

FONDAMENTI DI INFORMATICA

Massimiliano Salfi

massimiliano.salfi@unict.it



CALENDARIO LEZIONI

In presenza

martedì, 11.00 -13.00 (edificio 14, aula D02)

giovedì, 14.00 - 17.00 (edificio 14, aula D02)

venerdì 8.00 – 10.00 (edificio 14, aula D02)



RIFERIMENTI

Sito internet

<http://www.dmi.unict.it/salfi/home.htm>

Ricevimento

Solo per appuntamento.

Esami

Progetto e scritto/orale ?!?



PROGRAMMA

- Introduzione all'informatica
- Codifica e rappresentazione delle informazioni
- Algebra booleana
- Architettura dei calcolatori
- Sistema Operativo e applicativi software
- Reti di calcolatori ed internet
- Ipertesti e codice HTML
- La sicurezza in rete ed i malware
- Introduzione alle basi di dati
- Intelligenza Artificiale
- Il linguaggio di programmazione C
- Il pacchetto Microsoft Office



MATERIALE DIDATTICO

✓ *Slides del docente*

✓ Massimiliano Salfi

Fondamenti di informatica per ingegneri civili, ambientali e gestionali

Seconda Edizione – ISBD 9781307789744

McGraw-Hill

Il testo raccoglie le sole parti che tratteremo, dai seguenti testi:

✓ Luca Mari, Giacomo Bonanno, Donatella Sciuto

Informatica e cultura dell'informazione

McGraw-Hill - Seconda edizione

✓ Alessandro Bellini, Andrea Guidi

Linguaggio C

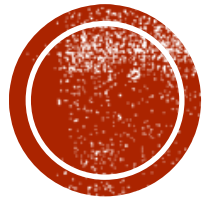
McGraw-Hill - VI edizione

✓ Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

Introduzione agli algoritmi e strutture dati

McGraw-Hill - Terza edizione





INTRODUZIONE ALL'INFORMATICA

CHE COS'È L'INFORMATICA?

Informatica

=

Informazione + Automatica

Si riferisce ai processi e alle tecnologie che rendono possibile l'immagazzinamento e l'elaborazione delle informazioni.



GLI ANTENATI DEL COMPUTER

- Macchina analitica di Babbage (1830).
- Macchina di Turing (1936).
- Macchina di Von Neumann (anni '40).



GLI ANTENATI DEL COMPUTER

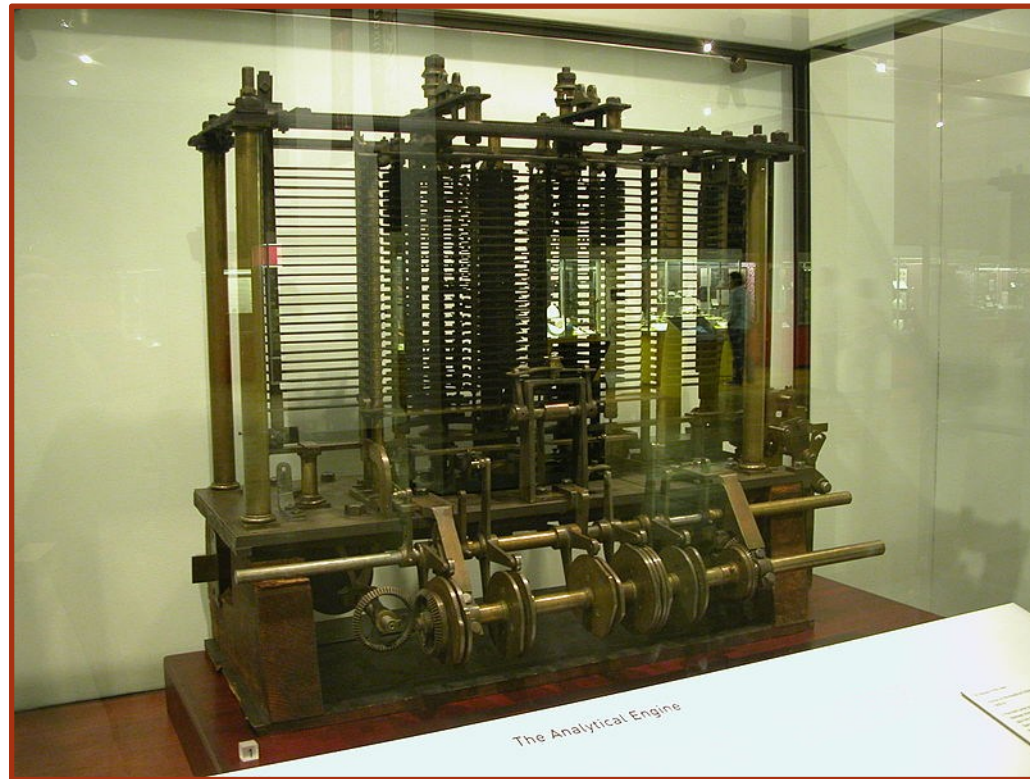
Il progetto della Macchina analitica di Babbage fu sviluppato dal matematico, filosofo e scienziato inglese Charles Babbage (1791–1871).

Pur essendo realizzata solo in parte per motivi politici e finanziari, rappresenta un importante passo nella storia dell'informatica.

Il moderno PC, infatti, presenta parecchie analogie con il progetto sviluppato da Babbage.



GLI ANTENATI DEL COMPUTER

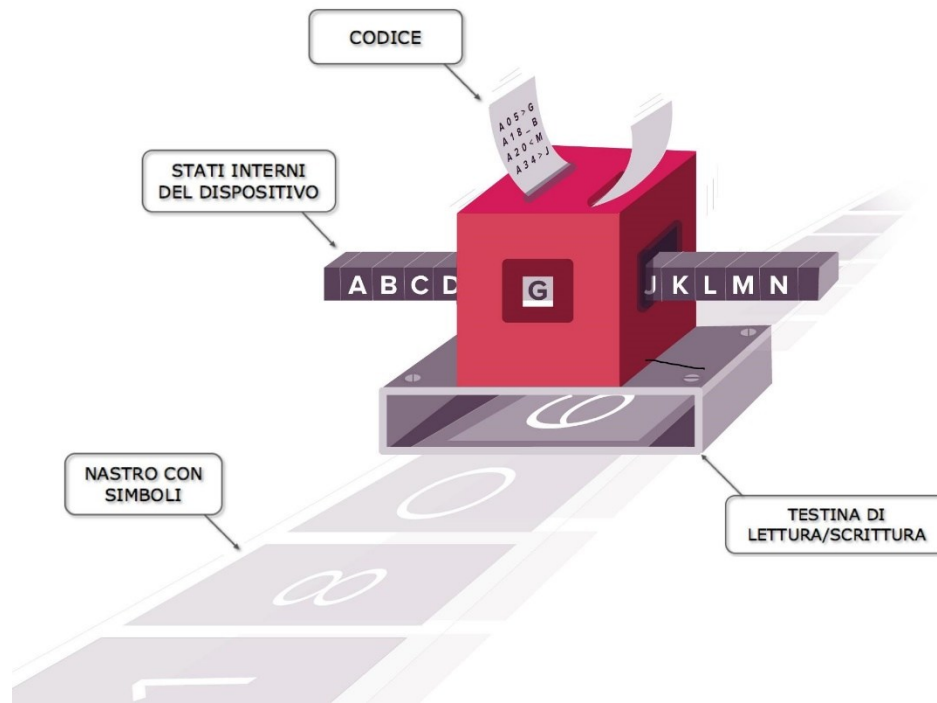


Modello di una parte dell'Analytical Engine di Babbage in mostra al Museo della scienza di Londra



GLI ANTENATI DEL COMPUTER

La macchina di Turing è una macchina ideale che manipola i dati contenuti su un nastro (memoria) di lunghezza infinita, con cui interagisce attraverso una testina di lettura/scrittura. Il dispositivo è caratterizzato da n (numero finito) stati interni.



GLI ANTENATI DEL COMPUTER

Ogni volta che la testina legge un dato dal nastro, la macchina di Turing può:

- cambiare stato interno (tra quelli ammessi);
- leggere/scrivere un nuovo dato da/sul nastro (memoria);
- spostare la testina, di una cella di memoria, avanti o indietro.

La macchina di Turing può essere programmata attraverso un codice che descrive le azioni che essa deve compiere.

In altre parole, è un modello astratto che definisce una macchina in grado di eseguire algoritmi e dotata di un nastro infinito (la memoria) su cui possono leggere e/o scrivere dei simboli (i dati).



GLI ANTENATI DEL COMPUTER

La macchina di Von Neumann è il modello architetturale secondo il quale è organizzata la maggior parte dei moderni elaboratori.

È dotata delle seguenti parti:

- processore;
- memorie;
- periferiche di Input;
- periferiche di Output;
- sistema di interconnessioni (bus).

Vedremo più avanti questo modello in dettaglio.



LA MACCHINA COMPUTER

In generale, un computer:

- elabora i dati, eseguendo operazioni logiche e aritmetiche;
- salva istruzioni e dati in memoria;
- interagisce con l'ambiente circostante prelevando informazioni da elaborare (input) o fornendo i risultati di tale elaborazione (output).

Un'altra caratteristica importante di un computer è la modularità e la scalabilità.

Modulare: utilizzo parti standard, pertanto in caso di guasti posso sostituirli con altri di pari funzionalità.

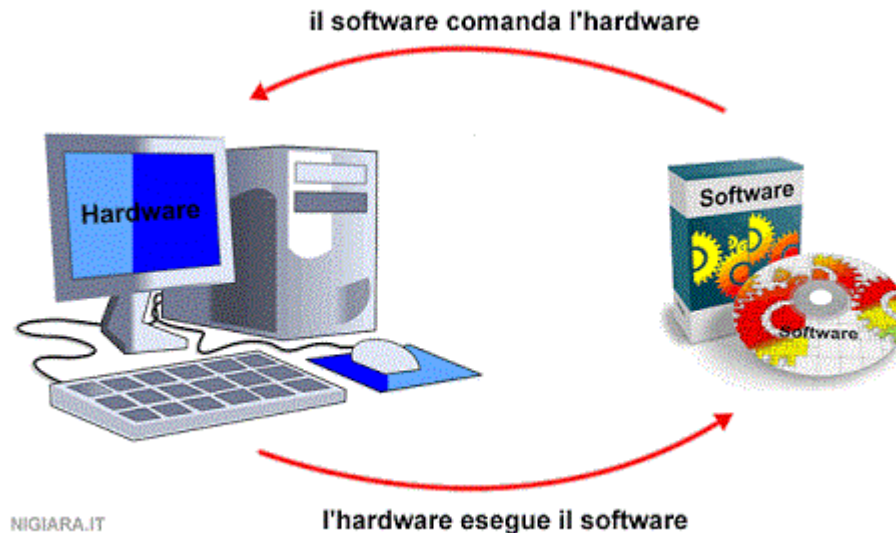
Scalabile: sostituisco la parti (i moduli) con altre parti compatibili ma aventi caratteristiche migliori.



HARDWARE VS SOFTWARE

L'*hardware* denota la struttura fisica del computer, costituita di norma da componenti elettronici che svolgono specifiche funzioni nel trattamento dell'informazione.

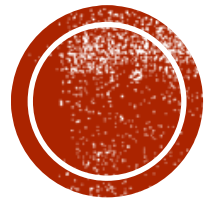
Il *software* indica l'insieme delle istruzioni che consentono alle varie parti hardware di svolgere i compiti per i quali sono stati progettati.



POSSIBILI SCENARI APPLICATIVI

- economico e commerciale;
- industriale;
- didattico e della formazione professionale;
- arte e spettacolo;
- ingegneristico;
- matematico e delle scienze;
- lavorativo e del tempo libero;
- medico;
- etc.





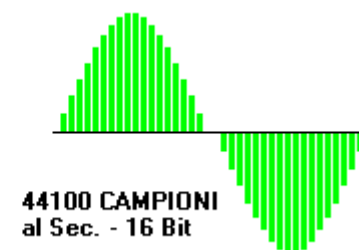
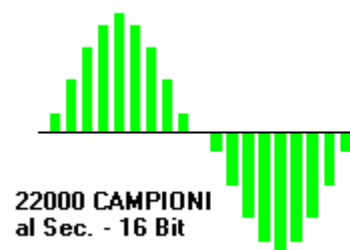
LA CODIFICA E LA RAPPRESENTAZIONE DELLE INFORMAZIONI

I SEGNALI PER COMUNICARE

Gli essere viventi ed il computer utilizzano modi tra loro differenti per comunicare:

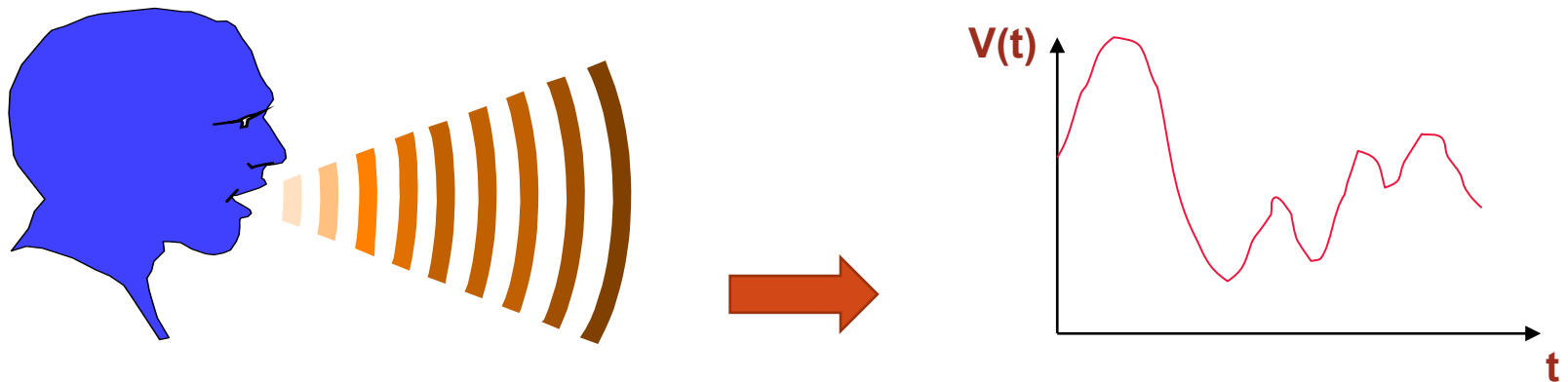
- segnali *analogici*;
- segnali *digitali*.

Ad un segnale analogico faremo corrispondere una grandezza continua, mentre ad un segnale digitale, una grandezza discreta.



INFORMAZIONE ANALOGICA

Ad esempio, la voce umana, i suoni presenti in natura o emessi da strumenti musicali, sono sistemi di comunicazione di tipo *analogico*, nei quali le grandezze fisiche che entrano in gioco sono funzioni continue del tempo.

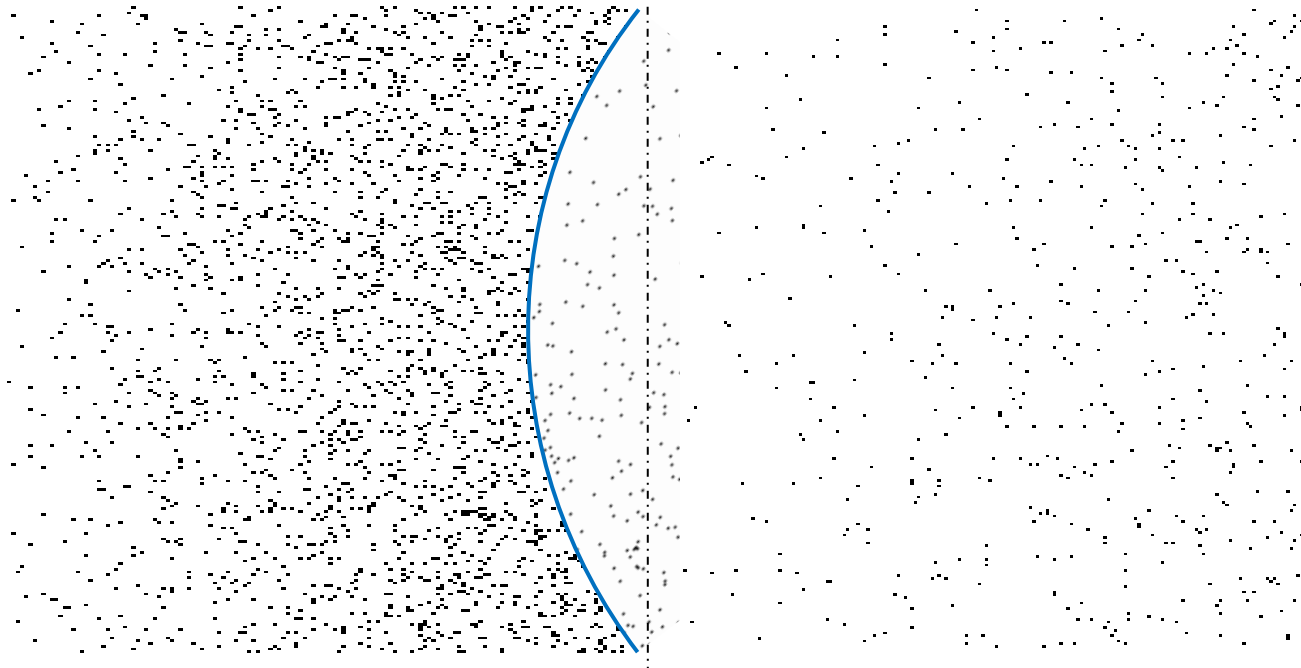


Vediamo in dettaglio cosa accade, ad esempio, quando con il microfono di un computer, durante una call, occorre trasmettere ad un computer remoto la nostra voce, o un qualsiasi suono.



INFORMAZIONE ANALOGICA

Consideriamo, ad esempio, la corda di uno strumento musicale che vibra in un mezzo quale l'aria.



INFORMAZIONE ANALOGICA

Inizialmente la densità delle molecole d'aria è costante, cioè in ogni unità di volume vi è lo stesso numero di molecole.

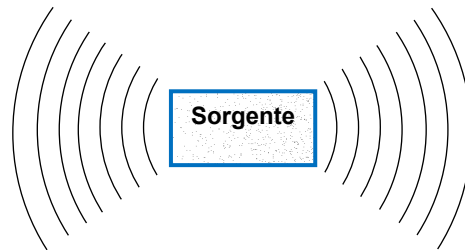
Quando viene sollecitata, la corda si sposta avanti e indietro, rispetto alla posizione di equilibrio indicata dalla linea tratteggiata in figura precedente e, durante questo spostamento, comprime le molecole d'aria da un lato e le espande dall'altro.

Ogni volta che il moto si inverte, le molecole che prima erano state compresse si espandono e viceversa.



INFORMAZIONE ANALOGICA

Le compressioni e le espansioni (cioè le perturbazioni dell'aria che inizialmente era in stato di quiete) si propagano poi con una certa velocità attraverso l'aria circostante in tutte le direzioni, dando luogo a onde sferiche.

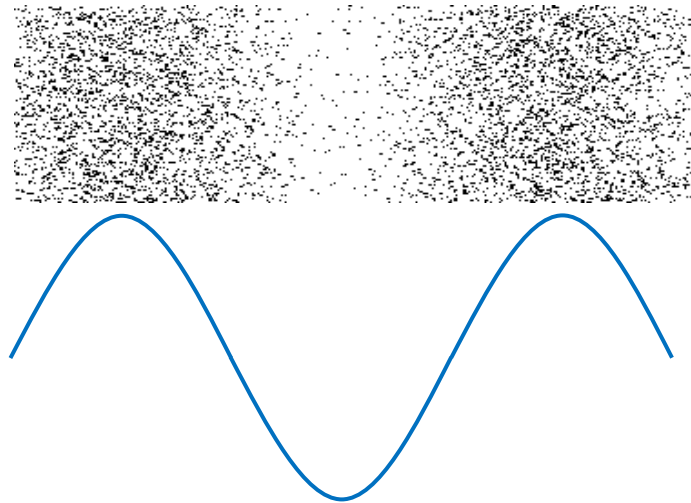


Queste onde arriveranno fino alla membrana del microfono che convertirà, attraverso un opportuno trasduttore, la variazione di pressione delle particelle di aria, in un segnale elettrico.



INFORMAZIONE ANALOGICA

Per quanto riguarda la forma d'onda del fenomeno appena illustrato, si noti che, ad esempio, può essere osservato **dal punto di vista del tempo**, misurando come varia il valore della pressione, in uno stesso punto (una molecola del mezzo), allo scorrere del tempo:

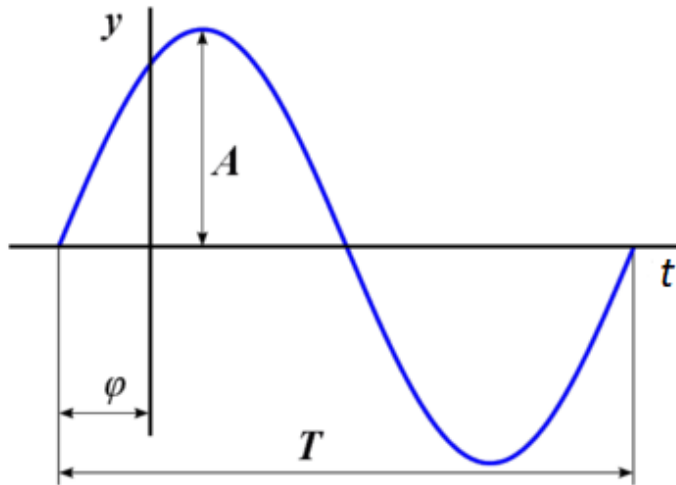


Osservando il fenomeno dal punto di vista dello spazio (propagazione dell'onda), comunque, la forma d'onda sarà sempre simile.



INFORMAZIONE ANALOGICA

Graficamente otteniamo, quindi, una sinusoide che è una grandezza continua:



$$y = A \sin(2\pi f t + \varphi)$$

Tuttavia, y assume infiniti valori, mentre in un computer, per quanto grande possa essere la memoria di cui è dotato, non potremo acquisire (e trasmettere) tutti valori.



INFORMAZIONE DIGITALE

La codifica dei segnali nei computer avviene in modo digitale, in quanto le grandezze fisiche sono rappresentate da coppie di stati discreti:



(0, 1) oppure (off, on) oppure (false, true)

Nei circuiti digitali il simbolo 0 (oppure *off*, oppure *false*) è associato ad un segnale a basso voltaggio (interruttore spento); di contro il simbolo 1 (oppure *on*, oppure *true*) è associato ad un segnale ad alto voltaggio (interruttore acceso).



DIGITALIZZAZIONE DEI SEGNALE

Qualsiasi segnale «reale», dunque, per poter essere elaborato attraverso un computer, deve essere convertito in un segnale digitale (e quindi in una sequenza finita di valori numerici).

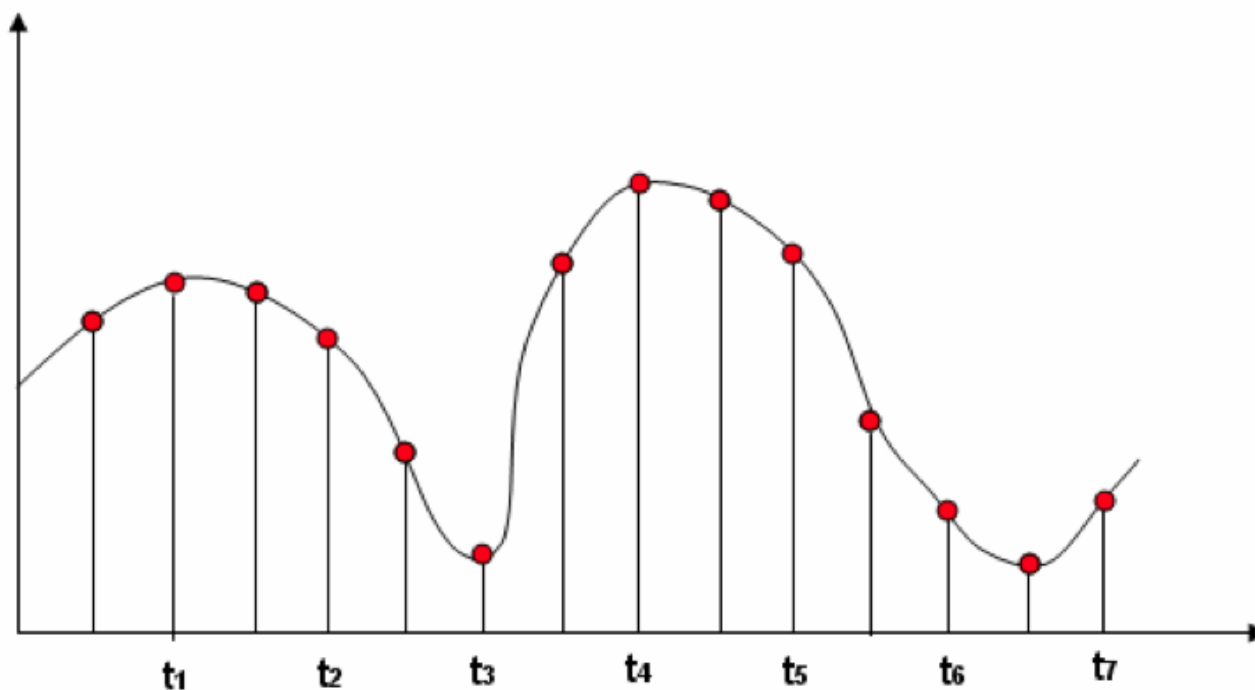
Nel farlo, devo attivare un procedimento noto come «campionamento», che mi permette di prelevare un certo numero di valori nell'unità di tempo (frequenza di campionamento).

Tale conversione, per quanto accurata, comporta sempre un certo grado di **approssimazione**, con conseguente perdita di informazione. Inoltre, gli strumenti di misura utilizzati, per quanto precisi, non mi permetteranno mai di disporre di tutti i valori, introducendo un **errore detto di quantizzazione**.



DIGITALIZZAZIONE DEI SEGNALE

Consideriamo ad esempio una grandezza continua e supponiamo di prelevare 7 campioni. Graficamente avremo:

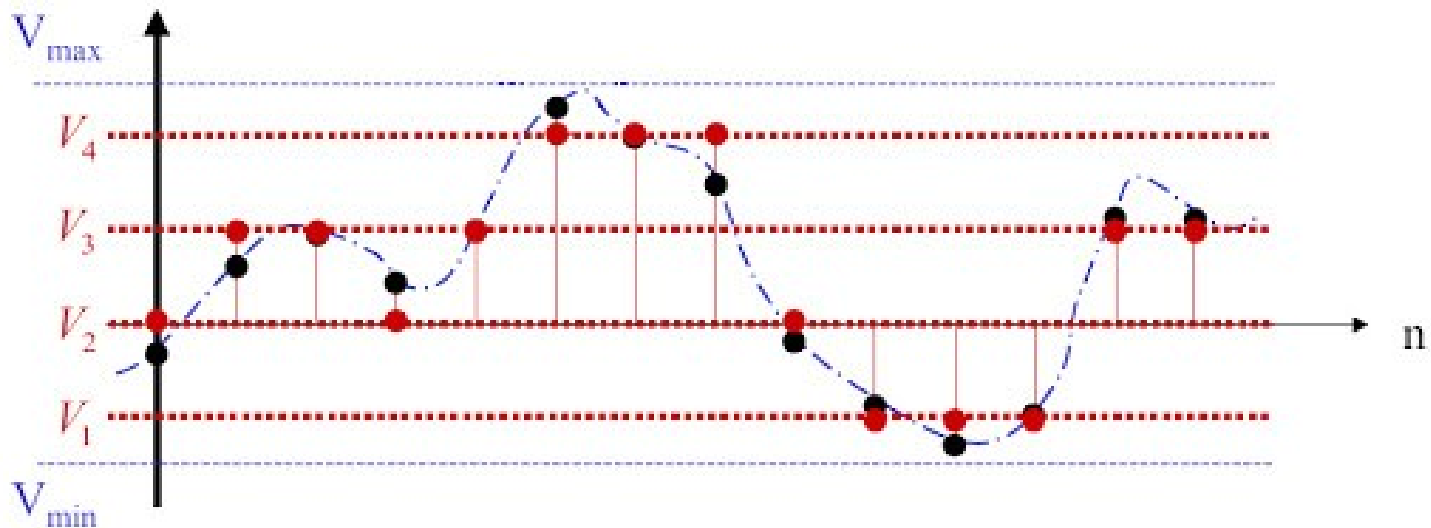


In questo caso, però, stiamo supponendo di poter misurare esattamente il valore di ciascun singolo campione.



DIGITALIZZAZIONE DEI SEGNALE

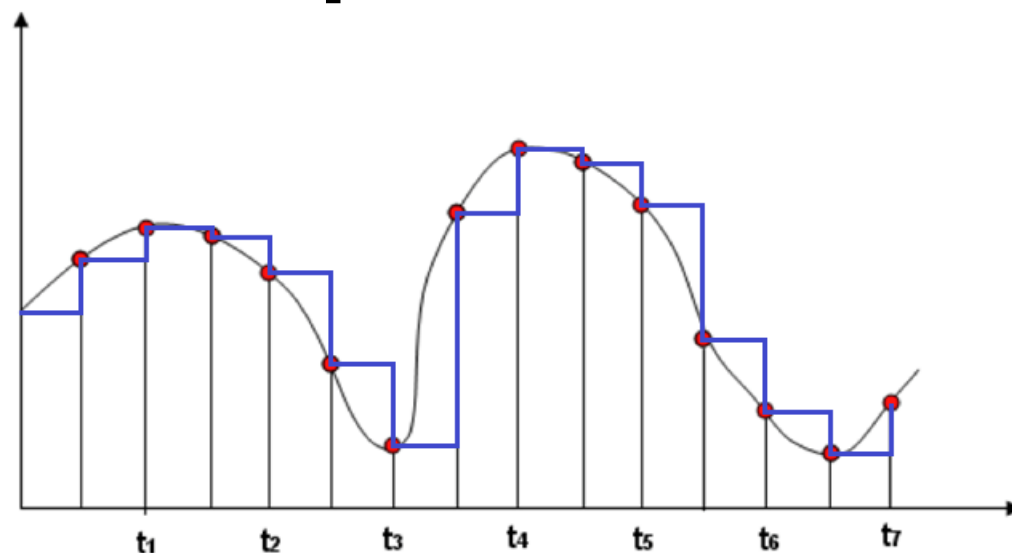
Nella realtà, come detto, dovremo fare i conti con l'errore di quantizzazione, in quanto ciascuna misura risentirà della scala di valori disponibile in ordinata tra V_{\min} e V_{\max} :



DIGITALIZZAZIONE DEI SEGNALE

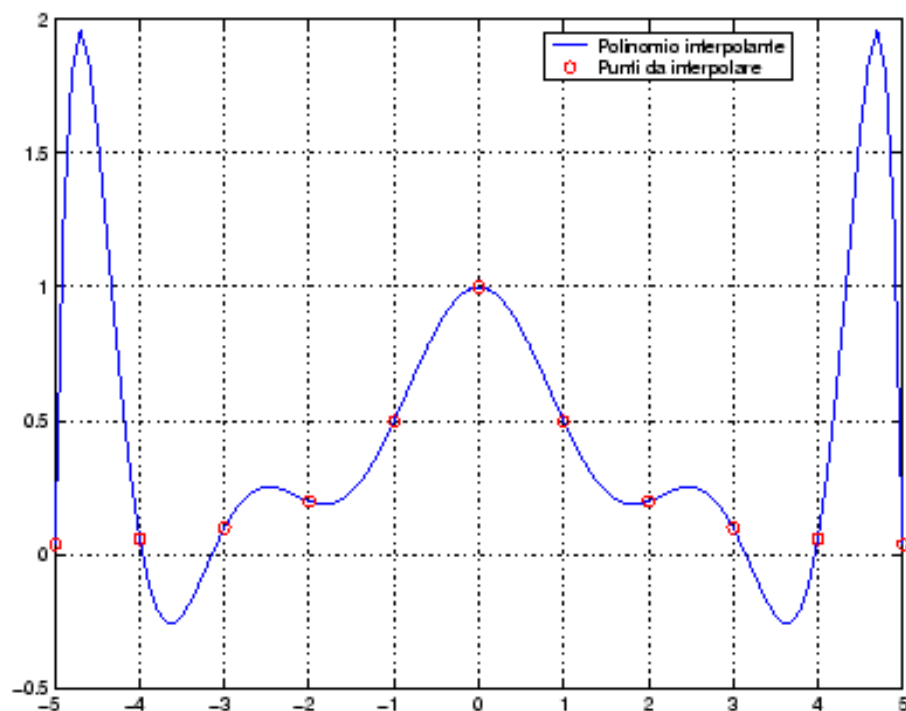
Una volta ricevuto il segnale digitale, il computer remoto dovrà ricostruire il segnale mancante tra i vari campioni. Per la ricostruzione, si possono utilizzare varie tecniche di interpolazione.

Ad esempio, quella del «sample and hold», che mantiene costante il valore fino al punto successivo:



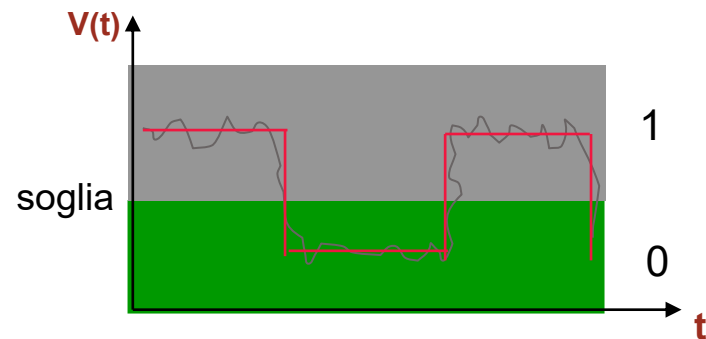
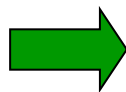
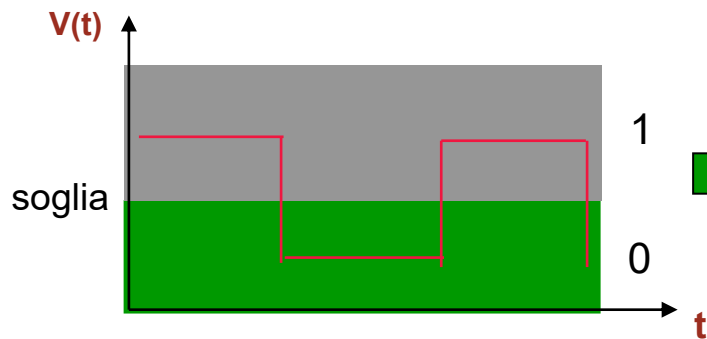
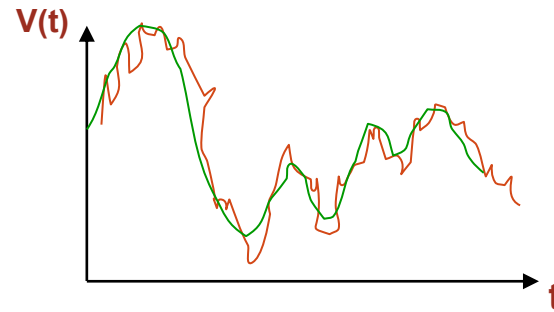
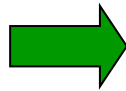
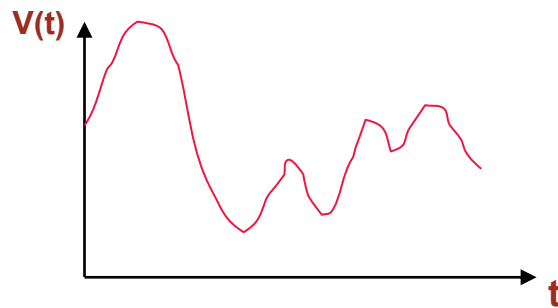
DIGITALIZZAZIONE DEI SEGNALE

O, per evitare un segnale troppo «spigoloso», attraverso interpolazione polinomiale (ad esempio interpolazione di Langrange):



PRECISIONE DEI SEGNALE

I segnali digitali presentano il vantaggio di essere meno affetti da disturbi di trasmissione (errori) rispetto a quelli analogici.



RAPPRESENTAZIONE DELLE INFORMAZIONI

In un calcolatore, tutte le informazioni sono rappresentate in forma binaria (o digitale), come detto, utilizzando due soli simboli. Di solito si adoperano i simboli 0 ed 1.

Con una cifra binaria si possono, quindi, rappresentare soltanto due informazioni.



RAPPRESENTAZIONE DELLE INFORMAZIONI

È semplice intuire, in ragione di quanto detto in precedenza, che tale scelta è legata alla necessità di rappresentare, ad esempio:

- due possibili stati di polarizzazione di una sostanza magnetizzabile;
- il passaggio (o non passaggio) di corrente attraverso un conduttore;
- Il passaggio (o non passaggio) della luce attraverso una fibra ottica.

Questo concetto verrà ripreso in seguito, nella macchina di Von Neumann, quando parleremo di memorie.



IL BIT

Il bit è *l'unità di misura dell'informazione* ed è definita come la quantità minima di informazione che serve a rappresentare due differenti stati: 0 oppure 1. Il nome proviene da **B**inary Digit.

Si utilizzano i seguenti multipli :

- Kilo (Kb), pari a $2^{10} \sim$ un migliaio (1024 bit);
- Mega (Mb) $2^{20} \sim$ un milione (1024 x 1024 bit);
- Giga (Gb) $2^{30} \sim$ un miliardo (1Mb x 1024 bit);
- Tera (Tb) $2^{40} \sim$ mille miliardi (1Gb x 1024 bit).



CODIFICA BINARIA

Per poter rappresentare un numero maggiore di dati, o informazioni, è necessario utilizzare sequenze di bit. Utilizzando stringhe lunghe due bit si possono rappresentare quattro informazioni diverse, vale a dire:

00 01 10 11

In generale, con n bit si possono avere 2^n stringhe (dette anche parole in linguaggio macchina) differenti.

Il processo che fa corrispondere ad una informazione una configurazione di bit, prende il nome di *codifica dell'informazione*.



SEQUENZE DI BIT

Numero di bit nella sequenza	Informazioni rappresentabili
2	4
3	8
4	16
5	32
6	64
7	128
8	256



SET DI CARATTERI

Nella comunicazione scritta, di norma, è necessario disporre dei seguenti caratteri:

- 52 lettere alfabetiche maiuscole e minuscole;
- 10 cifre (0, 1, 2, ..., 9);
- Segni di punteggiatura (, . ; : ! " ' ^ \ ...);
- Segni matematici (+, -, ×, :, ±, {, [, >, ...);
- Caratteri nazionali (à, è, ì, ò, ù, ç, ñ, ö, ...);
- Altri segni grafici (©, ←, ↑, @, €, ...).



CODICE

Si pone quindi la necessità di codificare almeno 220 caratteri utilizzando sequenze di bit.

La sequenza di bit necessaria a rappresentare 220 simboli deve essere composta da 8 bit (con 8 bit, infatti, dispongo di $2^8=256$ stringhe differenti).

La corrispondenza tra sequenze (o stringhe) di bit e simboli prende il nome di *codice*.



IL BYTE

Un gruppo di 8 bit viene denominato Byte. Un byte:

- corrisponde ad un carattere;
- rappresenta l'unità di misura della capacità di memoria.

Si utilizzano i seguenti multipli del Byte:

- Kilo (KB) 2^{10} ~ un migliaio (1024 byte)
- Mega (MB) 2^{20} ~ un milione (1024 x 1024 byte)
- Giga (GB) 2^{30} ~ un miliardo (1MB x 1024 byte)
- Tera (TB) 2^{40} ~ mille miliardi (1GB x 1024 byte)



CODIFICA SET DI CARATTERI

Codifiche standard:

- **ASCII**, 8 bit per carattere, rappresenta 256 caratteri.
- **UNICODE**, 16 bit per carattere (estende il codice ASCII con i caratteri etnici).

Codifiche proprietarie:

- **MSWindows**, 16 bit per carattere (molto simile ad UNICODE).



CODICE ASCII

Acronimo di:

American
Standard
Code for
Information
Interchange

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	\$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	
111 1101	175	125	7D	}
111 1110	176	126	7E	~



CODICE ASCII: ALTRO ESEMPIO

La parola *Computer* in codice ASCII diventa, dunque:

C	=	01000011
o	=	01101111
m	=	01101101
p	=	01110000
u	=	01110101
t	=	01110100
e	=	01100101
r	=	01110010

01000011 01101111 01101101 11100000 01110101 01110100 01100101 01110010



CODICE ASCII

Data una sequenza di bit, dividendola in gruppi di byte è possibile risalire ai caratteri originali:

01000001 01101101 01101111 01110010 01100101 00100001

01000001 01101101 01101111 01110010 01100101 00100001

A

m

o

r

e

!



CODICE ASCII

Con il codice ASCII è possibile rappresentare i numeri come sequenza di caratteri. Ad esempio il numero 234 sarà rappresentato come:

00110010 00110011 00110100

2

3

4

Con questo tipo di rappresentazione, però, non è possibile effettuare operazioni aritmetiche.



IL SISTEMA DECIMALE

Il sistema di numerazione decimale è un sistema *posizionale*. Questo significa che ogni cifra del numero, assume un valore differente in funzione della posizione. Consideriamo il numero seguente in *notazione compatta*:

221

corrisponde a:

$$2 \times 100 + 2 \times 10 + 1 \times 1$$

Che, in *notazione esplicita*, si scriverà:

$$2 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$



IL SISTEMA DECIMALE

Ogni numero si esprime come la somma dei prodotti di ciascuna cifra per la base, elevata all'esponente dato dalla posizione della cifra:

$$221 = 2 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

È quindi basato sulle potenze del 10 ed utilizza, come detto, una notazione posizionale basata sui 10 simboli:

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



LA NOTAZIONE POSIZIONALE

La notazione posizionale può essere usata con qualunque base, creando così differenti sistemi di numerazione. Per ogni base di numerazione si utilizza un numero di cifre uguale alla base.

In informatica si utilizzano prevalentemente i sistemi di numerazione:

- binaria,
- ottale,
- esadecimale.

Il sistema di numerazione romano non è posizionale:

XIII vs. CXII



IL SISTEMA BINARIO

Il sistema di numerazione binario utilizza una notazione posizionale basata sulle 2 cifre:

$$\{0, 1\}$$

e sulle potenze di 2. Il numero binario 1001 può essere rappresentato esplicitamente come:

$$1001_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{10}$$



IL SISTEMA OTTALE

Il sistema di numerazione ottale utilizza una notazione posizionale basata sulle 8 cifre:

$$\{0, 1, 2, 3, 4, 5, 6, 7\}$$

e sulle potenze di 8. Il numero ottale 534 può essere rappresentato, esplicitamente, come:

$$534_8 = 5 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 348_{10}$$



IL SISTEMA ESADECIMALE

La numerazione esadecimale utilizza una notazione posizionale basata sulle 16 cifre

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

e sulle potenze di 16. Si noti che $A=10$, $B=11$, $C=12$, $D=13$, $E=14$ ed $F=15$. Il numero esadecimale $B7FC_{16}$ può essere rappresentato esplicitamente come:

$$(11) \times 16^3 + 7 \times 16^2 + (15) \times 16^1 + (12) \times 16^0 = 47100_{10}$$



CONVERSIONE DA BASE B A BASE 10

Per convertire un numero da una qualunque base b alla base 10, è sufficiente rappresentarlo esplicitamente:

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_{10}$$

$$710_8 = 7 \cdot 8^2 + 1 \cdot 8^1 + 0 \cdot 8^0 = 456_{10}$$

$$A51_{16} = (10) \cdot 16^2 + 5 \cdot 16^1 + 1 \cdot 16^0 = 2641_{10}$$

....

$$C_{n-1}C_{n-2}\dots C_1 C_0 = C_{n-1}b^{n-1} + C_{n-2}b^{n-2} + \dots + C_1b^1 C_0b^0$$

dove ogni cifra $C_i \in [0, b[$ per $i = 0, 1, \dots, n-1$ (ad esempio, per $b=8$ ogni cifra, in decimale, assume valore appartenente a $[0, 7]$).



CONVERSIONE DA BASE 10 A BASE N

Per convertire un numero decimale ad una base n qualsiasi, occorre trovare tutti i resti delle successive divisioni del numero per la base n .

Proviamo, ad esempio, a trovare il valore binario del numero 210. Come mostrato nella slide successiva, basterà dividere 210 per la base 2, ripetutamente, fino quando non si ottiene zero come risultato e leggere i resti dall'ultimo verso il primo.



CONVERSIONE DA BASE 10 A BASE 2

210	2	resto	0
105	2		1
52	2		0
26	2		0
13	2		1
6	2		0
3	2		1
1	2		1
0			

Leggendo la sequenza
dei resti dal basso
verso l'alto, si ottiene il
numero:

11010010₂



VERIFICA DI CORRETTEZZA

Per una verifica di correttezza basta riconvertire il risultato alla base 10:

$$\begin{aligned} 11010010_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + \\ &+ 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 210_{10} \end{aligned}$$



COSTRUZIONE DEI NUMERI BINARI

Per poter costruire la successione dei numeri binari, si può seguire il seguente schema:

0	0	0	0	=	0
0	0	0	1	=	1
0	0	1	0	=	2
0	0	1	1	=	3
0	1	0	0	=	4
0	1	0	1	=	5
0	1	1	0	=	6
0	1	1	1	=	7



RAPPRESENTAZIONE DEI NUMERI

All'interno dei computer, a causa di vincoli tecnologici, per rappresentare qualsiasi tipo di numero, si utilizza sempre un numero fisso di cifre binarie.

I valori più utilizzati sono:

- 16 bit (2 byte)
- 32 bit (4 byte)

In alcuni casi si può arrivare anche a 64 bit (8 byte) o più a seconda del tipo di processore (vedremo in dettaglio tutto questo, quando parleremo della macchina di Von Neumann).



RAPPRESENTAZIONE DEI NUMERI

Tutti i numeri vengono distinti in tre categorie:

- Interi senza segno (interi positivi).
- Interi con segno (interi positivi e negativi).
- Reali (numeri positivi e negativi con virgola).

Ogni categoria viene rappresentata in modo differente.



RAPPRESENTAZIONE CON NUMERO FISSO DI CIFRE

Per comprendere il meccanismo alla base della rappresentazione con un numero fisso di cifre partiamo da un interrogativo:

Qual è il numero più grande rappresentabile con 4 cifre?

- In base 10, avremo 9999
- In base 2 avremo $1111 = 15_{10}$
- In base 8 avremo $7777 = 4095_{10}$
- In base 16 avremo $FFFF = 65535_{10}$



RAPPRESENTAZIONE CON NUMERO FISSO DI CIFRE

In generale, con n cifre vale la relazione seguente:

$$b^n - 1$$

essendo b la base del sistema di numerazione. Infatti:

- | | | |
|-------------------------|--------------|--------------|
| ▪ In base 10, avremo | 9999 | $= 10^4 - 1$ |
| ▪ In base 2 avremo 1111 | 15_{10} | $= 2^4 - 1$ |
| ▪ In base 8 avremo 7777 | 4095_{10} | $= 8^4 - 1$ |
| ▪ n base 16 avremo FFFF | 65535_{10} | $= 16^4 - 1$ |



RAPPRESENTAZIONE CON NUMERO FISSO DI CIFRE

Vale quindi la seguente regola:

Nella base di numerazione **b** , disponendo di **n** cifre, si possono rappresentare soltanto i numeri da

$$0 \text{ a } b^n - 1$$



RAPPRESENTAZIONE DEI NUMERI INTERI SENZA SEGNO

In un elaboratore, tenuto conto di quanto detto, considerato che nelle rappresentazioni si adotta il sistema binario e che le parole sono lunghe 16 o 32 bit, si ha:

- Nella rappresentazione a 16 bit i possibili valori saranno compresi tra 0 e 65.535;
- Nella rappresentazione a 32 bit i possibili valori saranno compresi tra 0 e 4.294.967.295



RAPPRESENTAZIONE DEI NUMERI INTERI CON SEGNO

Per rappresentare i numeri con il loro segno (interi positivi e negativi) esistono due possibili modi:

- Rappresentazione in modulo e segno;
- Metodo del complemento a due.

Nel primo metodo, dati n bit, il bit a sinistra si riserva al segno, lasciando gli altri $n - 1$ alla rappresentazione del numero. Di solito si considera 0 per il segno più (+) e 1 per il meno (-):

$$0000\ 0101_2 = +5_{10}$$

$$1000\ 0101_2 = -5_{10}$$



RAPPRESENTAZIONE DEI NUMERI INTERI CON SEGNO

Questo metodo, anche se molto semplice, presenta l'inconveniente che esistono due zeri:

$$1\ 0\ 0\ 0\ 0\ 0\ 1\ 1 = -3$$

$$1\ 0\ 0\ 0\ 0\ 0\ 1\ 0 = -2$$

$$1\ 0\ 0\ 0\ 0\ 0\ 0\ 1 = -1$$

$$1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = -0$$

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = +0$$

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 = +1$$

$$0\ 0\ 0\ 0\ 0\ 0\ 1\ 0 = +2$$

$$0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 = +3$$



RAPPRESENTAZIONE DEI NUMERI INTERI CON SEGNO

Utilizzando n bit e riservandone uno al segno, l'applicazione della formula precedente porterà alla rappresentazione del seguente intervallo:

$$\text{da } -(2^{n-1} - 1) \text{ a } 2^{n-1} - 1$$

dove, come detto, n vale o 16, permettendo quindi la rappresentazione dei numeri da -32.767 a +32.767, o 32 con rappresentazione dei numeri da -2.147.483.647 a +2.147.483.647



RAPPRESENTAZIONE DEI NUMERI INTERI CON SEGNO

Il secondo metodo è quello del complemento a 2:

Dato un numero composto da n bit, la rappresentazione in complemento a due si ottiene invertendo gli 1 in 0 e gli 0 in 1 (complemento a 1) e poi sommando 1 al risultato ottenuto.

In questo caso, ad ogni numero (senza segno) vengono aggiunti degli zeri a sinistra, fino a raggiungere un byte (o multipli di byte). Se il numero è positivo, si lascia invariato, se è negativo si converte in complemento a due (ottenendo sempre il primo bit = 1, come in precedenza).

È ovvio che 00000000 e 10000000 hanno significati differenti.



OPERAZIONI CON I BINARI

Per procedere con la somma tra binari, come operazioni in colonna, occorre tenere conto delle seguenti indicazioni:

$$\begin{array}{r} 0 + \\ 0 = \\ \hline 0 \end{array} \quad \begin{array}{r} 0 + \\ 1 = \\ \hline 1 \end{array} \quad \begin{array}{r} 1 + \\ 0 = \\ \hline 1 \end{array} \quad \begin{array}{r} 1 + \\ 1 = \\ \hline 0 \end{array}$$

1 ← riporto

Dal momento che un computer è in grado di compiere solo somme e confronti, ci serve una procedura che consenta di convertire ogni operazione, in somma (o sequenze di somme).

Vediamo come farlo in decimale (il procedimento può essere esteso, poi, ai binari con lo stesso ragionamento).



OPERAZIONI CON I BINARI

SOTTRAZIONI

La sottrazione può essere convertita in somma nel modo seguente:

$$25 - 14 = 25 + (-14)$$

MOLTIPLICAZIONI

La moltiplicazione può essere convertita sommando un fattore, per sé stesso, tante volte quanto è il valore assunto dall'altro fattore (in genere si sceglie il minimo tra i due):

$$25 \times 3 = 25 + 25 + 25$$

DIVISIONE

La divisione può essere ottenuta sottraendo il divisore al dividendo e iterando il procedimento fino a quando il risultato non diventi minore dello stesso divisore:

$$10 : 3 \rightarrow 10 - 3 = 7 - 3 = 4 - 3 = 1$$

Il numero di sottrazioni effettuate corrisponde al risultato mentre il valore finale ottenuto, indica il resto della divisione.



L'OVERFLOW

Come visto, per questioni tecnologiche tutti i computer, senza alcuna eccezione, trattano i numeri sempre con una quantità fissa di cifre binarie (ad esempio 16, 32 o più).

Quando l'elaboratore esegue un'operazione il cui risultato eccede il numero di cifre permesso, la computazione si arresta immediatamente e viene segnalato l'errore di OVERFLOW.



L'OVERFLOW

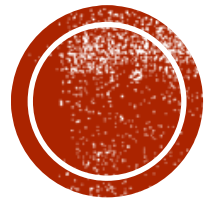
Ad esempio, se la rappresentazione è a 32 bit senza segno e si vuole eseguire la seguente operazione:

$$\begin{array}{r} 3.000.000.000 \times \\ 2.000.000.000 = \\ \hline \end{array}$$

***** Errore di **OVERFLOW**

In questo caso l'elaborazione si arresta.





ALGEBRA DI BOOLE

INTRODUZIONE

L'algebra booleana deve il proprio nome al matematico inglese George Boole che la formalizzò nel 1854.

Fu sviluppata per poter trattare qualsiasi tipo di proposizione, ovvero di espressione le cui variabili possono assumere solo due valori:

True/Vero

False/Falso

Deve la sua importanza al fatto che i circuiti elettronici digitali possono essere rappresentati con una funzione booleana e viceversa.



INTRODUZIONE

Un circuito elettronico digitale, infatti, è un sistema caratterizzato dalla presenza di segnali che possono assumere due soli valori di tensione:

0V

+5V

(anche se, in alcune tipologie di circuiti, si preferisce parlare di Low, o L, e High , o H, al fine di considerare valori differenti).

In genere si fa corrispondere il simbolo 0 al valore di tensione 0V (o Low, o L) ed il simbolo 1 al valore di tensione +5V (o High, o H).

L'algebra di Boole è diventata anche l'elemento fondamentale per la logica formale che costituisce la base per il ragionamento scientifico.



INTRODUZIONE

Secondo Boole, dunque, il ragionamento è basato su proposizioni che possono assumere valore Vero o Falso. Ad esempio:

Oggi piove!

è una proposizione che può essere vera o falsa. Le proposizioni possono anche essere combinate tra loro. Ad esempio, consideriamo:

«Piove»

«Esco»

«Apro l'ombrello»

Tutte e tre possono assumere valore Vero o Falso. Secondo Boole, queste possono anche essere combinate tra loro. Ad esempio:

Se «Piove» e «Esco» allora «Apro l'ombrello»

In questo caso, se sono vere le prime due, allora è vera anche l'ultima.



INTRODUZIONE

L'algebra di Boole, dunque, introduce i concetti di **Variabili** e **Funzioni**.

Una *variabile booleana* viene rappresentata, solitamente, attraverso una lettera maiuscola (A, B, C,...) e può assumere, come le proposizioni, solo valore *Vero* o *Falso*.

Una *funzione booleana*, invece, è una legge che ammette in input n variabili booleane e che fornisce in output un valore che può essere, ancora una volta, solo *Vero* o *Falso*.

Il comportamento di una funzione booleana può essere descritto attraverso la **tavola di verità**, ovvero una tabella avente 2^n righe (se n è il numero di variabili in input) che ci permettono di verificare per quali combinazioni di ingresso è vera la funzione.



INTRODUZIONE

ESEMPIO

Consideriamo la funzione:

$$Y = f(A, B, C)$$

La tabella a lato indica, in ogni riga, il valore assunto dalla funzione Y . Il valore assunto può essere indicato con T (per vero) e F (per falso) oppure con 1 (per vero) e 0 (per falso).

Noi faremo uso di questa seconda modalità.

A	B	C	Y
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	F
T	F	F	F
T	F	T	F
T	T	F	T
T	T	T	F



OPERATORI

L'algebra di Boole si basa su tre operatori: AND, OR e NOT (a cui corrispondono altrettante **porte logiche fondamentali**, usate negli schemi dei circuiti elettronici digitali).

AND

Prodotto logico (A and B)



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

In genere si usa l'operatore " \cdot " che viene scritto espressamente, oppure omissso:

$$Y = A \cdot B$$

oppure

$$Y = AB$$



OPERATORI

OR

Somma logica (A or B)



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

In genere si usa l'operatore " + " come nell'espressione:

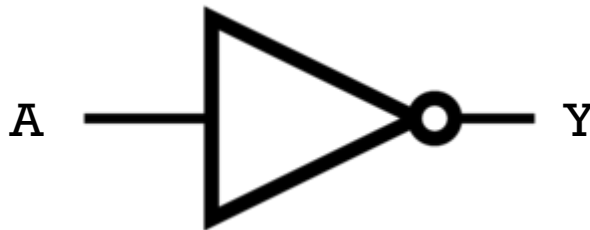
$$Y = A + B$$



OPERATORI

NOT

Negazione logica (A negato)



A	Y
0	1
1	0

In genere si usa l'operatore " $\bar{}$ " (trattino in alto alla variabile cui fa riferimento) come nell'espressione:

$$Y = \bar{A}$$

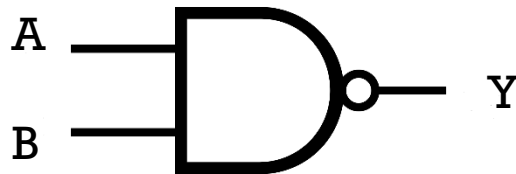


OPERATORI

Qualsiasi proposizione può essere scritta utilizzando la combinazione delle porte logiche fondamentali. Per semplificare gli schemi digitali o le proposizioni, però, spesso si fa uso delle seguenti **porte logiche derivate**.

NAND

Prodotto logico negato (A nand B)



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

In genere si usa l'operatore " $\overline{\cdot}$ " che viene scritto con il puntino, oppure senza, lasciando solo il trattino in alto:

$$Y = \overline{A \cdot B}$$

oppure

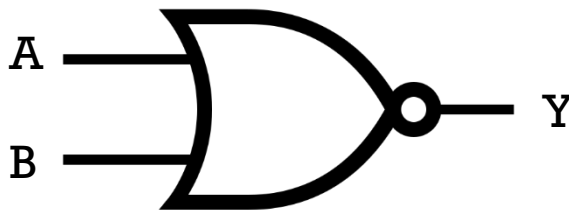
$$Y = \overline{AB}$$



OPERATORI

NOR

Somma logica negata (A nor B)



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

In genere si usa l'operatore " $\overline{+}$ " come nell'espressione:

$$Y = \overline{A + B}$$



OPERATORI

XOR

OR esclusivo (A or esclusivo B o A diverso da B)



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

In genere si usa l'operatore \oplus come nell'espressione:

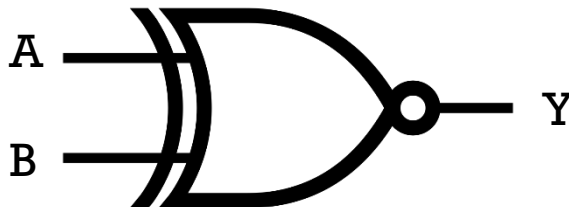
$$Y = A \oplus B$$



OPERATORI

XNOR

NOR esclusivo (A nor esclusivo B)



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

In genere si usa l'operatore \oplus come nell'espressione:

$$Y = \overline{A \oplus B}$$



FUNZIONI

Abbiamo visto che una funzione booleana (o funzione logica) è una legge F che presa una $n - upla$ di variabili booleane (x_1, x_2, \dots, x_n) fa corrispondere ad essa il valore:

$$F(x_1, x_2, \dots, x_n) = \begin{cases} true \\ 0 \\ false \end{cases}$$

Inoltre, il suo comportamento può essere descritto attraverso una **tavola di verità** o attraverso un **circuito digitale**.

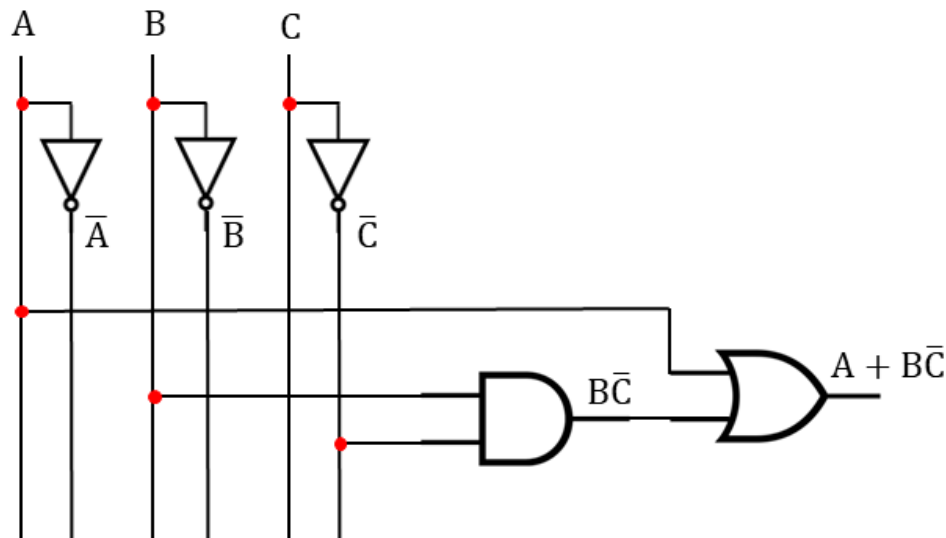


ESEMPIO

Consideriamo, ad esempio, la seguente funzione booleana:

$$F = A + B\bar{C}$$

La *tavola di verità* ed il *circuito digitale combinatorio* corrispondenti saranno:



A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



MINTERMINI E MAXTERMINI

Data una tavola di verità e presa una riga, si definisce *mintermine* quello ottenuto moltiplicando tra loro le variabili booleane di input, presenti in tale riga, invariate, se assumono valore 1 o negate se assumono valore 0.

Si definisce, invece, *maxtermine* quello ottenuto sommando tra loro le variabili booleane di input, presenti sempre nella riga considerata, questa volta prese invariate, se assumono valore 0, o negate se assumono valore 1.

In una tavola di verità con n variabili, come detto, si possono avere 2^n input, pertanto si avranno altrettanti mintermini e maxtermini.



MINTERMINI E MAXTERMINI

Se consideriamo nuovamente la seguente tavola di verità:

in essa, presa la terza riga (quella con 010 in input), il mintermine corrispondente sarà:

$$\bar{A}B\bar{C}$$

mentre il maxtermine corrispondente sarà:

$$A + \bar{B} + C$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



FORME CANONICHE

Data una generica funzione booleana e considerata la sua rappresentazione attraverso una tavola di verità, essa può essere espressa nella *prima forma canonica* (o forma disgiuntiva):

- considerando solo le righe in cui essa assume in uscita il valore 1;
- costruendo i mintermini corrispondenti;
- sommando logicamente (OR) tutti i mintermini ottenuti al punto precedente.



FORME CANONICHE

Essa può essere, invece, espressa nella *seconda forma canonica* (o forma congiuntiva):

- considerando solo le righe in cui essa assume in uscita il valore 0;
- costruendo i maxtermini corrispondenti;
- moltiplicando logicamente (AND) tutti i maxtermini ottenuti al punto precedente.



ESEMPIO

Si consideri sempre la tavola di verità vista in precedenza:

I termini che portano al valore $F = 1$ sono:

010, 100, 101, 110 e 111

Considerando gli input in cui essa vale 1, costruendo i mintermini corrispondenti e sommando logicamente tra loro tali termini, si ottiene:

$$F = \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

che è la prima forma canonica di F (si provi a costruire la seconda forma come esercizio).

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



FUNZIONI

Si noti, però, come la rappresentazione di F alla quale siamo appena giunti con questo procedimento, ovvero:

$$F = \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

sia di gran lunga più complessa rispetto a quella dalla quale siamo partiti nella slide in cui abbiamo costruito anche il circuito digitale equivalente, ovvero:

$$F = A + B\bar{C}$$

Ciò è dovuto al fatto che, per ciascuna funzione booleana **esistono più rappresentazioni equivalenti**. Cerchiamo di capire, allora, in che modo è possibile determinare la forma ridotta, trasformando la prima nella seconda.



PROPRIETÀ DELL'ALGEBRA DI BOOLE

Nell'algebra booleana valgono diversi assiomi, utilizzando i quali è possibile manipolare le espressioni logiche al fine, ad esempio, di semplificarle:

$$A + 0 = A$$

proprietà di identità

$$A + 1 = 1$$

proprietà di assorbimento

$$A + \bar{A} = 1$$

proprietà dell'inverso

$$A + A = A$$

proprietà dell'idempotenza

$$\bar{\bar{A}} = A$$

proprietà della doppia negazione



PROPRIETÀ DELL'ALGEBRA DI BOOLE

Principio di dualità

Una proprietà è valida se e solo se è valida la sua duale, ovvero quella che si ottiene scambiando tra loro somma con prodotto e 1 con 0.

Applicando tale principio, le proprietà precedenti possono essere estese al prodotto logico:

$$A \cdot 1 = A$$

proprietà di identità

$$A \cdot 0 = 0$$

proprietà di assorbimento

$$A \cdot \bar{A} = 0$$

proprietà dell'inverso

$$A \cdot A = A$$

proprietà dell'idempotenza



PROPRIETÀ DELL'ALGEBRA DI BOOLE

Valgono, inoltre, le seguenti proprietà e teoremi:

Proprietà commutativa

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Proprietà associativa

$$A + (B + C) = (A + B) + C$$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

Proprietà distributiva

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$



PROPRIETÀ DELL'ALGEBRA DI BOOLE

O ancora...

Teoremi di De Morgan

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

I° teorema dell'assorbimento

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

II° teorema dell'assorbimento

$$A + \bar{A} \cdot B = A + B$$

$$A \cdot (\bar{A} + B) = A \cdot B$$



ESEMPIO

Utilizzando le proprietà appena illustrate, vediamo in che modo è possibile semplificare la precedente funzione:

$$F = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

Modificando l'ordine e applicando la proprietà dell'idempotenza $A + A = A$ al penultimo termine possiamo scrivere:

$$F = A\bar{B}\bar{C} + A\bar{B}C + ABC + AB\bar{C} + AB\bar{C} + \bar{A}B\bar{C}$$

Tra i primi due termini possiamo mettere in evidenza $A\bar{B}$, tra terzo e quarto AB e tra quinti e sesto $B\bar{C}$, ottenendo:

$$F = A\bar{B} + AB + B\bar{C} = A(B + \bar{B}) + B\bar{C} = A + B\bar{C}$$

avendo messo in evidenza A tra i primi due termini e semplificato.



MINIMIZZAZIONE DI FUNZIONI BOOLEANE

Riscrivere una funzione booleana nella forma minima equivalente, significa semplificare (e quindi ridurre il costo di realizzazione) il circuito digitale con il quale implementarla, ovvero ridurre il numero di componenti elettronici da utilizzare, ma anche il livello (ovvero il numero massimo di componenti da attraversare tra input e output, fatta eccezione per le porte NOT).

Procedere alla semplificazione ricorrendo alle proprietà nel modo visto nell'ultimo esempio, però, non è sempre una operazione semplice da compiere.

Vediamo adesso un altro metodo che fa uso di una differente forma di rappresentazione di una funzione booleana, ovvero delle *Mappe di Karnaugh*.



MINIMIZZAZIONE DI FUNZIONI BOOLEANE

Una Mappa di Karnaugh può essere considerata come una variante delle tavole di verità. Vediamo come costruirle a 2, a 3 o a 4 variabili booleane:

A \ B	0	1
0		
1		

A \ BC	00	01	11	10
0				
1				

CD \ AB	00	01	11	10
00				
01				
11				
10				

È importante notare che le mappe sono costruire in modo tale che, nello spostamento tra righe o colonne adiacenti, solo una variabile per volta cambia il proprio valore.



MINIMIZZAZIONE DI FUNZIONI BOOLEANE

Consideriamo ancora una volta la seguente tavola di verità:

La mappa di karnaugh associata ad una funzione, si costruisce riportando il valore 1 in corrispondenza dei mintermini che assumono tale valore (gli 0 si possono anche omettere):

A \ BC	BC			
	00	01	11	10
0				1
1	1	1	1	1

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



MINIMIZZAZIONE DI FUNZIONI BOOLEANE

Disegnata la mappa di Karnaugh, per semplificare occorre:

- raggruppare i valori 1 adiacenti (considerando sempre un numero di valori che sia potenza di 2, ovvero 2, 4, 8,...);
- considerare solo i gruppi di bit essenziali, ovvero la combinazione minima tra quelli che contengono valori di uscita che non siano già coperti da altri gruppi.

Nel nostro esempio avremo quindi:

A \ BC	BC			
	00	01	11	10
0				1
1	1	1	1	1



MINIMIZZAZIONE DI FUNZIONI BOOLEANE

Fatto questo, per ciascun gruppo si mantiene solo la variabile (o le variabili) booleane il cui valore non cambia per tutti i mintermini del gruppo. Ciascuna variabile verrà presa negata se ha valore 0, o invariata se ha valore 1.

Rimane A

Rimane $B\bar{C}$

BC \ A	00	01	11	10
0				1
1	1	1	1	1

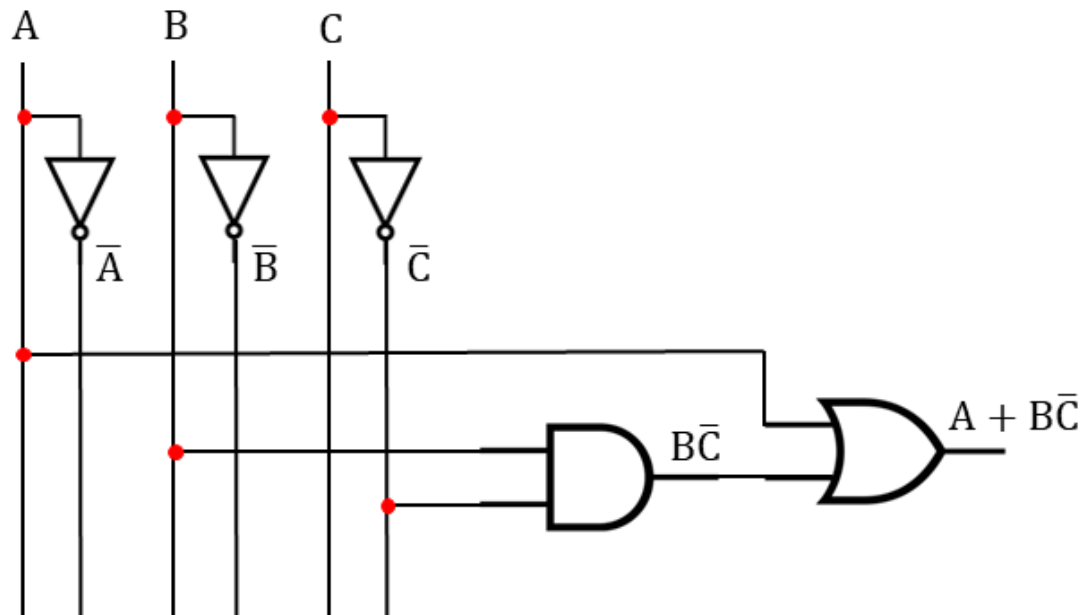
Questi nuovi termini semplificati, andranno sommati logicamente tra loro:

$$F = A + B\bar{C}$$



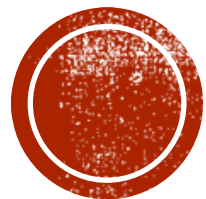
CIRCUITO DIGITALE

Ottenuta la funzione booleana equivalente a quella di partenza, ma in forma minima, si può procedere con il disegno dello schema relativo al circuito digitale combinatorio che la implementa:



Un circuito digitale è detto *combinatorio* quando è privo di retroazioni (il segnale viaggia a senso unico dall'input verso l'output), mentre è detto *sequenziale* quando presenta retroazioni (l'uscita di una porta può puntare a porte precedenti).





ARCHITETTURA DEI CALCOLATORI



LA MACCHINA DI VON NEUMANN

In un computer possiamo distinguere tre tipi di unità funzionali:

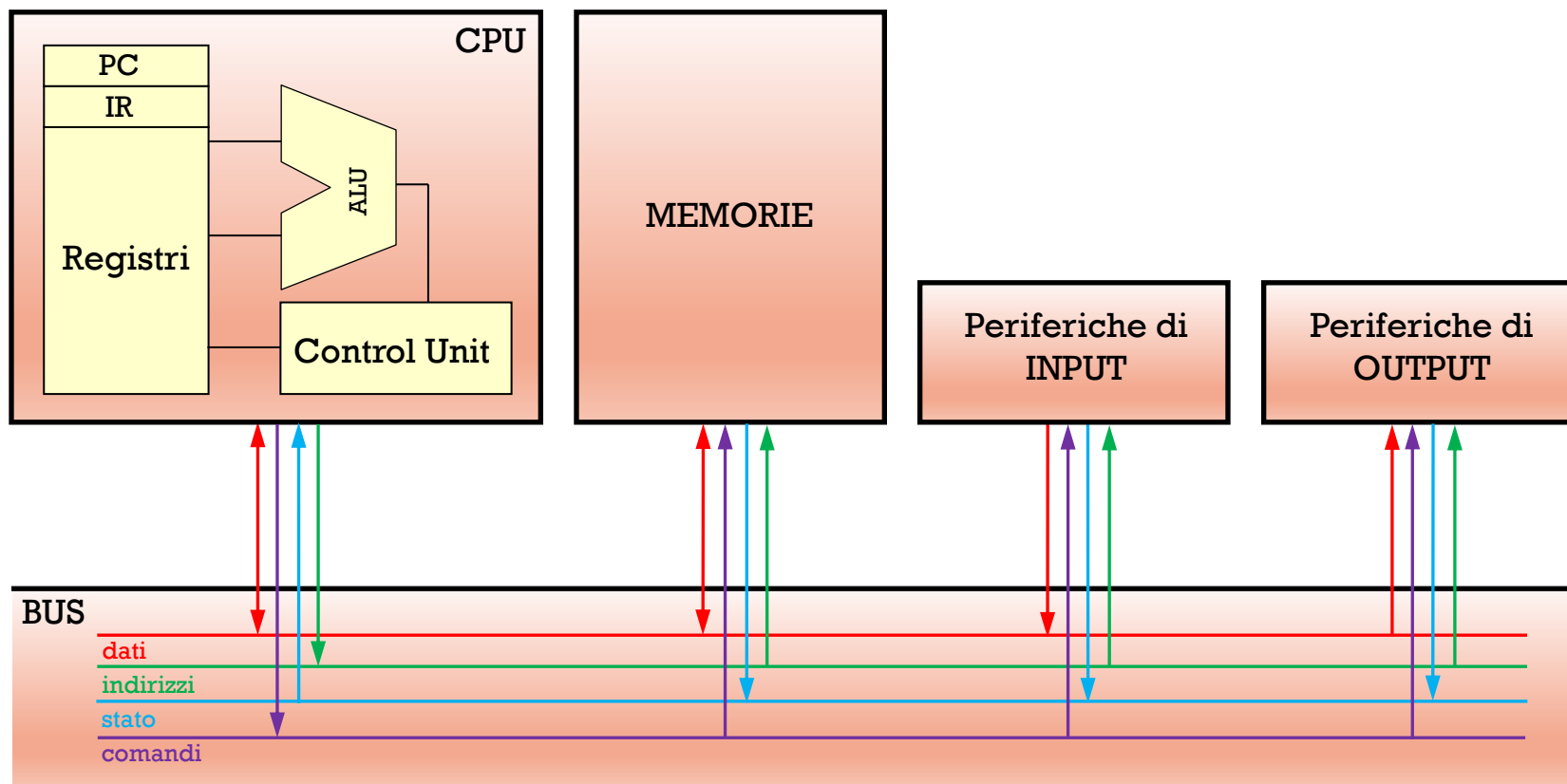
- Processore (fornisce la capacità di elaborazione delle informazioni);
- Memoria (centrale e di massa);
- Dispositivi di input/output.

Le parti comunicano attraverso un canale detto BUS, costituito da un insieme di linee elettriche digitali.

La maggior parte degli elaboratori moderni (compresi gli smartphone ed i tablet) sono basati sulla macchina di Von Neumann.

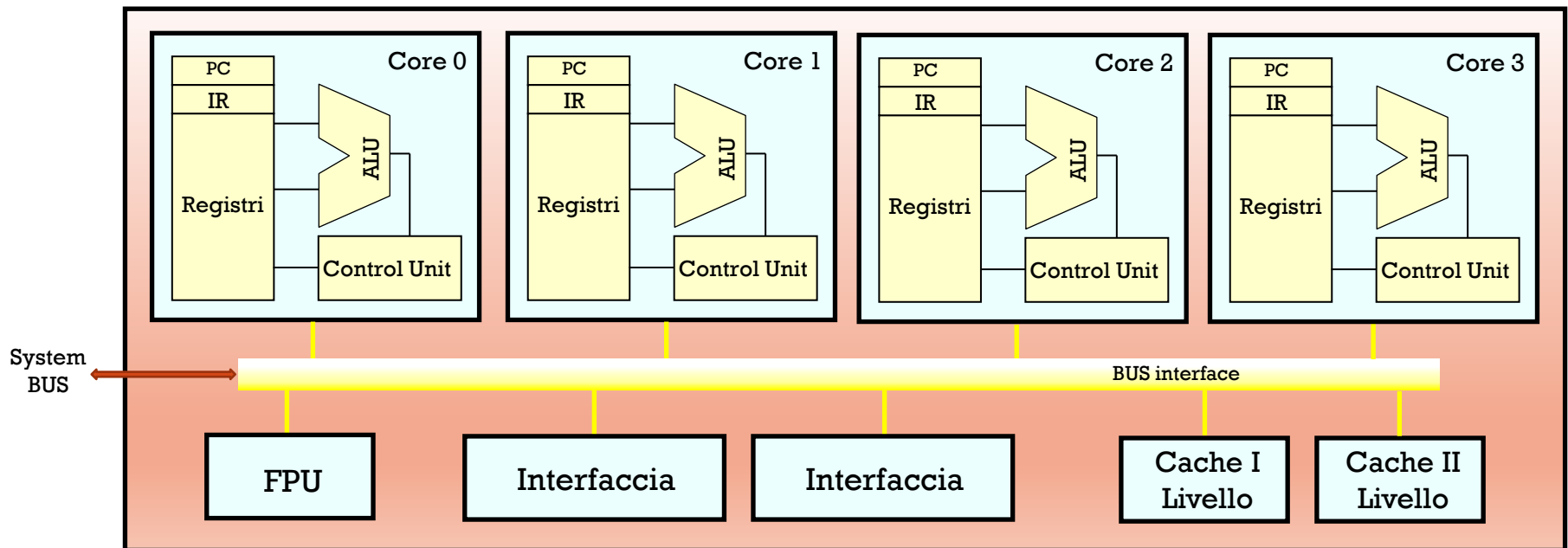


SCHEMA A BLOCCHI DI UN ELABORATORE



PROCESSORE

Oggi un processore non viene più identificato con una singola CPU, ma con più CPU (processori multicore). Sebbene sia più complesso, uno schema può essere approssimato come segue:



PROCESSORE

Dunque, è composto da blocchi con funzionalità diverse:

- 2^n CPU (Central Processing Unit);
- FPU (Floating Point Unit);
- Cache L1 (o di primo livello);
- Interfacce varie.

Se tutto questo è integrato all'interno di un unico chip, prende anche il nome di microprocessore. I moderni processori sono talvolta dotati anche di una Cache L2 (o di secondo livello).

Di solito le cache L1 sono più piccole (128KB o 256 KB) quindi integrate con lo stesso chip, mentre le cache L2 sono più grandi (valori tipici vanno da 512KB fino a 8MB o anche più), pertanto talvolta sono implementate in un circuito esterno (quindi con velocità di accesso minore).

In questo modo si riescono a gestire esigenze differenti in fase di elaborazione dei programmi.



CENTRAL PROCESSING UNIT (CPU)

Svolge tutte le operazioni di:

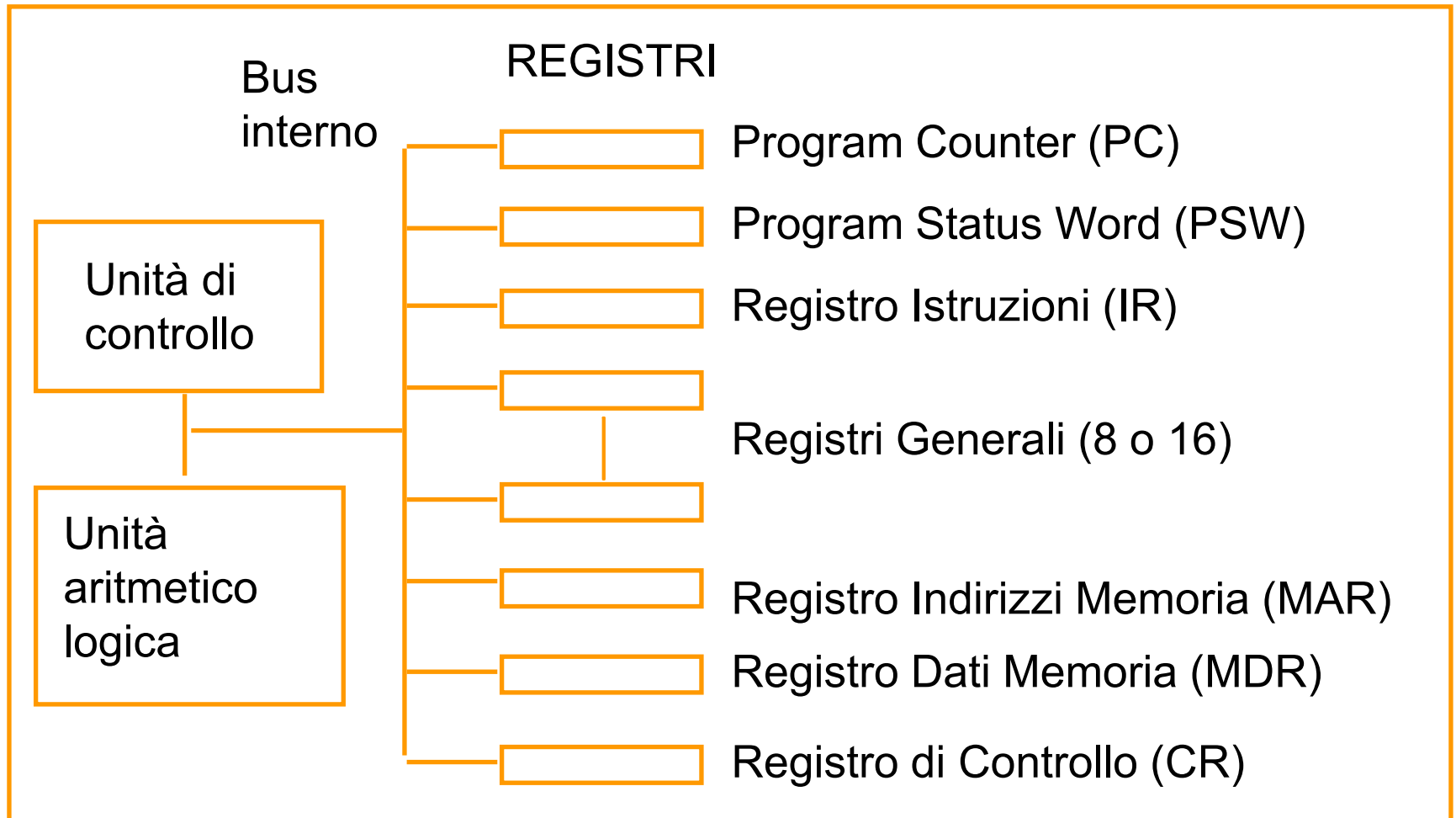
- elaborazione numerica;
- controllo e coordinamento di tutte le attività.

Ciascuna CPU si suddivide in:

- Unità logico-aritmetica (ALU);
- Unità di controllo (CU);
- Registri.



COMPONENTI DELLA CPU



FLOATING POINT UNIT (FPU)

L'unità di calcolo in virgola mobile è una parte del processore che si contraddistingue per essere specializzata nell'esecuzione di calcoli matematici in virgola mobile (dotati quindi di parte intera e parte decimale).

La maggior parte delle operazioni di calcolo svolte dalla CPU sono operazioni aritmetiche di base (somme, sottrazioni, moltiplicazioni e divisioni) effettuate sull'insieme dei numeri naturali o sull'insieme dei numeri relativi, utilizzando la ALU.

Quando invece è necessario effettuare calcoli più complessi (esponenziali, trigonometrici, etc.), lavorando con numeri reali, entra in gioco la FPU.



ARITHMETIC LOGIC UNIT (ALU)

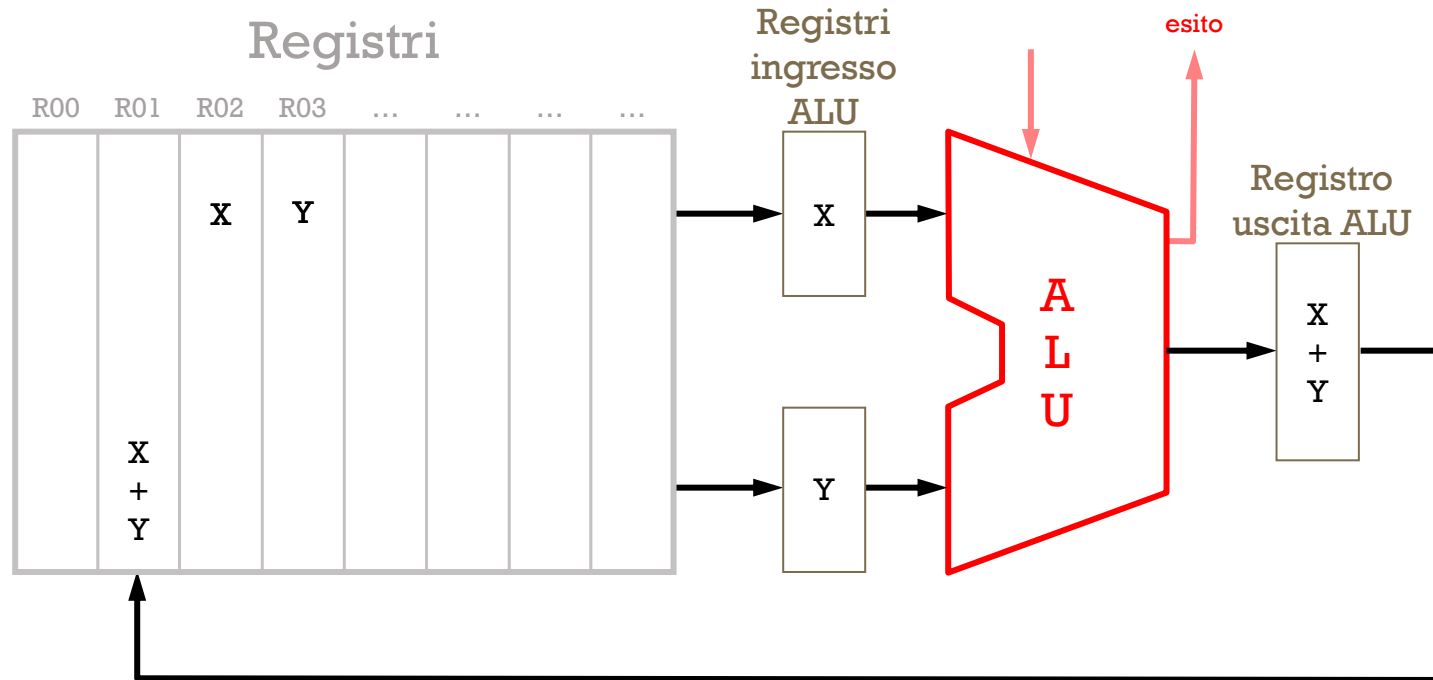
Svolge tutti i calcoli logici ed aritmetici operando direttamente con i registri.

In realtà, è costituita da circuiti elettronici in grado di eseguire solamente la somma di due numeri binari contenuti in due registri, oppure il confronto tra due numeri, sempre presenti in altrettanti registri.

Le altre operazioni vengono eseguite come combinazione delle sole operazioni indicate sopra (così come visto in precedenza).



ARITHMETIC LOGIC UNIT (ALU)



ARITHMETIC LOGIC UNIT (ALU)

Per quanto riguarda le operazioni logiche una ALU, di solito, è in grado di implementare le porte AND, OR, NOT e XOR.

Von Neumann, nel suo modello, ritenne ragionevole introdurre una unità specializzata allo scopo di compiere le operazioni elencate.

Si noti come la stessa ALU venga costruita facendo uso proprio delle porte logiche fondamentali AND, OR e NOT.



REGISTRI

Sono delle piccole porzioni di memoria locale utilizzata per il salvataggio di:

- dati acquisiti dalla memoria centrale o dalle unità di input;
- risultati delle operazioni eseguite dall' ALU.

I registri sono in numero limitato (tipicamente non oltre 256).

Sono unità di memoria estremamente veloci, di dimensione pari a 16, 32 o 64 bit (tale valore dipende dall'architettura del processore).



REGISTRI

I registri contengono dati ed istruzioni che vengono immediatamente elaborati.

Esistono due tipi di registri:

- registri speciali utilizzati dalla CU per scopi particolari;
- registri di uso generale (solitamente per operazioni aritmetiche).



IL SEGNALE DI CLOCK

Ogni elaboratore contiene un circuito di temporizzazione (detto clock), che genera un riferimento temporale comune per tutti gli elementi del sistema.

- T = **periodo** di clock
- f = **frequenza** di clock ($= 1/T$)

La frequenza di clock è misurata in Hertz (Hz).



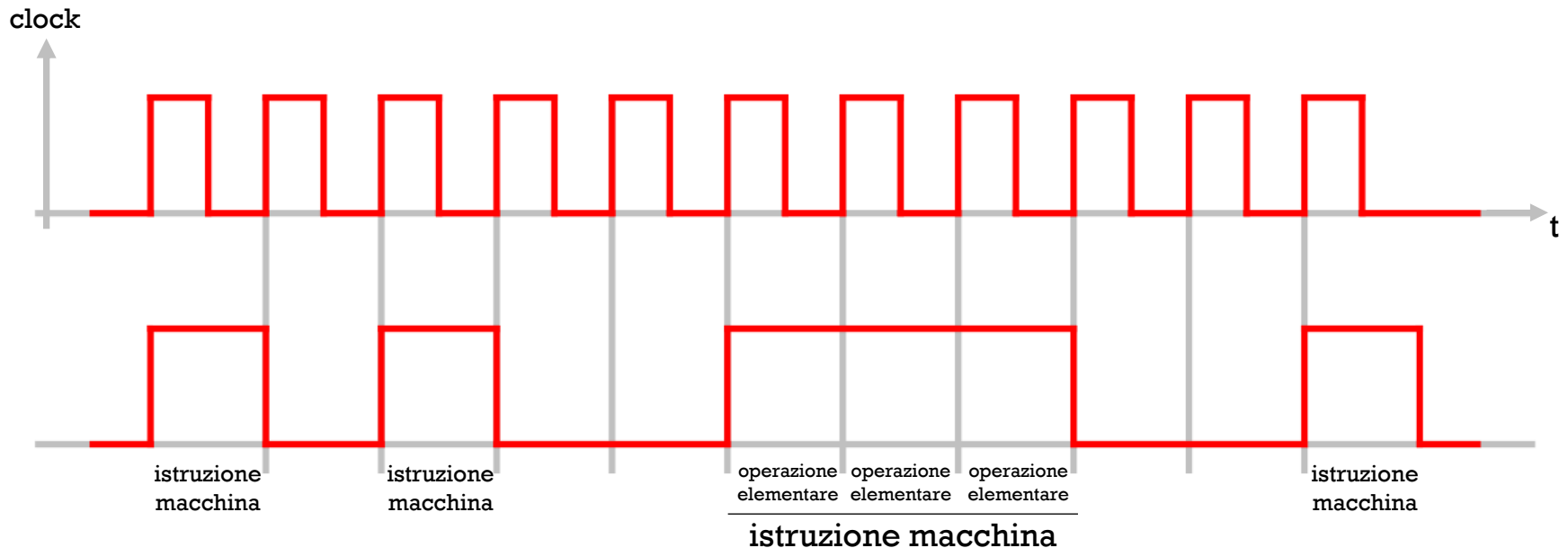
TEMPISTICA DELLE ISTRUZIONI

Un **ciclo-macchina** è il tempo necessario per svolgere una *operazione elementare*. È un multiplo del periodo del clock.

Un'*istruzione macchina* è ottenuta da una sequenza di operazioni elementari. Dunque, l'esecuzione di un'*istruzione macchina* richiede un numero intero di cicli macchina, variabile a seconda del tipo di istruzione.



TEMPISTICA DELLE ISTRUZIONI



Nell'esempio sopra, le prime due istruzioni macchina e l'ultima richiedono esattamente una operazione elementare ciascuna, la terza istruzione, invece, richiede l'esecuzione di 3 operazioni elementari. Si sta ipotizzando che 1 ciclo-macchina coincida con il periodo del clock.



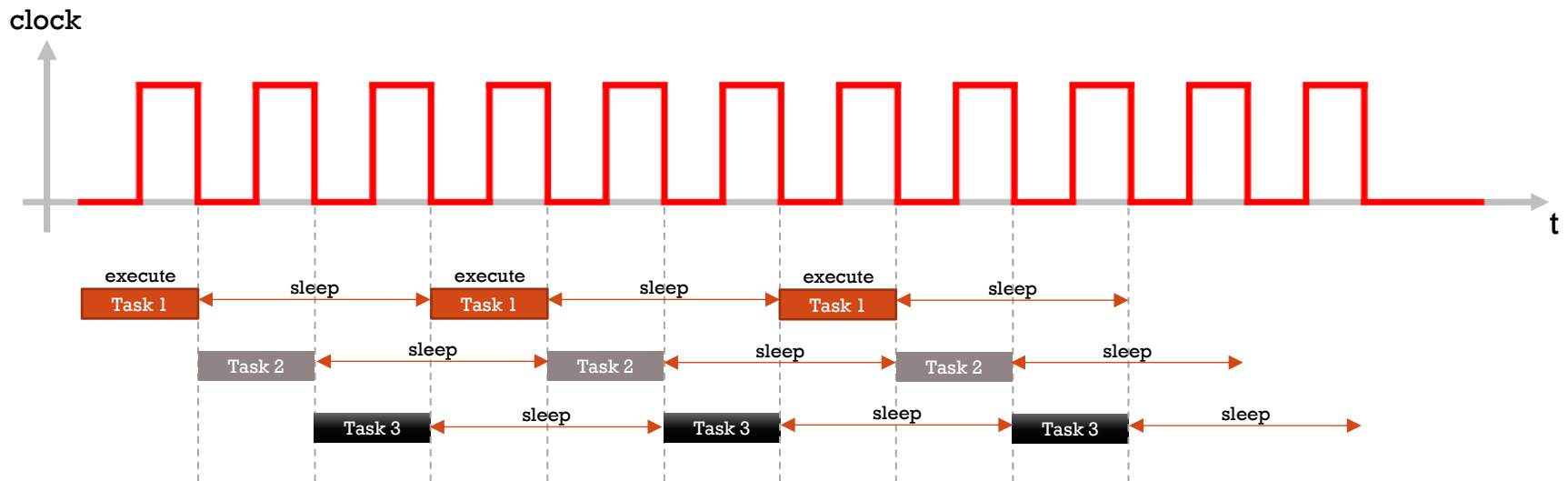
VELOCITÀ DEL MICROPROCESSORE

La velocità di elaborazione di un processore dipende dalla frequenza del clock.

I processori attuali hanno valori di frequenza di clock dell'ordine dei GHz (Giga Hertz), il che significa che un processore con tali caratteristiche è in grado di eseguire miliardi di operazioni elementari al secondo.



TIME SHARING



Nell'esempio sopra abbiamo ipotizzato che la fase di esecuzione di ciascuno dei 3 task preveda il coinvolgimento di una singola *operazione elementare*



MEMORIA

Viene utilizzata per conservare dati e programmi. Si suddivide in:

- **memoria principale;**

Memoria in grado di conservare dinamicamente dati e programmi che il processore sta utilizzando.

- **memoria di massa.**

Memoria in grado di conservare permanentemente dati e programmi degli utenti.



MEMORIA PRINCIPALE

La memoria *principale* (o centrale) è utilizzata per ospitare, in modo *temporaneo*, i dati ed i programmi che gli utenti richiedono di eseguire e/o elaborare.

Prevedono *piccole capacità* di memorizzazione e sono in genere costituite da RAM, memorie cache o registri. Alcune memorie centrali, invece, contengono in modo permanente informazioni necessarie al funzionamento dell'hardware (ROM).

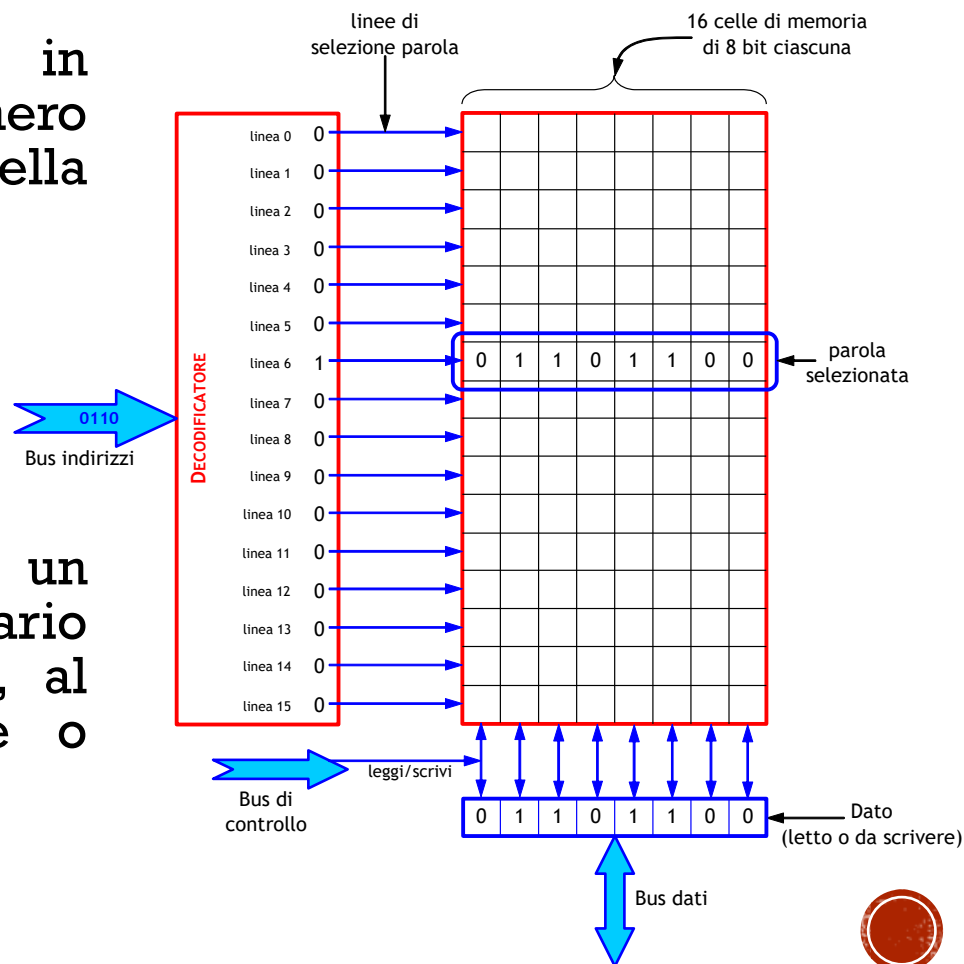
Quando si vuole eseguire un programma, esso viene sempre caricato nella memoria principale (insieme agli eventuali dati da elaborare).



ORGANIZZAZIONE DELLA MEMORIA PRINCIPALE

La memoria è organizzata in *celle*, ciascuna grande un numero di bit pari all'architettura della CPU, tra loro indipendenti.

Ad ogni cella è associato un *indirizzo*, cioè un numero binario progressivo a partire da 0, al quale puntare per leggere o scrivere dati.

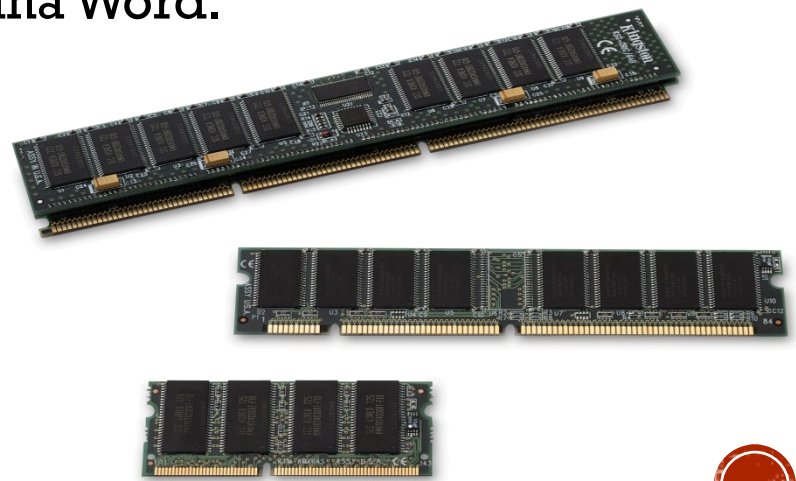


RANDOM ACCESS MEMORY (RAM)

È una memoria *volatile* basata su circuiti digitali, a disposizione della CPU per memorizzare programmi e dati necessari allo svolgimento delle elaborazioni richieste da un utente. A fine elaborazione, lo spazio occupato viene rilasciato.

Il tempo di accesso a tale memoria è il tempo necessario per leggere o scrivere l'informazione in una Word.

Il concetto «accesso casuale» indica che ogni cella ha lo stesso tempo di accesso, indipendentemente dall'indirizzo ad essa associato.



CARATTERISTICHE DELLA RAM

La dimensione di una RAM varia a seconda del tipo di computer e viene espressa in GB.

Le dimensioni tipiche della RAM di un calcolatore moderno vanno da 1GB a 16GB. La RAM di un Server, invece, può arrivare fino a 512GB.

Il tempo di accesso è dell'ordine di poche decine di nano-secondi (10^{-9} sec).

La RAM è una memoria costosa (il costo va inteso in valuta/bit).



READ ONLY MEMORY (ROM)

È una memoria di *sola lettura* che viene scritta direttamente dal produttore del computer su circuiti appositi.

Viene utilizzata per contenere le informazioni di inizializzazione usate ogni volta che si accende l'elaboratore. In alcuni casi, contiene un software particolare (chiamato firmware), installato all'interno sempre dal produttore.

Esistono diversi tipi di ROM, nate come evoluzione di quella sopra:

- PROM (Programmable ROM);
- EPROM (Erasable Programmable ROM);
- EEPROM (Electrically Erasable Programmable ROM);
- EAROM (Electrically Alterable ROM);
- Flash ROM.



READ ONLY MEMORY (ROM)

Una *PROM* (evoluzione di una ROM) viene prodotta vuota e può essere scritta da un utente, in seguito, utilizzando un'apposita attrezzatura, una volta sola.

Una *EPROM* (evoluzione di una PROM), può essere cancellata per intero, tramite esposizione ai raggi UV e riscritta.

Una *EEPROM* (evoluzione di una EPROM), può essere cancellata in blocchi di bit (al fine di essere riscritta), tramite l'applicazione di una tensione di valore superiore a quella di lavoro.

Una *EAROM* (che è un tipo particolare di EEPROM) può essere modificata in un singolo bit, invece che in blocchi di bit).

Una *Flash ROM*, infine, è una normale Flash utilizzata come ROM.

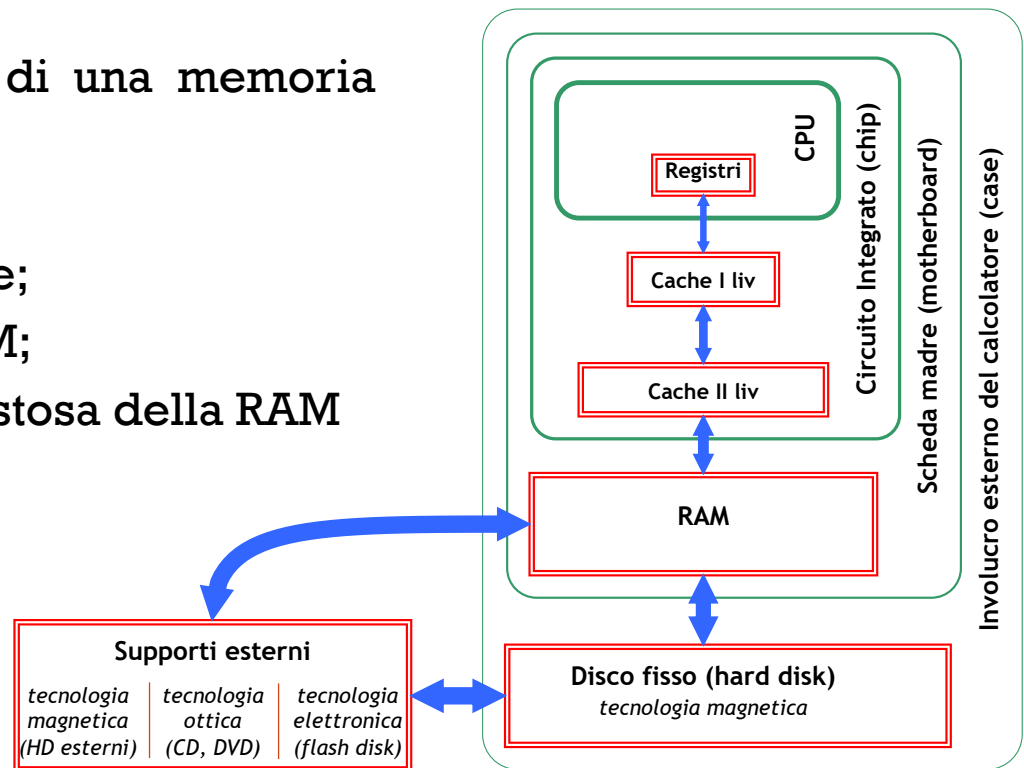


MEMORIA CACHE

Per migliorare le prestazioni di un computer si inserisce una memoria intermedia tra CPU e RAM, detta Cache.

Caratteristiche principali di una memoria Cache sono:

- è interna al processore;
- è più veloce della RAM;
- è di gran lunga più costosa della RAM ma anche più piccola.



MEMORIA CACHE

I dati e le istruzioni utilizzati con maggior frequenza vengono memorizzati nella *cache*, in modo da diminuirne il tempo di accesso ed aumentare, quindi, le prestazioni medie della CPU.

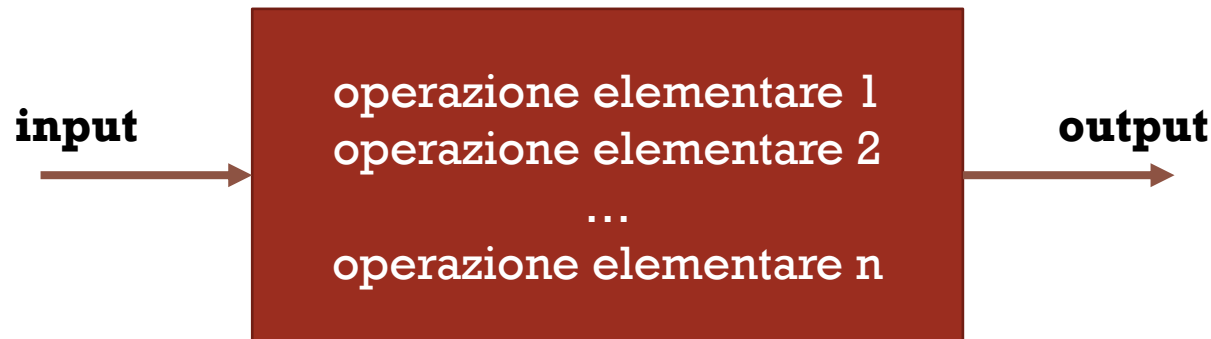
Diventa cruciale, dunque, il metodo per selezionare i dati e le istruzioni da inserire nella cache.

Le cache di I livello (L1), nei computer moderni, valgono 128KB o 256KB, mentre quelle di II livello (L2), hanno valori che vanno da 512 KB a 8÷ 12 MB.



ALGORITMO

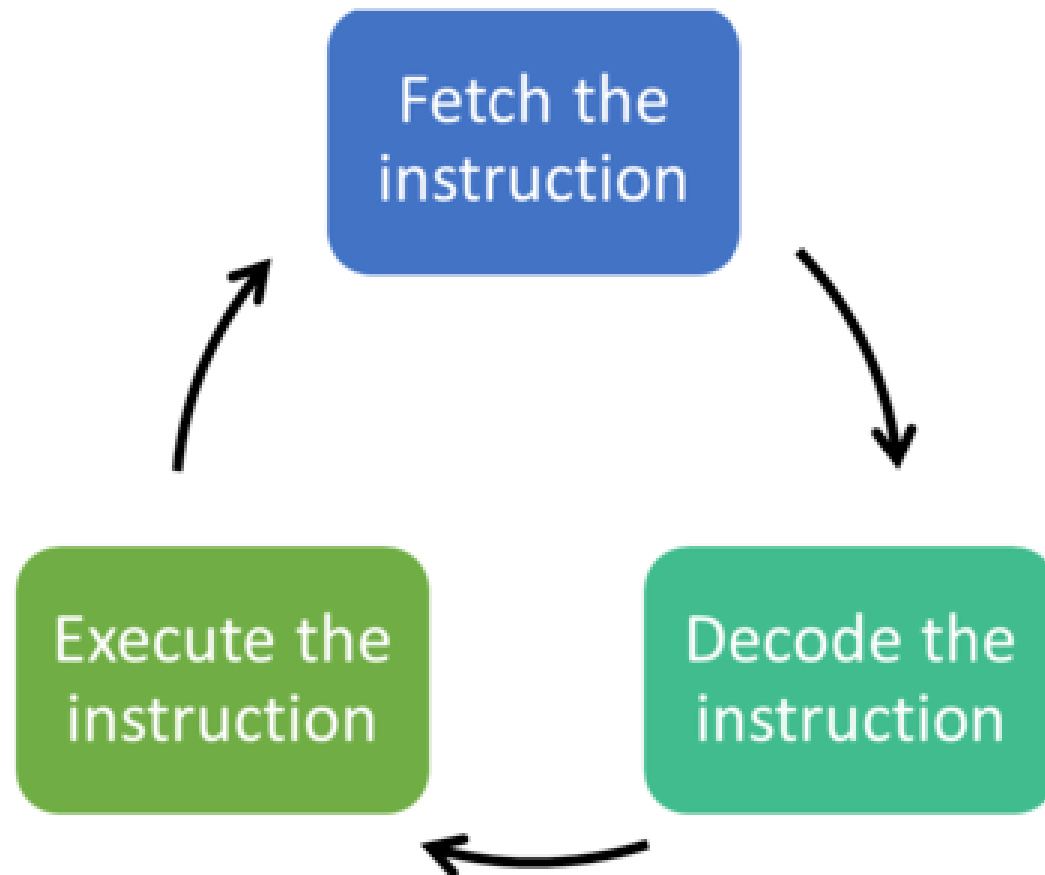
Si definisce *algoritmo* una sequenza ordinata e finita di operazioni elementari (istruzioni) che, presi un numero finito di dati in ingresso, dopo un tempo finito di elaborazione, produca un risultato univoco.



È importante osservare che le operazioni non devono poter essere scomposte in operazioni più semplici e non devono essere ambigue.



CICLO FETCH-DECODE-EXECUTE



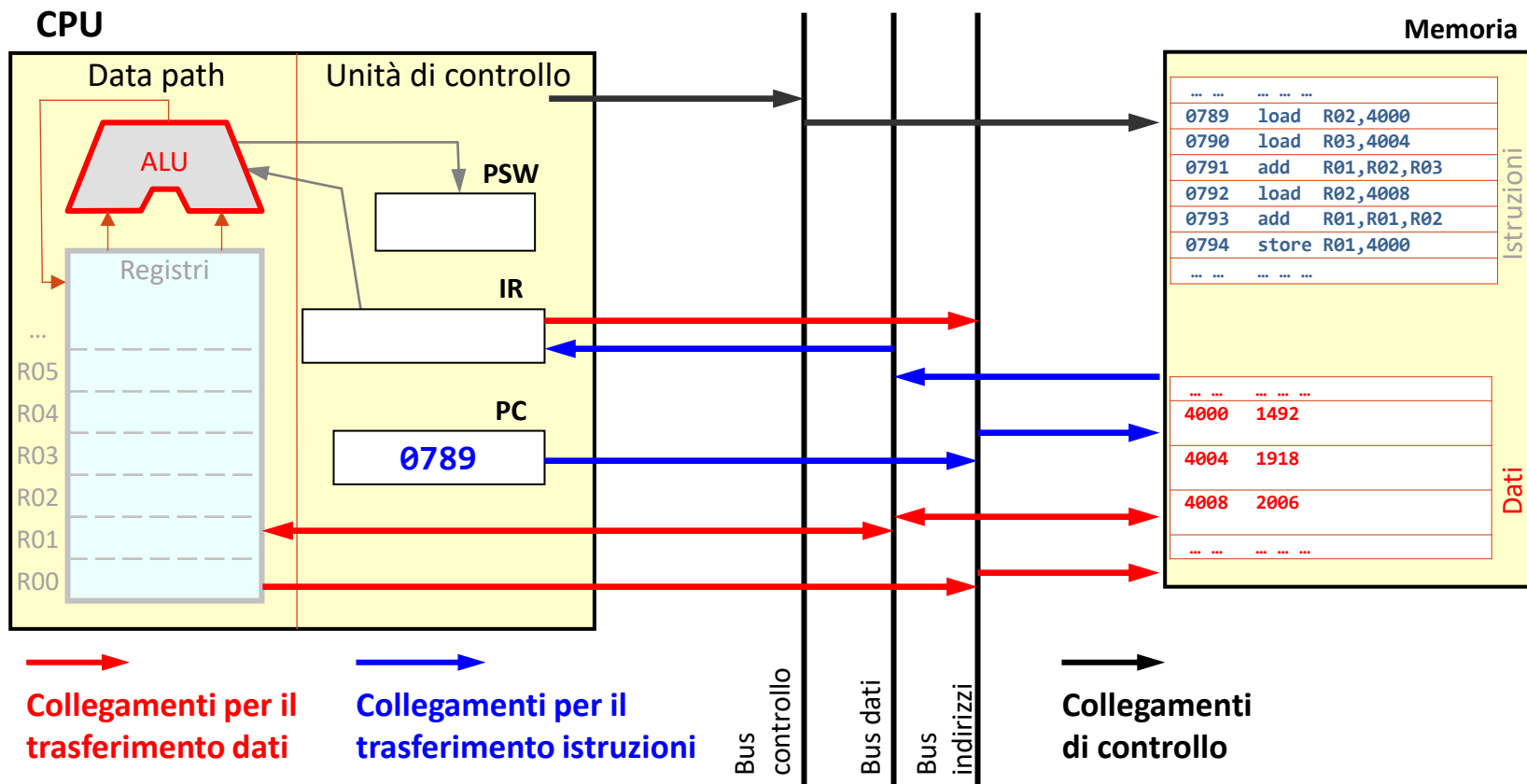
CICLO FETCH-DECODE-EXECUTE

1. Programma e dati sono caricati in memoria centrale (RAM).
2. La CU ESTRAE la prima istruzione del programma dalla memoria (puntandola attraverso l'indirizzo, indicato nel registro PC) e la trasferisce nel registro IR, quindi incrementa di uno l'indirizzo in PC, predisponendosi alla successiva istruzione.
3. L'istruzione viene DECODIFICATA ed ESEGUITA, prelevando eventuali dati associati, dalla memoria, sempre attraverso l'indirizzo corrispondente. I dati vengono in genere trasferiti nei registri generali.
4. La CU passa all'istruzione successiva, attraverso l'indirizzo presente nel registro PC.



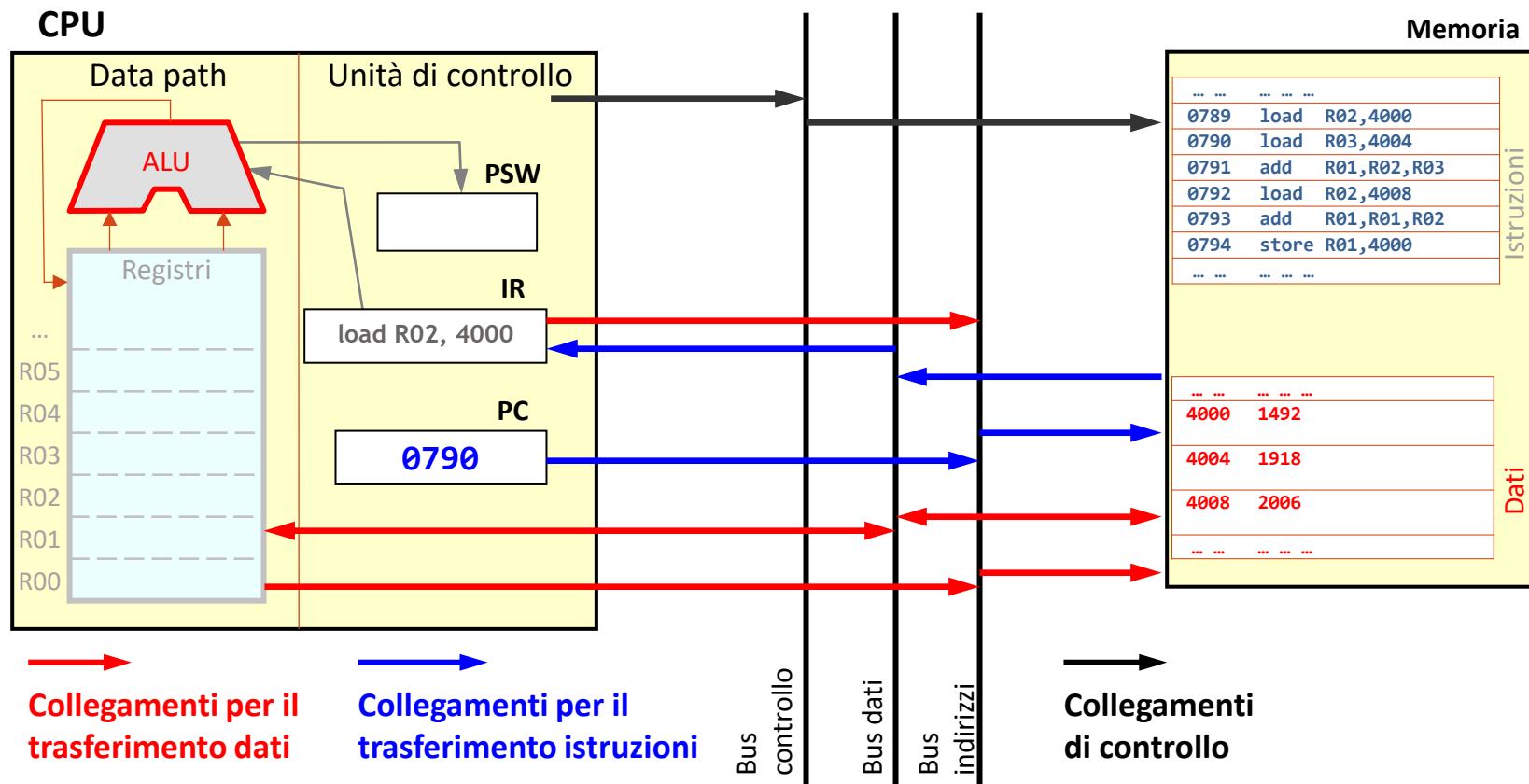
ESEMPIO

Di seguito, lo scenario dopo aver caricato programma e dati:



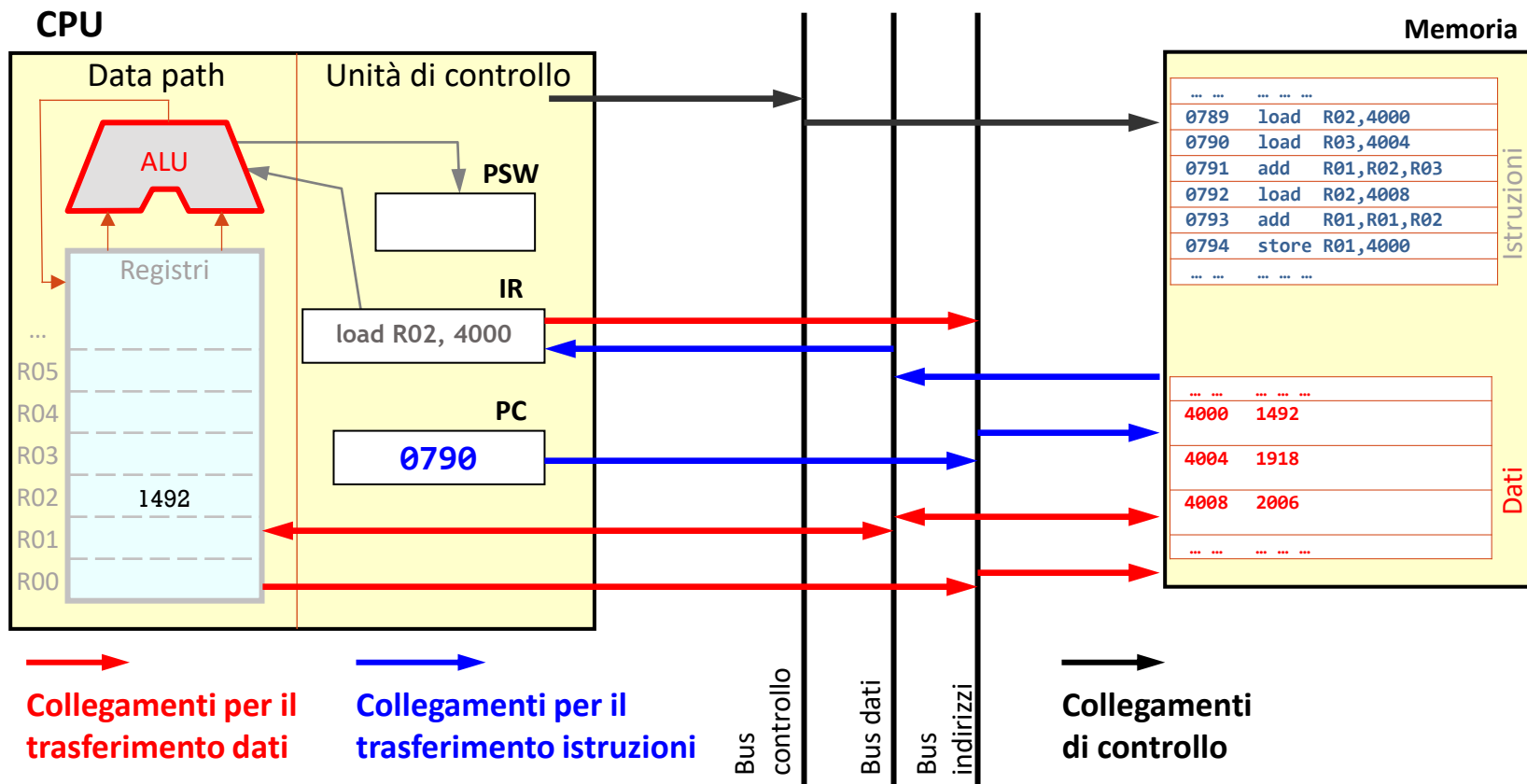
ESEMPIO

Vediamo un singolo ciclo: la CU ESTRAE l'istruzione all'indirizzo 0789, la trasferisce in IR, quindi incrementa PC:



ESEMPIO

La CU DECODIFICA l'istruzione in IR, (ovvero carica il dato all'indirizzo 4000 della RAM, nel registro R02) e la ESEGUE:



ESEMPIO

La CU prosegue con lo stesso lavoro visto nelle slides precedenti per il primo ciclo, fino a quando non completa l'esecuzione dell'intero blocco di istruzioni (l'algoritmo da eseguire), ovvero fino all'istruzione all'indirizzo 0794 della RAM.

Ad ogni ciclo, dunque, accede alla locazione di memoria associata all'indirizzo, estrae l'istruzione che vi trova dentro portandola nel registro IR (cancellando quindi l'eventuale precedente istruzione presente), incrementa l'indirizzo contenuto nel registro PC di una unità, decodifica l'istruzione in IR (eventualmente estraendo dati dalla RAM e portandoli in uno dei registri generali) e la esegue.



MEMORIA SECONDARIA

La memoria *secondaria* (o di massa) è utilizzata per conservare in modo *permanente* i dati ed i programmi degli utenti.

Richiede *grande capacità* di memorizzazione ed è principalmente costituita da dischi magnetici, CD-ROM, DVD, Flash Memory o nastri.

Quando si vuole eseguire un programma (così come visto nell'esempio precedente), esso viene sempre caricato dalla memoria di massa (insieme agli eventuali dati da elaborare) alla memoria principale.



MEMORIA SECONDARIA

Tre tecnologie possibili:

- **Magnetica**

- ✓ Dischi magnetici (Floppy-disk, Hard-disk)
 - Accesso casuale
 - Operazioni di Lettura/Scrittura
- ✓ Nastri magnetici
 - Accesso sequenziale (legato alla posizione del dato)
 - Operazioni di Lettura/Scrittura

- **Ottica**

- ✓ CD-ROM, DVD
 - Accesso casuale (tipicamente solo scrittura)

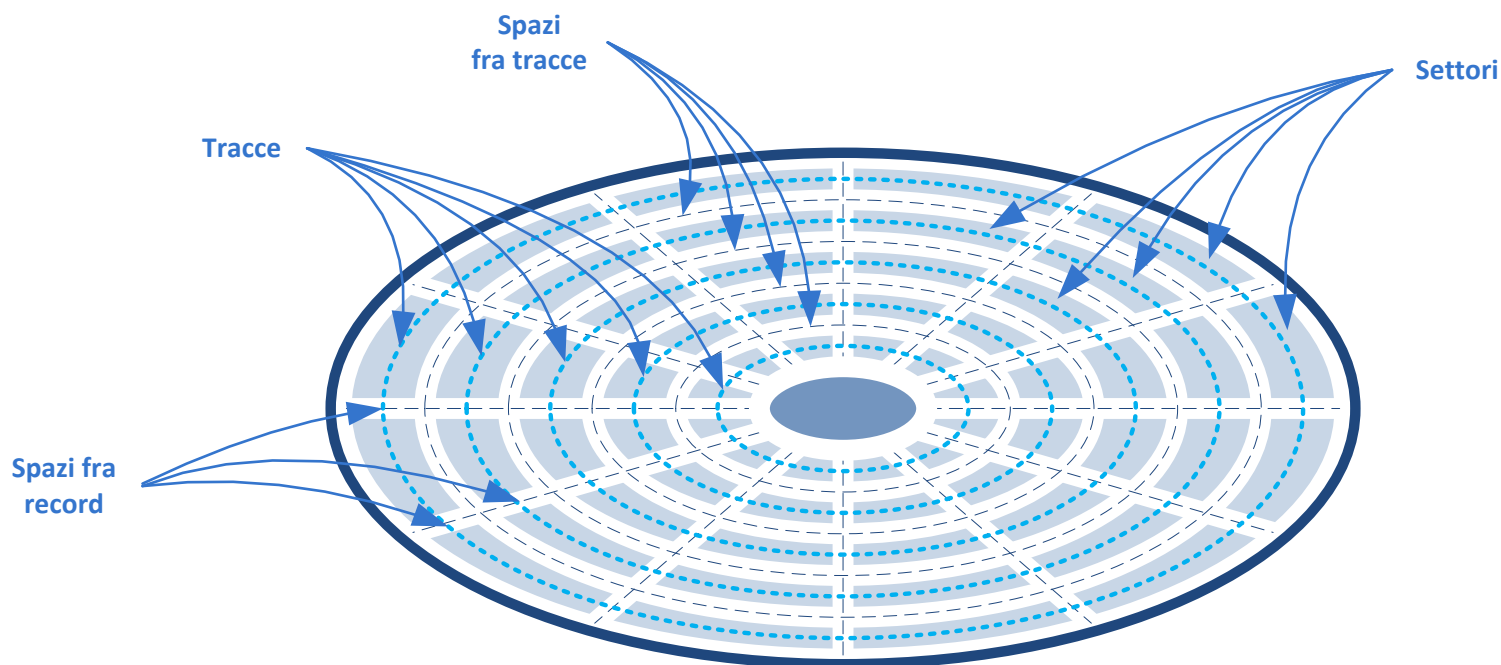
- **Flash memory**

- ✓ Pen drive, schede di memoria (Sd, micro Sd, memory stick, compact flash, etc.)
 - Accesso casuale (lettura/scrittura)



ORGANIZZAZIONE FISICA DEI DATI NEI DISCHI

Gli hard disk sono suddivisi in *tracce* concentriche e *settori*:



ORGANIZZAZIONE FISICA DEI DATI NEI DISCHI

Ogni *settore* è una “fetta” di disco. I settori suddividono ogni *traccia* in porzioni di circonferenza dette *blocchi*.

La suddivisione di un disco in tracce e settori viene indicata con il termine *formattazione*.

Per effettuare un'operazione di lettura o di scrittura su un blocco, è necessario che la testina raggiunga il blocco desiderato.

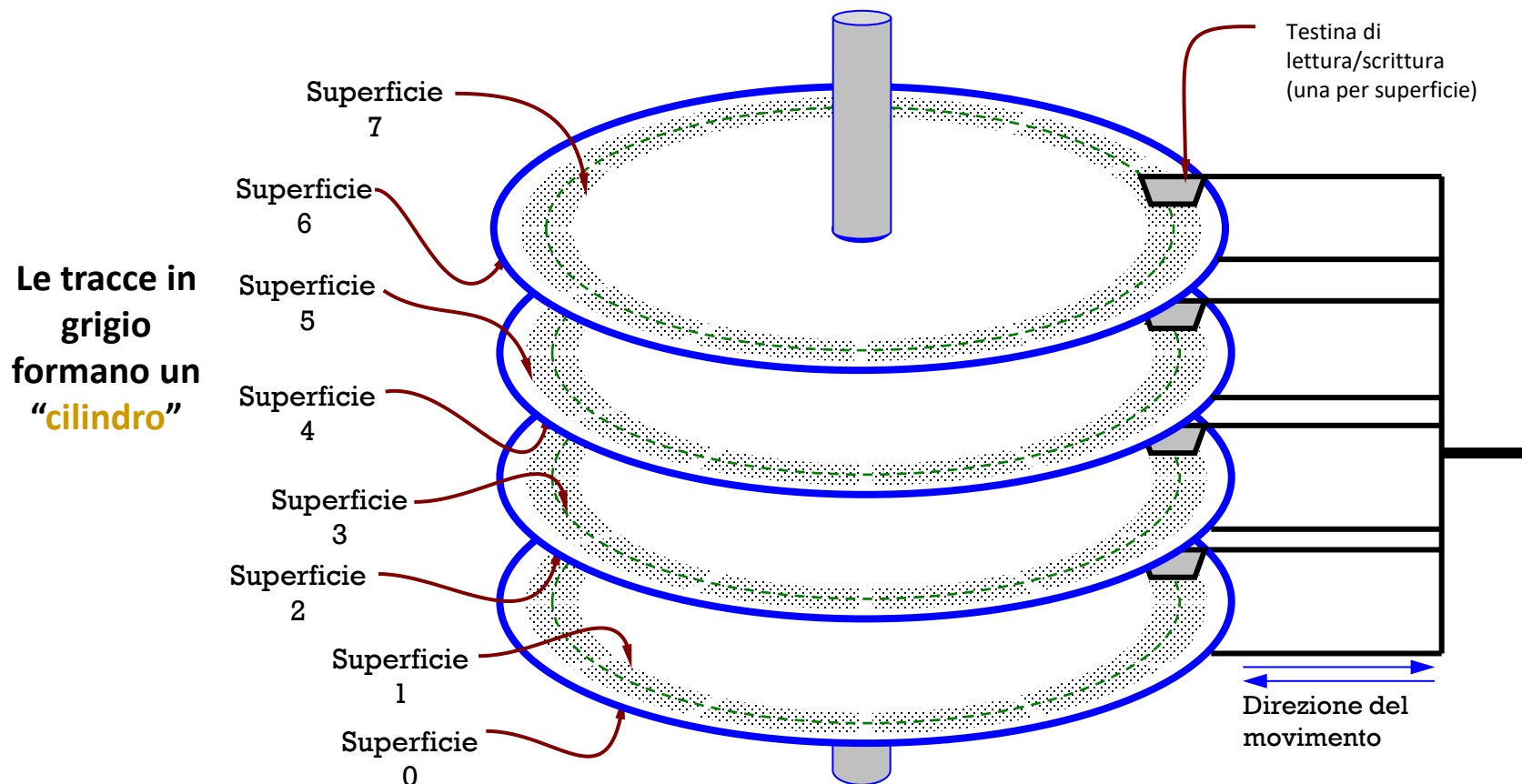
Ciascun disco può essere gestito in due modalità:

- Single-side.
- Double-side.

Per aumentare la capienza, spesso si montano più dischi, double-side, all'interno dello stesso case.



ORGANIZZAZIONE FISICA DEI DATI NEI DISCHI



ORGANIZZAZIONE FISICA DEI DATI NEI DISCHI

Il *tempo di accesso* ai dati (access time) dipende da tre fattori fondamentali:

- **Seek time**

La testina si sposta in senso radiale fino a raggiungere la traccia desiderata (impiega 1ms per tracce adiacenti, in generale tra 5 e 15ms in media).

- **Latency time**

Quando è in funzione, i dischi ruotano ad una velocità tra 5400 e 10800 rpm. Occorre attendere, quindi, che il settore desiderato passi sotto la testina. Varia tra 2,7 e 5,4ms.

- **Transfer time**

Tempo di lettura vero e proprio. In genere dipende dalla velocità di rotazione del disco e dalla densità di bit per cm lineare (della traccia). In genere si va da 5 a 20 MBps.



DISCHI MAGNETICI

Sono dotati di una grande capacità di memorizzazione. Fino a pochi anni fa i floppy disk rappresentavano il supporto mobile più usato.

Oggi sono stati largamente sostituiti da altri supporti quali, ad esempio, gli hard-disk esterni o i pen drive.

Un hard-disk, oggi, può raggiungere dimensioni di diversi TB, sono memorie più lente della memoria centrale, ma molto meno costose.



DISCHI OTTICI

Un disco CD-ROM (memoria ottica) è un disco di policarbonato, in genere dal diametro di 12 cm, che ha la capacità di memorizzare 640MB, 700MB, 800MB o 870 MB (ben più rari).

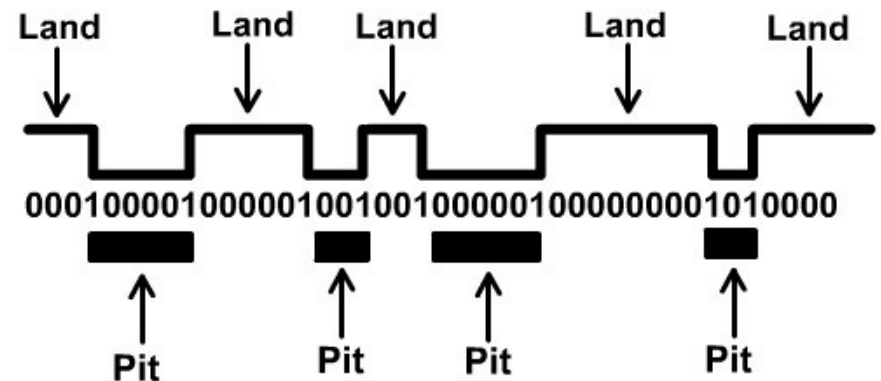
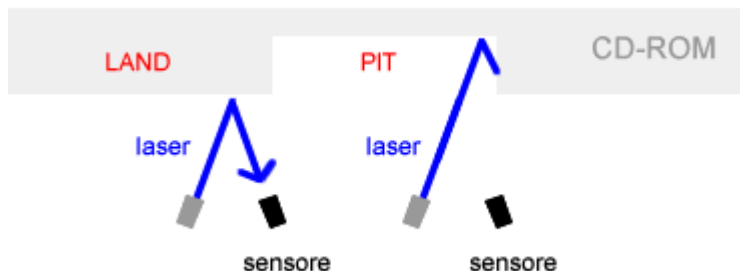
Sono stati oramai quasi totalmente sostituiti con i DVD, la cui capacità viene aumentata fino a 4,7 GB (se single layer - single side, ovvero lato unico - singolo strato) a 9,4 GB (se single layer double side, ovvero doppio lato - singolo strato) o a 17 GB (se double layer - double side, ovvero doppio lato – doppio strato).

I bit vengono impressi sotto forma di *pit* e di *land*. I pit sono dei solchi scavati nel substrato, le land sono zone non scavate del substrato. Il substrato viene realizzato vicino alla superficie e metallizzato, in modo da essere riflettente.



DISCHI OTTICI

Le transizioni *pit-land* e *land-pit*, nelle quali il sensore cambia di stato, vengono interpretate come un bit 1, mentre le aree piane presenti prima e dopo ogni transizione (nelle quali il sensore permane nello stato eccitato o non eccitato), vengono interpretate come 0 (o sequenze di zeri).



DISCHI OTTICI

Una recente evoluzione del DVD sono i Blu-ray, progettati dalla Sony per poter consentire la memorizzazione di video ad altissima risoluzione (4K).

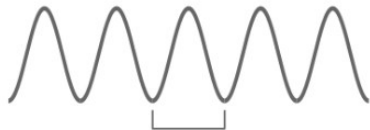
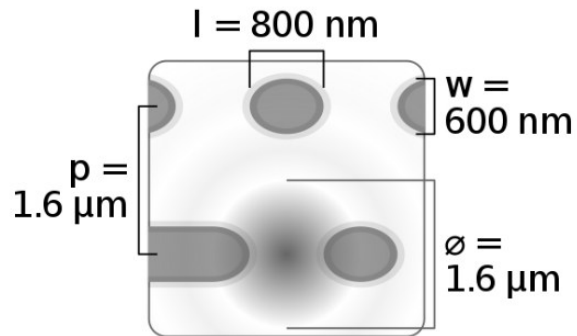
Un supporto Blu-ray, oggi, può raggiungere capienze fino a 25GB, ma sono già in rilascio dischi che raggiungono fino a 50GB. La Sony lavora da tempo a varianti in grado di raggiungere perfino 200GB.

La differenza sostanziale consiste nell'utilizzo di un raggio laser a minor lunghezza d'onda (che assume una colorazione blu).

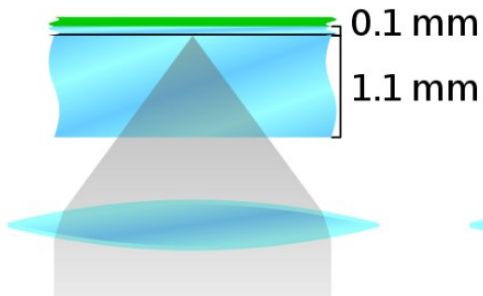


DISCHI OTTICI

CD

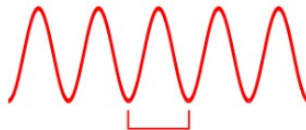
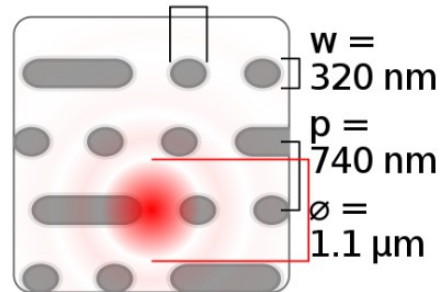


$\lambda = 780 \text{ nm}$

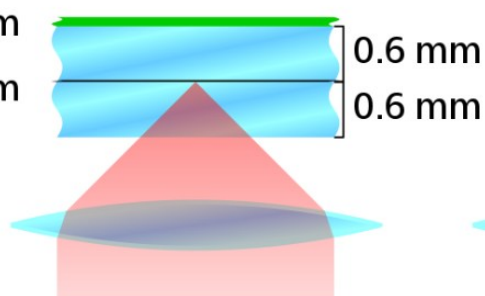


DVD

$l = 400 \text{ nm}$

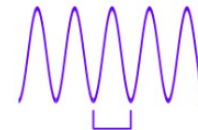
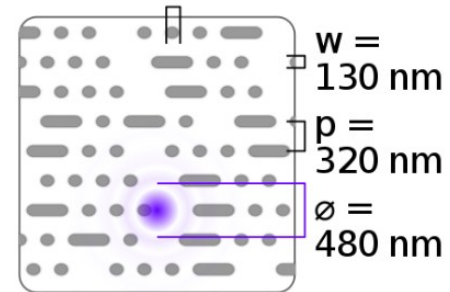


$\lambda = 650 \text{ nm}$

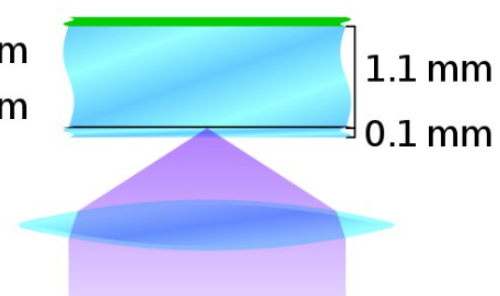


Blu-ray

$l = 150 \text{ nm}$



$\lambda = 405 \text{ nm}$



FLASH MEMORY

È una tipologia di memoria *a stato solido*, di tipo non volatile, che per le sue prestazioni può anche essere usata come memoria a lettura-scrittura.

Diversamente dalle tecnologie precedenti, la tecnologia Flash ha reso possibile il salvataggio o la cancellazione di dati in un unico step, introducendo quindi un incredibile guadagno in velocità, e grazie alla non-volatilità è usata frequentemente nelle fotocamere digitali, nei lettori di musica portatili, nei cellulari,

nei pendrive (chiavette), nei palmari, nei moderni computer portatili e in molti altri dispositivi che richiedono un'elevata portabilità e una buona capacità di memoria per il salvataggio dei dati.

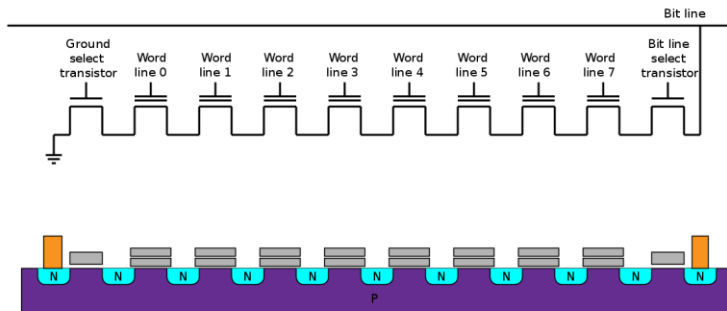
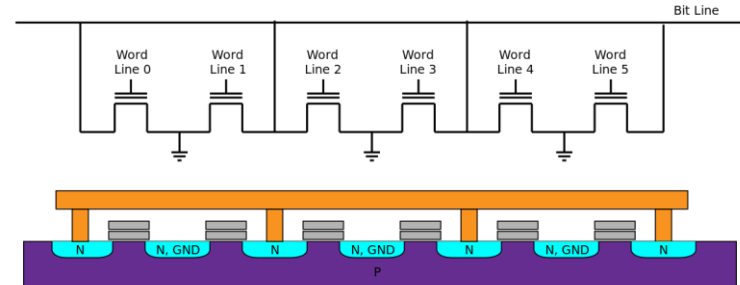


FLASH MEMORY

Esistono due tipologie differenti di memorie flash:

- NOR flash;
- NAND flash.

Nelle NOR, la bit line di ciascun array viene mantenuta al valore High (H), se tutte le celle sono poste al valore Low (L).

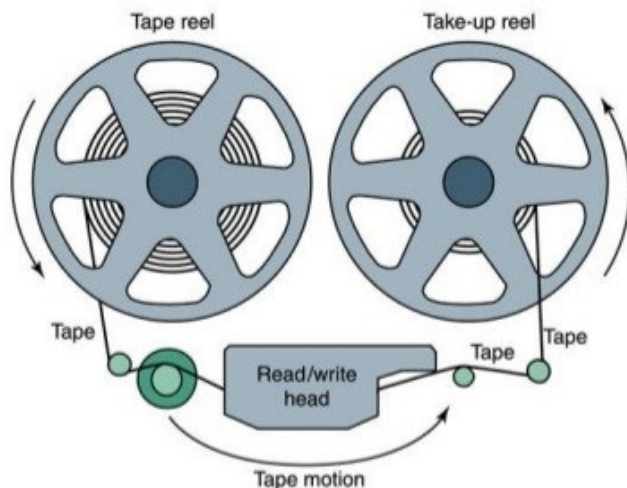


Nelle NAND, invece, la bit line di ciascun array può essere mantenuta al valore High (H), anche se una sola cella è posta al valore High (H).



NASTRI MAGNETICI

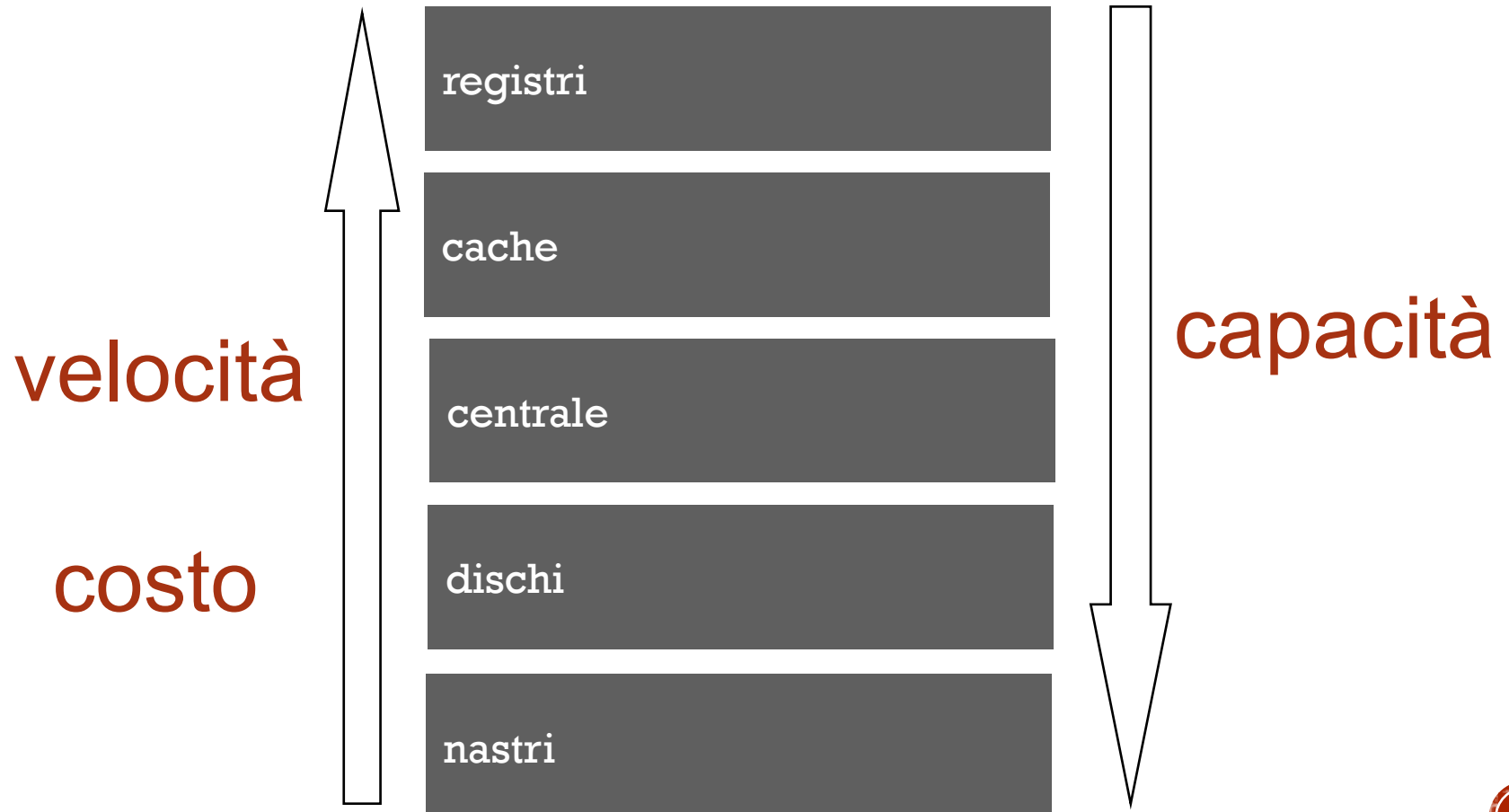
Sono stati molto utilizzati nei primi calcolatori (anche destinati al mercato «home»), come supporto di memorizzazione per programmi e dati.



I nastri magnetici, oggi, vengono utilizzati esclusivamente come unità di *backup* per data center, in virtù della loro enorme capacità di memorizzazione ed al basso costo.



GERARCHIA DI MEMORIE



DISPOSITIVI DI INPUT/OUTPUT

I dispositivi di I/O, detti anche *periferiche*, permettono di realizzare l'interazione uomo/macchina. È importante notare che oltre alle periferiche di tipo tradizionale, ne esistono di varia natura per applicazioni specifiche (ad esempio in campo medico, geologico, ingegneristico, etc.).

La loro funzione principale è quella di consentire l'ingresso dei dati da elaborare, attraverso il bus, fino al processore e/o l'uscita delle informazioni restituite dalle elaborazioni, fino alla stampa su carta o a video.

Il controllo è eseguito da un software, detto “*driver*”, che ne gestisce le funzionalità a basso livello (linguaggio macchina).



DISPOSITIVI DI INPUT/OUTPUT



Terminali

- Tastiera.
- Video.

Dispositivi di puntamento

- Mouse.
- Trackball.
- Penna ottica.
- Schermo sensibile (Touch Screen).



DISPOSITIVI DI INPUT/OUTPUT

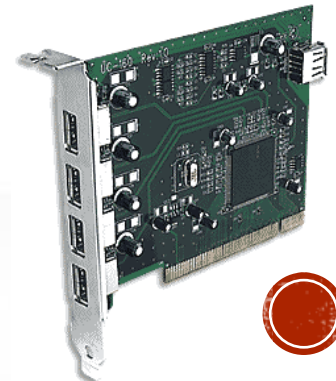
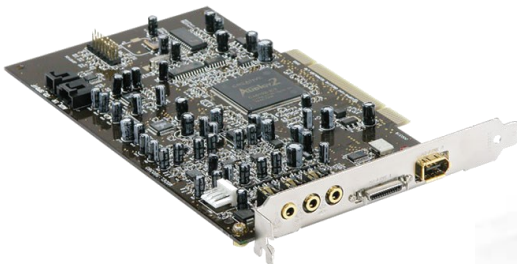
Stampanti

Presentano varie tecnologie:

- Ad aghi.
- Ink jet.
- Laser.



Scanner, modem, plotter, scheda audio, etc.



UNITÀ DI INPUT

Tastiera

La tastiera (o keyboard) può essere ridotta, con 86 tasti, oppure espansa, con almeno 102 tasti. Si distinguono i seguenti elementi:

- la tastiera alfanumerica;
- i tasti funzione;
- Il tastierino numerico;
- i tasti di controllo:
 - ✓ STAMP, utilizzabile per stampare il contenuto dello schermo su carta;
 - ✓ CANC (o DEL), per cancellare il carattere su cui è collocato il puntatore;
 - ✓ (HOME) e (End), utilizzati nei WORD PROCESSOR per posizionarsi rapidamente all'inizio o alla fine della riga;
 - ✓ (PagUp) e (PagDn), per spostarsi rapidamente una videata in su o in giù;
 - ✓ TASTI FRECCIA, per spostarsi rapidamente in alto, in basso, a destra o a sinistra.



UNITÀ DI INPUT

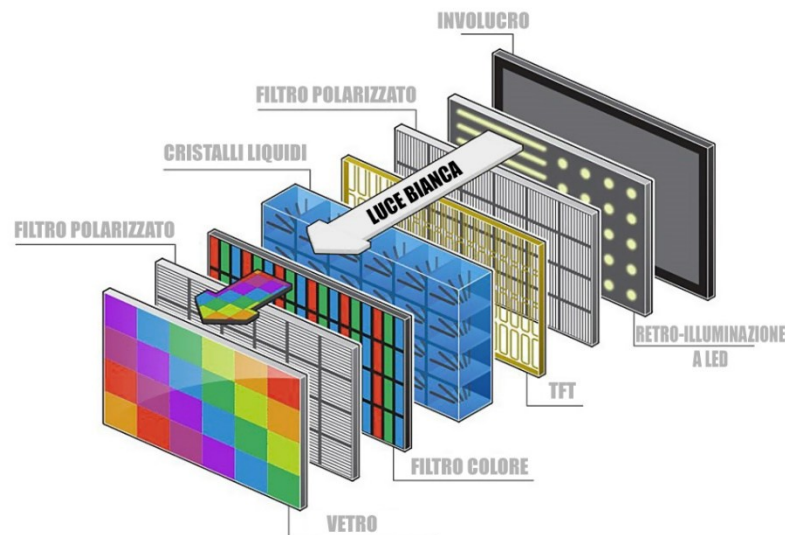
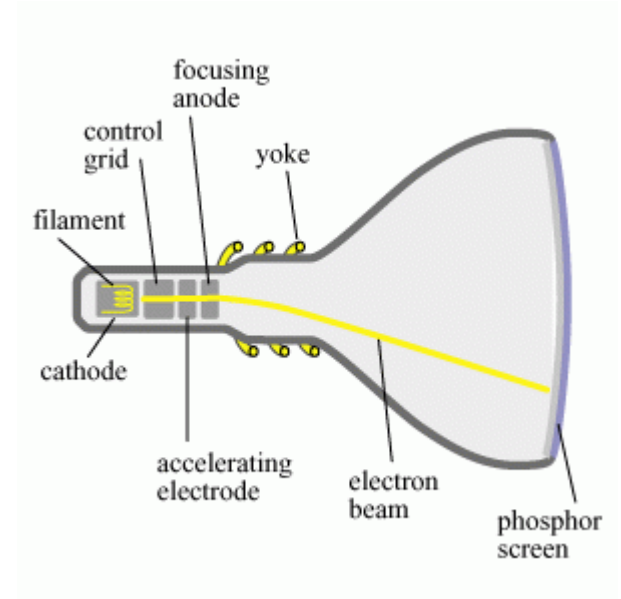
- **Mouse.**
- **Trackball.**
- **Touchpad** (sostituisce il mouse nei portatili).
- **Joystick.**
- **Penna ottica.**
- **Scanner:**
 - Manuali;
 - Piani.
- **Fotocamere e videocamere.**
- **Microfono.**



UNITÀ DI OUTPUT: MONITOR

Possono essere di tre tipi:

CRT (tubo a raggi catodici), utilizza la tecnologia simile a quella usata nei vecchi televisori.



LCD

(schermo a cristalli liquidi).

LED (schermo a cristalli liquidi con retroilluminazione a led).



UNITÀ DI OUTPUT: MONITOR

Parametri caratteristici di un monitor sono:

Pollici

unità di misura del monitor (calcolata sulla diagonale dello schermo). Un pollice equivale a 2,54cm. Il rapporto base/altezza può avere valore 4/3 o 16/9.

Pixel

Il monitor può essere pensato come una griglia composta da migliaia di quadratini detti *pixel*. Maggiore è il numero di pixel, più definita è l'immagine ottenuta nello schermo.

Risoluzione del monitor

Si esprime sempre in pixel e indica il numero di punti in orizzontale e in verticale sullo schermo (es. 1024 x 768). I formati standard più usati, oggi, prevedono le seguenti risoluzioni:

HD (o HDTV): 1280 x 720 (ovvero 921.600)

Full HD: 1920 x 1080 (ovvero 2.073.600 px)

Ultra HD: 3840 x 2160 (ovvero 8.294.400 px)

4K (Digital Cinema): 4096 x 2160 (ovvero 8.847.360 px)



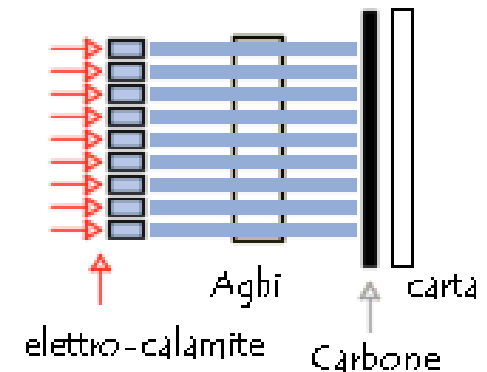
UNITÀ DI OUTPUT: STAMPANTI

Esistono diverse tecnologie per le stampanti:

- A impatto
 - ✓ aghi.
- A non impatto
 - ✓ getto di inchiostro.
 - ✓ laser.

Le **stampanti ad ago** usano una matrice in genere a 9 o a 24 aghi, posti in verticale, con la quale scrivono i caratteri per singola colonna.

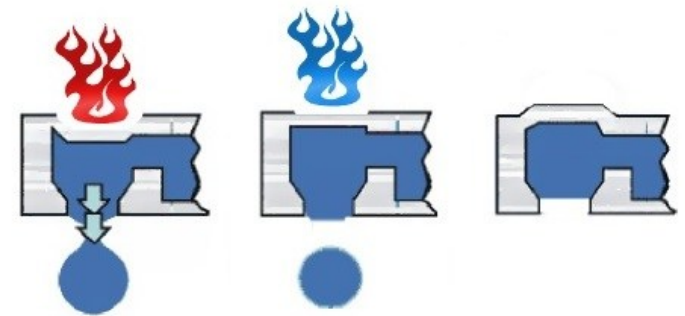
Tra il foglio e gli aghi, viene posizionato un nastro preinchiostroato. Quelle a 24 aghi arrivano ad una risoluzione di 216 ppp.



UNITÀ DI OUTPUT: STAMPANTI

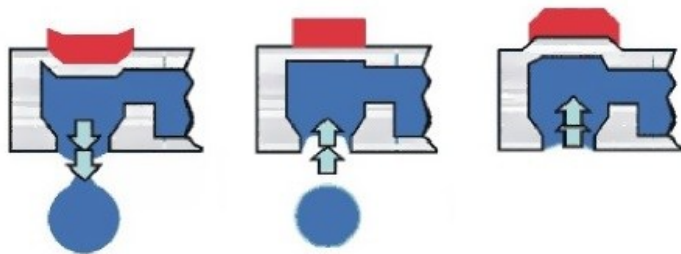
Le stampanti a **getto d'inchiostro** possono essere a *testina termica*:

ogni ugelli ha un elemento riscaldante (un resistore). Il calore fa uscire una goccia di inchiostro, mentre il raffreddamento fa rientrare l'inchiostro.



Vi sono poi quelle a *testina piezoelettrica (a bolla di inchiostro)*:

ogni ugello ha un quarzo piezoelettrico che, se eccitato sulla sua frequenza di risonanza, si deforma facendo prima uscire la goccia di inchiostro a pressione e poi staccandola per risucchio.



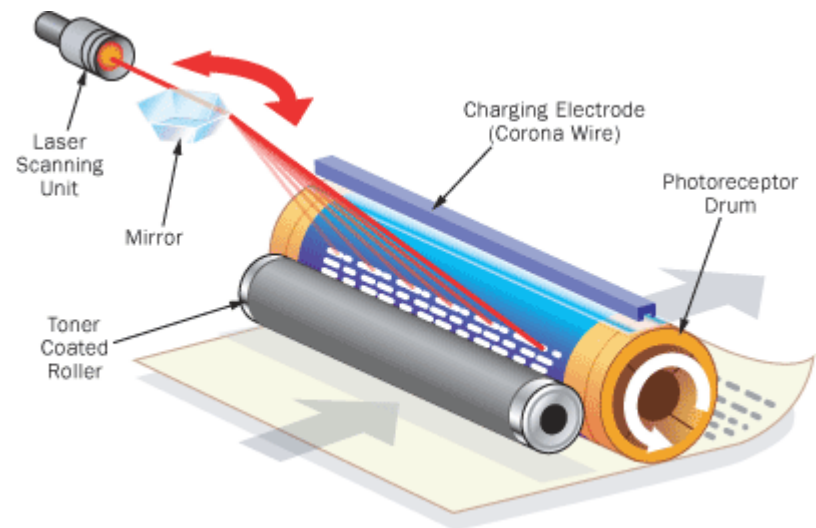
UNITÀ DI OUTPUT: STAMPANTI

Le **stampanti laser**, infine, fanno uso del toner, una polvere finissima contenente carbone e ossido di ferro.

Un rullo fotosensibile viene caricato negativamente. Attraverso un laser si neutralizza tale carica dove il toner non deve attaccare. Il toner, caricato positivamente, verrà attratto dal rullo.

Un foglio di carta viene caricato negativamente (con una intensità maggiore) in modo da catturare il toner.

Il foglio viene fatto passare nel cosiddetto «forno di cottura» fissandone il toner. Quello in eccesso verrà spazzolato via, prima dell'uscita.



UNITÀ DI OUTPUT

Altre periferiche di output sono:

Plotter

Simile al tecnigrafo, viene utilizzato principalmente per stampare su modulo continuo disegni di grandi dimensioni.

Microfilm

Consentono l'archiviazione di grandi quantità di dati sotto forma di immagine.

Altoparlanti

Sono collegati alla scheda audio.



LE PORTE

Seriale

Trasmette un bit alla volta (ha 9 o 25 pin).

Parallela

8 bit alla volta, 25 pin, LPT (**L**ine **P**rin**T**er).

SCSI

8, 16, 32 bit.

USB

Universal serial bus.

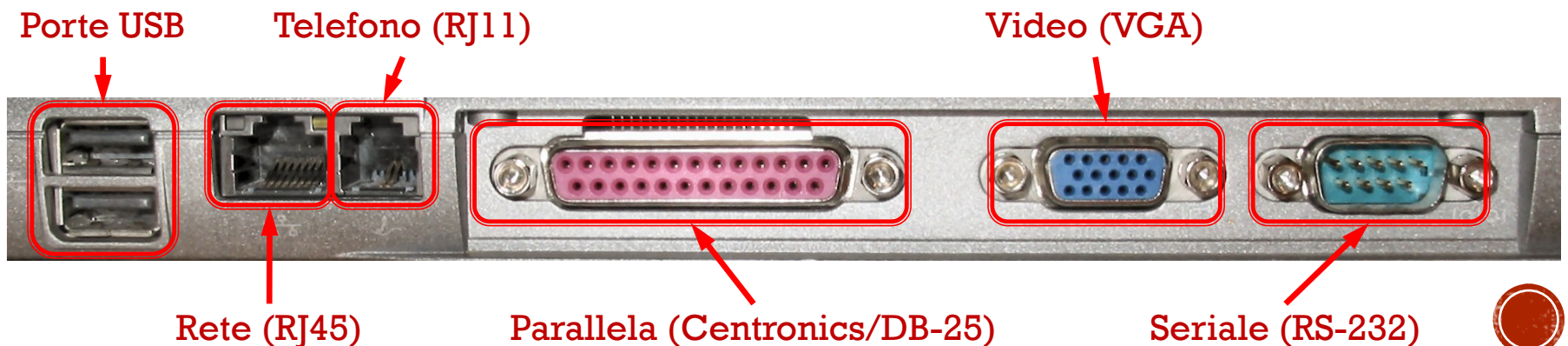
FireWire

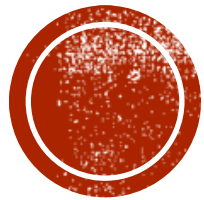
Detta anche IEEE 1394.

IRDA

infrared data association.

WiFi, Bluetooth, HDMI, ...





IL SISTEMA OPERATIVO E GLI APPLICATIVI SOFTWARE

IL SOFTWARE

Insieme di programmi la cui esecuzione permette di creare l'ambiente virtuale con cui l'utente si trova ad interagire.

I programmi eseguibili sono scritti in *linguaggio macchina*. Per linguaggio macchina si intende quel linguaggio di programmazione a basso livello, le cui singole istruzioni sono direttamente eseguibili dalla macchina Hardware e sono formate da stringhe binarie o al più dalla loro «astrazione» attraverso stringhe.

Talvolta si tende a confonderlo con il linguaggio Assembly, che in realtà, pur essendo un linguaggio a basso livello, rimane pur sempre un linguaggio simile a quelli evoluti (C, C++, Java, etc.) che, come vedremo, *richiedono un processo di traduzione*.



IL SOFTWARE



IL SISTEMA OPERATIVO

È un insieme di programmi che gestiscono e coordinano le varie risorse dell'elaboratore, facilitandone l'uso.

Esso costituisce l'interfaccia tra la macchina hardware e l'utente, fornendo una sorta di visione astratta della stessa.

In definitiva, il Sistema Operativo è un insieme di *moduli software* che virtualizzano l'hardware sottostante, creandone una *istanza astratta* più semplice da utilizzare rispetto a quella *fisica*.



IL SISTEMA OPERATIVO

Il Sistema Operativo (SO), principalmente, presenta le seguenti attività funzionali:

- gestione
 - ✓ del processore;
 - ✓ della memoria principale;
 - ✓ delle risorse;
 - ✓ delle informazioni in memoria secondaria (file system);
- configurazione della macchina all'accensione
- macchina estesa
 - ✓ costituisce la base sulla quale scrivere programmi;
 - ✓ mostra una macchina estesa più semplice da programmare rispetto all'hardware sottostante.



IL SISTEMA OPERATIVO

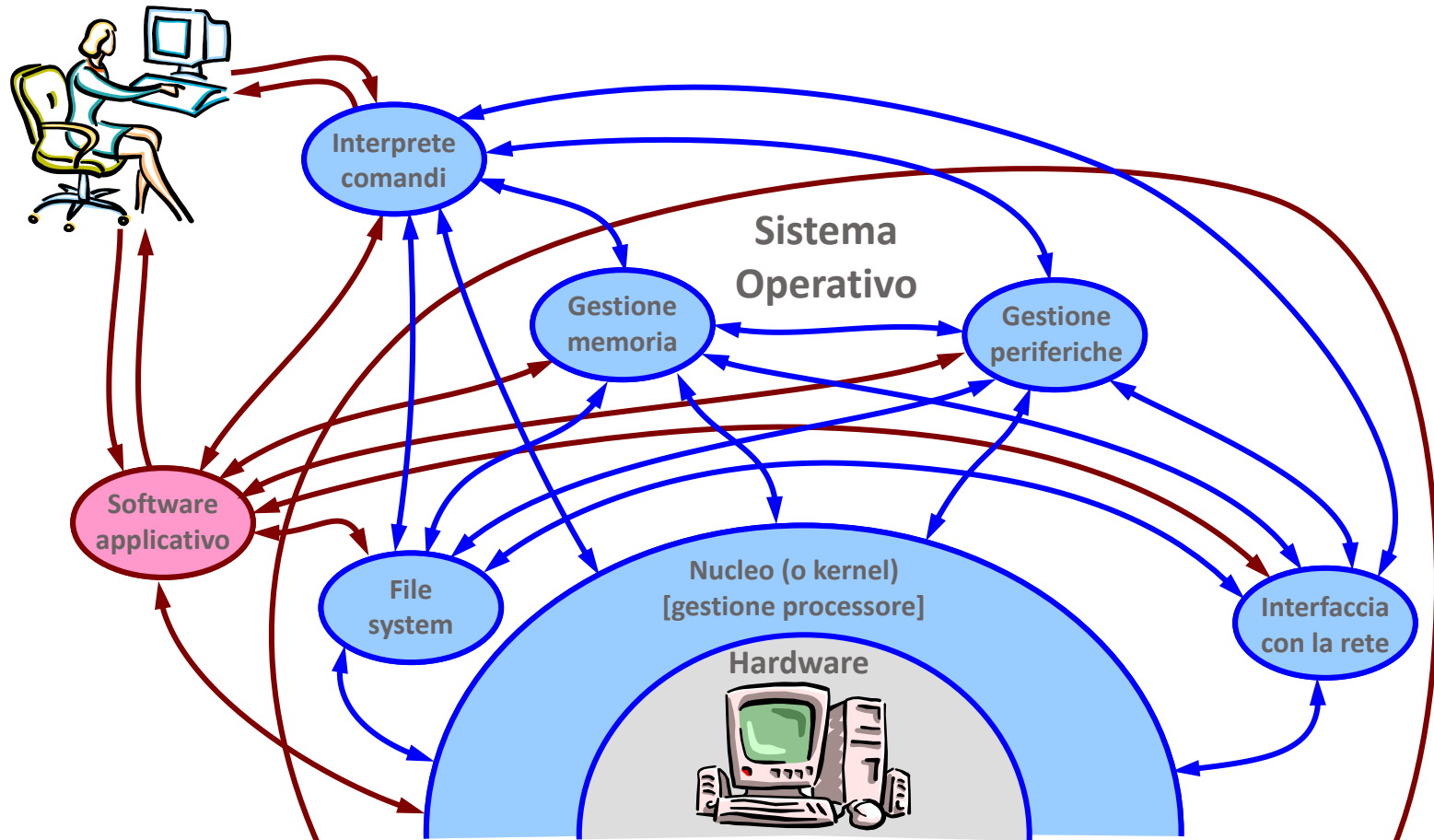
Se si vuole entrare in dettaglio, le funzioni di un SO sono:

- **esecuzione di programmi**
Li carica, insieme ai dati, in memoria centrale, inizializza le periferiche di I/O, prepara e gestisce risorse quali la rete, etc.
- **accesso ai dispositivi di I/O**
Gestisce i segnali nel bus, consentendo all'utente di farne un uso astratto.
- **archiviazione di dati e/o programmi**
Fornisce una organizzazione logica in file, cartelle (directory), gestendo il flusso di basso livello, nel bus, con le periferiche di I/O.
- **controllo di accesso**
Meccanismi di protezione e condivisione delle risorse tra utenti o processi, con relativa gestione dei conflitti.
- **Gestione dei malfunzionamenti**
Rilevare e, dove possibile risolvere, malfunzionamenti all'hardware causati da operazioni errate compiute dai programmi.



IL SISTEMA OPERATIVO

Utente



IL SISTEMA OPERATIVO

Classificazione dei sistemi operativi:

- *mono-utente (mono-user)*
il sistema può essere utilizzato da un solo utente per volta
- *multi-utente (multi-user)*
consente l'interazione contemporanea a più utenti (in questo caso il SO fornisce un sistema astratto dedicato a ciascuno di loro)
- *mono-programmato (mono-task)*
consente l'esecuzione di un solo programma per volta
- *multi-programmato (multi-task)*
come visto, i programmi vengono eseguiti «contemporaneamente» grazie al time sharing (ad ogni task, o processo, viene assegnato un certo numero di clock per l'esecuzione da parte del processore, rimanendo in standby mentre accade la stessa cosa per gli altri task)
- *distribuito (sistemi operativi di rete)*
più elaboratori, locali o remoti, vengono gestiti come un unico sistema al fine di compiere operazioni complesse (calcolo distribuito) o gestire un elevato numero di accessi (cluster in datacenter)



LA FASE DI BOOTSTRAP

In tutti i calcolatori, al momento dell'accensione vengono eseguite una serie di operazioni che si completano con il caricamento in memoria centrale del kernel del SO, prelevandolo dalla memoria secondaria.

Tale fase prende il nome di *bootstrap*. Se nella macchina sono presenti più SO, solitamente è possibile selezionare quale far partire attraverso il boot manager, per poi passare le indicazioni al *boot loader*.

Il *kernel* è la parte del Sistema Operativo che agisce a stretto contatto con l'hardware.



LE GUI

Windows

Nasce intorno alla fine degli anni ottanta, come ambiente operativo grafico per computer con Sistema Operativo MS-DOS.

Prima di allora, si disponeva solo di shell testuali attraverso le quali ogni operazione necessitava di digitare opportuni comandi, con relativi parametri (ove richiesti), con un prompt del tipo `c:\>`

I comandi erano classificati in comandi per la manipolazione di dischi e files, comandi di interazione con le risorse hardware e comandi di utilità per lo sviluppo di programmi.



LE GUI

Windows

L'interfacce grafica, detta GUI (Graphical User Interface), ha reso l'utilizzo dei computer User Friendly.

Alla grafica, si è aggiunta una importantissima categoria di periferiche hardware, ovvero i sistemi di puntamento grazie ai quali selezionare oggetti grafici visualizzati sullo schermo ed interagire con loro.

Ciò che compare a video può essere paragonato ad una scrivania (desktop) sulla quale sono presenti strumenti ed oggetti rappresentati mediante disegni (le icone) che ricordano in modo diretto la funzionalità implicita.



LE GUI

I documenti (files) possono essere raccolti in cartelle (le directory) dando loro dei nomi significativi come si farebbe con una etichetta cartacea attaccata su di una carpetta reale.

Quando si manda in esecuzione un programma, la relativa istanza in esecuzione avrà un ambiente grafico dedicato (la finestra, o windows) dentro il quale si svolgerà l'interazione completa con l'utente. Ogni finestra è dotata di pulsanti per ridurre ad icona, ridimensionare o chiuderla.

Chiudere una finestra equivale ad interrompere l'esecuzione del relativo programma, con il conseguente rilascio delle risorse utilizzate.



GESTIONE FILES

Per assegnare un nome ai files (che di fatto sono una sorta di contenitori logici per i bit che costituiscono dati o programmi), nella maggior parte dei sistemi operativi viene adottata la seguente convenzione:

nome_file.estensione

Nella directory dentro la quale è memorizzato il file, *nome_file* deve essere un identificativo univoco. L'estensione, invece, è una parte facoltativa costituita da un un testo lungo da 1 a 4 caratteri che identifica il contenuto del file stesso.

Esempio:

lettera.docx

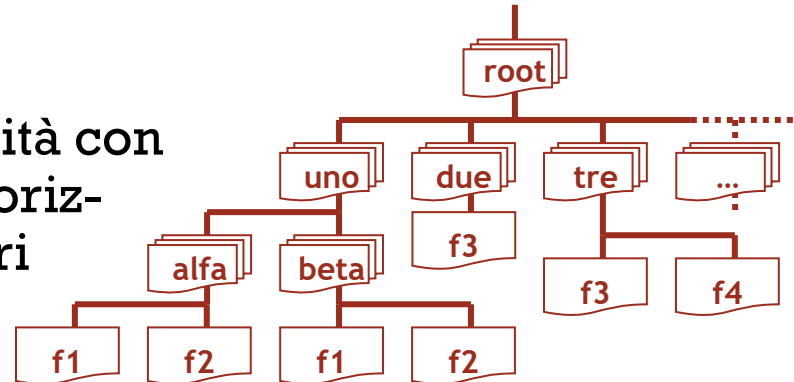
La directory principale è detta anche root e viene creata in automatico al momento della formattazione del disco rigido.



IL FILE SYSTEM

È il meccanismo attraverso il quale vengono organizzati i files all'interno di una memoria di massa. Per essere più formali, è l'insieme dei tipi di dati astratti utilizzati per la scrittura, l'organizzazione gerarchica, la modifica e la lettura dei file sulle memorie di massa.

Il file system organizza la modalità con la quale ciascun file viene memorizzato a livello fisico sui vari settori della memoria di massa.



Ciascun file system, poi, dispone di tabelle per associare ad ogni nome_file il corrispondente collegamento fisico al file stesso, ovvero l'indice correlato in una tabella di allocazione dei file. Non tutti i file system hanno bisogno di una memoria fisica.



SOFTWARE APPLICATIVO

Rappresenta l'insieme di programmi in grado di soddisfare le più svariate esigenze degli utilizzatori.

Le più diffuse classi di programmi applicativi sono:

- Elaboratori di testi (Word processors);
- Fogli elettronici (Spreadsheets);
- Basi di dati (Databases);
- Programmi di grafica ed animazione;
- Programmi multimediali;
- Traduttori (interpreti e compilatori);
- Contabilità aziendale;
- Intrattenimento e gioco;
- Di presentazione
- Desktop publishing (per l'editoria)

Il software applicativo può essere di due tipi:

- General purpose.
- Orientato alla risoluzione di uno specifico problema.



ELABORATORI DI TESTI

Editors di caratteri ASCII

- BloccoNote di Windows.
- Emacs.

Word processors

- Microsoft Word.
- Word-pad di Windows.
- Writer (incluso in Open Office).
- ...

Sistemi di impaginazione basati su testo formattato (il documento viene impaginato in seguito ad una traduzione).

- TeX (o la sua variante LaTeX).
- HTML.



FOGLI ELETTRONICI

Consentono di:

- effettuare elaborazioni matematiche e statistiche su dati raccolti in forma tabellare;
- realizzare grafici di vario tipo;
- filtrare i dati da elaborare/visualizzare;
- Formattare graficamente i dati gestiti.

Principali programmi commerciali:

- Lotus;
- Microsoft Excel;
- Calc (incluso in Open Office);
- ...



BASI DI DATI

Le basi di dati servono a raccogliere ed a gestire in maniera organizzata, enormi quantità di dati. Più avanti, vedremo meglio il concetto di dato ed i benefici apportati da una loro gestione attraverso un DBMS.

Basi di dati sul modello relazionale:

- Dbase;
- Microsoft Access;
- Oracle;
- SQL Server;
- ...



GRAFICA ED ANIMAZIONE

Consentono di creare o di elaborare immagini ed animazioni in maniera interattiva (ad esempio “foto-ritocco”).

Alcuni esempi di software appartenente a tale categoria:

- Photoshop.
- CorelDraw.
- Paint shop.
- Blender (di tipo Open Source).
- ...



PROGRAMMI MULTIMEDIALI

L'interazione multimediale è la mescolanza di:

- suoni;
- immagini;
- video;
- iper-testi.

L'uso di strumenti multimediali si è diffuso in modo significativo con Internet. Programmi per la creazione di siti Web:

- Flash;
- DreamWeaver;
- Webmatrix (gratuito);
- ...



TRADUTTORI

Quando un programmatore realizza un software utilizzando un linguaggio di programmazione di alto livello, di fatto realizza un file di testo chiamato *codice sorgente* (o semplicemente *sorgente*), seguendo le regole sintattiche e semantiche previste dal linguaggio stesso.

Un programma viene progettato a partire da un particolare problema che si intende risolvere, trasformandolo in un algoritmo (eventualmente ricorrendo ad un diagramma di flusso).

Di fatto, un file sorgente conterrà delle dichiarazioni di variabili e/o costanti (e loro inizializzazioni), funzioni, istruzioni e strutture di controllo del flusso di esecuzione, il tutto in accordo con il paradigma di programmazione scelto (dal quale dipende, in conseguenza, il linguaggio di programmazione).



TRADUTTORI

Per poter essere eseguito, il file sorgente dovrà essere tradotto in *codice oggetto* (o *file oggetto*), scritto in linguaggio macchina (quindi in binario).

Il procedimento di traduzione, (in un determinato caso anche attraverso il link alle varie librerie software utilizzate), arriverà a produrre un file eseguibile (in windows con estensione .exe) che potrà essere avviato con un doppio clic sulla relativa icona, in modo da fruire del programma realizzato.

Il file eseguibile, a quel punto, potrà essere distribuito in altri computer con caratteristiche simili a quello per il quale è stato sviluppato.



TRADUTTORI

La distribuzione del software proprietario (closed source), tuttavia, è soggetta a particolari forme contrattuali che prendono il nome di *licenze d'uso*.

Esistono varie forme di licenza. Giusto per citare alcune delle più importanti:

- **EULA**

End-User License Agreement (EULA) è un contratto attivato tra l'autore del software e l'utente finale che assegna la licenza d'uso del software secondo i termini stabiliti.

- **SHAREWARE**

Il software viene distribuito liberamente e può essere utilizzato gratuitamente in versione di prova o per un periodo di tempo limitato, o con funzionalità ridotte. È un modo molto diffuso tra chi desidera valutare se procedere con una EULA.

- **FREEWARE**

Il software può essere distribuito gratuitamente secondo le disposizioni dell'autore.

