

### Esercizio 1:

Convertire le seguenti coppie di numeri decimali in numeri di 5 bit in complemento a due. Per ogni coppia:

1. Eseguire somma e sottrazione.
2. Valutare se è avvenuto trabocco o meno.
3. Convertire il risultato da binario a decimale nel caso non sia avvenuto trabocco

- a) 5 e 12
  - b) -3 e 6
  - c) -13 e 6
  - d) 12 e -14
- 

a)

complemento a due:  $5 = 00101$       $12 = 01100$

somma:

$5 + 12 = 00101 + 01100 = 10001 \Rightarrow$  TRABOCCO

sottrazione:

$5 + -12 = 00101 + 10100 = 11001 \Rightarrow -7$

b)

complemento a due:  $-3 = 11101$       $6 = 00110$

somma:

$-3 + 6 = 11101 + 00110 = 00011 \Rightarrow 3$

sottrazione:

$-3 + -6 = 11101 + 11010 = 10111 \Rightarrow -9$

c)

complemento a due:  $-13 = 10011$       $6 = 00110$

somma:

$-13 + 6 = 10011 + 00110 = 11001 \Rightarrow -7$

sottrazione:

$-13 + -6 = 10011 + 11010 = 01101 \Rightarrow$  TRABOCCO

d)

complemento a due:  $12 = 01100$       $-14 = 10010$

somma:

$12 + -14 = 01100 + 10010 = 11110 \Rightarrow -2$

sottrazione:

$12 + 14 = 01100 + 01110 = 11010 \Rightarrow$  TRABOCCO

### Esercizio 2:

Assumiamo di voler rappresentare i numeri reali con sequenze binarie di un byte in complemento a due espresse in virgola fissa. Quale sarà l'intervallo e la precisione dei numeri rappresentabili se piazzassimo la virgola tra il bit numero 3 ed il bit numero 4? E tra il bit numero 5 ed il bit numero 6? Si assuma che i bit sono numerati da 0 a 7 a partire dal bit più a destra.

---

a) Virgola tra bit 3 e bit 4

Intervallo: da  $-2^3$  a  $2^3 - 2^{-4}$

Precisione:  $2^{-4}$

b) Virgola tra bit 5 e bit 6

Intervallo: da  $-2$  a  $2 - 2^{-6}$

Precisione:  $2^{-6}$

### Esercizio 3:

Trovare il valore decimale dei seguenti numeri binari rappresentati in complemento a due con virgola fissa:

a) 011,01000

b) 0010,1100

c) 11010,100

d) 1100,0100

---

a) 3,25

b) 2,75

c) -5,5

d) -3,75

#### Esercizio 4

Rappresentare i seguenti numeri reali decimali come numeri binari a virgola mobile e formato a precisione singola (32 bit). Nella conversione approssimare il numero alla 4 cifra binaria dopo la virgola.

- a) 25,45
- b) -13,55
- c) 32,5
- d) -11,25

---

a)

Valore assoluto in binario: 11001,0111

m: 10010111

e: 4

e':  $127 + 4 = 131 = 10000011$

segno: 0

Precisione singola: 0 10000011 100101110...0

b)

Valore assoluto in binario: 1101,1000

m: 1011000

e: 3

e':  $127 + 3 = 130 = 10000010$

segno: 1

Precisione singola: 1 10000010 10110...0

c)

Valore assoluto in binario: 100000,1

m: 000001

e: 5

e':  $127 + 5 = 132 = 10000100$

segno: 0

Precisione singola: 0 10000100 0000010...0

d)

Valore assoluto in binario: 1011,01

m: 01101

e: 3

e':  $127 + 3 = 130 = 10000010$

segno: 1

Precisione singola: 1 10000010 011010...0

### Esercizio 5

Rappresentare in decimale i seguenti numeri binari in formato a precisione singola:

- a) 010000010011010...0
  - b) 10111111010...0
  - c) 01111111011010...0
  - d) 1000000000...0
- 

a)

m: 01101

e': 130

e:  $130 - 127 = 3$

segno: 0

Decimale:  $1,01101 * 2^3 = 1011,01 = 11,25$

b)

m: 1

e': 126

e:  $126 - 127 = -1$

segno: 1

Decimale:  $1,1 * 2^{-1} = 0,11 = -0,75$

c)

m: 01101

e': 255

segno: 0

Decimale: NaN

d)

m: 0

e': 0

segno: 1

Decimale: 0

### Esercizio 6

Calcolare la distanza di Hamming di questo insieme di sequenze binarie a 7 bit:

- {0000000, 1111111, 0010110, 1101001, 0101010, 1010101}

Fino a quanti errori possono essere corretti?

---

Distanza di Hamming delle coppie:

$$1-2 = 7$$

$$1-3 = 3$$

$$1-4 = 4$$

$$1-5 = 3$$

$$1-6 = 4$$

$$2-3 = 4$$

$$2-4 = 3$$

$$2-5 = 4$$

$$2-6 = 3$$

$$3-4 = 7$$

$$3-5 = 4$$

$$3-6 = 3$$

$$4-5 = 3$$

$$4-6 = 4$$

$$5-6 = 7$$

Distanza di Hamming dell'insieme:

$$h = \min(7, 3, 4, 3, 4, 4, 3, 4, 3, 7, 4, 3, 3, 4, 7) = 3$$

Si può correggere al più 1 errore su di un singolo bit:

$$(h - 1) / 2 = 2 / 2 = 1$$

### Esercizio 7

Si abbia un archivio di immagini fotografiche digitali bitmap con risoluzione 2048x2048 e profondità di colore di 3 byte/pixel. Quante ne può ospitare una memoria da 6 GB senza compressione ( $1 \text{ GB} = 1024 \times 1024 \times 1024 = 2^{30}$  byte)

---

Numero di immagini = Spazio disponibile in byte / spazio occupato da un'immagine in byte =  
 $= (6 * 1024 * 1024 * 1024) / (3 * 2048 * 2048) = (6 * 2^{30}) / (3 * 2^{22}) = (6 / 3) * (2^{30} / 2^{22}) = 2^9 =$   
512 immagini