Pooling and Sharing

Vincent Haché

Director of Systems Architecture, Rambus

Agenda

- Pooling vs. Sharing
- Key Concepts
  - Multi Logical Device (MLD)
  - Multi Headed Device (MHD)
  - Dynamic Capacity Device (DCD)
- Sharing and Back-Invalidate

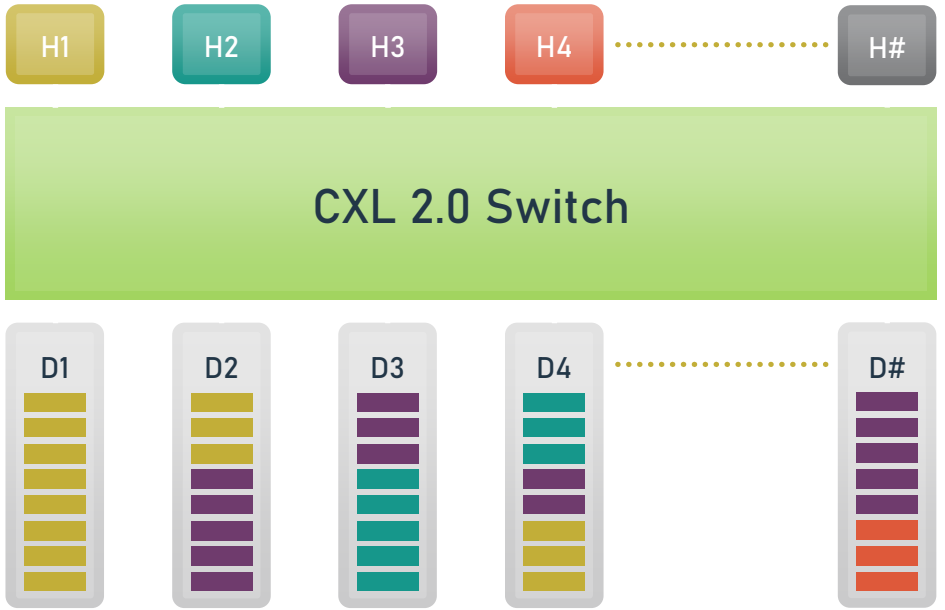
Pooling vs. Sharing

Pooling vs. Sharing

- Pooling: Flexibly assigned pool of media capacity provided by any combination of switches, MLDs, and/or MHDs

- Sharing: Concurrent (or serial) multi-host access to same data provided via DCD framework
  - Advertised as “Shareable” – host is unable to prevent FM from sharing media after use

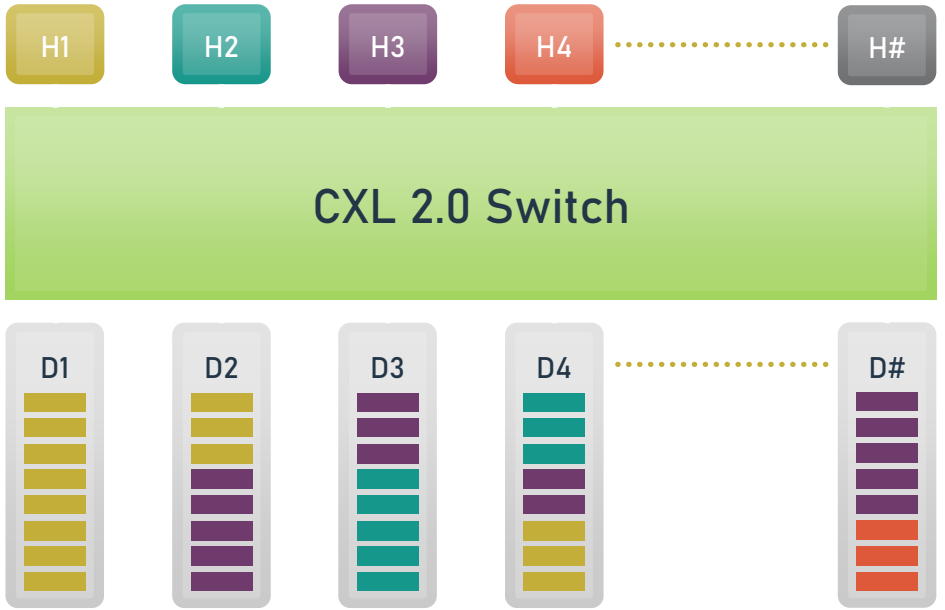
2.0 – MLD and switch enabled pooling



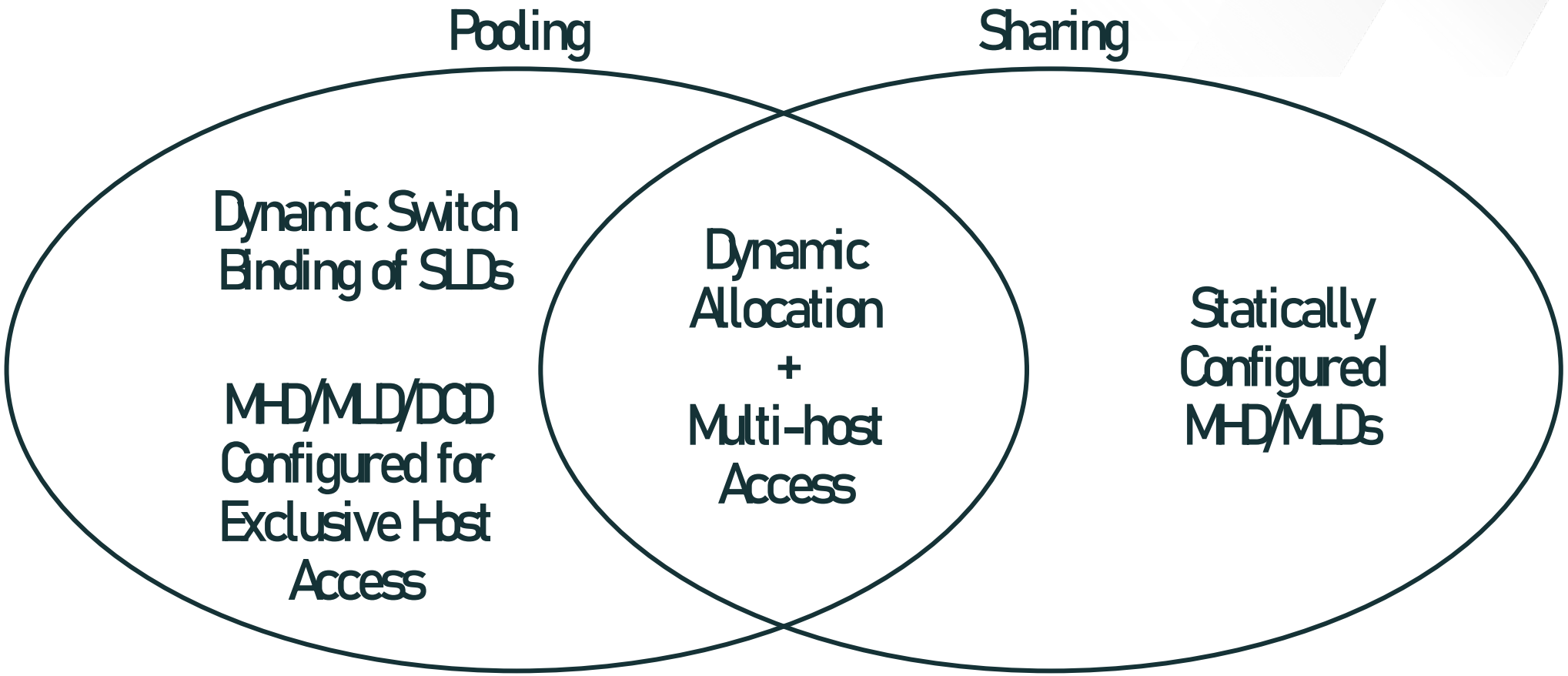
Why Pooling or Sharing?

- Pooling:
  - Provides more efficient utilization of memory resources at scale by allowing for dynamic allocation
  - Example use cases:
    - Per SLAs when VMs spin up / spin down
    - To provide required memory for peak workload utilization
- Sharing:
  - Reduces aggregate memory requirements by providing multi-host access to the same data
  - Efficient data movement
- Pooling & Sharing may exist together or separately

2.0 – MLD and switch enabled pooling



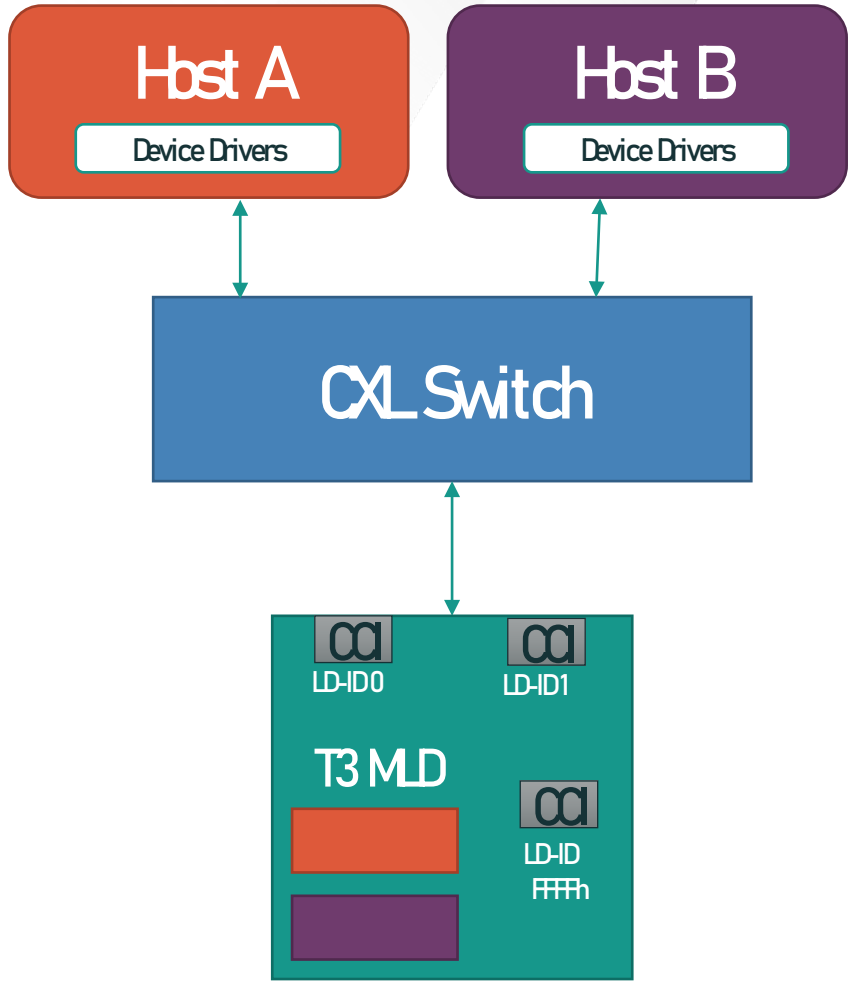
Pooling vs. Sharing



Key Concepts

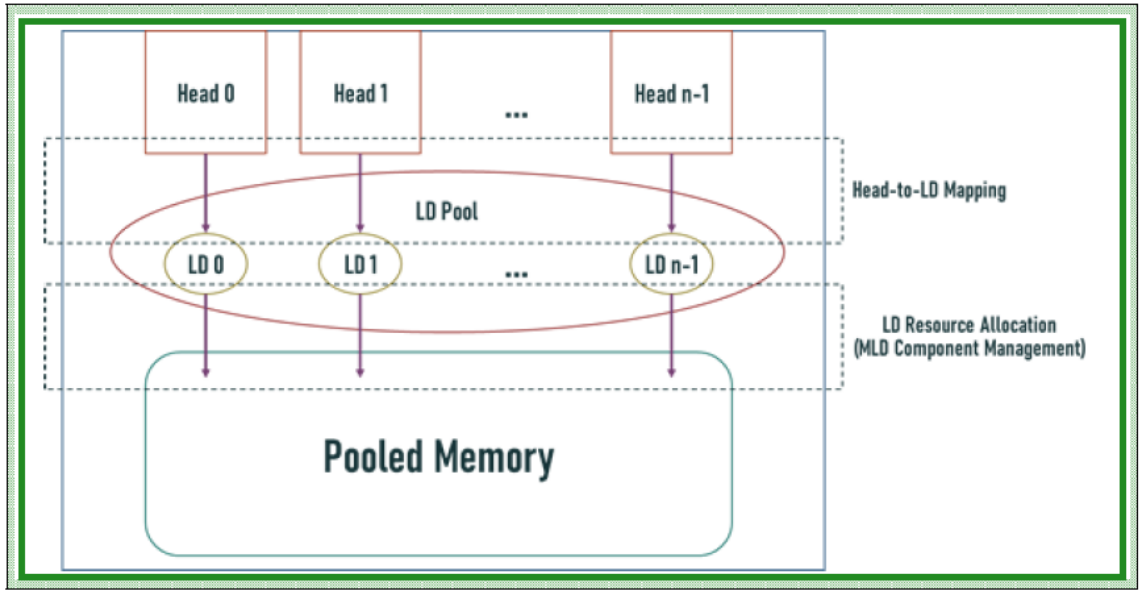
Multi-Logical Device (MLD)

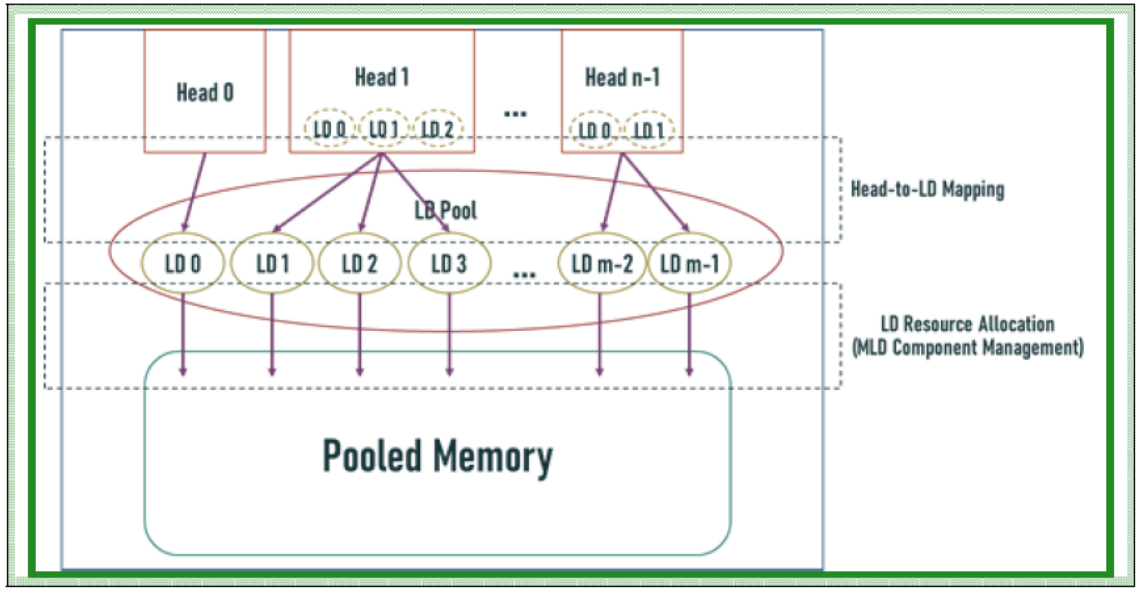
- Type 3 device with single CXL link
- Transactions across link carry 'LD-ID' to identify traffic to/from each host
- Requires a switch
  - Applies 'LD-ID' to traffic to device
  - Routes traffic from device to host based on 'LD-ID'



Multi-Headed Device (MHD)

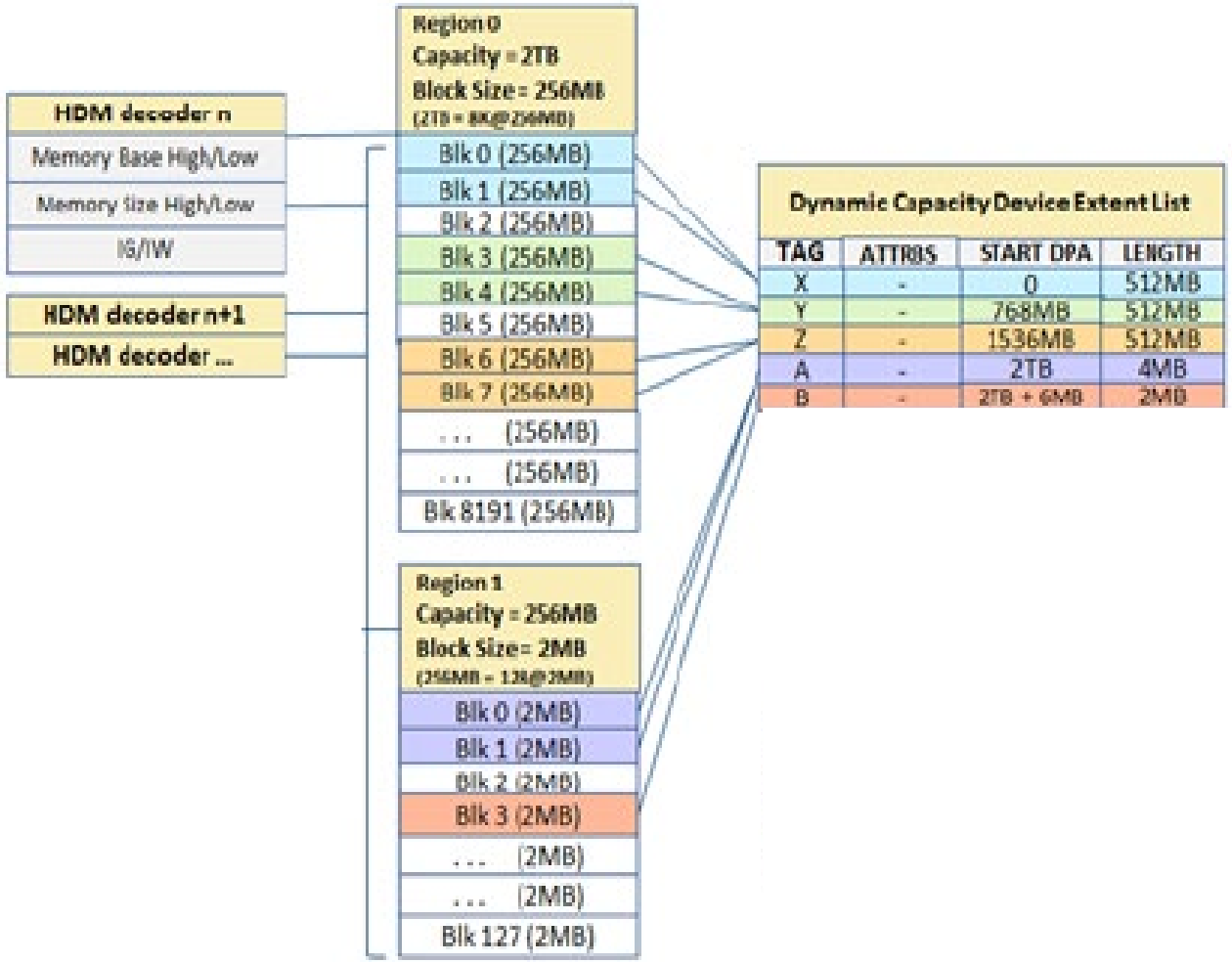
- Type 3 device with multiple host links (heads)
- Manages memory-to-LD allocation with differing head behavior
  - MH-SLD presents a 1-1 mapping of a single LD to each head
  - MH-MLD presents additional composability with up to 16 LDs mapped to each head





Dynamic Capacity Device (DCD)

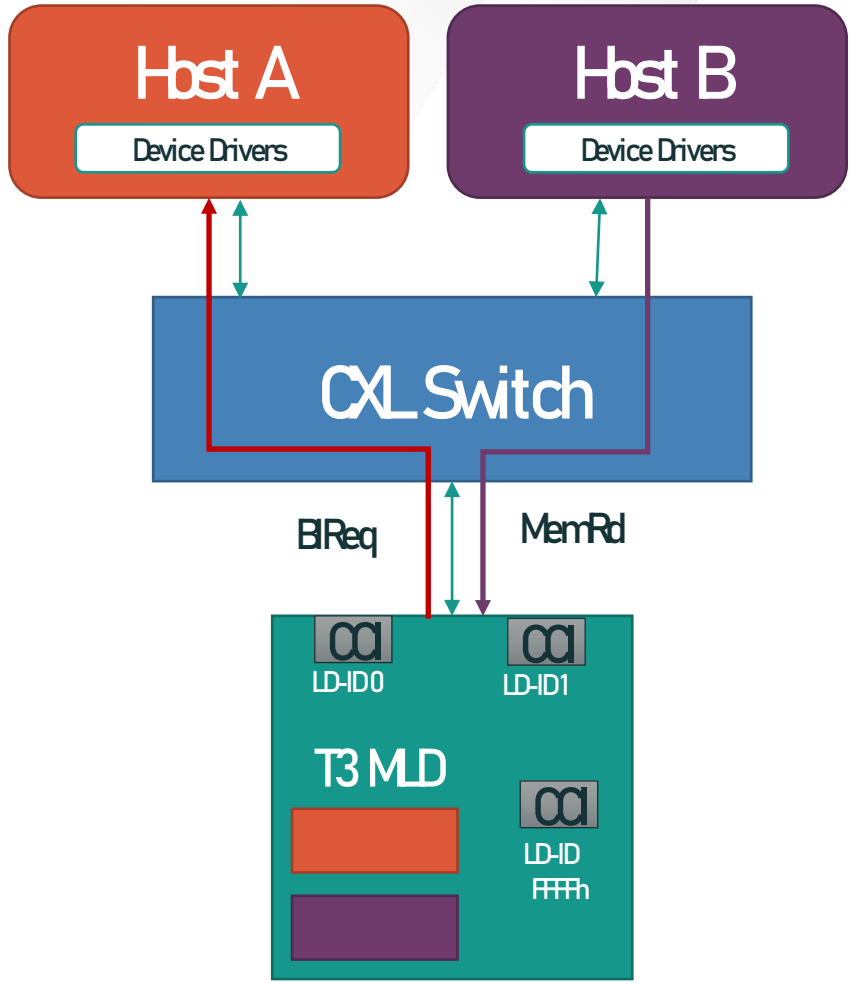
- Allows memory capacity to change dynamically without reprogramming HDM decoders
- DCD presents its maximum capacity to each host
  - HDM decoders programmed for entire DPA range
  - DCD command set used to discover the actual memory allocation
- Fabric Manager (FM) uses the DCD command set to query and configure the DCD
- The DPA is divided into 1-8 separate regions, and each region is subdivided by the DCD into fixed-size blocks



Sharing and Back-Invalidate

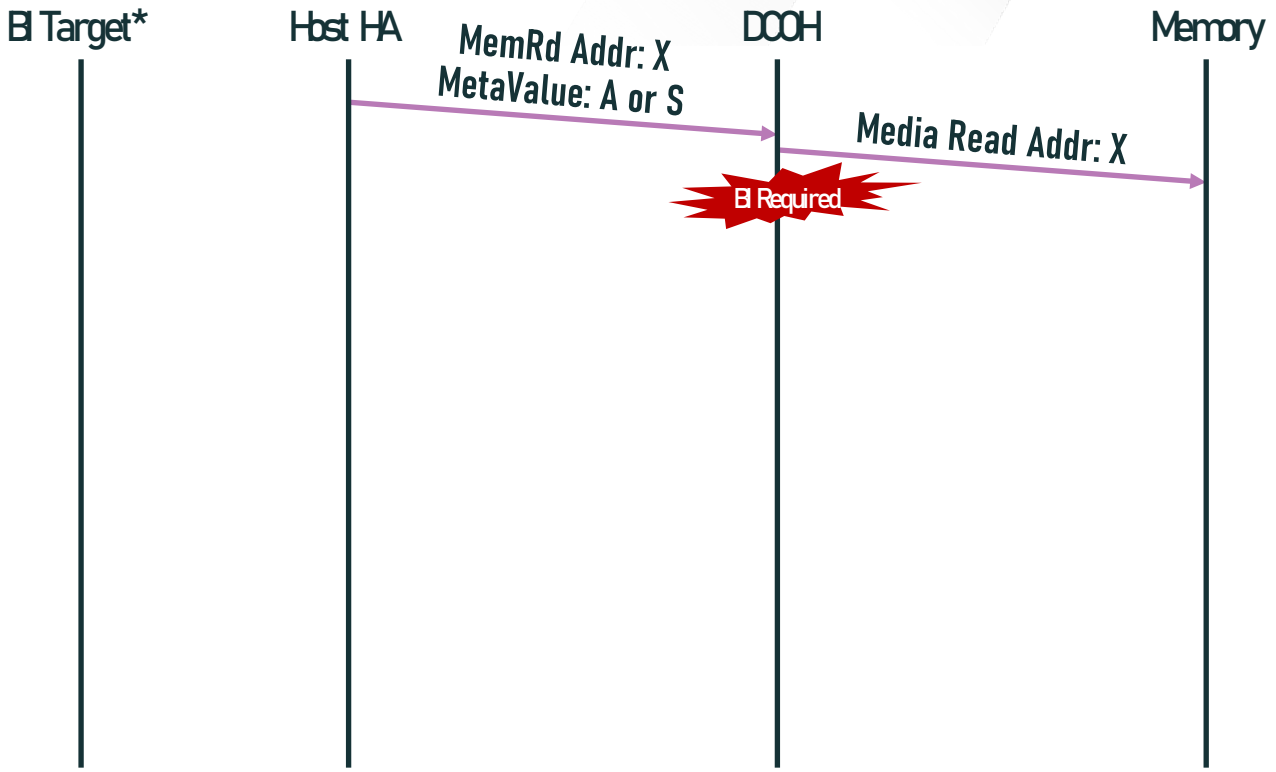
Back-Invalidate

- Provides cache coherent sharing of data
  - Host is notified to invalidate cache or write back modified data
- Allows multiple devices to work on same cacheline
  - P2P UIO transactions from T1/T2 devices
  - Multi-host sharing for T3 devices
- Simplifies sharing and exchange of data/control structures
  - Replaces data copies and doorbells/interrupts



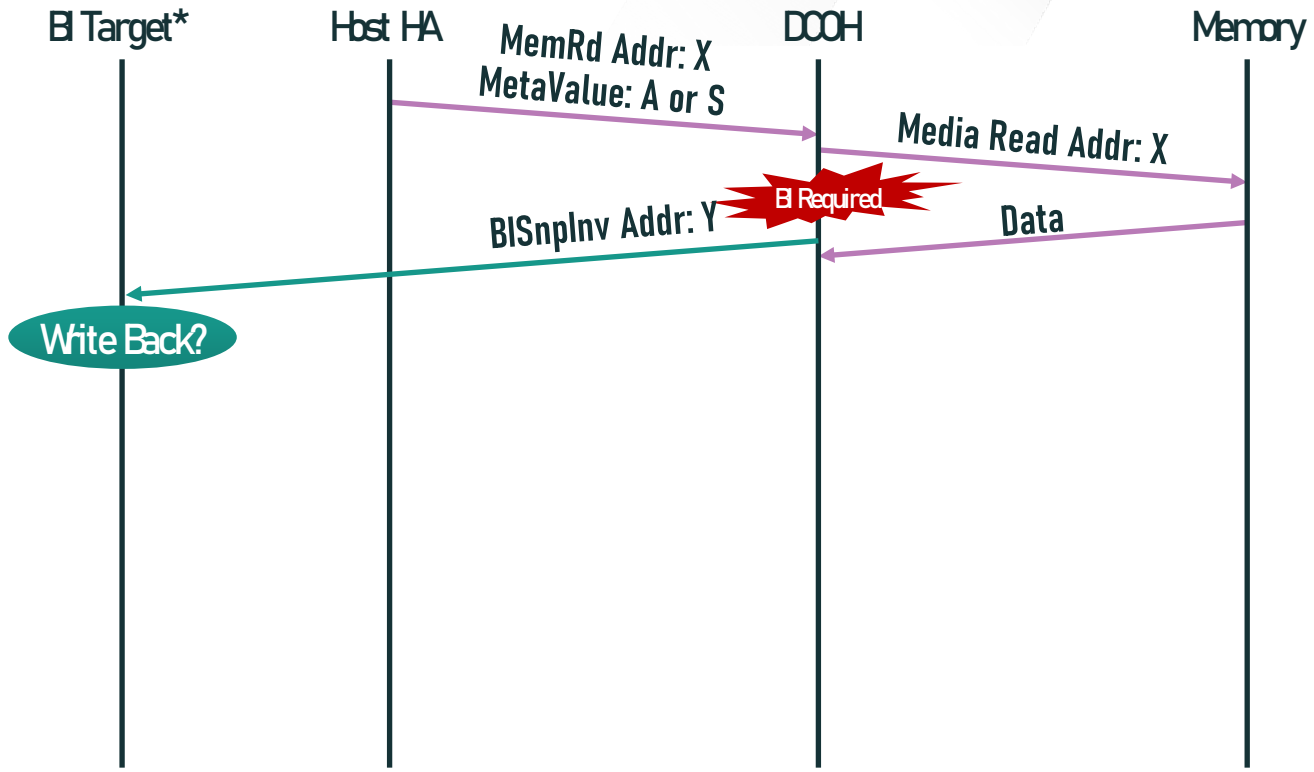
Back-Invalidate

- Host reads a cacheline
  - MetaValue indicates host will cache data
- Device determines BI is required:
  - Device is tracking line already, or
  - Device Snoop Filter is full
- Media Read runs in parallel with BI



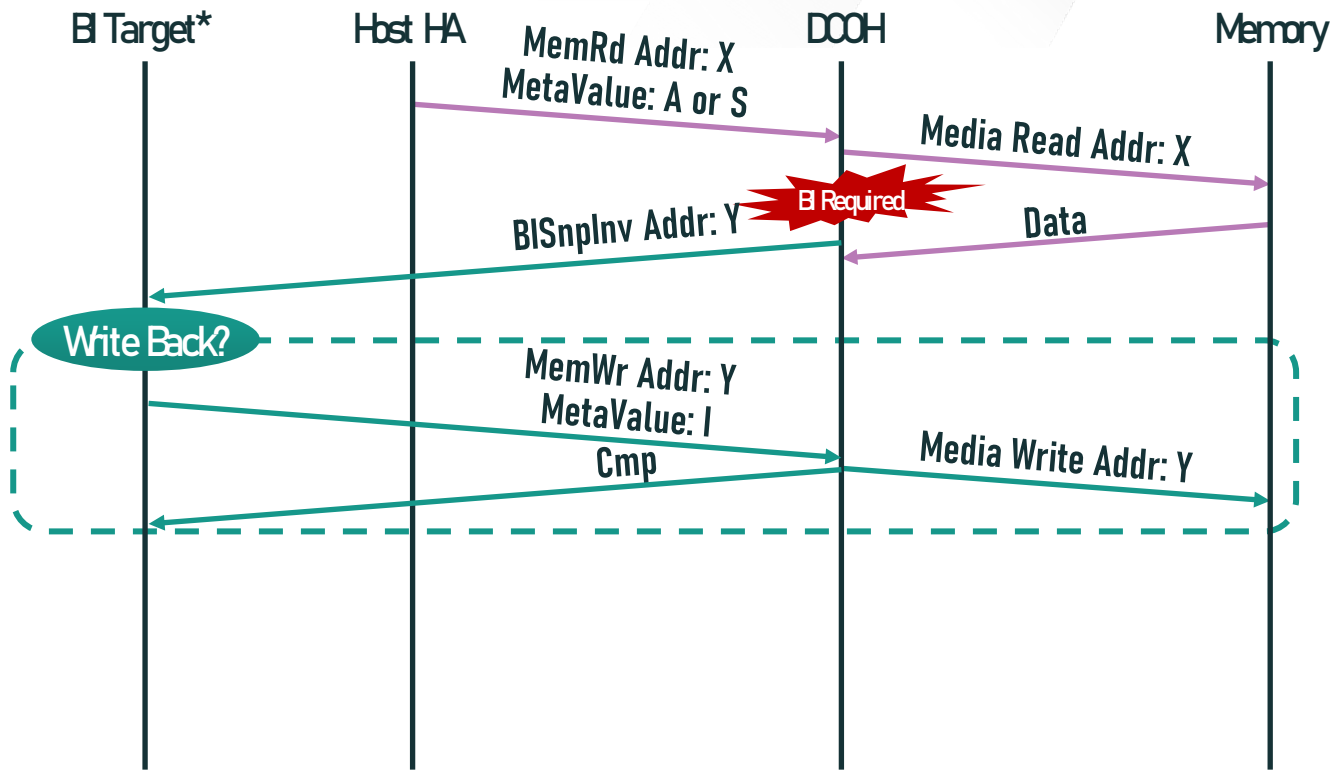
Back-Invalidate

- Device issues BISnpInv to BI Target
  - * - Could be same host in case of full Snoop Filter
- BI Target checks whether write back is required
  - Is cacheline modified locally?



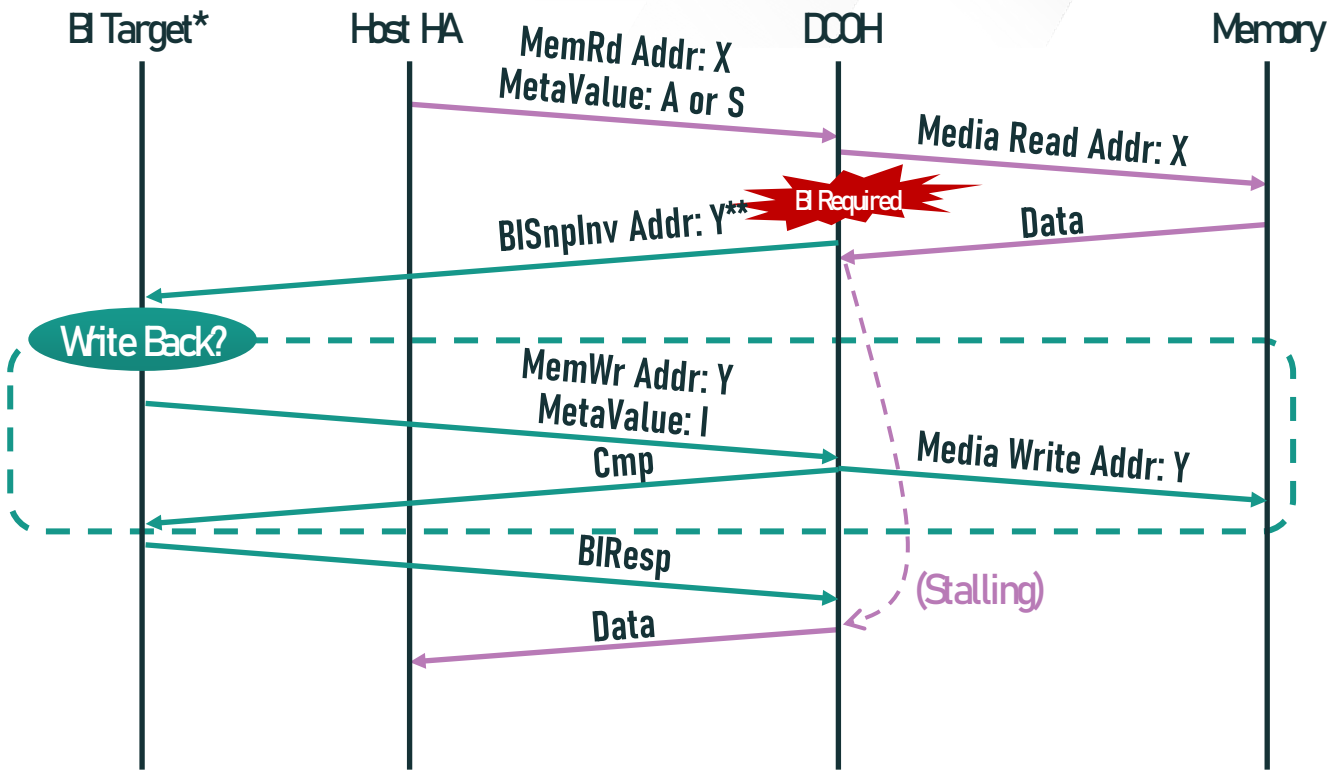
Back-Invalidate

- If required, BI Target issues MemWr of modified data
- Device has response data for Address X, but must wait for BIResp
- Media write runs in parallel



Back-Invalidate

- BI Target sends BI response after write back completes
- Device can complete read of Address X, which has been stalled awaiting BI exchange



Thank You