

# Calcolatori Elettronici

## Parte III: L'organizzazione generale del calcolatore

Prof. Riccardo Torlone

Università Roma Tre

# Terminologia di base

- **Calcolatore elettronico:** macchina fatta di dispositivi elettronici che può risolvere problemi eseguendo istruzioni fornitegli
- **Programma:** sequenza di istruzioni in un linguaggio
- **Linguaggio macchina:** eseguibile direttamente da un calcolatore (binario)

Con dispositivi elettronici si possono eseguire direttamente solo un numero limitato di istruzioni semplici (costi)

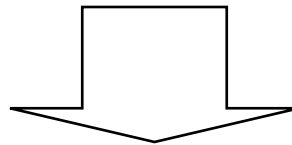
I linguaggi macchina non sono adatti per le persone

# Problema



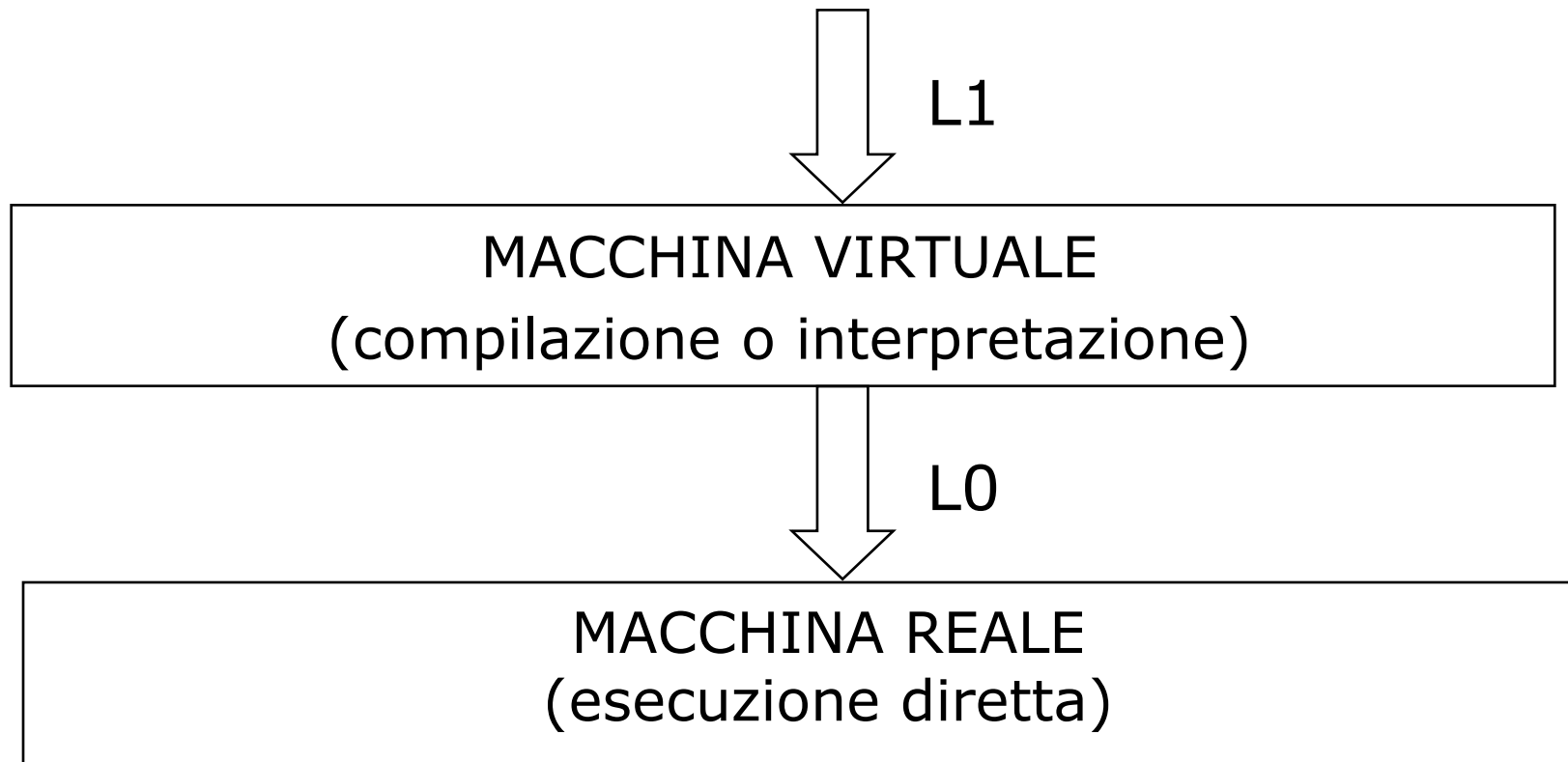
Voglio fare X (programma complesso  
in un linguaggio L1 di alto livello)

*Traduzione o  
interpretazione*



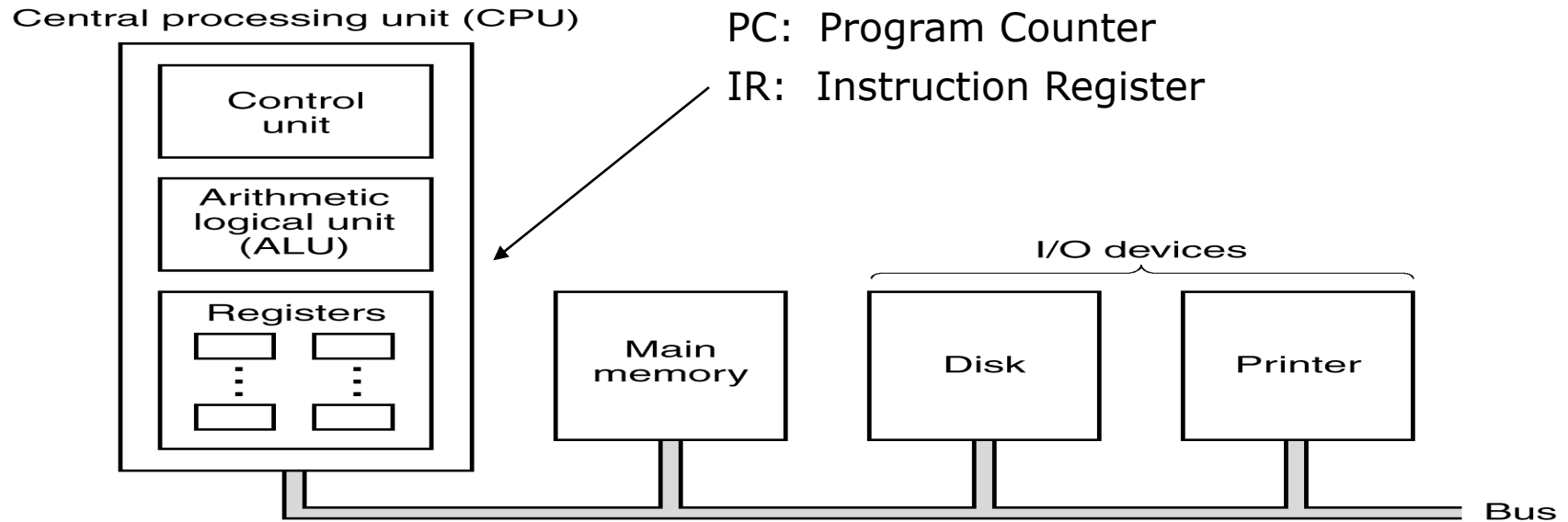
So fare Y (programma di istruzioni  
semplici in un linguaggio macchina L0)

# Soluzione adottata



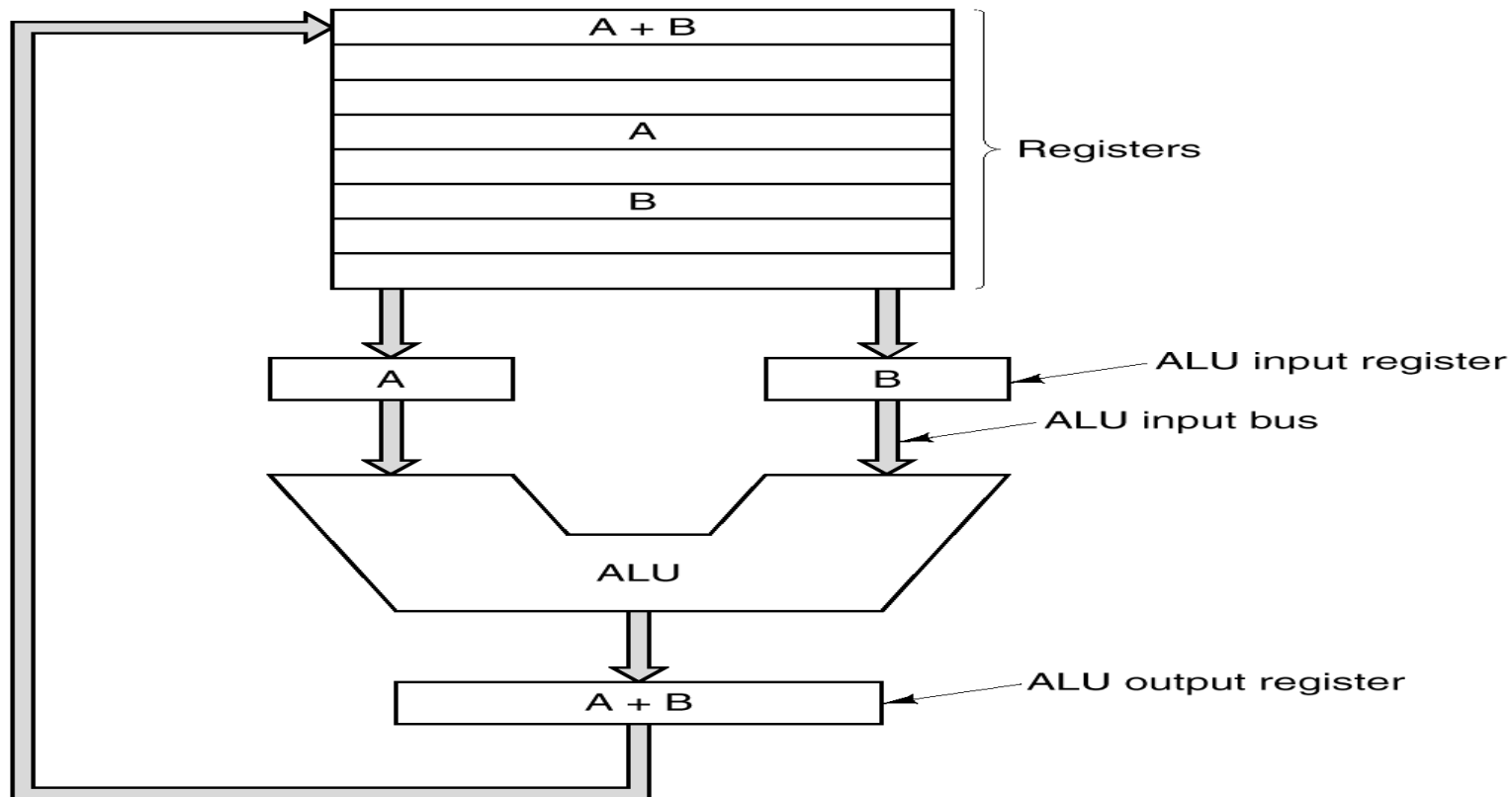
Se L0 ed L1 sono troppo diversi il problema si  
decompone introducendo livelli intermedi

# Struttura del computer



- La memoria contiene sia i dati che le istruzioni
- Il contenuto dei registri può essere scambiato con la memoria e l'I/O
- Le istruzioni trasferiscono i dati e modificano il contenuto dei registri
- Registri particolari:
  - PC: indirizza l'istruzione corrente
  - IR: contiene l'istruzione corrente

# Struttura della CPU



- Esecuzione di operazioni aritmetiche e logiche sui dati contenuti nei registri
- Spostamento di dati fra registri e fra registri e memoria
- Ciclo elementare: due operandi sono inviati alla ALU e il risultato è messo in un registro

# Il ciclo Fetch-Decode-Execute

L'esecuzione di ciascuna istruzione nella CPU richiede i seguenti passi:

1. Carica l'istruzione da memoria in IR (Instruction Register) (**Fetch**)
2. Incrementa PC (Program Counter)
3. Decodifica l'istruzione (**Decode**)
4. Se l'istruzione usa un dato in memoria calcolane l'indirizzo
5. Carica l'operando in un registro
6. Esegui l'istruzione (**Execute**)
7. Torna al passo 1. Per l'esecuzione dell'istruzione successiva

Accessi alla memoria sono effettuati sempre al passo 1, e non sempre ai passi 4 e 5

# Esecuzione e Interpretazione

## Esecuzione diretta

- Le istruzioni possono venire eseguite direttamente dai circuiti hardware
- Approccio molto complesso:
  - Repertorio di istruzioni limitato
  - Progettazione dell'HW complessa
  - Esecuzione molto efficiente

## Interpretazione

- L'hardware può eseguire solo alcune operazioni elementari molto semplici dette microistruzioni
- Ciascuna istruzione è scomposta in una successione di microistruzioni poi eseguite dall'hardware
- Vantaggi:
  - Repertorio di istruzioni esteso
  - HW più compatto
  - Flessibilità di progetto



# La Microprogrammazione

L'HW può eseguire *microistruzioni*:

- Trasferimenti tra registri
- Trasferimenti da e per la memoria
- Operazioni della ALU su registri

Ciascuna istruzione viene scomposta in una *sequenza di microistruzioni*

L'unità di controllo della CPU esegue un *microprogramma* per effettuare l'*interpretazione* delle istruzioni macchina

Il microprogramma è contenuto in una memoria ROM sul chip del processore

Vantaggi:

- Disegno strutturato
- Semplice correggere errori
- Facile aggiungere nuove istruzioni

# CISC e RISC

Architetture **RISC** (*Reduced Instruction Set Computer*):

- Esecuzione diretta
- Repertorio ristretto (alcune decine)
- Istruzioni prevalentemente su registri
- *Una istruzione per ciclo di macchina (del data path)*

Architetture **CISC** (*Complex Instruction Set Computer*):

- Interpretazione tramite microprogramma
- Repertorio esteso (alcune centinaia)
- Istruzioni anche su memoria
- *Molti cicli di macchina per istruzione*

Esempi:

- PowerPC, SPARC, MIPS, ARM: RISC
- VAX (DEC), Pentium II/III/IV/i7 (Intel): CISC

All'inizio degli anni '80 i progettisti di sistemi veloci riconsiderano l'approccio dell'*esecuzione diretta*

# Principi progettuali dei computer moderni

- Eseguire tutte le istruzioni dall'hardware
- Massimizzare la velocità con la quale le istruzioni sono eseguite misurata in MIPS (Millions of Instr. per Second)
- Semplificare la decodifica delle istruzioni: formati molto regolari
- Limitare i riferimenti alla memoria (solo LOAD e STORE)
- Ampliare il numero di registri

**N.B.** Questi principi sono tipici della filosofia RISC ma anche le architetture CISC vi si adeguano, almeno in parte

# Introduzione del parallelismo



# Vari Tipi di Parallelismo

Il parallelismo è ormai l'unica strada per aumentare le prestazioni  
Limite di un'esecuzione sequenziale: velocità della luce (30 cm in 1 nsec)

Due tipi di parallelismo:

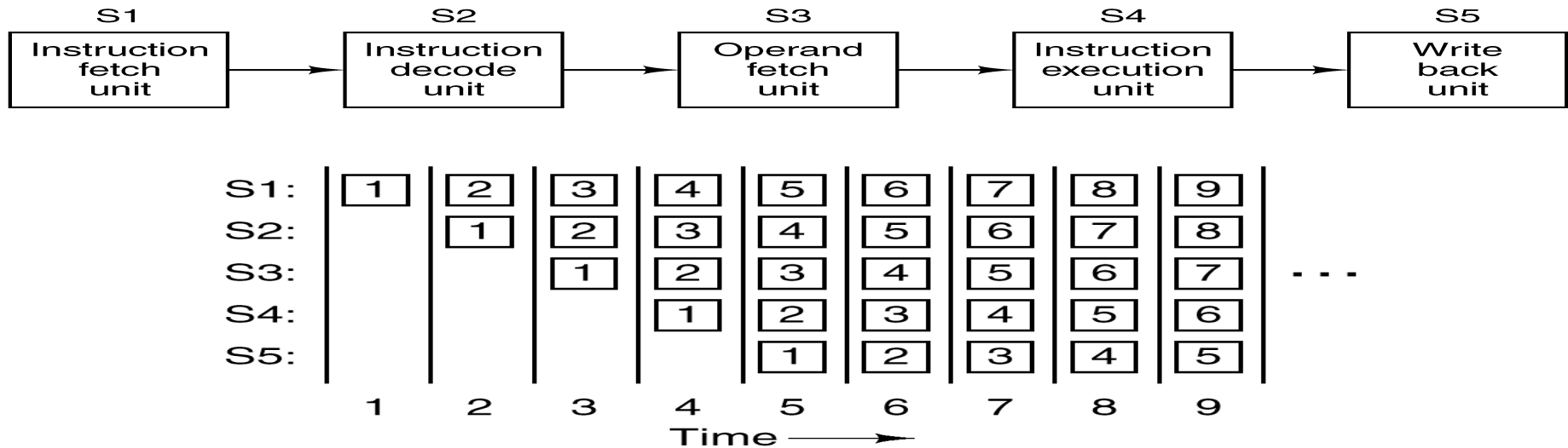
## **A)** a livello di istruzioni

- Diverse istruzioni eseguite insieme
- Diverse fasi della stessa istruzione eseguite insieme

## **B)** a livello di processori

- Molti processori lavorano insieme allo stesso problema
- Fattori di parallelismo molto elevati
- Diversi tipi di interconnessione e di cooperazione (più o meno stretta)

# Parallelismo a livello di istruzioni: Pipelining



- Ciascuna istruzione è divisa in fasi
- L'esecuzione avviene in una *pipeline* a più stadi
- Più istruzioni in esecuzione contemporanea
- Una istruzione completata per ogni ciclo

**N.B.** Si guadagna un fattore pari al numero di stadi della pipeline

# Caratteristiche di una pipeline

Una pipeline consente un compromesso tra:

- Latenza: tempo per eseguire una istruzione
- Ampiezza di banda: numero di istruzioni completate per unità di tempo misurata in MIPS (milioni di istruzioni al secondo) - oggi in GFLOPS o TFLOPS ( $10^9$  o  $10^{12}$  istruzioni in virgola mobile al secondo)

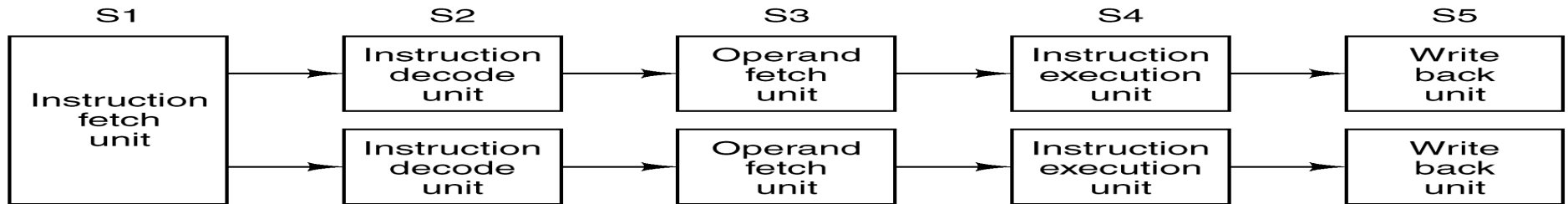
*Con:*

- Velocità di clock =  $T$  nsec (periodo del segnale di clock)
- Numero di stadi =  $n$

*Abbiamo:*

- **Latenza** =  $nT$
- **Ampiezza di banda** = 1 istr. ogni  $T$  nsec, ovvero:  $10^9/T$  istr. ogni sec., ovvero:  $1000/T$  MIPS

# Architetture Superscalari



Architetture nelle quali si avviano più istruzioni (4-6) insieme  
Si aumenta il parallelismo avendo più di una pipeline nel microprocessore

Le pipeline possono essere specializzate:

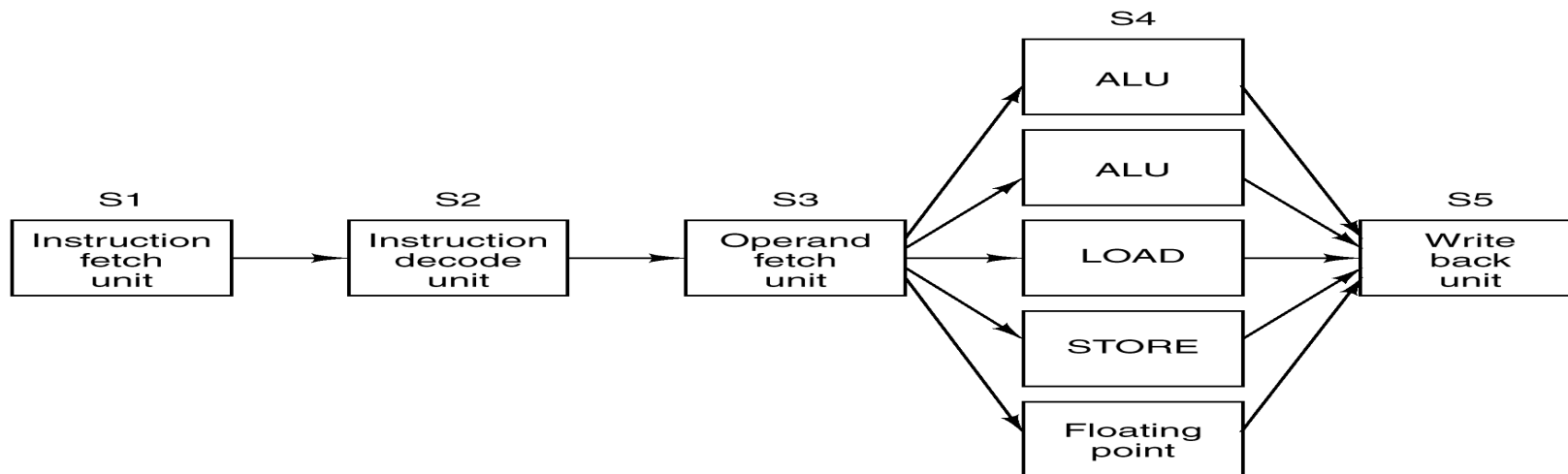
- Una versione dell'i7 ha diverse pipeline a più stadi
- Può eseguire fino a 6 micro-istruzioni in parallelo

**Problema:** compatibilità dell'esecuzione parallela

- Indipendenza tra le istruzioni
- Ciascuna istruzione non deve utilizzare i risultati dell'altra



# Unità Funzionali Multiple



- Variante: solo lo stadio più lento della pipeline (che condiziona la velocità) viene parallelizzato
- La CPU contiene al suo interno diverse unità funzionali indipendenti
- Architettura adottata nei processori Intel Core

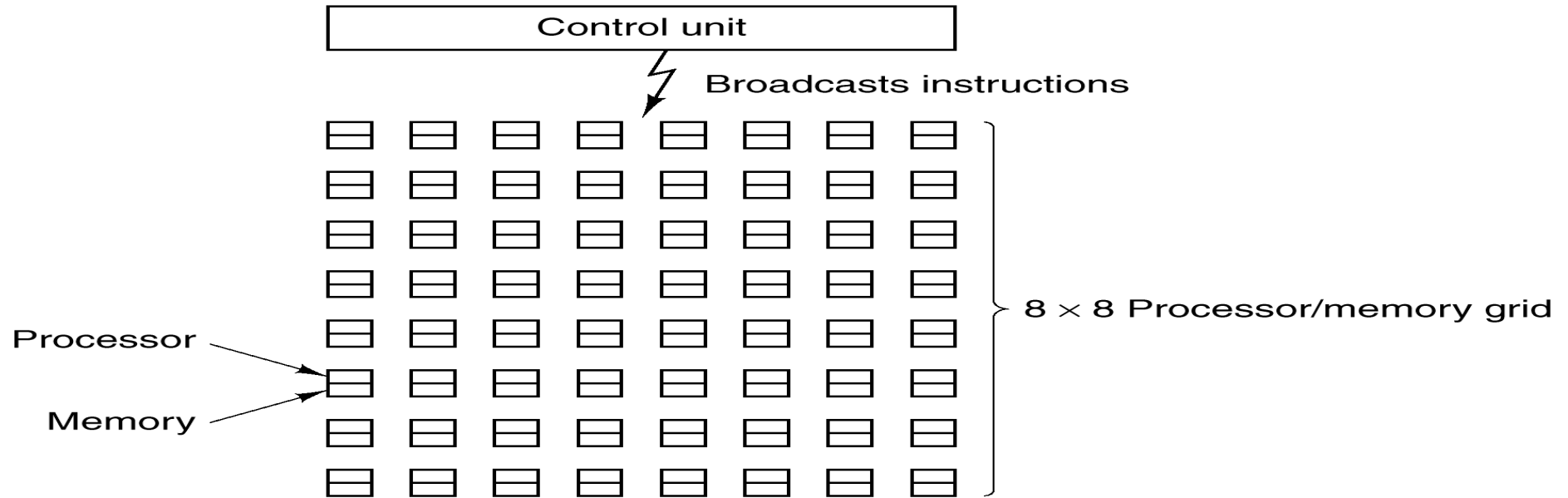
# Parallelismo a livello di processori

- Miglioramento delle prestazioni con parallelismo a livello di istruzioni: 5-10 volte
- Per migliorare ancora: CPU multiple
- Approcci:
  - Data parallelism (SIMD)
    - Processori matriciali
    - Processori vettoriali
    - GPU
  - Task parallelism (MIMD)
    - Multiprocessori
    - Multicore
    - Multicomputer

```
[ || (int i : 100) array[i]++; ]
```

```
[ a++; || b+c; ]
```

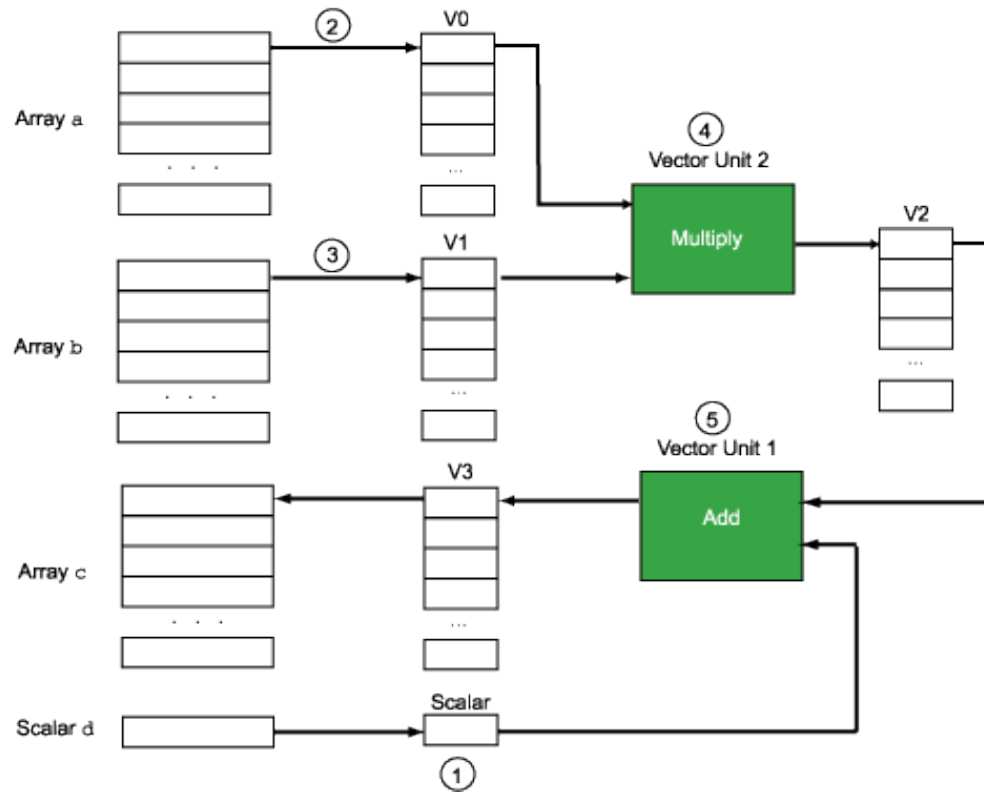
# Array Processor



- Processori identici ed autonomi (ma non indipendenti) eseguono le stesse istruzioni su dati diversi
- Es.: ILLIAC IV (1972) - 50 Mflops

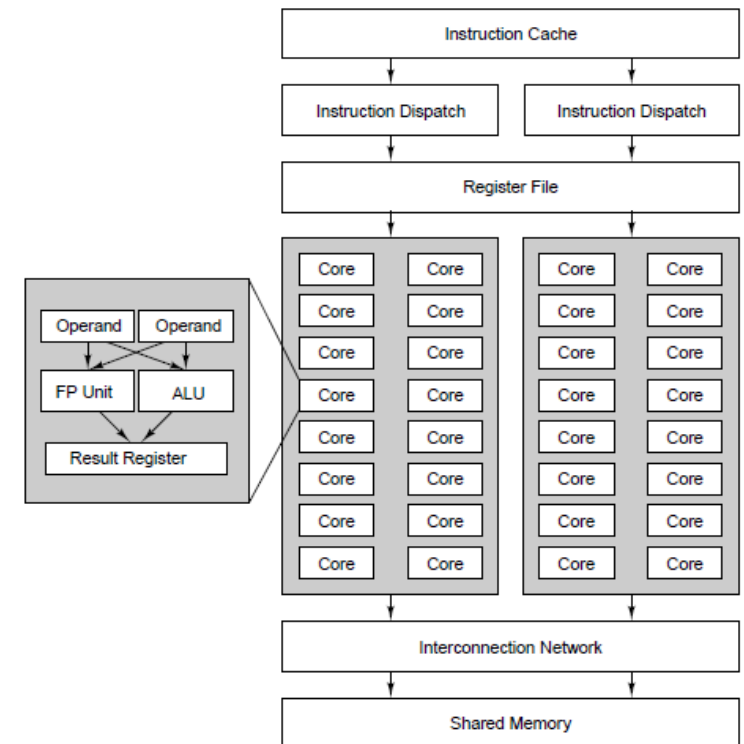
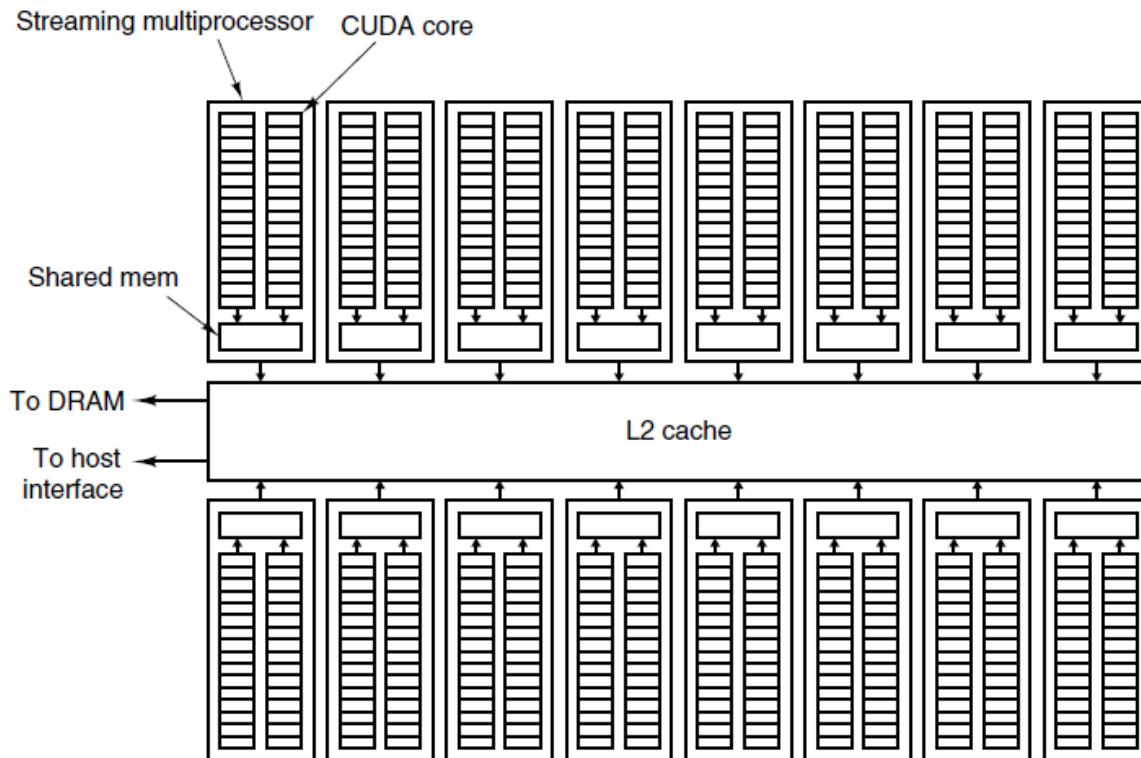
# Vector Processor

Vector processor: unità aritmetiche che operano su registri vettoriali (supercomputer)

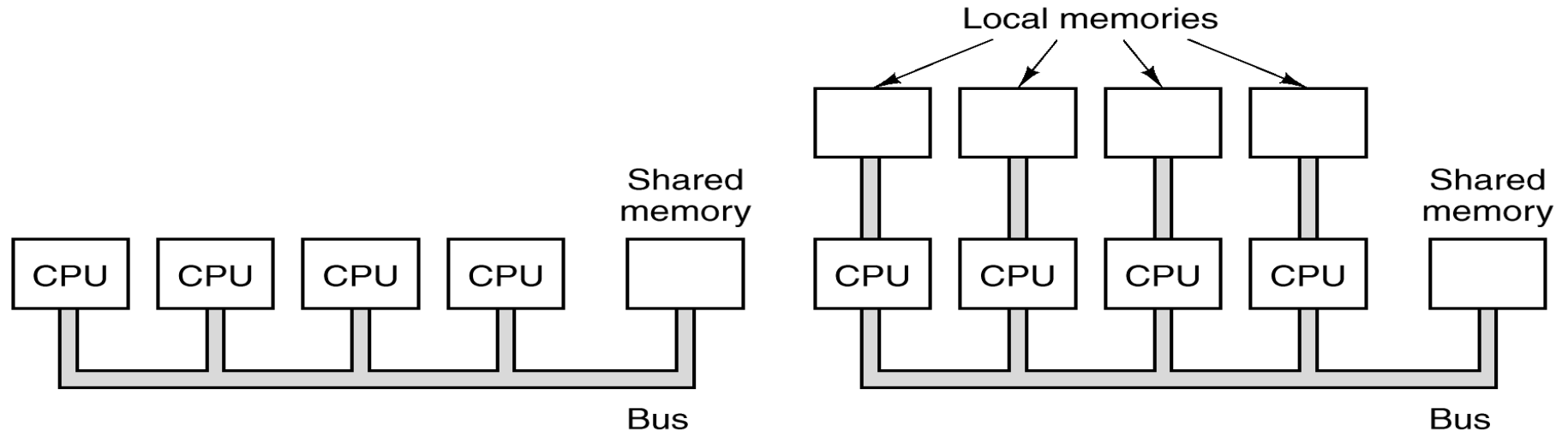


# GPU

- Operazioni comuni su pixel, vertici, archi, figure
- Es.: Nvidia Fermi GPU (2009)
  - 16 processori stream SIMD
  - Ogni processore ha 32 core
  - Fino a 512 operazioni per ciclo di clock



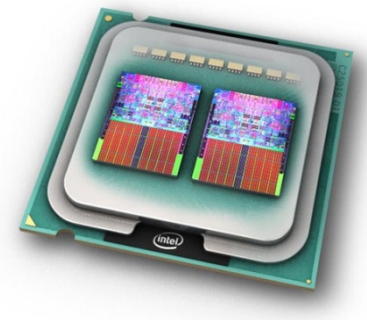
# Multiprocessori



- Le CPU lavorano indipendentemente
  - Shared memory: il bus può divenire collo di bottiglia
  - Private memory: contiene il codice e parte dei dati
- Scambio dati tramite la shared memory

# Architetture multicore

- La CPU è composta da più core, ovvero da più nuclei di processori fisici montati sullo stesso package
- Ogni core:
  - è un processore indipendente
  - può essere dotato di cache autonoma
- Architetture omogenee (core identici) o eterogenee
- Ogni core può essere multiscalare
- Accoppiamento dei core:
  - stretto: shared cache
  - lasco: private cache
- Nascono a partire dal 2003:
  - IBM: PowerPC
  - Intel: Pentium D, Core 2, Core I3-i5-i7
  - AMD: Opteron, Athlon, Phenom



# Multicomputer

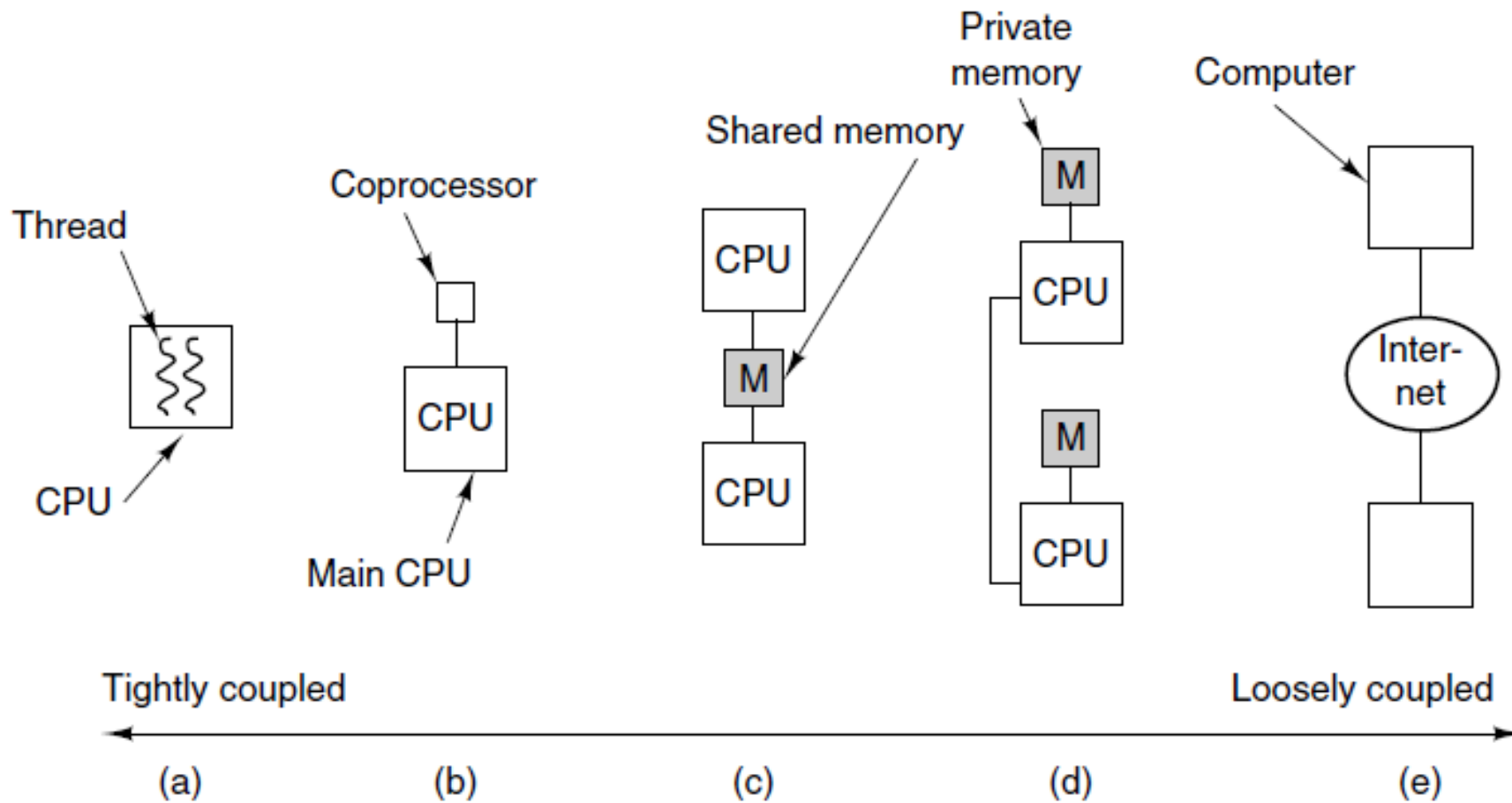
- I singoli elementi sono normali Workstation o PC
- Comunicazione tramite scambio di messaggi (shared nothing)



MIMD (Multiple Instruction Multiple Data)



# Le varie forme di parallelismo



# La Memoria Centrale

- Contiene sia i programmi che i dati
- Memorizzazione binaria (bit)
- Cella (o locazione): unità indirizzabile
  - byte: 8 bit (minimo indirizzabile)
  - word: insieme di K byte (K dipende dall'architettura)
- Indirizzo (della cella): tramite il quale la CPU accede al dato nella cella
- Indirizzi binari a m bit: spazio di indirizzamento  $2^m$  celle

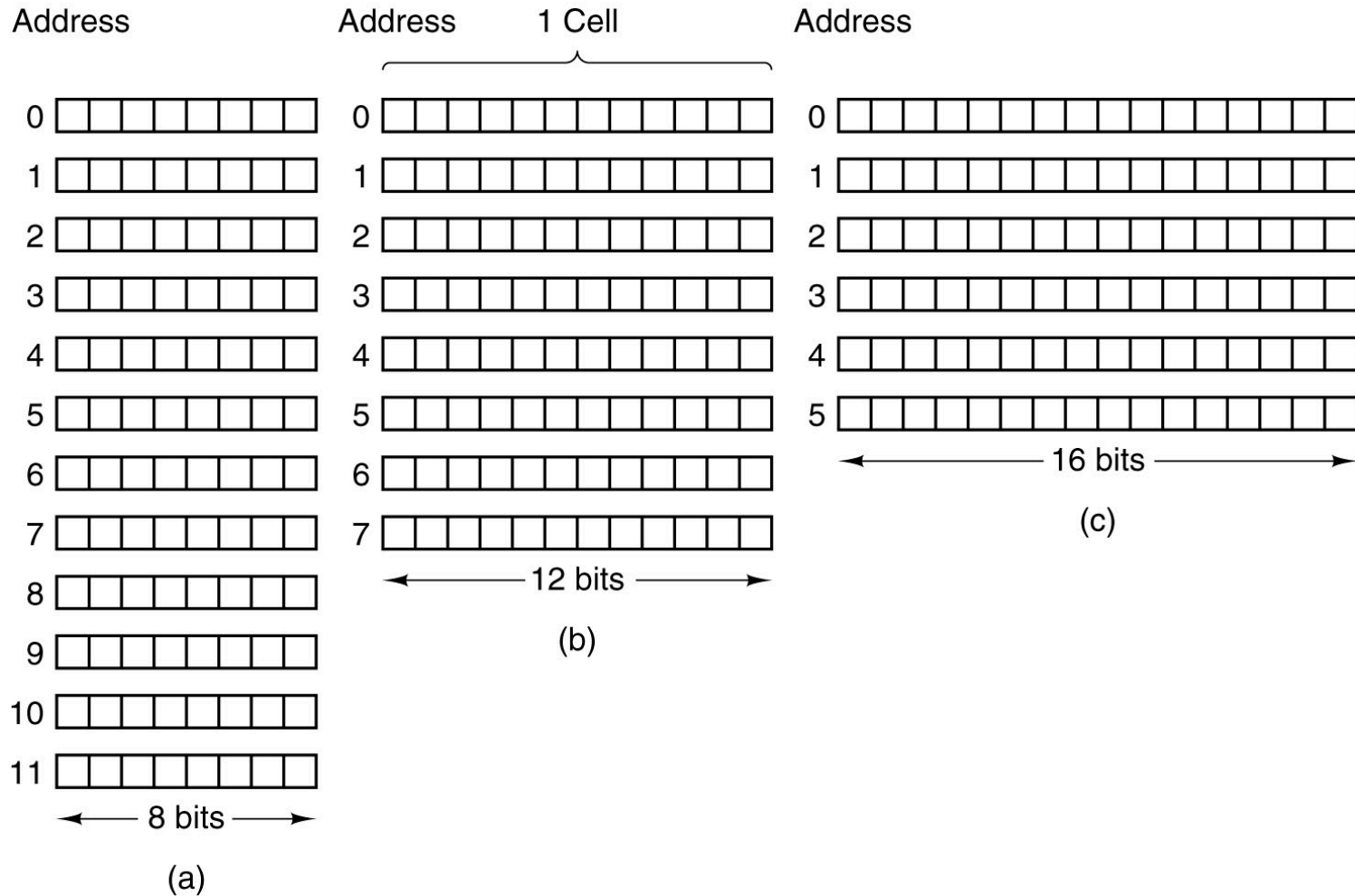
## **ES** Pentium IV

- Architettura a 32 bit
- Registri e ALU a 32 bit
- Word di 4 byte 32 bit
- Indirizzi a 32 bit
- Spazio indirizzabile  $2^{32} = 4$  GB  
(64GB con opportuni accorgimenti)

# Organizzazione della memoria

Diverse possibilità

- Esempio con 96 bit totali:

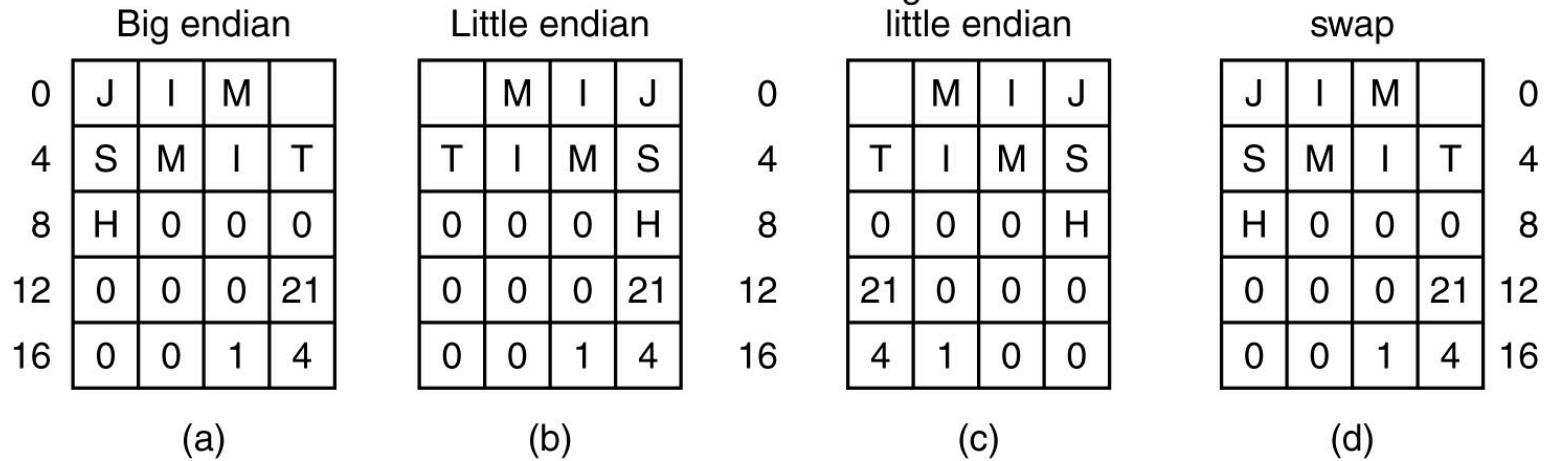
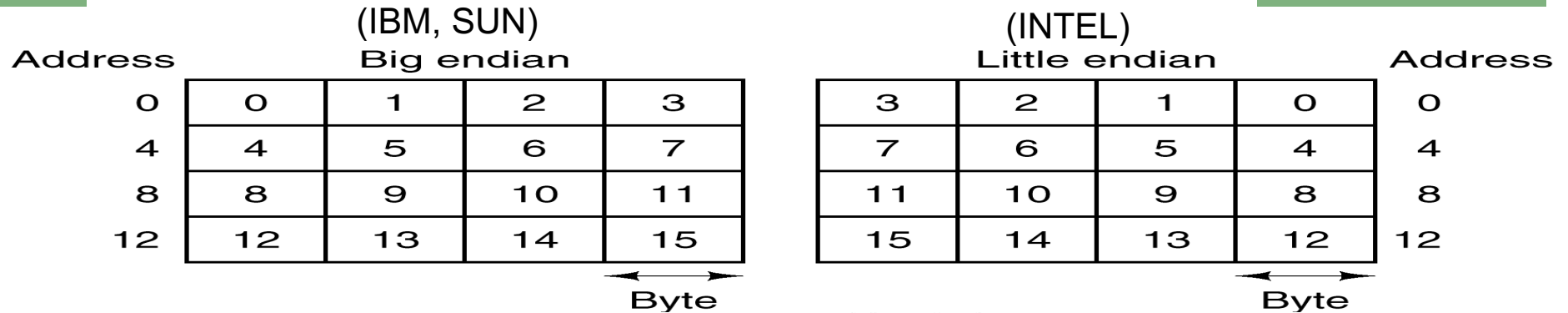


# Dimensione locazioni di memoria

Diverse soluzioni possibili

Computer	Bits/cell
Burroughs B1700	1
IBM PC	8
DEC PDP-8	12
IBM 1130	16
DEC PDP-15	18
XDS 940	24
Electrologica X8	27
XDS Sigma 9	32
Honeywell 6180	36
CDC 3600	48
CDC Cyber	60

# Ordinamento dei byte



- Diversi ordinamenti dei byte nella parola nella rappresentazione di stringhe
- Crea problemi nei trasferimenti fra computer di tipi diversi

# Codici a correzione di errore

Tecniche per garantire maggiore affidabilità nella registrazione / trasmissione di informazioni binarie

Recupero degli errori hardware tramite codifiche ridondanti

Codifiche con  $n = m + r$  bit

- $n$  bit complessivi codifica
- $m$  bit dati
- $r$  check bit (ridondanti)

Si utilizza solo un sottoinsieme delle codifiche (codifiche valide)

ES

**Codice** con  $n=10$ ,  $m=2$ ,  $r=8$

0000000000  
0000011111  
1111100000  
1111111111

}  $2^m = 4$  codifiche valide (su  $2^{10}$ )

# Distanza di Hamming

Distanza di Hamming tra due codifiche: numero di bit diversi:

0101 e 1001 sono a distanza 2

Distanza di Hamming di un codice:  $h$  = distanza di Hamming minima tra due codifiche valide del codice

ES

0000000000  
0000011111  
1111100000  
1111111111

} Distanza di Hamming del codice  $h=5$

- Per **rilevare** errori su  $k$  bit occorre che sia:
  - almeno  $h = k + 1$  ovvero  $k \leq h - 1$
- Per **correggere** errori su  $k$  bit occorre che sia:
  - almeno  $h = 2k + 1$  ovvero  $k \leq (h - 1)/2$

# Codici a correzione di errore (Esempio)

ES

Codice con  $n=10$ ,  $m=2$ ,  $r=8$

0000000000  
0000011111  
1111100000  
1111111111

Distanza di Hamming = 5

$h=5=k+1 \Rightarrow E'$  possibile *rilevare* errori quadripli

0000011111  $\rightarrow$  1111011111

1111011111 viene riconosciuto come errato

$h=5=2k+1 \Rightarrow E'$  possibile *correggere* errori doppi

0000011111  $\rightarrow$  1100011111

1100011111 viene corretto in 0000011111



## Rilevazione di errore singolo (controllo di parità)

- Nel caso più semplice si vogliono solo rilevare errori singoli
- Basta aggiungere un solo check bit  $r=1$ ,  $n=m+1$
- Bit di parità: scelto in modo che il numero complessivo di 1 nella codifica sia sempre pari (o dispari)
- Questo codice ha distanza  $h=2$
- Errore rilevato da circuiti molto semplici
- Le memorie segnalano *parity error* quando un errore si manifesta

ES.    11011010 bit di parità:1 → 11011010**1** OK  
      01100101 bit di parità:0 → 0110**1**101**0** Error

# Correzione di errore singolo

- m data bit, r check bit, n bit totali
- $2^m$  codifiche valide
- n codifiche errate a distanza 1 da ciascuna delle valide
- Ogni codifica valida ne richiede in tutto n+1

ES:

La codifica:

0000

Richiede le codifiche errate:

1000

0100

0010

0001

# Correzione di errore singolo

- Se ogni codifica valida ne richiede  $n+1$  deve essere:

$$(n+1) 2^m \leq 2^n \quad \text{cioè} \quad (m+r+1) \leq 2^r$$

Word size	Check bits	Total size	Percent overhead
8	4	12	50
16	5	21	31
32	6	38	19
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

**N.B.** Al crescere di  $m$  l'*overhead* scende

# Esercizio 1

Riferendosi all'organizzazione generale di un calcolatore, indicare se le seguenti affermazioni sono vere o false.

- Nelle architetture RISC le istruzioni macchina vengono tradotte in microistruzioni che vengono poi eseguite dall'hardware.
- Le tecniche del pipeline non è compatibile con una architettura superscalare.
- Una architettura con indirizzi a 16 bit con indirizzamento al byte non può gestire una memoria più grande di 64KB.
- In un processore con pipeline a 4 stadi e un clock con periodo di 2 nsec una istruzione macchina richiede 2 nsec per essere eseguita.
- Un processore con pipeline a 5 stadi e un clock con periodo di 5 nsec ha un'ampiezza di banda di 200 MIPS.
- L'ampiezza di banda (numero di istruzioni eseguite al secondo a regime) di un processore con pipeline non dipende dal numero di stadi della pipeline.
- In una architettura con pipeline sono necessari più cicli di clock per completare una istruzione macchina.
- In linea di principio, se si raddoppia la frequenza del clock si dimezza la latenza e si raddoppia l'ampiezza di banda.

# Esercizio 1

Riferendosi all'organizzazione generale di un calcolatore, indicare se le seguenti affermazioni sono vere o false.

- Nelle architetture RISC le istruzioni macchina vengono tradotte in microistruzioni che vengono eseguite dall'hardware. **FALSO**
- La tecnica del pipeline non è compatibile con una architettura superscalare. **FALSO**
- Una architettura con indirizzamento a 6 bit con indirizzamento al byte non può gestire una memoria principale di 64KB. **VERO**
- In un processore con pipeline a 4 stadi e un clock con periodo di 2 nsec una istruzione macchina richiede 8 nsec per essere eseguita. **FALSO**
- Un processore con pipeline a 4 stadi e un clock con periodo di 5 nsec ha un'ampiezza di banda di 200 MIPS. **VERO**
- L'ampiezza di banda (numero di istruzioni eseguite al secondo a regime) di un processore a pipeline non dipende dal numero di stadi della pipeline. **VERO**
- In una architettura con pipeline a 4 stadi sono necessari più cicli di clock per completare una istruzione macchina. **VERO**
- In linea di principio, se si raddoppia la frequenza del clock si dimezza la latenza e si raddoppia l'ampiezza di banda. **VERO**

## Esercizio 2

Con riferimento ai codici a rilevazione e correzione di errore indicare se le seguenti affermazioni sono vere o false.

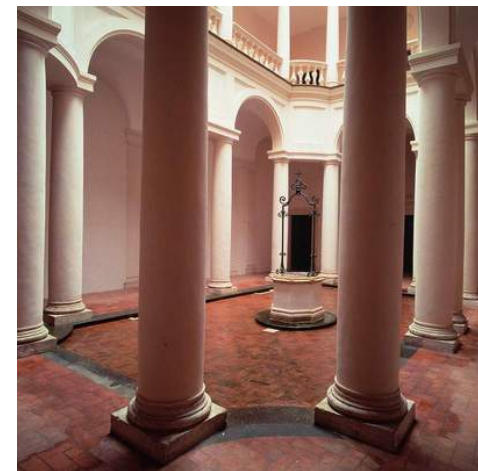
- La distanza di Hamming tra una codifica e il suo complemento a uno è pari alla lunghezza della codifica.
- Con distanza di Hamming  $h=3$  è possibile correggere 2 errori.
- Il numero di bit di controllo necessari per rilevare un errore singolo su un codice a 8 bit è minore rispetto al numero bit di controllo necessari per un codice a 16 bit.
- La distanza di Hamming nel codice composto solo dalle parole 1100, 0011 e 1111 è 4.
- La percentuale di bit di controllo rispetto alla lunghezza complessiva di un codice a correzione di errore singolo diminuisce all'aumentare della lunghezza del codice.
- Per rilevare  $r$  errori è necessario che un codice abbia una distanza di Hamming pari a  $2r+1$ .
- Un bit di parità permette solo di rilevare errori singoli.
- Se in una parola si commette un errore singolo ma si conosce la sua posizione, il bit di parità è sufficiente a correggerlo.

## Esercizio 2

Con riferimento ai codici a rilevazione e correzione di errore indicare se le seguenti affermazioni sono vere o false.

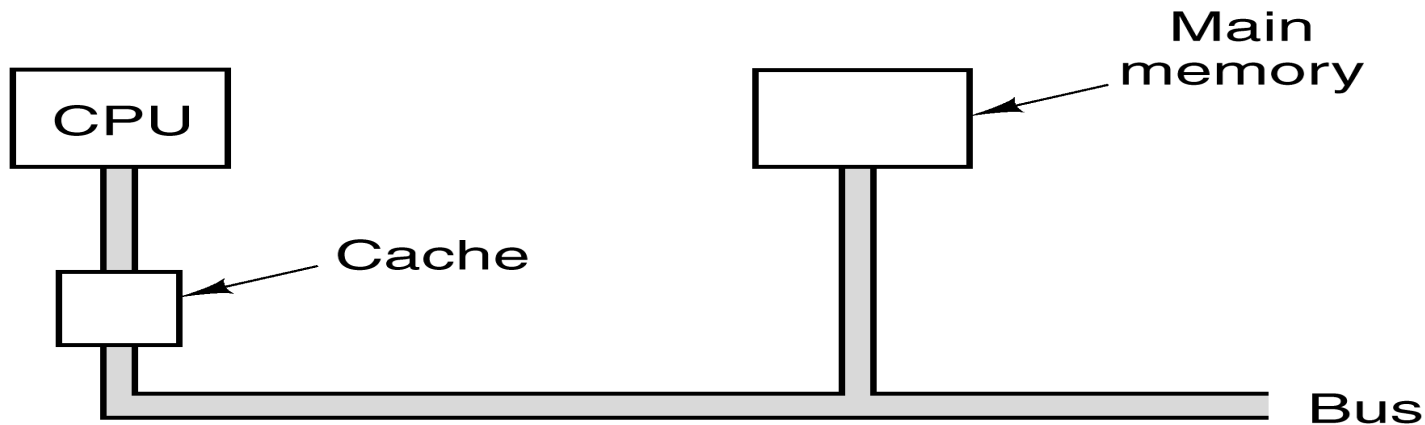
- La distanza di Hamming di un codice e il suo complemento a uno e pari alla lunghezza del codice. **VERO**
- Con distanza di Hamming 2 è possibile correggere 2 errori. **FALSO**
- Il numero di bit di controllo necessari per rilevare un errore singolo su un codice a 8 bit è minore rispetto al numero bit di controllo necessari per un codice a 16 bit. **FALSO**
- La distanza di Hamming di un codice composto solo dalle parole 1100, 0011 e 1111 è 2. **FALSO**
- La percentuale di bit di controllo rispetto alla lunghezza complessiva di un codice a rilevazione di errore singolo diminuisce all'aumentare della lunghezza del codice. **VERO**
- Per rilevare  $r$  errori è necessario che un codice abbia una distanza di Hamming pari a  $r+1$ . **FALSO**
- Un bit di parità è sufficiente solo di rilevare errori singoli. **VERO**
- Se in una parola si commette un errore singolo ma si conosce la sua posizione, il bit di parità è sufficiente a correggerlo. **VERO**

# Caching...





# Memorie Cache



- La memoria è sempre più lenta della CPU e tende a rallentarla
- Memorie veloci sono disponibili ma solo per piccole dimensioni
- La *cache* (da *catcher*) funziona alla velocità del processore, e quindi *nasconde* la "lentezza" della memoria
- Contiene le ultime porzioni di memoria acceduta: se la CPU vuole leggere una di esse evita un accesso a memoria
- Funziona bene a causa della *località* degli accessi

# Cache Hit Ratio

Se una parola viene letta  $k$  volte di seguito,  $k - 1$  volte sarà trovata in cache

- Cache hit ratio:

$$H = (k - 1) / k$$

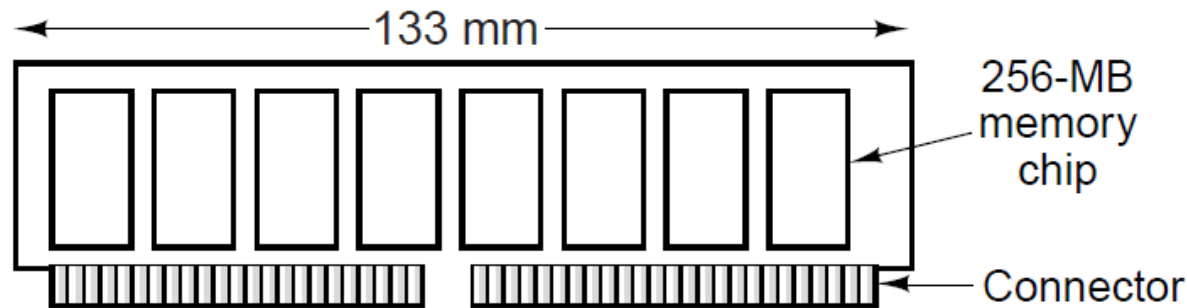
- Tempo medio di accesso a memoria:
  - $m$ : tempo di accesso della memoria
  - $c$ : tempo di accesso della cache

$$A = c + (1 - H)m$$

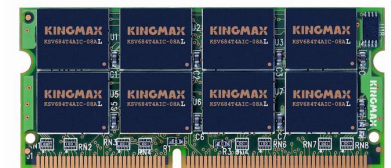
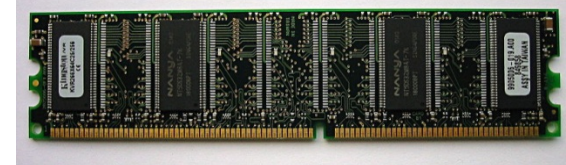
La memoria è organizzata in blocchi

Per ogni *cache miss* un intero blocco è spostato in cache

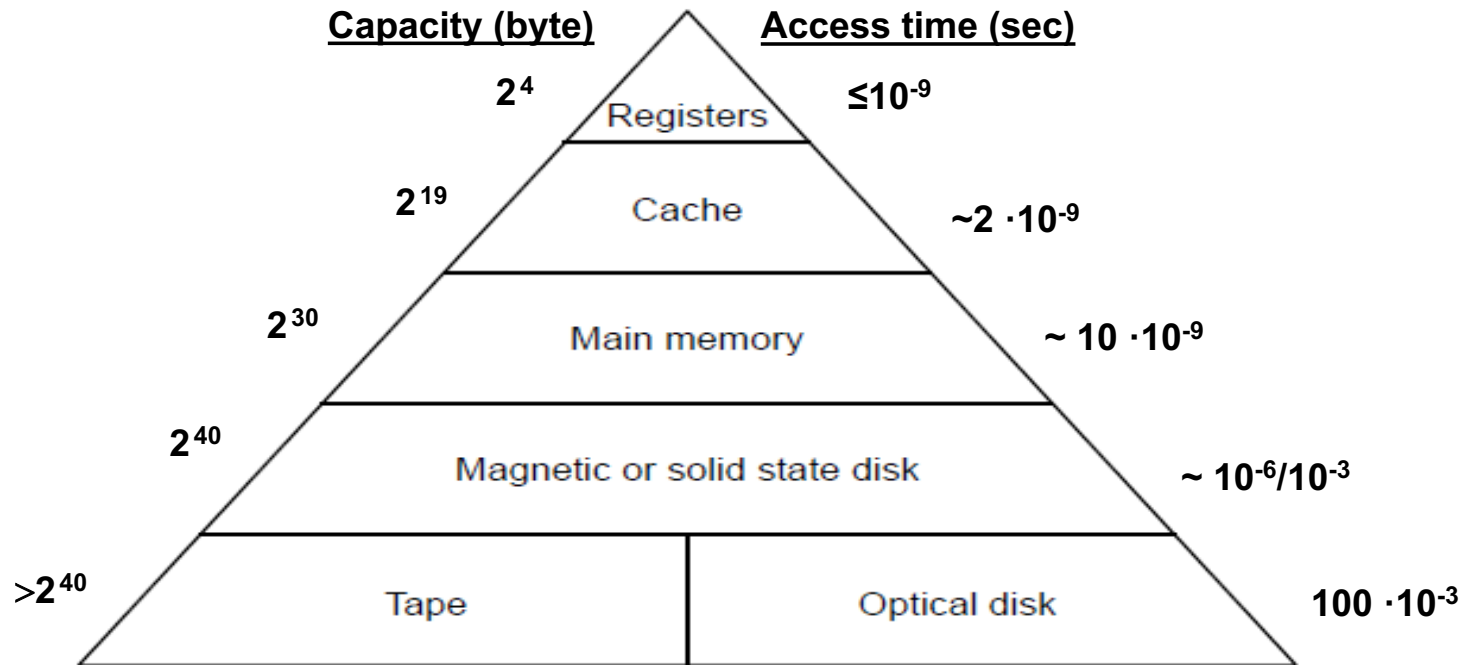
# Tipologie schede memoria



- SIMM (Single Inline Memory Module)
  - 72 piedini, 32 bit, 8-16 chip, 128 MByte
  - A coppie nel Pentium (bus dati 64 bit)
- DIMM (Double Inline Memory Module)
  - 120/240 piedini, 64 bit, 8 chip, 256 MByte
- SO-DIMM (Small Outline DIMM)
  - Per notebook di dimensioni più piccole
- Alcune hanno bit di parità altre no



# Gerarchie di memoria



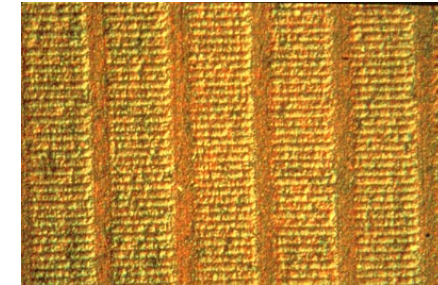
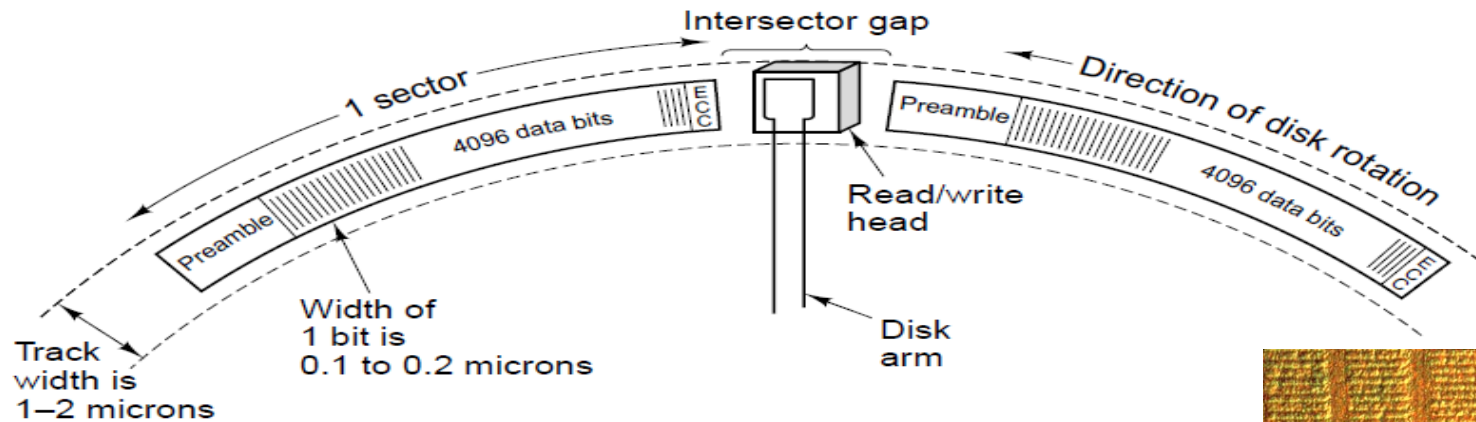
Scendendo nella gerarchia:

- Cresce il *tempo di accesso*
- Aumenta la *capacità*
- Diminuisce il *costo per bit*

Solo il livello più alto della gerarchia è a contatto con la CPU

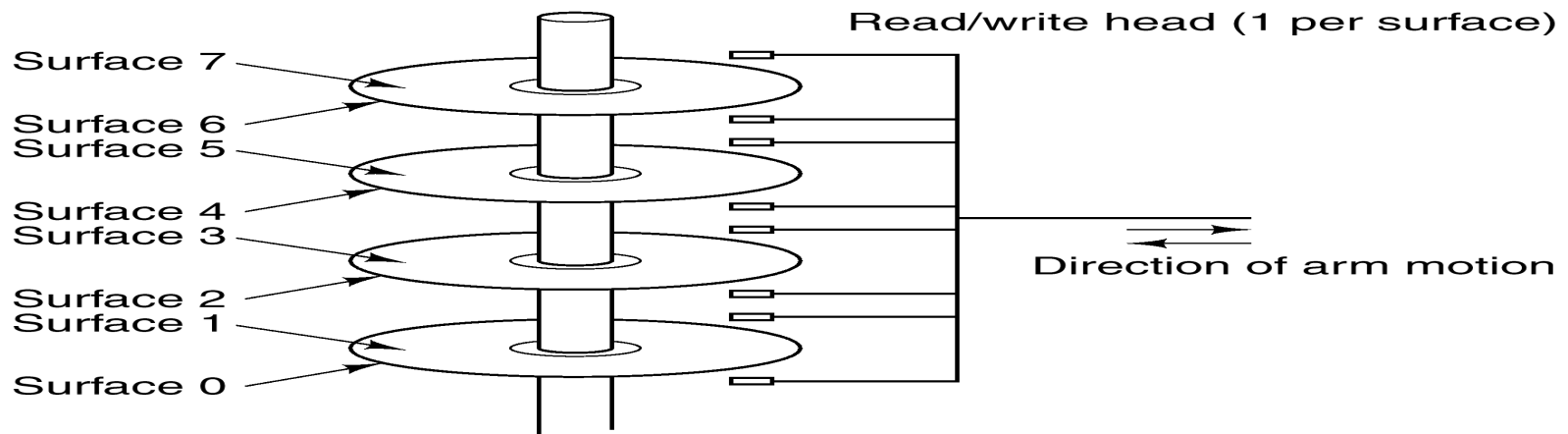
Migrazione dei dati fra livelli della gerarchia

# Dischi magnetici



- Dimensione: <10cm, Densità: 25Gb/cm
- Registrazione seriale su *tracce* concentriche
- 50.000 tracce/cm (larghe ~200nm)
- Dischi ad alta densità con bit registrati *perpendicolarmente*
- Tracce divise in *settori* contenenti i dati, un *preambolo* e un *ECC* (Error-Correcting Code) (la *capacità formattata* scende del 15%)
- Velocità di rotazione costante (5.400-10.800 RPM)
- Velocità di trasferimento di 150 MB/sec (1 settore in 3.5  $\mu$ sec)
- *Burst rate*: velocità da quando la testina è sopra il primo bit
- *Sustained rate*: velocità di trasferimento in un certo intervallo

## Dischi magnetici (2)

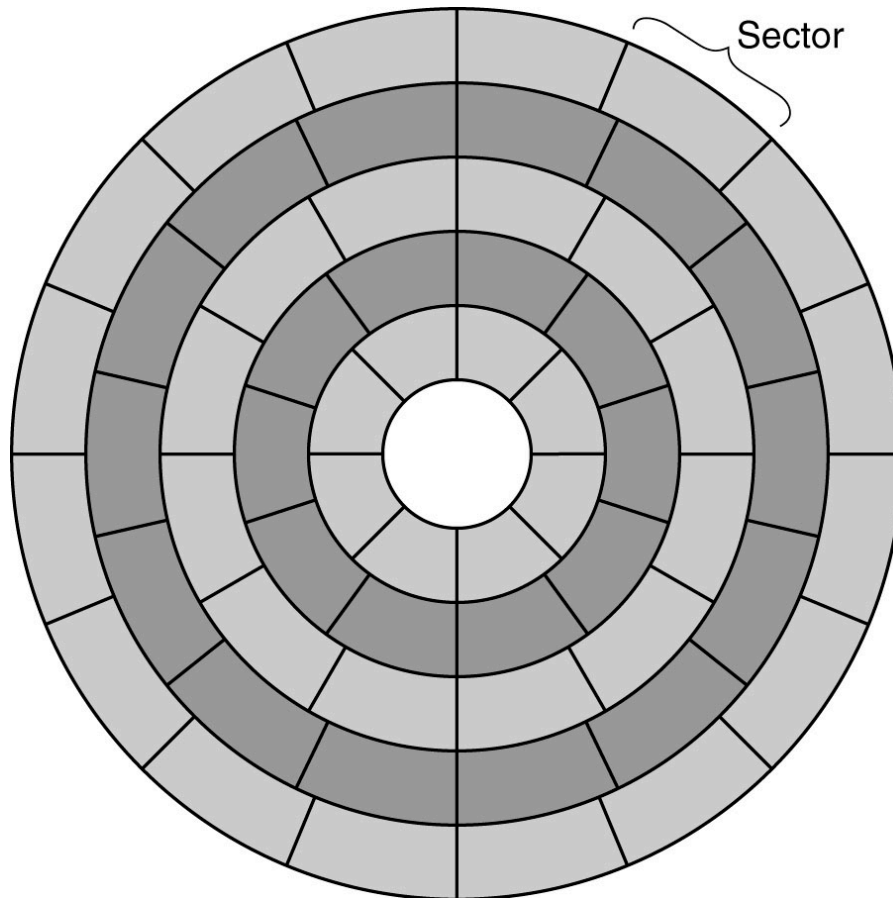


- *Cilindro*: insieme di tracce sulla stessa verticale
- *Tempo di seek*  $t_{seek}$ : spostamento delle testine sul cilindro desiderato, dipende in parte dalla distanza ( $\sim 5-10\text{ms}$ )
- *Tempo di latency*  $t_{lat}$ : spostamento sul settore desiderato ( $\sim 3-6\text{ms}$ )
- Tempo di accesso:

$$t_{acc} = t_{seek} + t_{lat}$$

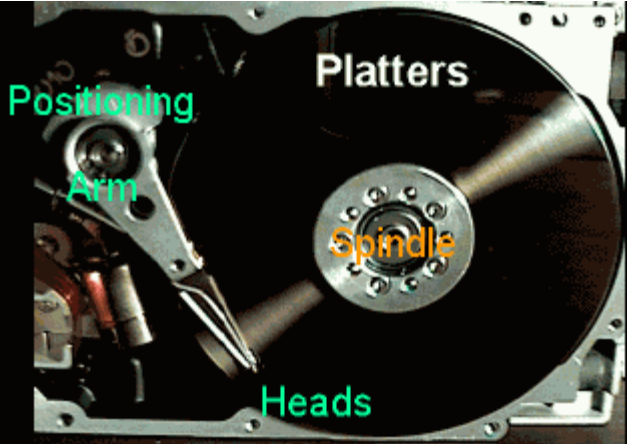
# Organizzazione dei dati su disco

Densità di registrazione variabile con il raggio della traccia ( $\sim 25$  Gbit/cm)



La gestione è fatta da **controllori di disco** (CPU specializzate)

# Un hard disk





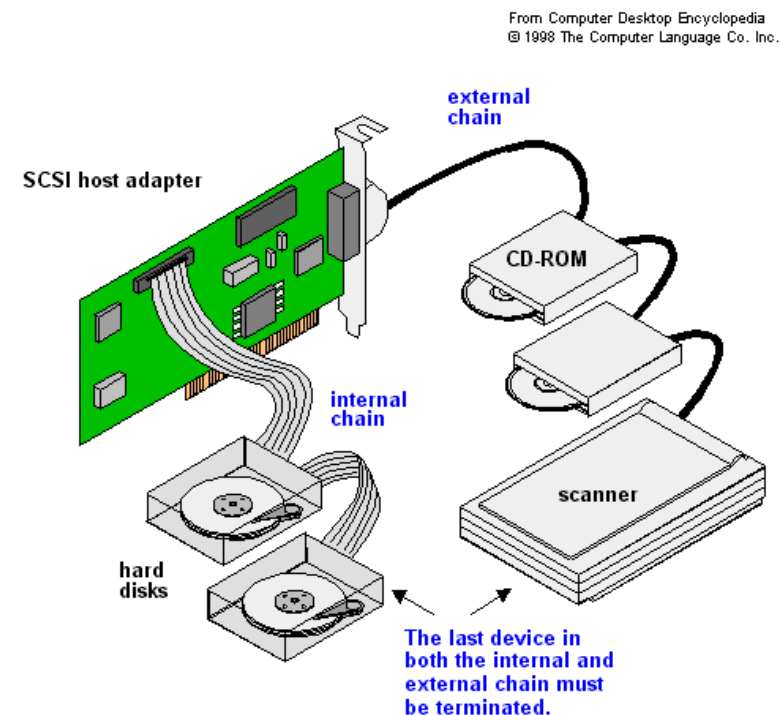
# Dischi IDE e EIDE

- IDE: standard nato con il PC XT IBM
- Limite di 16 testine, 63 settori e 1024 cilindri: in tutto 504 MB, transfer rate:  $\sim 4\text{MB/sec}$
- EIDE estende lo standard mediante lo schema LBA (Logical Block Addressing) che prevede  $2^{28}$  settori
  - Totale di  $2^{28} \times 2^9\text{B} = 128\text{GB}$
  - 2 controllori - 4 dischi per controllore
  - transfer rate più alta  $\sim 17\text{MB/sec}$
- ATA-3 (AT Attachment) a  $33\text{MB/sec}$
- ATAPI-5 (ATA PAcKet Interface) a  $66\text{MB/sec}$
- ATAPI-6 a  $100\text{MB/sec}$ 
  - LBA a 48 bit – Massimo:  $2^{48} \times 2^9\text{B} = 128\text{PB}$
- ATAPI-7 e successivi: basato su SATA (Serial ATA)
  - connettori a meno bit (da 80 a 7), tensioni più basse ( $0.5\text{V}$ ), velocità maggiori ( $> 500\text{MB/sec}$ )

# SCSI

- SCSI (Small Computer Interface), standard ANSI
- Controller e interfaccia più intelligente, migliori prestazioni
- Bus con connessione *daisy chain*
- Adatto a server. Usato nelle Workstation Unix
- Versione moderna: Serial attached SCSI (>1GB/sec)

Name	Data bits	Bus MHz	MB/sec
SCSI-1	8	5	5
Fast SCSI	8	10	10
Wide Fast SCSI	16	10	20
Ultra SCSI	8	20	20
Wide Ultra SCSI	16	20	40
Ultra2 SCSI	8	40	40
Wide Ultra2 SCSI	16	40	80
Wide Ultra3 SCSI	16	80	160
Wide Ultra4 SCSI	16	160	320
Wide Ultra5 SCSI	16	320	640



# Dischi RAID

Problema: miglioramento lento delle prestazioni dei dischi  
(1970:  $t_{\text{seek}}=50\text{ms}$ ; 2013:  $t_{\text{seek}}=5-10\text{ms}$ )

Soluzione: **RAID** (Redundant Array of Inexpensive Disks)

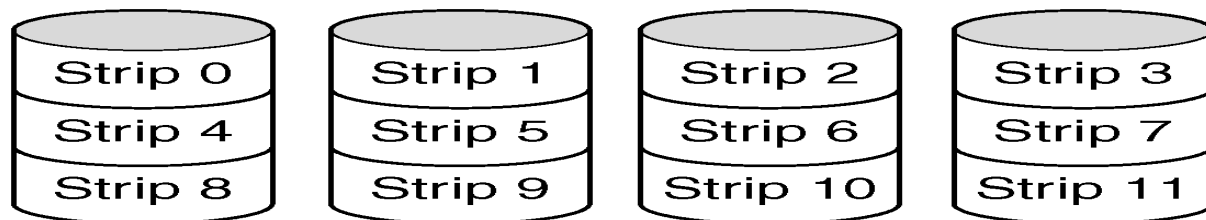
- Dividere i dati su più dischi
- Parallelizzare l'accesso
- Aumentare il data rate
- Introdurre una resistenza ai guasti

Contrapposti a **SLED** (Single Large Expensive Disk)

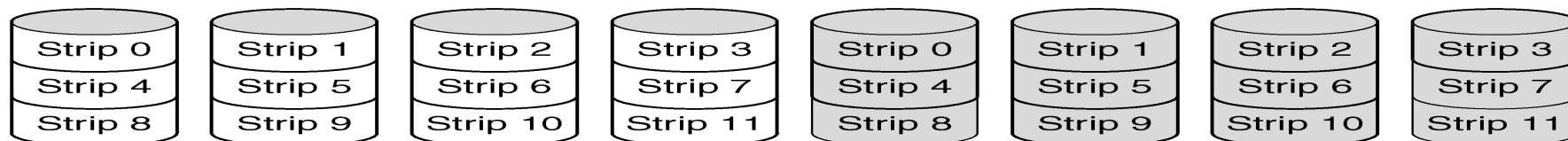


**Data Striping**: dati consecutivi nello stesso file vengono "affettati" e disposti su dischi diversi, dai quali possono essere letti (e scritti) in parallelo

# RAID Level 0 e Level 1

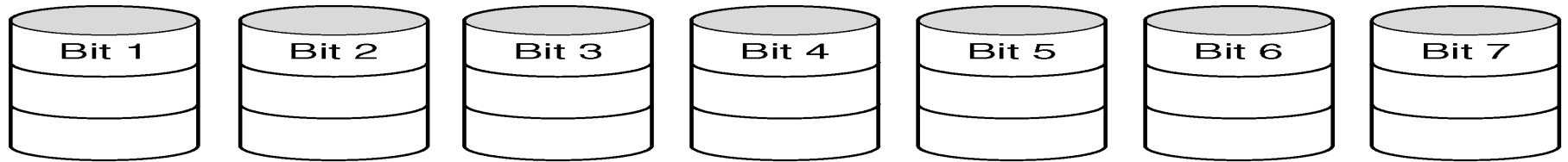


- Su n dischi si può guadagnare un fattore n sia in lettura che in scrittura
- Lo MTBF (*Mean Time Between Failures*) peggiora
- Non c'è ridondanza: non è un vero RAID



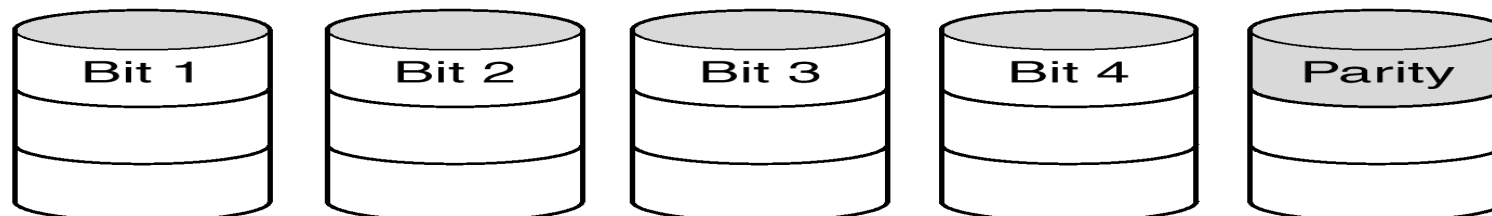
- Ciascun disco è duplicato: shadowing
- Ottime prestazioni soprattutto in lettura: molte possibilità di bilanciare carico
- Eccellente resistenza ai guasti
- Supportato anche da vari SO (Es. Windows)

# RAID Level 2



- *Striping* a livello di word o di byte
- Esempio: un *nibble* (mezzo byte) più 3 bit: codice di Hamming a 7 bit
- Registrazione ad 1 bit per ogni disco
- Rotazione dei dischi sincronizzata
- Resiste a guasti semplici
- Guadagna un fattore 4 in read e write
- Forte *overhead* (nell'esempio 75%)
- Ha senso con molti dischi:
  - $32 \text{ bit} + (6 + 1) \text{ parità} \Rightarrow 39 \text{ dischi}$
  - Overhead del 19%
  - Guadagna un fattore 32 in read e write

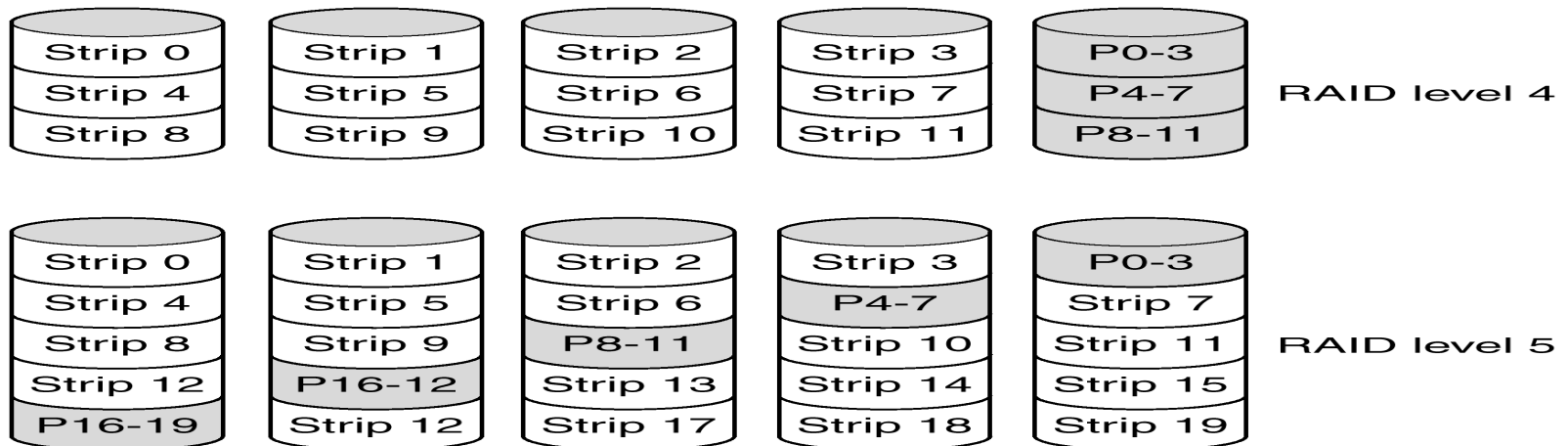
## RAID Level 3



- Versione semplificata di RAID 2
- Resiste a guasti semplici! Il bit di parità, *sapendo quale drive è rotto*, consente la correzione
- *Overhead* abbastanza contenuto

*RAID 2 e 3 offrono un'eccellente data rate ma permettono di gestire solo una operazione su disco per volta perché ciascuna operazione coinvolge tutti i dischi*

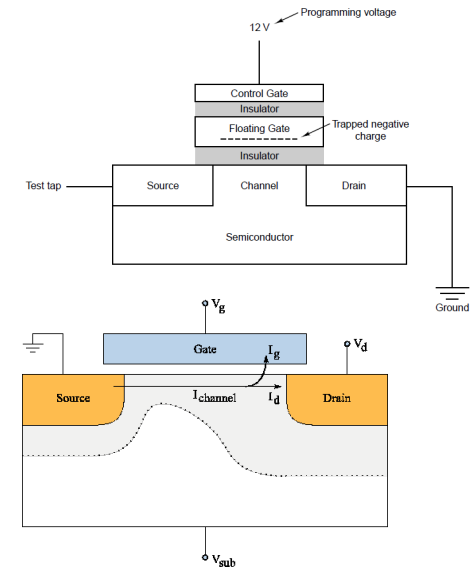
# RAID 4 e RAID 5



- *Striping* a livello di blocco: drive non sincronizzati
- RAID 4: la *strip* nell'ultimo disco contiene i bit di parità dell'insieme di bit omologhi di tutte le altre *strip*
- Resiste a guasti singoli (vedi RAID 3)
- Se una sola *strip* è scritta occorre leggere tutte le altre per calcolare la parità
- Il disco di parità è il collo di bottiglia
- RAID 5 distribuisce le *strip* di parità

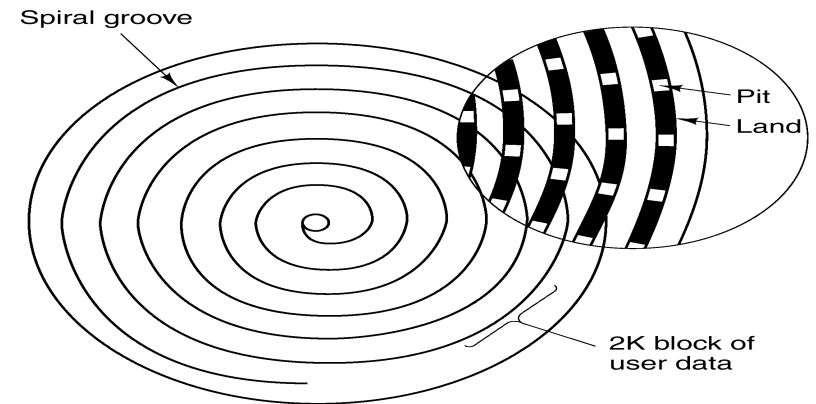
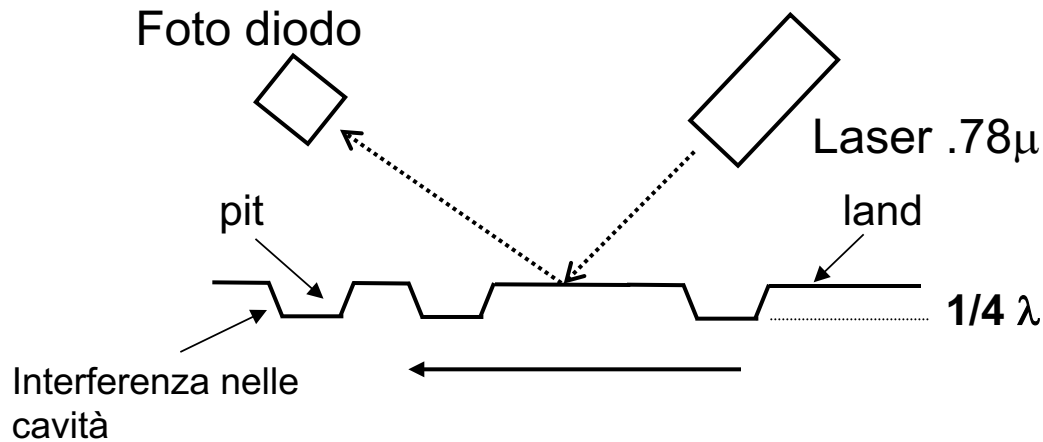
# Unità a stato solido (SSD)

- Basata sul fenomeno "Hot-carrier injection" dei transistor
- Celle di memoria flash a stato solido
- Montate sopra un normale transistor
- Applicando una tensione al CG:
  - Il FG si carica (no alimentazione)
  - Aumenta la tensione di commutazione
  - Test di commutazione a basso voltaggio
- Tempi di trasferimento:  $>200\text{MB/sec}$
- Adatto a dispositivi mobili
- Costi più alti:  $\sim 1\text{c/GB} \rightarrow \sim 1\text{€}/\text{GB}$
- Maggiore "failure rate":  $\sim 100.000$  Write
- Wear leveling: distribuzione uniforme delle scritture sulle celle dell'unità
- Aumento di capacità con celle multilivello



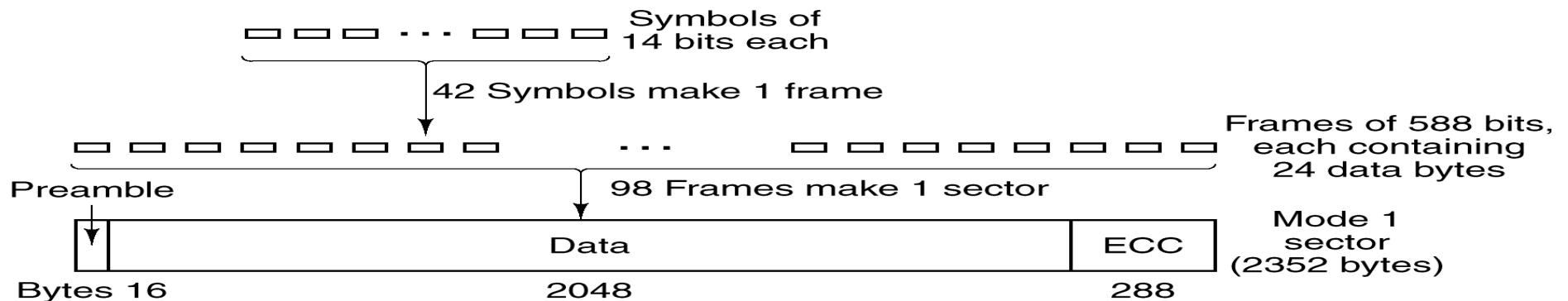


# Dischi ottici: CD-ROM



- Standard originariamente introdotto per i CD audio
- Registrazione basata su *pits* e *lands*
- L'interferenza nelle cavità diminuisce l'intensità luminosa
- Unica traccia a spirale ~22.000 giri a ~ 600/mm: lunghezza totale ~5.6Km
- Rotazione *variabile* 200-530 RPM per avere *flusso dati uniforme* (musica)

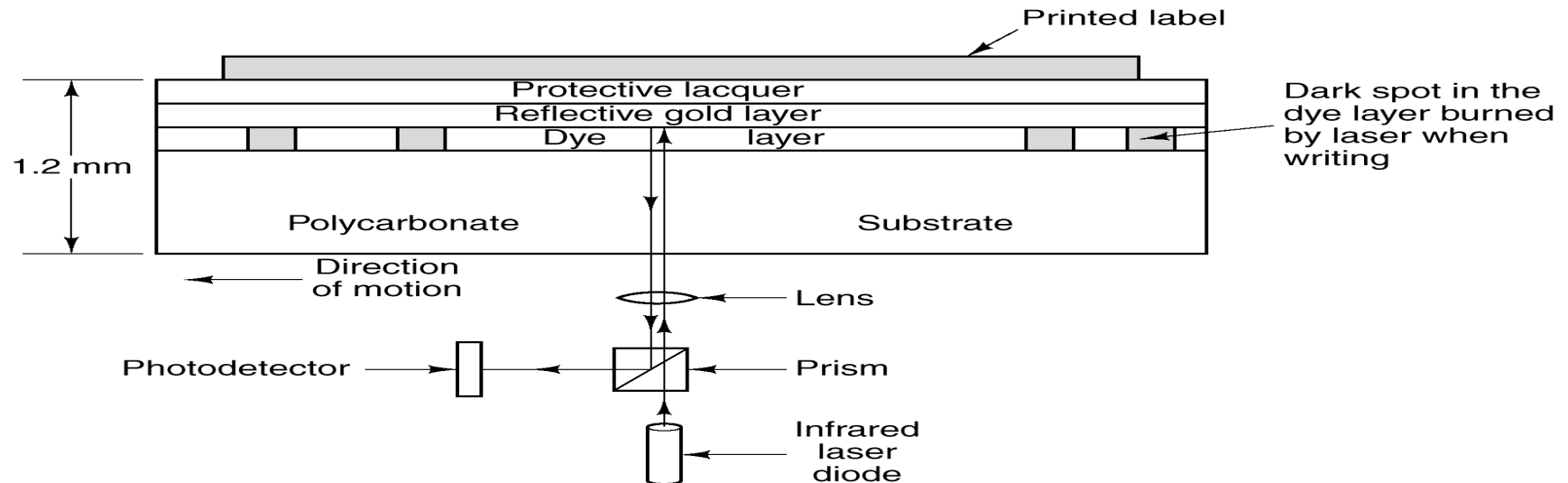
# CD-ROM: organizzazione dei dati



- Byte codificati in simboli di 14 bit
- Frame di 42 simboli di cui 24 utili
- Settori di 98 frame, con 2K byte utili più 16 di preambolo e 288 di ECC
- Contiene in tutto 650 MB utili
- Drive 1x: 75 settori/sec  $\approx$  150KB/sec
- Drive 32x:  $\sim$  5MB/sec

**N.B.** Tre livelli di correzione di errore (simbolo, frame e settore): la parte utile è solo del 28%

# CD scrivibili e riscrivibili



Laser a due potenze (scrivibili: CD-R):

- alta (scrive): 'brucia' delle piccole aree nello strato colorato
- bassa (legge): come nei CD-ROM

(Solco preinciso per guidare il laser)

Laser a tre potenze (riscrivibili: CD-RW):

(supporto a due stati: amorfo e cristallino)

- alta (scrive): cristallino → amorfo
- media (cancella): amorfo → cristallino
- bassa (legge): come nei CD-ROM

# Standard per i CD ROM

**Red Book** (1980) : standard ISO per il CD audio (Philips e Sony): simboli e frame

**Yellow Book** (1984) : standard ISO per i CD-ROM dati: settori

**Green Book**: standard CD multimediali

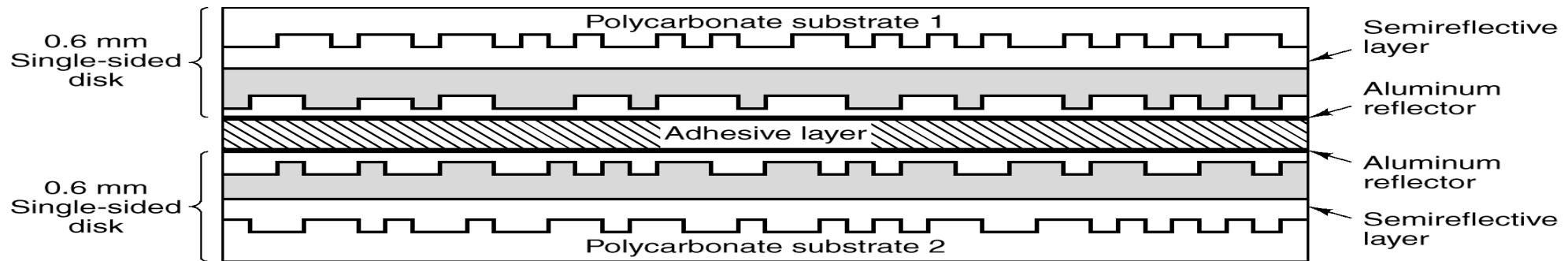
**High Sierra**: Standard ISO 9660 per il file system:

- Level 1: compatibile con tutti i SO
- Level 2: Unix (UID, link simbolici)
- Level 3: file non contigui

**Orange Book**: standard CD scrivibili

- Formato CD-R
- Formato CD-ROM XA (scrittura incrementale)
- CD-ROM multisezione
- Aggiornamento della directory
  - VTOC: Volume Table of Contents

# DVD: Digital Versatile Disk



Aumento della capacità:

- Laser  $\lambda=0.65\mu$  (era  $0,78\mu$ )
- Pit= $0.4\mu$  (erano  $0.8\mu$ )
- Spirale più stretta  $0.74\mu$  (erano  $1.6\mu$ )
- Capacità: 4.7 GB
- Data rate (1x): 1.4MB/sec

Basta per 133 minuti di film a  $720\times 480$ , compresso con MPEG-2

Ulteriore aumento di capacità:

- Singola faccia, layer doppio: 8.5 GB
- Doppia faccia, layer singolo: 9.4 GB
- Doppia faccia, layer doppio: 17 GB

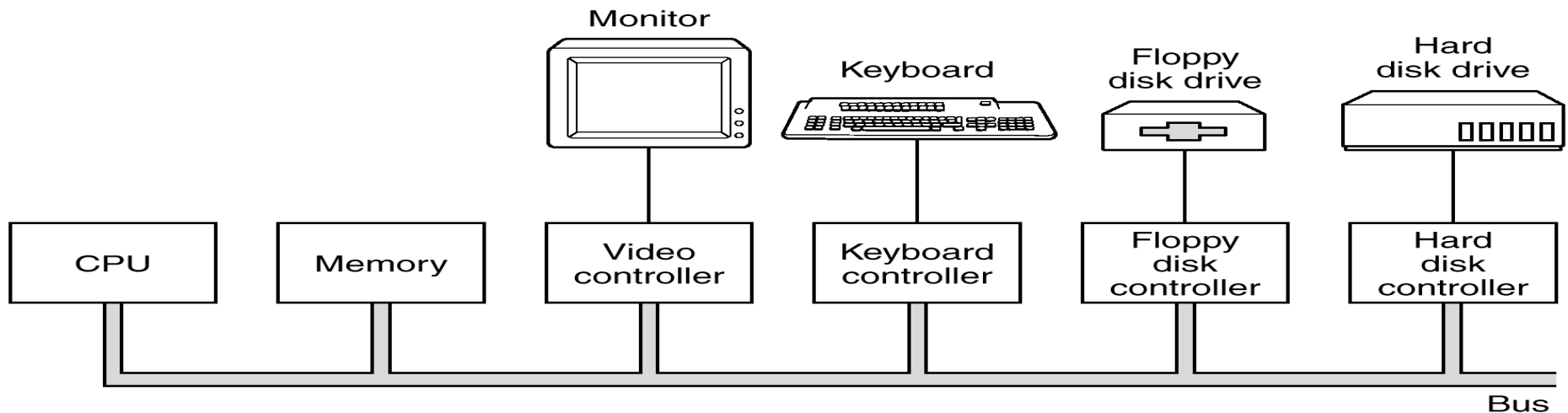
# Blu-Ray

## Successore del DVD

- Laser (blu)  $\lambda=0.405\mu$
- Singola faccia: 25 GB
- Doppia faccia: 50 GB
- Data rate: 4,5MB

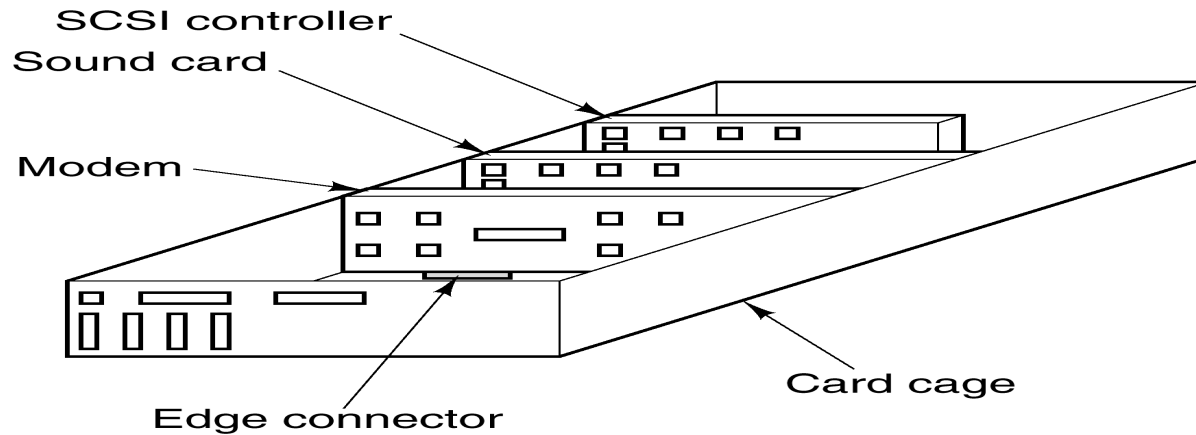


# Dispositivi di I/O



- I dispositivi di I/O sono connessi al bus tramite *controller*
- I controller gestiscono autonomamente i trasferimenti da e per la memoria: DMA (Direct Memory Access)
- Possono comunicare con la CPU tramite le *interruzioni*
- Il bus è condiviso da CPU e controller, e gli accessi sono regolati da un arbitro

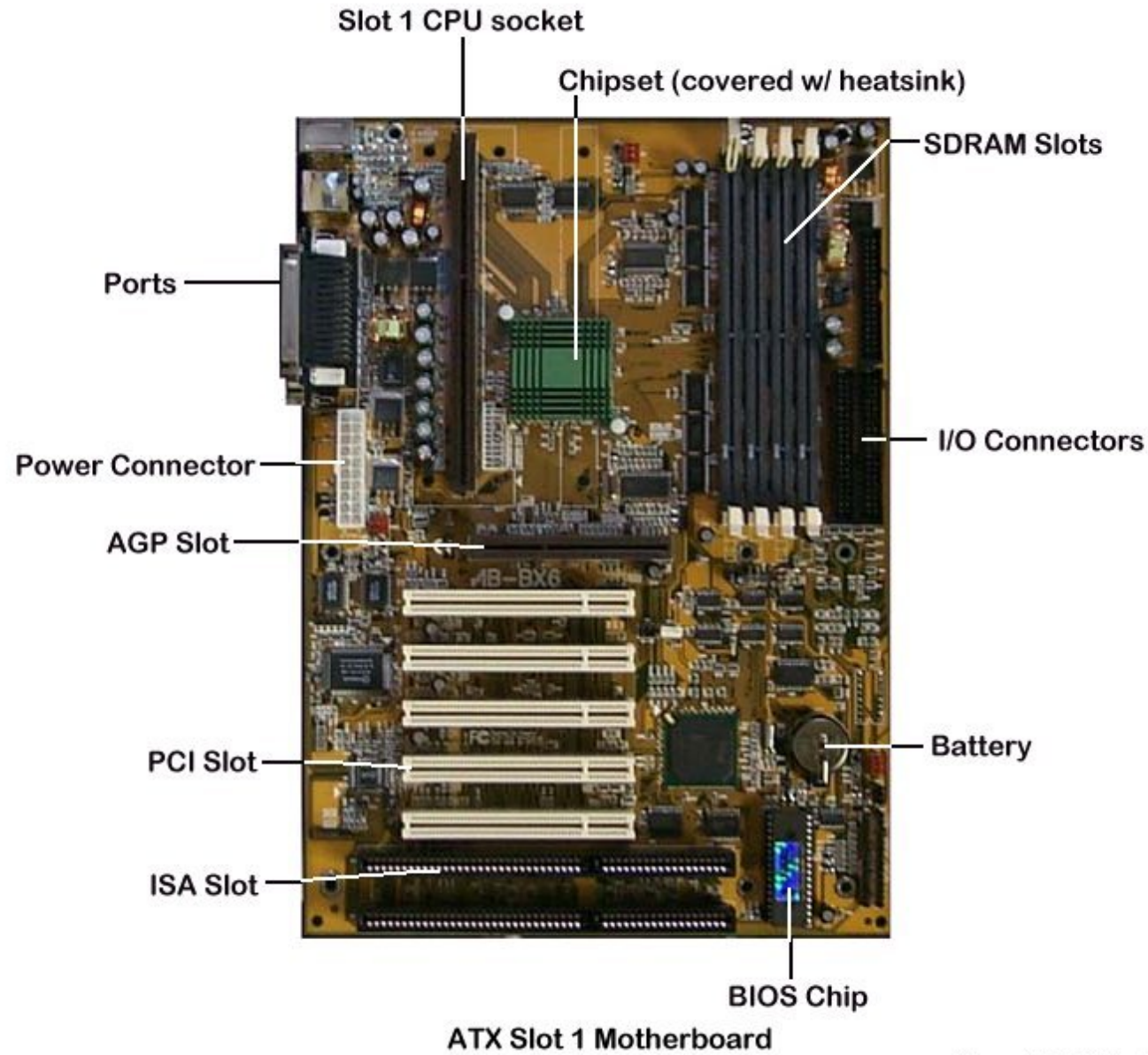
# Struttura fisica del PC



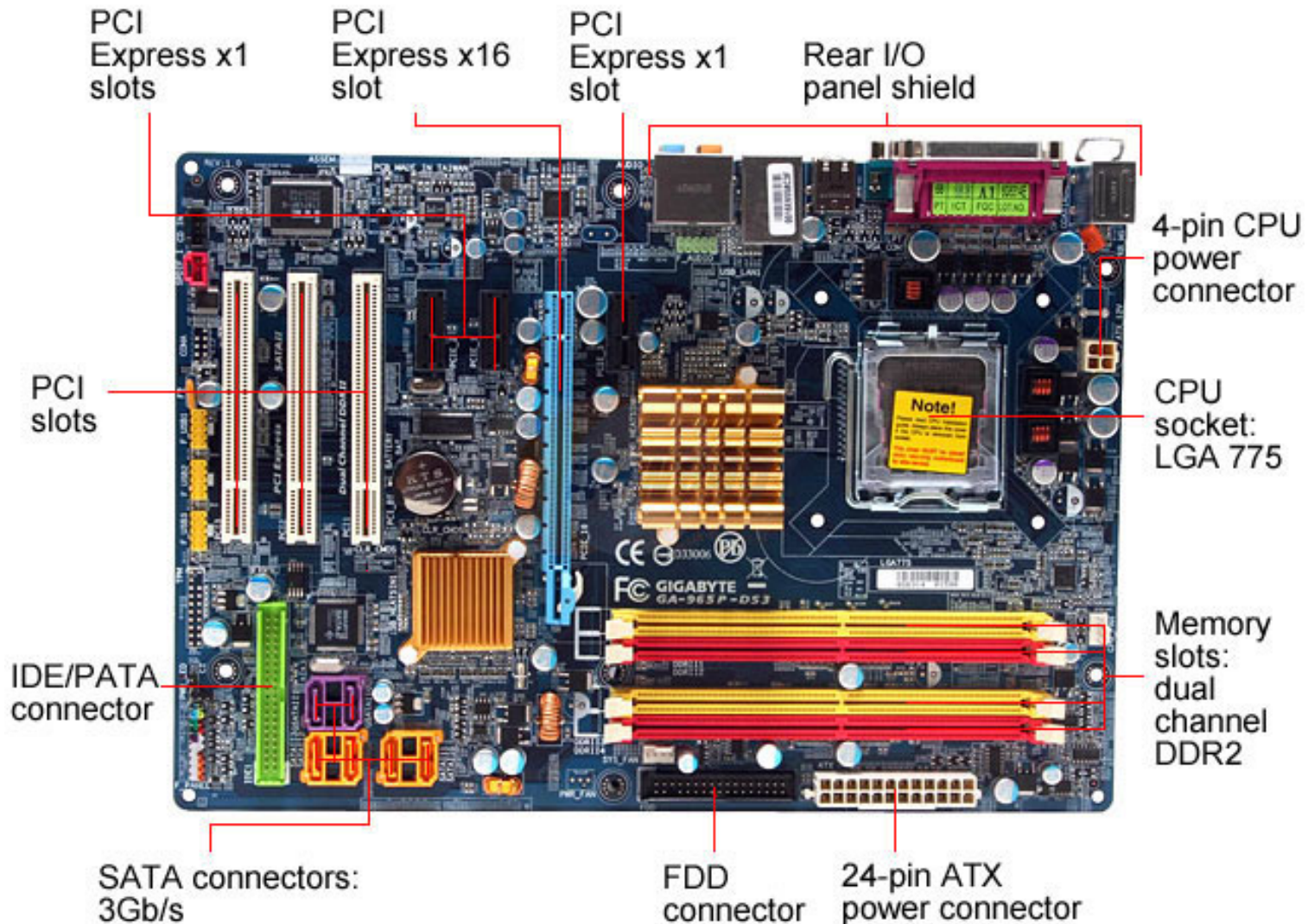
- La base della struttura è costituita dalla *Scheda Madre (Mother Board)*
- Sulla scheda madre sono la CPU, il *Chipset*, il bus e vari connettori per la memoria e i dispositivi di I/O
- Il bus è costituito da una serie di piste sul circuito stampato
- Spesso sono presenti più bus, secondo diversi standard
- Le schede di I/O vengono inserite nei connettori



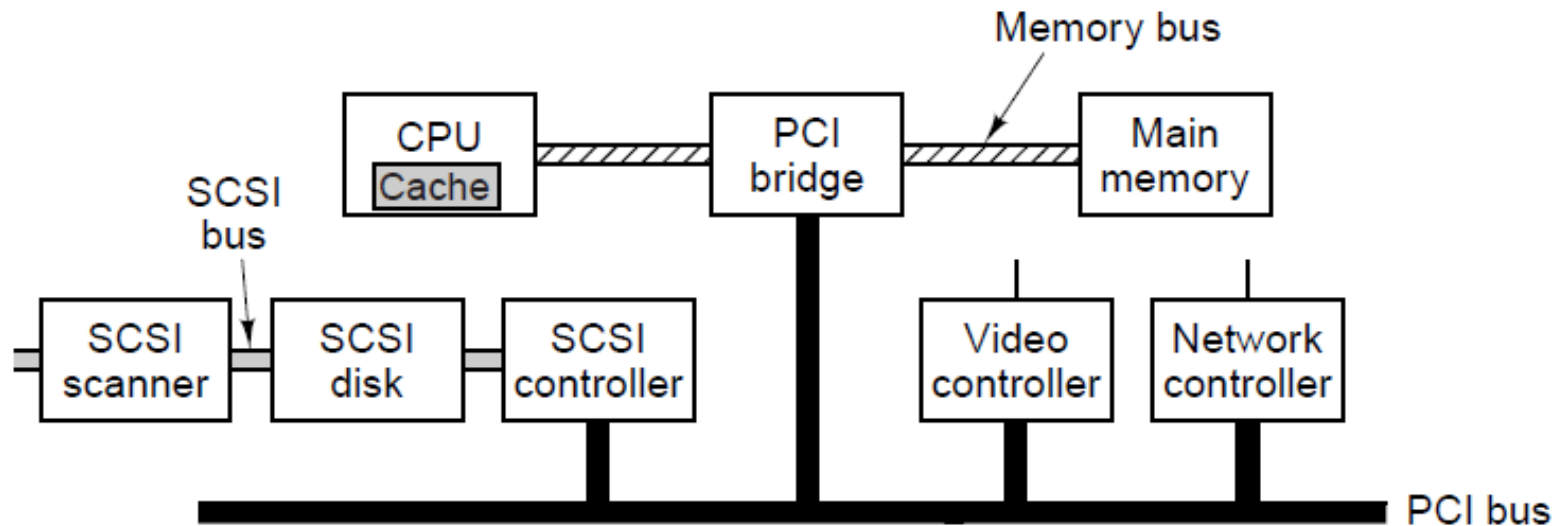
# Una schema madre



# Scheda madre "moderna"

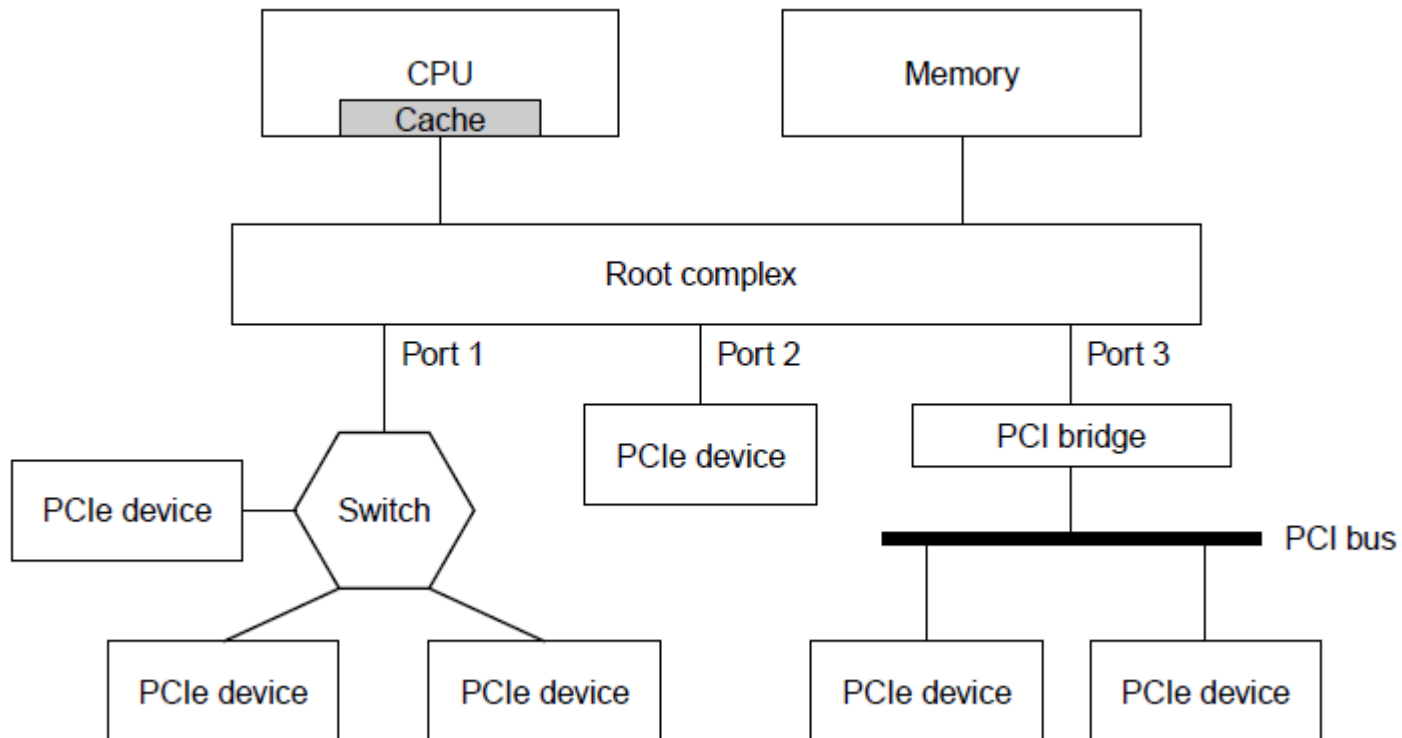


# Bus PCI



- ISA (*Industry Standard Architecture*) nasce col PC AT IBM (~1982)
- EISA (*Extended ISA*) estensione di ISA
- PCI (*Peripheral Component Interconnect*) di Intel (~1990)
  - ha più linee e maggiore frequenza (66MHz)
  - è di pubblico dominio
  - fino a 528 MB/sec

# PCIe



- Nuovo standard per periferiche veloci
- Basato su trasmissione seriale, PPP
- Risolve il problema del bus skew
- Fino a 16 GB/sec

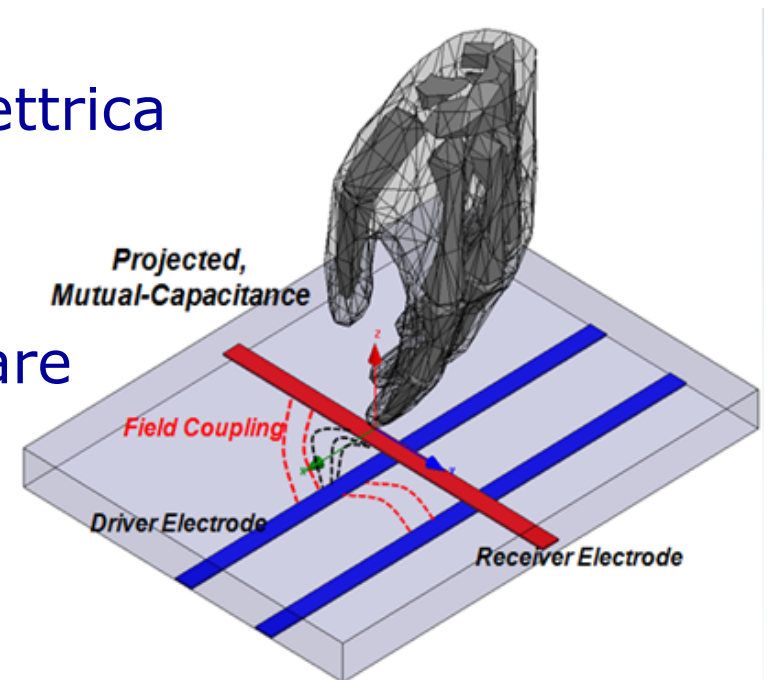
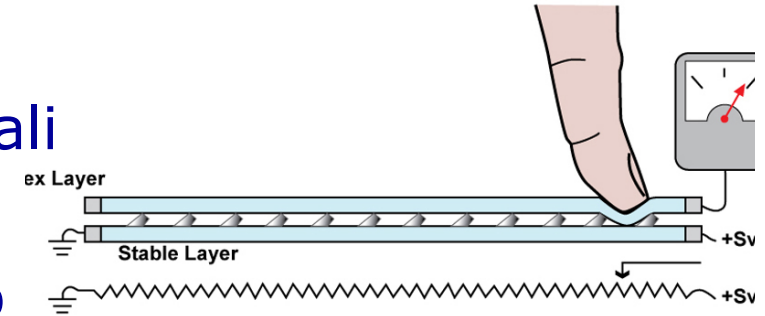
# Tastiere

- Codice del carattere inviato ogni volta che si pigia o rilascia un tasto
- Genera una interruzione della CPU
- Combinazioni di tasti gestite SW
- Mapping dei tasti (tastiere nazionali) gestito dal software

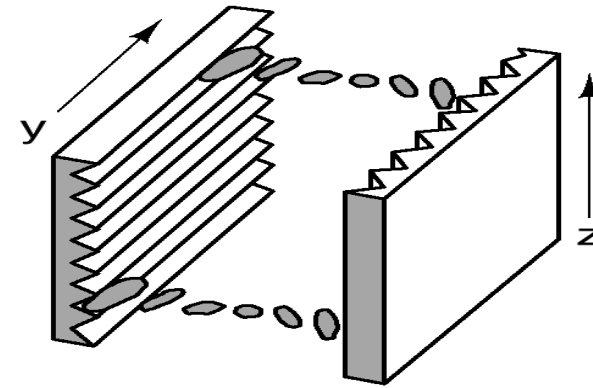
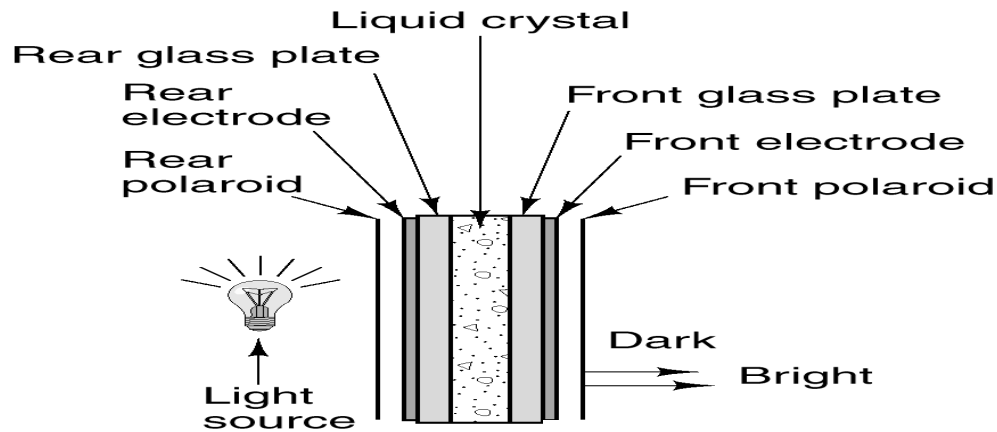


# Touch screen

- Tecnologia introdotta nel 1965
- Resistive touch screen
  - 1 strato flessibile con fili orizzontali
  - 1 strato rigido con fili verticali
  - La pressione produce un contatto
- Capacitive touch screen
  - Il contatto cambia la capacità elettrica
  - Dimensione dei fili:  $50\mu\text{m}$
  - Piastra di vetro protettiva
  - Movimenti interpretati dal software



# Display Flat Panel



- Basati sulla conduzione di luce polarizzata da parte di *cristalli liquidi* (LCD)
- Allineamento dei cristalli influenzato da campi elettrici
- Campi generati da elettrodi trasparenti
- Problemi: basso contrasto, angolo visuale

## Display a Matrice Attiva (TFT)

- Piccoli interruttori per ogni pixel controllati da una *matrice di selezione*
- Molto migliori ma costosi

## Display a colori

- Stessi principi, ma più strati e filtri

# Video RAM

Memoria dedicata alla gestione del video

Situata nel controller video

Dimensione che dipende dalla risoluzione

- L'immagine è costituita da una matrice di punti (pixel)
- Dimensione  $> P \times N \times K$  (es.  $(1920 \times 1080) \times 3 \times 1 = 6.2 \text{ MB}$ )
  - $P$  = Numero di pixel
  - $N$  = Numero di byte per pixel
  - $K$  = Numero di matrici (frame)
- $N$  è piccolo (8 bit) e indirizza una tabella (Color palette) che contiene il valore di 3 byte corrispondente al colore

Richiedono molta banda

- $6.2 \text{ MB} \times 25 \text{ frame/sec} = 155 \text{ MB/sec}$

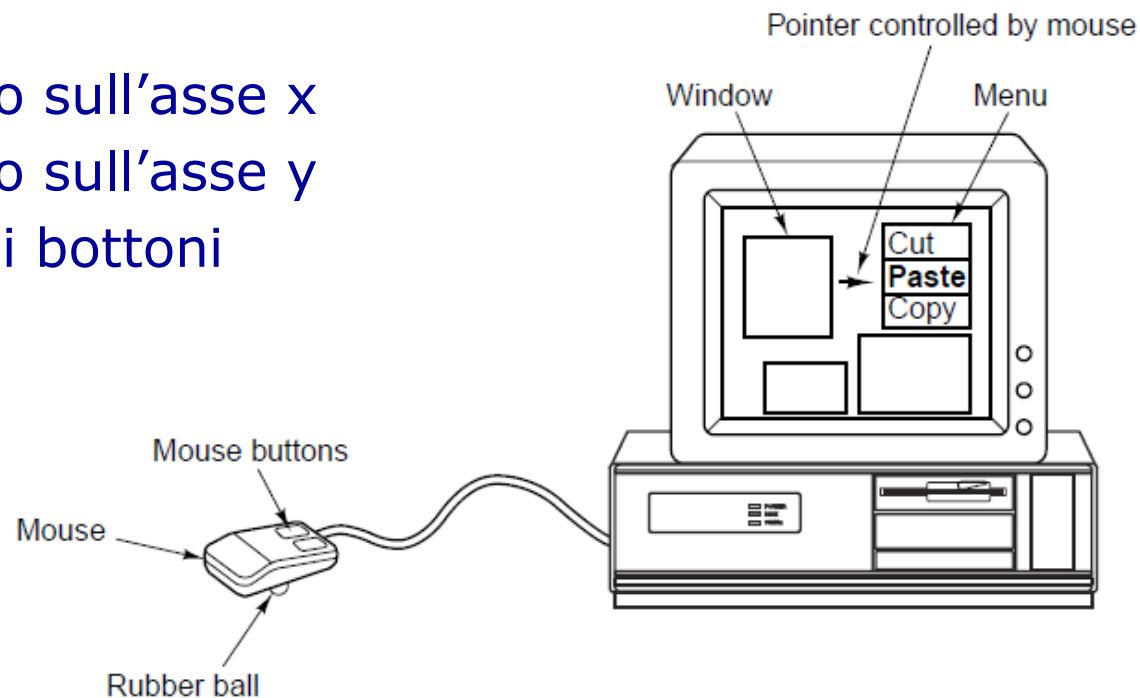
Si usano Bus speciali o moderni

- AGP: 32 bit, 66 Mhz, 252 MB/sec
- PCIe: 16 GB/sec



# Mouse

- Tecnologie
  - *Meccanica*: movimento di rotelle dotate di potenziometri
  - *Opto-meccanica*: come quelli meccanici ma rotazione rilevata da infrarossi
  - *Ottica*: dotate di LED (Light Emitting Diode) e fotorilevatore
- 3 byte inviati ogni volta che il mouse fa un certo spostamento minimo:
  - 1 - Spostamento sull'asse x
  - 2 - Spostamento sull'asse y
  - 3 - Posizione dei bottoni

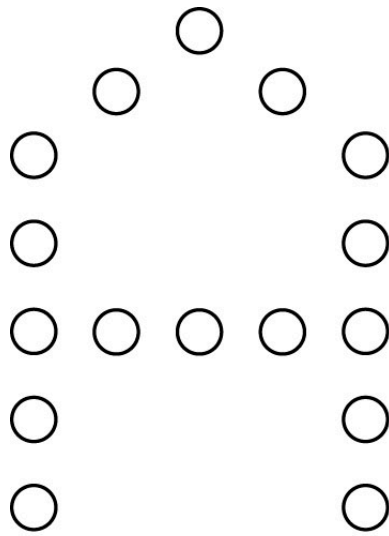


# Stampanti

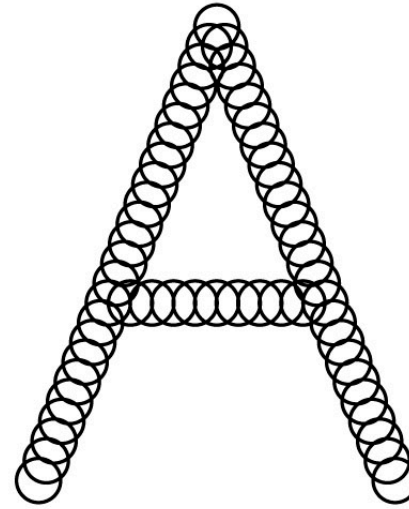
Vari tipi (e vari costi)

- A matrici
- A getto di inchiostro (inkjet)
- Laser

Composizione dell'immagine (e caratteri) per "punti" (dot)

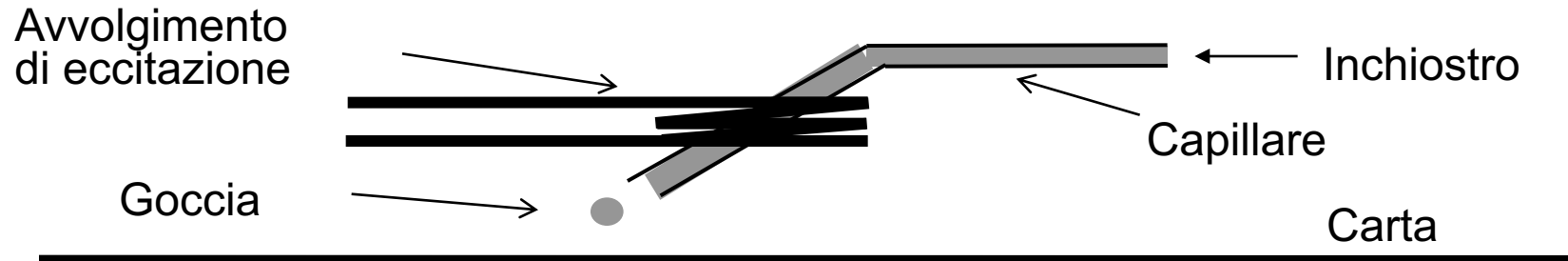


(a)



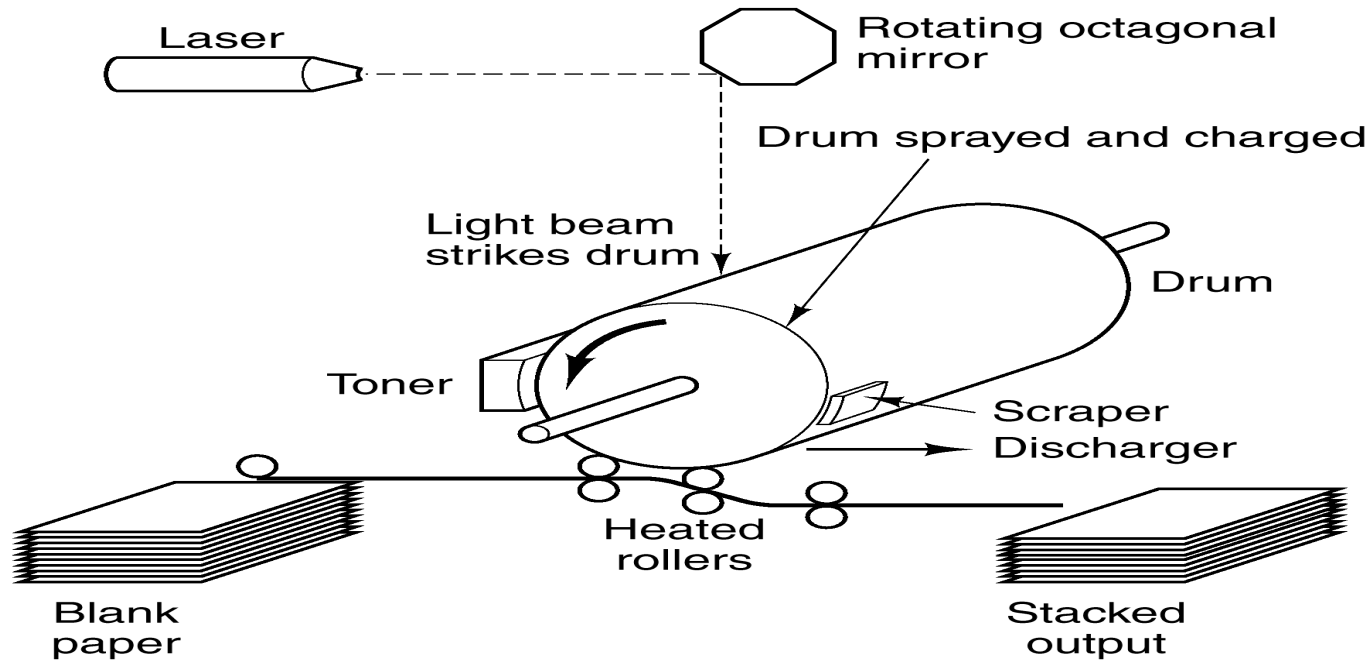
(b)

# Stampanti Inkjet



- Stampanti grafiche: 1200-4800 dot/inch
  - Termiche (bubblejet): espansione e contrazione dell'inchiostro per impulsi elettrici nell'avvolgimento
  - Piezoelettriche: contrazione meccanica del capillare
- Poco costose e silenziose, ma lente
- **Problemi:**
  - Trascinamento della carta
  - Intasamento dei capillari
  - Costo delle cartucce
  - Generazione dell'immagine, da parte del computer

# Stampanti Laser



Eccitazione elettrostatica del tamburo di selenio con pennello laser  
Adesione del *toner* (polvere plastica)  
Trasferimento del toner sulla carta a caldo e sua fusione (fissaggio)

## Vantaggi

- Silenziose, veloci e precise
- Maggiori costi fissi, minori marginali

# Gestione delle Stampanti Laser

- La generazione dell'immagine viene fatta dalla stessa stampante
- Stampante dotata di microprocessore e parecchi MB di ROM e RAM
- Set di caratteri contenuti nella ROM
- Scalatura delle fonti
- Il computer invia una descrizione dell'immagine in uno speciale linguaggio
- Postscript: linguaggio proprietario di Adobe per la descrizione di immagini
- Un file Postscript è un programma che descrive un documento:
  - Testo, con specifica di fonte e corpo
  - Immagine delle fonti non standard
  - Disegni (vettoriali e bitmap)
  - Formato della carta, ecc.

# Stampanti a Colori

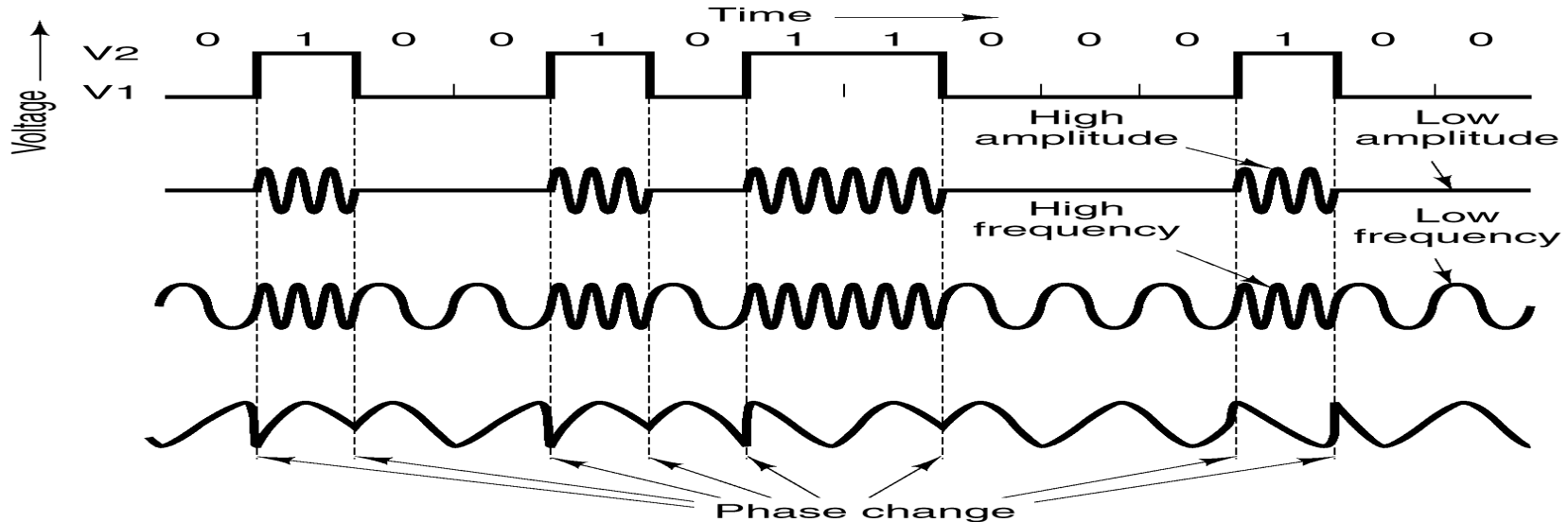
**CYMK** (**C**yan, **Y**ellow, **M**agenta, **B**lack): codice di stampa a 4 colori (per immagini a luce riflessa)

**RGB** (**R**ed, **G**reen, **B**lue): codice a 3 colori monitor (per immagini a luce trasmessa) – necessaria conversione

**Gamut**: set di colori gestiti

- Inkjet: Come B/N, con più testine di stampa
- Solid ink: usano inchiostri solidi
- Wax: usano cera al posto dell'inchiostro
- Laser: usano 4 toner di colori diversi e richiedono parecchia memoria
- Sublimazione: fondono inchiostri solidi che sublimano quando si depositano sul foglio
- Termiche: dotate di aghi sottili che, quando riscaldati, producono punti su carte speciali

# Modem



Informazione binaria trasmessa su linee analogiche modulando una portante

Modulazione di ampiezza, frequenza e fase

- Bit rate: frequenza di invio dei bit
- Baud rate: frequenza con cui varia il segnale

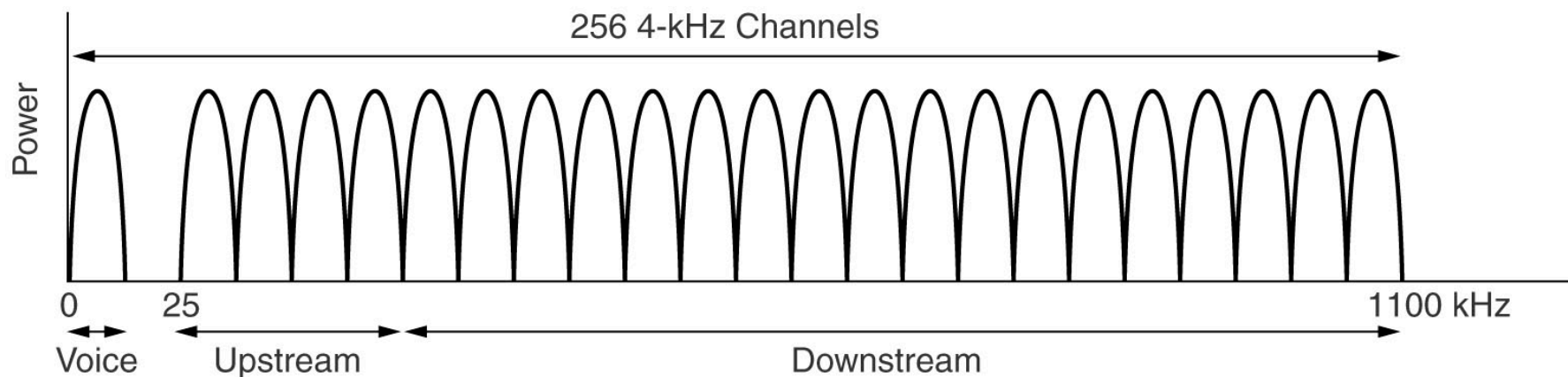
Bit rate tipiche: 28.800~57.600 bits/sec

Protocolli V.21, V.32, V.34, V.90 (56Kbps)

Modem programmabili tramite linguaggio standard (AT)

# ADSL

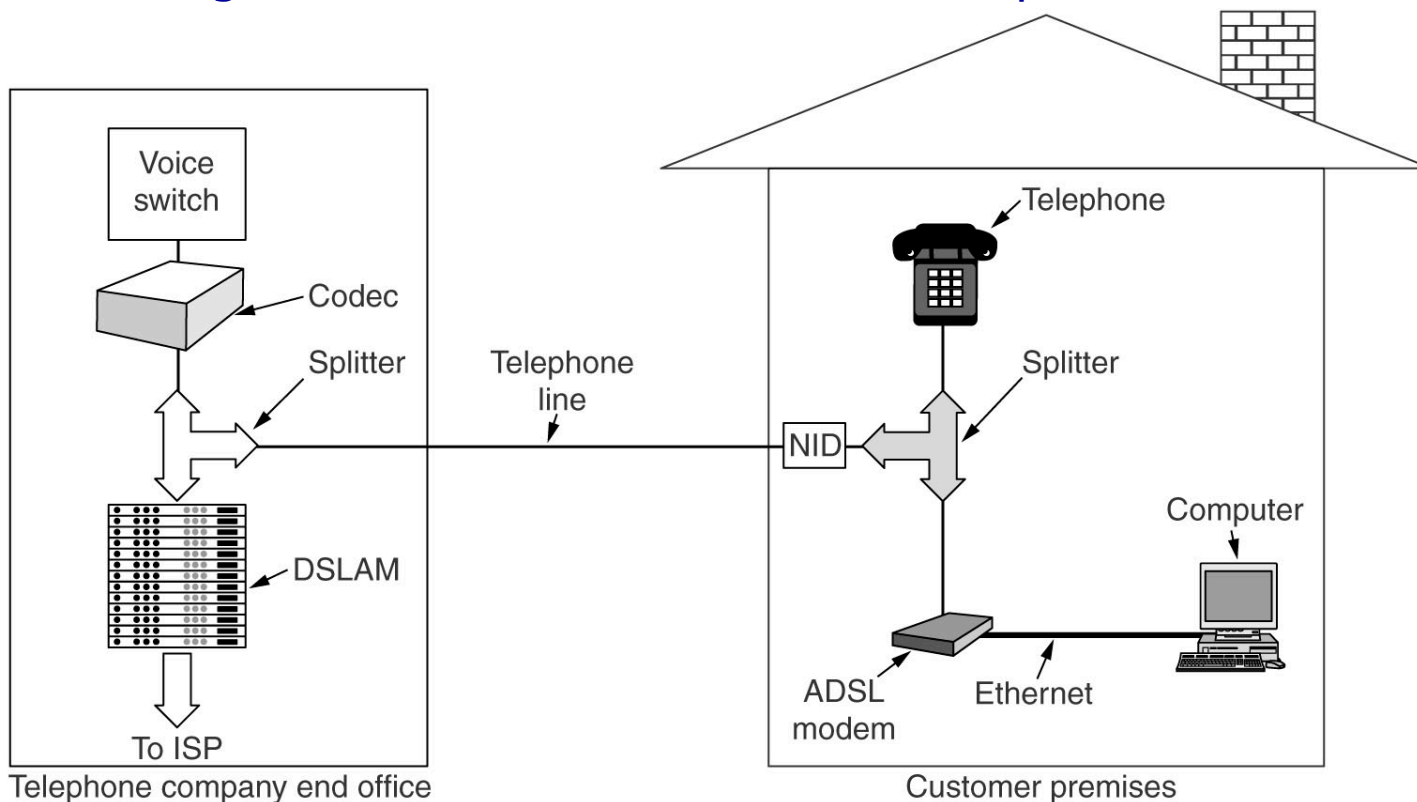
- Tecnica appartenente alla famiglia Digital Subscriber Line (xDSL): Asymmetric DSL
- Cavi telefonici tradizionali: filtro a 3000 Hz (banda fonica)
- Trasmissione ADSL: frequenza a 1.1Mhz
- Suddivisione della banda in 256 canali indipendenti da 4kHz
  - Canale 0: POTS (Plain Old Telephone Service)
  - Canali 1-5: non usati
  - Canali 6-7: per il controllo downstream e upstream
  - Canali 8-255: utilizzati per i dati
- Solo alcuni usati di cui l'80-90% in downstream (asymmetry)





# Caratteristiche ADSL

- Possibile velocità downstream:
  - $224 \text{ can.} \times 15 \text{ bit/ baud} \times 4000 \text{ baud} = 13.44 \text{ Mbps}$
  - Tipicamente 4 Mbps
- Possibile configurazione:
  - NID: Network Interface Device
  - Modem ADSL: 250 de/modulazioni in parallelo
  - DSLAM: Digital Subscriber Line Access Multiplexer



# Codice ASCII (Hex 0-1F)

Hex	Name	Meaning	Hex	Name	Meaning
0	NUL	Null	10	DLE	Data Link Escape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative Acknowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous idle
7	BEL	BELI	17	ETB	End of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End of Medium
A	LF	Line Feed	1A	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

# Codice ASCII (Hex 20-7F)

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
20	(Space)	30	0	40	@	50	P	60	'	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

# Codice UNICODE

- Codice ASCII a 7 bit, poi esteso a 8
- *Escape sequences*: per caratteri *speciali*
- Diversi codici nazionali *code pages*
- Impossibile usarli simultaneamente
- Codice UNICODE a 16 bit, nuova proposta di standard:
  - 65.536 *code points*
  - Semplifica la scrittura del software
  - 336 code points: alfabeti latini
  - 112 accenti e simboli diacritici
  - Greco, cirillico, ebraico, ecc.
  - 21.000 ideogrammi cinesi.....
  - Un consorzio assegna quello che resta
- UTF-8: codice a lunghezza variabile basato su unicode
  - 0xxxxxxx: ASCII
  - Altri prefissi che iniziano per 1: codifiche più lunghe

## Esercizio 3

Si consideri una CPU con pipeline a 6 stadi che lavora a una frequenza di 400 Mhz e in cui ogni stadio viene eseguito in un ciclo di clock; indicare se le seguenti affermazioni sono vere o false.

- A regime e in condizioni ideali la CPU completa un'istruzione ogni 2.5 nsec.
- Una istruzione richiede 10 nsec per essere eseguita.
- L'ampiezza di banda della CPU è di 500 MIPS.
- La latenza della CPU è di 15 nsec.
- In linea di principio, se la frequenza del clock aumenta a 800 Mhz si raddoppia l'ampiezza di banda.
- In linea di principio, se la frequenza del clock scende a 200 Mhz si raddoppia la latenza.
- Il tempo di esecuzione di un programma di 3 istruzioni è di 20 nsec.
- In linea di principio, togliendo uno stadio si aumenta la latenza e si diminuisce l'ampiezza di banda.

## Esercizio 3

Si consideri una CPU con pipeline a 6 stadi che lavora a una frequenza di 400 Mhz e in cui ogni stadio viene eseguito in un ciclo di clock; indicare se le seguenti affermazioni sono vere o false.

- A regime **VERO** condizioni ideali la CPU completa un'istruzione ogni 2.5 nsec.
- Una istruzione richiede 10 ns **FALSO** essere eseguita.
- L'ampiezza **FALSO** della CPU è di 500 MIPS.
- La latenza della CPU è **VERO** nsec.
- In linea di principio, se la frequenza del clock aumenta a 800 Mhz si raddoppia l'ampiezza **VERO** di banda.
- In linea di principio, se la frequenza del clock scende a 200 Mhz si raddoppia la late **VERO**.
- Il tempo di esecuzione di un programma di 3 istruzioni è di 20 nsec. **VERO**
- In linea di principio, raddoppiando uno stadio si aumenta la latenza e si diminuisce l'am **FALSO** di banda.

## Esercizio 4

Illustrare la composizione e funzionamento di un'unità RAID di 200 GB (spazio utilizzabile di memoria fisica) e con blocchi (strip) di 512 KB, con riferimento:

- (A) ad una configurazione di livello 1 con 4 dischi,
- (B) ad una configurazione di livello 2,
- (C) ad una configurazione di livello 4 con 5 dischi e
- (D) ad una configurazione di livello 5 con 3 dischi.

Indicare in entrambi i casi la dimensione effettiva di memoria fisica necessaria per la realizzazione (in numero di byte).

# Soluzione esercizio 4

