



## PCI-SIG ENGINEERING CHANGE NOTICE

<b>TITLE:</b>	PASID Translation
<b>DATE:</b>	March 31, 2011
<b>AFFECTED DOCUMENT:</b>	Address Translation Services Specification, Version 1.1
<b>SPONSOR:</b>	AMD, HP

### Part I

#### 5 **1. Summary of the Functional Changes**

The Process Address Space ID (PASID) ECN to the Base PCI Express Specification defines the PASID TLP Prefix. This companion ECN is optional normative and defines PASID TLP Prefix usage rules for ATS and PRI.

10 The PASID TLP Prefix is an End-End TLP Prefix. Any routing element that supports End-End TLP Prefixes (i.e. has the End-End TLP Prefix Supported bit Set in the Device Capabilities 2 register) can correctly forward TLPs containing a PASID TLP Prefix.

The PASID TLP Prefix contains a 20 bit PASID value and two request bits (Execute Requested and Privileged Mode Requested). The Base specification defines its usage in Untranslated Memory Requests. This ECN defines its usage in address translation.

15 In addition, three new bits are defined in the ATS Translation Completion Data Entry (Global Mapping, Execute Permitted, Privileged Mode Access), one new bit is defined in the ATS Invalidation Request Message (Global Invalidate), one new bit is defined in the ATS Extended Capability (Global Invalidate Supported), and one new bit is defined in the Page Request Extended Capability (PRG Response PASID Enable).

20 A Stop Marker Message is defined to indicate that Page Request Messages associated with certain PASIDs have stopped. This permits software to reallocate a PASID value and identify Page Requests associated with older uses of a PASID value.

25 The PASID extended capability is defined in the Base Specification and is used to discover PASID capabilities of a Function and to enable PASID operation by the Function. Software discovery of PASID Requester capabilities of a Function is not otherwise architected.

No mechanism is architected to describe the PASID support capabilities of a TA.

Software may enable PASID if (1) the Function supports it, (2) the TA supports it, (3) all Switches in the path support End-End TLP Prefixes and (4) the Function and TA both support adequate End-End TLP Prefix buffering.

30 A Function's level of PASID support is independent of its support of either ATS or PRI. A Function may support PASID even if it supports neither ATS nor PRI. This ECN describes the interactions between PASID and ATS / PRI.

#### **2. Benefits as a Result of the Changes**

35 PASIDs enable sharing of a single Function across multiple processes while providing each process a complete 64 bit virtual address space.

PASIDs enable the hierarchical management of address spaces. In Hypervisor systems, Untranslated Addresses without a PASID TLP Prefix represent Guest Physical Addresses while

Untranslated Addresses with a PASID TLP Prefix represent Guest Virtual Addresses. The Hypervisor manages the Guest Physical Address to System Physical Address mappings while the Guest OS running on top of the Hypervisor manages the Guest Virtual Address to Guest Physical Address mappings.

- 5 The Execute Requested and Exe bits support the labeling, by the TA, of address ranges that contain code that may be executed by a Function. The Privileged Mode Requested and Priv bits support labeling, by the TA, of address range protection attributes associated with two privilege levels of operation within a Function (Privileged and Non-Privileged) <sup>1</sup>. The Global Mapping bit supports the labeling, by the TA, of address mappings that apply to multiple PASIDs.

### 10 **3. Assessment of the Impact**

PASID TLP Prefix support is optional normative, and is applicable to Root Complexes, and components with Endpoint Functions. Routing of TLPs with prefixes is covered by the *PCI Express Base Specification*.

- 15 PASID support does not alter existing behavior. New behavior is defined for requests that contain a PASID TLP Prefix or TLPs that are the result of a request that contained a PASID TLP Prefix.

PASID support may be present with or without either ATS or PRI support.

### **4. Analysis of the Hardware Implications**

Endpoints that contain the PASID Extended Capability may generate Requests that contain the PASID TLP Prefix.

- 20 TAs that support PASID TLP Prefixes may set the Global mapping, Priv, and Exe bits in the Translation Completion Data Entry and may generate ATS Invalidation Requests and PRG Messages that contain PASID TLP Prefixes.

### **5. Analysis of the Software Implications**

- 25 Config Space structure enhancements defined in the *PCI Express Base Specification* enable software to discover a Function's PASID capabilities. A Function is not permitted to issue requests with a PASID TLP Prefix unless software specifically enables it.

Software discovery and enablement of Completer and TA PASID capabilities is outside the scope of this specification.

---

<sup>1</sup> These usually correspond to Supervisor and User mode operation but this correspondence is outside the scope of this specification.

## **Part II**

### **Detailed Description of the change**

Add the following item(s) to the Terms and Acronyms section:

Process Address Space ID (PASID)

The Process Address Space ID, in conjunction with the Requester ID, uniquely identifies the address space associated with a transaction.

Add the new section 1.3:

5     1.3.     Process Address Space ID (PASID)

Certain TLPs can optionally be associated with a Process Address Space ID (PASID). This value is conveyed using the PASID TLP Prefix. The PASID TLP Prefix is defined in the *PCI Express Base Specification*.

The PASID TLP Prefix is permitted on:

- 10      Memory Requests (including Untranslated AtomicOp Requests) with Untranslated Addresses
- Address Translation Requests
- Page Request Messages
- ATS Invalidation Requests
- 15      PRG Response Messages

Usage of the PASID TLP Prefix for Untranslated Memory Requests is defined in the *PCI Express Base Specification*. This specification describes PASID TLP Prefix for the remaining TLPs.

20     When a Request does not have a PASID TLP Prefix, the Untranslated Address represents an address space associated with the Requester ID.

When a Request has a PASID TLP Prefix, the Untranslated Address represents an address space associated with both the Requester ID and the PASID value.

When a Response has a PASID TLP Prefix, the PASID value reflects the address space associated the corresponding Request.

25     Each Function has an independent set of PASID values. The PASID field is 20 bits wide however the effective width is constrained by the lesser of the width supported by the Root Complex (TA) and the width supported by the Function (ATC). Unused upper bits of the PASID value must be 0b.

30     For Endpoints in systems where a Virtual Intermediary (VI) is present, Untranslated Addresses with an associated PASID are typically used to represent Guest Virtual Addresses (GVA) and Untranslated Addresses that are not associated with a PASID represent Guest Physical Addresses (GPA). The TA could be designed so that the VI manages the tables used to perform translations from GPA to Translated Addresses while the individual Guest Operating Systems manage tables used to perform translations from GVA to GPA. When translating an address with an associated PASID, the TA performs both translations and returns the resulting Translated Address (i.e. GVA to GPA followed by GPA to Translated Address).The intermediate GPA value is not visible to the ATC.

35

When an ATC invalidates a cached GPA mapping, it invalidates the GPA mapping and also invalidates all GVA mappings in the ATC. When the GPA invalidate completes, the VI can safely remove pages backing GPA memory range from a Guest Operating System. The VI does not need to know which GVA mappings involved the GPA mapping.

5 Add the new section 2.2.6:

### 2.2.6. PASID TLP Prefix

If a Translation Request has a PASID TLP Prefix, the Untranslated Address Field is an address within the process address space indicated by the PASID field.

10 If a Translation Request has a PASID TLP Prefix with either the Privileged Mode Requested or Execute Requested bit Set, these may be used in constructing the Translation Completion Data Entry.

The PASID Extended Capability indicates whether a Function supports and is enabled to send and receive TLPs with the PASID TLP Prefix.

Section 2.3, modify Figure 2-7 to:

- 15
1. Define new bit “Exe” at byte 3 bit 3.
  2. Define new bit “Priv” at byte 3, bit 4.
  3. Define new bit “Global” at byte 3, bit 5.
  4. Shrink the reserved bits to byte 2, bit 1 through byte 3, bit 6.

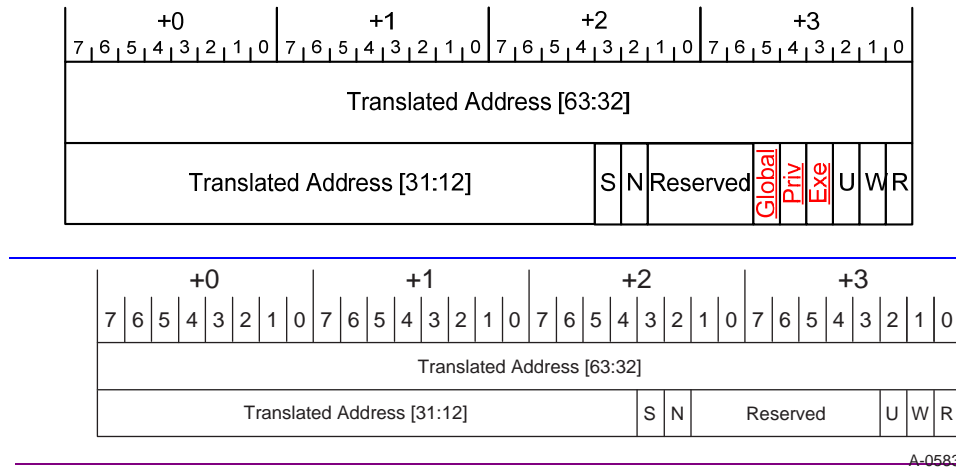


Figure 2-7: Translation Completion Data Entry

Add the following items to Table 2-3:

Field	Meaning
Reserved	These bits shall be ignored by the ATC.
<u>Global</u>	<u>Global Mapping</u> – If this bit is Set, the ATC is permitted to cache this mapping entry in all PASIDs. If Clear, the ATC is permitted to cache this mapping entry only in the PASID associated with the requesting PASID. This bit may only be Set if the associated Translation Request had a PASID TLP Prefix.

Exe	<p><b>Execute Permitted</b> – If this bit is Set, the requesting Function is permitted to execute code contained in the associated memory range.</p> <p><u>This bit may be Set only if the associated Translation Request had a PASID TLP Prefix with the Execute Requested field Set. If this bit is Set, R must also be Set.</u></p> <p><u>The Priv bit indicates the Privilege level associated with the Exe bit. If Priv is Set, the Exe bit indicates permissions associated with Privileged Mode entities in the Function. If Priv is Clear, the Exe bit indicates permissions associated with Non-Privileged Mode entities in the Function.</u></p> <p><u>This value may be cached if R is Set.</u></p>
Priv	<p><b>Privileged Mode Access</b> – If this bit is Set, R, W and Exe refer to permissions associated with Privileged Mode entities. If this bit is Clear, R, W and Exe refer to permissions associated with Non-Privileged Mode entities.</p> <p><u>This bit may only be Set if the associated Translation Request contained a PASID TLP Prefix with the Privileged Mode Requested bit Set.</u></p> <p><u>This value must be cached any of the R, W or Exe values are cached.</u></p>
U	<p><b>Untranslated access only</b> – When this field is Set <del>to 1b in a Translation Completion entry</del>, the indicated range may only be accessed using untranslated addresses, and the Translated Address field of this Translation Completion <a href="#">Data Entry</a> 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.</p>
R,W	<p><b>Read, Write</b> – These two fields indicate the transaction types that are allowed for <del>the</del> requests using the translation. The encodings are:</p> <ul style="list-style-type: none"> <li>00b Neither read nor write transactions are allowed. This translation is considered not to be valid. The contents of the Translated Address, N, <del>and</del> U, <del>and</del> Exe fields are undefined. A translation with this value may not be cached in the ATC.</li> <li>01b Write Requests that target this range are allowed, but Read Requests are not unless they are zero-length reads.</li> <li>10b Read Requests that target this range are allowed (<u>including zero-length reads</u>), but Write Requests are not.</li> <li>11b Read and Write Requests that target this range are allowed.</li> </ul> <p><u>The Priv bit indicates the Privilege level associated with R and W. If Priv is Set, R and W indicate permissions associated with Privileged Mode entities in the Function. If Priv is Clear, R and W indicate permissions associated with Non-Privileged Mode entities in the Function.</u></p>

Add the following paragraph to the end of Section 2.3.3:

#### 2.3.4. Non-snooped (N) Field

...

- 5 [A translation has a single value for the N field that is not affected by privilege level. An ATC is permitted to cache the N field without regard to the value of the Priv bit.](#)

Add the following paragraph to the end of Section 2.3.4:

### 2.3.4. Untranslated Access Only (U) Field

...

Editor: This paragraph is indented to line up under the existing Note:

5 | Note: When U is Set and either R or W is Set, the ATC is permitted to cache U, R, W Exe, and Priv, as well ~~the~~as the Translation Range Size (see Section 2.3.2). An Invalidation Request is required if these values change.

Add the following paragraph to the end of Section 2.3.5:

### 2.3.5. Read (R) and Write (W) Fields

10 | ...

15 | The Priv bit is used to qualify R and W. If Priv is Set, R and W indicate permissions granted to Privileged Mode entities in the Function. If Priv is Clear, R and W indicate permissions granted to Non-Privileged Mode entities in the Function. The R and W values for the two privilege levels are independent. The ATC must not assume any correlation between the Privileged Mode and Non-Privileged Mode permissions associated with a translation.

Add new sections 2.3.6, 2.3.7, and 2.3.8 as follows.

### 2.3.6. Execute Permitted (Exe)

20 | If Exe is Set, the requesting Function is permitted to execute code in the implied range of memory. If Exe is Clear, the requesting Function is not permitted to execute code in the implied range of memory.

The definition of what it means for a Function to execute code is outside the scope of this specification. Various system components may have different instruction sets. Behavior within the requesting Function when it attempts to execute code that is not permitted by this bit is outside the scope of this specification.

25 | The Exe bit may only be Set the TA supports Execute permissions, the associated Translation Request had a PASID TLP Prefix with an effective value of 1b for the Execute Requested bit<sup>2</sup> and R is Set in the Translation Completion Data Entry. Otherwise, the Exe bit must be Clear.

This value may be cached if R is Set.

30 | The Priv bit is used to qualify the Exe bit. If Priv is Set, the Exe bit indicates permission granted to Privileged Mode entities in the Function. If Priv is Clear, the Exe bit indicates permission granted to Non-Privileged Mode entities in the Function. The Exe bit values for the two privilege levels are independent. The ATC must not assume any correlation between the Privileged Mode and Non-Privileged Mode permissions associated with a translation.

35 | Functions may optionally check that:

---

<sup>2</sup> The effective value of the bit is 0b unless the bit in the PASID TLP Prefix is 1b and usage of the bit is enabled for the request (see the *PCI Express Base Specification*).

If the Execute Requested bit is Clear in a Translation Request, the Exe bits in the associated Translation Completion Data Entries are also Clear.

If Exe is Set, R is also Set.

If either optional check fails, the Function shall signal Unexpected Completion (UC). These checks are independently optional.

### 2.3.7. Privileged Mode Access (Priv)

If Priv is Set, R, W, and Exe refer to permissions granted to entities operating in Privileged Mode in the requesting Function. If Priv is Clear, R, W, and Exe refer to permissions granted to entities operating in Non-Privileged Mode in the requesting Function.

The meaning of Privileged Mode and Non-Privileged Mode and what it means for an entity to be operating as in Privileged Mode or in Non-Privileged Mode depends on the protection model of the system and is outside the scope of this specification.

Behavior is outside the scope of this specification when an entity in the requesting Function attempts to access memory that it is not permitted to access.

The Priv bit may only be Set if the TA supports Privileged Mode and the associated Translation Request had a PASID TLP Prefix with an effective value of 1b for the Privileged Mode Requested<sup>3</sup>. Otherwise, the Priv bit must be Clear.

The Privileged and Non-Privileged Mode versions of R, W and Exe are independent. An ATC may cache either or both versions of R, W and Exe. An ATC that receives a translation with R=W=0b for one privilege level may not assume anything about what it might receive for the other privilege level.

This value may be cached if R or W is Set. This value must be cached when the corresponding R, W, or Exe values are cached.

Note: Since the Priv bit is Set only when the requesting Function Sets Privileged Mode Requested, Functions that never set that bit should always receive the Priv bit Clear and thus don't need to cache it.

Functions may optionally check that when the Privileged Mode Requested bit is Clear in a Translation Request, the Priv bits in the associated Translation Completion Data Entries are also Clear. If this optional check fails, the Function shall signal Unexpected Completion (UC).



## IMPLEMENTATION NOTE

### Execute Permission and Privilege Mode Enforcement

The requesting Function determines whether a particular Memory Request needs Execute permission or is associated with a Privileged Mode or Non-Privileged Mode entity. The ATC implements the protection checks indicated by the Exe and Priv bits.

---

<sup>3</sup> The effective value of the bit is 0b unless the bit in the prefix is 1b and usage of the bit is enabled for the request (see the *PCI Express Base Specification*).

## 2.3.8. Global Mapping (Global)

If Global is Set, the requesting Function is permitted to create a Global Mapping entry in the ATC for this translation. If Global is Clear, the requesting Function is not permitted to create a Global Mapping entry in the ATC for this translation. Global Mapping entries apply to all PASIDs of the Function. They permit the ATC to reduce the number of translation requests needed and to reduce the memory needed for caching the results.

A Function is permitted to ignore this bit and always create non-Global Mapping entries in the ATC. This could request multiple translations being requested for the same Untranslated Address under different PASIDs.

Functions that use this bit must also have the Global Invalidate Supported bit Set (see Section 5.3.2).

Modify the name of Section 3 as follows:

### 3. ATS Invalidation

#### Section 3.1

1. Modify Figure 3-2 to new "**Global Invalidate**" bit
2. Add new paragraph at the end of the section

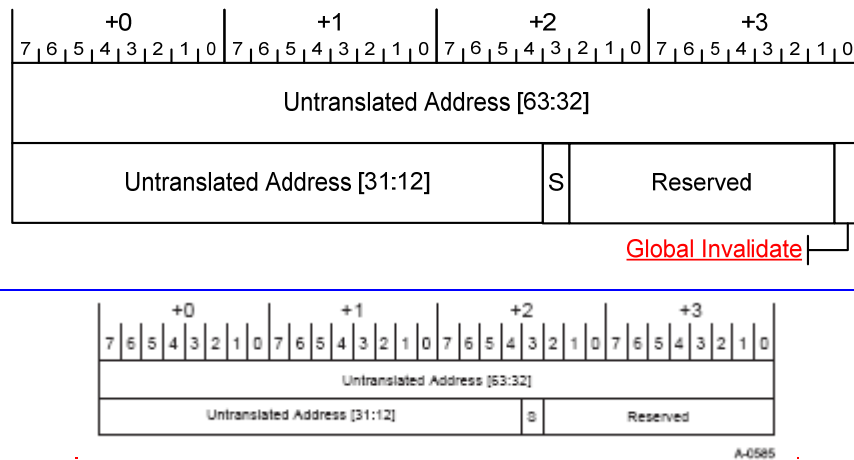


Figure 3-2: Invalidate Request Message Body

The S field is used to indicate if the range being invalidated is greater than 4096 bytes. Its meaning is the same as for the Translation Completion (see Section 2.3.1 and Section 2.3.2).

The Global Invalidate bit indicates that the Invalidation Request Message affects all PASID values (see Section 3.8). This bit is Reserved unless the Invalidation Request has a PASID TLP Prefix. The bit is ignored by the ATC if Global Invalidate Supported bits is Clear (see Section 3.8).

Add the new section 3.8:

### 3.8. PASID TLP Prefix and Global Invalidate

The requirements in this section apply to Functions that support the PASID TLP Prefix.

For Invalidation Requests that have a PASID TLP Prefix, the ATC shall:



Optionally signal Unsupported Request (UR) if the associated PASID value is greater than or equal to  $2^{\text{Max PASID Width}}$ . This error may be signalled anytime an out of range PASID value is present, even when the PASID value is ignored (see below).

Return an Invalidation Completion if PASID Enable is Clear.

5  If the Function supports Global Invalidate (see Section 5.1.2):

If the Global Invalidate bit in the Request is Set, invalidate Global and non-Global Mapping entries in the ATC within the indicated memory range associated with any PASID value and return an Invalidation Completion. The PASID value in the PASID TLP Prefix is ignored.

10  If the Global Invalidate bit in the Request is Clear, invalidate only non-Global Mapping entries in the ATC within the indicated memory range that were requested using the associated PASID value and return an Invalidation Completion.

Global Mapping entries in the ATC for some or all of the indicated memory range may be retained.

15  If the Function does not support Global Invalidate (see Section 5.1.2), invalidate entries in the ATC within the indicated memory range that were requested using the associated PASID value and return an Invalidation Completion.

If no matching entries are present in the ATC, invalidate no ATC entries and return an Invalidation Completion.

20 For Invalidation Requests that do not have a PASID TLP Prefix, the ATC shall:

Invalidate ATC entries within the indicate memory range that were requested without a PASID value.

Invalidate ATC entries at all addresses that were requested with any PASID value.

25 *Add Implementation Note at the end of Section 3.7:*

### 3.7. Implicit Invalidation Events

The following events will cause the invalidation of all ATC entries:

Conventional Reset (all forms)

Function Level Reset

30  E field in ATS Capability changes from Clear to Set

The following events will cause the invalidation of all non-Global Mapping ATC entries that were requested using a specific PASID:

Stopping the use of a PASID as defined in the *PCI Express Base Specification*.

35 No explicit Invalidate Completion message is sent when these implied invalidate events occur.



## IMPLEMENTATION NOTE

### Implicit Invalidation and PASID

Software may not change any of the PASID enable bits when the E field in the ATS Capability is Set. The invalidation that occurs when software Sets the E field also invalidates ATC entries with an associated PASID value.

5 *Modify Section 4.1 Table 4-1 as follows:*

#### 4.1. Page Request Messages

...

**Table 4-1: Page Request Message Data Fields**

Field	Meaning
R	<b>Read Access Requested</b> – This field, when Set, indicates that the requesting Function seeks read access to the associated page. When Clear, this field indicates that the requesting Function will not read the associated page. <u>The R field must be Set for Page Requests with a PASID TLP Prefix that has the Execute Requested bit Set.</u> <u>If R and W are both Clear and L is Set, this is a Stop Marker (see Section 4.1.2.1).</u>
W	<b>Write Access Requested</b> – This field, when Set, indicates that the requesting Function seeks write access and/or zero-length read access to the associated page. When Clear, this field indicates that the requesting Function will not write to the associated page. <u>If R and W are both Clear and L is Set, this is a Stop Marker (see Section 4.1.2.1).</u>
L	<b>Last Request in PRG</b> – This field, when Set, indicates that the associated page request is the last request of the associated PRG. A PRG can have a single entry, in which case the PRG consists of a single request in which this field is Set. When Clear, this field indicates that additional page requests will be posted using this record's PRG Index. <u>If R and W are both Clear and L is Set, this is a Stop Marker (see Section 4.1.2.1).</u>
...	...

10 *Add the new section 4.1.1:*

#### 4.1.1. PASID TLP Prefix Usage

The PASID Extended Capability indicates whether a Function supports PASID TLP Prefixes and whether it is enabled to send and receive them.

15 Functions that support the PASID TLP Prefix are permitted to send a PASID TLP Prefix on Page Request Messages. The PASID field contains the process address space of the page being requested and the Execute Requested and Privileged Mode Requested bits indicate the access being requested.

20 If one Page Request Message in a PRG has a PASID TLP Prefix, all Page Request Messages in that PRG must contain identical PASID TLP Prefixes. Behavior is undefined when the PASID TLP Prefixes are inconsistent.

5 Functions that support the PASID TLP Prefix and have the PRG Response PASID Required bit Set (see Section 5.1.3), expect that PRG Response Messages will contain a PASID TLP Prefix if the associated Page Request Message had a PASID TLP Prefix. For such PRG Response Messages, the Execute Requested and Privileged Mode Requested bits are reserved and the PASID field contains the PASID from the associated Page Request Message.

*Add the new section 4.1.2:*

#### 4.1.2. Managing PASID TLP Prefix Usage

10 The *PCI Express Base Specification* defines rules for stopping and starting the use of a PASID. This section describes additional rules that apply to Functions that have issued Page Request Messages in a PASID that is being stopped. No additional rules are required to start the usage of the Page Request Interface for a PASID.

15 When stopping the use of a particular PASID, a Stop Marker Message may be optionally used to avoid waiting for PRG Response Messages before the Function indicates that the stop request for a particular PASID has completed.

To stop without using a Stop Marker Message, the Function shall:

1. Stop queueing new Page Request Messages for this PASID.
2. Finish transmitting any multi-page Page Request Messages for this PASID (i.e. send the Page Request Message with the L bit Set).
- 20 3. Wait for PRG Response Messages associated any outstanding Page Request Messages for the PASID.
4. Indicate that the PASID has stopped using a device specific mechanism. This mechanism must indicate that a Stop Marker Message will not be generated.

To stop with the use of a Stop Marker Message the Function shall:

- 25 1. Stop queueing new Page Request Messages for this PASID.
2. Finish transmitting any multi-page Page Request Messages for this PASID (i.e. send the Page Request Message with the L bit Set).
3. Internally mark all outstanding Page Request Messages for this PASID as stale. PRG Response Messages associated with these requests will return Page Request Allocation credits and PRG Index values but are otherwise ignored.<sup>4</sup>
- 30 4. Indicate that the PASID has stopped using a device specific mechanism. This mechanism must indicate that a Stop Marker Message will be generated.
5. Send a Stop Marker Message to indicate to the host that all subsequent Page Request Messages for this PASID are for a new use of the PASID value.

35 Note: Steps 4 and 5 may be performed in either order, or in parallel.

---

<sup>4</sup> Page Request Allocation is shared across all PASIDs of the Function (see Section 5.2.5). If PRG Response PASID Required is Clear, PRG Index values are shared across all PASIDs of the Function (see Section 5.1.3).

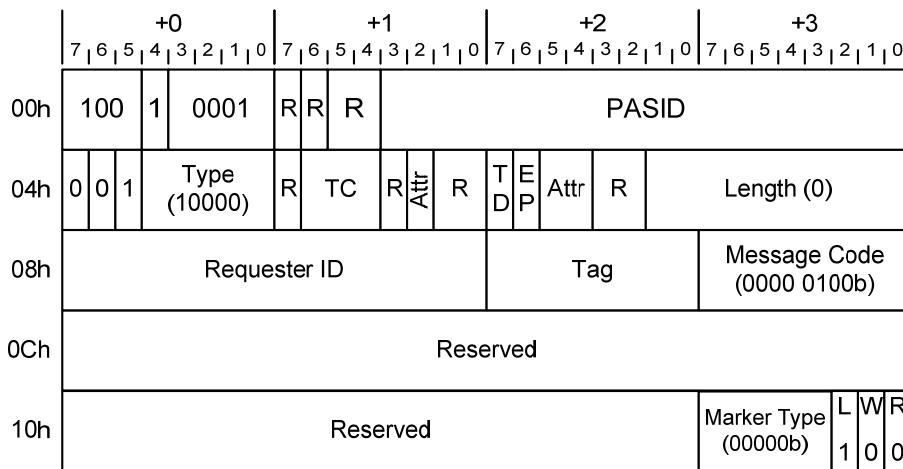
## 4.1.2.1. Stop Marker Messages

A Stop Marker Message indicates that a Function has stopped using the Page Request Interface and has transmitted all pending Page Request Messages for a specific PASID. Stop Marker Messages are strongly ordered with respect to Page Request Messages and serve to push Page Request Messages toward the Host. When the Host receives the Stop Marker Message, this indicated that all Page Request Messages associated with the PASID being stopped have been delivered and that any subsequent Page Request Message with the same PASID value are associated with a new incarnation of that PASID value.

Stop Marker Messages do not have a response. They do not have a PRG Index and do not consume Page Request allocation (see Section 5.2.5).

The Stop Marker Message bit layout is shown in Figure 4-1a.

*Note to Editor: Adjust Figure numbering when incorporating this change.*



**Figure 4-1a: Stop Marker Message**

A Stop Marker Message is encoded as a Page Request Message that contains a PASID TLP Prefix with the following exceptions:

- The L, W and R fields contain 1b, 0b and 0b respectively.
- The Untranslated Address field and upper bits of the PRG Index field are Reserved.
- The Marker Type field contains 00000b to indicate that this is a Stop Marker Message.
- The Execute Requested and Privileged Mode Requested bits in the PASID TLP Prefix are Reserved.
- The Traffic Class must be 0.
- The Relaxed Ordering attribute bit must be Clear.
- The ID Based Ordering attribute bit may be Set as specified in the *PCI Express Base Specification*.

Behavior is undefined if a Stop Marker Message is received and any of the following are true:

- Marker Type not equal to 00000b.

- No PASID TLP Prefix is present.
- The PASID value does not match an outstanding stop request.
- An incomplete Page Request Message for the PASID is outstanding (i.e. for some PRG Index, the most recently received Page Request Message did not have the L bit Set).

Modify Section 4.2 as follows:

...

There are ~~three~~ four possible ~~p~~Page ~~r~~Request failures:

1. The requested page is not a valid ~~u~~Untranslated ~~a~~Address ~~or both R and W fields are clear in the page request.~~
2. PASID TLP Prefix support exists, the Page Request has a PASID TLP Prefix, and either PASID TLP Prefix usage is not enabled for this request, the PASID value is not valid, or Execute Requested is Set when R is Clear.<sup>5</sup>
- ~~2.~~ The requested page does not have the requested access attributes (including Execute permission and/or Privileged Mode access when the Page Request has a PASID TLP Prefix).
3. The system is, for an unspecified reason, unable to respond to the request. This response is terminal (the host may no longer respond to any page requests and may not supply any further replies to the Function until the Function's page request interface has been reset). For example, a request that violates a Function's assigned request limit or overflows the RC's buffering capability may cause this type of failure.

A Function's response to ~~p~~Page ~~r~~request failure cases 1, 2 and ~~2-3~~ above is implementation dependent. ~~A~~ The failure is not necessarily persistent, that is, a failed request may, in some instances succeed if reissued. The range of possibilities precludes the precise specification of a generalized failure behavior, though on a per Function basis, the response to a failure will be an implementation dependent behavior.

...

Add the new section 4.2.2:

#### 4.2.2. PASID TLP Prefix Usage

If a Page Request has a PASID TLP Prefix, the corresponding PRG Response Message may optionally contain one as well.

If the PRG Response PASID Enable bit is Clear, PRG Response Messages do not have a PASID TLP Prefix.

If the PRG Response PASID Enable bit is Set, PRG Response Messages have a PASID TLP Prefix if the Page Request also had one. The Function is permitted to use the PASID value from the prefix in conjunction with the PRG Index to match requests and responses.

---

<sup>5</sup> Behavior when PASID TLP Prefix support does not exist is defined in the *PCI Express Base Specification*.

In PASID TLP Prefixes attached to PRG Response Messages, the Execute Requested and Privileged Mode Requested bits are Reserved and the PASID value is copied from the PASID value of the Page Request.

Section 5.1.2, modify Figure 5-3 to:

1. Correct name of bit 5 to **Page** Aligned Request (also noted in ATS Errata)
2. Change reserved bits 15:7 from RsvdZ to RsvdP
3. Define bit 6 as **Global Invalidate Supported**

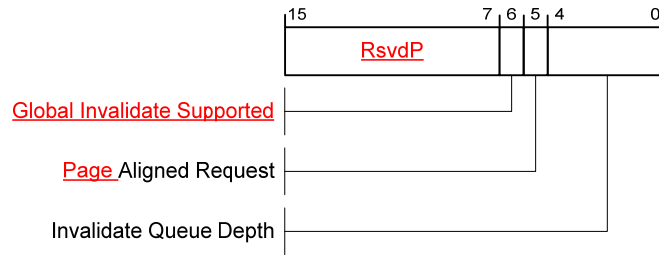


Figure 5-3: ATS Capability Register

Section 5.1.2, modify Table 5-2 as follows:

Table 5-2: ATS Capability Register

Bit Location	Register Description	Attributes
4:0	<b>Invalidate Queue Depth</b> – The number of Invalidate Requests that the Function can accept before putting backpressure on the upstream connection. If 0 0000b, the Function can accept 32 Invalidate Requests.	RO
5	<b>Page Aligned Request</b> – If Set indicates the Untranslated Address is always aligned to a 4096 byte boundary. Setting this <a href="#">field</a> is recommended. This <a href="#">field</a> permits software to distinguish between implementations <a href="#">compatible</a> with this specification and those compatible with an earlier version of this specification where a requester was permitted to supply anything in bits [11:2].	RO
<a href="#">6</a>	<b>Global Invalidate Supported</b> – If Set, the Function supports <a href="#">Invalidation Requests that have the Global Invalidate bit Set</a> . If Clear, the Function <a href="#">ignores the Global Invalidate bit in all Invalidate Requests (see Section 3.1.1)</a> . <a href="#">This bit is 0b if the Function does not support the PASID TLP Prefix.</a>	<a href="#">RO</a>

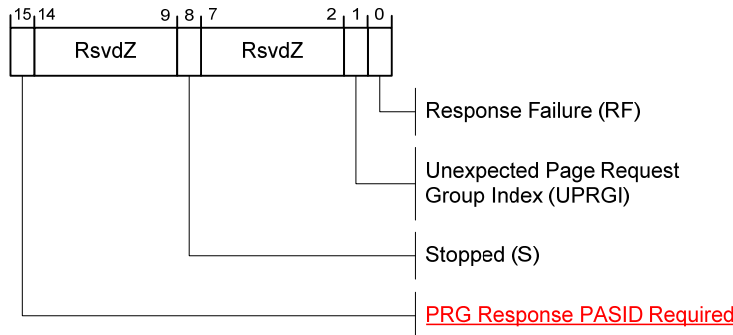
Section 5.1.3, modify Table 5-3 as follows:

**Table 5-3: ATS Control Register**

Bit Location	Register Description	Attributes
4:0	<b>Smallest Translation Unit (STU)</b> – This value indicates to the Function the minimum number of 4096-byte blocks that is indicated in a Translation Completions or Invalidate Requests. This is a power of 2 multiplier and the number of blocks is 2 <sup>STU</sup> . A value of 0 0000b indicates one block and a value of 1 1111b indicates 231 blocks (or 8 TB total). Default value is 0 0000b.	RW
15	<b>Enable (E)</b> – When Set, the Function is enabled to cache translations. <a href="#">Behavior is undefined if this bit is Set and the value of either PASID Enable, Execute Requested Enable, or Privileged Mode Requested are changed.</a> Default value is 0b.	RW

Section 5.2.3, modify Figure 5-8 to define bit 15 as PRG Response PASID Required

5



**Figure 5-8: Page Request Status Register**

Section 5.2.3, modify Table 5-6 as follows (note errata in caption):

**Table 5-6: Page Request ~~Error~~ Status Register**

Bit Location	Register Description	Attributes
<u>15</u>	<b><u>PRG Response PASID Required</u></b> – If Set, the Function expects a PASID TLP Prefix on PRG Response Messages when the corresponding Page Requests had a PASID TLP Prefix. If Clear, the Function does not expect PASID TLP Prefixes on any PRG Response Message. <a href="#">Function behavior is undefined if this bit is Clear and the Function receives a PRG Response Message with a PASID TLP Prefix.</a> <a href="#">Function behavior is undefined if this bit is Set and the Function receives a PRG Response Message with no PASID TLP Prefix when the corresponding Page Requests had a PASID TLP Prefix.</a> <a href="#">This bit is RsvdZ if the Function does not support the PASID TLP Prefix.</a>	<u>RO</u>

*Add new paragraphs at the end of Section 5.2.5:*

## 5.2.5. Outstanding Page Request Allocation (0Ch)

...

5

When PASID TLP Prefix is supported, the Request Allocation remains associated with the Function and is shared across the Function as well as all PASIDs of the Function.

Stopping a PASID does not affect any allocation used by that PASID. The system should continue to respond with PRG Response Messages in order to return Page Request and PRG Index resources to the Function (see Section 4.2.1).

10

Stop Marker Messages consume buffering but are not included in this allocation (see Section 4.1.2.1). Systems should provide additional buffering for Stop Marker Messages and should limit the number of outstanding Stop Marker Messages to avoid overrunning this additional buffering.