Integration and System Verification of PCI Express® IP

Ilya Granovsky, Elchanan Perlin

PCI Express IP Development

IBM Systems & Technology Group

Agenda

- Introduction
- Selecting PCIe® IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

Introduction

- Why so complex?
  - ✓ PCIe standard covers a wide range of applications

  thus

  - ✓ Many parameters
  - ✓ Many optional features
  - ✓ Many size-performance-cost trade-offs

- Make an intelligent decision and then validate your setup at the system level

Agenda

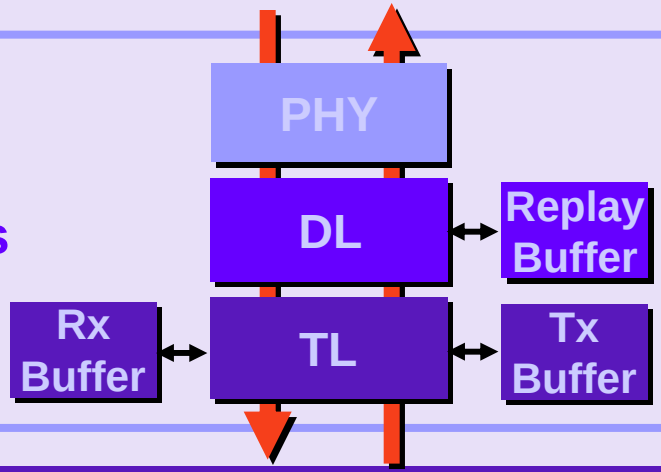
- Introduction
- Selecting PCIe IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

Selecting PCI Express® IP Parameters

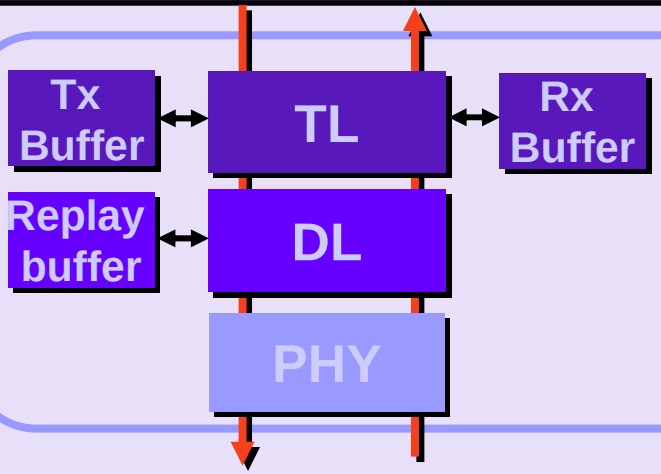
- Port Type Selection
  - ✓ Root Port, Endpoint, Dual Mode (RP/EP)
- Link Speed/Width Selection
- Buffering
  - ✓ Replay buffer, Rx/Tx Buffers, Completion Data Buffers
- Request/Payload Sizes
- Optional features
  - ✓ Multi-VC, ECRC, AER
- Configuration Space

Selecting PCI Express IP Parameters – cont.

- x16 upstream link
- EP mode
- 512B MPS, 512B Reads
- 1VC, AER, ECRC



Application Interconnect



- x4 downstream link
- Root mode
- 128B MPS, 512B Reads
- 1VC, AER, ECRC

PCIe IP Parameters Link Speed/Width

Theoretical Bandwidth

(per link, per direction, in GByte/sec)

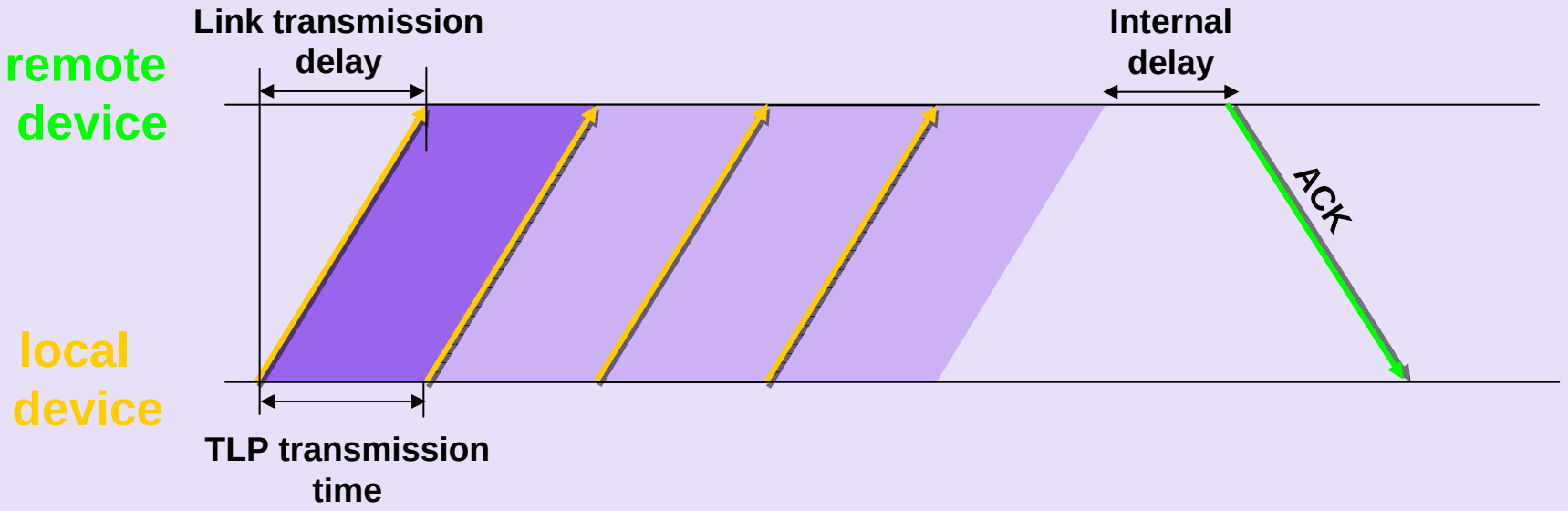
Link Width	5GT/s Speed	2.5GT/s Speed
x16	8	4
x8	4	2
x4	2	1
x1	0.5	0.25

Degradation Factors

- Packet overhead
  - ✓ TLP header
  - ✓ ECRC
  - ✓ Link header and framing
- Link Management
  - ✓ FC Updates
  - ✓ ACK/NAK DLLPs
- System Efficiency
- Congestion

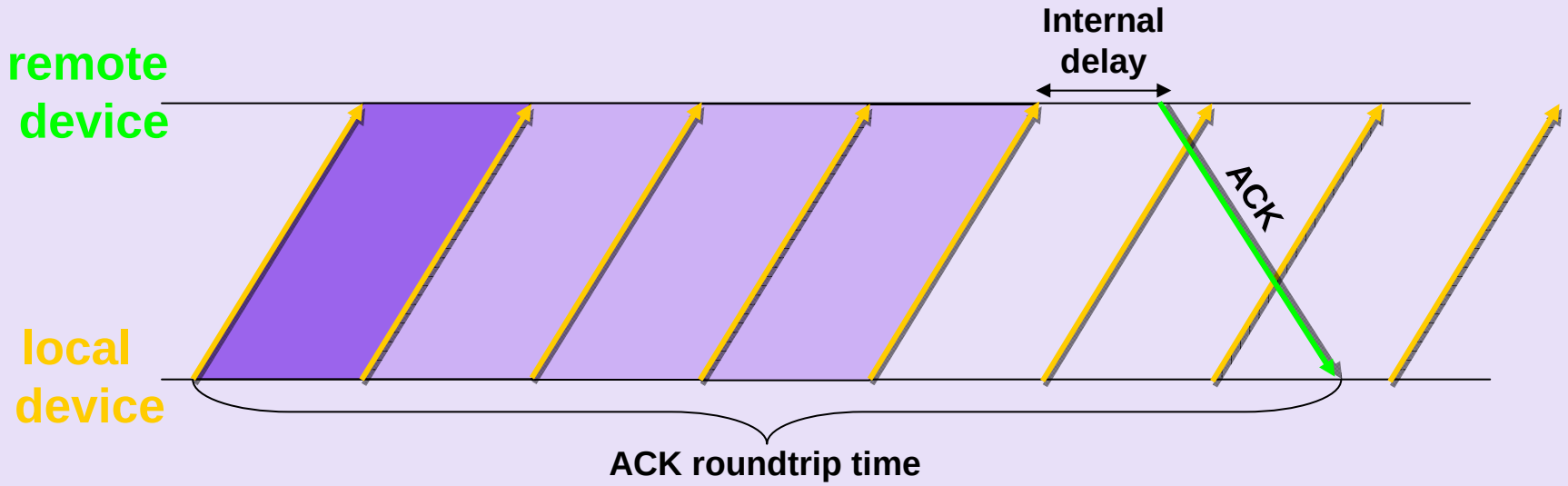
PCIe IP Parameters
Replay Buffer Selection

- Impacts transmit bandwidth
- Stores outbound TLPs until acknowledged by receiver to allow retransmission in case of LCRC error
- Usually ACK DLLPs are combined for number of TLPs ("ACK factor")



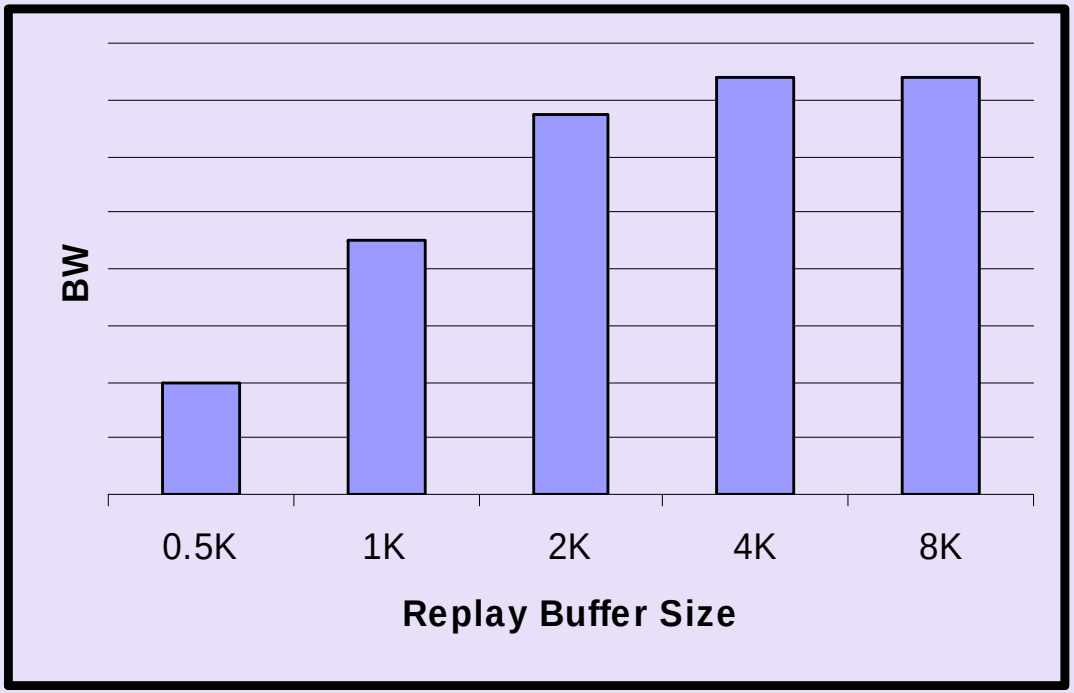
PCIe IP Parameters Replay Buffer – cont.

- Replay buffer should be large enough to compensate for ACK roundtrip time
  - ✓ (TLP transmission time) X ACK factor
  - ✓ TLP Transmission delay
  - ✓ Internal processing delay in receiver
  - ✓ ACK transmission delay



PCIe IP Parameters Replay Buffer – cont.

- Performance simulation results
  - ✓ x8 2.5GT/s link
  - ✓ TLPs with 512 Byte MPS (Max Payload Size)

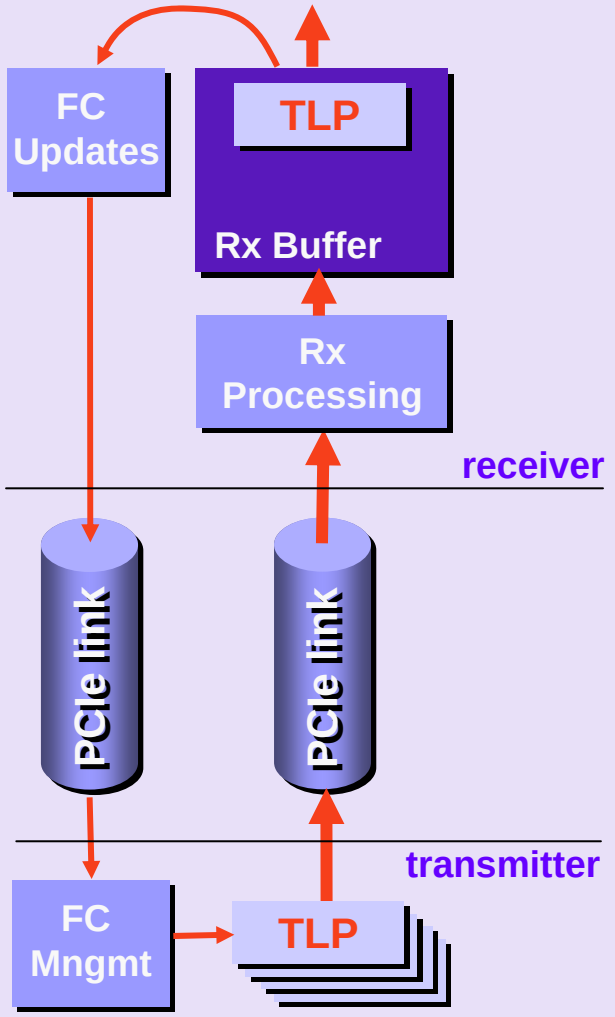


PCIe IP Parameters Receive Buffers

- Receive buffers cover for internal delays and FC Update round trip
  - ✓ Remote transmitter needs to have credits always available to support constant data flow
- Consider different traffic patterns
  - ✓ Mostly large packets – need several header entries and large data buffer.
    - – Consider data buffer of

      MPS X *[header queue depth]*
  - ✓ Small or mixed large/small packets – need large header buffer, reduced data buffer
    - – Consider data buffer of *MPS* X *N*

      (*N* << *[header queue depth]*)



PCIe IP Parameters Receive Buffers – cont.

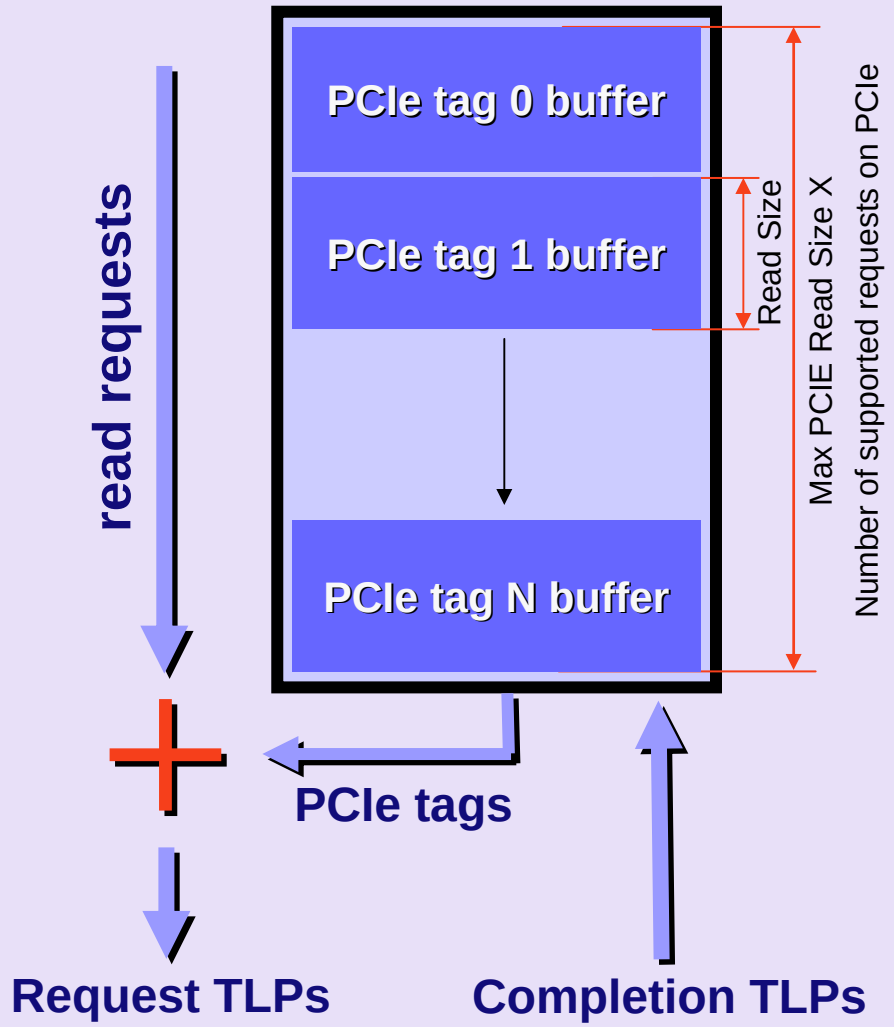
- Receive buffer evaluation example:
  - ✓ 5GT/s x8 link
  - ✓ MPS – 256B => TLP transmission delay of 72ns (inc. TLP header and digest)
  - ✓ TLP Tx+Rx+Link delay – 200ns
  - ✓ Rx buffer S&F delay – 64ns
  - ✓ FC Update latency – 20 ns
  - ✓ FC Update Tx+Rx+Link delay – 160ns
  - ⇒**FC Update roundtrip of 516ns**
  - ⇒**Need Rx buffer for 8 TLPs (8 headers, 2K data) for max bandwidth**

PCIe IP Parameters Transmit Buffers

- Provide temporary storage for TLPs before transmitted on the link
- Tightly coupled with application's interconnect architecture
  - ✓ May be a very thin buffering layer used for speed matching only.
  - ✓ Consider larger buffers to hold TLPs when the link becomes congested
- May implement cut-through transmission
  - ✓ Need to make sure that once TLP transmission started, it will not be paused.
  - ✓ May need to nullify TLP and restart its transmission if link bandwidth cannot be sustained
    - – OR use store & forward buffer management

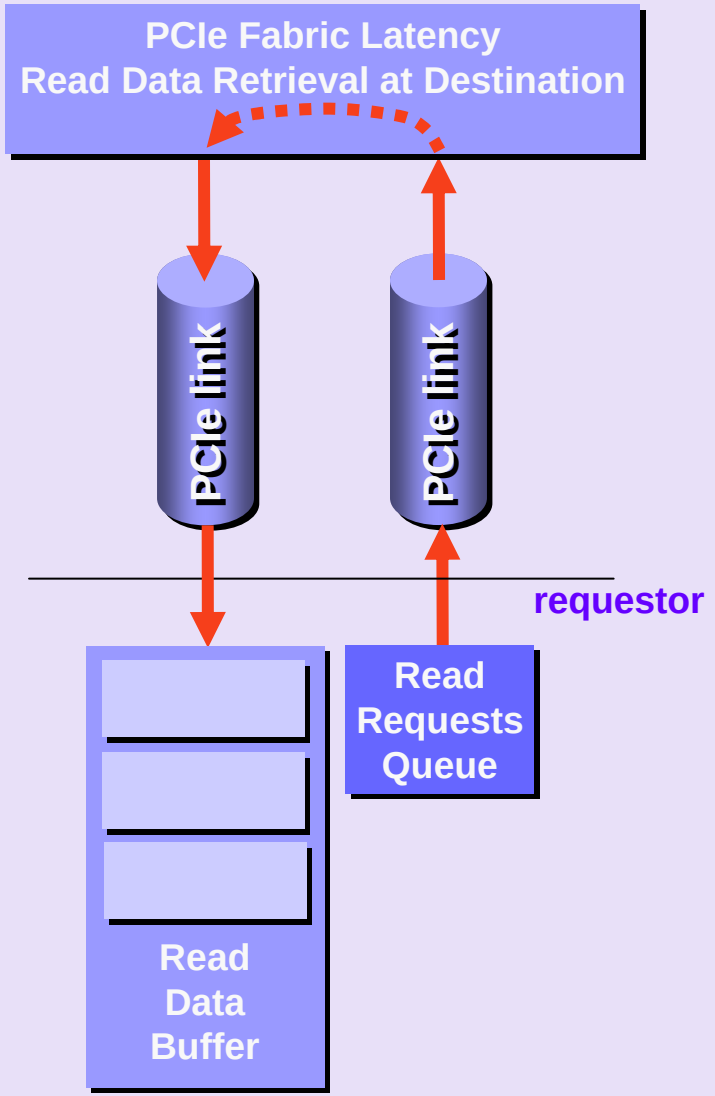
PCIe IP Parameters
Read Data Buffers

- Read data buffer is pre-allocated by requestor when forwarding read request to PCIe
- Read data buffer should compensate for read roundtrip latency
- Simple implementation – associate PCI tags with fixed data buffer segments.



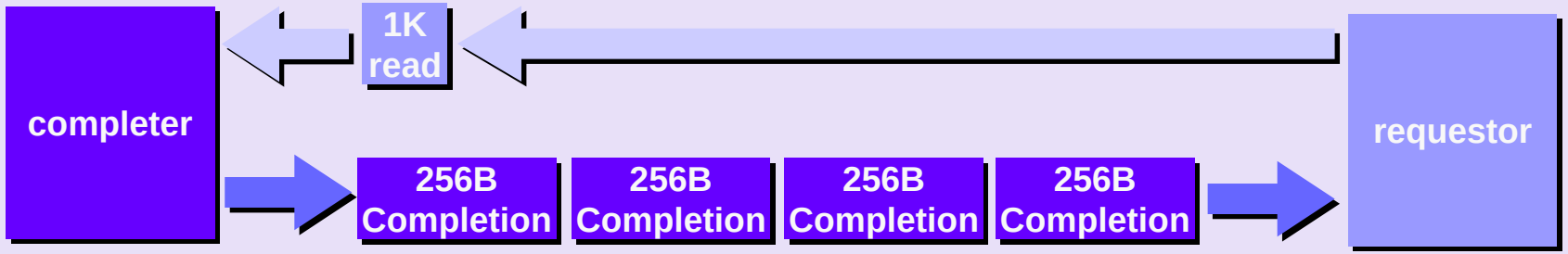
PCIe IP Parameters
Read Data Buffers - cont

- **Example**: target Reads Bandwidth – 3.2 Gbyte/sec (5GT/s x8 link)
- Read data return latency (round trip) – 1000 ns
- Need outstanding reads of 3200 bytes total to sustain max bandwidth
  - ✓ 2 reads of 2K
  - ✓ 4 reads of 1K
  - ✓ **8 reads of 512 bytes**
  - ✓ 16 reads of 256 bytes



PCIe IP Parameters
Read Request Size

- PCIe allows large read requests completed with multiple smaller completion TLPs
- Potentially reduces request traffic
  - ✓ Single request TLP for multiple completion TLPs
- May require large completion (read) data buffers in requestor
  - ✓ Some applications prefer small read request sizes for better flexibility in read buffer management



PCIe IP Parameters Payload Size

Large Payloads (1KB - 4 KB) - pros

- Better Link Utilization
  - ✓ Lower TL overhead (4 header DWords for 1024 data DWords), less frequent FC updates
- Less Header Credits
  - ✓ Smaller header buffers
  - ✓ Lower headers processing overhead

Large Payloads (1KB - 4 KB) - cons

- Requires Large Buffers
  - ✓ TL data buffers, Replay buffers
- Needs PCIe native software
- Has to be supported by all the devices in the system

PCIe IP Parameters Payload Size – cont.

Small Payloads (128 bytes - 256 bytes) - pros

- Require smaller buffers
  - ✓ Simplifies data buffer management if implemented in 128 byte units (no data credits management needed)
- Supported by all devices
- Compatible with existing PCI software

Small Payloads (128 bytes - 256 bytes) - cons

- Lower link utilization due to fixed per-packet overhead
  - ✓ TLP Framing (4 header DWords for 32 data DWords)
  - ✓ TLP digest (One DWord for 32 data DWords
  - ✓ TLP framing

PCIe IP Parameters Optional Features

- Advanced Error Reporting
  - ✓ Configuration Space Capability that provides additional error reporting resolution, header logging, etc.
  - ✓ Widely implemented, requires software support
- ECRC Support
  - ✓ Provides end-to-end protection for TLPs
    - Covers for corruption in switches/bridges
  - ✓ Adds 1DW overhead to the packet
  - ✓ Requires generation and checking logic which grows with link width
  - ✓ Very limited support, requires AER

PCIe IP Parameters Optional Features - VCs

- Multiple VCs allow traffic differentiation
- Require separate logical buffers
  - ✓ Header buffers usually require separate physical arrays to allow low-latency arbitration, data buffers may share same physical memory.
- Consider performance requirements for specific VCs and allocate adequate buffering resources
  - ✓ Larger buffers required to support max peak bandwidth for different VCs
  - ✓ Some VCs can be defined as low-performance and consume minimal buffering

Agenda

- Introduction
- Selecting PCIe IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

PCIe IP Integration System Requirements

- Understand possible system configurations
- Specify address regions types and sizes, address translation modes
- Consider software compatibility issues
- Check for PCIe features supported by other devices in the system
  - ✓ Read sizes, payload sizes, VCs, ECRC
- Transactions ordering
  - ✓ Check potential deadlock scenarios
- Interrupts handling

PCIe IP Integration Ordering

- Validate that “NO” rules are followed
  - ✓ Posted requests are ordered if RO attribute is cleared
  - ✓ NP requests push posted requests
  - ✓ Read completions with RO attribute cleared push posted requests
- Validate “YES” rules
  - ✓ Posted requests can bypass NP requests
  - ✓ Posted requests can bypass read completions
- Validate “Y/N” rules
  - ✓ Completions to non-posted writes may bypass posted requests
  - ✓ Reordering of NP requests

Row Pass Column?		Posted Request	Non-Posted Request		Completion	
		Memory Write or Message Request (Col 2)	Read Request (Col 3)	I/O or Configuration Write Request (Col 4)	Read Completion (Col 5)	I/O or Configuration Write Completion (Col 6)
Posted Request	Memory Write or Message Request (Row A)	a) No b) Y/N	Yes	Yes	a) Y/N b) Yes	a) Y/N b) Yes
Non-Posted Request	Read Request (Row B)	No	Y/N	Y/N	Y/N	Y/N
	I/O or Configuration Write Request (Row C)	No	Y/N	Y/N	Y/N	Y/N
Completion	Read Completion (Row D)	a) No b)Y/N	Yes	Yes	a) Y/N b) No	Y/N
	I/O or Configuration Write Completion (Row E)	Y/N	Yes	Yes	Y/N	Y/N

PCIe IP Integration Interrupts Support - RC

- Receive and terminate Assert/Deassert INTx messages
- Receive MSIs (appear as memory writes)
- Receive and terminate PCIe error messages
  - ✓ Reported to local configuration space
  - ✓ May be reported to the application as upstream MSIs or as application-specific core interrupt
- Local PCIe errors
  - ✓ Reported to local configuration space
  - ✓ May be routed to the application as upstream MSIs or as application-specific core interrupt

PCIe IP Integration Interrupts Support - EP

- Issue MSI writes based on device parameters and configuration space MSI/MSI-X capability controls
- Generate upstream INTx messages, based on configuration space controls
  - ✓ Consider ordering: read from status register reflected by the interrupt line must push the Assert_INTx message
- Generate upstream PCIe error messages upon PCIe errors detection

Agenda

- Introduction
- Selecting PCIe IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

PCIe System Verification Overview

- PCI Express IP may be already verified

but

- System level validation of PCIe port is required.
- Focus on:
  - ✓IP Integration
  - ✓Correct Parameters Selection
  - ✓System Level Sequences
  - ✓Performance

PCIe System Verification IP Integration

- Validate that all IP interfaces are connected correctly
  - ✓External memories
  - ✓Configuration signals attached to registers or assigned to correct fixed values
  - ✓System logic
    - e.g. Interrupt handler, Hot-plug controller, etc.
- How to verify IP integration ?
  - ✓Implementation review
  - ✓Collect coverage on IP interfaces

PCIe System Verification Parameters Selection

- What may be the impact of wrong parameters selection ?
  - ✓Dead silicon
  - ✓Data integrity
  - ✓Non-functional features
  - ✓Deadlocks
  - ✓Performance degradation

PCIe System Verification Parameters Selection

- What can we do ?
  - ✓Implementation review
  - ✓Cover queue empty and queue full conditions
  - ✓Performance tests
  - ✓Stress tests of large and small packets
  - ✓Check for pending transactions at the end of the test

PCIe System Verification Performance

- Missing critical performance targets may have significant system impact
- Needs to be checked early in the verification phase
- Integrator should identify and specify performance targets for each path
  - ✓ First phase - verification covers each path separately
  - ✓ Second phase - combination of different dependent and independent paths is tested
- Keep achieved performance numbers for future testing and lab testing reference

PCIe System Verification System Level Sequences

- PCIe Port Initialization
- Error Reporting
- Reset sequences
  - ✓ Fundamental Reset, Function Level Reset, Hot Reset
- Power Management
- Interrupts
- Ordering
- and … random tests

Does not mean that we need to cover everything in system testing!

PCIe System Verification How can we do it Efficiently ?

- Use a combination of tools
  - ✓ Reviews
  - ✓ Simulation
  - ✓ IP Interfaces toggle coverage
  - ✓ Assertions for checking and coverage
  - ✓ Vendor-provided PCIe verification IP
  - ✓ Formal verification of system logic units
  - ✓ Lint tools
- Maximize reuse
  - ✓ Assertions from the IP
  - ✓ Checkers
  - ✓ Test suites
  - ✓ System coverage attributes

PCIe System Verification Coverage Collection

- Focus on system level features
  - ✓ No need to re-test IP level features
- Trust the IP level coverage – it is much more extensive
- Examples:
  - ✓ No need to check every bit in the configuration space
  - ✓ No need to check all timeout options
  - ✓ No need to have white box coverage

PCIe System Verification Interrupt Testing Example

- Example test scenario: INTA interrupt testing in endpoint.
  - ✓ Create constant outbound packets flow
  - ✓ Clear Interrupt Disable bit (Command Register, bit 10), write 8`h01 to the Interrupt Pin register
  - ✓ Create a host model which sets internal INTA indication when receives Assert_INTA message and clears it when receives Deassert_INTA message
  - ✓ Program the host model to clear adapter's interrupt status with certain delay after detecting INTA virtual wire assertion.
  - ✓ Periodically set adapter's interrupt status, triggering Assert_INTA message to the host.
  - ✓ Make sure that at the end of the test the INTA virtual wire in the host is low.

Agenda

- Introduction
- Selecting PCIe IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

Debug Features

- Provide critical internal information to software
  - ✓ Rx/Tx/Replay buffers status
  - ✓ Transactions status
  - ✓ Available Credits
- Provide manual control over major port functions
  - ✓ Transmit disable/flush
  - ✓ Receive forwarding disable
  - ✓ Kill outstanding transactions, release resources
  - ✓ Disable/bypass/modify addressing checks
  - ✓ Ignore or disable detection of certain errors

Debug Features – cont.

- Provide manual control over link initialization
- Backdoor memory mapped configuration space access in endpoint mode.
- Control over configuration space parameters
- Trace receive and/or transmit interfaces
  - ✓ Use certain packet types as triggers for the trace
- Add performance measurements

Agenda

- Introduction
- Selecting PCIe IP Parameters
- PCIe IP System Integration Aspects
- System Verification
- Debug Features
- Summary

Summary - Integration

- Start with specifying basic IP parameters and providing performance targets
  - ✓ Port type, link width, number of VCs
- Identify system requirements
  - ✓ Required operational modes, transaction sizes, etc.
- Specify receive, transmit and replay buffering requirements based on estimated system latencies
- Review all the IP parameters early in integration phase
  - ✓ Consider providing control over critical parameters for increased flexibility
- Evaluate all combinations of the data flows
  - ✓ Check for potential dependencies and deadlock conditions

Summary - Verification

- Need to determine what is the minimal verification needed to integrate a fully verified IP
- Focus on system scenarios
- Performance needs special attention
- Implementation review in early integration phase increases system quality
- Need to understand software requirements

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

For more information please go to
www.pcisig.com

Integration and System Verification of PCI Express IP

Ilya Granovsky, Elchanan Perlin

PCI Express IP Development

IBM Systems & Technology Group