PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	Downstream Port Containment (DPC)
DATE:	Introduced: Aug 11, 2011; Last Updated: Feb 2, 2012 Final Approval by PWG: Feb 9, 2012
AFFECTED DOCUMENT:	PCI Express Base Specification Revision 3.0
SPONSORS:	Peter Onufryk, IDT; Joe Cowan, HP; Michael Krause, HP

Part I

1. Summary of the Functional Changes

This ECN defines a new error containment mechanism for Downstream Ports as well as minor enhancements that improve asynchronous card removal. Downstream Port Containment (DPC) is the automatic disabling of the Link below a Downstream Port following an uncorrectable error. This prevents the potential spread of data corruption (all TLPs subsequent to the error are prevented from propagating either Upstream or Downstream) and enables error recovery if supported by software.

This ECN defines functionality for both Switch Downstream Ports and Root Ports. A subsequent ECN is planned for defining DPC extensions, primarily for functionality that is specific to Root Ports.

2. Benefits as a Result of the Changes

As PCIe usage increases across multiple market segments, members and customers require more robust error containment and an opportunity for recovery. DPC improves PCIe error containment and allows software to recover from async removal events that are not possible with the existing PCIe specification.

3. Assessment of the Impact

The impacts are modest and only applicable to those implementations that support DPC. The ECN defines a new PCIe extended capability and associated semantics that are relatively lightweight in both hardware and software.

4. Analysis of the Hardware Implications

Hardware supporting this ECN will need to implement the extended capability and associated semantics including triggering containment and reporting the associated containment trigger event data within the defined capability structures.

5. Analysis of the Software Implications

Software supporting this ECN will need to implement the extended capability and associating semantics. While beyond the scope of this ECN, if error recovery is desired, software will need to take appropriate steps to perform error recovery. Key areas for software to take action on or to consider are included within the ECN text including implementation notes.

6. Analysis of the C&I Test Implications

If C&I tests are developed, then the associated extended capability as well as error trigger event will need to be developed. This may entail developing a DPC-focused DUT that can be instructed to simulate an async removal event and trigger DPC within a Downstream Port (a port that is physically upstream from the DUT).

Part II

Detailed Description of the change

Modify Terms and Acronyms as shown:

Terms and Acronyms

async removal	Removal of an adapter or cable from a slot without lock-step synchronization with the operating system (i.e., in an asynchronous manner without button presses, etc.)
Downstream Port Containment, DPC	The automatic disabling of the Link below a Downstream Port following an uncorrectable error, which prevents TLPs subsequent to the error from propagating Upstream or Downstream.
slot	Used generically to refer to an add-in card slot, ~~or~~ module bay, or cable receptacle.

Modify Section 1 as follows:

1. Introduction

This chapter presents an overview of the PCI Express architecture and key concepts. PCI Express is a high performance, general purpose I/O interconnect defined for a wide variety of future computing and communication platforms. Key PCI attributes, such as its usage model, load-store architecture, and software interfaces, are maintained, whereas its parallel bus implementation is replaced by a highly scalable, fully serial interface. PCI Express takes advantage of recent advances in point-to-point interconnects, Switch-based technology, and packetized protocol to deliver new levels of performance and features. Power Management, Quality Of Service (QoS), Hot-Plug/Hot-Swap support, Data Integrity, and Error Handling are among some of the advanced features supported by PCI Express.

Modify Section 1.1 as follows:

- Hot-Plug ~~and How-Swap~~ support
    - ♦ Ability to support existing PCI Hot-Plug ~~and Hot-Swap~~ solutions
    - ♦ Ability to support native Hot-Plug ~~and Hot-Swap solutions~~ (no sideband signals required)
    - ♦ Ability to support async removal
    - ♦ Ability to support a unified software model for all form factors

Modify Section 2.2.8.5 as follows:

The Set_Slot_Power_Limit Message includes a one DW data payload. The data payload is copied from the Slot Capabilities register of the Downstream Port and is written into the Device Capabilities register of the Upstream Port on the other side of the Link. Bits 1:0 of Byte 1 of the data payload map to the Slot Power Limit Scale field and bits 7:0 of Byte 0 map to the Slot Power

Limit Value field. Bits 7:0 of Byte 3, 7:0 of Byte 2, and 7:2 of Byte 1 of the data payload must be set to all 0's by the Transmitter and ignored by the Receiver. This Message must be sent automatically by the Downstream Port (of a Root Complex or a Switch) when one of the following events occurs:

- □ On a Configuration Write to the Slot Capabilities register (see Section 7.8.9) when the Data Link Layer reports DL_Up status.
- □ Any time when a Link transitions from a non-DL_Up status to a DL_Up status and the Auto Slot Power Limit Disable bit is Cleared in the Slot Control Register (see Section 2.9.2). This Transmission is optional if the Slot Capabilities register has not yet been initialized.

Add the following to the end of Section 2.8

IMPLEMENTATION NOTE

Completion Timeout Prefix/Header Log Capable

The prefix/header of the Request TLP associated with a Completion Timeout may optionally be recorded by Requesters that implement the Advanced Error Reporting Capability. Support for recording of the prefix/header is indicated by the value of the Completion Timeout Prefix/Header Log Capable bit in the Advanced Error Capabilities and Control register.

A Completion Timeout may be the result of improper configuration, system failure, or Async Removal (see Section 6.7.5). In order for host software to distinguish a Completion Timeout error after which continued normal operation is not possible (e.g., after one caused by improper configuration or a system failure) from one where continued normal operation is possible (e.g., after an Async Removal), it is strongly encouraged that Requesters log the Request TLP prefix/header associated with the Completion Timeout.

Modify Section 2.9.1 as follows:

DL_Down status indicates that there is no connection with another component on the Link, or that the connection with the other component has been lost and is not recoverable by the Physical or Data Link Layers. This section specifies the Transaction Layer's behavior if Downstream Port Containment has not been triggered and the Data Link Layer reports DL_Down status to the Transaction Layer, indicating that the Link is non-operational. Section 2.9.3 specifies the behavior if Downstream Port Containment has been triggered.

Modify Section 2.9.2 as follows:

For a Downstream Port on a Root Complex or a Switch:

- □ When transitioning from a non-DL_Up Status to a DL_Up Status and the Auto Slot Power Limit Disable bit is Cleared in the Slot Control Register, the Port must initiate the transmission of a Set_Slot_Power_Limit Message to the other component on the Link to convey the value programmed in the Slot Power Limit Scale and Value fields of the Slot Capabilities register. This Transmission is optional if the Slot Capabilities register has not yet been initialized.

Add Section 2.9.3:

2.9.3. Transaction Layer Behavior During Downstream Port Containment

During Downstream Port Containment (DPC), the LTSSM associated with the Downstream Port is directed to the Disabled state. Once it reaches the Disabled state, it remains there as long as the DPC Trigger Status bit in the DPC Status register is Set. See Section 6.2.10 for requirements on how long software must leave the Downstream Port in DPC. This section specifies the Transaction Layer's behavior once DPC has been triggered, and as long as the Downstream Port remains in DPC.

- Once DPC has been triggered, no additional (Upstream) TLPs are accepted from the Data Link Layer.
- If the condition that triggered DPC was associated with an Upstream TLP, any subsequent Upstream TLPs that were already accepted from the Data Link Layer must be discarded silently.

The Downstream Port handles (Downstream) TLPs submitted by the device core in the following manner.

- If the condition that triggered DPC was associated with a Downstream TLP, any prior Downstream TLPs are permitted to be dropped silently or transmitted before the Link goes down. Otherwise, the following rules apply.
- For each Non-Posted Request, the Port must return a Completion and discard the Request silently. The Completer ID field must contain the value associated with the Downstream Port.
    - If the DPC Completion Control bit is Set in the DPC Control register, then Completions are generated with Unsupported Request (UR) Completion Status.
    - If the DPC Completion Control bit is Clear, Completions are generated with Completer Abort (CA) Completion Status.
    - If DPC occurs while any Non-Posted Requests are still outstanding, the associated Requesters will encounter Completion Timeouts. The solution stack should comprehend and account for this possibility.
- The Port must terminate any PME_Turn_Off handshake Requests targeting the Port in such a way that the Port is considered to have acknowledged the PME_Turn_Off Request (see the Implementation Note in Section 5.3.3.2.1).
- The Port must handle Vendor Defined Message Requests as described in Section 2.2.8.6. (e.g., silently discard Vendor Defined Type 1 Message Requests that it is not designed to receive) since the DL_Down prevents the Request from reaching its targeted Function.
- For all other Posted Requests and Completions, the Port must silently discard the TLP.

Modify Section 6.2.2.1 as follows:

6.2.2.1. Correctable Errors

Correctable errors include those error conditions where hardware can recover without any loss of information. Hardware corrects these errors and software intervention is not required. For example, an LCRC error in a TLP that might be corrected by Data Link Level Retry is considered a correctable error. Measuring the frequency of Link-level correctable errors may be helpful for profiling the integrity of a Link.

Correctable errors also include transaction-level cases where one agent detects an error with a TLP, but another agent is responsible for taking any recovery action if needed, such as re-attempting the operation with a separate subsequent transaction. The detecting agent can be configured to report the error as being correctable since the recovery agent may be able to correct it. If recovery action is indeed needed, the recovery agent must report the error as uncorrectable if the recovery agent decides not to attempt recovery.

The triggering of Downstream Port Containment (DPC) is not handled as an error, but it can be signaled as if it were a correctable error, since software that takes advantage of DPC can sometimes recover from the uncorrectable error that triggered DPC. See Section 6.2.10.

Modify Figure 6-2 as follows:

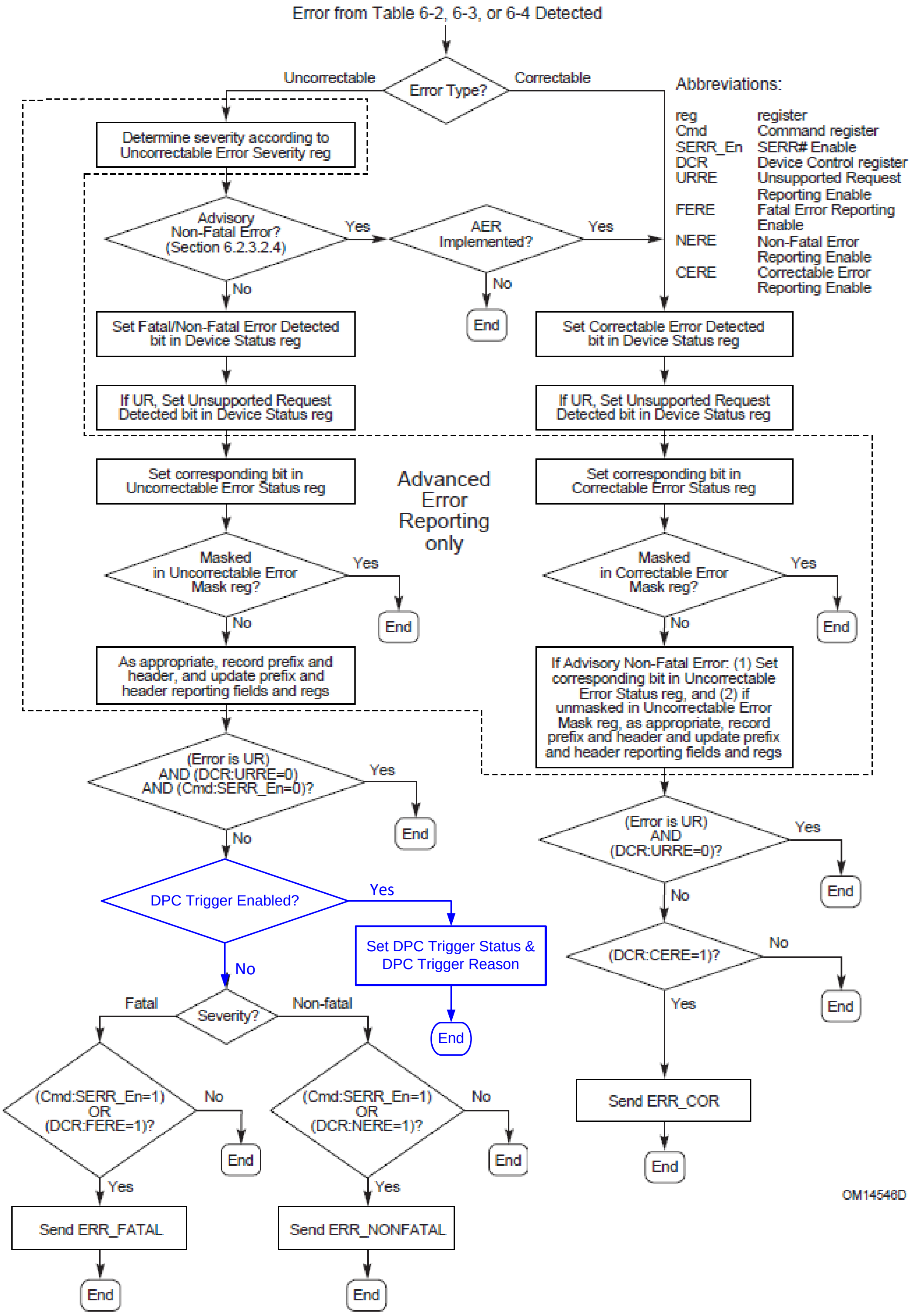

Error from Table 6-2, 6-3, or 6-4 Detected
Error Type?
Uncorrectable
Correctable
Abbreviations:
reg register
Cmd Command register
SERR_En SERR# Enable
DCR Device Control register
URRE Unsupported Request Reporting Enable
FERE Fatal Error Reporting Enable
NERE Non-Fatal Error Reporting Enable
CERE Correctable Error Reporting Enable
Determine severity according to Uncorrectable Error Severity reg
Advisory Non-Fatal Error? (Section 6.2.3.2.4)
Yes
AER Implemented?
Yes
No
End
No
Set Fatal/Non-Fatal Error Detected bit in Device Status reg
Set Correctable Error Detected bit in Device Status reg
If UR, Set Unsupported Request Detected bit in Device Status reg
If UR, Set Unsupported Request Detected bit in Device Status reg
Set corresponding bit in Uncorrectable Error Status reg
Advanced Error Reporting only
Set corresponding bit in Correctable Error Status reg
Masked in Uncorrectable Error Mask reg?
Yes
End
No
Masked in Correctable Error Mask reg?
Yes
End
No
As appropriate, record prefix and header, and update prefix and header reporting fields and regs
If Advisory Non-Fatal Error: (1) Set corresponding bit in Uncorrectable Error Status reg, and (2) if unmasked in Uncorrectable Error Mask reg, as appropriate, record prefix and header and update prefix and header reporting fields and regs
(Error is UR) AND (DCR:URRE=0) AND (Cmd:SERR_En=0)?
Yes
End
No
(Error is UR) AND (DCR:URRE=0)?
Yes
End
No
DPC Trigger Enabled?
Yes
Set DPC Trigger Status & DPC Trigger Reason
End
No
(DCR:CERE=1)?
No
End
Yes
Severity?
Fatal
Non-fatal
(Cmd:SERR_En=1) OR (DCR:FERE=1)?
No
End
Yes
(Cmd:SERR_En=1) OR (DCR:NERE=1)?
No
End
Yes
Send ERR_COR
End
OM14546D
Send ERR_FATAL
End
Send ERR_NONFATAL
End

Figure 6-2: Flowchart Showing Sequence of Device Error Signaling and Logging Operations

Modify Table 6-5 as follows:

Completion Timeout		*Requester:* Send ERR_NONFATAL to Root Complex or ERR_COR for the Advisory Non-Fatal Error case described in Section 6.2.3.2.4.4. If the Completion Timeout Prefix/Header Log Capable bit is Set in the Advanced Error Capabilities and Control register, log the prefix/header of the Request TLP that encountered the error.	Section 2.8

Add Section 6.2.10:

6.2.10. Downstream Port Containment (DPC)

Downstream Port Containment (DPC) is an optional normative feature of a Downstream Port. DPC halts PCI Express traffic below a Downstream Port after an unmasked uncorrectable error is detected at or below the Port, avoiding the potential spread of any data corruption, and permitting error recovery if supported by software. A Downstream Port indicates support for DPC by implementing a DPC Extended Capability structure, which contains all DPC control and status bits.

DPC is disabled by default, and cannot be triggered unless enabled by software using the DPC Trigger Enable field. When the DPC Trigger Enable field is set to 01b, DPC is enabled and is triggered when the Downstream Port detects an unmasked uncorrectable error or when the Downstream Port receives an ERR_FATAL Message. When the DPC Trigger Enable field is set to 10b, DPC is enabled and is triggered when the Downstream Port detects an unmasked uncorrectable error or when the Downstream Port receives an ERR_NONFATAL or ERR_FATAL Message.

When DPC is triggered due to receipt of an uncorrectable error Message, the Requester ID from the Message is recorded in the DPC Error Source ID register and that Message is discarded and not forwarded Upstream. When DPC is triggered by an unmasked uncorrectable error, that error will not be signaled with an uncorrectable error Message, even if otherwise enabled. However, when DPC is triggered, DPC can signal an interrupt or send an ERR_COR Message if enabled. See Sections 6.2.10.1 and 6.2.10.2.

When DPC is triggered, the Downstream Port immediately Sets the DPC Trigger Status bit and DPC Trigger Reason field to indicate the triggering condition (unmasked uncorrectable error, ERR_NONFATAL, or ERR_FATAL), and disables its Link by directing the LTSSM to the Disabled state. Once the LTSSM reaches the Disabled state, it remains in that state until the DPC Trigger Status bit is Cleared. To ensure that the LTSSM has time to reach the Disabled state or at least to bring the Link down under a variety of error conditions, software must leave the Downstream Port in DPC until the Data Link Layer Link Active bit in the Link Status register reads 0b. See Section 7.8.8. See Section 2.9.3 for other important details on Transaction Layer behavior during DPC.

IMPLEMENTATION NOTE

Data Value of All Ones

Many platforms return a data value of all ones to software when an error is associated with a PCI Express Configuration, I/O, or Memory Read Request. During DPC, the Downstream Port discards Requests destined for the Link and completes them with an error (i.e., either with an Unsupported Request (UR) or Completer Abort (CA) Completion Status). By ending a series of MMIO or configuration space operations with a read to an address with a known data value not equal to all ones, software may determine if a Completer has been removed or DPC has been triggered.

IMPLEMENTATION NOTE

Selecting Non-Posted Request Response During DPC

The DPC Completion Control bit determines how a Downstream Port responds to a Non-Posted Request (NPR) received during DPC. The selection needs to take into account how the rest of the platform handles PCI Express uncorrectable error recovery.

While specific PCI Express uncorrectable error recovery mechanisms in a platform are outside the scope of this specification, here are some guidelines based on general considerations.

If the platform or drivers do not support a PCI Express uncorrectable error recovery strategy, there's no envisioned benefit to enabling DPC, and thus no need to select the NPR response.

If the PCI Express uncorrectable error recovery strategy relies on software detecting containment by looking for all 1's returned by PIO reads, then a UR Completion may be the more appropriate selection, assuming the RP synthesizes an all 1's return value for PIO reads that return UR Completions. The all 1's synthesis would need to occur for PIO reads that target Configuration Space, Memory Space, and perhaps I/O Space.

If the PCI Express uncorrectable error recovery strategy utilizes a mechanism that handles UR and CA Completions differently for PIO reads, then a CA Completion might be the more appropriate selection. CA Completions coming back from a PCI Express device normally indicate a device programming model violation, which may need to trigger Root containment and error recovery.

IMPLEMENTATION NOTE

Selecting the DPC Trigger Condition

Non-Fatal Errors are uncorrectable errors that indicate that a particular TLP was unreliable, and in general the associated Function should not continue its normal operation. Fatal errors are uncorrectable errors that indicate that a particular Link and its related hardware are unreliable, and in general the entire hierarchy below that Link should not continue normal operation. This distinction between Non-Fatal and Fatal errors together with the Root Port error containment capabilities can sometimes be used to select the appropriate DPC trigger condition. The following assumes that there is no peer-to-peer traffic between devices.

Some RCs implement a proprietary feature that will be referred to generically as “Function Level Containment” (FLC). This is not an architected feature of PCI Express. A Root Port that implements FLC is capable of containing the traffic associated with a specific Function when a Non-Fatal Error is detected in that traffic. Switch Downstream Ports below a Root Port with FLC should be configured to trigger DPC when the Downstream Port detects an unmasked uncorrectable error itself or when the Downstream Port receives an ERR_FATAL Message. Under this mode, the Switch Downstream Port passes ERR_NONFATAL Messages it receives Upstream without triggering DPC. This enables Root Port FLC to handle Non-Fatal Errors that render a specific Function unreliable and Switch Downstream Port DPC to handle errors that render a sub-tree of the hierarchy domain unreliable. The Downstream Port still needs to trigger DPC for all unmasked uncorrectable errors it detects, since an ERR_NONFATAL it generates will have its own Requester ID, and the FLC hardware in the Root Port would not be able to determine which specific Function below the Switch Downstream Port was responsible for the Non-Fatal Error.

Switch Downstream Ports below a Root Port without FLC should be configured to trigger DPC when the Switch Downstream Port detects an unmasked uncorrectable error or when the Switch Downstream Port receives an ERR_NONFATAL or ERR_FATAL Message. This enables DPC to contain the error to the affected hierarchy below the Link and allow continued normal operation of the unaffected portion of the hierarchy domain.

6.2.10.1. DPC Interrupts

A DPC-capable Downstream Port must support the generation of DPC interrupts. DPC interrupts are enabled by the DPC Interrupt Enable bit in the DPC Control register. DPC interrupts are indicated by the DPC Interrupt Status bit in the DPC Status register.

If the Port is enabled for level-triggered interrupt signaling using INTx messages, the virtual INTx wire must be asserted whenever and as long as the following conditions are satisfied:

❑ The value of the Interrupt Disable bit in the Command register is 0b.

❑ The value of the DPC Interrupt Enable bit is 1b.

❑ The value of the DPC Interrupt Status bit is 1b.

Note that all other interrupt sources within the same Function will assert the same virtual INTx wire when requesting service.

If the Port is enabled for edge-triggered interrupt signaling using MSI or MSI-X, an interrupt message must be sent every time the logical AND of the following conditions transitions from FALSE to TRUE:

❑ The associated vector is unmasked (not applicable if MSI does not support PVM).

❑ The value of the DPC Interrupt Enable bit is 1b.

❑ The value of the DPC Interrupt Status bit is 1b.

The Port may optionally send an interrupt message if interrupt generation has been disabled, and the logical AND of the above conditions is TRUE when interrupt generation is subsequently enabled.

The interrupt message will use the vector indicated by the DPC Interrupt Message Number field in the DPC Capability register. This vector may be the same or may be different from the vectors used by other interrupt sources within this Function.

6.2.10.2. DPC ERR_COR Signaling

A DPC-capable Downstream Port must support ERR_COR signaling, independent of whether it supports Advanced Error Reporting (AER) or not. DPC ERR_COR signaling is enabled by the DPC ERR_COR Enable bit in the DPC Control register. DPC triggering is indicated by the DPC Trigger Status bit in the DPC Status register. DPC ERR_COR signaling is managed independently of DPC interrupts, and it is permitted to use both mechanisms concurrently.

If the DPC ERR_COR Enable bit is Set, and the Correctable Error Reporting Enable bit in the Device Control register is Set, the Port must send an ERR_COR Message each time the DPC Trigger Status bit transitions from Clear to Set. DPC ERR_COR signaling must not Set the Correctable Error Detected bit in the Device Status register, since this event is not handled as an error.

For a given DPC trigger event, if a Port is going to send both an ERR_COR Message and an MSI/MSI-X transaction, then the Port must send the ERR_COR Message prior to sending the MSI/MSI-X transaction. There is no corresponding requirement if the INTx mechanism is being used to signal DPC interrupts, since INTx Messages won't necessarily remain ordered with respect to ERR_COR Messages when passing through routing elements.

IMPLEMENTATION NOTE

Use of DPC ERR_COR Signaling

It is recommended that operating systems use DPC interrupts for signaling when DPC has been triggered. While DPC ERR_COR signaling indicates the same event, DPC ERR_COR signaling is primarily intended for use by platform firmware, when it needs to be notified in order to do its own logging of the event or provide "firmware first" services.

Modify Section 6.6.1 as follows:

6.6.1. Conventional Reset

❑ There are three distinct types of Conventional Reset: cold, warm, and hot:

- …
- There is an in-band mechanism for propagating Conventional Reset across a Link. This is called a hot reset and is described in Section 4.2.4.8.2.

  There is an in-band mechanism for software to force a Link into Electrical Idle, "disabling" the Link. The Disabled LTSSM state is described in Section 4.2.5.9, ~~and~~ the Link Disable control bit is described in Section 7.8.7, and the Downstream Port Containment mechanism is described in Section 6.2.10. Disabling a Link causes Downstream components to undergo a hot reset.

Modify Section 6.7 as follows:

6.7. PCI Express Hot-Plug Support

The PCI Express architecture is designed to natively support both hot-add and hot-removal ("hot-plug") of ~~adapters~~ cables, add-in cards, and modules. PCI Express hot-plug support ~~and~~ provides a "toolbox" of mechanisms that allow different user/operator models to be supported using a self-consistent infrastructure. These mechanisms may be used to implement orderly addition/removal that relies on coordination with the operating system (e.g., traditional PCI hot-plug), as well as async removal which proceeds without lock-step synchronization with the operating system. This section defines the set of hot-plug mechanisms and specifies how the elements of hot-plug, such as indicators and push buttons, must behave if implemented in a system.

Modify Table 6-6 as follows:

Table 6-6: Elements of Hot-Plug

Element	Purpose
Indicators	Show the power and attention state of the slot
Manually-operated Retention Latch (MRL)	Holds adapter in place
MRL Sensor	Allows the Port and system software to detect the MRL being opened
Electromechanical Interlock	Prevents removal of adapter from slot
Attention Button	Allows user to request hot-plug operations
Software User Interface	Allows user to request hot-plug operations
Slot Numbering	Provides visual identification of slots
Power Controller	Software-controlled electronic component or components that control power to a slot or adapter and monitor that power for fault conditions
Out-of-band Presence Detect	Method of determining physical presence of an adapter in a slot that does not rely on the Physical Layer

Modify Section 6.7.1.3 as follows:

IMPLEMENTATION NOTE

MRL Sensor Handling

In the absence of an MRL sensor, for some form factors, ~~staggered presence detect pins~~ out-of-band presence detect may be used to handle the switched signals. In this case, when ~~the presence pins break contact~~ out-of-band presence detect indicates the absence of an adapter in a slot, the switched signals will be automatically removed from the slot.

If an MRL Sensor is implemented without a corresponding MRL Sensor input on the Hot-Plug Controller, it is recommended that the MRL Sensor be routed to power fault input of the Hot-Plug Controller. This allows an active adapter to be powered off when the MRL is opened.

Add Section 6.7.5:

6.7.5. Async Removal

Async removal refers to the removal of an adapter or disabling of a Downstream Port Link due to error containment without prior warning to the operating system. This is contrast to standard PCI hot-plug where removal operations are performed in a lock-step manner with the OS through a well defined sequence of user actions and system management facilities. For example, the user presses the Attention Button to request permission from the OS to remove the adapter, but the user doesn't actually remove the adapter from the slot until the OS has quiesced activity to the adapter and granted permission for removal.

Since async removal proceeds before the rest of the PCI Express hierarchy or OS necessarily becomes aware of the event, special consideration is required beyond that needed for standard PCI hot-plug. This section outlines PCI Express events that may occur as a side effect of async removal and mechanisms for handling async removal.

Since async removal may be unexpected to both the Physical and Data Link Layers of the Downstream Port associated with the slot, Correctable Errors may be reported as a side effect of the event (i.e.~~;~~ Receiver Error, Bad TLP, and Bad DLLP). If these errors are reported, software should handle them as an expected part of this event.

Requesters may experience Completion Timeouts associated with Requests that were accepted, but will never be completed by removed Completers. Any resulting Completion Timeout errors in this context should be handled as an expected part of this event.

Async removal may result in a transition from DL_Active to DL_Down in the Downstream port. This transition may result in a Surprise Down error. In addition, Requesters in the PCI Express hierarchy domain may not become immediately aware of this transition and continue to issue Requests to removed Completers that must be handled by the Downstream Port associated with the slot.

The Surprise Down error resulting from async removal may trigger Downstream Port Containment (See Section 6.2.10). Downstream Port Containment following an async removal may be utilized to hold the Link of a Downstream Port in the Disabled LTSSM state while host software recovers from the side effects of an async removal.

Add the following to the end of Section 6.9:

IMPLEMENTATION NOTE

Auto Slot Power Limit Disable

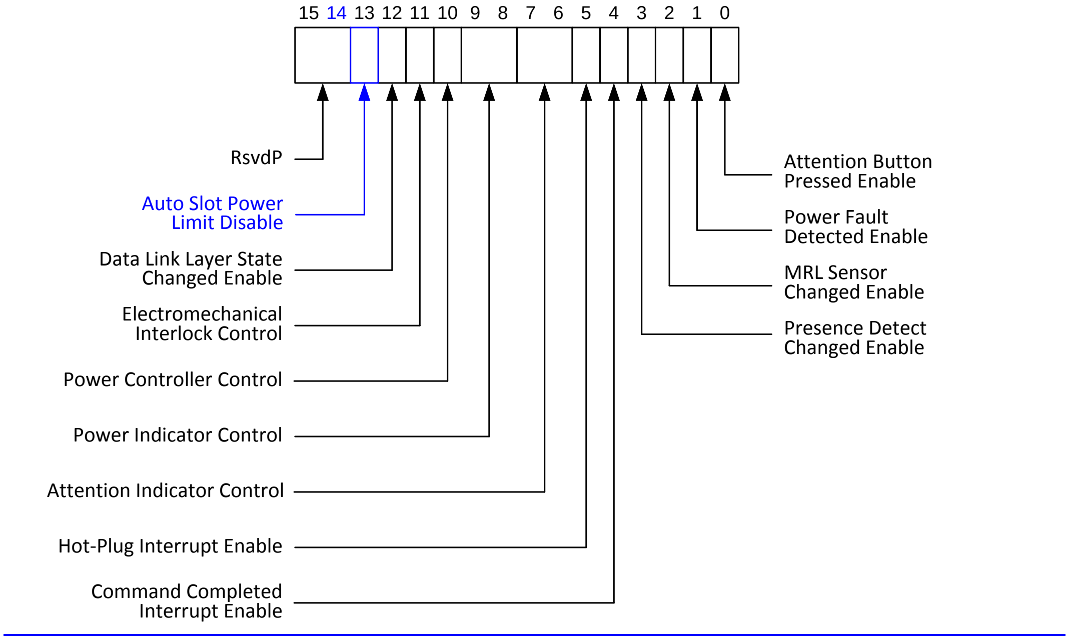
In some environments host software may wish to directly manage the transmission of a Set_Slot_Power_Limit message by performing a Configuration Write to the Slot Capabilities register rather than have the transmission automatically occur when the Link transitions from a non-DL_Up to a DL_Up status. This allows host software to limit power supply surge current by staggering the transition of Endpoints to a higher power state following a Link Down or when multiple Endpoints are simultaneously hot-added due to cable or card insertion.

Modify Table 7-13 as follows:

Table 7-13: Device Control Register

Bit Location	Register Description	Attributes
0	**Correctable Error Reporting Enable** – This bit, in conjunction with other bits, controls sending ERR_COR Messages (see Section 6.2.5,~~ and~~ Section 6.2.6, and Section 6.2.10.2 for details). For a multi-Function device, this bit controls error reporting for each Function from point-of-view of the respective Function. For a Root Port, the reporting of correctable errors is internal to the root. No external ERR_COR Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW

Modify Figure 7-20 as follows:

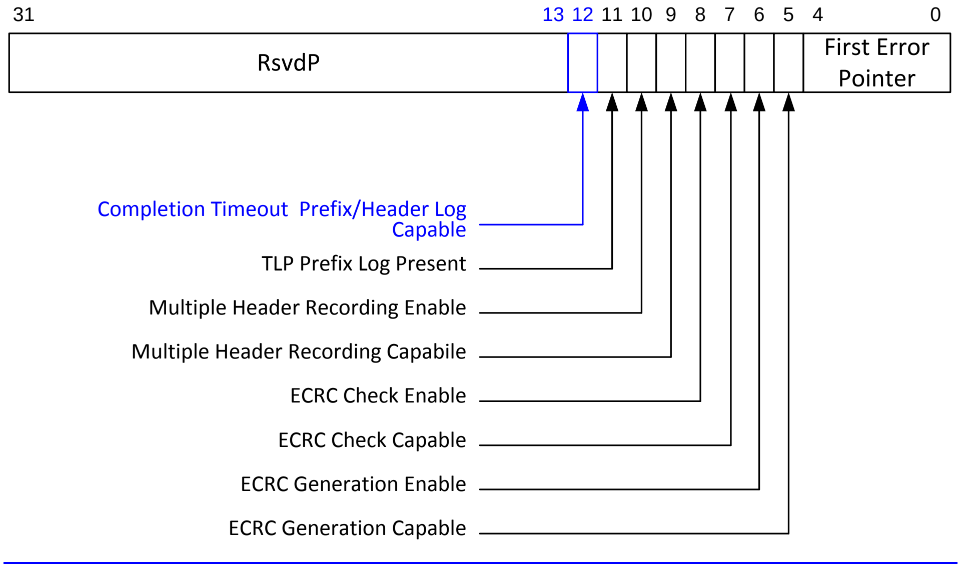


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RsvdP
Auto Slot Power Limit Disable
Data Link Layer State Changed Enable
Electromechanical Interlock Control
Power Controller Control
Power Indicator Control
Attention Indicator Control
Hot-Plug Interrupt Enable
Command Completed Interrupt Enable
Attention Button Pressed Enable
Power Fault Detected Enable
MRL Sensor Changed Enable
Presence Detect Changed Enable

Add the following row to Table 7-20:

13	**Auto Slot Power Limit Disable** – When Set, this disables the automatic sending of a Set_Slot_Power_Limit Message when a Link transitions from a non-DL_Up status to a DL_Up status. Default value of this bit is implementation specific.	RW

Modify Figure 7-39 as follows:



Add the following rows to Table 7-36:

12	**Completion Timeout Prefix/Header Log Capable** – If Set, this bit indicates that the Function records the prefix/header of Request TLPs that experience a Completion Timeout error.	RO

Add Section 7.xx:

7.xx. DPC Extended Capability

The Downstream Port Containment (DPC) Extended Capability is an optional normative capability that provides a mechanism for Downstream Ports to contain uncorrectable errors and enable software recover from them. See Section 6.2.10. This capability may be implemented by a Root Port or a Switch Downstream Port. It is not applicable to any other Device/Port type.

If a Downstream Port implements the DPC Extended Capability, that Port must also be capable of reporting the DL_Active state, and indicate so by Setting the Data Link Layer Link Active Reporting Capable bit in the Link Capabilities register. See Section 7.8.6.

31	0	Byte Offset
PCI Express Extended Capability Header		000h
DPC Control Register	DPC Capability Register	004h
DPC Error Source ID Register	DPC Status Register	008h

Figure 7-xx DPC Extended Capability

7.xx.1. DPC Extended Capability Header (Offset 00h)

31 20	19 16	15 0
Next Capability Offset		PCI Express Extended Capability ID

↑ Capability Version

Figure 7-xx DPC Extended Capability Header

Table 7-xx DPC 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 DPC Extended Capability is 001Dh.	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.xx.2. DPC Capability Register (Offset 04h)

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RsvdP</td><td colspan="5"></td></tr>
</table>

Figure 7-xx DPC Capability Register

Note to Editor: modify the above figure to indicate the field that is defined in the table below.

Table 7-xx DPC Capability Register

Bit Location	Register Description	Attributes
4:0	**DPC Interrupt Message Number** – This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with the DPC Capability structure. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this field must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this field must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this field is undefined.	RO

7.xx.3. DPC Control Register (Offset 06h)

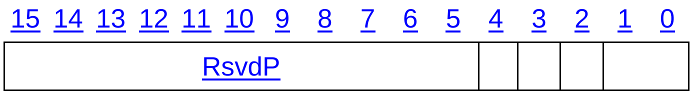


Figure 7-xx DPC Control Register

Note to Editor: modify the above figure to indicate the bits/fields that are defined in the table below.

Table 7-xx DPC Control Register

Bit Location	Register Description	Attributes
1:0	**DPC Trigger Enable** – This field enables DPC and controls the conditions that cause DPC to be triggered. Defined encodings are: 00b DPC is disabled 01b DPC is enabled and is triggered when the Downstream Port detects an unmasked uncorrectable error or when the Downstream Port receives an ERR_FATAL Message. 10b DPC is enabled and is triggered when the Downstream Port detects an unmasked uncorrectable error or when the Downstream Port receives an ERR_NONFATAL or ERR_FATAL Message. 11b Reserved Default value of this field is 00b.	RW
2	**DPC Completion Control** – This bit controls the Completion Status for Completions formed during DPC. See Section 2.9.3. Defined encodings are: 0b Completer Abort (CA) Completion Status 1b Unsupported Request (UR) Completion Status Default value of this bit is 0b.	RW
3	**DPC Interrupt Enable** – When Set, this bit enables the generation of an interrupt to indicate that DPC has been triggered. See Section 6.2.10.1. Default value of this bit is 0b.	RW
4	**DPC ERR_COR Enable** – When Set, this bit enables the sending of an ERR_COR Message to indicate that DPC has been triggered. See Section 6.2.10.2. Default value of this bit is 0b.	RW

7.xx.4. DPC Status Register (Offset 08h)

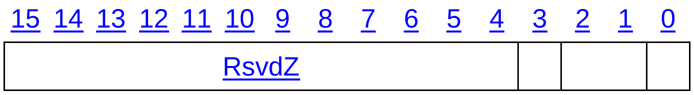


Figure 7-xx DPC Status Register

Note to Editor: modify the above figure to indicate the bits/fields that are defined in the table below.

Table 7-xx DPC Status Register

Bit Location	Register Description	Attributes
0	**DPC Trigger Status** – When Set, this bit indicates that DPC has been triggered. While this bit is Set, hardware must direct the LTSSM to the Disabled State. This bit must be cleared before the LTSSM can be released from the Disabled State. See Section 6.2.10 for requirements on how long software must leave the Downstream Port in DPC. After clearing this bit, software must honor timing requirements defined in Section 6.6.1 with respect to the first Configuration Read following a Conventional Reset. Default value of this bit is 0b.	RW1CS
2:1	**DPC Trigger Reason** – This field indicates why DPC has been triggered. Defined encodings are: 00b DPC was triggered due to an unmasked uncorrectable error. 01b DPC was triggered due to receiving an ERR_NONFATAL. 10b DPC was triggered due to receiving an ERR_FATAL. 11b Reserved This field is valid only when the DPC Trigger Status bit is Set; otherwise the value of this field is undefined.	ROS
3	**DPC Interrupt Status** – This bit is Set if DPC is triggered while the DPC Interrupt Enable bit is Set. This may cause the generation of an interrupt. See Section 6.2.10.1. Default value of this bit is 0b.	RW1CS

7.xx.5. DPC Error Source ID Register (Offset 0Ah)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPC Error Source ID															

Figure 7-xx DPC Error Source ID Register

Table 7-xx DPC Error Source ID Register

Bit Location	Register Description	Attributes
15:0	**DPC Error Source ID** – When the DPC Trigger Reason field indicates that DPC was triggered due to the reception of an ERR_NONFATAL or ERR_FATAL, this register contains the Requester ID of the received Message. Otherwise, the value of this register is undefined.	ROS