RAPPRESENTAZIONE DELLE INFORMAZIONI

Il bit

I calcolatori elettronici utilizzano come unità d'informazione di base il cosiddetto *bit* (*bi*nary digi*t* , cioè *cifra binaria*). Da un punto di vista prettamente fisico il *bit* è un sistema a 2 stati: può infatti essere indotto in uno dei due stati distinti rappresentanti 2 valori logici - no o si, falso o vero, o semplicemente 0 o 1. In termini pratici, senza scendere nei dettagli implementativi, il *bit* viene realizzato utilizzando le proprietà dell'energia elettrica (assenza di carica o presenza di carica). Un *bit* di informazione può ovviamente venire rappresentato anche attraverso altri mezzi: ad esempio con 2 differenti polarizzazioni di luce o 2 differenti stati elettronici di un atomo.

Rappresentazione binaria dell'informazione

Con un unico *bit* possono dunque essere rappresentate 2 differenti informazioni, ad esempio del tipo:

si/no, on/off, 0/1

Tuttavia, mettendo insieme più *bit* è possibile rappresentare un numero, anche molto elevato, di informazioni.
Attraverso 2 *bit*, per esempio, possono essere rappresentate 4 differenti informazioni:

00, 01, 10, 11

con 3 *bit* è possibile rappresentare 8 differenti informazioni:

000, 001, 010, 011, 100, 101, 110, 111

con 4 *bit* è possibile rappresentare 16 differenti informazioni:

0000, 0001, 0010, …, 1110, 1111

e così via.
In generale con *n bit* è possibile rappresentare 2^n differenti informazioni. Negli esempi precedenti, infatti, con 2 *bit* sono state rappresentate 2^2=4 informazioni, con 3 bit 2^3=8 informazioni e con 4 bit 2^4=16 informazioni differenti. Gli attuali personal computer operano su sequenze di ben 32 *bit*. Questo vuol dire che sono in grado di processare blocchi di informazione ognuno dei quali può codificare ben 2^{32}= 4'294'967'295 informazioni differenti.
Viceversa, per rappresentare *m* differenti informazioni occorrono *n bit*, tali che $2^n \geq m$. Ad esempio, per rappresentare 57 informazioni diverse sono necessari almeno 6 *bit*. In base alla formula precedente

$$2^6 = 64 > 57$$

Infatti, le possibili combinazioni di 6 *bit* sono 64:

000000, 000001, 000010, …, 111110, 111111

Si noti che 5 *bit* non sarebbero stati sufficienti, essendo $2^5 = 32 < 57$. Con 5 *bit* è cioè possibile rappresentare al più 32 differenti informazioni.

Il byte

In informatica ha assunto particolare importanza il concetto di *byte*. Un *byte* è l'equivalente di 8 *bit*:

$$1\ byte = 8\ bit$$

Pertanto, con un *byte* è possibile rappresentare $2^8 = 256$ differenti informazioni. Il *byte* è utilizzato come unità di misura per indicare le dimensioni della memoria, la velocità di trasmissione, la potenza di un elaboratore. Usando sequenze di *byte* (e quindi di *bit*) si possono rappresentare caratteri, numeri immagini, suoni.

Sistemi numerici

Il sistema numerico decimale

Il sistema decimale è quello utilizzato comunemente per la rappresentazione dei numeri. Esso è basato su 10 differenti cifre, dalla cifra 0, alla cifra 9, ed è di tipo posizionale.
Il termine posizionale deriva dal fatto che, a seconda della posizione che una cifra occupa nella rappresentazione di un numero, essa è caratterizzata da un peso. Ad esempio, si consideri il numero 1524; la posizione delle cifre obbedisce al seguente schema:

1	5	2	4
↑	↑	↑	↑
posizione 3	posizione 2	posizione 1	posizione 0

La cifra 4, nella posizione 0, è quella meno significativa poiché rappresenta le unità; la cifra 2, nella posizione 1, rappresenta le decine; la cifra 5, nella posizione 2, rappresenta le centinaia; la cifra 1, nella posizione 3, rappresenta le migliaia. In altri termini, il numero 1524 rappresenta 1 migliaia, 5 centinaia, 2 decine e 4 unità. Risulta chiaro che le cifre più significative sono quelle nelle posizioni più alte (a sinistra), mentre quelle meno significative sono quelle nelle posizioni più basse (a destra).
Grazie alla caratteristica posizionale, un numero decimale può essere espresso come somma di potenze di 10, le quali rappresentano i pesi delle posizioni, secondo il seguente schema:

Posizione	Peso	Potenza di 10
0	Unità	$10^0=1$
1	Decine	$10^1=10$
2	Centinaia	$10^2=100$
3	Migliaia	$10^3=1000$
4	Decine di migliaia	$10^4=10000$
…	…	…

In tal modo, il precedente numero, 1524, può essere espresso nel seguente modo:

$$1\cdot10^3 + 5\cdot10^2 + 2\cdot10^1 + 4\cdot10^0 = 1000 + 500 + 20 + 4 = 1524$$

Si noti che il numero più grande che è possibile rappresentare con *n* cifre in notazione decimale è:

$$10^n-1$$

Infatti, il numero più grande rappresentabile con 2 cifre è $99=10^2-1=100-1$. Allo stesso modo, il numero più grande rappresentabile con 4 cifre è $9999=10^4-1=10000-1$. E così via.

Il sistema numerico binario

Come il sistema numerico decimale, anche il sistema binario, basato sulle cifre 0 e 1, è di tipo posizionale (cioè, a ogni cifra è associato un peso in base alla sua posizione).
Le posizioni sono equivalenti a quelle della rappresentazione decimale. Se si considera il numero binario 10100101, si ha:

1	0	1	0	0	1	0	1
↑	↑	↑	↑	↑	↑	↑	↑
pos. 7	pos. 6	pos. 5	pos. 4	pos. 3	pos. 2	pos. 1	pos. 0

Il peso relativo alla posizione è definito di seguito:

Posizione (peso)	Potenza di 2
0	$2^0=1$
1	$2^1=2$
2	$2^2=4$
3	$2^3=8$
4	$2^4=16$
…	…
8	$2^8=256$
...	…

Conversione binario → decimale

Basta moltiplicare ogni bit per il suo peso e sommare il tutto. Ad esempio, il numero decimale corrispondente al numero binario 10100101 può essere espresso nel seguente modo:

$$1\cdot2^7+0\cdot2^6+1\cdot2^5+0\cdot2^4+0\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0=128+0+32+0+0+4+0+1=165$$

Esercizio

Convertire in decimale i seguenti numeri binari:

1) 10
2) 10001
3) 1001010101

Soluzione

1) $10_2=1\cdot2^1+0\cdot2^0=2_{10}$
2) $10001_2=1\cdot2^4+0\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0=17_{10}$
3) $1001010101_2=1\cdot2^9+0\cdot2^8+0\cdot2^7+1\cdot2^6+0\cdot2^5+1\cdot2^4+0\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0=597_{10}$

Si noti che il numero più grande che è possibile rappresentare con *n* cifre in notazione binaria è:

$$2^n-1$$

Ad esempio, il numero più grande rappresentabile con 2 cifre binarie è 11. Infatti:

$$11_2=1\cdot2^1+1\cdot2^0=3_{10}=2^2-1$$

Allo stesso modo, il numero più grande rappresentabile con 4 cifre è 1111. Infatti:

$$1111_2=1\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0=15=2^4-1$$

E così via. Si noti infine che per rappresentare il numero 16 in notazione binaria è necessario adottare almeno 5 *bit*. Infatti:

$$10000=16_{10}$$

Con 1 *byte* (8 *bit*) è, pertanto, possibile rappresentare i numeri decimali da 0 a 255:

$$00000000_2=0_{10},\ 00000001_2=1_{10},\ \ldots,\ 11111111_2=255_{10},$$

Fissate quante cifre (*bit*) sono usate per rappresentare i numeri, si fissa dunque anche il numero più grande che si può rappresentare:

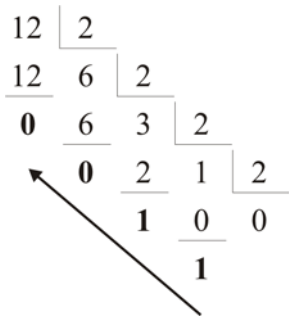
- con 16 bit è possibile rappresentare $2^{16} - 1 = 65.535$ numeri
- con 32 bit è possibile rappresentare $2^{32} - 1 = 4'294'967'295$ numeri
- con 64 bit è possibile rappresentare $2^{64} - 1 \approx 1.84 \cdot 10^{19}$ numeri

In realtà, come vedremo in una lezione successiva, è possibile anche rappresentare numeri più grandi, a patto di tollerare un certo grado di imprecisione.

Conversione decimale → binario

Si utilizza l'algoritmo della divisione. Si divide il numero decimale per 2 ripetutamente finché il risultato non è 0 e si prendono i resti delle divisioni in ordine inverso.

Esempio
Convertire il numero decimale 12 in binario.
Soluzione



La soluzione è dunque $12_{10} = 1100_2$.
Controprova
Basta convertire il numero binario ottenuto in decimale

$$1100_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 = 12$$

Esercizio
Convertire in binario i seguenti numeri decimali:

1) 128
2) 67217
3) 100169