



PCI Express® Basics & Background

Richard Solomon
LSI Corporation



Acknowledgements

Thanks are due to Ravi Budruk,
Mindshare, Inc. for much of the
material on PCI Express Basics

PCI Express Background

Revolutionary AND Evolutionary

■ PCI (1992/1993)

✓ Revolutionary

- Plug and Play jumperless configuration (BARs)
- Unprecedented bandwidth
 - 32-bit / 33MHz – 133MB/sec
 - 64-bit / 66MHz – 533MB/sec
- Designed from day 1 for bus-mastering adapters

✓ Evolutionary

- System BIOS maps devices then operating systems boot and run without further knowledge of PCI
- PCI-aware O/S could gain improved functionality
- PCI 2.1 (1995) doubled bandwidth with 66MHz mode

Revolutionary AND Evolutionary

■ PCI-X (1999)

✓ Revolutionary

- Unprecedented bandwidth
 - Up to 1066MB/sec with 64-bit / 133MHz
- Registered bus protocol
 - Eased electrical timing requirements
- Brought split transactions into PCI “world”

✓ Evolutionary

- PCI compatible at hardware *AND* software levels
- PCI-X 2.0 (2003) doubled bandwidth
 - 2133MB/sec at PCI-X 266 and 4266MB/sec at PCI-X 533

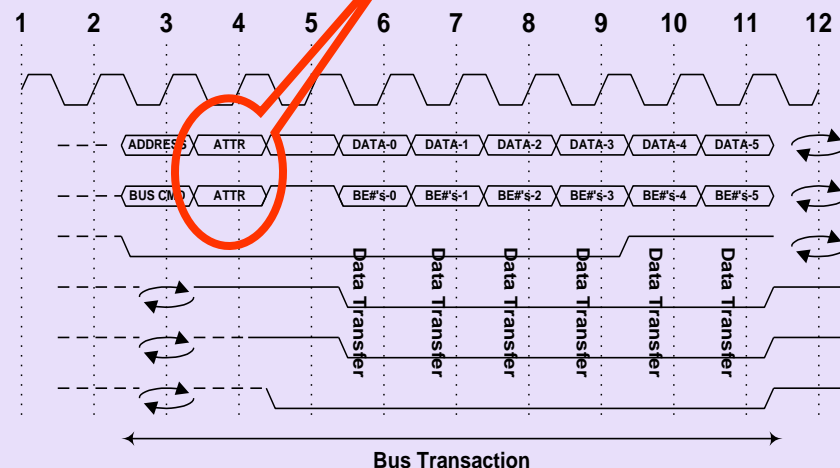
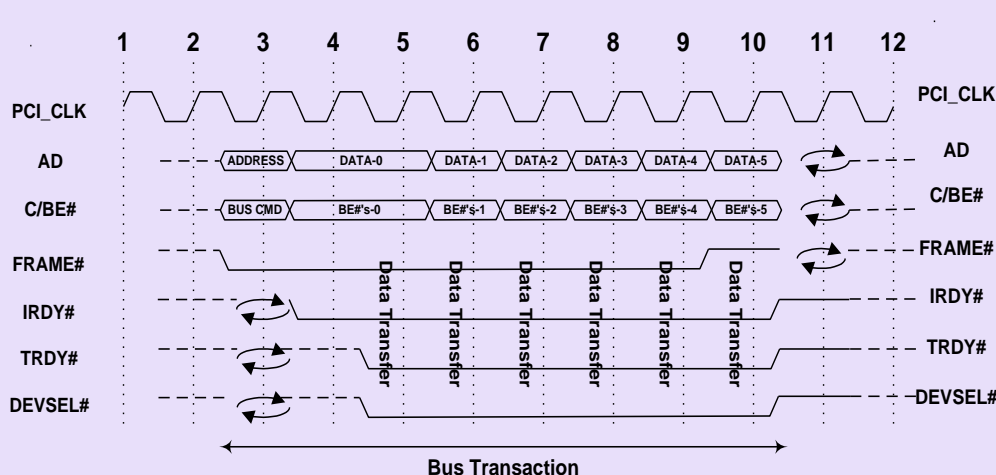
Revolutionary AND Evolutionary

- PCI Express – aka PCIe® (2002)
 - ✓ Revolutionary
 - Unprecedented bandwidth
 - x1: up to 1GB/sec in *EACH* direction
 - x16: up to 16GB/sec in *EACH* direction
 - “Relaxed” electricals due to serial bus architecture
 - Point-to-point, low voltage, dual simplex with embedded clocking
 - ✓ Evolutionary
 - PCI compatible at software level
 - Configuration space, Power Management, etc.
 - Of course, PCIe-aware O/S can get more functionality
 - Transaction layer familiar to PCI/PCI-X designers
 - System topology matches PCI/PCI-X
 - PCIe 2.0 (2006) doubled per-lane bandwidth: 250MB/s to 500MB/s
 - PCIe 3.0 (2010) doubled again to 1GB/s/lane!

Evolution – PCI to PCI-X

- Same bus and control signals
- Evolutionary protocol changes
- Clock frequency up to 133 MHz

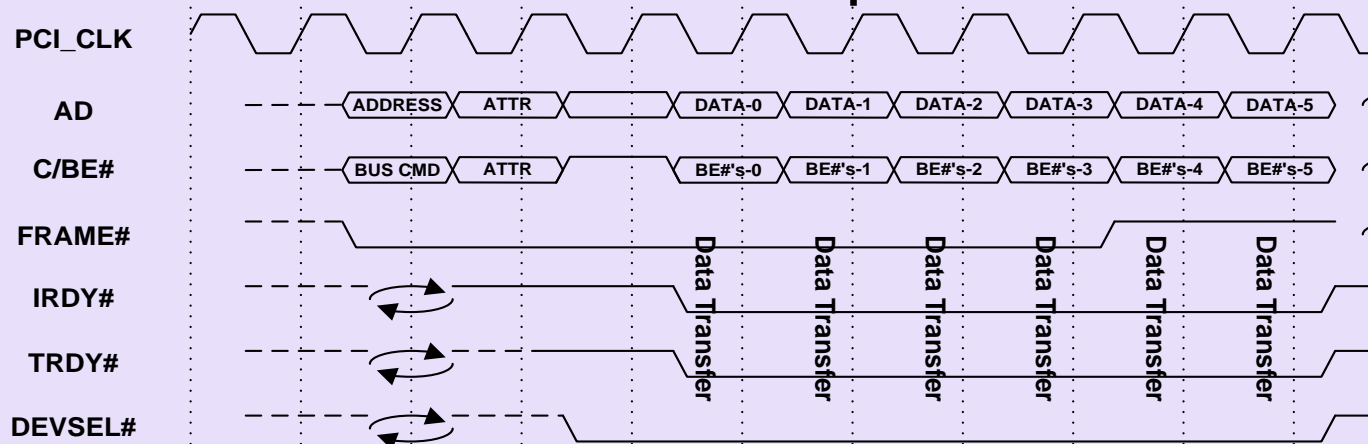
New “Attribute” phase for enhanced features



(Common clock)

Evolution – PCI-X to PCIe

■ PCI-X Address/Attribute phases:



■ Evolved into the PCIe Packet Header:

	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x	1	Type				R	TC		Reserved				T D	E P	Attr		R	Length												
Byte 4 >	Requester ID															Tag							Last DW BE				1st DW BE					
Byte 8 >	Address[63:32]																															
Byte 12 >	Address[31:2]																														R	

PCI Concepts

PCI Concepts

- PCI Express specifications are written assuming designers have these key background concepts from PCI
 - ✓ Address spaces
 - Memory
 - I/O
 - Configuration
 - ✓ Interrupts
 - “INTx” or “Legacy”
 - MSI
 - MSI-X
 - ✓ Split completions

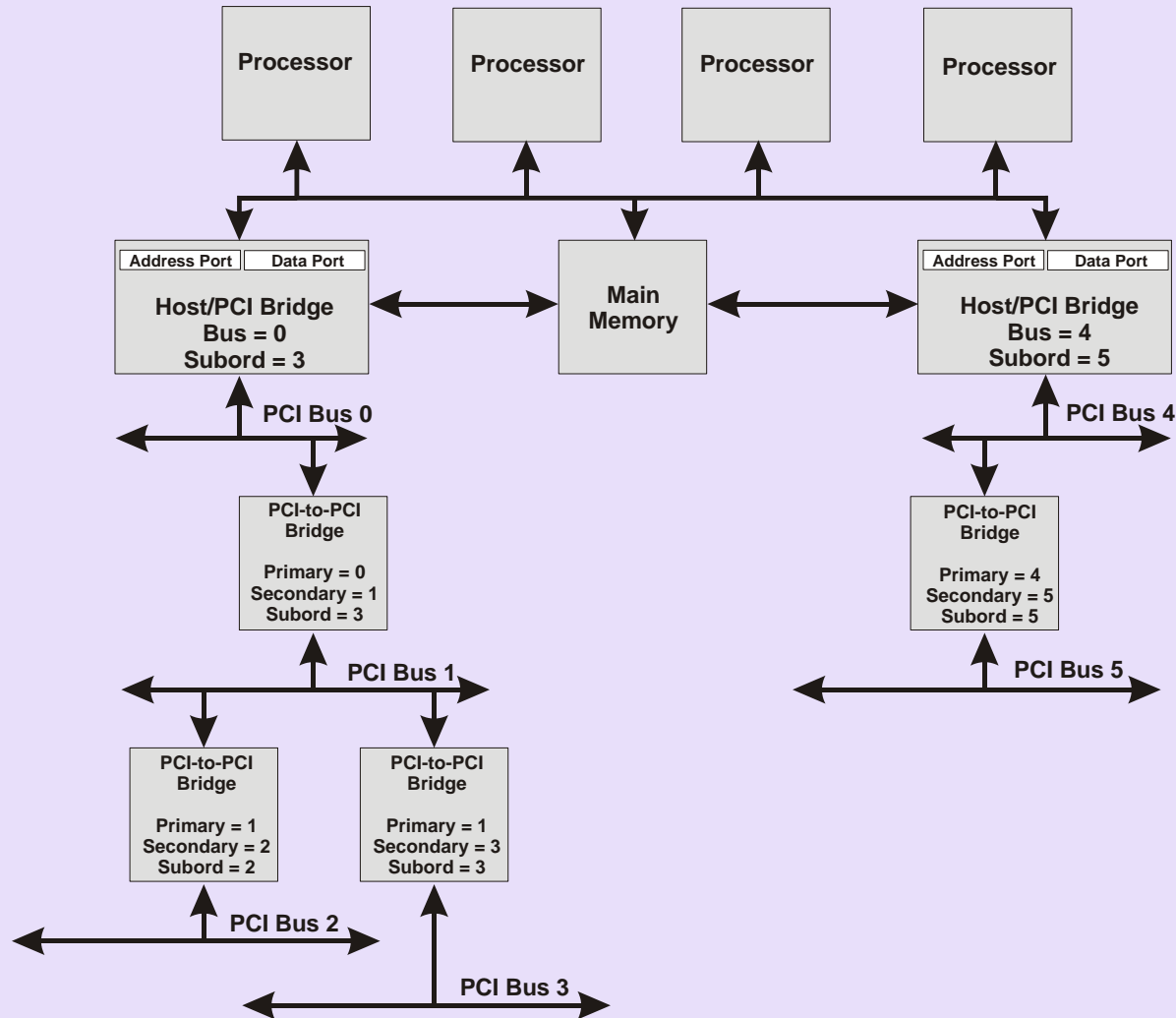
Address Spaces – Memory & I/O

- Memory space mapped cleanly to CPU semantics
 - ✓ 32-bits of address space initially
 - ✓ 64-bits introduced via Dual-Address Cycles (DAC)
 - Extra clock of address time on PCI/PCI-X
 - 4 DWORD header in PCI Express
 - ✓ Burstable
- I/O space mapped cleanly to CPU semantics
 - ✓ 32-bits of address space
 - Actually much larger than CPUs of the time
 - ✓ Non-burstable
 - Most PCI implementations didn't support
 - PCI-X codified
 - Carries forward to PCI Express

Address Spaces – Configuration

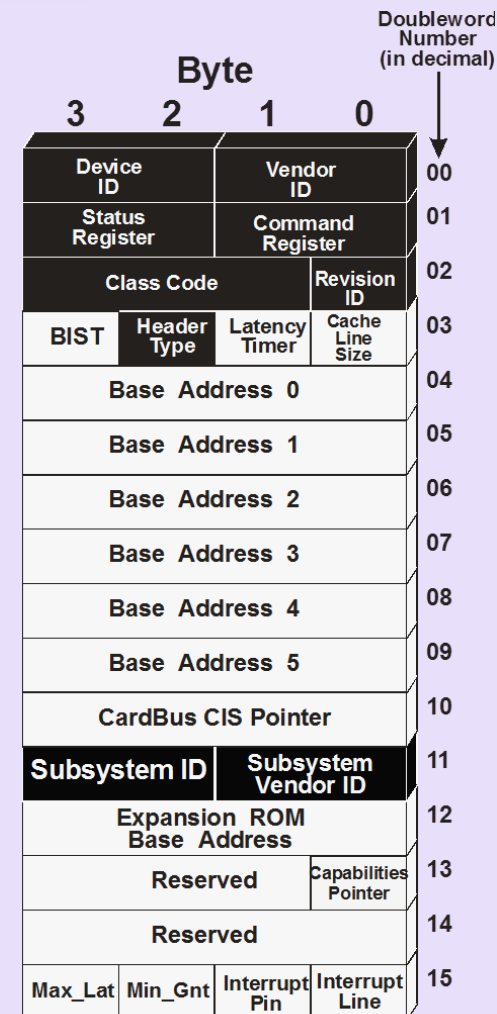
- Configuration space???
 - ✓ Allows control of devices' address decodes without conflict
 - ✓ No conceptual mapping to CPU address space
 - Memory-based access mechanisms in PCI-X and PCIe
 - ✓ Bus / Device / Function (aka BDF) form hierarchy-based address (PCIe 3.0 calls this “Routing ID”)
 - “Functions” allow multiple, logically independent agents in one physical device
 - E.g. combination SCSI + Ethernet device
 - 256 bytes or 4K bytes of configuration space per device
 - PCI/PCI-X bridges form hierarchy
 - PCIe switches form hierarchy
 - Look like PCI-PCI bridges to software
 - ✓ “Type 0” and “Type 1” configuration cycles
 - Type 0: to same bus segment
 - Type 1: to another bus segment

Configuration Space (cont'd)



Configuration Space

- Device Identification
 - ✓ VendorID: PCI-SIG assigned
 - ✓ DeviceID: Vendor self-assigned
 - ✓ Subsystem VendorID: PCI-SIG
 - ✓ Subsystem DeviceID: Vendor
- Address Decode controls
 - ✓ Software reads/writes BARs to determine required size and maps appropriately
 - ✓ Memory, I/O, and bus-master enables
- Other bus-oriented controls



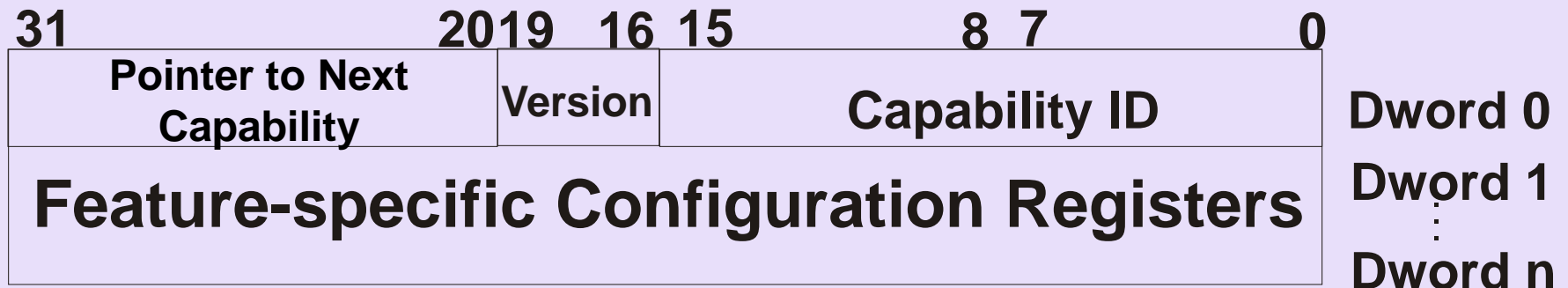
Configuration Space – Capabilities List

- Linked list
 - ✓ Follow the list! Cannot assume fixed location of any given feature in any given device
 - ✓ Features defined in their related specs:
 - PCI-X
 - PCIe
 - PCI Power Management
 - Etc.



Configuration Space – Extended Capabilities List

- PCI Express only
- Linked list
 - ✓ Follow the list! Cannot assume fixed location of any given feature in any given device
 - ✓ First entry in list is *always* at 100h
 - ✓ Features defined in PCI Express specification



Interrupts

- PCI introduced INTA#, INTB#, INTC#, INTD# - collectively referred to as INTx
 - ✓ Level sensitive
 - ✓ Decoupled device from CPU interrupt
 - ✓ System controlled INTx to CPU interrupt mapping
 - ✓ Configuration registers
 - report A/B/C/D
 - programmed with CPU interrupt number
- PCI Express mimics this via “virtual wire” messages
 - ✓ Assert_INTx and Deassert_INTx

What are MSI and MSI-X?

- Memory Write replaces previous interrupt semantics
 - ✓ PCI and PCI-X devices stop asserting INTA/B/C/D and PCI Express devices stop sending Assert_INTx messages once MSI or MSI-X mode is enabled
 - ✓ MSI uses one address with a variable data value indicating which “vector” is asserting
 - ✓ MSI-X uses a table of independent address and data pairs for each “vector”
- NOTE: *Boot devices* and any device intended for a non-MSI operating system generally must still support the appropriate INTx signaling!

Split Transactions – Background

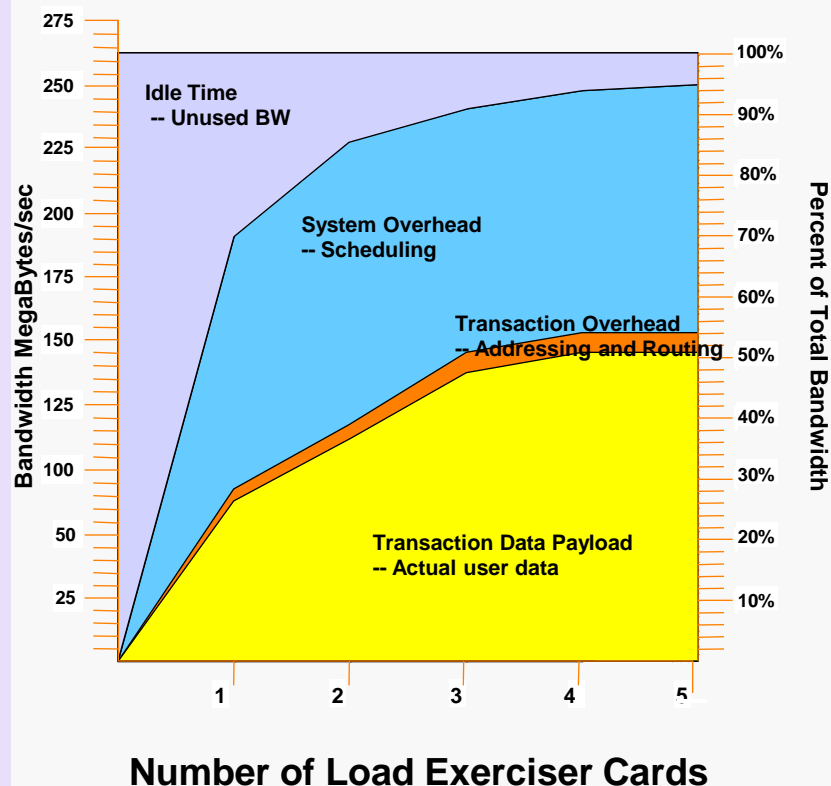
- PCI commands contained no length
 - ✓ Bus allowed disconnects and retries
 - ✓ Difficult data management for target device
 - Writes overflow buffers
 - Reads require pre-fetch
 - How much to pre-fetch? When to discard? Prevent stale data?
- PCI commands contained no initiator information
 - ✓ No way for target device to begin communication with the initiator
 - ✓ Peer-to-peer requires knowledge of system-assigned addresses

Split Transactions

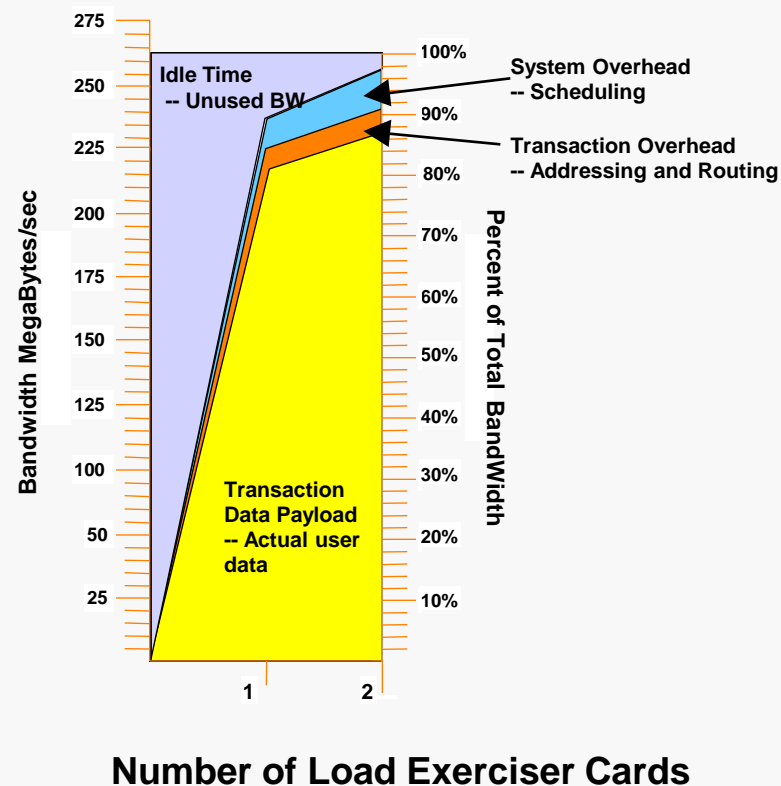
- PCI-X commands added length and Routing ID of initiator
 - ✓ Writes: allow target device to allocate buffers
 - ✓ Reads: Pre-fetch now deterministic
- PCI-X retains “retry” & “disconnect”, adds “split”
- Telephone analogy
 - ✓ Retry: “I’m busy go away”
 - Delayed transactions are complicated
 - ✓ Split: “I’ll call you back”
 - Simple
 - More efficient

Benefits of Split Transactions

Bandwidth Usage with Conventional PCI Protocols



Bandwidth Usage with PCI-X Enhancements

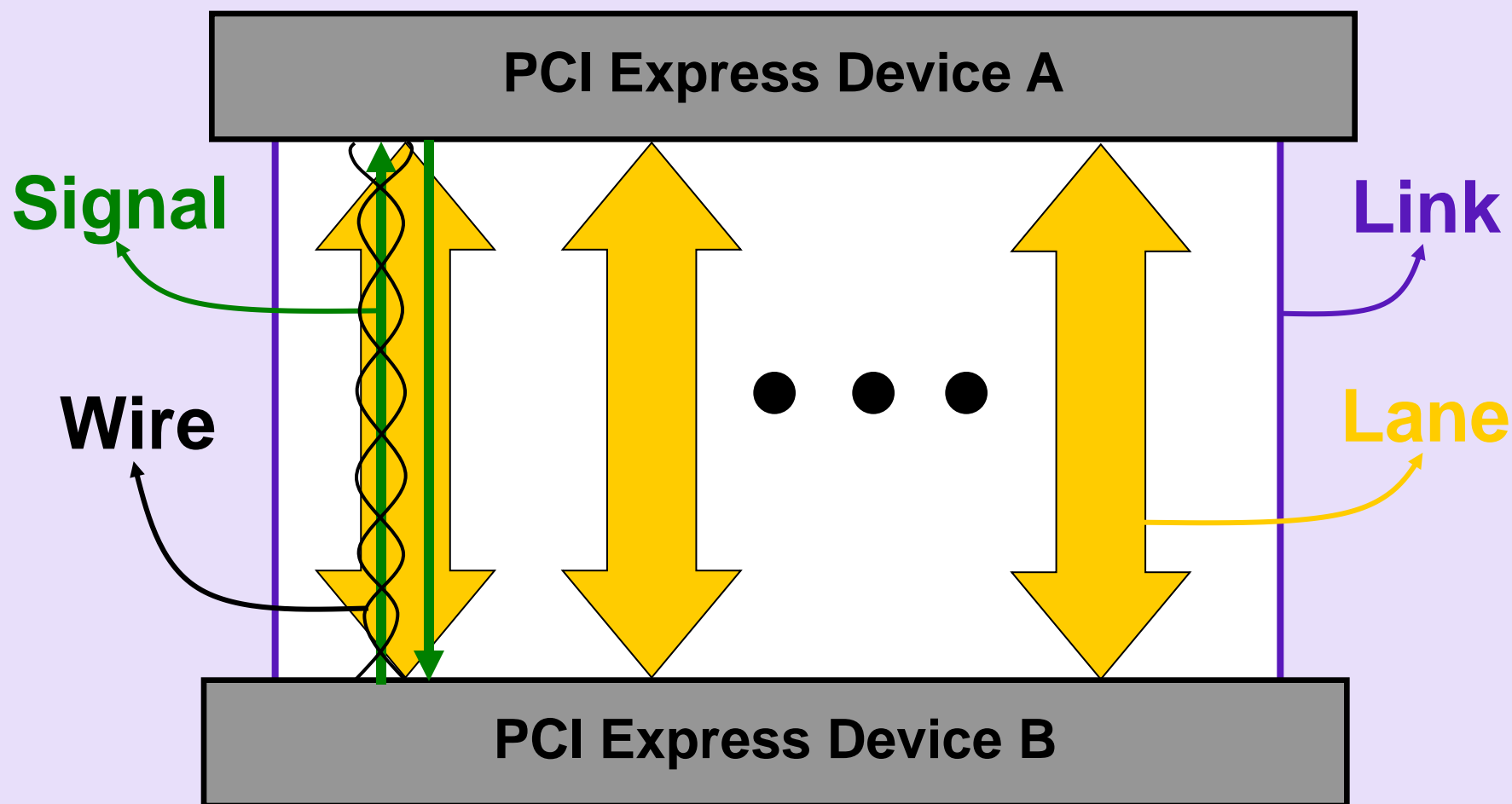


PCI Express Basics

PCI Express Introduction

- PCI Express architecture is a high performance, IO interconnect for peripherals in computing/communication platforms
- Evolved from PCITM and PCI-XTM architectures
 - ✓ Yet PCI Express architecture is significantly different from its predecessors PCI and PCI-X
- PCI Express is a serial point-to-point interconnect between two devices
- Implements packet based protocol for information transfer
- Scalable performance based on number of signal Lanes implemented on the PCI Express interconnect

PCI Express Terminology



PCI Express Throughput

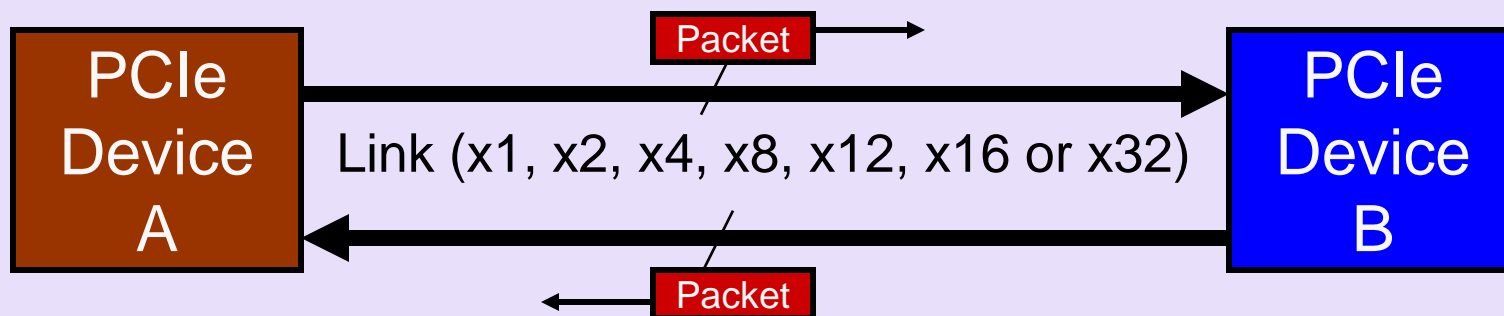
Bandwidth (GB/s)	Link Width						
	x1	x2	x4	x8	x12	x16	x32
PCIe 1.x “2.5 GT/s”	0.5	1	2	4	6	8	16
PCIe 2.x “5 GT/s”	1	2	4	8	12	16	32
PCIe 3.0 “8 GT/s”	2	4	8	16	24	32	64

Derivation of these numbers:

- 2.5 GT/s (PCIe 1.x), 5.0 GT/s (PCIe 2.x), or 8GT/s (PCIe 3.0) signaling in each direction
- 20% overhead due to 8b/10b encoding in 1.x and 2.x
- “Aggregate bandwidth”, implying traffic in both directions

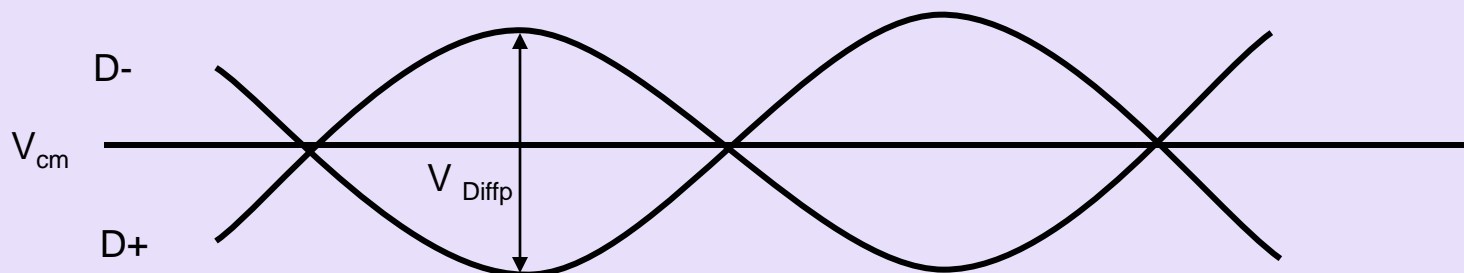
PCI Express Features

- Point-to-point connection
- Serial bus means fewer pins
- Scalable: x1, x2, x4, x8, x12, x16, x32
- Dual Simplex connection
- 2.5, 5.0 and 8.0 GT/s transfer/direction/s
- Packet based transaction protocol



Differential Signaling

- Electrical characteristics of PCI Express signal
 - ✓ Differential signaling
 - Transmitter Differential Peak voltage = 0.4 - 0.6 V
 - Transmitter Common mode voltage = 0 - 3.6 V



- Two devices at opposite ends of a Link may support different DC common mode voltages

Additional Features

- Switches used to interconnect multiple devices
- Packet based protocol
- Bandwidth and clocking
- Same memory, IO and configuration address space as PCI
 - ✓ Similar transaction types as PCI with additional message transaction
- PCI Express Transactions include:
 - ✓ memory read/write, memory read lock, IO read/write, configuration read/write, message requests
- Split transaction model for non-posted

Additional Features

- Data Integrity and Error Handling
 - ✓ RAS capable (Reliable, Available, Serviceable)
 - ✓ Data integrity at: 1) Link level, 2) end-to-end
- Virtual channels (VCs) and traffic classes (TCs) to support differentiated traffic or Quality of Service (QoS)
 - ✓ The ability to define levels of performance for packets of different TCs
 - ✓ 8 TC's and 8 VC's available

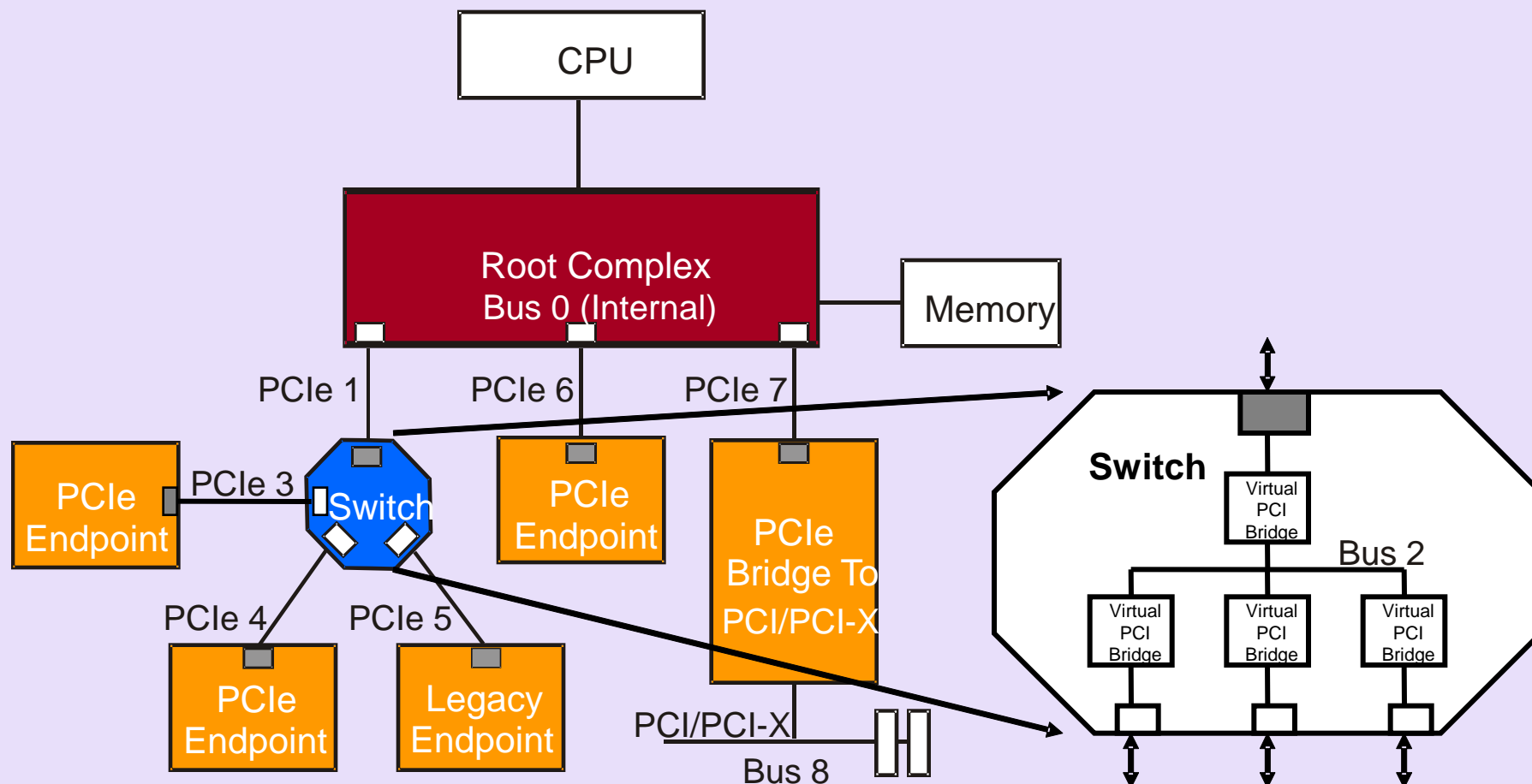
Additional Features

- Flow Control
 - ✓ No retry as in PCI
- MSI style interrupt handling
 - ✓ Also supports legacy PCI interrupt handling in-band
- Advanced power management
 - ✓ Active State PM
 - ✓ PCI compatible PM

Additional Features

- Hot Plug and Hot Swap support
 - ✓ Native
 - ✓ No sideband signals
- PCI compatible software model
 - ✓ PCI configuration and enumeration software can be used to enumerate PCI Express hardware
 - ✓ PCI Express system will boot existing OS
 - ✓ PCI Express supports existing device drivers
 - ✓ New additional configuration address space requires OS and driver update

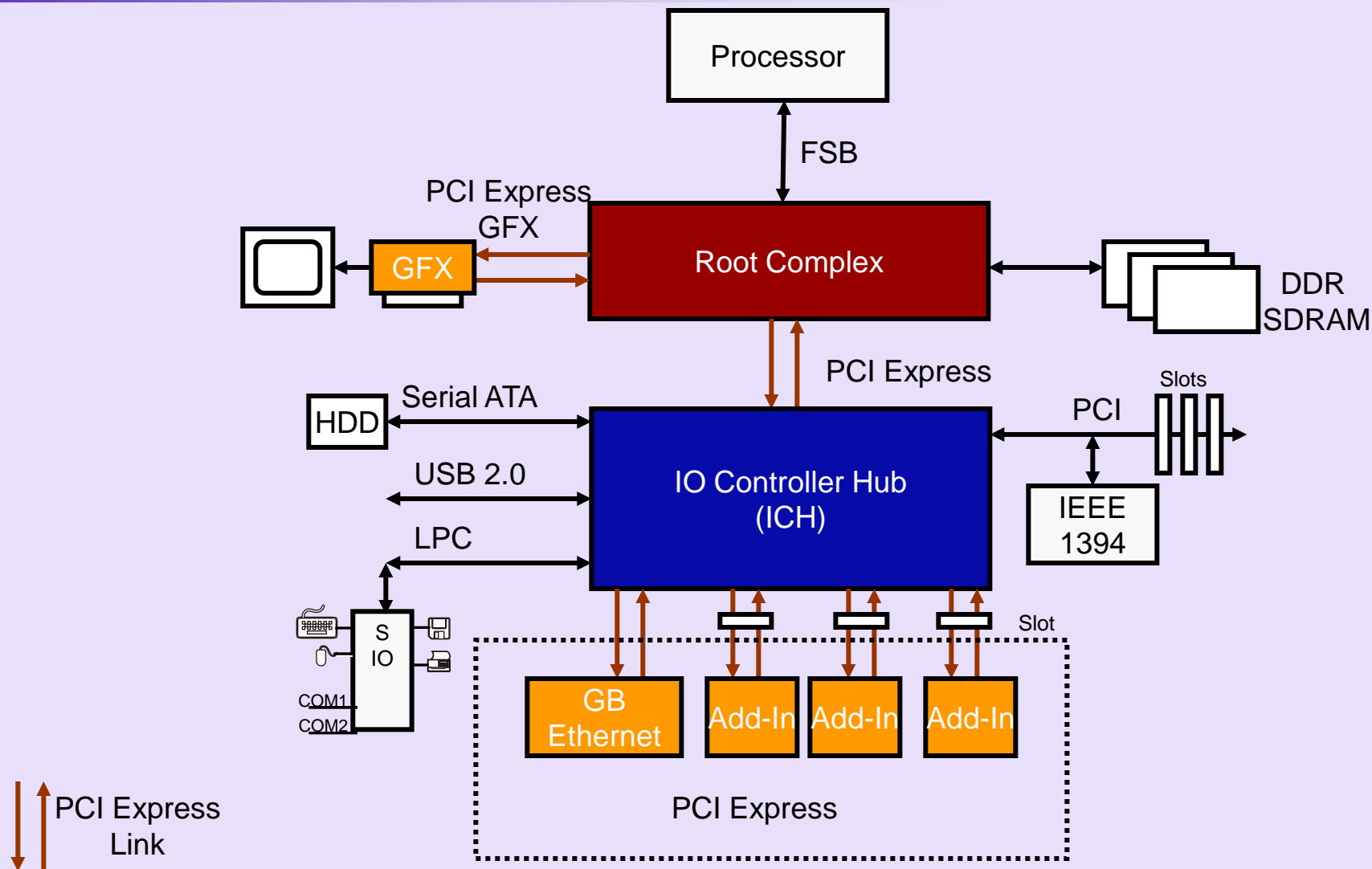
PCI Express Topology



Legend

- PCI Express Device Downstream Port
- PCI Express Device Upstream Port

PCI Express System



Transaction Types, Address Spaces

- Request are translated to one of four transaction types by the Transaction Layer:
 1. **Memory Read or Memory Write.** Used to transfer data from or to a memory mapped location.
 - The protocol also supports a *locked memory read* transaction variant
 2. **I/O Read or I/O Write.** Used to transfer data from or to an I/O location.
 - These transactions are restricted to supporting legacy endpoint devices
 3. **Configuration Read or Configuration Write.** Used to discover device capabilities, program features, and check status in the 4KB PCI Express configuration space.
 4. **Messages.** Handled like posted writes. Used for event signaling and general purpose messaging.

PCI Express 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

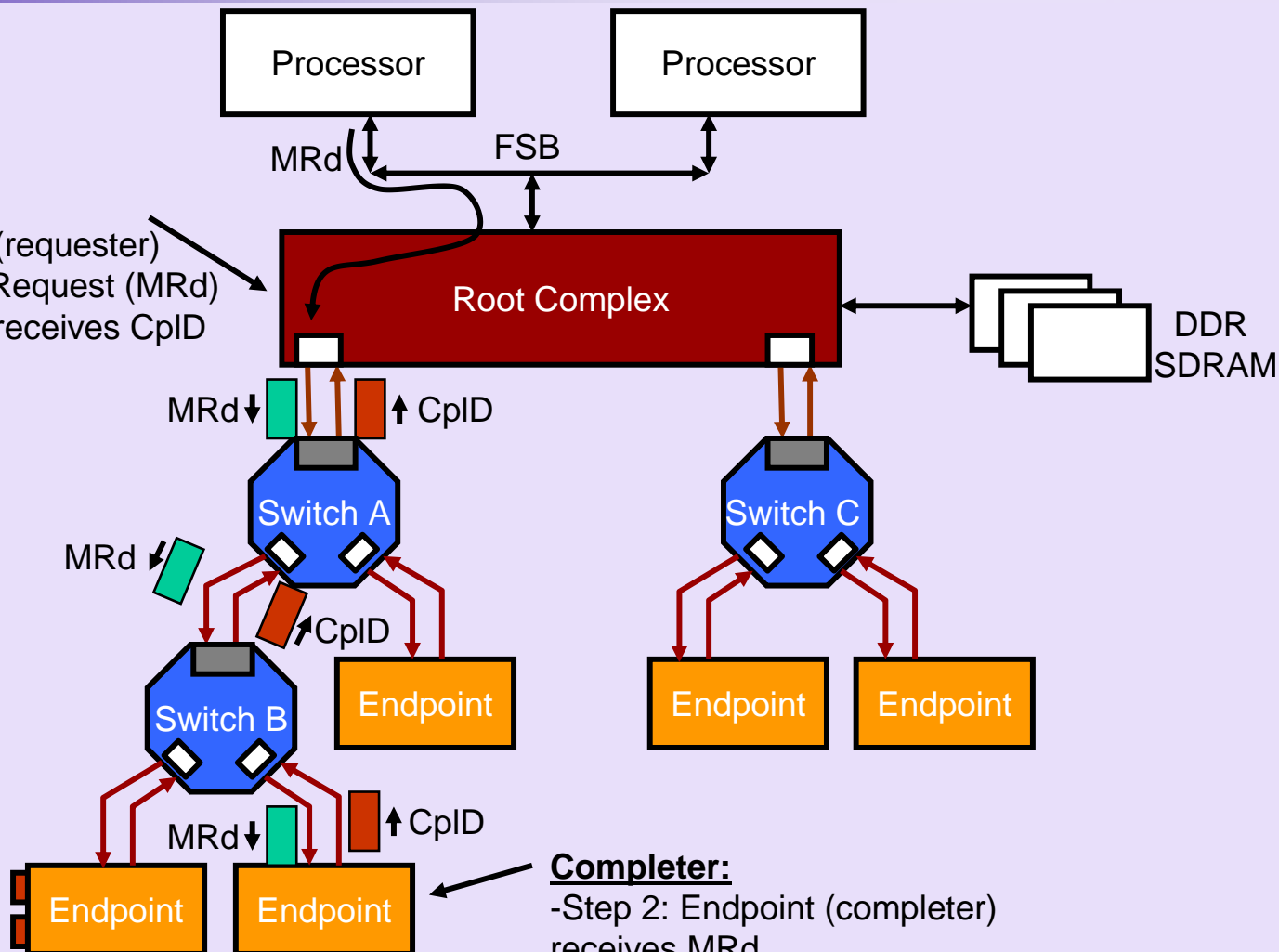
Three Methods For Packet Routing

- Each request or completion header is tagged as to its *type*, and each of the packet types is routed based on one of three schemes:
 - ✓ Address Routing
 - ✓ ID Routing
 - ✓ Implicit Routing
- Memory and IO requests use address routing
- Completions and Configuration cycles use ID routing
- Message requests have selectable routing based on a 3-bit code in the message routing sub-field of the header type field

Programmed I/O Transaction

Requester:

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



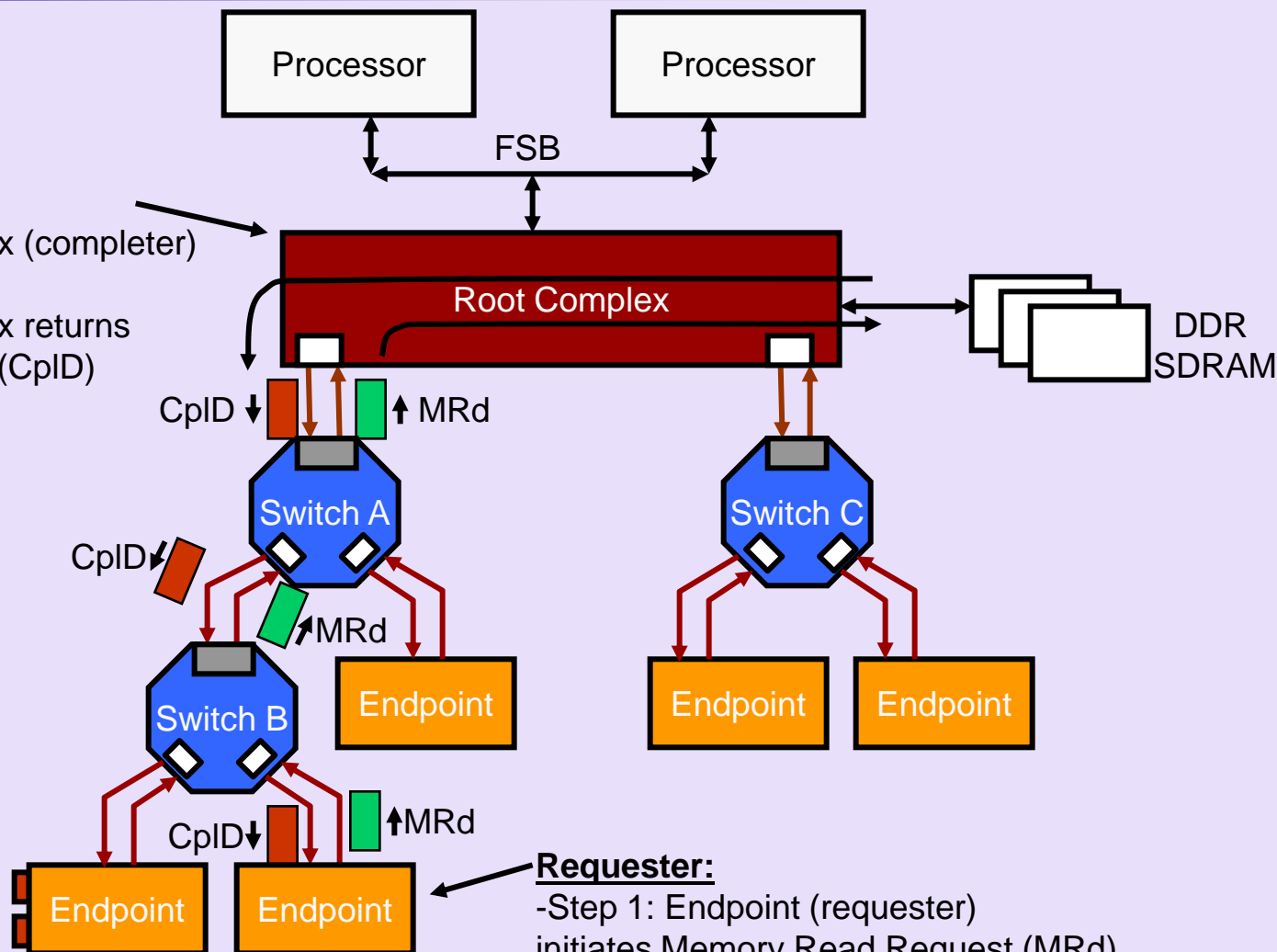
Completer:

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

DMA Transaction

Completer:

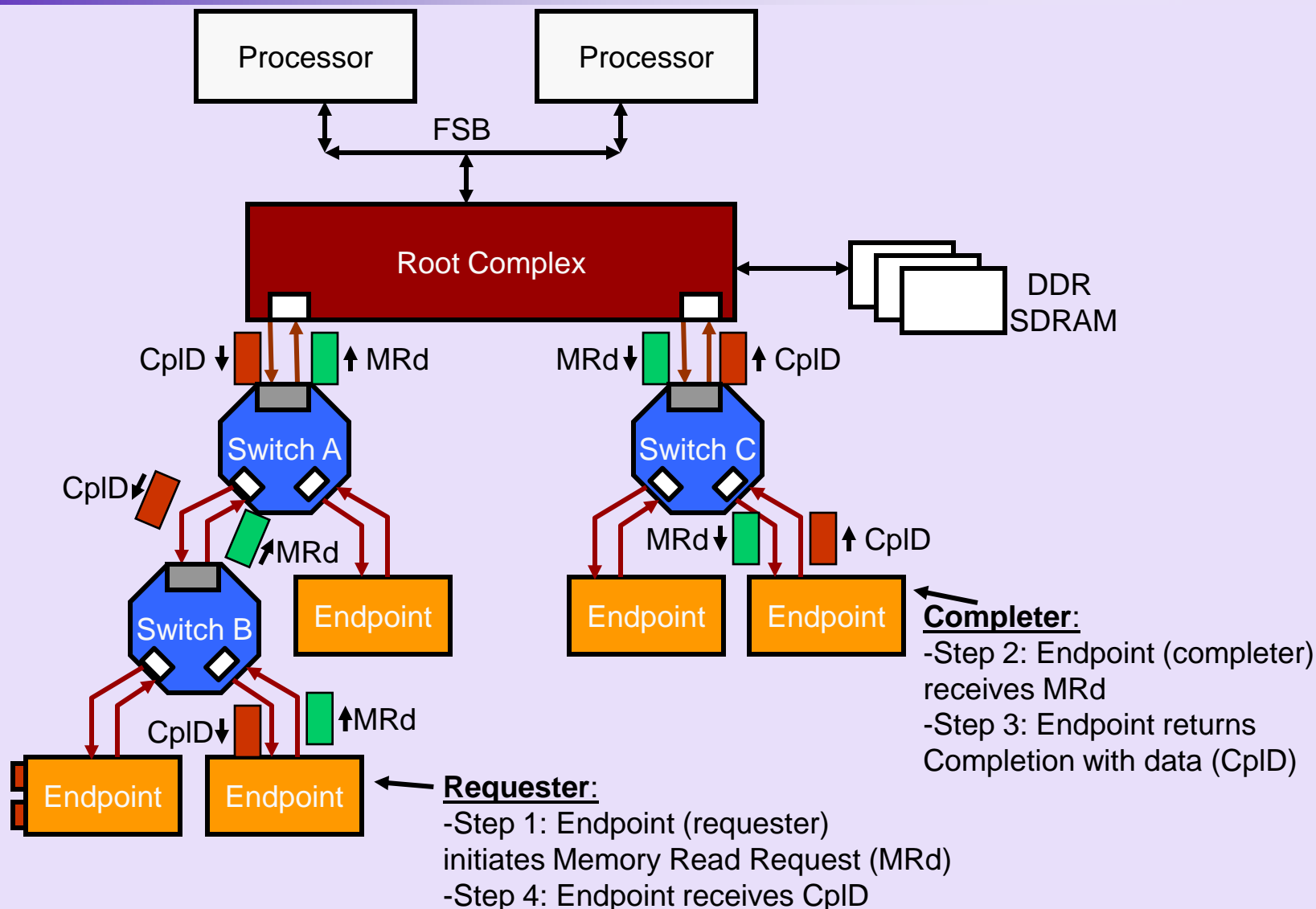
- Step 2: Root Complex (completer) receives MRd
- Step 3: Root Complex returns Completion with data (CpID)



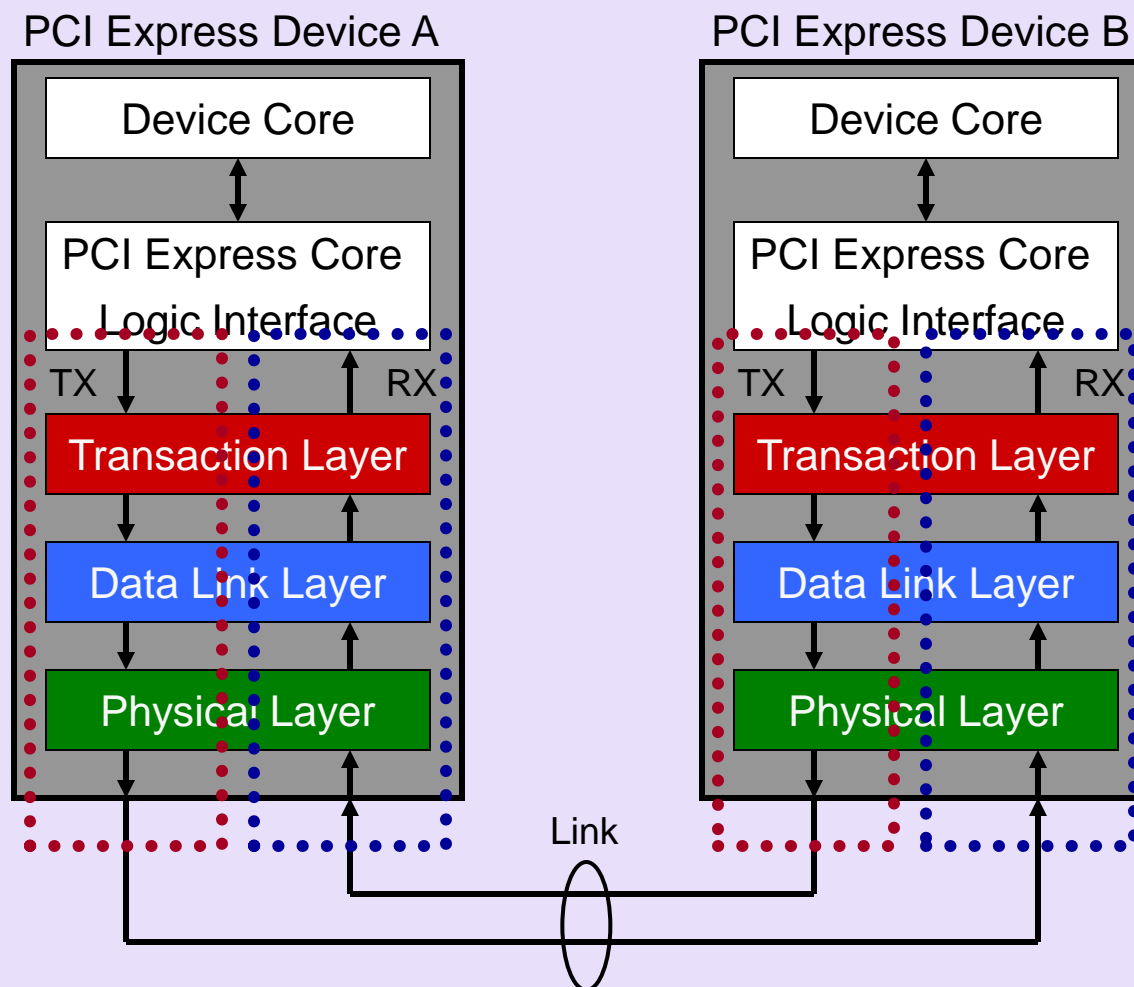
Requester:

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

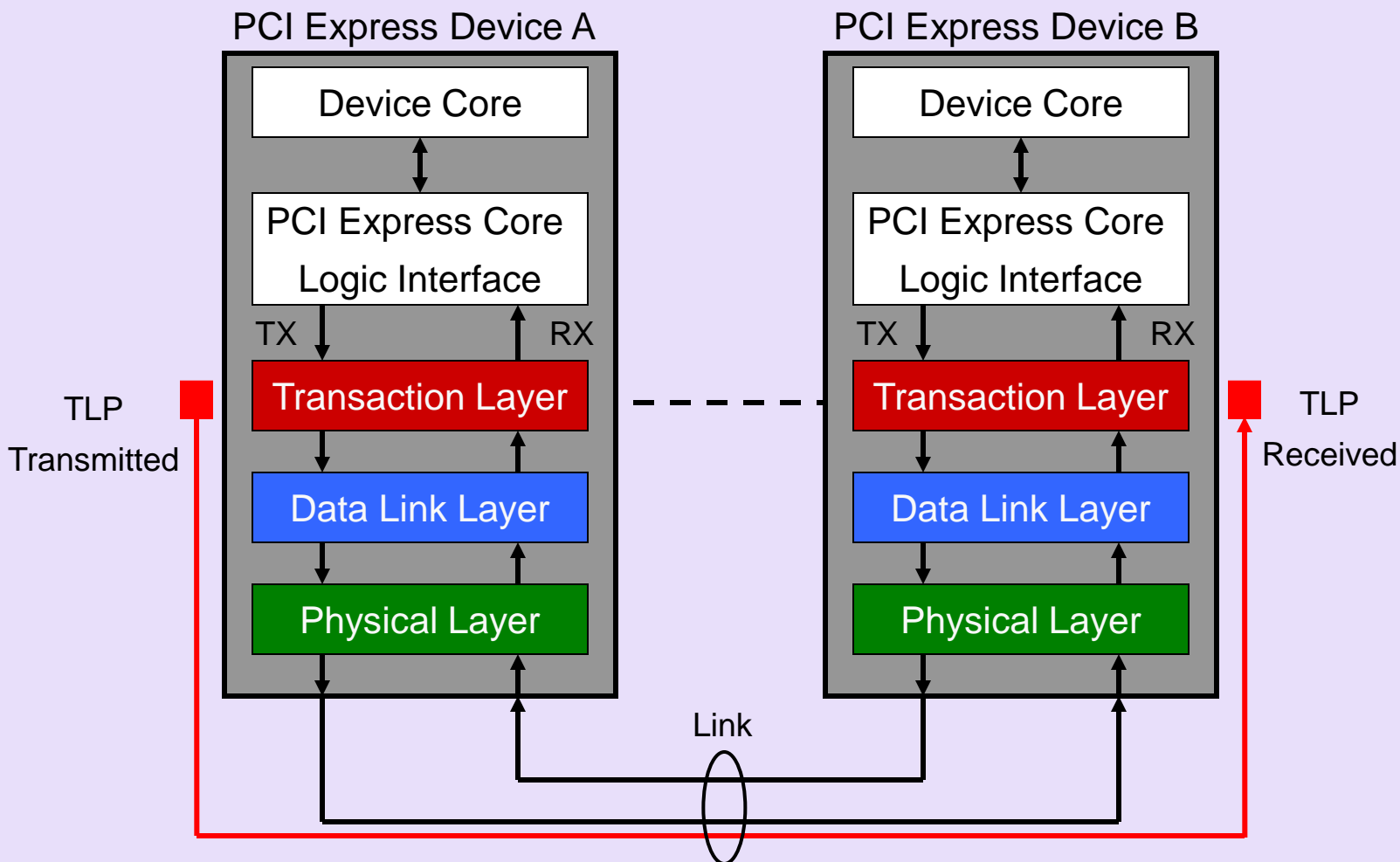
Peer-to-Peer Transaction



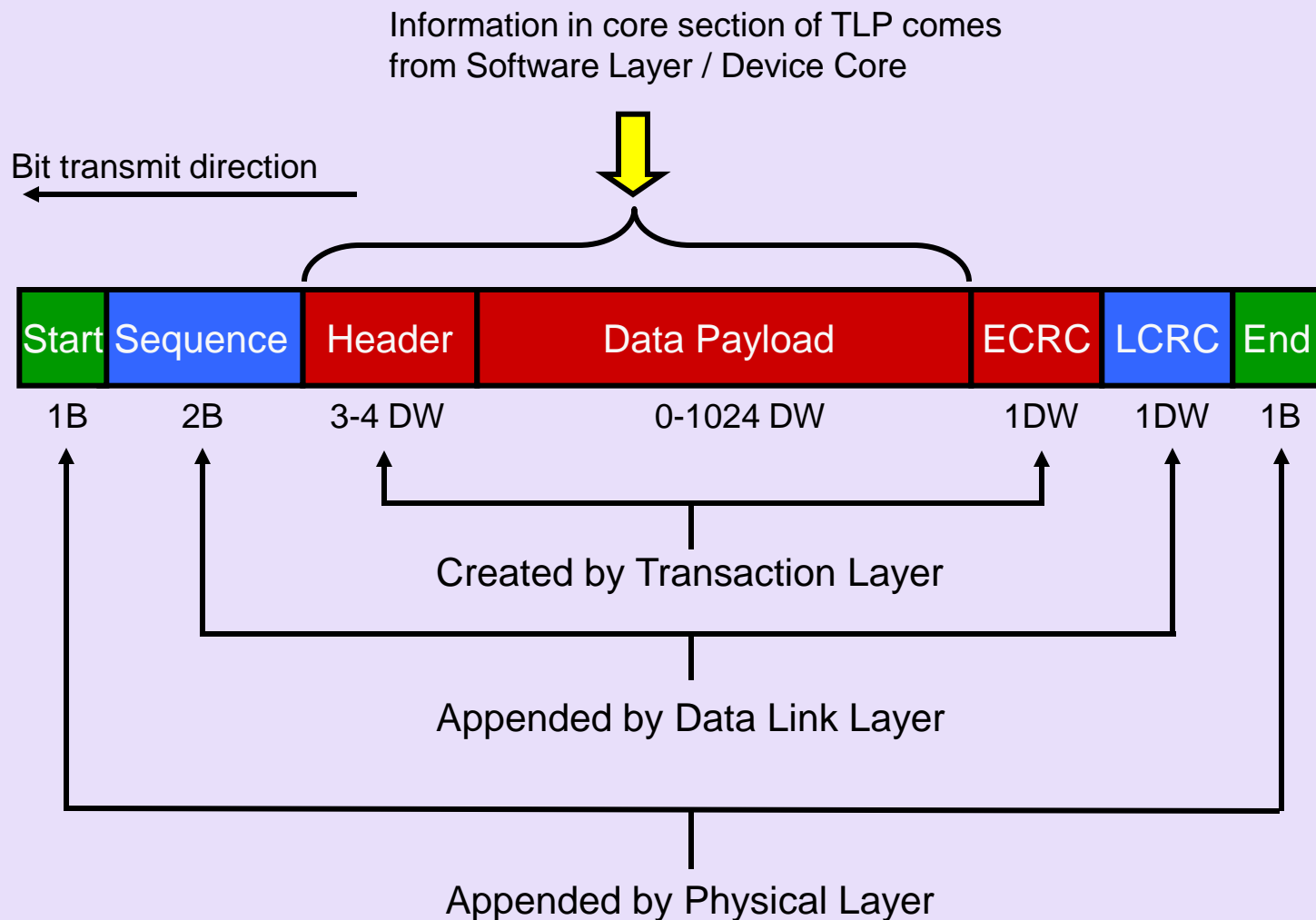
PCI Express Device Layers



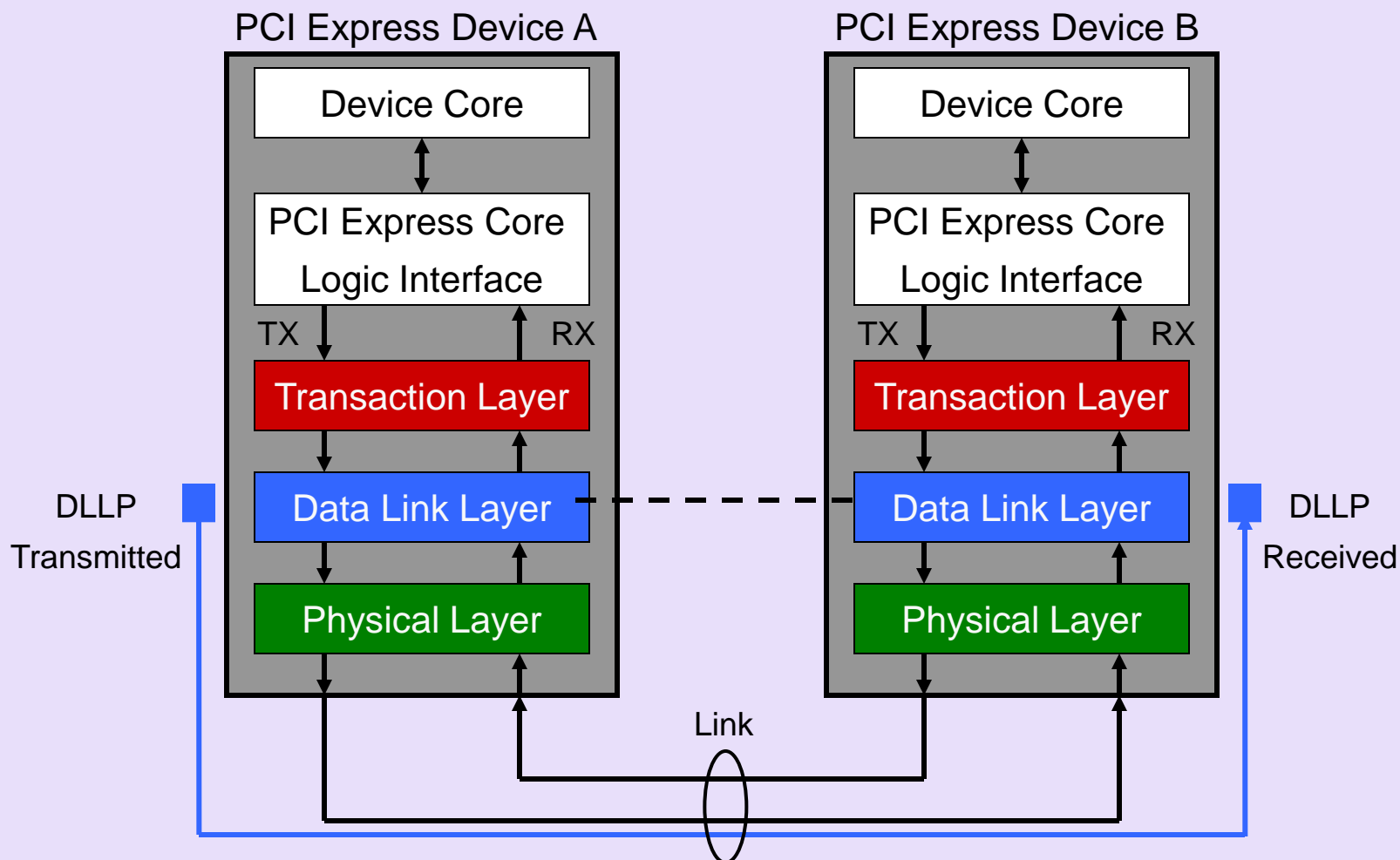
TLP Origin and Destination



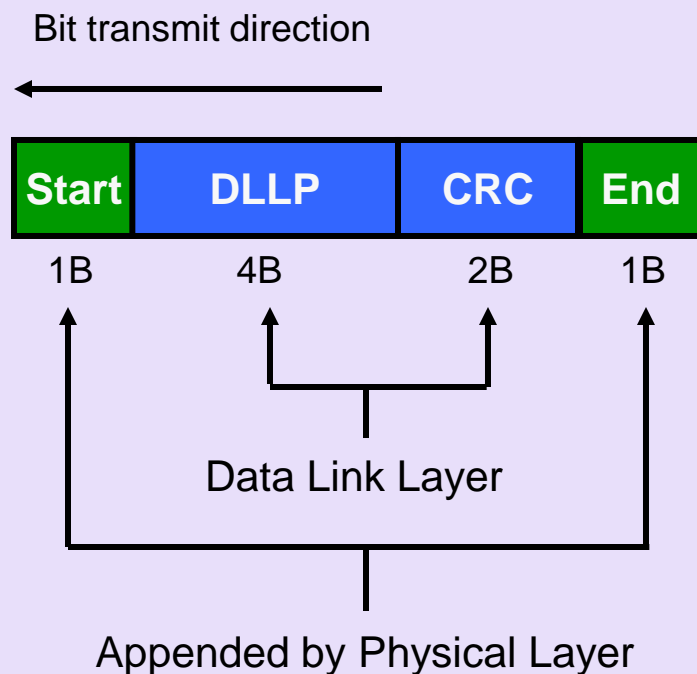
TLP Structure



DLLP Origin and Destination

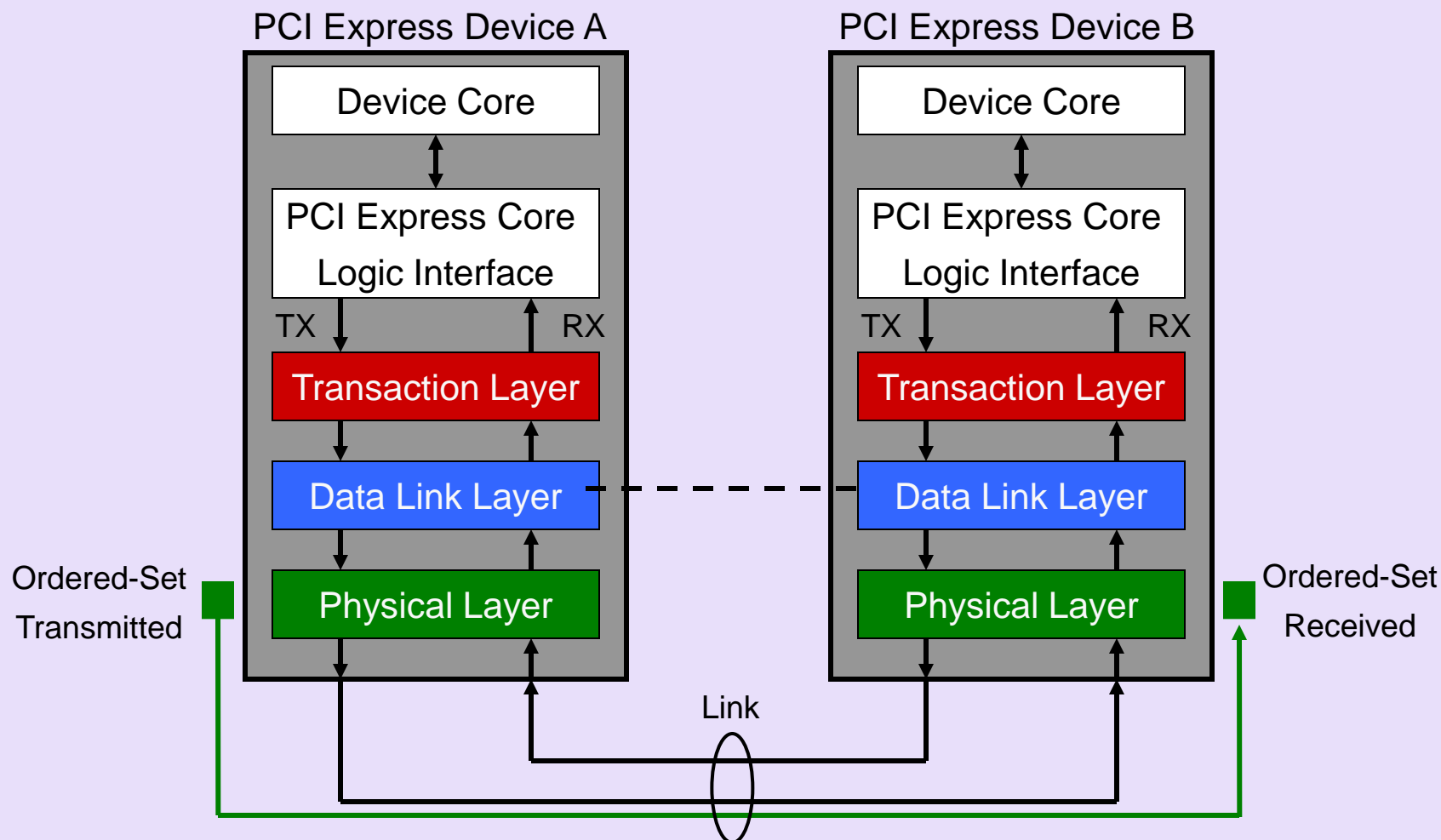


DLLP Structure

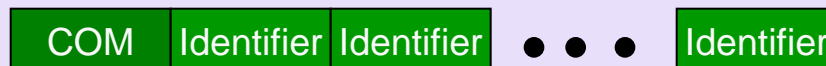


- ACK / NAK Packets
- Flow Control Packets
- Power Management Packets
- Vendor Defined Packets

Ordered-Set Origin and Destination



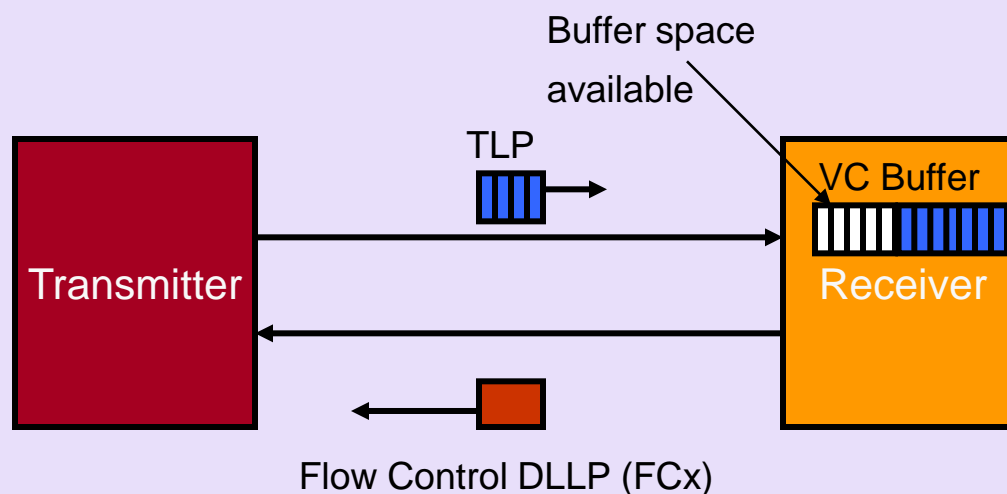
Ordered-Set Structure



- Training Sequence One (TS1)
 - ✓ 16 character set: 1 COM, 15 TS1 data characters
- Training Sequence Two (TS2)
 - ✓ 16 character set: 1 COM, 15 TS2 data characters
- SKIP
 - ✓ 4 character set: 1 COM followed by 3 SKP identifiers
- Fast Training Sequence (FTS)
 - ✓ 4 characters: 1 COM followed by 3 FTS identifiers
- Electrical Idle (IDLE)
 - ✓ 4 characters: 1 COM followed by 3 IDL identifiers
- Electrical Idle Exit (EIEOS) (new to 2.0 spec)
 - ✓ 16 characters

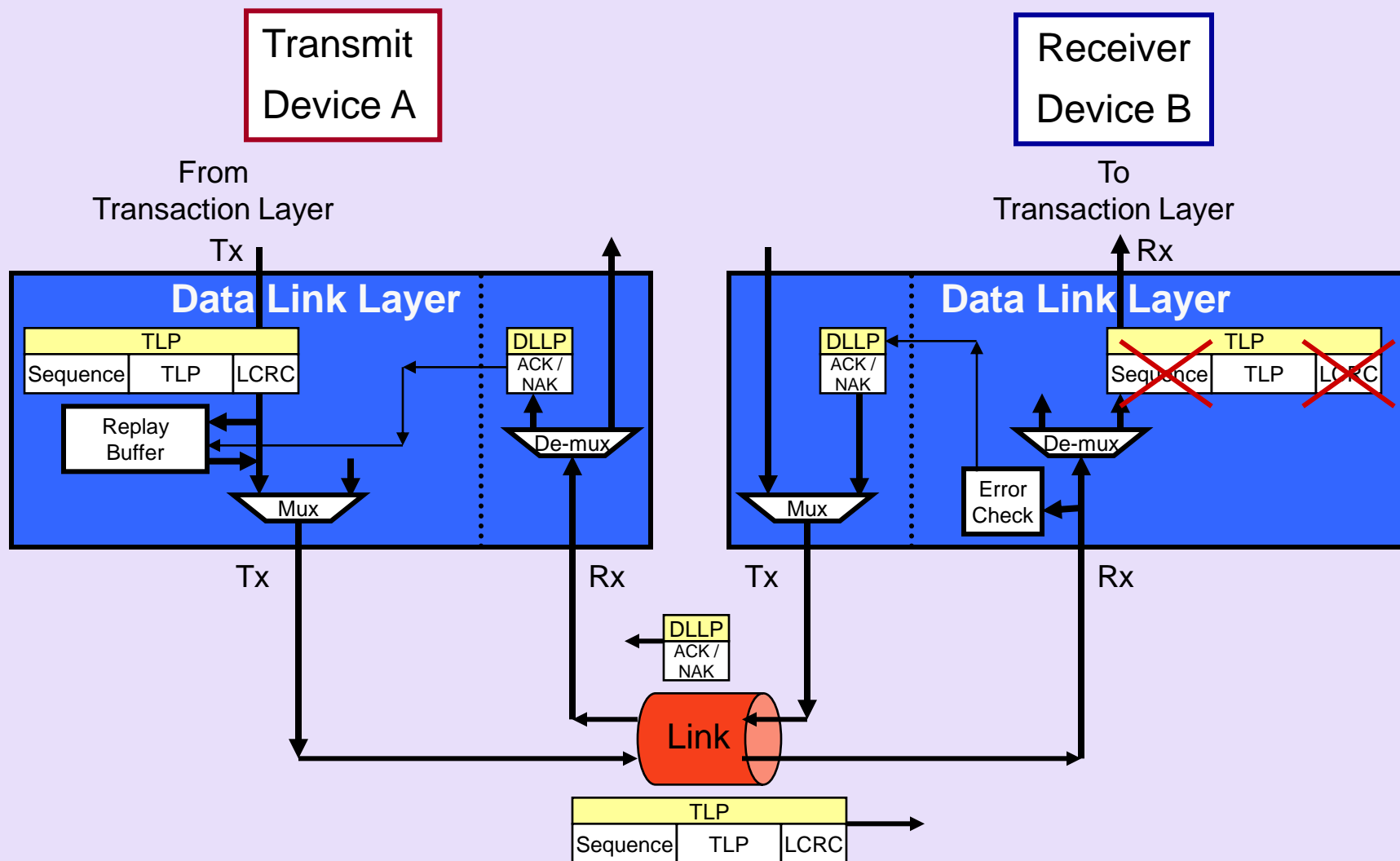
PCI Express Flow Control

- Credit-based *flow control* is point-to-point based, not end-to-end

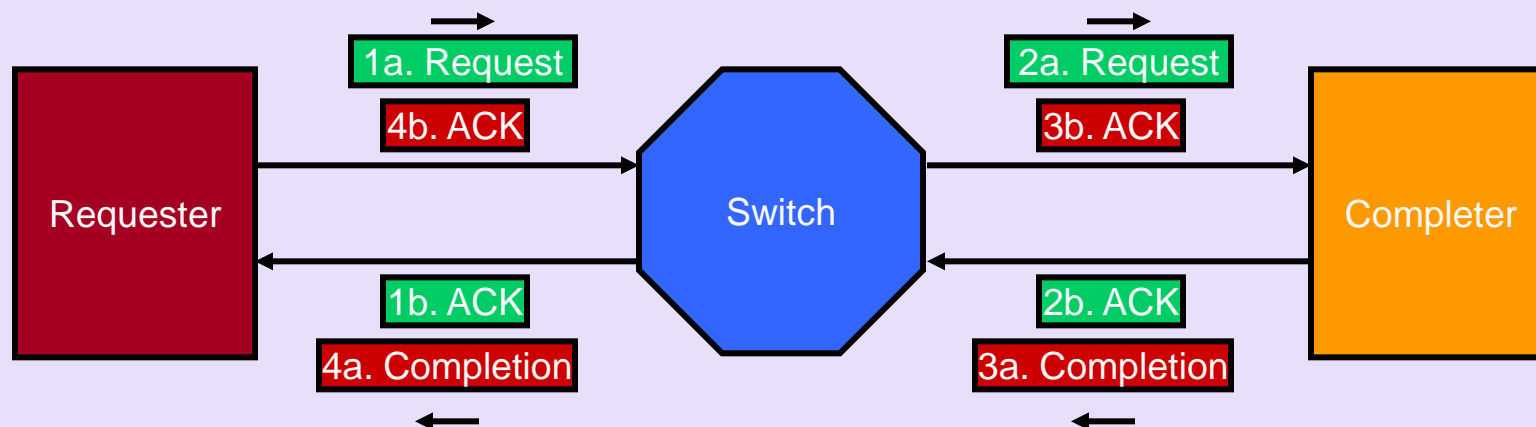


Receiver sends Flow Control Packets (FCP) which are a type of DLLP (Data Link Layer Packet) to provide the transmitter with credits so that it can transmit packets to the receiver

ACK/NAK Protocol Overview



ACK/NAK Protocol: Point-to-Point



ACK returned for good reception of Request or Completion
 NAK returned for error reception of Request or Completion

Thank you for attending the
PCIe Technology Seminar

For more information please go to
www.pcisig.com