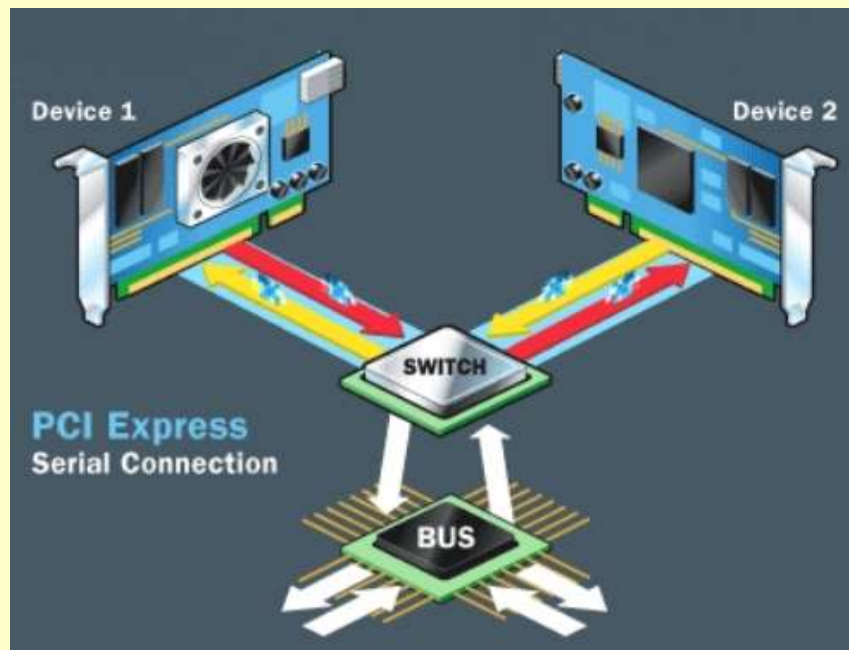


# C13. PCIe overview

- PCIe - caracteristici
- PCIe - arhitectura
  - PCI Express link
  - Topologia bus-ului
  - Arhitectura straturilor
  - Tranzactii
  - Interruperi
  - Perspective

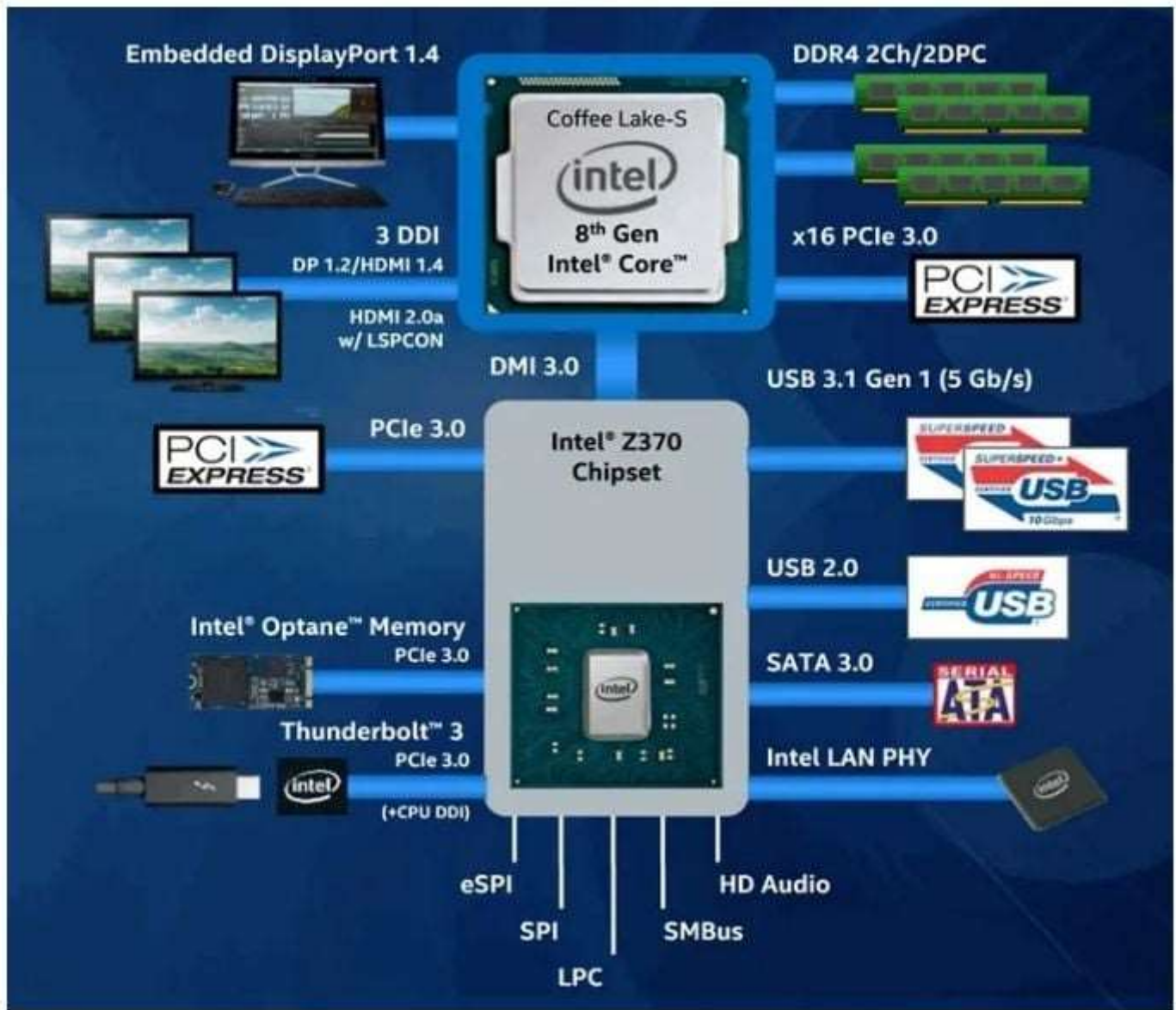


<https://www.youtube.com/watch?v=EHkuzkNWXFk>

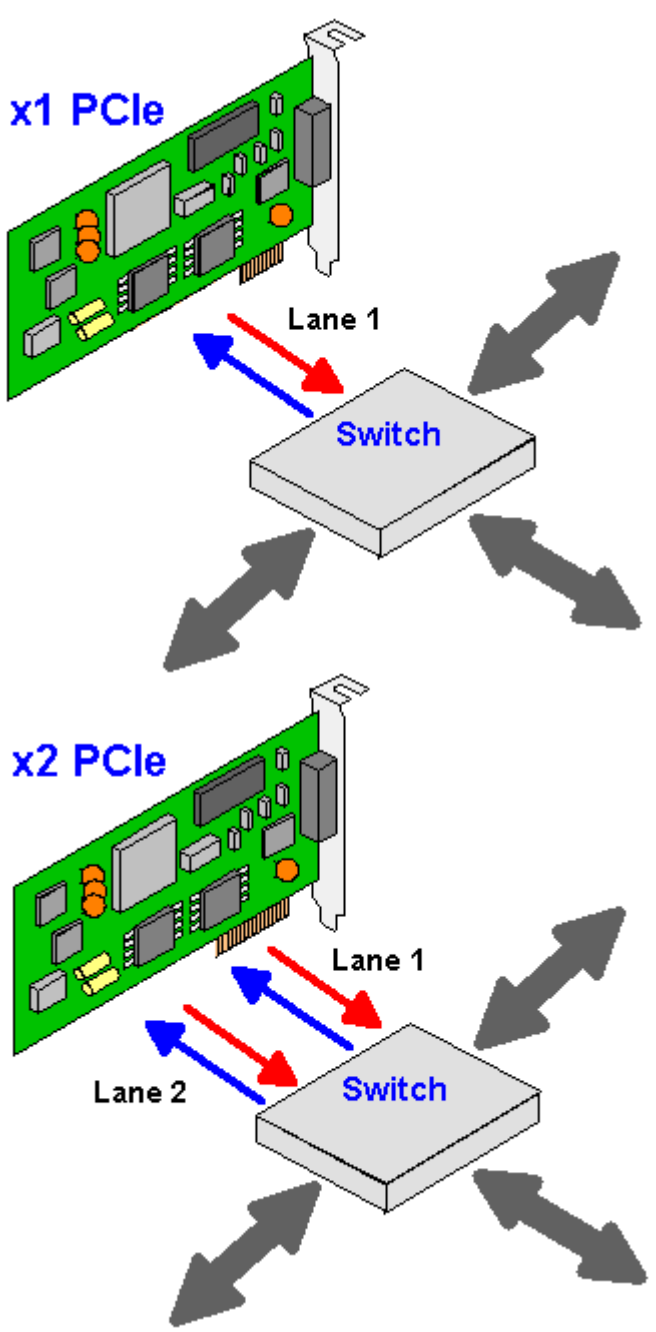
<https://www.youtube.com/watch?v=caiREMKPO-E>

<https://forum.huawei.com/enterprise/en/comparing-server-cpus-intel-vs-amd/thread/696490344071446528-667213860488228864>

- *“Industria electronică a adoptat PCIe ca fiind standardul dominant, cea mai bună interconectare de mare viteză pentru comunicațiile între chipuri și între plăci în cadrul unui sistem. Pur și simplu, nu există o altă alegere viabilă.”*
- 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 amestec de sloturi PCI și PCIe sau numai PCIe.
- *PCIe seamănă mai mult cu o rețea*, fiecare placă fiind conectată la un comutator de rețea printr-un set de fire dedicate.

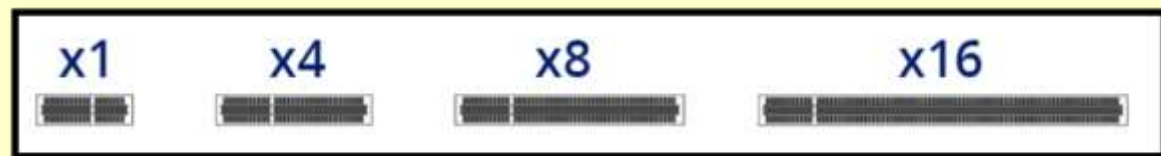


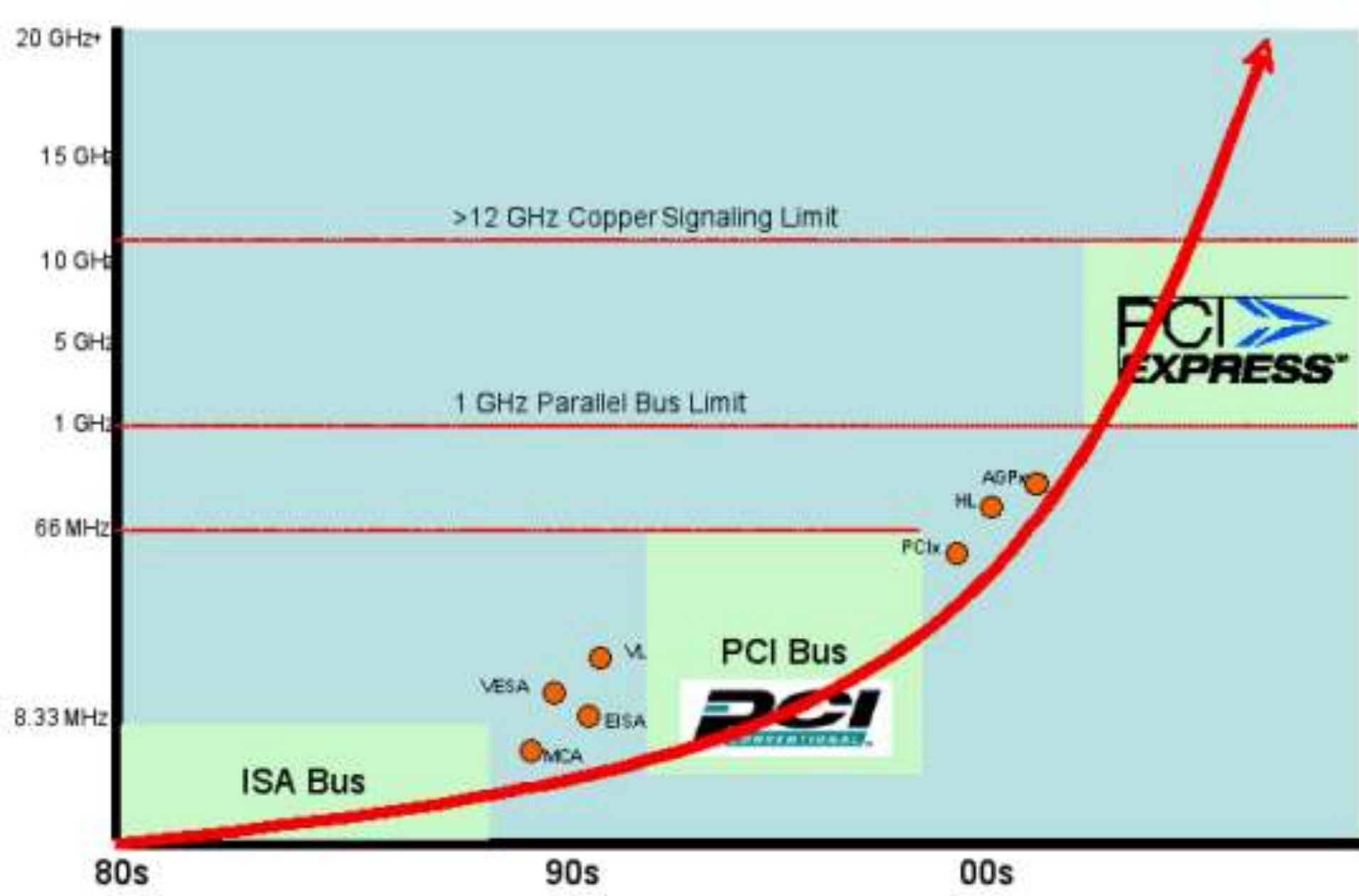
The blue lane highlights the PCIe bus



**PCIe** este o arhitectură comutată - mai multe conexiuni, mai degrabă decât structura de bus partajată ca PCI

- PCIe oferă o arhitectură comutată a canalelor care pot fi combinate în configurații x1 x2, x4, x8, x16, creând o interfață paralelă de „canale (lanes)” controlate independent.
- Placa de bază a comutatorului determină lățimea de bandă totală, iar cardurile și plăcile de bază sunt compatibile între versiuni.
- Ptr. comparatia tuturor tehnologiilor PCI, vezi [PCI-SIG](https://www.pci-sig.com/).
- Vezi [PCI](#), [ExpressCard](#), [PCI-X](#), [SATAexpress](#) and [Thunderbolt](#).





"Industria electronică a adoptat PCIe ca fiind standardul care domina, cea mai bună interconectare de mare viteză pentru comunicațiile între cipuri și între plăci în cadrul unui sistem. Pur și simplu nu există altă alegere viabilă."<sup>5</sup>

# Characteristicile PCI Express

- Point-to-point connection
- Serial bus means fewer pins
- Scalable: x1, x2, x4, x8, x12, x16, x32
- Dual Simplex connection
- 2.5 GT/s transfer/direction/s
- Packet based transaction protocol

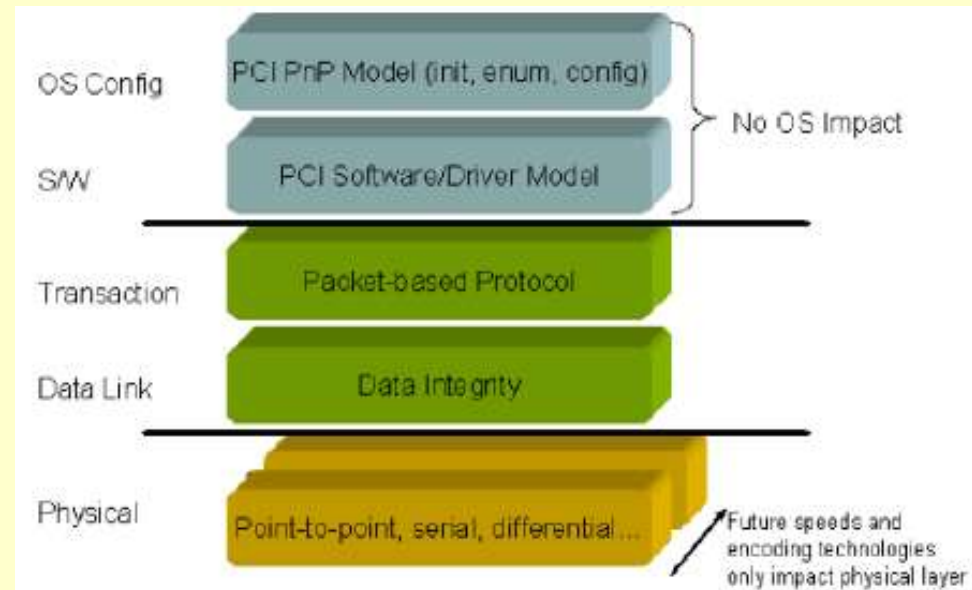


Figure 2. PCI Express Layered Architecture

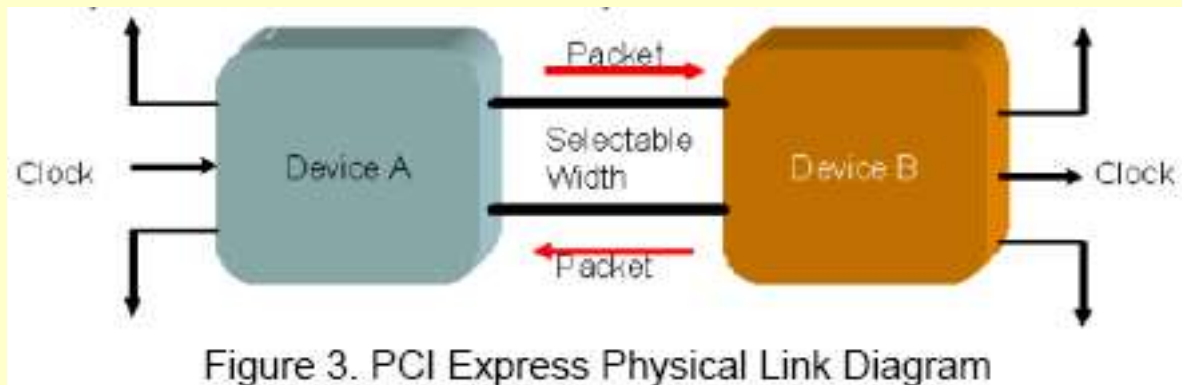


Figure 3. PCI Express Physical Link Diagram

# Caracteristicile bus-ului PCI express

- *PCI Express este un bus I/O de înaltă performanță pentru periferice, folosit pe platformele de calcul.*
- *PCIe a evoluat din arhitecturile PCI și PCI-X și utilizează același model de comunicație ca și PCI și PCI-X.*
- ***PCI și PCI-X sunt bus-uri paralele, în timp ce PCIe utilizează o interconectare serială punct-la-punct, pentru comunicarea între două dispozitive periferice.***
- *Această interfață serială permite reducerea numărului de pini și simplifică interconectările*
- *Acesta unifică arhitectura I/O pentru diferite tipuri de sisteme și sisteme embedded*
- *PCIe utilizează aceleași spații de adrese: memorie, I/O și configurație ca și PCI*
- ***PCIe implementează un protocol bazat pe pachete** pentru transferul de informații*
- *Bus-ul are performanțe scalabile bazate pe numărul de linii de semnal (lane)\* implementate pe conexiunea PCI Express (dual simplex)*
- *Compatibilitatea este menținută cu SO existente și cu driverele software care nu necesită modificări.*

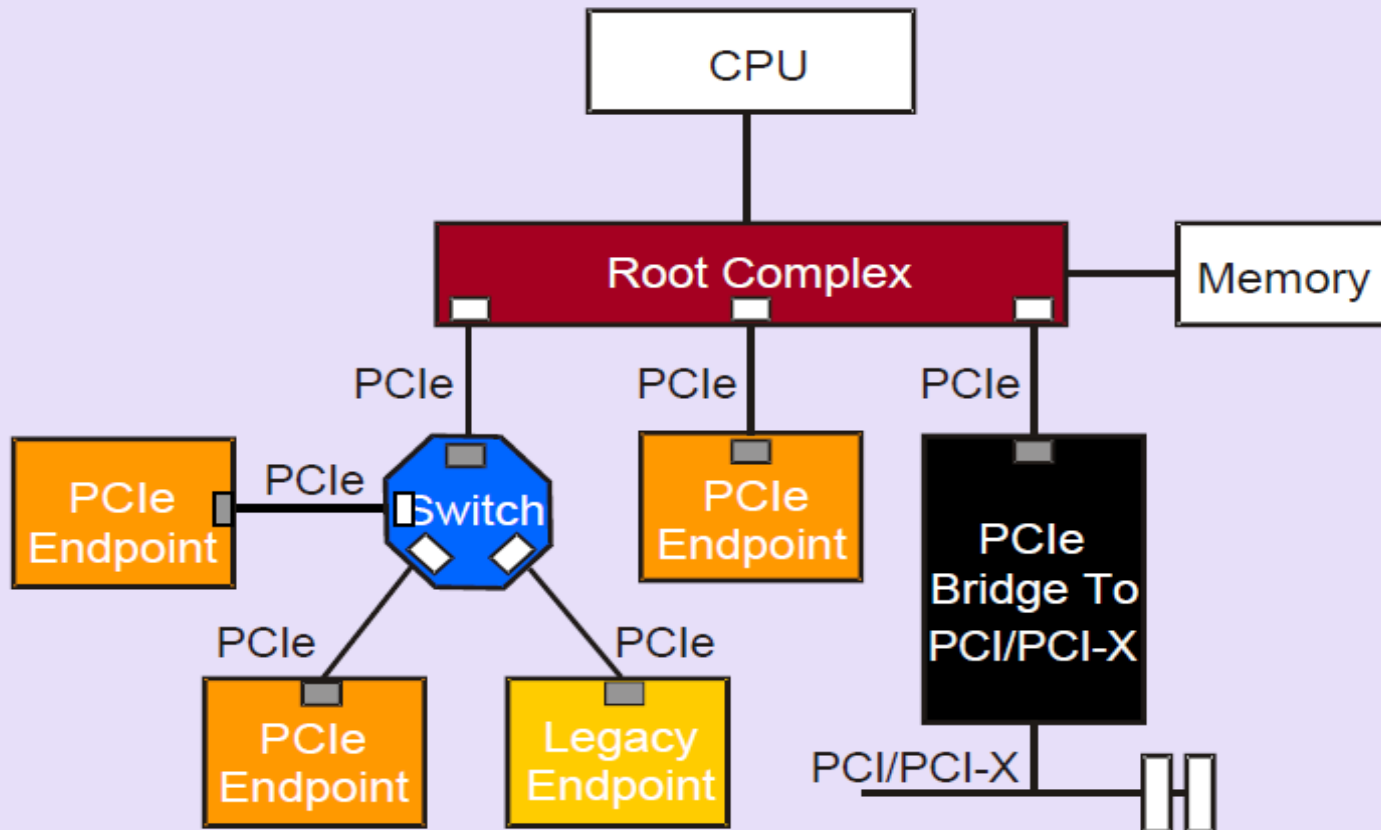
*\* Perechea de transmisie și recepție împreună se numește lane. Viteza inițială de 2,5 Gb/s oferă o lățime de bandă nominală de aproximativ 250 MB/s în fiecare direcție pe fiecare lane a PCI Express.*

## PCI Express caracteristici (cont.)

- Busul PCIe permite aceleași tipuri de tranzacții ca și bus-urile anterioare: R/W memorie, R/W I/O și R/W configurație etc.
- PCIe permite interconectarea CI pe placa de bază și *cartelele de extensie* prin conectori sau cabluri
- Oferă o calitate diferențiată a serviciului (QoS) prin:
  - capacitatea de a aloca resurse dedicate pentru anumite fluxuri de date
  - de a configura politicile de arbitraj QoS pentru fiecare componenta
  - de a realiza transferuri izocrone pentru aplicații în timp real
- Oferă o *gestionare avansată a alimentării* prin capacitatea de a identifica posibilitățile de control a alimentării a fiecărui dispozitiv periferic
- Acceptă *raportarea și manipularea avansată a erorilor*, pentru a îmbunătăți izolarea defecțiunilor și recuperarea erorilor
- Acceptă *hot-plugging & hot-swapping* a dispozitivelor periferice

# Topologia PCI Express

- Sistemul PCIe este compus din *legături PCIe* care interconectează un *set de componente*
- Un exemplu de topologie mentonata ca o ierarhie este compusa din:
  - un Root-Complex ,
  - m.m. Endpoints (I/O devices),
  - un Switch
  - o punte PCI Express la PCI/PCI-X, *toate interconectate prin legaturi PCIe*

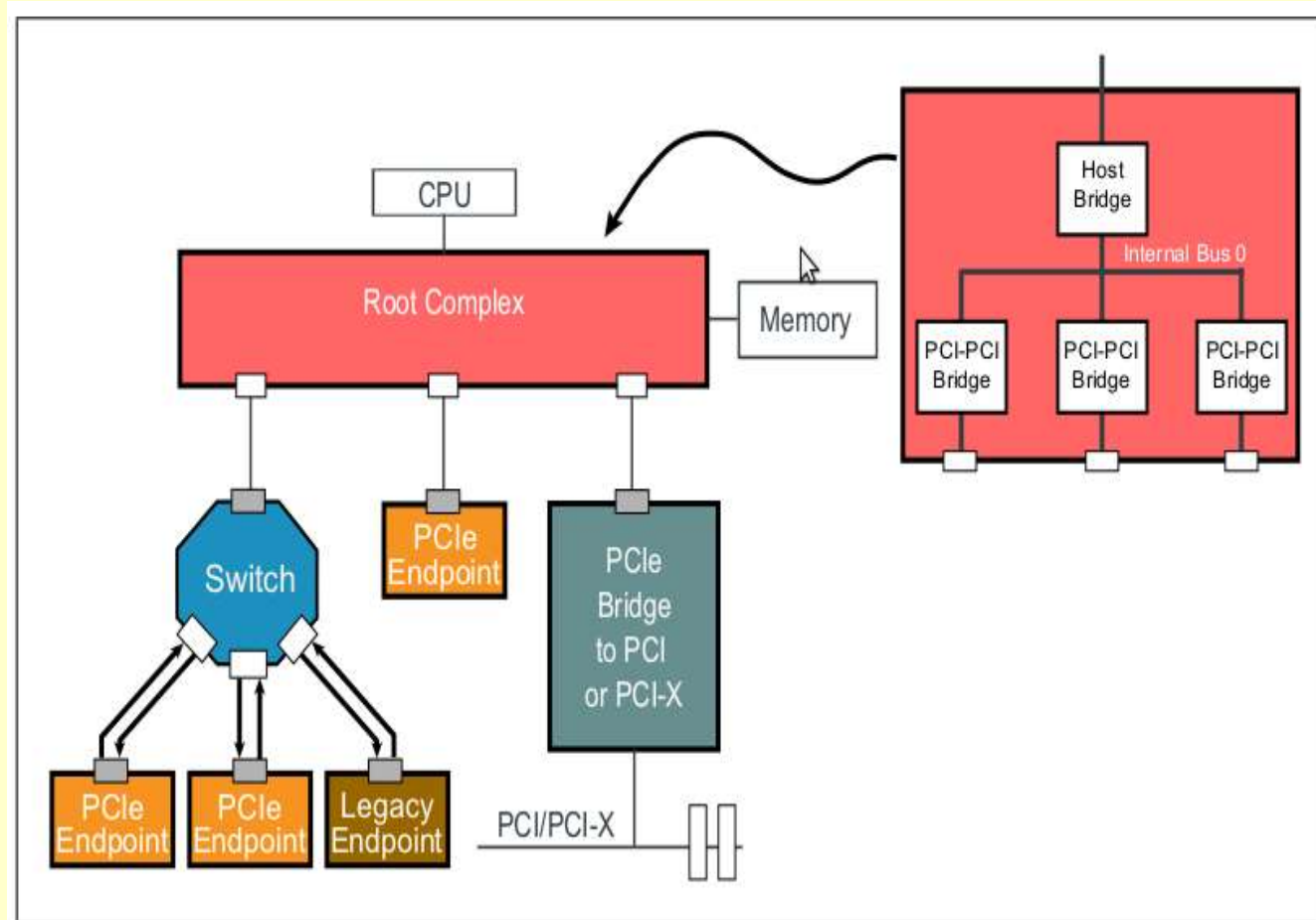




# Root Complex (RC)

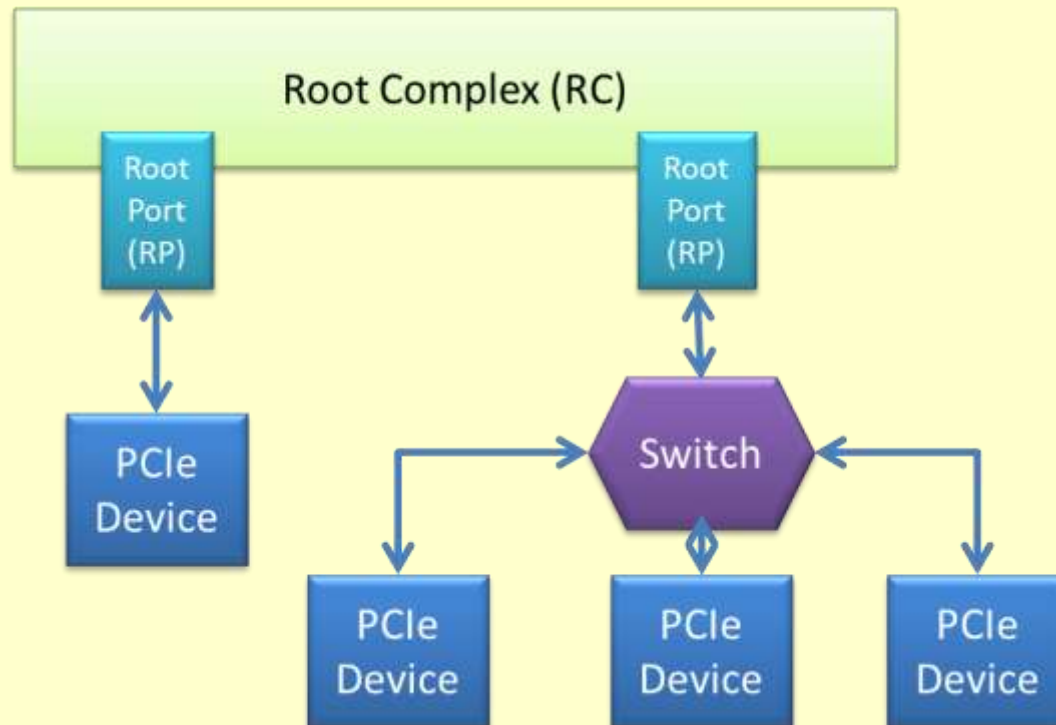
- **Root Complex (RC)** – este un device care *conecteaza unul sau mai multe procesoare si un subsistem de memorie la dispozitivele de I/O.*
- RC device reprezinta radacina unei ierarhii I/O
- Similar ca la *host bridge* dintr-un sistem PCI : - RC genereaza cereri de tranzactii din partea procesorului, cu care este interconectat printr-un *local bus.*
- RC poate avea unul sau mai multe porturi PCI Express – Root-Ports.

- RC implementeaza diferite resurse, ca: interrupt controller, power management controller, error detection
- RC contine un bus intern, care reprezinta bus-ul 0 in intreaga ierarhie
- RC reuneste ierarhiile domeniilor PCIe intr-o singura ierarhie PCIe



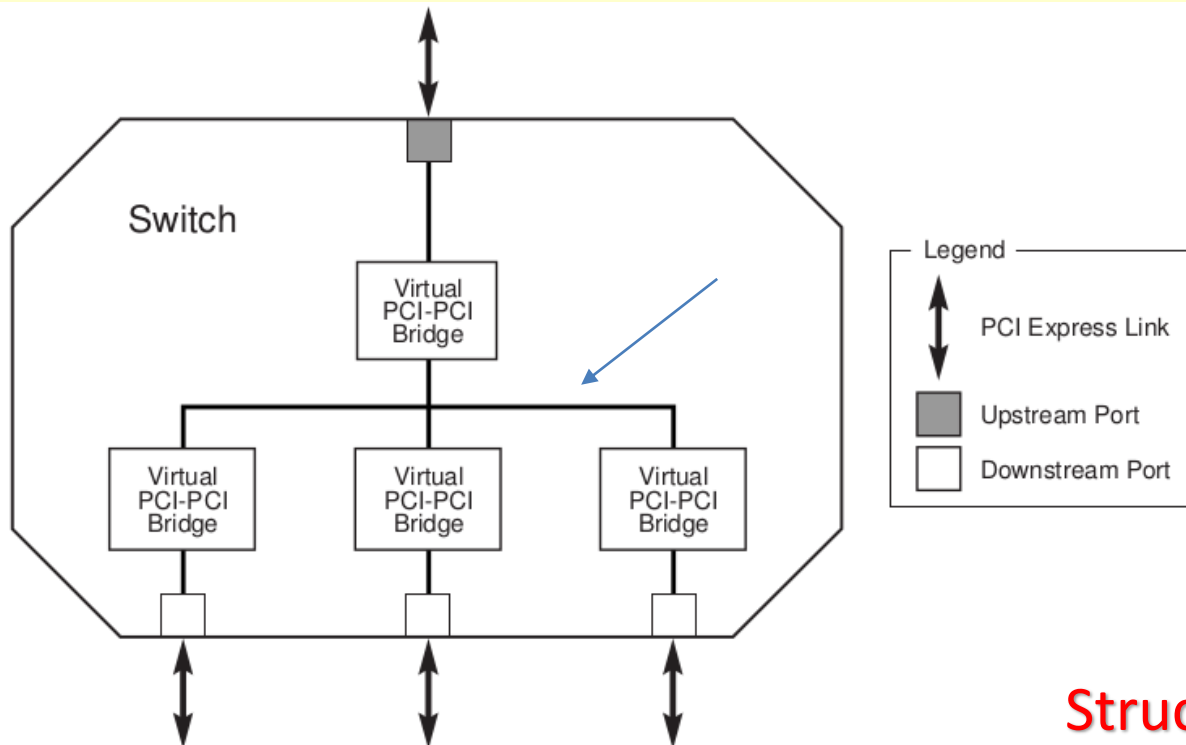
# PCIe Root Ports/RP

- Partea din placa de baza care *contine root complex* si permite porturilor (Root Ports) PCIe sa discute cu restul dispozitivelor computerului
- Fiecare Root Port defineste un domeniu de ierarhie separata
- Fiecare domeniu de ierarhie este compus dintr-un singur *endpoint sau o sub-ierarhie* continand unul sau m.m. switch-uri si endpoint-uri
- Posibilitatea de a ruta tranzactii peer-to-peer intre domeniile ierarhiei prin Root Complex este **optional** si dependent de implementare



# Comutatoare și punți (Switches and Bridges)

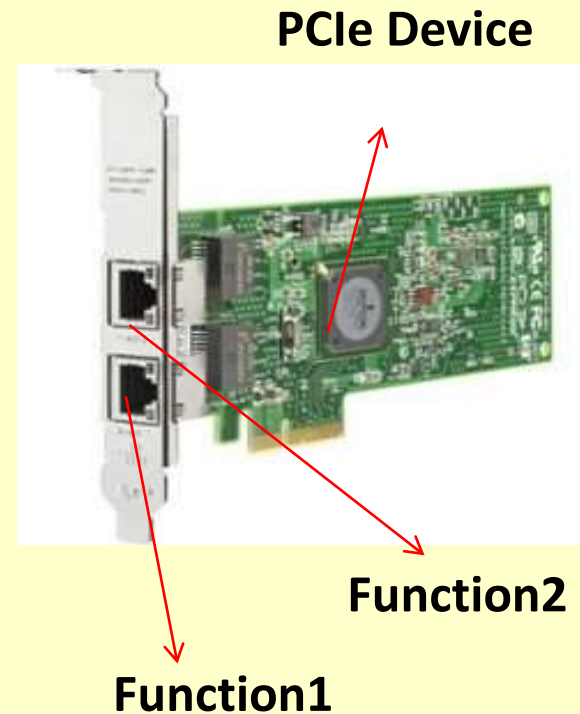
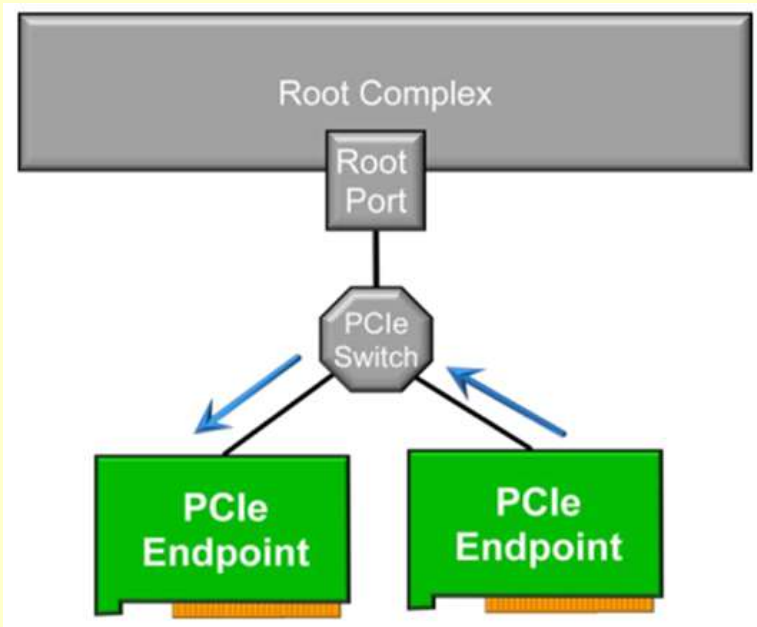
- **Un switch este** definit ca un *ansamblu de 2 sau m.m. punți logice PCI-to-PCI*, fiecare punte e asociata cu un switch-port; punțile sunt conectate via un bus intern
- **Switch-urile** ofera posibilitatea ca mai multe dispozitive sa fie atasate la un singur RP
- Switch-urile actioneaza ca *packet routers si* recunosc care cale trebuie sa o urmeze un anumit pachet pe baza adresei sale si a altor informatii de rutare
- Un Switch poate avea cateva Downstream Ports, dar numai un singur Upstream Port
- **Punțile** ofera interfatarea PCIe la alte bus-uri, ca PCI/PCI X, USB, InfiniBand, Ethernet, Fiber channel sau chiar la un alt PCI-e bus



Structura interna a unui switch

# PCIe Endpoint

- **Endpoints** reprezinta un *dispozitiv periferic* care participa la tranzactiile PCIe
- Un *initiator (requester)* endpoint initiaza o tranzactie in sistemul PCIe, in timp ce *target-ul (completer)* endpoint adresat raspunde la tranzactie
- *Switches/bridges NU pot actiona ca initiatori sau tinta* a tranzactiilor de pe bus
- *Endpoint-urile* rezida la capatul ramurilor arborelui topologic si implementeaza numai **un singur upstream-port** (indreptat spre Root Complex).

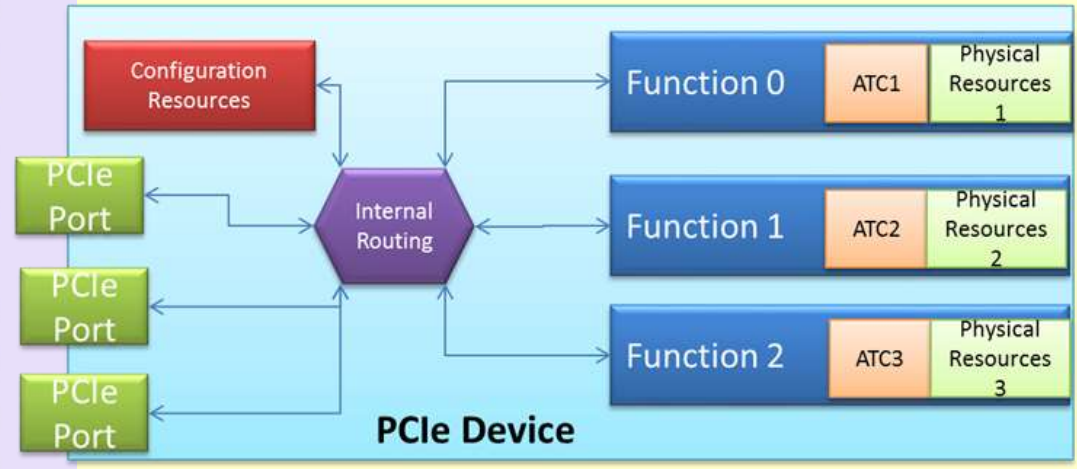


- Dispozitivele PCIe pot avea pana la 8 functii logice si *fiecarui endpoint* ii este asignat un *device identifier (ID)*, care consta dintr-un *bus number, device number si function number*.
- Legatura si functionalitatea PCIe este partajata de toate functiile si este administrata prin *Function 0*
- Toate functiile folosesc un *singur Bus Number* capturat prin procesul de enumerare

Byte				Doubleword Number (in decimal)
3	2	1	0	
Device ID		Vendor ID		00
Status Register		Command Register		01
Class Code			Revision ID	02
BIST	Header Type	Latency Timer	Cache Line Size	03
Base Address 0				04
Base Address 1				05
Base Address 2				06
Base Address 3				07
Base Address 4				08
Base Address 5				09
CardBus CIS Pointer				10
Subsystem ID		Subsystem Vendor ID		11
Expansion ROM Base Address				12
Reserved			Capabilities Pointer	13
Reserved				14
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	15

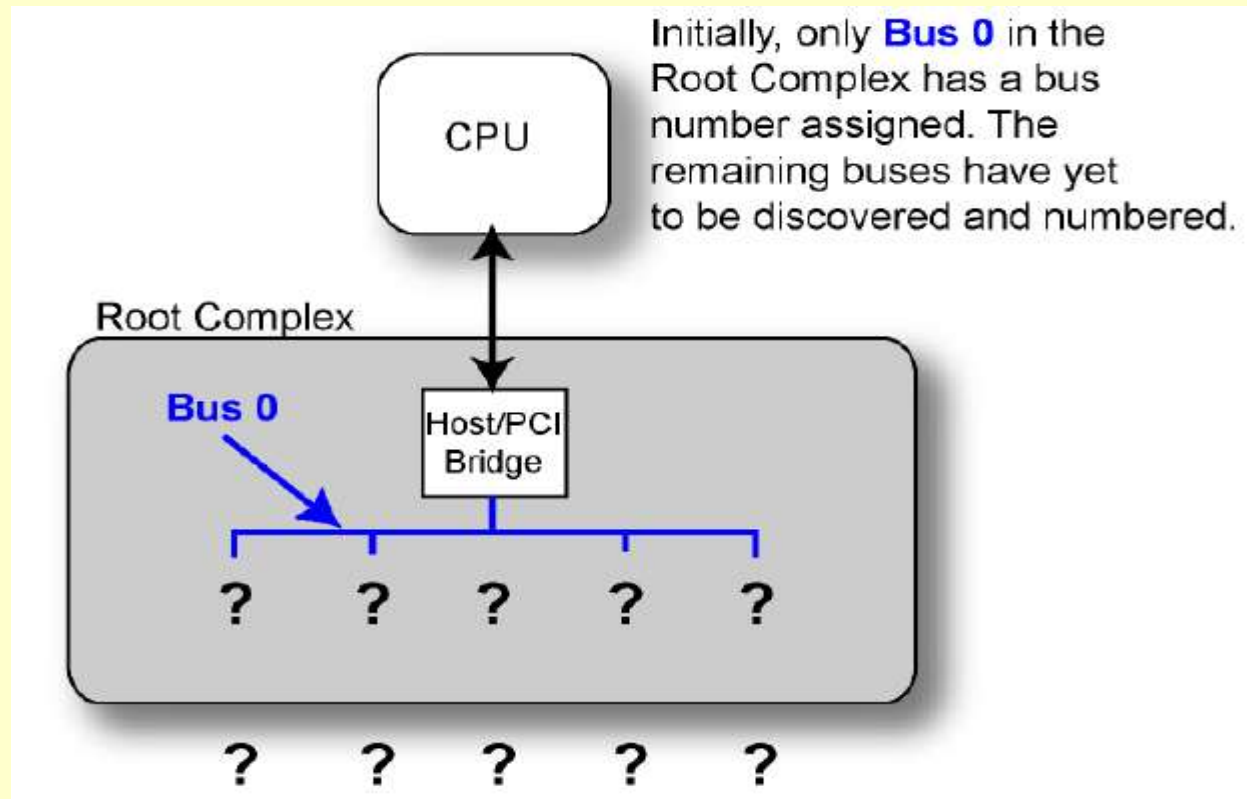
### Configuration Space

- Dispozitivele vor alocă resurse, precum memoria, I/O, și vor înregistra adresa în acest spațiu de configurare



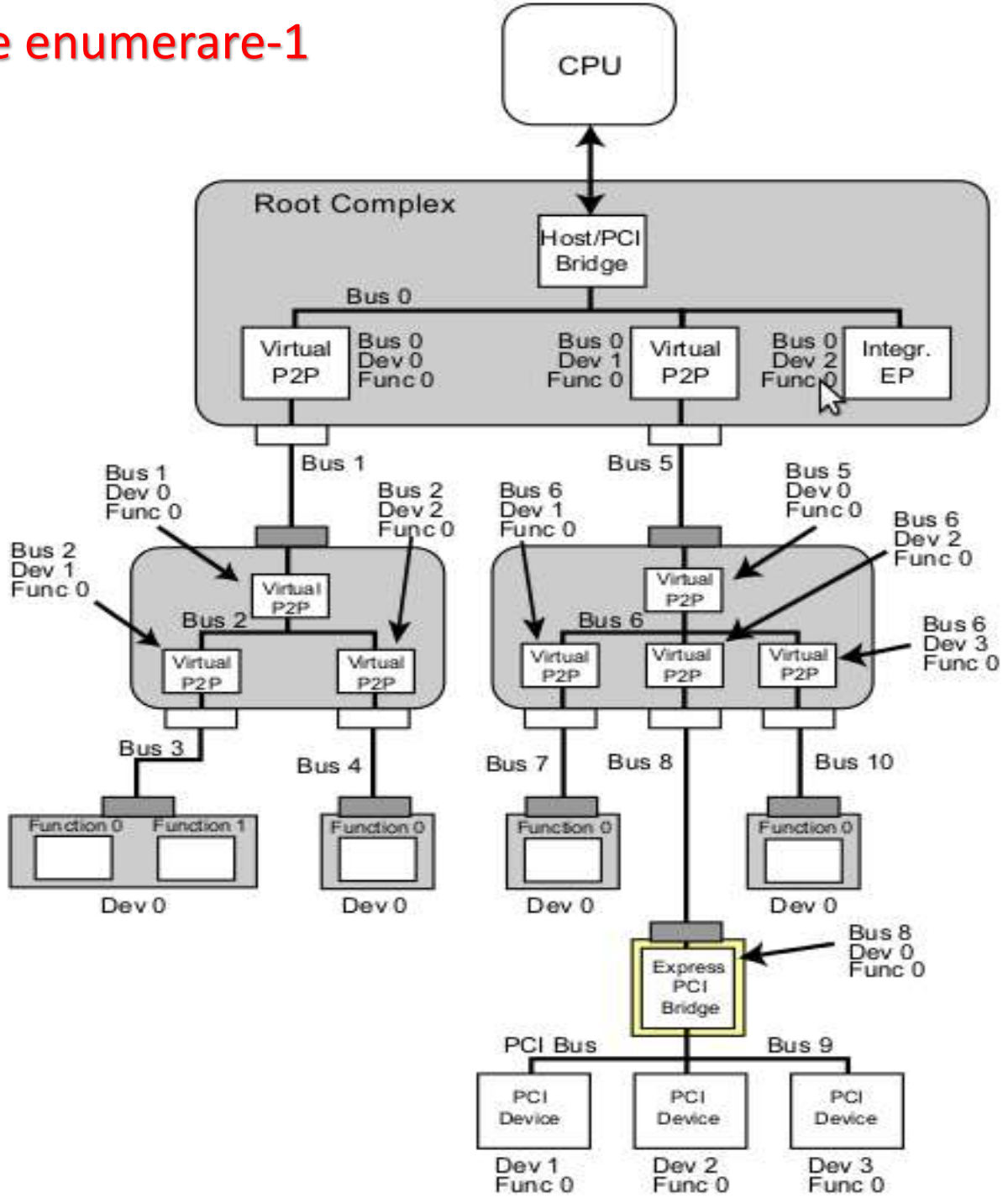
# Enumerarea

- Procesul prin care *software-ul de configurare descopera topologia sistemului si asigneaza numarul bus-ului si resursele sistem*
- *Root Complex/ Host* trimite *Configuration Packets* ptr. a asigna numere unice ptr. *Bus, Device si Function* la EndPoint-urile conectate
- Enumerarea ierarhiei PCIe (x86) este facuta de *BIOS in faza de initializare a hardware-ului* – toate registrele sunt configurate inainte, de bootload-are.

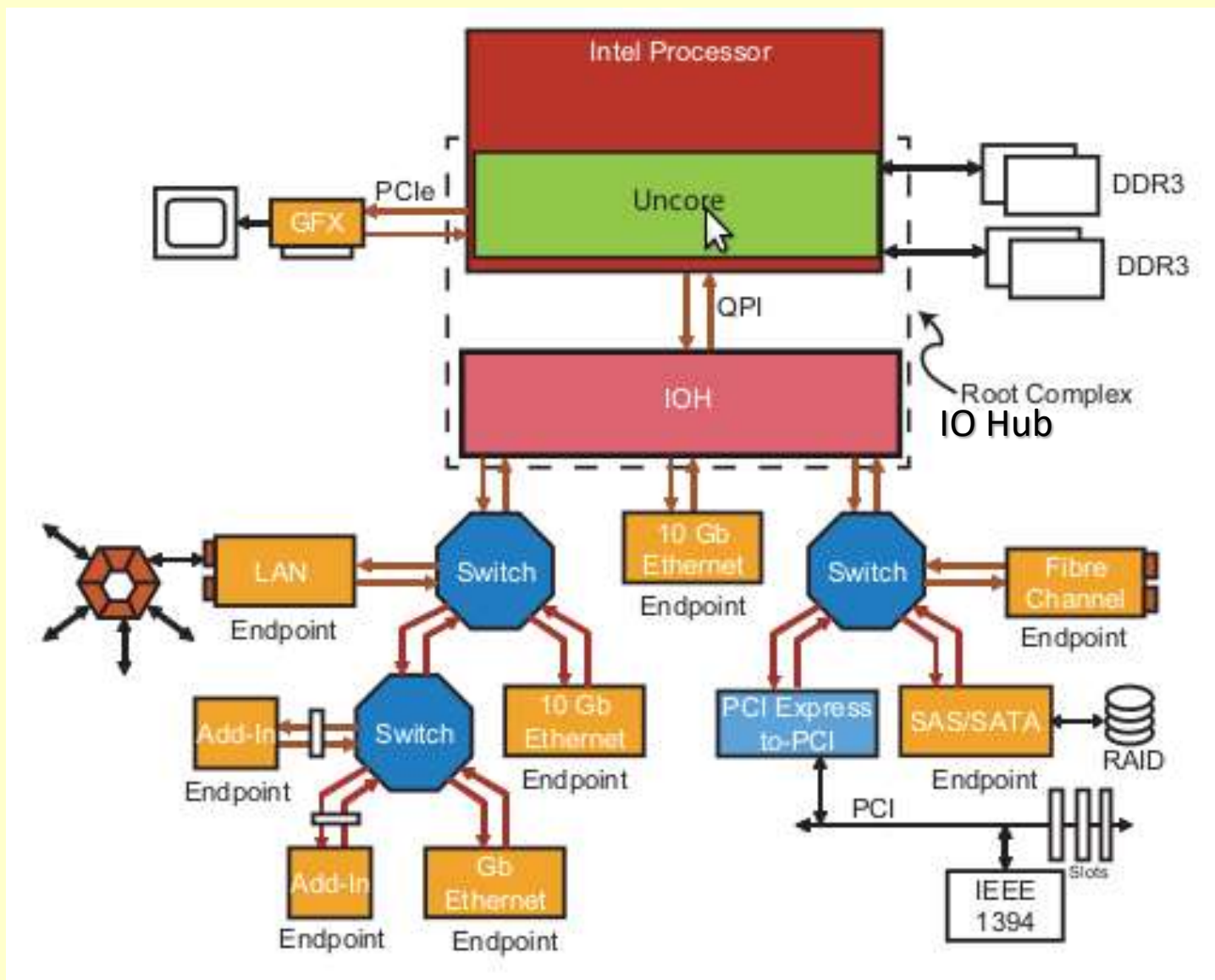


Vizualizarea topologiei la pornire

# Exemplu de enumerare-1

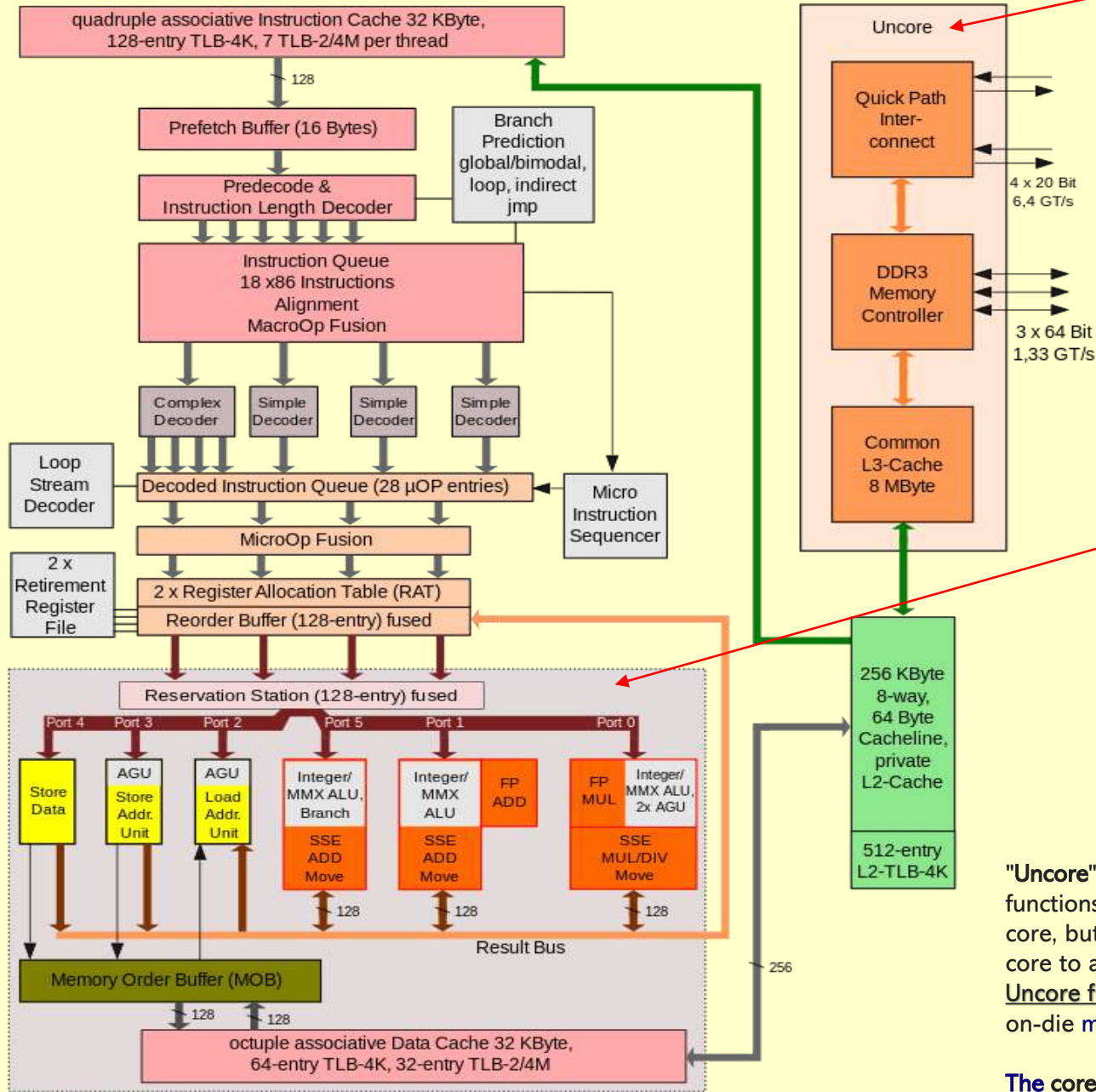


# Exemplu de sistem



Intel Quick Path Interconnect (QPI) este o interconectare punct-la-punct a procesorului, dezvoltată de Intel, care a înlocuit front-side bus (FSB) în Xeon, Itanium și în anumite platforme desktop și a crescut scalabilitatea și lățimea de bandă disponibilă. Înainte de anunțarea denumirii, Intel se referea la ea ca fiind *Common System Interface (CSI)*.





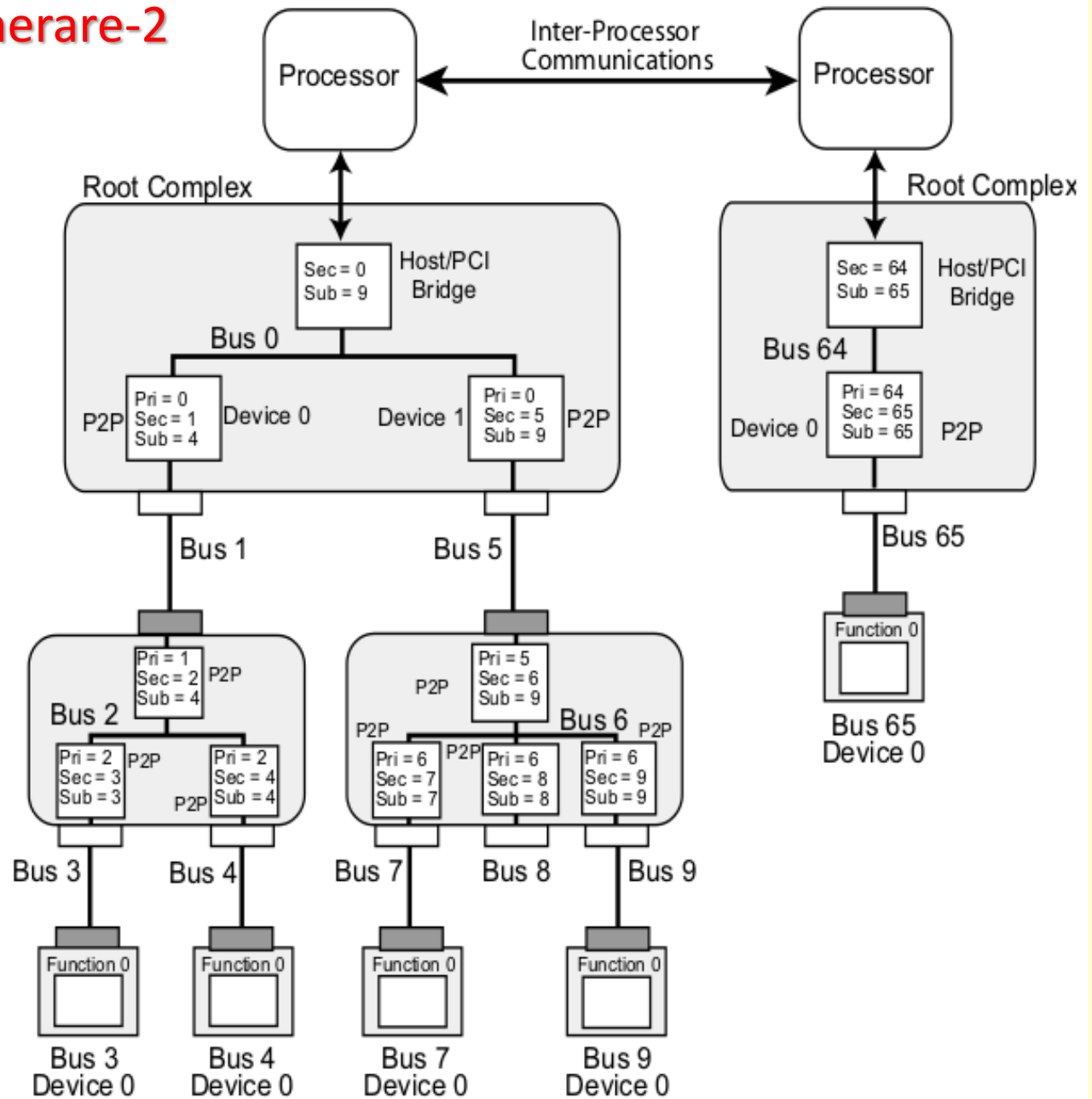
GT/s: gigatransfers per second

- "Uncore" este un termen utilizat de Intel pentru a descrie funcțiile unui microprocesor care nu se află în nucleu, dar care trebuie să fie strâns legate de acesta pentru a obține performanțe ridicate.
- Funcțiile uncore includ *controllele QPI, memoria cache L3, controlerul de memorie on-die.*
- Nucleul (core) conține componentele procesorului implicate în executarea instrucțiunilor, inclusiv ALU, FPU, cache L1 și L2.

"Uncore" is a term used by Intel to describe the functions of a **microprocessor** that are not in the core, but which must be closely connected to the core to achieve high performance. **Uncore functions** include QPI controllers, L3 cache, on-die memory controller.

The **core** contains the components of the processor involved in executing instructions, including the ALU, FPU, L1 and L2 cache.

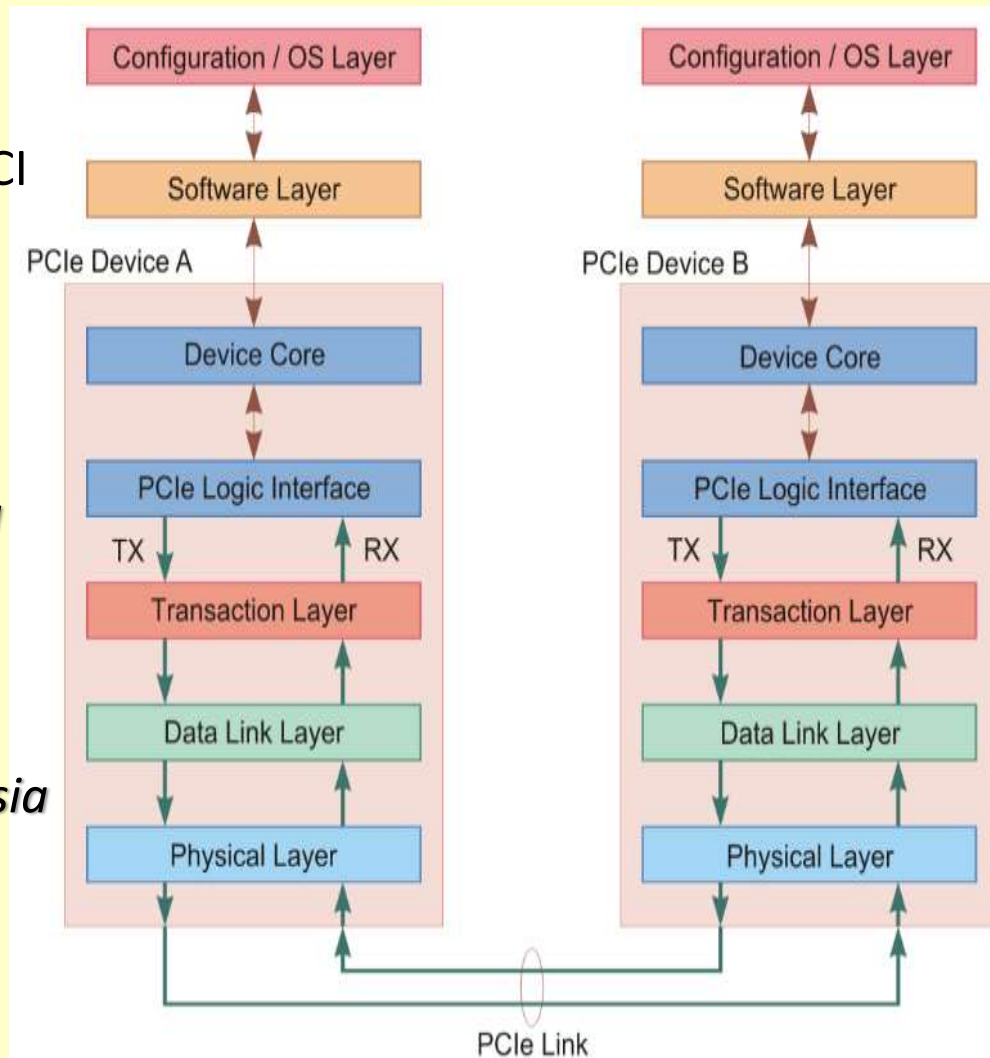
# Exemplu de enumerare-2



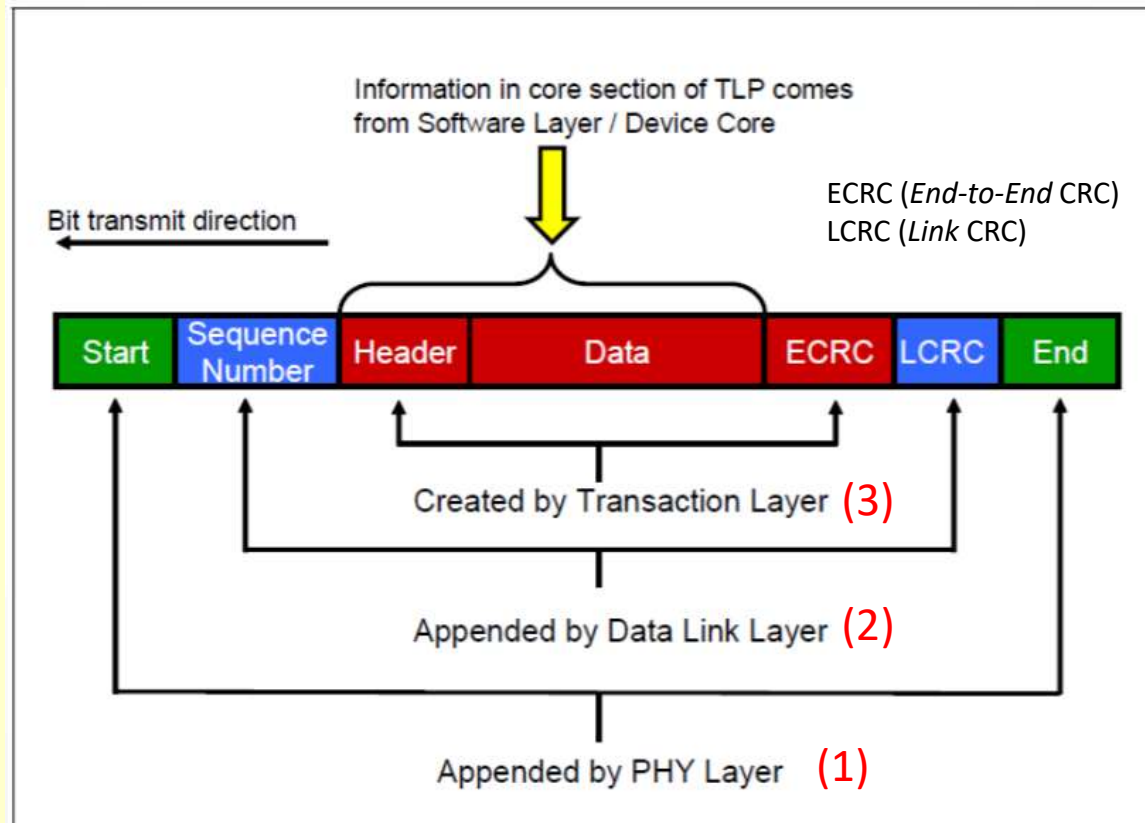
# Straturile arhitecturii PCIe

Sistemul PCIe este structurat **in 5 straturi logice**:

- **Configuration/OS layer (5)** administreaza configurarea dispozitivelor PCIe prin SO pe baza specificatiilor Plug-and-Play ptr. initializare, enumerare si configurare a dispozitivelor I/O
- **Software layer (4)** interactioneaza cu SO prin aceleasi drivere ca si bus-ul PCI conventional
- **Transaction layer (3)** administreaza transmissia si receptia informatiilor folosind packet-based protocol.
- **Data link layer (2)** asigura integritatea transferurilor de date prin detectia erorilor folosind Cyclic Redundancy Check(CRC).
- **Physical layer efectueaza (1) transmisia pachetelor pe legaturile seriale PCIe**
- Specificatiile PCIe **definesc arhitectura dispozitivelor PCIe ca avand 3 straturi logice**



- Bus-ul PCIe foloseste pachetele ptr. transferul informatiilor intre perechi de dispozitive conectate prin conexiuni PCIe seriale
- *Pachetele sunt formate in transaction layer (3)* pe baza informatiilor obtinute de la *the device core si aplicatie si stocata intr-un buffer*
- Data link layer (2) extinde pachetul cu informatii suplimentare necesare ptr. detectia erorilor la dispozitivul receptor
- Apoi pachetul este codat in the physical layer(1) (8b/10b) si transmis prin semnale diferentiale pe legatura PCIe



Trecerea pachetului prin starturile logice a dispozitivului PCIe

# Tipurile de tranzactii PCIe

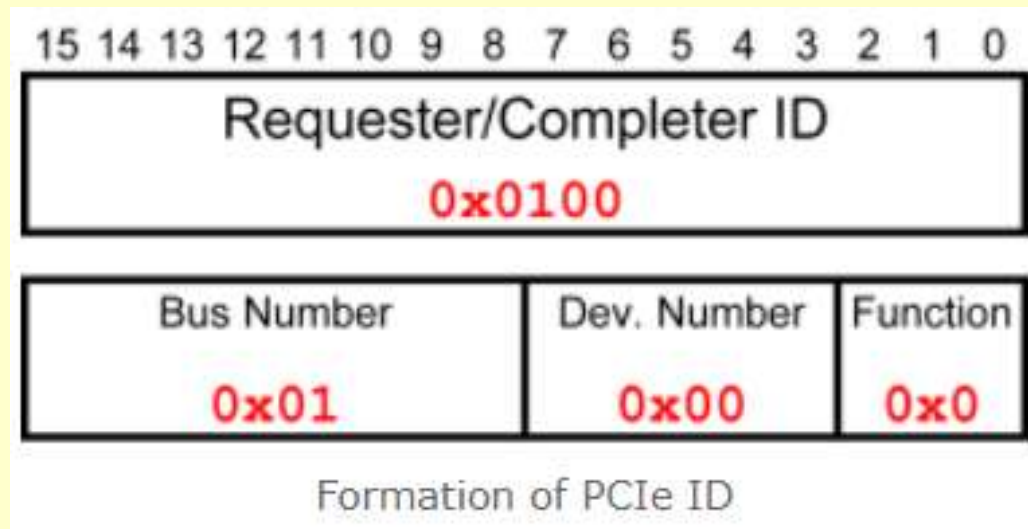
- **Tranzactia** este definita ca o *serie de una/ m.m. transmisii de pachete* necesare pentru a realiza un transfer de date între un inițiator și un dispozitiv țintă
- Solicitățile sunt traduse la *una dintre cele patru tipuri de tranzacții* de către Transaction Layer:
- **Memory R/W.** Folosit pentru a transfera date de la/către o locație mapată în memorie
- **I/O R/W.** Folosit pentru a transfera date de la/către o locație I/O
- **Configuration R/W.** Folosit pentru a descoperi capacitățile dispozitivului, caracteristicile programului și starea de verificare în spațiul de configurare de 4KB la PCI Express.
- **Messages.** Folosit pentru *semnalizarea evenimentelor și mesagerie de uz general*. *Tranzacțiile de mesaje* sunt **specifice magistralei PCIe** și sunt utilizate pentru:
  - *semnalizarea întreruperilor*
  - *gestionarea alimentării*
  - *semnalizarea erorilor.*

# Tipurile de pachete (TLP) PCI Express

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

# Identificare si rutare

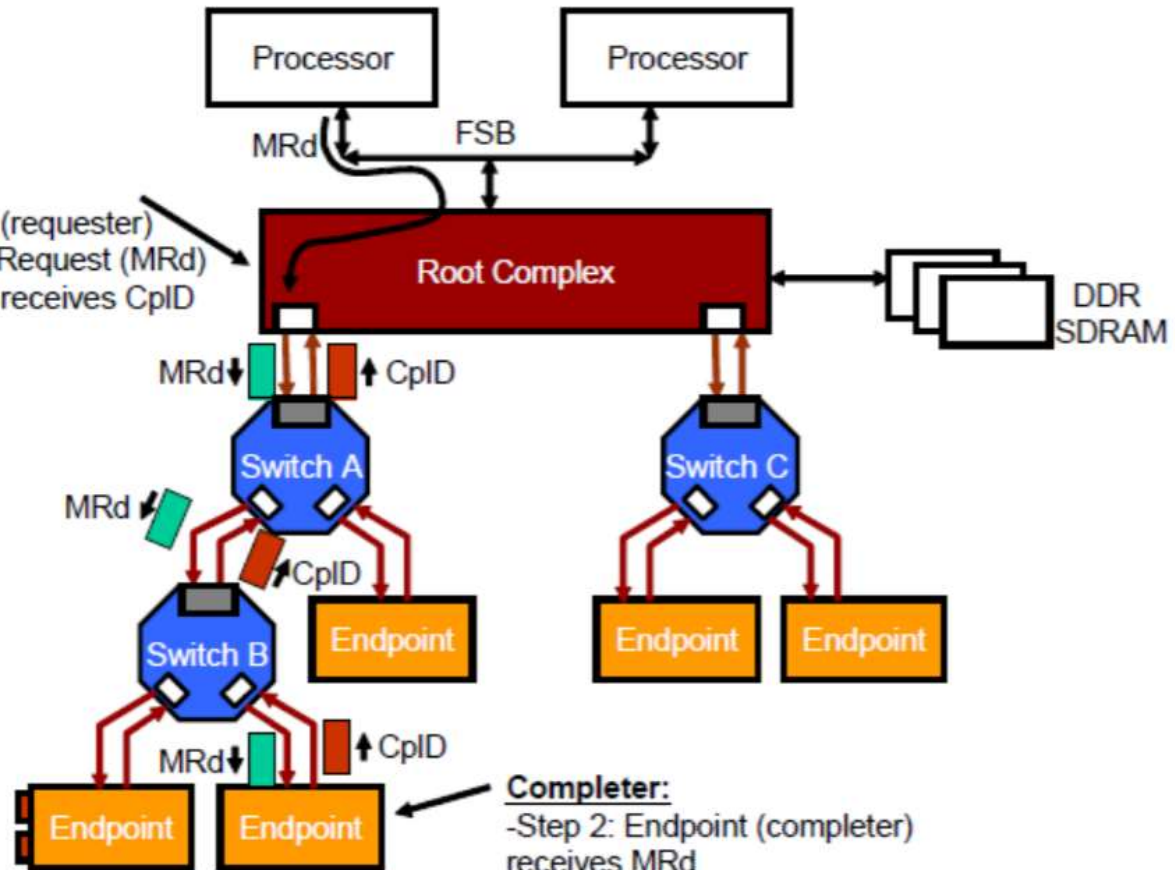
- *Deoarece PCIe este în esență o rețea de pachete, cu posibilitatea de comutare pe cale, aceste switch-uri trebuie să știe unde să trimită fiecare TLP*
- Sunt 3 metode de rutare:
  - Address routing este aplicata ptr. Memorie si I/O Requests (read si write)
  - Implicit routing este folosit numai ptr. anumite mesaje TLPs, ca broadcasts de la Root Complex si mesaje care merg intotdeauna la Root Complex.
  - ID routing – toate celelalte TLPs sunt routate de ID.
- ID este un cuvânt de 16-biti format din 3 campuri: *Bus number, Device number si Function number*. Sensul lor este exact ca în bus-urile PCI tradiționale.
- ID-ul se formează după cum urmează:



# CPU MRd targeting an Endpoint

## Requester:

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



## Completer:

- Step 2: Endpoint (completer) receives MRd
- Step 3: Endpoint returns Completion with data (CplID)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW 0	R	0	Fmt	0x0	Type	0x00	R	0	TC	0	R	0	TDEP	0	Attr	0	R	0	Length								0x001					
DW 1	Requester ID								Tag				Last BE		1st BE																	
	0x0000								0x0c				0x0		0xf																	
DW 2	Address [31:2]															R																
	0x3f6bfc10															0																

Example of Memory Read Request TLP

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW 0	R	0	Fmt	0x2	Type	0x0a	R	0	TC	0	R	0	TDEP	0	Attr	0	R	0	Length								0x001					
DW 1	Completer ID										Status		Byte Count																			
	0x0100										0x00		0x004																			
DW 2	Requester ID										Tag		Lower Address																			
	0x0000										0x0c		0x40																			
DW 3	Data DW 0																															
	0x12345678																															

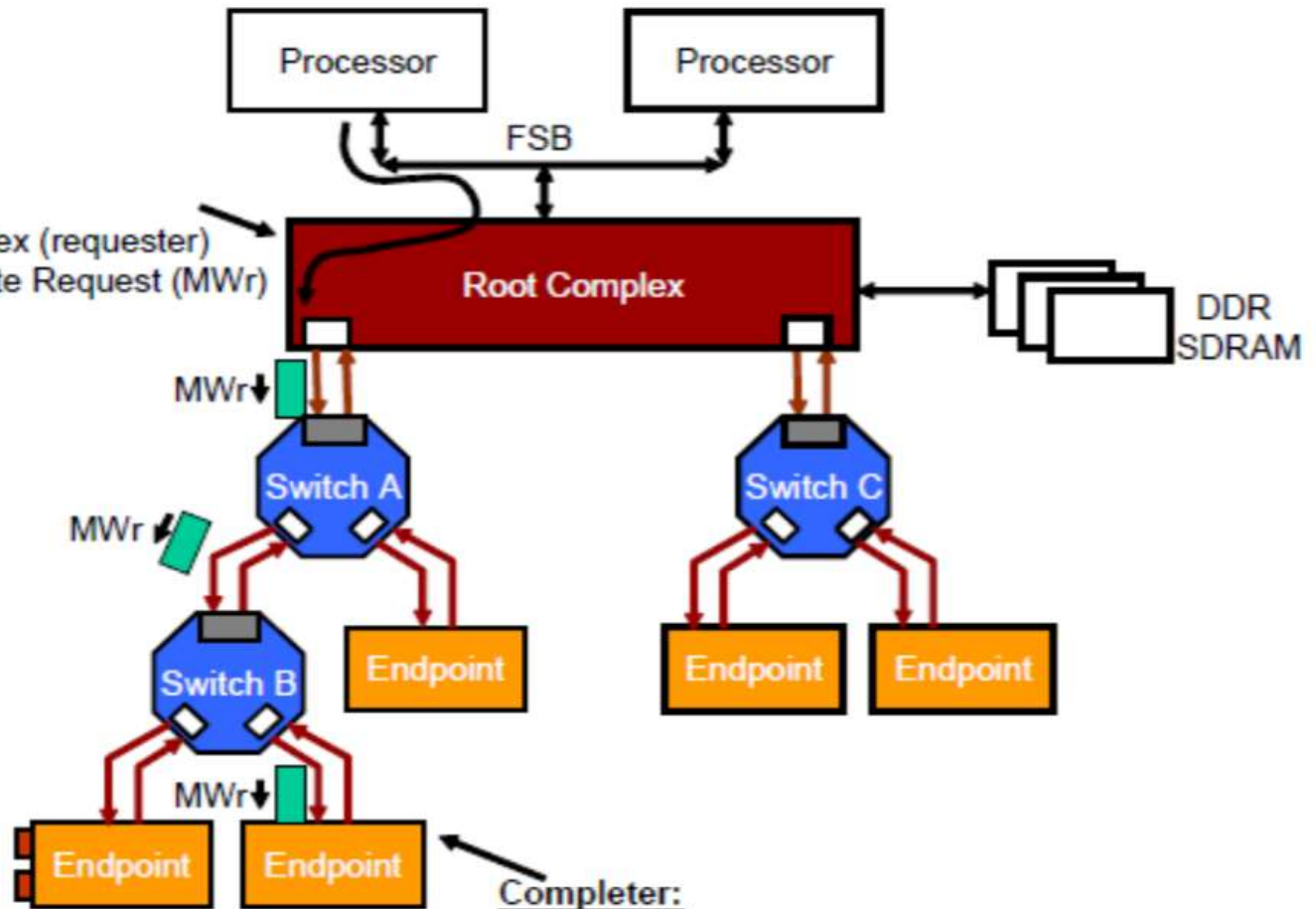
Example of Completion TLP



# CPU MWr targeting Endpoint

## Requester:

-Step 1: Root Complex (requester) initiates Memory Write Request (MWr)



## Completer:

- Step 2: Endpoint (completer) receives MWr

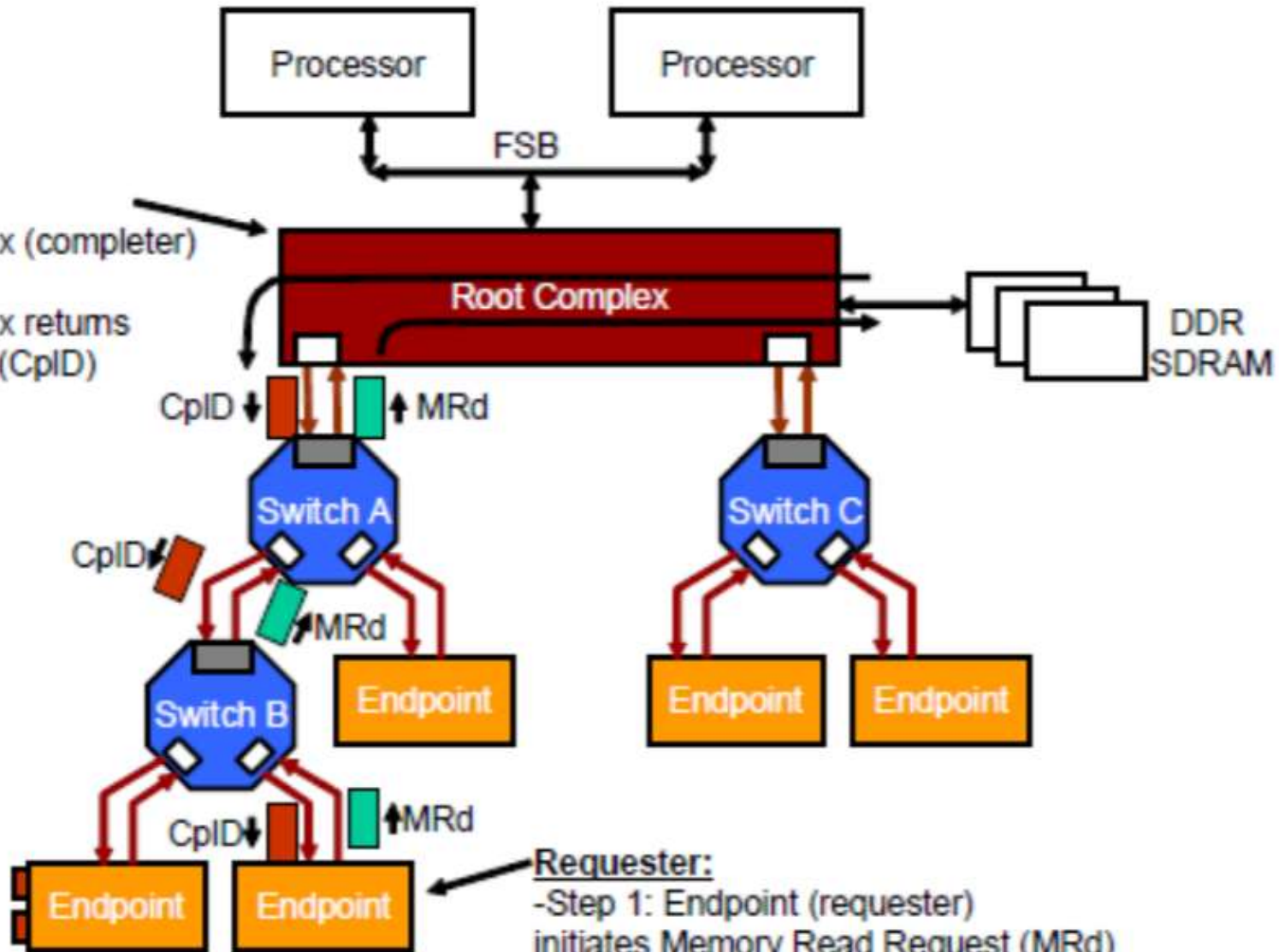
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW 0	R	0	Fmt	0x2	Type	0x00	R	0	TC	0	R	0	rdEP	0	Attr	0	R	0	Length								0x001					
DW 1	Requester ID										Tag (unused)				Last BE		1st BE															
	0x0000										0x00				0x0		0xf															
DW 2	Address [31:2]																										R	0				
	0x3f6bfc10																															
DW 3	Data DW 0																															
	0x12345678																															

Example of Memory Write Request TLP

# Endpoint MRd targeting system memory

## Completer:

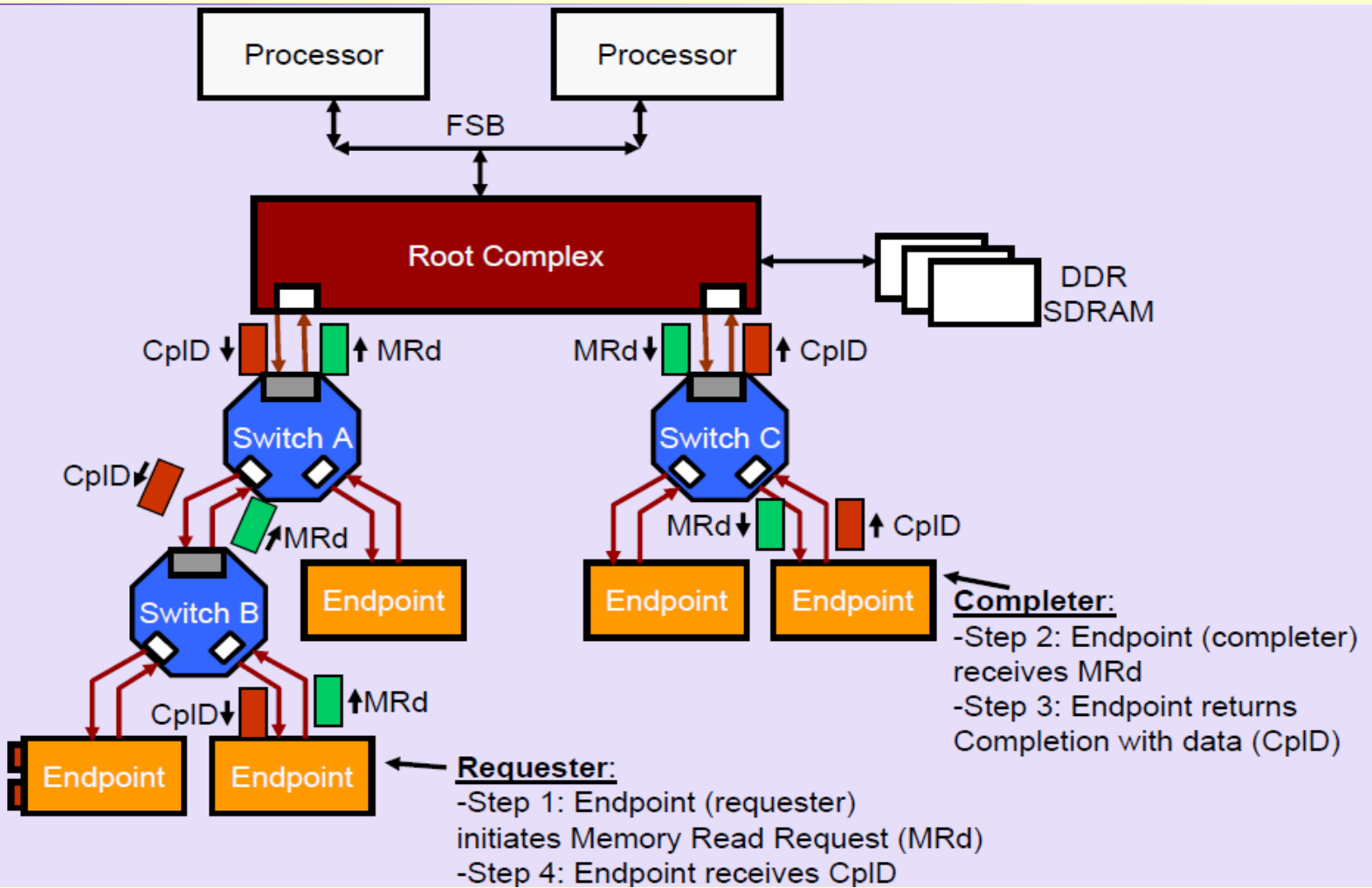
- Step 2: Root Complex (completer) receives MRd
- Step 3: Root Complex returns Completion with data (CpID)



## Requester:

- Step 1: Endpoint (requester) initiates Memory Read Request (MRd)
- Step 4: Endpoint receives CpID

# Peer-to-Peer Transaction



# Intreruperi

PCIe accepta doua feluri de intreruperi : Legacy INTx si MSI (Message Signaled Int.)

1. Întreruperile INTx sunt suportate pentru a păstra compatibilitatea cu software-ul moștenit și, de asemenea, pentru a permite legătura între magistralele PCI clasice și PCIe.
  - Întrucât întreruperile INTx sunt declanșate la nivel, există un pachet TLP pentru a spune că linia a fost activată, iar altul că a fost dezactivată.

2. Noua formă de întrerupere, MSI, a fost introdusă la PCI 2.2 (convențional)  
Idea a fost că, deoarece aproape toate perifericele PCI au capabilități de bus master, de ce sa nu fie lăsat perifericul sa semnalizeze o întrerupere prin scrierea la o anumită adresă?
  - PCIe face exact la fel ptr. a genera MSI:
    - Semnalizarea unei întreruperi constă în trimiterea unui pachet (TLP) pe bus, care este pur și simplu o postare a unui Write Request cu o adresă speciala, pe care host-ul a scris-o în spațiul de configurare al perifericului în timpul inițializării.

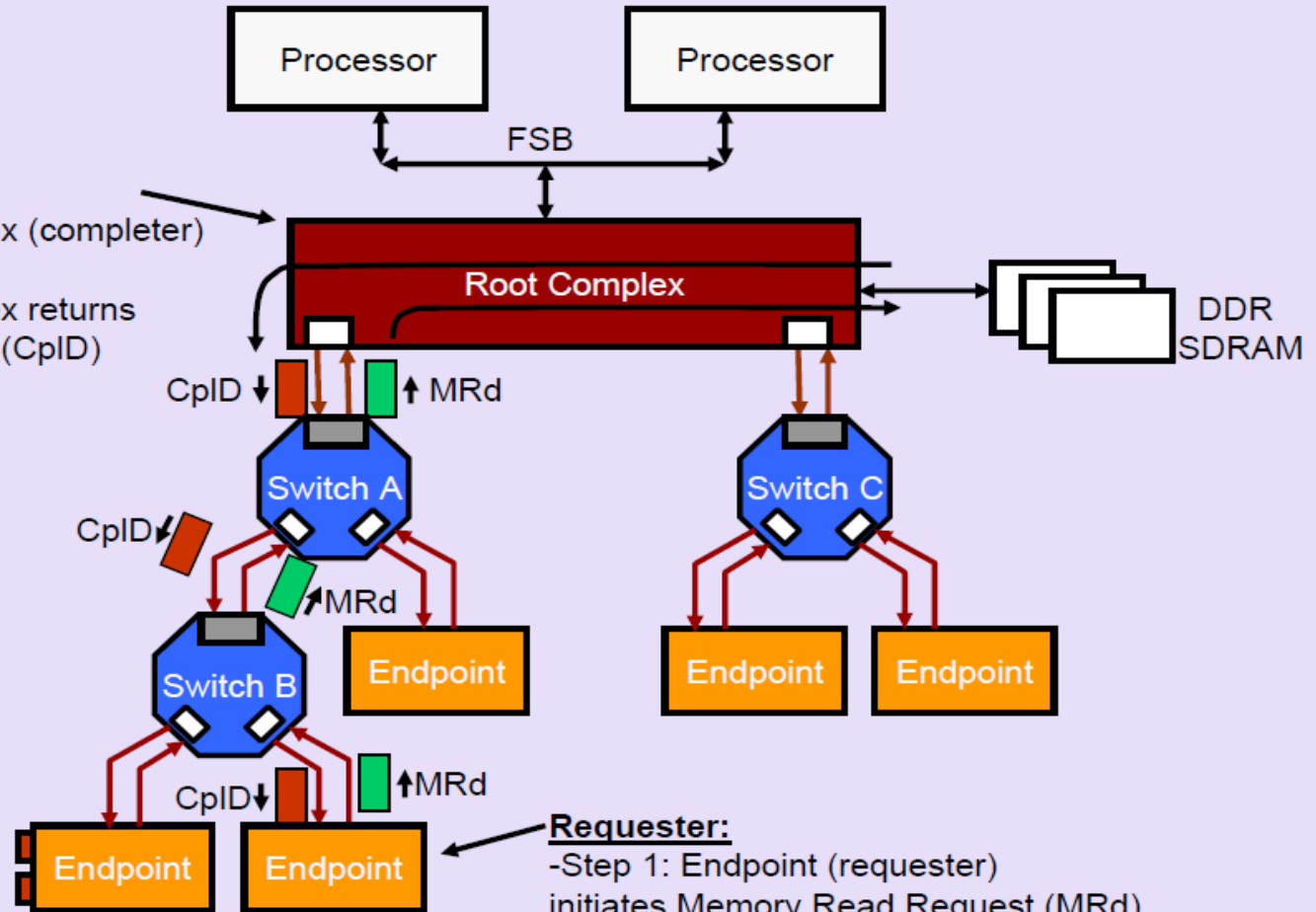
# Bus Mastering (DMA)

- Până la PCIe exista ceva intruziv în a spune CPU-ul să se retragă de pe bus pe durata DMA
- Pe 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).

# DMA Transaction

## Completer:

- Step 2: Root Complex (completer) receives MRd
- Step 3: Root Complex returns Completion with data (CpID)



## Requester:

- Step 1: Endpoint (requester) initiates Memory Read Request (MRd)
- Step 4: Endpoint receives CpID

# Funcția stratului fizic (Physical Layer)

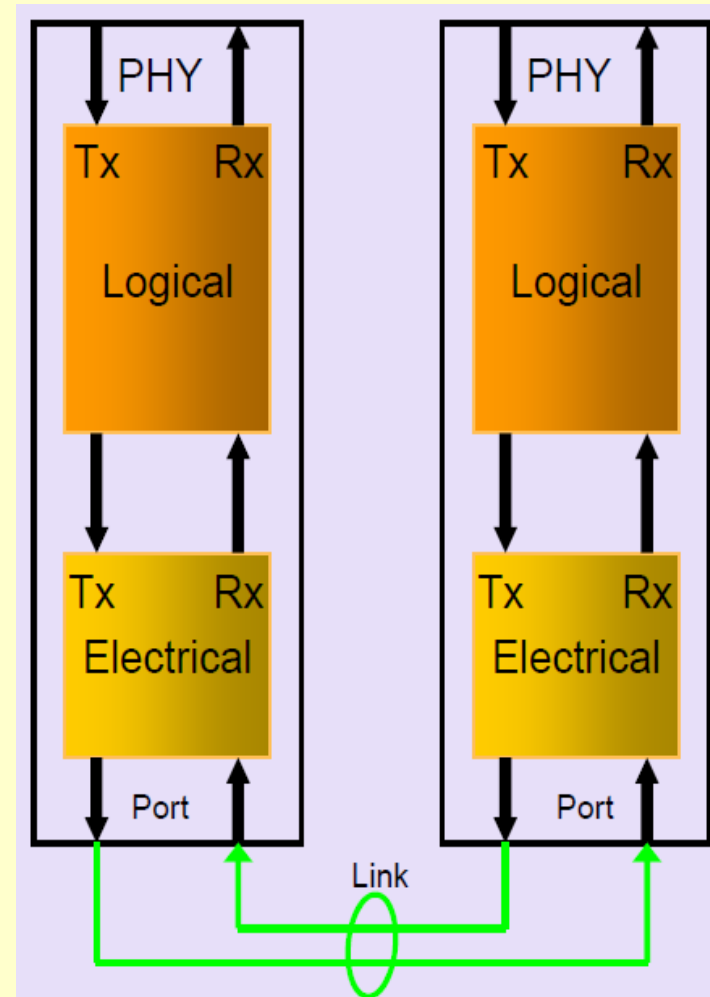
Oferă conexiunea fizică între dispozitive:

## Funcții logice

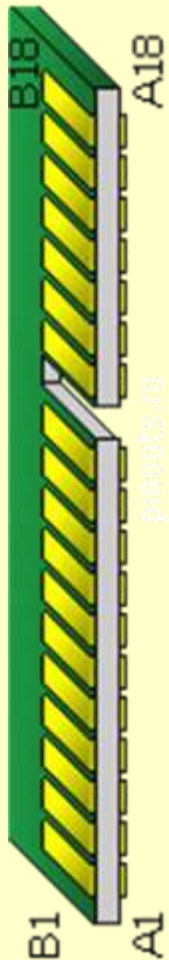
- Încadrarea pachetelor, asamblare de date
- Codificarea datelor, codare/decodare 8b/10b
- Blocarea simbolurilor

## Funcții electrice

- Detectarea receptorului
- Recuperarea ceasului de recepție
- Blocare bit, serializare/deserializare
- Semnalizare



# PCI-Express 1x Connector Pin-Out



Pin #	Side B Connector		Side A Connector	
	Name	Description	Name	Description
1	+12v	+12 volt power	PRSNT#1	Hot plug presence detect
2	+12v	+12 volt power	+12v	+12 volt power
3	+12v	+12 volt power	+12v	+12 volt power
4	GND	Ground	GND	Ground
5	SMCLK	SMBus clock	JTAG2	TCK
6	SMDAT	SMBus data	JTAG3	TDI
7	GND	Ground	JTAG4	TDO
8	+3.3v	+3.3 volt power	JTAG5	TMS
9	JTAG1	+TRST#	+3.3v	+3.3 volt power
10	3.3Vaux	3.3v volt power	+3.3v	+3.3 volt power
11	WAKE#	Link Reactivation	PWRGD	Power Good
Mechanical Key				
12	RSVD	Reserved	GND	Ground
13	GND	Ground	REFCLK+	Reference Clock Differential pair
14	HSOp(0)	Transmitter Lane 0, Differential pair	REFCLK-	
15	HSOn(0)		GND	Ground
16	GND	Ground	HSIp(0)	Receiver Lane 0, Differential pair
17	PRSNT#2	Hotplug detect	HSIn(0)	
18	GND	Ground	GND	Ground



# PCI Express Error Handling

- Toate dispozitivele PCI Express suporta o anumită combinație ptr. manipularea erorilor

# Software-ul existent scris pentru manipularea erorilor la PCI conventional, profită de faptul că PCI Express și-a mapat multe dintre condițiile de eroare la mecanismele existente de manipulare ale erorilor PCI.

# Mecanisme suplimentare de raportare specifice PCI Express

- Erorile sunt clasificate ca si *corectabile si necorectabile*.
- Erorile *necorectabile se impart* in:
  - # erori necorectabile Fatale
  - # erori necorectabile Non-fatale

# Erori Corectabile

- Erorile clasificate ca si corectabile, *degradeaza performantele sistemului*, dar recuperarea se poate face fara pierdere de informatii
  - # Hardware-ul este raspunzator de recuperare din erorile corectabile si nu necesita interventia software
- Chiar dacă hardware-ul se ocupă de corecție, înregistrarea frecvenței erorilor care pot fi corectate poate fi utilă în cazul în *care software-ul efectuează operații de monitorizare a legăturilor*.
- Un ex. de eroare corectabilă este detectarea unei erori de link CRC (LCRC) atunci când este trimis un TLP, rezultând un eveniment de retransmitere in Data Link Layer

# Erori necorectabile

- Erorile clasificate ca necorectabile afectează funcționalitatea interfeței și nu există niciun mecanism specific pentru a corecta aceste erori
- Cele doua grupuri sunt: fatale si non-fatale

## 1. Erori fatale necorectabile: Erori care fac link-ul nesigur

- *Strategia de nivel 1* pentru recuperare poate implica o resetare a legăturilor de către sistem
- Gestionarea erorilor fatale este specifică platformei

## 2. Erori necorectabile non-fatale: Erori nerecuperabile asociate cu o *anumită tranzacție*, în timp ce legătura în sine este fiabilă

- Software-ul poate limita strategia de recuperare la dispozitivul (dispozitivele) implicat
- Tranzacțiile între alte dispozitive nu sunt afectate

# PCI Express Throughput

Debitul unui sistem PCI Express depinde de următorii factori:

- Redundanta de protocol
- Dimensiunea încărcăturii utile
- Latența de finalizare
- Latența de actualizare a controlului fluxului
- Dispozitivele care formează legătura

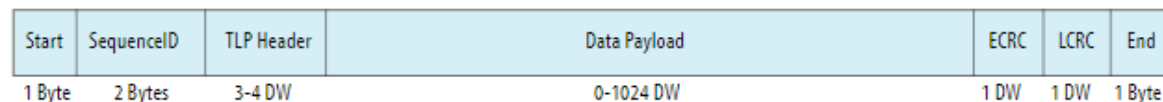
## Supraîncărcarea protocolului

includ următoarele trei componente:

- Codificarea/decodificarea 128b/130b (overheadde 1,56%) - legăturile Gen3 utilizează codificarea 128b/130b. Această codificare adaugă doi biți de sincronizare (sync) la fiecare transfer de date pe 128 de biți. În consecință, codificarea și decodificarea. Rata efectivă de date a unei legături Gen3 x8 este de aproximativ 8 GBps.
- Pachete pentru stratul de legătură de date (DLLP) și pachete pentru stratul fizic (PLP) PLP-urile constau în seturi ordonate SKP care au 16-24 de octeți. DLLP-urile sunt 2Dw. DLLP-urile implementează controlul fluxului și protocolul ACK/NAK.
- Redundanta pachetelor TLP – Overhead-ul asociat cu un singur TLP variază între 5-7 Dw dacă nu este inclus ECRC opțional. Cheltuielile generale includ următoarele câmpuri:

- The Start and End Framing Symbols
- The Sequence ID
- A 3- or 4-dword TLP header
- The Link Cyclic Redundancy Check (LCRC)
- 0-1024 Dw of data payload

Figure 4. TLP Packet Format



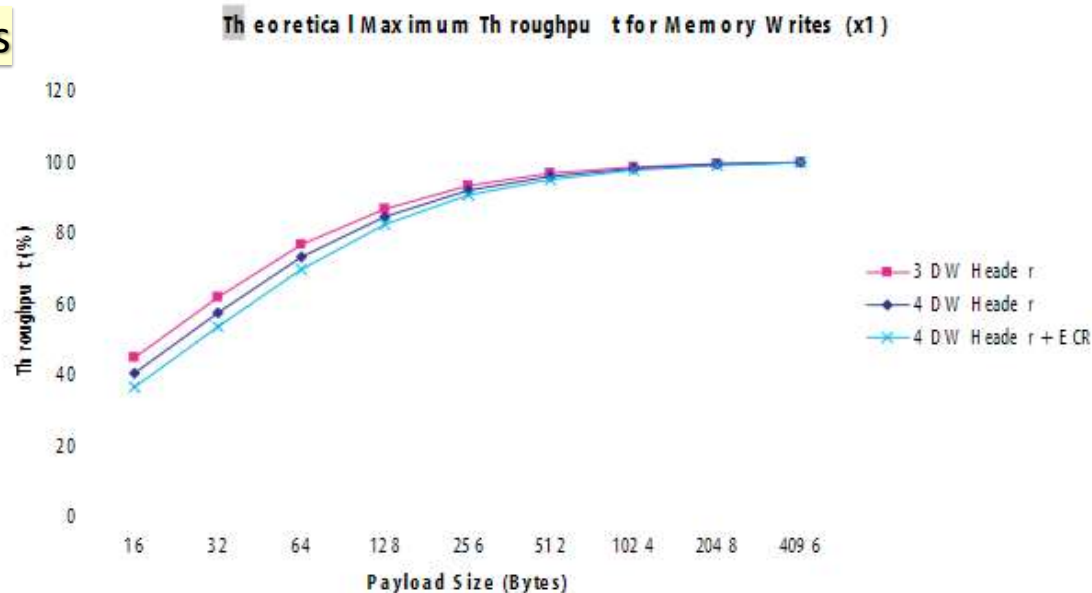
# Throughput for Posted Writes

Calculul debitului maxim teoretic utilizează următoarea formulă:

$$\text{Throughput} = \text{payload size} / (\text{payload size} + \text{overhead}) * \text{link data rate}$$

- Graficul prezintă debitul maxim cu diferite dimensiuni ale antetului TLP și ale încărcăturii utile.
- DLLP și PLP sunt excluse din acest calcul. Pentru o dimensiune maximă a încărcăturii utile de 256 de octeți și un antet de 3 cuvinte, depășirea este de 5 cuvinte. Deoarece interfața este de 256 de biți, antetul de 5 cuvinte necesită un singur ciclu de bus. Sarcina utilă de 256 de octeți necesită 8 cicluri de bus.

## Maximum Throughput for Memory Writes



- Următoarea ecuație arată debitul maxim teoretic:

$$\text{Maximum throughput} = 8 \text{ cycles} / 9 \text{ cycles} = 88.88\% * 8 \text{ GBps} = 7.2 \text{ GBps}$$

# Beneficiile PCI Express

- Pentru sistemele de măsurare și automatizare bazate pe PC, bus-ul PCI a fost magistrala aleasa pentru plăcile de extensie plug-in de mai mulți ani.
- Va continua să joace un rol important în viitor. Pe măsură ce calculatorul a evoluat, magistrala PCI (cu arhitectura sa paralelă) nu s-a scalat liniar cu restul platformei. PCI Express răspunde la aceste probleme și oferă beneficii în cinci domenii principale:

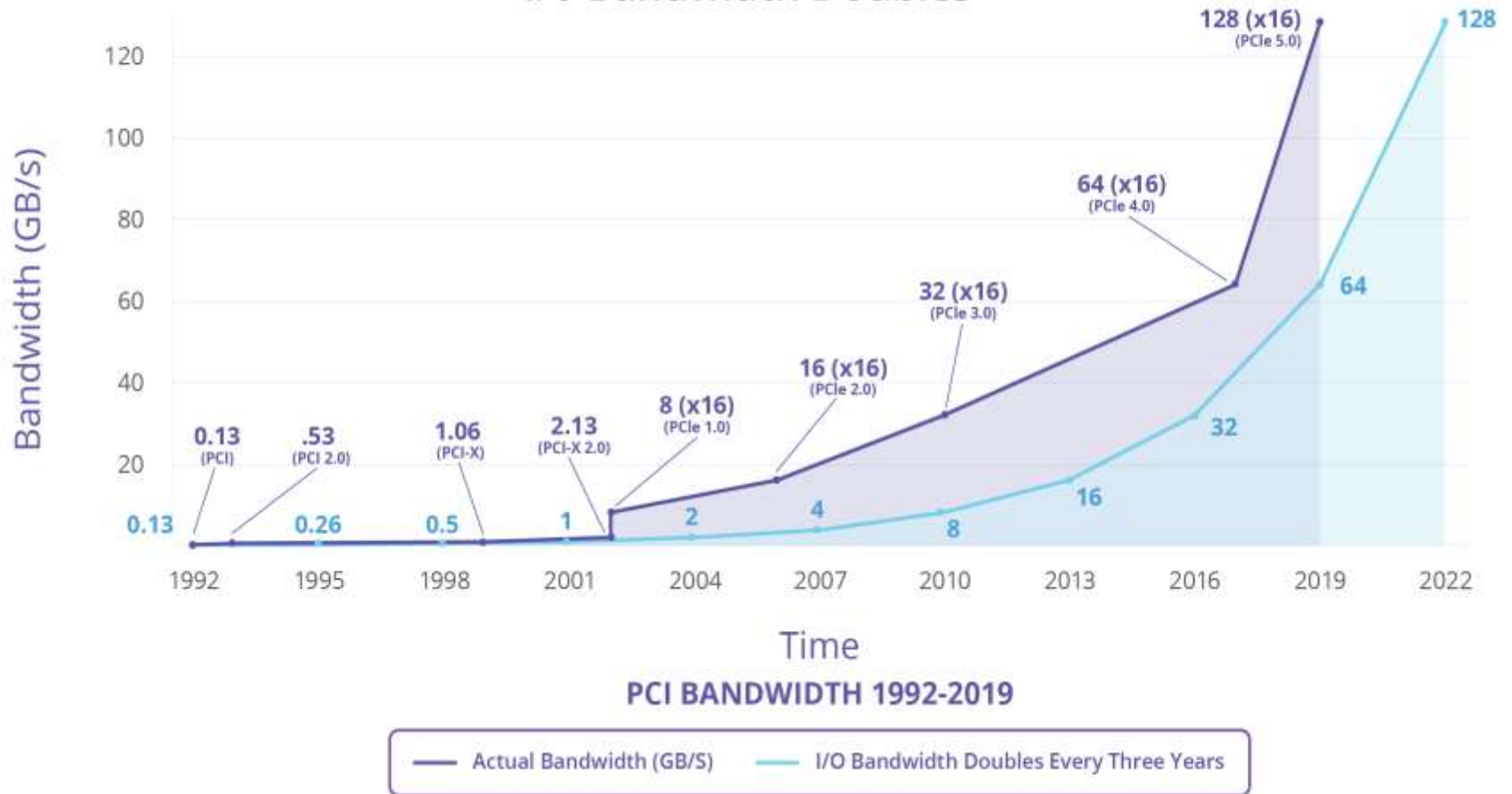
- **Performanță ridicată** - se referă în special la lățimea de bandă, care este mai mult decât dublul față de PCI într-o legătură x1 și crește liniar pe măsură ce se adaugă mai multe lane-uri. Un beneficiu suplimentar care nu este imediat evident este faptul că această lățime de bandă este disponibilă 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 capacități 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<sup>®</sup> (Supermicro I/O Module) și ExpressCard, crește mult capacitatea de a adăuga periferice de înaltă performanță în servere și notebook-uri.

# Evolutia PCIe



## EVERY 3 YEARS

I/O Bandwidth Doubles





# PCIe 5.0 Delivering 32GT/s



## ✓ Supports 400Gb Ethernet Solutions

- 400Gb = 50GB
- 50GB in both directions

## ✓ Full Duplex

- 128/130 bit encoding with 1.5% overhead
- x16 ~64GB/s sufficient to support 400Gb Ethernet solutions (64GB > 50 GB)
- Total Full Duplex = ~128GB

- CEM connector targeted to be backwards compatible for add-in cards
- **Targeted Release in 2019**

	RAW BIT RATE	LINK BW	BW/ LANE/WAY	TOTAL BW X16
PCIe 1.x	2.5GT/s	2Gb/s	250MB/s	8GB/s
PCIe 2.x	5.0GT/s	4Gb/s	500MB/s	16GB/s
PCIe 3.x	8.0GT/s	8Gb/s	~1GB/s	~32GB/s
PCIe 4.0	16GT/s	16Gb/s	~2GB/s	~64GB/s
PCIe 5.0	32GT/s	32Gb/s	~4GB/s	~128GB/s

[https://www.eejournal.com/article/pci-sigs-pcie-6-0-spec-continues-the-long-happy-evolution-of-pci-express/?vgo\\_ee=Z5loYywhGikONfrMaKiC9rtepGgMcmV8AxLw8cRwx78%3D](https://www.eejournal.com/article/pci-sigs-pcie-6-0-spec-continues-the-long-happy-evolution-of-pci-express/?vgo_ee=Z5loYywhGikONfrMaKiC9rtepGgMcmV8AxLw8cRwx78%3D)

**PCI Express Bandwidth**  
(Full Duplex: GB/second/direction)

<i>Slot Width</i>	PCIe 1.0 (2003)	PCIe 2.0 (2007)	PCIe 3.0 (2010)	PCIe 4.0 (2017)	PCIe 5.0 (2019)	PCIe 6.0 (2022)
<b>x1</b>	0.25GB/sec	0.5GB/sec	~1GB/sec	~2GB/sec	~4GB/sec	8GB/sec
<b>x2</b>	0.5GB/sec	1GB/sec	~2GB/sec	~4GB/sec	~8GB/sec	16GB/sec
<b>x4</b>	1GB/sec	2GB/sec	~4GB/sec	~8GB/sec	~16GB/sec	32GB/sec
<b>x8</b>	2GB/sec	4GB/sec	~8GB/sec	~16GB/sec	~32GB/sec	64GB/sec
<b>x16</b>	4GB/sec	8GB/sec	~16GB/sec	~32GB/sec	~64GB/sec	128GB/sec

# References

[https://indico.cern.ch/event/121654/attachments/68430/98164/Practical introduction to PCI Express with FPGAs - Extended.pdf](https://indico.cern.ch/event/121654/attachments/68430/98164/Practical%20introduction%20to%20PCI%20Express%20with%20FPGAs%20-%20Extended.pdf)

Budruk, R., Anderson, D., Shanley, T., *PCI Express System Architecture*, MindShare Inc., Addison-Wesley Developer's Press, 2008, <https://www.mindshare.com/files/ebooks/PCI%20Express%20System%20Architecture.pdf>

Ajanovic, J., "PCI Express (PCIe) 3.0 Accelerator Features", Intel Corporation, 2008, <http://www.intel.com/content/dam/doc/white-paper/pci-express3-accelerator-white-paper.pdf>.

PCI-SIG, "PCI Express Base Specification Revision 3.0", November 10, 2010.

[https://webcourse.cs.technion.ac.il/236376/Spring2017/ho/WCFiles/chipset\\_microarch.pdf](https://webcourse.cs.technion.ac.il/236376/Spring2017/ho/WCFiles/chipset_microarch.pdf)

<http://xillybus.com/tutorials/pci-express-tlp-pcie-primer-tutorial-guide-1>

<http://xillybus.com/tutorials/pci-express-tlp-pcie-primer-tutorial-guide-2>

<http://hardwareverification.weebly.com/pci---express-introduction.html>