"Citius, Altius, Fortius"

Three Steps towards PCIe® 2.0 Success

Jitendra Puri (JP)

Anurag Mangla, Nitin Gupta

Copyright © 2008, PCI-SIG, All Rights Reserved

Agenda

- Citius, Altius, Fortius
- Citius: Efficient Verification Environment
- Altius: Compliance Checklists
- Fortius: Common Pitfalls
- Towards PCIe® 2.0 success

Copyright © 2008, PCI-SIG, All Rights Reserved

Agenda

- Citius, Altius, Fortius
- Citius: Efficient Verification Environment
- Altius: Compliance Checklists
- Fortius: Common Pitfalls
- Towards PCIe® 2.0 success

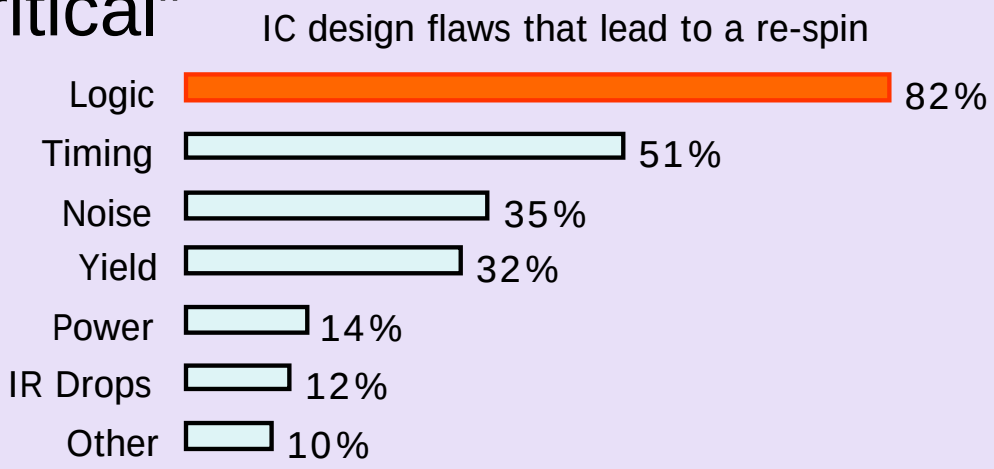
Copyright © 2008, PCI-SIG, All Rights Reserved

Citius, Altius, Fortius

Citius, Altius, Fortius

EETIMES ONLINE EDA Survey

- ~ 50% most recent chip project were late
- ~ 65% said completing functional verification is "very critical"



Design bugs are invariable TRUTHS

Copyright © 2008, PCI-SIG, All Rights Reserved

Citius, Altius, Fortius

- Just like an athlete is vying for Olympic glory by striving to go Citius (Faster), Altius (Higher), Fortius (Stronger), all of us here are striving to achieve Compliance faster, while building a Robust design.
- PCIe, being a complex protocol, presents a multitude of design and verification challenges.
- PCIe 2.0 brings with it a newer set of challenges.

Copyright © 2008, PCI-SIG, All Rights Reserved

Citius, Altius, Fortius

- The Three Steps:
  - ✓ Create an Efficient Verification Environment for Faster Verification Closure
  - ✓ Adhere to Compliance Checklists for Higher Interop
  - ✓ Avoid Common Pitfalls for Stronger Designs

Copyright © 2008, PCI-SIG, All Rights Reserved

Agenda

- Citius, Altius, Fortius
- Citius: Efficient Verification Environment
- Altius: Compliance Checklists
- Fortius: Common Pitfalls
- Towards PCIe® 2.0 success

Copyright © 2008, PCI-SIG, All Rights Reserved

Citius

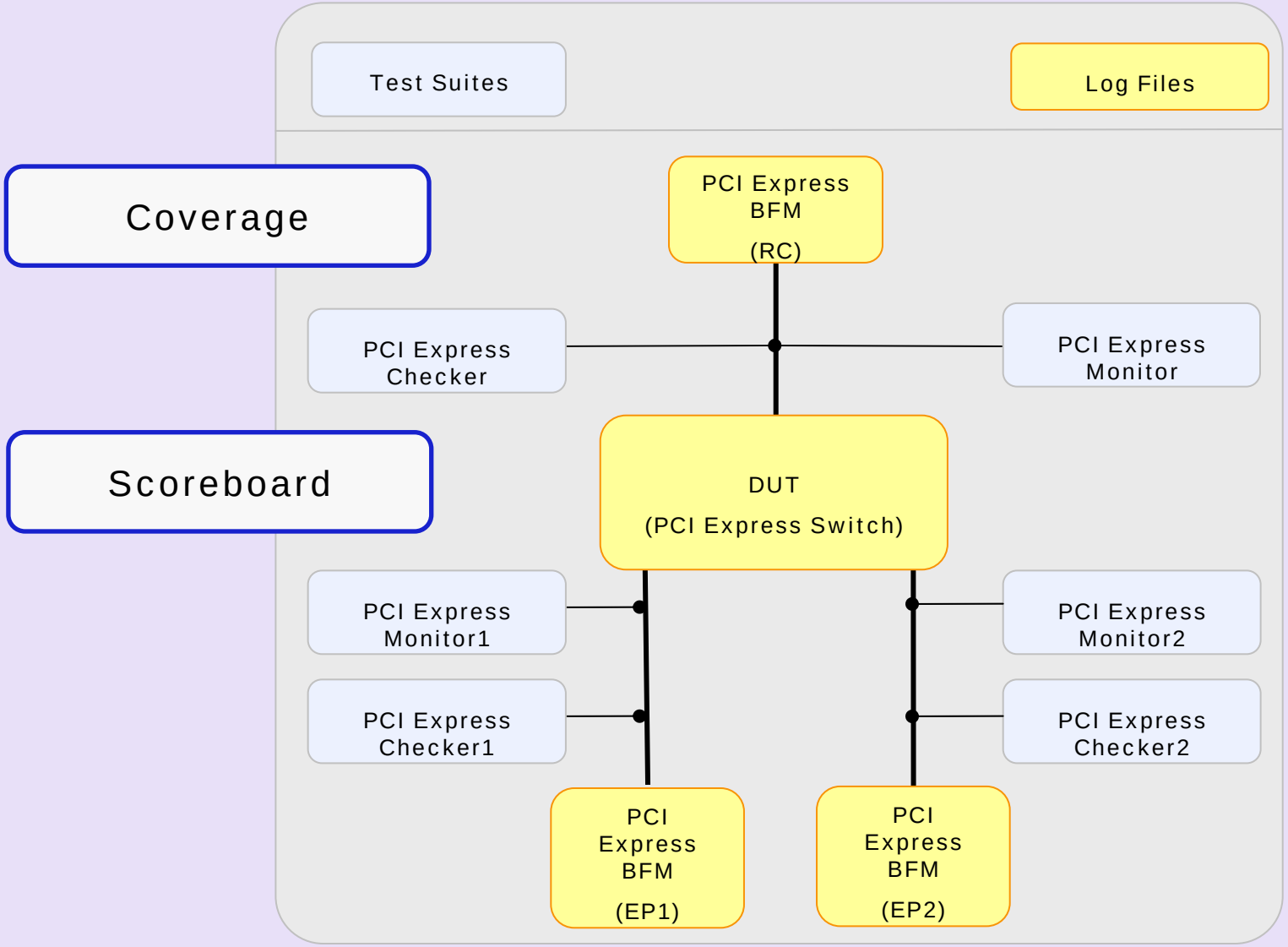
Faster Verification Closure

Efficient Verification Environment …

- Typical Verification Environment will have
  - ✓ Bus Functional Model
  - ✓ Protocol Checker
  - ✓ Bus Monitor
  - ✓ Scoreboard
  - ✓ Coverage bins
  - ✓ Test Suites

Copyright © 2008, PCI-SIG, All Rights Reserved

Efficient Verification Environment …



Efficient Verification Environment

- An Efficient Verification Environment must invariably adopt an HVL (Hardware Verification Language) and a Verification Methodology

  - ✓ User Developed/Proprietary
  - ✓ OVM (Open Verification Methodology)
  - ✓ VMM (Verification Methodology Manual)
  - ✓ eRM (e Reuse Methodology)

Efficient Verification Environment: Verification Methodology

- Verification Methodology Adoption
  - ✓ Methodologies provide many useful facilities for building powerful test benches
    - Transactions and components, environment, test
    - Configuration mechanism to control topology and run-time parameters
    - Sequential stimulus for a standard test writer interface
      - simplify creation of directed random stimulus
    - TLM (Transaction Level Modeling) API
    - Messaging facility

Copyright © 2008, PCI-SIG, All Rights Reserved

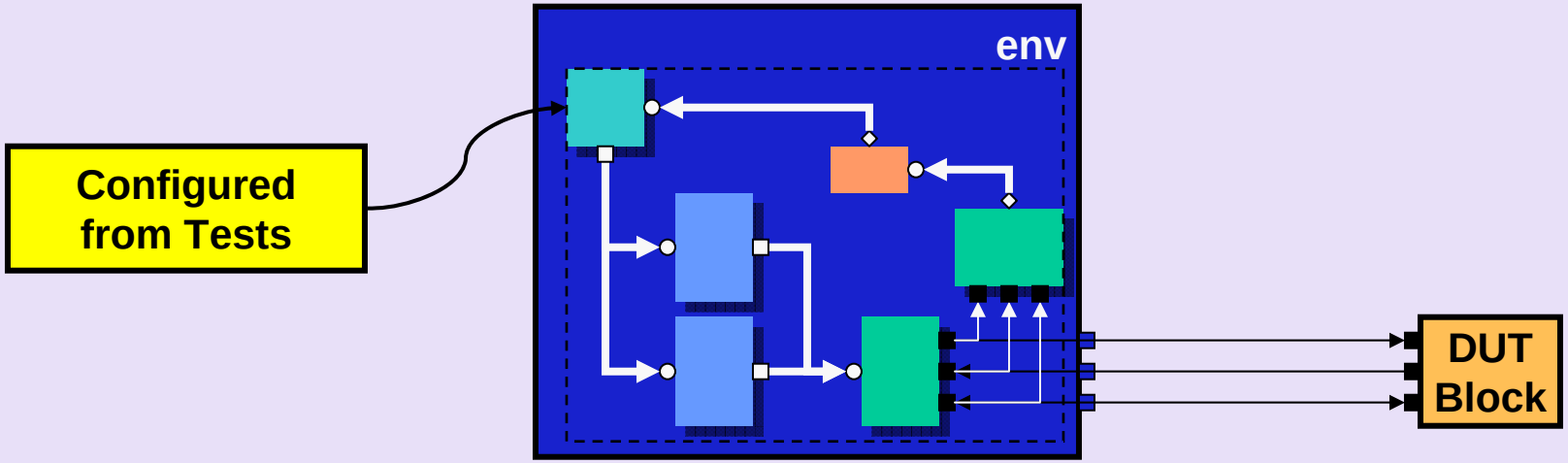
Efficient Verification Environment: Reuse

- Modularity + Configurability = Reuse
- Reuse:
  - ✓ Block Level to Chip Level
  - ✓ Across Projects

Copyright © 2008, PCI-SIG, All Rights Reserved

Efficient Verification Environment: Reuse

- Block Level to Chip Level
  - ✓ Reuse block-level Test Bench in system Test Bench
  - ✓ Environment may be instantiated hierarchically
  - ✓ Block-level environment contains all necessary block-level verification components
  - ✓ Extend environment as necessary to communicate with system-level test bench



Copyright © 2008, PCI-SIG, All Rights Reserved

Efficient Verification Environment: Reuse

- Across Projects
  - ✓Encapsulate components for reuse
    - Packages
  - ✓Reuse components in new environment
    - Additional stimulus & analysis
  - ✓Test customizes environment
    - Choose environment from library
    - Coordinate Stimulus
    - Additional tweaking
  - ✓Test instantiated by top-level module
    - Test chosen from library

Copyright © 2008, PCI-SIG, All Rights Reserved

Efficient Verification Environment: Score-Boarding

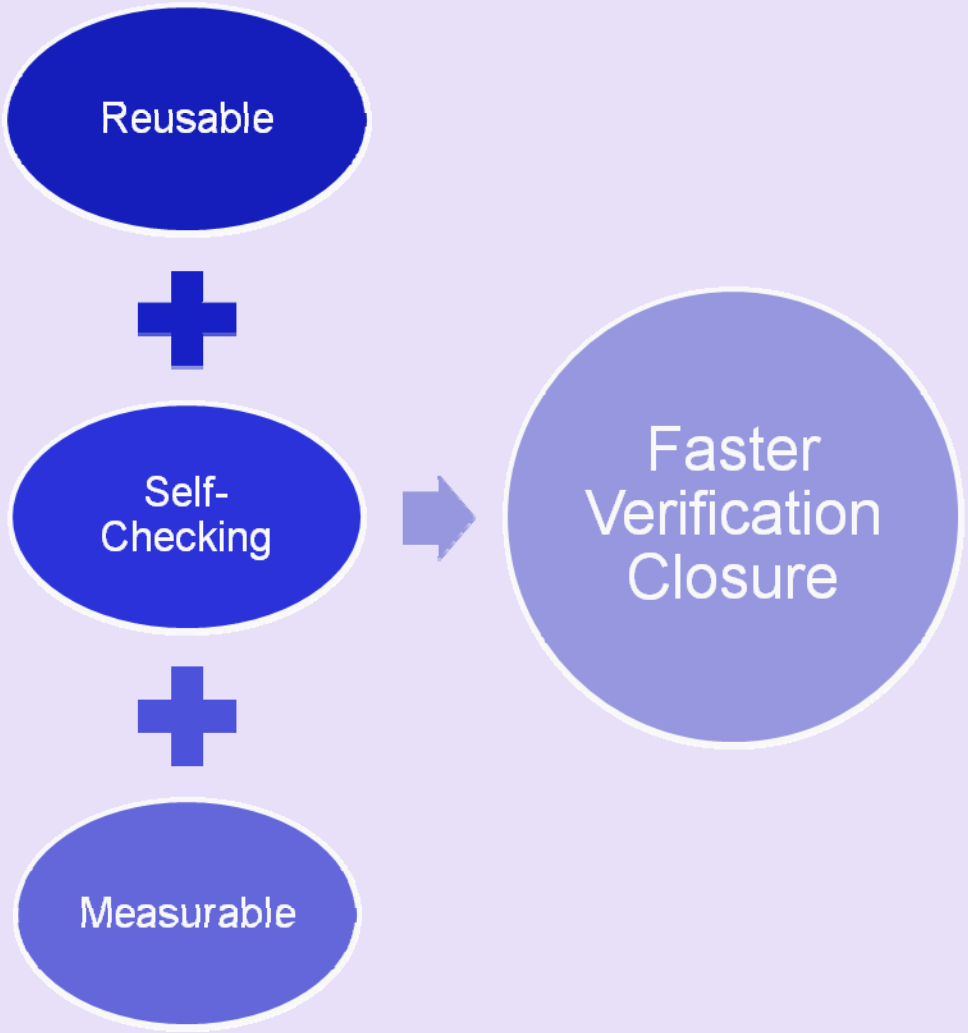
- End to End Checking → Self Checking
  - ✓ Extraneous and missing Packets
  - ✓ Data Integrity
  - ✓ Transaction Ordering
  - ✓ Design Latency

Copyright © 2008, PCI-SIG, All Rights Reserved

Efficient Verification Environment: Measurable Verification

- Define Goals
  - ✓ Functional coverage
  - ✓ Corner conditions
  - ✓ Key internal nodes toggle
- Coverage
  - ✓ Powerful hole targeting
  - ✓ Optimized for regression
  - ✓ Incremental adoption

Copyright © 2008, PCI-SIG, All Rights Reserved



Copyright © 2008, PCI-SIG, All Rights Reserved

Agenda

- Citius, Altius, Fortius
- Citius: Efficient Verification Environment
- Altius: Compliance Checklists
- Fortius: Common Pitfalls
- Towards PCIe® 2.0 success

Copyright © 2008, PCI-SIG, All Rights Reserved

Altius

Higher Interoperability

Copyright © 2008, PCI-SIG, All Rights Reserved

Leveraging Compliance Checklist

- Test suite based on Compliance Checklists should be a key component of the verification effort
- Compliance checklist to be discussed as part of Design Kick-Off Meetings
- Not very easy to derive finer points from a 400+ page specifications

Copyright © 2008, PCI-SIG, All Rights Reserved

Compliance Checklist: Test suite

- Compliance Suite available with 3rd Party Verification IPs
  - ✓ TXN: Transaction Layer
  - ✓ DLL: Data Link Layer
  - ✓ PHY: Physical Layer
  - ✓ PMG: Power Management
  - ✓ SYS: System/BIOS
  - ✓ CFG: Configuration

Copyright © 2008, PCI-SIG, All Rights Reserved

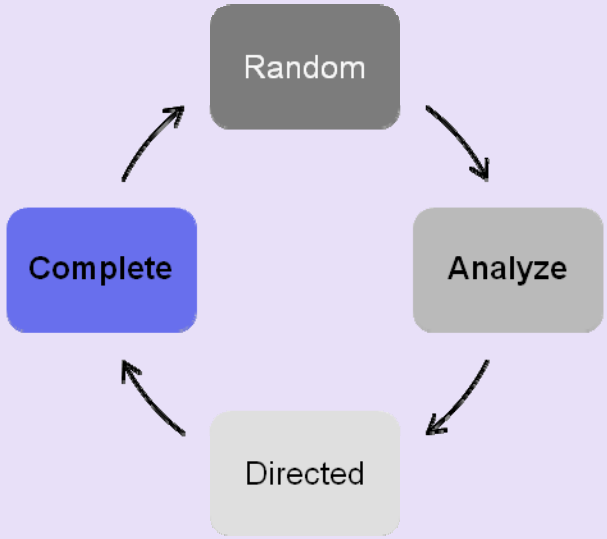
Compliance Checklist: Test suite

```
//-------------------------------------------------------------------------------------------
// GEN2.PHY.4.2.4.2#8
// An EIEOS must be sent before sending the first TS1 Ordered set
// in the sequence of TS1/TS2 ordered sets in GEN2.
//-------------------------------------------------------------------------------------------
// Moving to Gen2 speed
 `p0.do_cfg(`EX_PL_GEN2_DIR_SPEED_CHANGE,1'b1);
 @(`CHK0.RX.event_enter_l0);
@(nvs_ex_tests.bfm_idle);
  `p0.do_cmd(`EX_PL_PKT,0,0);
  fork:fork_54
          begin
            @(`CHK0.RX.event_chk_pl_eieos_rcvd);
            disable fork_54;
          end
          begin
            repeat (`EX_MAX_DELAY) @ (posedge `EX_TB.clk);
            print("log","Case 54 -> EP did not send an EIEOS before sending first TS1 OS");
            ->event_pl_gen2_compliance_error;
            disable fork_54;
          end
    join
    @(nvs_ex_tests.bfm_idle);
```


Copyright © 2008, PCI-SIG, All Rights Reserved

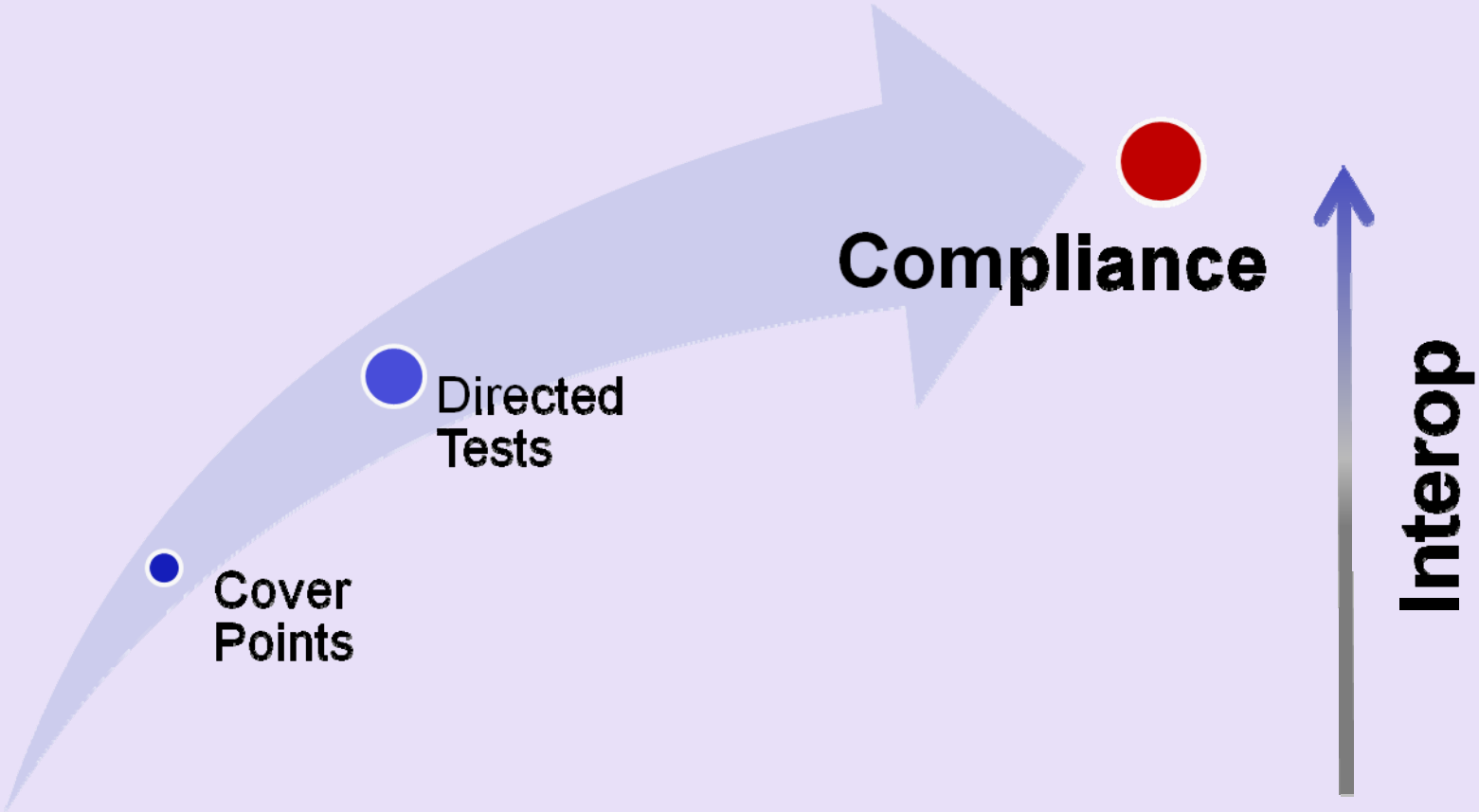
Compliance Checklist: Tracking

- Directed:
  - ✓ Make Compliance Checklist as part of the “Tracking Sheet”



- Random:
  - ✓ Define each of the checkpoints as a Cover Point

Copyright © 2008, PCI-SIG, All Rights Reserved



Copyright © 2008, PCI-SIG, All Rights Reserved

Agenda

- Citius, Altius, Fortius
- Citius: Efficient Verification Environment
- Altius: Compliance Checklists
- Fortius: Common Pitfalls
- Towards PCIe® 2.0 success

Copyright © 2008, PCI-SIG, All Rights Reserved

Fortius

Stronger Designs

Copyright © 2008, PCI-SIG, All Rights Reserved

Common Pitfalls

"Learn from the mistakes of others. You can't live long enough to make them all yourself."

Eleanor Roosevelt

- Common pitfalls based upon bugs found during Verification of several PCIe 1.x and PCIe 2.0 designs

Copyright © 2008, PCI-SIG, All Rights Reserved

Error Prone Areas in PCIe Designs …

- Based on the analysis of the bugs unearthed during the Verification of several designs, the Error prone areas are:

  ✓ Physical Layer
  - LTSSM design
    - Incorrect LTSSM state transitions
    - Setting/Resetting speed change variables
  - Recording Parameters
    - Data Rate Identifier
    - NFTS

Copyright © 2008, PCI-SIG, All Rights Reserved

Error Prone Areas in PCIe Designs …

- PL Ordered Set
  - EIEOS
  - EIOS
- Inferring Electrical Idle
  - Conditions to infer electrical idle
  - Inferred exit from electrical idle

Error Prone Areas in PCIe Designs …

- ✓ Data Link Layer
  - Flow Control
    - Update FCs not being sent for VCx
    - Credit considerations for Message Packets
  - DL_Inactive status consideration
    - Data Link Layer not getting reset on entry to DL_Inactive
  - Replay mechanism
    - Incorrect re-transmission from Retry buffer

Copyright © 2008, PCI-SIG, All Rights Reserved

Error Prone Areas in PCIe Designs …

- ✓ Transaction Layer
  - AER
    - Incorrect setting of configuration register bits
  - RCB Boundary handling
    - Incorrect interpretation of RCB parameter
  - Incorrect usage of Max_Read_Request_Size register in the receive logic

Error Prone Areas in PCIe Designs

- ✓ Power Management
  - ASPM L0s entry
    - Tx_L0s and Rx_L0s not kept independent
  - ASPM L0s exit
    - Incorrect number of N_FTS sent

Copyright © 2008, PCI-SIG, All Rights Reserved

Bugs Hall of Fame

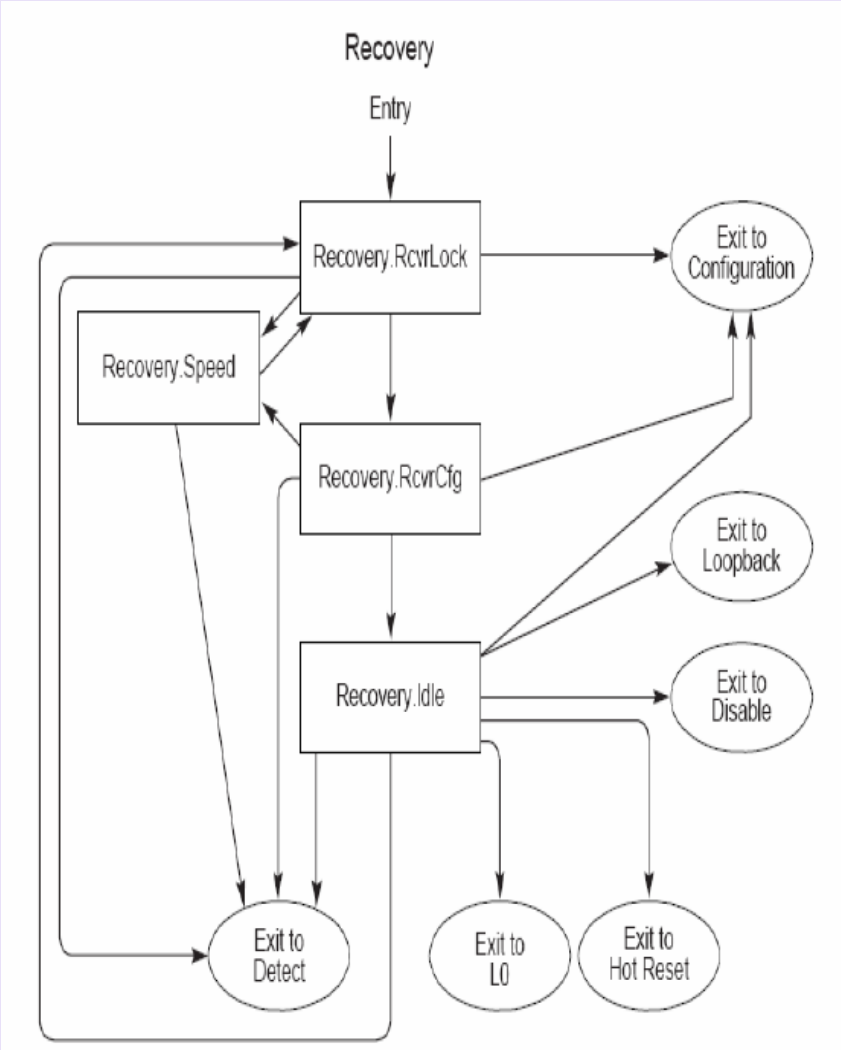
- **Receiver advertised 128 outstanding unused header credits**
  - ✓ PCIe 1.1 compliant switch design used to advertise 64 credits
  - ✓ On migrating to PCIe 2.0, the designer doubled “everything”, including credits issued
  - ✓ Spec. says, “A receiver must never cumulatively issue more than 2047 outstanding unused credits to the Transmitter for data and 127 for header.”
  - ✓ Impact: The Other device throttles

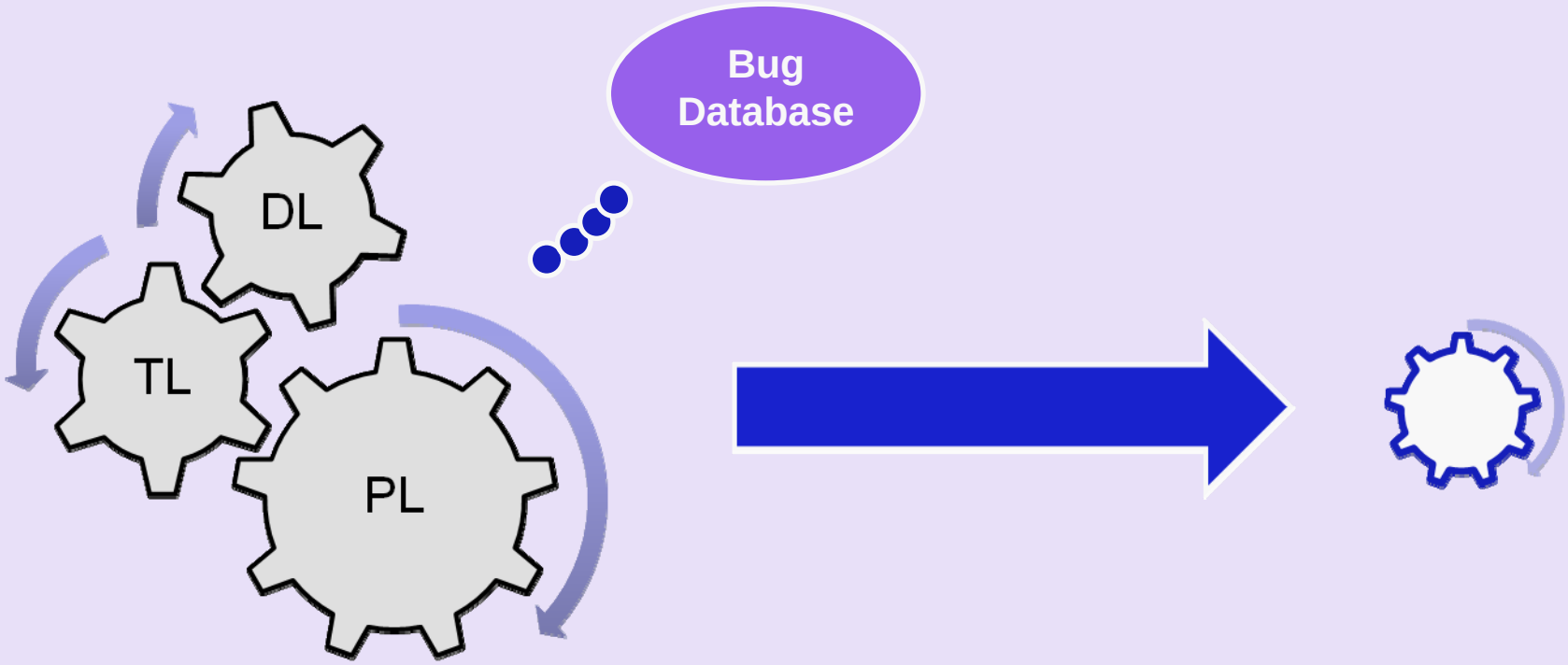
Bugs Hall of Fame

- **Various timers not set properly if link is established with lane count < maximum supported**
  - ✓ Assuming maximum supported lane count to be 8, but link is established at 4, various timers are still kept as per 8 lane configuration
    - AckNak_LATENCY_TIMER
    - REPLAY_TIMER
- **Timers at the TL and DL not scaled with the clock**
  - ✓ same timeout values implemented at both 2.5 GT/s and 5.0 GT/s

Bugs Hall of Fame

- **Incorrect LTSSM speed change transitions**
  - ✓ Transition to and from Recovery.Speed
    - – Device enters Recovery.Speed before transmitting 32 TS OS and changes speed to 5.0 GT/s
    - – Other device enters Detect at 2.5 GT/s
  - ✓ Impact:
    - – Loss of symbol lock
    - – Link failure

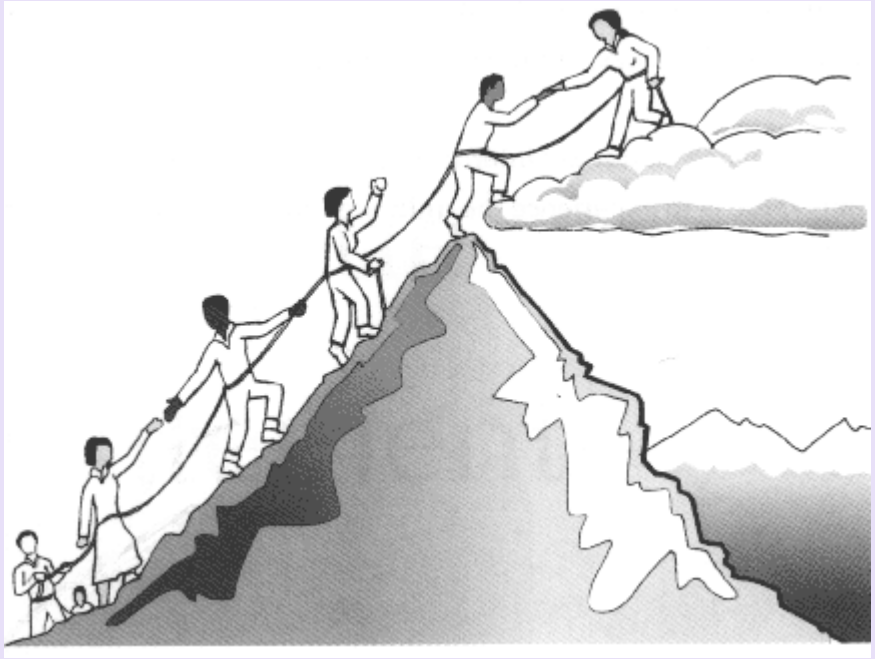




Copyright © 2008, PCI-SIG, All Rights Reserved

Towards PCIe 2.0 Success

- Achieving PCIe success is a challenge
- Follow the Citius, Altius, Fortius plan for success



Thank you for attending the
PCI-SIG Developers Conference 2008

For more information please go to
www.pcisig.com

Copyright © 2008, PCI-SIG, All Rights Reserved