Integrating the Gen-Z Interface in Your SoCs

Bastien Heneffe
Scope and Agenda

- **Gen-Z basics**
  - What’s different from other protocols in terms of functionality and integration

- **Gen-Z IP Core integration**
  - High-level view of a Gen-Z IP Core: subblocks and interfaces
  - Configuration challenges

- **Wrap-up**
Gen-Z basics

The motivations behind Gen-Z:

- Current IT infrastructure reaching its limits
- Need a memory semantic fabric to achieve efficient access
- Need a solution for Memory/Storage convergence
- Need a universal fabric for memory composability

Gen-Z consortium joined effort to specify:

- Gen-Z Core specification
- Gen-Z Physical layer specification
- Gen-Z Connectors specification
- Gen-Z Form Factor specification

Source: “The Future of Extreme Scale Computing” - HPE
Gen-Z architecture attributes

- **Memory semantic**
  MMU vs. IOMMU

- **Long-haul Fabric support & PHY agnostic**
  *Physical Layer Abstraction* interface between Core and PHY
  Allows IEEE 802.3 PHY (short & long haul) and PCIe PHY

- Compatible with **meshed topologies**
  Multi-link and multi-path support, subnet architecture

- **Scalable to many-components networks**
  Up to 4096 components per subnet, up to 65536 subnets

- **Advanced operations**
  Optional *OpClasses* for specific operations

- **RASM: Reliability, Availability, Serviceability & Manageability**
  Load balancing, automatic failover, security features, ...
Gen-Z PHY and connectors

- Gen-Z Physical Layer Specification v1.1 (latest version under member review) covers:
  - RZ PCIe up to Gen5
  - NRZ IEEE 25G (Local & Fabric)
  - PAM4 IEEE 50G (Local & Fabric)

- Gen-Z core is PHY agnostic:
  - Ready for higher rate (including 112G)

- Gen-Z also defines:
  - Mechanical Form Factors
  - Connectors
Gen-Z System topologies

LOCAL SYSTEM
- Copper, low-cost
- PCIe or IEEE 802.3 PHYs
- P2P, Daisy-chain, switched

CHASSIS / ENCLOSURE
- Copper, low-cost
- PCIe or IEEE 802.3 PHYs
- P2P, Mesh, Torus, switched

RACK-SCALE
- Copper or optical
- IEEE 802.3 PHYs
- Switched, Torus, Spine/Leaf

ROW-SCALE
- Optical
- IEEE 802.3 PHYs
- Switched, Fat Tree, Clos, Butterfly, Hyper-X, etc.

Source: “The Future of Extreme Scale Computing” - HPE
Flexibility is key for an interface controller.

Must haves:

- Scalable and efficient interfaces
  - Achieve required performance with optimal footprint

- Configurable features
  - Optimize resource usage and constrain footprint

- Equally support ASIC and FPGA
  - For test, emulation and prototyping
Gen-Z IP Core Interfaces

**PLA - Physical Layer Abstraction** *(from Gen-Z specification)*
- PHY-agnostic interface, with configurable datapath
- Flexible low power I/F to support various PHYs
- Single or multiple interfaces

**User I/F**
- Single or multiple Master and/or slave interfaces
- High bandwidth and many outstanding requests
- Application interface: typically AMBA AXI or similar

**Configuration interface**
- Access to control space (Gen-Z structures)
- Low bandwidth, only required for out-of-band support
Topology configuration: #1

**Link I/F**
- Implement data link features
- PLA interface towards the PHY

**Responder** (Gen-Z -> user I/F)
- Convert Gen-Z transaction into User transactions

**Component Manager**
- Control space and configuration management

**Use cases**
- Media only needing slave interface (e.g. memory)
- Small footprint
Topology configuration: #2

Requester (user I/F -> Gen-Z)
- convert User transaction into Gen-Z transactions

Interconnect
- Transport packets between sub-blocks

Use cases
- Media needing a master access to Gen-Z (e.g. accelerators)
- Host systems
Multiple Link I/F
- Increase fabric bandwidth
- Several PHY modules
- Multi-path

Switch
- Packet Routing capabilities

Multiple requesters/responders
- Increase user-side bandwidth
- QoS, Multiple VC, ...

Use cases
- Switch & fabric SoCs, hosts, ...
Configuration: Which type of packets?

Type of Packets (a.k.a OpClasses and OpCodes)
- Load/store packets:
  - Core64: Versatile header => all purpose transaction (with IDs, access key, properties,...)
  - P2P64: Compact header => point-to-point transaction only
- Control packets:
  - Access to control space from fabric (In-band management)
- Optional packets:
  - Many types: Multicast, Atomic, Large Data Move, Cache coherency, etc.

Recommendations:
- Media and Accelerators support -> packet that correspond to application
  - Load/store: Core64 and/or P2P64.
  - Control: for In-Band Management.
  - Optional packets: only if needed.
- SoC & Hosts support -> maximum compatibility needed
  - Load/store: Core64 and P2P64
  - Control: for In-Band Management.
  - Optional packets: as many as possible for maximum compatibility => trade-off to find
Configuration: some extra features

Additional Gen-Z features: to support or not?

- In-band & Out-of-band management
  - In-band: useful in all cases for remote configuration
  - Out-of-band: useful for Host/CPU use cases and debug
- Routing capabilities
  - For multi-link cases (e.g. switches)
  - Adaptive routing / route failover, etc.
- Virtual Channels and Traffic Classes
  - When Reliability and Quality of Service are needed
- Add additional Gen-Z Control Structure
  - E.g. media management, component statistics, vendor-defined, etc.

IP configuration

- Size parameters to fit to required performances/footprint
  - Datapath width and buffer sizes have a big impact on area
Wrap-up

- Applications can benefit from Gen-Z
  - Memory semantic support
  - Compatible with long-haul reach meshed networks
  - Multi-link and multi-host support
  - Building blocks are available to integrate Gen-Z in SoCs/FPGAs:
    - Scalable IP Cores
    - PoC designs/boards, VIPs, test equipment and PHYs
    - Growing ecosystem
  - Gen-Z is low-risk
    - Reuse of proven PHY technology
    - Possible to share PHY with other controller
Thank you!

Any questions?