



**UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE**

# Rappresentazione dell'informazione (2)

**Prof.ssa Giulia Cisotto**

[giulia.cisotto@units.it](mailto:giulia.cisotto@units.it)

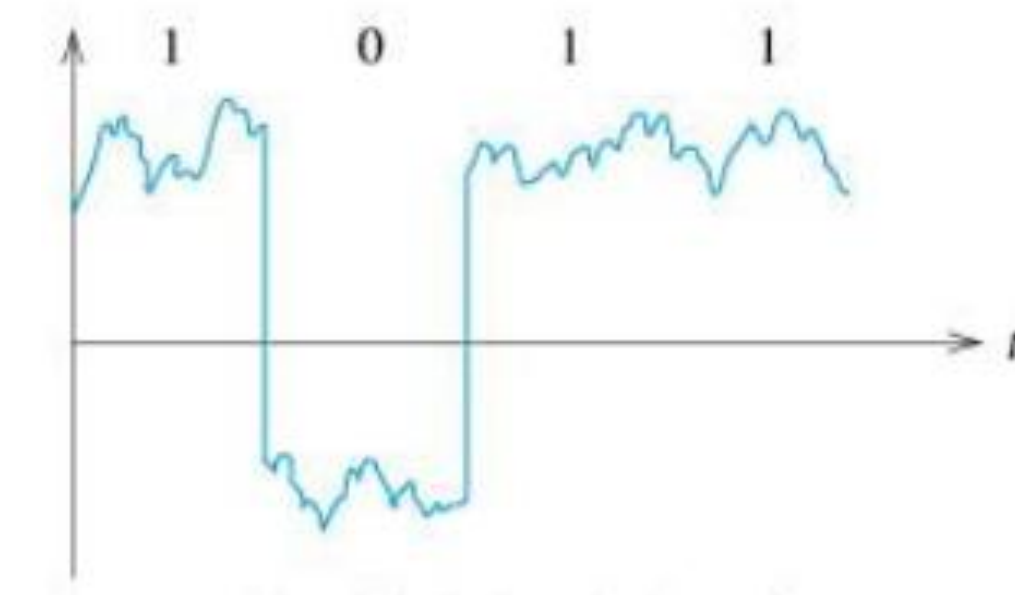
Trieste, 6 marzo 2025

# Come posso rappresentare l'informazione nel pc?

**Transistor e altra «logica» (circuiteria)** permette di trasformare il segnale continuo che oscilla tra livello basso e alto in un'onda quadra:

Se la tensione è **alta** (es. **5V** o **12V**) → viene interpretata come **1**.

Se la tensione è **bassa** (es. **0V**) → viene interpretata come **0**.



SISTEMA NUMERICO DECIMALE	SISTEMA NUMERICO BINARIO	SISTEMA NUMERICO ESADECIMALE
0	0	0
1	1	1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
		A
		B
		C
		D
		E
		F

Dati **k** bit (che formano una **sequenza**), il numero di **configurazioni** ottenibili è **2<sup>k</sup>**

*Per numeri positivi!*

$$N = [d_{n-1}, d_{n-2}, \dots, d_1, d_0, d_{-1}, \dots, d_{-m}]$$

$$N = d_{n-1} \cdot r^{n-1} + d_{n-2} \cdot r^{n-2} + \dots + d_1 \cdot r^1 + d_0 \cdot r^0 + d_{-1} \cdot r^{-1} + \dots + d_{-m} \cdot r^{-m} = \sum_{i=-m}^{n-1} d_i \cdot r^i$$

Conversioni da base 2 o 16 a base 10.

# AGENDA DI OGGI

1. Rappresentazione in base 2, 16, 10
2. Conversioni tra sistemi numerici
3. Rappresentabilità e overflow
4. Rappresentazione di numeri negativi
5. Rappresentazione di numeri reali in vigola fissa



# CONVERSIONE DA DECIMALE A QUALSIASI ALTRA BASE

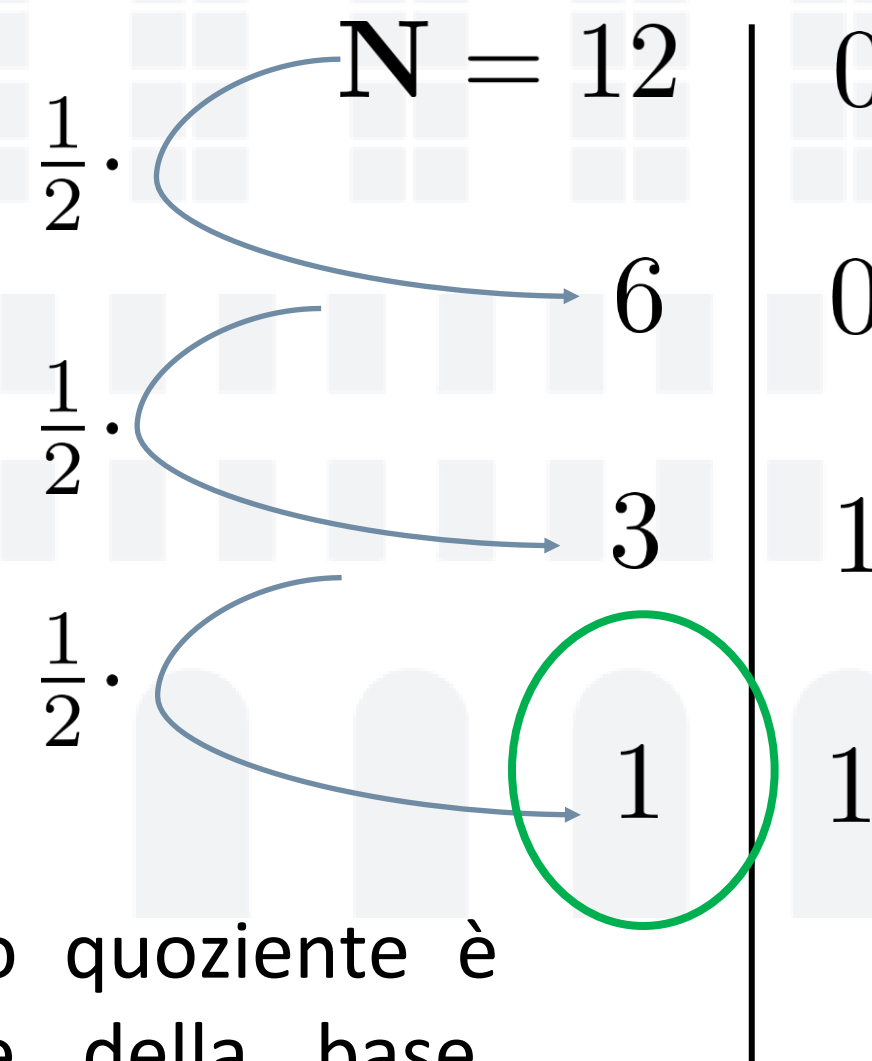
*Come si fa??*

# CONVERSIONE DA DECIMALE A QUALSIASI ALTRA BASE

Algoritmo da usare:

1. **Dividere** il valore numerico **N** per la base **r** fino a quando l'ultimo quoziente è minore della base stessa. In questo processo, tenere traccia di tutti i quozienti e tutti i resti.
2. «**Leggere**», in ordine inverso, tutti i resti delle divisioni. **Scrivere** tutti i valori da sinistra verso destra.

Esempio da **decimale**  
a **binario**:

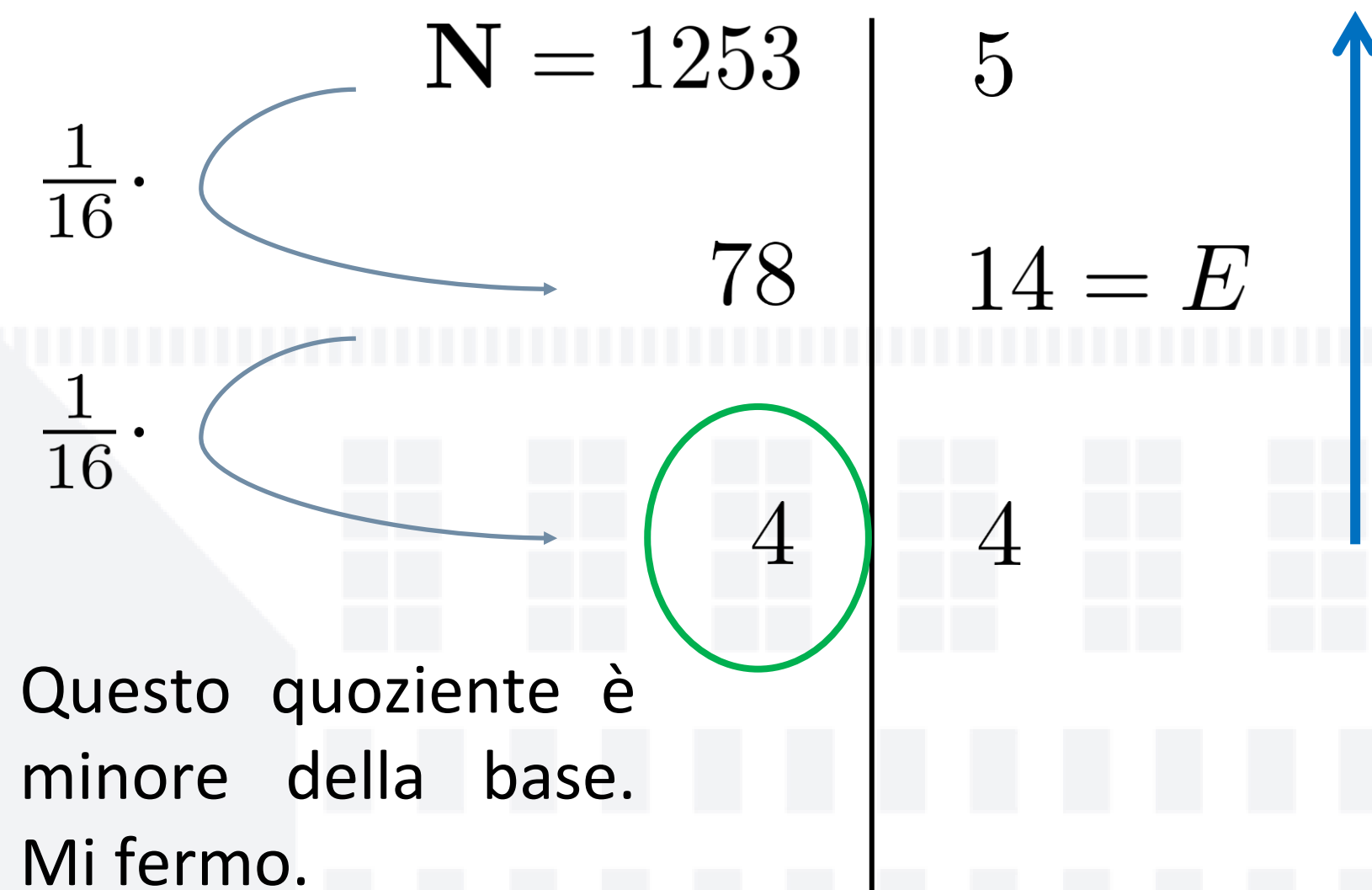


$$N = (12)_{10} = (1100)_2$$

Questo quoziente è  
minore della base.  
Mi fermo.

# CONVERSIONE DA DECIMALE A QUALSIASI ALTRA BASE

Esempio da **decimale** a **esadecimale**:



$$N = (1253)_{10} = (4E5)_{16} = 0x4E5$$

# CONVERSIONE DA BINARIO A ESADECIMALE

Algoritmo da usare:

1. **Divido** la sequenza binaria in gruppi da 4 bit
2. **Trasformo** ogni gruppo di 4 bit così formato nella corrispondente cifra dell'alfabeto esadecimale

$$\begin{array}{cccc} 0111 & 1111 & 0001 & 1010 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 7 & 15 = F & 1 & 10 = A \end{array}$$
$$(0111111100011010)_2 = (7F1A)_{16}$$

# CONVERSIONE DA ESADECIMALE A BINARIO

Provate a convertire questo numero in decimale:

$$N = (AB2)_{16}$$

$$A_{16} = 10_{10} = 1010_2$$

$$B_{16} = 11_{10} = 1011_2$$

$$2_{16} = 2_{10} = 0010_2$$

$$AB2_{16} = 101010110010_2$$

Esercizi per casa:

$$101011_2 = XX..X_8$$

$$5371_8 = YY.....Y_2$$

$$742_8 = ZZ..Z_{16}$$

**Suggerimento:**  
quanti bit ci vogliono  
per rappresentare  
una cifra nella nuova  
base numerica?



# NOTE SULLA BASE ESADECIMALE

Vantaggio: permette di ottenere una **rappresentazione molto più compatta** del sistema numerico binario o decimale.

$$A_{16} = 10_{10} = 1010_2$$

$$B_{16} = 11_{10} = 1011_2$$

$$C_{16} = 12_{10} = 1100_2$$

$$D_{16} = 13_{10} = 1101_2$$

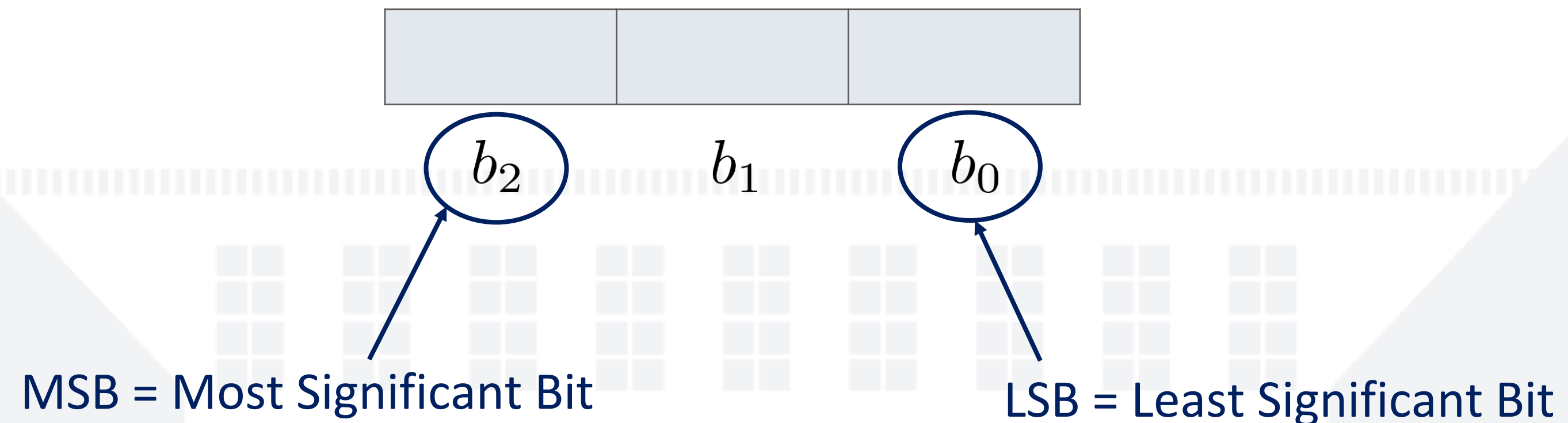
$$E_{16} = 14_{10} = 1110_2$$

$$F_{16} = 15_{10} = 1111_2$$

Applicazioni: codifica degli [indirizzi sorgente/destinazione nel protocollo IP](#), codifica simboli di testo nel [codice ASCII](#), [codifica colore nel web design](#) nel linguaggio CSS.

# RAPPRESENTABILITA' E *OVERFLOW*

Supponendo di voler rappresentare solo valori interi, qual è il range di valori decimali che posso rappresentare con  $n=3$  bit?

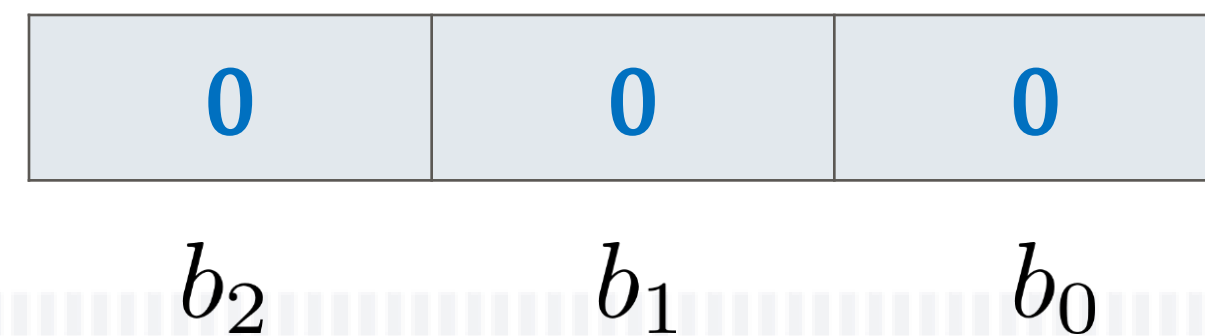


# RAPPRESENTABILITA' E OVERFLOW

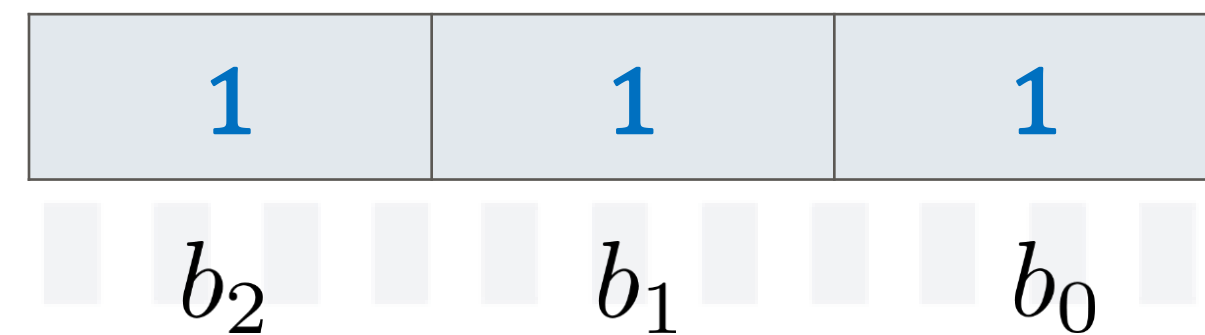
Supponendo di voler rappresentare solo valori interi, qual è il range di valori decimali che posso rappresentare con  $n=3$  bit?

Con  $n$  bit, posso rappresentare  $2^n$  valori interi

Con  $n=3$ , possiamo rappresentare tutti i numeri decimali interi da **0 a 7**, ovvero  **$2^3=8$  cifre decimali**. Posso quindi rappresentare **tutti i valori da 0 a  $2^n-1$** .



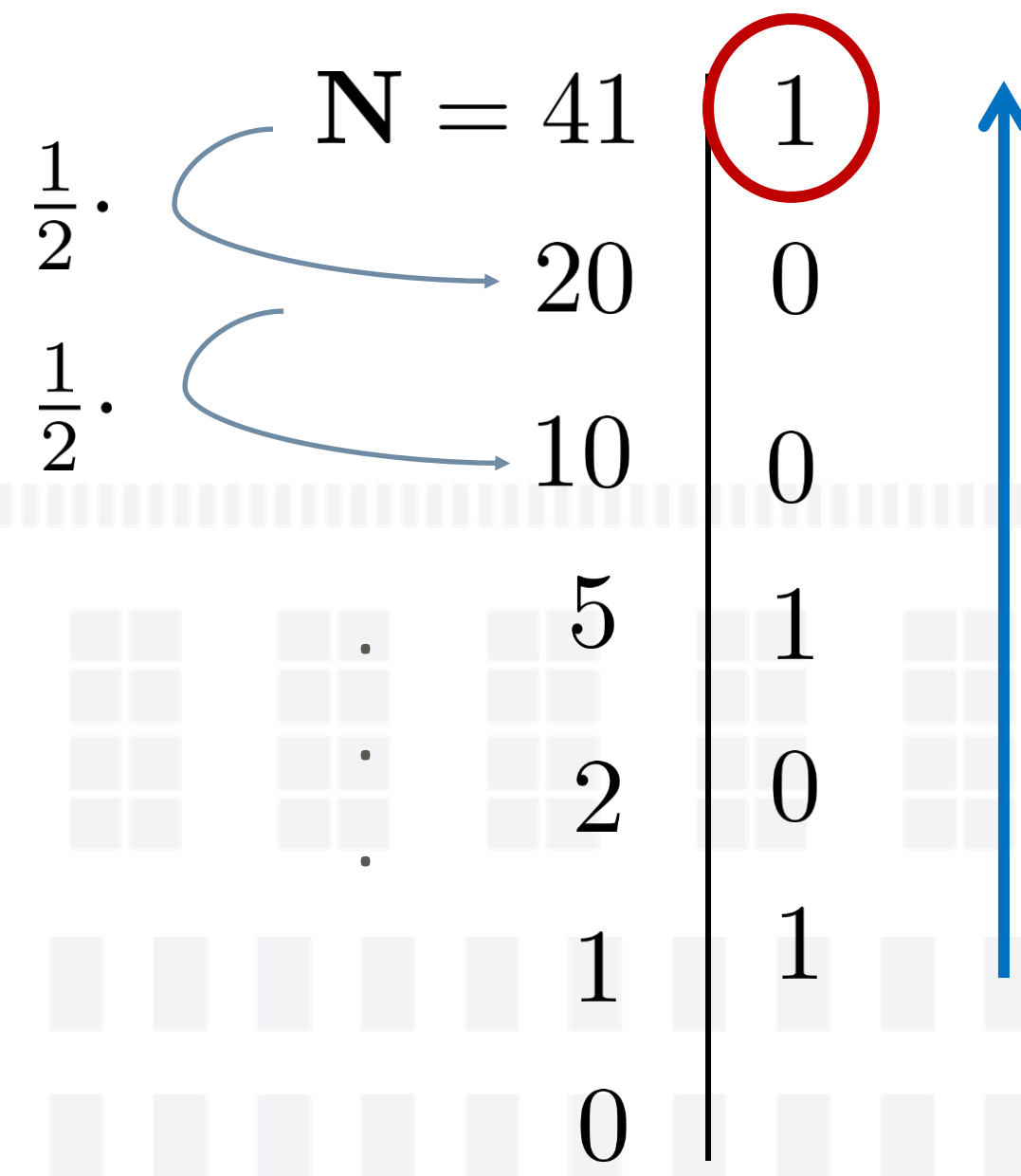
$$N = 000_2 = 0_{10}$$



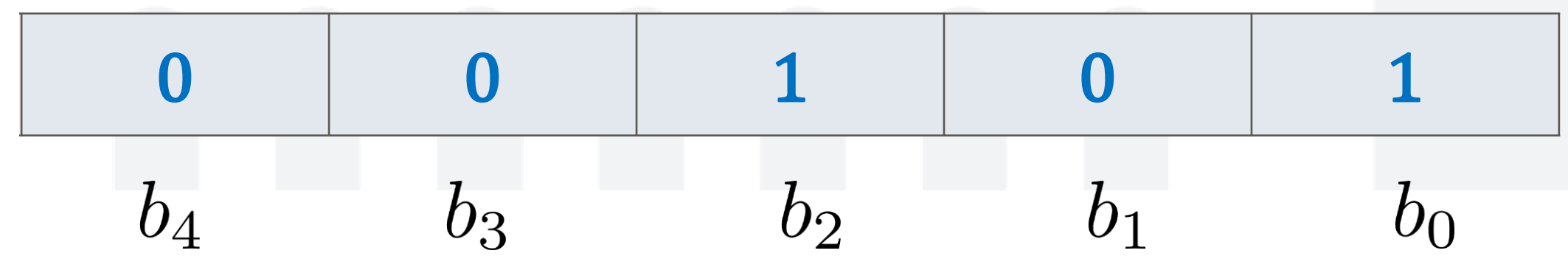
$$N = 111_2 = 7_{10}$$

# RAPPRESENTABILITA' E *OVERFLOW*

Posso rappresentare il numero 41 con  $n=5$  bit?



**Overflow!**



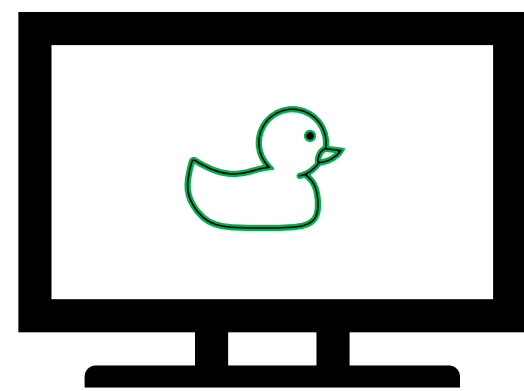
# RAPPRESENTABILITA' E *OVERFLOW*



Si ha **OVERFLOW** quando non si può rappresentare il dato con il numero di bit a disposizione. Questo può accadere anche come risultato di operazioni sui numeri già rappresentati (es. somme, sottrazioni..)

La **rappresentabilità** dei valori è quindi vincolata al numero di cifre disponibili (es. al numero di bit disponibili).

*In tutti i sistemi il numero di cifre è limitato!*



[DELL](#)

L'intensità di colore supportata dai monitor DELL è a:

- **6 bit:** 262 mila colori
- **8 bit:** 16,7 milioni di colori
- **10 bit:** 1,07 miliardi di colori

La colorazione è definita con il **sistema RGB** che ha **3 canali (rosso, verde, blu)**. Ogni canale è rappresentato da 6 bit. Il colore di ogni pixel è combinazione della particolare sfumatura di rosso, di verde, di blu. Ci sono  $2^6=64$  possibili sfumature di ogni singolo colore. Quindi il colore di ogni pixel ha  $64 \times 64 \times 64 = 262$  mila possibili tonalità di colore finale.

## ***ALLERTA NOTAZIONE PARTICOLARE!***

Il sistema numerico binario si basa su **potenze di 2**, non di 10 come il sistema decimale.

Per convenzione:

	SISTEMA DECIMALE	SISTEMA BINARIO
KILO	$10^3 = 1000$	$2^{10} = 1024$
MEGA	$10^6 = 1000000$	$2^{20} = 1048576$
GIGA	$10^9 = 1000000000$	$2^{30} = 1073741824$

*Quindi quanti byte sono 4 kilo byte?*

$$4\text{kB} = 4 \times 1024 \text{ byte} = 4096 \text{ byte!}$$