

Introduction to PCIe and CXL

Paolo Durante
(CERN EP-LBC)

Where can you find ?

PCI (Peripheral Component Interconnect) Express is a popular standard for high-speed computer expansion
overseen by PCI-SIG
(Special Interest Group)

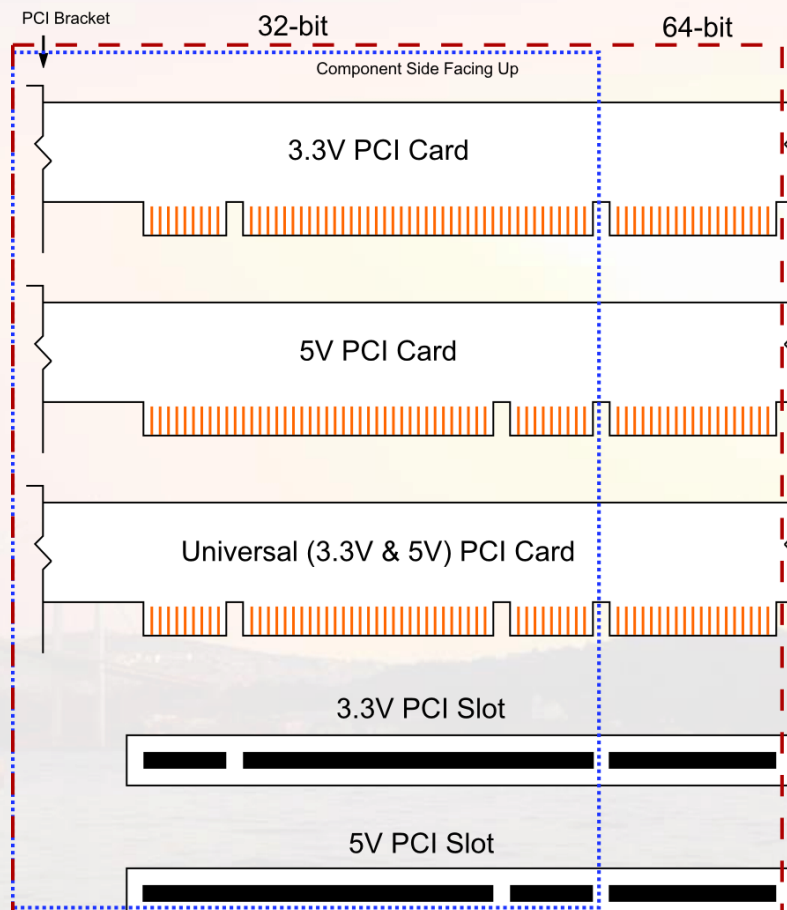
- PCIe interconnects can be present at all levels of your DAQ chain...
 - Readout boards
 - Storage media
 - Network interfaces
 - Compute accelerators (GPUs, FPGAs...)
- ...and may be even more so in the future (with CXL)
 - Memory expanders
- Understanding your data acquisition system requires (some) level of understanding of PCI Express

What is this presentation about?

- PCIe history and evolution
- PCIe concepts
- PCIe layers
- PCIe practical aspects
- PCIe performance
- PCIe future roadmap

PCI (“conventional PCI”)

- 1992
- *Peripheral Component Interconnect*
- Parallel Interface
- Bandwidth
 - 133 MB/s (~1.0 Gb/s) (32-bit@33 MHz)
 - 533 MB/s (~4.2 Gb/s) (64-bit@66 MHz)
- Plug-and-Play configuration (BARs)



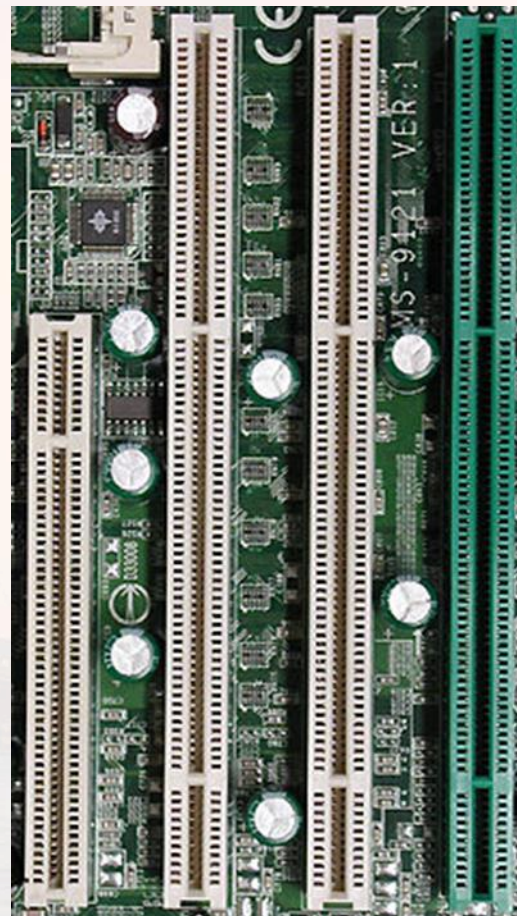
PCI example: ATLAS FILAR

- ~2003
- 4 optical channels
 - 160 MB/s (1.28 Gb/s)
- S-LINK protocol
 - 2 Altera FPGAs
- Burst-DMA over PCI
 - 3rd Altera FPGA
- 64-bit@66MHz PCI



PCI-X (“Extended PCI”)

- 1998
- PCI compatible
 - Hardware and software
 - Half-duplex bidirectional
- Higher bus efficiency
 - Split-responses
 - Message Signaled Interrupts
- Bandwidth
 - $\leq 1066 \text{ MB/s}$ ($\sim 8.5 \text{ Gb/s}$)
(64-bit@133 MHz)
 - 2133 MB/s ($\sim 17 \text{ Gb/s}$)
(*PCI-X 266*)
 - 4266 MB/s ($\sim 34 \text{ Gb/s}$)
(*PCI-X 533*)



PCI-X example: CMS FEROL

- ~2011
- 4 SFP+ cages
 - 1x 10 Gb/s Ethernet
 - 3x SlinkXpress
- PCI-X interface to legacy FE (Slink64)
- Altera FPGA
- Simplex TCP-IP



PCI Express (PCIe)

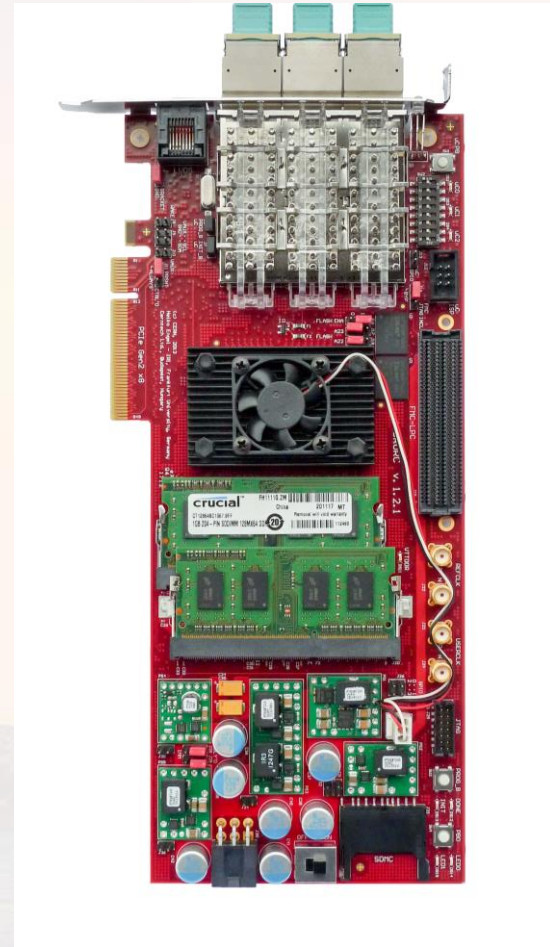
- 2004
- PCI “inspired”
 - software, topology
- Serial interface
- Full-duplex bidirectional
- Bandwidth (Gen4)
 - x1: ≤ 2 GB/s (16 Gb/s) (in each direction)
 - x16: ≤ 32 GB/s (256 Gb/s) (in each direction)
- Still evolving
 - 1.0, 2.0, 3.0, 4.0, 5.0, 6.0...

PCIe x16 | PCI | PCIe x8 | PCI-X



PCIe example: ALICE C-RORC

- ~2014
- 3x QSFP
 - 36 channels
 - up to 6.6Gb/s/channel
- 2x DDR SO-DIMM
- Xilinx Virtex-4 FPGA
- PCIe Gen2 x8
- Also used by ATLAS



PCIe example: LHCb TELL40

- Introduced for LHC Run3
- Currently in production
- ≤ 48 duplex optical links
 - GBT (3.2 Gb/s)
 - WideBus (4.48 Gb/s)
 - GWT (5.12 Gb/s)
- Altera Arria10 FPGA
- 110 Gb/s DMA
- PCIe 3.0 x16
- Also used by ALICE



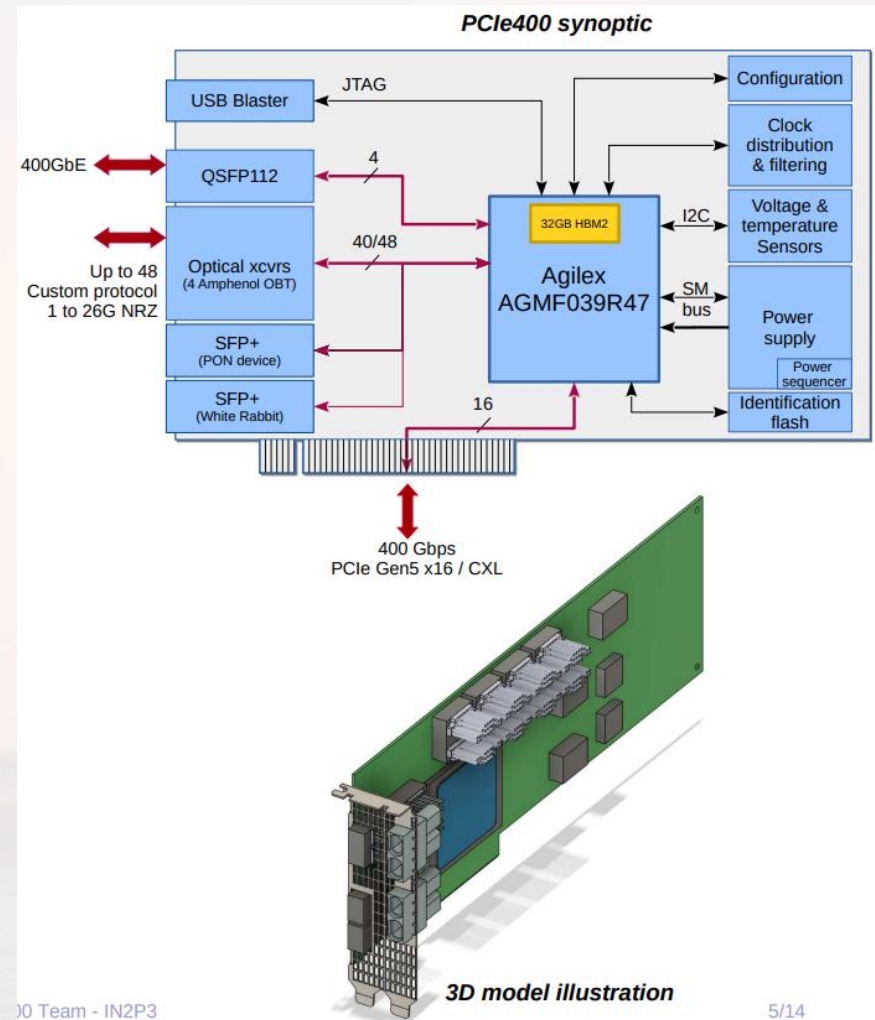
PCIe example: ATLAS FELIX

- Introduced for LHC Run3
- ≤ 48 duplex optical links
- Xilinx Ultrascale FPGA
- 2x DDR4 SO-DIMM
- PCIe 3.0 x16
- Wupper DMA
([Open Source](#))
- Also used by DUNE



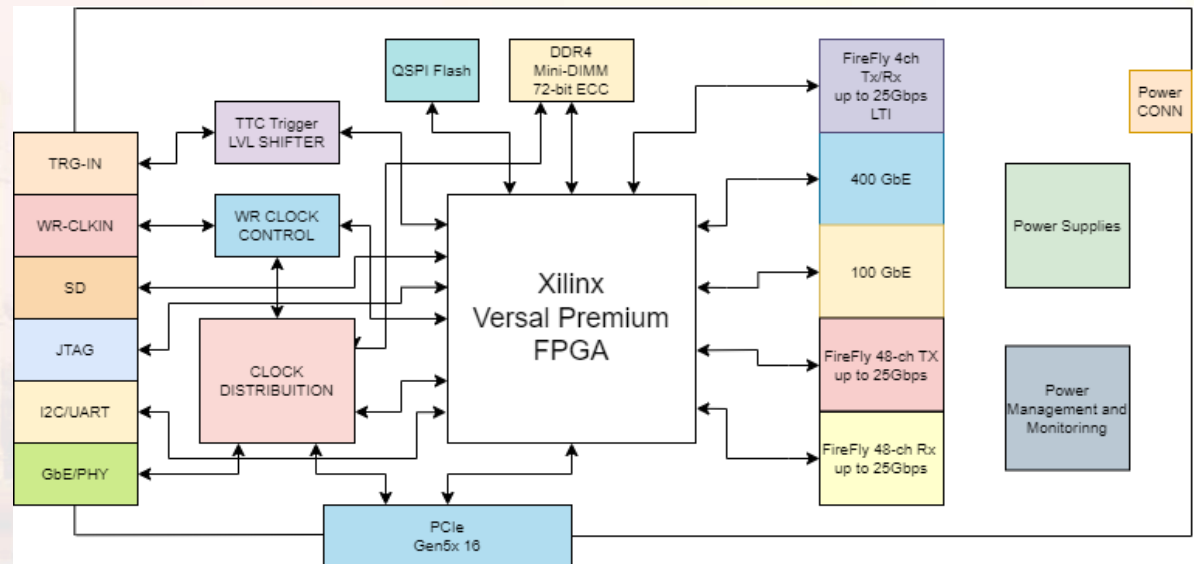
PCIe example: CPPM PCIe400

- PCIe Add in Card 3/4 length
- Agilex 7 M-series AGMF039R47A1E2V
 - Processing capabilities x8 - 12 compared to previous generation FPGA (Arria 10)
- No DDR memory
 - Use of server RAM or HBM2e instead
- Up to 48x26Gbps NRZ for FE
- PCIe Gen 5 / CXL
- QSFP112 for 400GbE (experimental)
- 2 SFP+ for White Rabbit clock distribution or PON fast control
- High precision PLLs jitter <100fs RMS with phase control



PCIe example: BNL FLX-155

- FPGA: Xilinx Versal Premium XCVP1552
- PCIe Gen5 x16, 512 GT/s
- 48 FireFly data links @25 Gb/s
- LTI link
- 100/400 GbE
- DDR4
- GbE
- SD3.0
- White Rabbit
- PetaLinux

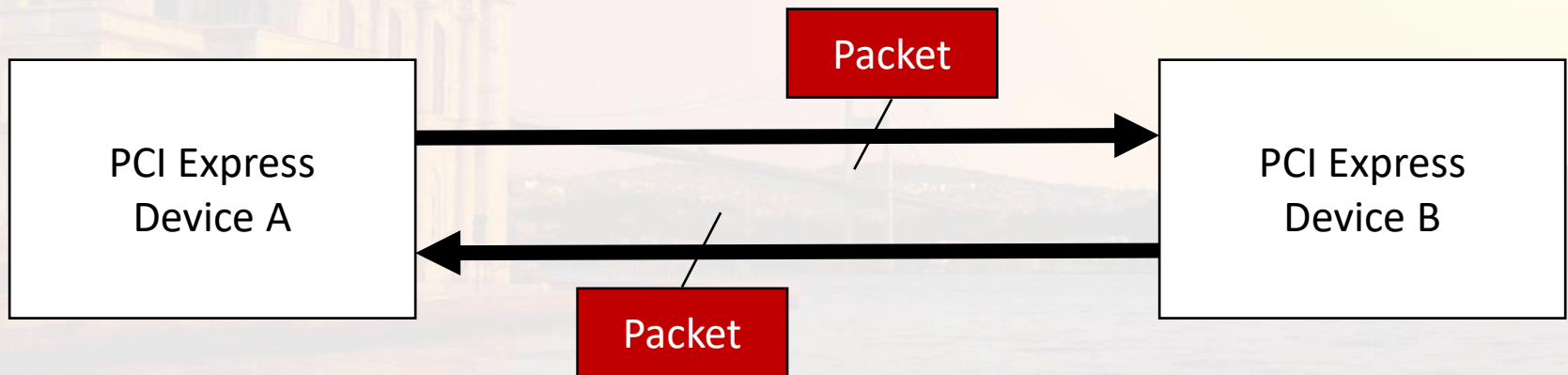


What is this presentation about?

- PCIe history and evolution
- **PCIe concepts**
- PCIe layers
- PCIe practical aspects
- PCIe performance
- PCIe future roadmap

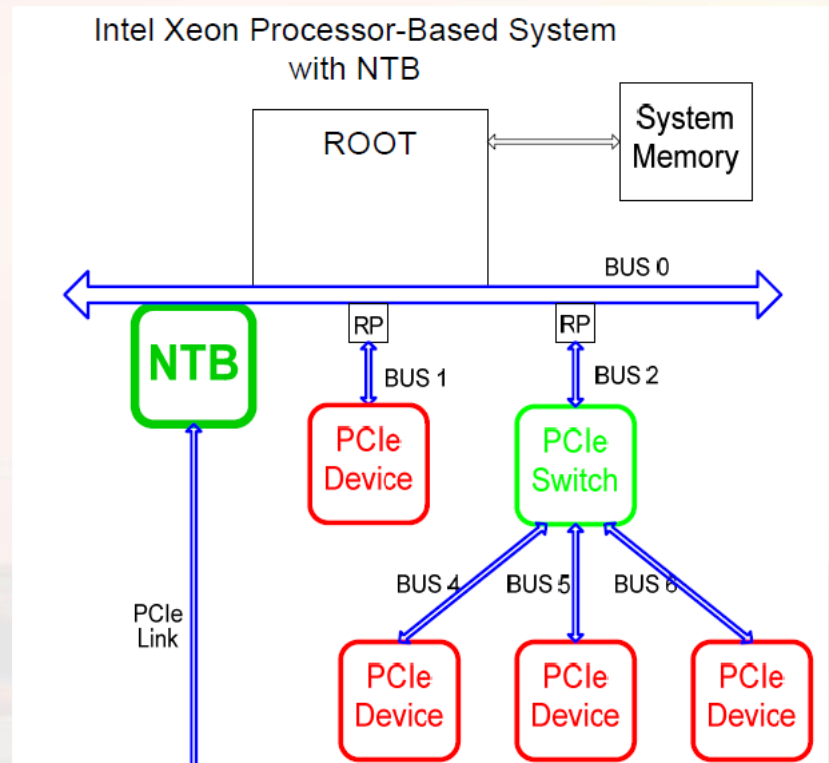
PCIe concepts – Packets

- Point-to-point connection
- “Serial” “bus” (fewer pins)
- Scalable link: **x1**, x2, **x4**, **x8**, x12, **x16**, x32
- Packet encapsulation



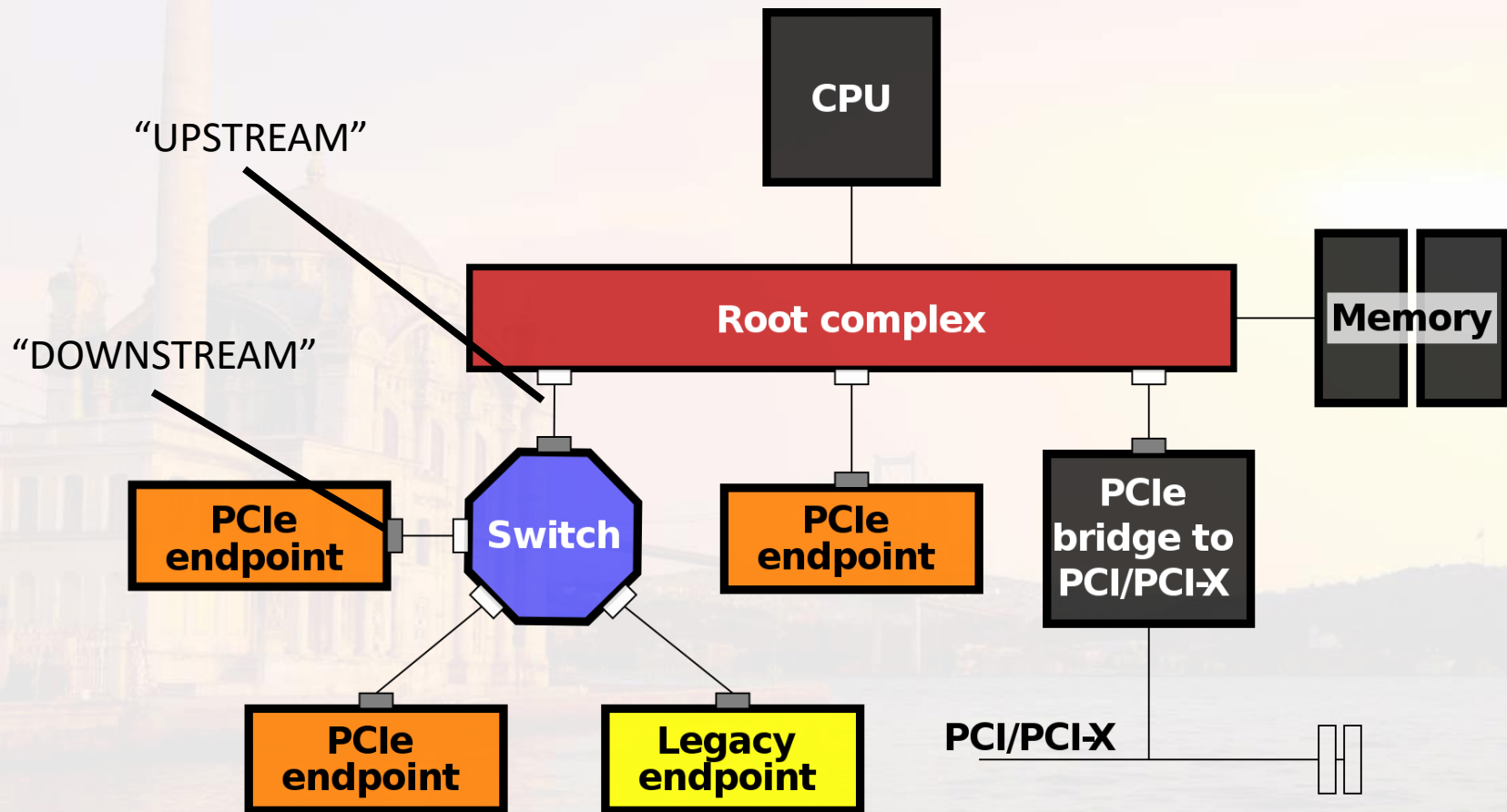
PCIe concepts – Root complex

- Connects the processor and memory subsystems to the PCIe fabric via a Root Port
- Generates and processes transactions with Endpoints on behalf of the processor



PCIe concepts – Topology

Relative to root – up is towards, down is away



PCIe concepts – BDF

“geographical
addressing”

- Bus : Device . Function

- Form a hierarchy-based address
- Multiple logical “Functions” allowed on one physical device
- Bridges (PCI/PCI-X) form hierarchy
- Switches (PCIe) form hierarchy

On linux: \$ man lspci

```
$ lspci -tv
+-[0000:ff]--+-08.0 Intel Corporation Xeon ...
    +-08.3 Intel Corporation Xeon ...
    +-08.4 Intel Corporation Xeon ...
    +-09.0 Intel Corporation Xeon ...
    ...
+-[0000:80]--+-00.0-[81]--
    +-01.0-[82]--
    +-02.0-[83]----00.0 Intel Corporation Xeon Phi coprocessor 31S1
    +-03.0-[84]--
    +-03.2-[85]----00.0 Intel Corporation Xeon Phi coprocessor 31S1
    +-05.0 Intel Corporation Xeon E5/Core i7 Address Map, VTd_Misc, System Management
    +-05.2 Intel Corporation Xeon E5/Core i7 Control Status and Global Errors
    \-05.4 Intel Corporation Xeon E5/Core i7 I/O APIC
+-[0000:7f]--+-08.0 Intel Corporation Xeon E5/Core i7 QPI Link 0
    +-08.3 Intel Corporation Xeon E5/Core i7 QPI Link Reut 0
    ...
\-[0000:00]--+-00.0 Intel Corporation Xeon E5/Core i7 DMI2
    +-01.0-[01]--
    +-01.1-[02]--
    +-02.0-[03]----00.0 Intel Corporation Xeon Phi coprocessor 31S1
    +-03.0-[04]----00.0 Intel Corporation Xeon Phi coprocessor 31S1
    +-05.0 Intel Corporation Xeon E5/Core i7 Address Map, VTd_Misc, System Management
    +-05.2 Intel Corporation Xeon E5/Core i7 Control Status and Global Errors
    +-05.4 Intel Corporation Xeon E5/Core i7 I/O APIC
    +-11.0-[05]--+-00.0 Intel Corporation C602 chipset 4-Port SATA Storage Control Unit
        \-00.3 Intel Corporation C600/X79 series chipset SMBus Controller 0
    +-1c.0-[06]----00.0 Intel Corporation 82574L Gigabit Network Connection
    ...

00:00.0 Host bridge Intel Corporation Xeon E5/Core i7 DMI2 (rev 07)
80:02.0 PCI bridge: Intel Corporation Xeon E5/Core i7 IIO PCI Express Root Port 2a (rev 07)
83:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 31S1 (rev 11)
```

Troubleshooting with lspci

- Device works but is “slow”
 - Link speed
 - Link width
 - MaxPayloadSize
 - Interrupts
 - Error flags
 - Look for bottlenecks upstream
- Device is “there” but driver fails to load
 - Unreadable config space
 - Unallocated BARs

PCIe concepts – Address spaces

- Address spaces
 - Configuration (Bus/Device/Function)
 - Memory (64-bit)
 - I/O (32-bit)
- Configuration space
 - Base Address Registers (BARs) (32/64-bit)
 - Capabilities (linked list)
 - On linux: \$ man setpci

31		16 15		0	
Device ID		Vendor ID		00h	
Status		Command		04h	
Class Code			Revision ID		
BIST	Header Type	Latency Timer	Cache Line Size		
Base Address Registers (BAR) 0					
Base Address Registers (BAR) 1					
Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number		
Secondary Status		I/O Limit	I/O Base		
Memory Limit		Memory Base			
Prefetchable Memory Limit		Prefecthable Memory Base			
Prefetchable Base Upper 32 Bits					
Prefetchable Base Limit 32 Bits					
I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits			
Reserved			Capabilities Pointer		
Expansion ROM Base Address Register (XROMBAR)					
Bridge Control		Interrupt Pin	Interrupt Line		

PCIe concepts – Memory & I/O

- Memory space maps cleanly to CPU semantics
 - 32-bits of address space initially
 - 64-bits introduced via Dual-Address Cycles (DAC)
 - Extra period of address time on PCI/PCI-X
 - 4DWORD header in PCI Express
 - Burstable (= Multiple DWORDs)
- I/O space maps cleanly to CPU semantics
 - 32-bits of address space
 - Non-burstable

PCIe concepts – Bus address

This is actually not specific to PCIe, but a generic reminder:

- Physical address: the address the CPU sends to the memory controller
- Virtual address: an indirect address created by the operating system, translated by the CPU to physical
- Bus address: an address understood by the devices connected to a specific bus
- On Linux, see: *pci_iomap()*, *remap_pfn_range()*, ...

PCIe concepts – Bridges

Transparent

- Single root (or SR-IOV)
- Single address space
- Multiple downstreams (switch)
- Downstreams appear in the same topology
- Addresses are passed through unchanged

Non-Transparent

- Joins two independent topologies
- One root on each side
- Each side has its own address space
- Needs translation table
- Fault tolerance, “networking”, HPC

PCIe concepts – Interrupts

- PCI
 - INTx#
 - $x \in \{A, B, C, D\}$
 - Level sensitive
 - Can be mapped to CPU interrupt number
- PCIe
 - “Virtual Wire” emulation
 - Assert_INTx code
 - Deassert_INTx code

```
pci_read_config_byte(dev,  
    PCI_INTERRUPT_PIN,  
    &(...));
```

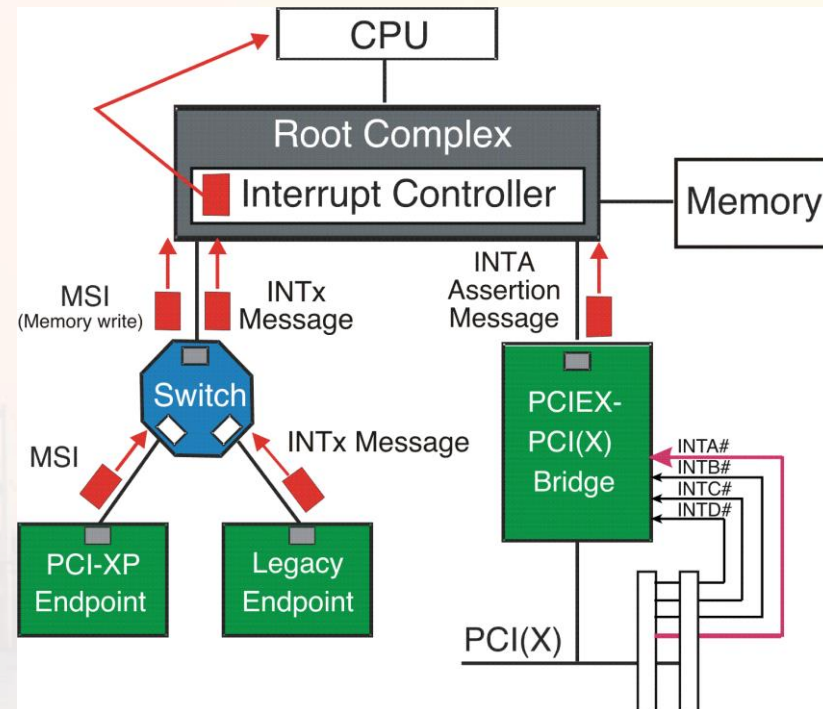
```
pci_read_config_byte(dev,  
    PCI_INTERRUPT_LINE,  
    &(...));
```

```
pci_enable_msi(dev);
```

```
request_irq(dev->irq, my_isr,  
    IRQF_SHARED, devname,  
    cookie);
```


PCIe concepts – MSI & MSI-X


- Based on messages (*MWr*)
- MSI uses one address with a variable data value indicating which “vector” is asserting
 - ≤ 32 per device (in theory)
- MSI-X uses a table of independent address and data pairs for each “vector”
 - ≤ 2048 per device (use affinity!)
- *Vector*: interrupt id




PCIe concepts – GT/s

<https://www.youtube.com/watch?v=ixPqXUEa1Fc&t=09m47s>

OCPSummit19 - EW: Server - PCI Express Enabling Performance in HPC



OCP
GLOBAL
SUMMIT



PCIe 4.0 Specification & Status

Adoption Is Well Under Way

Key Features:


- Delivers 16 GT/s
- Maintains backward compatibility with PCIe 3.x, 2.x, and 1.x
- Implements:
 - Extended tags and credits
 - Reduced system latency
 - Lane margining
- Superior RAS capabilities
- Scalability for added lanes and bandwidth
- Improved I/O virtualization and platform integration
- dB loss is 28dB

Compliance Status:

- PCI-SIG Launched Official FYI Testing for PCIe 4.0 in December 2018
- >50 participants in pre-FYI testing
- Formal Compliance testing targeted for Q3 2019

Adoption:

- Numerous vendors with 16GT/s PHYs and controllers in silicon
- Test equipment from multiple vendors
- Several member companies have publicly announced & exhibited PCIe 4.0 products


 **OCP**
SUMMIT

Open. Together.

San Jose Convention Center | San Jose, California

March 14-15, 2019

9:48 / 22:49 • PCI Express > Scroll for details



PCIe Gen1 (2003)

- Introduced at 2.5 GT/sec (32 Gb/s/d in x16)
- Also called 2.5 GHz, 2.5 Gb/s
- 100 MHz reference clock
 - Eases synchronization between ends
 - Can use Spread Spectrum Clocking to reduce EMI
 - Optional, but nearly universal
- 8b/10b encoding used to provide DC balance and reduce “runs” of 0s or 1s which make clock recovery difficult
- Specification Revisions: 1.0, 1.0a, 1.1

PCIe Gen2 (2006)

- Speed doubled to 5 GT/sec (64 Gb/s/d in x16)
- Reference clock remains at 100 MHz
 - Lower jitter clock sources required vs 2.5 GT/sec
 - Generally higher quality clock generation/distribution required
- 8b/10b encoding continues to be used
- Specification Revisions: 2.0, 2.1
- Devices choosing to implement a maximum rate of 2.5 GT/sec can still be fully 2.x compliant

PCIe Gen3 (2010)

$$2 \times 5 = ?$$

PCIe Gen3 (2010)

$$2 \times 5 = 8$$

- Speed “doubled” from 5 GT/sec (126 Gb/s/d in x16)
- More efficient encoding (20% → ~1%)
 - 8b/10b → 128b/130b
- 8 GT/sec electrical rate
 - 10 GT/sec required significant cost and complexity in channel, receiver design, etc.
- Reference clock remains at 100 MHz
- Backwards-compatible speed negotiation

PCIe Gen4 (2017)

$$2 \times 8 = ?$$

PCIe Gen4 (2017)

$$2 \times 8 = 16$$

- Speed doubled from 8 GT/sec (252 Gb/s/d in x16)
- Same 128b/130b encoding
- 16 GT/sec electrical rate
 - Channel length: $\leq 10''/14''$
 - Retimer mandatory for longer channels
 - More complex pre-amplification, equalization stages
- Reference clock remains at 100 MHz
- Backwards-compatible protocol negotiation and CEM spec

PCIe Gen5 (2019)

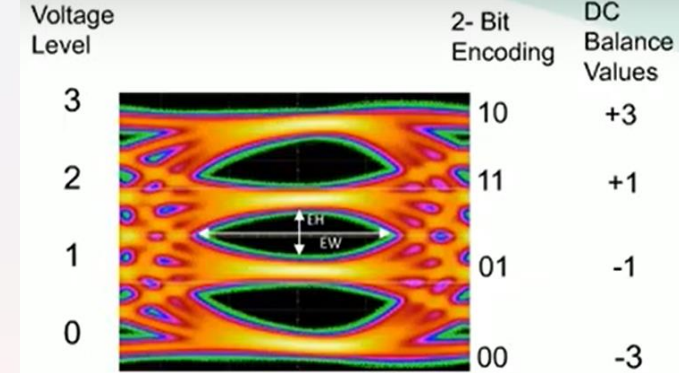
$$2 \times 16 = 32$$

- Speed doubled from 16 GT/sec (504 Gb/s/d in x16)
- Same 128b/130b encoding (with small differences)
- 32 GT/sec electrical rate
 - Channel length: $\leq 10''/14''$
 - Up to 2 retimers for longer channels
 - More complex pre-amplification, equalization stages
- Support for alternate protocols (see CXL)

PCIe Gen6 (2022)


$$2 \times 32 = 64$$

- Speed doubled from 32 GT/sec (1024 Gb/s/d in x16)
- NRZ → PAM4 signaling
 - 2 bits per Unit Interval
 - Lower eye-height and width, much higher First Bit-Error Rate (FBER)
- Forward Error Correction (FEC)
 - Light-weight and low-latency (2ns) FEC for initial correction
 - CRC and link-level retry for larger errors
- Flow Control Unit (FLIT) encoding
 - Fixed-size and fixed(lower)-latency, compared to TLPs



PCIe Gen7 (2023)

PCI Express 7.0: Coming in 2027/2028 External Inbox x Print Share

 **Dr. Ian Cutress from More Than Moore** <morethanmoore@substack.co...> 13 Jun 2023, 20:31 (5 days ago) Star Reply More


to xxxxxxxxxxxx+morethanmoore ▼

[Open in app or online](#)

PCI Express 7.0: Coming in 2027/2028

Initial Specification Now Live

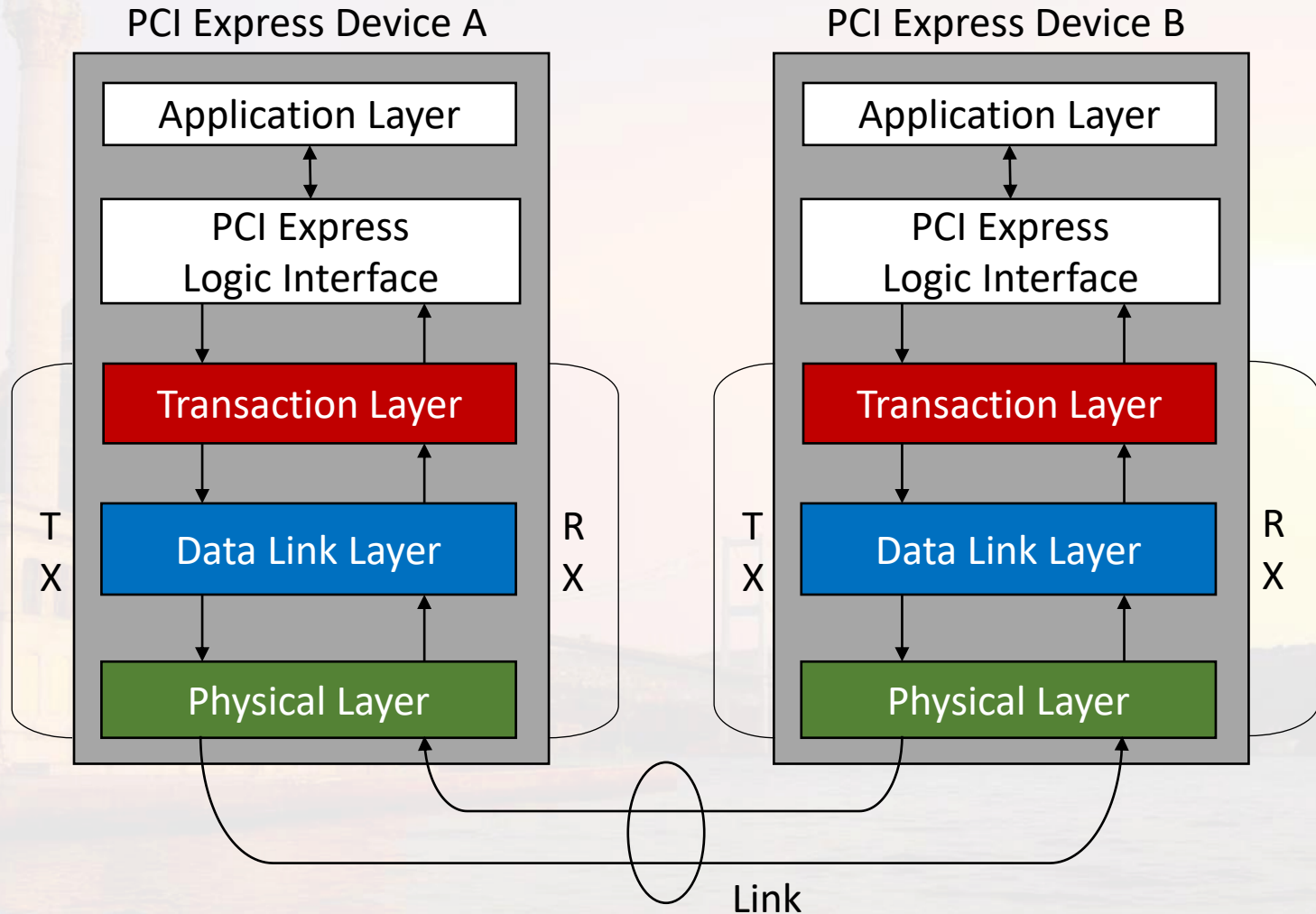
DR. IAN CUTRESS
JUN 13



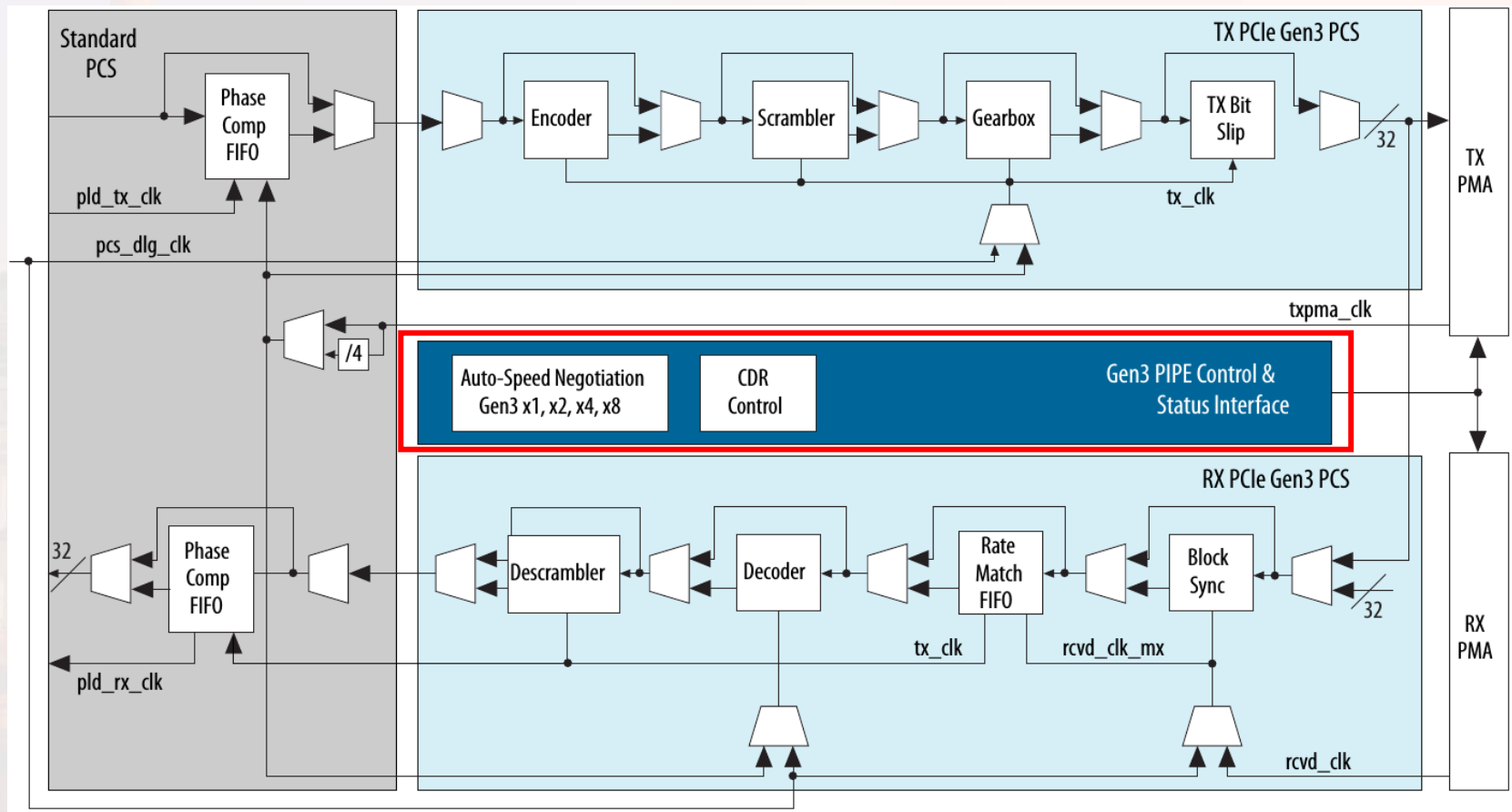
What is this presentation about?

- PCIe history and evolution
- PCIe concepts
- **PCIe layers**
- PCIe practical aspects
- PCIe performance
- PCIe future roadmap

PCIe – Protocol stack



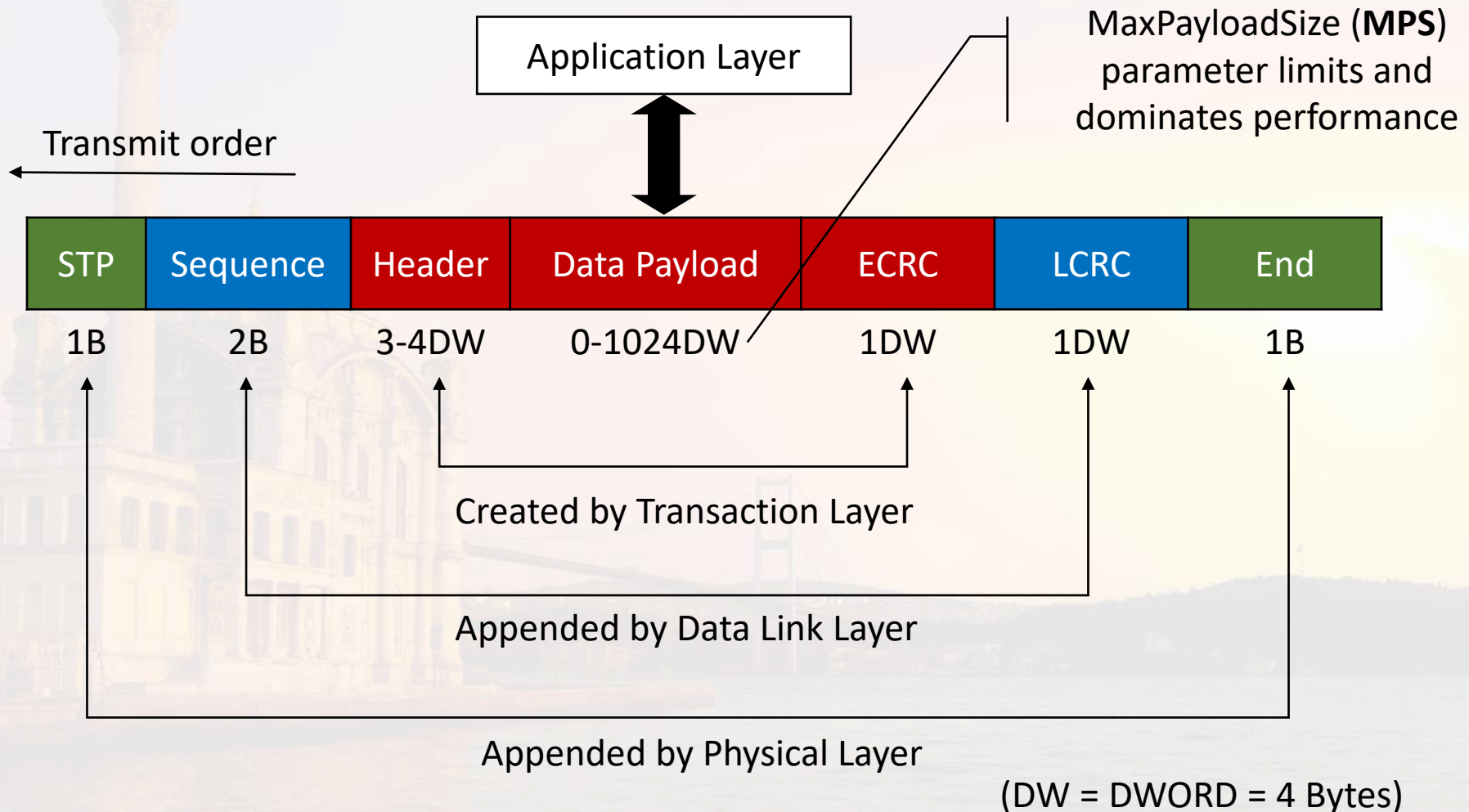
FPGA Hardened PCIe IP



PCIe – Transaction layer

- Four possible transaction types
 - **Memory Read | Memory Write**
 - Transfer data from or to a memory mapped location
 - Address routing
 - **IO Read | IO Write**
 - Transfer data from or to an IO location (on a legacy endpoint)
 - Address routing
 - **Config Read | Config Write**
 - Discover device capabilities, status, parameters
 - ID routing (BDF)
 - **Messages**
 - Event signaling

PCIe – TLP structure



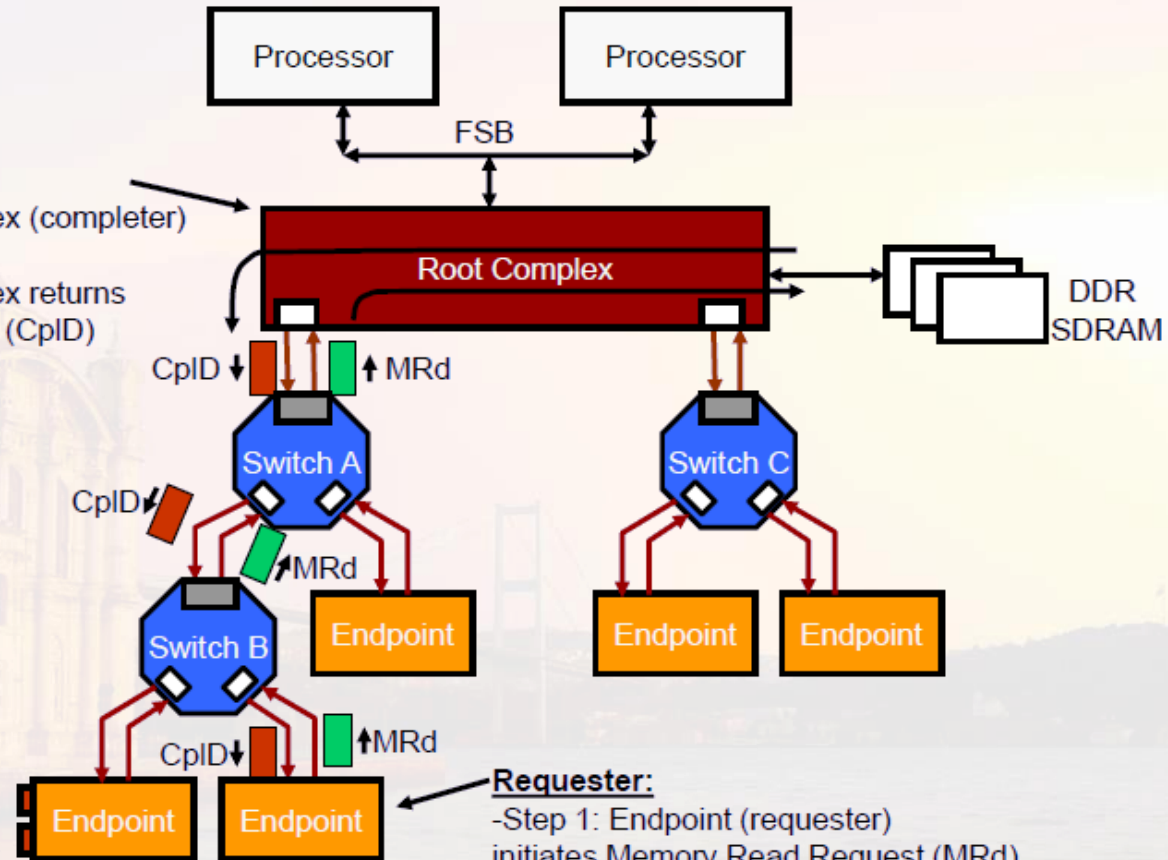
PCIe – Split transaction model

- Posted transaction
 - Single TLP, no completion
- Non-posted transaction
 - Split transaction model
 - Requester initiates transaction (Requester ID + Tag)
 - Requester and Completer IDs encode the sender BDF
 - Completer executes transaction internally
 - Completer creates completion transaction (Cpl/CpID)
- Bus efficiency of Read is different (lower) wrt Write
 - Writes are posted while Reads are not

PCIe – DMA transaction

Completer:

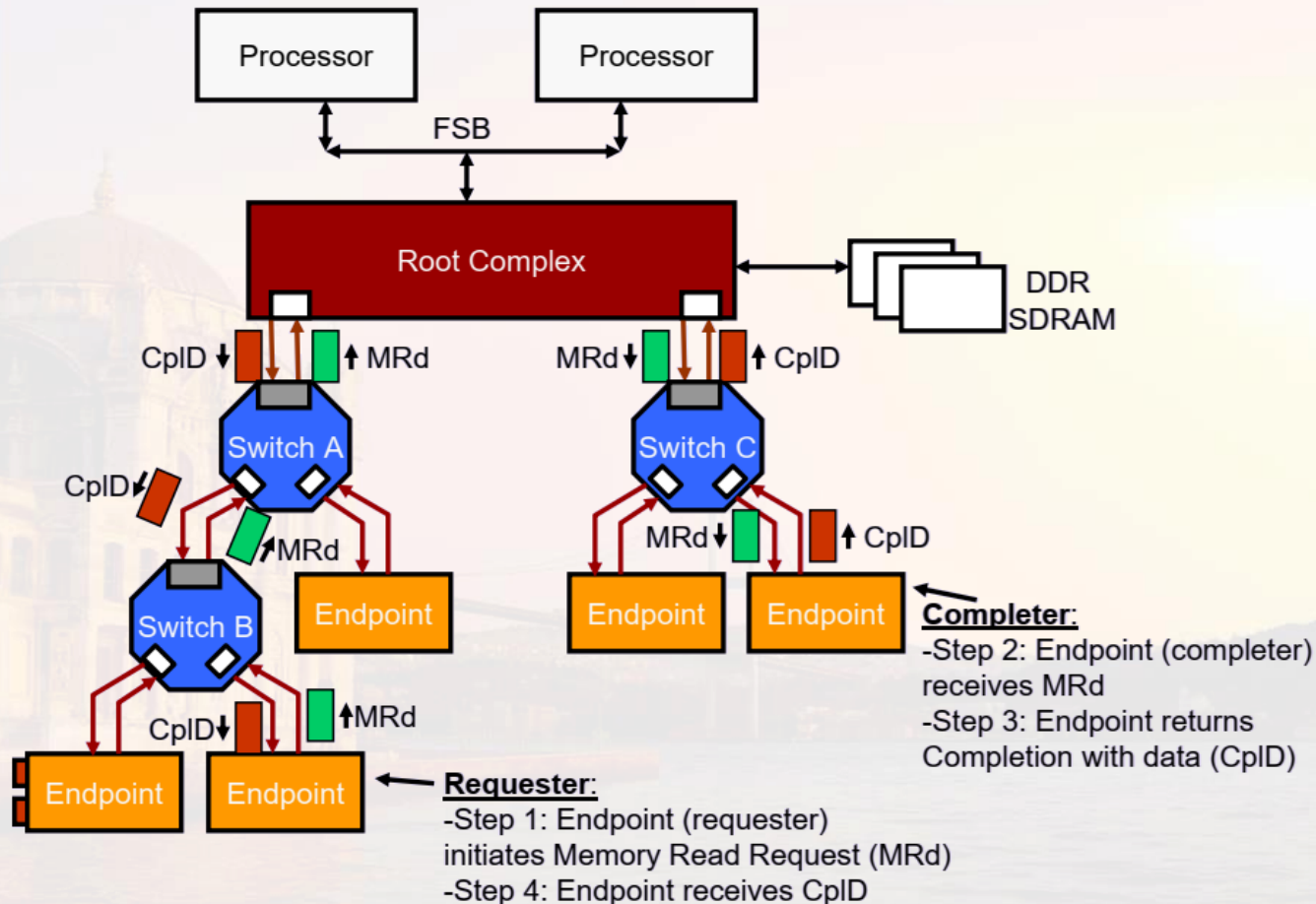
- Step 2: Root Complex (completer) receives MRd
- Step 3: Root Complex returns Completion with data (CpID)



Requester:

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

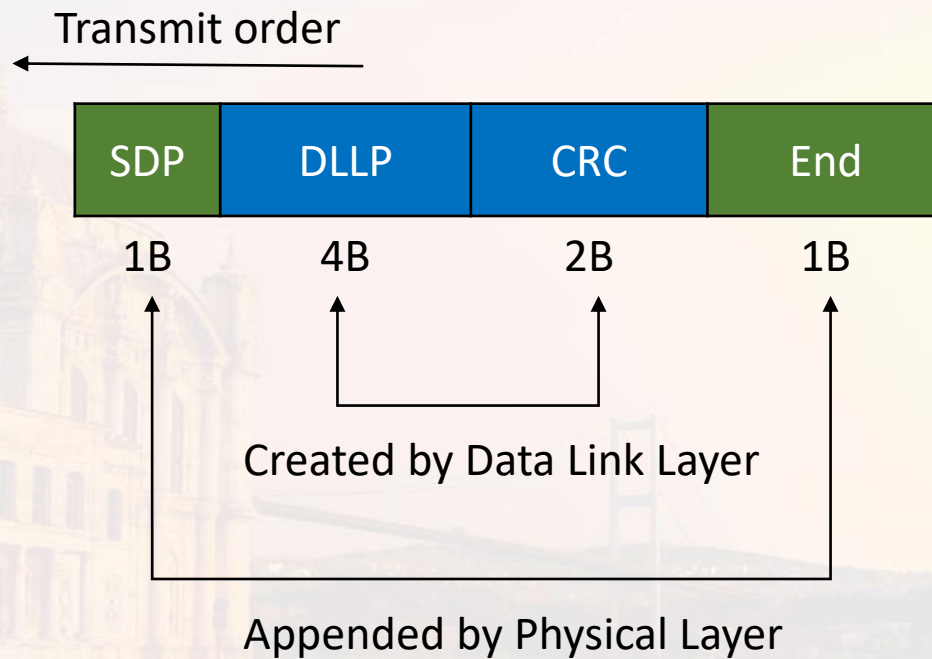
PCIe – Peer-to-Peer transaction



PCIe – Data Link Layer

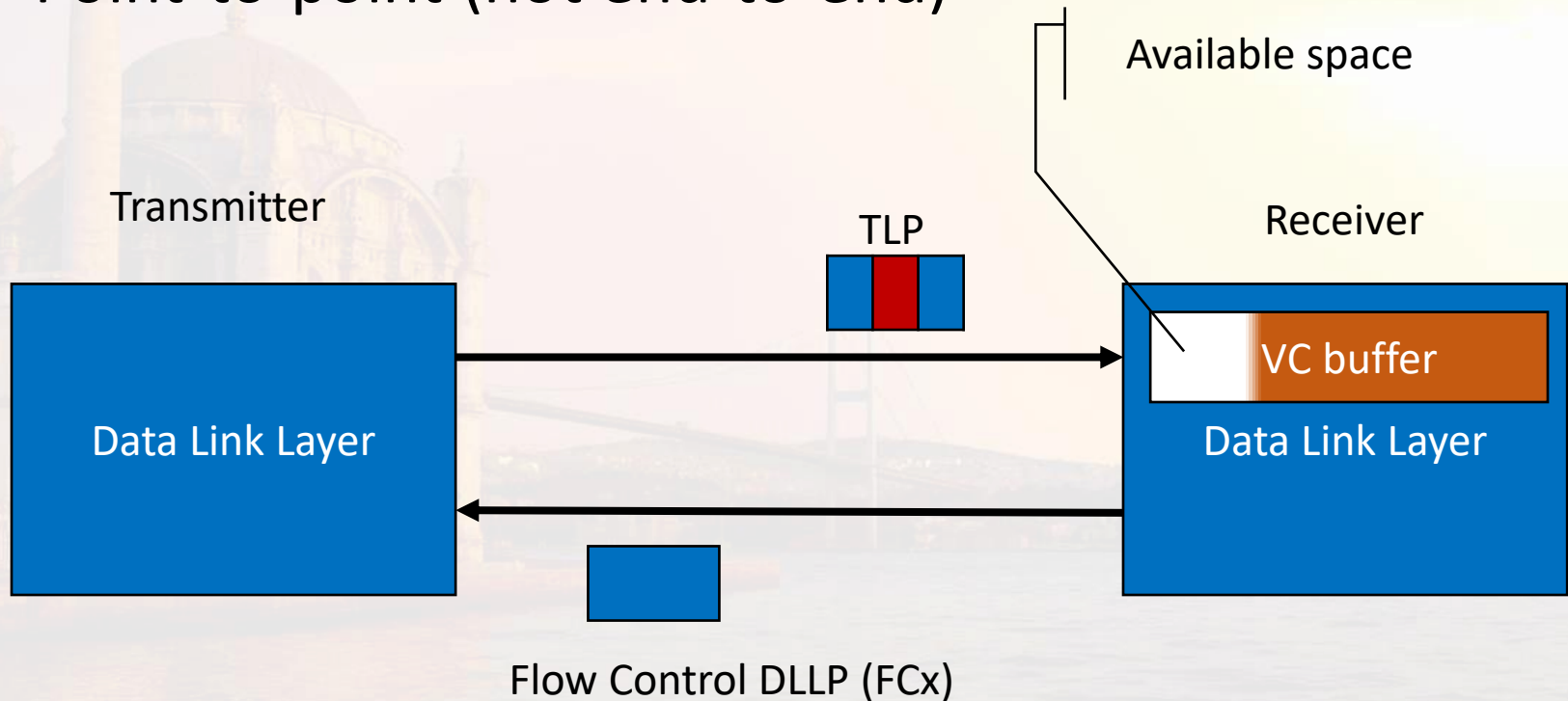
- ACK / NAK Packets
 - Error handling mechanism
- Flow Control Packets (FCPs)
 - Receiver sends FCPs (which are a type of DLLP) to provide the transmitter with **credits** so that it can transmit packets to the receiver
- Power Management Packets
- Vendor extensions
 - E.g.: CAPI, CCIX (memory coherency)

PCIe – DLLP structure



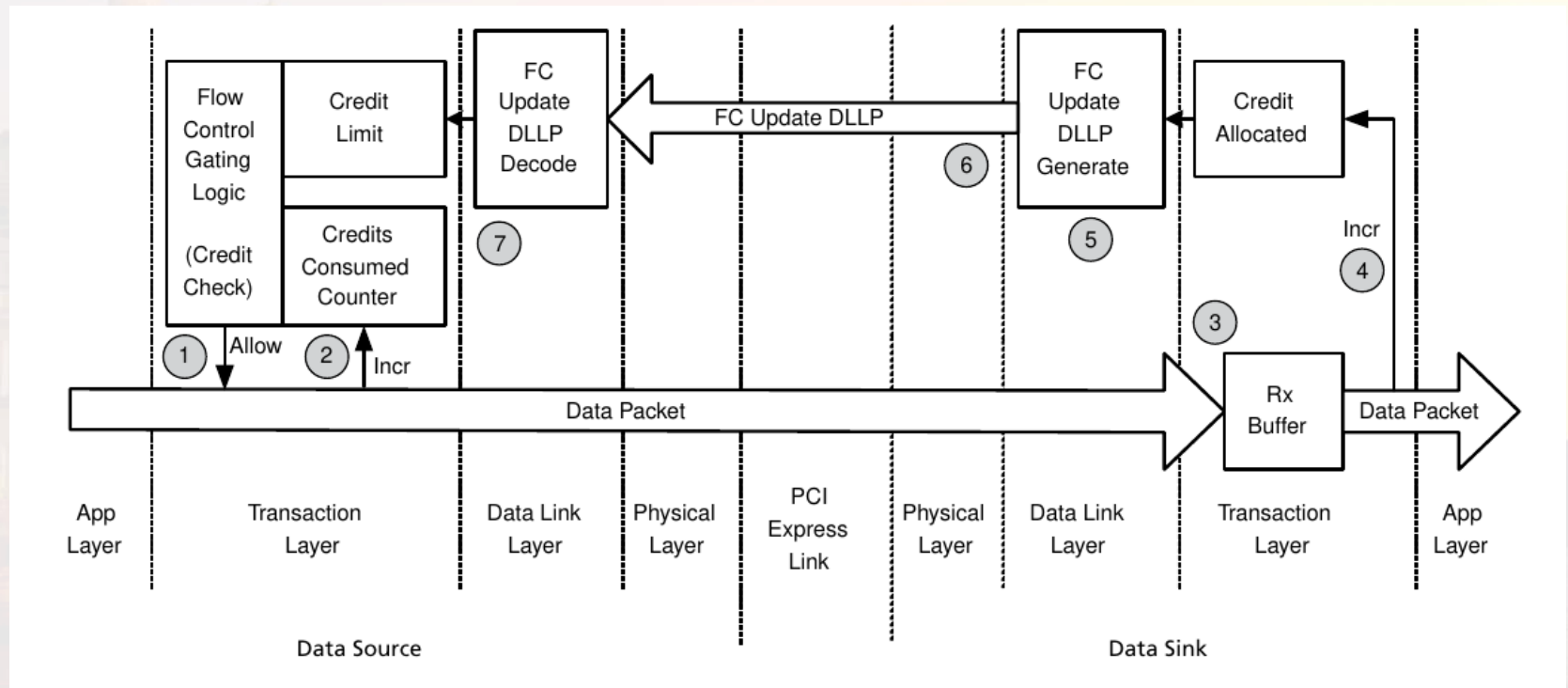
PCIe – Flow control

- Credit-based
- Point-to-point (not end-to-end)



PCIe – Flow Control Update Loop

“If the write requester sources the data as quickly as possible, and the completer consumes the data as quickly as possible, then the Flow Control Update loop may be the biggest determining factor in write throughput, after the actual bandwidth of the link.” (Intel)



PCIe – RAS/QoS features

- Data Integrity and Error Handling
 - PCIe is RAS (Reliable, Available, Serviceable)
 - Data integrity at
 - link level (LCRC)
 - end-to-end (ECRC, optional)
- Virtual channels (VCs) and traffic classes (TCs) to support differentiated traffic or Quality of Service (QoS)
- In theory
 - Ability to define levels of service for packets of different TCs
 - 8 TCs and 8 VCs available
- In practice
 - Rarely more than 1 VC and 1 TC are implemented

PCIe – Error handling

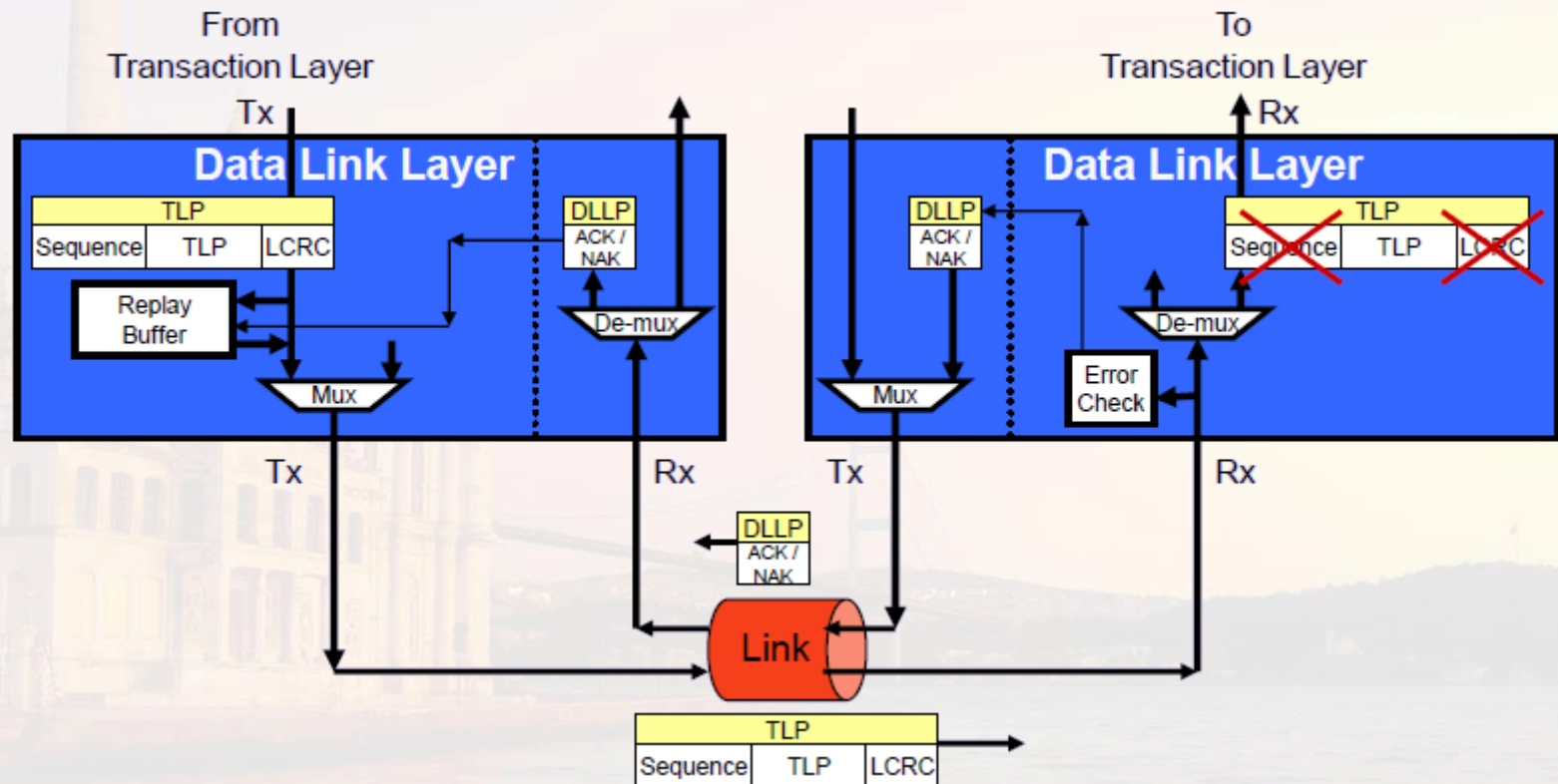
Correctable

- Recovery happens automatically in DLL
- Performance is degraded
- Example: LCRC error
→ automatic DLL retry
(there is no forward error correction until PCIe Gen 6.0)

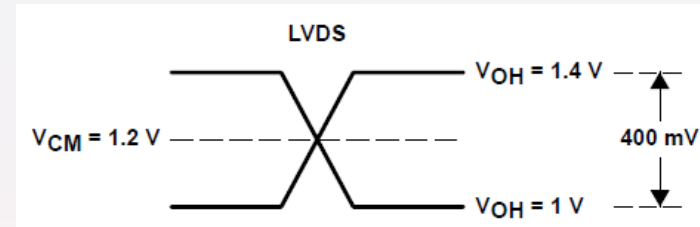
Uncorrectable

- **Fatal**
 - Platform-specific handling
- **Non-fatal**
 - Can be exposed to application layer and handled explicitly
- Can and do cause system deadlock / reset
- Recovery mechanisms are outside the spec
 - Example: failover for HA

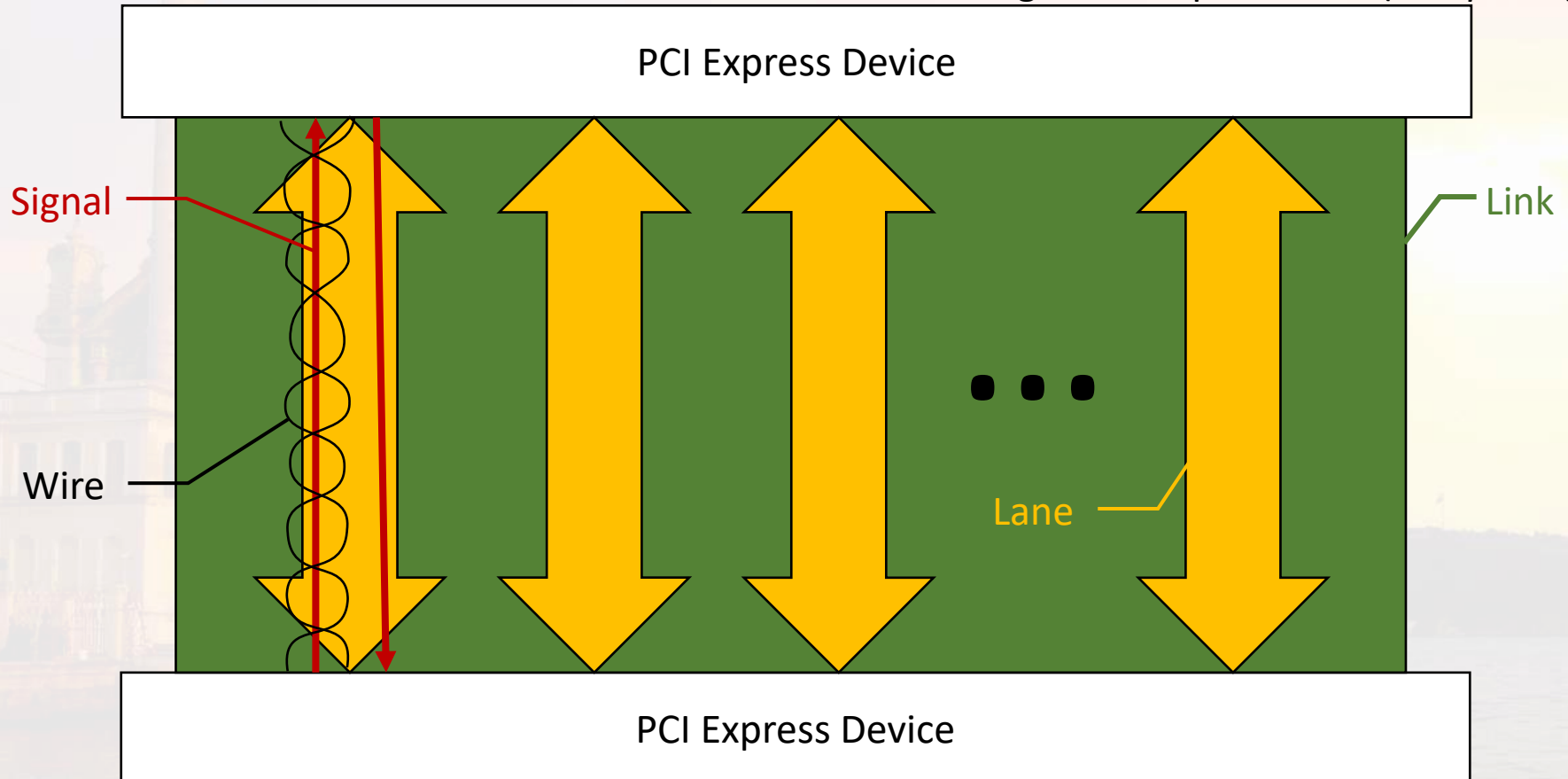
PCIe – ACK/NAK



PCIe – Physical layer



“While the lanes are not tightly synchronized, there is a limit to the lane to lane skew of 20/8/6 ns for 2.5/5/8 GT/s so the hardware buffers can re-align the striped data.” (*Wikipedia*)



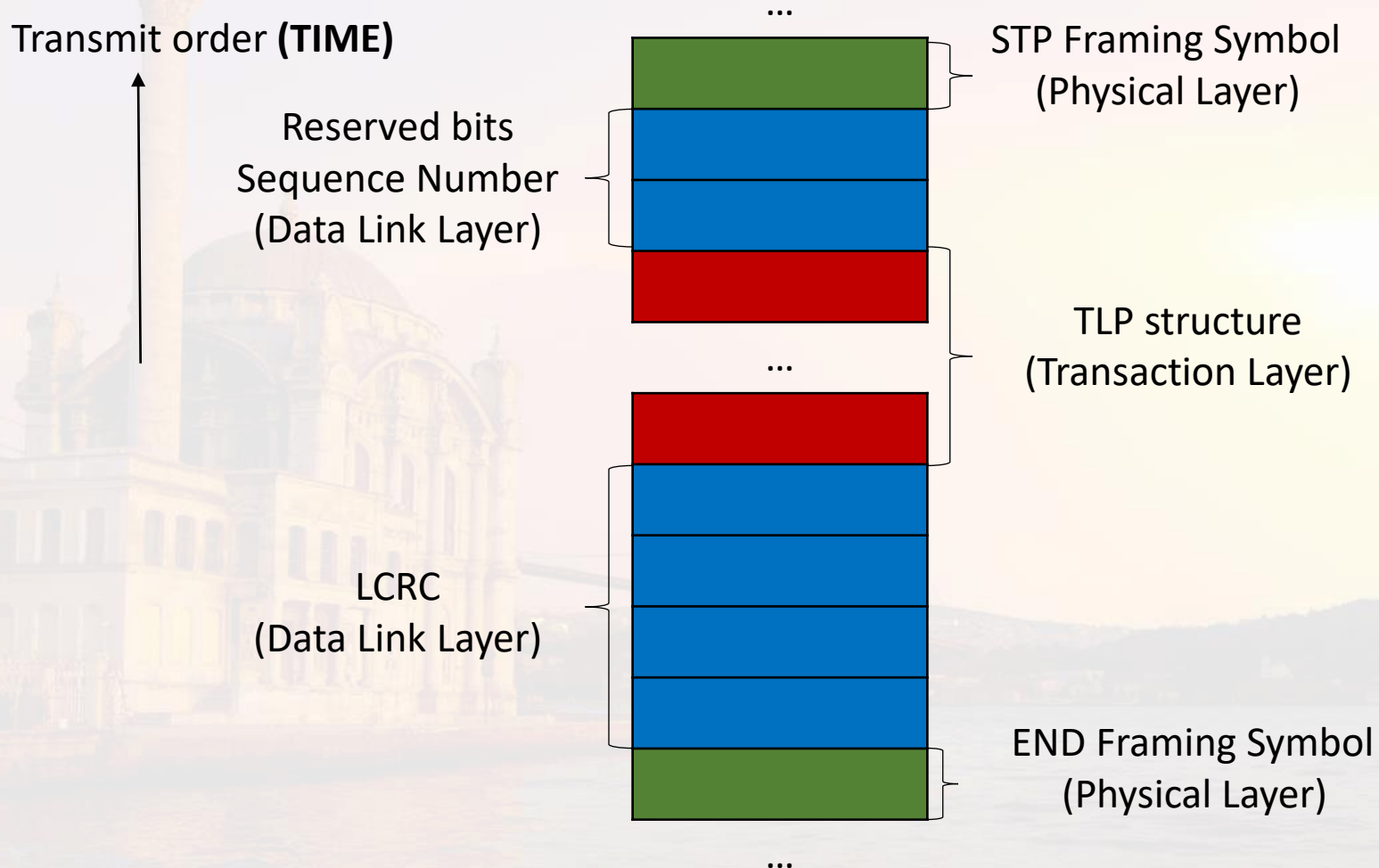
PCIe – Ordered-Set Structure



Six ordered sets are possible

- Training Sequences (TS1, TS2): 1 COM + 15 TS
 - Used to de-skew between lanes
- SKIP: 1 COM + 3 SKP identifiers
 - Used to recalibrate receiver clock
- Fast Training Sequence (FTS): 1 COM + 3 FTS
 - Power management
- Electrical Idle (IDLE): 1 COM + 3 IDL
 - Transmitted continuously when no data
- Electrical Idle Exit (EIEOS): 16 characters (since 2.0)
character: 8 unscrambled bits

PCIe – Framing (x1)



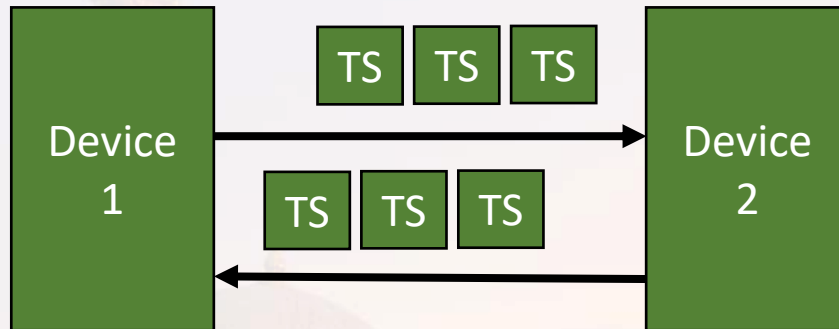
PCIe – Framing (x4)

Transmit order **(TIME)**

Lane order **(SPACE)**



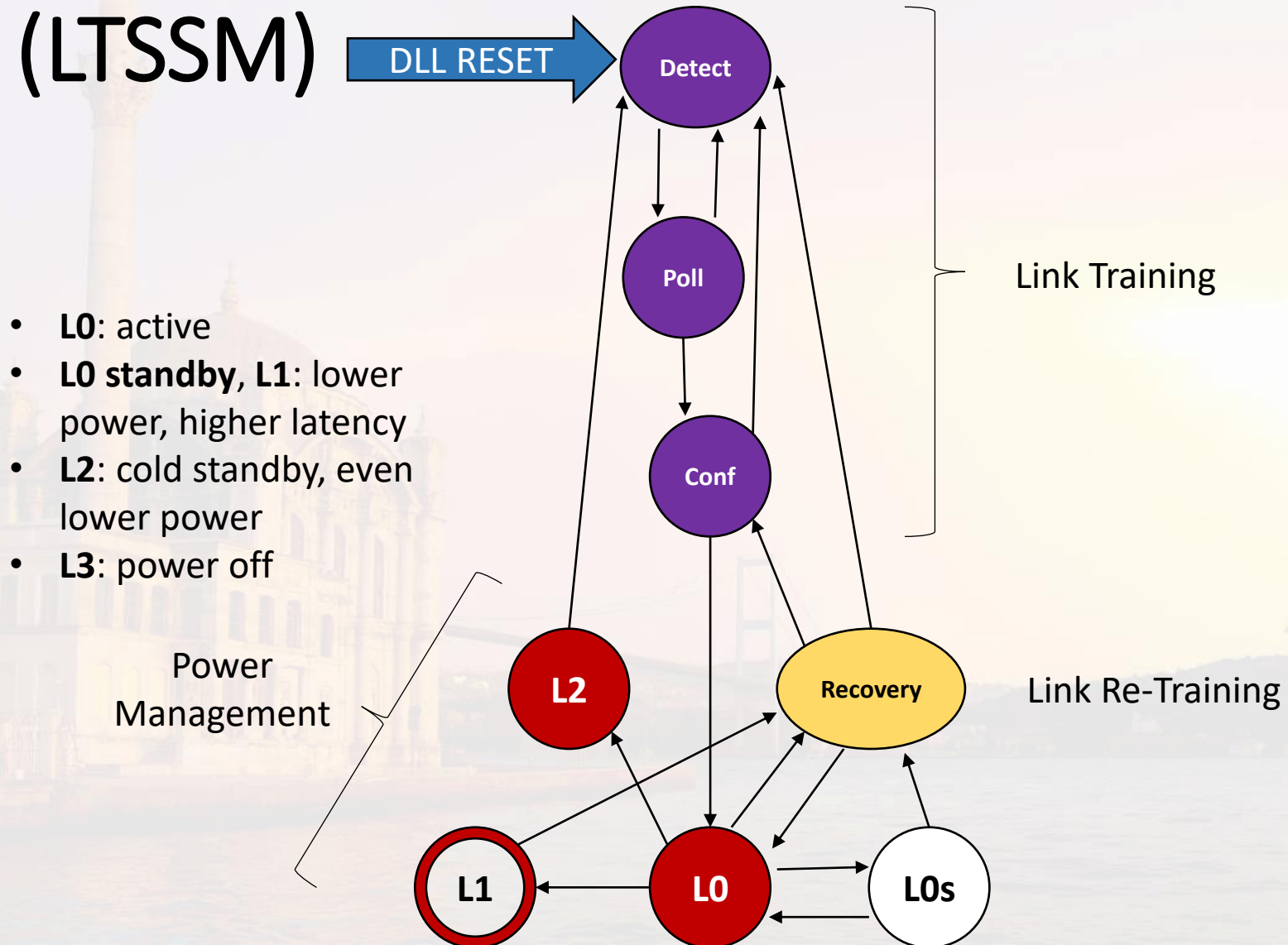
PCIe – Link training



- Lane polarity
- Link width / ordering
- Link equalization
 - Dynamic equalization!
- Link speed
- ...

```
37181 ns EP LTSSM State: RECOVERY.RCVRLOCK
37312 ns RP PCI Express Link Status Register
(1881):
37312 ns Negotiated Link Width: x8
37312 ns Slot Clock Config: System
Reference Clock Used
37949 ns EP LTSSM State: RECOVERY.RCVRCFG
38845 ns RP LTSSM State: RECOVERY.RCVRCFG
41053 ns RP LTSSM State: RECOVERY.SPEED
41309 ns EP LTSSM State: RECOVERY.SPEED
43573 ns EP LTSSM State: RECOVERY.RCVRLOCK
43765 ns RP LTSSM State: RECOVERY.RCVRLOCK
43797 ns RP LTSSM State: REC_EQUALZ.PHASE0
43825 ns RP LTSSM State: REC_EQUALZ.PHASE1
44141 ns EP LTSSM State: REC_EQUALZ.PHASE0
44673 ns EP LTSSM State: REC_EQUALZ.PHASE1
44929 ns RP LTSSM State: REC_EQUALZ.DONE
44949 ns RP LTSSM State: RECOVERY.RCVRLOCK
45209 ns EP LTSSM State: REC_EQUALZ.DONE
45229 ns EP LTSSM State: RECOVERY.RCVRLOCK
45425 ns EP LTSSM State: RECOVERY.RCVRCFG
45581 ns RP LTSSM State: RECOVERY.RCVRCFG
45925 ns RP LTSSM State: RECOVERY.IDLE
46073 ns EP LTSSM State: RECOVERY.IDLE
46169 ns EP LTSSM State: L0
46313 ns RP LTSSM State: L0
47824 ns Current Link Speed: 8.0GT/s
```

PCIe Link-Training State Machine (LTSSM)



Simulate a PCIe link on your own!

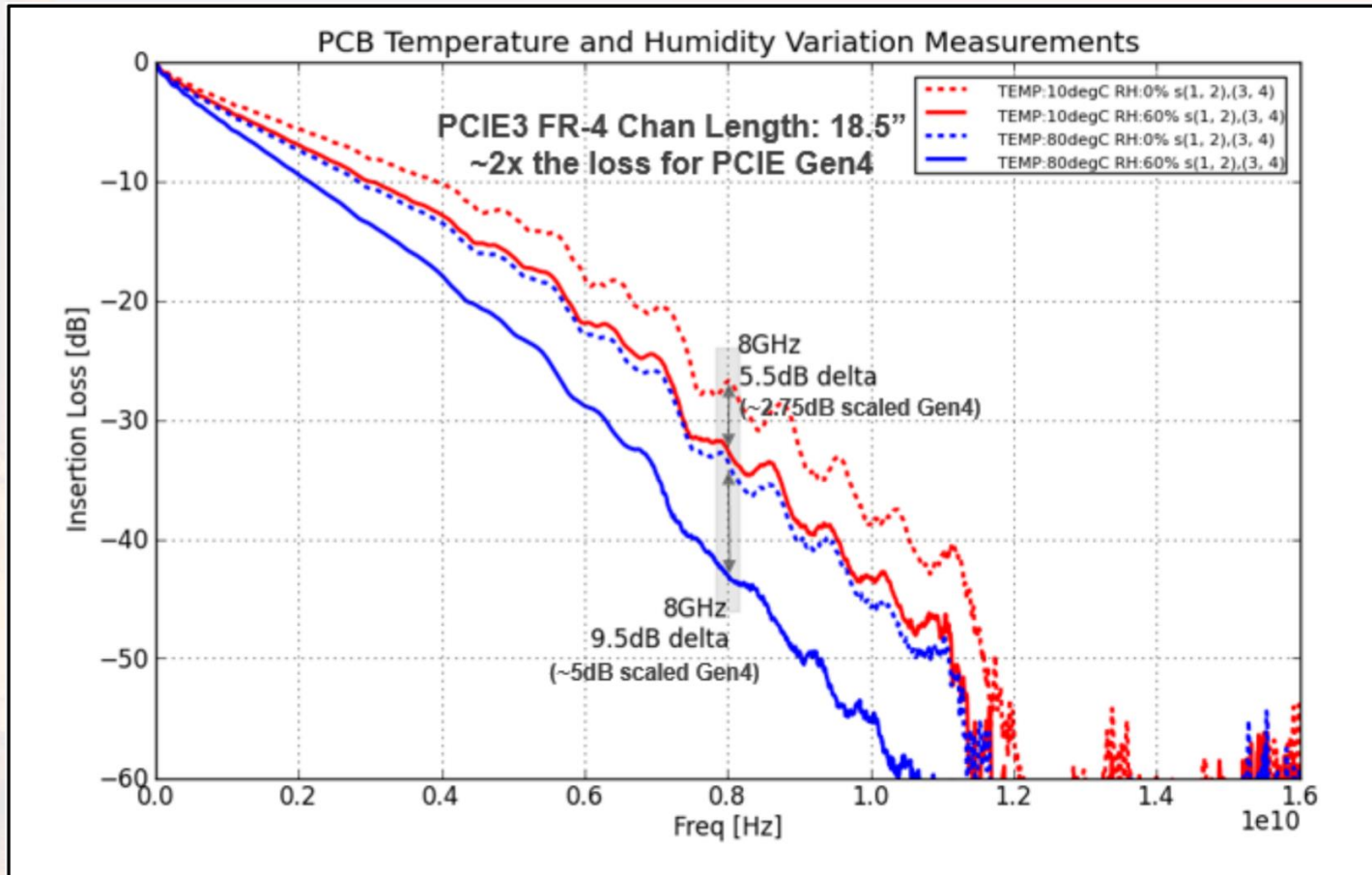
- <https://github.com/wyvernSemi/pcievhost>
- http://www.anita-simulators.org.uk/wyvernsemi/articles/pci_express.pdf
- Written in C/Verilog
- Compatible with ModelSim (via DPI)
- Simulates link training, flow control, ACK/NAK, completions...

What is this presentation about?

- History and evolution of PCIe
- PCIe concepts
- PCIe layers
- **PCIe practical aspects**
- PCIe performance
- PCIe future roadmap

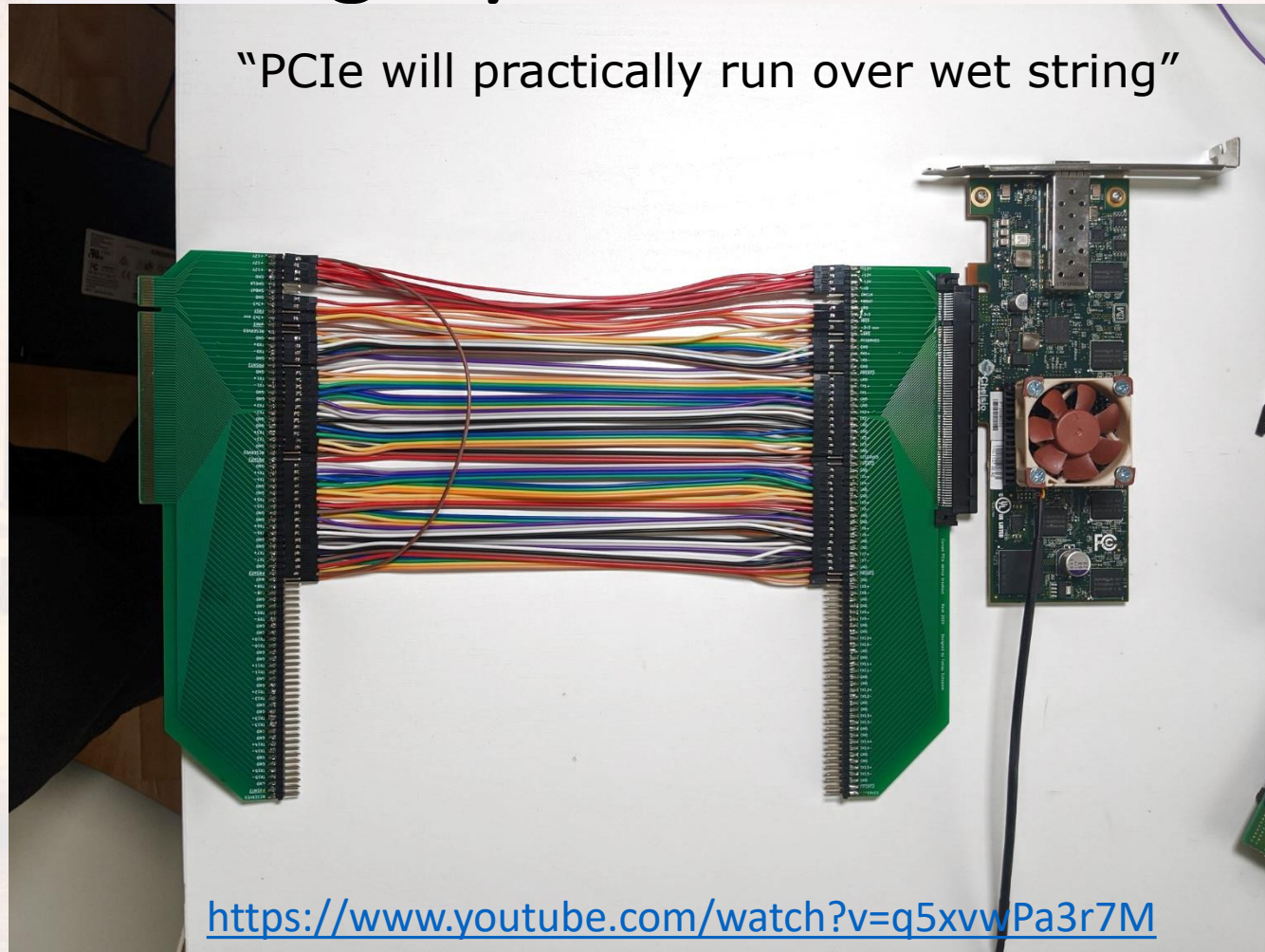
PCIe link training

Signal integrity – Environment



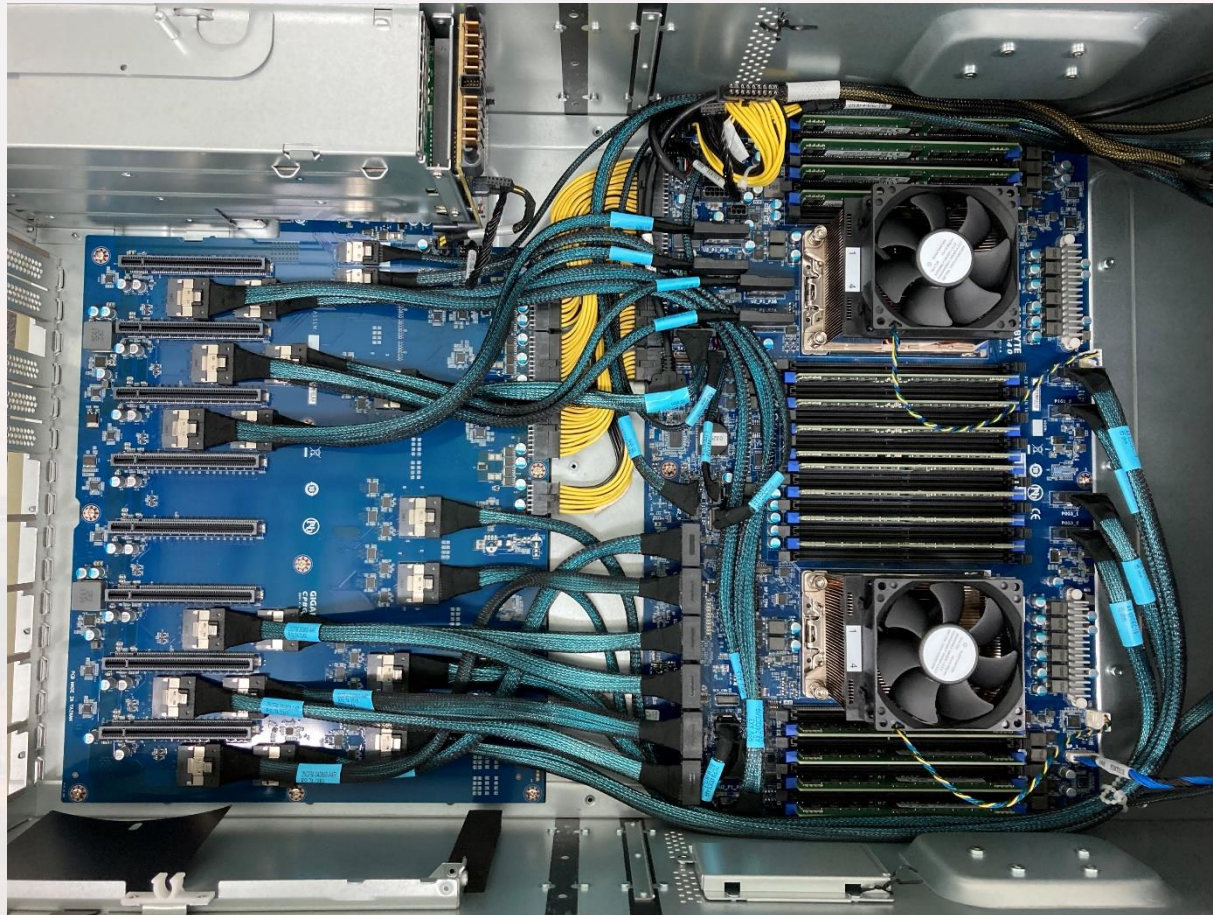
PCIe link training

Signal integrity – Robustness



PCIe link training

Signal integrity – Connectors



Troubleshooting PCIe deployments at scale

If you run a large data acquisition system, and most of your I/O goes through PCIe links, you **have to** monitor all your endpoints and root ports

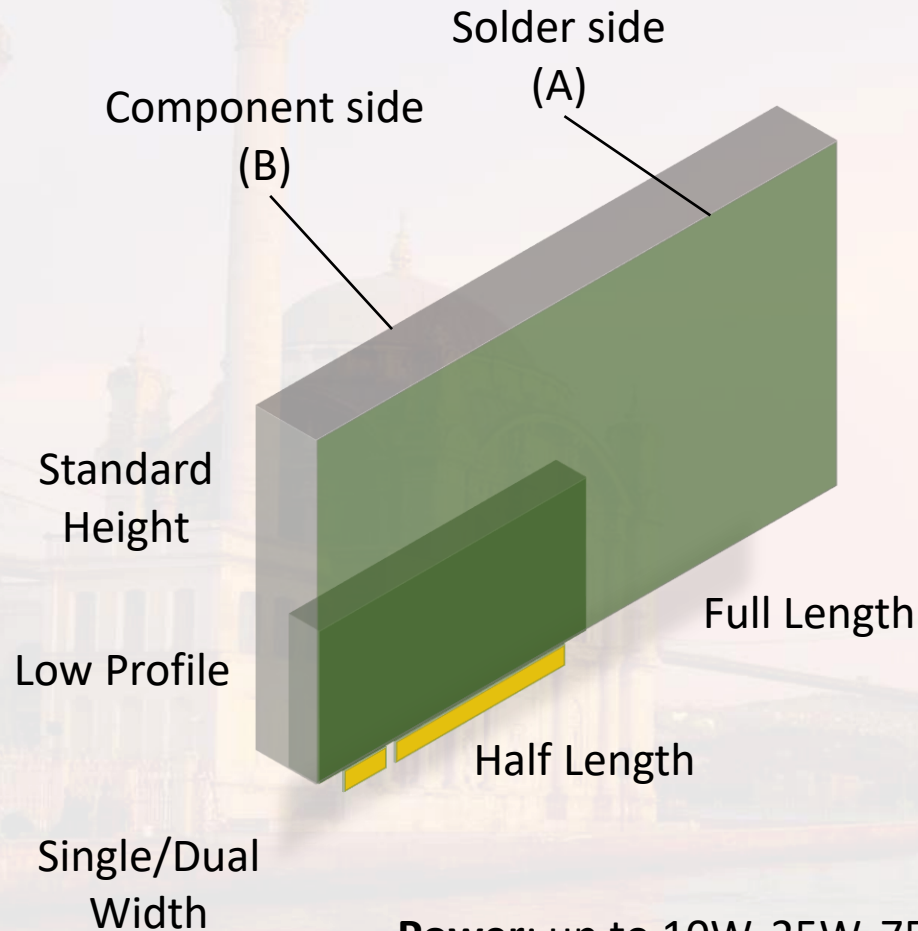
POSTED ON AUGUST 5, 2020 TO [OPEN SOURCE](#)

Pcicrawler: A Python-based command-line interface tool to debug PCI issues at scale

<https://github.com/facebook/pcicrawler>

<https://engineering.fb.com/2020/08/05/open-source/pcicrawler>

PCIe CEM Spec – AIC form factors



- Standard Height
 - 4.20" (106.7mm)
- Low Profile
 - 2.536" (64.4mm)
- Half Length (e.g. "HHHL")
 - 6.6" (167.65mm)
- Full Length (e.g. "FHFL")
 - 12.283" (312mm)

Power: up to 10W, 25W, 75W, 300W or 375W depending on form factor & optional extra power connectors

PCIe storage – More form factors

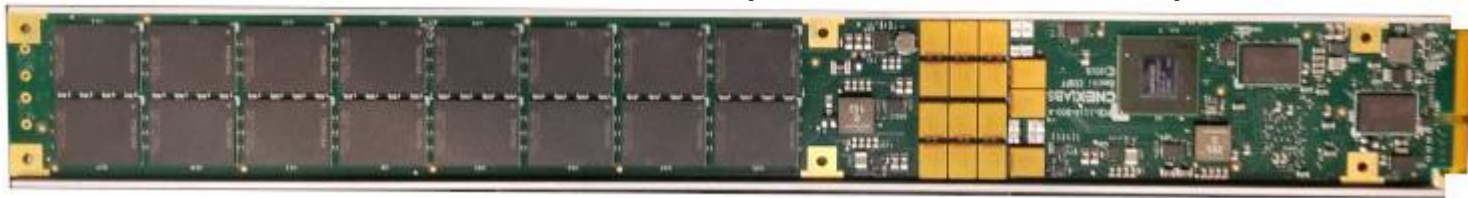


M.2
 ≤ 4 lanes

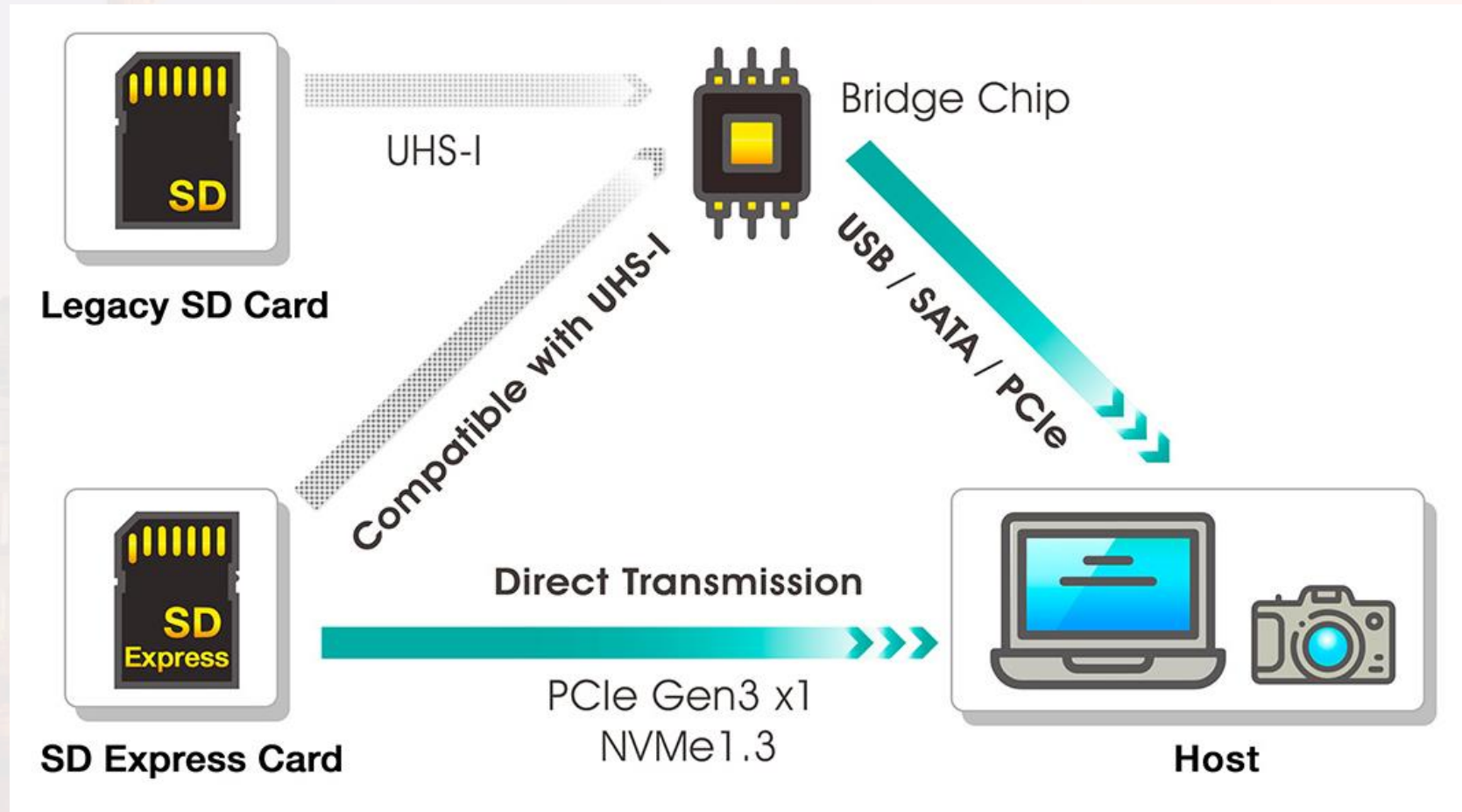


U.2
 ≤ 4 lanes

“ruler” (EDSFF, NGSFF) ≤ 8 lanes

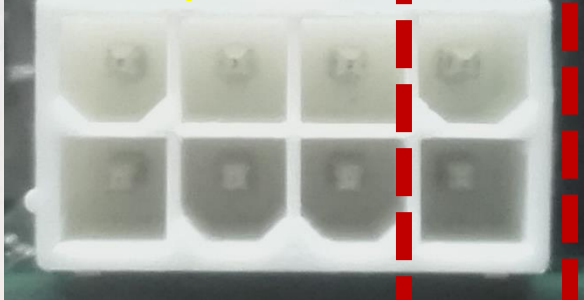


PCIe storage – SD Express

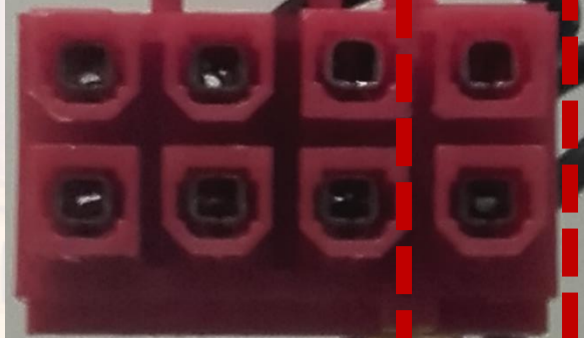


PCIe CEM Spec – Power Cables

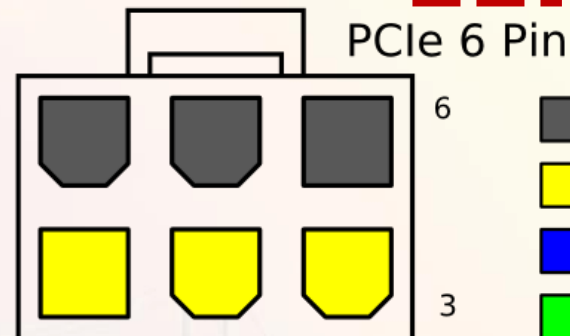
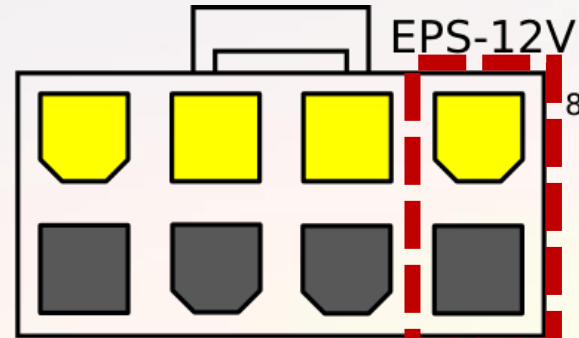
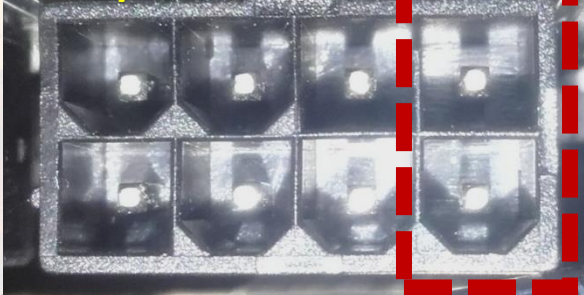
EPS receptacle



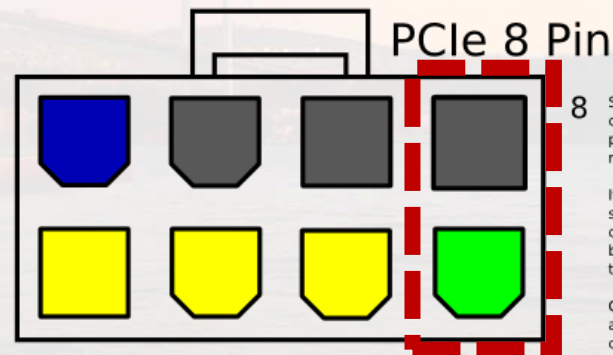
PCIe cable



GPU power



- Gnd
- +12 V
- Sense A
- Sense B



Sense A and B are used by a compatible power supply to provide enhanced voltage regulation.

If enhanced regulation is not supported then Sense A can be connected to Ground. Sense B can be left unconnected (or connected to ground).

Connection of ground to Pin 8 allows the card to detect an 8 pin connector, and select enhanced power mode



PCIe CEM Spec – Power Cables

Solutions

Products

Company



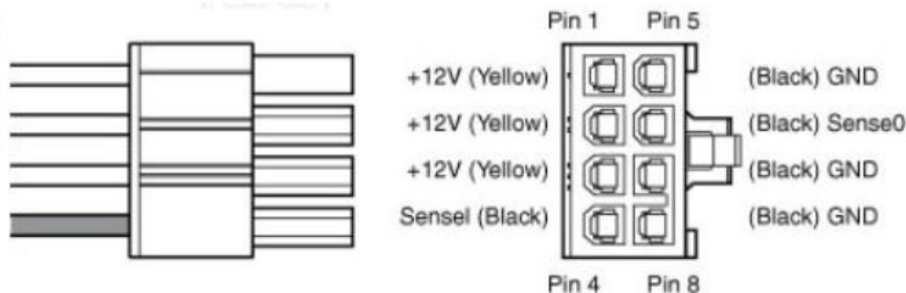
72298 - Alveo Data Center Accelerator Card - !CAUTION! Do not use EPS12V / ATX12V power source in place of PCI Express Auxiliary Power connector

🕒 Feb 16, 2023 · Knowledge

https://support.xilinx.com/s/article/72298?language=en_US

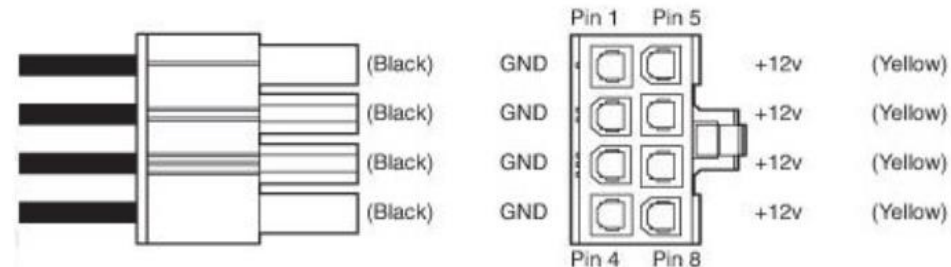
PCI Express Auxiliary 8-pin Power Cable pinout

For use with Alveo Data Center Accelerator Card



CPU 12V (EPS12V / ATX12V) 8-pin Power Cable pinout

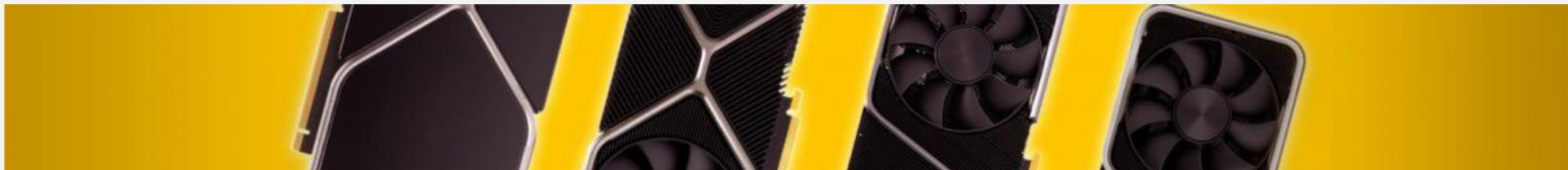
DO NOT USE with an Alveo Data Center Accelerator Card



PCIe – GPU power limits

Rumored power limit for NVIDIA AD102 GPU is 800W, AD103 up to 175W

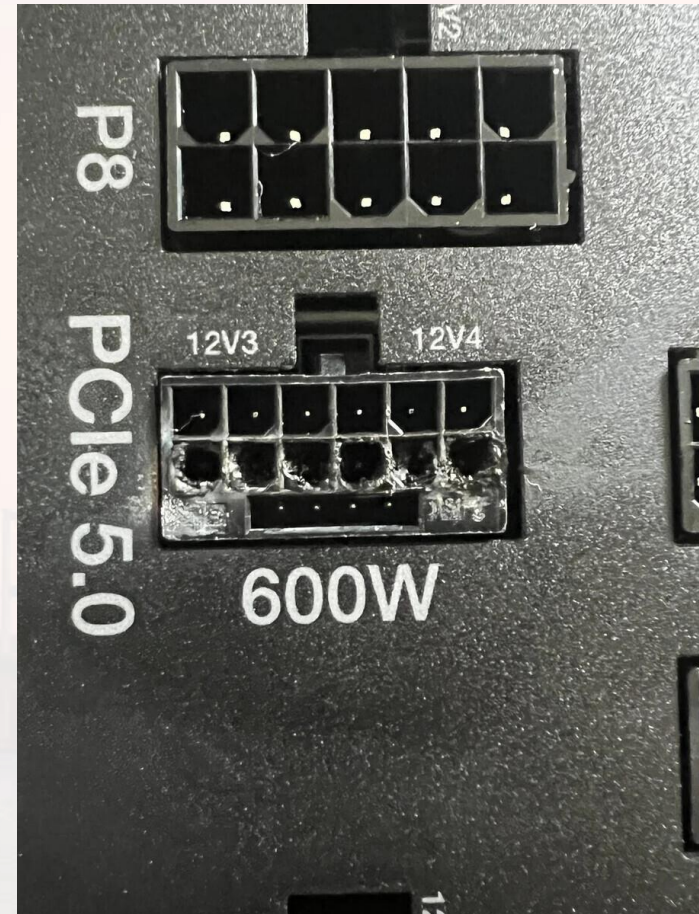
NVIDIA Ada GPU architecture of consumer GPUs is expected to have an increased TGP targets across the stack.



NVIDIA GeForce RTX 40 Power limits					
VideoCardz.com	SMs	CUDAs	Memory Bus Width	Power Limit (Desktop)	Max TGP (Mobile) with Dynamic Boost
AD102	144	18432	384-bit	800W	–
AD103	84	10752	256-bit	450W	175W
AD104	60	7680	192-bit	400W	175W
AD106	36	4608	128-bit	260W	140W
AD107	24	3072	128-bit	TBC	TBC

<https://videocardz.com/newz/nvidia-rtx-40-ada-gpu-power-limits-rumored-to-reach-800w-on-desktop-and-175w-on-laptops>

PCIe – 12VHPWR connector

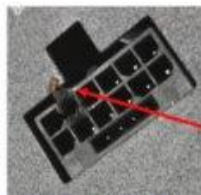


PCIe – 12VHPWR power failures

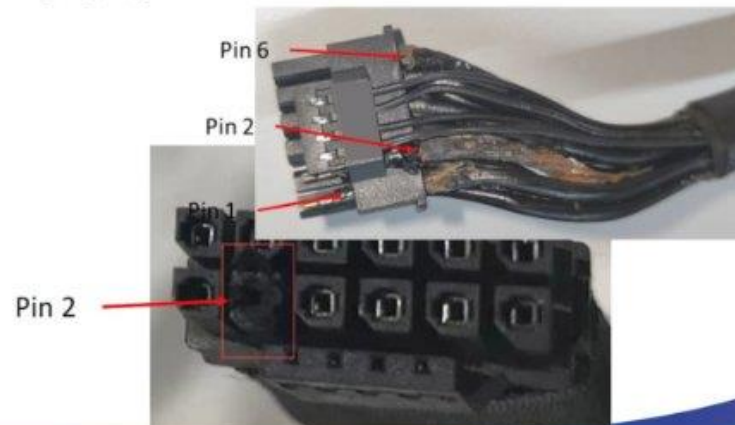
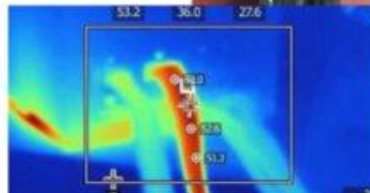
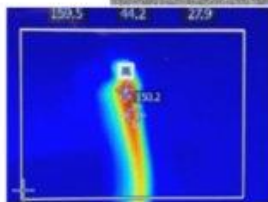
Observations



- Images below show overheating of the power connector at mating point. Multiple suppliers and designs have failed
- Cables with low cycles and without bend condition have not failed
- Failures observed on both rows of pins depending on load direction
- Hot spots observed @~2.5hrs, melting 10-30hrs
- Note - Also observed after high mating cycles ~40, straight plug w/o side load



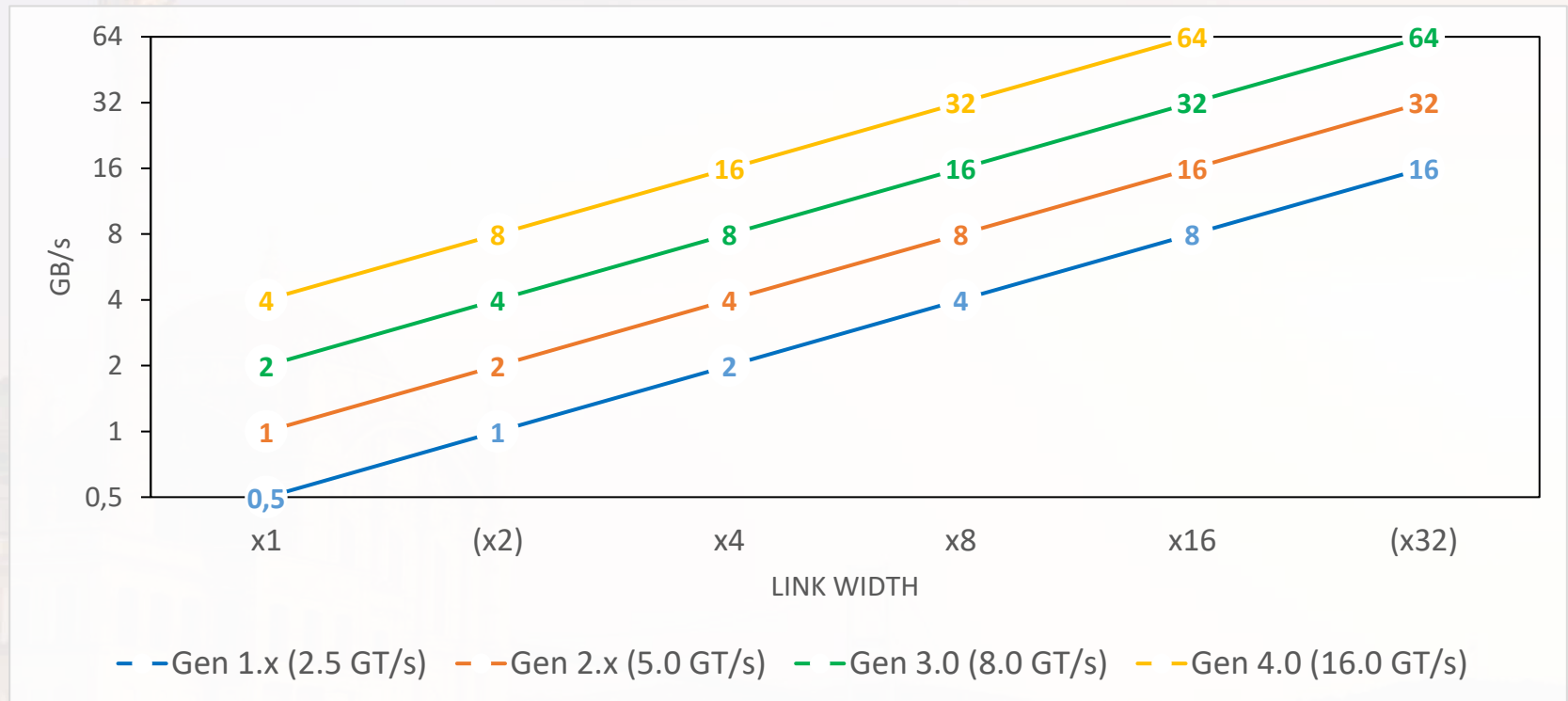
Pin 11



What is this presentation about?

- PCIe history and evolution
- PCIe concepts
- PCIe layers
- PCIe practical aspects
- PCIe performance
- PCIe future roadmap

PCIe – Theoretical data rates



- “Aggregate” bandwidth in both directions
- Considering 20% encoding overhead in 1.x and 2.x

PCIe – Theoretical data rates

PCIe® Speeds/Feeds - Pick Your Bandwidth

- Flexible to meet needs from handheld/client to server/HPC
- ~Max Total Bandwidth = Max RX bandwidth + Max TX bandwidth
- 35 Permutations yielding 11 unique bandwidth profiles
- Encoding overhead and header efficiency not included

Specifications	Lanes				
	x1	x2	x4	x8	x16
2.5 GT/s (PCIe 1.x +)	500 MB/S	1 GB/S	2 GB/S	4 GB/S	8 GB/S
5.0 GT/s (PCIe 2.x +)	1 GB/S	2 GB/S	4 GB/S	8 GB/S	16 GB/S
8.0 GT/s (PCIe 3.x +)	2 GB/S	4 GB/S	8 GB/S	16 GB/S	32 GB/S
16.0 GT/s (PCIe 4.x +)	4 GB/S	8 GB/S	16 GB/S	32 GB/S	64 GB/S
32.0 GT/s (PCIe 5.x +)	8 GB/S	16 GB/S	32 GB/S	64 GB/S	128 GB/S
64.0 GT/s (PCIe 6.x +)	16 GB/S	32 GB/S	64 GB/S	128 GB/S	256 GB/S
128.0 GT/s (PCIe 7.x +)	32 GB/S	64 GB/S	128 GB/S	256 GB/S	512 GB/S

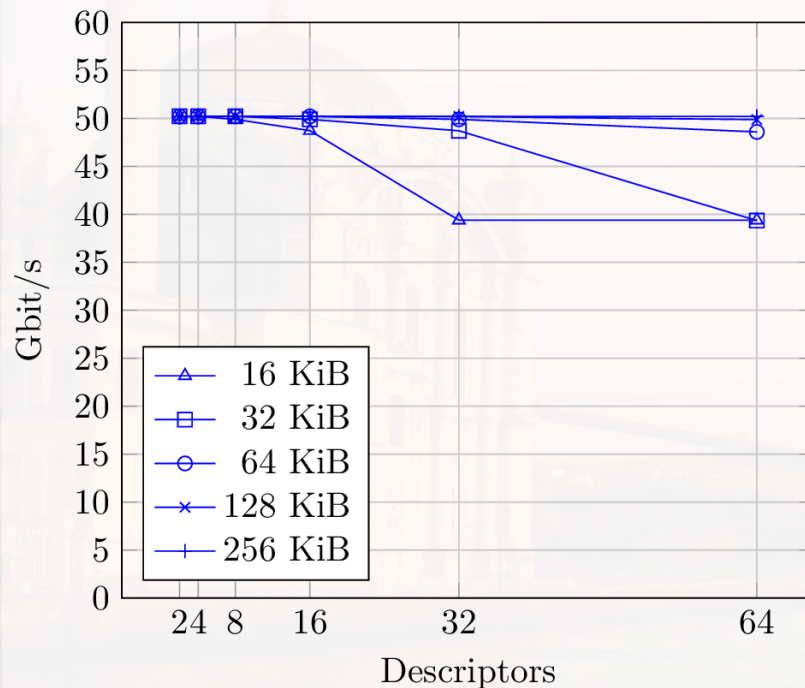
+ = data rate supported by this and subsequent spec revisions.

PCIe – Effective data rates

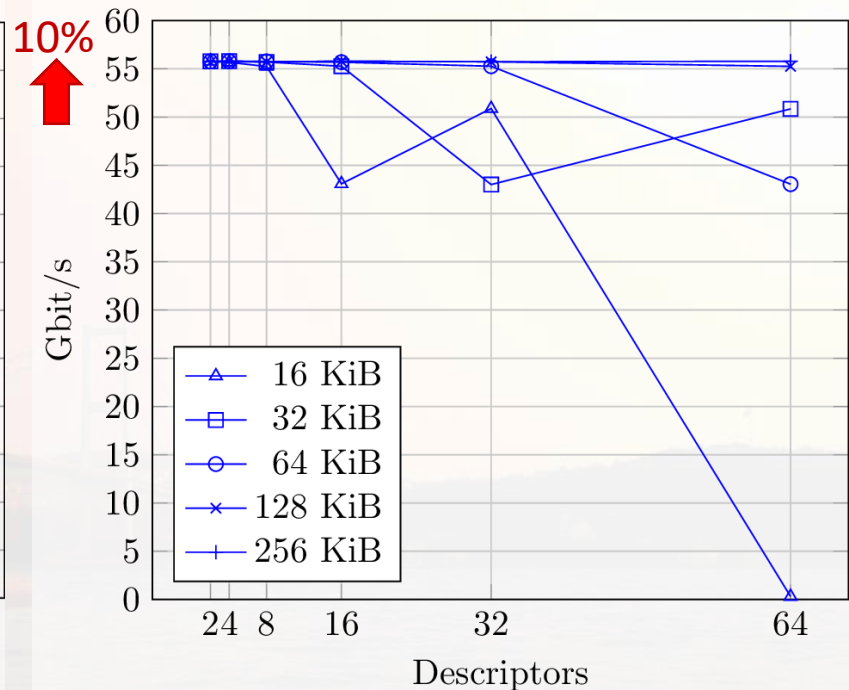
- $$\rho = \overbrace{\frac{\text{Lane rate} \times \text{Lane width}}{\text{Encoding}}}^{\text{Theoretical bandwidth}} \times \overbrace{\frac{\text{MPS}}{\text{MPS} + \text{Headers}}}^{\text{Packet efficiency}}$$
- Example: Gen2 x8, 128 Bytes MPS
 - $\rho = 40 \times 0.8 \times \frac{128}{128+24} = 32 \times 0.84 = 26.9 \text{ Gb/s}$
- Example: Gen3 x8, 128 Bytes MPS
 - $\rho = 64 \times 0.98 \times \frac{128}{128+24} = 62.7 \times 0.84 = 52.6 \text{ Gb/s}$
- Example: Gen3 x8, 256 Bytes MPS
 - $\rho = 64 \times 0.98 \times \frac{256}{256+24} = 62.7 \times 0.91 = 57 \text{ Gb/s}$

PCIe 3.0 x8 – DMA Performance

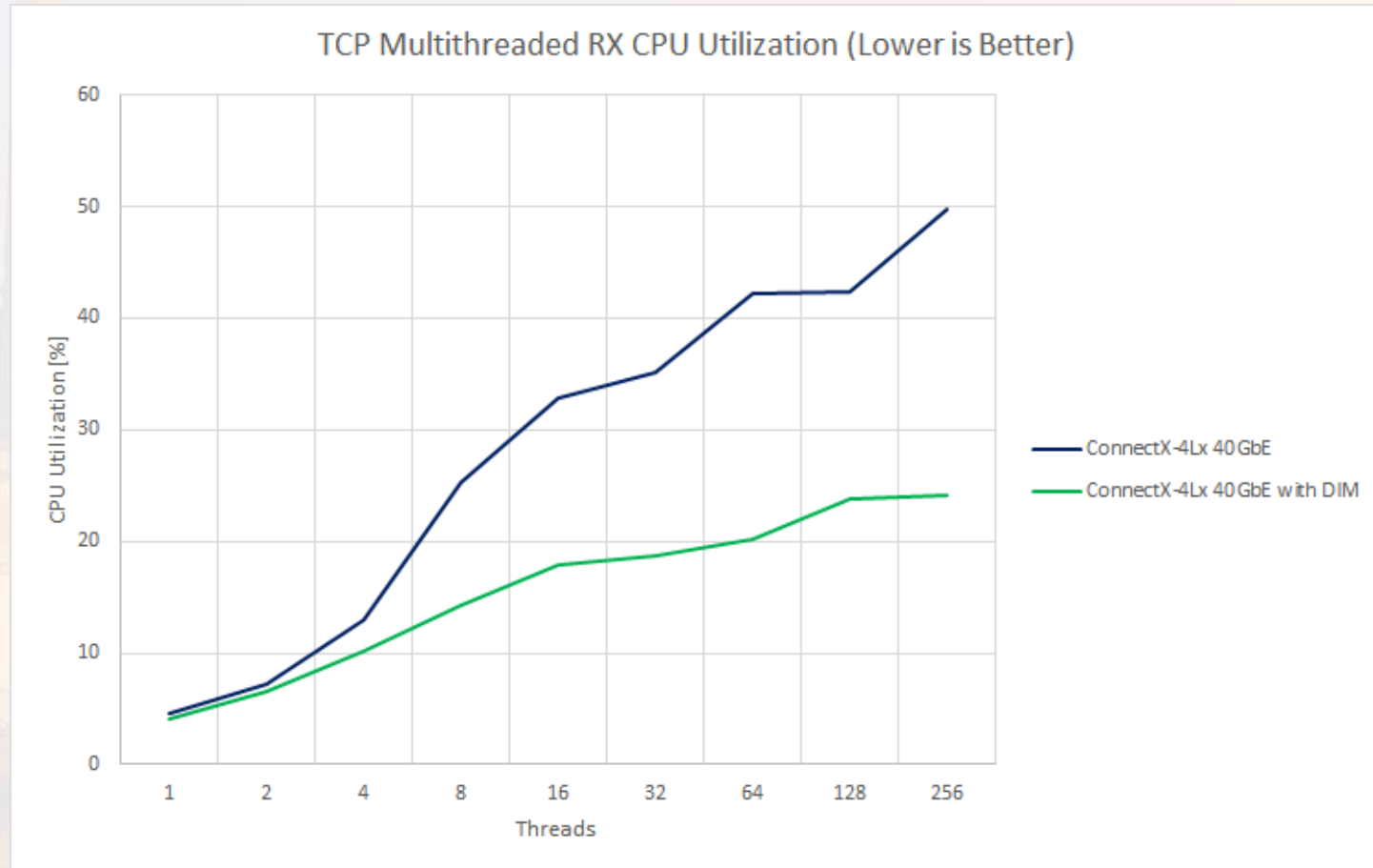
MPS = 128 Bytes



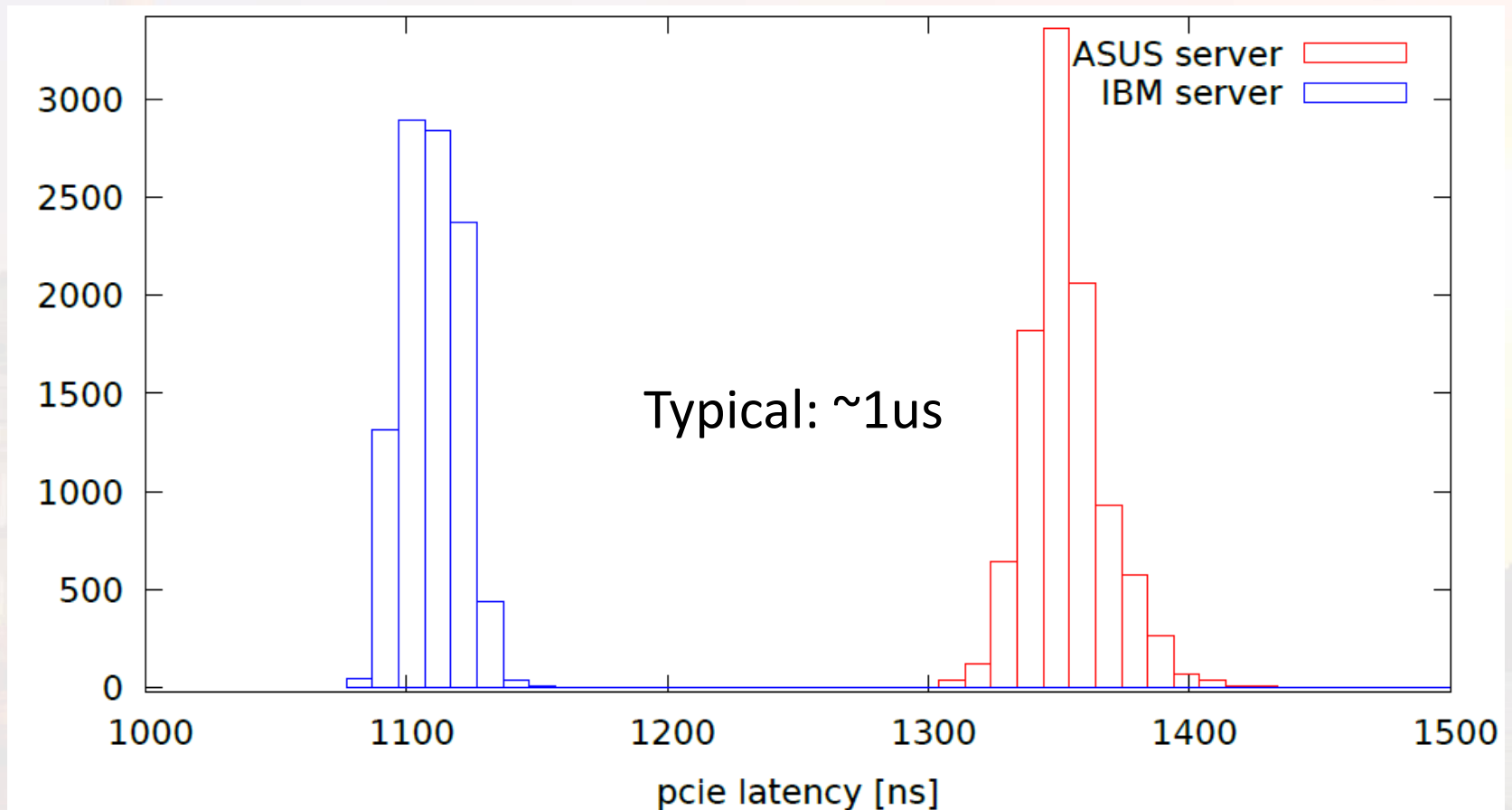
MPS = 256 Bytes



PCIe performance – int. coalescing



PCIe performance – latency

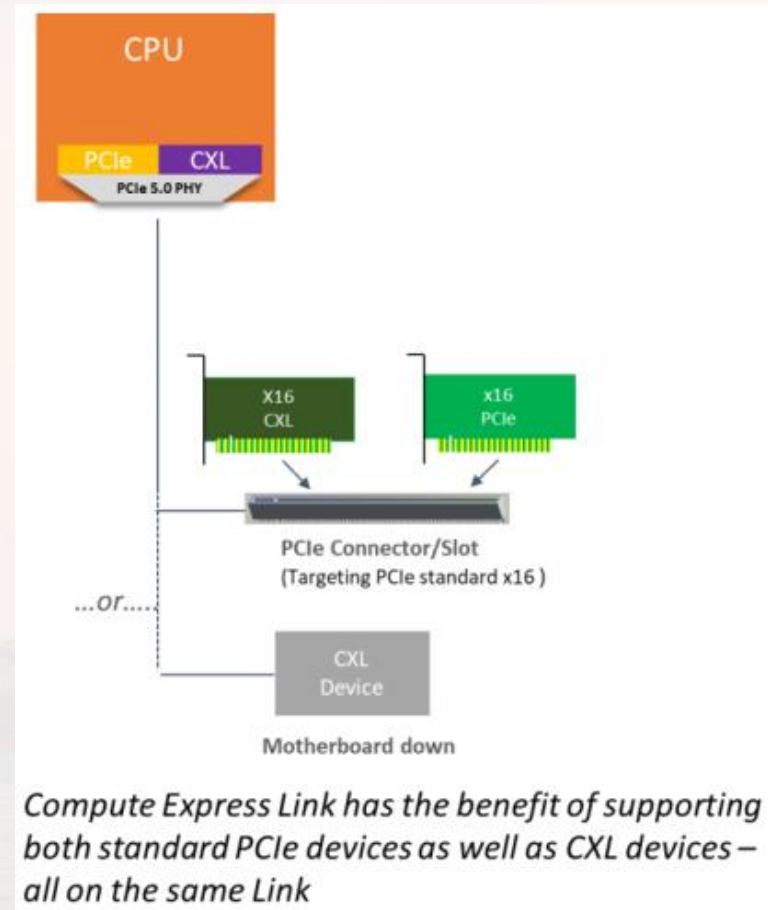


What is this presentation about?

- PCIe history and evolution
- PCIe concepts
- PCIe layers
- PCIe practical aspects
- PCIe performance
- **PCIe future roadmap**

What is Compute Express Link?

- Alternate protocol that runs across the standard PCIe physical layer
- Uses a flexible processor port that can auto-negotiate to either the standard PCIe transaction protocol or the alternate CXL transaction protocols
- First generation CXL aligns to 32 Gbps PCIe 5.0
 - 8 Gbps in degraded mode



CXL Consortium

- Alibaba, Cisco, Dell EMC, Facebook, Google, Hewlett Packard Enterprise, Huawei, Intel Corporation and Microsoft announced their intent to incorporate in March 2019.
- The Compute Express Link (CXL) Consortium and Gen-Z Consortium developed an execution of a Memorandum of Understanding (MOU), describing a mutual plan for collaboration between the two organizations in April 2020.



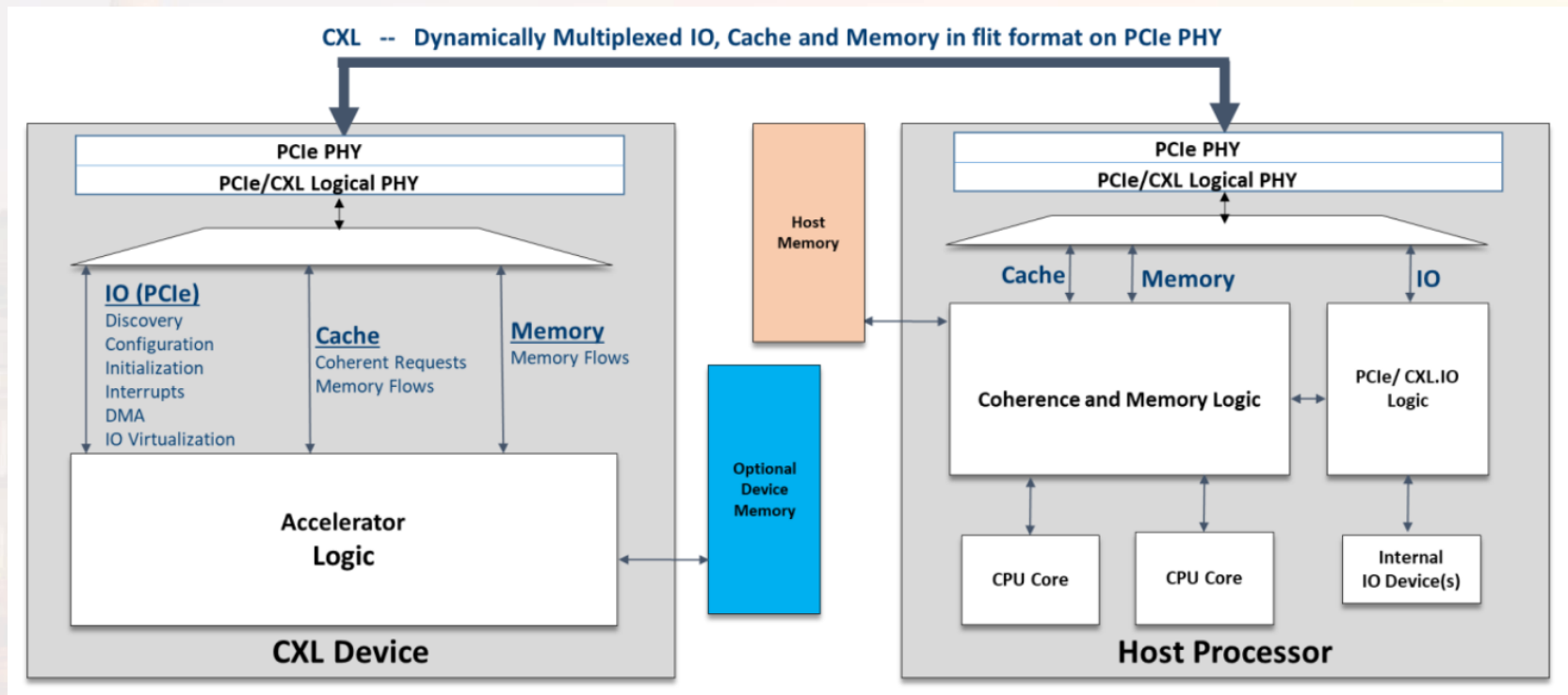
Why CXL?

Need a new class of interconnect for **heterogenous computing** and **disaggregation** usages:

- Efficient resource sharing
- Shared memory pools with efficient access mechanisms
- Enhanced movement of operands and results between accelerators and target devices
- Significant latency reduction to enable disaggregated memory

CXL – Dynamic Multiplexing

CXL multiplexes three different protocols at the PCIe PHY layer



CXL protocols

- **cxl.io**

- device discovery, configuration, initialization, I/O virtualization, and direct memory access (DMA)

- **cxl.cache**

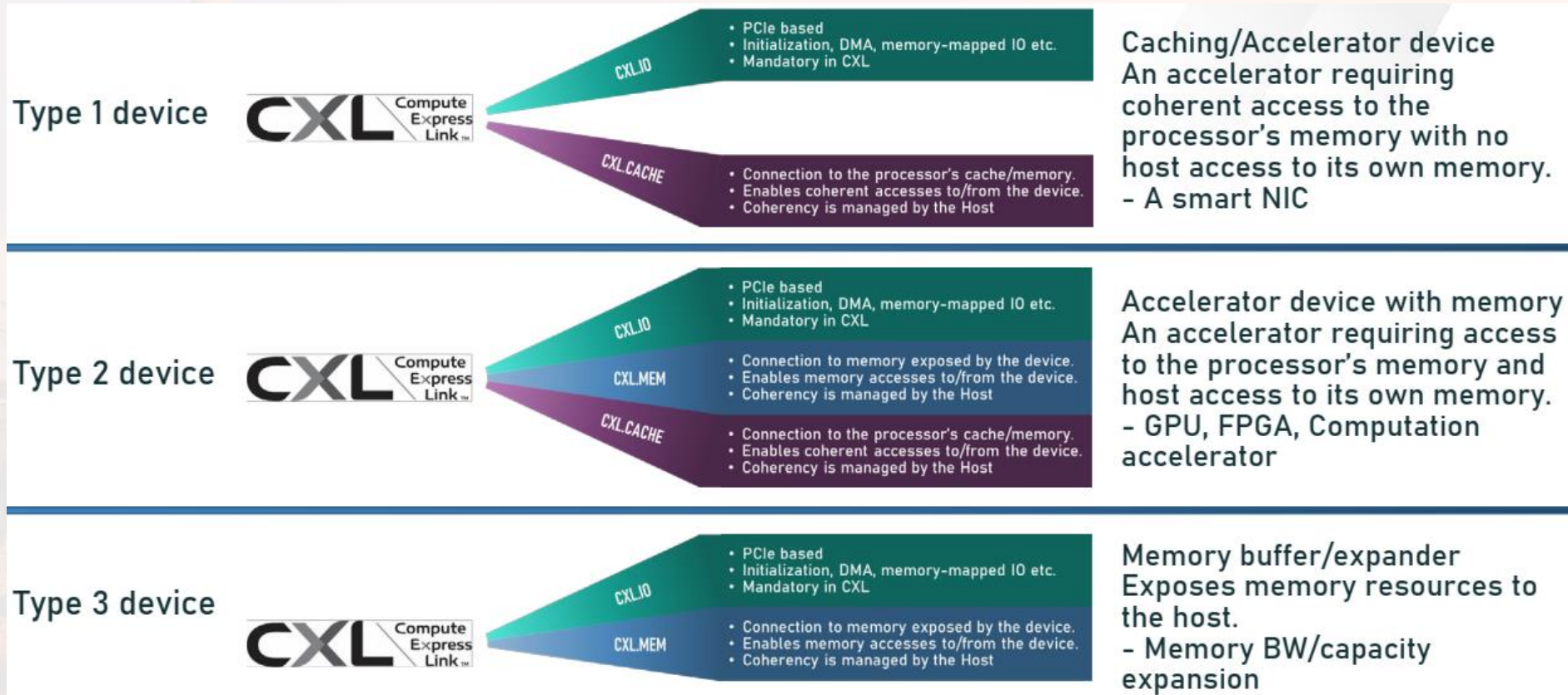
- enables a device to cache data from the host memory, employing a simple request and response protocol
- the host processor manages coherency of data

- **cxl.memory**

- allows a host processor to access memory attached to a CXL device

CXL device types

“Mix and match” protocols depending on application requirements

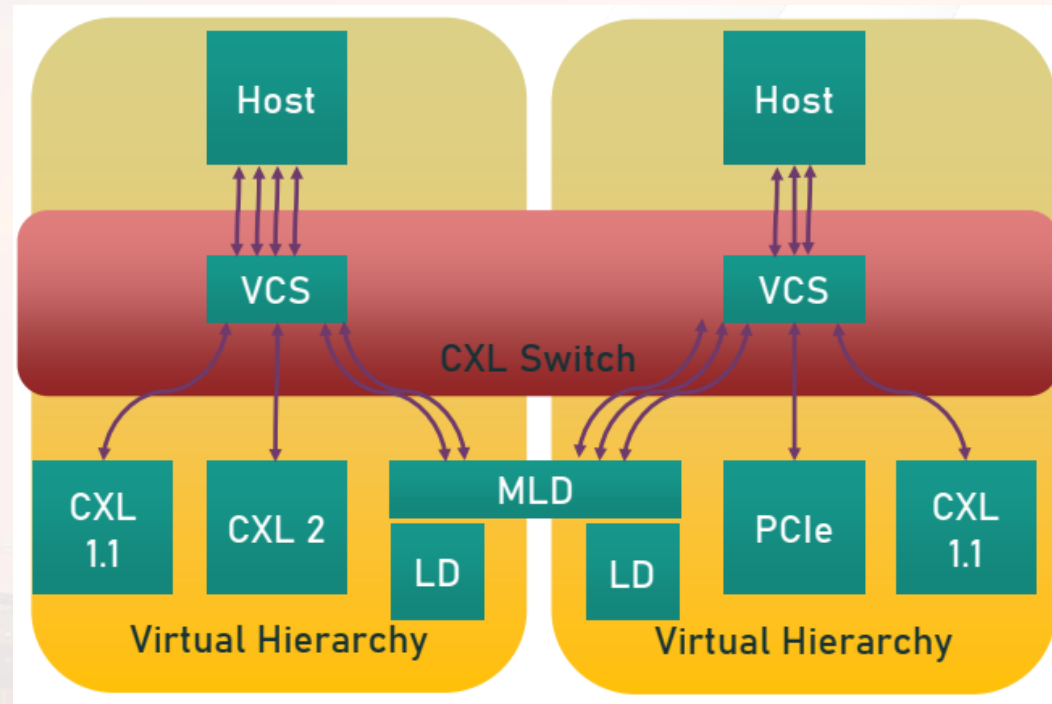


CXL evolution timeline

- CXL 1.0 – March 2019
 - enables device-level memory expansion and coherent acceleration modes
- CXL 1.1 – September 2019
- CXL 2.0 – November 2020
 - augments CXL 1.1 with enhanced fanout support and a variety of additional features
- CXL 3.0 – in the making
- CXL supporting platforms coming to market now

CXL 2.0 new features

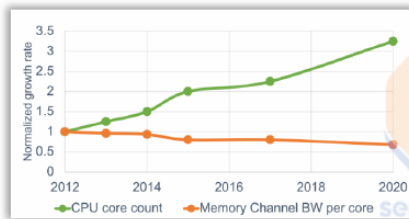
- CXL switches
 - Multiple host
 - Virtual hierarchies
- Multi-Logic devices
- Management
 - Fabric manager
 - Device allocation
 - QoS telemetry
 - Memory interleaving



CXL-attached memory

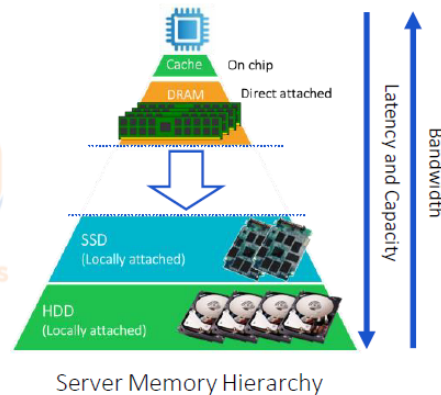
Summary of Data Center Memory Challenges

Decreasing memory bandwidth per core

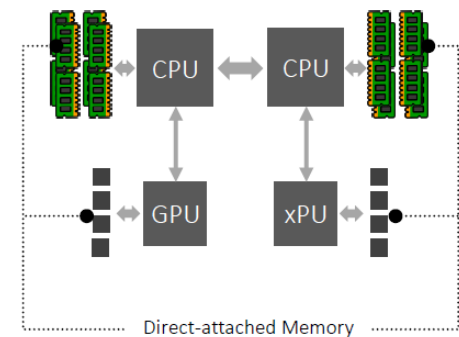


Source: Meta, OCP Summit Presentation Nov. '21

Huge latency and capacity gap



Stranded memory resources and low utilization



CXL-attached memory is the current focus of the CXL ecosystem

Vendor CPU support

- Intel
 - Sapphire Rapids ('23)
 - Emerald Rapids ('23?)
 - PCIe Gen 5.0
 - CXL 1.1 (only type 1 & 2)
- IBM
 - Power 10 ('21)
 - PCIe Gen 5.0
 - Coherent Accelerator Processor Interface (libcxl)
 - OpenCAPI Memory Interface (OMI)
- AMD
 - Genoa (Zen 4) ('23)
 - PCIe Gen 5.0
 - CXL 1.1 (only type 3)
- ARM
 - Graviton3 (AWS) ('22)
 - PCIe Gen 5.0
 - Grace (NVIDIA) ('23)
 - PCIe Gen 5.0
 - CXL 2.0
 - One (Ampere) ('23)
 - PCIe Gen 5.0

Conclusions

- PCIe has a track record of 2x throughput improvements per generation
- PCIe has maintained backwards compatibility for decades
- PCIe has won the interconnect wars
 - Gen-Z has joined the CXL consortium
 - All CCIX consortium members have moved to CXL
 - Future NVLink was announced to be CXL-compatible
 - CAPI never gained mindshare outside of IBM
- PCIe is proving suitable also for chip-to-chip comms
 - Universal Chiplet Interconnect Express (UCIe)

