

Elementi di Informatica

Facoltà di Economia

Rappresentazione di dati

Lezione: 10/10/19



Il **sistema numerico binario** è un sistema numerico posizionale in base 2. Esso utilizza solo due simboli, di solito indicati con 0 e 1, invece delle dieci cifre utilizzate dal sistema numerico decimale. Ciascuno dei numeri espressi nel sistema numerico binario è definito "numero binario".

In informatica il sistema binario è utilizzato per la rappresentazione interna dell'informazione dalla quasi totalità degli elaboratori elettronici, in quanto le caratteristiche fisiche dei circuiti digitali rendono molto conveniente la gestione di due soli valori, rappresentati fisicamente da due diversi livelli di tensione elettrica. Tali valori assumono convenzionalmente il significato numerico di 0 e 1 o quelli di vero e falso della logica booleana.



Analizziamo il metodo con cui è possibile **codificare le informazioni tramite bit**: testi, immagini , suoni.

Mediante l'utilizzo di 0 e 1 si possono rappresentare non solo tutti i possibili numeri esistenti, ma anche tutte le parole, le immagini, i video, i suoni, ed in generale tutte le altre più disparate tipologie di informazioni digitali.

Le informazioni testuali sono rappresentate tramite un codice in cui ad ogni carattere è associata una sequenza univoca di bit.

ANSI uniforma attraverso il codice ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) 8 bit per rappresentare caratteri, numeri, simboli

Consideriamo la parola

H	e	l	l	o
01001000	01100101	01101100	01101100	01101111

Conversione binario decimale

$$1101001 = 1*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 64 + 32 + 8 + 1 = 105$$



Conversione decimale binario

$$\begin{aligned}
 23 : 2 &= 11 + \text{RESTO } 1 \\
 11 : 2 &= 5 + \text{RESTO } 1 \\
 5 : 2 &= 2 + \text{RESTO } 1 \\
 2 : 2 &= 1 + \text{RESTO } 0 \\
 1 : 2 &= 0 + \text{RESTO } 1
 \end{aligned}$$

$$23_{(10)} = 10111_{(2)}$$

$$\begin{aligned}
 118 : 2 &= 59 + \text{RESTO } 0 \\
 59 : 2 &= 29 + \text{RESTO } 1 \\
 29 : 2 &= 14 + \text{RESTO } 1 \\
 14 : 2 &= 7 + \text{RESTO } 0 \\
 7 : 2 &= 3 + \text{RESTO } 1 \\
 3 : 2 &= 1 + \text{RESTO } 1 \\
 1 : 2 &= 0 + \text{RESTO } 1
 \end{aligned}$$

$$118_{(10)} = 1110110_{(2)}$$

Operazione di addizione

				1+
0+	1+	0+	1+	
0	0	1	1	1
0	1	1	10	11

1←	1←		1←	1←	1←	1←					
	1	1	0	0	0	1	0	1	0	1	+
		1	0	0	0	1	0	1	1	1	
	1	0	0	0	0	1	0	1	1	0	0



ESEMPI

Conversione binario decimale

$$101011_{(2)} = 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 32 + 8 + 2 + 1 = 43$$

$$1100_{(2)} = 1*2^3 + 1*2^2 + 0*2^1 + 0*2^0 = 8 + 4 = 12$$

$$10101010_{(2)} = 1*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 128 + 32 + 8 + 2 = 170$$

Conversione decimale binario

$$45:2=22 + \text{RESTO } 1$$

$$22:2=11 + \text{RESTO } 0$$

$$11:2= 5 + \text{RESTO } 1$$

$$5:2= 2 + \text{RESTO } 1$$

$$2:2= 1 + \text{RESTO } 0$$

$$1:2= 0 + \text{RESTO } 1$$



$$45_{(2)} = 101101_{(10)}$$

$$70:2=35 + \text{RESTO } 0$$

$$35:2=17 + \text{RESTO } 1$$

$$17:2= 8 + \text{RESTO } 1$$

$$8:2= 4 + \text{RESTO } 0$$

$$4:2= 2 + \text{RESTO } 0$$

$$2:2= 1 + \text{RESTO } 0$$

$$1:2 = 0 + \text{RESTO } 1$$

$$70_{(2)} = 1000110_{(10)}$$



Il programma Excel è un foglio elettronico della Microsoft facente parte del pacchetto Office. I fogli elettronici (o fogli di calcolo) sono applicazioni utili per gestire grossi quantitativi di dati organizzati generalmente in tabelle o sotto forma di elenchi, su cui è possibile effettuare calcoli anche complessi in modo relativamente semplice.

ES. conversione da decimale a binario

=INT(B2/2)

	A	B	C	D
1			resto	
2		70	0	
3		35	1	
4		17	1	
5		8	0	
6		4	0	
7		2	0	
8		1	1	
9		0		
10				
11				



Excel dispone di **1.048.576** righe per **16.384** colonne suddivise in celle. Ogni cella è identificata all'interno della griglia da una lettera e da un numero; la lettera indica la colonna, mentre il numero indica la riga. Caratteristica fondamentale di un foglio elettronico è che i dati inseriti possono essere facilmente gestiti; la modifica di un valore provoca l'aggiornamento dei valori contenuti nelle celle collegate.



Per rappresentare numeri interi negativi si usa la cosiddetta rappresentazione in complemento a 2

Calcolo del complemento a due di un numero su k bit

ES1. Calcolo del complemento a 2 di $12_{(10)}$ su 5 bit. Con n bit posso rappresentare numeri nell'intervallo $[0, 2^{n-1}-1]$. Con 5 bit sono nell'intervallo $[0, 15]$ e posso rappresentare 12

$12:2 = 6$ RESTO = 0
 $6:2 = 3$ RESTO = 0
 $3:2 = 1$ RESTO = 1
 $1:2 = 0$ RESTO = 1

→ $1100_{(2)}$

Andiamo ad invertire gli zero con gli 1

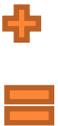
Se il numero è positivo il bit più significativo ha valore 0

0	1	1	0	0
1	0	0	1	1

Sommiamo 1



1



Sommando queste due quantità su 5 bit si ottengono tutti zero

1	0	1	0	0
---	---	---	---	---



$-12_{(10)}$

0	0	0	0	0
---	---	---	---	---



ES2. Calcolo del complemento a 2 di $35_{(10)}$ su 7 bit.

$35:2 = 17$ RESTO = **1**
 $17:2 = 8$ RESTO = **1**
 $8:2 = 4$ RESTO = **0**
 $4:2 = 2$ RESTO = **0**
 $2:2 = 1$ RESTO = **0**
 $1:2 = 0$ RESTO = **1**



→ $100011_{(2)}$

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

1

1	0	1	1	1	0	1
---	---	---	---	---	---	---



-35₍₁₀₎



Infatti:

0	1	0	0	0	1	1
---	---	---	---	---	---	---

+

1	0	1	1	1	0	1
---	---	---	---	---	---	---

=

0	0	0	0	0	0	0
---	---	---	---	---	---	---



Conversione dei dati

Nei calcolatori, le informazioni sono codificate come configurazione di zeri (0) e uno (1). Tali cifre vengono chiamate **bit** (binary digit). 1 **byte** è un gruppo di bit (8 bit)

bit 0 : valore falso

bit 1: valore vero

Manipolare bit significa manipolare valori vero/falso

Operazioni booleane (mat. George Boole 1815-1864) e operatori, essi combinano due valori di ingresso (input) per produrre un terzo valore in uscita (output).

AND

0 0 1 1

0 1 0 1

0 0 0 1

OR

0 0 1 1

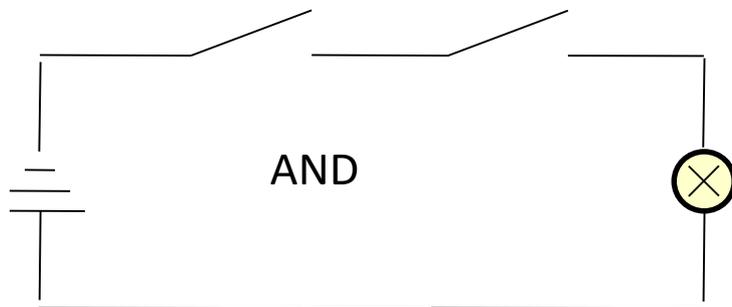
0 1 0 1

0 1 1 1



INPUT		OUTPUT
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

AND, prende in input due operandi e produce in output un booleano, attenendosi al seguente comportamento: Se entrambi gli operatori sono true(1) allora l'output è true; in tutti gli altri casi l'output è uguale a false(0).

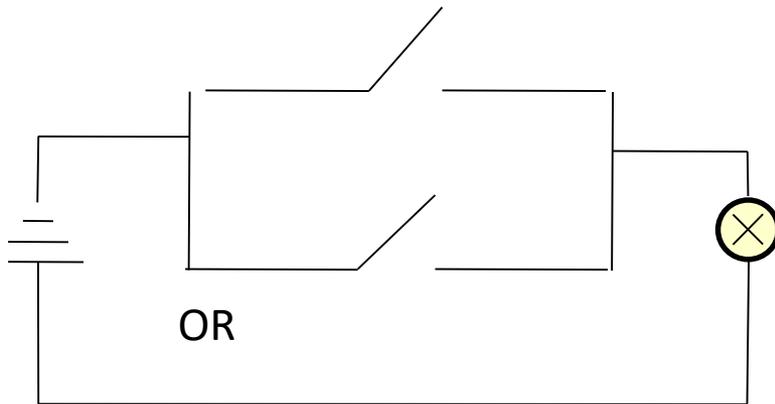


E' necessario chiudere entrambi gli interruttori per far accendere la lampadina



INPUT		OUTPUT
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

OR, prende in input due operandi e produce in output un booleano, attenendosi al seguente comportamento: Se almeno uno degli operandi è uguale a true, l'output è true; altrimenti, se nessuno dei due operandi è uguale a true l'output sarà false.

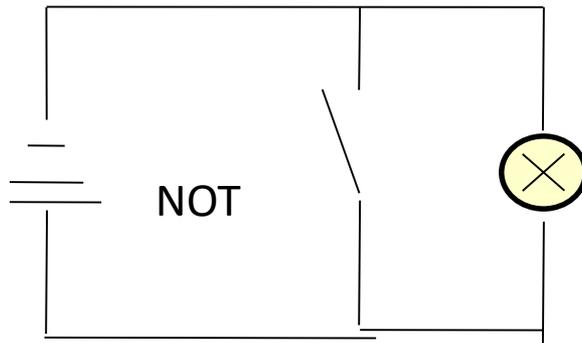


Basta chiudere uno degli interruttori per far accendere la lampadina



INPUT	OUTPUT
A	NOT A
0	1
1	0

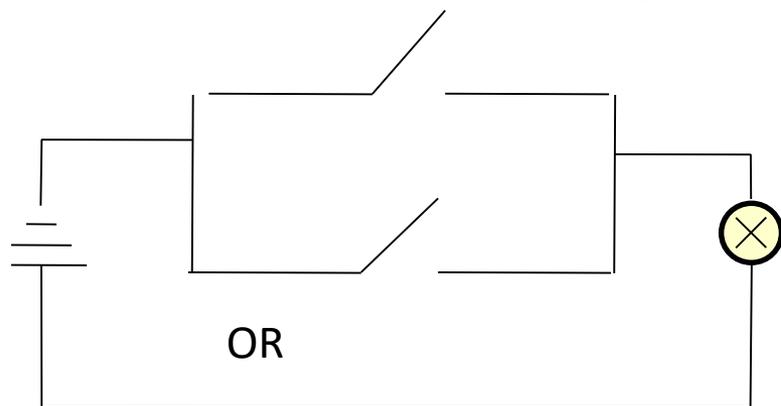
NOT, prende in input un solo operando e produce in output un booleano, attenendosi al seguente comportamento: Se l'operando di input è true allora l'output sarà false. Se, invece l'operando di input è false, allora l'output sarà uguale a true. In altri termini, l'operatore di NOT prende un input e ne restituisce l'esatto contrario.



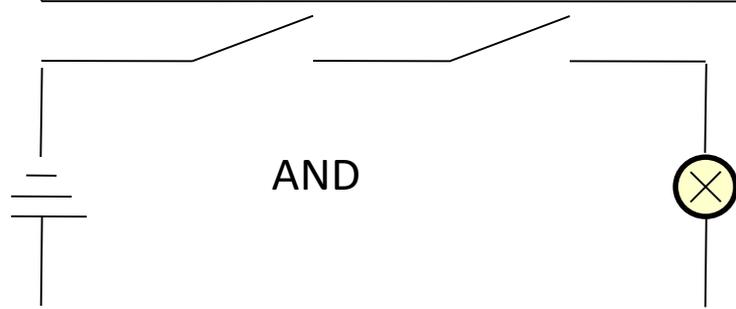
Ad interruttore aperto la lampadina è accesa



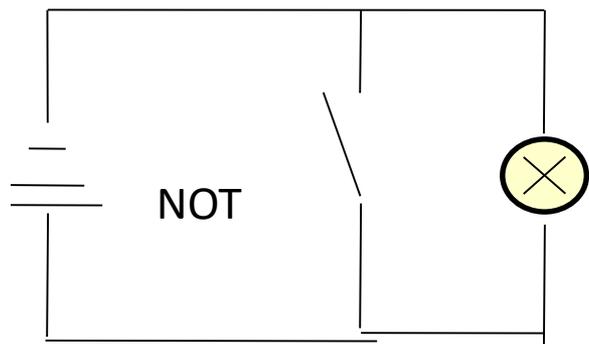
Conversione dei dati- Analogia con un circuito elettrico



Basta chiudere uno degli interruttori per far accendere la lampadina



E' necessario chiudere entrambi gli interruttori per far accendere la lampadina

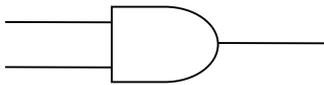


Ad interruttore aperto la lampadina è accesa

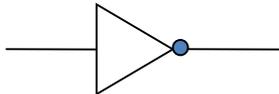
Un dispositivo che, dati i valori d'ingresso, produce l'uscita di un'operazione booleana è chiamato porta logica (gate).

Le prime porte logiche erano realizzate con dispositivi elettrici (relè), attualmente sono realizzate con circuiti elettronici, dove le cifre 0,1 sono rappresentati da diversi livelli di tensione.

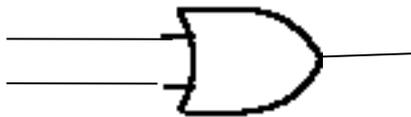
AND



NOT



OR



**La porta NOR restituisce la negazione di una porta OR e quindi restituisce 1 solo quando tutti i valori in ingresso sono 0.
Segue la tavola di verità:**

OR

INPUT		OUTPUT
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Negazione

NOR

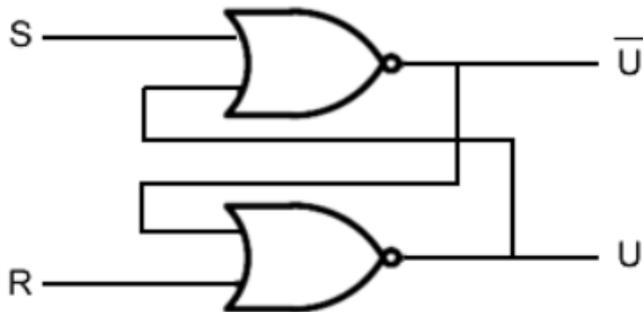
INPUT		OUTPUT
A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



CIRCUITO FLIP FLOP

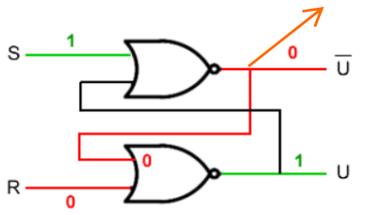
I **flip-flop** (o *bistabili*) sono circuiti elettronici sequenziali molto semplici, utilizzati nell'elettronica digitale come dispositivi di memoria elementare. Il nome deriva dal rumore che facevano i primi circuiti di questo tipo, costruiti con relè che permettevano il cambiamento di stato.

Si tratta di un circuito sequenziale, perché l'output non dipende soltanto dallo stato degli input correnti ma anche dallo stato precedente.

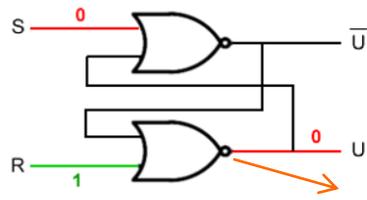


Esistono diversi modi per realizzare un flip flop. Ad esempio, posso usare due porte logiche NOR con retroazione.

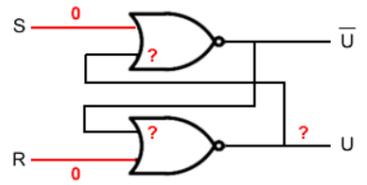




Sicuramente l'uscita è 0



Sicuramente l'uscita è 0



In questo caso l'output dipende dalle retroazioni ossia dallo stato precedente del circuito.

NOR

INPUT		OUTPUT
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



S	R	Q	\bar{Q}
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	0	0

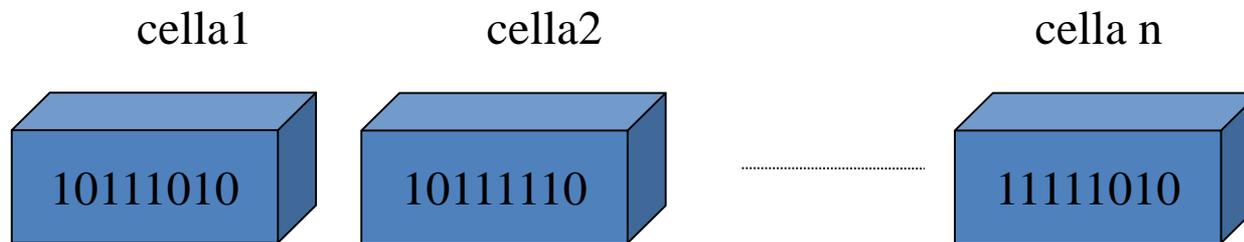


È il flip-flop più semplice dal punto di vista circuitale e fu anche il primo ad essere realizzato. Ha due ingressi s (*Set*) e r (*Reset*, detto anche *Clear*) e due uscite q e $q_{\bar{}}$ (q complementato). È una rete sequenziale asincrona che si evolve in accordo alle seguenti specifiche: **quando lo stato d'ingresso è $s=0$ e $r=1$ il flip-flop si resetta, cioè porta a 0 il valore della variabile d'uscita q e a 1 la variabile d'uscita $q_{\bar{}}$; quando lo stato d'ingresso è $s=1$ e $r=0$ il flip-flop si setta cioè porta a 1 il valore della variabile d'uscita q e a 0 la variabile d'uscita $q_{\bar{}}$; quando lo stato d'ingresso è $s=0$ e $r=0$ il flip-flop conserva, cioè mantiene inalterato il valore di entrambe le variabili d'uscita.**

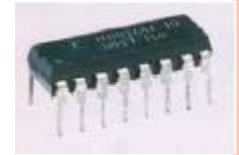


Per archiviare i dati, un computer contiene un gran numero di circuiti flip-flop, ognuno in grado di memorizzare un singolo bit. I circuiti di memoria sono organizzati in unità chiamate **celle** di memoria.

Per identificare le singole celle di memoria ad ognuna di esse è assegnata una denominazione **univoca** chiamata **indirizzo**



Poiché la memoria principale di un computer è organizzata sotto forma di singole celle dotate di indirizzo, è possibile accedere a tali celle a secondo della necessità.



- RAM (Random Access Memory)

Quando si esegue un qualsiasi programma su PC, la CPU ha bisogno di usare una parte di memoria per far girare il programma stesso: deve cioè trovare una parte di memoria “libera”, non occupata da dati. La RAM “mette a disposizione” della CPU proprio la memoria di cui ha bisogno. Una volta chiuso il programma, le modifiche effettuate, se non opportunamente salvate sul disco rigido o su altra memoria non volatile, verranno perse.



Non si potrebbe utilizzare la memoria del disco rigido



La risposta è sì, tuttavia questo farebbe aumentare significativamente i tempi di elaborazione. In un disco rigido la scrittura dei dati è sequenziale: i dati cioè vengono memorizzati l'uno dietro l'altro. Quindi prima di trovare della memoria libera la CPU potrebbe impiegare molto tempo.

- La memoria RAM

- è detta anche volatile, poiché perde il suo contenuto quando il computer viene spento,
- è detta random poiché l'accesso a ciascuna locazione di memoria è diretto e non sequenziale.

Avere un quantitativo abbastanza elevato di memoria RAM permette di utilizzare più applicazioni nello stesso momento senza che il dispositivo rallenti

- ROM (Read Only Memory)

La memoria di sola lettura (*“read only memory”* in inglese, detta anche **memoria ROM**) è un supporto di archiviazione di massa progettato e realizzato per conservare dati in maniera permanente e utilizzato su computer e altri dispositivi elettronici (lettori mp3, smartphone, tablet, fotocamere digitali). Dal momento che i dati contenuti nella memoria ROM possono essere modificati (se possibile) con grande difficoltà e in maniera molto lenta, le memorie di questo tipo sono utilizzate è generalmente usata per contenere programmi (firmware) che il computer frequentemente utilizza (per esempio le istruzioni eseguite quando il computer viene acceso,). La memoria di sola lettura, inoltre, è **non volatile**: ciò vuole dire che conserva i dati anche nel caso in cui non riceva alimentazione.

La non volatilità e l'accesso in sola lettura sono le due caratteristiche che distinguono la ROM dalla **RAM**: le celle di archiviazione che compongono quest'ultima, infatti, possono essere lette, scritte e cancellate un numero indefinito di volte e sono attive solo fino a quando la memoria è alimentata elettricamente.

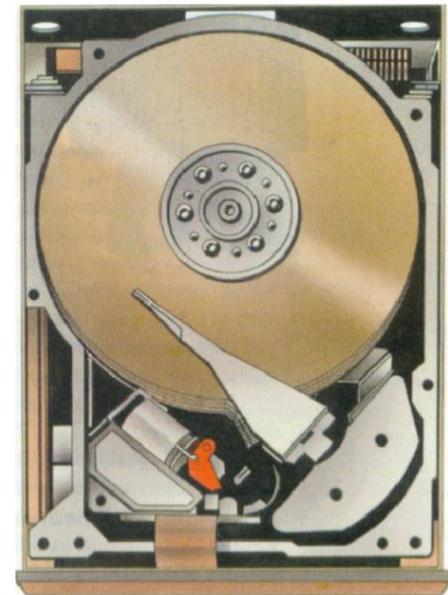


A causa della volatilità e della dimensione limitata della memoria principale, la maggior parte dei computer è fornita di dispositivi aggiuntivi chiamati sistemi di **memorie di massa**, HD, CD, DVD, nastri magnetici, memorie flash.

I vantaggi delle memorie di massa sono notevoli: minore volatilità, grandi capacità di memorizzazione, e possibilità di essere rimosse dalla macchina e archiviate separatamente

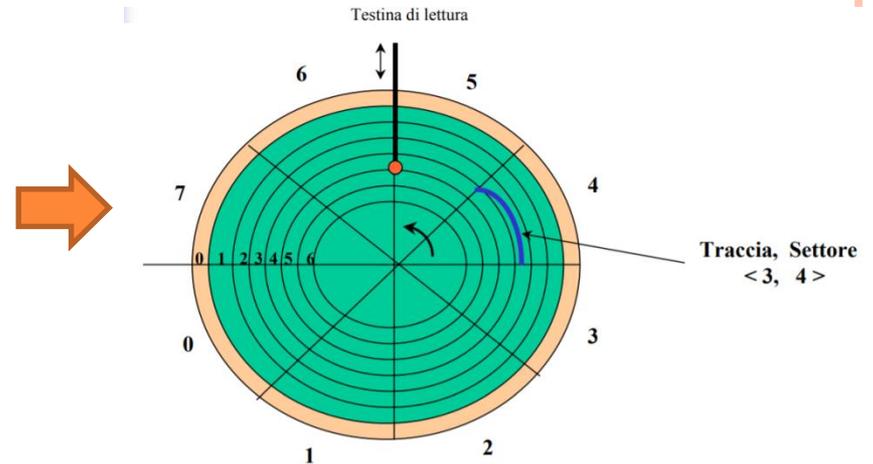
Sistemi Magnetici: disco rotante con una sottile pellicola magnetica.

Si sfrutta l'esistenza di sostanze che possono essere magnetizzate. La magnetizzazione può essere di due tipi (positiva e negativa). I due diversi tipi di magnetizzazione corrispondono ai due valori dell'unità fondamentali di informazione (bit)



Sono dei supporti di plastica, vinile o metallo, su cui è depositato del materiale magnetizzabile. Nel corso delle operazioni i dischi vengono mantenuti in rotazione a velocità costante e le informazioni vengono lette e scritte da testine

I dischi sono suddivisi in tracce concentriche e settori, ogni settore è una fetta di disco. I settori suddividono ogni traccia in porzioni di circonferenza dette blocchi (o record fisici)
Il blocco è dunque la minima unità indirizzabile in un disco magnetico e il suo indirizzo è dato da una coppia di numeri che rappresentano il numero della traccia e il numero del settore

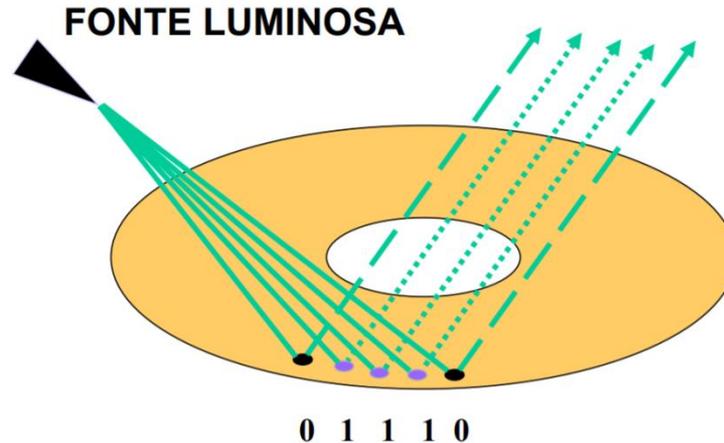


Per effettuare un'operazione di lettura (scrittura) su un blocco è necessario che la testina si posizioni in corrispondenza dell'indirizzo desiderato



Sistemi ottici (CD): Informazioni create in base a buchi creati su strato riflettente e recuperate tramite laser che ne verifica le irregolarità

Il raggio laser è un particolare tipo di raggio luminoso estremamente focalizzato che può essere emesso in fasci di dimensioni molto ridotte. La riflessione, o meno, del fascio luminoso può essere riconosciuta e corrisponde ai due valori dell'unità fondamentali di informazione (bit)



Una caratteristica comune ai sistemi sia ottici che magnetici è che per recuperare i dati è necessario un movimento fisico, come la rotazione dei dischi. Ciò significa riduzione del tempo di scrittura e lettura del dato stesso

Unità flash: I bit vengono memorizzati inviando segnali elettrici, dove gli elettroni vengono intrappolati in piccole celle di biossido di silicio, alterandone le caratteristiche fisiche. Nel tempo sono meno affidabili di dispositivi ottici.



Esempio 1 Operatori AND e OR su Excel



=E(C3>20;D3>20;E3>20)

	A	B	C	D	E	F	G	H
1		PRODOTTO	MAGAZZINO1	MAGAZZINO2	MAGAZZINO3		Ho almeno 20 pezzi in ogni magazzino	Ho almeno 20 pezzi in uno dei miei magazzini
2							E(AND)	O(OR)
3		A	25	24	1		FALSO	VERO
4		B	4	82	87		FALSO	VERO
5		C	46	85	13		FALSO	VERO
6		D	86	94	75		VERO	VERO
7		E	34	31	85		VERO	VERO
8		F	9	18	4		FALSO	FALSO
9		G	91	2	54		FALSO	VERO
10								
11								
12								

=SE(C4<20;"Ordina subito";"Giacenza")

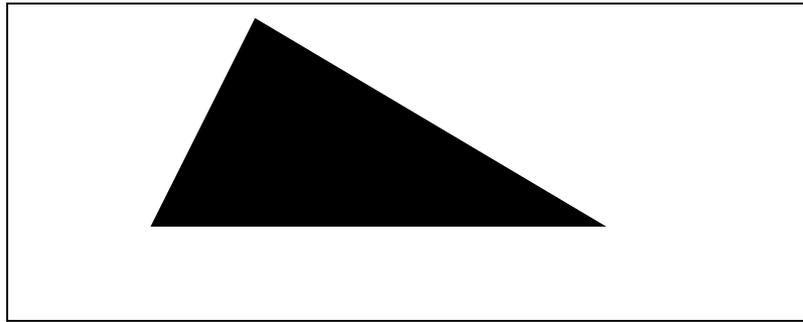
	A	B	C	D	E
1					
2					
3		Articolo	Quantità in magazzino	Riordinare?	
4		Penne blu		5	Ordina subito
5		Penne rosse		41	Giacenza
6		Penne nere		34	Giacenza
7		Righello		56	Giacenza
8		Gomme		22	Giacenza
9		Evidenziatore		6	Ordina subito
10		Quaderni		12	Ordina subito
11		Temperini		22	Giacenza
12					
13					
14					



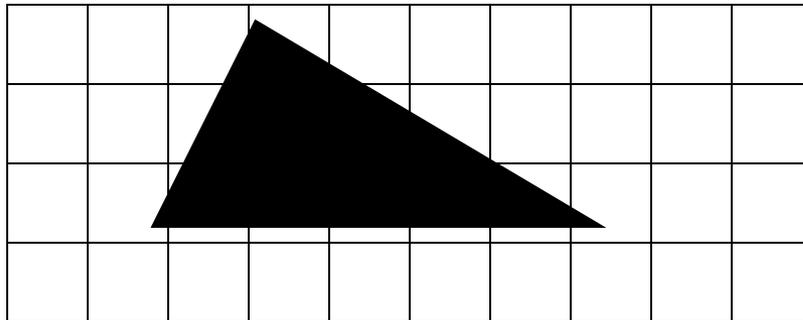
Utilizzo del SE condizionale su Excel



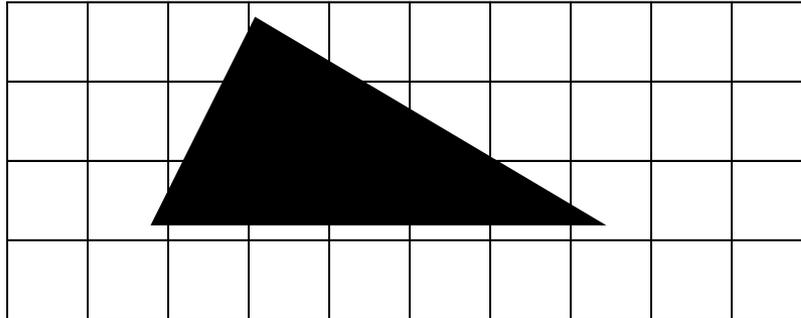
CODIFICA DELLE IMMAGINI



Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante



- Ogni quadratino derivante da tale suddivisione prende il nome di *pixel* (picture element) e può essere codificato in binario secondo la seguente convenzione:
 - Il simbolo “0” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il bianco è predominante
 - Il simbolo “1” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il nero è predominante



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



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

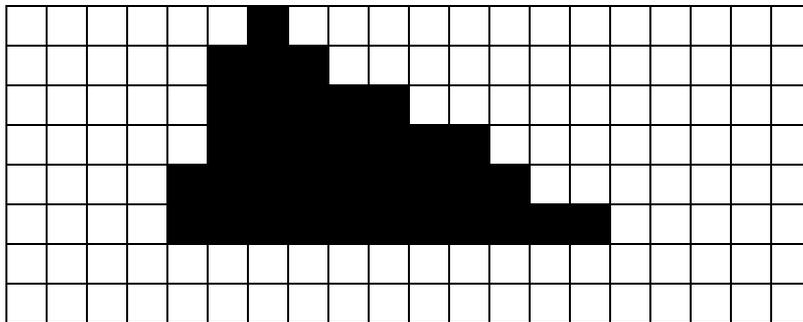
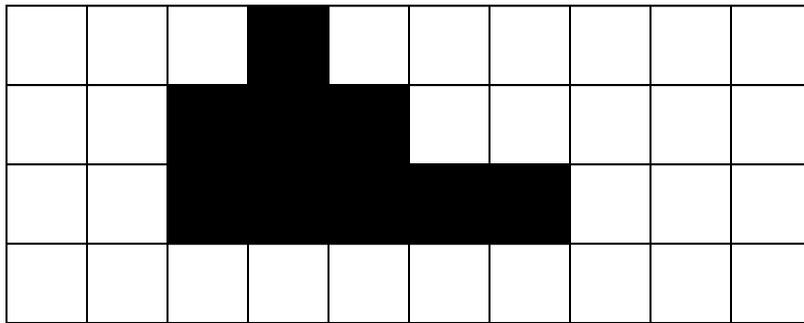
Poiché una sequenza di bit è lineare, è necessario definire convenzioni per ordinare la griglia dei pixel in una sequenza. Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra

0000000000 0011111000 0011100000 0001000000



Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria

Se riconvertiamo la sequenza di stringhe
000000000 0011111000 0011100000 0001000000
in immagine otteniamo



La rappresentazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine



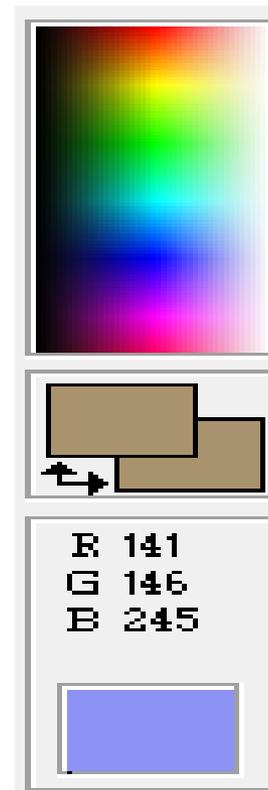
- Assegnando un bit ad ogni pixel è possibile codificare solo immagini in bianco e nero
- Per codificare le immagini con *diversi livelli di grigio* oppure *a colori* si usa la stessa tecnica: per ogni pixel viene assegnata una *sequenza* di bit
- Per memorizzare un pixel non è più sufficiente un solo bit
 - Per esempio, se utilizziamo quattro bit possiamo rappresentare $2^4 = 16$ livelli di grigio o 16 colori diversi
 - Mentre con otto bit ne possiamo distinguere $2^8 = 256$, ecc.



- Il colore può essere generato componendo 3 colori: red, green, blue (RGB)
- Ad ogni colore si associa una possibile sfumatura
- Usando 2 bit per ogni colore si possono ottenere 4 sfumature per il rosso, 4 per il blue e 4 per il verde che, combinate insieme, danno origine a 64 colori diversi
- Ogni pixel per essere memorizzato richiede 6 bit
- Usando 8 bit per ogni colore si possono ottenere:
 - 256 sfumature per il rosso ...
 - ... 256 per il verde e ...
 - ... 256 per il blu ...

che, combinate insieme, danno origine a circa 16,8 milioni di colori diversi (precisamente 16.777.216 colori)

- Ogni pixel per essere memorizzato richiede 3 byte



- Il *numero di pixel presenti sullo schermo* (colonne x righe) prende il nome di *risoluzione*

- Risoluzione tipiche sono

640 x 480 800 x 600 1024 x 768

1280 x 1024 1600 x 1200

- Esempio:

- Per distinguere 256 colori sono necessari otto bit per la codifica di ciascun pixel
- La codifica di un'immagine di 256 colori, formata da 640 x 480 pixel, richiederà $640 \times 480 = 307.200$ byte (2.457.600 bit)



PIXEL

da (“PICTureELement”) identifica una piccola porzione rettangolare dello schermo.

Rappresenta un’unità logica elementare di riferimento per la rappresentazione delle immagini digitali.

RISOLUZIONE

indica le dimensioni dell’immagine espresse in pixel (es. 640x480)



- Le immagini codificate *pixel per pixel* sono dette immagini in grafica ***bitmap***
 - Le immagini bitmap occupano parecchio spazio
- Esistono delle tecniche di *compressione* che permettono di *ridurre le dimensioni*
 - Ad esempio, se più punti vicini di un'immagine assumono lo stesso colore, si può memorizzare la codifica del colore una sola volta e poi ricordare per quante volte deve essere ripetuta



- I formati come GIF, JPEG e PNG sono formati compressi
 - Per esempio: rispetto al bitmap, il formato JPEG dedica meno bit alla descrizione delle sfumature cromatiche di un'immagine
- Postscript e PDF per i documenti



- Distinzione tra compressione *lossless* e compressione *lossy*
 - Lossless: compressione senza perdita di informazioni
 - Dalla versione compressa, si può ricostruire perfettamente la versione non-compressa
 - Per esempio, GIF (fino a 256 colori)
 - Lossy: compressione con perdita di informazioni
 - Dalla versione compressa, non è possibile recuperare la versione originale
 - Per esempio, JPEG, PNG

È chiaro che i dati non sono più direttamente disponibili ma occorre svolgere un passaggio intermedio (decompressione).



Finché si resta al di sotto di una ben determinata soglia, le perdite introdotte risultano essere pressoché nulle per l'utente medio

- Mano a mano che si aumenta il fattore di compressione, la codifica lascia passare solo frequenze spaziali basse dando luogo ad ampie chiazze monocolori
- Aumentando il fattore di compressione risulta anche evidente la sovrastruttura a blocchi applicata all'immagine



Immagine
.bmp

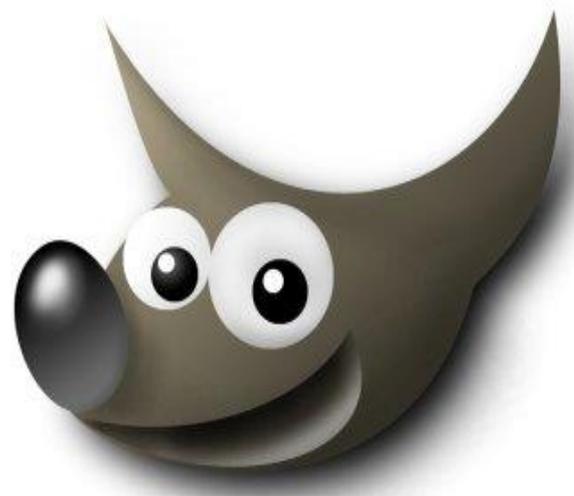
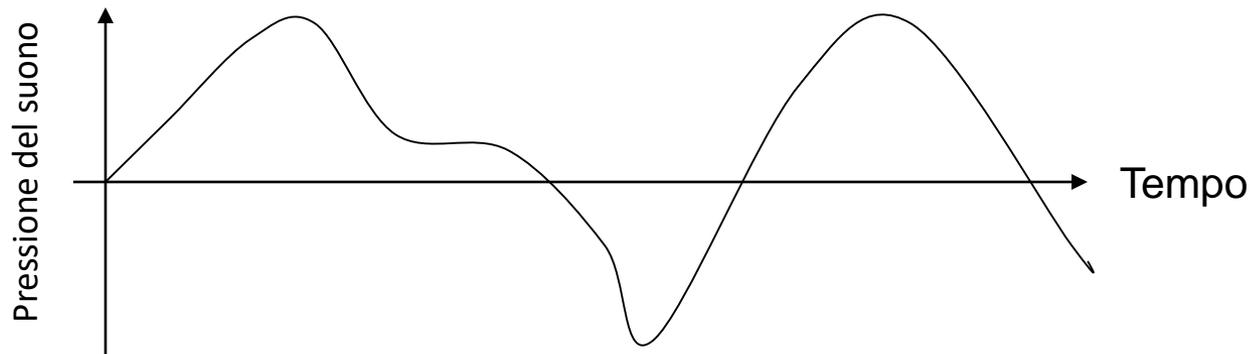


Immagine
.jpg

- Fisicamente un suono è rappresentato come un'onda che descrive la variazione della pressione dell'aria nel tempo (onda sonora)



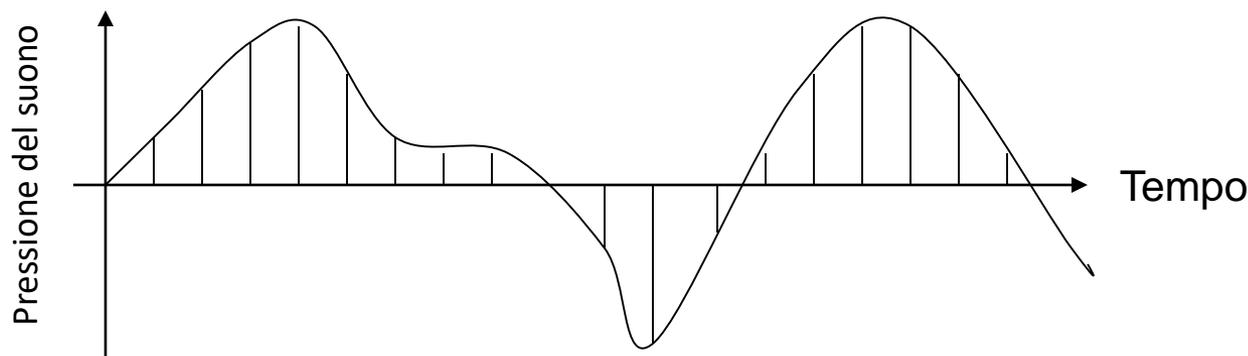
- Sull'asse delle ascisse viene rappresentato il tempo e sull'asse delle ordinate viene rappresentata la variazione di pressione corrispondente al suono stesso



- Un'idea chiave: il suono è continuo (la pressione varia senza salti)
- È possibile ottenere una registrazione *analogica* continua dell'onda
 - Una registrazione analogica memorizza ogni variazione dell'onda originale (almeno in linea di principio)
- La rappresentazione digitale di un suono funziona in un modo diverso



- Si effettuano dei ***campionamenti*** sull'onda (cioè si misura il valore dell'onda a intervalli di tempo costanti) e si codificano in forma digitale le informazioni estratte da tali campionamenti

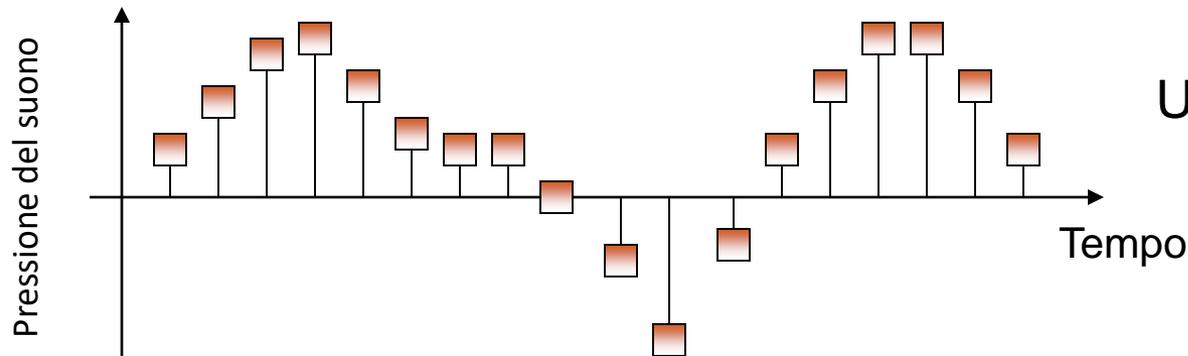


- Quanto più frequentemente il valore di intensità dell'onda viene campionato, tanto più precisa sarà la sua rappresentazione
- Il numero di campioni raccolti per ogni secondo definisce la frequenza di campionamento che si misura in Hertz (Hz, numero di campionamento ogni secondo)



CODIFICA DEI SUONI

- La sequenza dei valori numerici ottenuti dai campioni può essere facilmente codificata con sequenze di bit



Un'approssimazione!

- La rappresentazione è tanto più precisa quanto maggiore è il numero di bit utilizzati per codificare l'informazione estratta in fase di campionamento

