Università degli Studi
di Cassino e del Lazio Meridionale

Corso di Fondamenti di Informatica

Rappresentazione dei numeri reali

Anno Accademico 2013/2014

Francesco Tortorella

Numeri reali in base 2

- La rappresentazione dei numeri reali in base 2 è completamente analoga a quella in base 10:
  - Parte intera + parte frazionaria, separate da un punto
- La parte frazionaria è formata da cifre che pesano le potenze di 2 a esponente negativo.
  - Esempio: $110.0101_2 \rightarrow 2^{+2}+2^{+1}+2^{-2}+2^{-4} = 6.3125$
- Conversione: si convertono separatamente la parte intera e quella frazionaria.
- Come si converte la parte frazionaria ?

Conversione base 10 → base 2 (frazionari)

Consideriamo un numero F minore di 1.

$$F = c_{-1}x2^{-1} + c_{-2}x2^{-2} + \ldots + c_{-n}x2^{-n} \qquad c_i \in \{0,1\}$$

$$Fx2 = c_{-1} + (c_{-2}x2^{-1} + \ldots + c_{-n}x2^{-(n-1)}) = c_{-1} + P_1 \quad P_1<1$$

$$P_1x2 = c_{-2} + (c_{-3}x2^{-1} + \ldots + c_{-n}x2^{-(n-2)}) = c_{-2} + P_2$$

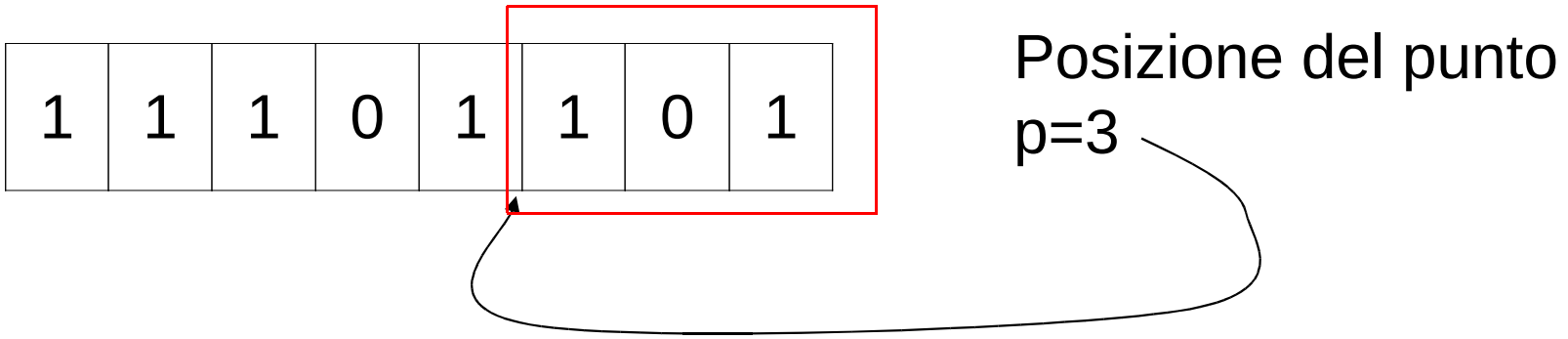
Rappresentazione nei registri dei numeri reali

- Come rappresentiamo 22.315 ?
- A differenza dei numeri interi, per rappresentare i numeri reali è necessario codificare la posizione del punto frazionario

- Due soluzioni:
  - Codifica esplicita → virgola fissa
  - Codifica implicita → virgola mobile

Rappresentazione in virgola fissa

- Con la codifica implicita, si assume prefissata la posizione del punto all'interno del registro → Rappresentazione in virgola fissa (fixed point)
- Esempio:



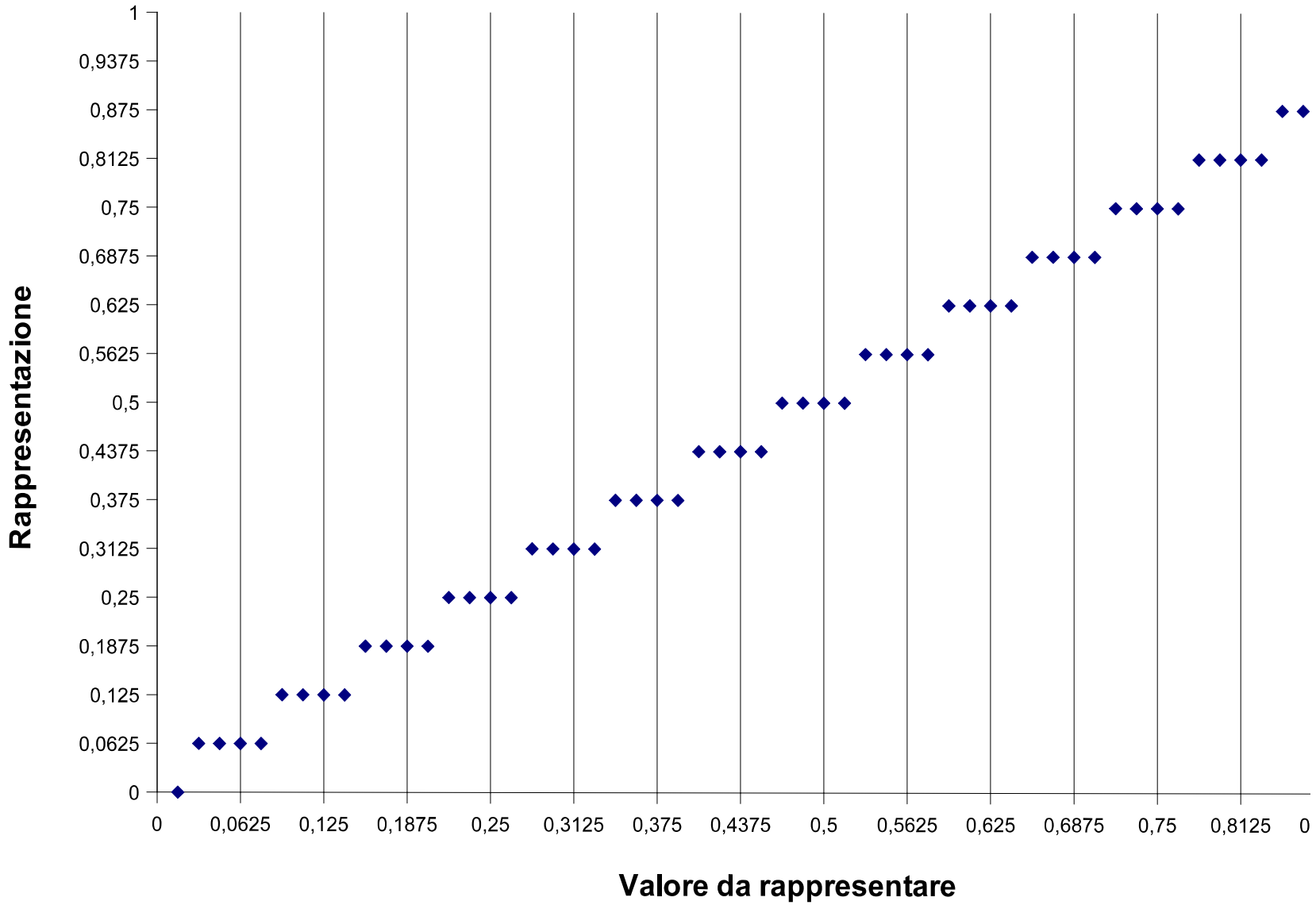
il numero rappresentato è 11101.101

Rappresentazione in virgola fissa

- Con questa convenzione, il valore X rappresentato nel registro è $K*2^{-p}$, dove K è il valore che otterremmo se interpretassimo come un intero il contenuto del registro.
- Qual è l'insieme dei valori rappresentabili su un registro a N bit ?

  K: $0,1,2,\ldots,2^N-1$ → X: $0, 2^{-p}, 2*2^{-p}, \ldots, (2^N-1)*2^{-p}$
- Esempio: N=8, p=4

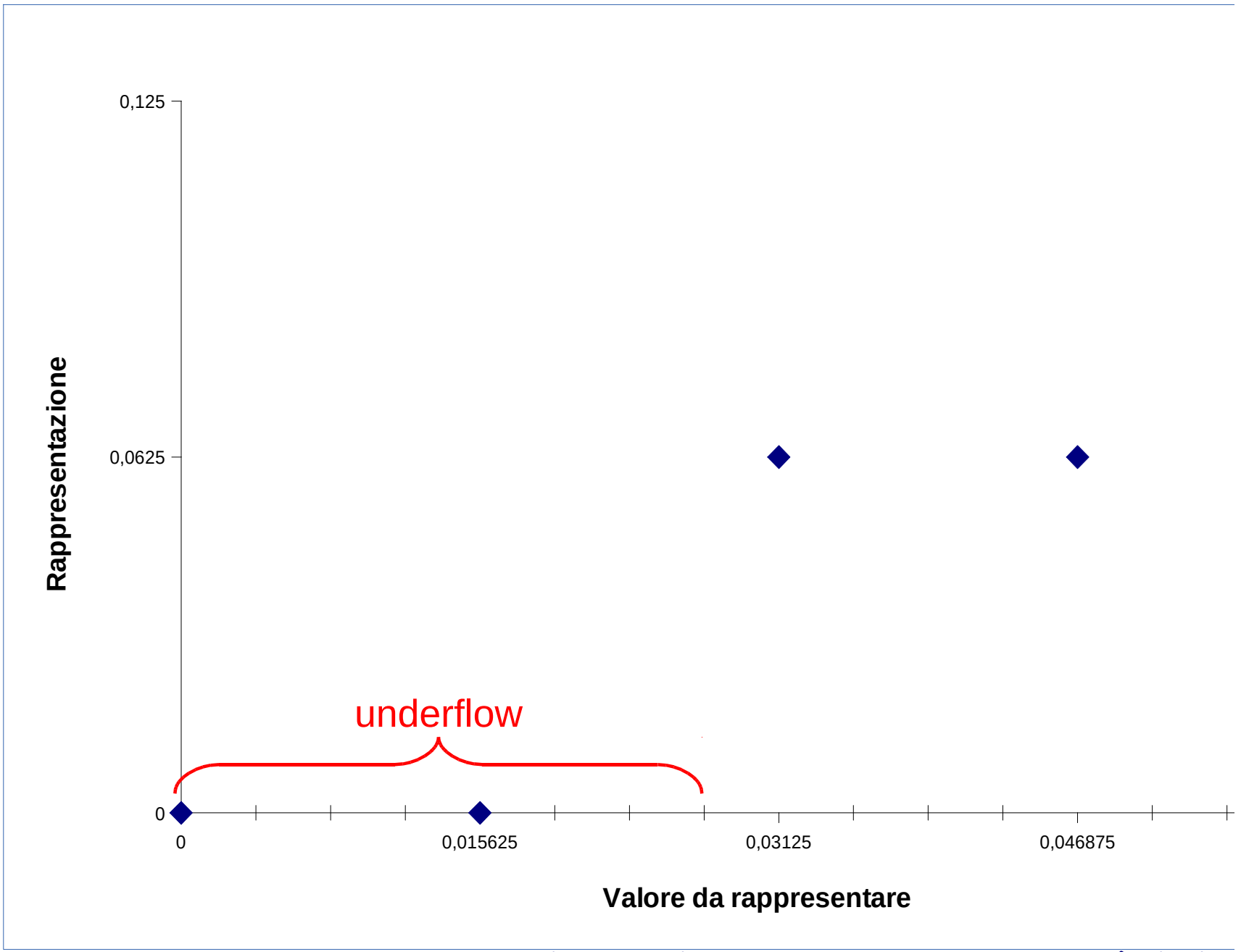
  X= 0, 0.0625, 0.125, 0.1875,…, 15.9375


Rappresentazione
1
0,9375
0,875
0,8125
0,75
0,6875
0,625
0,5625
0,5
0,4375
0,375
0,3125
0,25
0,1875
0,125
0,0625
0
0
0,0625
0,125
0,1875
0,25
0,3125
0,375
0,4375
0,5
0,5625
0,625
0,6875
0,75
0,8125
0
Valore da rappresentare

Rappresentazione in virgola fissa

- I numeri sono rappresentati con una certa approssimazione
  - Esempio: tutti i valori compresi tra 0.03125 e 0.09375 sono rappresentati da 0.0625

- Tutti i valori compresi tra 0 e 0.03125 sono rappresentati da 0.0000 → underflow


0,125
0,0625
0
Rappresentazione
underflow
0
0,015625
0,03125
0,046875
Valore da rappresentare

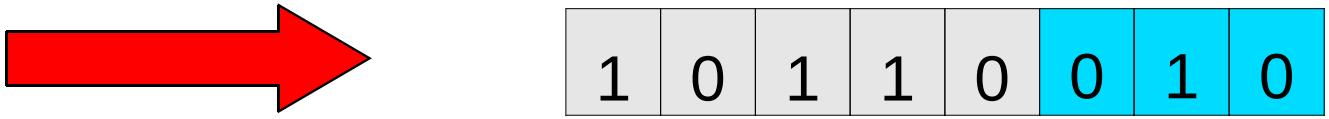
Rappresentazione di un numero in virgola fissa

Supponiamo di voler rappresentare il numero 22.315 in virgola fissa in un registro ad 8 bit con p=3.

Separiamo parte intera e parte frazionaria:

$22_{10} \rightarrow 10110_2$

$0.315_{10} \rightarrow 0.010100\ldots_2$



Riassumendo…

- La rappresentazione in virgola fissa ha innegabili vantaggi:
  - Semplicità
  - Piena compatibilità con la rappresentazione degli interi e possibilità di usare circuiti aritmetici comuni.
- Ma ha anche grossi problemi:
  - Errore relativo elevato per x →0
  - Compromesso range/precisione
  - Entrambi legati al fatto che il fattore di scala è fisso.

La virgola è mobile…

- Si potrebbero mitigare i problemi andando a rappresentare esplicitamente il fattore di scala.
- In questo modo la virgola non è più “fissa”, ma diventa “mobile”.
- Rappresentazione in virgola fissa → Rappresentazione in virgola mobile (floating point)

Rappresentazione in virgola mobile

- Fissata la base b, il valore viene considerato nella forma $M*b^E$ (notazione scientifica) ed è rappresentato tramite la coppia (M,E)

  Esempio: $22.315=0.22315*10^2$ → (0.22315,2)

  $10110.010=10.110010*2^3$ → (10.110010,11)

- Nel registro saranno quindi prefissate zone diverse per la mantissa e per l'esponente

Rappresentazione in virgola mobile

Come si rappresentano M ed E ?

- M
  - numero reale
  - segno e modulo
  - virgola fissa
- E
  - numero intero con segno
  - eccessi
- La disposizione nel registro facilita il confronto

+/-	E	M

Intervallo di numeri rappresentabili

- M rappresentato su m bit con p cifra frazionarie
  M: $0, 2^{-p}, 2*2^{-p}, \ldots, (2^m-1)*2^{-p}$

- E rappresentato su e bit
  E: $-2^{e-1},\ldots,+2^{e-1}-1$

- $N_{min}=M_{min}*2^{Emin} = 2^{-p}*2^{-2^{e-1}}$

- $N_{max}=M_{max}*2^{Emax} = (2^m-1)* 2^{-p}*2^{+2^{e-1}-1}$

Intervallo di numeri rappresentabili

- Esempio:
  - m=23 p=23
  - e=8
- $N_{min} = 2^{-23} * 2^{-128} \cong 3.5 * 10^{-46}$
- $N_{max} = (2^{23}-1) * 2^{-23} * 2^{127} \cong 1.7 * 10^{+38}$

Esempio

Rappresentazione in FP di –12.6:

$12.6_{10} = 1100.\overline{1001}_2 = 0.1100\overline{1001} * 2^4$

Segno: 1

Mantissa: 0.11001001100110011001100

Esponente: $4+128 = 132_{10} = 10000100_2$

Rappresentazione normalizzata

- Con la virgola mobile non c'è unicità di rappresentazione:
  $N = M*2^E = (M*2)*2^{E-1} = (M*4)*2^{E-2} = (M/2)*2^{E+1}$

- Quale scegliere ? Quella che massimizza la precisione:
  prima cifra della mantissa diversa da 0
  → rappresentazione normalizzata

Rappresentazione normalizzata

- Esempio: N = 0.0003241892
  mantissa a 5 cifre decimali
- Diverse rappresentazioni possibili:

  $0.00032*10^{0}$

  $0.00324*10^{-1}$

  $0.03241*10^{-2}$

  $0.32418*10^{-3}$ ← normalizzata

Rappresentazione normalizzata

- L'intervallo di rappresentazione si modifica :
  $N_{min} = 2^{m-1} * 2^{-p} * 2^{-2^{e-1}}$

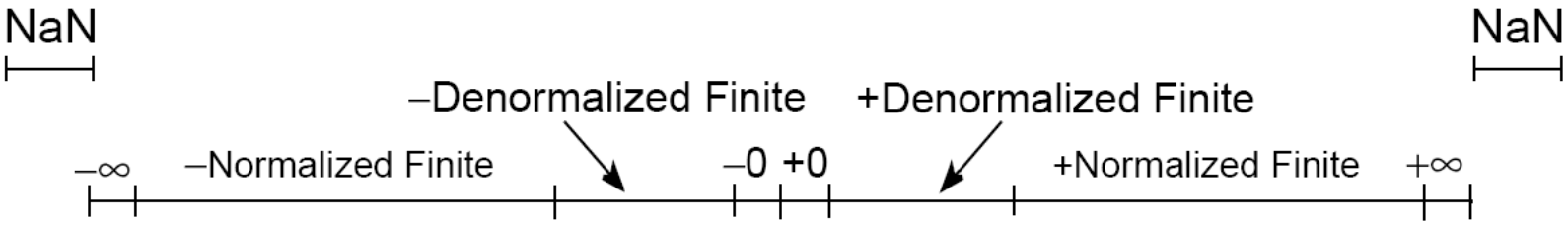
- Esempio:
  - m=23 p=23
  - e=8

- $N_{min} = 2^{-23} * 2^{-128} \cong 3.5 * 10^{-46}$ (non normalizzata)
- $N_{min} = 2^{22} * 2^{-23} * 2^{-128} \cong 1.5 * 10^{-39}$ (normalizzata)

Lo standard IEEE754

- Due formati
  - 32 bit: 23 bit mantissa + 8 bit esp. + 1 bit segno, bias=127
  - 64 bit: 52 bit mantissa + 11 bit esp. + 1 bit segno, bias=1023
- Mantissa con hidden bit
  $N = (-1)^s * (1.M) * 2^{E-bias}$
- Esponente polarizzato
  - Sono riservate le rappresentazioni dell'esponente 00…0 e 11…1
- Underflow graduale, denormalizzazione

Lo standard IEEE754



Real Number and NaN Encodings For 32-Bit Floating-Point Format

S	E	F		S	E	F
1	0	0	–0	0	0	0
1	0	$0.XXX^2$	–Denormalized Finite	0	0	$0.XXX^2$
1	1...254	Any Value	–Normalized Finite	0	1...254	Any Value
1	255	0	–∞	0	255	0
X^1	255	$1.0XX^2$	–SNaN	X^1	255	$1.0XX^2$
X^1	255	1.1XX	–QNaN	X^1	255	1.1XX

			+0			
			+Denormalized Finite			
			+Normalized Finite			
			+∞			
			+SNaN			
			+QNaN			

NOTES:
1. Sign bit ignored.
2. Fractions must be non-zero.

Lo standard IEEE754

	Range denormalizzato	Range normalizzato	Decimale
32 bit	Min: 2^{-149} Max:$(1-2^{-23})\times2^{-126}$	Min: 2^{-126} Max: $(2-2^{-23})\times2^{127}$	1.4×10^{-45} 3.4×10^{38}
64 bit	Min: 2^{-1074} Max:$(1-2^{-52})\times2^{-1022}$	Min: 2^{-1022} Max: $(2-2^{-52})\times2^{1023}$	4.9×10^{-324} 1.8×10^{308}

Lo standard IEEE 754

- Esistono rappresentazioni riservate (definite "numeri speciali") che permettono l'estensione dell'aritmetica a casi particolari:
  - NaN (0/0, sqrt(-2^k))
  - $+\infty$, $-\infty$

denormalizzato 

E	M	N
255	**≠0**	**NaN**
255	**=0**	**$(-1)^s \infty$**
1-254	**qualunque**	+/- numero fp
0	**0**	**0**
0	**≠0**	**$(-1)^s * 2^{-126} * (0.M)$**