Case Studies of Difficult Scenarios in Functional Verification

Dimitry Pavlovsky

Cadence Design Systems

Agenda

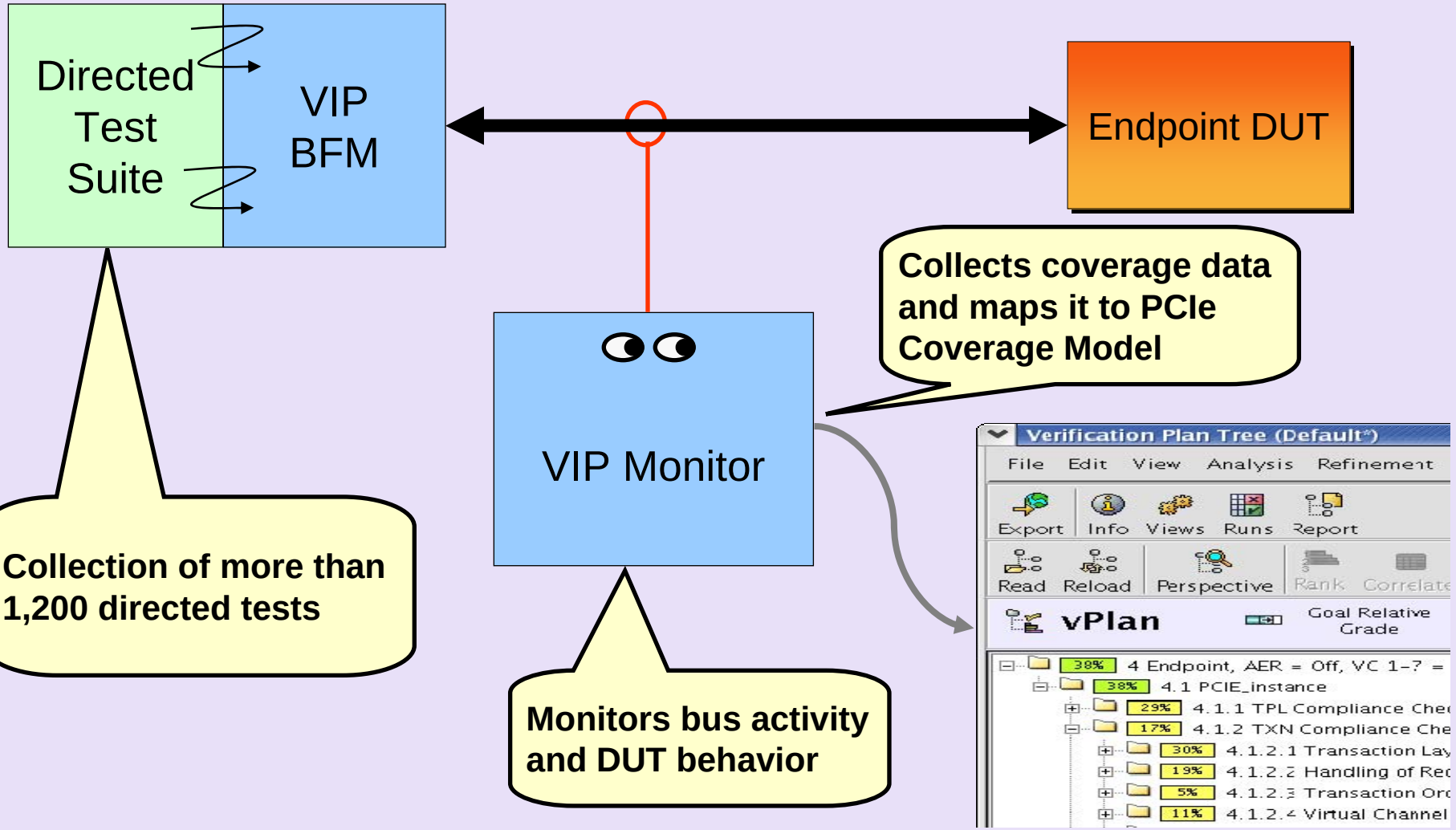
- Two case studies are examined
- For each:
  - ✓ Initial verification approach and its shortcomings are examined
  - ✓ Verification challenges are scrutinized
  - ✓ New, improved verification approach used to reveal hidden flaws are described
  - ✓ Revealed functional flaws are identified and the reasons why they were missed using the initial verification approach are examined

Case Study #1: General Background

Company Type	Large global semiconductor company
Device Type	PCIe® 2.0 compliant Endpoint
Initial Verification Approach	Home-grown test suite of more than 1,200 directed tests

Case Study #1

Initial Verification Environment



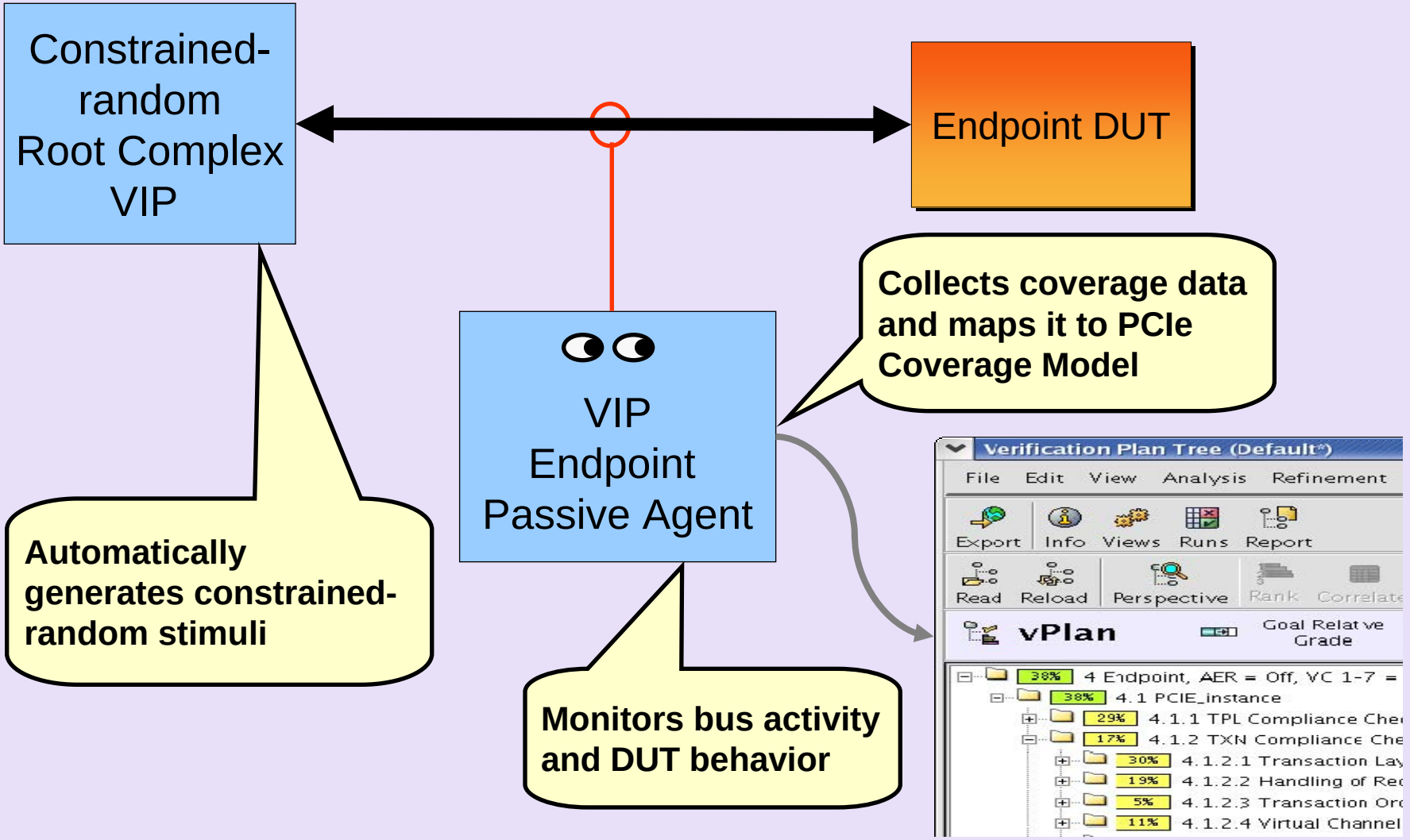
Case Study #1

Shortcomings Of Initial Verification Approach

- Functional coverage was inadequate
- Among other low-coverage sections, low LTSSM functional coverage was of concern
- To increase functional coverage percentage from 64% to over 95%, the company chose to replace home-grown directed test suite with commercially purchased constrained-random Verification IP (VIP)

Case Study #1

Improved Verification Environment



Case Study #1

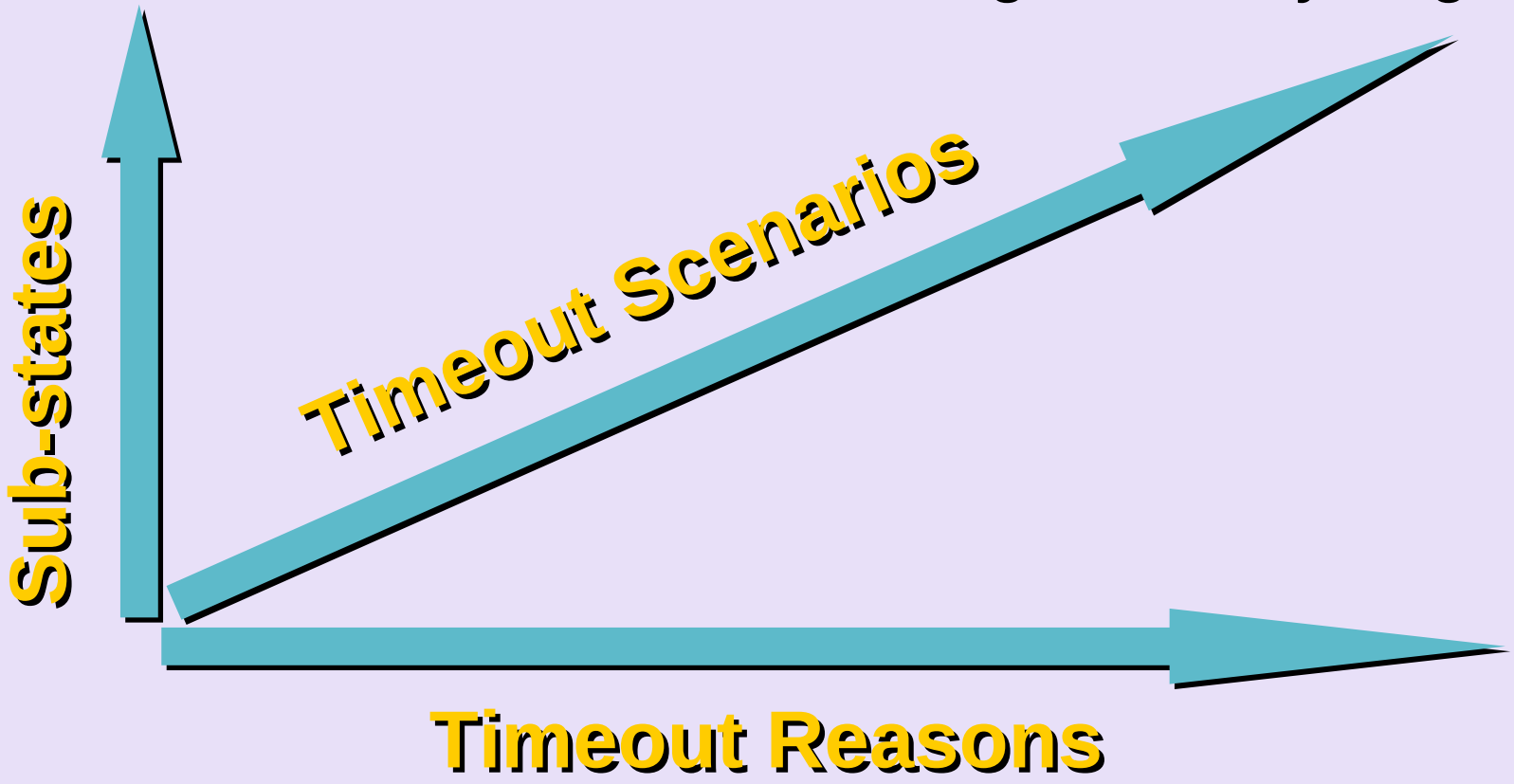
Technical Background

- Almost all LTSSM sub-states have a timeout associated with them
- Each timeout can be caused by various reasons
  - Delay with receiving OSs
  - Receiving wrong TS OSs
  - Receiving wrong number of consecutive TS OSs
  - Receiving TS OSs with link/lane mismatching number
  - Receiving TS OSs with PAD when it is not expected or without PAD when it is expected
  - Receiving EIOS
  - Device on the far side of the link going to Electrical Idle without sending EIOS

Case Study #1

Verification Challenge

- With multiple LTSSM sub-states having timeout times various timeout reasons, the sub-state/timeout reasons matrix grows very large



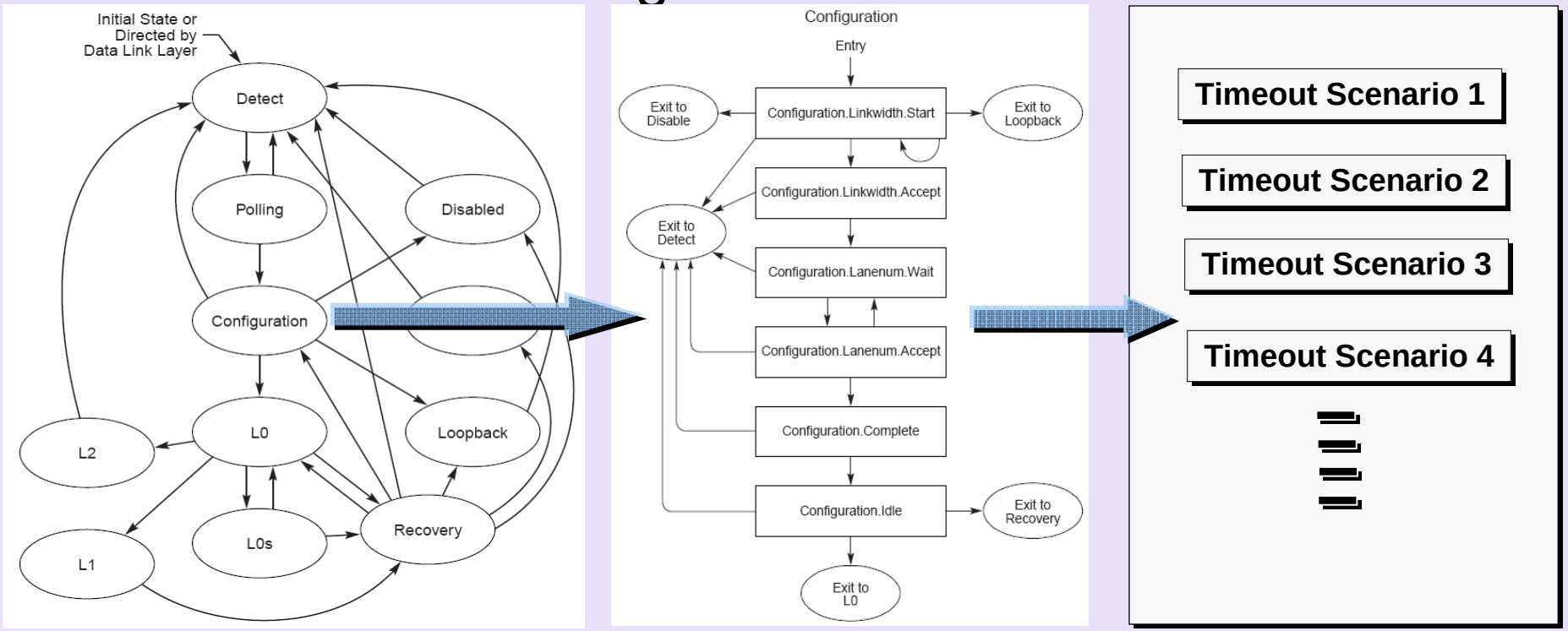
Case Study #1

Verification Challenge (cont.)

- During simulation, timeout in each LTSSM sub-state can be caused by one reason at a time
- To verify all timeout scenarios using directed testing approach requires a large number of tests each explicitly invoking timeout in each sub-state
  - ✓ Requires lots of manual work
  - ✓ Consumes large amounts of time

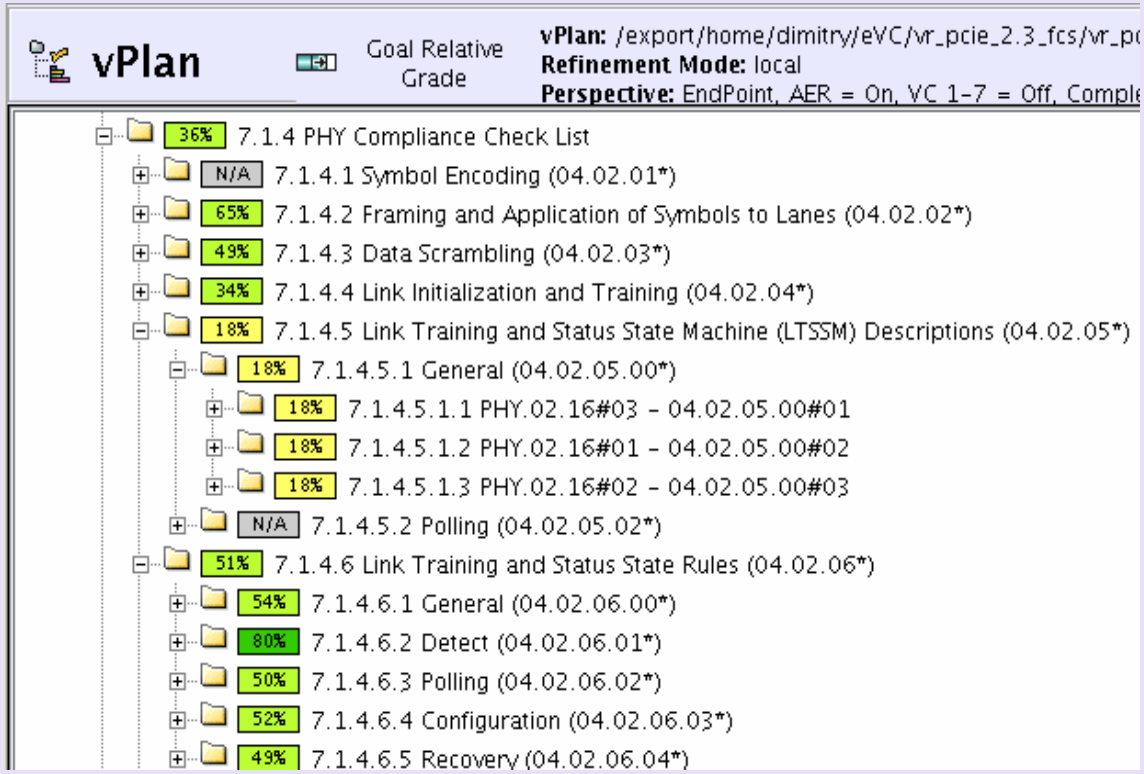
Case Study #1 Implemented Solution

- Various constrained-random timeout scenarios are created and employed in VIP
- Timeout scenarios randomly invoked as VIP transitions through various LTSSM states



Implemented Solution (cont.)

- By running VIP long enough all possible timeout scenarios in all LTSSM sub-states are created
- Pre-supplied functional coverage model is used to assess what test scenarios have or have not been exercised



Case Study #1

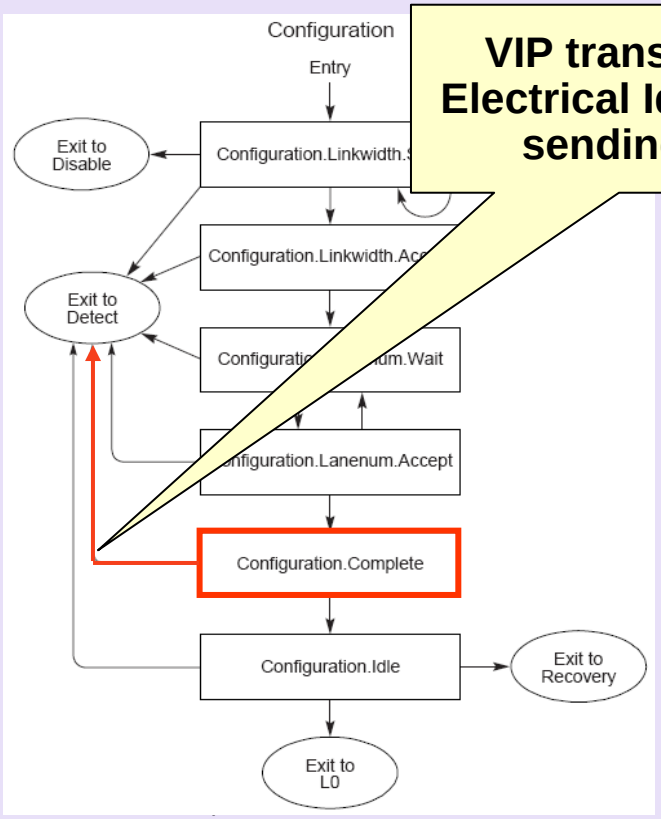
Constrained-Random Verification Results

- Generally the DUT handled the various types of timeout scenarios being in compliance with specification
- However, the advanced constrained-random verification of DUT's LTSSM revealed a severe bug
  - ✓ When both VIP and DUT were in the Configuration.Complete state, the VIP exercised one of the timeout test scenarios by transitioning to Electrical Idle without sending EIOS
  - ✓ That event stalled the DUT's LTSSM in the Configuration.Complete state causing **permanent deadlock**

Case Study #1 Bug Discovered

Constrained-random Root Complex VIP

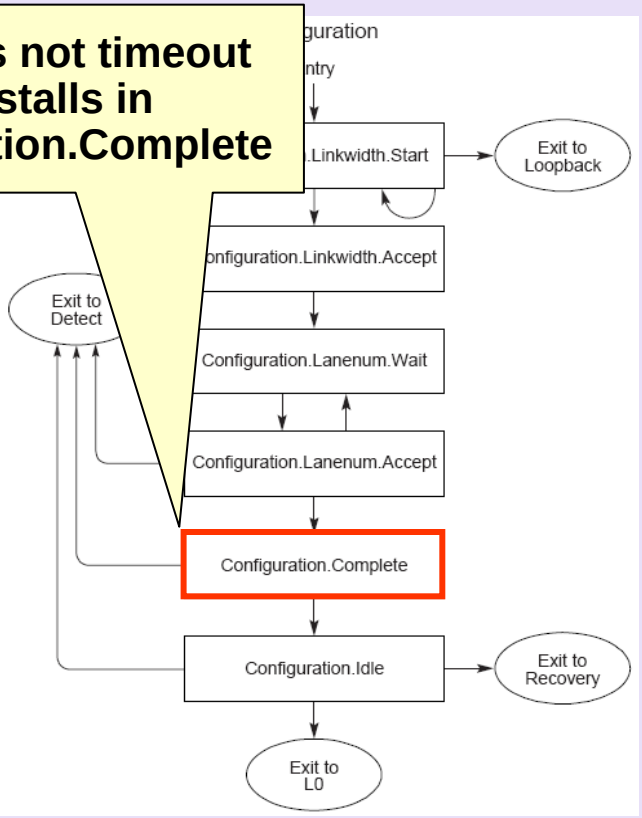
Endpoint DUT



VIP transitions to Electrical Idle without sending EIOS

DUT does not timeout and stalls in Configuration.Complete

VIP Endpoint Passive Agent



Case Study #1 Bug Analysis

- Possible Configuration.Complete state timeout scenarios
  1. Receiving TS1 OSs instead of TS2 OSs
  2. Receiving TS2 OSs with link/lane mismatching numbers
  3. Receiving TS2 OSs with link/lane matching numbers but less than 8 OSs in a raw
  4. Receiving EIOS
  5. Device on the other side of the link (VIP in this case) switching to Electrical Idle without sending EIOS
- Functional coverage model revealed that only scenarios #1 and #2 were covered by directed testing
- Device behaved according to the spec in situations 1-4
- DUT's LTSSM RTL code did NOT account for scenario #5 which caused LTSSM to stall in Configuration.Complete state

Case Study #1

Case #1 Summary

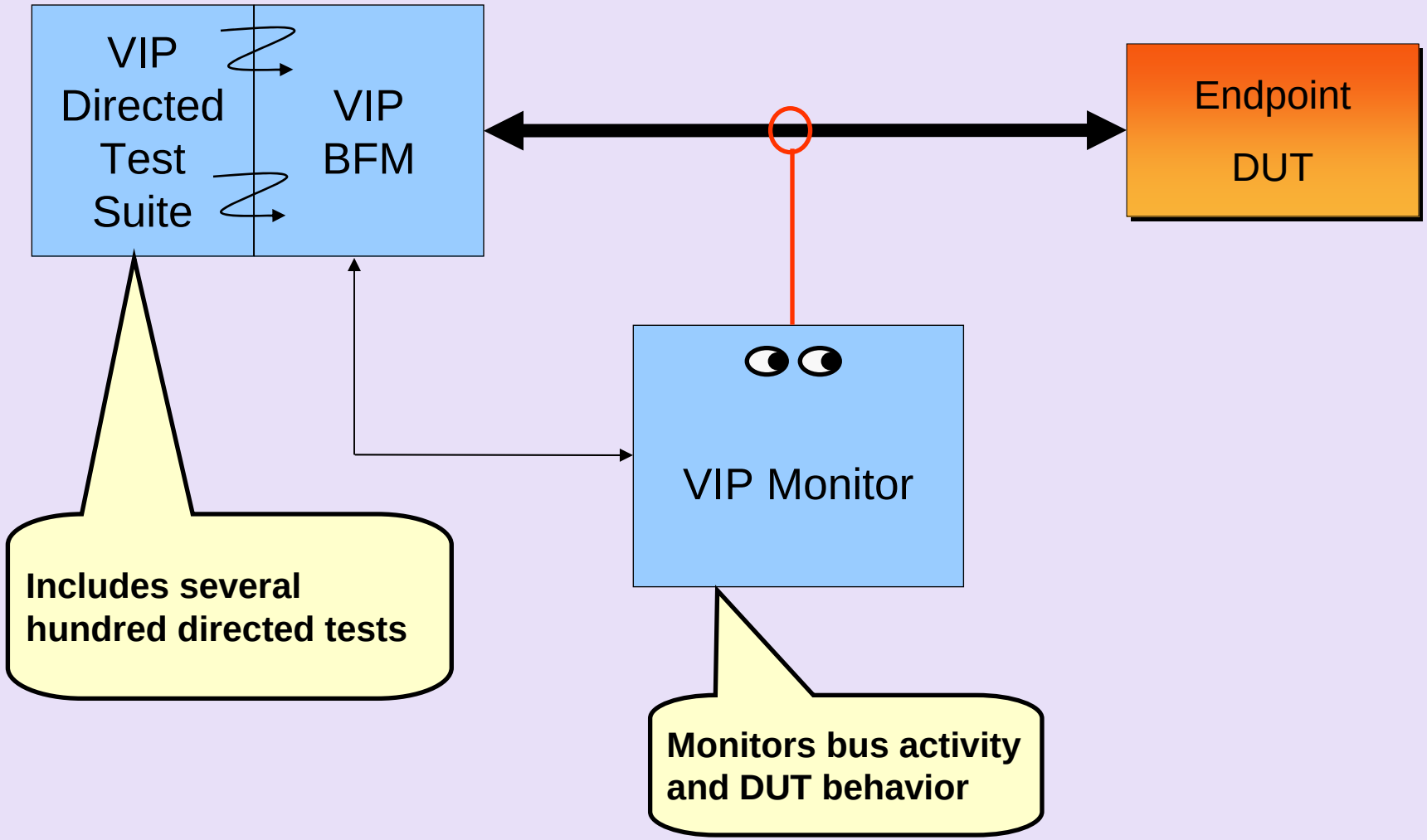
- Advanced constrained-random verification technique enabled company to:
  - ✓ Exercise a wide array of LTSSM timeout test scenarios
  - ✓ Reveal a deeply-hidden functional fault which would lead to a permanent deadlock in the real-life systems
  - ✓ Increase verification quality and predictability

Case Study #2: General Background

Company Type	Fabless Semiconductor startup
Device Type	PCIe 2.0 compliant Endpoint
Initial Verification Approach	Commercially purchased VIP utilizing directed testing methodology

Case Study #2

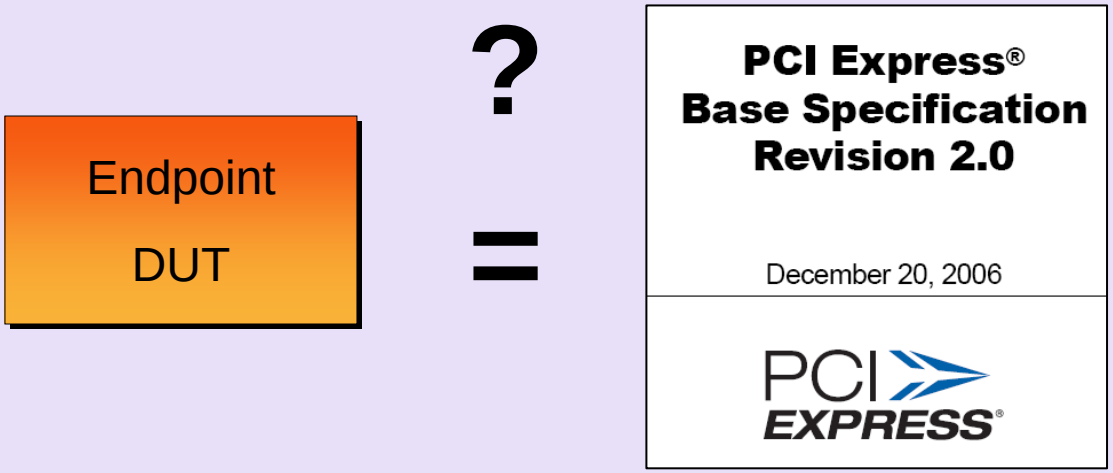
Initial Verification Environment



Case Study #2

Shortcomings Of Initial Verification Approach

- Functional coverage was not employed
- Without functional coverage used, even with all the directed tests passing there was no confidence that all aspects of the PCIe protocol were adequately verified



Case Study #2

Improved Verification Environment

Constrained-random Root Complex VIP

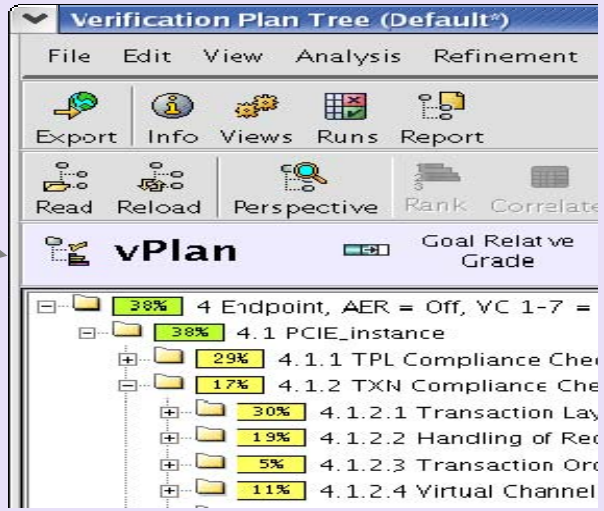
Endpoint DUT

VIP Endpoint Passive Agent

Automatically generates constrained-random stimuli

Collects coverage data and maps it to PCIe Coverage Model

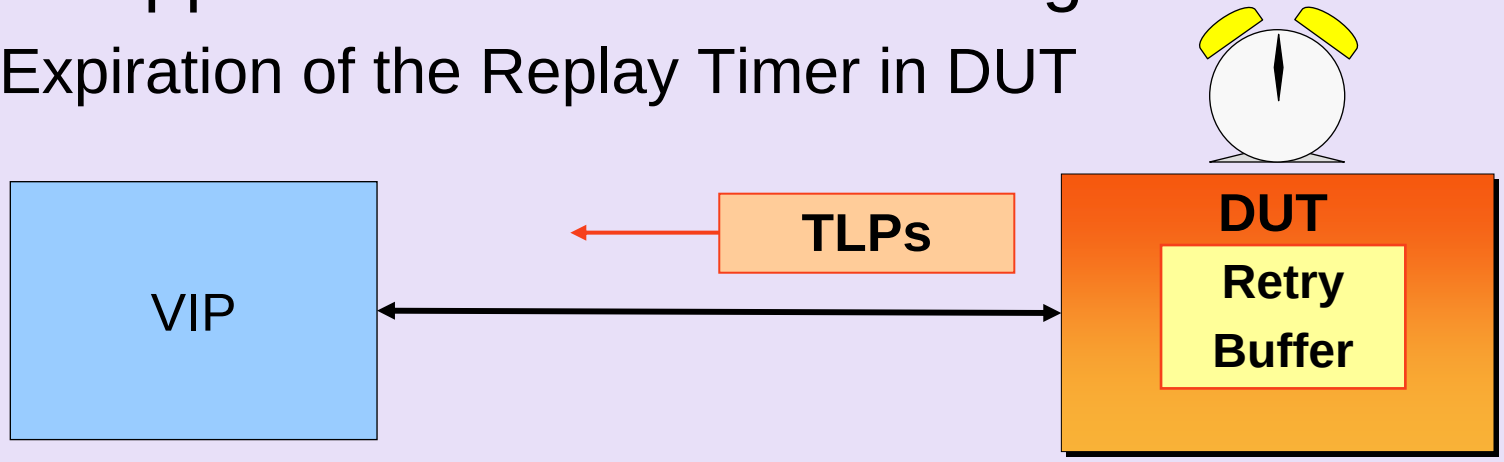
Monitors bus activity and DUT behavior



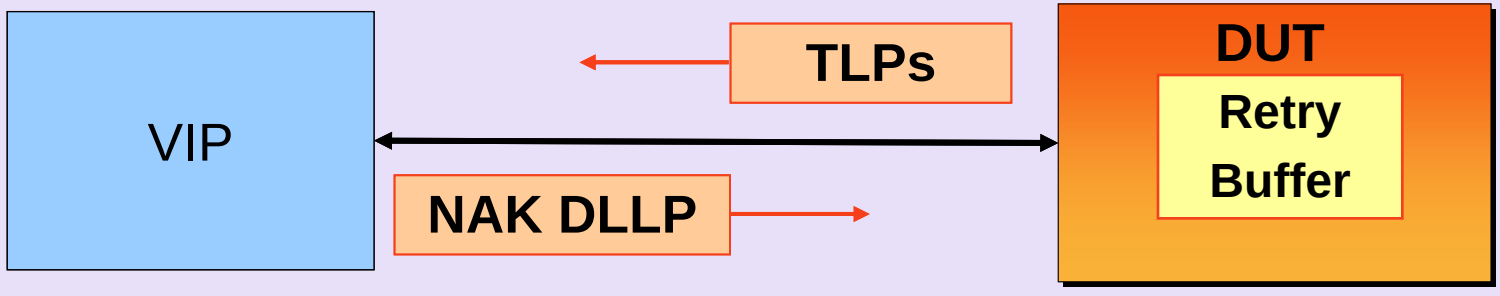
Technical Background

- Replay of TLPs from the DUT's Retry Buffer can happen for one of the following reasons

1. Expiration of the Replay Timer in DUT



2. Reception of a NAK DLLP by DUT



Case Study #2

Verification Challenge

- Multitude of underlying test scenarios must be considered
- Example: Subset of Replay Timer scenarios
  - ✓ Replay Timer resets and holds while there are no outstanding unacknowledged TLPs
  - ✓ Replay Timer resets and starts upon transmission/retransmission of the last TLP symbol
  - ✓ Upon reception of ACK DLLP, Replay Timer restarts if there are unacknowledged TLPs in the Retry Buffer and the received ACK acknowledges some TLPs in the Retry Buffer
  - ✓ Replay Timer must not advance during Link retraining

Case Study #2

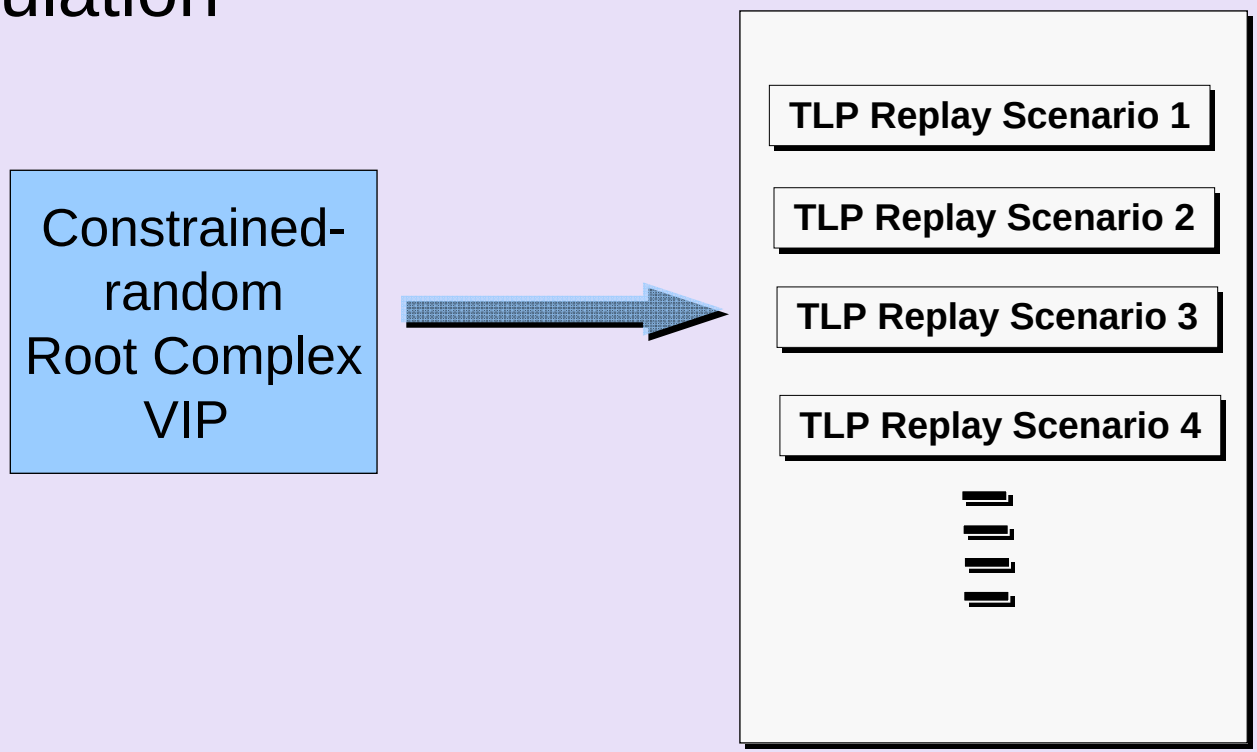
Verification Challenge (cont.)

- Example: Subset of NAK DLLP scenarios
  - ✓ Upon reception of NAK DLLP, all unacknowledged TLPs in the Retry Buffer are replayed starting with the oldest ones
  - ✓ The REPLAY_NUM counter increments at each replay of the same oldest TLP sequence number in the Retry Buffer
  - ✓ Transmitter retrains the link when REPLAY_NUM rolls over from 11b to 00b
- The TLP Replay Mechanism can only be declared fully verified when all underlying test scenarios have been fully exercised
  - ✓ Requires lots of work using directed testing

Case Study #2

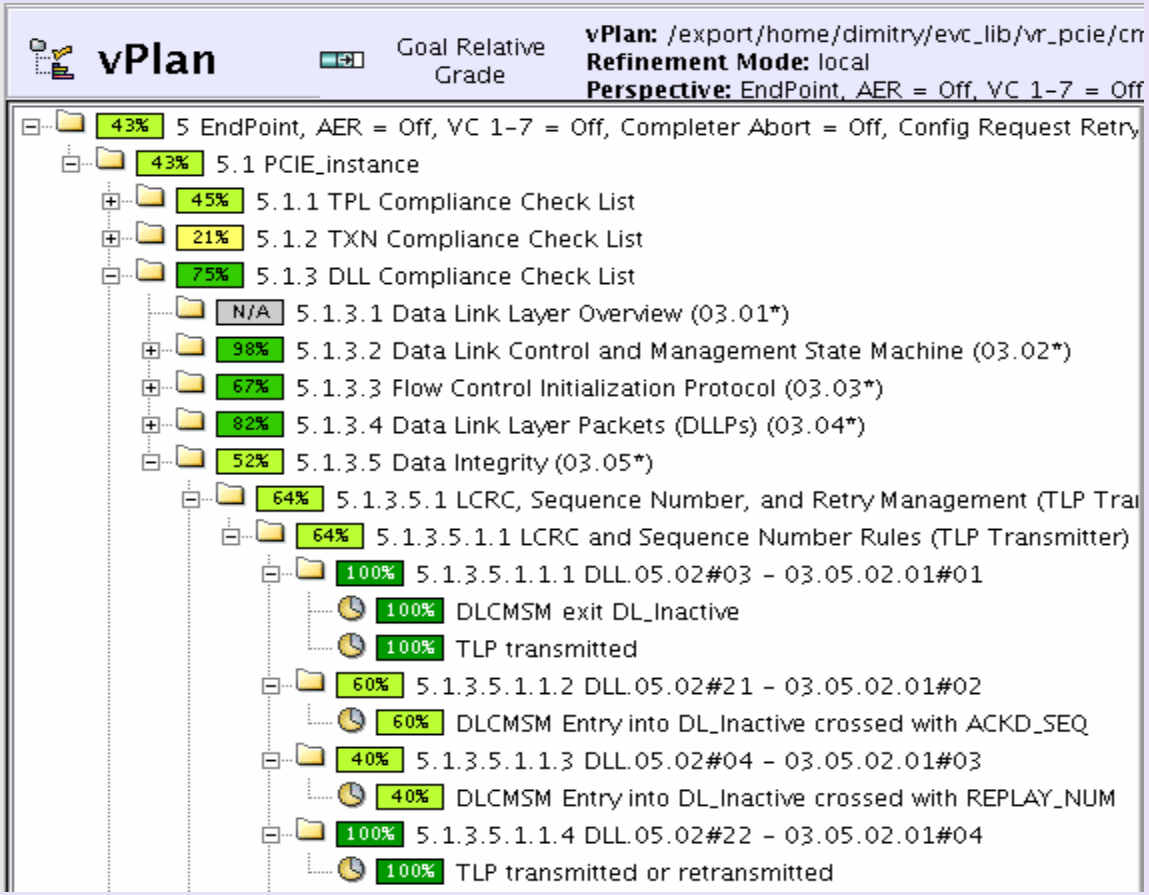
Implemented Solution

- Various constrained-random TLP Replay Mechanism scenarios are implemented in VIP
- These scenarios are randomly invoked during simulation



Case Study #2 Implemented Solution (cont.)

- Supplied coverage model is used to assess what test scenarios have or have not been exercised



Case Study #2

Implemented Solution (cont.)

- One of the built-in tests in the constrained-random VIP targeted the NAK DLLP caused TLP Replays
  - ✓ The DUT's Replay Timer was programmed to a very big value to prevent the Timer's expiration
  - ✓ The same value was programmed into the VIP Monitor to prevent false error firing by the VIP

Case Study #2 Test Description

- VIP's ACKing of incoming TLPs is disabled
- VIP NAKs TLPs sent by DUT
- The time period between NAK DLLPs is randomized to allow filling of the DUT Retry Buffer to different levels of capacity
- Sequence number of NAK DLLPs issued by the VIP have random distribution – some match the last acknowledged TLP's sequence number (if this repeats four times, the DUT should cause link retraining as a result of Replay_Num rollover), others are increments with respect to the last NAK's Sequence number (to allow filling of the DUT Retry Buffer without Link retraining)

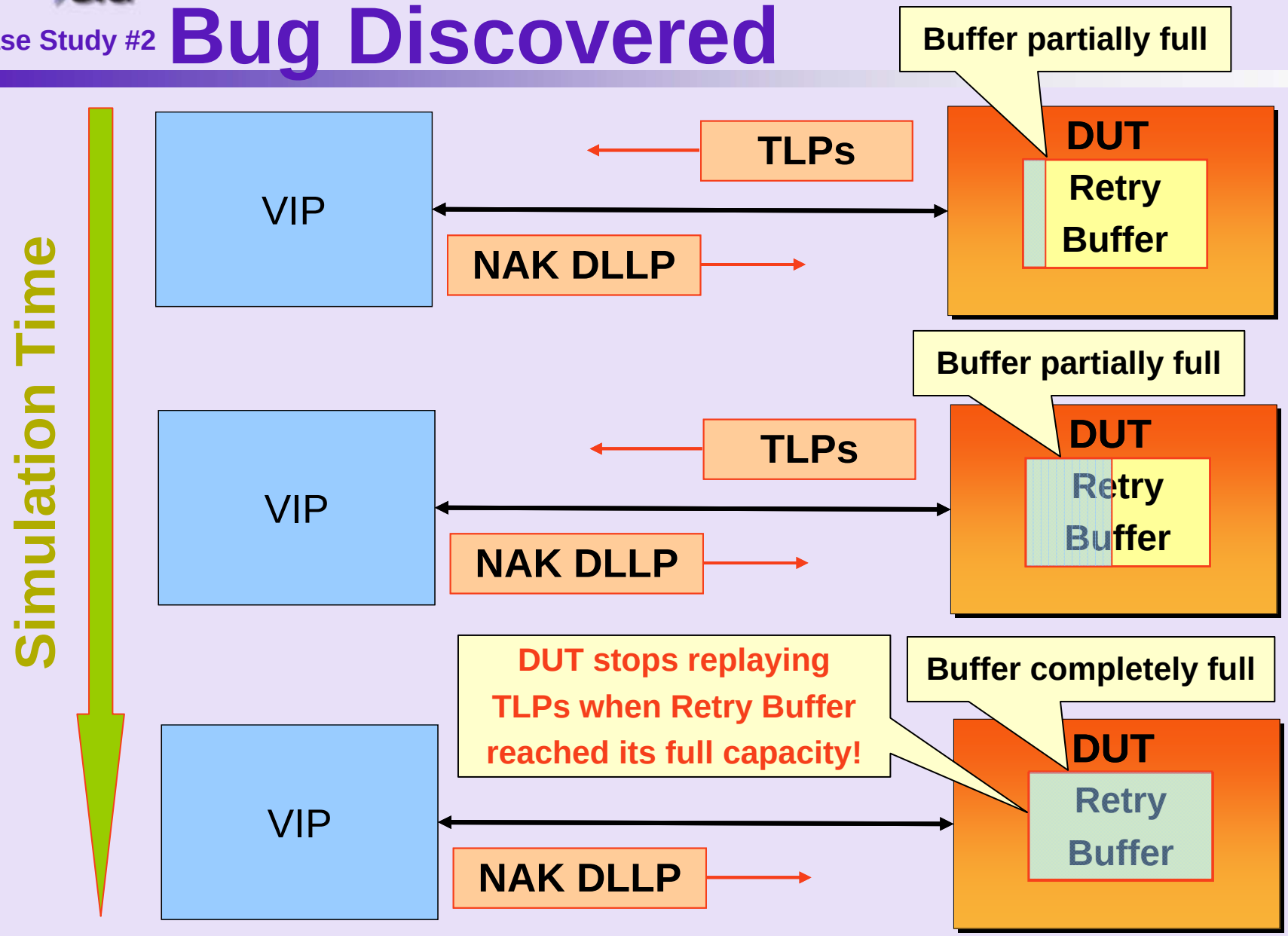
Case Study #2

Constrained-Random Verification Results

- Generally the DUT handled the NAK-caused TLP replays well – including replays after the Link retraining
- The Retry Buffer was gradually filled from low to high level of capacity
- However, the advanced constrained-random verification revealed a severe corner-case bug

Case Study #2

Bug Discovered



Case Study #2 Bug Analysis

- TLPs Replay Mechanism must work with Retry Buffer filled at any level of capacity including the fully filled Buffer
- Originally employed directed tests never hit the corner case when NAK DLLP was received when the Retry Buffer was full
- Bug in RTL logic was discovered: the FIFO Full flag was mistakenly gating the TLP Replay logic

Case #2 Summary

- Advanced constrained-random verification techniques enabled company to:
  - ✓ Find an unanticipated corner case scenario
  - ✓ Reveal a deeply-hidden functional fault
  - ✓ Increase product quality

Summary

- Advanced constrained-random coverage-driven verification consistently found functional faults missed by traditional directed testing approach
- Main advantages of constrained-random coverage-driven verification approach:
  - ✓ Creates test scenarios not thought of by humans
  - ✓ Finds deeply hidden corner-case bugs
  - ✓ Small number of tests cover a wide array of functional features
  - ✓ Functional coverage increased quality, productivity and predictability by driving verification process based on metrics (e.g., generated and missing test scenarios, compliance coverage completion)

Thank you for attending the
PCI-SIG Developers Conference 2008

For more information please go to
www.pcisig.com