PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	Latency Tolerance Reporting
DATE:	22 January 2008, updated 14 August 2008
AFFECTED DOCUMENTS:	PCI Express Base Specification version 2.0
SPONSORS:	Intel

Part I

1. Summary of the Functional Changes

This ECR proposes to add a new mechanism for Endpoints to report their service latency requirements for Memory Reads and Writes to the Root Complex such that central platform resources (such as main memory, RC internal interconnects, snoop resources, and other resources associated with the RC) can be power managed without impacting Endpoint functionality and performance.

Current platform Power Management (PM) policies guesstimate when devices are idle (e.g. using inactivity timers). Guessing wrong can cause performance issues, or even hardware failures. In the worst case, users/admins will disable PM to allow functionality at the cost of increased platform power consumption.

This ECR impacts Endpoint devices, RCs and Switches that choose to implement the new optional feature.

2. Benefits as a Result of the Changes

Using this mechanism, platform power management can be improved by using actual Endpoint service requirements when determining platform PM behaviors.

3. Assessment of the Impact

This is an optional normative capability.

Endpoints implementing the capability must determine their Request latency tolerance requirements and report these using the new Message described. A new Extended Capability structure is defined to limit the maximum latency values sent by an Endpoint.

Switches implementing the capability must support coalescing LTR Messages as described below.

Root Complexes implementing the capability must provide mechanisms for receiving and interpreting LTR Messages.

4. Analysis of the Hardware Implications

LTR requires new hardware and is an optional normative capability. Hardware that is not LTR capable will continue to operate as it does today.

5. Analysis of the Software Implications

LTR requires new software to enable this functionality and thus is optional normative. Software that does not comprehend the new functionality will interoperate with LTR capable hardware per the existing PCI Express Base Specification. Software must not enable LTR in an Endpoint unless all Upstream Switches and the Root Complex indicate support for LTR. Software is responsible for enabling this feature on Endpoints and the Ports to which they are connected during a hotplug event. Software is responsible for

programming maximum latency registers in a new LTR Extended Capability, which specify the maximum no-snoop and snoop latencies that each device is permitted to request. Software should set these to the platform's maximum supported latencies or less.

Part II

Detailed Description of the change

Add Section 2.2.8.x:

2.2.8.x. Latency Tolerance Reporting (LTR) Message

The LTR Message is optionally used to report device behaviors regarding its tolerance of Read/Write service latencies. Refer to Section <6.x> for details on LTR. The following rules apply to the formation of the LTR Message:

- Table <LTR1> defines the LTR Message.
- The LTR Message does not include a data payload (the TLP Type is Msg).
- The Length field is Reserved.
- The LTR Message must use the default Traffic Class designator (TC0). Receivers that implement LTR support must check for violations of this rule. If a Receiver determines that a TLP violates this rule, it must handle the TLP as a Malformed TLP.
  - This is a reported error associated with the Receiving Port (see Section 6.2).

Table <LTR1>: LTR Message

Name	Code[7:0] (b)	Routing r[2:0] (b)	Support[1]				Req ID	Description/Comments
			RC	Ep	Sw	Br		
LTR	0001 0000	100	r	t	tr		BD	Latency Tolerance Reporting

Note to reviewers: the Req ID column in the above table is deleted by the ARI ECN, and should be deleted when this ECN is integrated into future Base specifications.

	+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			Reserved				TD	EP	Attr		R				Length							
Byte4	Requester ID																Tag								Message Code							
Byte8	Reserved																Reserved															
Byte12	No-Snoop Latency																Snoop Latency															

Figure <LTR1>: LTR Message

[1] Support for LTR is optional. Functions that support LTR must implement the reporting and enable mechanisms described in Chapter 7.

Add Section 6.x:

6.x. Latency Tolerance Reporting (LTR) Mechanism

The Latency Tolerance Reporting (LTR) mechanism enables Endpoints to report their service latency requirements for Memory Reads and Writes to the Root Complex, so that power management policies for central platform resources (such as main memory, RC internal interconnects, and snoop resources) can be implemented to consider Endpoint service requirements. The LTR Mechanism does not directly affect Link power management or Switch internal power management, although it is possible that indirect effects will occur.

The implications of "latency tolerance" will vary significantly between different device types and implementations. When implementing this mechanism, it will generally be desirable to consider if service latencies impact functionality or only performance, if performance impacts are linear, and how much it is possible for the device to use buffering and/or other techniques to compensate for latency sensitivities.

The Root Complex is not required to honor the requested service latencies, but is strongly encouraged to provide a worst case service latency that does not exceed the latencies indicated by the LTR mechanism.

LTR support is discovered and enabled through reporting and control registers described in Chapter 7. Software must not enable LTR in an Endpoint unless the Root Complex and all intermediate Switches indicate support for LTR. Note that it is not required that all Endpoints support LTR to permit enabling LTR in those Endpoints that do support it. When enabling the LTR mechanism in a hierarchy, devices closest to the Root Port must be enabled first, then moving downwards towards the leaf Endpoints.

If an LTR Message is received at a Root Port that does not support LTR or if LTR is not enabled, the Message must be treated as an Unsupported Request.

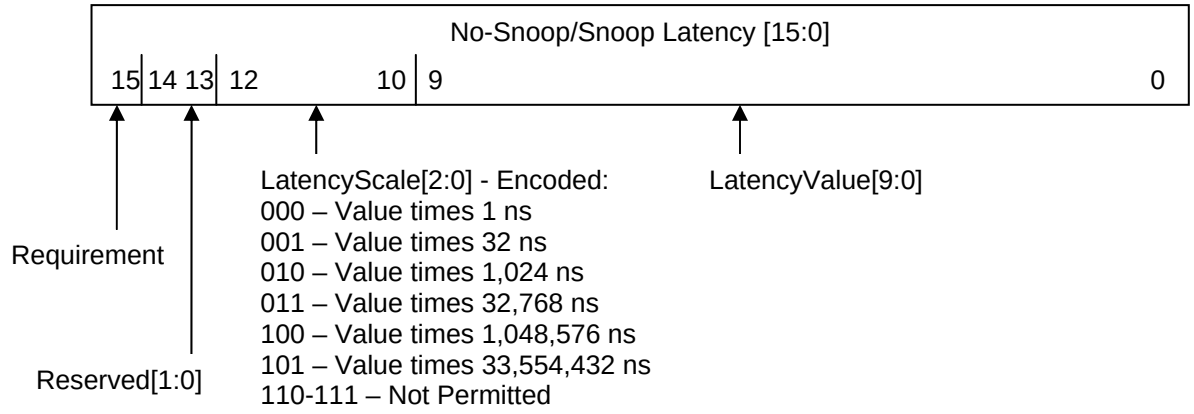


Figure <LTR2>: Latency Fields Format for LTR Messages

No-Snoop Latency & **Snoop Latency**: As shown in Figure <LTR2>, these fields include a Requirement bit that indicates if the device has a latency requirement for the given type of Request. With any LTR Message transmission, it is permitted for a device to indicate that a requirement is being reported for only no-snoop Requests, for only snoop Requests, or for both types of Requests. It is also permitted for a device to indicate that it has no requirement for either type of traffic, which it does by clearing the Requirement bit in both fields.

Each field also includes value and scale fields that encode the reported latency. Values are multiplied by the indicated scale to yield an absolute time value, expressible in a range from 1 ns to $2^{25}*(2^{10}-1)$ = 34,326,183,936 ns.

Setting the value and scale fields to all 0's indicates that the device will be impacted by any delay and that the best possible service is requested.

If a device doesn't implement or has no service requirements for a particular type of traffic, then it must have the Requirement bit clear for the associated latency field.

When directed to a non-D0 state by a Write to the PMCSR register, if a device had previously reported one or both latency fields with the Requirement bit set, it must send a new LTR Message with both Requirement bits clear prior to transitioning to the non-D0 state.

When the LTR Enable bit is cleared, if a device had previously reported one or both latency fields with the Requirement bit set, it must send a new LTR Message with both Requirement bits clear.

An LTR Message from a device reflects the tolerable latency from the perspective of the device, for which the platform must consider the service latency itself, plus the delay added by the use of Clock Power Management (CLKREQ#), if applicable. The service latency itself is defined as follows:

- If the device issues a Read Request, latency is measured as delay from transmission of the END symbol in the Request TLP to the receipt of the STP symbol in the first Completion TLP.
- If the device issues one or more Write Requests such that it cannot issue another Write Request due to Flow Control backpressure, the latency is measured from the transmission of the END symbol of the TLP that exhausts FC credit to the receipt of the SDP symbol of the DLLP returning more credits.

If Clock Power Management is used, then the platform implementation-dependent period between when a device asserts CLKREQ# and the device receives a valid clock signal constitutes an additional component of the platform service latency that must be comprehended by the platform when setting platform power management policy.

It is recommended that Endpoints transmit an LTR Message shortly after the LTR capability is enabled, and ideally before issuing any other Request TLPs.

It is strongly recommended that Endpoints send no more than two LTR Messages within any 500 µs time period. Downstream Ports must not generate an error if more than two LTR Messages are received within a 500 µs time period.

Multi-Function devices (MFDs) associated with an Upstream Port must transmit a "conglomerated" LTR Message Upstream according to the following rules:

- The acceptable latency values for the Message sent Upstream by the MFD must reflect the lowest values associated with any Function.
    - It is permitted that the snoop and no-snoop latencies reported in the conglomerated Message are associated with different Functions.
    - If none of the Functions report a requirement for a certain type of traffic (snoop/no-snoop), the Message sent by the MFD must not set the Requirement bit corresponding to that type of traffic.

- The MFD must transmit an LTR Message Upstream when any Function of the MFD changes the values it has reported internally in such a way as to change the conglomerated value earlier reported by the MFD.

Switches must collect the Messages from Downstream Ports and transmit a "conglomerated" Message Upstream according to the following rules:

- If a Switch supports the LTR feature, it must support the feature on its Upstream Port and all Downstream Ports.
- A Switch must only transmit LTR Messages when the LTR Mechanism Enable bit is set at the Upstream Port.
- The acceptable latency values for the Message sent Upstream by the Switch must reflect the lowest values received from any Downstream Port.
    - When any Downstream Port reports a LatencyValue of all 0's (regardless of the LatencyScale value), the Message sent Upstream must report a LatencyScale of 000b and a LatencyValue of all 0's.
    - If none of the Downstream Ports receive an LTR Message containing a requirement for a certain type of traffic (snoop/no-snoop), the Message sent by the switch must not set the Requirement bit corresponding to that type of traffic.
    - Any additional latency induced by the Switch must be accounted for in the conglomerated Message. A Switch must ensure that its Link and internal power management and other internal operation shall not cause its conglomerated latency to be reduced by more than 20% of the lowest received latency.
- If any Downstream Port reports a field(s) for which the Requirement bit is clear, or uses a Not Permitted LatencyScale value, that Port must not be considered when determining the corresponding field(s) reported in the Message sent Upstream.
    - Valid, permitted values for other fields must still be considered.
- When a Switch Downstream Port goes to DL_Down status, the latencies recorded for that Port must be treated as invalid, and the latencies to be transmitted Upstream updated and a new conglomerated Message transmitted Upstream if the conglomerated latencies are changed as a result.
- If a Switch Downstream Port has the LTR Mechanism Enable bit cleared, the Latency Tolerance values recorded for that Port must be treated as invalid, and the latencies to be transmitted Upstream updated and a new conglomerated Message transmitted Upstream if the conglomerated latencies are changed as a result.
- A Switch must transmit an LTR Message Upstream when any Downstream Port / Function changes the latencies it has reported in such a way as to change the conglomerated latency reported by the Switch.
- A Switch must not transmit LTR Messages Upstream unless triggered to do so by one of the events described above.

The RC is permitted to delay processing of device Request TLPs provided it satisfies the device's service requirements.

When the latency requirement is updated during a series of Requests, it is required that the updated latency figure be comprehended by the RC no later than the larger of either (a) waiting as long as the

previously indicated latency or (b) following the servicing of a subsequent Request. It is permitted for the RC to comprehend the updated latency figure earlier than this limit.

IMPLEMENTATION NOTE

Optimal Use of LTR

It is recommended that Endpoints transmit an updated LTR Message each time the Endpoint's service requirements change. If the latency tolerance is being reduced, it is recommended to transmit the updated LTR Message ahead of first anticipated Request with the new requirement, allowing the amount of time indicated in the previously issued LTR Message. If the tolerance is being increased, then the update should immediately follow the final Request with the preceding latency tolerance value.

Typically, the Link will be in ASPM L1, and, if Clock Power Management (Clock PM) is supported, CLKREQ# will be deasserted, at the time an Endpoint reaches an internal trigger that causes the Endpoint to initiate Requests to the RC. The following text shows an example of how LTR is applied in such a case. Key time points are illustrated in the following figure:

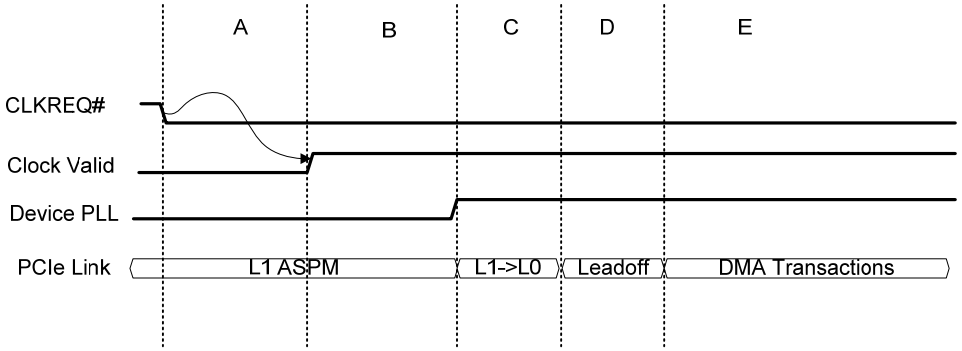


Figure <LTR3>: CLKREQ# and Clock Power Management

Time A is a platform implementation-dependent period between when a device asserts CLKREQ# and the device receives a valid clock signal. This value will not exceed the latency in effect.

Time B is the device implementation-dependent period between when a device has a valid clock and it can initiate the retraining sequence to transition from L1 ASPM to L0.

Time C is the period during which the transition from L1 ASPM to L0 takes place. This value will not exceed the maximum L1 exit latency reported by the Endpoint.

Time D for a Read transaction is the time between the transmission of the END symbol in the Request TLP to the receipt of the STP symbol in the Completion TLP. Time D for a Write transaction is the time between the transmission of the END symbol of the TLP that exhausts FC credit to the receipt of the SDP symbol in the DLLP returning more credits. This value will not exceed the latency in effect.

Time E is the period where the data path from the Endpoint to system memory is open, and data transactions are not subject to the leadoff latency.

The LTR latency semantic reflects the tolerable latency seen by the device as measured by one or both of the following:

Case 1: the device may or may not support Clock PM, but has not deasserted its CLKREQ# signal – The latency observed by the device is represented in Figure <LTR3> as time D.

Case 2: the device supports Clock PM and has deasserted CLKREQ#- The latency observed by the device is represented as the sum of times A and D.

To effectively use the LTR mechanism in conjunction with Clock PM, the device will know or be able to measure times B and C, so that it knows when to assert CLKREQ#. The actual values of Time A and Time D may vary dynamically, and it is the responsibility of the platform to ensure the sum will not exceed the latency.

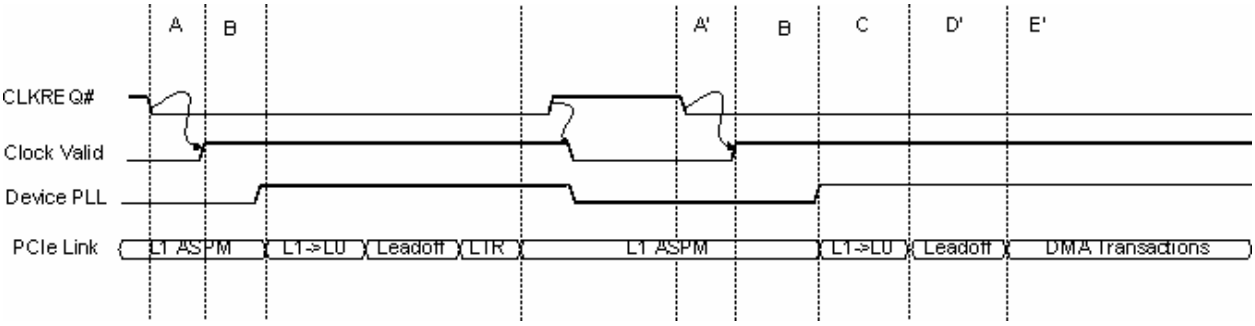


Figure <LTR4>: Use of LTR and Clock Power Management

In a very simple model, an Endpoint may choose to implement LTR as shown in Figure <LTR4>. When an Endpoint determines that it is idle, it sends an LTR Message with the software configured maximum latency or the maximum latency the Endpoint can support.

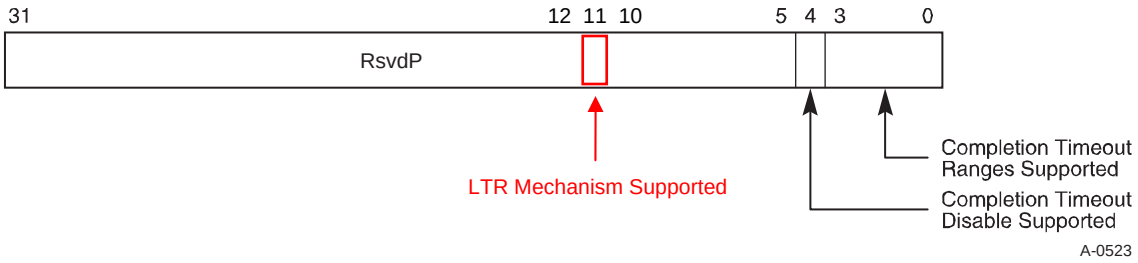
When the Endpoint determines that it has a need to maintain sustained data transfers with the Root Complex, the Endpoint sends a new LTR Message with a shorter latency (at Time E). This LTR Message is sent prior to the next data flush by a time equal to the maximum latency sent before (the time between Time E and Time D'). In between Time E and Time A', the Endpoint can return to a low power state, while the platform transitions to a state where it can provide the shorter latency when the device next needs to transmit data.

Note that the RC may delay processing of device Request TLPs, provided it satisfies the device's service requirements. If, for example, an Endpoint connected to Root Port 1 reports a latency tolerance of 100 µs, and an Endpoint on Root Port 2 report a value of 30 µs, the RC might implement a policy of stalling an initial Request following an idle period from Root Port 1 for 70 µs before servicing the Request with a 30 µs latency, thus providing a perceived service latency to the first Endpoint of 100 µs. This RC behavior provides the RC the ability to batch together Requests for more efficient servicing.

It is recommended that Endpoints buffer Requests as much as possible, and then use the full Link bandwidth in bursts as long as the Endpoint can practically support, as this will generally lead to the best overall platform power efficiency.

Note that LTR may be enabled in environments where not all Endpoints support LTR, and in such environments, Endpoints that do not support LTR may experience suboptimal service.

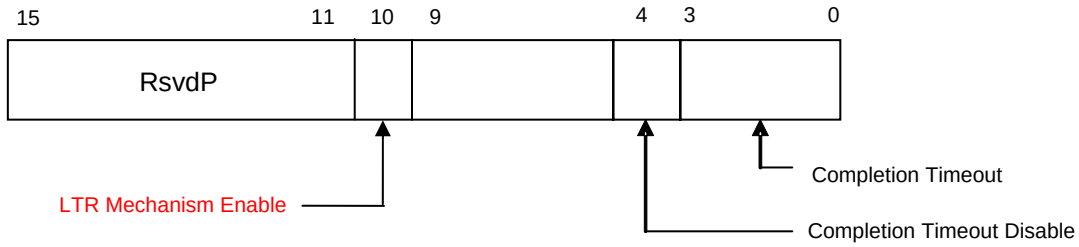
7.8.15. Device Capabilities 2 Register (Offset 24h)



...

Bit Location	Register Description	Attributes
...		
11	**LTR Mechanism Supported** – A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism capability. Root Ports, Switches and Endpoints are permitted to implement this capability. For a multi-Function device associated with an Upstream Port, each Function must report the same value for this bit. For Bridges, Downstream Ports, and components that do not implement this capability, this bit must be hardwired to 0b.	RO

7.8.16. Device Control 2 Register (Offset 28h)



...

Bit Location	Register Description	Attributes
...		
10	**LTR Mechanism Enable** – When Set to 1b, this bit enables the Latency Tolerance Reporting (LTR) mechanism. For a Multi-Function device associated with an Upstream Port of a device that implements LTR, the bit in Function 0 is RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is RsvdP. Components that do not implement LTR are permitted to hardwire this bit to 0b. Default value of this bit is 0b. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status.	RW / RsvdP

7.xx. Latency Tolerance Reporting (LTR) Capability

The PCI Express Latency Tolerance Reporting (LTR) Capability is an optional Extended Capability that allows software to provide platform latency information to components with Upstream Ports (Endpoints and Switches), and is required if the component supports LTR. It is not applicable to Root Ports or Downstream Ports in a Switch.

For a multi-Function device associated with the Upstream Port of a component that implements LTR, this Capability structure must be implemented only in Function 0, and must control the component's Link behavior on behalf of all the Functions of the device.

31 16	15 0	Byte Offset
PCI Express Extended Capability Header		00h
Max No-Snoop Latency Register	Max Snoop Latency Register	04h

Figure <LTR4>: LTR Extended Capability Structure

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

Next Capability Offset		PCI Express Extended Capability ID
	Capability Version	

Figure <LTR5>: LTR Extended Capability Header

Table <LTR2>: LTR 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 for the LTR Extended Capability is 0018h.	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. Max Snoop Latency Register (Offset 04h)

15 13	12 10	9 0
RsvdP	Max Snoop LatencyScale	Max Snoop LatencyValue

Figure <LTR7>: Max Snoop Latency Register

Table <LTR4>: Max Snoop Latency Register

Bit Location	Register Description	Attributes
9:0	**Max Snoop LatencyValue —**Along with the Max Snoop LatencyScale field, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. The default value for this field is 0.	RW
12:10	**Max Snoop LatencyScale —** This register provides a scale for the value contained within the Maximum Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. See Section 6.x. The default value for this field is 0.	RW

7.xx.3. Max No-Snoop Latency Register (Offset 06h)

15 13	12 10	9 0
RsvdP	Max No-Snoop LatencyScale	Max No-Snoop LatencyValue

Figure <LTR6>: Max No-Snoop Latency Register

Table <LTR3>: Max No-Snoop Latency Register

Bit Location	Register Description	Attributes
9:0	**Max No-Snoop LatencyValue —** Along with the Max No-Snoop LatencyScale field, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. The default value for this field is 0.	RW
12:10	**Max No-Snoop LatencyScale —** This register provides a scale for the value contained within the Max No-Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. See Section 6.x. The default value for this field is 0.	RW