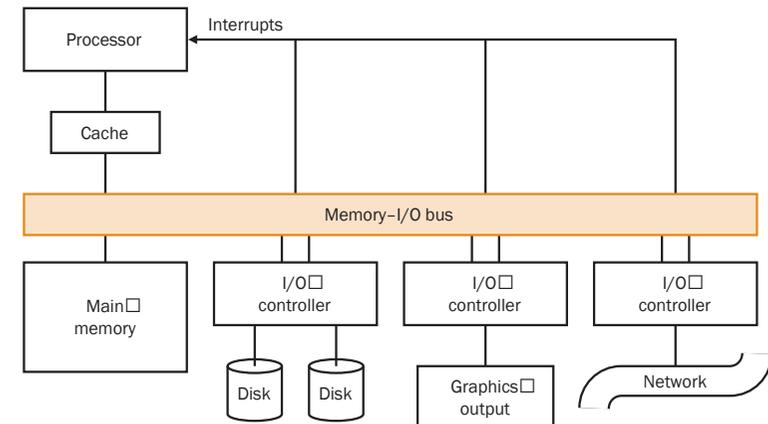


## Architetture dei Calcolatori

### Dispositivi di I/O

Prof. Francesco Lo Presti

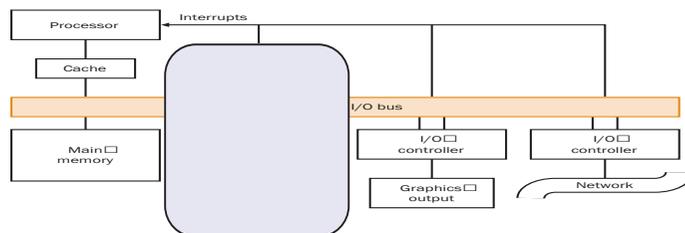
## Organizzazione di un Calcolatore



I/O 1

## Dispositivi di I/O

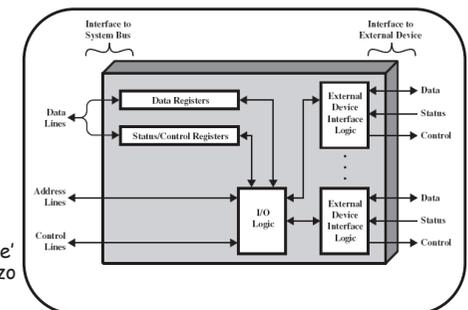
- Un dispositivo di I/O è costituito da due componenti:
  - Il **dispositivo fisico** effettivo: disco, stampante, mouse, video,...
  - Il **device controller** (o interfaccia) che gestisce tutte le operazioni che il dispositivo è in grado di svolgere
- Il device controller è collegato attraverso il **bus di sistema** con CPU e memoria principale



I/O 2

## Device Controller

- I Device Controller sono sottosistemi specializzati nel controllo dei dispositivi periferici
  - Svolgono funzioni di
    - ✓ Controllo e temporizzazione delle operazioni
    - ✓ Comunicazione con il Processore e con i Dispositivi fisici
    - ✓ Individuazione Errori
  - Scambio informazioni con il processore per mezzo di registri di I/O
    - ✓ I/O Isolato
      - Istruzione privilegiate IN/OUT
    - ✓ I/O Mappato in Memoria
      - Istruzioni load/store
      - Ciascun registro di I/O e' mappato su di un indirizzo di memoria



I/O 3

## Dispositivi di I/O

- I dispositivi di I/O hanno caratteristiche molto diverse tra loro, classificabili in base a:
  - Comportamento
  - Controparte (Partner)
  - Tasso di trasferimento dati

Dispositivo	Comportamento	Partner	Data rate (KB/sec)
Tastiera	input	umano	0.01
Mouse	input	umano	0.02
Stampante Laser	output	umano	200.00
Scheda Grafica	output	umano	60,000.00
Network/LAN	input o output	macchina	500.00-6000.00
Floppy disk	storage	macchina	100.00
Disco Magnetico	storage	macchina	2,000.00-10,000.00

4

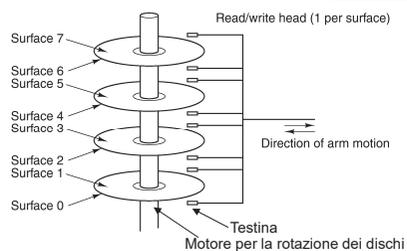
## Prestazioni ed I/O

- Le Prestazioni Complessive di un Sistema dipendono
  - Dalle prestazioni dei device e dei relativi controller
  - Dalla connessione tra i device e il resto del sistema
  - Dal Sistema Operativo
- **Misure di Prestazione dell I/O**
  1. **I/O bandwidth (throughput)**
    1. quantità di dati che possono essere trasferiti tra il sottosistema di I/O ed il processore/memoria per unità di tempo
    2. Quante operazioni di I/O possono essere completate per unità di tempo?
  2. **Tempo di risposta I/O (latenza)** - il tempo richiesto per il completamento di una operazione di input/output

I/O

5

## Disco Magnetico

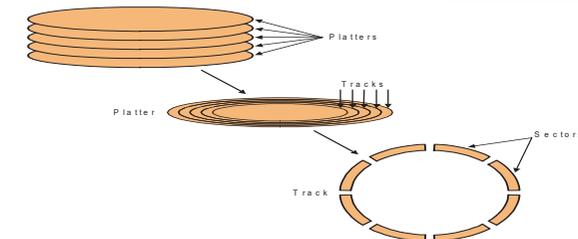


- Costituito da un insieme di piatti rotanti (da 1 a 15)
- Piatti rivestiti di una superficie magnetica
- Esiste una testina (bobina) per ogni faccia
  - Generalmente piatti a doppia faccia
- Le testine di facce diverse sono collegate tra di loro e si muovono contemporaneamente
- Velocità di rotazione costante (ad es. 7200 RPM)

I/O

6

## L'organizzazione dei dati sul disco



- Suddivisione della superficie del disco in anelli concentrici, detti **tracce**
- Registrazione seriale su tracce concentriche
  - 1000-5000 tracce
- Tracce adiacenti separate da spazi
- Ciascuna traccia è divisa in **settori**
  - Il settore è la più piccola unità che può essere trasferita (scritta o letta)
  - Centinaia di settori per traccia, generalmente di lunghezza fissa (ad es., 512 byte)
- La stessa traccia su piatti diversi forma un **cilindro**

I/O

7

## Prestazioni degli Hard Disk

- *Latenza per read/write*
- 1. **Tempo di seek** (seek time): tempo per muovere la testina sulla traccia corretta
  - Da 3 a 14 ms (può diminuire del 75% se si usano delle ottimizzazioni)
- 2. **Tempo di rotazione** (rotational latency): tempo per raggiungere il settore da trasferire (in media tempo richiesto per 1/2 rotazione del disco)
  - Tempo di rotazione =  $0.5/\text{Numero di giri al minuto}$
  - Ex, Numero di giri al minuto = 7200
  - ⇒ Tempo di rotazione =  $(0.5/(7200/60)) \cdot 1000 = 4.2 \text{ ms}$
- 3. **Tempo di trasferimento** (transfer time): tempo per trasferire un blocco di bit
  - Da 30 a 80 MB/sec (fino a 320 MB/sec se il controller del disco ha una cache built-in)
- 4. **Tempo per il controller**: tempo per le operazioni del disk controller
  - 0.2ms

I/O 8

## Prestazioni degli Hard Disk (2)

- Calcolare il tempo medio necessario a leggere o scrivere un settore di 512 byte sapendo che
  - Il disco ruota a 10000 RPM
  - Il tempo medio di seek è 6 ms
  - Il transfer rate è di 50 MB/sec
  - L'overhead del controller è di 0.2 ms

Tempo di seek + tempo medio di rotazione + tempo medio di trasferimento + overhead del controller =  
= 6 ms +  $(0.5/(10000/60)) \cdot 1000 \text{ ms}$  + 0.5 KB/50 MB/sec + 0.2 ms  
= (6.0 + 3.0 + 0.01 + 0.2) ms = 9.2 ms

I/O 9

## Invio dei comandi ad un dispositivo di I/O

- I comandi devono essere inviati al corrispondente device controller
- Un'istruzione di I/O in un linguaggio ad alto livello viene trasformata in una serie di comandi per il controller
  - La trasformazione avviene ad opera del compilatore che traduce l'istruzione in una chiamata al sistema operativo
- A runtime la chiamata del sistema operativo richiama uno dei moduli del SO che si occupano della gestione dell'I/O (device driver)
- Il device controller ha una serie di registri (**porte di I/O**) in cui memorizza
  - Lo stato della periferica (ad es.: idle, busy, down, ...)
  - Il comando in esecuzione
  - I dati da/verso il dispositivo di I/O

I/O 10

## Invio dei comandi ad un dispositivo di I/O (2)

- Il device controller può essere visto come un processore (con potenzialità ridotte)
  - Si parla di **processori di I/O**
- Per richiedere un'operazione di I/O il processore deve
  - Predisporre il contenuto dei registri del controller a valori predeterminati
  - Avviare il controller stesso
- L'operazione di selezione del controller e di predisposizione dei suoi registri può avvenire in due modi
  - **Memory-mapped I/O**
  - **Istruzioni di I/O dedicate**

I/O 11

## Memory-mapped I/O

- ❑ Lo spazio di indirizzamento dell'I/O appartiene allo stesso spazio di indirizzamento della memoria
  - I registri dei vari device controller sono considerati logicamente come locazioni di memoria, pur essendo fisicamente localizzati all'interno del device controller
- ❑ I device controller devono essere quindi dotati di un meccanismo che permetta loro di riconoscere le transazioni ad essi indirizzate
- ❑ I controller ascoltano tutti i segnali in transito sul bus (*bus snooping*) e si attivano solo quando riconoscono sul bus un indirizzo corrispondente ad una propria locazione di memoria

I/O 12

## Programmazione I/O

- ❑ Tre modalita' principali per la gestione/programmazione dei dispositivi di I/O
1. I/O Programmato con attesa
    - Forma piu' rudimentale di I/O
      - ✓ Durante l'intero svolgimento dell'operazione di I/O, il processore e' direttamente impegnato ad eseguirne le operazioni
  2. I/O Controllato da interruzioni
    - Basato sul meccanismo delle interruzioni
    - Processore comanda operazione
    - A completamento dell'operazione l'unita' di I/O richiederà "l'attenzione" da parte del processore tramite interruzione
      - ✓ Il processore - non impegnato a controllare l'I/O puo' svolgere altri compiti
  3. I/O con Accesso Diretto alla Memoria (DMA)
    - Le unita' di I/O hanno capacita' di procedere autonomamente al trasferimento dati da e verso la memoria
      - ✓ Il processore non interviene nel trasferimento
      - ✓ A fine operazione il processore verra' informato della conclusione dell'operazione di I/O

I/O 14

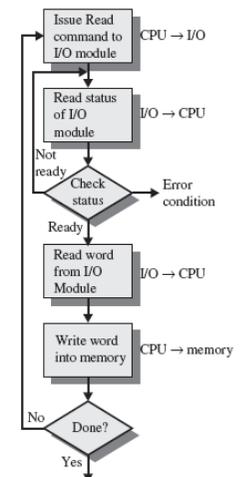
## Istruzioni dedicate

- ❑ Lo spazio di indirizzamento di I/O è separato dallo spazio di indirizzamento della memoria
- ❑ Per consentire al processore di accedere ai registri dei controller delle periferiche vengono inserite delle *istruzioni specifiche* nell'insieme delle istruzioni, dedicate alla gestione dell'I/O
- ❑ Queste istruzioni dedicate fanno riferimento esplicitamente al dispositivo interessato all'operazione di I/O

I/O 13

## L'I/O programmato con attesa

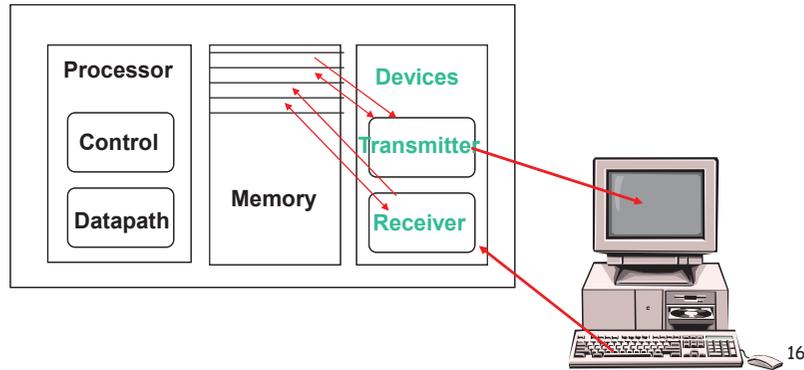
- ❑ Si richiede l'operazione di I/O (ex. Read da disco)
  - Scrivendo nel registro di controllo del device
- ❑ L'Unita' effettua l'operazione
  - Lo stato dell'operazione e' disponibile nel registro controllo
  - Il dato letto e' disponibile nel registro dati
  - L'unita' non informa il processore dello stato/conclusione dell'operazione
- ❑ Per conoscere lo stato dell'operazione bisognerà interrogare il registro di stato
- ❑ Ad operazione conclusa il dato puo' essere letto dal registro dati



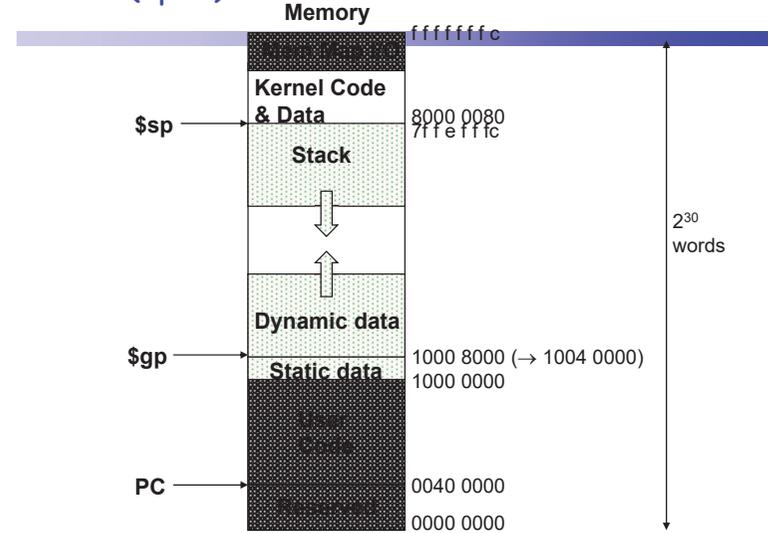
I/O 15

## I/O in SPIM

- SPIM supporta un dispositivo di I/O mappato un memoria - un terminale con due unità
  - Receiver legge caratteri dalla tastiera
  - Transmitter scrive caratteri sul display



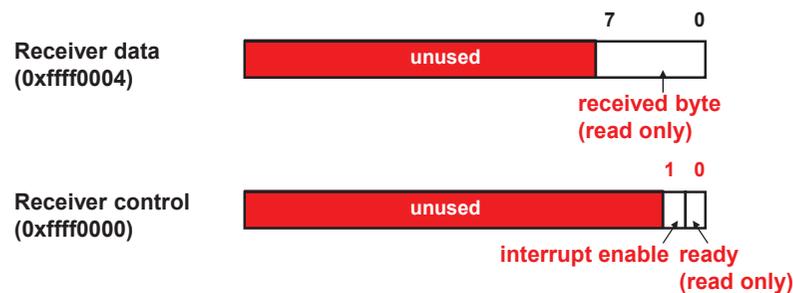
## MIPS (spim): Allocazione della Memoria



I/O 17

## Controllo dell'Input da Terminale in SPIM

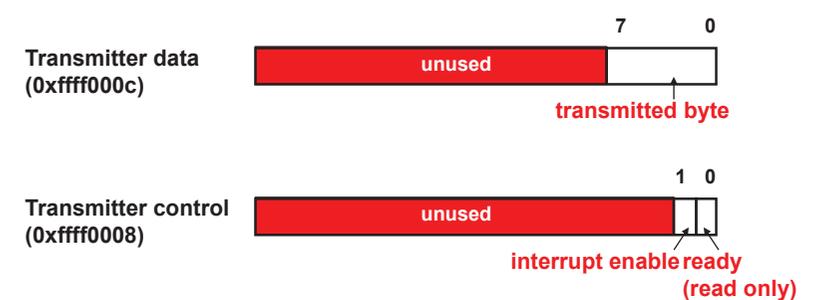
- Input da terminale si realizza tramite due registri del dispositivo di I/O mappati in memoria
  - ogni registro appare come un locazione di memoria speciale



I/O 18

## Controllo dell'Output in SPIM

- Output del terminale si controlla tramite due registri del dispositivo di I/O mappati in memoria
  - ogni registro appare come un locazione di memoria speciale



I/O 19

## I/O Programmato

- Bisogna aspettare che il bit di ready in ingresso/uscita sia pari ad 1 per poter leggere/scrivere un carattere

### Ingresso

```
lui $t0, 0xFFFF
wait: lw $t1, 0($t0)
      andi $t1, $t1, 0x0001
      beq $t1, $0, wait
      lw $v0, 4($t0)
```

### Uscita

```
lui $t0, 0xFFFF
wait: lw $t1, 8($t0)
      andi $t1, $t1, 0x0001
      beq $t1, $0, wait
      sw $a0, 12($t0)
```

I/O 20

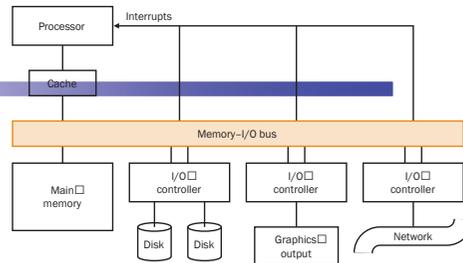
## L'I/O programmato con attesa: Polling

- Problemi con l'I/O con attesa
  - Con periferiche lente, un eccessivo spreco di tempo di processore che per la maggior parte del tempo è occupato nel ciclo di busy waiting
  - Con periferiche veloci, il lavoro svolto dal processore è quasi esclusivamente dovuto al trasferimento dati
- La Tecnica di Polling permette di avere piu' operazioni su dispositivi di I/O in contemporanea
  - Durante il ciclo di attesa, il software di controllo
    - ✓ Controlla via via le varie periferiche attive
    - ✓ Serve quelle che necessitano di un intervento

I/O 21

## Interruzioni

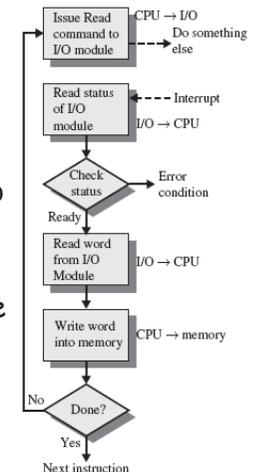
- Interruzione (interrupt) e' un meccanismo che consente di comunicare al processore il verificarsi di eventi esterni ed asincroni
- Il processore le gestisce in modo simile alle eccezioni
  - Eccezioni sono causate da eventi interni al processore
  - Riconoscimento dell'unita' che causa interruzione
    - ✓ Interruzioni Vettorizzate o Registro Cause
  - Esecuzione di una routine di servizio del sistema operativo che "serve" l'interruzione
- Nel caso dell'I/O permette alle unita' di richiedere "attenzione" al processore al completamento dell'operazione di I/O



I/O 22

## L'I/O con Interruzioni

- Si richiede l'operazione di I/O (ex. Read da disco)
  - Scrivendo nel registro di controllo del device
- L'Unita' esegue il comando inviatogli dal processore e quando è pronto allo scambio dei dati invia al processore un segnale di interrupt
- Il processore, attraverso una routine di gestione dell'interrupt (*interrupt handler*), provvederà a salvare il contesto esecutivo ed elaborare l'interrupt



I/O 23

## L'I/O con Interruzioni

- **Vantaggi**
- Tra il momento in cui termina l'invio del comando di I/O al controller e la ricezione dell'interrupt inviato dal controller, il processore è completamente svincolato dall'operazione di I/O e può dedicarsi ad altre attività
- **Limiti**
- Il meccanismo di interrupt non svincola il processore dal dover eseguire l'operazione di trasferimento dati
  - Per periferiche veloci, come i dischi, l'I/O con interruzioni richiede comunque una frazione significativa di tempo di CPU

I/O 24

## L'I/O con Interruzioni

- Meccanismi per identificare il dispositivo di I/O che ha generato l'interruzione:
  - **Più linee di interruzione**
    - ✓ Non è pratico dedicare molte linee del bus agli interrupt
    - ✓ Usato in combinazione ad uno degli altri 3 meccanismi
    - ✓ Semplice per priorità dell'interrupt
  - **Interrogazione software (software poll)**
    - ✓ Quando rileva un'interruzione, il processore esegue una routine che interroga i dispositivi di I/O per individuare chi l'ha causata
    - ✓ Svantaggio: perdita di tempo
  - **Interrogazione hardware, vettorizzata**
    - ✓ Linea di richiesta di interruzione comune a tutti i dispositivi di I/O
    - ✓ Quando rileva l'interruzione, il processore invia a sua volta un'interruzione di riconoscimento
    - ✓ Il dispositivo richiedente risponde inviando un vettore, usato dal processore come puntatore alla routine di gestione appropriata

I/O 25

## Priorità delle interruzioni

- Quando ci sono diversi dispositivi che possono inviare interruzioni al processore, oltre ad identificare il dispositivo che ha generato l'interruzione e la routine di gestione opportuna, occorre anche:
  - Gestire la priorità delle interruzioni
  - Gestire interruzioni annidate
    - ✓ Arriva un'interruzione mentre si sta gestendo un'altra interruzione
- Gestione della priorità
  - Semplice con linee di interruzioni individuali
  - Con l'interrogazione hardware (linea di interruzione comune) la priorità si gestisce tramite un'organizzazione dei dispositivi di tipo daisy chain
    - ✓ L'interruzione di riconoscimento viene ricevuta prima dal dispositivo a priorità più elevata e da questi eventualmente propagato al dispositivo successivo

I/O 26

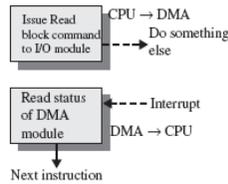
## Interruzioni annidate

- Soluzione semplice: si disabilitano le interruzioni durante il servizio di un'interruzione
  - Soluzione restrittiva, in contrasto con le differenti priorità dei dispositivi
- Soluzione più adottata: un'interruzione a priorità più alta può interrompere il servizio di un'interruzione a priorità minore

I/O 27

## Direct Memory Access (DMA)

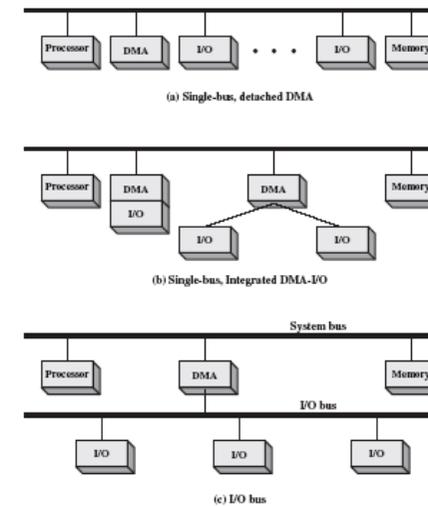
- Il DMA controller è un processore specializzato nel trasferimento dei dati tra dispositivi di I/O e memoria principale
  - Il DMA controller attua direttamente il trasferimento dati tra periferiche e memoria principale *senza l'intervento del processore*
- A fronte di una richiesta di I/O, il processore invia al DMA controller
  - Tipo di operazione richiesta
  - Dispositivo di I/O
  - Indirizzo di memoria da cui iniziare a leggere/scrivere i dati
  - Numero di byte da leggere/scrivere
- Il DMA controller avvia l'operazione richiesta e trasferisce i dati da/verso la memoria
- Completato il trasferimento, il DMA controller invia un interrupt al processore per segnalare il completamento dell'operazione richiesta



I/O 28

## Direct Memory Access (DMA)

- Possibili configurazioni DMA



I/O 29

## I/O e Sistema Operativo

- Sistema Operativo fornisce interfaccia tra i Dispositivi di I/O ed i Programmi Utente
  - Tramite chiamate di Sistema
    - ✓ La Risorsa e' "virtualizzata"
    - ✓ I programmi utente non hanno accesso diretto ai dispositivi
- Il Sistema Operativo ha il compito di inviare i comandi ai dispositivi di I/O, di gestire le interruzioni da questi generati, di schedare le operazioni di lettura/ scrittura degli utenti

I/O 30