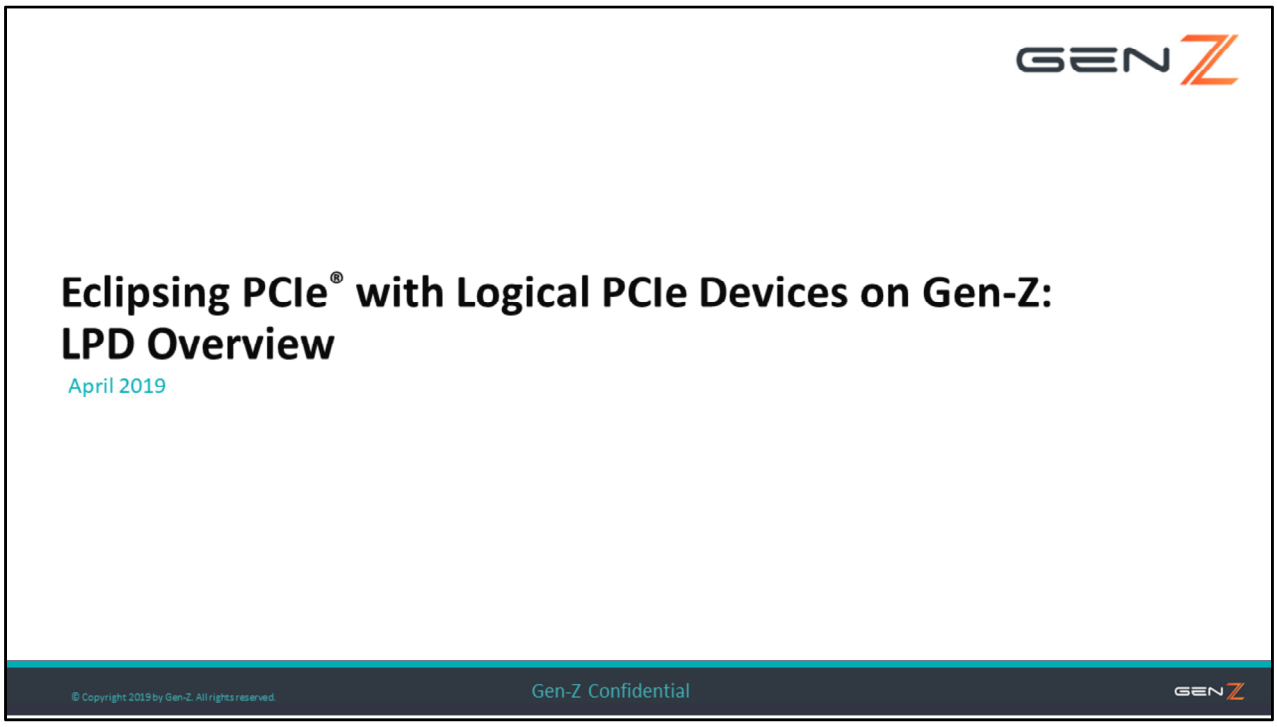


This presentation gives an overview of Logical PCIe Devices (LPDs). This optional LPD mechanism is a critical feature for Gen-Z that enables native Gen-Z I/O components to present themselves as one or more PCIe devices to a host system running an unmodified OS. Moreover, LPDs on Gen-Z go far beyond PCIe capabilities, creating immediate compelling I/O solutions with Gen-Z.

A related presentation on PCIe-Compatible Ordering (PCO) for LPDs covers in detail how PCO works.

Disclaimer

This document is provided 'as is' with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Gen-Z Consortium disclaims all liability for infringement of proprietary rights, relating to use of information in this document. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

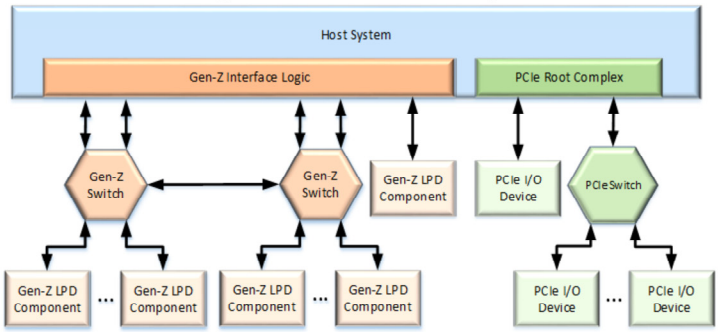
Gen-Z is a trademark or registered trademark of the Gen-Z Consortium.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

All material is subject to change at any time at the discretion of the Gen-Z Consortium

http://genzconsortium.org/

Gen-Z: a revolutionary new memory-semantic fabric



- Gen-Z and PCIe® may coexist indefinitely, with Gen-Z offering break-through solutions for:
  - Fabric-attached memory, Composable infrastructure, Device I/O
- Gen-Z uses **logical PCIe devices (LPDs)**, which can be supported by unmodified OSs
- Gen-Z components with LPDs capitalize on Gen-Z's revolutionary fabric while emulating PCIe Root Complex Integrated Endpoints (RCiEPs)
  - Multiple links, multiple paths, adaptive/dispersive routing, dynamic congestion management
  - Scaling increased by orders of magnitude and Gen-Z components readily shared among multiple hosts
- ***LPDs on Gen-Z go far beyond PCIe capabilities, creating immediate compelling I/O solutions with Gen-Z***

This drawing depicts an example Gen-Z system that contains both Gen-Z and PCIe® I/O components. Such I/O components might include HDDs, SSDs, HBAs, NICs, cluster interconnects, GPUs, computational accelerators, etc.

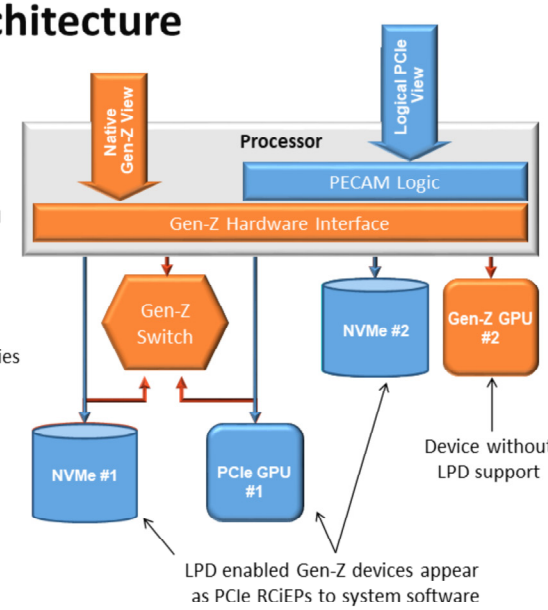
Note that Gen-Z supports advanced fabric topologies beyond simple tree-based ones, providing increased bandwidth, scalability, and availability. Bandwidth increases can come via higher link signaling rates, multiple aggregated links, or both. Scalability increases can come from larger fabrics with advanced topologies and congestion management. Availability increases can come from multipath link failover, end-to-end retry, and more precise error containment.

In 2001, the launch of PCIe was eased and accelerated dramatically by PCIe devices and switches presenting themselves to software as conventional PCIe devices and bridges, allowing the new PCIe ecosystem to be supported by unmodified OSs. Over time, OSs added support for advanced PCIe features.

Gen-Z is using a similar strategy with LPDs. LPDs enable immediate support of Gen-Z I/O components by unmodified OSs. Over time, OSs can add support for certain advanced Gen-Z features, though LPDs can immediately take advantage of many advanced Gen-Z features without waiting for OS changes.

LPDs can fully exploit Gen-Z Architecture

- Gen-Z devices can be discovered/configured
  - Via standard PCIe system software
- LPDs can fully exploit Gen-Z Architecture
  - Low-latency switching
    - Gen-Z 30 ns vs. PCIe 130-150 ns translates to 200-240 ns savings per read
  - Memory-speed CPU-to-device communication
  - Security and fine-grain hardware-enforced isolation (any-to-any communication without compromise)
  - Supports all x86 / ARM / Power architecture Atomics
  - Simplified single and multi-host I/O virtualization and sharing capabilities
  - Multipath—aggregation / resiliency / robust topologies
  - PCIe 2.5-32 GT/s PHY and 25-112 GT/s 802.3 / OIF Electrical
  - Legacy plus New Gen-Z Scalable Connector and Scalable Form Factors
  - CPU-based data movers to enable new software paradigms
  - Scale-up and scale-out connectivity and performance
  - Simplified software—any mix coherent and non-coherent operations
  - And much more…



Though LPDs emulate PCIe Root Complex Integrated Endpoints (RCiEPs), Gen-Z components with LPDs can fully exploit Gen-Z's revolutionary fabric and architecture.

Much of this is facilitated by Gen-Z-aware system firmware discovering and configuring local Gen-Z components before the OS boots. Some of this is enabled by Gen-Z "PECAM" logic (covered on a subsequent slide) presenting LPDs as being directly connected to the host, and not making the Gen-Z fabric visible to an unmodified OS.

Many performance benefits come from Gen-Z links supporting higher signaling rates, Gen-Z switches inherently providing lower latency through their architecture, and Gen-Z fabrics supporting advanced topologies with link aggregation and dynamic congestion management.

Future performance benefits will come from applications and OS infrastructure becoming Gen-Z aware and utilizing new software paradigms or hardware primitives supported by Gen-Z. Examples include load/store access to storage and use of processor-specific atomic operations.

Innovative LPD mechanisms deliver value while maintaining compatibility

PECAM

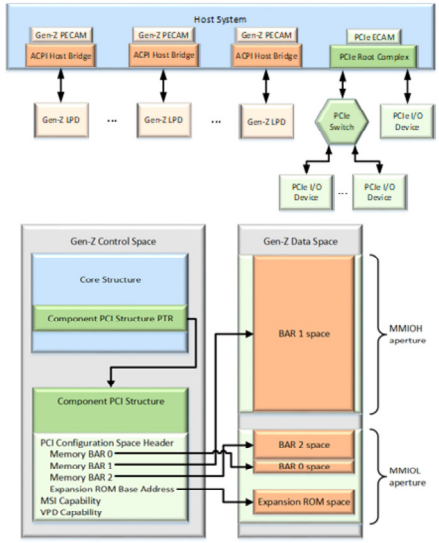
- Compatible with PCIe's standard ECAM mechanism, supports LPD discovery/enumeration/configuration by mapping emulated PCIe Configuration Space for each LPD function
- Presents LPDs as being directly attached to the host, regardless of their actual Gen-Z fabric connections
- Enables massive per-system LPD scaling via the use of virtual bus/device/function routing tuples and virtual PCI Segment Groups

MMIO apertures

- Enables (unmodified) OS software to remap BARs without breaking the MMIO mappings established by System Firmware

PCIe Compatible Ordering (PCO)

- Native Gen-Z ordering is optimized to support advanced fabric features like multipath and congestion avoidance, but LPDs using native Gen-Z ordering may require changes to PCIe driver and infrastructure software
- With PCO, LPDs can support PCIe-compatible ordering with minimal Gen-Z HW burden & no changes to ported PCIe software for ordering



Gen-Z's PECAM mechanism is compatible with PCIe's standard ECAM mechanism, which supports device/function discovery, enumeration, and configuration. Mentioned earlier, a PECAM presents LPDs as being directly attached to the host, regardless of their actual Gen-Z fabric connections. This avoids an unmodified OS from attempting to manage a Gen-Z fabric, which the unmodified OS doesn't comprehend.

MMIO apertures are a clever use of existing ACPI mechanisms to constrain MMIO (memory-mapped I/O) mappings within specified ranges ("apertures"). This enables Gen-Z hardware to tolerate if (unmodified) OSs reprogram LPD function Base Address registers (BARs) from the values configured earlier by system firmware.

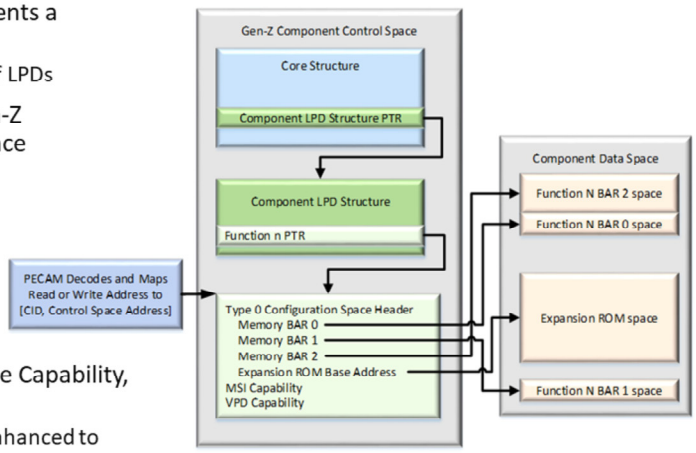
PCIe and Gen-Z have fundamentally different ordering models. Notably, PCIe requires its fabric to do ordering enforcement while Gen-Z does not, which enables Gen-Z fabrics to perform substantially better, taking advantage of multipath, congestion avoidance, and reduced head-of-line blocking. PCO is an optional feature for LPDs that enables PCIe driver and infrastructure software to work without changes to accommodate Gen-Z's different ordering model. PCO basically works by configuring the Gen-Z fabric to deliver all PCO packets between an LPD and its host in order. Due to fundamental fabric ordering differences, PCO uses totally different and innovative approaches to guarantee forward progress when avoiding deadlock.

How do LPDs work?

- A Gen-Z component that supports LPDs implements a Component LPD structure for each LPD.
  - One Gen-Z component can support thousands of LPDs
- Each LPD function has a 4096-byte region in Gen-Z Control Space that looks like PCI/PCIe Config Space

Logical PCIe Configuration Space content:

- Type 0 Configuration Space Header
  - BARs that map the LPD function's run-time CSRs
  - Base Address for optional Expansion ROM
- Interrupt controls (in MSI or MSI-X capability)
- Optional PCI/PCIe Capability structures; e.g., PCIe Capability, SR-IOV, ATS, etc
  - Page Request Group (PRG) services for LPDs is enhanced to provide better paging controls



A Gen-Z component that supports LPDs implements a Component LPD structure for each Logical PCIe Device. Like PCIe, each Gen-Z Logical PCIe Device may support up to 256 functions. A Gen-Z I/O component can support many thousands of LPD functions, supporting massive scaling even within a single component!

Each function contains a 4096-byte region that looks like PCI/PCIe Configuration Space to the host system. Using its PECAM, the host reads & writes the function's Configuration Space to discover its attributes, bind a suitable driver to it, and configure its essential resources like MMIO space, interrupts, and advanced PCIe capabilities.

System firmware uses the Gen-Z Requester ZMMU to map PCIe Configuration Space for the PECAM. Similarly, system firmware uses the Gen-Z Requester ZMMU to map PCI Memory Space for the function's run-time CSRs.

Gen-Z transactions can carry LPD-specific information

- ***Enables LPDs to support advanced PCIe functionality***
- Some transactions used by LPDs require information unique to LPDs, for example:
  - TA field: Translated Address; used by PCIe Address Translation Services (ATS)
  - LPD BDF – bus/device/function associated with the LPD function; used by ATS, IOMMU, and interrupts
  - PASID: Process Address Space ID; used by advanced IOMMUs for VMs
  - PH: Processing Hints; used by PCIe TLP Processing Hints (TPH)
  - Steering Tag: used by TPH
- Here are Gen-Z transactions used by LPDs and optionally carrying LPD-specific info:
  - Read/Write – used where PCI/PCIe devices would use Memory Read/Write transactions
  - LPD interrupts – used where PCI/PCIe devices would use Memory Write transactions for MSI/MSI-X
  - LPD ATS transactions – used where PCIe devices would use corresponding PCIe ATS transactions
  - Atomics – used where PCIe devices would use corresponding PCIe AtomicOp transactions
    - Note: LPDs can use a much broader set of atomics than those supported by PCIe
- These are highly efficient, native Gen-Z transactions
- For increased efficiency, LPDs may use the same transactions without LPD-specific info when not needed

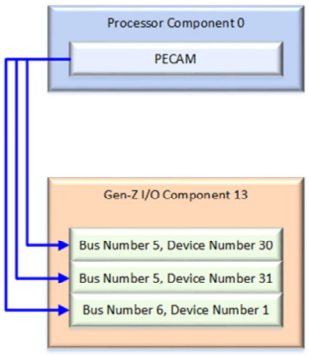
Some functionality associated with advanced PCIe capabilities requires information not carried by normal Gen-Z transactions. LPDs supporting these advanced PCIe capabilities use special versions of Gen-Z transactions that carry the required additional information. For example, LPDs that support PCIe Address Translation Services (ATS) must use Gen-Z transactions that carry the TA field. The slide gives other key examples of additional information and how it is used.

Gen-Z architects a much broader superset of the atomics supported by PCIe, and LPDs can use any atomics from that superset.

Gen-Z transactions carrying this additional information are still highly efficient, native Gen-Z transactions. For cases where the LPD isn't using functionality that requires the additional information, the LPD can use Gen-Z transactions without it, for even greater efficiency.

LPDs support massive scaling

- ***One Gen-Z I/O component can support up to thousands of LPDs***
  - Each LPD can be single-function or multifunction device
  - Each LPD can support up to 256 RCiEP functions
- Bus/device/function tuples (BDFs) mapped by PECAMs are virtual, and do not correspond to physical busses or devices
- For each host, multiple bus/device number tuples can be mapped to the same Gen-Z component as shown
- PECAMs present no bridges or switches, so these elements consume no virtual BDFs
- There's no need for system firmware or OSs to allocate multiple bus numbers per physical slot to support add-in cards with embedded PCIe switches or SR-IOV multi-bus capability
- There's no need for OSs to rebalance bus numbers during run time



LPDs support massive scaling!

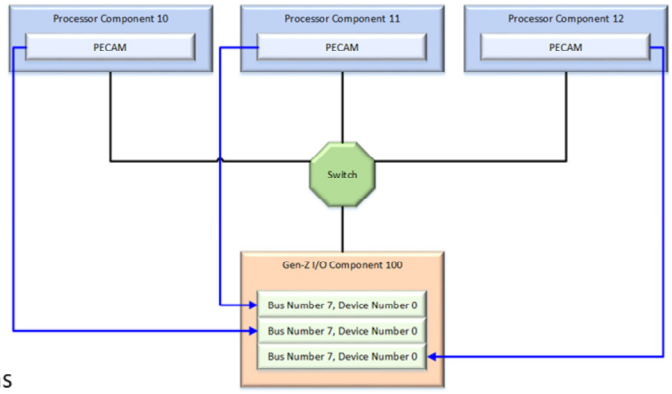
Bus/device/function tuples (BDFs) mapped by PECAMs are virtual, allowing a single Gen-Z component to support an arbitrary number of virtual devices, consuming an arbitrary number of virtual busses. This enables a single Gen-Z component to support up to thousands of LPDs, and each LPD can contain up to 256 functions.

Bus and device numbers assigned for LPDs within a Gen-Z component don't need to be consecutive or even monotonically increasing. This avoids the need for OSs to ever "rebalance" bus numbers during run time, which many OSs do not support.

PECAMs present no bridges or switches, so bridges/switches consume no virtual bus numbers.

I/O Component Sharing

- With suitable Gen-Z fabric management SW, a single Gen-Z I/O component can be shared among multiple Gen-Z host systems
- Component sharing offers significant availability benefits, e.g., system failover for continued operation and avoidance of stranded resources
- Each Gen-Z host system running an unmodified OS only "sees" LPDs assigned to it by Gen-Z fabric management SW
- Each Gen-Z host system's BDF space for LPDs is orthogonal to the BDF space in other host systems



- A Gen-Z I/O component designed to be shared by multiple host systems must ensure that the virtual device instances do not interfere with each other
- Gen-Z can deliver the major benefits of PCIe MR-IOV* with less cost & complexity

*MR-IOV: Multi-Root I/O Virtualization and Sharing

Gen-Z fabrics can readily connect multiple systems, and LPDs are able to be shared by multiple host systems, given suitable Gen-Z fabric management software.

Component sharing provides significant benefits for availability. For example, if multiple systems are providing a service and one of the systems fails, the remaining systems can take over the failed system's I/O components, allowing continued operation of the service.

OSs do not need to be aware that they are sharing a common Gen-Z I/O component. Each system running an unmodified OS will only "see" the LPDs assigned to it by Gen-Z fabric management software and mapped by its PECAM, so there's no problem with unmodified OSs attempting to claim the same logical devices.

I/O component sharing does not mean a loss of security. Host systems and LPDs can be configured to use R-Keys, ensuring that host systems can access only the LPDs that they own within a shared I/O component, and LPDs can access only the host systems that own and control them.

As shown in the figure, the BDF space for each host system is orthogonal to the BDF spaces in other host systems. A single component can have the same bus/device number tuple assigned to multiple LPDs, as long as each LPD belongs to a different host system.

Gen-Z Logical PCIe Devices Summary

- **Gen-Z systems may contain a mix of Gen-Z I/O components and PCIe I/O devices**
  => Enables a graceful transition period, with timing driven by industry needs
- **LPDs enable Gen-Z I/O components to work immediately with unmodified OSs**
  => Avoids waiting months or years for OS I/O infrastructure to comprehend Gen-Z
- **LPDs are native Gen-Z components that deliver key Gen-Z benefits**
  => Provides immediate boosts in bandwidth, scalability, and availability
- **LPDs may optionally support PCIe-Compatible Ordering (PCO)**
  => Avoids needing to modify ported PCIe software to accommodate Gen-Z's ordering model
- **Virtual mappings for LPDs enable massive scaling**
  => Avoids the number of I/O components used by a single system being limited by the I/O fabric
- **Gen-Z can support I/O component sharing with less cost & complexity**
  => Makes sharing a single I/O component between multiple hosts more viable

Here are the key takeaways for LPDs, each with their associated benefit.

GEN Z

Thank you