

Se abbiamo un valore decimale in base 10, ad esempio 0.99, come troviamo la sua rappresentazione in base 2? Ragioniamo come segue:

Supponiamo che

$$(.99)_{10} = .b_1b_2b_3\dots b_k \text{ (in binario)}$$

$$\text{allora } 2 \times .99 = 1.98 = b_1.b_2\dots b_k$$

quindi  $b_1$  è 1

e .98 è rappresentato da  $.b_2\dots b_k$

**Quindi:** per trovare la rappresentazione binaria di un valore decimale lo moltiplichiamo per 2 e osserviamo la cifra che appare nella parte intera.

$$\text{Esempio: } (.59)_{10} = (?)_2$$

$$.59 \times 2 = 1.18$$

$$.18 \times 2 = 0.36$$

$$.36 \times 2 = 0.72$$

$$.72 \times 2 = 1.44$$

$$.44 \times 2 = 0.88$$

$$.88 \times 2 = 1.76$$

...

$$(0.59)_{10} = (.100101\dots)_2$$

**E così via... dipende da quanti bit abbiamo a disposizione!**

**N.B.** Mi posso fermare se moltiplicando per 2 ottengo una parte decimale = 0

# Esempio Completo

$$(18.59)_{10} = ( ? )_2$$

$$(18)_{10} = (10010)_2$$

$$(.59)_{10} = (.100101\dots)_2$$

Ne deriva che:

$$(18.59)_{10} = (10010.100101\dots)_2$$

Osserviamo che spostando la virgola di  $h$  posti verso destra nella rappresentazione decimale di un numero  $N$  moltiplichiamo  $N$  per  $10^h$  mentre spostandola verso sinistra di  $h$  posti  $N$  viene diviso per  $10^h$ .  
Ad esempio:

$$\begin{aligned} 18.59 &= 18.59 \times 10^0 \\ &= 1.859 \times 10^1 && \text{rappresentazione normalizzata} \\ &= 0.1859 \times 10^2 \\ &= 185.9 \times 10^{-1} \\ &= 1859 \times 10^{-2} \end{aligned}$$

Si parla di rappresentazione in virgola mobile. Quindi in virgola mobile lo stesso numero decimale ha piu' rappresentazioni possibili. Solitamente si usa la rappresentazione normalizzata in cui la parte intera ha un'unica cifra decimale diversa da zero.

La rappresentazione in virgola mobile per i numeri binari e` del tutto analoga. Ad esempio:

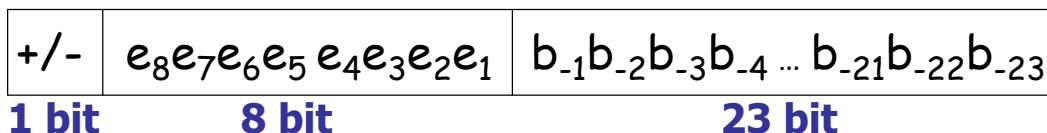
$$\begin{aligned}
 101.01 &= 101.01 \times 2^0 \\
 &= 10.101 \times 2^1 \\
 &= 1.0101 \times 2^2 \quad \text{Rappresentazione normalizzata} \\
 &= 1010.1 \times 2^{-1} \\
 &= 10101 \times 2^{-2}
 \end{aligned}$$

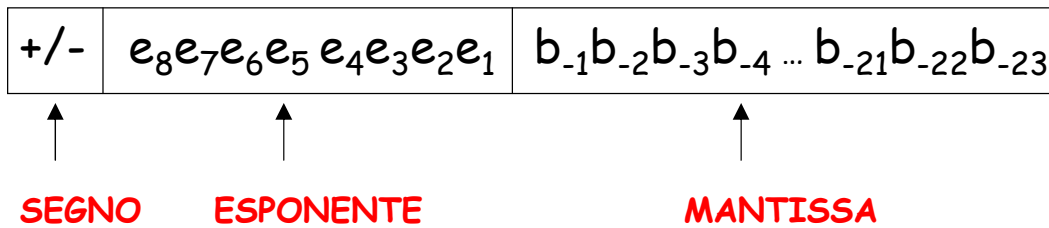
La rappresentazione in virgola mobile normalizzata ha sempre parte intera uguale a 1.

Per registrare un numero reale  $x$  in memoria si adotta la rappresentazione binaria in virgola mobile normalizzata:

$$x = 2^e \times 1.b_{-1}b_{-2}b_{-3}b_{-4}...$$

Naturalmente non e` possibile memorizzare la sequenza possibilmente infinita di bit della parte frazionaria ma si memorizzano soltanto i primi bit. Anche per l'esponente si utilizzano un numero finito di bit. Generalmente per rappresentare un numero reale si usano 4 byte nel modo seguente:





Il primo bit rappresenta il segno del numero: 0 per +, 1 per -.  
 Gli otto bit successivi si usano per rappresentare l'esponente  $esp$ .  
 Dovendo rappresentare sia esponenti positivi che negativi la  
 sequenza di bit  $e_8 e_7 e_6 e_5 e_4 e_3 e_2 e_1$  e' la rappresentazione  
 binaria di  $esp + 127$ . Con 8 bit si rappresentano gli interi compresi  
 tra 0 e 255. Siccome le sequenze di bit 00000000 (0) e 11111111  
 (255) sono riservate (problema che non consideriamo), gli esponenti  
 rappresentabili sono quelli per cui  
 $1 \leq esp + 127 \leq 254$  ovvero  $-126 \leq esp \leq 127$ .

## Rappresentazione dei decimali



Quindi si rappresenta il numero

$$N = s \cdot 2^{e-127} \times 1.m$$

0	10000011	0011100101100000000000
---	----------	------------------------

$$10000011 = 131$$

$$\text{esp} = 131 - 127 = 4. \text{ Quindi}$$

$$2^4 \times 1.00111001011 = 10011.1001011$$

$$= 2^4 + 2^1 + 2^0 + 2^{-1} + 2^{-4} + 2^{-6} + 2^{-7}$$

$$= 19.5859375$$

**Esercizio:** Fornire la rappresentazione in virgola mobile normalizzata del valore 10.543 avendo a disposizione 8 bit per l'esponente e 8 per la mantissa.

(1) Rappresentiamo 10 in binario

$$10 = 2^3 + 2^1 = (1010)_2$$

(2) Rappresentiamo 0.543 in binario

$$0.543 \times 2 = 1.086$$

$$0.086 \times 2 = 0.172$$

$$0.172 \times 2 = 0.344 \quad \text{Quindi: } 0.543 = (0.10001\dots)_2$$

$$0.344 \times 2 = 0.688$$

$$0.688 \times 2 = 1.376$$

$$0.376 \times 2 = 0.752$$

(3) Riassumendo:

$$10.543 = (1010.100010\dots)_2$$

(4) Rappresentazione normalizzata

$$1.01010001 \times 2^3 = 1010.10001$$

(5) Rappresentiamo l'esponente -3

$$3 + 127 = 130$$

$$130 = (10000010)_2$$

**QUINDI:**

<b>0</b>	<b>10000010</b>	<b>01010001</b>
----------	-----------------	-----------------