

Corso di Sistemi Informativi

Ingegneria Gestionale V.O. - V anno

<http://www.dei.unipd.it/~capri/SI/index.htm>

Andrea Pietracaprina

Dipartimento di Ingegneria dell'Informazione

Università di Padova

E-mail: andrea.pietracaprina@unipd.it

Programma del Corso

1. Nozioni base su sistemi informatici [5%]
2. Basi di Dati [50%]
3. Reti di Calcolatori e Internet [35%]
4. Data Warehousing e Data Mining [10%]

Materiale di Riferimento

- **Materiale On-Line** (scaricabile dal sito)
- **P.Atzeni, S.Ceri, S.Paraboschi, R.Torlone.** *Basi di Dati*, 2a edizione, McGraw Hill, 1999.

Testi di Consultazione

R.A. Elmasri, S.B. Navathe: Fundamentals of Data Base Systems, Benjamin-Cummings, 1994 (3rd edition, 2000)

L.L. Peterson, B.S. Davie: Computer Networks -- A Systems Approach, 2nd edition, Morgan Kaufmann, 1999

D.E. Comer: Internet e Reti di Calcolatori, Addison Wesley, 2000

D.A. Patterson , J.L. Hennessy: Computer Organization & Design: The Hardware/Software Interface, Morgan Kaufmann, 1994

A.S. Tanenbaum: Modern Operating Systems, Prentice Hall, 1992

G. Bracchi, C. Francalanci, G. Motta : Sistemi Informativi e Aziende in Rete, McGraw Hill, 2001

K.C. Laudon, J.P. Laudon: Management Information Systems, (5th edition), Prentice Hall, 1998.

Modalità d'esame

- **Progetto** (facoltativo)

- Consegna entro scritto I appello (per chi sostiene tale appello) ed entro scritto II appello per tutti gli altri
- Max 2 punti di incremento sul voto finale

- **Scritto** (obbligatorio)

- **Orale** (facoltativo) diventa obbligatorio se il voto dello scritto è 16 o 17, o a discrezione del docente

N.B. Scritto e orale **devono** essere sostenuti nello stesso appello

Orario delle lezioni:

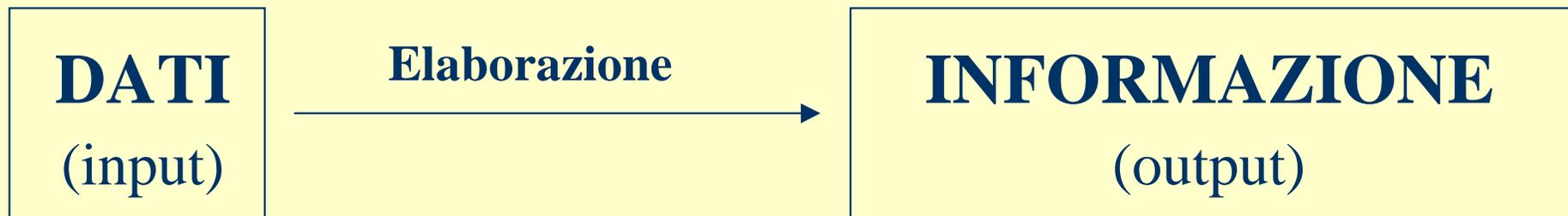
- **Lunedì: 12:00-13:30**
- **Martedì: 14:30-16:00**
- **Mercoledì: 12:00-13:30 e 16:00-17:30**

Orario di Ricevimento: Martedì: 16:00-18:00



**CONSULTARE
FREQUENTEMENTE IL
SITO WEB DEL CORSO**

Sistema Informativo: Insieme di componenti in relazione tra loro che *raccoglie, estrae, elabora, memorizza e distribuisce* informazione che serve come supporto alle funzioni operative, ai processi decisionali e al controllo di una organizzazione.



Sistema Informativo basato su computer: Sistema informativo che utilizza (principalmente) tecnologie informatiche e di telecomunicazioni, e si fonda su metodi rigorosi di definizione ed elaborazione di dati.

ICT = Information & Communication Technology

Supercomputer

- Molte CPU (custom o microprocessori): [2÷10⁶]
- Costo elevato
- Usato per (singole) applicazioni che richiedono elevata capacità di calcolo e la cooperazione di numerosi processori. Si tratta solitamente di applicazioni scientifiche (meteorologia, fisica, biologia) ma può anche essere usato in applicazioni commerciali (datamining)

Es. Quad-processor Pentium Pro, IBM SP-3/4, CRAY T3E, Earth Simulator, Blue Gene

Tipologie di Computer

Earth Simulator



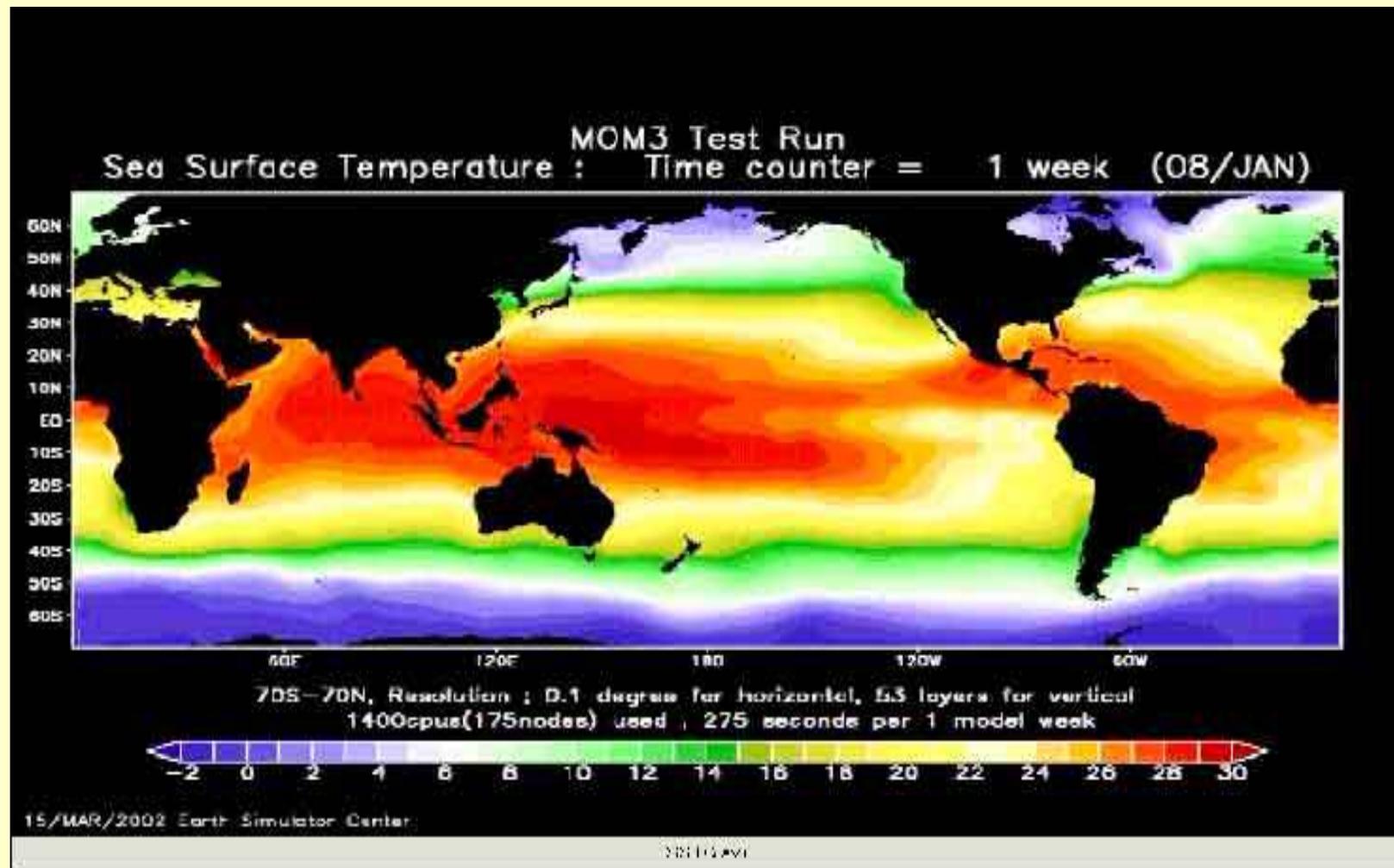
Numero totale di processori	5,120
Numero di cluster SMP	640
Interconnessione (crossbar)	20 TB/s
Memoria complessiva	10 TB

Costo	350M\$
Sito	Giappone
Prestazioni di picco	40.0Tflops
Prestazioni raggiunte	35.9Tflops

Tipologie di Computer

Applicazione: simulazione del Global Change

Es. Temperatura oceanica



Blue Gene/L (progetto IBM – 32K processori)

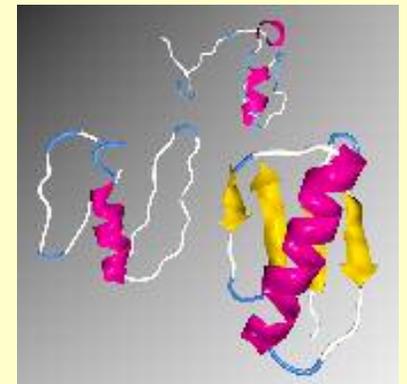
(1 nella top-500 list del 2004)

Prestazioni: 70/90 Teraflops ($=7 \times 10^{13}$ flop/sec)

Applicazioni

Scientifiche: ad es. *protein folding* (simulazione di 100 μ s di folding di una proteina di 32000 atomi richiederebbe 3M anni su un PENTIUM4! .. Ma ancora alcune decine di anni su Blue Gene!)

Commerciali: content distribution, webserving, data mining



IBM SP/3 (Configurazione a PD-ACL)

Proc. (PE)	Power3@375 MHz
Numero di PE	24
Cluster (nodi)	6 nodi SMP a 4 PE
RAM	24 Gbytes
Spazio disco	0.5 TB
Prestazioni	36 Gflops
Rete	Switch "MX2"

- Applicazioni scientifiche e commerciali
- Potenza media



Mainframe

- Il termine si riferisce storicamente a un computer potente collegato a molti *terminali stupidi*. Attualmente è usato come server per grandi sistemi informativi
- Molti processori, alcuni dei quali con funzioni particolari
- Gestione simultanea di molti utenti e di molti job
- Alta affidabilità e “backward compatibility”
- Intensa attività di Input/Output
- Esempi: IBM S/390, Unisys ES7000. IBM occupa circa il 90% del mercato.

Workstation

- Utente singolo (indipendente o *client*)
- Media potenza, costo moderato
- Elevate capacità grafiche
- Esempi: SUN SPARC, Silicon Graphics

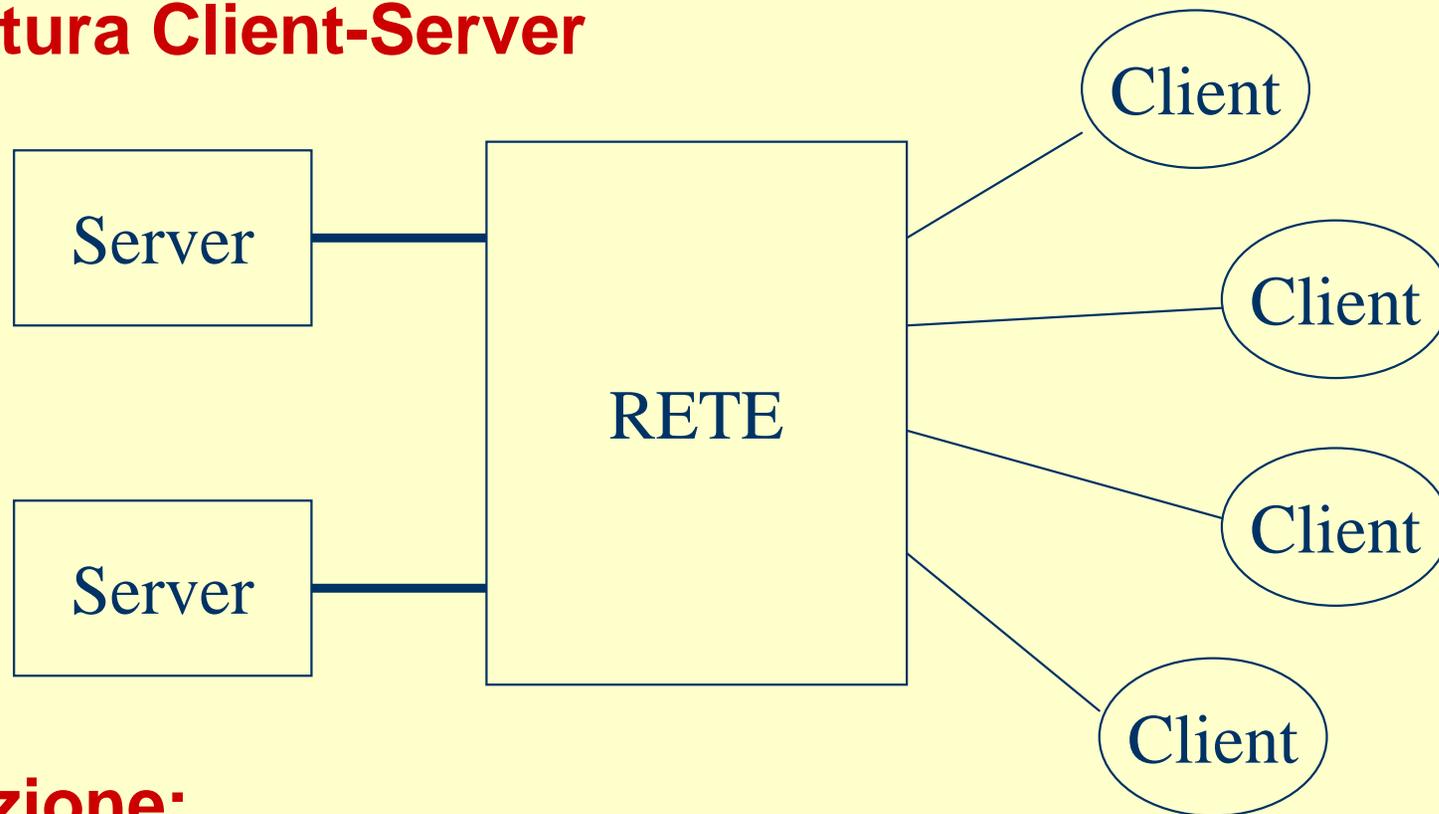
Personal Computer

- Utente singolo (indipendente o *client*)
- Bassa/media potenza, basso costo
- Esempi: Intel/AMD-based PC, Apple

Notebook: (Portatile)

Personal Digital Assistant: (Computer Palmare)

Architettura Client-Server



Applicazione:

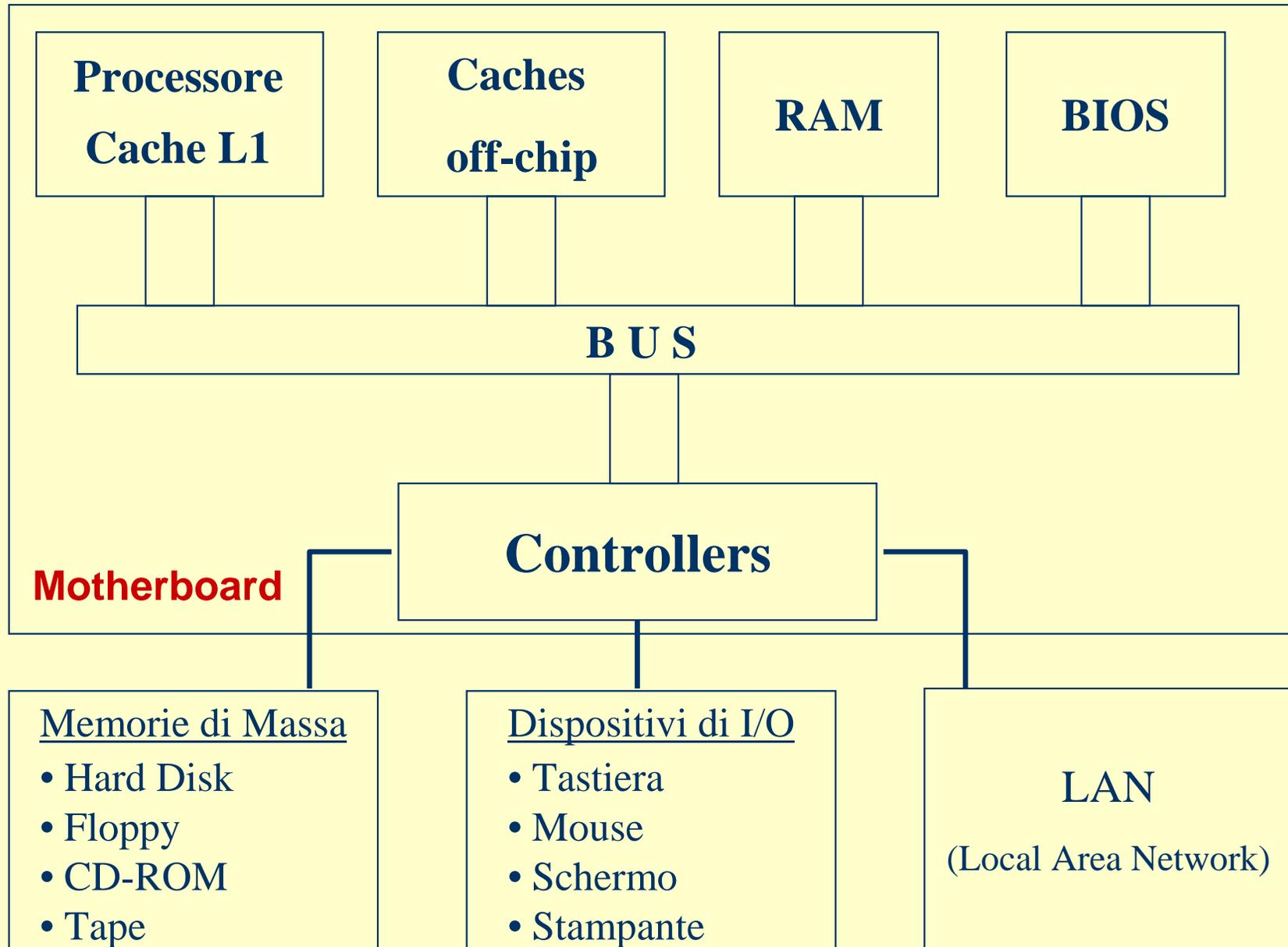
- Modulo *client* (*front end*): eseguito sul client, personalizzato per l'utente
- Modulo *server* (*back-end*): eseguito sul server, eroga servizi a più client

Network Computer

- Architettura Client-Server
- Client *leggeri*
- High-Speed Network

Downsizing: Passaggio da architetture basate su mainframe ad architetture di tipo Client-Server (o Network Computer), dovuto al migliore rapporto costo/prestazioni, alla maggiore flessibilità, alla minore obsolescenza e alla più facile manutenzione di queste ultime

Architettura Hardware di un Computer



VLSI (*Very Large Scale Integration*)

tecnologia per realizzare circuiti integrati su **chip** di silicio

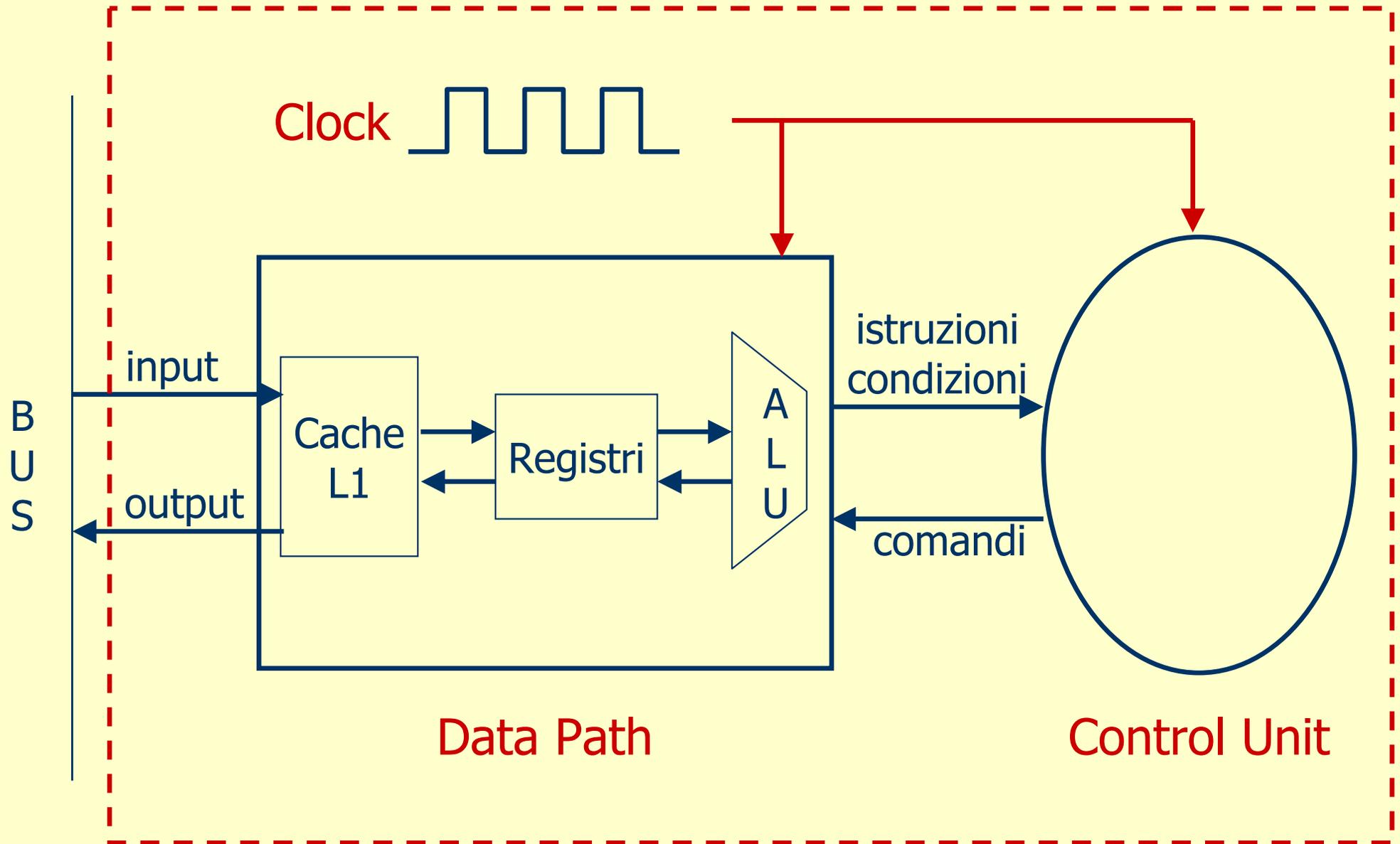
Microprocessore: Processore su un unico chip

Dimensione chip = $O(1)$ cm²

No. transistor per chip = $[10^7, 10^8]$

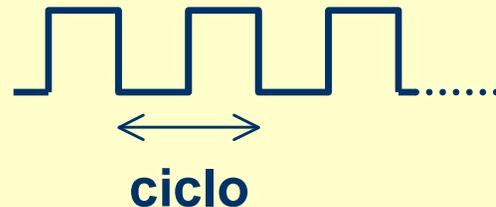
PROCESSORE

Processore



Caratteristiche Architeturali

1. Clock



500 Mhz \equiv $500 \cdot 10^6$ cicli per secondo

$\Rightarrow 1/(500 \cdot 10^6)$ secondi = 2 ns per ciclo

2. Instruction set: Insieme di istruzioni offerte dall'architettura

3. Superscalarità

Capacità di una architettura di eseguire più di una istruzione per ciclo di clock, ad es., tramite parallelismo nella circuiteria

Esempi: P6 (Pentium Pro, II, III) *three-way superscalar*, (sino a 3 istruzioni per ciclo di clock)

Misure di Performance

- **MIPS** (*Million Instructions Per Second*)
 - dipende dall'Instruction set (poco obiettivo)
- **FLOPS** (*FLoating-point Operations Per Second*)
 - dipende dall'insieme di istruzioni floating-point offerte dall'architettura (più obiettivo rispetto a MIPS)
- **Benchmark**: programmi scritti in linguaggi ad alto livello rispetto ai quali si valutano le architetture. E' la misura più obiettiva e più usata.

MEMORIA

1. Memorie volatili: Non mantengono i dati in assenza di alimentazione

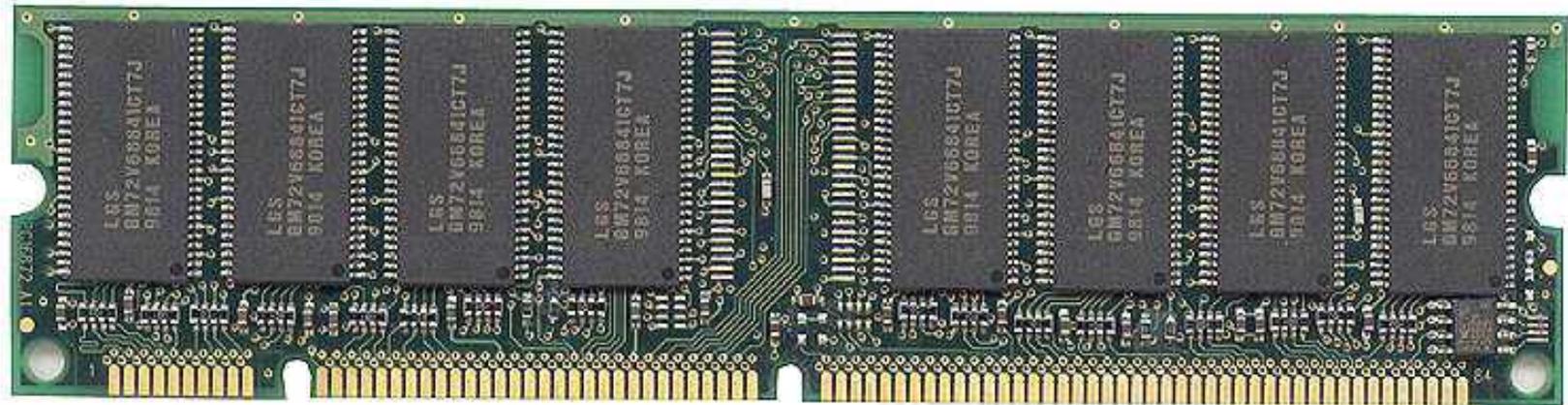
» **SRAM** (*Static Random Access Memory*)

- tempo di accesso 1÷10 ns
- utilizzata per cache L1/L2

» **DRAM/SDRAM** (*(Synchronous) Dynamic RAM*)

- tempo di accesso 50÷100 ns
- utilizzate per la memoria principale (RAM)
- costo 5÷10 volte inferiore alle SRAM

DIMM: Dual In-line Memory Module (SDRAM)



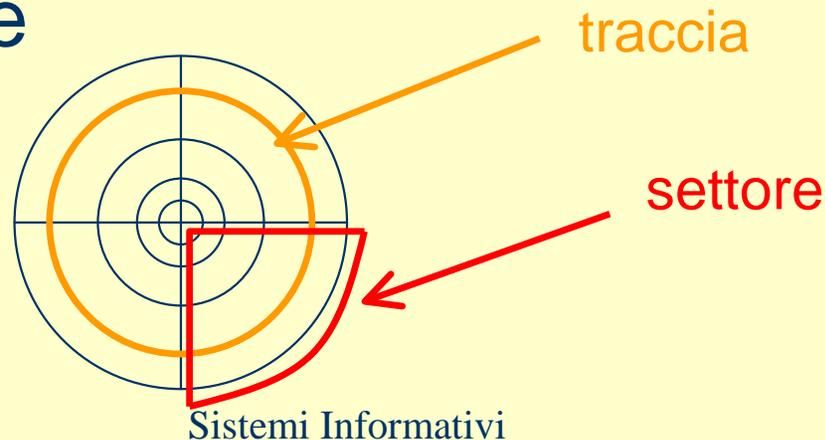
2. Memorie ROM (*Read Only Memory*)

- non volatili e, in alcuni casi, read-only
- utilizzate per memorizzare software di base, ad es. il BIOS
 - **PROM**: programmabile una sola volta
 - **(Flash) EPROM**: programmabile più volte

3. Memorie Magnetiche

3.1 Dischi

- Disk Pack (uno o più dischi)
- Testina con movimento radiale
- Ogni faccia suddivisa in tracce e settori
- Cilindro = insieme di tracce corrispondenti su facce diverse



Tipologie:

- **Floppy**

- non in continua rotazione
- tempo di accesso: $\cong O(100)$ ms

- **Hard Disk**

- in continua rotazione
- tempo di accesso: $\cong O(10)$ ms

Tempo di accesso (a un settore) =

$$SEEK + LATENCY + TRANSMISSION$$

- SEEK = ricerca della traccia
- LATENCY = ricerca del settore da trasferire
- TRANSMISSION = trasmissione del settore

Oss. Si possono accedere blocchi di più settori

Esempio: tempo di accesso a un settore di 2^9B per un disco con i seguenti valori medi:

- SEEK = 0.009 sec
- Velocità di Rotazione = 7200 rpm (revolutions per minute)
- Transfer rate = 4 MB/sec = 2^{22}B/sec

$$0.009 + 0.5 \cdot 60 / 7200 + 2^{-13} \cong 0.0133 \text{ sec} = 13.3 \text{ ms}$$

RAID

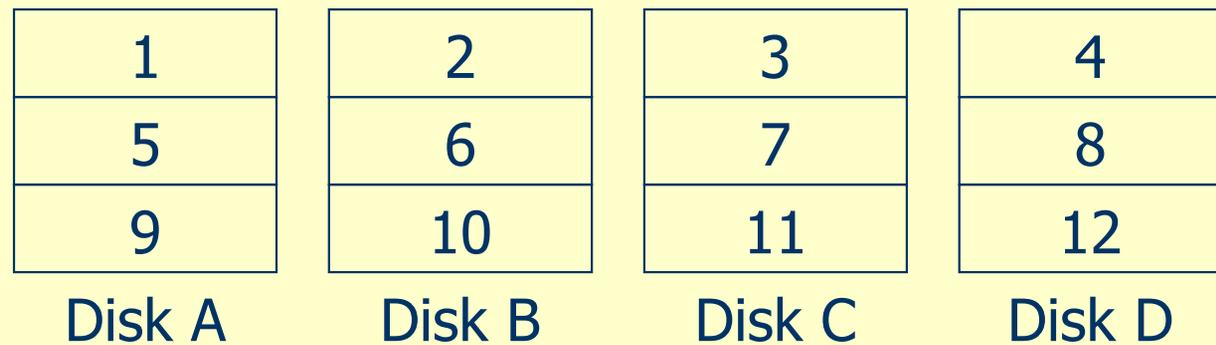
Redundant Array of Independent Disks

Combina diversi hard disk in una stessa unità logica offrendo (a costi contenuti):

- Elevato throughput (accesso parallelo ai dati)
- Protezione contro la perdita di dati
 - Riduzione del downtime

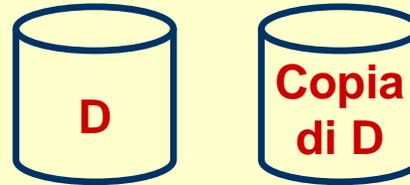
Tecnologie usate nei RAID

Striping:



accesso parallelo → elevato throughput

Mirroring:



Parity:

disk	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
A	1	0	0	0	0	1	1	1
B	0	0	0	0	1	0	1	1
C	0	1	1	0	0	0	0	0
D	1	0	0	1	1	1	0	1
parity	0	1	1	1	0	0	0	1

- Entrambe le tecniche servono per la protezione contro la perdita di dati
- Mirroring ha un costo maggiore

Livelli di RAID

livello	striping	mirroring	parity
RAID 0	X		
RAID 1		X	
RAID 0/1	X	X	
RAID 5	X		X

3.2 Nastri

- Usati prevalentemente come unità di back-up
- Accesso sequenziale ai dati (vs dischi)

4. Memorie Ottiche

- Basate sulla riflessione di raggi laser
- Tipologie: CD-ROM, DVD (Digital Video Disc)

PROCESSORE vs MEMORIA

Processore vs Memoria

- Gap crescente tra **ciclo del processore e tempo di accesso alla memoria (RAM)**. Trend negli ultimi 20 anni:

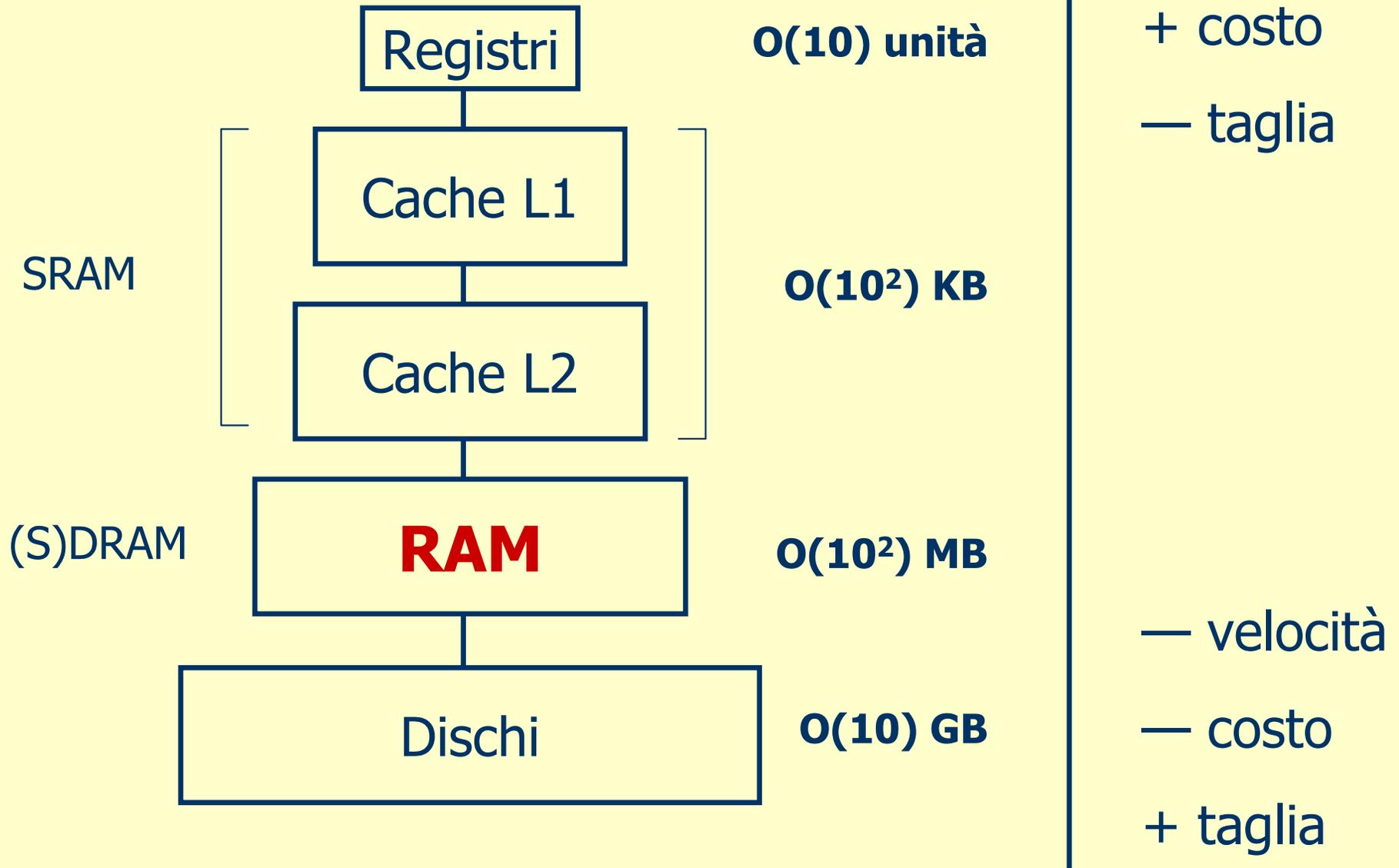
Clock rate	+30%	per anno
Memory access time	+7%	“ “
Performace	+55%÷80%	“ “ (su benchmark)

- Aumento della richiesta di memoria e di prestazioni da parte delle applicazioni (applicazioni più complesse e/o taglie di input maggiori)

⇒ **Gerarchia di memoria e parallelismo**

Gerarchia di memoria

Gerarchia di memoria:



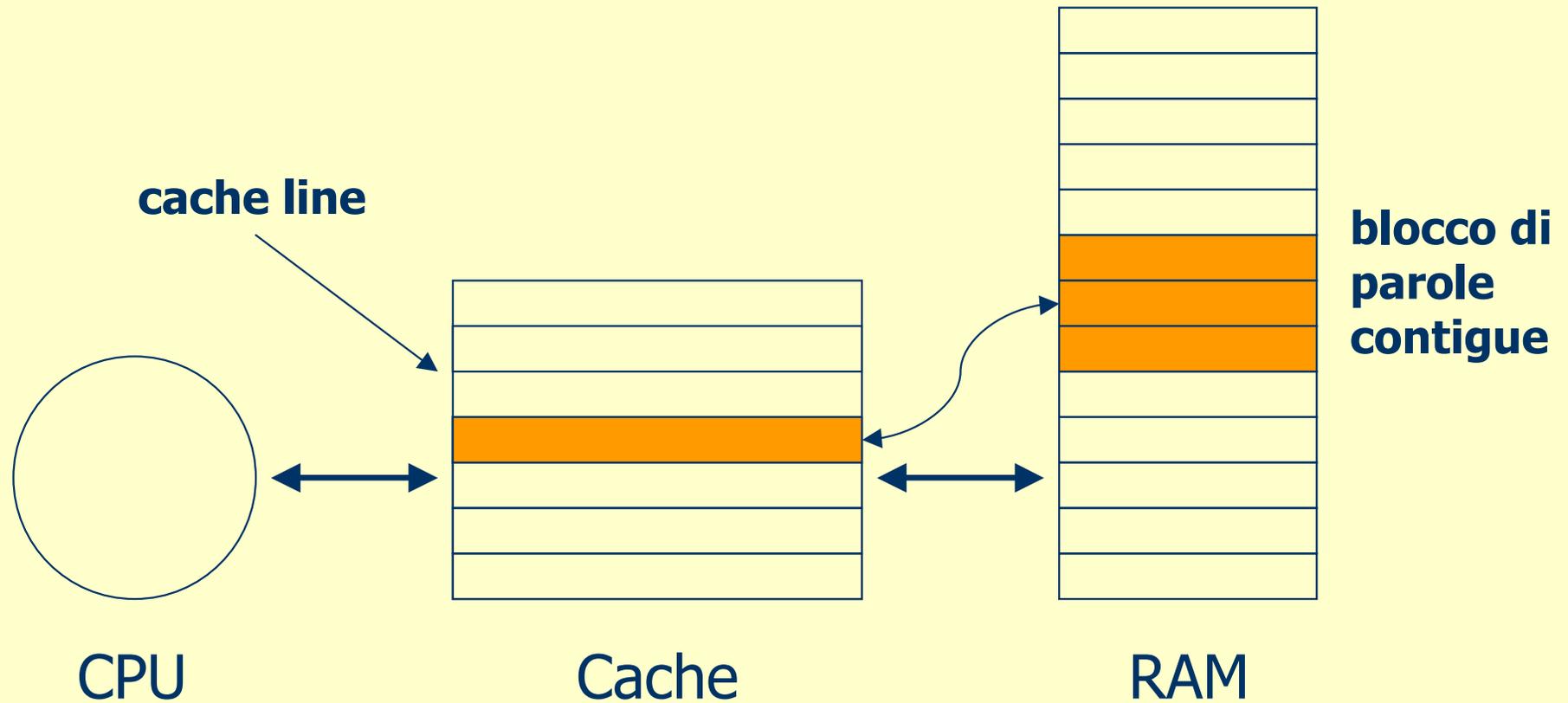
Obiettivo: Fornire molta memoria al costo delle tecnologie più economiche e con le prestazioni delle tecnologie più costose.

Principio di Località: I programmi accedono porzioni limitate dei memoria durante (breve) intervalli di tempo.

LOCALITA' TEMPORALE: riuso ravvicinato nel tempo delle stesse parole di memoria

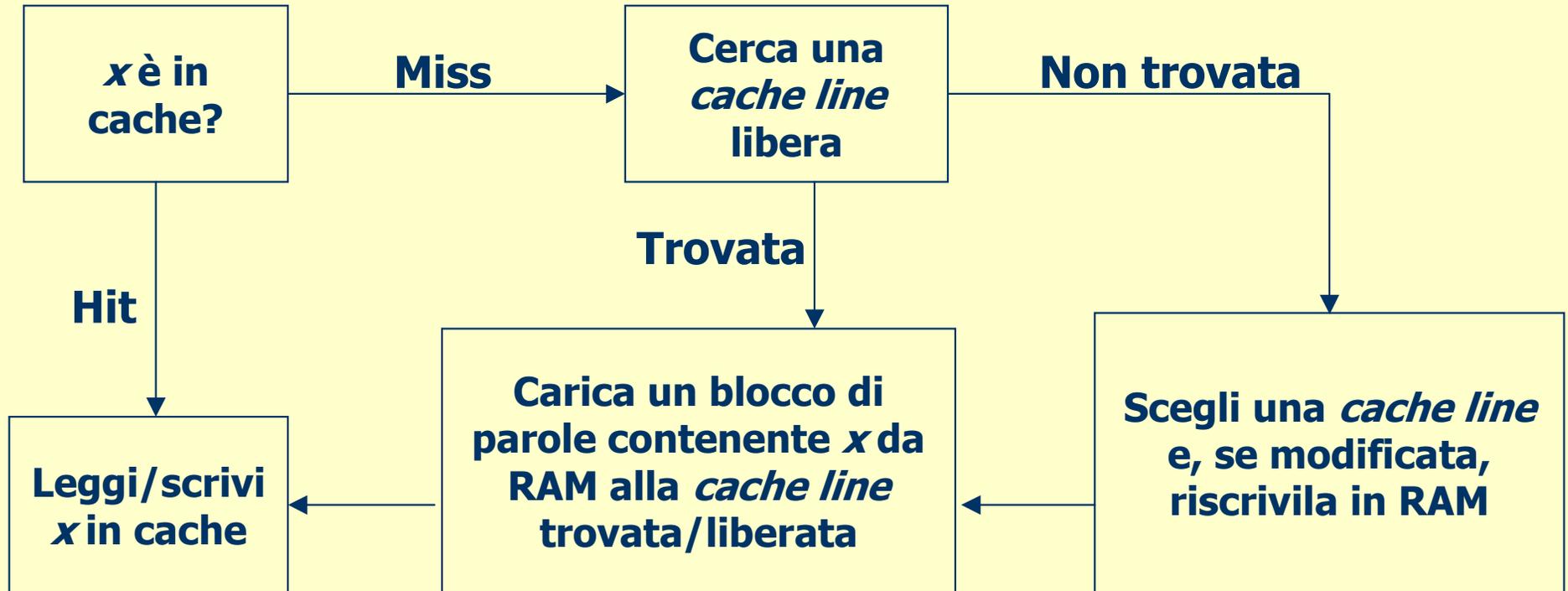
LOCALITA' SPAZIALE: Uso di parole consecutive in istanti di tempo successivi

Funzionamento della Cache



Gerarchia di memoria

Lettura/scrittura di una parola x :



Sfruttamento della Località:

Località Temporale → accessi alla stessa parola in cache

Località Spaziale → accessi a parole consecutive nella stessa cache line

SISTEMA OPERATIVO

Sistema Operativo

Sistema Monotasking: (obsoleto) un solo programma in memoria in ogni istante

Sistema Multiprogrammato/Multitasking: più programmi in memoria, in esecuzione concorrente

pre-emptive multitasking: ogni processo usa la CPU per un dato quanto di tempo, passato il quale la cede a un altro processo. Lo scheduler sceglie, dalla ready list il processo a cui assegnare la CPU di volta in volta.

Timesharing: multitasking per più utenti

Sistema real time: vincoli di tempo rigidi/critici

Kernel del sistema operativo:

- Gestione dei processi/processori
- Gestione della memoria centrale
- Gestione della memoria secondaria
- Gestione delle periferiche

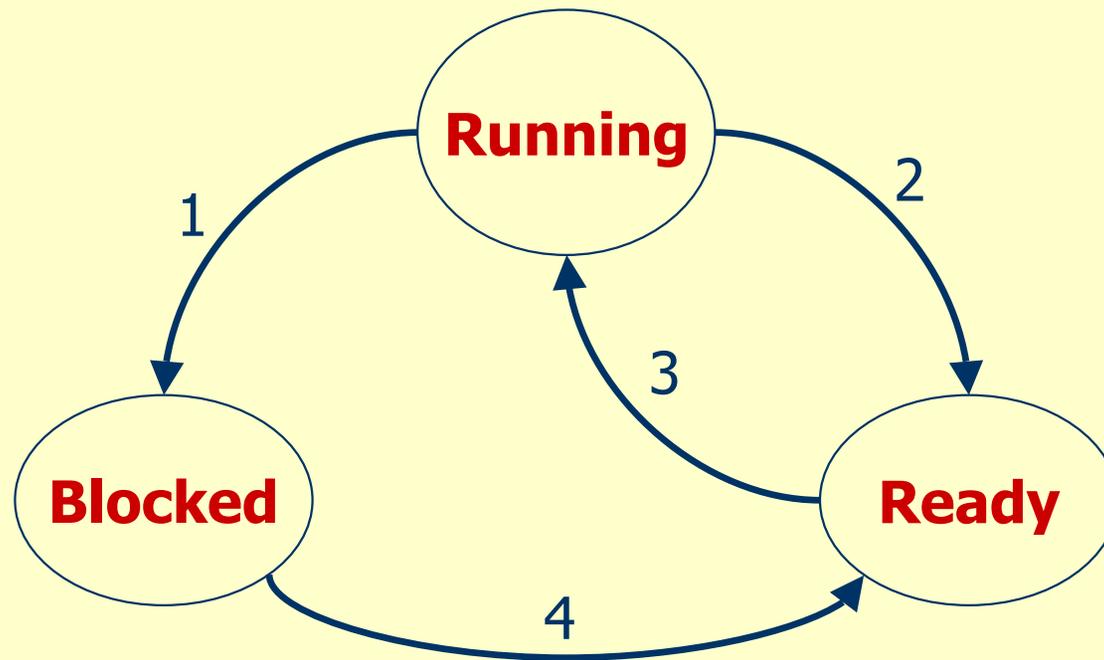
Processo:

- programma eseguibile
- stato (stack, registri, owner, ecc.)

N.B.: Ci possono essere più processi in esecuzione relativi a uno stesso programma

Process Table: array di record utilizzati per mantenere lo stato dei processi

Stati di un processo: (pre-emptive multitasking)



- 1: Attesa di un evento (es. input)
- 2: *Interrupt* (es. esaurimento del quanto di tempo)
- 3: Selezione dalla *ready list* da parte dello *scheduler*
- 4: Avvenimento dell'evento atteso

Context switching: processo p lascia la CPU al processo q

- salva lo stato di p nella process table
- ripristina lo stato di q dalla process table

Memory manager: svolge le seguenti funzioni:

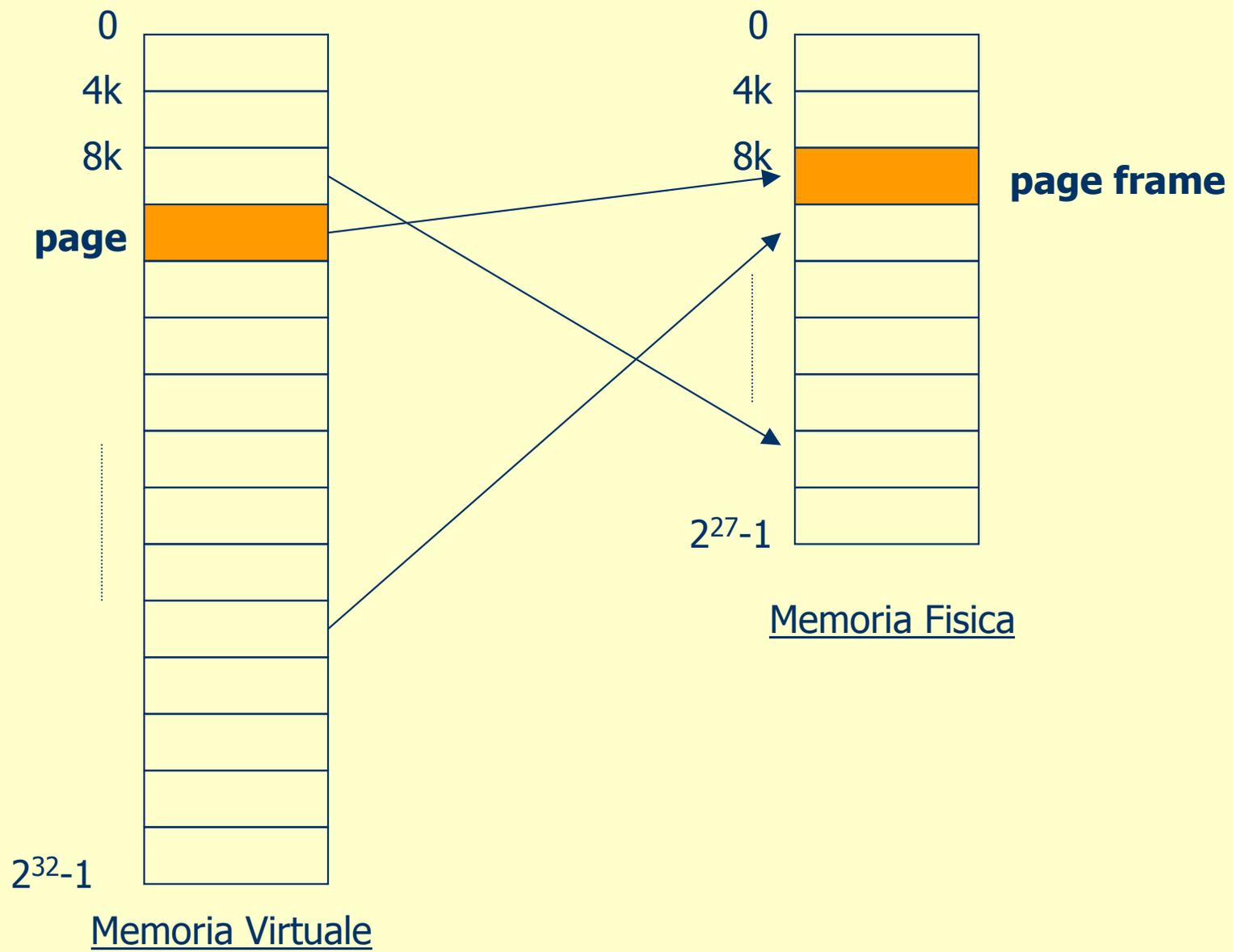
- tiene traccia delle zone di memoria libere e occupate
- assegna/toglie la memoria RAM ai processi
- gestisce lo *swapping* tra RAM e dischi

Memoria Virtuale

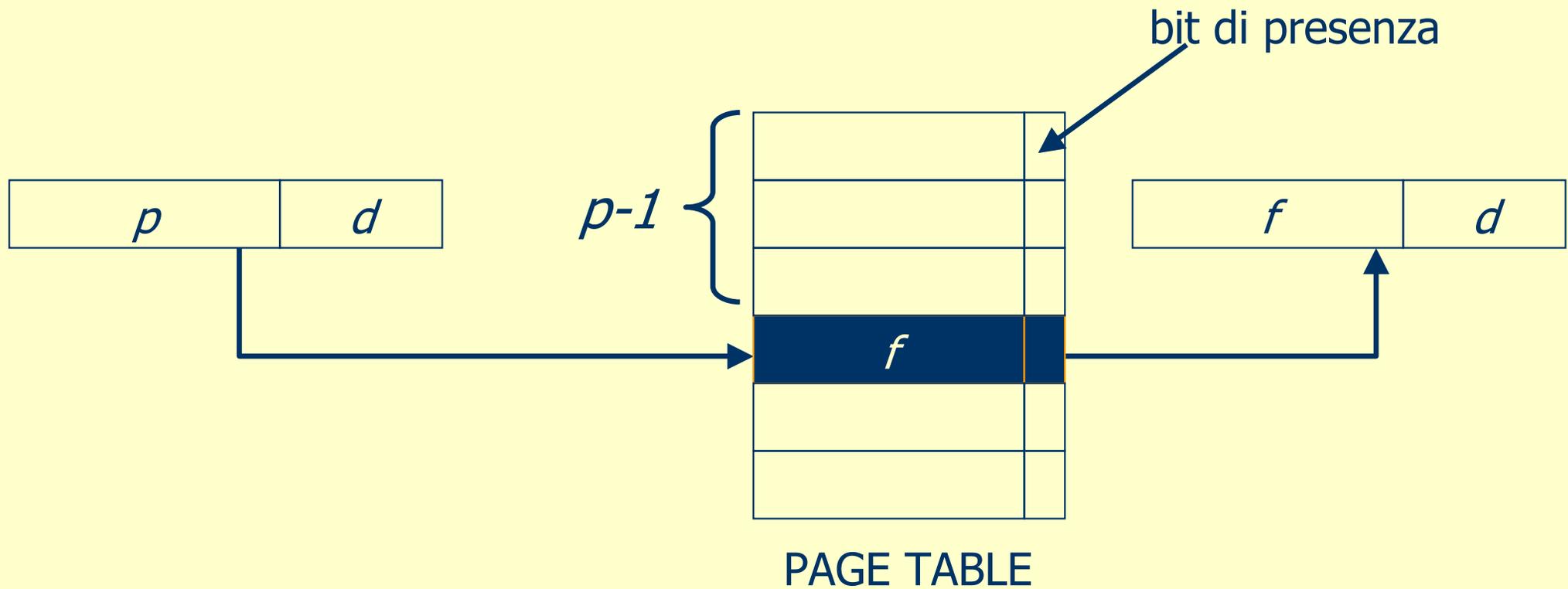
- ogni programma fa riferimento a uno spazio di indirizzi virtuali o memoria virtuale
es. indirizzi a 32 bit \Rightarrow indirizzi virtuali $\in [0, 2^{32}-1]$
- $|memoria\ virtuale| \geq |RAM|$
(es. $2^{32} = 4\ GB \geq 128\ MB$)
- la memoria virtuale è implementata utilizzando RAM e disco

Paginazione

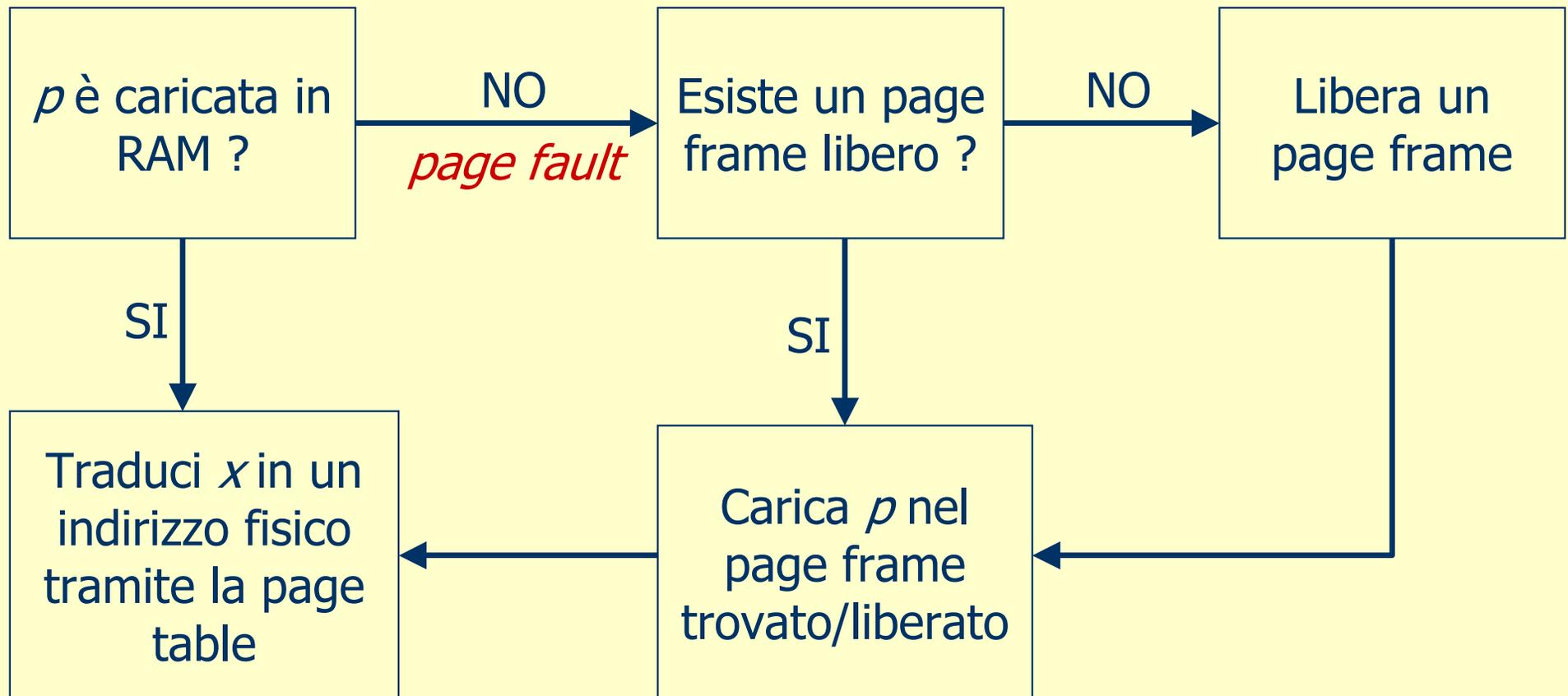
- memoria virtuale e memoria fisica suddivise in *pagine* della stessa taglia (es. 4 KB). Il termine *page* si riferisce alla pagina virtuale, mentre il termine *page frame* a una pagina fisica
- per ogni processo si crea una corrispondenza tra le proprie pagine virtuali e le pagine fisiche. Tale corrispondenza è mantenuta in una *page table* associata al processo
- ogni riga della page table, univocamente associata a una pagina virtuale, indica se essa è caricata nella memoria fisica e, in questo caso, indica il page frame corrispondente
- le pagine virtuali non in memoria fisica sono mantenute in una *area di swap* su disco, in una porzione allocata al processo



Traduzione: indirizzo virtuale \rightarrow indirizzo fisico



Accesso a un indirizzo di virtuale $x \in \text{page } p$



Osservazioni:

- la page table di un processo risiede in RAM
- le righe della page table più recentemente accedute sono mantenute nel *Translation Lookaside Buffer (TLB)* una memoria cache ad accesso veloce all'interno del processore. In questo modo, in assenza di page fault, si riducono gli accessi in RAM dovuti alla page table
- nel caso di context switch è sufficiente salvare (ripristinare) un puntatore alla page table del processo uscente (entrante) e ripulire il TLB

File System: organizza l'informazione in *file*. Svolge le seguenti funzionalità:

- realizza mapping *nomefile* → *indirizzo su disco*
- offre primitive di accesso ai file
- gestisce i diritti di accesso ai file
- associa a ogni file un descrittore (ID, nome, owner, ecc.)

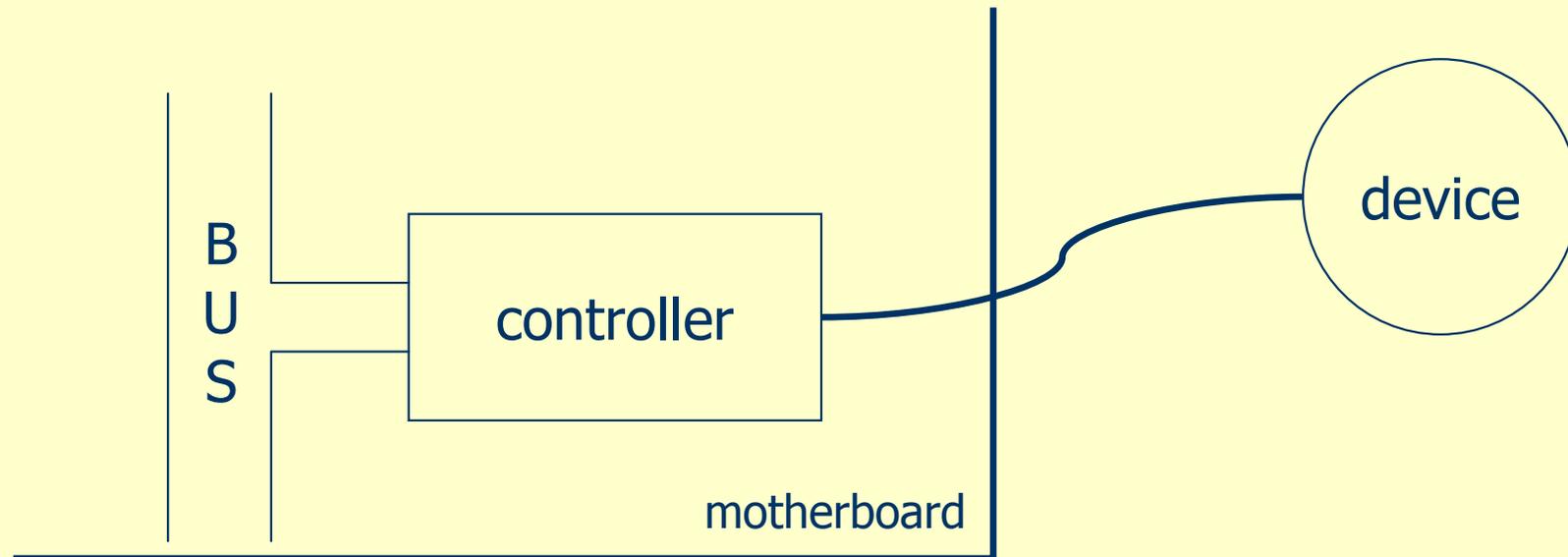
File System distribuito: implementato in un'architettura client-server in cui tutto o parte del file system visibile da un computer (client) risiede fisicamente sul disco o sui dischi di altri computer (*file server*)

Mounting: collegamento di un (pezzo di) file system residente su un disco come sottoalbero di un file system

Gestione delle periferiche

Device Driver: software di gestione della periferica (device dependent)

Device Controller/Adapter: circuiteria che controlla l'attività della periferica



Funzionalità del driver:

- offre una visione uniforme dell'organizzazione dei dati nella periferica e delle funzionalità di I/O
- traduce in comandi per il controller le richieste provenienti da strati di software superiori (device-independent)
- legge informazioni di stato dai registri del controller
- verifica la presenza di errori

Funzionalità del controller:

- riceve/invia segnali dalla/alla periferica
- interpreta i comandi messi nei suoi registri dal driver
- invia interrupt alla CPU
- interagisce con la memoria
- memorizza in un buffer interno i dato ricevuti o da inviare alla periferica
- esegue (parziale) correzione di errori sui dati ricevuti dalla periferica