SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

Compute Express Link™ (CXL™): Memory and Cache Protocols

Robert Blankenship
Intel Corporation

Topics

What is CXL?

Caching 101

CXL Caching Protocol

CXL Memory Protocol

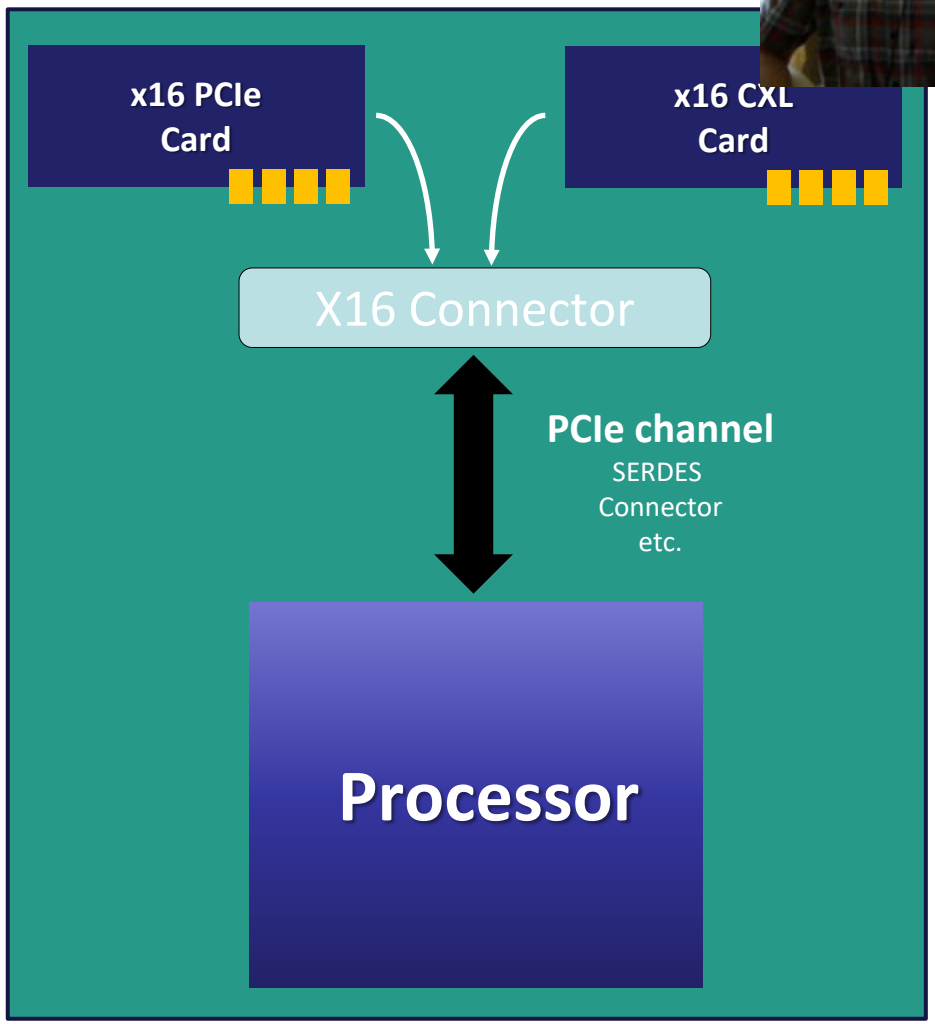
Merging Memory and Caching for Accelerators

What is CXL?

- CXL runs across the standard PCIe physical layer with **new protocols** optimized for cache and memory

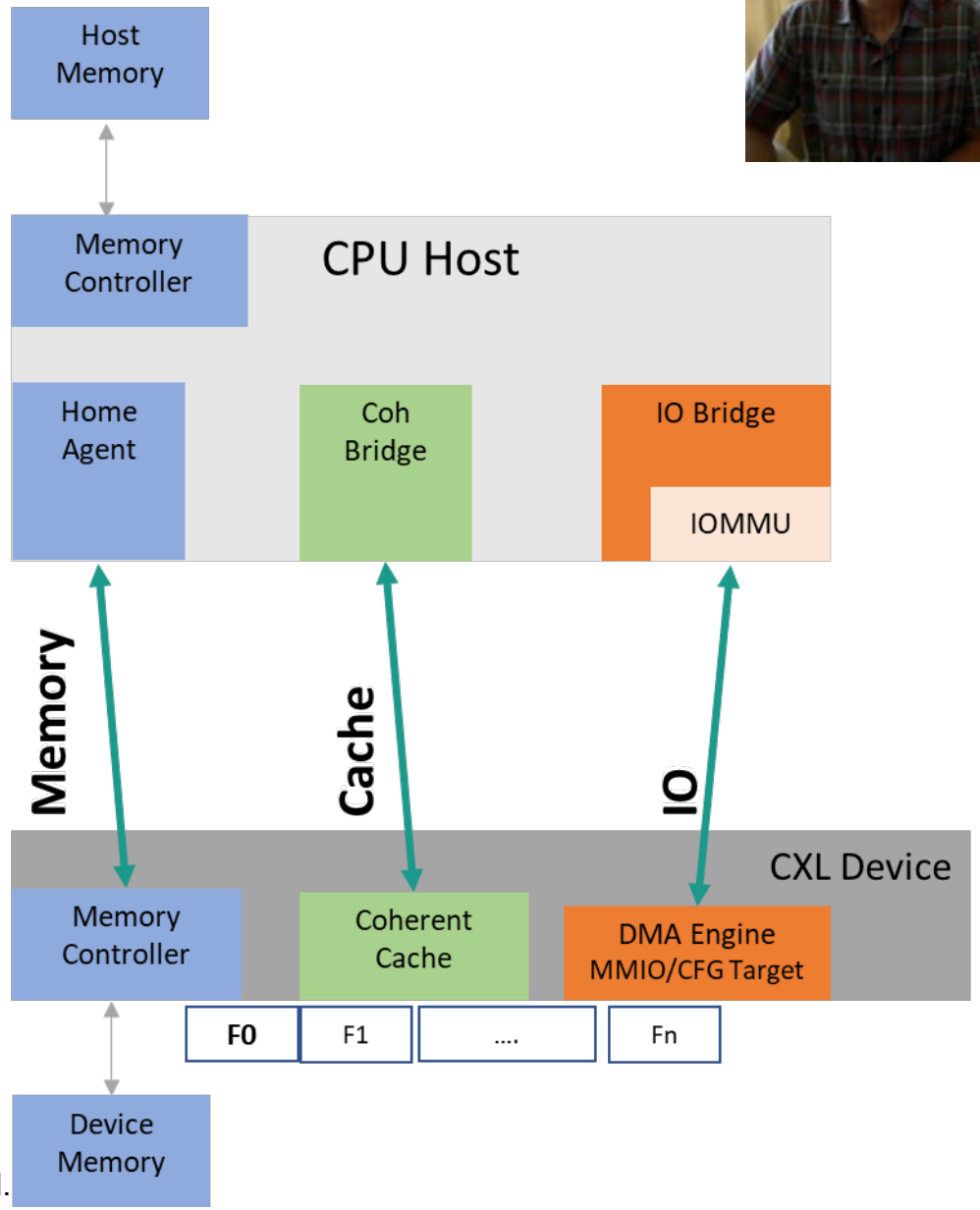
- CXL uses a flexible processor Port that can **auto-negotiate** to either the standard PCIe transaction protocol or the alternate CXL transaction protocols

- First generation CXL aligns to 32 GT/s PCIe 5.0 specification
- CXL usages expected to be key driver for an aggressive timeline to PCIe 6.0 architecture

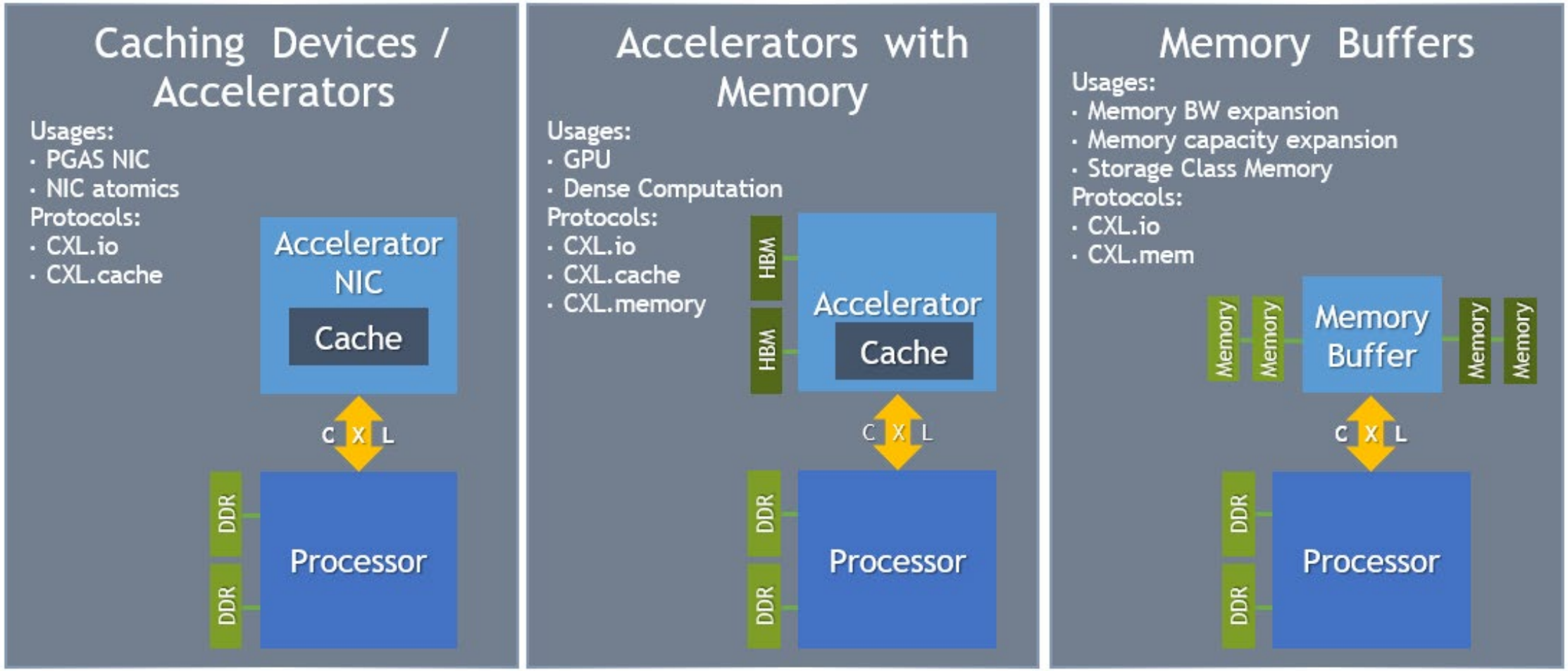


CXL Uses 3 Protocols

- IO (CXL.io) →
  Same as PCI Express®
- **Memory (CXL.mem) →**
  Memory Access from CPU Host to Device
- **Cache (CXL.cache) →**
  Coherent Access from Device to CPU Host



Representative CXL Usages



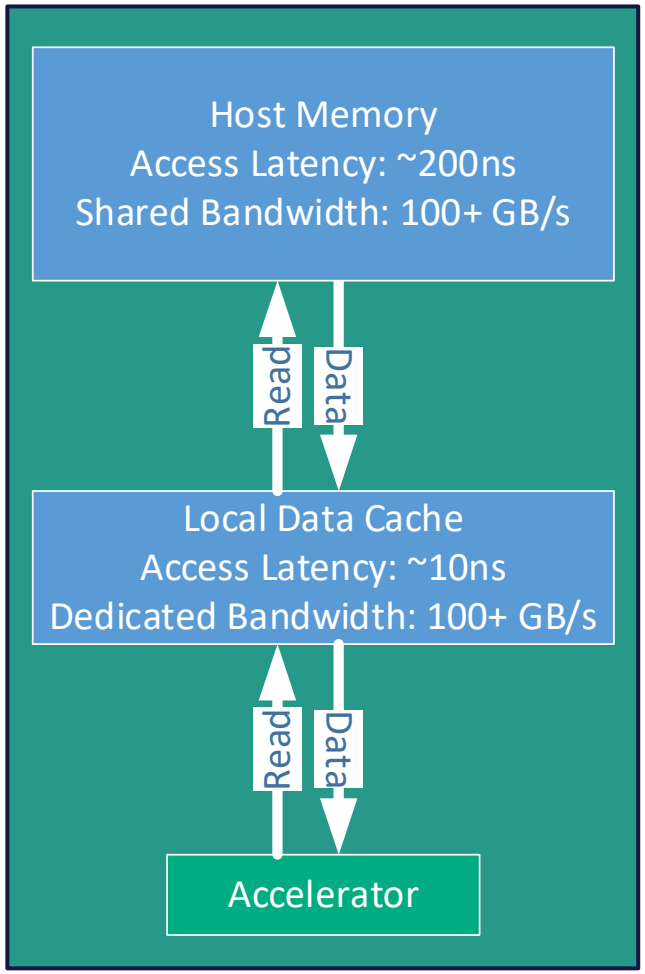
Caching 101

Caching Overview

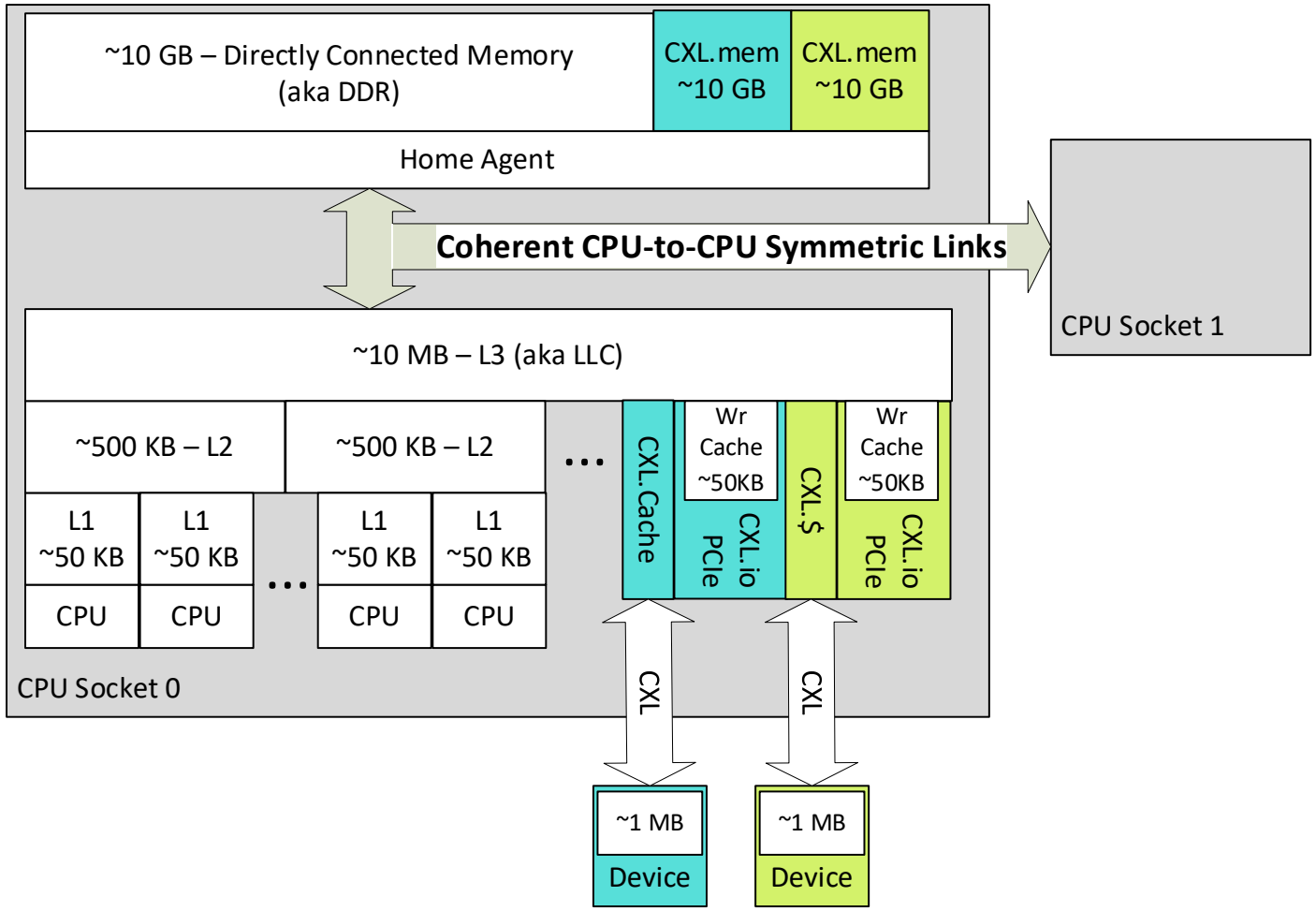
Caching temporarily brings data closer to the consumer

Improves latency and bandwidth using prefetching and/or locality

- Prefetching: Loading Data into cache before it is required
- Spatial Locality (locality is space): Access address X then X+n
- Temporal Locality (locality in Time): Multiple access to the same Data



CPU Cache/Memory Hierarchy with CXL



- Modern CPUs have 2 or more levels of coherent cache
- Lower levels (L1), smaller in capacity with lowest latency and highest bandwidth per source.
- Higher levels (L3), less bandwidth per source but much higher capacity and support more sources
- Device caches are expected to be up to 1MB.

Note: Cache/Memory capacities are examples and not aligned to a specific product.

Cache Consistency

How do we make sure updates in cache are visible to other agents?

- Invalidate all peer caches prior to update
- Can managed with software or hardware → CXL uses hardware coherence

Define a point of "Global Observation" (aka GO) when new data is visible from writes

Tracking granularity is a "cacheline" of data → 64-bytes for CXL

All addresses are assumed to be Host Physical Address (HPA) in CXL cache and memory protocols → Translations using existing Address Translation Services (ATS).

Cache Coherence Protocol

- Modern CPU caches and CXL are built on M,E,S,I protocol/states
  - **M**odified – Only in one cache, Can be read or written, Data **NOT** up-to-date in memory
  - **E**xclusive – Only in one cache, Can be read or written, Data **IS** up-to-date in memory
  - **S**hared – Can be in many caches, Can only be read, Data IS up-to-date in memory
  - **I**nvalid – Not in cache
- M,E,S,I is tracked for each cacheline address in each cache
  - Cacheline address in CXL is Addr[51:6]
- Notes:
  - Each level of the CPU cache hierarchy follows MESI and layers above must be consistent
  - Other extended states and flows are possible but not covered in context of CXL

How are Peer Caches Managed?

- All peer caches managed by the "Home Agent" within the cache level → Hidden from CXL device

- A "Snoop" is the term for the Home to check cache state and may cause cache state changes

- CXL Snoops:
  - Snoop Invalidate (SnpInv): Cache to degrade to I-state, and must return any Modified data
  - Snoop Data (SnpData): Cache to degrade to S-state, and must return any Modified data.
  - Snoop Current (SnpCurr): Cache state does not change, but must return any Modified data

CXL Cache Protocol

Cache Protocol Summary

- Simple set of 15 cache-able reads and writes from the device to host memory
- Keep complexity of global coherence management in the host

Cache Protocol Channels

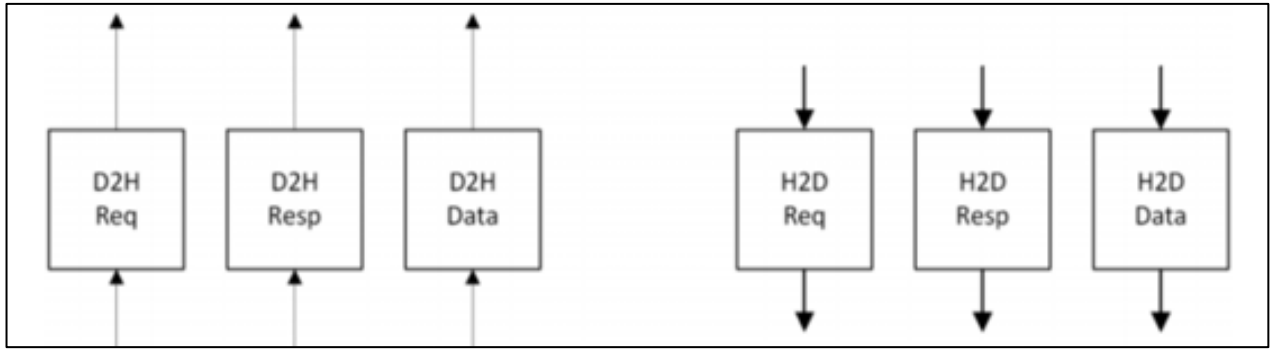
3 channels in each direction: D2H vs H2D

Data and RSP channels are pre-allocated

D2H Requests from the device

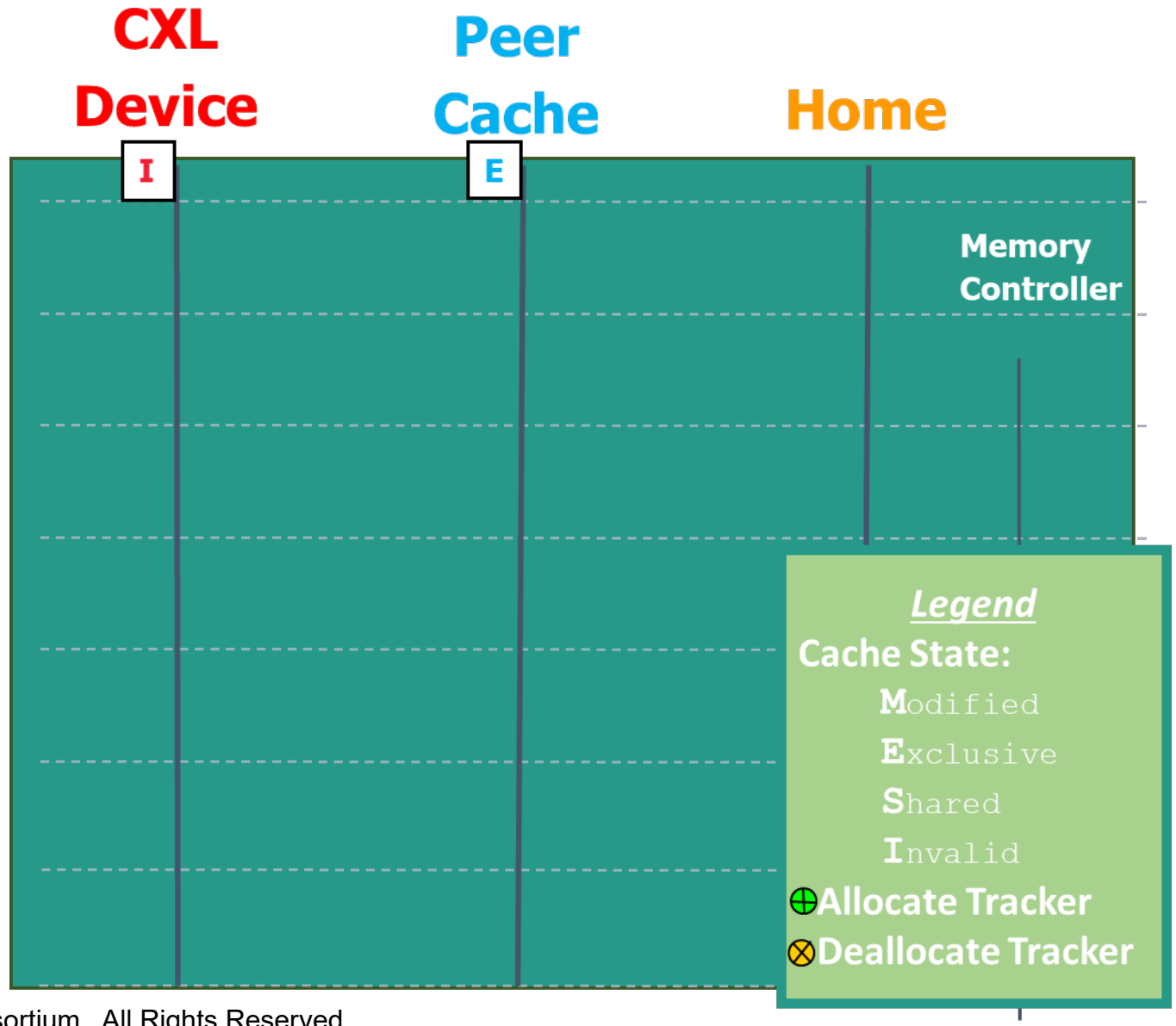
H2D Requests are snoops from the host

Ordering: H2D Req (Snoop) push H2D RSP



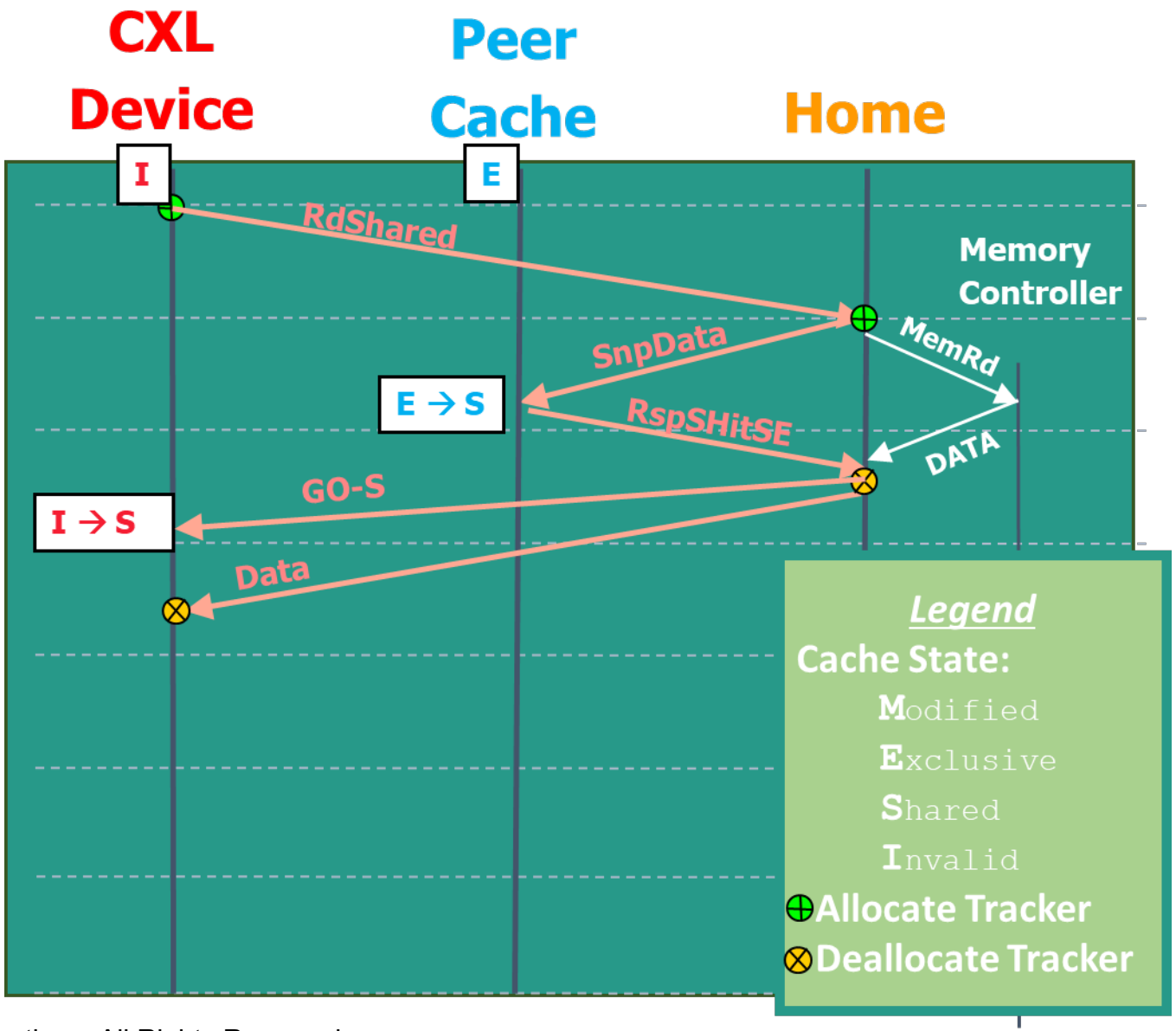
Read Flow

- Diagram to show message flows in time
  - X-axis: Agents
  - Y-axis: Time

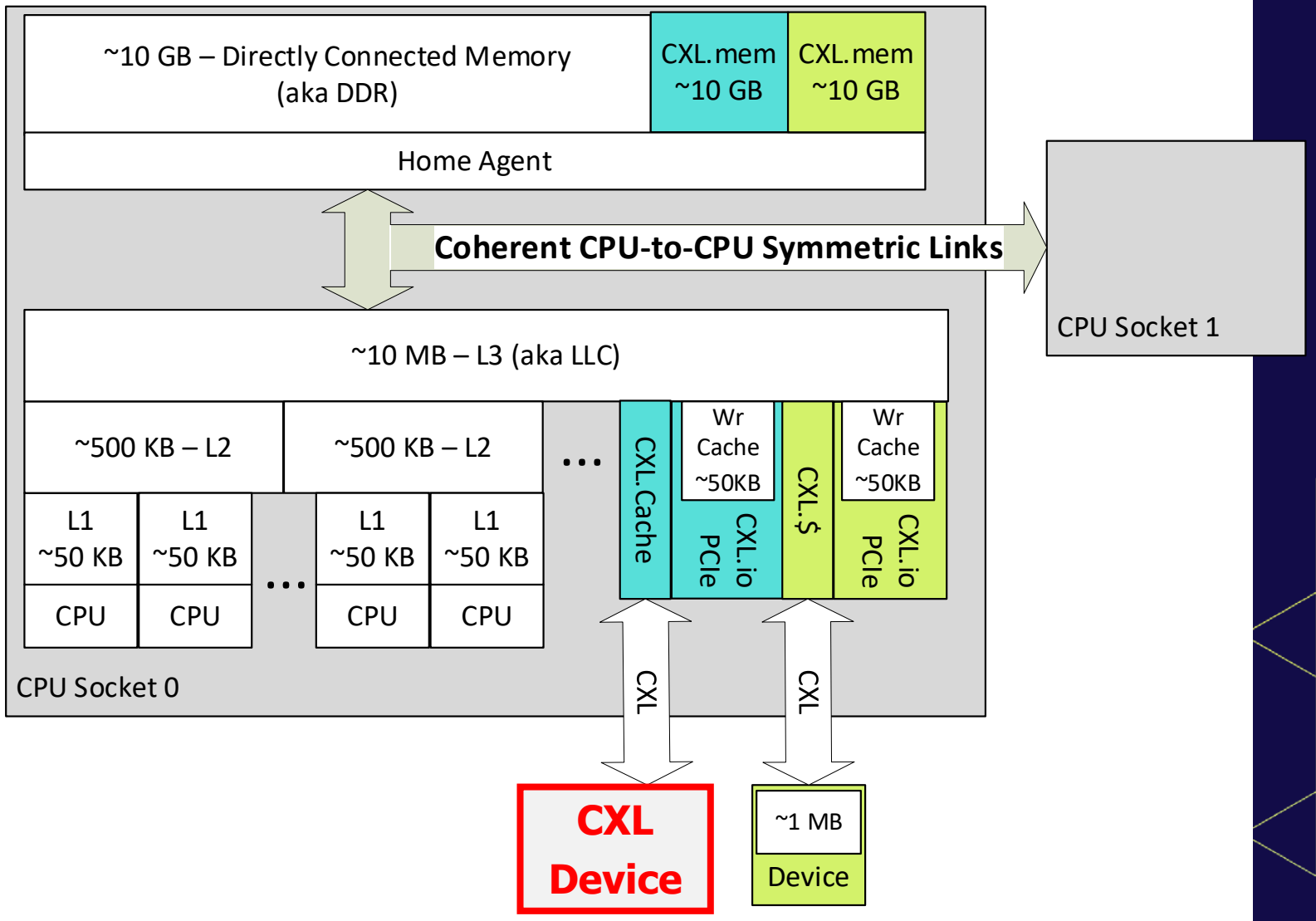


Read Flow

- Diagram to show message flows in time
  - X-axis: Agents
  - Y-axis: Time

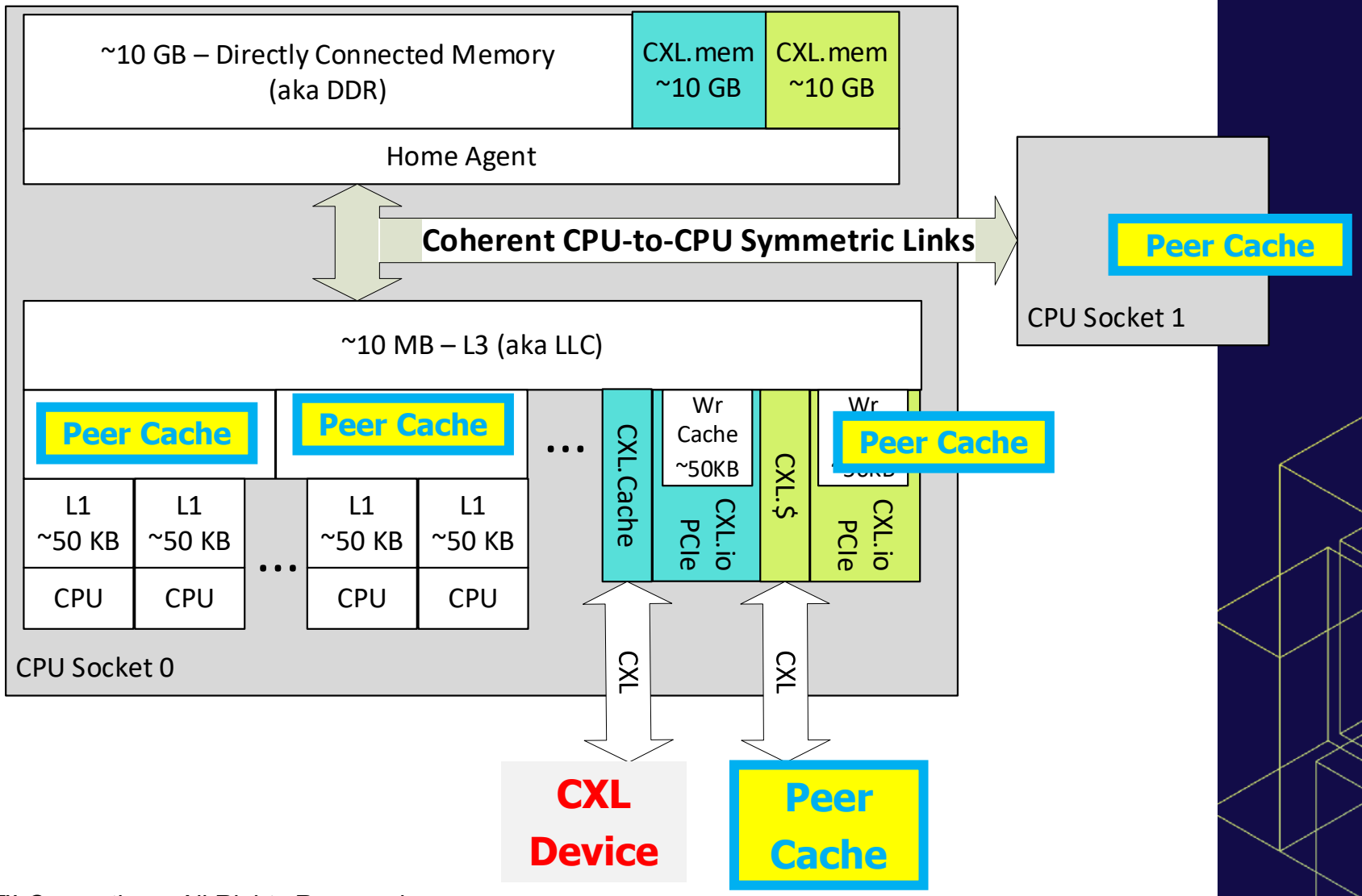


Mapping Flow Back to CPU Hierarchy



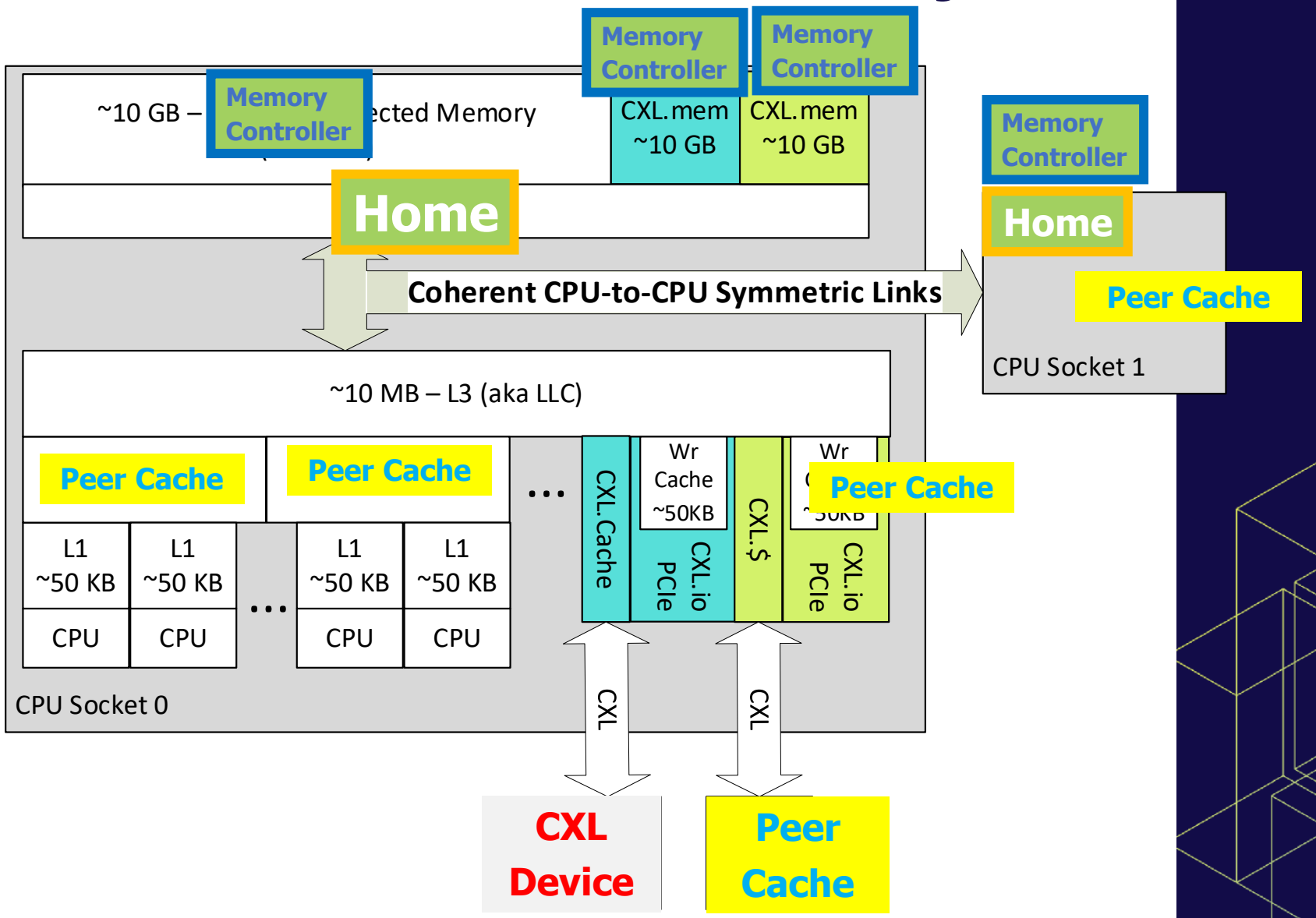
Mapping Flow Back to CPU Hierarchy

- Peer Cache can be:
  - Peer CXL Device with Cache
  - CPU Cache in Local Socket
  - CPU Cache in Remote Socket



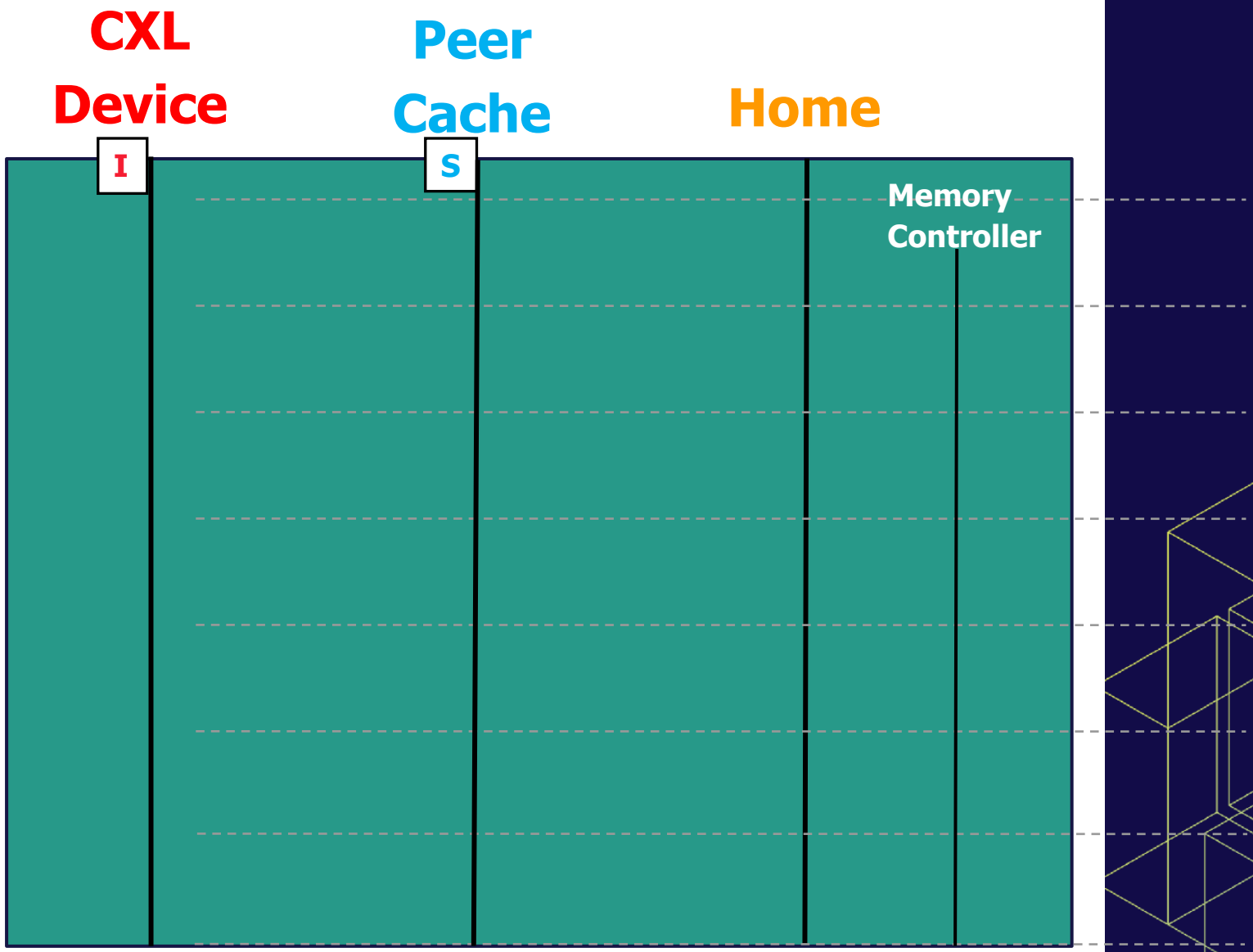
Mapping Flow Back to CPU Hierarchy

- Peer Cache can be:
  - Peer CXL Device with Cache
  - CPU Cache in Local Socket
  - CPU Cache in Remote Socket
- Memory Controller can be:
  - Native DDR on Local Socket
  - Native DDR on Remote Socket
  - CXL.mem on peer Device



Example #2: Write

- For Cache Writes there are three phases:
  - Ownership
  - Silent Write
  - Cache Eviction



Legend

Cache State:

Modified

Exclusive

Shared

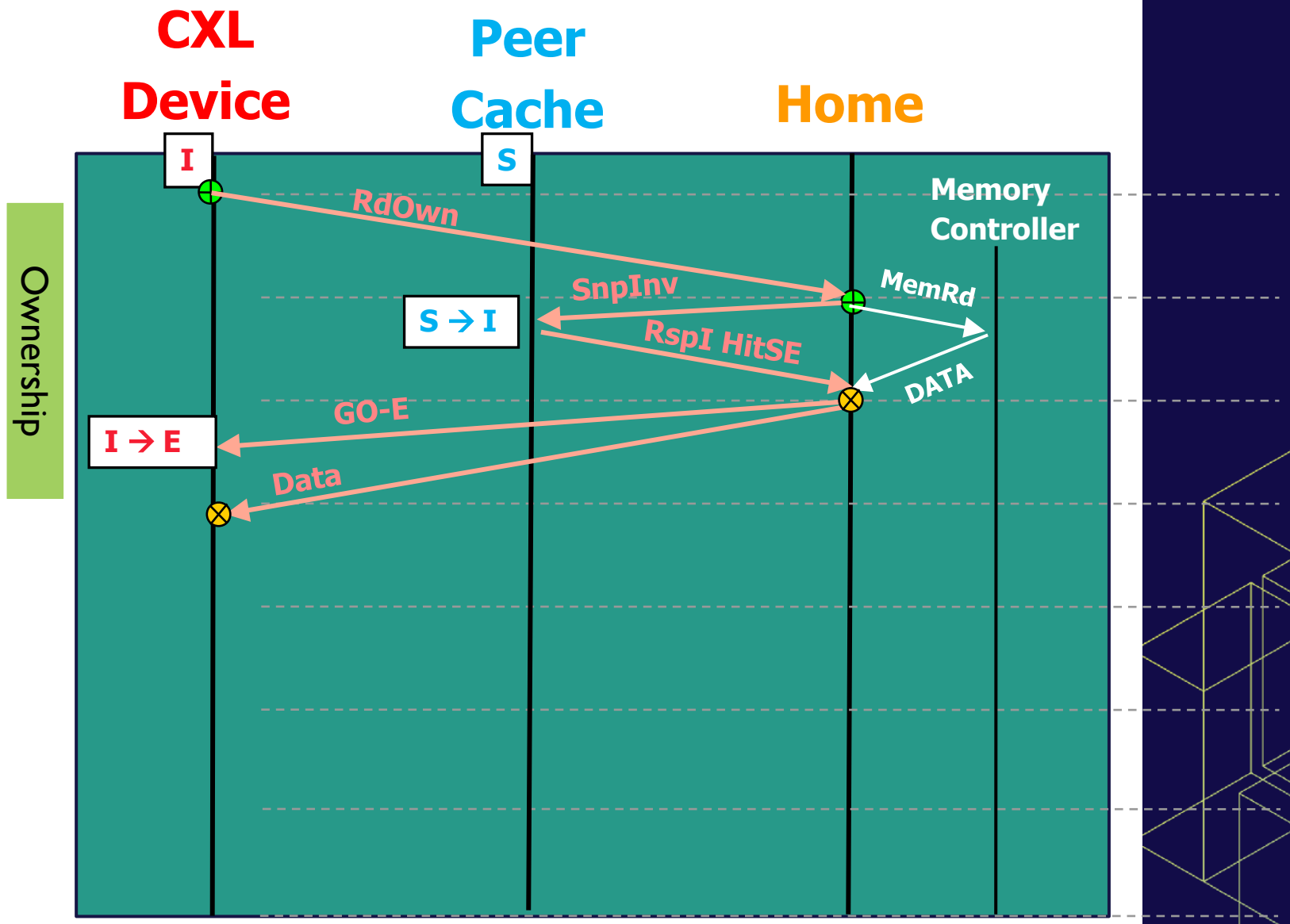
Invalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Example #2: Write

- For Cache Writes there are three phases:
  - **Ownership**
  - Silent Write
  - Cache Eviction



Legend

Cache State:

Modified

Exclusive

Shared

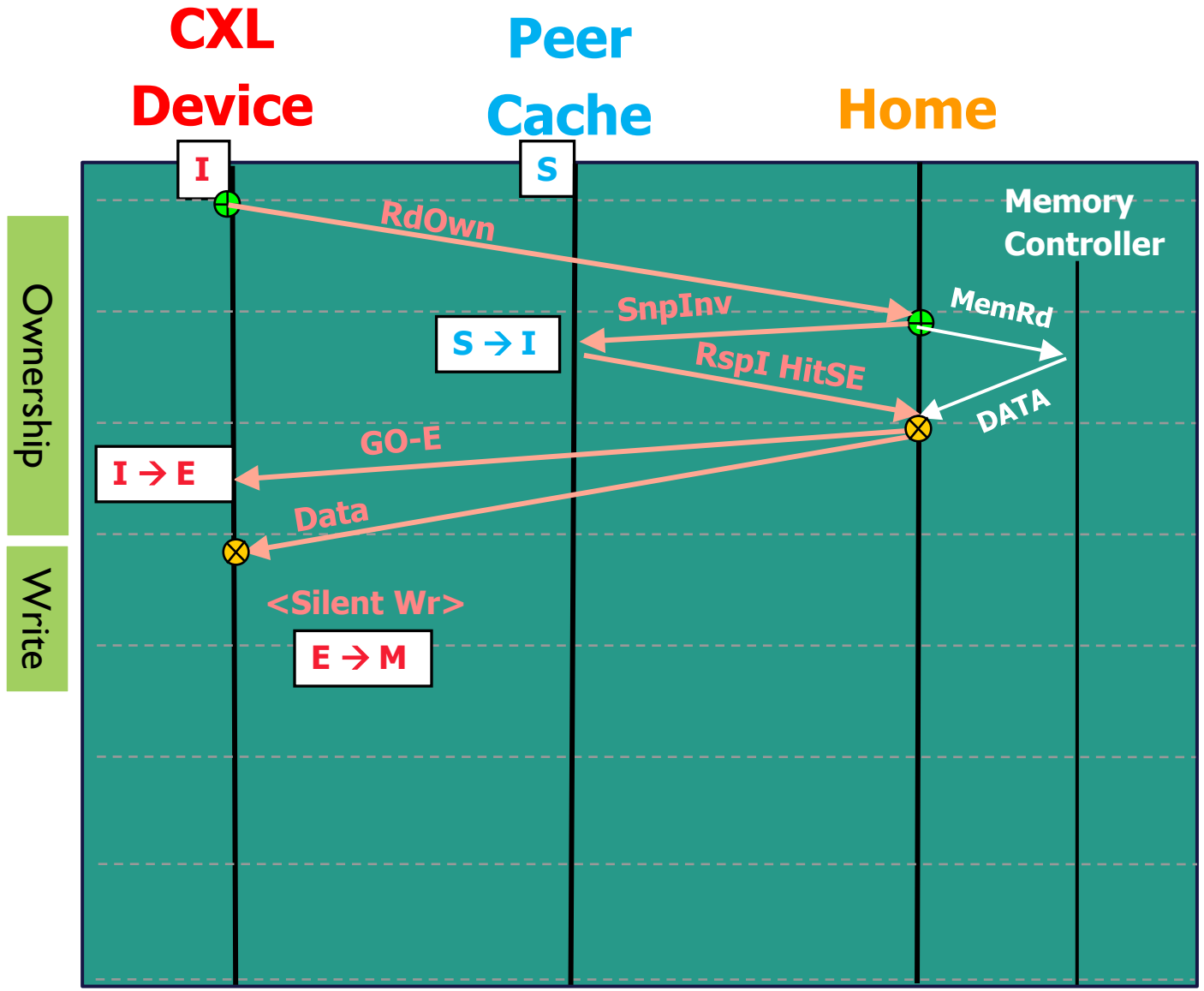
Invalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Example #2: Write

- For Cache Writes there are three phases:
  - Ownership
  - **Silent Write**
  - Cache Eviction



Legend

Cache State:

Modified

Exclusive

Shared

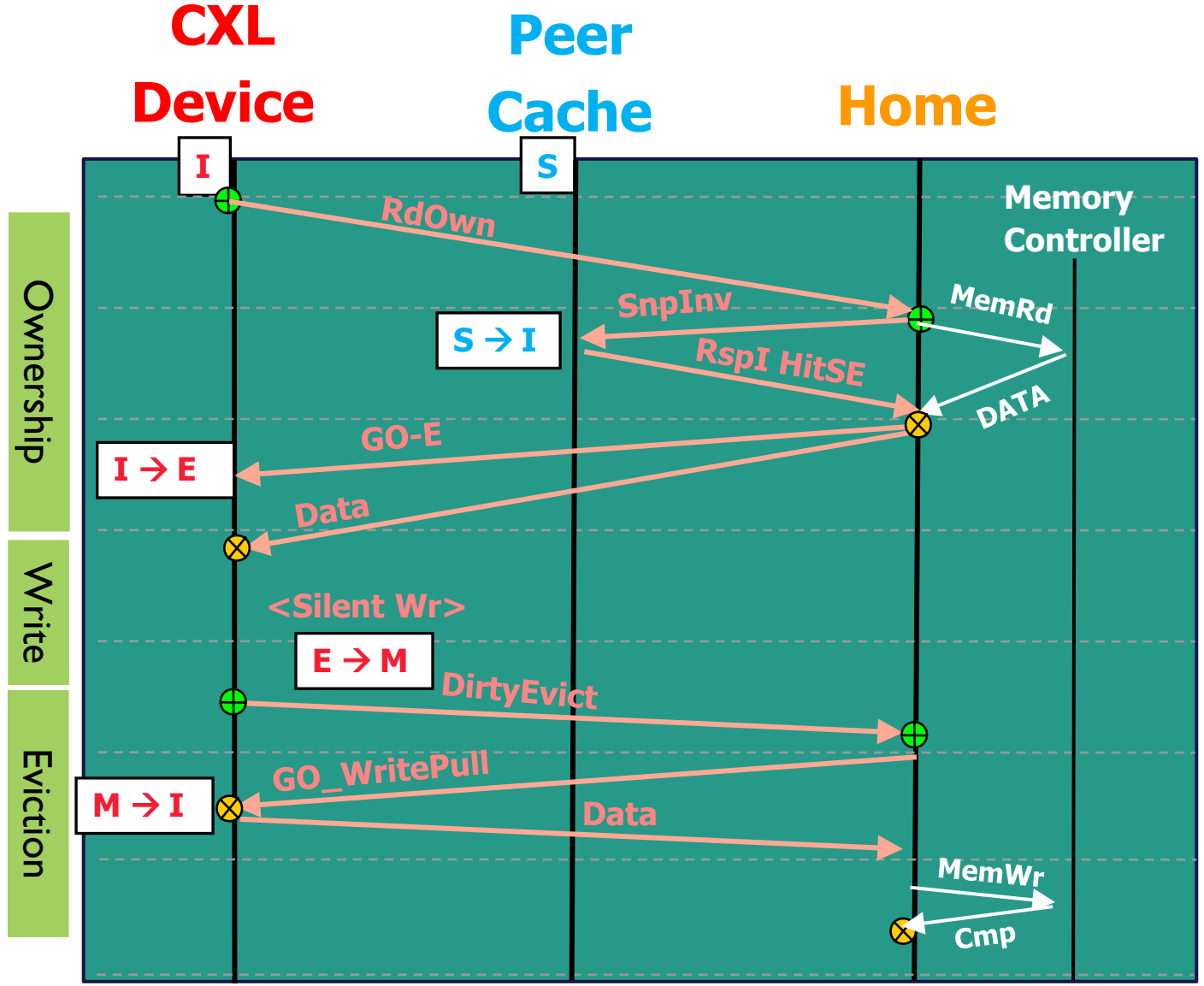
Invalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Example #2: Write

- For Cache Writes there are three phases:
  - Ownership
  - Silent Write
  - **Cache Eviction**



Legend

Cache State:

Modified

Exclusive

Shared

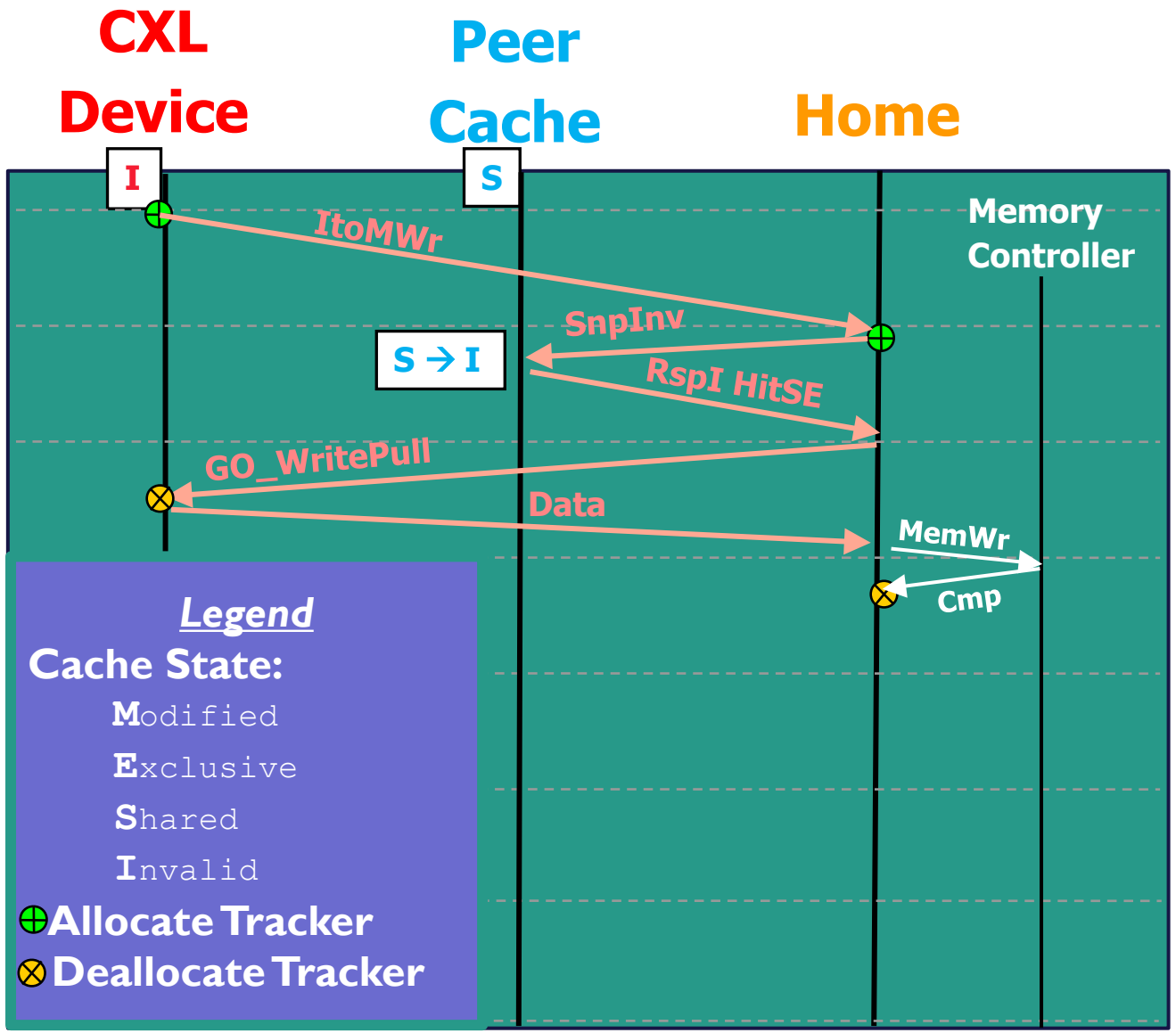
Invalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Example #3: Steaming Write

- Direct Write to Host
  - Ownership + Write in a single flow.
- Rely on completion to indicate ordering
  - May see reduced bandwidth for ordered traffic
- Host may install data into LLC instead of writing to memory



15 Request in CXL

- Reads: RdShared, RdCurr, RdOwn, RdAny
- Read-0: RdownNoData, CLFlush, CacheFlushed
- Writes: DirtyEvict, CleanEvict, CleanEvictNoData
- Streaming Writes: ItoMWr, MemWr, WOWrInv, WrInv(F)

CXL Memory Protocol

Memory Protocol Summary

Simple reads and writes from host to memory

Memory Technology Independent

- HBM, DDR,
- Architected hooks to manage persistence

Includes 2-bits of "meta-state" per cacheline

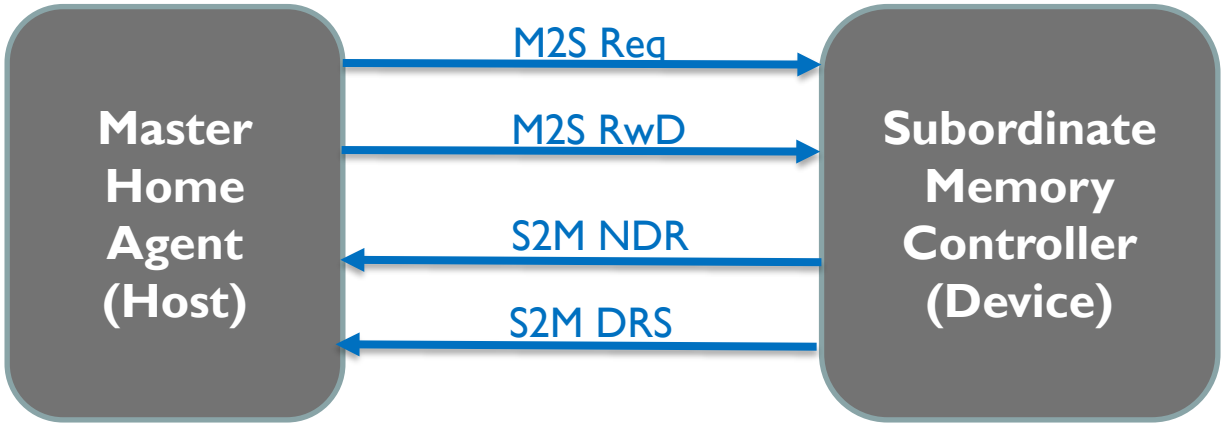
- Memory Only device: Up to host to define usage.
- For Accelerators: Host encodes required cache state.

Memory Protocol Channels

2 channels in each direction

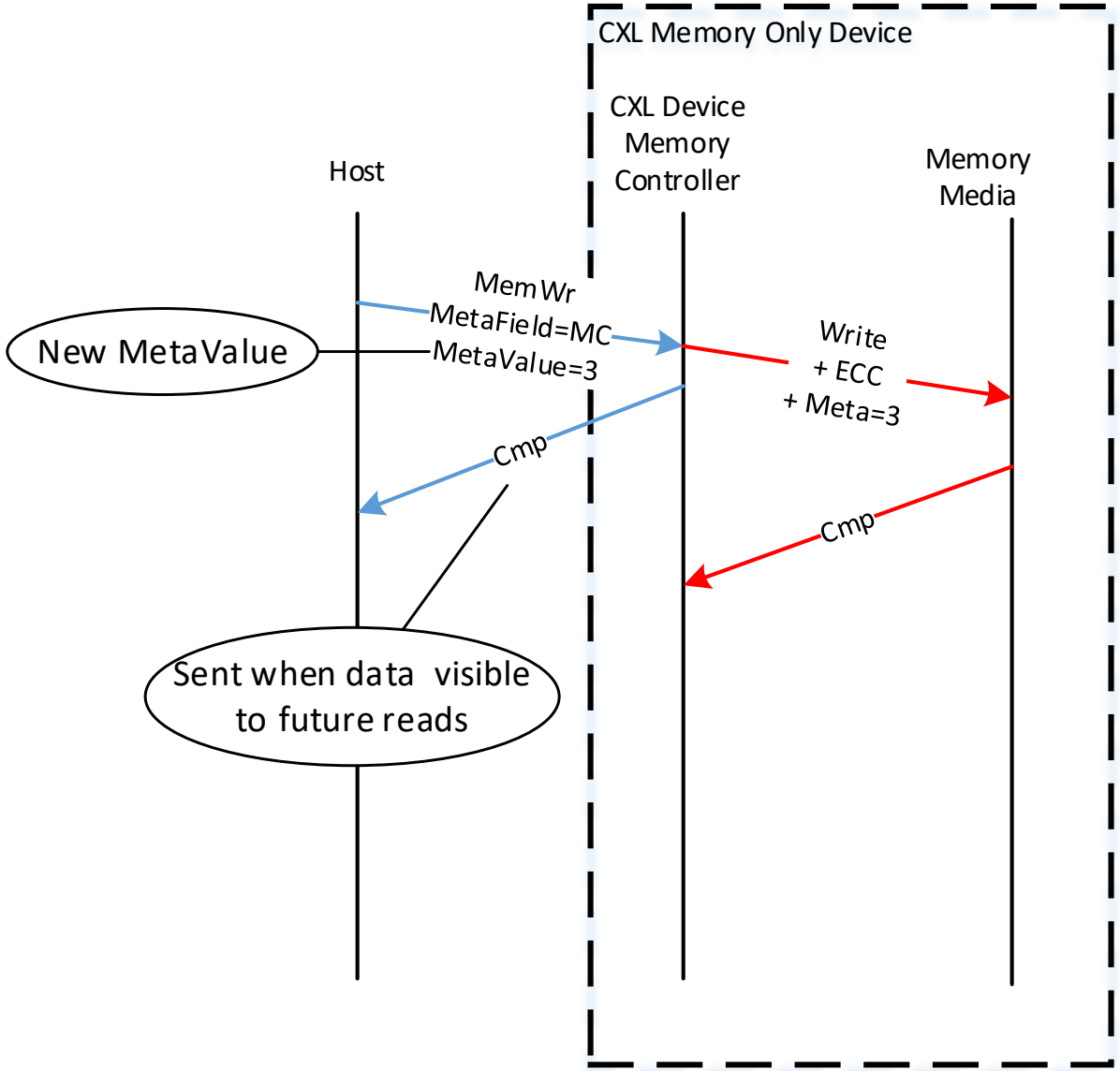
- M2S Request (Req), Request w/ Data (RwD)
- S2M Non-Data Response (NDR), Data Response (DRS) which are pre-allocated.

No Ordering, except special accelerator flow on Req channel



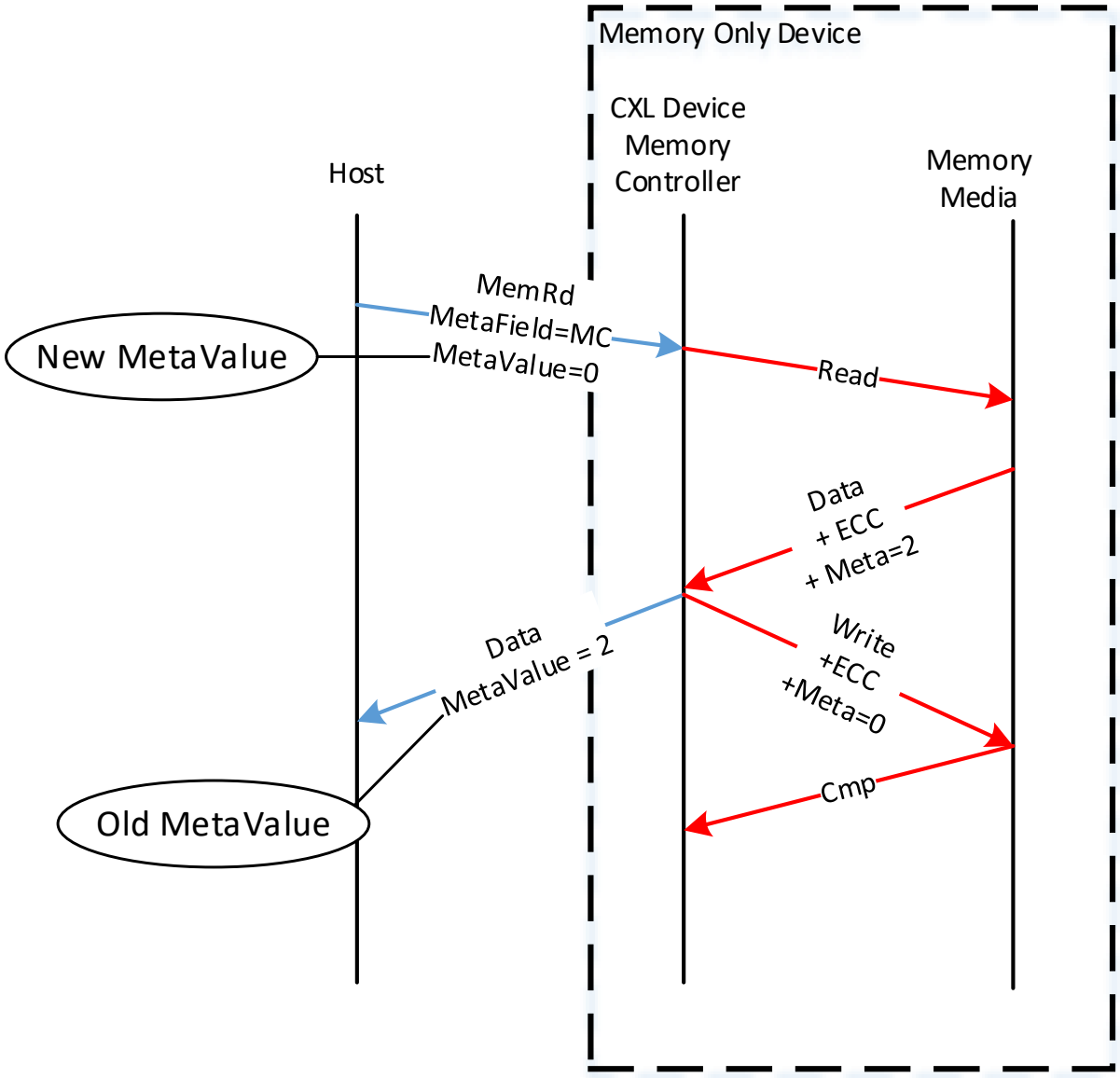
Example #1: Write

ECC handled by device



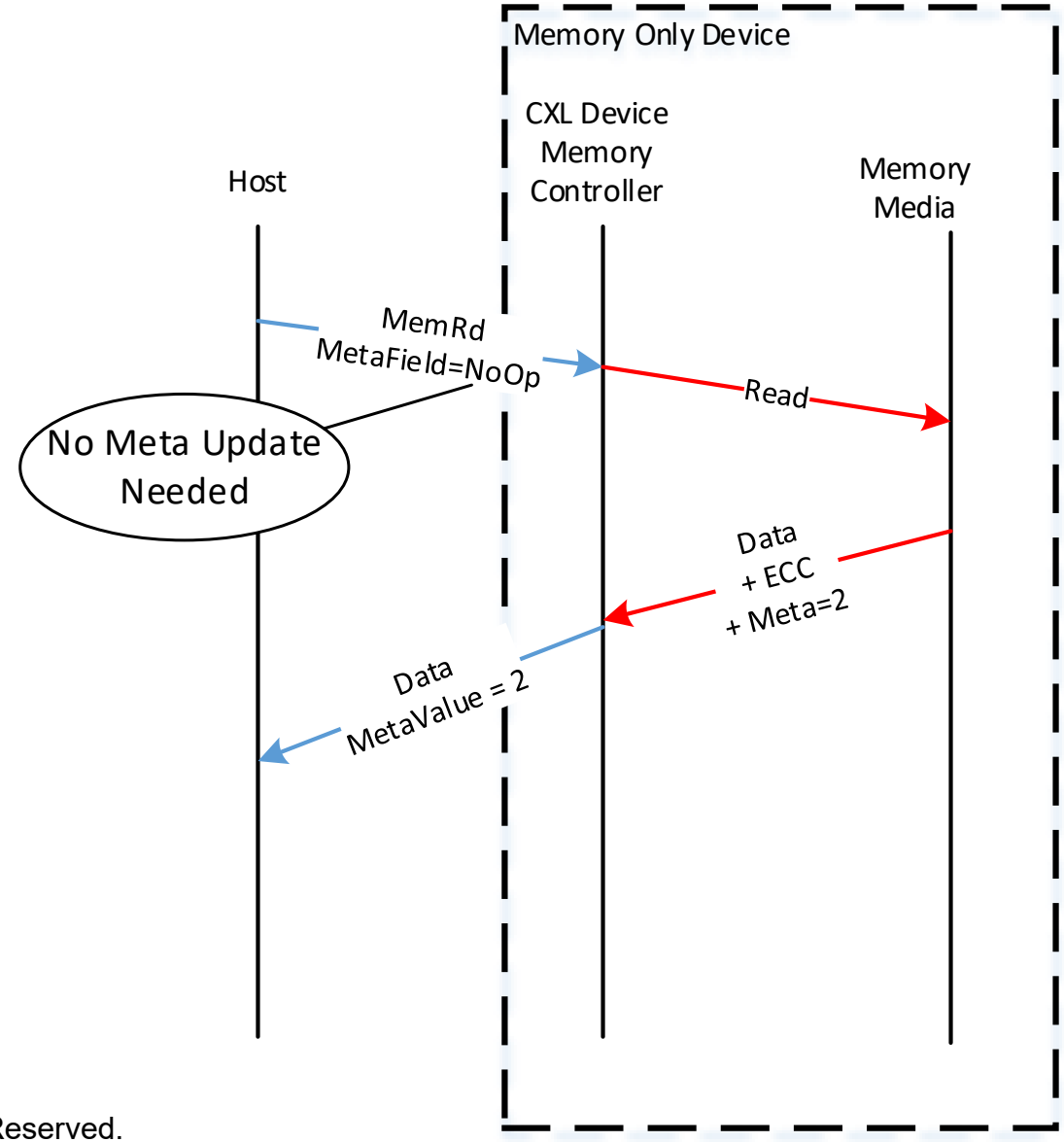
Example #2: Read

Meta Value Change requires device to write.



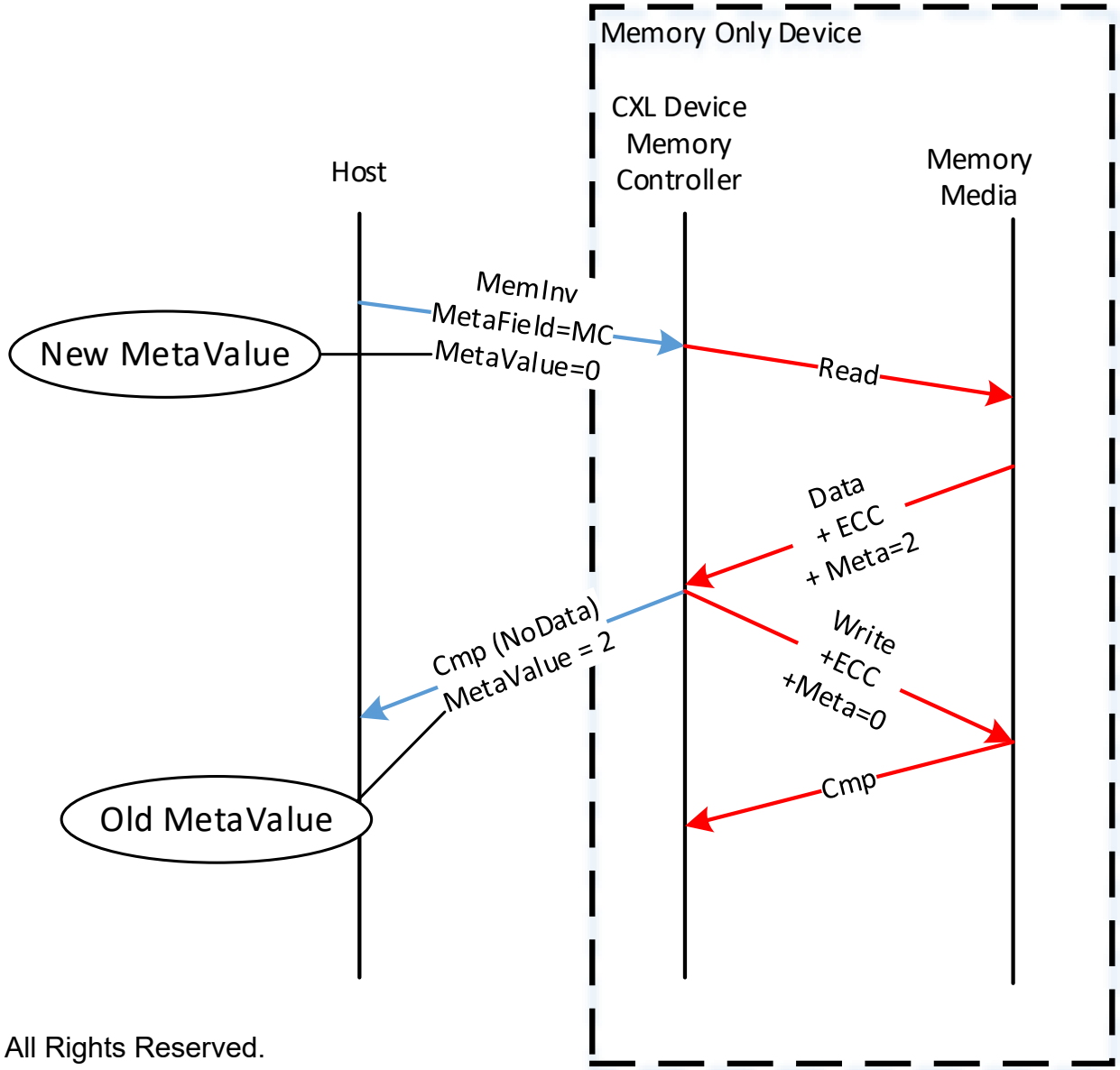
Example #3: Read no Meta

Host may indicate no Meta-state update required on reads



Example #4: MemInv

Used to read/update Meta-state without reading the data itself.



CXL Accelerators using Caching and Memory Protocols

Mixing Protocols for Accelerators

Device memory is coherent, and host manages coherence.

Can device directly read its own "Device-Attached Memory" without violating coherence?

Bias Table

Table holds 1-bit state indicating if host has a cached copy

- Device Bias: No host caching, allowing direct reads
- Host Bias: Host may have a cached copy, so read goes through the host

Host tracks which peer caches have copies

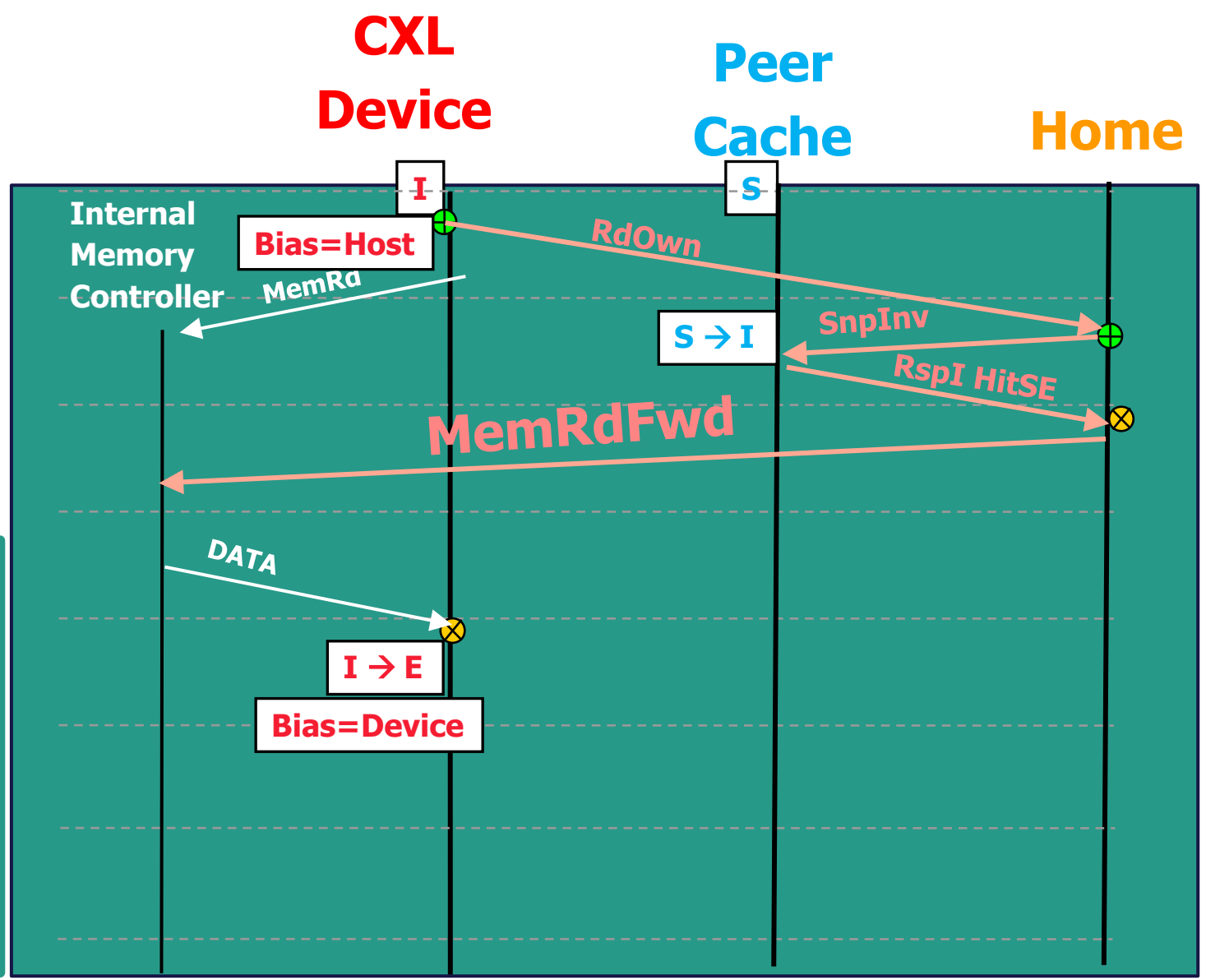
Other Differences

Snoop indication added to Memory Requests → Host does not track device caching of its own "Device-Attached Memory"

Reads/Writes can return "Forward" indication from the host avoiding ping-pong of data

Host Bias Read

MemRdFwd message sent after coherence resolved



Legend

Cache State:

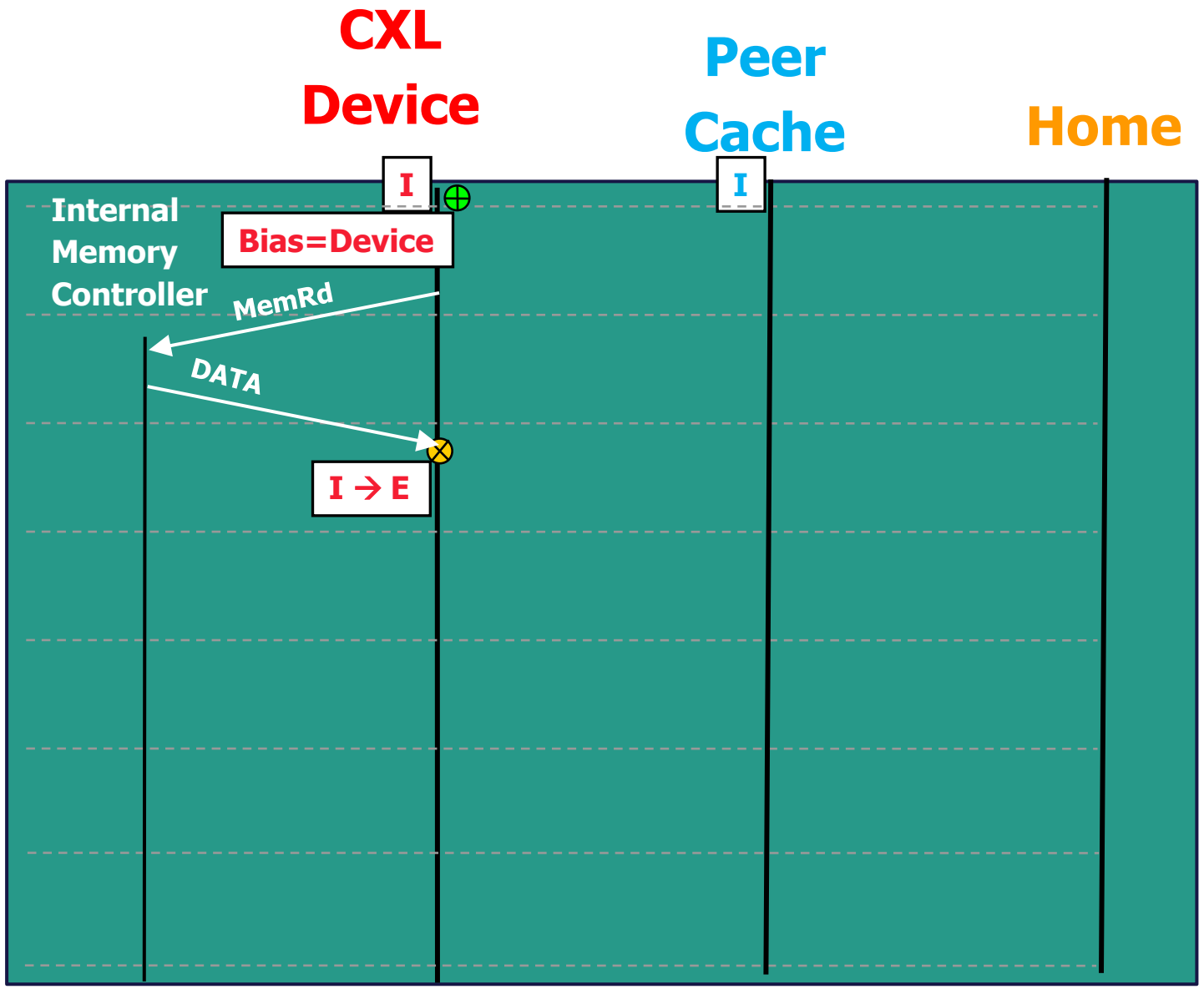
- **M**odified
- **E**xclusive
- **S**hared
- **I**nvalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Device Bias Read

No messages on CXL interface



Legend

Cache State:

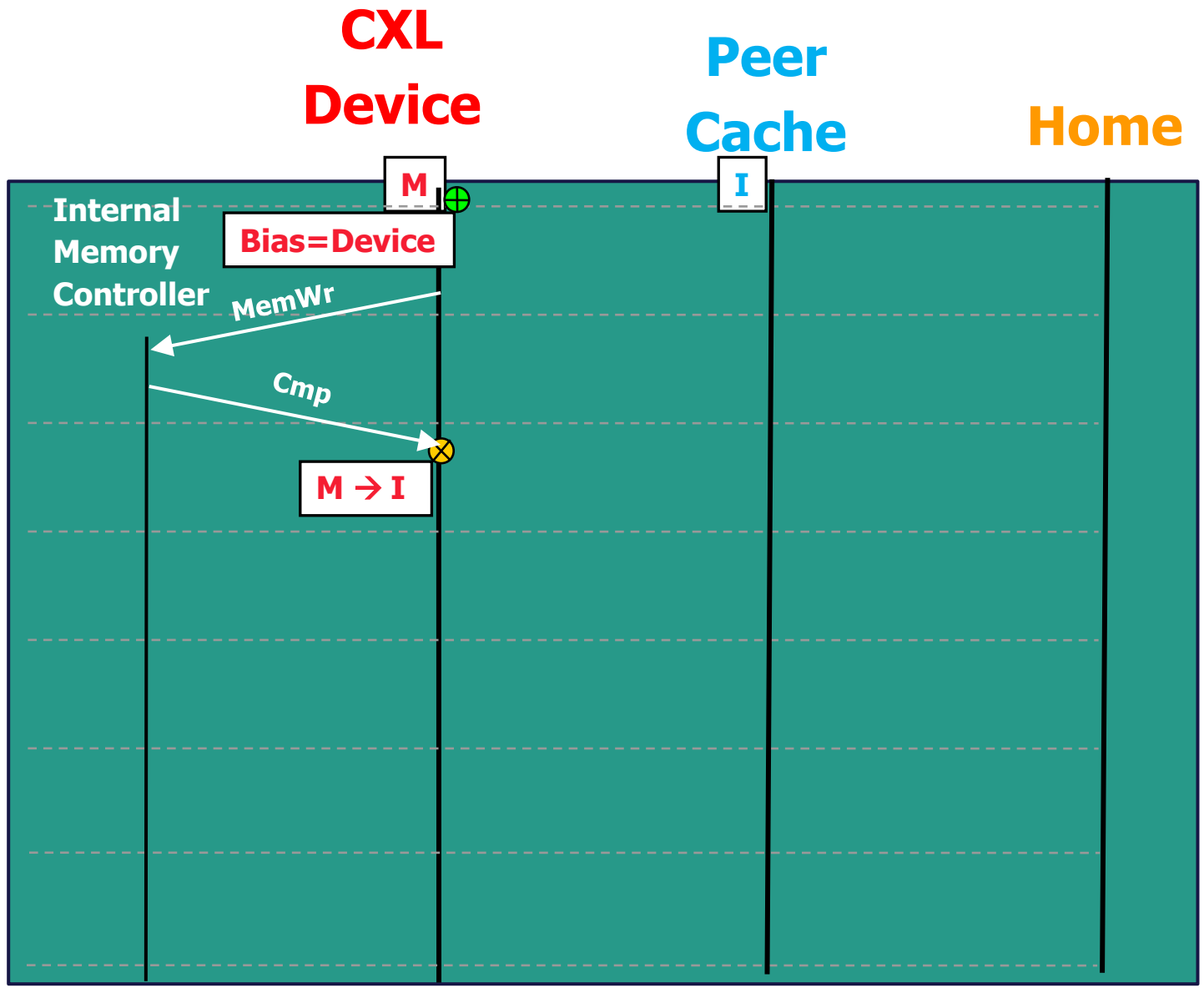
- **M**odified
- **E**xclusive
- **S**hared
- **I**nvalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Device Cache Evictions

E/M in cache imply Bias=Device so no indication to host



Legend

Cache State:

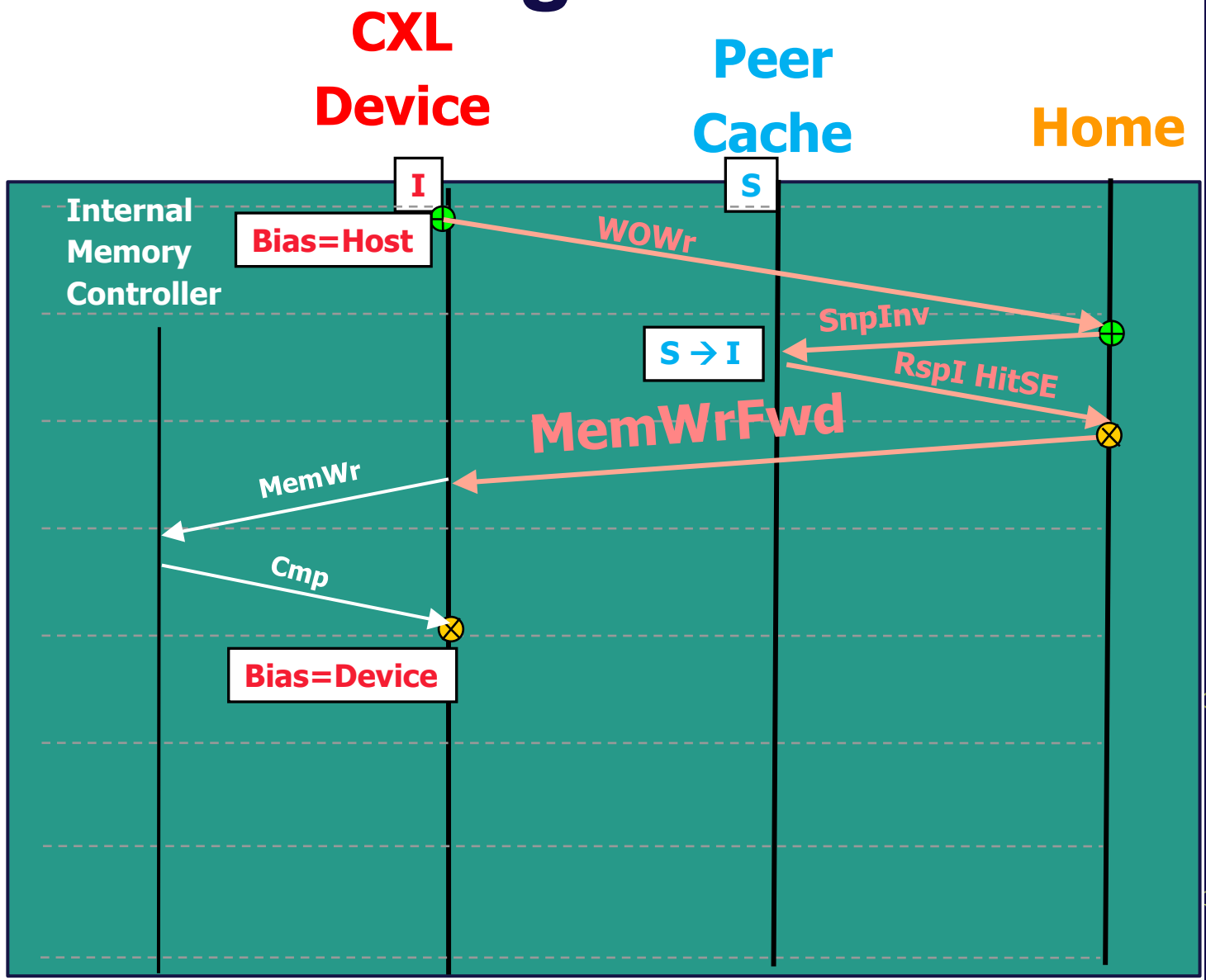
- **M**odified
- **E**xclusive
- **S**hared
- **I**nvalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Host Bias Streaming Write

MemRdFwd message sent after coherence resolved



Legend

Cache State:

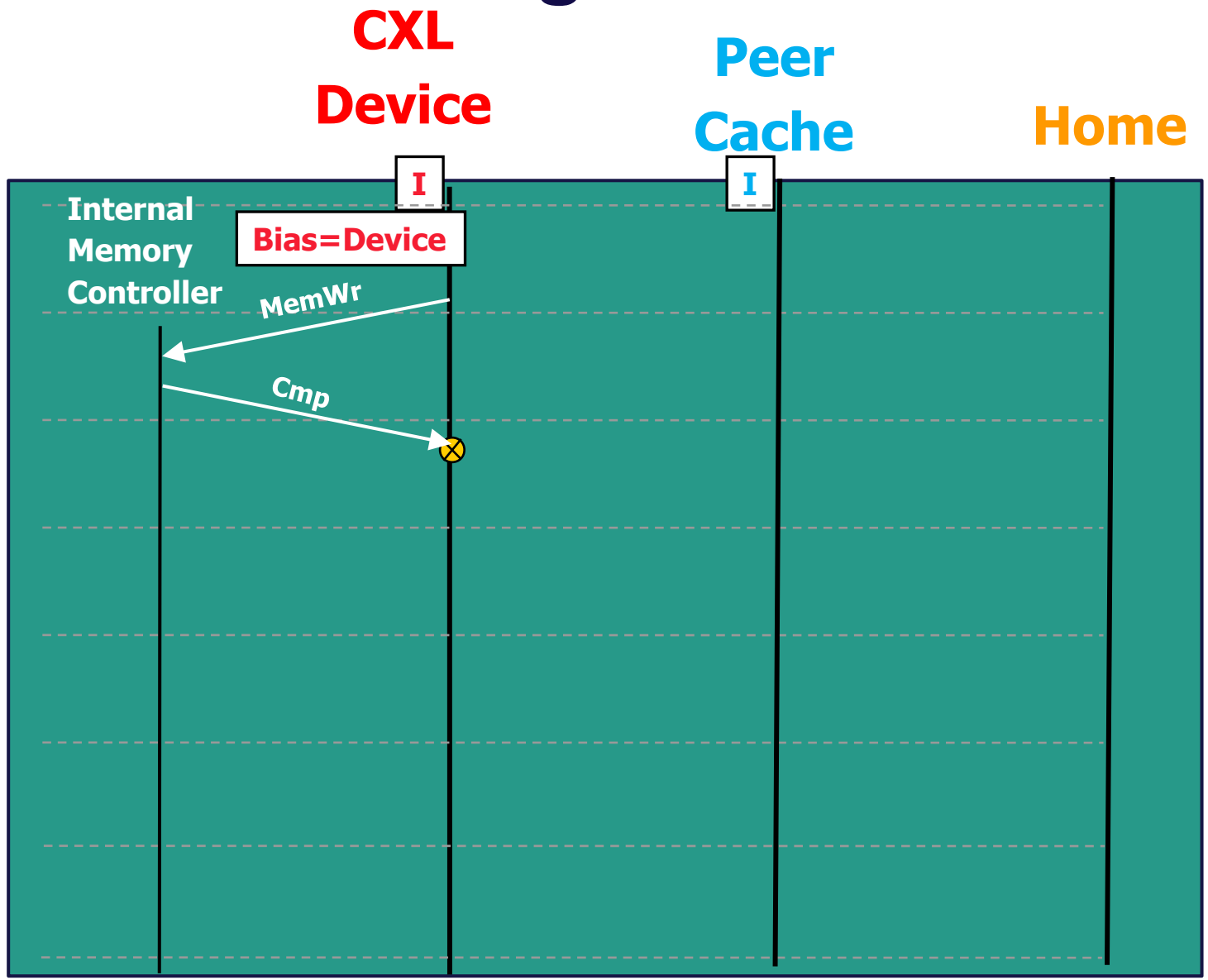
- **M**odified
- **E**xclusive
- **S**hared
- **I**nvalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Device Bias Streaming Write

No message to host



Legend

Cache State:

- **M**odified
- **E**xclusive
- **S**hared
- **I**nvalid

⊕ **Allocate Tracker**

⊗ **Deallocate Tracker**

Thank You

Join Today!

www.computeexpresslink.org/join

Follow Us on Social Media

@ComputeExLink

www.linkedin.com/company/cxl-consortium/

CXL Consortium Channel

Audience Q&A