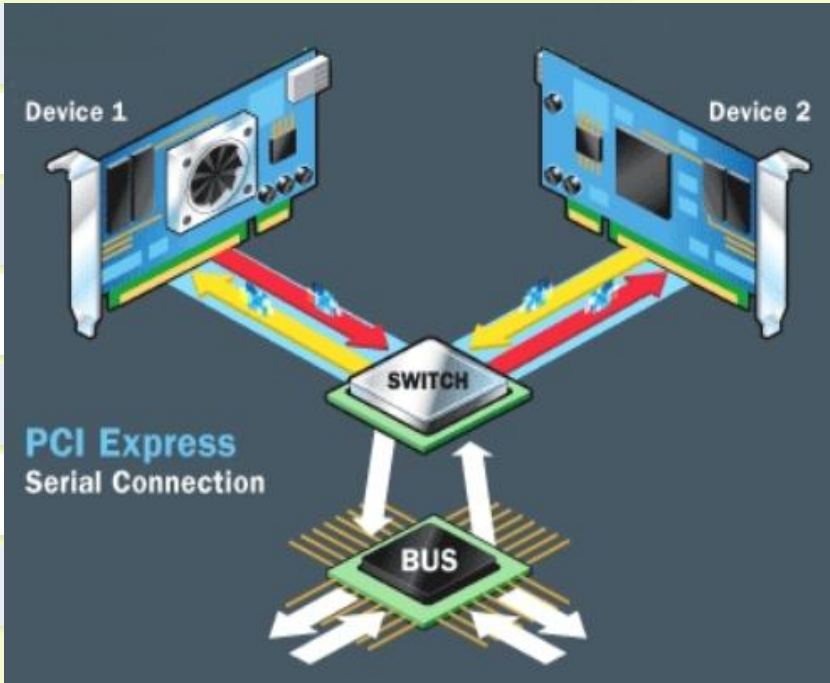
C12 PCI Express

Arhitectură, Protocol și Evoluție

Cuprins



Introducere și Istoric

De la magistrala paralelă PCI la legăturile seriale PCIe

Ce este PCI Express?

- PCI Express (PCIe) este un standard de magistrală de extensie serială de mare viteză
- Dezvoltat de Intel și standardizat de PCI-SIG (PCI Special Interest Group)
- Înlocuiește arhitecturile mai vechi de magistrală paralelă PCI, PCI-X și AGP
- Topologie de legătură serială punct-la-punct vs. magistrală paralelă partajată
- Interconexiunea I/O fundamentală pentru PC-uri, servere și sisteme embedded moderne
- Lățime de bandă scalabilă prin benzi/lanes multiple (×1, ×2, ×4, ×8, ×16, ×32)
- Utilizat pentru GPU-uri, SSD-uri (NVMe), plăci de rețea, controlere RAID etc.

- Introdus sub denumirea de "Third Generation I/O" (3GIO) în 2002, PCI Express (PCIe) a înlocuit atât PCI, cât și PCI-X, iar noile plăci de bază pot veni cu un mix de sloturi PCI și PCIe sau numai PCIe.
- *PCIe seamănă mai mult cu o rețea*, fiecare placă fiind conectată la un switch printr-un set de fire dedicate.

Cronologia evoluției PCIe

- 1992 – PCI 1.0 lansat: magistrală paralelă partajată 32 biți, 33 MHz (133 MB/s)
- 1998 – PCI-X introdus: 64 biți, 133 MHz, lățime de bandă mai mare pentru servere
- 2003 – PCIe 1.0 lansat: 2,5 GT/s, arhitectură serială punct-la-punct
- 2007 – PCIe 2.0: viteză dublată la 5,0 GT/s per bandă
- 2010 – PCIe 3.0: 8,0 GT/s cu codificare 128b/130b
- 2017 – PCIe 4.0: 16 GT/s, adoptat pe scară largă în platformele moderne
- 2019 – PCIe 5.0: 32 GT/s, implementat în servere și desktop-uri de vârf
- 2022 – PCIe 6.0 specificație finalizată: 64 GT/s cu semnalizare PAM-4

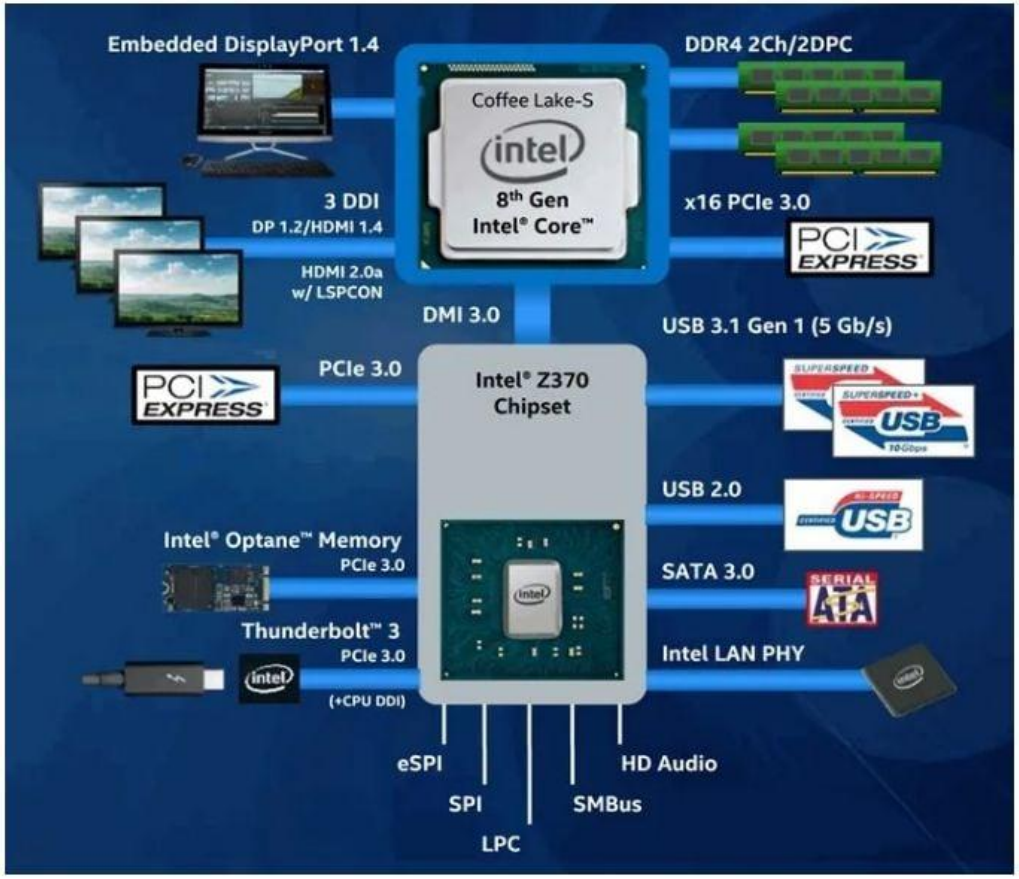
PCI vs. PCI Express

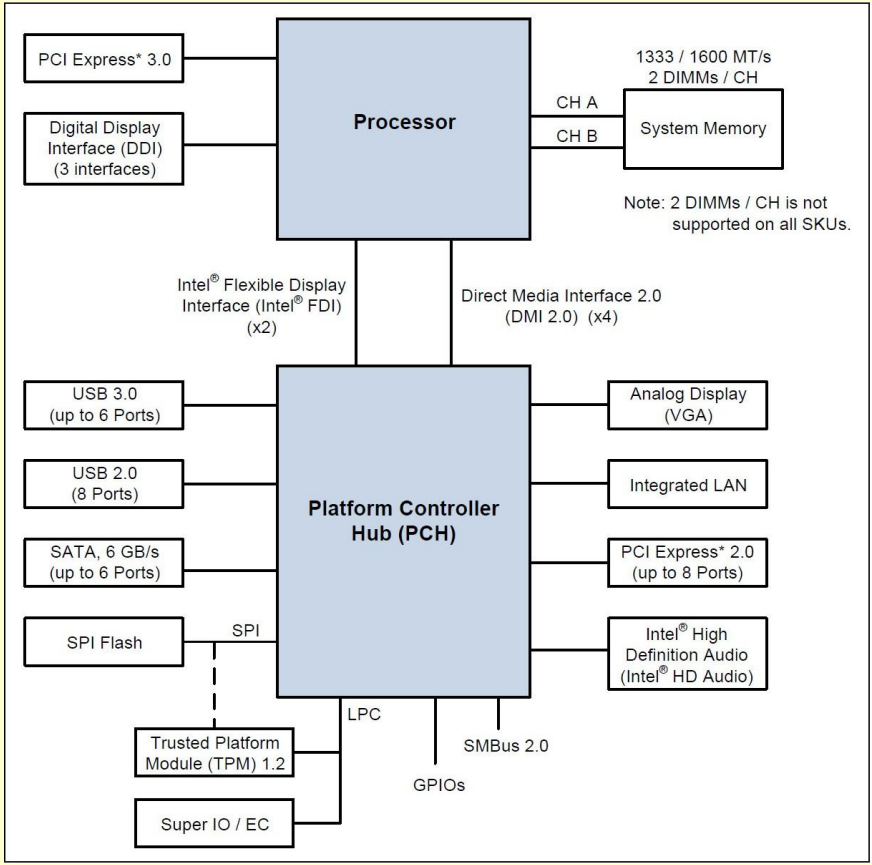
PCI tradițional (Paralel)

- Magistrală partajată – toate dispozitivele pe aceleași fire
- Cale de date de 32 sau 64 biți
- Frecvență de ceas 33/66 MHz
- Max 533 MB/s (PCI-X 2.0)
- Necesită arbitrare pe magistrală
- Semnalizare single-ended
- Scalabilitate limitată

PCIe (Serial)

- Legături dedicate punct-la-punct
- Perechi diferențiale seriale (benzi/lanes)
- 2,5 GT/s până la 64 GT/s per bandă
- Până la 128 GB/s (×16, Gen 6)
- Fabric bazat pe switch-uri, fără arbitrare
- Semnalizare diferențială de tensiune joasă (LVDS)
- Scalează prin lățimea benzii și generație

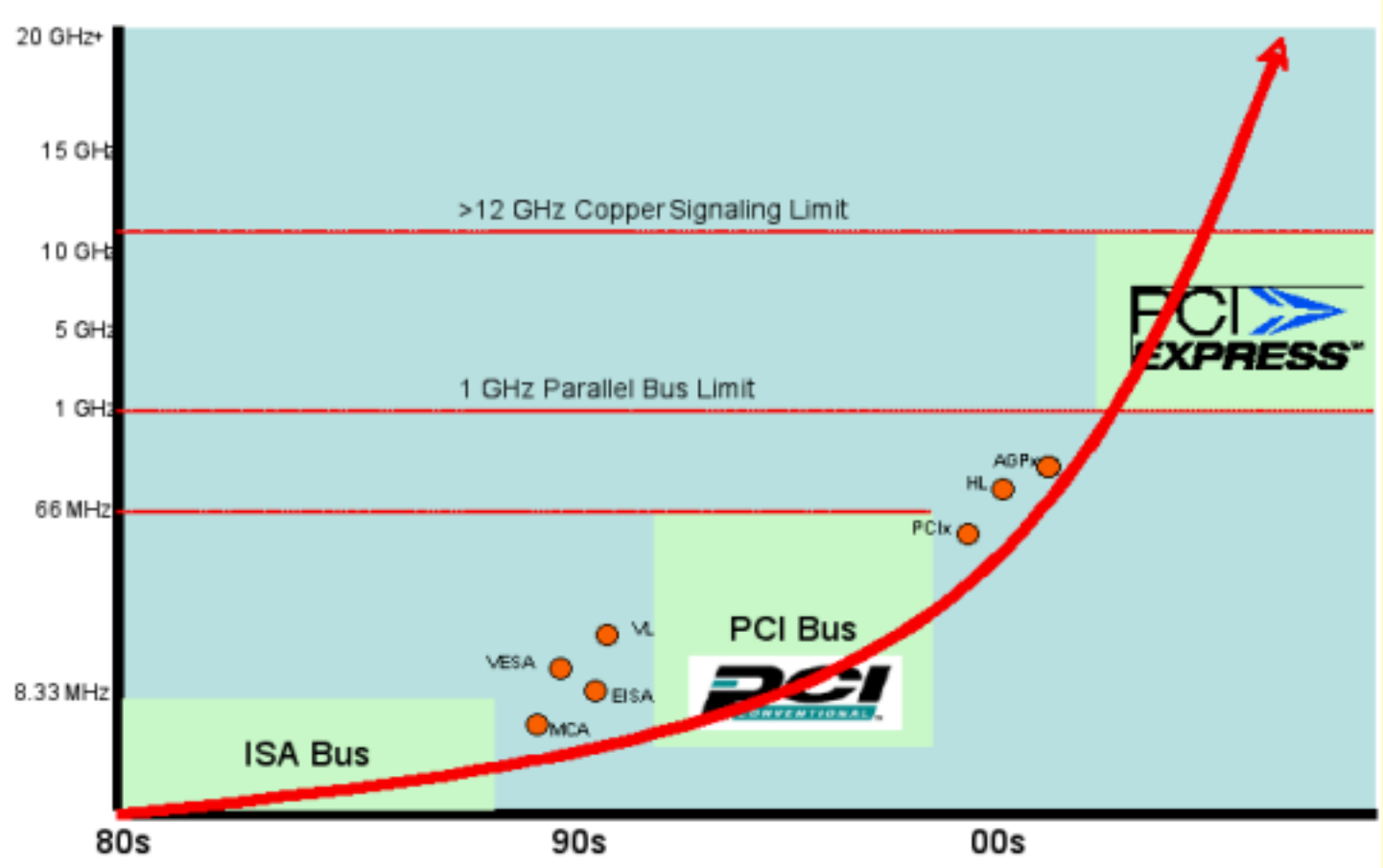




Traseul albastru evidențiază magistrala PCIe

Avantajele cheie ale PCIe

- **Lățime de bandă mai mare:** benzile seriale depășesc cu mult limitele magistralei paralele
- **Număr mai mic de pini:** pereche diferențială per bandă vs. magistrală paralelă largă
- **Funcționare full-duplex:** transmisie și recepție simultane
- **Conexiuni punct-la-punct:** lățime de bandă dedicată per dispozitiv, fără competiție
- **Compatibil software cu PCI:** același model de spațiu de configurare
- **Suport hot-plug și hot-swap** integrat în specificație
- **Detecție/corecție avansată a erorilor** (ECRC, LCRC, replay)
- **Management al puterii** (ASPM, stări L)



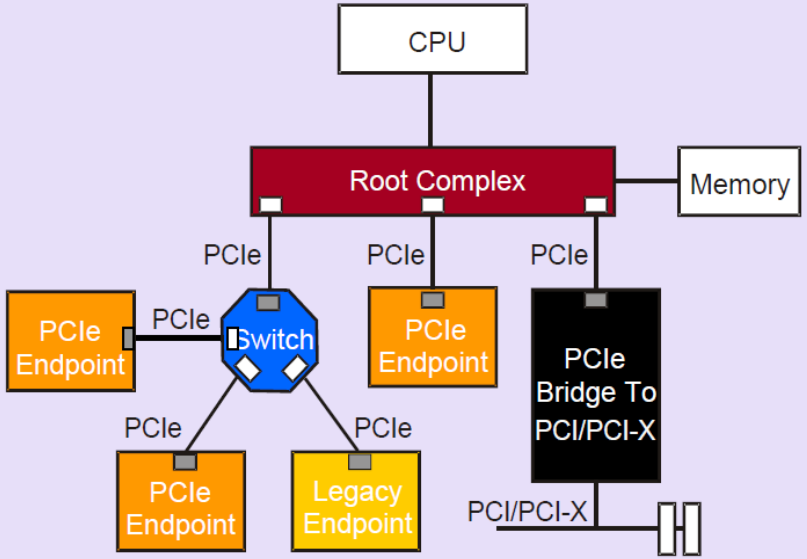
- "Industria electronică a adoptat PCIe ca standardul dominant, cea mai bună interconectare de mare viteză pentru comunicațiile între chipuri și plăci în cadrul unui sistem. Pur și simplu nu există altă alegere viabilă."

Prezentare generală a arhitecturii

Topologie, componente și structura legăturii

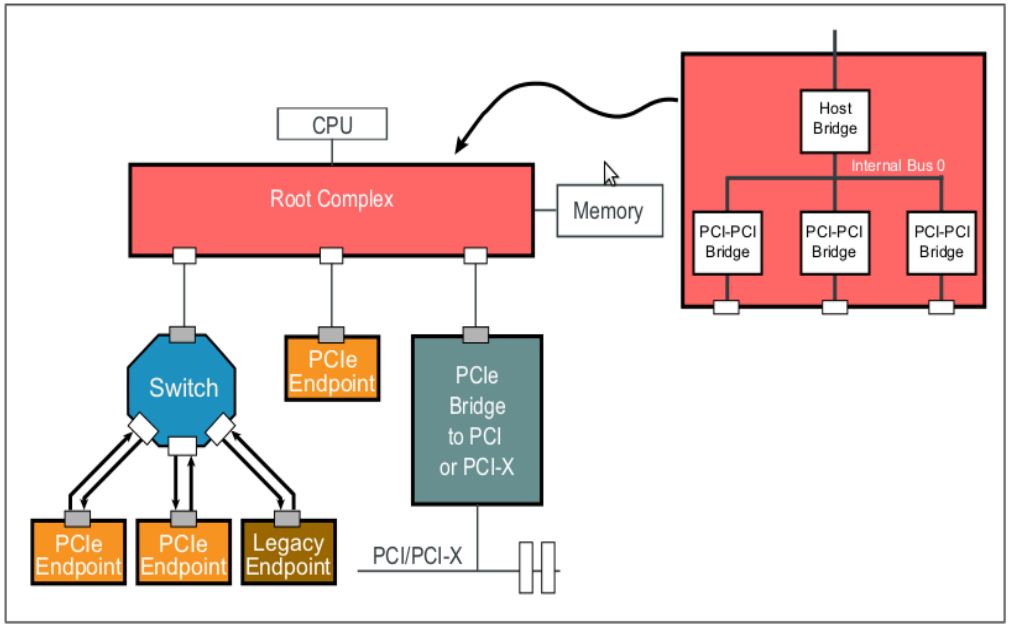
Topologia sistemului PCIe

- Structură arborescentă/ierarhică cu rădăcina în Root Complex (RC)
- **Root Complex conectează subsistemul CPU/memorie la fabric-ul PCIe**
- **Switch-urile PCIe oferă fan-out** (porturi downstream multiple)
- **Endpoint-urile sunt dispozitivele terminale** (GPU-uri, SSD-uri NVMe, plăci de rețea etc.)
- **Bridge-urile asigură compatibilitate** cu
- segmente PCI/PCI-X mai vechi
- Fiecare legătură (link) este o conexiune punct-la-punct între 2 porturi
- Topologia este enumerată și configurată de software-ul de sistem (BIOS/OS)



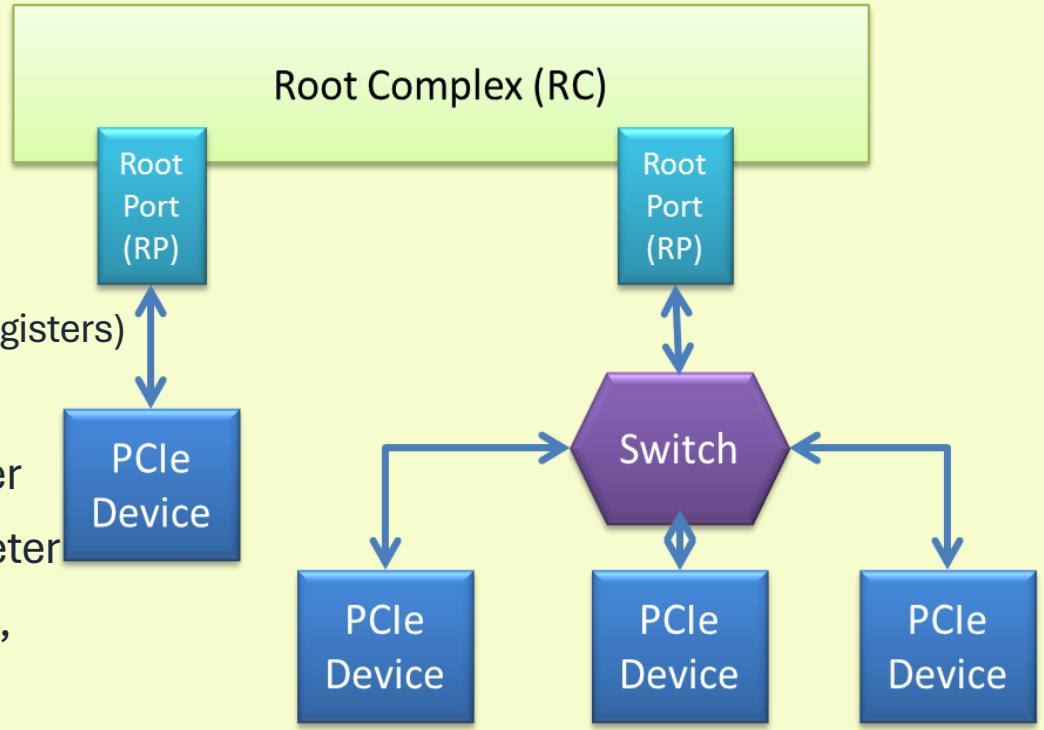
Root Complex (RC)

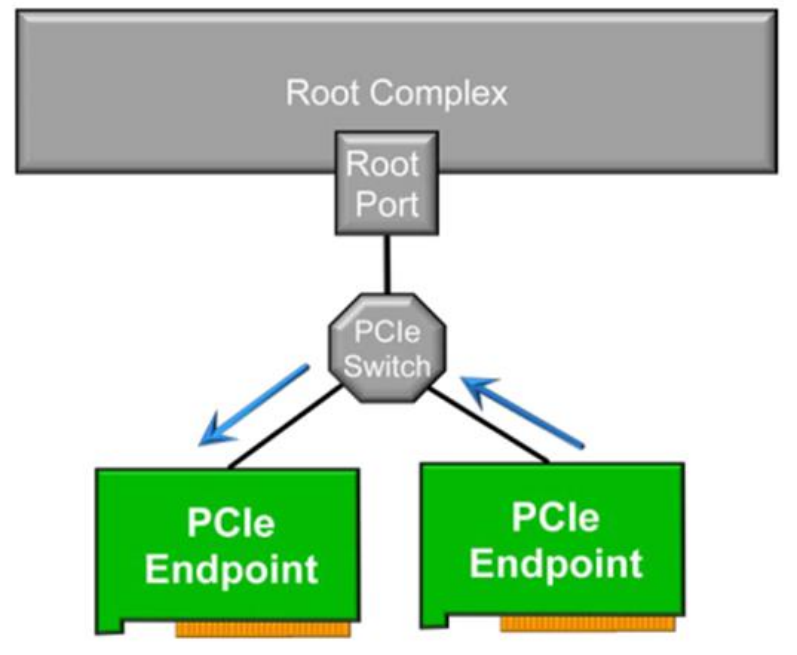
- **Este originea ierarhiei tranzacțiilor PCIe** – vârful arborelui
- De obicei **integrat în CPU sau chipset** (ex. Intel PCH, AMD SoC)
- Oferă **unul sau mai multe Root Ports**, fiecare ofera o legătură PCIe independentă
- Generează **tranzacții de citire/scriere, configurare** în timpul enumerării
- Rutează I/O mapate în memorie (MMIO) și tranzacții de port I/O
- Gestionează rutarea întreruperilor (INTx, MSI, MSI-X)
- **Poate include endpoint-uri integrate** (ex. controlere USB, SATA încorporate)

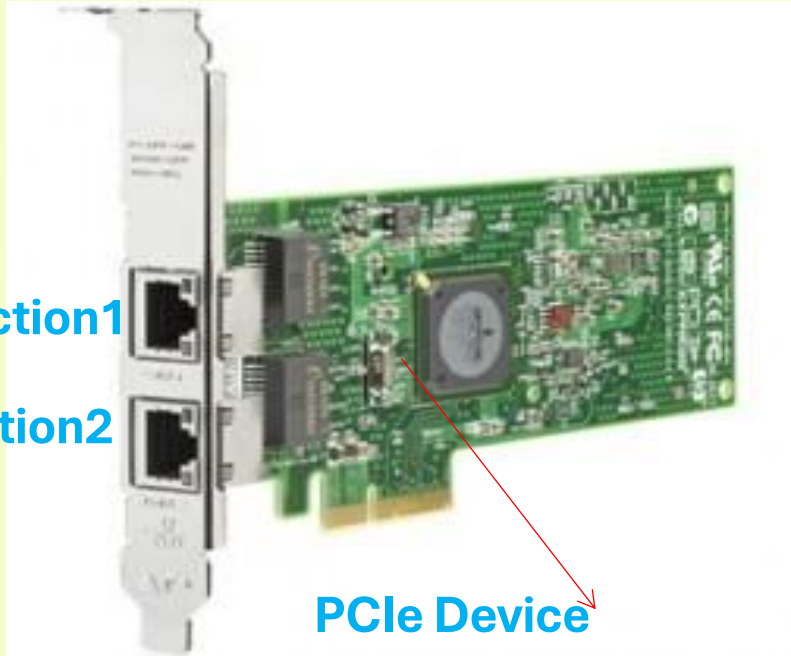


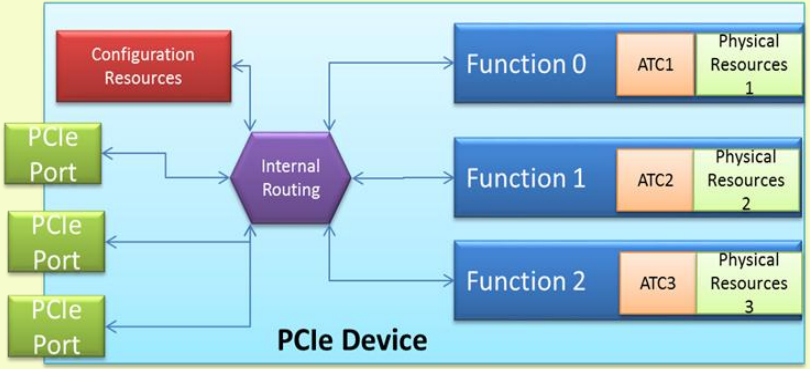
Endpoint-uri PCIe

- **Dispozitive terminale** care inițiază sau completează tranzacții
- Două tipuri: **Endpoint-uri tradiționale (suportă tranzacții I/O) și Endpoint-uri native**
- Endpoint-urile PCIe native nu trebuie să genereze cereri I/O (doar MMIO)
- Fiecare endpoint are una sau mai multe funcții (până la 8/dispozitiv)
- Funcțiile conțin BAR-uri (Base Address Registers) pentru maparea memoriei
- Endpoint-urile pot fi solicitanți /requester (inițiază tranzacții) sau completori/completer
- Exemple: plăci grafice, controlere NVMe, adaptoare Ethernet etc




Root Complex
Root Port
PCIe Switch
PCIe Endpoint
PCIe Endpoint


Function1
Function2
PCIe Device


Configuration Resources
PCIe Port
PCIe Port
PCIe Port
Internal Routing
Function 0
ATC1
Physical Resources 1
Function 1
ATC2
Physical Resources 2
Function 2
ATC3
Physical Resources 3
PCIe Device

Switch-uri și Bridge-uri

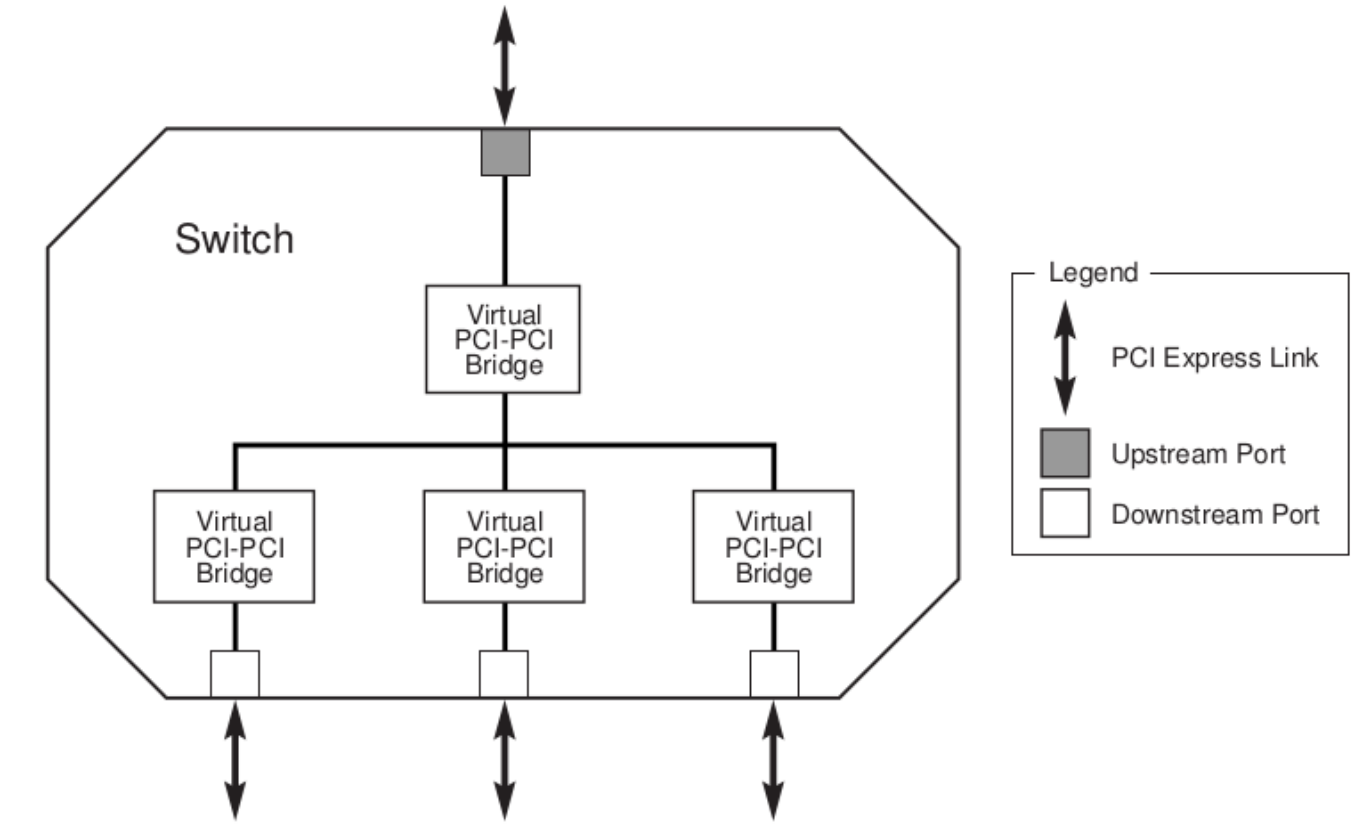
Switch-uri PCIe

- Oferă fan-out de la un singur port upstream
- Porturi downstream multiple către endpoint-uri
- Rutarea pachetelor prin adresă sau ID
- Tranzacții peer-to-peer posibile
- Un port upstream + N porturi downstream
- Acționează ca bridge-uri virtuale PCI-to-PCI

Bridge-uri PCI/PCIe

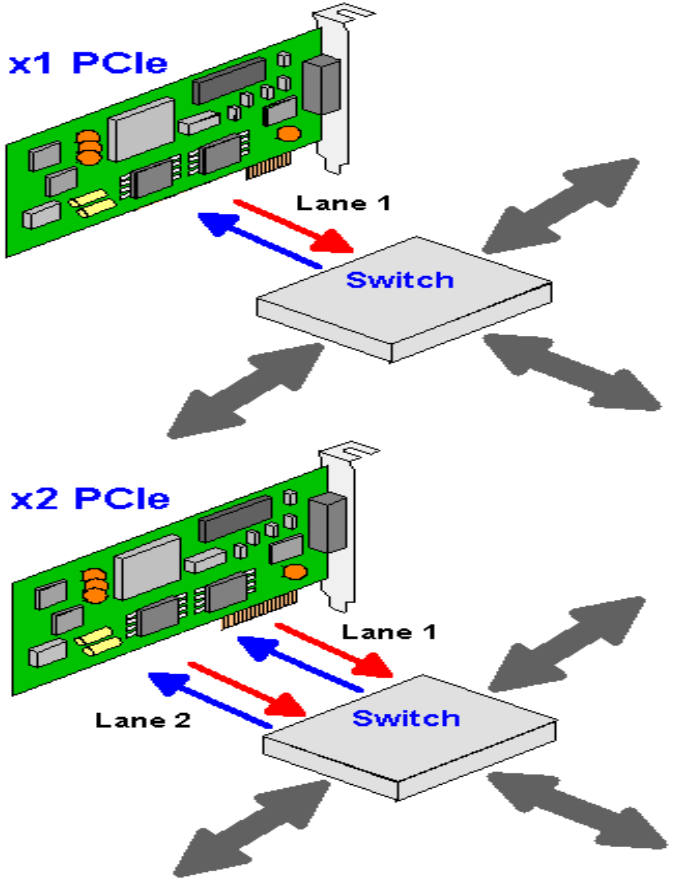
- Punțile oferă *interfațarea PCIe la alte bus-uri*, ca PCI/PCI X, USB, InfiniBand, Ethernet, Fiber channel sau chiar la un alt PCI-e bus
- Traducere de protocol între domenii
- Bridge direct: PCIe → PCI
- Bridge invers: PCI → PCIe
- Mențin regulile de ordonare între domenii
- Din ce în ce mai rare în sistemele moderne

Switch-uri și Bridge-uri



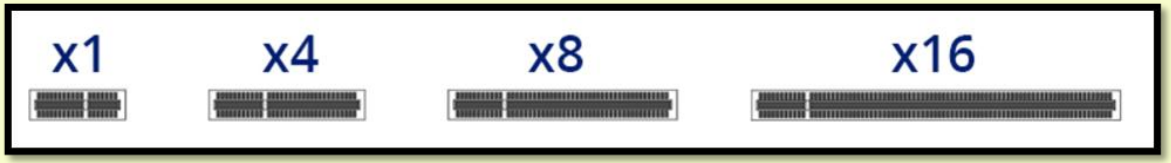
Benzi (Lanes) și Legături (Links)

- O Bandă (Lane) = o pereche de semnale diferențiale (TX + RX) = full-duplex
- O Legătură (Link) este o colecție de 1, 2, 4, 8, 12, 16 sau 32 de benzi
- Lățimea legăturii este notată ×1, ×4, ×8, ×16 etc.
- Datele sunt distribuite pe benzi octet-cu-octet (byte striping)
- Legătura se poate antrena la o lățime mai mică dacă nu toate benzile funcționează
- Fiecare bandă transportă aceeași rată de date (GT/s depinde de generație)
- Legături mai largi = proporțional mai multă lățime de bandă
- Inversarea benzilor și inversarea polarității sunt gestionate automat la antrenare



Factori de formă fizici

- Sloturi standard pentru plăci add-in (AIC): conectori edge ×1, ×4, ×8, ×16
- Un slot ×16 poate accepta plăci ×1, ×4, ×8 (compatibil retroactiv)
- O placă ×1 poate funcționa într-un slot ×16 (sloturi cu capăt deschis)
- M.2 (NGFF): factor de formă compact pentru SSD-uri și plăci wireless (key M, B etc.)
- U.2 (SFF-8639): factor de formă 2,5" pentru SSD-uri NVMe enterprise
- EDSFF (E1.S, E1.L, E3.S): factori de formă de nouă generație pentru centre de date
- Specificația CEM definește dimensiunile plăcii, puterea și pinout-ul semnalelor
- Puterea plăcii: 25W (×1), 25W (×4), 75W (×16), până la 600W cu alimentare auxiliară



Nivelurile protocolului

Nivelurile Tranzacție, Legătură de date și Fizic

Stiva de niveluri a protocolului PCIe

- PCIe folosește o **arhitectură de protocol pe trei niveluri** (similar modelului OSI)
- **Nivelul Tranzacție (TL)** – generează/consumă pachete (TLP-uri)
- **Nivelul Legătură de Date (DLL)** – asigură livrarea fiabilă, adaugă secvență și CRC
- **Nivelul Fizic (PHY)** – codificare, serializare, semnalizare electrică
- Nivelul software deasupra – driverele și OS-ul interacționează prin TL
- Fiecare nivel adaugă/elimină propriul antet/trailer (încapsulare)
- Separarea clară permite evoluția independentă a fiecărui nivel

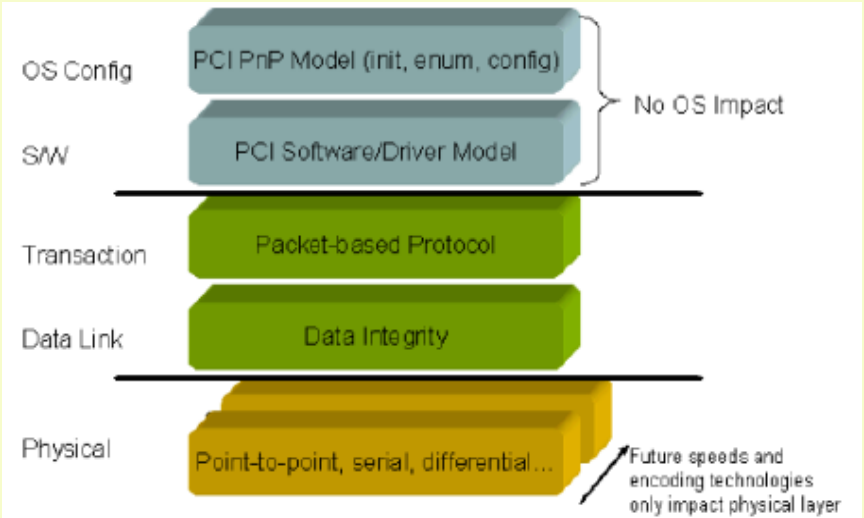


Figure 2. PCI Express Layered Architecture

Tipuri de pachete TLP

Description	Abbreviated Name
Memory Read Request	MRd
Memory Read Request – Locked Access	MRdLk
Memory Write Request	MWr
IO Read Request	IORd
IO Write Request	IOWr
Configuration Read Request Type 0 and Type 1	CfgRd0, CfgRd1
Configuration Write Request Type 0 and Type 1	CfgWr0, CfgWr1
Message Request without Data Payload	Msg
Message Request with Data Payload	MsgD
Completion without Data (used for IO, configuration write completions and read completion with error completion status)	Cpl
Completion with Data (used for memory, IO and configuration read completions)	CplD
Completion for Locked Memory Read without Data (used for error status)	CplLk
Completion for Locked Memory Read with Data	CplDLk

Nivelul Legătură de Date (DLL)

- **Asigură livrarea fiabilă a TLP-urilor** între două porturi conectate direct
- Adaugă **un Număr de Secvență de 12 biți fiecărui TLP** pentru ordonare și reluare
- Anexează **LCRC pe 32 biți (Link CRC)** pentru detecția erorilor la nivel de legătură
- **Protocol ACK/NAK:** receptorul trimite ACK (succes) sau NAK (eroare detectată)
- **Buffer de reluare:** transmițătorul reține TLP-urile până la ACK; reia la NAK/timeout
- **Generează DLLP-uri** (Data Link Layer Packets) pentru gestionarea legăturii
- **Tipuri DLLP: ACK, NAK, FC Init, FC Update, Power Management**
- Gestionează inițializarea controlului fluxului la nivel de legătură

Protocolul ACK/NAK și reluarea

- Fiecare TLP transmis primește un număr de secvență (12 biți, se resetează la 4095)
- Receptorul verifică LCRC; dacă e valid, trimite DLLP ACK cu numărul de secvență
- Dacă se detectează eroare CRC, receptorul trimite DLLP NAK
- La NAK sau timeout (REPLAY_TIMER), transmițătorul reia de la TLP-ul eșuat
- Bufferul de reluare stochează toate TLP-urile neconfirmate
- Contorul REPLAY_NUM limitează reluările consecutive (evită buclele infinite)
- După numărul maxim de reîncercări, legătura intră în starea Recovery din LTSSM
- Garantează livrarea fiabilă fără intervenția nivelurilor superioare

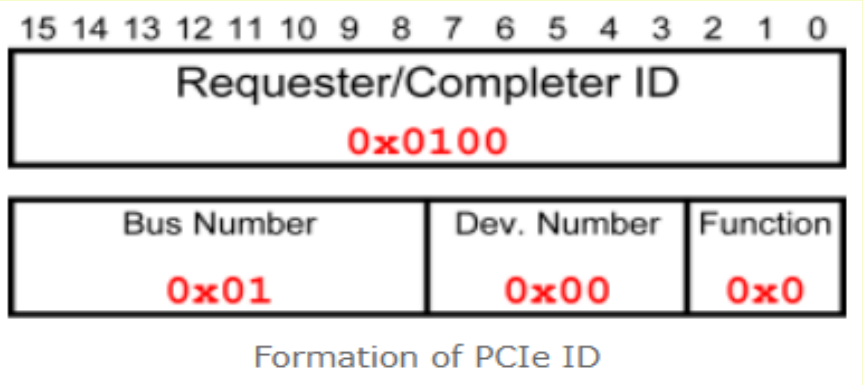
Identificare și rutare

- *Deoarece PCIe este în esență o rețea de pachete, cu posibilitatea de comutare pe cale, switch-urile trebuie să știe unde să trimită fiecare TLP*
- Sunt 3 metode de rutare:

  - Address routing este aplicată ptr. Memory și I/O Requests (read and write)

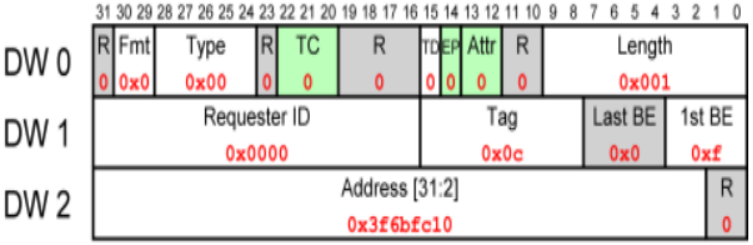
  - Implicit routing este folosit numai ptr. anumite mesaje TLPs, ca broadcasts de la Root Complex și mesaje care merg întotdeauna la Root Complex.

  - ID routing – toate celelalte TLPs sunt rutate de ID.
- ID este un cuvânt de 16-biti format din 3 câmpuri: *Bus number, Device number și Function number*. Sensul lor este exact ca la bus-urile PCI tradiționale.
- ID-ul se formează după cum urmează:

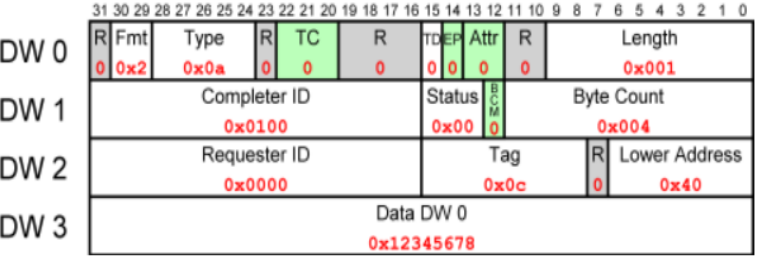


Formation of PCIe ID

CPU MRd către un Endpoint

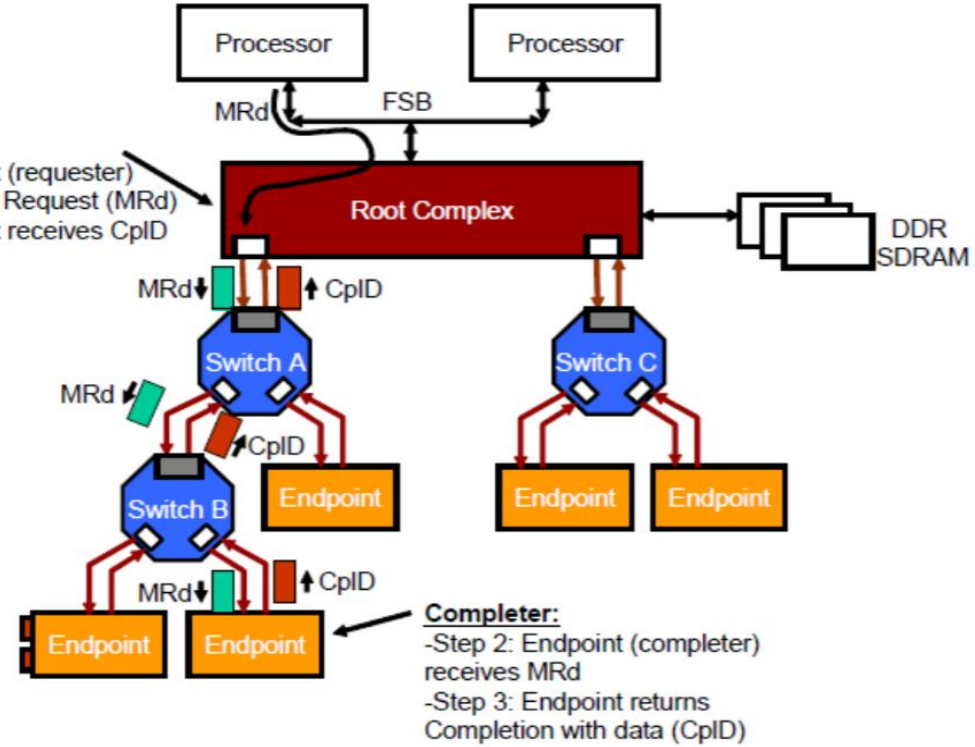


Example of Memory Read Request TLP

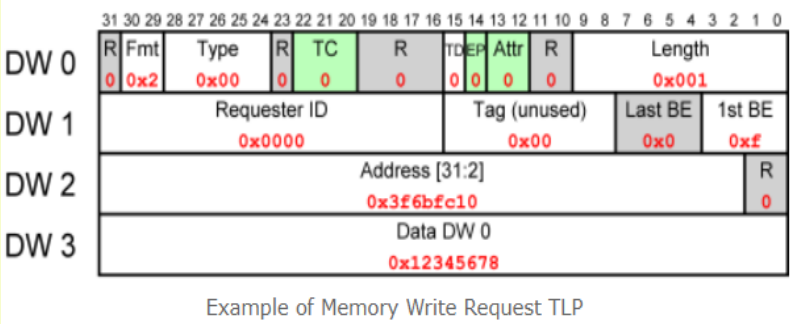


Example of Completion TLP

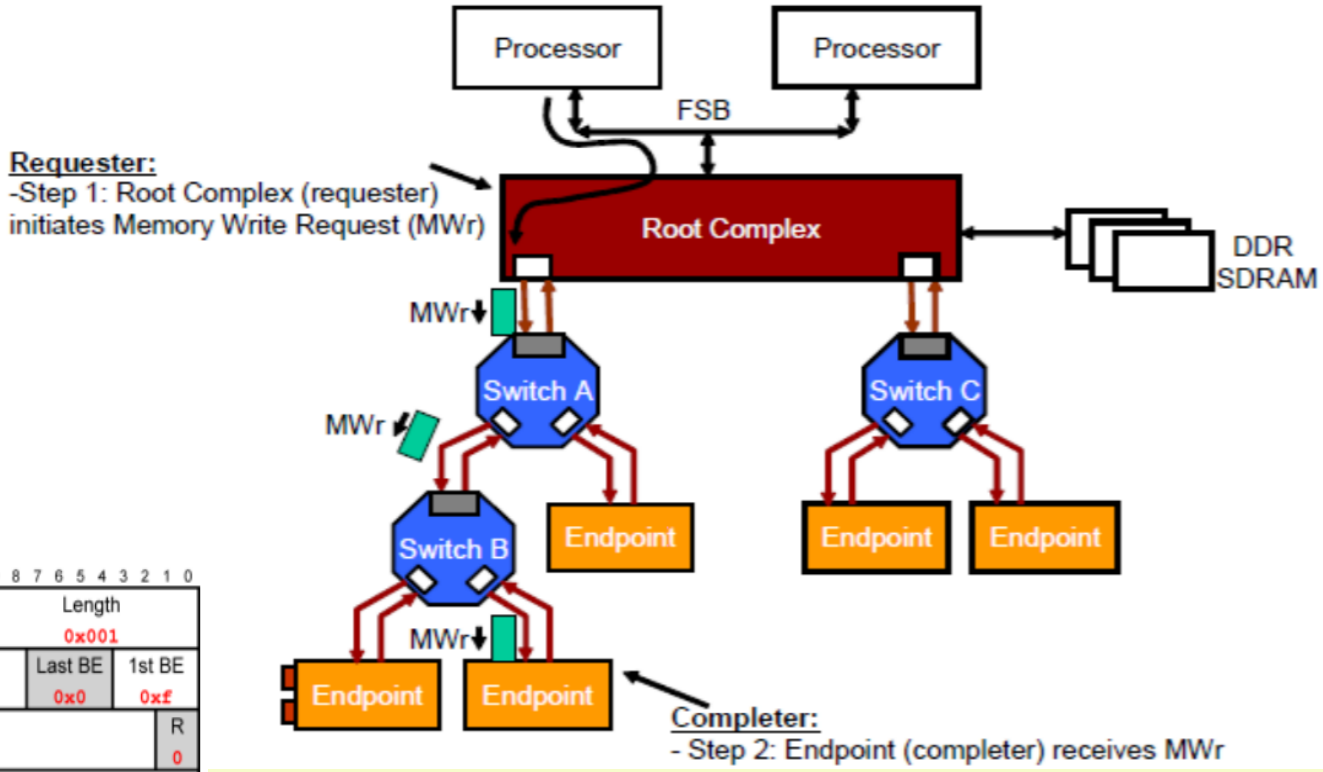
Requester:
-Step 1: Root Complex (requester) initiates Memory Read Request (MRd)
-Step 4: Root Complex receives CplD



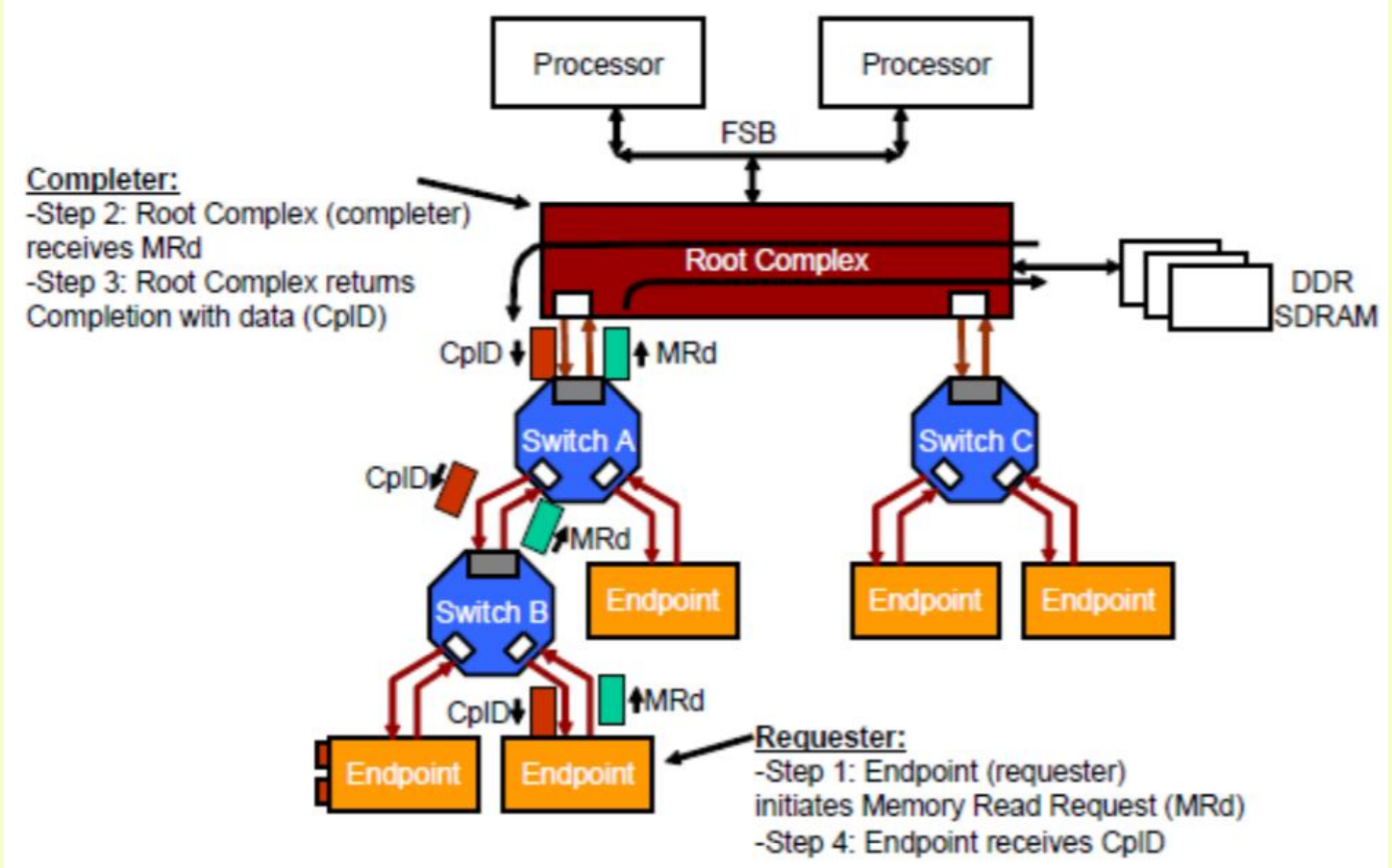
CPU MWr către Endpoint



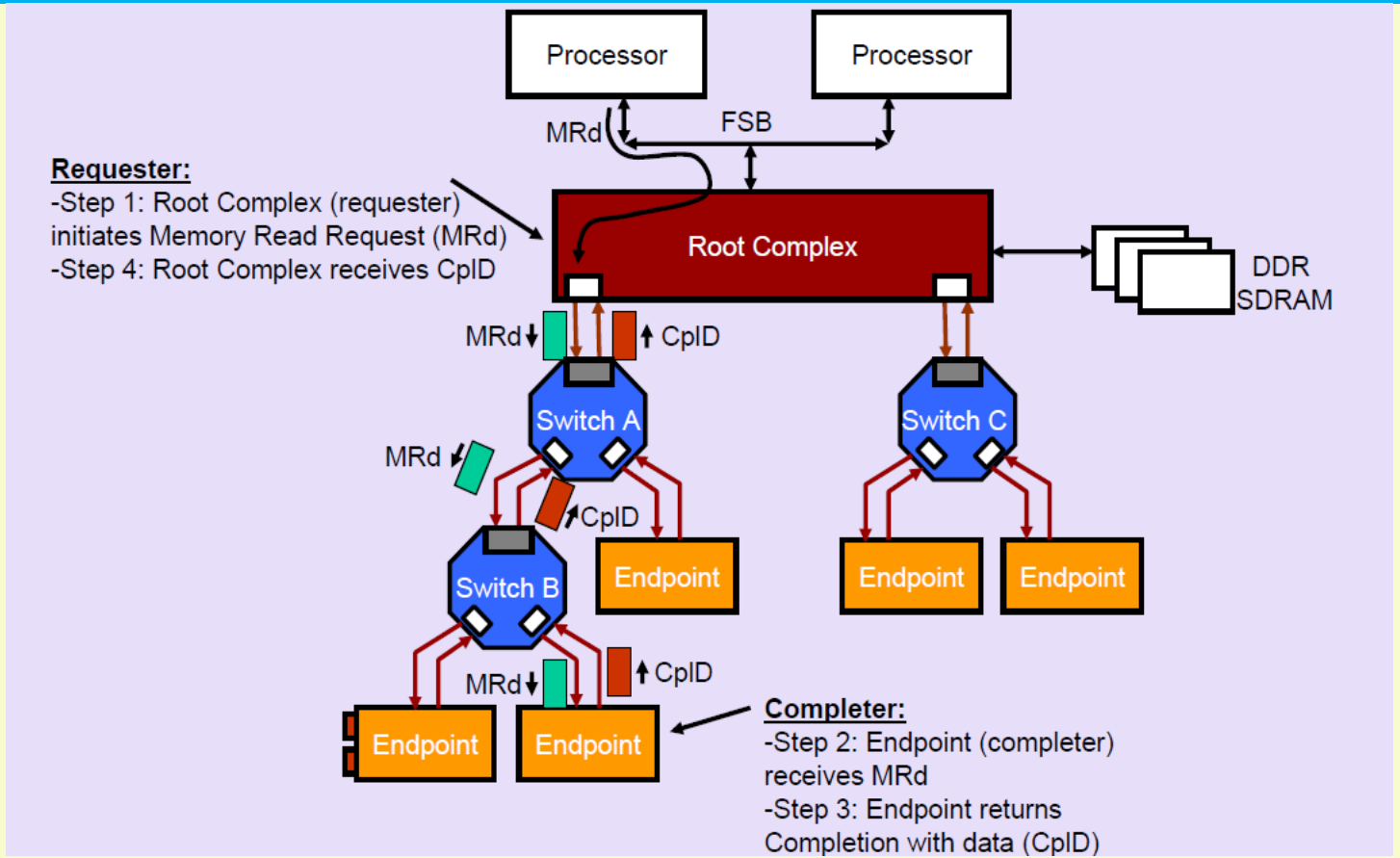
Example of Memory Write Request TLP



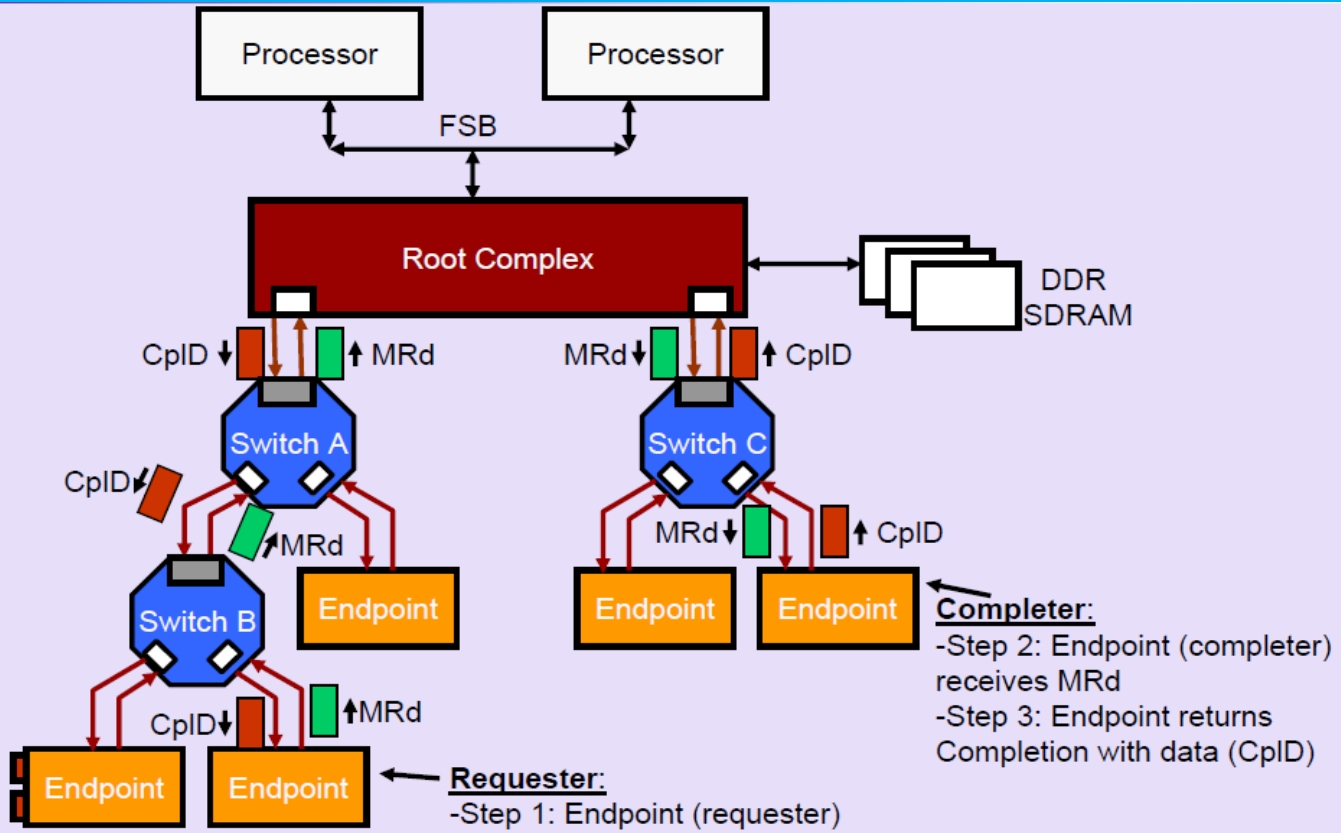
Endpoint MRd către memoria de sistem



Tranzacție I/O programată



Tranzacție Peer-to-Peer



Tranzacție DMA

Bus Mastering (DMA)

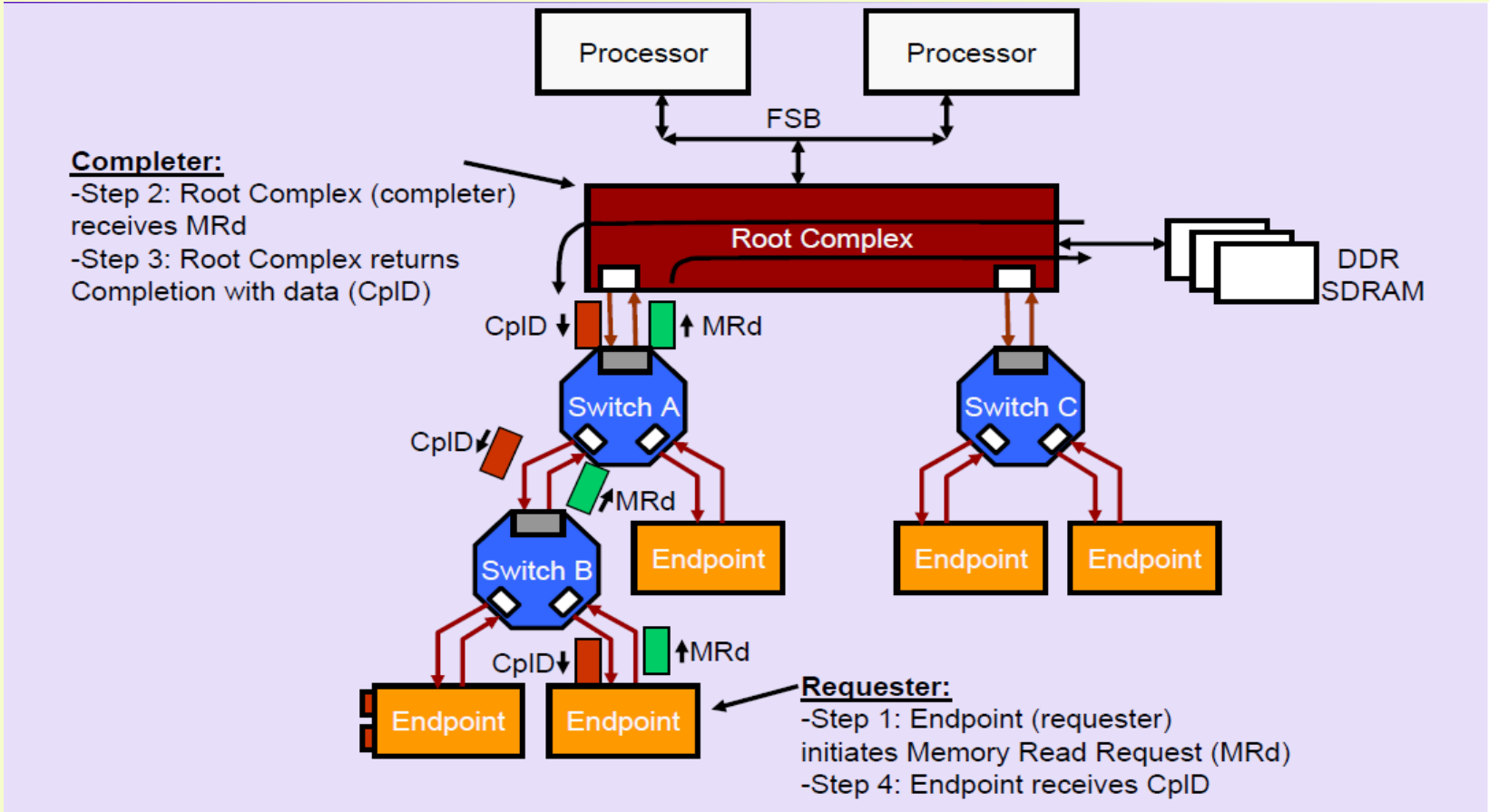
- Până la PCIe exista ceva intruziv în a spune CPU-ul să se retragă de pe bus pe durata DMA
- La PCIe, este mult mai ușor ca **oricine să poată trimite TLP-uri de citire/scriere** pe bus, exact ca Root Complex. Acest lucru permite perifericului să acceseze direct memoria procesorului (DMA) sau să facă schimb de pachete cu alte periferice de la egal la egal (în măsura în care entitățile de comutare acceptă acest lucru)

- Există două lucruri care trebuie să se întâmple întâi, ca și în cazul oricărui dispozitiv PCI:

1. Perifericul trebuie să primească **controlul bus-ului prin setarea *bitului "Bus Master Enable"*** într-unul din registrele standard de configurare.

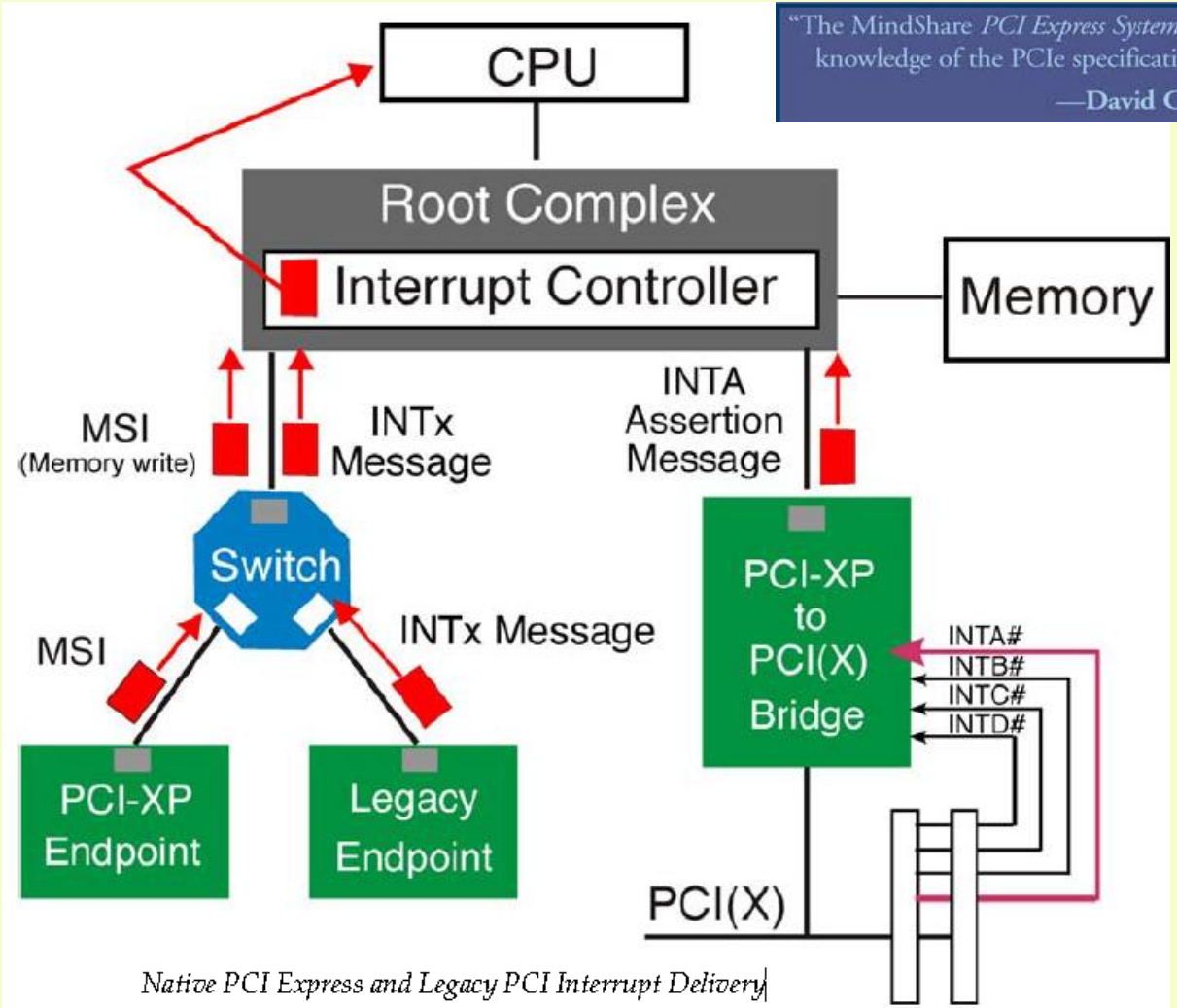
2. **Driver-ului trebuie să informeze perifericul despre adresa fizică a buffer-ului** relevant, cel mai probabil scriind într-un registru mapat Base Address Register (configuration space).

Tranzacție DMA



Tranzacții de mesaje și întreruperi

- Mesajele înlocuiesc multe din semnalele sideband PCI cu TLP-uri in-band
- **Emulare INTx:** mesaje Assert_INTx / Deassert_INTx (compatibilitate tradițională)
- **MSI (Message Signaled Interrupts):** scriere în memorie la adresa de întrerupere CPU
- **MSI-X: MSI îmbunătățit cu mascare** per-vector și până la 2048 vectori
- **Mesaje PME** (Power Management Event)
- **Mesaje de eroare:** ERR_COR, ERR_NONFATAL, ERR_FATAL
- Mesaje definite de furnizor pentru extensii proprietare
- *MSI-X este preferat în sistemele moderne pentru scalabilitate și performanță*



Native PCI Express and Legacy PCI Interrupt Delivery

Nivelul Fizic - Sub-nivelul Logic

- Codificare: 8b/10b (Gen 1–2), 128b/130b (Gen 3–5), PAM-4 + FLIT (Gen 6)
- 8b/10b: mapează 8 biți de date în simbol de 10 biți (20% overhead)
- 128b/130b: scramblează 128 biți, antet de sincronizare de 2 biți (1,56% overhead)
- Scrambling-ul cu LFSR îmbunătățește spectrul semnalului și reduce EMI; echilibrul DC este în general statistic, nu absolut (Linear Feedback Shift Register)
- Ordered Sets: simboluri speciale pentru antrenare, skip, idle (caractere comma)
- Deskew între benzi: compensează diferențele de lungime a traseelor
- Buffer elastic: absoarbe diferențele de frecvență de ceas (toleranță ppm)

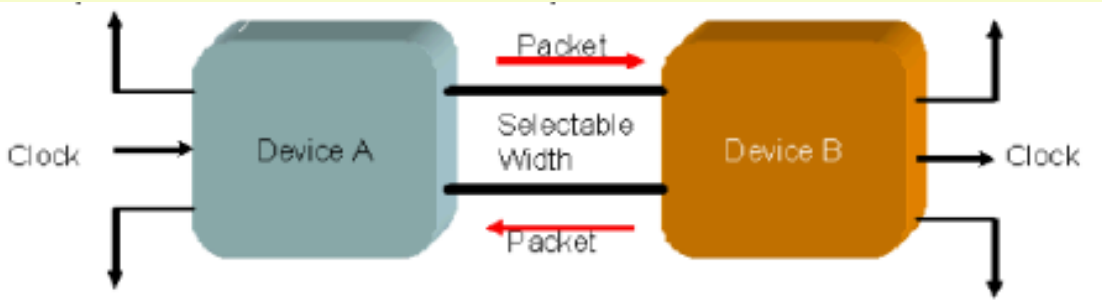
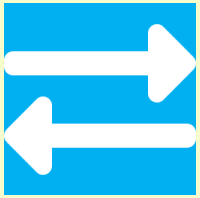


Figure 3. PCI Express Physical Link Diagram

Nivelul Fizic - Sub-nivelul Electric

- **Semnalizare diferențială: 0.8–1.2 V** vârf-la-vârf (Gen 1–2)
- Amplitudine mai mică la viteze mai mari pentru consum redus și integritate mai bună
- De-emphasis și pre-shoot la transmițător pentru egalizare (Gen 2+)
- Egalizare la receptor: DFE (Decision Feedback Equalizer) pentru Gen 3+
- CTLE (Continuous Time Linear Equalizer) în front-end-ul receptorului
- Arhitecturi de ceas de referință: ceas comun, ceas de date, ceas separat
- Cuplaj AC prin condensatoare pe fiecare bandă (blocare DC)
- Măsurarea diagramei ochi (eye diagram) pentru verificarea calității semnalului



Transferul datelor și pachete

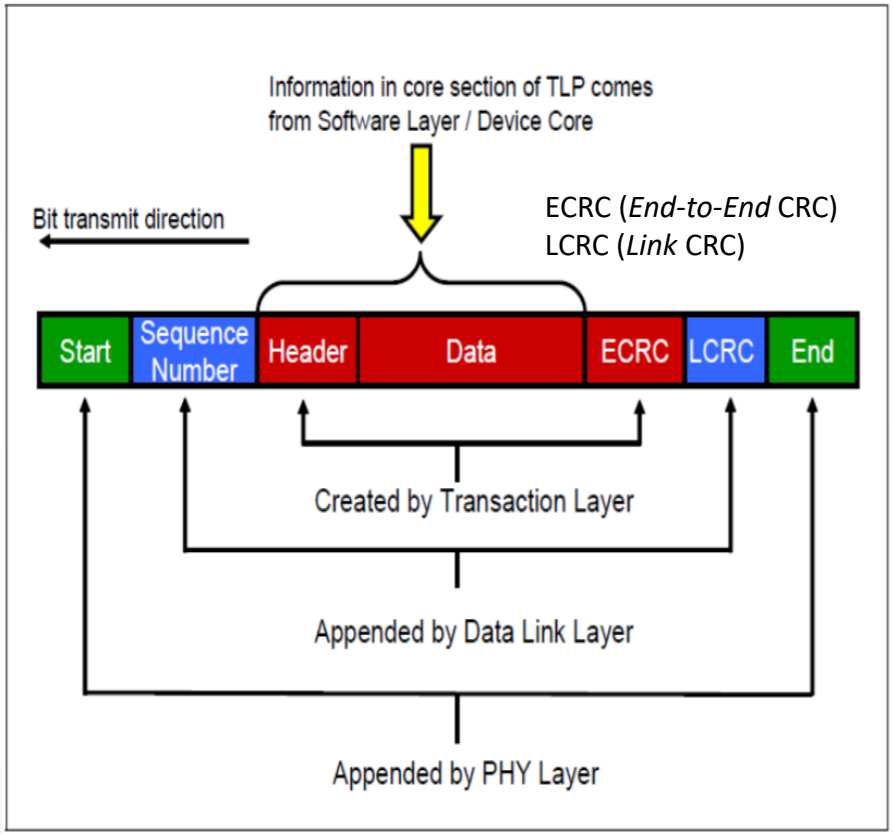
Structura pachetelor, moduri de adresare și întreruperi

Încapsularea TLP (împachetarea pachetului)

- Datele aplicației provin de la Nivelul Tranzacție
- **TL adaugă**: Antet (3 sau 4 DW) + Payload de date ECRC opțional (Digest)
- **DLL adaugă**: Număr de Secvență de 2 octeți la început, LCRC de 4 octeți la final
- **PHY adaugă**: tokeni de încadrare Start/End (STP, SDP, END)
- Pachetul complet pe fir:

ST | Seq# | Antet TLP | Payload | ECRC | LCRC | END

- Receptorul elimină nivelurile în ordine inversă
- DLLP-urile urmează o încapsulare similară dar mai simplă (fără antet TL)



Trecerea pachetului prin starturile logice a dispozitivului PCIe

Energie și tratarea erorilor

ASPM, stări L, AER și fiabilitate

Stările de management al energiei PCIe

- D0 (Activ): complet operațional, dispozitivul consumă putere maximă
- D1, D2: consum redus intermediar (specifice dispozitivului, opționale)
- D3hot: consum redus controlat software, contextul dispozitivului poate fi pierdut
- D3cold: alimentarea eliminată complet, dispozitivul trebuie reinițializat
- L0: legătura activă și operațională
- L0s: standby – latență de ieșire rapidă (~1 µs), economie moderată de energie
- L1: sleep mai profund al legăturii (~2–10 µs ieșire), economie mai mare de energie
- L2/L3: legătura oprită – utilizat în suspend/hibernare de sistem

Clasificarea erorilor PCIe

- **Erori corectabile:** hardware-ul poate recupera fără pierdere de date
- – Eroare receptor, TLP defect, DLLP defect, Timeout Replay Timer, Replay Num Rollover
- **Erori necorectabile non-fatale:** pierdere de date posibilă, dar legătura rămâne operațională
- – Completion Timeout, TLP otrăvit, Completare neașteptată
- **Erori necorectabile fatale:** fiabilitatea legăturii sau dispozitivului compromisă
- – Eroare de antrenare, Eroare protocol DLL, TLP malformat, Eroare control flux
- Severitatea erorilor poate fi reprogramată prin registrele AER
- Erorile sunt raportate prin mesaje ERR_COR, ERR_NONFATAL, ERR_FATAL către Root Complex

Performanță și lățime de bandă

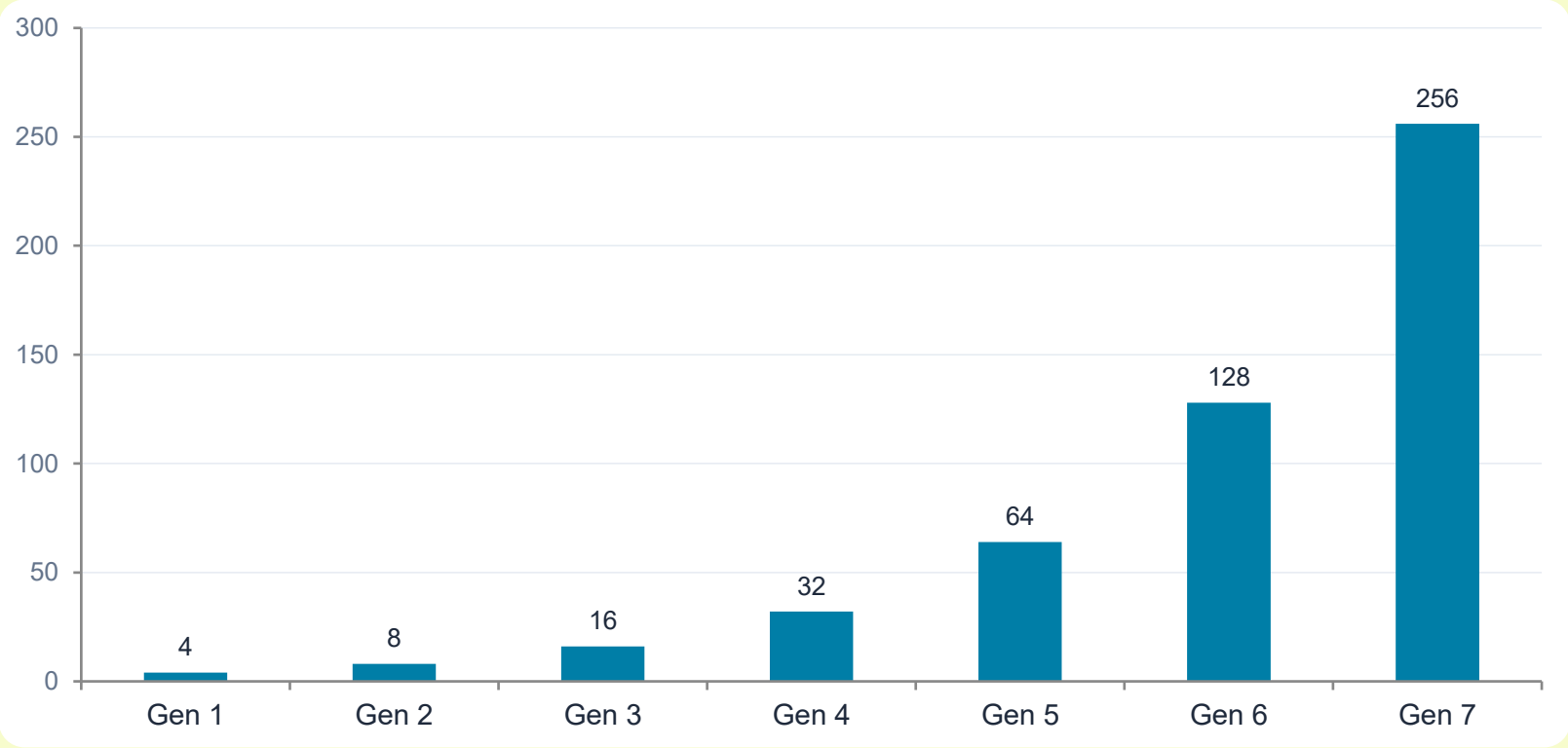
Rate de date, codificare și debit

Lățimea de bandă PCIe per generație

Gen	Rată (GT/s)	Codificare	×1 BW	×16 BW	An
1.0	2,5	8b/10b	250 MB/s	4 GB/s	2003
2.0	5,0	8b/10b	500 MB/s	8 GB/s	2007
3.0	8,0	128b/130b	~1 GB/s	~16 GB/s	2010
4.0	16,0	128b/130b	~2 GB/s	~32 GB/s	2017
5.0	32,0	128b/130b	~4 GB/s	~64 GB/s	2019
6.0	64,0	PAM-4 + FLIT	~8 GB/s	~128 GB/s	2022
7.0	128,0	PAM-4 + FLIT	~16 GB/s	~256 GB/s	~2025

* Valorile sunt per-direcție (full-duplex dublează banda agregată)

Creșterea lățimii de bandă ×16 (per direcție)



Eficiența codificării

Codificare 8b/10b (Gen 1-2)

- Fiecare octet de 8 biți este codificat ca simbol de 10 biți
- 20% overhead (2 biți pierduți/octet)
- Echilibru DC prin urmărirea disparității
- Caractere speciale (K-codes) pentru încadrare
- Gen 1: 2,5 GT/s → 2,0 Gb/s efectiv
- Simplă dar costisitoare la viteze mari

Codificare 128b/130b (Gen 3+)

- 128 biți de date + antet de sincronizare de 2 biți
- Doar ~1,54% overhead
- Scrambling LFSR pentru echilibru DC
- Nu sunt necesare simboluri speciale
- Gen 3: 8,0 GT/s → ~7,88 Gb/s efectiv
- Îmbunătățire majoră a eficienței față de 8b/10b

Generații PCIe și viitorul

Gen 5, Gen 6, Gen 7 și CXL

Detalii PCIe 5.0 și 6.0

PCIe 5.0 (2019)

- 32 GT/s per bandă, codificare 128b/130b
- Lățime de bandă ×16: ~64 GB/s per direcție
- Necesită egalizare îmbunătățită a canalului
- Implementat în Intel Sapphire Rapids, AMD Genoa
- CXL 1.0/2.0 rulează peste PHY PCIe 5.0
- SSD-uri enterprise și GPU-uri adoptă Gen5

PCIe 6.0 (2022)

- 64 GT/s per bandă prin semnalizare PAM-4
- Modulație de amplitudine a pulsului pe patru niveluri
- FEC obligatoriu (Forward Error Correction)
- Mod FLIT: pachete de dimensiune fixă de 256 octeți
- Îmbunătățiri ale sub-stărilor L1
- Lățime de bandă ×16: ~128 GB/s per direcție

- PAM4 folosește patru niveluri de amplitudine și transportă 2 biți per simbol; în PCIe 6.0 permite dublarea ratei de transfer la aceeași bandă, dar cu cerințe mai stricte de integritate a semnalului.
- În PCIe,TLP transportă tranzacții, DLLP gestionează linkul, iar FLIT este unitatea fixă de transport din PCIe6.0

PCIe 7.0 și direcții viitoare

- PCIe 7.0 vizează 128 GT/s per bandă (×16 = ~256 GB/s per direcție)
- Finalizarea specificației așteptată ~2025/2026
- Continuă semnalizarea PAM-4 cu FEC îmbunătățit
- Provocări noi: pierderi pe canal, diafonie (cross-talk), consum de energie
- Interconexiunile optice în curs de investigare pentru generațiile viitoare
- CXL (Compute Express Link) partajează PHY PCIe și evoluează în paralel
- PCIe rămâne interconexiunea dominantă CPU-către-dispozitiv în viitorul previzibil

Tehnologii conexe: CXL, CCIX, Gen-Z

- CXL (Compute Express Link): interconexiune coerentă cache construită pe PHY PCIe
- CXL.io (protocol I/O ≈ PCIe), CXL.cache (coerență dispozitiv-gazdă), CXL.mem (extensie memorie)
- CXL 1.0/2.0 pe PHY PCIe 5.0; CXL 3.0 pe PHY PCIe 6.0
- CCIX (Cache Coherent Interconnect for Accelerators): competitor CXL, acum în declin
- NVLink (NVIDIA): interconexiune proprietară de bandă largă GPU-la-GPU
- UALink (Ultra Accelerator Link): standard deschis emergent pentru acceleratoare AI
- Gen-Z: protocol de memorie atașată la fabric (integrat în foaia de drum CXL)
- Toate au ca obiectiv interconectarea coerentă de bandă largă a dispozitivelor

Aplicații PCIe

Grafică (GPU)
Sloturi ×16 de bandă largă pentru GPU-uri NVIDIA/AMD/Intel

Stocare (NVMe)
SSD-uri M.2, U.2, EDSFF – până la 14 GB/s (Gen 5 ×4)

Rețelistică
NIC-uri 100/200/400 GbE, HCA-uri InfiniBand

Acceleratoare AI/HPC
TPU-uri, FPGA-uri, SmartNIC-uri – PCIe Gen 5/6

Cloud/Centre de date
Virtualizare SR-IOV, stocare dezagregată

Embedded și Industrial
Mini PCIe, M.2 pentru gateway-uri IoT, routere

Beneficiile PCIe

- **Performanță ridicată -** se referă în special la lățimea de bandă, care este mai mult decât dublul față de PCI într-o legătură x1 și crește liniar pe măsură ce se adaugă mai multe lane-uri. Un beneficiu suplimentar care nu este imediat evident este faptul că această lățime de bandă este disponibilă simultan în ambele direcții de pe fiecare legătură. În plus, viteza de semnalizare inițială de 2,5 Gb/s este de așteptat să crească, obținând îmbunătățiri suplimentare ale vitezei.
- **Simplificarea I/O** - se referă la eficientizarea multitudinii de bus-uri atât de la IC-IC, cât și a utilizatorilor interni, precum AGP, PCI-X și HubLink. Această caracteristică reduce complexitatea proiectării și costul implementării.
- **Arhitectură stratificată** - PCIe stabilește o arhitectură care se poate adapta la noile tehnologii, păstrând în același timp investițiile în software. Două domenii cheie care beneficiază de arhitecturile stratificate sunt: stratul fizic, cu rate de semnalizare crescute și compatibilitate software.
- **Generațiile următoare de I/O** - PCI Express oferă noi capabilități pentru achiziția de date și multimedia prin transferuri de date izocrone. Transferurile izocrone oferă un tip de garanție a calității serviciilor (QoS) care asigură furnizarea datelor la timp prin metode deterministe, dependente de timp.
- **Ușor de utilizat** - PCI Express va simplifica foarte mult modul în care utilizatorii adaugă și modernizează sistemele. PCI Express oferă atât hot-swap cât și hot-plug. Deoarece funcția hot-plug se bazează pe anumite funcții ale SO, acesta poate rămâne la lansarea hardware-ului. În plus, varietatea de formate pentru dispozitivele PCI Express, în special SIOM(® Supermicro I/O Module) și ExpressCard, crește mult capacitatea de a adăuga periferice de înaltă performanță în servere și notebook-uri.

Referințe și lectură suplimentară

[1] PCI Express Base Specification, Rev. 6.0 – PCI-SIG, 2022

[2] PCI Express Card Electromechanical Specification (CEM), Rev. 5.0

[3] M. Jackson & R. Budruk – „PCI Express Technology 3.0" (MindShare, 2012)

[4] R. Budruk, D. Anderson, T. Shanley – „PCI Express System Architecture" (Addison-Wesley)

[5] pcisig.com – Depozitul oficial de specificații PCI-SIG și programe de conformitate

[6] NVM Express Specification 2.0 – nvmexpress.org

[7] CXL Specification, Rev. 3.0 – computeexpresslink.org, 2022

[8] Intel PCIe Architecture References – developer.intel.com

[9] Documentația subsistemului PCIe din kernelul Linux – kernel.org/doc/

[10] D. Wilen, A. Schade, R. Thornburg – „Introduction to PCI Express" (Intel Press)