PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	TLP Prefix
DATE:	December 15, 2008
AFFECTED DOCUMENT:	PCIe Base Specification 2.0
SPONSOR:	Intel Corporation, AMD, HP, NextIO

Part I

1. Summary of the Functional Changes

Emerging usage model trends indicate a requirement for increase in header size fields to provide additional information than what can be accommodated in currently defined TLP header sizes. The TLP Prefix mechanism extends the header size by adding DWORDS to the front of headers that carry additional information. The base size of a TLP Prefix is a single DWORD and the mechanism allows for multiple TLP Prefixes to be present in a TLP to either provide space for additional fields or to allow for multiple separate TLP Prefixes to be present in a TLP. The TLP Prefix mechanism changes the format of a TLP such that a TLP is comprised of an optional TLP Prefix, TLP Header, optional Data Payload and optional TLP Digest.

There are two types of TLP Prefixes defined, namely Local TLP Prefix and End-End TLP Prefix. The Local TLP Prefix scope is limited to the Link whereas the End-End TLP Prefix scope is between the originator and target of a Request or Completion TLP. For End-End TLP Prefixes, the originator and target may be directly connected or may be connected through a hierarchy of routing elements.

2. Benefits as a Result of the Changes

The TLP Prefix mechanism provides architectural headroom for PCIe headers to grow in the future. Switches and Switch related software can be built that are transparent to the encoding of future End-End TLPs. The End-End TLP Prefix mechanism defines rules for routing elements to route TLPs containing End-End TLP Prefixes without requiring the routing element logic to explicitly support any specific End-End TLP Prefix encoding(s).

3. Assessment of the Impact

TLP Prefix is an optional normative feature and the impact is incurred by those devices that choose to support this mechanism. The End-End TLP Prefix mechanism requires an increase in the buffering requirements for Receivers without impacting the flow control accounting. The header flow control credit takes into account the amount of buffering required for TLP Prefixes.

TLP processing logic is required to comprehend the presence of the Optional TLP Prefix in a TLP and, for Functions that support TLP Prefix, will have to handle the control of TLP Prefixes as a requestor or completer or both as the case may be.

For Local TLP Prefix the impact is dependent on the Local TLP Prefix type.

4. Analysis of the Hardware Implications

TLP Prefix is an optional normative feature and the impact is incurred only by those devices that choose to support this mechanism. Functions supporting TLP Prefix have to consider the buffering requirements associated with TLP Prefix. For Functions that support End-End TLP Prefixes, the header flow control credit unit takes into account the maximum size of TLP Prefixes. For End-End TLP Prefixes the maximum allowed size is 4 DWORDs.

The hardware implications of the Local TLP Prefix are dependent on the TLP Prefix type encoding. Devices that don't support Local TLP Prefixes (or don't support a specific Local TLP Prefix) are required to terminate the TLP as a Malformed TLP.

5. Analysis of the Software Implications

There are new fields in the Device Capabilities 2 and Device Control 2 registers for software to broadly manage and control the TLP Prefix mechanism. Additionally it is expected that there will be TLP Prefix specific management and control registers to configure specific TLP Prefix related attributes.

For End-End TLP Prefixes, software must ensure that the entire communications path from requestor (completer) to target supports TLP Prefix before enabling the feature.

6. Analysis of I/O Virtualization (IOV) Implications

End-End, SR-IOV:	A Virtual Intermediary (VI) can hide the presence of support for End-End TLP Prefixes from guest operating systems by hiding the appropriate capabilities. Depending on the nature of the End-End Prefix, the VI may enable its use even when the associated Guest OS does not comprehend the Prefix (e.g. crypto capabilities).
Local, SR-IOV:	Local Prefixes affect all Functions on either end of a Link and will likely be managed by the VI.
End-End, MR-IOV:	End-End prefixes operate within a specific VH. Software operating in that VH enables or disables usage. Malfunctioning software that enables End-End Prefixes over Link(s) that do not support them will result in Malformed TLP or TLP Prefix Blocked errors signaled within the affected VH.
Local, MR-IOV:	Local TLP Prefixes affect all Functions in all VHs on either end of a Link and will likely be managed by an MR-PCIM. The MR-IOV TLP Prefix is a Local Prefix that is automatically enabled when a Link trains in MR-IOV mode.

Part II

Detailed Description of the change

Add the following terms to Terms & Acronyms

TLP Prefix	**Additional information that may be optionally prepended to a TLP. TLP Prefixes are either Local or End-End. A TLP can have multiple TLP Prefixes. See Section 2.2.10.**
End-End TLP Prefix	**A TLP Prefix that is carried along with a TLP from source to destination. See Section 2.2.10.2.**
Local TLP Prefix	**A TLP Prefix that is carried along with a TLP on a single Link. See Section 2.2.10.1.**

Modify section 2.1.2 as follows

2.1.2. Packet Format Overview

Transactions consist of Requests and Completions, which are communicated using packets. Figure 2-2 shows a high level serialized view of a Transaction Layer Packet (TLP), consisting of one or more optional TLP Prefixes, a TLP header, a data payload (for some types of packets), and an optional TLP digest. Figure 2-3 shows a more detailed view of the TLP. The following sections of this chapter define the detailed structure of the packet headers and digest.

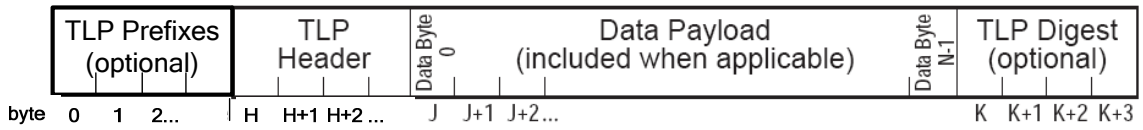


Figure 2-2: Serial View of a TLP

PCI Express conceptually transfers information as a serialized stream of bytes as shown in **Figure** 2-2. Note that at the byte level, information is transmitted/received over the interconnect with the leftmost byte ~~0~~ of the TLP as shown in Figure 2-2 being transmitted/received first (byte 0 if one or more Optional TLP Prefixes are present else byte H). Refer to Section 4.2 for details on how individual bytes of the packet are encoded and transmitted over the physical media.

Detailed layouts of the TLP Prefix, TLP Header and TLP Digest (presented in generic form in Figure 2-3) are drawn with the lower numbered bytes on the left rather than on the right as has traditionally been depicted in other PCI specifications. The header layout is optimized for performance on a serialized interconnect, driven by the requirement that the most time critical information be transferred first. For example, within the TLP header, the most significant byte of the address field is transferred first so that it may be used for early address decode.

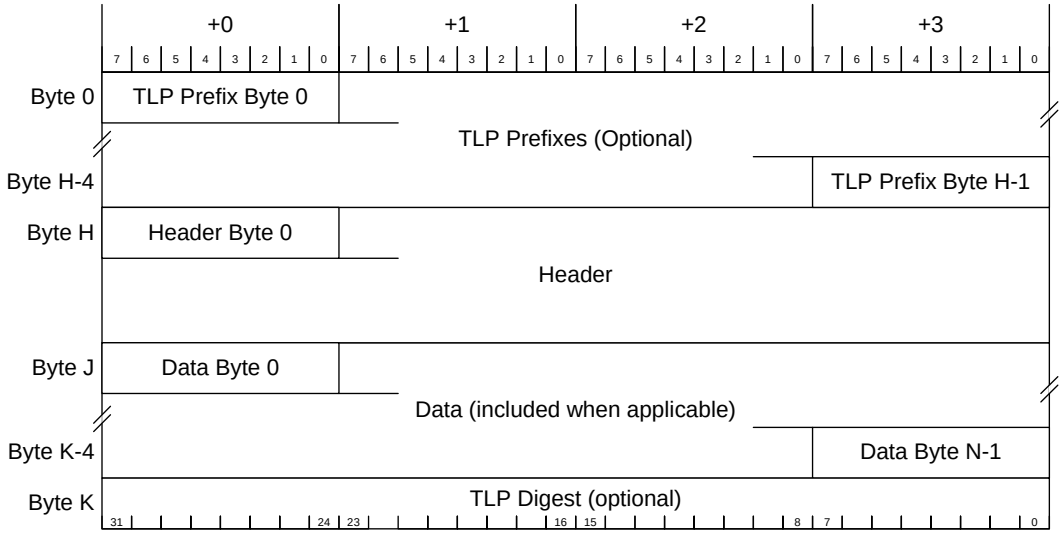


Figure 2-3: Generic TLP Format

Payload data within a TLP is depicted with the lowest addressed byte (byte J in Figure 2-3) shown to the upper left. Detailed layouts depicting data structure organization (such as the Configuration Space depictions in Chapter 7) retain the traditional PCI byte layout with the lowest addressed byte shown on the right. Regardless of depiction, all bytes are conceptually transmitted over the Link in increasing byte number order.

Depending on the type of a packet, the header for that packet will include some of the following types of fields:

- Format of the packet
- Type of the packet
- Length for any associated data
- Transaction Descriptor, including:
  - Transaction ID
  - Attributes
  - Traffic Class
- Address/routing information
- Byte Enables
- Message encoding
- Completion status

Modify section 2.2 as follows

2.2. Transaction Layer Protocol – Packet Definition

PCI Express uses a packet based protocol to exchange information between the Transaction Layers of the two components communicating with each other over the Link. PCI Express supports the following basic transaction types: Memory, I/O, Configuration, and Messages. Two addressing formats for Memory Requests are supported: 32 bit and 64 bit.

Transactions are carried using Requests and Completions. Completions are used only where required, for example, to return read data, or to acknowledge Completion of I/O and Configuration Write Transactions. Completions are associated with their corresponding Requests by the value in the Transaction ID field of the Packet header.

All TLP fields marked Reserved (sometimes abbreviated as R) must be filled with all 0's when a TLP is formed. Values in such fields must be ignored by Receivers and forwarded unmodified by Switches. Note that for certain fields there are both specified and reserved values – the handling of reserved values in these cases is specified separately for each case.

2.2.1. Common Packet Header Fields

All Transaction Layer Packet (TLP) prefixes and headers contain the following fields (see Figure 2-x):

- Fmt[2:0] – Format of TLP (see Table 2-1) – bits 7:5 of byte 0
- Type[4:0] – Type of TLP – bits 4:0 of byte 0

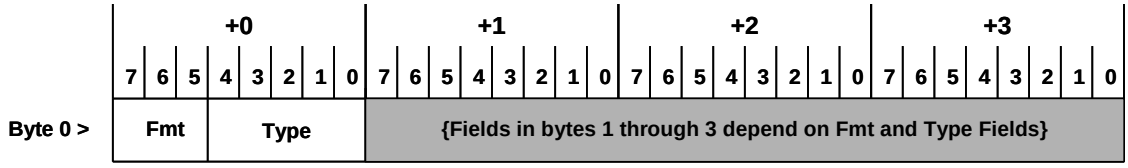


Figure 2-x: Fields Present in All TLPs

The Fmt field(s) indicate the presence of one or more TLP Prefixes and the Type field(s) indicates the associated TLP Prefix type(s).

The Fmt and Type fields of the TLP Header provide the information required to determine ~~if the packet contains ,~~ the size of the remaining part of the TLP H~~h~~eader, and if the packet contains a data payload following the header.

The Fmt, Type, TD, and Length fields of the TLP Header contain all information necessary to determine the overall size of the non-Prefix portion of the TLP ~~itself~~. The Type field, in addition to defining the type of the TLP also determines how the TLP is routed by a Switch. Different types of TLPs are discussed in more detail in the following sections.

- Permitted Fmt[2:0] and Type[4:0] field values are shown in Table 2-3.
  - All other encodings are reserved (see Section 2.3).
- TC[2:0] – Traffic Class (see Section 2.4.2) – bits [6:4] of byte 1
- Attr[1:0] – Attributes (see Section 2.2.6.3) – bits [5:4] of byte 2
- TD – 1b indicates presence of TLP digest in the form of a single DW at the end of the TLP (see Section 2.2.3) – bit 7 of byte 2

- EP – indicates the TLP is poisoned (see Section 2.7) – bit 6 of byte 2
- Length[9:0] – Length of data payload in DW (see Table 2-4) – bits 1:0 of byte 2 concatenated with bits 7:0 of byte 3
  - TLP data must be 4-byte naturally aligned and in increments of 4-byte Double Words (DW).
  - Reserved for TLPs that do not contain or refer to data payloads, including Cpl, CplLk, and Messages (except as specified)

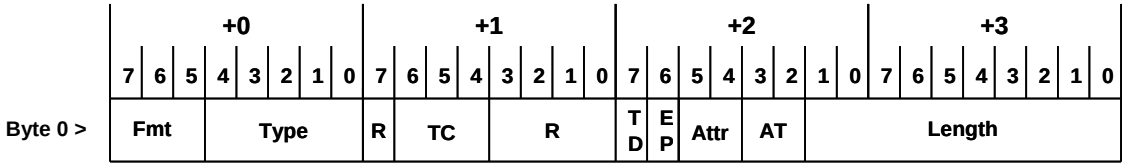


Figure 2-4: Fields Present in All TLP Headers

Table 2-1: Fmt[1:0] Field Values

Fmt[2:0]	Corresponding TLP Format
000b	3 DW header, no data
001b	4 DW header, no data
010b	3 DW header, with data
011b	4 DW header, with data
100b	TLP Prefix
	All encodings not shown above are reserved (see Section 2.3).

Table 2-3: Fmt[1:0] and Type[4:0] Field Encodings

TLP Type	Fmt [2:0][1] (b)	Type [4:0] (b)	Description
MRd	000 001	0 0000	Memory Read Request
MRdLk	000 001	0 0001	Memory Read Request-Locked
MWr	010 011	0 0000	Memory Write Request
IORd	000	0 0010	I/O Read Request
IOWr	010	0 0010	I/O Write Request
CfgRd0	000	0 0100	Configuration Read Type 0
CfgWr0	010	0 0100	Configuration Write Type 0
CfgRd1	000	0 0101	Configuration Read Type 1
CfgWr1	010	0 0101	Configuration Write Type 1
TCfgRd	000	1 1011	Deprecated TLP Type[2]
TCfgWr	010	1 1011	Deprecated TLP Type[2]
Msg	001	1 0$r_2r_1r_0$	Message Request – The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-12).
MsgD	011	1 0$r_2r_1r_0$	Message Request with data payload – The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-12).
Cpl	000	0 1010	Completion without Data – Used for I/O and Configuration Write Completions and Read Completions (I/O, Configuration, or Memory) with Completion Status other than Successful Completion.
CplD	010	0 1010	Completion with Data – Used for Memory, I/O, and Configuration Read Completions.
CplLk	000	0 1011	Completion for Locked Memory Read without Data – Used only in error case.
CplDLk	010	0 1011	Completion for Locked Memory Read – otherwise like CplD.
LPrfx	100	0$L_3L_2L_1L_0$	Local TLP Prefix – The sub-field L[3:0] specifies the Local TLP Prefix type (see Table 2-xx)

[1] Requests with two Fmt[2:0] values shown can use either 32 bits (the first value) or 64 bits (the second value) Addressing Packet formats.

[2] Deprecated TLP Types: previously used for TCS, which is no longer supported by this specification. If a Receiver does not implement TCS, the Receiver must treat such Requests as Malformed Packets.

TLP Type	Fmt [2:0][1] (b)	Type [4:0] (b)	Description
EPrfx	100	$1E_3E_2E_1E_0$	End-End TLP Prefix – The sub-field E[3:0] specifies the End-End TLP Prefix type (see Table 2-yy)
			All encodings not shown above are Reserved (see Section 2.3).

Add a new section 2.2.10

2.2.10 TLP Prefix Rules

The following rules apply to any TLP that contains a TLP Prefix

- For any TLP, a value of 100b in the Fmt[2:0] in byte 0 of the TLP indicates the presence of a TLP Prefix and the Type[4] field indicates the type of TLP Prefix
  - A value of 0b in the Type[4] field indicates the presence of a Local TLP Prefix
  - A value of 1b in the Type[4] field indicates the presence of an End-End TLP Prefix
- The format for bytes 1 through 3 of a TLP Prefix are defined by its TLP Prefix type
- A TLP that contains a TLP Prefix must have an underlying TLP Header. A received TLP that violates this rule is handled as a Malformed TLP. This is a reported error associated with the Receiving Port (see Section 6.2)
- It is permitted for a TLP to contain more than one TLP Prefix of any type
  - When a combination of Local and End-End TLP Prefixes are present in TLP, it is required that all the Local TLP Prefixes preceed any End-End TLP Prefixes. A received TLP that violates this rule is handled as a Malformed TLP. This is a reported error associated with the Receiving Port (see Section 6.2)
- The size of each TLP Prefix is 1DW. A TLP Prefix may be repeated to provide space for additional data.
- If the value in the Fmt and Type field indicates the presence of a Local TLP Prefix, handle according to the Local TLP Prefix handling (see Section 2.2.10.1)
- If the value in the Fmt and Type field indicates the presence of an End-End TLP Prefix, handle according to the End-End TLP Prefix handling (see Section 2.2.10.2)

2.2.10.1 Local TLP Prefix Processing

The following rules apply to Local TLP Prefixes

- Local TLP Prefix types are determined using the L[3:0] sub-field of the Type field
  - Type[4] must be 0b
  - Local TLP Prefix L[3:0] values are defined in Table 2-xx

Table 2-xx Local TLP Prefix Types

Local TLP Prefix Type	L[3:0] (b)	Description
MR-IOV	0000	**MR-IOV TLP Prefix** – Refer to the *Multi-Root I/O Virtualization and Sharing* specification for details
VendPrefixL0	1110	**Vendor Defined Local TLP Prefix** – Refer to section 2.2.10.1.1 for further details.
VendPrefixL1	1111	**Vendor Defined Local TLP Prefix** – Refer to section 2.2.10.1.1 for further details.
		All other encodings are reserved

- The size, routing and flow control rules are specific to each Local TLP Prefix type
- It is an error to receive a TLP with a Local TLP Prefix type not supported by the Receiver. If the Extended Fmt Field Supported bit is Set, TLPs in violation of this rule are handled as a Malformed TLP unless explicitly stated differently in another specification. This is a reported error associated with the Receiving Port (see Section 6.2). If the Extended Fmt Field Supported bit is Clear, behavior is device specific.
- No Local TLP Prefixes are protected by ECRC even if the underlying TLP is protected by ECRC.

2.2.10.1.1 Vendor Defined Local TLP Prefix

As described in Table 2-xx, Types VendPrefixL0 and VendPrefixL1 are reserved for usage as Vendor Defined Local TLP Prefixes. To maximize interoperability and flexibility the following rules are applied to such prefixes:

- Components must not send TLPs containing Vendor Defined Local TLP Prefixes unless this has been explicity enabled (using vendor specific mechanisms).
- Components that support any usage of Vendor Defined Local TLP Prefixes must support the 3 bit definition of the Fmt field and have the Extended Fmt Field Supported bit Set (see Section 7.8.15).
- It is recommended that components be configurable (using vendor specific mechanisms) so that all vendor defined prefixes can be sent using either of the two Vendor Defined Local TLP Prefix encodings. Such configuration need not be symmetric (for example each end of a Link could transmit the same Prefix using a different encoding).

2.2.10.2 End-End TLP Prefix Processing

The following rules apply to End-End TLP Prefixes

- End-End TLP Prefix types are determined using the E[3:0] sub-field of the Type field
  - Type[4] must be 1b
  - End-End TLP Prefix E[3:0] values are defined in table 2-yy

Table 2-yy End-End TLP Prefix Types

End-End TLP Prefix Type	E[3:0] (b)	Description

End-End TLP Prefix Type	E[3:0] (b)	Description
ExtTPH	0000	**Extended TPH** – Refer to section <tbd> for further details
VendPrefixE0	1110	**Vendor Defined End-End TLP Prefix** – Refer to section 2.2.10.2.1 for further details
VendPrefixE1	1111	**Vendor Defined End-End TLP Prefix** – Refer to section 2.2.10.2.1 for further details
		All other encodings are reserved

Note to Editor: ExtTPH is covered by another ECN. Please correct the <tbd> reference to point to that ECN when incorporating this change into the base specification.

- The maximum number of End-End TLP Prefixes permitted in a TLP is 4;
  - A Receiver supporting TLP Prefixes must check this rule. If a Receiver determines that a TLP violates this rule, the TLP is a Malformed TLP. This is a reported error associated with the Receiving Port (see Section 6.2)
- The presence of an End-End TLP Prefix does not alter the routing of a TLP. TLPs are routed based on the routing rules covered in section 2.2.4.
- Functions indicate how many End-End TLP Prefixes they support by the Max_End_End TLP Prefixes field in the Device Capabilities 2 register (see Section 7.8.15). TLPs received that contain more End-End TLP Prefixes than are supported by a Function must be handled as Malformed TLPs. AER logging (if supported) occurs as specified in section 6.2.4.4. This is a reported error associated with the Receiving Port (see Section 6.2).
- Switches must support forwarding of TLPs with up to 4 End-End TLP Prefixes if the End-End TLP Prefix Supported bit is Set.
- Root Complexes must support forwarding of TLPs with up to Max End-End TLP Prefixes if the End-End TLP Prefix Supported bit is Set in both the Ingress and Egress Ports and forwarding of TLPs is supported between those Ports.[3]
- All End-End TLP Prefixes are protected by ECRC if the underlying TLP is protected by ECRC.
- It is an error to receive a TLP with an End-End TLP Prefix by a Receiver that does not support End-End Prefixes. A TLP in violation of this rule handled as a Malformed TLP. This is a reported error associated with the Receiving Port (see Section 6.2)
- Software should ensure that TLPs containing End-End TLP Prefixes are not sent to components that do not support them. Components where the Extended Fmt Field Supported bit is Clear may misinterpret TLPs containing TLP Prefixes.
- If one Function of an Upstream Port has the End-End TLP Prefix Supported bit Set, all Functions of that Upstream Port must handle the receipt of a Request addressed to them

[3] Root Port indication of End-End TLP Prefix Supported does not imply any particular level of peer-to-peer support by the RC, or that peer-to-peer traffic is supported at all (see Section 2.2.10.2.2).

that contains an unsupported End-End TLP Prefix type as an Unsupported Request. This is a reported error associated with the Receiving Port (see Section 6.2).

- If one Function of an Upstream Port has the End-End TLP Prefix Supported bit Set, all Functions of that Upstream Port must handle the receipt of a Completion addressed to them that contains an unsupported End-End TLP Prefix type as an Unexpected Completion. This is a reported error associated with the Receiving Port (see Section 6.2).
- For routing elements, the End-End TLP Prefix Egress Blocking bit in each Egress Port determines whether TLPs containing End-End TLP Prefixes can be transmitted via that Egress Port (see Section 7.8.16). If forwarding is blocked the entire TLP is dropped and a TLP Prefix Blocked Error is reported. If the blocked TLP is a Non-Posted Request, the Egress Port returns a Completion with Unsupport Request Completion Status. The TLP Prefix Blocked Error is a reported error associated with the Egress Port (see Section 6.2).
- For routing elements where Multicast is enabled (see Section <tbd>), End-End TLP Prefixes are replicted in all Multicast copies of a TLP. TLP Prefix Egress Blocking of Multicast packets is performed independently at each Egress Port.

Note to Editor: Multicast is covered by another ECN. Please correct the <tbd> reference to point to that ECN when incorporating this change into the base specification.

2.2.10.2.1 Vendor Defined End-End TLP Prefix

As described in Table 2-yy, Types VendPrefixE0 and VendPrefixE1 are reserved for usage as Vendor Defined End-End TLP Prefixes. To maximize interoperability and flexibility the following rules are applied to such prefixes:

- Components must not send TLPs containing Vendor Defined End-End TLP Prefixes unless this has been explicity enabled (using vendor specific mechanisms).
- It is recommended that components be configurable (using vendor specific mechanisms) to use either of the two Vendor Defined End-End TLP Prefix encodings. Doing so allows two different Vendor Defined End-End TLP Prefixes to be in use simultaneously within a single PCIe topology while not requiring that every source understand the ultimate destination of every TLP it sends.

2.2.10.2.2 Root Ports with End-End TLP Prefix Supported

Support for peer-to-peer routing of TLPs containing End-End TLP Prefixes between Root Ports is optional and implementation dependent. If an RC supports End-End TLP Prefix routing capability between 2 or more Root Ports, it must indicate that capability in each associated Root Port via the End-End TLP Prefix Supported bit in the Device Capabilities 2 register.

An RC is not required to support End-End TLP Prefix routing between all pairs of Root Ports that have the End-End TLP Prefix Supported bit Set. A Request with End-End TLP Prefixes that would require routing between unsupported pairs of Root Ports must be handled as an Unsupported Request (UR). A Completion with End-End TLP Prefixes that would require routing between unsupported pairs of Root Ports must be handled as an Unexpected Completion (UC). In both cases, this error is reported by the "sending" Port.

The End-End TLP Prefix Supported bit must be Set for any Root Port that supports forwarding of TLPs with End-End TLP Prefixes initiated by host software or Root Complex

Internal Endpoints. The End-End TLP Prefix Supported bit must be Set for any Root Ports that support forwarding of TLPs with End-End TLP Prefixes received on their Ingress Port to Root Complex Integrated Endpoints.

All Root Ports with the End-End TLP Prefix Supported bit Set must have the same value for the Max End-End TLP Prefixes field in the Device Capabilities 2 register.

Modify section 2.3 as follows

2.3. Handling of Received TLPs

This section describes how all Received TLPs are handled when they are delivered to the Receive Transaction Layer from the Receive Data Link Layer, after the Data Link Layer has validated the integrity of the received TLP. The rules are diagramed in the flowchart shown in Figure 2-21.

- Values in Reserved fields must be ignored by the Receiver.
- If the value in the Fmt field indicates the presence of at least one TLP Prefix
  - Detect if additional TLP Prefixes are present in the header by checking the Fmt field in the first byte of subsequent DWORDs until the Fmt field does not match that of a TLP Prefix.
  - Handle all received TLP Prefixes according to TLP Prefix Handling Rules (see section 2.2.10).
- If the Extended Fmt Field Supported bit is Set, Received TLPs which use encodings of Fmt and Type that are reserved are Malformed TLPs (see Table 2-1 and Table 2-3).
  - This is a reported error associated with the Receiving Port (see Section 6.2)
- If the Extended Fmt Field Supported bit is Clear, processing of Received TLPs that have Fmt[2] Set is undefined.[4]
- All Received TLPs with Fmt[2] Clear and which use undefined Type field values are Malformed TLPs.
  - This is a reported error associated with the Receiving Port (see Section 6.2)
- All Received Malformed TLPs must be discarded.
  - Received Malformed TLPs that are ambiguous with respect to which buffer to release or are mapped to an uninitialized Virtual Channel must be discarded without updating Receiver Flow Control information.
  - All other Received Malformed TLPs must be discarded, optionally not updating Receiver Flow Control Information.

Modify section 2.6.1 as follows

2.6.1. Flow Control Rules

[4] An earlier version of this specification reserved the bit now defined for Fmt[2].

In this and other sections of this specification, rules are described using conceptual "registers" that a device could use in order to implement a compliant implementation. This description does not imply or require a particular implementation and is used only to clarify the requirements.

- Flow Control information is transferred using Flow Control Packets (FCPs), which are a type of DLLP (see Section 3.4)
- The unit of Flow Control credit is 4 DW for data
- For headers, the unit of Flow Control credit is one maximum-size header plus TLP digest
  - The unit of Flow Control credit for Receivers that support End-End TLP Prefix is sum of one maximum-size Header, TLP Digest and maximum number of End-End TLP Prefixes permitted in a TLP.
  - The management of Flow Control for Receivers that support Local TLP Prefix is dependent on the Local TLP Prefix type.
- Each Virtual Channel has independent Flow Control

6.2.3.2.3 Error Pollution

Add the following item to the error precedence list.

- Malformed TLP
- AtomicOp Egress Blocked
- TLP Prefix Blocked
- ACS Violation
- MC Blocked TLP
- Unsupported Request (UR), Completer Abort (CA), or Unexpected Completion44

Add section 6.2.4.4.

6.2.4.4 TLP Prefix Logging

For any device Function that supports both TLP Prefixes and Advanced Error Reporting the TLP Prefixes associated with the TLP in error are recorded in the TLP Prefix Log register according to the same rules as the Header Log register (such that both the TLP Prefix Log and Header Log registers always correspond to the error indicated in the First Error Pointer register, when the First Error Pointer register is valid).

The TLP Prefix Log Present bit (see Section 7.10.7) indicates that the TLP Prefix Log Register (see Section 7.10.12) contains information.

Only End-End TLP Prefixes are logged by AER. Logging of Local TLP Prefixes may occur elsewhere using Prefix specific mechanisms.[5]

[5] For Example, errors involving MR-IOV TLP Prefixes are logged in MR-IOV structures and are not logged in the AER Capability.

End-End TLP Prefixes are logged in the TLP Prefix Log Register. The underlying TLP Header is logged in the Header Log Register subject to two exceptions:

- If the Extended Fmt Field Supported bit is Set (see Section 7.8.15), a Function that does not support TLP Prefixes and receives a TLP containing a TLP Prefix will signal Malformed TLP and the Header Log Register will contain the first 4 DWs of the TLP (TLP Prefixes followed by as much of the TLP Header as will fit).
- A Function that receives a TLP containing more End-End TLP Prefixes than are indicated by the Function's Max End-End TLP Prefixes field must handle the TLP as a Malformed TLP and store the first overflow End-End TLP Prefix in the 1st DW of the Header Log register with the remainder of the Header Log register being undefined.

Modify section 6.2.6 as indicated

6.2.6 Error Message Controls

Modify Table 6-4 Transport Layer Error List, Malformed TLP entry to include references to Sections 2.2.10, 2.2.10.1, 2.2.10.2, 2.3 and6.2.4.4

Add the following item to Table 6-4 Transport Layer Error List

TLP Prefix Blocked	*Egress Port*: Send ERR_NONFATAL to Root Complex or ERR_COR for the Advisory Non-Fatal Error case described in Section 6.2.3.2.4.1. Log the header of the TLP that encountered the error.	Section 2.2.10.2

7.8.15. Device Capabilities 2 Register (Offset 24h)

The following fields are added to Device Capabilities 2:

Table 7-xx Device Capabilities 2 Register

Bit Location	Register Description	Attributes
20	**Extended Fmt Field Supported** – If Set, the Function supports the 3 bit definition of the Fmt field. If Clear, the Function supports a 2 bit definition of the Fmt field. See Section 2.2. Must be Set for Functions that support End-End TLP Prefixes. All Functions in an Upstream Port must have the same value for this bit. Each Downstream Port of a component may have a different value for this bit. It is strongly recommended that Functions support the 3 bit definition of the Fmt field.	RO
21	**End-End TLP Prefix Supported** – Indicates whether End-End TLP Prefix support is offered by a Function. Values are: 0b No Support 1b Support is provided to receive TLPs containing End-End TLP Prefixes. This bit is HwInit for Root Ports and is RO for all other Functions.	RO / HwInit (see description)

Bit Location	Register Description	Attributes
	All Ports of a Switch must have the same value for this bit.	
23:22	**Max End-End TLP Prefixes**– Indicates the maximum number of End-End TLP Prefixes supported by this Function. TLPs received by this Function that contain more End-End TLP Prefixes than are supported must be handled as Malformed TLPs (see Section 2.2.10.2). Values are: 01b 1 End-End TLP Prefix 10b 2 End-End TLP Prefixes 11b 3 End-End TLP Prefixes 00b 4 End-End TLP Prefixes If End-End TLP Prefix Supported is Clear, this field is RsvdP. This field is HwInit for Root Ports and is RO for all other Functions. All Root Ports that have the End-End TLP Prefix Supported bit Set must contain the same value for this field. For Switches where End-End TLP Prefix Supported is Set, this field must be 00b indicating support for up to 4 End-End TLP Prefixes (see Section 2.2.10.2).	RO/ HwInit (see description)

7.8.16. Device Control 2 Register (Offset 28h)

The following fields are added to Device Control 2:

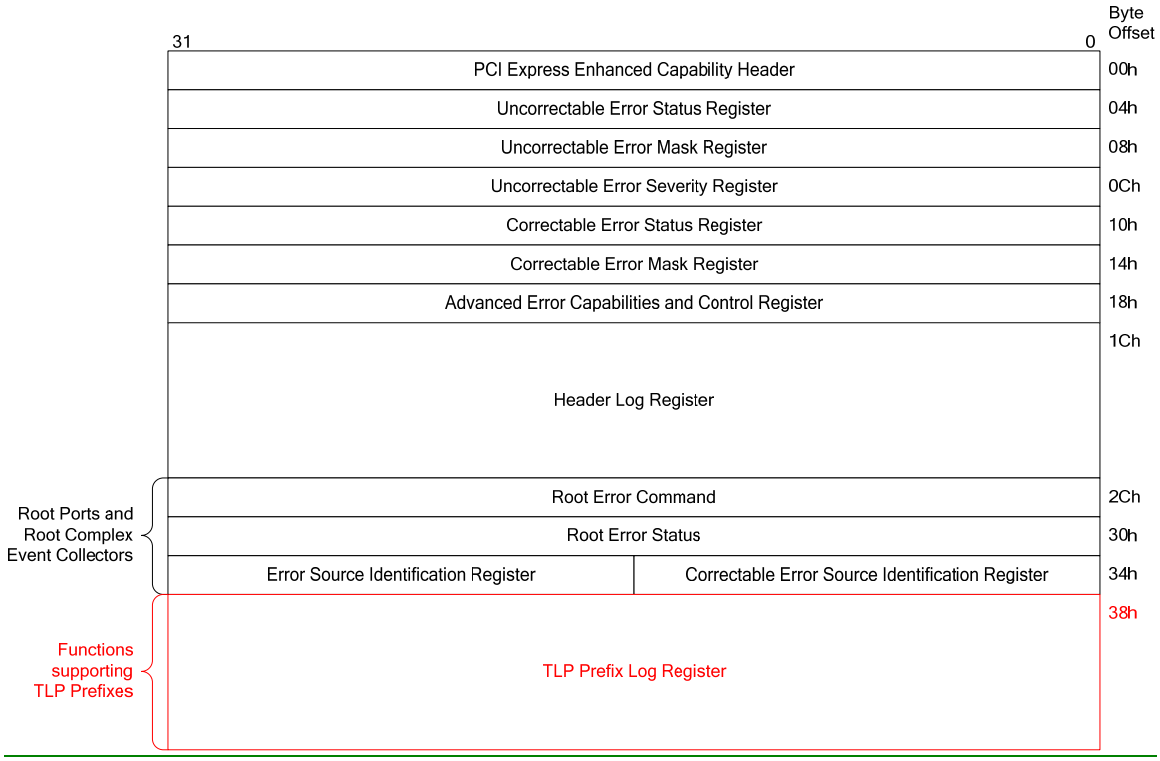
Table 7-xx TLP Prefix Control Fields

Bit Location	Register Description	Attributes
15	**End-End TLP Prefix Blocking** – Controls whether the routing function is permitted to forward TLPs containing an End-End TLP Prefix. Values are: 0b Forwarding Enabled – Function is permitted to send TLPs with End-End TLP Prefixes. 1b Forwarding Blocked – Function is not permitted to send TLPs with End-End TLP Prefixes. This bit affects TLPs that exit the Switch / Root Complex using the associated Port. It does not affect TLPs forwarded internally within the Switch / Root Complex. It does not affect TLPs that enter through the associated Port, that originate in the associated Port or originate in a Root Complex Integrated Device integrated with the associated Port. As described in Section 2.2.10.2, blocked TLPs are reported by the TLP Prefix Blocked Error. The default value of this bit is 0b. This bit is hardwired to 1b in Root Ports that support End-End TLP Prefixes but do not support forwarding of End-End TLP Prefixes. This bit is applicable to Root Ports and Switch Ports where the End-End TLP Prefix Supported bit is Set. This bit is not applicable and is RsvdP in all other	RW (see description)

Bit Location	Register Description	Attributes
	cases.	

7.10 Advanced Error Reporting Capability

Add TLP Prefix Log to Figure 7-30 (for the ECN, this is shown in red).



7.10.1 Advanced Error Reporting Enhanced Capability Header (Offset 00h)

Change Capability Version to 2h.

7.10.2. Uncorrectable Error Status Register (Offset 04h)

Add the following fields to Figure 7-32 and Table 7-29.

Bit Location	Register Description	Attributes	Default
25	**TLP Prefix Blocked Error (Optional)**	RW1CS	0b

7.10.3. Uncorrectable Error Mask Register (Offset 08h)

Add the following fields to Figure 7-33 and Table 7-30.

Bit Location	Register Description	Attributes	Default
25	**TLP Prefix Blocked Error Mask (Optional)**	RWS	0b

7.10.4. Uncorrectable Error Severity Register (Offset 0Ch)

Add the following fields to Figure 7-34 and Table 7-31.

Bit Location	Register Description	Attributes	Default
25	**TLP Prefix Blocked Error Severity (Optional)**	RWS	0b

7.10.7 Advanced Error Capabilities and Control Register (Offset 18h)

Add the following fields to Figure 7-37 and Table 7-34.

Bit Location	Register Description	Attributes
11	**TLP Prefix Log Present** – If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information. If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. Default value of this bit is 0. This bit is RsvdP if the End-End TLP Prefix Supported bit is Clear.	ROS

7.10.8 Header Log Register (Offset 1Ch)

Add paragraph as indicated.

The Header Log register captures the header for the TLP corresponding to a detected error; refer to Section 6.2 for further details. Section 6.2 also describes the conditions where the packet header is logged. This register is 16 bytes and adheres to the format of the headers defined throughout this specification.

The header is captured such that the fields of the header read by software in the same way the headers are presented in this document, when the register is read using DW accesses. Therefore, byte 0 of the header is located in byte 3 of the Header Log register, byte 1 of the header is in byte 2 of the Header Log register and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log register are used and values in bytes 12 through 15 are undefined.

In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information. See Section 6.2.4.4 for details.

Figure 6-1 details allocation of register fields in the Header Log register;

Table **6-2** provides the respective bit definitions.

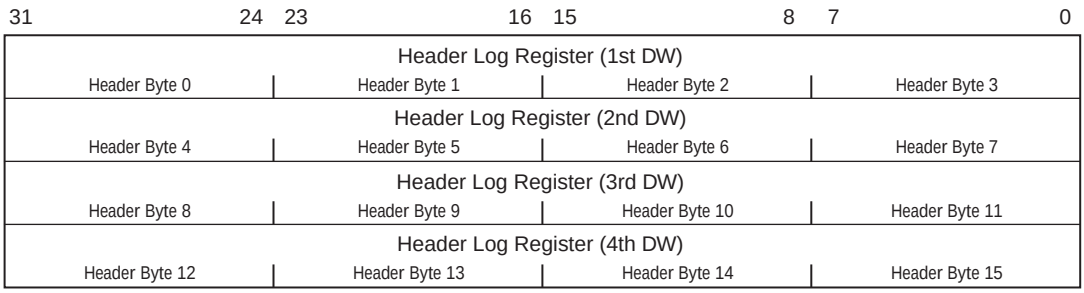


Figure 6-1: Header Log Register

Table 6-2: Header Log Register

Bit Location	Register Description	Attributes	Default
127:0	Header of TLP associated with error	ROS	0

Add section 7.10.12.

7.10.12 TLP Prefix Log Register (Offset 38h)

The TLP Prefix Log register captures the End-End TLP Prefix(s) for the TLP corresponding to the detected error; refer to Section 6.2 for further details. The TLP Prefix Log register is only meaningful when the TLP Prefix Log Present bit is Set (see Section 7.10.7).

The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log register, byte 1 of a TLP Prefix is located in byte 2 and so forth.

The First TLP Prefix Log Register contains the first End-End TLP Prefix from the TLP (see Section 6.2.4.4). The Second TLP Prefix Log register contains the second End-End TLP Prefix and so forth. If the TLP contains fewer than four End-End TLP Prefixes, the remaining TLP Prefix Log Registers contain zero. A TLP that contains more End-End TLP Prefixes than are indicated by the Function's Max End-End TLP Prefixs field must be handled as a Malformed TLP (see Section 2.2.10.2). To allow software to detect this condition, the supported number of End-End TLP Prefixes are logged in this register, the first overflow End-End TLP Prefix is logged in the first DW of the Header Log register and the remaining DWs of the Header Log Register are undefined (see Section 6.2.4.4).

The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero. For example, if a Functions, Max End-End TLP Prefixes field contains 10b (indicating 2 DW of buffering) then the third and fourth TLP Prefix Log Registers are hardwired to zero.

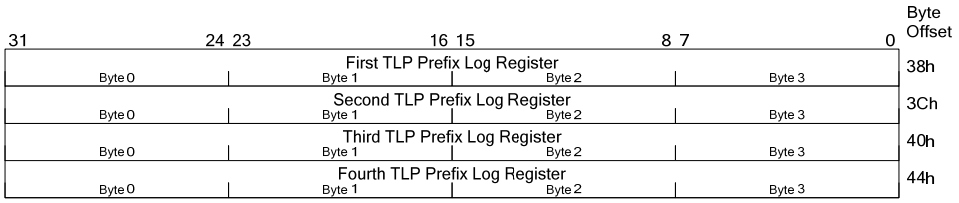


Figure 7-xx TLP Prefix Log Register

Table 7-xx TLP Prefix Log Register

Bit Location	Register Description	Attributes	Default
127:0	**TLP Prefix Log**	ROS	0