CXL HDM-DB Support in Linux

Linux Plumbers Conference

Davidlohr Bueso. Tokyo, Japan. December 2025.

Introduction

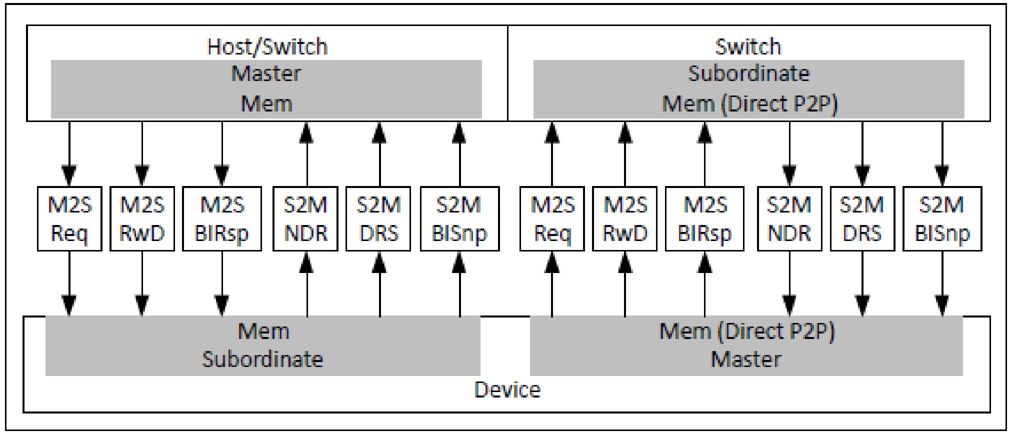
- Flavors of CXL Host-managed Device Memory (HDM): HDM-H, HDM-D, **HDM-DB**.

- In HDM-DB, the Device takes the responsibility of tracking ownership of its lines.

- Back-Invalidate (BI) allows direct snooping by the device to the host/peer using two dedicated channels BISnp and BIRsp.
  - Allow devices to manage coherence by using an inclusive snoop filter, tracking coherence for individual cachelines that may block new M2S Requests until BISnp messages are processed by the host.
  - CXL.mem for deadlock avoidance.

- Lines on the device may be changed by
  - Accelerators.
  - Near-data Processing (NDP).
  - Other hosts.

- Introduced in CXL 3.0+, 256b FLIT (PCIe 6.0+),



Host/Switch
Master
Mem

Switch
Subordinate
Mem (Direct P2P)

M2S Req
M2S RwD
M2S BIRsp
S2M NDR
S2M DRS
S2M BISnp
M2S Req
M2S RwD
M2S BIRsp
S2M NDR
S2M DRS
S2M BISnp

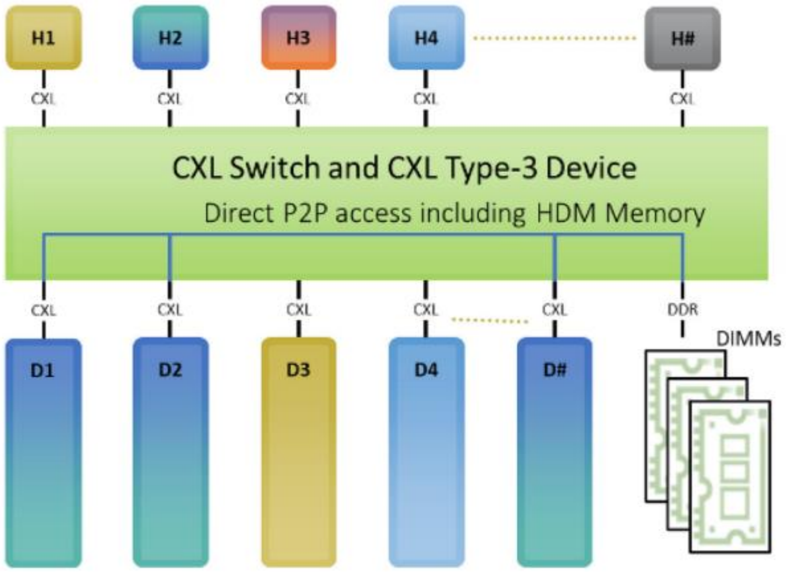
Mem
Subordinate

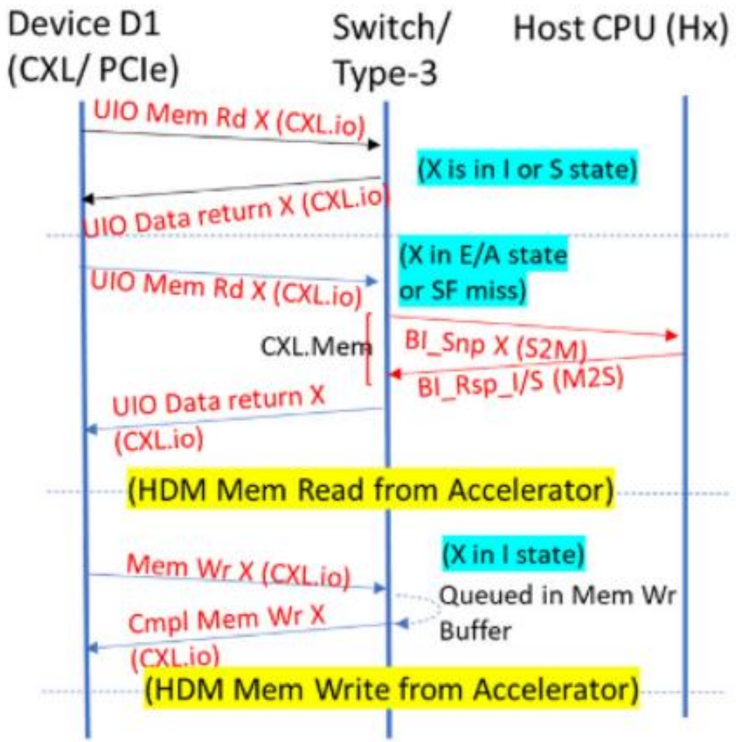
Mem (Direct P2P)
Master

Device

Use Case: Direct Peer-to-Peer to HDM

- PCIe Unordered IO (UIO) + BI facilitates direct P2P access from CXL or PCIe devices to HDM-DB memory.
  - CXL 1.1/2.0 must always go through the host CPU to resolve coherence for HDM accesses.


H1
H2
H3
H4
H#
CXL
CXL
CXL
CXL
CXL
CXL Switch and CXL Type-3 Device
Direct P2P access including HDM Memory
CXL
CXL
CXL
CXL
CXL
DDR
DIMMs
D1
D2
D3
D4
D#



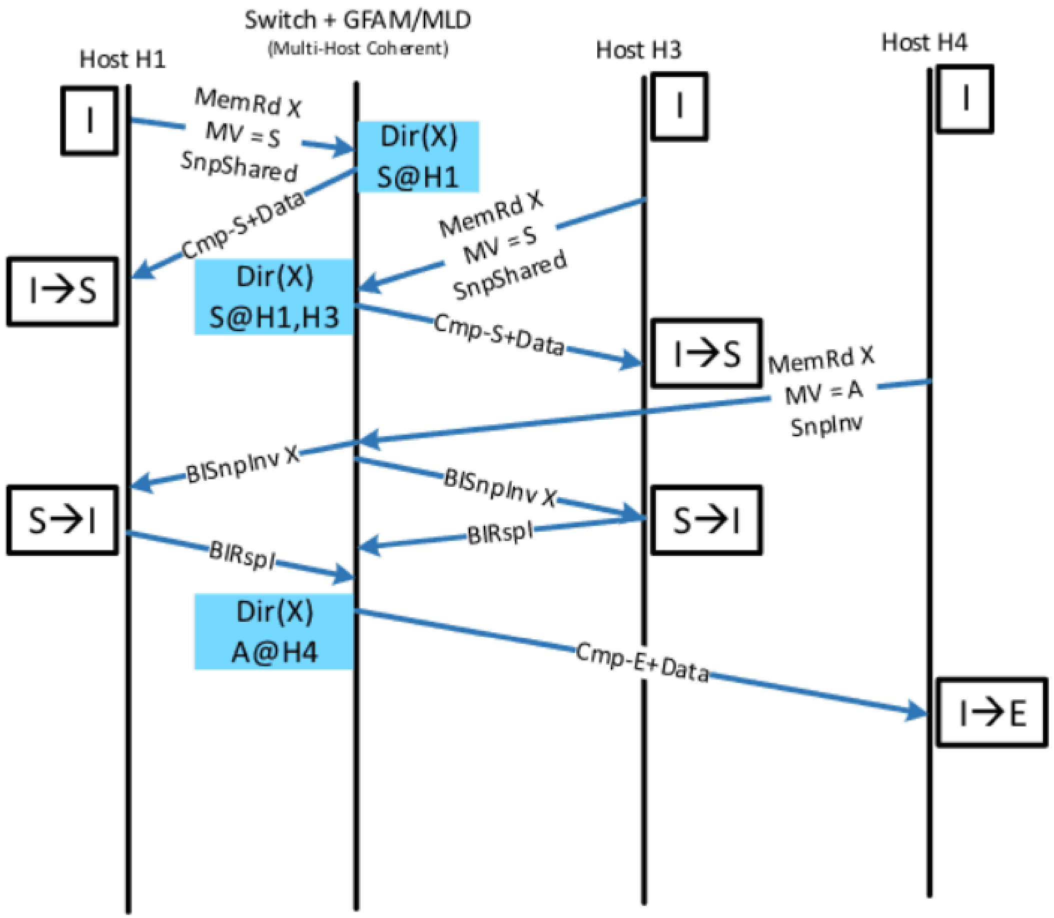
Use Case: Distributed/shared coherent memory

1. Host 1 gets ownership of cache line X as a shared copy, it updates the directory from “I” state to “S”.

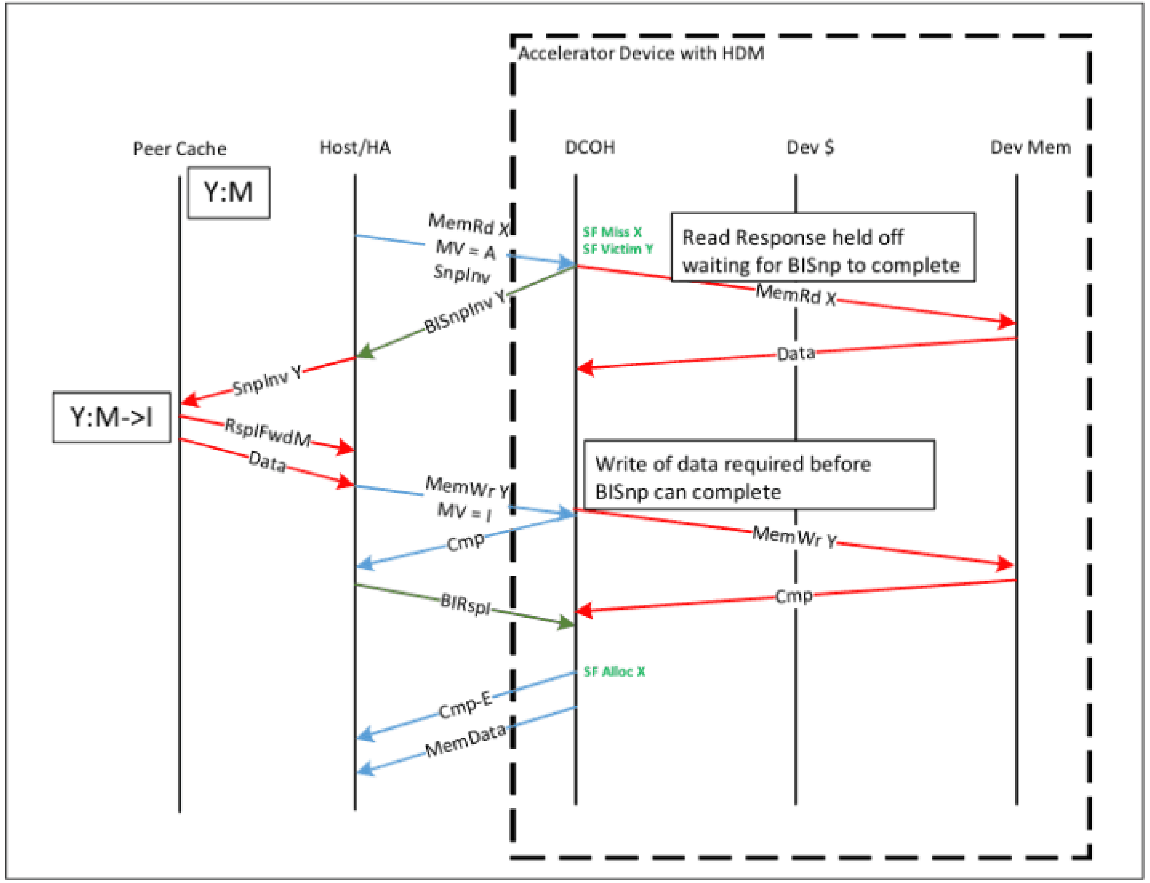
2. Host 3 asks for the same line X in shared state, it provides the data and updates the directory for X to indicate that both H1 and H3 have it Shared.

3. Host 4 request exclusive copy, it issues a BI to both H1 and H3 and waits for the response.

4. Device updates its directory to mark X as “E” by H4, prior to sending the data and ownership to H4.



Use Case: Larger mappings by Accelerators



- ○ Enable Type2 devices to have a snoop filter instead of having a full directory and invoking the existing bias-flows.
- ○ A major constraint with using the CXL.cache D2H Request channel for coherence management is that it can be blocked waiting on the forward progress of the M2S Request channel. This architectural constraint **disallows** the implementation of an inclusive SF.

Events triggering BISnp

- Device-initiated coherence for HDM.

- Snoop Filter capacity miss/overflow (SF victim).

- Memory Operations related to RAS. Sanitize/Zero as well as Maintenance commands for PPR and Memory Sparing will cause BISnp to be generated implied as a consequence of the action.

- TEE state changes: When a TEE state changes for a memory region, BISnp is used to snoop back all affected addresses before the memory contents or TEE state is updated.

The Linux part

Current State (qemu)

- [PATCH v4 -qemu 0/5] hw/cxl: Support Back-Invalidate

```
qemu-system-x86_64 -M q35,cxl=on -m 4G,maxmem=8G,slots=8 -smp 4
...
-object memory-backend-ram,id=cxl-mem0,share=on,size=256M
-object memory-backend-ram,id=cxl-mem1,share=on,size=256M
-object memory-backend-ram,id=cxl-mem2,share=on,size=256M
-object memory-backend-ram,id=cxl-mem3,share=on,size=256M
-device pxb-cxl,bus_nr=12,bus=pcie.0,id=cxl.1
-device cxl-rp,port=0,bus=cxl.1,id=root_port0,chassis=0,slot=0
-device cxl-rp,port=1,bus=cxl.1,id=root_port1,chassis=0,slot=1
-device cxl-upstream,bus=root_port0,id=us0,x-256b-flit=on
-device cxl-downstream,port=0,bus=us0,id=swport0,chassis=0,slot=4
-device cxl-type3,bus=swport0,volatile-memdev=cxl-mem0,id=cxl-mem0,sn=0x1,x-256b-flit=on,hdm-db=on
-device cxl-downstream,port=1,bus=us0,id=swport1,chassis=0,slot=5
-device cxl-type3,bus=swport1,volatile-memdev=cxl-mem1,id=cxl-mem1,sn=0x2,x-256b-flit=on,hdm-db=on
-device cxl-downstream,port=2,bus=us0,id=swport2,chassis=0,slot=6
-device cxl-type3,bus=swport2,volatile-memdev=cxl-mem2,id=cxl-mem2,sn=0x3
-device cxl-downstream,port=3,bus=us0,id=swport3,chassis=0,slot=7
-device cxl-type3,bus=swport3,volatile-memdev=cxl-mem3,id=cxl-mem3,sn=0x4
-M cxl-fmw.0.targets.0=cxl.1,cxl-fmw.0.size=4G,cxl-fmw.0.interleave-granularity=4k
```

Current State (kernel)

- [PATCH RFC 0/3] cxl: Initial support for Back-Invalidate

- Discovery/probe

```
int cxl_bi_setup(struct cxl_dev_state *cxlds);
```

- Adding BI support to the kernel via type3 devices is probably the most straightforward approach.

- All possible CXL HDM coherence models can be easily detected by ways of the device state.

```
type2 hdm-db: cxlds->type == CXL_DEVTYPE_DEVMEM && cxlds->bi == true
type2 hdm-d:  cxlds->type == CXL_DEVTYPE_DEVMEM && cxlds->bi == false
type3 hdm-h:  cxlds->type == CXL_DEVTYPE_CLASSMEM && cxlds->bi == false
type3 hdm-db: cxlds->type == CXL_DEVTYPE_CLASSMEM && cxlds->bi == true
```

Current State (kernel)

- [PATCH RFC 0/3] cxl: Initial support for Back-Invalidate

- Discovery:
  - Support across full topology.
  - Consume BI Decoder and Routing Table cachemem registers.
  - Does not modify HDM decoder BI bit – left for region provisioning.

- Deallocating BI-ID requires the device be offline/unmapped. Ie: `cxl_detach_ep()`

```
int cxl_bi_dealloc(struct cxl_dev_state *cxlds);
```

Current State (kernel)

- Type3 device can use HDM-H and HDM-DB for different ranges of its memory.

- Type3 region provisioning requires configuring the endpoint decoder and the adhoc region in the root decoder.

```
# cxl list -D
[
  {
    "decoder":"decoder0.0",
    …
    "volatile_capable":true,
    "accelmem_capable":true,
  }
]
# echo ram > /sys/bus/cxl/devices/decoder2.0/mode
# echo 1 > /sys/bus/cxl/devices/decoder2.0/bi
# echo 0x20000000 > /sys/bus/cxl/devices/decoder2.0/dpa_size
# echo region0 > /sys/bus/cxl/devices/decoder0.0/create_ram_bi_region
# echo 256 > /sys/bus/cxl/devices/region0/interleave_granularity
# echo 1 > /sys/bus/cxl/devices/region0/interleave_ways
# echo 0x20000000 > /sys/bus/cxl/devices/region0/size
# echo decoder2.0 > /sys/bus/cxl/devices/region0/target0
# echo 1 > /sys/bus/cxl/devices/region0/commit
```

Further Discussion

- We are a way from having use cases in Linux.
    - Type2 + cxl.cache code is out there (AMD).
    - UIO support (TODO).
    - Distributed/shared memory
        - famfs?

- DAX Region semantics
    - Decouple uAPI from coherence – make it creator defined.
    - An HDM-DB region could want the device-dax uAPI (ie: sharing).

- Memregion invalidations (sw enforced coherence) has always been the worse case.
    - Reality is probably a compromise between hardware and software approaches.
        - SF capacity misses vs device cost ($).

Conclusions

- Back-Invalidate is a key component for next generation CXL scalability.
  - Plenty of academic papers pointing at theoretical improvements.
  - Hardware is catching up with CXL3 IP.
    - CXL4 just released.

- Linux beyond the direct attached Type3 device use case.
  - Requires a lot of path finding for supporting the use cases opened up by Back-Invalidate.

Thank you.

THE NEXT CREATION STARTS HERE

Placing **memory** at the forefront of future innovation and creative IT life

Backup Slides

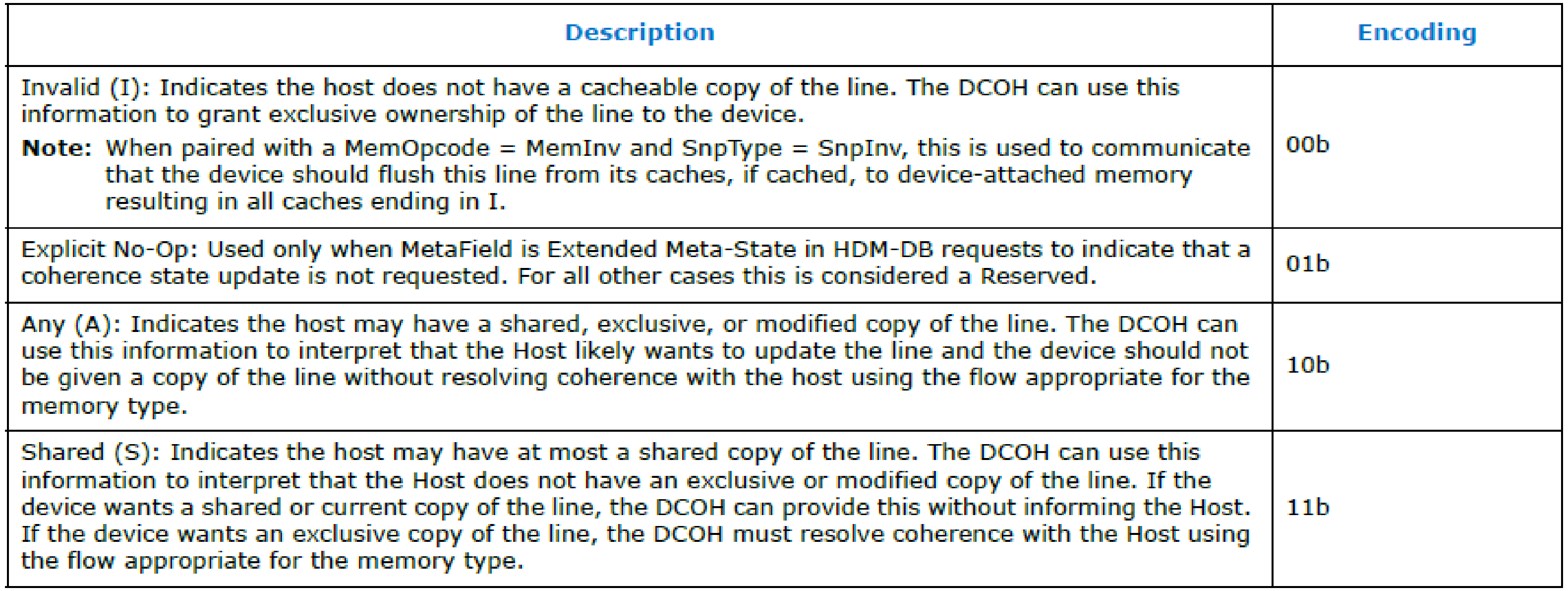
Resources

- An Introduction to the Compute Express Link Interconnect. Das Sharma, Blankenship, Berger. 2023
- Coherence Deep Dive for CXL. Blankenship, Robert. 2022.
- CXL® Specification - Compute Express Link (latest).

Inclusive Snoop Filter in the DCOH

- The snoop filter in the DCOH is a directory-like structure that maintains a 2-bit state, allowing it to encode up to 4 states.

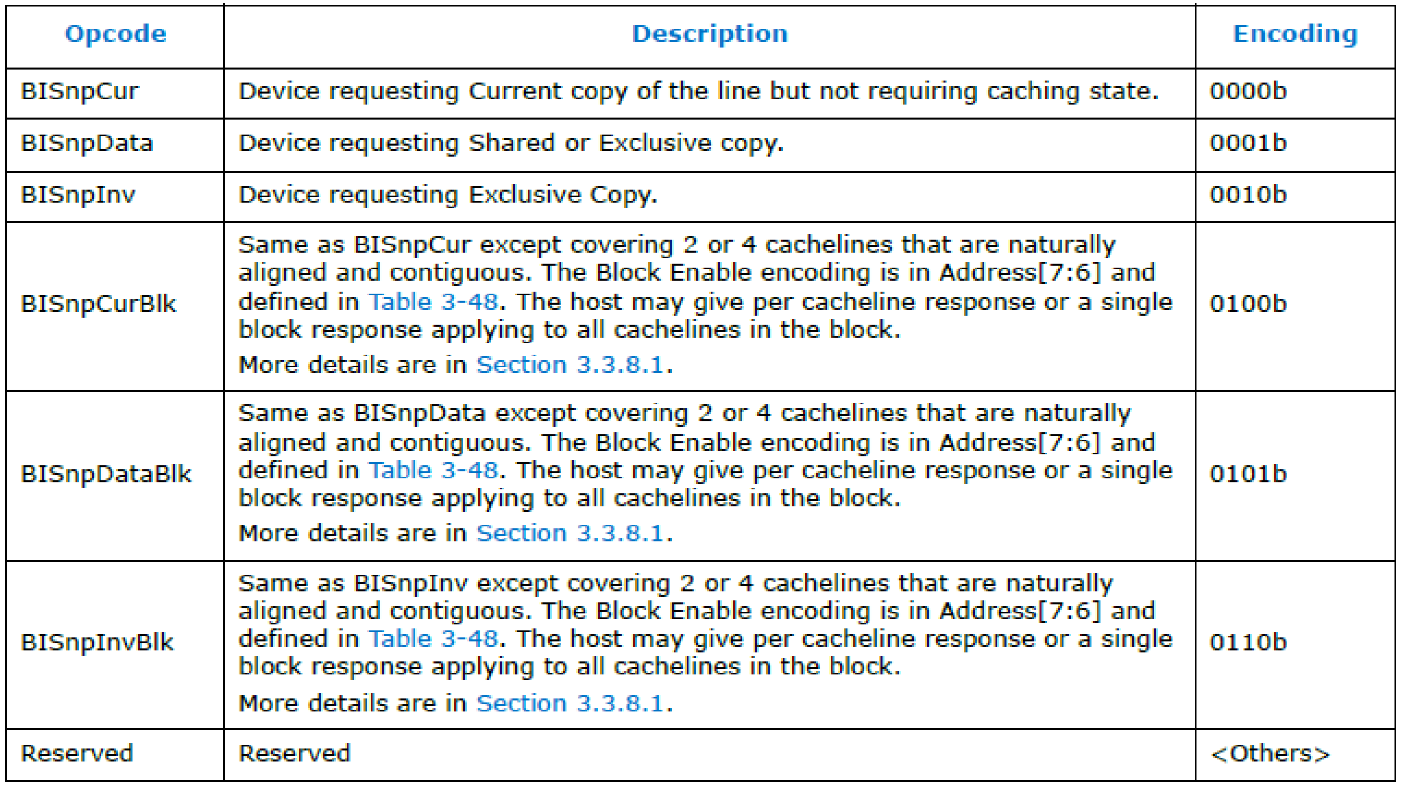
- The "Any" is special to CXL, the SF does not distinguish between "Modified" and "Exclusive".
  - Device unconditionally needs to resolve coherence.
  - Host can manipulate this cacheline without requiring further cache coherence transactions.



Description	Encoding
Invalid (I): Indicates the host does not have a cacheable copy of the line. The DCOH can use this information to grant exclusive ownership of the line to the device. **Note:** When paired with a MemOpcode = MemInv and SnpType = SnpInv, this is used to communicate that the device should flush this line from its caches, if cached, to device-attached memory resulting in all caches ending in I.	00b
Explicit No-Op: Used only when MetaField is Extended Meta-State in HDM-DB requests to indicate that a coherence state update is not requested. For all other cases this is considered a Reserved.	01b
Any (A): Indicates the host may have a shared, exclusive, or modified copy of the line. The DCOH can use this information to interpret that the Host likely wants to update the line and the device should not be given a copy of the line without resolving coherence with the host using the flow appropriate for the memory type.	10b
Shared (S): Indicates the host may have at most a shared copy of the line. The DCOH can use this information to interpret that the Host does not have an exclusive or modified copy of the line. If the device wants a shared or current copy of the line, the DCOH can provide this without informing the Host. If the device wants an exclusive copy of the line, the DCOH must resolve coherence with the Host using the flow appropriate for the memory type.	11b

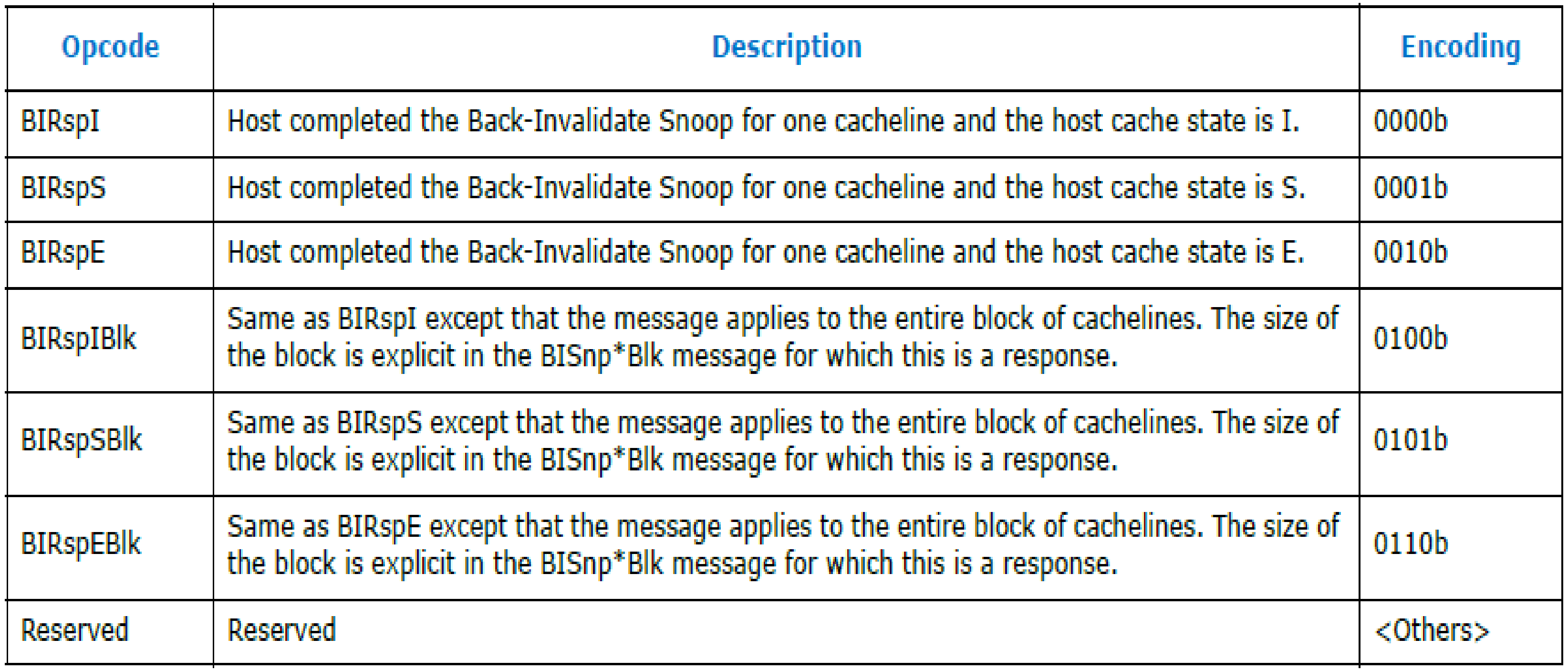
S2M Back-Invalidate Snoop (BISnp)

○ Specifies what snoop type, if any, needs to be issued by the DCOH and the minimum coherency state required by the Host.



Opcode	Description	Encoding
BISnpCur	Device requesting Current copy of the line but not requiring caching state.	0000b
BISnpData	Device requesting Shared or Exclusive copy.	0001b
BISnpInv	Device requesting Exclusive Copy.	0010b
BISnpCurBlk	Same as BISnpCur except covering 2 or 4 cachelines that are naturally aligned and contiguous. The Block Enable encoding is in Address[7:6] and defined in Table 3-48. The host may give per cacheline response or a single block response applying to all cachelines in the block. More details are in Section 3.3.8.1.	0100b
BISnpDataBlk	Same as BISnpData except covering 2 or 4 cachelines that are naturally aligned and contiguous. The Block Enable encoding is in Address[7:6] and defined in Table 3-48. The host may give per cacheline response or a single block response applying to all cachelines in the block. More details are in Section 3.3.8.1.	0101b
BISnpInvBlk	Same as BISnpInv except covering 2 or 4 cachelines that are naturally aligned and contiguous. The Block Enable encoding is in Address[7:6] and defined in Table 3-48. The host may give per cacheline response or a single block response applying to all cachelines in the block. More details are in Section 3.3.8.1.	0110b
Reserved	Reserved	<Others>

M2S Back-Invalidate Response (BIRsp)



Opcode	Description	Encoding
BIRspI	Host completed the Back-Invalidate Snoop for one cacheline and the host cache state is I.	0000b
BIRspS	Host completed the Back-Invalidate Snoop for one cacheline and the host cache state is S.	0001b
BIRspE	Host completed the Back-Invalidate Snoop for one cacheline and the host cache state is E.	0010b
BIRspIBlk	Same as BIRspI except that the message applies to the entire block of cachelines. The size of the block is explicit in the BISnp*Blk message for which this is a response.	0100b
BIRspSBlk	Same as BIRspS except that the message applies to the entire block of cachelines. The size of the block is explicit in the BISnp*Blk message for which this is a response.	0101b
BIRspEBlk	Same as BIRspE except that the message applies to the entire block of cachelines. The size of the block is explicit in the BISnp*Blk message for which this is a response.	0110b
Reserved	Reserved	<Others>