

# Rappresentazione dell'informazione

*Corso di Elementi di Programmazione*

---

Prof. Salvatore Venticinque

# I requisiti

---

Per realizzare l'esecuzione automatica dell'informazione occorre:

- Disporre di un elaboratore
- ***Rappresentare l'informazione (dati)***
- Rappresentare l'elaborazione (programma)

# Informazione

---



Il numero di telefono di casa di Andrea è 081 7651831

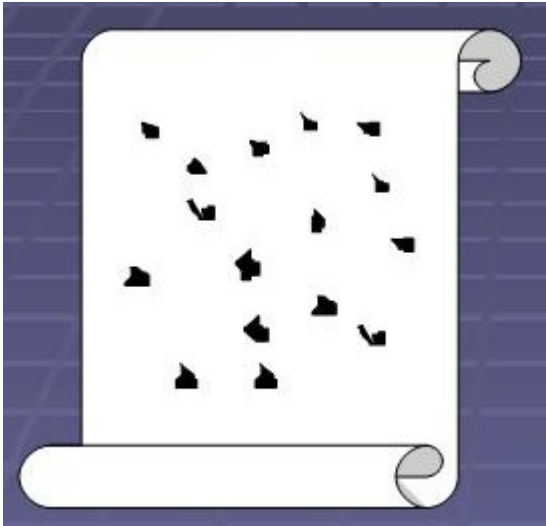
# Cos'è l'informazione

---

- Qualcosa che si può comunicare
- Qualcosa in funzione di cui si possono operare delle scelte
- Qualcosa che si può conservare
- ...

# Rappresentazioni e formati

---



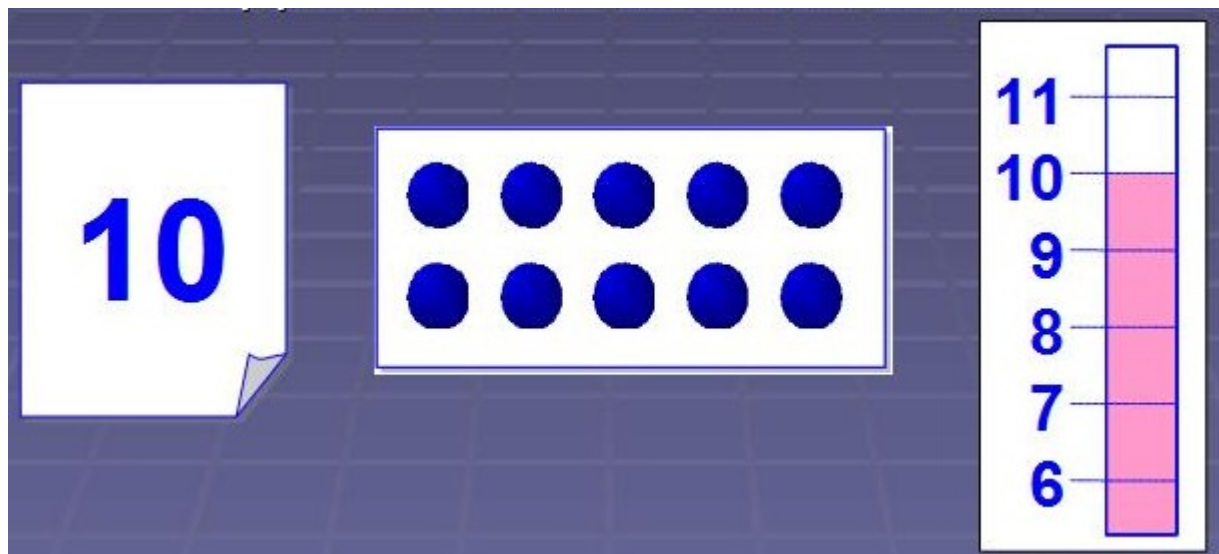
Immagine

- Un foglio pieno di macchie

Testo

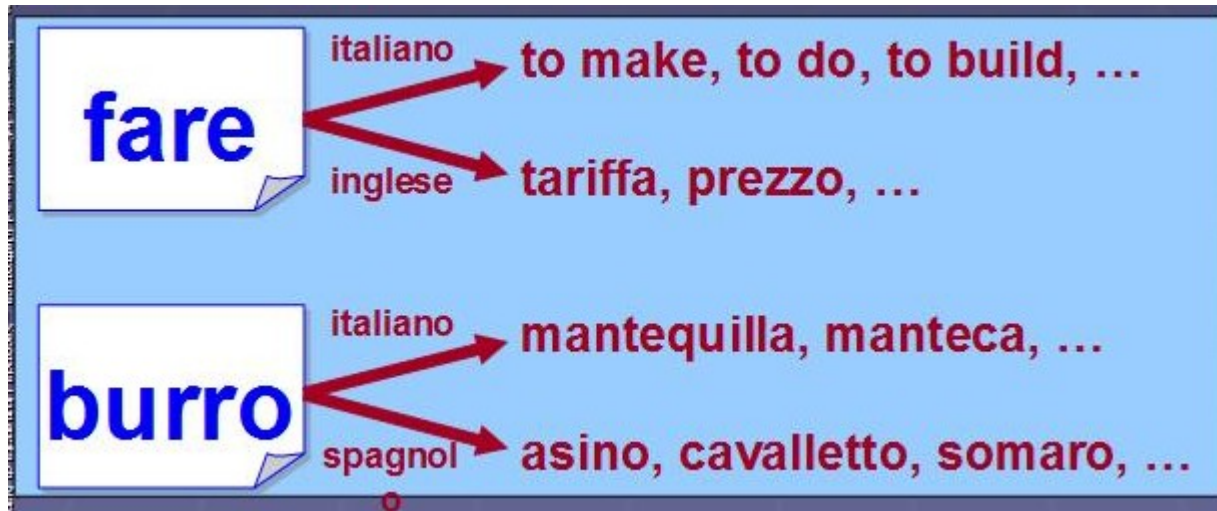
# Diverse rappresentazioni

---



# Rappresentazione e significato

---



# Necessità della rappresentazione

---

- L'informazione esiste a prescindere dalla sua rappresentazione:

*Il numero 4 astratto esiste a prescindere dal fatto che lo si rappresenti*

- Ma senza una rappresentazione non è possibile
  - Elaborarla
  - Memorizzarla
  - Comunicarla



# Codifica

---

La codifica è l'operazione che consente di rappresentare un'informazione definiti:

- Un insieme di simboli
- Una regola di rappresentazione

Un dato è:

- la rappresentazione di un'informazione secondo una determinata codifica.

# Codificare l'informazione

---

- Stabilire un'insieme di simboli
- Stabilire una informazione tra informazione e simboli (il codice)

# Informazione binaria

---

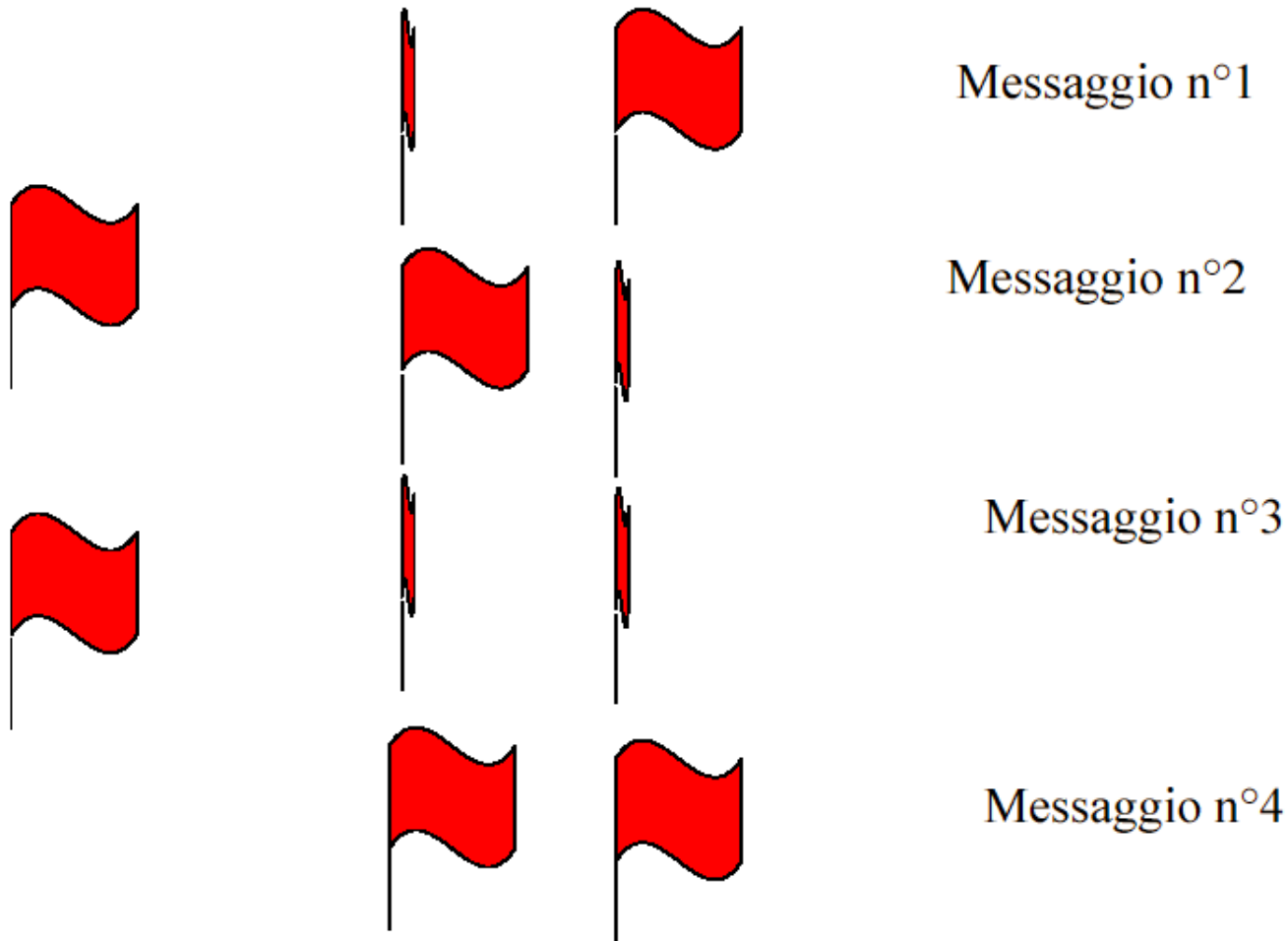
Bandiera= strumento di rappresentazione



1 bandiera  $\rightarrow$  2 messaggi diversi

# Utilizzo di più simboli binari

---



2 bandiere → 4 messaggi diversi

# Codice MORSE

Lettera	Morse	Lettera	Morse	Lettera	Morse
0	- - - - -	C	- . - .	O	- - -
1	. - - - -	D	- . .	P	. - - -
2	. . - - -	E	.	Q	- - . -
3	. . . - -	F	. . - .	R	. - .
4	. . . . -	G	- - .	S	. . .
5	. . . . .	H	. . . .	T	-
6	- . . . .	I	. .	U	. . -
7	- - . . .	J	. - - -	V	. . . -
8	- - - . .	K	- . -	W	. - -
9	- - - - .	L	. - . .	X	- . . -
A	. -	M	- -	Y	- . - -
B	- . . .	N	- .	Z	- - . .

# Stringhe di lunghezza assegnata

---

- Una sequenza di valori appartenenti a un insieme finito  $R$  viene detta *stringa*
- Una stringa è caratterizzata dalla sua lunghezza
- L'insieme delle stringhe di elementi di  $R$  di lunghezza  $m$  è il prodotto cartesiano

$$R^m = R \times \cdots \times R$$

  
*m* volte

# Binary Digit (bit)

---

- $R \equiv \{ 0, 1 \}$
- Può rappresentare qualunque informazione a due valori ( $|D| = 2$ )
- Una stringa di  $m$  bit può assumere  $2^m$  valori diversi
- Esempio:

♥	00
♦	01
♣	10
♠	11

$$|D| = 4$$

$$m = 2$$

# Codifica con stringhe di bit

- Per un qualunque insieme  $D$  finito:

$$c : D \rightarrow \{0, 1\} \times \cdots \times \{0, 1\}$$



$\lceil \log_2 |D| \rceil$  volte

- Esempio:

Lunedì	000	111
Martedì	001	001
Mercoledì	010	110
Giovedì	011	000
Venerdì	100	101
Sabato	101	100
Domenica	110	010

$$|D| = 7$$

$$\lceil \log_2 |D| \rceil = 3$$

$$2^3 = 8$$

codifica che non usa 111

codifica che non usa 011



## Rappresentazione posizionale dei numeri naturali

---

- *Numeri e rappresentazione* dei numeri:

**quindici**

15    XV     $1111_2$

- Rappresentazione posizionale: base di rappresentazione (es. 10), si usano 10 simboli (*cifre*) che rappresentano i numeri da 0 a 9

$$15 = 1 \times 10^1 + 5 \times 10^0$$

- Il numero viene rappresentato dalla lista di cifre

# Rappresentazione dei caratteri

---

- Codice ASCII (American Standard Code for Information Interchange)
- Rappresentazione su 7 bit: 128 combinazioni
  - da 0 a 31: “caratteri” di controllo
  - da 32 a 47: interpunzione e caratteri speciali
  - da 48 a 57: cifre decimali
  - da 58 a 64: interpunzione e caratteri speciali
  - da 65 a 90: lettere maiuscole dell’alfabeto inglese
  - da 91 a 96: interpunzione e caratteri speciali
  - da 97 a 122: lettere minuscole dell’alfabeto inglese
  - da 123 a 127: caratteri speciali

carattere	codice ASCII	equivalente esadecimale	equivalente numerico	carattere	codice ASCII	equivalente esadecimale	equivalente numerico
@	01000000	40	64	‘	01100000	60	96
A	01000001	41	65	a	01100001	61	97
B	01000010	42	66	b	01100010	62	98
C	01000011	43	67	c	01100011	63	99
D	01000100	44	68	d	01100100	64	100
E	01000101	45	69	e	01100101	65	101
F	01000110	46	70	f	01100110	66	102
G	01000111	47	71	g	01100111	67	103
H	01001000	48	72	h	01101000	68	104
I	01001001	49	73	i	01101001	69	105
J	01001010	4A	74	j	01101010	6A	106
K	01001011	4B	75	k	01101011	6B	107
L	01001100	4C	76	l	01101100	6C	108
M	01001101	4D	77	m	01101101	6D	109
N	01001110	4E	78	n	01101110	6E	110
O	01001111	4F	79	o	01101111	6F	111
P	01010000	50	80	p	01110000	70	112
Q	01010001	51	81	q	01110001	71	113
R	01010010	52	82	r	01110010	72	114
S	01010011	53	83	s	01110011	73	115
T	01010100	54	84	t	01110100	74	116
U	01010101	55	85	u	01110101	75	117
V	01010110	56	86	v	01110110	76	118
W	01010111	57	87	w	01110111	77	119
X	01011000	58	88	x	01111000	78	120
Y	01011001	59	89	y	01111001	79	121
Z	01011010	5A	90	z	01111010	7A	122
[	01011011	5B	91	{	01111011	7B	123
\	01011100	5C	92		01111100	7C	124
]	01011101	5D	93	}	01111101	7D	125
^	01011110	5E	94	~	01111110	7E	126
_	01011111	5F	95	□	01111111	7F	127

# Rappresentazione dei caratteri

---

- Relazioni tra caratteri e numeri
  - le stringhe di bit non hanno significato di per se: a ogni carattere corrisponde un numero da 0 a 255
  - il valore numerico di una cifra si ottiene sottraendo al numero corrispondente alla cifra quello corrispondente a 0
- Ordinamento dei caratteri:
  - rispettato l'ordinamento relativo tra: cifre, maiuscole, minuscole
  - spazio < cifre < maiuscole < minuscole

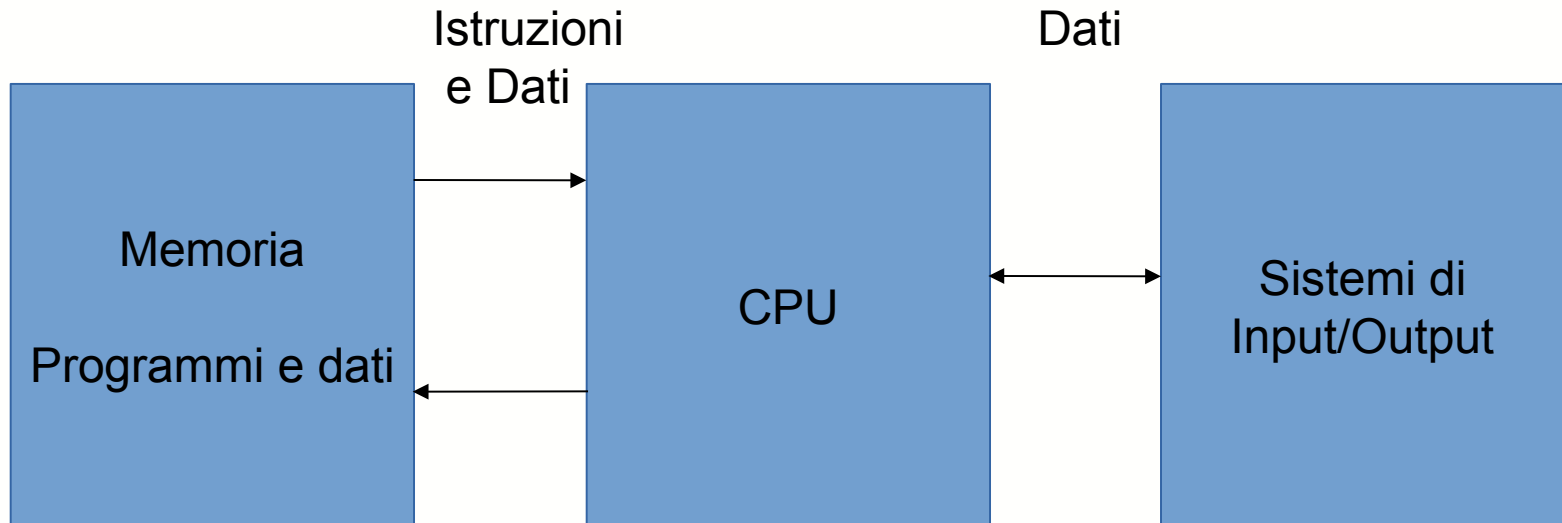
## Significato della rappresentazione (codice)

---

- Un codice non ha significato di per sé
- Il valore è stabilito dalla codifica (cioè dalla funzione  $c$ )
- L'associazione stringa-codifica (cioè il *tipo*) è data dall'operatore umano
- Ad esempio, la stringa 1000 0101 rappresenta:
  - il numero naturale **133** in binario naturale
  - il numero naturale **-123** in complemento a 2
  - il carattere **à** in codice ASCII esteso

# Modello Von Neumann

---



# Supporto

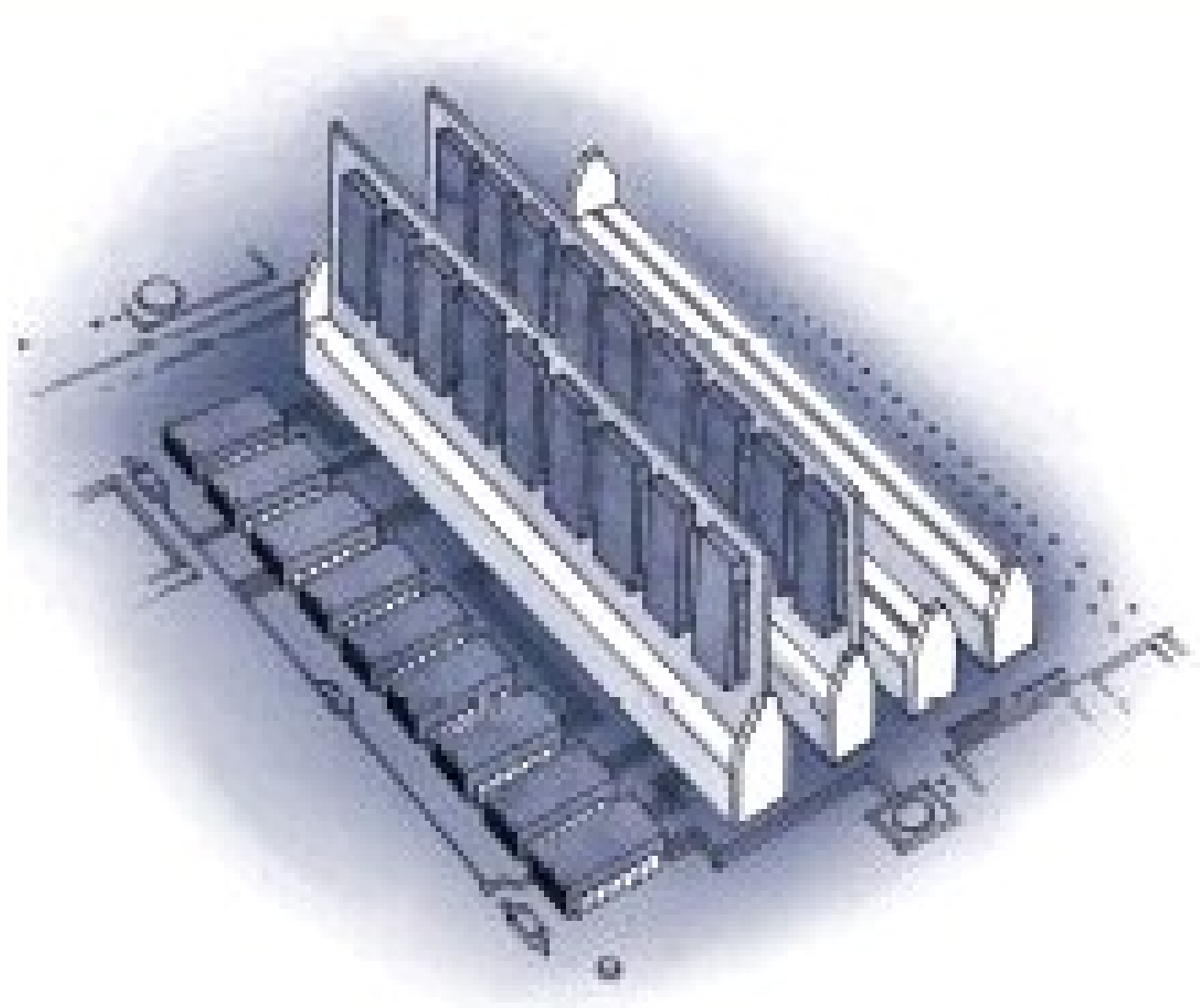
---

E' necessario un supporto per:

- comunicare
- memorizzare

# Memory

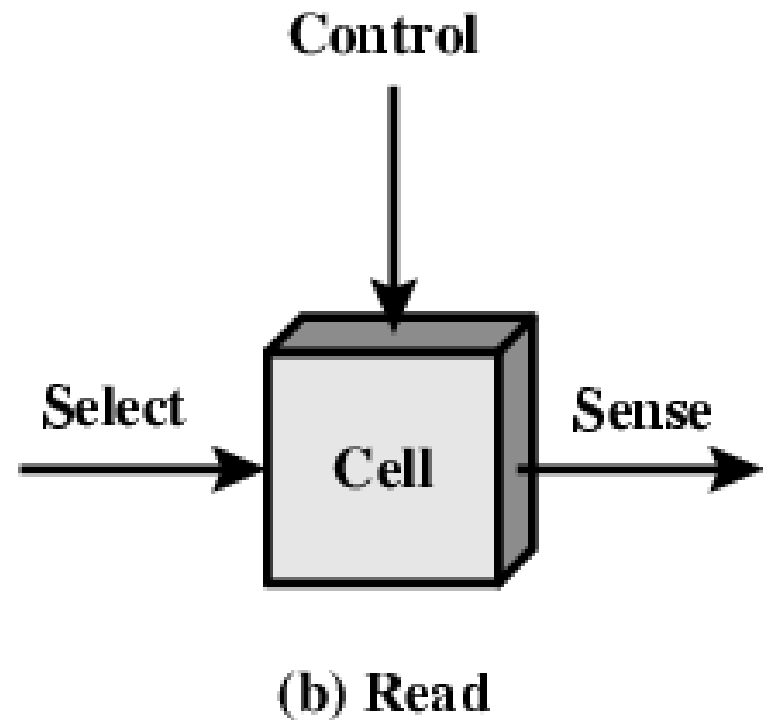
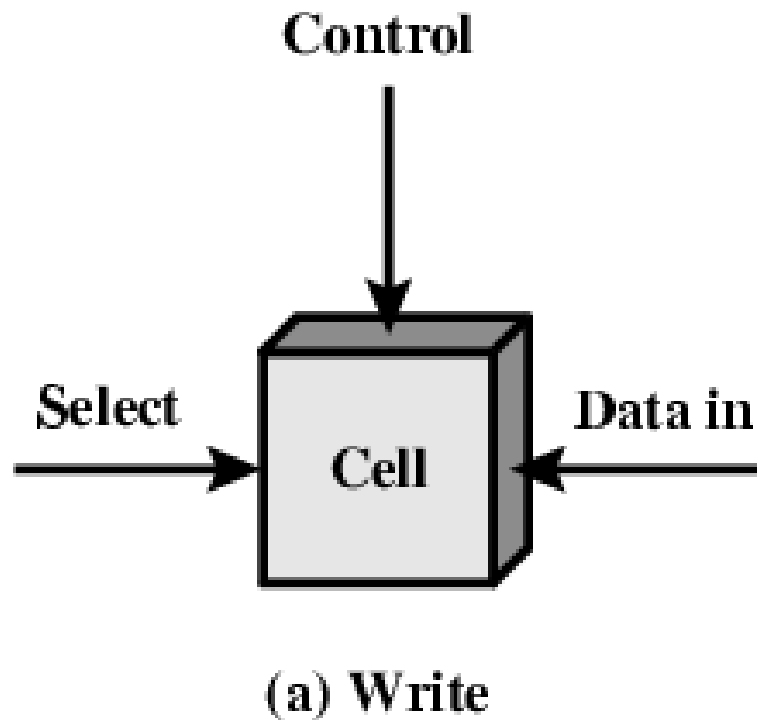
---





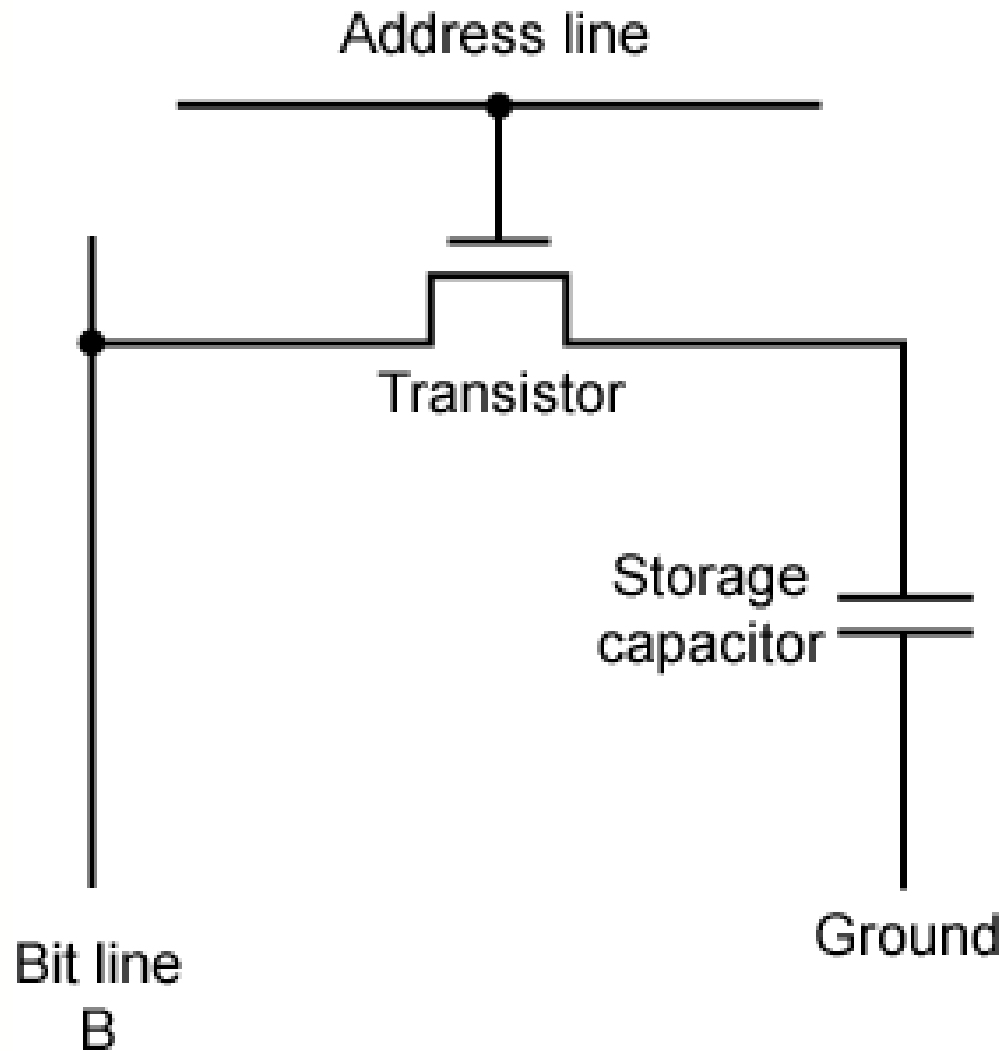
# Memory Cell Operation

---



# Dynamic RAM Structure

---



# Array of bytes

- Address: byte line
- Value: stored binary array

## Binary representation of information:

- Instruction
- Data

[illegible]

# Problema: Realizzazione

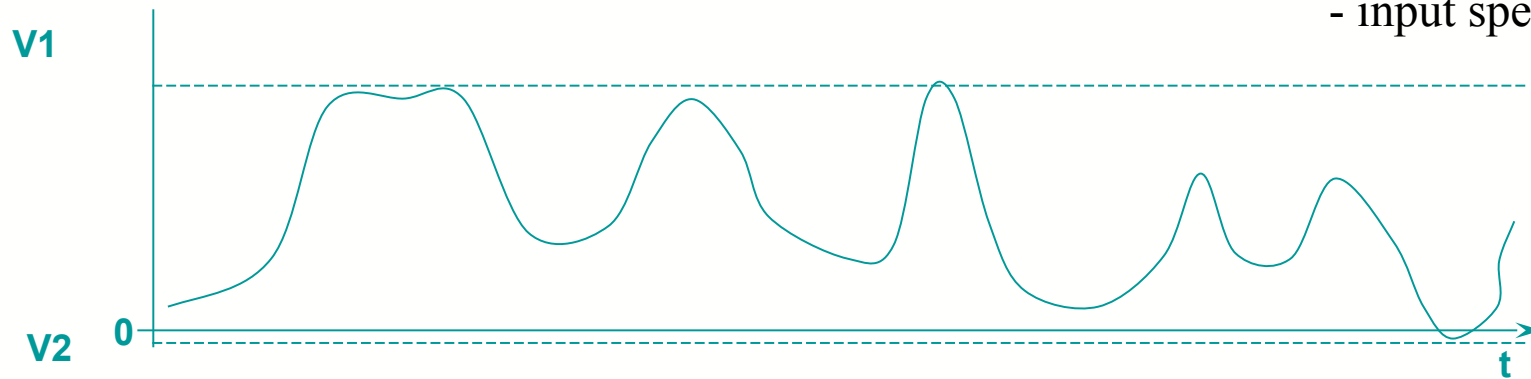
---

- Come rappresentare l'informazione?
  - Il mondo è analogico!!!
- Come memorizzare i dati ?
  - La migliore tecnologia per la costruzione della memoria!
- Come elaborare i dati ?
  - Dipende da come sono rappresentati ...

# Grandezze analogiche e digitali

- Una grandezza analogica varia con continuità in un qualunque intervallo di valori.

Es:  
- radio FM, AM  
- output microfono  
- input speaker



La funzione rappresentata è limitata nei valori e varia con continuità nel tempo e nell'intervallo  $V1, V2$ .

# Campionamento e quantizzazione

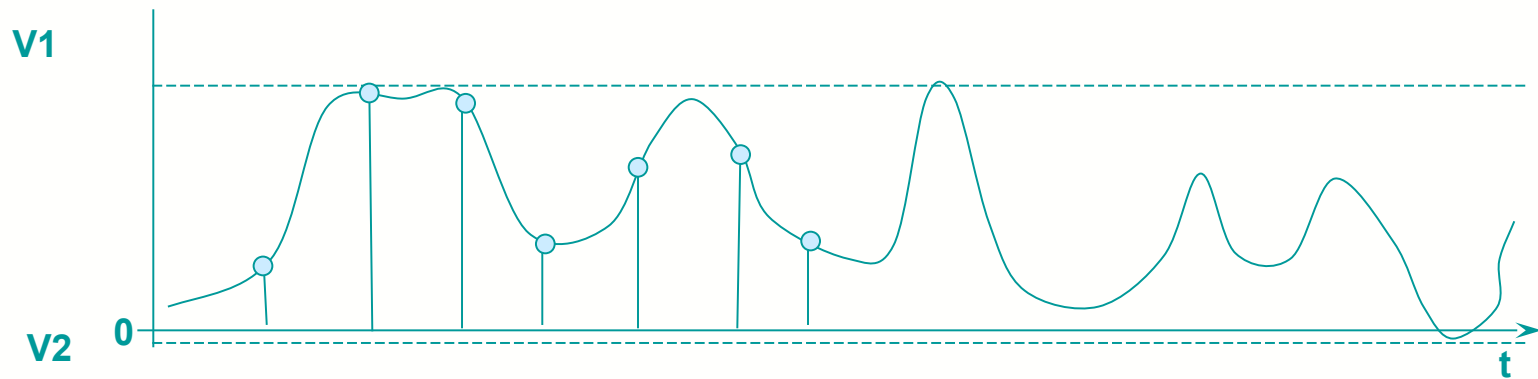
---

- Una grandezza analogica non è rappresentabile in un calcolatore che possiede una memoria limitata.
- Occorre:
  - Campionare
  - Quantizzare

# Campionamento

---

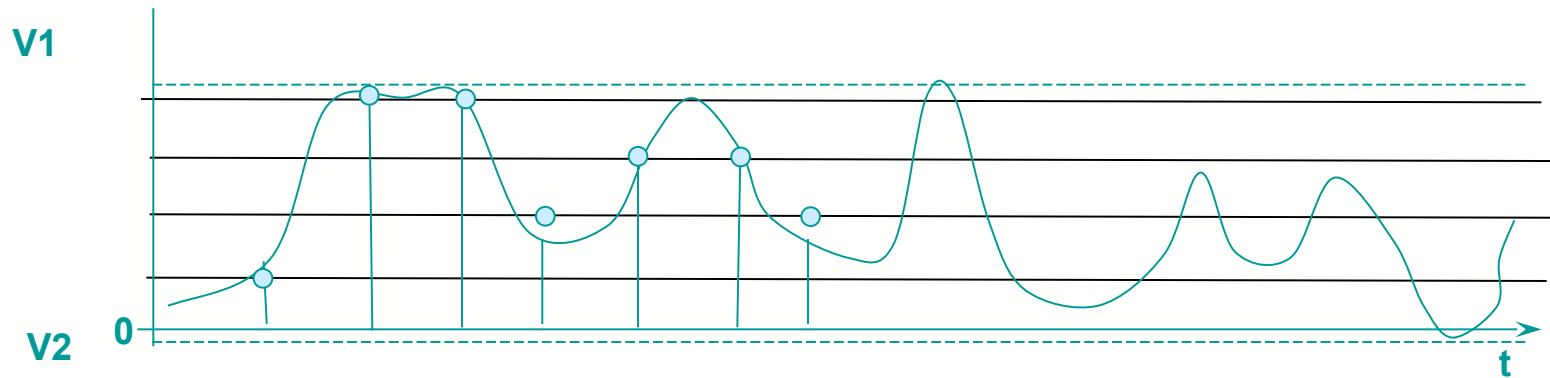
- Con il campionamento e la quantizzazione si converte la grandezza da analogica in digitale:



Il campionamento consiste nel prelevare solo alcuni campioni al variare del tempo.

# Quantizzazione

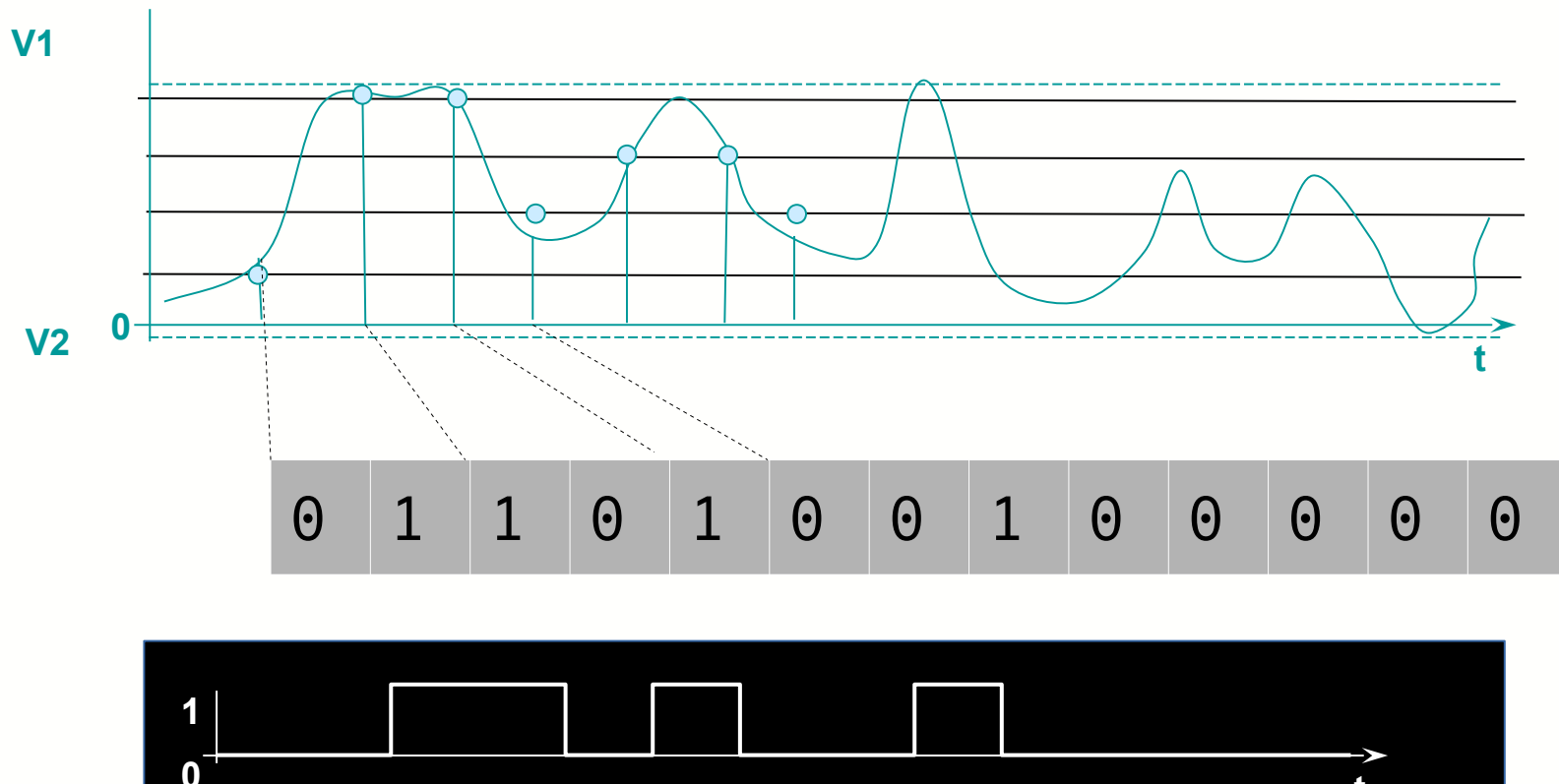
- Con il campionamento e la quantizzazione si converte la grandezza da analogica in digitale:



La quantizzazione consiste nell'approssimare i valori reali al quello più vicino in un set finito e prefissato.



# Campionamento



---

Unità di misura	simbolo	Equivale a	cioè
byte	B	8 bit	Un carattere alfanumerico
kilobyte	kB	$2^{10}$ byte	Un terzo di pagina di testo
megabyte	MB	$2^{20}$ byte	Circa 300 pagine di testo
gigabyte	GB	$2^{30}$ byte	Circa 300000 pagine di testo
terabyte	TB	$2^{40}$ byte	Circa 300 milioni di pagine di testo