Richiami sulle architetture dei calcolatori

Corso di Laurea in Ingegneria Informatica

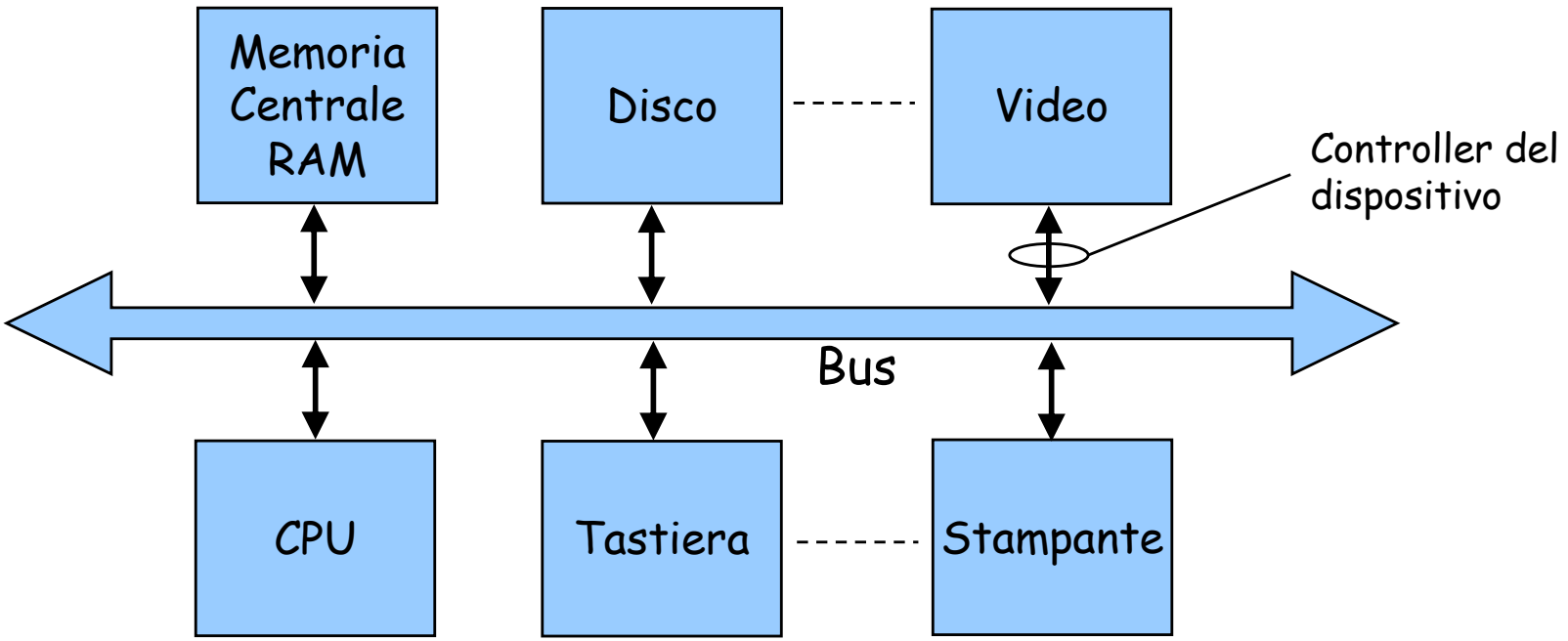
Università degli Studi di Napoli Federico II

Anno Accademico 2024/2025, Canale San Giovanni

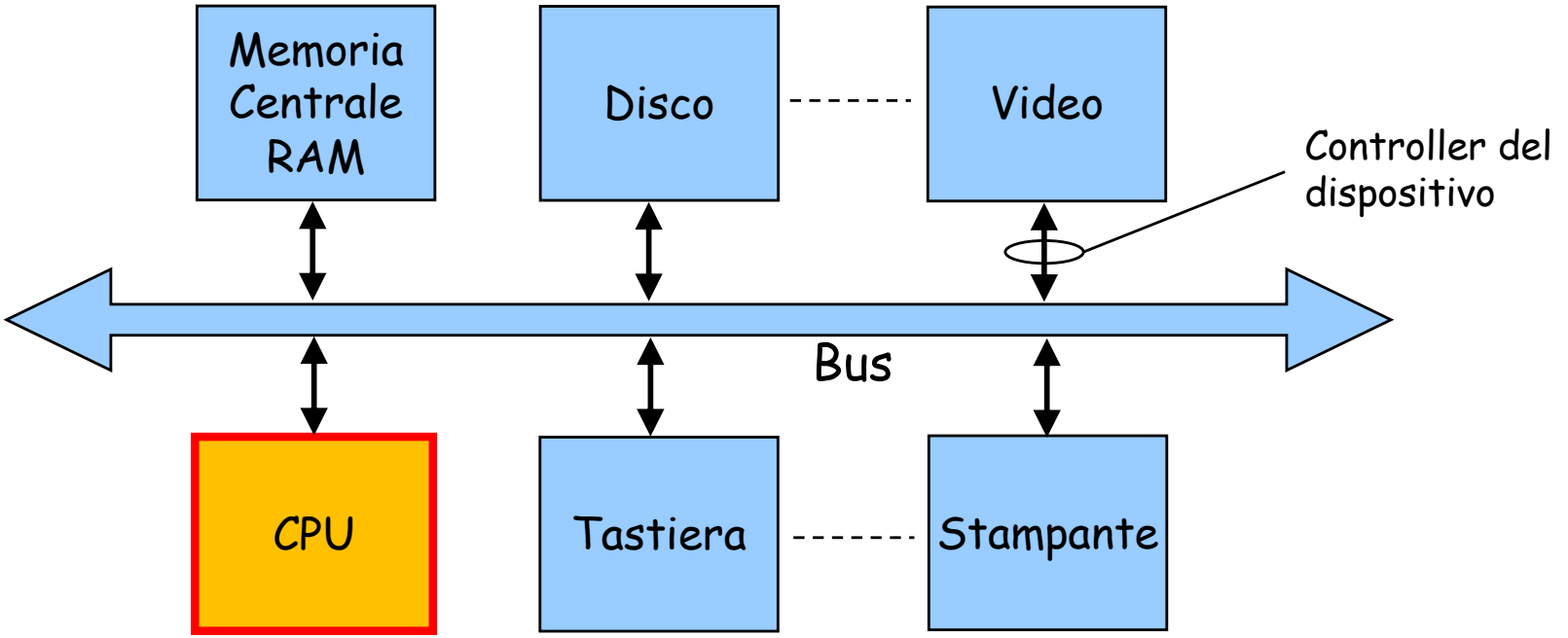
Richiami sulle architetture dei calcolatori

- Sommario
  - Gestione della comunicazione fra processore e dispositivi periferici
  - Gestione delle interruzioni
  - Protezione della memoria
  - Gerarchia di memoria
- Riferimenti
  - P. Ancilotti, M.Boari, A. Ciampolini, G. Lipari, “Sistemi Operativi”, Mc-Graw-Hill (Cap.1, Par. 1.3)

Architettura di un sistema di elaborazione



Architettura di un sistema di elaborazione

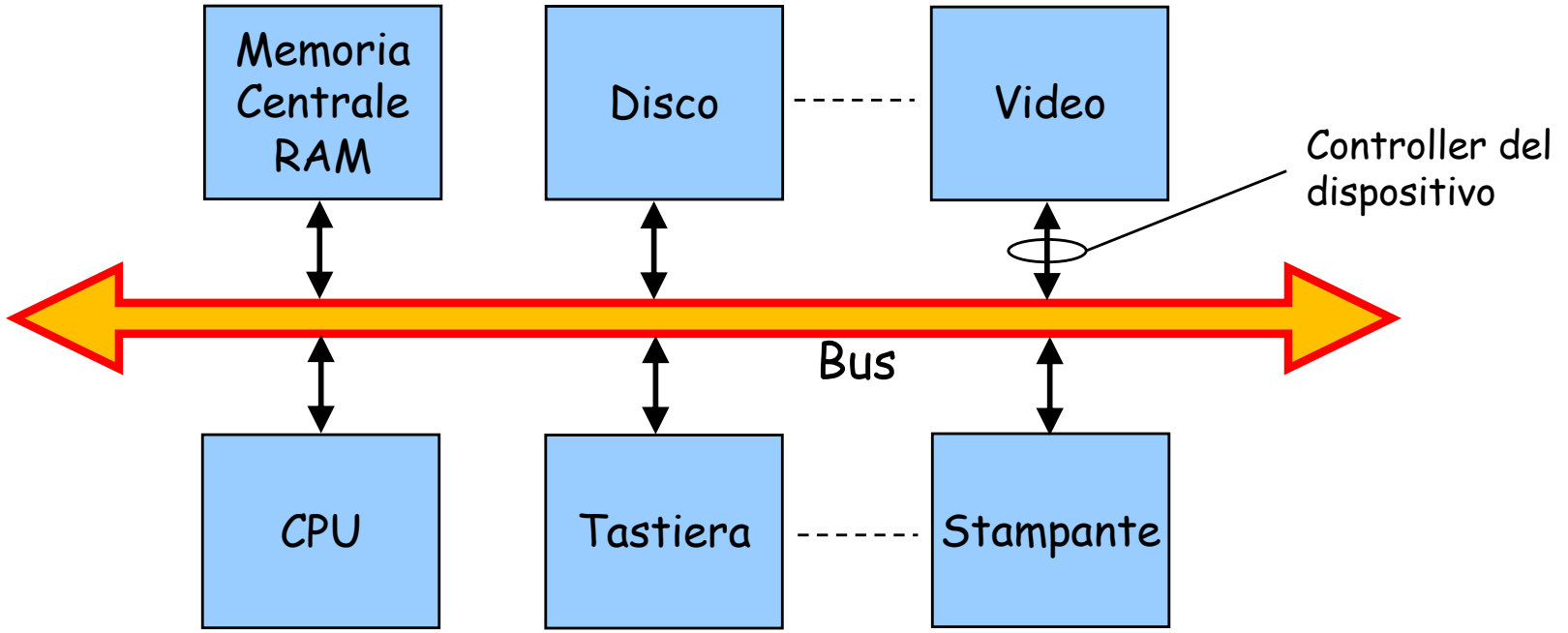


Durante l'esecuzione di un programma, la **CPU** accede ai propri **registri interni**, alla **memoria** e alle **periferiche di I/O**

Registri interni della CPU

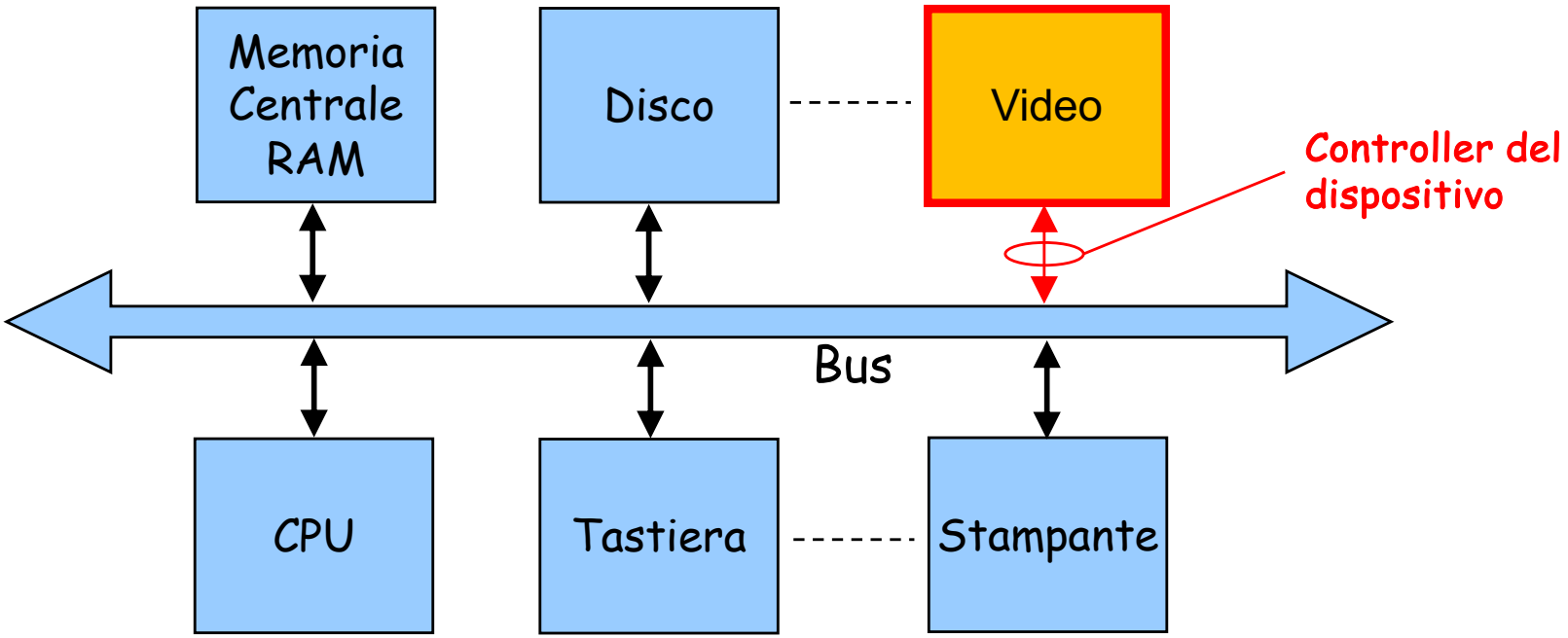
- Program Counter (PC)
- Registro di Stato (PS)
- Stack Pointer (SP)
- Registri generali
- ...

Architettura di un sistema di elaborazione



Vi è scambio di **dati** e **indirizzi** su un **bus** di sistema, con parallelismo a 8, 16, 32, oppure 64 bit

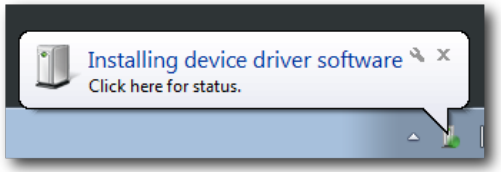
Architettura di un sistema di elaborazione



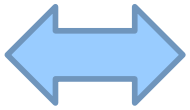
La CPU accede ai dispositivi di I/O (meccanici/elettrici), comunicando con un **controller** (circuito elettronico)

Interfacciamento tra CPU e I/O

- **Controller**:
  - Sistema elettronico, che gestisce il dialogo tra dispositivo e CPU
  - Comunica con la CPU tramite il bus di sistema
  - Comunica con il dispositivo tramite collegamento fisico es. USB
- **Device driver**:
  - Software di controllo del dispositivo, in esecuzione sulla CPU
  - È parte del sistema operativo (verrà approfondito più avanti nel corso)

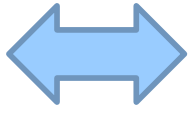


CPU



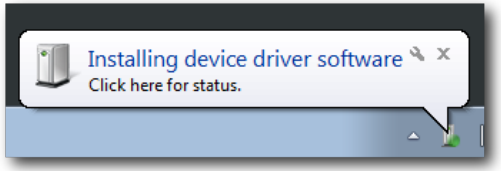
Bus di sistema

Controller

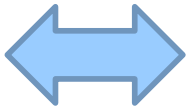


Interfacciamento tra CPU e I/O

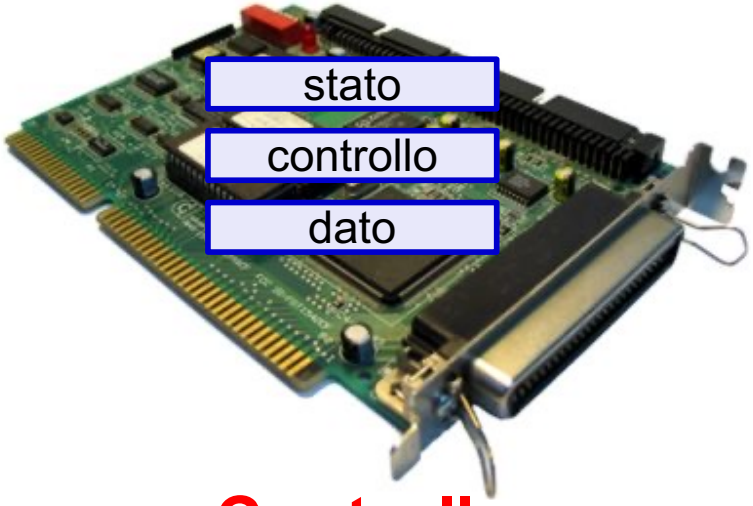
- Il controller ha 3 tipi di registro: **dato**, **stato**, **controllo**
  - Il device driver **legge il registro di stato**, per determinare lo stato della periferica (es., "*in attesa di dati*", "*trasferimento in corso*", "*trasferimento completato*", etc.)
  - Il device driver **scrive sul registro di controllo**, per impartire dei comandi alla periferica (es., "*avvia trasferimento*", "*reset*", etc.)
  - Il device driver **legge/scrive sul registro dato**, per prelevare/inviare dati alla periferica



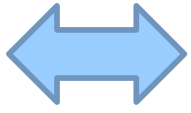
CPU



Bus di sistema

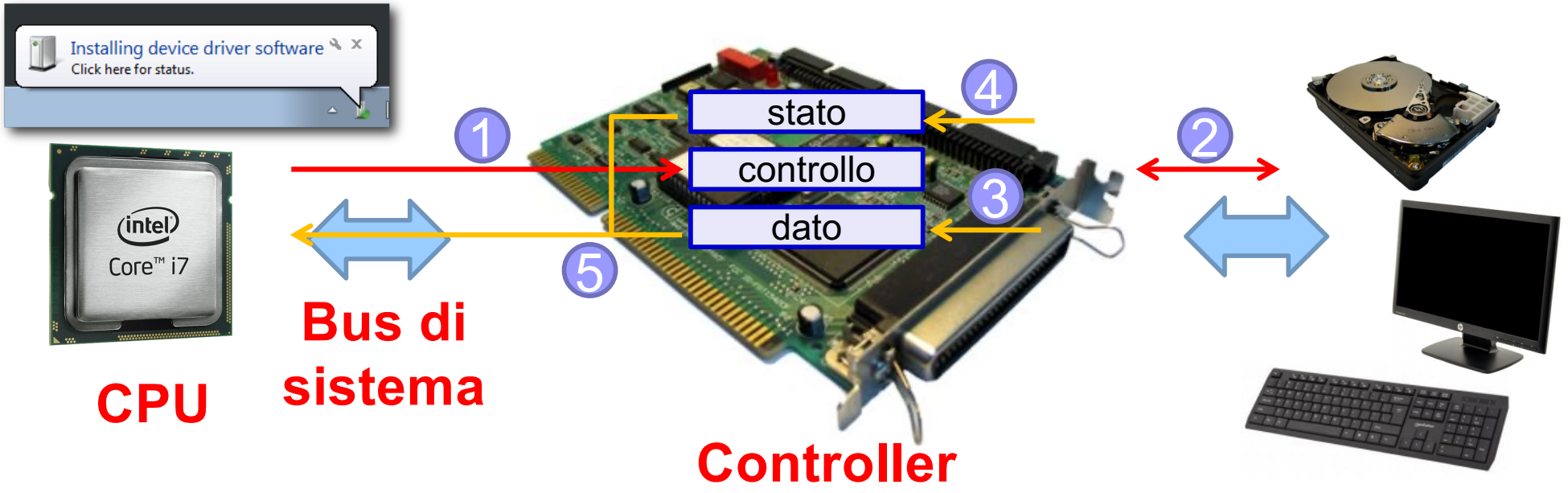


Controller



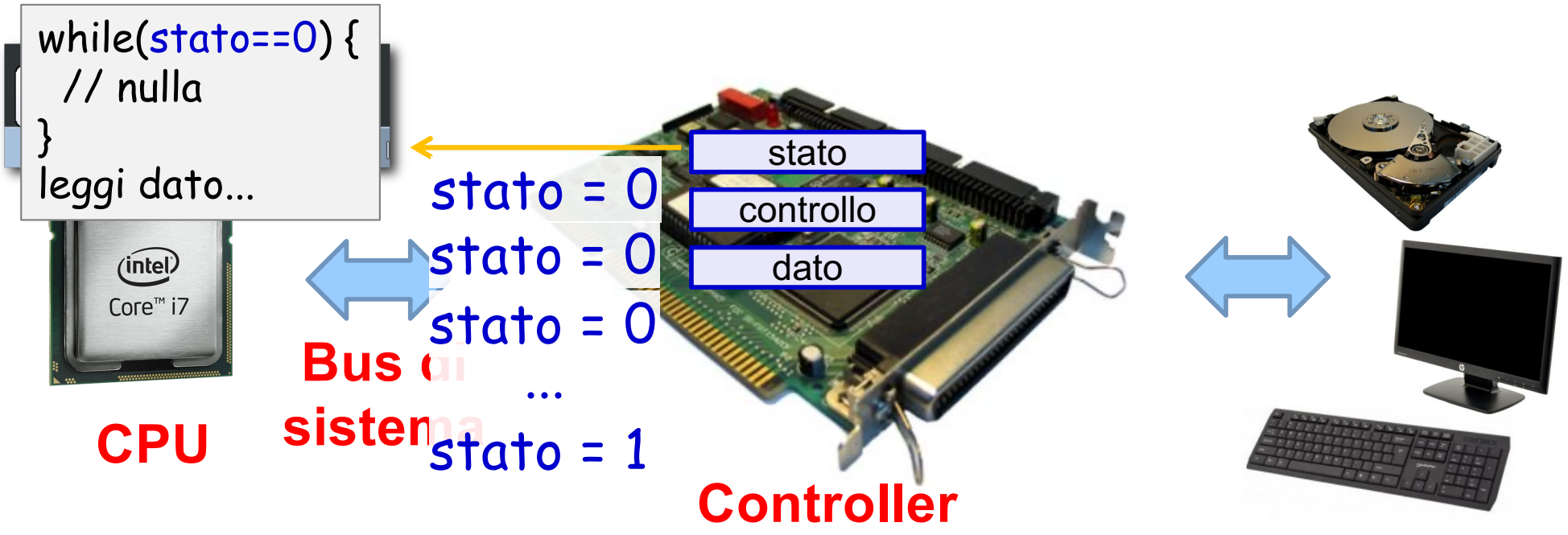
Interfacciamento tra CPU e I/O

- **Esempio**: lettura di un dato da un dispositivo di I/O
  1. Il driver scrive su **registro controllo** un valore che rappresenta il comando di lettura
  2. Il controller invia alla periferica dei segnali di comando per la acquisizione di dati
     ... attesa ...
  3. Il controller memorizza nel **registro dato** il dato ricevuto
  4. Il controller modifica il **registro stato** per indicare la presenza del dato, e la disponibilità del controller ad eseguire nuove operazioni
  5. Il driver nota il cambiamento di stato (tramite **polling** o **interrupt**), e preleva il dato



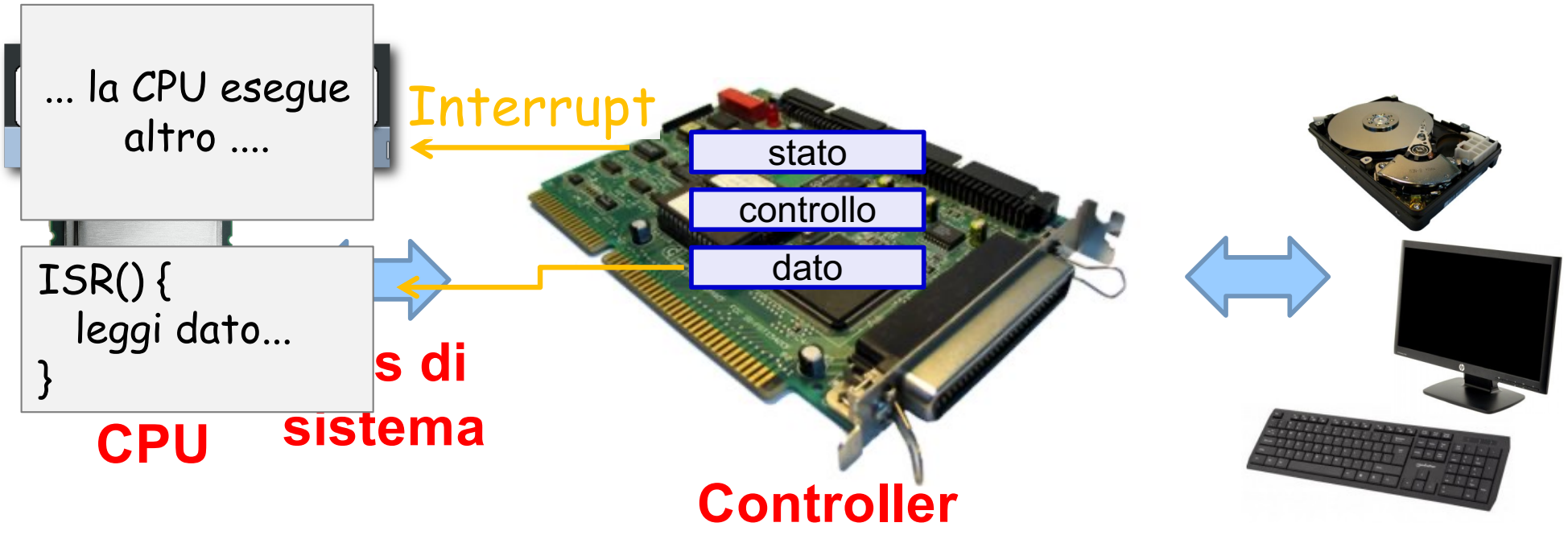
Polling e interrupt-based I/O

- **Polling** (o "I/O controllato dal programma")
  - Il driver **accede ripetutamente (in loop) al registro di stato**, fin quando non ci sono variazioni che indichino la disponibilità del dispositivo
  - Questa tecnica **impedisce di usare la CPU per altre attività** (**busy waiting**)



Polling e interrupt-based I/O

- **Interrupt-based**
  - Quando il dispositivo non è disponibile, la **CPU viene usata per altre attività** (si evita il busy waiting)
  - Quando diventa **disponibile**, il dispositivo solleva un interrupt, passando il **controllo della CPU alla ISR** (è parte del driver)



Interfacciamento di I/O

- La CPU accede ai registri del controller in due possibili modi

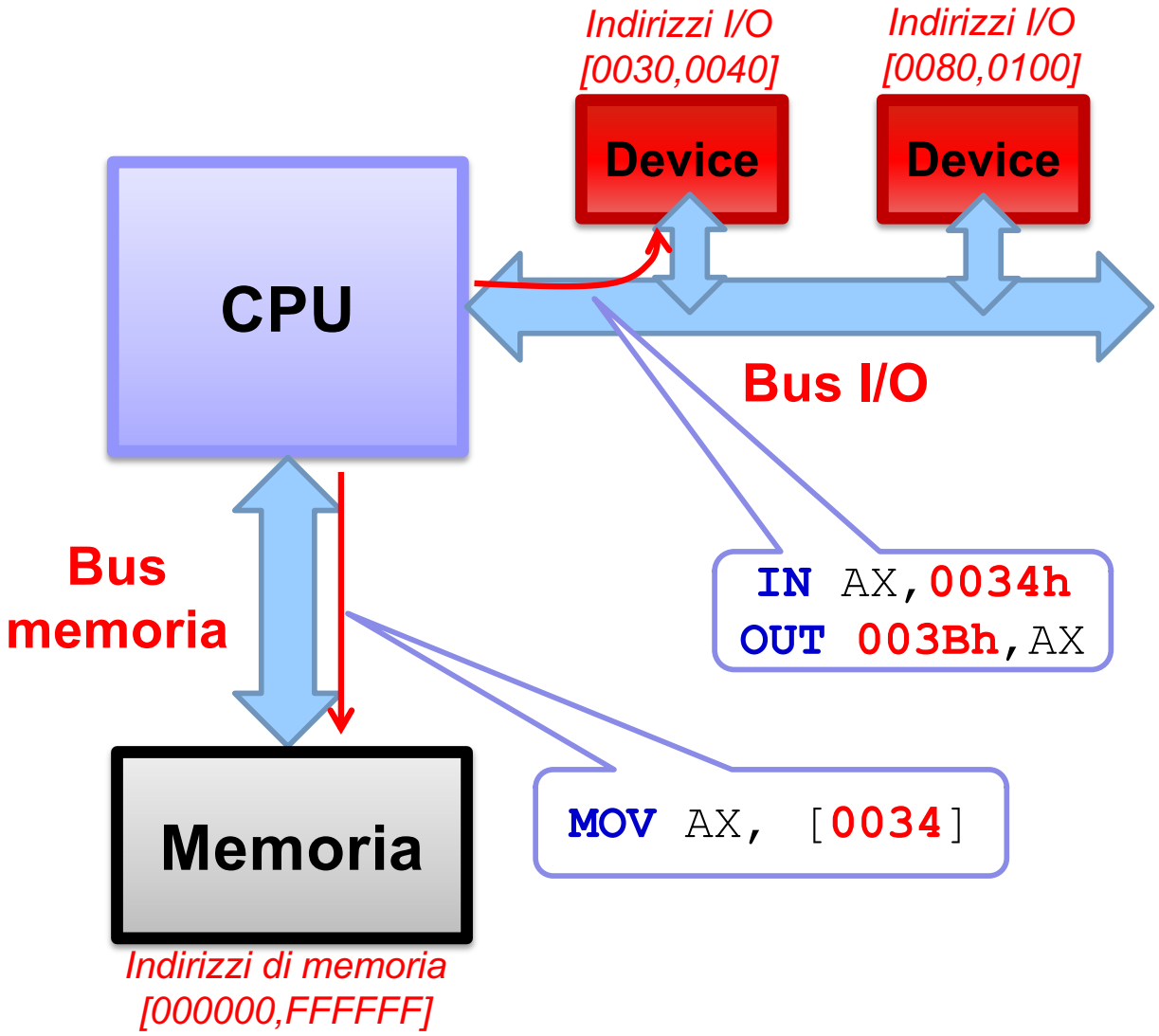
Port-mapped I/O

- Il driver legge/scrive sui registri con **istruzioni macchina speciali** della CPU
- Approccio dei sistemi più vecchi, con bus a parallelismo ridotto (es. 8 bit)

Memory-mapped I/O

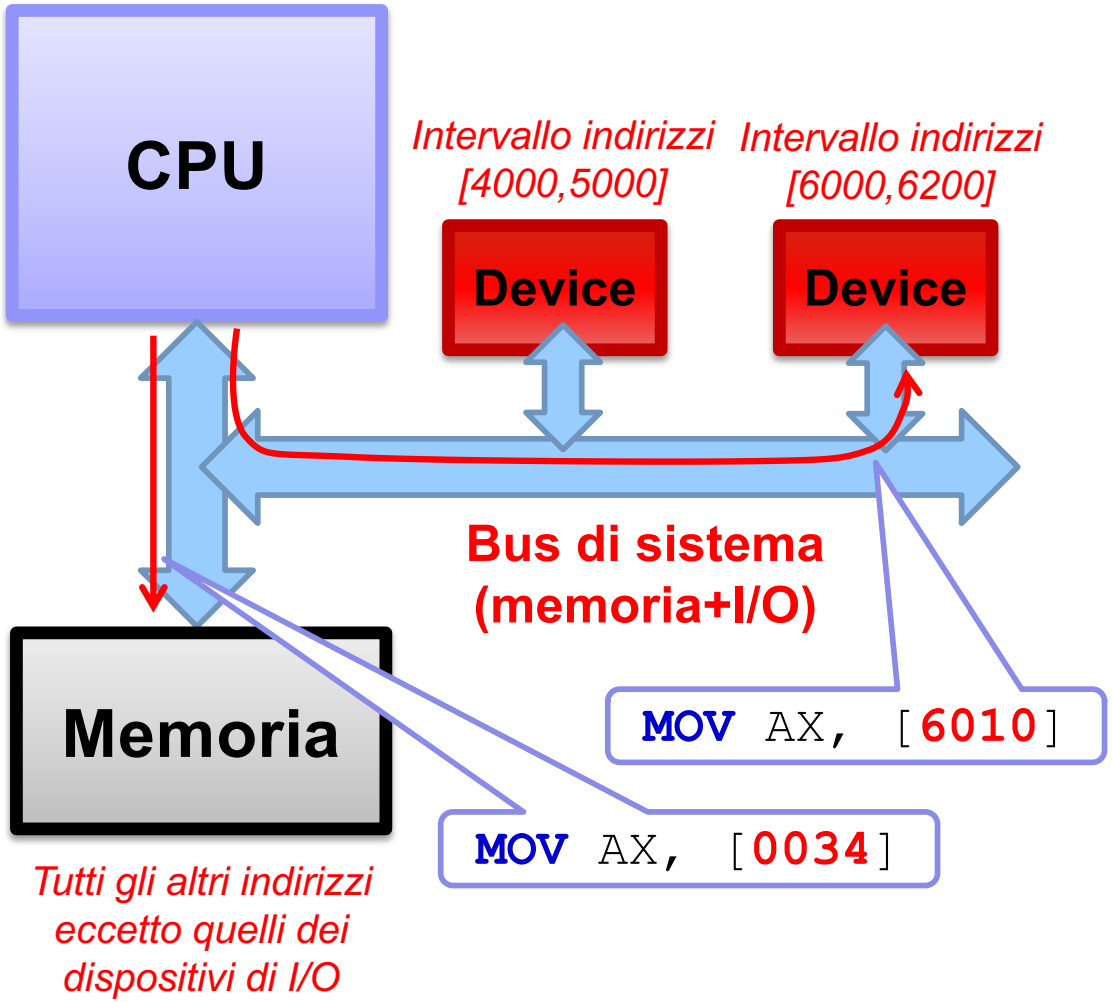
- Il driver legge/scrive sui registri utilizzando le **stesse istruzioni utilizzate per accedere alla memoria** (es. MOV)
- Sovrapposizione degli spazi di indirizzamento
- Usato nei sistemi moderni

Port-mapped I/O (es. Intel 8086)



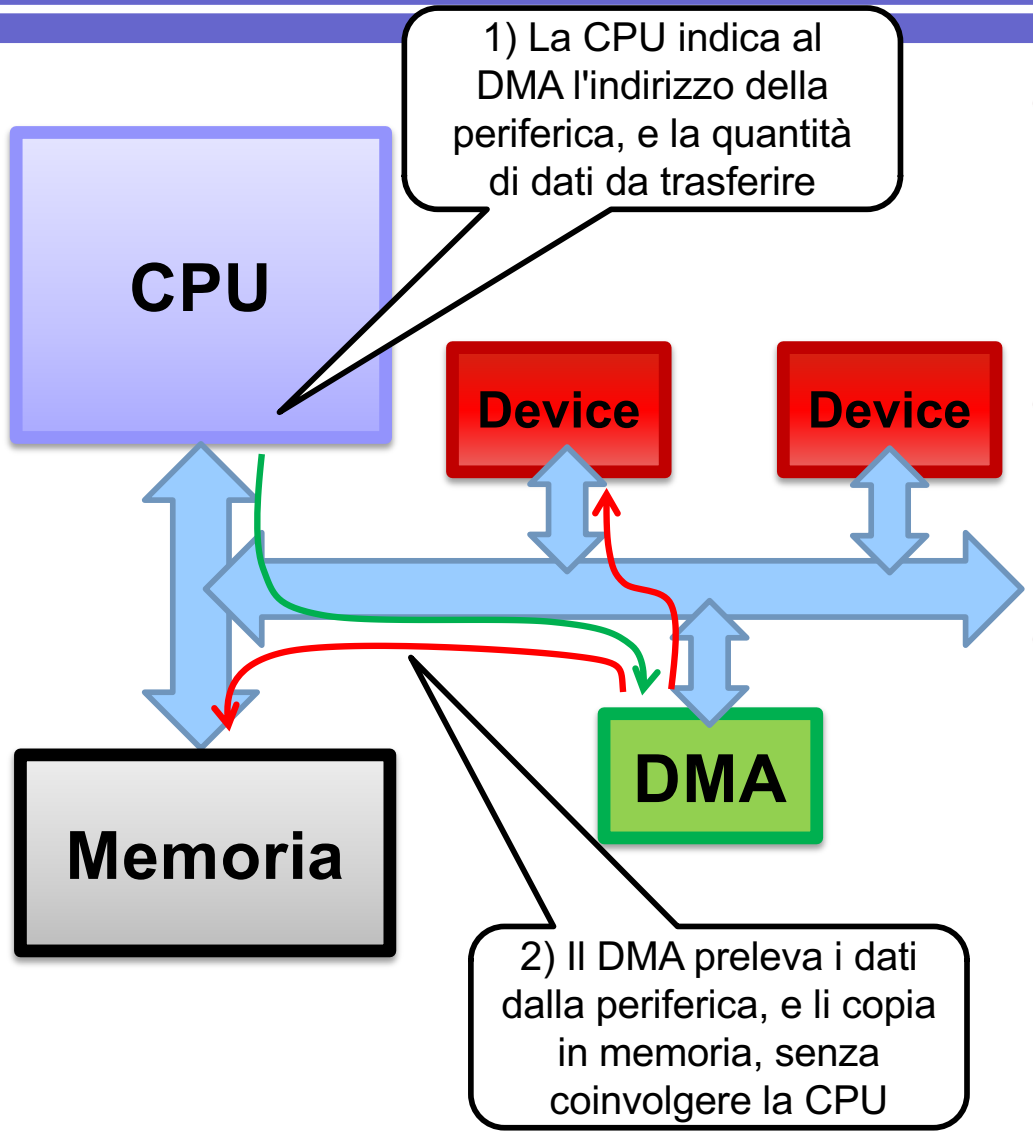
- Esempio di Intel 8086: si usano istruzioni speciali di I/O ("IN", "OUT")
- Distinte dalle operazioni di accesso alla memoria (es. "MOV")
- Le istruzioni di I/O e di memoria usano spazi di indirizzamento distinti

Memory-mapped I/O



- Le operazioni di I/O sono effettuate con le **stesse istruzioni che operano su memoria (es. "MOV")**
- Condividono lo stesso **spazio di indirizzamento**
- Ad ogni dispositivo è riservato un **intervallo di indirizzi**

Direct Memory Access (DMA)



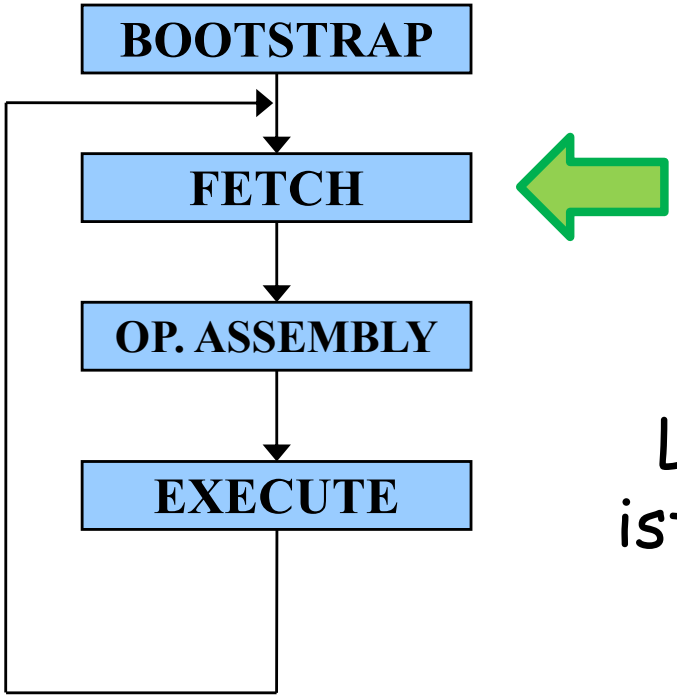
- **DMA (Direct Memory Access)**: dispositivo che opera trasferimenti dati da/verso la memoria **per conto della CPU**
- Utile per trasferimenti di **grossi blocchi di dati** (es. file di grandi dimensioni)
- La CPU gestisce un minor numero di interrupt

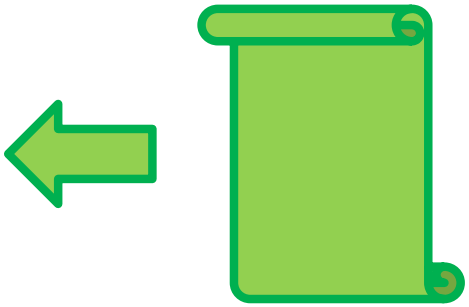
Interrupt

- Le interrupt permettono di alternare la CPU tra l'esecuzione dei programmi dell'utente e la gestione dell'I/O
  - Quando i dispositivi di I/O sollevano interrupt, la CPU si dedica (temporaneamente) alla loro gestione

Le interrupt sono alla base della multiprogrammazione e del time-sharing, aumentano l'**utilizzo delle risorse (efficienza)**

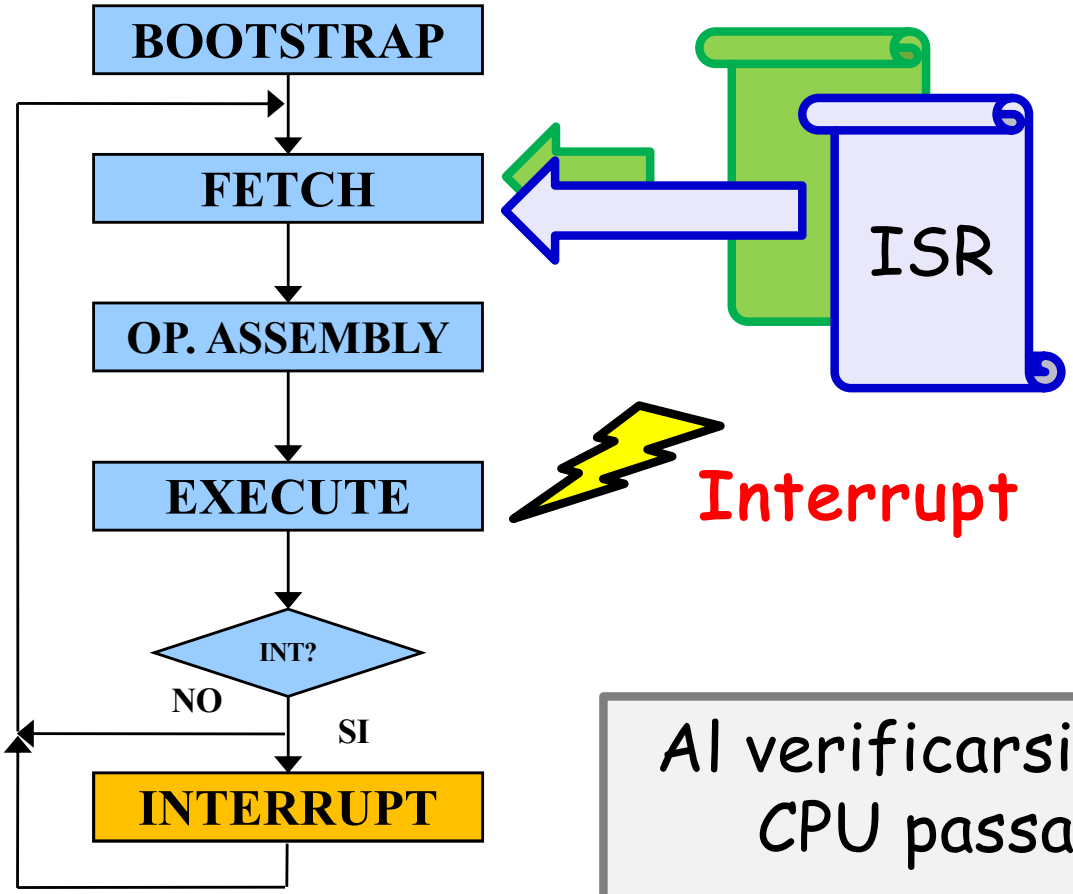
Ciclo della CPU senza gestione degli Interrupt





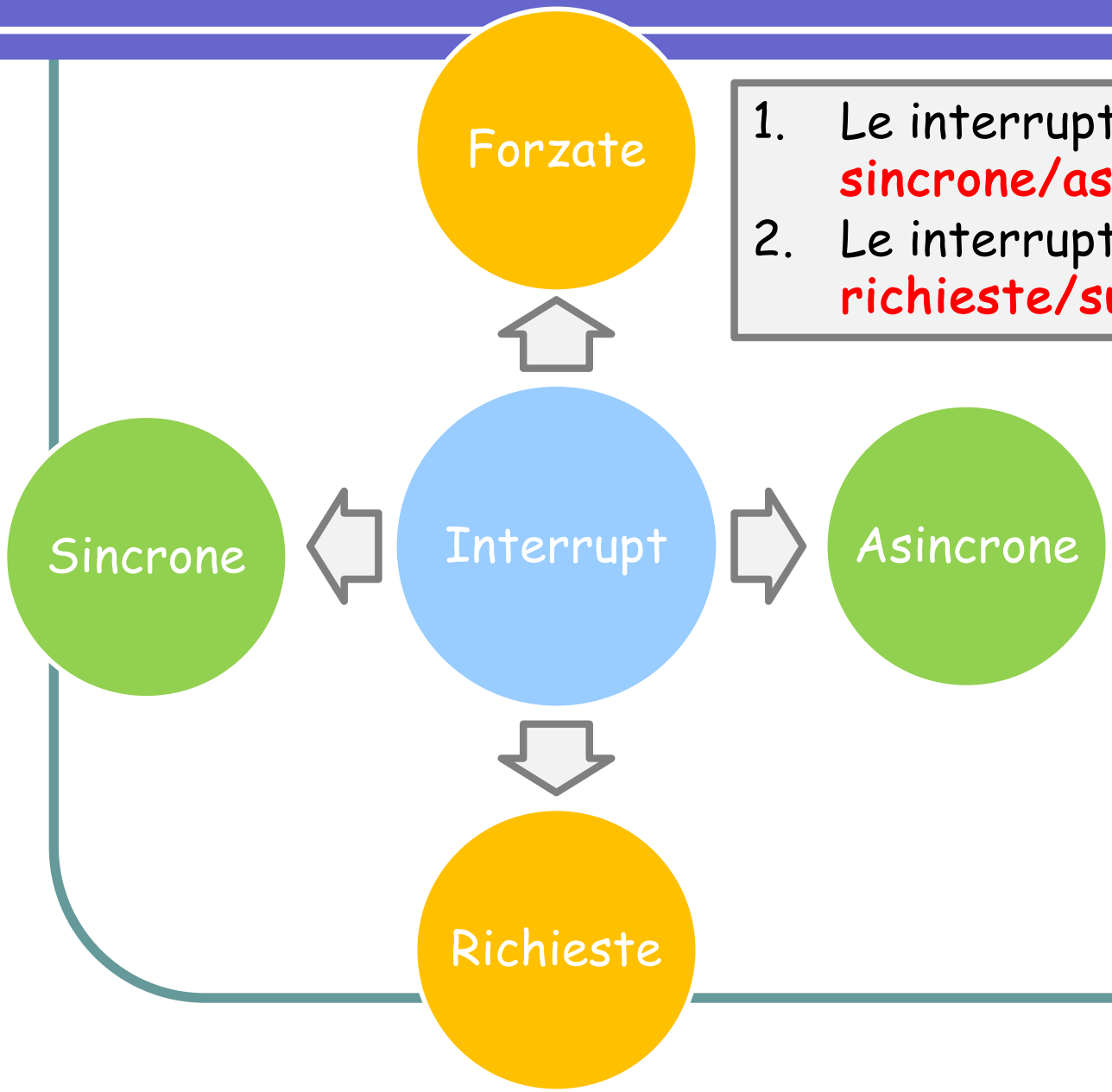
La CPU esegue le istruzioni di un solo programma

Ciclo della CPU con gestione degli Interrupt



Al verificarsi di un interrupt, la CPU passa ad eseguire un programma di gestione (Interrupt Service Routine, ISR)

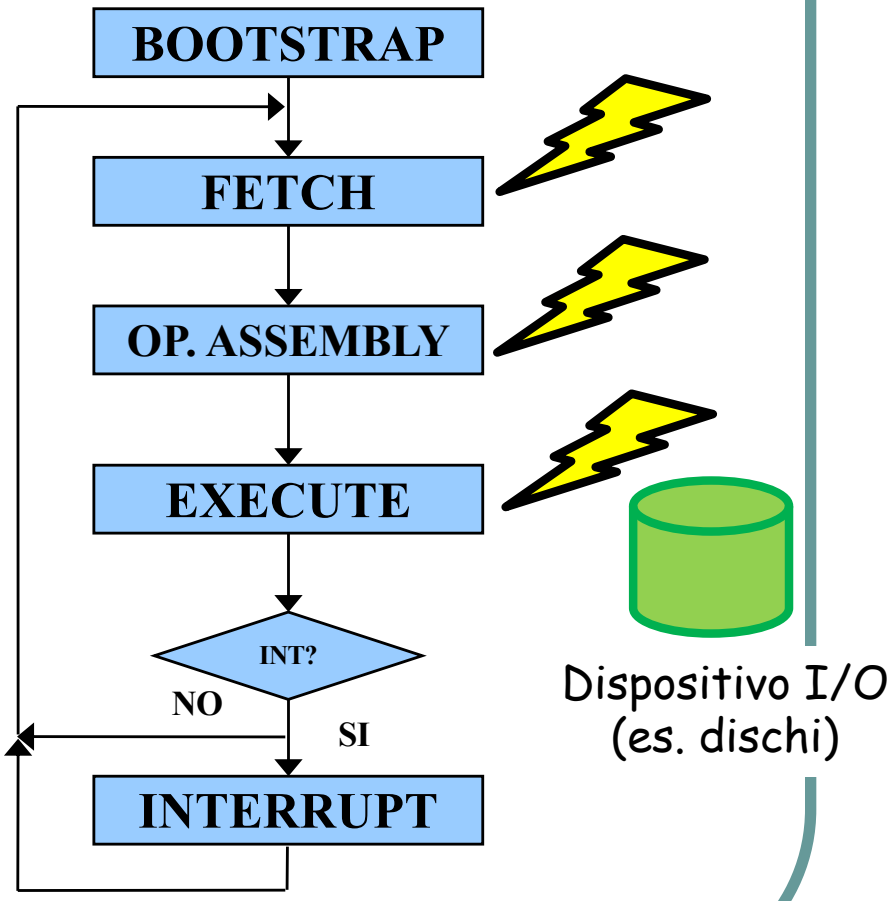
Tipologie di interrupt



1. Le interrupt possono essere **sincrone/asincrone** con il programma
2. Le interrupt possono essere **richieste/subite** dal programma

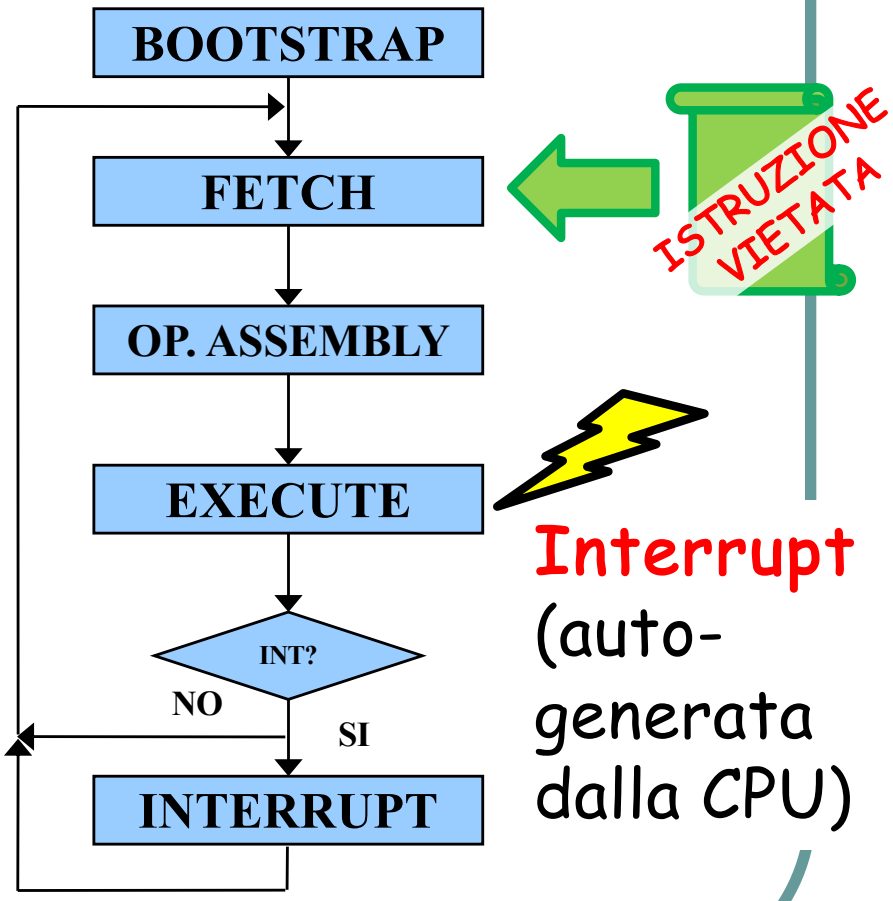
Interrupt asincroni

- Interruzioni **asincrone** (spesso semplicemente "*interrupts*")
  - Richieste di "attenzione" dai dispositivi di I/O
  - Possono verificarsi in qualunque momento durante l'esecuzione di un programma



Interrupt sincroni

- Interruzioni **sincrone** (o "*traps*", "*exceptions*")
  - Si verificano con **specifici eventi del programma in esecuzione**
  - "Sincrone" rispetto a ciò che avviene nel programma



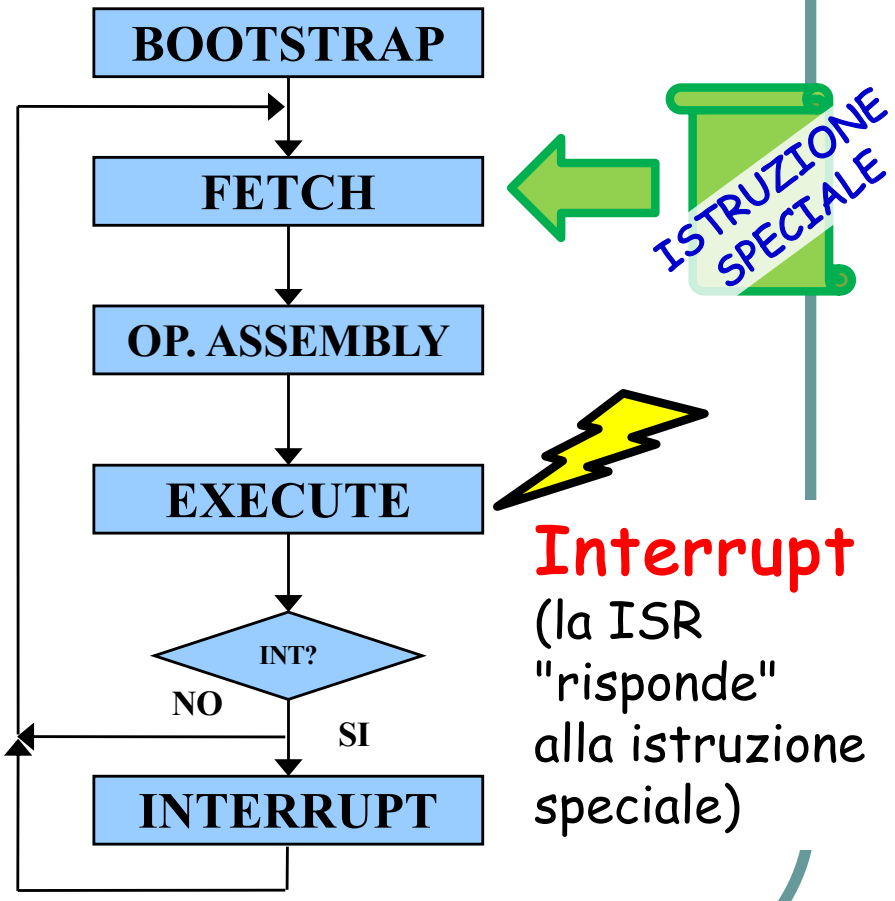
Interrupt sincroni - esempi

- Accessi in **indirizzi di memoria "illeciti"** o non allineati
  - es., il classico errore di uso di puntatori NULL
  - es., uso di indirizzi dispari (**non allineati**) in alcune architetture (es., ARM, M68000)
- Condizioni "limite" in **operazioni aritmetiche**
  - es., "divisione per zero", overflow, NaN, etc.

Il SO risponde in molti casi "uccidendo" il programma

Interrupt sincroni richiesti dal programma

- In alcuni casi, le interruzioni possono essere **volutamente richieste dal programma**
  - Sono **sincrone**
  - Si utilizza il meccanismo delle interrupt come canale per **innescare l'esecuzione del sistema operativo**



Interrupt (la ISR "risponde" alla istruzione speciale)

Interrupt sincroni - esempi

- **Breakpoint**:
  - Usate dai tool di debug
  - **"ferma" l'esecuzione** di un programma quando **arriva ad una certa istruzione**
  - si **inserisce una istruzione speciale** nel programma
  - l'utente può studiare il contenuto delle variabili del programma

```
mov
mov
add
INT 3
mov   ...
call  ...
```

La CPU interromperà il programma qui

Interrupt sincroni - esempi

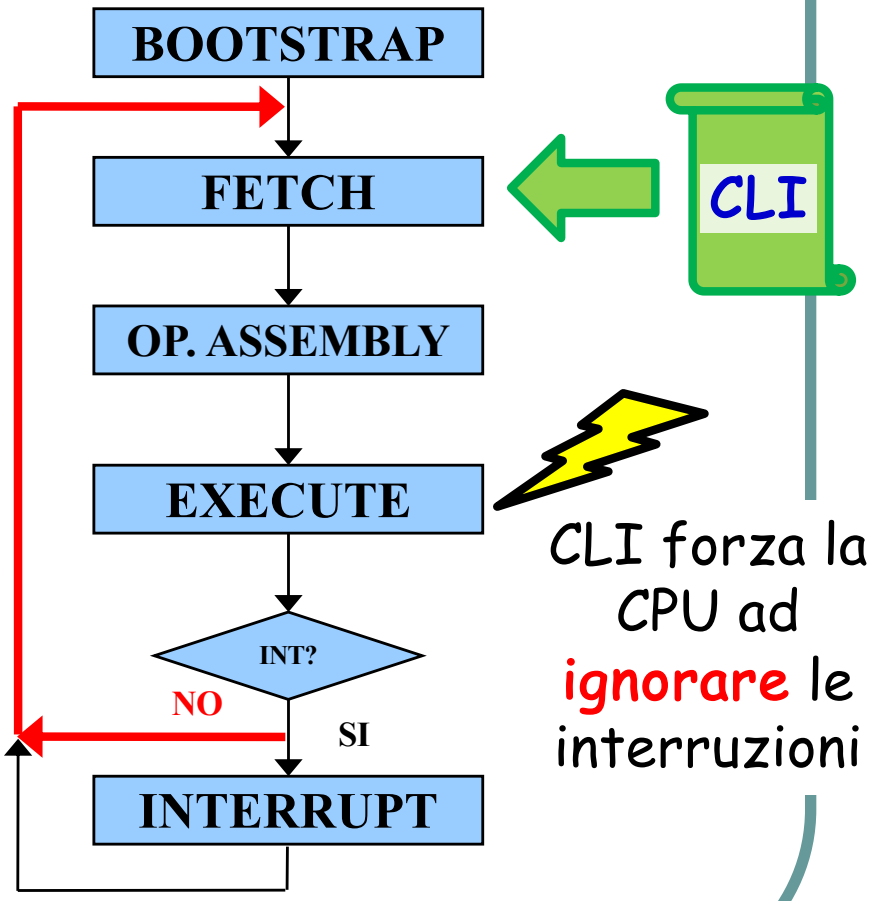
- **Tracing**:
  - la CPU interrompe il programma dopo **ogni singola istruzione**
  - questa modalità si attiva tipicamente scrivendo un bit nello status register
- **Supervisor Call**:
  - Il programma innesca una ISR, per **chiedere al SO** di svolgere delle operazioni speciali (es. su hardware)
  - Saranno discusse in una prossima lezione

Tipi di interrupt

Tipo	Sincrone / asincrone	User request / forzate
I/O device request	Asincrone	Forzate
Invoke operating system	Sincrone	User request
Instruction tracing	Sincrone	User request
Breakpoint	Sincrone	User request
Integer arithmetic overflow	Sincrone	Forzate
Floating-point arithmetic overflow or underflow	Sincrone	Forzate
Page fault	Sincrone	Forzate
Misaligned memory access	Sincrone	Forzate
Memory protection violations	Sincrone	Forzate
Using undefined instructions	Sincrone	Forzate
Hardware malfunctions (machine check exceptions)	Asincrone	Forzate
Power failure	Asincrone	Forzate

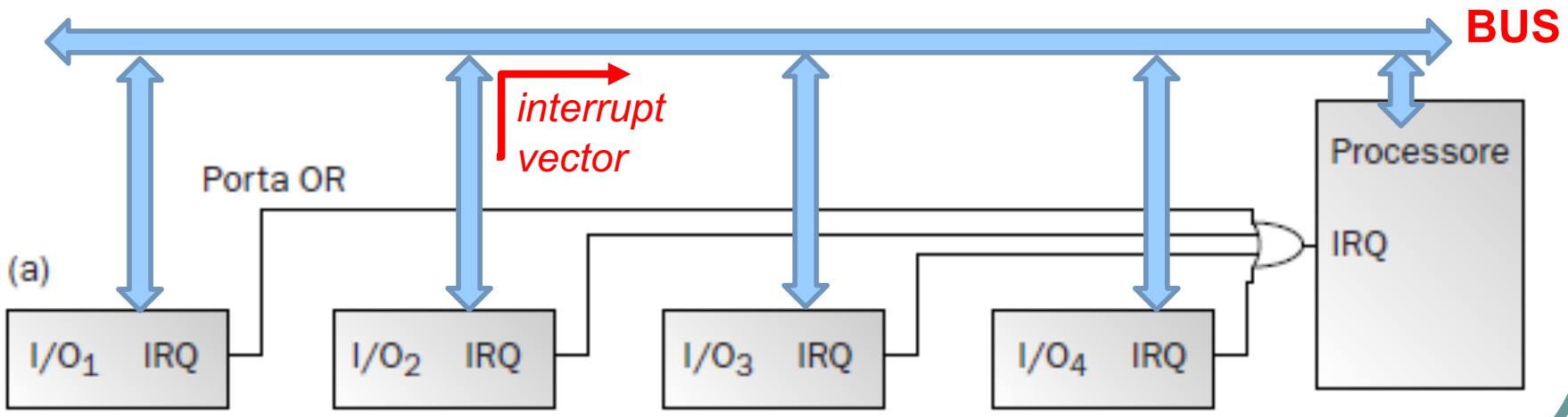
Set/Clear Interrupt

- Le interruzioni possono essere abilitate o disabilitate per mezzo di apposite istruzioni macchina
- Set Interrupt - STI
- Clear Interrupt - CLI

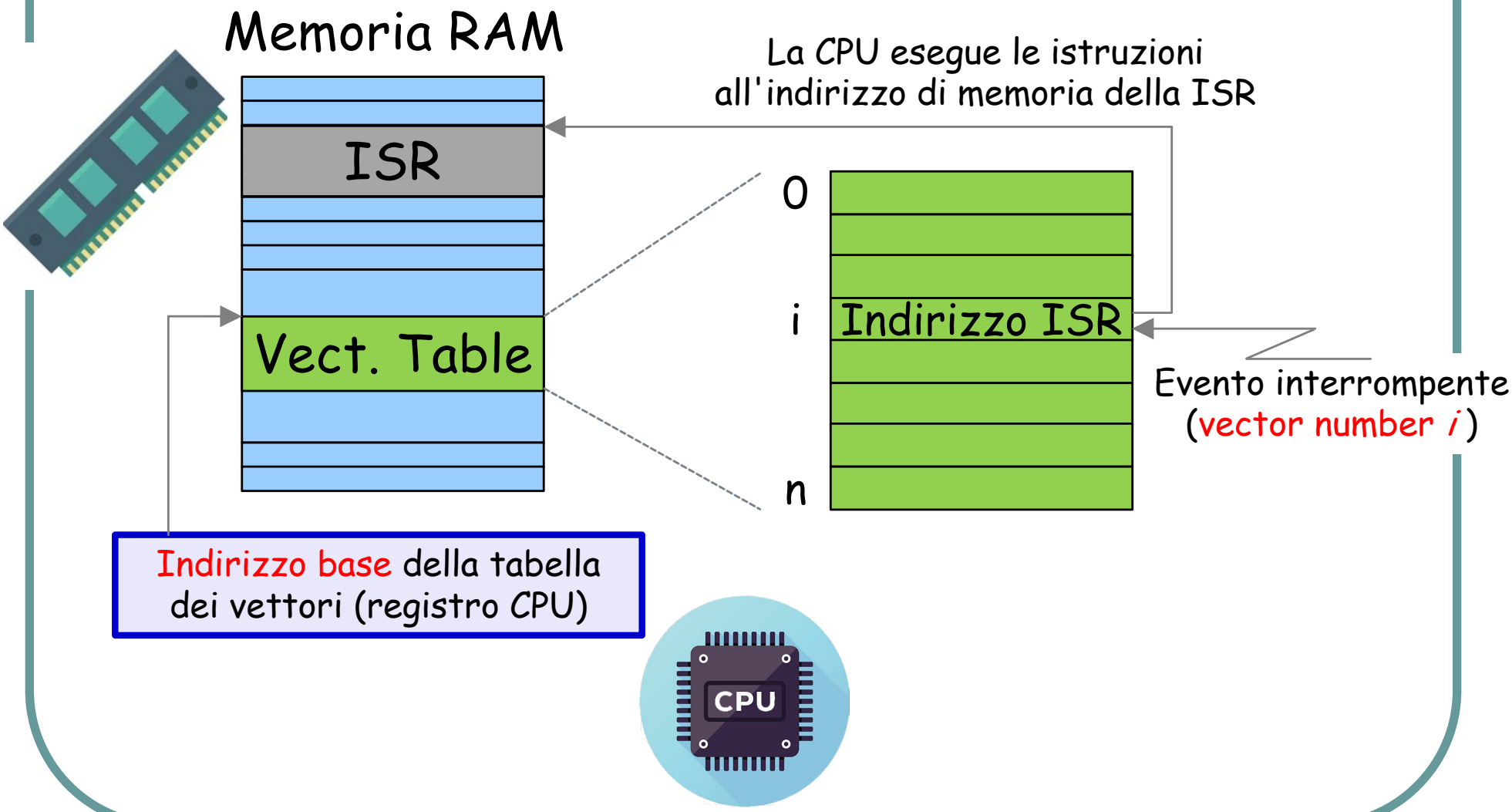


Interrupt vettorizzati

- Più periferiche possono essere collegate alla stessa linea
- La CPU identifica il dispositivo tramite un **identificativo** (**vector number**), fornito dal dispositivo
- L'identificativo indica la ISR da eseguire da una tabella (**interrupt vector table**)



Interrupt vettorizzati



Vettore delle interruzioni in x86

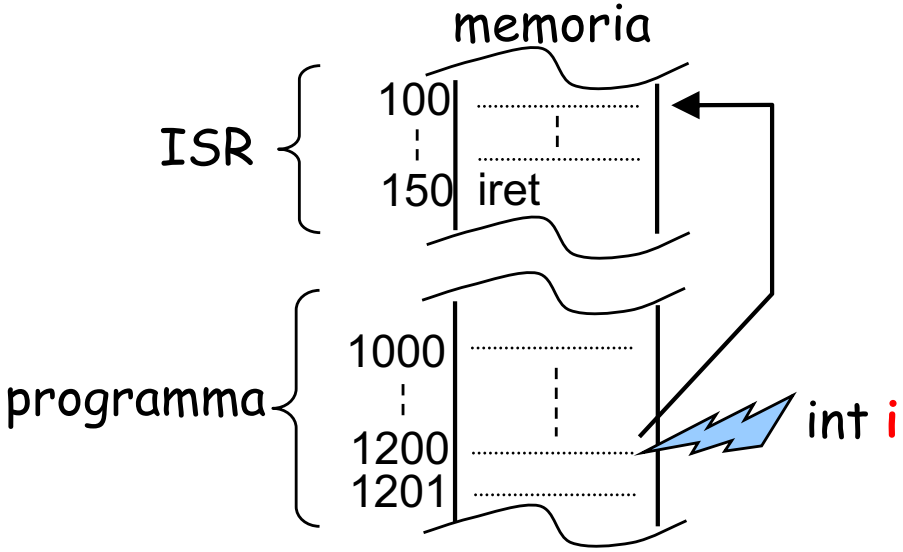
Vector No.	Mnemonic	Description	Type	Error Code	Source
0	#DE	Divide Error	Fault	No	DIV and IDIV instructions.
1	#DB	RESERVED	Fault/ Trap	No	For Intel use only.
2	—	NMI Interrupt	Interrupt	No	Nonmaskable external interrupt.
3	#BP	Breakpoint	Trap	No	INT 3 instruction.
4	#OF	Overflow	Trap	No	INTO instruction.
5	#BR	BOUND Range Exceeded	Fault	No	BOUND instruction.
6	#UD	Invalid Opcode (Undefined Opcode)	Fault	No	UD2 instruction or reserved opcode.[1]
7	#NM	Device Not Available (No Math Coprocessor)	Fault	No	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Abort	Yes (zero)	Any instruction that can generate an exception, an NMI, or an INTR.

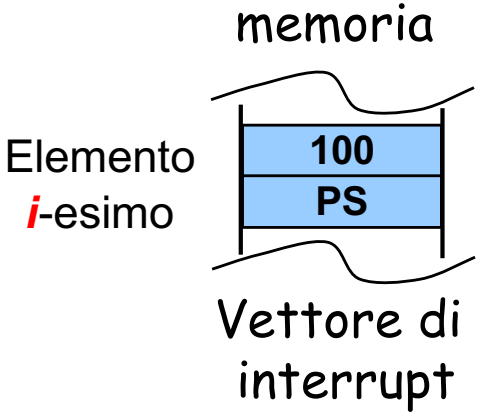
Gestione degli Interrupt (dettagli)

1. Un segnale di interrupt request (INT) viene inviato alla CPU
2. La CPU finisce l'esecuzione dell'istruzione corrente
3. La CPU verifica la presenza del segnale INT ed invia un segnale di conferma (ACK) al dispositivo
4. La CPU salva sullo stack del sistema operativo le informazioni necessarie a riprendere il programma interrotto

   I registri critici da salvare sono:
   - Program Counter (PC)
   - Stack Pointer (SP)
   - Registro di Stato (o Program Status, PS)

Gestione degli Interrupt (dettagli)

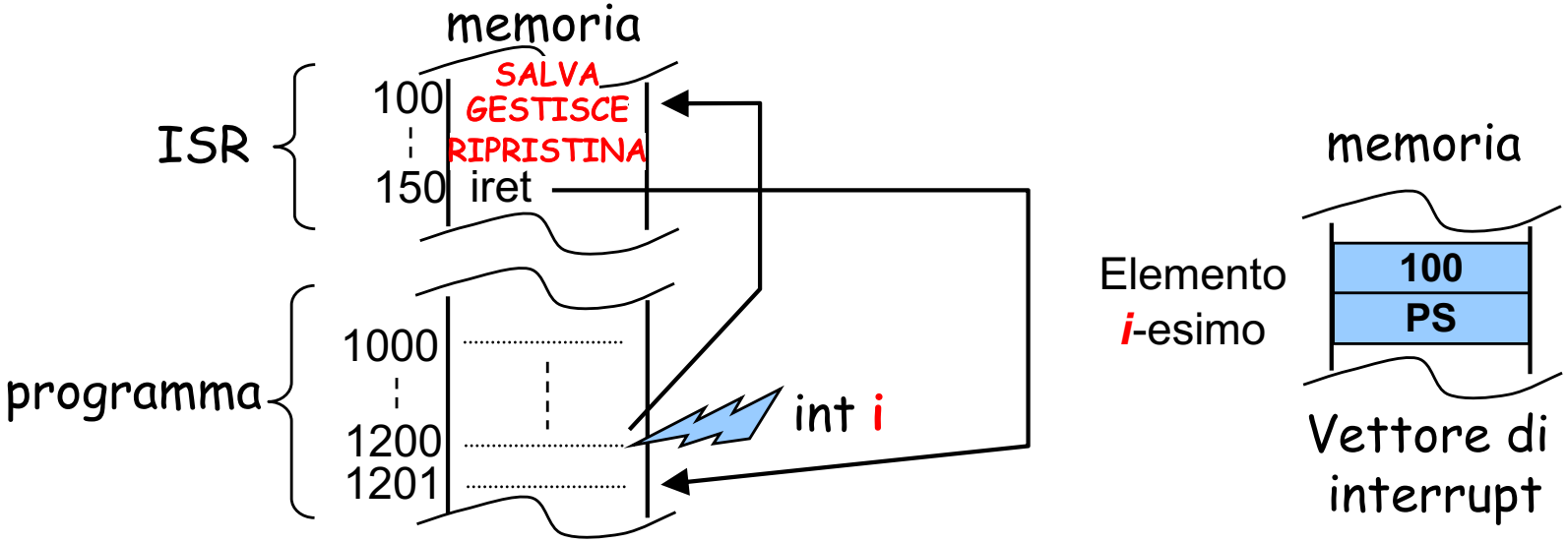




5. La **CPU** seleziona la ISR tramite il **vettore di interrupt**
6. La **CPU** carica dal vettore di interrupt:
   - il registro PC, con l'indirizzo iniziale della ISR
   - il registro PS (attiva anche la **modalità supervisore** della CPU)

LE OPERAZIONI FINO A QUI SONO SVOLTE IN **HARDWARE**

Gestione degli Interrupt (dettagli)



7. La **ISR** salvata lo **stato del processore** su uno stack (es., i registri-dato non già salvati dalla CPU)
8. La **ISR** gestisce l'interrupt
   - Interazione con il controller del dispositivo
9. La **ISR** ripristina lo stato del processore (op. inversa alla 7)
10. Ritorno del controllo al **processo interrotto** (istruzione **iret**)
    - Ripristino dei registri PC e PS salvati sullo stack

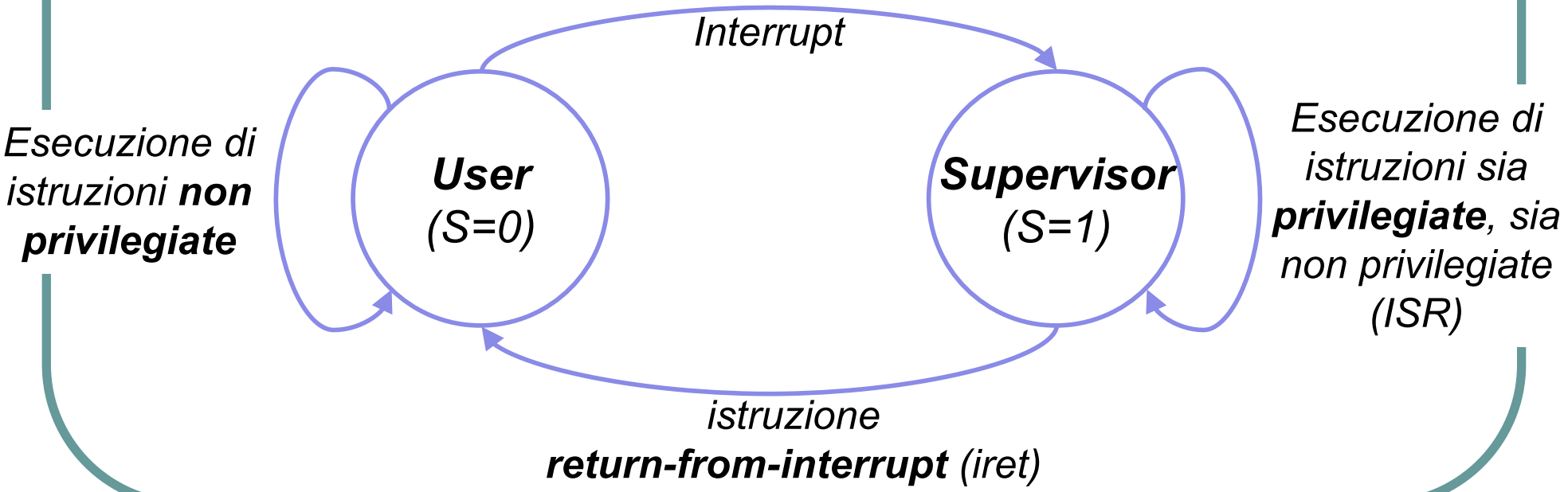
LE OPERAZIONI 7, 8 E 9 SONO SVOLTE IN **SOFTWARE**

Protezione delle risorse hardware

- I processori moderni presentano almeno due stati di funzionamento:
  - User mode (o "non privilegiato")
  - Kernel mode (o "supervisore" o "privilegiato")
- La **modalità supervisore** abilita le istruzioni per l'accesso alle risorse fisiche (*istruzioni privilegiate*)
  - Ad esempio, le **ISR** eseguono in modo supervisore

Protezione delle risorse hardware

- Lo stato supervisore/utente della CPU è tipicamente indicato nel **registro di stato PS** da un **bit (S=1 oppure S=0)**
- La CPU pone automaticamente **S=1** all'avvio della ISR
- Il bit **S=0** (insieme al resto del registro PS) è ripristinato al termine della ISR (istruzione **iret**)

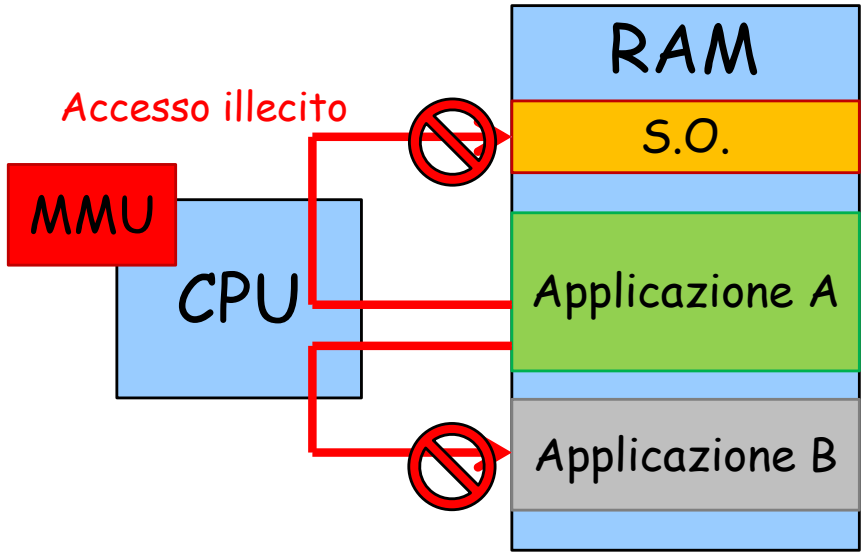


Esempi di istruzioni privilegiate

- Istruzioni **STI** e **CLI** per l'abilitazione e disabilitazione delle interruzioni
- Modifica dei **registri per la gestione degli interrupt** (es. indirizzo base interrupt vector table)
- Istruzioni per la **gestione della memoria**
- Istruzioni per **I/O port-mapped**
- Istruzioni per la **modifica del registro di stato PS**
  - PS contiene anche il bit di stato utente o supervisore, e quindi non deve essere modificabile in modo utente

Protezione di memoria

- Nei sistemi multiprogrammati e multiutente, più applicazioni **condividono la memoria**
- La CPU (**Memory Management Unit, MMU**) protegge:
  - codice e dati del SO dalle applicazioni
  - codice e dati di una applicazione dalle altre applicazioni (es. privacy degli utenti)

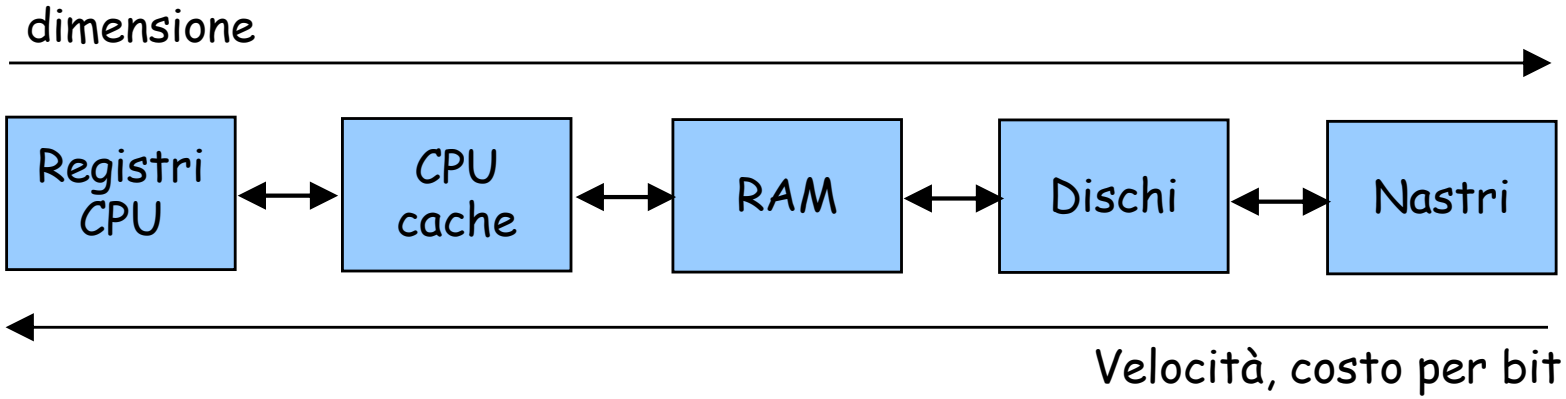


Tipologie di memorie

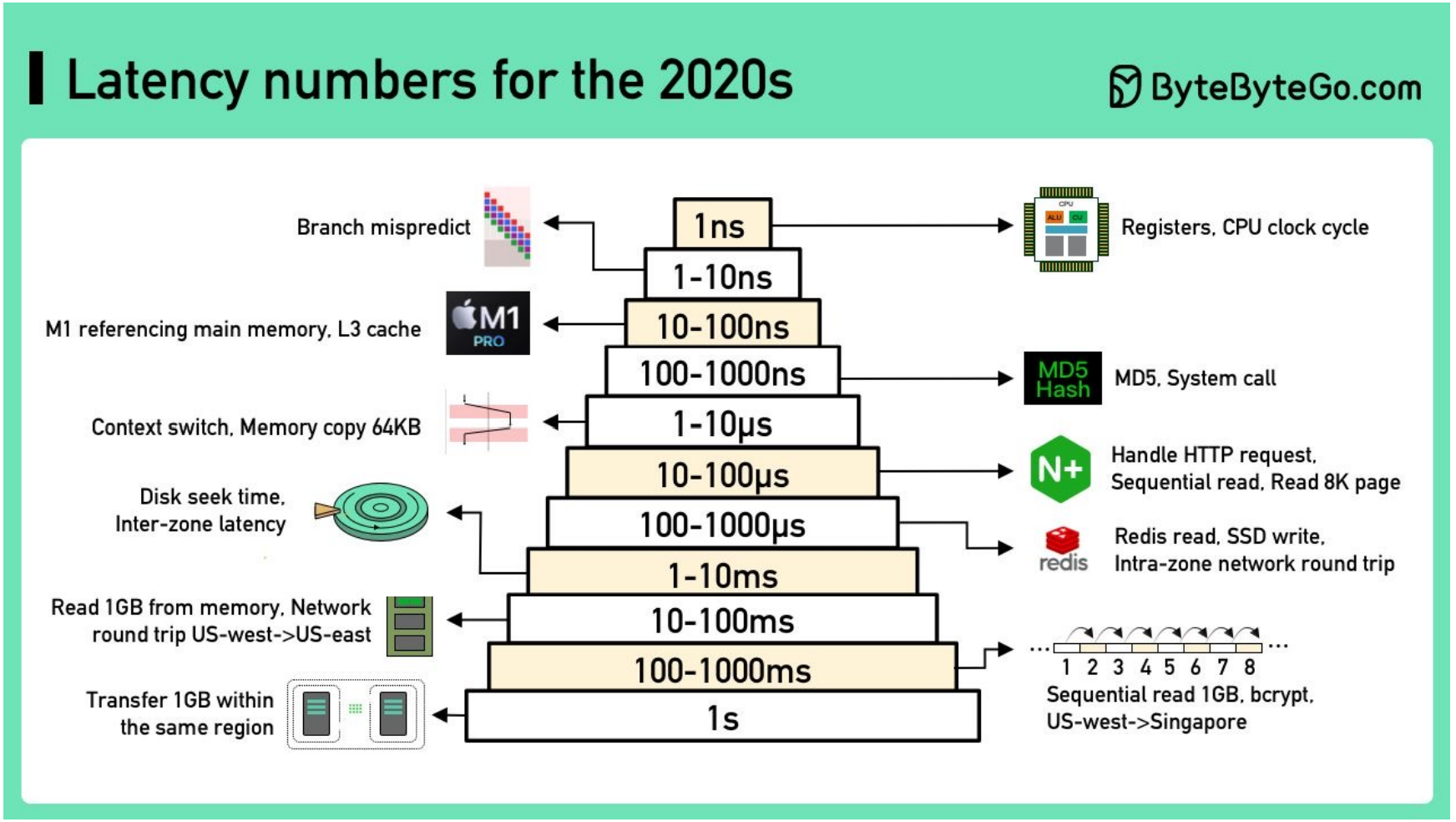
- **Memoria Centrale** – spazio di memorizzazione volatile che può essere acceduto direttamente dalla CPU
- **Memoria Secondaria** – memoria non volatile con un'alta capacità di memorizzazione
  - Dischi Magnetici
  - Memorie a stato solido
  - Nastri Magnetici

Gerarchia di memoria

- Le memorie possono essere organizzate in una **gerarchia** secondo 3 caratteristiche
  - Velocità
  - Costo
  - Dimensione



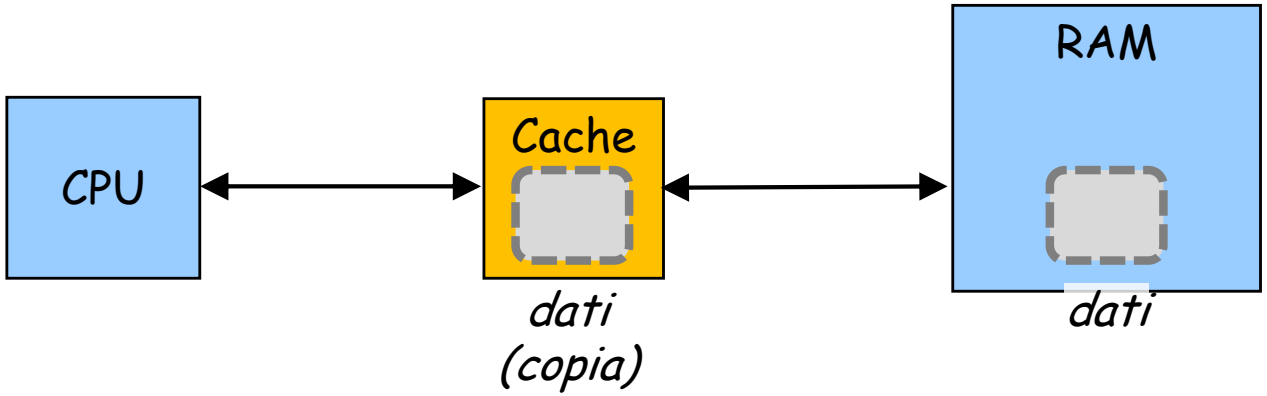
Prestazioni dell'I/O



Latency Numbers Every Programmer Should Know: https://colin-scott.github.io/personal_website/research/interactive_latency.html
Do you know how much your computer can do in a second?: https://computers-are-fast.github.io/

Caching

- Il caching **conserva nelle memorie veloci** una copia (di un sottoinsieme) dei dati delle **memorie lente**
  - Se il dato è presente in cache (**cache hit**), vi si accede velocemente
  - Altrimenti (**cache miss**), si accede alla memoria centrale



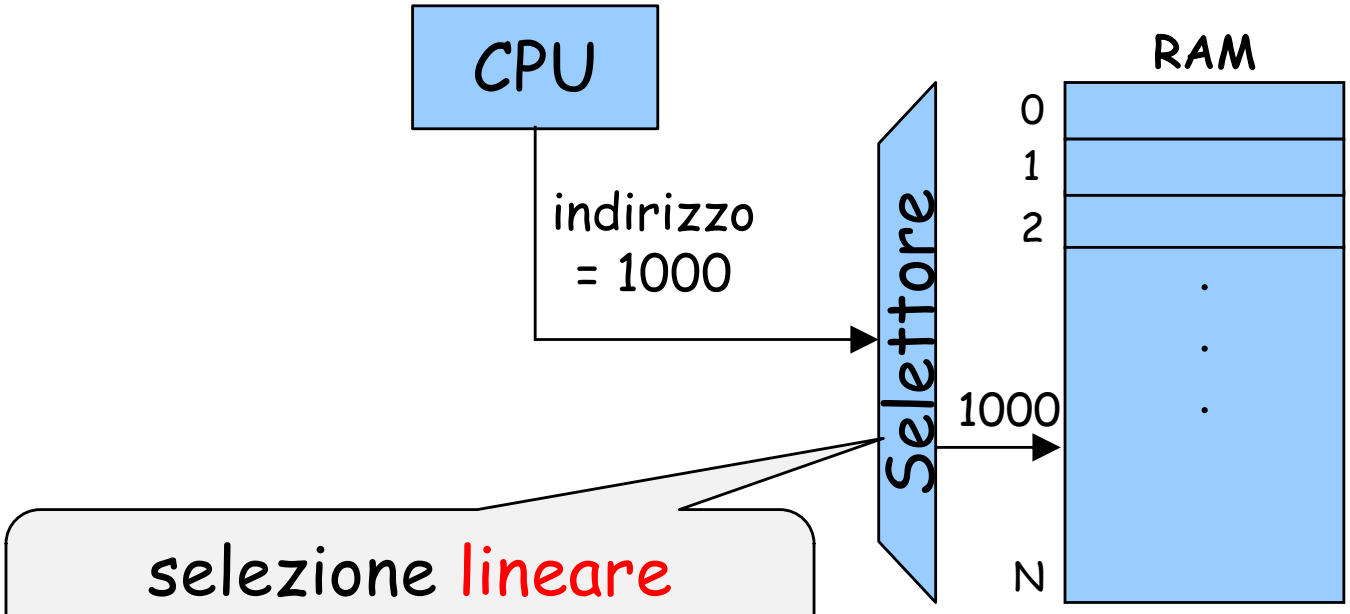
Il caching è efficace grazie ai principi di località:

Località spaziale: i programmi tendono ad accedere a dati vicini

Località temporale: i programmi tendono ad accedere più volte agli stessi dati

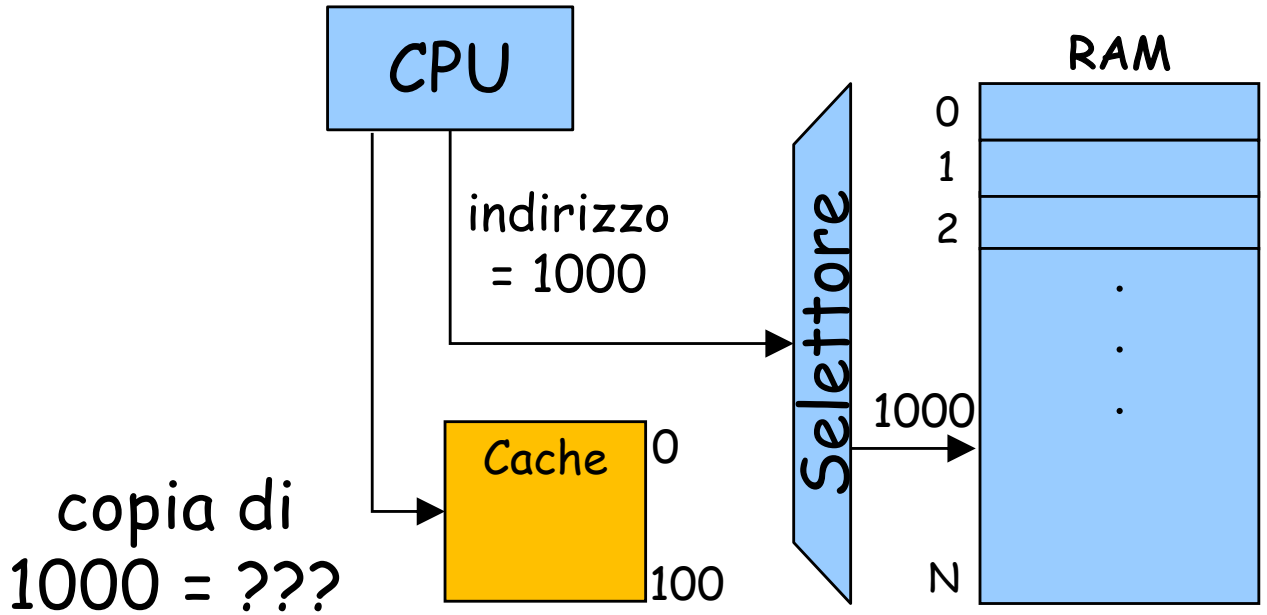
Selezione lineare

- La RAM è una memoria capiente (decine di GB), il suo spazio di indirizzamento è un insieme di valori molto ampio
- La CPU accede alla RAM in base alla posizione del codice/dati, indicandone l'indirizzo (selezione lineare)



Selezione associativa

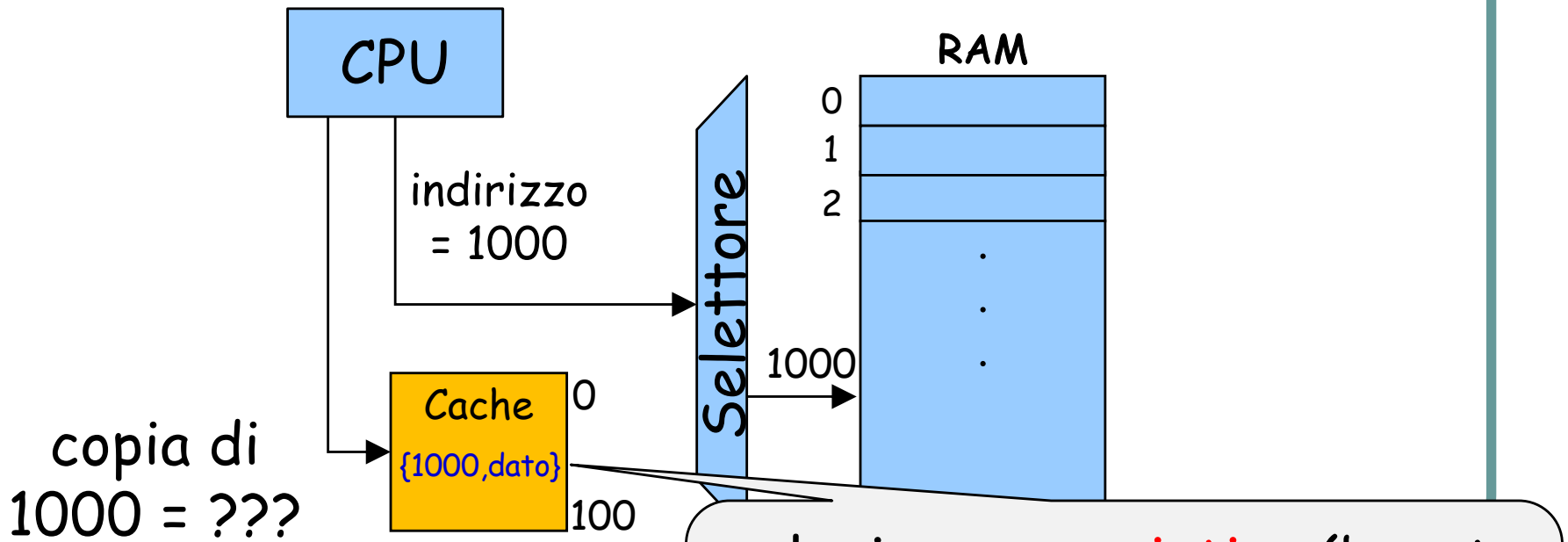
- La cache è una memoria molto più piccola della RAM
- Non è fattibile usare l'indirizzo come "posizione" per trovare la copia del dato



Selezione associativa

La cache risolve tramite selezione associativa, con cui:

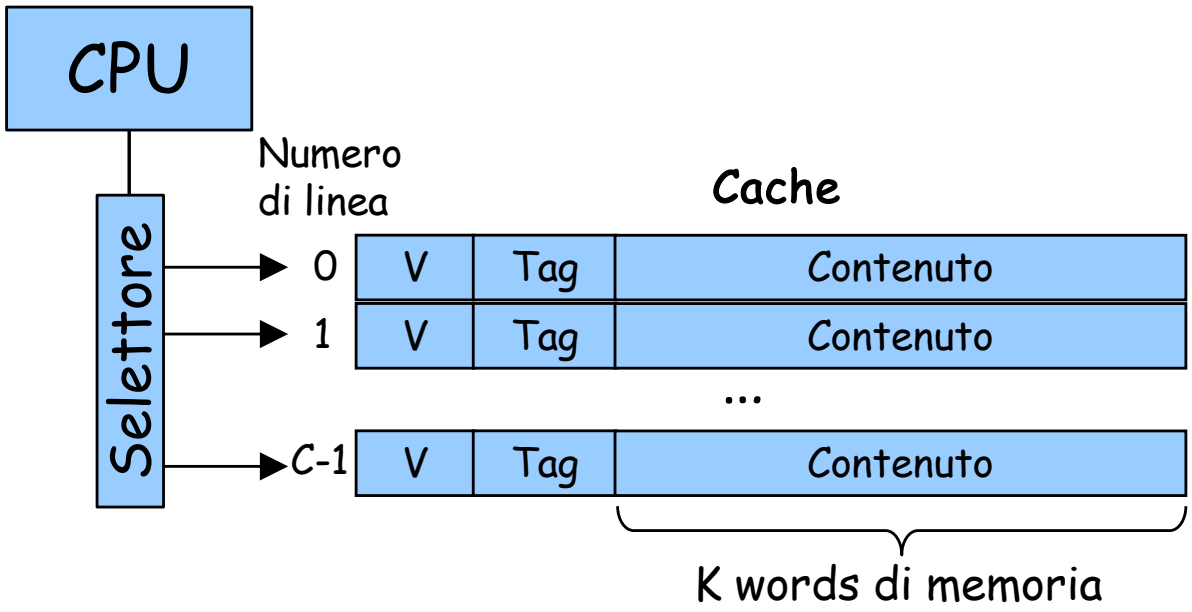
- si memorizza la coppia {indirizzo, dato} in posizione arbitraria
- per trovare il dato, si fa una ricerca della coppia che contiene l'indirizzo indicato dalla CPU



selezione associativa (basata sul contenuto), fa una ricerca della coppia {1000, dato}

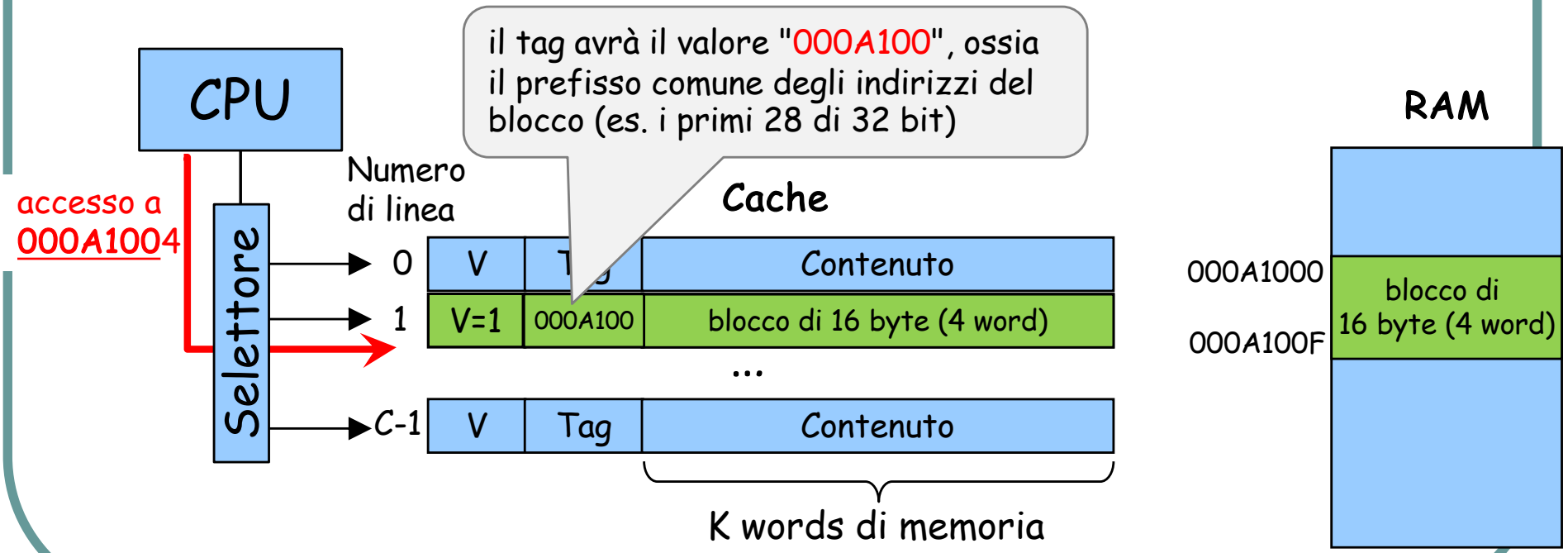
Caching (dettagli)

- **Contenuto**: copia di un **blocco di dati contigui** della memoria centrale
- **V**: bit di validità, indica se la linea di cache è occupata
- **Tag**: identifica il contenuto (prefisso comune degli indirizzi del blocco)
- L'accesso alla cache, con verifica del tag e validità del contenuto, avviene in **parallelo** (***selezione associativa***)



Caching (dettagli)

- **Contenuto**: copia di un **blocco di dati contigui** della memoria centrale
- **V**: bit di validità, indica se la linea di cache è occupata
- **Tag**: identifica il contenuto (prefisso comune degli indirizzi del blocco)
- L'accesso alla cache, con verifica del tag e validità del contenuto, avviene in **parallelo** (***selezione associativa***)



Caching

- Un concetto che si applica a **vari livelli**
  - Cache della memoria centrale nella CPU
  - Cache del disco in memoria centrale
  - Cache di filesystem *distribuiti* in filesystem *locali*
- Richiede una politica di gestione della cache
  - **Dimensione** della cache e dei blocchi
  - Una **funzione di mapping** del blocco nella cache
  - Algoritmi di **sostituzione**
- Problema della **coerenza**

Quiz

1. Quali di queste interruzioni sono sincrone? (selezionare più di una)

- [ ] Accessi di memoria illeciti
- [ ] Istruzione macchina non riconosciuta dalla CPU
- [ ] Interruzione dai dispositivi di I/O
- [ ] Divisione per zero

2. Nella gestione delle interruzioni, quali di queste operazioni sono effettuate in hardware? (selezionare più di una)

- [ ] Salvataggio dei registri-dato del processore
- [ ] Salvataggio del registro Program Counter (PC)
- [ ] Salvataggio del registro di stato (o Program Status, PS)
- [ ] Cambio del livello di privilegio di esecuzione del processore

https://forms.office.com/r/rvjdafpkjq

Quiz

3. Quale di queste modalità di I/O ha le prestazioni peggiori? (selezionare solo una)

- ○ Interrupt-based
- ○ Polling ("I/O programmato")
- ○ DMA ("Direct Memory Access")

https://forms.office.com/r/rvjdafpkjq