

# Memorizzazione dei Dati

## Capitolo 1 del testo

---

Alberto Policriti



17 Ottobre, 2019

## Bits

## Binary Digits

- numeri,
- cifre in base 2,
- simboli,
- acceso/spento,
- si/no,
- vero/falso,
- ...

## Stringhe

*stringhe di bits*  $\Rightarrow$  *stringhe in un alfabeto di 2 caratteri*  $\Rightarrow$  ...

*stringhe in un alfabeto di 4 caratteri*  $\Rightarrow$  DNA

I bit possono essere visti anche come la quantità di informazione contenuta in un carattere:  $S$  insieme qualsiasi

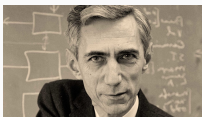
Quante domande devo fare per trovare  $x \in S$ ?

**Entropia (di un insieme  $S$ )**

$$\mathcal{H}(S) = \log |S|$$

Quanta *informazione* contiene un elemento in  $S$ ?

BITs vs. SHANNONs



## Operazioni Booleane

AND    OR    NOT    XOR    IF-THEN    etc.

### The AND operation

$$\begin{array}{r} 0 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{AND } 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 1 \\ \hline 1 \end{array}$$

### The OR operation

$$\begin{array}{r} 0 \\ \text{OR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

### The XOR operation

$$\begin{array}{r} 0 \\ \text{XOR } 0 \\ \hline 0 \end{array}$$

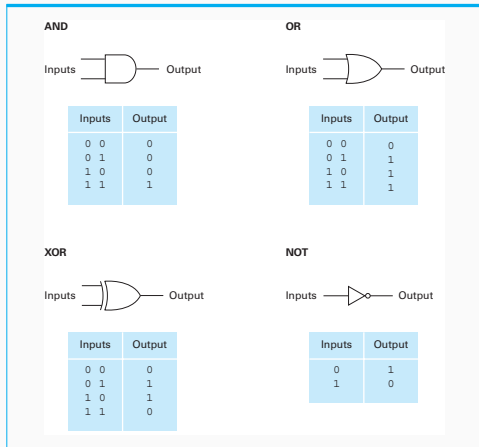
$$\begin{array}{r} 0 \\ \text{XOR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 0 \\ \hline 1 \end{array}$$

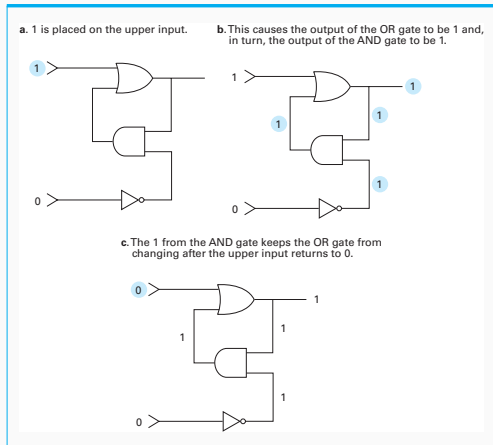
$$\begin{array}{r} 1 \\ \text{XOR } 1 \\ \hline 0 \end{array}$$

## Porte (logiche)—gates

Realizzazioni in *hardware* degli operatori del calcolo proposizionale.



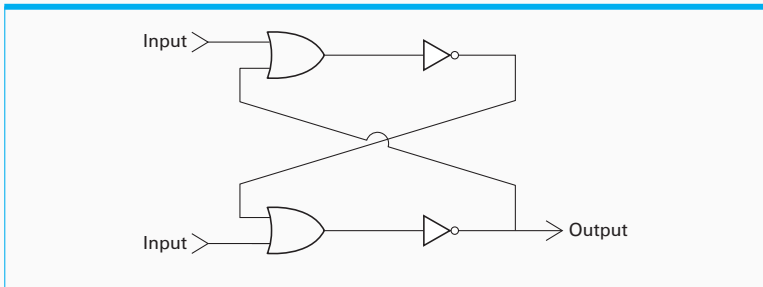
## Memorizzazione di un bit di informazione: **flip-flop** Realizzazione in *hardware* della memoria per 1 bit.



## Flip-Flop

1. Combinando porte ottengo circuiti che hanno comportamenti complessi (memoria).
2. Una volta costruito il circuito non mi interessa più ai dettagli: astrazione.
3. *Tanti circuiti tanta* potenza (in poco spazio: Very Large Scale Integration—VLSI—, computers on a chip)

**Esercizio: un altro modo di implementare un flip-flop**  
(in realtà non vengono implementati né così né come prima).





## Stringhe, Pattern, Stream, ... di bits

Idea (vecchia): cambiamo base—i.e. introduciamo nuovi simboli che corrispondono a stringhe di bit.

### Example

La *codifica* dei numeri naturali in base 16 (hexadecimal encoding system):

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

# Memoria Principale: RAM (Random Access Memory)



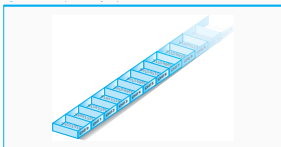
# Memoria Principale: RAM (Random Access Memory)

## Terminologia

- **main memory**: hw per memorizzare tutti i bit che mi servono durante il calcolo (... “tanti flip-flop”);
- **cella di memoria**: collezioni di bit (normalmente 8) che vengono manipolati insieme;
- **byte**: 8 bit;



- **indirizzo (di una cella)** : posizione della cella nella lista delle celle:



2. Suppose you want to interchange the values stored in memory cells 2 and 3. What is wrong with the following sequence of steps:

*Step 1.* Move the contents of cell number 2 to cell number 3.

*Step 2.* Move the contents of cell number 3 to cell number 2.

Design a sequence of steps that correctly interchanges the contents of these cells. If needed, you may use additional cells.

### I dati (le celle) sono ordinati

- Posso parlare non solo della cella che si trova ad un dato indirizzo, ma anche della cella che *viene dopo/prima ...*
- Posso memorizzare stringhe lunghe usando *celle consecutive*

### RAM

Posso accedere ad una cella qualunque semplicemente specificandone l'indirizzo

DRAM: Dynamic RAM (*refresh*)

SDRAM: Synchronous DRAM (*synchronous refresh*)

- 1 bit: 2 informazioni
- 2 bit: 4 informazioni
- 3 bit: 8 informazioni
- ...
- 8 bit: 256 informazioni  $\Rightarrow$  un byte B.

- Kilo byte:  $2^{10} = 1024$  B
- Mega byte:  $2^{10} = 1024$  K
- Giga byte:  $2^{10} = 1024$  M
- Tera byte:  $2^{10} = 1024$  G
- Peta byte:  $2^{10} = 1024$  T
- ...

## Convenzioni

Memoria in **byte**.

Banda di trasmissione in **bit**.

## Memorie di massa: memoria non volatile



### On-line and Off-line

Memorie *lente*. Spesso richiedono (addirittura) movimenti di parti *meccaniche*.

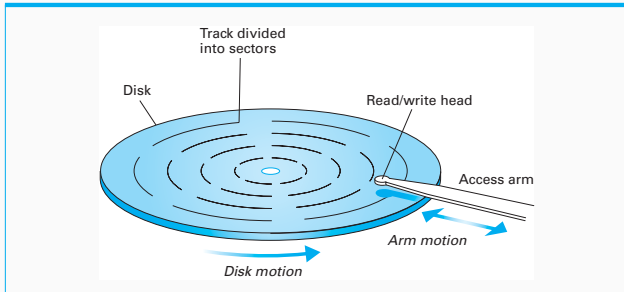
Solid-state vs. mechanical



## Esempi

- dischi magnetici
- CD
- DVD
- nastri magnetici
- flash drives

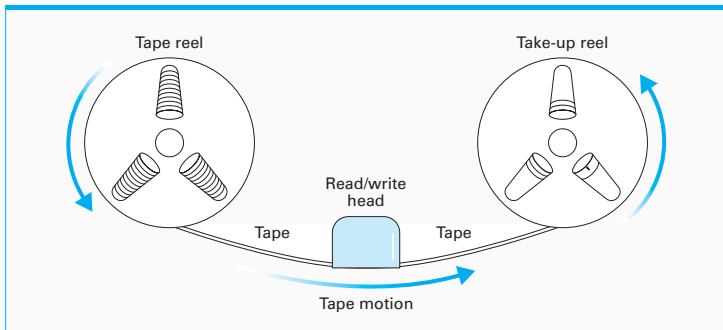
## Dischi Magnetici



Tecnologicamente molto sofisticati (e.g. coated, zoned)

- cilindri
- settori
- **formattazione**
- seek time
- latency time
- access time

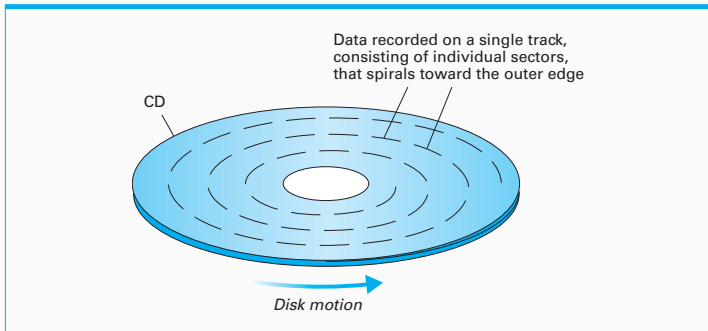
## Nastri Magnetici



### Basso transfer rate (associato ad ogni tecnologia)

- Molti fattori coinvolti.
- Molte tecniche per migliorarlo.
- Dell'ordine dei MB al secondo

## Compact Disks



## Digital Versatile Disks e Blu-ray Disks

DVD: superfici sovrapposte con diversi fuochi

BD: usano la parte blu dello spettro della luce

### SSD: niente più parti in movimento!

- Spedisco segnali e memorizzo informazioni *intrappolando* elettroni.
- In linea di principio dovrebbero funzionare come delle RAM ma le tecnologie di oggi consentono solo il trasferimento di (grossi) *blocchi* di dati.
- Non hanno una vita molto lunga.

### Varianti

1. Secure Digital memory cards (SD)
2. Secure Digital High Capacity memory cards (SDHC)
3. Secure Digital Extended Capacity memory cards (SDXC, arrivano ai TB!)

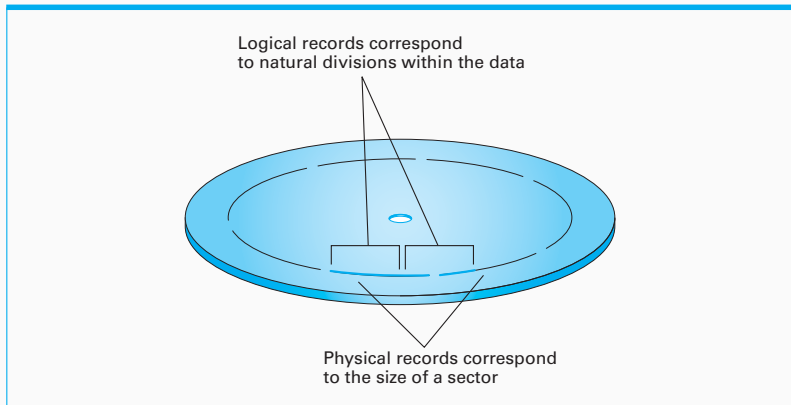
## Cos'è un file?

Formalmente si definisce come una:

*unità concettuale di memoria di massa*

## Definizioni (usiamo l'inglese)

- **physical record** blocco minimo trasferibile dalla tecnologia utilizzata per la memorizzazione dei dati
- **logical record** blocco naturalmente definibile in funzione del tipo di dati memorizzati
- **field** sotto-unità dei record logici
- **chiave** campo identificativo (ci ritorneremo parlando di basi di dati)
- **buffer** (in questo caso) regione della RAM usata per contenere adeguate quantità di record fisici e per indirizzare record logici



American National Standards Institute (ANSI)

**Caratteri:** codice **ASCII**

American Standard Code for Information Interchange:

7 bit: 128 combinazioni

codifico lettere maiuscole, minuscole e digit da 0 a 9 ...

Non è l'unico codice ma l'idea è sempre la stessa.

International Organization for Standardization (ISO)



# Rappresentazione dei dati: testo

## La tabella:

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-`	63	3F	?	95	5F	`	127	7F	DEL

### Esercizio

Che differenza c'è tra la rappresentazione di

124924596

come testo o come numero?

## Text files

- text editors
- word processors

## Dati numerici

- Numeri naturali: (semplice) rappresentazione binaria.
- Numeri interi: uso il primo bit per il segno.
- Numeri reali:
  - virgola fissa: un intero (parte intera) ed un razionale (parte decimale);
  - virgola mobile: mantissa ed esponente.

## Pixel (Picture element)

- bit map: un pixel un(o o più) bit(s)
  - ok per printer e schermi,
  - un pixel può essere 0/1 o un valore numerico (scala di grigi, RGB, ...),
  - problematico se dobbiamo riscaldare l'immagine;
- rappresentazione vettoriale: l'immagine è rappresentata come collezione di oggetti geometrici ed il dispositivo “decide” come rappresentarla (deve essere in grado di farlo ⇒ CPU)
  - TrueType (Microsoft and Apple per i caratteri)
  - PostScript (Adobe Systems per i caratteri e non solo)

## Suono

- ampiezze d'onda
- MIDI (Musical Instrument Digital Interface)

**Le istruzioni sono rappresentate come ...**  
... sequenze di bit costituite da due parti:



Che operazione voglio eseguire? Su quali dati?