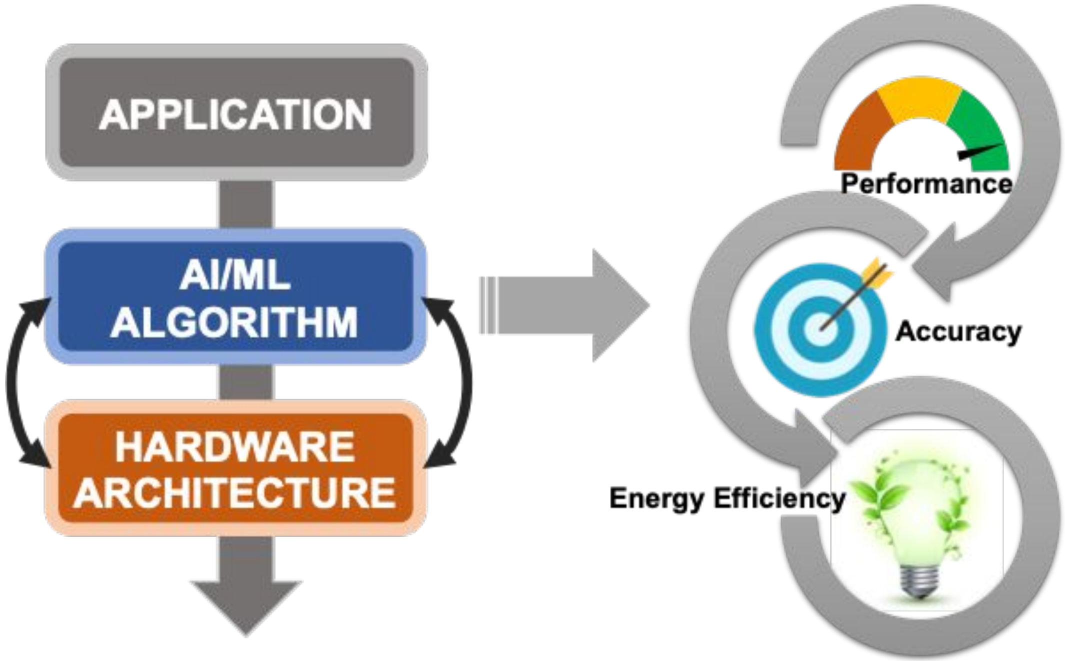
SystemC/HLS Prototyping of UCIe Interfaces

Adam Alhousiki, Thierry Tambe

December 3, 2025

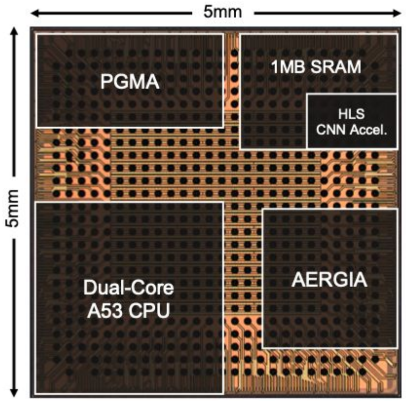
Stanford | ENGINEERING
Electrical Engineering

HW/SW Co-Design of AI workloads

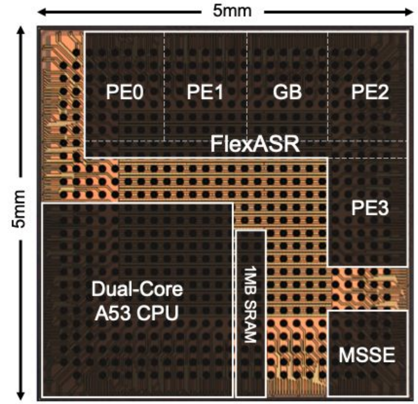


Chip Tapeouts of HLS-based Embedded ML Systems

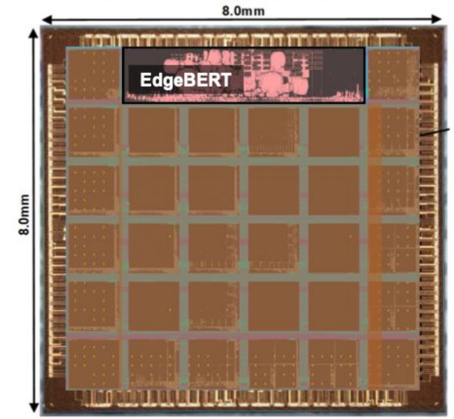
Unsupervised Probabilistic Perception. TSMC16. VLSI'20, HotChips'20



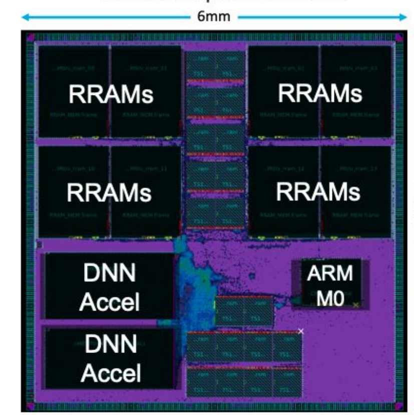
Inference of Attention-based DNNs w/ Adaptive Floating-Point. TSMC16. ISSCC'21, JSSC'22



Inference of Transformer Workloads w/ Mixed-Precision, and Early Exit. GF12. ISSCC'23



DNN Inference with ReRAMs TSMC40. Tape-out date: TBD

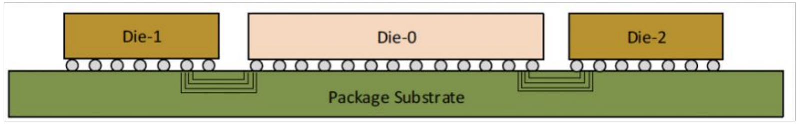


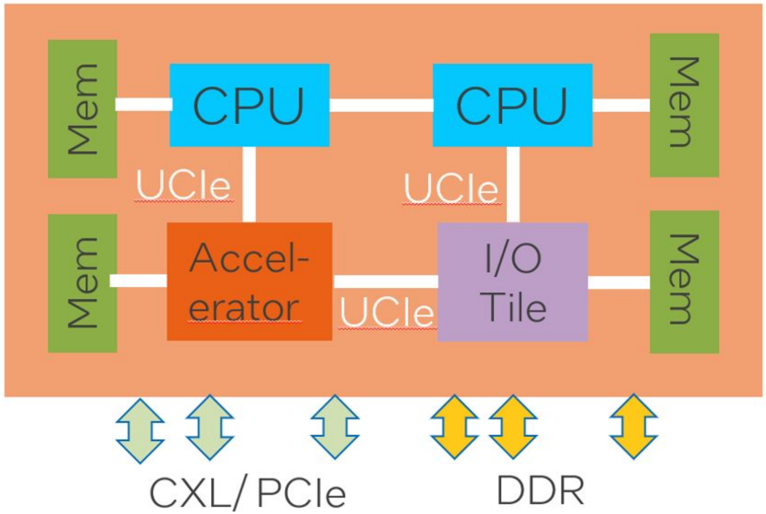
MADSIM: Multi-chiplet Agile Design and SIMulation

Objective: Developing a framework for designing and co-simulating multi-chiplet packages following **UCIe** (Universal Chiplet Interconnect Express) for chiplet interconnects.

Tools: Leveraging **Coside** with **SystemC** and **SystemC AMS** (for physical layer interconnects).

Impact: Aiming to create a reusable, modular, and open-source library for multi-chiplet design that accelerates chiplet-based system design, verification, and testing.

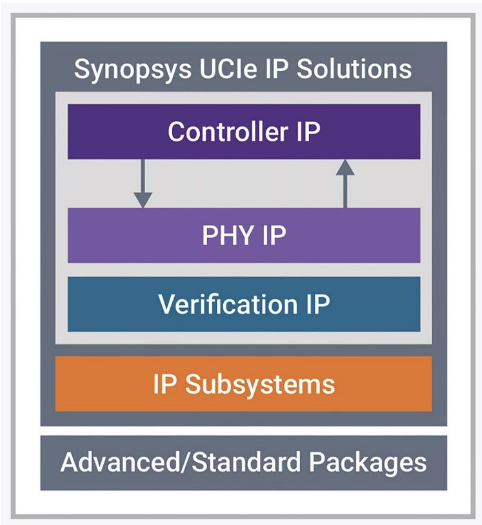




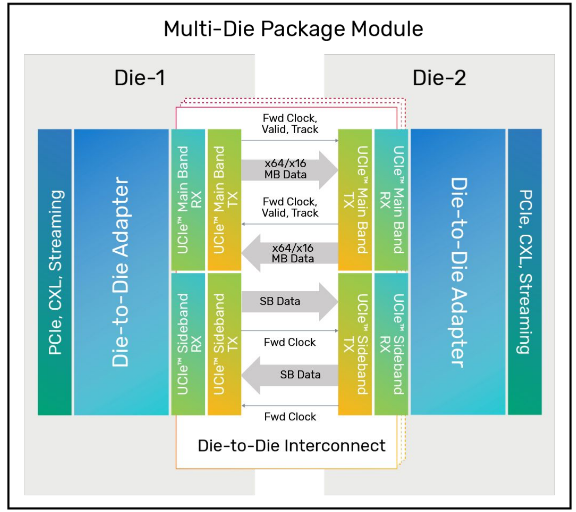
UCIe as communication protocol between various chiplets [1]

Source: [1] UCIe Consortium, *UCIe Specification 2.0*, Universal Chiplet Interconnect Express, 2024.

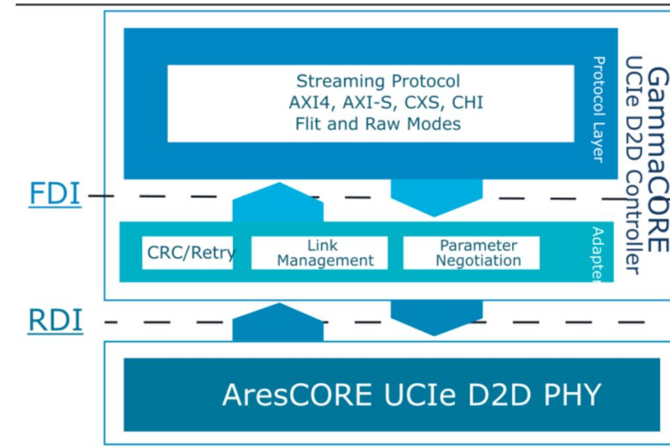
Examples of UCIe IPs



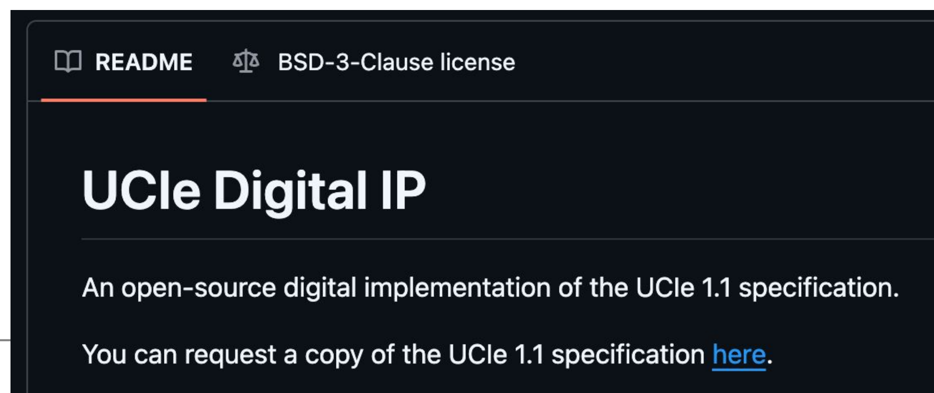
Synopsys



Cadence



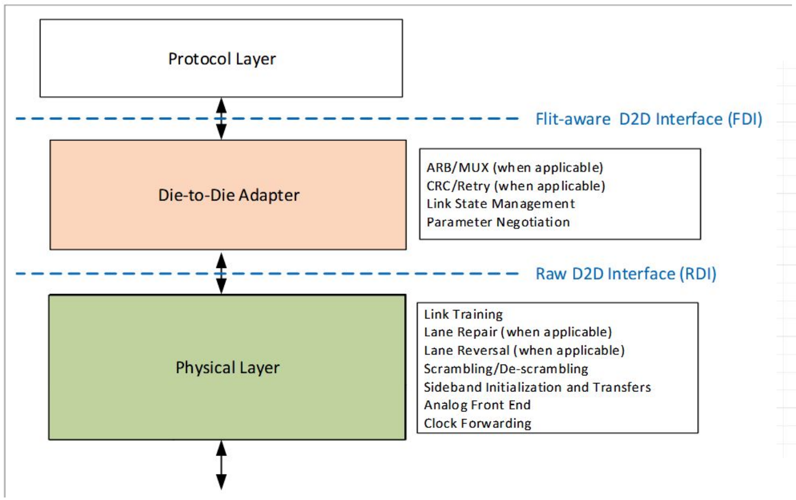
Alphawave Semi

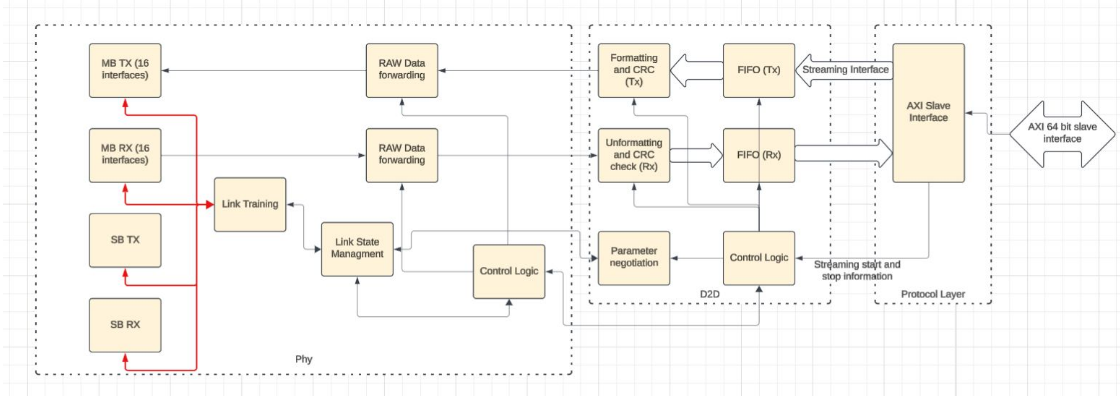


UC Berkeley
(designed in Scala)

UCIe Layered Architecture

UCIe Layered Architecture





Protocol Layer

Supports multiple protocols:

- PCIe (Flit and Non-Flit mode)
- CXL
- Streaming Protocol (User Defined)

Responsible for:

- Formatting data for transfer over FDI
- Following negotiated flit format rules
- Driving protocol-specific headers, CRC, and marker bits where required

Protocol Layer

Currently supports the following formats:

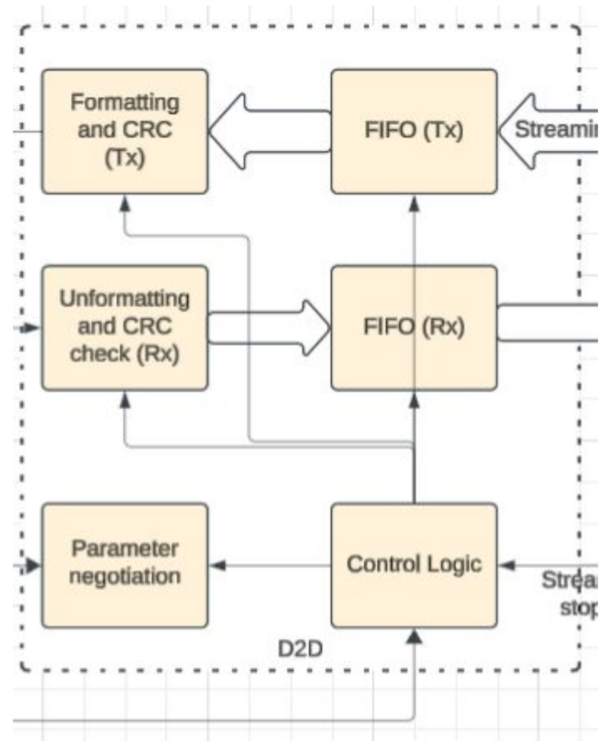
- **RAW Streaming Format**
  – 64 B frames direct from AXI (8 × 64-bit words)
  – No header or CRC (minimal overhead, simple mapping from AXI)
  – Best for low-latency, high-throughput, intra-package streaming

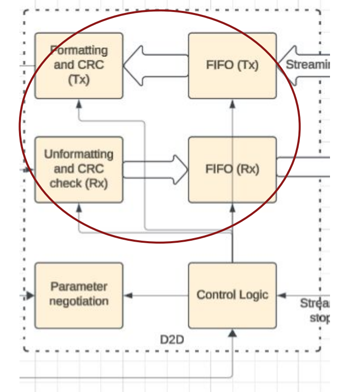
- **UCIe 68B Flit Format (No-Retry)**
  – 2 B header + 64 B payload + 2 B CRC (UCIe-compliant)
  – Supports Protocol ID, Stack ID, Pause Data Stream (PDS)
  – Better interoperability, error detection via CRC-16

Die-to-Die Layer

Responsibilities:

- Bridges the Protocol Layer to the physical chip-to-chip interface.
- Implements format-specific packetization (Streaming, 68B Flit).
- Handles headers, CRC, and special control flits (PDS, NOP).
- Performs byte-wise serialization for PHY and reassembly on receive.





D2D RX & TX

Transmit Path (TX):

- Converts Protocol Layer payload into PHY layer read format
- Does Format selection, Header generation, CRC computation, Immediate PDS insertion when idle or commanded, Serialize to 1B/cycle for PHY

Receive Path (RX):

- Reassembles incoming byte stream into frames/flits and validates them
- Does Byte deserialization, header parsing, CRC check, Discard PDS and NOP flits, deliver valid payload in 64-bit AXI words to protocol layer, maintain 256B alignment.

Parameter Negotiation

Purpose:

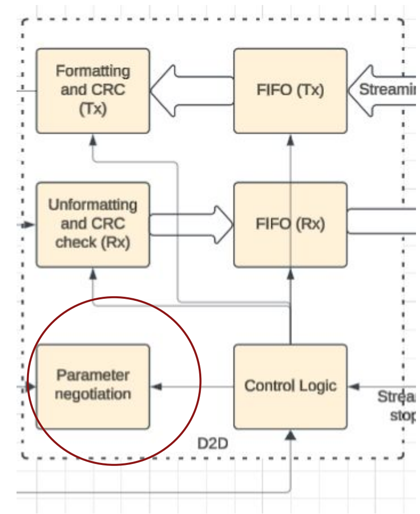
- Ensure both dies agree on protocol mode, flit format, and feature set before data transfer

Negotiation Parameters:

- Supported flit formats
- Protocol Type
- Stack ID and lane mapping

Process:

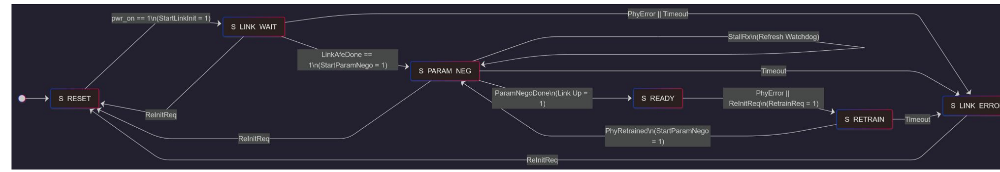
- Run after Link Training completes
- Exchange capabilities over Sideband or initial Mainband.
- Program D2D TX/RX with agreed settings.
- Select common operational mode.



D2D Controller

Purpose:

- Orchestrates the D2D link bring-up, mode switching and error handling.



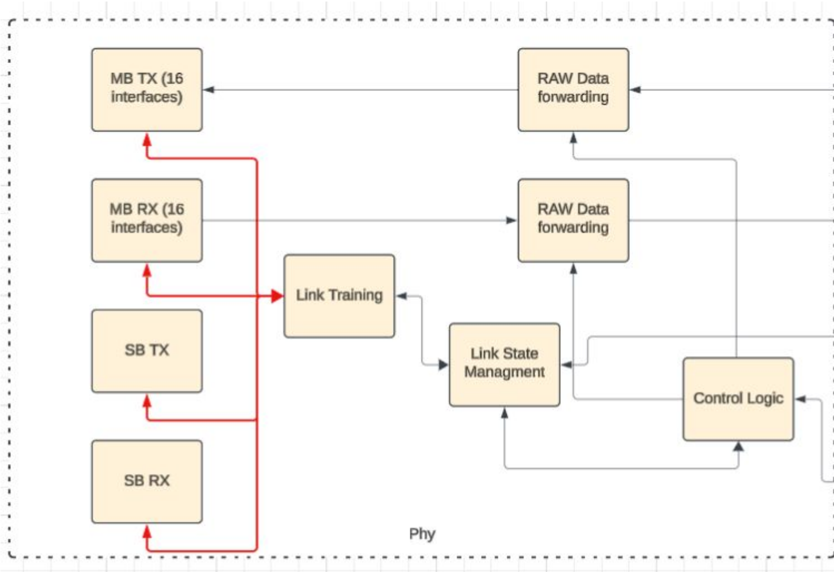
Key Interface Signals (SystemC)

- Control Outputs:
  - StartLinkInit → Triggers PHY RDI bring-up.
  - StartParamNego → Kicks off the Sideband AdvCap exchange.
  - FlitFormatNegotiated → Signal to PHY to enable Scrambler (per UCIe §9.3).
- Status Inputs:
  - LinkAfeDone ← Indicates RDI is active and physical link is stable.
  - ParamNegoDone ← Indicates successful FinCap handshake.
  - StallRx ← Handles remote partner firmware download delays.

- Modular Design: Decoupled from PHY complexity for better reuse.

PHY Layer - Digital

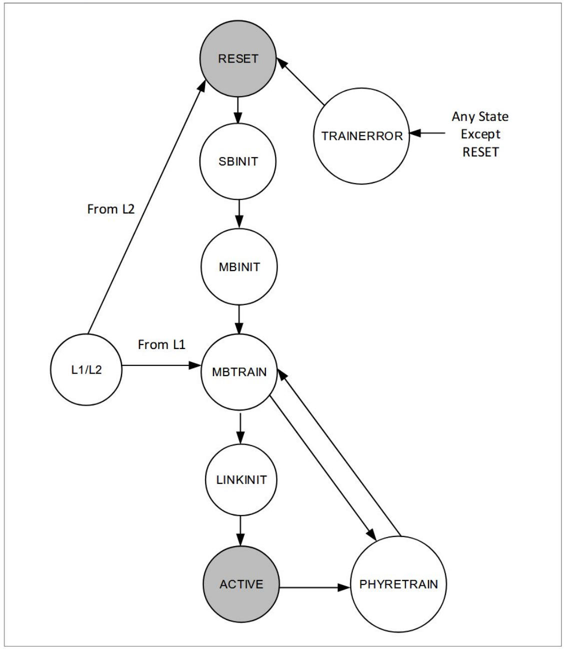
- Terminates the FDI at Physical level
- Provides the electrical, timing, and link training foundation for the protocol & adapter layers above.
- Responsible for link bring-up, maintenance, and teardown.

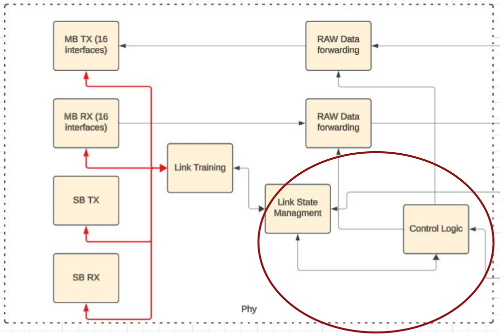


PHY Controller (Control Logic + LSM)

- RDI State Machine (RESET, ACTIVE, L1/L2, RETRAIN, ERROR)
- Sideband Protocol Control, Mailbox messaging
- Scrambler enable, timer, watchdogs
- L1/L2 = Low-power entry/exit sequences with clock gating and wake handshakes

Link Training State Machine





Link Training

- Orchestrates bring-up of the mainband lanes so they can reliably carry UCIe flits.
- Validated full Link Training State Machine (LTSM) traversal: RESET → SBINIT → MBINIT → MBTRAIN → ACTIVE.
- Verified correct execution of the Sideband Protocol handshake (REQ/RSP) for parameter negotiation and calibration.
- Total Link-Up Time: 4.45 ms (from cold boot to ACTIVE).
- The training sequence completes well within the 8 ms Watchdog Timeout window mandated by the UCIe Specification, ensuring robust startup.

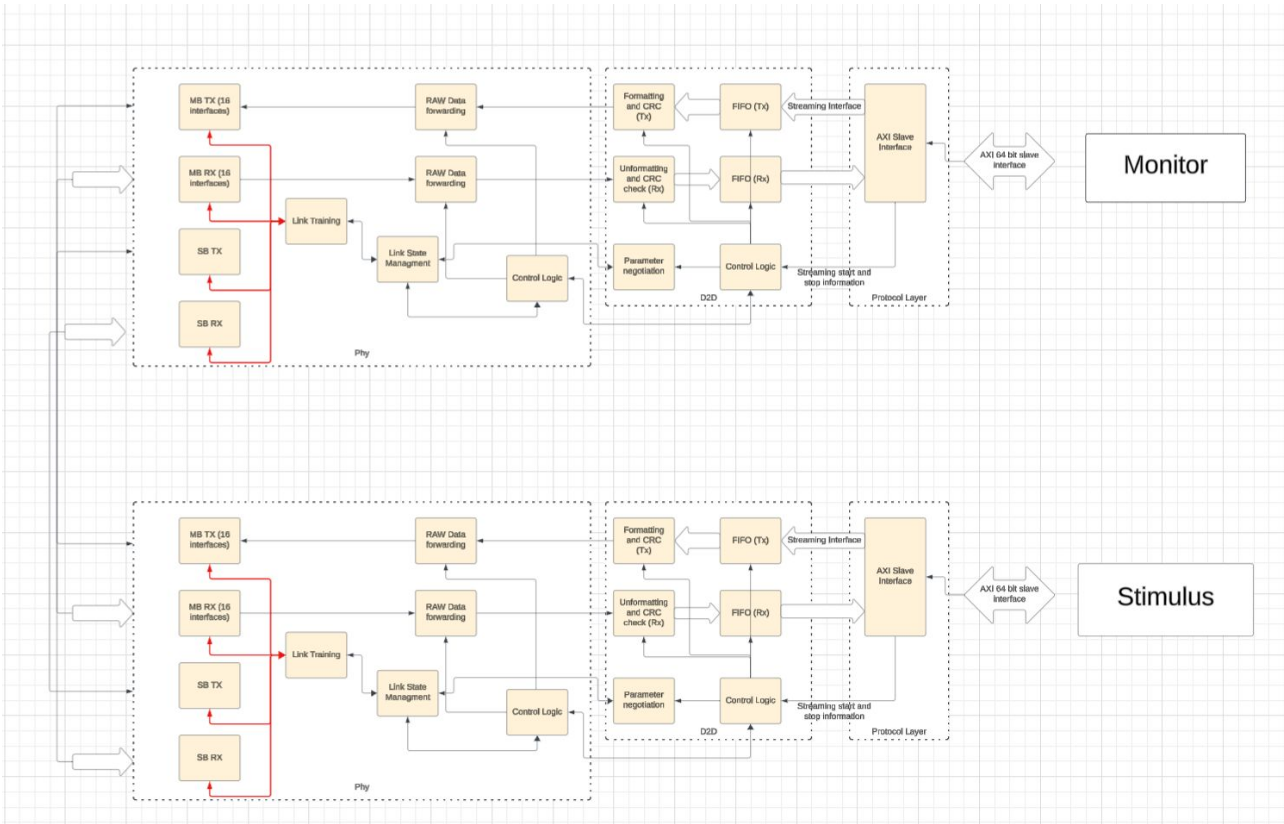
```
==================================
Connections Clock: tb.clk Period: 1 ns
1 ns  LTSM=0
1 ns  TB  Kick LT
0.00 ms | LTSM: RST → SB_PAT_TX
4 ns  LTSM=10
@100 ns  Assert RST
@4000100 ns  Deassert RST
4000101 ns  LTSM=0
4000101 ns  TB  Kick LT
4.00 ms | LTSM: RST → SB_PAT_TX
4000104 ns  LTSM=10
4.13 ms | LTSM: SB_PAT_TX → SB_PAT_WAIT
4.13 ms | LTSM: SB_PAT_WAIT → SB_START
4128104 ns TB Received: {c:0x10 s:0x0 i:0x0 d:0x0}
4129104 ns TB Sending: {c:0x11 s:0x0 i:0x0FFFF d:0x0}
4.13 ms | LTSM: SB_START → SB_DONE
4129106 ns TB Received: {c:0x12 s:0x0 i:0x0 d:0x0}
4130106 ns TB Sending: {c:0x13 s:0x0 i:0x0FFFF d:0x0}
4.13 ms | LTSM: SB_DONE → MB_PARAM
4130109 ns  LTSM=11
4130109 ns TB Received: {c:0x20 s:0x0 i:0x0 d:0x42}
4131109 ns TB Sending: {c:0x21 s:0x0 i:0x0FFFF d:0x102}
4131109 ns TB MB_CMD: 5
4.13 ms | LTSM: MB_PARAM → MB_CAL
4132112 ns  LTSM=12
4132112 ns TB Received: {c:0x22 s:0x0 i:0x0 d:0x0}
4133112 ns TB Sending: {c:0x23 s:0x0 i:0x0FFFF d:0x0}
4133112 ns TB MB_CMD: 6
4.23 ms | LTSM: MB_CAL → MB_REV_INIT
4233115 ns  LTSM=13
4233115 ns TB Received: {c:0x24 s:0x0 i:0x0 d:0x0}
4234115 ns TB Sending: {c:0x25 s:0x0 i:0x0FFFF d:0x0}
4234115 ns TB MB_CMD: 3
4.24 ms | LTSM: MB_REV_INIT → MB_REV_WAIT
4235118 ns TB Received: {c:0x26 s:0x0 i:0x0 d:0x0}
4236118 ns TB Sending: {c:0x27 s:0x0 i:0x0FFFF d:0x0}
4.24 ms | LTSM: MB_REV_WAIT → MB_REV_DONE
4236120 ns TB Received: {c:0x28 s:0x0 i:0x0 d:0x0}
4237120 ns TB Sending: {c:0x29 s:0x0 i:0x0FFFF d:0x0}
4.24 ms | LTSM: MB_REV_DONE → MB_SIDLE
4237123 ns  LTSM=20
4237123 ns TB Received: {c:0x30 s:0x0 i:0x0 d:0x0}
4238123 ns TB Sending: {c:0x31 s:0x0 i:0x0FFFF d:0x0}
4238123 ns TB MB_CMD: a
4.24 ms | LTSM: MB_SIDLE → MB_RXCLK
4239126 ns  LTSM=23
4239126 ns TB Received: {c:0x36 s:0x0 i:0x0 d:0x0}
4240126 ns TB Sending: {c:0x37 s:0x0 i:0x0FFFF d:0x0}
4240126 ns TB MB_CMD: b
4.24 ms | LTSM: MB_RXCLK → MB_VALVREF
4241129 ns  LTSM=21
4241129 ns TB Received: {c:0x32 s:0x0 i:0x0 d:0x0}
4242129 ns TB Sending: {c:0x33 s:0x0 i:0x0FFFF d:0x0}
4242129 ns TB MB_CMD: 6
4.34 ms | LTSM: MB_VALVREF → MB_DATAVREF
4342132 ns  LTSM=22
4342132 ns TB Received: {c:0x34 s:0x0 i:0x0 d:0x0}
4343132 ns TB Sending: {c:0x35 s:0x0 i:0x0FFFF d:0x0}
4343132 ns TB MB_CMD: 7
4.44 ms | LTSM: MB_DATAVREF → MB_VALCENTER
4443135 ns  LTSM=24
4443135 ns TB Received: {c:0x38 s:0x0 i:0x0 d:0x0}
4444135 ns TB Sending: {c:0x39 s:0x0 i:0x0FFFF d:0x0}
4444135 ns TB MB_CMD: 8
4.45 ms | LTSM: MB_VALCENTER → MB_LSPEED
4445138 ns  LTSM=25
4445138 ns TB Received: {c:0x3A s:0x0 i:0x0 d:0x0}
4446138 ns TB Sending: {c:0x3B s:0x0 i:0x0FFFF d:0x0}
4446138 ns TB MB_CMD: f
4.45 ms | LTSM: MB_LSPEED → LINKINIT
4447141 ns  LTSM=16
4447141 ns TB Received: {c:0x40 s:0x0 i:0x0 d:0x0}
4448141 ns TB Sending: {c:0x41 s:0x0 i:0x0FFFF d:0x0}
4.45 ms | LTSM: LINKINIT → ACTIVE
4448143 ns  TB  DONE
4448144 ns  LTSM=17

Info: /OSCI/SystemC: Simulation stopped by user.
=== SIM DONE ===
```

Table 7-9. Link Training State Machine related Message encodings for messages without data (Sheet 1 of 4)

Message	MsgInfo[15:0]	MsgCode[7:0]	MsgSubcode[7:0]
{Start Tx Init D to C point test resp}	0000h	8Ah	01h
{LFSR_clear_error req}	0000h	85h	02h
{LFSR_clear_error resp}	0000h	8Ah	02h
{Tx Init D to C results req}	0000h	85h	03h
{End Tx Init D to C point test req}	0000h	85h	04h
{End Tx Init D to C point test resp}	0000h	8Ah	04h
{Start Tx Init D to C eye sweep resp}	0000h	8Ah	05h
{End Tx Init D to C eye sweep req}	0000h	85h	06h
{End Tx Init D to C eye sweep resp}	0000h	8Ah	06h
{Start Rx Init D to C point test resp}	0000h	8Ah	07h
{Rx Init D to C Tx Count Done req}	0000h	85h	08h
{Rx Init D to C Tx Count Done resp}	0000h	8Ah	08h
{End Rx Init D to C point test req}	0000h	85h	09h
{End Rx Init D to C point test resp}	0000h	8Ah	09h
{Start Rx Init D to C eye sweep resp}	0000h	8Ah	0Ah
{Rx Init D to C results req}	0000h	85h	0Bh
{End Rx Init D to C eye sweep req}	0000h	85h	0Dh
{End Rx Init D to C eye sweep resp}	0000h	8Ah	0Dh
{SBINIT out of Reset}	[15:4] : Reserved [3:0] : Result[a]	91h	00h
{SBINIT done req}	0000h	95h	01h
{SBINIT done resp}	0000h	9Ah	01h
{MBINIT.CAL Done req}	0000h	A5h	02h
{MBINIT.CAL Done resp}	0000h	AAh	02h
{MBINIT.REPAIRCLK init req}	0000h	A5h	03h
{MBINIT.REPAIRCLK init resp}	0000h	AAh	03h
{MBINIT.REPAIRCLK result req}	0000h	A5h	04h
{MBINIT.REPAIRCLK result resp}	[15:4]: Reserved [3]: Compare Results from RRDCK_L [2]: Compare Results from RTRK_L [1]: Compare Results from RCKN_L [0]: Compare Results from RCKP_L	AAh	04h
{MBINIT.REPAIRCLK apply repair req}	[15:4]: Reserved [3:0]: Repair Encoding Fh: Reserved 0h: Repair RCLKP_L 1h: Repair RCLKN_L 2h: Repair RTRK_L 7h: Reserved	A5h	05h
{MBINIT.REPAIRCLK apply repair resp}	0000h	AAh	05h

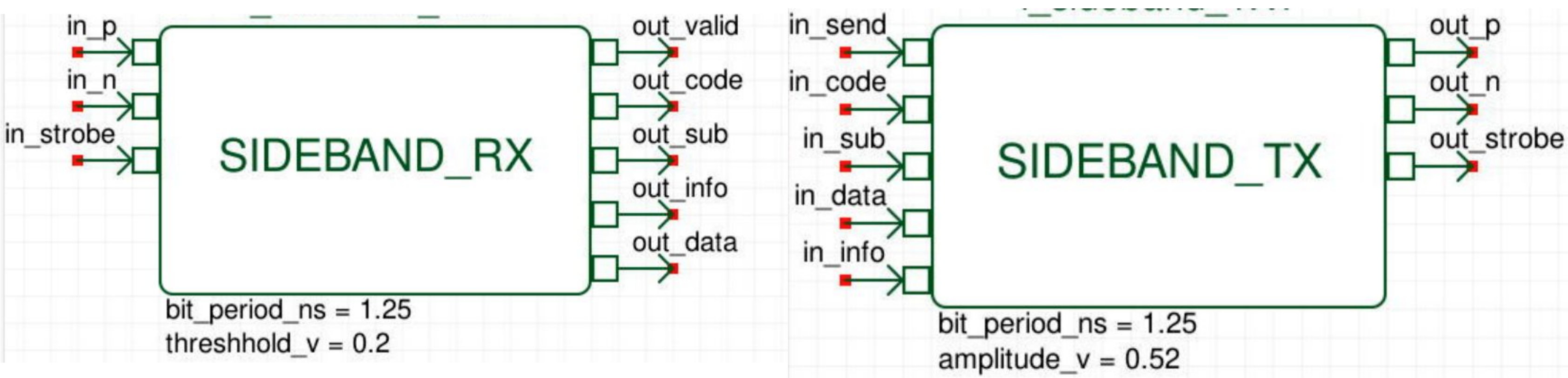
SystemC AMS Simulation

Full System Integration



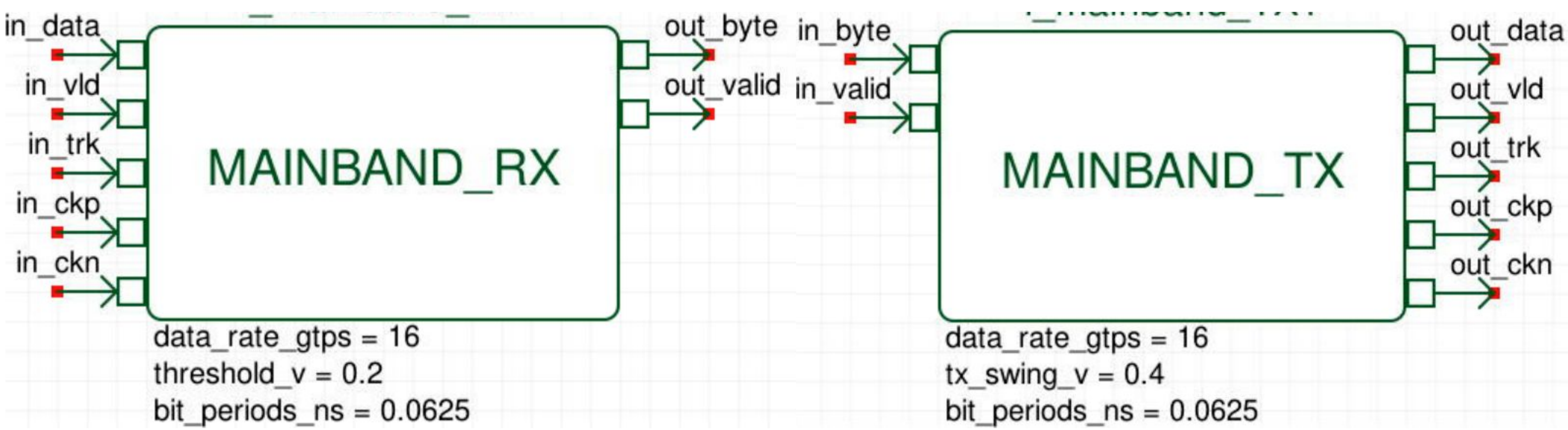
RX + TX Sideband

- The Sideband is a dedicated low-speed communication channel in UCIe that operates separately from the high-speed Mainband data lanes. Think of it as the "control plane" while Mainband is the "data plane."
- It Receives and deserializes control messages from the remote die, extracting opcodes and payloads for the PHY controller.

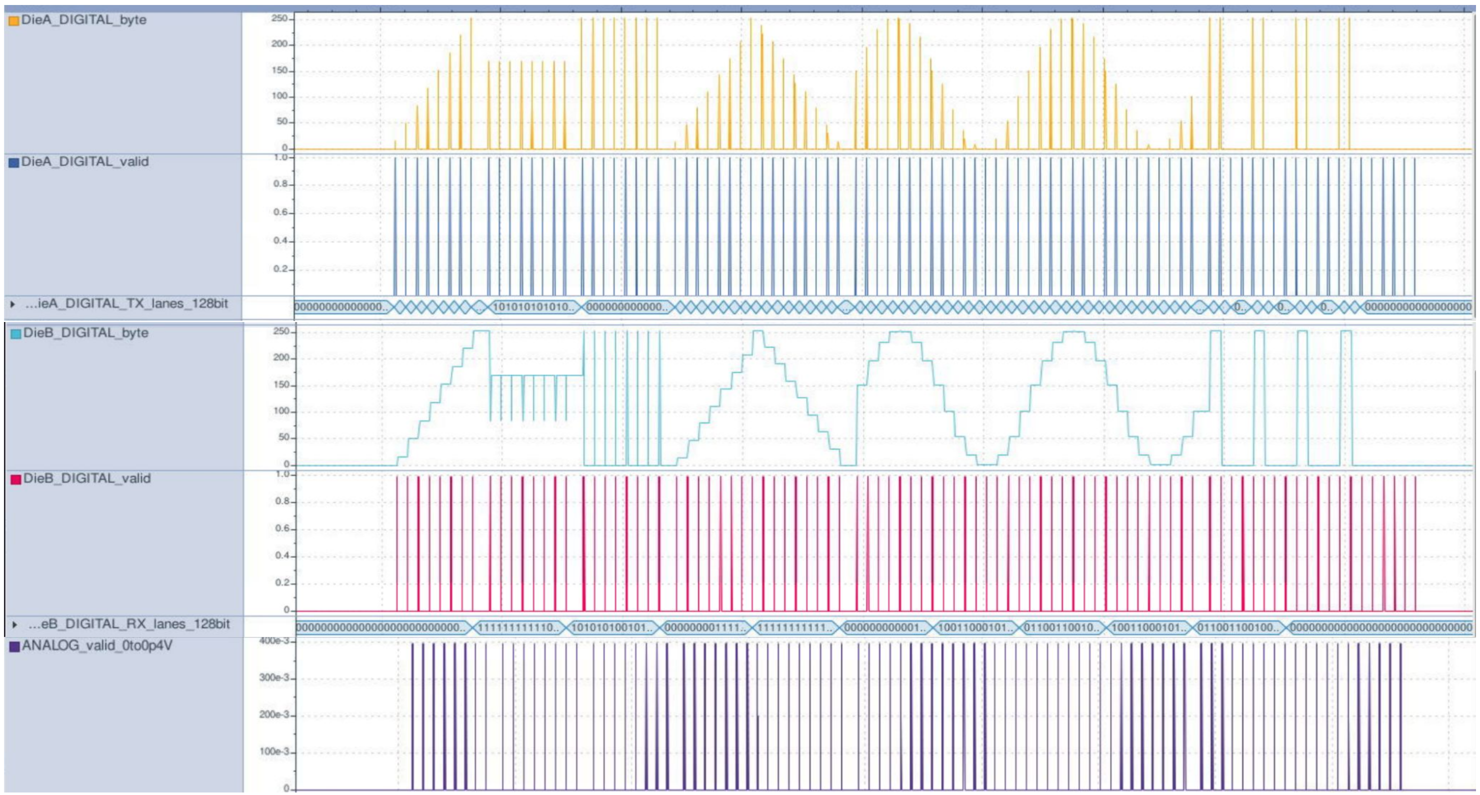


RX + TX Mainband

- The Mainband is the high-speed data path in UCIe that carries user data (flits) between chiplets. It operates at 4-32 GT/s using 16 parallel lanes in the Standard Package configuration.
- Serializes parallel byte data into high-speed serial streams on the physical lanes, with clock forwarding and valid framing per UCIe specification.



Waveform Results



Waveform Results

- We used different test patterns including: incrementing, checkerboard, all 1s/0s, triangle wave, sine wave, pulse train.
- For the incrementing pattern for example, the MB_TX sends bytes at different time staps: [404 ns] TX[1]: 0x011, [406 ns] TX[2]: 0x022….., and then we receive it at the MB_RX as a 16-byte accumulated word: [551 ns] RX_128bit_LANES: 0x0FFEEDDCCBBAA99887766554433221100

System Latency

[4.450100000 ms] MB_TX: byte=0x11 bit=0 val=1 level=0.4

[4.450100062 ms] MB_TX: byte=0x11 bit=1 val=0 level=0

[4.450100125 ms] MB_TX: byte=0x11 bit=2 val=0 level=0

...

[4.450101900 ms] MB_RX: BYTE COMPLETE = 0x11

- **Specification Compliance:** According to UCIe Specification Rev 2.0 (Table 1-4), the maximum latency target for the Adapter and Physical Layer in a Standard Package is **≤2.0 ns**.
- **Calculated Latency:** The timestamp delta (4.450102000 ms−4.450100000 ms) confirms a total system latency of **1.9 ns**, meeting the strict timing budget for short channel reach.

Thank you!