HotStorage'24

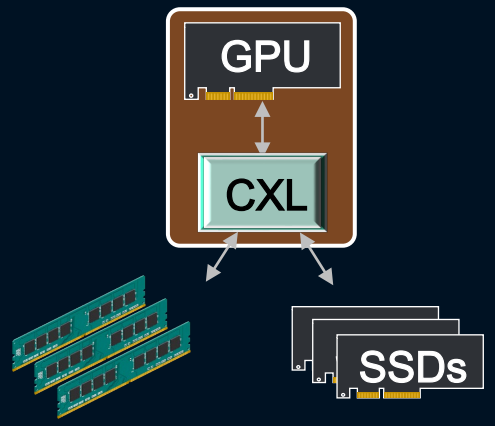
Breaking Barriers: Expanding GPU Memory with Sub-Two Digit Nanosecond Latency CXL Controller

Donghyun Gouk, **Seungkwan Kang*** , Hanyeoreum Bae, Eojin Ryu,
Sangwon Lee, Dongpyung Kim, Junhyeok Jang, Myoungsoo Jung

KAIST CAMEL panmnesia

High-Level Summary

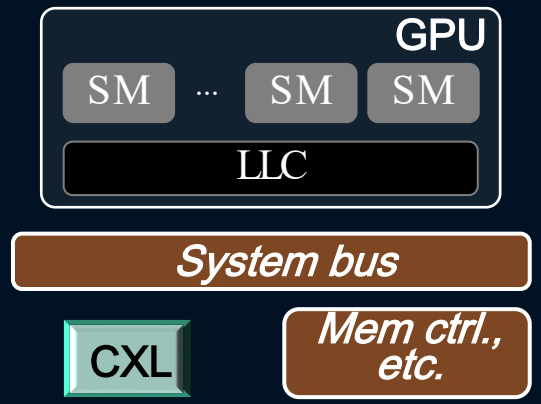
We introduce the potential of GPU storage expansion utilizing CXL



Direct access
- Load/store access to EPs

Async. access
- Diverse backend media

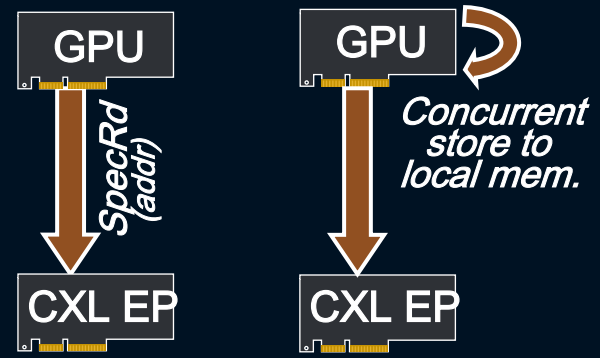
We designed and prototyped a CXL-integrated GPU



CXL-integ . GPU
- CXL root complex
- GPU architecture

Memory space
- Memory map
- Initialization

We further minimized the impact of backend media latency



Speculative read
- Minimize read latency

Determinstic store
- Mitigate tail latency of writes

CAMEL

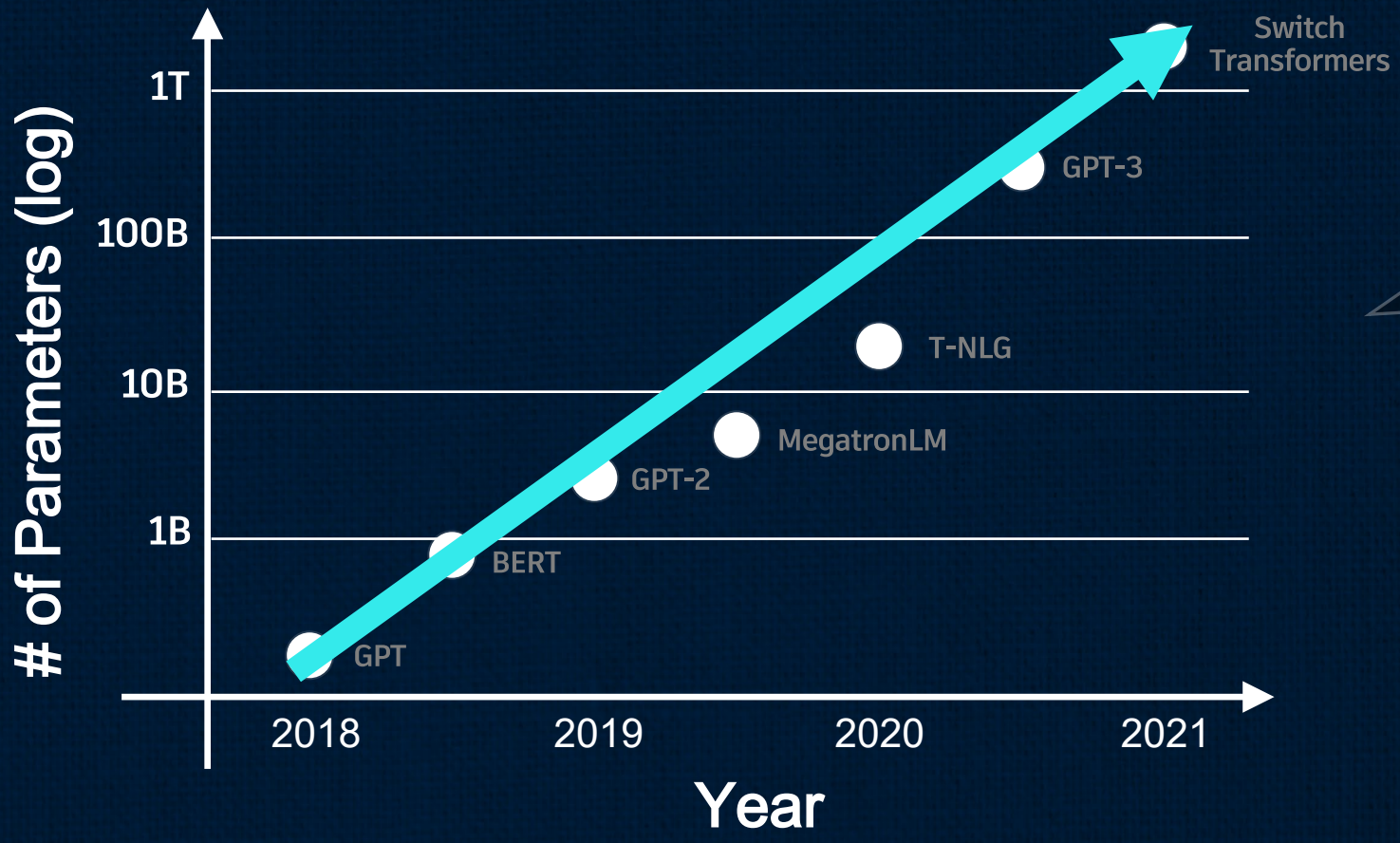
1. GPU Memory Expansion and Potential of CXL

2. Designing a CXL-integrated GPU

3. Mitigating Backend Media Latency

4. Evaluation Results

Growth of GPU Memory Requirements



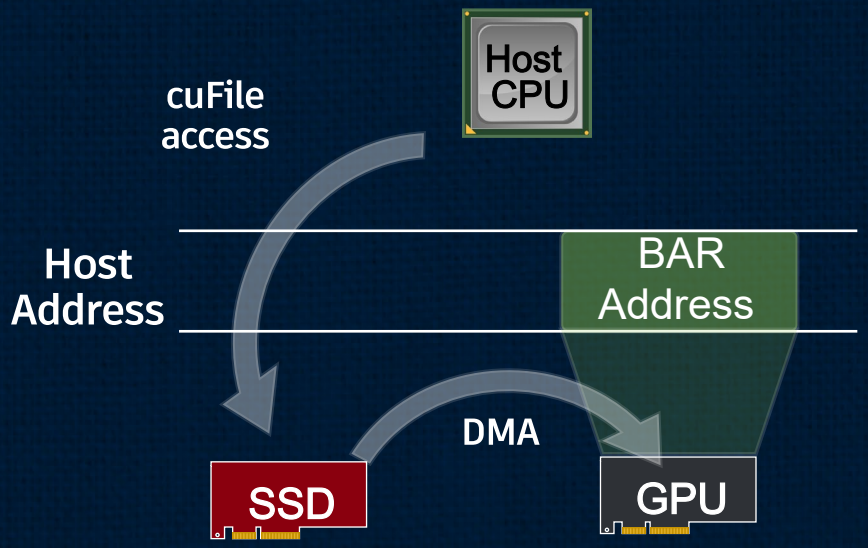
Growth of 10x every year

~1B params -> ~20GB memory
~100B params -> ~TB memory

GPU Memory Expansion

Storage Solutions

(e.g., GPU-Direct Storage)

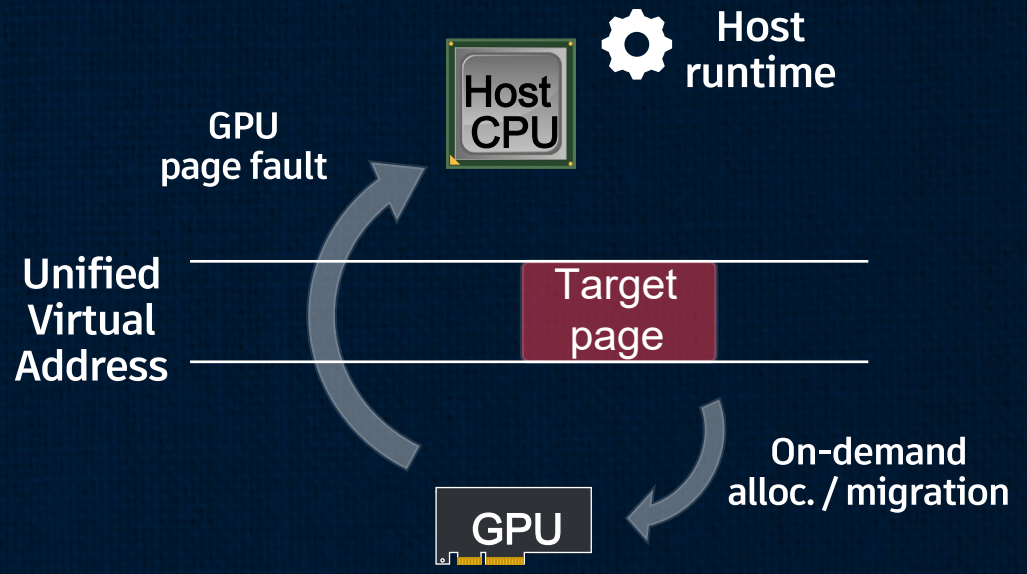


- ✓ Enables large-scale models via storage
- ✓ Direct access to storage

- x Complexity of usage (Manual handling)
- x Disparity in I/O granularity

Host Memory Solutions

(e.g., Unified Virtual Memory)



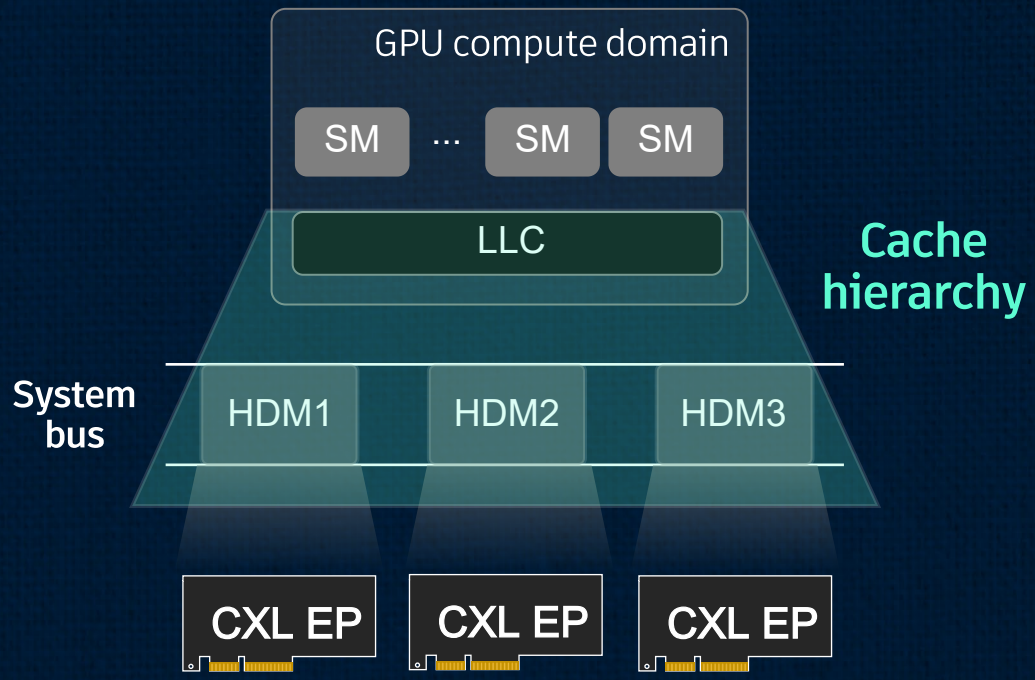
- ✓ On-demand allocation/migration
- ✓ Wide adoption (e.g., Tensorflow, DGL)

- x Requires host runtime
- x Induce performance bottlenecks

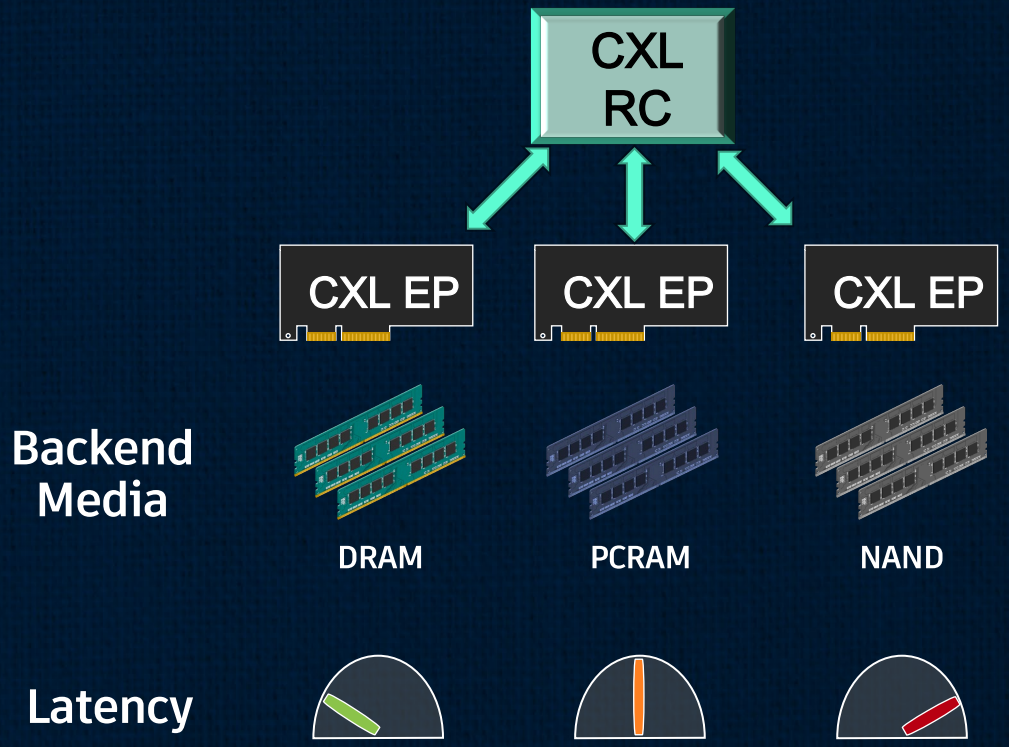
CAMEL

Potential of CXL -integrated GPUs

Direct access to EPs via ld/st memory access



Asynchronous communication w/ media



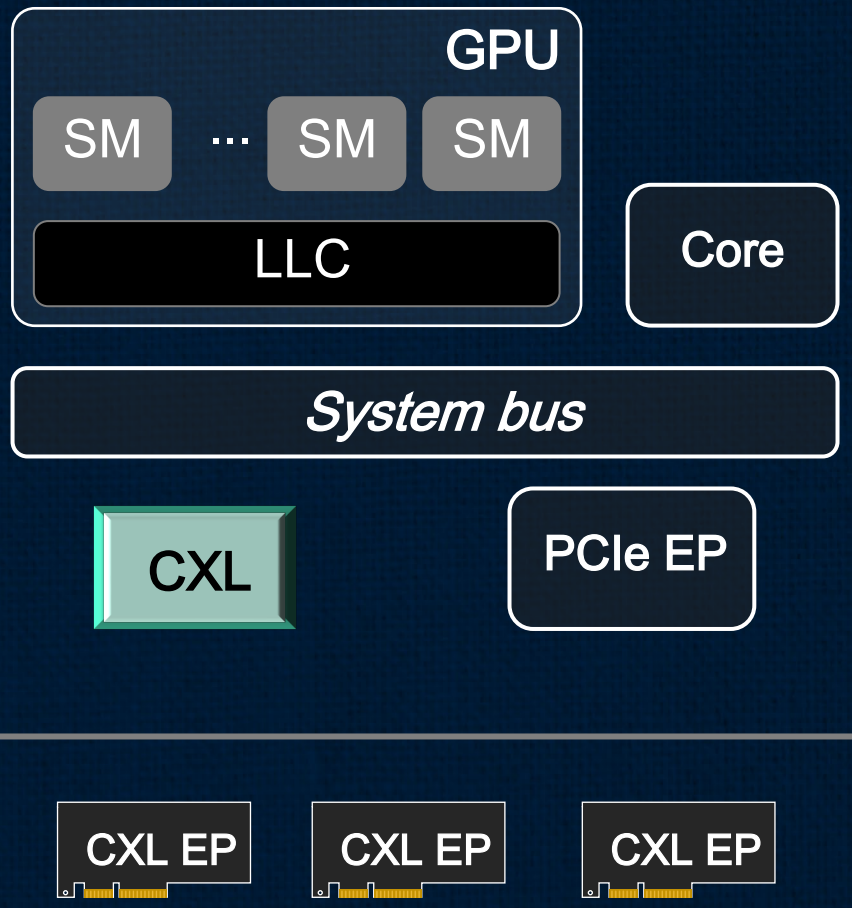
1. GPU Memory Expansion and Potential of CXL

2. Designing a CXL-integrated GPU

3. Mitigating Backend Media Latency

4. Evaluation Results

Challenges



CXL root complex

How should we design a CXL controller for memory expansion?

CXL-integrated GPU architecture

How should we connect the CXL controller to the GPU?

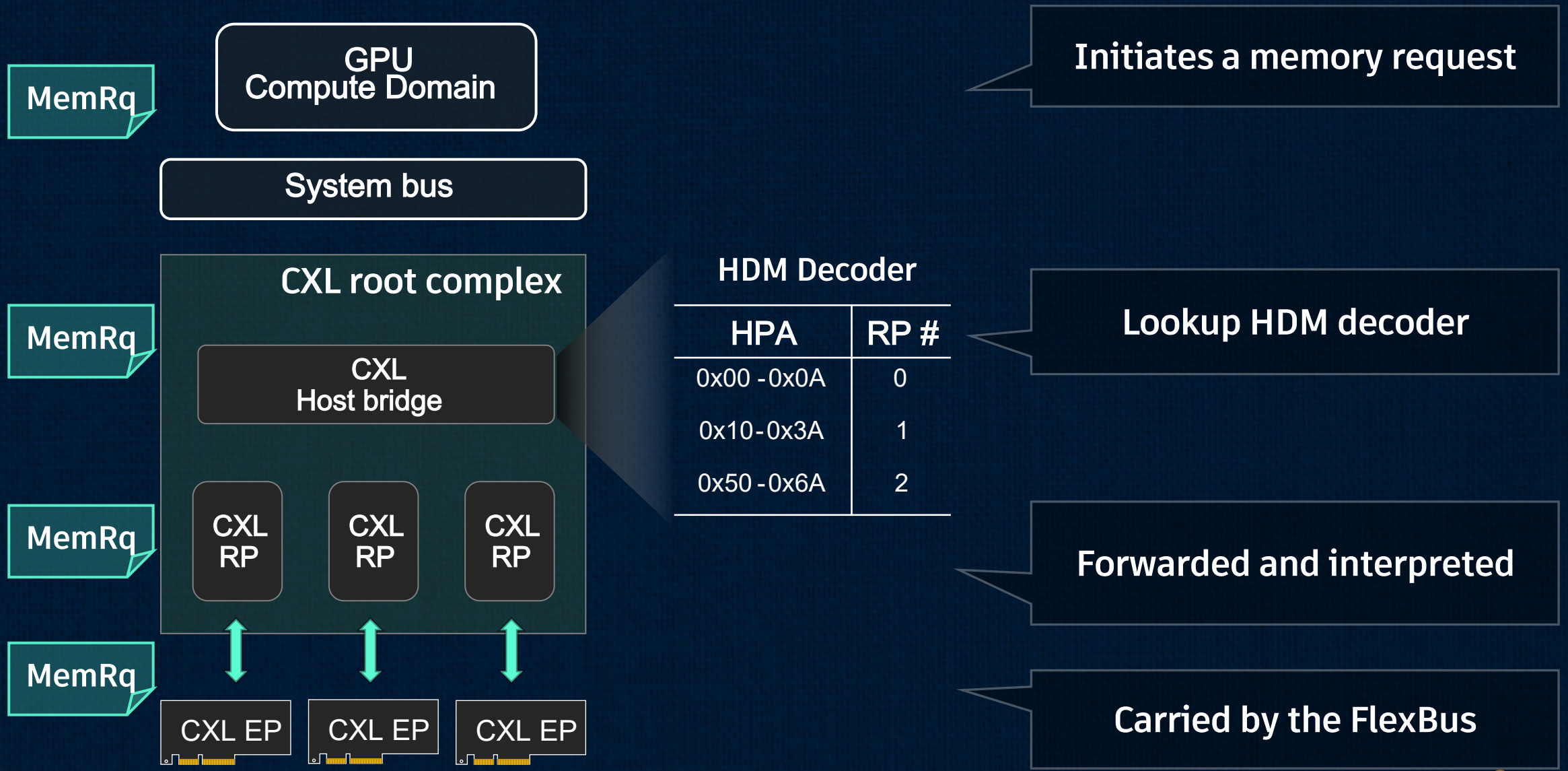
System initialization

System bus memory map for routing

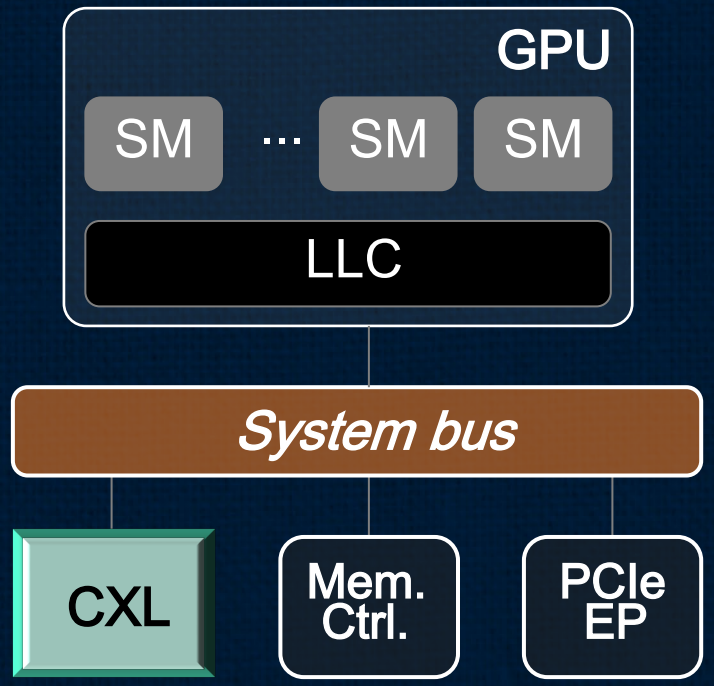
Diverse memory backends

CAMEL

CXL Root Complex - E2E Data Movement



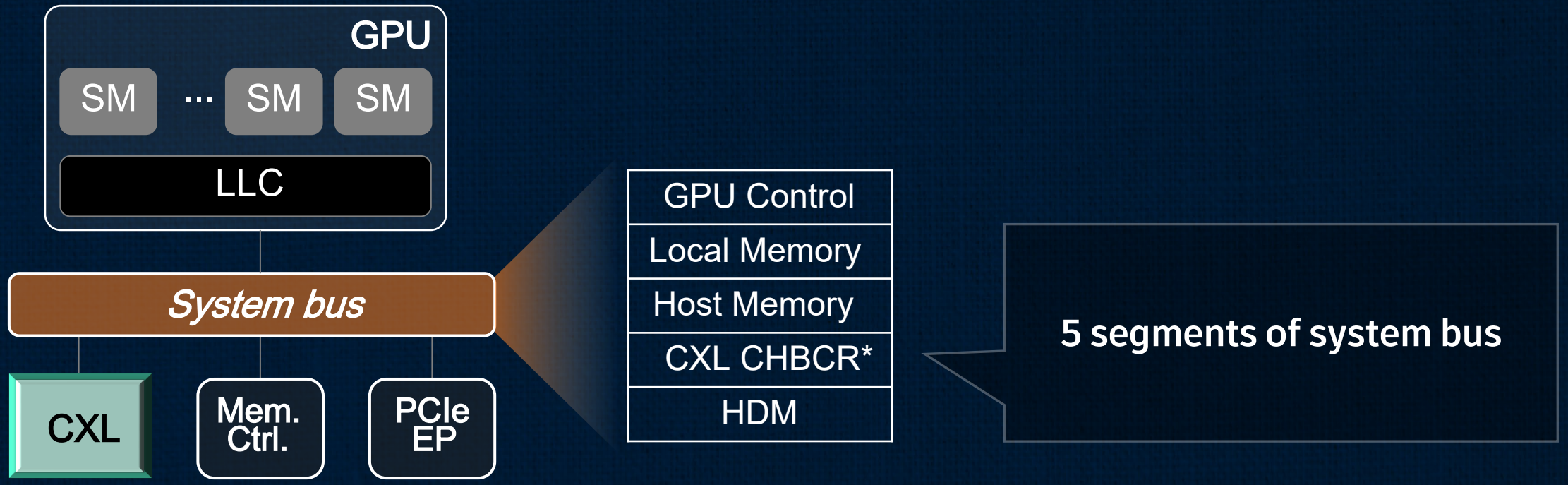
CXL-integrated GPU Architecture



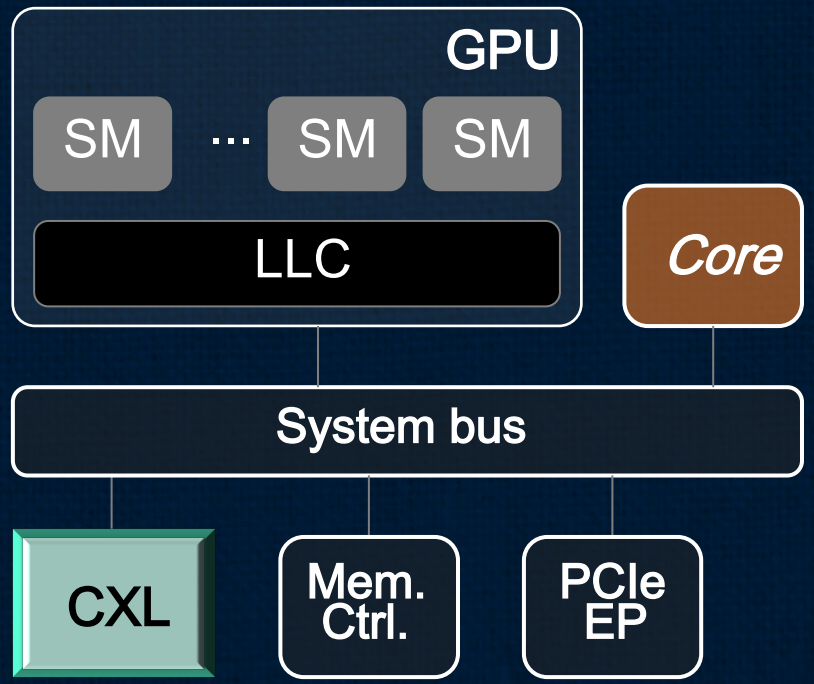
Multiple SMs connected to the LLC

Connected 3 components to the system bus

CXL-integrated GPU Architecture



System Initialization



The memory map must be initialized beforehand!

Simplified complimentary general core

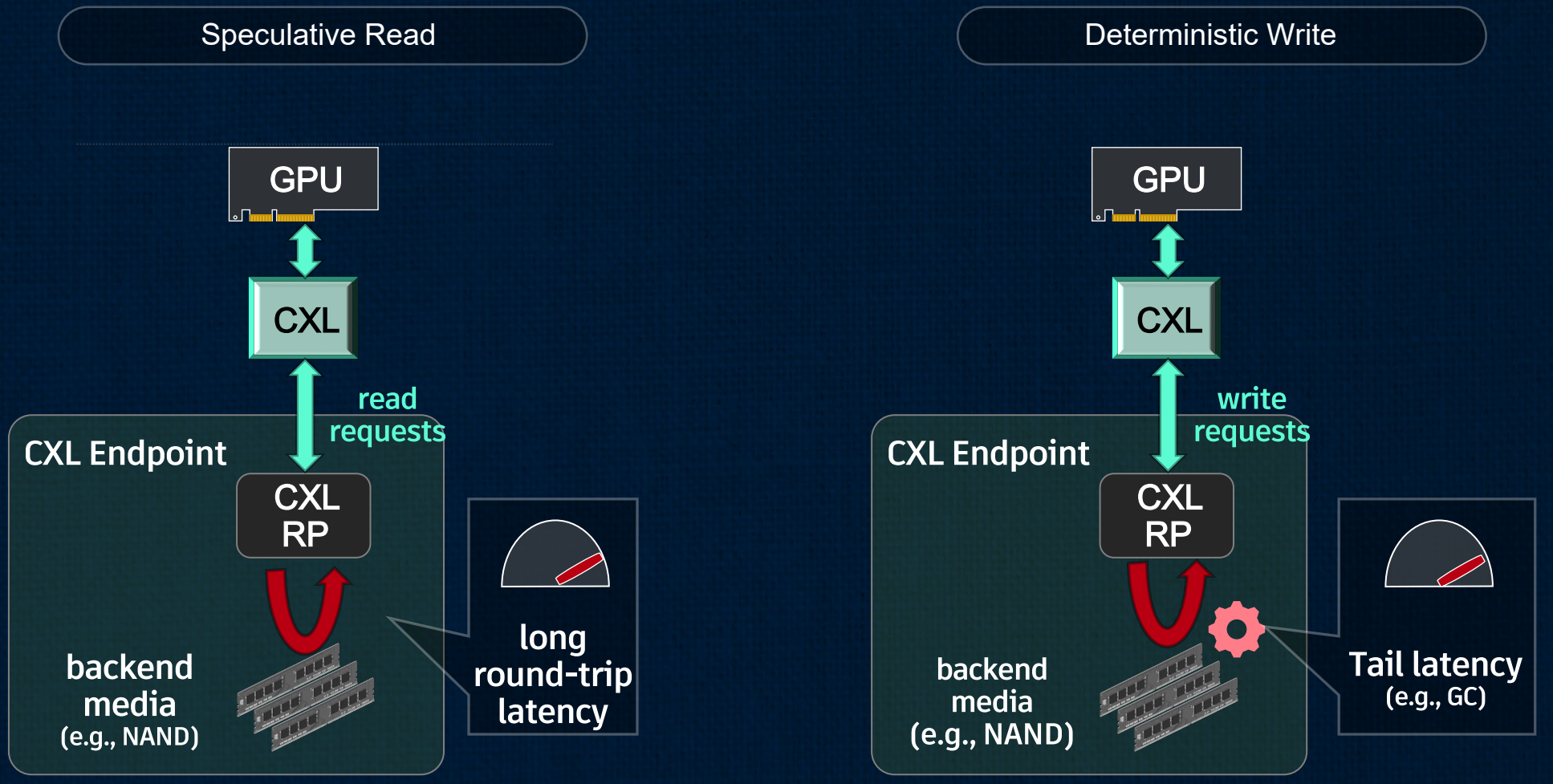
1. GPU Memory Expansion and Potential of CXL

2. Designing a CXL-integrated GPU

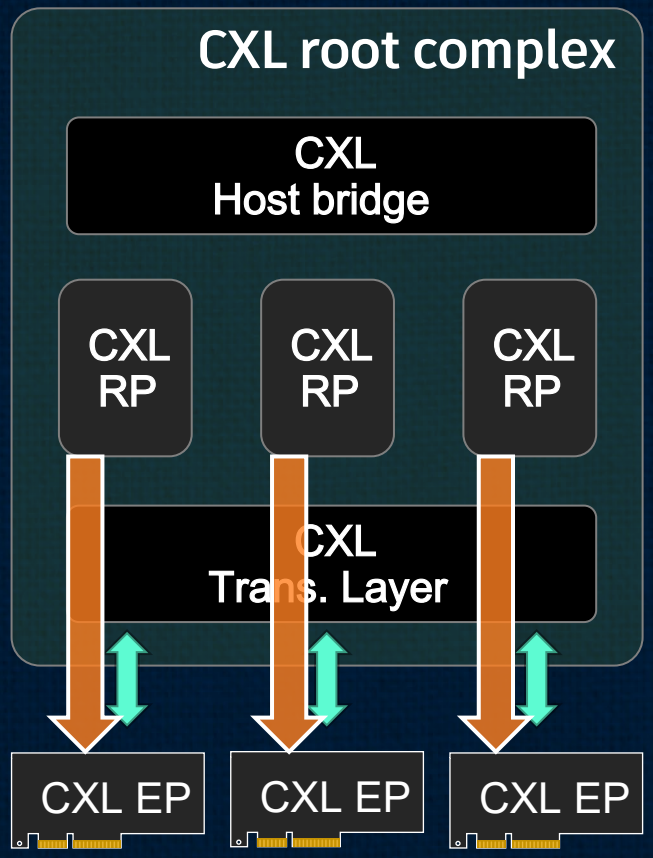
3. Mitigating Backend Media Latency

4. Evaluation Results

Mitigating Backend Media Latency

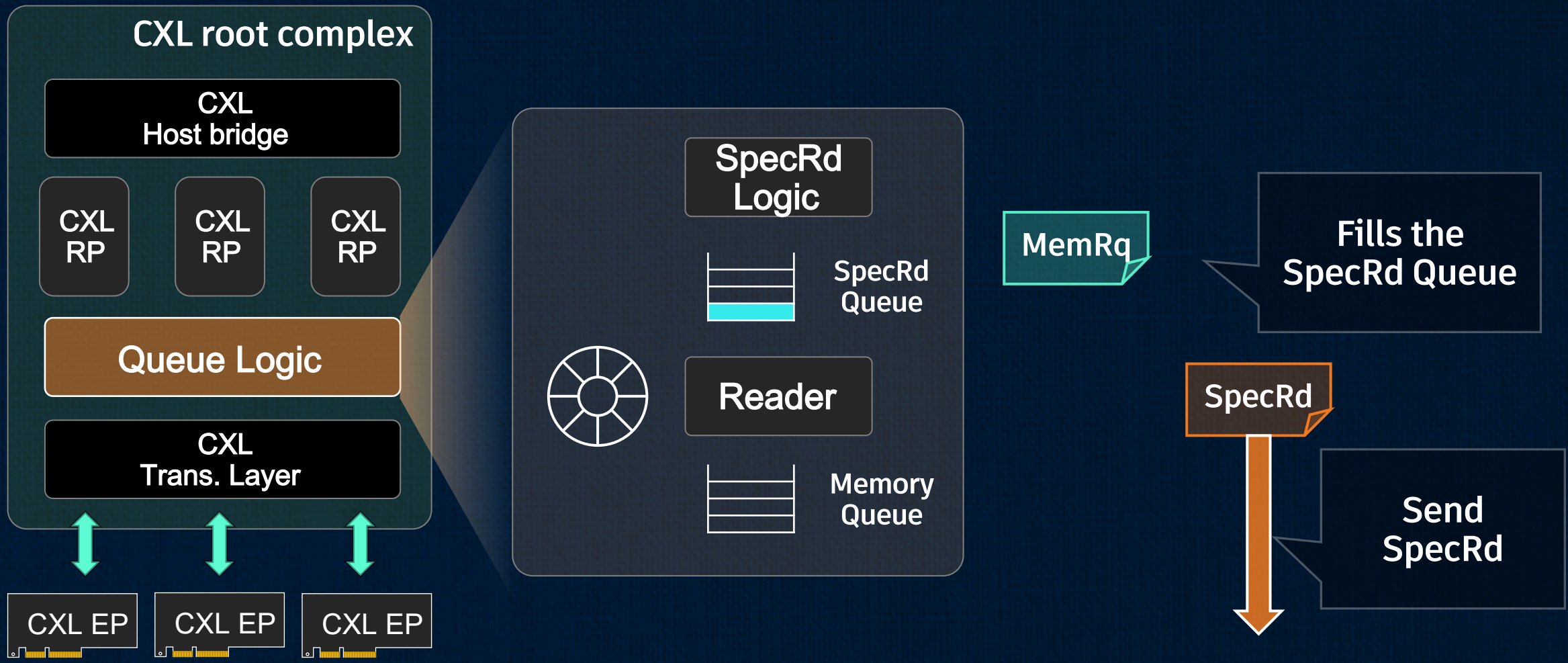


Speculative Read

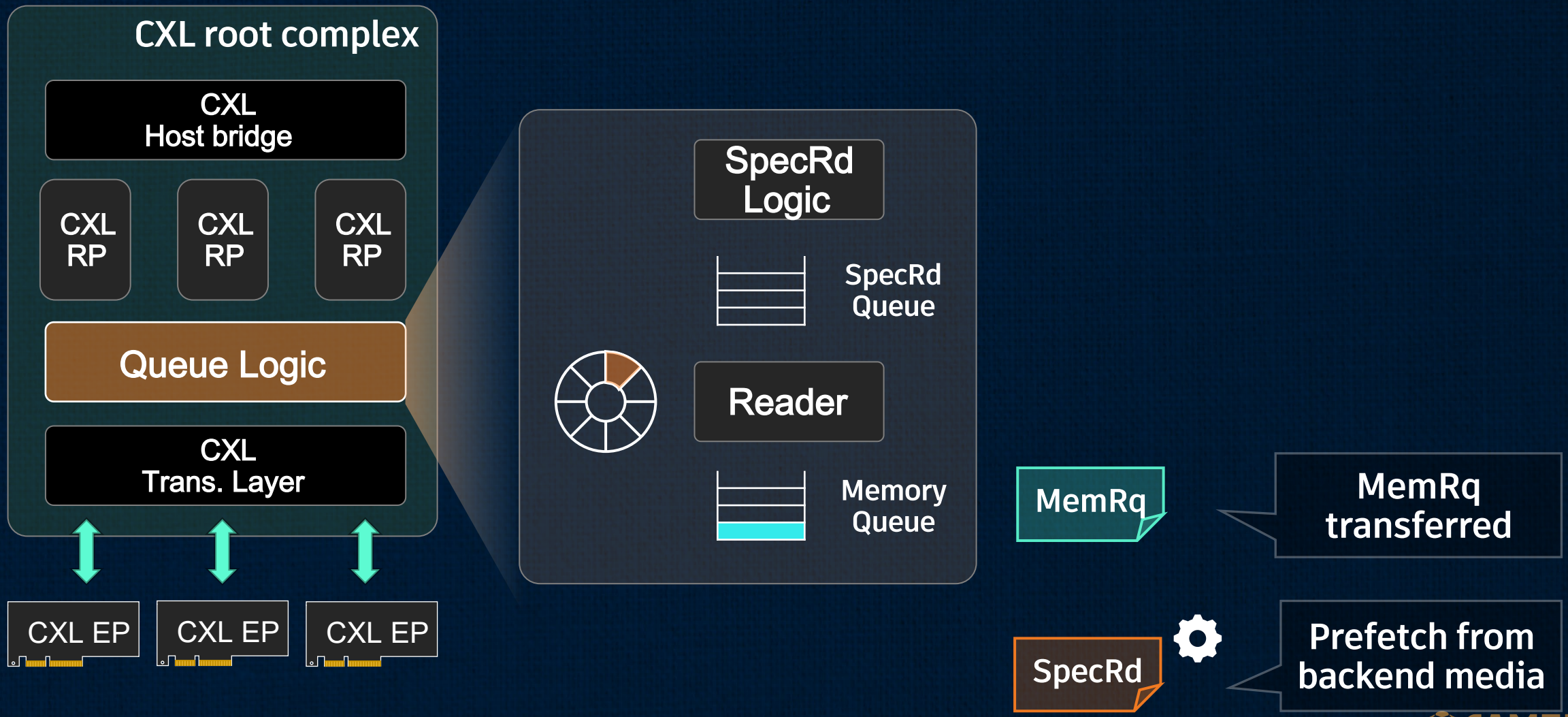


Provide hints about
in-coming memory requests

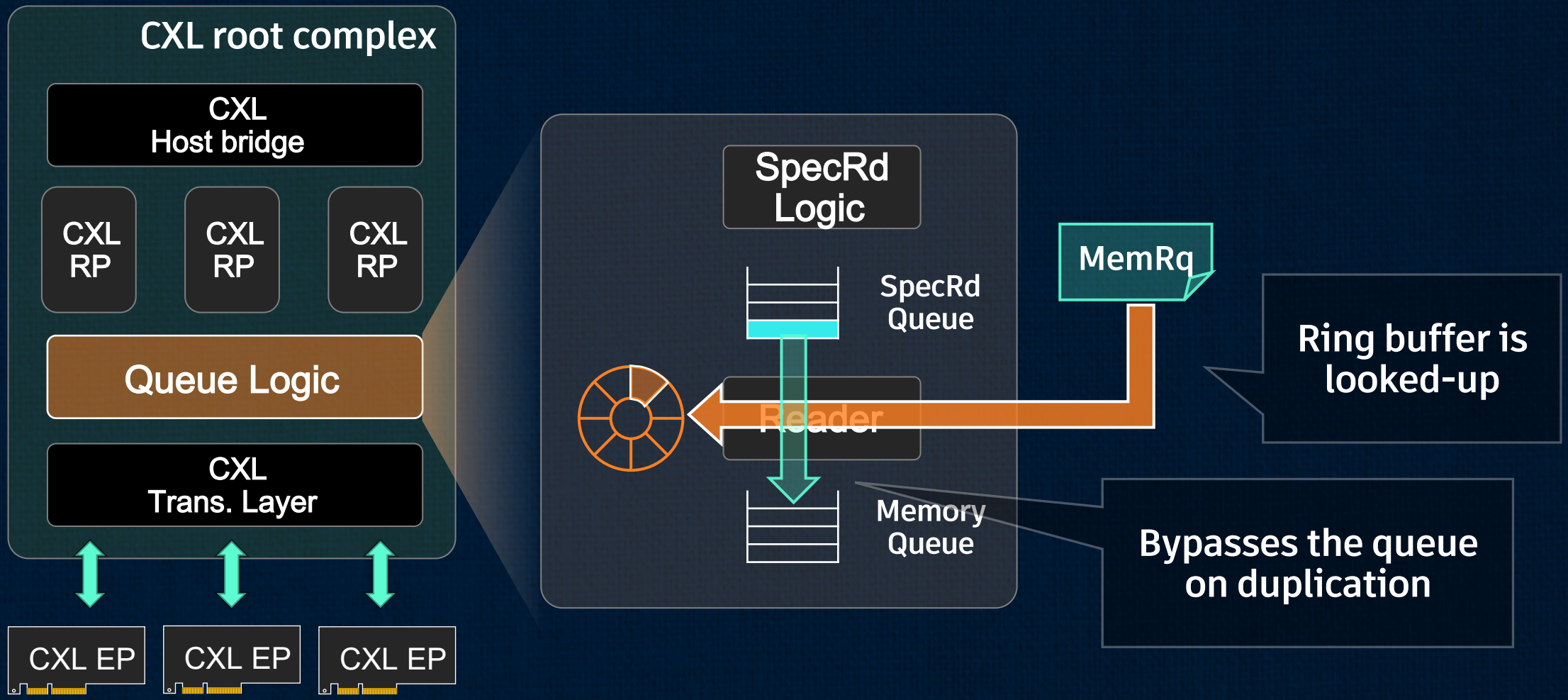
Speculative Read



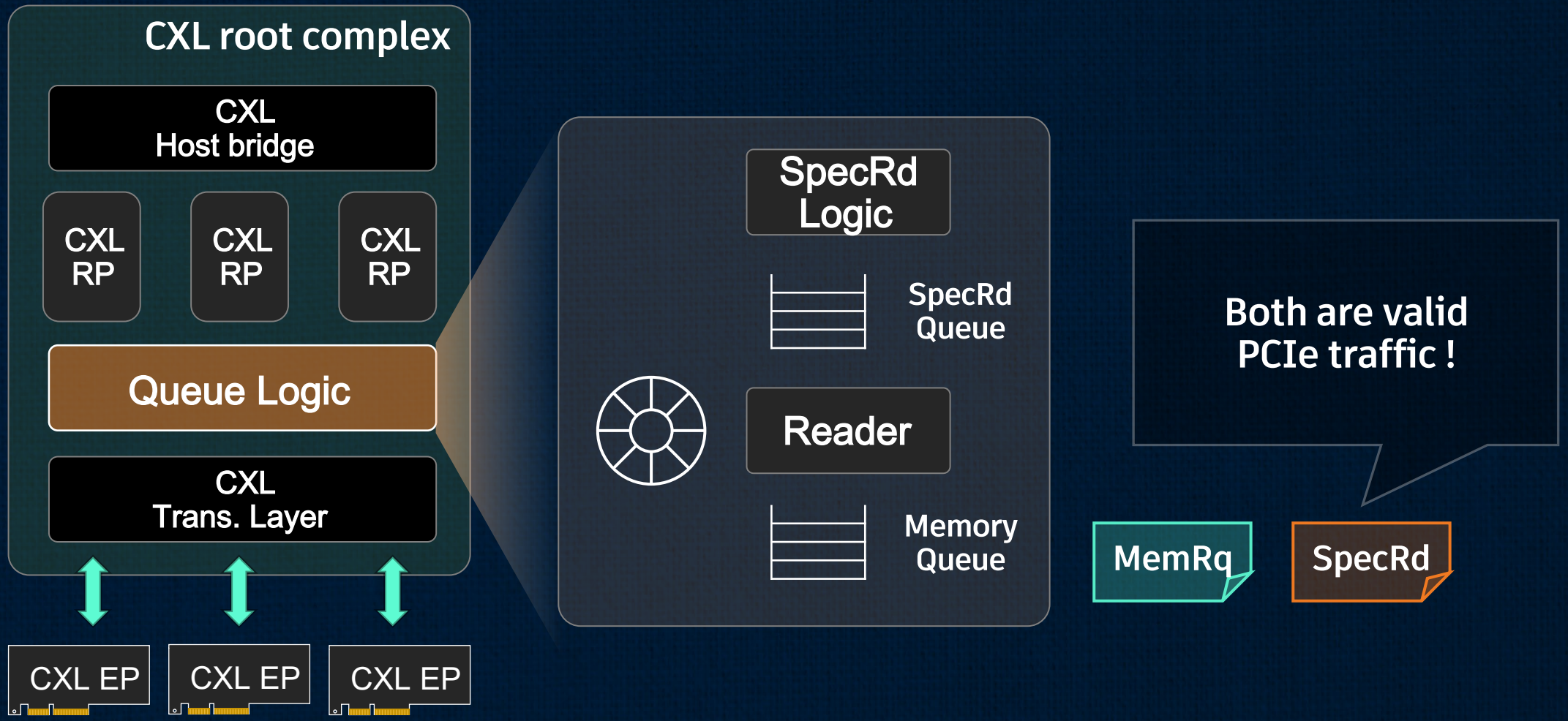
Speculative Read



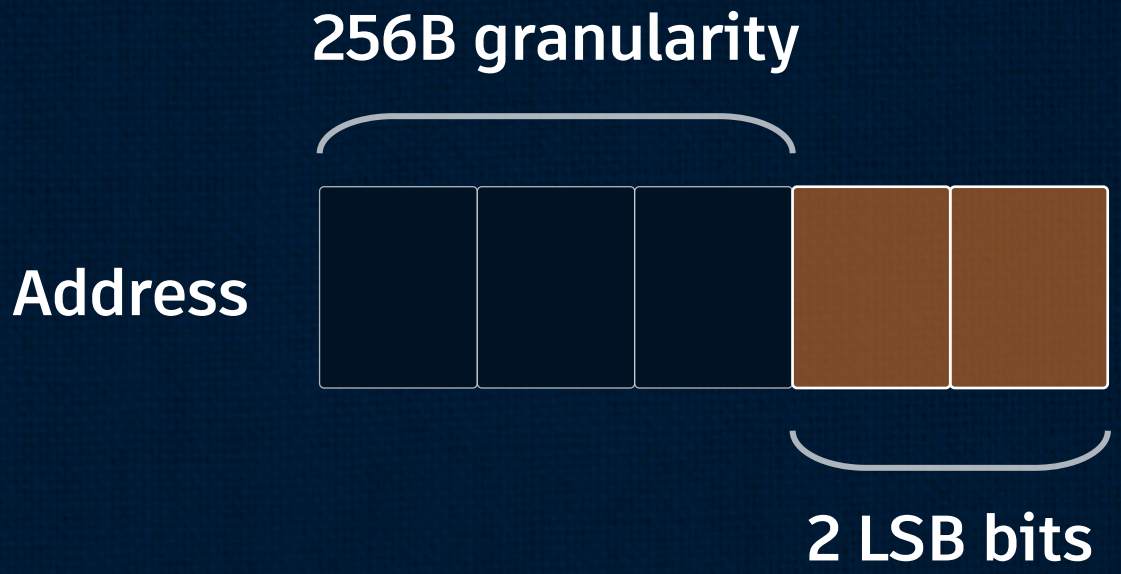
Speculative Read



Speculative Read



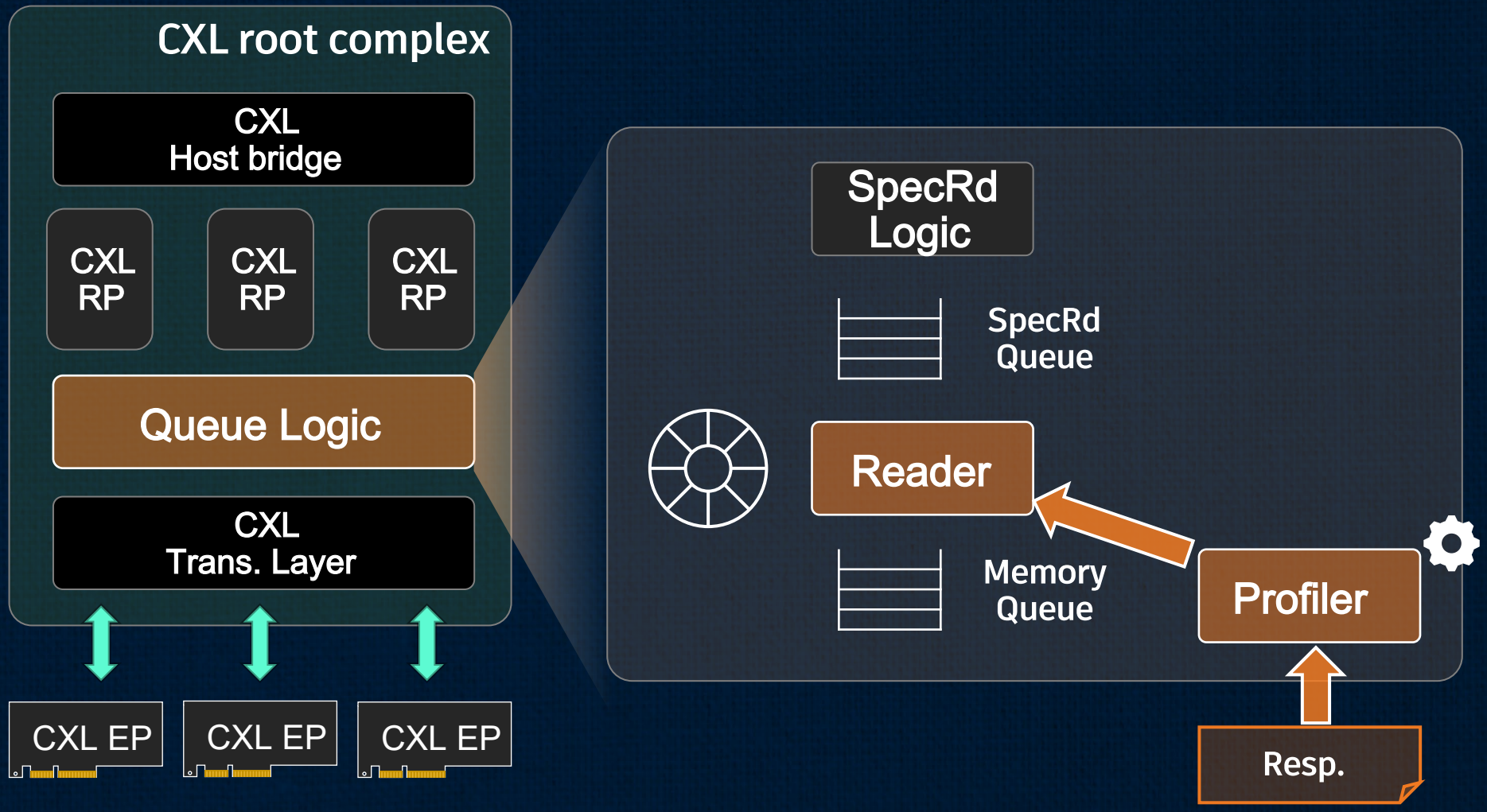
Speculative Read - Optimizations



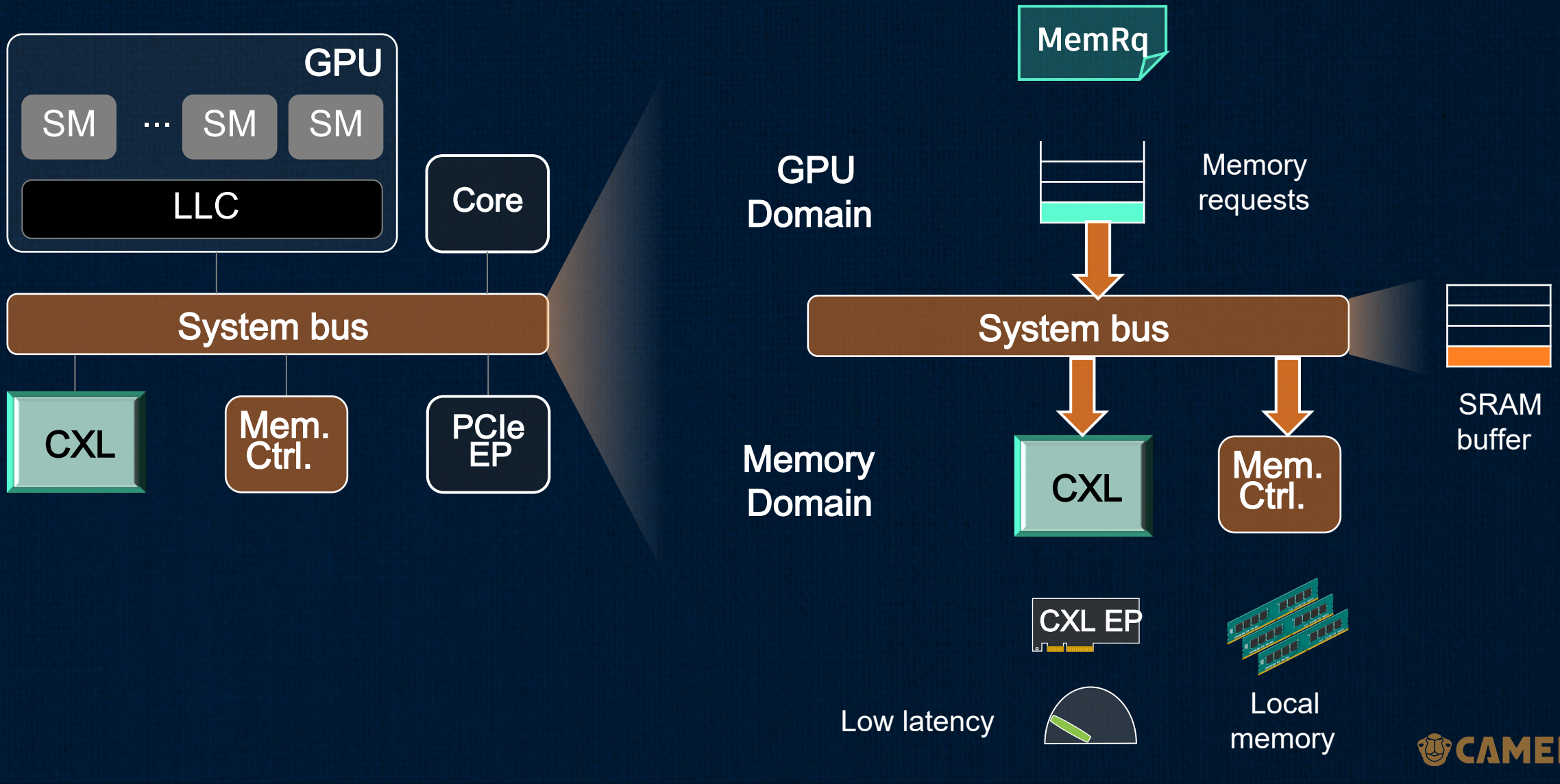
256B ~ 1024B SpecRd

Support GPU's large reads

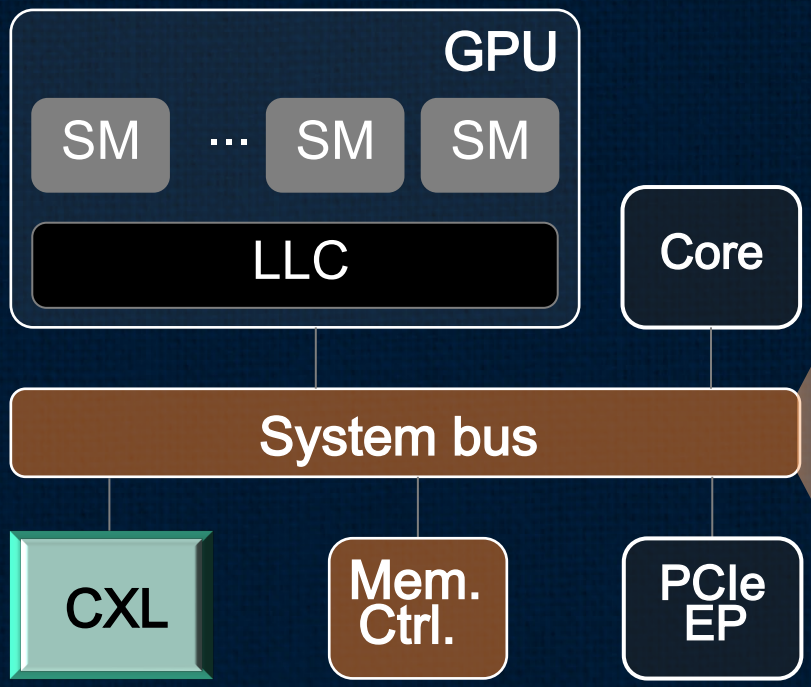
Speculative Read - Optimizations

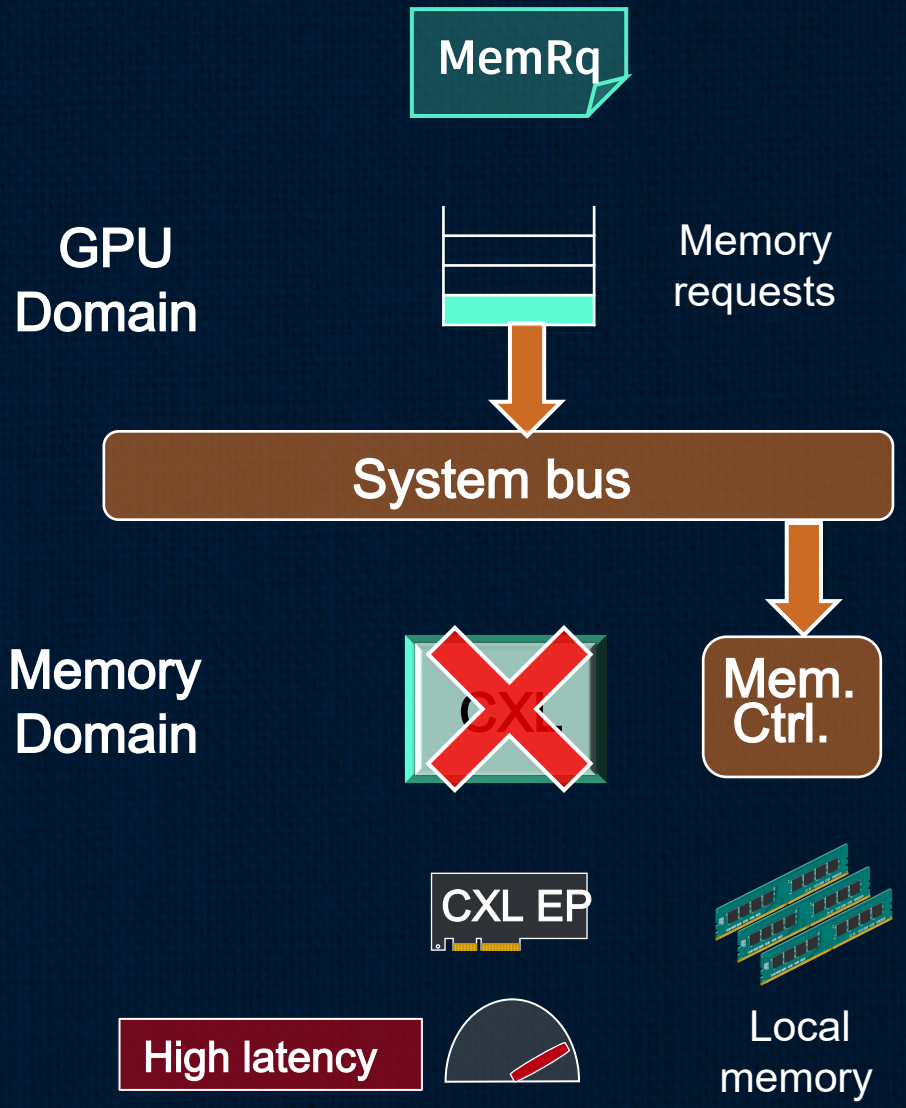


Deterministic Store - Idle Scenario



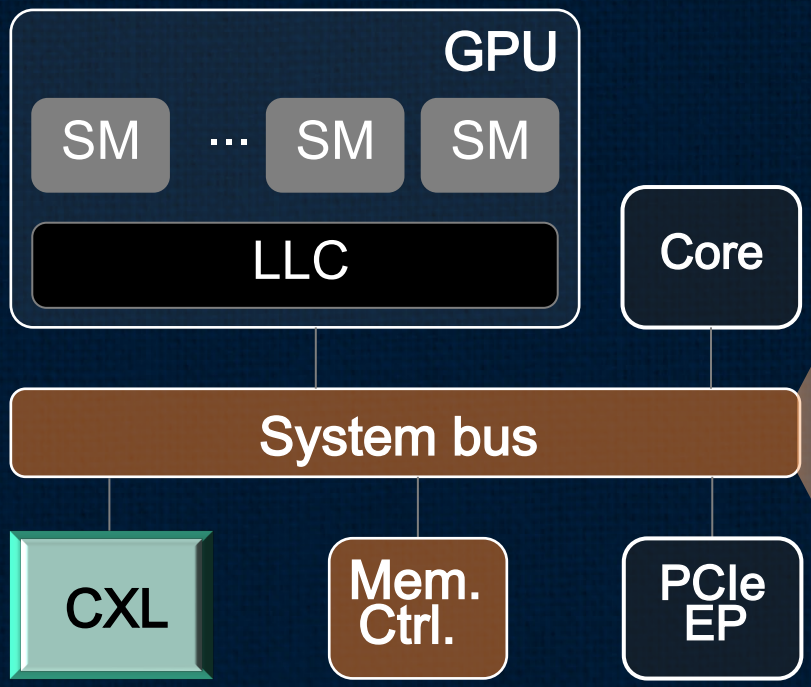
Deterministic Store - Tail Latency Scenario

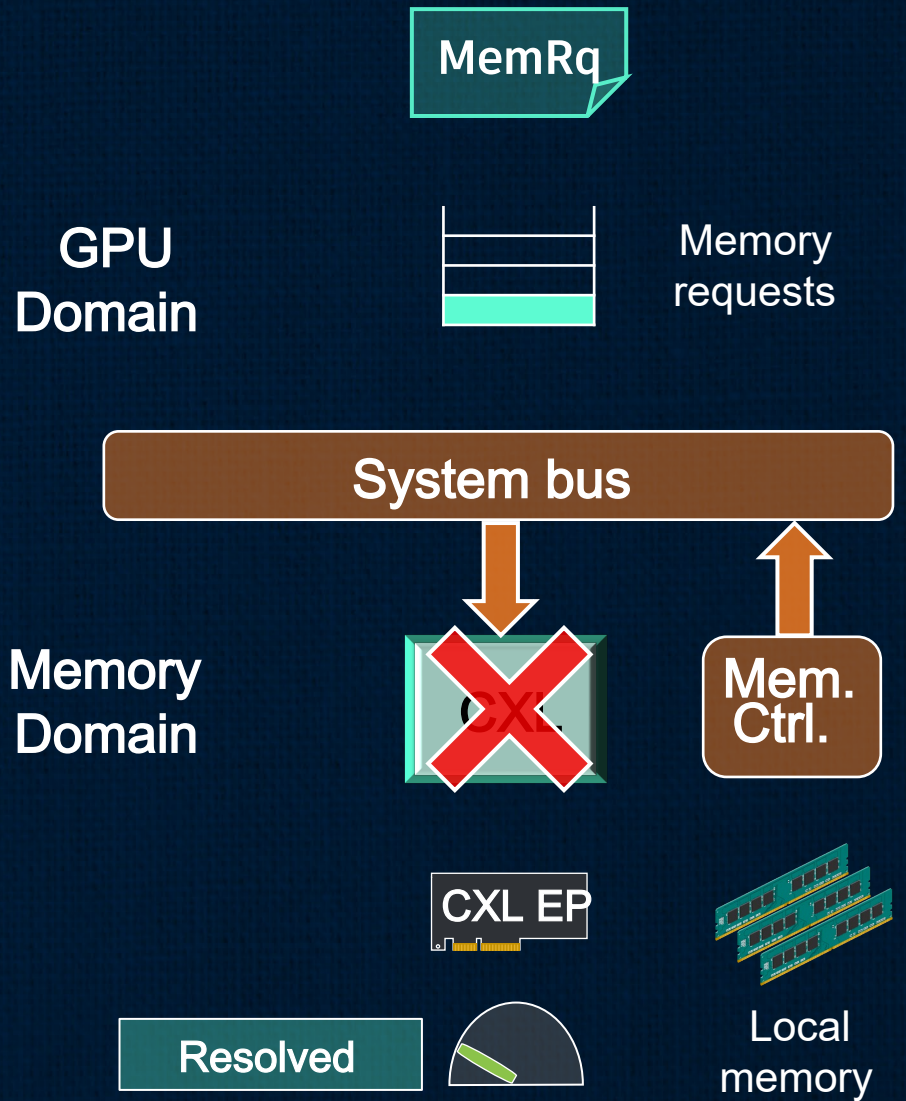




CAMEL

Deterministic Store - Tail Latency Scenario





1. GPU Memory Expansion and Potential of CXL

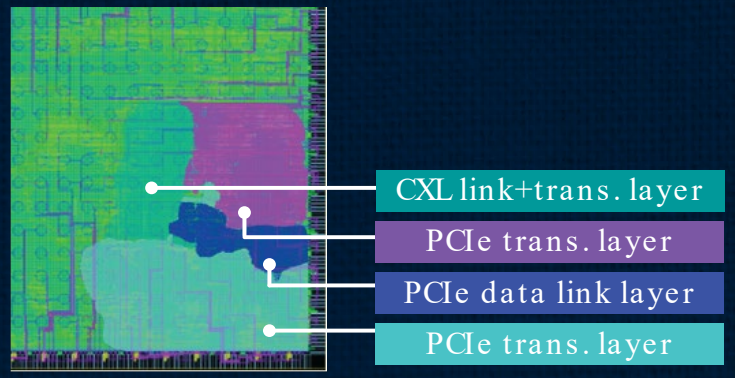
2. Designing a CXL-integrated GPU

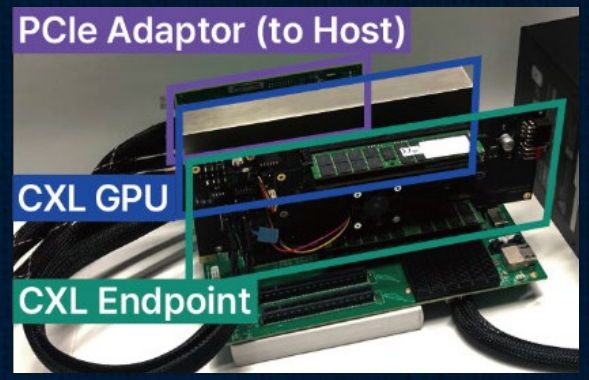
3. Mitigating Backend Media Latency

4. Evaluation Results

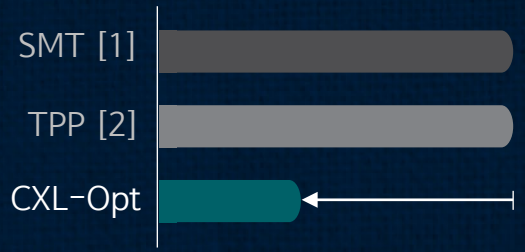
Evaluation Methodology

ASIC Prototype (CXL Controller)





Based on open-sourced GPU RTL (Vortex)



<Round-trip latency>

The proposed CXL controller showed >3x shorter latency

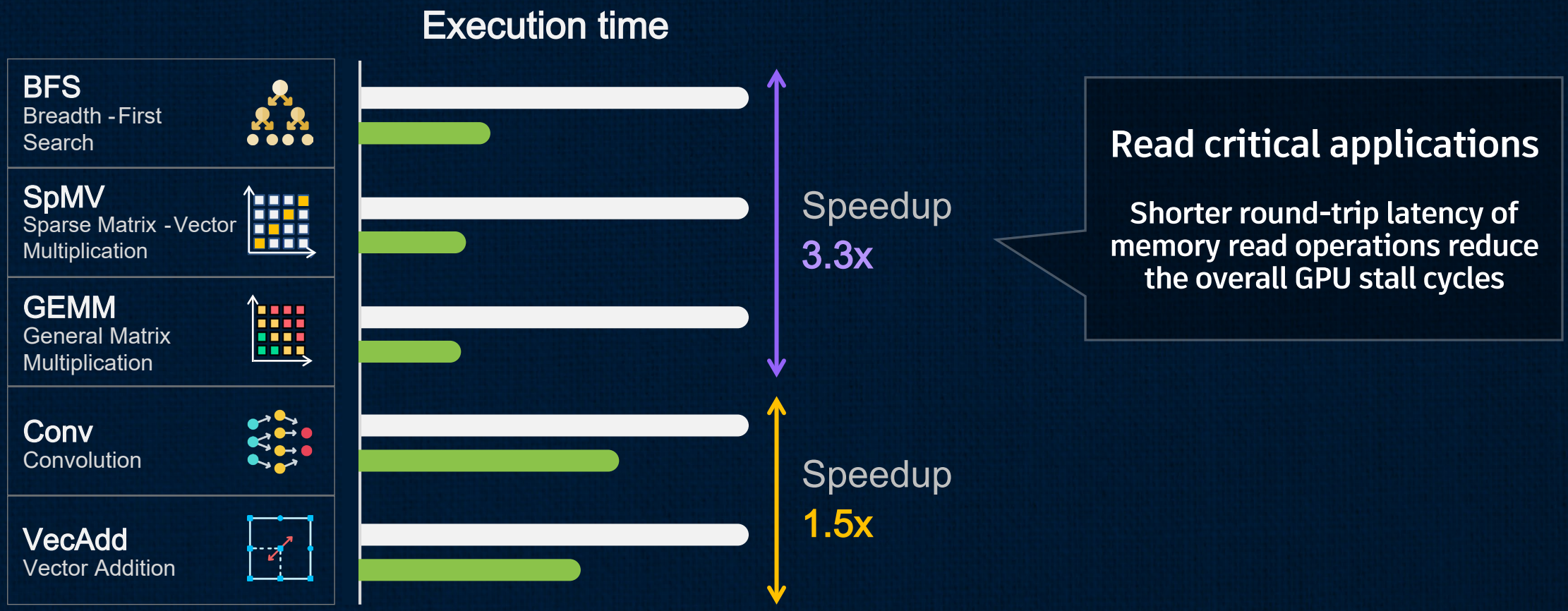
Common setup
• GPU: SMs x 2, cores x 4 • PCIe 5.0 x 8 • CXL 3.1

Compared systems	
UVM	Page fault-based virtual memory management
CXL-GPU	CXL with sub-two digit nanosecond latency

[1] SMT: Software-defined Memory Tiering for Heterogeneous Computing System with CXL Memory Expander
[2] TPP: Transparent Page Placement for CXL-enabled Tiered-memory

CAMEL

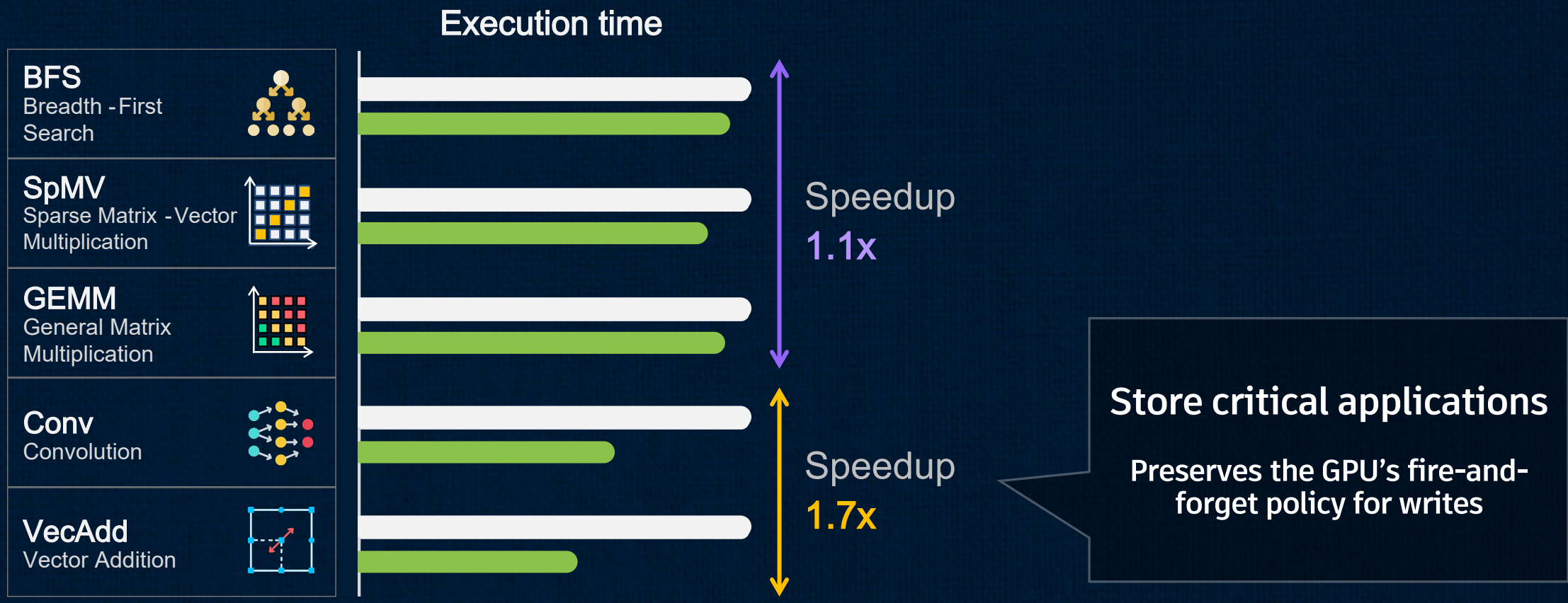
Overall Performance



Read critical applications

Shorter round-trip latency of memory read operations reduce the overall GPU stall cycles

Deterministic Store



Store critical applications

Preserves the GPU's fire-and-forget policy for writes

CAMEL


Thank You
KAIST
CAMEL
panmnesia