**Introduzione all’architettura dei calcolatori**

Qualunque informazione numerica può essere codificata e trasformata attraverso un codice appropriato.

**Cosa si fa per “scrivere numeri su un pezzo di carta”?**   
Scrivendo un numero io riconosco il “codice” che viene usato, per esempio: nella scritta ‘105’, io leggo un numero a base decimale. Questo lo interpreto in tale maniera perché si sta usando un determinato codice a me noto. Stesso procedimento se voglio rappresentare unità, decine ecc., numeri decimali, usando un numero di cifre decimali adeguato, evitando gli zeri inutili.  
Scrivere 002 è uguale a 2, si usa la seconda notazione perché si cerca di economizzare sull’uso del codice, a sua volta a *lunghezza variabile*, in cui sulla base di una convenzione si minimizza il numero di cifre utilizzate.

**Nei sistemi di calcolo**  
Quando passiamo a un sistema di calcolo, usiamo dei codici a *lunghezza fissa*: il numero di cifre è prefissato. Ad esempio, in un codice a quattro cifre prefissate, se volessi rappresentare il numero 2, scrivo ‘0002’.

Il cambiamento di codice di lunghezza da variabile a fissa è dettato da vincoli (fisici) che nascono quando si vuole codificare l’informazione. Ad esempio, un foglio A4 su cui scrivo ha dei limiti fisici, tanto quanti ne ha la penna. Questi limiti fisici sono limiti “hardware” nei calcolatori. Il complesso hardware è uguale alla quantità di risorse che possiamo impiegare al massimo nell’elaborazione. Questi limiti sono imposti dall’esterno quando si passa da una sfera ideale a una pratica, fisica e concreta.

**Rappresentazione binaria**Altra cosa è la rappresentazione decimale: ogni posizione ha un significato, e un numero è formato da diverse cifre significative, ognuna è potenza di un certo numero che vogliamo rappresentare. In relazione alla base che utilizziamo (esempio base 16) cambiano i coefficienti per posizione: la prima cifra è 16^0, la seconda 16^1 e così via a crescere.   
In base al codice posso affermare anche che i numeri proseguono nell’ordine 0,1,2,3,4,5,6,7,8,9, A,B,C… dove A vale 11, B 12, C 13 e così via.  
Nella base 16 posso scrivere quindi un numero “01 A5” in cui ogni cifra viene moltiplicata per la potenza in relazione alla base scelta (es. 16^0 \* 11\* 5) e sommata con le cifre delle altre posizioni.

Stessa procedura vale per la base binaria (sequenza di zero e uno): in un numero a 16 bit, posso rappresentare tutti i numeri interi fra 0 e 2^(15), quest’ultimo numero ottenibile ponendo tutti 1 nelle 16 posizioni. Ricorda, le posizioni partono convenzionalmente da 2^0 e si termina con 2^(16-1).

**Il codice binario nella pratica quotidiana**Altra faccenda è come rappresentare i colori stampati o visualizzati. La procedura da impiegare è costruire degli elenchi: se associo ad ogni lettera dell’alfabeto un numero, potendo arrivare a codificare i caratteri stampabili, posso costruire un codice per colore.   
Col codice ASCII, con 7-bit posso rappresentare diversi elementi, fra cui numeri e caratteri. Dunque da qualsiasi informazione codificata è possibile giungere a una sorgente numerica del codice.

Stessa cosa vale per la codifica di audio e video, che possono essere codificati sotto una forma numerica.Per la registrazione di suoni si attua il campionamento (sampling), che trasforma la vibrazione sonora in una elettromagnetica, tale output viene “osservato” (analizzato) in certi istanti predefiniti (campioni) e qui viene misurata una differenza di tipo elettrico (es. il potenziale), che viene registrata. Tali segnali, posti su una sequenza, è possibile leggerli per convertire il suono da analogico a digitale e riprodurlo a sua volta attraverso un dispositivo audio.

Per la visualizzazione a video si hanno dei pixel, su cui viene rappresentata un’immagine adattando un’approssimazione per ogni pixel, che contiene un singolo colore. L’immagine inizialmente viene suddivisa nei 3 colori primari: Rosso, Verde (Green), Blu (RGB). Tale grandezza elettrica viene eventualmente trasformata e campionata a sua volta in un numero. La rappresentazione RGB consiste nell’usare una rappresentazione a 8 bit per il Rosso, 8 bit per il verde e 8 bit per il Blu. Combinando questi colori (numeri) è possibile ottenere altrettanti diversi colori: a ogni pixel viene assegnata una terna di numeri, per un totale di 24bit per ogni pixel.

**Introduzione al calcolo di numeri binari**

E’ possibile a sua volta codificare un numero con segno, con un numero senza alcun segno: una cifra binaria viene usata per rappresentare l’esponente della base binaria, un bit viene riservato al segno (0 = +, 1 = -), mentre la restante sequenza di numeri viene lasciata al coefficiente (detto modulo o mantissa).  
Con 8 bit posso rappresentare tutti i numeri da 0 a 2^(8-1) ovvero 127. Posso rappresentare i numeri col + e col – da 0 a 127. La rappresentazione non è canonica con modulo e segno, ovvero non ci sono più modi per rappresentare uno stesso valore.

Se voglio rappresentare un numero positivo, pongo 0 nella cella tutto a sinistra del segno, per poi rappresentare nelle celle (posizioni) della mantissa il numero. Per rappresentare un numero negativo inverto tutti i bit da 1 a 0 e viceversa, potendo costruire i numeri da 0 a -127. Il valore -128 posso rappresentarlo ponendo tutti 1, che è un vantaggio della rappresentazione in modulo e segno rispetto a valore e segno.