Università
degli Studi
della Campania
Luigi Vanvitelli

Scuola Politecnica e
delle Scienze di Base

Dipartimento di
Ingegneria Industriale e
dell'Informazione

Rappresentazione dei numeri

Prof. Salvatore Venticinque
Prof. Mauro Iacono

Rappresentazione dei numeri

- Intervallo numerico
- Base di numerazione e numero di cifre
- Approssimazione
- Condizione di overflow
- Condizione di underflow

Rappresentazione dei numeri

- Nel **sistema romano** ciascuna cifra esprime una quantità indipendente dalla propria posizione

  MML (2050)

  MDCCLXXXVIII (1988)

- Nel sistema decimale ogni cifra ha un peso variabile a seconda della posizione nel numero (codice)

  $84079 = 8*10^4+4*10^3+0*10^2+7*10^1+9*10^0=$

  $80000 + 4000 + 0 + 70 +9$

Sistemi di numerazione posizionali

Fissati k simboli si può definire un sistema posizionale in base k.

La base di rappresentazione viene specificata in pedice:

$(4791)_{10}$ $(1431)_5$ $(10101)_2$

- Per convertire la rappresentazione di numero da base k a base 10:

  è sufficiente considerare la successione di cifre con il peso nella base k ed eseguire la somma

- Esempio:

  $(10101)_2 = 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 21$

Conversioni base 2^k ← → base 2

- Conversione diretta
  - Ogni cifra deve essere sostituita con la corrispondente parola codice di k bit
- Conversione inversa
  - Si raggruppano i bit in gruppi di k a partire dal bit meno significativo (da sinistra verso destra)
  - Ogni gruppo deve essere sostituito con la corrispondente cifra.

Esempio:

A78F → 1010 0111 1000 1111

Rappresentazione di Interi Positivi

- Base=2
- Numero di bit=n
- Numero rappresentazioni: 2^n
- Intervallo: $[0, 2^n[$

n=5 → massimo rappresentabile: $2^5-1=31$

5←→00101

20←→ 10100

Addizione

```
2+6=?
00010+
00110=
---------
01000
```

```
1+31=?
11111+
00001=
---------
00000
```

Riporto=1

- Quando si verifica la condizione di overflow?
- Qual è il suo significato?

Modulo e Segno

- Rappresentazione interi segno e modulo:
- 1° bit per il segno (1: numeri negativi; 0: numeri positivi)
- Intervallo: $]-b^n/2, b^n/2[$

n=5 ➔ [-15,15]

00000,100000 sono due zeri

15➔01111 -15➔11111

6 ➔00110 -6➔10110

Si opera separatamente su segno e modulo

Operazione Aritmetiche

Rappresentazione modulo e segno

```
-3-1=?
 10011+
 10001=
 ---------
 10100
```

Occorre trattare diversamente numero e segno

```
15+1=?
01111+
00001=
---------
10000
```

Overflow !!!!

La migliore codifica

- Modulo e segno è la migliore codifica?
- Ne esistono altre?
- Quali sono utilizzate nella realtà?

Rappresentazione in complementi alla base

- Una seconda tecnica per la rappresentazione dei numeri relativi consiste nel associare a ciascun numero il suo resto modulo $M=2^n$, definito come:

$$|x|_M = x - [x/M] \times M$$

- Questo tipo di codifica, su n bit, è equivalente ad associare:
  - il numero stesso (cioè X=x), ai numeri positivi compresi tra 0 e $2^{n-1} - 1$;
  - il numero $X = 2^n - |x|$, ai numeri negativi compresi tra 2^{n-1} e – 1;
- I numeri rappresentati sono quelli compresi nell'intervallo

$$[-2^n/2;\ 2^n/\ /2[$$

Complementi alla base

- Intervallo: $[-b^n/2, b^n/2[$
- Un solo zero

n=5

X=10 Occorre codificare: 10➔01010

X=-10 Occorre codificare: b^n-10=22➔ 10110

Proprietà:

- Come complementare il numero
- Rappresentazione di $-b^k$ 1111111000
- Operazioni con le rappresentazioni

La complementazione

- A partire dalla rappresentazione di un numero, è anche particolarmente semplice ottenere la rappresentazione del suo opposto.
  - *Complementare tutti i bit a partire da sinistra, tranne l'uno più a destra ed eventuali zero successivi*.
- E' possibile realizzare le sottrazioni attraverso la composizione di una complementazione (nel senso suddetto) ed un'addizione.
- L'addizionatore e il complementatore rappresentano i componenti fondamentali per la realizzazione di tutte le operazioni.

Esempi di complementazione su 4 bit

- La rappresentazione di 6_{10} su 4 bit è 0110_2.
- Complementando tutti i bit tranne l'uno più a destra e gli zero successivi si ottiene: 1010_2.
- 1010_2 è la rappresentazione di –6 in complementi alla base.

- La rappresentazione di 5_{10} su 4 bit è 0101_2.
- Complementando tutti i bit tranne l'uno più a destra e gli zero successivi si ottiene: 1011_2.
- 1011_2 è la rappresentazione di –5 in complementi alla base.

- La rappresentazione di 1_{10} su 4 bit è 0001_2.
- Complementando tutti i bit tranne l'uno più a destra e gli zero successivi si ottiene: 1111_2.
- 1111_2 è la rappresentazione di –1 in complementi alla base.

Operazione per complementi

Basta sommare le rappresentazioni

-3+1=?
11101+
00001=

11110

Risultato=-2

- Quando si verifica la condizione di overflow?
- Qual è il suo significato?

Rappresentazione in virgola fissa dei numeri reali.

- Quando di un numero frazionario si rappresentano separatamente la parte intera e la parte frazionaria si parla di rappresentazione in *virgola fissa*.
- La rappresentazione dei due contributi (che sono numeri interi) può essere realizzata secondo una delle tecniche viste in precedenza.
- In questo caso la posizione della virgola è fissa e resta sottintesa.

Rappresentazione in virgola mobile dei numeri reali

- Un numero reale x può essere rappresentato dalla coppia

$$(m,e)$$

tale che:

$$x = m \times b^e$$

dove:

  - *m* è detta *mantissa*;
  - *e* è detto *esponente*;
  - *b* è la base di numerazione adottata.
- Il metodo, in questo caso, prende il nome di *codifica in virgola mobile*.

Normalizzazione

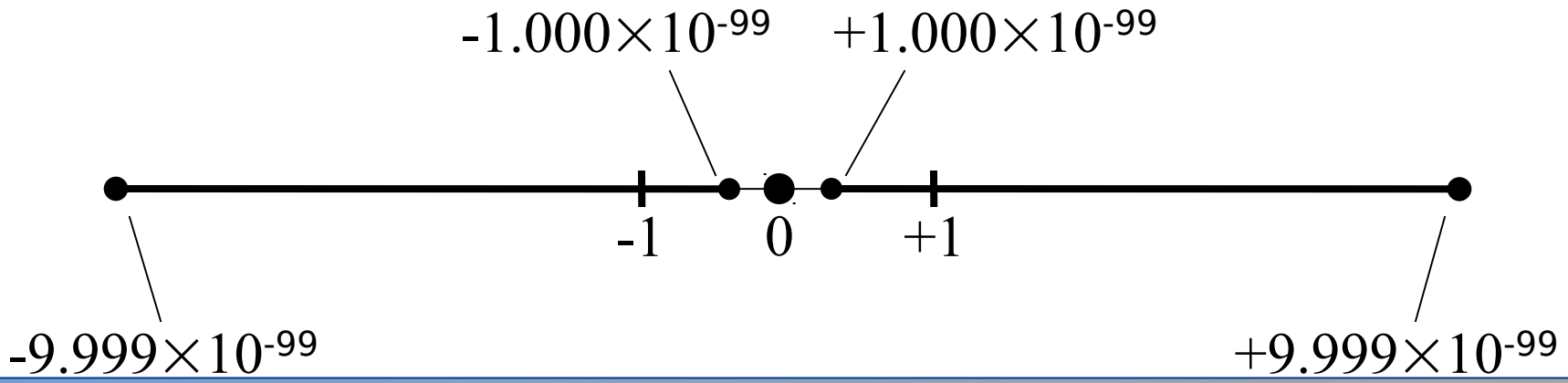
- Per ciascun numero esistono infinite coppie che lo rappresentano.
- Esempio (b=10):
  - 346.09801 è rappresentato da
    - » (346.09801, 0) oppure
    - » (346098.01, -3) oppure
    - » (0.034609801, 4) etc…
- Allo scopo di uniformare le rappresentazioni, si fissa convenzionalmente la posizione della virgola subito dopo la prima cifra significativa, ottenendo l'*unico rappresentante*:

(3.4609801, 2)

Esempio: intervallo di rappresentazione

- Con b=10, usando 4 cifre per *m* e 2 per *e* (più due bit per i relativi segni), l'insieme rappresentabile (utilizzando solo rappresentazioni normalizzate) è:

$$[-9.999\times10^{99},-1.000\times10^{-99}]\cup\{0\}\cup[+1.000\times10^{-99},+9.999\times10^{99}]$$



Approssimazione

- Come è facile verificare, in questo tipo di rappresentazione l'approssimazione non è costante.
- In particolare la precisione assoluta è molto spinta in prossimità dello zero e va diminuendo progressivamente a mano a mano che il numero aumenta (in valore assoluto).
- Ad esempio:
  - in prossimità dello zero l'errore massimo che può essere commesso è pari a $1.001 \times 10^{-99} - 1.000 \times 10^{-99} = 0.001 \times 10^{-99}$;
  - in prossimità dell'estremo superiore dell'intervallo di rappresentazione, invece, l'errore massimo che si può commettere è $9.999 \times 10^{99} - 9.998 \times 10^{99} = 0.001 \times 10^{99}$.
- Si commettono quindi "errori piccoli" su "numeri piccoli" ed "errori grandi" su "numeri grandi".
- Quello che resta inalterato è invece l'errore relativo, costante su tutto l'asse di rappresentabilità.

Overflow e Underflow

- L'errore relativo dipende dal numero di cifre della mantissa.
- Gli estremi dell'intervallo di rappresentazione dipendono dal numero di cifre dell'esponente.

- Nel caso precedente di 2 cifre per l'esponente, si ha overflow per numeri maggiori (in modulo) di 10^{99} e si ha underflow per numeri minori (in modulo) di 10^{-99}.