Gen-Z Memory Access

April 2019
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/
In this architecture, processor memory management and media-specific logic are interlocked within a single, monolithic memory controller. The memory controller is responsible for ensuring semantic ordering, selecting a memory channel, sequencing each memory access phase, power / error management, data integrity, etc.
• Gen-Z can be integrated into a processor without impacting the traditional memory controller. For example, a DDR memory controller would continue independently service a portion of the processor’s address space, and Gen-Z would independently service a different portion. Depending upon the design, such a processor could deliver double or more than application memory bandwidth. For example, a processor with 8 DDR 5 6400 channels and 64 Tx / Rx 32 GT/s lanes Gen-Z could deliver ~800 GB/s of application bandwidth (~400 GB/s DDR plus ~400 GB/s Gen-Z).
  • If 56 GT/s signaling is used, then DDR + Gen-Z could deliver ~1.1 TB/s of application bandwidth
  • If 112 GT/s signaling is used, then DDR + Gen-Z could deliver ~1.8 TB/s of application bandwidth
  • If 112 GT/s signaling and 128 Tx / Rx lanes are used, then DDR + Gen-Z could deliver ~3.2 TB/s of application bandwidth
• Gen-Z logic is directly attached to the processor’s internal coherency interconnect for optimal latency and bandwidth.
As illustrated, a portion of the processor’s address range mapped to Gen-Z. In direct-attached point-to-point topologies, the processor physical address is directly mapped to an individual component. Alternatively, in point-to-point and switch-based topologies, the processor physical address is translated by Gen-Z memory management unit (ZMMU) to a component identifier and a component-relative address.

Memory components can be reached through multiple component interfaces which provides the following benefits:

- Eliminates stranded resources and single points of failure
- Improves aggregate bandwidth—multiple links, memory interleave, etc.
- Enables higher physical layer signaling rates
- Enables multiple components, not just processors, to directly access memory components thereby reducing data movement, improving latency, enabling new solution architectures, etc.
Media Controller Basics

- Translates protocol request packets into media-specific operations.
- Media and component-specific data-integrity, resiliency (e.g., media-specific error recovery and memory device or device row sparing), availability, and, if applicable, wear-leveling services.
- Gen-Z and media controller-specific management services, e.g., power management including refresh for volatile media (i.e., self-refresh), error, statistics, sensors, etc.
- Media-specific internal communications
- Aggregation and interleaving media-specific operations across one or more media devices.
Media Controller Capabilities

- In general, a media controller is co-located with the media devices and has intimate knowledge of their unique properties and organization, the media controller can:
  - Optimize / tighten media timing budgets to minimize / eliminate wait cycles
  - Optimize power management to reduce or smooth power consumption
  - Provide real-time thermal management
  - Provide automatic wear leveling for non-volatile media
  - Provide automatic or software-controlled garbage collection services for non-volatile media
  - Provide data migration between multiple tiers of media or diverse types of media
  - Provide data integrity and data encryption services
  - Provide mailbox communication services to enable memory management
  - Support Meta read and write semantics to support value-add functionality, application-level data integrity services, customized caching services, etc.
  - Provide dynamic post package repair, e.g., automatic row remapping upon detecting defects or too many correctable / uncorrectable errors during power-on self-test (POST) or during run-time operation
Media Controller Value-Add Capabilities

- Media controller can incorporate a cache resource to enable the following:
  - Break the interlock between outstanding requests and the number of physical banks
    - Cache can hold 100s of media rows to improve parallelism, aggregate bandwidth, and subsequent access latency
  - Media controller can re-arrange media operations to improve latency, e.g., if using DRAM media:
    - Issue multiple consecutive Activates without interlocking each Activate with a Precharge
    - Issue multiple consecutive Precharges, i.e., use lazy Precharge in the background
  - Media controller can use a portion of the cache to perform volatile media refresh, patrol scrubbing, etc. without moving data to / from the host memory controller
  - Media controller can apply solution-driven heuristics or other policies to perform prefetch to reduce subsequent load-to-use access latency
  - Media controller can consider spatial locality to optimize access in place of an all-or-nothing open or closed page policy
  - Media controller can reduce the size of the physical row to reduce data movement and power consumption
- Media controller can support data-centric acceleration
  - Discrete or integrated acceleration logic or embedded processing
  - Improves performance and power consumption by reducing data movement and offloading computation and data manipulation
Thank you