

Corso di Architettura degli Elaboratori e Laboratorio (M-Z)

Interfacce di I/O

Nino Cauli

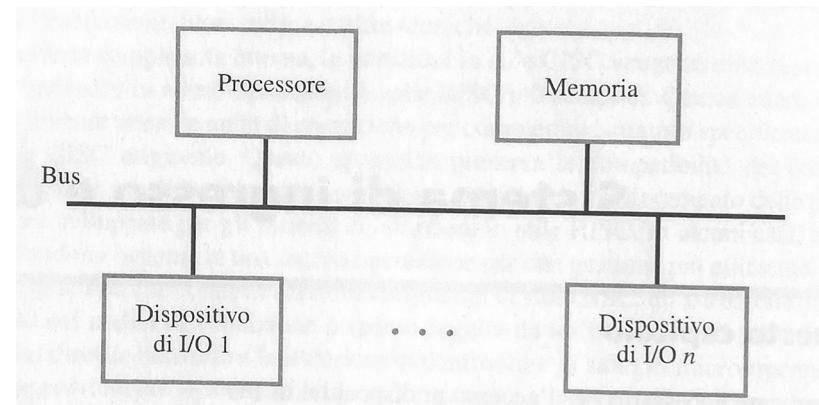


UNIVERSITÀ
degli STUDI
di CATANIA

Dipartimento di Matematica e Informatica

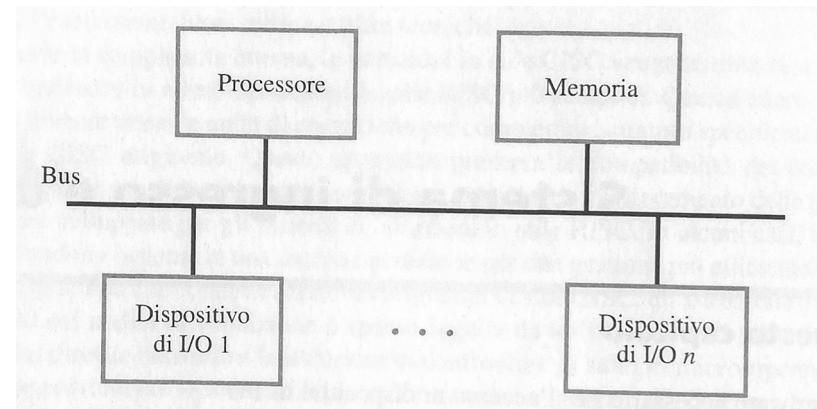
Bus di sistema

- Il Bus di sistema è la rete di interconnessione tra processore, memoria e dispositivi di I/O
- Il Bus è formato da un fascio di linee da un bit ciascuna
- Le linee sono divise in tre gruppi:
 - Dati**: contenenti i dati trasmessi
 - Indirizzi**: contenenti l'indirizzo dei registri dell'unità slave
 - Controllo**: contenenti comandi relativi all'operazione sul bus (i.e. un bit R/W) e bit di sincronizzazione
- Tutte le unità sono connesse al bus tramite delle porte chiamate **BUS DRIVER**



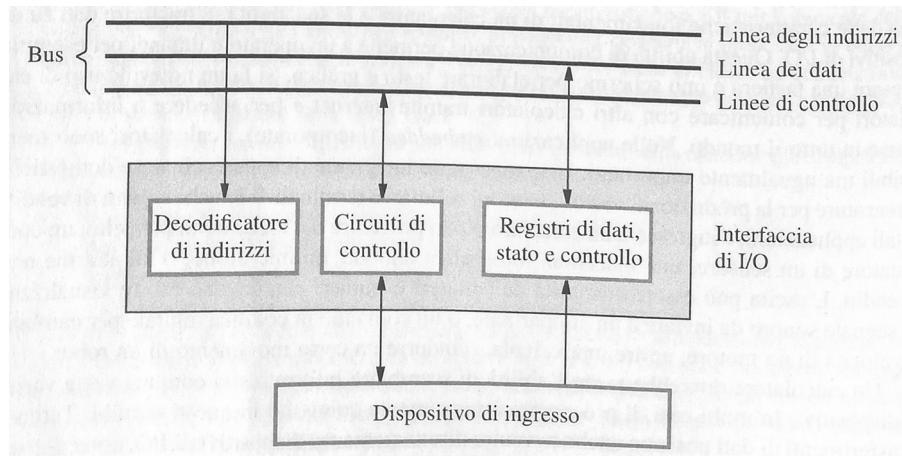
Bus di sistema

- Le due unità coinvolte in un trasferimento di dati assumono due ruoli distinti:
 - **Unità master**: che richiede il trasferimento (lettura o scrittura)
 - **Unità slave**: che risponde alla richiesta
- Solo **un'unità alla volta** può trasferire dati sul bus
- I **Bus driver** sono implementati con porte **tri-state** attive solo quando avviene un trasferimento
- 2 tipi di funzionamento del Bus: **SINCRONO** e **ASINCRONO**



Interfacce di I/O

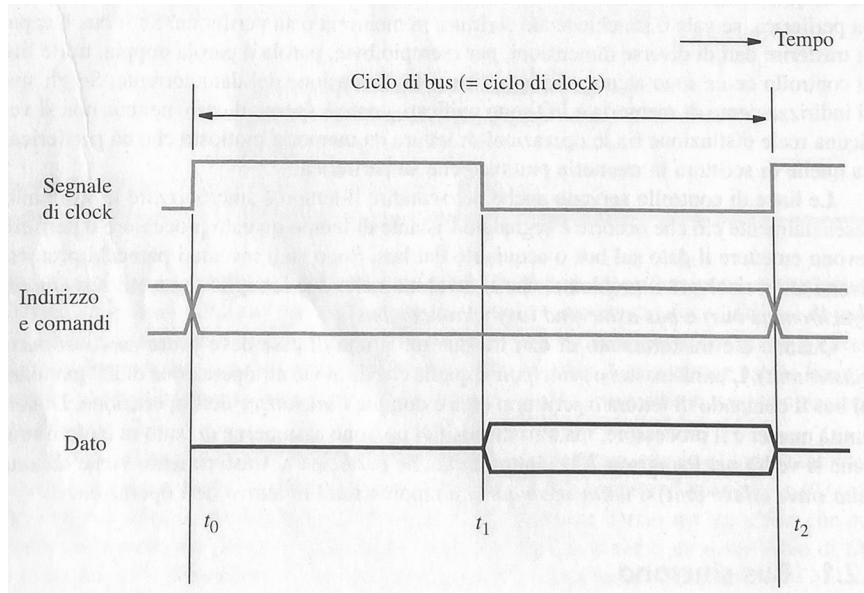
- I dispositivi di I/O sono collegati al bus attraverso le **Interfacce di I/O**
- Le interfacce di I/O contengono:
 - **Decodificatore di indirizzi**: per sapere se rispondere alla richiesta
 - **Circuiti di controllo**: per sapere che tipo di operazione eseguire
 - **Registri di dati, stato e controllo**: per gestire la richiesta
- L'interfaccia ha un lato dispositivo ed un lato Bus



- Un **segnale di clock** è distribuito tramite una linea del bus a tutte le unità collegate
- Un **ciclo di clock** è il tempo che intercorre tra un fronte di salita (o discesa) ed il successivo
- L'operazione di trasferimento è divisa in **cicli di bus**
- Un ciclo di bus può durare uno o più cicli di clock

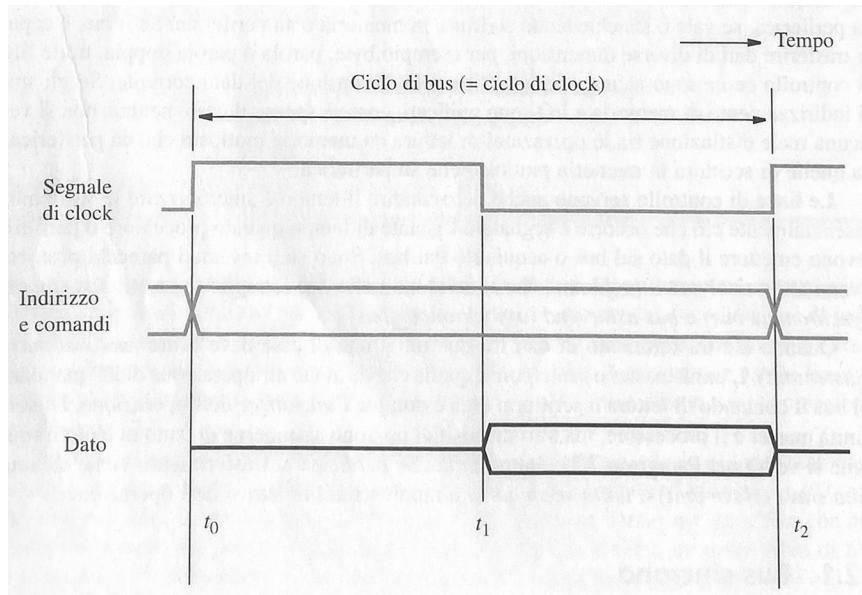
Diagrammi temporali

- Ogni riga rappresenta lo stato di uno o di un gruppo di segnali
- Le ascisse rappresentano il tempo
- Se il segnale nella riga è singolo:
 - Linea alta = segnale a 1
 - Linea bassa = segnale a 0
- Se la riga rappresenta un gruppo di segnali
 - Doppia linea = bit con valore differente tra loro (il valore di ogni bit può essere scritto tra le linee)
 - Incrocio linee = cambio valore dei bit
 - Linea centrale = valore indefinito, non significativo o linea disattivata



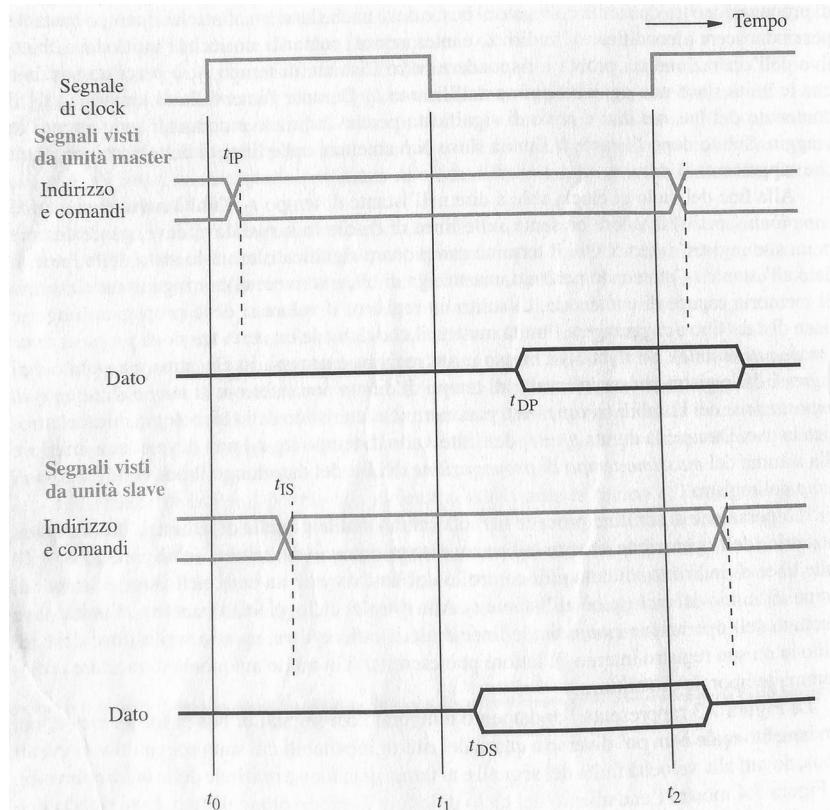
Bus sincrono semplice (lettura)

- 1 ciclo di bus = 1 ciclo di clock
- L'intera operazione di lettura avviene in un ciclo di clock
- Nessuna procedura di riscontro per accertarsi dell'esito dell'operazione
- L'operazione è divisa in 2 fasi:
 - $[t_0 - t_1]$: il master invia sul Bus indirizzo del dispositivo di ingresso e comandi (bit R/W=1 e dimensione dato). lo slave riconosce il suo indirizzo e legge i comandi
 - $[t_1 - t_2]$: lo slave scrive i dati sul Bus e a fine ciclo di clock i dati vengono scritti su dei registri del master



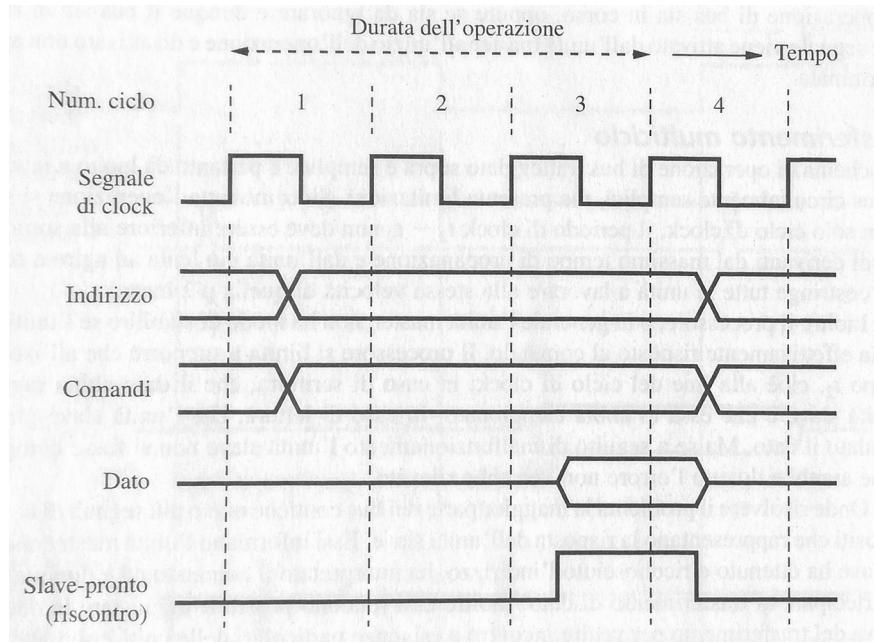
Bus sincrono semplice (lettura) - Ritardi

- I segnali impiegano del tempo a percorrere i circuiti
- Un diagramma più preciso tiene conto dei ritardi:
 - $[t_0 - t_{IP}]$: tempo che il master impiega a produrre e immettere sul Bus i segnali
 - $[t_{IP} - t_{IS}]$: tempo che impiegano i segnali a percorrere il Bus
 - $[t_1 - t_{DS}]$: tempo che lo slave impiega a produrre e immettere sul Bus i dati
 - $[t_{DS} - t_{DP}]$: tempo che impiegano i segnali a percorrere il Bus



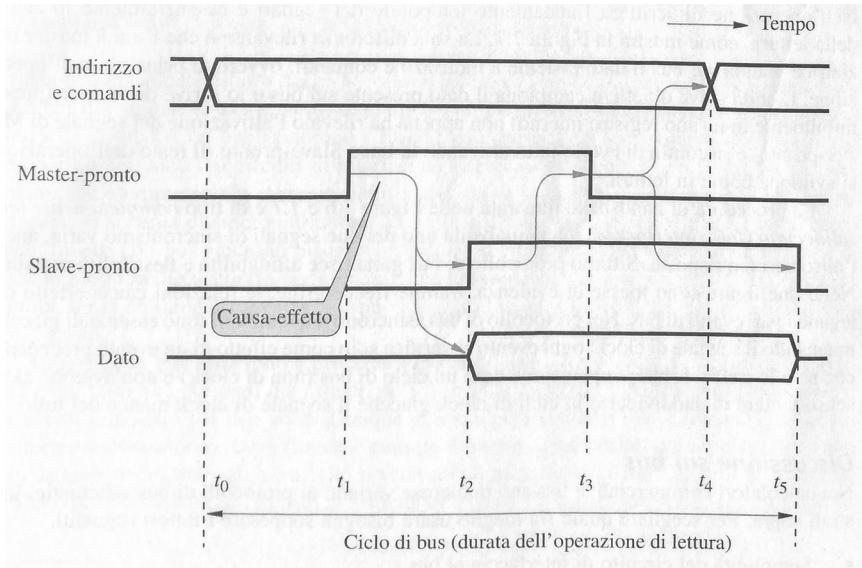
Bus sincrono multiciclo (lettura)

- Usato nel caso l'unità slave necessiti di diversi cicli per rispondere (i.e. accessi di memoria)
- 1 ciclo di bus = 2 o più cicli di clock
- Segnale di controllo **Slave-pronto** messo a 1 dallo slave quando i dati sono stati immessi nel Bus
- Il master scrive i dati nei registri e termina il trasferimento quando Slave-pronto = 1
- Riscontro solo dell'unità slave



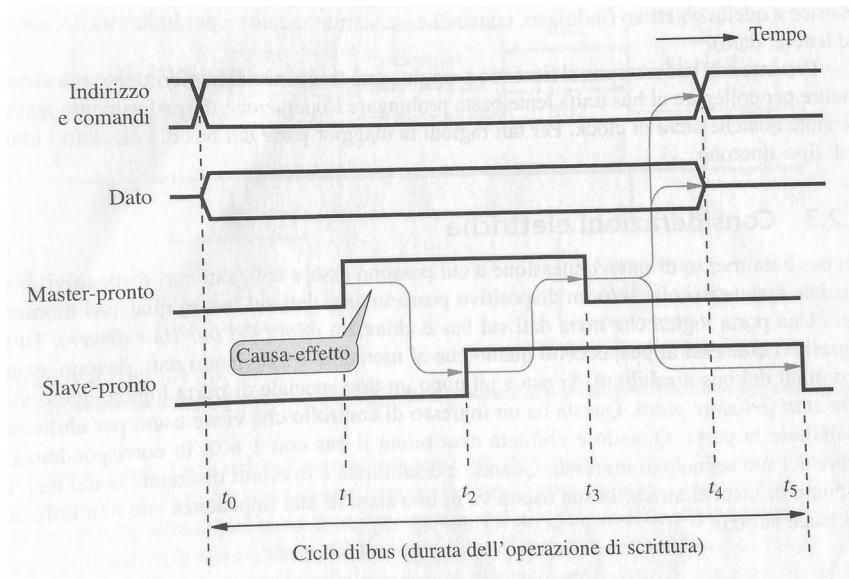
Bus asincrono (lettura)

- Nessun segnale di clock condiviso tra master e slave
- Sincronismo ottenuto tramite una procedura di **handshake**
- Due segnali di riscontro:
 - **Master-pronto**: attivata quando il master è pronto all'operazione e disattivata quando viene espletata
 - **Slave-pronto**: usata per riscontrare l'unità master
- Il ritardo nell'attivazione del master-pronto è dovuto allo **sfasamento temporale (skew)** tra unità slave

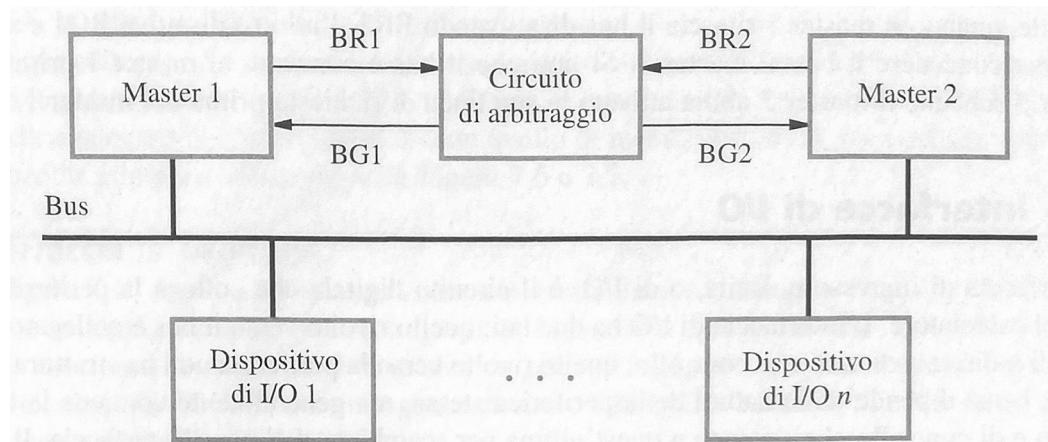


Bus asincrono (scrittura)

- Procedimento simile a quello di lettura
- Ad inizio operazione il master inserisce nel Bus indirizzo, comandi ($R/W = 0$) e dati da scrivere
- Master-pronto viene messo a 1 (ritardato per tener conto dello skew)
- L'unità slave riscontra la scrittura dei dati mettendo a 1 Slave-pronto
- Ricevuto il riscontro, l'unità master disattiva Master-pronto e rimuove i dati dal bus
- L'unità slave disattiva Slave-pronto

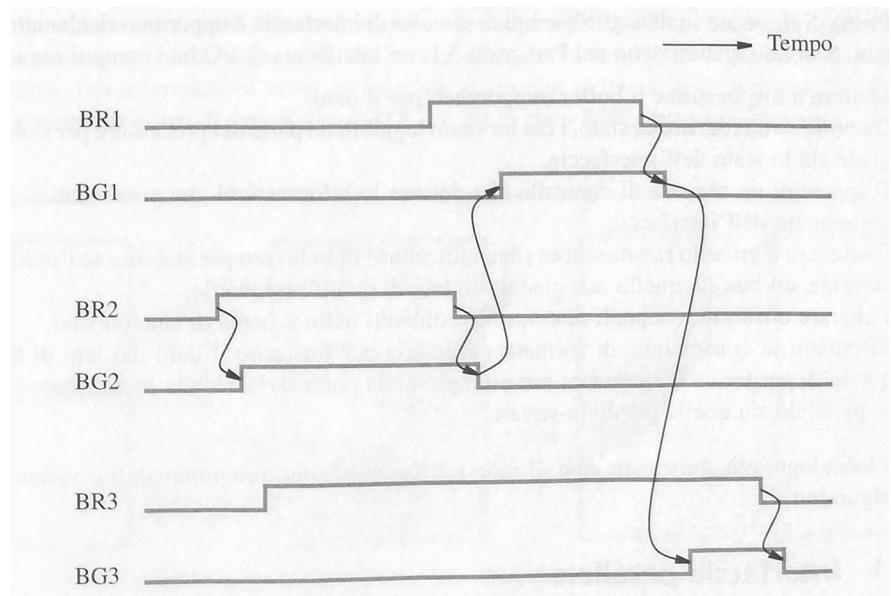


- Per gestire l'accesso al Bus simultaneo da parte di più unità master viene usato un sistema di priorità
- Il **circuito di arbitraggio** regola l'ordine di accesso al Bus nel caso di richiesta simultanea
- Le unità master inviano una richiesta di accesso al circuito di arbitraggio tramite i segnali **BR (bus request)**
- Il circuito di arbitraggio concede l'accesso al bus secondo un ordine di priorità attraverso i segnali **BG (bus granted)**



Arbitraggio - Esempio

- Esempio in cui tre unità master richiedono accesso al Bus
- Le unità sono numerate seguendo il loro ordine di priorità (Master 1 priorità più alta)
- Il master 2 richiede accesso per primo e gli viene concesso
- Mentre il Master 2 accede al Bus, i Master 1 e 3 richiedono l'accesso
- Liberato il Bus dal Master 2, il circuito di arbitraggio da accesso al Bus al Master 1, seguendo l'ordine di priorità e non di richiesta



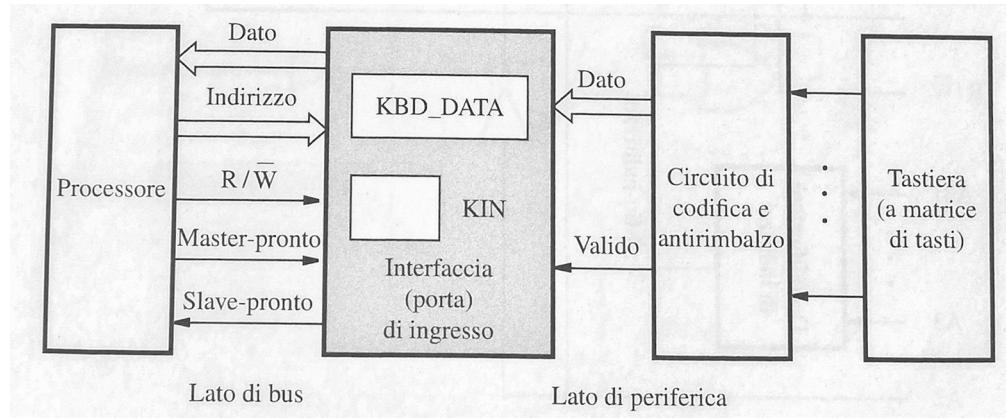
Interfacce di I/O parallele e seriali



- L'interfaccia di I/O è il circuito che collega la periferica al Bus di sistema
- L'interfaccia ha due lati:
 - **Lato di bus**
 - **Lato periferica**
- Le interfacce vengono chiamate **porte**
- Spesso le porte si trovano nel calcolatore e le periferiche si collegano ad esse tramite connettori di vario tipo (lato periferica della porta)
- Le interfacce di I/O si possono dividere in **porte parallele** e **porte seriali**

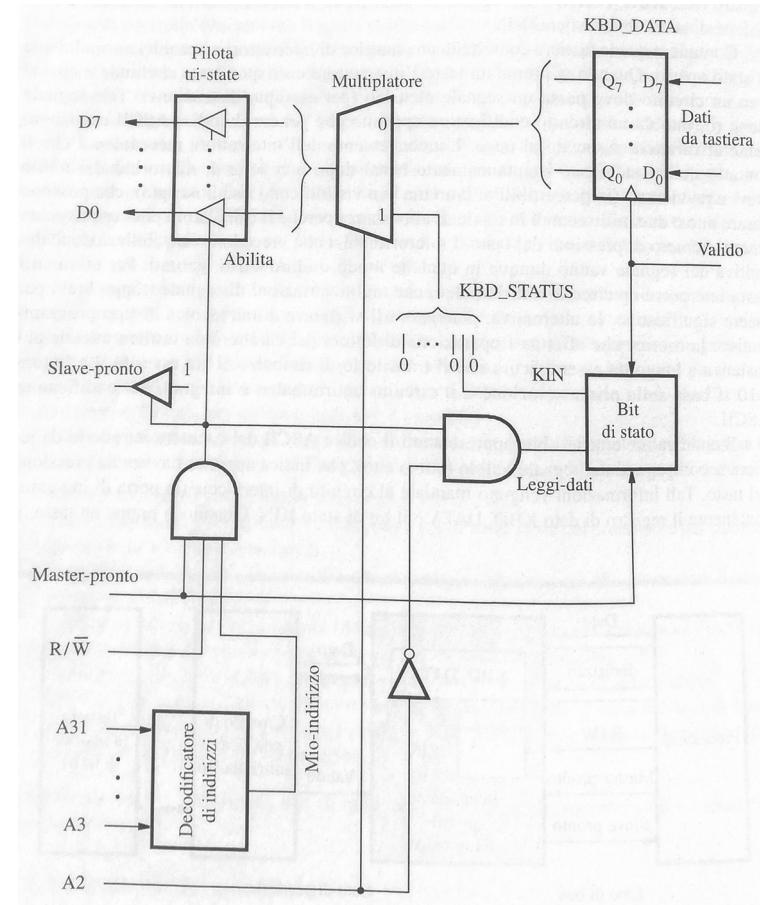
Interfaccia parallela di ingresso

- Esempio di una semplice porta parallela per una tastiera (senza interruzioni)
- Il valore di un tasto premuto viene trasmesso **in parallelo come sequenza di 8 bit**
- La porta contiene i registri **KBD_DATA** e **KBD_STATUS** (interessa solo il bit KIN)
- Bus gestito in modo asincrono
- Lato periferica riceve gli 8 bit di dato e un bit **Valido** che segnala l'avvenuta pressione di un tasto



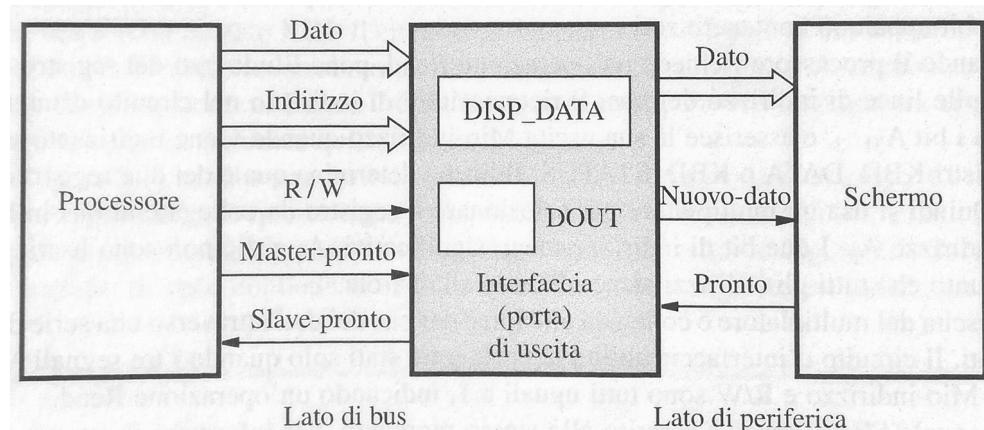
Circuito d'interfaccia parallela di ingresso

- L'indirizzo **A** ricevuto dal Bus viene riconosciuto
- **Slave-pronto** ed il **driver** vengono attivati quando Master-pronto, R/W e Mio-indirizzo sono uguali a 1
- Il driver invia sul Bus i contenuti di KBD_DATA o KBD_STATUS a seconda del valore del bit A2
- Nel caso di lettura da KBD_DATA il bit KIN viene azzerato
- Nel caso di lettura da KBD_STATUS tutti i bit ad eccezione di KIN sono azzerati



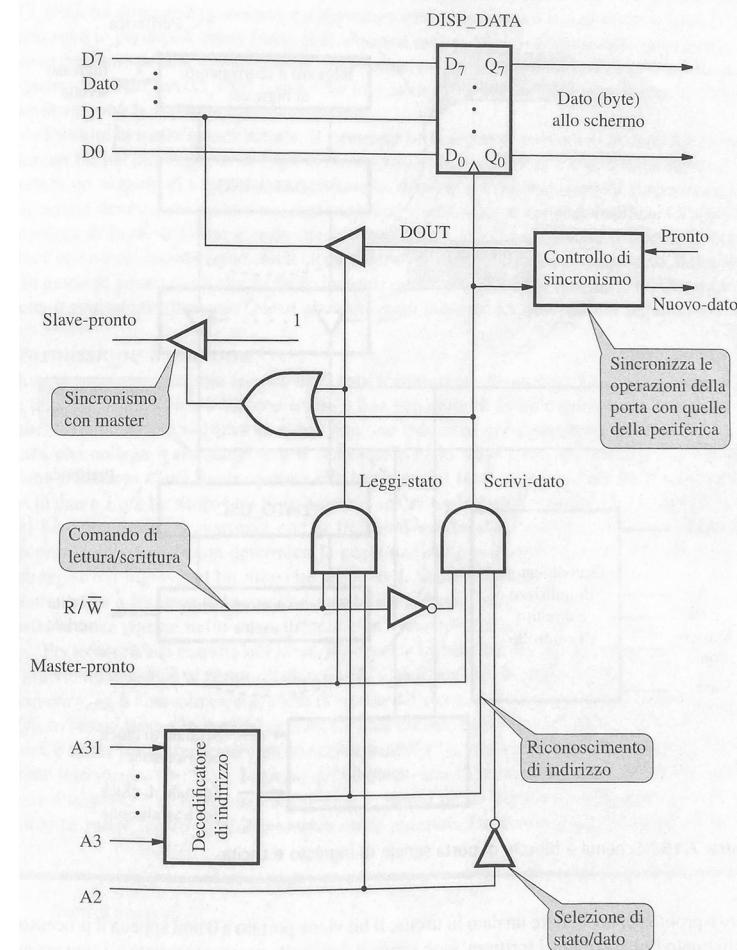
Interfaccia parallela di uscita

- Esempio di una semplice porta parallela per uno schermo (senza interruzioni)
- Il carattere da visualizzare su video viene trasmesso **in parallelo come sequenza di 8 bit**
- La porta contiene i registri **DISP_DATA** e **DISP_STATUS** (interessa solo il bit DOUT)
- Bus gestito in modo asincrono
- Lato periferica invia gli 8 bit di dato, un bit **Nuovo-dato** che segnala l'invio di un carattere da visualizzare e riceve un bit **Pronto** che avvisa quando lo schermo è pronto a visualizzare



Circuito d'interfaccia parallela di uscita

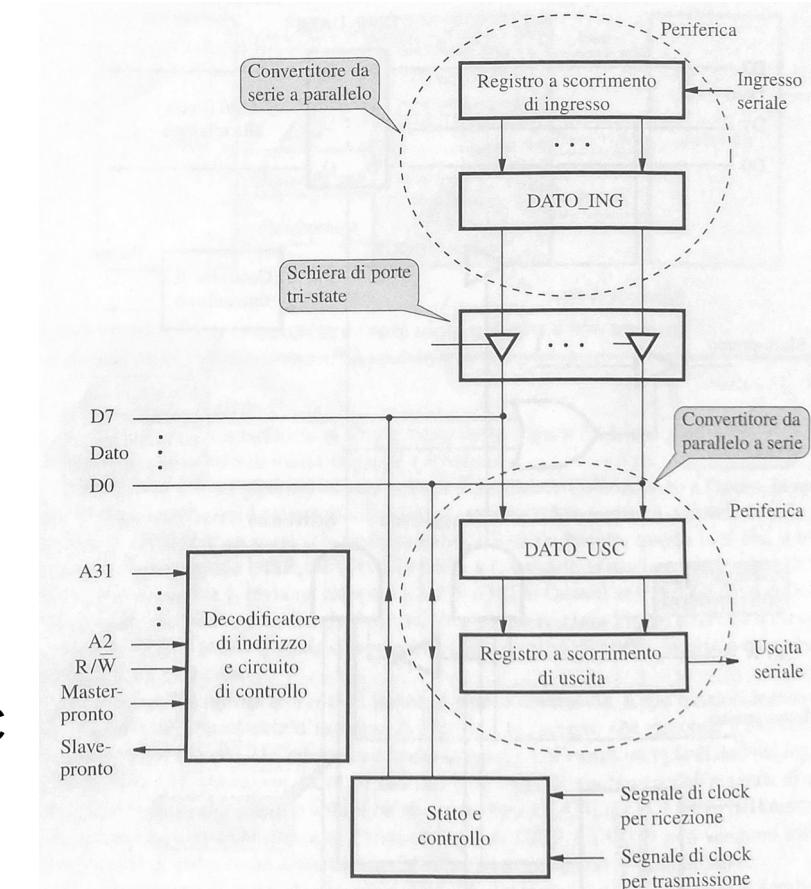
- Il circuito ha sia funzione di scrittura su DISP_DATA che di lettura da DISP_STATUS
- L'indirizzo **A** ricevuto dal Bus viene riconosciuto
- Slave-pronto** ed il **driver** del bit DOUT vengono attivati quando Master-pronto, R/W, Mio-indirizzo e A2 sono uguali a 1
- Il driver scrive i dati **D** dal Bus su DISP_DATA o inoltra sul Bus il valore di DOUT a seconda del valore del bit A2
- Il **Controllo di sincronismo** gestisce il sincronismo tra porta di uscita e schermo evitando che uno stesso carattere venga visualizzato più volte



Interfaccia seriale

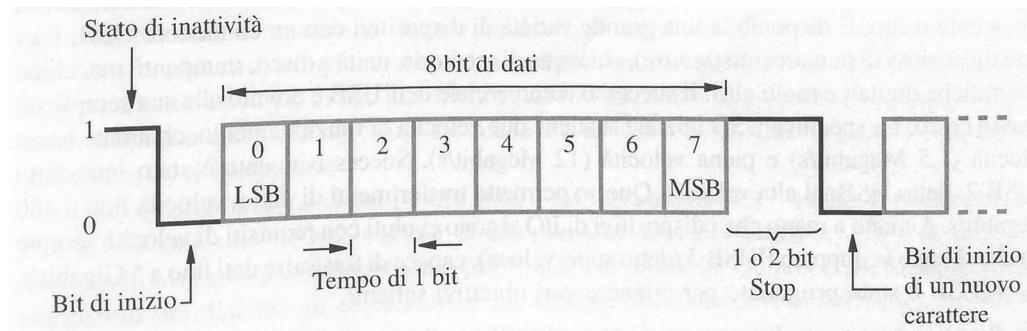
- La porta seriale collega le periferiche che scambiano i dati in forma seriale (un bit alla volta) al Bus
- Il lato bus è identico a quello delle porte parallele
- La conversione seriale-parallelo e parallelo-seriale nel lato periferica è fatta da dei registri a scorrimento
- I registri DATO_ING e DATO_USC corrispondono ai registri KBD_DATA e DISP_DATA visti in precedenza
- I registri a scorrimento permettono una lettura/scrittura in parallelo da/verso i registri DATO_ING e DATO_USC

Bisogna sincronizzare la frequenza di scorrimento dei registri con il clock delle periferiche collegate



Trasmissione asincrona

- Nella trasmissione asincrona vengono aggiunti dei bit ad inizio e fine dato per definirne i limiti:
 - Un segnale 1 indica **stato di inattività**
 - Un bit 0 viene aggiunto in testa alla sequenza di bit da inviare
 - Uno o due bit a 1 vengono aggiunti in coda alla sequenza di bit da inviare
- Il ciclo di clock del ricevente è impostato più rapido (diciamo 16 volte) di quello del trasmittente
- Un contatore viene usato dal ricevente per campionare il dato in corrispondenza del centro di ciascun bit



Trasmissione sincrona

- Nella trasmissione sincrona il ciclo di clock del ricevente viene sincronizzato con quello del trasmittente osservando i tempi di transizione da 0 e 1 e viceversa nei dati ricevuti
- Schemi di codifica particolari vengono usati per garantire che avvengano sufficienti transizioni per mantenere la sincronizzazione
- I dati codificati sono trasmessi in grandi blocchi con codici di inizio e fine blocco
- Le trasmissioni sincrone permettono frequenze di trasferimento maggiori rispetto a quelle asincrone

Standard di trasmissione

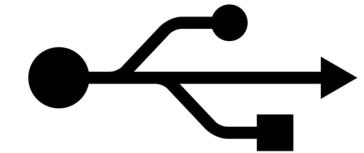
- Le interfacce I/O sono realizzate seguendo degli standard specifici:
 - **USB**
 - **Firewire**
 - **PCI**
 - ...
- I dispositivi con connettori di un certo standard possono essere collegati a tutti i calcolatori aventi delle porte dello stesso standard
- Con questo approccio si evita di dover realizzare delle interfacce hardware specifiche per ciascun dispositivo da collegare al calcolatore
- Gli standard sono solitamente realizzati da aziende o dall'**IEEE (Institute of Electrical and Electronics Engineers)**

Caratteristiche dei dispositivi

- I dispositivi di I/O esistenti possiedono vincoli di velocità, volume di dati e sincronizzazione molto differenti
- Tastiere, mouse e joypad hanno **frequenza bassa**, i dati sono **asincroni** e richiedono una **risposta rapida** da parte del calcolatore
- Microfoni e altre periferiche audio inviano un flusso continuo di campioni digitalizzati ad **intervalli regolari** (isocrono). La frequenza di campionamento è **mediamente veloce** (~1.4 Megabit/s)
- Videocamere e webcam richiedono frequenze di trasferimento molto alte (~44 Megabit/s)

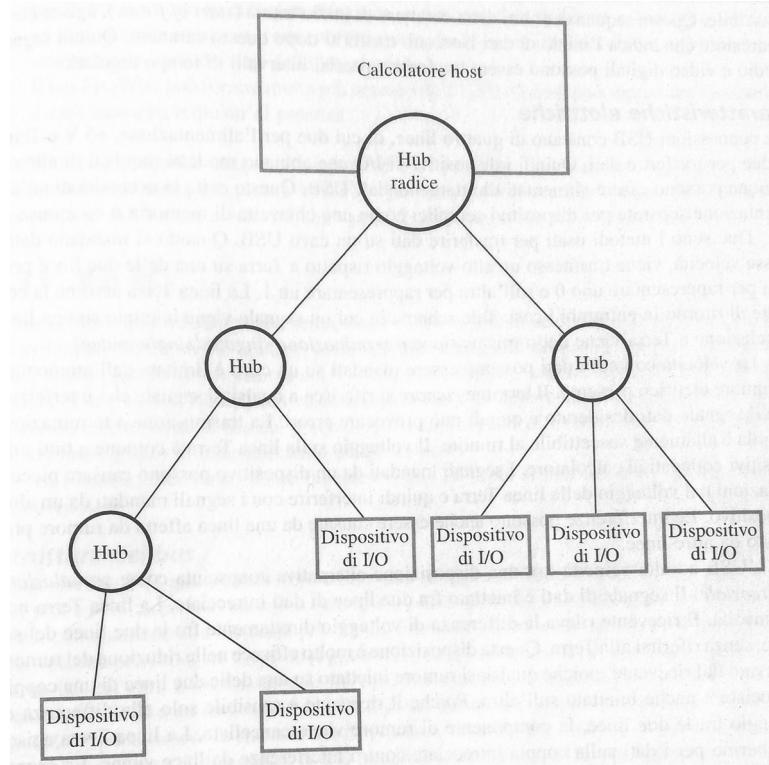
Universal serial bus (USB)

- **L'Universal Serial Bus (USB)** è uno standard di interconnessione largamente usato:
 - Mouse, tastiere e joypad
 - Chiavette di memoria e unità disco
 - Stampanti e scanner
 - Macchine fotografiche e videocamere
 - Etc.
- Negli anni sono state sviluppate diverse versioni:
 - **USB 1** (1.5/12 Mbps)
 - **USB 2** (480 Mbps)
 - **USB 3** (5 Gbps)
 - **USB 4** (40 Gbps)



Struttura USB

- Trasmissione seriale
- Connessioni punto-punto
- Struttura ad albero:
 - **Foglie:** Dispositivi
 - **Nodi:** Hub (punti trasferimento intermedi)
- Trasmissione dati tramite Polling (in risposta ad un'interrogazione del processore)
- Hub radice appare come singolo dispositivo al processore



Caratteristiche elettriche

- Connessioni USB constano di 4 linee:
 - 2 linee di alimentazione (Terra e +5 V)
 - 2 linee dati
- I dispositivi con consumi modesti sono alimentati direttamente dall'USB
- Le 2 linee dati possono avere due configurazioni differenti:
 - **Single-ended:** Basse velocità. Ciascuna linea rappresenta uno stato (0/1). Dati trasmessi mandando un alto voltaggio rispetto a Terra
 - **Segnalazione differenziale:** Alte velocità. Linee intrecciate. Stato del segnale sulla base della differenza di voltaggio tra le due linee. Robusto al rumore

USB plug-and-play

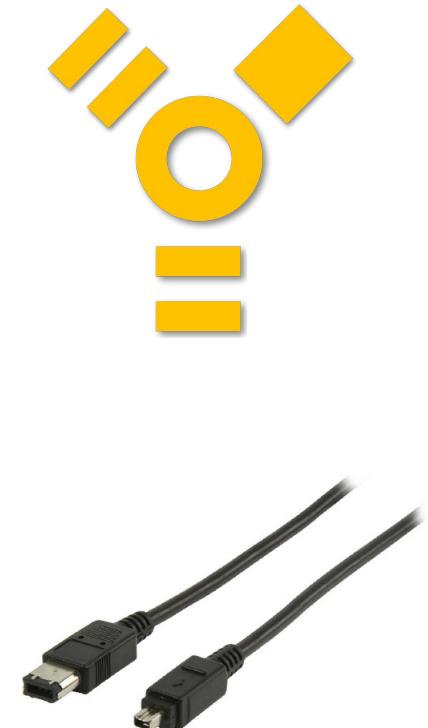
- Il **Plug-and-Play** è una funzionalità che permette di impostare automaticamente un dispositivo appena viene collegato
- Il sistema operativo dovrà conoscere il **tipo di dispositivo** connesso ed i **suoi indirizzi**
- Il calcolatore (**Host**) interroga periodicamente gli hub per apprendere di nuovi dispositivi connessi
- Al momento del collegamento di un dispositivo:
 - L'indirizzo del dispositivo è 0
 - L'host legge le informazioni in una memoria speciale del dispositivo
 - L'host assegna al dispositivo un indirizzo USB univoco e lo scrive in un registro dell'interfaccia del dispositivo

Traffico isocrono su USB

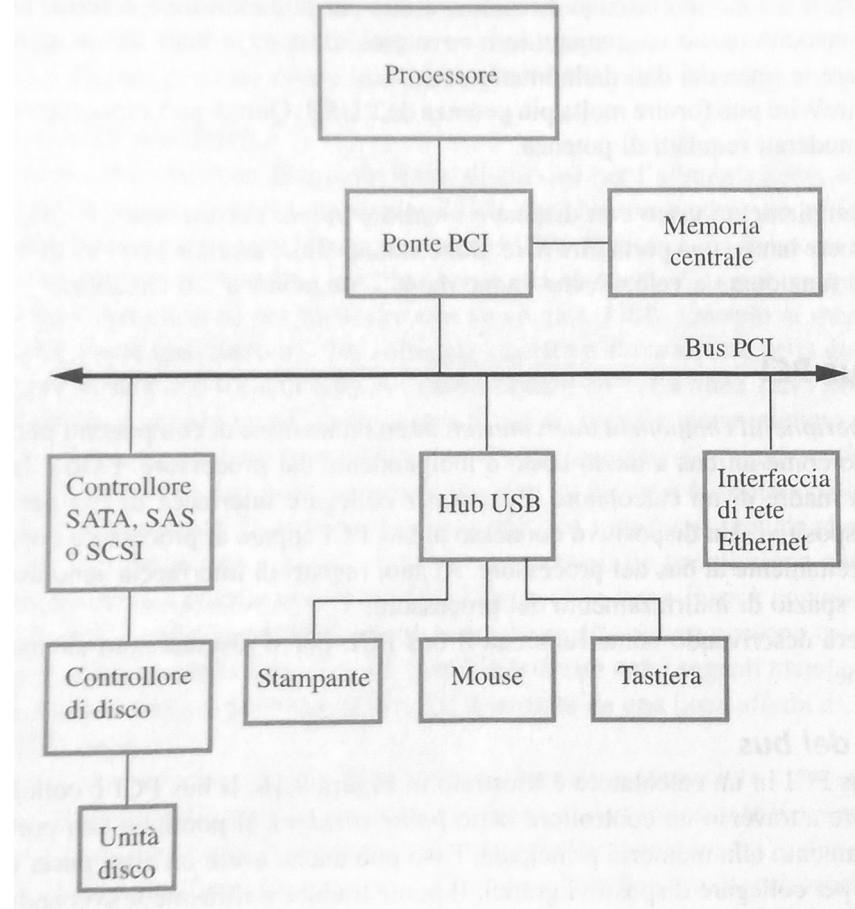


- USB permette un trasferimento isocrono per la gestione di dati audio e video
- I dati isocroni devono essere trasferiti con precisione ad intervalli di tempo regolari
- Hub radice trasmette ogni millisecondo sull'albero USB una sequenza di bit univoca (**start of frame**)
- La sequenza di bit **start of frame** segna l'inizio dei dati asincroni

- Il Firewire (IEEE 1394) è uno standard di connessione usato principalmente per dispositivi audio e video
- Caratteristiche:**
 - Collegamenti seriali punto-punto differenziali
 - Struttura del collegamento a festone (**daisy chain**)
 - Ottimizzato per traffico isocrono ad alta velocità
 - Permette trasferimenti **peer-to-peer** tra dispositivi
 - Connettore con 6 piedini: due coppie di linee per i dati, alimentazione e Terra
 - Può alimentare dispositivi aventi moderati requisiti di potenza
 - Velocità di trasferimento che va da 400 Mbps a 3.6 Gbps



- Il bus PCI (peripheral component interconnect) è incorporato nella scheda madre di un calcolatore
- I dispositivi di I/O si collegano al bus PCI attraverso porte che usano standard come Ethernet, USB, SATA, etc.
- Il bus PCI è connesso a processore e Memoria attraverso un controllore detto **Bridge**
- Il trasferimento dati avviene in gruppi di parole detti **raffiche (burst)**
- Nei trasferimenti dati il dispositivo di I/O ha il ruolo di master connettendosi direttamente dalla memoria



Trasferimento dati PCI

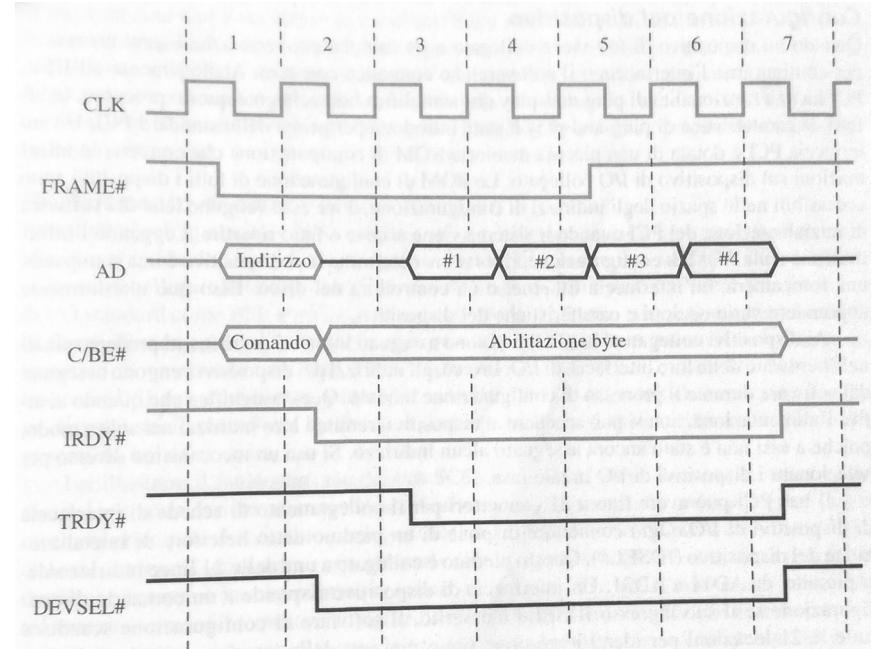
- Un'operazione PCI di trasferimento dati completa è chiamata **transazione**
- Il master del bus è chiamato **iniziatore (initiator)**
- Lo slave del bus è chiamato **obiettivo (target)**
- Il trasferimento di dati avviene con uno schema di segnalazione sincrona multiciclo
- In tabella i principali segnali PCI

Nome	Funzione
CLK	Un clock a 33 MHz o 66 MHz
FRAME#	Inviato dall'iniziatore per indicare la durata di una trasmissione
AD	32 linee di indirizzi/dati, che possono essere optionalmente incrementate a 64
C/B/E#	4 linee per comando/abilità-byte (8 per un bus a 64 bit)
IRDY#, TRDY#	Segnali Iniziatore-pronto e Obiettivo-pronto
DEVSEL#	Una risposta dal dispositivo che indica che ha riconosciuto il suo indirizzo ed è pronto per una transazione di trasferimento di dati
IDSEL#	Selezione di dispositivo da inizializzare

Trasferimento dati PCI (esempio)

Esempio di transazione read con un burst di 4 parole a 32 bit

1. Nel ciclo 1 l'iniziatore asserisce **FRAME#**, invia l'indirizzo nelle linee **AD** e il comando nelle linee **C/B/E#**
2. Nel ciclo 2 l'iniziatore rimuove l'indirizzo, si disconnette dalle linee AD e asserisce **IRDY#**. L'obiettivo asserisce **DEVSEL#**, si connette alle linee **AD** e imposta le linee attive usando i segnali **C/B/E#**
3. Nel ciclo 3 l'obiettivo asserisce **TRDY#** e inizia il trasferimento della prima parola
4. Nel ciclo 5 (ricevuta la penultima parola) l'iniziatore disattiva **FRAME#**
5. Nel ciclo 7 l'obiettivo disattiva **TRDY#** e **DEVSEL#** e sconnette i suoi piloti dalle linee **AD**

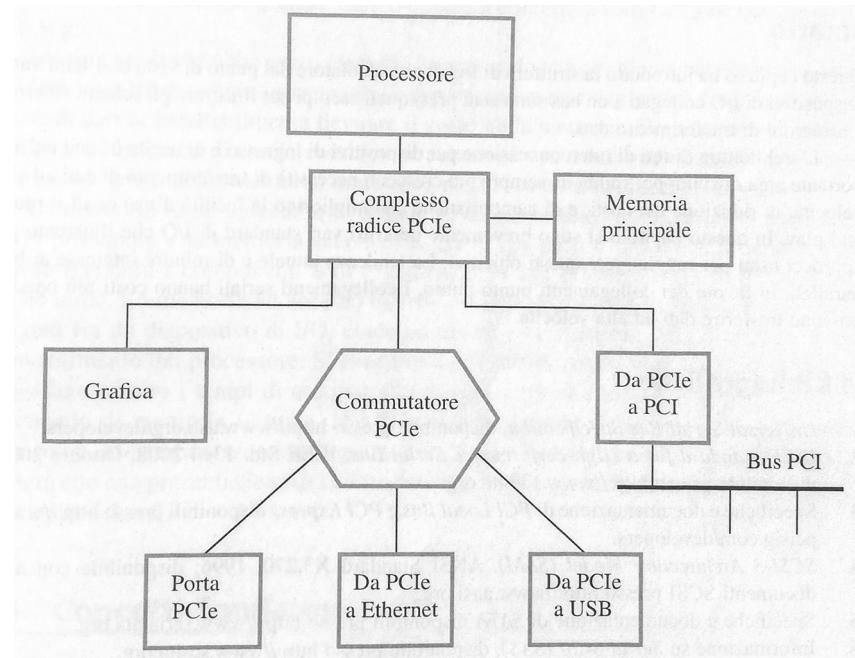


Configurazione dispositivi su PCI

- Anche il bus PCI ha la funzionalità plug-and-play
- I dispositivi PCI possiedono una piccola memoria ROM di configurazione contenente le informazioni sul dispositivo
- I bus PCI possono avere fino a 21 connettori
- Ogni connettore ha un piedino (IDSEL#) collegato ad una delle 21 linee AD più alte (AD11-AD31)
- All'avvio il calcolatore invia un comando di configurazione ad ogni locazione con un indirizzo in cui i primi 11 bit (AD0-AD10) sono usati per indirizzare la ROM e gli ultimi 21 (AD11-AD31) sono posti disattivati tranne il bit IDSEL# della locazione
- Se un dispositivo risponde, gli si assegna un indirizzo memorizzato in un suo registro

PCI Express (PCIe)

- Il PCI Express PCIe usa collegamenti seriali punto-punto
- Nodi commutatore vengono usati per avere una struttura a albero
- Il Complesso radice ha un collegamento diretto speciale al processore e alla memoria principale
- Si possono collegare interfacce ponte per altri standard
- Un collegamento base PCIe (corsia) è formato da 2 coppie di linee intrecciate con frequenza di 2.5 Gbps ciascuna
- Per avere trasferimenti più veloci si possono usare più corsie (X2, X3, X4, X8 o X16)



- Il bus SCSI (Small Computer System Interface) è uno standard usato principalmente per collegare unità a dischi al calcolatore
- I dispositivi sono collegati tramite un cavo a 50 linee
- I dati sono trasferiti a gruppi di 8 o 16 bit in parallelo ad una frequenza di 80 MHz
- I dati vengono trasferiti in forma di messaggi a più byte detti pacchetti
- Per trasferire dati tra processore e dispositivi i pacchetti vengono prima memorizzati in memoria e poi letti dal richiedente
- Esiste una versione seriale del bus SCSI chiamata SAS (Serial attached SCSI) con velocità di trasferimento fino a 12 Gbps

- SATA (Serial AT Attachment) è uno standard di connessione seriale punto-punto usata per connettere unità a disco
- Presenta un connettore a 7 piedini: 2 coppie intrecciate per i dati e tre linee Terra
- Frequenze di trasferimento vanno da 1.5 a 6.0 Gbps
- Le versioni moderne offrono trasmissione isocrona per supporto audio e video