Università degli Studi di Palermo
Dipartimento di Ingegneria Informatica

Elaborazione di Immagini e Suoni / Riconoscimento e Visioni Artificiali 12 c.f.u.

Anno Accademico 2008/2009

Docente: ing. Salvatore Sorce

Rappresentazione delle informazioni

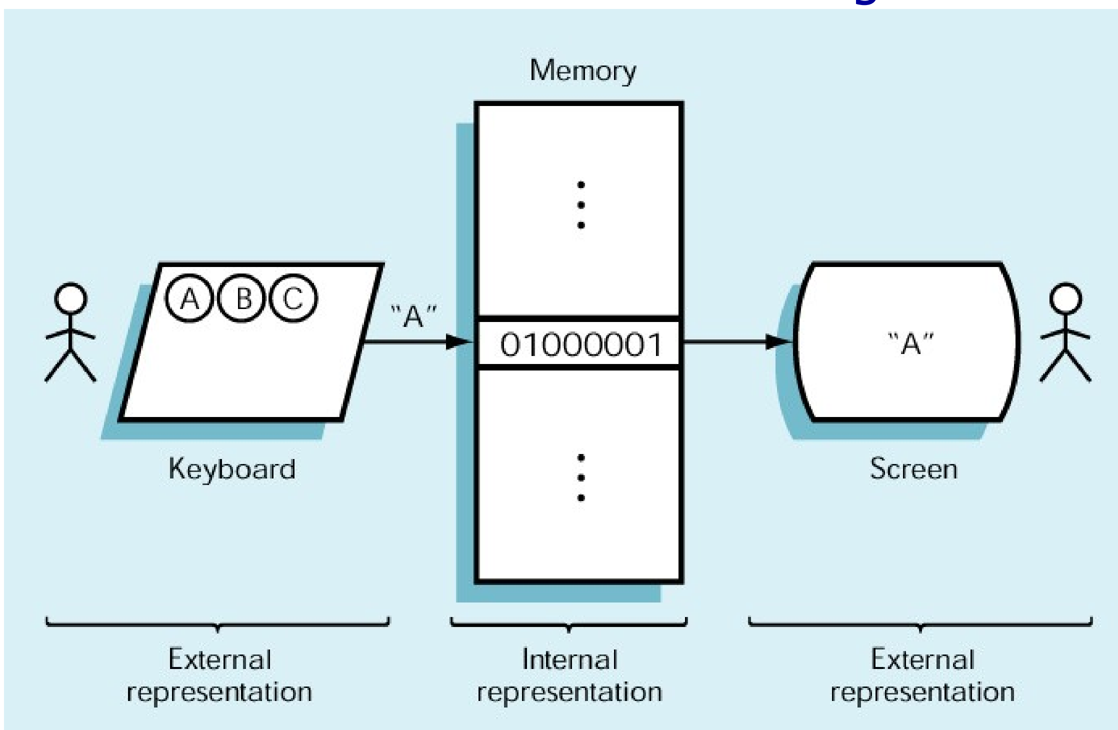
I parte: Sistemi di numerazione

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	Millesimi	…
	…	1	2	3 .	…			
Peso	1000	100	10	1	1/10			
Posizione	3	2	1	0	-1			
Potenza	10^3	10^2	10^1	10^0	10^{-1}			

Dissezione di un numero decimale

	1	2	3
Peso	100	10	1
Posizione	2	1	0
Potenza	10^2	10^1	10^0

Dissezione di un numero decimale

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

$$\boxed{1} \times 10^2 + \boxed{2} \times 10^1 + \boxed{3} \times 10^0$$

Peso	100	10	1
Posizione	2	1	0
Potenza	10^2	10^1	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 **bi**nary digi**t**

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)	Contare in binario (0, 1)
0	0
1	1

Contare…

Contare in decimale: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Contare in binario (0, 1)
0	0
1	1
2	10

Contare…

Contare in decimale: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Contare in binario (0, 1)
0	0
1	1
2	10
3	11

Contare…

Contare in decimale: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Contare in binario (0, 1)
0	0
1	1
2	10
3	11
4	100

Contare…

Contare in decimale: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Contare in binario (0, 1)
0	0
1	1
2	10
3	11
4	100
…	…
9	1001

Contare…

Contare in decimale: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)	Contare in binario (0, 1)
0	0
1	1
2	10
3	11
4	100
…	…
9	1001
10	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)	Contare in binario (0, 1)
0	0
1	1
2	10
3	11
4	100
…	…
9	1001
10	1010
11	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

	1	**1**	**0**	**1**	
Posizione	3	2	1	0	
Peso	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	
	1x8+	1x4+	0x2+	1x1=	$\mathbf{13_{10}}$

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

Conversione da decimale a binario

Numeri interi

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

Metodo delle "divisioni successive"

11	2	
10	5	2
1	4	2
	1	

2	2	
2	1	2
0	1	0
	1	

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

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:	Contare in decimale:	Contare in binario:
(0, 1, …, 9, A, B, … , F)	0, 1, 2, 3, … , 9	0, 1
0	0	0

Contare…

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

0
1

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

0
1

Contare in binario:
0, 1

0
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

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
…	…	…
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
…	…	…
A	10	1010
B	11	1011

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
…	…	…
A	10	1010
B	11	1011
…	…	…
F	15	1111

Esadecimale e binario

Contare in esadecimale (HEX)	Contare in binario (BIN)
{0,1,2,…,9,A,B,C,D,E,F}	{0,1}
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

$$FA93_{16} = ?$$

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

$$FA93_{16} = \text{xxxx } 1010\ 1001\ 0011_{2}$$

Università degli Studi di Palermo

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

$$B78D_{16} = ?_2$$

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

$$B78D_{16} = 1011$$

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

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

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

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

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

Esadecimale e binario

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

$$FA93_{16} = 1111\ 1010\ 1001\ 0011_{2}$$

$$B78D_{16} = 1011\ 0111\ 1000\ 1101_{2}$$

Esadecimale e binario

$FA93_{16} = 1111\ 1010\ 1001\ 0011_{2}$

Esadecimale e binario

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

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Esadecimale e binario

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

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Esadecimale e binario

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

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

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Esadecimale e binario

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

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

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

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

$7312_8 = 1110\ 1100\ 1010_2$

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

$1101111101011011_2 = 6\ F\ 5\ B_{16}$

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

- $1C_{16} = ?_{10}$

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

Conversione da binario a decimale

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

	1	1	0	1 ,	1	0	1	
Posizione	3	2	1	0	-1	-2	-3	
Peso	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	$2^{-1}=1/2$	$2^{-2}=1/4$	$2^{-3}=1/8$	
	1x8+	1x4+	0x2+	1x1+	1x1/2+	0x1/4+	1x1/8=	13,625

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

Università degli Studi di Palermo

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":

$0{,}75 \times 2 = \mathbf{1}{,}5$

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":

$$0{,}75 \times 2 = \mathbf{1}{,}5$$

$$0{,}5 \times 2 = \mathbf{1}{,}0$$

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":

$$0{,}75 \times 2 = \mathbf{1}{,}5$$

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

$$12{,}75_{10} = 1100{,}11_{2}$$

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)

Es.: $(11{,}62)_{10} = (1101{,}1010011001 1\ldots)_2$

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 **bi**nary digi**t**

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:

  28.8×2^{10} bit/s = 29491,20 bit/s

  ovvero, essendo 1 bit = 1/8 byte:

  $28,8 \times 2^{10} \times (1/8)$ byte/s = 3686,40 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 x 2^{10} bit/s = 29491,20 bit/s

  ovvero, essendo 1 bit = 1/8 byte:

  28,8 x 2^{10} x (1/8) byte/s = 3686,40 byte/s

- **Un disco fisso da 10 GB quanti bit può contenere?**

  10x2^{30} Byte = 10x2^{30}x8 bit =85.899.345.920 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**

Università degli Studi di Palermo

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 → |<u>1</u>|<u>1111100</u>| cioè 11111100

+071 → |<u>0</u>|<u>1000111</u>| 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 → |1|1111100| cioè 11111100

+071 → |0|1000111| cioè 01000111

Con 8 bit otteniamo

-127 → 11111111

.

.

.

+127 → 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 → |1|1111100| cioè 11111100

+071 → |0|1000111| cioè 01000111

Con 8 bit otteniamo

-127 → 11111111

.

- 0 → 10000000

+ 0 → 00000000 *PROBLEMA !*

.

+127 → 01111111

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:

96	→ *01100000*
invertiamo	→ *10011111+*
sommiamo 1	→ *00000001=*

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+*
sommiamo 1	→ *<u>00000001=</u>*
-96	→ *10100000*

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 → 10000000

-127 → 10000001

…

- 1 → 11111111

0 → 00000000

+ 1 → 00000001

…

+127 → 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 => $0{,}1 \leq M < 1$) ed *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

Università degli Studi di Palermo

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)
  - $0.75_{10} = 0.5 + 0.25 = ½ + ¼ = 1 * 2^{-1} + 1 * 2^{-2} = 0.11_2$

$$5.75_{10} = 101.11_2 = 10.111 * 2^1 = 1.0111 * 2^2 =$$

$$= \mathbf{0.10111 * 2^3}$$

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