
PCI
SIG®

Common Pitfalls in PCIe® 2.0 Migration

Jitendra Puri (JP)

Anurag Mangla, Nitin Gupta

Agenda

- Introduction
- Migration from PCI™/PCI-X™ to PCIe® onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Introduction

- Migration to PCIe 2.0 is apparently seamless and easy
  - ✓ But it is not without its own design and verification challenges
- Common pitfalls observed while verifying the PCIe 2.0 design migration efforts
- Potential pitfalls that designers intending to initiate a migration should avoid, to achieve faster time to market
- Suggested Defect Prevention steps

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Migration From PCI/PCI-X to PCIe onto PCIe 2.0…

- PCI limited to 64-bit, 66-MHz
- PCI-X mode-1 limited to 64-bit,133-MHz

- PCIe is an effort to move everything to a common IO standard
- PCIe is offering high bandwidth per pin
- Scalable performance via aggregated Lanes

Migration From PCI/PCI-X to PCIe onto PCIe 2.0

- PCIe 2.0 doubles the interconnect bit rate from 2.5 GT/s to 5 GT/s in a seamless and compatible manner
- Increases the aggregate bandwidth of a 16-lane link to approximately 16 GB/s
- Narrower interconnect links to achieve high performance while reducing cost

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Recap of Error Prone Areas in PCIe Design…

- 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
  - ✓ 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

Recap of Error Prone Areas in PCIe Design

- Replay mechanism
  - Incorrect re-transmission from Retry buffer

✓ 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

✓ Power Management

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

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Recap of The Most Occurring Bugs in PCIe Design…

- ✓ The received SSPL message not taken into account while sending update FCs.
  - Majority of the EP design specs say that the “data received as part of SSPL would be ignored”
  - Somehow the whole of packet gets ignored including the Header and the Data Credits consumed
  - The UpdateFC going back does not take into account the above mentioned Credits
  - The Other device finds lesser available Credits with the Target device

Recap of The Most Occurring Bugs in PCIe Design…

- Impact of the “SSPL” bug on the system…
  - ✓ Caused the TX of the other device to believe that lesser Header/Data credits are available
  - ✓ Direct impact on performance
  - ✓ In one of the designs, the system was to have 64-byte and 128-byte Writes. But with the Credit not being freed in terms of Update for the SSPL, system would see only 64-byte Writes, and never 128-byte Writes…. Performance Degradation….

Recap of The Most Occurring Bugs in PCIe Design…

- ✓ 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

Recap of The Most Occurring Bugs in PCIe Design

- Impact of “Timer” bug on the system…

  - ✓ AckNak_LATENCY_TIMER
    - The ACK/NAK is still all right, it is scheduled faster
    - This may cause the TX to come out of L0s faster than normal

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Error Prone Areas in PCIe 2.0 Design…

- Impact of PCIe 2.0 changes felt mostly on PHY
- More error prone areas that need special attention are the “newer” concepts:

✓ LTSSM Speed Change Transitions
  - Setting/Resetting speed change variables
  - Transition to and from the new state Recovery.Speed

✓ Recording Parameters
  - Data Rate Identifier
  - NFTS

Error Prone Areas in PCIe 2.0 Design…

- ✓ Link Width Upconfigurability
  - Optimistic lane upconfiguration
  - Bandwidth change notification
- ✓ PL Ordered Set
  - EIEOS
  - EIOS
- ✓ Inferring Electrical Idle
  - Conditions to infer electrical idle
  - Inferred exit from electrical idle

Error Prone Areas in PCIe 2.0 Design

- ✓ Newer Timeouts
  - TL and DL Time out
  - LTSSM Time outs
- ✓ Function Level Reset (FLR)
  - Scheduling of FLR
  - Resetting of Configuration Space Register

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Bug Details…

- LTSSM Speed Change Transitions

  - ✓ Setting/Resetting LTSSM variables

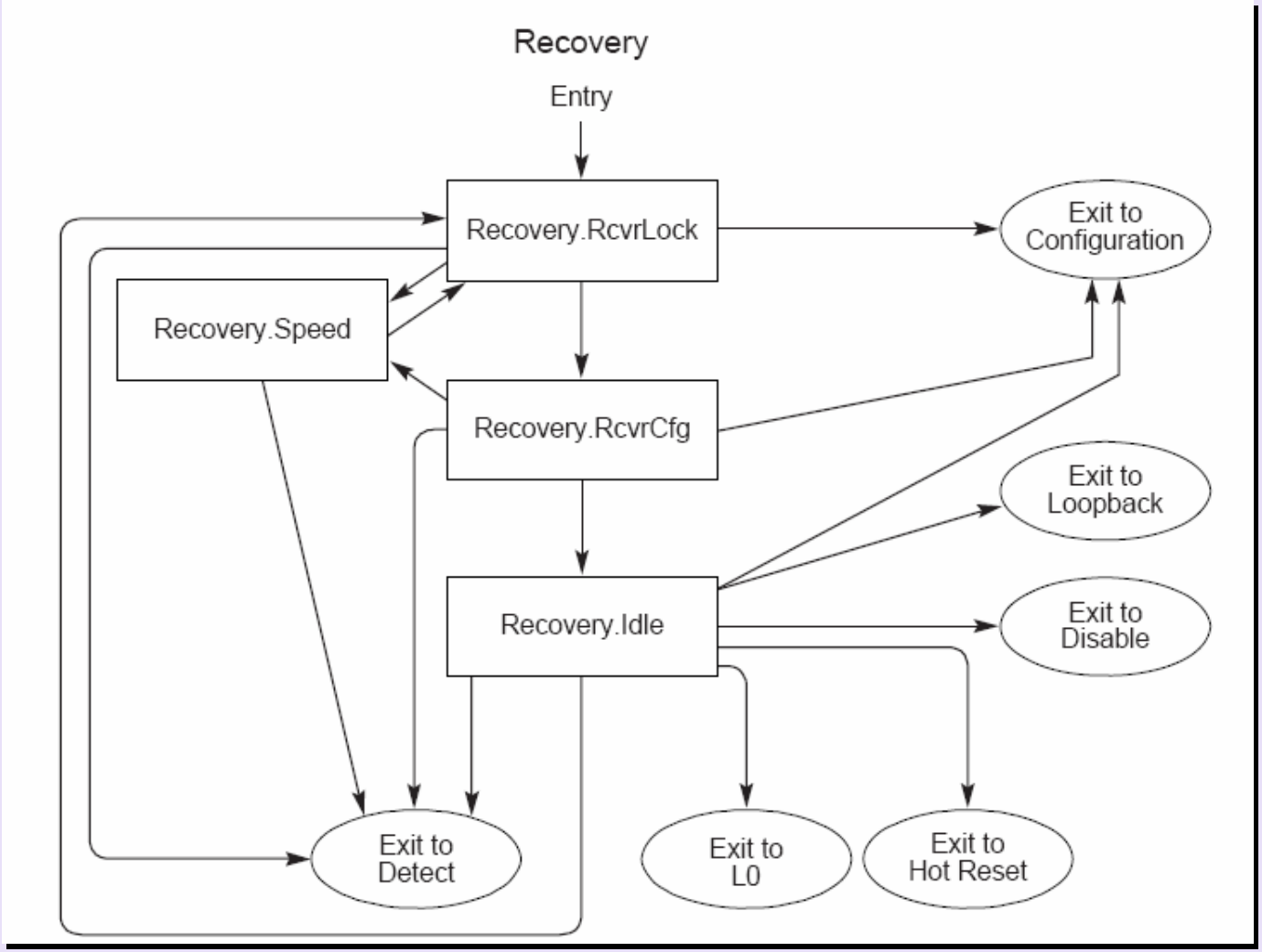
	directed_ speed_ change	successful_ speed_ negotiation	changed_ speed_ recovery	upconfigure_ capable	idle_to_rlock
SET	L0/ Recovery.Lock	Recovery.RcvrCfg	Recovery.Speed	Configuration. Complete	Entry to Recovery.RcvrLock from Configuration.Idle or Recovery.Idle
RESET	Exit from Recovery.Speed	Recovery.RcvrLock	(i) Recovery. Speed (ii) Exit from Recovery.RcvrCfg to Configuration (iii) L0	Configuration. Complete	L0

Bug Details…

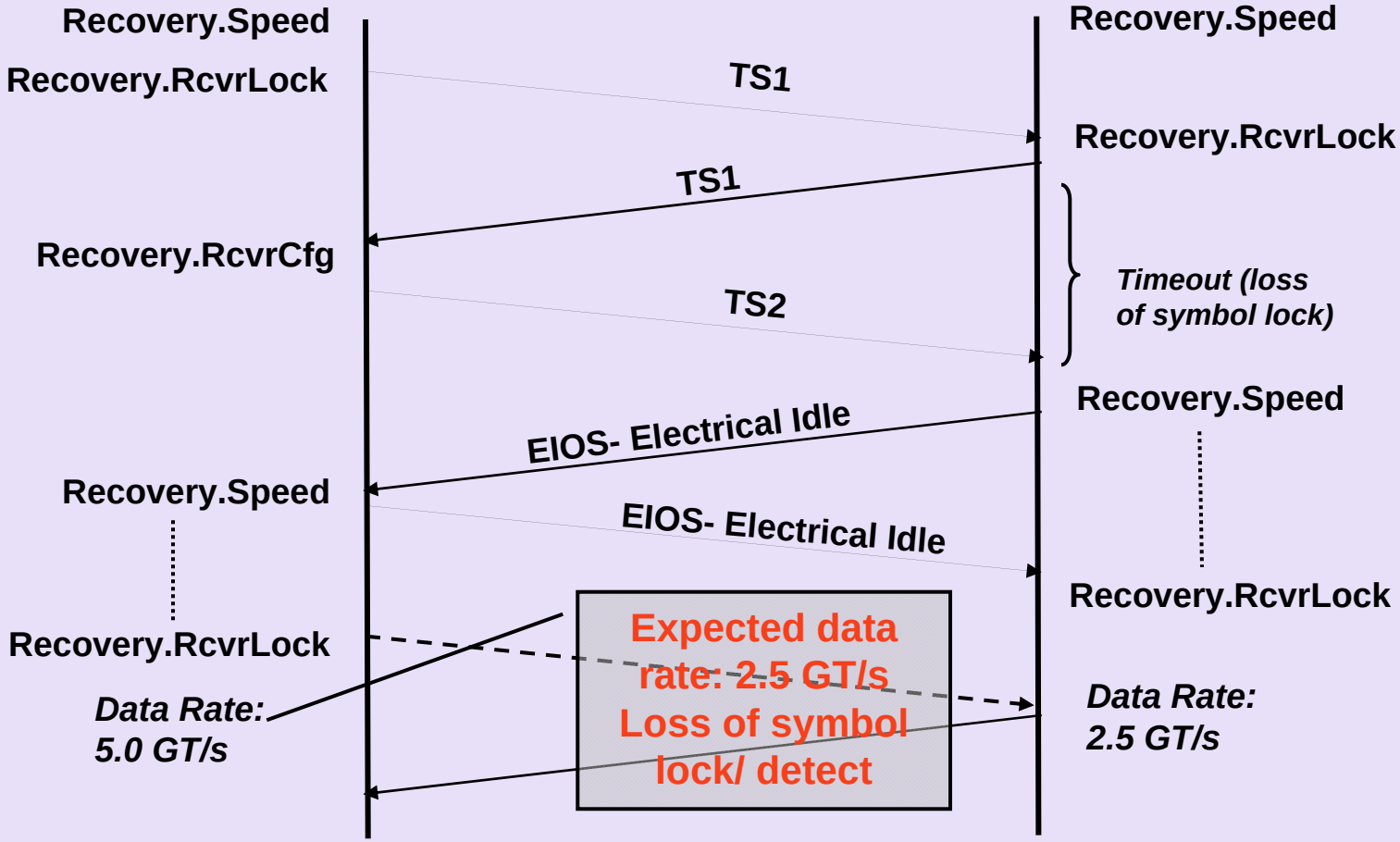
- LTSSM Speed Change Transitions

  - ✓ Transition to and from the new state Recovery.Speed
    - – Exit from Recovery.Speed without the other device entering electrical idle
      - Loss of symbol lock
      - Link failure
  - ✓ Incorrect speed of operation after exit from Recovery.Speed

Bug Details…



Bug Details…



Bug Details…

- Data Rate
  - ✓ Component must record the data rate supported by the other device in Configuration.Complete, Recovery.RcvrLock
    - Device advertises 5.0 capability in Polling state
    - Changes supported value to 2.5 in Configuration
    - Opposite device still initiates speed change
- NFTS
  - ✓ Component permitted to advertise different N_FTS rates at different speeds
    - Value not latched in Recovery at 5.0 GT/s.
    - Device sends less FTS than required by the other device.
    - Receiver's symbol lock circuitry fails

Bug Details…

- Link Width Upconfiguration
  - ✓ Initiating upconfiguration of the link width beyond the maximum negotiated width
  - ✓ Autonomous bandwidth change not notified
  - ✓ Link Bandwidth Notification registers not updated

Bug Details…

- EIOS
  - ✓ Not sent before entering electrical idle
  - ✓ 2 EIOS not sent at speeds >2.5 GT/s
    - – Required by electrical idle detection circuitry
- EIE (K28.7) symbol and EIEOS
  - ✓ Difficult to scale capabilities of Exit From Idle circuits with data rates
  - ✓ EIE patterns guarantee voltage and time requirements

Bug Details…

- EIEOS
  - ✓ Scheduling
    - Transmitted at wrong intervals of TS1 and TS2
    - eies_interval_counter reset in Recovery.RcvrCfg
  - ✓ Failure to understand exit from electrical idle
- EIE symbol not prepended to FTS.
  - ✓ Less than four transmitted

Bug Details…

- Inferring Electrical Idle
  - ✓ L0 state: idle not inferred even when SKP/UFC is missing the 128µs window
  - ✓ Exit from electrical idle determined as a converse of inferring electrical idle
    - The only condition for electrical idle exit: reception of EIEOS

COM	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	TS1
COM	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	TS1
COM	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	TS1
COM	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	EIE	TS1

Bug Details…

- New Timers
  - ✓ 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
    - Specifically:
      - UFC timer
      - Replay Timer
      - Ack Latency Timer

Bug Details…

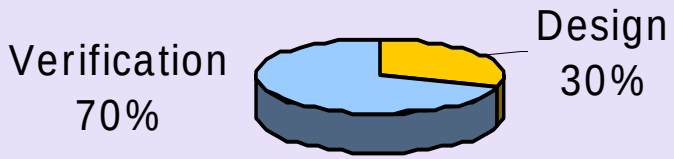
- ✓ Function Level Reset (FLR)
  - Initiated before sending out the completion of the configuration write
  - Resetting of configuration registers
    - Captured Slot Power Limit Value in the Device Capabilities register
    - Captured Slot Power Limit Scale in the Device Capabilities register
    - Max_Payload_Size in the Device Control register
    - Active State Power Management (ASPM) Control in the Link Control register
    - Read Completion Boundary (RCB) in the Link Control register
    - Common Clock Configuration in the Link Control register
    - Extended Synch in the Link Control register
    - Enable Clock Power Management in the Link Control register
    - Hardware Autonomous Width Disable bit in Link Control register
    - Hardware Autonomous Speed Disable bit in the Link Control 2 register
    - Registers in the Virtual Channel Capability structure
    - Registers in the Multi-Function Virtual Channel Capability structure

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Defect Prevention…

- The real Mantra is “Defect Prevention”
- Aim is to prevent injection of these bugs at the first place itself
- Spend some more time in Design phase and drastically cut the Verification efforts
- Review error prone areas more thoroughly

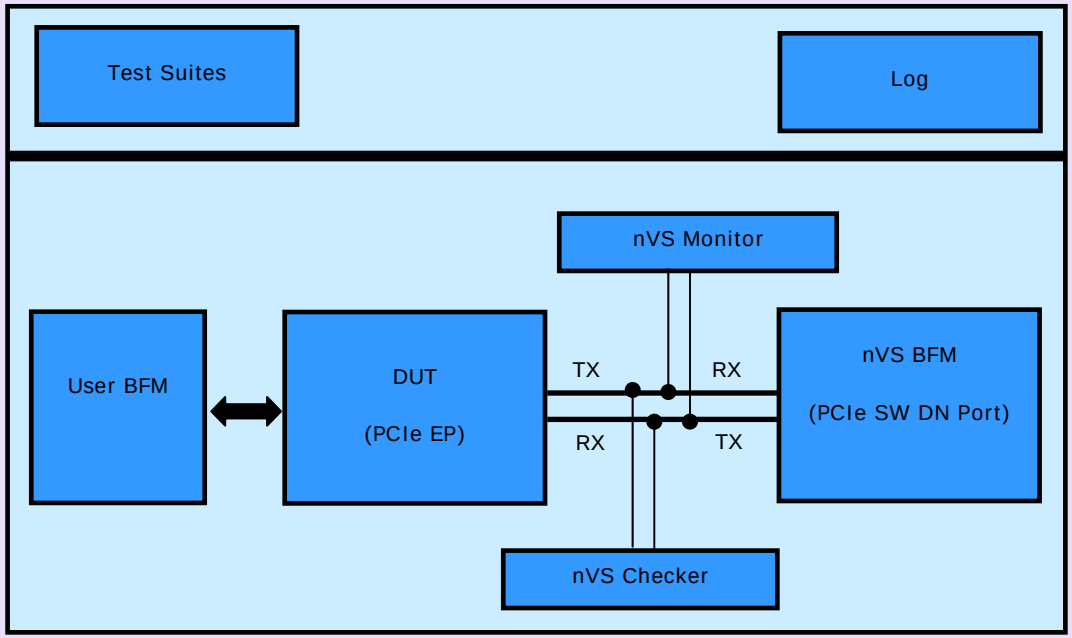


Defect Prevention

- Use proven Verification IP
  - ✓ Having Compliance Test Suite
  - ✓ Extensive error injection and detection capabilities
  - ✓ Ability to create any real life test scenario
  - ✓ Providing coverage reports
- Test suite based on traceability matrix & compliance checklist should be a key component of the verification effort
- Verification Methodology Adoption

Defect Prevention: Verification IP

- Typical Verification environment will have
  - ✓ Bus Functional Model (to generate and receive traffic)
  - ✓ Protocol Checker
  - ✓ Bus Monitor (to show the formatted log)
  - ✓ Test Suites (including Compliance)



Defect Prevention: Compliance Suite

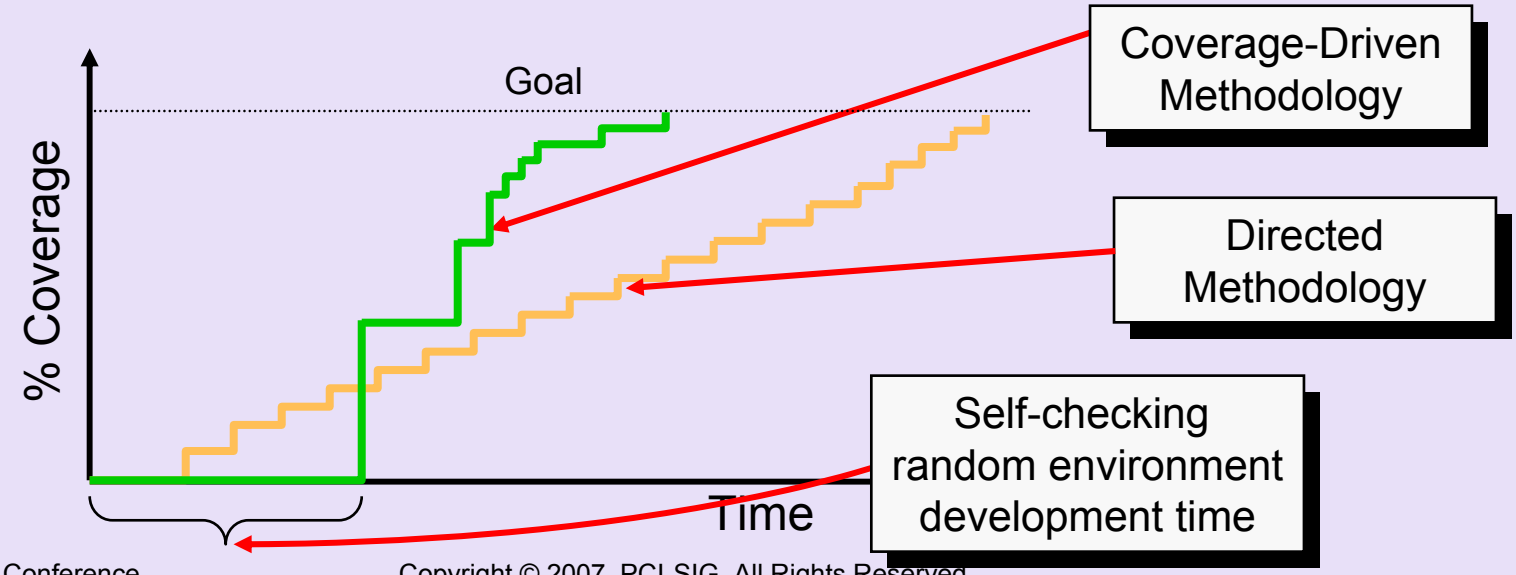
- Compliance Suite available with 3rd Party Verification IPs
- Directed Test Suite based on the Compliance Checklist
- Map each and every check-point to a test case
- Self-checking Test case
- Randomness in even in the directed tests to generate a newer test pattern in every run

Defect Prevention: Verification Methodology Adoption

- AVM : Advanced Verification Methodology
- VMM : Verification Methodology Manual

✓ Constrained Random

✓ Functional Coverage-Driven Verification



Defect Prevention: Coverage Matrices

- LTSSM

	DET.QUIET	DET.ACTIV	POL.ACTIV	POL.CMPLI	POL.CFGIR	POL.SPEED	CFG.LWST	CFG.LWAC	CFG.LNAC	CFG.LNWT	CFG.CMPLT	CFG.IDLE	RCV.RLOCK	RCV.SPEED	RCV.RCFG	RCV.IDLE	L0	RXL0s.EN	RXL0s.ID	RXL0s.FT	TXL0s.EN	TXL0s.ID
CFG.LN.AC	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
CFG.LN.WT	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
CFG.CMPLT	0	--	--	--	--	--	--	--	--	--	--	8	--	--	--	--	--	--	--	--	--	--
CFG.IDLE	0	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	8	--	--	--	--	--
RCV.RLOCK	1	--	--	--	--	--	0	--	--	--	--	--	--	1	23	--	--	--	--	--	--	--
RCV.SPEED	0	--	--	--	--	--	--	--	--	--	--	--	12	--	--	--	--	--	--	--	--	--
RCV.RCFG	6	--	--	--	--	--	0	--	--	--	--	--	--	9	--	8	--	--	--	--	--	--
RCV.IDLE	0	--	--	--	--	--	0	--	--	--	--	--	0	--	--	--	8	--	--	--	--	--
L0	0	--	--	--	--	--	--	--	--	--	--	--	15	--	--	--	--	0	--	--	0	--
RXL0s.EN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	0	--	--	--
RXL0s.ID	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	0	--	--
RXL0s.FT	--	--	--	--	--	--	--	--	--	--	--	--	0	--	--	--	0	--	--	--	--	--
TXL0s.EN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	0	--
TXL0s.ID	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	0

Agenda

- Introduction
- Migration from PCI/PCI-X to PCIe onto PCIe 2.0
- Recap of Error Prone Areas in PCIe Design
- Recap of The Most Occurring Bugs in PCIe Design
- Error Prone Areas in PCIe 2.0 Design
- Bug Details
- Defect Prevention
- Summary

Summary

- PCIe has been a natural upgrade for PCI/PCI-X designers
- “To Err is Human”… the bugs are going to be there !
- Review error prone areas more thoroughly
- Use back-to-back operation of Verification IP to improve understanding
- Use Verification IP with proven ability to detect bugs
- Compliance test suite to be one of the key components of the verification effort

Thank you for attending the
PCI-SIG Developers Conference 2007.

For more information please go to
www.pcisig.com

Common Pitfalls in PCIe 2.0 Migration

Jitendra Puri (JP)

Anurag Mangla, Nitin Gupta


PCI
SIG®