Informatica B
Esercitazione 7 (Soluzioni)

15 novembre 2021

Codifica binaria

7.1 Sono dati i seguenti interi **senza segno**:

$$x = (111001)_2$$
$$y = (27)_{10}$$

Si effettuino (a mano) le seguenti operazioni:

1. Convertire x in base 10

   Usiamo la definizione:
   $(111001)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 = 32 + 16 + 8 + 1 = (57)_{10}$

2. Convertire y in base 2. Quanti bit sono necessari per rappresentarlo?

   Usiamo il metodo delle divisioni successive:

27	2
13	1
6	1
3	0
1	1
0	1

   Leggendo i resti dal basso, $(27)_{10} = (11011)_2$; occorrono 5 bit.
   Verifica (conversione inversa usando la definizione): $16 + 8 + 2 + 1 = 27$.

3. Calcolare la somma $x + y$ in aritmetica binaria **senza segno**

(1)	(1)	(1)		(1)	(1)		
	1	1	1	0	0	1	+
	0	1	1	0	1	1	=
1	0	1	0	1	0	0	

4. *(Bonus)* scrivere x e y in base 8 e in base 16.

   $(\underbrace{111}_{7}\ \underbrace{001}_{1})_2 = (71)_8$
   $(\underbrace{0011}_{3}\ \underbrace{1001}_{9})_2 = (39)_{16}$
   $(\underbrace{011}_{3}\ \underbrace{011}_{3})_2 = (33)_8$
   $(\underbrace{0001}_{1}\ \underbrace{1011}_{11=B})_2 = (1B)_{16}$

Complemento a due (CP2)

7.2 Si vogliono memorizzare delle temperature in gradi centigradi. Sappiamo che la temperatura sul pianeta Terra è compresa tra -90 e 60 (inclusi). Ipotizzando di rappresentare le temperature con la **codifica CP2**:

1. Quanti bit sono necessari?

   Il valore assoluto del numero negativo più piccolo possibile è 90. Richiede (senza segno) 7 bit.
   Il numero positivo più grande possibile è 60. Richiede (senza segno) 6 bit.
   Un numero di bit sufficienti è quindi $m = \max(7, 6) + 1 = 8$.
   Con $m = 8$ possiamo rappresentare tutti i numeri nell'intervallo:
   $[-2^7, 2^7 - 1] = [-128, 127]$, estremi inclusi.
   Con un bit in meno ($m = 7$), potremmo rappresentare tutti i numeri nell'intervallo: $[-64, 63]$, il che non sarebbe sufficiente. Il numero di bit necessari è quindi 8.

2. Quali sono le temperature massima e minima effettivamente memorizzabili?

   Vedi il punto precedente.

3. Quante temperature si possono memorizzare avendo a disposizione 500 byte di memoria?

   Una memoria di 500 byte corrisponde a $500 \times 8 = 4000$ bit.
   Siccome una temperatura richiede 8 bit, possiamo memorizzare $4000/8 = 500$ temperature.

4. (*Bonus*) Come cambiano le risposte se si vogliono memorizzare temperature del pianeta Marte, sapendo che sono sempre comprese tra -128 e 20 (inclusi)?

   Il valore assoluto del numero negativo più piccolo possibile è 128. Richiede (senza segno) 8 bit.
   Il numero positivo più grande possibile è 20. Richiede (senza segno) 5 bit.

Un numero di bit sufficienti è quindi $m = \max(8, 5) + 1 = 9$.
Con $m = 9$ possiamo rappresentare tutti i numeri nell'intervallo:
$[-256, 255]$, estremi inclusi.
Tuttavia, con un bit in meno ($m = 8$), potremmo rappresentare tutti i numeri nell'intervallo $[-128, 127]$, che è sufficiente!
Servono quindi 8 bit.

7.3 Sono dati i seguenti interi:

$$x = (10100101)_{CP_2}$$
$$y = (-62)_{10}$$

Si effettuino (a mano) le seguenti operazioni, precisando sempre il bit di carry e il bit di overflow:

1. Convertire x in base 10

   Usiamo la definizione:
   $(10100101)_{CP2} = -2^7 + 2^5 + 2^2 + 1 = -128 + 32 + 4 + 1 = (-91)_{10}$

2. Convertire y in codifica CP2. Quanti bit sono necessari per rappresentarlo?

   Scriviamo prima la codifica binaria di $|y| = 62$:

62	2
31	0
15	1
7	1
3	1
1	1
0	1

   Quindi, $(63)_{10} = (111110)_2$; occorrono 6 bit per il numero senza segno, quindi 7 per il numero in CP2.
   Aggiungiamo uno 0 a sinistra di 111110 (per utilizzare effettivamente 7 bit) ed eseguiamo l'operazione di complemento a 2. Prima complementiamo le cifre di 0111110:

0	1	1	1	1	1	0	!
1	0	0	0	0	0	1	

   Poi sommiamo 1:

					(1)		
1	0	0	0	0	0	1	+
						1	
1	0	0	0	0	1	0	

   Quindi, $(-62)_{10} = (1000010)_{CP2}$

3. Calcolare la differenza $x - y$ in aritmetica **CP2**.

   Questo equivare a calcolare $(-91)_{10} + (62)_{10}$. Dovremmo quindi negare (con l'operazione di complemento a 2) il secondo operando e poi effetuare la somma. In questo caso sappiamo già, dal punto 2, la rappresentazione binaria di $(62)_{10}$. Quindi:

			(1)	(1)	(1)	(1)				
		1	0	1	0	0	1	0	1	+
		0	0	1	1	1	1	1	0	=
[0]	(0)	1	1	1	0	0	0	1	1	

   Per allineare i due numeri, abbiamo aggiunto zeri alla sinistra del secondo addendo, in quanto è un numero **positivo**.
   Il bit di carry è (0) e viene ignorato.
   Essendo una somma di un numero positivo e uno negativo, il bit di overflow è sicuramente [0].

4. Calcolare la somma $x + y$ in aritmetica **CP2**.

   Questo equivare a calcolare $(-91)_{10} + (-62)_{10}$. In CP2:

		1	0	1	0	0	1	0	1	+
		1	1	0	0	0	0	1	0	=
[1]	(1)	0	1	1	0	0	1	1	1	

   Per allineare i due numeri, abbiamo aggiunto uni alla sinistra del secondo addendo, in quanto è un numero **negativo**.
   Il bit di carry è (1) e viene ignorato.
   Abbiamo sommato due numeri negativi e ottenuto un numero positivo. Il risultato è inconsistente, quindi settiamo il bit di overflow a [1].