

# APPUNTI di Tecnologie Informatiche

<b>Paragrafo 1:</b>	<b>bit e Byte</b>
<b>Paragrafo 2:</b>	<b>il codice ASCII</b>
<b>Paragrafo 2bis:</b>	<b>la rappresentazione dei numeri negativi e le operazioni algebriche fondamentali</b>
<b>Paragrafo 3:</b>	<b>il processore</b>
<b>Paragrafo 4:</b>	<b>la memoria di sistema</b>
<b>Paragrafo 5:</b>	<b>Hard Disk</b>
<b>Paragrafo 6:</b>	<b>CD e DVD</b>
<b>Paragrafo 7:</b>	<b>unità video</b>
<b>Paragrafo 8:</b>	<b>stampanti</b>

## 1. Bit e Byte

L'unità di misura dell'informazione per l'elaboratore è il **bit**, cioè una cifra binaria.

Poiché con un solo bit è possibile rappresentare due sole informazioni (vero o falso, oppure alto o basso, ecc), per rappresentare un numero superiore di informazioni è necessario utilizzare una sequenza ordinata di bit.

Analizziamo come il numero di bit determini il numero di combinazioni.

Con  $n = 2$  bit si possono avere 4 combinazioni (00,01,10,11)

Con  $n = 3$  bit si possono avere 8 combinazioni (000,001,010,011,100,101,110,111)

**in generale con  $N$  bit si possono ottenere  $2^N$  combinazioni !**

In particolare sono usate le seguenti sequenze:

**semibyte (o nibble)** è una sequenza di quattro bit e si possono ottenere  $2^4 = 16$  combinazioni

**byte o otetto** è una sequenza di otto bit e i possono ottenere  $2^8 = 256$  combinazioni

**parola ( word)** è una sequenza di due byte.

Nei computer il microprocessore e la memoria di sistema RAM utilizzano come dato **parole di 4 byte** (quindi i dati sono a **32 bit**) o di **8 byte** (ovvero trattano dati a **64 bit**).

Il **bit** si indica con la lettera **b**. Il **byte** si indica con la lettera **B**.

Quindi con un **semibyte** (cioè 4 bit), mentre con **1 byte** (cioè 8 bit).

### CURIOSITA'

Il simbolo utilizzato per il byte come *unità di misura della quantità di informazione* è **B** (identico al simbolo del *bel*); la lettera maiuscola sarebbe riservata alle sole unità di misura tratte dai cognomi degli ideatori, ma la *Commissione Elettrotecnica Internazionale* (IEC) ha deciso di fare un'eccezione dato che **b** è generalmente usato per indicare il *bit* (il cui simbolo standard sarebbe *bit per esteso*).

Per comodità di calcolo i multipli del byte vengono generalmente arrotondati a **potenze di 2** (benché questo sia formalmente sbagliato), invece che di 10; tale ambiguità ha portato l'IEC a definire nuovi **prefissi per multipli binari**; tali valori non sono comunque entrati nell'uso comune.

- Riepilogo:
- 4 bit = metà di un byte, formano un **nibble**.
  - 2 byte = 16 bit, sono chiamati anche **word**.
  - 4 byte = 32 bit = 2 word, sono chiamati anche **double word**.
  - 8 byte = 64 bit = 2 double word, sono chiamati anche **quad word**.

È importante sottolineare che i valori di *word*, *double word* e *quad word* qui riportati sono indicativi e fanno riferimento ad una architettura di computer in cui **una word** è uguale a **2 byte**.

I valori dei multipli del Byte rispetto ad esso

**Bit (b):** 1 bit (=1/8 B)

**Byte (B):** 8 bit

**Kilobyte (kB):** 1 024 B

**Megabyte (MB):** 1 024 kB, 1048576 B

**Gigabyte (GB):** 1 024 MB, 1048576 kB, 1 073 741 824 B

**Terabyte (TB):** 1024GB, 1 099 511 627 776 B

Consideriamo il byte l'unità di base: 1024 volte un byte equivale ad 1 kilobyte. 1048576 volte un byte equivale ad 1 megabyte, espresso anche in 1024 kilobyte. Lo stesso vale per i gigabyte, dove a 1024 corrispondono i megabyte, a 1 048 576 i kilobyte, mentre i byte corrispondenti sono 1 073 741 824. Anche per i terabyte è lo stesso principio: 1024 sono i gigabyte, i megabyte sono 1 048 576, i kilobyte sono 1 073 741 824, ed in byte sono invece 1 099 511 627 776.

1 b = 1/8 byte --- (binary digit)

1 B = 1 byte --- (byte)

1 kB = 1 024 B --- (kilobyte) --- mille

1 MB = 1 048 576 B --- (megabyte) --- milione

1 GB = 1 073 741 824 B --- (gigabyte) --- miliardo

1 TB = 1 099 511 627 776 B --- (terabyte) --- trilione

Esercizi:

1. Costruite una tabella dove riportiamo tutte le combinazioni nel caso in cui si abbia 4 e 5 bit
2. avere un processore a 32 bit significa che tratta dati composti da ..... byte
3. avere un processore a 64 bit significa che tratta dati composti da ..... byte
4. se una memoria è composta da 1KB, quanti byte contiene effettivamente ?
5. se una memoria è composta da 2KB, quanti byte contiene effettivamente ?
6. se una memoria è composta da 8KB, quanti byte contiene effettivamente ? E quante word?
7. se una memoria è composta da 1MB, quanti byte contiene effettivamente ? E quante word?
8. se una memoria è composta da 16MB, quanti byte e word contiene effettivamente ?
9. Se una pendrive ha una capacità di 1GB, quanti byte contiene effettivamente ?
10. Un Hard disk contiene 1TB di dati e ha una capacità di 2TB. Quanti filmati da 700MB può ancora memorizzare?
11. Un file di 82.554 KB viene trasmesso a una velocità di 12500 bps. Quanto tempo è necessario per il trasferimento completo del file?
12. 1KB e 1000B hanno lo stesso valore? Se no, qual è superiore? Spiega il perché.

## 2. Il codice ASCII

I computer trattano dati di tipo numerico ma anche testuale. Ad esempio un testo scritto con Wordpad presenta parole composte da lettere, magari in alfabeti diversi se si usano lingue diverse, e anche i numeri riportati potrebbero non avere un significato matematico, ma solo essere un codice mnemonico a scopo identificativo. Per intenderci un numero di telefono non è un dato aritmetico con cui fare delle operazioni aritmetiche, ma viene trattato come un testo, quindi eventualmente ordinato in senso crescente o decrescente, ecc...

Il problema è allora il seguente: come codificare, usando il solo codice binario, dati di tipo profondamente diverso?

Si parla infatti di caratteri alfanumerici intendendo:

- le lettere dell'alfabeto, distinguendo le minuscole e le maiuscole
- le dieci cifre da "0" a "9" con le quali costruiamo i numeri decimali
- i simboli speciali, tra i quali quelli di interpunzione (la punteggiatura!) e i segni grafici usati in matematica (le parentesi, simboli che indicano le operazioni matematiche, ...)
- i simboli grafici, linee, archi, gli emoticon,...
- infine esistono caratteri speciali che non sono visibili, ma servono per ordinare il testo, come ad esempio la spaziatura, l'invio, ecc.

Per rappresentare tutto questo è stato necessario definire una regola generale, che stabilisse una corrispondenza fra ciascun carattere e il codice binario corrispondente.

La prima soluzione storica fu il codice **ASCII**, che è l'acronimo di **American Standard Code for Information Interchange** (ovvero *Codice Standard Americano per lo Scambio di Informazioni*).

La tabella ASCII è un codice convenzionale usato per la rappresentazione dei caratteri di testo attraverso i **byte**: ad ogni byte viene fatto corrispondere un diverso carattere della tastiera (lettere, numeri, segni).

In realtà lo standard ASCII classico copre solo i primi 128 byte (da 00000000 a 01111111) ottenuti tramite la permutazione dei primi sette bit, i successivi byte fino al 256° costituiscono la *tabella ASCII estesa*, definito su otto bit e quindi con 256 possibili combinazioni, che presenta varie versioni a carattere nazionale.

Nella [tabella ASCII standard](#) si trovano le cifre numeriche, le lettere maiuscole e minuscole (maiuscole e minuscole hanno codici ASCII differenti) la punteggiatura, i simboli aritmetici e altri simboli (\$, &, %, @, #, ecc.). Essendo stata concepita in America, la tabella ASCII standard non comprende le lettere accentate (sconosciute all'ortografia inglese). I primi 32 byte della tabella standard sono inoltre riservati per segnali di controllo e funzioni varie.

L'alfabeto latino, usato nella scrittura di molte lingue nel mondo, presenta una grande quantità di varianti grafiche: si va dalle semplici vocali accentate (accento grave à, acuto á, circonflesso â, dieresi ä, tilde ã) a lettere modificate (lettere con barrette, cediglie, segni), lettere speciali usate solo in una lingua, segni di punteggiatura particolari (il punto interrogativo ed il punto esclamativo capovolti usati nello spagnolo), simboli di valuta, e così via, senza considerare poi che gran parte di questi segni presentano le due forme maiuscola e minuscola. Le varianti sono talmente numerose che i 128 Byte della tabella estesa non sono purtroppo sufficienti a rappresentarle tutte, per questo motivo esistono diverse estensioni della tabella ASCII: lo standard ISO 8859 prevede 15 diverse estensioni, comprese quelle per gli alfabeti diversi dal latino, ma esistono anche ulteriori estensioni non riconosciute dall'ISO e create per esempio dalla Microsoft per i sistemi Windows o dalla Apple per i Macintosh.

La **tabella ASCII estesa** tipicamente utilizzata in Italia è quella dell'Europa occidentale, creata per le lingue germaniche e neolatine (escluso il rumeno). Altre estensioni usate in Europa sono la Centro Europea per i paesi dell'Europa orientale (lingue slave, ungherese, rumeno), la Turca, la Cirillica e la Greca. Questa coesistenza fra diverse versioni del codice ASCII produce spesso discordanze nella visualizzazione dei file di testo. Sarà capitato a molti di aprire un file di testo o ricevere una E-mail e trovare segni assurdi al posto di tutte le lettere accentate, questo perché chi l'ha scritto stava usando una tabella estesa diversa dalla vostra e quindi il vostro computer interpreta alcuni byte del file in modo diverso. Certi tipi di file, come i file html, possono contenere al loro interno il nome esplicito dell'estensione ASCII usata per la loro creazione, così il computer ricevente saprà come regolarsi.

Per cercare di ovviare al problema è stato creato un nuovo standard internazionale detto *Unicode*, definito dalla Unicode Consortium e dalla International Organization for Standardization (ISO 10646), che rappresenta i caratteri usando 2 byte. Con 2 byte il numero di combinazioni possibili diventa  $256 \times 256 = 65.536$ , perciò Unicode supporta 65.536 diversi segni, al posto dei 256 del set ASCII. Si riescono così a rappresentare non solo tutte le varianti dell'alfabeto latino, ma anche tutti gli altri alfabeti (greco, cirillico, arabo, ebraico, hindi, thai,...) oltre all'insieme degli ideogrammi cinesi e giapponesi (che sono in tutto circa 30.000, anche se poi ne vengono effettivamente utilizzati solo poche migliaia).

**Lo svantaggio dell'Unicode, rispetto all'ASCII, è che le dimensioni dei file di testo risultano comunque raddoppiate (vengono usati 2 byte per carattere, invece di 1 solo).**

Se si sta usando Windows si può ottenere ogni carattere ASCII tenendo premuto il tasto **Alt** e digitando il codice decimale corrispondente col [tastierino numerico](#) (se il tastierino numerico non fosse attivo, premere prima il tasto *Num lock* o *Bloc Num* per attivarlo. Per esempio la chiocciola @ si ottiene digitando 64 mentre si tiene premuto il tasto Alt.

Nella tastiera inglese sono già presenti tutti i caratteri della tabella standard; nella tastiera italiana invece mancano l'apice (96), le parentesi graffe (123,125) e la tilde (126).

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	~

## 2bis. La rappresentazione dei numeri negativi e le operazioni algebriche fondamentali

### COMPLEMENTO A 1

**Per trovare la rappresentazione in complemento a uno si invertono semplicemente tutti i bit della parola.**

Facciamo un esempio. Prendiamo il numero 10 rappresentato su 8 bit in base 2: 0000 1010 (10)

Il suo complemento a uno sarà: 1111 0101 (-10)

Per ottenere invece il valore in decimale dal numero negativo si procede così: dopo avere verificato che è negativo, basta negare (invertire) i bit e ricostruire il valore decimale corrispondente:

se si parte dalla sequenza 11110101 basta invertire di nuovo i singoli bit:

0000 1010 ovvero il numero in modulo vale +10 ed essendo negativo, vale in realtà -10

### Rappresentazione dello zero in complemento a uno

Esistono in *complemento a uno* due differenti rappresentazioni del numero zero, quella con tutti zero e quella con tutti uno:

0000 0000 (0+) e 1111 1111 (0-)

Tale doppia codifica rappresenta uno "spreco" di informazione e quindi il metodo in complemento a 1 non è di fatto utilizzato per la rappresentazione dei numeri negativi al calcolatore.

### COMPLEMENTO A DUE

Il **complemento a due** (*two's complement*) è il metodo più diffuso per la rappresentazione dei numeri negativi in *informatica*. La sua enorme diffusione è data dal fatto che i circuiti di addizione e sottrazione non devono esaminare il segno di un numero rappresentato con questo sistema per determinare quale delle due operazioni sia necessaria, permettendo tecnologie più semplici e maggiore precisione; si utilizza un solo circuito, il *sommatore*, sia per l'addizione che per la sottrazione.

Col complemento a due, il bit iniziale (più a sinistra) del numero definisce il segno (negativo o positivo) del numero stesso; da questo deriva che tutti i numeri che cominciano con un "1" sono *numeri binari* negativi, mentre tutti i numeri che cominciano con uno "0" sono *numeri binari* positivi. Si può così ottenere il valore assoluto di un numero binario negativo, complementando (invertendo) il valore dei singoli bit e aggiungendo 1 al numero binario risultante.

Un numero binario di 8 bit può rappresentare con tale metodo i numeri compresi fra -128 e +127.

**Questo metodo consente di avere un'unica rappresentazione dello zero** (quando tutti i bit sono zero, eliminando così la ridondanza dello zero che si verifica con la rappresentazione in *modulo e segno*), e di operare efficientemente addizione e sottrazione sempre avendo il primo bit a indicare il segno.

### Calcolo dell'opposto in complemento a due

Per rappresentare l'opposto di un numero binario in complemento se ne invertono, o negano, i singoli bit. Si aggiunge infine 1 al valore del numero trovato con questa operazione.

Facciamo un esempio rappresentando il numero -5 con 8 bit in complemento a 2:

- Partiamo dalla rappresentazione in binario del numero 5: 0000 0101 (5)
- Invertiamo i bit: 0 diventa 1, e 1 diventa 0: 1111 1010
- A questo punto abbiamo ottenuto il *complemento a uno* del numero 5; per ottenere il complemento a due aggiungiamo 1 a questo numero: 1111 1011 (-5)

Il risultato è un numero binario con segno che rappresenta il numero negativo -5 secondo il complemento a due. Il primo bit, pari a 1, evidenzia che il numero è negativo.

Verifica dell'operazione inversa:

Il complemento a due di un numero negativo ne restituisce il numero positivo pari al valore assoluto: invertendo i bit della rappresentazione del numero -5 (sopra) otteniamo: 0000 0100

Aggiungendo 1 otteniamo: 0000 0101

Che è appunto la rappresentazione del numero +5 in forma binaria.

Si noti che il complemento a due dello zero è zero stesso: invertendone la rappresentazione si ottiene un byte di 8 bit pari a 1, e aggiungendo 1 si ritorna a tutti 0 (l'overflow viene ignorato).

### ADDIZIONE

Operare l'addizione di due interi rappresentati con questo metodo non richiede processi speciali se essi sono di segno opposto, e il segno viene determinato automaticamente. Facciamo un esempio addizionando 15 e -5:

```

1111 1110 (riporto)
 0000 1111 (15)
+ 1111 1011 (-5)
=====
 0000 1010 (10)
    
```

Questo processo gioca sulla lunghezza fissa di 8 bit della rappresentazione: viene ignorato un riporto di 1 che causerebbe un overflow, e rimane il risultato corretto dell'operazione (10).

Gli ultimi due bit (da destra a sinistra), ovvero i più significativi, della riga dei riporti contengono importanti informazioni sulla validità dell'operazione: se il risultato è compreso o non è compreso nell'intervallo dei numeri rappresentabili. Si verifica se il riporto è stato eseguito sul bit del segno ma non è stato portato fuori, o viceversa; più semplicemente, se i due bit più a sinistra sulla riga dei riporti non sono entrambi 0 o 1. Per verificare la validità del risultato è conveniente eseguire su questi due bit un'operazione. Vediamo un esempio di addizione a 4 bit di 7 e 3:

```

01110 (riporto)
 0111 (7)
+ 0011 (3)
=====
 1010 (-6)
    
```

In questo caso, come si può notare dal riporto presente solo sul bit più significativo, si è in presenza di *overflow*, per cui il risultato non è 10 (come sarebbe corretto) ma -6, infatti il massimo numero positivo rappresentabile in complemento a due su quattro bit è 7 (con  $n=4$ :  $2^{n-1} - 1 = 7$ ).

### SOTTRAZIONE

Anche se la sottrazione potrebbe essere eseguita aggiungendo il complemento a due del sottraendo al minuendo, questo procedimento è poco utilizzato in quanto porta più complicazioni che semplicemente costruire un circuito per la sottrazione. Ma come per l'addizione, il vantaggio del complemento a due è l'eliminazione della necessità di esaminare i segni degli operandi per determinare quale operazione sia necessaria. Per esempio, sottrarre -5 a 15 è come aggiungere 5 a 15, ma questo è nascosto dal complemento a due:

```

1111 0000 (riporto)
 0000 1111 (15)
- 1111 1011 (-5)
=====
 0001 0100 (20)
    
```

L'overflow viene individuato con lo stesso metodo usato per l'addizione, esaminando i due bit più a sinistra sulla riga dei riporti: se sono differenti si è verificato un overflow.

Facciamo un altro esempio con una sottrazione con risultato negativo:  $15 - 35 = -20$ :

```

1110 0000 (riporto)
0000 1111 (15)
- 0010 0011 (35)
=====
1110 1100 (-20)
    
```

**PARTICOLARITÀ**

A parte una singola eccezione, cercando il complemento a due di ogni numero rappresentato con questo metodo, otteniamo il suo opposto: 5 diventa -5, -12 diventa 12, ecc.

Il minor numero rappresentabile (cioè quello negativo con maggior valore assoluto) costituisce l'unica eccezione: vediamo l'esempio del numero -128 nella rappresentazione a 8 bit:

```

1000 0000 (-128)
0111 1111 (bit invertiti)
1000 0000 (aggiunto 1)
    
```

Questo perché 127 è il maggior numero con segno rappresentabile con 8 bit. Si noti che viene segnalato un overflow perché c'è un riporto sul bit del segno ma non fuori di esso.

Nonostante questo sia un numero unico, la sua rappresentazione è valida. Tutte le operazioni possono funzionare con esso sia come operando che come risultato (a meno che non sia successo un overflow)

### 3. Il processore

La CPU ( Central Processing Unit ) è il vero cervello del computer. Per descrivere le prestazioni di un microprocessore si utilizzano i seguenti parametri fondamentali:

Parametro microprocessore	Descrizione
Velocità	La velocità viene espressa in due modi: <ul style="list-style-type: none"> <li>tramite la frequenza del segnale detto <b>clock</b> [GHz]</li> <li>oppure tramite il <b>numero di istruzioni al secondo</b> [MIPS]</li> </ul> attualmente i processori hanno raggiunto frequenze di 3,6 Ghz mentre circa 15 anni fa la velocità era solo di 25MHz.
Numero di bit di dato (parallelismo del bus di dati)	Indica il numero di bit utilizzato dal processore per trattare i dati. Attualmente i processori più recenti sono orientati a <b>trattare dati a 64bit</b>
Numero di bit per gli indirizzi (parallelismo del bus degli indirizzi)	Il numero di bit riservati all'indirizzo è determinante per stabilire quante celle di memoria può colloquiare il processore. Si deve tenere conto infatti che ogni cella di memoria, così come ogni periferica ha un indirizzo, e che tale indirizzo permette al processore di potere colloquiare con essa. Ciò che non ha indirizzo non è "visto" dal processore e dunque non è utilizzabile dall'utente.
Presenza di coprocessori e memoria cache	La velocità di esecuzione delle istruzioni migliora se sono presenti nel processore delle memorie di supporto dette " <b>cache memory</b> " e se è presente un ulteriore processore dedicate alle sole operazioni matematiche (coprocessore matematico). Un fattore di qualità è determinato proprio dalla quantità di memoria cache nei processori. Ad esempio i processori <b>Celeron</b> della <b>INTEL</b> e <b>Duron</b> della <b>AMD</b> sono dotati di piccola memoria cache

All'interno del microprocessore (CPU) possiamo individuare quattro aree distinte che hanno funzioni diverse:

- il **blocco ALU** , il **blocco UC**, **registri interni** , la **cache memory interna**

il blocco UC (Unit control) genera i segnali di controllo per stabilire il corretto funzionamento di tutte parti del sistema computer . Ad esempio il blocco UC deve generare i segnali per:

- trasferire i dati dalla memoria permanente (HDisk) alla memoria centrale
- traferire i dati interessati dalla memoria centrale alla ALU
- comandare la ALU in modo che esegua l'operazione voluta. Il tipo di operazione è stabilita da una **istruzione**
- prelevare il risultato finale e trasferirlo alla memoria centrale in una nuova cella di memoria di sistema
- presentare il risultato finale all'utente usando un dispositivo di uscita
- eventualmente trasferire il risultato finale dalla memoria di sistema a una cella di presente nella memoria permanente (Hdisk)

il blocco ALU (Arithmetic Logic Unit) ha il compito di svolgere operazioni di tipo matematico (Arithmetic) e logico , tra cui possiamo citare ad esempio il confronto fra dati.

il blocco ALU comunica con la memoria centrale , da dove sono presi i dati su cui lavorare, e il blocco UC, che provvede a pilotare il funzionamento della ALU

i registri interni sono dei blocchi di memoria di tipo RAM, che hanno velocità pari al processore stesso, in cui sono contenuti le istruzioni e i dati parziali e più utilizzati

ATTUALMENTE le case costruttrici INTEL E AMD sono le principali nel campo delle CPU. I modelli di processori più diffusi sono INTEL CORE 2 Duo e AMD ATHLON 64 DUAL CORE

## 4. La memoria di sistema

La funzione della memoria centrale è quella di memorizzare i programmi eseguiti dal microprocessore e i dati elaborati.

I parametri caratteristici della memoria di sistema sono i seguenti:

capacità	Rappresenta il numero di byte , e quindi di bit, totali contenuti nella memoria . La capacità in Byte ci aiuta a comprendere qual'è la massima quantità di informazione che può essere contemporaneamente presente in memoria e quindi elaborata dal microprocessore
Tempo di accesso	Indica il tempo necessario per accedere ai dati per leggerli o per scriverli. Se i due tempi sono diversi, allora si prende come riferimento il tempo peggiore
Operazioni disponibili	Le operazioni possibili possono essere : <ul style="list-style-type: none"> <li>• di sola lettura</li> <li>• di scrittura e lettura</li> </ul>
Tipo	Può essere <b>volatile</b> o <b>non volatile</b> si dice volatile se tutto il suo contenuto è cancellato dopo che si è spento il PC si dice non volatile se il suo contenuto rimane intatto anche dopo lo spegnimento dell'elaboratore



La memoria principale è suddivisa in tre aree ben distinte, che hanno caratteristiche e funzioni diverse

Tipo Memorie	Caratteristiche fisiche	Funzione
ROM	<b>Read Only Memory.</b> È una memoria a sola lettura, non volatile, con elevati tempi di accesso. Il contenuto della memoria è scritto dalla ditta costruttrice e non può essere modificato.	Contiene il programma di avvio del calcolatore e di gestione dell'hardware, chiamato BIOS. Il BIOS è scritto e realizzato dal costruttore, non viene creato dall'utente né viene installato.
EEPROM	<b>Electrically erasable programmable ROM</b> E' una memoria a scrittura e lettura, non volatile, con tempi di accesso maggiore della RAM. L'utente può modificare il contenuto della memoria, ma solo un numero limitato di volte (fino a un milione di riscritture).	Contiene tutti i dati caratteristici del computer. Tali dati sono modificabili, ma è bene farlo raramente e solo quando si è esperti. Tale modifica può essere eseguita durante la fase di avvio ma prima del caricamento del sistema operativo.
RAM	<b>Random Access Memory:</b> memoria ad accesso diretto. E' una memoria di scrittura e lettura, dunque è bidirezionale, ed è di tipo volatile, con bassi tempi di accesso (decine di nanosec). Il tempo per scrivere o leggere una cella non dipende dalla posizione della cella, questo assicura dei tempi medi molto bassi.	Contiene i dati e i programmi eseguiti dalla CPU. Essendo la RAM volatile, il PC deve possedere una memoria permanente per archiviare i dati allo spegnimento della macchina . Se si aprono più applicazioni la memoria RAM viene sempre più occupata , fino al caso limite di una totale occupazione. In questo caso il sistema operativo segnala la impossibilità di procedere con nuove applicazioni.

### La necessità di aumentare la velocità della RAM: nascita della cache memory

L'utilità della **memoria cache** (=ripostiglio) nasce dal fatto che *il processore* ha una velocità di elaborazione molto elevata (dell'ordine dei **GHz**, quella del **clock**) e decisamente superiore a quella della memoria di sistema.

Quando la **veloce** CPU è chiamata ad elaborare dati è, quindi, **costretta ad aspettare** che questi arrivino dalla sua memoria esterna di sistema; in questo modo le prestazioni complessive degradano inevitabilmente.

Per questo è stata inventata la **cache memory**, che trova posto tra il processore e la memoria RAM; si tratta di una memoria di piccole dimensioni ma particolarmente veloce; la sua velocità può variare infatti da quella di clock a valori comunque superiori a quella della Ram, che è la parte più veloce della memoria di sistema.

In questo modo, almeno nell'immediato e con sufficiente probabilità, la CPU (il microprocessore) troverà nella cache i dati necessari in seguito, senza dover attendere troppo. Naturalmente prima o poi capiterà che il dato o la istruzione richiesti non siano nella cache: in questo caso la CPU sarà comunque costretta ad accedere normalmente alla meno veloce RAM di sistema.

Una delle ragioni della diversa velocità dei due tipi di memoria è di tipo tecnologico:

la **cache memory** è nota come **SRAM** (RAM Statica); per questo è molto veloce (meno di **2ns**, più di 500MHz) ma relativamente piccola (il valore tipico è di **256kBytes**÷**512kBytes**, fino ad un massimo di **2MBytes**) e più costosa. Il microprocessore comunica con la cache memory tramite un percorso fisico (chiamato bus) dove la velocità consentita è molto alta : la frequenza vale 500 MHz, cioè corrisponde a tempi tipici di  $T = 1/f = 2\text{nsecondi}$ )

la **memoria centrale** di sistema è costituita in gran parte dalla RAM ( per antonomasia, molto semplice dal punto di vista costruttivo; è nota come **DRAM**(RAM Dinamica) e, ha una elevata capacità ( 128MB, 256MB, 512 MB.....).

Il microprocessore comunica con la memoria centrale tramite un percorso fisico ( chiamato bus) dove la velocità consentita è minore rispetto alla cache memory: la frequenza vale al massimo 233MHz, cioè corrisponde a tempi tipici di  $T = 1/f = 3,75$  nanosecondi).

Fin dalle prime architetture è stata prevista la presenza di cache memory tra CPU e memoria di sistema, direttamente sulla scheda madre e, data la grande efficienza di questo meccanismo, **ben presto si è pensato di introdurre parte di essa addirittura dentro il processore;**

per velocizzare lo scambio di dati tra memoria e processore sono oggi disponibili:

la **cache di 1° livello**, non grandissima (da 8kB fino a 128kB) **ma funzionante con la stessa velocità (clock)** del processore che la ospita. La cache di primo livello si trova direttamente dentro il processore!

la **cache di 2° livello**, è posta usualmente in un chip diverso dal processore ed ha una dimensione tipica è di 256kB.

Riassumendo:

**La presenza della Cache Memory rende molto più rapida l'esecuzione dei programmi, dato che la CPU trova con buona probabilità i bytes che gli servono direttamente dentro se stessa (Cache di 1° livello) e nelle sue immediate vicinanze (Cache di 2° livello), senza essere costretta a perdere tempo per indirizzare ed aspettare risposta dalla lenta Ram convenzionale esterna.**

Due considerazioni finali:

- l'aumento della cache produce aumento di prestazioni, anche se non direttamente proporzionale e comunque subordinato al caso (per altro piuttosto probabile) di contenere dentro di sé i dati necessari alla CPU.
- Poiché i suoi costi sono intrinsecamente elevati, la tendenza dei produttori è quella di rendere le frequenze del bus (e delle memorie DRAM convenzionali) sempre più simili a quella di clock.

## 5. HARD DISK

### DESCRIZIONE

Il disco rigido è costituito fondamentalmente da uno o più piatti in rapida rotazione, realizzati in alluminio o vetro, rivestiti di materiale ferromagnetico e da due testine per ogni disco (una per lato), le quali, durante il funzionamento "volano" alla distanza di poche decine di [nanometri](#) dalla superficie del disco leggendo e scrivendo i dati. La testina è tenuta sollevata dall'aria mossa dalla rotazione stessa dei dischi che può superare i 15.000 giri al minuto; attualmente i valori standard di rotazione sono 4.200, 5.400, 7.200, 10.000 e 15.000 giri al minuto (root per minute).

### CARATTERISTICHE E PRESTAZIONI

Questi dischi rigidi moderni hanno capacità e prestazioni enormemente superiori a quelle dei primi modelli ma poiché nel frattempo la velocità e le prestazioni delle memorie ad accesso casuale (RAM e ROM) sono aumentate molto di più, la loro velocità nella lettura e scrittura dei dati restano comunque di diversi ordini di grandezza, al di sotto delle prestazioni della RAM. Per questo motivo il disco rigido è spesso la causa principale del rallentamento di un computer soprattutto quando, a causa di una memoria RAM inferiore alla [memoria virtuale](#) richiesta dai programmi in esecuzione, il sistema operativo è costretto ad effettuare un gran numero di operazioni di *swap* tra il disco e la memoria centrale. Le caratteristiche principali di un disco rigido sono:

- la capacità
- il tempo di accesso
- la velocità di trasferimento

La **capacità** è espressa in **gigabyte** (GB). I produttori usano i gigabyte decimali, invece delle approssimazioni per potenze di due usate per la **memoria**. Questo significa che la capacità di un disco rigido è in realtà un poco più piccola di quella di un modulo di memoria con la stessa capacità, e lo scarto aumenta all'aumentare delle dimensioni. Quando la capacità è espressa in GB, il fattore di correzione è di  $(1000/1024)^3$ , pari a circa 0,93, per cui un disco rigido da 320 GB ha una capacità effettiva di circa 298 GB. Attualmente i dischi rigidi si trovano in vendita con capacità fino a 2 TB. Si può aumentare la capacità incrementando la densità con cui le informazioni vengono memorizzate sui piatti che compongono l'hard disk o impiegandone un numero maggiore.

Il **tempo di accesso** è la variabile più importante nel determinare le prestazioni di un disco rigido. Si tratta del tempo medio necessario perché un dato, residente in un punto casuale del disco, possa essere reperito. Il tempo impiegato dipende dalla velocità della testina a spostarsi sulla traccia dove risiede il dato e dalla velocità di rotazione del disco; maggiore è la velocità e più breve è il tempo impiegato dal dato a ripassare sotto la testina nel caso questa non fosse arrivata in tempo sul dato, durante la rotazione precedente (*latenza rotazionale*). I produttori cercano perciò di realizzare testine sempre più leggere (che possono spostarsi più in fretta perché dotate di minore **inerzia**) e dischi che girano più velocemente. Il tempo di accesso tipico per un disco rigido da 7200 rpm è di circa 9 millisecondi. Per uno da 15.000 rpm è inferiore a 4 ms.

La **velocità di trasferimento** è la quantità di dati fornita dall' HDisk in un determinato tempo (in genere si prende 1 secondo come riferimento). Usare dischi che ruotino più velocemente o incrementare la densità di memorizzazione porta ad un miglioramento diretto della velocità di trasferimento. Va ricordato che la velocità di trasferimento cala in modo proporzionale al numero di discontinuità nei settori che compongono il file ricercato (vedi **frammentazione (informatica)**).

Inoltre, altre caratteristiche influenzano in misura minore le prestazioni di un disco rigido. Tra queste:

- il **buffer** di memoria
- la velocità dell'**interfaccia**

Il **buffer** è una piccola **memoria cache** (in genere di alcuni **megabyte**) posta a bordo del disco rigido, che ha il compito di memorizzare gli ultimi dati letti o scritti dal disco. Nel caso in cui un programma legga ripetutamente le stesse informazioni, queste possono essere reperite nel buffer invece che sul disco. Essendo il buffer un componente elettronico e non meccanico, la velocità di trasferimento è molto maggiore; nel tempo, la capacità di questa memoria è andata sempre aumentando, attualmente 32 MB sono una dimensione abbastanza usuale.

L'**interfaccia** di collegamento tra il disco rigido e la **scheda madre** (o, più specificatamente, il **controller**) può influenzare le prestazioni perché specifica la velocità massima alla quale le informazioni possono essere trasferite da o per il disco. Le moderne interfacce tipo **ATA133**, **Serial ATA** o **SCSI** possono trasferire centinaia di megabyte per secondo, molto più di quanto qualunque singolo disco fisso possa fare, e quindi l'interfaccia non è in genere un fattore limitante.

## TEMPO DI ACCESSO AL DISCO

Il tempo di accesso a disco è influenzato da **quattro fattori** significativi:

**Seek time** (tempo di ricerca): è il tempo necessario a spostare la testina sulla traccia; è il fattore più critico poiché si tratta di un movimento meccanico e non di un impulso elettrico; questo fa sì che non si possa scendere al di sotto di **qualche decina di millisecondo (10msec)**;

**Assessment time** (tempo di assestamento): è il tempo necessario all'assestamento della testina sulla traccia dopo lo spostamento; spesso viene inglobato nel 'Seek time';

**Latency time** (tempo di latenza): (anche *rotational latency*) è il tempo necessario perché, a causa della rotazione del disco, l'inizio del settore desiderato arrivi a trovarsi sotto la testina; ovviamente dipende dalla velocità di rotazione del disco; per esempio con una velocità (tipica) di 5400 rpm, il tempo di latenza massimo è di circa 11 millisecondi;

**Transfer time** (tempo di trasferimento): è il tempo durante il quale il settore viene letto o scritto.

Tempo di accesso: SeekTime + Latency + TransferTime

## ORGANIZZAZIONE FISICA DELLA MEMORIA DEI DATI

<p>I dati sono generalmente memorizzati su disco seguendo uno schema di allocazione fisica ben definito in base al quale si può raggiungere la zona dove leggere/scrivere i dati sul disco. Uno dei più diffusi è il cosiddetto <i>CHS</i> acronimo per il termine inglese <i>Cylinder/Head/Sector</i> (Cilindro/Testina/Settore); in questa struttura i dati sono memorizzati avendo come indirizzo fisico un numero per ciascuna delle seguenti entità fisiche:</p> <p><b>Piatto</b></p> <p>un disco rigido si compone di uno o più dischi paralleli, di cui ogni superficie, detta "piatto" e identificata da un numero univoco, è destinata alla memorizzazione dei dati.</p> <p><b>Traccia</b> ogni piatto si compone di numerosi anelli concentrici numerati, detti <i>tracce</i>, ciascuna identificata da un numero univoco.</p> <p><b>Cilindro</b> l'insieme di tracce alla stessa distanza dal centro presenti su tutti i dischi è detto <i>cilindro</i>. Corrisponde a tutte le tracce aventi il medesimo numero, ma diverso piatto.</p> <p>Settore ogni piatto è suddiviso in <b>settori circolari</b>, ovvero in "spicchi" radiali uguali ciascuno identificato da un numero univoco.</p> <p><b>Blocco (Cluster)</b> L'insieme di settori posti nella stessa posizione in tutti i piatti.</p>	<p>A traccia                  B settore                  C settore di una traccia                  D cluster, insieme di settori adiacenti</p> <p><b>Testina</b></p> <p>Su ogni piatto è presente una testina per accedere in scrittura o in lettura ai dati memorizzati sul piatto; la posizione di tale testina è solidale con tutte le altre sugli altri piatti. In altre parole, se una testina è posizionata sopra una traccia, tutte le testine saranno posizionate nel cilindro a cui la traccia appartiene.</p>
--	---

Questa struttura introduce una *geometria fisica del disco* che consta in una serie di "coordinate" CHS, esprimibili indicando cilindro, testina, settore. In questo modo è possibile indirizzare univocamente ciascun blocco di dati presente sul disco. Ad esempio, se un disco rigido si compone di 2 dischi (o equivalentemente 4 piatti), 16384 cilindri (o equivalentemente 16.384 tracce per piatto) e 16 settori, e ciascun settore di una traccia ha una capacità di 4096 byte, allora la capacità del disco sarà di  $4 \times 16384 \times 16 \times 4096$  byte, ovvero 4 GiB.

## DISCHI RIGIDI IDE

L'interfaccia più comune è quella **IDE** (prima versione dello standard **ATA**), poi evolutasi in **EIDE** e **Serial ATA**. Un cavo piatto, solitamente grigio, è usato per connettere il disco rigido alla scheda madre. Spesso il cavo ha un terzo connettore per poter usare un altro disco (o altre periferiche ATA come i lettori cd) con lo stesso cavo. In tal caso, per poter distinguere tra le due periferiche, esse devono essere configurate una come *master* (padrone) e una come *slave* (schiavo). Questa configurazione può avvenire sia manualmente, spostando dei **jumper** presenti sulle periferiche, sia automaticamente se esse sono impostate come *cable select*. In quest'ultimo caso è la scheda madre a decidere chi è il master e chi lo slave. Questo è particolarmente utile quando si utilizzano dischi rigidi vecchi, o nel caso di bassa compatibilità tra unità diverse (ad esempio due dischi rigidi, ma anche un disco rigido e un lettore CD).

Una scheda madre ha solitamente due connettori IDE (*primario* e *secondario*, detti spesso *canalio* impropriamente *controller*), ad ognuno dei quali è possibile connettere due unità per un totale di quattro periferiche. Non mancano schede madri con quattro connettori. Il cavo IDE non porta l'alimentazione elettrica necessaria per il funzionamento delle periferiche, che quindi devono essere connesse all'alimentatore per mezzo di un cavo separato.

Ecco un esempio delle possibili connessioni all'IDE di un pc:

<ul style="list-style-type: none"> <li>• canale primario:                         <ul style="list-style-type: none"> <li>• <i>master</i>: disco rigido;</li> <li>• <i>slave</i>: lettore cd (con il <b>jumper</b> su <i>cable select</i>)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• canale secondario:                         <ul style="list-style-type: none"> <li>• <i>master</i>: disco rigido;</li> <li>• <i>slave</i>: masterizzatore DVD</li> </ul> </li> </ul>
--	--

## 6. CD e DVD

La tecnologia dei dischi ottici per i computer deriva da quella dei CD. Il supporto di memoria è un disco di materiale plastico rigido molto leggero, sul quale è posto uno strato di materiale riflettente. Su una superficie è incisa un'unica lunghissima traccia avvolta a spirale dall'interno verso l'esterno.

La **traccia** è costituita da piccolissime aree elementari che possono essere:

- perfettamente riflettenti ( sono chiamate "land"), le quali simulano un bit "1"
- perfettamente assorbenti, cioè non riflettono il raggio incidente (sono dette "pit") e simulano il bit "0"

Il **fascio di luce** che incide contro il CD è un raggio laser ; il raggio riflesso ( che si ha solo se si incontra un bit "1" cioè un "land" ) viene separato da quello incidente tramite uno prisma riflettente

Il **disco** viene fatto ruotare da un motore a velocità molto elevate e variabile, visto che la testina nel frattempo si sposta dal centro verso l'esterno del disco.

I **dischi ottici** si possono dividere dunque in :

**CR ROM** (CD read only memory). Memoria a sola lettura, scritti dal costruttore

**CD -R** (recordable): sono una memoria esterna a sola lettura , scritti una volta dall'utente e non più modificabili

**CD -RW** (CD Rewritable): possono essere scritti più volte mediante una apposita unità di incisione.

I dispositivi per scrivere e/o leggere un CD o DVD sono:

**Lettores CD-ROM:** è in grado di svolgere operazioni di sola lettura su dischi CD-ROM e CD-R

**Masterizzatore:** esegue operazioni di scrittura su dischi CD -R e Cd -RW e di lettura per tutti i tipi di disco quando le unità di scrittura e lettura sono interne al computer sono connesse tramite interfaccia hardware ( ad esempio IDE) al bus interno dei dati, in modo simile ai dischi fissi.

Particolari forme di CD ROM sono i **DVD** (Digital Versatile Disk) formati da dischi ottici ad alta risoluzione in grado di memorizzare almeno 4,7GB di dato su ciascuno dei due lati. I DVD devono venir letti da opportuni lettori (lettori DVD) che sono in grado di leggere anche i CD ROM. Esistono anche specifici masterizzatori DVD.

Il **tempo di accesso** ai dati è di tipo **sequenziale**.

I dati tecnici relativi ai lettori Cd e DVD sono:

- **velocità di trasferimento del dato:** espressa in byte o carattere al secondo, dove la comunicazione si intende dal supporto alla memoria centrale (lettura) o viceversa (scrittura). I lettori CD r avevano come velocità iniziale 150KB/sec

Tale velocità è diventata lo standard rispetto al quale si definiscono ora le attuali misure. Ad esempio:

2X indica una velocità di lettura di 300 KB/sec,

3X indica una velocità di lettura di 450 KB/sec

50X indica una velocità di lettura di 7500 KB/sec, ecc.

- il secondo parametro dovrebbe essere il **tempo di accesso:** dato che l'accesso non è diretto ma sequenziale, il tempo di accesso dipende da dove è collocato il dato specifico che vogliamo leggere.

Velocità di trasferimento e tempo di accesso sono collegati: se si aumenta la velocità di trasferimento, aumenta la probabilità di trovare in breve tempo le informazioni ricercate e quindi diminuisce il tempo medio di attesa. Tempi medi di attesa vanno da 100msec a 1 sec

## 7. Unità video

Si tratta del dispositivo di output più utilizzato, la cui funzione è quella di visualizzare su uno schermo informazioni testuali o grafiche.

Nella comunicazione fra l'utente e l'elaboratore si possono individuare i seguenti ruoli:

- la **sorgente** di dati è la memoria centrale
- **interfaccia verso l'esterno**, cioè l'utente, è realizzata tramite un monitor
- **interfaccia verso interno** è definita da un cavo di collegamento ( cavo dedicato per la scheda video), il bus del calcolatore e opportuni componenti hardware che hanno la funzione di aumentare la velocità di comunicazione.
- il **destinatario** è l'utente.

Determinante nella comunicazione fra video e processore è **la funzione della scheda video**, che contiene:

- una memoria volatile, **chiamata RAM video**, che ha la funzione di tutte le informazioni necessarie per visualizzare l'immagine sullo schermo
- coprocessori I/O chiamati **acceleratori grafici** per aumentare la velocità dello scambio dati fra scheda e processore

la scheda video è controllata da un software di interfaccia, chiamato **driver della scheda video**.

Un'unità video può lavorare in due distinte modalità: **modalità testo** e **modalità grafica**.

**La modalità testo** prevede la divisione dello schermo in 25 righe e 80 colonne (quindi allora si hanno massimo 80 caratteri). La memoria centrale deve riservare un insieme di 2000 (25x80) celle di memoria a 8bit (1B) nelle quali è posto il codice ASCII del simbolo da presentare sullo schermo

**la modalità grafica** prevede che il video venga gestito in termini di piccoli punti chiamati pixel organizzati in righe e colonne. Ogni immagine è visualizzata in modo discreto per cui tutti i particolari sono rappresentati tramite una matrice di punti. La risoluzione è data dal numero dei pixel riga per colonna: **numero pixel riga x numero pixel colonna**. La risoluzione rappresenta la capacità del video di distinguere fra due punti vicini fra loro: se aumentiamo il numero di pixel, l'utente esterno non riesce a distinguerli a occhio nudo e ha la percezione di una immagine continua nei suoi contorni. Valori tipici sono:

Standard	Risoluzione	Numero di colori per pixel disponibili
VGA (VideoGraphicAdapter)	640x480	16, 256, 65536(64k),16.8 milioni ( 4M)
Super VGA	1280x768,1152x864, 1024x768, 800x600,	16, 256, 65536(64k),16.8 milioni ( 4M)

La disponibilità dei colori per il video viene chiamata **palette** (tavolozza) dei colori. Se lavoriamo con uno schermo monocromatico, avremo a disposizione solo un numero limitato di grigio, di solito **256 livelli di grigio**.

Con i monitor a colori invece la disponibilità cresce.

Pertanto nella RAM video deve essere memorizzato:

- **la posizione in termini di riga e colonna di ciascun pixel**
- **e il colore o la tonalità di grigio associato a ciascun pixel**

Caso pratico: immaginiamo di disporre di un video con una risoluzione di 800x600 pixel con una palette di colori pari a 65536 (64k di colore). Ciò significa che:

- *abbiamo un numero di pixel pari a  $800 \times 600 = 480000$  pixel*
- *per ogni pixel dobbiamo indicare il colore fra i 64K disponibili: se ci sono 64K possibili colori diversi allora i bit riservati al colore sono 16 , poiché  $2^{16} = 65536$  , quindi per ogni pixel sono necessari 16bit, cioè 2B, per descrivere il relativo colore*

Ricapitolando la memoria RAM video dovrà essere capiente almeno di  $480000 \times 2B = 960000B$ , cioè circa 10MB

Schematizzando quanto riportati, vengono ora elencati i parametri caratteristici di una unità video

Parametri unità video	Descrizione
Risoluzione complessiva	Numero di pixel riga x colonna, oppure la capacità della memoria RAM
Modalità	Grafica o alfanumerica ( testuale)
tipo	Monocromatica o a colori, indicando allora la disponibilità della palette
Dimensione fisica dello schermo	Viene indicata in pollici sulla diagonale ( il rapporto fisico fra i lati è sempre 4:3, quello di visualizzazione può essere diverso ) si ricorda che 1 pollice = 2.54 cm

## 8. Stampante

La stampante è un dispositivo di uscita (**output**) la cui funzione è quella di fornire e registrare in modo permanente su carta i risultati di una elaborazione.

Nella comunicazione fra l'utente e l'elaboratore si possono individuare i seguenti ruoli:

- la **sorgente** di dati è la memoria centrale
- **interfaccia verso l'esterno**, cioè l'utente, è definita da una serie di meccanismi meccanici o elettronici che hanno lo scopo di stampare caratteri e immagini
- **interfaccia verso interno** è definita da un cavo di collegamento (prima un cavo che collega la stampante alla porta parallela ora USB), e il driver della stampante.
- il **destinatario** è l'utente

I parametri caratteristici di una stampante sono i seguenti:

Parametri	Descrizione
Tecnica di stampa	Può essere a impatto, a getto di inchiostro, laser
Velocità	Indicata con CPS (numero di caratteri per secondi)/ righe al minuto
Tipo	Monocromatica o a colori
tipo di collegamento verso il calcolatore	Mediante cavo seriale, parallelo, o attualmente USB
Formato della carta	La dimensione più diffusa è l'A4 che corrisponde a 21,0X29,7cm
<b>Analisi delle tecniche di stampa impiegate</b>	
Stampanti a impatto (ad aghi)	<p>Usano una tecnica simile alle macchine da scrivere. Il foglio scorre verticalmente su un rullo per definire le varie righe. Per ciascuna riga saranno le testine da sinistra a destra a battere il carattere giusto. Ogni carattere è definito da una matrice di punti bianchi e neri, per cui la testina per ogni carattere non deve far altro che riempire i punti in nero del carattere, mediante l'impatto degli aghi sul foglio tramite il nastro inchiostro; la qualità della stampa dipende dal numero di punti per matrice e la modalità di stampa è detta</p> <p><b>Draft (bozza)</b> con velocità da 100 a 500 CPS</p> <p><b>NLQ (Near Letter Quality)</b> con velocità da 70 a 100 CPS</p>
Stampante a getto di inchiostro (ink jet)	<p>Il principio è simile a quello delle stampanti a impatti, con la differenza che la testina non è composta da aghi che martellano il foglio ma da piccoli fori (ugelli) da cui sprezza l'inchiostro. Ogni carattere è descritto da una matrice di punti bianchi o neri,. Tale stampante si presta a stampe a colori, se usiamo cartucce a tre colori RGB e al posto del singolo foro ne abbiamo tre.</p> <p>La qualità è molto buona ed è definita dal parametro <b>dpi (dot per inch = numero di punti per pollice quadrato)</b>. Valori tipici <b>600x600 dpi</b> oppure <b>2400x1200 dpi</b>.</p>
Stampanti laser	<p>Si basano sul principio di funzionamento delle macchine fotocopiatrici. La stampa avviene in tre fasi:</p> <ul style="list-style-type: none"> <li>• <b>l'immagine viene disegnata su un rullo</b> tramite un pennello di luce laser: il rullo è sensibile alla luce e si carica di elettricità nei punti in cui è colpito</li> <li>• <b>sul rullo viene depositato uno strato di polvere di inchiostro (toner)</b> che si deposita solo sulle parti elettrizzate</li> <li>• il rullo viene fatto ruotare su un foglio di carta e <b>il toner si deposita sul foglio</b>.</li> </ul> <p>La qualità è ottima, raggiungendo i <b>600 e 1200 dpi</b>.</p>