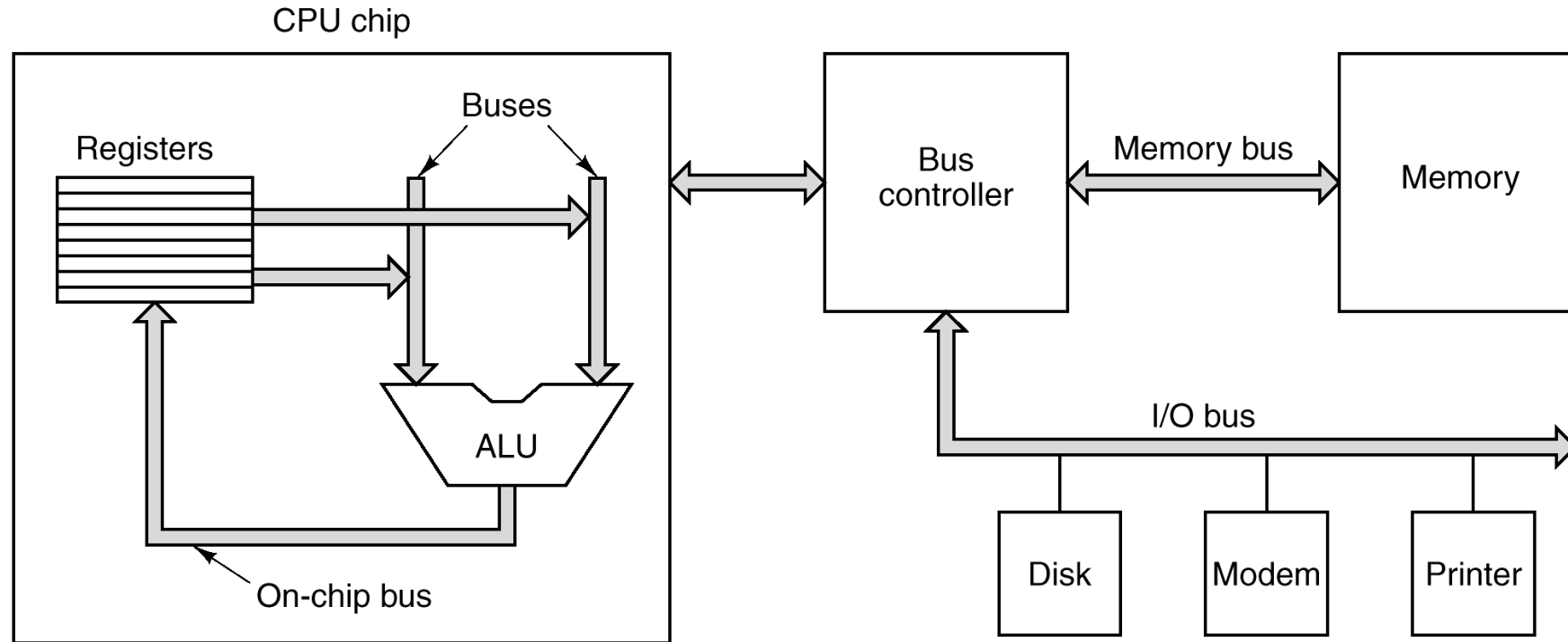


# Calcolatori Elettronici

## Parte V: Bus

Prof. Riccardo Torlone  
Universita di Roma Tre

# Architettura a più Bus

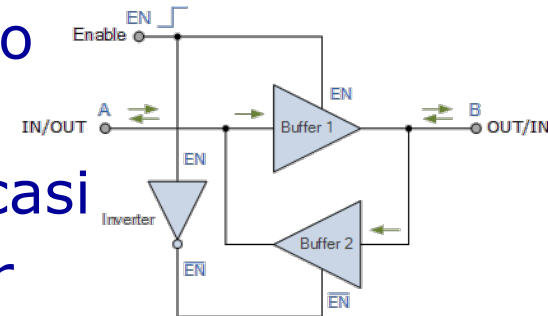


- Diversi bus, interni ed esterni al chip
- Soddisfano diverse esigenze:
  - Velocità di trasferimento
  - Numero di linee
  - Compatibilità all'indietro
- Negli attuali PC almeno due bus esterni

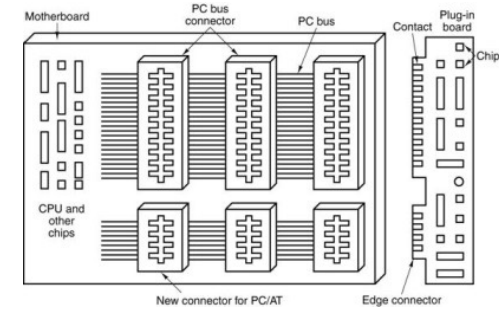
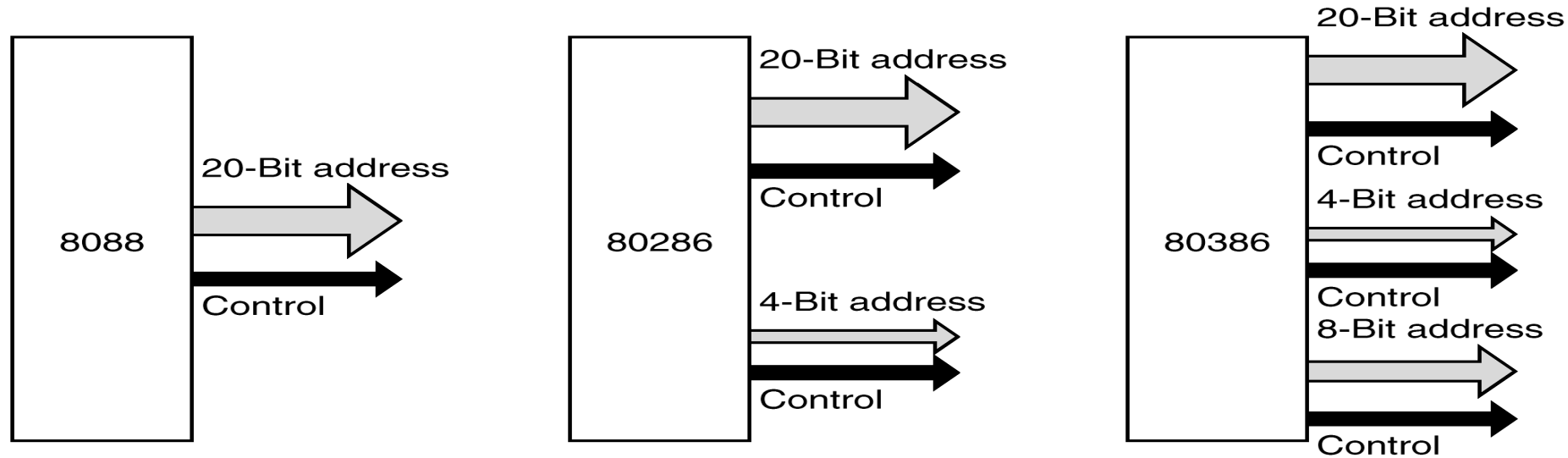
# Comunicazione sul Bus

Master	Slave	Example
CPU	Memory	Fetching instructions and data
CPU	I/O device	Initiating data transfer
CPU	Coprocessor	CPU handing instruction off to coprocessor
I/O	Memory	DMA (Direct Memory Access)
Coprocessor	CPU	Coprocessor fetching operands from CPU

- La comunicazione sul bus è regolata da un protocollo di bus
- In ciascun ciclo comunicano due soli dispositivi il master e lo slave
- Lo stesso dispositivo può avere ruoli diversi a seconda dei casi
- I dispositivi sono connessi al bus tramite un bus transceiver
- La connessione al bus o avviene tramite dispositivi a tre stati oppure è di tipo open collector



# "Larghezza" del Bus



- Larghezza = numero di linee
- Linee indirizzo: dimensione dello spazio (di memoria) indirizzabile,  $2^n$  locazioni con n bit di indirizzo
- Linee + velocità di trasmissione: banda di trasferimento
- Condivisione di più segnali sulla stessa linea per diminuire i costi
- **Problema:** al crescere della velocità del bus aumenta il bus skew (differenza nella velocità di propagazione dei segnali su linee diverse)

# Segnali asseriti e negati

In alcuni casi (a seconda delle scelte di progetto) un segnale provoca l'azione corrispondente quando la sua tensione è alta (1), in altri quando è bassa (0).

Per evitare confusione si parla di:

- Segnale asserito: quando assume il valore (alto o basso) che provoca l'azione
- Segnale negato: altrimenti

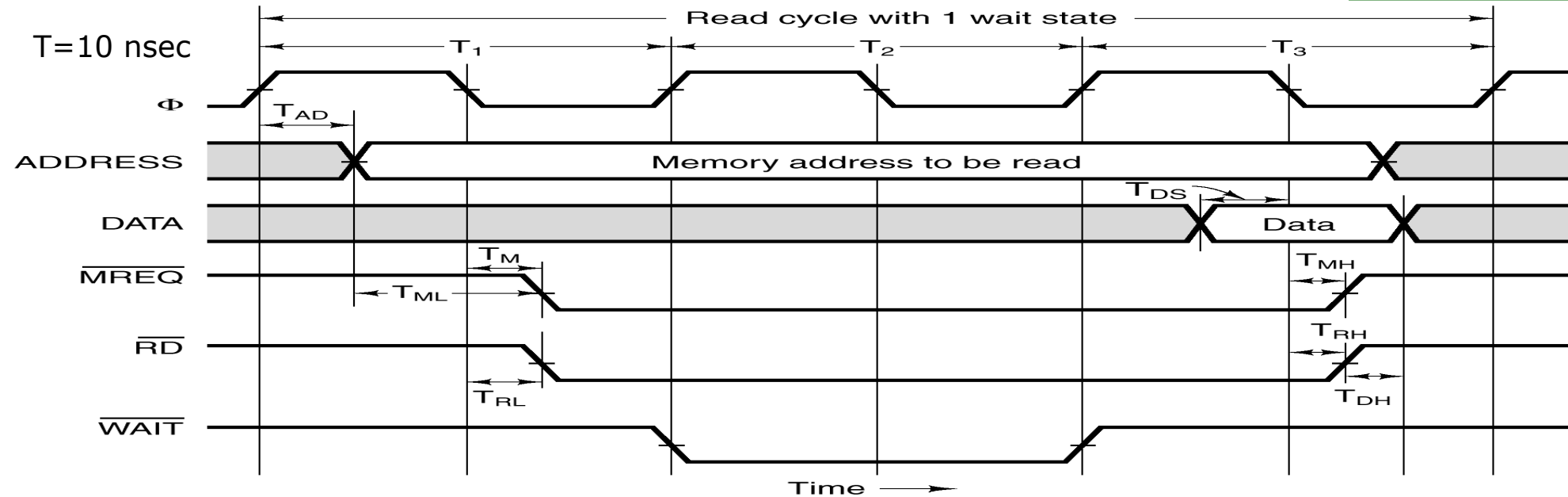
Si adotta la seguente notazione:

- $S$ : segnale che è asserito alto
- $\overline{S}$ : segnale che è asserito basso

Ulteriore notazione (usata da Intel):

- $S$ : segnale che è asserito alto
- $S\#$ : segnale che è asserito basso
  - (adatta al set di caratteri ASCII)

# Bus Sincroni: ciclo di lettura



Symbol	Parameter	Min	Max	Unit
$T_{AD}$	Address output delay		4	nsec
$T_{ML}$	Address stable prior to $\overline{\text{MREQ}}$	2		nsec
$T_M$	$\overline{\text{MREQ}}$ delay from falling edge of $\Phi$ in $T_1$		3	nsec
$T_{RL}$	$\overline{\text{RD}}$ delay from falling edge of $\Phi$ in $T_1$		3	nsec
$T_{DS}$	Data setup time prior to falling edge of $\Phi$	2		nsec
$T_{MH}$	$\overline{\text{MREQ}}$ delay from falling edge of $\Phi$ in $T_3$		3	nsec
$T_{RH}$	$\overline{\text{RD}}$ delay from falling edge of $\Phi$ in $T_3$		3	nsec
$T_{DH}$	Data hold time from negation of $\overline{\text{RD}}$	0		nsec

# Bus Sincrono: Temporizzazione

ES

Frequenza 100 MHz, periodo 10 nsec.

- primo vincolo: tempo a disposizione della memoria fra:
  - la comparsa dell'indirizzo sul Bus
  - la disponibilità dei dati sul Bus

$$\tau_1 = 2.5 \times T - T_{AD} - T_{DS} = 25 - 4 - 2 = 19 \text{ nsec}$$

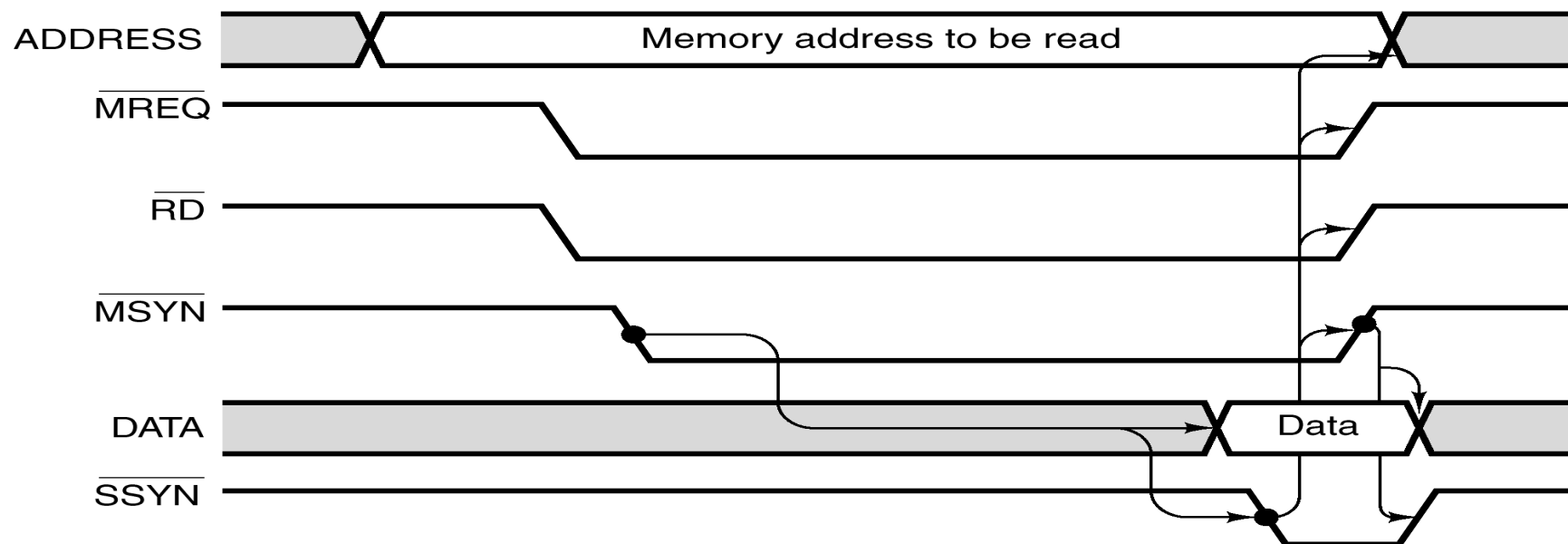
(una memoria da 10 nsec ce la fa di sicuro)

- secondo vincolo: tempo a disposizione della memoria fra:
  - l'asserzione di  $\overline{\text{MREQ}}$  e  $\overline{\text{RD}}$
  - la disponibilità dei dati sul Bus

$$\tau_2 = 2 \times T - T_M - T_{DS} = 20 - 3 - 2 = 15 \text{ nsec}$$

Se il chip di memoria non soddisfa questi requisiti mantiene asserito il segnale di  $\overline{\text{WAIT}}$  per introdurre stati di wait, cioè cicli di bus aggiuntivi.

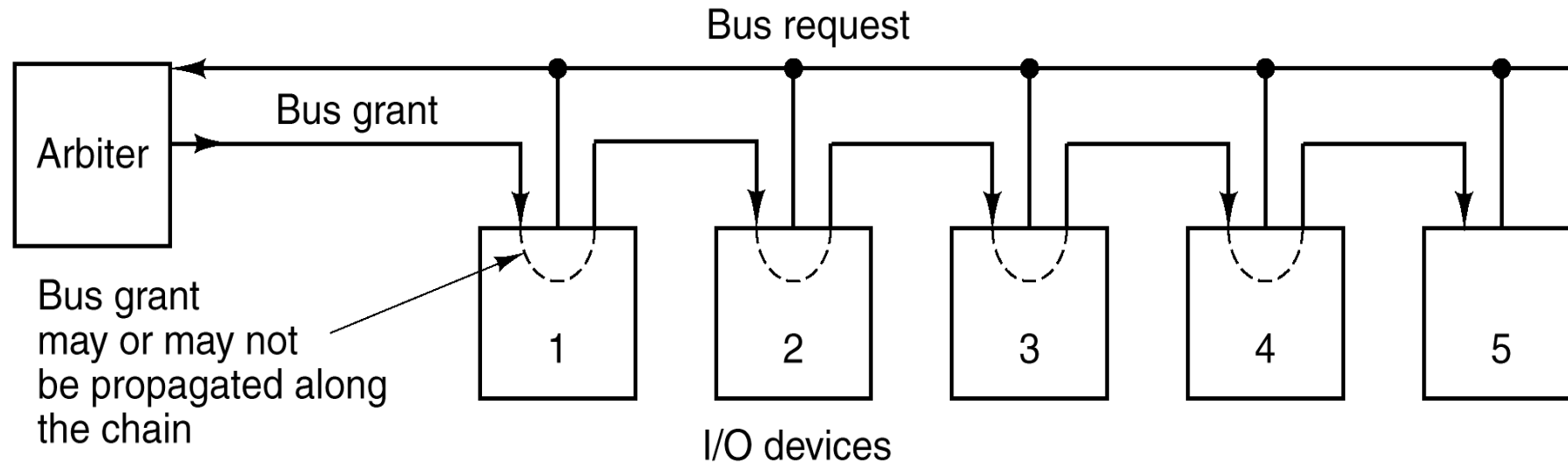
# Bus Asincrono: ciclo di lettura



- Accoppiamento di dispositivi con velocità diverse
- Gli eventi avvengono in risposta ad altri eventi
- Full handshake:
  - $\overline{\text{MSYN}}$  asserito
  - $\overline{\text{SSYN}}$  asserito in risposta a  $\overline{\text{MSYN}}$  quando il dato è pronto
  - $\overline{\text{MSYN}}$  negato in risposta a  $\overline{\text{SSYN}}$
  - $\overline{\text{SSYN}}$  negato in risposta alla negazione di  $\overline{\text{MSYN}}$

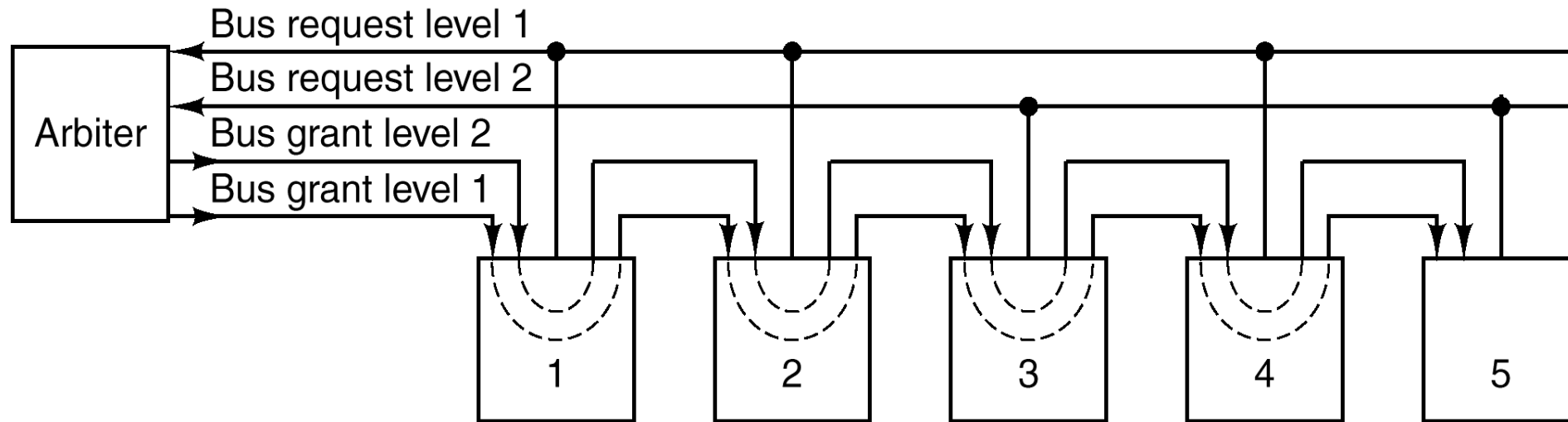


# Arbitraggio del Bus



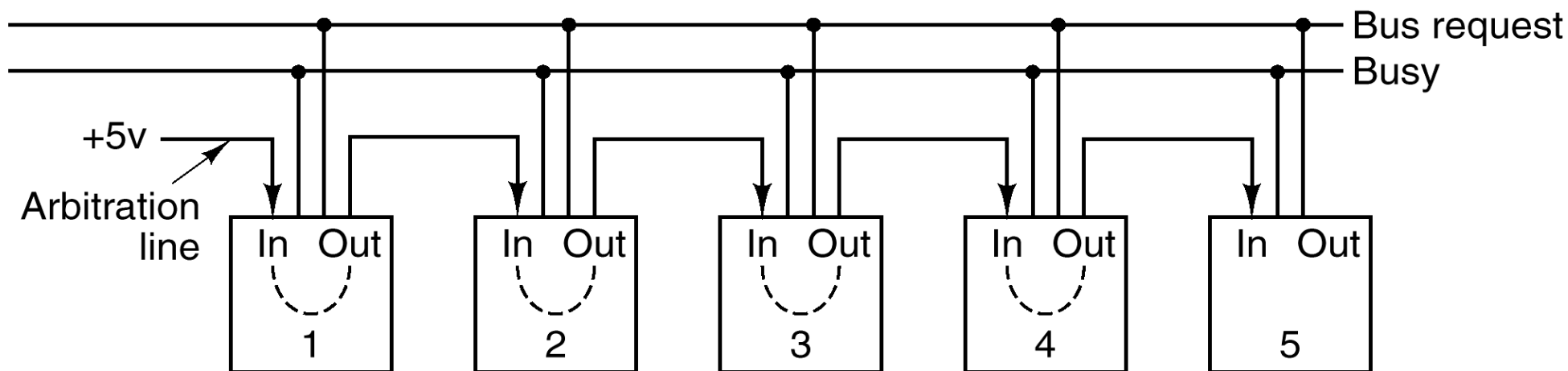
- Permette di decidere quale dispositivo sarà il prossimo Bus Master resolvendo eventuali conflitti
- Spesso l'arbitro è nel chip del microprocessore
- Linea di richiesta condivisa
- Il Bus grant è propagato dall'arbitro prima dell'inizio del ciclo
- Viene intercettato dal futuro master
- NB: Favoriti i dispositivi situati vicino all'arbitro

# Livelli Multipli di priorità



- Diverse linee di richiesta associate a diversi livelli di priorità
- In caso di conflitto favorite le catene a priorità più alta
- All'interno di ciascuna catena vale la posizione
- In genere se c'è un solo bus con anche la memoria, la CPU ha priorità più bassa dei dispositivi di I/O (e.g. dischi)

# Arbitraggio Decentralizzato



- Quando nessun dispositivo vuole il Bus, la linea di arbitraggio è asserita con propagazione a tutti i dispositivi
- Quando un dispositivo vuole il Bus:
  - invia una richiesta di bus
  - verifica se il bus è libero
  - se In è asserito diventa master, nega Out e asserisce Busy
  - se In non è asserito non diventa master e nega Out
- Non necessita di arbitro, è più semplice e più veloce

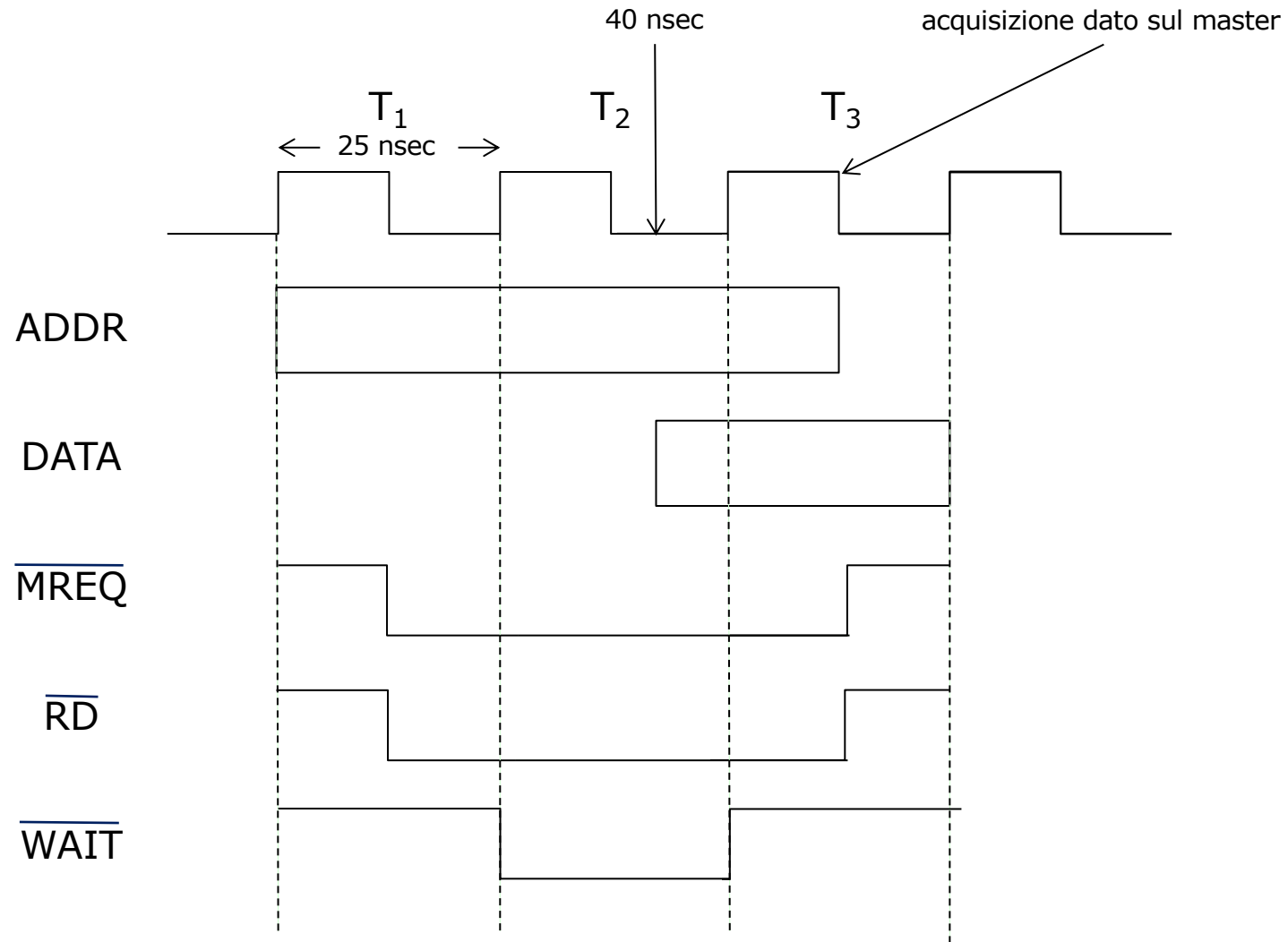
## Esercizio su BUS

Con riferimento al funzionamento dei bus di un calcolatore:

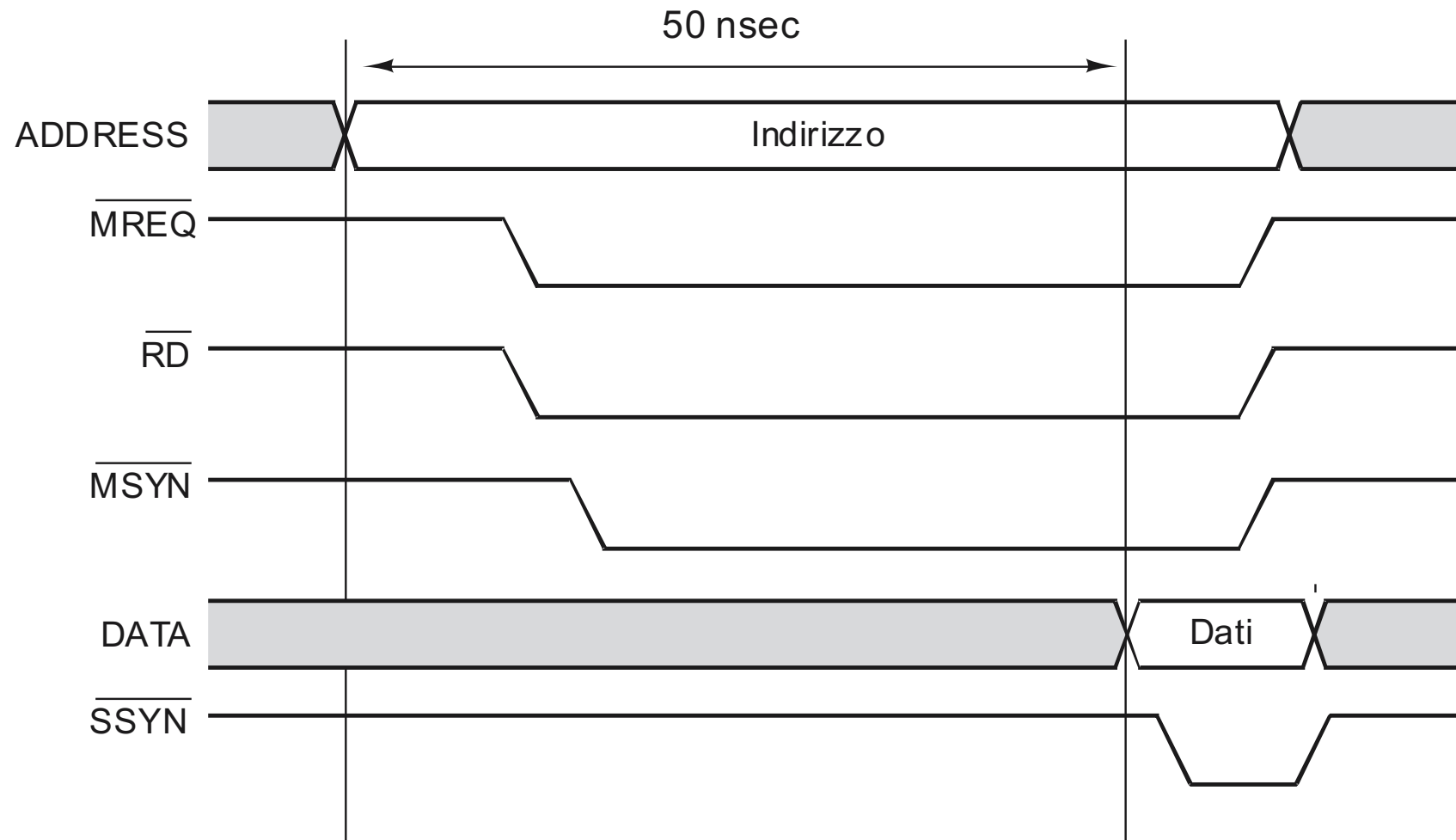
- tracciare e illustrare il diagramma di temporizzazione di un bus sincrono a 40 Mhz *con linee separate per dati e indirizzi* e segnali di MREQ, RD e WAIT, per una lettura da una memoria con un tempo di risposta di 40 nsec *dal momento in cui gli indirizzi sono disponibili*
- tracciare e illustrare il diagramma di temporizzazione di un bus asincrono con linee separate per dati e indirizzi per una scrittura in una memoria con un tempo di risposta di 50 nsec.

Si assuma di lavorare in condizioni ideali (nessun ritardo nell'asserimento dei segnali)

# Soluzione esercizio su BUS sincrono



# Soluzione esercizio su BUS asincrono



## Esercizio su BUS

Con riferimento al funzionamento dei bus di un calcolatore:

- tracciare e illustrare il diagramma di temporizzazione di un bus sincrono con linee separate per dati e indirizzi che lavora alla frequenza di 50 Mhz, per una lettura da un dispositivo I/O con un tempo di risposta di 100 nsec *dal momento in cui gli indirizzi sono disponibili*;
- tracciare e illustrare il diagramma di temporizzazione di un bus asincrono con linee separate per dati e indirizzi per una scrittura in una memoria con un tempo di risposta di 30 nsec *dal momento in cui il segnale di master synchronization è stato asserito*.

# Esercizio sull'arbitraggio di bus

Riferendosi agli schemi di arbitraggio dei bus discussi a lezione, indicare le affermazioni esatte tra le seguenti.

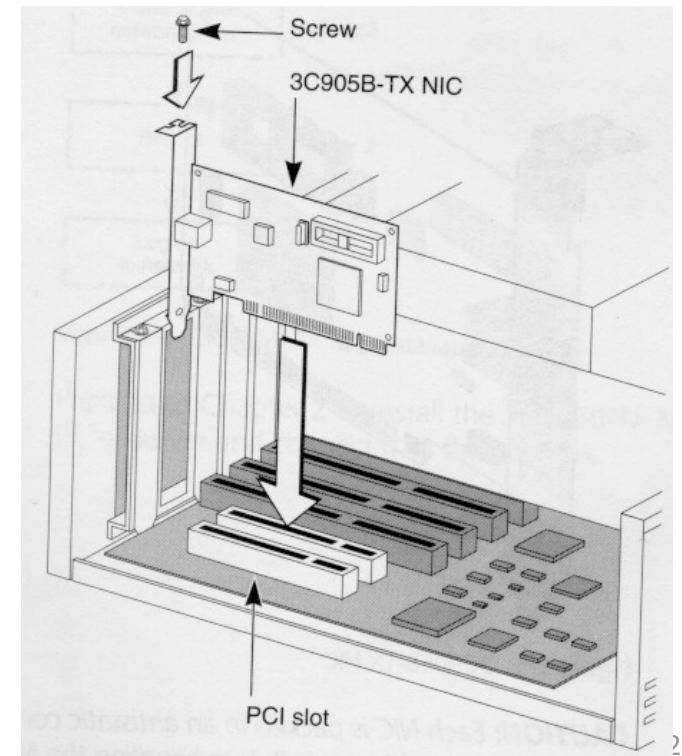
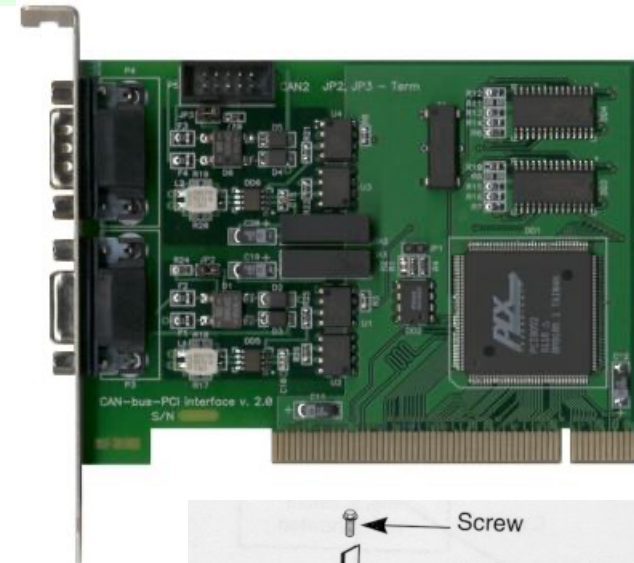
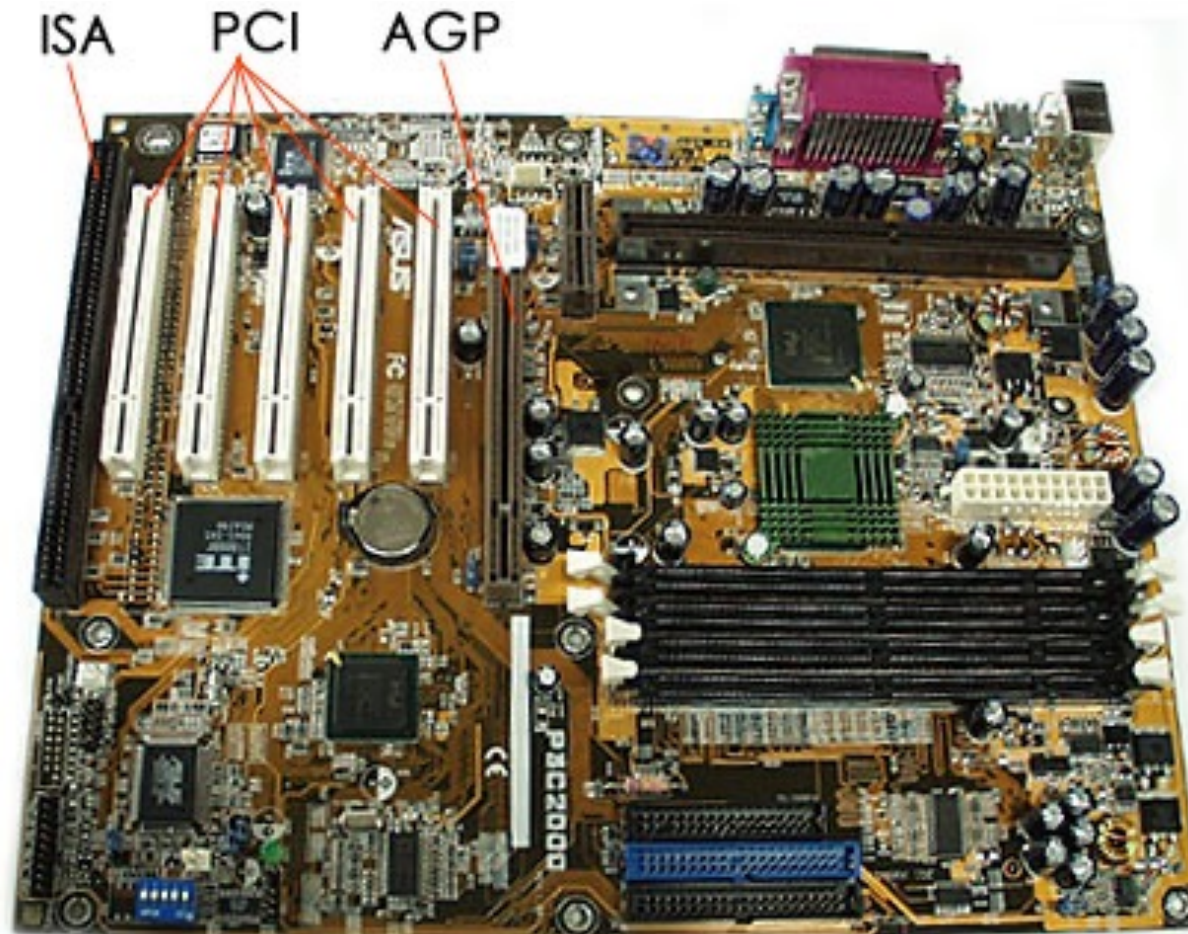
- Nell'arbitraggio decentralizzato non è necessaria la linea che segnala l'occupazione del bus. **FALSO**
- Nell'arbitraggio centralizzato con livelli di priorità la posizione del dispositivo non influisce sulla concessione del grant. **FALSO**
- Nell'arbitraggio centralizzato non è possibile che due dispositivi si prenotino contemporaneamente asserendo la linea di request. **FALSO**
- Nell'arbitraggio decentralizzato i tempi delle attese la posizione fisica dei dispositivi è comunque influenzata. **FALSO**
- Nell'arbitraggio centralizzato a più livelli di priorità un dispositivo può dover attendere un tempo indefinitamente lungo. **VERO**
- Per motivi di imparzialità l'arbitro è sempre un dispositivo diverso ed esterno al microprocessore. **FALSO**
- La priorità dei dispositivi I/O è generalmente più alta della CPU. **VERO**



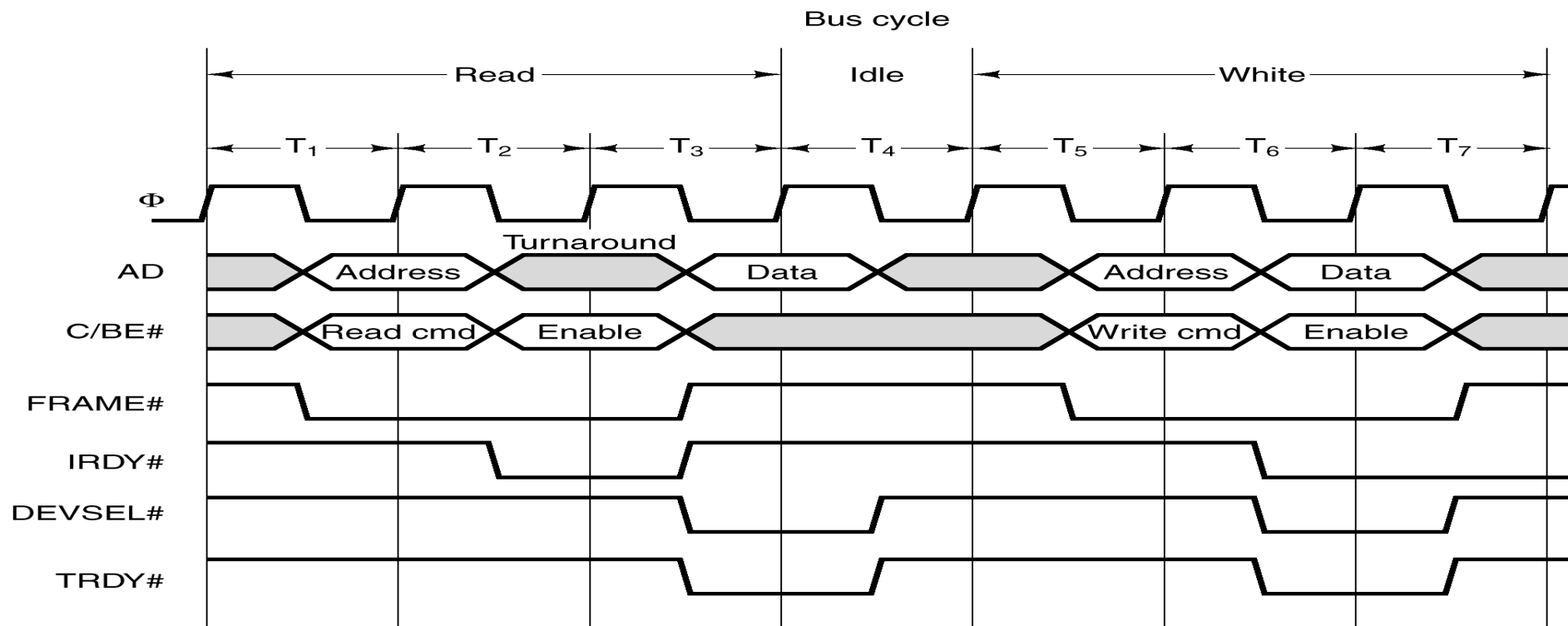
# Bus reali

- Requisiti:
  - Video a  $1024 \times 768$  con 3B per pixel (true color)
    - 1 frame: 2.25 MB
    - 30 frame al secondo: 67.5 MB/sec
    - HD  $\rightarrow$  RAM  $\rightarrow$  VRAM: 135 MB/sec
  - Video a  $1920 \times 1080$  con 3B per pixel
    - 1 frame:  $\sim 5.2$  MB
    - 30 frame al secondo: 155 MB/sec
    - HD  $\rightarrow$  RAM  $\rightarrow$  VRAM: 310 MB/sec
- Bus legacy:
  - ISA: 8.33Mhz, 2B per ciclo, 16.7MB/sec
  - EISA: 8.33Mhz, 4B per ciclo, 33.3MB/sec
- Bus PCI (1990, Intel): fino a 528 MB/sec
- Bus AGP (fine anni '90): fino a 2.1 GB/sec
- Bus PCIe (2004): 20GB/sec e oltre
- Bus USB (1995): 10Gb/sec e oltre

# Schede e slot PCI e AGP



# Bus PCI: Transazioni



- In T1 il master invia l'indirizzo su AD e il comando su C/BE#
- Poi asserisce FRAME# e poi IRDY#
- In T2 C/BE# specifica quali byte leggere
- In T3 lo slave asserisce DEVSEL# e quando i dati sono su AD asserisce TRDY#
- Tra due transazioni c'è un ciclo di idle
- La transazione di scrittura è più compatta

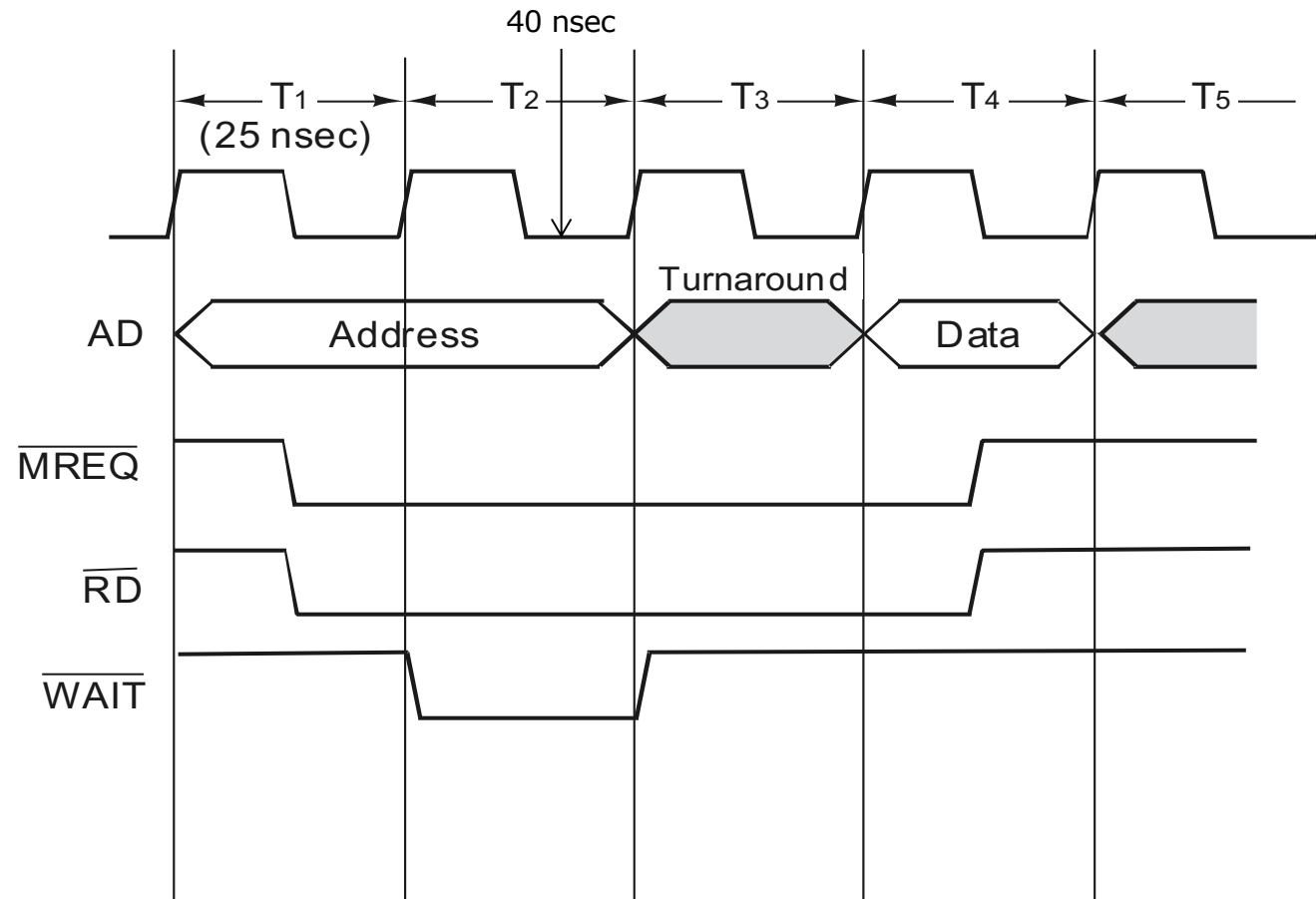
## Esercizio su BUS

Con riferimento al funzionamento dei bus di un calcolatore:

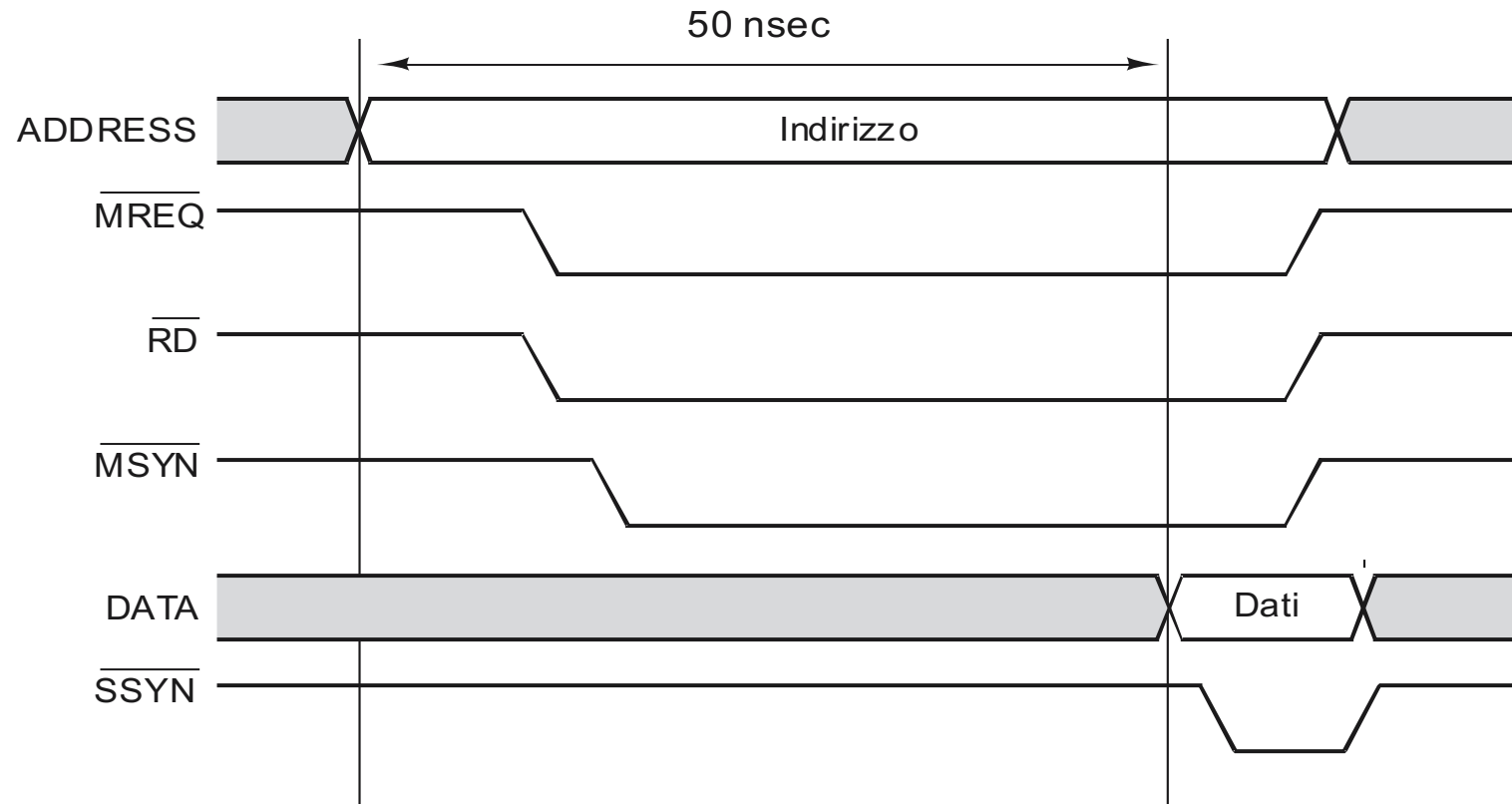
- tracciare e illustrare il diagramma di temporizzazione di un bus sincrono a 40 Mhz con linee *condivise* per dati e indirizzi e segnali di  $\overline{\text{MREQ}}$ ,  $\overline{\text{RD}}$  e  $\overline{\text{WAIT}}$ , per una lettura da una memoria con un tempo di risposta di 40 nsec *dal momento in cui gli indirizzi sono disponibili*
- tracciare e illustrare il diagramma di temporizzazione di un bus asincrono con linee separate per dati e indirizzi e segnali di  $\overline{\text{MREQ}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{MSYN}}$  e  $\overline{\text{SSYN}}$ , per una lettura da una memoria con un tempo di risposta di 50 nsec *dal momento in cui gli indirizzi sono disponibili*

Si assuma in entrambi i casi di lavorare in condizioni ideali (nessun ritardo nell'asserimento dei segnali)

# Soluzione esercizio su BUS sincrono



# Soluzione esercizio su BUS asincrono





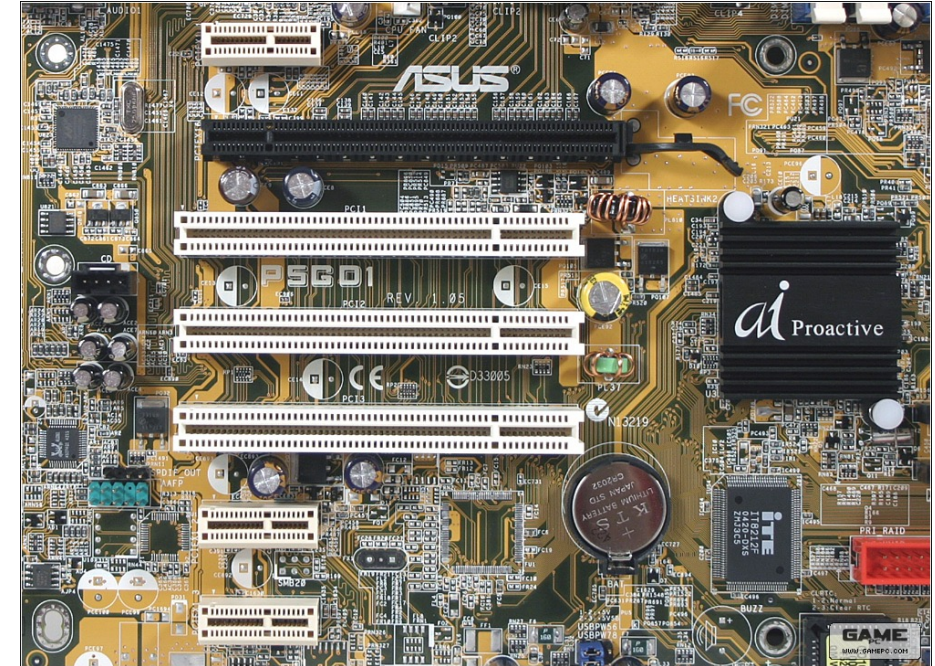
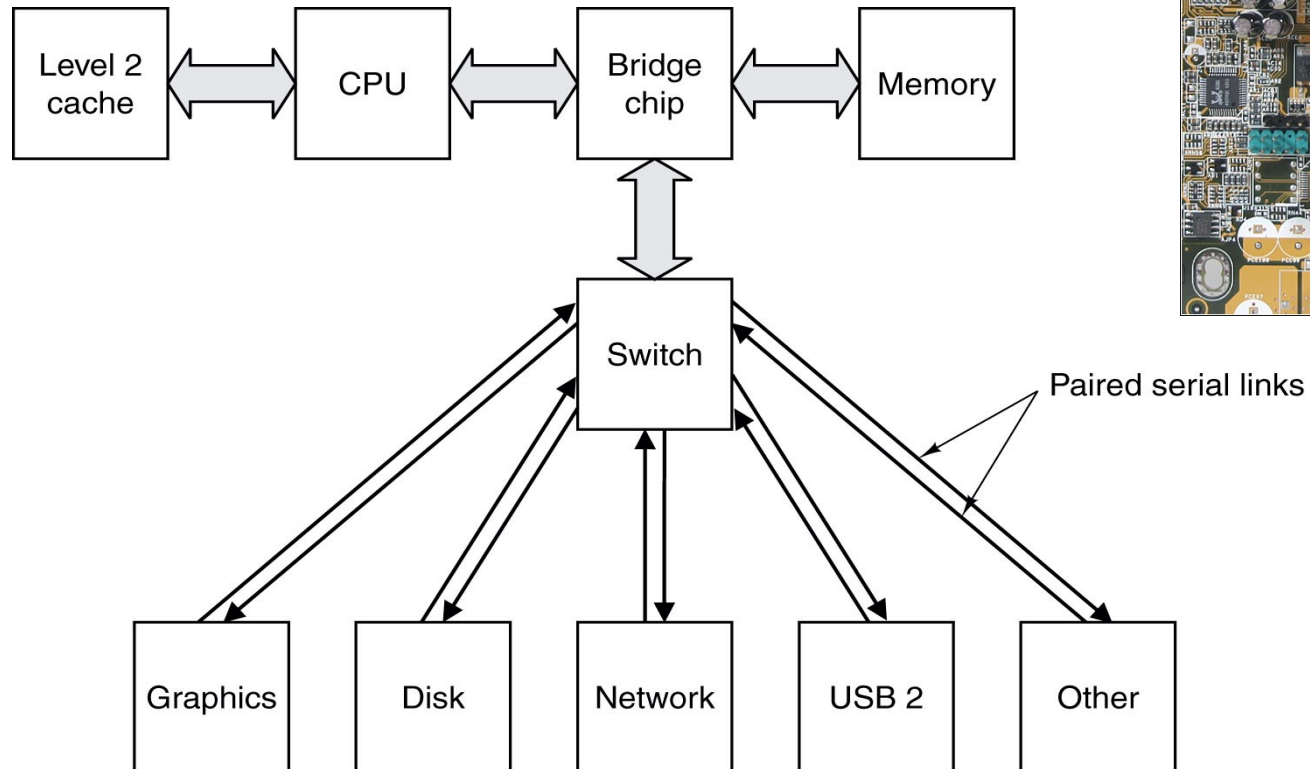
# PCI express

Linea di tendenza nei bus:

- bypassare il bus PCI nel caso di periferiche veloci
- comunicazione seriale!
- usare slot più piccole

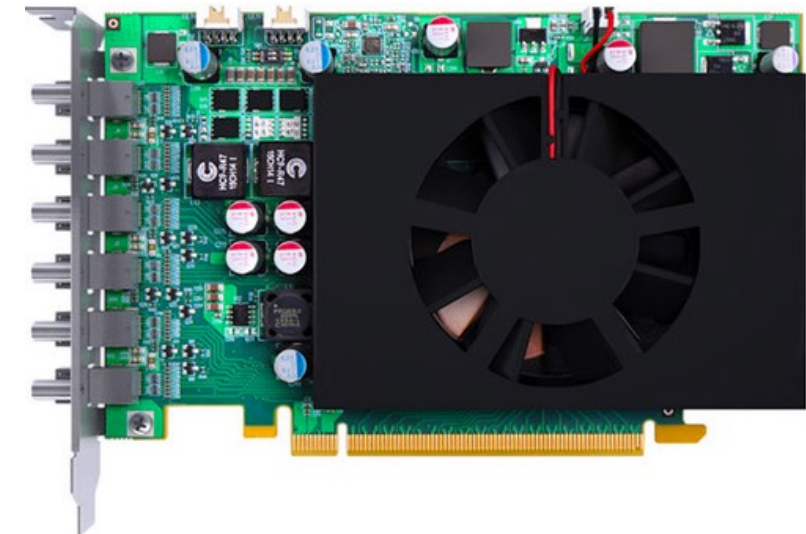
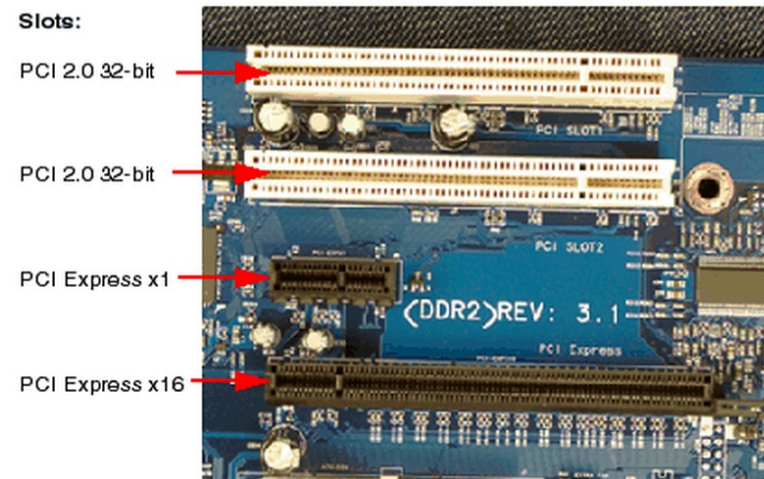
Soluzione: PCI express

- Connessione punto-a-punto



# Caratteristiche PCI express

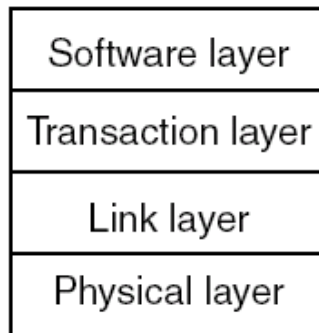
- Trasmissione seriale
- Trasferimenti come in una rete di computer:
  - dati in pacchetti (header + payload + CRC)
  - maggiore lunghezza dei cavi
  - plug-and-play
- Banda: attualmente >120GB/sec (ver. 6.0, x16)
- Controllo del flusso in base alle dimensioni dei buffer
- 4 spazi di indirizzamento (ovvero 4 tipologie di comunicazione):
  - Memoria
  - I/O
  - Configurazione
  - Messaggi



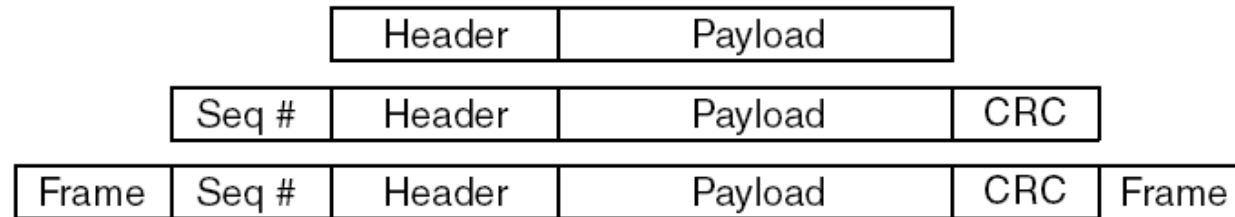


# PCI Express Protocol Stack

- Trasmissione basata su protocollo multi-layer lungo coppie di corsie (lane)
- Codifica: 8b/10b
- Un meccanismo di acknowledgment garantisce maggiore affidabilità
- Il software layer garantisce:
  - la gestione dei pacchetti
  - la compatibilità con il passato



(a)





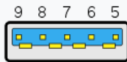

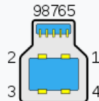

(b)

# Bus USB (Universal Serial Bus)

- Bus economico concordato da varie aziende per la gestione di dispositivi di I/O a bassa velocità (~ 1995)
- Obiettivi:
  - 1) Evitare switch, jumpers
  - 2) Installazione di tipo esterno
  - 3) Cavo di connessione unificato
  - 4) Alimentazione fornita dal cavo
  - 5) Fino a 127 dispositivi collegabili
  - 6) Supporto di dispositivi real-time
  - 7) Installazione a PC acceso
  - 8) Reboot non necessario
  - 9) Bus e dispositivi economici
- Tutti gli obiettivi sono di fatto rispettati

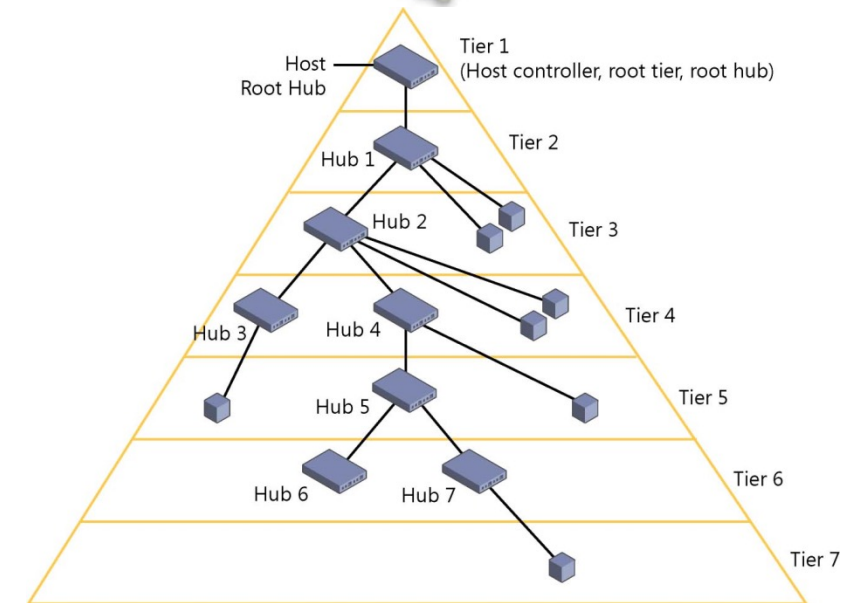
# USB (Universal Serial Bus): Specifiche di base

- Bus economico concordato da varie aziende per la gestione di dispositivi di I/O a bassa velocità (~ 1995)
- Differenti connettori (A/B/C)
- Banda complessiva
  - USB 1.X: 1.5 – 12 Mb/sec
  - USB 2.0: 480 Mb/sec
  - USB 3.X: 5 Gb/sec – 20 Gb/sec (<3GB/s)
  - USB4: 40Gb/sec
  - USB4 2.0: 120Gb/sec
  -

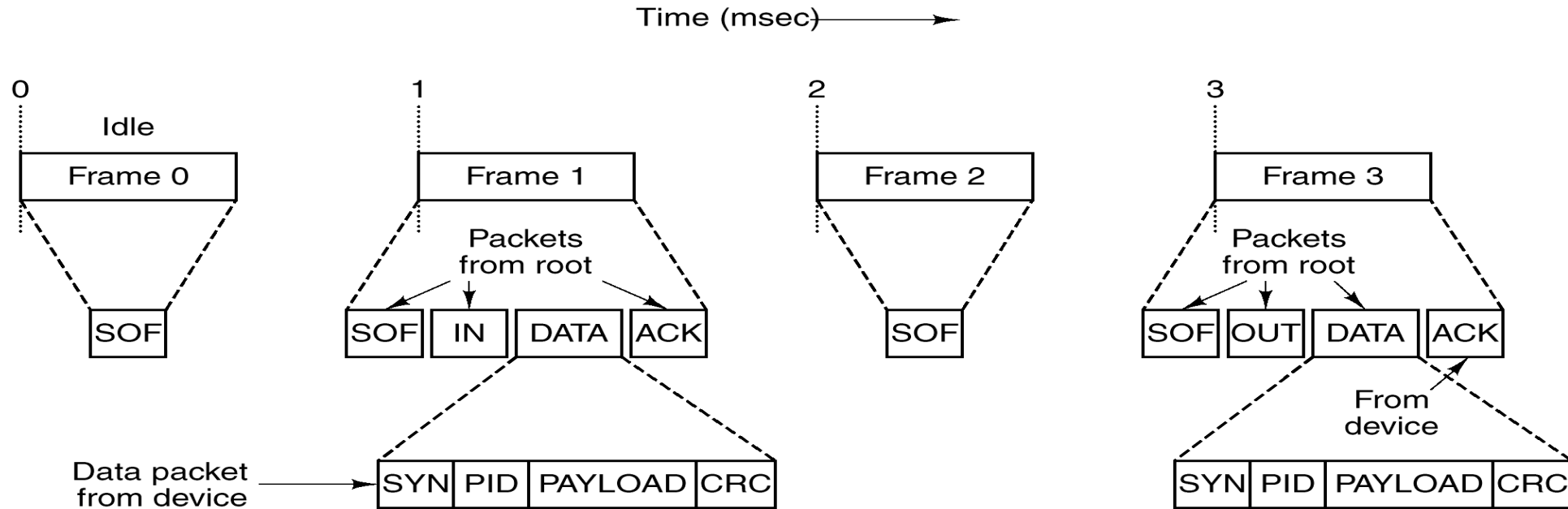
Standard	USB 1.0 1996	USB 1.1 1998	USB 2.0 2001	USB 2.0 Revised	USB 3.0 2008	USB 3.1 2013	USB 3.2 2017	USB4 2019
Maximum transfer rate	12 Mbps		480 Mbps		5 Gbps	10 Gbps	20 Gbps	40 Gbps
Type A connector	 Type-A 1.0 - 1.1		 Type-A 2.0		 Type-A SuperSpeed		Deprecated	
Type B connector	 Type-B				 Type-B SuperSpeed		Deprecated	
Type C connector	N/A				 USB-C (Enlarged)			

# USB: comunicazione su bus

- *Root hub* di connessione a bus interno
- Connessione con dispositivi e con altri hub
- Struttura complessiva ad albero
- Cavo a 4 fili: +5V, GND, 2 di segnale
- Alla connessione di un dispositivo:
  - Interrupt: intervento del SO
  - Richiesta di banda
  - Assegnazione di indirizzo
- Logicamente: connessione dedicata tra root hub e ciascun device
- Competitor:
  - FireWire IEEE 1394 serial bus



# USB: Struttura dei Frame



- Frame emessi ogni  $1.00 \pm 0.05$  msec
- Idle frame se non c'è comunicazione
- Contenuto del frame:
  - SOF: Start of Frame
  - IN / OUT: richiesta in lettura/scrittura
  - DATA: payload fino a 64 byte più controllo e codice di errore
  - ACK / NACK: acknowledge o errore
- Polling usato invece delle interruzioni