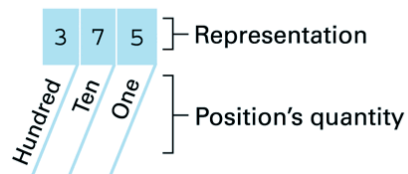


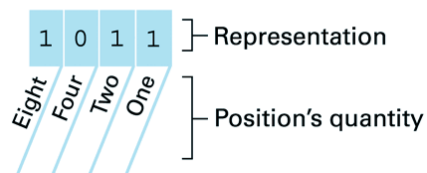
# Notazione binaria, ottale, esadecimale. Algebra di Boole.

## Base 10 e base 2

### a. Base ten system



### b. Base two system



## Rappresentazione decimale e binaria

- Base 10 → cifre da 0 a 9
- Base 2 → cifre 0 e 1
- Sequenza di cifre decimali

$$d_k d_{k-1} \dots d_1 d_0$$

→ numero intero

$$d_k \times 10^k + d_{k-1} \times 10^{k-1} + \dots d_1 \times 10 + d_0$$

- Esempio: 102 in base 10 è  $1 \times 100 + 0 \times 10 + 2 \times 1$
- In generale:  $\sum_{(k=n, n-1, \dots, 0)} d_k 10^k$

## Valore di una rappresentazione binaria

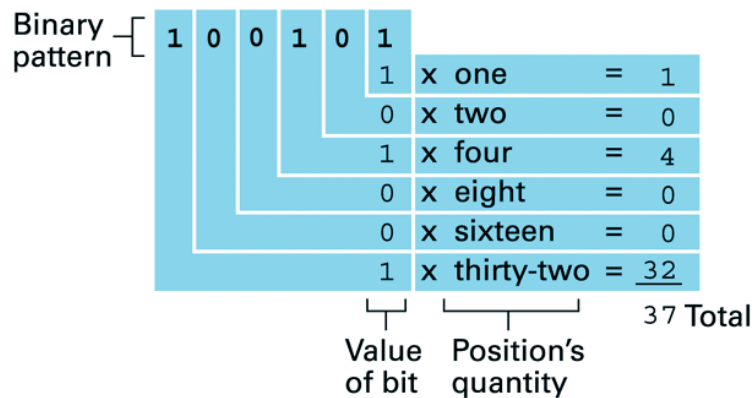
- Per un numero binario  $d_k d_{k-1} \dots d_1 d_0$
- Stesso procedimento ma su base 2:

$$\sum_{(k=n, n-1, \dots, 0)} d_k 2^k$$

- Esempio:

$$\begin{aligned} 0101101_2 &= 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 \\ &= 32 + 8 + 4 + 1 \\ &= 45_{10} \end{aligned}$$

## Valore di una rappresentazione binaria



## Rappresentazione binaria

- Valore minimo di una sequenza di  $n$  cifre binarie:  $000 \dots 0$  ( $n$  volte)  $= 0_{10}$
- Valore massimo:  $1111 \dots 111$  ( $n$  volte)  $= 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2^1 + 2^0 = 2^n - 1$
- Esempio con  $n=3$ :  $111 = 2^2 + 2 + 1 = 7 = 2^3 - 1$
- Da 0 a 8 (su 4 bit):  
0    1    2    3    4    5    6    7    8  
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000

## Kilo, Mega, Giga, Tera, ...

- Byte = 8 bit
- Kilo, dal greco khiloi ( $1000 = 10^3$ )
  - $2^{10} = 1024 = 1K$  (vicino a 1000)
- Mega, dal greco mega (grande)
  - $1.000.000 = 10^6$
  - $2^{20} = 1.048.576$
- Giga, dal latino gigas (gigante)
  - $1.000.000.000 = 10^9$
  - $2^{30}$
- Tera, dal greco tera (mostro)
  - $10^{12}$
  - $2^{40}$
- Peta, dal greco pente (5)
  - $1000^5 = 10^{15}$
  - $2^{50}$

## Notazione ottale (base 8)

- 8 simboli: 0, 1, 2, ..., 7
- Un simbolo per rappresentare ogni gruppo di 3 cifre binarie (ce ne sono 8 diversi)
- Es.: 101101010011 (binario)

101	101	010	011	
↑	↑	↑	↑	
5	5	2	3	→ 5523 (ottale)
- Di solito lunghezza multipla di 3
  - Es.: 3 simboli per 8 bit

## Notazione esadecimale

- 16 simboli: 0, 1, 2, ..., 9, A, B, ..., F
- Un simbolo per rappresentare ogni gruppo di 4 cifre binarie (ce ne sono 16 diversi)
- Es.: 101101010011
- Di solito lunghezza multipla di 4
- Es.: 3 simboli per 12 bit

## Notazione esadecimale

- Es.: 101101010011 diventa  
B53

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

# Manipolazione logica di bit

## ■ Algebra di Boole (utile per la specifica di funzioni logiche):

- variabili logiche (binarie) e operazioni logiche
- una variabile A può prendere valore 0 (FALSO) o 1 (VERO)
- operazioni logiche di base: AND, OR, NOT

A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

$$R = A \text{ AND } B$$

A	B	R
0	0	0
0	1	1
1	0	1
1	1	1

$$R = A \text{ OR } B$$

A	R
0	1
1	0

$$R = \text{NOT } A$$

## ■ Operatori booleani su due variabili

$$A \text{ AND } B = A \cdot B$$

$$A \text{ OR } B = A + B$$

$$\text{NOT } A = \bar{A}$$

**Esempio:**  $D = A + (\bar{B} \cdot C)$

D è uguale a 1 se A è 1 o se B = 0 e C = 1.  
Altrimenti D è uguale a 0.

# Manipolazione logica di bit

## ■ Operatori booleani su due variabili

P	Q	NOT P ( $\bar{P}$ )	P AND Q ( $P \cdot Q$ )	P OR Q ( $P + Q$ )	P NAND Q ( $\overline{P \cdot Q}$ )	P NOR Q ( $\overline{P + Q}$ )	P XOR Q ( $P \oplus Q$ )
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

## ■ Algebra booleana: postulati e identità

Basic Postulates		
$A \cdot B = B \cdot A$	$A + B = B + A$	Commutative Laws
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Distributive Laws
$1 \cdot A = A$	$0 + A = A$	Identity Elements
$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	Inverse Elements
Other Identities		
$0 \cdot A = 0$	$1 + A = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	Associative Laws
$\bar{A} \cdot B = \bar{A} + \bar{B}$	$A + B = \bar{A} \cdot \bar{B}$	DeMorgan's Theorem