# 4 Sistemi di numerazione e rappresentazione binaria dei numeri interi

La numerazione decimale è di tipo **posizionale** perché la quantità oltre che dal simbolo è definita anche in base alla posizione.
Al contrario della numerazione romana che è un sistema **additivo** perché il valore complessivo del numero è dato dalla somma dei valori dei simboli, indipendentemente dalla loro posizione.

**Un sistema di numerazione è definito da:**

- Un intero B detto BASE
- Un insieme di B simboli SB = {s0, ..., sB-1}, ognuno dei quali rappresenta le quantità 0,1,2,...,B−1

**TRASFORMARE NUMERI DALLA LORO BASE A DECIMALE**

**BINARIO**

$$(1100)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = (12)_{10}$$

**OTTALE**

$$(126)_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 = (86)_{10}$$

**DECIMALE**

$$(126)_{10} = 1 \cdot 10^2 + 2 \cdot 10^1 + 6 \cdot 10^0 = (126)_{10}$$

**ESADECIMALE**

$$(126)_{16} = 1 \cdot 16^2 + 2 \cdot 16^1 + 6 \cdot 16^0 = (294)_{10}$$



**CONVERSIONE DA BASE 10 A BASE B**

La **CONVERSIONE** di un numero da **base 10** a **base B** usa la tecnica delle **DIVISIONI SUCCESSIVE:**

1) Sia **N** il numero (in **base 10**) da convertire

2) Si calcola la divisione intera **N = N/B** e si mette da parte il resto **R** della divisione

3) Se **N > 0** si va al **PASSO 2**

4) Se **N = 0**, si riportano i vari **RESTI** da destra verso sinistra: essi rappresentano il numero convertito in **base B**

### Convertire 13 in base 2

| | |
|---|---|
| 13 | |
| 6 | 1 |
| 3 | 0 |
| 1 | 1 |
| 0 | 1 |

$$\Rightarrow (1101)_2 = 1 \cdot 2^2 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (13)_{10}$$

---

### Convertire 13 in base 5

| | |
|---|---|
| 13 | |
| 2 | 3 |
| 0 | 2 |

$$\Rightarrow (23)_5 = 2 \cdot 5^1 + 3 \cdot 5^0 = (13)_{10}$$

- Progettare e realizzare circuiti elettrici di tipo **ON/OFF** e molto più **SEMPLICE** ed **IMMEDIATO** rispetto a dover gestire diversi livelli di tensione
- I concetti **ON/OFF** possono essere rappresentati tramite **NUMERI**:
- **OFF = 0**
- **ON = 1**
- Il **SISTEMA DI NUMERAZIONE BINARIA** è la soluzione perfetta per rappresentare valori **ON/OFF**
- Individuata una tipologia di **INFORMAZIONE**, si possono inserire delle **REGOLE** non ambigue per **RAPPRESENTARE** l'informazione come **SEQUENZE BINARIE**

**TRASFORMARE NUMERI IN BASE 10 IN BINARIO**

$$P = p_{(n-1)}p_{(n-2)}\dots p_1 p_0 \,,\, p_{(i)} \in \{0,1\} \text{ e } i=0,\dots,n-1$$

$$\sum_{i=0}^{n-1} p_{(i)} \cdot 2^i$$

Numero di valori rappresentabili = $[0, 2^n)$

Esempio: $(54)_{10}$ -> $1*2^5+1*2^4+0*2^3+1*2^2+1*2^1+0*2^0$
1 1 0 1 1 0

**SOMMA DI NUMERI BINARI**

Per **SOMMARE** numeri binari ad 1 bit:

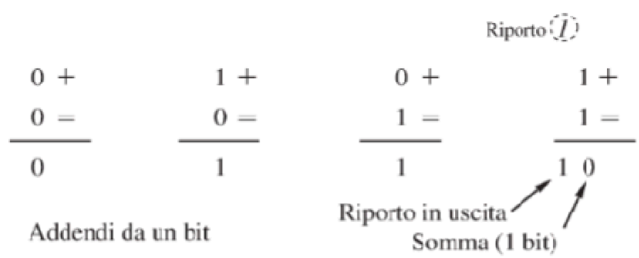


**Figura 1.4** - Addizione di numeri a un bit

Il **RIPORTO IN USCITA** della cifre precedente viene assegnato come **RIPORTO IN ENTRATA** alla successiva

**Per rappresentare il segno si usa 1 bit**
**0 = positivo**
**1 = negativo**

# Somma modulare

Definiamo la funzione **MODULO** nel modo seguente:

$$A \bmod n = \text{resto di } (A / n)$$

La **SOMMA MODULARE**:

$$(A + B) \bmod n = \text{resto di } ((A + B) / n)$$

Assumerà sempre valori compresi tra **0 e *n*-1**