

---

# Introduzione all'Architettura del Calcolatore

lezione 3

**Luciano Bononi**

**[bononi@cs.unibo.it](mailto:bononi@cs.unibo.it)**

**<http://www.cs.unibo.it/~bononi/>**

Figure credits: some figures have been taken from web presentations on the web:  
Larry and Nancy Long, Mike Schulte

© 2004 Luciano Bononi

1

---

## Testi consigliati:

**William Stallings, *Computer Organization and Architecture, 5/6-th edition*  
Prentice Hall, 2003, ISBN: 0-13-035119-9**

**Andrew Tanenbaum, [Structured Computer Organization, Fourth Edition](#)  
Prentice Hall (1999) ISBN: 0-13-095990-1**

**...altri testi potranno essere citati in seguito.**

© 2004 Luciano Bononi

2

## ASCII Chart (vedi lezione precedente)

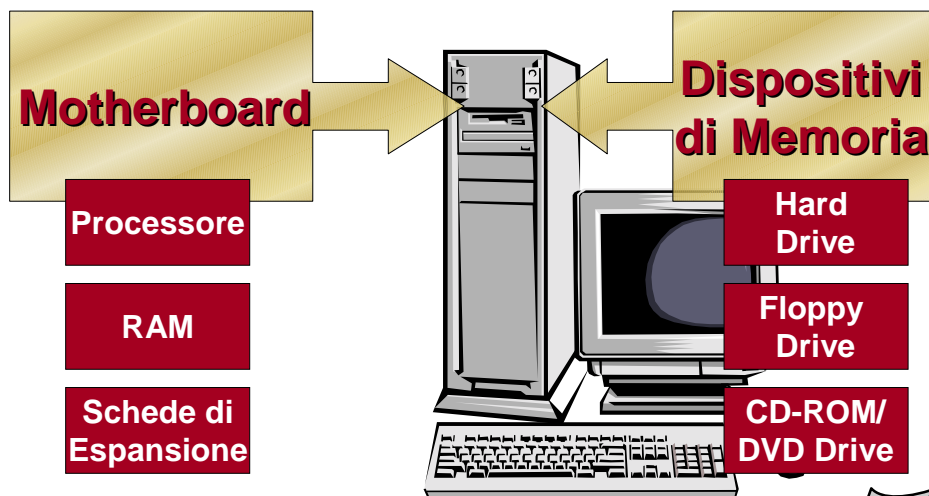
Character	ASCII Code	Character	ASCII Code
A	100 0001	0	011 0000
B	100 0010	1	011 0001
C	100 0011	2	011 0010
D	100 0100	3	011 0011
E	100 0101	4	011 0100
F	100 0110	5	011 0101
G	100 0111	6	011 0110
H	100 1000	7	011 0111
I	100 1001	8	011 1000
J	100 1010	9	011 1001
K	100 1011	Space	010 0000
L	100 1100	.	010 1110
M	100 1101	(	010 1000
N	100 1110	+	010 1011
O	100 1111	&	010 0110
P	101 0000	\$	010 0100
Q	101 0001	*	010 1010
R	101 0010	)	010 1001
S	101 0011	;	011 1011
T	101 0100	,	010 1100
U	101 0101	-	101 1111
V	101 0110	?	011 1111
W	101 0111	:	011 1010
X	101 1000	=	011 1101
Y	101 1001		

**Altre codifiche:**  
ANSI (8 bit)  
Unicode (16 bit)  
Display Esadecimale

© 2004 Luciano Bononi

3

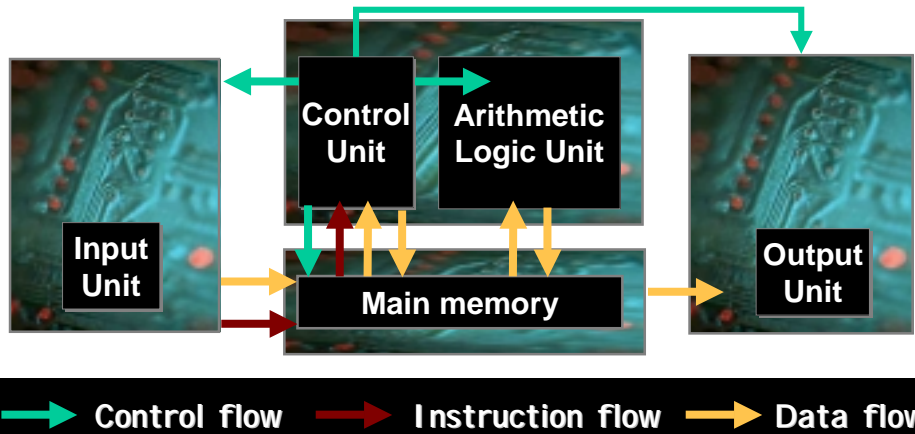
## PC System Unit



© 2004 Luciano Bononi

4

Cosa succede all'interno della motherboard?

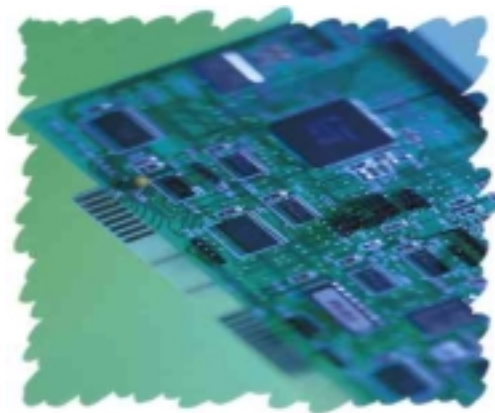


© 2004 Luciano Bononi

5

Motherboard

- Scheda contenente circuiti elettronici principali
- Fornisce il supporto comune a una serie di dispositivi:
  - Microprocessore
  - Chipset
  - Chip di Memoria
  - Schede di espansione
  - Bus
  - Controller dei device



© 2004 Luciano Bononi

6

## Es. Intel: evoluzione dei microprocessori



- 286, 386, 486
- Pentium®
- Pentium® Pro
- Pentium® II
- Pentium® III
- Celeron®
- Itanium™

© 2004 Luciano Bononi

7

## Componenti del microprocessore

### Unità di Controllo

- Legge e Interpreta le istruzioni del programma
- Dirige le operazioni dei componenti interni
- Controlla il flusso di dati/istruzioni da/verso RAM

Decoder

Program Register

Instruction Register

Accumulator

Unità Aritmetico/logica (ALU)

© 2004 Luciano Bononi

8

## Es. Progettazione delle componenti di una CPU

- **Un insieme di componenti (circuiti elettronici)**
  - Ogni circuito svolge elaborazioni su segnali definiti attraverso funzioni logiche
- **Come realizzare funzioni logiche?**
  - Algebra di Boole (1815,1864): consente di gestire la progettazione di circuiti attraverso metodi matematici
- **Funzione logica (rete combinatoria):**
  - $f$  (bit di ingresso) = bit di uscita
  - Es.  $F(b_2, b_1, b_0) = z_0$
  - Può essere completamente specificata mediante tabella di verità

© 2004 Luciano Bononi

9

## Es. Progettazione delle componenti di una CPU

- **Tabella di verità**
  - Associa ad ogni combinazione possibile degli ingressi il valore di uscita della funzione

$2^n$  possibili configurazioni diverse degli  $n$  bit di ingresso ad ognuna delle quali corrisponde un valore  $z$  (0 oppure 1) di uscita

B(n-1)...	B(1)	b(0)	z
0 ... 0	0	0	$z_0$
0 ... 0	0	1	$z_1$
0 ... 1	0	1	$z_2$
0 ... 1	1	1	$z_3$
1 ... 1	1	1	$z_{(2^n)-1}$

- Quante sono le diverse possibili funzioni logiche su  $n$  bit di input?
  - ...una per ogni possibile configurazione degli output  $z_0 \dots z_{(2^n)-1}$
  - ...quindi in totale  $2^{(2^n)}$  possibili funzioni diverse

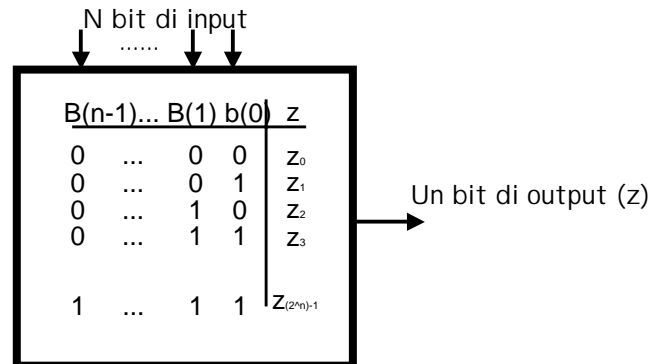
© 2004 Luciano Bononi

10

## Es. Progettazione delle componenti di una CPU

### ▪ Rete logica sequenziale

- È un circuito che visto da fuori si comporta come specificato dalla sua funzione logica



© 2004 Luciano Bononi

11

## Es. Progettazione delle componenti di una CPU

### ▪ Le porte logiche

- Sono funzioni logiche che vengono usate come “mattoni” per realizzare circuiti (e funzioni logiche) complesse

### ▪ Vediamo le più importanti...

AND: prodotto logico

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

OR: somma logica

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

NOT: negazione logico

A	NOT A
0	1
1	0

Con queste componenti si possono realizzare tutte le funzioni logiche possibili (And, Or, Not costituiscono un insieme funzionalmente completo)... La dimostrazione intuitiva è stata mostrata a lezione.

© 2004 Luciano Bononi

12

## Es. Emulazione di una CPU

---

### ▪ A questo punto...

- Possiamo capire meglio come siano realizzate le unità funzionali e il controllo di alcuni prototipi di CPU
- Vediamo due esempi di CPU in emulazione RETRO (scaricabili dalla pagina web del corso)
- La discussione delle funzionalità delle componenti viene svolta a lezione...

Esempio di CPU ed esecuzione di due semplici programmi di ordinamento di valori di un vettore e conteggio di occorrenze di una stringa in memoria

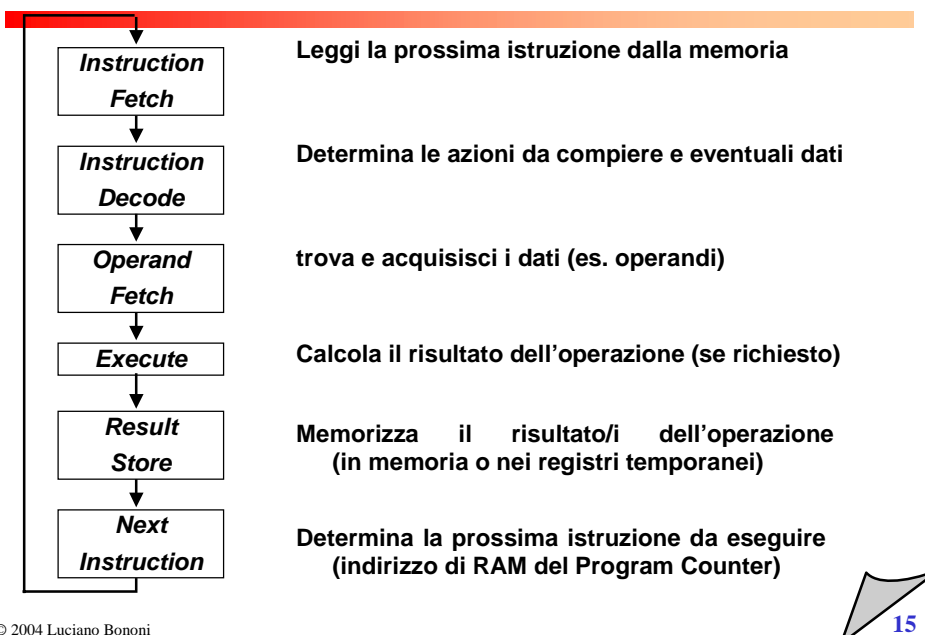
## Es. Emulazione di una CPU

---

### ▪ Quale è la morale?

- L'architettura della CPU è fortemente caratterizzata sulla base delle istruzioni (operazioni) che si intendono realizzare: la libreria di istruzioni della CPU e i rispettivi codici operativi scelti per istruire l'unità di controllo
- La CPU stessa sarà in grado di compiere compiti ed elaborazioni complesse attraverso l'esecuzione di programmi espressi come sequenze di istruzioni della propria libreria (e non di altre).
- La CPU esegue un ciclo di istruzione "infinito" ed è gestita in modo da eseguire il flusso delle istruzioni del programma in memoria: cambiare il programma e i dati permette di riciclare la stessa CPU per compiti diversi

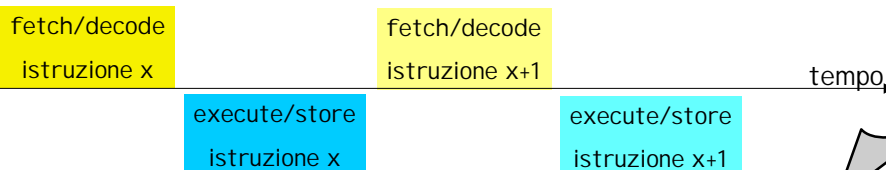
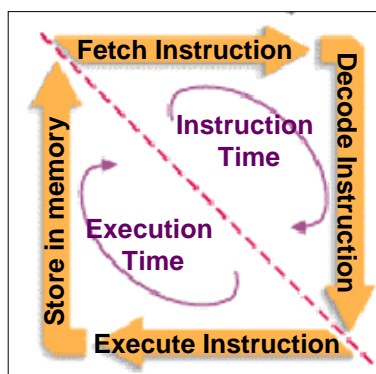
## Ciclo di esecuzione dell'istruzione



Fasi del Ciclo di Istruzione = Instruction-time + Execution-time

### Throughput?

Indica quanti "risultati" si producono in una determinata unità di tempo

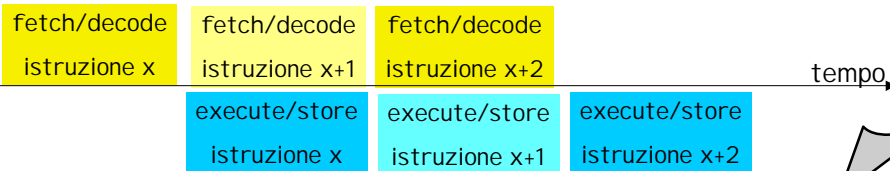
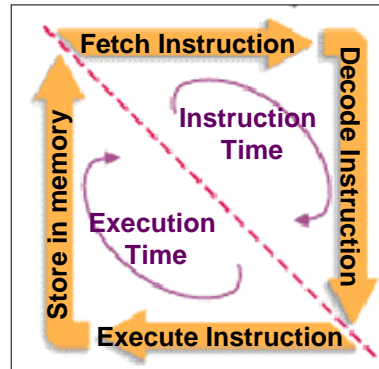




Fasi del Ciclo di Istruzione= Instruction-time + Execution-time

## Pipelining?

Cercare di eseguire  
in parallelo le due  
fasi del ciclo di  
istruzione:



© 2004 Luciano Bononi

17

## Il Processore tipico attuale

- **Unità di elaborazione dei dati**
  - 32/64-bit
- **Velocità**
  - MHz, MIPS, MegaFLOPS?
  - da 750 MHz to 3 GHz clock rate
- **Memoria cache**
  - 32-512 kB livello 1
  - Fino a 6 MB livello 3
- **Memoria RAM (motherboard)**
  - Da 128 MB a 4 GB e oltre...

© 2004 Luciano Bononi

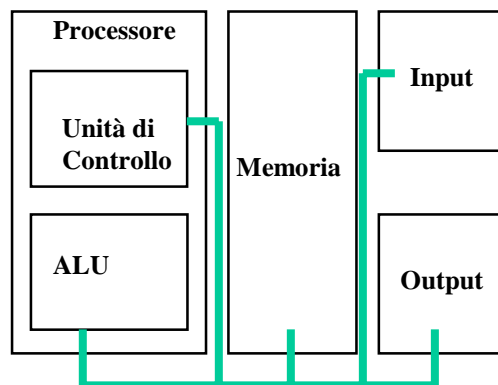
18

## Capacità di memoria: unità di misura

- un Byte → 8 bits (circa un carattere ASCII)
- Kilobyte (KB) –  $2^{10} = 1024$  bytes
- Megabyte (MB) –  $2^{20} =$  circa un milione di bytes
- Gigabyte (GB) –  $2^{30} =$  circa un miliardo di bytes
- Terabyte (TB) –  $2^{40} =$  circa mille miliardi di bytes

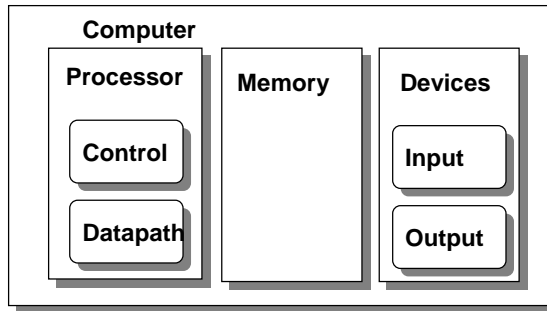
## Schema Generale di un PC: John Von Numann

- il PC ha 5 componenti principali connesse da una serie di Bus per il trasporto dei dati

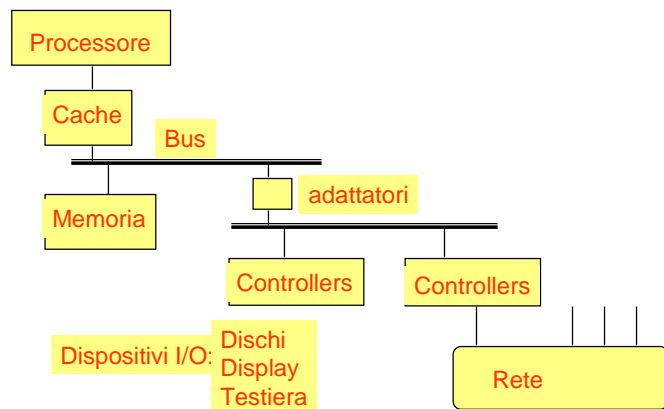


## Costo relativo delle componenti

Obiettivi della progettazione:  
25% costo processore  
25% costo memoria  
50% tutto il resto

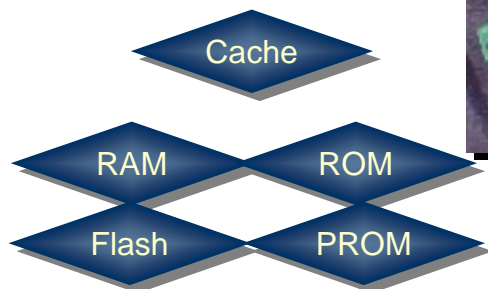


## Componenti e organizzazione



- **Ogni componente ha interfacce verso altri componenti e gestori o coordinatori delle funzioni**

## Tipi di dispositivi di memoria (chip)



© 2004 Luciano Bononi

23

## RAM

- **RAM: Random Access Memory (accesso casuale)**
- **serve a mantenere i dati e programmi in esecuzione**
- **permette di indirizzare direttamente porzioni di dati o istruzioni (indirizzo di RAM)**
- **Richiede alimentazione costante** es. 1,2,4,8 Byte

- volatile

es. 256  
indirizzi da  
8 bit

indirizzo 00	dato "A"
indirizzo 01	istruzione X
indirizzo FE	dato "\$"
indirizzo FF	dato 3

© 2004 Luciano Bononi

24

## Tipi di RAM

- **SDRAM (Synchronous Dynamic RAM)**

- Riescono a sincronizzarsi con la velocità del bus verso il processore (fino a 133 MHz)

- **RDRAM = Rambus DRAM (vs. SDRAM)**

- Innovative e costose
- 6 volte più veloci delle SDRAM (clocked fino a 800-1600 MHz)

- **Installazione fisica**

SIMMs: (single inline mem mod.) 32-bit data path verso CPU

DIMMs: (dual) 64-bit data path verso la CPU

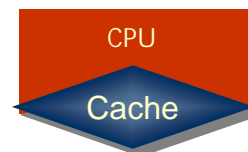
RIMMs: RDRAM chips (+ veloci)



## Cache Memory

- **Velocissima e costosa**

- (molto più della RAM)



- **Capacità di memorizzazione molto inferiore alla RAM**

- **Mantiene le prossime istruzioni da eseguire**

- come fa a sapere quali sono?

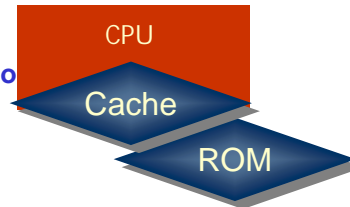
- **Aumenta notevolmente il throughput del sistema di calcolo**

- vedremo come....

## ROM

---

- **ROM: Read Only Memory**
- **Mantiene permanentemente dati e istruzioni**
  - non volatile
  - non programmabile dall'utente
- **es. Mantiene istruzioni del sistema operativo nella fase di boot**
- **PROM: variante della ROM**
  - programmabile una volta (non volatile)
  - costosa
- **EPROM: variante della ROM**
  - (ri-)programmabile e cancellabile (non volatile)
  - costosa



© 2004 Luciano Bononi

27

## Memoria Flash

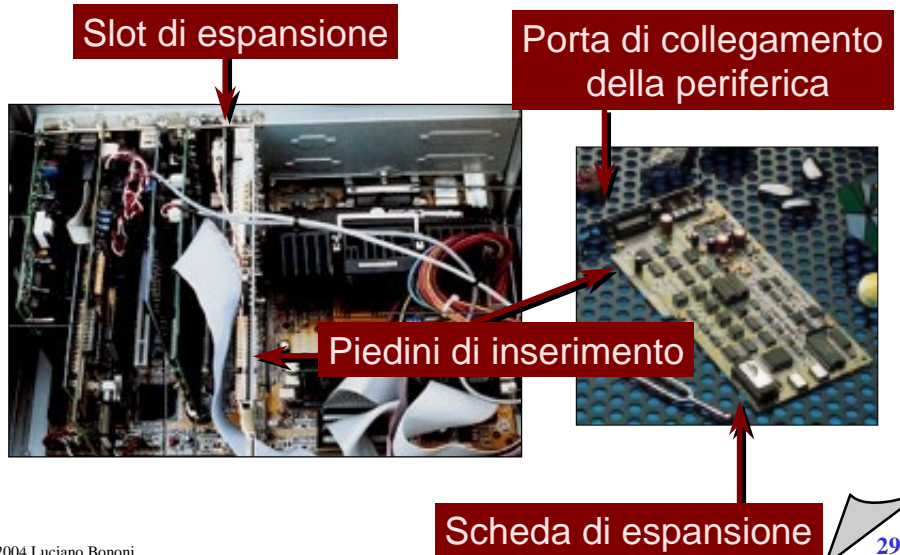
---

- **simile alle PROM**
- **Può essere facilmente programmata (scritta) da utenti**
- **Non-volatile**
  - es. usata per fare upgrade di software da Web o Disco
  - Non richiede di fare sostituzione di chips o schede
- **Es. Chiavette USB (USB key)**

© 2004 Luciano Bononi

28

## Aggiunta di dispositivi periferici alla motherboard



## Tipi di Bus

- **Bus ISA (industry Std. Architecture)**
  - dal 1990 su IBM XT/AT
- **bus (locale) PCI (Peripheral Component Interconnect)**
  - 64bit (32 bit di fatto), 33/66Mhz clock (133 MBps)
- **bus SCSI (Small Computer System Interface)**
  - usato per dischi e stampanti, fino a 80Mbps
  - possibile attaccare più dispositivi allo stesso port
  - connettori diversi (incompatibili)
  - IDE variante Macintosh (controller sul device)...continua

**Le schede di espansione sono costruite per collegamenti specifici a vari tipi di bus**

## Tipi di Bus (SCSI)

- **bus SCSI (Small Computer System Interface)**
  - più dispositivi connessi allo stesso port, connettori diversi (incompatibili), IDE variante Macintosh (controller sul device)
  - **SCSI-1:** bus a 8-bit, data rates fino a 4 MBps
  - **SCSI-2:** vedi SCSI-1, ma usa connettore a 50-pin invece di 25-pin, e supporta dispositivi multipli. plain SCSI.
  - **Wide SCSI:** 68 pins, 16-bit.
  - **Fast SCSI:** bus a 8-bit, doppio clock rate, 10 MBps.
  - **Fast Wide SCSI:** bus a 16-bit, 20 MBps.
  - **Ultra SCSI:** bus a 8-bit, 20 MBps.
  - **SCSI-3:** bus a 16-bit, 40 MBps. (*Ultra Wide SCSI*)
  - **Ultra2 SCSI:** bus a 8-bit, 40 MBps.
  - **Wide Ultra2 SCSI:** bus a 16-bit, data rates of 80 MBps.

© 2004 Luciano Bononi

31

## Tipi di Bus

- **bus USB 1.1 (Universal Serial Bus)**
  - bus esterno, 12 Mbps, un port connette fino a 127 device
  - usato per sostituire porte seriali e parallele (mouse, stampanti, ecc.)
- **bus USB 2.0 (Hi-speed USB)**
  - fino a 480Mbps compatibile con USB1.1
- **bus 1394 (FireWire, i.link, Lynx)**
  - bus esterno fino a 400Mbps (a) e 800Mbps (b)
  - fino a 63 dispositivi, permette trasmissione a rate garantito
- **bus AGP (Accelerated Graphic Ports) N.B. ha varie restrizioni**
  - versione PCI a supporto di dati per grafica 3-D
  - canale diretto da controller grafico a memoria (32bitx66Mhz)

© 2004 Luciano Bononi

32



## Porte di connessione (ports)

---

### Parallel

- 8 bits in parallelo
- Veloci
- es. stampanti
- brevi distanze

### Dedicated

- tastiera
- mouse
- connettore 5-pin

### 1394

- +nuove e +veloci
- costose
- es. videocam, HD esterni



© 2004 Luciano Bononi

### Serial

- 1 bit alla volta
- Lente
- es. mouse, modem
- "lunghe" distanze

### IrDA

- Luce infrarossa
- es. stampanti, mouse
- brevi distanze, LoS

### SCSI

- 15 periferiche
- concatenazione

### USB

- alta velocità

33

## PCMCIA Card

---



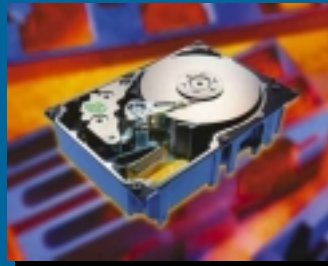
© 2004 Luciano Bononi

34

## Dischi fissi (o Dischi rigidi)

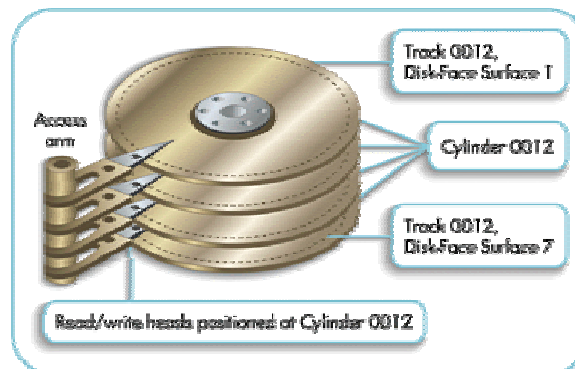
---

- **Più veloci e capienti dei dischetti da 3,5 pollici**
- **da 10 GB a >120 GB**
  - Seek time (track), latency time (rpm), data rate
- **Anatomia:**
  - superficie di registrazione
  - dischi multipli
  - testina di lettura/scrittura
  - braccia multiple



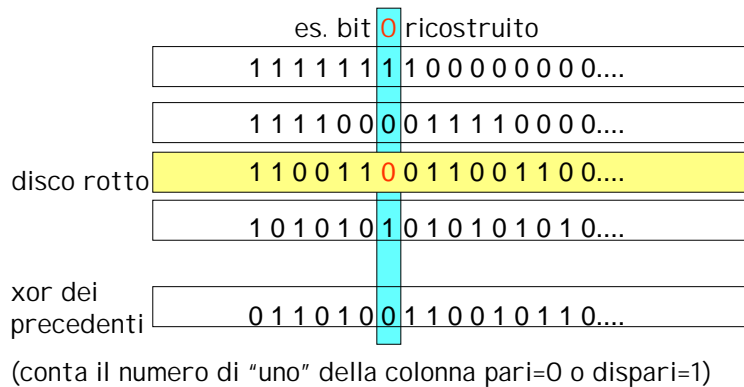
## Schema di un Hard Disk

---



## Tecnologia RAID (Hard Disk)

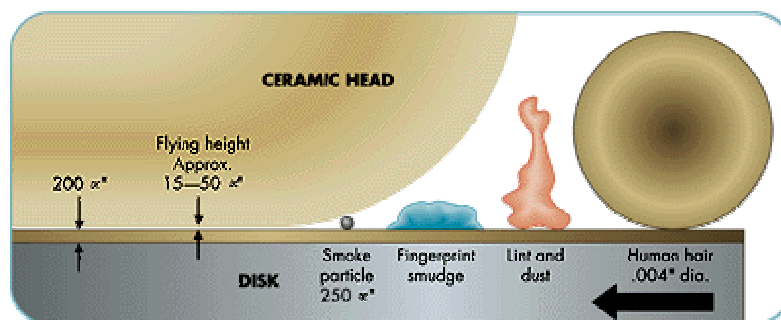
- Tecnologia RAID: Redundant Array of Independent Disks
  - permette di avere i dati sparsi su HD multipli
  - i dischi possono essere sostituiti "a caldo"
  - Fault Tolerance



© 2004 Luciano Bononi

37

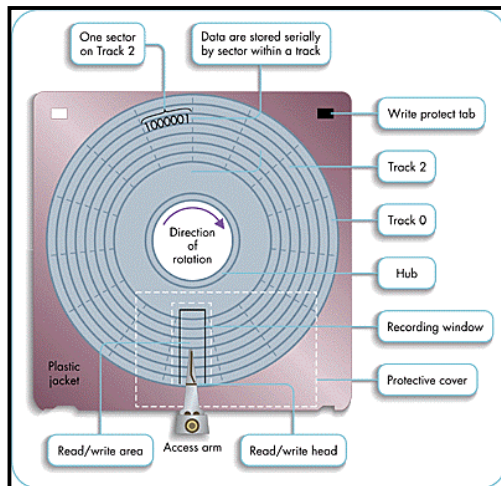
## Perchè gli HD sono delicati?



© 2004 Luciano Bononi

38

## Organizzazione dei dischi magnetici (HD e Floppy)



© 2004 Luciano Bononi

39

### Indirizzi su disco

- Traccia/settore/cluster
- Cilindro

### Scrittura

- sottile superficie magnetizzata

### Lettura

- testina rileva presenza/assenza di magnetizzazione

## Organizzazione del disco

### VFAT:

- Virtual File Allocation Table

### ScanDisk

- rileva cluster danneggiati

### Deframmentazione

### Formattazione

### Fattori e Parametri di velocità del disco

- Tempo di accesso
  - Seek + latency
- Data transfer rate
- Cache del disco?

© 2004 Luciano Bononi

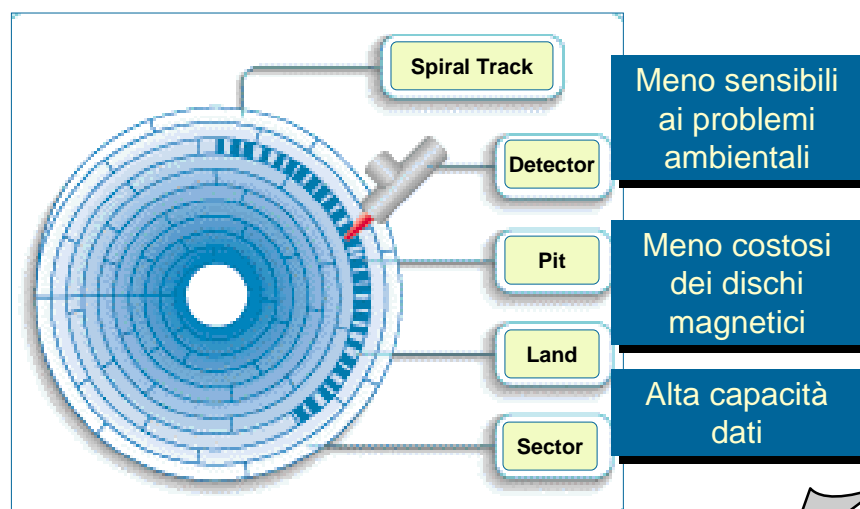
40

## Computer Virus: cosa accade al disco?

- **Non si deve cambiare il computer!**
  - non fanno danni fisici all'HW (per ora ☺)
  - fanno danni "fisici e logici" ai dati (comunque sufficienti a farvi arrabbiare!)
- **Infezioni:**
  - Internet
  - e-mail (attachments)
  - dischetti
  - via rete locale
- **Usare antivirus e fare copie di backup spesso!**



## Dispositivi Ottici/laser



## Dischi ottici: Write-Only

---



### CD-ROM

- Compact Disk: memoria read only
- Non può essere "aggiornato"
- Velocità: 32X, 40X, 75X (rotazione)
- Jukeboxes
- Capacità: 650/700... MB

### DVD-ROM

- Capacità: 4.7 GB fino a 17 GB
- retro-compatibile con CD-ROM

© 2004 Luciano Bononi

43

## Dischi ottici: Read/Write

---

### CD-R

- Compact disk-  
registrabile

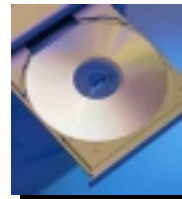
### CD-RW

- Compact disk-  
riscrivibile
- Non compatibile con  
vecchi lettori CD-ROM

### DVD-R(+R)

### DVD-RW (+RW), DVD-RAM

- Riscrivibile



### FMD-ROM

- (Fluorescent MultiLayer Disc)  
Capacità fino a 140 GB e retro-  
compatibile, fino a 10 strati di  
materiale fluorescente con lettura  
dati in parallelo dai 10 strati

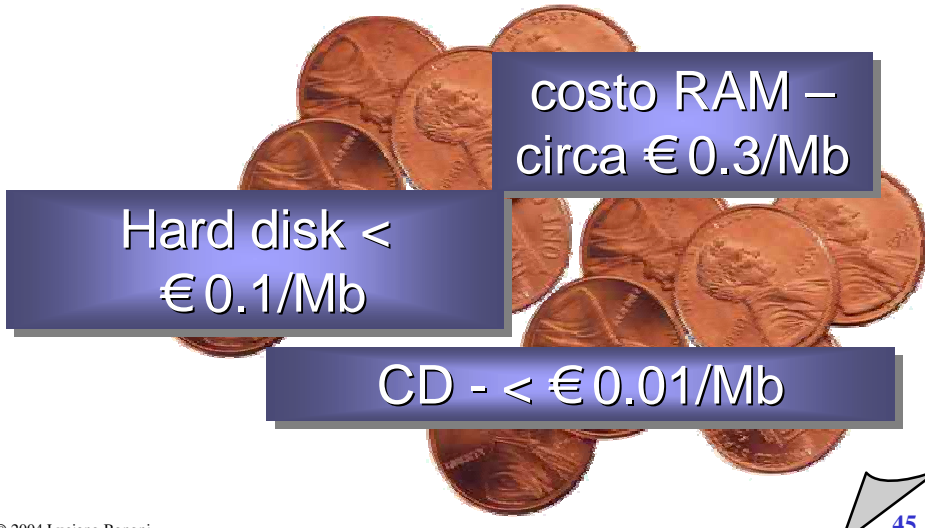
### HD-DVD (blue ray)

- Fino a 27 GB (one side)

© 2004 Luciano Bononi

44

## Confronto tra i costi



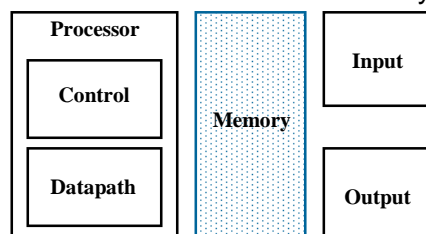
© 2004 Luciano Bononi

45

## Memoria: struttura e gerarchia

### ▪ Memoria

- Dynamic Random Access Memory (DRAM) – usata per memoria principale
- Static Random Access Memory (SRAM) – per la cache



## Trend tecnologico delle memorie e del processore

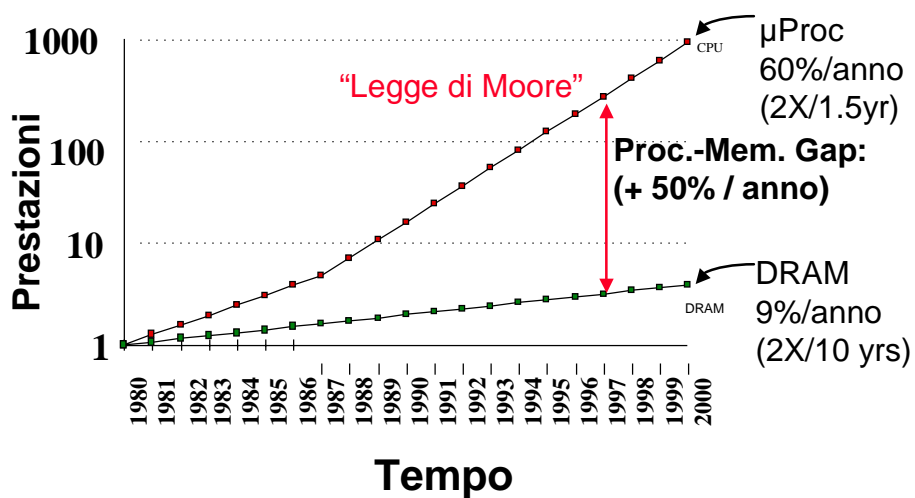
	Capacità	Velocità di accesso
Processore:	2x in 3 years	2x in 3 years
DRAM:	4x in 3 years	2x in 10 years
Dischi:	4x in 3 years	2x in 10 years

es. DRAM		
Anno	Capacità	Cycle Time
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	145 ns
1995	64 Mb	120 ns
1998	256 Mb	100 ns
2001	1 Gb	80 ns

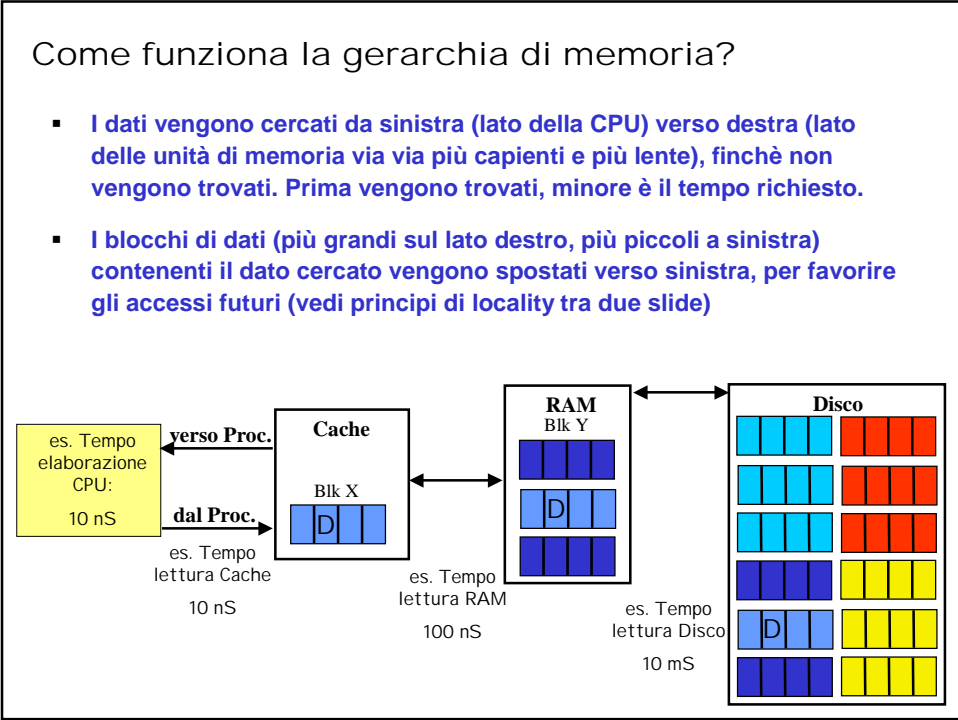
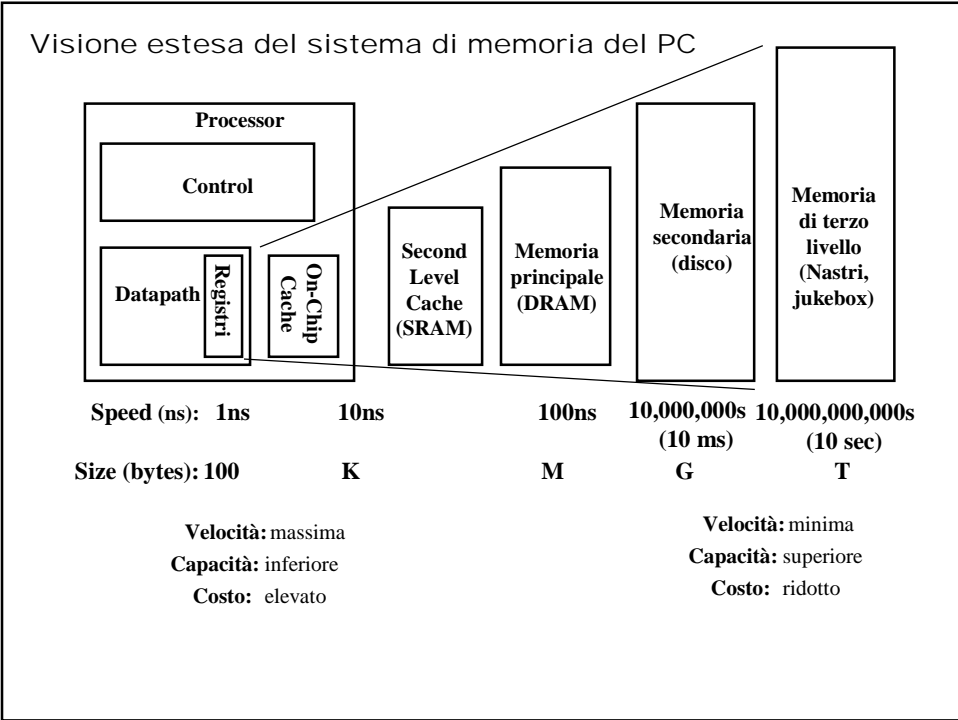
1000:1! (Capacità 1980-1995)  
2:1! (Cycle Time 1980-1995)

## Il GAP Memoria/Processore

### Andamento della latenza Memoria/Processore







## Gerarchia di memoria: terminologia

- **Hit: quando il dato richiesto si trova nella memoria di alto livello (es. cache)**
  - **Hit Rate:** frazione di accessi in memoria il cui dato si trova in cache
  - **Hit Time:** tempo di accesso al livello superiore (cache)= tempo di accesso a memoria + tempo per rilevare HIT/MISS
- **Miss: quando il dato richiesto non si trova nella memoria di alto livello (es. cache) ma deve essere caricato dalla memoria di basso livello**
  - **Miss Rate** =  $1 - (\text{Hit Rate})$
  - **Miss Penalty:** tempo necessario a sostituire un blocco + tempo per recapitare il blocco al processore
- **Hit Time  $\ll$  Miss Penalty**

## Gerarchia di memoria: prestazioni

- **Prestazioni della gerarchia (1 solo livello: cache-RAM)**
  - $h$ =hit rate in cache, e quindi  $(1-h)$ =miss rate
  - $m$ =tempo di accesso dato in RAM (miss penalty)
  - $c$ =tempo di accesso dato in cache (hit time)
- **Tempo medio di accesso (TMA) a un dato memorizzato**
  - $TMA = c \cdot h + (c+m) \cdot (1-h) = c + (1-h) \cdot m$
  - es. se  $h=90\%$ ,  $c=5$  ns e  $m=30$  ns,  $TMA=8$  ns
  - Domanda: conviene investire 1000 euro in cache arrivando a  $h=99\%$ ?
    - $TMA = 5 + (1-0.99) \cdot 30 = 5.3$  ns contro gli 8 ns precedenti.
  - Risposta... dipende!

Come viene gestita la gerarchia?

- **Registri <-> Memoria**
  - es. attraverso il compilatore (e il programmatore)
- **cache <-> memoria**
  - controllo hardware (+ “arte” programmazione)
- **memoria <-> dischi**
  - controllo hardware
  - supporto del sistema operativo (es. memoria virtuale)
  - programmatore (definizione dei files)

Tecnologia di memoria

- **Accesso casuale (Random Access):**
  - “Random”: tempo di accesso costante per ogni dato in ogni indirizzo
  - **DRAM**: Dynamic Random Access Memory
    - alta densità, basso consumo, costa poco, lenta
    - Dynamic: deve essere “rinfrescata” regolarmente
  - **SRAM**: Static Random Access Memory
    - bassa densità, alto consumo, costosa, veloce
    - Static: mantiene il contenuto senza necessità di “refresh” (se alimentata)
- **“Non-così-random” (mixed):**
  - il tempo di accesso varia a seconda della posizione dei dati e del momento in cui accedo
  - Es.: Dischi, CDROM
- **Sequenziale (Sequential Access):**
  - tempo di accesso dipende linearmente dalla locazione dati (nastro)

## Principi generali della gerarchia di memoria

### ▪ **Locality**

- *Locality Temporale*: il dato al quale si accede in memoria viene di solito richiesto nuovamente entro poco tempo (es. ciclo ripetuto)
- *Locality Spaziale*: i dati vicini a quello appena richiesto di solito sono richiesti entro poco tempo (es. scansione di un vettore)
  - Questi due principi sono alla base dello sfruttamento efficace della gerarchia di memoria!
  - Sappiamo che i dati richiesti vanno portati "vicino al processore", possibilmente con i dati a loro "vicini", in quanto in questo modo siamo in grado di ottenere h elevato, e quindi TMA ridotto, anche se la cache è molto inferiore alla RAM, la RAM è inferiore al disco, ecc.

## Quadro riassuntivo

Level	Memory Technology	Typical Size	Typical Access Time	Cost per Mbyte
Registers	D Flip-Flops	64 32-bit	2 -3 ns	N/A
L1 Cache (on chip)	SRAM	16 Kbytes	5 - 25 ns	\$100 - \$250
L2Cache (off chip)	SRAM	256 Kbytes	5 - 25 ns	\$100 - \$250
Main Memory	DRAM	256 Mbytes	60 - 120 ns	\$5 - \$10
Secondary Storage	Magnetic Disk	8 Gbytes	10 - 20 ms	\$0.10-\$0.20

## Miglioramenti delle DRAM (tempo di accesso m)

### ▪ **Page mode e EDO (Extended Data Out) DRAM:**

- si accede la RAM per "righe", memorizzate in buffer che funziona come una SRAM (veloce).
- a questo punto, mediante gli indirizzi di colonna, i diversi bit della riga possono essere recuperati in parallelo.
- Prestazioni: 120 ns -> 60 ns (page mode) -> 25 ns (EDO)

### ▪ **SDRAMs (S =synchronous): i bit sono espulsi dal buffer secondo un clock (~100 MHz).**

- questo evita il bisogno di passare gli indirizzi di colonna dei bit. E' utile per accessi burst. Il tempo per accedere ai bit **dopo che si è fatto l'accesso alla riga è = 8-10 ns !**

### ▪ **usano tecnologia DRAM, poco costo aggiuntivo porta vantaggi consistenti**

© 2004 Luciano Bononi

57

## Locality: permette di sfruttare la gerarchia

### ▪ **Sebbene le memorie permettano accesso casuale a larghi spazi di indirizzamento, i programmi in esecuzione accedono solo a piccole porzioni dei dati in ogni istante.**

### ▪ **Se un dato viene richiesto a un momento dell'esecuzione**

- esso tende a essere richiesto di nuovo, entro poco tempo (**Temporal locality**)
- i dati a esso vicini tendono a essere richiesti, entro poco tempo (**Spatial locality**)

### ▪ **Trucco: copiamo queste piccole porzioni di dati in una piccola e veloce memoria vicina alla CPU.**

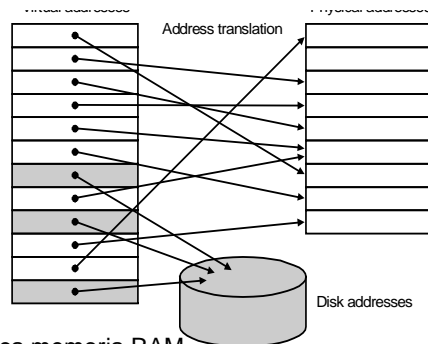
### ▪ **Programmatori e compilatori lavorano insieme sulle ipotesi della gerarchia per cooperare ad aumentare l'effetto del principio di locality, e quindi a trarre vantaggio dalla gerarchia.**

© 2004 Luciano Bononi

58

## Memoria Virtuale

- **La memoria RAM può funzionare come una cache rispetto alla memoria secondaria (disco):**



- **Obiettivi:**
  - **Illusione** di avere una gigantesca memoria RAM
  - **Rilocazione dei programmi (tanti programmi in RAM)**
  - **Protezione dei programmi (spazi di indirizzi separati)**

## Memoria Virtuale: terminologia

- **Pagina: equivalente del "blocco" di dimensione fissata.**
- **Page faults: equivalente del "miss"**
- **Virtual address: equivalente del "tag."**
  - **N.B. gli indirizzi virtuali sono a molti bit. La posizione in RAM fisica di una pagina viene mantenuta attraverso tabelle di paginazione (non attraverso i campi ID o TAG).**
- **Physical address: è l'indirizzo effettivo nella RAM fisica del dato puntato dal virtual address. Non ha l'equivalente per la cache.**
- **Memory mapping (address translation): è la conversione degli indirizzi virtuali in indirizzi fisici. Non ha equivalente per cache.**

## Trasmissione di informazione

---

- **Come realizzare trasmissione di informazione binaria?**
  - **l'informazione può essere codificata e trasmessa sotto forma di dati digitali (bit): Claude Elwood Shannon (1916, 2001)**
- **Questo principio consente di usare tecniche digitali per codificare e trasmettere l'informazione su bus interni di un architettura di calcolo, oppure su reti di calcolatori**
- **Oltre a rendere possibile trasmettere informazione nel formato "ideale" e comprensibile per i calcolatori elettronici, questo principio ha altri vantaggi, che vedremo parlando delle reti di calcolatori**