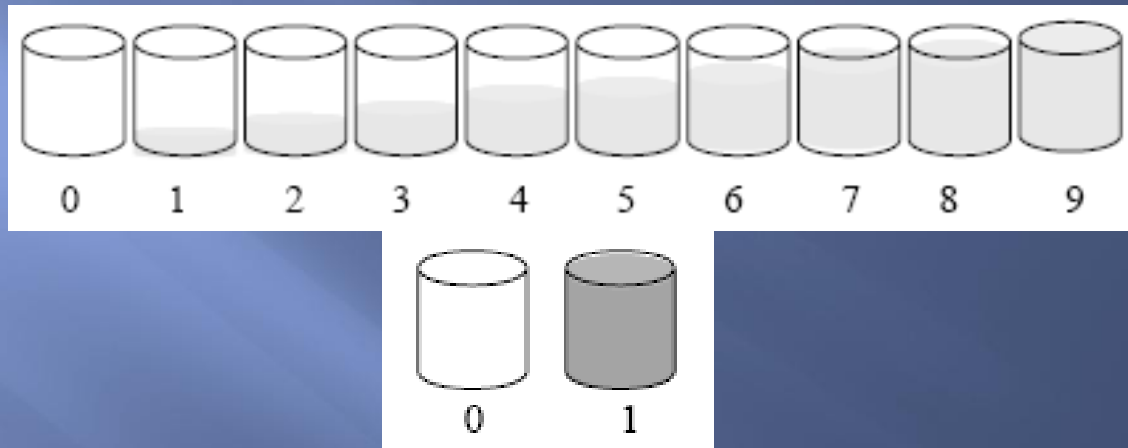
### Semplicità

Adottando una rappresentazione binaria, l'elaboratore può essere realizzato con componenti elementari semplici, che operano in **due soli stati** possibili (**transistor**).

- sostanze magnetiche con due opposte polarizzazioni
- passaggio o meno di corrente
- passaggio o meno di luce



## Perché la rappresentazione binaria?



### **Affidabilità** (probabilità di errore bassa)

Disturbi provenienti dall'ambiente o interferenze (rumore) indotte da altri componenti possono far variare lo stato di un componente.

Adottando due soli stati, la separazione tra le corrispondenti bande di valori è massima -> il rumore, sommato ad un qualsiasi valore, ha probabilità minima di spostare il valore nella banda successiva.



Qualsiasi sistema fisico è soggetto a degradazione nel tempo.

Usare il minimo numero di simboli garantisce che piccole degradazioni del sistema non corrispondano a differenze significative nella informazione rappresenta. Lo stesso principio vale anche nella comunicazione, che è sempre soggetta a disturbi e degradazione dell'informazione.



Quante informazioni posso rappresentare in forma binaria?

A patto di usare una sequenza (combinazione) di elementi di base sufficientemente lunga mi bastano 2 soli simboli per rappresentare qualunque informazione, dalla Divina Commedia alla Nona di Beethoven.

La differenza è notevole:

1 simbolo = zero informazioni

2 simboli = infinite informazioni



Es. Codice binario di Morse: . -

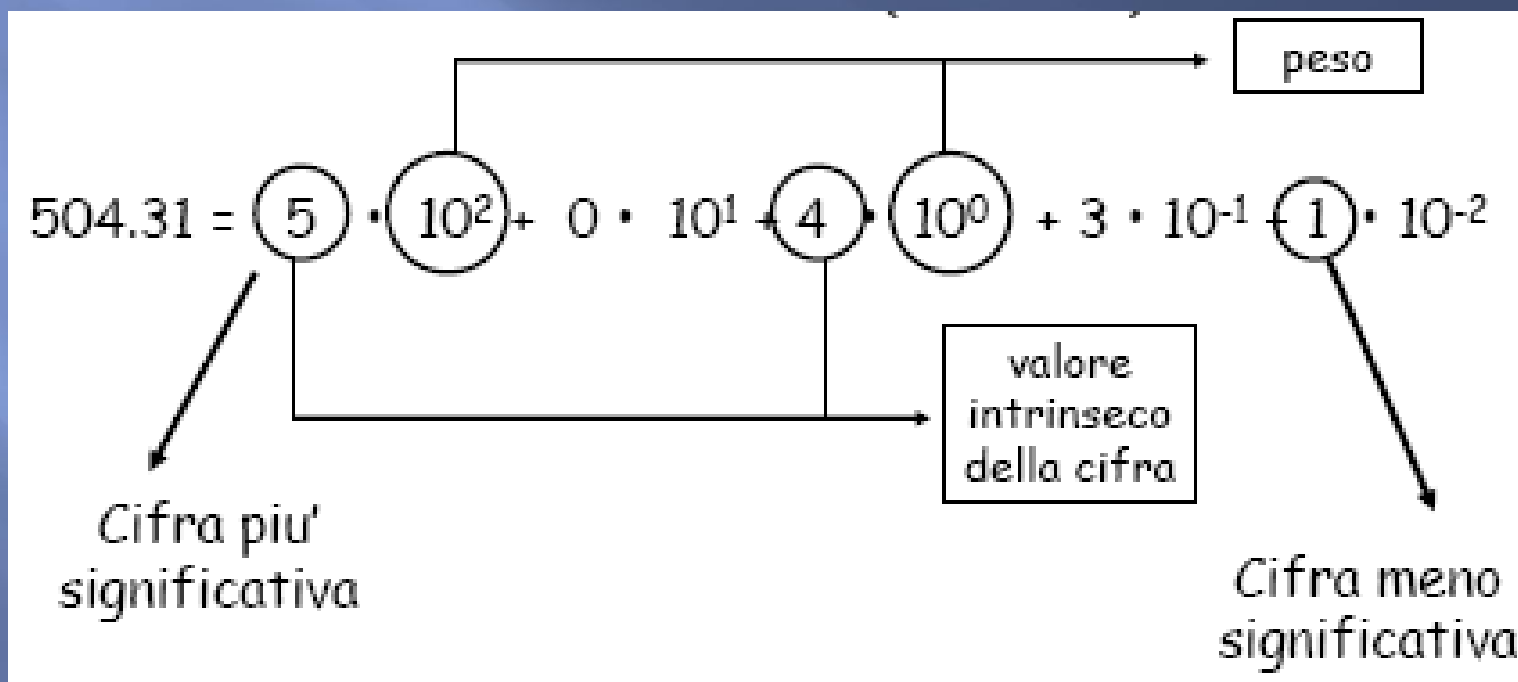
A	. -	J	. - - -	S	. . .
B	- . . .	K	- . -	T	-
C	- . . .	L	. - . .	U	. . -
D	- . .	M	- -	V	. . . -
E	.	N	- .	W	. - -
F	. . . .	O	- - -	X	- . . .
G	- . .	P	. - - .	Y	- . - -
H	. . . .	Q	- . . -	Z	- . . .
I	. .	R	. - .		





## Sistema di numerazione decimale

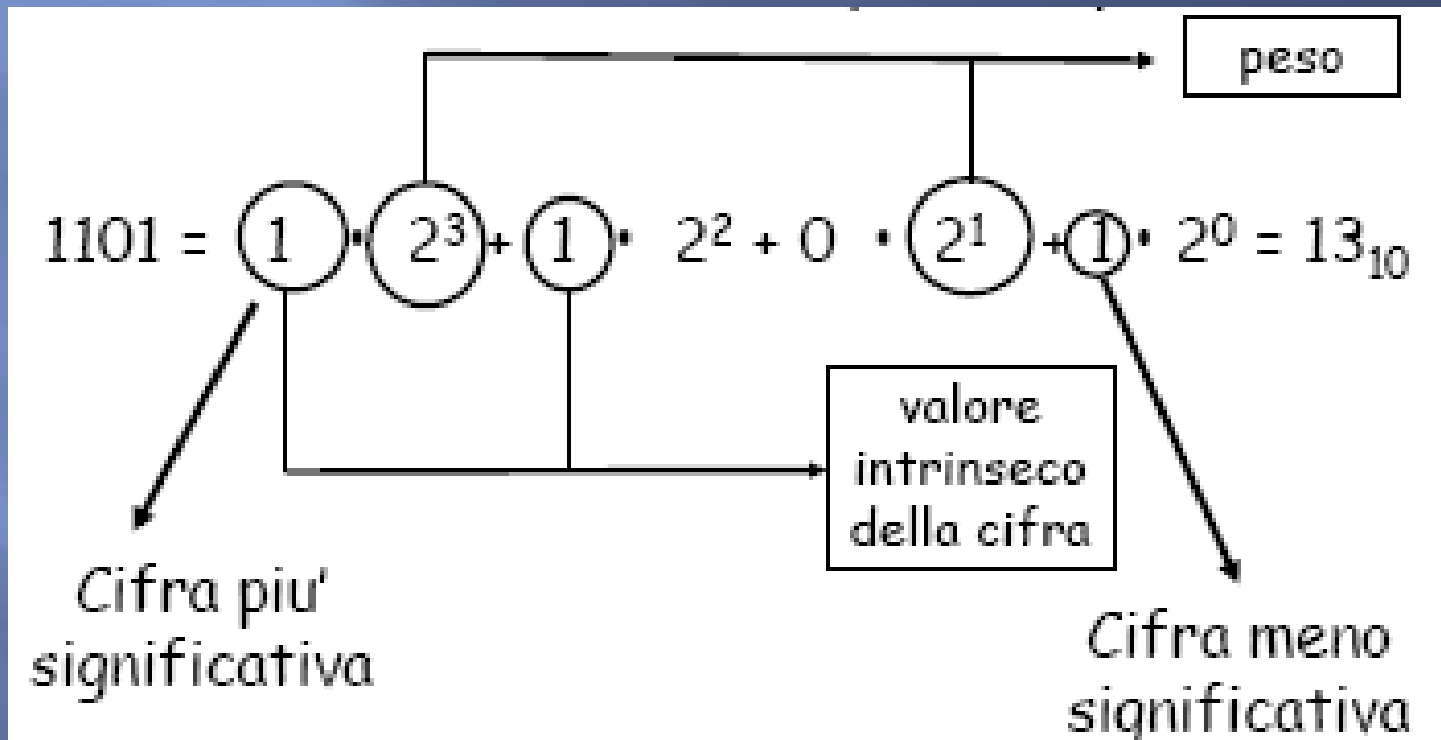
1. E' posizionale (unità, decine, centinaia)
2. E' costituito da 10 cifre (base 10)





## Sistema di numerazione binaria

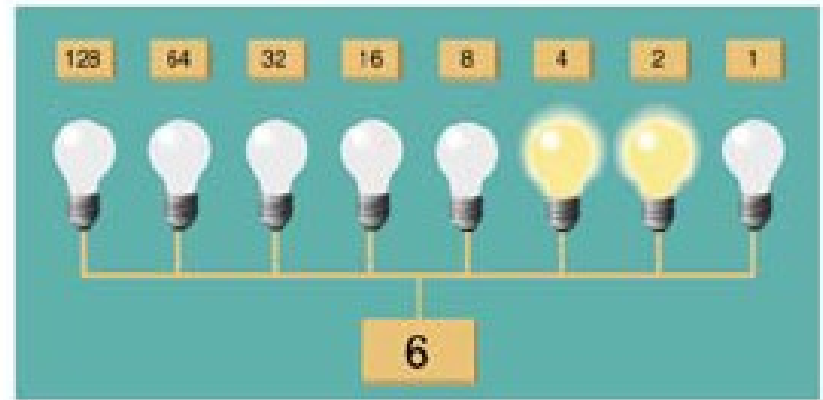
1. E' posizionale
2. E' costituito da 2 cifre (base 2)





# Rappresentazione dell'Informazione

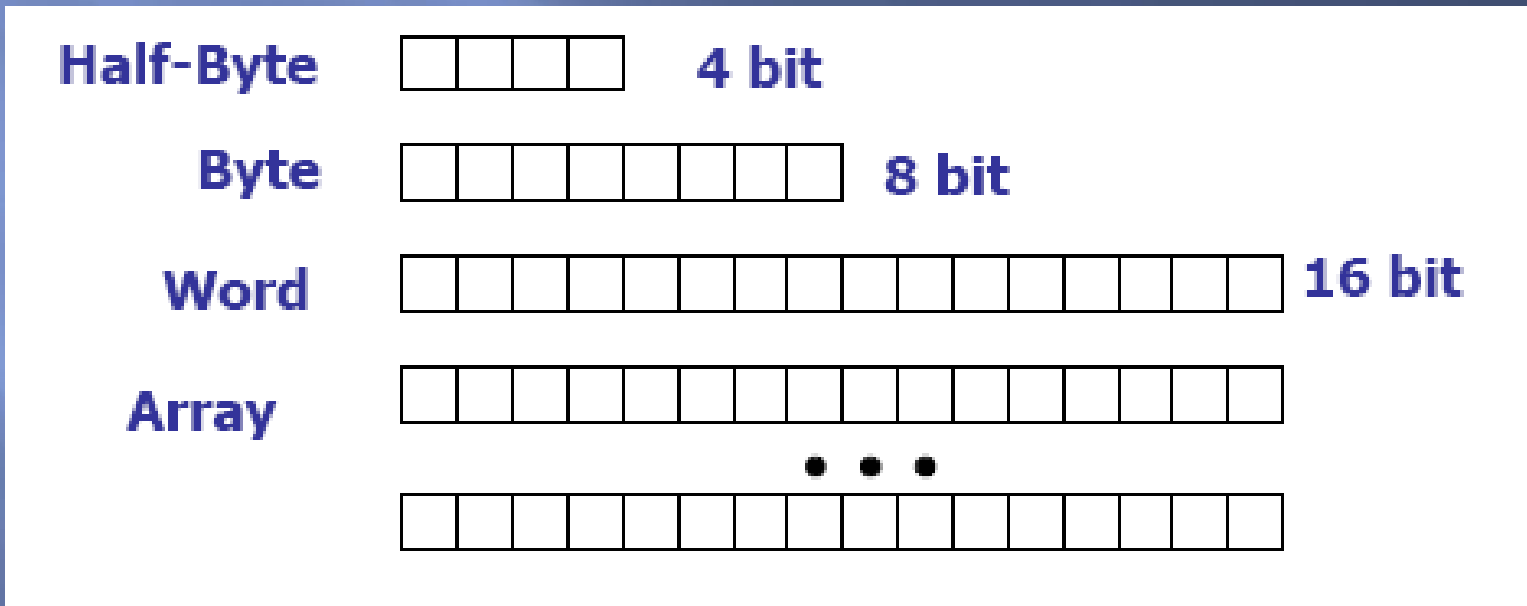
Numeri decimali	Numeri binari			
0				0
1				1
2		1		0
3		1		1
4	1	0		0
5	1	0		1
6	1	1		0
7		1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0





Ogni informazione si struttura in termini della unità di informazione elementare (bit)

Strutture logiche di informazione





## Multipli delle strutture logiche

Multiplo	Sigla	Valore	Approssimazione
Kilo	k	$2^{10} = 1024$	$\approx 10^3$
Mega	M	$2^{20} = 1024^2$	$\approx 10^6$
Giga	G	$2^{30} = 1024^3$	$\approx 10^9$
Tera	T	$2^{40} = 1024^4$	$\approx 10^{12}$
Peta	P	$2^{50} = 1024^5$	$\approx 10^{15}$
Exa	E	$2^{60} = 1024^6$	$\approx 10^{18}$



# Algebra Booleana

L'algebra che useremo è dovuta a Boole ed è detta ALGEBRA BOOLEANA. Le variabili di questa algebra sono binarie, possono assumere solo due valori (0,1).  
*Le variabili si indicano con le lettere A,B,C,X,Y,W,Z.*  
*Le operazioni base sono AND ( $\cdot$ ), OR ( $+$ ), NOT ( $\bar{\phantom{x}}$ )*  
Possiamo definire questi operatori tramite la *tabella di verità*



*La Tabella di Verità permette di definire gli operatori AND, OR NOT.*

# Algebra Booleana

AND			OR			NOT	
X	Y	$Z = X \cdot Y$	X	Y	$Z = X + Y$	X	$Z = \bar{X}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Table 2-1 Truth Tables for the Three Basic Logical Operations

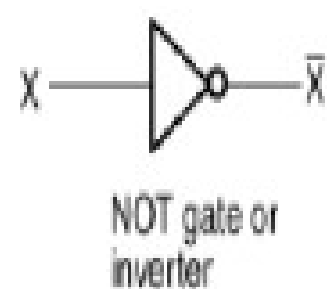
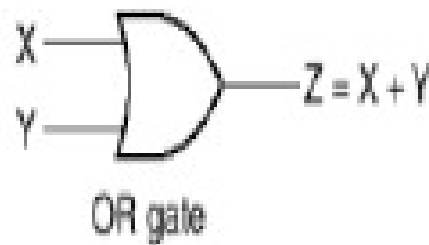
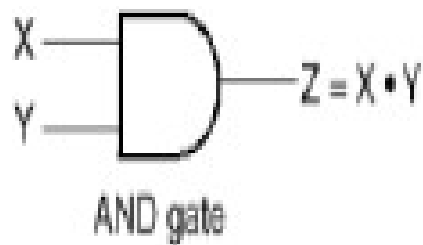


Sono stati individuati e costruiti circuiti elettronici che realizzano le operazioni elementari, questi sono detti *PORTE LOGICHE*.

Le *porte logiche* sono circuiti che operano su più segnali di ingresso e producono un segnale di uscita.

Rispondono a due valori di range di tensioni che associamo ai valori logico 0 e 1.

Le porte logiche che rappresentano le operazioni AND OR e NOT sono di seguito riportate.



(a) Graphic symbols



## Complemento a uno e complemento a due

Quando si fa riferimento a numeri in base due, il complemento alla base è più noto come «complemento a due» (che evidentemente è la stessa cosa). D'altro canto, il complemento a uno è ciò che è già stato descritto con l'esempio seguente, dove si ottiene a partire dal numero  $0011001100_2$ :

Si comprende intuitivamente che il complemento a uno si ottiene semplicemente invertendo le cifre binarie:

Pertanto, il complemento a due di un numero binario si ottiene facilmente invertendo le cifre del numero di partenza e aggiungendo una unità al risultato.

$$\begin{array}{c} 0011001100_2 \\ \downarrow \\ 1100110011_2 \end{array}$$



## Addizione binaria

L'addizione binaria avviene in modo analogo a quella del sistema decimale, con la differenza che si utilizzano soltanto due cifre numeriche: 0 e 1. Pertanto, si possono presentare solo i casi seguenti: Segue l'esempio di una somma tra due numeri in base due:

$$\begin{array}{r} 10011001_2 + (153_{10}) \\ 00110011_2 = (51_{10}) \\ \hline 11001100_2 (204_{10}) \end{array}$$



## Sottrazione binaria

La sottrazione binaria può essere eseguita nello stesso modo di quella che si utilizza nel sistema decimale. Come avviene nel sistema decimale, quando una cifra del minuendo (il numero di partenza) è minore della cifra corrispondente nel sottraendo (il numero da sottrarre), si prende a prestito una unità dalla cifra precedente (a sinistra), che così si somma al minuendo con il valore della base di numerazione. L'esempio seguente mostra una sottrazione con due numeri binari:

Generalmente, la sottrazione binaria viene eseguita sommando il complemento alla base del sottraendo. Il complemento alla base di  $00110011_2$  con otto cifre è  $11001101_2$ :

Pertanto, la sottrazione originale diventa una somma, dove si trascura il riporto:

$$\begin{array}{r} 10011001_2 - \\ 00110011_2 = \\ \hline 01100110_2 \end{array} \quad \begin{array}{l} (153_{10}) \\ (51_{10}) \\ (102_{10}) \end{array}$$



# Algebra Booleana

$$\begin{array}{r} 10011001_2 + (153_{10}) \\ 11001101_2 = \\ \hline 101100110_2 - \\ 10000000_2 = \\ \hline 01100110_2 (102_{10}) \end{array}$$



## Moltiplicazione binaria

La moltiplicazione binaria si esegue in modo analogo a quella per il sistema decimale, con il vantaggio che è sufficiente sommare il moltiplicando, facendolo scorrere verso sinistra, in base al valore del moltiplicatore. Naturalmente, lo spostamento di un valore binario verso sinistra di  $n$  posizioni, corrisponde a moltiplicarlo per  $2^n$ . Si osservi l'esempio seguente dove si moltiplica  $10011001_2$  per  $1011_2$ :

$10011001_2$	$\times$	$(153_{10})$
$1011_2$	$=$	$(11_{10})$
<hr/>		
$10011001_2$	$+$	
$10011001_2$	$+$	
$00000000_2$	$+$	
$10011001_2$	$=$	
<hr/>		
$11010010011_2$		$(1683_{10})$



## Strutture fisiche di informazione

Strutture fisiche nelle quali sono memorizzate le strutture logiche di informazione.

Supporti alle strutture fisiche di informazione nel computer:

- **Registri** (1 word)
- **Memorie** (n word)



Mentre per codificare numeri si usano tecniche basate sul loro valore, per codificare dei caratteri c'è bisogno di una relazione convenzionale, ovvero di una tabella che faccia corrispondere a una data sequenza di bit un dato carattere.

Nel progettare tale tabella bisogna tener conto ovviamente anche di caratteri non direttamente stampabili, ma che rappresentino la formattazione del testo, come ad esempio il *carriage return* CR ('a capo')

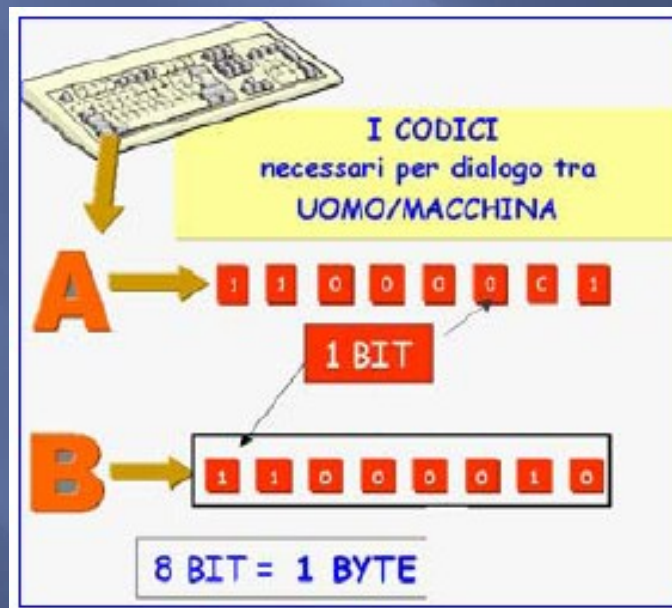


L'informazione è rappresentata mediante stringhe di bit (rappresentazione binaria).

**CODICE:** Associazione di un significato alle diverse configurazioni che una stringa può assumere

1. Codifica diretta in binario delle informazioni
2. Codifica dell'alfabeto esterno in binario

codici dei caratteri alfanumerici





## Codifica dei caratteri

I caratteri alfanumerici sono rappresentati attraverso codici che associano ad ogni simbolo dell'alfabeto una stringa di bit.

### 1. Codice **ASCII**

E' il codice più usato nei personal computer

0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	1
0	1	0	0	0	1	0	0
.	.	.	.	.	.	.	.

<b>A</b>
<b>B</b>
<b>C</b>
<b>D</b>

**Codice ASCII**  
**American**  
**Standard**  
**Code for**  
**Information**  
**Interchange**





# Tabella ASCII

digit				zone															
3	2	1	0	7	6	5	4												
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	NUL	DLE	SPACE	0	@	P		p								
0	1	0	1	SOH	DC1	!	1	A	O	a	q								
0	1	1	0	STX	DC2	"	2	B	R	b	r								
0	1	1	1	ETX	DC3	#	3	C	S	c	s								
1	0	0	0	EOT	DC4	\$	4	D	T	d	t								
1	0	0	1	ENO	NAK	%	5	E	U	e	u								
1	0	1	0	ACK	SYN	&	6	F	V	f	v								
1	0	1	1	BEL	ETB	'	7	G	W	g	w								
1	1	0	0	BS	CAN	(	8	H	X	h	x								
1	1	0	1	HT	EM	)	9	I	Y	i	y								
1	1	1	0	LF	SUB	-	:	J	Z	j	z								
1	1	1	1	VT	ESC	+	:	K		k	{								
1	1	1	0	FF	FS	,	<	L	\	l									
1	1	1	1	CR	GS	-	=	M		m	}								
1	1	1	0	SO	RS	.	>	N	^	n	~								
1	1	1	1	SI	US	/	? :	O	_	o	del								



## Esempio

Pepe... in ASCII:

01010000 01100101 01110000 01100101



P

e

p

e



## 2. Codice **EBCDIC**

Acronimo di Extended Binary Coded Decinal Interchange Code.

Ogni simbolo dell'alfabeto esterno (carattere) è codificato con **1 byte**

E' il codice più usato nei mainframe

## 3. Codice **UNICODE**

Ogni simbolo dell'alfabeto esterno (carattere) è codificato con **2 byte**

E' il codice che consente di codificare non solo i simboli della lingua inglese ma anche quelli delle principali lingue moderne ed antiche.

Windows NT utilizza tale codice



## Codifica dei numeri naturali

Numero naturale  $\leftrightarrow$  intero senza segno

Cifra binaria  $\leftrightarrow$  bit

Un numero naturale è rappresentato da una stringa di bit di lunghezza fissa (sistema binario).

- La lunghezza fissa è dovuta alla limitatezza delle risorse
- Conseguenza: **non tutti i numeri naturali possono essere rappresentati all'interno dell'elaboratore**





La lunghezza della rappresentazione dipende dal tipo di CPU

- 8 bit per Z80, 8080
- 8, 16 bit per 8086, 80286
- 8, 16, 32 bit per 80386, 80486, Pentium
- 8, 16, 32, 64 bit per Titanium

Se  $n$  è il numero di bit utilizzati, si possono rappresentare numero da 0 a  $2^n-1$

- Se il risultato dell'operazione è maggiore di  $2^n-1$ , si ha una condizione di overflow
- Se il risultato di una operazione è minore do 0, si ha una operazione di underflow.



Lo schema della macchina di Von Neumann, modello sul quale si basano praticamente tutti i moderni computer è basato su quattro elementi fondamentali:

- La memoria
- L'unità aritmetico logica (ALU)
- L'unità di controllo
- Le unità di Ingresso/uscita (I/O)



Processore  
(CPU)



# La memoria RAM

Indirizzo	Contenuto
0	Dato o istruzione
1	Dato o istruzione
2	Dato o istruzione

→ Dati e istruzioni sono codificati

Operazioni possibili:

- **Fetch (Y)** = leggi il contenuto della cella che ha indirizzo y
- **Store (Y, X)** = memorizza il valore X nella cella che ha indirizzo Y

**Random Access Memory:** il tempo necessario per leggere il contenuto di una cella è uguale per tutte le celle (cioè è uguale sia se leggo la cella 1 che la cella 65536).

La RAM è una memoria VOLATILE

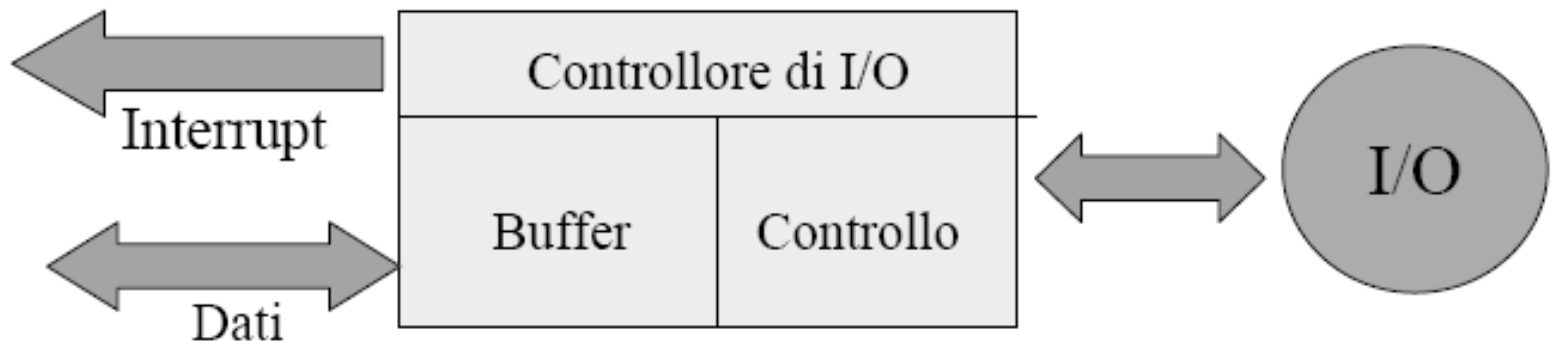


Dispositivi di interfaccia utente: tastiera, monitor, mouse...

Dispositivi di comunicazione: porte e dispositivi ad esse connessi (stampanti, modem,..)

Dispositivi di memoria di massa: Hard disk, Floppy disk, CDROM, DVD)

Tempi caratteristici di accesso: diversi ordini di grandezza più lenti della memoria RAM





# L'unità aritmetico logica: ALU

Il cuore calcolante del computer: effettua un insieme finito e predeterminato di operazioni matematiche e logiche.

Gli operandi vengono letti da registri in ingresso, e il risultato dell'operazione è scritto su un registro in uscita.

Il registro di stato (SR) riporta il segno del risultato e la presenza di riporto o di una condizione di errore.





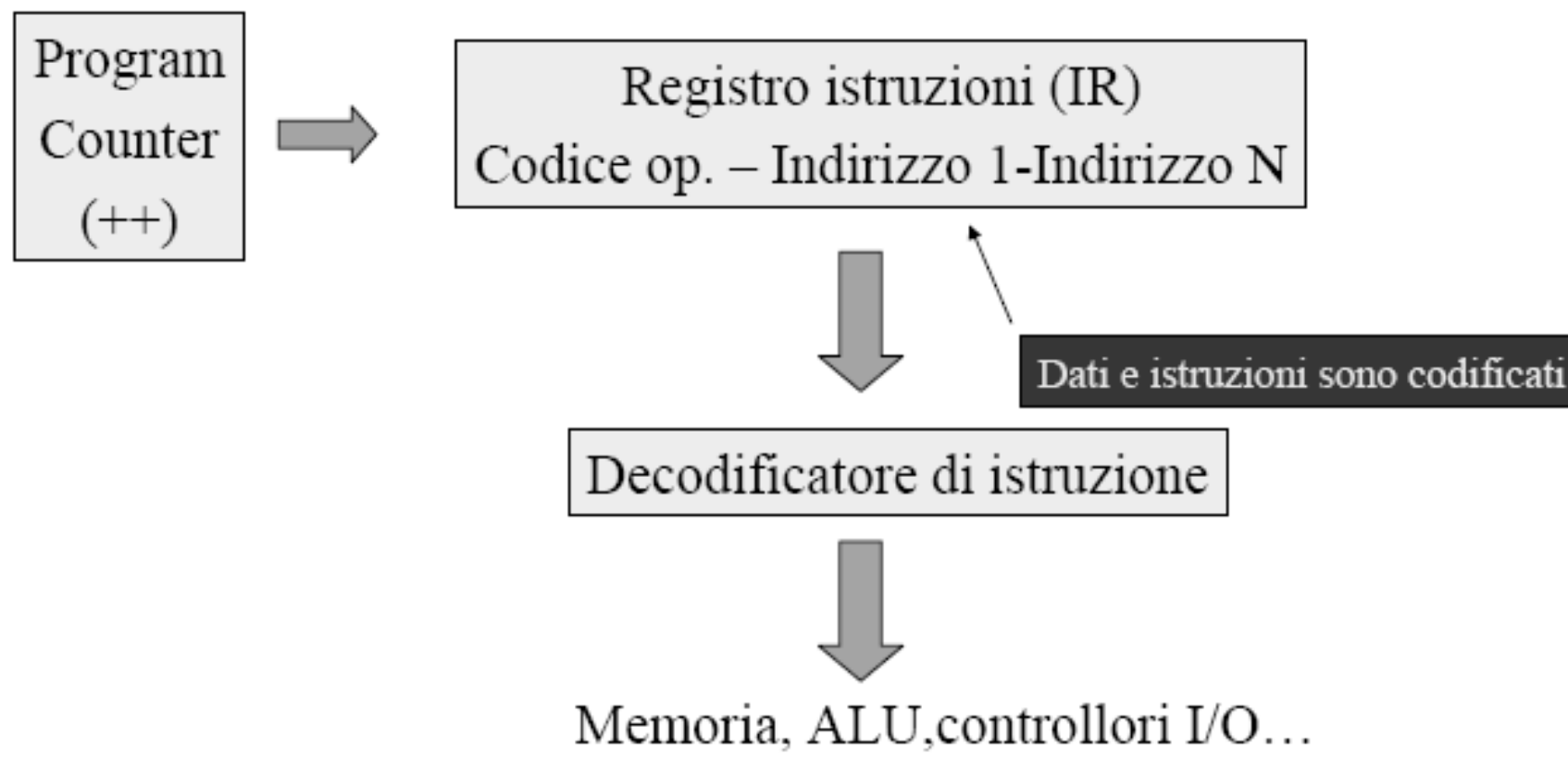
Supponiamo di voler risolvere un problema di fisica. La calcolatrice ci può servire a svolgere un determinato calcolo, ma dobbiamo essere noi a scegliere la sequenza di operazioni che ci porterà al risultato esatto. La ALU è come una calcolatrice, e la parte della CPU che gestisce la sequenza di operazioni è chiamata unità di controllo (Control Unit o CU).

Ciclo di esecuzione di un programma:

- Instruction fetch
- Decodifica
- Esegui



# L'unità di controllo





## Un esempio

Per capire bene come funziona, facciamo per un momento il lavoro della CU.

Supponiamo di avere una ALU che sa fare solo le quattro operazioni. Ad esse corrispondono i quattro codici operativi 1,2,3,4 per +,-, \* e / rispettivamente. Supponiamo di avere una convenzione per cui la prima cifra dell'istruzione è il codice operativo e i dati sono sempre scritti su quattro cifre (es. 3 = 0003; 112 = 0112 etc). A cosa corrispondono le seguenti istruzioni che leggiamo nella RAM?

100030002 = calcola 3+2  
400080002 = calcola 8/2  
300100010 = calcola 10\*10  
500110003 = Non valida



## Riassumendo

La struttura di Von Neumann permette di calcolare una sequenza di istruzioni opportunamente codificate e memorizzate e di controllare il flusso dell'esecuzione. La gestione dell'I/O è delegata a opportuni controllori per ottimizzare le prestazioni.

La condizione chiave per la realizzazione del calcolatore è disporre di un sistema efficiente e affidabile di codifica dell'informazione, ovvero dei dati e delle istruzioni che devono essere via via eseguite.

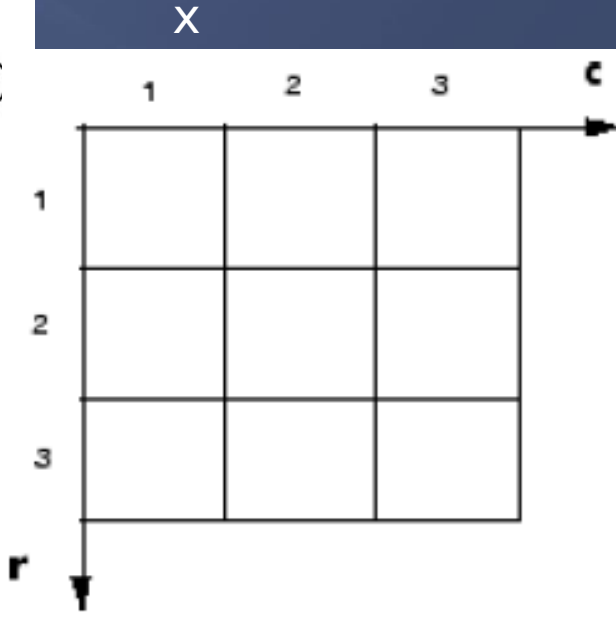
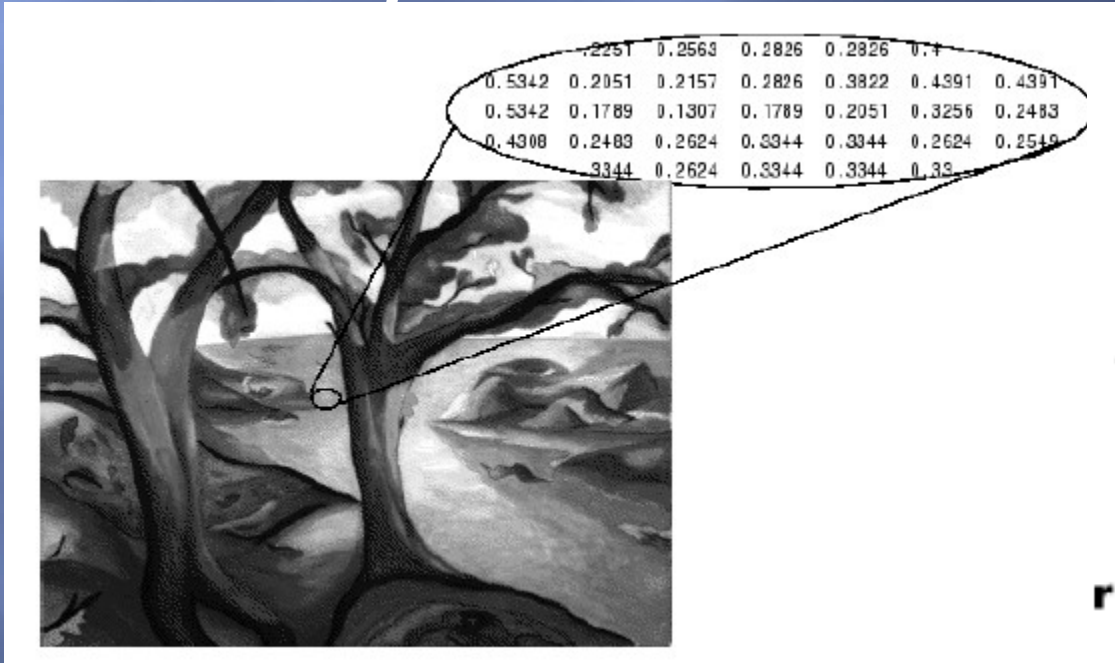


# Le immagini digitali



Una immagine digitale è una matrice di pixel quadrati ordinati in righe e colonne.

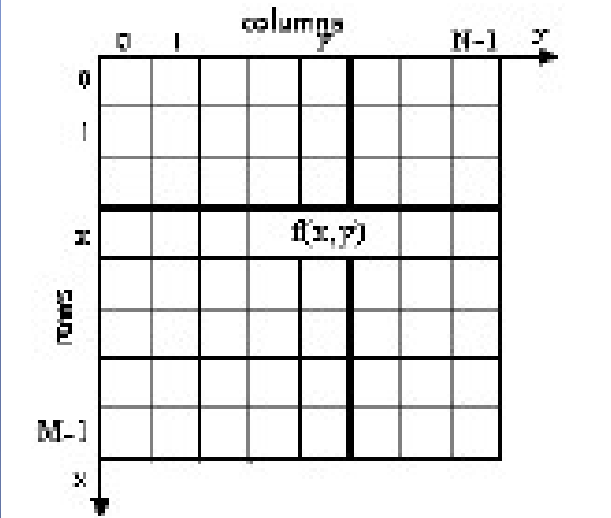
Una immagine è una funzione dell'intensità  $f(x,y)$  dove  $f$  è la brillantezza del punto  $(x,y)$ , e  $x$  e  $y$  rappresentano le coordinate spaziali di un **picture element (pixel)**



The Pixel Coordinate System

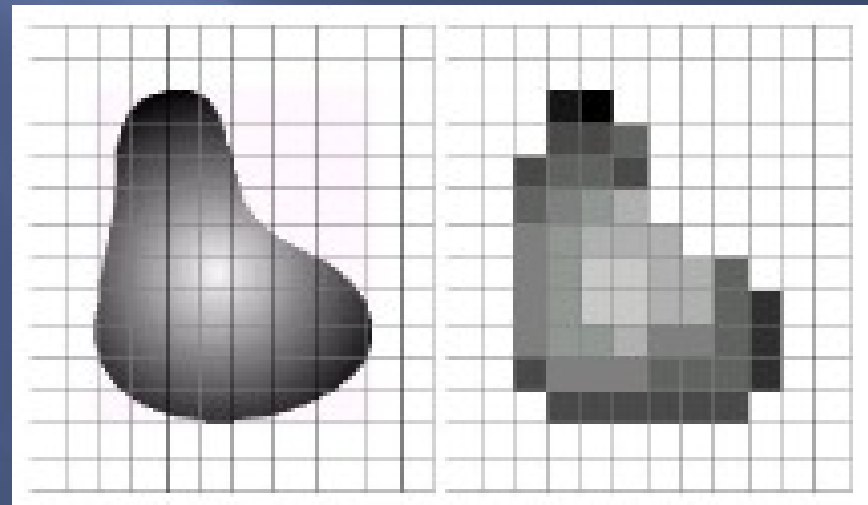


# Immagini



Ogni pixel rappresenta l'intensità nella corrispondente posizione della griglia di campionamento.

Un pixel rappresenta in realtà non soltanto un punto dell'immagine, ma piuttosto una regione rettangolare coincidente con una cella della griglia. Il valore associato al pixel deve rappresentare pertanto l'intensità media nella cella







La **risoluzione spaziale** è il più piccolo dettaglio distinguibile in una immagine.

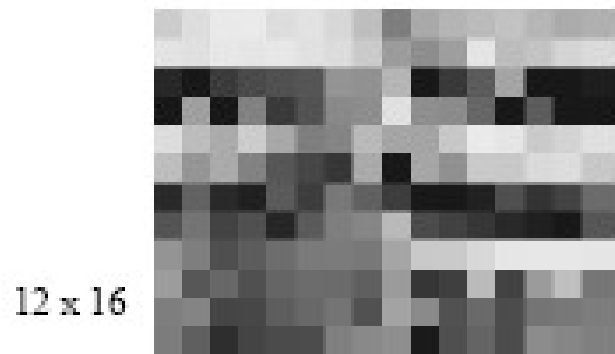
Viene espressa normalmente in elementi per unità di lunghezza, per esempio pixel (oppure dot) per **inch**.

Quando non è necessaria una valutazione della risoluzione che metta in relazione il numero di pixel con il livello di dettaglio della scena originale, è usuale dire semplicemente che una immagine di dimensioni  $M \times N$  ha una risoluzione spaziale  $M \times N$ .



Una stessa immagine può essere rappresentata con un numero differente di pixel, per esempio modificando le dimensioni dei pixel, a parità di dimensioni dell'immagine:

# Immagini





Con pixel di grande dimensione, non solo la risoluzione spaziale è scadente, ma appaiono ben visibili le discontinuità di grigio al confine tra i pixel.

Man mano che la dimensione dei pixel si riduce, l'effetto diventa meno visibile, fino al punto che si ha l'impressione di una immagine continua.

Questo accade quando la dimensione dei pixel diventa più piccola della risoluzione spaziale del sistema visivo umano.

Siccome quest'ultima dipende dalla distanza e dalle altre condizioni di osservazione, in generale non è definibile a priori il numero di pixel necessari a garantire una buona qualità dell'immagine.

Sicuramente la dimensione dei pixel deve essere piccola in relazione alla scala degli oggetti rappresentati nell'immagine.



Se la dimensione del pixel resta invariata, la relazione di risoluzione provoca invece una variazione delle dimensioni dell'immagine:







## • Numero di Piani

il numero di piani in un'immagine è il numero di matrici che compongono l'immagine

gray level = 1 piano

true color = 3 piani (Red, Green, Blue)

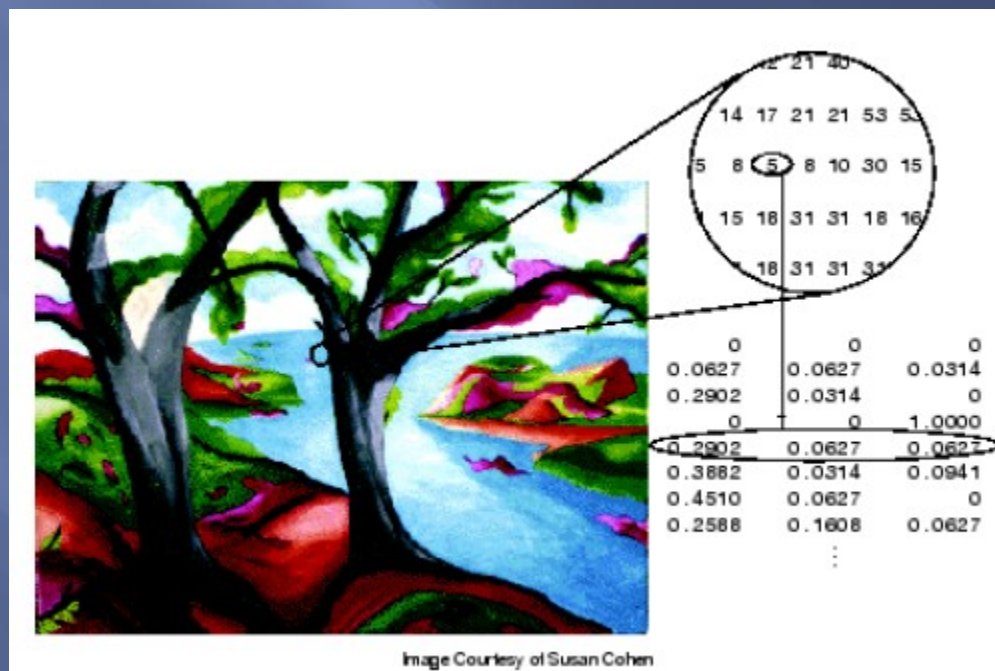


Image Courtesy of Susan Cohen



## -Immagine Binaria

$$A(i, j) = \{1 \text{ or } 0\}$$

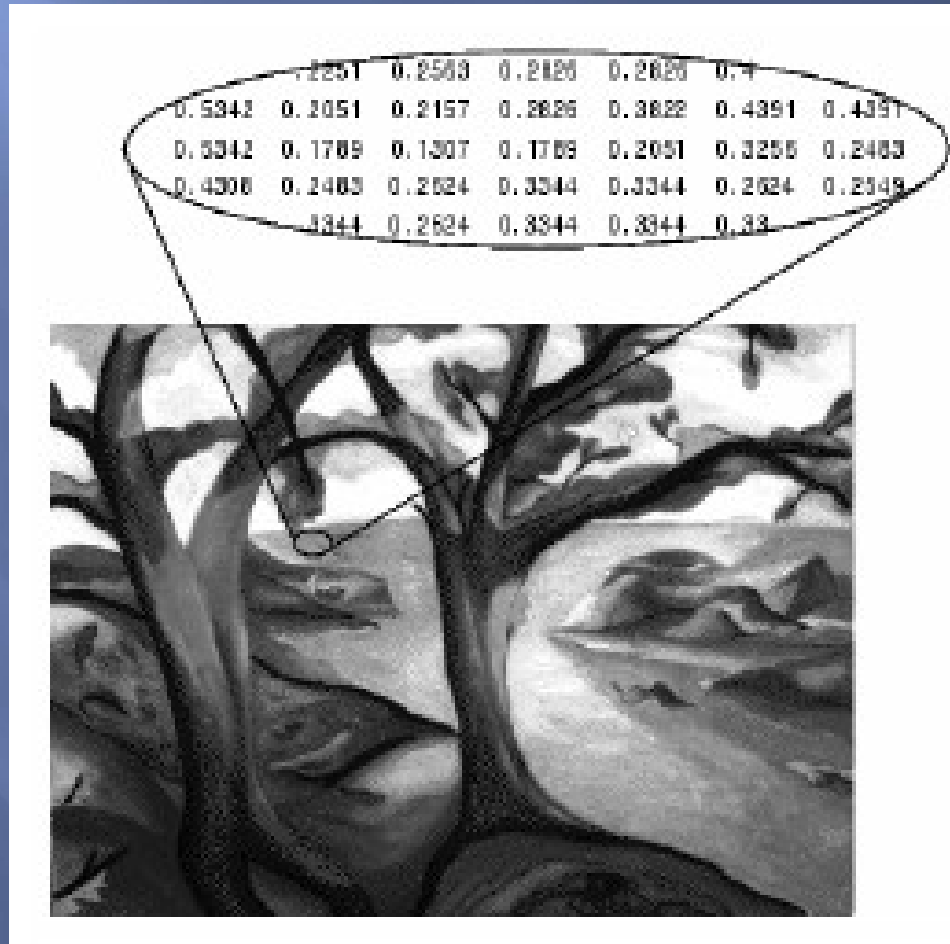


0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	1	0	0	1	0	0
0	0	1	0	1	0	0
0	0	0	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0



# Immagine a livello di grigio

8 bit (uint8, 0-255) -> BMP, TIFF, PNG, JPEG  
16 bit (uint16, 0-65535) -> PNG





# Immagini a colori RGB

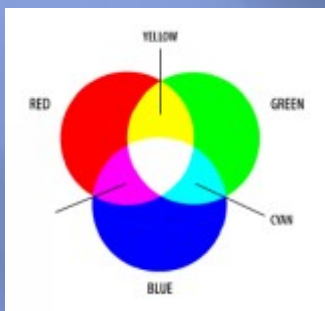
24 bit =  $256^3$  = 16 milioni di colori

-> BMP, TIFF, PNG, JPEG

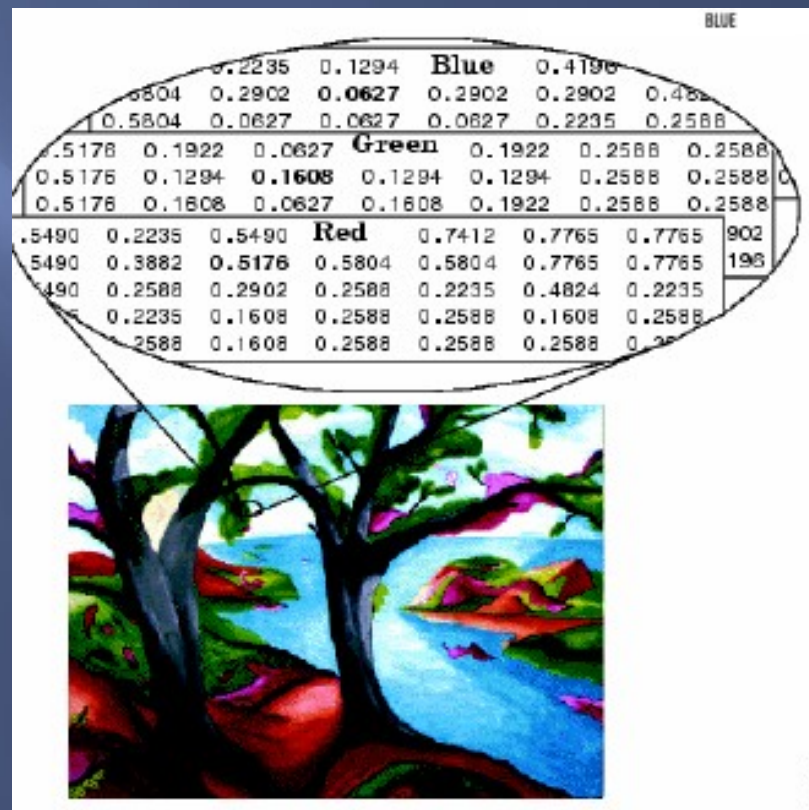
8 bit -> red

8 bit -> green

8 bit -> blue



Immagini Digitali





In alcune applicazioni può essere utile evidenziare il contributo di *specifici* bit alla apparenza complessiva dell'immagine.

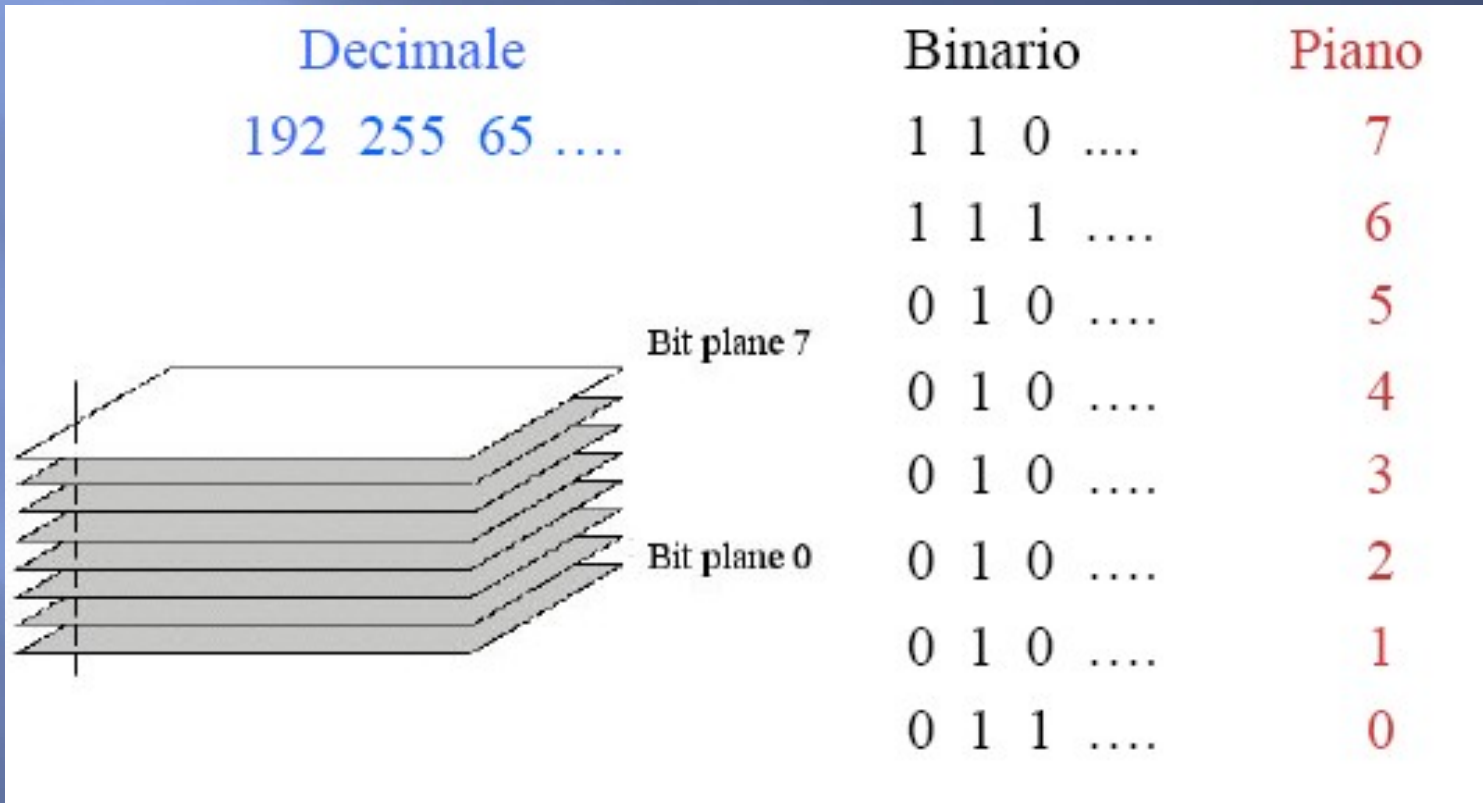
Se per codificare i pixel di una immagine si utilizzano 8 bit, si può considerare l'immagine costituita da 8 piani di 1 bit, dal piano 0 (che contiene il bit meno significativo di ogni pixel) al piano 7 (che contiene il bit più significativo).

Si consideri il valore di ogni pixel dell'immagine espresso in binario. Ovviamente il bit più significativo si trova nel piano 7, il bit di peso immediatamente inferiore si trova nel piano 6, e così via per tutti gli altri.



# Bit plane slicing

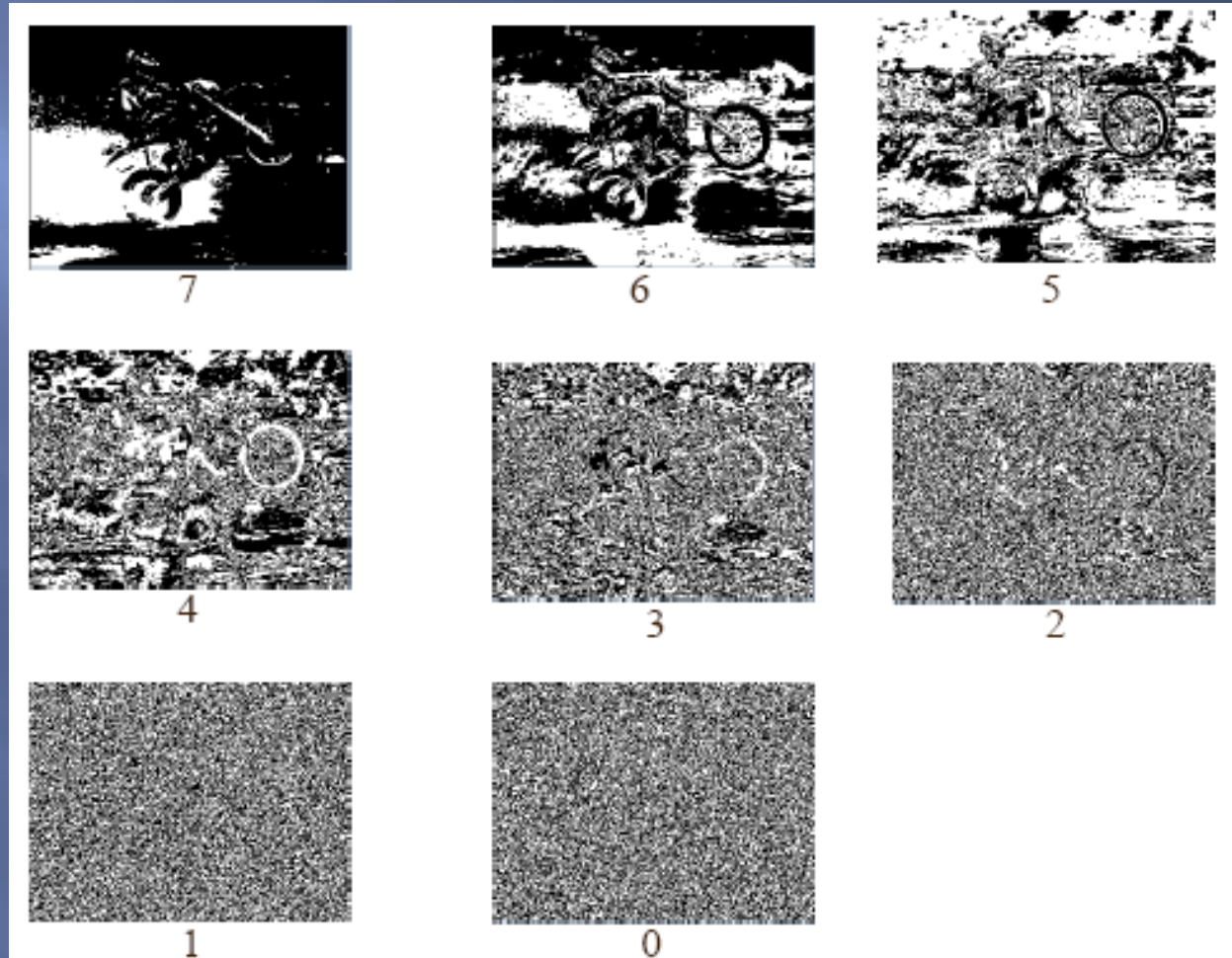
Per esempio, consideriamo un'immagine i cui primi pixel abbiano rispettivamente i valori 192, 255, 65, .... Si ha:





Un esempio reale:

# Bit plane splicing





Il **bit-plane slicing** consiste quindi nello scomporre un'immagine in otto immagini binarie, rappresentate dai vari piani.

I piani di bit più significativi contengono informazioni sulla struttura dell'immagine, mentre quelli via via meno significativi forniscono i dettagli sempre più piccoli.

Si noti che solo il piano dal 7 al 3 contengono dati significativi dal punto di vista visuale.

Inoltre l'immagine binaria del piano 7 è esattamente quella che si ottiene binarizzando con soglia pari a 128 l'immagine di partenza.



Le tecniche di compressione dei dati, unite a studi di psicoacustica hanno permesso di realizzare rappresentazioni di filmati e di sequenze audio di dimensioni accettabili, dell'ordine di qualche MB. Questo ha aperto la strada all'utilizzo della multimedialità informatica.

Tipici formati audio/video sono MP3, MPEG, AVI, RealAudio, RealVideo.



## Immagini Bitmap

Un'immagine bitmap è una matrice di pixel di vari colori. Le dimensioni sono determinate dal numero di pixel che la compongono.

Un'immagine 640x480 ha 640 pixel orizzontali e 480 pixel verticali. Se un'immagine ha 24 bit/pixel ogni pixel può assumere 16 milioni ( $2^{24}$ ) di colori diversi.

Lo spazio necessario per memorizzarla è  $640 \times 480 \times 24 = 7372800$  bit ovvero circa 921 kB. (byte = 8 bit).

Prima di stampare un'immagine la si converte in bitmap, con un programma chiamato rasterizzatore.



## Formati Bitmap

- .bmp formato standar. Visualizza fino a 16.7 milioni (24 bit/pixel) di colori diversi.
- .pcx è utilizzato con i programmi di tipo paint di word processing. Visualizzo fino a 24 bit/pixel
- .gif (Graphic Interchange Format) visualizza fino a 256 colori diversi (8 bit/pixel). È diffuso nel web in quanto compresso.
- .jpeg (Joint Picture Experts Group) visualizza fino a 24 bit/pixel, ma è ancora più compresso del .gif
- .tiff (Tagged Image File Format) offre l'immagine di qualità più alta e visualizza fino a 24 bit/pixel. I file in questo formato hanno grandi dimensioni.



# Compressione

- Lossless (ovvero senza perdite o reversibile) riduca non meno di  $1/3$  delle dimensioni originali. Viene impiegato quando bisogna salvaguardare anche i dettagli dei files originali.
- Lossy (con perdita) offre una qualità inferiore a quella di partenza perché alcuni dettagli vanno persi. Consente di comprimere fino a 200:1.



# Compressione

.bmp 3841 Kb



.tiff 3842 Kb



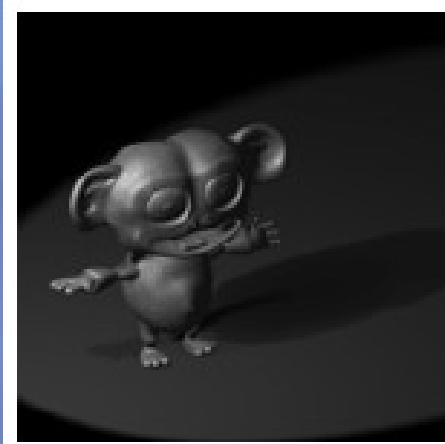
.jpeg 425 Kb



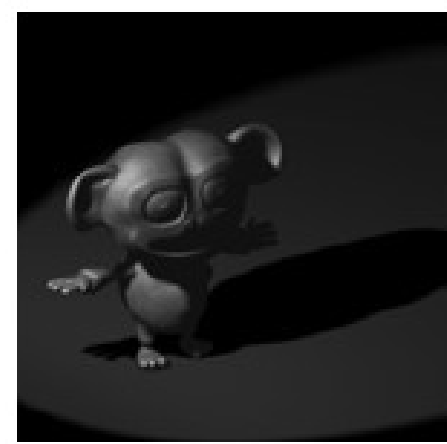


Ombreggiatura – serve a dare corpo e solidità ad un'immagine

Morphing - fonde insieme due immagini per crearne una terza.



Due diversi tipi di ombreggiature

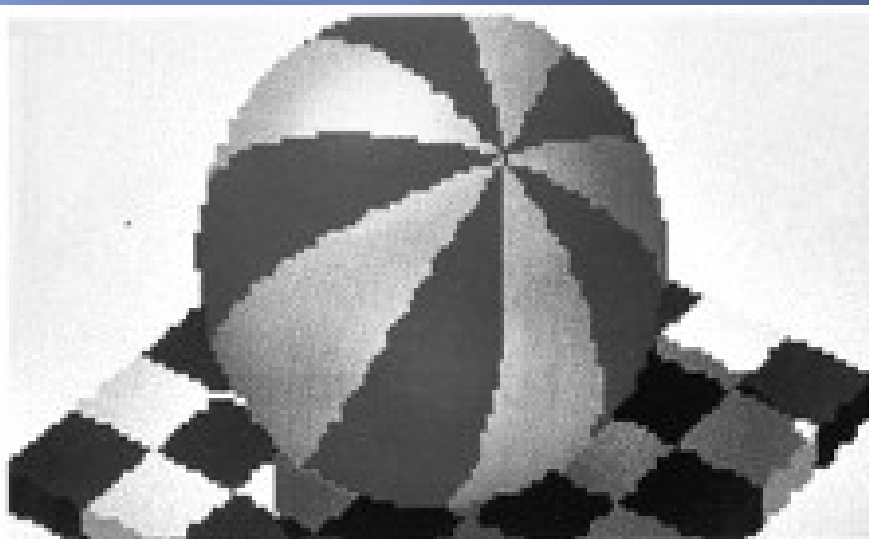


Programmi per la manipolazione avanzata di immagini:

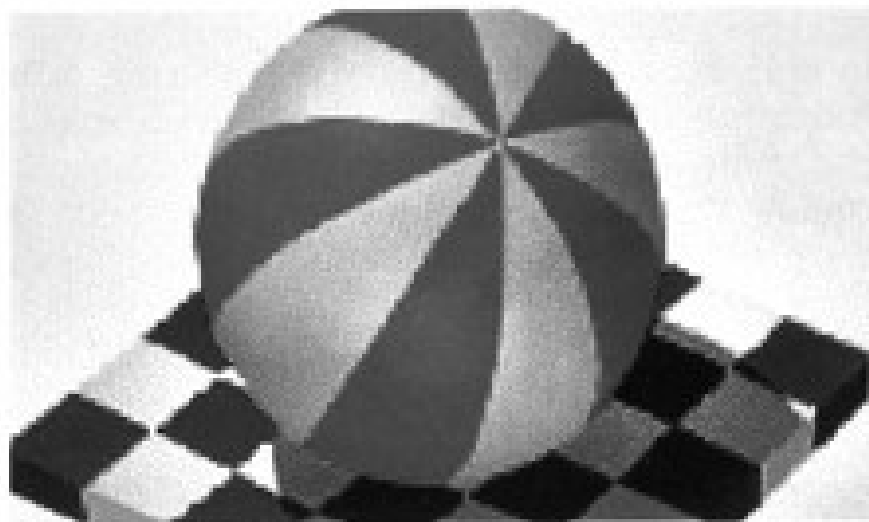
- Adobe Photoshop
- GIMP (gratuito !!!!)



# Antialiasing



Prima dell'anti-aliasing



Dopo l'anti-aliasing

## Antialiasing

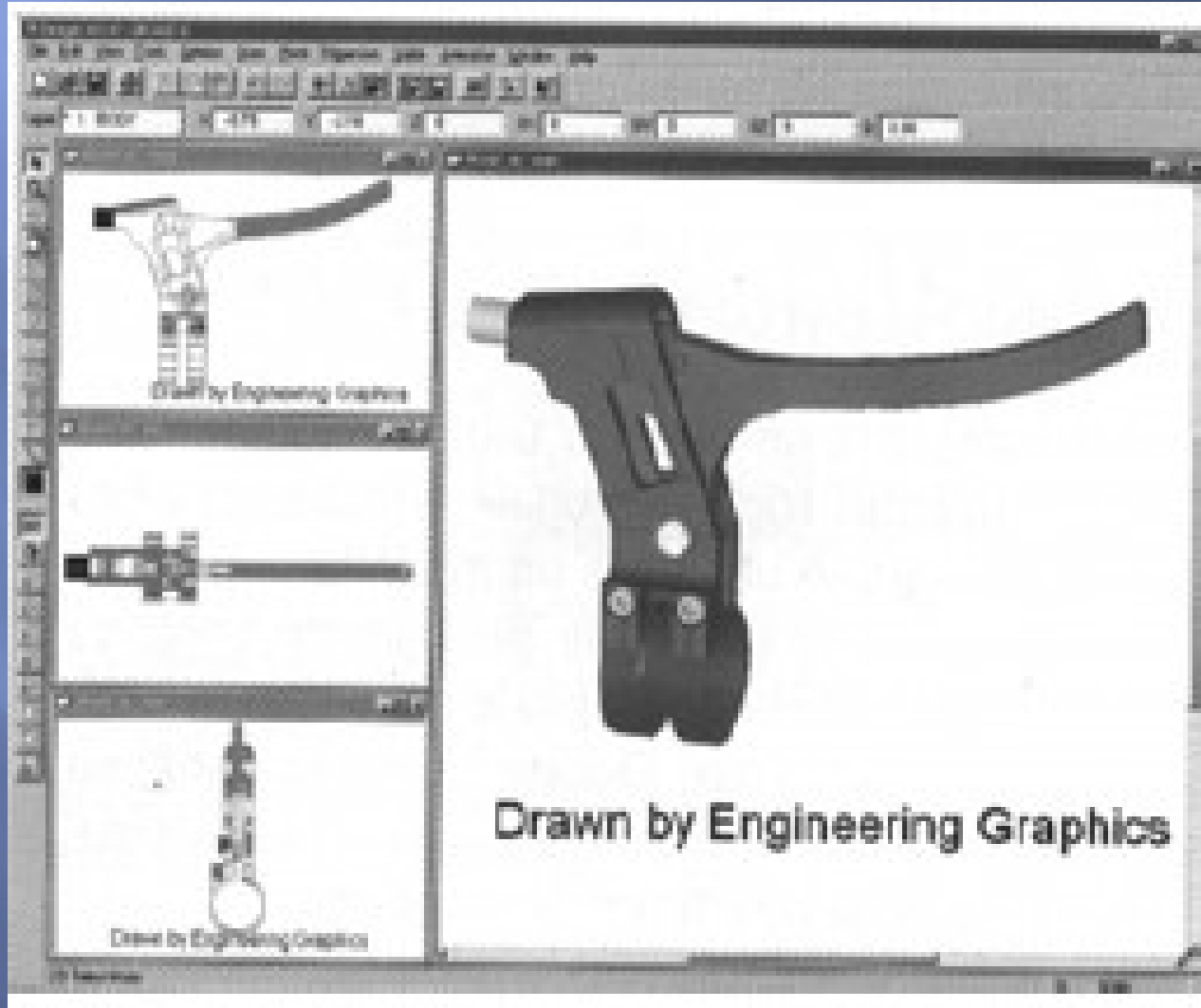
smorza le  
frastagliature dei  
contorni di un  
oggetto, dovute  
alla struttura  
discreta dei pixel.





Immagine tridimensionale vista da diverse prospettive  
ottenuta con un programma di CAD

Immagini CAD





I dati audio sono digitalizzati, mediante una tecnica di campionamento delle onde sonore analogiche, e memorizzati. Una volta riascoltati la scheda audio e i diffusori del PC utilizzano i valori digitali memorizzati per ricostruire le onde analogiche originali.

Un esempio di programma audio è il Sound Recorder di Windows, con il quale è possibile modificare i files sonori.

Il formato più comune per i files sonori di Windows è WAVE.

Il formato più utilizzato per audio compressi è mp3.

È possibile creare musica con il computer con programmi che funzionano da sintetizzatore musicale.



Un filmato visto su PC deve essere prima digitalizzato. Ciò comporta la creazione di files di dimensioni immense. Un lungometraggio digitalizzato può occupare fino a 100 GB! Un secondo di video su uno schermo 640x480 ad una velocità di 30 fotogrammi/sec. Richiede 27 MB di dati!



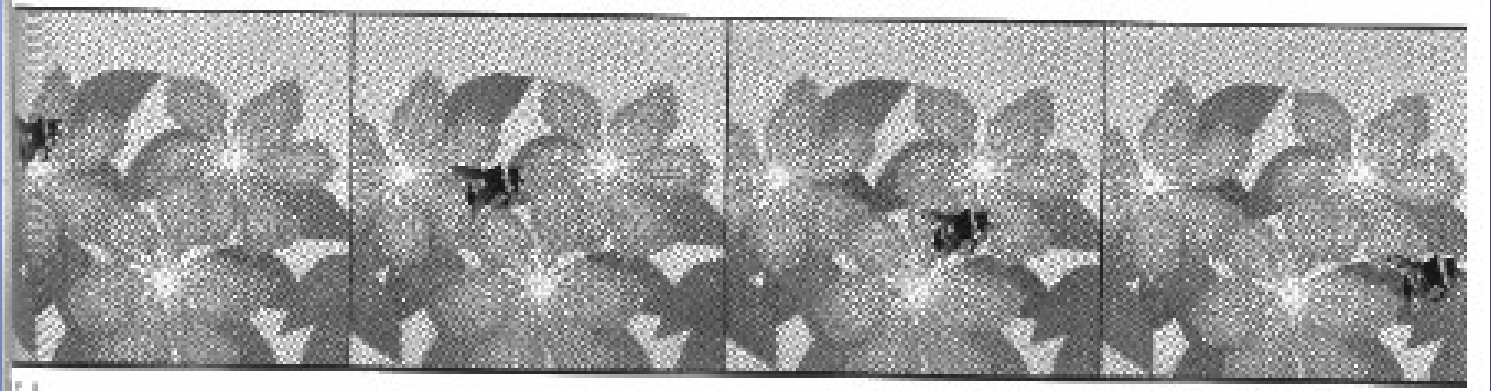
È necessario comprimere le immagini e i video (compressione-decompressione CODEC).



- Intraframe (o spaziale) elimina dati all'interno di un singolo fotogramma
- Interframe (o temporale) elimina dati che si ripetono in fotogrammi successivi. Il formato MPEG (Moving Picture Experts Group) lascia non compresso un fotogramma ogni 30 sec. (fotogramma chiave).



## Video non compresso



## Video compresso con tecnica interframe

