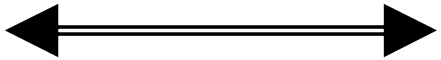
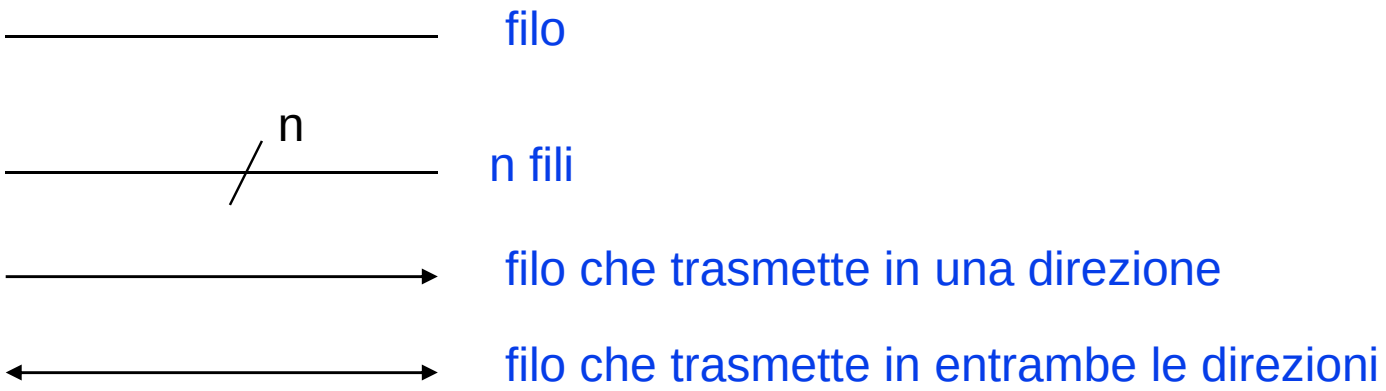
Lezione 11: BUS

- BUS = veicolo (lento) nel quale molte persone viaggiano

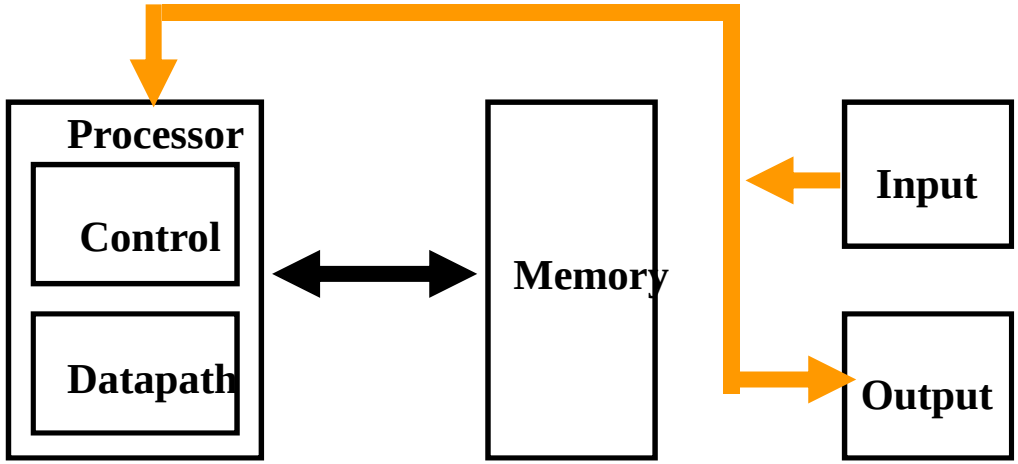
  … ed inoltre…

- Un insieme di fili…  **BUS**



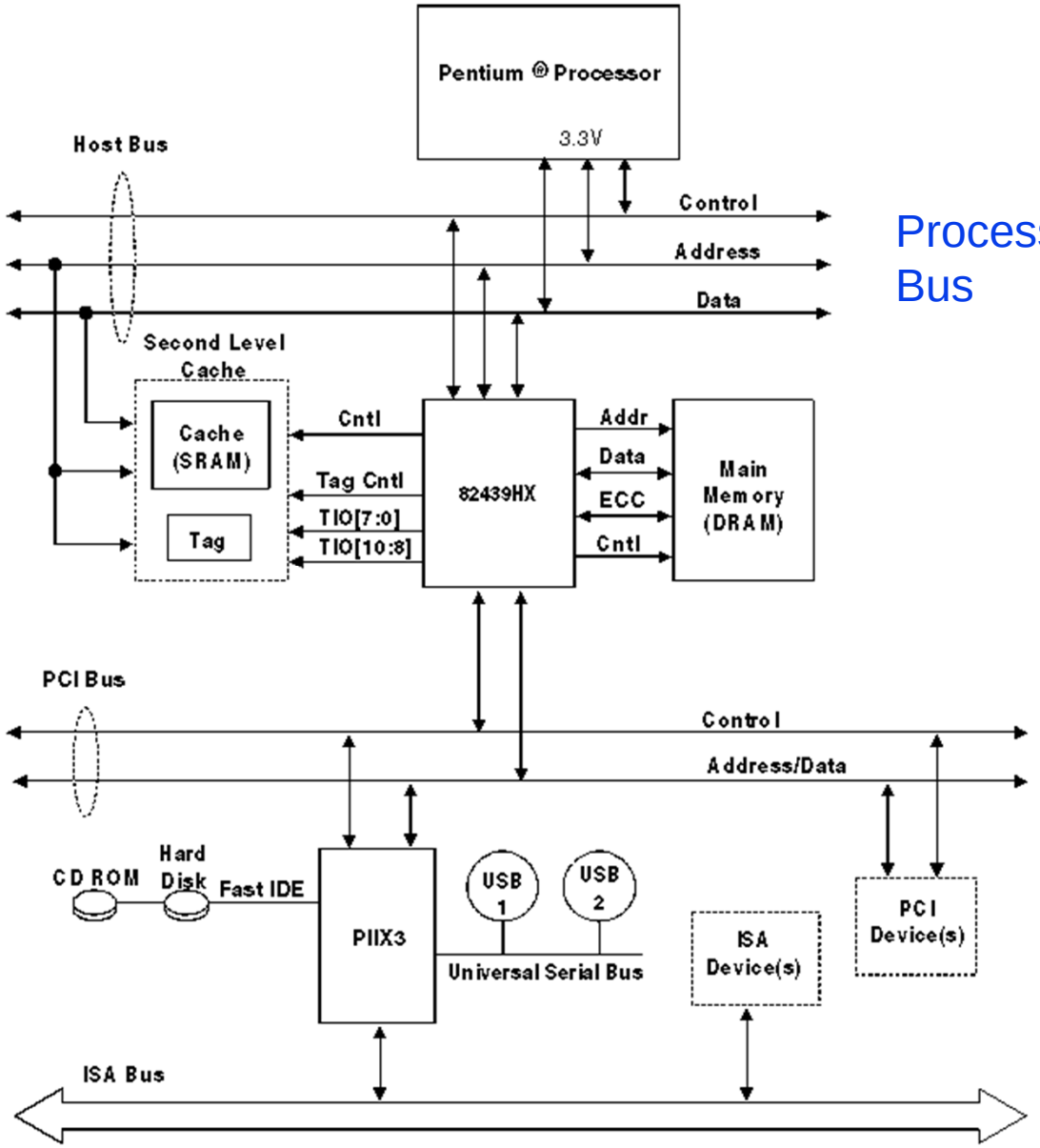
Un bus e':

- Un collegamento condiviso per comunicare
- Un insieme di fili usato per connettere vari sottosistemi



- Un bus e' anche il mezzo fondamentale per costruire sistemi anche grandi e complessi
  - E' uno strumento di astrazione molto utilizzato

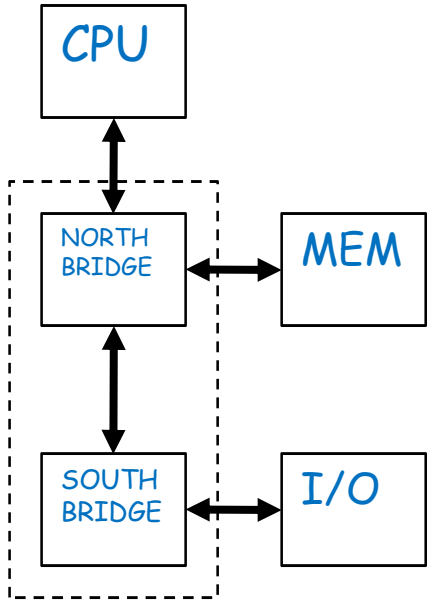
Esempio: Organizzazione di Sistema con CPU Pentium

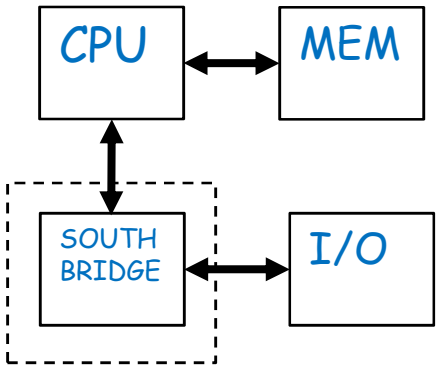


Processor/Memory Bus

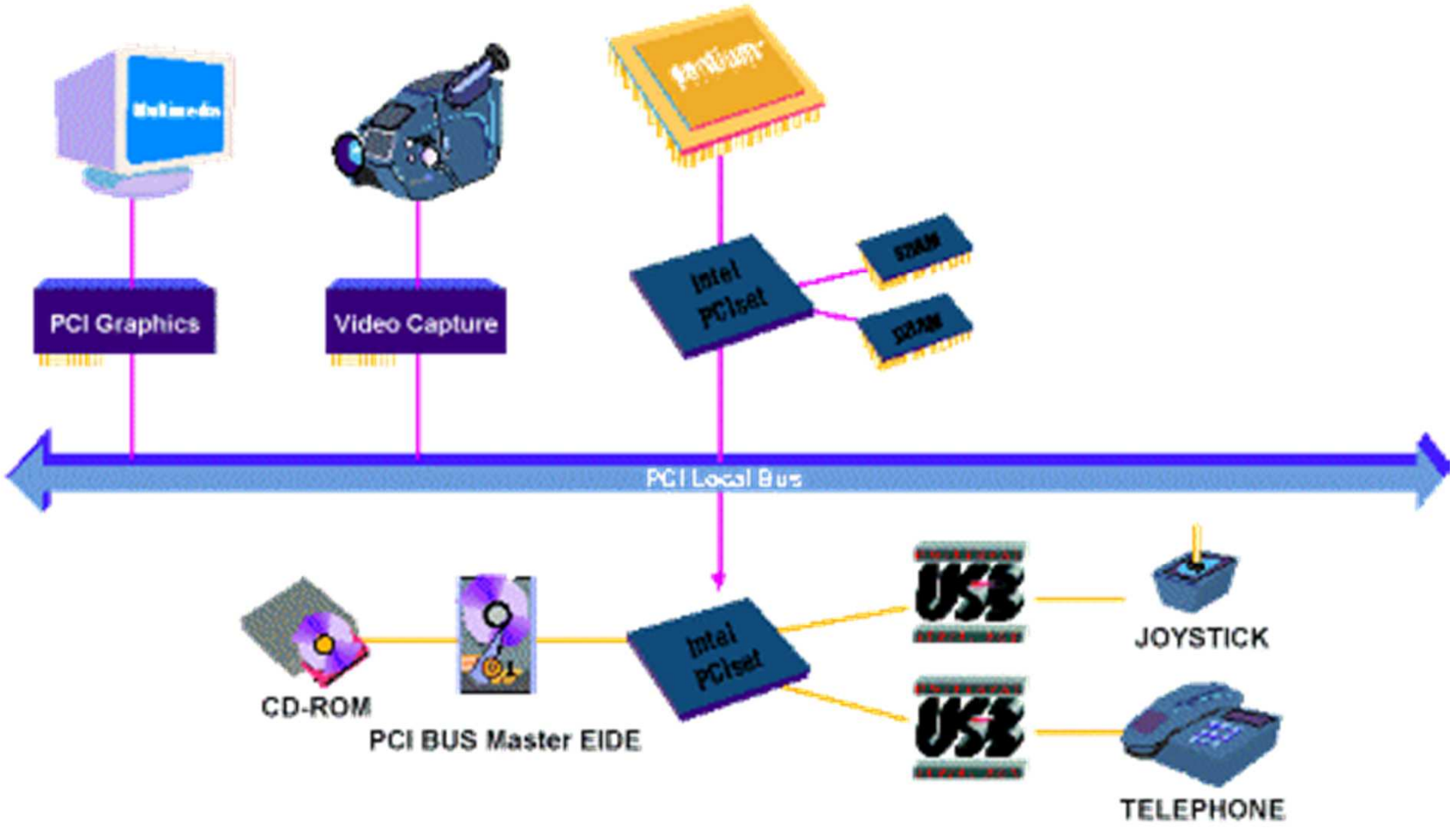
PCI Bus

I/O Bus

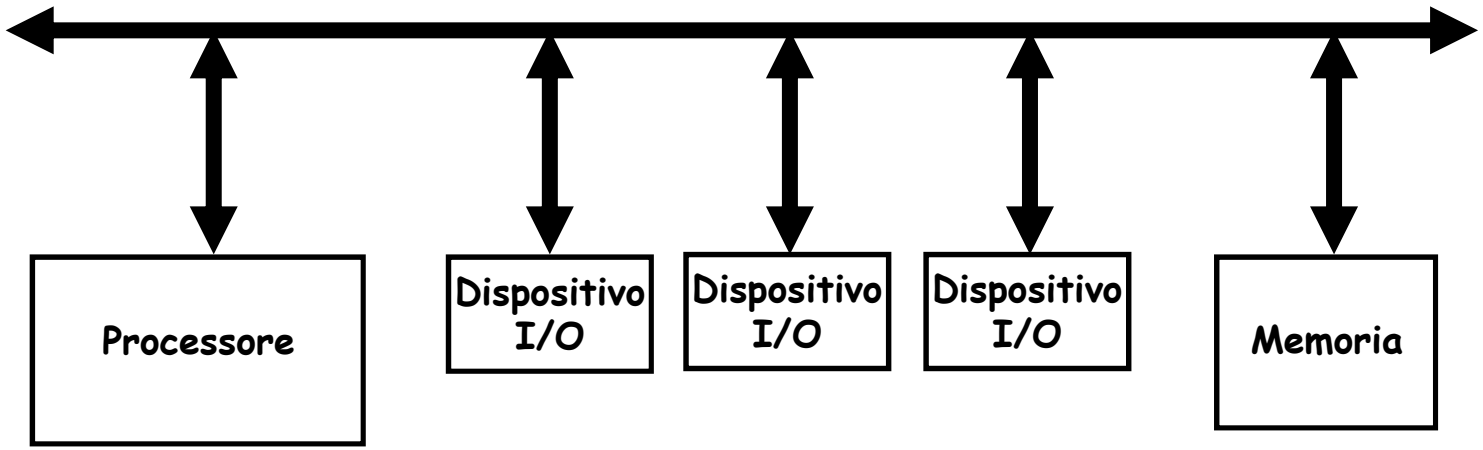




Esempi di dispositivi collegabili tramite bus

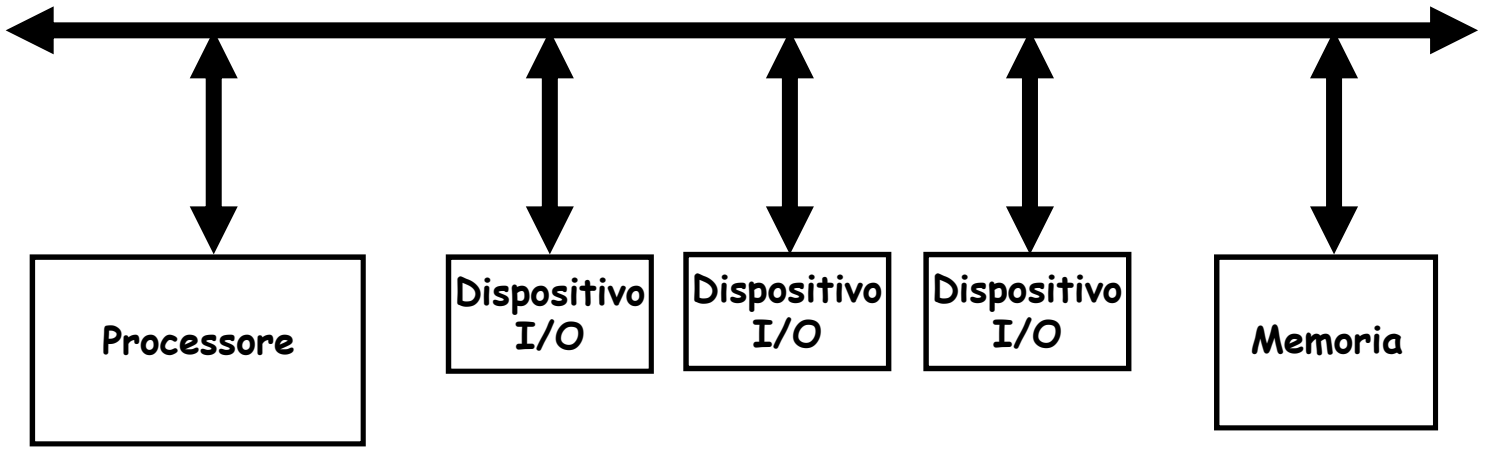


Vantaggi dei bus



- Versatilita'
  - Si possono aggiungere facilmente nuovi dispositivi
  - Una data periferica puo' essere usata su un calcolatore diverso che usa un bus che segue lo stesso standard
- Basso costo
  - Un unico insieme di fili puo' essere condiviso da dispositivi di vario tipo
- Gestire la complessita' suddividendo il sistema

Svantaggi dei bus

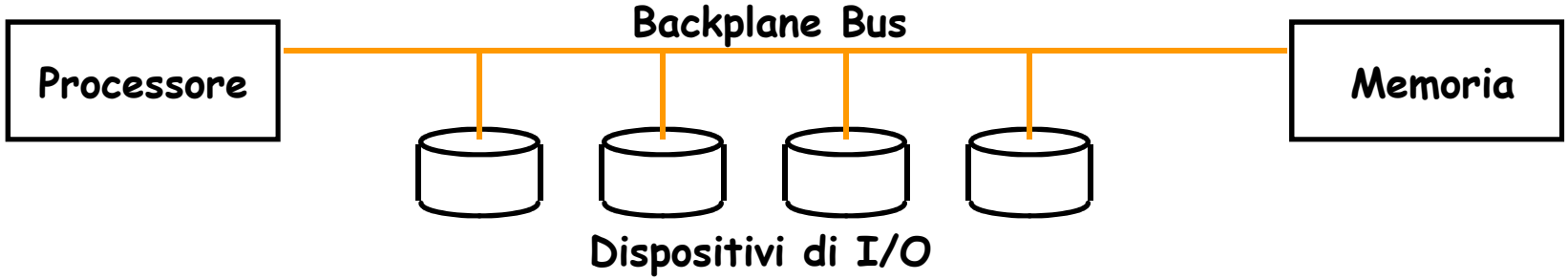


- Genera un "collo di bottiglia" nelle comunicazioni
  - La banda del bus limita il massimo throughput ottenibile per l'I/O
- La velocita' massima del bus e' fortemente limitata da
  - Lunghezza del bus
  - Numero di dispositivi collegati al bus
  - Necessita' di supportare un ampio numero di dispositivi aventi
    - Tempi di latenza ampiamente variabili
    - Velocita' di trasferimento dati ampiamente variabili

Tipi di Bus

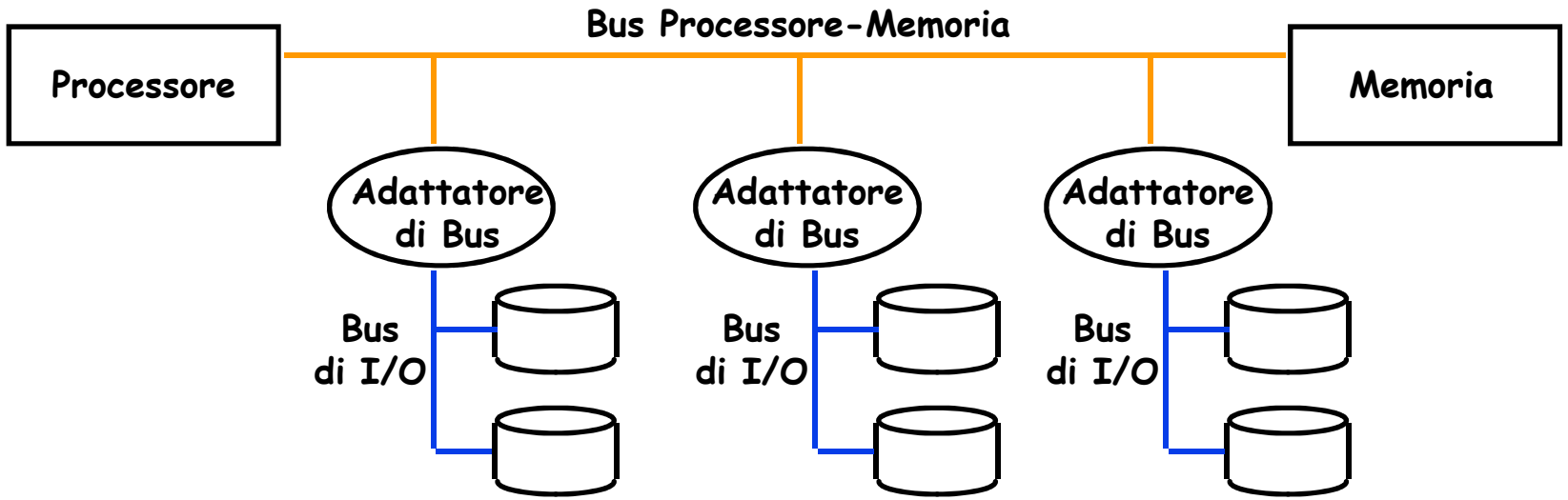
- Bus Processore-Memoria (specifico del sistema)
  - Corto e ad alta velocita'
  - Deve solo essere adatto al sistema di memoria utilizzato
    - Tipicamente cerca di massimizzare la banda processore-memoria
  - Connette direttamente il processore
  - Ottimizzato per i trasferimenti di blocchi di cache
- Bus di I/O Bus (standard industriale)
  - Tipicamente abbastanza lungo e lento
  - Deve essere adatto a collegare dispositivi di I/O molto diversi fra loro
  - Si connette al bus processore-memoria o al backplane bus
- Backplane Bus (standard o proprietario)
  - Backplane: interconnessione che si svolge all'interno dello chassis
  - Consente al processore, alla memoria e ai dispositivi di I/O di coesistere e cooperare fra loro
  - Si puo' ottenere un risparmio se si usa un solo bus per tutti i componenti del sistema

Un Calcolatore con un solo Bus: Backplane Bus



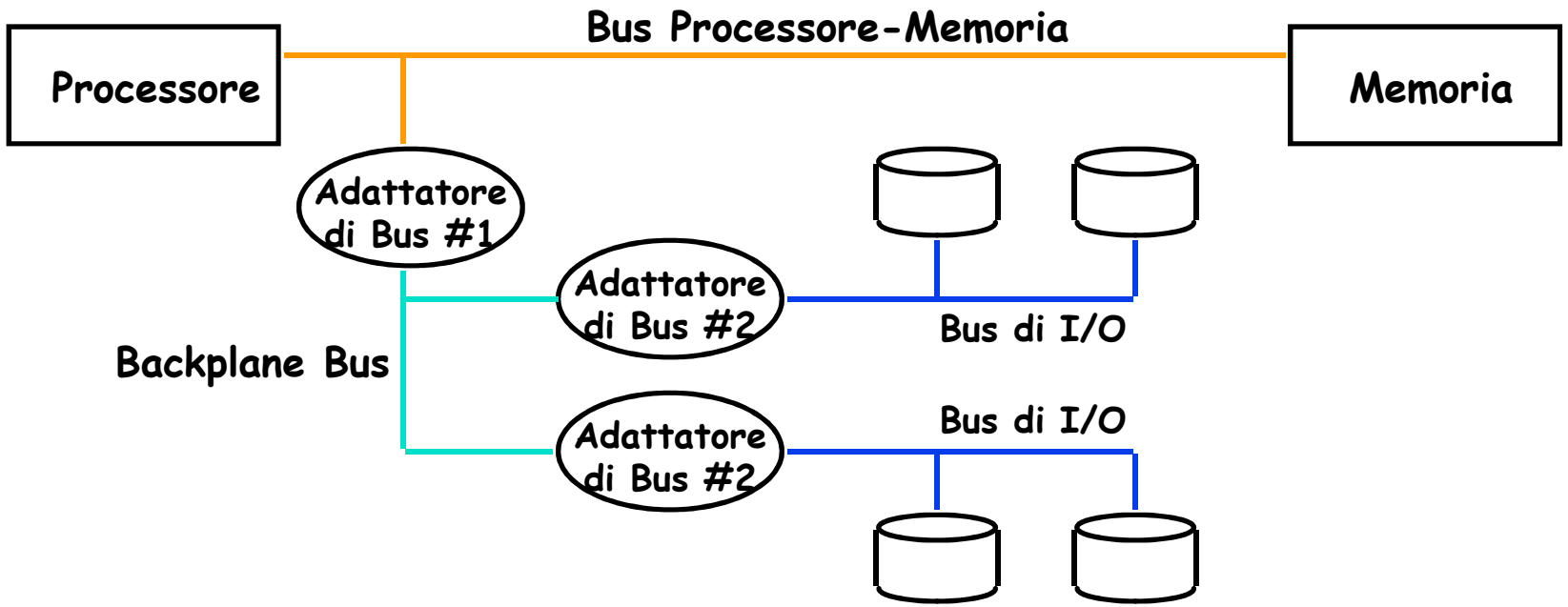
- Si puo' usare un sistema a singolo bus (backplane bus) per la
  - Comunicazione fra processore e memoria
  - Comunicazione fra dispositivi di I/O e memoria
- Vantaggi: semplice e a basso costo
- Svantaggi: lento e principale collo di bottiglia del sistema
- Esempio: PC-AT della IBM

Sistema con due bus



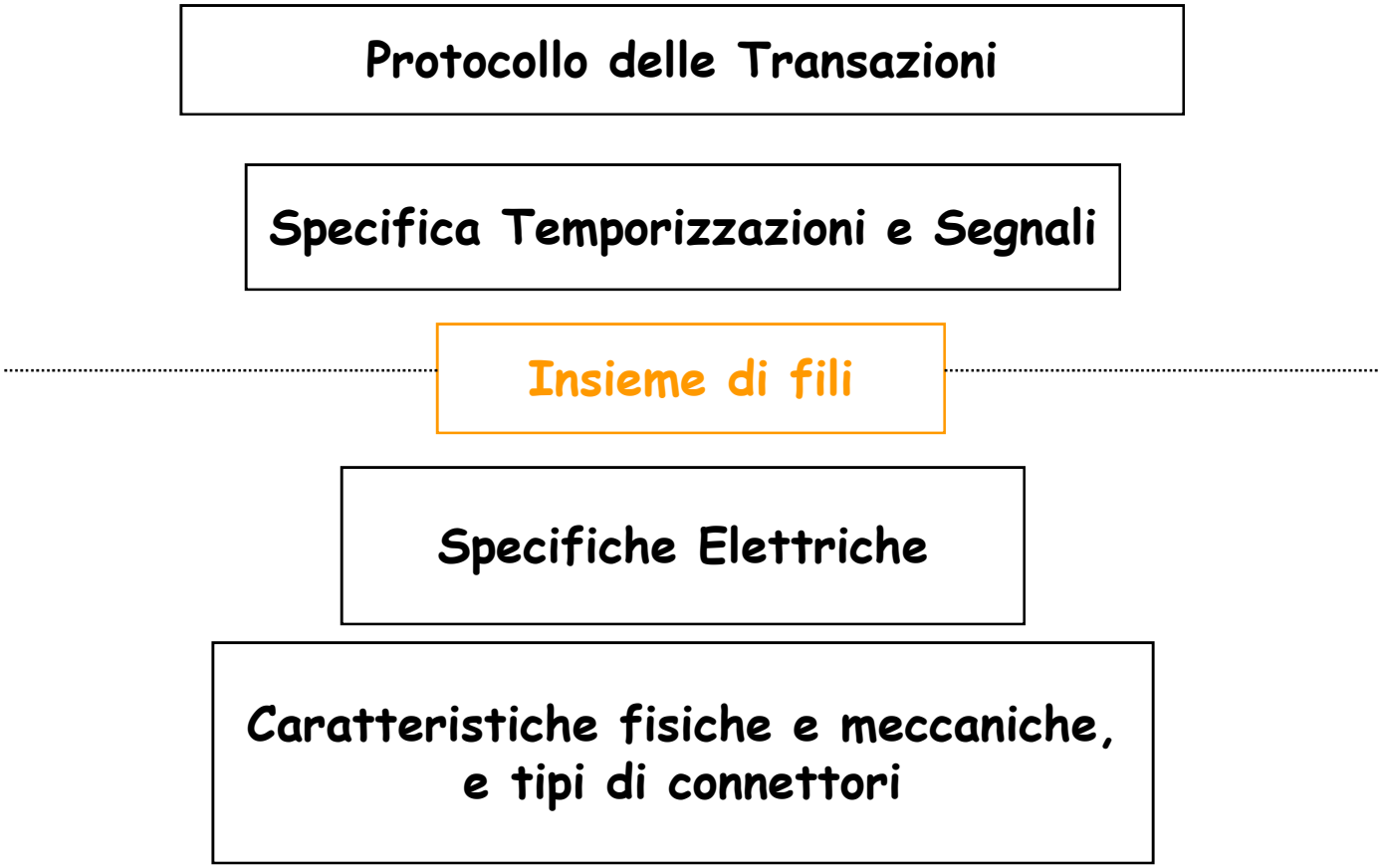
- I bus di I/O si connettono al bus processore-memoria attraverso degli "adattatori" o "bridge"
  - Bus processore-memoria: usato principalmente per il traffico processore-memoria
  - Bus di I/O: forniscono gli slot di espansione per i dispositivi di I/O
- Esempio: Macintosh-II della Apple
  - NuBus: collega(va) processore, memoria, e alcuni dispositivi di I/O
  - Bus SCSI: collega(va) il resto dei dispositivi di I/O

Sistema con tre bus



- Un basso numero di backplane bus si inserisce sul bus processore-memoria
  - Il bus processore-memoria e' usato per il traffico processore memoria
  - I bus di I/O sono connessi al backplane bus
- Vantaggio: il carico sul bus del processore e' fortemente ridotto

Come si definisce un bus?

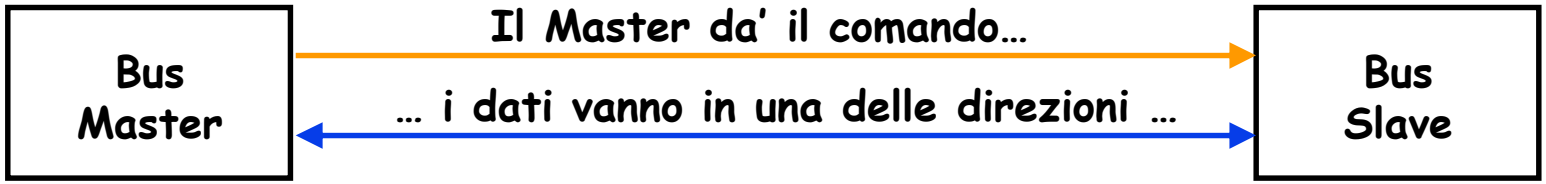


Organizzazione generale di un bus

- Linee di Controllo
  - Segnalano la richiesta (request) e la conferma (acknowledgment)
  - Indicano che tipo di informazione sta transitando sulle linee dati

- Linee Dati trasportano le informazioni fra chi invia (sorgente dati o source) e destinatario (destination)
  - Dati (veri e propri) e Indirizzi
  - Comandi complessi

Master/Slave e Transazioni



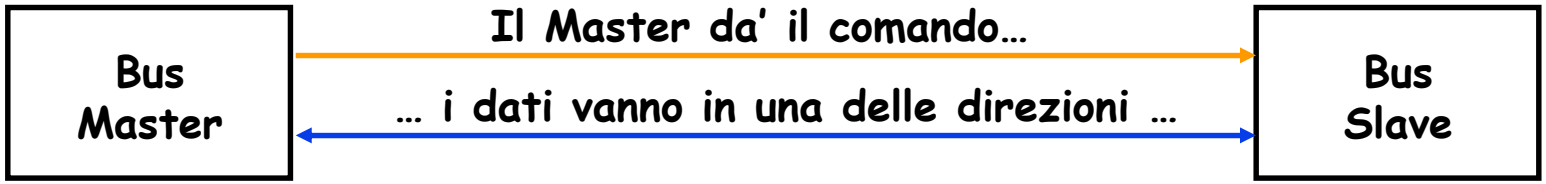
- Una transazione sul bus include tre fasi
  - Decidere chi e' master – fase di arbitraggio
  - Dare il comando (o l'indirizzo) – fase di richiesta
  - Trasferire i dati – fase di azione
- Il Master e' l'unita' che:
  - Ha il controllo del bus
  - Inizia la transazione dando un comando o specificando un'indirizzo
- Lo Slave e' l'unita' che:
  - Viene attivata dalla transazione ovvero risponde alla richiesta
  - Invia i dati al master, se il master richiede dati
  - Preleva i dati inviati dal master, se il master vuole mandare dati

Bus Sincroni e Bus Asincroni

- **Bus Sincrono**:
  - Fra le linee di controllo c'e' il segnale di clock
  - Il protocollo di comunicazione e' fisso ed e' riferito al clock
  - Vantaggio: richiede pochissima logica e va molto veloce
  - Svantaggi:
    - Ogni dispositivo sul bus deve andare alla stessa frequenza di clock
    - Per evitare il "clock skew" per avere bus lunghi devo abbassare le velocita'

- **Bus Asincrono**:
  - Sono sempre necessarie due linee di controllo (req, ack) per gestire il trasferimento (protocollo di handshaking)
  - Non ha bisogno del clock
  - Puo' collegarsi alla quasi totalita' dei dispositivi
  - Puo' essere anche abbastanza lungo senza problemi di "clock skew"

Arbitraggio: ottenere accesso al Bus

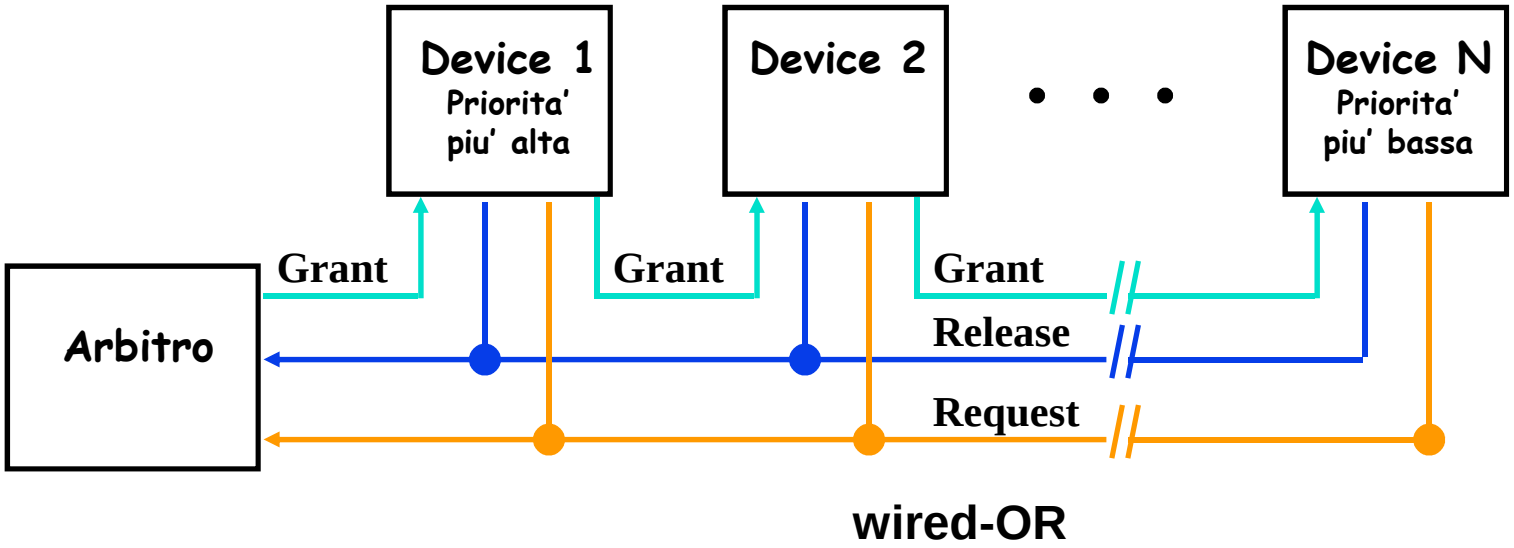


- E' uno dei problemi piu' importanti nella progettazione di un bus
  - Come viene prenotato il bus da un dispositivo che desidera usarlo?
- Innanzitutto per semplificare la situazione si deve arrivare alla definizione di chi e' il master e chi e' lo slave
  - A quel punto solo il bus-master potra' controllore l'accesso al bus
    - Tutte le successive richieste saranno iniziate e controllate da lui
  - Uno slave rispondera' alle richieste per leggendo o scrivendo qualcosa
- Il sistema piu' semplice, che usa questo schema, potrebbe funzionare cosi':
  - Il processore e' il solo bus-master
  - Tutte le richieste saranno controllate dal processore
  - Principale svantaggio: il processore e' coinvolto in ogni transazione

Bus-Master multipli: necessita' di Arbitraggio

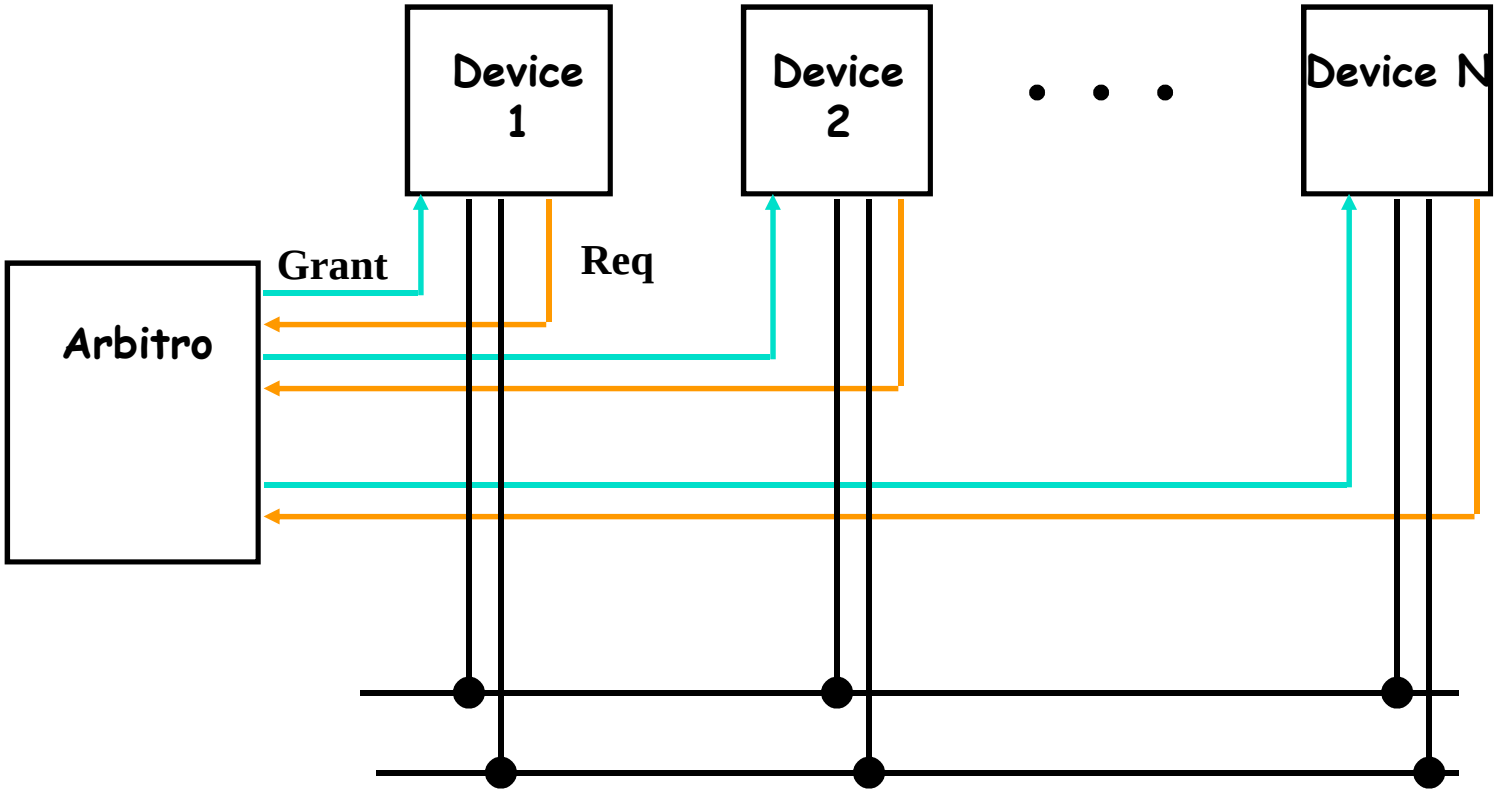
- Schema di arbitraggio del bus
  - Un bus-master che voglia usare il bus segnala la richiesta ad un Arbitro
  - Il bus-master non puo' usare subito il bus resta in attesa del permesso (grant)
  - Il bus-master dovra' infine segnalare all'Arbitro che ha finito di usare il bus
- L'Arbitro cerca di bilanciare due fattori
  - Priorita' della richiesta: il dispositivo a priorita' maggiore deve essere servito per primo
  - Fairness: anche se un dispositivo ha bassa priorita' bisogna garantire che questo possa ottenere il controllo del bus entro un tempo ragionevole
- Categorie principali degli schemi di arbitraggio
  - Arbitraggio Centralizzato Daisy-Chain
  - Arbitraggio Centralizzato Parallelo
  - Arbitraggio Distribuito con Auto-Selezione
  - Arbitraggio Distribuito con Rilevamento delle Collisioni

Arbitraggio Centralizzato Daisy Chain



- Vantaggio: semplice
- Svantaggi:
  - Non e' possibile garantire la fairness:
    - Un dispositivo a valle potrebbe rimanare bloccato indefinitamente
  - Il tempo di propagazione del segnale di grant puo' limitare la velocita' del bus
- Es.: VMEbus

Arbitraggio Centralizzato Parallelo

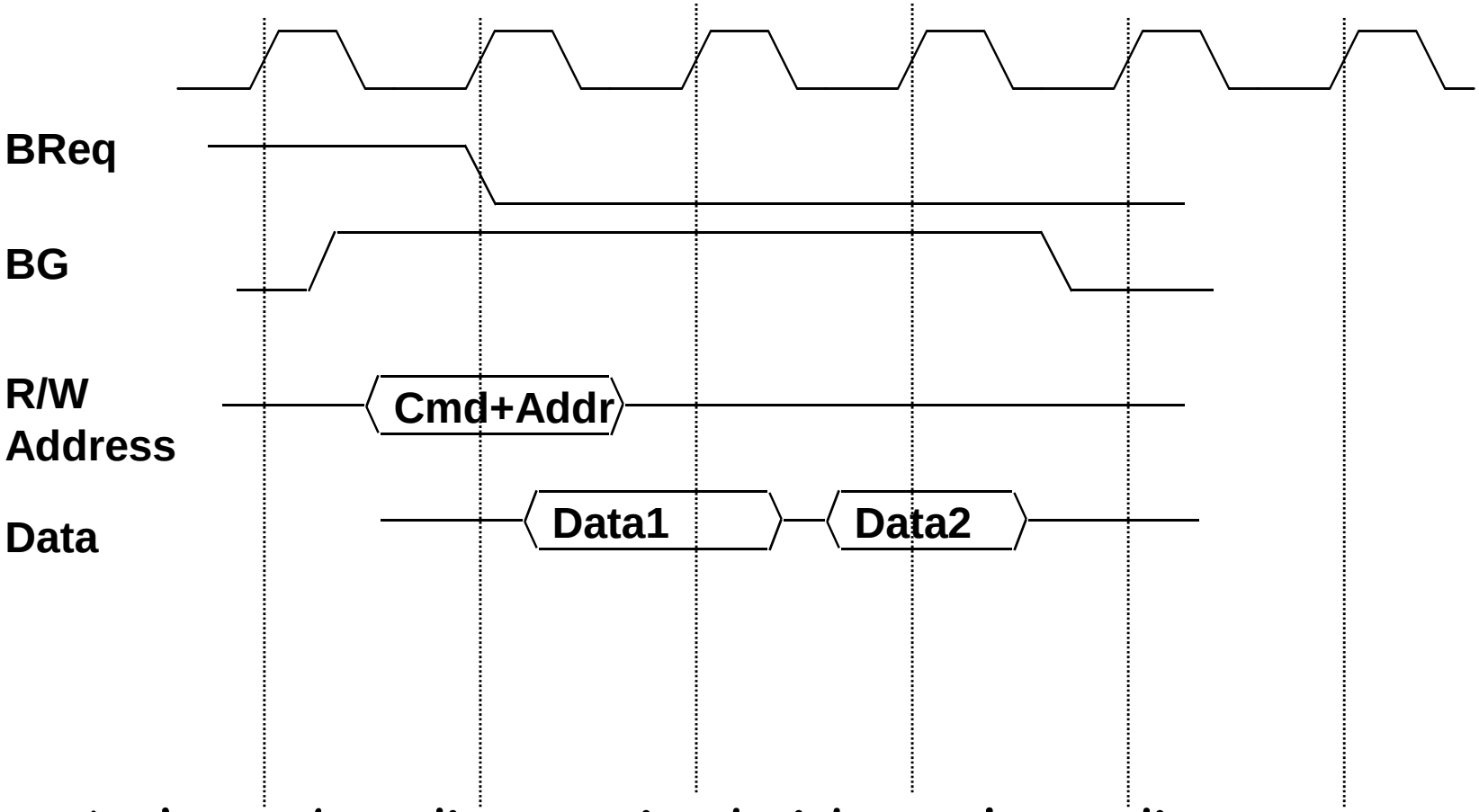


- Usato nella quasi totalita' dei bus processore-memoria e nei bus di I/O ad alta velocita'
- Inoltre e' usato nei bus: Multibus I, EISA, PCI

Arbitraggio Distribuito

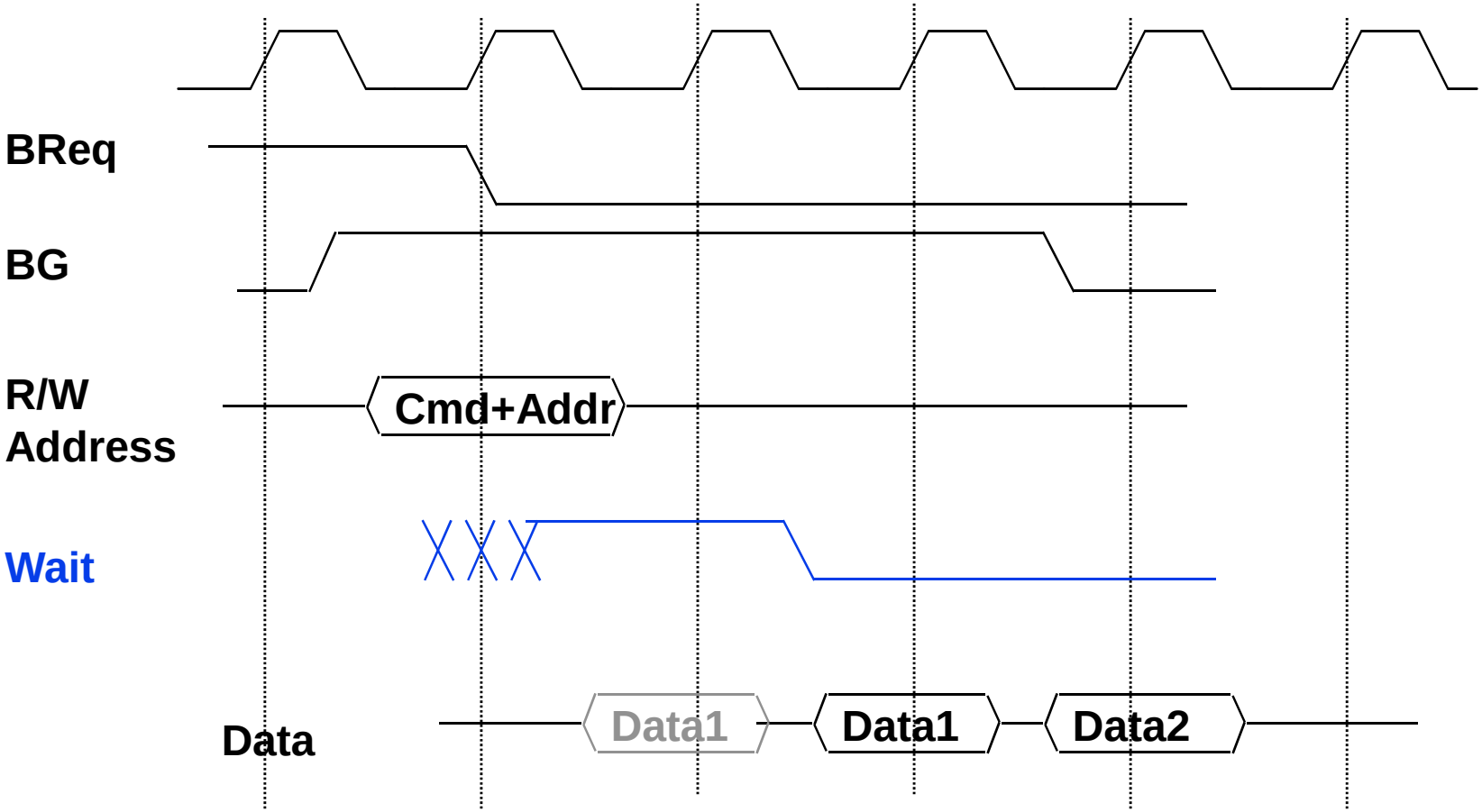
- La scelta del Master non viene fatta da una entita' precisamente identificabili
- Esempi:
  - Multibus I: arbitraggio distribuito in daisy-chain
    - Inizialmente il bus e' controllato da un'unita' a priorita' intermedia M
    - La richiesta da parte di un'unita' A a priorita' piu' alta attiva la logica per "convincere" M al rilascio del bus
  - Ethernet: arbitraggio distribuito con rilevamento delle collisioni
    - Inizialmente un'unita' detiene il controllo del bus ma lo fara' solo per un tempo limitato
    - Un potenziale master prova a trasmettere: se il bus e' libero bene, altrimenti riprova piu' tardi
    - Il tempo di attesa si calcola con una legge di decadimento casuale esponenziale, per evitare che si ripeta la stessa combinazione di unita' che tentano l'accesso

Protocollo Sincrono (Semplice)



- Anche un bus di memoria e' piu' complesso di questo
  - La memoria (slave) puo' prendere piu' cicli per rispondere
  - Dovra' quindi segnalare che non e' in grado di accettare indirizzi...

Protocollo Sincrono (Tipico)



- Con WAIT lo slave indica che e' pronto per il traferimento dei dati (es. lettura)…
- Poi il trasferimento ha luogo alla velocita' del bus

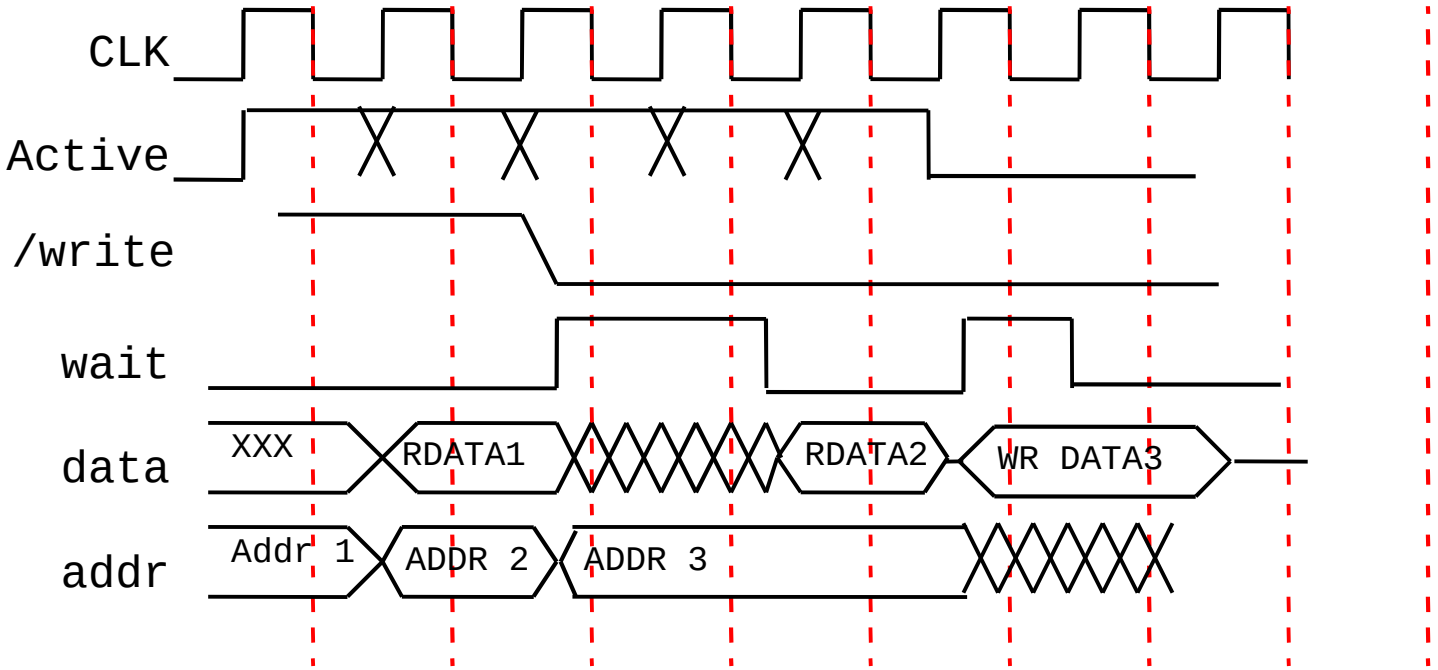
Come aumentare la banda del bus

- Uso di linee indirizzi/dati separate anziche' multiplexate
  - posso trasmettere nello stesso ciclo indirizzo e dati
  - Svantaggi: (a) necessito di piu' fili, (b) aumenta la complessita'
- Aumentare la larghezza del bus dati
  - Il trasferimento di piu' word richiedera' meno cicli di bus
  - Esempio: nella SPARCstation 20 il bus di memoria e ' a 128 bit
  - Svantaggio: ancora necessito di piu' fili
- Trasferimento a blocchi (burst transfer)
  - Evito di ripetere l'indirizzo quando trasferisco K word successive
  - Specifico solo l'indirizzo della prima word
  - Il bus non viene rilasciato finche' l'ultima word non e' arrivata
  - Svantaggi: (a) maggiore complessita', (b) aumento del tempo di risposta nei casi di singola richiesta

Protocolli basati sul concetto di "Pipeline"

Durante la fase di accesso ai dati inizio gia' a specificare
l'indirizzo del trasferimento successivo

Esempio con singolo master (processore-cache)



Aumento delle Transazioni su Bus Multimaster

- Arbitraggio sovrapposto (overlapped)
  - Si guadagna tempo effettuando la fase di arbitraggio per la transazione successiva a quella in corso
- "Bus parking"
  - Il master mantiene il bus ed effettua varie transazioni senza passare nuovamente alla fase di arbitraggio fino a che qualche altro master non si fa avanti
- Sovrapposizione delle fasi di indirizzamento e accesso
  - Visto nella pagina precedente
- "Split-phase" (o "packet switched") bus
  - Fasi di indirizzamento e accesso completamente separata
  - Ognuna necessita di un arbitraggio separato
  - Durante l'indirizzamento si produce anche un identificatore (tag) del pacchetto che verra' associato ai dati in fase di risposta per permettere l'abbinamento indirizzo-dato
- Nei bus moderni si usano tutte le tecniche precedenti combinate fra loro

Bus di Memoria per server multiprocessore (1993)

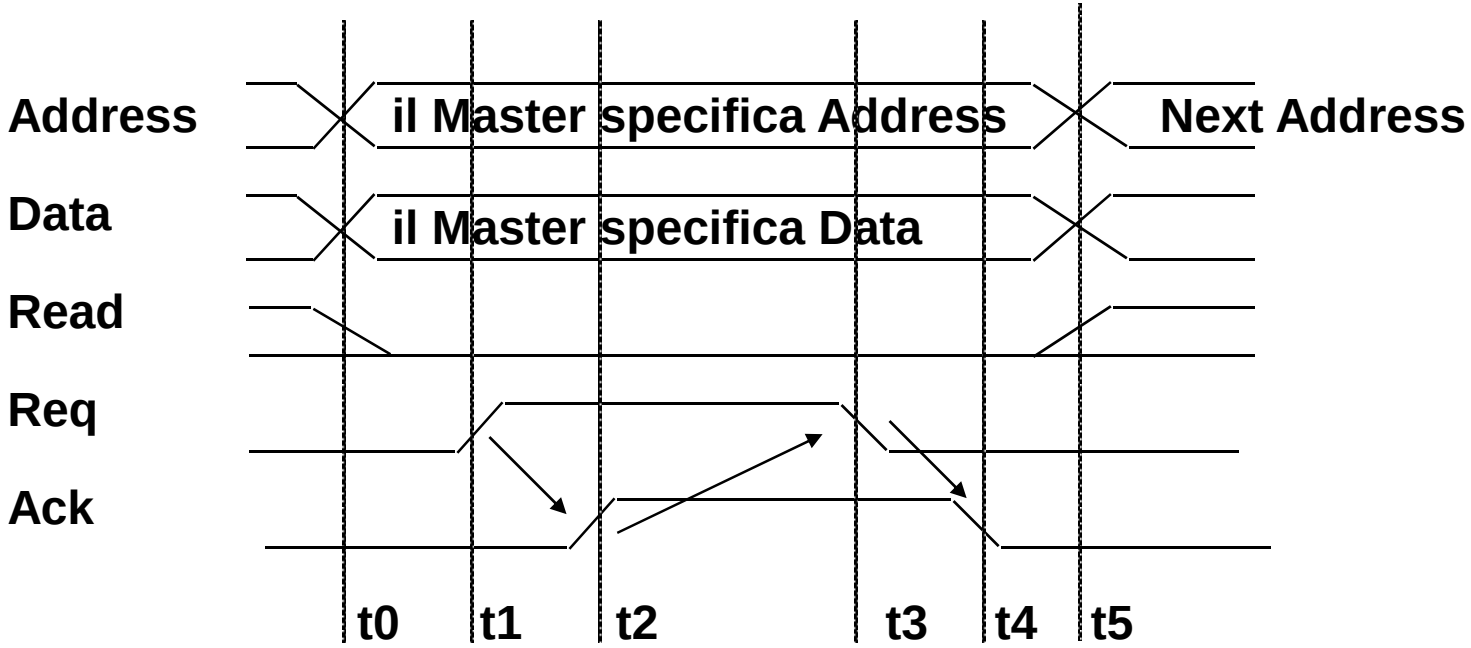
Nome_bus	MBus	Summit	Challenge	XDBus
Ditta	Sun	HP	SGI	Sun
Clock (MHz)	40	60	48	66
Linee Indirizzi	36	48	40	muxed
Linee Dati	64	128	256	144 (parity)
Bit dati	256	512	1024	512
Clock/trasferim.		4	5	4?
Picco (MB/s)	320(80)	960	1200	1056
Master	Multi	Multi	Multi	Multi
Arbitraggio	Centrale	Centrale	Centrale	Centrale
Slots		16	9	10
Busses/system	1	1	1	2
Lunghezza		~30 cm	~30 cm	~35 cm

Bus di I/O

- Progettato per supportare una vasta gamma di dispositivi
  - L'insieme dei dispositivi si potra' evolvere nel tempo

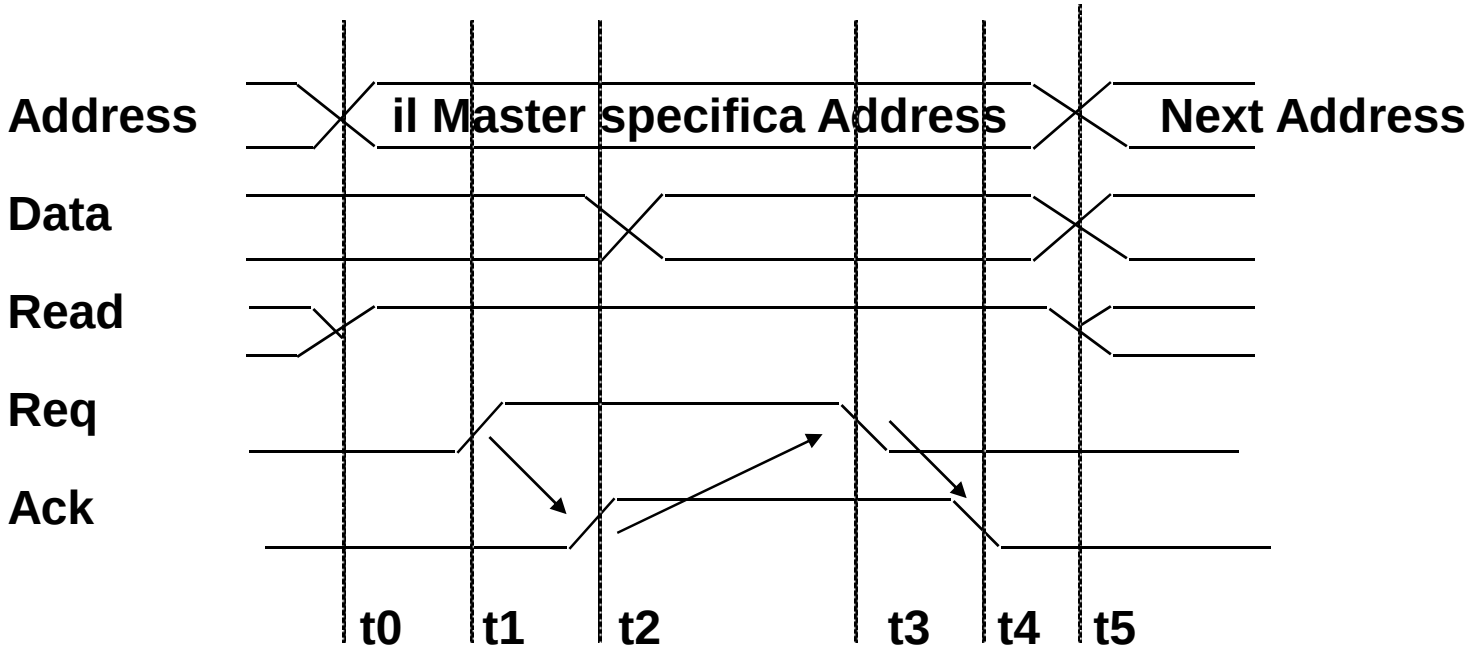
- E' necessario poter gestire velocita' di comunicazione arbitrarie e situazioni con:
  - Processore veloce – I/O lento
  - Processore lento - I/O veloce

Handshake Asincrono: Scrittura



- t0 : Il Master ha ottenuto il controllo e specifica l'indirizzo, la direzione e dati; attende poi un certo intervallo di tempo affinche' lo Slave capisca di essere coinvolto in quel trasferimento…
- t1: Il Master attiva la linea "Request"
- t2: Lo Slave attiva la linea "Ack", per indicare di aver ricevuto il dato → ha finito
- t3: Il Master rilascia "Req" → ha capito che lo Slave ha finito
- t4: Lo Slave rilascia "Ack" → ha capito che il Master ha capito

Handshake Asincrono: Lettura



- t0 : Il Master ha ottenuto il controllo e specifica l'indirizzo, la direzione e i dati; attende poi un certo intervallo di tempo affinche' lo Slave capisca di essere coinvolto in quel trasferimento…
- t1: Il Master attiva la linea "Request"
- t2: Lo Slave attiva la linea "Ack", per indicare che e' pronto a trasmettere il dato
- t3: Il Master rilascia "Req" avendo ricevuto il dato → ha finito
- t4: Lo Slave rilascia "Ack" → ha capito che il Master ha finito

Backplane/IO Bus (1993)

Nome_Bus	SBus	TurboChannel	MicroChannel	PCI
Ditta	Sun	DEC	IBM	Intel
Clock (MHz)	16-25	12.5-25	async	33
Indirizzamento	Virtuale	Fisico	Fisico	Fisico
Bit dati	8,16,32	8,16,24,32	8,16,24, 32,64	8,16,24, 32,64
Master	Multi	Singolo	Multi	Multi
Arbitraggio	Centrale	Centrale	Centrale	Centrale
32bit read (MB/s)	33	25	20	33
Picco (MB/s)	89	84	75	111 (222)
Potenza max (W)	16	26	13	25

Bus di I/O ad alta velocita'

- Esempi
  - grafica
  - Rete ad alta velocita'

- Consentono di collegare un basso numero di dispositivi
- I trasferimenti dati viaggiano sempre a velocita' massima
- Si adottano tecniche particolari di trasferimento come il DMA (Direct Memory Access)
  - Un piccolo processore pompa byte da/verso la memoria
- Ognuna delle due parti puo' aver bisogno di bloccare momentaneamente il trasferimento
  - i buffer che prima o poi si riempiono

Riepilogo degli caratteristiche dei bus

Caratteristica	*Obiettivo=prestazioni*	*Obiettivo=basso_costo*
Larghezza Bus	linee indirizzi e dati separate	linee indirizzi e dati multiplexate
Dim. Dati	larga=velocita' (e.g., 32 bit)	bassa=basso_costo (e.g., 8 bit)
Dim. Trasfer.	Word multiple → meno gestione bus	trasferim. a singola word sono piu' sempici
Bus-masters	Multipli (rechiede arbitraggio)	Singolo master (nessun arbitraggio)
Clock	Sincrono	Asincrono
Protocollo	con "pipeline"	seriale

Bus Punto-Punto

- Bus utilizzati per collegamenti ad altissima velocita', ad es. (anche nei PC):
  - HYPERTRANSPORT™ – dal 2001 (www.hypertransport.org)
    - Es. Processori AMD
  - QUICKPATH™ – dal 2008
    - Es. Processori Intel (Nahelem e successivi)

- Velocita' di trasferimento: 30/40 GB/s