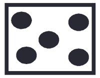
RAPPRESENTAZIONE DI DATI: SISTEMI DI NUMERAZIONE

Rappresentazione binaria

- Tutta l'informazione interna ad un computer è codificata con sequenze di due soli simboli : 0 e 1
  - è facile realizzare dispositivi elettronici che discriminano fra due stati, molto meno se gli stati sono tanti
- L'unità elementare di informazione si chiama *bit* da *'binary digit'*
- *Byte* : sequenza di 8 bit

Sistema decimale posizionale (1)

- Un numero (es. 5) può essere rappresentato in molti modi :
  - cinque, *five, 5,* V, XXXXX, ,…..
- Rappresentazioni diverse hanno proprietà diverse
  - moltiplicare due numeri in notazione romana è molto più difficile che moltiplicarli in notazione decimale ….
- Noi siamo abituati a lavorare con numeri rappresentati in notazione posizionale in base 10 (decimale)
- La rappresentazione di un numero intero in base 10 è una sequenza di cifre scelte fra 0 1 2 3 4 5 6 7 8 9
- Il valore di una rappresentazione è dato da

  $a_N \cdot 10^N + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} + \dots$
  - b = 10 è la base
  - a → cifra
  - 10^i dove i è il peso della cifra a_i nel valore del numero

Sistema decimale posizionale (2)

- $253 = 2 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 = 2 \times 100 + 5 \times 10 + 3 \times 1$
- $23{,}47 = 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2} = 2 \times 10 + 3 \times 1 + 4 \times 0.1 + 7 \times 0.01 = 2 \times 10 + 3 \times 1 + 4 \times (1/10) + 7 \times (1/100)$
- Sistema *posizionale*
- Sistema *decimale*

3	2	1	0	← Posizione della cifra
7	3	0	9	← NUMERO

$9 * 10^{0} = 9$

$0 * 10^{1} = 0$

$3 * 10^{2} = 300$

$7 * 10^{3} = 7000$

$9 + 0 + 300 + 7000$

Notazione posizionale in base 2

- La rappresentazione di un numero intero in base 2 è una sequenza di cifre scelte fra 0 1 : es: 10, 110, 1101101
- Il valore di una rappresentazione è dato da

$$a_N \cdot 2^N + a_{N-1} \cdot 2^{N-1} \ldots .+ a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + a_{-2} \cdot 2^{-2} + \ldots$$

  - b= 2 è la base
  - $\mathbf{2^i}$ i è il peso della cifra $\mathbf{a_i}$ nel valore del numero

Esempi :

- 10 = $1*2^1 + 0*2^0 = 2$
- 110 = $1*2^2 + 1*2^1 + 0*2^0 = 4 + 2 + 0 = 6$

- *10 si legge uno zero e non dieci !!!*

Conversione di interi: Base 10 $\rightarrow$ Base 2

- Successione di divisioni per 2: termina quando il resto è 0
- Resti determinati in ordine inverso

es.: 13	6	3	1	0	Quozienti
	1	0	1	1	Resti

$$13_{10} = \mathbf{1101_2}$$

Conversione di interi: Base 2 $\rightarrow$ Base 10

- Somma pesata delle cifre binarie:

es.: $1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1$

$= \mathbf{13}_{10}$

Numeri binari interi: esempi

0	0	8	1000
1	1	9	1001
2	10	10	1010
3	11	11	1011
4	100	12	1100
5	101	13	1101
6	110	14	1110
7	111	15	1111

$2^0 = 1$		$2^8 = 256$
$2^1 = 2$		$2^9 = 512$
$2^2 = 4$		$2^{10} = 1024$
$2^3 = 8$		$2^{11} = 2048$
$2^4 = 16$		$2^{12} = 4096$
$2^5 = 32$		...
$2^6 = 64$		$2^{16} = 65536$
$2^7 = 128$		...
		$2^{24} \cong$ 16 milioni
		...

Aritmetica binaria (addizione-sottrazione)

- Necessità di codificare nel mondo dei numeri binari ogni operazione aritmetica.

- addizione: es.:

0+0=0
0+1=1
1+0=1
1+1=0 col riporto di 1

$$\begin{array}{r} 0101 + \\ 0011 = \\ \hline 1000 \end{array} \qquad \begin{array}{r} 5_{10} + \\ 3_{10} = \\ \hline 8_{10} \end{array}$$

- Sottrazione
  - Regola del prestito: chi presta diventa 0 chi riceve diventa 10 tutti gli 0 tra chi presta e chi riceve diventano 1

0-0=0
1-1=0
1-0=1
0-1=1 col prestito di 1 dalla cifra precedente

11011 -
10101=
00110

Aritmetica binaria (moltiplicazione-divisione)

- Moltiplicazione: es.: per 2, 2^2, 2^3, ... ↔ 'shift' (traslazione) verso sx di 1, 2, 3 bit

1101 x 100 = 110100

(13 x 4 = 52)

- 1 per 1 = 1
- 0 per 1 = 0

```
   1101 *
    101=
--------
   1101-
  0000-
 1101-
--------
1000001
```

divisione

- 1101 | 11
- 11   100
- 0001

RISULTATO 100
RESTO 1

Sistema esadecimale

- Base 16
- Simboli : 0…9 + le sei lettere maiuscole A,B,C,D,E,F
- Assume importanza perché la base 16 è una potenza di 2, base del sistema di numerazione binario. $16 = 2^4$

Da esadecimale a decimale

- Per la conversione da hex → dec si usa lo stesso metodo usato precedentemente per bin → dec
- numero esadecimale 2BC1

$$2*(16)^3 + 11 *(16)^2 + 12*(16)^1 + 1*(16)^0$$

$$2*4096 + 11*256 + 12*16 + 1*1 = 11201$$

Decimale in esadecimale

- Viene usato il metodo delle divisioni intere successive e vengono presi i resti.
- 11201 : 16 = resto 1
- 700 : 16 = resto 12 C
- 43 : 16 = resto 11 B
- 2