

Corso di Architettura degli Elaboratori e Laboratorio (M-Z)

Sistemi di numerazione e rappresentazione binaria dei numeri interi

Nino Cauli



UNIVERSITÀ
degli STUDI
di CATANIA

Dipartimento di Matematica e Informatica

- 10 **SIMBOLI**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Ogni simbolo rappresenta una quantità ben precisa
- Ma la quantità dipende anche dalla **POSIZIONE** del simbolo

278

8 unità

7 decine (gruppi da 10)

2 centinaia (gruppi da 10×10)

- La **QUANTITÀ** dipende anche dalla **POSIZIONE** della cifra
- Il sistema decimale è appunto **POSIZIONALE**
- La **NUMERAZIONE ROMANA** non è posizionale, ma **ADDITIVA** perché il valore complessivo del numero è dato dalla somma dei valori dei simboli, indipendentemente dalla loro posizione

I = 1, **IV** = 4, **V** = 5, **IX** = 9, **X** = 10, **XL** = 40, **L** = 50, **XC** = 90, **C** = 100

CCLXXVIII =

$$(2*\mathbf{C} + \mathbf{L} + 2*\mathbf{X} + \mathbf{V} + 3*\mathbf{I}) = (2*100 + 50 + 2*20 + 5 + 3*1) \\ = \mathbf{278}$$

- Il sistema **DECIMALE/ARABO** è estremamente economico e flessibile
- Con un numero **LIMITATO DI SIMBOLI** (solo 10) è possibile rappresentare **QUALUNQUE QUANTITÀ**
- Questo non è vero per i sistemi **ADDITIVI** i quali, al crescere della quantità, hanno sempre bisogno di **NUOVI SIMBOLI**

Perché 10 simboli?

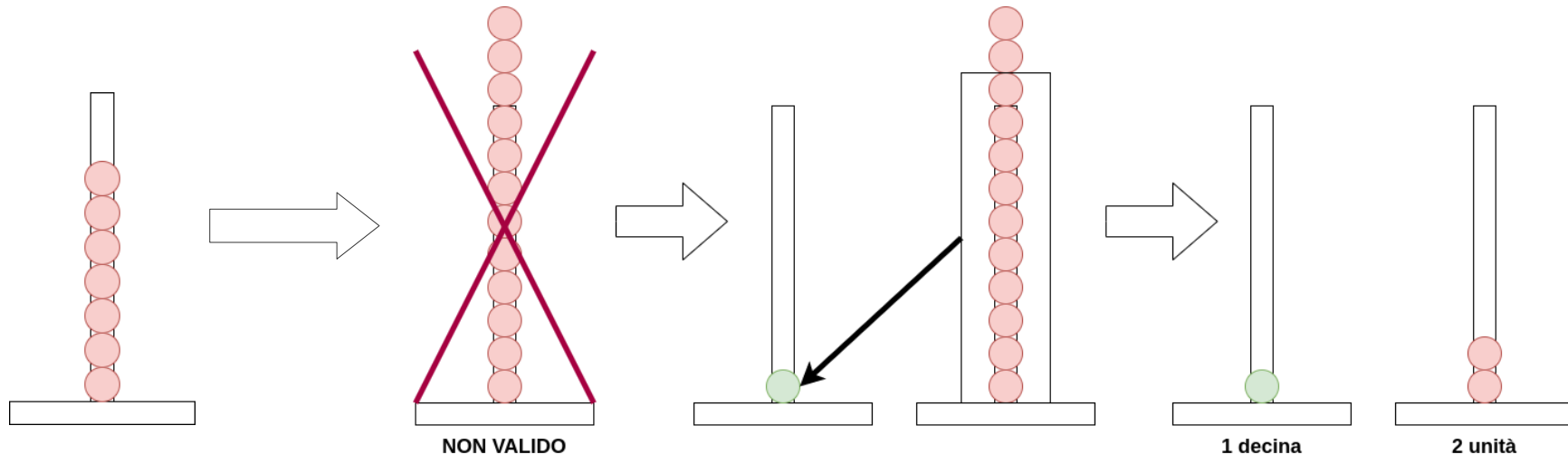
Potremmo usarne di più o di meno?

È possibile averne un numero arbitrario (7, 3, 2, 15, 16, etc.)?

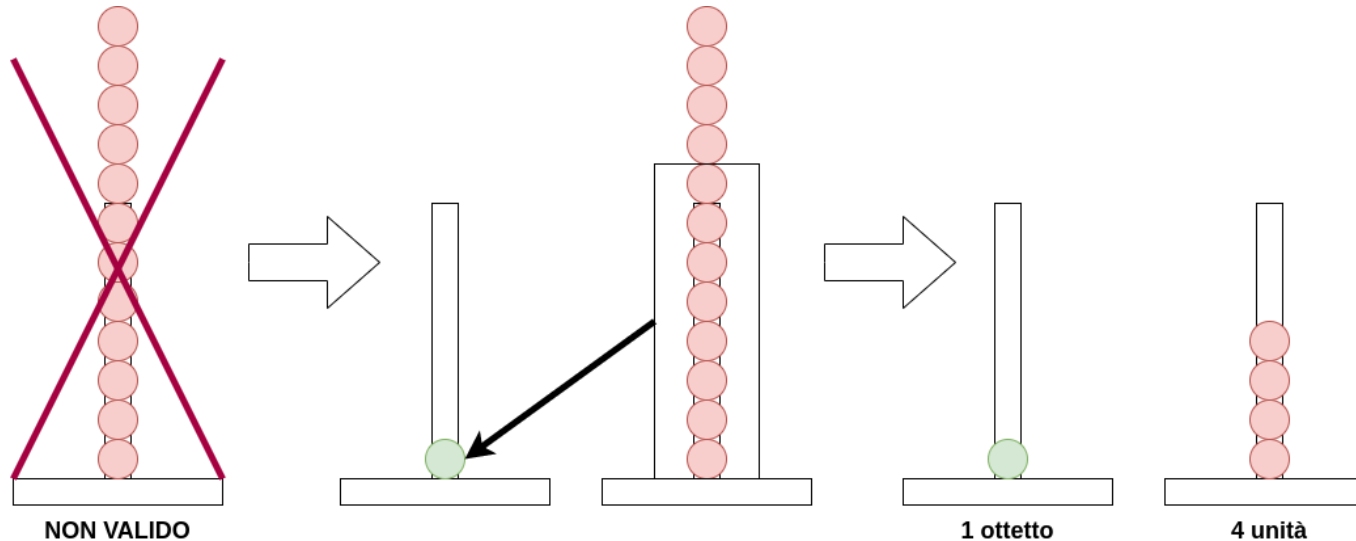
- Da piccoli ci hanno insegnato a rappresentare i numeri con gli abachi
- Un abaco è una serie di asticelle dove è possibile impilare una quantità di palline corrispondenti al numero che vogliamo rappresentare
- **Ma, attenzione!** Ogni asticella di un abaco non può contenere più di 9 palline



- Se, ad una asticella con **7 PALLINE**, aggiungiamo altre **5 PALLINE** andiamo oltre la capacità dell'asticella
- Usiamo l'asticella successiva che **“RACCOGLIE”** gruppi di 10 palline
- Non appena la prima asticella supera la sua capacità, raggruppiamo le palline e trasformiamole in una pallina singola posta nell'asticella **SUCCESSIVA**
- Chiamiamo questa configurazione **ABACO-10 = SISTEMA DECIMALE**

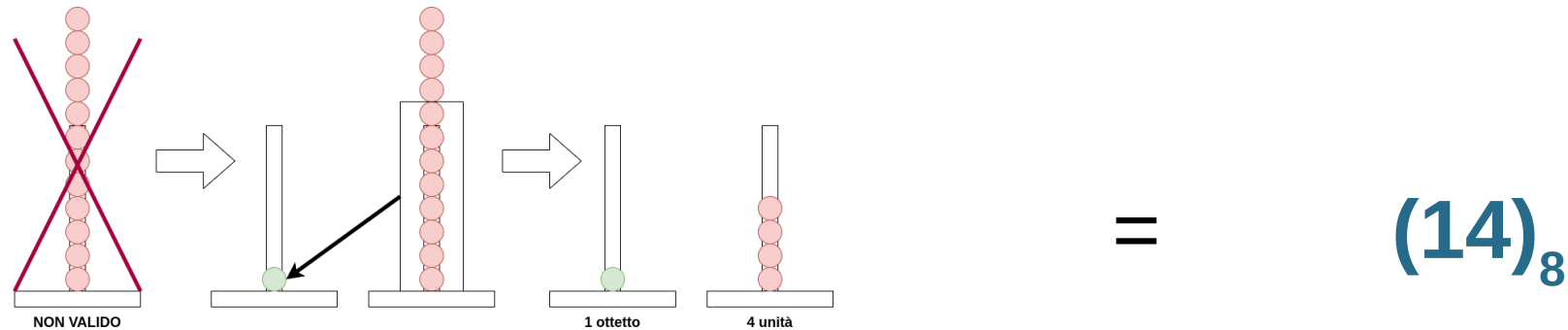
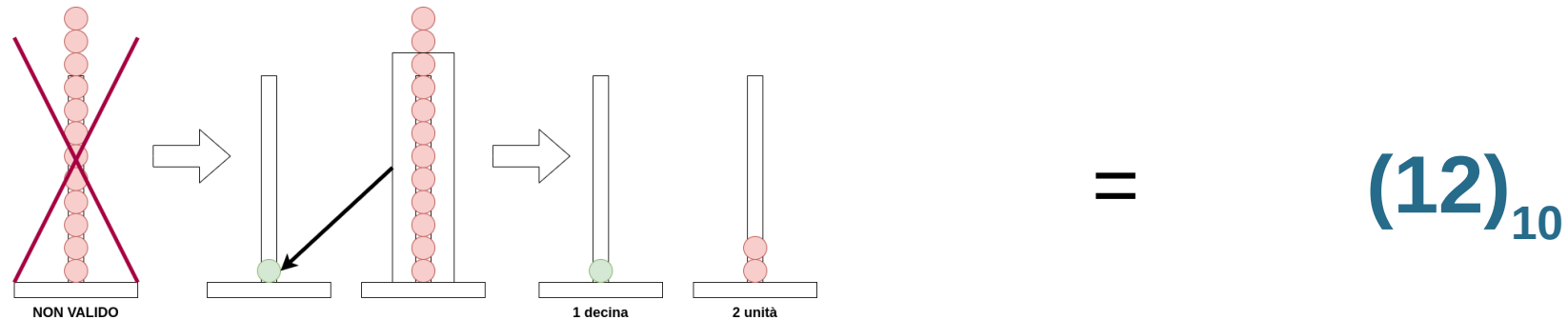


- Supponiamo di avere lo stesso numero di palline, ma le asticelle non possono contenerne più di **7**
- L'asticella successiva raccoglierà dunque gruppi di 8 palline (**OTTETTI**)
- Nell'asticella delle **UNITÀ** rimarranno **4** palline
- Chiamiamo questo sistema **ABACO-8 = SISTEMA OTTALE**



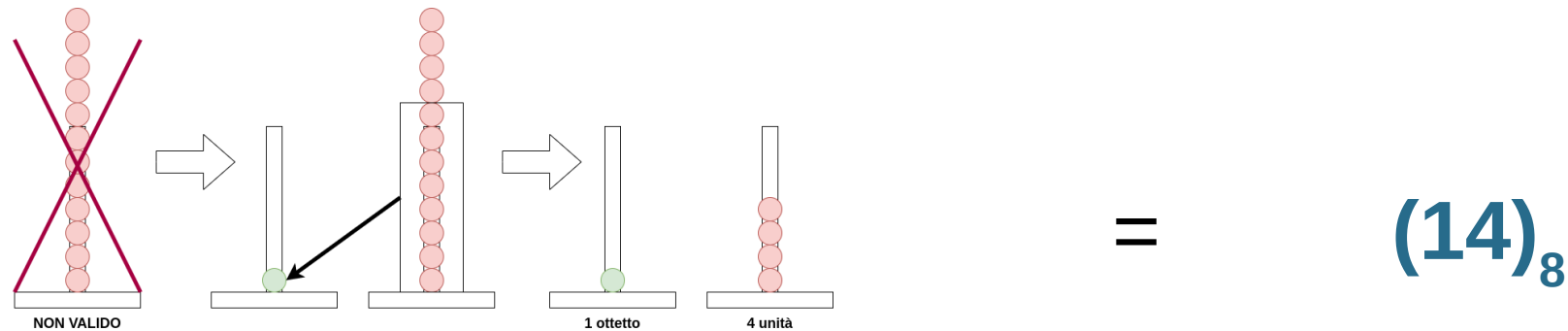
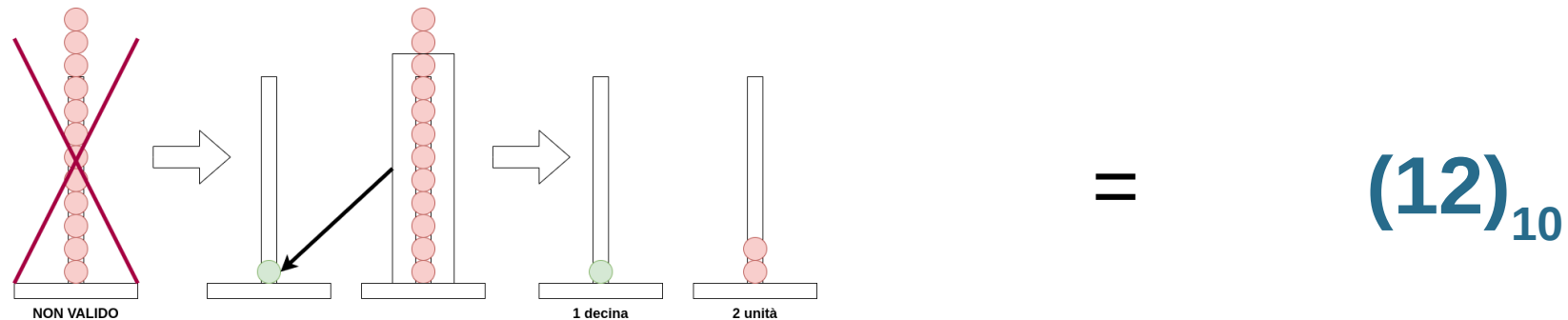
Abaco-8 vs Abaco-10

- La dicitura **14** nell'**ABACO-8** corrisponderà alla dicitura **12** nell'**ABACO-10**
- Entrambe le diciture rappresentano la stessa quantità, ma utilizzano un sistema di “**scrittura**” che ha regole differenti



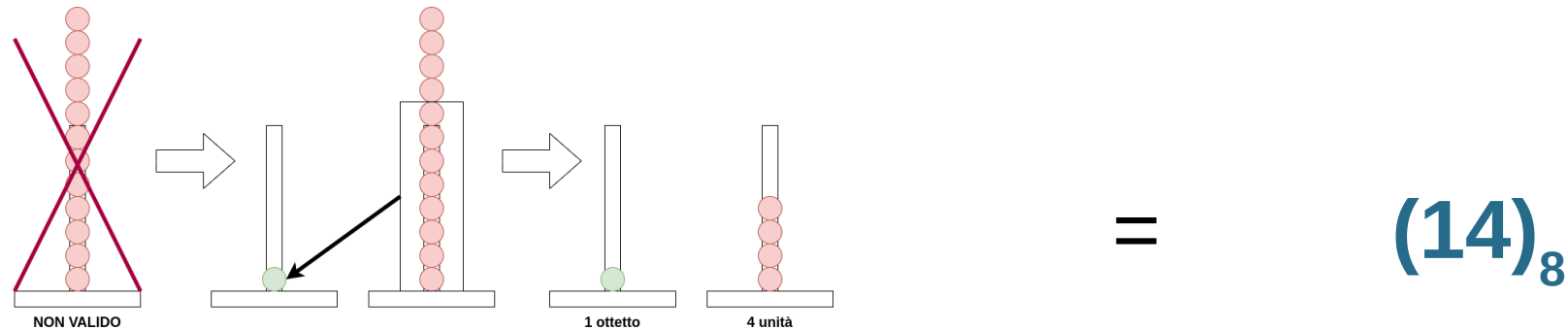
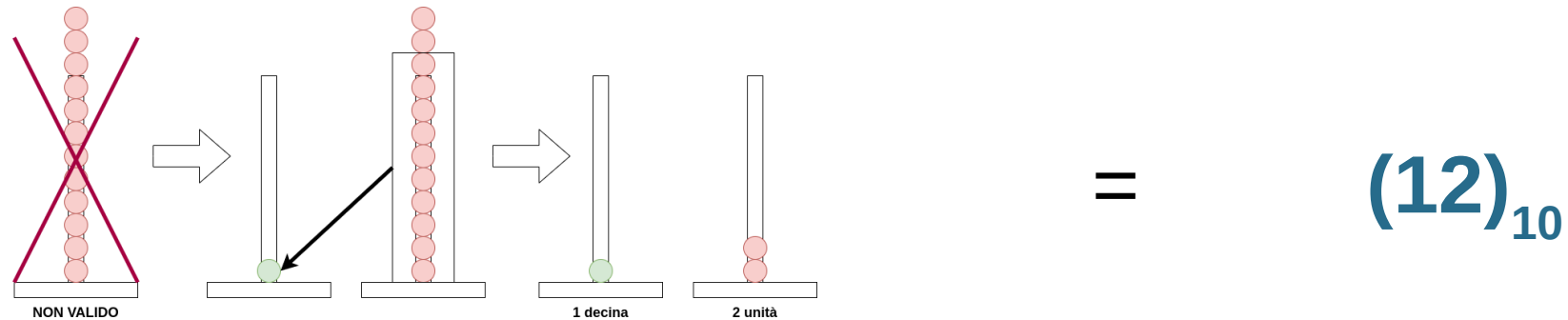
Abaco-8 vs Abaco-10

- L'**ABACO-10** ha bisogno di **10 SIMBOLI** -> $\{0, \dots, 9\}$
- L'**ABACO-8** ha bisogno solo di **8 SIMBOLI** -> $\{0, 1, 2, 4, 5, 6, 7\}$

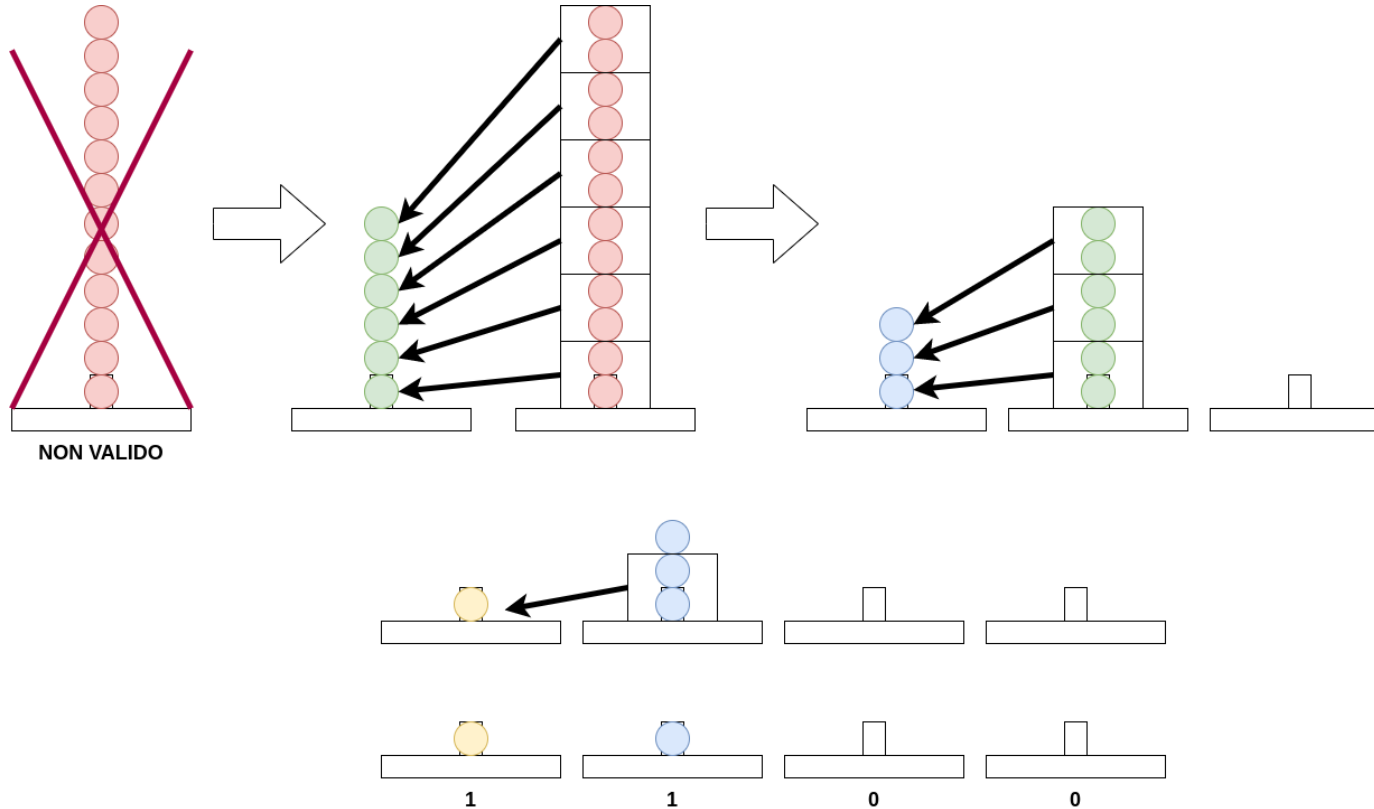


Abaco-8 vs Abaco-10

- Nell'**ABACO-10** usiamo un **SISTEMA A BASE 10**
- Nell'**ABACO-8** usiamo un **SISTEMA A BASE 8**



- Consideriamo solo **2 SIMBOLI** = {0, 1}
- Ogni asticella potrà contenere **AL PIÙ UNA PALLINA**



ABACO-2 (Sistema Binario)

- {0, 1}

$$(12)_{10} = (1100)_2$$

ABACO-8 (Sistema Ottale)

- {0, 1, 2, 3, 4, 5, 6, 7}

$$(12)_{10} = (14)_8$$

ABACO-10 (Sistema Decimale)

- {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

$$(12)_{10} = (12)_{10}$$

ABACO-16 (Sistema Esadecimale)

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

$$(12)_{10} = (C)_{16}$$

Un sistema di numerazione è definito da:

- Un intero B detto **BASE**
- Un insieme di B simboli $S_B = \{s_0, \dots, s_{B-1}\}$, ognuno dei quali rappresenta le quantità **0,1,2,...,B-1**

Un numero a n cifre $p_{(n-1)}p_{(n-2)}\dots p_1p_0$ con $p_{(i)} \in S_B$ e $i=0,\dots,n-1$ può essere rappresentato come **SOMMA DI POTENZE DELLA BASE**:

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

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

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

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**

| Quoziente | Resto |
|-----------|-------|
| 13 | |
| 6 | 1 |
| 3 | 0 |
| 1 | 1 |
| 0 | 1 |

Convertire 13 in base 2

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

| Quoziente | Resto |
|-----------|-------|
| 13 | |
| 2 | 3 |
| 0 | 2 |

Convertire 13 in base 5

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

$$\mathbf{P} = p_{(n-1)}p_{(n-2)}\dots p_1p_0, \quad p_{(i)} \in S_B \text{ e } i=0,\dots,n-1$$

Quanti valori possono essere rappresentati dal numero \mathbf{P} espresso in base B ?

$$n = 1$$

$$|\mathbf{P}| = |p_0| = B$$

$$[0, B)$$

$$n = 2$$

$$|\mathbf{P}| = |p_1p_0| = B*B = B^2$$

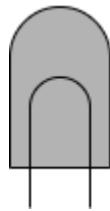
$$[0, B^2)$$

$$n$$

$$|\mathbf{P}| = |p_{(n-1)}p_{(n-2)}\dots p_1p_0| = B^n$$

$$[0, B^n)$$

- Il calcolatore è una macchina composta da **CIRCUITI E COLLEGAMENTI ELETTRICI**
- Immaginiamo di poter connettere delle lampadine o dei **LED** ai vari collegamenti presenti dentro un computer
- Immaginiamo di poter effettuare delle “**ISTANTANEE**” per valutare la luminosità delle lampadine
- Scopriremmo che ogni lampadina è sempre o **TOTALMENTE SPENTA** oppure **TOTALMENTE ACCESA**
- Non troveremo mai una lampadina accesa “con luminosità parziale”



- 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**

$$\mathbf{P} = p_{(n-1)}p_{(n-2)}\cdots p_1p_0, \quad p_{(i)} \in \{0,1\} \text{ e } i=0,\dots,n-1$$

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

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

Come rappresentare i **NUMERI INTERI RELATIVI**?



IDEA

Usare il bit più a sinistra per rappresentare il segno:

- **0 = POSITIVO**
- **1 = NEGATIVO**

Come rappresentare il **VALORE ASSOLUTO**?

3 SOLUZIONI

- **SEGNO E VALORE ASSOLUTO**
- **COMPLEMENTO A UNO**
- **COMPLEMENTO A DUE**

Numeri con segno

| Stringa binaria | | | Segno Valore assoluto | Comp. 1 | Comp. 2 |
|-----------------|---|---|------------------------------|------------------------------|---------|
| 0 | 1 | 1 | +3 | +3 | +3 |
| 0 | 1 | 0 | +2 | +2 | +2 |
| 0 | 0 | 1 | +1 | +1 | +1 |
| 0 | 0 | 0 | +0 | +0 | +0 |
| 1 | 0 | 0 | -0 | -3 | -4 |
| 1 | 0 | 1 | -1 | -2 | -3 |
| 1 | 1 | 0 | -2 | -1 | -2 |
| 1 | 1 | 1 | -3 | -0 | -1 |
| | | | ↓ | ↓ | ↓ |
| | | | $(-2^{n-1}, \dots, 2^{n-1})$ | $[-2^{n-1}, \dots, 2^{n-1})$ | |

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

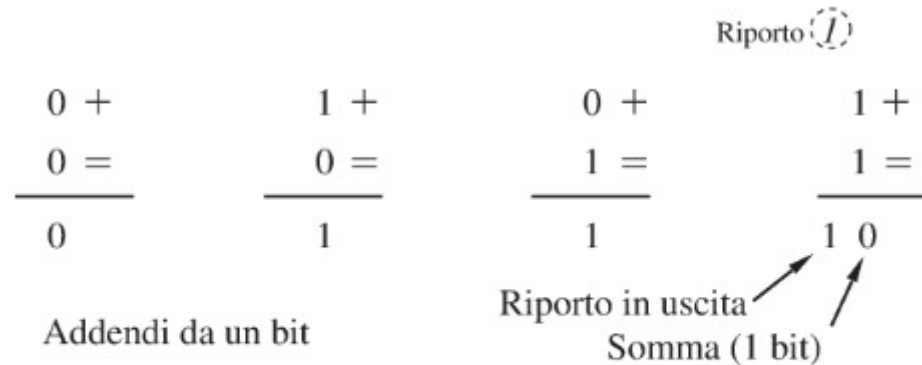


Figura 1.4 - Addizione di numeri a un bit

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

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 **$n-1$**

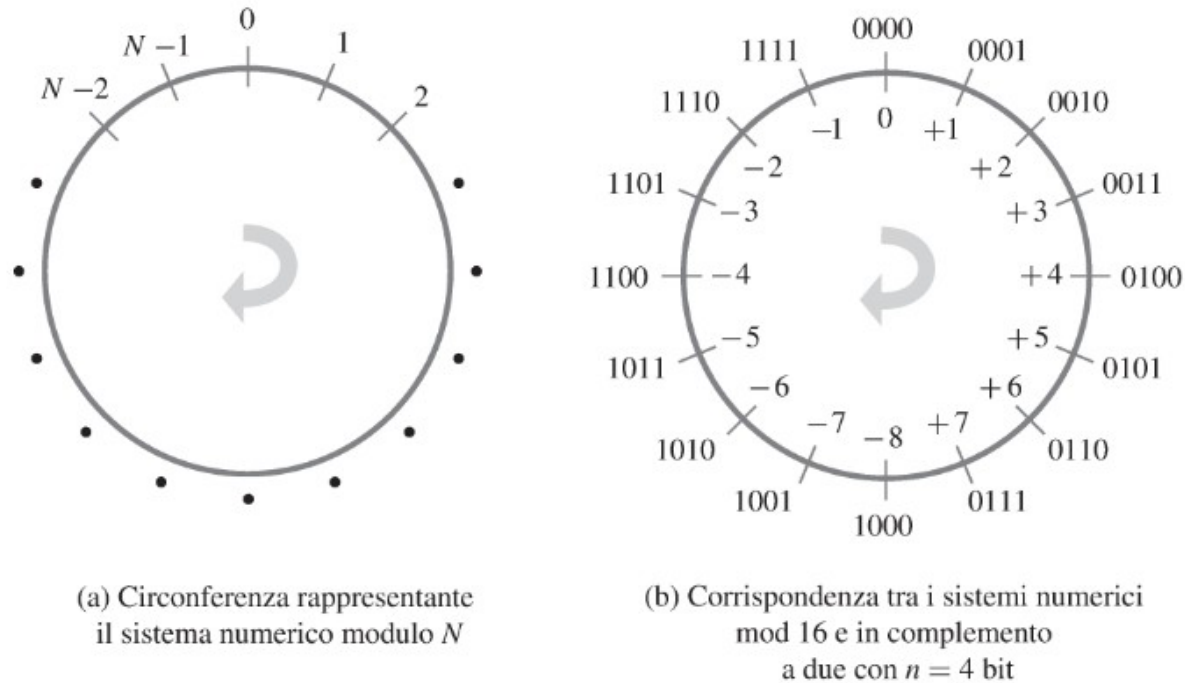


Figura 1.5 - Rappresentazione geometrica di un sistema numerico in complemento a due

Come quella binaria naturale, ma **trascurando il riporto in uscita**

ADDIZIONE: come quella binaria naturale, ma trascurando il riporto in uscita

SOTTRAZIONE: addizione con il complemento a due del sottraendo

| | | | | |
|-----|---|---------------|-----|---|
| (a) | $\begin{array}{r} 0010 + (+2) \\ 0011 = (+3) \\ \hline 0101 \quad (+5) \end{array}$ | | (b) | $\begin{array}{r} 0100 + (+4) \\ 1010 = (-6) \\ \hline 1110 \quad (-2) \end{array}$ |
| (c) | $\begin{array}{r} 1011 + (-5) \\ 1110 = (-2) \\ \hline 1001 \quad (-7) \end{array}$ | | (d) | $\begin{array}{r} 0111 + (+7) \\ 1101 = (-3) \\ \hline 0100 \quad (+4) \end{array}$ |
| (e) | $\begin{array}{r} 1101 - (-3) \\ 1001 = (-7) \\ \hline \end{array}$ | \Rightarrow | | $\begin{array}{r} 1101 + \\ 0111 = \\ \hline 0100 \quad (+4) \end{array}$ |
| (f) | $\begin{array}{r} 0010 - (+2) \\ 0100 = (+4) \\ \hline \end{array}$ | \Rightarrow | | $\begin{array}{r} 0010 + \\ 1100 = \\ \hline 1110 \quad (-2) \end{array}$ |

Figura 1.6 (parte) - Operazioni di addizione e sottrazione in complemento a due

Il risultato di addizione e sottrazione in complemento a 2 è corretto se è **COMPRESO** nell'intervallo:

$$[-2^{n-1}, 2^{n-1})$$

In caso contrario avviene un evento di **TRABOCCO**

Il **TRABOCCO** può avvenire solo se:

- 1) I due addendi sono **CONCORDI IN SEGNO**
- 2) Il **BIT DI SEGNO** della somma degli addendi è **DIVERSO** da quello degli addendi

Spesso si presenta la necessità di aumentare o diminuire il numero di bit usati per codificare un numero

Regole molto semplici:

- **ESTENSIONE DEL SEGNO**: si replica a sinistra il bit del segno tante volte quanto occorre
- **RIDUZIONE DEL SEGNO**: si rimuove il bit più a sinistra tante volte quante occorre, purché il bit successivo abbia ugual valore