Università degli Studi dell’Insubria
Dipartimento di Scienze Teoriche e Applicate

Architettura degli elaboratori

La rappresentazione dell’informazione nei calcolatori

Numeri binari frazionari

- Come si esprime un numero non intero in binario?
- Esattamente come in base dieci:

$$235.13_{10}$$

$$=$$

$$2 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 1 \times 10^{-1} + 3 \times 10^{-2}$$

2 centinaia + 3 decine + 5 unità + 1 decimi + 3 centesimi

- Esempio in base 2: 101.1001_2

$$=$$

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$$

$$= 4 + 1 + 1/2 + 1/16 = 5.5625_{10}$$

Numeri binari non interi

- oppure: cosi come, in base 10:

$$235.13 = 235 \text{ unità} + 13 \text{ centesimi } (/10^2)$$

(perché sono 2 cifre dopo la virgola)

$$235.138 = 235 \text{ unità} + 138 \text{ millesimi } (/10^3)$$

(perché sono 3 cifre dopo la virgola)

- nello stesso modo, in base 2:

$$101.1001_2 = 101_2 \text{ unità} + 1001_2 / 2^4 = 5 + 9/16 = 5.5625_{10}$$

Numeri binari frazionari

- Come si esprime un numero compreso tra zero e uno in binario?
- Esattamente come in base dieci: il MSB pesa 1/2, il successivo 1/4, il successivo 1/8, ecc.

- Generalizzando quindi:

$$v = 2^{-1} * d_{k-1} + 2^{-2} * d_{k-2} + \ldots + 2^{-k+1} * d_1 + 2^{-k} * d_0$$

- Esempi:
  - $00101101 = 2^{-3} + 2^{-5} + 2^{-6} + 2^{-8} = 0.17578125$
  - $00010000 = 2^{-4} = 0.0625$
  - $10100000 = 2^{-1} + 2^{-3} = 0.625$

Conversione da decimale a binario di numeri frazionari

- Dato un valore v, tale che 0<v<1, esso è denotato da

  $v = 2^{-1}*d_{k-1} + 2^{-2}*d_{k-2} +...+ 2^{-k}*d_0$

- Raccogliendo 2^{-1} si ottiene $v = 2^{-1}*(d_{k-1} + 2^{-1}*d_{k-2} +...+ 2^{-k+1}*d_0)$
- Pertanto se moltiplico v per 2 la parte intera del risultato, ovvero d_{k-1}, corrisponde alla prima cifra frazionaria della rappresentazione binaria di v

  $2v = d_{k-1} + \boxed{2^{-1}*d_{k-2} +...+ 2^{-k+2}*d_1 + 2^{-k+1}*d_0}$

  parte intera parte frazionaria

- $2v < 1$ iff $d_{k-1} = 0$ (e viceversa, $2v \geq 1$ iff $d_{k-1} = 1$)
- Una volta noto d_{k-1}, il problema diventa codificare la parte frazionaria di 2v, cioè $2v-d_{k-1}$
- Si moltiplica la parte frazionaria di 2v per 2 e si guarda se è < 1, ecc… Si reitera fino ad ottenere una parte frazionaria nulla, o un numero sufficiente di cifre

Conversione da decimale a binario di numeri frazionari: esempio

0.748

1.496 raddoppio della parte frazionaria (quella dopo la virgola)

0.992

1.984

1.968

1.936

1.872

1.744

1.488

0.976

$0.748_{10} = 0.101111110\ldots_2$

Conversione da decimale a binario di numeri frazionari: esempio

0.15

raddoppio della parte frazionaria (quella dopo la virgola)

0.30

0.60

1.20

0.40

0.80

1.60

1.20

0.40

0.80

$0.15_{10} = 0.001001100\ldots_2$

Note matematiche sullo sviluppo di numeri con la virgola

- Tutti (e soli) i numeri razionali ($\in \mathbb{Q}$, le frazioni) hanno sempre sviluppi che sono o finiti o periodici:
  - finiti: es. ½ = 0.5, ¼ = 0.25
  - periodici: es 1/3 = 0.33333333….   1/9 = 0.1111111…..

- I numeri non razionali ($\notin \mathbb{Q}$) hanno sviluppi infiniti e NON periodici
  - es: sqrt(2) = 1.41421356237….   pi = 3.14159265359….

- Questo vale a prescindere dalla base
  - ma una stessa frazione può avere uno sviluppo periodico in una base e finito in un'altra
    es: 1/5 = 0.2   in base 10,
    1/5 = 0.0011001100110011001100110011….   in base 2