Rappresentazione dei numeri

Antonella Santone

Anno Accademico 2008/2009

Rappresentazione dei numeri

- ❖ Naturali
- ❖ Interi
- ❖ Reali

Interi

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

L'esecuzione delle operazioni aritmetiche è un'operazione frequentissima: se fosse inefficiente l'efficienza complessiva della macchina ne risentirebbe.

Interi (cont.)

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
- ❖ …

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

Rappresentazione in M&S

$$\rho_B(n) = \begin{cases} <0,\ \rho_B(|n|)> & \text{se } n \geq 0 \\ <1,\ \rho_B(|n|)> & \text{se } n < 0 \end{cases}$$

bit di segno

Osservazione

Da ora in poi consideriamo solo B=2

Esempio

$p = 3$

$\rho_2(-3) = <1,\ \rho_2(3)> = <1,11> = 111_2$

$\rho_2(3) = <0,\ \rho_2(3)> = <0,11> = 011_2$

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

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

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

$c_{p-2} \dots c_o \rightarrow n$

Lo sappiamo fare!!

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

Caratteristica

000
001 positivi
010
011

100
101 negativi
110
111

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

Naturale ➔

$1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 = 15$

Intero rappresentato in M&S ➔

$-(1 \cdot 2^2 + 1 \cdot 2^1 + 1) = -7$

Esercizio

Rappresentare in M&S il numero intero -9

Data la stringa 11101_2, che numero intero rappresenta?

Soluzione

$-9_{10} = 11001$

$11101_2 = -13$

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ Moltiplicazione
- ❖ Divisione

Complementazione

$$s = c_{p-1}\, c_{p-2} \cdots c_o\ {}_2$$

$$\overline{s} = \begin{cases} 0\ c_{p-2} \cdots c_o\ {}_2 & \text{se } c_{p-1} = 1 \\ 1\ c_{p-2} \cdots c_o\ {}_2 & \text{se } c_{p-1} = 0 \end{cases}$$

Basta prendere il bit + significativo e:

- se è uno diventa zero;
- se è zero diventa uno.

Operazioni

- Complementazione
- **Somma**
- Sottrazione
- Moltiplicazione
- Divisione

Somma informale

Siano `i1` e `i2` due interi:

se hanno lo stesso segno:
la somma avrà lo stesso segno e come modulo, la somma dei moduli (es. −6−5 = −11)

se hanno segno diverso e moduli diversi:
la somma avrà il segno del modulo maggiore e come modulo la differenza dei moduli (es. −6+5 = −1)

se hanno segno opposto e modulo uguale:
la somma avrà come segno + e come modulo 0
(es. −5+5 = +0)

Somma formale

$$s = \underbrace{c_{p-1}}_{seg}\ \underbrace{c_{p-2}\ \ldots\ c_{o}}_{n}\ {}_{2}$$

$$s' = \underbrace{c'_{p-1}}_{seg'}\ \underbrace{c'_{p-2}\ \ldots\ c'_{o}}_{n'}\ {}_{2}$$

Sappiamo fare:
numeri naturali

$$s+s' = \begin{cases} <seg,\ n+n'> & \text{se } seg = seg' \\ <seg,\ n-n'> & \text{se } seg \neq seg' \text{ e } n > n' \\ <seg',\ n'-n> & \text{se } seg \neq seg' \text{ e } n' > n \\ <+,\ 0> & \text{se } seg \neq seg' \text{ e } n' = n \end{cases}$$

Analisi

- Per fare la somma serve un sottrattore
- Il sommatore lavora solo su `p-1` bit e se il riporto $r_{p-1} = 1$ ➔ overflow

---

Esempi

`p = 4` `[-(2`$^{p-1}$`-1) .. 2`$^{p-1}$` -1]` `[-7..7]`

1	1	0	0	+
1	0	1	1	=
		?		

lavorare sui moduli:
in questo caso fare la somma
(stesso segno)

no overflow

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

In decimale

-4 - 3 = -7

risultato

1 1 1 1

bit di segno

Esempi (cont.)

`p = 4` `[-7..7]`

1	1	0	0	+
1	1	0	0	=
		?		

lavorare sui moduli:
in questo caso fare la somma
(stesso segno)

si overflow

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

la somma non è rappresentabile su 4 bit

Esempi (cont.)

p = 4

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

lavorare sui moduli:

in questo caso fare la sottrazione

(segno diverso)

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

risultato

0 0 1 0

bit di segno =
bit dell'intero con modulo maggiore

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ Moltiplicazione
- ❖ Divisione

Sottrazione

In modo analogo alla somma

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ Moltiplicazione
- ❖ Divisione

Moltiplicazione e Divisione

Lavorare sui moduli e studiare separatamente il segno:

Operandi concordi

- \-
\+ +

➡ segno: 0

Operandi discordi

\- +
\+ -

➡ segno: 1

Esempio

p = 3

risultato su p = 6

lavorare sui moduli

Operandi discordi:

Segno: 1

risultato

1 0 0 1 1 0

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

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

estendere con zero, perché risultato deve essere su 6

Vantaggi e svantaggi del M&S

Vantaggio

coincide con la nostra usuale rappresentazione

Svantaggio

richiede il trattamento separato di segno e modulo: algoritmi aritmetici più pesanti ...

... nei calcolatori, per ovviare agli svantaggi dell'aritmetica della rappresentazione in segno e modulo, si adottano altre rappresentazioni ...

Esercizi

... modulo e segno

Esercizio

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 0 1 0 1 0`
`B = 0 1 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;

Dire, nei due casi sopra citati, l'esito dell'operazione:

`(A / B) + C`

sapendo di avere a disposizione solo 4 bit per il risultato e per il quoziente della divisione.

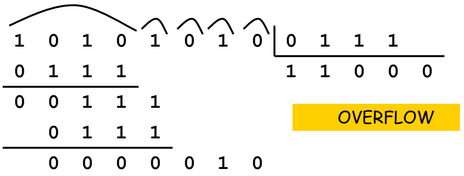
Soluzione: naturali

a = 170

b = 7

c = 10

`(A / B) + C`



```
1 0 1 0 1 0 1 0 | 0 1 1 1
0 1 1 1         |__________
-------           1 1 0 0 0
0 0 1 1 1
  0 1 1 1
---------
  0 0 0 0 0 1 0
```

OVERFLOW

Soluzione: M&S

`a = -42`
`b = 7`
`c = -2`

`(A / B)`

lavorare sui moduli

Operandi discordi:

Segno: 1

Soluzione: M&S

`(A / B)`

```
7 bit   0  1  0  1  0  1  0 | 1  1  1    3 bit
              1  1  1       | 1  1  0
        0  0  0  1  1  1
                 1  1  1
                 0  0  0  0
```

NO OVERFLOW

(A / B) = 1 1 1 0

bit di segno

Soluzione: M&S

`(A / B) + C`

`(A / B) = 1110`
`C = 1010`

lavorare sui moduli:
in questo caso fare la somma
(stesso segno)

OVERFLOW

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

Varie rappresentazioni

- Modulo & segno
- Complemento alla base
- …

Rappresentazione in complemento alla base

$$\rho_B(n) = \begin{cases} \rho_B(|n|) & \text{se } n \geq 0 \\ \rho_B(B^p - |n|)> & \text{se } n < 0 \end{cases}$$

L'opposto di n è $B^p - n$:

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

Esempio

$B = 2$
$p = 3$

$\rho_B(-3) = \rho_B(2^3-3) = \rho_B(8-3) = \rho_B(5)$
$= 101$

$\rho_B(3) = \rho_B(3)$
$= 011$

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

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

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

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

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

Lo sappiamo fare!!

$$c_{p-1} \dots c_o = \begin{cases} n & \text{se } c_{p-1} = 0 \text{ (oppure se } n \in I_{rapp}) \\ -(B^p - n) & \text{se } c_{p-1} = 1 \text{ (oppure se } n \notin I_{rapp}) \end{cases}$$

Caratteristica

000
001 positivi
010
011

100
101 negativi
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

$p = 3$

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

$[-4 \;..\; 3]$

una rappresentazione per lo zero → asimmetria dell'intervallo

(tanti neg quanti pos + una rappresentazione per lo zero in contrasto con 2^p che è pari)

Propagazione del segno

$110 \rightarrow -(2^3 - 6) = -2$

$-2 = 110$

$2 = 010$

Come si rappresentano su 5 bit??

Basta propagare il segno:

$-2 = 11110 = -(2^5 - 30)$

$2 = 00010$

101?

Naturale

5

Modulo & Segno

-1

Complemento alla base

-3 -(8 - 5)

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ Moltiplicazione
- ❖ Divisione

Complementazione

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

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

$$= \boxed{2^p - 1} - n + 1$$

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

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:

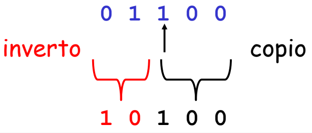
01100 ➔ 12
10100 ➔ -(32 -20) = -12

...quindi

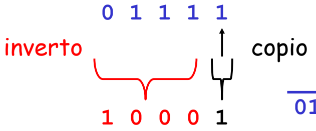
L'opposto di 01100 è 10100

Un'altra regola equivalente è:

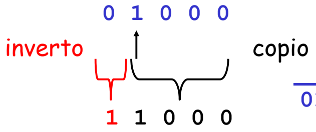
La rappresentazione dell'opposto di un numero intero in complemento alla base si ottiene ricopiando gli zeri meno significativi ed il primo uno (meno significativo) e invertendo tutte le altre cifre



Altri esempi



$\overline{01111} = 10001$



$\overline{01000} = 11000$

Esempio

p = 4 [-8 .. 7]

- ❖ $\overline{1010} = 0110$ vale che
  1010 +
  0110 =
  0000 (con riporto di 1)
- ❖ $\overline{1000} = 1000$ **autocomplementate**: dà luogo ad overflow se si cerca di complementare
  -8

Ora … più semplicemente

①11110 → i?

Si può fare:

Complemento: 000010
Calcolo l'intero: 2
Lo nego: -2

Operazioni

- Complementazione
- **Somma**
- Sottrazione
- Moltiplicazione
- Divisione

Somma

Indipendentemente dai segni degli operandi si usa lo stesso algoritmo dei naturali pur di troncare il risultato su p cifre disponibili → si dice modulo B^p

---

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

$$\begin{array}{r} \boxed{1\ 0}\ 0\ 1\\ 1\ 0\ 0\ 1\ +\\ 1\ 0\ 0\ 1\ =\\ \hline 0\ 0\ 1\ 0 \end{array} \qquad \begin{array}{r} \\ -\ 7\ +\\ -\ 7\ =\\ \hline 2\ \text{NOK} \end{array} \qquad \begin{array}{r} \boxed{0\ 1}\ 1\ 1\\ 0\ 1\ 1\ 1\ +\\ 0\ 1\ 1\ 1\ =\\ \hline 1\ 1\ 1\ 0 \end{array} \qquad \begin{array}{r} \\ 7\ +\\ 7\ =\\ \hline -\ 2\ \text{NOK} \end{array}$$

$$\begin{array}{r} \boxed{1\ 1}\ 1\ 1\\ 0\ 1\ 1\ 1\ +\\ 1\ 0\ 1\ 1\ =\\ \hline 0\ 0\ 1\ 0 \end{array} \qquad \begin{array}{r} \\ 7\ +\\ -\ 5\ =\\ \hline 2\ \text{OK} \end{array} \qquad \begin{array}{r} \boxed{1\ 1}\ 1\ 1\\ 1\ 0\ 1\ 1\ +\\ 1\ 1\ 1\ 1\ =\\ \hline 1\ 0\ 1\ 0 \end{array} \qquad \begin{array}{r} \\ -\ 5\ +\\ -\ 1\ =\\ \hline -\ 6\ \text{OK} \end{array}$$

Non basta controllare r_p per determinare l'overflow

CONDIZIONE
Di
OVERFLOW

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

Somma

Si può usare il sommatore visto per i naturali e controllare gli ultimi riporti per vedere se c'è overflow

---

Operazioni

- Complementazione
- Somma
- **Sottrazione**
- Moltiplicazione
- Divisione

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

```
$\boxed{0\ 0}$ 0  0
```

$\boxed{0\quad 0}\quad 0\quad 0$

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

OK

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ **Moltiplicazione**
- ❖ Divisione

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

(esistono algoritmi + efficienti)

Esempio

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

Complementare il risultato

Complementare il primo fattore

```
      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

Operazioni

- ❖ Complementazione
- ❖ Somma
- ❖ Sottrazione
- ❖ Moltiplicazione
- ❖ **Divisione**

Divisione

Simile alla moltiplicazione

Esempio

1 1 1 0 1 0 | 0 1 1

Complementare il risultato

Complementare il dividendo

0 0 0 1 1 0 | 0 1 1

0 1 1   1 0

0 0 0 0

0 1 0

estendere su 3 cifre

e dopo complementare

Risultato:

1 1 0

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

Esercizi

… home page

Esercizio

Sia data la seguente sequenza di bit:

A = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1

Se tale sequenza è la rappresentazione di un numero intero e la macchina usa la rappresentazione in complemento alla base, ricavare tale numero.

Su tre bit `101 = -3`

Per la propagazione del segno `A = -3`

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 alla base.

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`

$$\begin{array}{cccccc} \boxed{0} & 0 & 0 & 0 & & \\ & 1 & 0 & 0 & 1 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 1 & 1 & 0 & 1 & \end{array} \qquad \begin{array}{cccccc} \boxed{1} & 1 & 0 & 0 & & \\ & 1 & 1 & 0 & 1 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 0 & 0 & 0 & 1 & \end{array}$$

`A + (B + C)`

$$\begin{array}{cccccc} \boxed{0} & 1 & 0 & 0 & & \\ & 0 & 1 & 0 & 0 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 1 & 0 & 0 & 0 & \end{array} \qquad \begin{array}{cccccc} \boxed{1} & 0 & 0 & 0 & & \\ & 1 & 0 & 0 & 0 & + \\ & 1 & 0 & 0 & 1 & = \\ \hline & 0 & 0 & 0 & 1 & \end{array}$$

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

Soluzione: interi

`a = -7` `b = 4` `c = 4`

Somma

`(A + B) + C`

$$\begin{array}{cccccc} \multicolumn{2}{c}{\boxed{0 \quad 0}} & 0 & 0 & & \\ & 1 & 0 & 0 & 1 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 1 & 1 & 0 & 1 & \end{array} \qquad \begin{array}{cccccc} \multicolumn{2}{c}{\boxed{1 \quad 1}} & 0 & 0 & & \\ & 1 & 1 & 0 & 1 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 0 & 0 & 0 & 1 & \end{array}$$

OK

`A + (B + C)`

$$\begin{array}{cccccc} \multicolumn{2}{c}{\boxed{0 \quad 1}} & 0 & 0 & & \\ & 0 & 1 & 0 & 0 & + \\ & 0 & 1 & 0 & 0 & = \\ \hline & 1 & 0 & 0 & 0 & \end{array}$$

OVERFLOW

`(A + B) + C ≠`
`A + (B + C)`

Osservazione

Rappresentazione finita:

proprietà associativa non vale

Siano date, in base due, le seguenti sequenze di bit:

Esercizio

`A = 00110101`
`B = 0110`
`C = 0111`

Supponendo che tali sequenze siano la rappresentazione di tre numeri **a**, **b** e **c**, ricavare tali numeri nei seguenti due casi:

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

Eseguire, solo nel secondo caso sopra citato, l'operazione:

`B-(-A/-C)`

sapendo di avere a disposizione solo 4 bit per il risultato e per il quoziente della divisione.

Soluzione

Naturali & interi

`a = 53`
`b = 6`
`c = 7`

Soluzione (cont.)

Operazione: `B-(-A/-C)` in modulo e segno

`-A/-C`: lavoriamo sui moduli

```
0 1 1 0 1 0 1 | 1 1 1
    1 1 1       -----
    -----       1 1 1
    1 1 0 0
      1 1 1
      -----
      1 0 1 1
        1 1 1
        -----
        1 0 0
```

`-A/-C = R = 0111`

`B-R`: lavoriamo sui moduli, `|R|>|B|` allora dobbiamo eseguire `R-B` ed il risultato avrà segno negativo:

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

`B-(-A/-C) = 1001`

Non si è verificato overflow

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

`B – (A/C)`

sapendo di avere a disposizione solo 4 bit per il risultato e per il quoziente della divisione.

Soluzione

Il bit più significativo di **A** è 1, quindi *a* è negativo.

$\overline{\mathtt{A}}$ = 01001111

$a = -\ (2^6 + 2^3 + 2^2 + 2^1 + 2^0\ ) = -79$

Il bit più significativo di **B** è 1, quindi *b* è negativo.

$\overline{\mathtt{B}}$ = 0100

$b = -\ 2^2 = -4$

Il bit più significativo di **C** è 0, quindi *c* è positivo,

$c = 2^2 + 2^1 = 6$

Soluzione (cont.)

Per calcolare `(A/C)`, calcoliamo $\overline{\mathtt{A}}$`/C = 1101`

```
0 1 0 0 1 1 1 1 | 0 1 1 0
    1 1 0         -------
    -----         1 1 0 1
      1 1 1
      1 1 0
      -----
      0 0 1 1 1
          1 1 0
          -----
          0 0 1
```

Il risultato non è rappresentabile su 4 bit in complemento alla base (il quoziente deve essere positivo, mentre il primo bit è negativo).