



## PCI-SIG ENGINEERING CHANGE REQUEST

<b>TITLE:</b>	ACPI additions for ASPM, OBFF, LTR ECNs
<b>DATE:</b>	October 30, 2009 <b>Updated February 1, 2010</b>
<b>AFFECTED DOCUMENT:</b>	PCI Firmware Spec 3.0
<b>SPONSOR:</b>	Intel Corporation

### **Part I**

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

A number of PCIe base specifications ECNs have been approved that require software support. In some cases, platform firmware needs to know if the OS running supports certain features, or the OS needs to be able to request control of certain features from platform firmware. In other cases, the OS needs to know information about the platform that cannot be discovered through PCI enumeration, and ACPI must be used to supply the additional information.

The list of ECNs requiring firmware changes includes:

1. Latency Tolerance Reporting
2. Optimized Buffer Flush and Fill
3. ASPM Optionality

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

Adding these changes will allow native OS support for new features, as well as providing a mechanism for graceful handoff from platform firmware to the OS. Additionally, where a legacy OS is used, BIOS will be able to manage/enable these features during a hot-add event.

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

System software- both platform firmware and OS will need to incorporate support for these changes in order for the features referenced above to function optimally. Without OS support, certain run-time events may cause the above-referenced features to be disabled.

#### **Analysis of the Hardware Implications**

No hardware implications beyond those already existent in the corresponding base specification and ECNs.

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

Each of the proposed changes will require modified ACPI BIOS firmware to support the corresponding features, if implemented on the platform. OS-level bus driver and OSPM modifications are recommended in order to take advantage of the corresponding features, allowing for OS policy ownership. Without OS support, firmware will be able to enable LTR and OBFF during boot, and may optionally enable the features during a hot-add event when firmware retains control of hot-plug events. Other events, such as FLR and D0→D3→D0 transitions may

## **Request Request Request Request Request Request Request Request Request**

cause these features to be disabled entirely without OS support. It is recommended that any OS that supports FLR also support ASPM Optionality, LTR, and OBFF.

### **5. Analysis of the C&I Test Implications**

C & I implications are covered in the respective feature ECNs.

# Request Request Request Request Request Request Request Request

## **Part II**

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

Modify table 4-4 as follows:

**Table 0-1: Interpretation of the \_OSC Support Field**

<b>Support Field Bit Offset</b>	<b>Interpretation</b>
...	
<b>5</b>	<b>Optimized Buffer Flush and Fill supported</b> The operating system sets this bit to 1 if it supports the OBFF feature and will enable this feature during a native hot plug insertion event if supported by the newly added device in a hierarchy where OBFF is supported. Otherwise, the operating system sets this bit to 0.
<b>6</b>	<b>ASPM Optionality supported</b> The operating system sets this bit to 1 if it properly recognizes and manages ASPM support on PCI Express components which report support for ASPM L1 only in the ASPM Support field within the Link Capabilities Register. Otherwise, the operating system sets this bit to 0.
<b>47-31</b>	Reserved

Add Implementation Note below table 4-4 as follows:



## **IMPLEMENTATION NOTE**

### **Recommended Use of ASPM Optionality Support Bit**

The ASPM Optionality bit in the \_OSC Support field is designed to be used by platform firmware for use with devices known a priori by platform firmware to have an ASPM Support field that is write-once or otherwise lockable, configurable by platform firmware, and support ASPM L1 but do not support L0s.

In the case where the OS does not support ASPM Optionality, such devices would be configured by firmware to report support for both L0s and ASPM L1.

In the case where the OS supports ASPM Optionality, such devices would be configured to report support for only ASPM L1.

**Request Request Request Request Request Request Request Request**

*Modify table 4-5 as follows:*

**Table 0-2: Interpretation of the \_OSC Control Field, Passed in via Arg3**

<b>Control Field Bit Offset</b>	<b>Interpretation</b>	<b>Interdependencies</b>
...		
<b>5</b>	<p><b>Latency Tolerance Reporting control</b></p> <p>The operating system sets this bit to 1 to request control over PCI Express Latency Tolerance Reporting. If the operating system successfully receives control of this feature, it must manage configuration of the LTR Extended Capability structure and LTR Mechanism Enable fields as described in the PCI Express Base Specification; further, the operating system retains control of LTR across power transitions for S1, S2, S3 system power states.</p>	<b>Yes, refer to 4.5.2.4</b>
<b>56-31</b>	Reserved	

## Request Request Request Request Request Request Request Request

Modify table 4-6 as follows:

**Table 0-3: Interpretation of the \_OSC Control Field, Returned Value**

Control Field Bit Offset	Interpretation	Interdependencies
...		
5	<p><b>Latency Tolerance Reporting control</b></p> <p>The firmware sets this bit to 1 to grant control over control over PCI Express Latency Tolerance Reporting. If firmware allows the operating system control of this feature, then in the context of the _OSC method, it must ensure that the LTR Extended Capabilities Structure is initialized properly in all devices that support LTR as described in the PCI Express Base Specification. If control of this feature was requested and denied or was not requested, firmware returns this bit set to 0.</p>	<b>Yes, refer to 4.5.2.4</b>
<del>56</del> -31	Reserved	

Modify table 4-7 as follows:

**Table 0-4: \_DSM Definitions for PCI**

UUID	Revision	Function	Description
E5C937D0-3553-4d7a-9117-EA4D19C3434D	<del>42</del>	1	PCI Express Slot Information
	<del>42</del>	2	PCI Express Slot Number
	<del>42</del>	3	Vendor-specific Token ID
	<del>42</del>	4	PCI Bus Capabilities
	<del>42</del>	5	Ignore PCI Boot Configuration
	2	<del>56</del>	<del>LTR Maximum Latency</del>

## Request Request Request Request Request Request Request Request

Modify table 4-9 as follows:

**Table 0-5: PCI Bus Capability Structure**

Field	Byte Length	Byte Offset	Description
...			
Attributes	1	4	The bit corresponding to each attribute of the bus should be set: 01h: 64-bit Device – This bit should be set if the secondary interface of the bus is 64 bits wide. 02h: PCI-X Mode 1 ECC Capable – This bit should be set if the bus is capable of supporting ECC in PCI-X Mode 1. 04h: Device ID Messaging Capable – This bit should be set if the bridge is capable of forwarding Device ID Message transactions. 08h: OBFF Capable – This bit should be set if the OBFF/WAKE# signal is provided to devices beneath this bridge, but the bridge does not report support for OBFF in its Device Capabilities 2 Register.
...			

Insert section 4.6.4.1.x as follows:

### **4.6.x \_DSM Definitions for Latency Tolerance Reporting**

This section describes how PCI Express firmware describes Latency Tolerance Reporting information to the operating system.

For Root Complexes that support Latency Tolerance Reporting, there is a set of maximum latency values that could never be exceeded in normal operation. Platform Firmware must convey the maximum values for each downstream port embedded within the platform. System software is responsible for calculating latencies along the path between each downstream port and any Endpoint supporting LTR beneath the port, and programming the sum into the Endpoint's Latency Tolerance Reporting Extended Capability Structure.

#### **Arguments:**

Arg0: UUID: E5C937D0-3553-4d7a-9117-EA4D19C3434D

Arg1: Revision ID: 2

Arg2: Function Index: 6

Arg3: Empty Package

#### **Return:**

A Package of four integers corresponding with the LTR encoding defined in the PCI Express Base Specification, as follows:

Integer 0: Maximum Snoop Latency Scale

## **Request Request Request Request Request Request Request Request**

Integer 1: Maximum Snoop Latency Value

Integer 2: Maximum No-Snoop Latency Scale

Integer 3: Maximum No-Snoop Latency Value

These values correspond directly to the LTR Extended Capability Structure fields described in the PCI Express Base Specification.