Advanced PCIe® Features Implementation Considerations

Paul J. Mattos (with Ilya Granovsky)

IBM Systems and Technology Group

Agenda

- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

First, a little history…

- **2004** – Implementing PCIe® on a System-On-Chip
  - ✓ Highlight critical trade-offs
  - ✓ Sort through optional features and implementation-specific choices
- **2006** – Optimizing PCIe Port Performance
  - ✓ Support system requirements and achieve optimal resource utilization for target performance
- **2007** – Integration and System Verification of PCI Express®
  - ✓ Intelligently select PCIe parameters and then utilize verification resources effectively
- **2008** – Advanced PCIe Features Implementation Considerations

Why am I doing this pitch?

- As PCIe IP developers at IBM, we have multiple layers of customers
- Internal customers develop chips ending up in IBM products – pretty straight forward
- OEM ASIC or Foundry customers
  - ✓ Many variations, but one that is popular is where internal teams assist external customers with chip architecture and design
  - ✓ Gives some additional insight into areas where some of the complexity of PCIe may (at least initially) be underestimated

Introduction

- PCIe tends to be the main I/O for many customers
- Some choices on PCIe configuration specifics can negatively impact performance
- Decisions become more challenging when advanced features and especially newer or newly specified features are required

Tell 'em what you're gonna tell 'em!

- Development projects, like a strategic plan, shouldn't be cast in stone
- They both need to adapt through continually changing events and circumstances
- PCIe continues to evolve
- This presentation will identify some key items to help keep pace when advanced and newer PCIe features are a must have

What do we mean by ‘basic’ vs ‘advanced?’

- Basic PCIe Implementation
  - ✓ x1/x2/x4, single function, single Virtual Channel (VC), endpoint
  - ✓ 2.5GT/s data rate
  - ✓ 128B or 256B Maximum Payload Size (MPS)
  - ✓ Interrupt support: INTx or MSI
  - ✓ Minimal Power Management (PM) and optional feature support

- Advanced Feature Support: Basic +
  - ✓ x8, x16, Multiple VCs and multiple functions
  - ✓ 5.0GT/s data rate
  - ✓ ≥ 512B MPS
  - ✓ Advanced Error Reporting (AER) with optional Malformed TLP checks and ECRC
  - ✓ TLP Poisoning support
  - ✓ Single-Root I/O Virtualization (SR-IOV) & Multi-Root I/O Virtualization (MR-IOV)
  - ✓ Aggressive PM and rich optional feature support
  - ✓ New protocol extension ECNs
  - ✓ Critical IP features (Non-PCIe)

Agenda

- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

Customer Observations

- Generally knowledgeable about PCIe or become knowledgeable via:
  - ✓ Industry training – materials tend to lag specification definition, however
  - ✓ Hitting the books – industry and IP specifications
    - Literally thousands of pages and associated learning curve
  - ✓ Interaction with IP development team(s)
- IP development team needs to understand the customer application in order to suggest an appropriate configuration(s)

Customer Observations

- What we find is that customers aren't always aware of the details and implications of certain features
  - ✓ Takes a lot of effort to get up to speed on PCIe
  - ✓ Seek to engage with customers as early as possible
  - ✓ Watch for an 'inflated' PCIe feature list
    - Could be legitimate, but could also be an indication of someone who may be relatively new to PCIe
    - Not a problem: getting to the next level of detail often results in paring down the list of required features
  - ✓ There are many cases, though, where advanced features (a lot of 'em) are required
    - The evolving nature of PCIe makes this particularly interesting for developers at many levels

Agenda

- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

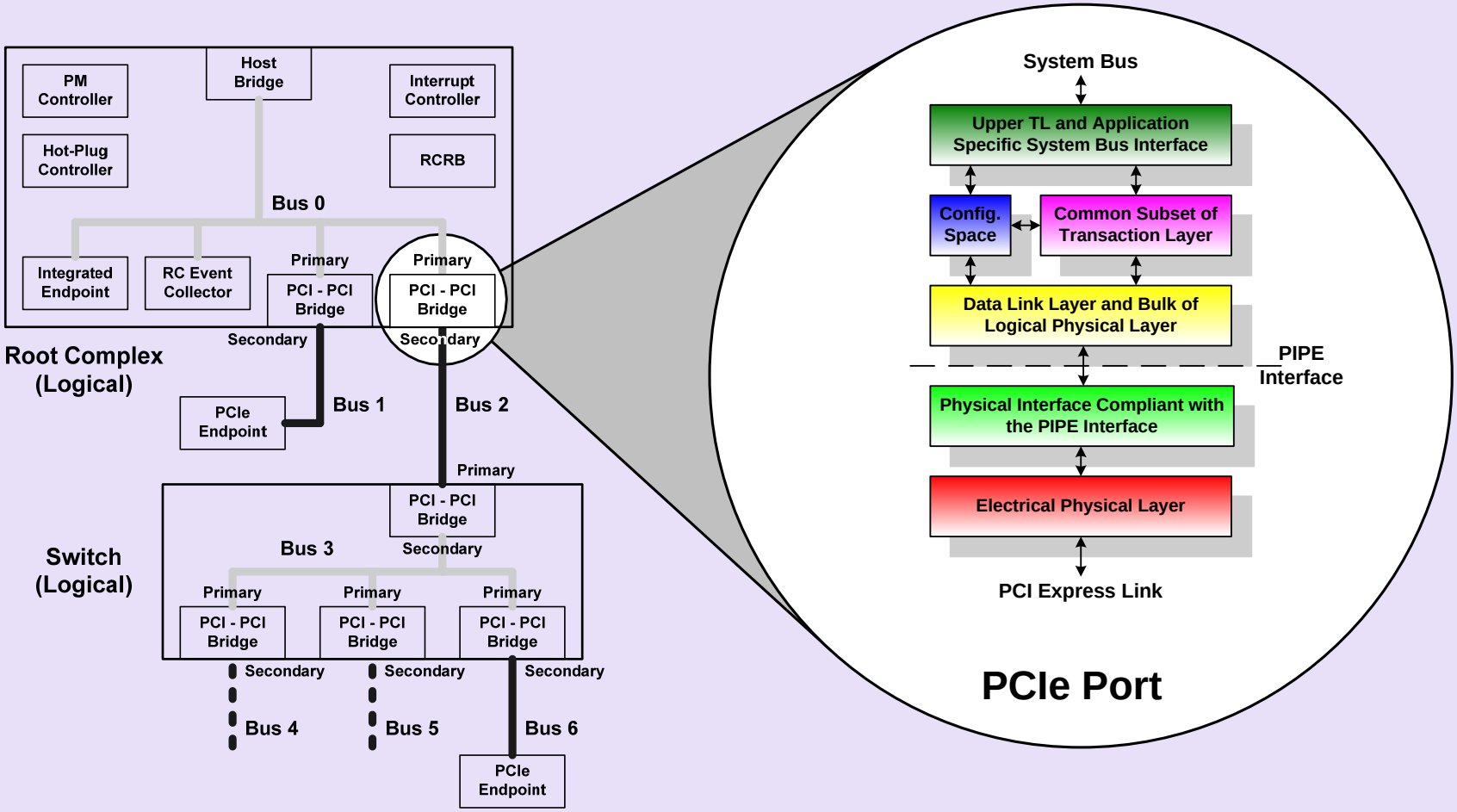
PCIe covers a wide range of applications

- This is where due diligence comes in…
- Many parameters
- Many optional features
- Many size/performance/cost trade-offs

Key Parameters

- A ‘short’ list with many considerations required at the next level of detail…
- Device Type
- Data Rate & Link Width
- Request and Payload Sizes
- Virtual Channels
- AER and ECRC
- Buffering and Interaction between parameters
- Multiple Physical Functions and Virtual Functions (SR-IOV)

PCIe Port/Device Relationship



PCIe IP Parameters Link Speed/Width

Theoretical Bandwidth

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

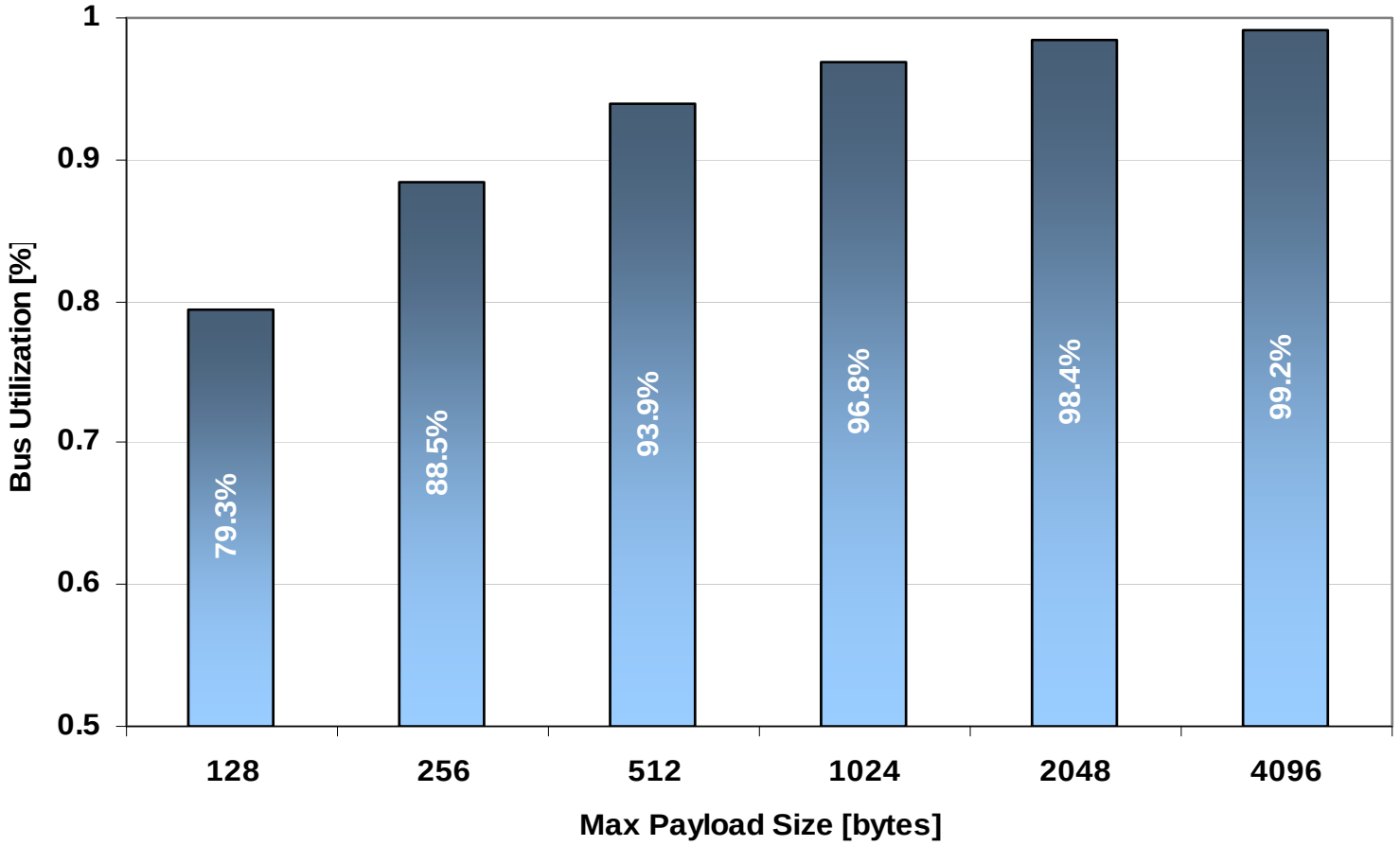
Link Width	PCIe 2.0 Speed	PCIe 1.x Speed
x16	8	4
x8	4	2
x4	2	1
x1	0.5	0.25

Degradation Factors

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

Baseline PCIe Performance

Typical link efficiencies for different MPS settings



PCIe IP Parameters Payload Size

Large Payloads (1KB - 4 KB) - **Pros**

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

Large Payloads (1KB - 4 KB) - **Cons**

- Requires Larger Buffers
  - ✓ TLP 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 legacy 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 - 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

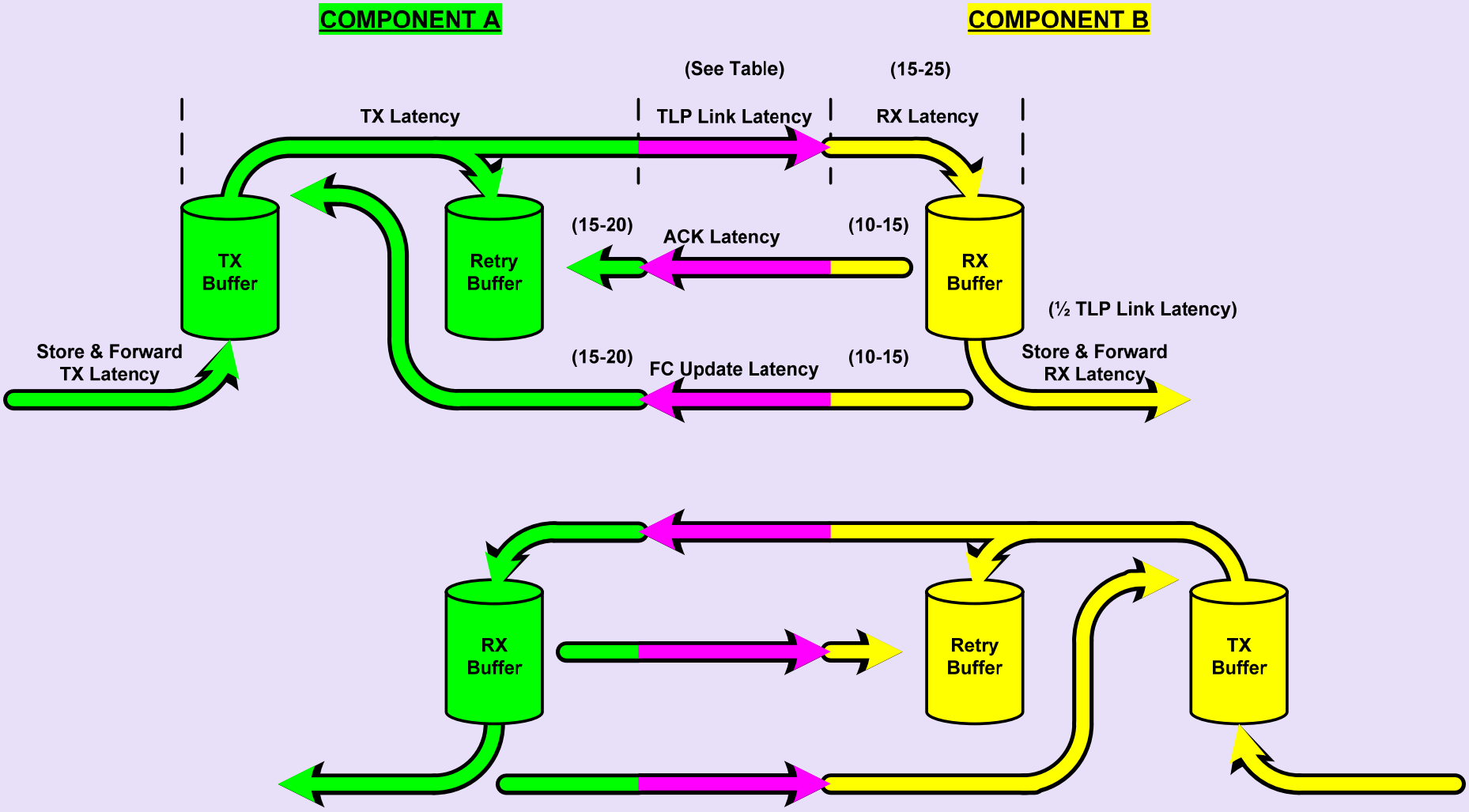
PCIe IP Parameters Optional Features - AER

- Advanced Error Reporting
  - ✓ Configuration Space Capability - provides additional error reporting resolution, header logging, etc.
  - ✓ Widely implemented, requires software support
- ECRC Support
  - ✓ Provides end-to-end protection for TLPs
    - – Covers for potential corruption in switches/bridges
  - ✓ Adds 1DW overhead to the packet
  - ✓ Generation/checking logic - grows with link width
  - ✓ Limited, but growing industry support
  - ✓ Requires AER

Parameter Interaction Example

- The following slides show possible interactions between device latencies, payload sizes, link widths, and implications for **minimum buffer sizes**

Latency Contributors (Symbol Times)



PCIe TLP Link and S&F Latencies

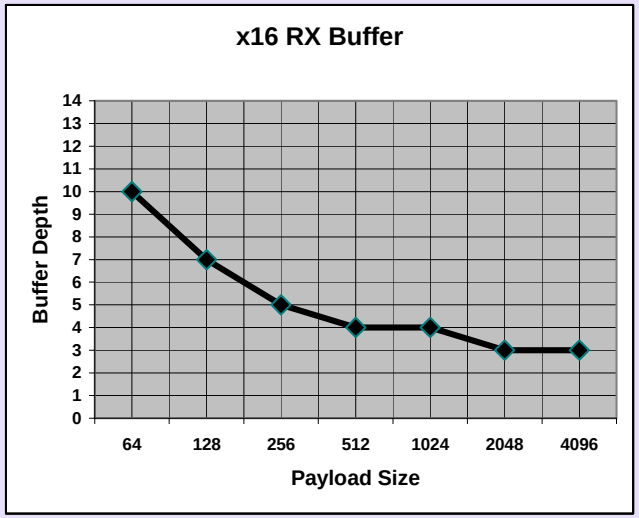
Symbol times to transmit TLPs or empty TLPs from RX Buffer at various link widths & payloads (also symbol times to xmit DLLPs)

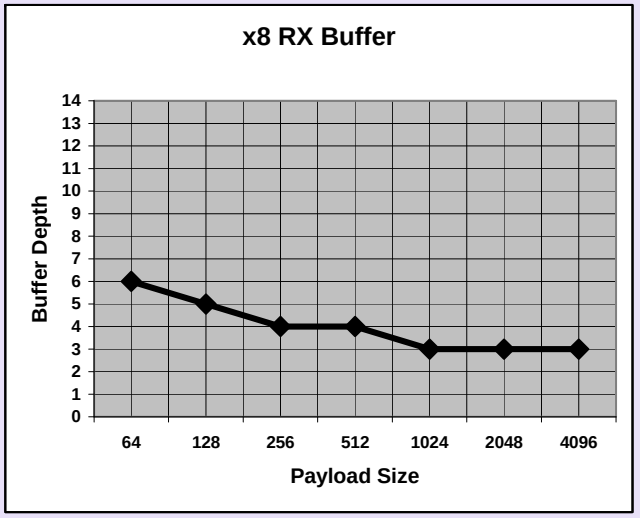
Payload	**x1** (DLLP = 8)		**x4** (DLLP = 2)		**x8** (DLLP = 1)		**x16** (DLLP = 1)	
	Link	**S&F**	**Link**	**S&F**	**Link**	**S&F**	**Link**	**S&F**
64B	92	46	23	12	12	6	6	3
128B	156	78	39	20	20	10	10	5
256B	284	142	71	36	36	18	18	9
512B	540	270	135	68	68	34	34	17
1,024B	1,052	526	263	132	132	66	66	33
2,048B	2,076	1,038	519	260	260	130	130	65
4,096B	4,124	2,062	1,031	516	516	258	258	129

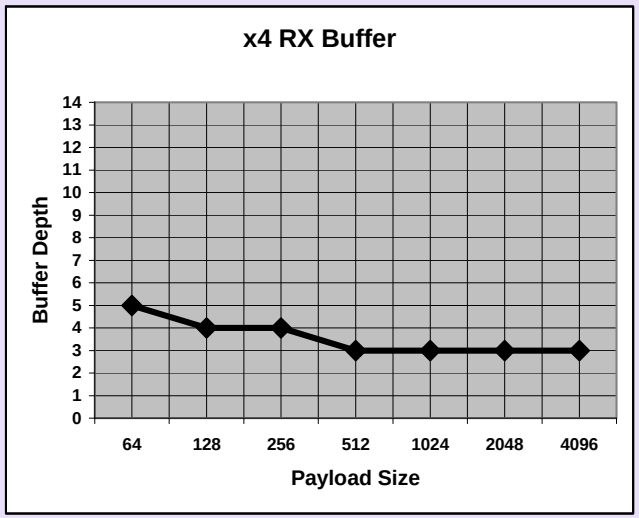
Notes:

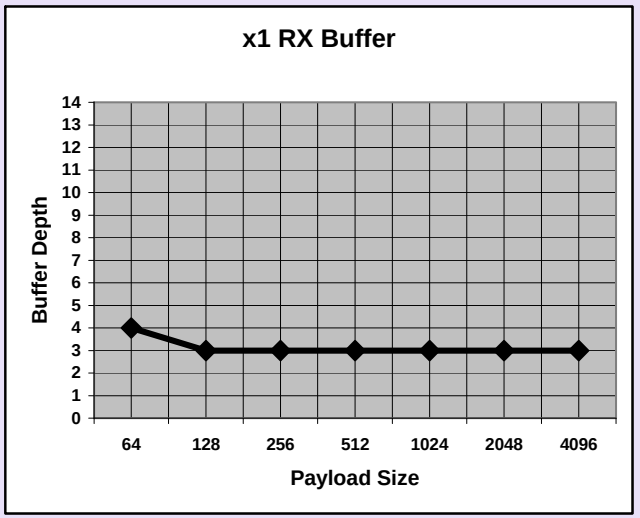
- Symbol time @ 2.5GT/s = 4ns, @ 5.0GT/s = 2ns, @ 8.0GT/s = 1ns
- TLP Overhead = 4DW Header + 2DW LCRC & Framing + 1DW ECRC

RX Buffer – Lower Latencies

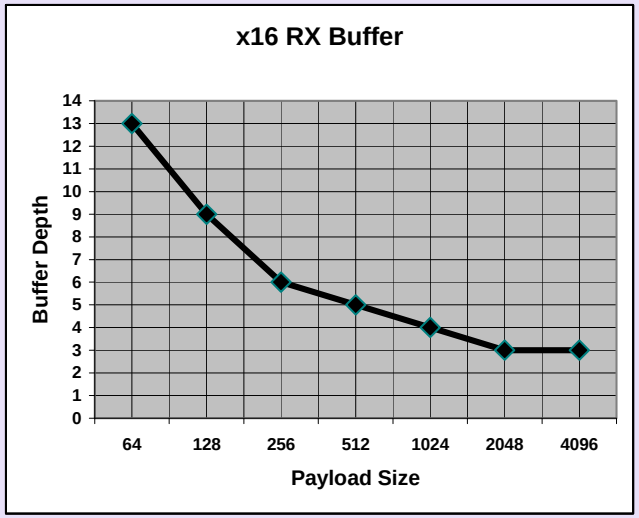


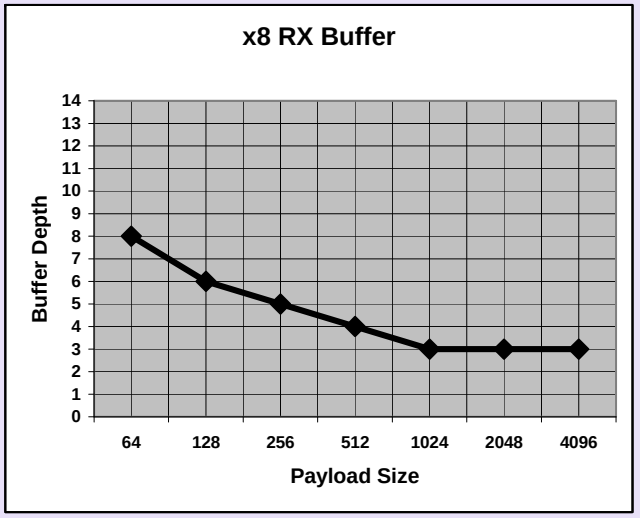


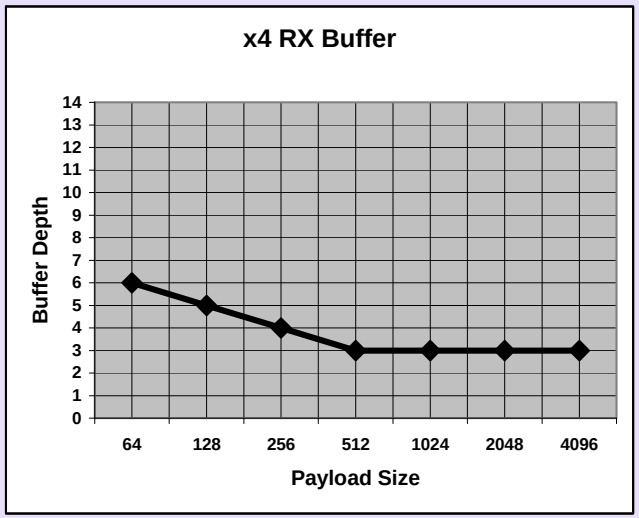


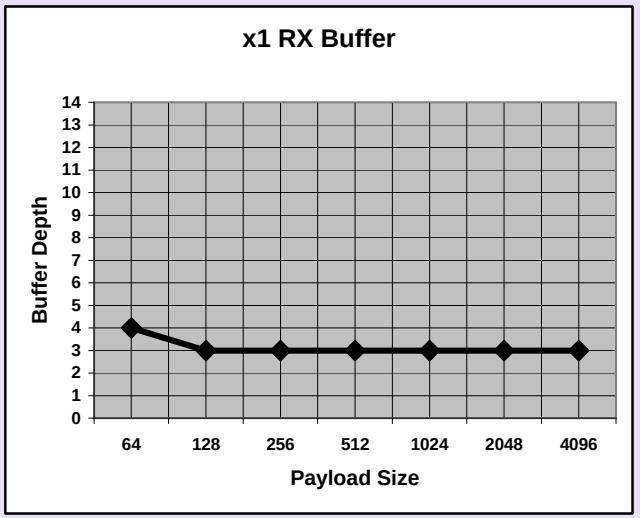


RX Buffer – Higher Latencies

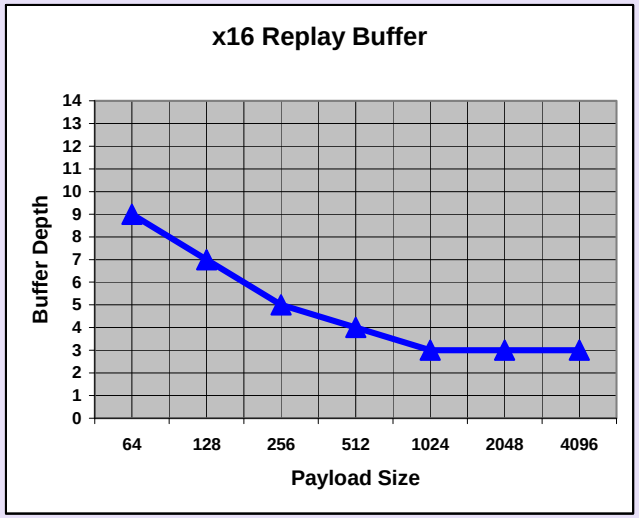


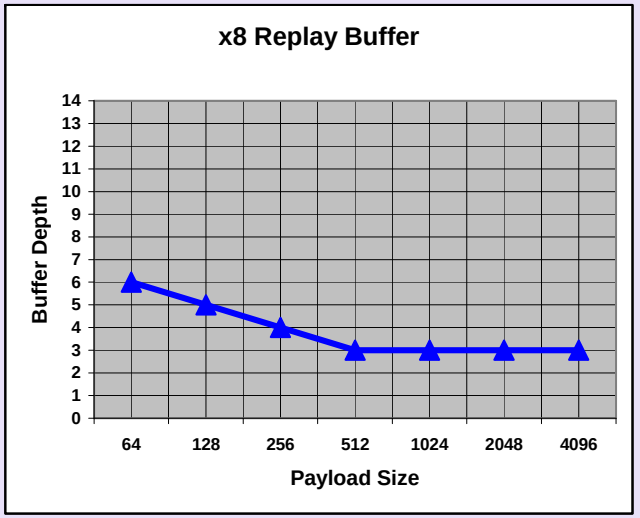


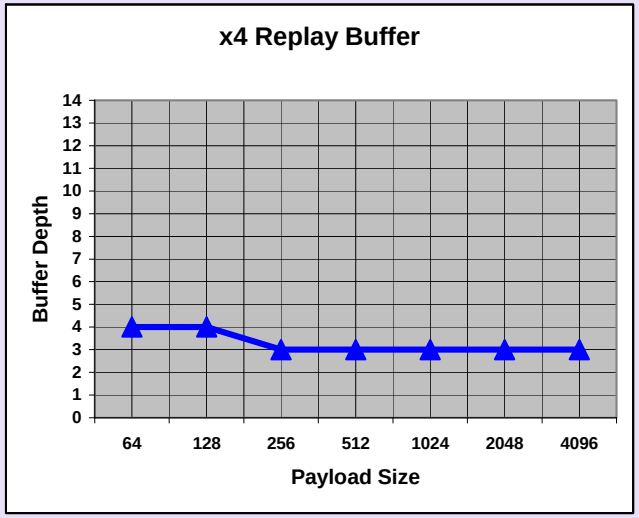


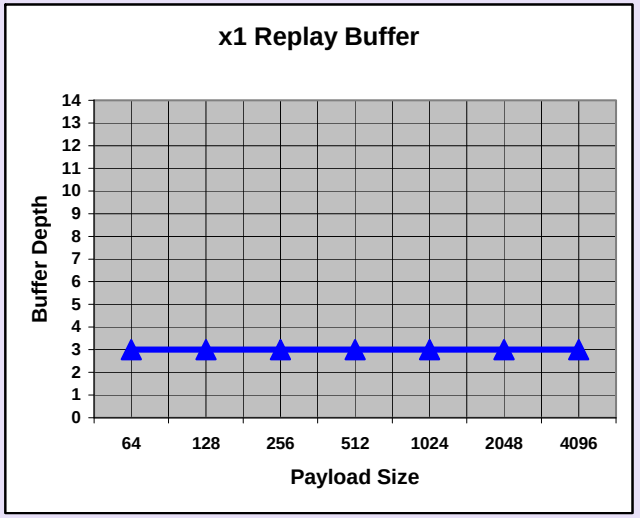


Replay Buffer – Lower Latencies

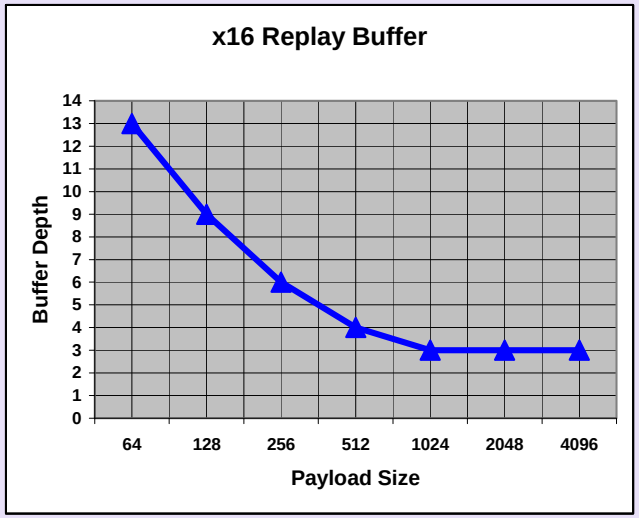


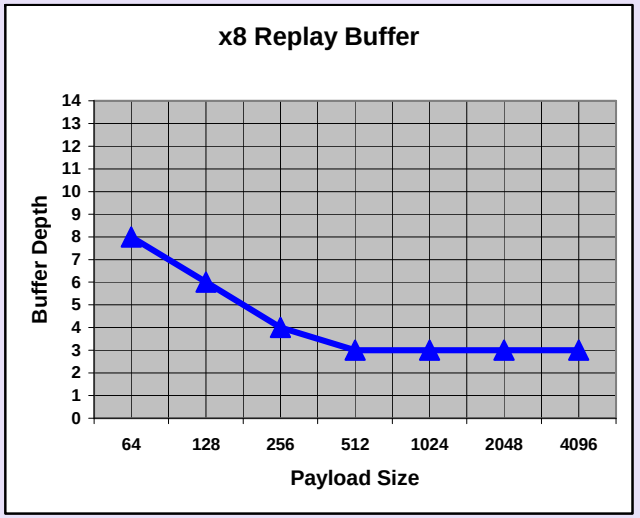


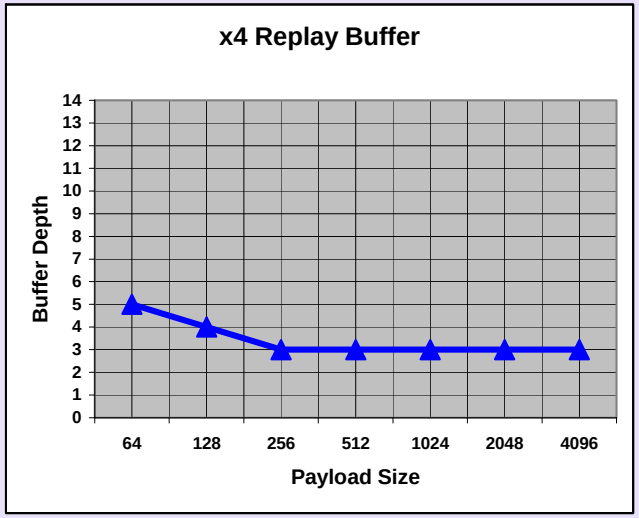


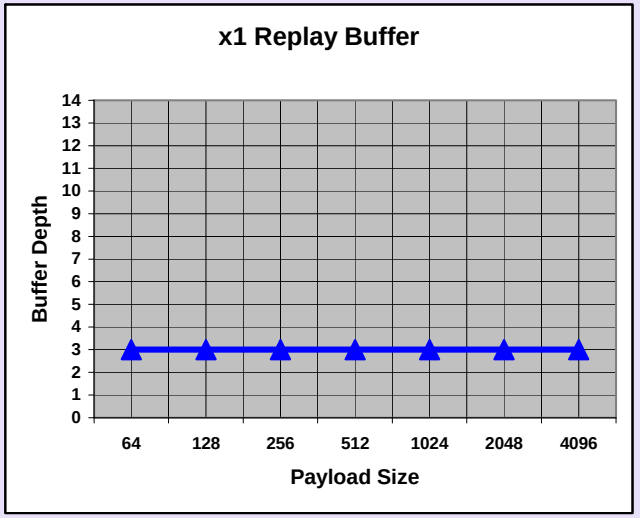


Replay Buffer – Higher Latencies









Latencies can be a big deal

- Do you know the device latencies in your system?
- Do you take them into consideration?
- Becomes important if you have a lot of small packets and you're concerned about performance

SR-IOV Parameters

- Considerations for Root Port versus Endpoint
- Physical and Virtual Functions
  - ✓ How many of each?
  - ✓ Additional configuration space and buffer resources
- ARI – Alternate Routing-ID Interpretation
  - ✓ For devices with > 8 functions
  - ✓ New configuration space capability structure and capability/control bits
- FLR – Function Level Reset
  - ✓ New capability and control bits

- ACS & ATS – not covered here

Agenda

- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

Function Support

- Consider ‘Function Co-location’ as mentioned in the SR-IOV Specification
  - ✓ Co-locates all Functions within the captured Bus Number
  - ✓ Does not require additional Bus Numbers to access VFs
  - ✓ Through the use of ARI, any combination of functions (PFs + VFs) up to 256 is supported
- Total functions supported = 256
  - ✓ Physical Functions (PFs) support
    - 0 – 7
  - ✓ Virtual Functions (VFs) support
    - VFs = 256 – PFs

Function Support 'Costs'

- VFs are described as "light-weight" PCIe functions
  - ✓ VFs require about 15-20% of the registers of a full configuration space
  - ✓ Gate counts for PF/VF configuration space registers can add up quickly in order to support large PF/VF configurations
  - ✓ Gate counts for BAR address decoding can also add up in order to support large PF/VF configurations

Arbitration/transmission gates

- Replay Buffer full/not full
- Flow Control credits
- Packet Type
- Virtual Channel/Traffic Class
- Port (switch)
- **Function** (and MFVC)
  - ✓Consider where this would best be handled
    - In the PCIe Port IP
    - In User Logic

ARI Support

- Root Ports
  - ✓ Device Capabilities 2: ARI Forwarding Support
  - ✓ Device Control 2: ARI Forwarding Enable
- Endpoints
  - ✓ ARI Capability and Control registers

- Requires specific system software support
- Disabled on reset
- Applications using more than 8 functions (physical and virtual)
  - ✓ Must also support a configuration that only consumes up to 8 functions

FLR Support

- Endpoints
- FLRs Targeting VFs or PFs must be supported
- Consider ‘handshake’ between user logic and port prior to initializing function-specific portion of configuration space
  - ✓ Need to respect 100ms FLR completion time limit
- Returning Configuration Request Retry status is okay (if needed) for this type of software initiated reset

SR-IOV Considerations (1)

- SR-IOV is not typically something added late in a chip development cycle
  - ✓ PCIe IP support for SR-IOV is only part of the story
  - ✓ User logic and software
- SR-IOV support can have a significant impact on chip functionality as well as how resources are provisioned

SR-IOV Considerations (2)

- Servers today are typically enabling up to 16 LPARs (system images)
  - ✓ Expect more system images in the future
- First generation of SR-IOV capable adapters likely to support 64-128 VFs anticipating expansion in number of system images
- Error handling becomes increasingly complex.
  - ✓ In case of error on certain VF (e.g. local resource allocation violation etc.) you need to stop traffic for that VF, without impacting other VFs.

SR-IOV Considerations (3)

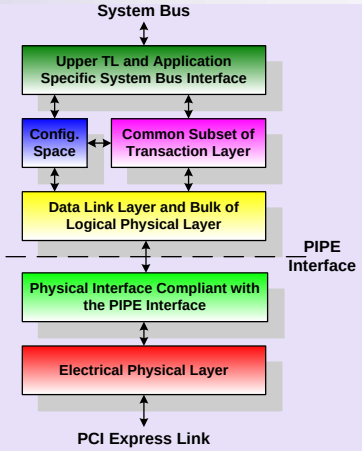
- Need to carefully consider application resources (not PCIe-specific), that cannot be shared between VFs:
  - ✓ Interrupts are unique per VF
    - MSI-X is typically used to allow multiple messages per function
  - ✓ Application layer resources, like DMA channels, communication FIFOs, local memory regions

Agenda

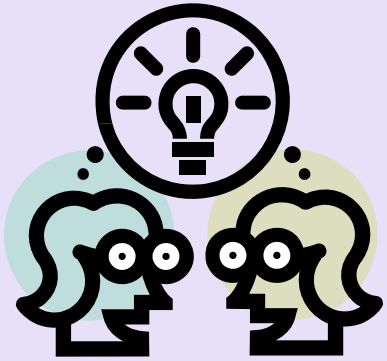
- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

‘Bottoms up’ Implementation Hierarchy

- PCIe Port IP



- User Logic



- Integration

IP Development Strategy (Phase 1)

- Define broad general feature set and provide flexible IP architecture to support
- Interlock with early adopter customer(s) and focus on subset of general feature set first
- Complete remaining features

IP Development Strategy (Phase 2)

- Respond to PCIe Base Spec. errata, as required
- Prioritize next set of enhancements
  - ✓ Example: ECNs or new specifications such as next generation data rate or IOV
- Essentially, IP continually evolves and needs to continually evolve along with the evolving PCIe and customer requirements

IP Development Strategy - Upgrades

- We had a generally very good PCIe 1.x PCIe IP solution, but…
  - ✓ Didn't easily lend itself to speed upgrades
  - ✓ Essentially 'maxed-out' at PCIe 1.1 level of support
- For PCIe 2.0 (and beyond), we learned from our PCIe 1.x experience and…
  - ✓ Brought forward what worked well (quite a bit actually) and ditched the rest
  - ✓ Incorporated generous amounts of customer feedback
- From PCIe 1.x to PCIe 2.0, we were successful at…
  - ✓ Maintaining and enhancing key interfaces
  - ✓ Developing a much more flexible architecture to accommodate major enhancements (such as IOV) and future speed upgrades

Agenda

- Introduction
- Customer Observations
- Considerations for Advanced Features
- SR-IOV Trade-offs
- PCIe 1.x/PCIe 2.0 Upgrades
- Summary

Take homes

- We (everyone in this room) are in a unique position to effectively enable customers to succeed with PCIe technology
- As PCIe evolves, it can be a challenge to keep pace at the various levels of development from PCIe IP to user logic to chip integration and verification
- SR-IOV needs to be addressed at all levels of the application, not only at PCIe port level
- Interface compatibility is key for successful integration of successive generations of PCIe Port IP
- Stick to ‘good engineering practice’ – due diligence while working through the details from basic to advanced PCIe implementations

Thank you for attending the
PCI-SIG Developers Conference 2008

For more information please go to
www.pcisig.com