

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

## Interfacce di I/O

*Nino Cauli*

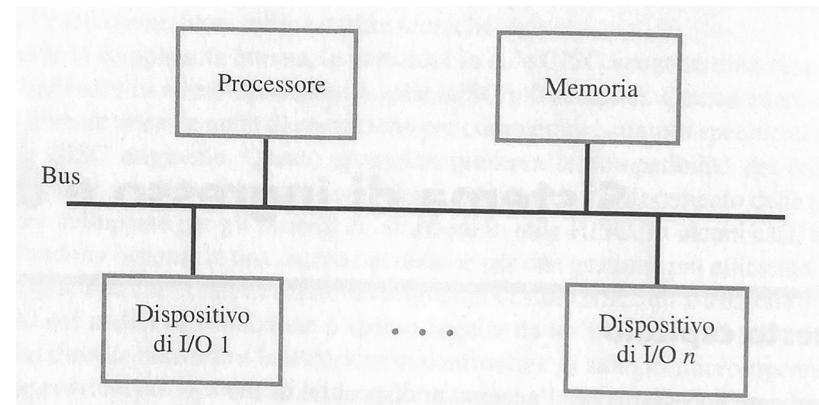


UNIVERSITÀ  
degli STUDI  
di CATANIA

Dipartimento di Matematica e Informatica

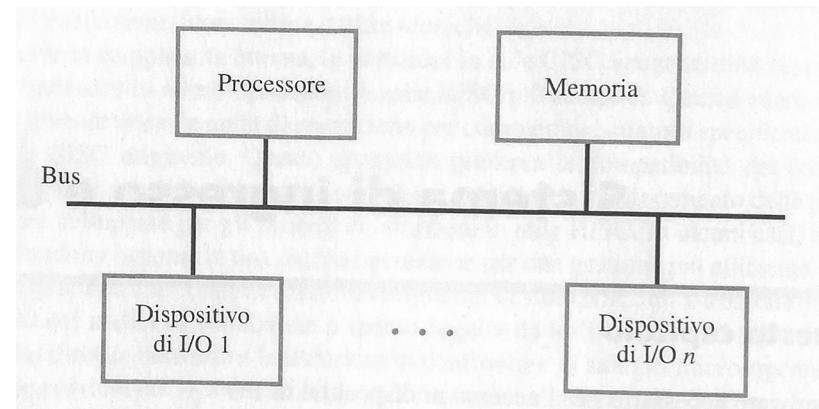
# Bus di sistema

- 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**



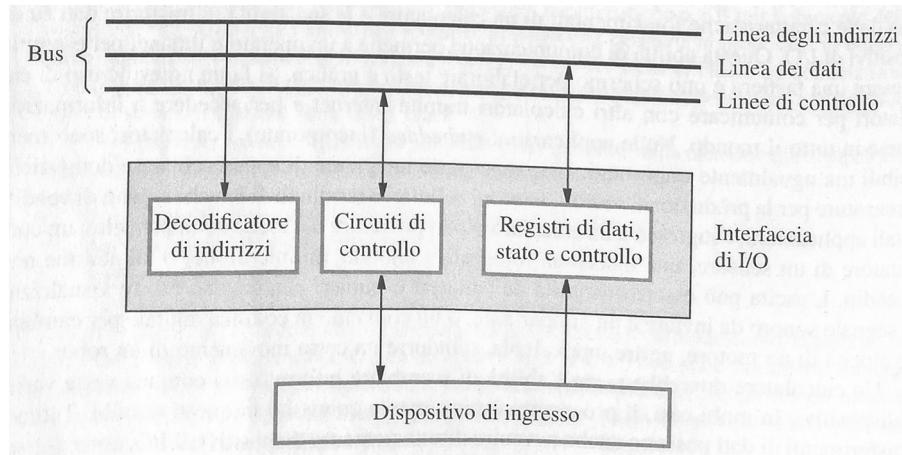
# Bus di sistema

- 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

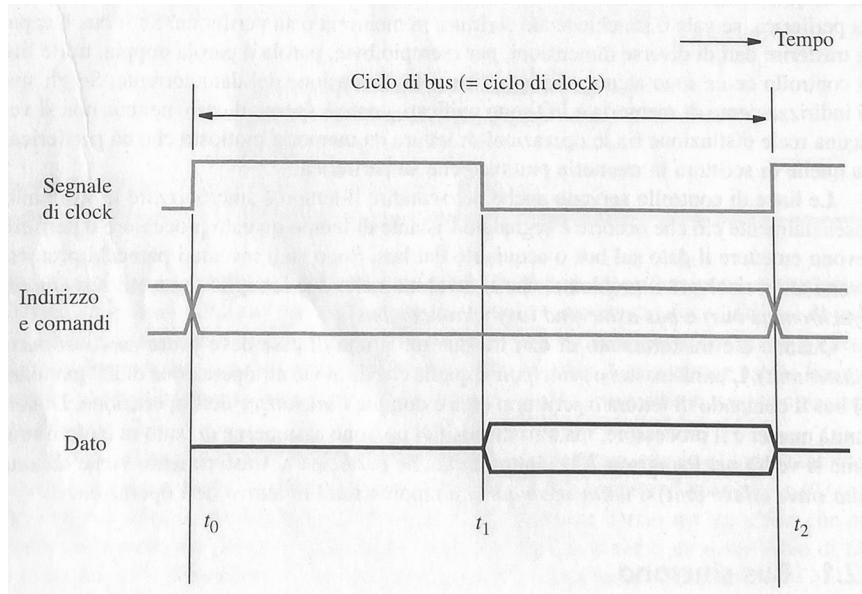
- I dispositivi di I/O sono collegati al bus attraverso le **Interfacce di I/O**
- Le interfacce di I/O contengono:
  - **Decodificatore di indirizzi**: per sapere se rispondere alla richiesta
  - **Circuiti di controllo**: per sapere che tipo di operazione eseguire
  - **Registri di dati, stato e controllo**: 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

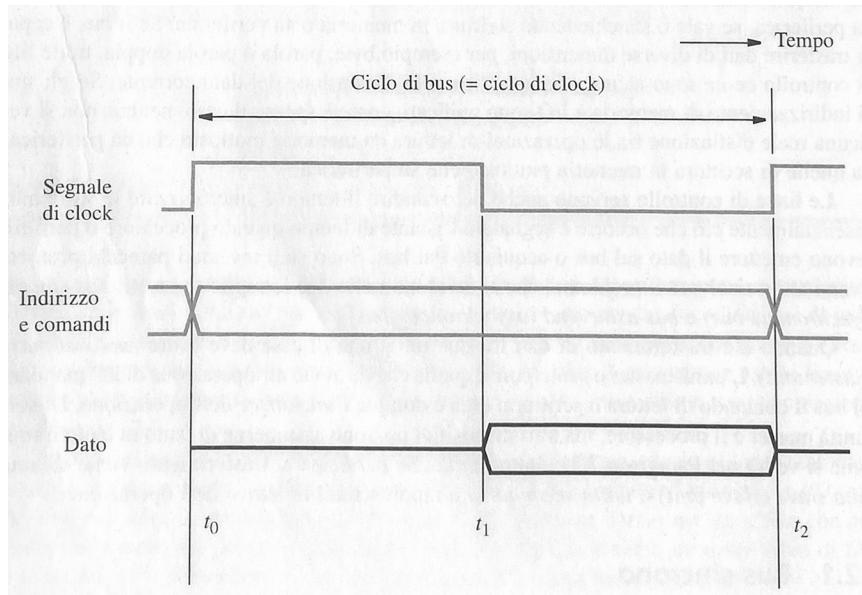
# Diagrammi temporali

- 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



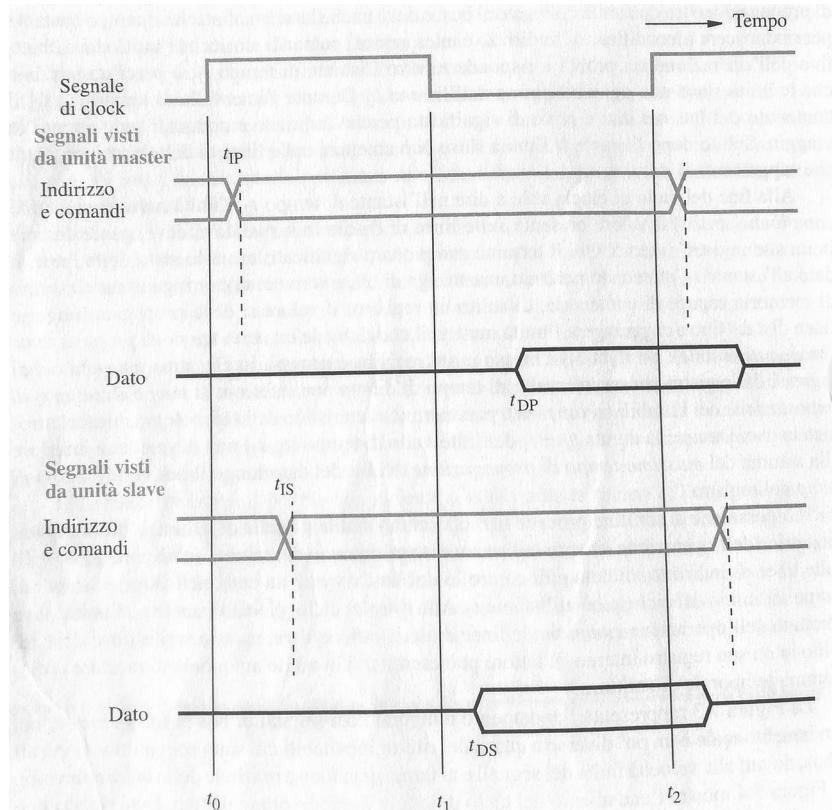
# Bus sincrono semplice (lettura)

- 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



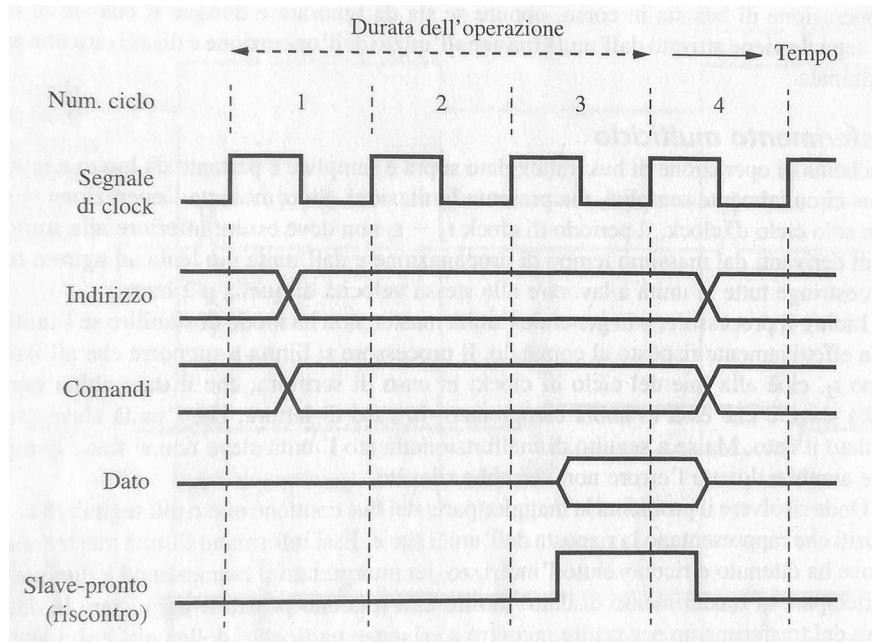
# Bus sincrono semplice (lettura) - Ritardi

- 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



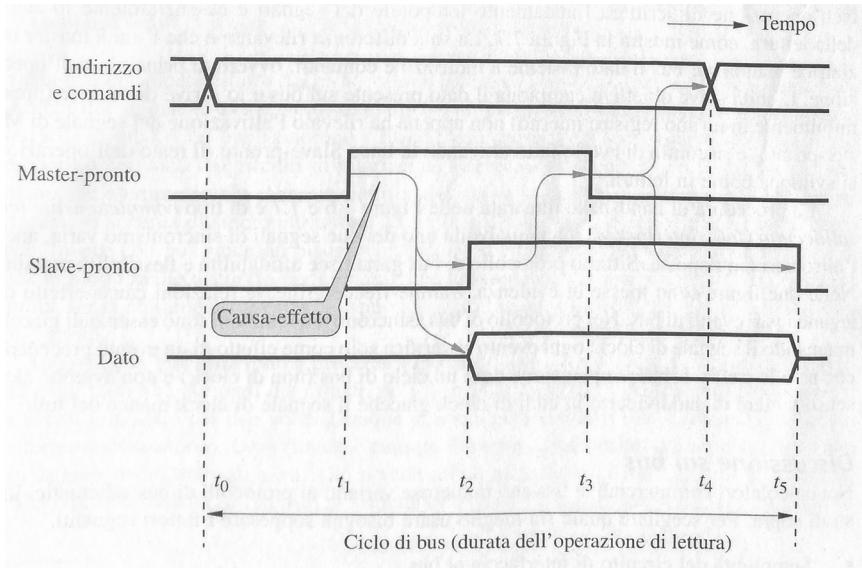
# Bus sincrono multiciclo (lettura)

- 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



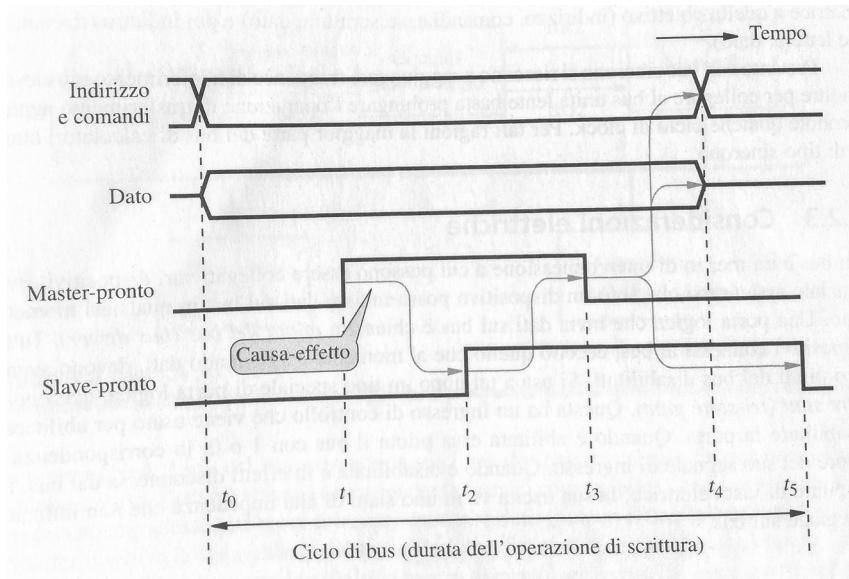
# Bus asincrono (lettura)

- 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

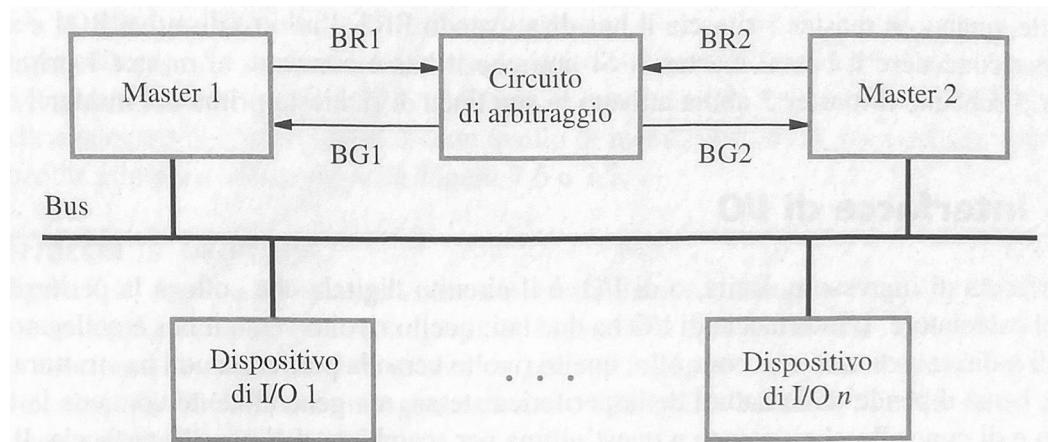


# Bus asincrono (scrittura)

- 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

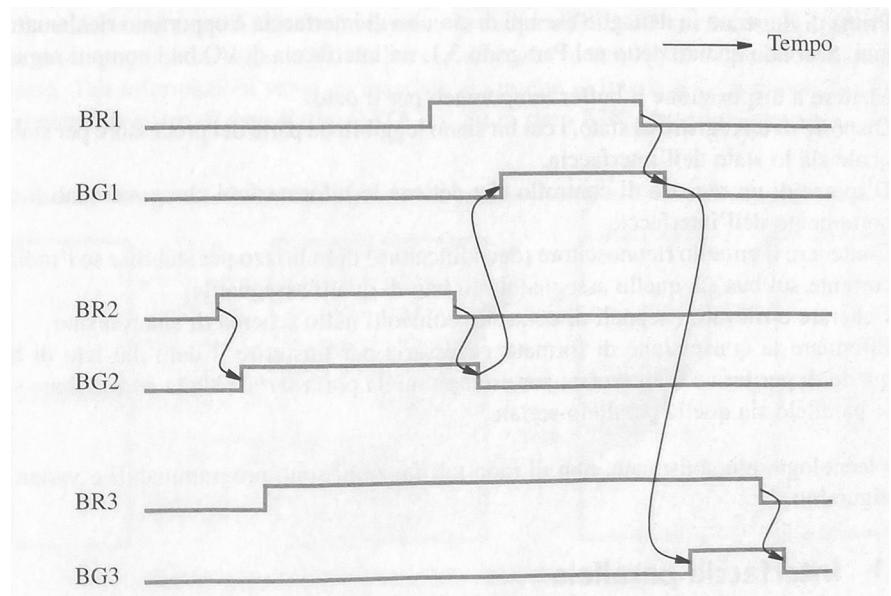


- Per gestire l'accesso al Bus simultaneo da parte di più unità master viene usato un sistema di priorità
- Il **circuito di arbitraggio** regola l'ordine di accesso al Bus nel caso di richiesta simultanea
- Le unità master inviano una richiesta di accesso al circuito di arbitraggio tramite i segnali **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



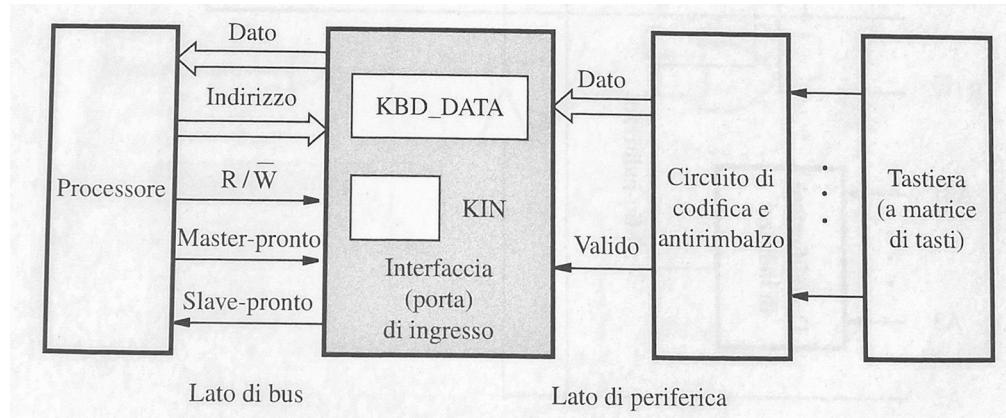
# Interfacce di I/O parallele e seriali



- 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**

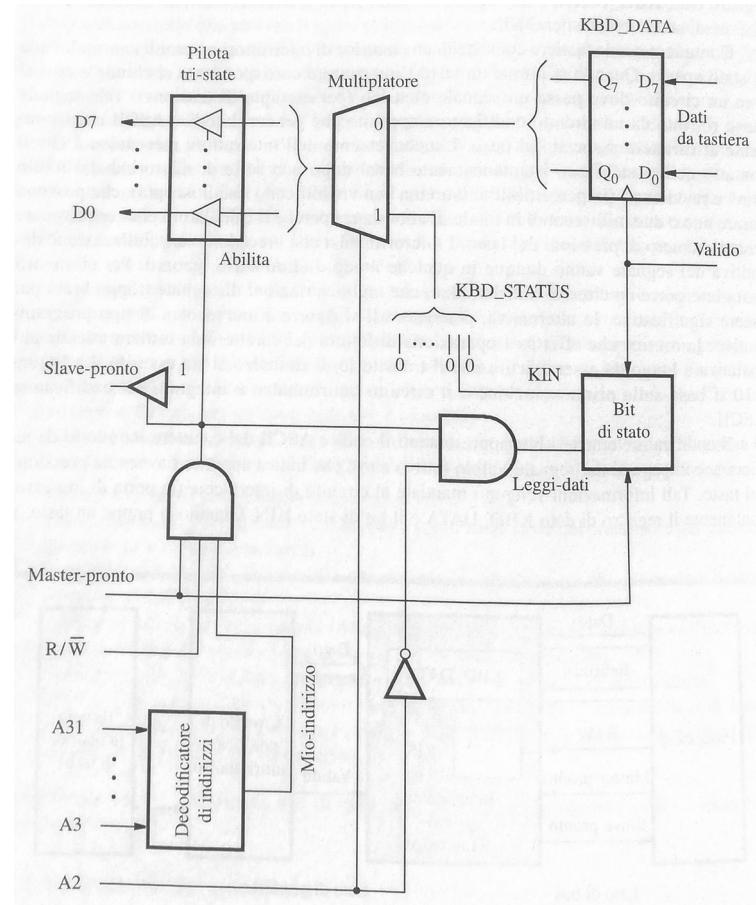
# Interfaccia parallela di ingresso

- 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



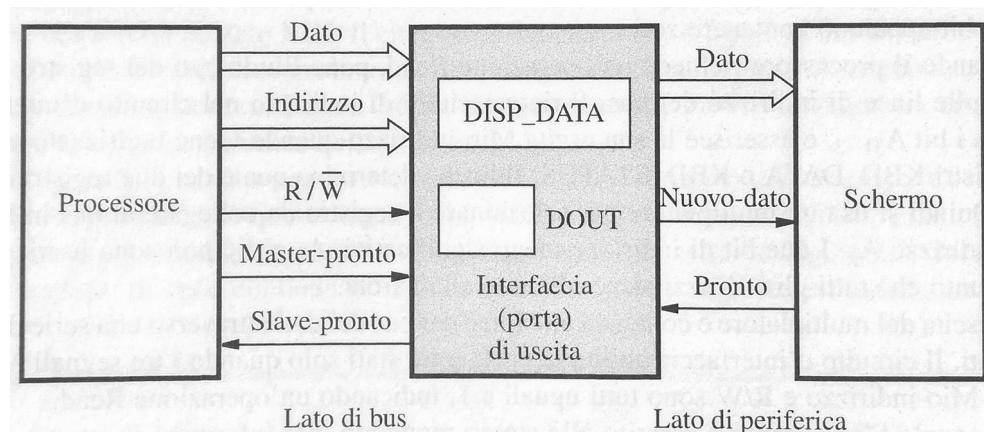
# Circuito d'interfaccia parallela di ingresso

- 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



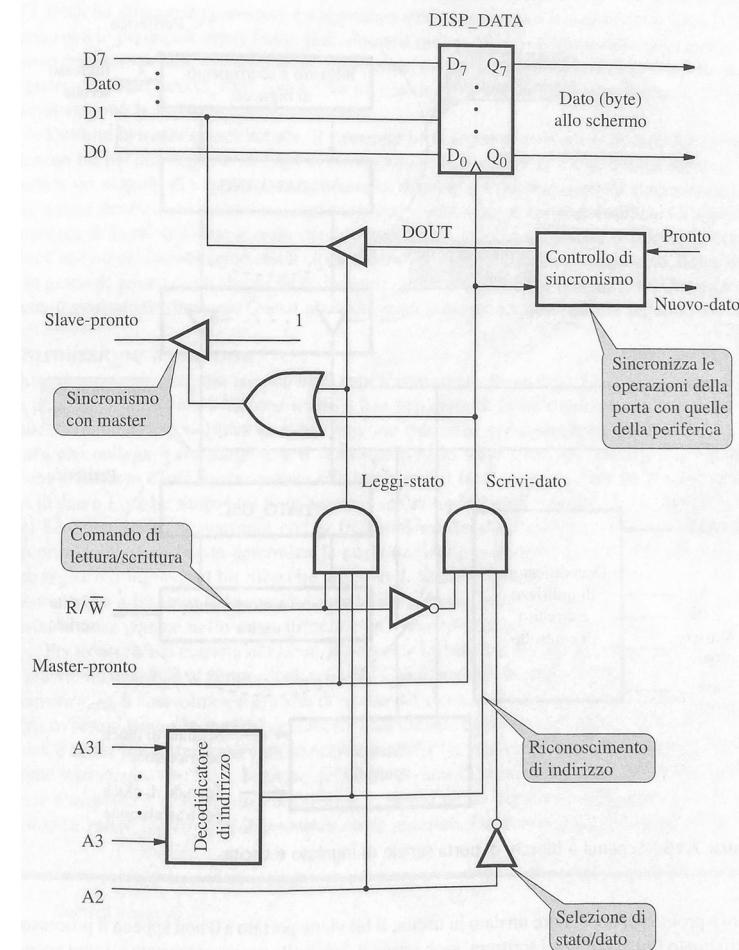
# Interfaccia parallela di uscita

- 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

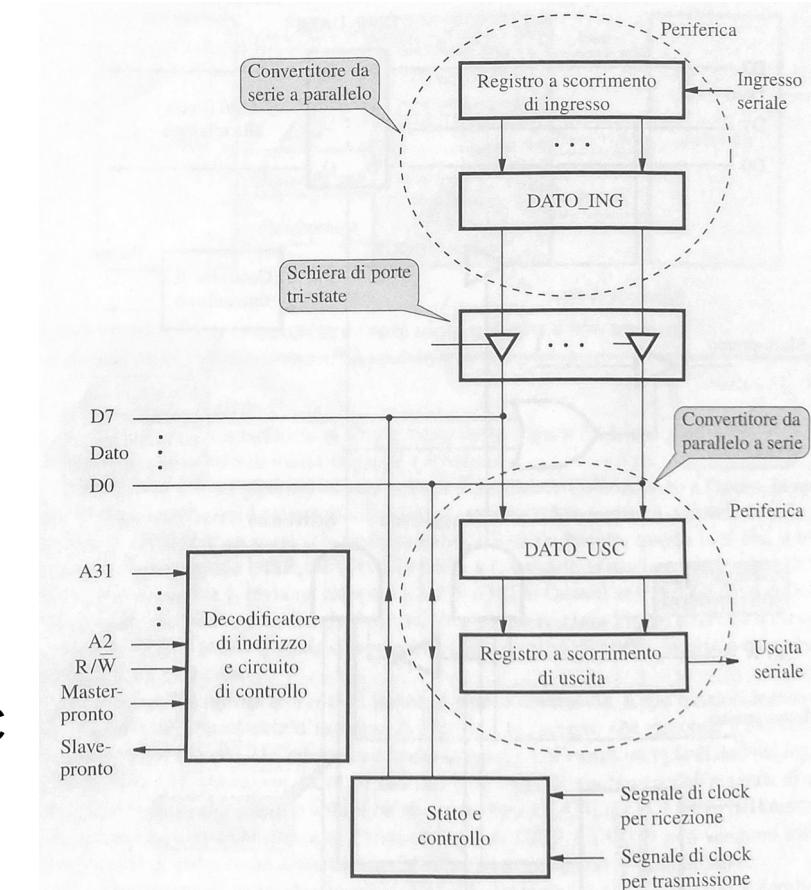
- Il circuito ha sia funzione di scrittura su DISP\_DATA che di lettura da DISP\_STATUS
- L'indirizzo **A** ricevuto dal Bus viene riconosciuto
- Slave-pronto** ed il **driver** del bit DOUT vengono attivati quando Master-pronto, R/W, Mio-indirizzo e A2 sono uguali a 1
- Il driver scrive i dati **D** 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

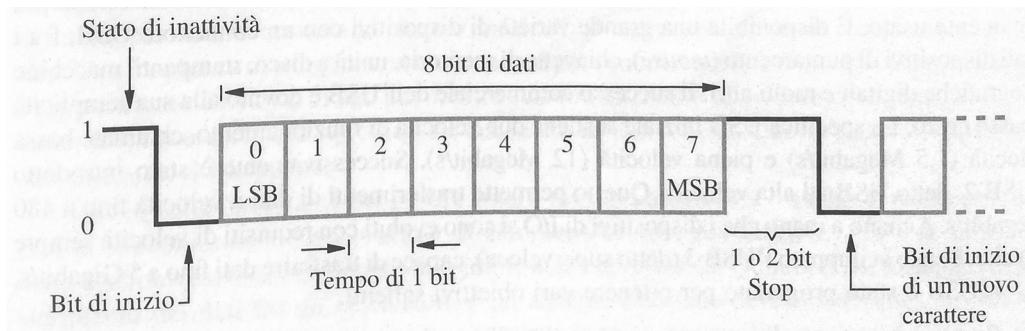
- La porta seriale collega le periferiche che scambiano i dati in forma seriale (un bit alla volta) al Bus
- Il lato bus è identico a quello delle porte parallele
- La conversione seriale-parallelo e parallelo-seriale nel lato periferica è fatta da dei registri a scorrimento
- I registri DATO\_ING e 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