

PCIe Switch Applications in ML/AI



A Leading Provider of Smart, Connected and Secure Embedded Solutions



SMART | CONNECTED | SECURE

Tam Do

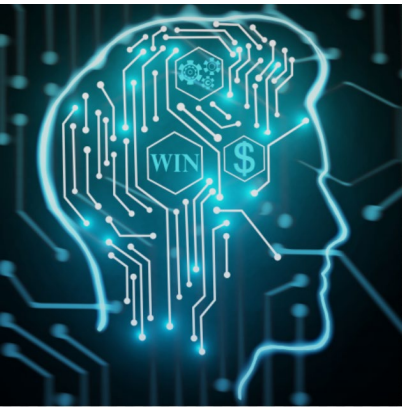
Aug, 2022

Topics

- PCIe and Switch Basics
- Switch Functionalities
- Latency
- AI/ML use case examples
- Summary

PCIe® is Everywhere

PCIe® technology is pervasive in systems such as CPUs, GPUs, NICs, SSDs, FPGA, and other multiple end points that require low latency, high performance interconnect for diverse applications.



AI/ML



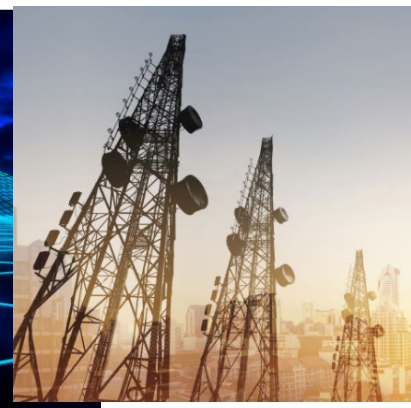
**Cloud Infrastructure
Enterprise Storage**



Cloud Gaming



Military Systems



**Telecommunications
Systems**



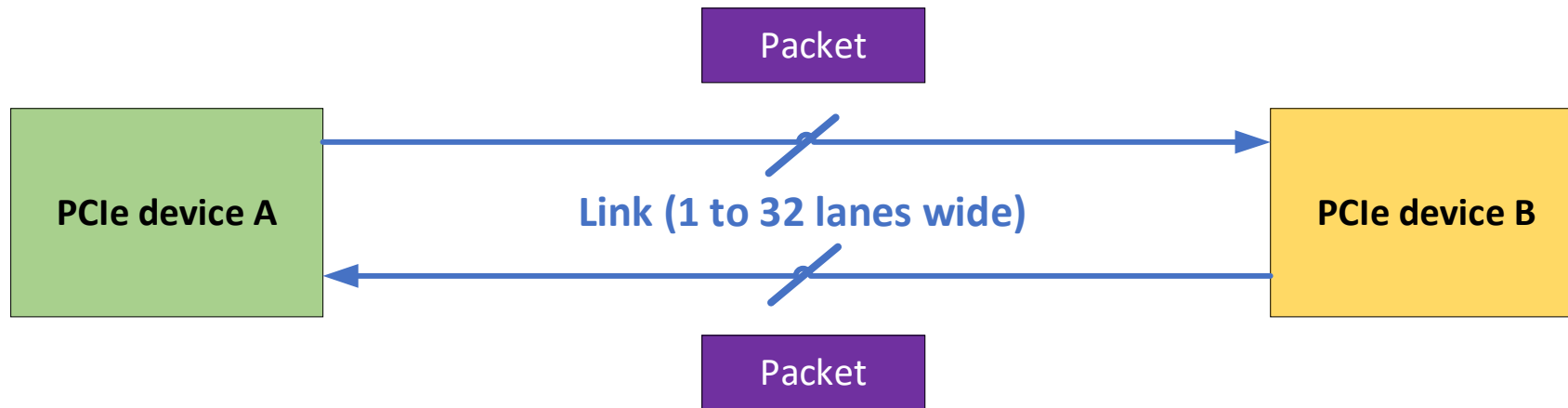
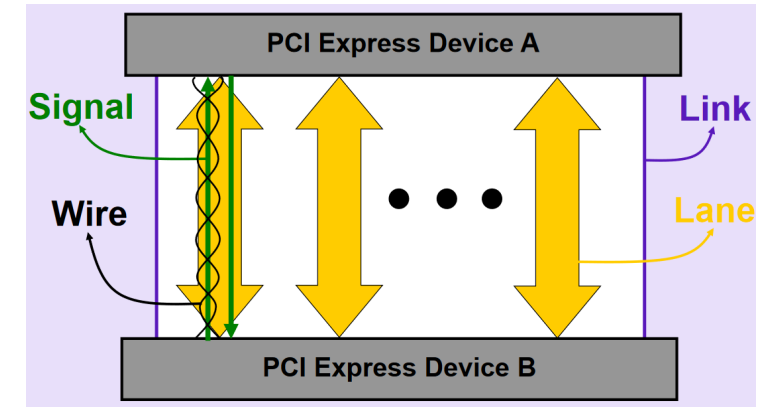
**Automotive Self-Driving
Cars**

...and much more

PCIe switch serves as the bridging interface between hosts and Endpoints

PCIe® Express Basics

- Serial point-to-point communication bus
- Scalable Link width: x1, x2, x4, x8, x12, x16, x32 lanes
- Symmetric: same number of lanes in each direction
- Dual-Simplex connection, transfers in both direction simultaneously
- Packet-based transaction protocol



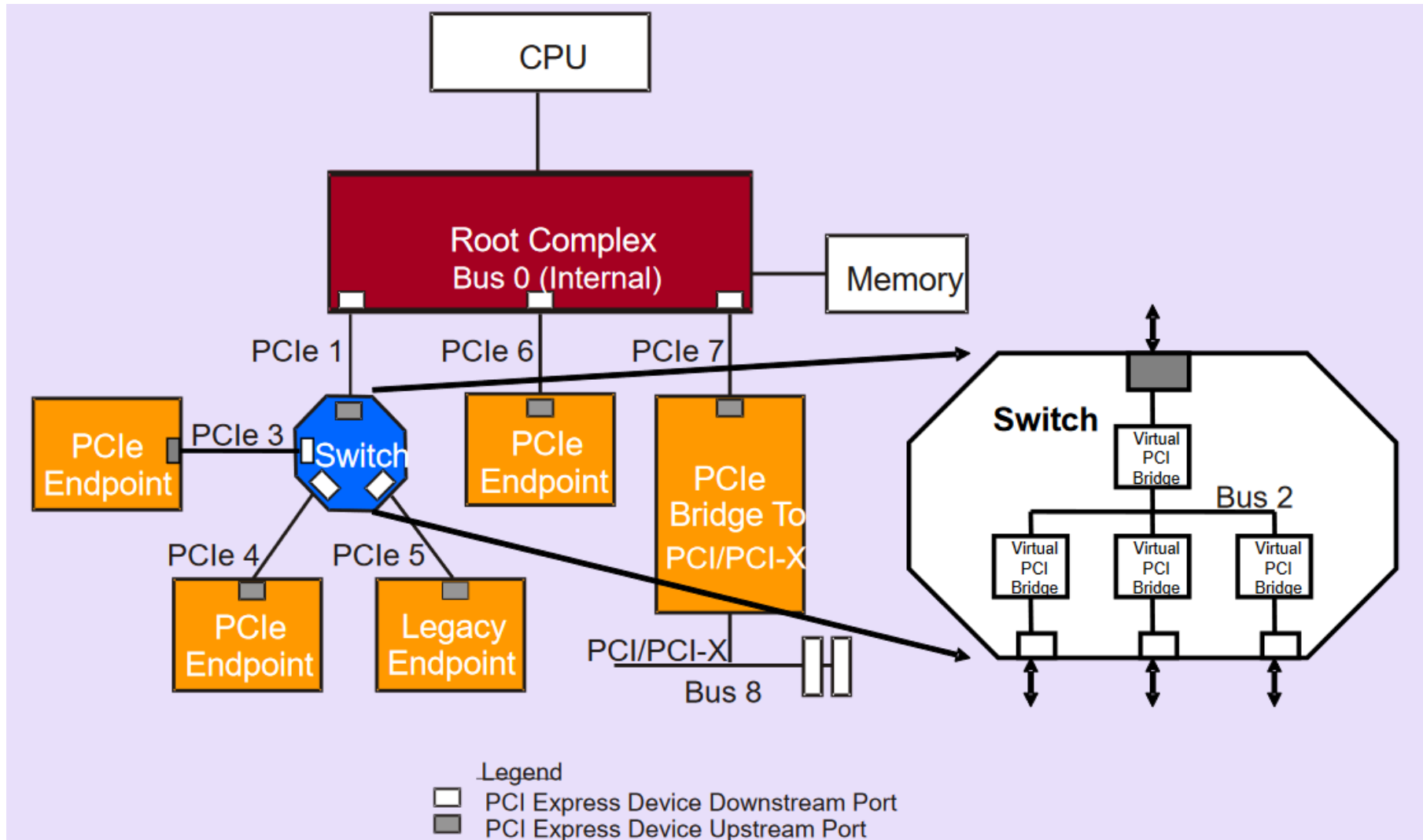
PCIe® Data Rate

Link Width

Aggregate BW (GB/s)	x1	x2	x4	x8	x12	x16	x32
2.5 GT/s (Gen1)	0.5	1	2	4	6	8	16
5.0 GT/s (Gen2)	1	2	4	8	12	16	32
8.0 GT/s (Gen3)	2	4	8	16	24	32	64
16.0 GT/s (Gen4)	4	8	16	32	48	64	128
32.0 GT/s (Gen5)	8	16	32	64	96	128	256
64.0 GT/s (Gen6)	16	32	64	128	256	512	1024

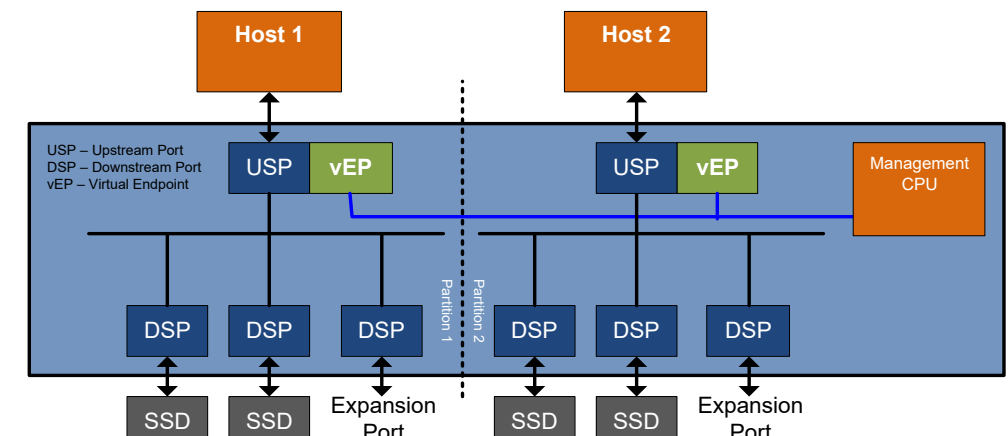
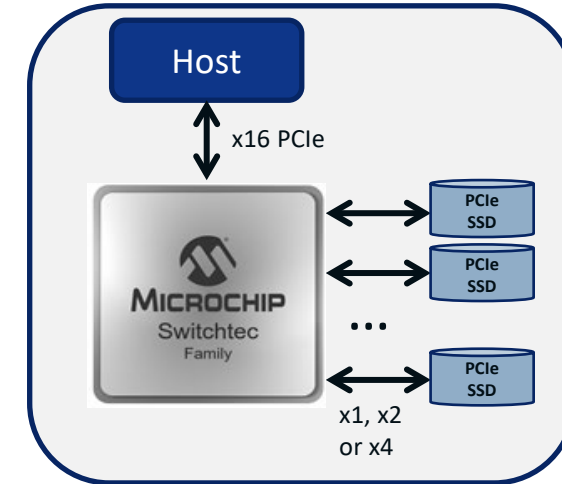
- NRZ 8b/10b encoding in 1.x and 2.x
- NRZ 128/130 encoding not reflected above in 3.x and 4.0
- PAM4 in Gen6

PCIe® Topology with Switch



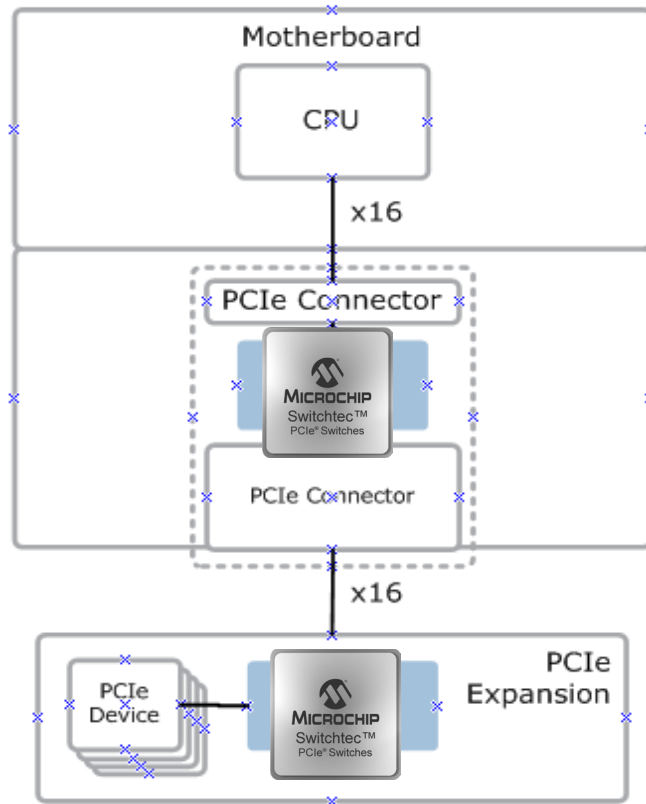
Upstream, Downstream Ports and Virtual Switch Partitions

- Host ports are typically x4, x8 or x16 and are connected to an Upstream Switch Port (USP)
- SSDs typically support x1, x2 or x4 ports and are connected to Downstream Switch Ports (DSP)
- Virtual Switch Partitions:
 - A single physical PCIe switch appears as multiple virtual PCIe switches. A virtual switch partition comprises one USP and one or more DSPs
- PCIe switches support multiple virtual switch partitions.

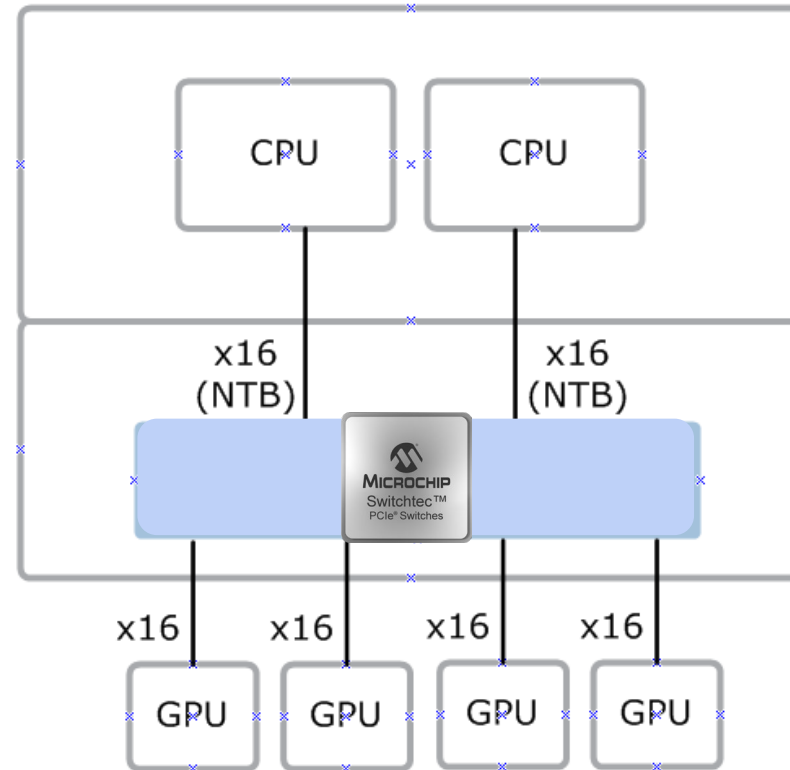


PCIe® Switch Application Examples

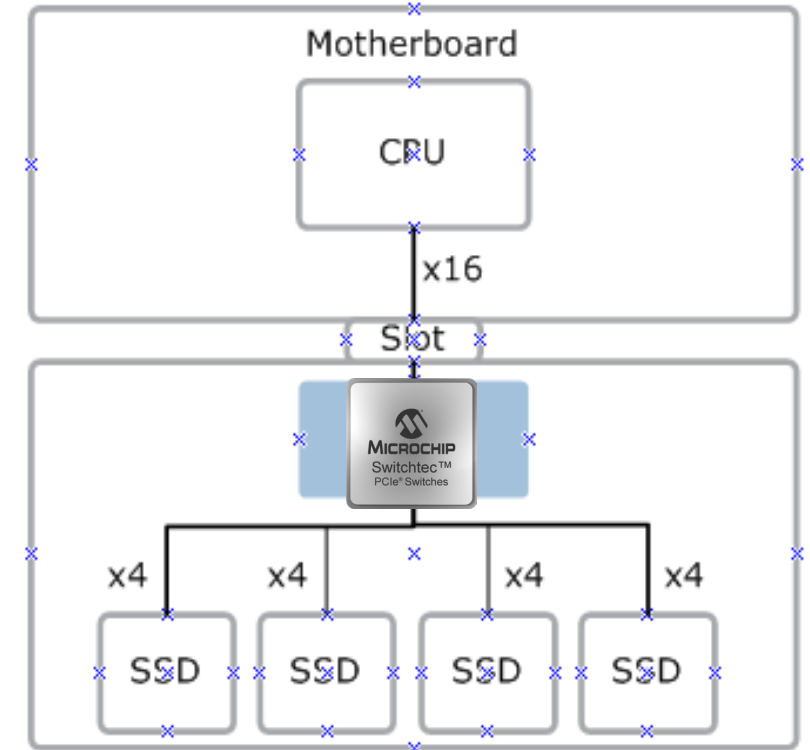
PCIe Expansion



Machine Learning / AI

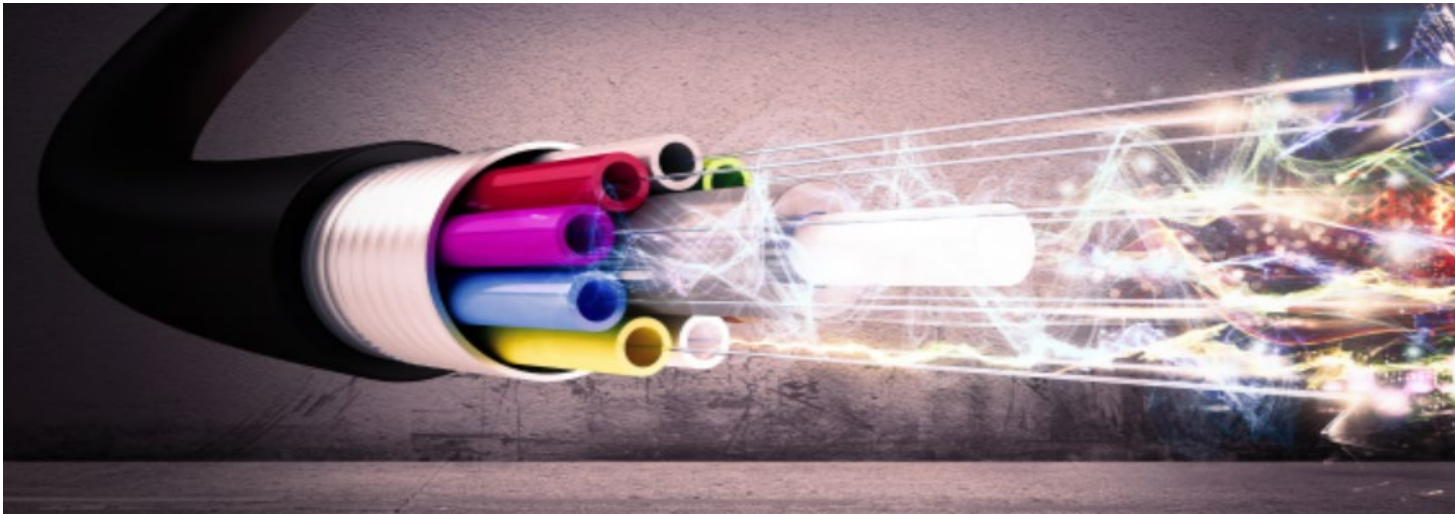


Server

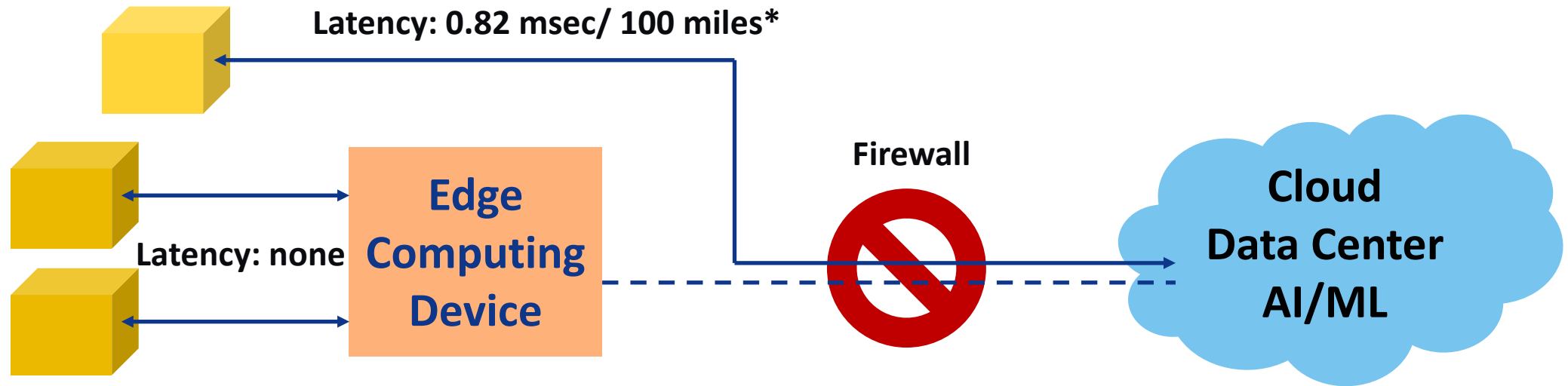


The Need for Speed: PCIe® 5

- Data analytics, autonomous-driving and medical diagnostics are driving extraordinary demands for machine learning and hyperscale compute infrastructure at a higher data rate
- Accelerators, graphic processing units (GPUs), central processing units (CPUs) and high-speed network adapters continue to drive the need for higher performance PCIe infrastructure
- Gen5 data rate at 32GT/s also benefits video processing and gaming
- CPUs and GPUs support PCIe 5.0 data rate



Latency in AI/ML and Hyperscale Computing



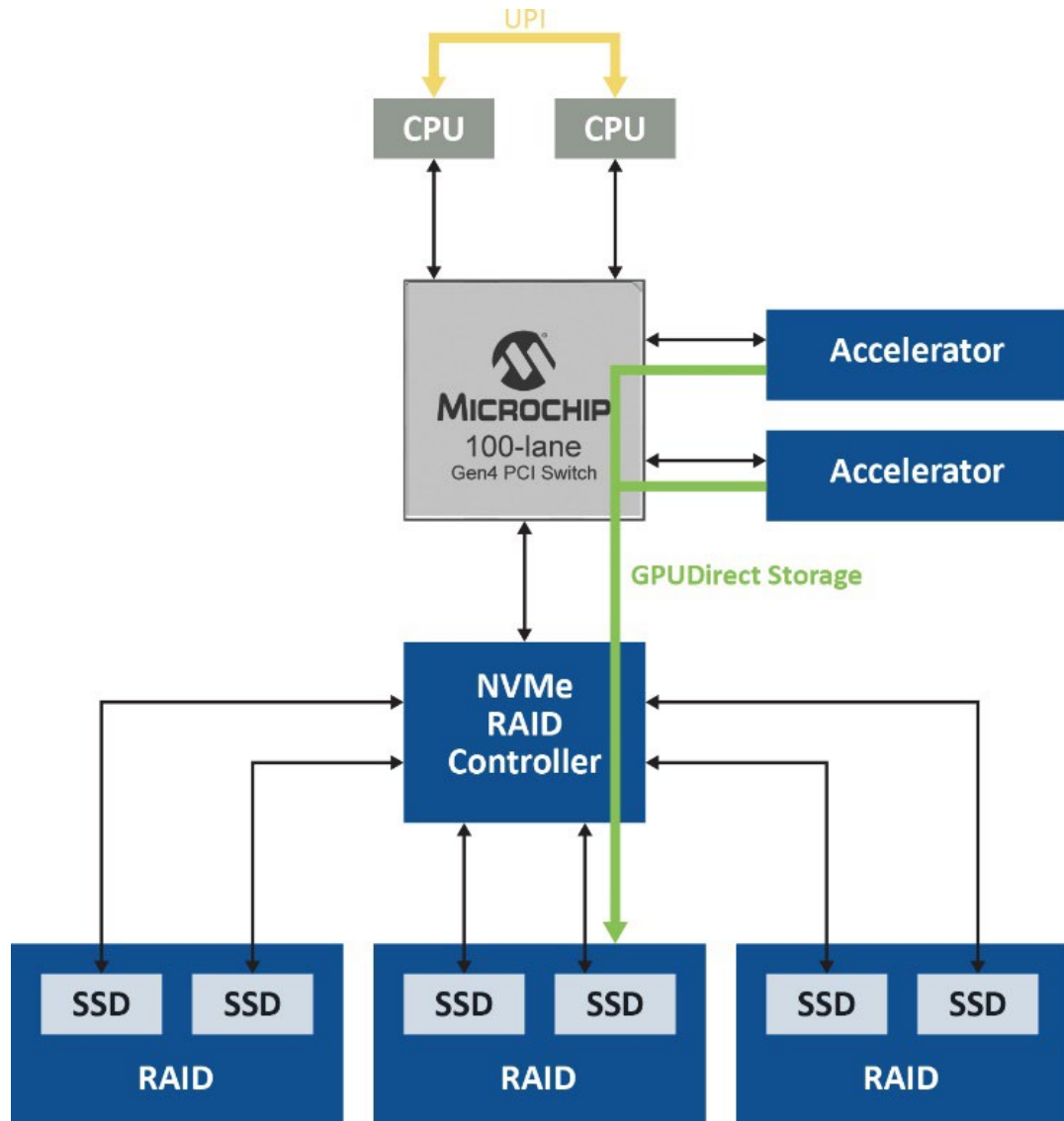
IoT Devices:

Smart Cities, Autonomous vehicles, manufacturing robots, etc..

- Latency in the data center plays a critical role in AI/ML processing
- Low latency is a must for all data processing devices

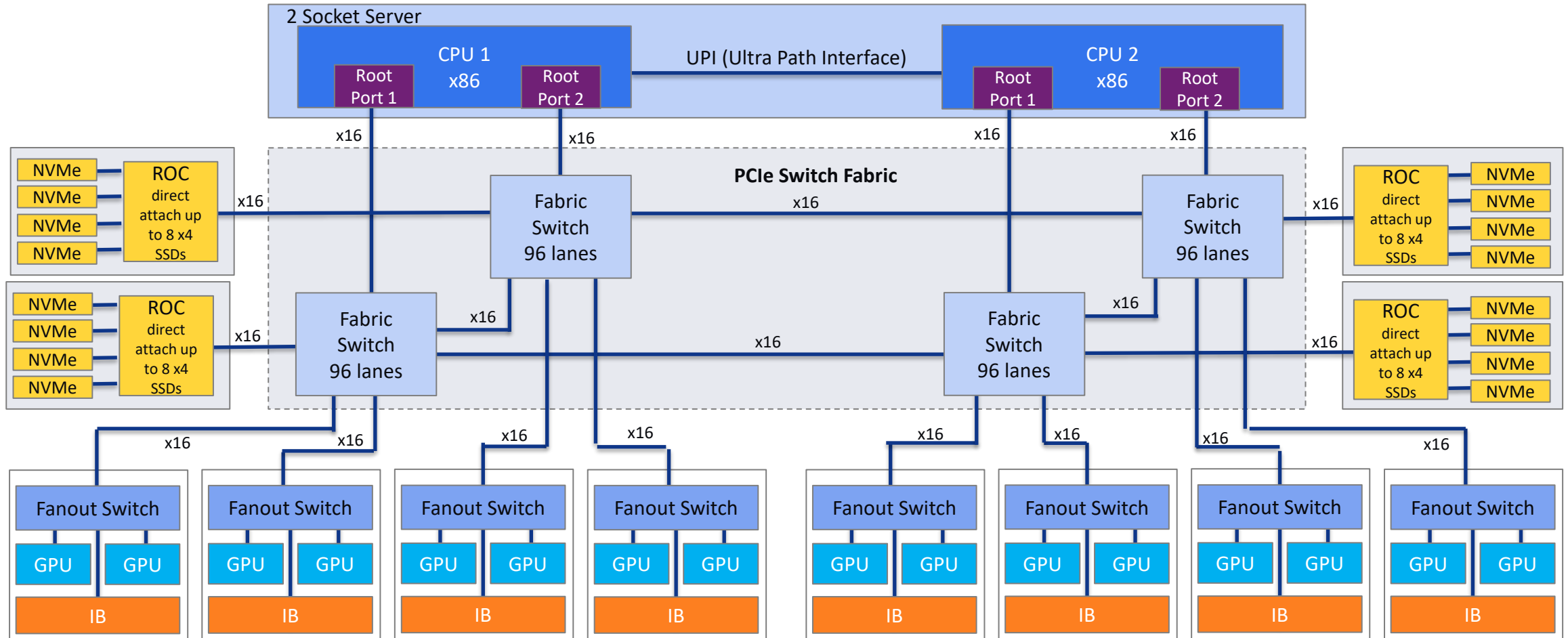
*<https://iotbusinessnews.com/download/white-papers/AVNET-ai-at-the-edge-whitepaper.pdf>

Latency Reduction: GPUDirect Storage



- GPUDirect Storage removes the latency bottleneck by not using system memory in the CPU to load data from storage into GPUs for processing
- A low latency PCIe switch facilitates the interface between GPUs and storage

Two Socket Server with 16 GPUs



- Switch provides peer to peer connectivity between all root ports, ROCs and GPUs
- NVMe used as a replay buffer (write once, read many times) minimizing UPI and network traffic
- NVMe direct (NVMe to GPU direct data transfer) enhancing performance by avoiding UPI
- Data set in NVMe protected by SmartRAID

Summary

- PCIe switching at Gen5 rate enables more bandwidth bridging for AI/ML processing
- A switch must be scalable with flexible port bifurcation from x1 to x32
- Low latency is a requirement
- End to end signal integrity to meet the PCI-SIG channel loss based on data rate
- More applications with CPUs, GPUs, and SoC interfacing for intelligent data processing the data center



Thank You
