
# System Implementation Challenges for PCIe® 1.1 and 2.0

Olga Buchonina

Staff Applications Engineer

AMCC

# Disclaimer

**Presentation Disclaimer:** All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG®.

# Agenda

- Overview and Background
- System implementation common mistakes
- Hardware implementation issues and solutions
- Software common issues and solutions
- Performance optimization
- Summary

# Overview and Background

- PCI-Express is now proliferated across a wide range of system applications
- Designers, producers are facing challenges implementing PCI Express at the system level
- We will concentrate only on most frequently issues faced during PCI Express implementation

# System Implementation: Common Issues

- Component Selection for the board:
  - ✓ Topography
  - ✓ Clock generator selection: LVPECL, PECL, LVDS, CMOS, CML
  - ✓ ICL selection
  - ✓ Clock generation on the board
  - ✓ PHY considerations
  - ✓ Driver strength
  - ✓ Noise with/without Spread Spectrum
  - ✓ TX/RX trace lengths
  - ✓ Running BIST testing
  - ✓ Running compliance testing
  - ✓ Running HSpice simulations

# System Implementation Common Mistakes

- ✓ Using Oscilloscope to run noise analysis
- ✓ Component placement on the board
- ✓ Running system verification with oscilloscope and PCI Express Analyzer/Exerciser

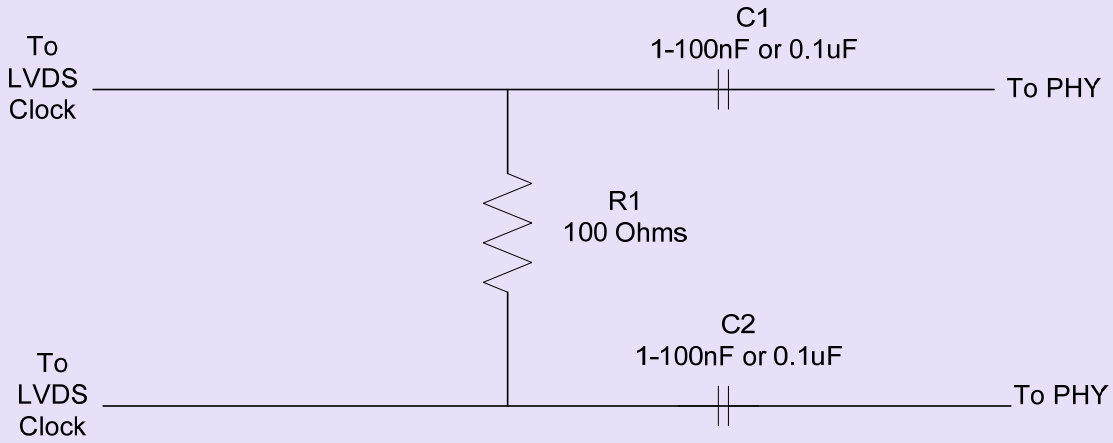
# Topography

- PCI Express most common application is single function device
  - PCI Express 2.0 use of multi-function is becoming more common
- Use of Multiport switches since 2006 is steadily growing
  - Often many issues arise with device enumeration in complex topologies

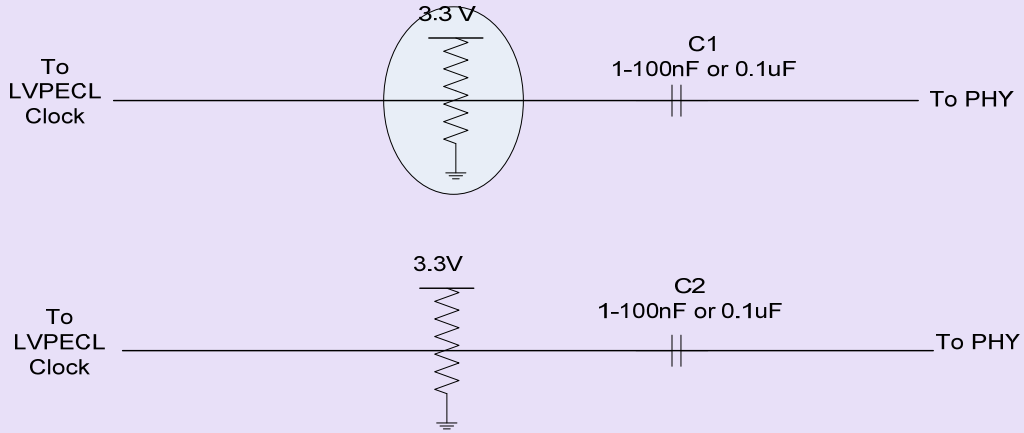
# Clock Reference Selection: CML,PECL, LVDS, CMOS,

- Most PCI Express Phy interfaces support CML, PECL, LVPECL and LVDS
  - Use of CMOS is less widespread when it comes to systems
- Very common mistake: understanding of the reference clock and how it needs to be generated
- Examples of common clock sources:
  - ✓ End Point uses Motherboard clock
  - ✓ End Point and Root Port are using different clock sources
  - ✓ Synchronous clock vs asynchronous clock
  - ✓ Setup correct CDR(clock data recovery) requirements for the system
  - ✓ Differential clock versus single ended clock
- Some very common issues:
  - ✓ Selecting termination for reference clocks
  - ✓ Termination and use of Spread Spectrum

# Clock Reference Selection




Recommendation: Do not use 0 Ohm resistor




- Cases where 0.1uF cap is not optimal
- Use of 0.01uF component is very frequently too low for AC coupling
- Best practice is to use 0.1uF.

# Clock Components

- ✓ PCI-SIG recommends Motherboard clock reference of 100 to 125Mhz
- ✓ PCI Express 2.0 uses nominal frequency of 100MHz +/- 300ppm on a single-ended swing of the reference clock
- ✓ Common issues:
  - Use of Spread spectrum versus non-spread spectrum
    - PCI Express 1.0a and 1.1 use of spread spectrum is optional
    - Ensure that PCIe 2.0 implementations have support during system architecture for spread spectrum
  - Use of a clock generator that is not PCI compliant
    - May cause issues with jitter, BER, etc.

# PHY Considerations

- Pre-silicon verification has very limited ability to mimic phy operation
  - ✓ unless mixed signal verification environment is deployed
- Analog settings for PHY need to be taken into consideration during chip physical design placement
  - ✓ Signal routing
  - ✓ PLL nearness
  - ✓ Noise
  - ✓ Timing analysis
- Device post-silicon validation finds optimum settings, however in some cases there are design incompatibilities:
  - ✓ Support of power management
  - ✓ Hot plug support
  - ✓ Post/Pre-emphasis range settings
  - ✓ Amplitude level
  - ✓ Compliance patterns
- During board layout Phy settings such as post/pre-emphasis can be offset
  - ✓ signal trace lengths
  - ✓ increasing/decreasing space on control signals

# Driver Strength, Noise with/without Spread Spectrum

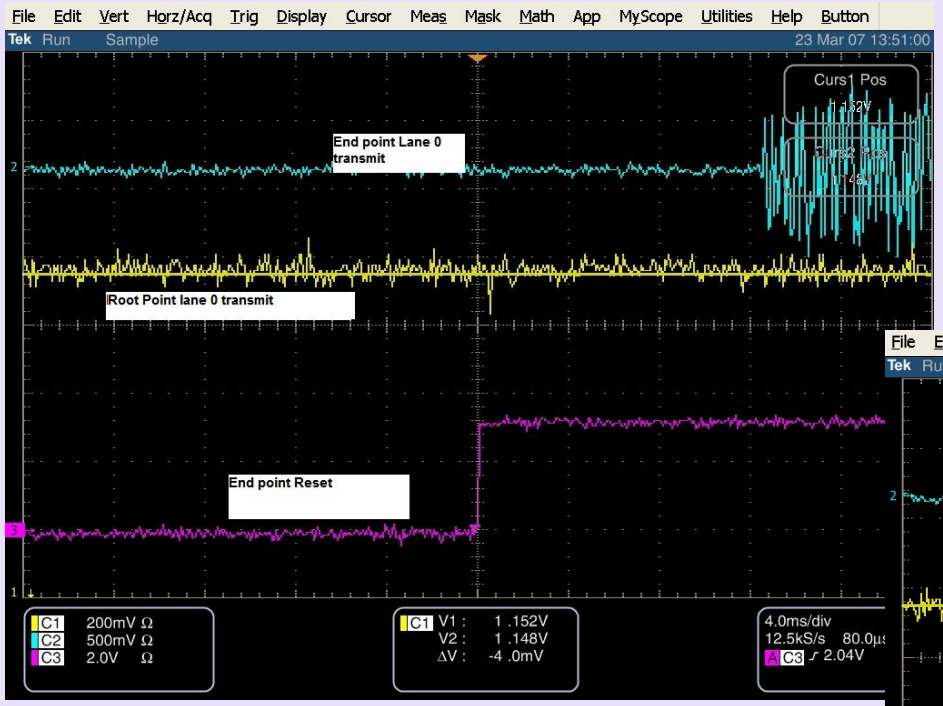
- When it comes to system applications designs are driven by time constraints
  - running HSpice and timing analysis is often bypassed
- Use of tools/equipment to find valid driver strength for specific application is not taken into considerations
  - uses default device settings from chip provider
  - it is highly recommended to find settings specific for end application

- Recommendation:
  - Run noise analysis of the system
  - Run compliance patterns on the system
  - Run signal integrity and timing analysis of the system for the application

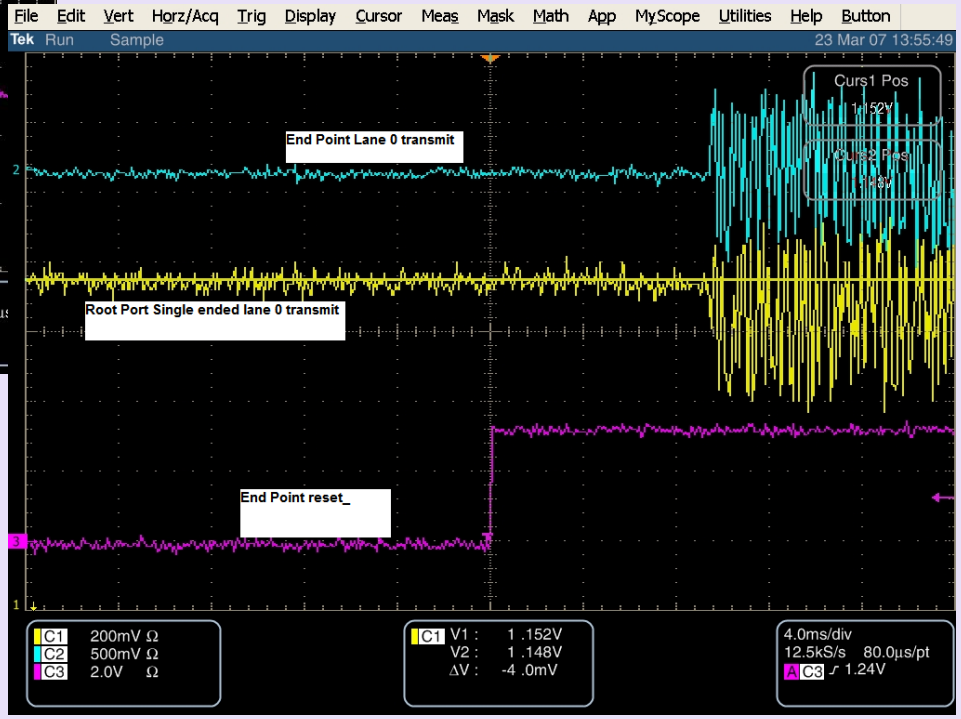
# TX/RX Trace Lengths

- PCI Express specification and design layout guide from PCI-SIG provides recommendation on trace lengths
  - ✓ If traces are too short use expander connector on the board to increase the length of the trace
- Termination of the unused IO's for TX/RX signals on mother board and plug-in card
- Coupling: AC/DC coupling

# TX Coupling Example




**<= Bad board**




**Good board =>**

# BIST Testing

- Running BIST testing
  - Common issues:
    - BIST testing is very limited
    - Some designs with BIST feature enabled do not support PCI Express Compliance pattern, but provide support for K28.5+/-, D21.5 and K28.7 patterns
- BIST test can be used as part of lab validation vehicle

NOTE: special consideration to setup: Spread Spectrum ON/OFF

# Compliance Testing

- PCI-SIG provides compliance testing workshops
- Product tested at the workshop has longer life cycle and better position on the market
- Common mistake:
  - Settings for  external clock versus internal clock while running testing

# HSpice Simulations

- Common issue:
  - ✓ PCI Express is a high speed interface and it is very common to run HSpice simulations using industry offered or in-house simulators
  - ✓ Bypassing HSpice simulations results in
    - respin of PCB
    - longer design cycles due to debug/bring up activities

# Component Placement on The Board

- ✓ Use TX/RX traces for different PCI Express ports on different board layers
- ✓ Use clock oscillators which are PCI Express compliant
- ✓ Use PHY and MAC which are PCI Express compliant
- ✓ Closely follow routing rules developed by PCI-SIG
- ✓ Coupling and capacitance placement can affect design
- ✓ Trace lengths can also have a big affect on the design
- ✓ External cabling and use of connectors recommended by PCI-SIG

# System Verification

- Currently there are some developments of system level verification with use of mixed signal environments

- Early involvement into system level architecture and verification will help to improve testability and reliability of the product

# Hardware Implementation Issues and Solutions

- Considerations while product is under architectural specification
  - PCI Express IP use/application in SOC
  - PCI Express use in FPGA
- Consideration while product is under design/verification stage
  - Chip Design considerations
    - PHY
    - PCI Express stack
    - Physical layout on the chip
    - HSS and LTSM
  - Chip Verification considerations
    - Use of the VIP suite
    - CPU + VIP
    - Error handling on the bus/cpu/pcie
    - Interrupt handling
    - Timeouts
    - Aborts
    - Stack overflow

# Hardware Implementation Issues and Solutions

- Considerations while product (chip) is under bring up in lab on the validation board:
  - Script writing
  - Corner case hardware
  - Software for multithread testing
  - Open source software
  - Firmware support
  - Multi threading testing
  - Compliance testing of the entire chip
  - Running under real OS: Linux, Unix, in-house OS
  - Testing under stress conditions while running real-time applications:
    - heavy traffic
    - voltage
    - temperature

# Software Common Issues and Solutions

- ✓ System mapping
  - Use of the PCIe memory map for the system
- ✓ Programming mistakes
  - Endianness
  - Addressing, consideration for PECFG [35]
  - MSI handling
  - Error recovery, error handling
  - Interrupt handling
  - Unsupported requests handling
  - 16/32/64 bit PCI Express device handling
  - Device present/not present
  - Interrupt handling support ( Legacy interrupts versus Message
  - Signaled interrupts standardization
  - PCI Express use of attribute bits (transaction attributes):
    - Relax ordering
    - Snoop
    - Split Completion Message
    - Split Completion Error
    - Byte Count Modified

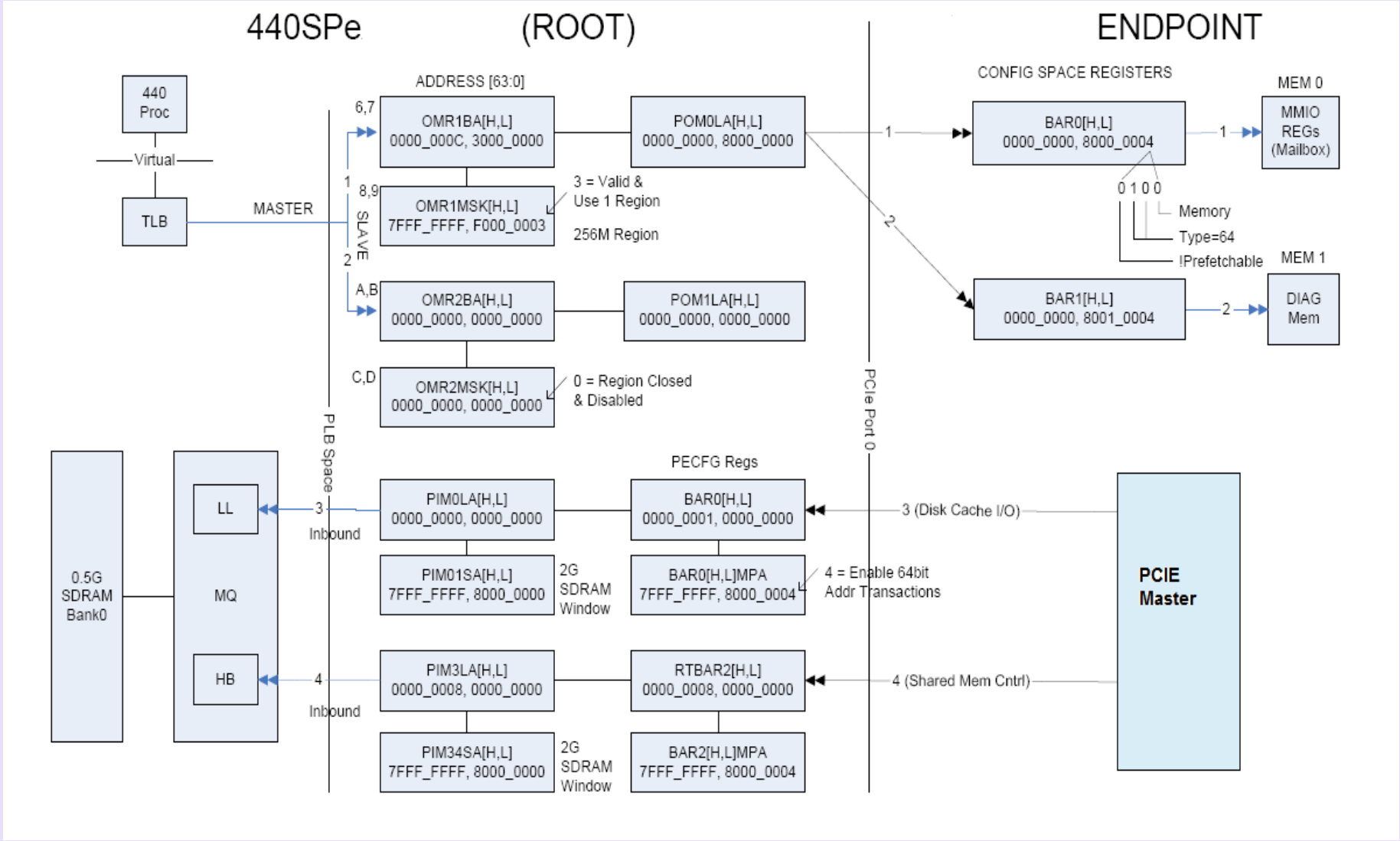
# System Mapping

- Use of the PCIe memory map for the system
  - Understanding of the PCI Address Mapping
  - Opaque vs Non-Opaque bridge

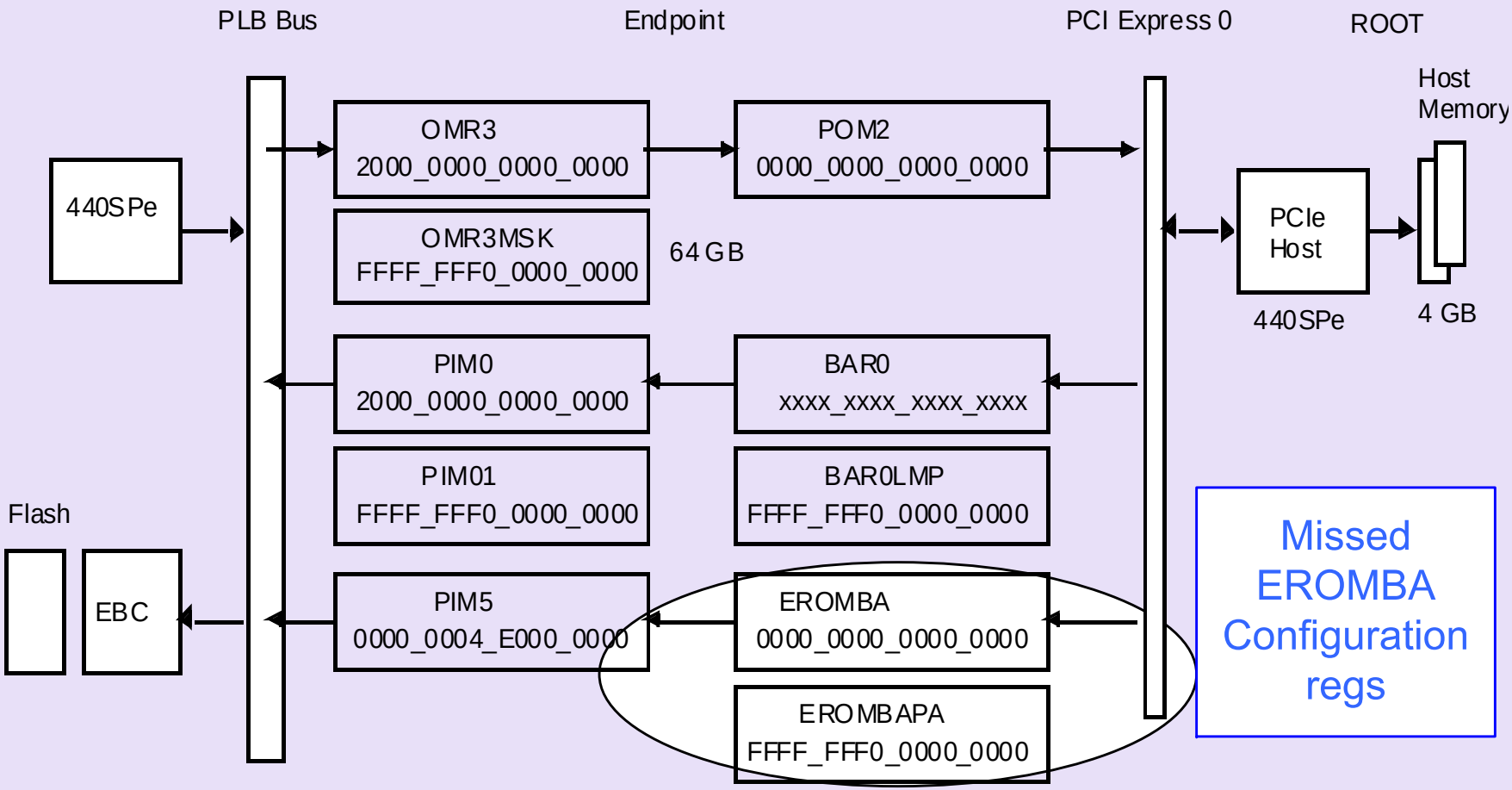
# Debugging Shortcuts

- ✓ Use of PCI Express analyzer/exerciser
- ✓ Advanced Error handling support
- ✓ Reuse of the existing testing
- ✓ Use of mixed signal environments
- ✓ Leveraging previous experiences and creating so-called post-mortem documentation
- ✓ Creating generic suite of tests to validate/verify IP's

# Root Port Configuration Example



# End Point Configuration Example



# Common configuration Questions

- Requests supported: (when max payload size and or max read request size is <= 256Bytes)
  - ✓ What is the difference between payload size and read request size?
- If the max payload size and or max read request size is > 256Bytes, or =512B, how many posted outbound/inbound write requests supported?
- If source PCIe launches a 512B request, and target only supports max payload size 128B, how does SoC process it?

# Performance Optimization

- Consideration for performance improvements
  - Manufacturing, board layout
  - Hardware optimizations
  - Software optimizations

# Performance: Layout

- Use of corner cases hardware on the board, speed sorting
- Short traces allow faster flight time
- Limit number of layers on the board
- Selecting components which have higher margins in terms of timing constraints ( overshoot/undershoot, setup/hold time)
- Selecting specific memory DIMMs, SODIMMS for Dynamic Memory allows to improve performance of the PCI Express accesses in Hardware
- Selecting devices which are PCI Express compliant

# Performance: Hardware

- Improving programmability of the hardware devices
  - ✓ Support for Advanced Capabilities
  - ✓ Support for Advanced Testing
- Memory
  - ✓ Prefetching capabilities support
  - ✓ Relaxed Ordering support
  - ✓ Non-volatile memory
- Common Bus protocol support for all PCI Express features

# Performance: Software

- Code optimization and consistency
  - ✓ Open source community input to source code for PCI Express drivers needs to have common driver architecture
  - ✓ PCI Express DMA drivers available across many platforms
  - ✓ Enabling optimization for PCI Express throughput via prefetching, target direct completion, optimization, relaxed ordering, ack on first request or bucket modes
  - ✓ Dynamic Memory (SDRAM, SDRAM DDR1/2/3) optimizations for PCI Express memory accesses
  - ✓ Optimization for peripherals

# Summary

- Real-time implementation and use of PCI Express is very challenging and requires knowledge and expertise
- Use of PCI-SIG compliance testing and attending PCI-SIG workshops enables shorten design time and system implementations
- Leveraging previous experience and existing knowledge helps to improve time to market timelines

# Questions?

Thank you for attending the PCI-SIG Developers Conference Europe 2009

For more information please go to www.pcisig.com