Rappresentazione di un naturale in binario

Sia dato il numero naturale 87. Rappresentarlo su un byte, se possibile.

Soluzione

Prima di tutto occorre sapere se il numero è esprimibile su 8 bit. L'intervallo di rappresentazione dei naturali su 8 bit è $[0, 2^8-1] = [0, 255]$. Il numero 87 rientra in tale intervallo, quindi è esprimibile su 8 bit.

Applichiamo quindi l'algoritmo div&mod per portarlo in base 2:

q0 = 87

q1 = q0 div 2 = 43 **a0** = q0 mod 2 = **1**

q2 = q1 div 2 = 21 **a1** = q1 mod 2 = **1**

q3 = q2 div 2 = 10 **a2** = q2 mod 2 = **1**

q4 = q3 div 2 = 5 **a3** = q3 mod 2 = **0**

q5 = q4 div 2 = 2 **a4** = q4 mod 2 = **1**

q6 = q5 div 2 = 1 **a5** = q5 mod 2 = **0**

q7 = q6 div 2 = **0** **a6** = q6 mod 2 = **1**

Quindi 87 = $(\mathbf{1010111})_2$. Il numero sta su 7 bit, se vogliamo rappresentarlo su 8 basta aggiungere un bit a 0 in testa: 87 = $(\mathbf{01010111})_2$.

Controprova con metodo della sommatoria:

$0\ 2^7 + 1\ 2^6 + 0\ 2^5 + 1\ 2^4 + 0\ 2^3 + 1\ 2^2 + 1\ 2^1 + 1\ 2^0 = 64 + 16 + 4 + 2 + 1 = 87.$

Conversione da binario ad esadecimale

Portare il numero in base 2 $(01011101)_2$ in base 16.

Soluzione

Esiste una scorciatoia per convertire da base 2 a base 16 e viceversa. Ogni cifra in base 16 è rappresentata da 4 bit. Quindi basta raggruppare i bit a gruppi di 4, e convertire singolarmente ogni gruppo di quattro bit in una cifra esadecimale. Per convertire i gruppi di 4 bit si può utilizzare il metodo della sommatoria, tenendo in mente che $10 = (A)_{16}$, $11 = (B)_{16}$, ecc.

$(01011101)_2 = (0101|1101)_2$.

$(0101)_2 = 0\ 2^3 + 1\ 2^2 + 0\ 2^1 + 1\ 2^0 = 4 + 1 = 5 = (5)_{16}$

$(1101)_2 = 1\ 2^3 + 1\ 2^2 + 0\ 2^1 + 1\ 2^0 = 8 + 4 + 1 = 13 = (D)_{16}$

Quindi $(01011101)_2 = (\mathbf{5D})_{16}$.

Controprova con metodo della sommatoria:

$5\ 16^1 + 13\ 16^0 = 80 + 13 = 93.$

$0\ 2^7 + 1\ 2^6 + 0\ 2^5 + 1\ 2^4 + 1\ 2^3 + 1\ 2^2 + 0\ 2^1 + 1\ 2^0 = 64 + 16 + 8 + 4 + 1 = 93.$

Rappresentazione di un intero in complemento a due

Sia dato il numero a = -107. Rappresentarlo su un byte in complemento a due, se possibile.

Soluzione

L'intervallo di rappresentazione in complemento a due su 8 bit è [-2-7, 2-7-1] = [-128, 127]. Il numero -107 rientra in tale intervallo, quindi è esprimibile su 8 bit.

Prima di tutto occorre trasformare l'intero nel naturale che ha la stessa rappresentazione:

2^8 - |a| = 256 - |-107| = 149.

Adesso andiamo a rappresentare 149 su 8 bit con il metodo div&mod:

q0 = 149

q1 = q0 div 2 = 74 **a0** = q0 mod 2 = **1**

q2 = q1 div 2 = 37 **a1** = q1 mod 2 = **0**

q3 = q2 div 2 = 18 **a2** = q2 mod 2 = **1**

q4 = q3 div 2 = 9 **a3** = q3 mod 2 = **0**

q5 = q4 div 2 = 4 **a4** = q4 mod 2 = **1**

q6 = q5 div 2 = 2 **a5** = q5 mod 2 = **0**

q7 = q6 div 2 = 1 **a6** = q6 mod 2 = **0**

q8 = q7 div 2 = **0** **a7** = q7 mod 2 = **1**

Quindi la rappresentazione in complemento a due di a è A = **10010101**. È importante verificare che il bit più significativo (quello in testa) sia 1, perché esso rappresenta il bit del segno, che nel caso di -107 deve essere negativo.

Controprova con metodo della sommatoria:

$1\ 2^7 + 0\ 2^6 + 0\ 2^5 + 1\ 2^4 + 0\ 2^3 + 1\ 2^2 + 0\ 2^1 + 1\ 2^0 = 128 + 16 + 4 + 1 = 149$.

-(256 – 149) = -107.

Somma di due naturali tramite le loro rappresentazioni

Siano date le rappresentazioni su un byte dei naturali a e b, rispettivamente A=01000111 e B=10011101. Si calcoli la rappresentazione C della somma c = a + b.

Soluzione

Si esegue l'operazione in colonna con calcoli binari, ovvero tenendo in mente che 1+0 = 1 e porto 0, 1+1=0 e porto 1, ecc.

```
A = 01000111 +

B = 10011101 =

    ________

      11111 (riporti)

    11100100
```

Non c'è un riporto finale, quindi possiamo concludere che il risultato **11100100** è la corretta rappresentazione C della somma c = a + b. Se ci fosse stato un riporto finale, avrebbe significato che c non poteva essere rappresentato su un byte.

Somma di due interi tramite le loro rappresentazioni in complemento a due

Siano date le rappresentazioni su un byte degli interi a e b, rispettivamente A=01100111 e B=10011100. Si calcoli la rappresentazione C della somma c = a + b.

Soluzione

Si esegue l'operazione in colonna con calcoli binari sulle rappresentazioni in complemento a due.

```
A = 01100111 +

B = 10011100 =

    ________

   111111   (riporti)

    00000011
```

C'è un riporto finale che viene scartato. Esso non indica una condizione di overflow, perché la somma non è tra naturali, ma tra interi in complemento a due. Nel caso degli interi in complemento a due, la condizione di overflow si capisce confrontando il segno degli operandi con quello del risultato. Se il segno degli operandi è concorde, ma il risultato è discorde da loro, allora c'è overflow. In tutte le altre condizioni, non c'è overflow. In questo caso i segni degli operandi sono discordi, quindi non c'è stato overflow. Possiamo concludere che **00000011** è la corretta rappresentazione C della somma c = a + b.