LA CODIFICA DELL'INFORMAZIONE

Tutta l'informazione in un calcolatore è rappresentata tramite numeri. Questi numeri sono espressi come sequenze di 0 e 1.
I calcolatori elaborano l'informazione effettuando operazioni su numeri.
I sistemi di comunicazione scambiano informazioni spostando sequenze di numeri.
I computer sono **sistemi digitali** perché lavorano con **digit** (**0** e **1**), e che sono dei numeri.
Bit : valore 0 (tensione bassa), valore 1 (tensione alta). **unità di misura dell'informazione.**
Byte: sequenza di 8 bit. Esempio: **01110010.** Byte unità di misura della capacità di memoria.
L'informazione viene creata ed acceduta in unità di informazione dette **word**.
Una word è creata come un multiplo di byte. Lunghezze di una **word**: 8 (byte), 16, 32, 64, 128 bit.
Un word a **k** bit può contenere 2^k valori (da 0 a 2^{k-1}).

LA CODIFICA CONSISTE NELL'ASSEGNARE AD OGNI CARATTERE O SIMBOLO UNA SEQUENZA UNIVOCA DI BIT, CHE PRENDE IL NOME DI CODICE.

Quanti oggetti, caratteri o simboli posso codificare con k bit:
- 1 bit ⇒ 2 stati (0, 1) ⇒ 2 oggetti (Vero/Falso)
- 2 bit ⇒ 4 stati (00, 01, 10, 11) ⇒ 4 oggetti
- 3 bit ⇒ 8 stati (000, 001, …, 111) ⇒ 8 oggetti
- **k bit ⇒ 2^k stati ⇒ 2^k oggetti**

Quanti bit mi servono per codificare N oggetti: $N \leq 2^k \Rightarrow k \geq \log_2(N)$

ESEMPIO DI CODIFICA BINARIA

1- Problema: assegnare un codice binario univoco a tutti i giorni della settimana

• Giorni della settimana: $N = 7 \Rightarrow k \geq \log_2 7 \Rightarrow k = 3$bit
• Con 3 bit possiamo ottenere 8 diverse.
configurazioni:
- Ne servono 7, quali utilizziamo?
- Quale configurazione associamo a quale giorno?

• Attenzione: ipotesi che i codici abbiano tutti la stessa lunghezza.

Giorno	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato	Domenica
Codice binario	000	001	010	011	100	101	110
Numero	0	1	2	3	4	5	6

2- Con quanti bit rappresento 256 numeri?
- $\log_2 256 = \log_2 2^8 = 8$ bit

Unità	Equivalenza		Unità	Equivalenza
bit	solo due stati, "0" oppure "1"		Byte	8 bit, quindi $2^8 = 256$ livelli
kiloByte [KB]	2^{10} Byte = 1024 Byte ~ 10^3 Byte		MegaByte [MB]	2^{20} Byte ~ 10^6 Byte
GigaByte [GB]	2^{30} Byte ~ 10^9 Byte		TeraByte [TB]	2^{40} Byte ~ 10^{12} Byte
PetaByte [PB]	2^{50} Byte ~ 10^{15} Byte		ExaByte [EB]	2^{60} Byte ~ 10^{18} Byte

CODICI A LUNGHEZZA FISSA

Se si usa un numero prestabilito di cifre si ha un codice a lunghezza fissa;
In questo modo si pone anche un limite al numero massimo rappresentabile.

In generale, con N cifre a disposizione e base b il più grande numero (intero positivo) rappresentabile si può esprimere come b^{N-1}
Esempio: qual è il numero più grande rappresentabile con 4 cifre?

base	Valore massimo	base	Valore massimo

10	9999	16	FFFF ($=65535_{10}$)
2	1111 ($=15_{10}$)	8	7777 ($=4095_{10}$)

La **codifica** dei numeri maggiori di quello **massimo** rappresentabile causano problemi di **overflow (trabocco);** Ovvero per essere rappresentati richiedono più cifre di quelle a disposizione.
Esempio: se si vuole sommare un numero al valore massimo dell'esempio precedente (con 4 cifre)

- In base 10: 9999 + 1 = 10000_{10}
- In base 2: 1111 + 1 = 10000_2 ($=16_{10}$)
- In base 16: FFFF + 1 = 10000_{16} ($=65536_{10}$)
- In base 8: 7777 + 1 = 10000_8 ($=4096_{10}$)

CODICE ASCII (American **S**tandard **C**ode for **I**nformation **I**nterchange: *Codice Standard Americano per lo Scambio di Informazioni*)**:**
Il calcolatore, nato per trattare numeri, col passare degli anni ha trovato impiego anche in quelle attività che devono trattare caratteri, frasi, parole, ecc.
A tale scopo, vengono create delle tabelle che fanno corrispondere ad ogni carattere una sequenza di bit.
La tabella del codice ASCII codifica 128 simboli o caratteri diversi utilizzando 7 bit .
I caratteri ASCII da 0 a 127:

- I primi 32 (numerati da 0 fino a 31) sono "caratteri di controllo" *non stampabili,*
- I successivi 95 simboli (numerati da 32 fino a 126) sono caratteri *stampabili,*
- Il 128-esimo simbolo è ancora un "carattere di controllo" *non stampabile.*

Dal momento che l'unità di memorizzazione dei dati in un calcolatore è il byte. Utilizzando la codifica ASCII estesa.

- Con 8 bit rappresento 256 caratteri (ASCII esteso)
- Si stanno diffondendo codici più estesi (UNICODE a 16 bit) per rappresentare anche i caratteri delle lingue orientali.

Esempio: "Computer" in ASCII diventa: ogni simbolo viene rappresentato tramite 8 bit.
C=67=01000011, o =111= 01101111, m =109=001101101, p=112=01110000, u=117=01110101, t=116=01110100, e=101=01100101, r=114=01110010.
Il risultato: 01000011- 01101111- 01101101- 11100000-01110101-01110100-01100101- 01110010

INSERIMENTO DI UN CARATTERE ASCII IN UN DOCUMENTO
Tenere premuto ALT mentre si digita l'equivalente numerico decimale.

Ad esempio, per inserire il simbolo dei gradi (º), digitare 0176 sul tastierino numerico tenendo premuto ALT.

Nota: Per digitare i numeri, è necessario utilizzare il tastierino numerico anziché la tastiera.

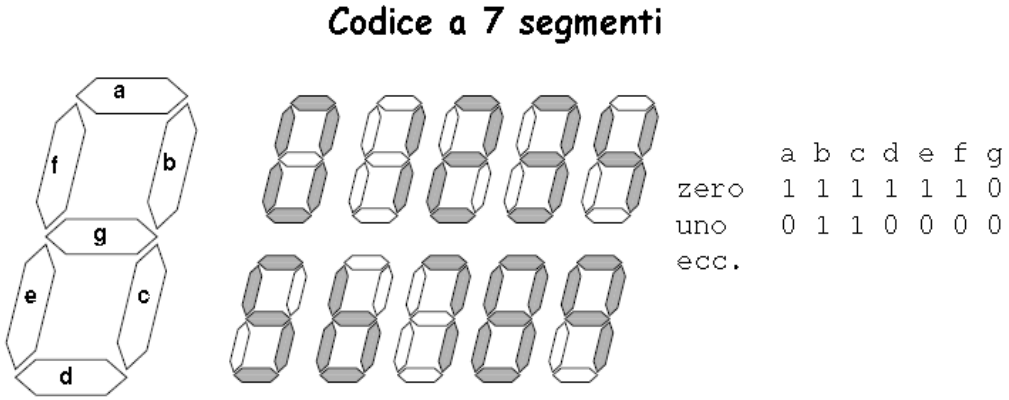
LA CODIFICA BCD (BINARY CODED DECIMAL, PRONUNCIA BAINERI CODED DESIMOL: CODIFICA BINARIA DEL CODICE DECIMALE)
L'elettronica degli elaboratori è **binaria**, mentre la mente umana è abituata a ragionare in **decimale**. Per mettere d'accordo i due mondi, sono nati i codici **BCD**.

il codice BCD permette la codifica, mediante quattro bit primari, delle dieci cifre decimali del nostro sistema di numerazione.

DECIMALE	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Essendo 10 i **simboli** da codificare (da **0 a 9**) sono necessari 4 bit che, come è noto, possono dar luogo a **16 combinazioni** che in binario puro corrispondono ai numeri naturali da **0 a 15**.
Poiché le **6 combinazioni dal 10 al 15** non si usano, la codifica BCD è ***ridondante***.

La codifica BCD del numero è, infatti, una operazione per la visualizzazione su un display numerico decimale a 7 segmenti.



I quattro bit di ciascuna cifra codificata BCD vengono inviati ad un circuito di decodifica che provvederà a pilotare il visualizzatore della cifra corrispondente (display a 7 segmenti).
Il codice BCD gode di un'interessante proprietà relativa alla somma:

- si sommano i quattro omologhi di quattro bit che costituiscono le singole cifre degli addendi.
- Si potranno quindi avere dei riporti, se il risultato dovesse superare i codici esprimibili con 4 bit, dovremmo aggiungere 6 = 0110 al risultato stesso. Se invece non vi è riporto ma il risultato della somma supera i quattro bit cade in una delle sei combinazioni che non hanno significato in codice BCD dovremmo, anche qui, aggiungere 6 = 0110 al risultato stesso.

In egual modo risolveremo delle sottrazioni.

- Esempi d'operazioni (somma e sottrazione)

SOMMA **$(520)_{10} + (150)_{10}$**

0101	0010	0000	520 +
0001	0101	0000	150 =
0110	0111	0000	670

$(526)_{10} + (159)_{10}$

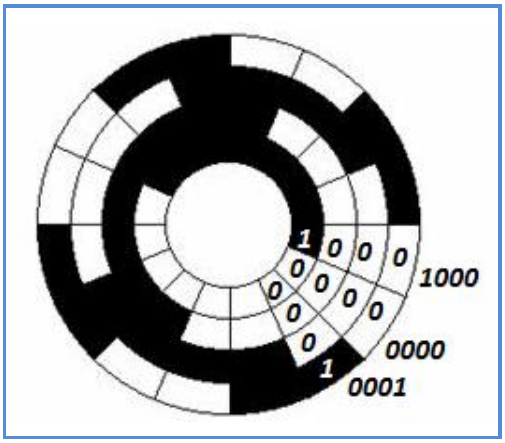
0101	0010	0110	526 +
0001	0101	1001	159 =
0110	0111	1111	685
		0110	
0110	1000	0101	

SOTTRAZIONE **$(526)_{10} - (159)_{10}$**

0101	0010	0110	526 –
0001	0101	1001	159 =
0110	0111	1111	367
		0110	
0110	1000	0101	

Codice Gray

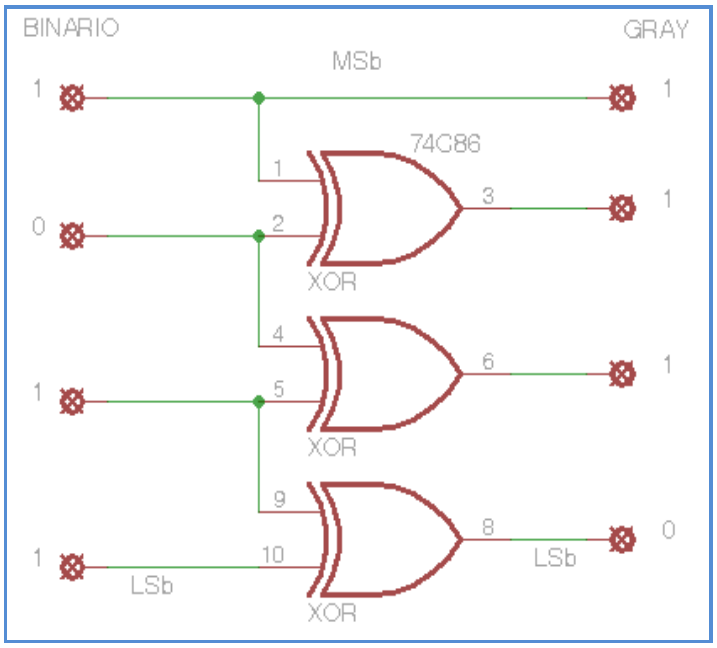
Il codice Gray non è un codice pesato, ovvero non c'è relazione tra il valore (peso) di un bit e la sua posizione nel numero.



Il codice Gray viene utilizzato nel controllo del movimento di ciascuna parte meccanica di un sistema di controllo, nei robot, come gli **encoder rotativi**, usati in sostituzione di potenziometri e commutatori in strumentazione, regolazione di volume in apparecchi audio, sintonia in ricevitori radio, ecc, codificano il valore digitale della posizione **chiudendo o aprendo due o più contatti elettrici**.

Nel **codice Gray** il passaggio da un numero al successivo avviene sempre variando un'unica cifra binaria. le tracce di codice vengono lette trasversalmente rispetto alla direzione di movimento, evitando così errori di codifica causati dalla variazione di più bit tra settori contigui.

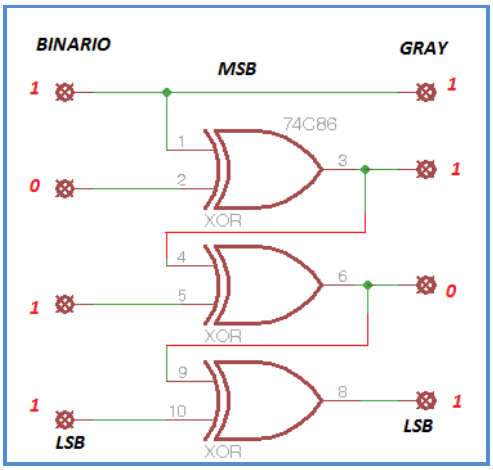
Per **convertire un numero binario in codice Gray** è molto semplice: si utilizza la funzione **XOR**, è un comparatore digitale ed è definibile come una somma in modulo 2.



Codice binario a 4 bit	Codice Gray a 4 bit
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

Da Gray a binario

Il procedimento di conversione da codice di Gray a codifica binaria normale è molto simile a quello appena descritto ma con qualche piccola differenza.



CODICE DI PARITÀ

Nella trasmissione delle informazioni con codici a due bit (0, 1) si possono verificare degli errori, provocati dalla presenza di rumore sulla linea di trasmissione.

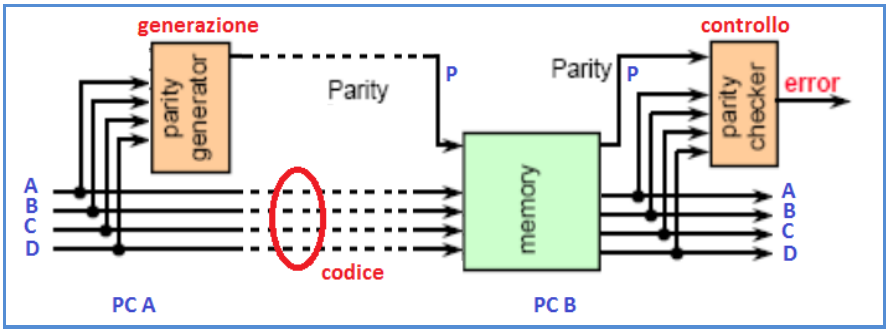
Una delle soluzioni del problema è la realizzazione di un circuito di controllo per la rilevazione di errori di trasmissione, definito come generatore/rivelatore di **parità pari** (**parity even: peti iven**) oppure **dispari** (**parity odd: peti aad**).

- Per i codici a parità dispari: il numero di 1 della codeword è dispari (10110)
- Per i codici a parità pari: il numero di 1 della codeword è pari (10010)

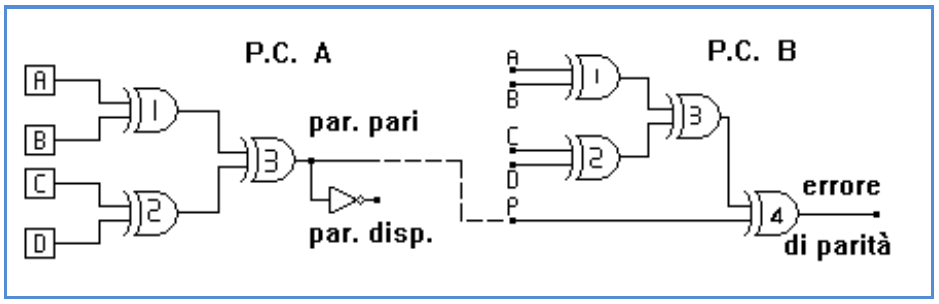
Nel caso di un rivelatore di **parità pari** su un codice a cinque bit, di cui quattro rappresentano la codifica esadecimale ed il quinto la parità. Il bit di parità ottenuto da un generatore, viene confrontato con quello ricevuto tramite una porta EXOR.

- Se i bit sono uguali, l'uscita del rilevatore è "0" e la parola o il codice ricevuto viene accettato, cioè non è errato.
- Se sono diversi l'uscita è "1". Codice errato.

Come successivo approfondimento sul problema della rilevazione degli errori, vediamo una rete logica relativa al controllo di parità su un codice a quattro bit che è trasmesso dal computer PC A al PC B come indicato in figura.



In questo caso sono possibili due tipi di controllo, quello di **parità pari** (**parity even**), e quello di **parità dispari** (**parity odd**), il codice completo quindi, nei due casi, invierà parole di cinque simboli. È possibile con questa tecnica correggere errori di molteplicità dispari, infatti tale codice non produce nessuna segnalazione se nella stessa parola sono presenti un numero pari di bit errati. Il circuito che genera il bit di parità, che sarà inviato sulla linea tratteggiata, richiede l'immagazzinamento temporaneo della parola, mentre per la generazione e l'invio del bit di parità può essere utilizzato un circuito combinatorio con porte logiche EX-OR (7486) come indicato in figura.



Infatti le porte logiche (1) e (2) eseguiranno separatamente i controlli sui bit A e B, C e D ed a loro volta questi risultati parziali saranno nuovamente confrontati tramite la porta (3). L'uscita della porta logica (3) segnala lo stato logico uno se fra i bit della parola saranno presenti un numero dispari di uno e viceversa segnala lo stato logico zero nel caso in cui il numero di uno sia pari. È sufficiente l'aggiunta di un invertitore per ottenere il controllo d tipo dispari.

Qualora sia richiesto un controllo di parità su una parola formata da più bit, è sufficiente aumentare il numero degli EX-OR nel circuito.

Il bit di parità P così generato e trasmesso dal PC A con la parola; è ricevuto dal PC B ed utilizzato per il controllo di errore. Infatti sarà sufficiente in ricezione un circuito analogo per controllare i quattro bit ricevuti e qualora l'uscita della porta (3) coincida con il bit di parità inviato dal trasmettitore, la parola potrà essere accettata come valida.

LA CODIFICA DELL'IMMAGINE

La codifica dell'immagine fissa mediante una serie di bit prevede, la suddivisione del quadro stesso in un reticolo "griglia" a elementi rettangolari o quadretti detti "pixel" (**pict**ure **el**ement).
Il pixel è il più piccolo elemento che può essere disegnato sullo schermo e prende abitualmente il nome punto "dot", ha tuttavia una dimensione spaziale ben definita (0,26 mm). Ogni pixel dell'immagine viene codificato usando gruppi di 0 e 1.

COS'È LA RISOLUZIONE?

Un'immagine digitale è formata da pixel, quadrati piccolissimi dove ciascuno "porta" con se una parte delle informazioni relative all'immagine acquisita.
Maggiore sarà il loro numero (pixel), e maggiore sarà la quantità di informazioni nell'immagine.
la risoluzione indica la densità dei punti elementari (*dot*) che restituiscono l'immagine, più ce n'è, più l'immagine è risoluta.

Se immaginiamo un rettangolo composto da tanti quadratini, moltiplicando il numero dei pixel dei due lati che formano le dimensioni otterremo la risoluzione complessiva dell'immagine.

Ad esempio:
n. 2048x1536 = 3.145.728 di pixel
n. 3008x2000 = 6.016.000 di pixel
n. 3264x2448 = 7.990.272 di pixel

La **risoluzione** dell'immagine è il numero di pixel che la costituiscono, espressi in termine di **larghezza** x **altezza**. Aumentando il numero di pixel a disposizione, migliora la qualità dell'immagine.
Esempio: una immagine di risoluzione 205 x 154 pixel a 24 bit = 3 byte, il formato in bitmap, pari a =205*154*3=94710 byte.

CONVERSIONE DA PIXEL IN CM:

$$\text{Dimensione in centimetri} = \frac{(\text{Dimensione in pixel})}{(\text{risoluzione in ppi o dpi})} * 2{,}54$$

dpi (dot per inch = punti per pollice), ppi (**pixel** per inch = **pixel** per pollice)
Mi potreste dire a quanti pixel corrispondono 5,9 mm e 8,6 mm?

5,9 mm = 22,23 pixel
8,6 mm = 32,5 pixel

Conosci il rapporto tra Pixel e DPI? Questa domanda è un vero classico della computer grafica! Insomma **pixel e dpi** sono due cose sconnesse fra loro?

- **I pixel sono semplici punti**
- **I dpi (dippiai) sono la densità di punti su una determinata superficie.**

Formato	Misure in cm	DPI	PIXEL
A4	29.7 * 21	300	3508 * 2480
A3	42 * 29.7	250	4134 * 2923

1 inch (pollice) = 25,4 [mm] 1cm è equivalente a 0.39370 pollici.

CONVERSIONE DA INCH IN CM: **cm = in / 0.39370**

x=larghezza dell'immagine (in pixel)
y= altezza dell'immagine (in pixel)
w= larghezza dell'immagine in pollici (una volta stampata)
h= altezza dell'immagine in pollici (una volta stampata)
Allora:
x= w * risoluzione.
y= h * risoluzione.

La codifica RGB (Reed, Green, Blu ovvero i tre colori primari), ogni pixel viene rappresentato con una combinazione dei tre colori.
Per ogni colore primario si usa un certo numero di bit per rappresentare la gradazione (la qualità), usando 8 bit per colore primario otteniamo: 256x256x256 = 16777216 colori diversi, in questo caso un pixel richiede tre byte di informazione.

Grafica bitmap (bmp)
Le immagini codificati pixel per pixel sono dette bitmap (mappa di bit), per ciascun pixel dell'immagine una opportuna sequenza di bit.
Il formato bitmap non prevede alcuna forma di compressione e di conseguenza le dimensioni del file risultano elevate.
Se **l'immagine** è solo in **bianco** e **nero**, basterà usare "**1**" per **pixel neri** e **"0"** per i **pixel bianchi**.
Se l'immagine ha più di due colori, si faranno corrispondere a gruppi diversi di "0" e "1", sfumature diverse di colori (o di grigio). Se si fa corrispondere a ogni pixel 1 byte (8 bit), potremo differenziare 2^8=256 colori.

GIF (Graphics Interchange Format): Nel formato GIF il numero massimo dei colori è di 256, ma tra i punti di forza di questo formato vi sono la possibilità di creare immagini animate.
JPEG (**J**oint **P**hotographic **E**xperts **G**roup: un comitato che ha definito il primo standard internazionale di compressione per immagini) **numero massimo di colori è di 256.**
GIF e JPEG sono largamente utilizzati su internet che adottano metodi di compressione dell'informazione prima di memorizzarle e che tendono ad eliminare i pixel ripetitivi. Sono soluzioni con compressione che porta a perdita di informazione, senza recupero in alcun modo.
La stampa di tali immagini è di pessima qualità.

PNG (Portable Network Graphic), Non ha limiti di colori memorizzati. Dal PNG deriva il formato **MNG** (Multiple-image Network Graphics) analogo alla GIF animata.

LA CODIFICA VIDEO
Un filmato è una sequenza di immagini statiche dette (fotogrammi o frame), un fotogramma è un insieme di pixel.
Per codificare un filmato si digitalizzano i fotogrammi, seguendo due possibili impostazioni

- Codificare in PCM
- Codificare in formato bitmap i fotogrammi, eseguendo poi una compressione JPEG.

Esempio:
un formato bitmap di un segnale video composto da 25 fotogrammi al secondo, ipotizzando una dimensione dell'immagine pari a 700 x 525 pixel, con 16 bit per pixel ha una dimensione pari a 700 x 525 x 16 x 25 = 147Mbit , in JPEG con un fattore di compressione del 10% corrisponde a 14,7 Mbit.
L'effetto della compressione elimina l'effetto della ridondanza (ripetitzione), aumentando la velocità di trasmissione.

LA CODIFICA DEL SUONO

Il procedimento della digitalizzazione del suono, avviene mediante un **campionamento del suono analogico** (cioè la lettura del valore del suono analogico a intervalli di tempo regolari), secondo una frequenza di campionamento detta di **Shannon** (che stabilisce il valore minimo della frequenza di campionamento che permette di riprodurre fedelmente il segnale utile sia il doppio dalla frequenza del segnale da convertire), la tecnica che viene utilizzata è il **PCM**.
le normali schede presenti nel PC permettono campionamenti da 8000 a 48000 Hz e ogni campione può essere codificato da 8, 16 o 32 bit, l'ultimo valore fornisce la massima fedeltà.

ESERCIZI DA SVOLGERE

1- Quanti bit si devono utilizzare per rappresentare 300 informazioni distinte?
2- Quanti byte occupa la parola "psicologia" se la si codifica utilizzando il codice ASCII?
3- Dati 12 bit per la codifica, quante informazioni distinte si possono rappresentare?
4- Quanto spazio occupa un suono della durata di 30 secondi campionato a 100 Hz (100 campioni al secondo), in cui ogni campione occupa 4 byte?
5- Codificare il numero 175_{10} nella corrispondente rappresentazione binaria.
6- Ordinare in modo crescente i seguente numeri: 105_{10} , 12_8 , 100110000_2 , 10011_{10}.
7- Codificare il numero negativo -15_{10} nella rappresentazione in complemento a due.
8- Quanti bit occupa un'immagine di 100 x 100 pixel in bianco e nero?
9- Quanti byte occupa un'immagine di 100 x 100 pixel a 256 colori?
10- Se un'immagine a 16777216 di colori occupa 2400 byte, da quanti pixel sarà composta?

11- determinare il numero di pixel presenti in 1 cm^2 di un'immagine scansionata con risoluzione di 100x100dpi. Se l'immagine è grande 5x2 cm quanto occupa in byte la relativa bitmap a 256 colori?

La risoluzione indica usualmente in dpi (dot per inch: punti per pollice – si ricorda che un pollice è uguale a 2,54 cm)
I livelli che corrispondono ai colori della bitmap sono L = 256 = 2^n; n = 8bit per pixel
la dimensione dell'immagine è = 5x2 = 10 cm^2
il numero di dpi = 100x100 (100 punti per pollice in orizzontale che in verticale) = 10.000 dpi
1 pollice = 2,54 cm, allora si ha 100 punti in un 2,54cm.
In un solo cm^2 = $(100/2,54)^2$ = $(39,37)^2$ = 1550 pixel
Il numero di pixel in 10cm^2 = 1550 x 10 = 15500 pixel
Il numero di bit = 15500 x 8 = 124000 bit = 15500 byte = 15,5 kbyte

12- un'immagine grande 12x18 cm deve essere scansionata e salvata come bitmap a 24 bit. Quale deve essere la risoluzione massima affinché l'immagine non occupa più di 1 Mbyte?
R = 150 dpi 1 dpi = 2,54 cm

13- Dati 6 bit per la codifica, quante informazioni distinte si possono rappresentare?
2^6=64 informazioni distinti

14- Dato un byte per la codifica, quante informazioni distinte si possono rappresentare?
Un byte = 8 bit, 2^8=256 informazioni distinti

15- Quanti bit si devono utilizzare per rappresentare 20 informazioni distinte?
Almeno 5 bit, perché 2^5=32 => 20 (4 bit non sono sufficiente, perché 2^4=16 < 20)

16- Quanti byte occupa la frase "cervello" scritta in ASCII? 8 (in ASCII, un carattere corrisponde a un byte)
17- Quanti byte occupa la frase "dipartimento di psicologia" scritta in ASCII? 26

18- Quanti byte occupa un suono della durata di 5 secondi campionato a 30 Hz (30 campioni per secondo), in cui ogni campione occupa 6 byte?
5 x 30 x 6 = 900 byte

19- Un secondo di suono campionato a 512 Hz occupa 1 KB. Quanti valori distinti possono avere i campioni?
1 KB = 1024 byte; numero di campioni = 1 x 512 = 512; ogni campione contiene 1024/512 = 2 byte; 2 byte = 16 bit; 2^{16} valori distinti

20- Un'immagine a 256 colori è formata da 400x400 pixel. Quanto spazio occupa?
Ogni pixel richiede un byte (=8 bit, perché 2^8=256, sufficiente per rappresentare 256 colori); l'immagine ha 400 x 400 = 160000 pixel; l'immagine occupa 160000 byte (= 1280000 bit)

21- Hai ricevuto un messaggio di posta elettronica da un amico. Il messaggio contiene:
   a. un testo di 300 caratteri scritto in ASCII,
   b. un'immagine di 120x150 pixel con 1024 colori.
   c. Quanti byte occupa il messaggio?
      Testo: 300 byte. Immagine: ogni pixel richiede 10 bit (perché 2^{10}=1024); l'immagine ha 120 x 150 = 18000 pixel; l'immagine occupa 10 x 18000 = 180000 bit = 22500 byte. Testo + immagine: 300 + 22500 = 22800 byte

22- Un'immagine di 300x400 pixel occupa 15000 byte. L'immagine è a colori oppure in bianco e nero?
L'immagine ha 300 x 400 = 120000 pixel, ed occupa 15000 x 8 = 120000 bit. Cioè ad ogni pixel corrisponde a un bit, e l'immagine è in bianco e nero

23- Quanto spazio occupa un'immagine animata di 100x100 pixel a 128 colori, formata da 6 frame?
Ogni frame ha 100 x 100 = 10000 pixel; ogni pixel richiede 7 bit (perché 2^7=128); ogni frame occupa 10000 x 7 = 70000 bit; l'immagine animata occupa 70000 x 6 = 420000 bit (= 52500 byte)