

# I/O, dispositivi e tecniche

Giuseppe Pozzi

Impianti di Elaborazione  
Facoltà di Ingegneria dell'Informazione  
Politecnico di Milano

giuseppe.pozzi@polimi.it  
- versione del 30 settembre 2003 -

## Dispositivi di Ingresso/Uscita (I/O)

### • Bibliografia:

Tanenbaum A. S., Goodman J. R.,  
"Architettura dei computer - Un approccio strutturato",  
Prentice Hall International, 2000 - paragrafo 2.4, capitolo 5

### • Sommario:

- Bus
- Terminali
- Mouse
- Stampanti
- Modem

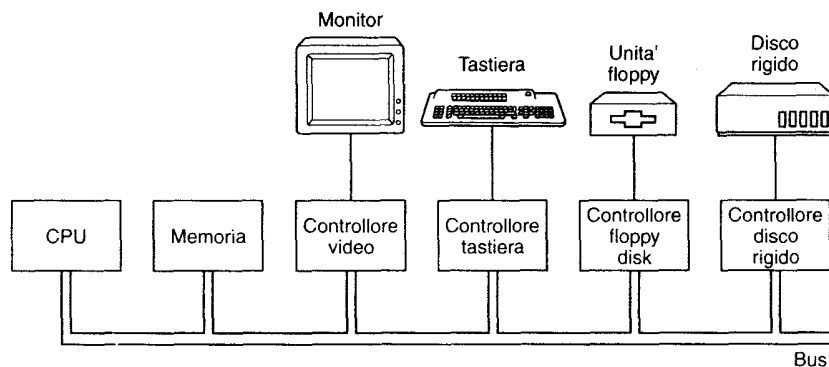


30 settembre 2003

Impianti di Elaborazione - Ingresso/uscita

2

## Struttura logica di un PC



30 settembre 2003

Impianti di Elaborazione - Ingresso/uscita

3

## Struttura fisica di un PC

- Nella scatola (case) sono contenuti:
  - una **scheda madre** che contiene una CPU, alcuni connettori nei quali inserire moduli DIMM e vari chip di supporto;
  - uno o due **bus**, uno ad alta velocità (per schede moderne) e uno a bassa velocità (per schede più vecchie);
  - prese in cui si possono inserire i connettori delle **schede di I/O** che agiscono da **controllori** dei dispositivi di I/O, cioè ne **gestiscono l'accesso al bus**:
    - un controllore che legge o scrive dati verso e da una memoria senza interventi da parte della CPU effettua un accesso diretto alla memoria (**Direct Memory Access – DMA**)
    - completato il trasferimento, il controllore **effettua un interrupt**, la CPU sospende il programma in corso e inizia una procedura speciale, (**interrupt handler**); quando l'interrupt handler termina, la CPU continua con il programma.

30 settembre 2003

Impianti di Elaborazione - Ingresso/uscita

4

## Terminali

- Composti di due parti: **tastiera** e **schermo**.
  - Nel mondo dei mainframe, sono integrati in un dispositivo singolo e collegati al calcolatore principale per mezzo di una linea seriale
  - Nel settore dei personal computer, sono dispositivi separati.
- **Tastiere**
  - molti tipi diversi, meccaniche o elettromagnetiche;
  - quando si preme un tasto viene generato un interrupt e viene avviato il gestore degli interrupt della tastiera, che legge un registro hardware all'interno del controllore della tastiera per avere il numero del tasto (da 1 a 102) premuto;
  - quando il tasto viene rilasciato si verifica un secondo interrupt.

## Terminali grafici

- Visualizzazione "**bit map**": lo schermo è una matrice di **pixel indipendenti**
  - per indicare il **colore** di ogni pixel si usano fino a **32 bit** (8 bit per ogni colore fondamentale + 8 bit per la trasparenza);
  - per rappresentare un carattere si usa un rettangolo di pixel e si configurano i bit necessari per visualizzare il carattere (così si possono realizzare diversi **font**);
  - comodi per i **sistemi operativi a finestre**;
  - richiedono una **memoria video** di grandi dimensioni
    - VGA: 640 x 480 x 4 byte = 1.2 Mbyte
    - SVGA: 800 x 600 x 4 byte = 1.9 Mbyte
    - XGA: 1024 x 768 x 4 byte = 3.2 Mbyte
    - UXGA: 1600 x 1200 x 4 byte = 7.5 Mbyteriducibili grazie all'utilizzo di una "**palette**" (scelta di **2<sup>8</sup>=256** colori tra i **2<sup>32</sup>** possibili).

## Porte Standard

- **Interfaccia Seriale**
  - Trasporta un bit per volta.
  - Velocità massima di 115 kbps
  - Utilizzata per periferiche lente come tastiera, mouse e modem esterni
- **Interfaccia parallela**
  - Trasporta 8 bit alla volta.
  - Velocità di 150 KB/sec (2MB/s in modalità EPP)
  - Usata per stampanti, scanner e unità di backup (nastri, Zip).
- Direzione della comunicazione
  - **Simplex**: la linea trasmette solo in una direzione;
  - **Half-duplex**: la linea trasmette in entrambe le direzioni ma non contemporaneamente (una direzione per volta);
  - **Full-duplex**: la linea trasmette contemporaneamente in entrambe le direzioni.

## Modem

- Connessione di calcolatori attraverso la rete telefonica (**analogica**).
- Velocità crescenti dal 1980 in poi
  - V.22bis, V.32 & V.32bis furono i primi standard per velocità di 2.4, 9.6 e 14.4Kbit/s.
  - V.34 (1994) supporta 28.8Kbit/s e corrisponde al minimo livello attualmente accettato
  - V.34+ (1996) arriva a 33.6Kbit/s
  - V.90 arriva a 56Kbit/s downstream e a 33.6Kbit/s upstream.
    - downstream indica dal digitale all'analogico
    - upstream indica dall'analogico al digitale

## Integrated Services Digital Network - ISDN

- Linea analogica sostituita da **linea digitale**
  - in realtà non viene sostituita la linea, ma solo le **attrezzature alle due estremità**.
  - **Uso domestico: due canali** digitali indipendenti, ognuno da 64'000 bit/sec, e un canale di segnalazione da 16'000 bit/sec (per un totale di **144'000 bps**)
  - **Uso commerciale:** 30 canali per uso commerciale.
- Caratteristiche
  - tempo di **setup** della connessione praticamente nullo (1 s);
  - non serve più un modem analogico (**connessione digitale-digitale**);
  - è molto più **affidabile** (meno errori) di una linea analogica.

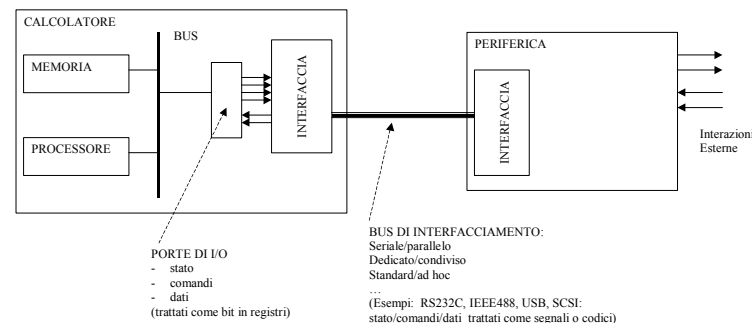
## Tecniche di Ingresso/Uscita (I/O)

- **Bibliografia:**
  - Tanenbaum A. S., Goodman J. R.,  
"Architettura dei computer - Un approccio strutturato",  
Prentice Hall International, 2000 - **capitolo 5**
- **Sommario:**
  - A controllo da programma
  - A interruzione
  - In DMA (Direct Memory Access)

## I/O

- L' I/O può essere effettuato in 3 modalità:
  - controllo da programma;
  - interruzione (interrupt);
  - DMA (Direct Memory Access).

## Collegamento di periferica a calcolatore

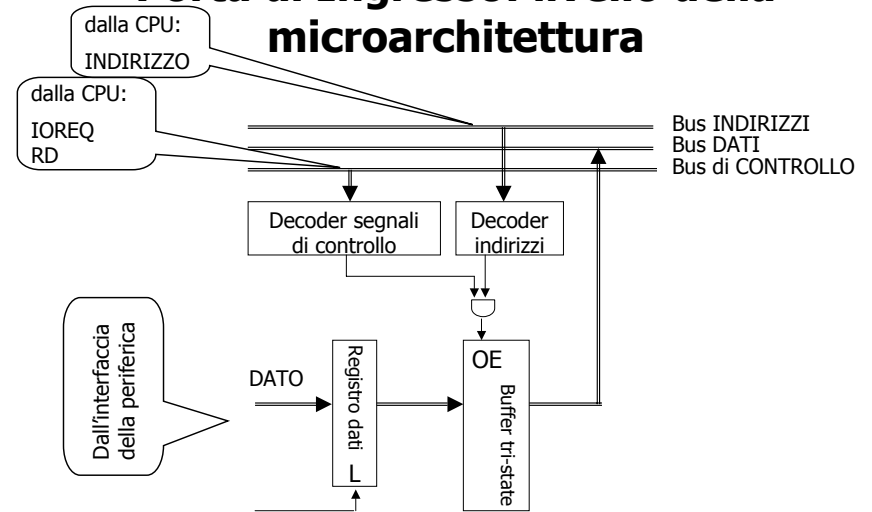


# Buffer tri-state

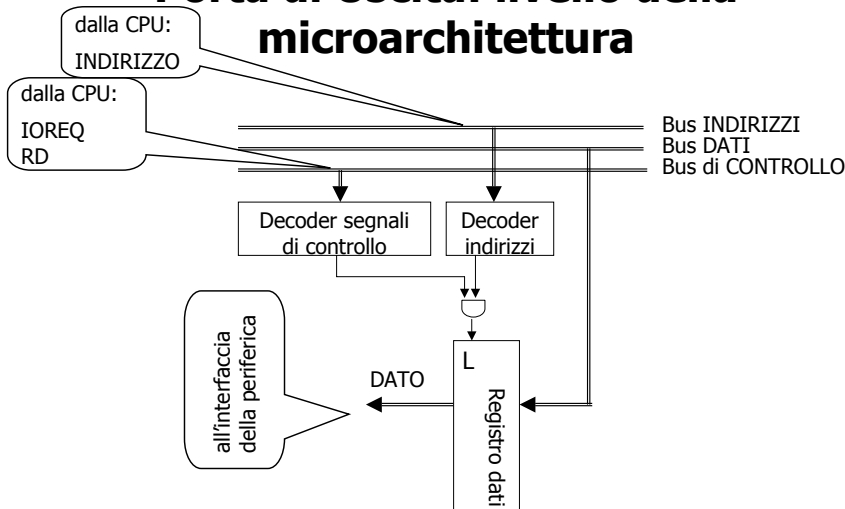
E' il circuito elementare modellabile come un contatto a tre posizioni:

- in stato di bassa impedenza consente di avere in uscita o il **livello alto** (1) o il **livello basso** (0)
- in stato di alta impedenza (Z) **isola elettricamente** l'uscita
- l'uscita tri-state viene gestita da un apposito ingresso di controllo (**O**uput **E**nable) che, se non attivo, forza lo stato di alta impedenza

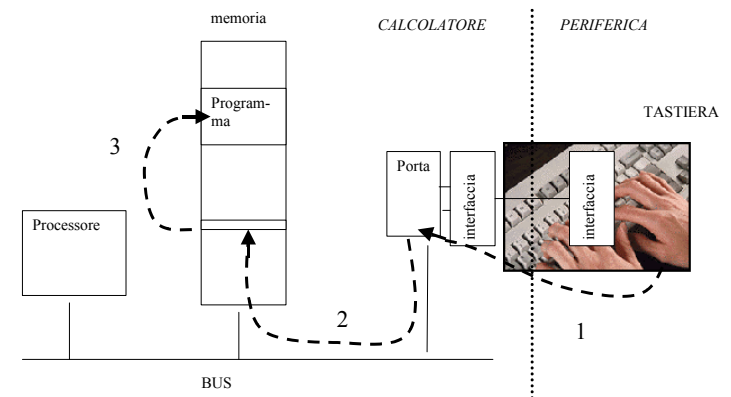
# Porta di Ingresso: livello della microarchitettura



# Porta di Uscita: livello della microarchitettura



# Esempio di ingresso



## Esigenze

- Evitare perdite o duplicazioni di dati.
- Consentire comunicazioni asincrone.
- Nel caso di lettura da tastiera, le comunicazioni sono:
  - da tastiera a porta;
  - da porta a cella di memoria;
  - da cella di memoria a programma che utilizza il dato;

indipendentemente dalle modalità di gestione di I/O adottata.

## A controllo da programma

- Durante la sua normale esecuzione un programma esegue una istruzione di lettura della porta:
  - es Intel: `IN R0, INDIRIZZOPORTA`
  - es Motorola: `MOV INDIRIZZOPORTA, R0`
- Nella fase di esecuzione di questa istruzione il processore esegue il ciclo di bus di lettura della porta.
- Il programmatore ha deciso dove, nel programma, inserire questa istruzione.
- Il flusso dell'esecuzione del programma stabilirà **quando** l'istruzione verrà eseguita.

## A interruzione

- La parte di programma che legge la porta (ad es. con la istruzione `IN R0, INDIRIZZOPORTA`) **NON** è nel programma ma è silente in memoria in una locazione convenuta.
- Quando l'interfaccia della periferica porta il dato alla porta di ingresso, con un segnale allerta il processore.
- Il processore interrompe l'esecuzione del programma in corso e salta automaticamente a eseguire la parte di programma che legge la porta. La lettura avviene come nel caso precedente.
- Al termine di questo, il processore riprende il programma interrotto.
- In pratica, la periferica ha deciso **quando** l'istruzione di lettura della porta deve essere eseguita.

## DMA

- Quando l'interfaccia della periferica porta il dato alla porta di ingresso, manda un segnale al processore, imponendogli di lasciare libero il bus.
- Appositi circuiti generano un ciclo di bus che forza l'attivazione della porta, genera l'indirizzo in memoria dove deve finire il dato, comanda la memoria alla scrittura.
- Intanto, il processore non utilizza il bus.
- Terminato il ciclo, l'interfaccia della periferica manda un altro segnale al processore, lasciandolo libero di proseguire.
- In pratica, alcuni circuiti di I/O hanno scritto il dato in memoria, pochi nanosecondi dopo il suo arrivo.

## Analizziamo le 3 modalità

- Controllo da programma;
- interruzione;
- DMA.

## A controllo da programma (ingresso)

- Esiste un apposito programma che:
  - legge le porte a cui si presentano i bit di stato ed i dati dell'interfaccia della periferica;
  - scrive le porte che forniscono bit di comando all'interfaccia di periferica.
- In questo caso, il programma che acquisisce il dato, lo memorizza e lo usa è unico.

## A controllo da programma (ingresso)

- Ideale se:
  - il ciclo di istruzioni per la lettura è inserito nel programma utente;
  - quando il programma utente è in attesa che venga premuto un tasto, non ha null'altro da fare;
  - non c'è nessun altro programma utente da eseguire;
  - quindi la soluzione di ripetere all'infinito il ciclo di istruzioni per la lettura non ha controindicazioni.

## A controllo da programma

- La soluzione a controllo di programma è molto veloce:
  - viene eseguito un ciclo di tre istruzioni per sentire se il codice è arrivato;
  - vengono eseguite due istruzioni per acquisire il codice e scriverlo in memoria.

## A controllo da programma

```
LS: MOV Stato, R0  
    CMP R0, #0  
    BEQ LS
```

```
MOV DATO, R1  
MOV R1, Memoria
```

## A controllo da programma (uscita)

- Stampa di un carattere.
  - L'interfaccia della stampante tramite un bit comunica che è pronta ad accettare un nuovo codice di carattere da stampare.
- Occorrono circuiti analoghi a quelli visti per la tastiera:
  - una porta di ingresso, per leggere il bit di pronta ad accettare (indirizzo `INDCONTRST`);
  - una porta di uscita, su cui il programma scrive il codice del carattere (indirizzo `INDDATOST`);
  - il circuito che genera il bit di pronto ad accettare (analogo a quello che generava il bit di codice arrivato).

## Problemi tipici del controllo da programma

- Con più periferiche da gestire, e da gestire entro tempi massimi prestabiliti, la cosa si fa complessa.

## Gestione a interruzione

- Ricordando che se nessun ciclo di bus legge la porta, il dato continua a rimanere disponibile all'infinito e non entra mai nel calcolatore, la gestione a interruzione permette il passaggio di un dato:
  - da una porta di ingresso a una cella di memoria (ingresso);
  - da una cella di memoria a una porta di uscita (uscita).

## Funzionamento a interruzione

- La parte di programma che legge la porta (ad es. con l'istruzione `IN R0, INDIRIZZOPORTA` la quale nella sua esecuzione provoca il ciclo di bus che abilita la porta) **NON** è nel programma ma è silente in memoria, isolata, in un luogo di memoria convenuto.
- Quando l'interfaccia della periferica pone il dato alla porta di ingresso, con un segnale allerta il processore.
- Il processore interrompe l'esecuzione del programma in corso, salva lo stato corrente, e salta automaticamente a eseguire la parte di programma che legge la porta. La lettura avviene come nel caso precedente.

## Funzionamento a interruzione

- Al termine di questo, il processore ricarica lo stato precedente e riprende l'esecuzione del programma interrotto.
- In pratica, la periferica ha deciso quando l'istruzione di lettura della porta deve essere eseguita.
- L'interruzione "scatena l'esecuzione del programma di risposta all'interruzione" (o "routine di servizio").

## La gestione a interruzione

- Richiede componenti hardware e software addizionali a livello di:
    - sistema;
    - ISA;
    - microarchitettura
- sia nel calcolatore che nella interfaccia della periferica.

## Caso reale

- Vettore di interruzione:
  - durante la fase di accettazione di una interruzione, il processore attiva il segnale `INTACKNOWLEDGE` e genera un ciclo di bus di lettura senza indirizzo;
  - questo ciclo è identificato da appositi segnali di controllo;
  - solo la periferica interrompente che è di turno (tra tutte le contemporaneamente interrompenti è quella più prioritaria) partecipa a questo ciclo di bus, ponendo sul bus dati un numero che la identifica (il **vettore dell'interruzione**);
  - il processore usa il vettore per determinare l'indirizzo in memoria della procedura di risposta corrispondente all'interruzione generata.



## Gestione a DMA

- La gestione a DMA (accesso diretto alla memoria) del passaggio di un dato:
  - da una porta di ingresso a una cella di memoria (ingresso);
  - da una cella di memoria a una porta di uscita (uscita).
- Consideriamo ancora il nostro esempio di ingresso.

## Gestione a DMA

- Quando l'interfaccia della periferica porta il dato alla porta di ingresso, manda un segnale al processore, imponendogli di lasciare libero il bus.
- Non appena il bus è libero, appositi circuiti dell'interfaccia di periferica generano **un ciclo di bus** che:
  - forza l'attivazione della porta, che pone il dato sul bus dati;
  - genera l'indirizzo della cella di memoria in cui il dato deve essere scritto;
  - comanda la memoria alla scrittura (dal bus dati alla cella di memoria).

## Gestione a DMA

- Intanto, il processore è fermo, bloccato, isolato dal bus, non deve fare nulla (non si salva lo stato).
- Terminato il ciclo, l'interfaccia della periferica disattiva il segnale di richiesta al processore, lasciandolo libero di riattivarsi.
- In pratica, dei circuiti di I/O hanno scritto il dato in memoria **pochi nanosecondi dopo il suo arrivo** (il tempo richiesto per eseguire un ciclo di bus).
- Il DMA "ruba" un ciclo di bus al processore, che **non partecipa** al passaggio del dato dalla periferica alla memoria. Il trasferimento avviene tutto a livello di microarchitettura.

## Osservazioni sul DMA

- Il vantaggio principale del controllo a DMA consiste nei tempi molto rapidi di trasferimento, e quindi permette "raffiche" di trasferimenti molto rapide e intense di blocchi di dati (utili in molte periferiche: ad es. nei controllori dei dischi una raffica può trasferire molto velocemente un settore intero);
- i circuiti necessari nelle interfacce sono complessi, ma aumentandone di poco la complessità si ottengono circuiti in grado di controllare sequenze di trasferimenti (una "raffica", appunto), perché basta usare un registro (ad autoincremento) contenente l'indirizzo di memoria dove deporre il singolo dato della raffica, e un registro (ad autodecremento) per contare il numero di dati che rimangono da trasferire nella raffica;
- di conseguenza, nelle architetture solite, le interfacce a DMA gestiscono raffiche di trasferimenti (ogni raffica, un blocco di dati: DMA a blocchi).

# Osservazioni sul DMA

- Il furto di bus è molto breve;
- il ciclo di bus non può essere spezzato in parti;
- quindi per semplicità richieste contemporanee di DMA vengono accettate ma eseguite solo in successione. Padrona del bus è sempre la CPU, la periferica che ha chiesto e ottenuto il bus non è in grado di accettare richieste di DMA più prioritarie e di cedere temporaneamente a queste l'uso del bus, per poi riprenderselo quando questi DMA più prioritari sono finiti.

# Esempio di funzionamento del DMA

- Il controllore del disco muove la testina posizionandola sulla traccia voluta. Il disco ruota, e il controllore del disco legge i vari settori. Finalmente sotto la testina passa il settore voluto, che viene letto dal controllore. Quando il singolo carattere è disponibile, il controllore del disco lo passa all'interfaccia di DMA e le chiede un ciclo di DMA;
- l'interfaccia di DMA scrive il carattere in memoria, nella cella di turno del buffer, incrementa l'indirizzo memoria, pronto per il successivo carattere, decrementa il conta caratteri;
- dopo aver letto l'ultimo carattere del settore, il controllore del disco smette di agire;
- ricevendo l'ultimo carattere, quando l'interfaccia di DMA riceve l'ultimo carattere del settore, lo scrive in memoria. Ora il suo conta caratteri si è azzerato;
- l'interfaccia di DMA genera un segnale di interruzione (interruzione di fine DMA).

# Azioni del programma di risposta a interruzione di fine DMA

- Il programma di risposta all'interruzione di fine DMA trova il buffer ormai pieno della copia del settore voluto.
- Il programma di risposta disabilita l'interfaccia di DMA sia al DMA sia all'interruzione di fine DMA, scrivendo gli appositi bit di controllo nel registro di controllo dell'interfaccia.
- Il programma di risposta chiama la parte di sistema operativo che toglie il programma da quelli in attesa di un evento esterno e lo mette tra quelli pronti per essere eseguiti, e poi torna: il programma di risposta viene concluso. Il programma era certamente in attesa, altrimenti non si sarebbe scatenata la sequenza di DMA.

# Confronti tra tecniche di I/O

|                              | Tempo minimo tra due trasferimenti successivi                                   | Costo circuitale tipico                      | Esecutore del trasferimento del singolo dato  | Periferiche tipiche  |
|------------------------------|---|--|---|--|
| <b>Controllo a programma</b> | Medio   | Molto basso                                  | Programma   | Periferica che fornisce/accetta il dato quando richiesto dal programma, <i>sempre e velocemente</i> , in modo certo e con temporizzazioni note; inoltre, la perdita o la duplicazione di un dato è irrilevante.<br><i>Esempio: termometro da forno</i> |
| <b>Interruzione</b>          | Alto (lento, per la necessità di salvataggio e ripristino dello stato)          | Medio (uso di un controllore d'interruzione) | Programma, a esecuzione imposta dall'interfaccia  | Periferica che di tanto in tanto, in tempi imprevedibili dai programmi, scambia un dato che non deve essere perso o duplicato, oppure richiede attenzione e reazione.<br><i>Esempi: tastiera, sensore di allarme</i>                                   |
| <b>DMA</b>                   | Bassissimo (molto veloce, vicino al limite massimo consentito da memoria e bus) | Alto (uso di un controllore di DMA)          | Circuito controllore di DMA, su richiesta dei circuiti di interfaccia della periferica. | Periferica che di tanto in tanto scambia raffiche di dati velocissime; i dati non devono essere persi o duplicati.<br><i>Esempio: disco magnetico</i>  |