

# Rappresentazione binaria di valori negativi

leo

March 5, 2023

Ci sono 4 modi per rappresentari numeri binari negativi. *Si suppone di avere a disposizione 1B.*

## 1 Modulo e Segno

Utilizzando il metodo modulo e segno utilizzeremo il MSB per indicare il segno ( $0$ : +,  $1$ : -) ed i rimanenti 7b per rappresentare il numero in valore assoluto.

Con questo metodo si possono rappresentare i numeri **interi** nell'intervallo:  $[-(2^{n-1} - 1), (2^{n-1} - 1)]$ . *Viene sottratto 1 all'esponente perchè un bit a disposizione viene utilizzato per il segno.*

Il problema di utilizzare questo metodo è che 0 può essere sia:

- $0000_2$ : +0
- $1000_2$ : -0

### 1.1 Somma aritmetica

Se il segno del risultato è diverso dai 2 numeri c'è un **overflow**.

Si confrontano i MSB:

- se i bit sono di segno **uguale**:
  - il bit di segno rimane invariato e si esegue la somma normalmente
- se i bit sono di segno **diverso**:
  - si confronta il valore assoluto dei 2 numeri
  - il segno risultante sarà il segno del più grande in valore assoluto
  - si esegue la sottrazione normalmente

## 1.2 Sottrazione aritmetica

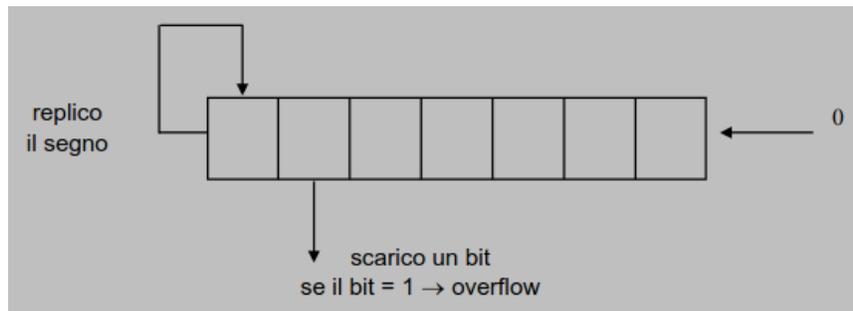
Se il segno del risultato è diverso dai 2 numeri c'è un **overflow**.

Si confrontano i MSB:

- se i bit sono di segno **uguale**:
  - si confronta il valore assoluto dei 2 numeri
  - il segno risultante sarà il segno del più grande in valore assoluto
  - il risultato sarà il valore assoluto della sottrazione dei valori assoluti dei 2 numeri
- se i bit sono di segno **diverso**:
  - il segno risultante sarà il segno del primo numero
  - si esegue la somma dei valori assoluti

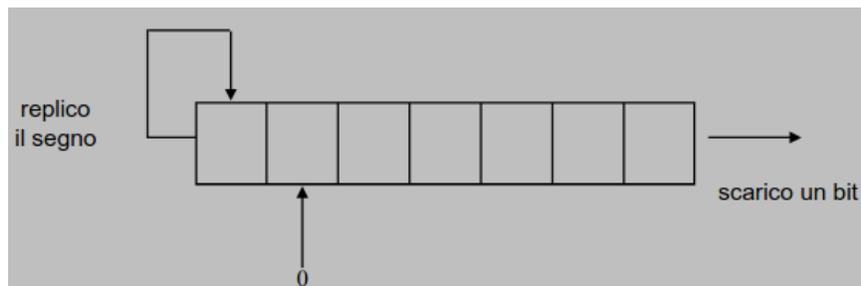
## 1.3 Shift left

Equivale a **moltiplicare** il numero per la base.



## 1.4 Shift right

Equivale a **dividere** il numero per la base.



## 2 Complemento a 1 (CA1)

Questo metodo si basa sull'operazione di complemento.

- +3:  $0011_2$
- -3:  $1100_2$

Utilizzando questo metodo sussiste il problema delle 2 diverse rappresentazione dello 0:

- $0000_2$ : +0
- $1111_2$ : -0

## 3 Complemento a 2 (CA2)

Se il numero da codificare è positivo viene utilizzata la normale rappresentazione binaria. Se il numero da codificare è negativo:

- Si effettua il complemento a 1
- Si somma +1 al risultato

Come per la rappresentazione con modulo e segno i numeri negativi hanno MSB a 1. Questo metodo non ha il problema della doppia rappresentazione dello 0.

Per effettuare operazioni aritmetiche gli operandi **devono** avere lo stesso numero di bit. Per far sì che gli operandi abbiano lo stesso numero di bit si replica il MSB fino ad ottenere una rappresentazione di lunghezza desiderata.

### 3.1 Somma aritmetica

L'overflow si presenta quando si sommano 2 numeri con lo stesso segno e si ottiene un risultato di segno opposto

### 3.2 Sottrazione aritmetica

La sottrazione tra 2 numeri in CA2 viene trasformata in somma applicando:

$$A - B = A + \text{CA2}(B)$$

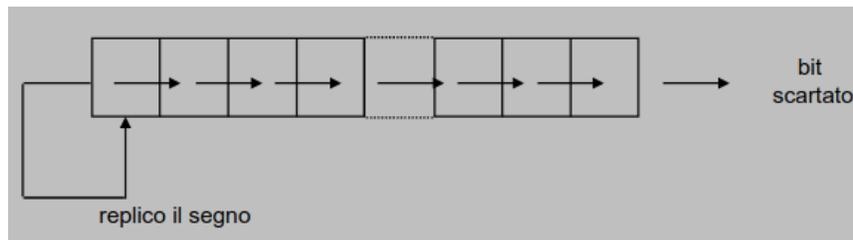
### 3.3 Shift left

Equivale a **moltiplicare** il numero per la base.



### 3.4 Shift right

Equivale a **dividere** il numero per la base.



## 4 Eccesso $2^{n-1}$

*I numeri in eccesso  $2^{n-1}$  si ottengono da quelli in CA2 complementando il bit più significativo. (0: -, 1: +)*

Nella rappresentazione eccesso  $2^{n-1}$  un numero generico  $X$  viene rappresentato come  $X + 2^{n-1}$  dove  $n$  è il numero di bit disponibili.

*Esempio 8bit:*  $5 = 5 + 128 = 1000\ 0101_2$

*Esempio 8bit:*  $-3 = -3 + 128 = 0111\ 1101_2$