

# **Input-Output**

## **Periferiche**

## **Interrupt**

## **DMA**

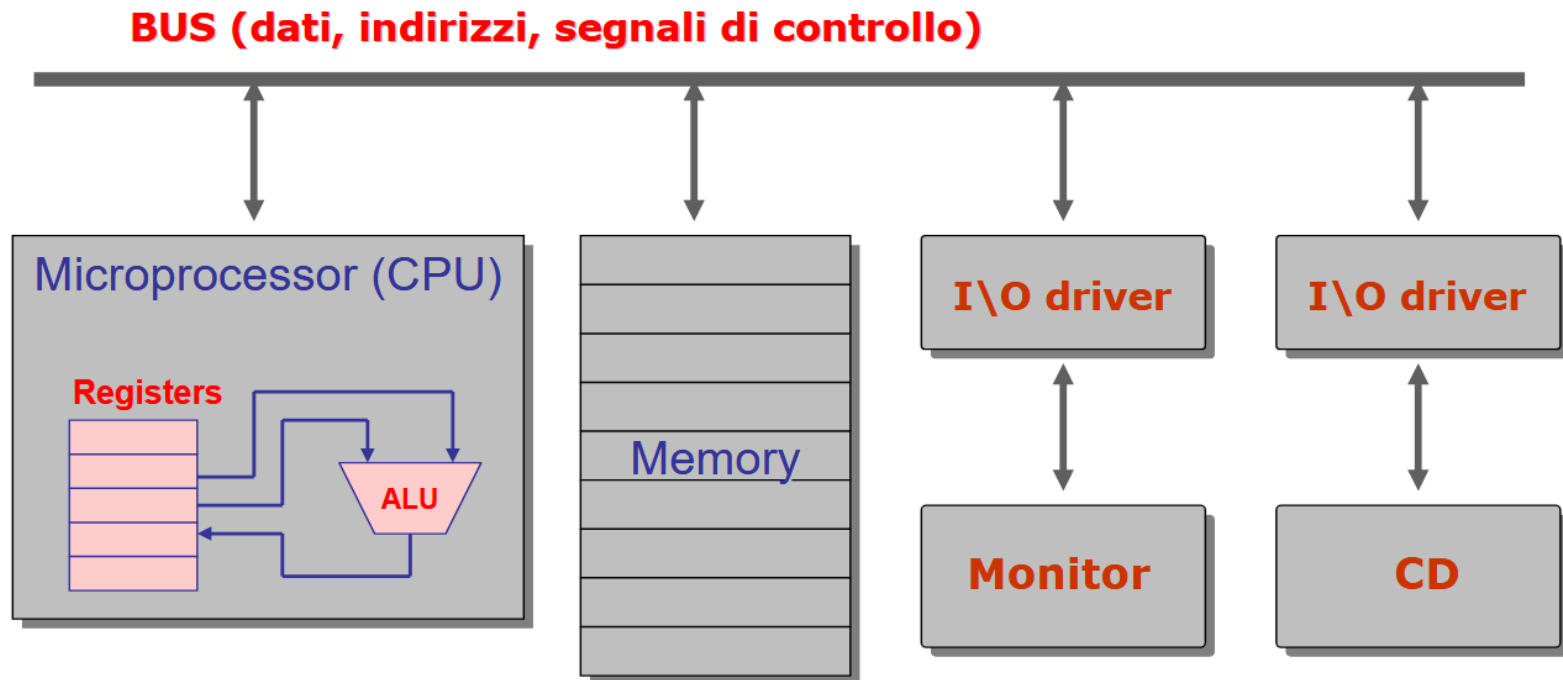
Claudia Raibulet  
[claudia.raibulet@unimib.it](mailto:claudia.raibulet@unimib.it)

# Tecniche di gestione ingresso-uscita

- Input-output – Periferiche
- Bus di sistema
- Periferiche mappate in memoria
- Controllo di programma
- Banda passante e latenza
- Input-Output gestito direttamente dal programma
- Input-Output con interruzioni
- Input-Output con DMA (Direct Memory Access)
- Periferiche in SPIM

- I/O – insieme di architetture e dispositivi per il trasferimento di informazioni da e verso l'elaboratore
- Dispositivi eterogenei per:
  - Velocità di trasferimento
  - Latenze
  - Sincronizzazione
  - Modalità di interazione (sia con l'uomo sia con la macchina)

- Esistono vari tipi di bus nei computer di oggi
- In questo corso: bus comune, di sistema che collega la CPU sia con la memoria sia con le periferiche



- Il bus di sistema e' composto da:
  - **Bus di dati**: le linee per trasferire **dati** e **istruzioni** da/verso dispositivi
  - **Bus di controllo**: trasporta **informazioni** per la definizione delle **operazioni da compiere** e per la **sincronizzazione tra i dispositivi**
  - **Bus degli indirizzi**: la CPU trasmette gli **indirizzi** di memoria o di periferica che identificano i dati da leggere/scrivere dalla memoria/periferiche
- ***Tutte le unità dell'elaboratore sono connesse al bus***

- **Vantaggi:**
  - elevata flessibilità
  - semplicità
  - basso costo
- **Svantaggi:**
  - gestione complessa del canale condiviso

- Dispositivi per I/O di informazioni
- Collegate alla CPU tramite il bus di sistema e/o interfacce
- Le interfacce sono standardizzate per la comunicazione
- Le interfacce hanno
  - una **componente hardware** (e.g., il **controllore** della periferica)
  - una **componente software** (e.g., **driver**)
- Struttura di un'interfaccia (e.g., tramite i **registri della periferica**):
  - Contiene registri di dati (oppure buffer)
  - Ha associato registri di stato

## Periferiche mappate in memoria

- **Una parte della memoria riservata al sistema viene usata per la comunicazione con le periferiche**
- Ogni periferica ha uno spazio dedicato nella memoria; a ogni periferica viene assegnato un identificatore unico (i.e., indirizzo unico)
- **I registri (dell'interfaccia della periferica) sono mappati in memoria**
- ***Tali registri NON sono accessibili da un programma utente***
- I programmi utente devono passare dal SO per accedere a questi registri riservati
- L'accesso alla periferica e' simile all'accesso alla memoria (e.g., lw, sw) dal punto di vista della CPU
- **Osservazione:**
  - non tutte le architetture degli elaboratori usano questa soluzione



# Registri di interfaccia della periferica

- **Registro di stato della periferica**
  - Rappresenta lo stato della periferica
  - Viene letto dalla CPU
  - **Esempi di stato:**
    - periferica pronta per ricevere dati
    - dati richiesti disponibili per il trasferimento
- **Registro dei dati della periferica**
  - Rappresenta i dati di input o di output in base al tipo della periferica

## Passi di I/O (versione molto semplificata)

- CPU interroga lo stato della periferica
- La periferica restituisce il suo stato
- Se la periferica e' pronta a trasmettere/ricevere dati, la CPU richiede il trasferimento dei dati
- La CPU invia o riceve i dati

## Tecniche di gestione I/O

- I/O gestito da programma (**Programmed I/O**)
- I/O guidato da interrupt (**Interrupt Driven I/O**)
- Accesso Diretto alla Memoria (**Direct Memory Access - DMA**)

## I/O gestito da programma

- I/O gestito dal controllo di programma
- La periferica ha un ruolo passivo
- La CPU si occupa sia del controllo sia del trasferimento dati
- La CPU predispone il controllore della periferica all'esecuzione dell'I/O
- La CPU si ferma e interroga il registro di stato della periferica in attesa che sia pronta (e.g., **ready bit** assume un determinato valore)
- **Vantaggio:** risposta veloce al ready bit
- **Svantaggio:** la CPU bloccata in stato di **busy waiting**

## I/O guidato da interrupt

- Interrupt – evento asincrono che genera l'interruzione del normale funzionamento del processore
- La periferica segnala alla CPU di aver bisogno di attenzione mediante un **segnale sul bus di controllo**
- La periferica avvisa la CPU attivando un segnale **interrupt request**
- Quando il processore se ne accorge (in una **fase di fetch**) informa la periferica con un segnale **interrupt acknowledge**
- La CPU interrompe l'esecuzione del programma corrente (salvando il contesto dell'esecuzione del programma per poter riprendere la sua esecuzione) ed esegue la procedura di risposta all'interrupt
- Terminata l'esecuzione della **procedura di interrupt**, la CPU riprende l'esecuzione del programma interrotto
- Il programma utente continua la sua esecuzione

## I/O guidato da interrupt

- **Vantaggio:** la CPU non fa piu' busy waiting come per I/O gestito da programma
- **Svantaggio:** la CPU deve comunque gestire le operazioni di trasferimento
- Per evitare l'intervento della CPU nella fase di trasferimento dati e' stato introdotto il protocollo di trasferimento  
**DMA – Direct Memory Access**

## DMA – accesso diretto alla memoria

- Usato quando si trasferiscono velocemente grandi quantità di dati
- Con DMA la periferica diventa autonoma nell'accesso alla memoria
- E' la periferica che gestisce i trasferimenti; la CPU non interviene nel trasferimento dei dati
- Necessita' **2 registri in più per ogni periferica oltre al registro di stato e registro dei dati:**
  - Un registro che indichi **l'indirizzo di memoria da/dove trasferire i dati**
  - Un registro che indichi la **quantità dei dati da trasferire**
- Anche questi 2 registri aggiuntivi sono mappati in memoria
- Alla fine del trasferimento la periferica invia un interrupt alla CPU per segnalare il completamento del trasferimento

## Prestazioni – Banda passante e latenza

- Misure che permettono di valutare l'efficienza della gestione delle operazioni di I/O
- **Banda passante:**
  - rappresenta la **quantità dei dati** che si può trasferire **per unità di tempo**;
  - rappresenta una **misura di flusso**.
- **Latenza:**
  - rappresenta il tempo che intercorre tra l'istante in cui una periferica è pronta per il trasferimento e l'istante in cui il dato viene trasferito;
  - è una **misura di tempo**.



## Prestazioni – I/O gestito da programma

- **Banda passante:** alta perche' la CPU trasferira subito il dato (e.g., molti dati per l'unita' di tempo) e la gestione della periferica richiede poche istruzioni
- **Latenza:** minima in quanto la CPU noterà subito lo stato della periferica (tempo massimo: un ciclo intero di busy waiting)
- Esempio: se si trasferiscono piu' dati (1KB) si ha un ciclo interno per il trasferimento di ogni byte e un ciclo esterno per il trasferimento di tutti i byte

## Prestazioni – I/O gestito da interrupt

- **Banda passante:** minor banda passante in quanto il trasferimento di ogni dato necessita' di piu' tempo
- **Latenza:** maggior latenza per la maggior quantita' di operazioni da eseguire

## Prestazioni – DMA

- **Banda passante:** massima perché la CPU non deve eseguire nessuna istruzione
- **Latenza:** minima dato che nessuna istruzione è eseguita dalla CPU

## Da leggere

- **Appendice A – Capitolo 8**
- **Tutte le presentazioni disponibili sul sito del corso**