
PCI
SIG®

Advanced Programming Interfaces for PCI Devices

Al Yanes,

Senior Technical Staff Member,
Engineering & Technology Services,

IBM

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Introduction

- Performance
  - ✓ More than feeds and speeds
  - ✓ What differentiates two I/O adapters with the same hardware capability
  - ✓ Device Drivers can impact hardware performance
  - ✓ Hardware programming interface is crucial in achieving best of breed performance

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Advantages of MSI

- Level Signal Interrupts
  - ✓ Do not push Posted Memory Writes
  - ✓ Could cause an MMIO Load to the adapter to obtain status
- Message Signaled Interrupts
  - ✓ Push Posted Memory Writes
  - ✓ Write the status of the operation to a predetermined location

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Interrupt Coalescing

- Methods to avoid one operation per interrupt
  - ✓ Establish a receive queue control block that status gets written to when the operation gets completed
  - ✓ First interrupt occurs indicating status for the first completed task has been written but additional status for subsequent operations can occur and no additional interrupts will happen
  - ✓ Device driver rearms the hardware only after servicing all receive queue entries thus allowing for a better ratio of interrupts per task completed
  - ✓ Hardware only generates an interrupt after a predetermined count threshold has been achieved of tasks completed
  - ✓ Hardware only generates an interrupt after a predetermined time threshold after the first task has been completed

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

IO Adapter Performance

- Three Worst things for IO adapter Performance
  - ✓ MMIO Loads
  - ✓ MMIO Loads
  - ✓ MMIO Loads
- Why are MMIO Loads so bad
  - ✓ Processor stalls or has to do a context switch waiting for the MMIO Load Reply Data
  - ✓ MMIO Load Reply Data takes a long time due to PCI ordering rules
  - ✓ MMIO Load Request have to push MMIO store data
  - ✓ MMIO Load Reply Data have to push DMA store data

Ways to reduce MMIO Loads

- Add Load/Store Assist Hardware
  - ✓ Architect a control block that the hardware reads and determine what register values need to be written back into the control block
- Write status of operation into control block
  - ✓ Only do MMIO loads in the error case thus ensuring good performance
- Send Messages (write only) when in need to interrogate the hardware
  - ✓ Adapter responds with message containing the status

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Descriptors - Usage

- Things to strive for
  - ✓ Address aligned data blocks
  - ✓ Contiguous address help due to the prefetching done by chips
  - ✓ Bigger operations allow more efficient bus utilization reducing the overhead of the protocol
- Things to avoid
  - ✓ Small (less than the cache line) DMA writes cause read modify writes in system memories

Descriptors

- Control block linked list which can contain multiple pairs of DMA address and bytecount which inform the adapter where and how much data to move
  - ✓ Optionally the control block can have an end flag
  - ✓ Polling for work causes unnecessary reads similar to pci reads causing overall system performance to go down

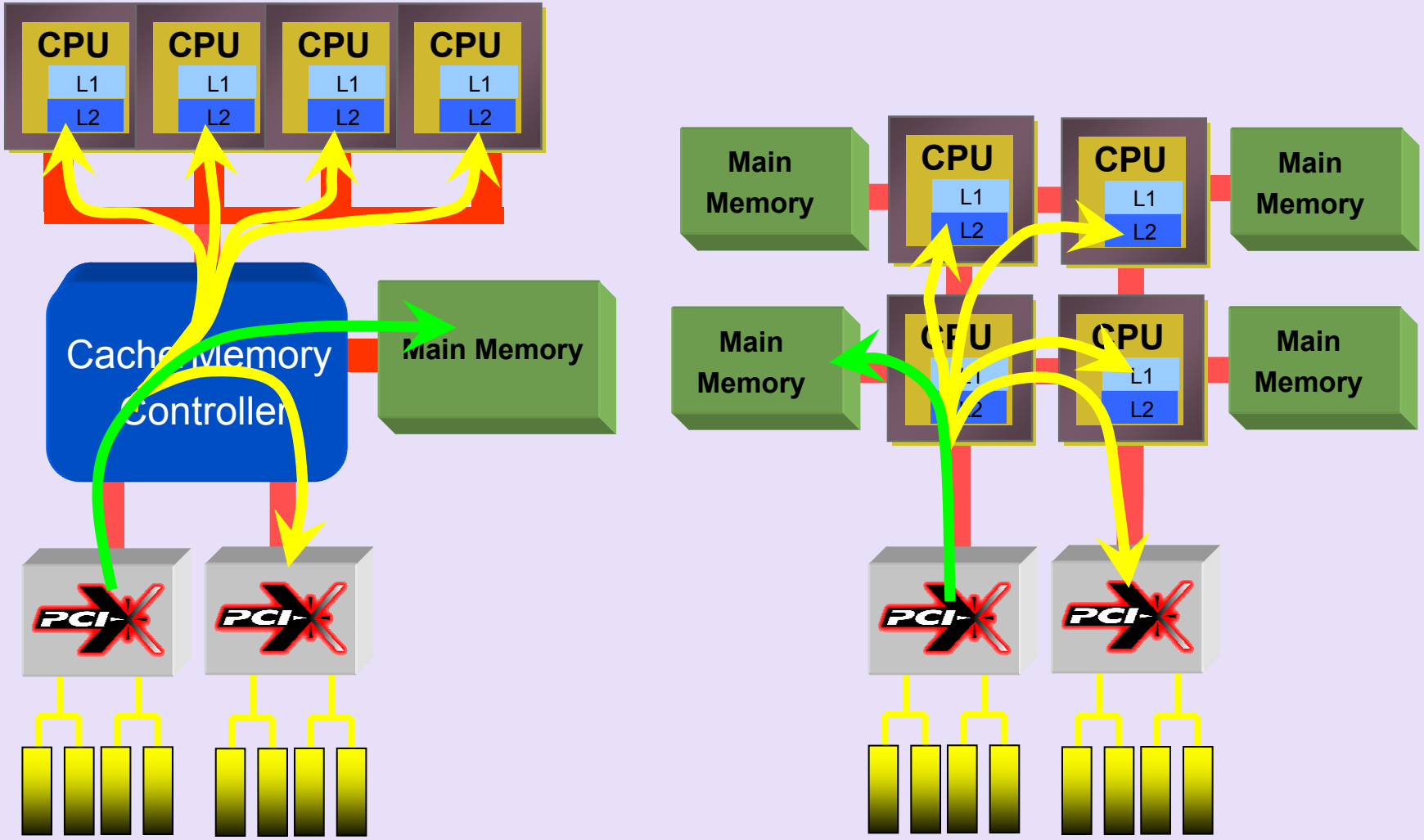
Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Effects of Cache Snooping

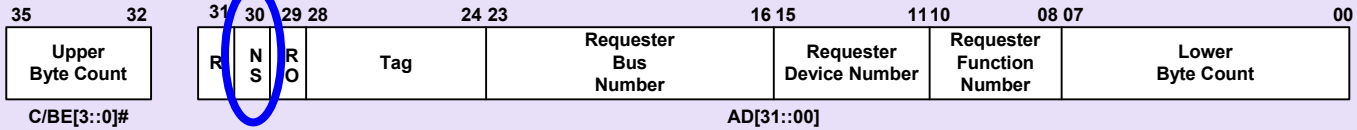
- Even a cache-miss has system impact
- Snooping a write consumes bandwidth
  - ✓ Internal chipset buses
  - ✓ External cache subsystems
  - ✓ CPU front-side bus or inter-CPU links
  - ✓ Internal CPU buses
- Snooping a read
  - ✓ Consumes bandwidth
  - ✓ Increases latency: Completion must wait for all snoops to complete

There may be more snoop impact than you think!

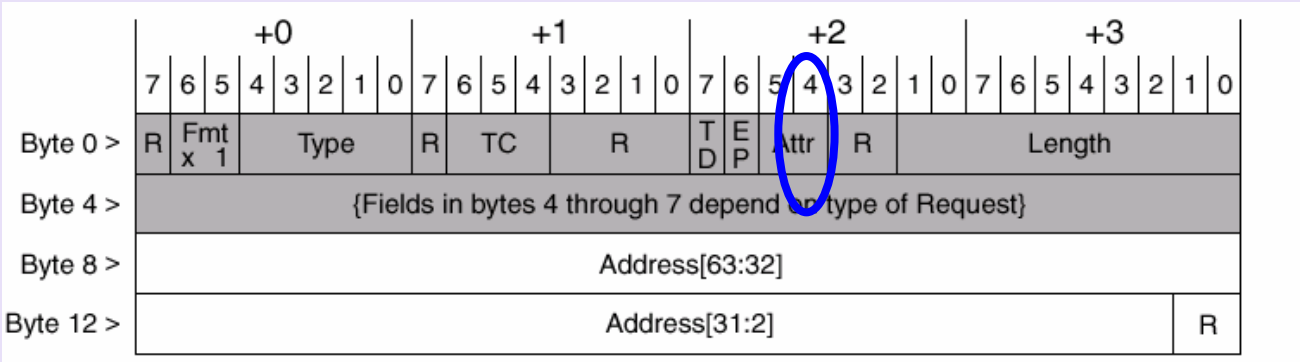


Don't Snoop If Guaranteed to Miss

- Device sets "No Snoop" attribute bit for each transaction
  - ✓ PCI-X attribute phase



  - ✓ PCI Express TLP header



When to Set the No Snoop Bit

- Not specified
- Dependent on system design
- Examples
  - ✓ Non-cacheable address range
  - ✓ Software-assisted cache-flush algorithms
  - ✓ Device-to-device transfer temporary buffer

Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

PCI Transaction Ordering Rules

- Two classes of rules
  - ✓ Required blocking to support “strong write ordering” programming model
  - ✓ Required passing to avoid deadlock (not discussed further in this presentation)
- Strong write ordering programming model definition
  - ✓ If one device writes two locations in a specific order, no other device can ever observe those locations updated in the opposite order

Device A (writer) execution sequence

```
.
.
Write “payload”
.
.
Write “control”
.
.
```

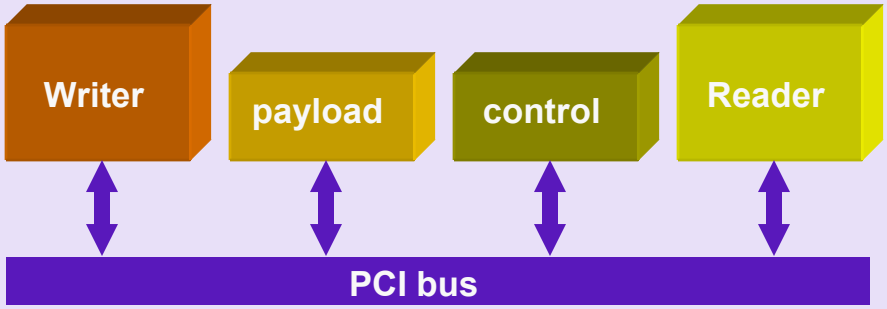
Device B (reader) execution sequence

```
.
.
Read “control”
.
.
Read “payload”
.
.
```

If Device B sees a new value of “control,” it must always see the new value of “payload.”

PCI Transaction Ordering Rules

- Strong write ordering would be automatic in system with atomic transactions

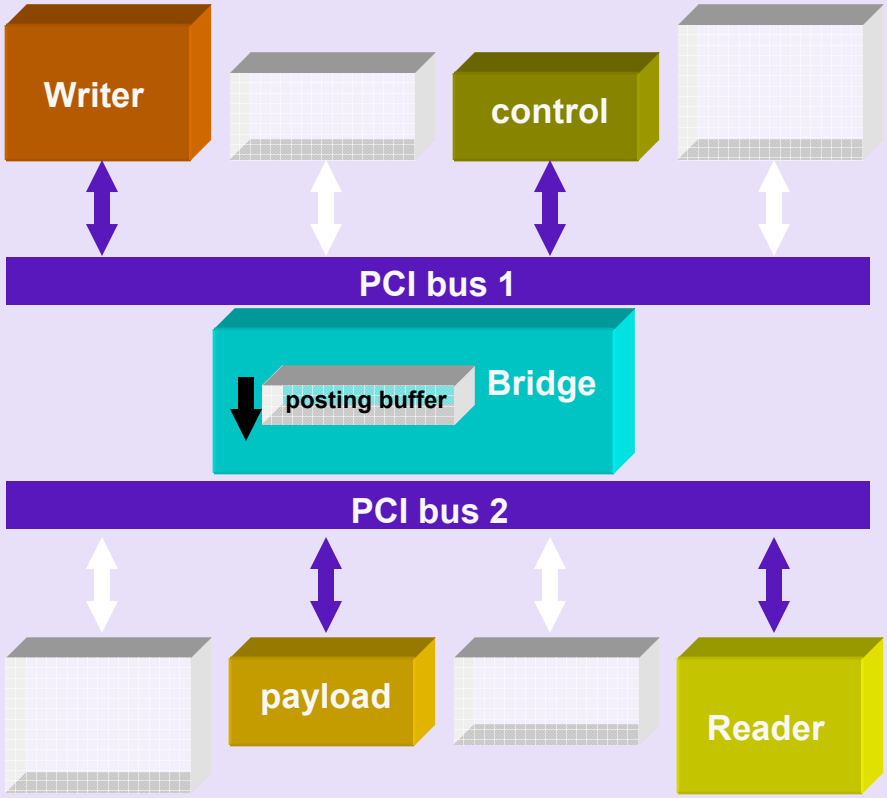


Transaction sequence on bus

Bus owner	Transaction
.	
.	
Reader	Read "old control"
Reader	Read "old control"
Writer	Write "new payload"
Reader	Read "old control"
Writer	Write "new control"
Reader	Read "new control"
Reader	Read "new payload"
.	
.	

PCI Transaction Ordering Rules

- Multiple buses connected by bridges
- Cross-bus writes posted to increase performance



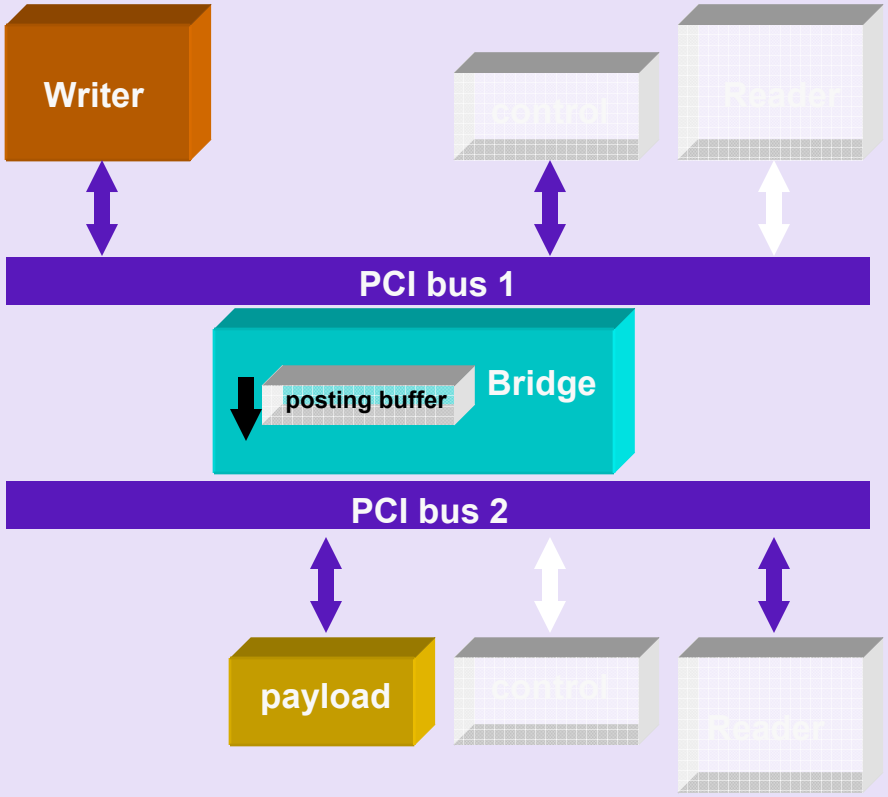
Transaction sequence on buses (completions only)

Transactions Bus 1	Transactions Bus 2
.	.
.	.
Read "old control"	Read "old control"
Read "old control"	Read "old control"
Write "new payload"	Read "old control"
(posted)	Read "new control"
Read "old control"	Read "old payload"
Write "new control"	Write "new payload"
Read "new control"	.
.	.
.	

Broken!

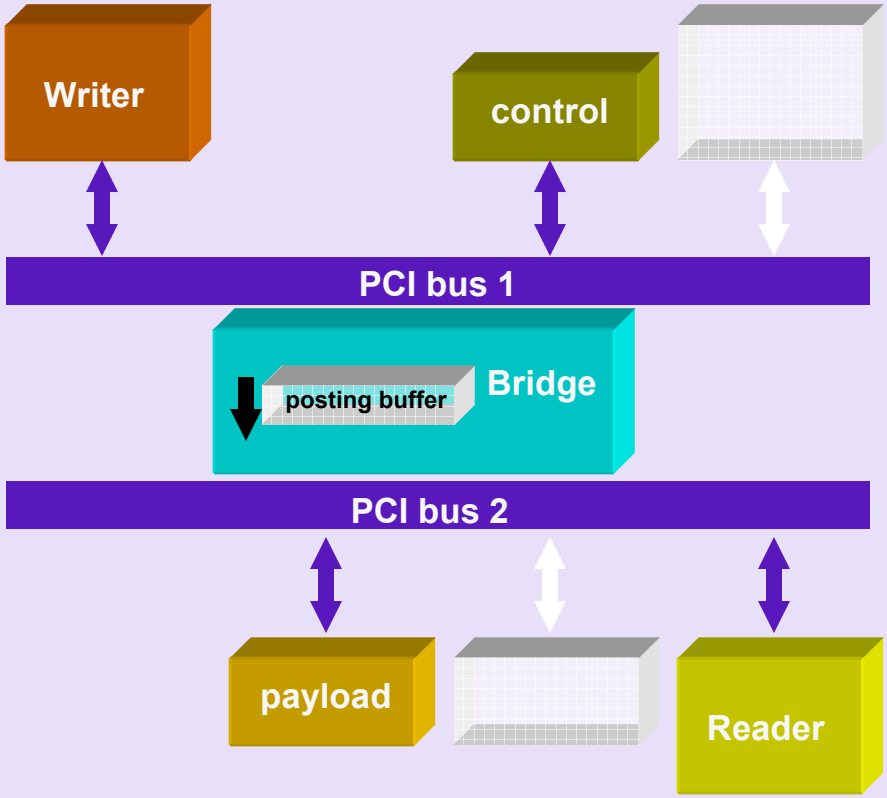
PCI Transaction Ordering Rules

- 4 devices in each of 2 locations yields 16 total cases
- Eliminate half by symmetry (consider Writer only on one side)
- Eliminate another half for non-posted payload (consider only cases in which payload is opposite Writer)
- Results in 4 interesting cases (Reader and control in each of 2 places)



PCI Transaction Ordering Rules

- Case 1
- Rule: Read Completion must pull previous posted write toward Reader



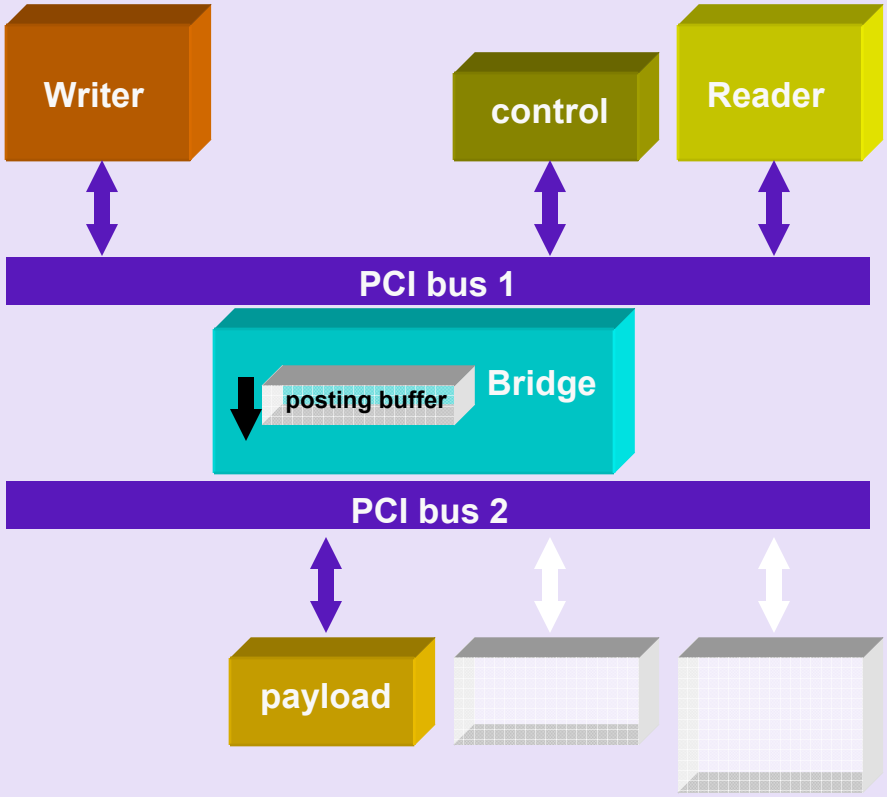
Transaction sequence on buses (completions only)

Transactions Bus 1	Transactions Bus 2
.	.
.	.
Read "old control"	Read "old control"
Read "old control"	Read "old control"
Write "new payload" (posted)	Read "old control"
Write "new control"	Write "new payload"
Read "new control"	Read "new control"
.	Read "new payload"
.	.
	.

Posted write forced out of bridge before read completion

PCI Transaction Ordering Rules

- Case 2
- Rule: Read Request must push previous posted write away from Reader



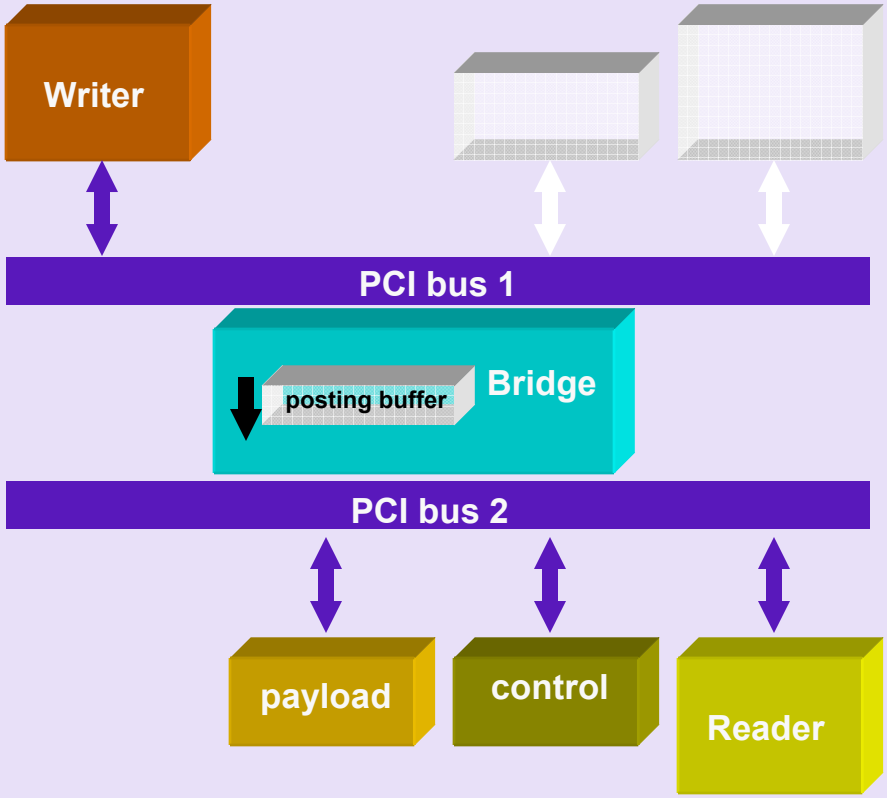
Transaction sequence on buses (completions only)

Transactions Bus 1	Transactions Bus 2
.	.
.	.
Read "old control"	.
Read "old control"	.
Write "new payload"	.
(posted)	.
Read "old control"	Write "new payload"
Write "new control"	Read "new payload"
Read "new control"	.
Read "new payload"	.
.	.
.	.

Posted write forced out of bridge before read request

PCI Transaction Ordering Rules

- Case 3
- Rule: Writes must exit bridge in same order they entered it



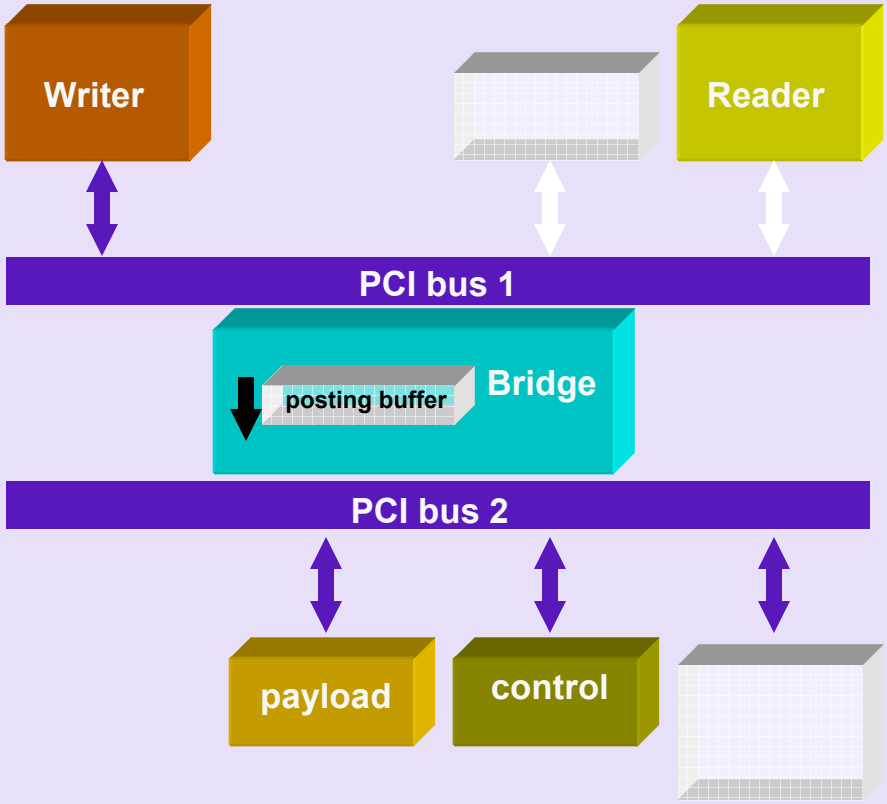
Transaction sequence on buses (completions only)

Transactions Bus 1	Transactions Bus 2
.	.
.	.
.	Read "old control"
Write "new payload"	Read "old control"
(posted)	Write "new payload"
Write "new control"	Read "old control"
(posted)	Write "new control"
.	Read "new control"
.	Read "new payload"
.	.

Later writes must push earlier writes through the bridge

PCI Transaction Ordering Rules

- Case 4
- Same as Case 3



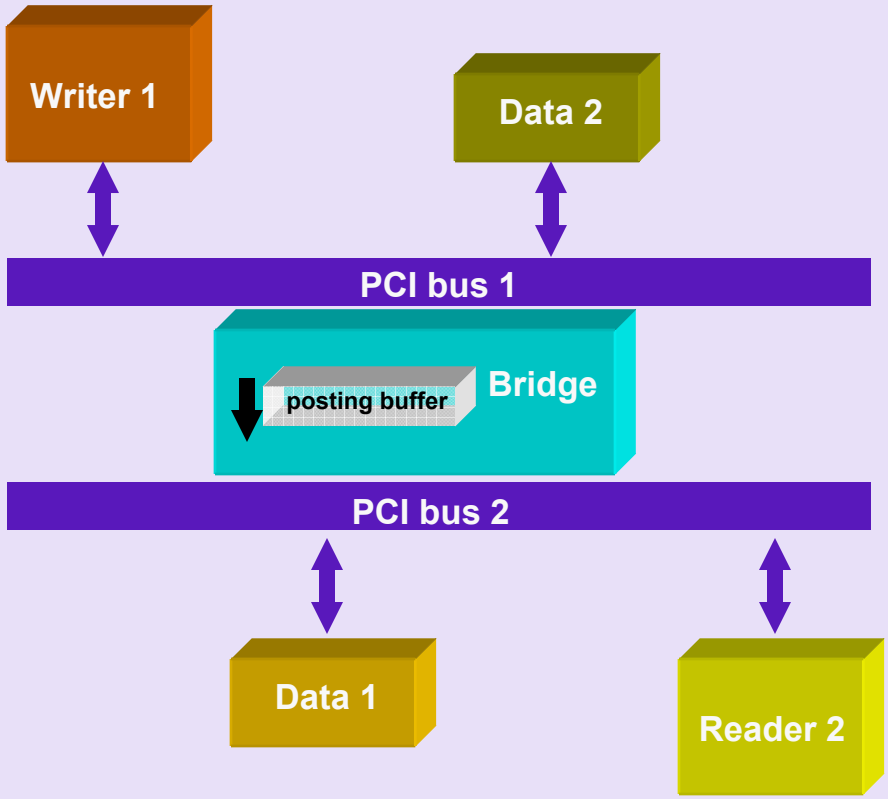
Transaction sequence on buses (completions only)

Transactions Bus 1	Transactions Bus 2
.	.
.	.
Read "old control"	.
Read "old control"	Read "old control"
Write "new payload"	Read "old control"
(posted)	Write "new payload"
Write "new control"	Write "new control"
(posted)	Read "new control"
Read "new control"	Read "new payload"
Read "new payload"	.
.	.
.	

Later writes must push earlier writes through the bridge

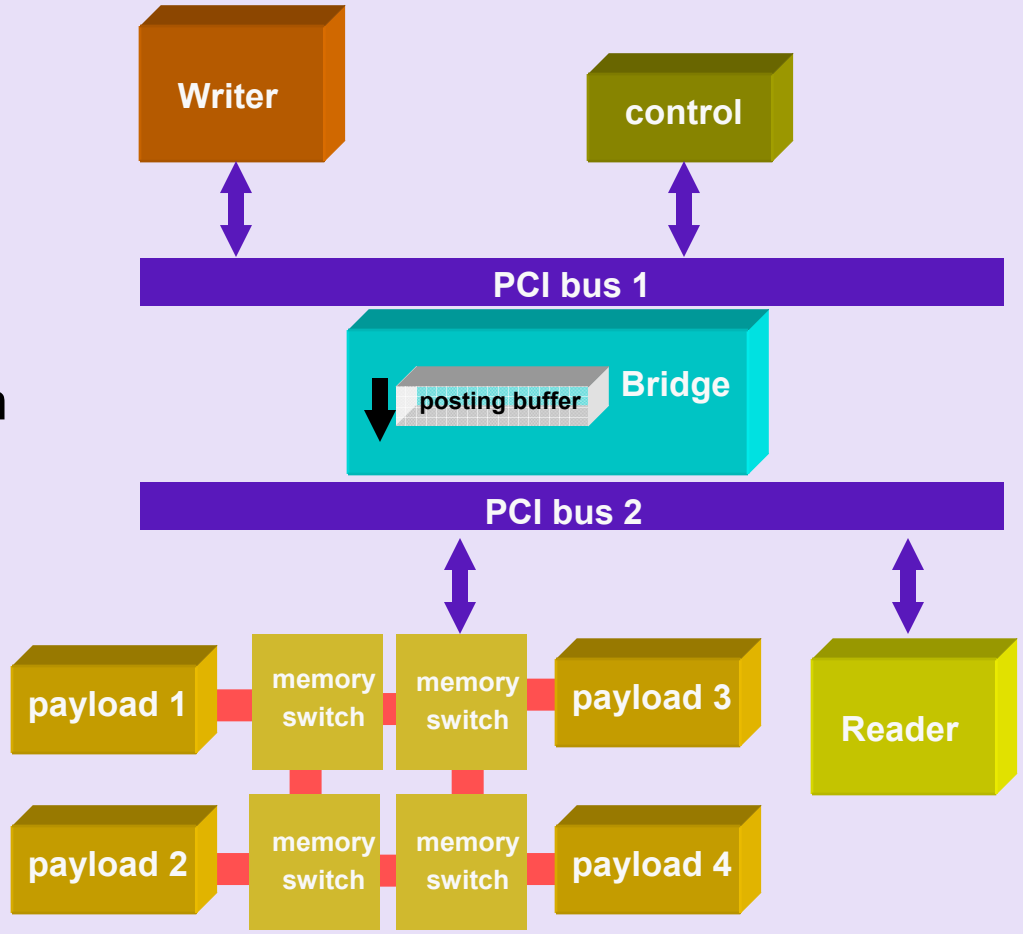
Side Effects of Required Blocking Rules

- PCI bridges have no way of knowing which transactions are associated with which writer/reader pairs
  - ✓ Read completions can be blocked by writes from unrelated Writer (common problem addressed by Relaxed Ordering attribute)
  - ✓ Read requests can be blocked by writes from unrelated Writer (uncommon problem not addressed by Relaxed Ordering attribute)



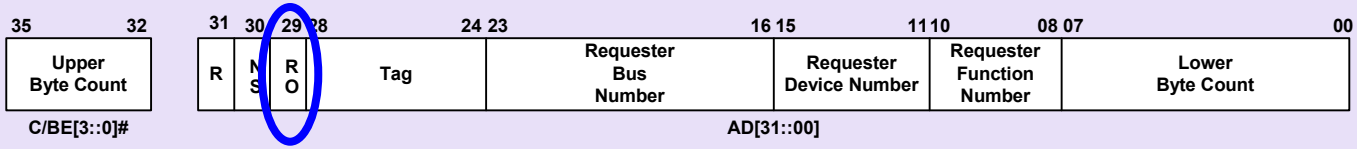
Side Effects of Required Blocking Rules

- In-order write BW much lower than unordered write BW in complex networked memory systems
  - ✓ In-order writes must wait for round-trip acknowledgement of each write before starting next one
- Payload writes commonly require no ordering with respect to other payload
  - ✓ Keeping payload writes in order unnecessarily lowers memory performance

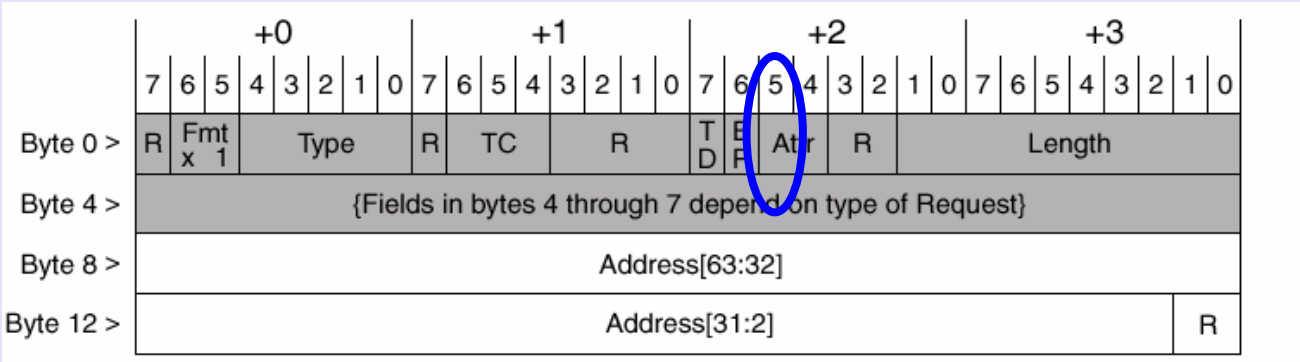


Relaxed Ordering Attribute Bit

- Device sets “Relaxed Ordering” attribute bit for each transaction
  - ✓ PCI-X attribute phase



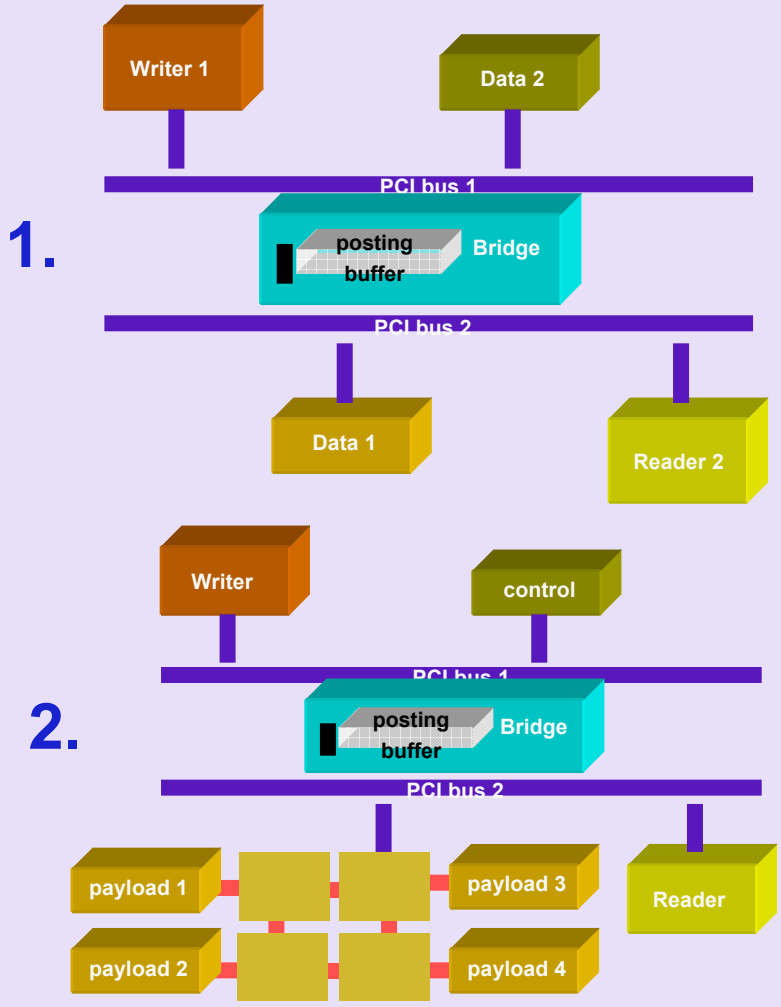
  - ✓ PCI Express TLP header



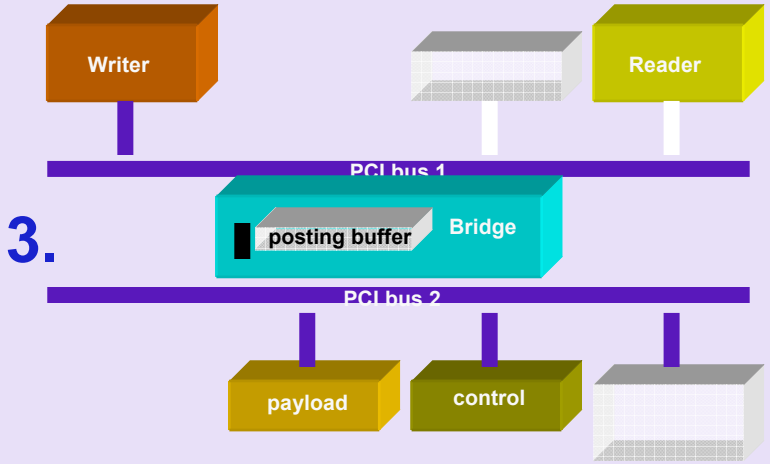
Relaxed Ordering Attribute Bit

- PCI-X behavior
  - ✓ Read Completions allowed to pass posted writes moving in same direction
  - ✓ Posted writes allowed to pass other posted writes moving in same direction (host bridge only; not PCI-to-PCI bridge)
- PCI Express behavior
  - ✓ Same, except PCI-to-PCI bridges (switches) included in posted write reordering rule

Relaxed Ordering Attribute Bit Application



- Device hardware sets "Relaxed Ordering" for
  1. All payload reads
  2. All payload writes
  3. Control reads if payload and data are colocated
- Other cases supported by specific programming model



Agenda

- Introduction
- Message Signaled Interrupts (MSI)
- Interrupt coalescing
- Avoiding Memory-Mapped I/O (MMIO)
- Descriptors
- Minimizing cache snooping
- Relaxed ordering
- Conclusion

Conclusions

- Use MSI for interrupts
- Coalesce as many interrupts as possible
- Avoid MMIO
- Use queues of efficiently aligned work items (“descriptors”)
- Set the No Snoop attribute bit whenever you can guarantee the location is not in any cache
- Eliminate unnecessary transaction blocking by setting the Relaxed Ordering attribute bit
  - ✓ Simple hardware rules can automatically cover most cases for “payload/control” programming models
  - ✓ Software can help identify other cases

Thank you for attending the
PCI-SIG Developers Conference 2005.

For more information please go to
www.pcisig.com


PCI
SIG®