

# CODIFICA NUMERI INTERI

[1]

## ○ RAPPRESENTAZIONE DEI NUMERI NEGATIVI

- ↔ Modulo e segno
- ↔ Complemento a 1
- ↔ Complemento a 2

## ○ RANGE ED OVERFLOW

- ↔ Range  $-2^{n-1}$   $2^{n-1}-1$   
 $2^{16} \rightarrow [-32768 \quad 32767]$   
 $2^{32} \rightarrow [-2147483648 \quad 2147483647]$

### ↔ Overflow:

errore dovuto alla necessità di rappresentare il risultato di un'operazione con più cifre di quelle a disposizione

# CODIFICA NUMERI INTERI

[2]

## ○ MODULO E SEGNO

- ↔ 1 bit di segno e  $n-1$  per il modulo

Es:  $N=6$  0|110

$N=-6$  1|110

- ↔ Ho problema per doppia rappresentazione dello 0

Es: 0|000 1|000

- ↔ Scorretta la somma bit a bit

Es: 1 0|001 +

-1 1|001 =

-2 1|010

- ↔ Operazioni complesse (Necessita sottrattore)

# CODIFICA NUMERI INTERI

[3]

## ○ COMPLEMENTO A 1

↵ Per il numero negativo complemento (vale ancora segno)

Es: N=6 0110

N=-6 1001

↵ Somma corretta a meno del riporto

Es: N=25 011001 + N=25 011001 +

N=3 000011 = N=-3 111100 =

28 011100 21 1|010101 +

000001 =

N=22 010110

↵ Ho problema per doppia rappresentazione dello 0

↵ Complessità per il riporto

# CODIFICA NUMERI INTERI

[4]

## ○ COMPLEMENTO A 2

↵ Complemento bit a bit e sommo 1 (e viceversa)

Es: N=6 0110

N=-6 1001 +

0001 =

1010

↵ (Complemento dopo il primo 1 a partire da sinistra)

Es: N=6 0110

← ↓ ↓

1010

↵ Ho unica rappresentazione per lo 0

↵ Non ho problemi di riporto

↵ Tecniche per individuare Overflow

# CODIFICA NUMERI REALI

[1]

## ○ NUMERI IN VIRGOLA MOBILE [FLOATING POINT]

↗ Estendo il campo dei valori rappresentabili con errore di quantizzazione percentuale costante

↗ Con numeri decimali

$$N = \pm M \cdot 10^E \quad \text{Es: } 354.14 = 3.5414 \cdot 10^2$$
$$0.0001 = 1 \cdot 10^{-4}$$
$$1.1367 = 1136.7 \cdot 10^{-3}$$

↗ Con numeri binari

$$N = \pm M \cdot 2^E \quad \text{mi basta M ed E}$$

↗ M e' un numero frazionario in base 2

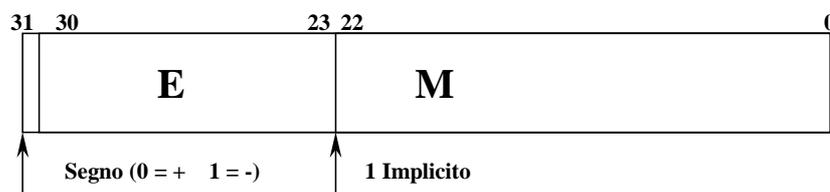
↗ M compresa tra 1/2 e 1 -> forma normalizzata

# CODIFICA NUMERI REALI

[2]

## ○ STANDARD IEEE (INSTITUTE ELECTRICAL AND ELECTRONICS ENGINEERS)

↗ Uso 32 bit con seguente suddivisione



↗ Il valore numerico viene calcolato come:

$$N = (-1)^S \cdot 2^E \cdot (1, M)$$

↗ E=00000000 -> N=0

↗ E=11111111 -> Non valido

↗ Valori approssimati tra:

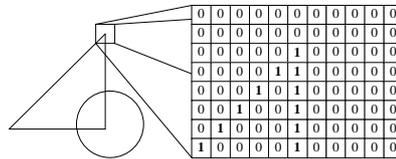
$$\pm 1.17 \cdot 10^{-38} \quad \pm 3.40 \cdot 10^{38}$$

# CODIFICA IMMAGINI

[1]

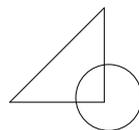
## ○ BITMAP

- ↗ Scomposizione (discretizzazione) dell'immagine in elementi di informazione poi codificati (Scanner, Videocamere, Sintesi)



## ○ VETTORIALI

- ↗ Descrizione dell'immagine attraverso elementi grafici di alto livello (Progettazione meccanica, elettronica architettonica, ...)



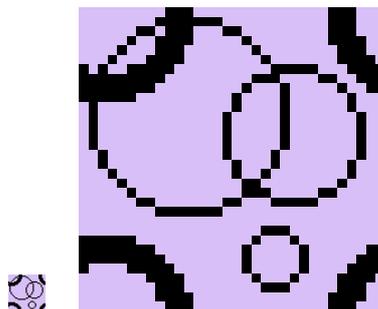
Circle 98 66 50  
Polyline 0 48 88 152 88 48 0 48

# CODIFICA IMMAGINI (BITMAP)

[2]

## ○ BITMAP

- ↗ Matrice di punti detti **pixel** (Picture Element)
- ↗ Qualità caratterizzata dalla risoluzione (Numero di pixel per unità di misura)
- ↗ Elevato spazio occupato in memoria per avere buone risoluzioni
- ↗ Non è possibile un'eccessivo ZOOM perché non si notano più dettagli, si notano solo i pixel



# CODIFICA IMMAGINI (BITMAP)

[3]

- CODIFICA DEI PIXEL (PROFONDITÀ COLORE)
  - ↗ 1 bit Immagine b/n
  - ↗ 4 bit Immagine 16 Colori o Livelli Grigio (VGA)
  - ↗ 8 bit Immagine 256 Colori o Livelli Grigio (SVGA)
  - ↗ 16 bit Immagine 64K Colori
  - ↗ 24 bit Immagine 16M Colori (True Color)
  - ↗ 32 bit Immagine 16M Colori + Alpha Channel
- COMPOSIZIONE COLORI (COLORIMETRIA)
  - ↗ Sintesi Sottrattiva: dal bianco tolgo le lunghezze d'onda dei colori primari (CMY+K -> Stampanti)
  - ↗ Sintesi Additiva: sommo al nero le lunghezze d'onda dei colori primari (RGB -> Televisori)
- CODIFICO OGNI CANALE (8 BIT)

# CODIFICA IMMAGINI (BITMAP)

[4]

- CODIFICA CON PALETTE (COLOR LOOK UP TABLE)
  - ↗ Codifico solo colori dominanti dell'immagine
  - ↗ Uso solo alcuni dei 16M ma con stessa qualità
  - ↗ Creo una palette con 256 (o più) colori RGB
  - ↗ Ogni singolo pixel punta ad un elemento della CLUT
  - ↗ Ottengo profondità di 256, 32K, 64K anche con 8 bit
- DIMENSIONE IMMAGINE
  - ↗  $N_{\text{bit}} = \# \text{Pixel}_x \cdot \# \text{Pixel}_y \cdot \# \text{bit/Pixel}$

Tipo immagine	Risoluzione	Colori	Dimensione
Televisiva	720 x 625	256	440 KB
SVGA (PC)	1024 x 768	64K	1.5 MB
Fotografica	15000 x 10000	16M	430 MB

# CODIFICA IMMAGINI (BITMAP)

[5]

## ○ FORMATI DI MEMORIZZAZIONE

- ↔ RAW: matrice dell'insieme di punti
- ↔ TIFF(Tagged Image File Format): etichette che descrivono proprietà seguite dai dati (24 bit, LZW, ...)
- ↔ GIF(Graphic Interchange Format): brevetto compuserve, possibile immagazzinare più immagini (animate,LZW)
- ↔ JFIF(JPEG File Interchange Format): implementa JPEG, presenza pixel trasparenti e interlacciamento
- ↔ BMP(BitMaP): sviluppato da Microsoft e IBM per windows ed OS/2 (supporta RLE)
- ↔ PCX(PC Paintbrush): formato supportato da numerose applicazioni Microsoft (supporta RLE)
- ↔ PICT: formato di interscambio standard in ambiente Macintosh.

# CODIFICA IMMAGINI (VETTORIALI)

[6]

## ○ CODIFICO ELEMENTI DI ALTO LIVELLO

- ↔ Figure geometriche semplici parametriche
- ↔ Rappresentazione compatta dell'informazione
- ↔ Indipendenza dal dispositivo di visualizzazione e dalla sua risoluzione
- ↔ Partono da modello quindi è possibile avere un qualsiasi fattore di ZOOM
- ↔ Poco adatto a foto e immagini naturali, usato per la descrizione ad alto livello di informazione grafica

## ○ CONVERSIONE BITMAP VETTORIALI

- ↔ Facile da vettoriali a bitmap (rendering)
- ↔ Complessa l'estrazione di elementi di alto livello dalle bitmap (soprattutto per immagini naturali)

# CODIFICA IMMAGINI (VETTORIALI)

[7]

## ○ FORMATI DI MEMORIZZAZIONE

- ↔ PostScript: formato brevettato da Adobe per la rappresentazione dei documenti (linguaggio per stampanti)
- ↔ EPS (Encapsuled PostScript): usato per trasferire disegni tra applicazioni PostScript
- ↔ PDF (Portable Document File): usato per la descrizione di documenti e immagini (include ps e navigazione ipertestuale)
- ↔ DXF (Drawing Exchange Format): usato per il progetto meccanico (proprietario Autodesk)
- ↔ IGES (Initial Graphics Exchange Specifications): dati geometrici e topologici realizzati con programmi 3D

## ○ WINDOWS METAFILE (WMF)

- ↔ Bitmap + vettoriali
- ↔ Tipico per lo scambio di immagini sotto Windows

# CODIFICA AUDIO

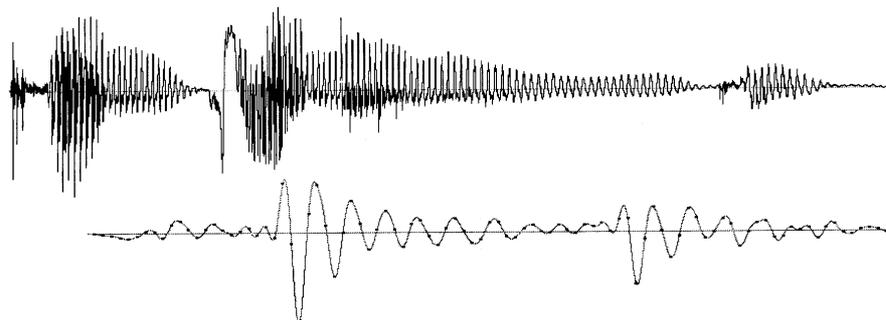
[1]

## ○ SEGNALE ANALOGICO

- ↔ Continuità nel tempo
- ↔ Continuità nelle ampiezze

## ○ SEGNALE DIGITALE

- ↔ Discretizzazione nel tempo (campionamento)
- ↔ Discretizzazione nelle ampiezze (quantizzazione)



# CODIFICA AUDIO

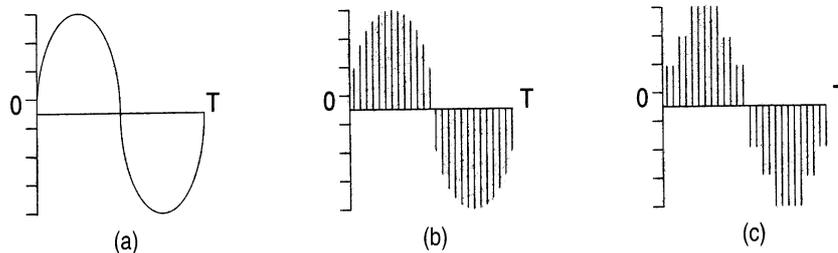
[2]

## ○ CAMPIONAMENTO

- ↗ Considero i valori del segnale in determinati istanti
- ↗ Con segnale a banda limitata posso ricostruire interamente l'originale ( $f_c \geq 2B$ )

## ○ QUANTIZZAZIONE

- ↗ Rappresento il campione con numero di bit finito
- ↗ Introduco perdita informazione (non reversibile)
- ↗ Introduco rumore di quantizzazione



# CODIFICA AUDIO

[3]

## ○ MEMORIZZAZIONE

- ↗  $N_{\text{bit}} = \text{Durata} \cdot f_c \cdot \text{bit/campione}$

## ○ PLAYBACK

- ↗ Richiede un certo bitrate (bit / sec.)

## ○ ESEMPI:

	<b>Telefonia</b>	<b>CD Audio</b>
Segnale	Voce	Musica
Canali	1 (mono)	2 (stereo)
Banda	300 Hz - 3.4 kHz	20 Hz - 20 kHz
$f_c$	8 kHz	44.1 kHz
Bit/campione	8	16
Bitrate	64 kbps	1.4 Mbps

# CODIFICA AUDIO

[4]

## ○ FORMATI DI MEMORIZZAZIONE

- ↔ Wav (Wave)
- ↔ Mod
- ↔ PCM (Pulse Code Modulation)
- ↔ RA (Real Audio)
- ↔ AIFF (Audio Interchange File Format)
- ↔ PASC (Philips Precision Sub-band Adaptive Coding)
- ↔ MP3

## ○ FORMATO MIDI (MUSICAL INSTRUMENT DIGITAL INTERAFACE)

- ↔ Protocollo per la trasmissione della musica
- ↔ Non codifica il suono, ma l'azione di uno strumento
- ↔ 30 KB/min contro 10MB/min (1:333)

# CODIFICA VIDEO

[1]

## ○ TECNICA DI CODIFICA

- ↔ Campione nel tempo (24, 25, 30 fps)
- ↔ L'occhio umano ricostruisce la continuità

## ○ DIMENSIONI

- ↔ 1 Minuto 25 fps -> 644 MB

## ○ FORMATI DI MEMORIZZAZIONE

- ↔ AVI: Video for Windows
- ↔ MOV: Quicktime
- ↔ MPG: Codifica MPEG
- ↔ ...

# COMPRESSIONE DATI

[1]

- SENZA PERDITA

- ↗ RLE (Run Length Encoding): sequenze di byte
- ↗ Huffman: codifica variabile su frequenza statistica
- ↗ LZW (Lempel Ziv Welch): tabelle di simboli

- CON PERDITA

- ↗ JPEG (Joint Photographer Expert Group)
- ↗ MPEG (Moving JPEG), MP3

