PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	TLP Processing Hints
DATE:	September 11, 2008
AFFECTED DOCUMENT:	PCI Express Base Specification version 2.0, Atomic Operations ECN
SPONSOR:	IBM, Intel Corporation, AMD, HP

Part I

1. Summary of the Functional Changes

This optional normative ECR defines a mechanism by which a Requester can provide hints on a per transaction basis to facilitate optimized processing of transactions that target Memory Space. The architected mechanisms may be used to enable association of system processing resources (e.g. caches) with the processing of Requests from specific Functions or enable optimized system specific (e.g. system interconnect and Memory) processing of Requests.

For Posted Requests and Non-Posted Requests that target Memory Space, a bit in the first DWORD of the TLP header is used to indicate the presence of TLP Processing Hints (TPH). Baseline TPH defines 10 bits for Processing Hints and Steering Tags. The Processing Hints use 2 reserved bits in the TLP header to indicate the communication usage model. The remaining 8 bits are used for a Steering Tag. For Memory Write Requests, the Tag field is repurposed to carry the Steering Tag. For Memory Read Requests and AtomicOp Requests, the first DW Byte Enables and last DW Byte Enables are repurposed to carry the Steering Tag.

For systems where more than 10 bits of TPH fields are required for scalability, an optional TPH TLP Prefix may be used. The TPH TLP Prefix extends the TPH fields by adding a DW Prefix in front of the TLP header. The first byte of the Prefix indicates that it is a TPH TLP Prefix, and the remaining 24 bits of the Prefix carry the Extended TPH fields.

2. Benefits as a Result of the Changes

The mechanisms defined in this optional normative ECR provide an opportunity for Requesters to make optimized use of the system fabric and improve system efficiency. The mechanisms provided can be effectively utilized to:

- Facilitate data residency/allocation within the system cache hierarchy
- Minimize device memory access latencies & minimize statistical variation in latencies
- Reduce memory & system interconnect BW & associated power consumption
- Not limited to the benefits described above

Providing such information enables the Root Complex and Endpoint to optimize handling of Requests by differentiating data likely to be reused soon from bulk flows that could monopolize system resources. System fabric resources, while large, are often system performance limiters, so providing information about intended data usage patterns allows best use of system resources including memory, caches, and system interconnects.

3. Assessment of the Impact

The mechanisms defined in this ECR are hints and make use of reserved and/or repurposed fields in Memory Read, Memory Write and AtomicOp Requests. It is expected that these fields should not impact the processing of Requests by intermediate routing elements. Endpoints take advantage only if needed, so there is not cost if not used.

Extended TPH requires hardware support for Requests with the TPH TLP Prefix. Details of the TLP Prefix mechanism are covered by the TLP Prefix ECR.

4. Analysis of the Hardware Implications

On the host side, if this capability is supported then the Root Ports are required to support the Capability structures and comprehend the mechanisms as described in this specification.

On the device side, if this capability is supported then the Endpoint Function is required to support the Capability structure and comprehend the mechanisms as described in this specification. It is assumed that there are device specific mechanisms/policies to determine which Requests will have TPH applied.

Extended TPH requires TLP Prefix support by the Requester, Completer and intermediate routing elements. Hardware implications for TLP Prefix support are covered by the TLP Prefix ECR.

5. Analysis of the Software Implications

A Capability structure is provided for software discovery, identification and control of TPH mechanisms. Software support (Device driver, OS, VMM and system specific driver) is required to retrieve the Steering Tag information from the system and provide the information to the Requester.

Part II

Detailed Description of the change

Modify section 2.2.1 as shown

2.2.1. Common Packet Header Fields

All Transaction Layer Packet (TLP) headers contain the following fields (see Figure 2 4):

…

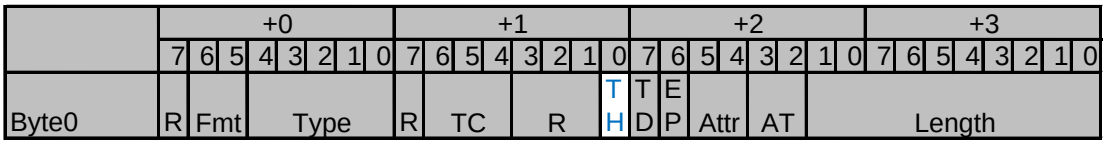

Figure 2-4: Fields Present in All TLPs

…

- TC[2:0] – Traffic Class (see Section 2.4.2) – bits [6:4] of byte 1
- TH – 1b indicates the presence of TLP Processing Hints (TPH) in the TLP header and optional TPH TLP Prefix (if present) – bit 0 of byte 1 (see Section 2.2.x.1)
- Attr[1:0] – Attributes (see Section 2.2.6.3) – bits [5:4] of byte 2

…

Modify section 2.2.5 as shown

2.2.5. First/Last DW Byte Enables Rules

Byte Enables are included with Memory, I/O, and Configuration Requests. This section defines the corresponding rules. Byte Enables, when present in the Request header, are located in byte 7 of the header (see Figure 2-9). For Memory Read Requests that have the TH bit Set, the Byte Enable fields are repurposed to carry the ST[7:0] field, and values for the Byte Enables are implied as defined below. Such Requests must only be issued when it is acceptable to complete the Requests as if all bytes for requested payload were enabled.

- For Memory Reads that have the TH bit Set, the following values are implied for the Byte Enables.
  - If the Length field for this Request indicates a length of 1 DW, then the value for the 1st DW Byte Enables is implied to be 1111b and the value for the Last DW Byte Enables is implied to be 0000b.
  - If the Length field for this Request indicates a length of greater than 1 DW, then the value for the 1st DW Byte Enables and the Last DW Byte Enables is implied to be 1111b.

IMPLEMENTATION NOTE

Read Request with TPH to Non-Prefetchable Space

Memory Read Requests with the TH bit Set and target Non-Prefetchable Memory Space should only be issued when it can be guaranteed that completion of such reads will not create undesirable side effects.

- ❑ The 1st DW BE[3:0] field contains Byte Enables for the first (or only) DW referenced by a Request.
  - If the Length field for a Request indicates a length of greater than 1 DW, this field must not equal 0000b.
- ❑ The Last DW BE[3:0] field contains Byte Enables for the last DW of a Request.
  - If the Length field for a Request indicates a length of 1 DW, this field must equal 0000b.
  - If the Length field for a Request indicates a length of greater than 1 DW, this field must not equal 0000b.

…

Modify section 2.2.6.2 as shown

2.2.6.2. Transaction Descriptor – Transaction ID Field

The Transaction ID field consists of two major sub-fields: Requester ID and Tag as shown in Figure 2 11.

…

- ❑ For Posted Requests ~~that do not require a Completion (Posted Requests)~~with the TH bit Set, the ~~value in the~~ Tag[7:0] field is repurposed for the ST[7:0] field (Refer to Section 2.2.x.1 for details). For Posted Requests with TH bit Clear, the Tag[7:0] field is undefined and may contain any value. (Refer to Section 2.2.8.6 for exceptions to this rule for certain Vendor_Defined Messages.)
  - For Posted Requests with the TH bit Clear, the value in the Tag[7:0] field must not affect Receiver processing of the Request
  - For Posted Requests with the TH bit Set, the value in the ST[7:0] field may affect Completer processing of the Request (Refer to Section 2.2.x.1 for details).

…

Modify Section 2.2.7. as shown

2.2.7. Memory, I/O, and Configuration Request Rules

The following rule applies to all Memory, I/O, and Configuration Requests. Additional rules specific to each type of Request follow.

- All Memory, I/O, and Configuration Requests include the following fields in addition to the common header fields:
  - Requester ID[15:0] and Tag[7:0], forming the Transaction ID
  - Last DW BE[3:0] and 1st DW BE[3:0]. For AtomicOp Requests with the TH bit Set, the byte location for the Last DW BE[3:0] and 1st DW BE [3:0] fields in the header are repurposed to carry ST[7:0] field, and the values for the DW BE fields are implied to be reserved. Otherwise, the DW BE fields are reserved.

Editorial note: The Atomic Operations ECN specified DW BE fields rules for AtomicOps, and this ECN augments those rules for AtomicOp Requests issued with TPH. The highlighted text above was introduced by the Atomic Operations ECN.

For Memory Requests, the following rules apply:

…

Add section 2.2.x.1 as shown

2.2.x.1 TPH Rules

- Two formats are specified for TPH. The Baseline TPH format (see Figures 2-x2 and 2-x3) must be used for all Requests that provide TPH. The format with the Optional TPH TLP Prefix extends the TPH fields (see Figure 2-x1) to provide additional bits for the Steering Tag (ST) field.

	+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
TLP Prefix	<see note>								ST(15:8)								Reserved															

Figure 2-x1: TPH TLP Prefix

- The Optional TPH TLP Prefix is used to extend the TPH fields.
  - The presence of a TPH TLP Prefix is determined by decoding byte 0.

Table 2-x1: TPH TLP Prefix Bit Mapping

Fields	TPH TLP Prefix
ST(15:8)	Bits 7:0 of byte 1
Reserved	Bits 7:0 of byte 2
Reserved	Bits 7:0 of byte 3

Editorial note: The TLP Prefix mechanism, its associated rules, and the TPH TLP Prefix byte 0 encoding are specified in the TLP Prefix ECN.

- For Requests that target Memory Space, a value of 1b in the TH bit indicates the presence of TPH in the TLP header and optional TPH TLP Prefix (if present).
  - Must be set for Requests that provide TPH
  - Must be set for Requests with a TPH TLP Prefix
  - The TH bit is not applicable and is reserved for all other Requests.
- The Processing Hints (PH) fields mapping is shown in Figure 2-x2, Figure 2-x3, and Table 2-x2.

	+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
Byte0	R	Fmt		Type					R	TC			R			TH	TD	EP	Attr		AT		Length										
Byte4	{Fields in byte 4 through 7 depend on type of Request}																															
Byte8	Address[63:32]																															
Byte12	Address[31:2]																														PH	

Figure 2-x2: Location of PH[1:0] in a 4DW Request Header

	+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
Byte0	R	Fmt		Type					R	TC			R			TH	TD	EP	Attr		AT		Length										
Byte4	{Fields in byte 4 through 7 depend on type of Request}																															
Byte8	Address[31:2]																														PH	

Figure 2-x3: Location of PH[1:0] in a 3DW Request Header

Table 2-x2: Location of PH[1:0] in TLP Header

PH	32-bit Addressing	64-bit Addressing
1:0	Bits 1:0 of Byte 11	Bits 1:0 of Byte 15

- The PH[1:0] field provides information about the data access patterns and is defined as follows

Table 2-x3: Processing Hint Encoding

PH[1:0]	Processing Hint	Description
00	Bi-directional data structure	Indicates frequent read and/or write access to data by Host and device
01	Requester	Indicates frequent read and/or write access to data by device
10	Target	Indicates frequent read and/or write access to data by Host
11	Target with Priority	Indicates frequent read and/or write access by Host and indicates high temporal locality for accessed data

- The Steering Tag (ST) fields are mapped to the TLP header as shown in Figure 2-x4, Figure 2-x5, and Table 2-x4.

	+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
Byte0	R	Fmt		Type					R	TC			R			TH	TD	EP	Attr		AT		Length										
Byte4	Requestor ID																ST(7:0)								Last DW				1st DW			

Figure 2-x4: Location of ST[7:0] in the Memory Write Request Header

	+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
Byte0	R	Fmt		Type					R	TC			R			TH	TD	EP	Attr		AT		Length										
Byte4	Requestor ID																Tag								ST(7:0)							

Figure 2-x5: Location of ST[7:0] in Memory Read and AtomicOp Request Headers

Table 2-x4: Location of ST bits 7:0 in TLP Headers

ST Bits	Memory Write Request	Memory Read Request or AtomicOp Request
7:0	Bits 7:0 of Byte 6	Bits 7:0 of Byte 7

- ST[7:0] field carries the Steering Tag value
  - A value of all zeroes indicates no Steering Tag preference

- A total of 255 unique Steering Tag values are provided

- A Function that does not support TPH Completer or Routing capability and receives a transaction with TH bit Set is required to ignore the TH bit and handle the Request the same as with other Requests of the same transaction type without the TH bit Set.

Modify section 2.2.7 as shown below

2.2.7. Memory, I/O, and Configuration Request Rules

…

- I/O Requests have the following restrictions:
  - TC[2:0] must be 000b
  - TH is not applicable to IO Requests and the bit is reserved.

…

- Configuration Requests have the following restrictions:
  - TC[2:0] must be 000b
  - TH is not applicable to Configuration Requests and the bit is reserved.

…

Modify section 2.2.8 as shown below

2.2.8. Message Request Rules

…

- Except as noted, the Attr[1:0] field is reserved.
- Except as noted, TH is not applicable to Message Requests and the bit is reserved.

…

Add new section 6.xx

6.xx. TLP Processing Hints (TPH)

TLP Processing Hints is an optional feature that provides hints in Request TLP headers to facilitate optimized processing of Requests that target Memory Space. These Processing Hints enable the system hardware (e.g. the Root Complex and/or Endpoints) to optimize platform resources such as system and memory interconnect on a per TLP basis. The TPH mechanism defines Processing Hints that provide information about the communication models between Endpoints and the Root-complex. Steering Tags are system-specific values used to identify a processing resource that a Requester explicitly targets. System software discovers and identifies TPH capabilities to determine the Steering Tag allocation for each Function that supports TPH.

6.xx.1. Processing Hints

The Requester provides hints to the Root Complex or other targets about the intended use of data and data structures by the host and/or device. The hints are provided by the Requester, which has knowledge of upcoming Request patterns, and which the Completer would not be able to deduce autonomously (with good accuracy). Cases of interest to distinguish with such hints include:

DWHR: Device writes then host reads soon

HWDR: Device reads data that the Host is believed to have recently written

D*D*: Device writes/reads, then device reads/writes soon

Includes DWDW, DWDR, DRDW, DRDR

Bi-Directional: Data structure that is shared and has equal read/write access by host and device.

The usage models are mapped to the Processing Hint encodings as described in Table 6.xx.

Table 6-xx: Processing Hint Mapping

PH[1:0]	Processing Hint	Usage Model
00	Bi-directional data structure	Bi-Directional shared data structure
01	Requester	D*D*
10	Target	DWHR HWDR
11	Target with Priority	Same as target but with temporal re-use priority

6.xx.2. Steering Tags

Functions that intend to target a TLP towards a specific processing resource such as a host processor or system cache hierarchy require topological information of the target cache, e.g. which host cache. Steering Tags are system-specific values that provide information about the host or cache structure in the system cache hierarchy. These values are used to associate processing elements within the platform with the processing of Requests.

Software programmable Steering Tag values to be used are stored in an ST Table that is permitted to be located in the TPH Requester Capability structure (see Section 7.x) or combined with the MSI-X Table (see Section 7.7), but not in both locations for a given Function. When the ST Table is combined with the MSI-X Table, the 2 most significant bytes of the Vector Control register of each MSI-X Table entry are used to contain the Steering Tag value.

The choice of ST Table location is implementation specific and is discoverable by software. Each ST Table entry is 2 bytes. The size of the ST Table is indicated in the TPH Requester Capability structure.

For some usage models the Steering Tags are not required or not provided, and in such cases a Function is permitted to use a value of all zeroes in the ST field to indicate no ST preference. The association of each Request with an ST Table entry is device specific and outside the scope of this specification.

6.xx.3. ST Modes of Operation

The ST Table Location field in the TPH Requester Capability Structure indicates where (if at all) the ST Table is implemented by the Function. If an ST Table is implemented, software can program it with the system-specific Steering Tag values.

Table 6-xx: ST Modes of Operation

ST Mode Select [2:0]	ST Mode Name	Description
000	No ST Mode	The Function must use a value of all zeroes for all Steering Tags
001	Interrupt Vector Mode	Each Steering Tag is selected by an MSI/MSI-X interrupt vector number. The Function is required to use the Steering Tag value from an ST Table entry that can be indexed by a valid MSI/MSI-X interrupt vector number.
010	Device Specific Mode	It is recommended for the Function to use a Steering Tag value from an ST Table entry, but it is not required.
	All other encodings	Reserved for future use

In the No ST Mode of operation, the Function must use a value of all zeroes for each Steering Tag, enabling the use Processing Hints without software-provided Steering Tags.

In the Interrupt Vector Mode of operation, Steering Tags are selected from the ST Table using MSI/MSI-X interrupt vector numbers. For Functions that have MSI enabled, the Function is required to select tags within the range specified by the Multiple Message Enable field in the MSI Capability structure. For Functions that have MSI-X enabled, the Function is required to select tags within the range of the MSI-X Table size.

In the Device Specific Mode of operation, the assignment of the Steering Tags to Requests is device specific. The number of Steering Tags used by the Function is permitted to be different than the number of interrupt vectors allocated for the Function, irrespective of the

ST Table location, and Steering Tag values used in Requests are not required to come from the architected ST Table.

A Function that is capable of generating TPH Requests is required to support the No ST Mode of operation, and support for the other modes of operations is optional. Only one mode of operation can be selected at a time by programming ST Mode Select.

 IMPLEMENTATION NOTE

ST Table Programming

To ensure that deterministic Steering Tag values are used in Requests, it is recommended that software either quiesce the Function or disable the TPH Requester capability during the process of performing ST Table updates. Failure to do so may result in non-deterministic values of ST values being used during ST Table updates.

6.xx.4. TPH Capability

TPH capabilities are optional normative. Each Function capable of generating Request TLPs with TPH is required to implement a TPH Requester Capability structure. Functions that support processing of TLPs with TPH as Completers are required to indicate TPH Completer capability via the Device Capabilities 2 Register. TPH is architected to be applied for transactions that target Memory Space, and is applicable for transaction flows between device-to-host, device-to-device and host-to-device. In each case, the Requester, Completer, and all intermediate routing elements must support the associated TPH capabilities.

Software discovers the Requester capabilities via the TPH Requester Capability structure and Completer capabilities via the Device Capabilities 2 register (see section 7.8.15). Software must program the TPH Requester Enable field in the TPH Requester Capability structure to enable the Function to initiate Requests with TPH.

TPH only provides additional information to enable optimized processing of Requests that target Memory Space, so existing mechanisms and rules for managing Memory Space access like Bus Master Enable, Memory Space Enable, and Base Address Registers are not altered.

Modify section 7.7 as shown. Editorial note: the Base Specification text shown here already incorporates errata, and indicated changes are relative to the errata text.

7.7. MSI and MSI-X Capability Structures

All PCI Express device Functions that are capable of generating interrupts must implement MSI or MSI-X or both. MSI, MSI-X, and their Capability structures are defined in the *PCI Local Bus Specification, Revision 3.0*. The functionality associated with these structures defined by conventional PCI is also required for PCI Express. Only added requirements associated with PCI Express are described here.

7.7.1. Vector Control for MSI-X Table Entries

If a Function implements a TPH Requester Capability structure and an MSI-X Capability structure, the Function can optionally use the Vector Control register in each MSI-X Table Entry to store a Steering Tag. See Section 6.xx.

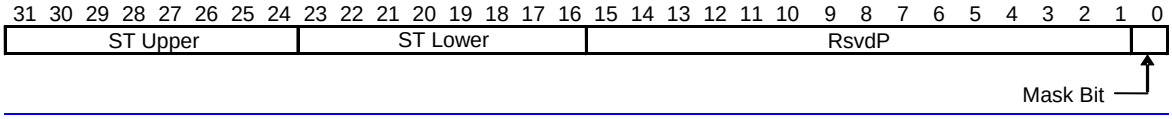

Figure 7-xx: Vector Control for MSI-X Table Entries

Table 7-xx: Vector Control for MSI-X Table Entries

Bit Location	Register Description	Attributes
0	**Mask Bit** – No added requirements	Unchanged
23:16	**ST Lower** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 10b, then this field contains the lower 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW
31:24	**ST Upper** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 10b, and the Extended TPH Requester Supported bit is Set, then this field contains the upper 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW

7.x TPH Requester Capability

The TPH Requester Capability structure is required for all Functions that are capable of generating Request TLPs with TPH. For a multi-Function device, this capability must be present in each Function that is capable of generating Requests with TPH.

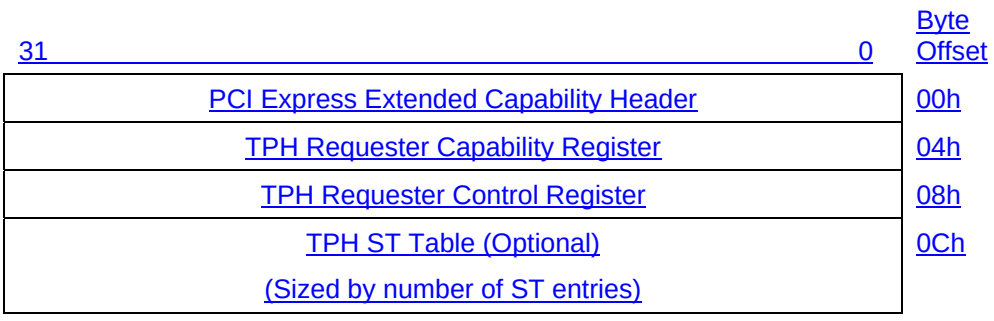

Figure 7-xx-1: TPH Extended Capability Structure

7.x.1 TPH Requester Extended Capability Header (Offset 00h)

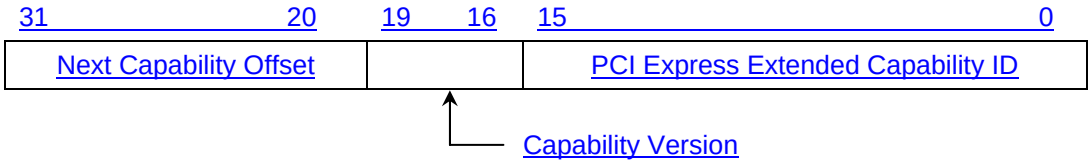

Figure 7-xx-2: TPH Requester Extended Capability Header

Table 7-xx-1: TPH Requester Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the TPH Requester Capability is **17h.**	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.x.2 TPH Requester Capability Register (Offset 04h)

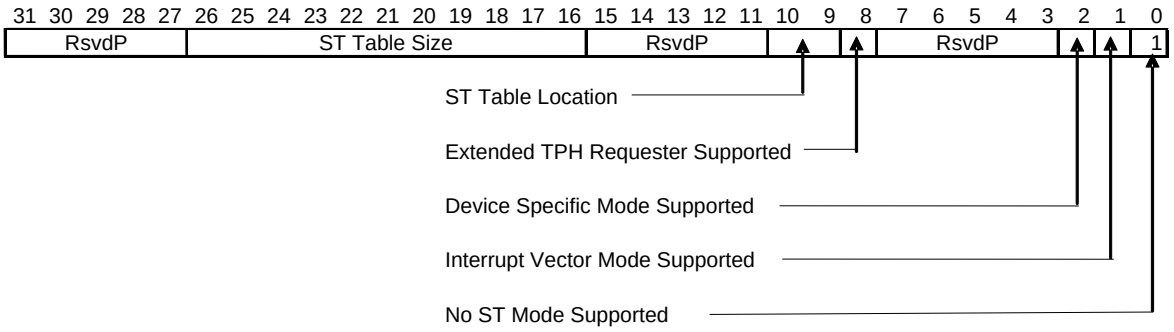

Figure 7-xx-3: TPH Requester Capability Register

Table 7-xx-2: TPH Requester Capability Register

Bit Location	Register Description	Attributes
0	**No ST Mode Supported** – If set indicates that the Function supports the No ST Mode of operation. This mode is required to be supported by all Functions that implement this Capability structure. This field must have a value of 1b.	RO
1	**Interrupt Vector Mode Supported** – If set indicates that the Function supports the Interrupt Vector Mode of operation.	RO
2	**Device Specific Mode Supported** – If set indicates that the Function supports the Device Specific Mode of operation.	RO
8	**Extended TPH Requester Supported** – If Set indicates that the Function is capable of generating Requests with a TPH TLP Prefix. See section 2.2.x.1 for additional details.	RO
10:9	**ST Table Location** – Value indicates if and where the ST Table is located. Defined Encodings are: 00 – ST Table is not present. 01 – ST Table is located in the TPH Requester Capability structure. 10 – ST Table is located in the MSI-X Table structure. 11 – Reserved A Function that only supports the No ST Mode of operation must have a value of 00b in this field.	RO
26:16	**ST Table Size** – Software reads this field to determine the ST Table Size N, which is encoded as N-1. For example, a returned value of “00000000011” indicates a table size of 4. There is an upper limit of 64 entries when the ST Table is located in the TPH Requester Capability structure. There is an upper limit of 2K entries when the ST Table is located in the MSI-X Table. This field is only applicable for Functions that implement an ST Table as indicated by the ST Table Location field. Otherwise, the value in this field is undefined.	RO

7.x.3 TPH Requester Control Register (Offset 08h)

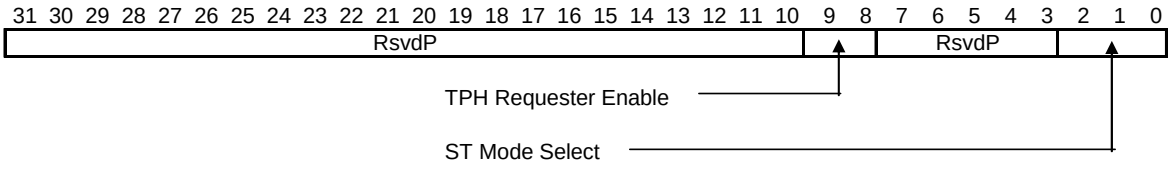

Figure 7-xx-4: TPH Requester Control Register

Table 7-xx-3: TPH Requester Control Register

Bit Location	Register Description	Attributes
2:0	**ST Mode Select** – selects the ST mode of operation. Defined encodings are: 000b – No ST Mode 001b – Interrupt Vector Mode 010b – Device Specific Mode 0thers – reserved for future use Functions that support only the No ST Mode of operation must hardwire this field to 000b. The default value of this field is 000b. See section 6.xx.2 for details on ST modes of operation.	RW
9:8	**TPH Requester Enable** – defined encodings are: 00b – Function operating as a Requester is not permitted to issue Requests with TPH or Extended TPH. 01b – Function operating as a Requester is permitted to issue Requests with TPH and is not permitted to issue Requests with Extended TPH. 10b – Reserved. 11b – Function operating as a Requester is permitted to issue Requests with TPH and Extended TPH. The default value of this field is 00b.	RW

7.x.4 TPH ST Table (Starting from Offset 0Ch)

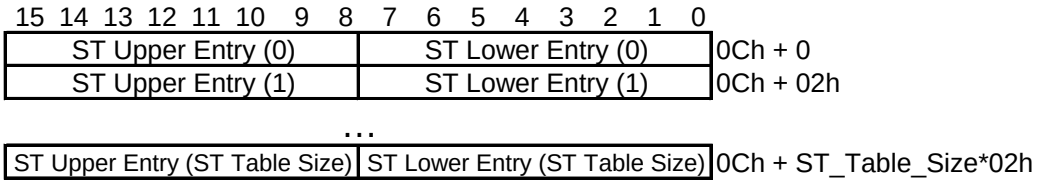

Figure 7-xx-5: TPH ST Table

The ST Table must be implemented in the TPH Requester Capability Structure if the value of the ST Table Location field is 01b. For all other values, the ST Entry Registers must not be

implemented. The number of ST Entry Registers implemented must be equal to the number of ST entries supported by the Function, which is the value of the ST Table Size field plus one.

Table 7-xx-4: TPH ST Table

Bit Location	Register Description	Attributes
7:0	**ST Lower** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 01b, then this field contains the lower 8 bits of a Steering Tag. Default value of this field is 0h.	RW
15:8	**ST Upper** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 01b, and the Extended TPH Requester Supported bit is Set, then this field contains the upper 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW

Modify Section 7.8.15 as shown

7.8.15. Device Capabilities 2 Register (Offset 24h)

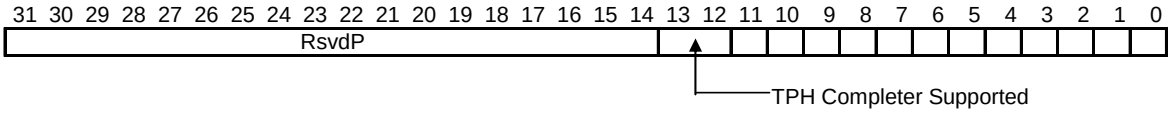

Figure 7-24: Device Capabilities 2 Register

Table 7-23: Device Capabilities 2 Register

Bit Location	Register Description	Attributes
…	…	…
13:12	**TPH Completer Supported** – Applicable only to Root Ports and Endpoints; must be 00b for other Function types. Defined Encodings are: 00b – TPH and Extended TPH Completer not supported. 01b – TPH Completer supported; Extended TPH Completer not supported. 10b – Reserved. 11 – Both TPH and Extended TPH Completer supported. See section 6.xx for details.	RO