Rappresentazione dell’Informazione

Rappresentazione delle informazioni in codice binario

- Caratteri
- Naturali e Reali positivi
- Interi
- Razionali

Rappresentazione del testo

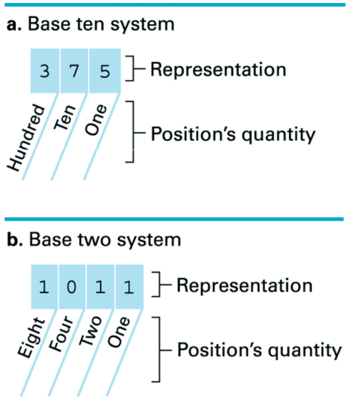
- Una **stringa di bit** per ogni simbolo (caratteri maiuscoli, caratteri minuscoli, cifre, ...)
- ANSI (American National Standards Institute) ha adottato il **codice ASCII** (American Standard Code for Information Interchange): **7 bit** per ogni simbolo + **0** come **bit piu’ significativo** =un byte

01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

Rappresentare numeri

- Il codice **ASCII e’ inefficiente**: per rappresentare **numeri** con **n** cifre servono n byte
- Meglio usare metodi che sfruttano la **notazione binaria** (base 2)
- Base 2: solo le cifre 0 e 1 invece che 0, 1, ..., 9 (base 10)

Base 10 e base 2



Rappresentazione decimale

- **Base 10** ➔ cifre da 0 a 9
- Sequenza di cifre decimali

$$\mathbf{d_{k-1} \ldots d_1\, d_0}$$

➔ numero intero

$$\Sigma_{j=0\ldots k-1}\, d_j\, 10^j$$

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

Esempio: **102** in base 10 è
1x100 + 0x10 + 2x1

Rappresentazione binaria

- **Base 2** ➔ cifre 0 e 1
- Sequenza di cifre binarie

$$\mathbf{d_{k-1} \ldots d_1\, d_0}$$

➔ numero intero (stesso procedimento ma su base 2)

$$\Sigma_{j=0\ldots k-1}\, d_j\, 2^j$$

Esempio:

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

Rappresentazione binaria

- **Valore minimo** di una sequenza di n cifre binarie: 000 … 0 (n volte) = 0_{10}
- **Valore massimo**: 1111…111 (n volte) = $2^{n-1} + 2^{n-2} + \ldots + 2^2 + 2^1 + 2^0 = 2^n - 1$

- Esempio con $n=3$: $111 = 2^2 + 2 + 1 = 7 = 2^3 - 1$
- Da 0 a 8: 0, 1, 10, 11, 100, 101, 110, 111, 1000

Una proprietà dei numeri binari

$\boxed{100100}1$ = 73

100100 = 36 = 73/2

e questo è il resto

Eliminare il bit più a destra corrisponde a dividere per 2 il valore, ed il bit eliminato è il resto

Trasformazione di un numero in base 10 a numero binario

125

125/2=62	resto **1**
62/2=31	resto **0**
31/2=15	resto **1**
15/2=7	resto **1**
7/2=3	resto **1**
3/2=1	resto **1**
1/2=0	resto **1**

125 **in binario è**

1111101

rappresenta 62

rappresenta 31

Etc.

Esercizio 1

- Scrivere la rappresentazione binaria dei numeri decimali:
  - 30
  - 36
  - 15

Esercizio 2

- Scrivere la rappresentazione decimale dei numeri binari:
  - 1000
  - 1010
  - 01011
  - 10111

Correzione degli esercizi

- Scrivere la rappresentazione binaria dei numeri decimali:
  - 30 → 11110
  - 36 → 100100
  - 15 → 1111

Correzione degli esercizi

- Scrivere la rappresentazione decimale dei numeri binari:
  - 1000 → 8
  - 1010 → 10
  - 01011 → 11
  - 10111 → 23

Somma binaria

Somma binaria

- **Colonna per colonna**, da destra a sinistra
- **Riporto** se la somma su una colonna supera la base

$$\begin{array}{r}0\\+0\\\hline 0\end{array}\qquad\begin{array}{r}1\\+0\\\hline 1\end{array}\qquad\begin{array}{r}0\\+1\\\hline 1\end{array}\qquad\begin{array}{r}1\\+1\\\hline 10\end{array}$$

- Tre cifre binarie (prima riga, seconda riga, riporto), **somma =1** se **una o tre sono 1**, **riporto = 1** se **almeno due sono 1**

Riporto: 1 1 1 1 0 0

011100_2 +

100111_2 =

1000011_2

```
 1    11      riporti
  1010011 +
  1100011 =
----------
10110110
```

Si vuole quindi **costruire un circuito** per **sommare due numeri binari**

```
10000110    riporti
 1010011 +
 1100011 =
----------
10110110
```

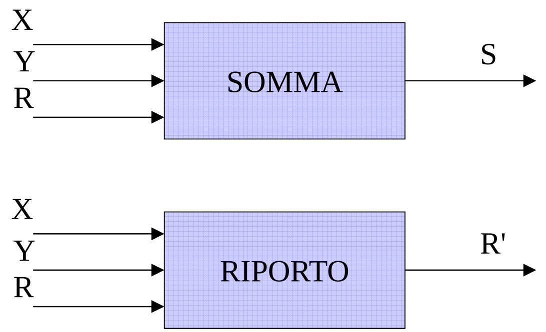
Iniziamo con un **circuito** che faccia la **somma su una colonna**

Abbiamo **tre** cifre binarie **X, Y, R** in **input** mentre in **output** vogliamo ottenere la **somma S** ed il **riporto R'**

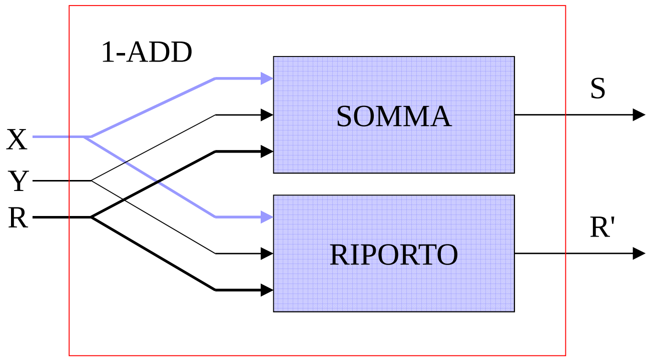
Tabella di verità

X	Y	R	S	R'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

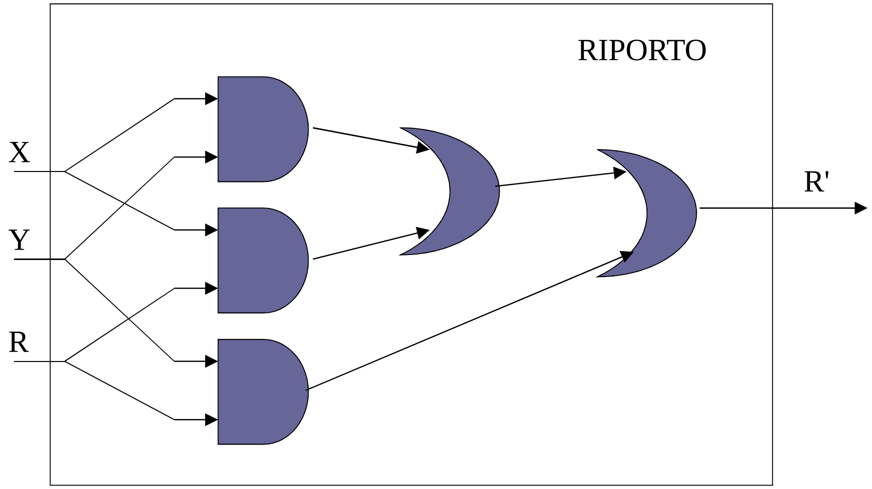
Supponiamo di avere i circuiti che calcolano **somma** e **riporto**



Possiamo allora **combinare** i circuiti **SOMMA** e **RIPORTO** per ottenere il seguente circuito **1-ADD**



Il circuito **RIPORTO** puo` essere realizzato nel seguente modo



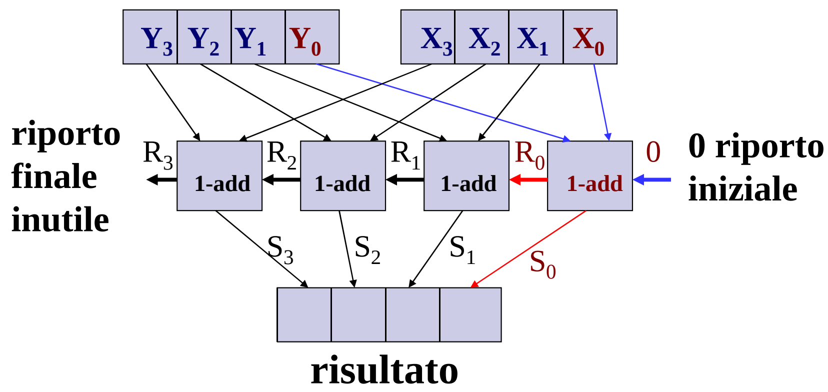
Basta infatti verificare la corrispondente tabella di verita’

Il circuito **SOMMA** naturalmente puo' pure essere realizzato (vedi dispensa).

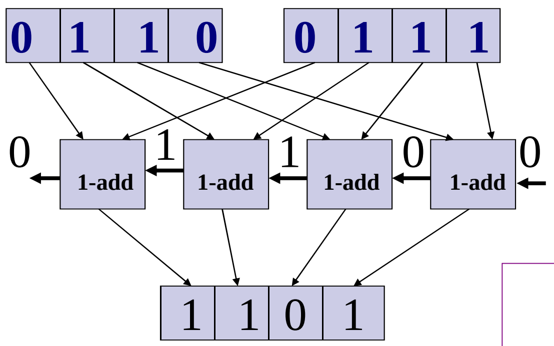
A questo punto **componendo K circuiti 1-ADD** e` possibile realizzare **un circuito K-ADD** che **somma due numeri binari di K cifre**.

Vediamo l'esempio della somma di due numeri binari di 4 cifre.

Somma di numeri di 4 bit



Esempio




```
 0111 +
 0110 =
-------
 1101
```

Attenzione

Si e` **trascurato** il problema del cosiddetto **overflow**, cioe' il risultato e' troppo grande per essere contenuto nei bit disponibili.

Per esempio:

```
 0111 +
 1110 =
-------
10101
```

Esercizi

```
11011+
 1100
---------
```

```
11111+
     1
---------
```

Correzioni

```
 1
 11011+
  1100
 ---------
100111
```

```
11111
 11111+
      1
 ---------
100000
```

Rappresentazione dei reali

Reali in notazione binaria

- $b_{k-1}\ b_{k-2}\ \ldots\ b_2\ b_1\ b_0\ ,\ b_{-1}\ b_{-2}\ \ldots$
- $b_{k-1} \times 2^{k-1} + b_{k-2} \times 2^{k-2} + \ldots + b_2 \times 2^2 + b_1 \times 2 + b_0 \times 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \ldots$

- Da decimale a binario:
  - Per la **parte intera**, **come sappiamo fare** (**metodo delle divisioni**)

REALE--> BINARIO

cosa significa una **parte frazionaria binaria**:

.1101001

$2^{-1} + 2^{-2} + 2^{-4} + 2^{-7}$

.1101001

$2^{-1}\ 2^{-2} \ldots$

moltiplicarlo per 2 significa spostare il punto di un posto a destra

1.101001

$2^0\ 2^{-1} \ldots\ldots$

Se abbiamo un valore decimale in base 10:

0.99 come troviamo la sua **rappresentazione in base 2**? Ragioniamo come segue:

Supponiamo che $.99 = .b_1b_2b_3...b_k$ (binario)

Allora $2\times .99 = \mathbf{1}.98 = \mathbf{b_1}.b_2b_3...b_k$

Quindi $\mathbf{b_1}$ è 1

e $.98$ è rappresentato da $.b_2b_3...b_k$

Per trovare la **rappresentazione binaria di un decimale** lo moltiplichiamo per 2 ed osserviamo se **1** appare **nella parte intera:**

rappresentazione binaria di .59

.59×2= 1.18	**.72×2= 1.44**
.18×2= 0.36	**.44×2= 0.88**
.36×2= 0.72	**.88×2= 1.76**

.100101.....

.... dipende da quanti bit abbiamo

Esempio

18.59

18 → 10010 (metodo della divisione per 2)

.59 → .100101...(metodo della moltiplic. per 2)

10010.100101....

Esercizi

- Convertire i seguenti numeri binari in formato decimale:
  - 11,01
  - 101,111
  - 10,1
- Esprimere i seguenti valori in notazione binaria:
  - 4.5
  - 2.75
- Eseguire le seguenti somme binarie:
  - 1010,001+1,101
  - 111,11+0,01

Correzione degli esercizi

- Convertire i seguenti numeri binari in formato decimale:
  - **11,01** → 3 +1/4 = 13/4 = 3.25
  - **101,111** → 5 + 7/8 = 47/8 = 5.87
  - **10,1** → 2.5
- Esprimere i seguenti valori in notazione binaria:
  - **4.5** →100,1
  - **2.75** → 10,11
- Eseguire le seguenti somme binarie:
  - **1010,001 + 1,101** →1011,110
  - **111,11 + 0,01** →1000,00