Rappresentazione dei numeri interi

Prof. Daniele Gorla

Rappresentazione degli Interi

Rispetto ai naturali, il problema è la rappresentazione del segno
Esistono tre modalità di rappresentazione:

- in **modulo e segno**
- in **complemento a uno**
- in **complemento a due**.

I primi due metodi rendono le operazioni di somma e sottrazione delicate (sono necessari controlli preliminari sul segno e sui valori assoluti degli operandi)

Col terzo, invece, la somma è immediata e la sottrazione si esegue semplicemente come somma dell'opposto (a patto di ignorare l'eventuale overflow derivante dalla somma di numeri negativi).

Rappresentazione in Complemento a 2 (Ca2)

La sequenza di cifre $c_{n-1}\ldots c_1 c_0$ nella notazione in complemento alla base b è l'intero N dato dalla seguente espressione:

$$-c_{n-1}b^{n-1} + \sum_{i=0}^{n-2} c_i b^i, c_i \in \{0, \ldots, b-1\}$$

N.B.: nella rappresentazione in complemento alla base è *fondamentale* sapere la lunghezza della codifica

Es.: 1101 come numero in Ca2 da 4 bit è $-2^3+2^2+1 = -3$
come numero in Ca2 da 5 bit è $2^3+2^2+1 = 13$

OSS: la cifra più significativa è un indicatore di segno:

- Se è 0, il numero è non-negativo (sommo solo quantità non-negative);
- altrimenti, il numero è negativo $\left(b^{n-1} > \sum_{i=0}^{n-2} c_i b^i\right)$

Esempio

Diciamo di lavorare in Ca2 con numeri da 8 bit

Considero 11111101

Applicando la formula, si ha:

11111101 $= -2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0$
$= -128+64+32+16+8+4+1$
$= -128+125 = -3$

Intervallo di rappresentabilità in Ca2

Il numero più grande ha il primo bit 0 e tutti gli altri 1
Il numero più piccolo ha il primo bit 1 e tutti gli altri 0

8 bit (Ca2)

- $01111111 = 2^7 - 1 = +127$
- $10000000 = -2^7 = -128$

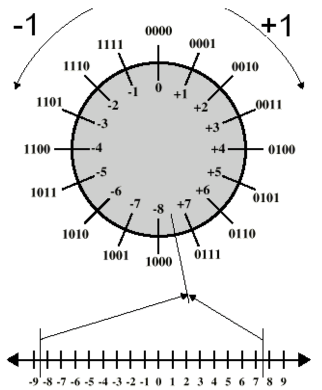
In generale, con n bit (Ca2)

$\underbrace{0111\ldots1}_{n\text{-}1} = 2^{n-1} - 1$ $\qquad$ $1\underbrace{000\ldots0}_{n\text{-}1} = -2^{n-1}$

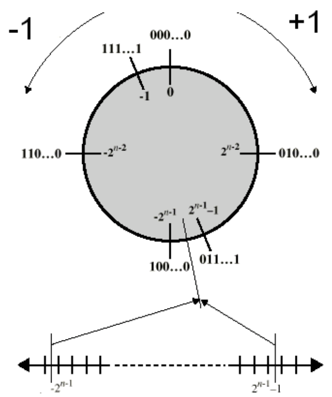
OSS.: l'intervallo di rappresentabilità non è simmetrico, nel senso che 10…0 non ha l'opposto

→ spesso si esclude 10...0 e si rappresenta l'intervallo $\{-2^{n-1}+1, \ldots, 2^{n-1}-1\}$

Descrizione geometrica della rappresentazione in Ca2 a 4 bit



Descrizione geometrica della rappresentazione in Ca2 a *n* bit



Procedimento per trovare l'opposto

Dato N in Ca2, il suo opposto si trova

- **complementandone i bit**
- **sommando 1** al numero risultante

Esempi (con rappresentazione a 4 bit):

l'opposto di 1101 ($= -2^3+2^2+1 = -3$) è
$0010+1 = 0011$ ($=2^1+1=3$)

l'opposto di 0101 ($= 2^2+1 = 5$) è
$1010+1 = 1011$ ($=-2^3+2^1+1=-5$)

Dimostrazione

SAPIENZA Università di Roma Dipartimento di Informatica

Siano $N = -c_{n-1}b^{n-1} + \sum_{i=0}^{n-2} c_i b^i$ e $N' = -\overline{c_{n-1}}b^{n-1} + \sum_{i=0}^{n-2} \overline{c_i} b^i + 1$

Dimostro che N ed N' sono opposti, cioè $N + N' = 0$

$$N + N' = \left(-c_{n-1}b^{n-1} + \sum_{i=0}^{n-2} c_i b^i\right) + \left(-\overline{c_{n-1}}b^{n-1} + \sum_{i=0}^{n-2} \overline{c_i} b^i + 1\right)$$

$$= -\left(c_{n-1} + \overline{c_{n-1}}\right)b^{n-1} + \sum_{i=0}^{n-2}\left(c_i + \overline{c_i}\right)b^i + 1$$

$$= -b^{n-1} + \sum_{i=0}^{n-2} b^i + 1 = -b^{n-1} + \left(b^{n-1} - 1\right) + 1 = 0$$

9

Conversione di interi da base 10

SAPIENZA Università di Roma Dipartimento di Informatica

Trasformare un intero N da base 10 a base 2 in Ca2 con n bit

Se $N \geq 0$, applica il metodo delle divisioni iterate
- Se servono meno di n bit, allora il numero è rappresentabile e la codifica inserirà 0 in testa, fino ad arrivare a n bit
- altrimenti il numero non è rappresentabile nel formato utilizzato

Se $N < 0$, applica il metodo delle divisioni iterate a $-N$
- Se servono meno di n bit, allora il numero è rappresentabile
  - la codifica inserirà 0 in testa, fino ad arrivare a n bit
  - calcola l'opposto del numero ottenuto
- altrimenti il numero non è rappresentabile nel formato utilizzato

10

Esempi

SAPIENZA Università di Roma Dipartimento di Informatica

Formato: Ca2 con 4 bit
Rappresentare:

9: la codifica di 9 è 1001, che richiede 4 bit $\rightarrow$ NON RAPPR.

5: la codifica di 5 è 101, quindi in Ca2 è 0101

-3: la codifica di 3 è 11, che in Ca2 è 0011. Il suo opposto è 1100+1=1101

-9: la codifica di 9 è 1001, che richiede 4 bit $\rightarrow$ NON RAPPR.

OSS.: -8 è rappresentabile come 1000, ma il suo opposto sarebbe 0111+1=1000. Quindi, tipicamente -8 viene considerato non rappresentabile in questo formato.

11

Artimetica dei numeri interi in Ca2

SAPIENZA Università di Roma Dipartimento di Informatica

La somma si esegue come con i naturali
(l'unica differenza sono le condizioni di overflow)

La differenza m-s si esegue come la somma tra m e l'opposto di s (cioè, m-s = m+(-s))

Moltiplicazione e divisione seguono di conseguenza

12

Somma in complemento a 2

SAPIENZA
Università di Roma
Dipartimento di Informatica

Lavoriamo in Ca2 con 8 bit e facciamo

6+8 6-8 -6+8 -6-8

6 in Ca2 è 00000110
-6 in Ca2 è 11111001+1=11111010
8 in Ca2 è 00001000
-8 in Ca2 è 11110111+1=11111000

C'è un riporto finale!!

6+8	6-8	-6+8	-6-8
00000110+	00000110+	11111010+	11111010+
00001000=	11111000=	00001000=	11111000=
00001110	11111110	00000010	11110010
(14_{10})	(-2_{10})	(2_{10})	(-14_{10})

Ma il risultato è rappresentabile

Condizione di overflow in Ca2

SAPIENZA
Università di Roma
Dipartimento di Informatica

Quindi, la condizione di overflow in Ca2 non è il semplice riporto alla fine della somma (come lo era per i naturali)

Lavoriamo in Ca2 con 4 bit ed eseguiamo

- 7+2: 0111+0010=1001 (cioè -7)
- -7-2: 1001+1110=0111 (cioè 7)

→ ***Condizione: operandi concordi e risultato discorde***

(N.B.: il segno di un numero è dato dal MBS)

Se inoltre assumiamo che la codifica 1000 non è ammessa (perché non ha un opposto rappresentabile), allora ogni operazione che produce tale sequenza genera un overflow.