



**Università degli Studi di Palermo**  
*Dipartimento di Ingegneria Informatica*



# **Informatica di Base - 6 c.f.u.**

Anno Accademico 2007/2008

Docente: ing. Salvatore Sorce

## **Rappresentazione delle informazioni**

Sistemi di numerazione

Facoltà di Lettere e Filosofia

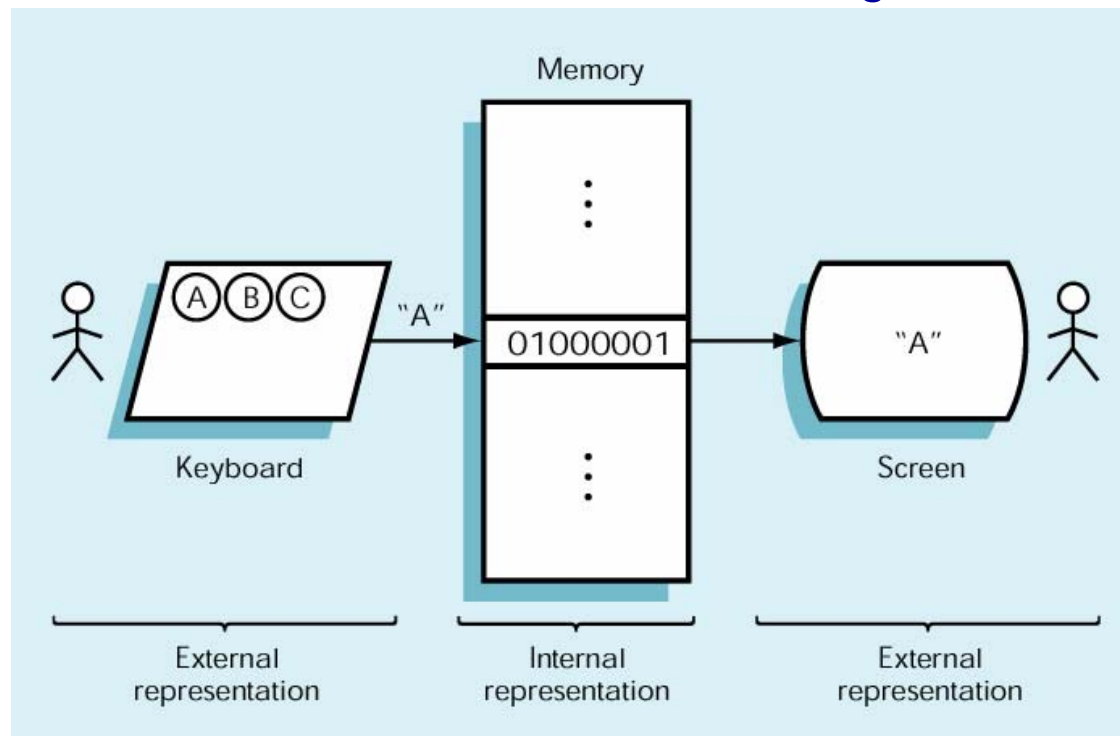


## Rappresentazione delle informazioni

- Notazioni convenzionali per la rappresentazione di informazioni allo scopo di renderne possibile lo scambio tra esseri umani
- Rappresentazione dei dati di tipo numerico
  - 10 cifre decimali: 0,1,2,3,4,5,6,7,8,9
- Rappresentazione dei dati di tipo testuale
  - 26 caratteri dell'alfabeto maiuscolo, minuscolo, segni di interpunzione e simboli speciali (£, \$, %, &, @, #, etc.)
- Notazione segno/grandezza per i numeri relativi
  - +47, -53
- Notazione decimale per i numeri reali
  - $n = i + f$
  - 12,34 dove 12 è la parte intera **i** e 0,34 è la parte frazionaria **f**

## Rappresentazione interna ed esterna

- Rappresentazione esterna
  - diretta all'interpretazione umana
- Rappresentazione interna
  - diretta ad essere usata all'interno dell'agente di calcolo





# Dissezione di un numero decimale

	Parte intera					Parte frazionaria	
	Migliaia	Centinaia	Decine	Unità		Decimi	Centesimi
	...	1	2	3		...	
<b>Peso</b>	1000	100	10	1		1/10	
<b>Posizione</b>	3	2	1	0		-1	
<b>Potenza</b>	$10^3$	$10^2$	$10^1$	$10^0$		$10^{-1}$	

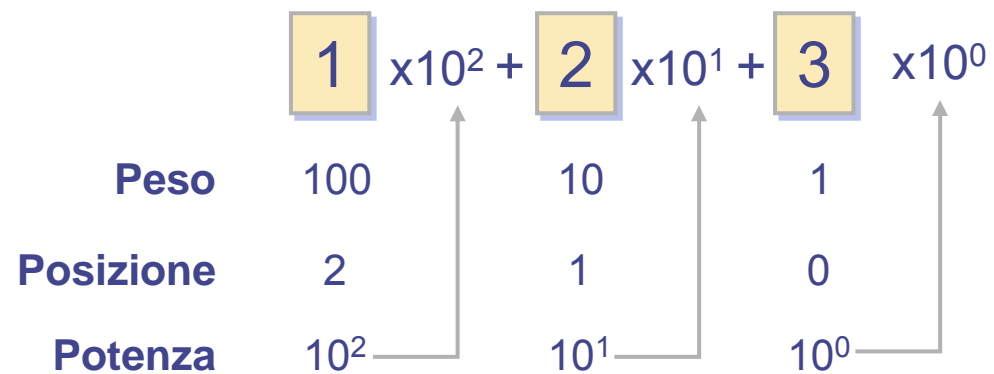


## Dissezione di un numero decimale

	<b>1</b>	<b>2</b>	<b>3</b>
<b>Peso</b>	100	10	1
<b>Posizione</b>	2	1	0
<b>Potenza</b>	$10^2$	$10^1$	$10^0$

## Dissezione di un numero decimale

$$123_{10} = 1 \times 100 + 2 \times 10 + 3 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$





## Sistema di numerazione binario

- All'interno di un elaboratore le informazioni sono rappresentate usando il sistema di numerazione binario
- Sistema di numerazione posizionale
  - Il valore di una cifra non dipende solo dalla cifra ma anche dalla posizione che occupa nella sequenza che rappresenta il numero
- Sistema di numerazione decimale
  - Sistema di numerazione *posizionale* in base 10
  - Utilizza soltanto le dieci cifre decimali (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
  - $123_{10} = 1 \times 100 + 2 \times 10 + 3 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$
- Sistema di numerazione binario
  - Sistema di numerazione *posizionale* in base 2
  - Utilizza soltanto le prime due cifre decimali (0 ed 1)
  - $1101_2 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
  - Le due cifre binarie, 0 e 1, sono chiamate **bit**, da **binary digit**

## Contare...

Contare in decimale:

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0

Contare in binario

(0, 1)

0



## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0  
1

Contare in binario  
(0, 1)

0  
1

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0

1

2

Contare in binario  
(0, 1)

0

1

10

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0  
1  
2  
3

Contare in binario  
(0, 1)

0  
1  
10  
11

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0  
1  
2  
3  
4

Contare in binario  
(0, 1)

0  
1  
10  
11  
100

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0

1

2

3

4

...

9

Contare in binario  
(0, 1)

0

1

10

11

100

...

1001

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0  
1  
2  
3  
4  
...  
9  
10

Contare in binario  
(0, 1)

0  
1  
10  
11  
100  
...  
1001  
1010

## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

Contare in binario  
(0, 1)

0

0

1

La base è sempre espressa come

1

2

**10**

**10**

3

Uno-Zero

11

4

100

...

...

9

1001

**10**

1010



## Contare...

Contare in decimale:  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

0

1

2

3

4

...

9

10

11

Contare in binario  
(0, 1)

0

1

10

11

100

...

1001

1010

1011



## Contare in binario

- In qualunque sistema di numerazione, la base è sempre espressa come

**10**

(leggi uno-zero)

Decimale	Binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

## Contare in binario

- In qualunque sistema di numerazione, la base è sempre espressa come 10
- La convenzione implicita è che il numero può essere riempito con zeri, muovendosi da destra a sinistra, in modo da mantenere lo stesso numero di cifre
- Continuando a contare,
  - $16 = 10000$
  - $17 = 10001$
  - etc.

Decimale	Binario
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111



## Conversione da binario a decimale

**Numeri interi**

$$1101_2 = ?_{10}$$

Notazione posizionale

	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	
Posizione	3	2	1	0	
Peso	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	
	$1 \times 8 +$	$1 \times 4 +$	$0 \times 2 +$	$1 \times 1 =$	<b><math>13_{10}</math></b>

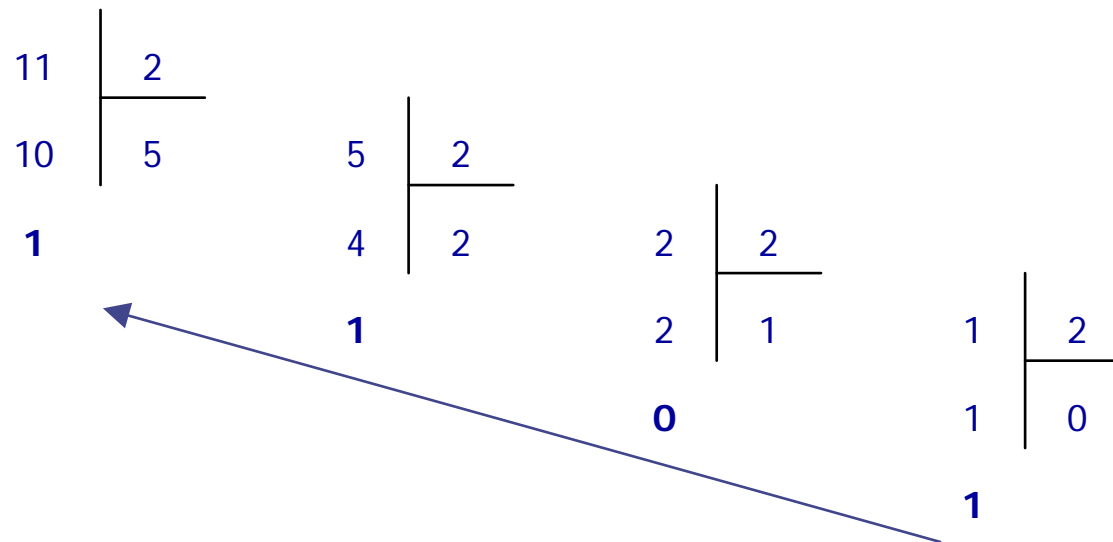
$$1101_2 = 13_{10}$$

# Conversione da decimale a binario

**Numeri interi**

$$11_{10} = ?_2$$

Metodo delle "divisioni successive"



$$11_{10} = 1011_2$$

## Una comoda alternativa ....

- Il sistema di numerazione esadecimale rappresenta i numeri **in base 16**
- Le cifre sono:
  - 0, 1, 2, ..., 9, A, B, C, D, E, F

## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ..., F)

0

Contare in decimale:  
0, 1, 2, 3, ..., 9

0

Contare in binario:  
0, 1

0

## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ..., F)

Contare in decimale:  
0, 1, 2, 3, ..., 9

Contare in binario:  
0, 1

0

0

0

1

1

1

## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ..., F)

Contare in decimale:  
0, 1, 2, 3, ..., 9

Contare in binario:  
0, 1

0

0

0

1

1

1

2

2

$10_2$



## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ..., F)

Contare in decimale:  
0, 1, 2, 3, ..., 9

Contare in binario:  
0, 1

0

0

0

1

1

1

2

2

$10_2$

...

...

...

A

10

1010

## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ..., F)

Contare in decimale:  
0, 1, 2, 3, ..., 9

Contare in binario:  
0, 1

0

0

0

1

1

1

2

2

$10_2$

...

...

...

A

10

1010

B

11

1011

## Contare...

Contare in esadecimale:  
(0, 1, ..., 9, A, B, ... , F)

0

1

2

...

A

B

...

F

Contare in decimale:  
(0, 1, 2, 3, ... , 9)

0

1

2

...

10

11

...

15

Contare in binario:  
(0, 1)

0

1

10<sub>2</sub>

...

1010

1011

...

1111

## Esadecimale e binario

Contare in esadecimale:

(0, 1, ..., 9, A, B, ... , F)

0

1

2

...

A

B

...

F

Contare in binario:

(0, 1)

0000

0001

0010

...

1010

1011

...

1111

## Esadecimale e binario

$$FA93_{16} = ?$$

## Esadecimale e binario

$$FA93_{16} = \text{XXXX XXXX XXXX } 0011_2$$

## Esadecimale e binario

$$FA93_{16} = \text{XXXX XXXX } 1001 \text{ } 0011_2$$

## Esadecimale e binario

$$FA93_{16} = \text{XXXX } 1010 \ 1001 \ 0011_2$$



## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

$$B78D_{16} = ?_2$$

## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

$$B78D_{16} = 1011$$

## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

$$B78D_{16} = 1011\ 0111$$

## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

$$B78D_{16} = 1011\ 0111\ 1000$$

## Esadecimale e binario

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_2$$

$$B78D_{16} = 1011\ 0111\ 1000\ 1101_2$$

## Esadecimale e binario

$$FA93_{16} = 001\ 111\ 101\ 010\ 010\ 011_2$$

$$1\ 7\ 5\ 2\ 2\ 3_8$$

$$B78D_{16} = 001\ 011\ 011\ 110\ 001\ 101_2$$

$$1\ 3\ 3\ 6\ 1\ 5_8$$

$$7312_8 = 1110\ 1100\ 1010_2$$

$$E\ C\ A_{16}$$

$$110111101011011_2 = 6\ 7\ 5\ 3\ 3_8$$

$$110111101011011_2 = 6\ F\ 5\ B_{16}$$

➤  $1C_{16} = ?_{10}$



➤  $1C_{16} = 1 \times 16^1 + C \times 16^0 = 16 + 12 = 28_{10}$

## Conversione da binario a decimale

$$1101,101_2 = ?_{10}$$

	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b> ,	<b>1</b>	<b>0</b>	<b>1</b>	
Posizione	3	2	1	0	-1	-2	-3	
Peso	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	$2^{-1}=1/2$	$2^{-2}=1/4$	$2^{-3}=1/8$	
	$1 \times 8 +$	$1 \times 4 +$	$0 \times 2 +$	$1 \times 1 +$	$1 \times 1/2 +$	$0 \times 1/4 +$	$1 \times 1/8 =$	13,625

$$1101,101_2 = 13,625_{10}$$



## Conversione da decimale a binario

$$12,75_{10} = ?_2$$

- Per la parte intera, metodo delle “divisioni successive”
- Per la parte frazionaria, metodo delle “moltiplicazioni successive”:

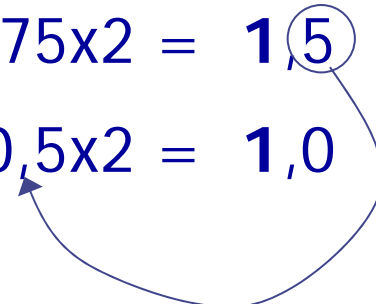
$$0,75 \times 2 = 1,5$$

## Conversione da decimale a binario

$$12,75_{10} = ?_2$$

- Per la parte intera, metodo delle “divisioni successive”
- Per la parte frazionaria, metodo delle “moltiplicazioni successive”:

$$0,75 \times 2 = 1,5$$

$$0,5 \times 2 = 1,0$$


## Conversione da decimale a binario

$$12,75_{10} = ?_2$$

- Per la parte intera, metodo delle “divisioni successive”
- Per la parte frazionaria, metodo delle “moltiplicazioni successive”:

$$0,75 \times 2 = 1,5$$

$$0,5 \times 2 = 1,0 \longrightarrow \text{STOP}$$

$$12,75_{10} = 1100,11_2$$



## Conversione da decimale a binario

- Per la parte intera, metodo delle “divisioni successive”
- Per la parte frazionaria, metodo delle “moltiplicazioni successive”
- N.B.: non sempre, da un numero di cifre finito (in base 10) si arriva ad un numero di cifre finito (in base 2)

$$\text{Es.: } (11,62)_{10} = (1101,10100110011\dots)_2$$



## Sistema di numerazione binario

- All'interno di un elaboratore le informazioni sono rappresentate usando il sistema di numerazione binario
- Sistema di numerazione posizionale
  - Il valore di una cifra non dipende solo dalla cifra ma anche dalla posizione che occupa nella sequenza che rappresenta il numero
- Sistema di numerazione decimale
  - Sistema di numerazione *posizionale* in base 10
  - Utilizza soltanto le dieci cifre decimali (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
  - $123_{10} = 1 \times 100 + 2 \times 10 + 3 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$
- Sistema di numerazione binario
  - Sistema di numerazione *posizionale* in base 2
  - Utilizza soltanto le prime due cifre decimali (0 ed 1)
  - $1101_2 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
  - Le due cifre binarie, 0 e 1, sono chiamate **bit**, da **binary digit**

## Bit, byte e multipli

- Un **bit** è l'unità di informazione, e rappresenta **uno** di due valori possibili, 0 e 1.
  - La scelta tra due alternative è la minima quantità di informazione possibile (Shannon)
  - Il valore massimo che può essere rappresentato con 1 bit è 1.
- Con due bit, si possono rappresentare tutte le combinazioni di 0 e 1, 00, 01, 10, 11, ovvero  $2^2$  possibili valori distinti (0,1,2,3).
  - Il valore massimo che può essere rappresentato con 2 bit è 3.
- Con quattro bit, si possono rappresentare tutte le combinazioni di 0 e 1, 0000, 0001, ..., 1110, 1111, ovvero  $2^4$  possibili valori distinti (0,1,2,...,14, 15).
  - Il valore massimo che può essere rappresentato con 4 bit è 15.
- In generale, con  $n$  bit possono essere rappresentati  $2^n$  valori distinti, da 0 a  $2^n - 1$



## Bit, byte e multipli

- Un **byte** (B) è costituito da 8 bit
  - 1 byte può rappresentare  $2^8 = 256$  valori distinti
  - Il massimo numero rappresentabile con 1 byte è  $2^8 - 1 = 256 - 1 = 255$
- Multipli del bit o del byte sono indicati con i prefissi
  - K – kilo,  $2^{10} = 1.024$
  - M – mega,  $2^{20} = 1.048.576$
  - G – giga,  $2^{30} = 1.073.741.824$
  - T – tera,  $2^{40} = 1.099.511.627.776$
- Se un modem lavora alla velocità di 28.8 Kbit/s, significa che trasmette:
 

$28.8 \times 2^{10} \text{ bit/s} = 29491,20 \text{ bit/s}$

ovvero, essendo 1 bit = 1/8 byte:

$28,8 \times 2^{10} \times (1/8) \text{ byte/s} = 3686,40 \text{ byte/s}$
- **Un disco fisso da 10 GB quanti bit può contenere?**

## Bit, byte e multipli

- Un **byte** (B) è costituito da 8 bit
  - 1 byte può rappresentare  $2^8 = 256$  valori distinti
  - Il massimo numero rappresentabile con 1 byte è  $2^8 - 1 = 256 - 1 = 255$
- Multipli del bit o del byte sono indicati con i prefissi
  - K – kilo,  $2^{10} = 1.024$
  - M – mega,  $2^{20} = 1.048.576$
  - G – giga,  $2^{30} = 1.073.741.824$
  - T – tera,  $2^{40} = 1.099.511.627.776$
- Se un modem lavora alla velocità di 28.8 Kbit/s, significa che trasmette:
 

$28.8 \times 2^{10} \text{ bit/s} = 29491,20 \text{ bit/s}$

ovvero, essendo 1 bit = 1/8 byte:

$28,8 \times 2^{10} \times (1/8) \text{ byte/s} = 3686,40 \text{ byte/s}$
- **Un disco fisso da 10 GB quanti bit può contenere?**

**$10 \times 2^{30} \text{ byte} = 10 \times 2^{30} \times 8 \text{ bit}$**

**$= 85.899.345.920 \text{ bit}$**



## Rappresentazione delle informazioni

- Notazioni convenzionali per la rappresentazione di informazioni allo scopo di renderne possibile lo scambio tra esseri umani
- Rappresentazione dei dati di tipo numerico
  - 10 cifre decimali: 0,1,2,3,4,5,6,7,8,9
- Rappresentazione dei dati di tipo testuale
  - 26 caratteri dell'alfabeto maiuscolo, minuscolo, segni di interpunzione e simboli speciali (£, \$, %, &, @, #, etc.)
- Notazione segno/grandezza per i numeri relativi
  - +47, -53
- Notazione decimale per i numeri reali
  - $n = i + f$
  - 12,34 dove 12 è la parte intera **i** e 0,34 è la parte frazionaria **f**

## Rappresentazione dei numeri interi con segno

### ➤ Modulo e segno:

- Usiamo 1 bit per il segno (+ o -) ed n bit (per es. 7) per il modulo

Es.  $-124 \rightarrow |1|1111100|$  cioè  $11111100$

$+071 \rightarrow |0|1000111|$  cioè  $01000111$

## Rappresentazione dei numeri interi con segno

- Modulo e segno:
  - Usiamo 1 bit per il segno (+ o -) ed n bit (per es. 7) per il modulo

Es.  $-124 \rightarrow |1|1111100|$  cioè  $11111100$

$+071 \rightarrow |0|1000111|$  cioè  $01000111$

Con 8 bit otteniamo

$-127 \rightarrow 11111111$

.

.

.

.

$+127 \rightarrow 01111111$

## Rappresentazione dei numeri interi con segno

### ➤ Modulo e segno:

- Usiamo 1 bit per il segno (+ o -) ed n bit (per es. 7) per il modulo

Es.  $-124 \rightarrow |1|1111100|$  cioè  $11111100$

$+071 \rightarrow |0|1000111|$  cioè  $01000111$

Con 8 bit otteniamo

$-127 \rightarrow 11111111$

.

$- 0 \rightarrow 10000000$

$+ 0 \rightarrow 00000000$

.

$+127 \rightarrow 01111111$

**PROBLEMA !**

## Rappresentazione dei numeri interi con segno

- Soluzione: complemento alla base (complemento a 2)
  - **Se il numero è negativo** scriviamo il corrispondente positivo da  $n$  bit (per es. 7) in un campo di  $n+1$  bit (nel nostro caso 8)
  - invertiamo i bit 1 e 0
  - sommiamo 1 al risultato

*Es. -96:*

96

→ 01100000

## Rappresentazione dei numeri interi con segno

- Soluzione: complemento alla base (complemento a 2)
  - **Se il numero è negativo** scriviamo il corrispondente positivo da  $n$  bit (per es. 7) in un campo di  $n+1$  bit (nel nostro caso 8)
  - invertiamo i bit 1 e 0
  - sommiamo 1 al risultato

*Es. -96:*

*96* → *01100000*

*invertiamo* → *10011111*



## Rappresentazione dei numeri interi con segno

- Soluzione: complemento alla base (complemento a 2)
  - **Se il numero è negativo** scriviamo il corrispondente positivo da  $n$  bit (per es. 7) in un campo di  $n+1$  bit (nel nostro caso 8)
  - invertiamo i bit 1 e 0
  - sommiamo 1 al risultato

*Es. -96:*

<i>96</i>	<i>→ 01100000</i>
<i>invertiamo</i>	<i>→ 10011111+</i>
<i>sommiamo 1</i>	<i>→ <u>00000001=</u></i>

## Rappresentazione dei numeri interi con segno

- Soluzione: complemento alla base (complemento a 2)
  - **Se il numero è negativo** scriviamo il corrispondente positivo da  $n$  bit (per es. 7) in un campo di  $n+1$  bit (nel nostro caso 8)
  - invertiamo i bit 1 e 0
  - sommiamo 1 al risultato

*Es. -96:*

<i>96</i>	<i>→ 01100000</i>
<i>invertiamo</i>	<i>→ 10011111+</i>
<i>sommiamo 1</i>	<i>→ <u>00000001</u>=</i>
<i>-96</i>	<i>→ 10100000</i>

## Rappresentazione dei numeri interi con segno

- Soluzione: complemento alla base (complemento a 2)
  - **Se il numero è negativo** scriviamo il corrispondente positivo da  $n$  bit (per es. 7) in un campo di  $n+1$  bit (nel nostro caso 8)
  - invertiamo i bit 1 e 0
  - sommiamo 1 al risultato

*Con 8 bit otteniamo:*

$-128 \rightarrow 10000000$

$-127 \rightarrow 10000001$

...

$- 1 \rightarrow 11111111$

$0 \rightarrow 00000000$

$+ 1 \rightarrow 00000001$

...

$+127 \rightarrow 01111111$



## Rappresentazione dei numeri reali

- Come rappresentare un numero del tipo 12,34?
- Occorre utilizzare la cosiddetta *notazione scientifica*, in base alla quale, un qualsivoglia numero reale  $N$  può sempre essere espresso come:

$$N = \pm M \times B^{\pm E}$$

Dove  $M$  è la mantissa (che deve essere maggiore o uguale di 0,1 e minore di 1  $\Rightarrow 0,1 \leq M < 1$ ) e  $E$  è l'esponente.

- Se  $B=10$ ,  $12,34 = 12,34 \times 10^0 = 1,234 \times 10^1 = 0,1234 \times 10^2$
- Nel caso di numeri binari, la base deve ovviamente essere 2



## Rappresentazione dei numeri reali

- Sia dato come esempio il numero  $5.75_{10}$ 
  - $5_{10} = 101_2$  ( $5 / 4 = 1$ ,  $(5 - 4) / 2 = 0$ ,  $1 / 1 = 1$ )
  - $0.75_{10} = 0.5 + 0.25 = \frac{1}{2} + \frac{1}{4} = 1 * 2^{-1} + 1 * 2^{-2} = 0.11_2$
  - $5.75_{10} = 101.11_2 = 10.111 * 2^1 = 1.0111 * 2^2 = 0.10111 * 2^3$

## Rappresentazione dei numeri reali

- Usando 16 bit, per esempio, si può indicare:
  - Il segno della mantissa (1 se -, 0 se +) nel bit 15
  - La mantissa nei 9 bit seguenti
  - Il segno dell'esponente (1 se -, 0 se +) nel bit 5
  - L'esponente negli ultimi 5 bit

