I SISTEMI DI NUMERAZIONE
E LA NUMERAZIONE BINARIA

Indice

Introduzione

Per sistema di numerazione si intende un insieme di simboli di rappresentazione dei numeri e di regole per contare ed eseguire operazioni.

Nel corso della storia l'uomo seppe inventare mezzi pratici che gli permisero di designare, prima oralmente e poi per iscritto, molti numeri con pochi simboli. A tal fine gli fu necessaria una "scala convenzionale" di simboli, che ora noi chiamiamo "base", per classificare numeri sempre più grandi, evitando sforzi di memoria o di rappresentazione macchinosa.

Nel corso della storia l'uomo ha usato diverse basi e tuttora sono in uso, o se ne trovano tracce, sistemi di base 2, 3, 4, 5, 6, 8, 10, 12, 16, 20 e 60.

Osserviamo subito però che quando diciamo che un popolo usa, ad esempio, la base 5, intendiamo soltanto dire che esso costruisce i nomi dei numeri usando in maniera cospicua la parola che indica 5; una vera base richiede però, che si usino nomi speciali per 5, $25 = 5^2$, $125 = 5^3$,...,e ciò non si trova in nessuna lingua attuale e forse non si è mai verificato in alcuna lingua.

Alla base dieci, che l'uomo primitivo ha scelto per ovvie ragioni antropomorfiche, sono state fatte diverse critiche a partire dal seicento e molti hanno proposto basi alternative da adottare. Blaise Pascal, che sembra sia stato il primo a realizzare che un qualunque numero intero maggiore di 1 può essere usato come base, avrebbe scelto la base 12; E. Wiegel proponeva invece, a partire dal 1673, la base 4; il re Carlo XII di Svezia pensò, all'inizio del settecento, di introdurre la base 8 nel suo regno ecc….

Il sistema decimale

Il sistema di numerazione oggi più diffuso è quello decimale, dove i numeri inferiori o uguali a dieci, come pure le potenze successive di dieci, ricevono ciascuno un nome individuale, mentre i nomi dei numeri intermedi sono parole composte a partire dalle precedenti, seguendo un principio additivo o moltiplicativo. L'adozione, quasi universale, della base dieci è stata indubbiamente imposta dall'anatomia delle mani, perché sulle dieci dita l'uomo ha imparato a contare. Escluso però il merito anatomico, la base dieci presenta pochi vantaggi da un punto di vista matematico o pratico. Tuttavia essa presenta un vantaggio indiscusso in confronto a basi più grandi come venti o sessanta, poiché il suo ordine di grandezza è soddisfacente per la memoria umana dato che bastano pochi nomi di numeri, ed è anche vantaggiosa in confronto alle basi piccole come due o tre perché permette di evitare sforzi di rappresentazione derivanti dalla grande quantità di cifre necessarie per esprimere numeri relativamente piccoli. In questo caso però sarebbe stata comunque migliore come base universale di numerazione la base dodici in quanto costituisce un buon ordine di grandezza per la memoria e ha molti più divisori rispetto al dieci (4 invece di 2). L'abitudine di contare per dieci è talmente ancorata nella tradizione, che la scelta della base si rivela indistruttibile.

Il sistema binario

La numerazione binaria, che adotta la base due e utilizza solo le cifre “0” e “1” è, oltre a quella decimale, di impiego piuttosto frequente. Si tratta di una numerazione semplicissima, la più arcaica e insieme la più moderna numerazione posizionale, tant’è che la potenza del calcolo dei computer deriva proprio dall’utilizzo del codice binario. Infatti questo sistema trova corrispondenza con i componenti elettronici che funzionano in on/off, cioè con le condizioni di acceso/spento oppure di si/no.

Per spiegare l’aritmetica binaria, il grande filosofo e matematico tedesco Leibniz, che è stato il primo sostenitore di questa numerazione, scrive nel 1703: “Invece della progressione di dieci in dieci, impiego da molti anni la progressione più semplice di tutte, che va di due in due, ritenendo che sia perfettamente adeguata alla scienza dei numeri. Utilizzo solo due caratteri, “0” e “1” e poi, quando sono arrivato a due, ricomincio”.

Un altro sistema entrato nell’uso comune in ambito informatico per esempio per il codice RGB è quello **esadecimale**, cioè in base 16, le cui 16 cifre sono:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

in cui le lettere hanno i seguenti “valori” nel sistema decimale:

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

Tali sistemi di numerazione sono **sistemi posizionali**, cioè il valore delle cifre dipende dalla posizione che occupano nel numero. Spostandosi a sinistra di una posizione il valore della cifra viene moltiplicato per dieci (nel caso di un sistema decimale) o per due (nel caso di un sistema binario), ma in ogni caso sempre per la sequenza “10” (da leggere criticamente in funzione della base).

(Un sistema non posizionale è, ad esempio, quello degli antichi numeri romani:

XVIII = 18, XIX = 19, …)

In tali sistemi, il numero N viene rappresentato come:

$N = a_0 \times B^0 + a_1 \times B^1 + a_2 \times B^2 + \ldots + a_n \times B^n \ldots$

Dove gli a_i = cifre del sistema (nel caso decimale ogni a_i può assumere uno tra i seguenti valori: 0, 1, …, 9), B = base

Per esempio, il numero 350 in base **dieci** si rappresenta come: (← = verso di lettura)

350 = $0 \times 10^0 + 5 \times 10^1 + 3 \times 10^2 = 300 + 50 + 0$

in base **due** si rappresenta come:

101011110 = $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 0 \times 2^7 + 1 \times 2^8 = 2^8 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1 = 256 + 0 + 64 + 0 + 16 + 8 + 4 + 2 + 0 = 350$

mentre in base **sedici** si rappresenta come:

$\mathbf{15E} = 14 \times 16^0 + 5 \times 16^1 + 1 \times 16^2 = 256 + 80 + 14 = 350.$

Nel seguito utilizzeremo la notazione $(15E)_{16} = (350)_{10}$ ad indicare che il numero 15E in base 16 corrisponde al numero 350 in base 10

Ci occuperemo principalmente della numerazione binaria, del suo rapporto con la numerazione decimale e delle sue regole aritmetiche; ci occuperemo degli altri sistemi di numerazione, invece, solo in rapporto al sistema decimale.

Il sistema decimale considera dieci cifre che sono: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Il sistema binario ne considera due: 0, 1.

Per rappresentare una quantità numerica viene usata una sequenza di simboli suddivisa in:

a) Parte intera (N_i)

b) Parte frazionaria (N_f)

in modo che il numero N sia espresso dalla seguente relazione:

$$N = N_i + N_f$$

Esempio:

$$N = 1936.27 \quad \Rightarrow \quad N_i = 1936 \,;\, N_f = 0.27$$

Un numero N può essere rappresentato

1) Nel sistema decimale come:

$$N= \underbrace{a_0 \times 10^0 + a_1 \times 10^1 + a_2 \times 10^2 +\ldots+ a_n \times 10^n}_{N_i} + \underbrace{a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} +\ldots+ a_{-n} \times 10^{-n}}_{N_f}$$

Esempio:

$N = 809.205 \quad \Rightarrow \quad N_i = 809 = 9 \times 10^0 + 0 \times 10^1 + 8 \times 10^2$

$N_f = 0{,}205 = 2 \times 10^{-1} + 0 \times 10^{-2} + 5 \times 10^{-3} \quad \Rightarrow$

$N = N_i + N_f = 9 \times 10^0 + 0 \times 10^1 + 8 \times 10^2 + 2 \times 10^{-1} + 0 \times 10^{-2} + 5 \times 10^{-3}$

2) Nel sistema binario come:

$$N = \underbrace{a_0 \times 2^0 + a_1 \times 2^1 + a_2 \times 2^2 +\ldots+ a_n \times 2^n}_{N_i} + \underbrace{a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} +\ldots+ a_{-n} \times 2^{-n}}_{N_f}$$

Esempio:

$N = 11010.101 \quad \Rightarrow \quad N_i = 11010 = 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 +1 \times 2^4$

$N_f = 0.101 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \quad \Rightarrow$

$N = N_i + N_f = 0\times2^0 + 1\times2^1 + 0\times2^2 + 1\times2^3 + 1\times2^4 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

In generale, in un sistema posizionale, la forma normalizzata di un numero è:

$N = m \times b^n$ dove:

m = mantissa, b = base del sistema, n = esponente, b^n = peso della cifra o della mantissa.

Conversione di un numero da base 10 a base 2 e viceversa

Supponiamo di voler convertire nel sistema binario il numero decimale

N = 47.485 (= 47 + 0.485)

Si procede come segue. Trasformiamo prima la parte intera: $N_i = 47$

47 : 2 =	con Resto = 1
23 : 2 =	con R = 1
11 : 2 =	con R = 1
5 : 2 =	con R = 1
2 : 2 =	con R = 0
1	R = 1

verso di lettura ⟹ $(\mathbf{47})_{10} = (\mathbf{101111})_2$

Il procedimento si arresta quando, dopo aver effettuato n divisioni per 2, si ottiene come risultato un numero più piccolo di due (ovvero 0 oppure 1) che, pertanto, resta indiviso.

È ora possibile verificare il risultato ottenuto trasformandolo da binario a decimale, secondo la regola precedentemente riportata:

$(\mathbf{101111})_2 = 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 32 + 8 + 4 + 2 + 1 = (\mathbf{47})_{10}$

Vogliamo ora trasformare la parte frazionaria: $N_f = 0.485$

0	0.485 × 2 =
0	0.970 × 2 =
1	1.940 × 2 =
1	3.880 × 2 =
1	7.760 × 2 =
1	15.520
	…

verso di lettura ⟹ $(\mathbf{0.485})_{10} = (\ \mathbf{0.01111\ldots})_2$

Si osservi che se nella moltiplicazione per 2 si ottiene un numero decimale la cui cifra prima della virgola è diversa da 0 e da 1 allora: se è pari la si sostituisce con 0 mentre se è dispari la si sostituisce con 1.

Il procedimento si arresta quando la parte decimale si annulla. In caso contrario si può arrestare dopo n passi ottenendo però un numero approssimato. In altre parole, poiché la trasformazione da a decimale a binario della parte frazionaria può produrre un numero di cifre anche di molto superiore, potremmo fermarci ad un certo grado di approssimazione stabilendo il numero di cifre binarie volute: ad esempio, se nella conversione del numero $(47.485)_{10}$ si richiedessero "solo" tre cifre nella parte frazionaria, il risultato voluto sarebbe $(101111.011)_2$.

È ora possibile verificare il risultato ottenuto trasformandolo da binario a decimale, secondo la regola precedentemente riportata:

$$(\mathbf{0.01111})_2 = 0 \times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} + 1\times 2^{-5} =$$

$$= \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} = \frac{15}{32} = \mathbf{0.46875}$$

Il risultato ottenuto è quindi un numero approssimato. Se si continuasse però si otterrebbe un numero sempre più vicino al numero iniziale 0,485.

In conclusione si ottiene:

$$N = (\mathbf{47.485})_{10} = (\mathbf{101111.01111})_2$$

<u>Esempio 1.</u> Vogliamo convertire nel sistema binario il numero decimale

N = 109.3125 (= 109 + 0.3125)

Trasformiamo dapprima la parte intera: $N_i = 109$

109 : 2 = con Resto = 1

54 : 2 = con R = 0

27 : 2 = con R = 1

13 : 2 = con R = 1

6 : 2 = con R = 0

3 : 2 = con R = 1

1 R = 1

verso di lettura $\Rightarrow$ $(\mathbf{109})_{10} = (\mathbf{1101101})_2$

verifichiamo la correttezza della conversione appena eseguita:

$(\mathbf{1101101})_2 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0 = 64 + 32 + 8 + 4 + 1 = (\mathbf{109})_{10}$

Ora Trasformiamo la parte frazionaria: $N_f = 0.3125$

0	$0.3125 \times 2 =$
0	$0.6250 \times 2 =$
1	$1.250 \times 2 =$
0	$2.50 \times 2 =$
1	5.0

verso di lettura $\Longrightarrow$ $(0.3125)_{10} = (\,0.0101)_2$

Verifichiamo la correttezza della conversione:

$$(\mathbf{0.0101})_2 = 0 \times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-4} = \frac{1}{4} + \frac{1}{16} = \frac{5}{16} = \mathbf{0.3125}$$

La conversione trovata è esatta (cioè non approssimata); in generale la conversione termina quando nel processo moltiplicativo in colonna la parte frazionaria diventa nulla (se la parte frazionaria è nulla la moltiplicazione di un numero intero per 2 produce sempre un numero pari, per cui continuando nel nostro processo avremmo solo zeri). In conclusione si ottiene:

$$N = (\mathbf{109.3125})_{10} = (\mathbf{101111.01111})_2$$

<u>Esempio 2.</u> Trasformare il numero 52 da decimale a binario:

52: 2=	con Resto = 0
26: 2=	con R = 0
13: 2=	con R = 1
6: 2=	con R = 0
3: 2=	con R = 1
1	R = 1

verso di lettura $\Longrightarrow$ $(52)_{10} = (110100)_2$

Verifica: $(\mathbf{110100})_2 = 2^5 + 2^4 + 2^2 = 32 + 16 + 4 = (\mathbf{52})_{10}$

Esempio 3. Convertire il numero 0.324 da decimale a binario:

0 $\quad 0.324 \times 2 =$

0 $\quad 0.648 \times 2 =$

1 $\quad 1.296 \times 2 =$

0 $\quad 2.592 \times 2 =$

1 $\quad 5.184 \times 2 =$

0 $\quad 10.368$

… …

verso di lettura $\Rightarrow$ $(0.324)_{10} = (\ 0.01010\ldots)_2$

Esempio 4. Trasformare il numero 1001100.1011 da binario a decimale:

$N = (\mathbf{1001100.1011})_2 = 1\times 2^2 + 1\times 2^3 + 1\times 2^6 + 1\times 2^{-1} + 1\times 2^{-3} + 1\times 2^{-4} =$

$$= 4 + 8 + 64 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = 76 + \frac{11}{16} = 76 + 0.6875 = (\mathbf{76.6875})_{10}$$

Verifica:

76 : 2 = con Resto = 0

38 : 2 = con R = 0

19 : 2 = con R = 1

9 : 2 = con R = 1

4 : 2 = con R = 0

2 : 2 = con R = 0

1 R = 1

verso di lettura $\Rightarrow$ $(76)_{10} = (1001100)_2$

0 $\quad 0.6875 \times 2 =$

1 $\quad 1.3750 \times 2 =$

0 $\quad 2.750 \times 2 =$

1 $\quad 5.50 \times 2 =$

1 $\quad 11.0$

…

verso di lettura $\Rightarrow$ $(0.6875)_{10} = (\ 0.1011)_2$

In conclusione $N = (\mathbf{1001100.1011})_2 = (\mathbf{76.6875})_{10}$

Conversione in altri sistemi di numerazione

Siano dati due sistemi di numerazione: il primo in base b, il secondo in base B e supponiamo inoltre che b < B. Allora valgono le seguenti regole:

"da b a B": *per trasformare un numero N_b (in base b) nel corrispondente numero N_B in base B (quindi da una base più piccola ad una più grande) bisogna interpretare N_b in senso posizionale e trasformare gli ordini di grandezza delle posizioni (potenze di 10) nei corrispondenti valori nella nuova base B; il numero N_B è determinato dalla loro somma nel sistema in base B.*

"da B a b": *per trasformare un numero N_B (in base B) nel corrispondente numero N_b in base b (quindi da una base più grande ad una più piccola) bisogna dividere N_B ripetutamente per il numero b e determinare, in tal modo, un insieme di resti; il numero N_b è determinato dalla sequenza di cifre così ottenute, presa nell'ordine inverso.*

Esempio 1. Trasformare il numero 52 dal sistema decimale al sistema ternario:

52: 3= con Resto = 1

17: 3= con R = 2

5: 3= con R = 2

1 → R = 1

verso di lettura ⇒ $(\mathbf{52})_{10} = (\mathbf{1221})_2$

Si osservi che le cifre di una numerazione ternaria sono {0, 1, 2} e quindi il resto della divisione può essere 0, 1 oppure 2. Dunque anche il processo di divisione termina quando si raggiunge, come quoziente intero, un valore compreso fra 0 e 2.

Verifica: $(\mathbf{1221})_3 = 1\times 3^3 + 2 \times 3^2 + 2 \times 3^1 + 1 \times 3^0 = 27 + 18 + 6 + 1 = (\mathbf{52})_{10} = 52$

Esempio 2. Trasformare il numero 52 dal sistema decimale al sistema quaternario:

52: 4= con Resto = 0

13: 4= con R = 1

3 → R = 3

verso di lettura ⇒ $(\mathbf{52})_{10} = (\mathbf{310})_4$

Si osservi che le cifre di una numerazione ternaria sono {0,1,2,3} e quindi il resto della divisione può essere 0, 1, 2 oppure 3.

Verifica: $(\mathbf{310})_3 = 3\times 4^2 + 1 \times 4^1 + 0 \times 4^0 = 48 + 4 = (\mathbf{52})_{10} = 52$

Esempio 3. Trasformare il numero 52 dal sistema decimale al sistema esadecimale (ovvero in base sedici). Ricordiamo che la base sedici ha come simboli, ordinati in maniera crescente: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F per cui A equivale al valore 10 nel sistema decimale, B ad 11, C a 12, D a 13, E a 14 ed F a 15.

52 : 16 = 3 con Resto = 4

3 → R = 3

verso di lettura ⟹ $(\mathbf{52})_{10} = (\mathbf{34})_{16}$

Verifica: $(\mathbf{34})_{16} = 3 \times 16^1 + 4 \times 16^0 = 52 = (\mathbf{52})_{10}$

Esempio 4. Trasformare il numero 253 dal sistema decimale a quello esadecimale:

253 : 16 = 15 con Resto = 13 (=D)

15 → R = 15 (=F)

verso di lettura ⟹ $(\mathbf{253})_{10} = (\mathbf{FD})_{16}$

Verifica: $(\mathbf{FD})_{16} = 15 \times 16^1 + 13 \times 16^0 = 240 + 13 = (\mathbf{253})_{10}$

Esempio 4. Trasformare il numero 3107 dal sistema decimale a quello esadecimale:

3107 : 16 = 194 con Resto = 3

194 : 16 = 12 R = 2

12 → R = 12 (=C)

verso di lettura ⟹ $(\mathbf{3107})_{10} = (\mathbf{C23})_{16}$

Verifica: $(\mathbf{C23})_{16} = 12 \times 16^2 + 2 \times 16^1 + 3 \times 16^0 = 3072 + 32 + 3 = (\mathbf{3107})_{10}$

Aritmetica Binaria

Le regole che caratterizzano l'artimetica binaria sono analoghe alle regole ben conosciute che valgono nel sistema decimale, con il necessario adattamento derivante dall'uso limitato ai due simboli 0 e 1. Sono riportate di seguito le tabelle con tali regole - addizione, moltiplicazione e sottrazione binaria - con alcuni esempi esplicativi su ciascuna operazione, inclusa la divisione tra numeri binari.

SOMMA

+	**0**	**1**
0	0	1
1	1	10

ovvero 0 con riporto di 1

Osservazione. Nell'operare la somma il valore "10" (da leggere uno-zero) è da intedere scrivendo "0" con riporto di "1", come nell'usuale addizione fra numeri decimali.

Esempio 1. Eseguire la somma fra i numeri binari 101 e 111:

$$\begin{array}{ccccl} {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \text{(riporti)} \\ & 1 & 0 & 1 & + \\ & 1 & 1 & 1 & = \\ \hline 1 & 1 & 0 & 0 & \end{array}$$

Esempio 2. Eseguire la somma fra i numeri binari 10111 e 11110:

$$\begin{array}{cccccccl} {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & \text{(riporti)} \\ & 1 & 0 & 1 & 1 & 1 & + \\ & 1 & 1 & 1 & 1 & 0 & = \\ \hline 1 & 1 & 0 & 1 & 0 & 1 & \end{array}$$

Esempio 3. Eseguire la somma fra i numeri binari 1101 e 111:

$$\begin{array}{cccccl} {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \text{(riporti)} \\ & 1 & 1 & 0 & 1 & + \\ & & 1 & 1 & 1 & = \\ \hline 1 & 0 & 1 & 0 & 0 & \end{array}$$

SOTTRAZIONE

-	**0**	**1**
0	0	1
1	1	0

La tabella a fianco è da leggere prendendo un valore sulla 1° colonna a sinistra e sottraendo a questo un valore sulla 1° riga in alto; così il valore "1" scritto nell'incrocio tra 2° riga e 1° colonna è determinato dall'operazione 1 – 0 .

con prestito di 1

Da notare che negli esempi che seguono il "prestito" di 1 da una posizione a quella immediatamente più piccola corrisponde ad un prestito di 10 in quella posizione. Inotre si consiglia di ricordare sempre che vale 10 – 1 = 1.

Esempio 1. Eseguire la sottrazione fra i numeri binari 1101 e 1011:

$$\begin{array}{ccccl} & & {\scriptstyle 10} & & \text{(prestiti)} \\ 1 & 1 & 0 & 1 & - \\ 1 & 0 & 1 & 1 & = \\ \hline 0 & 0 & 1 & 0 & \end{array}$$

Esempio 2. Eseguire la sottrazione fra i numeri binari 10101 e 1011:

$$
\begin{array}{ccccccl}
 & {\scriptstyle 10} & & {\scriptstyle 10} & & & \text{(prestiti)} \\
1 & 0 & 1 & 0 & 1 & - & \\
 & 1 & 0 & 1 & 1 & = & \\
\hline
 & 1 & 0 & 1 & 0 & &
\end{array}
$$

Esempio 3. Eseguire la sottrazione fra i numeri binari 11000 e 111:

$$
\begin{array}{ccccccl}
 & & {\scriptstyle 10} & {\scriptstyle 10} & {\scriptstyle 10} & & \text{(prestiti)} \\
1 & 1 & 0 & 0 & 0 & - & \\
 & & 1 & 1 & 1 & = & \\
\hline
1 & 0 & 0 & 0 & 1 & &
\end{array}
$$

MOLTIPLICAZIONE

×	**0**	**1**
0	0	0
1	0	1

Nella moltiplicazione binaria, il primo numero viene moltiplicato di volta in volta per le cifre del secondo numero: se la cifra è 0 allora si otterranno tutti 0, altrimenti si otterrà il numero stesso; poi si sommano i numeri ottenuti secondo la regola precedentemente data della somma, tenendo conto dei vari riporti.

Esempio 1. Eseguire la moltiplicazione fra i numeri binari 1101 e 101:

$$
\begin{array}{cccccccl}
 & & & 1 & 1 & 0 & 1 & \times \\
 & & & & 1 & 0 & 1 & = \\
\hline
{\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & & \text{(riporti)} \\
 & & & 1 & 1 & 0 & 1 & + \\
 & & 0 & 0 & 0 & 0 & & + \\
 & 1 & 1 & 0 & 1 & & & \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 1 &
\end{array}
$$

Esempio 2. Eseguire la moltiplicazione fra i numeri binari 1011 e 1011:

$$
\begin{array}{cccccccl}
 & & & 1 & 0 & 1 & 1 & \times \\
 & & & 1 & 0 & 1 & 1 & = \\
\hline
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & \text{(riporti)} \\
 & & & 1 & 0 & 1 & 1 & + \\
 & & 1 & 0 & 1 & 1 & & + \\
 & 0 & 0 & 0 & 0 & & & + \\
1 & 0 & 1 & 1 & & & & \\
\hline
1 & 1 & 1 & 1 & 0 & 0 & 1 &
\end{array}
$$

Esempio 3. Eseguire la moltiplicazione fra i numeri binari 11110 e 1011:

$$
\begin{array}{ccccccccccl}
 & & & & 1 & 1 & 1 & 1 & 0 & \times & \\
 & & & & & 1 & 0 & 1 & 1 & = & \\
\hline
1 & 1 & 10 & 10 & 1 & 1 & & & & & \text{(riporti)} \\
 & & & & 1 & 1 & 1 & 1 & 0 & + & \\
 & & & 1 & 1 & 1 & 1 & 0 & & + & \\
 & & 0 & 0 & 0 & 0 & 0 & & & + & \\
 & 1 & 1 & 1 & 1 & 0 & & & & & \\
\hline
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & &
\end{array}
$$

Osservazione. Quando nel processo moltiplicativo la somma finale su una colonna è 10 o 11 il riporto è 1, se la somma è 100, 101, 110, …, 111 allora il riporto è 10, e così via. Pertanto nell'esempio 3:

la somma sulla prima colonna (iniziando da destra) è 0 $\rightarrow$ si scrive 0;
la somma sulla seconda colonna è 1 $\rightarrow$ si scrive 1;
la somma sulla terza col. è 10 $\rightarrow$ si scrive 0 e si riporta 1;
la somma sulla quarta col. (incluso il riporto precedente) è 11 $\rightarrow$ si scrive 1 e si riporta 1;
la somma sulla quinta col. (incluso il riporto precedente) è 100 $\rightarrow$ si scrive 0 e si riporta 10;
la somma sulla sesta col. (incluso il riporto precedente) è 100 $\rightarrow$ si scrive 0 e si riporta 10;
la somma sulla settima col. (incluso il riporto precedente) è 11 $\rightarrow$ si scrive 1 e si riporta 1;
la somma sulla ottava col. (incluso il riporto precedente) è 10 $\rightarrow$ si scrive 0 e si riporta 1;
la somma sulla nona col. (data dal solo riporto precedente) è 1 $\rightarrow$ si scrive 1.

DIVISIONE

La divisione nel sistema binario risulta più semplice perché per stabilire quante volte il divisore sia contenuto in un gruppo di cifre del dividendo non è necessario procedere per tentativi in quanto il risultato può essere soltanto 0 oppure 1.

Esempio 1. Eseguire la divisione fra i numeri binari 11011 e 11:

$$
\begin{array}{ccccc|cccc}
1 & 1 & 0 & 1 & 1 & 1 & 1 & & \\
\cline{6-9}
- & - & 0 & & & 1 & 0 & 0 & 1 \\
 & & - & 1 & & & & & \\
 & & & 1 & 1 & & & &
\end{array}
$$

(4°) 11 : 11 = 1 con resto 0
(3°) 1 : 11 = 0 con resto 1
(2°) 0 : 11 = 0 con resto 0
(1°) 11 : 11 = 1 con resto 0

Dunque 11011 : 11 = 1001 con resto 0 ovvero 1001 × 11 = 11011; facciamone le verifica:

$$
\begin{array}{cccccc}
 & 1 & 0 & 0 & 1 & \times \\
 & & & 1 & 1 & = \\
\hline
 & 1 & 0 & 0 & 1 & + \\
1 & 0 & 0 & 1 & & \\
\hline
1 & 1 & 0 & 1 & 1 &
\end{array}
$$

Esempio 2. Eseguire la divisione fra i numeri binari 1101011 e 1011:

$$
\begin{array}{ccccccc|cccc}
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ \cline{8-11}
1 & 0 & 1 & 1 & & & & 1 & 0 & 0 & 1 \\ \cline{1-4}
- & - & 1 & 0 & 0 & 1 & 1 & & & & \\
 & & & 1 & 0 & 1 & 1 & & & & \\ \cline{3-7}
 & & - & 1 & 0 & 0 & 0 & & & &
\end{array}
$$

(4°) 10011 : 1011 = 1 con resto 1000

(3°) 1001 : 1011 = 0 con resto 1001

(2°) 100 : 1011 = 0 con resto 100

(1°) 1101 : 1011 = 1 con resto 10

Dunque 1101011 : 1011 = 1001 con resto 1000, ovvero 1011 $\times$ 1001 + 1000 = 1101011; facciamone la verifica:

$$
\begin{array}{cccccccc}
 & & & 1 & 0 & 1 & 1 & \times \\
 & & & 1 & 0 & 0 & 1 & = \\ \hline
 & 1 & 1 & & & & & \text{(riporti)} \\
 & & & 1 & 0 & 1 & 1 & + \\
 & & 0 & 0 & 0 & 0 & & + \\
 & 0 & 0 & 0 & 0 & & & + \\
1 & 0 & 1 & 1 & & & & \\ \hline
1 & 1 & 0 & 0 & 0 & 1 & 1 &
\end{array}
$$

e sommando il resto

$$
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & 1 & + \\
 & & & 1 & 0 & 0 & 0 & = \\ \hline
1 & 1 & 0 & 1 & 0 & 1 & 1 &
\end{array}
$$

Esercizi.

1) Eseguire le seguenti somme nel sistema binario:
101011 + 10111 =
11111 + 101111 =
1100111 + 10111 =
10101111 + 1111111 =

2) Eseguire le seguenti sottrazioni nel sistema binario:
11101 - 101 =
110110 - 101101 =
1100111 - 101111 =
100000 - 10101 =

1) Eseguire le seguenti moltiplicazioni nel sistema binario:
1010 $\times$ 101 =
110101 $\times$ 1011 =
111011 $\times$ 10111 =
111111 $\times$ 11111 =

1) Eseguire le seguenti divisioni nel sistema binario:

11001 : 101 =	con Resto =
11111 : 110 =	con Resto =
1011011 : 1101 =	con Resto =
1111011 : 10011 =	con Resto =