

Arbitraggio

Definizione

- ad ogni istante una sola unita può funzionare da master del bus
- l'arbitraggio entra in funzione quando 2 o più unità fanno richiesta di diventare master del bus
- arbitraggio avviene in modo
 - centralizzato** esiste un **arbitro**
 - distribuità** ogni unità contiene la logica di arbitraggio che permette di definire il nuovo master

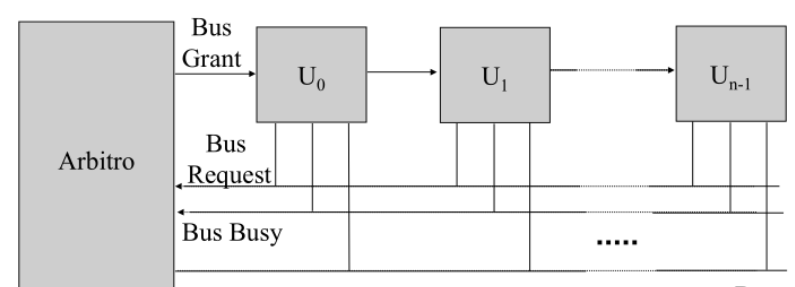
Arbitraggio distribuito: bus SCSI

- il bus SCSI possiede 8 linee per il **trasferimento dati** e per l'**arbitraggio** DB(0),...,DB(7)
- durante l'arbitraggio ogni **linea** è associata a una **unità** la priorità massima s **DB(7)**
- quando la LINEA **BSY**
 - indica se la il BUS è occupato
 - quando diventa **inattiva**
 - tutte le unità che voglio fare accesso al bus alzano la ripetitiva linea **DB**
 - tutte le unità leggono le linee quella a priorità più alta vince

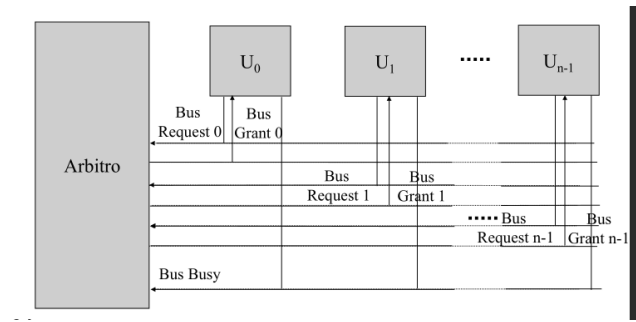
Arbitraggio Centralizzato

- Esistono 3 meccanismi
 - Daisy Chaining**
 - Polling**
 - Richieste indipendenti**
- Differiscono per
 - numero di linee di controllo
 - velocità di risposta del bus controller
 - flessibilità nella priorità
 - tolleranza ai guasti

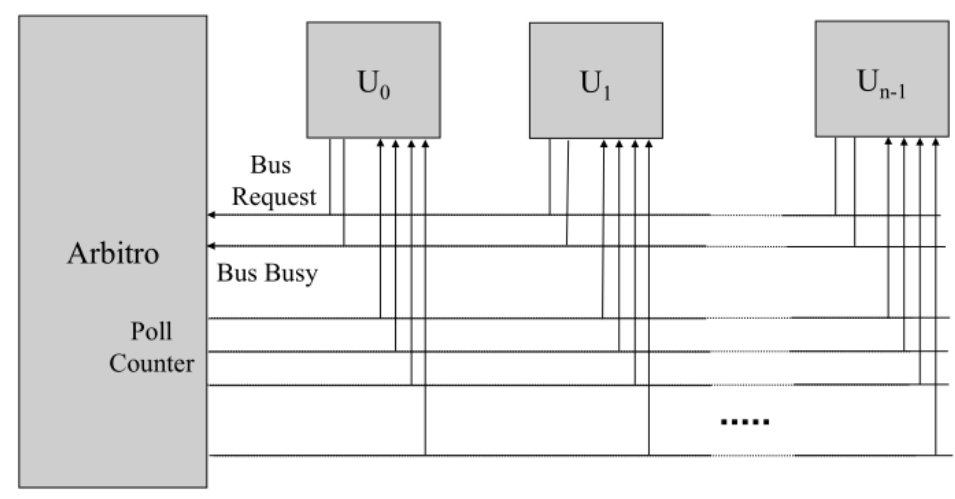
Arbitraggio: Daisy Chaining

- 
 - BUS REQUEST** una linea che va ad **1** se almeno un'unità fa richiesta del bus
 - BUS BUSY** una linea che va ad 1 se un'unità a preso il controllo del BUS
 - BUS GRANT** una linea che da il permesso di accedere al BUS
- Funzionamento**
 - Se un' unità deve usare il BUS attende che il BUS sia libero osservando **BUS BUSY**
 - è quindi fa richiesta del BUS attivando il **BUS REQUEST**
 - l'arbitro attiva il segnale di **BUS GRANT**
 - ogni **unità** quando riceve il **BUS GRANT**
 - se ha richiesto attiva il **BUS BUSY**
 - prende il controllo del bus
 - se non ha richiesto attiva il **BUS GRANT** verso la prossima unità
- Caratteristiche**
 - Vantaggi
 - usa solo 3 segnali
 - Svantaggi
 - priorità **immutabile**
 - non va bene ad alti numeri di unità connesse
 - troppo lente
 - deve passare per ogni unità
 - non è tollerante ai guasti

Arbitraggio: Richieste indipendenti

- 
 - BUS REQUEST i** la linea che va ad **1** se **i-esima** unità fa richiesta del BUS
 - BUS BUSY** una linea che va ad 1 se un'unità a preso il controllo del BUS
 - BUS GRANT i** la linea che da il permesso all' **i-esima** unità di accedere al BUS
- Funzionamento**
 - l'unità i-esima**
 - osservando **BUS BUSY** attende che il BUS sia libero
 - attivando il **BUS REQUEST i** è quindi fa richiesta del BUS
 - l'arbitro gestisce tutte le richieste e concedendo il BUS con una certa priorità
 - attiva il **BUS GRANT i**
 - quando **unità** riceve il suo **BUS GRANT**
 - attiva il **BUS BUSY**
 - prende il controllo del bus
- Caratteristiche**
 - Vantaggi
 - la priorità dipende dal meccanismo deciso nell' arbitro
 - Svantaggi
 - il sistema tollera eventuale guasti
 - $2n + 1$ segnali richiede

Arbitraggio: Polling

- 
 - BUS REQUEST** una linea che va ad **1** se almeno un'unità fa richiesta del bus
 - BUS BUSY** una linea che va ad 1 se un'unità a preso il controllo del BUS
 - POLL COUNTER** l'output di un counter per scandisce a chi bisogna dare il **BUS**
- Funzionamento**
 - Se un' unità deve usare il BUS attende che il BUS sia libero osservando **BUS BUSY**
 - attivando il **BUS REQUEST** è quindi fa richiesta del BUS
 - l'arbitro scandisce tutte le unità collegate, tramite l'ausilio di un counter che genera gli indirizzi dei moduli
- unità** quando è indirizzata
 - se ha richiesto
 - attiva il **BUS BUSY**
 - prende il controllo del bus
 - se non ha richiesto non fa nulla
- Caratteristiche**
 - Vantaggi
 - la priorità è modificabile cambiando la sequenza di scansione
 - Svantaggi
 - il sistema tollera eventuale guasti
 - $2 + \log_2(n)$ segnali richiede