Address Translation Services

Michael Krause (HP, co-chair)

Mark Hummel (AMD)

David Wooten (Microsoft)

Topics

- Introduction to Address Translation for DMA
- Problem Statement
- Address Differentiated Memory Requests
- Translation Requests
- Transaction Completions
- Invalidation Request
- Invalidation Completions
- Configuration

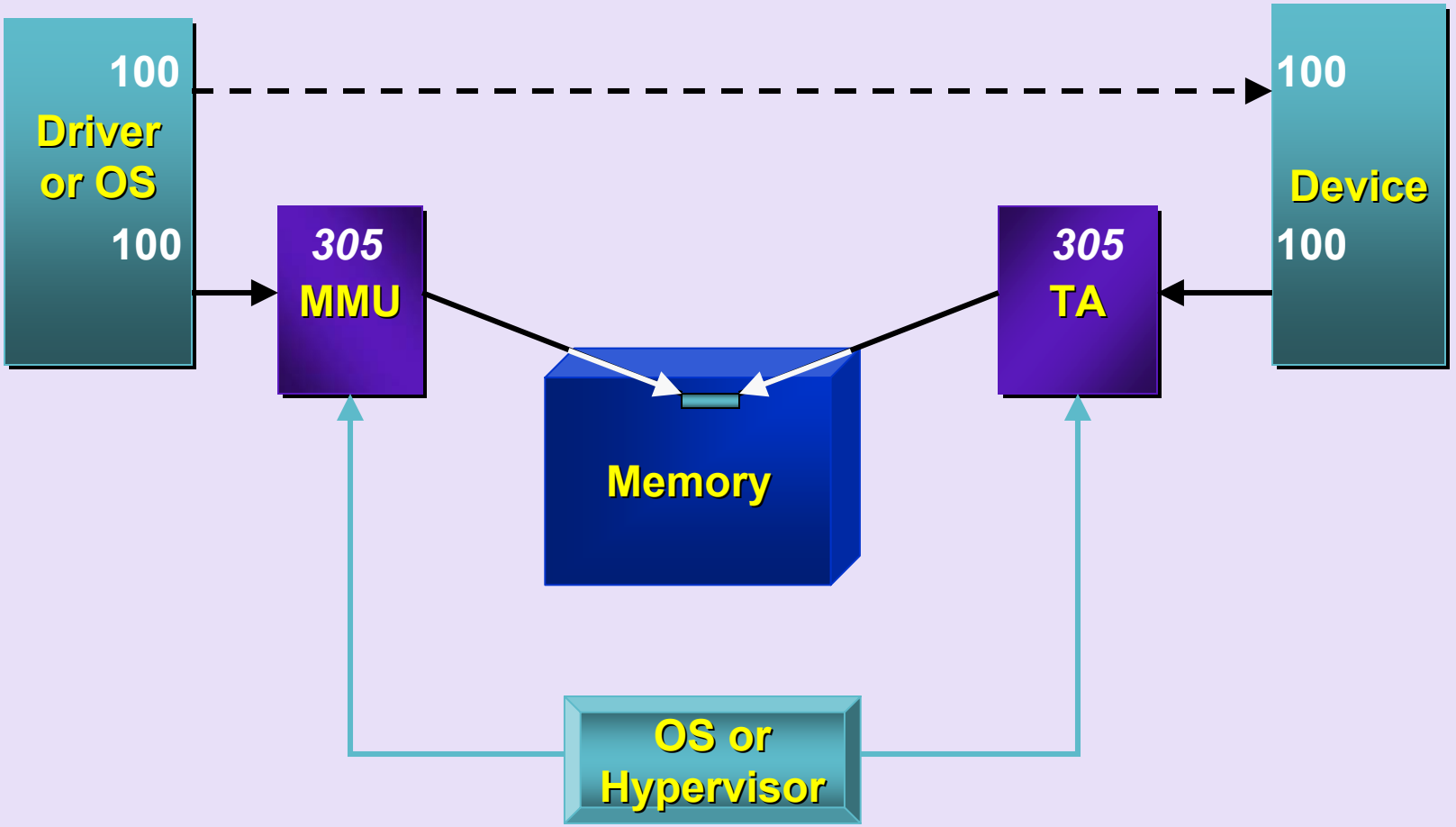
ATS Specification Status

- The material in this presentation represents the contents of the 0.7 version of the Address Translation Services Specification.
  - ✓ http://www.pcisig.com/members/downloads/specifications/pciexpress/specification/draft/ats-spec-07_draft-060327.pdf
- The technical details of 0.7 specifications are considered to be stable unless a specific problem is found and needs to be fixed.
- The draft 0.9 specification will arrive shortly.

Introduction to DMA Address Translation

- Address translation and an access check is applied to DMA request from an IO device.
- In the Address Translation Services (ATS) specification, the entity doing the translation and checking is called a Translation Agent (TA).

DMA Remapping Illustration



DMA Remapping on High-Volume systems

- Multiple companies have announced support for DMA remapping in future chipsets for the volume market
- These systems use tree-structured translation tables that are similar to CPU tables
- Different tree for each Bus/Dev/Func is possible
  - ✓ Devices can share a device address space
  - ✓ A devices can have a dedicated address space

Potential Issues of DMAr

- Increased latency of accesses
  - ✓ Might need one or two accesses to find address of tree associated with a BDF
  - ✓ Might need 3 or 4 accesses to walk the tree
- Translation caches (*ATC* or *IOTLB*) will be necessary to reduce overhead.
- Caches may not provide good behavior if not sized correctly
  - ✓ Only two possibilities for sizing caches: too large or too small
- “Untimely” latency may cause issues with isochronous devices

ATS to the "rescue"

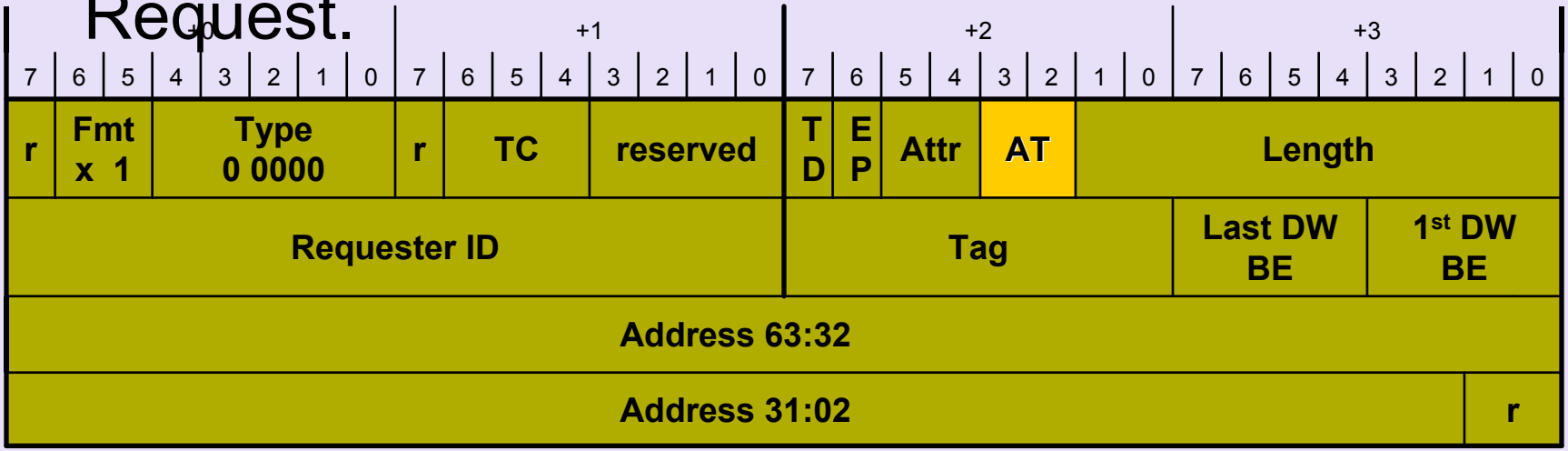
- ATS attempts to mitigate the impact of DMA Remapping by providing ways for Endpoints to participate in translation cache management
  - ✓ Device can maintain their own cache of translations – an "Address Translation Cache" (ATC)
  - ✓ TA provides table-walking services to device to avoid excess bus traffic – also means that translation table format is uniform in a system
- Device manages its ATC using its intimate knowledge of future access pattern
  - ✓ Look-ahead for isochronous devices to avoid "untimely" table walk latencies.
  - ✓ High-load devices (graphics) don't thrash ATC in TA.
  - ✓ Application specific caching in devices – ring buffer
  - ✓ Enable peer-to-peer in virtualized bus

New Protocols for ATS

- Differentiated memory address type – device will be issuing Requests that use both translated and non-translated addresses
- Translation Request – Address Translation Cache (ATC) in device requests a translation from central TA
- Translation Completion – translation is returned in response to Translation Request
- Invalidation Request – when change occurs in central table, need to inform remote ATCs
- Invalidation Completion – when ATC completes the invalidation operation, it needs to tell TA.

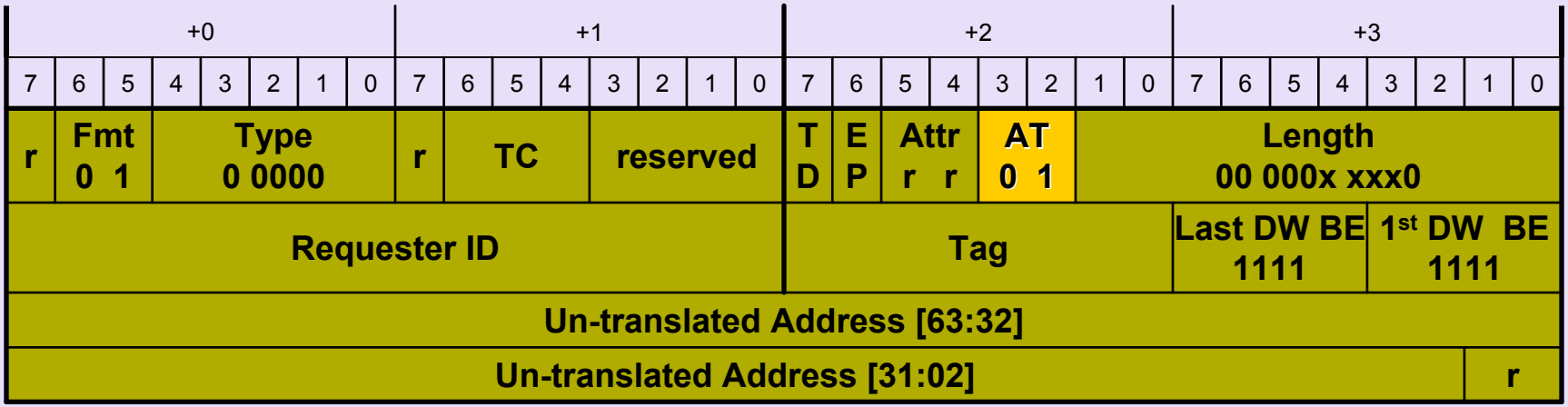
Differentiated Memory Request

- Previously reserved field in Memory Request used to differentiate Address Type (AT) in Request.



AT	Meaning	AT	Meaning
00b	Default (un-translated)	10b	Translated
01b	Translation Request	11b	Reserved

Translation Request



+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
r	Fmt 0 1		Type 0 0000					r	TC			reserved				TD	EP	Attr r r		AT 0 1		Length 00 000x xxx0										
Requester ID																Tag								Last DW BE 1111				1st DW BE 1111			
Un-translated Address [63:32]																															
Un-translated Address [31:02]																														r	

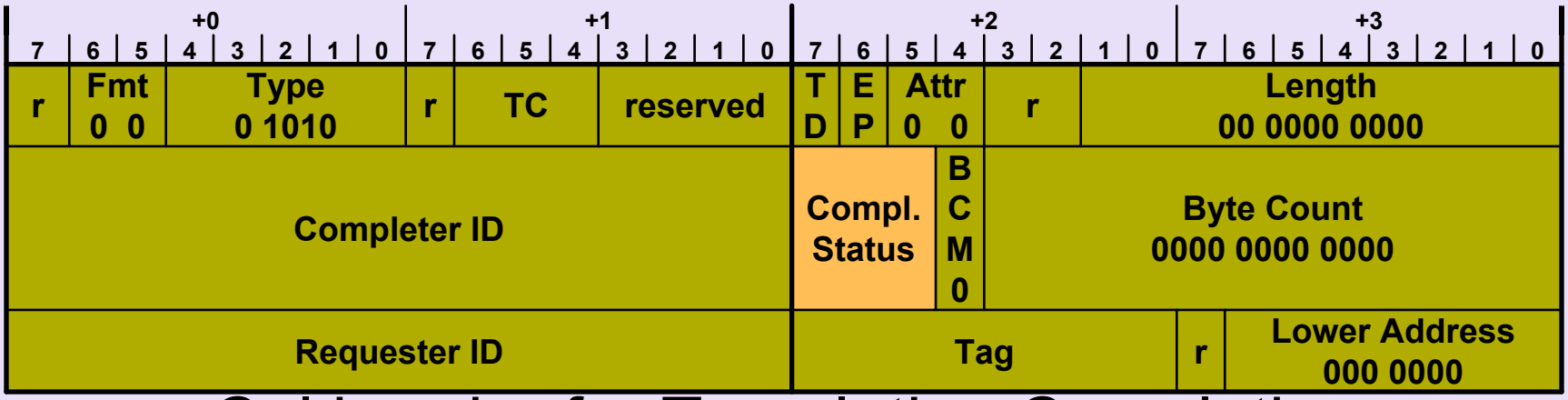
- Translation Request is Memory Read Request with AT field set to 01b.
- Request is for TA to return the translated reference for the memory range starting at the location referenced in the Un-translated Address field.

Translation Request (cont.)

- Always uses 64-bit form of address
- *Length* always an even number of dwords up to RCB
- Device can request translations for a range of addresses
  - ✓ range is determined from *Length* and Smallest Translation Unit (STU)*
  - ✓ Size in bytes of range request is *Length*/2 * STU

*Note: The STU is specified in binary multiples of 4KB and is nominally the same size as a page on the system.

Translation Completion
Error Completion



Cpl header for Translation Completion

Value	Status	Meaning
000b	Success	In Cpl, means no translation found. Nominally means that table walk did not reach a leaf (page table pointer) entry.
001b	Unsupported Request	Translation Requests from this Function are not supported by the TA.
100b	Completer Abort	Error in the TA. Translation Request may be retried.
All others	Reserved	Reserved – malformed packet

Translation Completion
Normal Completion

+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
r	Fmt 1 0		Type 0 1010					r	TC			reserved				TD	EP	Attr 0 0		r				Length							
Completer ID																Compl. Status 000			BCM 0	Byte Count											
Requester ID																Tag								r	Lower Address						

CplD header for Translation Completion

Translated Address [63:32]																															
Translated Address [31:12]																				S	N	Reserved								U	W	R

Translation Completion translation entry (1 of N)

Translation Completion (cont.)

Field	Meaning
S	**Size of translation:** This field is 0b if the translation applies to a 4KB range of memory. If this field is 1b, then the translation applies to a range of memory that is larger than 4KB. See 3.3.1
N	**Non-snooped accesses:** If this field is 1b, then the read and write requests that use this translation **must not set the No Snoop bit in the Attribute field**. If it is zero, then the Endpoint may use other means to determine if No Snoop should be set.
Reserved	These bits shall be ignored by the ATC.
U	**Un-translated access Only:** When this field is set to 1b in a Translation Completion entry, the indicated range **may only be accessed using un-translated addresses** and the Translated Address field of this Translation Completion entry **may not be used in a subsequent Read/Write Request with AT set to *Translated***. This value may be cached if R or W is set to 1b.
R,W	**Read, Write –** These two fields indicate what the transaction types that are allowed for the requests using the translation. If neither field is Set, then the translation is not valid and all the remaining fields of this dword are undefined. **A value with R=W=0b may not be cached in the ATC**.

Translation Completion (cont.)

- TA can return 0 or more translations
  - ✓ TA decides
  - ✓ The TA may not return more than *Length* dwords
- May take one or two CplDs to close out the request.
- In a CplD, each translation entry covers the same sized range of addresses.
  - ✓ Smallest range is STU
  - ✓ All ranges will have some overlap with requested range
  - ✓ All ranges in the completion must be the same size
    - – If completion in two CplDs, then ranges in both must have same size.
  - ✓ Different completions can have different range size.

Range Determination

Address Bits							S	Translation Range Size in Bytes
63:18	17	16	15	14	13	12		
x	x	x	x	x	x	x	0	4K
x	x	x	x	x	x	0	1	8K
x	x	x	x	x	0	1	1	16K
x	x	x	x	0	1	1	1	32K
x	x	x	0	1	1	1	1	64K
x	x	0	1	1	1	1	1	128K

- Starting with S, look for first 0b (bit N)
- Range size is 2^(N+1) bytes

Translation Completions
Multiple CplDs

- TA may split a translation completion any time *Length* is greater than 2
- Since Length is no larger than allowed by RCB, completion will not take more than two CplDs
- In all CplD:
  - ✓ Byte Count indicates bytes remaining to complete request, including the bytes in the “current” CplD
  - ✓ Length indicates the number of dwords in the current Cpld.
- For Translation Completions, in the first CplD, Lower Address is set to:

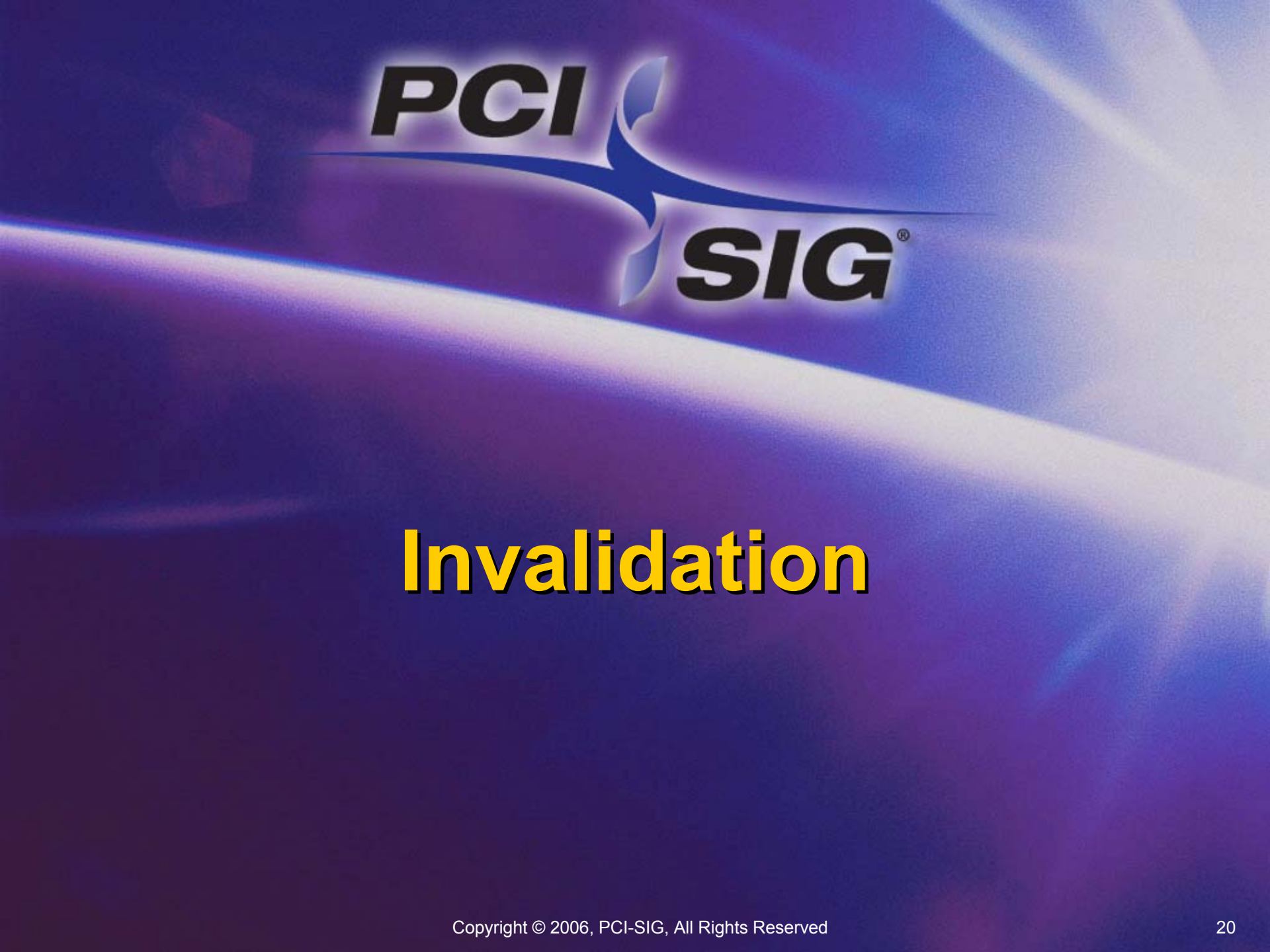
(000 0000b) – (Length * 4)

Example: if first CplD contains 2 translations the Lower Address is:

(000 0000b) – (00 0000 0100b * 100b) = (111 0000b)

Translation Completions
Missing CplDs

- If *Bytes* > (*Length* * 4), then this is a first CplD of a two CplD Completion.
- Else if *Bytes* < (*Length* * 4) then this is a malformed TLP
- Else, if *Lower Address* + *Byte Count* is not a multiple of RCB, then this is a second CplD of two; and if previous Transaction Completion CplD with same Tag was not received, then a CplD is missing.



Invalidation

Invalidation

- Purpose
  - ✓ Maintain consistency between TA and ATC
- Mechanism
  - ✓ Invalidate Request Message
  - ✓ Invalidate Complete Message

Invalidate Request

+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	Fmt 1 1		Type 1 0010					R	TC			Reserved				TD	EP	Attr 0 0		R		Length 00 0000 0001										
Requester ID																0	0	0	ITag					Message Code 0000 0001							
Device ID																Reserved															
Reserved																															

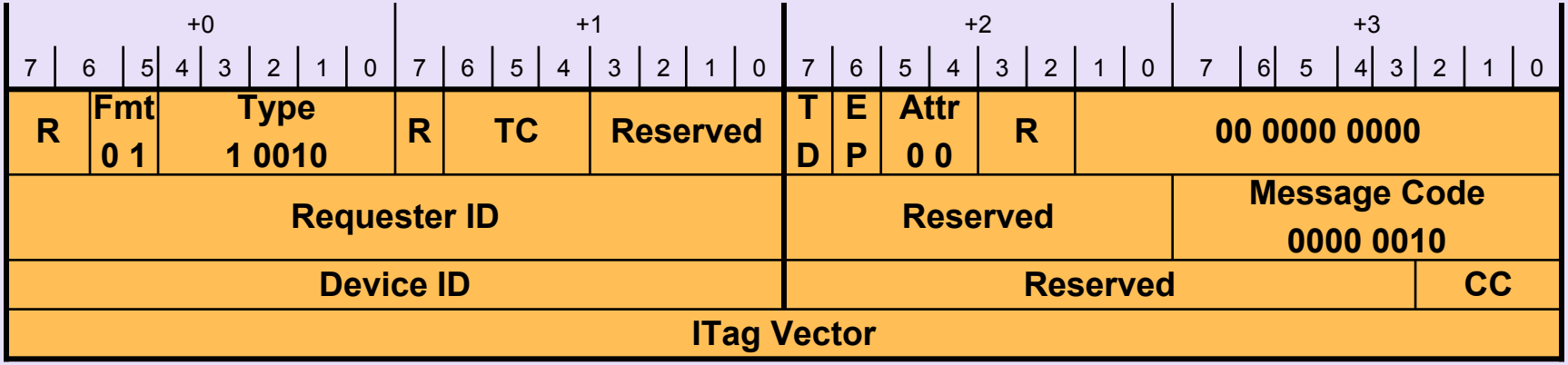
Un-translated Address [63:32]		
Un-translated Address [31:12]	S	Reserved

- MSGD packet with 4 DW of header and 2 DW of data
  - ✓ Message code = 0000 0001b
  - ✓ Route by Device ID
- ITag
  - ✓ Invalidate tag – Uniquely identifies each Invalidate Request
- Un-translated Address
  - ✓ Starting address of block to be invalidated
- S
  - ✓ Size bit (same encoding as in Translation Request)

Invalidation Range Constraints

- Each Invalidate Request specifies a single memory Region
  - ✓ Power of 2 in size
  - ✓ Naturally aligned
  - ✓ Must be at least as big as a STU (Standard Translation Unit)
  - ✓ S field and least significant address bits encode range size

Invalidate Completion



+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	Fmt 0 1		Type 1 0010					R	TC			Reserved				TD	EP	Attr 0 0		R				00 0000 0000							
Requester ID																Reserved								Message Code 0000 0010							
Device ID																Reserved														CC	
ITag Vector																															

- MSG packet with 4 DW of header
  - ✓ Message code = 0000 0010
  - ✓ Route by Device Id
- ITag Vector
  - ✓ Invalidation tag vector – Uniquely identifies each Invalidate Completion
  - ✓ Multiple completions may be compressed into a single response
- CC
  - ✓ Completion Count – Indicates number of completions that have been sent for the corresponding request (0 => 8 completions)
  - ✓ Always 1 for single TC devices

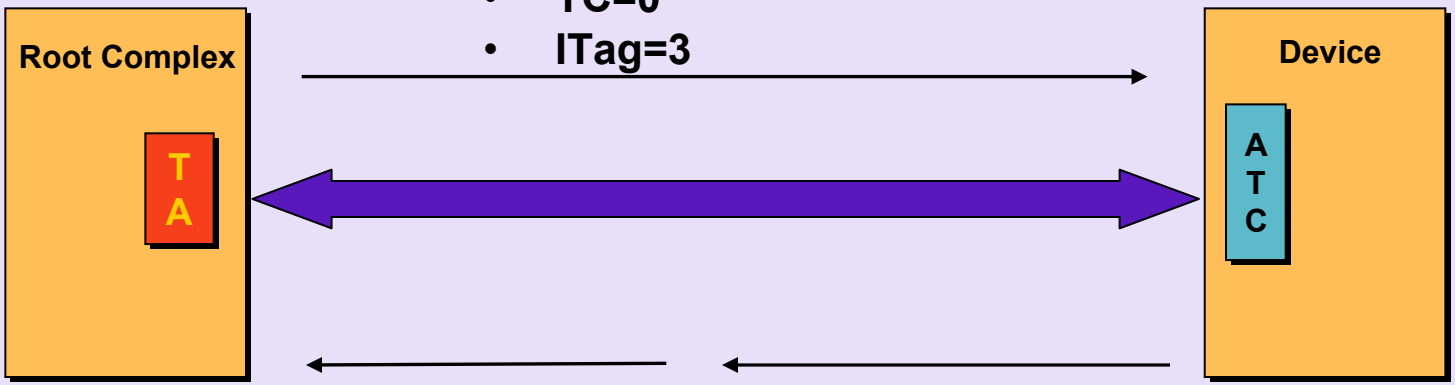
Invalidation Completion Semantics

- Properties to satisfy
  - ✓ Maintain consistency
  - ✓ Prevent silent data corruption
  - ✓ Prevent data leakage
- Invalidate Complete must not be returned until:
  - ✓ Stale address translation is flushed from ATC
  - ✓ Conflicting outstanding writes are pushed to the TA
  - ✓ Conflicting outstanding reads are either completed or tagged for discard

Invalidation Flow – Single TC

2) Flush matching ATC entries. Drain or discard conflicting Reads .

1) Invalidate Request
- TC=0
- ITag=3

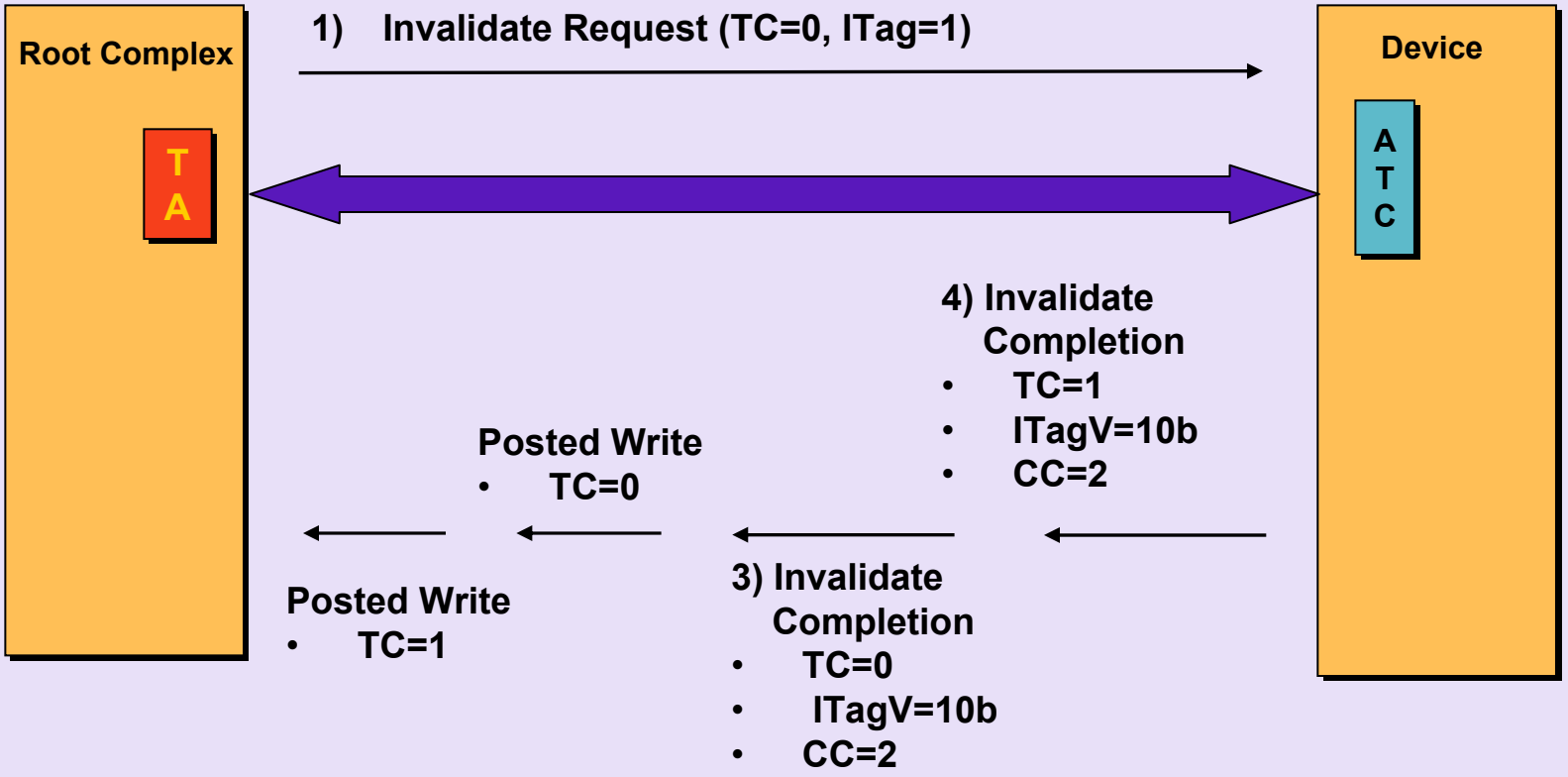


Posted Write
- TC=1

3) Invalidate Completion
- TC=1
- ITagV=1000b
- CC=1

Invalidation Flow – Multi TC

2) Flush matching ATC entries. Drain or discard conflicting Reads



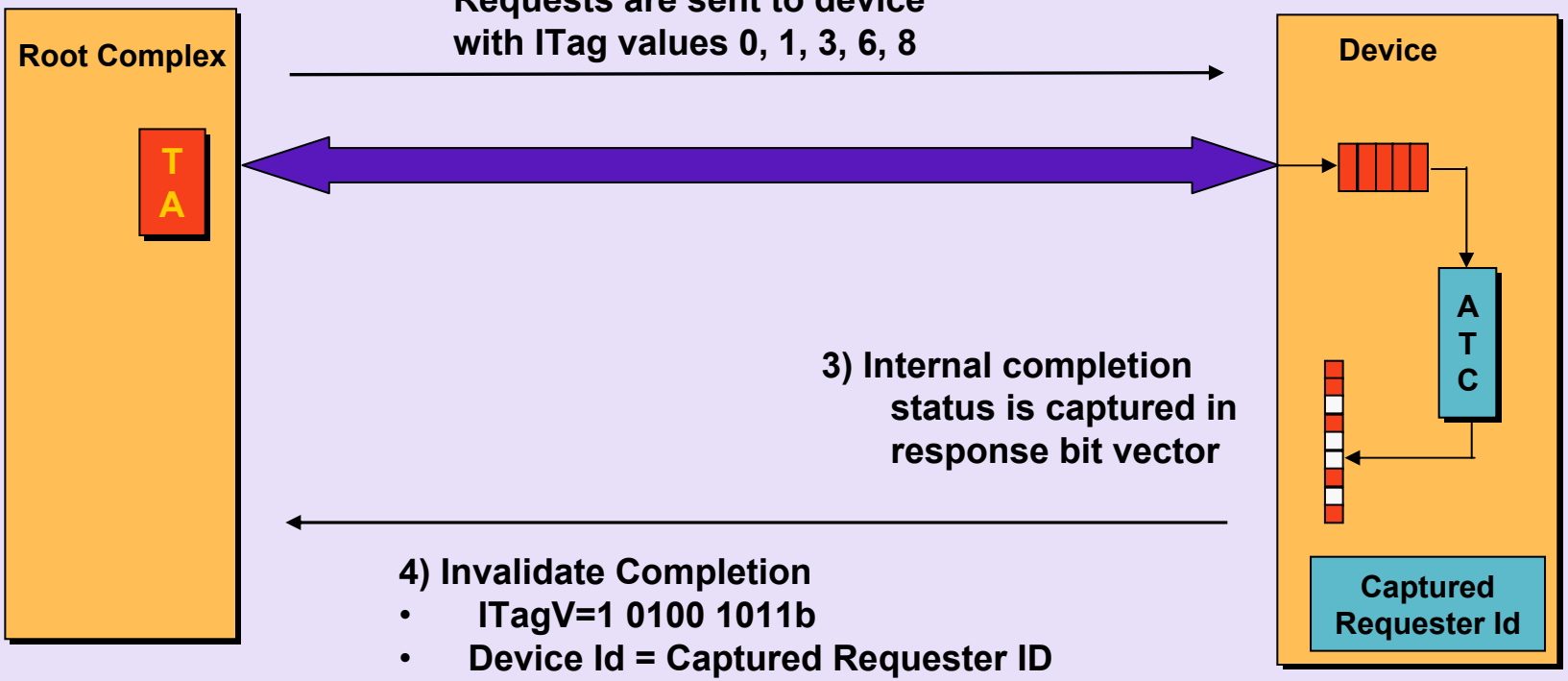
Request Acceptance Rules

- To avoid deadlock:
  - ✓ Endpoints are not allow to create a dependency in which the acceptance of posted transaction is dependent upon the transmission of a posted transaction
- Invalidate Requests and Completions both flow in the posted channel
  - ✓ Could result in deadlock unless…

Request Acceptance Rules (cont)

- ATC must support worst case Invalidate Request queue depth (32 entries)
  - ✓ Use input command queuing
    - Must buffer most of command
  - ✓ Use output response queuing
    - Only requires single bit of state per invalidate
    - Invalidate Completions are collapsible
    - Requester ID of source is captured once and used to route Invalidate Response

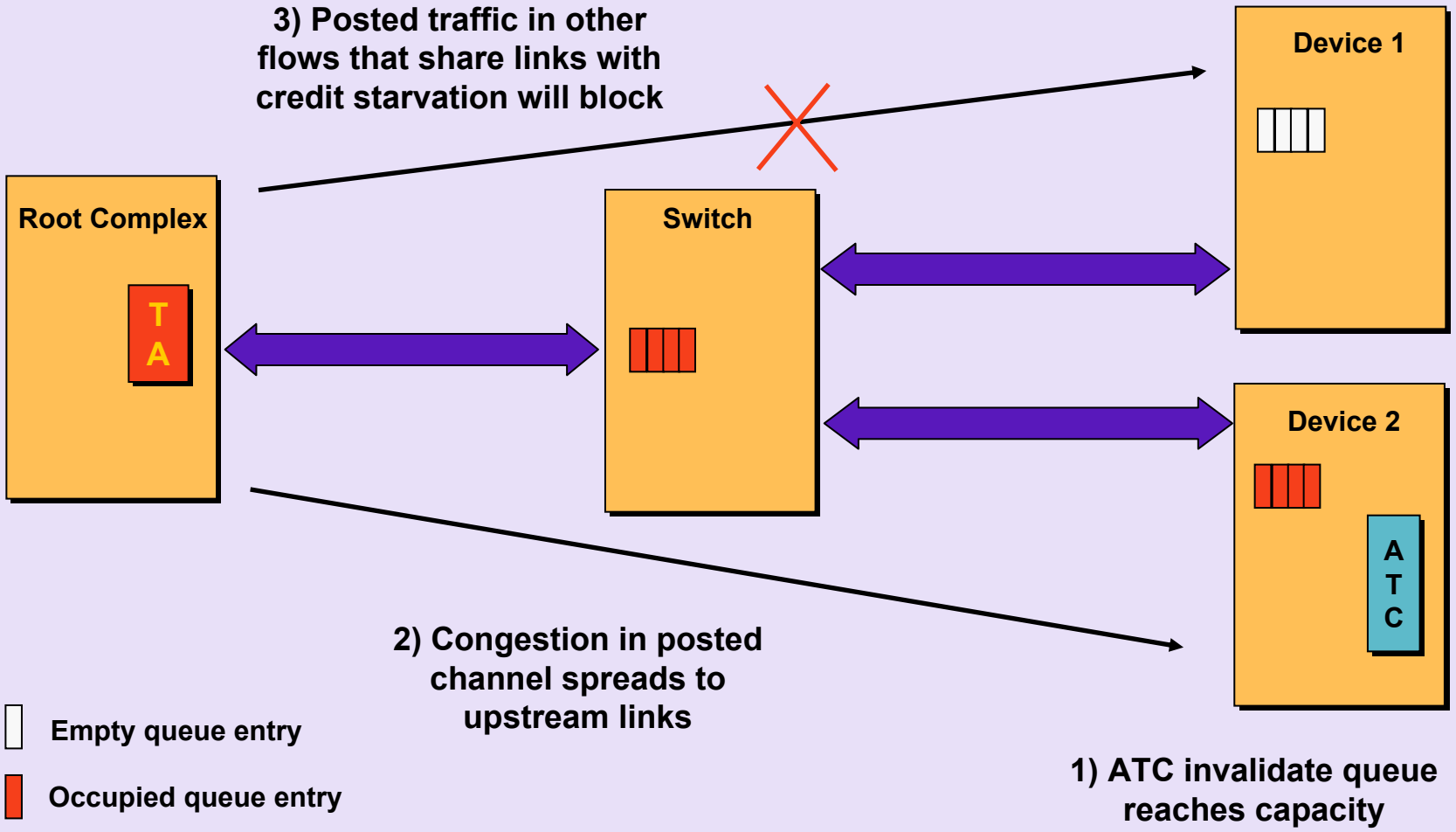
Request Acceptance Rules (cont)



Invalidate Flow Control

- Invalidation time may vary:
  - ✓ Invalidate Requests have variable block size
    - May be larger than cached page size
  - ✓ Dependent upon translation cache architecture
    - Page size
    - Associativity
- Could have negative impact on performance
  - ✓ Posted channel may stall
  - ✓ May effect other I/O flows due to credit based congestion spreading.

Invalidate Flow Control (cont)



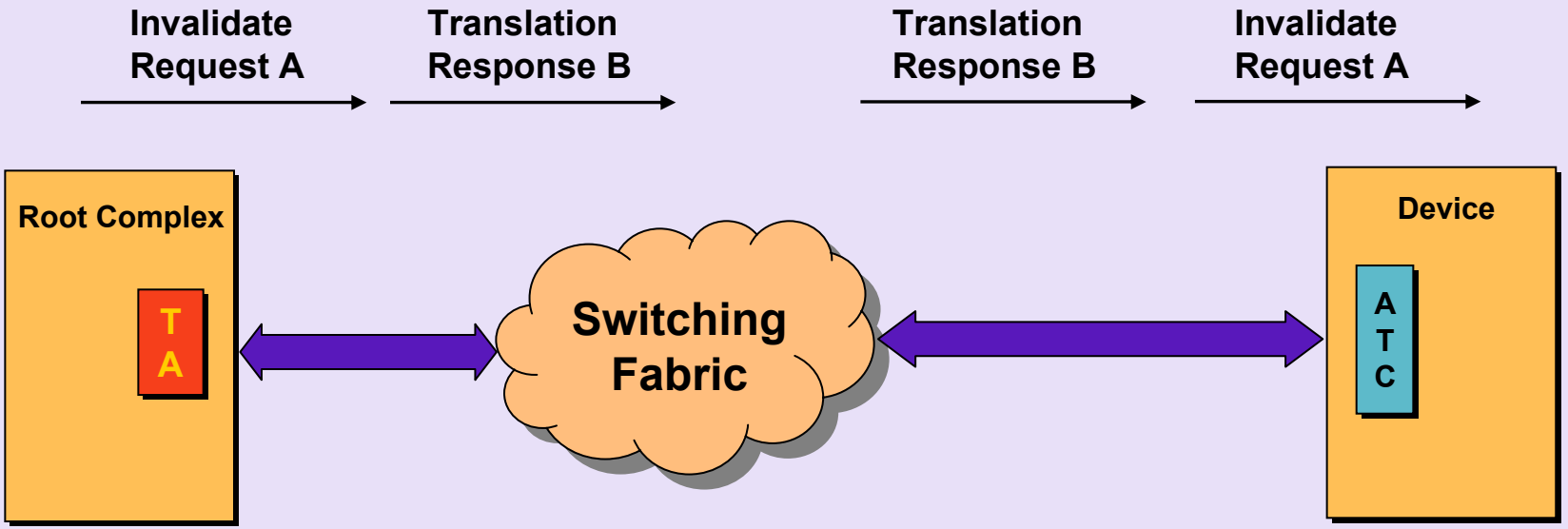
Invalidate Flow Control (cont)

- Enable TA to flow control Invalidate Requests
  - ✓ ATC must publish its Invalidate Queue Depth
- Not required if endpoint will:
  - ✓ Handle invalidations at maximum arrival rate
  - ✓ Rarely cause link backpressure
  - ✓ Fully buffer maximum number of incoming invalidations with out backpressure

Invalidation Ordering Semantics

- Properties
  - ✓ Translation Completion travels in completion channel
  - ✓ Invalidate Request travels in posted channel
  - ✓ Translation Completion and Invalidate Request travel in same direction
- Consequence
  - ✓ Invalidate Request may bypass Translation Completion
  - ✓ Result is a stale address translation may persist in the ATC.

Invalidation Ordering Semantics (cont)



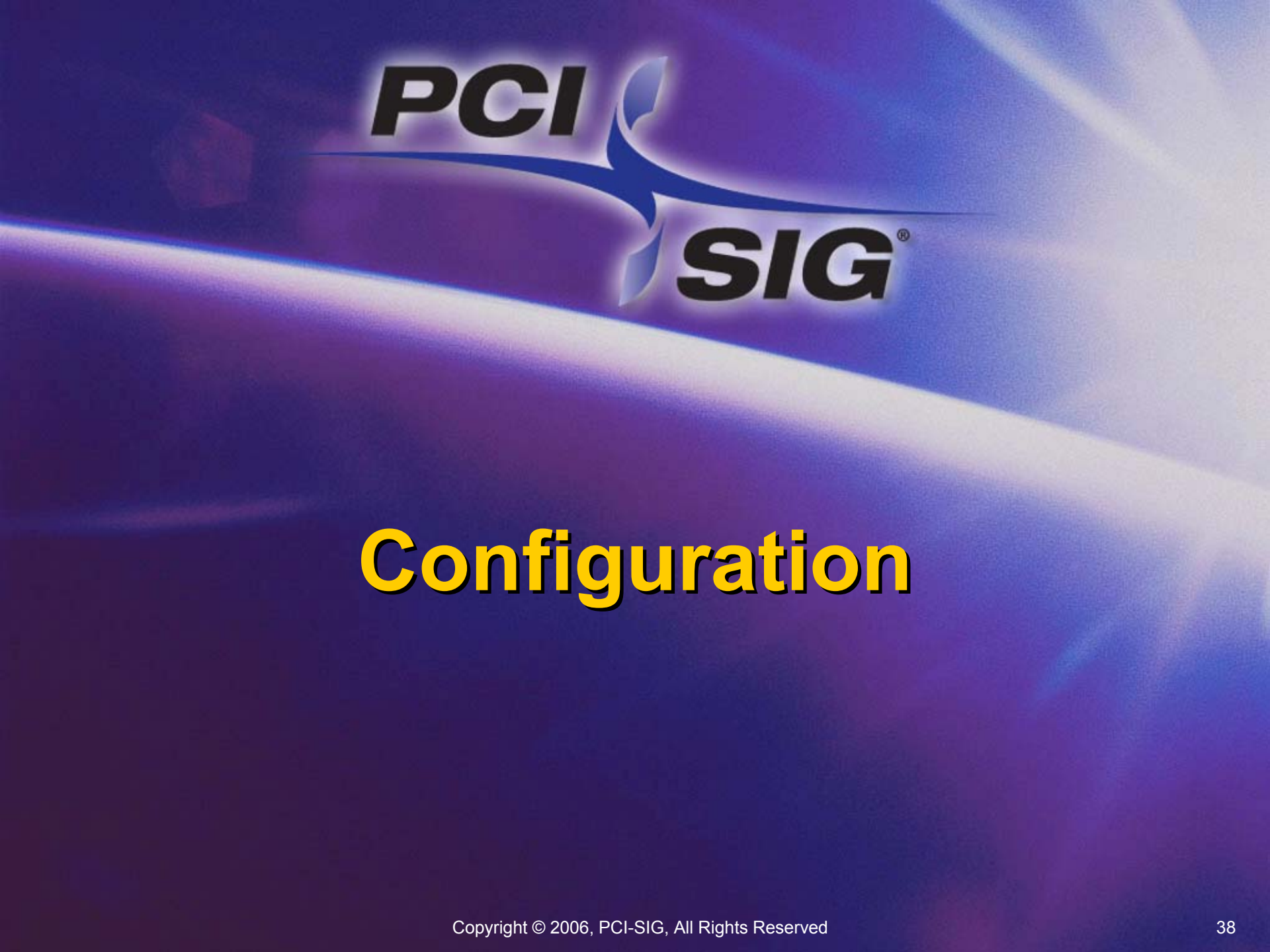
- Invalidate Request and Translation Response correspond to overlapping memory regions
- Invalidate Request passes Translation Response
- Stale address translation gets installed in ATC

Invalidation Ordering Semantics (cont)

- To eliminate stale entries, the ATC must:
  - ✓ Snoop its outstanding Translation Request queue against incoming Invalidate Requests
  - ✓ On hit:
    - Mark Translation Request as invalid
    - Discard results of Translation Response before issuing Invalidate Completion
      - Results of Translation Response may be used in new requests sent prior to transmission of Invalidate Completion

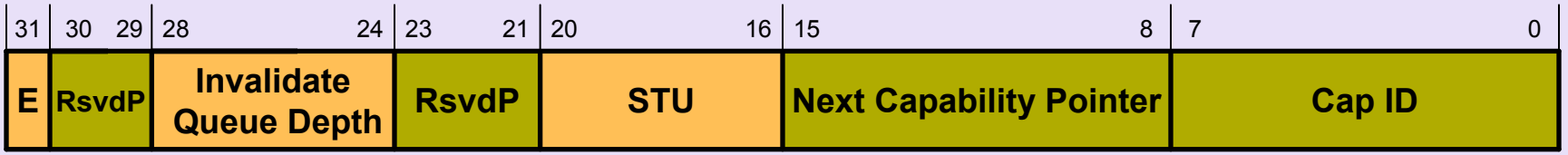
Implicit Invalidation Events

- Invalidations triggered by following events
  - ✓ Fundamental Reset
    - Cold Reset
    - Warm Reset
    - Hot Reset
    - PERST#
  - ✓ Function Level Reset
- Invalidate Complete Response not sent



Configuration

ATS Capability Structure



Bit Location	Register Description	Attributes
20:16	**Smallest Translation Unit (STU):** This value indicates to the Function the minimum number of 4KB blocks that will indicated in a Translation Completion or Invalidate Requests. This is a power of 2 multiplier and the number of blocks is 2^STU. A value of 0 indicates one 4KB block and a value of 1 1111b would indicate an 8TB block. Default value is 0 0000b.	RW
28:24	**Invalidate Queue Depth:** The number of Invalidate Requests that the Endpoint can accept before putting backpressure on the upstream connection. If zero, the Endpoint can accept 32 Invalidate Requests.	RO
31	**Enable (E):** When Set, the Endpoint is enabled to cache translations. Default value is 0b	RW

Questions

Thank you for attending the
PCI-SIG Developers Conference 2006.

For more information please go to
www.pcisig.com

Address Translation Services

Michael Krause (HP, co-chair)

Mark Hummel (AMD)

David Wooten (Microsoft)