

# Tramonta Emanuele

## Conversione 1 – 0,3 punti

Converti nel formato decimale il numero 10000000 00000000 00000000 00000000

### Conversione senza segno o complemento

Eleva tutte le cifre alla  $2^x$ , dove x è la posizione della cifra partendo con 0 da destra.

$$(1 \cdot 2^{31}) + (0 \cdot 2^{30}) + \dots + (0 \cdot 2^0) = 1 \cdot 2^{31} = \mathbf{2.147.483.648}$$

### Conversione complemento a 2

Innanzitutto notiamo che il bit più significativo è 1, quindi il numero sarà negativo.

Per convertirlo secondo il complemento a due, ci sono due step:

- Complemento a 1 : inverte bit
  - 01111111 11111111 11111111 11111111
- Complemento a 2 : sommo 1
  - 01111111 11111111 11111111 11111111 +
  - 00000000 00000000 00000000 00000001 =
  - -----
  - 10000000 00000000 00000000 00000000
  - La somma applicata al numero porta l'1 al bit più a sinistra tramite riporto, come nei decimali:  $0999 + 1 = 1000$

10000000 00000000 00000000 00000000 è il numero finale

$$(1 \cdot 2^{31}) + (0 \cdot 2^{30}) + \dots + (0 \cdot 2^0) = 1 \cdot 2^{31} = 2.147.483.648$$

Ricordandoci che il numero è negativo, il risultato sarà: **-2.147.483.648**

### Conversione con segno senza complemento a due

10000000 00000000 00000000 00000000

Prendo il primo bit come segno, 1=negativo.

I rimanenti 31 bit rappresentano il valore, in questo caso 0.

Risultato: **0**.

## Conversione floating point

10000000 00000000 00000000 00000000

Il primo bit rappresenta il segno, in questo caso possiamo aspettarci di avere un numero negativo.

I seguenti 8 bit rappresentano la mantissa, o l'esponente in notazione scientifica.

L'essere tutti 0 lo rende un caso speciale:

se mantissa != 0, si ha un numero denormalizzato

se mantissa == 0, si ha il valore 0

Il nostro caso è il secondo, quindi il valore di questo numero è: **-0**.

Notare il -, in quanto 0 e -0 sono entrambi rappresentabili secondo IEEE-754.

## Conversione 2 - 0,3 punti

Converti nel formato decimale il numero 11111111 11111111 11111111 11111111

## Conversione senza segno o complemento

Il valore sarà semplicemente il massimo valore rappresentabile con 32 bit, ovvero

$$2^{32}-1=4.294.967.295$$

Alternativamente, si può calcolare con il valore posizionale

$$(1*2^{31})+(1*2^{30})+...+(1*2^0)$$

## Conversione complemento a 2

Bit più significativo è 1, quindi numero negativo

- Applico complemento a 1, inverto bit:

- 00000000 00000000 00000000 00000000

- Applico complemento a 2, sommo 1:

- 00000000 00000000 00000000 00000000 +

- 00000000 00000000 00000000 00000001 =

- -----

- 00000000 00000000 00000000 00000001

Valore finale: **-1**

## Conversione con segno senza complemento a due

11111111 11111111 11111111 11111111

Primo bit: 1, quindi negativo.

Valore: massimo valore ottenibile con 31 bit,  $2^{31}-1 = 2.147.483.648$

Risultato: **-2.147.483.648**

## Conversione floating point

11111111 11111111 11111111 11111111

Primo bit: 1, numero negativo

Exp: 255, caso speciale:

- Se mantissa==0, rappresenta -infinito
- Se mantissa!=0, rappresenta NaN

Nel nostro caso, la mantissa è decisamente !=0, quindi

Risultato: **NaN**

Entrambi i casi in floating point erano casi speciali con exp 0 o 255, quindi non è stato necessario depolarizzare l'esponente.