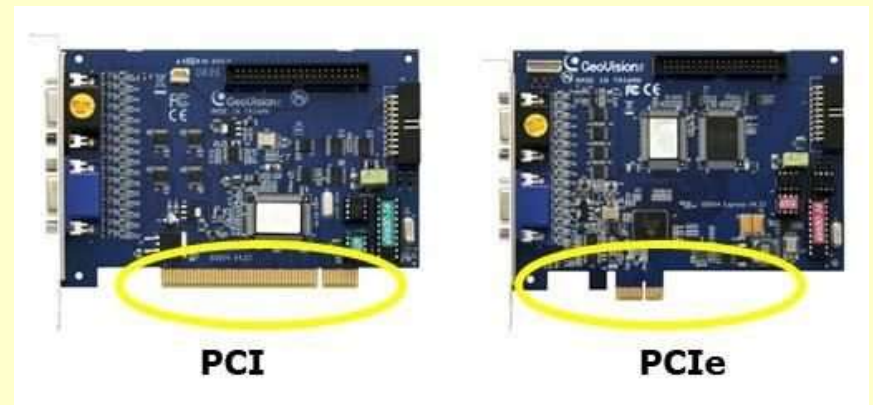


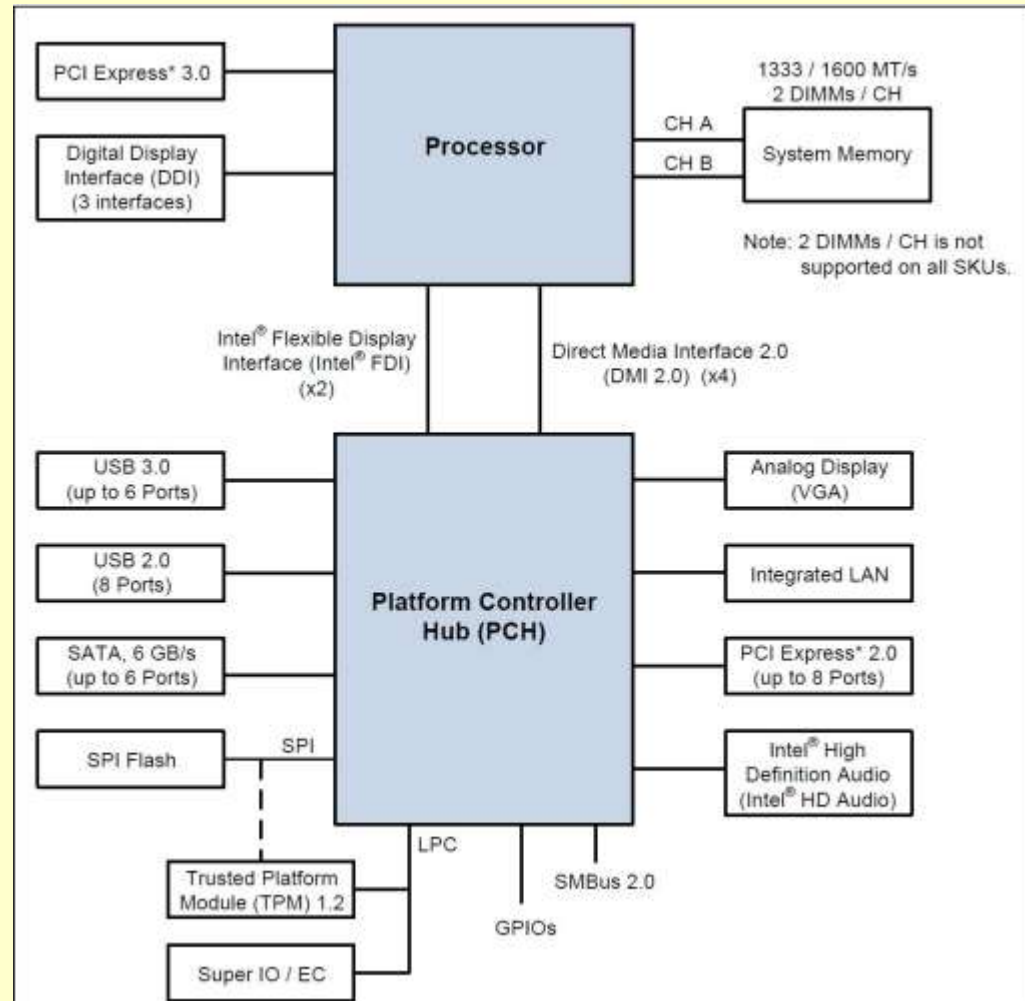
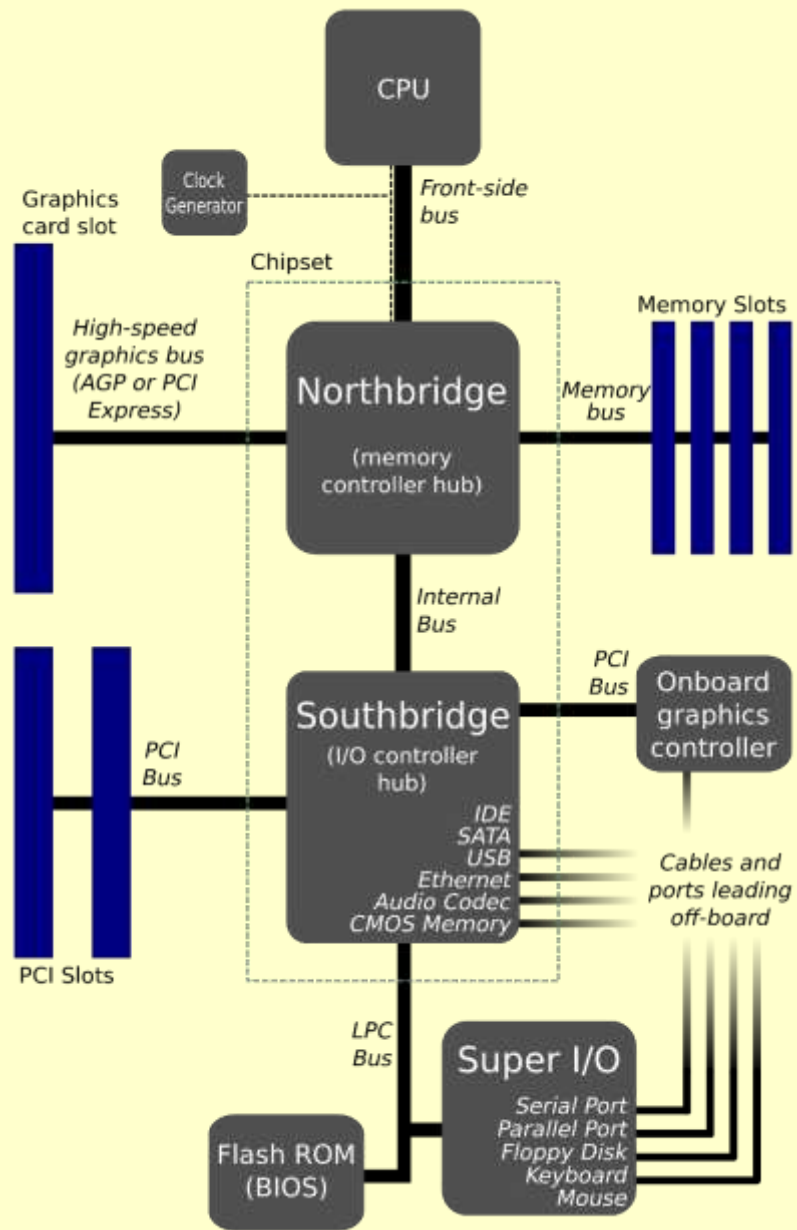
C12. PCIe overview

OUTLINE

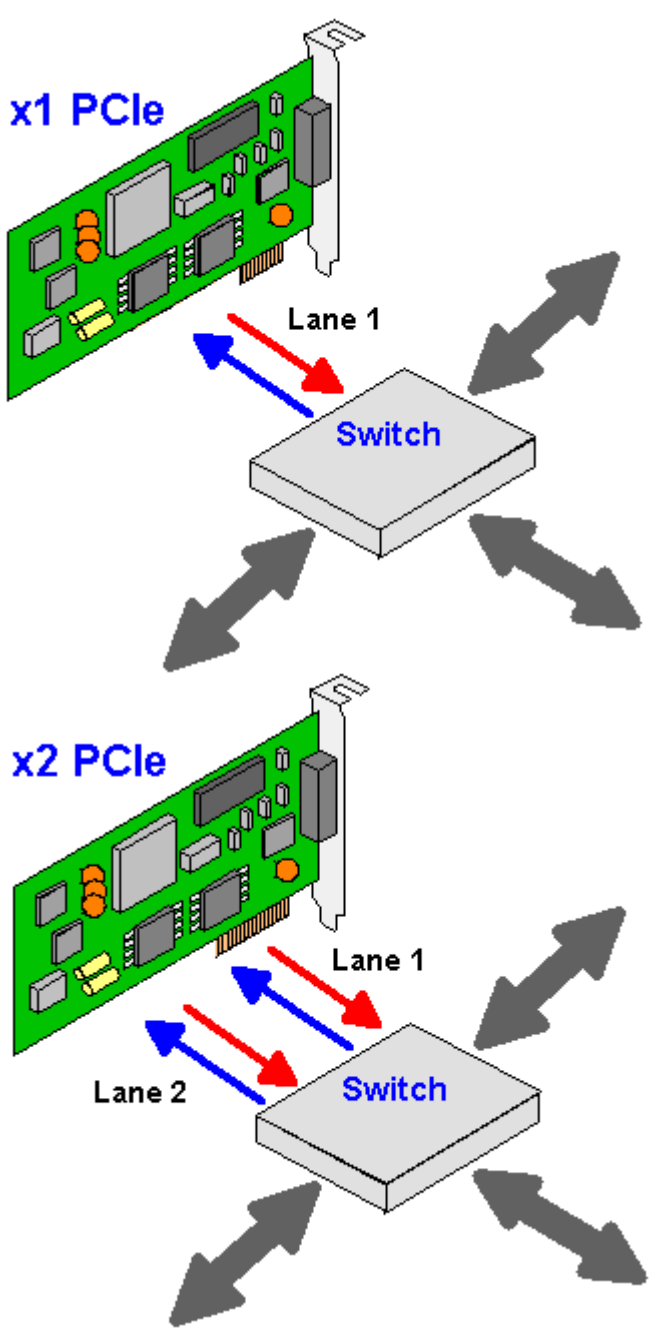
- PCI Express overview
- PCI architecture
 - PCI Express link
 - bus topology
 - architecture layers
 - transactions
 - interrupts



- Introduced as "Third Generation I/O" (3GIO), PCI Express (PCIe) superseded both PCI and PCI-X and new motherboards may come with a mix of PCI and PCIe slots or only PCIe.



PCI Express overview

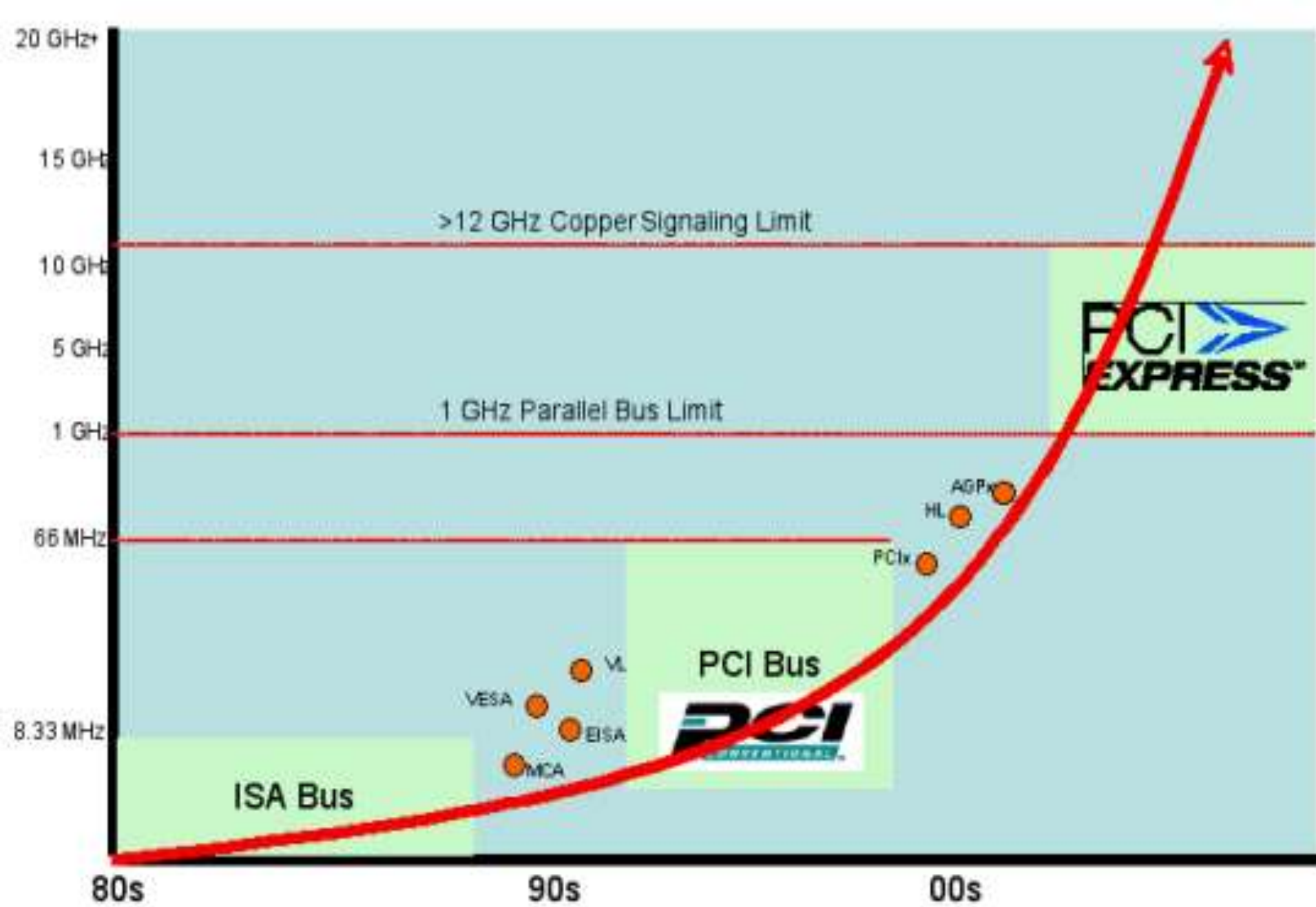


PCIe is a Switched Architecture - Multiple Lanes* rather than the shared parallel-bus structure of PCI

- PCIe provides a switched architecture of channels that can be combined in x2, x4, x8, x16 and x32 configurations, creating a parallel interface of independently controlled "lanes."
- The switch backplane gives the total bandwidth, and cards and motherboards are compatible between versions.
- For comparisons of all PCI technologies, see [PCI-SIG](#), [PCI](#), [ExpressCard](#), [PCI-X](#), [SATAExpress](#) and [Thunderbolt](#).

* *The transmit and receive pair together are called a lane.*

The initial speed of 2.5 Gb/s provides a nominal bandwidth of about 250 MB/s in each direction per PCI Express lane.



Year created	2004
Created by	Intel · Dell · HP · IBM
Supersedes	AGP · PCI · PCI-X
Width in bits	1–32
	One device each on each endpoint of each connection.
Number of devices	PCI Express switches can create multiple endpoints out of one endpoint to allow sharing one endpoint with multiple devices.
	Per lane (each direction):
	•v1.x: 250 MB/s (2.5 GT/s)
	•v2.x: 500 MB/s (5 GT/s)
	•v3.0: 985 MB/s (8 GT/s)
	•v4.0: 1969 MB/s (16 GT/s)
	So, a 16-lane slot (each direction):
	•v1.x: 4 GB/s (40 GT/s)
	•v2.x: 8 GB/s (80 GT/s)
	•v3.0: 15.75 GB/s (128 GT/s)
	•v4.0: 31.51 GB/s (256 GT/s)
Capacity	
	Serial
Style	
Hotplugging interface	Yes, if Express Card, Mobile PCI Express Module or XQD card
External interface	Yes, with PCI Express External Cabling, such as Thunderbolt

PCI express features

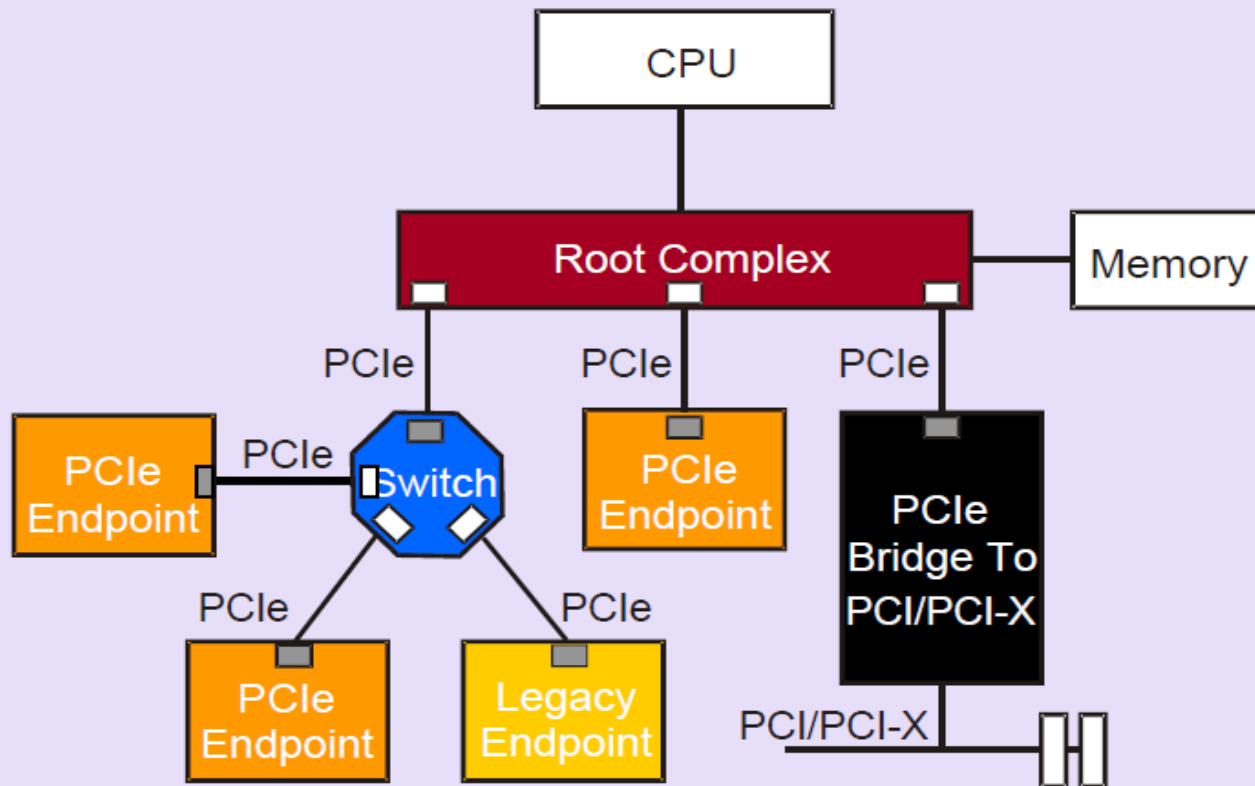
- *PCI Express architecture is a high performance, I/O interconnect for peripherals in computing communication platforms.*
- Evolved from PCI and PCI-X architectures and uses the same communication model as these buses.
- The same address spaces are retained: memory, I/O and configuration.
- PCI and PCI-X generations shared parallel buses, the PCIe bus uses a *serial point-to-point interconnect* for communication between two peripheral devices.
- PCIe implements *packet-based protocol* for information transfer
- Scalable performance based on *number of signal lanes** implemented on the PCIe interconnect (dual simplex)
- The *PCIe bus allows the same types of transactions* as the previous buses: memory read/write, I/O read/write, and configuration read/write...
- The compatibility is maintained with existing OS and software drivers, which do not require changes.

PCI Express Features

- The interface is serial, which enables to reduce the pin count and to simplify the interconnections
- It unifies the I/O architecture for different types of systems and embedded systems
- It enables to interconnect IC on the motherboard and expansion cards via connectors or cables
- The communication is based on packets with high transfer rate and efficiency
- The bus is scalable, by ability to implement a particular interconnection via several communication lanes
- The software model is compatible with the classical PCI architecture, which allows to configure PCIe devices, to use existing software drivers, without the need for changes
- It provides a differentiated quality of service (QoS) through the ability to allocate dedicated resources for certain data flows, to configure the QoS arbitration policies for each component, and to use isochronous transfers for real-time applications
- It provides an advanced power management through the ability to identify power management capabilities of each peripheral device
- It ensures link-level data integrity for all types of transactions.
- It supports advanced error reporting and handling to improve fault isolation and error recovery
- It supports hot-plugging and hot-swapping of peripheral devices

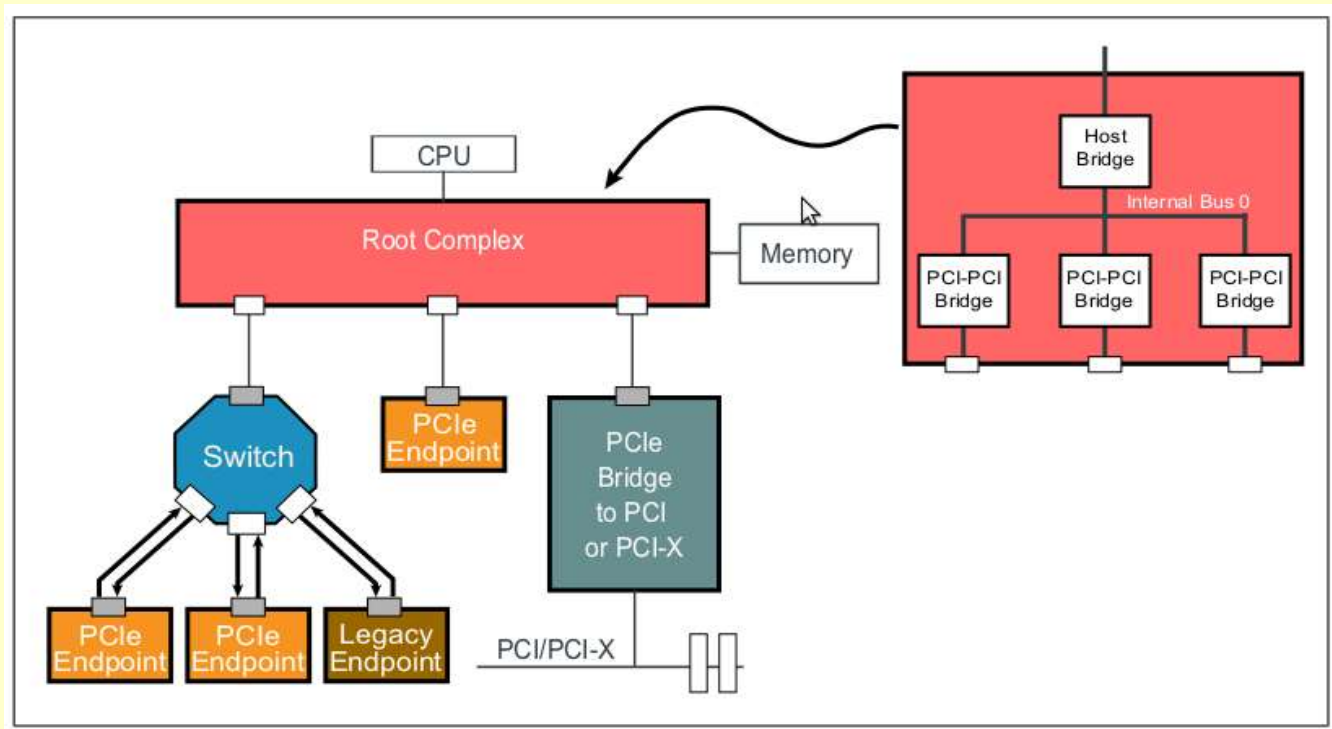
PCI Express Topology

- PCIe system is comprised of PCIe links that interconnect a set of components
- An example of topology referred to as a hierarchy – composed of:
 - a Root Complex ,
 - multiple Endpoints (I/O devices),
 - a Switch
 - a PCI Express to PCI/PCI-X Bridge, all interconnected via PCI Express Links



Root Complex (RC)

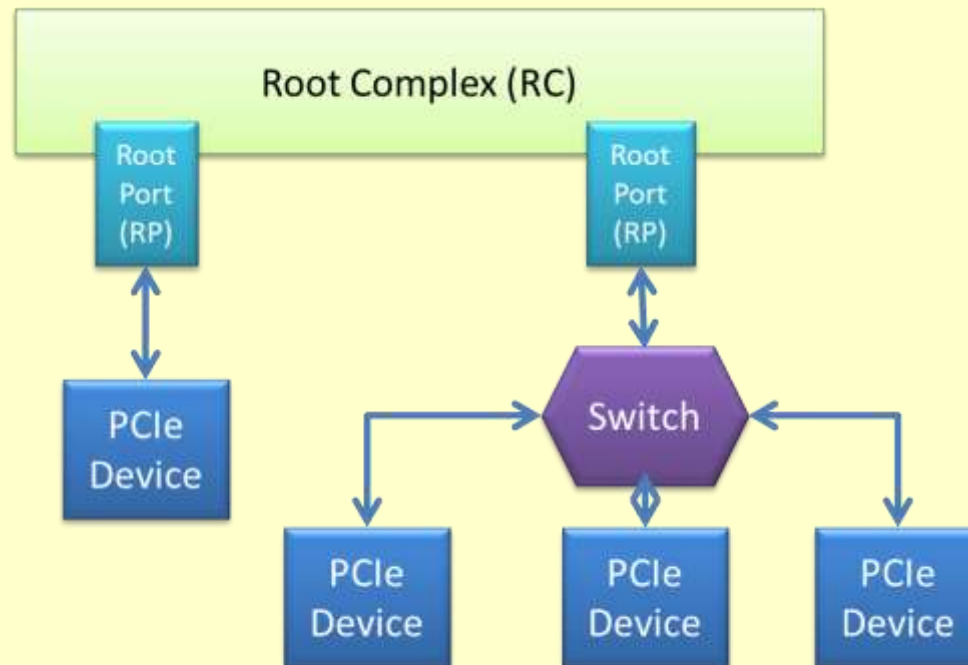
- **Root Complex (RC)** – is the device that connects one or more processors and the memory subsystem to the I/O devices.
- RC device represents the root of an I/O hierarchy
- Similar to a host bridge in a PCI system : - RC generates transaction requests on behalf of the processor, which is interconnected through a local bus.
 - RC may support one or more PCI Express Ports – Root Ports.



- A RC implements various resources, such as interrupt controller, power management controller, error detection and reporting logic.
- RC contains an internal bus, which represents the bus 0 in the entire hierarchy
- RC **logically** aggregates PCIe hierarchy domains into one single PCIe hierarchy.

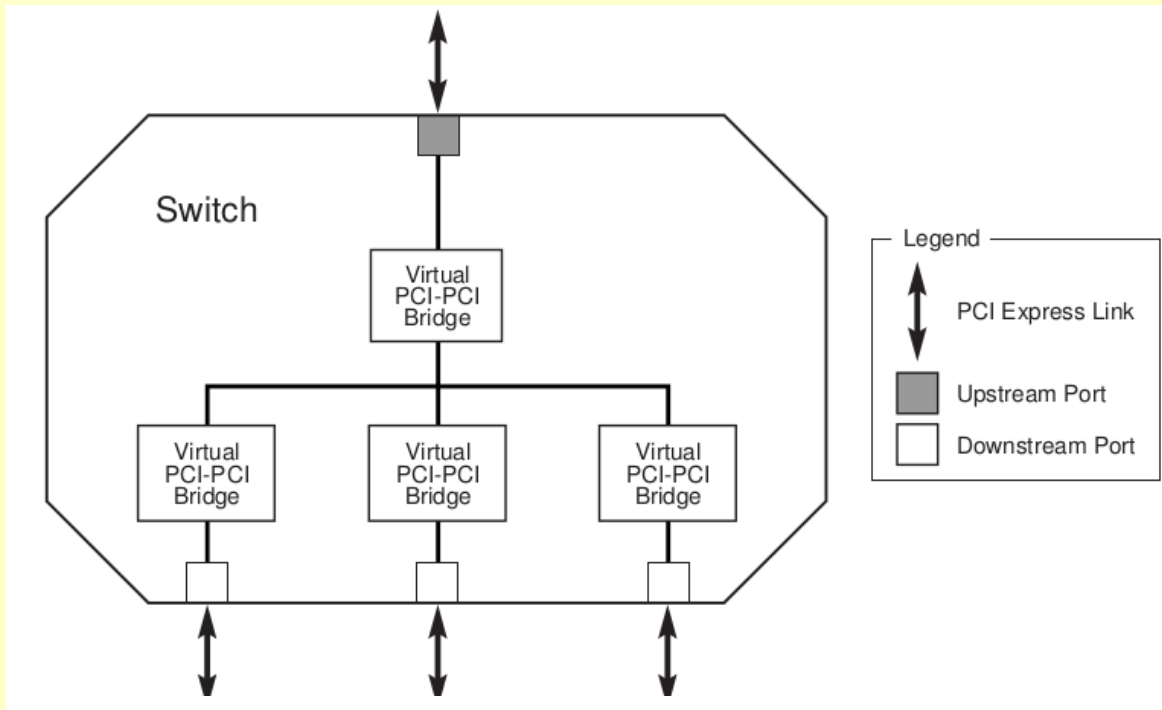
PCIe Root Port

- The part of the motherboard that contains the host bridge. The host bridge allows the PCIe ports to talk to the rest of the computer
- Each Root Port defines a separate hierarchy domain
- Each hierarchy domain may be composed of a single Endpoint or a sub-hierarchy containing one or more Switch components and Endpoints
- The capability to route peer-to-peer transactions between hierarchy domains through a Root Complex is *optional* and implementation dependent.



Switches and Bridges

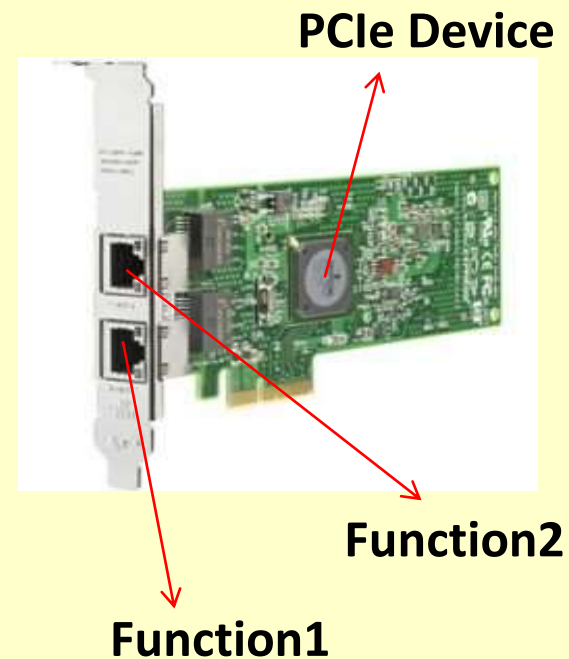
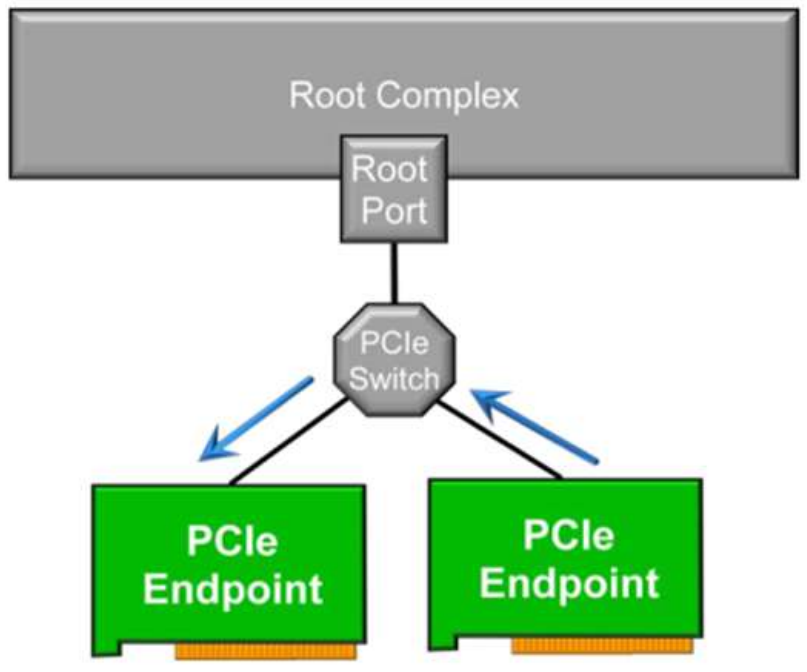
- **A switch** is defined as *an assembly of two or more logical PCI-to-PCI bridges*, each bridge associated with a switch port; these bridges are connected via an internal bus.
- **Switches** provide the capability to more devices to be attached to a single RP
- They act as *packets routers* and recognize which path a given packet will need to take based on its address or other routing information
- **Bridges** provide an interface to other buses, such as PCI/PCI X, USB, InfiniBand, Ethernet or even another PCI-e bus
- Switches may have *several Downstream Ports* but can only have *one Upstream Port*



Internal structure of a switch

PCIe Endpoint Devices

- **Endpoints** represent peripheral devices that participate to PCIe transactions
- An *initiator (requester)* endpoint initiates a transaction in the PCIe system, while a *target (completer)* endpoint responds to transactions that are addressed to it
- Switches/bridges *don't act* as initiators and completers of transactions on the bus
- Endpoints reside at the bottom of the branches of the tree topology and *only implement a single Upstream Port (toward the Root)*.

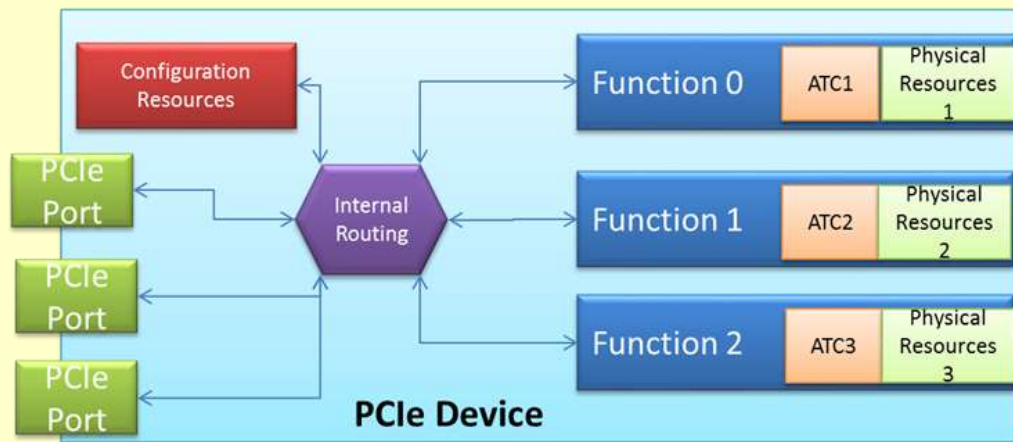


- PCIe devices may have up to *8 logical functions* and each endpoint is assigned a *device identifier (ID)*, which consists of *a bus number, device number, and function number*.
- The link and PCIe functionality shared by all functions is managed through *Function 0*
- All functions use a single *Bus Number* captured through the PCI enumeration process

Byte Offset			
31			
0			
Device ID		Vendor ID	
Status		Command	
Class Code		Revision ID	
BIST	Header Type	Master Latency Timer	Cache Line Size
Base Address Registers			
Cardbus CIS Pointer			
Subsystem ID		Subsystem Vendor ID	
Expansion ROM Base Address			
Reserved		Capabilities Pointer	
Reserved			
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line

OM14316

Figure 3-2: Type 0 Configuration Space Header

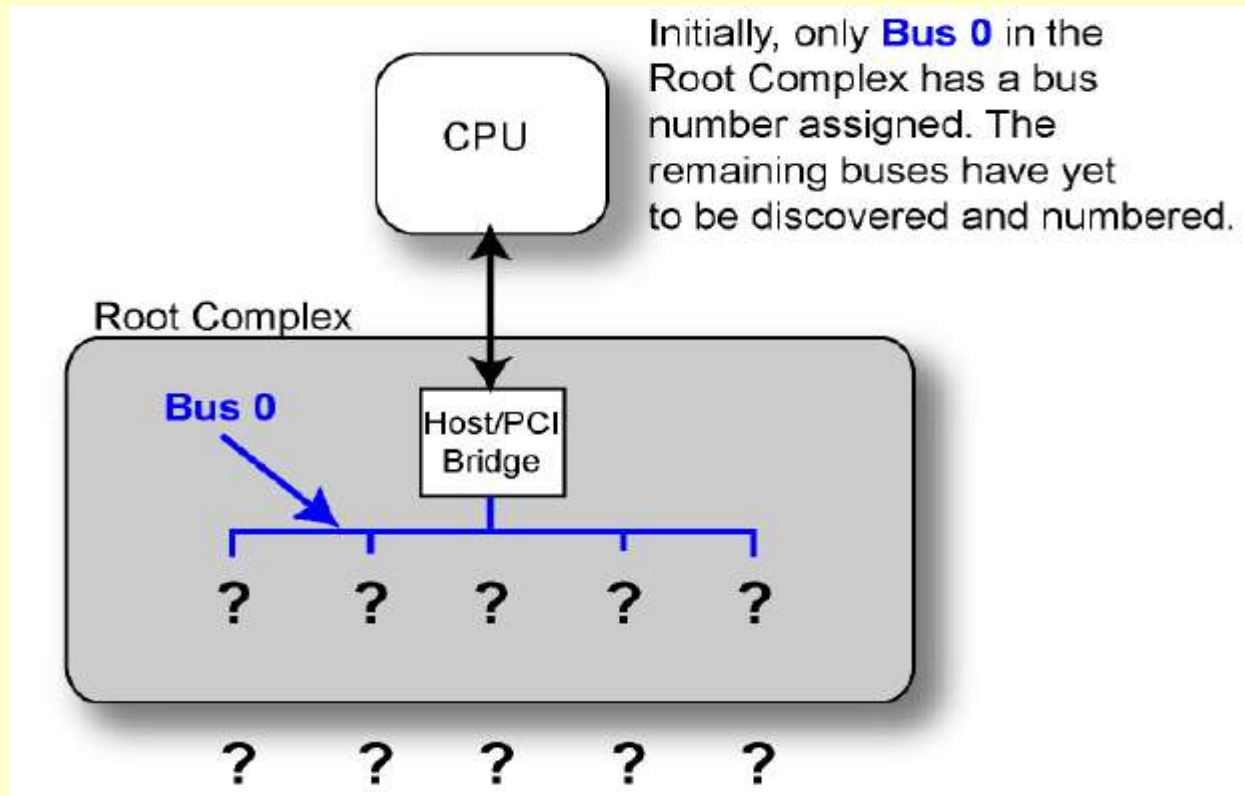


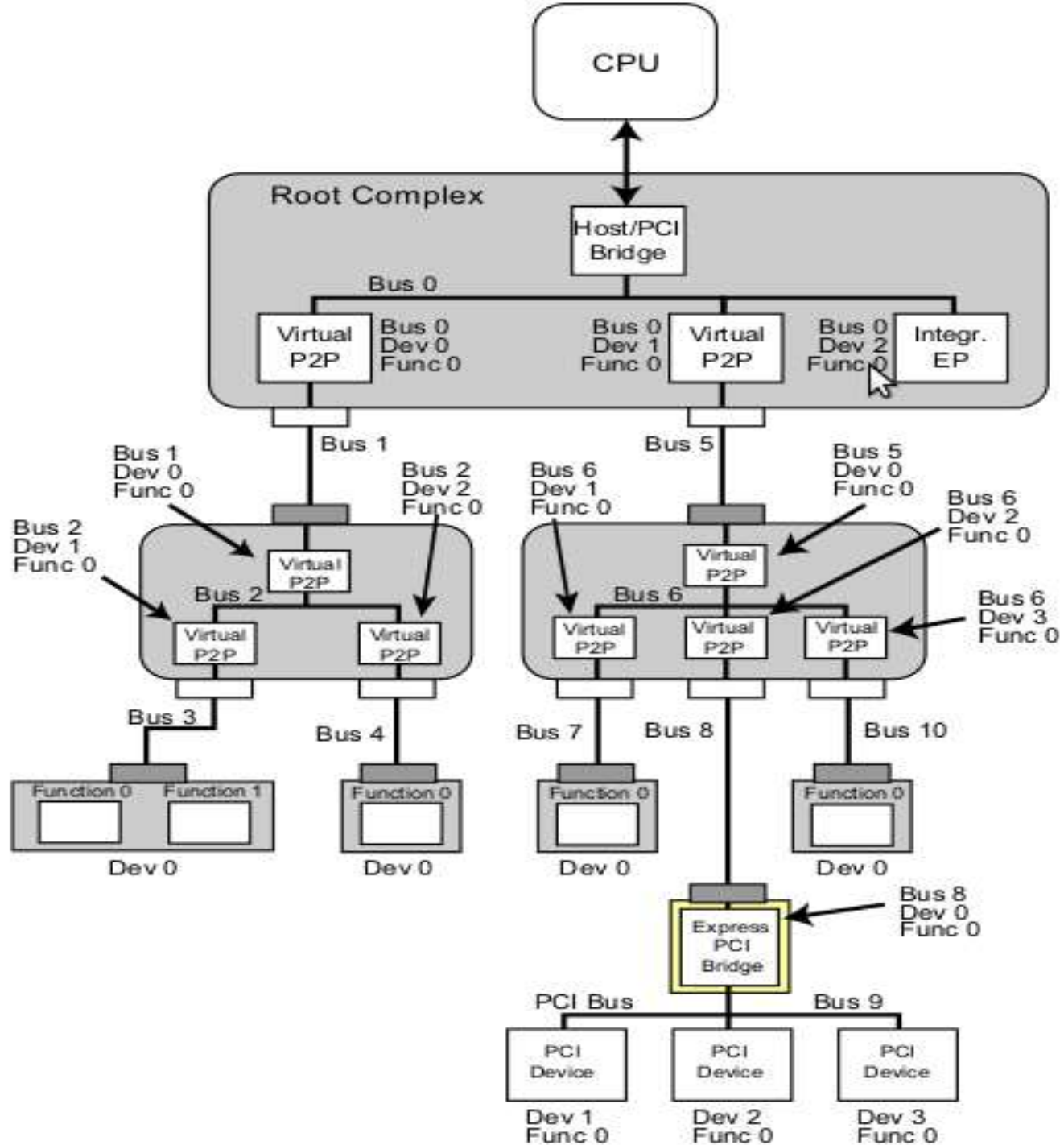
Configuration Space

- Devices will *allocate resources such as memory* and record the address into this configuration space

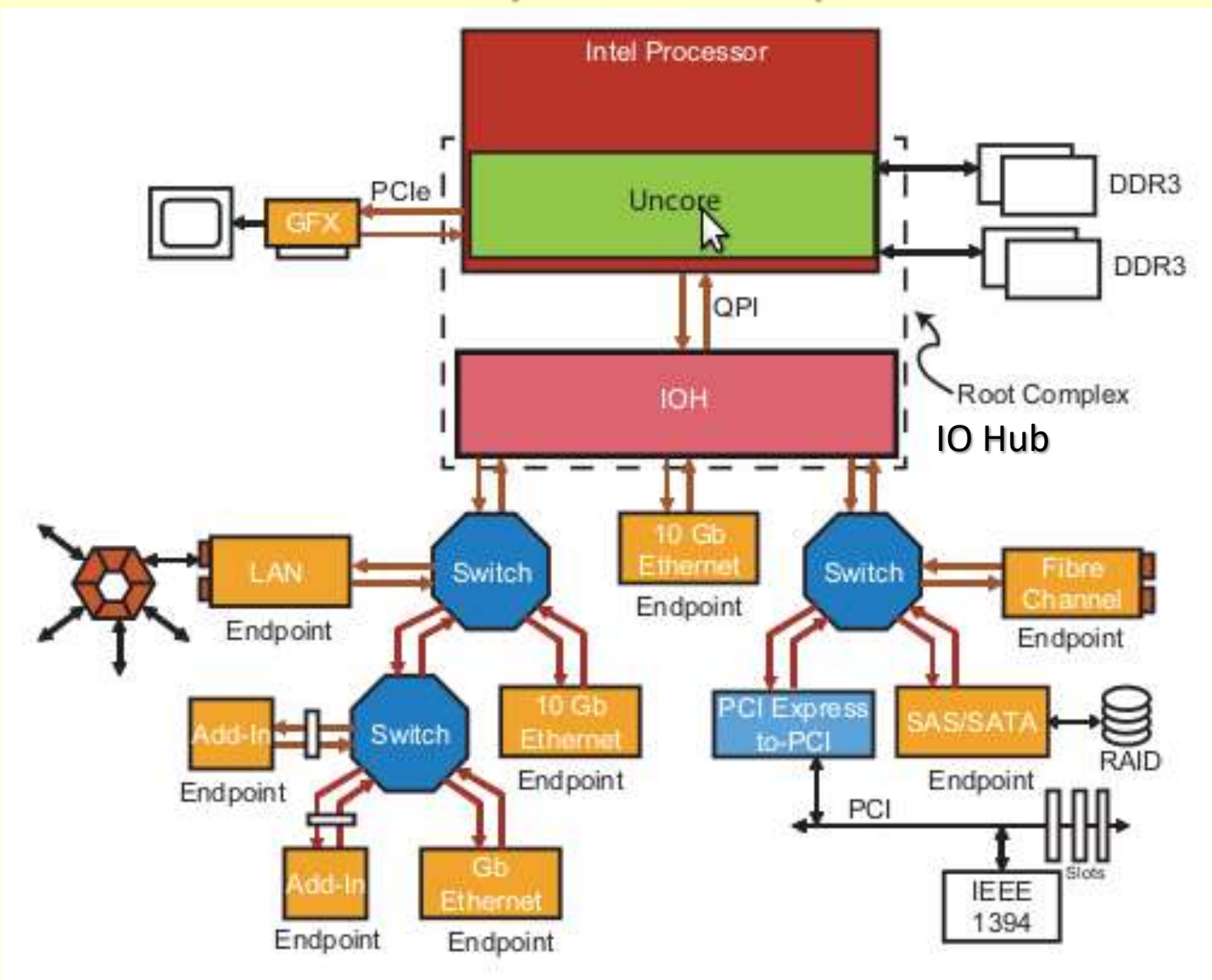
Enumeration

- The process by which configuration software discovers *the system topology* and *assigns bus numbers and system resources*.
- RC or Host sends *Configuration Packets* to assign unique Bus, Device and Function numbers to the End Points connected to the bus
- On x86 PCI-e hierarchy enumeration done by *BIOS on hardware initialization state* – all registers are configured before bootloader
- System software can reassign enumeration according to enumeration rules.



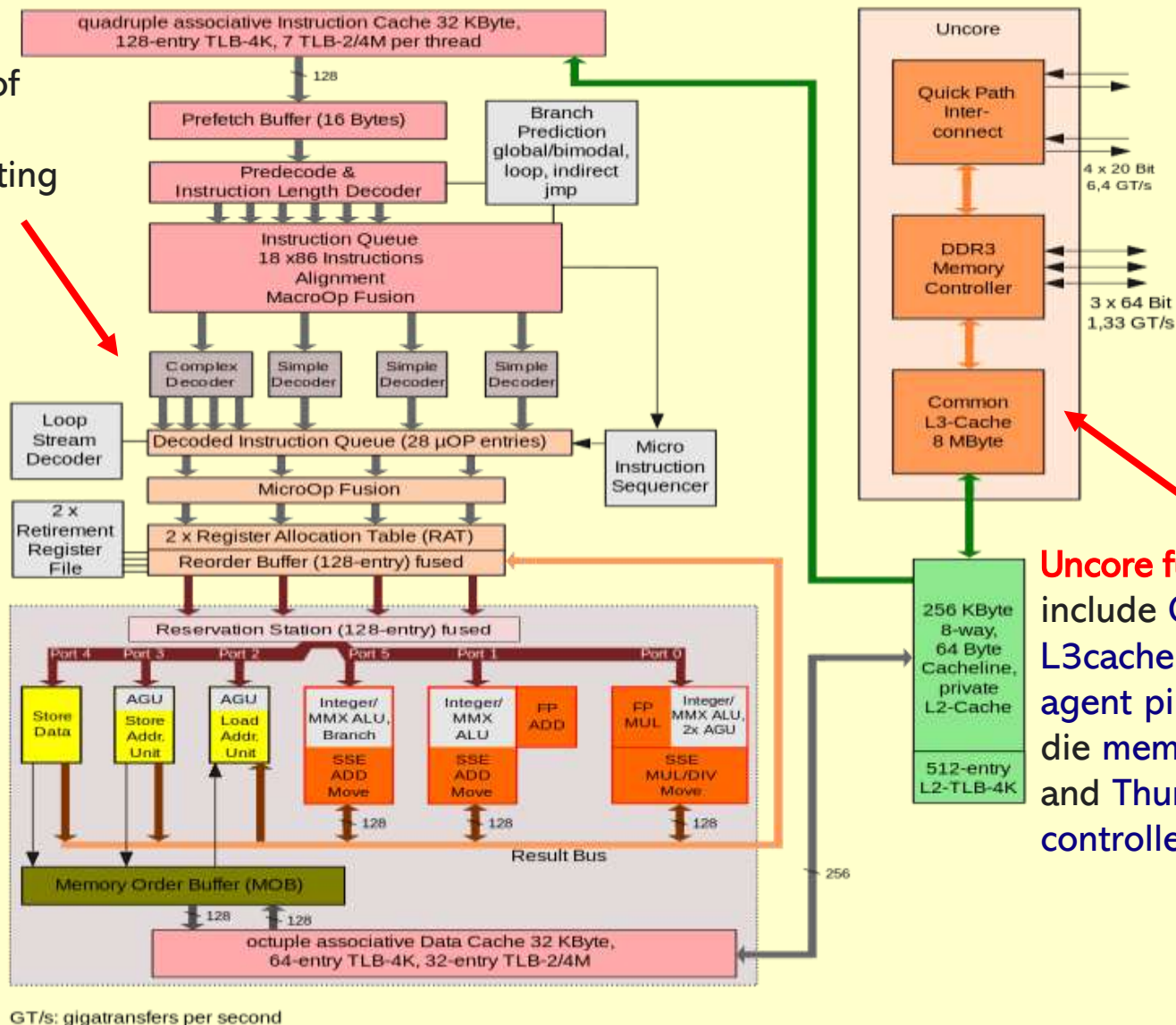


System Example



The Intel Quick Path Interconnect (QPI) is a point-to-point processor interconnect developed by Intel which replaced the front-side bus (FSB) in Xeon, Itanium, and certain desktop platforms. It increased the scalability and bandwidth available. Prior to the name's announcement, Intel referred to it as Common System Interface (CSI).

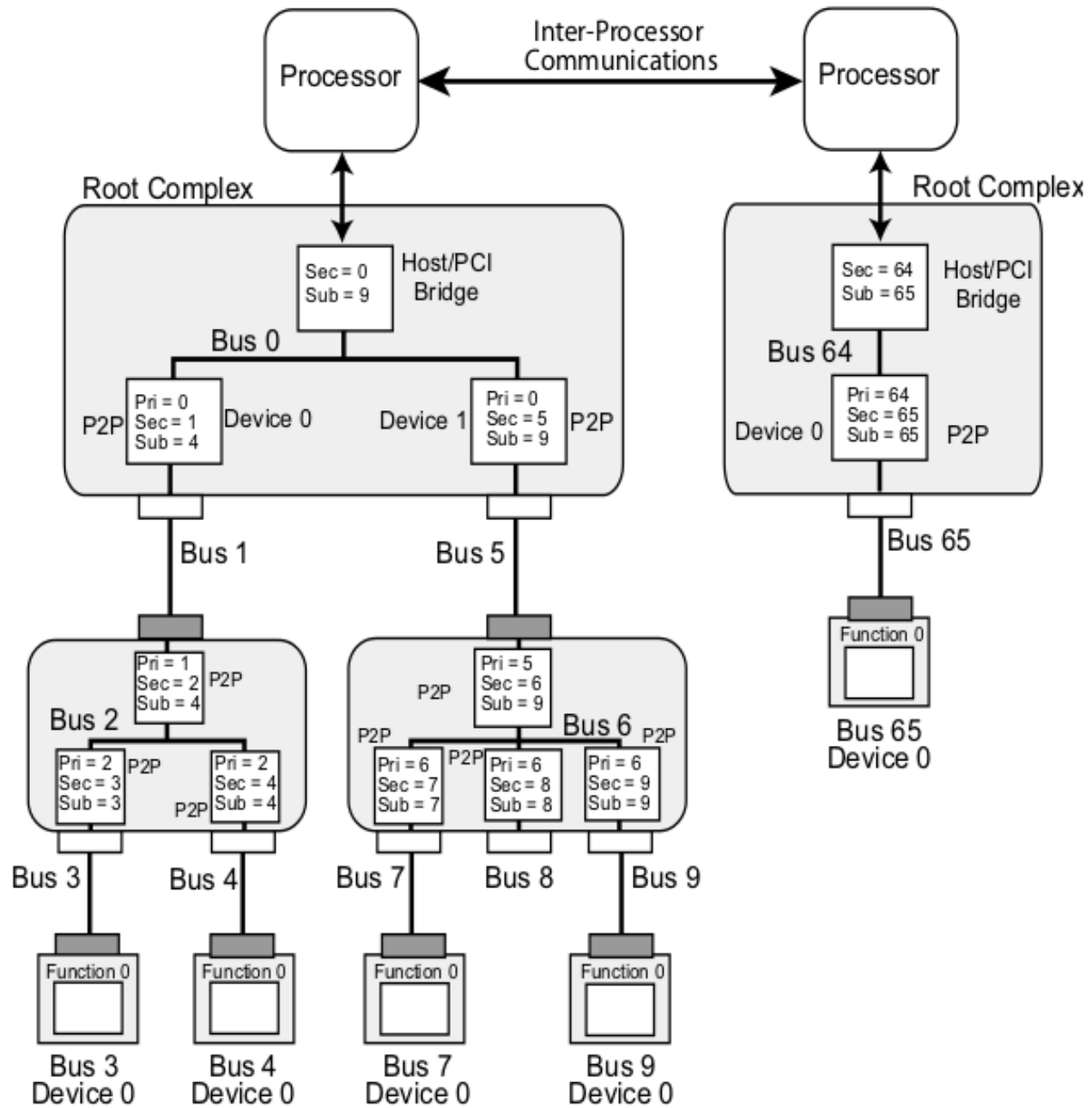
The core contains the components of the processor involved in executing instructions, including the ALU, FPU, L1 and L2 cache.



GT/s: gigatransfers per second

Uncore functions include QPI controllers, L3cache, snoop agent pipeline, on-die memory controller, and Thunderbolt controller.

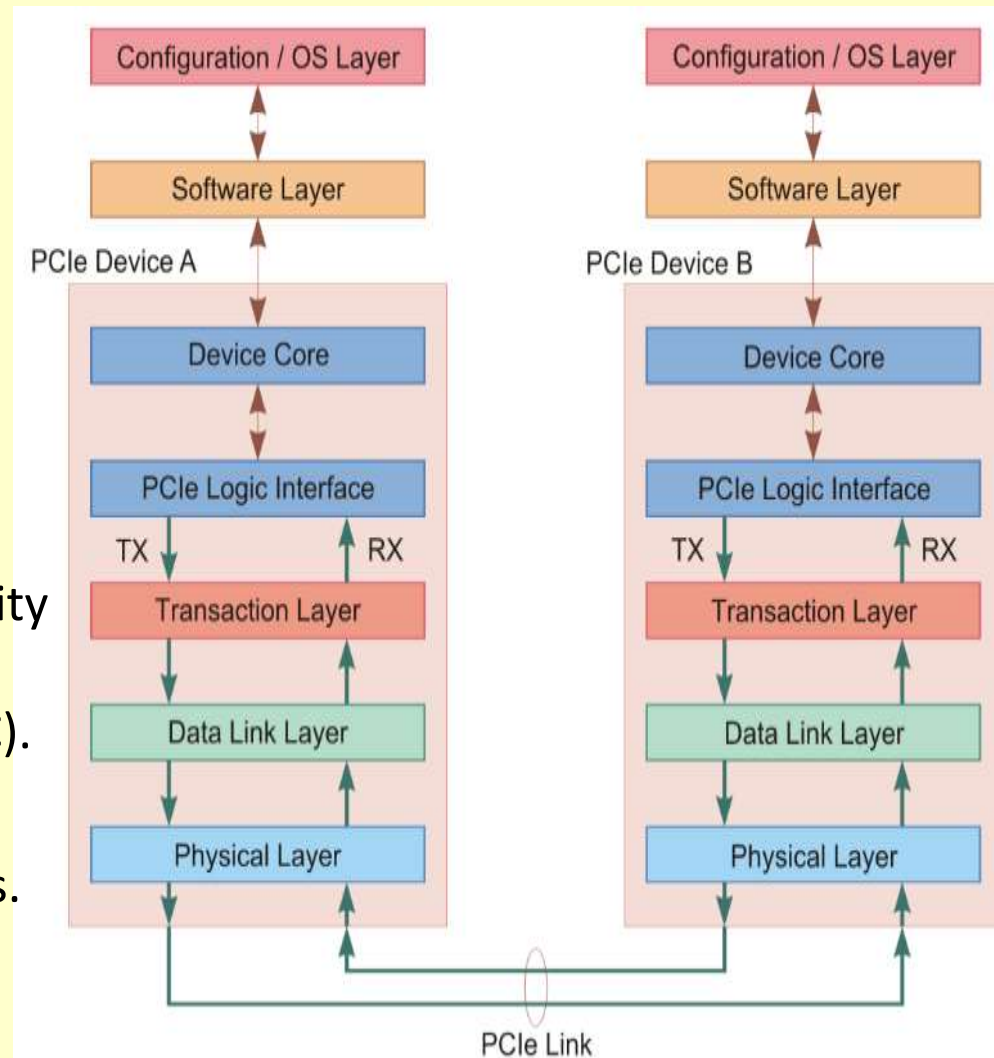
"Uncore" is a term used by Intel to describe the functions of a microprocessor that are not in the core, but which must be closely connected to the core to achieve high performance. It has been called "system agent" since the release of the Sandy Bridge microarchitecture.



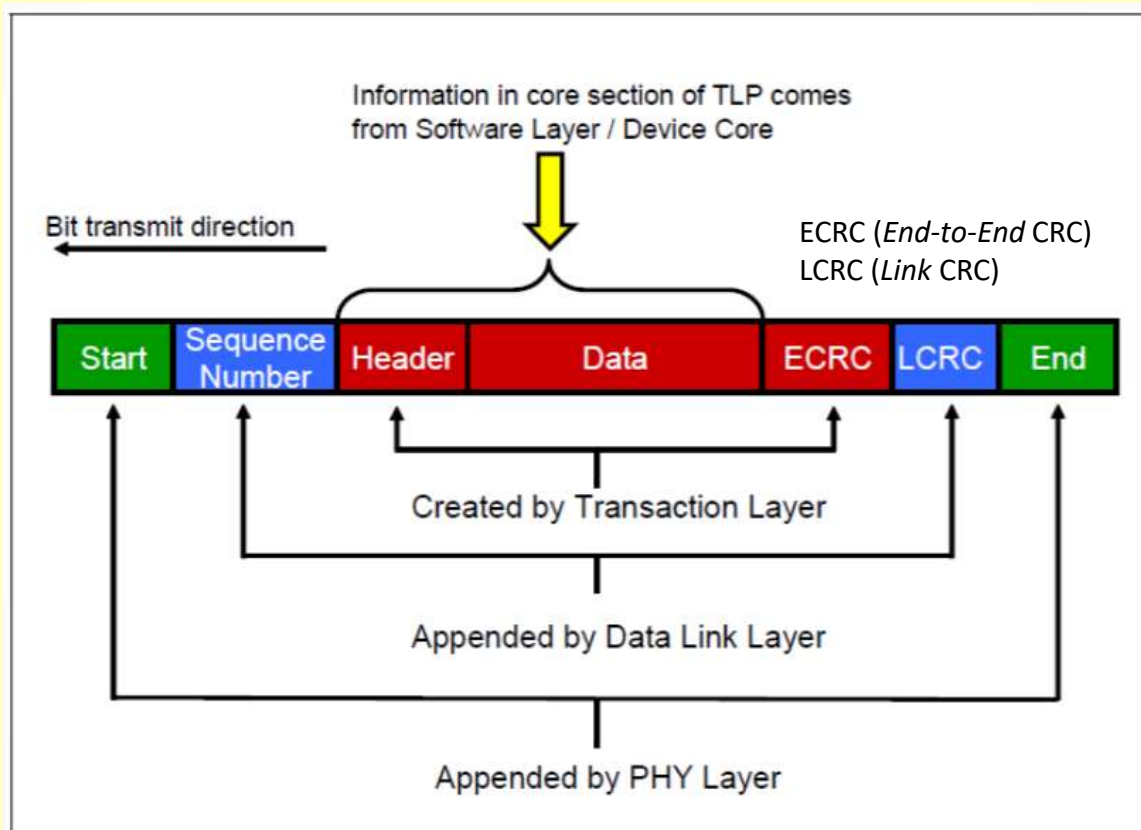
PCIe Architecture Layers

PCIe system may be structured **into five logical layers**:

- The **configuration/OS layer** manages the configuration of PCIe devices by the OS based on the Plug-and-Play specifications for initializing, enumerating, and configuring I/O devices.
- The **software layer** interacts with the OS through the same drivers as the conventional PCI bus.
- The **transaction layer** manages the transmission and reception of information using a packet-based protocol.
- The **data link layer** ensures the integrity of data transfers via error detection using a Cyclic Redundancy Check(CRC).
- The **physical layer** performs packet transmission over the PCIe serial links.
- *PCIe specification defines the architecture of PCIe devices in terms of three logical layers*



- The PCIe bus uses packets for transferring information between pairs of devices connected via a PCIe connection
- Packets are *formed in the transaction layer* based on information obtained from the device core and application and stored in a buffer
- *The data link layer* extends the packet with additional information required for error detection at a receiver device
- The packet is then *encoded in the physical layer* and transmitted through differential signals over the PCIe link



Packet flow through the logical layers of PCIe devices

PCIe Transaction Types

- **Transaction** is defined as *a series of one or more packet transmissions* required to accomplish a data transfer between an initiator and a target device
- *Requests* are translated to one of *four transaction types* by the Transaction Layer:
- **Memory Read or Memory Write.** Used to transfer data from/to a memory mapped location
- **I/O Read or I/O Write.** Used to transfer data from/to an I/O location
- **Configuration Read or Configuration Write.** Used to discover device capabilities, program features, and check status in the 4KB PCI Express configuration space.
- **Messages.** Used for event signaling and general purposes messaging. *Message transactions* are specific to the PCIe bus and are used for: *interrupt signaling, power management or error signaling.*

PCI Express transaction layer packet (TLP) types

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

Methods for Data Routing

- Each request or completion header is tagged as to its *type* and each of the packet types are routed based on one of three schemes.

- PCI Compatible Routing Methods

Address Routing

- Memory and I/O read/write
- Optional for messaging

ID Routing

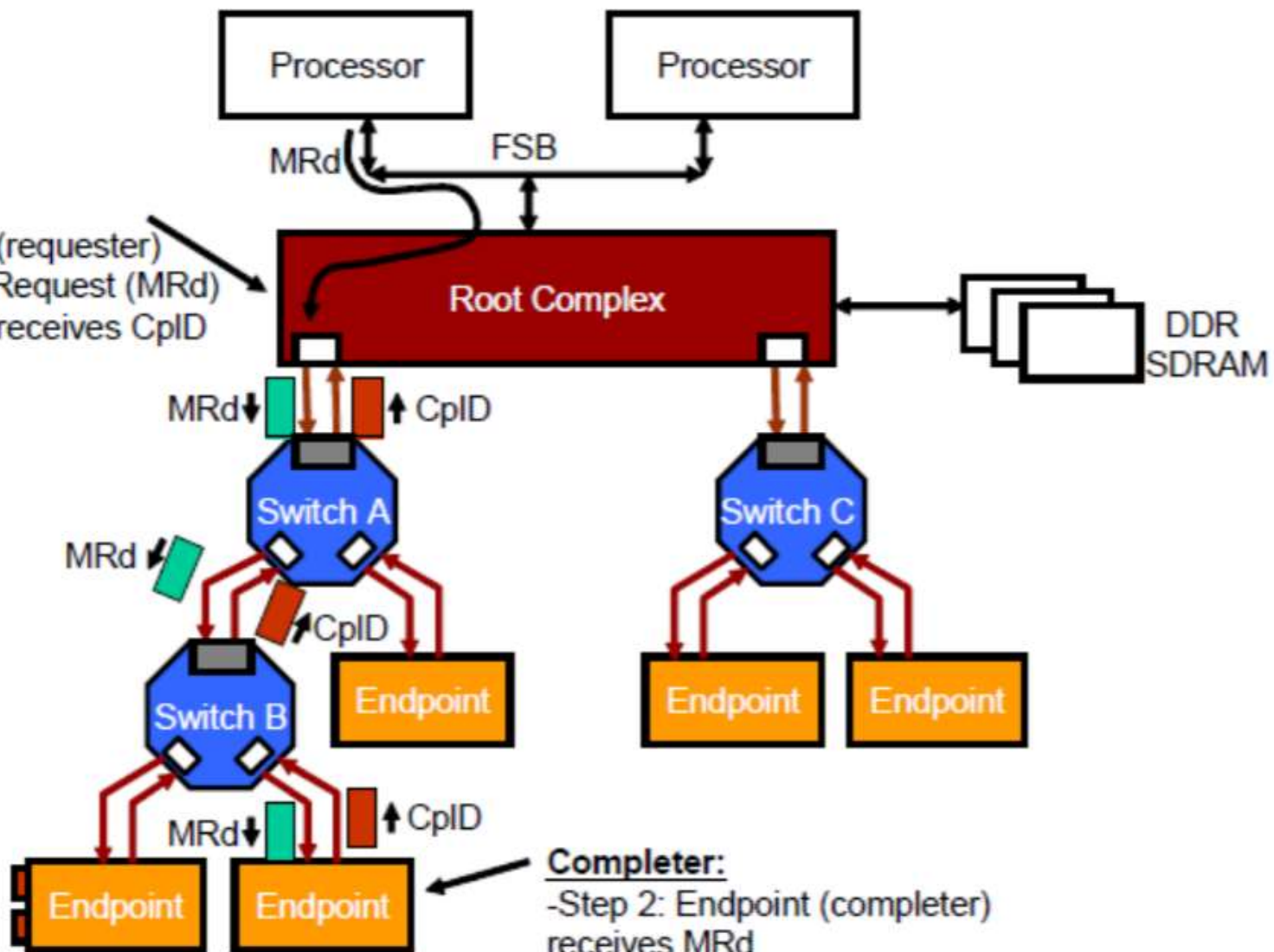
- Configuration read write
- Completions
- Optional for messaging

- PCI Express only routing methods

Implicit Routing

- Messaging
 - packets are routed based on a sub-field in the packet header.
 - Implicitly routed messages eliminates most of the sideband signals for interrupts, error handling, and power management.

CPU MRd targeting an Endpoint



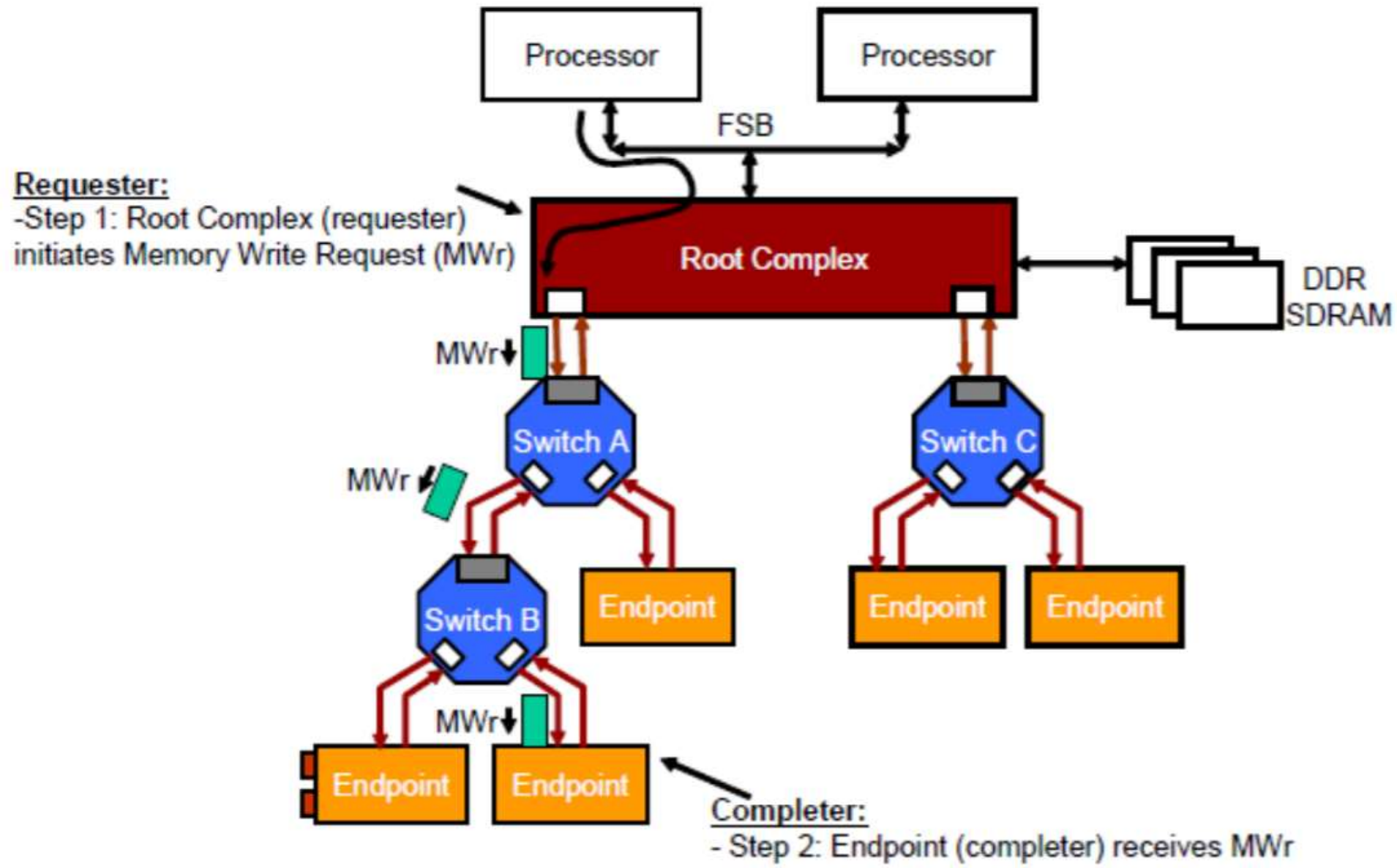
Requester:

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

Completer:

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

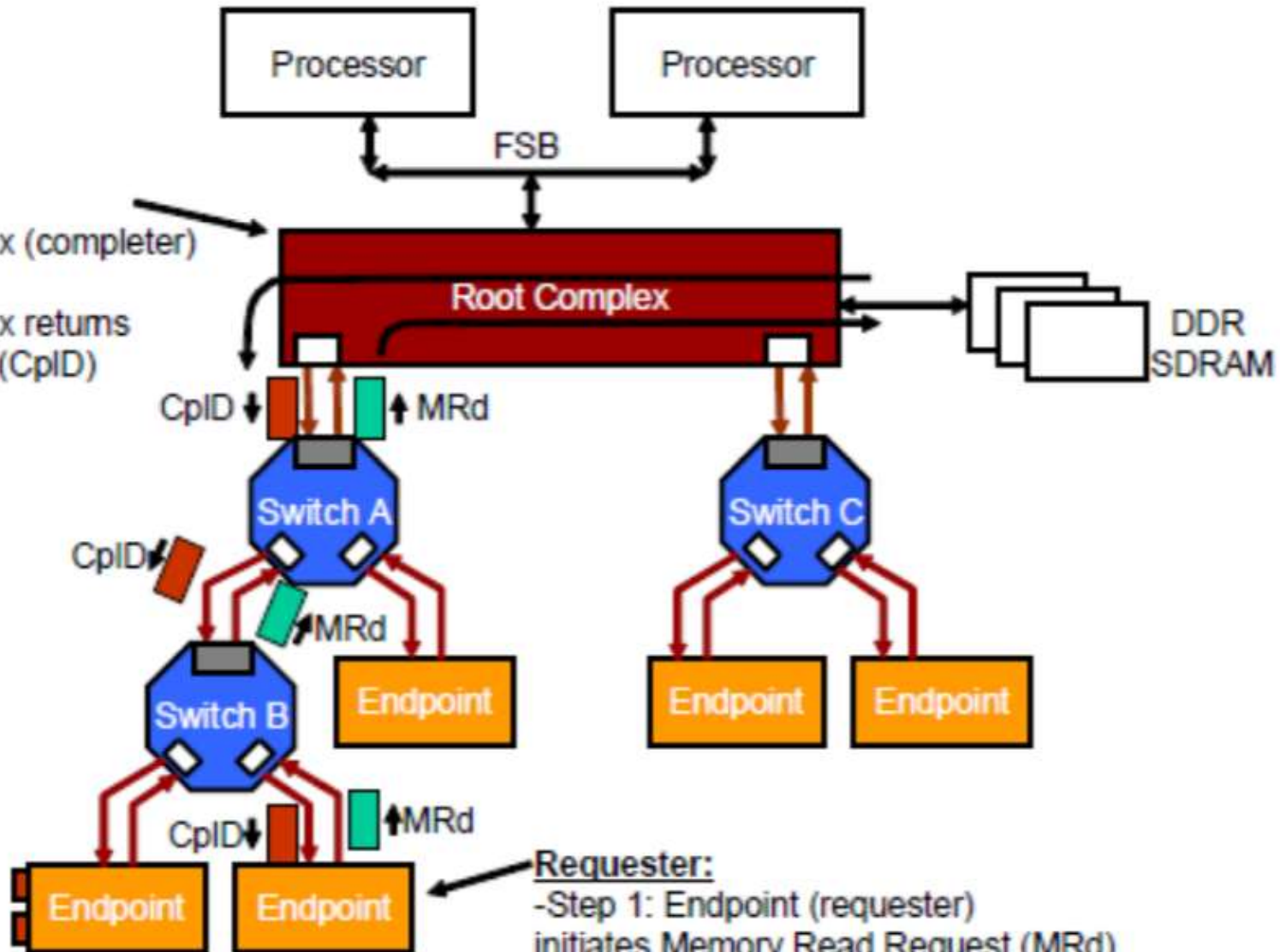
CPU MWr targeting Endpoint



Endpoint MRd targeting system memory

Completer:

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



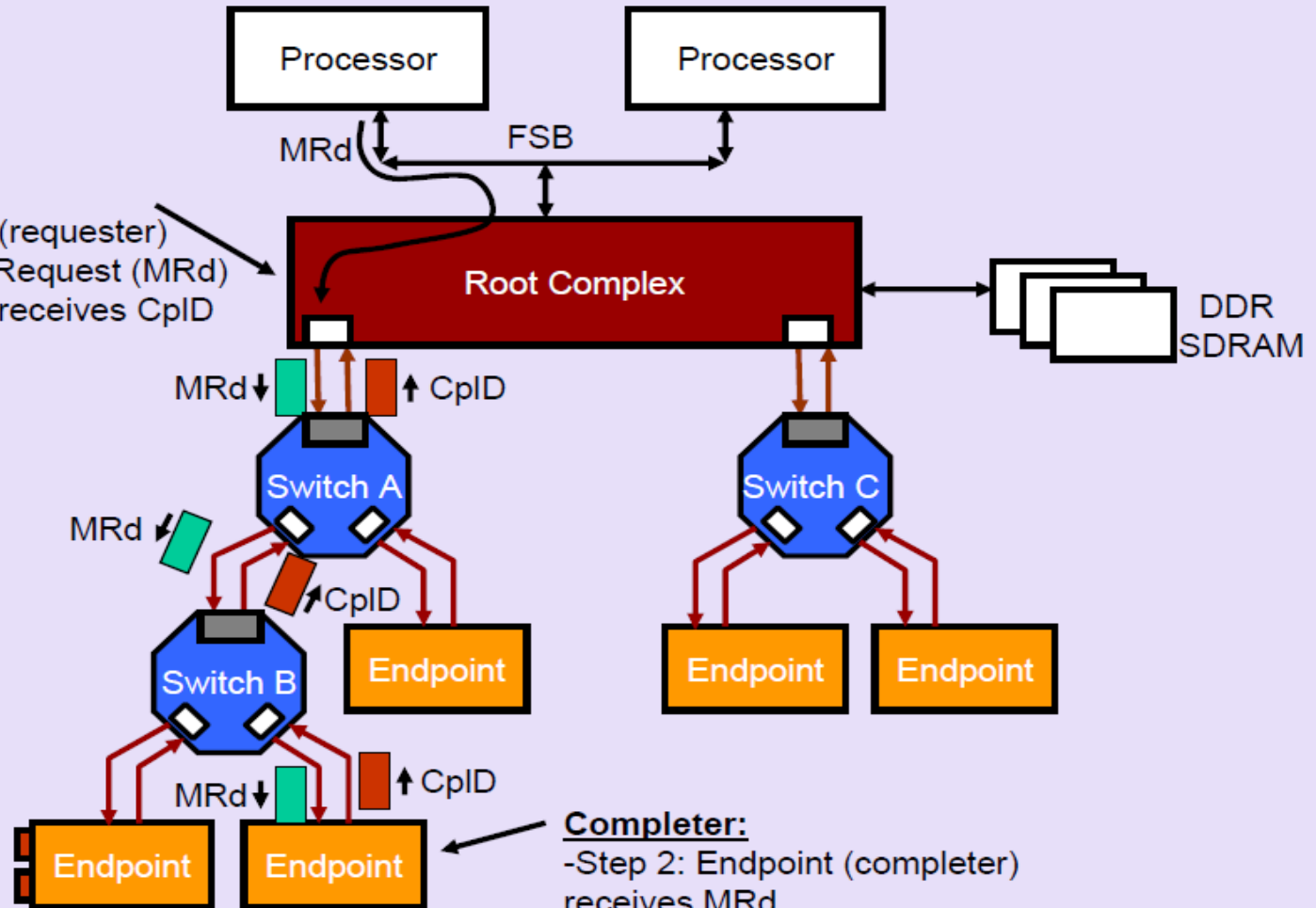
Requester:

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

Programmed I/O Transaction

Requester:

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



Bus Mastering (DMA)

- Until PCIe there was something intrusive in telling the CPU to withdraw from the bus during DMA
- On PCIe, it is much easier for any device to send read/write TLPs to the bus, just like Root Complex. This allows the device to directly access the processor memory (DMA) or exchange packets with other peripherals on a peer-to-peer basis (as long as switching entities accept this).

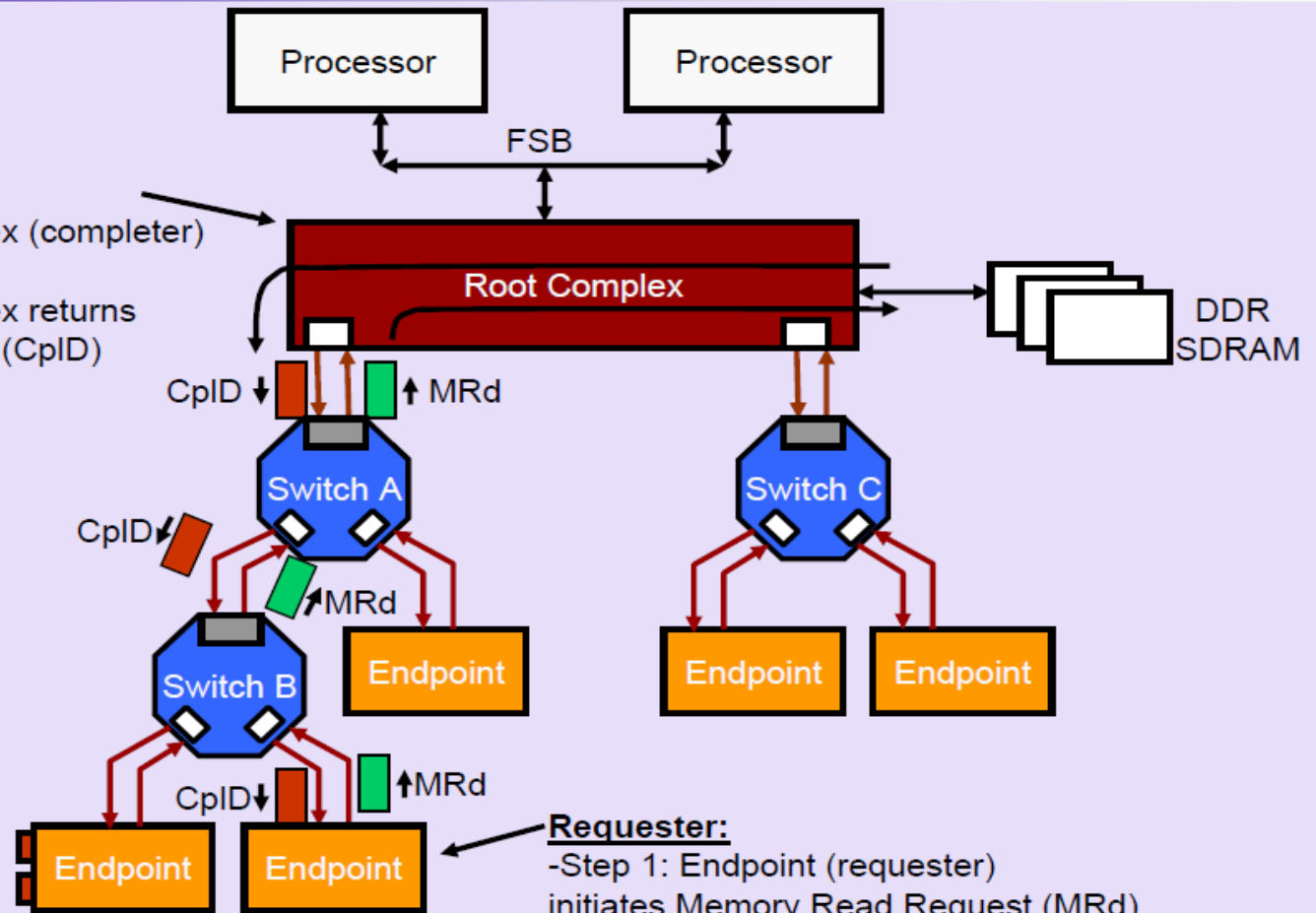
There are two things that need to happen first, as with any PCIe device:

1. The device must receive bus control by setting the "*Bus Master Enable*" bit in one of the *standard configuration registers*.
2. The software driver must inform the device about *the physical address of the relevant buffer*, most likely by writing in a mapped Base Address Register (configuration space).

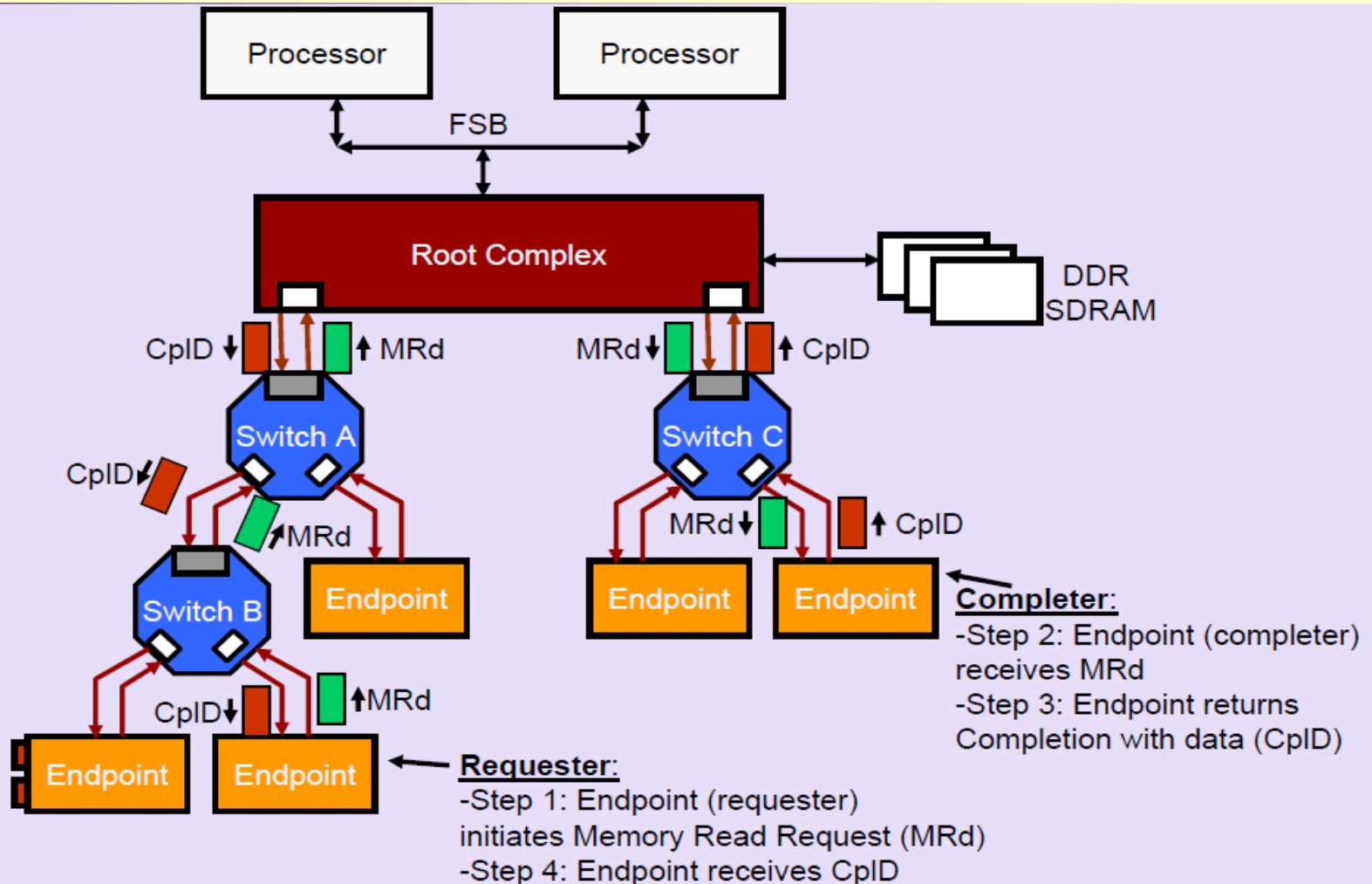
DMA Transaction

Completer:

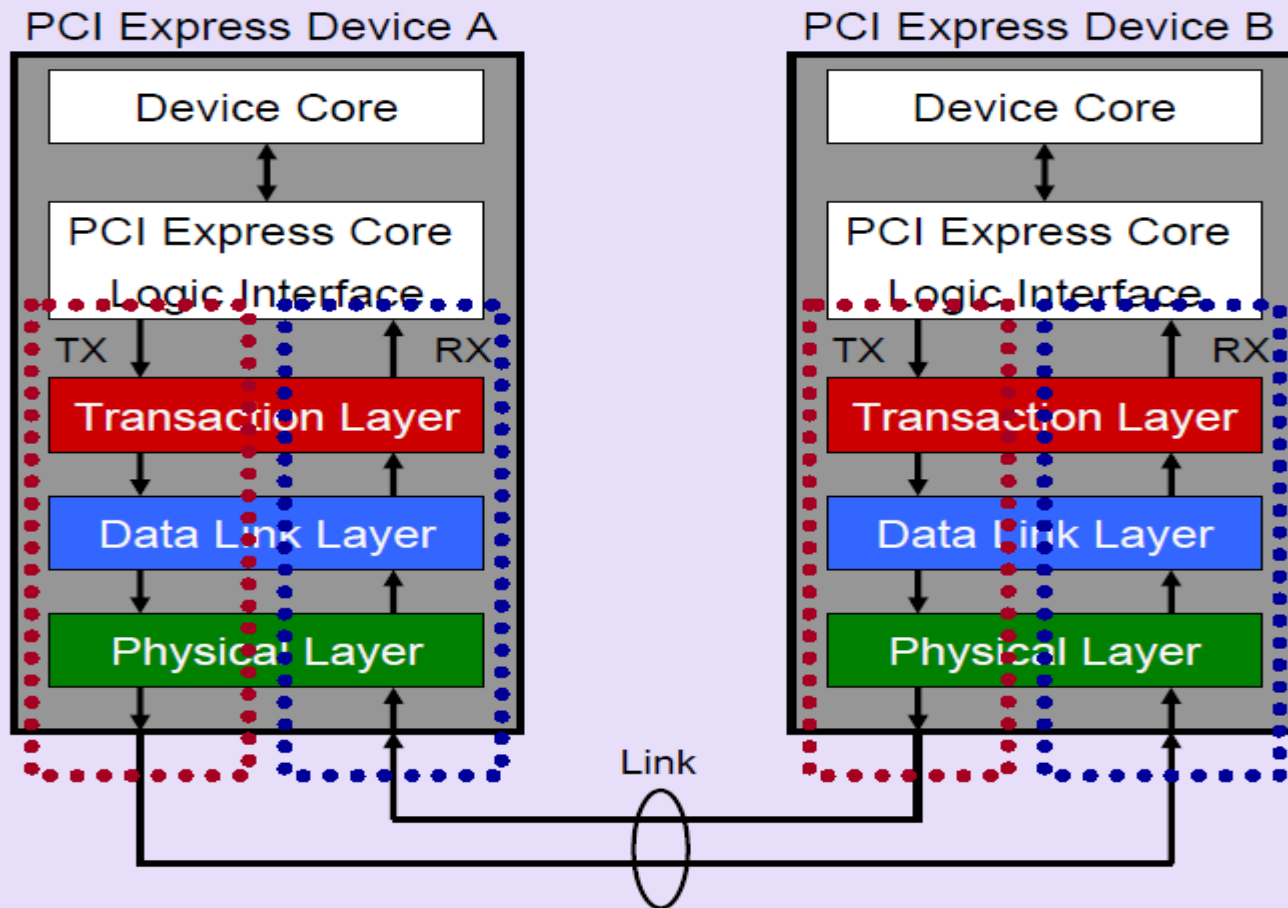
- Step 2: Root Complex (completer) receives MRd
- Step 3: Root Complex returns Completion with data (CpID)



Peer-to-Peer Transaction



PCI Express Device Layers



Interrupt Model: Three Methods

- PCI Express supports three interrupt reporting mechanisms:

1. Message Signaled Interrupts (MSI)

- Legacy endpoints are required to support MSI (or MSI-X) with 32- or 64-bits MSI capability register implementation
- Native PCI Express endpoints are required to support MSI with 64-bits MSI capability register implementation

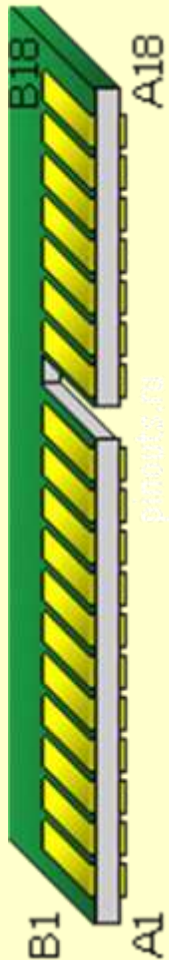
2. Message Signaled Interrupts - X (MSI-X)

- Legacy and native endpoints are required to support MSI-X (or MSI) and implement the associated MSI-X capability register

3. INTx Emulation.

- Native and Legacy endpoints are required to support Legacy INTx Emulation
- PCI Express defines in-band messages which emulate the four physical interrupt signals (INTA-INTD) routed between PCI devices and the system interrupt controller
- Forwarding support required by switches

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

- All PCI Express devices are required to support some combination of:
 - # Existing software written for generic PCI error handling, and which takes advantage of the fact that PCI Express has mapped many of its error conditions to existing PCI error handling mechanisms.
 - # Additional PCI Express-specific reporting mechanisms
- Errors are classified as ***correctable*** and ***uncorrectable***.
- ***Uncorrectable*** errors are further divided into:
 - # Fatal uncorrectable errors
 - # Non-fatal uncorrectable errors.

Correctable Errors

- Errors classified as correctable, degrade system performance, but recovery can occur with no loss of information

Hardware is responsible for recovery from a correctable error and no software intervention is required.

- Even though hardware handles the correction, logging the frequency of correctable errors may be useful if software is monitoring link operations.
- An example of a correctable error is the detection of a link CRC (LCRC) error when a TLP is sent, resulting in a Data Link Layer retry event.

Uncorrectable Errors

- Errors classified as uncorrectable impair the functionality of the interface and there is no specification mechanism to correct these errors
- The two subgroups are fatal and non-fatal
 - 1. Fatal Uncorrectable Errors:** Errors which render the link unreliable
 - First-level strategy for recovery may involve *a link reset by the system*
 - Handling of fatal errors is *platform-specific*
 - 2. Non-Fatal Uncorrectable Errors:** Uncorrectable errors associated with a particular transaction, while the link itself is reliable
 - Software may limit recovery strategy to the device(s) involved
 - Transactions between other devices are not affected

Benefits of PCI Express

For PC-based measurement and automation systems, the PCI bus has been the bus of choice for plug-in expansion boards for many years. It will continue to play an important role in the future. As the PC has evolved, the PCI bus (with its parallel architecture) has not scaled linearly with the rest of the platform. PCI Express answers these issues and provides benefits across five main areas:

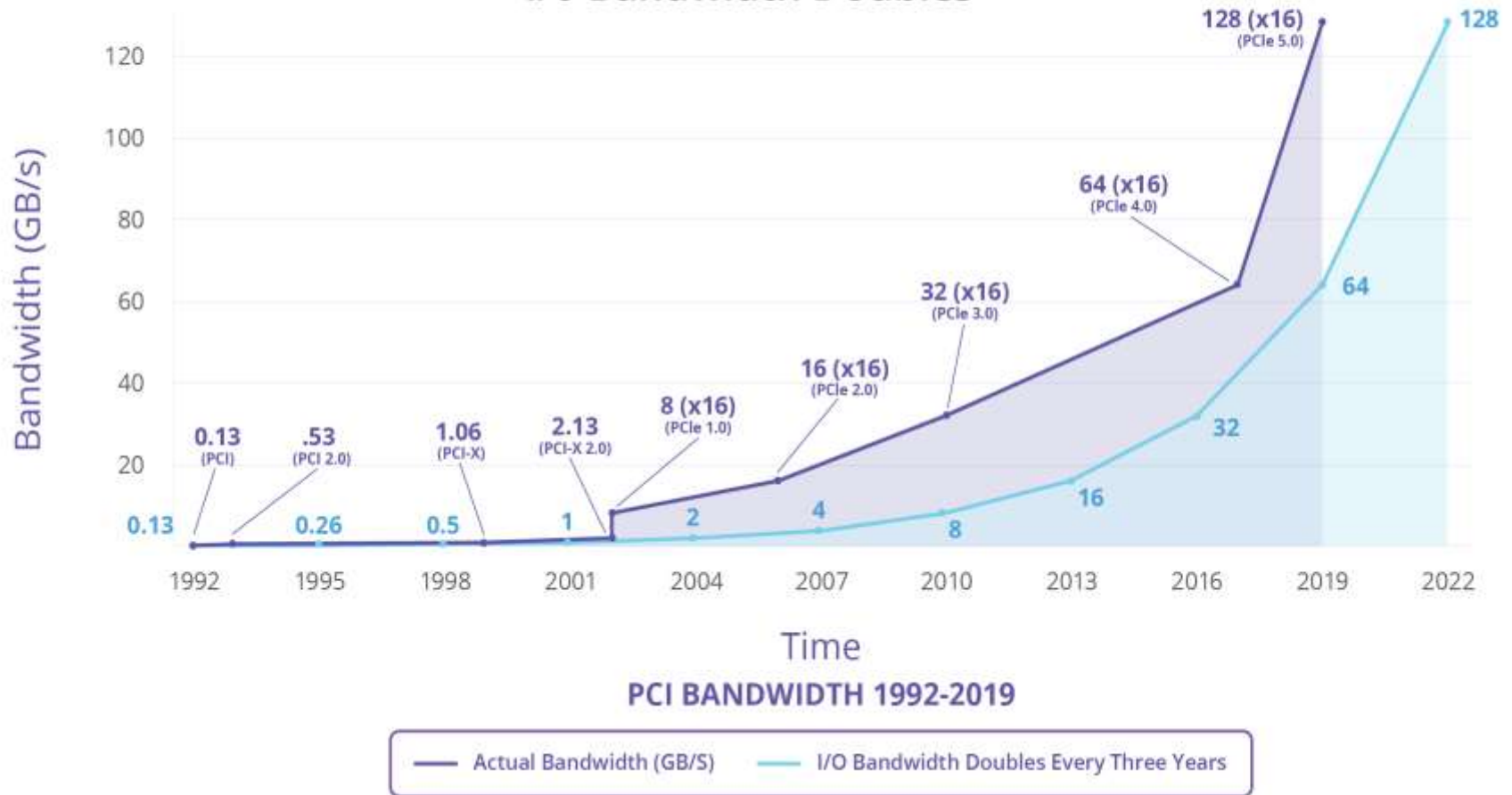
- * **High Performance** – relates specifically to bandwidth, which is more than double that of PCI in an x1 link and grows linearly as more lanes are added. An additional benefit that is not immediately evident is that this bandwidth is simultaneously available in both directions on each link. In addition, the initial signaling speed of 2.5 Gb/s is expected to increase, yielding further speed improvements.
- * **I/O Simplification** – relates to the streamlining of the plethora of both chip-to-chip and internal user accessible buses, such as AGP, PCI-X, and HubLink. This feature reduces the complexity of design and cost of implementation.
- * **Layered Architecture** – PCI Express establishes an architecture that can adapt to new technologies, while preserving software investment. Two key areas that benefit from the layered architectures are the physical layer, with increased signaling rates, and software compatibility.
- * **Next-Generation I/O** – PCI Express provides new capabilities for data acquisition and multimedia through isochronous data transfers. Isochronous transfers provide a type of quality of services (QOS) guarantee that ensures on-time data delivery through deterministic, time-dependent methods.
- * **Ease of Use** – PCI Express will greatly simplify how users add and upgrade systems. PCI Express offers both hot-swap and hot-plug. Because the hot-plug feature relies on specific OS features, it may lag the hardware launch. In addition, the variety of formats for PCI Express devices, especially SIOM and Express-Card, greatly increases the ability to add high-performance peripherals in servers and notebooks.

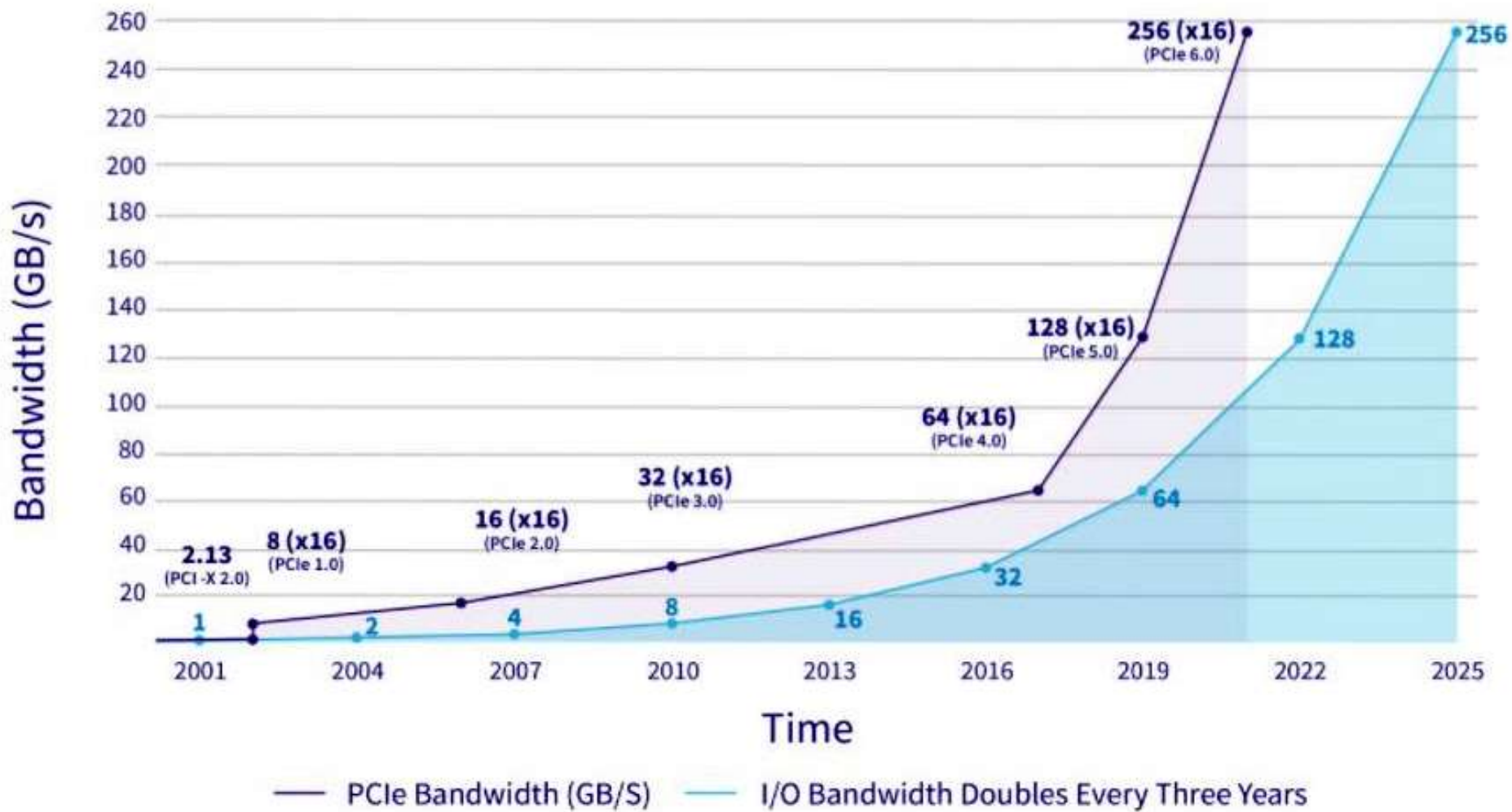
Evolution of PCIe



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

Summary

- Higher speed (5.0 GT/s), supported by:
 - Selectable de-emphasis levels
 - Selectable transmitter voltage range
- Dynamic speed and link width changes
 - Power savings, higher bandwidth, reliability
- Virtualization support
 - Access Control Services
- Other New Features
 - Completion timeout control
 - Function Level Reset
 - Modified Compliance Pattern for testing

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

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

<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>