

In questo modulo saranno presentati i principi di funzionamento dei moderni bus all'interno di un moderno calcolatore.

# BUS

Prof. Michele Tarantino

*Tutti i diritti riservati.*

*Il presente testo può essere utilizzato liberamente per motivi di studio, didattica e attività di ricerca purché sia presente il riferimento bibliografico.*

---



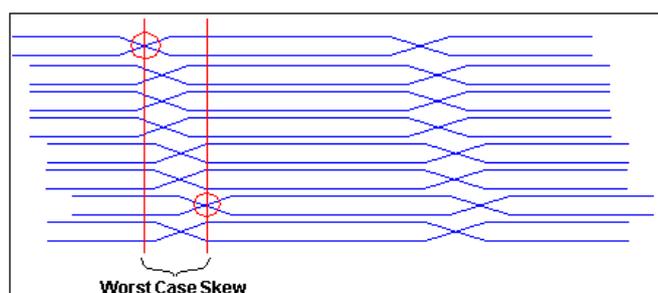
L'architettura di Von Neumann prevede che i diversi componenti siano collegati per mezzo di infrastrutture dedicate chiamate BUS. Il bus è un insieme di linee di comunicazione generalmente costituite da rame che permettono il trasferimento dei segnali elettrici tra due o più dispositivi. Ogni dispositivo è collegato al bus per mezzo di un chip dedicato che permette di inviare i segnali sul bus (*bus driver*), di ricevere i segnali dal bus (*bus receiver*) o sia di inviare sia di ricevere (*bus transceiver*).

Il dispositivo che gestisce l'invio dei segnali sul bus prende il nome di *Master* mentre il dispositivo che lo riceve prende il nome di *Slave*. All'interno di un calcolatore per ottimizzare i tempi di trasferimento delle informazioni possono essere presenti più bus dedicati (come ad esempio il bus tra CPU e memoria). Per nessun motivo la memoria può diventare un dispositivo Master. Il bus è caratterizzato da una larghezza di banda che identifica il numero di segnali (e quindi il numero di bit) che possono essere trasferiti contemporaneamente.

Il bus da un punto di vista elettronico è costituito da un insieme di cavi in rame paralleli tra loro. Ogni bus è caratterizzato da bus di dati (per il trasferimento dei dati da dispositivo Master a Slave), bus di controllo per comunicare con i chip di comunicazione con il bus e il bus indirizzi per identificare la configurazione dell'indirizzo di memoria. In particolar modo se un bus indirizzi presenta  $n$  linee fisiche può indirizzare  $2^n$  indirizzi di memoria differenti. All'aumentare del numero delle linee di indirizzo aumenta però anche il costo e lo spazio fisico necessario per contenerli: per questo motivo la larghezza del bus deve essere opportunamente calibrata secondo le specifiche dell'intero sistema.

Per quanto riguarda il clock del bus, ossia la quantità di dati che può essere trasferita in un determinato intervallo di tempo, i bus si suddividono in bus sincroni e bus asincroni. Nei bus sincroni, il ciclo di clock è definito a priori, non è modificabile e per il trasferimento dei dati si utilizza per intero il ciclo di clock anche se questo potrebbe rimanere non utilizzato. Deve adattarsi a diversi dispositivi eterogenei tra loro e deve adattarsi al dispositivo con frequenza più bassa. Nei bus asincroni, invece, il ciclo di clock può essere modificato a seconda della coppia master-slave che lo sta utilizzando; è più difficile da implementare in quanto per ogni coppia Master-Slave bisogna identificare la frequenza di clock.

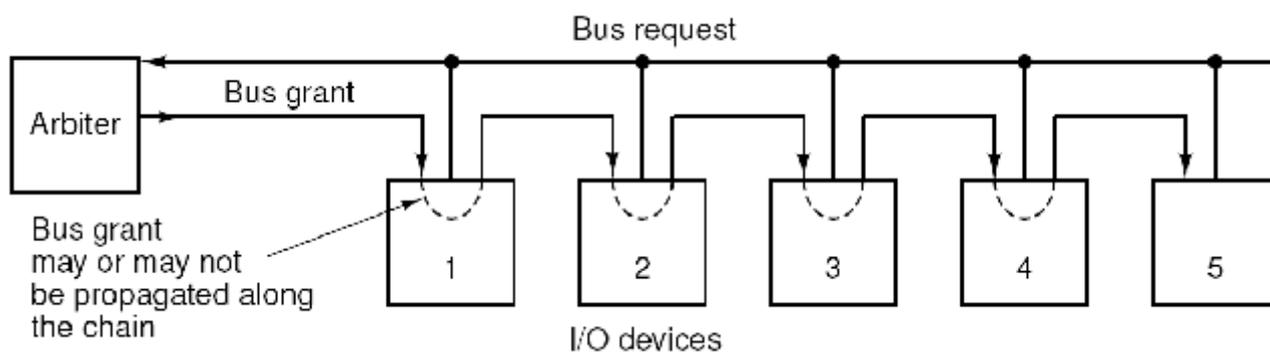
Per ottimizzare l'utilizzo del bus si possono utilizzare più linee di bus ma questo comporta un problema conosciuto come *bus Skew*, che identifica il fatto che i segnali che attraversano i diversi cavi di rame che compongono il bus possono raggiungere il dispositivo slave in istanti differenti.



Maggiore è la frequenza di clock maggiore sarà il bus Skew. Per questo motivo allo stato attuale vengono utilizzati bus sincroni, suddivisi in sotto sistemi.

Nei moderni calcolatori le componenti nobili (CPU, Memoria, Cache, unità di elaborazione grafica) sono collegate al bus denominato *Northbridge* mentre i dispositivi di I/O o quelli più lenti sono collegati al *Southbridge*, in questo modo possono lavorare a frequenze diverse e il collegamento tra i due avviene per mezzo di buffer dedicati in modo da ottimizzare il trasferimento. Un bus può essere dedicato come quello tra CPU e memoria (in quanto la maggior parte delle istruzioni a livello macchina sono proprio di trasferimento tra di due elementi) o può essere condiviso tra più dispositivi.

Se il bus è condiviso solo una coppia Master-Slave alla volta può utilizzare il bus stesso, per questo motivo è introdotto un meccanismo di arbitraggio del bus: quando un dispositivo richiede l'utilizzo del bus, questo viene prenotato tramite un'apposita linea (*wired-OR*) che collega in serie i diversi dispositivi (quindi è possibile che più dispositivi richiedano l'utilizzo del bus). Una linea dedicata di assegnazione del bus (*bus grant*) collega in serie i diversi dispositivi. Il dispositivo più vicino all'arbitro del bus (*chipset*) otterrà per primo l'assegnazione del bus se è stata richiesta, diversamente passa l'assegnazione al dispositivo successivo. Quindi può accadere che un dispositivo più lontano dall'arbitro del bus effettui la richiesta e nel frattempo un altro dispositivo più vicino richieda altrettanto uso del bus, e quindi i dispositivi più lontani potrebbero non utilizzare il bus o utilizzarlo ad intervalli di tempo elevati. Per questo motivo sono utilizzate più linee *wired-OR* e più linee di *grant bus* per fornire una priorità a livelli dei vari dispositivi. Questa tecnica chiamata *daisy-chain* è alla base dell'arbitraggio centralizzato.



È possibile un arbitraggio decentralizzato dove sono presenti tre linee: una per la richiesta del bus, una linea per identificare che il bus è impegnato ed è attivata dal dispositivo master corrente, la terza linea consente di collegare i diversi dispositivi in daisy chain. Questo metodo richiede più linee di bus ma risparmia il costo dell'arbitro.



Resta connesso e informato sui prossimi eventi, corsi e seminari:

## **Web**

[www.profmicheletarantino.com](http://www.profmicheletarantino.com)

## **Email**

[profmicheletarantino@gmail.com](mailto:profmicheletarantino@gmail.com)

## **Telefono**

349 83 54 521

## **Facebook**

[@micheletarantinodocente](https://www.facebook.com/micheletarantinodocente)

## **Instagram**

[@profmicheletarantino](https://www.instagram.com/profmicheletarantino)

Hai bisogno di un modulo personalizzato? Non esitare a contattarmi!