Advanced UCIe-based Chiplets verification from IP to SoC

Anunay Bajaj
Moshik Rubin

March 2024

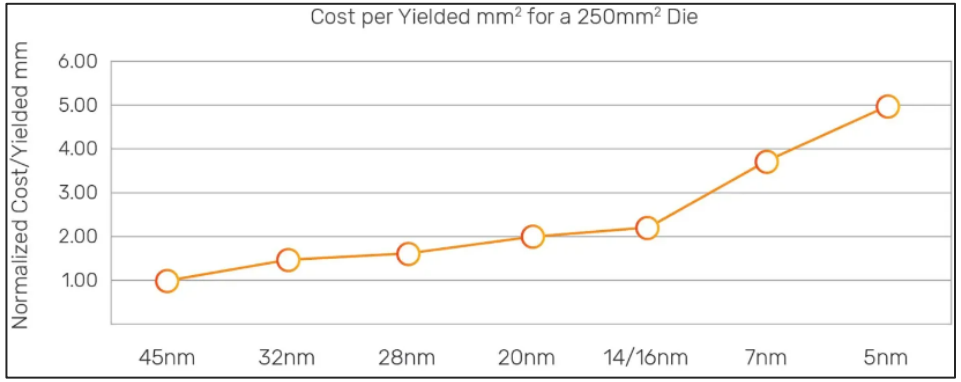
cadence®

Agenda

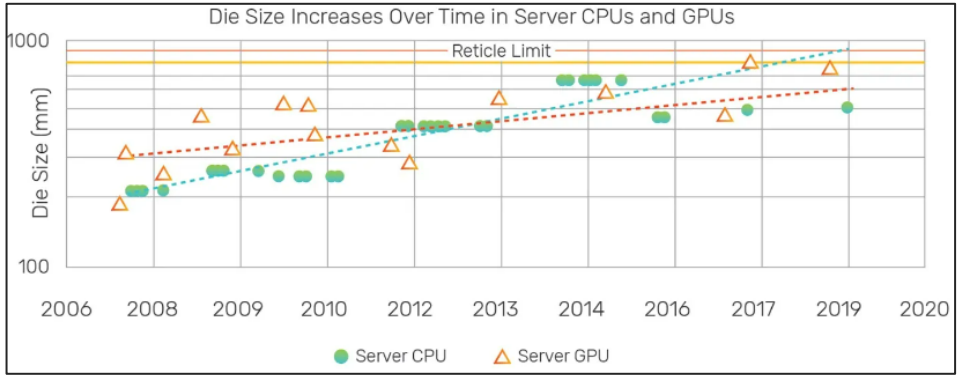
- Chiplets designs trends and examples
- UCIe based Chiplets – Design topologies => verification challenges
- Verification approaches and examples from IP to SoC
- Q & A

Factors leading to Chiplet Adoption

- Increase in Die size for CPUs/GPUs
- Expensive to fabricate Die with 7nm or lower nodes
- Improve Time-to-solution (Die reuse)
- Need to Lower manufacturing costs by purchasing known-good die (KGD)
- Room for Customization needed (bespoke solutions)
- Need to Scale innovation (manufacturing and process locked IPs)



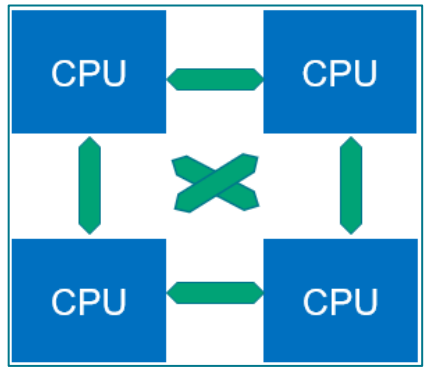
Source:

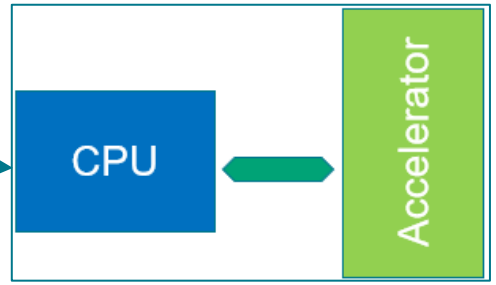


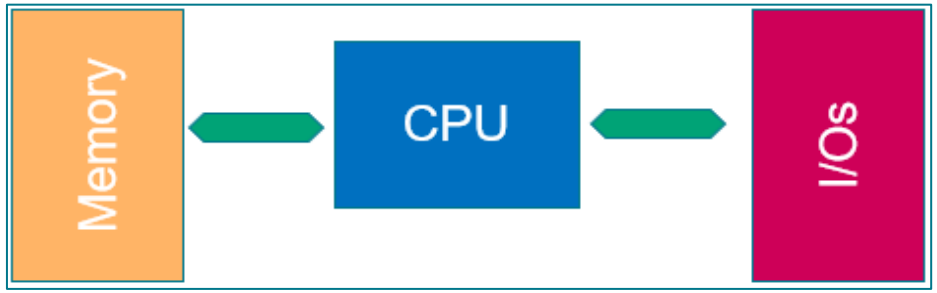
Source:

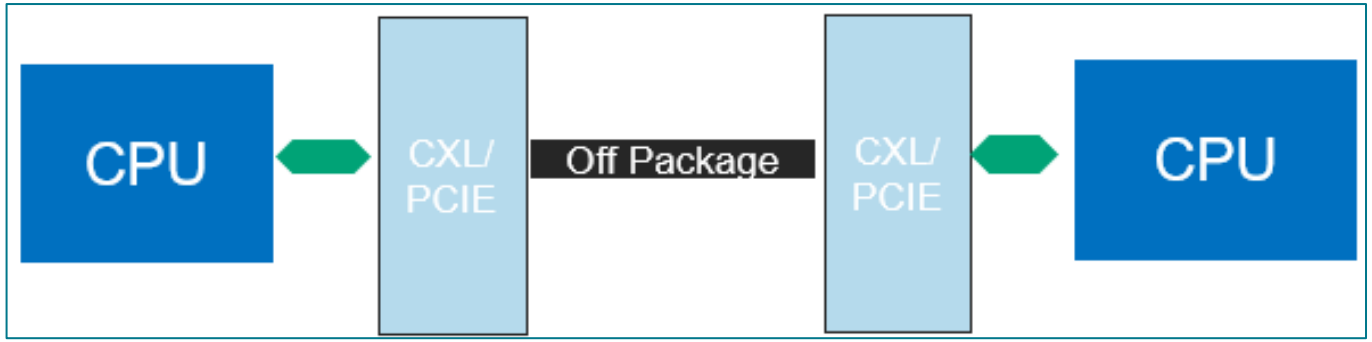
Chiplet Industry Applications

- Scalable CPU clusters
- Heterogenous Compute
- IO & Memory Split
- Resource Pooling

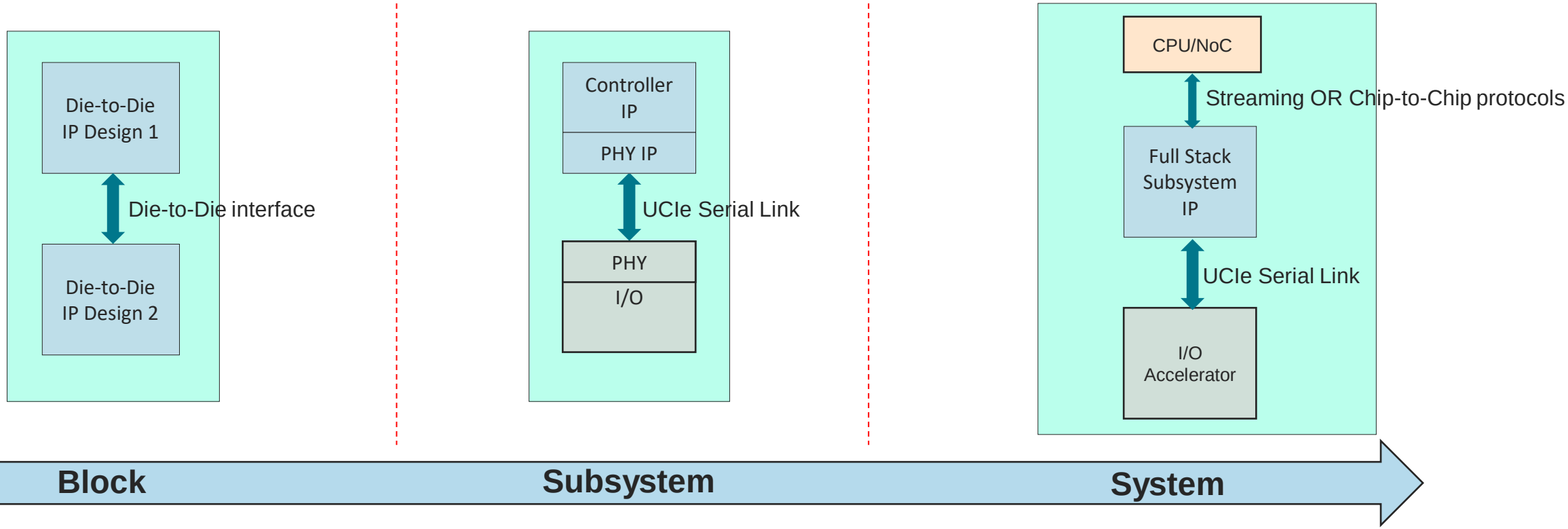








Simplified UCIe based Design Topologies

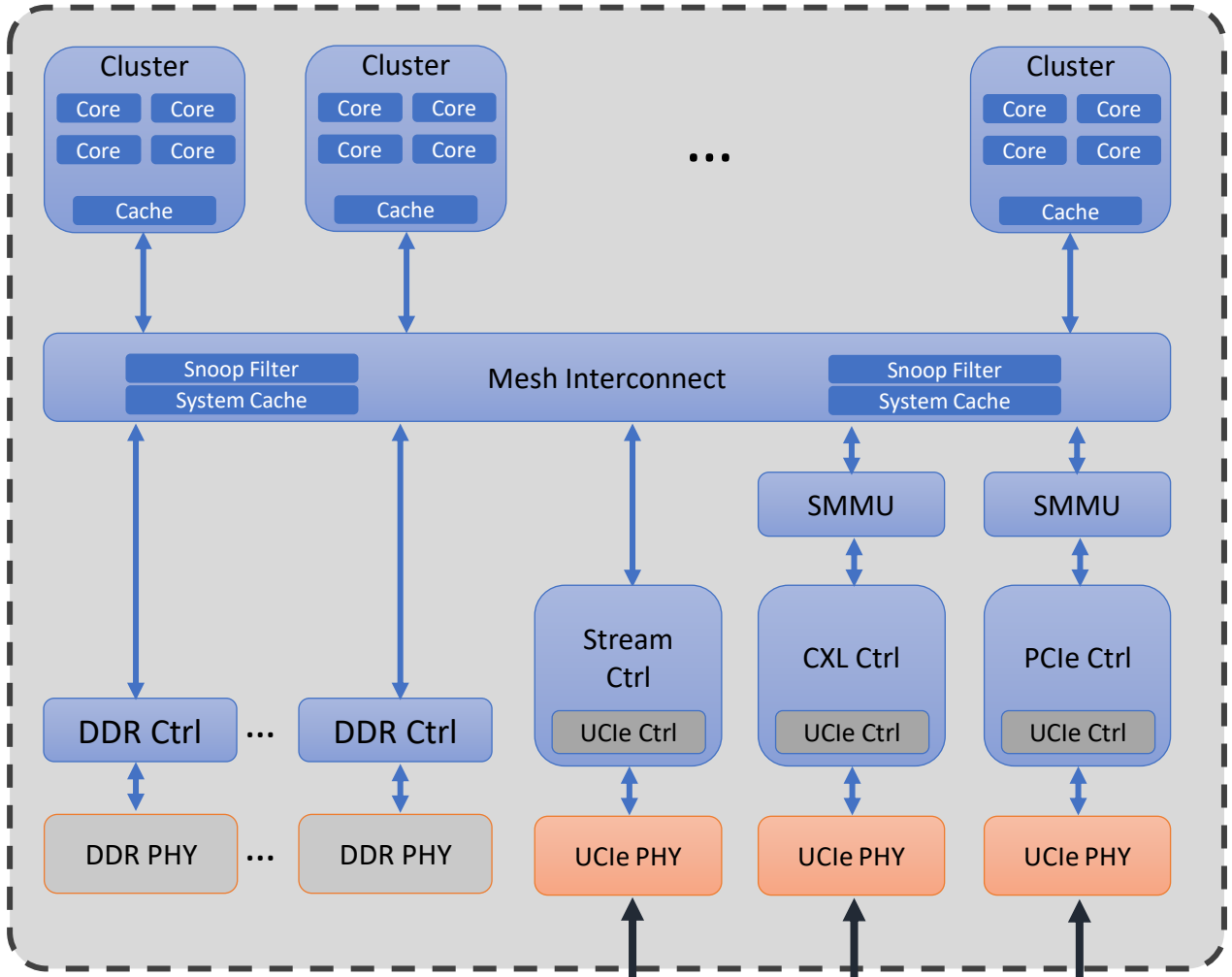


- Homogenous Die-to-Die FDI/RDI/PHY only connect
- UCIe Individual Block developers

- Memory or I/O Split Serial Peripheral connect
- Controller Developers

- Heterogenous Compute Coherency Streaming & Peripheral connect
- System Developers with CPU Bus protocols & PCIE/CXL

UCIe in Action -Example Chiplet Design



- Explosion of Design Topologies
  - Several Unit Levels in different combinations
- Multiprotocol Verification
  - PCI Express using UCIe as the Transport Layer
  - CXL using UCIe as the Transport Layer
  - Streaming interface
- System Level Implications
  - -End-to-end Data Integrity
  - -Latency Calculation or Turn Around Time (TAT)

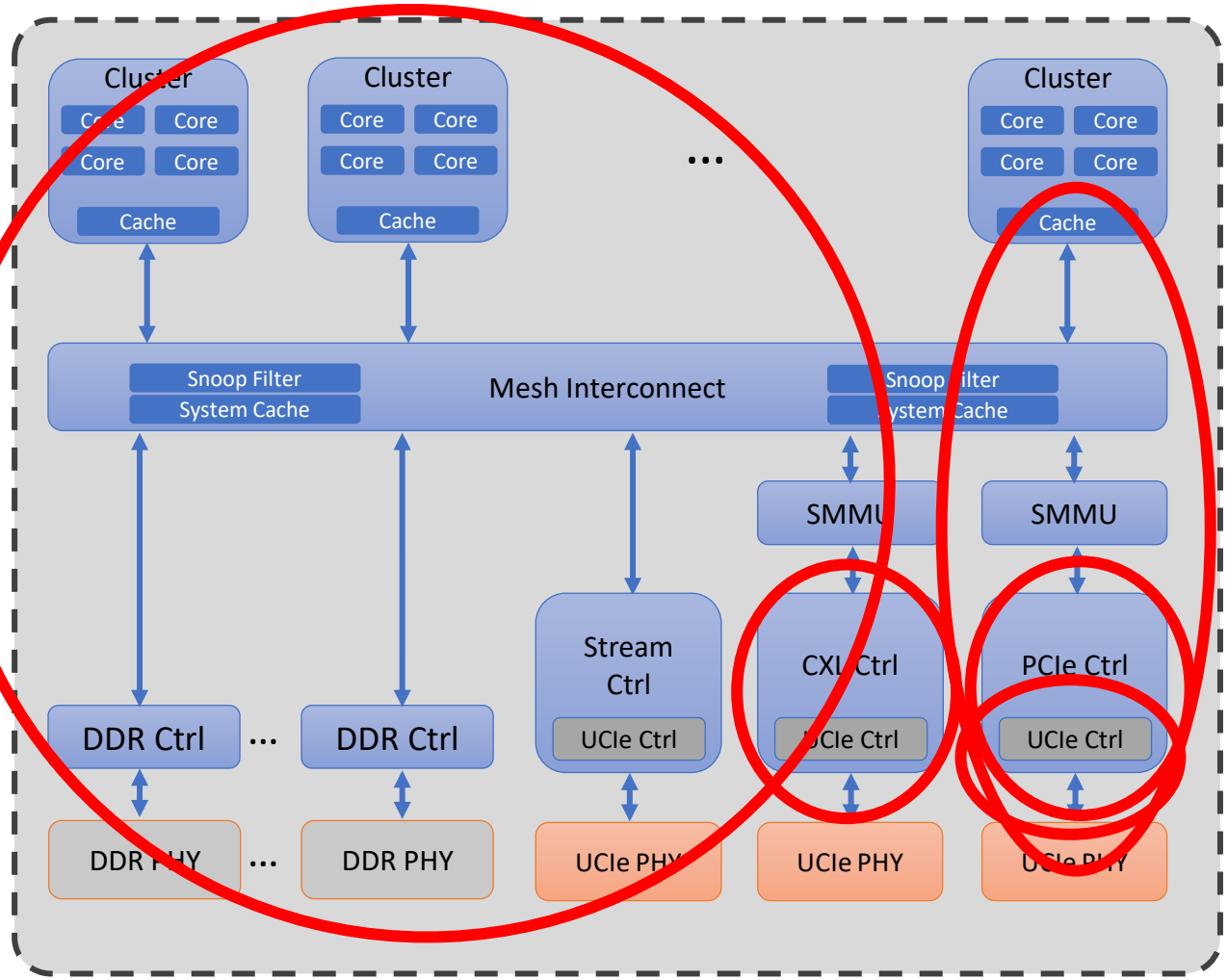
and more…

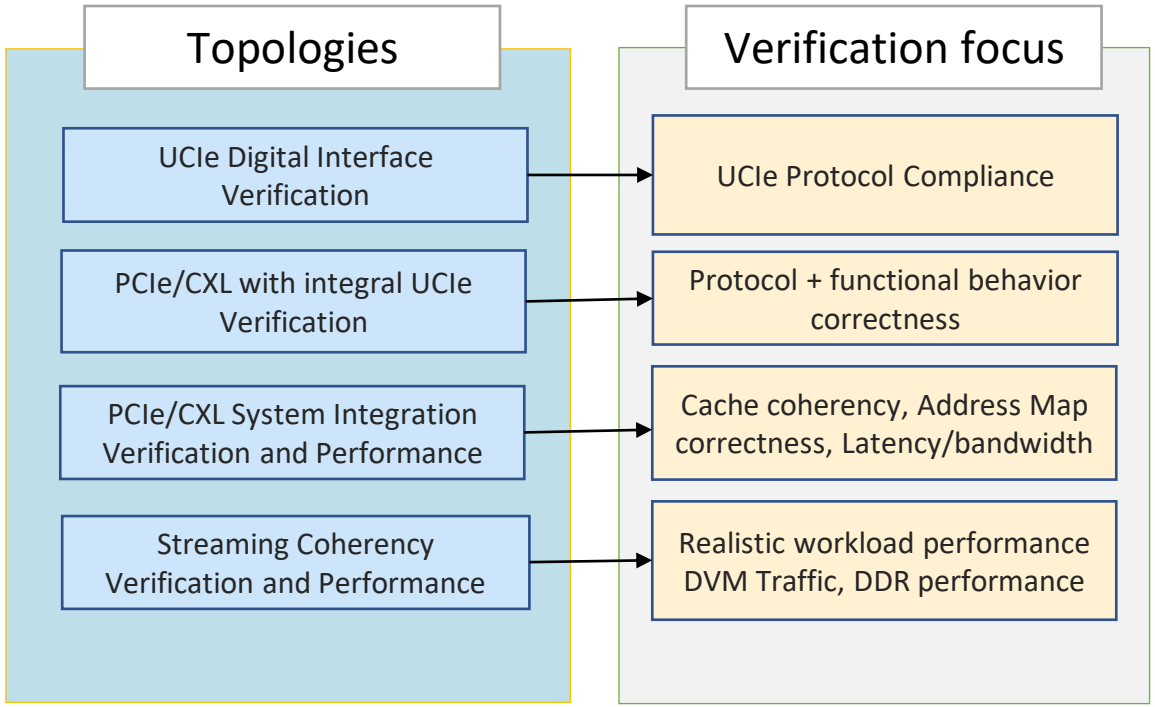
Companion Die

- Storage (CXL)
- High Speed I/O (PCIe)
- Symmetric Multi-Processor (Streaming)

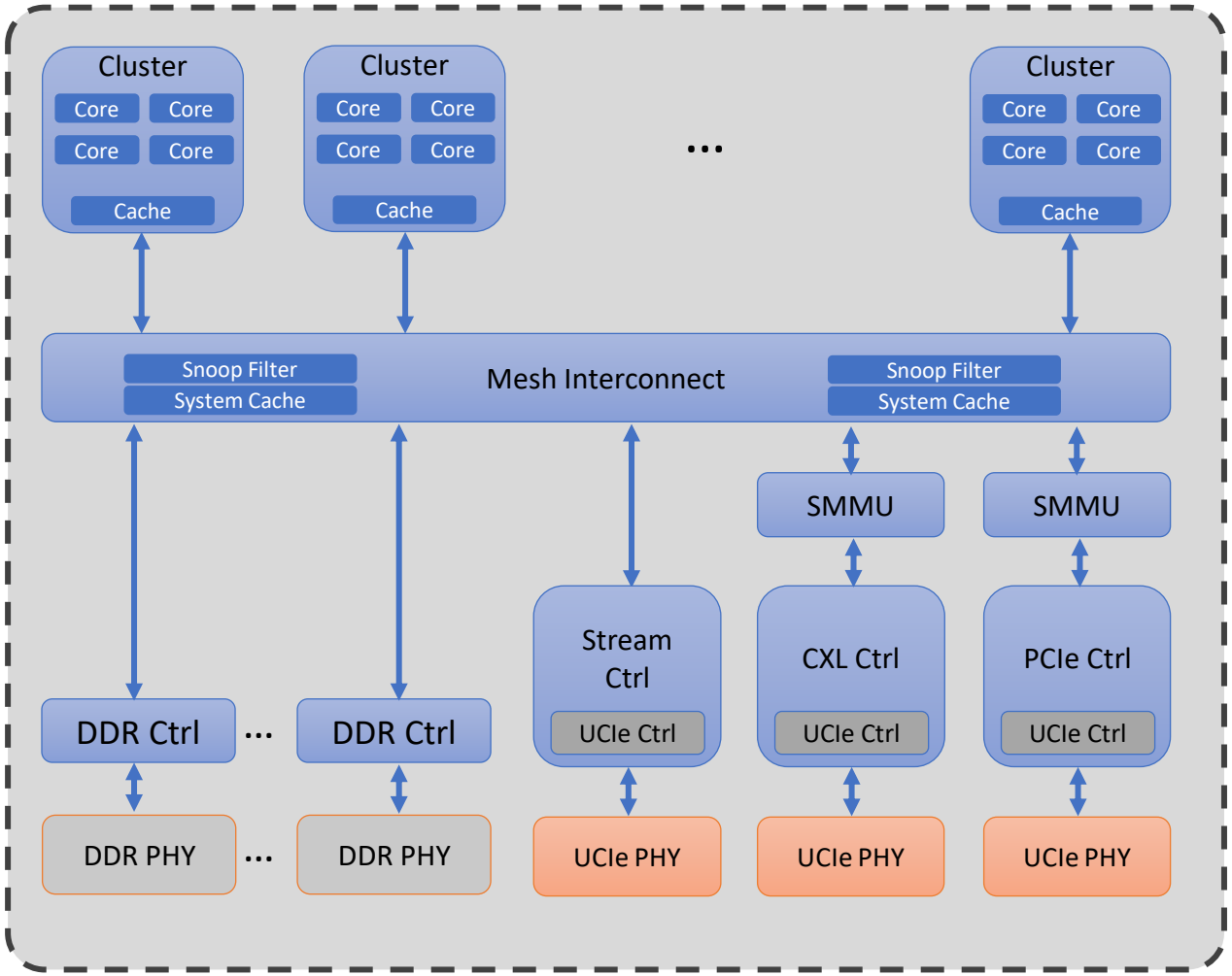
UCIe Serial Link

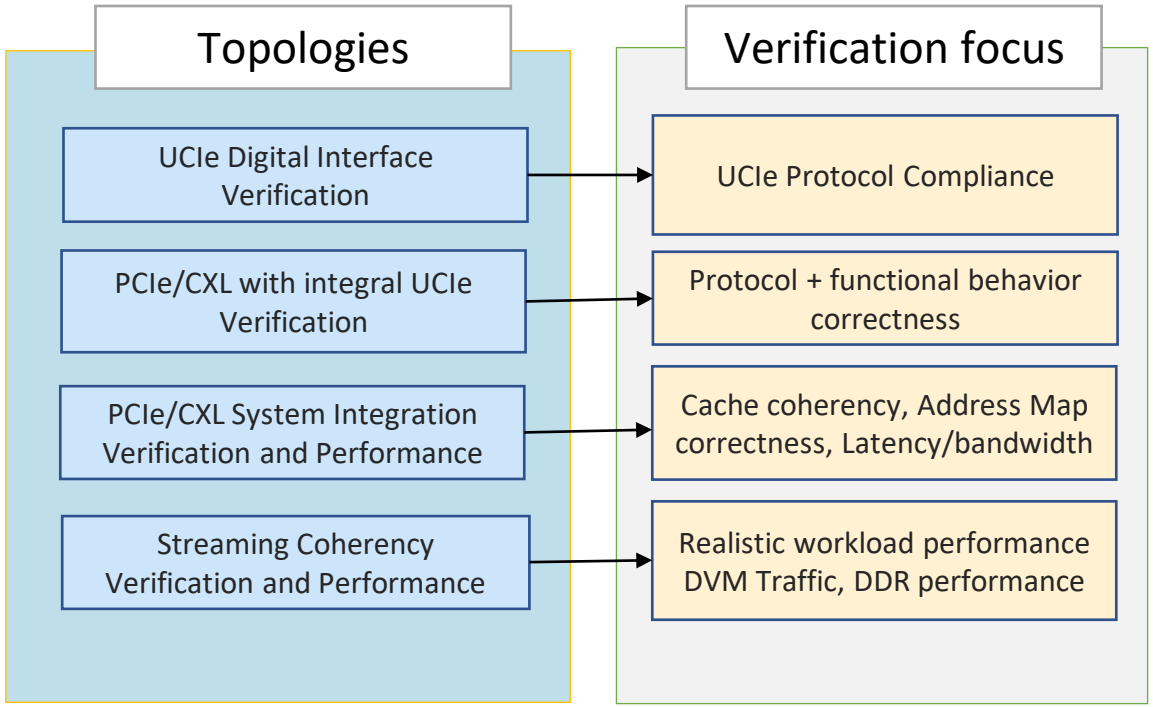
Verification Areas – Leverage Multiple Technologies



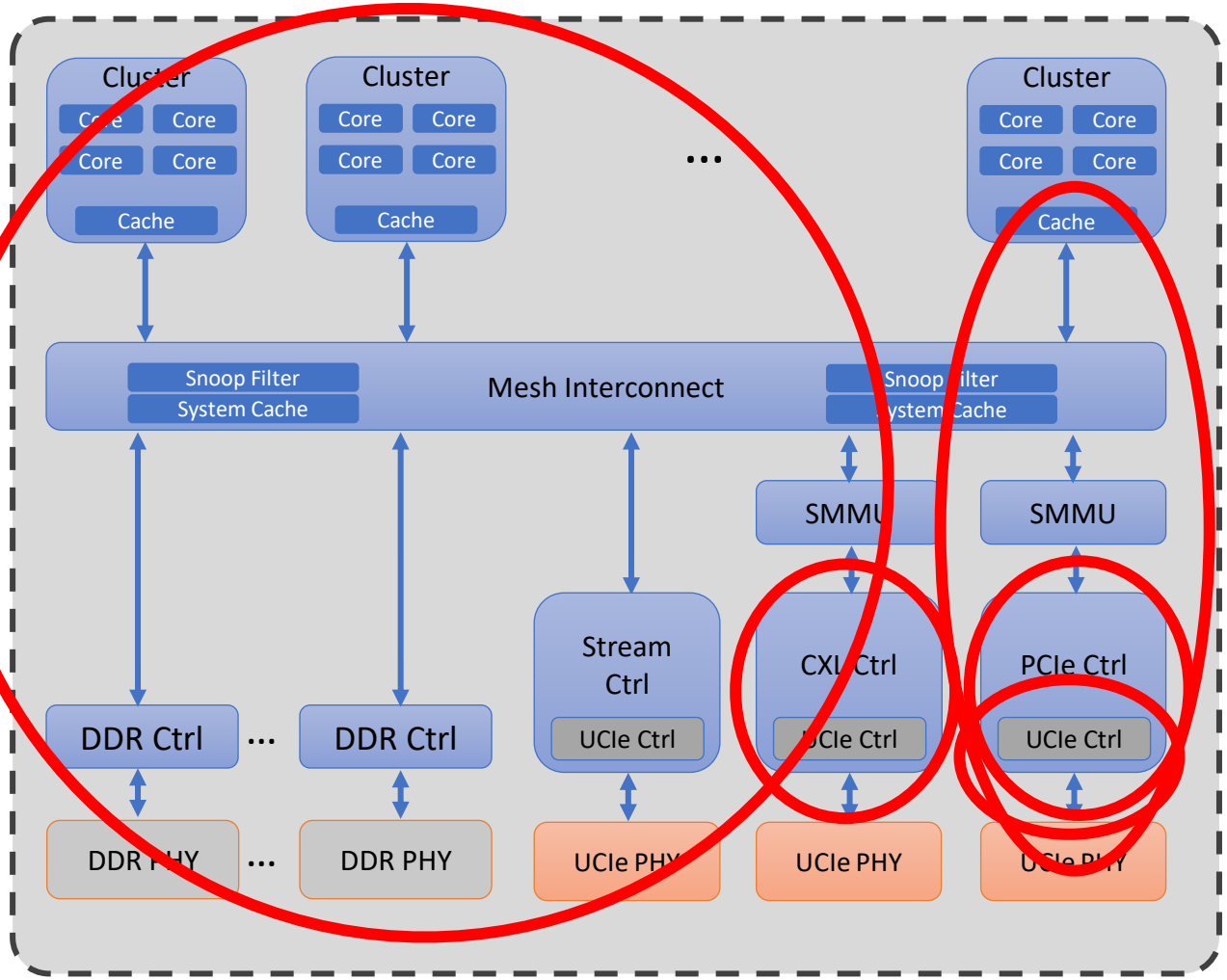


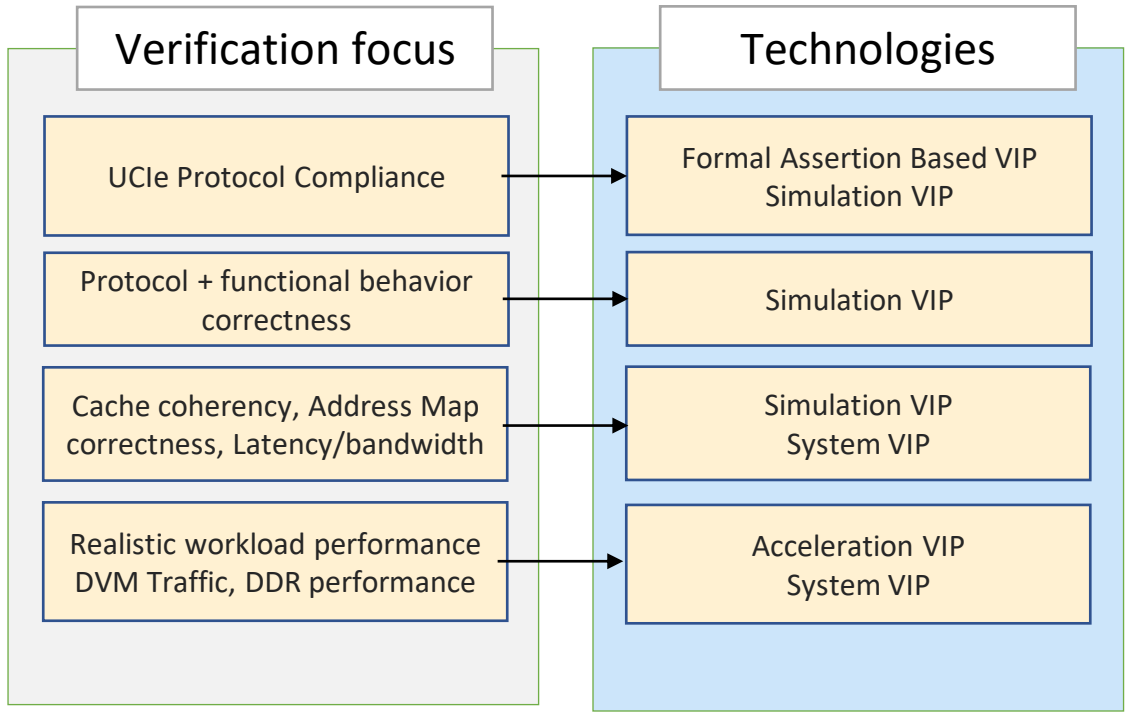
Verification Areas – Leverage Multiple Technologies





Verification Areas – Leverage Multiple Technologies



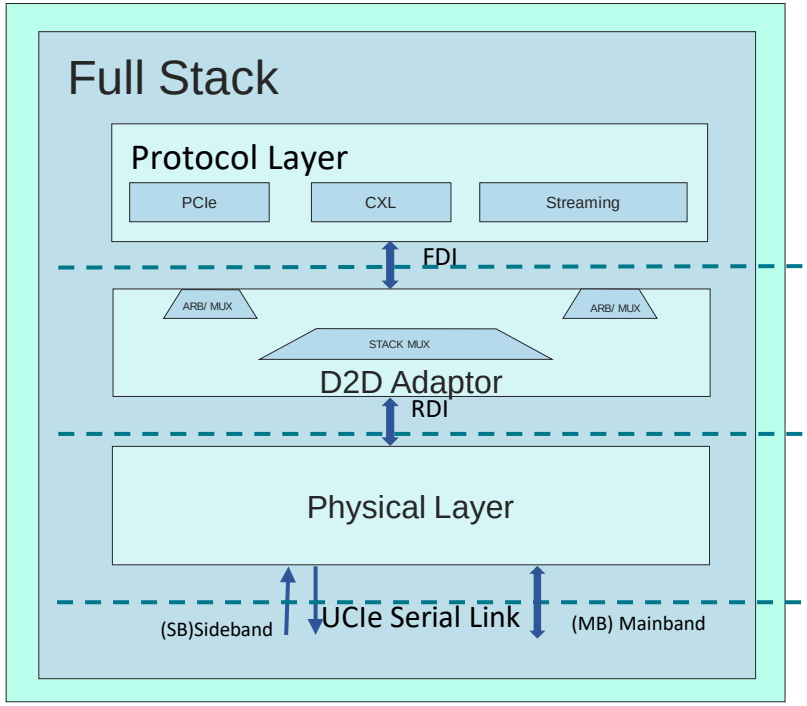


Glossary:

- **Formal VIP**: Set of formal assertions
- **Simulation VIP**: UVM based VIP (BFM, Monitor, seq library)
- **Accelerated VIP**: Emulation ready VIP
- **System VIP**: SoC level tools, integrated with VIP/AVIP

UCIe IP level Verification

UCIe Architecture



- **PCIe 6.0,** (PCIe non-FLIT via CXL.io)
- **CXL 2.0 - 68B, CXL3.0 - 256B**
- **Streaming Protocol**

(FDI) Flit Aware Die-to-Die Interface

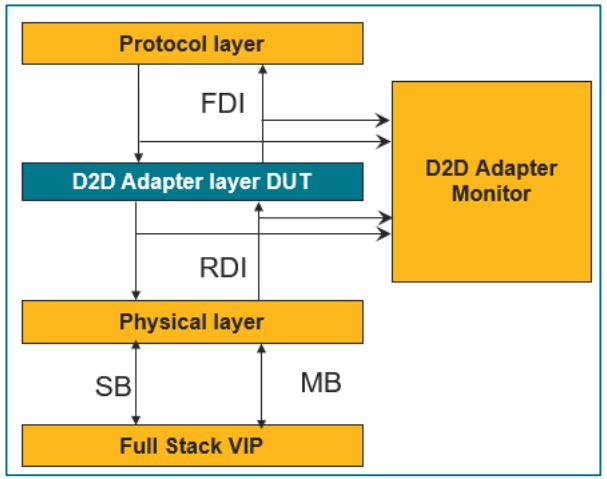
- **D2D Adapter coordinates with the Protocol Layer and the Physical Layer**
- **Link State Management and Parameter negotiation**

(RDI) Raw Die-to-Die Interface

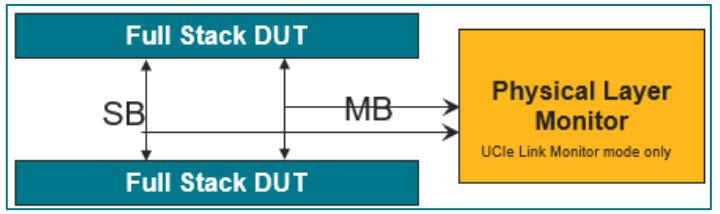
- **Physical Layer** for Data Path, Link Training, Lane Repair, parameter exchanges, register access etc.
- Standard and Advanced Package
- Single Module & Multi module PHY

UCIe Serial Link (Mainband & Sideband Lanes)

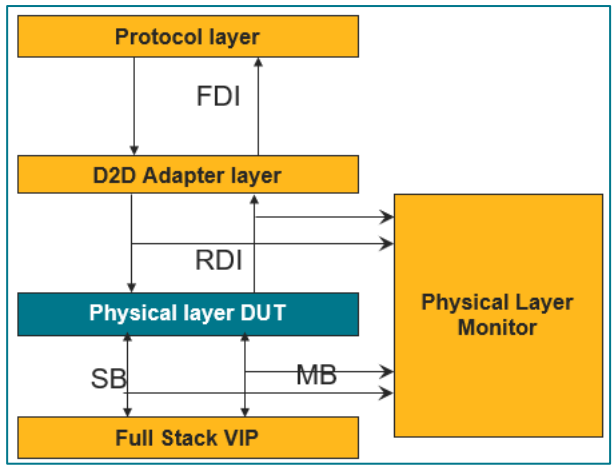
Various IP Verification Topologies



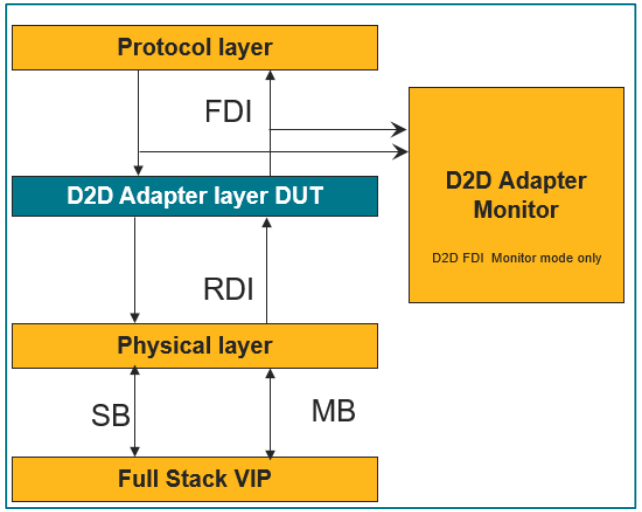
- Checks FDI Flits inbound and outbound to Adapter Layer DUT
- Checks RDI Flits inbound and outbound to Adapter Layer DUT
- Multi Protocol and Multi Stack capable
- D2D FDI Interface Coverage
- D2D RDI Interface Coverage



- Checks Serial Mainband and Sideband flow of PHY DUT
- PHY Serial/UCIe Link Coverage

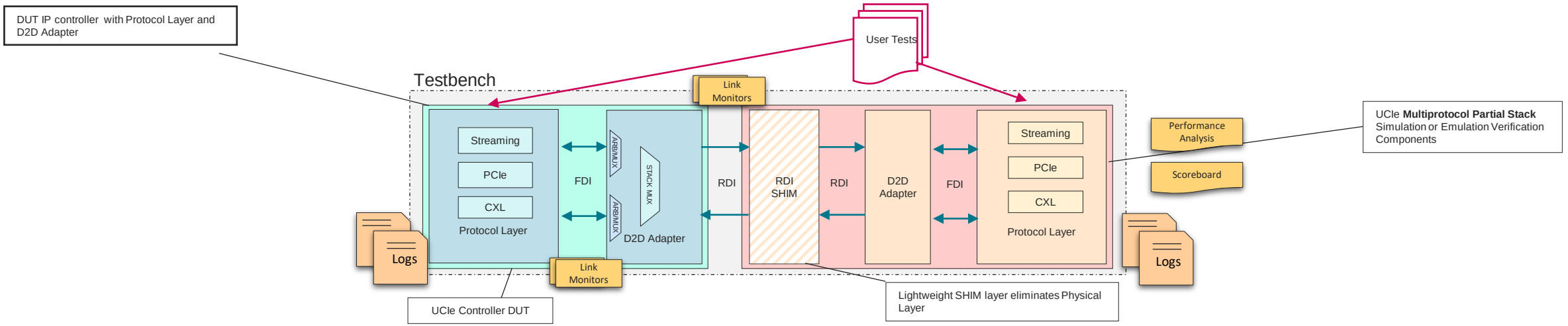


- Checks RDI Flits inbound and outbound to PHY DUT
- Checks Serial Mainband and Sideband flow of PHY DUT
- PHY RDI Interface Coverage
- PHY Serial/ UCIe Link Coverage



- Checks FDI Flits inbound and outbound to Adapter Layer DUT
- Multi Protocol and Multi Stack capable
- D2D FDI Interface Coverage

Unit Level - Adapter & Protocol Layer
Standalone IP Verification



Verification Focus

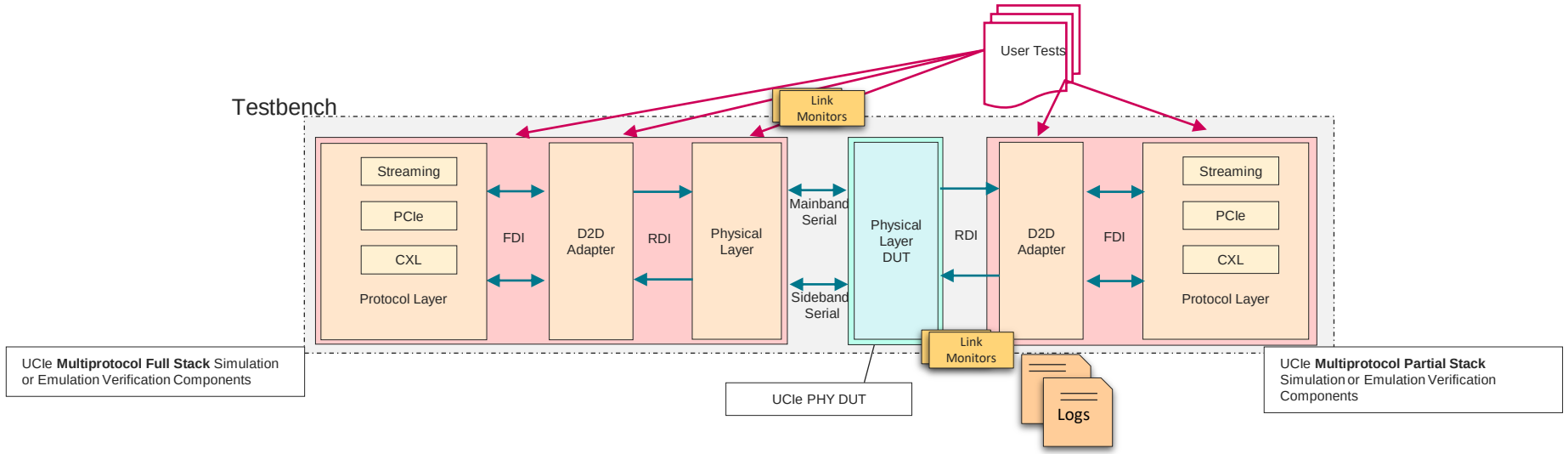
- Fast Simulation/Emulation as no Physical Layer involved
- Bypass Link Training & Setup controls
- Die-to-Die Adapter Correctness
- Protocol Layer Correctness
- Stress Testing with Bulk Reads/Writes
- End-to-end Transaction performance

Legend

Verification Components or Modules

DUT RTL

Unit Level - Physical Layer
Standalone IP Verification



Verification Focus

- Detailed Physical Layer Verification
- Handshake with local RDI and remote link partner
- Complexities include Packaging/ Multimodule/ LTSM training and failures/ Reversal/ Retry
- Physical Layer Sideband and Mainband transactions
- RDI throttling

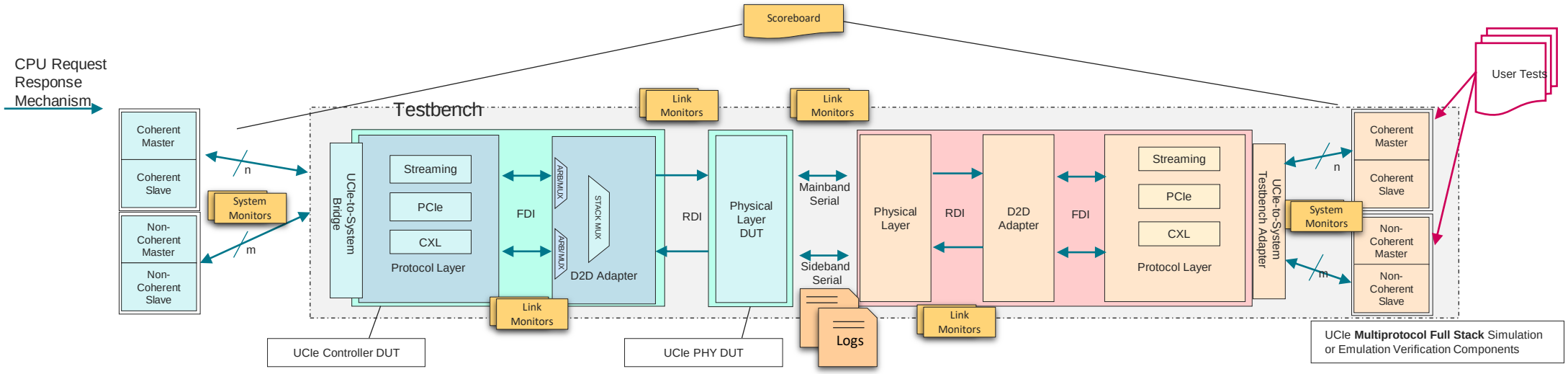
Legend

Verification Components or Modules

DUT RTL

UCIe Sub-System level Verification

Subsystem Level Verification– With Coherent & Non-Coherent Protocols



Verification Focus

- End-to-end Request & Response correctness
- Protocol Translation Monitoring at Interface boundaries
- Layer-wise Monitors for step-by-step checks
- Handshake with local RDI and remote link partner

Legend

Verification Components or Modules

DUT RTL

Interface connections

Add Pureview slide

Instantiate Full Stack US agent and Protocol+D2D+Physical (acting as Full Stack DS agent) in the testbench module. Instantiations are shown below.

1

- Physical layer interface of DS Agent
- D2D Adapter layer interface of DS Agent
- Protocol layer interface of DS Agent

Downstream

2

- Full Stack interface of US Agent

Upstream

```
module testbench;
…
// Instantiate the DS agent.
  ucciePhy_interface   dsAgent(
       .tx_ck_p            (tx_ck_p),
       .rx_ck_n            (rx_ck_p),
       .tx_vld             (rx_vld),
       .tx_data            (rx_data),
       .rx_vld             (tx_vld),
       .rx_data            (tx_data),
       …
       .rdi_lp_data        (dsAgent_rdi_lp_data),
  );
  ucieAdp_interface   adpDsAgent(
       .rdi_lp_data        (dsAgent_rdi_lp_data),
       …
       .fdi_lp_data0       (dsAgent_fdi_lp_data0),
  );
  ucieProtocol_interface    protoDsAgent(
       .fdi_lp_data0       (dsAgent_fdi_lp_data0),
  );

// Instantiate the US agent.
  ucieFullStack_interface   fullStackUsAgent(
     .tx_ck_p              (rx_ck_p),
     .rx_ck_p              (tx_ck_p),
     .tx_vld_rd            (rx_vld_rd),
     .tx_data_rd           (rx_data_rd),
     .rx_vld_rd            (tx_vld_rd),
     .rx_data_rd           (tx_data_rd),
     …
  );
endmodule
```

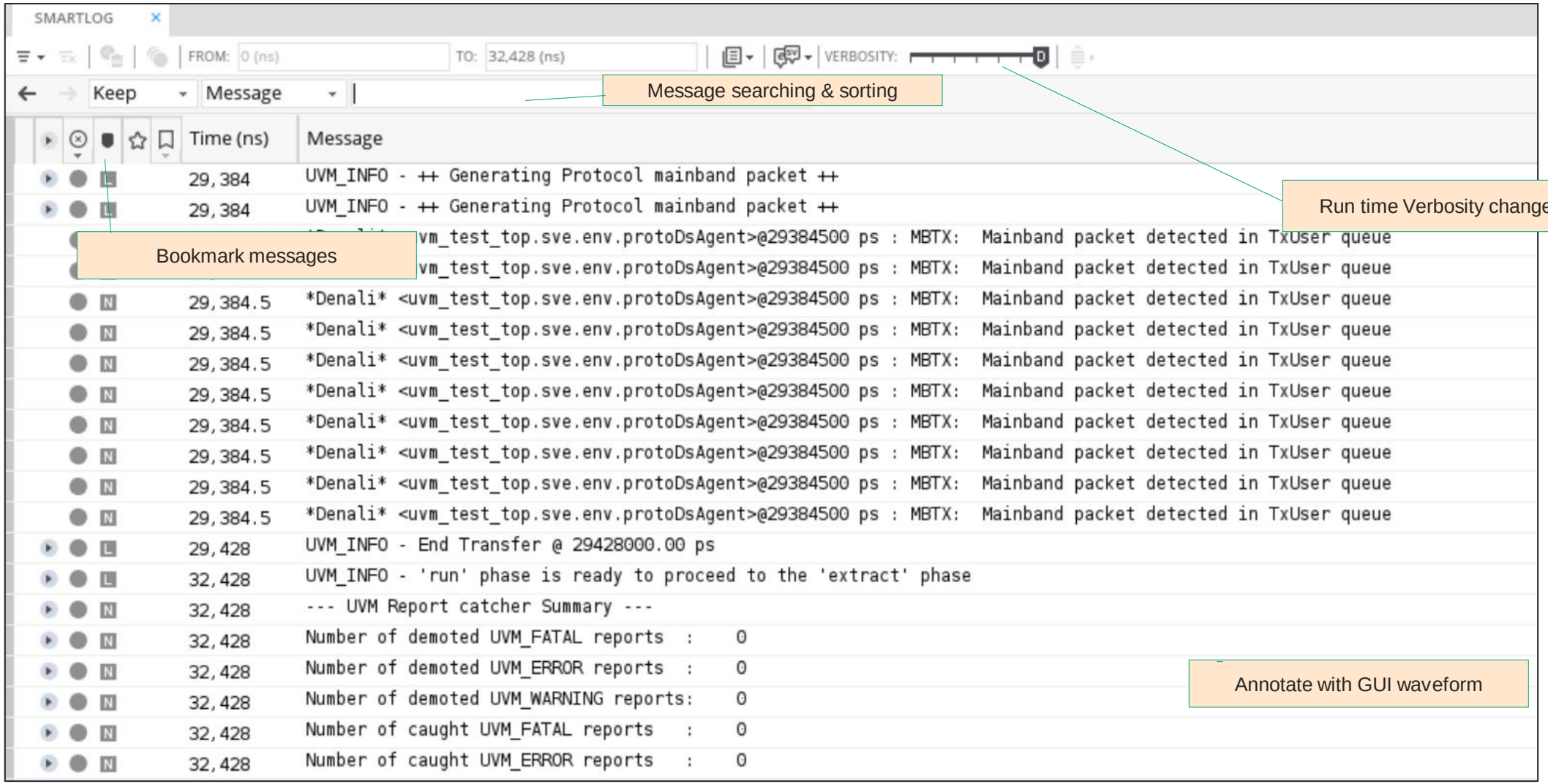
cdnUcieUvmUserFullStackTb.sv

3 Connection between UP and DS Agent. (Tx ports are connected to Rx ports and vice versa)

4 Internal connection between Phy layer and D2D Adapter layer of DS

5 Internal connection between Protocol layer and D2D Adapter layer of DS

Verisium Smart Log

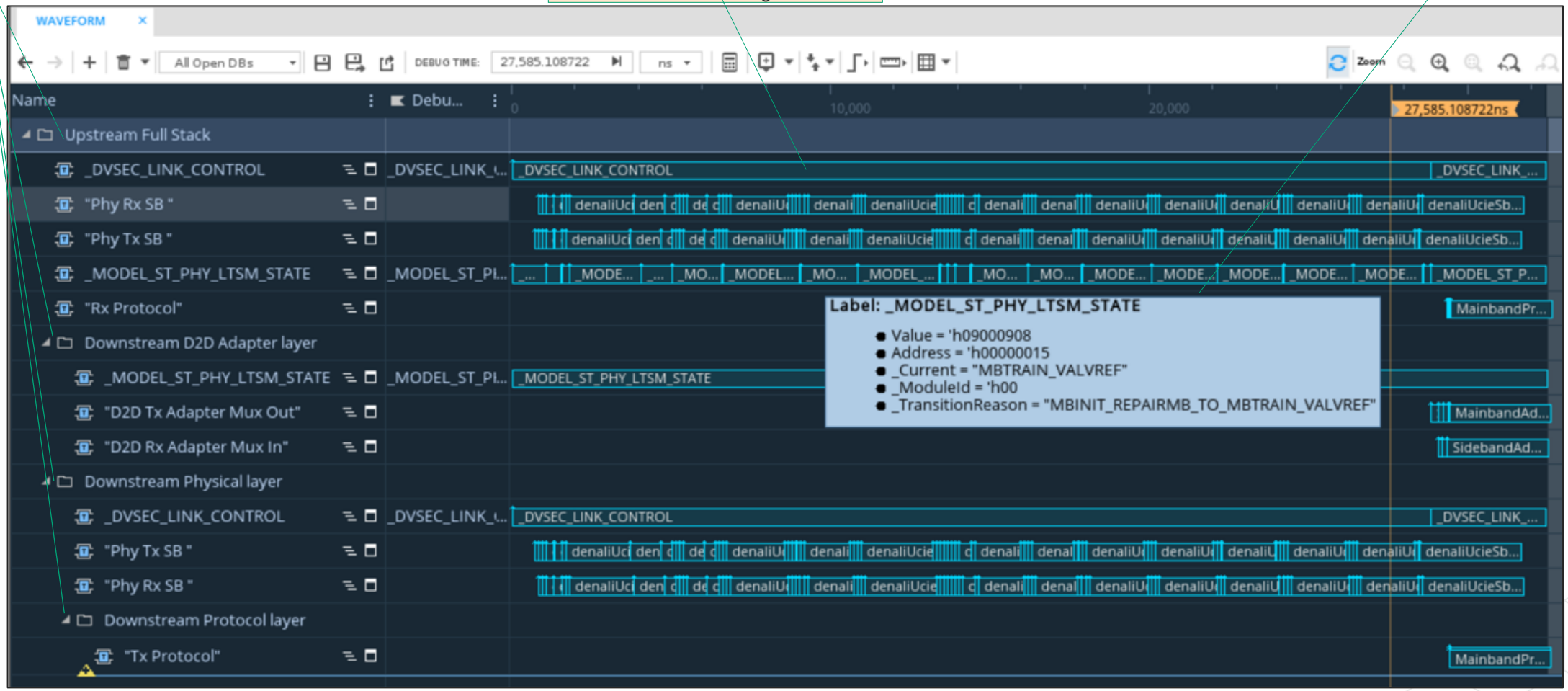


Debugging UCIe data flow

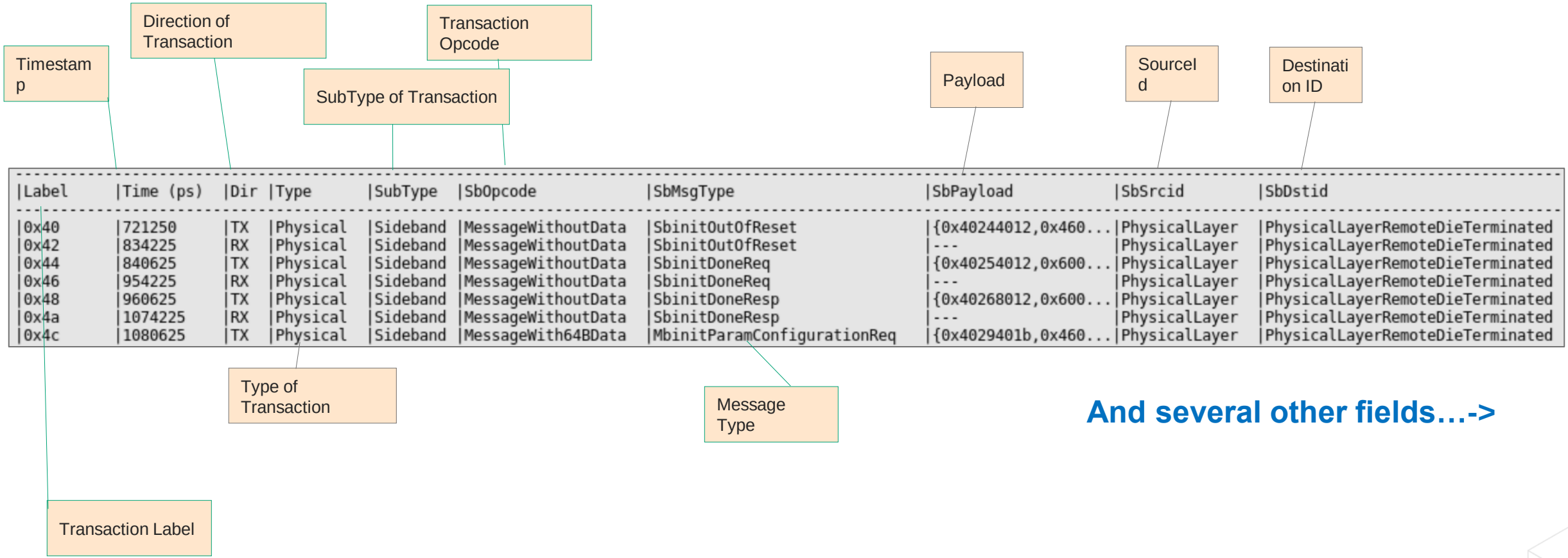
Upstream & Downstream agents in groups

GUI view
- Registers
- Packets
- State Machine Changes

Labels with detailed information



UCIE Packets view

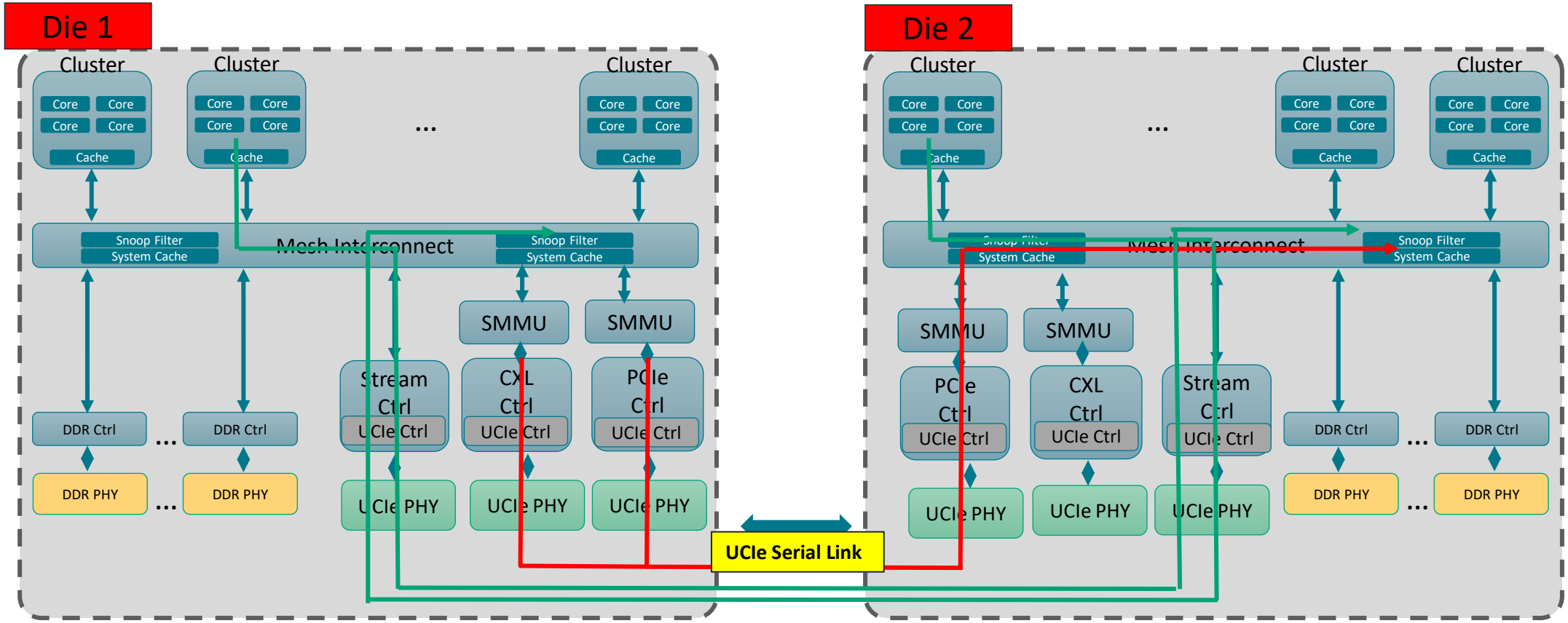


Label	Time (ps)	Dir	Type	SubType	SbOpcode	SbMsgType	SbPayload	SbSrcid	SbDstid
0x40	721250	TX	Physical	Sideband	MessageWithoutData	SbinitOutOfReset	{0x40244012,0x460...	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x42	834225	RX	Physical	Sideband	MessageWithoutData	SbinitOutOfReset	---	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x44	840625	TX	Physical	Sideband	MessageWithoutData	SbinitDoneReq	{0x40254012,0x600...	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x46	954225	RX	Physical	Sideband	MessageWithoutData	SbinitDoneReq	---	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x48	960625	TX	Physical	Sideband	MessageWithoutData	SbinitDoneResp	{0x40268012,0x600...	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x4a	1074225	RX	Physical	Sideband	MessageWithoutData	SbinitDoneResp	---	PhysicalLayer	PhysicalLayerRemoteDieTerminated
0x4c	1080625	TX	Physical	Sideband	MessageWith64BData	MbinitParamConfigurationReq	{0x4029401b,0x460...	PhysicalLayer	PhysicalLayerRemoteDieTerminated

And several other fields…->

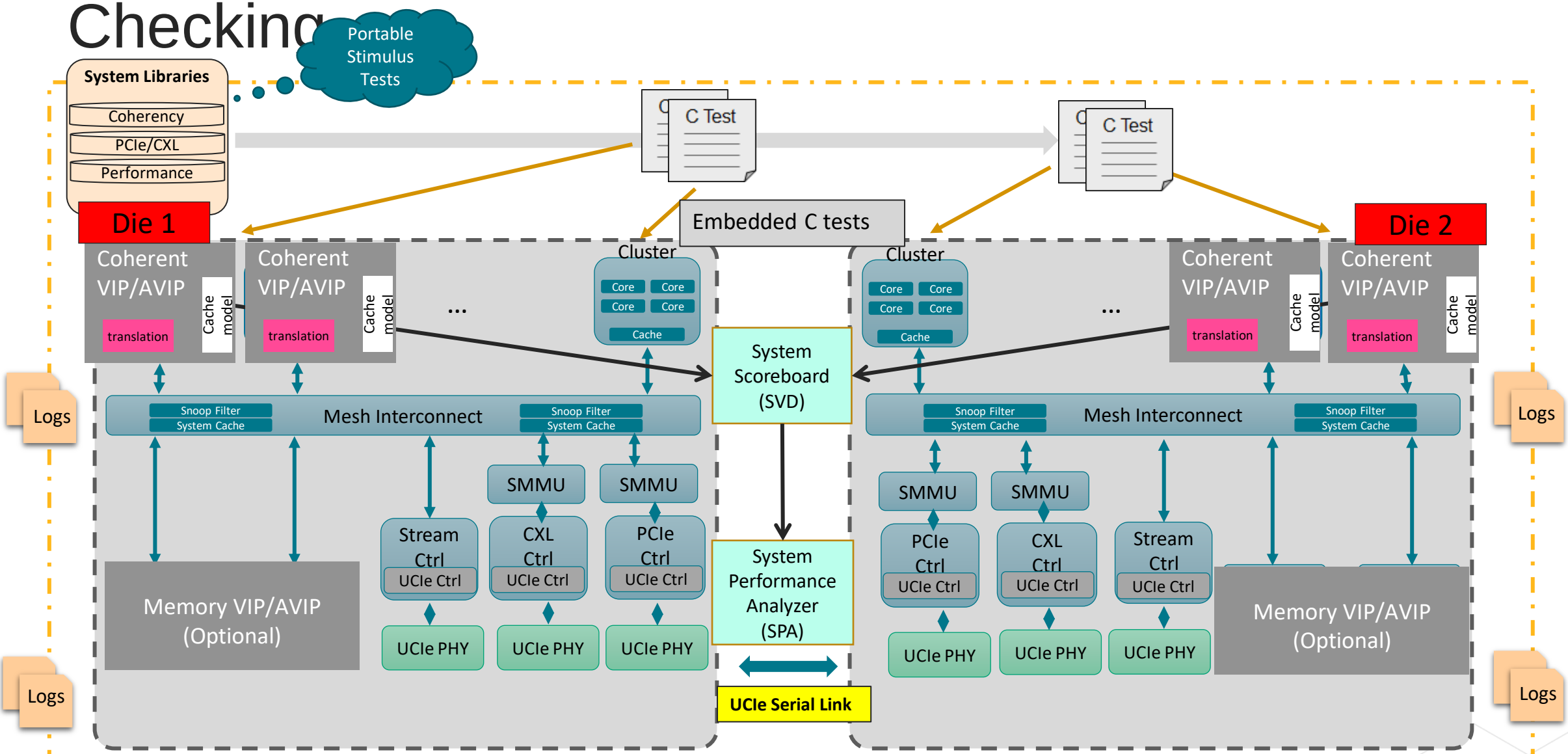
UCIe System level Verification

System Level- Chiplet Performance Concerns

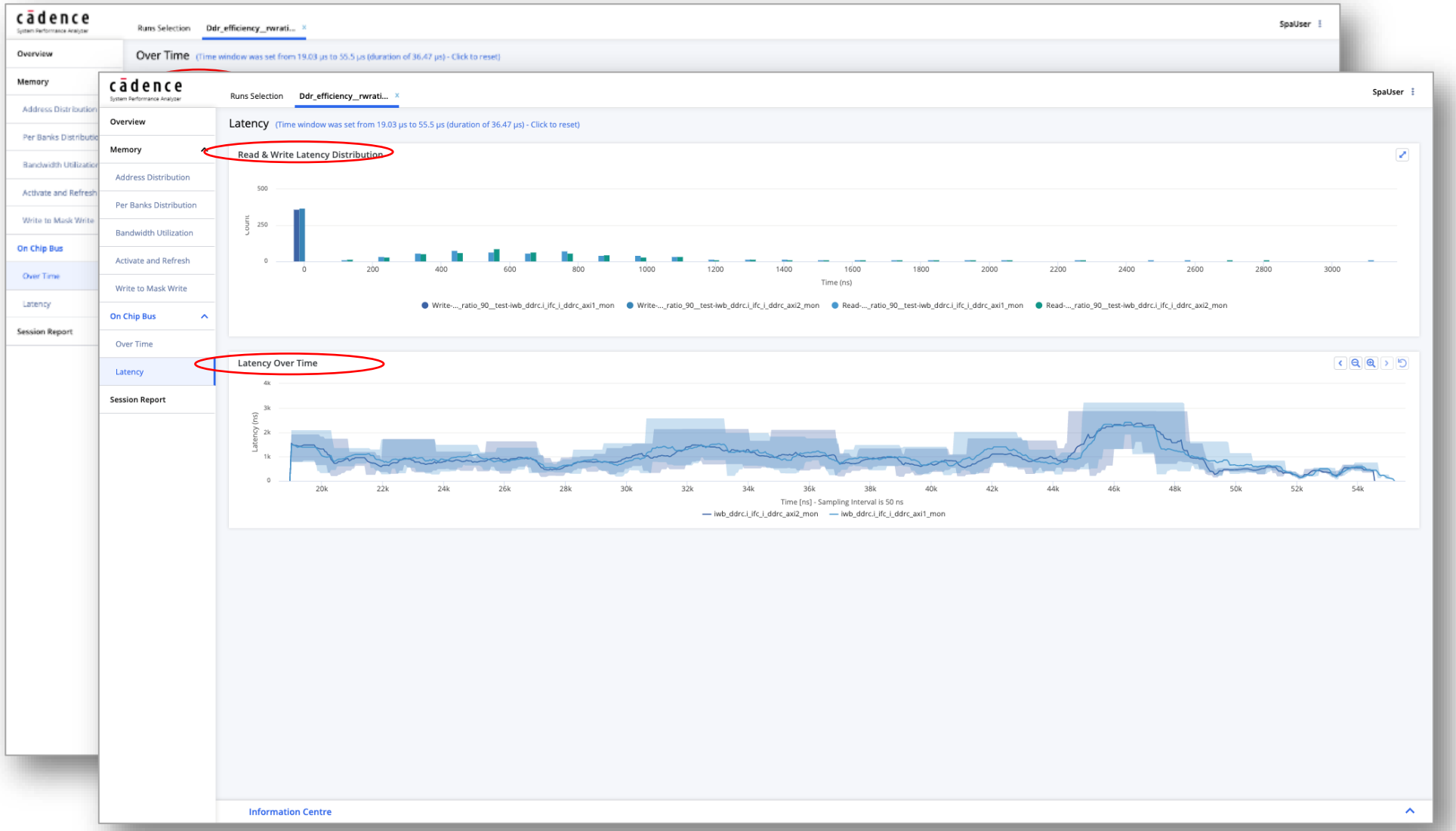


- Many steps in path from initiator to target, there are many potential ”throttle” points
- Understanding and debugging bandwidth throttling requires visualization of “throttle” proxies
  -Outstanding Transaction (OT) count is the standard measure
- Localization of cache lines can have a huge impact on latency and reduce die-to-die traffic (which itself can be a throttle)
- Data packet assembly and decomposition across the Chiplet Interconnects can add significant overhead

Solution- Complete System Performance & Integrity Checking



Identifying performance bottlenecks



Advantages

- Quickly understand the relationship between:
  - Bandwidth Over Time
  - Latency Over Time
  - Outstanding Transaction Over Time
- Allows bottlenecks to be identified and investigated
- Quickly identify outlier transactions with high latency and investigate the time period of occurrence

Summary

- **UCIe based Chiplets verification has multiple levels and challenges**
  - Unit-level detailed verification with multi, higher-level protocols
  - Early verification of sub-system before full design availability
  - System level cache coherency and performance measurement and debugging
- **Each verification challenge requires a tailored solution strategy**
  - IP level requires flexible VIP with support of multiple topologies/interfaces.
  - Sub-system level requires emulation-friendly tb with cache coherency-aware scoreboard.
  - System level requires stimulus that can be injected through the core/VIP/AVIP.
- **Technology and tools are available now!**
  - Simulation and Emulation ready Verification IP
  - Multi-engine performance Analyzer to identify Bottlenecks, Latency and bandwidth.
  - Die-to-Die Scoreboard for cache and data integrity.
  - Portable Stimulus tests that can be run on multiple engines and topologies.