Insegnamento di Elementi di informatica (6 CFU)

La rappresentazione dei numeri interi

ing. Nadia Ranaldo
Facoltà di Ingegneria
Università degli Studi del Sannio

Materiale realizzato sulla base delle trasparenze redatte dalla prof.sa Antonella Santone

Rappresentazione dei numeri

- Naturali
- Interi
- Reali

Interi

- Includono i naturali, lo zero e tutti i valori negativi della forma -n, dove n un naturale
- Sono detti anche numeri relativi
- La rappresentazione dei numeri interi in un elaboratore pone alcuni problemi:
  - Come rappresentare il "segno meno"
  - Come eseguire le operazioni in modo efficiente

Interi

Opposto $\overline{n} = -n$

$\overline{n} + n = 0$

Complementazione

Operazione che trasforma un numero nel suo opposto

Sottrazione

somma + complementazione

$$n_1 - n_2 = n_1 + \overline{n_2}$$

Varie rappresentazioni

- Modulo & Segno
- Complemento alla base
- …

Osservazione

da ora in poi consideriamo solo **Base = 2**

Modulo & Segno

- Usa un bit per rappresentare esplicitamente il segno

  0 = numero positivo

  1 = numero negativo

- Usa gli altri bit disponibili per rappresentare il modulo

Definizione di modulo

$$|n| = \begin{cases} n & \text{se } n \geq 0 \\ -n & \text{se } n < 0 \end{cases}$$

è un naturale: sappiamo rappresentarlo

Esempio

p = 3 (p = numero di cifre)

Bit + significativo

$-3_{10} = 111_2$

$3_{10} = 011_2$

Dato un numero rappresentato in M&S, che numero intero rappresenta?

$s = c_{p-1} \dots c_0 \rightarrow i?$

Calcoliamo il numero naturale corrispondente alla stringa : $c_{p-2} \dots c_0$

$c_{p-2} \dots c_0 \rightarrow n$ Lo sappiamo fare!!

$$c_{p-1} \dots c_0 = \begin{cases} n & \text{se } c_{p-1} = 0 \\ -n & \text{se } c_{p-1} = 1 \end{cases}$$

Es: $10110_2 = -0110_2 = -(1*2^2 + 1*2^1 + 0*2^0)_{10} = -6_{10}$

Caratteristica

000
001
010
011 positivi

100
101
110
111 negativi

Tabella riassuntiva

Modulo & segno	Interi sistema decimale
000	0
001	1
010	2
011	3
100	-0
101	-1
110	-2
111	-3

p = 3
$[-(2^{p-1}-1) \,..\, 2^{p-1}-1]$

[-3 .. 3]

2 rappresentazioni per lo zero!

Esempio

Data la stringa
1111_2

Se rappresenta un naturale → ?
$1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 = 15$

Se rappresenta un intero in M&S → ?
$-\ 1 \cdot 2^2 + 1 \cdot 2^1 + 1 = -7$

Esercizi

- Rappresentare in M&S il numero intero
  -9

- Data la stringa **11101_2**, che numero intero rappresenta?

Esercizi

1. Rappresentare **–5** e **-8** su **4** bit.

2. Date le seguenti stringhe di bit

   110_2
   110001_2
   0110_2

   Quali interi rappresentano?

Soluzione

–5 = 1101
-8 = 11000 ➔ **non è rappresentabile su 4 bit**

•-----------------------------

110 ➔ **-2**
110001 ➔ **-17**
0110 ➔ **+6**

- Siano date le sequenze di bit:

  A = 1 0 1
  B = 0 1 1
  C = 1 0 1 0

  Sapendo che tali sequenze siano la rappresentazione di tre numeri
  a, b, c, rispettivamente, ricavare tali numeri nei seguenti casi:

  1. **a**, **b**, **c** sono numeri naturali;

  2. **a, b, c** sono numeri interi e la macchina usa la rappresentazione in modulo e segno;

Varie rappresentazioni

- Modulo & segno
- Complemento alla base
- …

Rappresentazione in complemento alla base

La rappresentazione per complemento alla base è una rappresentazione in cui un numero relativo viene rappresentato mediante un numero naturale.

Se il numero n è positivo (non negativo) questo naturale è il modulo *|n|* del numero, altrimenti è *il complemento alla base* del modulo, definito come:

$$\bar{n} = B^p - n$$

dove B è la base prescelta e p sono le cifre a disposizione.

Se B = 2 questa rappresentazione si chiama rappresentazione in complemento a due

Esempio

B = 2

p = 3

$-3_{10} = (2^3-3)_2 = (8-3)_2 = (5)_2$

$= 101$

$(3)_{10} = (3)_2$

$= 011$

- Vale che: $(n)_2 + (\bar{n})_2 = (B^p)_2$

su p+1 cifre

```
1 1 1
  1 0 1 +
  0 1 1 =
  -----
  0 0 0
```


Intervallo di rappresentazione

$$I_{rapp} = [-2^{p-1} \;..\; 2^{p-1} - 1]$$

lo zero ha una sola rappresentazione

Es. con p=4 sono rappresentabili tutti i numeri interi dell'intervallo [-8, +7]

Dato un numero rappresentato in complemento alla base, che numero intero rappresenta?

$s = c_{p-1} \dots c_0$ ➔ i?

Calcoliamo il numero naturale corrispondente alla stringa : $c_{p-1} \dots c_0$ ➔ n

Lo sappiamo fare!!

$$c_{p-1} \dots c_0 = \begin{cases} n & \text{se } c_{p-1} = 0 \\ -(B^p - n) & \text{se } c_{p-1} = 1 \end{cases}$$

Caratteristica

000	positivi
001	
010	
011	
100	negativi
101	
110	
111	

Tabella riassuntiva

Complemento alla base	Interi sistema decimale
100	-4 -(8-4)
101	-3 -(8-5)
110	-2 -(8-6)
111	-1 -(8-7)
000	0
001	1
010	2
011	3

p = 3

$[-2^{p-1} \, .. \, 2^{p-1} - 1]$

[-4 .. 3]

Esercizio

- Quanto vale la stringa binaria 101_2 nei seguenti casi?
  - Naturale
    5
  - Modulo & Segno
    -1
  - Complemento alla base
    -3 (8 – 5)

Operazioni

- Complementazione
- Somma
- Sottrazione
- Moltiplicazione

Complementazione

$n + \bar{n} = 2^p$

$\bar{n} = 2^p - n$

$= (2^p - 1) - n + 1$

$= \boxed{1111\ldots1 \text{ (p volte)}} - n + 1$

•Sottrarre **(2^p-1) – n** significa negare (invertire **0->1** ed **1->0**) tutti i bit di **n** quindi sommare **1**

```
1 1 1 1 1  –
0 1 0 0 1  =
---------
1 0 1 1 0
```

Complementazione:

- si negano i bit
- si somma **1**

Complementazione: esempio

01100

Primo passo: si negano i bit, ottenendo **10011**

Secondo passo: si somma **1**, ottenendo **10100**

Quindi $\overline{01100} = 10100$

Infatti:

01100 ➔ 12

10100 ➔ -(32 -20) = -12

…quindi

- L'opposto di **01100** è **10100**
- Un'altra regola equivalente è:
- La rappresentazione dell'opposto di un numero intero in complemento alla base si ottiene ricopiando gli zeri meno significativi ed il primo uno (meno significativo) e invertendo tutte le altre cifre

0 1 1 0 0

inverto copio

1 0 1 0 0

Altri esempi

$$\begin{array}{ccccc} 0 & 1 & 1 & 1 & 1 \\ \underbrace{1 \; 0 \; 0 \; 0}_{\text{inverto}} & & & & \underbrace{1}_{\text{copio}} \end{array}$$

inverto copio

0 1 1 1 1

1 0 0 0 1

$\overline{01111} = 10001$

inverto copio

0 1 0 0 0

1 1 0 0 0

$\overline{01000} = 11000$

Dato un numero rappresentato in complemento alla base, che numero intero rappresenta?

$111110_2 \rightarrow$ i?

Si può fare in maniera più semplice nel seguente modo:

Complemento: 000010

Calcolo l'intero: 2

Lo nego: -2

Somma

- Indipendentemente dai segni degli operandi si usa lo stesso algoritmo dei naturali pur di troncare il risultato su **p** cifre disponibili. Questo è vero sempre che non vi sia overflow, ovvero supero di capacità, segnalata dalla discordanza tra i riporti in ingresso e in uscita relativi all'ultimo stadio di addizione.

Esempi: B=2, p=4 [-8 .. 7]

```
1 0 0 1                              0 1 1 1
  1 0 0 1 +      - 7 +                 0 1 1 1 +      7 +
  1 0 0 1 =      - 7 =                 0 1 1 1 =      7 =
  -------        -----                 -------      -----
  0 0 1 0          2 NOK               1 1 1 0      - 2 NOK
1 1 1 1                              1 1 1 1
  0 1 1 1 +        7 +                 1 0 1 1 +    - 5 +
  1 0 1 1 =      - 5 =                 1 1 1 1 =    - 1 =
  -------        -----                 -------      -----
  0 0 1 0          2 OK                1 0 1 0      - 6 OK
```

Non basta controllare r_p per determinare l'overflow

CONDIZIONE
Di
OVERFLOW

$$r_p \neq r_{p-1}$$

Sottrazione

B=2, p =4 [-8..7]

```
0 1 0 0 -      4 -
0 1 1 1 =      7 =
---------      ---
    ?          -3
```

somma + complementazione:

$n1 - n2 = n1 + \overline{n2}$

Sottrazione (cont.)

```
0 1 0 0 -      4 -
0 1 1 1 =      7 =
---------      ---
    ?          -
               3
```

Complemento: $\overline{0\ 1\ 1\ 1} = 1\ 0\ 0\ 1$

```
  0 0 0 0
    0 1 0 0 +
    1 0 0 1 =
    -------
    1 1 0 1
```

OK

Moltiplicazione

1. In base ai segni dei fattori, determinare il segno del risultato
2. Complementare gli eventuali fattori negativi
3. Eseguire la moltiplicazione dei naturali così ottenuti
4. Complementare il risultato del passo precedente, se necessario

5. (esistono algoritmi + efficienti)


```
1 0 1 *
0 1 1 =
-------
   ?
```

Complementare il primo fattore

Complementare il risultato

```
    0 1 1 *
    0 1 1 =
    -------
    0 1 1
  0 1 1
  -------
  1 0 0 1
0 0 0
---------
0 1 0 0 1
```

0 0 1 0 0 1

estendere su 6 cifre

e dopo complementare

Risultato:

1 1 0 1 1 1

Tabella riassuntiva

		I_{rapp}	Overflow	Complementaz.
Naturali		[0 .. 2^p-1]	$r_p = 1$	NO
Interi	M&S	[-(2^{p-1}-1) .. 2^{p-1}-1]	$r_{p-1} = 1$	Modificare solo il bit di segno
	CB	[-2^{p-1} .. 2^{p-1}-1] (1 sola rappr. dello zero)	$r_p \neq r_{p-1}$	$\overline{00100}$ = 11100

Un'altra tabella riassuntiva

	Nat	M&S	CBase
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7

	Nat	M&S	CBase
1000	8	-0	-8
1001	9	-1	-7
1010	10	-2	-6
1011	11	-3	-5
1100	12	-4	-4
1101	13	-5	-3
1110	14	-6	-2
1111	15	-7	-1

Esercizio

- Siano date le seguenti sequenze di bit:
  - A = 1001
  - B = 0100
  - C = 0100
- Supponendo che tali sequenze siano la rappresentazione di tre numeri *a*, *b* e *c*, ricavare tali numeri nei seguenti due casi:
  - *a*, *b* e *c* sono numeri naturali;
  - *a*, *b* e *c* sono numeri interi e la macchina usa la rappresentazione in complemento a due.
- Sapendo di avere a disposizione solo 4 bit dire, motivando la risposta, se nei due casi sopra citati, è vera la seguente affermazione:

  (A + B) + C = A + (B + C)

Soluzione: naturali

a = 9   b = 4   c = 4   OVERFLOW

Somma

(A + B) + C

```
0 0 0 0          1 1 0 0
  1 0 0 1 +        1 1 0 1 +
  0 1 0 0 =        0 1 0 0 =
  1 1 0 1          0 0 0 1
```

A + (B + C)

```
0 1 0 0          1 0 0 0
  0 1 0 0 +        1 0 0 0 +
  0 1 0 0 =        1 0 0 1 =
  1 0 0 0          0 0 0 1
```

(A + B) + C =
A + (B + C)

Soluzione: interi

a = -7   b = 4   c = 4

Somma

(A + B) + C

```
 0 0 0 0          1 1 0 0
   1 0 0 1 +        1 1 0 1 +
   0 1 0 0 =        0 1 0 0 =
   -------          -------
   1 1 0 1          0 0 0 1
```

OK

A + (B + C)

```
 0 1 0 0
   0 1 0 0 +
   0 1 0 0 =
   -------
   1 0 0 0
```

OVERFLOW

(A + B) + C ≠ A + (B + C)

Esercizio

- Siano date le seguenti sequenze di bit:
  - A = 10110001
  - B = 1100
  - C = 0110
- Sapendo che tali sequenze rappresentano tre numeri interi *a*, *b* e *c*, rispettivamente e che la macchina usa la rappresentazione in complemento alla base:
  - ricavare *a*, *b* e *c*;
  - eseguire, in aritmetica binaria, l'operazione

    A – (B*C)