

Esercitazione Informatica I
AA 2012-2013

Nicola Paoletti

6 Giugno 2013

Conversioni (6 punti)

Effettuare le seguenti conversioni, tenendo conto del numero di bit con cui si rappresenta il numero da convertire/convertito.

1. 16 bit $(B6C9)_H =$
 - (a) $= (?)_2 = (?)_8$
 - (b) $= (?)_{10}$ (supponendo una rappresentazione senza segno)
 - (c) $= (?)_{10}$ (supponendo una rappresentazione segno + modulo)
 - (d) $= (?)_{10}$ (supponendo una rappresentazione complemento a 2)
2. $(-749)_{10} =$
 - (a) $(?)_H = (?)_2 = (?)_8$ (segno + modulo, 12 bit)
 - (b) $(?)_H = (?)_2 = (?)_8$ (complemento a 2, 12 bit)

Aritmetica binaria (9 punti)

3. Calcolare la somma tra i seguenti interi senza segno a 24 bit. **A5F20B + 125ECC + 3126B9**
4. Calcolare la stessa somma nel caso in cui gli operandi siano rappresentati in complemento a 2.
5. Calcolare la somma tra i seguenti interi rappresentati in modulo e segno a 16 bit. **F2C9 - 1F01**.

Floating point (8 punti)

6. Si rappresenti in formato IEEE 754 single precision il valore $(77.875)_{10}$
7. Dato il seguente valore rappresentato in IEEE 754 single precision, si determini il valore in decimale con notazione normalizzata $x, y \times 10^z$. **F2578000**.

Porte logiche e algebra booleana (4 punti)

8. Si scriva la tabella di verità per la porta logica nota come CNOT (Controlled Not). Tale porta mappa una coppia di bit (C, T) (che stanno per control-target), in una coppia di bit (C_1, T_1) tale che $C_1 = C$ e T_1 viene negato se e solo se C è 1.
9. Si scriva la tabella di verità per l'operazione di che effettua lo shift a sinistra (moltiplicazione per 2) su un registro a tre bit XYZ . La funzione quindi mappa una la tripla di bit (X, Y, Z) in un'altra tripla di bit (X_1, Y_1, Z_1) risultato dello shift.

Algoritmi e programmazione JavaScript (4 punti)

10. Creare una funzione `replica`, che dato in ingresso un array `a` restituisce un'array di lunghezza `a.length*2`, in cui ogni elemento di `a` viene ripetuto 2 volte consecutive. Ovvero, se `r` è l'array restituito in risultato, vale che: `r[2*i]=r[2*i+1]=a[i]`, `i=0, ..., a.length -1`

Conversioni - Soluzioni

1) $-(B6C9)_H = (\boxed{1011 \mid 0110 \mid 1100 \mid 1001})_2 = (133311)_8$
 - senza segno

$$(1011011011001001)_2 = (2^0 + 2^3 + 2^6 + 2^7 + 2^9 + 2^{10} + 2^{12} + 2^{13} + 2^{15})_{10} = (46793)_{10}$$

- modulo + segno. Il bit più significativo indica che la quantità è negativa. Basta quindi sottrarre 2^{15} al risultato precedente.

$$(1011011011001001)_2 = -(46793 - 2^{15})_{10} = -14025_{10}$$

- complemento a 2. Essendo negativo, dobbiamo complementare per conoscere il modulo.

$$\begin{aligned} (1011011011001001)_2 &= -f_2(1011011011001001)_2 = \\ &= -(0100100100110111)_2 = -(2^0 + 2^1 + 2^2 + 2^4 + 2^5 + 2^8 + 2^{11} + 2^{14})_{10} = -18743_{10} \end{aligned}$$

2) [12 bit]. $(-749)_{10} = \dots$

- Mostro le div. successive per 16.

Q	R
749/16	D
46/16	E → -(2ED)_H
2/16	2
0	•

- M+S: $-2ED_H = AED_H = 1010111011101_2 = 5355_8$

- Complemento a 2: $-2ED_H = f_2(2ED)_H = D13_H = 110100010011_2 = 6423_8$

Aritmetica binaria - Soluzioni

$$\begin{array}{r} 00000 \quad 1 \quad 1 \quad 0 \quad 1 \\ 1010 \quad 5 \quad F \quad 2 \quad 0 \quad B \quad + \\ 0001 \quad 2 \quad 5 \quad E \quad C \quad C \quad = \\ \hline 3) \quad 00110 \quad 0 \quad 0 \quad 1 \quad 1 \\ \quad 1011 \quad 8 \quad 5 \quad 0 \quad D \quad 7 \quad + \\ \quad 0011 \quad 1 \quad 2 \quad 6 \quad B \quad 9 \quad = \\ \hline \quad 1110 \quad 9 \quad 7 \quad 7 \quad 9 \quad 0 \end{array}$$

Quindi $A5F20B + 125ECC + 3126B9 = E97790$ (no overflow, riporto fuori dal MSB=0 in entrambe le somme).

- 4) Anche nel caso di rappresentazione con complemento a 2 non abbiamo overflow. Infatti i riporti fuori dal MSB e sul MSB coincidono in entrambe le somme. Quindi il risultato è lo stesso $A5F20B + 125ECC + 3126B9 = E97790$.
- 5) Il segno va determinato a priori. $F2C9 = -72C9$, quindi $F2C9 - 1F01 = -(72C9 + 1F01)$.

$$\begin{array}{r}
 1111 \ 0 \ 0 \\
 111 \ 2 \ C \ 9 \ + \\
 001 \ F \ 0 \ 1 \ = \\
 \hline
 001 \ 1 \ C \ A
 \end{array}$$

In questo caso abbiamo overflow, in quanto il riporto fuori dal bit più significativo del modulo è 1.

Floating point - Soluzioni

- 6) “Sottrazioni successive”: $(0.875)_{10} = 0.5 + 0.25 + 0.125 = 2^{-1} + 2^{-2} + 2^{-3} = (0.111)_2$, oppure
 “Moltiplicazioni successive”: $0.875 \times 2^3 = 7$, quindi $(0.875)_{10} = (111)_2 \times 2^{-3} = (0.111)_2$. Quindi $(77.875)_{10} = (1001101.111)_2 = \mathbf{1.001101111} \times 2^6$.
- Mantissa: 001101111;
 - esponente (eccesso -127): $6 + 127 = 133_{10} = 10000101_2$;
 - segno: 0

s	e										m																						
0	1	1	0	0	0	1	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4		2		9		B		C		0		0		0		0		0		0		0		0		0		0				

Il risultato è **429BC000**.

- 7) solito procedimento inverso

F	2				5				7				8				0				0				0							
1	1	1	1	0	0	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	e										m																					

- Mantissa: 101011110...;
- esponente (eccesso -127): $11100100_2 = 128 + 64 + 32 + 4 = (228)_{10} \rightarrow 228 - 127 = 101$;
- segno: 1

Quindi avremmo $-1.10101111_2 \times 2^{101} = -110101111_2 \times 2^{93} = -(431)_{10} \times 2^{93}$.

Possiamo ottenere la notazione desiderata con le approssimazioni conosciute:

$$-431 \times 2^3 \times 2^{90} \approx -431 \times 2^3 \times 10^{27} = -3448 \times 10^{27} = \mathbf{-3.448 \times 10^{30}}$$

Porte logiche e algebra booleana - Soluzioni

- 8) Riassumendo C viene replicato in C_1 e T in T_1 . Quando però il bit di controllo è 1, T_1 viene negato. La tabella di verità per il CNOT è la seguente:

C	T	C_1	T_1
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

9)

X	Y	Z	X_1	Y_1	Z_1
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

Algoritmi e programmazione JavaScript - Soluzioni

- 10) Se $a = [0, 8, -99.2, 7]$, allora il metodo deve restituire un array

$$r = [0, 0, 8, 8, -99.2, -99.2, 7, 7]$$

```
function replica(a){
  var r = new Array(a.length*2);
  for(var i=0; i<a.length; i++){
    r[2*i]=a[i];
    r[2*i+1]=a[i];
  }
  return r;
}
```

o più compattamente:

```
function replica(a){
  var r = new Array(a.length*2);
  for(var i=0; i<a.length; i++)
    r[2*i]=r[2*i+1]=a[i];
  return r;
}
```