Rappresentazione binaria e dati

Loriano Storchi

loriano@storchi.org

http::://www.storchi.org/

BREVE DIGRESSIONE

Rappresentazione binaria dei numeri (breve digressione)

- In un sistema di **numerazione posizionale** data la **base** questa definisce direttamente il numero di **simboli (cifre)** che si usano per scrivere il numero.
  - Ad esempio nel sistema decimale usiamo 10 simboli (0,1,2,3,4,5,6,7,8,9)
- I moderni sistemi di numerazione sono posizionali, **quindi il numero e' scritto specificando l'ordine delle cifre, ed ogni cifra assume un valore a seconda della sua posizione**
  - Ad esempio $423 = 4 * 10^2 + 2 * 10^1 + 3 * 10^0$

Se volete 4 centinaia 2 decine e 3 unita'

Rappresentazione binaria dei numeri (breve digressione)

- In generale data una **base b avro' b simboli** (cifre) e quindi **un numero intero N** sara' scritto come:
  - Valore N = $c_n * b^n + c_{n-1} * b^{n-1} + \ldots + c_0 * b^0$
- Similmente se ho un numero N = $0.c_1c_2...c_n$
  - Valore N = $c_1 * b^{-1} + c_2 * b^{-2} + \ldots + c_n * b^{-n}$
- Consideriamo adesso il caso piu' semplice quello delle rappresentazione di **numeri interi senza segno (i Naturali)**
- Se uso un sistema di numerazione con **base b con n cifre** potro' rappresentare un **massimo di b^n numeri diversi**, quindi tutti i numeri da 0 fino a b^n - 1
- Ad esempio in **base 10** e' chiaro che usando **due cifre posso rappresentare tutti i numeri da 0 a 99** quindi **10^2 = 100** numeri distinti

La base binaria

- Usando una base 2, quindi solamente due simboli 0 ed 1, sempre rimanendo nell'ambito della rappresentazione di numeri interi positivi usando n cifre potro' rappresentare al massimo tutti i numeri compresi fra **0 e 2^n -1**
  - Ad esempio usando due cifre potro' rappresentare 4 numeri distinti:
  - $00 = 0 * 2^1 + 0 * 2^0 = 0$
  - $01 = 0 * 2^1 + 1 * 2^0 = 1$
  - $10 = 1 * 2^1 + 0 * 2^0 = 2$
  - $11 = 1 * 2^1 + 1 * 2^0 = 3$

DIGITALIZZAZIONE

La base binaria

- Un byte (un boccone) rappresenta modernamente la sequenza di 8 bit ed e' divenuto storicamente l'elemento base dell'indirizzabilita' e quindi l'unita' di misura base dell'informazione.
- **8 bit significa che con 1 byte posso rappresentare 2^8 = 256 valori differenti**. Quindi nel caso di numeri interi unsigned i numeri da 0 a 255. **Se uso un bit per indicare il segno ad esempio 0 positivo ed 1 negativo,** posso rappresentare i numeri interi da -128 a 127 (da 10000000 a 01111111)
- Oppure con 8 bit posso rappresentare 256 differenti caratteri.

ASCII

- Extended ASCII usa 8-bit
- ASCII originale US-ASCII 7-bit

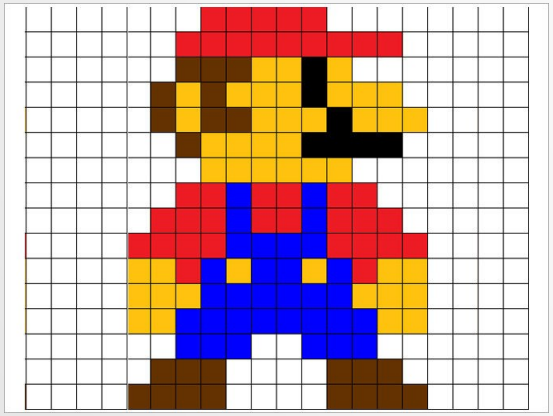
000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ã	231	þ
001	Start Of Header (SOH)	034	"	067	C	100	d	133	à	166	ª	199	Ã	232	Þ
002	Start Of Text (STX)	035	#	068	D	101	e	134	å	167	º	200	╚	233	Ú
003	End Of Text (ETX)	036	$	069	E	102	f	135	ç	168	¿	201	╔	234	Û
004	End Of Transmission (EOT)	037	%	070	F	103	g	136	ê	169	®	202	╩	235	Ù
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	╦	236	ý
006	Acknowledge (ACK)	039		072	H	105	i	138	è	171	½	204	╠	237	Ý
007	Bell	040	(	073	I	106	j	139	ï	172	¼	205	═	238	¯
008	Backspace (BS)	041	)	074	J	107	k	140	î	173	¡	206	╬	239	´
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	¤	240	-
010	Line Feed **(LF)**	043	+	076	L	109	m	142	Ä	175	»	208	ð	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Å	176	░	209	Ð	242	‗
012	Form Feed **(FF)**	045	-	078	N	111	o	144	É	177	▒	210	Ê	243	¾
013	Carriage Return **(CR)**	046	.	079	O	112	p	145	æ	178	▓	211	Ë	244	¶
014	Shift Out	047	/	080	P	113	q	146	Æ	179	│	212	È	245	§
015	Shift In	048	0	081	Q	114	r	147	ô	180	┤	213	ı	246	÷
016	Dataline Escape (DLE)	049	1	082	R	115	s	148	ö	181	Á	214	Í	247	¸
017	DC 1 (XON)	050	2	083	S	116	t	149	ò	182	Â	215	Î	248	°
018	DC 2	051	3	084	T	117	u	150	û	183	À	216	Ï	249	¨
019	DC 3 (XOFF)	052	4	085	U	118	v	151	ù	184	©	217	┘	250	·
020	DC 4	053	5	086	V	119	w	152	ÿ	185	╣	218	┌	251	¹
021	Negative Acknowledge (NAK)	054	6	087	W	120	x	153	Ö	186	║	219	█	252	³
022	Synchronous Idle	055	7	088	X	121	y	154	Ü	187	╗	220	▄	253	²
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	╝	221	¦	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	¢	222	Ì	255	
025	End Of Medium	058	:	091	[	124	|	157	Ø	190	¥	223	▀		
026	Substitude	059	;	092	\	125	}	158	×	191	┐	224	Ó		
027	Escape **(ESC)**	060	<	093	]	126	~	159	ƒ	192	└	225	ß		
028	File Seperator	061	=	094	^	127 (DEL)	⌂	160	á	193	┴	226	Ô		
029	Group Seperator	062	>	095	_	128	Ç	161	í	194	┬	227	Ò		
030	Record Seperator	063	?	096	`	129	ü	162	ó	195	├	228	õ		
031	Unit Seperator	064	@	097	a	130	é	163	ú	196	─	229	Õ		
032	*SPACE* (SP)	065	A	098	b	131	â	164	ñ	197	┼	230	µ		

Codifica dell’informazione

- Una codifica e’ una convenzione, come visto prima quindi la sequenza di bit **01001100 puo’ rappresentare il carattere L (L maiuscolo) oppure se invece intendiamo un valore intero unsigned rappresenta il numero decimale 76**.
- Ad esempio **ogni immagine e’ composta da pixel, se usassi 1 solo bit per ogni pixel potrei avere solo immagini in bianco e nero (1 pixel nero , 0 pixel bianco).**

Codifica dell’informazione

- se uso piu’ bit per rappresentare ogni pixel posso invece avere range di grigi o colori. E da qui suoni, video ...



Il pixel rappresenta il piu’ piccolo elemento autonomo dell’immagine. Ogni pixel e’ dunque caratterizzato, dalla propria posizione

Il numero totale di pixel di un’immagine digitale e’ detto **risoluzione** . Ad esempio se ho una griglia 10 x 10 l’immagine sarà composta dal 100 pixel

dpi = numero di punti per pollice , ad esempio in un monitor tipicamente avro’ 72 pixel per pollice

Profondita’ : nel caso di un’immagine in scala di grigi s possono usare 8 bit per ogni pixel avendo cosi’ a disposizione **2^8 = 256 sfumatiure di grigio**

Codifica dell’informazione

- Immagini a colori

Nel caso di immagini a colori **ogni pixel** e’ caratterizzato da **tre scale di colori** per i tre colori fondamentali, **RGB (Red, Green, Blue)**

Ad esempio un’immagine ad **8-bit** avra’ per ogni colore 256 possibili sfumature per il rosso, 256 per il verde e 256 per il blu. Dunque in totale **16777216** possibili sfumature di colore. Nel caso di immagini a 12-bit avremo invece $\mathbf{2^{12} = 4096}$ **sfumature per ogni colore**, in totale **68718476736** possibili colori per ognii pixel.

CENNI; OVERFLOW, UNDERFLOW, TIPI DATO

Le memoria dei calcolatori e’ finita

- **OVERFLOW**: i bit a disposizione non bastano per rappresentare il risultato. **UNDERFLOW** numero troppo piccolo 3/2 non posso rappresentare 1.5 , posso rappresentare solo 1 o 2
- Ad esempio se uso 8 bit per rappresentare i numeri interi positivi posso rappresentare solo i numeri da 0 a 255, quindi cosa succede se provo a sommare 1 a 255 ?

1 1 1 1 1 1 1 1 +

0 0 0 0 0 0 0 1 =

---

1 0 0 0 0 0 0 0 0 per rappresentare il risultato non sono sufficienti 8 bit

Tipi di dato

- Vedi differenza fra linguaggi fortemente tipizzati e non, tipizzazione dinamica e statica
- I linguaggi di programmazione hanno dei tipi di dato nativi, come ad esempio gli integer, i floating-point ,i boolean e i character. Diversi tipi diverse dimensioni e dunque diversi range (Algebra esatta e non …)

label	size (bytes)	smallest value	largest value
byte	1	-128	127
short	2	-32768	32767
int	4	-2147483648	2147483647
long	8	-9223372036854775808	9223372036854775807
char	2	0	65535

CENNI CONVERSIONE ANALOGICO DIGITALE

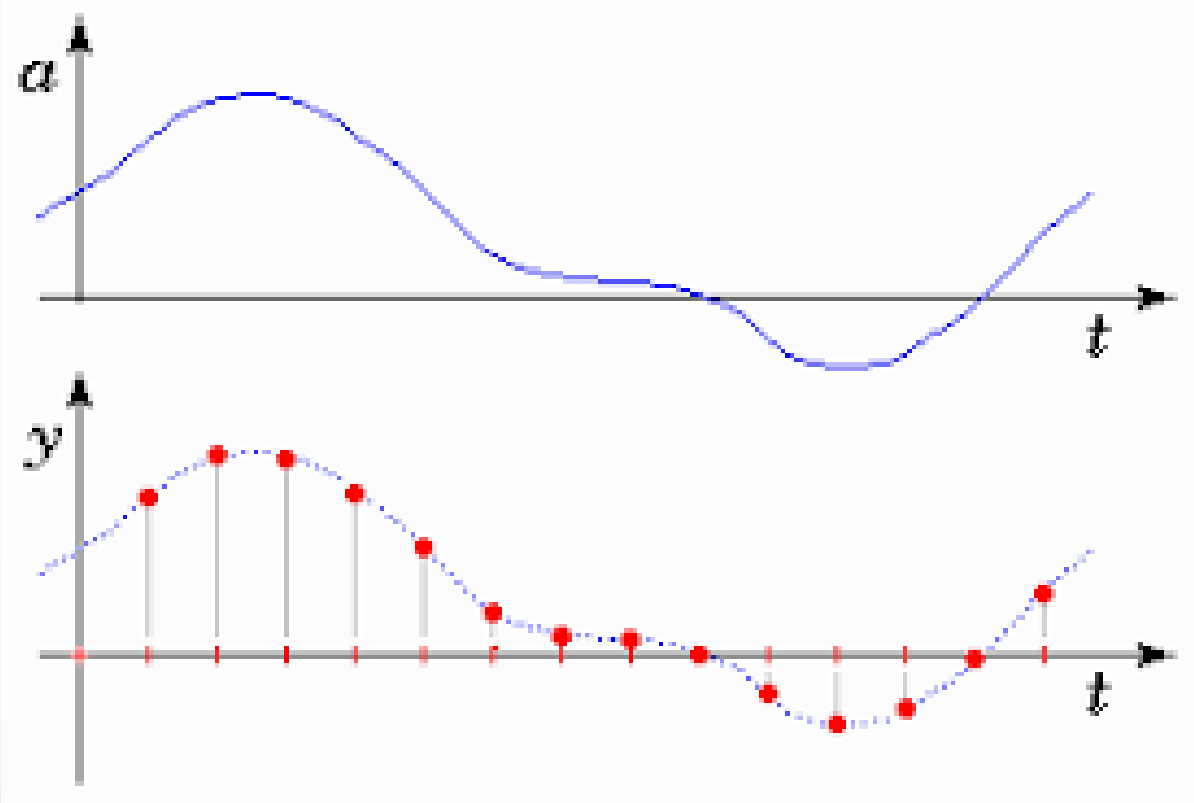
Segnale analogico

- Conversione Analogico Digitale

campionamento - discretizzazione del tempo (teorema del campionamento di Nyquist-Shannon)

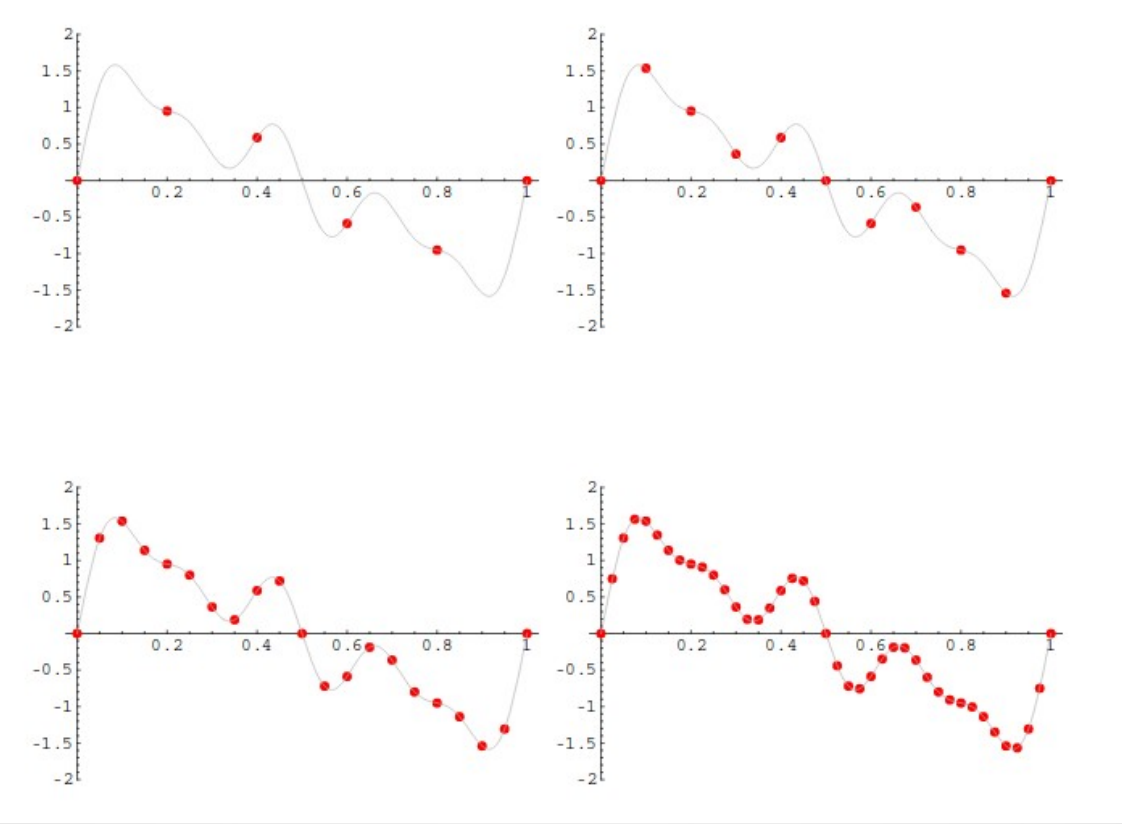
quantizzazione - discretizzazione della ampiezza

codifica - uso di “parole” binarie per esprimere il valore del segnale



Segnale analogico

- Ovviamente la fedelta’ migliora man mano che cresce il numero di campioni per unita’ di tempo (frequenza di campionamento) ed il numero di livelli di quantizzazione



Audio digitale

- Visto che l'orecchio umano riesce a udire frequenze in un certo range (20, 20000 Hz circa) il teorema del campionamento ci dice che dobbiamo campionare a 40 kHz
- Solitamente ssi usa un numero di livelli di quantizzazione molto piu' grande di 256, spesso 16 bit
- Ad esempio nel caso di un **CD** abbiamo due canali (stereo) a **44.1 kHz e 16 bit (2^{16} = 65536)**
  - Quindi il **bit rate = 44100 campioni/s * 16 bit * 2 canali = 1411200 bit/s = 176400 Byte/s**
  - Se vogliamo **1 minuto** di musica abbiamo bisogno di **60*176400 Byte/s = 10584000 Bytes** che sono circa 10 MiB

Considerazioni finali

- Compressione di immagini e suono (quindi anche video) si basa essenzialmente sull'eliminazione di informazioni a cui l'orecchio umano e l'occhio umano sono poco sensibili
- Ma non solo ….