

## Sistemi numerici

Il sistema che utilizza il pc per elaborare le istruzioni è il **sistema binario**, esso è composto da due sole cifre 0 e 1.

Utilizzando il sistema binario e avendo n cifre a disposizione possiamo rappresentare  $2^n$  numeri, questo per via del calcolo combinatorio:

$$2 \cdot 2_1 \dots 2_n$$

### • Numeri decimali

$$5374_{10} = 5 \cdot 10^3 + 3 \cdot 10^2 + 7 \cdot 10^1 + 4 \cdot 10^0$$

### • Numeri binari

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$$

## Potenze di 2

• $2^0 = 1$	• $2^8 = 256$
• $2^1 = 2$	• $2^9 = 512$
• $2^2 = 4$	• $2^{10} = 1024$
• $2^3 = 8$	• $2^{11} = 2048$
• $2^4 = 16$	• $2^{12} = 4096$
• $2^5 = 32$	• $2^{13} = 8192$
• $2^6 = 64$	• $2^{14} = 16384$
• $2^7 = 128$	• $2^{15} = 32768$

## Conversione numerica

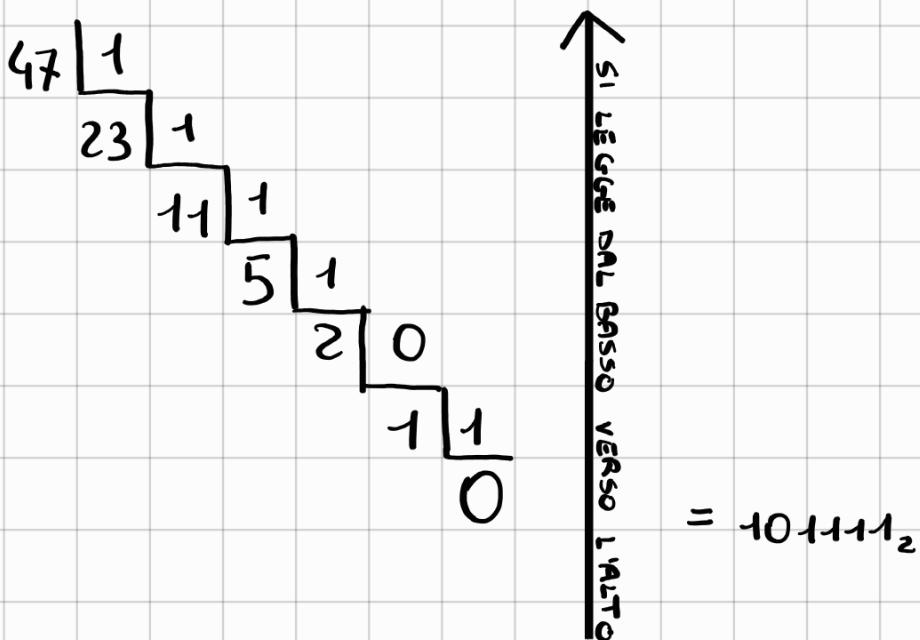
### • Conversione da binario a decimale:

Un esempio di conversione in decimale può essere 10011<sub>2</sub> che in decimale diventa:

$$\begin{aligned}10011_2 &= 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2^1 \cdot 1 + 2^0 \cdot 1 = \\&= 16 \cdot 1 + 8 \cdot 0 + 4 \cdot 0 + 2 \cdot 1 + 1 \cdot 1 = \boxed{19_{10}}\end{aligned}$$

• Conversione da decimale a binario:

un esempio di conversione in decimale può essere  $47_{10}$  che in decimale diventa:



Per la conversione da decimale a binario ci sono 2 metodi:

• Metodo 1:

- Trova la più grande potenza di 2 più adatta;
- sottrai;
- se non esce 0 ripeti.

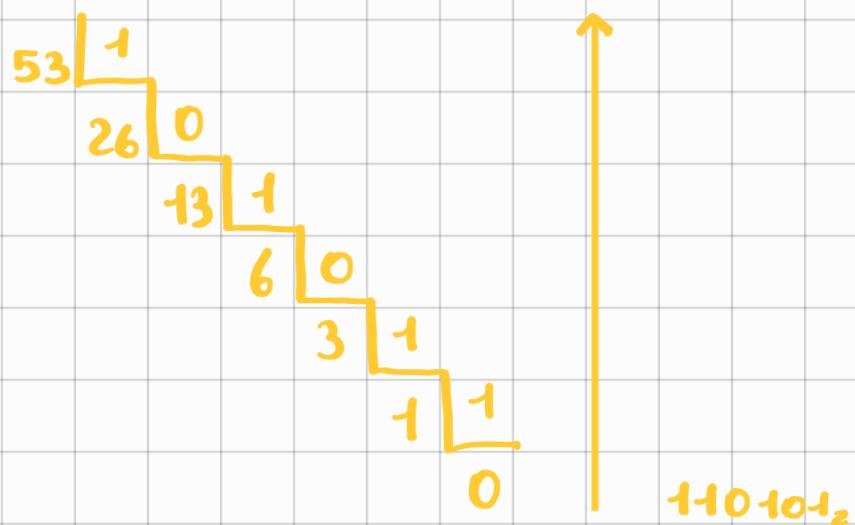
Esempio:



• Metodo 2:

- dividi per 2;
- metti il resto a sinistra della rappresentazione binaria.

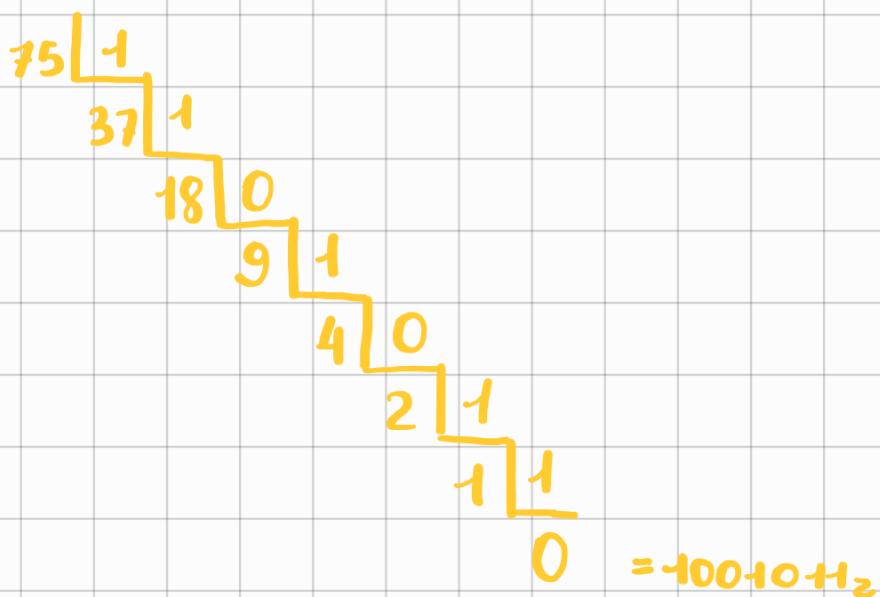
Esempio:



Altri esempi dei due metodi possono essere:

$$75_{10} = 64 + 8 + 2 + 1 = 100101_2$$

oppure



Da  $x$  a  $y$

Quando invece abbiamo una base  $x$  e vogliamo convertirla ad una base  $y$  prima convertiamo da  $x$  a decimale e poi da decimale a  $y$ .

Esempio:

Abbiamo  $25_{20}$  e vogliamo convertirlo in base 4:

1) Convertiamo prima in base 10 quindi:

$$25_{20} = 2 \cdot 20^1 + 5 \cdot 20^0 = 40 + 5 = 45$$

2) convertiamo 45 in base 4 e per fare questo basterà dividere 45 finché non esce 0 e ad ogni divisione metteremo da parte i resti:

RESTI DI SOTTRAZIONE

$$\begin{array}{r} 45 \\ | \quad 1 \\ 11 \quad | \quad 3 \\ | \quad 2 \quad | \quad 2 \\ 0 \end{array}$$

$= 231_4$

## Valori binari e intervalli

### • Numero decimale a N cifre

- Quanti valori?  $10^n$

- Range?  $[0, 10^n - 1]$

Esempio con un numero decimale a 3 cifre:

- $10^3 = 1000$  possibili valori;

- Range:  $[0, 999]$

### • Numero binario a n cifre

- Quanti valori?  $2^n$

- Range  $[0, 2^n - 1]$

Esempio con un numero binario a 3 cifre:

- $2^3 = 8$  possibili valori

- Intervallo:  $[0, 7] = [000_2, 111_2]$

## Numeri esadecimali

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Le caratteristiche principali dei numeri esadecimali sono 2:

- ha come base 16
- abbreviazione binaria

## Conversione da esadecimale a binario

Proviamo per esempio a convertire 4AF (anche scritto 0x4AF) in binario:

0 1 0 0 1 0 1 0 1 1 1 1  
  4      A      F      2

## Conversione da esadecimale a decimale

Per la conversione da esadecimale a decimale invece proviamo a convertire sempre 4AF<sub>16</sub> in decimale:

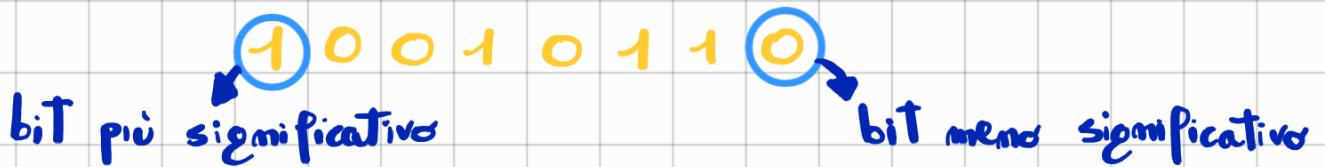
$$16^2 \cdot 4 + 16^1 \cdot 10 + 16^0 \cdot 15 = 1199_{10}$$

## Bits, Bytes, Nibble...

Un bit è una singola unità di un numero binario ad esempio 0 o 1.

Invece un **byte** sono 8 bit ed un nibble 4 bit, quindi 2 nibble formano un byte.

- Un esempio di bits può essere:



- Esempio di Bytes & Nibble invece può essere:



- Infine un esempio di Bytes può essere.



## Grandi potenze di 2

Esempi di grandi potenze di 2 sono:

- $2^{10} = 1 \text{ Kilo} \approx 1000(1024)$
- $2^{20} = 1 \text{ mega} \approx 1 \text{ milione}(1048576)$
- $2^{30} = 1 \text{ giga} \approx 1 \text{ miliardo}(1073741824)$

## Addizioni

Le addizioni con numeri decimali e binari si fa in due modi diversi:

- decimale

$$\begin{array}{r} 1 \\ + 3734 \\ + 5168 \\ \hline 8902 \end{array}$$

RIPORTI

obinario

RIPORTI

$$\begin{array}{r} 1011 \\ + 0011 \\ \hline 1110 \end{array}$$

In binario le cose che cambiano nell'addizione sono:

- 1+1 fa 0 con riporto di 1
- 0+0 fa 0
- 1+0 fa 1
- 0+1 fa 1

### Esempi di operazioni esadecimali

Facciamo 2 esempi di operazione con numeri esadecimali:

• Addizione:

$$\begin{array}{r} 3A09 \\ + 1B17 \\ \hline 5520 \end{array}$$

9  
A  
B  
C  
D  
E  
F  
0 com riporto di 1

• Sottrazioni:

$$\begin{array}{r} 3D09 \\ - 1B27 \\ \hline 21E2 \end{array}$$

### Overflow

I sistemi digitali operano su un numero fisso di bit ed overflow si riferisce a quando il risultato è troppo grande per adattarsi al numero di bit disponibile.

Cio' avviene se avendo una nibble a disposizione proviamo a computare per esempio  $12 + 11$  che fa 23 e sarai

maggiori rispetto a  $2^{4-1}$  ovvero 15

## Numeri Sign/Magnitude

Hanno 1 bit di segno e N-1 magnitude bits.

Il bit più a sinistra va ad identificare il segno del bit

- numero positivo il bit varrà 0;
- numero negativo il bit varrà 1.

$$A = (-1)^{a_{N-1}} \sum_{i=0}^{N-2} a_i z^i$$

Per esempio la rappresentazione in sign/mag a 4 bit di  $\pm 6$  è

$$+6 = 0110$$

$$-6 = 1110$$

Il range di un numero N bit sign/mag è:

$$[-(2^{N-1}-1), 2^{N-1}-1]$$

## Numeri Sign/Magnitude

Esistono vari problemi con i numeri sign/mag per esempio le addizioni non funzionano, tipo  $-6 + 6$

$$\begin{array}{r} 1110 \\ + 0110 \\ \hline 10100 \end{array} \text{(SBAGLIATO!)}$$

Un altro problema è che esistono 2 rappresentazioni di  $\pm 0$

$$\blacktriangleright -0 = 1000$$

$$\blacktriangleright +0 = 0000$$

## Numeri in complemento a due

Nel complemento a due non abbiamo lo stesso problema che abbiamo nei numeri sign/magnitude quindi troveremo:

- addizioni funzionanti
- una sola rappresentazione dello 0

Nel complemento a due abbiamo

- il bit più significativo che ha come valore  $-2^{N-1}$

$$A = a_{N-1}(-2^{N-1}) + \sum_{i=0}^{N-2} a_i z^i$$

- numero a 4 bit più positivo è: 0 1 1 1
- numero a 4 bit più negativo è: 1 0 0 0

- Il bit più significativo indica ancora il segno quindi 1=negativo e 0=positivo
- Il range di un numero in complemento a due di N bit è:

$$[(z-1), 2^{N-1}-1]$$

Un numero col complemento a 2 avente n bit può rappresentare numeri che vanno da un range di: