

# Abilità Informatiche

Luigi Catuogno

[luigi.catuogno@uniparthenope.it]

*Corso di Laurea in Economia e Commercio - Anno Accademico 2022-23*

1

Libro di testo

**[IdB]**

Dennis P. Curtin, Kim Foley, Kunal Sen, Cathleen Morin

**Informatica di base**

VII edizione (2016), MacGraw Hill Education

ISBN: 978-88-386-1537-5

2

Altro materiale di utile consultazione

**[Sli]**

Slides, appunti e altro materiale distribuito dal docente

**[Misc]**

Altra fonte diversamente specificata di volta in volta

3

Codifica binaria dei numeri interi

4

## Rappresentare gli interi

- In un numero rappresentato in binario con  $l$  cifre si possono rappresentare  $2^l$  valori diversi
  - Da 0 a  $(2^l - 1)$

### Esempio

Un numero binario composto da  $l = 8$  cifre può rappresentare  $2^l = 2^8 = 256$  valori diversi compresi in  $[0, 255]$  cioè  $[00000000_2, 11111111_2]$

5

## LSB e MSB

- In un numero binario composto da  $l$  cifre, queste sono numerate da sinistra a destra da  $b_{l-1}$  a  $b_0$ :



- Il **Least Significant Bit (LSB)** è il bit più a destra (posizione 0)
- Il **Most Significant Bit (MSB)** è il più a sinistra (posizione  $l - 1$ )

### Esempio

Un numero binario composto da  $l = 8$  cifre **LSB** è  $b_0$  e **MSB** è  $b_7$

6

## Fino a ora...

- Abbiamo trattato numeri binari con una precisione *infinita*
  - Abbiamo sempre avuto la possibilità di utilizzare il numero di cifre che ci occorre in base alle grandezze coinvolte nei calcoli
- Abbiamo trattato numeri interi positivi
  - Il segno algebrico positivo restava *sottinteso*
  - Tuttavia:
    - Dal punto di vista teorico, il segno algebrico ha lo stesso significato anche nel sistema binario
    - Infatti nulla impedisce di rappresentare  $-14_{10}$  con  $-1110_2$

7

## Tuttavia...

- In un calcolatore elettronico:
  - Il numero di cifre binarie che è possibile utilizzare è *limitato* e *costante*;
  - Frequentemente è necessario rendere il segno algebrico *esplicito*;
  - Ma...

*...come rappresentarlo?*

8

## Codifica binaria dei numeri interi

- Un numero intero con segno, «porta con sé» due informazioni: il **segno** e il **modulo** (o *valore assoluto*)
  - Nei calcolatori elettronici, non esiste una rappresentazione *ad hoc* per il segno algebrico.
  - Entrambe le informazioni devono essere rappresentata nella stessa sequenza di cifre binarie (bit)

9

## Codifica binarie dei numeri interi

- Il numero di interi distinti rappresentabili dipende solo dal numero delle cifre a disposizione.
- Avendo a disposizione non più di  $2^l$  possibili configurazioni di cifre binarie, dobbiamo stabilire una *convenzione* che ci permetta di assegnare a ciascuna di esse un valore positivo o negativo.

10

## Codifica binaria dei numeri interi

- Un numero **naturale** binario di  $l$  cifre può rappresentare qualsiasi valore intero non negativo compreso nell'intervallo  $[0, 2^l - 1]$ , cioè:
  - lo zero
  - $2^l - 1$  valori maggiori di zero;

11

## Codifica binaria dei interi

- Un numero **naturale** binario di  $l$  cifre può rappresentare qualsiasi valore intero non negativo compreso nell'intervallo  $[0, 2^l - 1]$ , cioè:
  - lo zero
  - $2^l - 1$  valori maggiori di zero;
- Una codifica binaria di interi *con segno* su  $l$  cifre deve comprendere:
  - Almeno **un** modo di rappresentare lo **zero**
  - La rappresentazione di al più  $2^l - 1$  valori diversi da **zero**, tra negativi e positivi...

12

## Codifica binaria dei numeri interi

- Una codifica binaria di interi *con segno* su  $l$  cifre deve comprendere:
  - Almeno **un** modo di rappresentare lo **zero**
  - La rappresentazione di al più  $2^l - 1$  valori diversi da **zero**, tra negativi e positivi...

### Esempio

Avendo a disposizione  $l = 8$  cifre è possibile rappresentare gli interi con segno compresi nell'intervallo  $[-127, +127]$

*oppure compresi nell'intervallo  $[-128, +127]$*

*...o magari compresi nell'intervallo  $[-127, +128]$*

13

## Codifica binaria dei numeri interi

Ci sono diversi metodi per rappresentare gli interi con segno utilizzando numeri binari. La differenza tra l'uno e l'altro sta nel modo in cui i bit a disposizione sono utilizzati per *codificare* il segno e il valore assoluto

Tra le codifiche utilizzate, le più note sono:

Modulo e segno (M&S)
Complemento a uno (C1)
Complemento a due (C2)
Rappresentazione «per eccesso»

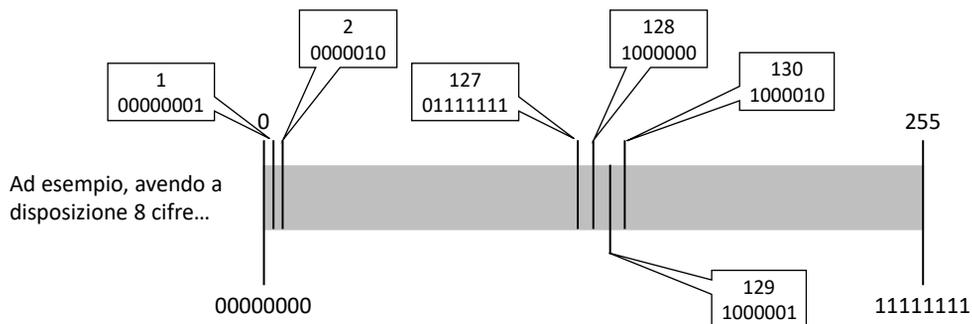
14

## Codifica degli interi in «Modulo & Segno»

15

## Codifica degli interi in «Modulo & Segno»

Quando codifichiamo i numeri positivi, utilizziamo tutte le cifre a disposizione per comporre il numero che ci interessa.



16

## Codifica degli interi in «Modulo & Segno»

	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
...								
127	0	1	1	1	1	1	1	1
128	1	0	0	0	0	0	0	0
129	1	0	0	0	0	0	0	1
130	1	0	0	0	0	0	1	0
131	1	0	0	0	0	0	1	1
...								
254	1	1	1	1	1	1	1	0
255	1	1	1	1	1	1	1	1

Nella codifica dei numeri dell'intervallo da 0 a 255...

Ogni combinazione delle 7 cifre meno significative si ripete due volte:

Una volta quando l'MSB vale 0, e un'altra volta quando l'MSB vale 1

17

## Codifica degli interi in «Modulo & Segno»

	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
...								
127	0	1	1	1	1	1	1	1
128	1	0	0	0	0	0	0	0
129	1	0	0	0	0	0	0	1
130	1	0	0	0	0	0	1	0
131	1	0	0	0	0	0	1	1
...								
254	1	1	1	1	1	1	1	0
255	1	1	1	1	1	1	1	1

Nella codifica dei numeri dell'intervallo da 0 a 255...

Ogni combinazione delle 7 cifre meno significative si ripete due volte:

Una volta quando l'MSB vale 0, e un'altra volta quando l'MSB vale 1

18

## Codifica degli interi in «Modulo & Segno»

	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	2
3	0	0	0	0	0	0	1	1	3
...									
127	0	1	1	1	1	1	1	1	127
128	1	0	0	0	0	0	0	0	0
129	1	0	0	0	0	0	0	1	1
130	1	0	0	0	0	0	1	0	2
131	1	0	0	0	0	0	1	1	3
...									
254	1	1	1	1	1	1	1	0	126
255	1	1	1	1	1	1	1	1	127

In altre parole, l'intervallo da 0 a 255, lo possiamo dividere in due sotto intervalli, i cui elementi hanno valore da 0 a 127

(senza cioè considerare il contributo dell'MSB, che adesso indica solo l'insieme di appartenenza)

19

## Codifica degli interi in «Modulo & Segno»

	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$	
0	+	0	0	0	0	0	0	0	0
1	+	0	0	0	0	0	0	1	1
2	+	0	0	0	0	0	1	0	2
3	+	0	0	0	0	0	1	1	3
...									
127	+	1	1	1	1	1	1	1	127
128	-	0	0	0	0	0	0	0	0
129	-	0	0	0	0	0	0	1	1
130	-	0	0	0	0	0	1	0	2
131	-	0	0	0	0	0	1	1	3
...									
254	-	1	1	1	1	1	1	0	126
255	-	1	1	1	1	1	1	1	127

In altre parole, l'intervallo da 0 a 255, lo possiamo dividere in due sotto intervalli, i cui elementi hanno valore da 0 a 127

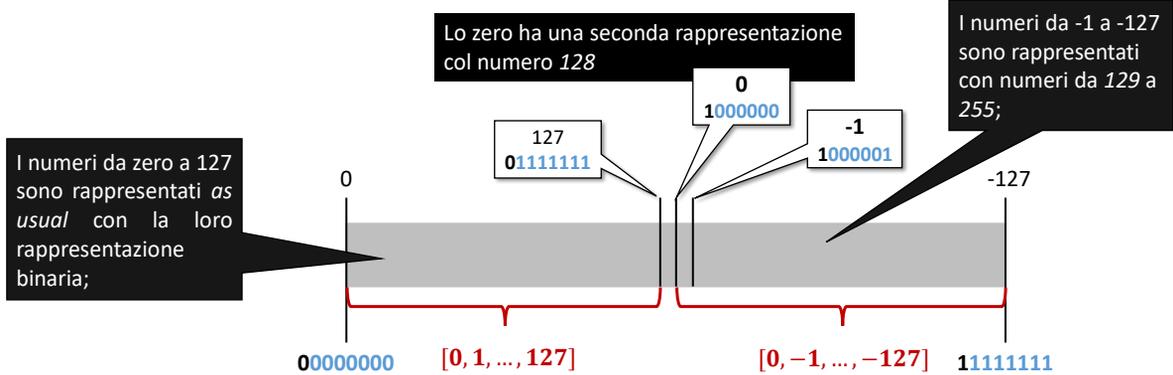
Potremmo dare al MSB un *significato* *differente*: **Il segno**.

**Se MSB=0, il resto delle cifre va a formare un numero positivo**

**Se MSB=1, il resto delle cifre va a formare un numero negativo**

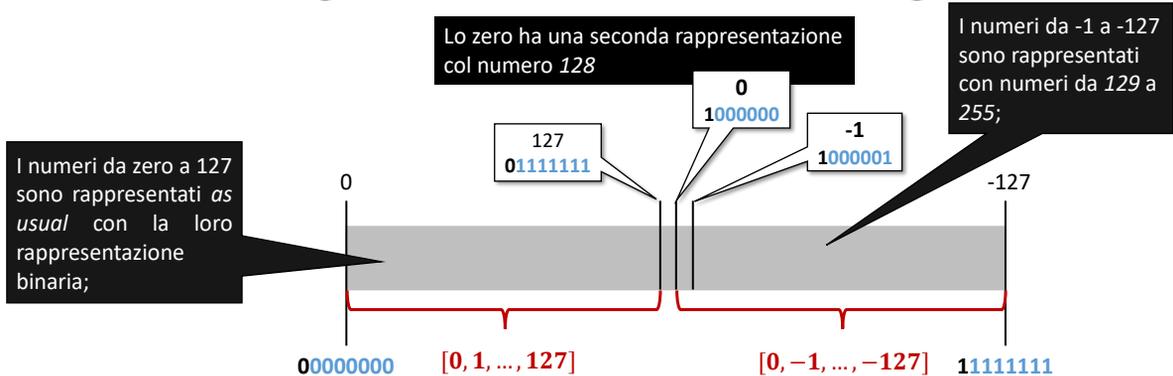
20

## Codifica degli interi in «Modulo & Segno»



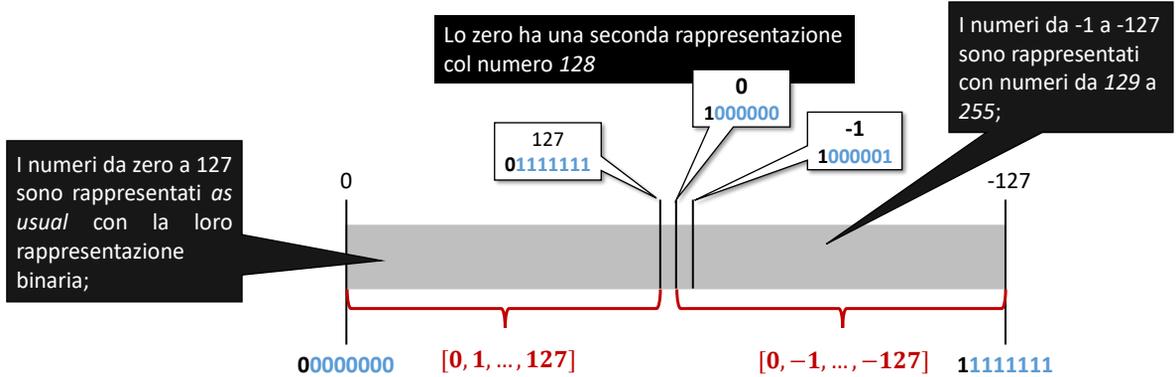
21

## Codifica degli interi in «Modulo & Segno»



22

## Codifica degli interi in «Modulo & Segno»



### Rappresentazione con «Modulo e Segno»

23

## Codifica degli interi in «Modulo & Segno»

In sistemi di calcolo in cui sono a disposizione  $l$  bit per la rappresentazione di un numero intero  $C$

La codifica in «**Modulo & Segno**» (M&S), utilizza:

➔ Il bit più significativo ( $b_{l-1}$ ) per indicare il segno di  $C$  cosicchè:

Se  $C \geq 0$  allora  $b_{l-1} = 0$ ;

Se  $C \leq 0$  allora  $b_{l-1} = 1$ ;

➔ I restanti  $l - 1$  bit per rappresentare il valore assoluto di  $C$  cosicchè:

$|C|$  è compreso tra  $0$  e  $2^{l-1} - 1$ ;

$C$  è compreso tra  $-2^{l-1} + 1$  e  $2^{l-1} - 1$ ;

24

## Codifica degli interi in «Modulo & Segno»

In sistemi di calcolo in cui sono a disposizione  $l$  bit per la rappresentazione di un numero intero  $C$

La codifica in «**Modulo & Segno**» (M&S), utilizza:

➔ Il bit più significativo ( $b_{l-1}$ ) per indicare il segno di  $C$  cosicchè:

Se  $C \geq 0$  allora  $b_{l-1} = 0$ ;

Se  $C \leq 0$  allora  $b_{l-1} = 1$ ;

Eh sì, ci sono due modi di codificare lo zero:  
con tutti i bit a zero  
con  $b_{l-1} = 1$  e il resto a zero

➔ I restanti  $l - 1$  bit per rappresentare il valore assoluto di  $C$  cosicchè:

$|C|$  è compreso tra  $0$  e  $2^{l-1} - 1$ ;  
 $C$  è compreso tra  $-2^{l-1} + 1$  e  $2^{l-1} - 1$ ;

25

## Codifica degli interi in «Modulo & Segno»

In sistemi di calcolo in cui sono a disposizione  $l$  bit per la rappresentazione di un numero intero  $C$

La codifica in «**Modulo & Segno**» (M&S), utilizza:

➔ Il bit più significativo ( $b_{l-1}$ ) per indicare il segno di  $C$  cosicchè:

Se  $C \geq 0$  allora  $b_{l-1} = 0$ ;

Se  $C \leq 0$  allora  $b_{l-1} = 1$ ;

Eh sì, ci sono due modi di codificare lo zero:  
con tutti i bit a zero  
con  $b_{l-1} = 1$  e il resto a zero

➔ I restanti  $l - 1$  bit per rappresentare il valore assoluto di  $C$  cosicchè:

$|C|$  è compreso tra  $0$  e  $2^{l-1} - 1$ ;  
 $C$  è compreso tra  $-2^{l-1} + 1$  e  $2^{l-1} - 1$ ;

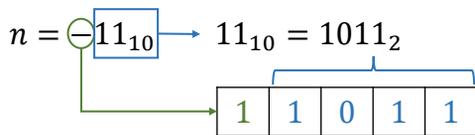
con parole di 4 bit i valori rappresentabili vanno  
da  $-7$  (1111) a  $+7$  (0111)  
con parole di 8 bit i valori rappresentabili vanno  
da  $-127$  (11111111) a  $+127$  (01111111)

26

## Modulo e segno: esempi

- Da decimale a binario (m&s)  $l = 5$ :

- $n = -11_{10} = ?$
- $n < 0 \Rightarrow b_4 = 1$  ( $n$  è negativo quindi il bit di segno è posto a 1)
- $\langle b_3 \dots b_0 \rangle = |n|_2 = 1011_2$  (nei restanti  $l-1$  bit si rappresenta il modulo di  $n$ )
- $n_2 = 11011_2$

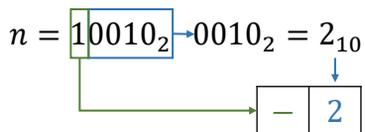


27

## Modulo e segno: esempi

- Da binario (m&s) a decimale:

- $n_2 = 10010_2$
- $b_4 = 1 \Rightarrow n_{10} < 0$  (il MSB è 1, il numero rappresentato è negativo)
- $\langle b_3 \dots b_0 \rangle = 0010_2 = 2_{10}$  (il modulo si ottiene convertendo i restanti  $l-1$  bit)
- $n_{10} = -2$



28

## Esercizi

**Convertire in decimale i seguenti numeri binari in notazione «modulo e segno»**

- 0001111=
- 1001111=
- 1010011=
- 1011011=

**Convertire i seguenti interi decimali in binario ( $l = 7$ ) usando la notazione «modulo e segno»**

- -63=
- -45=
- 64=
- -21=

29

## Mappa

*Per lo studio e l'approfondimento degli argomenti trattati*

30

## Mappa

- [Sli]** Conversioni tra il sistema di numerazione decimale e quelli con base arbitraria (prioritariamente 2,8 e 16);  
Per le codifiche binarie dei numeri interi con segno;

31

## Approfondimenti (*facoltativi*)

- [Misc]** Il sito «Calculand» <https://www.calculand.com> fornisce strumenti per:
- rappresentare sistemi di numerazione in qualsiasi base
  - effettuare la conversione di numeri tra sistemi di numerazione a scelta
- Selezionando la lingua italiana, il link «Sistemi di numerazione» appare sulla barra scorrevole in alto.

A New York c'è la sede della Società Duodecimale d'America (Dozenal Society of America) che ha lo scopo di promuovere l'uso della numerazione in base 12. Il sito della società è <https://dozenal.org> riporta articoli e fornisce strumenti per l'uso della base 12 per le più svariate applicazioni.

32