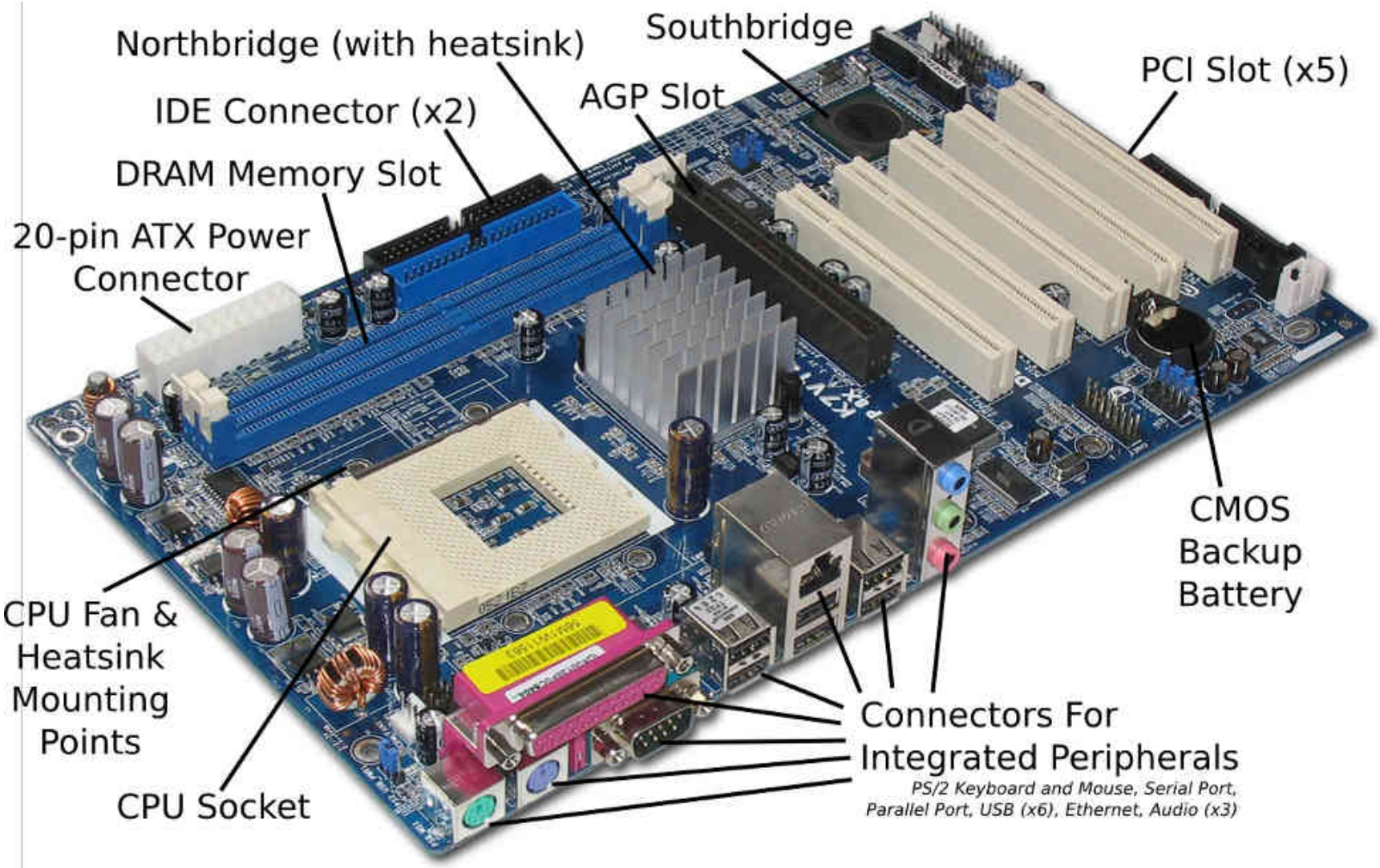
La scheda madre (motherboard)

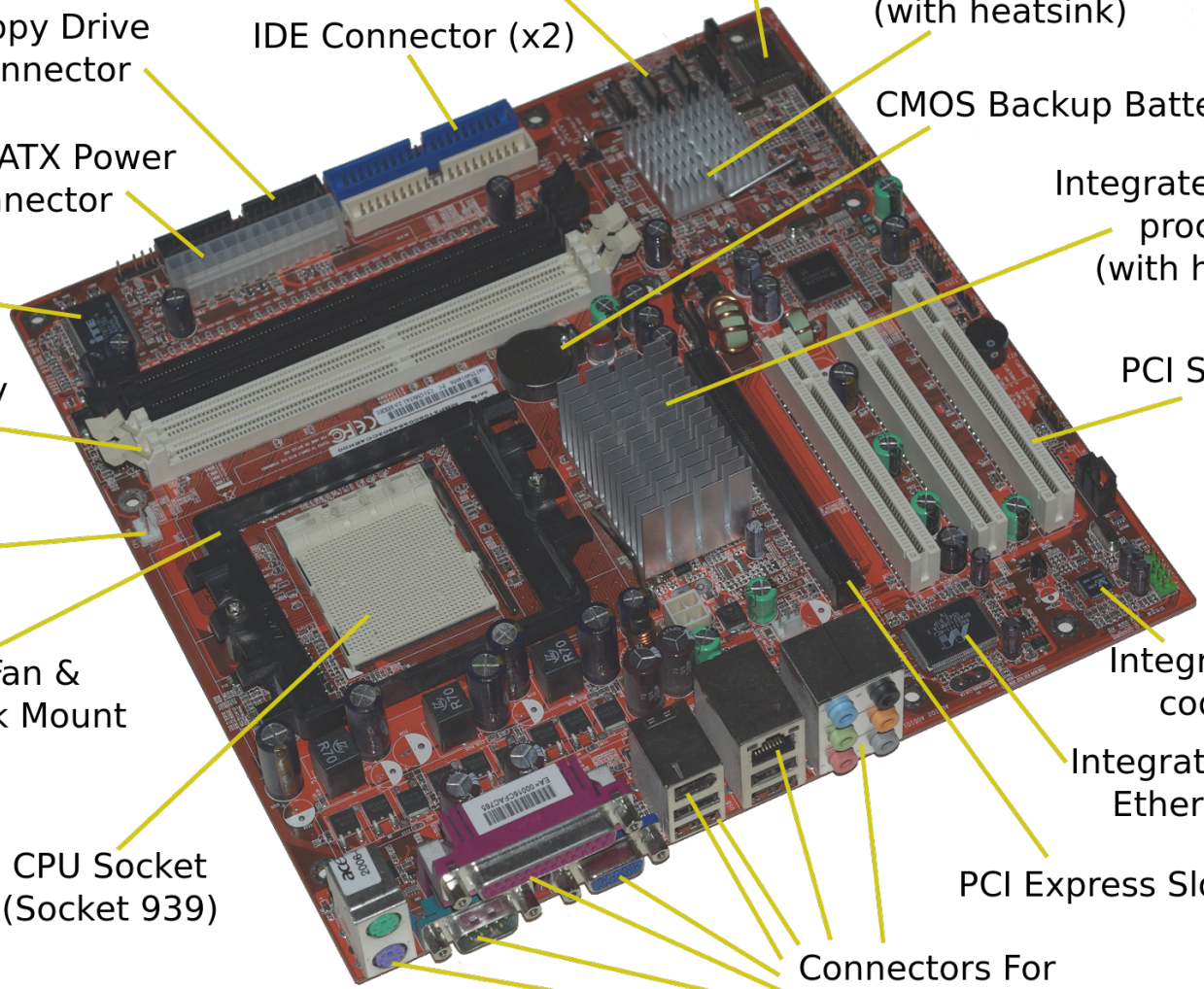
La CPU risiede sulla scheda madre (motherboard), un circuito stampato contenente anche:

- bus (canali di comunicazione)
- integrati (bridge) per il controllo dei bus
- alcuni circuiti di controllo e relative connessioni a periferiche (USB, tastiera, rete cablata)
- diversi slot per l'aggiunta di dispositivi esterni alla scheda madre
- collegamento all'alimentazione
- componenti per l'elettronica di potenza (resistori, condensatori, . . .).

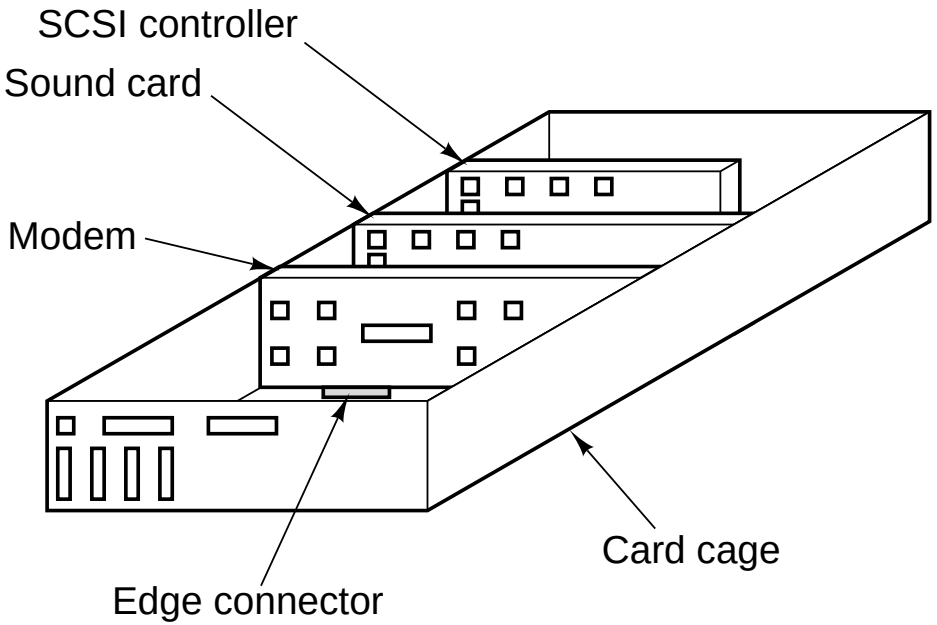
Esempio di motherboard: ASRock



Esempio di motherboard: Acer

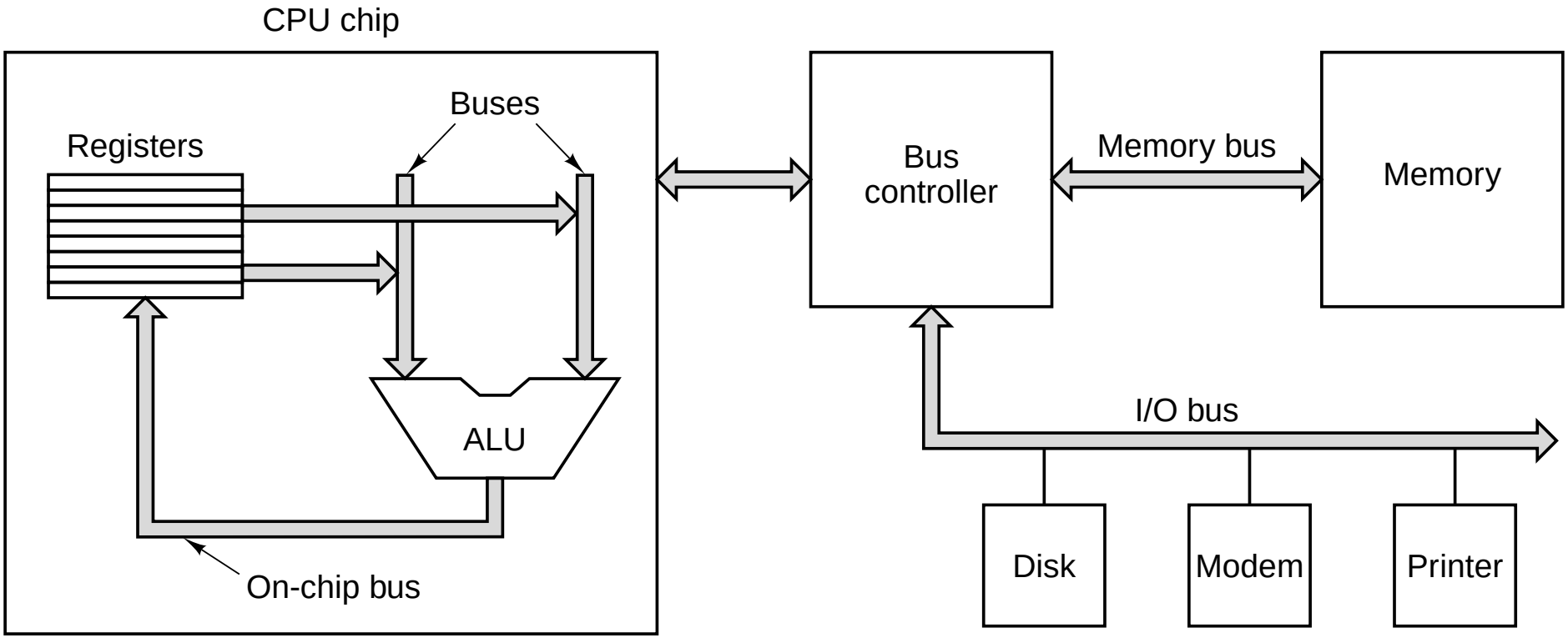


Esterni alla scheda madre

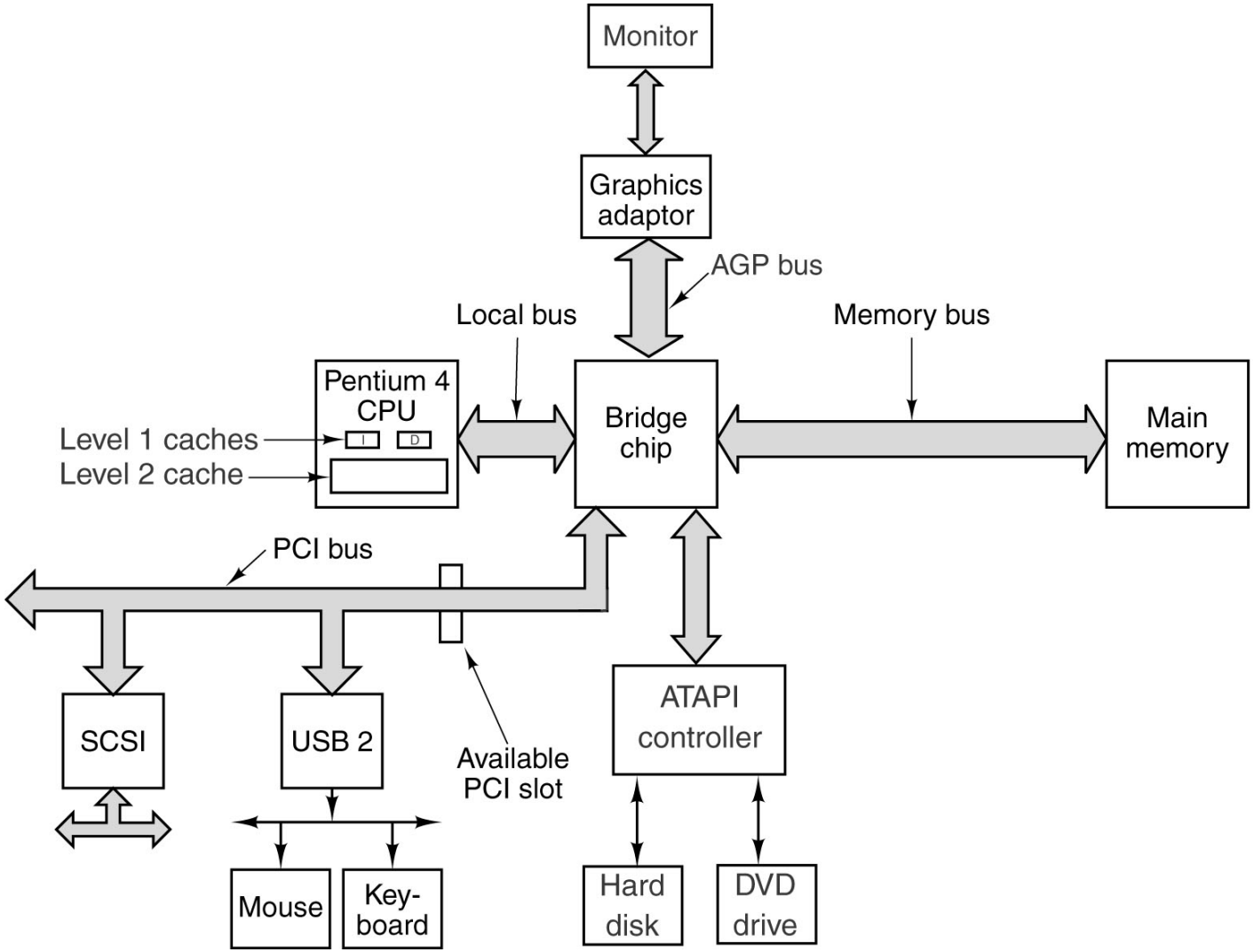


- schede di memoria principale
- schede controllori esterni
- acceleratore grafico, interfaccia audio
- memoria di massa (hard disk, stato solido)
- alimentatore.

Motherboard: struttura logica



Es.: motherboard Pentium 4 ('90)



Bus

Bus: canale fisico di comunicazione tra due o più dispositivi.

La condivisione di un unico bus tra più dispositivi è una soluzione economica e scalabile: è semplice aggiungere nuovi dispositivi.

Processore, memoria, controllori e connettori sono collegati attraverso un sistema complesso di bus.

Per gestire dispositivi con velocità diversa si usano più bus sincroni oppure bus asincroni.

Un bus dunque è caratterizzato fisicamente e dai protocolli di comunicazione.

Caratteristiche fisiche di un bus

Il bus trasporta uno o più segnali digitali lungo un insieme di connessioni fisiche che variano con la distanza da coprire

- nel circuito integrato: tracce di alluminio o rame

Caratteristiche fisiche di un bus

Il bus trasporta uno o più segnali digitali lungo un insieme di connessioni fisiche che variano con la distanza da coprire

- nel circuito integrato: tracce di alluminio o rame
- entro un circuito stampato (es.: scheda madre): tracce di rame

Caratteristiche fisiche di un bus

Il bus trasporta uno o più segnali digitali lungo un insieme di connessioni fisiche che variano con la distanza da coprire

- nel circuito integrato: tracce di alluminio o rame
- entro un circuito stampato (es.: scheda madre): tracce di rame
- da/verso la motherboard: cavi isolati.

Caratteristiche fisiche di un bus

Il bus trasporta uno o più segnali digitali lungo un insieme di connessioni fisiche che variano con la distanza da coprire

- nel circuito integrato: tracce di alluminio o rame
- entro un circuito stampato (es.: scheda madre): tracce di rame
- da/verso la motherboard: cavi isolati.

L'isolamento prevede due accorgimenti:

- coppie di cavi coassiali o intrecciate per non generare campi magnetici
- trasmissione differenziale (3 cavi) del segnale per la reiezione dei disturbi.

Condivisione del bus

Quando un bus è condiviso da più dispositivi, solo un dispositivo può imporre il valore di tensione mentre l'uscita di tutti gli altri deve restare indeterminata.

Tornano utili le porte logiche con uscita tri-state (buffer invertente / non invertente).

Gli stessi dispositivi spesso si occupano di svolgere anche il ruolo di amplificatori di segnale quando si trovano in una posizione intermedia nel bus:

$$\text{driver} \Rightarrow \text{transceiver} \Rightarrow \text{receiver}$$

In tal modo il segnale viene rigenerato durante il percorso verso il ricevente.

Frequenza e banda passante

Frequenza del bus: numero di slot temporali distinti presenti nel segnale in un secondo (1/s = Hz).

Frequenza e banda passante

Frequenza del bus: numero di slot temporali distinti presenti nel segnale in un secondo (1/s = Hz).

Banda passante = frequenza $\times$ numero di bit presenti in uno slot (bit/s).

Frequenza e banda passante

Frequenza del bus: numero di slot temporali distinti presenti nel segnale in un secondo (1/s = Hz).

Banda passante = frequenza $\times$ numero di bit presenti in uno slot (bit/s).

É un parametro ideale. La banda passante reale è minore a causa di fasi di inattività e di sincronizzazione/negoziazione tra i dispositivi.

Gli slot possono essere definiti mediante un clock condiviso (trasmissione sincrona) oppure mediante informazione temporale inglobata nello slot stesso (trasmissione asincrona).

Bus sincroni

Il clock occupa una linea del bus:

- semplicità del protocollo di comunicazione
- permette di raggiungere le massime velocità
- occorre aggiungere fisicamente una linea
- inadatto a comunicazioni con tempi di risposta non predeterminati tra dispositivi
- esiti disastrosi se il sincronismo viene perduto.

Bus sincroni sono specialmente adatti alle comunicazioni dentro il chip (bus privati: solo CPU, CPU $\Leftrightarrow$ memoria).

Bus asincroni

La sincronizzazione iniziale richiede una negoziazione (handshaking).

- condivisione più semplice di dispositivi con diversi tempi di risposta
- comunicazione più semplice e trasmissione più economica tra dispositivi fisicamente distanti
- circuiteria più complessa per la gestione dell'handshaking.

Bus asincroni sono specialmente adatti alle comunicazioni da/alla motherboard (bus pubblici: CPU $\Leftrightarrow$ I/O).

Bus seriali e paralleli

In un bus seriale l'informazione è trasmessa in modo logicamente sequenziale. In un bus parallelo tipicamente si riconoscono

- linee dati: bit da recapitare

Bus seriali e paralleli

In un bus seriale l'informazione è trasmessa in modo logicamente sequenziale. In un bus parallelo tipicamente si riconoscono

- linee dati: bit da recapitare
- linee indirizzi: bit che identificano locazioni di memoria o registri di dispositivi

Bus seriali e paralleli

In un bus seriale l'informazione è trasmessa in modo logicamente sequenziale. In un bus parallelo tipicamente si riconoscono

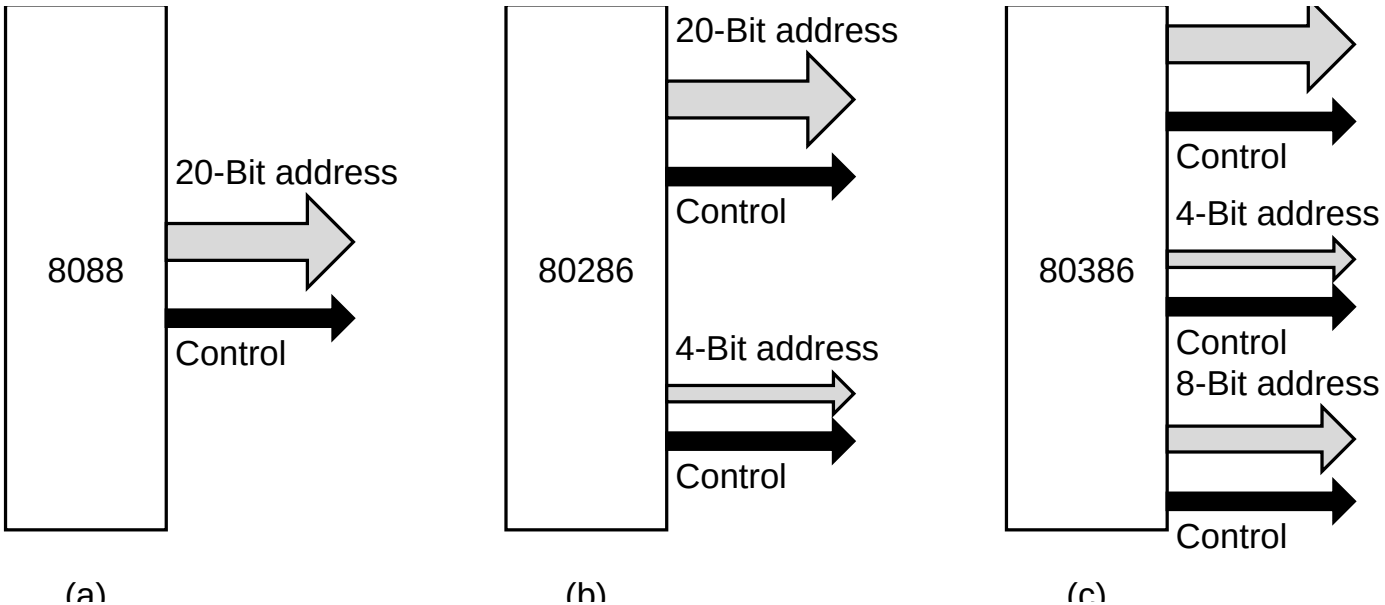
- linee dati: bit da recapitare
- linee indirizzi: bit che identificano locazioni di memoria o registri di dispositivi
- linee di controllo: bit che segnalano comandi e informazioni sul funzionamento del bus.

Spesso le linee dati e indirizzi coincidono (multiplexed bus). Es.: nelle memorie con indirizzamento a due passi N linee multiplexed sono sufficienti a indirizzare 2^{2N} locazioni di N bit.

Aggiornamento del bus parallelo

L'aumento dello spazio di memoria indirizzabile richiede di aggiungere linee. Ciò genera il problema della retrocompatibilità in un bus parallelo.

Es.: evoluzione del bus Intel x86:



Bus parallelo: bus skew

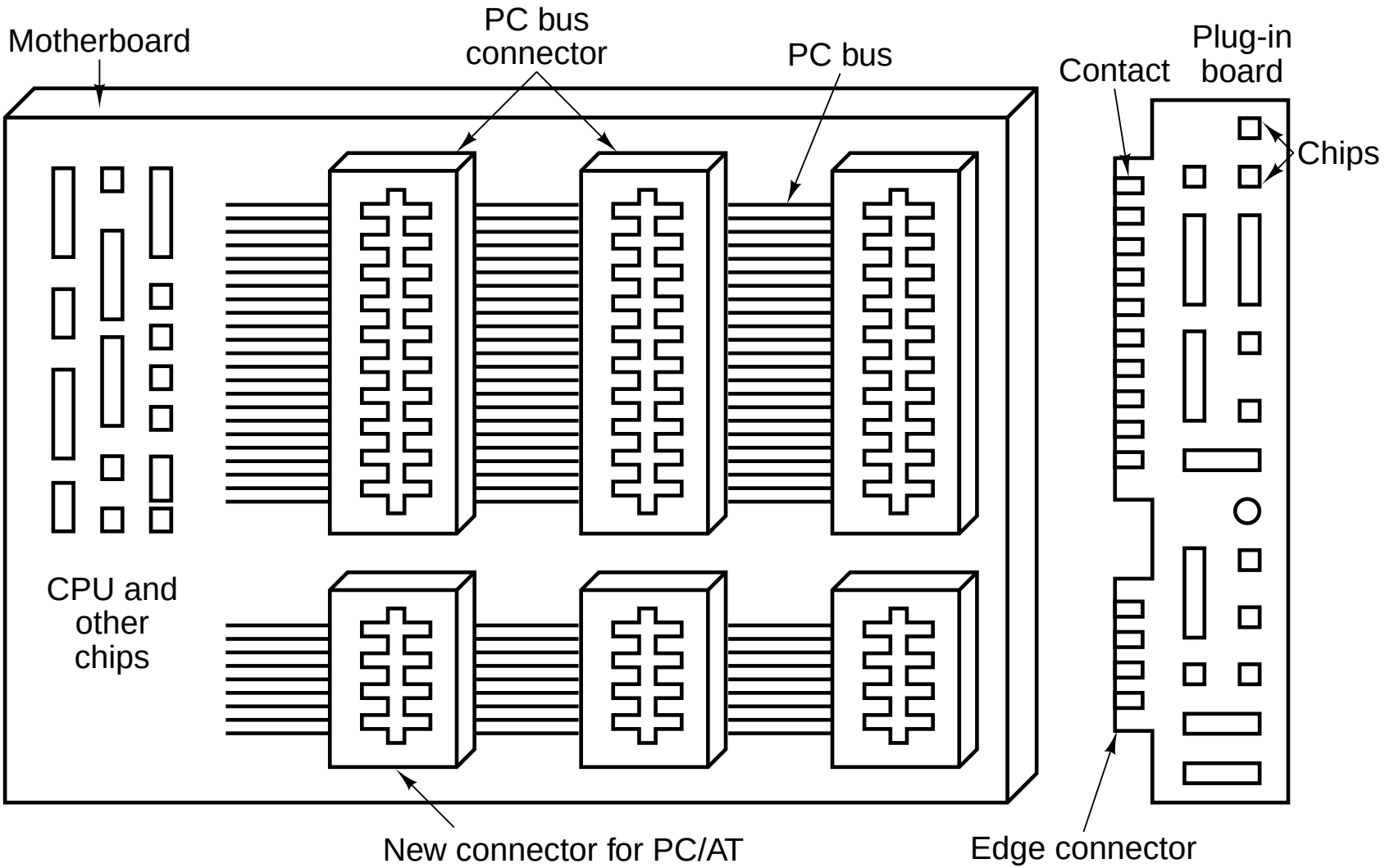
Tanto più alta è la frequenza, quanto più fragile la sincronizzazione in un bus parallelo in quanto ad alta frequenza i ritardi di propagazione diventano significativi.

Bus skew: bit su linee diverse sono recapitati in tempi diversi, in quanto segnali allineati in partenza vengono ricevuti non allineati a causa di deviazioni della velocità di propagazione del segnale da quella massima ($\approx 10\% \sim 1\%$ di quella della luce).

Il periodo = 1/frequenza del clock deve essere maggiore del bus skew.

Bus skew nelle tecnologie attuali: ~ 1 ns.

Aggiunta di controllori nel bus



Correzione degli errori di trasmissione

La trasmissione nel bus è esposta a errori:

- in un bus parallelo l'errore è raro, spesso corrompe la singola linea e comunque non genera asincronismo; tipicamente si dedica una linea al bit di parità
- in un bus seriale l'errore può coinvolgere un treno di bit distribuiti su più slot; si accodano a ogni pacchetto dei bit di controllo aggiuntivi in grado di rilevare anche errori multipli. Es.: codici a ridondanza ciclica (CRC), molto potenti.

Protocollo master-slave

Una transazione su bus condiviso prevede un trasmettitore (master: dispositivo che prende il controllo del bus) e uno o più ricevitori (slave) attraverso le seguenti fasi:

Protocollo master-slave

Una transazione su bus condiviso prevede un trasmettitore (master: dispositivo che prende il controllo del bus) e uno o più ricevitori (slave) attraverso le seguenti fasi:

- un dispositivo prende il controllo dell'intero bus attivando il buffer d'uscita eventualmente dopo una contesa, assumendo il ruolo di master
- il master trasmette dati indirizzati a uno o più dispositivi slave
- lo (gli) slave eventualmente dà (danno) un riscontro (acknowledgment)
- il master rilascia il buffer, liberando il bus per una futura comunicazione.

Dispositivi master e slave

Transazione	Master	Slave
Lettura, scrittura	CPU	Memoria
I/O	CPU	Controllore
Operazione DMA	Controllore	Memoria
Acquisizione operandi	Coprocessore	CPU

In transazioni diverse, un dispositivo può essere a volte master e a volte slave.

Concessione del bus

La concessione del bus obbedisce a due possibili politiche

- centralizzata: i dispositivi richiedono l'accesso al bus a un arbitro, gestito dal bus
- decentralizzata: l'assegnazione del master è determinata da un protocollo distribuito, gestito dai dispositivi collegati al bus.

Concessione del bus

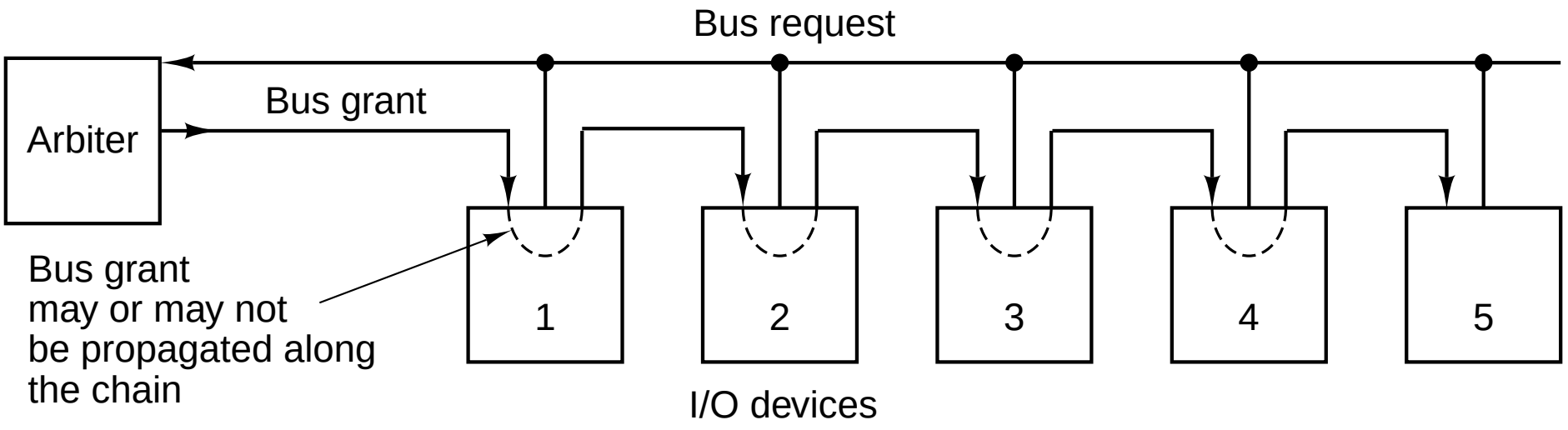
La concessione del bus obbedisce a due possibili politiche

- centralizzata: i dispositivi richiedono l'accesso al bus a un arbitro, gestito dal bus
- decentralizzata: l'assegnazione del master è determinata da un protocollo distribuito, gestito dai dispositivi collegati al bus.

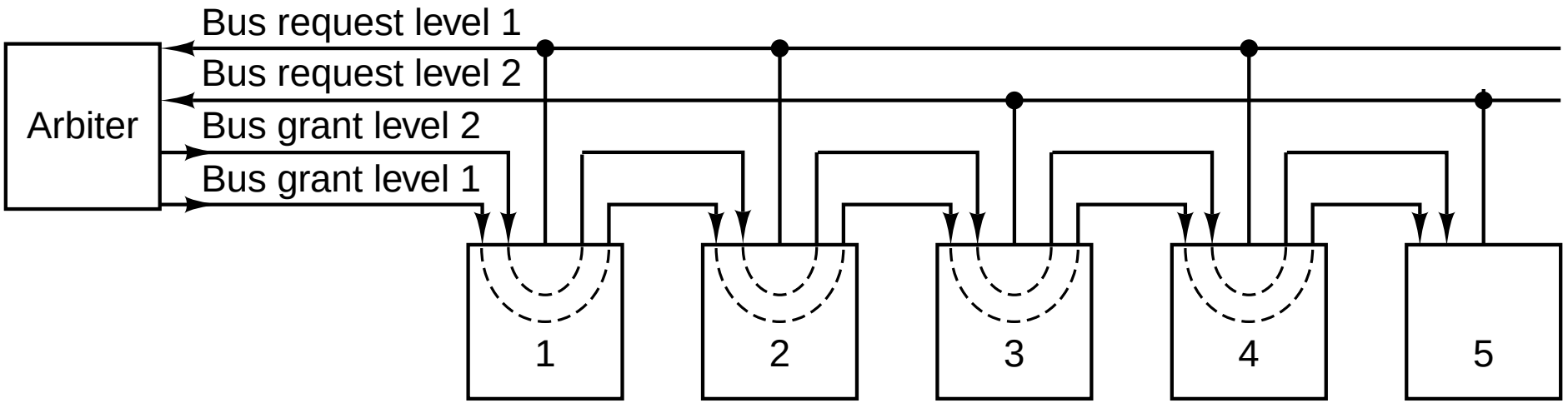
Criteri per l'arbitraggio

- priorità: l'accesso è concesso al dispositivo che ne fa richiesta posto più in alto gerarchicamente
- fairness (equità): ogni dispositivo accede prima o poi al bus.

Es.: protocollo daisy chain



(a)



(b)

Priorità in daisy chain

Daisy chain realizza un arbitraggio centralizzato:

- uno o più dispositivi asseriscono la linea di richiesta grazie alla connessione wired-OR
- se il bus è libero l'arbitro emette un token asserendo la linea di concessione (bus grant)
- un dispositivo che ha fatto richiesta non lascia transitare il grant (raccoglie il token)
- il dispositivo che vede entrambi i segnali asseriti inizia a trasmettere attraverso il bus.

Priorità in daisy chain

Daisy chain realizza un arbitraggio centralizzato:

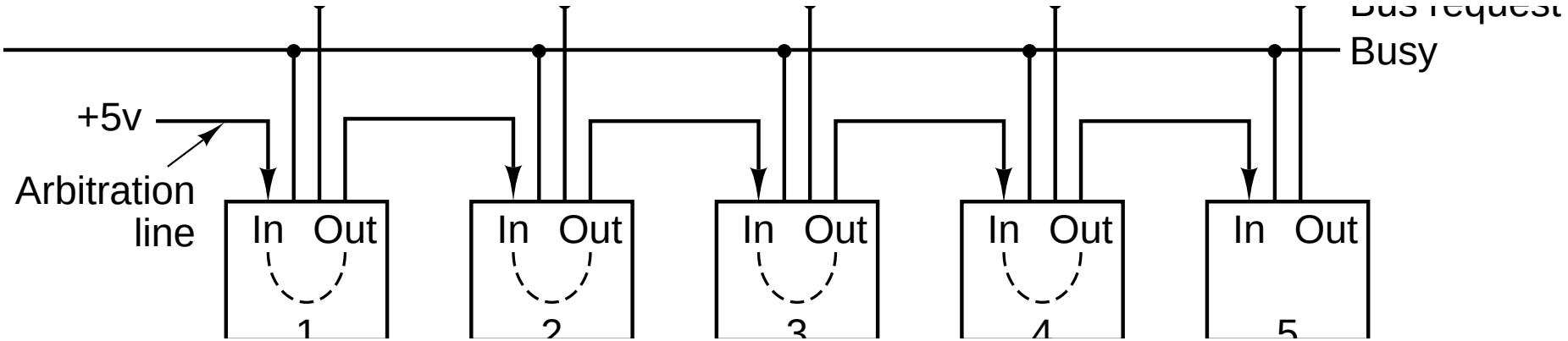
- uno o più dispositivi asseriscono la linea di richiesta grazie alla connessione wired-OR
- se il bus è libero l'arbitro emette un token asserendo la linea di concessione (bus grant)
- un dispositivo che ha fatto richiesta non lascia transitare il grant (raccoglie il token)
- il dispositivo che vede entrambi i segnali asseriti inizia a trasmettere attraverso il bus.

La priorità quindi è determinata dalla posizione fisica lungo la linea di bus grant.

Più gerarchie di priorità adoperando più bus grant.

Arbitraggio decentralizzato

Gerarchia di priorità identica a daisy chain



Tre linee per i comandi: busy, request, grant.

- il bus è disponibile se il segnale busy non è asserito
- i dispositivi che contendono il bus disattivano il grant verso il dispositivo alla loro destra
- solo un dispositivo richiedente che vede il grant asserito può impegnare il bus.

Esempi di protocollo decentralizzato

Il bus SCSI rendeva disponibile una linea di richiesta per dispositivo.

Ethernet fa a meno di linee di comando dedicate:

- ogni dispositivo che rileva la disponibilità del bus in ogni momento può occuparlo e trasmettere
- eventuali conflitti sono rilevati successivamente all'occupazione simultanea del bus
- in caso di rilevamento di un conflitto, i dispositivi che stanno trasmettendo annullano la comunicazione e, dopo un ritardo casuale, effettuano un nuovo rilevamento di disponibilità
- il successo del protocollo è dato dalla possibilità di collegare dispositivi molto distanti tra loro.

La prossimità con la CPU

La prossimità alla CPU dei dispositivi determina la banda passante, il costo e la flessibilità del bus:

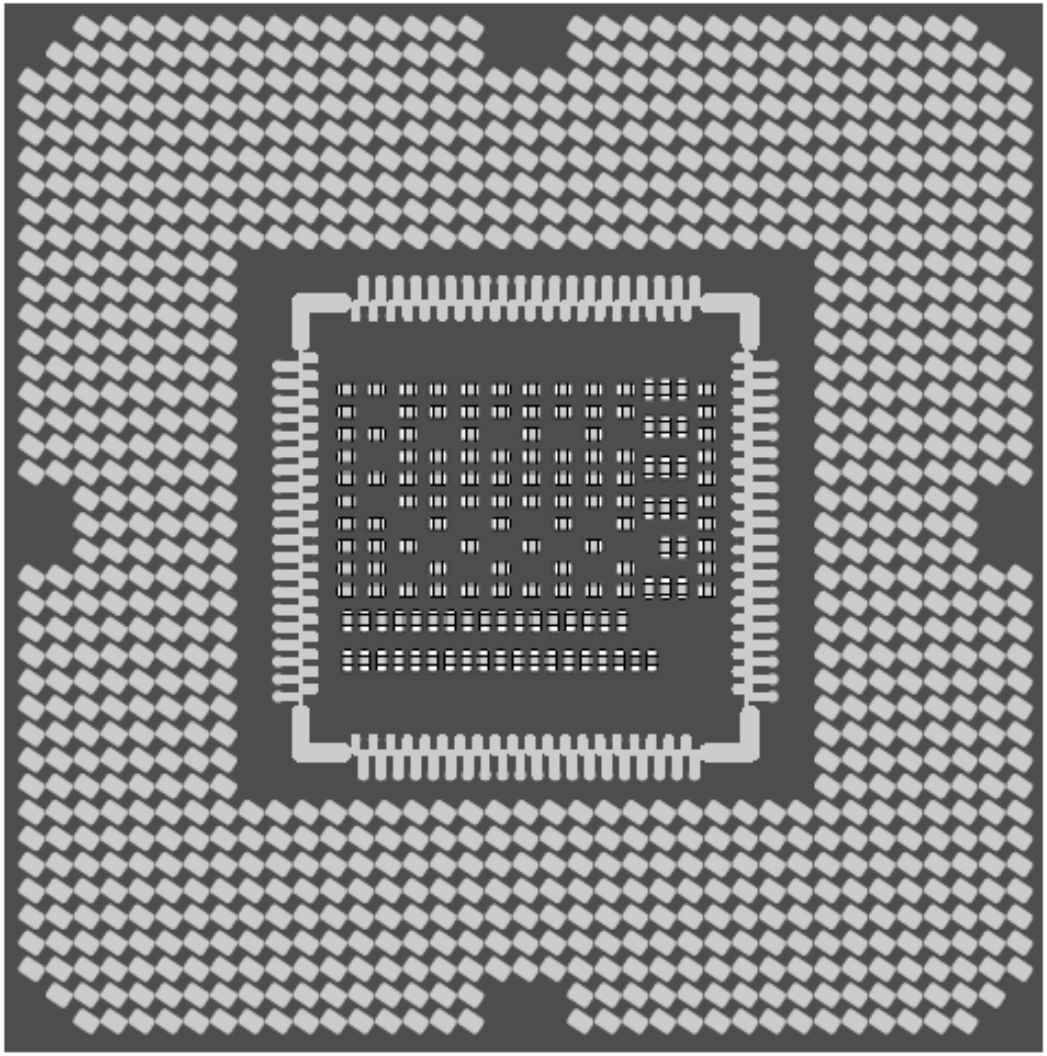
- bus locale (front-side bus), proprietario, banda elevata
- bus della memoria cache (back-side bus), proprietario
- bus della memoria principale, con specifiche stringenti
- bus della scheda video (Accelerated Graphics Port, AGP)
- bus per specifici controllori (ISA, ATA, SCSI)
- bus di sistema per i principali dispositivi (PCIe)
- bus per dispositivi esterni: USB, Thunderbolt.

L'accessibilità della CPU

Attualmente una CPU possiede centinaia di connessioni

- bus locale, bus alle memorie, bus ai dispositivi: dati, indirizzi, correzione degli errori, comandi
- alimentazione: centinaia di punti di tensione e di massa per ridurre il flusso di corrente (riduzione dei disturbi elettromagnetici)
- sensoristica (temperature, potenza assorbita)
- configurazione
- diagnostica
- clock (unità esterna).

Es.: piedinatura Core i7

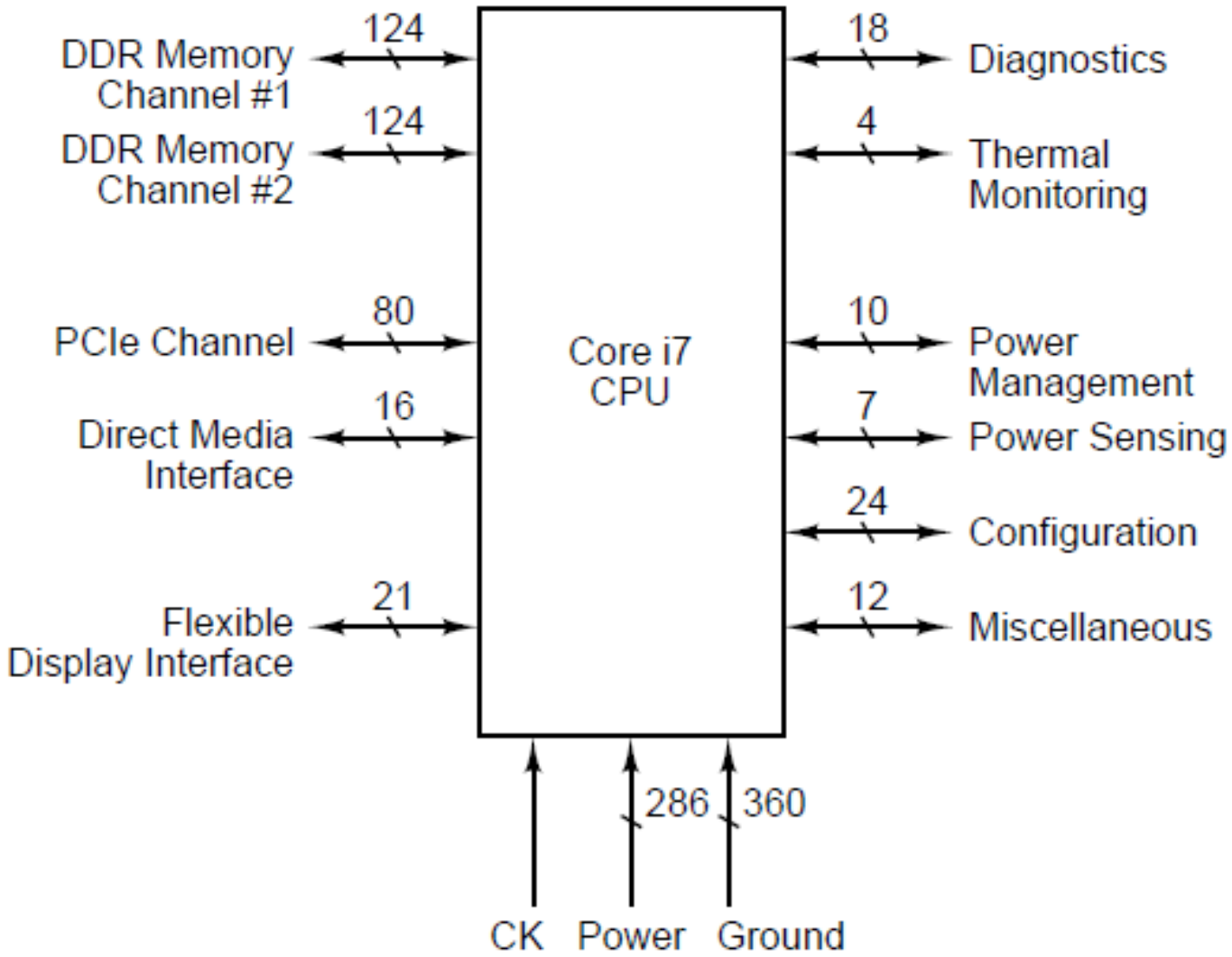


Es.: connessioni dati Intel Core i7

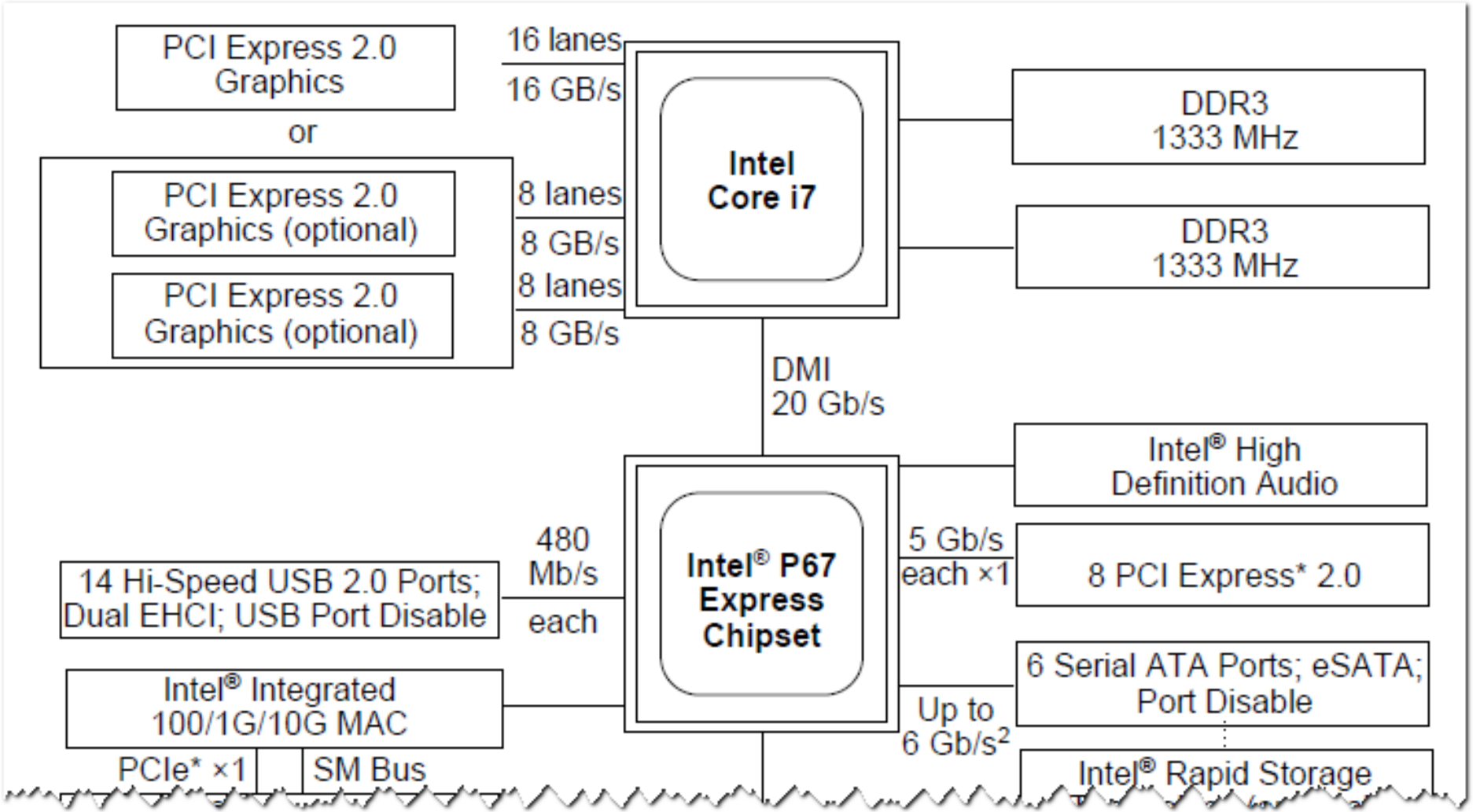
Core i7 comunica con l'esterno attraverso

- 2 bus proprietari a 666 MHz e 64 linee dati per connessione a due memorie DDR3 SDRAM (due transazioni per ciclo di clock): 20 GB/s
- 1 bus PCIe per la connessione alla scheda grafica: 16 GB/s; in alternativa la grafica Intel è connessa tramite un bus proprietario Flexible Display Interface
- 1 bus Direct Media Interface, bus proprietario simile a PCIe per connessione a un chipset esterno (evoluzione dei chip DMA) a cui si collegano i restanti dispositivi: 20 GB/s.

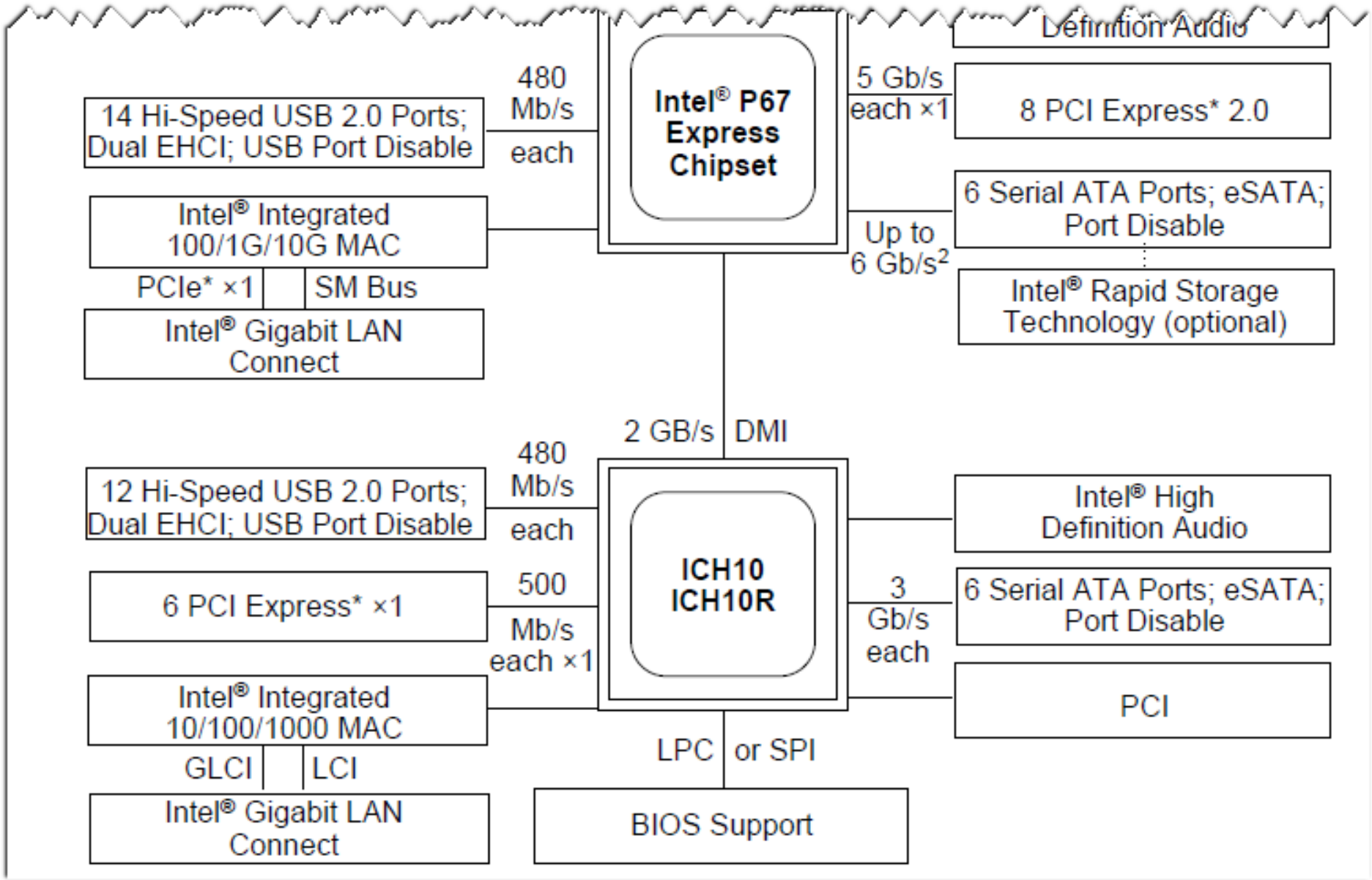
Core i7: schema connessioni



Core i7: struttura logica bus



Core i7: struttura logica bus

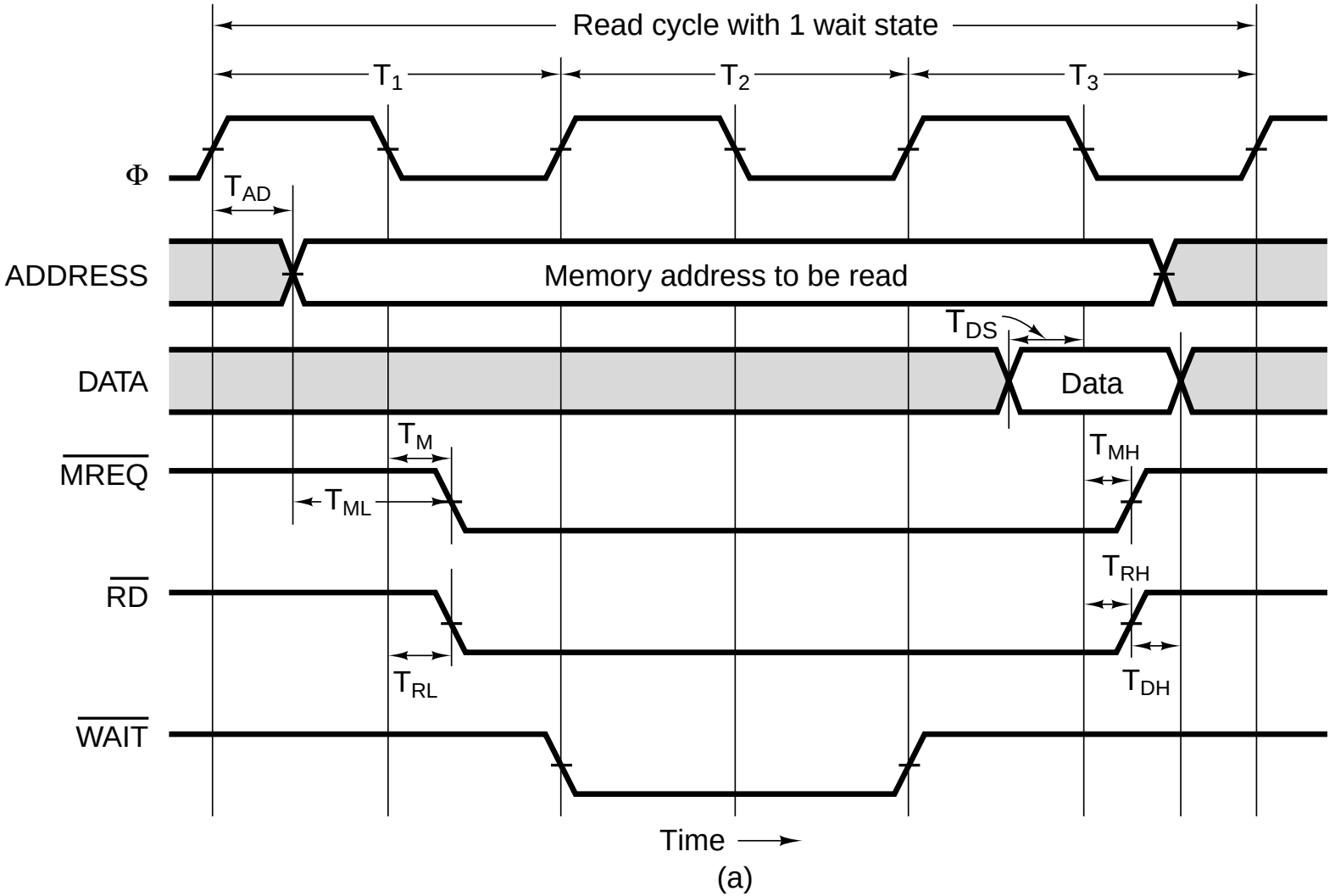


Accesso alla memoria: ciclo di bus

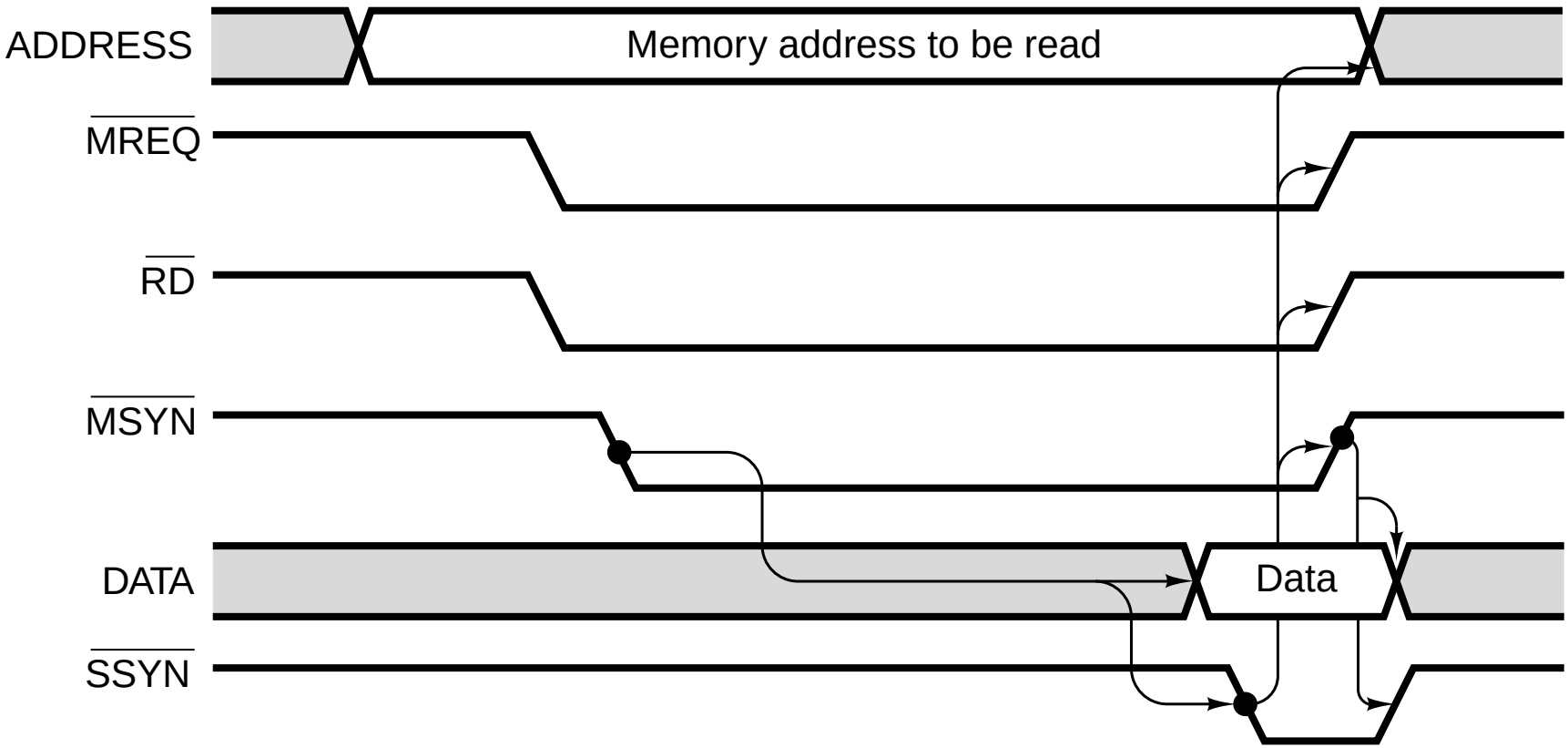
Un accesso alla memoria principale può essere realizzato in diversi modi, i quali richiedono in generale uno specifico ciclo di bus:

- lettura o scrittura di una locazione di memoria
- lettura o scrittura di un blocco di locazioni consecutive
- lettura e scrittura immediata di una locazione (accesso a dati condivisi: read-modify-write)
- invio richiesta e vettore di interrupt dalla memoria
- configurazione e test della memoria (es.: accensione).

Es.: ciclo di bus per la lettura sincrona di una locazione di memoria



Es.: ciclo di bus per la lettura asincrona di una locazione di memoria



Es.: accesso alla memoria DDR3

Parallelizzazione accessi alla memoria (pipelining).

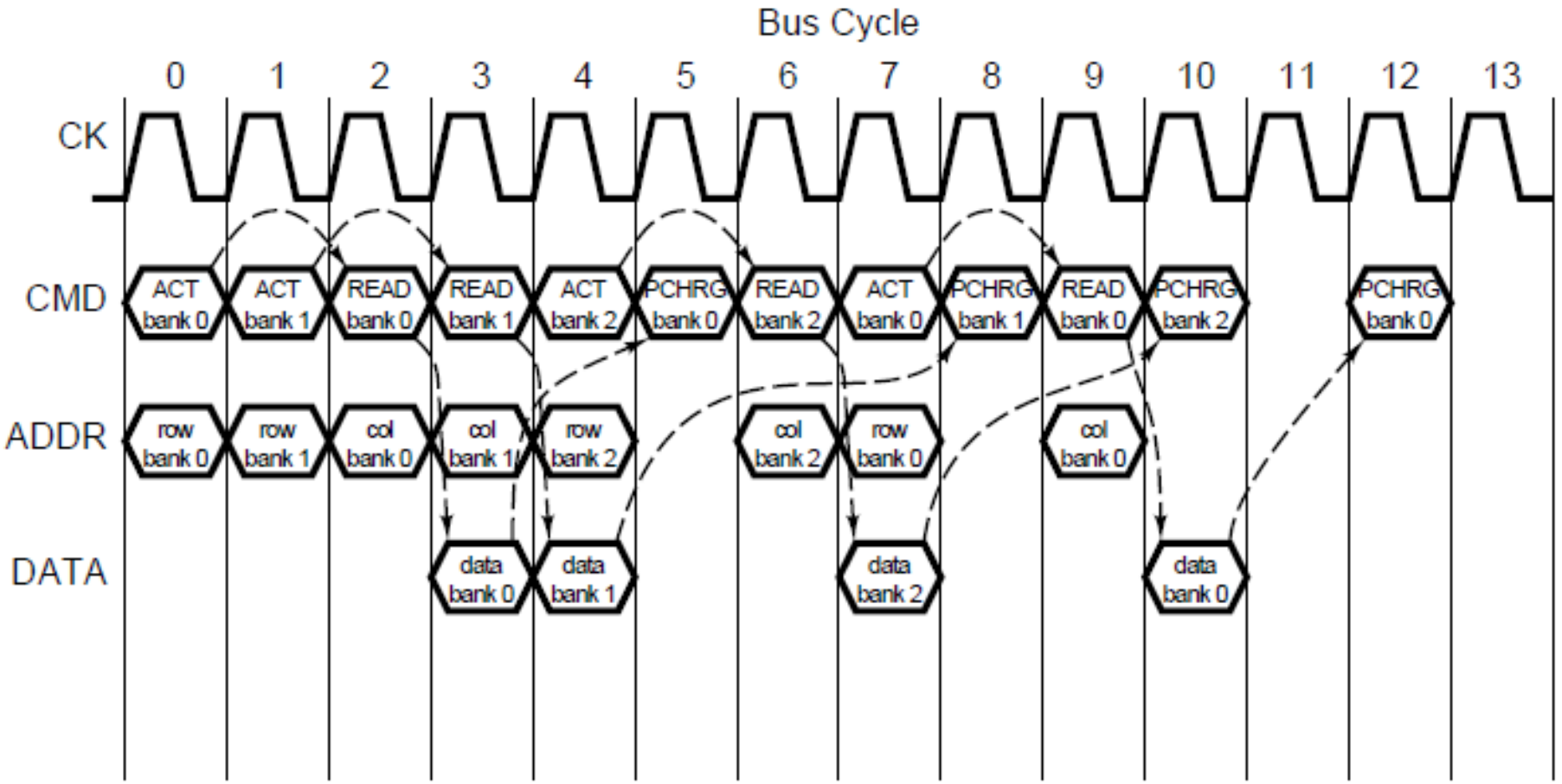
Accesso ai dati in 3 fasi

- ACTIVATE: si prepara la lettura, una riga della matrice di celle viene preparata all'accesso
- READ/WRITE: si eseguono accessi multipli a singole o sequenze di parole (burst mode) nella riga attivata
- PRECHARGE: chiude la riga corrente e prepara la memoria a una nuova ACTIVATE.

La memoria DDR è divisa in banchi (tipicamente 8); fino a 4 banchi possono essere attivati simultaneamente.

Es.: schema accesso alla DDR3

i segnali di comando, indirizzi, dati, possono operare su transazioni diverse:



Bus di sistema: ISA (anni ’90)

ISA (Industry Standard Architecture) era un bus di sistema dei primi PC, evoluzione del PC bus (IBM PC) e del PC/AT bus (Intel 80286).

Bus di sistema: ISA (anni ’90)

ISA (Industry Standard Architecture) era un bus di sistema dei primi PC, evoluzione del PC bus (IBM PC) e del PC/AT bus (Intel 80286).

Contiene 64 + 36 linee:
- 20 + 4 linee indirizzi
- 8 + 8 linee dati

Sincrono con clock a 8.33 MHz.

I bus IDE, ATA sono una sua diretta derivazione.

A volte presente, per legacy, anche nei PC attuali oppure sostituito da LPC Bus che lo simula a livello software.

Bus di sistema: PCI (fine anni ’90)

PCI (Peripheral Component Interconnect)

Intel 1992, in sostituzione bus ISA (EISA, VESA).
Brevetti resi pubblici da Intel.
Evoluzione gestita dal consorzio PCI-SIG (PCI Special Interest Group).

Diverse versioni: PCI, PCI 2.0, PCI 2.1, PCI 2.2, PCI-X, PCI-X DDR

- 32-64 linee dati/indirizzi (multiplexed);
- clock a 33, 66, 133, 266 MHz;
- alimentazione a 5 V e poi 3,3 V (entrambe possibili).

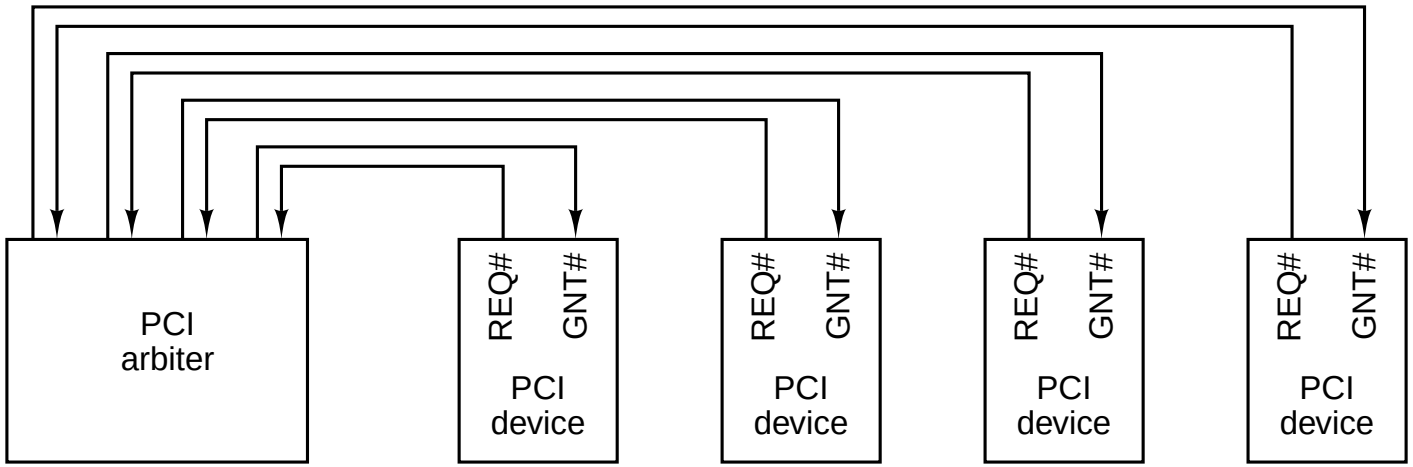
Versatilità di PCI

PCI permette il collegamento con diversi dispositivi:

Versatilità di PCI

PCI permette il collegamento con diversi dispositivi:

- numero di piedini variabile
- più tensioni di alimentazione su linee fisicamente diverse
- più frequenze di clock (selezionate asserendo una linea)
- arbitraggio centralizzato nel bridge.



PCI: arbitraggio

PCI realizza un arbitraggio semplice e veloce, più costoso richiedendo due linee distinte per ogni dispositivo.

REQ# = segnale di richiesta
GNT# = segnale di assegnazione.

I segnali operano in logica negata (vantaggi nei casi di guasto).

Il master può occupare il bus per più cicli se gli viene concesso il grant.

Terminologia: Initiator (master), Target (slave).

PCI: segnali

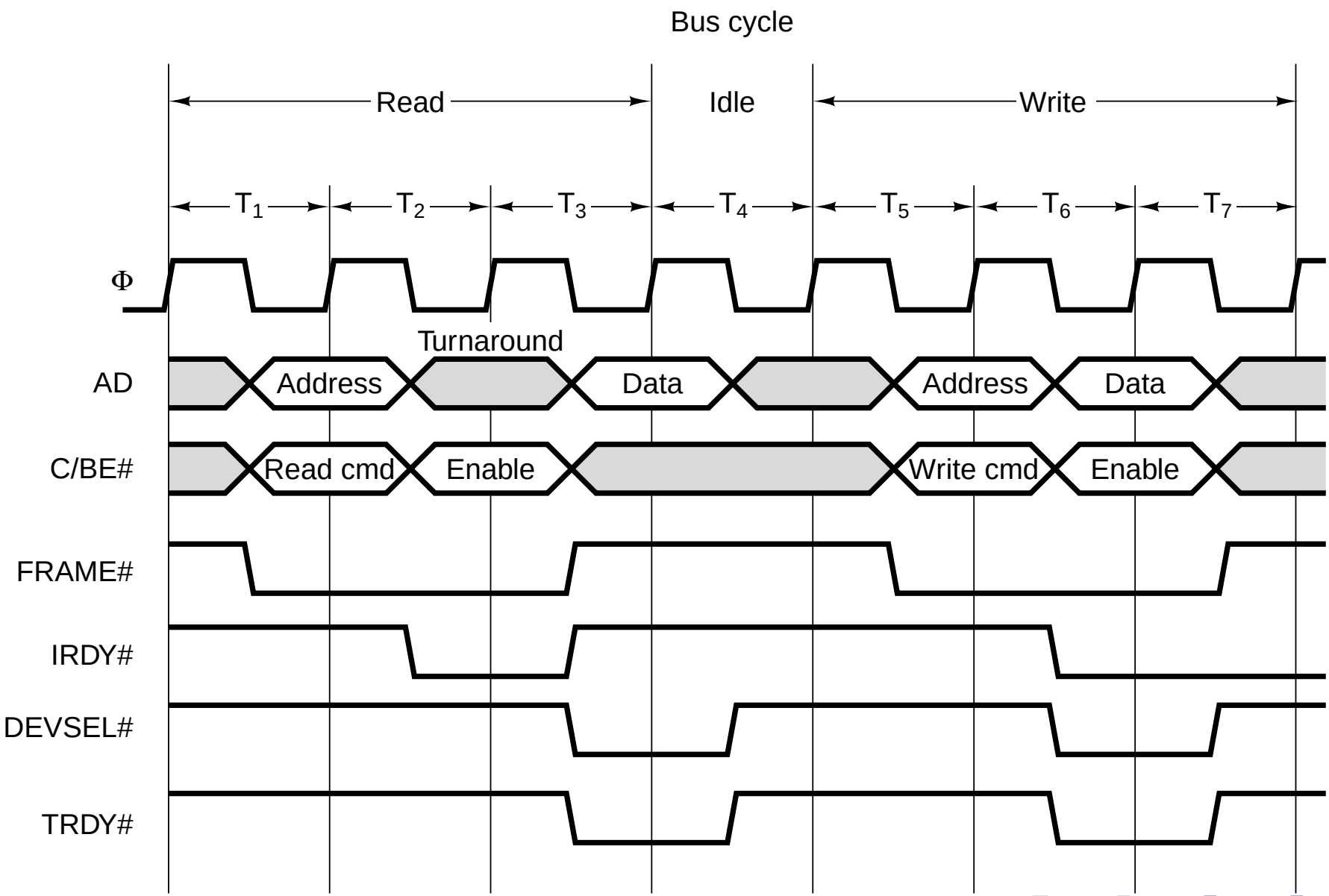
Non sono da ricordare!

CLK: clock (inizio periodo: fronte di discesa)
AD: (32 linee) indirizzi e poi dati - M o S
PAR: bit di parità per AD - M o S
C/BE: (4 linee) tipo di comando e poi byte validi nel word a 32 bit - M
FRAME: via libera, comandi e indirizzo validi - M
IRDY: M disponibile a leggere dati, o correttezza dei dati uscenti dal master - M
IDSEL: lettura configurazione dispositivo (P&P) - M
DEVSEL: risposta di disponibilità - S
TRDY: duale a IRDY - S
STOP: richiesta (inattesa) fine transazione - S.

PCI: segnali ausiliari e per chip 64 bit

PERR: errore sul controllo di parità - M o S
SERR: errore di sistema o di indirizzamento
REQ - GRN: richiesta - assegnazione bus
RST: reset manuale del sistema, errore irrimediabile
REQ64: richiesta transazione a 64 bit - M
ACK64: riscontro a REQ64 - S
AD: (32 linee) estensione indirizzi e dati - M o S
PAR64: extra bit di parità
C/BE: (4 linee) estensione del comando C/BE
LOCK: lock per transazioni multiple
SBO - SDONE: snooping, sistemi multiprocessore
INTx: (4 linee) - richiesta di interrupt
M66EN: selezione frequenza del clock.

PCI: transazione lettura/scrittura



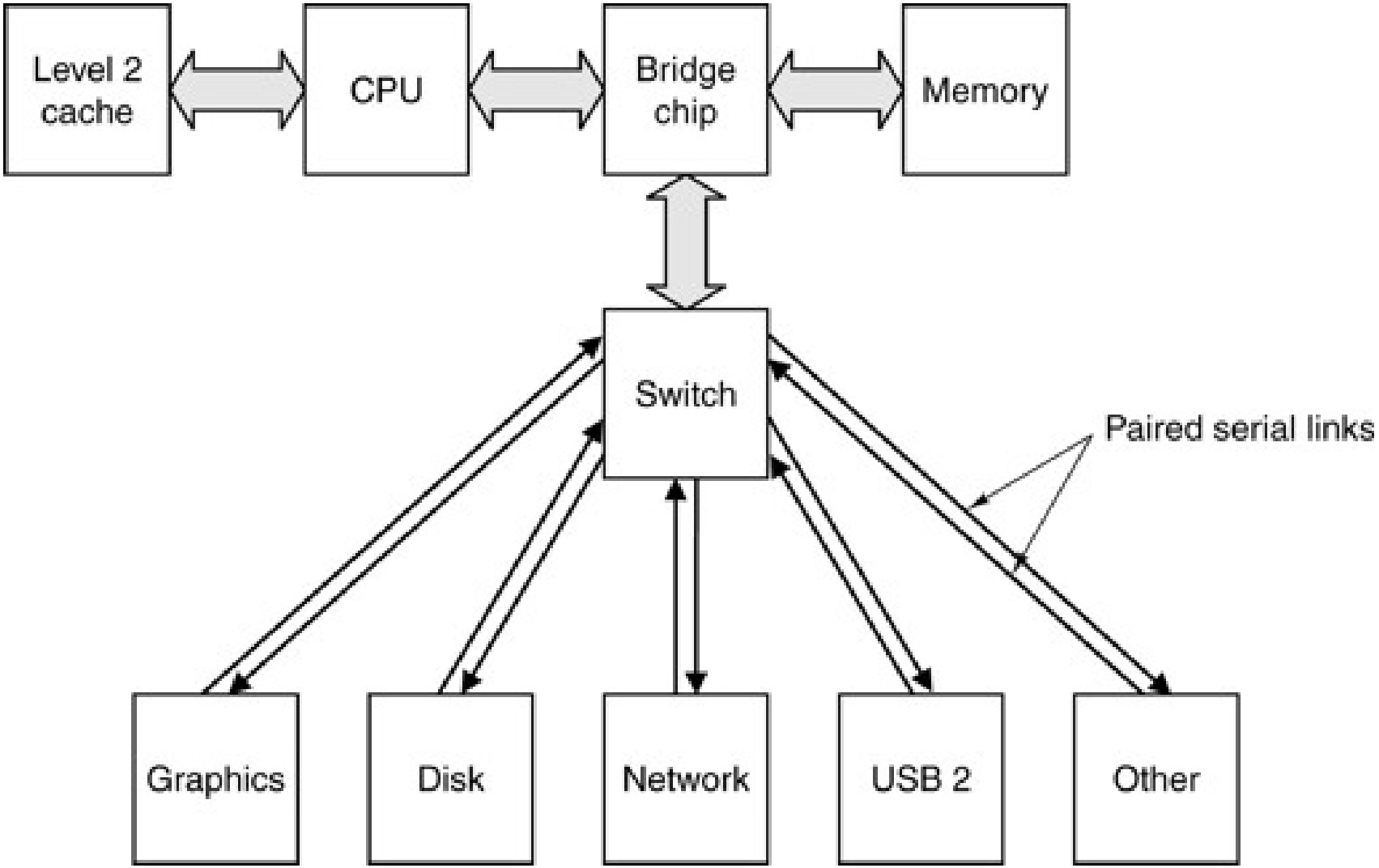
PCI Express (PCIe)

Totalmente differente nella tecnologia, PCIe conserva l'interfaccia del bus PCI, uniformando così le transazioni col sistema operativo e garantendo la retrocompatibilità del nuovo hardware.

Motivazioni

- la tecnologia del bus PCI aveva raggiunto i suoi limiti in termini di banda passante
- avere un bus fisicamente più compatto, con minori vincoli di lunghezza
- avere un solo bus nella motherboard: eliminare AGP, ATA, bus della memoria eccetera; connettere tutta la memoria principale sotto uno stesso protocollo.

PCIe: struttura logica



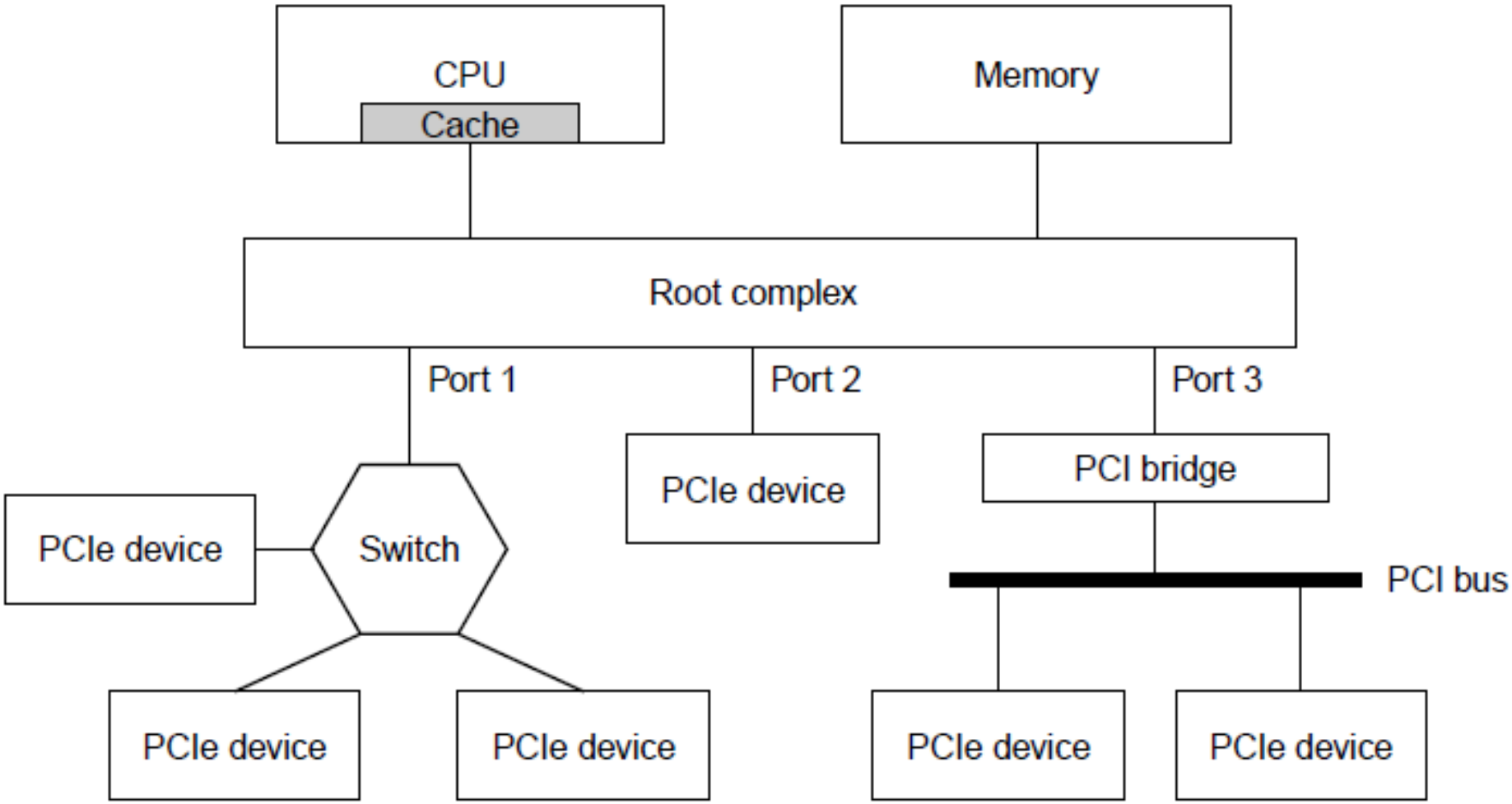
PCIe: tecnologia

Diversamente da PCI, è un bus seriale

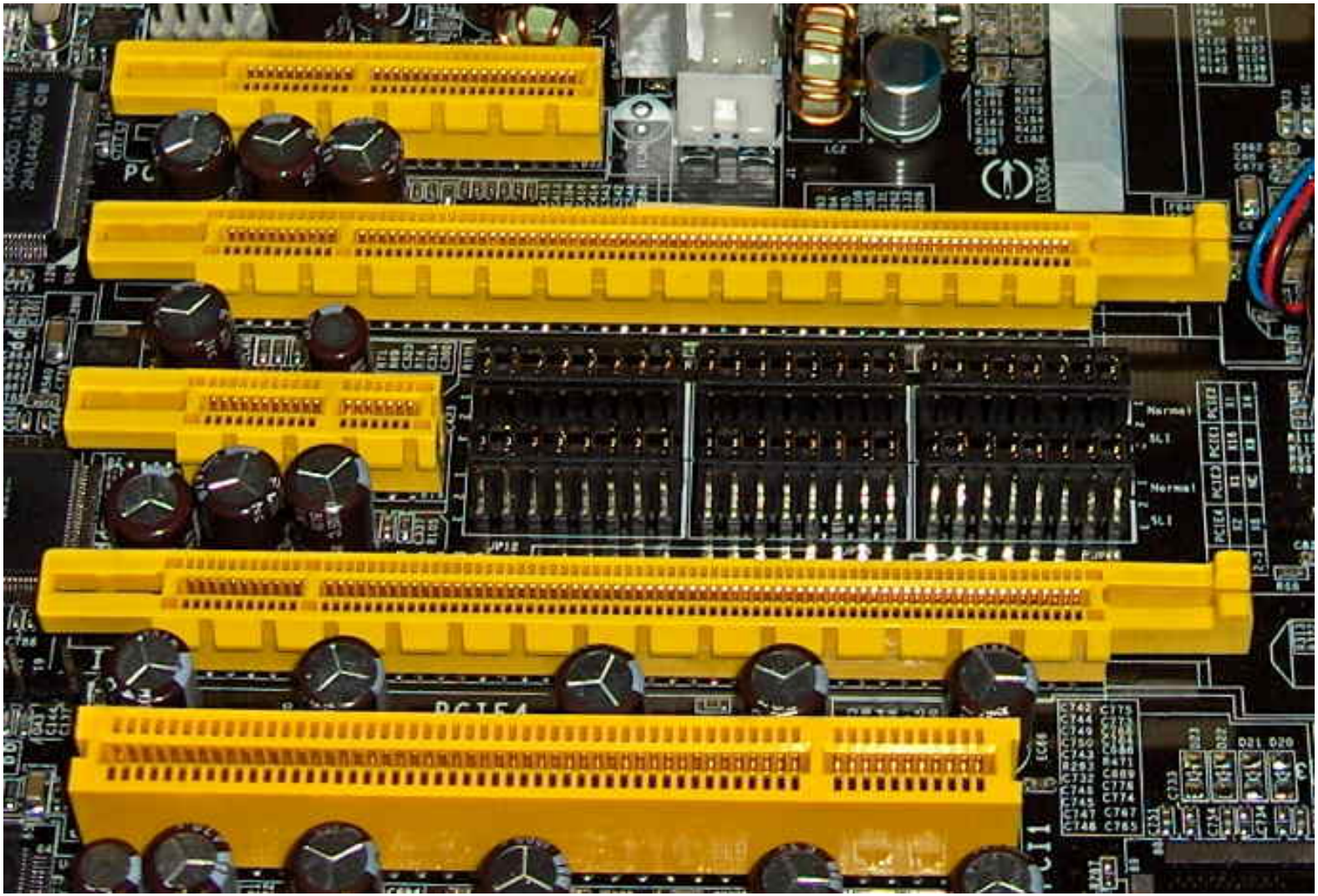
- ogni dispositivo ha una sua linea di input e una di output; la connessione è punto a punto: collegamento indipendente per ogni dispositivo, configurazione a stella (centro stella: switch)
- dispositivi con più connessioni dati (2,4,8,12,16) impegnano più linee PCIe
- trasmissione dati a pacchetto (sequenze di bit)
- banda passante: 2.5, 5, 8, 16 Gb/s.

PCIe: il root complex

Possibilità di connettere dispositivi al root complex direttamente oppure attraverso switch e/o bridge



PCIe: slot di connessione

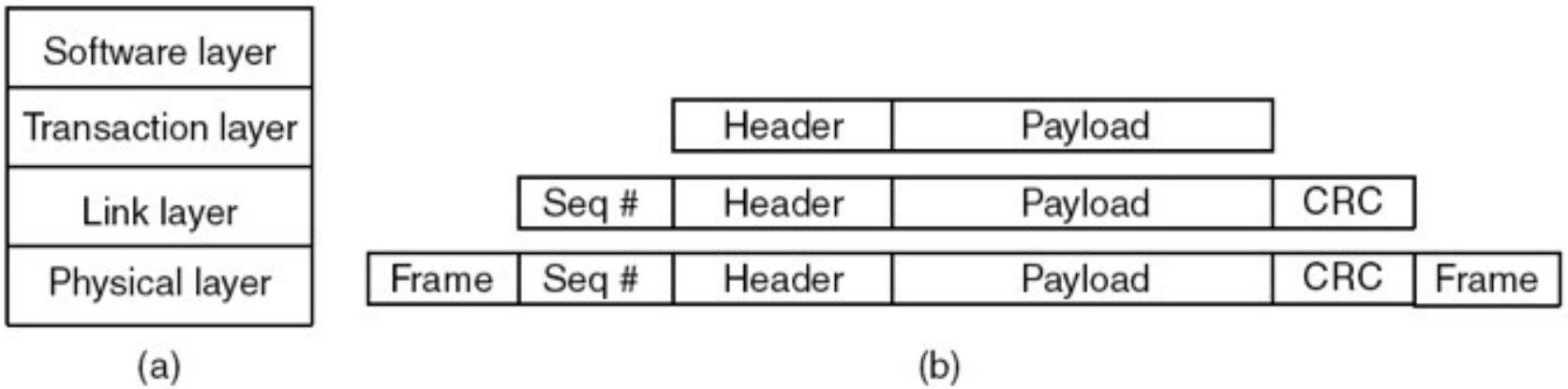


PCIe: livelli di comunicazione

La trasmissione avviene per livelli di comunicazione (eredità di Ethernet).

Ogni livello garantisce funzionalità, aggiungendo a un pacchetto le relative informazioni necessarie.

Una volta trasmesso fisicamente, il pacchetto si compone di una serie di campi che il ricevitore progressivamente decodifica fino al recapito del messaggio al livello di programma.



PCIe: livelli fisico e trasmissione

Livello fisico

- assicura la compatibilità tra slot e schede fisicamente diverse
- il segnale differenziale è trasmesso lungo due linee intrecciate
- presenza di un bit di clock, per agevolare la sincronizzazione
- codice 8b/10b (128b/130b da PCIe 3.0).

PCIe: livelli fisico e trasmissione

Livello fisico

- assicura la compatibilità tra slot e schede fisicamente diverse
- il segnale differenziale è trasmesso lungo due linee intrecciate
- presenza di un bit di clock, per agevolare la sincronizzazione
- codice 8b/10b (128b/130b da PCIe 3.0).

Livello trasmissione

- correzione di errori sul pacchetto mediante un codice a ridondanza ciclica
- invio pacchetti di riscontro (acknowledgement)
- generazione dell'interrupt.

PCIe: livelli transazione e programma

Livello transazione

- sistema di crediti per gestire le priorità durante la comunicazione
- possibilità di definire circuiti virtuali: la comunicazione tra due dispositivi è suddivisa su più canali (fino a otto) al fine di permettere più trasmissioni indipendenti

Livello programma

- interoperabilità con il software che adoperava il bus PCI. Caso di successo di indipendenza dall'hardware.

N.B.: molti bus tradizionali non emulati da PCIe continuano a essere presenti nelle motherboard.

Universal Serial Bus (metà ’90)

USB è un bus per il collegamento di periferiche esterne sviluppato dal 1995 dal consorzio USB-IF, a cui l’industria ha via via aderito.

L’obiettivo è massimizzare la semplicità dell’utilizzo:

- un unico bus per tutte le periferiche
- accesso a slot interni non più necessario
- un unico collegamento per tutte le periferiche
- scalabilità del bus
- connettività “a caldo” (“hot swapping”, “plug’n play”)
- supporto dispositivi a tempo reale (es.: audio, webcam)
- disponibilità di alimentazione elettrica.

USB: evoluzione

Larghezza di banda crescente con gli aggiornamenti

- USB 1.0: 1.5 Mb/s (1995)
- USB 1.1: 12 Mb/s (1998)
- USB 2.0: 480 Mb/s (2001)
- USB 3.0: 4.8 Gb/s (2010)
- USB 3.1: 10 Gb/s (2013), connettore USB-C parzialmente retrocompatibile (2014)
- USB 4.0: 40 Gb/s (2019), compatibile con il bus Thunderbolt, di cui eredita l'esperienza.

Inoltre, USB sta abbandonando la linea cablata per migrare sul collegamento wireless.

USB: connessioni

USB realizza un bus seriale

- fino a USB 2.0, half-duplex differenziale a 4 linee:
  2 linee per lettura o scrittura, $v/2$ e $-v/2$
  1 linea di alimentazione (5 V)
  1 linea di massa (0 V)
- USB 3.0, full-duplex differenziale a 5 linee: 0 V, 5 V, v_1, v_2, $v_1 + v_2$.
- USB Type-C (24 linee, 12×2): 2 canali a bassa velocità, 4 canali ad alta velocità, configurazione e alimentazione; più tensioni di alimenzazione, fino a 20 V $\times$ 5 A = 100 W (! difficile da sostenere per i laptop).

USB: struttura della rete

USB organizza i dispositivi in un albero che si estende (a livello di protocollo) fino a 127 nodi

- root hub (radice) - collegata a PCIe (o al South-Bridge)
- USB bay (nodi intermedi) - hub di espansione
- devices (foglie) - tastiera, mouse, scanner, webcam, memoria di massa, audio, ...

Comunicazione esclusivamente tra radice e foglie: i device non comunicano tra loro.

Connessione distribuita su più circuiti virtuali in grado di trasmettere tipi di dati diversi: fino a 16 canali individuali full-duplex per dispositivo.

USB: inserimento di un nuovo dispositivo

Quando una nuova foglia viene inserita, root hub

- identifica l'evento
- interroga il dispositivo per ottenere dati su tipo, banda richiesta, driver di sistema associato
- esegue una chiamata al sistema operativo, che termina dopo che è stata verificata la disponibilità di driver in grado di comunicare con quel dispositivo
- assegna un indirizzo (ID) unico (1-127).

USB: framing

La comunicazione è organizzata in frame (pacchetti sincronizzati di dati).

Per mantenere la sincronizzazione tra gli orologi interni dei dispositivi, esattamente ogni 1 msec root hub invia un frame che viene letto da tutti i dispositivi (frame broadcast).

Esistono 4 tipi di frame

USB: framing

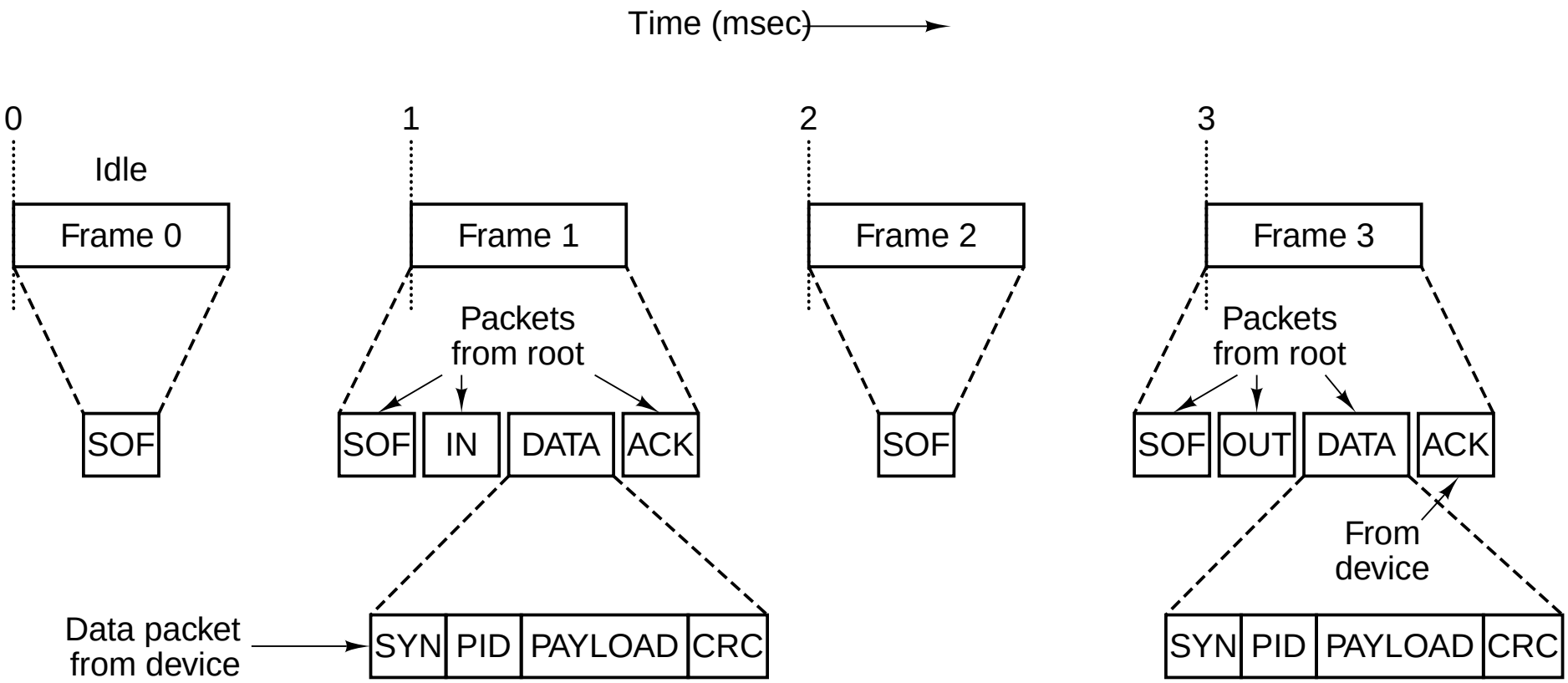
La comunicazione è organizzata in frame (pacchetti sincronizzati di dati).

Per mantenere la sincronizzazione tra gli orologi interni dei dispositivi, esattamente ogni 1 msec root hub invia un frame che viene letto da tutti i dispositivi (frame broadcast).

Esistono 4 tipi di frame

- control: comandi al dispositivo e diagnostica
- bulk: dati
- isochronous: per dispositivi tempo reale
- interrupt: per chiamare una procedura di sistema.

USB: comunicazione



Frame 1 e 3 rappresentano tre transazioni:

1. interrogazione lettura/scrittura (SOF + IN/OUT)
2. trasmissione (DATA)
3. riscontro (ACK).

USB: struttura dei frame

Ogni frame contiene dei sottopacchetti

- token (da root al dispositivo) per il controllo della comunicazione: SOF, IN, OUT, SETUP

USB: struttura dei frame

Ogni frame contiene dei sottopacchetti

- token (da root al dispositivo) per il controllo della comunicazione: SOF, IN, OUT, SETUP
- data (nelle due direzioni) con formato: sincronizzazione (SYN), tipo di pacchetto (PID), dati (PAYLOAD), codice di controllo (CRC)

USB: struttura dei frame

Ogni frame contiene dei sottopacchetti

- token (da root al dispositivo) per il controllo della comunicazione: SOF, IN, OUT, SETUP
- data (nelle due direzioni) con formato: sincronizzazione (SYN), tipo di pacchetto (PID), dati (PAYLOAD), codice di controllo (CRC)
- handshake: ACK, NAK, STALL

USB: struttura dei frame

Ogni frame contiene dei sottopacchetti

- token (da root al dispositivo) per il controllo della comunicazione: SOF, IN, OUT, SETUP
- data (nelle due direzioni) con formato: sincronizzazione (SYN), tipo di pacchetto (PID), dati (PAYLOAD), codice di controllo (CRC)
- handshake: ACK, NAK, STALL
- special.