

Problema 1.4

Convertire i seguenti numeri nella notazione binaria ad 8 bit: 23_{10} , 34_5 , 63_{10} , 63_8 . (Per 34_5 , passare prima alla notazione decimale).

Soluzione

- $N = 23_{10}$

$$\begin{array}{rclcl} S_0 & = & 23 & = & a_0 + 2 \cdot S_1 = 1 + 2 \cdot 11 \\ S_1 & = & 11 & = & a_1 + 2 \cdot S_2 = 1 + 2 \cdot 5 \\ S_2 & = & 5 & = & a_2 + 2 \cdot S_3 = 1 + 2 \cdot 2 \\ S_3 & = & 2 & = & a_3 + 2 \cdot S_4 = 0 + 2 \cdot 1 \\ S_4 & = & 1 & = & a_4 + 2 \cdot S_5 = 1 + 2 \cdot 0 \\ S_5 & = & 0 & = & a_5 + 2 \cdot S_6 = 0 + 2 \cdot 0 \\ S_6 & = & 0 & = & a_6 + 2 \cdot S_7 = 0 + 2 \cdot 0 \\ S_7 & = & 0 & = & a_7 + 2 \cdot S_8 = 0 + 2 \cdot 0 \end{array}$$

Si ottiene $N = 23_{10} = 00010111_2$.

Ogni singola somma parziale S_{i+1} si ottiene dividendo per 2 la somma precedente S_i , il resto della divisione fornisce la cifra a_i .

L'iterazione $S_i = a_i + 2S_{i+1}$ è stata ripetuta, per completezza, fino all'ultima cifra richiesta. In realtà il processo può essere interrotto alla prima somma parziale 0 ($S_5 = 0$) dato che le cifre successive risultano ovviamente nulle.

- $N = 34_5$

La preliminare trasformazione in decimale fornisce:

$$\begin{array}{rcl} S_1 & = & a_1 = 3 \\ S_0 & = & a_0 + 5 \cdot S_1 = 4 + 5 \cdot 3 = 19 \end{array}$$

La successiva trasformazione in binario:

$$\begin{array}{rclcl} S_0 & = & 19 & = & a_0 + 2 \cdot S_1 = 1 + 2 \cdot 9 \\ S_1 & = & 9 & = & a_1 + 2 \cdot S_2 = 1 + 2 \cdot 4 \\ S_2 & = & 4 & = & a_2 + 2 \cdot S_3 = 0 + 2 \cdot 2 \\ S_3 & = & 2 & = & a_3 + 2 \cdot S_4 = 0 + 2 \cdot 1 \\ S_4 & = & 1 & = & a_4 + 2 \cdot S_5 = 1 + 2 \cdot 0 \end{array}$$

Si ottiene $N = 34_5 = 19_{10} = 00010011_2$.

- $N = 63_{10}$

$$\begin{aligned}
 S_0 &= 63 = a_0 + 2 \cdot S_1 = 1 + 2 \cdot 31 \\
 S_1 &= 31 = a_1 + 2 \cdot S_2 = 1 + 2 \cdot 15 \\
 S_2 &= 15 = a_2 + 2 \cdot S_3 = 1 + 2 \cdot 7 \\
 S_3 &= 7 = a_3 + 2 \cdot S_4 = 1 + 2 \cdot 3 \\
 S_4 &= 3 = a_4 + 2 \cdot S_5 = 1 + 2 \cdot 1 \\
 S_5 &= 1 = a_5 + 2 \cdot S_6 = 1 + 2 \cdot 0
 \end{aligned}$$

Si ottiene $N = 63_{10} = 00111111_2$.

Si riduce il numero di iterazioni se si passa prima alla notazione esadecimale (utile per numeri elevati) e successivamente alla binaria (passaggio facile).

$$\begin{aligned}
 S_0 &= 63 = a_0 + 16 \cdot S_1 = 15 + 16 \cdot 3 \\
 S_1 &= 3 = a_1 + 16 \cdot S_2 = 3 + 16 \cdot 0
 \end{aligned}$$

Dato che 15 decimale si rappresenta con la cifra **F** esadecimale si ottiene $N = 63_{10} = 3F_{16}$.

Dato che una cifra esadecimale si rappresenta con quattro cifre binarie ($3_{16} = 0011_2$ e $F_{16} = 1111_2$) si ottiene ancora $N = 63_{10} = 3F_{16} = 00111111_2$.

Dato che il motivo essenziale per cui si utilizza il codice esadecimale è quello di rappresentare il codice binario in forma “compatta” è utile ricordare “a memoria” la corrispondenza tra ogni cifra esadecimale e la corrispondente rappresentazione binaria a quattro cifre.

Dec.	Esad.	Binario
0 =	0 =	0000
1 =	1 =	0001
2 =	2 =	0010
3 =	3 =	0011
4 =	4 =	0100
5 =	5 =	0101
6 =	6 =	0110
7 =	7 =	0111
8 =	8 =	1000
9 =	9 =	1001
10 =	A =	1010
11 =	B =	1011
12 =	C =	1100
13 =	D =	1101
14 =	E =	1110
15 =	F =	1111

Ricordare a memoria le corrispondenze sopra riportate significa rendere praticamente inutile la scrittura della notazione binaria quando si dispone di quella esadecimale.

Il codice $3F_{16}$ mostra un dato binario a 8 bit, con la cifra più significativa 0, la meno significativa 1, ecc...

- $N = 63_8$

Anche questo tipo di rappresentazione ha una semplice corrispondenza con il codice binario. Ogni cifra ottale corrisponde a tre cifre binarie

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

In questo caso si ottiene $N = 63_8 = 110011_2$.
Valgono le stesse considerazioni fatte per il codice esadecimale.