Experiences and Insights Verifying PCI Express® Cores

Luis Rodriguez

PCIe VIP Manager

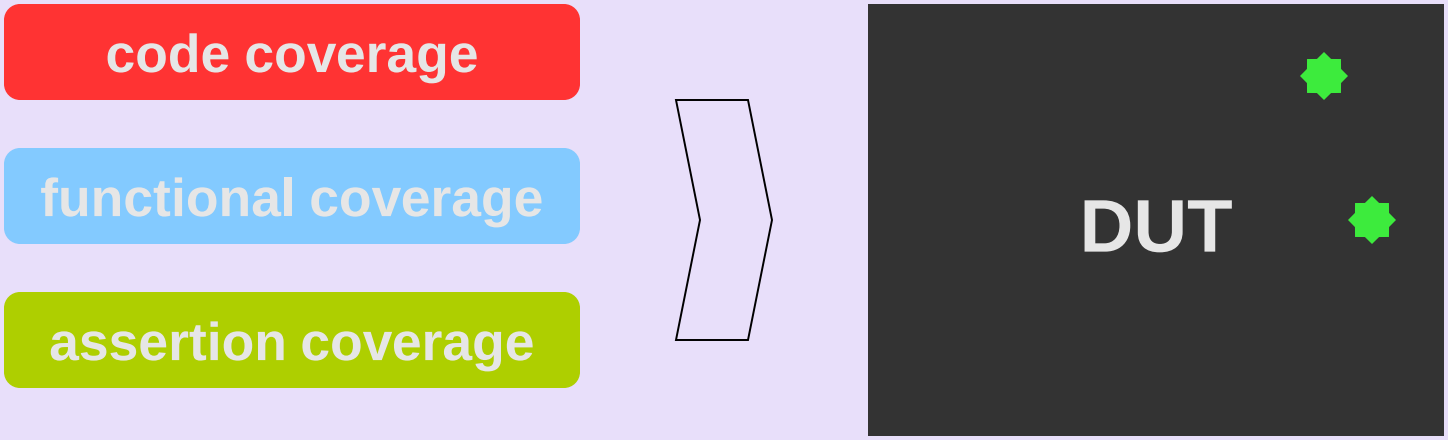
Avery Design Systems

Agenda

- Verification Challenges
- Common Design Bugs (PCIe 1.1/2.0/3.0)
- Verification Framework
- Constrained Random Testing
- Reference Model Scoreboarding
- Collecting and Using Coverage Data

Ultimate Goal of Verification

- to achieve complete coverage for the design under test (DUT) in order to uncover all of the bugs



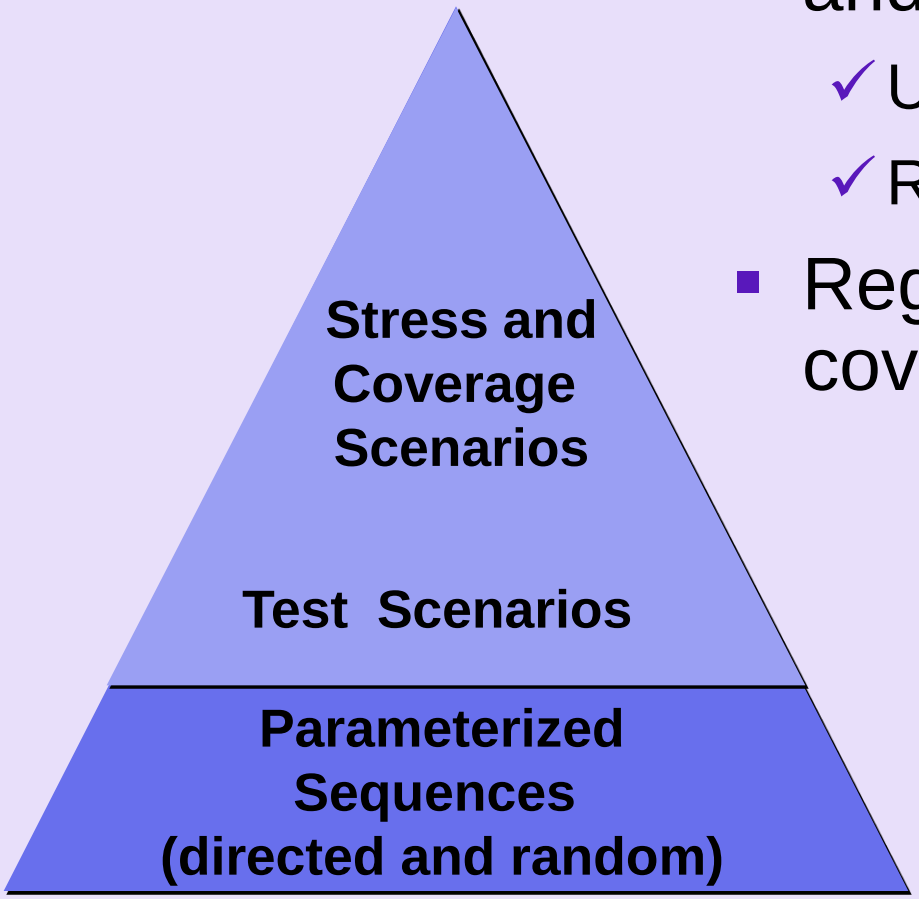
PCIe Core-level Verification Challenges

- Multiple targets
  - ✓ Many component types (EP, RC, SW, PIPE PHY)
  - ✓ Specification revisions (1.0a, 1.1, 2.0, 3.0, SR-IOV)
  - ✓ ECNs (ARI, ATS)
- Robustness & Portability of BFMs
  - ✓ Implementation variations (ex. LW Upconfigure)
- Out of spec design issues
  - ✓ Not all designs require 100% compliance
  - ✓ “Quick” simulation design modes

PCIe Core-level Verification Challenges

- End 2 End verification
  - ✓ Port driver Application Interfaces are non-standard and impose limitations effecting how core can be used
  - ✓ End 2 end checkers provide background verification (ex. Transaction ordering rules across switch)
- Testsuite design
  - ✓ Context-based verification
  - ✓ Reuse and portability
- Debug and observability

Testing Framework



- Supports functional verification and coverage
  - ✓ Unit-level directed testing
  - ✓ Random stress testing
- Regression test and checklist coverage reports

Compliance Coverage Report

Assertion checker implemented in BFM

Assertion checker triggered during regression

Applies	Name	Description	BFM	Test	Dynamic Trigger	Scenario_Coverage
RC, EP, SW, BR	CFG.09.00#01	Extended capabilities in a device configuration space must begin at offset 100h with a PCI Express Enhanced Capability Header.	B		Y	config_1_7.vh config_1_8.vh
RC, EP, SW, BR	CFG.09.01#01	Absence of any extended capabilities must be indicated by an Enhanced Capability Header with a Capability ID of 0000h, Capability Version of 0h, and Next Capability Offset of 0h.	B		Y	avery_fcpe_sw.vh sys_2_2_sw.vh
RC, EP, SW, BR	CFG.09.03#01	The PCI Express Extended Capability ID field of a PCI Express Enhanced Capability Header must be a valid PCI-SIG defined ID number (or 0xFFFF h or 0x0000g for a terminating header).	B		Y	config_1_7.vh.run config_1_8.vh.run
RC, EP, SW, BR	CFG.09.03#02	The Capability Version field of a PCI Express Enhanced Capability Header must correctly indicate the version of the capability structure present.	B		Y	avery_fcpe_sw.vh.run sys_2_2_sw.vh.run

Provides key checklist coverage data to drive compliance checklist closure

Top ranked test cases to exercise checker

Typical Design Bugs Encountered

- 1.1
  - ✓ Linkwidth down configuration
    - Premature idling of lanes prior to Configuration.Complete (PHY.02.19#27)
  - ✓ L1 and L2/L3 link state transitions protocol
    - Upstream component fails to detect EI (PMG.03.09#11)
  - ✓ Byte enable and byte count correctness
    - Unsuccessful completions (TXN.03.02#32, #34,#35)
  - ✓ Transaction ordering across switch
    - Read cannot pass write (TXN.04.00#06)
  - ✓ VC Negotiation
    - VC Negotiation Pending not cleared (CFG.11.08#04)

Typical Design Bugs Encountered

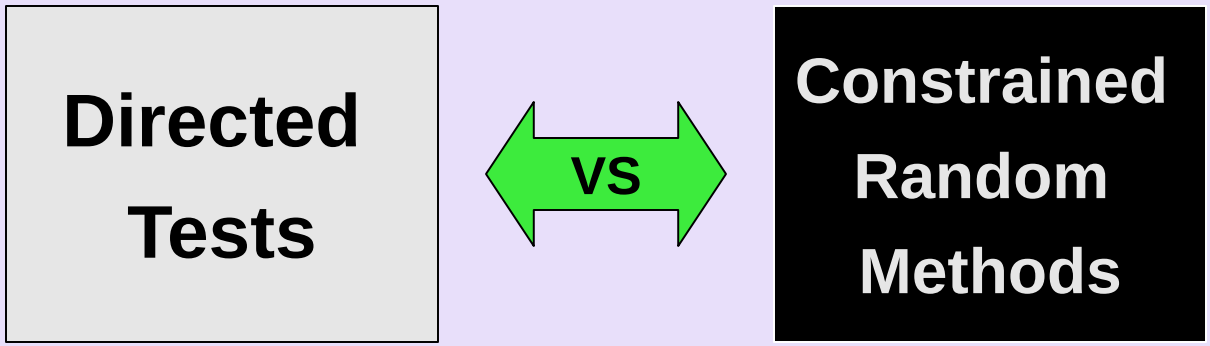
- 2.0
  - ✓ Linkwidth up/down configure
    - Initiator enters Configuration.Linkwidth.Start prior to sending EIEOS on lanes it plans to activate (PHY.02.06#16)
    - Downstream lanes not initiating upconfiguration activates more lanes than initiator (PHY.02.19#58)
  - ✓ Loopback exit
    - Loopback slave does not detect EI (PHY.02.26#06, PHY.02.26#13)
  - ✓ Gen2 extended sync in Recovery.RcvrLock
    - Fails to transmit 1024 TS1 (PHY.02.20#19)
  - ✓ Gen2 configuration space default values

Typical Design Bugs Encountered

- 3.0
  - ✓ Linkwidth up/down configure
    - Initiator enters Configuration.Linkwidth.Start prior to sending EIEOS on lanes it plans to activate (PHY.02.06#16)
    - Downstream lanes not initiating upconfiguration activates more lanes than initiator (PHY.02.19#58)
  - ✓ Receiver Equalization
  - ✓ Cross Mode Speed Up/Down

Verification Methods

- Best solution for a PCI Express DUT:
  - ✓ combines Directed Testing (based on Checklist items) and C.R.T in a same framework
  - ✓ Tracks Functional Coverage



Directed Testing

- First line of Defense
- Based on Checklist items
  - ✓ PCIe 1.0a/1.1: Checklists provided by PCISIG
  - ✓ PCIe 2.0/3.0: Avery maintains it's own Checklists
- Pros:
  - ✓ Uncover basic Compliance issues
  - ✓ Easy to debug
- Cons:
  - ✓ Time consuming to write and maintain

Constrained Random Tests

- Constrained Random Testing Environments
  - ✓ Pure SystemVerilog
  - ✓ OVM/AVM
  - ✓ VMM
- Pros:
  - ✓ Find hard to uncover bugs, and reduced test writing time (high reuse)
- Cons:
  - ✓ require deeper knowledge of PCIe Protocol and RTL Core to debug
  - ✓ long simulation time and hard to reach deep design states

Exhaustive testing

Some areas are best covered through random tests:

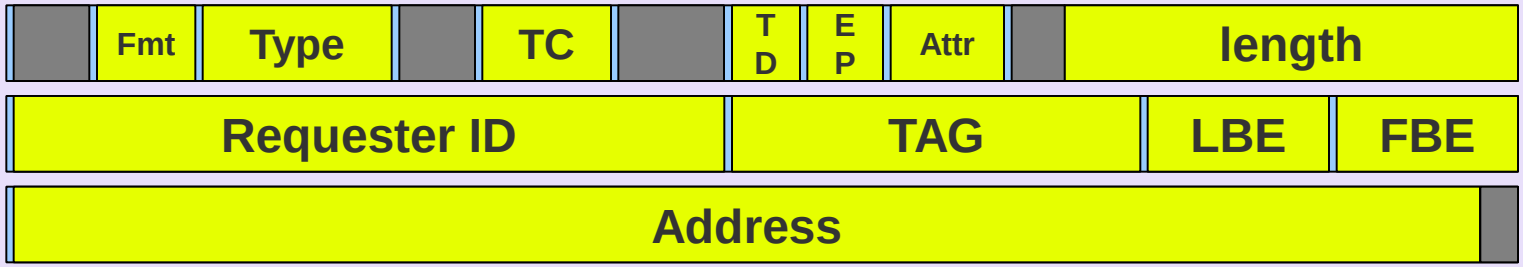
- High stress traffic testing combining Config/Memory/IO Transactions
- Memory/TLP fields
- Byte enables, Payload sizes
- LTSSM State transitions
- Error Reporting

Need for Constrained Random

- How to create Valid Random TLP traffic?
- TLP packets and any other data stream must be constrained by the DUT's current context (reqID, MayPayloadSize, etc…)

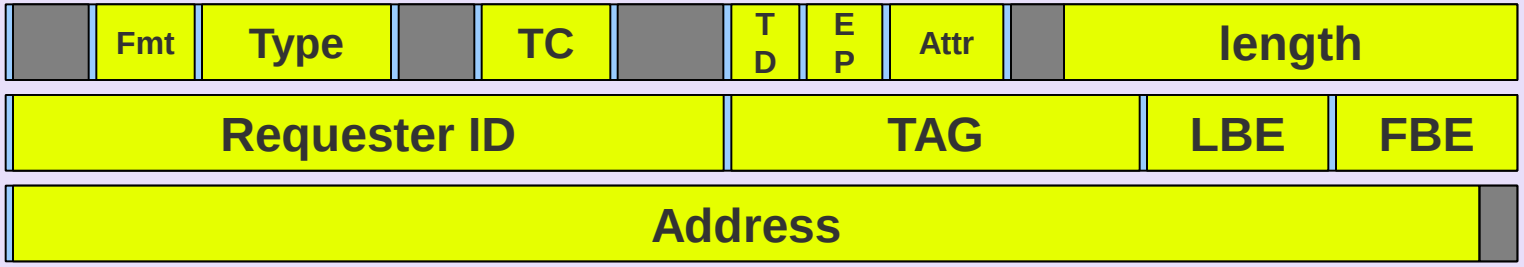
Reserved / non-random

Randomizable

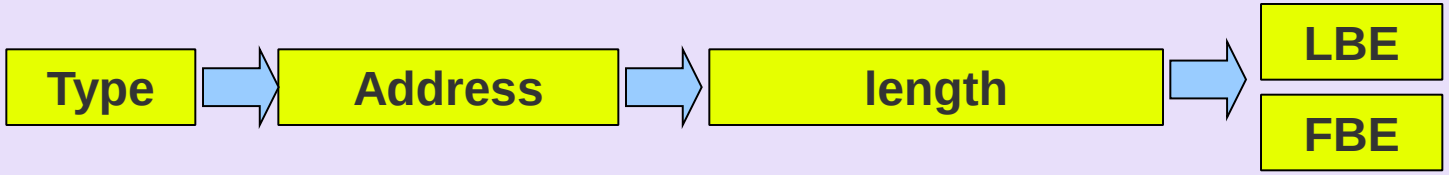


Need for Constrained Random

- Fields in a TLP depend on other fields in the TLP

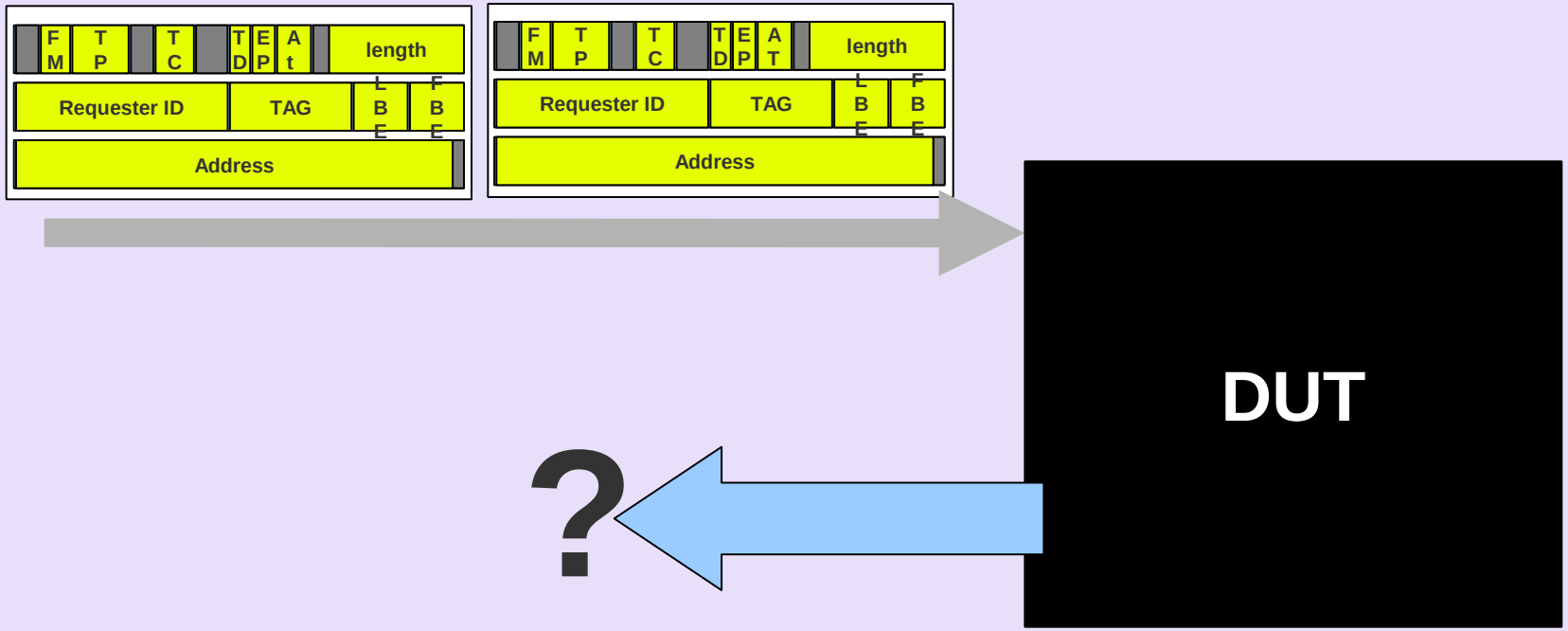


- Constraints tackle these dependencies

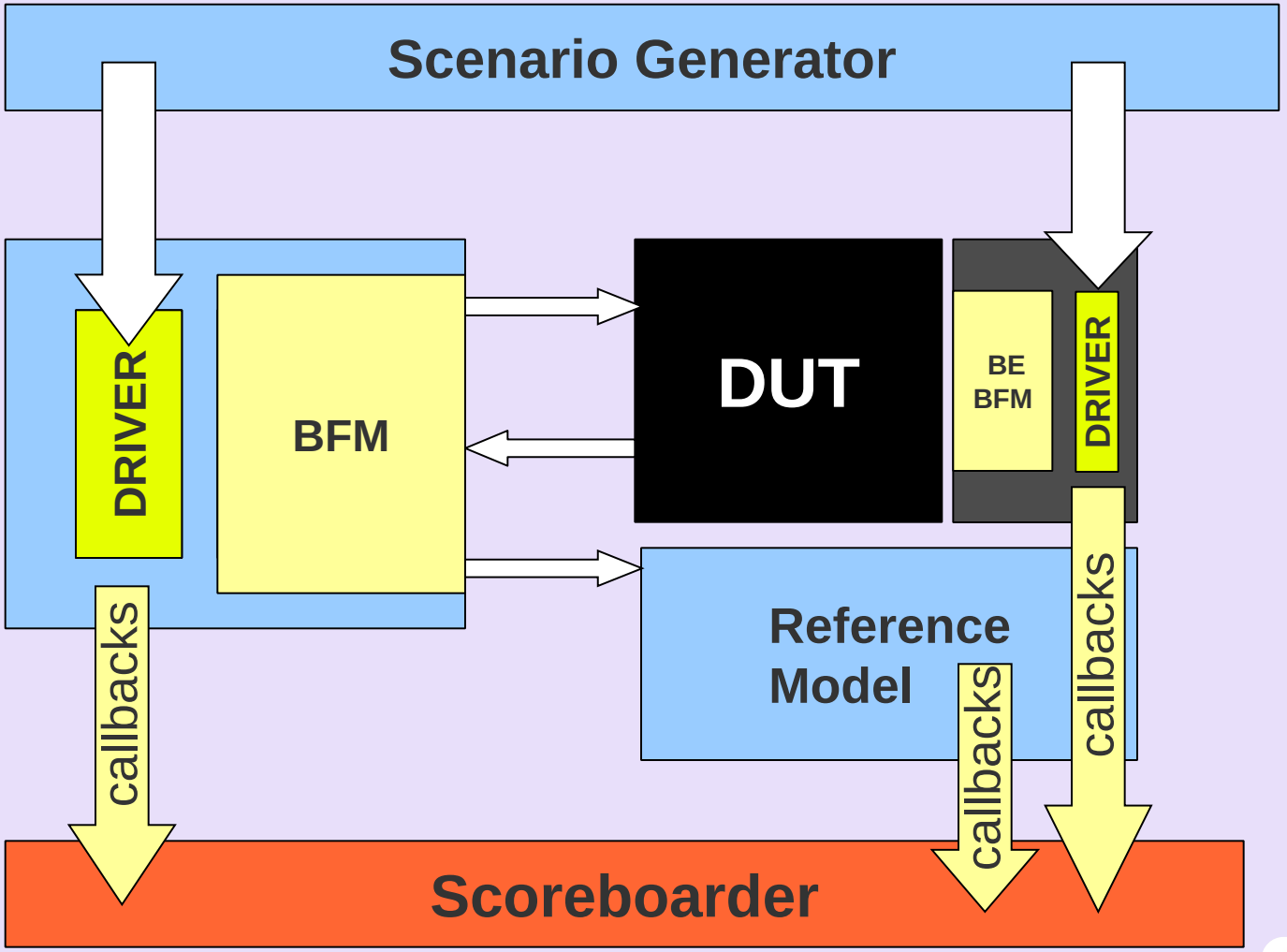


Self Checking

- How to check the response of the DUT to random traffic is correct ?



Reference Model based Scoreboarding



Challenges for Verification at 8GT/s

- Exhaustive TLP/DLLP @ 8GT/s
- Exhaustive LTSSM testing @ 8GT/s
- Stress Memory tests @ 8GT/s

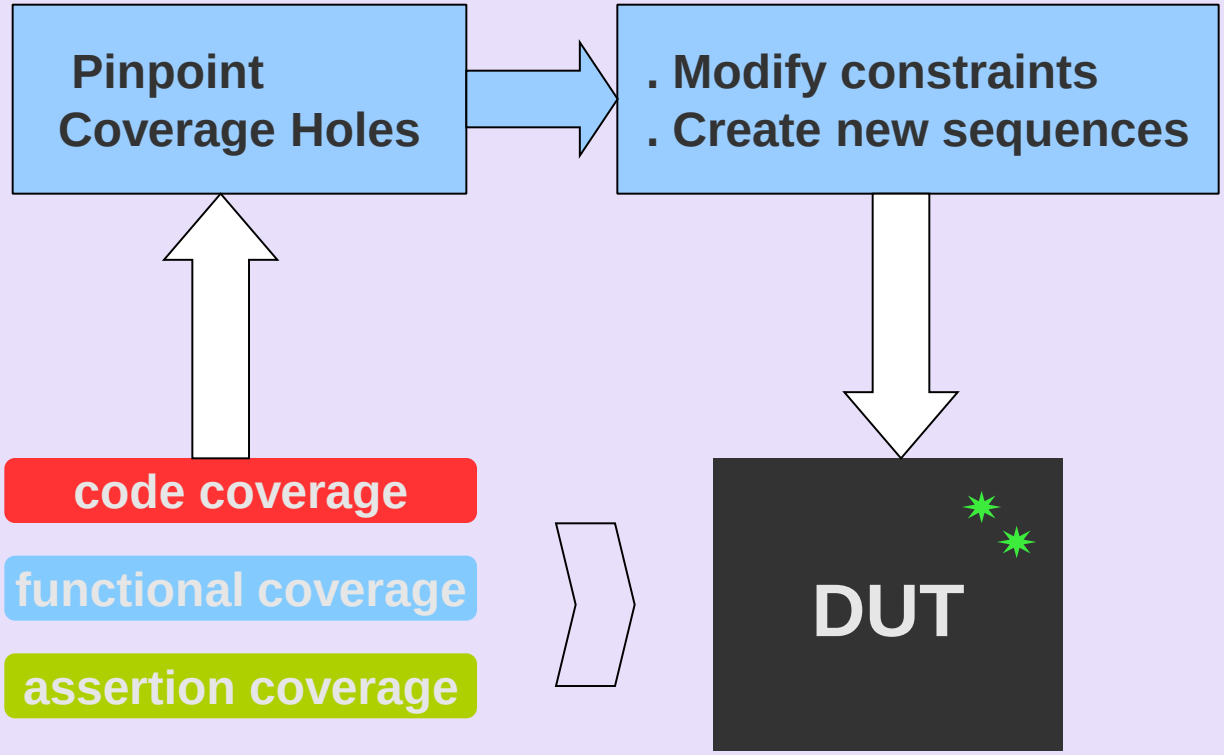
First

After

- Equalization Testing
- Cross Mode Testing PCIe 2.0 <> PCIe 3.0

Collecting and Using Coverage Data

- Use Coverage data as feedback, and close the loop



Lessons Learned

- Company/Perspective
  - ✓ Customer and vendor alignment is very important
  - ✓ IP partners are essential to ensure robust solution based on multi-sourced testing and interoperability
  - ✓ Must support for early drafts of specifications to support early adopters
- Role of commercial VIP
  - ✓ Improves productivity and provides objective confirmation of design intent
  - ✓ Customer needs to become expert user of BFMs to get the most out of the solution. There are no push button solutions.

Lessons Learned

- Core-level verification challenges
  - ✓ BFMs and tests must be flexible to support implementation specific design decisions
    - Spec allows user-defined algorithm
    - Spec has “room for interpretation”
- Compliance checklist and testing foundations
  - ✓ Documentation of testsuites/sequences is essential to understanding 3rd party tests (source code if possible)
  - ✓ Indexing tests to checklist items and coverage presents a more complete compliance picture
  - ✓ Debug and observability improved via verbose messaging modes and source code debug

Lessons Learned

- Beyond compliance checklist-based testing
  - ✓ Alternative methods offer limited improvements
    - Formal verification not suited for PCIe
    - Reference Model based verification provides sequential consistency checking however requires detailed knowledge of core
- Chip-level verification challenges
  - ✓ BFMs need to support multiple domains for embedded system designs
  - ✓ Some embedded designs are not always100% compliant and require BFM to support optional out-of-spec features

Lessons Learned

- Need for Directed Testing
  - ✓ First line of defence against Compliance bugs
  - ✓ Some states are hard to reach through just random traffic (lock combination argument)
- Constrained Random Testing and Scoreboarding
  - ✓ Easier way to tackle exhaustively testing a DUT
  - ✓ Combine with coverage metrics to drive testing plans
  - ✓ Reduces test writing time

Thank you for attending the
PCI-SIG Developers Conference 2010.

For more information please go to
www.pcisig.com