

I bus

M. Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

1

M. Sonza Reorda - a.a. 2001/2002

Sommario

- **Introduzione**
- **Tempistica e protocollo**
- **Arbitraggio**
- **Esempi**

2

M. Sonza Reorda - a.a. 2001/2002

Introduzione

Un bus è una struttura che interconnette due o più dispositivi.

Un bus è una struttura condivisa: i valori che un dispositivo scrive su di esso sono accessibili a tutti gli altri dispositivi connessi.

Si possono avere bus:

- interni ad un singolo circuito integrato
- per la connessione di più circuiti integrati su una scheda
- per la connessione di più schede in un sistema (*backplane*).

3

M. Sonza Reorda - a.a. 2001/2002

Architetture di Bus

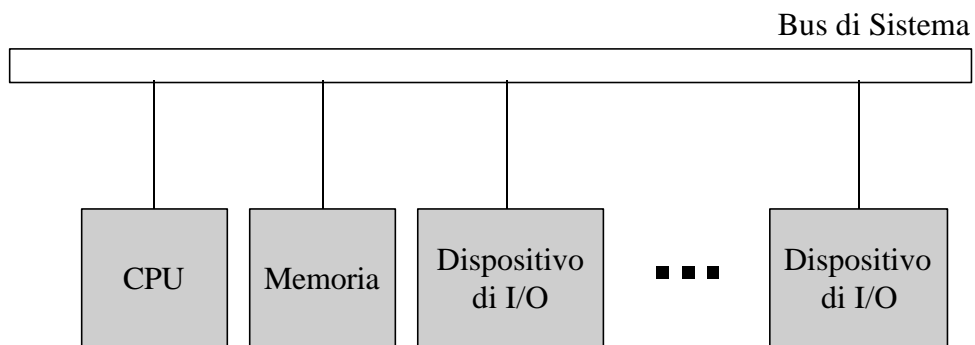
Si possono avere 2 tipi di architetture a bus:

- *bus singolo*: è la configurazione più semplice
- *bus multiplo*: è utile laddove si desiderano prestazioni elevate, oppure quando si devono connettere diverse classi di dispositivi, con caratteristiche tra loro diverse.

4

M. Sonza Reorda - a.a. 2001/2002

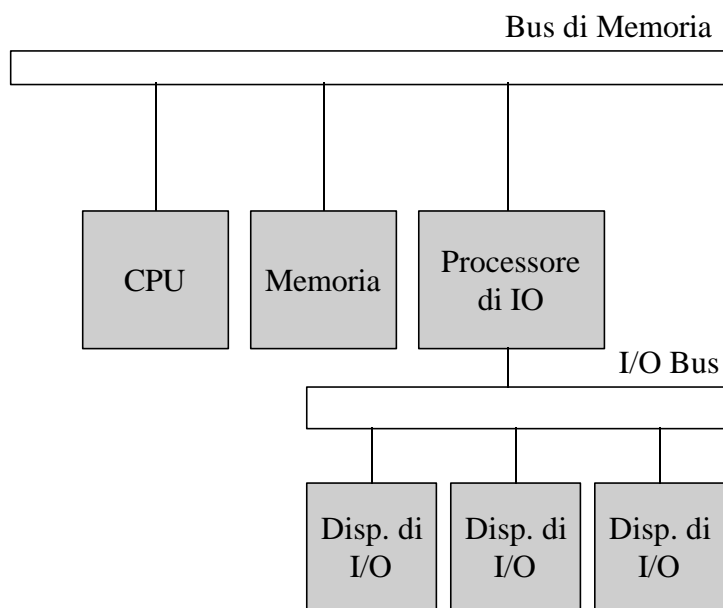
Bus Singolo



5

M. Sonza Reorda - a.a. 2001/2002

Bus Multiplo (esempio)



6

M. Sonza Reorda - a.a. 2001/2002

Struttura di un Bus

Un bus è composto da 3 gruppi di segnali:

- segnali di *dato*: normalmente sono in numero pari ad un multiplo di 8; possono essere bidirezionali o unidirezionali (in tal caso è necessario un numero doppio di linee);
- segnali di *indirizzo*: identificano lo slave con cui il master vuole comunicare (nonchè quale parte dello slave è coinvolta); possono essere multiplexati con i segnali di dato per ridurre le linee del bus e/o i pin dei dispositivi connessi;
- segnali di *controllo*: forniscono informazioni di stato, di temporizzazione, di tipo (dei dati sul bus).

7

M. Sonza Reorda - a.a. 2001/2002

Problemi

- *Multiplexato o dedicato?*
 - Il primo è più economico come hardware, il secondo è più veloce.
- *Comunicazioni sincrone o asincrone?*
 - Dipende dalla distanza
- *Quale interfaccia?*
 - Esistono degli standard (es. *Multibus*, *VME*, *EISA*, *Futurebus+*).

8

M. Sonza Reorda - a.a. 2001/2002

Master e Slave

Considereremo un sistema a bus singolo nel quale le unità connesse sono di 2 tipi:

- unità *master*: inizia ogni procedura di trasferimento dati e sceglie lo slave con cui comunicare; è anche denominata Bus Control Unit e spesso coincide con la CPU;
- unità *slave*: risponde ai comandi dell'unità master; le memorie sono unità slave.

9

M. Sonza Reorda - a.a. 2001/2002

Bus Sincroni/Asincroni

Bus Sincroni:

- ogni unità di dato è trasferita in un periodo di tempo prefissato (normalmente un periodo di clock);
- le unità sorgente e destinazione utilizzano lo stesso segnale di clock, che fa parte del bus stesso;
- alternativamente, le 2 unità possono avere clock separati, ma alla stessa frequenza, e scambiare periodicamente segnali di sincronizzazione;
- la frequenza del clock è imposta dai dispositivi più lenti;
- il meccanismo funziona su distanze ridotte.

10

M. Sonza Reorda - a.a. 2001/2002

Bus Sincroni/Asincroni (II)

Bus Asincroni:

- ogni sessione di comunicazione può avere una sua velocità, determinata da appositi segnali di controllo che accompagnano i segnali di dato;
- si acquisisce così la massima flessibilità, a spese di una maggiore complessità del protocollo.

11

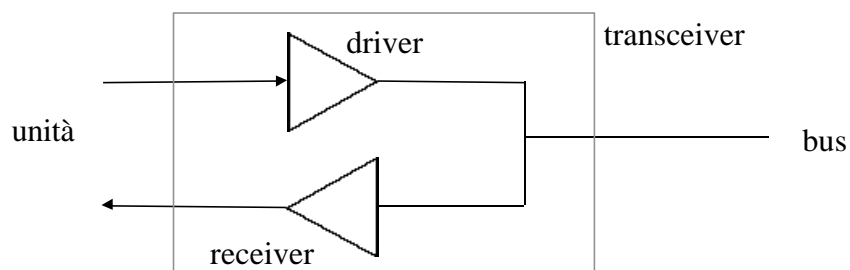
M. Sonza Reorda - a.a. 2001/2002

Interfaccia al Bus

Le unità connesse al bus utilizzano 2 tipi di dispositivi:

- *driver*: per pilotare le linee del bus
- *receiver*: per leggere i valori sul bus.

I due dispositivi sono spesso raggruppati in un'unica entità denominata *transceiver*.



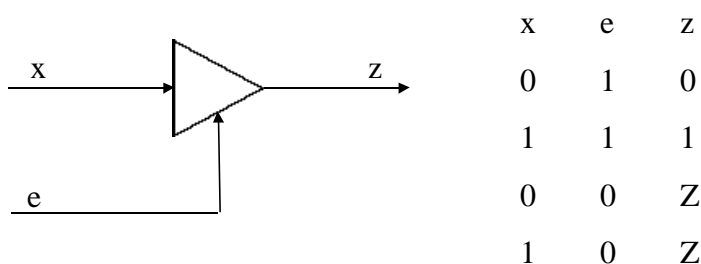
12

M. Sonza Reorda - a.a. 2001/2002

Tristate

La tecnologia *tristate* è quella comunemente utilizzata per implementare i driver: l'uscita di questi può assumere 3 valori: 0, 1, Z.

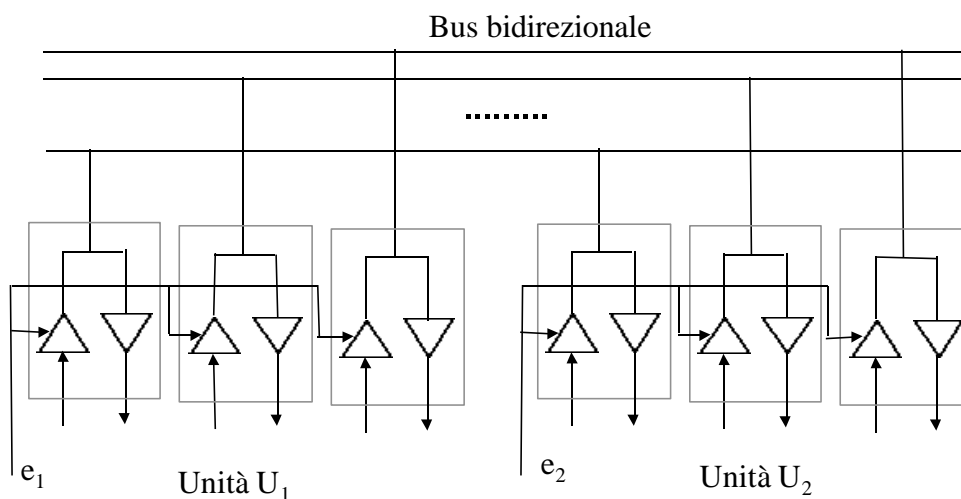
Quando una linea assume il valore Z (alta impedenza), essa corrisponde ad un circuito aperto.



13

M. Sonza Reorda - a.a. 2001/2002

Connessione tramite tristate



14

M. Sonza Reorda - a.a. 2001/2002

Problemi

L'utilizzo di un sistema a bus richiede la soluzione di 2 principali problemi:

- la definizione delle tempistiche per i segnali sul bus
- l'introduzione di un meccanismo per la gestione dei conflitti nell'accesso al bus.

15

M. Sonza Reorda - a.a. 2001/2002

Tempistica e Protocollo

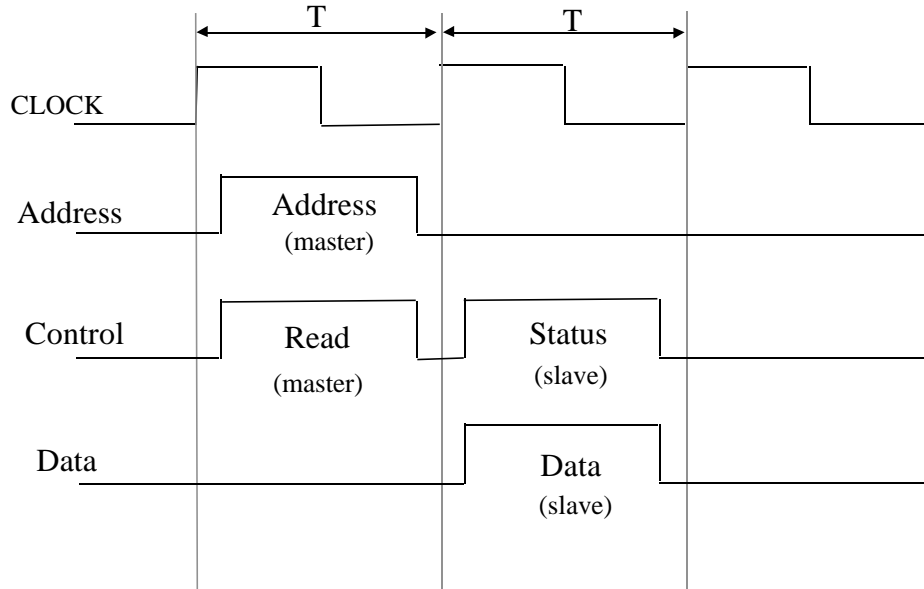
Le specifiche di un bus comprendono la descrizione del *protocollo* che i segnali dovranno seguire, nonché i limiti di tempo che dovranno essere rispettati.

Nel seguito sono riportati alcuni diagrammi che sintetizzano il protocollo e la tempistica per alcune operazioni su bus.

16

M. Sonza Reorda - a.a. 2001/2002

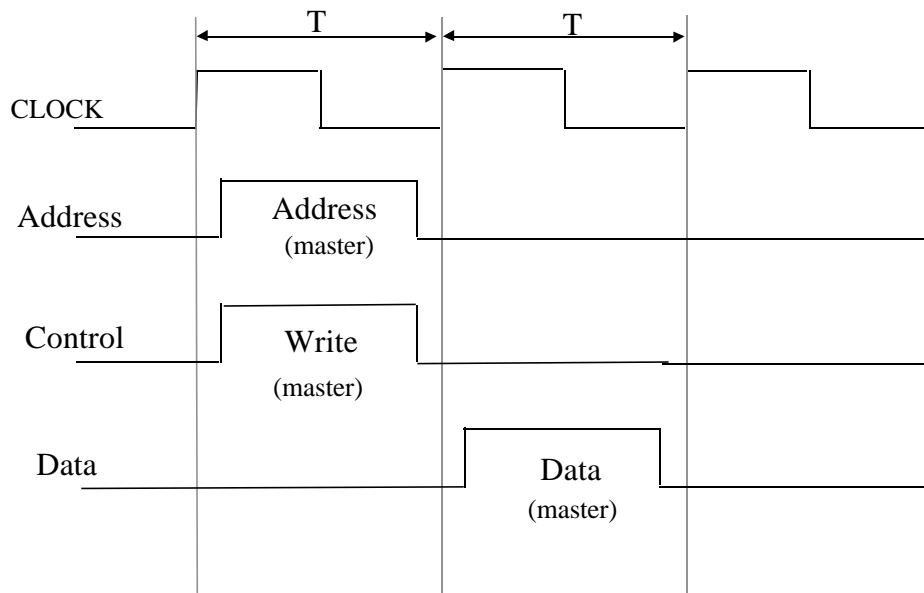
Bus Sincrono - Lettura



17

M. Sonza Reorda - a.a. 2001/2002

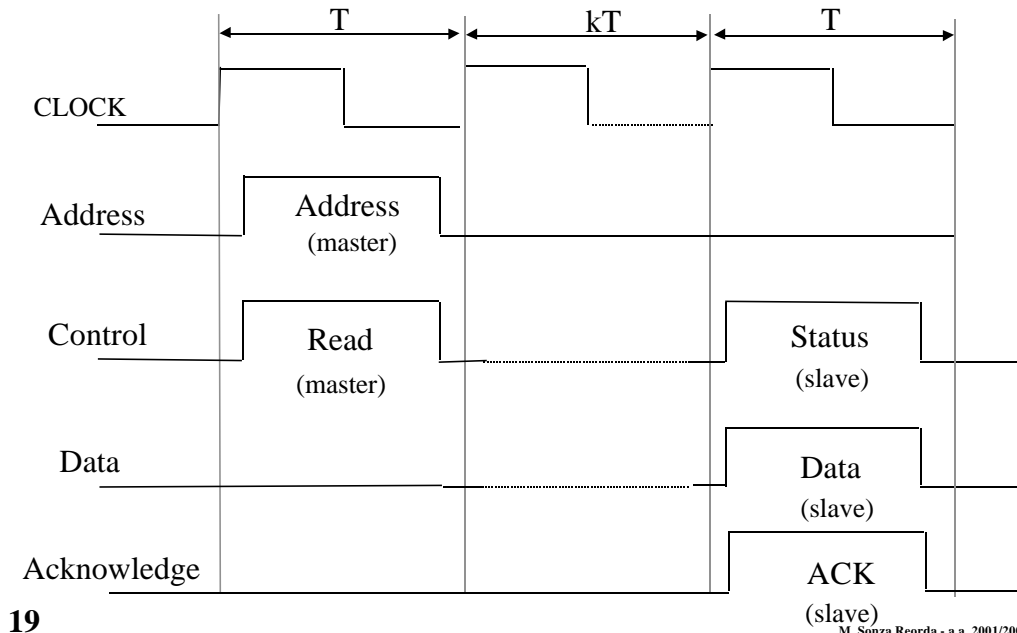
Bus Sincrono - Scrittura



18

M. Sonza Reorda - a.a. 2001/2002

Bus Sincrono - Lettura con Stati di Wait

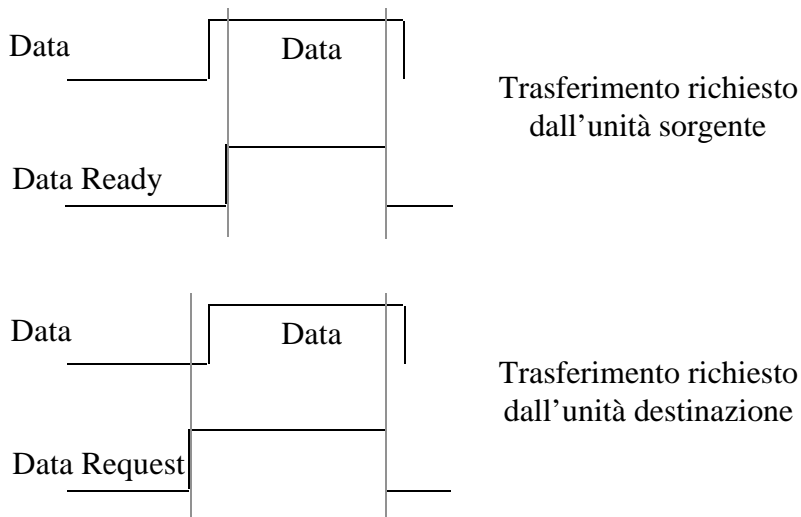


Bus Asincroni

Si possono avere due casi:

- *controllo unidirezionale*: i segnali di temporizzazione sono generati da uno solo dei due dispositivi coinvolti;
- *controllo bidirezionale, o interlacciato*: ambedue i dispositivi partecipano alla generazione dei segnali di temporizzazione.

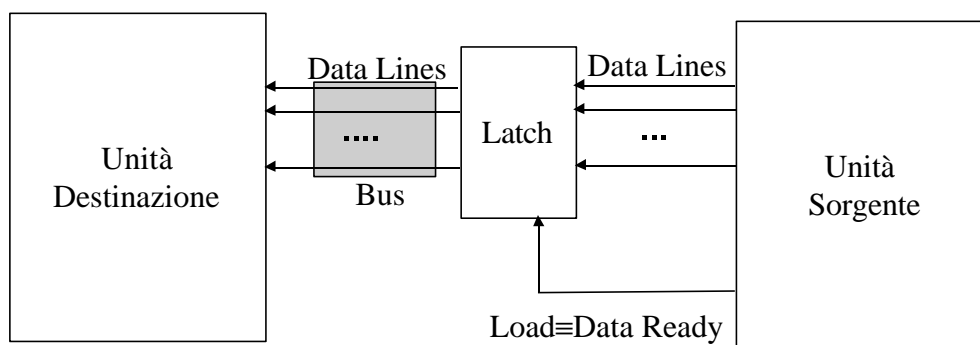
Bus Asincrono con Controllo Unidirezionale



21

M. Sonza Reorda - a.a. 2001/2002

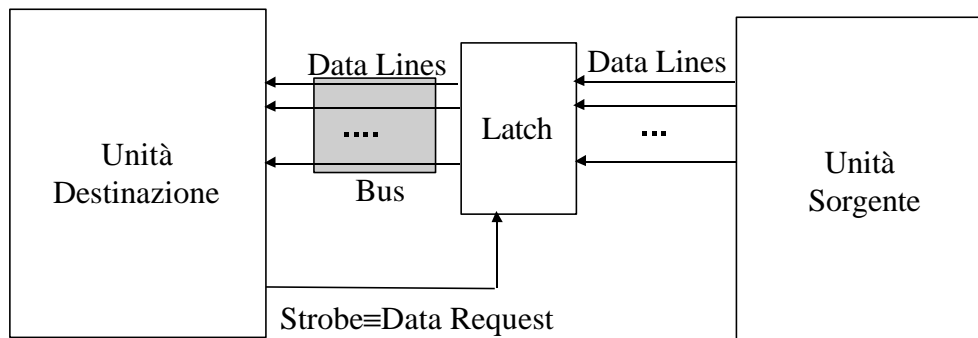
Latch (controllo del produttore)



22

M. Sonza Reorda - a.a. 2001/2002

Latch (controllo del consumatore)



23

M. Sonza Reorda - a.a. 2001/2002

Bus Asincroni con Controllo Bidirezionale

Permettono di verificare che il trasferimento sia andato a buon fine.

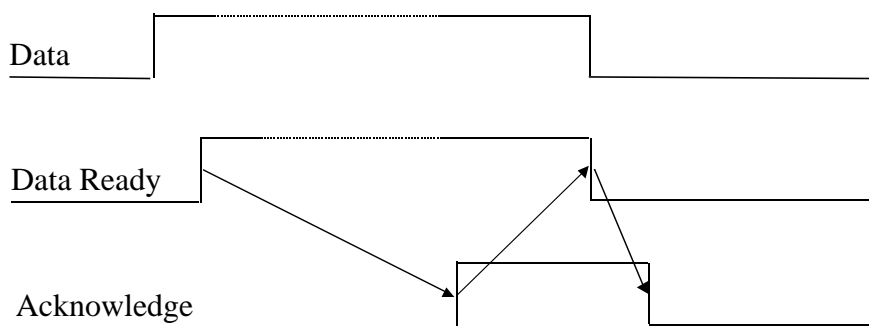
Per fare questo introducono un ulteriore segnale di Acknowledge, generato dall'unità che non ha richiesto il trasferimento.

La sequenza di valori che i segnali devono assumere prende il nome di *handshaking*.

24

M. Sonza Reorda - a.a. 2001/2002

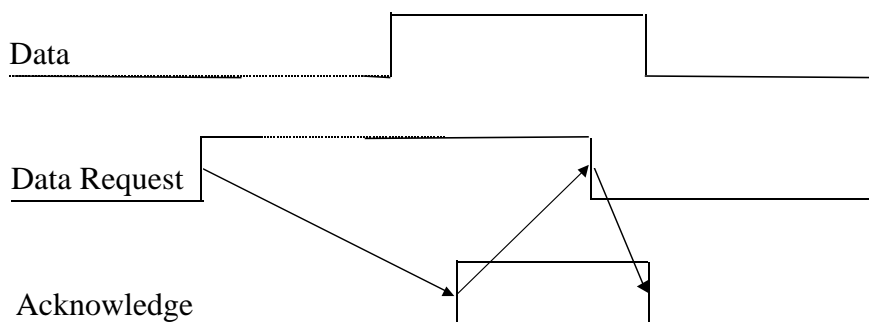
Bus Asincrono con Controllo Bidirezionale: trasferimento iniziato dall'unità sorgente



25

M. Sonza Reorda - a.a. 2001/2002

Bus Asincrono con Controllo Bidirezionale: trasferimento iniziato dall'unità destinazione



26

M. Sonza Reorda - a.a. 2001/2002

Arbitraggio

Il meccanismo di arbitraggio del bus entra in funzione quando 2 o più unità fanno contemporaneamente richiesta di accesso al bus. Il meccanismo deve allora designare il nuovo dispositivo master.

L'arbitraggio può avvenire in maniera

- *centralizzata*: esiste un *arbitro*, che a volte fa parte della CPU
- *distribuita*: ogni modulo contiene la logica necessaria per implementare un meccanismo di arbitraggio che permette di definire il nuovo master.

Arbitraggio distribuito: il bus SCSI

Il bus SCSI possiede 8 linee $DB(0), \dots, DB(7)$ che vengono utilizzate sia per il trasferimento dati che per l'arbitraggio.

Durante l'arbitraggio, ogni linea è associata ad un dispositivo: la linea $DB(7)$ ha la priorità massima.

Quando la linea BSY diventa inattiva, tutti i dispositivi (al più 8) che desiderano fare accesso al bus alzano la rispettiva linea $DB(i)$. Tutti i dispositivi osservano il valore sulle linee $DB(0), \dots, DB(7)$, ed il dispositivo con priorità massima vince la contesa; gli altri attendono che BSY torni inattiva.

Arbitraggio Centralizzato

Esistono 3 meccanismi di arbitraggio centralizzato:

- *daisy-chaining*
- *polling*
- *richieste indipendenti.*

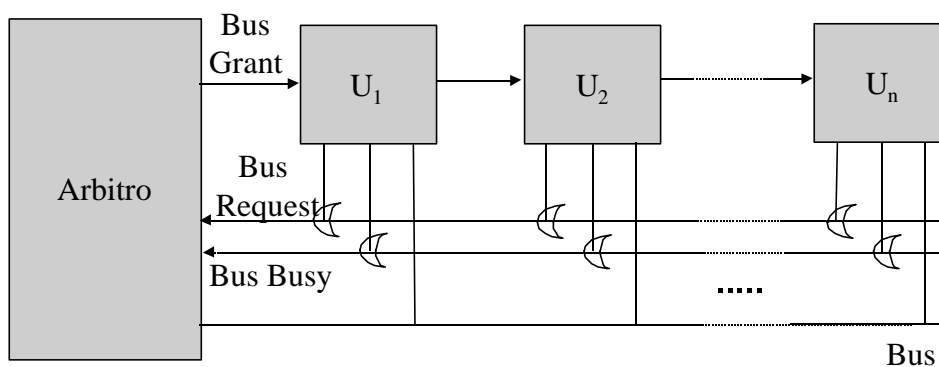
Essi differiscono per:

- numero di linee di controllo richieste
- velocità di risposta del bus controller
- flessibilità nella gestione delle priorità
- tolleranza ai guasti.

29

M. Sonza Reorda - a.a. 2001/2002

Daisy Chaining: struttura



30

M. Sonza Reorda - a.a. 2001/2002

Daisy Chaining: funzionamento

- una unità fa richiesta del bus (**BUS REQUEST**) attendendo che il bus sia libero (**BUS BUSY**)
- l'arbitro attiva il segnale di **BUS GRANT**
- ogni unità, quando riceve il **BUS GRANT**:
 - se ha richiesto il bus: attiva **BUS BUSY**
 - se non ha richiesto il bus: attiva **BUS GRANT** verso l'unità a valle.

31

M. Sonza Reorda - a.a. 2001/2002

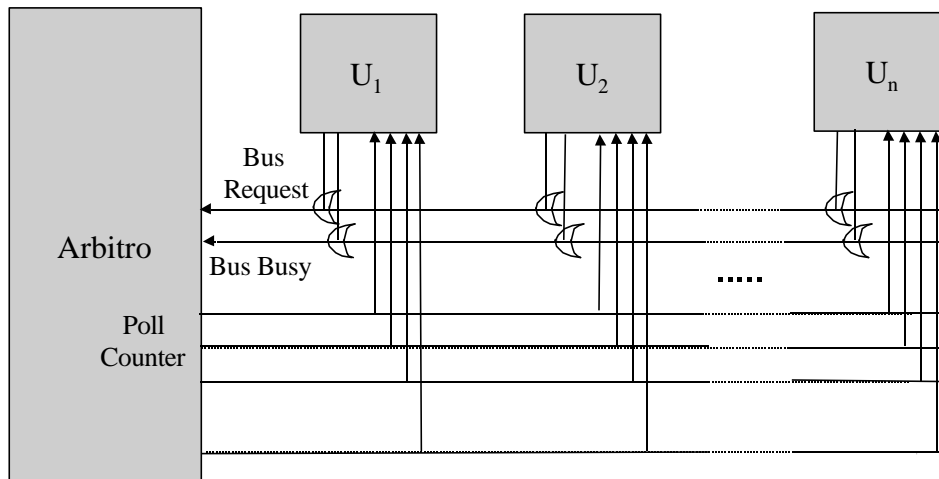
Daisy Chaining: caratteristiche

- + richiede solo 3 segnali di controllo
- non permette di modificare le priorità
- non è adatta a numeri elevati di dispositivi connessi
- non è tollerante ai guasti.

32

M. Sonza Reorda - a.a. 2001/2002

Polling: struttura



33

M. Sonza Reorda - a.a. 2001/2002

Polling: funzionamento

- Una unità fa richiesta del bus (BUS REQUEST), attendendo che il bus sia libero (BUS BUSY)
- l'arbitro scandisce tutte le unità collegate, mettendo sul Poll Counter gli indirizzi di ciascuna, in sequenza
- quando una unità viene indirizzata, ed ha fatto richiesta, attiva il segnale di BUS BUSY; a questo punto l'arbitro interrompe la scansione.

34

M. Sonza Reorda - a.a. 2001/2002

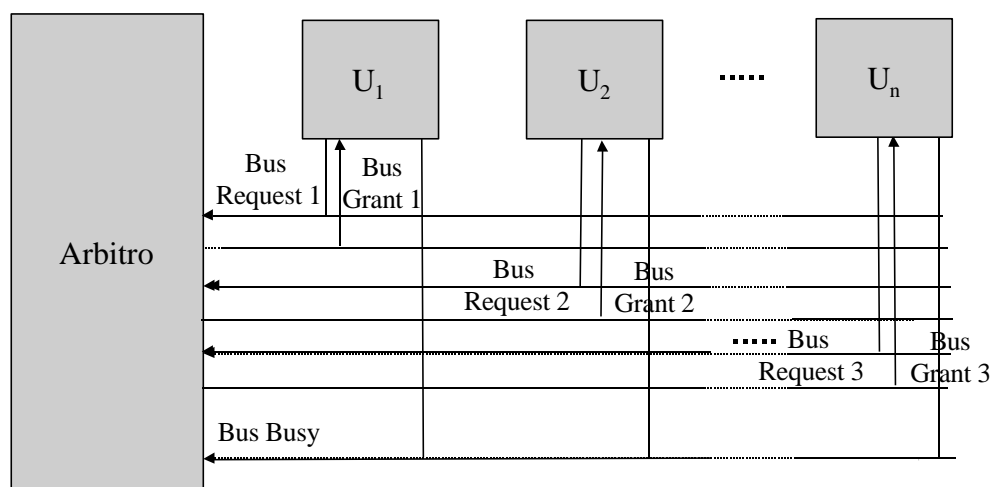
Polling: caratteristiche

- richiede $2 + \log(n)$ segnali di controllo per gestire n unità
- + la gerarchia delle unità può essere cambiata modificando la sequenza di scansione
- + il sistema è tollerante ad un eventuale guasto in una unità.

35

M. Sonza Reorda - a.a. 2001/2002

Richieste Indipendenti: struttura



36

M. Sonza Reorda - a.a. 2001/2002

Richieste Indipendenti: funzionamento

- l'unità i -esima fa richiesta del bus (BUS REQUEST i), attendendo che il bus sia libero (BUS BUSY)
- l'arbitro gestisce tutte le richieste, e concede il bus all'unità con priorità maggiore (BUS GRANT j)
- l'unità j assume il controllo del bus (BUS BUSY j).

37

M. Sonza Reorda - a.a. 2001/2002

Richieste Indipendenti: caratteristiche

- richiede $2 \cdot n + 1$ segnali di controllo per gestire n unità
- + le priorità dei dispositivi dipendono dai meccanismi implementati dall'arbitro
- + il sistema può tollerare ad un eventuale guasto in una unità.

38

M. Sonza Reorda - a.a. 2001/2002

Esempio: MultibusII

È un bus a 32 bit definito dall'Intel nel 1983.

Costituisce il successore del Multibus, che ha ampiezza 16 bit.

È stato standardizzato dall'ANSI/IEEE nel 1987 col codice 1296.

Usa una temporizzazione sincrona ed uno schema di arbitraggio distribuito.

39

M. Sonza Reorda - a.a. 2001/2002

Esempio: VMEbus

Il bus VMEbus (*Versa Module European bus*) prevede 32 linee per i dati, 32 per gli indirizzi, ed altre per il controllo. Ogni scheda deve presentare un connettore da 96 pin.

È il successore del Versabus, che ha ampiezza pari a 16 bit.

È stato definito congiuntamente nel 1981 da Motorola, Mostek e Philips/Signetics ed è adatto a sistemi di piccole e medie dimensioni. Si presta particolarmente per essere utilizzato con processori 68000.

È un bus di tipo asincrono con schema di arbitraggio centralizzato. La frequenza di trasferimento dei dati si aggira sui 30 Mbyte/s.

40

M. Sonza Reorda - a.a. 2001/2002

Esempio: il bus SCSI

Lo SCSI (*Small Computer System Interface*) è stato standardizzato dall'ANSI con la denominazione X3.131.

È rivolto ai sistemi di piccole e medie dimensioni e prevede un bus cui si possono collegare sino ad 8 dispositivi (principalmente dischi), ciascuno dotato di un proprio *controllore*.

I controllori possono essere di due tipi: iniziatori e target. I trasferimenti di dati sono sempre controllati dal controllore target, mentre l'iniziatore sceglie il target e definisce l'operazione da eseguire.

Ogni trasferimento prevede l'instaurazione di una *connessione logica*.

Ha una lunghezza massima di 25 metri. Può essere sincrono o asincrono.

41

M. Sonza Reorda - a.a. 2001/2002

Esempio: PCI

È un bus di backplane, adatto a collegare schede di tipo diverso.

Ha un'ampiezza variabile (32 o 64 bit) per la parte dati. Il bus dati ed il bus indirizzi condividono gli stessi segnali.

Può gestire sino a 32 dispositivi, e adotta un arbitraggio di tipo centralizzato.

È di tipo sincrono, e può funzionare ad una frequenza massima di 66 MHz, raggiungendo una velocità di trasferimento di picco pari a 80 MByte/s.

Ha una lunghezza massima di 50 cm.

42

M. Sonza Reorda - a.a. 2001/2002