SYNOPSYS®

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
JAPAN

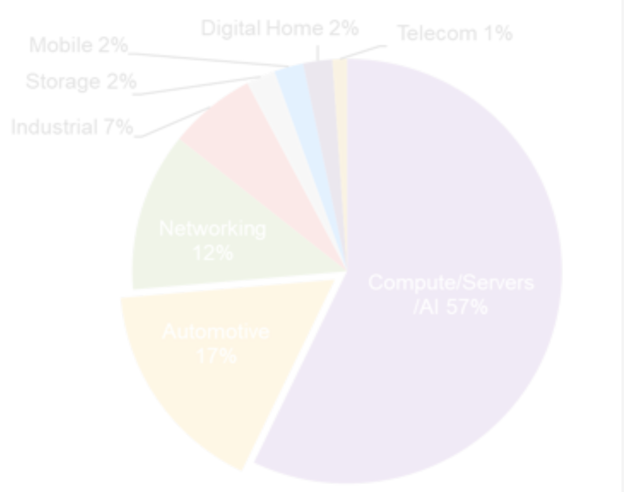
Conquering UCIe 1.1 Multi-die System Verification Challenges

accellera
SYSTEMS INITIATIVE

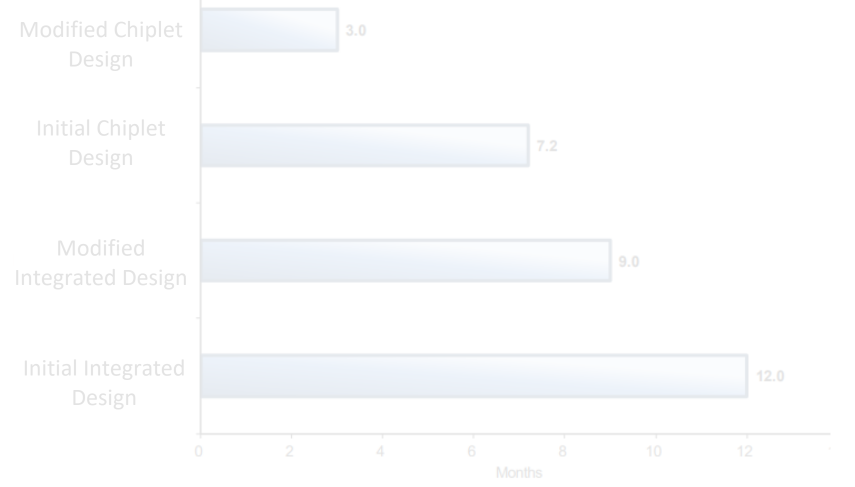
Interface Verification Critical for Chiplet and UCIe Adoption

Die-to-Die IP design starts to grow 5X in 5 years***

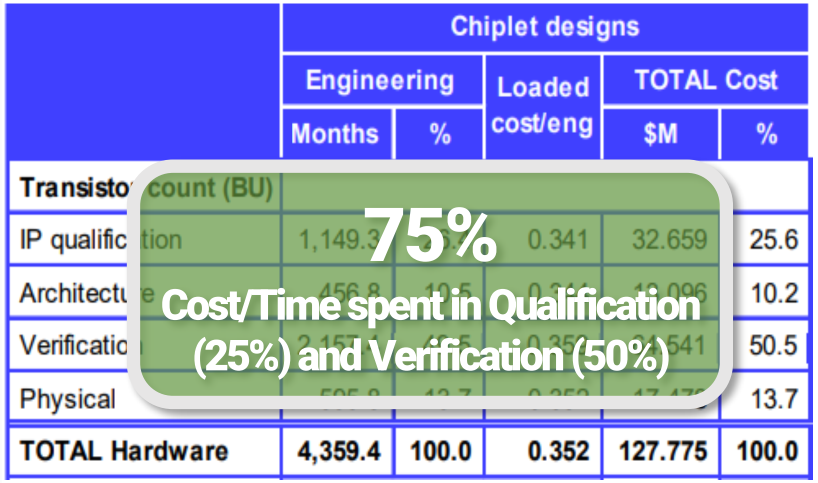
Design Start by Application



Time to Design (Integrated vs Chiplet)



Cost and Effort Spent across Functions



	Chiplet designs				
	Engineering		Loaded cost/eng	TOTAL Cost	
	Months	%		$M	%
Transistor count (BU)					
IP qualification	1,149.3	[illegible]	0.341	32.659	25.6
Architecture	456.8	10.5	[illegible]	[illegible]	10.2
Verification	[illegible]	[illegible]	[illegible]	[illegible]	50.5
Physical	[illegible]	[illegible]	[illegible]	[illegible]	13.7
TOTAL Hardware	4,359.4	100.0	0.352	127.775	100.0

75%

Cost/Time spent in Qualification (25%) and Verification (50%)

“ $50B in Chiplet revenue forecasted by 2024 ”

“Led by HPC & Automotive ”

“ The time-to-market may be 60% faster for initial chiplet designs compared to integrated designs ”

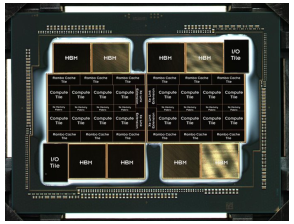
*“ The **highest-cost design step is functional verification**, and **emulation capabilities are very important** ”*

*Source: Gartner 2021 **Source: IBS 2023 ***Source: IP Nest 2022

Multi-Die System Examples

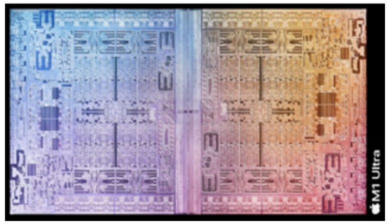
Enabled by More Cost-Effective Multi-Die Packaging Technologies!

INTEL - Ponte-Vechio Xe-HPC GPU



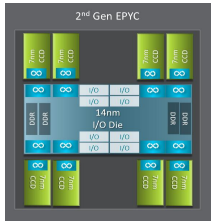
100B+ Tr’s, 47 Active Tiles, 5 Process Nodes, EMIB/Foveros

APPLE - M1 UltraMax CPU



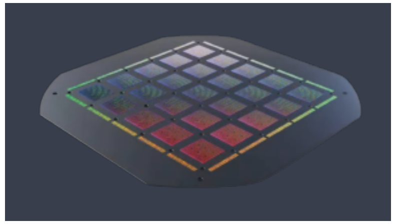
2x Dies, 114B Transistors, 2.5TB D2D BW, Silicon Connected

AMD - EPIC Server CPU



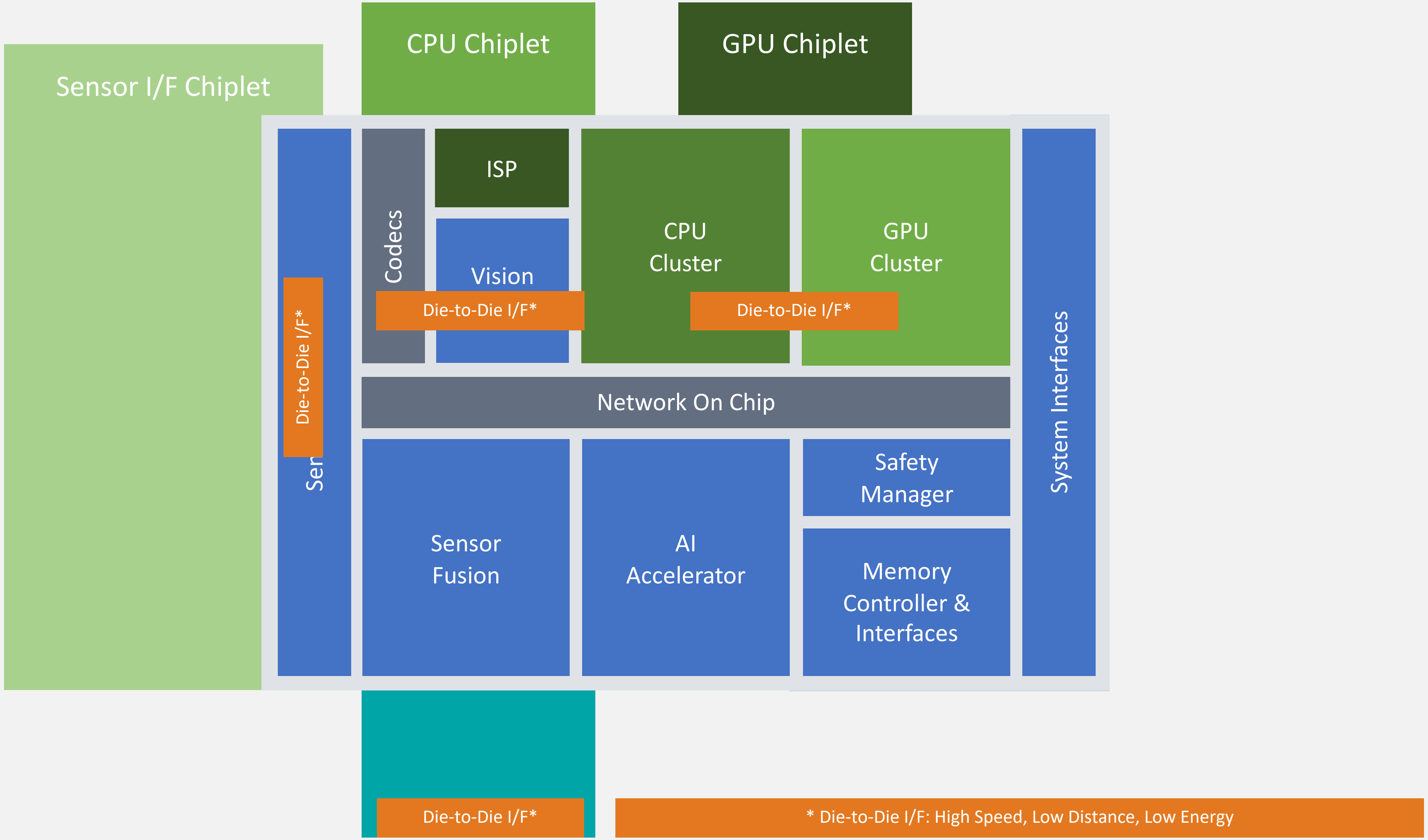
100B+ Tr’s, 47 Active Tiles, 5 Process Nodes, EMIB/Foveros

TESLA - 9 Peta Flops AI-Training



25x 50B Transistors, Reconstructed fanout wafer

Optimizing Architectural Functions into Chiplets



* Die-to-Die I/F: High Speed, Low Distance, Low Energy

UCIe 1.1 Focuses on Key Automotive Customer Asks !

Safety, Security, Reliability and Ecosystem Adoptions

- Preventive Monitoring
- On-field Repairability
- Ecosystem fit through multiple existing protocols
- More …

UCIe sees automotive chiplet group with latest specification

Business news | August 9, 2023

The UCIe Consortium has launched an automotive chiplet group alongside the public release of Universal Chiplet Interconnect Express specification version 1.1 specification.

UCIe 1.1 Verification Requirements

1 Preventive Monitoring

2 On-field Repairability

3 Ecosystem Adoption

4 Cost Optimization

5 Compliance Testing

Preventive Monitoring

- Link Health Monitoring Considerations
  - *Continuous Monitoring:* Per-lane eye-margin measurement and control to shift eye-margin
  - *Failure rate of link:* Periodic insertion of parity bytes
  - *PHYRETRAIN:* Retraining of link during runtime to ensure link safety

- Failure Detection Considerations
  - *Reporting of failure:* Standard UHM registers to log error along with timestamp
  - *Interrupt:* Indication to the system about link failure

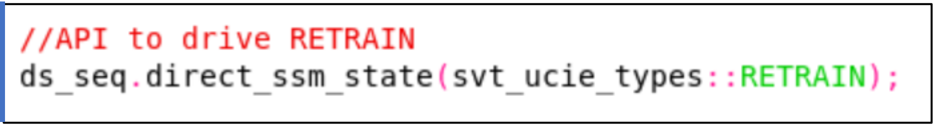
Preventive Monitoring – Synopsys UCIe VIP

Design and Verification considerations	VIP features
Continuous monitoring & reporting	• API • Control eye margins during LTSM states • Read/write UHM registers
Failure rate of link	• Parity computation and correction • Callback to inject parity error
PHYRETRAIN	• Local and remote Adapter and PHY initiated retrain • API • Control retrain pattern count • Corrupt retrain results • Forcibly move to retrain state
Interrupt	• API to enable/disable interrupts

Protocol checks to catch unexpected DUT behaviours

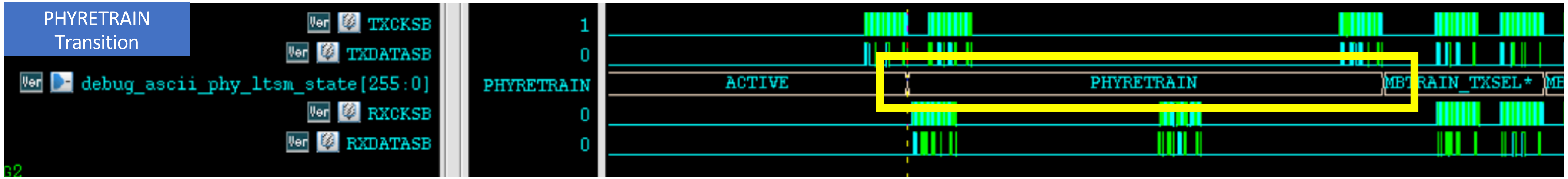
Preventive Monitoring – Synopsys UCIe VIP

API to initiate RETRAIN from Adapter layer

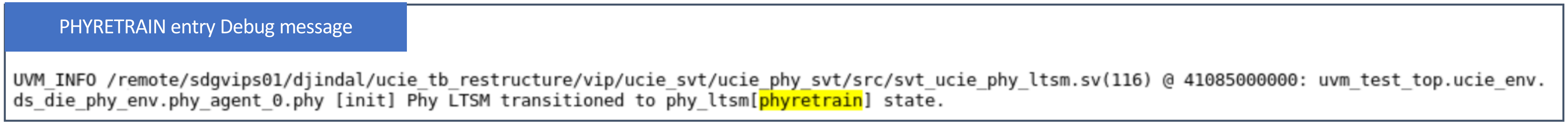


```
//API to drive RETRAIN
ds_seq.direct_ssm_state(svt_ucie_types::RETRAIN);
```

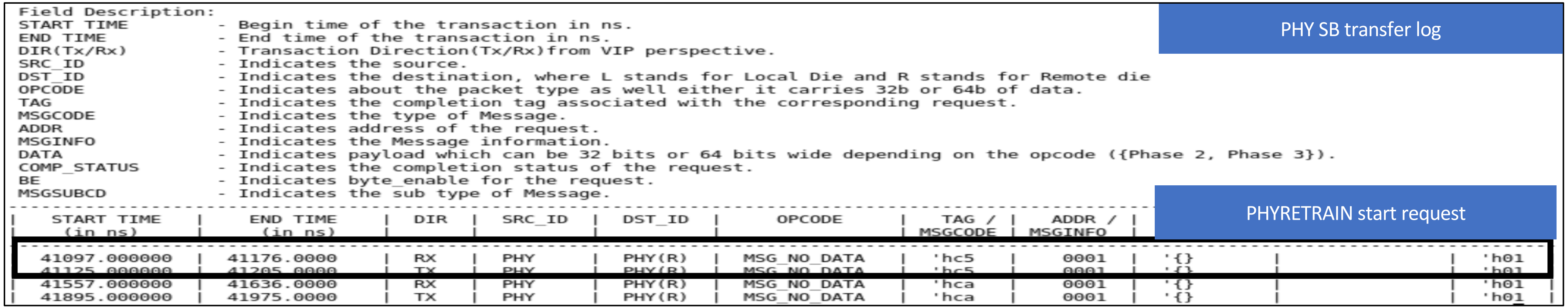
PHYRETRAIN Transition



PHYRETRAIN entry Debug message



```
UVM_INFO /remote/sdgvips01/djindal/ucie_tb_restructure/vip/ucie_svt/ucie_phy_svt/src/svt_ucie_phy_ltsm.sv(116) @ 41085000000: uvm_test_top.ucie_env.
ds_die_phy_env.phy_agent_0.phy [init] Phy LTSM transitioned to phy_ltsm[phyretrain] state.
```



PHY SB transfer log

PHYRETRAIN start request

```
Field Description:
START TIME          - Begin time of the transaction in ns.
END TIME            - End time of the transaction in ns.
DIR(Tx/Rx)          - Transaction Direction(Tx/Rx)from VIP perspective.
SRC_ID              - Indicates the source.
DST_ID              - Indicates the destination, where L stands for Local Die and R stands for Remote die
OPCODE              - Indicates about the packet type as well either it carries 32b or 64b of data.
TAG                 - Indicates the completion tag associated with the corresponding request.
MSGCODE             - Indicates the type of Message.
ADDR                - Indicates address of the request.
MSGINFO             - Indicates the Message information.
DATA                - Indicates payload which can be 32 bits or 64 bits wide depending on the opcode ({Phase 2, Phase 3}).
COMP_STATUS         - Indicates the completion status of the request.
BE                  - Indicates byte_enable for the request.
MSGSUBCD            - Indicates the sub type of Message.
-----------------------------------------------------------------------------------------------------------------------------------
| START TIME       |  END TIME      | DIR  | SRC_ID  | DST_ID   |    OPCODE      |  TAG /  |  ADDR /  |
|  (in ns)         |   (in ns)      |      |         |          |                | MSGCODE | MSGINFO  |
-----------------------------------------------------------------------------------------------------------------------------------
| 41097.000000     | 41176.0000     | RX   | PHY     | PHY(R)   | MSG_NO_DATA    | 'hc5    |  0001    | '{}  |          | 'h01 |
| 41125.000000     | 41205.0000     | TX   | PHY     | PHY(R)   | MSG_NO_DATA    | 'hc5    |  0001    | '{}  |          | 'h01 |
| 41557.000000     | 41636.0000     | RX   | PHY     | PHY(R)   | MSG_NO_DATA    | 'hca    |  0001    | '{}  |          | 'h01 |
| 41895.000000     | 41975.0000     | TX   | PHY     | PHY(R)   | MSG_NO_DATA    | 'hca    |  0001    | '{}  |          | 'h01 |
```

UCIe 1.1 Verification Requirements

1 Preventive Monitoring

2 On-field Repairability

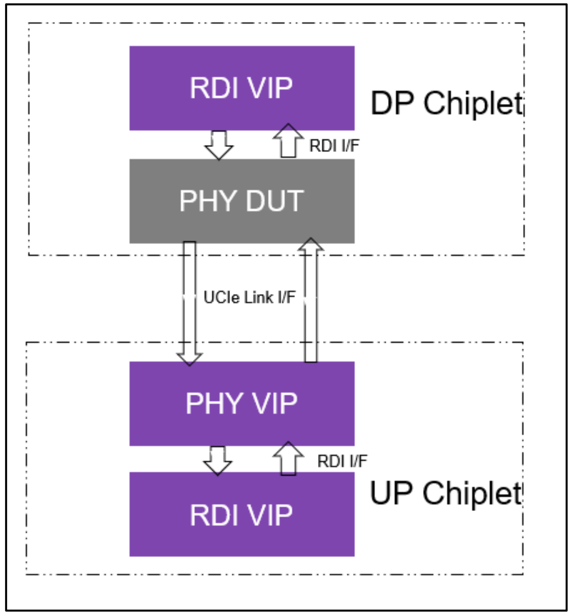
3 Ecosystem Adoption

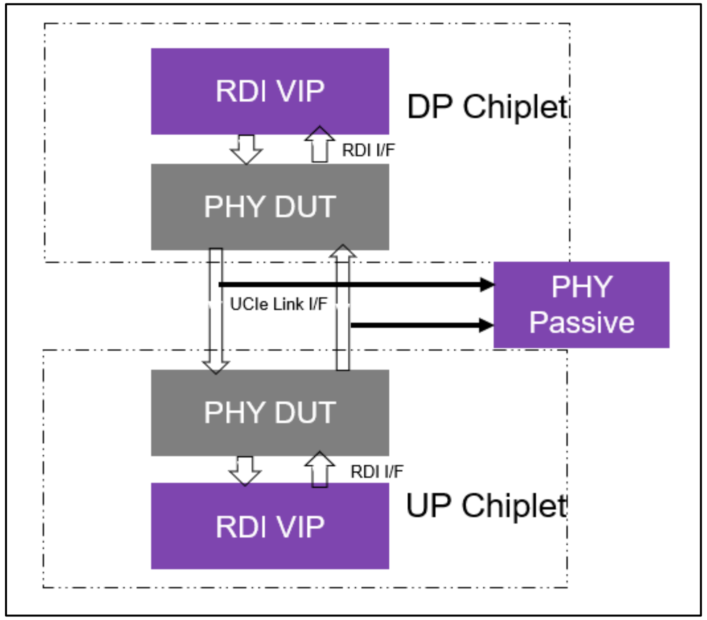
4 Cost Optimization

5 Compliance Testing

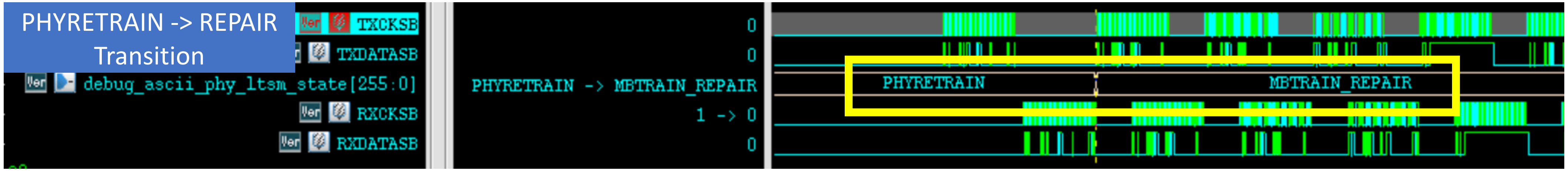
On-field Repairability

- Repairability Considerations
  - *Redundancy mapping:* Clock and valid lane mapping, single lane and two-lane data mapping
  - *REPAIR:* Usage of Redundant pins to repair clock, valid and data lanes
  - *TRAINERROR:* Repair of lanes is not feasible



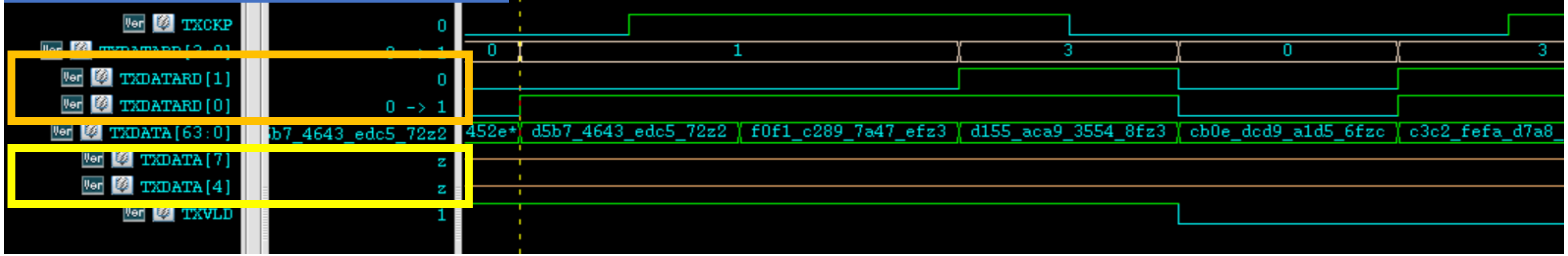


On-field Repairability – Synopsys UCIe VIP



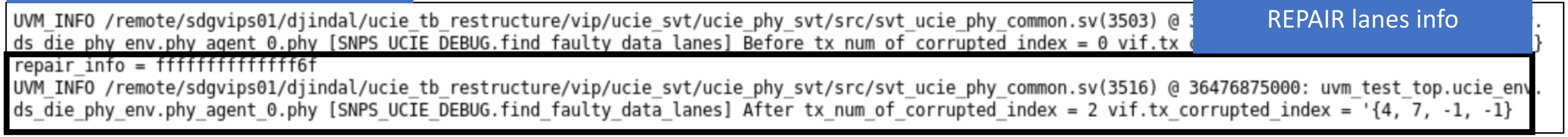
PHYRETRAIN -> REPAIR Transition

TXCKSB
TXDATASB
debug_ascii_phy_ltsm_state[255:0]
RXCKSB
RXDATASB
0
0
PHYRETRAIN -> MBTRAIN_REPAIR
1 -> 0
0
PHYRETRAIN
MBTRAIN_REPAIR



REPAIR lane 7 & 4 with RD 1 & 0

REPAIR Debug message



REPAIR lanes info

```
UVM_INFO /remote/sdgvips01/djindal/ucie_tb_restructure/vip/ucie_svt/ucie_phy_svt/src/svt_ucie_phy_common.sv(3503) @ 3
ds_die_phy_env.phy_agent_0.phy [SNPS_UCIE_DEBUG.find_faulty_data_lanes] Before tx_num_of_corrupted_index = 0 vif.tx
repair_info = ffffffffffffff6f
UVM_INFO /remote/sdgvips01/djindal/ucie_tb_restructure/vip/ucie_svt/ucie_phy_svt/src/svt_ucie_phy_common.sv(3516) @ 36476875000: uvm_test_top.ucie_env.
ds_die_phy_env.phy_agent_0.phy [SNPS_UCIE_DEBUG.find_faulty_data_lanes] After tx_num_of_corrupted_index = 2 vif.tx_corrupted_index = '{4, 7, -1, -1}
```

UCIe 1.1 Verification Requirements

1 Preventive Monitoring

2 On-field Repairability

3 Ecosystem Adoption

4 Cost Optimization

5 Compliance Testing

Ecosystem Adoption

- Streaming Protocol usage considerations
  - *Flit formats*: Usage of existing PCIe/CXL flit formats for various streaming protocol chiplets e.g. AXI, CHI, vendor defined etc.
  - *Features:* Use D2D features like CRC, Retry, parity, etc.

Flit Format Number	Flit Format Name	PCIe Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	Streaming	
					UCIe 1.0	UCIe 1.1
1	Raw	Optional	Optional	Optional	Mandatory	Mandatory
2	68B	N/A	Mandatory	N/A	N/A	Supported
3	Standard 256B End Header	Mandatory	N/A	N/A	N/A	Supported
4	Standard 256B Start Header	Optional	N/A	Mandatory	N/A	Supported
5	Latency Optimized 256B without optional Bytes	N/A	N/A	Optional	N/A	Supported
6	Latency Optimized 256B with optional Bytes	Strongly Recommended	N/A	Strongly Recommended	N/A	Supported

Ecosystem Adoption

- Streaming Protocol usage considerations
  - *Flit formats*: Usage of existing PCIe/CXL flit formats for various streaming protocol chiplets e.g. AXI, CHI, vendor defined etc.
  - *Features:* Use D2D features like CRC, Retry, parity, etc.

Flit Format Number	Flit Format Name	PCIe Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	Streaming	
					UCIe 1.0	UCIe 1.1
1	Raw	Optional	Optional	Optional	Mandatory	Mandatory
2	68B	N/A	Mandatory	N/A	N/A	Supported
3	Standard 256B End Header	Mandatory	N/A	N/A	N/A	Supported
4	Standard 256B Start Header	Optional	N/A	Mandatory	N/A	Supported
5	Latency Optimized 256B without optional Bytes	N/A	N/A	Optional	N/A	Supported
6	Latency Optimized 256B with optional Bytes	Strongly Recommended	N/A	Strongly Recommended	N/A	Supported

Ecosystem Adoption

- Streaming Protocol usage considerations
  - *Flit formats*: Usage of existing PCIe/CXL flit formats for various streaming protocol chiplets e.g. AXI, CHI, vendor defined etc.
  - *Features:* Use D2D features like CRC, Retry, parity, etc.

Flit Format Number	Flit Format Name	PCIe Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	Streaming	
					UCIe 1.0	UCIe 1.1
1	Raw	Optional	Optional	Optional	Mandatory	Mandatory
2	68B	N/A	Mandatory	N/A	N/A	Supported
3	Standard 256B End Header	Mandatory	N/A	N/A	N/A	Supported
4	Standard 256B Start Header	Optional	N/A	Mandatory	N/A	Supported
5	Latency Optimized 256B without optional Bytes	N/A	N/A	Optional	N/A	Supported
6	Latency Optimized 256B with optional Bytes	Strongly Recommended	N/A	Strongly Recommended	N/A	Supported

Ecosystem Adoption

- Streaming Protocol usage considerations
  - *Flit formats*: Usage of existing PCIe/CXL flit formats for various streaming protocol chiplets e.g. AXI, CHI, vendor defined etc.
  - *Features*: Use D2D features like CRC, Retry, parity, etc.

Flit Format Number	Flit Format Name	PCIe Flit Mode	CXL 68B Flit Mode	CXL 256B Flit Mode	Streaming	
					UCIe 1.0	UCIe 1.1
1	Raw	Optional	Optional	Optional	Mandatory	Mandatory
2	68B	N/A	Mandatory	N/A	N/A	Supported
3	Standard 256B End Header	Mandatory	N/A	N/A	N/A	Supported
4	Standard 256B Start Header	Optional	N/A	Mandatory	N/A	Supported
5	Latency Optimized 256B without optional Bytes	N/A	N/A	Optional	N/A	Supported
6	Latency Optimized 256B with optional Bytes	Strongly Recommended	N/A	Strongly Recommended	N/A	Supported

Ecosystem Adoption

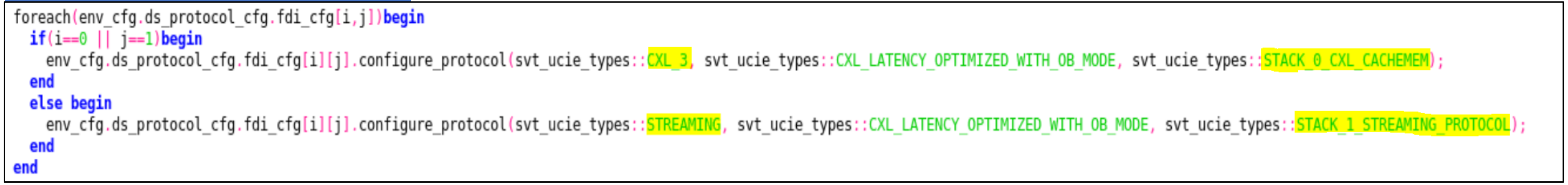
- Enhanced Protocol considerations
  - *Stack mux:* Muxing of different protocol and utilizing the bandwidth efficiently
  - *Arbitration:* Per flit arbitration to utilize 100% bandwidth and didn’t violate maximum percentage
- Link initialization considerations
  - *Initialization & Negotiation:* Negotiate and finalize multi-protocol mode, stack 0/1 enable, flit-format and stack bandwidth with remote link partner
  - *State Machine:* Independent link state management for state machine for individual FDI interface in Multi-protocol mode
- Data flow considerations
  - *Multiplexing:* Muxing of data coming from independent data source
  - *Error Detection:* CRC/parity error detection in D2D Adapter for all flit formats

Ecosystem Adoption

Design and Verification considerations	VIP features
Link initialization	• Parameter exchange advertisement • Clock gate handshake for independent FDI interface • API to move to specific state
Data Flow	• Independent interfaces to transport data using multiple instances • API • Configure flit format • Configure protocol for both stacks • Data integrity check using CRC, parity etc.
Throughput	• Per flit arbitration with 50% bandwidth
Error injection	• Callback • To inject error in flit • To corrupt sideband messages

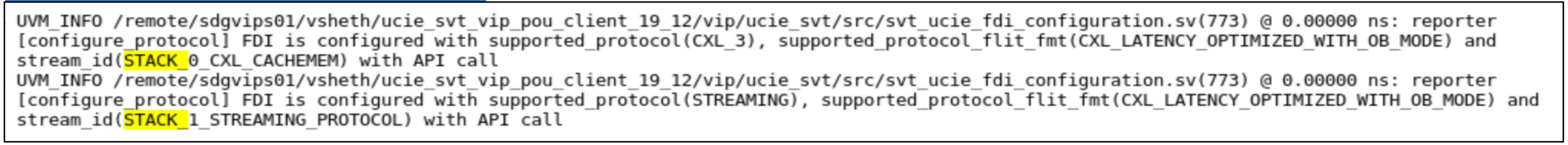
Ecosystem Adoption – Synopsys UCIe VIP

API to configure multi-protocol



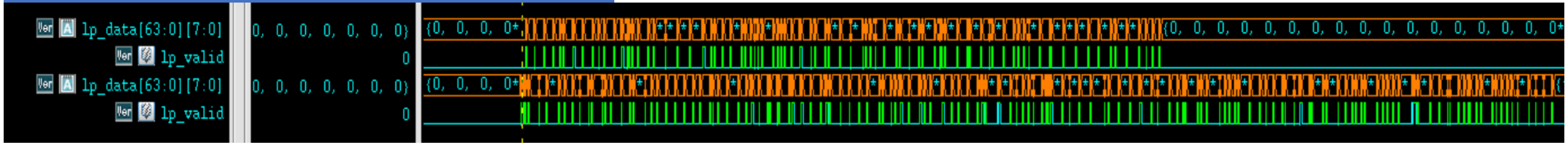
```
foreach(env_cfg.ds_protocol_cfg.fdi_cfg[i,j])begin
  if(i==0 || j==1)begin
    env_cfg.ds_protocol_cfg.fdi_cfg[i][j].configure_protocol(svt_ucie_types::CXL_3, svt_ucie_types::CXL_LATENCY_OPTIMIZED_WITH_OB_MODE, svt_ucie_types::STACK_0_CXL_CACHEMEM);
  end
  else begin
    env_cfg.ds_protocol_cfg.fdi_cfg[i][j].configure_protocol(svt_ucie_types::STREAMING, svt_ucie_types::CXL_LATENCY_OPTIMIZED_WITH_OB_MODE, svt_ucie_types::STACK_1_STREAMING_PROTOCOL);
  end
end
```

Configure Protocol debug message



```
UVM_INFO /remote/sdgvips01/vsheth/ucie_svt_vip_pou_client_19_12/vip/ucie_svt/src/svt_ucie_fdi_configuration.sv(773) @ 0.00000 ns: reporter
[configure_protocol] FDI is configured with supported_protocol(CXL_3), supported_protocol_flit_fmt(CXL_LATENCY_OPTIMIZED_WITH_OB_MODE) and
stream_id(STACK_0_CXL_CACHEMEM) with API call
UVM_INFO /remote/sdgvips01/vsheth/ucie_svt_vip_pou_client_19_12/vip/ucie_svt/src/svt_ucie_fdi_configuration.sv(773) @ 0.00000 ns: reporter
[configure_protocol] FDI is configured with supported_protocol(STREAMING), supported_protocol_flit_fmt(CXL_LATENCY_OPTIMIZED_WITH_OB_MODE) and
stream_id(STACK_1_STREAMING_PROTOCOL) with API call
```

Data Transfer on individual FDI lp_data and lp_valid



UCIe 1.1 Verification Requirements

1 Preventive Monitoring

2 On-field Repairability

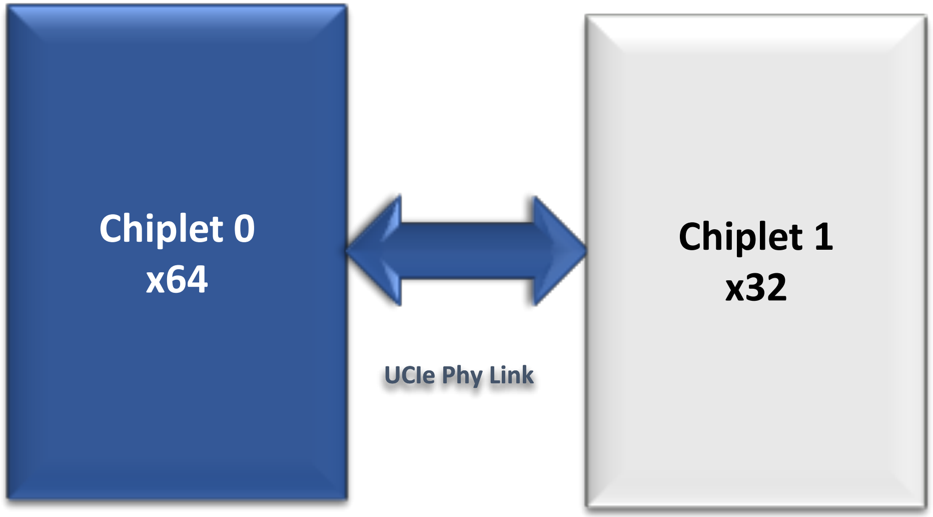
3 Ecosystem Adoption

4 Cost Optimization

5 Compliance Testing

Cost Optimization

- Advance package considerations
  - *Reusability:* x64 link can operate as a x32 when supported and negotiated in MBINIT.PARAM
  - *Optimization:* Lesser fan-out and reduced die cost by reducing pins
- Enable x32 support using below two methods:
  - *Link capability DVSEC register* – APMW
  - *Physical Layer control register* – Force x32 width mode

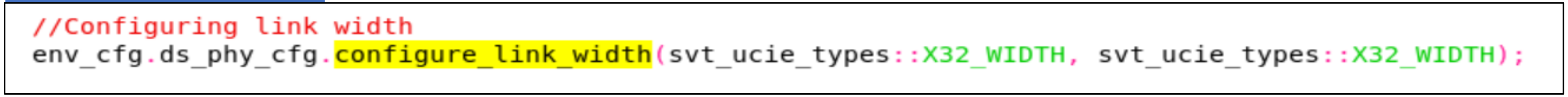


Cost Optimization

Design and Verification considerations	VIP features
Parameter Exchange	• UCIe-A x32 parameter exchange • API • Enable x32 support • Bypass link states • Move to specific link state
MB Repair	• API • Control repair pattern count • Corrupt repair result
MB Reversal	• API • Control reversal pattern count • Corrupt reversal result
Data Flow	• API • Control inter packet delay • Inject back pressure

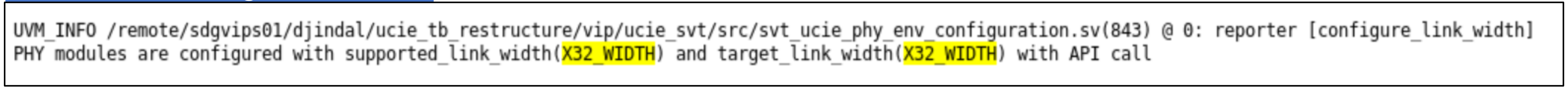
Cost Optimization – Synopsys UCIe VIP

API to configure x32



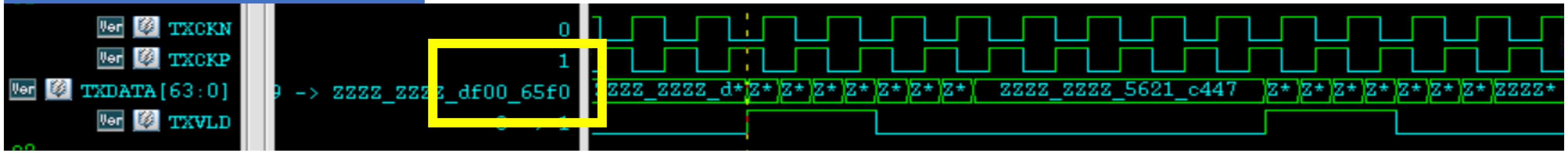
```
//Configuring link width
env_cfg.ds_phy_cfg.configure_link_width(svt_ucie_types::X32_WIDTH, svt_ucie_types::X32_WIDTH);
```

Configure link width debug message

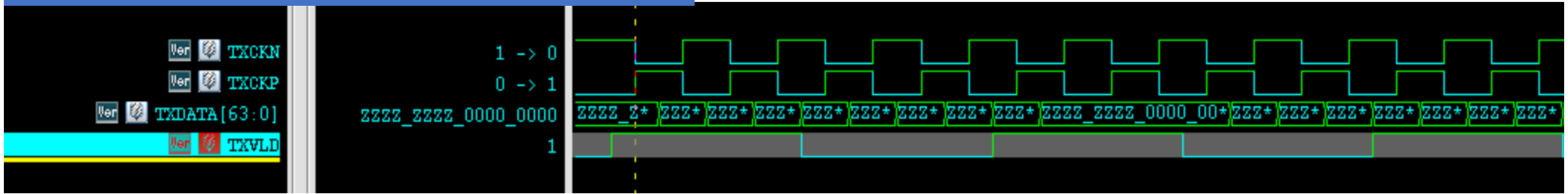


```
UVM_INFO /remote/sdgvips01/djindal/ucie_tb_restructure/vip/ucie_svt/src/svt_ucie_phy_env_configuration.sv(843) @ 0: reporter [configure_link_width]
PHY modules are configured with supported_link_width(X32_WIDTH) and target_link_width(X32_WIDTH) with API call
```

Data Transfer on LSB 32-bits



B2B Data Transfer utilizing optimal link bandwidth



UCIe 1.1 Verification Requirements

1 Preventive Monitoring

2 On-field Repairability

3 Ecosystem Adoption

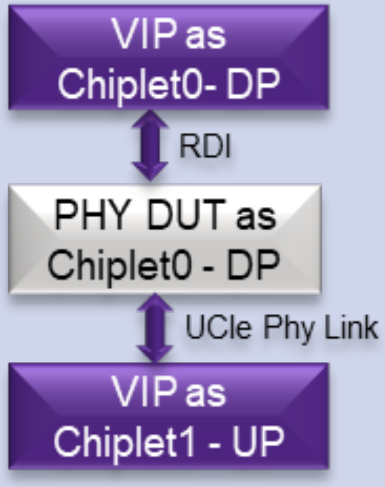
4 Cost Optimization

5 Compliance Testing

Compliance Testing

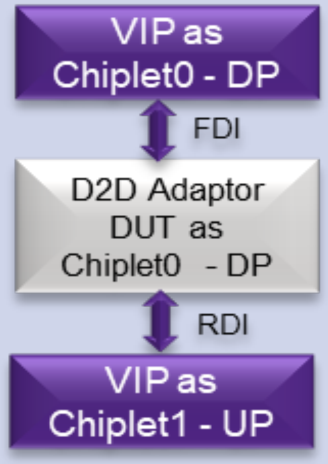
- PHY Compliance
- Adapter Compliance
- Protocol Layer Compliance

PHY Compliance

Test criteria	VIP features	VIP topology
Link Initialization	• API • Trigger link initialization • Control training pattern counts	 VIP as Chiplet0- DP RDI PHY DUT as Chiplet0 - DP UCIe Phy Link VIP as Chiplet1 - UP
Timing/Voltage margining	• API to control eye margins during LTSM states	
BER measurement	• Callback • To inject CRC, parity errors • To timeout sideband request	
Error injection	• API • Control timeout of state • Timeout sideband request • Corrupt training pattern results • Callback to block sideband response message	

Golden Die: All above and ability to inject errors that cause timeouts on sideband messages as well as states

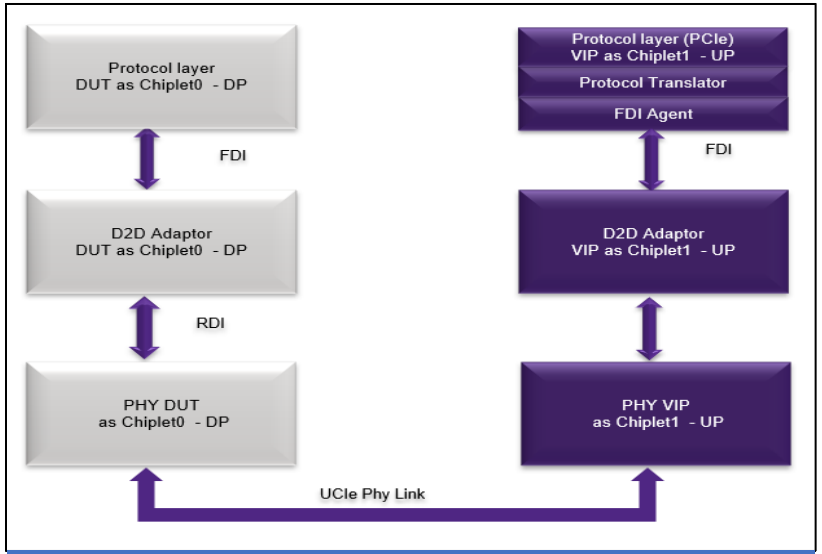
Adapter Compliance

Test criteria	VIP features	VIP topology
Injection of NOP/Test flit	• API to enable flit injection • Callback to inject flits	 VIP as Chiplet0 - DP FDI D2D Adaptor DUT as Chiplet0 - DP RDI VIP as Chiplet1 - UP
State Request	• API • Enable state request injection • Move to a specific state	
Response sideband messages	• API to enable sideband response message injection • Callback to block sideband response	
Retry	• API • Enable retry injection • Enable/disable retry rules groups • ACK/NAK indication of received flit	

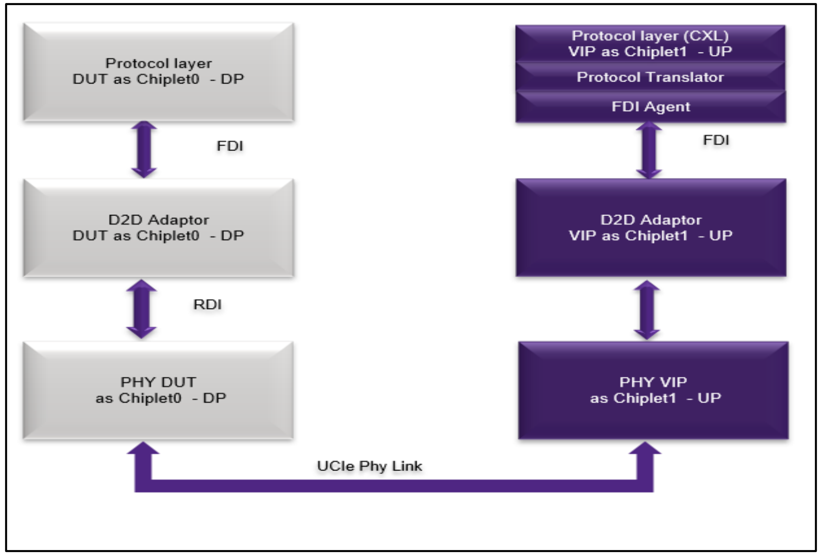
Golden Die: Support all formats, ability to inject above errors

Protocol Compliance

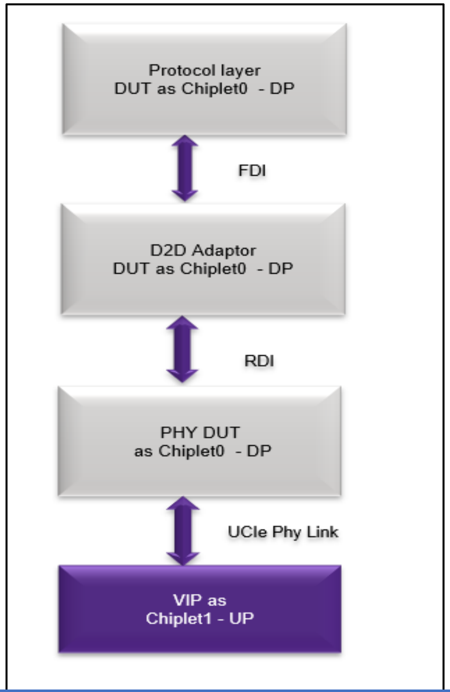
- Leverage PCIe and CXL Protocol compliances defined by those specifications
- Streaming protocols: use their respective compliance
- VIP features:
  - Ability to generate PCIe, CXL and streaming protocol scenarios
  - Callback to inject errors



Full Stack DUT
PCIe Protocol layer



Full Stack DUT
CXL Protocol layer



Full Stack DUT
Streaming Protocol layer

Compliance testing

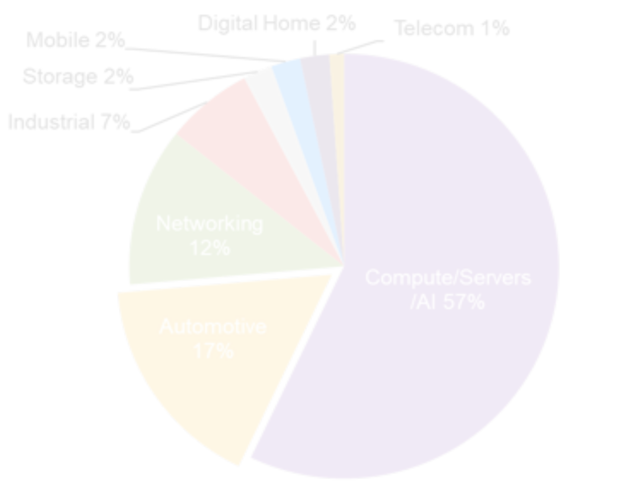
Testing Phases

Phases	Phase1	Phase2	Phase 3
Goal	Bring Up	Data Flow	Error injection
Design Verification Consideration	• Link up • RDI bring up • FDI bring up	• Flit formats • CRC, parity, retry etc. • Data integrity	• Error injection at each layer
VIP feature	• API • Trigger link initialization • Control training pattern counts	• API • Drive sideband and main-band traffic. • Control parity and retry features • Analysis port at each interface for score-boarding • All flit formats	• Callback • To inject errors • To block sideband response message • API • Control timeout of state • Delay sideband response message

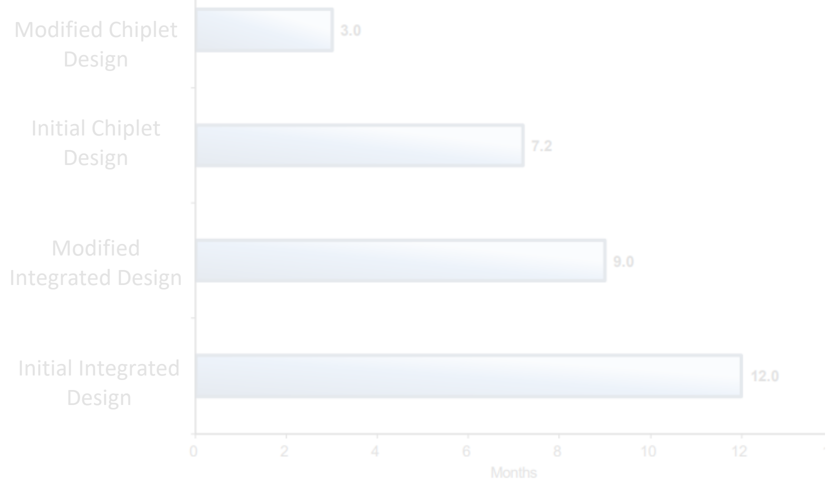
Interface Verification Critical for Chiplet and UCIe Adoption

Die-to-Die IP design starts to grow 5X in 5 years***

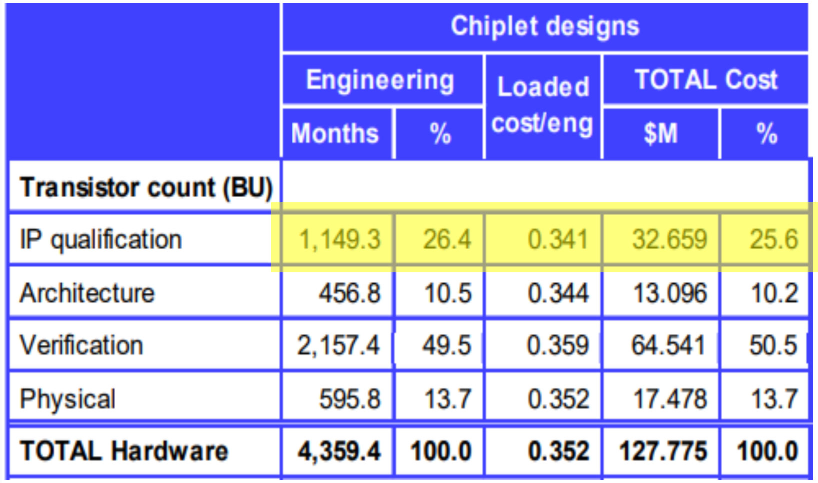
Design Start by Application



Time to Design (Integrated vs Chiplet)



Cost and Effort Spent across Functions



	Chiplet designs				
	Engineering		Loaded cost/eng	TOTAL Cost	
	Months	%		$M	%
Transistor count (BU)					
IP qualification	1,149.3	26.4	0.341	32.659	25.6
Architecture	456.8	10.5	0.344	13.096	10.2
Verification	2,157.4	49.5	0.359	64.541	50.5
Physical	595.8	13.7	0.352	17.478	13.7
TOTAL Hardware	**4,359.4**	**100.0**	**0.352**	**127.775**	**100.0**

“ $50B in Chiplet revenue forecasted by 2024 ”

“Led by HPC & Automotive ”

“ The time-to-market may be 60% faster for initial chiplet designs compared to integrated designs ”

75%

Cost/Time spent in Qualification (25%) and Verification (50%)

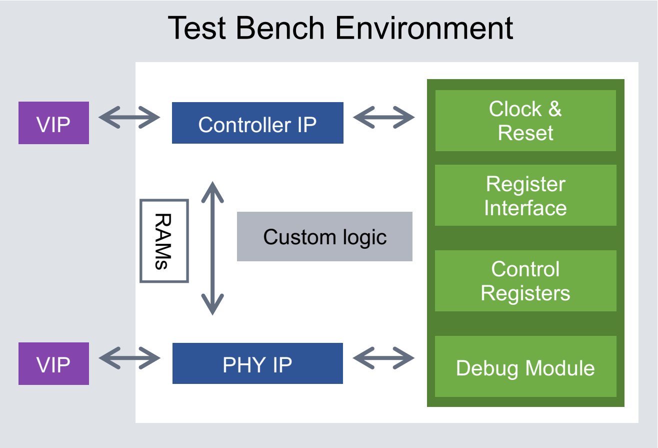
*Source: Gartner 2021 **Source: IBS 2023 ***Source: IP Nest 2022

Reducing Inter-logic SoCs Tape-out Risks

Pre-validated IP-VIP SoC Verification Kits Reduce Design Risk & Accelerate Time-to-Market

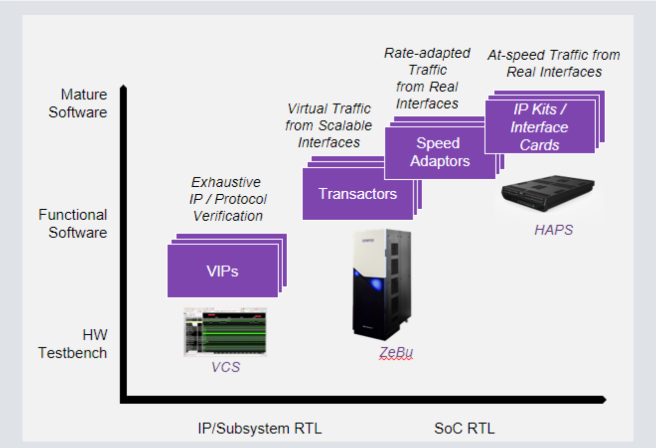
IP/Vendor Selection

- Broadest IP VIP portfolio
- Pre-tested, silicon-proven IP Subsystems for your SoC
- IP & SoC experts configure and customize to your requirements



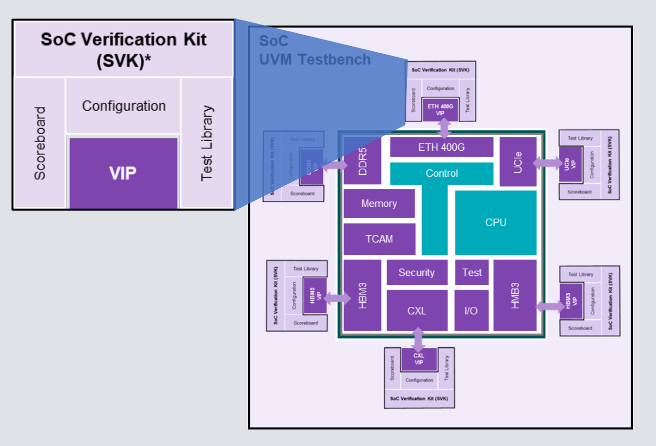
IP Qualification

- Active engagement with spec bodies and eco-system partners
- Ready-to-go Compliance test suites for IP qualification
- Frees your team to work on your product differentiation


Mature Software
Functional Software
HW Testbench
IP/Subsystem RTL
SoC RTL
Exhaustive IP / Protocol Verification
VIPs
VCS
Virtual Traffic from Scalable Interfaces
Transactors
ZeBu
Rate-adapted Traffic from Real Interfaces
Speed Adaptors
At-speed Traffic from Real Interfaces
IP Kits / Interface Cards
HAPS

IP Integration

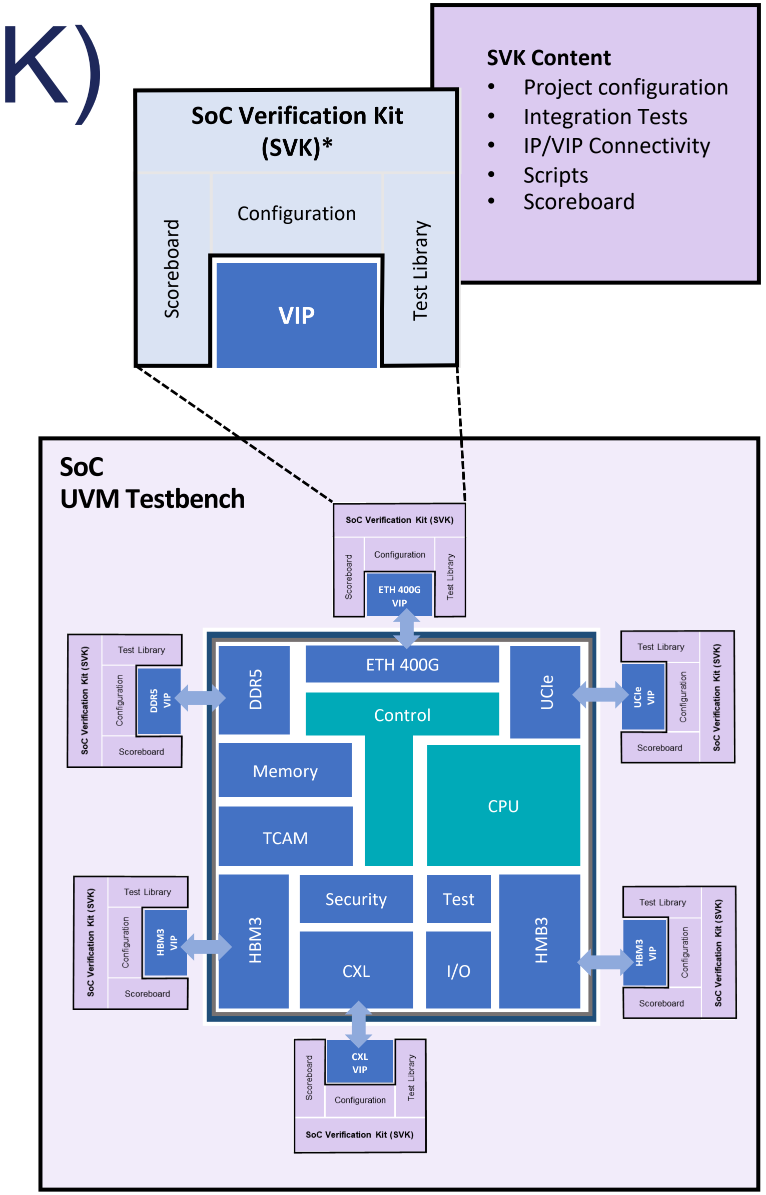
- Reference flows and services for IP integration and convergence between project teams
- First-time-right SoC integration speeds TTM



Synopsys SoC Verification Kit (SVK)

Accelerating SoC Verification with Synopsys IP, VIP and VCS

- Challenges
  - Expertise for UVM-based, scalable testbenches
  - Verification resource limitations
- SVK Benefits
  - Out-of-the-box verification solution for complex protocols
  - Tailor made for project-specific IP configuration
  - Accelerates the SoC testbench development
  - Enables testing of Synopsys IPs in Subsystem/SoC environment
  - Lowers integration risk through proven verification methodologies
- Available for HPC/Automotive protocols: UCIe, PCIe, CXL, DDR, HBM, Ethernet

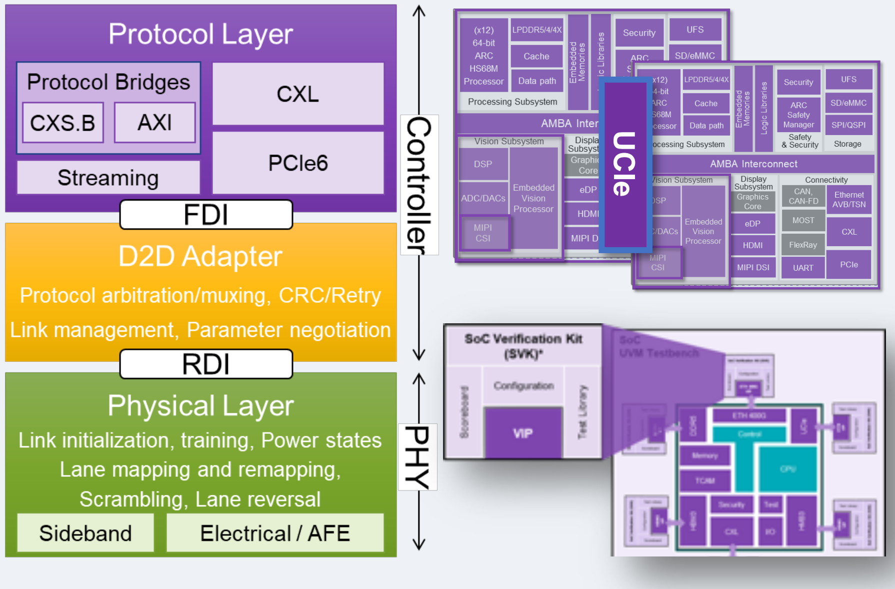


SVK Content

- Project configuration
- Integration Tests
- IP/VIP Connectivity
- Scripts
- Scoreboard

Verify and Validate Your Multi-Die Interfaces

Synopsys UCIe IP–VIP Validated Together to Reduce Project Risk



Synopsys Protocol Verification Solutions

- Verification IP
- Test Suite
- Transactors
- System Verification Solutions
- Virtual/Hybrid
- SoC /Subsystem Verification Kits (IP-VIP-XTOR)
- Software Development and System Validation

Latest UCIe Specifications and Topologies

Synopsys UCIe VIP supports entire UCIe standard from PHY to Protocol layer covering all IP to System Level topologies

Co-validated with Synopsys IPs

Synopsys UCIe VIP used to validate UCIe IP from Synopsys and leading partners ensuring

Debug Productivity

Native VCS and Verdi integration provides high simulation and debug productivity for UCIe

Protocol Verification Solutions for all Use-cases

IP to SoC Level RTL Verification

Software development and system validation

Integrated IP-VIP setups for quick bring-up and verification synopsys®