Address Translation Services (ATS) Overview

David Mayhew

AMD

Disclaimer

- NOTE: Some information in this presentation refers to specifications still in the development process. This presentation reflects the current thinking of the workgroup, but all material is subject to change before the specifications are released.

Agenda

- ATS 1.0
  - ✓ Translation Request
  - ✓ Translation response
- ATS 1.1
  - ✓ Page Request Interface
    - Page Group Request Messages
    - Page Group Response Messages
  - ✓ Access Rights Reduction

TLAs and Associated Confusions

- ATPT: Address Translation and Protection Table
- ATC: Address Translation Cache
- ATS: Address Translation Services
- BDF: Bus:Device:Function
- DMA: Direct Memory Access
- IOTLB: Input/Output Translation Look-Aside Buffer
- TA: Translation Agent

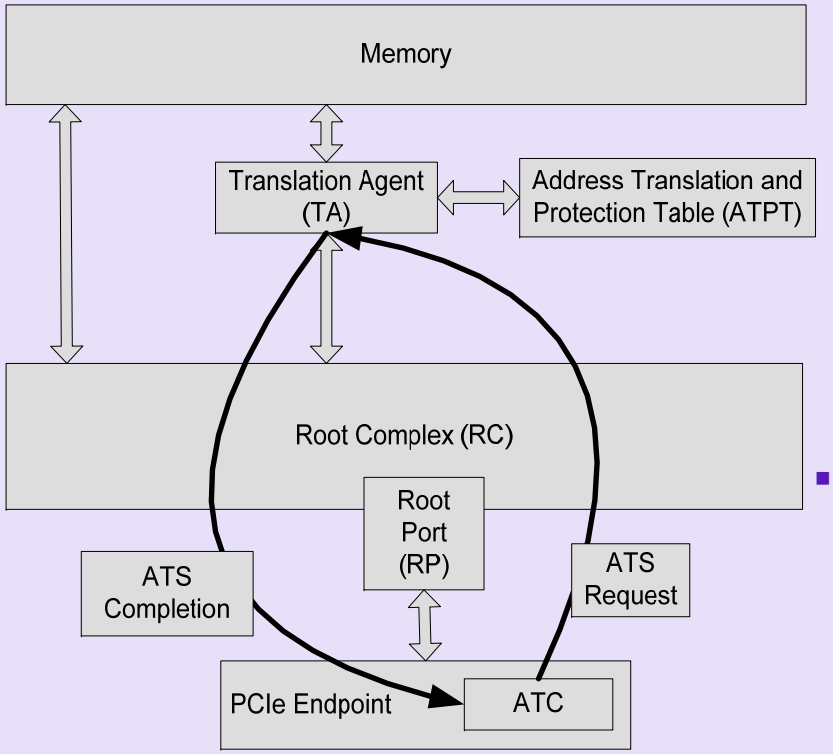
ATS 1.0

- You saw these slides last year
  - ✓ Smoke um if you got um
- Motivation
- Translations
  - ✓ Requests
  - ✓ Responses (Completions)
- Invalidations
  - ✓ Requests
  - ✓ Responses (Completions)

DMA Address Translation

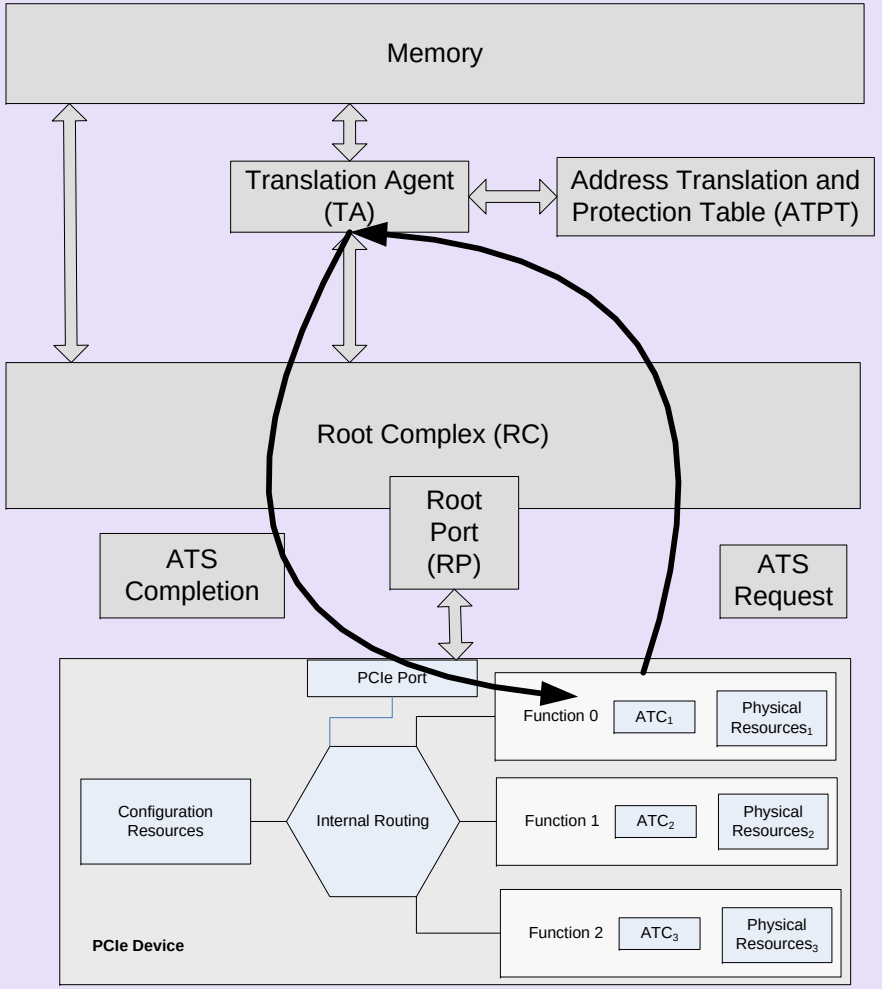
- Translation Agent (TA) performs:
  - ✓ Address translation and an access right validation per Device DMA request
  - ✓ One or more accesses into the ATPT to acquire translation
    - – May need to walk multiple table entries to acquire platform physical address
- Potential Issues with DMA Translation
  - ✓ 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”



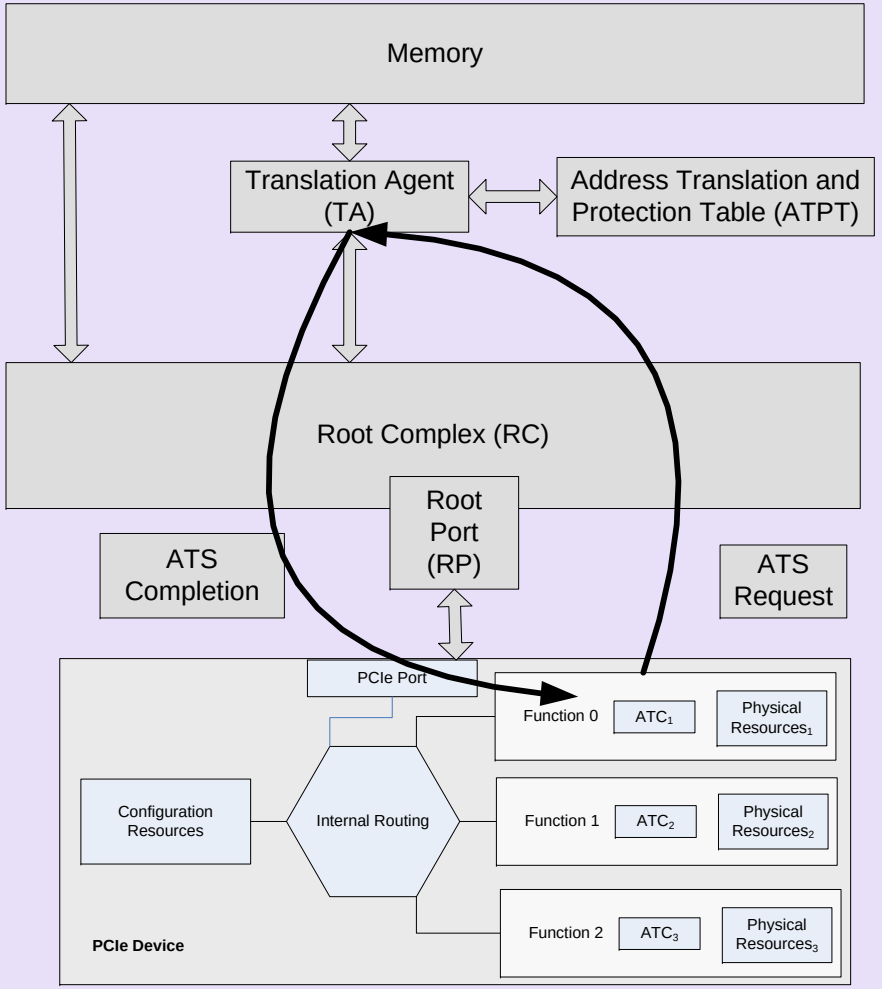
- ATS attempts to mitigate the impact of DMA translation by providing ways for Devices to participate in translation cache management
  - ✓ Device can maintain their own cache of translations – an “Address Translation Cache” (ATC)
  - ✓ TA provides table-walking services to device to avoid excess bus 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

ATS Request



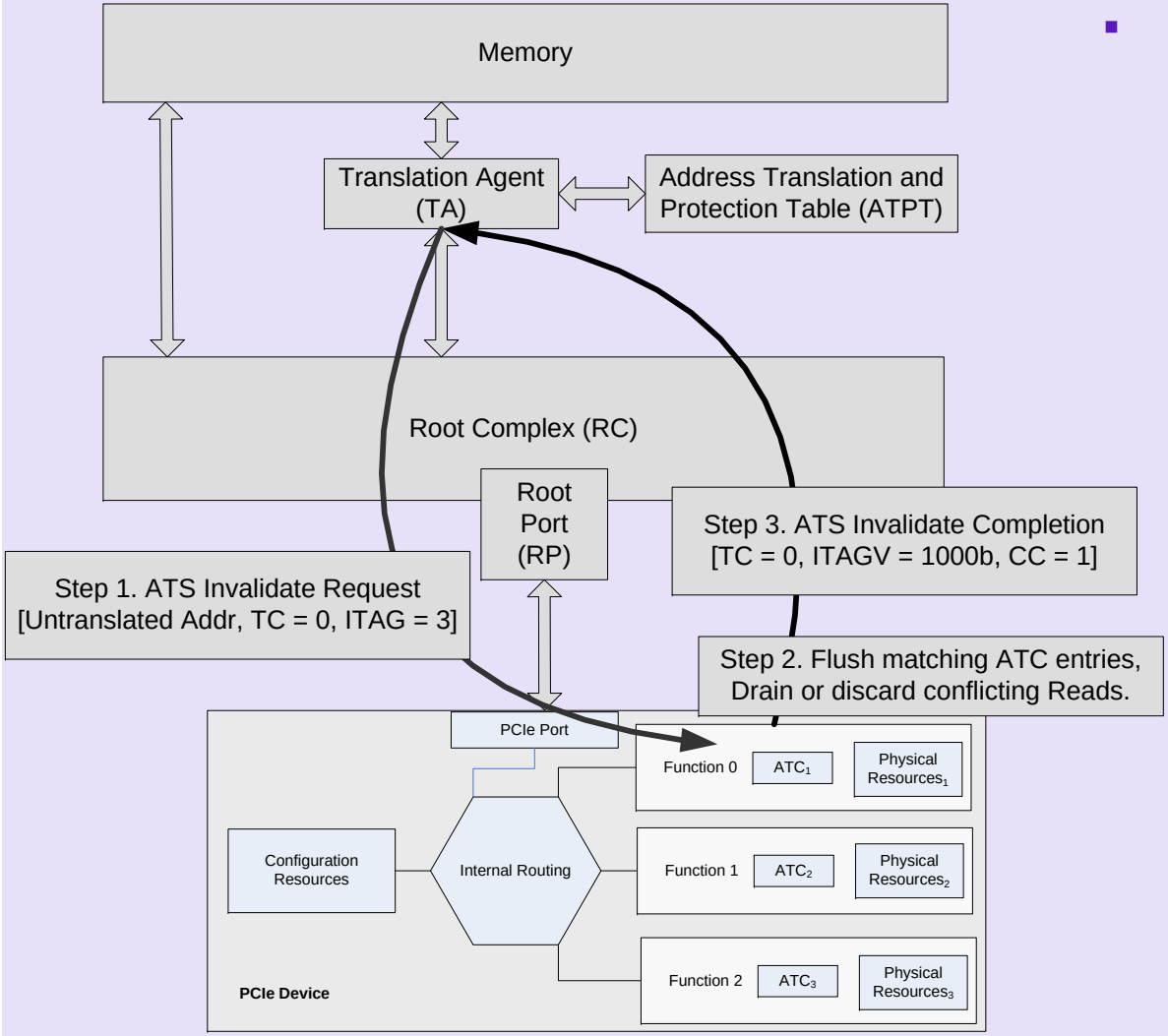
- ATS Request
  - ✓ Used by the Endpoint to request translated address
    - Address can be 32-bit or 64-bit
  - ✓ Requests can flow on any TC
  - ✓ Multiple ATS Requests may be pipelined
- Function Responsibilities
  - ✓ Function implements an ATC
    - ATC is a cache of translated addresses
    - ATC may be implemented independent of IOV support
  - ✓ Function must only populate cache via ATS protocol
  - ✓ Function must not allow ATC to be modified or accessed by software (local or remote) to prevent data leakage
  - ✓ For each ATS Request, there will be a corresponding ATS Completion.

ATS Completion



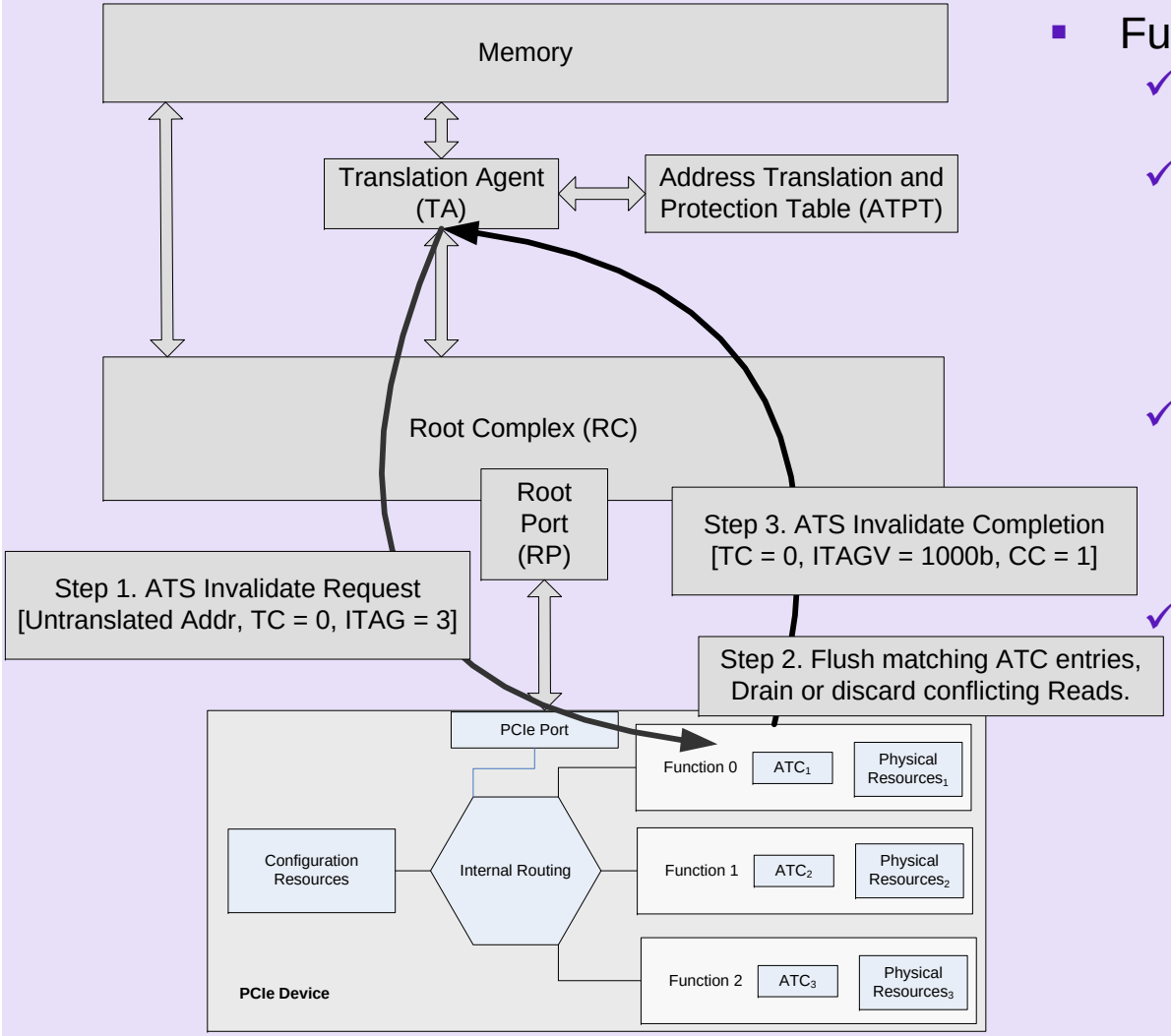
- ATS Completion
  - ✓ Used to return either:
    - A translated address which may cover a range of pages
    - A translation failure
  - ✓ An ATS Completion must be generated for each ATS Request
    - Multiple ATS Completions may be in-flight
    - ATS Completions may be returned in any order relative to the ATS Requests
  - ✓ An ATS Completion must use the same TC as the associated ATS Request
  - ✓ Completions must be sent using the same TC as the ATS Request
  - ✓ ATS Completions return access rights
    - Read-only, Write-only, Read / Write
      - Read = Write = 0 indicates the there is a hole in the translation space
    - A Function can only issue subsequent TLP Requests that match the access rights on the associated address range
- Function responsibilities:
  - ✓ Be capable of sinking all ATS Completions without causing backpressure on the PCIe Link
  - ✓ Be able to discard ATS Completions that are stale
    - If an ATS Invalidation is received prior to the ATS Completion, then the subsequent Completion is stale and should not be used

ATS Invalidation



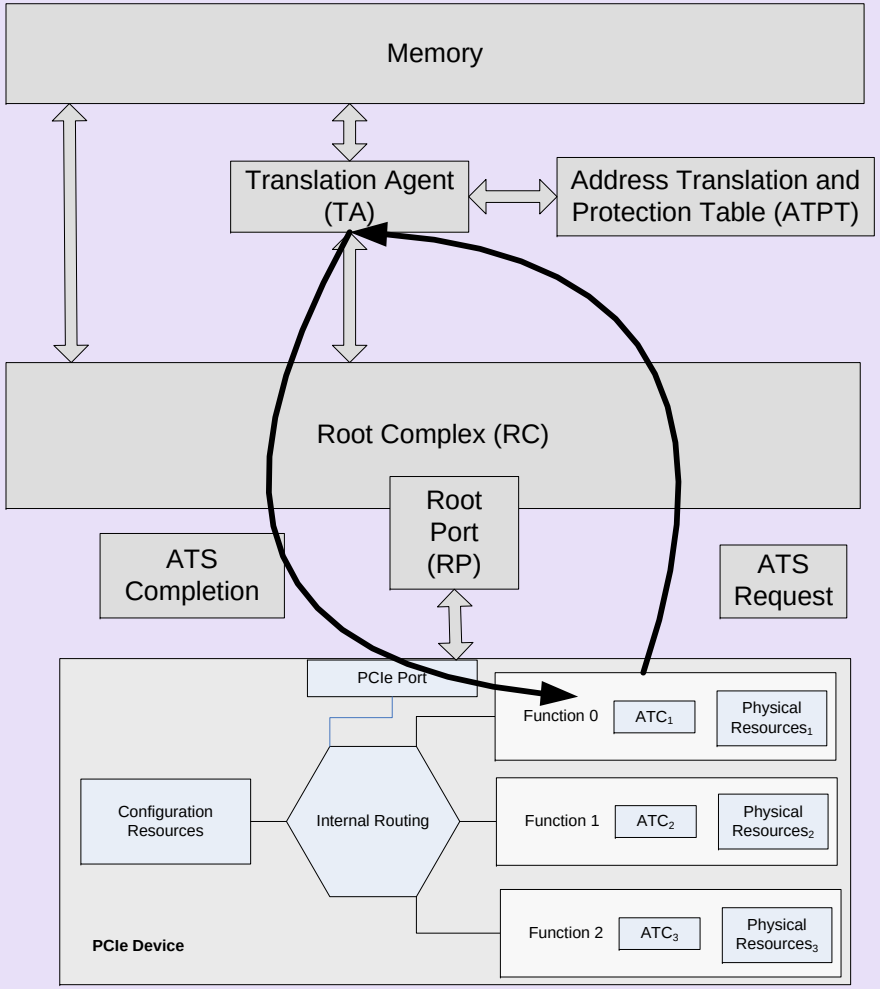
- ATS Invalidation Request
  - ✓ Used to maintain consistency between the ATPT Translation Agent (TA) and the ATC
  - ✓ ATS Invalidation Requests invalidate translations within an ATC
    - A Request may cover a range of address
  - ✓ ATS Invalidation Request may be issued at any time
    - When a translation is changed within the ATPT, the TA issues an ATS Invalidation Request to the impacted ATC
  - ✓ ATS Invalidation Requests may be issued on any TC
    - The TA does not track the TC of prior ATS Request(s)
  - ✓ Each ATS Invalidation Request has an ITAG which uniquely identifies the Request
    - The associated Invalidation Completion returns this ITAG to enable the TA to quickly associate the Completion with the prior Request

ATS Invalidation (cont.)



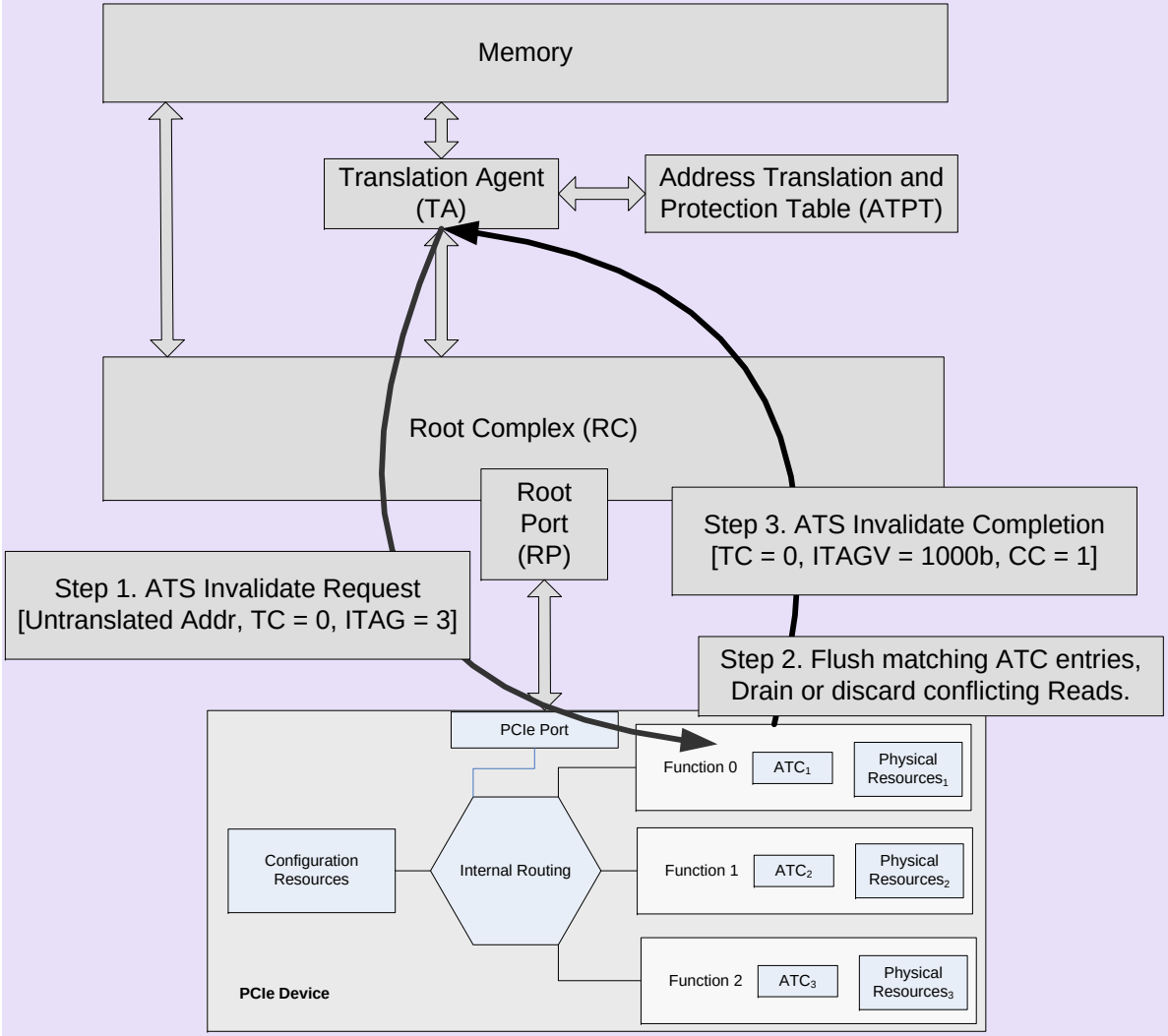
- Function responsibilities:
  - ✓ Must return an ATS Invalidation Completion for each Request
  - ✓ Must not indicate an invalidation has completed until all outstanding Read Requests that reference the associated translated address have been retired.
  - ✓ Must insure that the invalidation completion indication to the RC will arrive at the RC after any previously posted writes that use the 'stale' address.
  - ✓ Is NOT required to immediately flush all pending requests upon receipt of an Invalidate Request. If transactions are in a queue waiting to be sent, it is not necessary for the device to expunge requests from the queue even if those transaction use an address that is being invalidated.

Example ATS Request Flow



1. Endpoint determines address ranges that will benefit from ATS
   1. For example, an Endpoint may want to translate addresses associated with work queues while not for single-use scatter-gather addresses
2. Endpoint issues an ATS Request
3. Translation Agent (TA) examines ATPT to determine if a translation exists
4. If translation exists, issues an ATS Completion with associated translation and access rights
   1. A TA may selectively ignore Requests for a given Function even if a translation exists
5. If translation does not exist, issues an ATS Completion indicating no such translation
   1. Endpoint may still issue DMA TLP but must not set the “Translation bit”
6. Endpoint issues DMA TLP to access associated address range

Example ATS Invalidation Flow



- The ATPT requires a translation update
- TA issues ATS Invalidation Request
- Function may flush or complete all pending transactions associated with the effected address range
- Once the Function has cleaned up all state associated with the effected address range, it returns an ATS Invalidation Completion
- TA processes ATS Invalidation Completion and updates ATPT accordingly

ATS 1.1 Extensions

- Motivations
- Page Request Message
- Page Request Group Response Message
- Access Right Reduction

I/O Page Faulting

- Page Fault == Residence Request
  - ✓ Page may be in memory and just not visible to I/O device.
  - ✓ Page may need to be fetched on disk.
  - ✓ Page may be in gray area between being swapped out and no longer being resident in memory.
- Pinning all I/O accessible memory:
  - ✓ Potentially wasteful of memory resources.
  - ✓ Sometimes difficult to guarantee.
  - ✓ Ignores capabilities of intelligent I/O devices.

Page Request Group

- Device may request and RC will respond to a single page request.
- Device may request group of pages and RC will respond when entire group has been made resident.
- Single page requests correspond to Page Request Groups with a size of 1.
- Failure (inability/refusal by RC to provide residency) of any page within a Page Request Group will result in failure for entire group.
  - ✓ Nothing can be assumed about residence status of any pages in group.

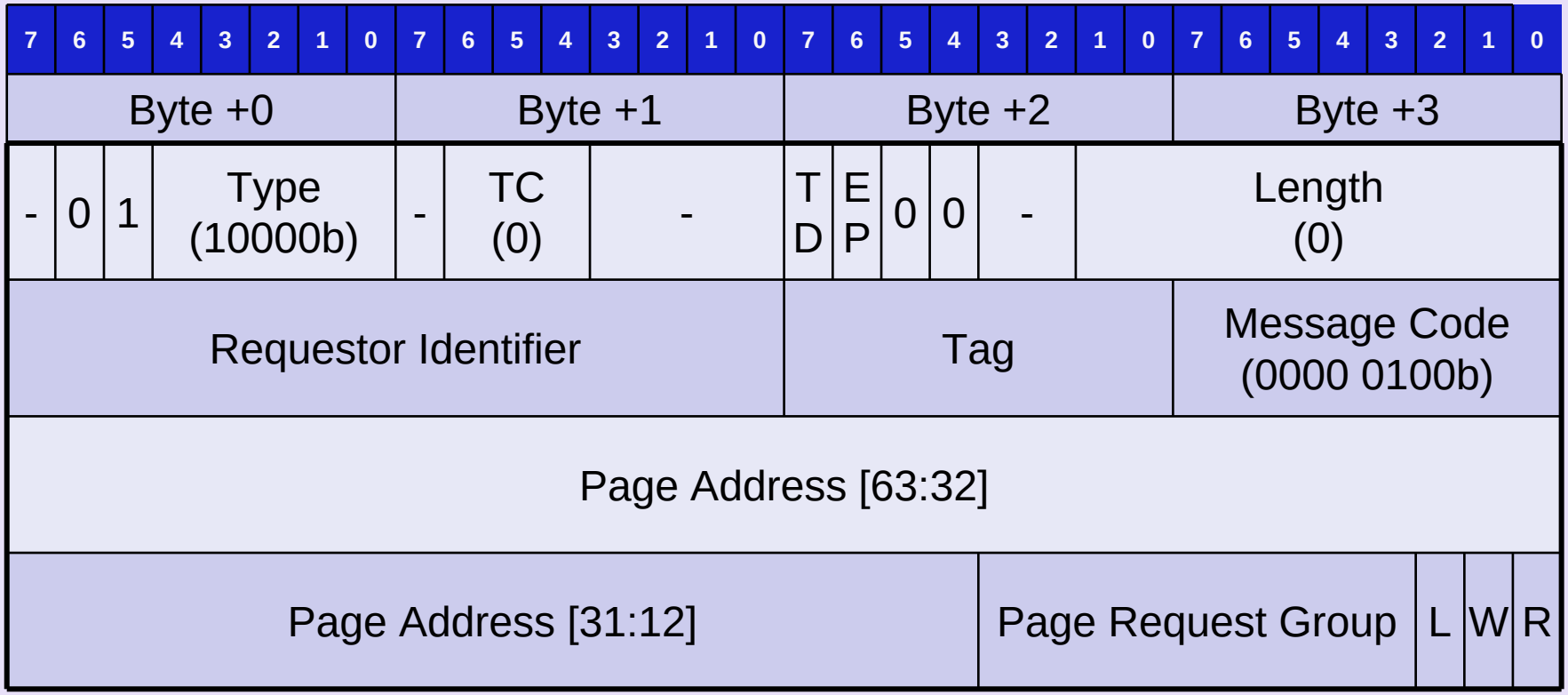
Multi-Page Request Motivation

- 500 Megabyte file needs to be transferred
  - ✓ Fetch 16 megabytes at once and transfer in-order while overlapping fetch and forward operations.
- Data Base inverted index indicates 15 records require update
  - ✓ Fetch all 15 records and update and commit when all 15 records are concurrently available.
- 16 megabyte graphics texture will be required for a future rendering
  - ✓ Fetch all pages in texture, notify when complete
- Aside: Page requests and translation requests are independent – a group of pages can be speculatively requested and translations requested prior to the actual completion of the larger group request.

Page Request Message

- PCIe Message
  - ✓ Not read or write packet
  - ✓ Employs route to root
  - ✓ Travels in TC 0
- Contains:
  - ✓ Address of page to be loaded
    - – Untranslated (virtual) address used
  - ✓ Page Request Group Index
  - ✓ Last flag
    - – Indicates last page in group

Page Request Message

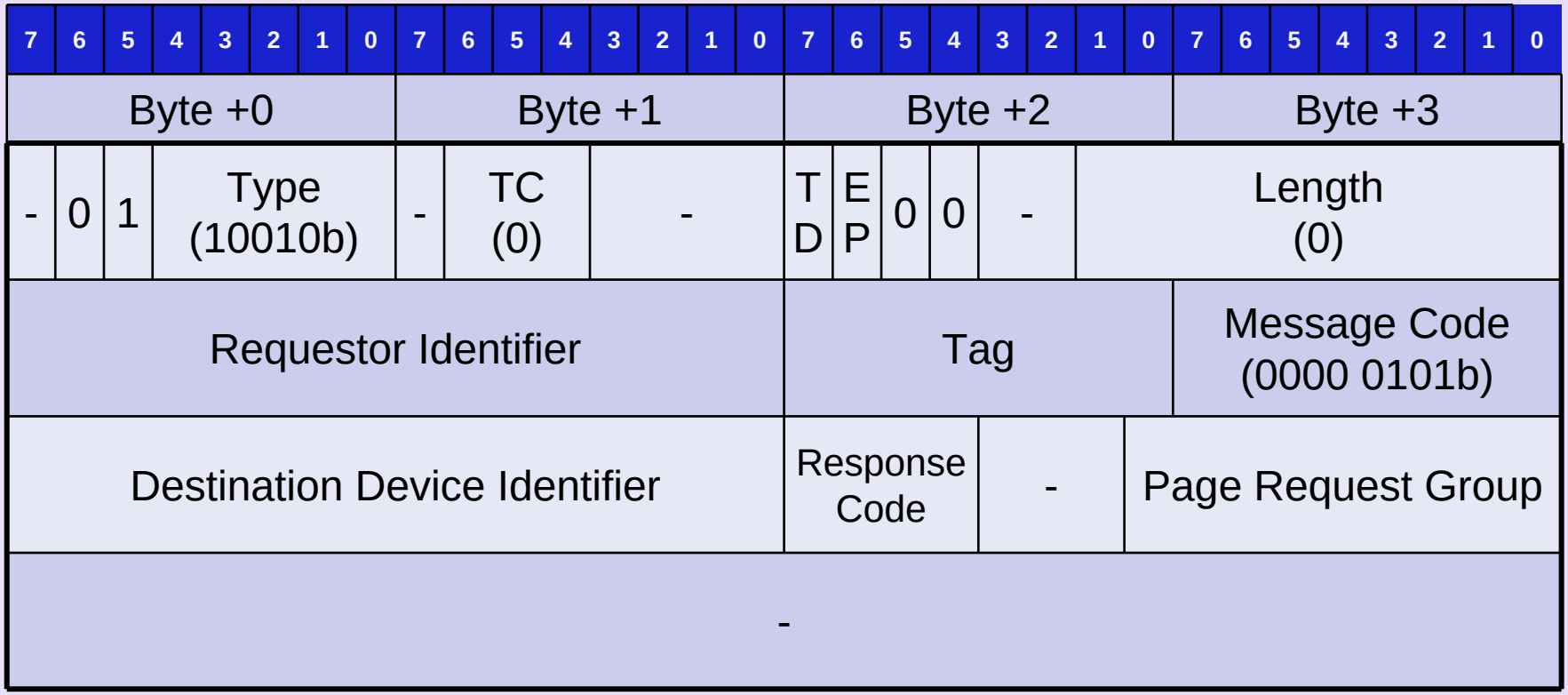


L = Last
W = Write
R = Read

Page Request Group Response Message

- PCIe Message
  - ✓ Not read or write packet
  - ✓ Employs B:D:F routing
  - ✓ Travels in TC 0
- Contains:
  - ✓ Page Request Group Index
  - ✓ Response Code
    - – Success: all pages made resident
    - – Failure
      - Invalid Request: Requested access rights unavailable
      - Response Failure: Catastrophic failure, PRI disabled

Page Request Group Response Message



Page Request / Translation Request Interaction

- Page Request/Response does not imply page pinning.
  - ✓ A page request may be positively acknowledged but not resident.
- Page request does not provide page translation.
  - ✓ Translation must be independently acquired.
- Page translations must be affirmatively withdrawn before page is considered unavailable to device.
  - ✓ ATS translation invalidation flow maintains integrity of memory access model.

Access Right Restriction

- I/O page faults removes necessity of pinning all I/O accessible pages.
- Writable I/O pages must be considered dirty.
  - ✓ I/O device memory operations are transparent to MMU.
- ATS 1.0 Completions automatically grants least restrictive access rights.
  - ✓ Requestor does not indicate usage expectations.
  - ✓ ATS 1.0 results in read-only device accessible pages being marked as dirty.
- ATS 1.1 adds Write Permission Abrogation flag to translation request.
  - ✓ Request translation, but only grant read access.
  - ✓ An independent translation request will be issued if write access is required at some future point

ATS 1.1
Translation Request Header

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
Byte +0								Byte +1								Byte +2								Byte +3							
-	0	1	Type (00000b)					-	TC (0)			-				TD	EP	0	1	-	Length (00 000x xxx0)								
Requestor Identifier																Tag								Last DW BE				First DW BE			
Untranslated Address [63:32]																															
Untranslated Address [31:2]																														-	X

X = Restrict to Read-Only

The ‘X’ Flag

- Least significant bit of dword containing least significant bits of untranslated address allocated to X flag.
- Set X flag indicates that device will not write to associated page without first requesting new, unrestricted translation.
- X-flag reserved in ATS 1.1
  - ✓ Backward compatible
  - ✓ ATS 1.0 ignores access restriction request.
    - – Write access granted independent of restriction request.
- ATS 1.1 capable RCs can restrict or not restrict
  - ✓ Restricted access to a page that is already dirty can be ignored.

The ‘X’ Flag (continued)

- ATS Translation Response contains granted access rights
  - ✓ Independent of requested access rights.
  - ✓ Device must adhere to granted access rights.
    - Same as ATS 1.0
  - ✓ If device has read-only access and wants to write
    - Device should “remember” whether access rights were set by restricted translation request.
      - If yes, then unrestricted translation request may gain write permission.
      - If no, then change in configuration of associated page required to gain access
        - Repeated translation request will return same response as initial request.
    - Not remembering and re-requesting does not break anything.

Conclusion

- ATS 1.0 supports address virtualization between device and root complex.
  - ✓ Address Translation and Completion
  - ✓ Invalidation Request and Completion
- ATS 1.1 extends ATS 1.0
  - ✓ Page Request and Page Request Group Response
  - ✓ Address Translation Write Permission Abrogation
- ATS provides increasing levels of parity between CPU and I/O devices for memory virtualization.

Workgroup Members

- AMD
- Broadcom
- Dolphin
- Emulex
- HP
- IBM
- IDT
- Intel
- LSI
- Microsoft
- NextIO
- Neterion
- Nvidia
- PLX
- Qlogic
- Sun
- VMWare

Thank you for attending the PCI-SIG Developers Conference 2008

For more information please go to
www.pcisig.com