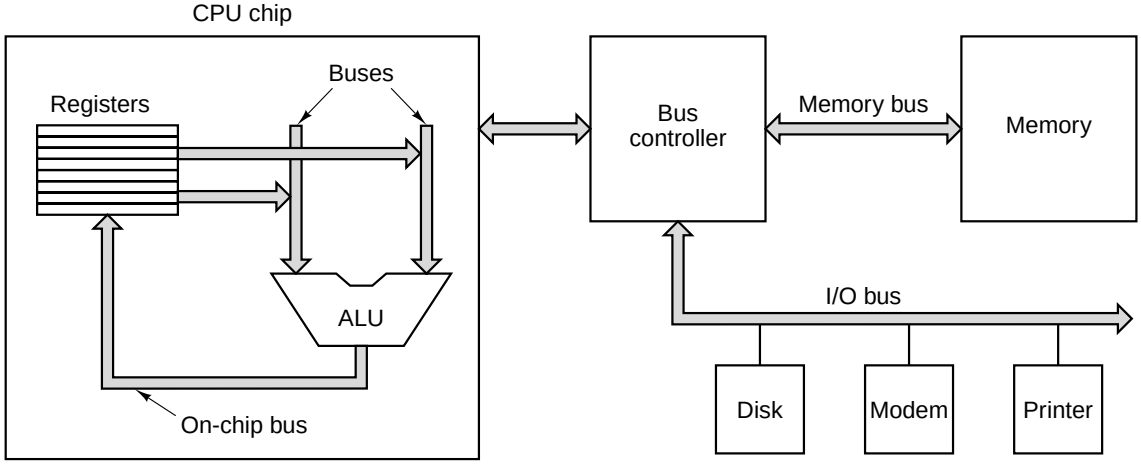
Strutture di Interconnessione in un calcolatore

- Introduzione ai Bus e ruoli dei dispositivi.
- Linee dei Bus, Ampiezza.
- Temporizzazione: Bus sincroni e asincroni.
  - Un esemepio di read da memoria
- Arbitraggio: centralizzato e decentralizzato.
- Operazioni dei Bus.
- Un cenno al Bus PCI.
- Tendenze attuali: bus seriale con connessione punto a punto.

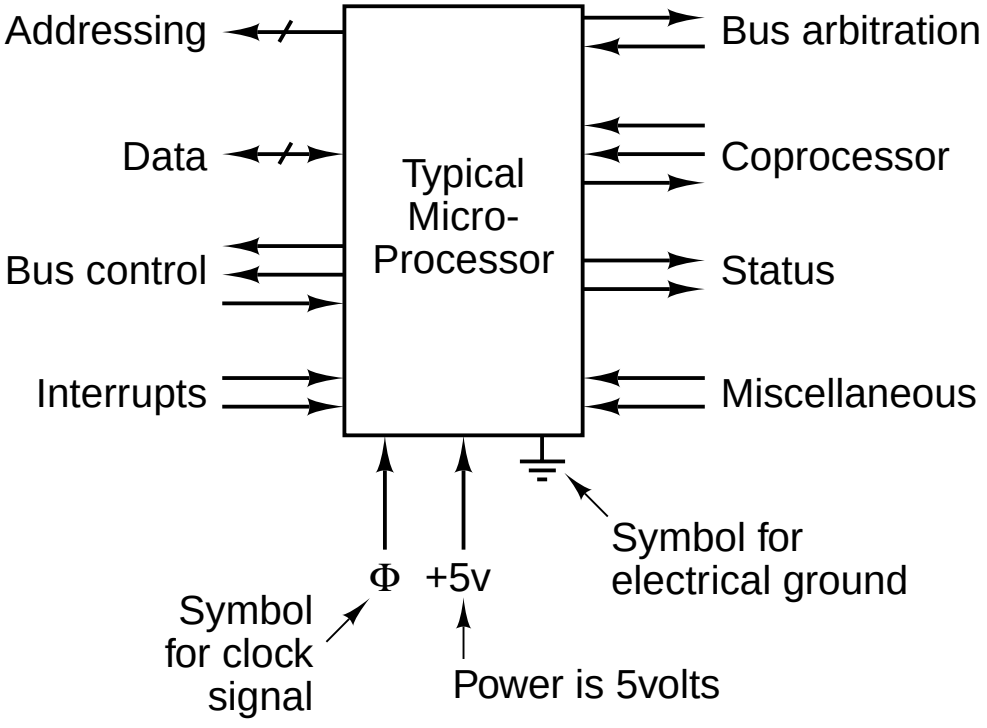
Chip per la CPU

- Tutte le CPU moderne sono contenunte in un unico chip.
- I pin di una CPU possono essere divisi in tre categorie:
  - Pin per gli **indirizzi**.
  - Pin per i **dati**.
  - Pin di **controllo**.
- La comunicazione tra la CPU e la memoria avviene attraverso il bus e coinvolge i pin per gli indirizzi, i dati e i pin di controllo.
- Il numero m dei pin per gli indirizzi e il numero n dei pin per i dati sono due parametri fondamentali di una CPU.
- I pin di controllo possono essere (approssimativamente) raggruppati come segue:
  - **Controllo del Bus**.
  - **Interrupt**.
  - **Arbitraggio del Bus**.
  - **Comunicazione con il Coprocessore**.
  - **Stato**.
  - **Altro**.

Sistema di Calcolo con Più Bus



Tipico Chip per la CPU



Bus Del Calcolatore

- Un **bus** è un collegamento elettrico costituito da una sequenza di fili paralleli, che serve a far comunicare componenti diverse di un sistema di calcolo.
- In questa parte del corso ci occuperemo dei bus che connettono la CPU con la memoria e con i dispositivi di I/O.
  - Esistono anche bus interni alla CPU. Ne parleremo in una parte successiva del corso.
- Nei primi PC vi era un unico **bus di sistema**, mentre oggi c'è:
  - Almeno un bus per la comunicazione tra CPU e memoria.
  - Almeno un bus per la comunicazione tra CPU e dispositivi di I/O.
- Spesso ai bus esterni alla CPU possono essere connessi dispositivi diversi. Per questo occorre stabilire:
  - Un'insieme di regole di funzionamento, detto **protocollo del bus**.
  - Un'insieme di **specifiche meccaniche ed elettriche**.

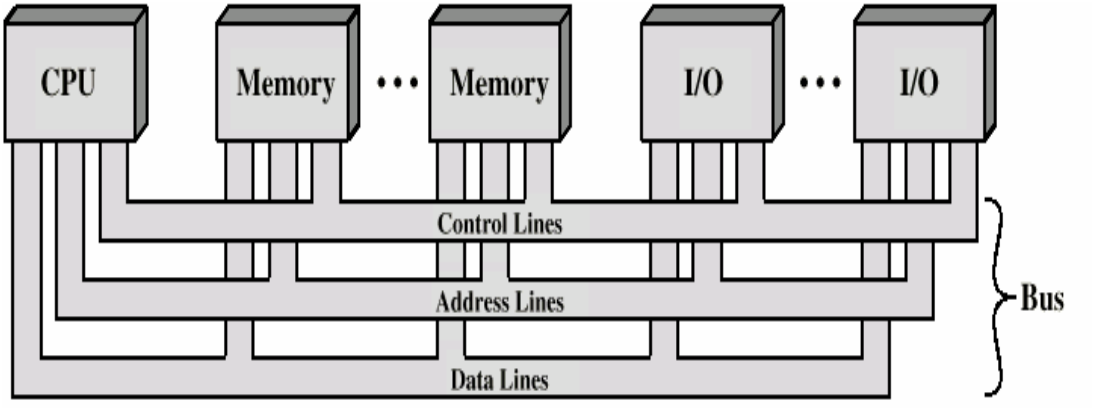
Bus del calcolatore: ruoli

Tra i componenti che sono connessi ad un Bus distinguiamo:

- i componenti che sono attivi e possono iniziare un trasferimento dati (chiamati **master**)
- i componenti che sono passivi e restano in attesa di una richiesta di trasferimento dati (chiamati **slave**).
    - alcuni componenti possono svolgere entrambi i ruoli.

- Se il Bus è lungo il segnale si può indebolire eccessivamente. Per ovviare a tale problema spesso ciascun master può essere conneso al Bus mediante un **amplificatore digitale** detto **driver del Bus**.
- Analogamente per gli slave, connessi al Bus tramite un **ricevitore del Bus**.

Bus del calcolatore: linee del bus



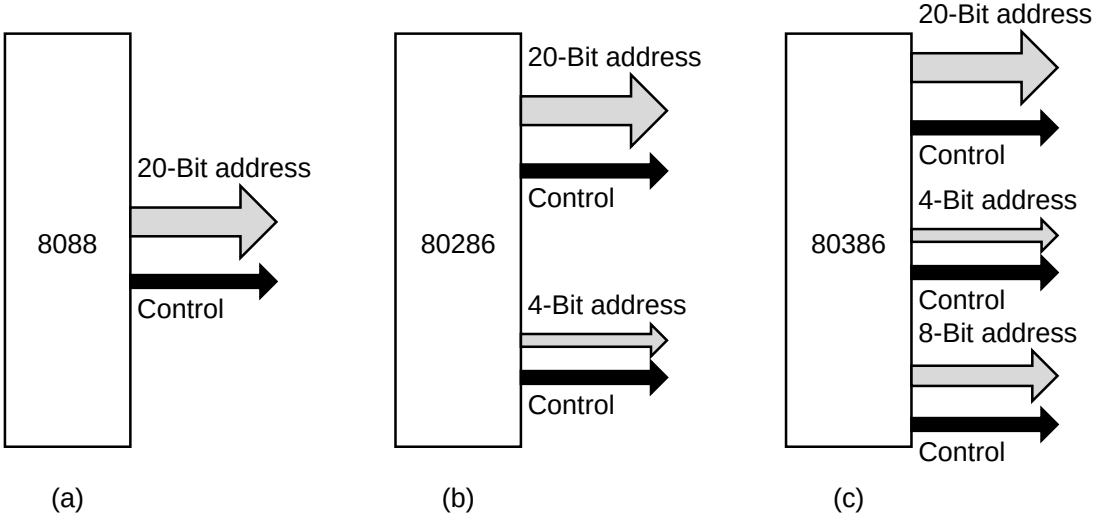
– **Bus dati**: usato per trasferire dati fra CPU e memoria o interfacce di I/O. La sua ampiezza dovrebbe essere quella della parola di memoria.
– **Bus indirizzi**: usato per trasferire l'indirizzo della cella di memoria in cui la CPU v a scrivere/leggere. La sua ampiezza dipenderà dallo spazio di indirizzamento.
– **Bus di controllo**: dove passano i segnali di controllo che indicano il tipo di operazione da effettuare. Esempio, supponiamo vi siano due linee:

- R/W: indica (se alto) un'operazione di lettura (READ) oppure (se basso) di scrittura (WRITE).
- M/IO: distingue tra operazione con la memoria (alto) o con un'interfaccia di I/O (basso).

Ampiezza del Bus

- Un certo numero di fili tra quelli componenti un bus sarà dedicato agli **indirizzi**.
- Se un bus ha n linee d'indirizzi, la CPU potrà indirizzare 2^n locazioni di memoria.
- Occorre trovare un compromesso tra l'esigenza di indirizzare un grande numero di locazioni di memoria e l'esigenza di non occupare troppo spazio.
- Come le linee d'indirizzi, anche le linee di dati tendono a crescere di numero, aumentando quindi la **larghezza di banda** dei dati sul bus.
- Anche la velocità del bus può essere aumentata, ma non oltre un certo limite.
  - I dati su linee distinte viaggiano a velocità leggermente distinte (**disallineamento del bus**).
- Per aggirare il problema di bus troppo ampi, i progettisti optano a volte per un **bus multiplexato**: le stesse linee vengono utilizzate sia per gli indirizzi che per i dati.

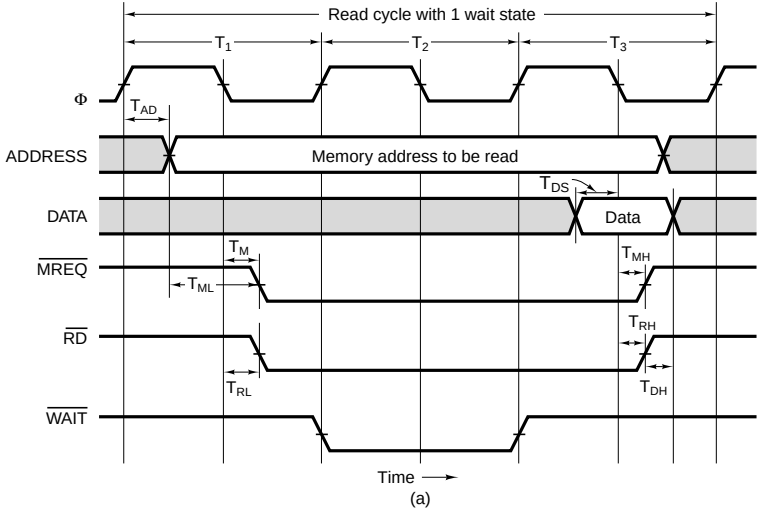
Crescita dell'Ampiezza del Bus



Temporizzazione del Bus

- I bus possono essere separati in due categorie distinte in base alla loro temporizzazione.
- Nei **bus sincroni**, esiste una linea pilotata da un clock, con frequenza generalmente compresa tra 5 e 100 MHz. Tutte le operazioni sul bus richiedono un numero intero di questi cicli.
- Nei **bus asincroni**, invece, non c'è un orologio principale: le operazioni sul bus possono avere una qualsiasi lunghezza.
- Per ragioni storiche e per rendere facile il progetto, i bus sincroni sono utilizzati molto più spesso dei bus asincroni.
- I bus asincroni, però, hanno un fondamentale vantaggio: permettono di sfruttare pienamente la velocità delle periferiche in gioco.
  - I bus sincroni, d'altro canto, devono sempre essere regolati alla velocità della periferica più lenta.

Bus Sincrono



(a)

Symbol	Parameter	Min	Max	Unit
T_{AD}	Address output delay		11	nsec
T_{ML}	Address stable prior to $\overline{MREQ}$	6		nsec
T_{M}	$\overline{MREQ}$ delay from falling edge of Φ in T_1		8	nsec
T_{RL}	RD delay from falling edge of Φ in T_1		8	nsec
T_{DS}	Data setup time prior to falling edge of Φ	5		nsec
T_{MH}	$\overline{MREQ}$ delay from falling edge of Φ in T_3		8	nsec
T_{RH}	$\overline{RD}$ delay from falling edge of Φ in T_3		8	nsec
T_{DH}	Data hold time from negation of $\overline{RD}$	0		nsec

(b)

Esempio di lettura della CPU in memoria

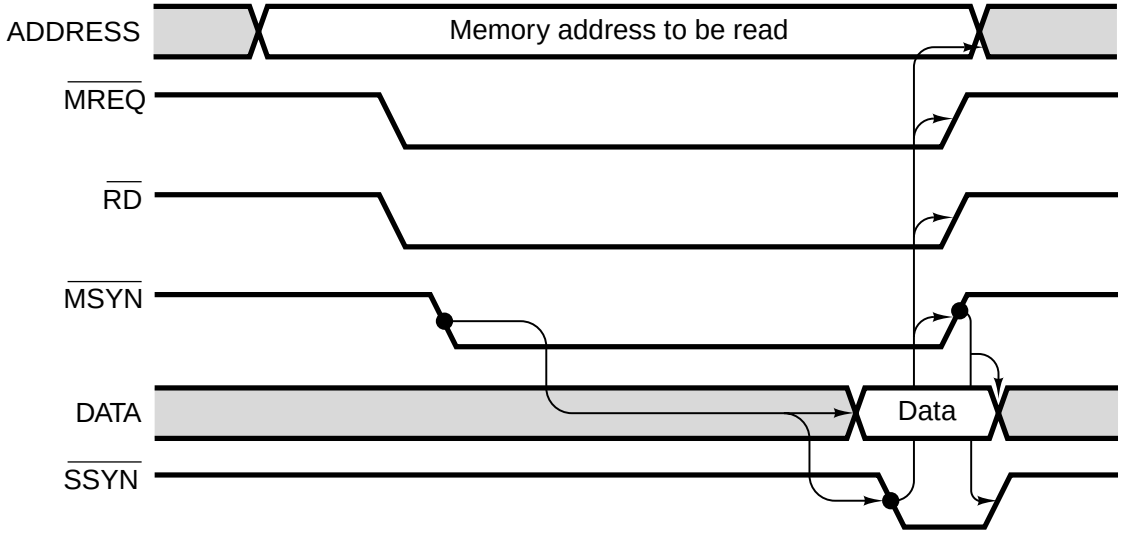
assumiamo che:

- il Clock operi a 40 MHz (tempo di ciclo di 25 ns).
- la lettura dalla memoria richieda 40ns dal momento in cui l'indirizzo è stabile sul BUS.

Si noti come tutti gli eventi siano sincronizzati con i fronti di salita o discesa del Clock.

- Per eseguire una lettura la CPU mette l'indirizzo sul BUS sul fronte di salita di T1.
- MREQ ed RD vengono attivati (attivi bassi) dalla CPU: il primo indica che si sta accedendo alla memoria (e non a un dispositivo di I/O), il secondo che si vuole leggere e non scrivere.
- Poiché la memoria richiede 40 ns dal momento in cui l'indirizzo è stabile, essa non è in grado di fornire la risposta entro T2; per avvertire la CPU di non aspettare dati, la memoria attiva il segnale WAIT (attivo basso) che fa sì che vengano aggiunti 1 o più stati di attesa.
- Quando la memoria è pronta a fornire la parola disabilita WAIT (nell'esempio WAIT è stato mantenuto attivo solo durante T2); durante la prima metà di T3 la memoria mette i dati sul BUS.
- Sul fronte di salita di T3 la CPU legge i dati sul BUS e disattiva MREQ e RD.

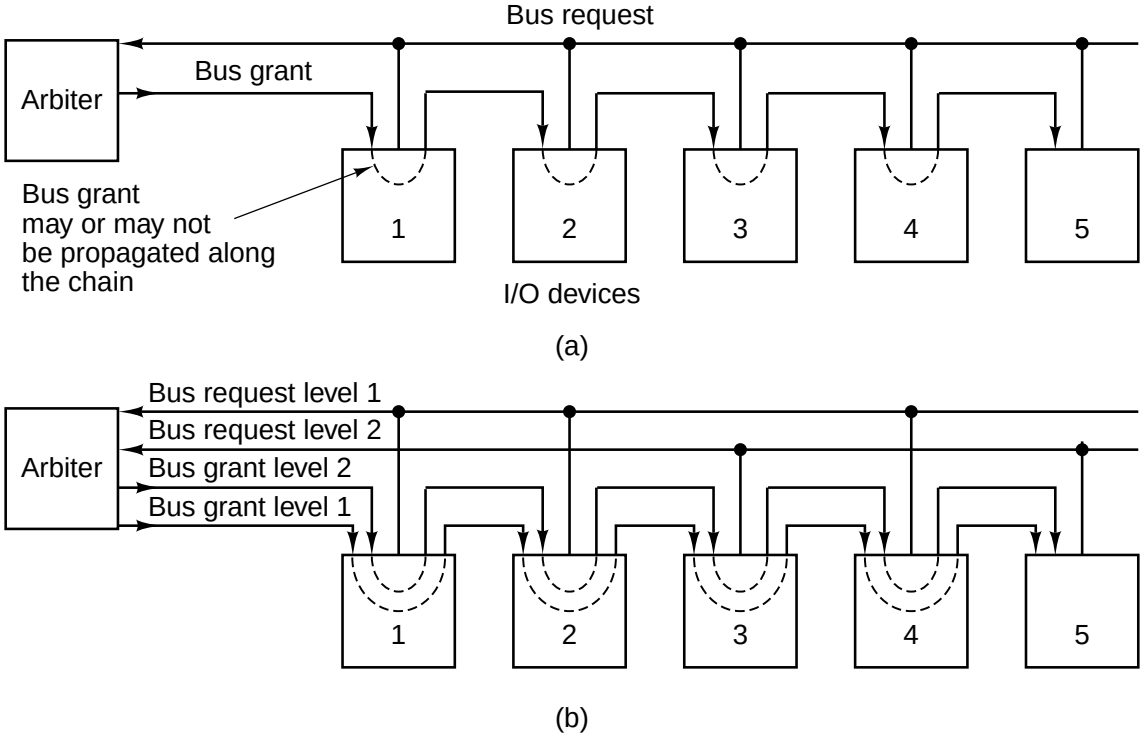
Bus Asincrono



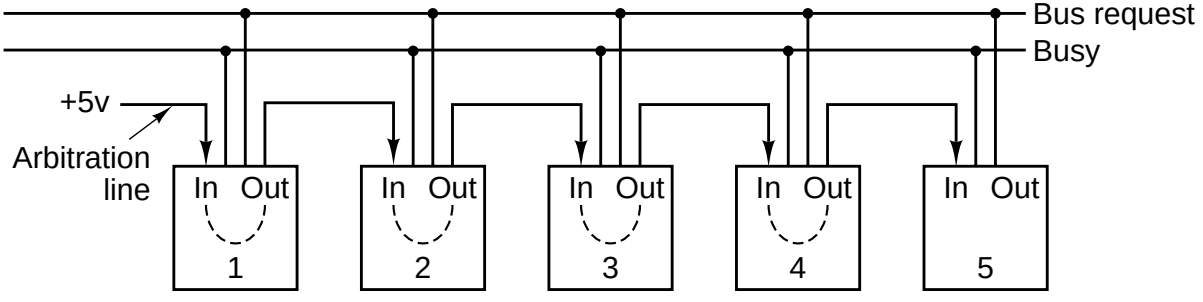
Arbitraggio del Bus

- Quando più dispositivi connessi allo stesso bus possono fungere da master, si rende necessaria qualche forma di **arbitraggio del bus**.
- L'arbitraggio del bus può essere **centralizzato**: un singolo arbitro determina chi sarà il prossimo master.
  - Esistono due linee: una per la richiesta del bus e una per la concessione del bus.
  - Questo schema è anche chiamato **daisy chaining** e assegna implicitamente una priorità più alta ai dispositivi fisicamente più vicini all'arbitro.
  - Per ovviare a questa rigidità, si possono utilizzare schemi con priorità.
- L'arbitraggio del bus può però anche essere **decentralizzato**:
  - Per esempio, potrebbe esserci una linea di richiesta per ogni dispositivo, ciascuna con la propria priorità.
  - Alternativamente, si potrebbe avere uno schema con una singola linea di richiesta, una linea asserita dal master corrente e una terza linea per la gestione dell'arbitraggio.

Arbitraggio Centralizzato



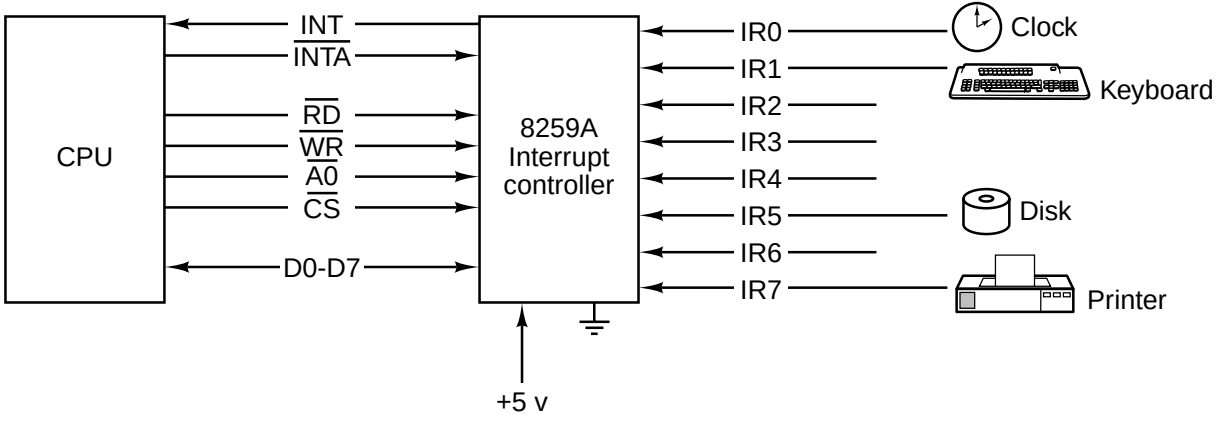
Arbitraggio Decentralizzato



Operazioni del Bus

- Oltre ai trasferimenti di dati tra un master e uno slave, esistono altri tipi di eventi che riguardano i bus esterni.
- Prima di tutto, si possono effettuare trasferimenti di **blocchi di dati**, anziché di singole parole di memoria.
  - Utili soprattutto in presenza di una cache.
- Può esistere uno speciale ciclo di bus che consente ad una CPU di leggere una parola dalla memoria, analizzarla e riscriverla in memoria.
  - Utile per gestire l'accesso concorrente alle strutture dati.
- Infine, attraverso il bus viaggiano anche i segnali di **interrupt**.
  - Tra l'altro, servono a comunicare alla CPU la fine di un'operazione di I/O.
  - Diverse periferiche potrebbero aver bisogno di interrompere la CPU contemporaneamente. Serve quindi un meccanismo di arbitraggio, gestito da un **controllore degli interrupt** (per esempio il chip Intel 8259A).

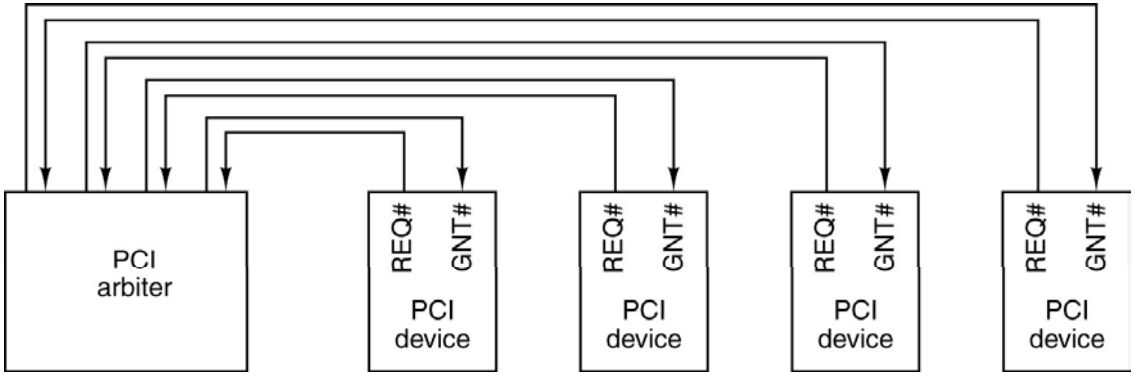
Controllore degli Interrupt



Bus PCI

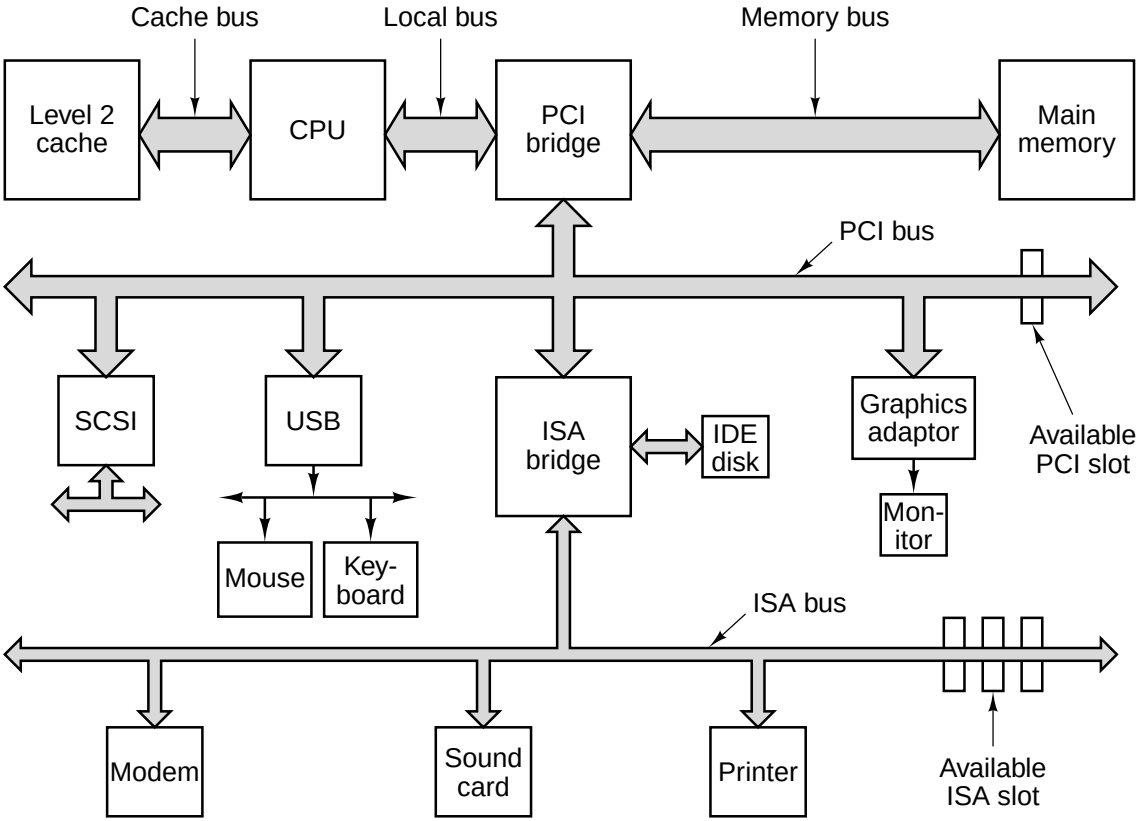
- I bus ISA, EISA sono troppo lenti per tutte le applicazioni che richiedono una certa larghezza di banda.
- Attorno al 1990, Intel progettò un nuovo bus con una larghezza di banda molto più ampia e lo chiamò **PCI** (Peripheral Component Interconnect).
  - La larghezza di banda del bus PCI originario era di 133 MB al secondo, contro una larghezza di banda di 16,7 MB al secondo per il bus ISA.
  - Nelle versioni successive, la larghezza di banda per il bus PCI aumenta ulteriormente.
- Nei PC moderni, i bus in gioco sono in genere molti, in modo tale da bilanciare esigenze diverse.
- Il bus PCI è sincrono.
- Le linee degli indirizzi e dei dati sono multiplexate.

PCI Bus Arbitration



The PCI bus uses a centralized bus arbiter.

Architettura di uno dei primi Pentium



Il cuore del problema è che vengono introdotti dispositivi sempre più veloci per i quali il bus PCI non è più adeguato, e la soluzione fino ad ora portata avanti è stata quella di prevedere porte addizionale sul/i bridge.

La strada intrapresa verso **PCI-Express** (il nome deriva da una pura operazione commerciale legata alla fama del nome PCI, in quanto non si tratta di un'estensione ma di una cosa completamente diversa!) porterà a un cambio di rotta radicale rispetto al passato:

- il bus diventa **seriale** con connessioni **punto a punto** ad alta velocità
- lo scambio di dati avviene attraverso un protocollo basato su pacchetti (simile alla comunicazione in rete).