PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	ASPM Optionality
DATE:	June 19, 2009; approved by PWG August 20, 2009
AFFECTED DOCUMENT:	PCIe Base Specification Revision 2.1
SPONSOR:	Hewlett-Packard, Intel, Microsoft

Part I

1.   Summary of the Functional Changes

Prior to this ECN, all PCIe external Links were required to support ASPM L0s. This ECN changes the Base Specification to permit ASPM L0s support to be optional unless the applicable form-factor specification explicitly requires it.

Given the new optionality of ASPM L0s, the ECN clarifies that software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s.

Prior to this ECN, Transmitters enabled for L0s were required to enter L0s under specified idle conditions. This ECN relaxes this requirement, keeping it as a recommendation, but making it never mandatory for a Transmitter to enter L0s.

Finally, this ECN defines a new capability bit, ASPM Optionality Compliance, which software is permitted to use to help determine whether to enable ASPM or whether to run ASPM compliance tests.

2.   Benefits as a Result of the Changes

The cost of implementing, validating, and supporting L0s can be avoided for components used in applications where the benefit of L0s is not compelling.

Software can know for sure when not to enable L0s on components that do not support L0s. Software is also permitted to use the ASPM Optionality Compliance bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. These reduce the need for software to use "blacklists" as well as the need for hardware components to use unintended practices that discourage the use of L0s.

3.   Assessment of the Impact

For some cases, adapter IHVs and platform vendors can avoid unnecessary costs of developing and supporting L0s, and avoid time-to-market delays caused by L0s problems. Support costs for OSVs supporting such platforms should be reduced.

4.   Analysis of the Hardware Implications

Components intended for applications where L0s benefits are non-compelling can choose not to support L0s.

5.   Analysis of the Software Implications

New software needs to recognize newly defined values (00b & 10b) for the ASPM Support field in the Link Capability register of the PCI Express Capability structure. On a given Link, software must not enable L0s in either direction unless components on both sides of the Link indicate their support of L0s.

Legacy software (either OS or firmware) that encounters the previously reserved value 00b (No ASPM Support), will most likely refrain from enabling L1, which is intended behavior. Legacy software will also most likely refrain from enabling L0s for that component's Transmitter (also intended behavior), but it's unclear if such software will also refrain from enabling L0s for the component on the other side of the Link. If software enables L0s on one side when the component on the other side does not indicate its support of L0s, the result is undefined. Situations where the resulting behavior is unacceptable may need to be handled by updating the legacy software, resorting to "blacklists" or similar mechanisms directing the legacy software not to enable L0s, or simply not supporting the problematic system configurations.

On some platforms, firmware controls ASPM, and the OS may either preserve or override the ASPM settings established by firmware. This will be influenced by whether the OS supports controlling ASPM, and in some cases by whether the firmware permits the OS to take control of ASPM. Also, ASPM control with hotplug operations may be influenced by whether native PCIe hotplug versus ACPI hotplug is used. Addressing any legacy software issues with L0s may require updating the firmware, the OS, or both.

Legacy software that encounters the previously reserved value 10b (L1 Support), may refrain from enabling either L0s or L1, which unfortunately avoids using L1 with new components that support only L1. While this may result in additional power being consumed, it should not cause any functional misbehavior. However, the same legacy software issues with respect to enabling L0s exist for this 10b case as described above for the 00b case.

6.   Analysis of the C&I Test Implications

If any C&I tests check the values in the ASPM Support field in the Link Capability register, they will need to comprehend the newly defined values (00b & 10b).

If any C&I tests normally enable ASPM L0s, they will need to allow for new components that choose not to support L0s.

C&I test software is permitted to use the ASPM Optionality Compliance bit to help determine whether to run ASPM compliance tests.

Part II

Detailed Description of the change

Modify Section 4.2.6.5 as shown:

4.2.6.5.   L0

…

❑ Next state is L0s for only the Transmitter if directed to this state and the Transmitter implements L0s. See Section 4.2.6.6.2.

- Note: "if directed" applies to a Port that is instructed by a higher Layer to initiate L0s (see Section 5.4.1.1.1).

- Note: This is a point where the TX and RX may diverge into different LTSSM states.

❑ Next state is L0s for only the Receiver if an EIOS is received on any Lanes, the Receiver implements L0s, and the Port is not directed to L1 or L2 states by any higher layers. See Section 4.2.6.6.1.

- Note: This is a point where the TX and RX may diverge into different LTSSM states.

❑ Next state is Recovery if an EIOS is received on any Lane, the Receiver does not implement L0s, and the Port is not directed to L1 or L2 states by any higher layers. See Section 4.2.6.6.1.

Modify Section 4.2.6.6.1 as shown:

4.2.6.6.1. Receiver L0s

A Receiver must implement L0s if its Port advertises support for L0s, as indicated by the ASPM Support field in the Link Capability register. It is permitted for a Receiver to implement L0s even if its Port does not advertise support for L0s.

Modify Section 4.2.6.6.2 as shown:

4.2.6.6.2. Transmitter L0s

A Transmitter must implement L0s if its Port advertises support for L0s, as indicated by the ASPM Support field in the Link Capability register. It is permitted for a Transmitter to implement L0s even if its Port does not advertise support for L0s.

Modify Section 5.1.1 as shown:

5.1.1.   Statement of Requirements

…

ASPM ~~is a required feature~~ requirements are form-factor specific. Refer to Section 5.4.1 for more information on ASPM.

Modify Section 5.2 as shown:

5.2. Link State Power Management

…

PCI Express-PM defines the following Link power management states:

…

❑ L0s – A low resume latency, energy saving "standby" state.

L0s support is ~~required~~ optional for ASPM unless the applicable form-factor specification for the Link explicitly requires L0s support. It is not applicable to PCI-PM compatible power management.

…

Table 5-1: Summary of PCI Express Link Power Management States

	L-State Description	Used by S/W Directed PM	Used by ASPM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux
...	…	…	…	…	…	…	…
L0s	Standby state	No	Yes [1] (opt.,D0)	On	On	On	On/Off
L1	Lower power standby	Yes (D1-D3$_{hot}$)	Yes [2] (opt., D0)	On/Off [7]	On	On/Off [3]	On/Off
...	…	…	…	…	…	…	…

Notes:
1. L0s exit latency will be greatest in Link configurations with independent reference clock inputs for components connected to opposite ends of a given Link (vs. a common, distributed reference clock).
2. ~~L1 entry may be requested within ASPM protocol; however, its support is optional unless specifically required by a particular form factor.~~

Modify Section 5.3.2 as shown:

5.3.2. PM Software Control of the Link Power Management State

…

Table 5-2: Relation Between Power Management States of Link and Components

Downstream Component D-State	Permissible Upstream Component D-State	Permissible Interconnect State
D0	D0	L0, L0s, L1[(1)], L2/L3 Ready
…	…	…

Notes:
1. ~~All components are required to support ASPM with L0s entry during idle at a minimum. The use of L1 within D0 is optional unless specifically required by a particular form factor.~~ Requirements for ASPM L0s and ASPM L1 support are form-factor specific.

Modify Section 5.4.1 as shown:

5.4.1. Active State Power Management (ASPM)

…

Two low power "standby" Link states are defined for ASPM. The L0s low power Link state is optimized for short entry and exit latencies, while providing substantial power savings. If the L0s state is enabled in a device, it is ~~required to~~ recommended that the device bring ~~any~~ its Transmit Link into the L0s state whenever that Link is not in use (refer to Section 5.4.1.1.1 for details relating to the L0s invocation policy). ~~All~~

~~components must~~ Component support ~~of~~ the L0s Link state from within the D0 device state is optional unless the applicable form-factor specification for the Link explicitly requires it.

…

Each component must report its level of support for ASPM in the ASPM Support field. ~~Each~~ As applicable, each component shall also report its L0s and L1 exit latency (the time that they require to transition from the L0s or L1 state to the L0 state). Endpoint Functions must also report the worst-case latency that they can withstand before risking, for example, internal FIFO overruns due to the transition latency from L0s or L1 to the L0 state. Power management software can use the provided information to then enable the appropriate level of ASPM.

…

Power management software enables or disables ASPM in each Port of a component by programming the ASPM Control field. Note that new BIOS code can effectively enable or disable ASPM functionality ~~even~~ when running with a legacy operating system, but a PCIe-aware operating system might choose to override ASPM settings configured by BIOS.

…

Note that the components must be capable of changing their behavior during runtime as device Functions enter and exit low power device states. For example, if one Function within a multi-Function device is programmed to disable ASPM, then ASPM must be disabled for that device while that Function is in the D0 state. Once the Function transitions to a non-D0 state, ASPM can be enabled ~~to at least the L0s state~~ if all other Functions are enabled for ASPM.

Modify Section 5.4.1.1 as shown:

5.4.1.1.   L0s ASPM State

~~All devices must~~ Device support ~~of~~ the L0s low power Link state is optional unless the applicable form-factor specification for the Link explicitly requires it.

📌 IMPLEMENTATION NOTE

Potential Issues With Legacy Software When L0s is Not Supported

In earlier versions of this specification, device support of L0s was mandatory, and software could legitimately assume that all devices support L0s. Newer hardware components that do not support L0s may encounter issues with such "legacy software". Such software might not even check the ASPM Support field in the Link Capability register, might not recognize the subsequently defined values (00b & 10b) for the ASPM Support field, or might not follow the policy of enabling L0s only if components on both sides of the Link each support L0s.

Legacy software (either OS or firmware) that encounters the previously reserved value 00b (No ASPM Support), will most likely refrain from enabling L1, which is intended behavior. Legacy software will also most likely refrain from enabling L0s for that component's Transmitter (also intended behavior), but it's unclear if such software will also refrain from enabling L0s for the component on the other side of the Link. If software enables L0s on

one side when the component on the other side does not indicate that it supports L0s, the result is undefined. Situations where the resulting behavior is unacceptable may need to be handled by updating the legacy software, resorting to "blacklists" or similar mechanisms directing the legacy software not to enable L0s, or simply not supporting the problematic system configurations.

On some platforms, firmware controls ASPM, and the OS may either preserve or override the ASPM settings established by firmware. This will be influenced by whether the OS supports controlling ASPM, and in some cases by whether the firmware permits the OS to take control of ASPM. Also, ASPM control with hotplug operations may be influenced by whether native PCIe hotplug versus ACPI hotplug is used. Addressing any legacy software issues with L0s may require updating the firmware, the OS, or both.

When a component does not advertise that it supports L0s, as indicated by its ASPM Support field value being 00b or 10b, it is recommended that the component's L0s Exit Latency field return a value of 111b, indicating the maximum latency range. Advertising this maximum latency range may help discourage legacy software from enabling L0s if it otherwise would do so, and thus may help avoid problems caused by legacy software mistakenly enabling L0s on this component or the component on the other side of the Link.

Modify Section 5.4.1.1.1 as shown:

5.4.1.1.1. Entry into the L0s State

Entry into the L0s state is managed separately for each direction of the Link. It is the responsibility of each device at either end of the Link to initiate an entry into the L0s state on its transmitting Lanes. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

A Port that is disabled for the L0s state must not transition its transmitting Lanes to the L0s state. It must still however However, if the Port advertises that it supports L0s, the Port must be able to tolerate having its Receiver Port Lanes enter L0s, (as a result of the device at the other end bringing its transmitting Lanes into L0s state), and then later returning to the L0 state.

L0s Invocation Policy

Ports that are enabled for L0s entry must generally should transition their Transmit Lanes to the L0s state if the defined idle conditions (below) are met for a period of time, recommended not to exceed 7 µs. Within this time period, the policy used by the Port to determine when to enter L0s is implementation specific. It is never mandatory for a Transmitter to enter L0s.

Definition of Idle

The definition of an "idle" Upstream Port varies with device Function category. An Upstream Port with a multi-Function device is considered idle only when all of its Functions are idle.

An Endpoint Function or a Root Port is determined to be idle if the following conditions are met:

❑ No TLP is pending to transmit over the Link, or no FC credits are available to transmit any TLPs

❑ No DLLPs are pending for transmission

A Switch Upstream Port Function is determined to be idle if the following conditions are met:

❑ ~~All~~ None of the Switch's Downstream Port Receive Lanes are in the ~~L0s~~ L0, Recovery, or Configuration state

❑ No pending TLPs to transmit, or no FC credits are available to transmit anything

❑ No DLLPs are pending for transmission

A Switch's Downstream Port is determined to be idle if the following conditions are met:

❑ The Switch's Upstream Port's Receive Lanes are not in the ~~L0s~~ L0, Recovery, or Configuration state

❑ No pending TLPs to transmit on this Link, or no FC credits are available

❑ No DLLPs are pending for transmission

Modify Section 5.4.1.2 as shown:

5.4.1.2. L1 ASPM State

A component may optionally support the ASPM L1 state; a state that provides greater power savings at the expense of longer exit latency. L1 exit latency is visible to software, and reported via the Link Capabilities register defined in Section 7.8.6.

📌 IMPLEMENTATION NOTE

Potential Issues With Legacy Software When Only L1 is Supported

In earlier versions of this specification, device support of L0s was mandatory, and there was no architected ASPM Support field value to indicate L1 support without L0s support. Newer hardware components that support only L1 may encounter issues with "legacy software"; i.e., software that does not recognize the subsequently defined value for the ASPM Support field.

Legacy software that encounters the previously reserved value 10b (L1 Support), may refrain from enabling both L0s and L1, which unfortunately avoids using L1 with new components that support only L1. While this may result in additional power being consumed, it should not cause any functional misbehavior. However, the same issues with respect to legacy software enabling L0s exist for this 10b case as are described in the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in Section 5.4.1.1.

Modify Section 5.4.1.2.1 as shown:

5.4.1.2.1. Entry into the L1 State

…

Rules in case of rejection:

❑ In the case of a rejection, the Upstream component must schedule, as soon as possible, a rejection by sending the PM_Active_State_Nak Message to the Downstream component.

Once the PM_Active_State_Nak Message is sent, the Upstream component is permitted to initiate any TLP or DLLP transfers.

❑ If the request was rejected, ~~the~~ it is generally recommended that the Downstream component ~~must~~ immediately transition its Transmit Lanes into the L0s state, provided L0s is enabled and that conditions for L0s entry are met.

❑ Prior to transmitting a PM_Active_State_Request_L1 DLLP associated with a subsequent ASPM L1 negotiation sequence, the Downstream component must either enter and exit L0s on its Transmitter, or it must wait at least 10 µs from the last transmission of the PM_Active_State_Request_L1 DLLP associated with the preceding ASPM L1 negotiation. …

Modify Section 5.4.1.3 as shown:

5.4.1.3. ASPM Configuration

All Functions must implement the following configuration bits in support of ASPM. Refer to Chapter 7 for configuration register assignment and access mechanisms.

Each component reports its level of support for ASPM in the ASPM Support field below. ~~All components must support transition to the L0s Link state. Support for transition to the L1 Link state while in D0~~$_{active}$ ~~state is optional unless specifically required by a particular form factor.~~

Table 5-3: Encoding of the ASPM Support Field

Field	Description
ASPM Support	00b – ~~Reserved~~No ASPM support 01b – L0s supported 10b – ~~Reserved~~L1 supported 11b – L0s and L1 supported

Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined.

…

Each Port reports the L0s and L1 exit latency (the time that they require to transition their Receive Lanes from the L0s or L1 state to the L0 state) in the L0s Exit Latency and the L1 Exit Latency configuration fields, respectively. If a Port doesn't support L0s or ASPM L1, the value of the respective exit latency field is undefined.

~~If L1 state is not supported for ASPM (as reported in the ASPM Support field), the L1 Exit latency field is ignored.~~

Modify Section 7.8.6 as shown:

7.8.6. Link Capabilities Register (Offset 0Ch)

…

Table 7-14: Link Capabilities Register

Bit Location	Register Description	Attributes
…	**…**	…
11:10	**Active State Power Management (ASPM) Support** – This field indicates the level of ASPM supported on the given PCI Express Link. See Section 5.4.1 for ASPM support requirements. Defined encodings are: 00b ~~Reserved~~No ASPM Support 01b L0s ~~Entry~~ Supported 10b ~~Reserved~~L1 Supported 11b L0s and L1 Supported Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO
14:12	**L0s Exit Latency** – This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. If L0s is not supported, the value is undefined; however, see the Implementation Note "Potential Issues With Legacy Software When L0s is Not Supported" in Section 5.4.1.1 for the recommended value. …	RO
17:15	**L1 Exit Latency** – This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0. If ASPM L1 is not supported, the value is undefined. ...	RO
…	**…**	…
22	**ASPM Optionality Compliance** – This bit must be set to 1b. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests.	HwInit

~~If the L1 state is not supported for ASPM (as reported in the ASPM Support field), then the L1 Exit latency field is ignored.~~

Use of the ASPM Optionality Compliance Bit

Correct implementation and utilization of ASPM can significantly reduce Link power. However, ASPM feature implementations can be complex, and historically, some implementations have not been compliant to the specification. To address this, some of the ASPM optionality and ASPM entry requirements from earlier revisions of this document have been loosened. However, clear pass/fail compliance testing for ASPM features is also supported and expected.

The ASPM Optionality Compliance bit was created as a tool to establish clear expectations for hardware and software. This bit is Set to indicate hardware that conforms to the current specification, and this bit must be Set in components compliant to this specification.

System software as well as compliance software can assume that if this bit is Set, that the associated hardware conforms to the current specification. Hardware should be fully capable of supporting ASPM configuration management without needing component-specific treatment by system software.

For older hardware that does not have this bit Set, it is strongly recommended for system software to provide mechanisms to enable ASPM on components that work correctly with ASPM, and to disable ASPM on components that don't.

Modify Section 7.8.7 as shown:

7.8.7. Link Control Register (Offset 10h)

…

Table 7-15: Link Control Register

Bit Location	Register Description	Attributes

1:0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM ~~supported~~ enabled on the given PCI Express Link. See Section 5.4.1.3 for requirements on when and how to enable ASPM.
	
	Defined encodings are:
	
	  00b Disabled
	  01b L0s Entry Enabled
	  10b L1 Entry Enabled
	  11b L0s and L1 Entry Enabled
	
	Note: "L0s Entry Enabled" ~~indicates~~ enables the Transmitter ~~entering~~ to enter L0s ~~is supported~~. ~~The~~ If L0s is supported, the Receiver must be capable of entering L0s even when the ~~field~~ Transmitter is disabled from entering L0s (00b or 10b).
	
	…
		RW

Modify Section 7.14.2 as shown:

7.14.2. Root Complex Link Capabilities Register

…

<p align="center">**Table 7-57: Root Complex Link Capabilities Register**</p>

Bit Location	Register Description	Attributes
…	**…**	…
11:10	**Active State Power Management (ASPM) Support** – This field indicates the level of ASPM supported on the given Link. Defined encodings are: 00b No ASPM Support 01b L0s ~~Entry~~ Supported 10b L1 ~~Entry~~ Supported 11b L0s and L1 Supported	RO
14:12	**L0s Exit Latency** – This field indicates the L0s exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. <u>If L0s is not supported, the value is undefined.</u> Defined encodings are: …	RO
17:15	**L1 Exit Latency** – This field indicates the L1 exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from <u>ASPM </u>L1 to L0. <u>If ASPM L1 is not supported, the value is undefined.</u> Defined encodings are: …	RO

Modify Section 7.14.3 as shown:

7.14.3.   Root Complex Link Control Register

…

<p align="center">**Table 7-58: Root Complex Link Control Register**</p>

Bit Location	Register Description	Attributes

1:0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM ~~supported~~ enabled on the given Link.
	
	Defined encodings are:
	
	   00b Disabled
	
	   01b L0s Entry Enabled
	
	   10b L1 Entry Enabled
	
	   11b L0s and L1 Entry Enabled
	
	Note: "L0s Entry Enabled" ~~indicates~~ enables the Transmitter ~~entering~~ to enter L0s ~~is supported~~. ~~The~~ If L0s is supported, the Receiver must be capable of entering L0s even when the ~~field~~ Transmitter is disabled from entering L0s (00b or 10b).
	
	Default value of this field is implementation specific.
	
	Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s, as indicated by their ASPM Support field values. Otherwise, the result is undefined.
	
	ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. …
	
	…
		RW