PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	ID-Based Ordering
DATE:	January 16, 2008, updated 29 May 2008
AFFECTED DOCUMENTS:	PCI Express Base Specification version 2.0 Errata to 2.0 Base Spec Section 2.4.1
SPONSORS:	Intel, Advanced Micro Devices, Hewlett-Packard, IBM

Part I

1. Summary of the Functional Changes

This ECN proposes to add a new ordering attribute which devices may optionally support to provide enhanced performance for certain types of workloads and traffic patterns. The new ordering attribute relaxes ordering requirements between unrelated traffic by comparing the Requester/Completer IDs of the associated TLPs.

This ECN impacts Endpoint devices, Root Ports and Switches that choose to implement the new optional feature.

2. Benefits as a Result of the Changes

PCI Express (PCIe) Conventional Ordering rules (CO) have been written to preserve the producer consumer programming model and to prevent deadlocks in PCIe-based systems (potentially including bridges to PCI/PCI-X). The producer/consumer model places restrictions on re-ordering of transactions (Table 2-24: A2a, B2a, D2a) which have implications on performance, especially read-latency.

ID-Based Ordering provides opportunity independent request streams to bypass another congested stream, yielding performance improvement.

3. Assessment of the Impact

ID-Based Ordering is optional normative functionality that is applicable to endpoint functions, switches and root-ports. The summary impact to specification is noted below

1. The functionality is enabled in endpoints through configuration bits
2. A reporting capability for Switches and RCs that support peer to peer
3. A new ordering attribute bit in TLP header

4. Analysis of the Hardware Implications

ID-Based Ordering requires new hardware and is therefore optional normative. Hardware that is not ID-Based Ordering capable will default to Conventional Ordering and thus be fully compatible.

5. Analysis of the Software Implications

ID-Based Ordering requires new software to enable this functionality and thus is optional normative. Software that does not comprehend the new functionality will interoperate with ID-Based Ordering capable hardware per the existing PCI Express Base Specification.

Part II

Detailed Description of the change

Modify Section 1.5.1 as shown

Every request packet requiring a response packet is implemented as a split transaction. Each packet has a unique identifier that enables response packets to be directed to the correct originator. The packet format supports different forms of addressing depending on the type of the transaction (Memory, I/O, Configuration, and Message). The Packets may also have attributes such as No Snoop, Relaxed Ordering and ID-Based Ordering (IDO).

Modify Section 1.5.4.1 as shown

1.5.4.1 Transaction Layer Services

…

Ordering rules:

- PCI/PCI-X compliant producer consumer ordering model
- Extensions to support R~~r~~elaxed O~~o~~rdering
- Extensions to support ID-Based Ordering

Modify Section 2.2.1 as shown

2.2.1 Common Packet Header Fields

All Transaction Layer Packet (TLP) headers contain the following fields

……

- TC[2:0] – Traffic Class (see Section <>) – bits [6:4] of byte 1
- Attr[2] - Attribute (see Section 2.2.6.3) bit 2 of byte 1
- Attr[1:0] – Attributes (see Section <>) – bits [5:4] of byte 2

Modify Figure 2-4 as shown

	+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			Rsvd	Attr		Rsvd	TD	EP	Attr		AT				Length							

Modify Figure 2-5 as shown

	+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 x1		Type					R	TC			Rsvd	Attr	Rsvd		TD	EP	Attr		AT				Length							
Byte 4	{Fields in bytes 4 through 7 depend on type of Request}																															
Byte8	Address[63:32]																															
Byte12	Address[31:2]																														R	

Modify Figure 2-6 as shown

	Figure 2-6																															
	+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 x0		Type					R	TC			Rsvd	Attr	Rsvd		TD	EP	Attr		AT				Length							
Byte 4	{Fields in bytes 4 through 7 depend on type of Request}																															
Byte12	Address[31:2]																														R	

Modify Section 2.2.6.3 as shown:

The Attributes field is used to provide additional information that allows modification of the default handling of Transactions. These modifications apply to different aspects of handling the Transactions within the system, such as:

- ❑ Ordering
- ❑ Hardware coherency management (snoop)

Note that attributes are hints that allow for optimizations in the handling of traffic. Level of support is dependent on target applications of particular PCI Express peripherals and platform building blocks. Refer to PCI-X 2.0 for additional details regarding these attributes. Note that attribute bit 2 is not adjacent to bits 1 and 0 (see figure 2-13 and 2-14).

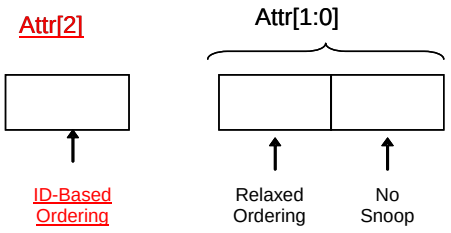


Figure 2-12: Attributes Field of Transaction Descriptor

Modify Section 2.2.6.4 as shown:

2.2.6.4. Relaxed Ordering and ID-Based Ordering Attributes

Table 2-9 defines the states of the Relaxed Ordering and ID-Based Ordering attribute fields. These attributes are ~~This attribute is~~ discussed in Section <>. Note that Relaxed Ordering and ID-Based Ordering attributes are not adjacent in location (see figure 2-4).

Table 2-9: Ordering Attributes

Attribute bit [2]	Attribute bit [1]	Ordering Type	Ordering Model
0	0	Default Ordering	PCI Strongly Ordered Model
0	1	Relaxed Ordering	PCI-X Relaxed Ordering Model
1	0	ID-Based Ordering	Independent ordering based on Requester/Completer ID
1	1	Relaxed Ordering + ID-Based Ordering	Logical "OR" of Relaxed Ordering and IDO

~~This~~ A~~a~~ttribute bit [1] is not applicable and must be set to 0b for Configuration Requests, I/O Requests, Memory Requests that are Message Signaled Interrupts, and Message Requests (except where specifically permitted).

Attribute bit [2], IDO, is reserved for Configuration Requests and I/O Requests. IDO is not reserved for all Memory Requests, including Message Signaled Interrupts. IDO is not reserved for , Message Requests, unless specifically prohibited. A Requester is permitted to set IDO only if the IDO Request Enable bit in the Device Control 2 Register is set.

The value of the IDO bit must not be considered by Receivers when determining if a TLP is a Malformed Packet.

A Completer is permitted to set IDO only if the IDO Completion Enable bit in the Device Control 2 Register is set. It is not required to copy the value of IDO from the Request into the Completion(s) for that Request. If the Completer has IDO enabled, it is recommended that the Completer set IDO for all Completions, unless there is a specific reason not to (see Appendix E)

A Root Complex that supports forwarding TLPs peer to peer between Root Ports is not required to preserve the IDO bit from the Ingress to Egress Port.

Modify Figure 2-13 as shown

	+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
Byte 0	R	Fmt		Type					R	TC			Rsvd	Attr	Rsvd		TD	EP	Attr		AT				Length							
Byte 4	Requestor ID																Tag								Last DW BE				1st DW BE			
Byte 8	Address[63:32]																															
Byte 12	Address[31:2]																								R							

Modify Figure 2-14 as shown

	+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
Byte 0	R	Fmt		Type					R	TC			Rsvd	Attr	Rsvd		TD	EP	Attr		AT				Length							
Byte 4	Requestor ID																Tag								Last DW				1st DW			
Byte 8	Address[31:2]																								R							

Modify section 2.2.7 as follows

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

……

For I/O Requests, the following rules apply:

- ❑ I/O Requests route by address, using 32-bit Addressing (see <>)
- ❑ I/O Requests have the following restrictions:
  - TC[2:0] must be 000b
  - Attr[2] is reserved
  - Attr[1:0] must be 00b
  - AT[1:0] must be 00b

…..

For Configuration Requests, the following rules apply:

- ❑ Configuration Requests route by ID, and use a 3 DW header

……

- ❑ Configuration Requests have the following restrictions:
  - TC[2:0] must be 000b
  - Attr[2] is reserved
  - Attr[1:0] must be 00b

Modify Figure 2-15 as shown

	+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
Byte 0	R	Fmt	x 0	Type					R	TC	000		Rsvd	Attr 0	Rsvd		T D	E P	Attr	00	AT	00	Length		0	0	0	0	0	0	0	1
Byte 4	Requestor ID																Tag								BE 0000				1st DW			
Byte 8	Address[31:2]																														R	

Modify Figure 2-16 as shown

	+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
Byte 0	R	Fmt x 0		Type					R	TC 000			Rsvd	Attr 0		Rsvd	TD	EP	Attr 00		AT 00		Length 0 0 0 0 0 0 0 0 0 1										
Byte 4	Requestor ID																Tag								BE 0000				1st DW			
Byte 8	Address[31:2]																															R

Modify Section 2.2.8 as shown

2.2.8 Message Request Rules

This document defines the following groups of Messages:

- INTx Interrupt Signaling
- ….

The following rules apply to all Message Requests. Additional rules specific to each type of Message follow.

- All Message Requests ..

…….

- The Attr[2]field is not reserved unless specifically indicated as reserved
- Except as noted, the Attr[1:0] field is reserved.
- AT[1:0] must be 00b.
- Except as noted, bytes 8 through 15 are reserved.
- Message Requests are posted and do not require Completion.

...

Modify Figure 2-17 as shown

	+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
Byte 0	R	Fmt		Type					R	TC			Rsvd	Attr		Rsvd	TD	EP	Attr 00		AT 00				Length							
Byte 4	Requestor ID																Tag								Message Code							
Byte 8	{Except as noted,bytes 8 through 15 are reserved.}																															
Byte 12																																

Modify Figure 2-18 as shown:

	+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
Byte 0	R	Fmt x 1		Type					R	TC			Rsvd	Attr		Rsvd	TD	EP	Attr		AT				Length							
Byte 4	Requestor ID													Function			Tag								Message Code - Vendor Defined							
Byte 8	Bus Number								Device Number					Number			Vendor Id															
	Reserved																															
Byte 12	For Vendor Definition																															

Modify section 2.2.8.6 as shown

2.2.8.6 Vendor_Defined Messages

The Vendor_Defined Messages ……

- A data payload may be included with either type of Vendor_Defined Message (TLP type is Msg if no data payload is included and MsgD if a data payload is included)
- For both types of Vendor_Defined Messages, the Attr[1:0] and Attr [2] fields are~~is~~ not reserved.
- Messages defined by different vendors or by PCI-SIG are distinguished by the value in the Vendor ID field.

Modify figure 2-19 as shown

	+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			Rsvd	Attr		Rsvd	TD	EP	Attr		AT				Length							
Byte4	Completer ID																Compl Stat			BCM	Byte Count											
Byte8	Requestor ID																Tag								R	Lower Address						

Modify Table 2-24 and following text as shown

(NOTE: These changes build upon errata to this section; the highlighted text below is from the AtomicOps ECN, and is not part of this change, but is included to show how this change and the AtomicOps ECN will ultimately be merged):

Row Pass Column?		Posted Request (Col 2)	Non-Posted Request		Completion (Col 5)
			Read Request (Col 3)	NPR with Data (Col 4)	
Posted Request (Row A)		a) No b) Y/N	Yes	Yes	a) Y/N b) Yes
Non-Posted Request	**Read Request** (Row B)	a) No b) Y/N	Y/N	Y/N	Y/N
	NPR with Data (Row C)	a) No b) Y/N	Y/N	Y/N	Y/N
Completion (Row D)		a) No b) Y/N	Yes	Yes	a) Y/N b) No

Explanation of the row and column headers in Table 2-24:

- A **Posted Request** is a Memory Write Request or a Message Request.
- A **Read Request** is a Configuration Read Request, an I/O Read Request, or a Memory Read Request.
- An **NPR** (Non-Posted Request) **with Data** is a Configuration Write Request or an I/O Write Request.
- A **Non-Posted Request** is a Read Request or an NPR with Data.

Explanation of the entries in Table 2-24:

A2a	A Posted Request must not pass another Posted Request unless A2b applies.
A2b	A Posted Request with RO[1] Set is permitted to pass another Posted Request[2]. A Posted Request with IDO Set is permitted to pass another Posted Request if the two Requester IDs are different.
A3, A4	A Posted Request must be able to pass Non-Posted Requests to avoid deadlocks.
A5a	A Posted Request is permitted to pass a Completion, but is not required to be able to pass Completions unless A5b applies.

[1] In this section, "RO" is an abbreviation for the Relaxed Ordering Attribute field.

[2] Some usages are enabled by not implementing this passing – refer to the "No RO-enabled PR-PR Passing" bit in Section 7.8.15.

A5b	Inside a PCI Express to PCI/PCI-X Bridge whose PCI/PCI-X bus segment is operating in conventional PCI mode, for transactions traveling in the PCI Express to PCI direction, a Posted Request must be able to pass Completions to avoid deadlock.
B2a	A Read Request must not pass a Posted Request unless B2b applies.
B2b	A Read Request with IDO Set is permitted to pass a Posted Request if the two Requester IDs are different.
C2a	An NPR with Data must not pass a Posted Request unless C2b applies.
C2b	An NPR with Data and with RO Set[3] is permitted to pass a Posted Request. An NPR with Data and with IDO Set is permitted to pass a Posted Request if the two Requester IDs are different.
B3, B4, C3, C4	A Non-Posted Request is permitted to pass another Non-Posted Request.
B5, C5	A Non-Posted Request is permitted to pass a Completion.
D2a	A Completion must not pass a Posted Request unless D2b applies.
D2b	An I/O or Configuration Write Completion[4] is permitted to pass a Posted Request. A Completion with RO Set is permitted to pass a Posted Request. A Completion with IDO Set is permitted to pass a Posted Request if the Completer ID of the Completion is different from the Requester ID of the Posted Request.
D3, D4	A Completion must be able to pass Non-Posted Requests to avoid deadlocks.
D5a	Completions with different Transaction IDs are permitted to pass each other.
D5b	Completions with the same Transaction ID must not pass each other. This ensures that multiple Completions associated with a single Memory Read Request will remain in ascending address order.

[3] Note: Not all NPR with Data transactions are permitted to have RO Set.

[4] Note: Not all components can distinguish I/O and Configuration Write Completions from other Completions. In particular, routing elements not serving as the associated Requester or Completer generally can't make this distinction. A component must not apply this rule for I/O and Configuration Write Completions unless it is certain of the associated Request type.

Modify Section 2.2.9. as shown

- Completion headers must supply the same values for the Requester ID, Tag, ~~Attribute,~~ and Traffic Class as were supplied in the header of the corresponding Request.
- Completion headers must supply the same values for the Attribute as were supplied in the header of the corresponding Request, except as explicitly allowed when IDO is used (see Section 2.2.6.4.).

Modify Section 6.1.4 as shown

IMPLEMENTATION NOTE

Per Vector Masking with MSI/MSI-X

Devices and drivers that use MSI or MSI-X have the challenge of coordinating exactly when

…..

A Legacy Endpoint that implements MSI is required to support either the 32-bit or 64-bit Message Address ……

The Requester of an MSI/MSI-X transaction must set the No Snoop and Relaxed Ordering attributes of the Transaction Descriptor to 0b. A Requester of an MSI/MSI-X transaction is permitted to set the ID-Based Ordering (IDO) attribute if use of the IDO attribute is enabled.

Note that, unlike INTx emulation Messages, MSI/MSI-X transactions are not restricted to TC0 traffic class.

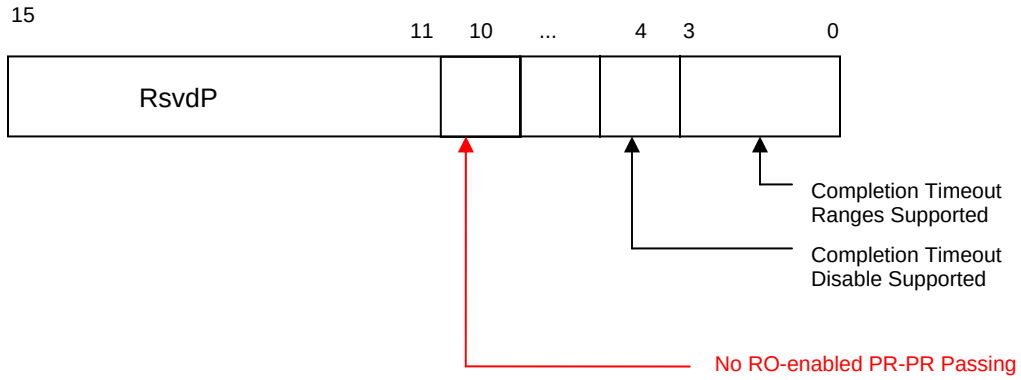
Modify Section 6.4 as shown

…..

However, in all cases the device hardware and software implementers should thoroughly understand the ordering rules described in Section <>. This is especially true if the Relaxed Ordering or ID-Based Ordering ~~flag is~~ attributes are set for any Requests initiated by the device.

Modify Section 7.8.15 as shown

7.8.15. Device Capabilities 2 Register (Offset 24h)



...

...		
10	**No RO-enabled PR-PR Passing** – If this bit is Set, the routing element never carries out the passing permitted by Table 2-24 entry A2b that's associated with the Relaxed Ordering Attribute field being Set. This bit applies only for Switches and RCs that support peer to peer traffic between Root Ports. This bit applies only to Posted Requests being forwarded through the Switch or RC and does not apply to traffic originating or terminating within the Switch or RC itself. All ports on a Switch or RC must report the same value for this bit. For all other functions, this bit must be 0b.	HwInit

...

Add Implementation Note at end of section:

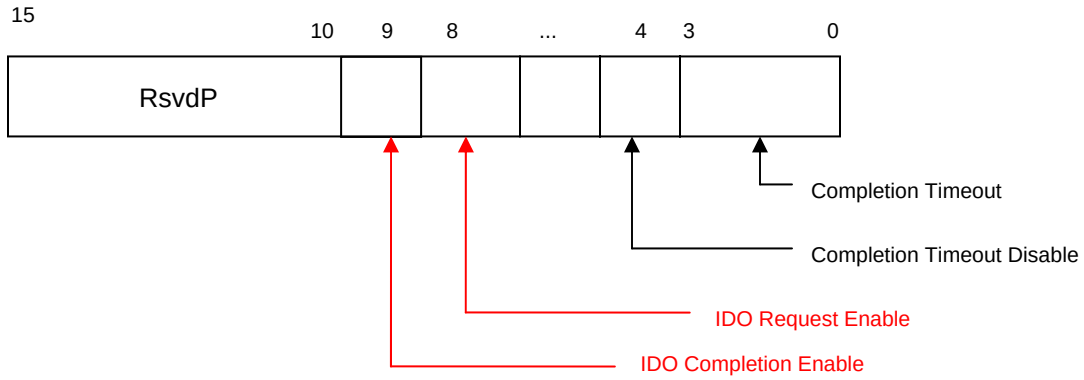
IMPLEMENTATION NOTE

Use of No RO-enabled PR-PR Passing bit

The "No RO-enabled PR-PR Passing" bit allows platforms to utilize PCI-Express switching elements on the path between a requester and completer for requesters that could benefit from a slightly less relaxed ordering model. An example is a device that cannot ensure that multiple overlapping posted writes to the same address are outstanding at the same time. The method by which such a device is enabled to utilize this mode is out of scope of this specification.

Modify Section 7.8.16 as shown

7.8.16. Device Control 2 Register (Offset 28h)



...

...		
8	**IDO Request Enable** – If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Requests it initiates (see Section 2.2.6.3. and Section 2.4). Endpoints, including RC Integrated Endpoints, and Root Ports are permitted to implement this capability. A Function is permitted to hardwire this bit to 0b if it never sets the IDO attribute in Requests. Default value of this bit is 0b.	RW
9	**IDO Completion Enable** – If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Completions it returns (see Section 2.2.6.3. and Section 2.4). Endpoints, including RC Integrated Endpoints, and Root Ports are permitted to implement this capability. A Function is permitted to hardwire this bit to 0b if it never sets the IDO attribute in Requests. Default value of this bit is 0b.	RW

Add new Appendix E.

E. ID-Based Ordering Usage

E.1. Introduction

ID-Based Ordering (IDO) is a mechanism that permits certain ordering restrictions to be relaxed as a means to improve performance. IDO permits certain TLPs to pass other TLPs in cases where otherwise such passing would be forbidden. The passing permitted by IDO is not required for proper operation (e.g., deadlock avoidance); it is only a means of improving performance.

For discussing IDO, it's useful to introduce the concept of a "TLP stream", which is a set of TLPs that all have the same originator[5]. For several important cases where TLP passing is normally forbidden, IDO permits such passing to occur if the TLPs belong to different TLP streams.

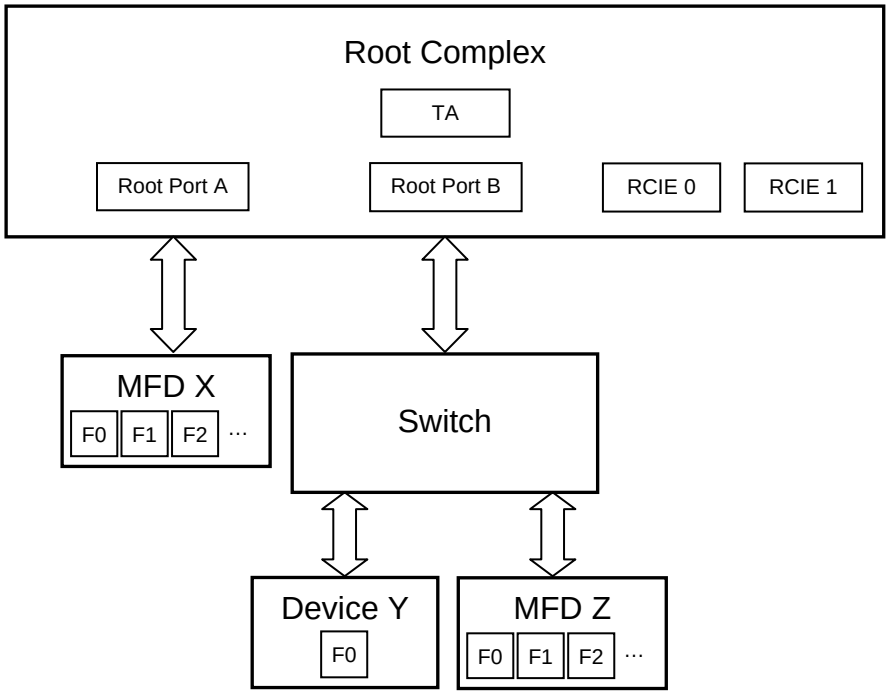


Figure E-1: Reference Topology for IDO Use

Figure E-1 shows a reference topology. The reference topology is not intended to discourage the use of IDO with other topologies, but rather to provide specific examples for discussion.

Devices X and Z are multi-Function devices (MFDs); device Y is a single-Function device. The RCIEs are Root Complex Integrated Endpoint Functions, and might or might not be part of the same Device. We'll assume that one or more Functions are using the Translation Agent (TA) in the Root Complex (RC).

Referring to the ordering table and descriptions in Section 2.4.1, having the IDO bit set in a Posted Request, Non-Posted Request, or Completion TLP permits that TLP to pass a Posted Request TLP if the two TLPs belong to different TLP streams. In the following examples, DMAR and DMAW stand for Direct Memory Access Read and Write; PIOR and PIOW stand for Programmed I/O Read and Write.

[5] That is, the Requester IDs of Requests and the Completer IDs of Completions are all the same.

E.2. Potential Benefits with IDO Use

Here are some example potential benefits that are envisioned with IDO use. Generally IDO provides the most benefit when multiple TLP streams share a common Link and that Link becomes congested, either due to high utilization or due to temporary lack of Flow Control (FC) credit.

E.2.1. Benefits for MFD/RP Direct Connect

Here are some examples in the context of traffic between MFD X and the RC in Figure E-1.

- Posted Request passing another Posted Request: when a DMAW from F0 is stalled due to an TA miss, if IDO is set for a DMAW from F1, it is permitted within the RC for this DMAW to pass the stalled DMAW from F0.
- Non-Posted Request passing a Posted Request: when a DMAW from F0 is stalled due to an TA miss, if IDO is set for a DMAR Request from F1, it is permitted within the RC for this DMAR Request to pass the stalled DMAW from F0.
- Completion passing a Posted Request: when a DMAW from F0 is stalled due to an TA miss, if IDO is set for a PIOR Completion from F1, it is permitted within the RC for this PIOR Completion to pass the stalled DMAW from F0.

E.2.2. Benefits for Switched Environments

Here are some examples in the context of traffic within the Switch in Figure E-1.

- Non-Posted Request passing a Posted Request: when a DMAW from Device Y is stalled within the Switch due to a lack of FC credit from Root Port B, if IDO is set for a DMAR Request from MFD Z, it is permitted within the Switch for this DMAR Request to pass the stalled DMAW from Device Y. The same also holds for a DMAR Request from one Function in MFD Z passing a stalled DMAW from a different Function in MFD Z.
- Completion passing a Posted Request: when a DMAW from Device Y is stalled within the Switch due to a lack of FC credit from Root Port B, if IDO is set for a PIOR Completion from MFD Z, it is permitted within the Switch for this PIOR Completion to pass the stalled DMAW from Device Y. The same also holds for a PIOR Completion from one Function in MFD Z passing a stalled DMAW from a different Function in MFD Z.
- Posted Request passing another Posted Request: within a Switch, there is little or no envisioned benefit from having a DMAW from one TLP stream passing a DMAW from a different TLP stream. However, it is not prohibited for Switches to implement such passing as permitted by IDO.

E.2.3. Benefits for Integrated Endpoints

Here are some examples for the Root Complex Integrated Endpoints (RCIEs) in Figure E-1. The benefits are basically the same as for the MFD/RP Direct Connect case.

- Posted Request passing another Posted Request: when a DMAW from RCIE 0 is stalled due to an TA miss, if IDO is set for a DMAW from RCIE 1, it is permitted for this DMAW to pass the stalled DMAW from RCIE 0.

- Non-Posted Request passing a Posted Request: when a DMAW from RCIE 0 is stalled due to an TA miss, if IDO is set for a DMAR Request from RCIE 1, it is permitted for this DMAR Request to pass the stalled DMAW from RCIE 0.
- Completion passing a Posted Request: when a DMAW from RCIE 0 is stalled due to an TA miss, if IDO is set for a PIOR Completion from RCIE 1, it is permitted for this PIOR Completion to pass the stalled DMAW from RCIE 0.

E.2.4. IDO Use in Conjunction with RO

IDO and RO[6] are orthogonal. Certain instances of passing, e.g. a Posted Request passing another Posted Request, might be permitted by IDO, RO, or both at the same time. While IDO and RO have significant overlap for some cases, it is highly recommended that both be used whenever safely possible. RO permits certain TLP passing within the same TLP stream, which is never permitted by IDO. For traffic in different TLP streams, IDO permits control traffic to pass any other traffic, and generally it is not safe to Set RO with control traffic.

E.3. When to Use IDO

With Endpoint Functions[7], it is safe to Set IDO in all applicable TLPs originated by the Endpoint when the Endpoint is directly communicating with only one other entity, most commonly the RC. For the RC case, "directly communicating" specifically includes DMA traffic, PIO traffic, and interrupt traffic; communicating with RCIEs or communicating using P2P Root Port traffic constitutes communicating with multiple entities.

With a Root Port, there are no envisioned high-benefit use models where it is safe to Set IDO in all applicable TLPs that it originates. Use models where a Root Port Sets IDO in a subset of the applicable TLPs it originates are outside the scope of this specification.

E.4. When Not to Use IDO

E.4.1. When Not to Use IDO with Endpoints

With Endpoint Functions, it is not always safe to Set IDO in applicable TLPs it originates if the Endpoint directly communicates with multiple entities. It may be safe to Set IDO in some TLPs and not others, but such use models are outside the scope of this specification.

For example, in Figure E-1 if Device Y and MFD Z are communicating with P2P traffic and also communicating via host memory, it is not always safe for them to Set IDO in the TLPs they originate. As an example failure case, let's assume that Device Y does a DMAW (to host memory) followed by a P2P Write to MFD Z. Upon observing the P2P Write, let's assume that MFD Z then does a DMAW to the same location earlier targeted by the DMAW from Device Y. Normal ordering rules would guarantee that the DMAW from Device Y would be observed by host memory before the DMAW from MFD Z. However, if IDO is set in the DMAW from MFD Z, the RC

---

[6] In this Appendix, "RO" is an abbreviation for the Relaxed Ordering Attribute field.

[7] Endpoint Functions include PCI Express Endpoints, Legacy PCI Express Endpoints, and Root Complex Integrated Endpoints.

would be permitted to have the second DMAW pass the first, causing a different end result in host memory contents.

Synchronization techniques like performing zero-length Reads might be used to avoid such communication failures when IDO is used, but specific use models are outside the scope of this specification.

E.4.1. When Not to Use IDO with Root Ports

With Root Ports, it is not always safe to Set IDO in applicable TLPs it originates if Endpoint Functions in the hierarchy do any P2P traffic. It may be safe to Set IDO in some TLPs and not others, but such use models are outside the scope of this specification.

As an example, in Figure E-1 if Device Y and MFD Z are communicating with P2P traffic and also communicating with host software, it is not always safe for Root Port B to Set IDO in the TLPs it originates. For example, let's assume that Device Y does a P2P Write to MFD Z followed by a DMAW (to host memory). Upon observing the DMAW, let's assume that the host does a PIOW to MFD Z. Normal ordering rules would guarantee that the P2P Write from Device Y would be observed by MFD Z before the PIOW from the host. However, if IDO is set in the PIOW from the host, the Switch would be permitted to have the PIOW pass the P2P Write, ultimately having the two Writes arrive at MFD Z out of order.

IMPLEMENTATION NOTE

Requester and Completer IDs for RC-Originated TLPs

With RC implementations where the Requester ID in a PIO Request does not match the Completer ID in a DMAR Completion, this enables another potential communication failure case if IDO is Set in the Completion. For this case, if a PIOW is followed by a DMAR Completion with IDO Set, a Switch below the Root Port could permit the DMAR Completion to pass the PIOW, violating the normal ordering rule that a non-RO Read Completion must not pass Posted Requests. The PIOW and DMAR Completion would appear to belong to different TLP streams, though logically they belong to the same TLP stream. Special caution is advised in setting IDO with TLPs originating from such RCs.

E.5. Software Control of IDO Use

E.5.1. Software Control of Endpoint IDO Use

By default, Endpoints are not enabled to Set IDO in any TLPs they originate.

IMPLEMENTATION NOTE

The "Simple" Policy for IDO Use

It is envisioned that Endpoints designed primarily to communicate directly with only one other entity (e.g., the RC) may find a "simple" policy for setting IDO to be adequate. Here's the envisioned "simple" policy. If the IDO Request Enable bit is Set, the Endpoint Sets IDO in all applicable Request TLPs that it originates. If the IDO Completion Enable bit is Set, the Endpoint Sets IDO in all Completion TLPs that it originates.

It is envisioned that a software driver associated with each Endpoint will determine when it is safe for the Endpoint to set IDO in applicable TLPs it originates. A driver should be able to determine if the Endpoint is communicating with multiple other entities, and should know the Endpoint's capabilities as far as setting IDO with all applicable TLPs when enabled, versus setting IDO selectively. If a driver determines that it is safe to enable the setting of IDO, the driver can set the IDO Request Enable and/or IDO Completion Enable bits either indirectly via OS services or directly, subject to OS policy.

If an Endpoint is designed for communication models where it is not safe to utilize the "simple" policy for IDO use, the Endpoint can implement more complex policies for determining when the Endpoint sets the IDO bit. Such implementations might utilize device-specific controls that are managed by the device driver. Such policies and device-specific control mechanisms are outside the scope of this specification.

E.5.2. Software Control of Root Port IDO Use

Since there are no envisioned high-benefit "simple" use models for Root Ports setting the IDO bit with TLPs they originate, and there are known communication failure cases if Root Ports set the IDO bit with all applicable TLPs they originate, it is anticipated that Root Ports will rarely be enabled to set IDO in TLPs they originate. Such use models and policies for Root Ports setting IDO are outside the scope of this specification.