Energy-Efficient Platforms

White Paper

Designing Devices Using the New Power Management Extensions for Interconnects

July 2009

Revision 1.0

Document Number: 322304-001

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

ENERGY STAR denotes a system level energy specification, defined by the US Environmental Protection Agency, that relies upon all of the system's components, including processor, chipset, power supply, HDD, graphics controller and memory to meet the specification. For more information, see http://www.energystar.gov/index.cfm?fuseaction=find_a_product.showProductGroup&pgw_code=CO.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
322304	001	• Initial release.	July 2009

§

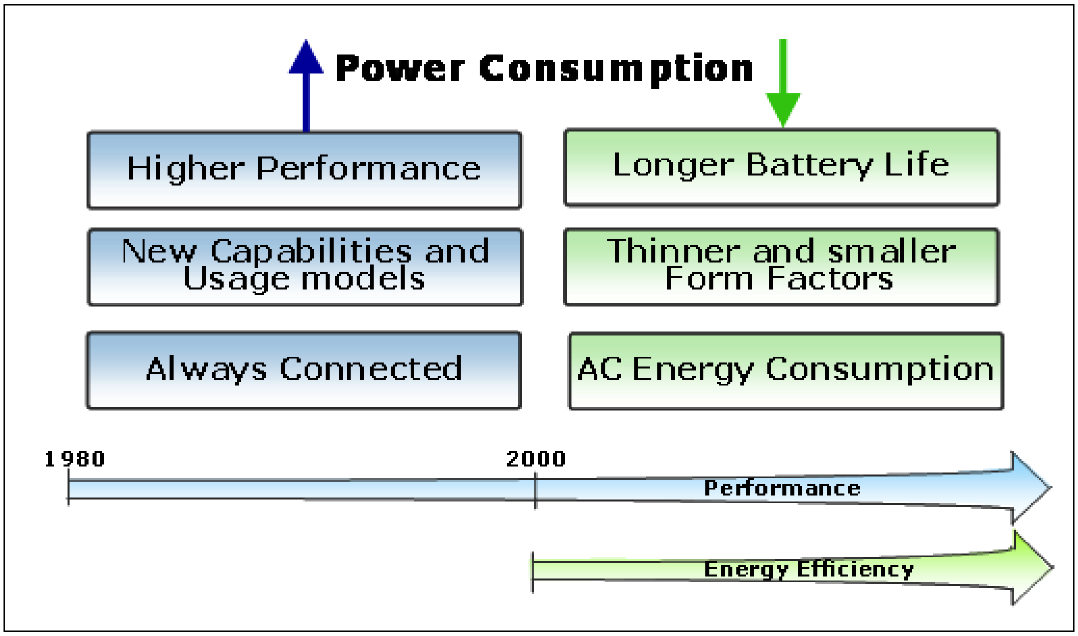
1 Introduction

1.1 Overview

Ever since the personal computing (PC) platforms were launched in the 1980s, there has been a constant demand for increased performance. Computing performance has followed Moore's Law during the last three decades, allowing consumers to have the performance, capability and connectivity undreamt of just a few years ago. But these advancements have also contributed to an increase in system-level energy consumption in spite of remarkable gains on processing efficiency.

Starting around 2000, the popularity and demand of mobile computing platforms have been steadily rising. Consumers of mobile platforms demand the same performance as desktops but also view battery life and small form factors as very important usability factors. With Energy Star* active and idle power requirements, system level power consumption is not just a battery life issue.

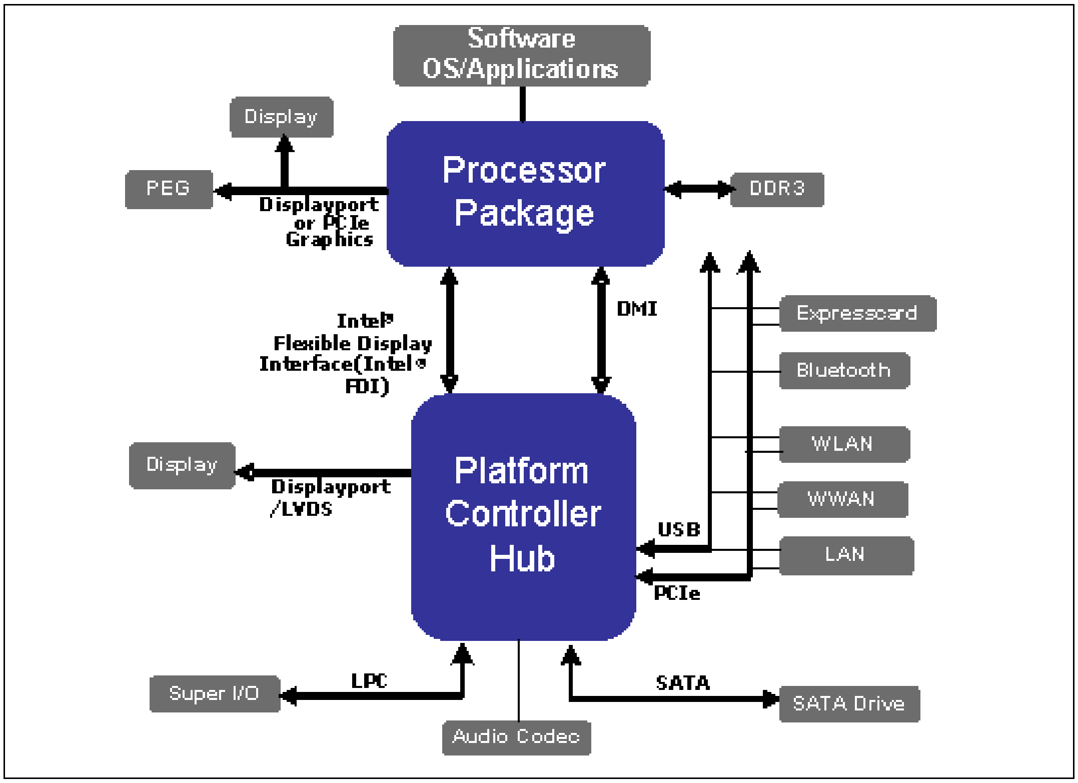
Figure 1. Importance of Performance AND Energy-Efficiency Is Growing



1.2 Energy Efficiency in Open Systems

Intel® Architecture (IA) platforms are open systems where operating systems, software applications and hardware devices are created and sold by other vendors.

Figure 2. Platform Ecosystem

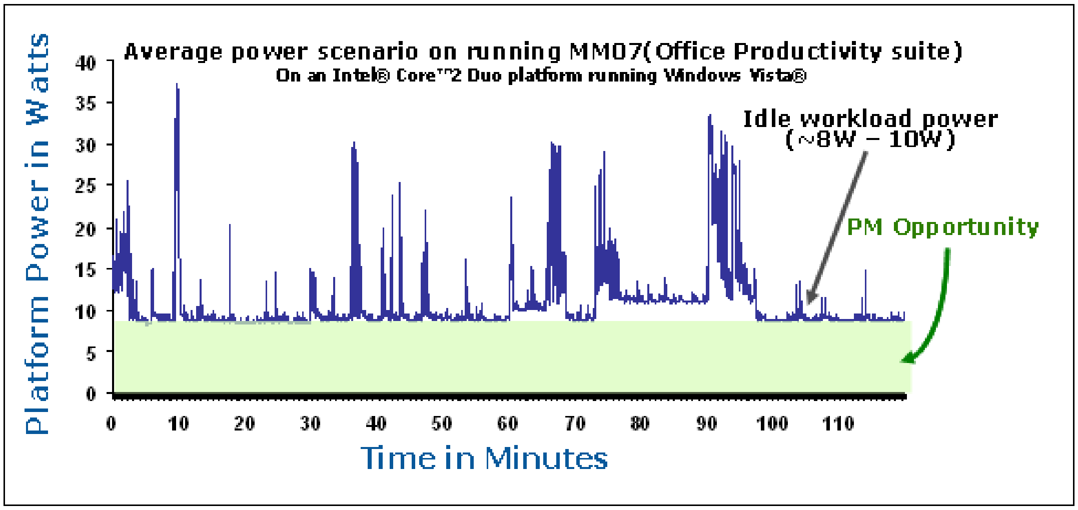


In spite of remarkable progress in processor power management and efforts to address the power efficiency of other platform components, a single ill-behaving device or application can impede all these benefits by preventing the platform components from residing in low power states. Enabling long periods of idle time even with increasing functionality is crucial for improving energy efficiency. Platform level energy efficiency requires all components in the platform ecosystem to cooperate.

1.3 Typical Mobile Platform Power Profile

Usage analysis has shown that a typical mobile platform in S0 working state is doing nothing for about 95% of the time as measured by the CPU C0 state residency. The platform in this idle state still consumes about 8 – 10 W of power due to large portions of system resources being kept powered up for best performance.

Figure 3. Typical Mobile Platform Power Profile in S0 State

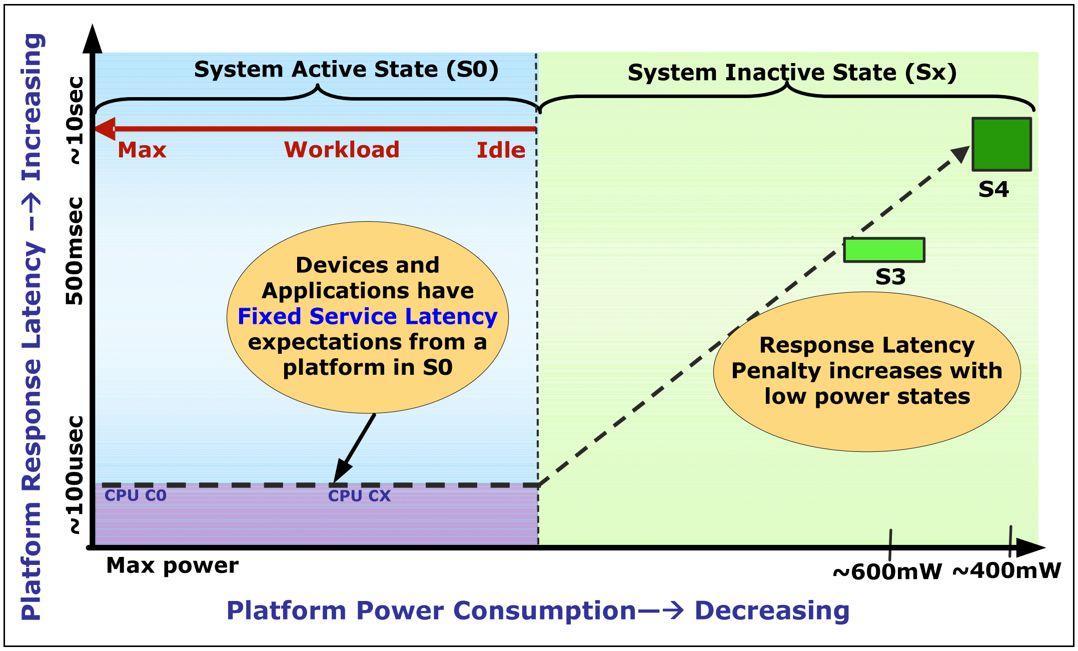


There can be significant platform power savings with a scalable architecture. Keeping system resources powered on when doing useful work and powered down when idle gives a good balance between performance and energy consumption. The vision is to increase energy efficiency by dynamically powering off large portions of the platform while the system is in the operational (but idle) S0 state for extended periods of time. This can close the gap between contemporary idle power (~8 W – 10 W) and sleeping power (~500 mW).

1.4 Responsiveness and Power Management

Platform power management in today’s systems gets course-grained guidance from the operating system. But the operating system cannot give fine-grained guidance due to unpredictability of bus master activity or asynchronous interrupt activity initiated by devices. Hence platform power management controllers use internal heuristics and inactivity timers to do fine-grained power management.

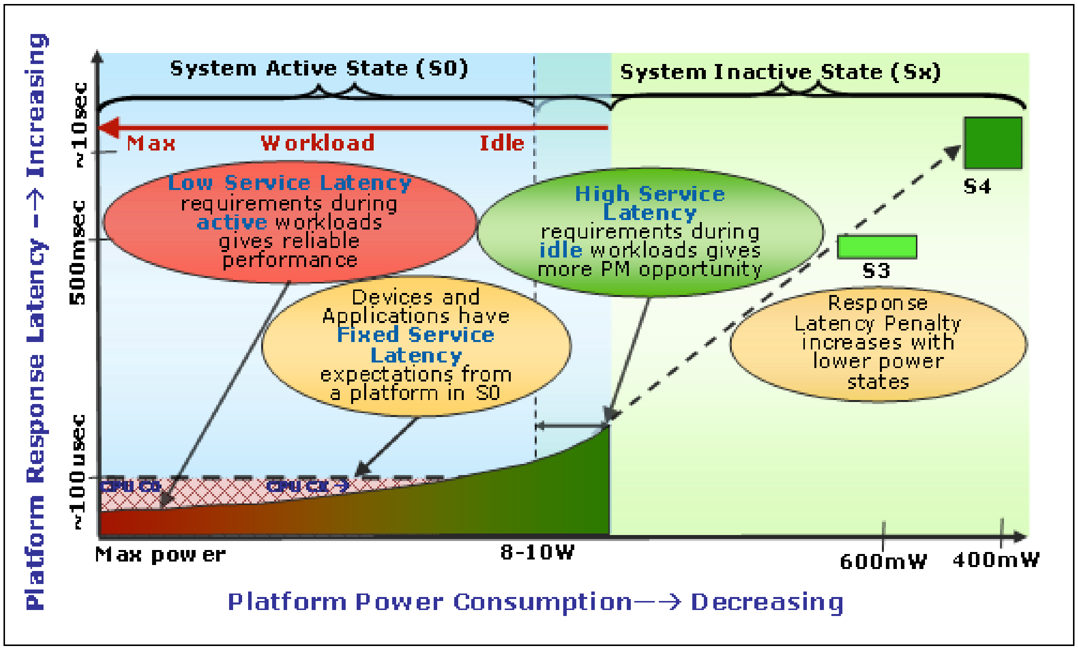
Figure 4. Fixed Service Latency Expectations in Today's Platforms



There is a platform response latency penalty seen by devices when a platform goes into lower power states. The platform response latency increases with each progressively lower power state. This is observed across the entire range of power management states. An example at a system level is with ACPI S-states: S3 standby (suspend to RAM) has a much faster response (exit) latency compared to S4 hibernate (suspend to disk). An example at a lower level is with processor C-states: the CPU HLT C1 state has a much faster response (exit) latency compared to the CPU C6 state.

Utilizing lower power states causes longer response latencies introducing a "Quality of Service (QoS)" issue. Today's systems provide devices a fixed minimal platform response latency (~50 μs for mobile, <5 μs for desktop and <1 μs for server) when the platform is in the functional ACPI S0 state, and the device interconnect is active. Using power saving techniques that extend platform response latencies beyond this minimal amount may cause performance issues or even device failures.

Figure 5. Variable Service Latency Expectations Based on Workload



Platform Response Latency --> Increasing
~10sec
500msec
~100usec
System Active State (S0)
System Inactive State (Sx)
Max
Workload
Idle
Low Service Latency requirements during active workloads gives reliable performance
High Service Latency requirements during idle workloads gives more PM opportunity
Devices and Applications have Fixed Service Latency expectations from a platform in S0
Response Latency Penalty increases with lower power states
S4
S3
CPU C0
CPU Cx ->
Max power
8-10W
600mW
400mW
Platform Power Consumption--> Decreasing

If the platform were provided with dynamic device latency requirements, then the platform would be capable of doing the following:

- Enter deeper power saving states with larger response latency when QoS is less constrained
- Enter low power states with smaller response latency states when QoS is more constrained

As shown in Figure 5, this enables the platform to lower its power consumption much more than what is possible in today's platforms during idle workloads without sacrificing performance.

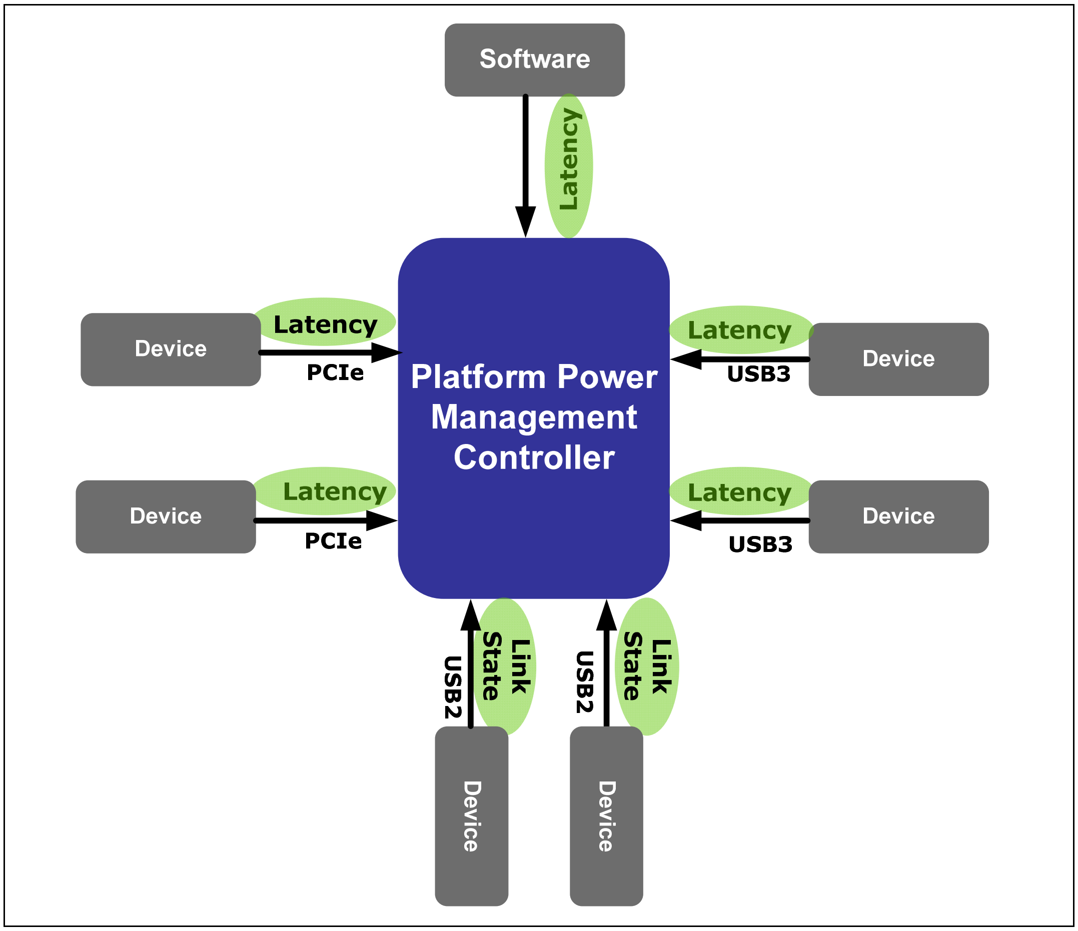
§

2 Interconnect Power Management Extensions

2.1 Dynamic Latency-Based Infrastructure

The ability to reliably and effectively employ deeper platform power management states when idle is key to improving platform energy efficiency. An important part of this is ascertaining when it is safe to incur delays related to power management transitions. One way to achieve this is by providing an infrastructure throughout the system which enables all components to provide their service latency requirements. This can be used by the platform power management controller to determine the depth of platform power management state that may be entered at any given point in time.

Figure 6. Dynamic Latency-Based Infrastructure



Conveyance of dynamic service latency requirements by platform ecosystem components enables the following:

- Aggressive power management that is reliable across all workloads. Certain devices, workloads and applications are sensitive to platform response latencies. Platform power management controllers not aware of these requirements are forced to adopt very conservative policies all the time to avoid reliability issues. When a platform PM controller is kept abreast of device latency constraints, it can adopt the best power management depth possible within those constraints.
- Opportunity to further reduce power during idle workloads. Deeper power states can be safely entered without degradation to performance or power state thrashing.

2.1.1 Device Interconnects

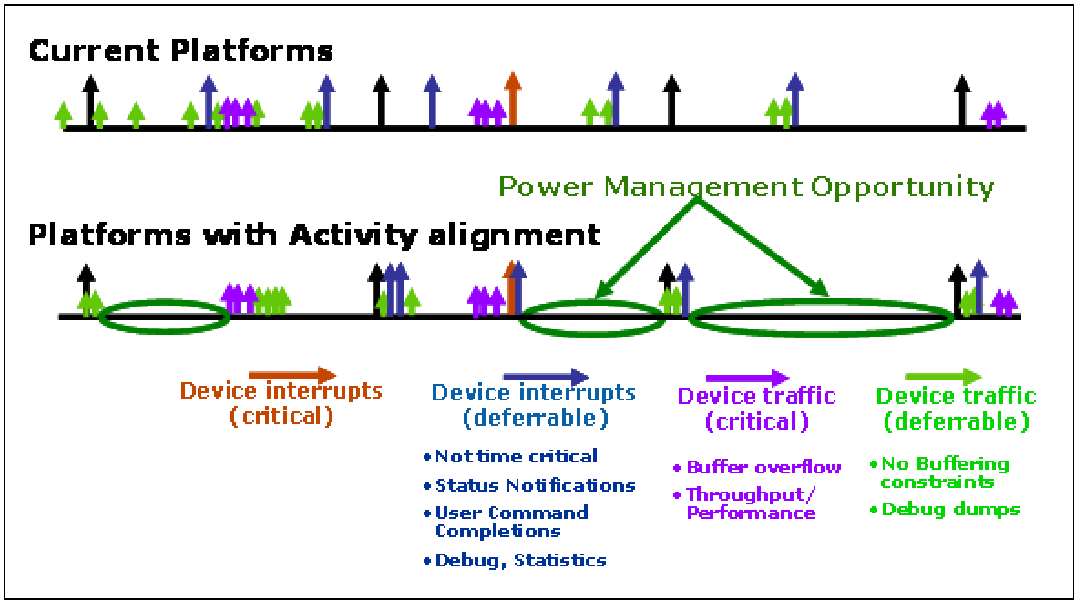
Two general methods have been defined for devices to convey their latency tolerance requirements:

1. ***Link Power Management (LPM) States:*** The LPM state implicitly conveys to the platform whether the device is idle. Link states such as SATA Partial and Slumber states and USB2 LPM L1 and Suspend states will be translated into latency requirements by the host controller.
2. ***Latency Tolerance Messages:*** Interconnects such as PCIe Gen2/Gen3 and USB3 have defined new messages to convey the latency requirements. This allows for the device latency tolerance to be decoupled from the link states thus enabling latencies to be conveyed more dynamically and at finer resolution.

2.2 Platform Activity Alignment

Asynchronous and frequent activity from multiple devices can generate a complex access pattern with very short idle periods, preventing optimal platform power management. There is an inability to reduce platform power even at very light loads. If DMA accesses and Interrupts across multiple devices are aligned into bursts, the idle periods are extended across the platform creating more opportunity for power savings.

Figure 7. Platform Activity Alignment

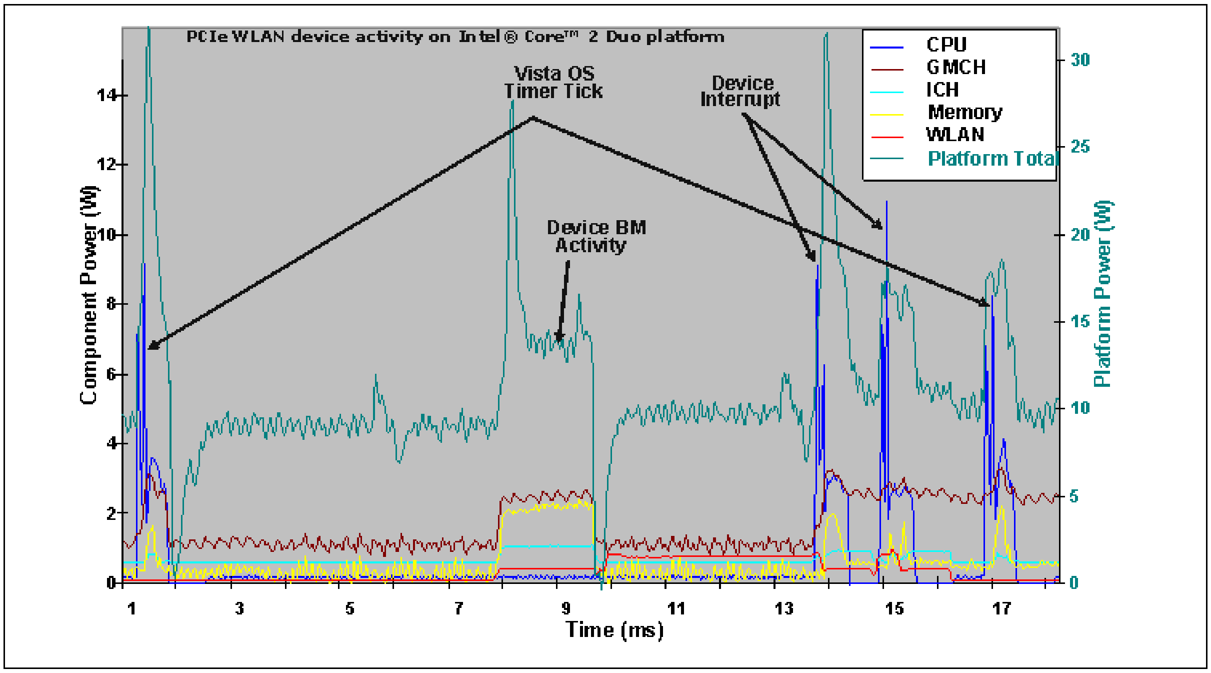

Current Platforms
Power Management Opportunity
Platforms with Activity alignment
Device interrupts (critical)
Device interrupts (deferrable)
• Not time critical
• Status Notifications
• User Command Completions
• Debug, Statistics
Device traffic (critical)
• Buffer overflow
• Throughput/ Performance
Device traffic (deferrable)
• No Buffering constraints
• Debug dumps

Information about optimal platform activity windows for device bus mastering and interrupt activity would be broadcast throughout the system providing an opportunity for devices to shape their traffic to these windows. The primary purpose of this technique is to present activity to the processor, memory and other system components in a manner that promotes energy efficiency without violating any performance or QoS constraints.

As Figure 7 illustrates, current platforms exhibit enough activity from multiple sources that would preclude the platform from doing efficient power management. If devices were able to coalesce activity by intelligent buffering and deferring non-critical events, and align activity to other platform events such that all activity occurs in bursts with long periods of idleness in between, the platform would be able to use deeper power management states during these idle periods.

2.3 Designing Devices for Platform Energy-Efficiency

Figure 8. Impact of Device Activity on Platform Power



Good device behavior is important for platform energy efficiency. It is recommended that all devices participating in the energy-efficiency paradigm follow these recommendations:

- ***Take platform power impact into consideration***

  It is very important to analyze how specific device design decisions impact the power consumption of both the device and the rest of the platform. Every bus master access or interrupt from a device brings several high power components in the platform out of a low power state. A device should consider the following aspects to reduce the impact on platform power.

  - Ensure there is sufficient buffering to generate traffic bursts with periods of idleness in between. The period of idleness should at least be 300 µs for meaningful power savings
  - Move all fine-grain control to the device to reduce frequency of interactions with the platform
  - When idle, should not cause platform to consume additional power

- ***Devices dynamically determine their service latency requirements for the platform and convey these dynamically to the platform PM controller.***

  Devices having stringent response latency requirements from the platform indicate a lower latency requirement when active and a higher latency tolerance when idle. Devices convey this information to the platform using the new Interconnect Power Management Extensions.

- Avoid generating frequent interrupts and temporally scattered and frequent bus master accesses to system DRAM

  Devices coalesce DMA activity into bursts and align the burst traffic to platform activity windows. Devices also defer traffic that is not critical.

§

3 PCI Express* Devices

3.1 PCIe Latency Tolerance Requirement (LTR)

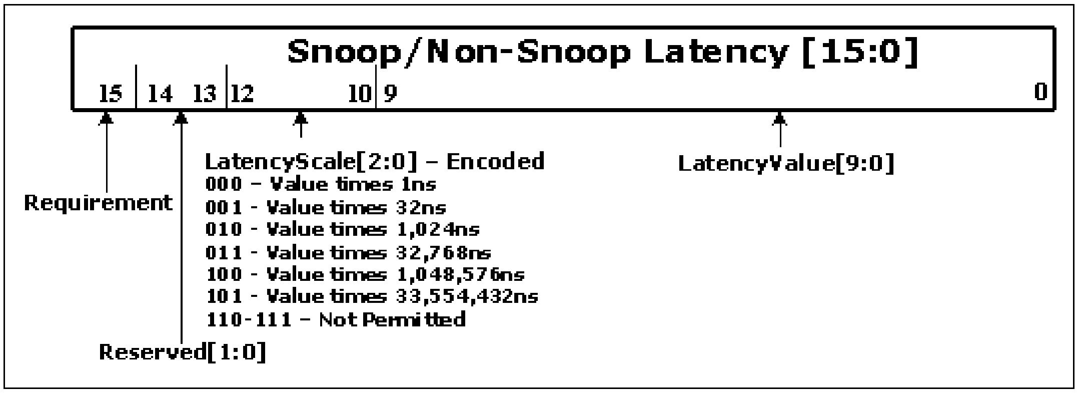
An Engineering Change Request (ECR) was approved by PCIe SIG to add explicit latency tolerance messaging referred to as Latency Tolerance Requirements Reporting (LTR), as an extension to PCIe Gen2 and will be a native part of PCIe Gen3. The LTR mechanism enables PCIe endpoints to explicitly convey their service latency requirements for memory reads and writes to the root complex. LTR support is discovered and enabled by software/firmware through various reporting and control registers.

The LTR message is used by devices to convey their service latency requirements to upstream entities (switches, root complex, etc.). The latency values specified entail a budget for the entire path between the device and main memory – excluding link exit latencies and any device induced latencies that are dealt with internally by the device.

LTR feature requires software/firmware support to enable this capability. 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 in the endpoints and the ports to which they are connected in a hotplug event. Software is also responsible for programming the platform specific 'Maximum Latency Register' in the Extended Capability Structure.

3.2 LTR Reporting Guidelines for Client Platforms

Figure 9. LTR Latency Field


Snoop/Non-Snoop Latency [15:0]
15
14
13
12
10
9
0
Requirement
LatencyScale[2:0] – Encoded
000 – Value times 1ns
001 - Value times 32ns
010 - Value times 1,024ns
011 - Value times 32,768ns
100 - Value times 1,048,576ns
101 - Value times 33,554,432ns
110-111 – Not Permitted
LatencyValue[9:0]
Reserved[1:0]

- If a device has no latency requirements, it shall send an LTR message with the 'Requirement' bit set to 0.
- It is recommended that devices do not generate more than 2 LTR messages in a 500-μs window. These messages give guidance to platform power management. Receiving messages too frequently from multiple devices will not be efficient and cause power state trashing.
- Some events that cause a device to send an LTR message are:
  - Device moving to an ACPI D0 state, LTR feature enabled
  - When device is in D0 state, activity level changes
  - Device moves out of D0 state, LTR feature disabled
- The table below outlines the latency tolerance guidelines for device-initiated access to main memory for devices residing in an operational (D0) state. Note that all devices must support at least 5-μs latency tolerance at all times to ensure correctness, where a minimum of 100-μs is strongly recommended.

Table 1. Device Latency Tolerance Guidelines for Client Platforms

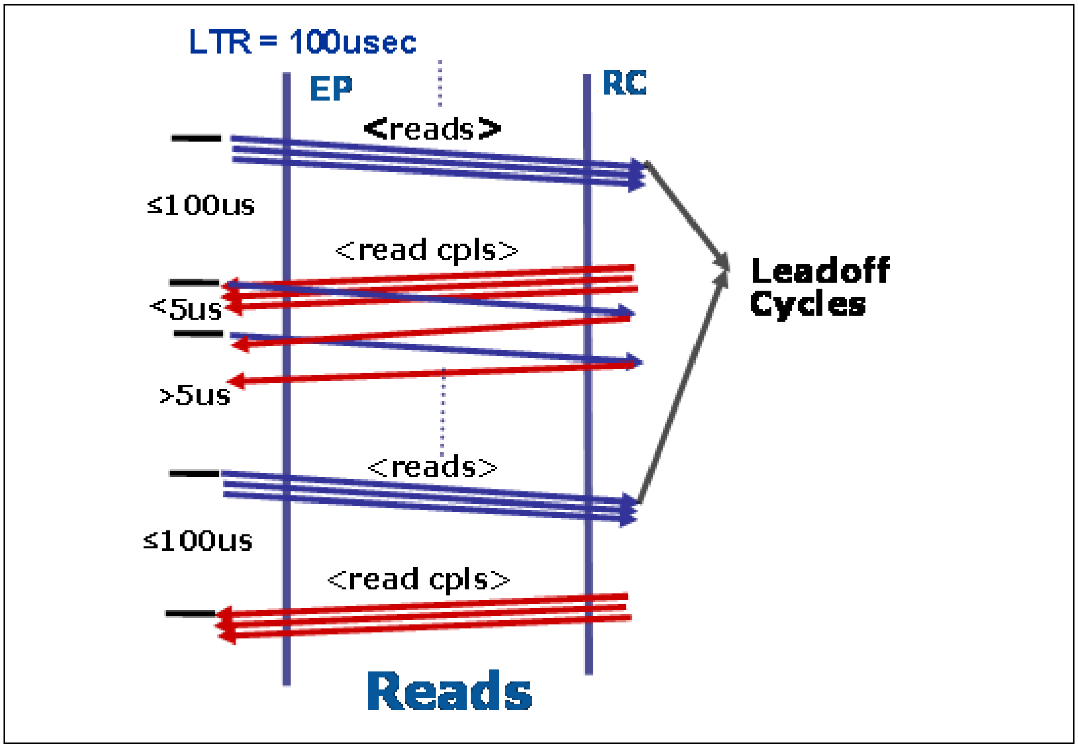
Device Phase	Latency Tolerance	Notes
Low-Latency (Active)	5 to 20 μs	Intended only for use by specific devices with limited buffering combined with high data rates, and only when absolutely necessary to prevent data loss or other critical failures. Note a minimum of 5-μs latency tolerance is required by all devices at all times, but higher values (e.g., 20 μs) are obviously preferred. Devices which consistently operate at stringent values will significantly impact (increase) platform power consumption.
Normal (Active, Light Idle)	100 μs	The recommended latency tolerance for most devices under active to pseudo-idle workloads. Allows the platform to employ relatively deep PM states.
Pervasively Idle (Idle)	Max Platform Latency Value (<1 ms)	The recommended latency tolerance when a device is pervasively idle, generally characterized as the absence of meaningful activity for many milliseconds to seconds. Facilitates the use of the deepest platform idle PM states.

3.3 LTR Semantics for Reads and Writes

The latency values in the LTR message are only applicable to 'leadoff cycles' where the leadoff cycle is the first memory transaction of potentially multiple memory transactions that will occur in quick succession (<5 µs). For transactions that are not leadoff cycles, no power management delays will be introduced by the platform.

3.3.1 Endpoint Initiated Memory Reads and Completions

Figure 10. Endpoint Initiated Memory Reads

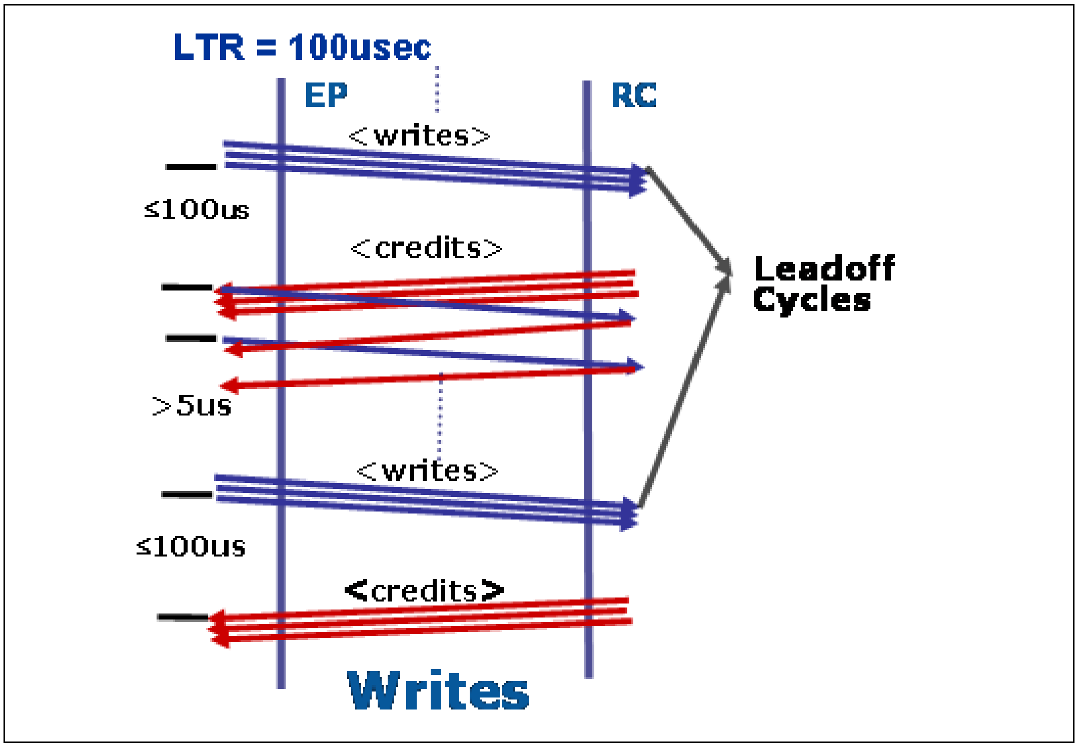


If an endpoint initiates a memory read transaction that is a leadoff cycle, there might be a delay in platform response with the delay not exceeding the latency value sent by the device in the last LTR message. If subsequent requests are pipelined, these transactions would also see the delays incurred while the platform is coming out of low power state. The endpoint must initiate the next transaction within 5 us of the root complex initiated memory read completion to ensure that this transaction does not see any power management latency.

Note: Even when a device does not see any power management related latencies, it might see latencies dependant on the memory bandwidth and other platform device activity, though these will not be as large as power management latencies.

3.3.2 Endpoint Initiated Memory Writes and Flow Control

Figure 11. Endpoint Initiated Memory Writes

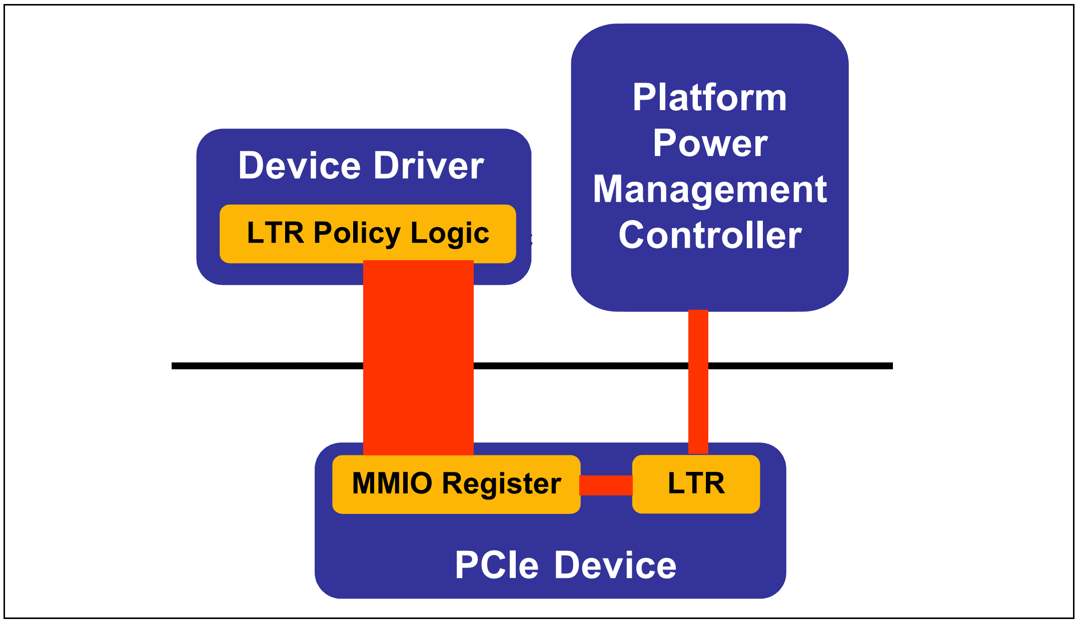


If an endpoint initiates a memory write transaction that is a leadoff cycle, there might be a delay in platform response with the delay not exceeding the latency value sent by the device in the last LTR message. For subsequent endpoint initiated transactions to avoid any power management latency, the transaction must be initiated within 5 µs of the memory write or flow control transaction.

3.4 Software-Guided Latency Messages

The PCIe Latency Tolerance Requirement reporting (LTR) extensions, allow the latency requirement of the device to be communicated to the platform power management controllers without generating an interrupt to the platform. This is efficient as the CPU is not brought to the high power executing state to process power management messages. But for some devices which are not latency sensitive or change their latency requirements very infrequently, a software guided model, as shown in Figure 12, may be preferable.

Figure 12. Software Guided LTR Messages



Devices fall into three categories depending on the frequency of their latency requirement changes:

- ***Static*** – These devices do not have stringent latency requirements. They can tolerate maximum platform response latency at all times.
- ***Slow Dynamic*** – The latency tolerance for these devices changes infrequently.
- ***Fast Dynamic*** – The latency requirement for these devices change frequently and these devices have stringent latency constraints.

Static and slow dynamic devices may choose to implement the LTR policy logic in software. The device hardware LTR logic will send LTR messages upstream based on the guidance given by software. One way to implement this would be to have a memory-mapped IO (MMIO) register in the device. A write to this register by software would trigger an LTR message to be sent.

3.5 LTR Usage Examples

3.5.1 Ethernet Adapter

Unlike most devices on the platform which are slave devices (like storage, graphics, etc) and initiate activity when commanded by the platform, Ethernet LAN devices receive data via the Ethernet link from remote generators of network traffic and are therefore sensitive to platform response latencies. These latencies can sometimes cause data loss due to buffer overflows, especially at higher data rates.

Figure 13. Ethernet Adapter in ACPI D0 State Sending LTR Messages

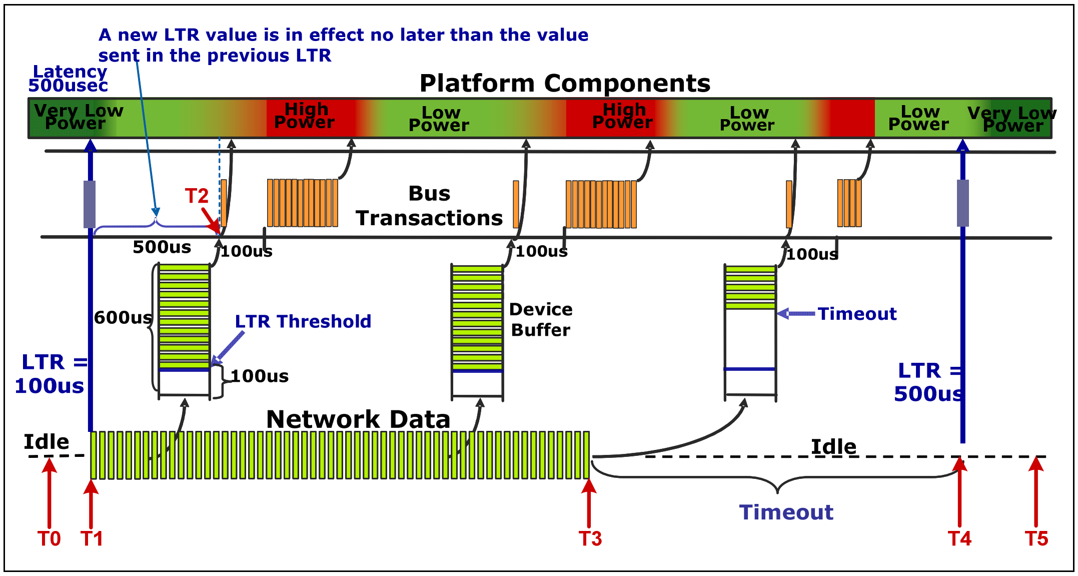


Figure 13 shows an example of an Ethernet LAN adapter sending LTR values when in the ACPI D0 state. At the start of the example at time T0, the adapter is idle and the platform is in a very low power state. At time T1, data starts coming over the link. As soon as the adapter starts receiving data, it sends an LTR value of 100 µs. This corresponds to the excess buffer capacity and allows the platform to be sufficiently power managed between bursts of data. At time T2, the buffer has been filled to the threshold and the adapter releases the data to the platform as a burst. It might see an initial latency of up to 100 µs as indicated by the LTR value. At time T3, network activity stops. The adapter hits a timeout at time T4 and releases the data to the platform. At time T5 after an inactivity timeout, the adapter sends an LTR value of 500 µs which allows the platform to go back into a very low power state.

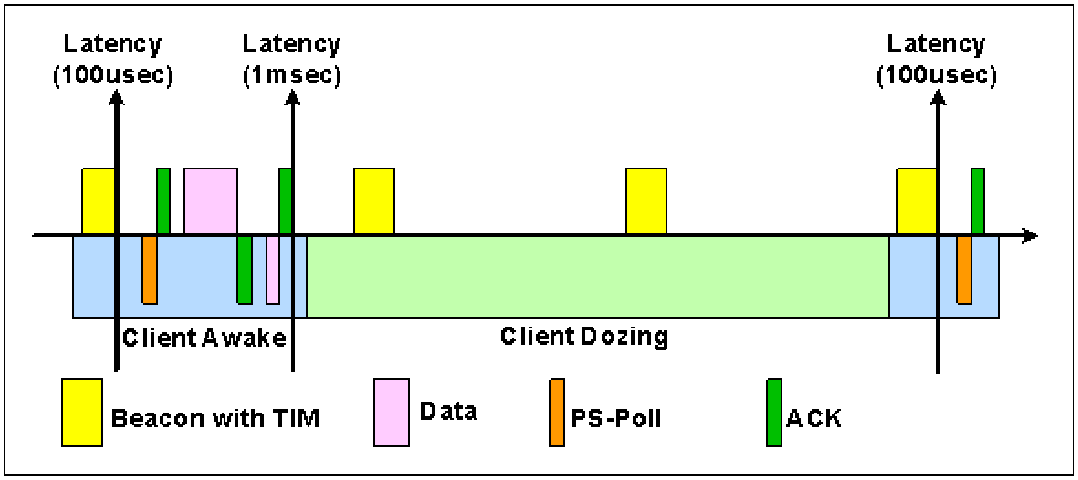
In the above example, the Ethernet LAN adapter uses its buffering as one of the metrics to give latency guidance. The latency values will also depend on data rate – 10 Mbps, 100 Mbps or 1 Gbps. Higher latency values can be sent at lower data rates. The device/device driver may also take into account the type of network traffic when determining latency values. If the adapter is in the disabled state, or if the link is down or if the adapter is not in the ACPI D0 state, the LTR message should have been sent with 'Requirement' bit set to 0.

3.5.2 WLAN Adapter

Many wireless devices like WLAN, WiMax, 3G, etc., power manage their radios when not very active. The wireless protocols inherently support power management features that allow these devices to indicate to the Access Point or Base stations that they are going into a low power (sleep) state. No data is sent to these devices when they are in this state. If these devices could send a high service latency tolerance message to the platform during these sleep states then the platform components can also be aggressively power managed. Often the amount of savings that can be got from platform components is significantly higher than the device power savings. The device can indicate a lower service latency requirement when it is ready to move data.

A WLAN device using the legacy Wi-Fi power save mode negotiates a listen interval with the Access Point. During periods of low activity, the WLAN device will indicate to the Access Point (AP) that it is going into a low power state so that the AP can buffer data that is to be received by the device. During the listen interval, the device will check the beacon to see if data is buffered and, if so, will come out of low power state. In Figure 14, an LTR message of 100 µs is sent when coming out of low power state and an LTR message of 1 ms is sent when going into low power state.

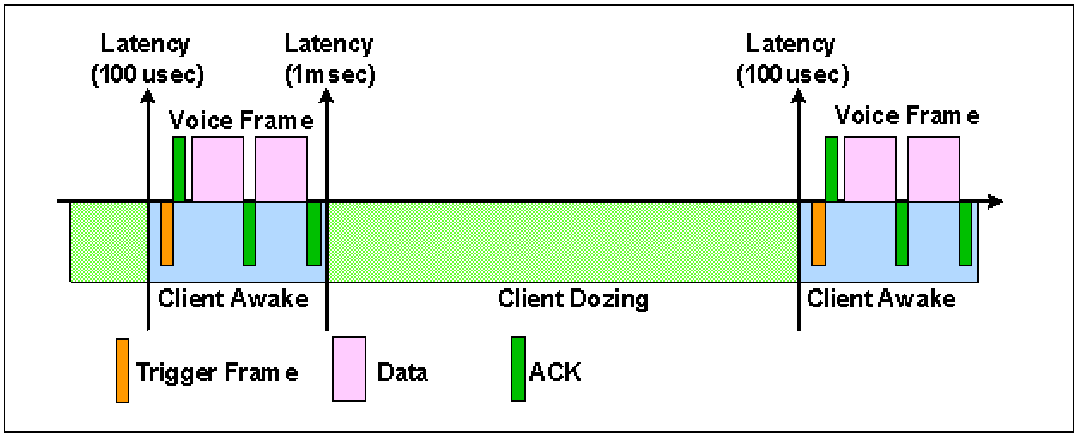
Figure 14. LTR Messages from WLAN Device Using Wi-Fi Legacy Power Save



WMM Power Save is an advanced power save mechanism which allows for optimum power management when running latency sensitive voice, audio or video applications. With WMM Power Save the WLAN client device does not wait for a Beacon frame to request a data download from the Access Point. Individual applications decide at what interval the client device needs to communicate with the AP and how long it can remain in the dozing state.

Figure 15 shows a WLAN client using WMM Power Save sending LTR messages to the host platform. It sends an LTR message of 100 µs when it comes out of doze state and needs to communicate with the AP and sends an LTR message of 1 ms prior to going into a doze state.

Figure 15. LTR Messages from WLAN Device Using WMM Power Save



3.6 PCIe Optimized Buffer Flush/Fill (OBFF)

An Engineering Change Request (ECR) was approved by PCIe SIG to add Optimized Buffer Flush/Fill (OBFF) as an extension to PCIe Gen2 and native part of PCIe Gen3. OBFF extension provides a mechanism for the platform to indicate optimal windows to endpoints for bus mastering and interrupt activity. In these windows, the incremental cost in terms of platform power consumption for the bus mastering or interrupt activity is relatively low. Typically this will correspond to the time that the host processors, memory and other platform resources are active to service some other activity on the platform such as a timer tick or bus mastering from another device.

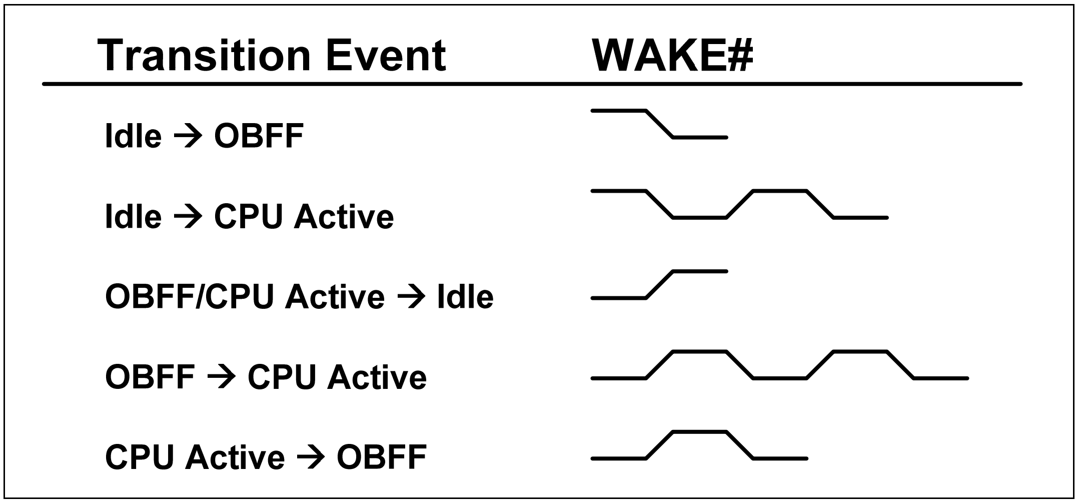
An OBFF indication is a hint – devices are still permitted to initiate bus mastering or interrupt traffic outside the optimal windows. But this will not be ideal for platform power and should be avoided as much as possible. The OBFF events are signaled using the WAKE# signal as this prevents needless link activation.

The three OBFF events are:

- ***CPU Active:*** Platform active for all actions including bus mastering and interrupts.
- ***OBFF:*** Path to main memory available for read/write bus master activities.
- ***Idle:*** Platform is in an idle, low power state.

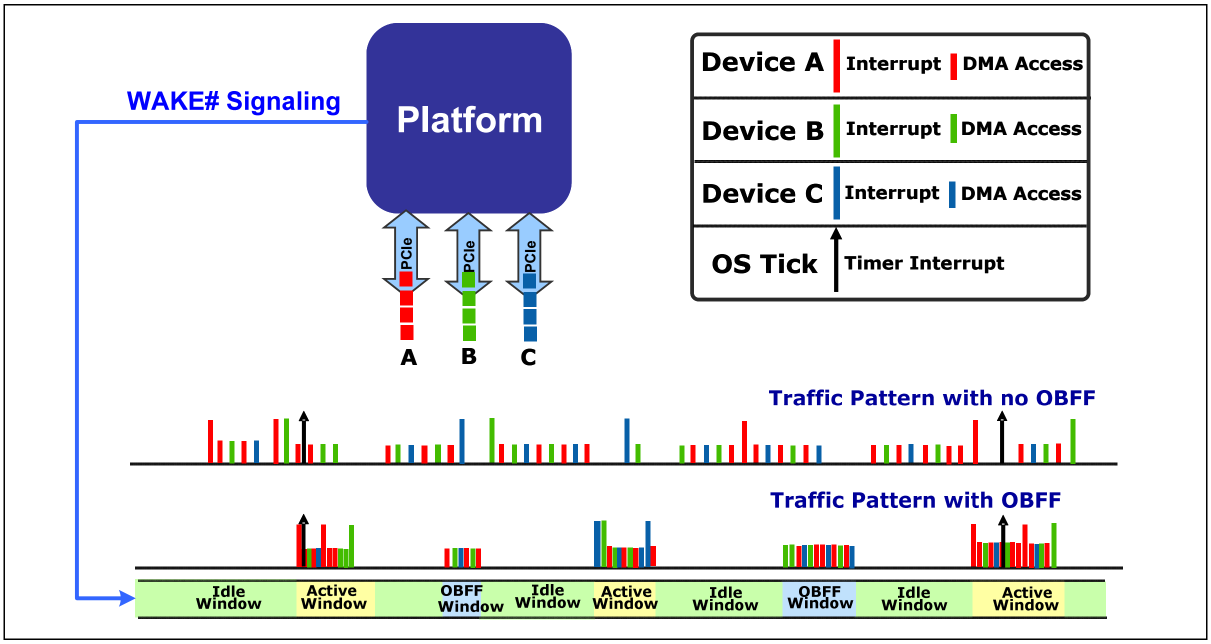
Figure 16 shows the WAKE# signaling for the OBFF transition events.

Figure 16. WAKE# Signaling for OBFF Event Transitions



OBFF support is discovered and enabled through reporting and control registers by software/firmware. Software must not enable OBFF in an endpoint unless the platform supports delivering OBFF indications to the endpoint.

Figure 17. Example of PCIe OBFF



3.7 PCIe Active State Link Power Management

The PCIe specification defines several low power link states for a device in an active (ACPI D0) state and Active State Power Management (ASPM) allows individual serial Links in a PCI Express fabric to have power incrementally reduced as a Link becomes less active. L0 is the active link state wherein transactions may be in flight; L0s is the first stage of idleness and is known as the standby state, which must be entered by a device supporting L0s in under 7 µs. ASPM L1 is the next level of power savings known as lower power standby, where the link enters a deeper level of power savings. The device can optionally power off its PLL as the PCIe specification also has the concept of turning REFCLK off (and device PLL power down) via CLKREQ# protocol coupled with L1 state.

Table 2. PCIe Link Power Management States

L-State	Allowable D-state	Description
L0	D0	Link is on and fully functional.
L0s	D0	Very low latency link state intended for aggressive use during short intervals of idleness. Power savings opportunities include the ability to clock gate much of the transceiver circuitry and link layer logic.
L1	D0, D1, D2, D3$_{hot}$	Low latency link state intended for use when devices are meaningfully idle. Power savings opportunities include the ability to power down most transceiver circuitry except host-initiated wake detection, and clock gating of all PCIe logic and device-side PLLs.

L-State	Allowable D-state	Description
L2	D3$_{cold}$	Maps to a device D3$_{cold}$ state with aux power with either sideband (WAKE#) or in-band (Beacon) wake detection. Power savings opportunities are similar but generally greater than L1 given the more relaxed resume latency requirements.
L3	D3$_{cold}$	Link off. No aux power or wake detection. Platform must go through a full boot sequence to bring the link out of this state. The link and device consume no (or negligible) power when in this state.

The PCIe specification also specifies a model for software to programmatically discover the link latency structures from the top of the system hierarchy to the endpoint and then evaluate whether the path for these latencies exceed what the device can tolerate, thereby setting the link active state power management policy accordingly on link-by-link basis.

3.7.1 Recommendations for Link State Transitions

- In current platforms, the PCIe links have power incrementally reduced as the link becomes less active, using timeout based policy for progression from L0->L0s->L1. Although the policy is simple, it is not power efficient. When a device accesses the platform in bursts, during the idle periods when the device is filling buffers the link can be efficiently transitioned from L0->L1.
- Devices should consider the use of the CLKREQ# protocol for turning REFCLK off (Device PLL power down) for additional power savings in the L1 state. If using this feature, it is important to understand the link exit latencies are critical. If link exit latencies are too long, host processor as well as peer device stalls may be observed.

3.8 Power Management Checklist for PCIe Devices

	Description	Yes/No
1	No or minimal contribution to Platform power when idle	
2	LTR support	
3	Active LTR =<100 µsec	
4	Idle LTR = Max Platform Latency	
5	OBFF BM – Bus Master activity coalescing and alignment	
6	OBFF Interrupt – Interrupt alignment and non-critical INT deferral	
7	ASPM L1 – Intelligent entry policy	
8	Use of CLKREQ# protocol	

§

4 USB 2.0 Devices

4.1 Link Power Management

The Universal Serial Bus 2.0 (USB 2.0) is a polled bus. When the device has no data to move, the device will continue to be polled if there are transactions pending at the host controller. The USB 2.0 Suspend state can save power on the USB link, but it is difficult to use dynamically due to limitations. It takes considerable time to enter and exit this state (3 ms+OS overhead for entry and 30 ms+OS overhead for exit), and devices are restricted on how much power they can consume in this state.

The L1 state is a new Link Power Management (LPM) state that addresses the deficiencies of the Suspend state (herein referred to as L2). The new low-latency LPM L1 state is intended to be used dynamically when the device is operational (ACPI D0) state, but otherwise idle and able to quickly enter and exit this low power state without disrupting normal operation.

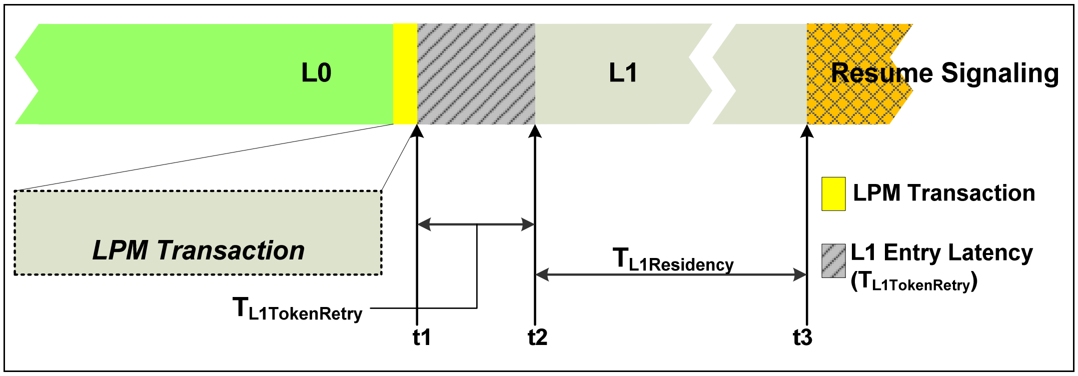
Supporting the LPM L1 state requires modifications to both USB host controllers and devices. It is backward compatible in that the new host can determine whether a device supports L1. L1 will only be used if the device acknowledges support for this feature.

Table 3. Comparison of LPM L1 and LPM L2

	L1 (Sleep)	L2 (Suspend)
Entry	Explicitly entered via LPM extended transaction	Implicitly entered via 3 ms of link inactivity
Exit	Device- or host-initiated via resume signaling; Remote-wake can be (optionally enabled/disabled via the LPM transaction	Device- or host-initiated via resume signaling; Remote-wake can be (optionally enabled/disabled by software
Signaling	Low- and full-speed idle	Low- and full-speed idle
Latencies	*Entry:* ~10 µs *Exit:* ~70 µs to 1 ms (host-specific)	*Entry:* ~3 ms *Exit:* >1 ms (OS dependent)
Link Power Consumption	~0.6 mW (data line- pull-ups)	~0.6 mW (data line- pull-ups)
Device Power Consumption	Device power consumption level is application/implementation specific	Device consumption is limited to ≤2.5 mA
Hot Removal	Natively detected per USB 2.0 mechanisms	Natively detected per USB 2.0 mechanisms

LPM L1 entry can only be initiated by the host by an explicit LPM transaction that the device can acknowledge by an ACK transaction or reject by an NYET transaction. Both device and host initiated wake events are supported from the LPM L1 state.

Figure 18. LPM L1 Transaction and Transition to L1



The host platform communicates to the device the duration of how long the host will drive resume when it initiates exit from L1 via the HIRD (Host initiated Resume Duration) parameter. This field indicates to the device the depth of platform power management. The HIRD value is a 4-bit encoded value. 0000b = 50 µs, each increment adds 75 µs (50 µs, 125 µs, 200 µs…1.2 ms). Longer values imply that the platform is going into deeper low power states. The device should take advantage of the longer exit timings for power managing its internal resources.

Table 4. USB 2.0 Latency Tolerance Support

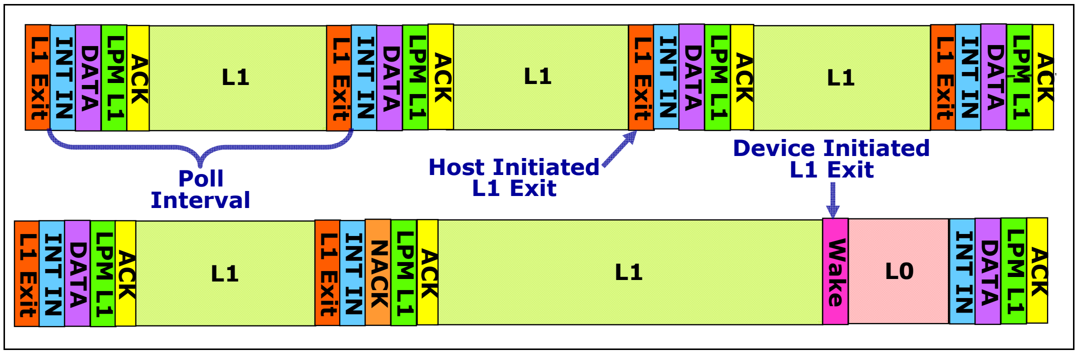
Type	Mechanism	Description
Implicit	L1 State	The latency indicated by the HIRD value can be tolerated by devices when their upstream link resides in the newly defined L1 state. Long latencies can also be tolerated for L2 state.
Explicit	–	There are currently no plans to support explicit latency tolerance messaging.
Related Documents		USB 2.0 ECN: USB 2.0 Link Power Management Addendum

4.2 LPM L1 Usage Guidelines

4.2.1 Devices with Periodic Endpoints

For devices with periodic endpoints, if there is sufficient time between polls, the host controller will place the link in L1 state immediately after the poll. For active endpoints where data transfer occurs at every poll, the host controller will bring the link out of L1 state to L0 state so that timely service is guaranteed. A periodic device should always acknowledge the L1 entry request.

Figure 19. LPM L1 Usage for USB 2.0 Devices with Periodic Endpoints

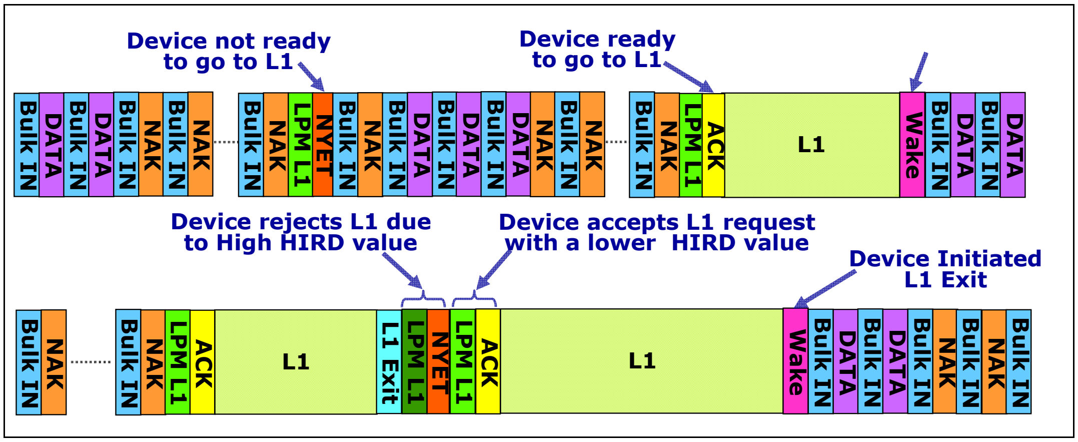


For Interrupt endpoints that are idle and do not have data to transfer, the host controller will put the link into L1 state after the NAK transaction and the endpoint will not be polled at the next poll interval. The link will continue to stay in the L1 state till the device initiates an L1 exit. This feature is not valid for isochronous endpoints which are expected to transfer data at every service interval.

4.2.2 Devices with Bulk Endpoints

For devices with Bulk endpoints, the host controller will initiate an LPM L1 transaction after some number of NAK responses to Bulk IN or Bulk OUT/PING transactions.

Figure 20. LPM L1 Usage for USB 2.0 Devices with Bulk Endpoints



The device can ACK or reject the L1 entry request based on knowledge of its activity. It can also reject a request for L1 entry if the value of HIRD is high and the device requires lower exit latencies. The host controller may then retry L1 entry with lower HIRD value.

4.3 Power Management Checklist for USB 2.0 Devices

	Description	Yes/No
1	Avoid continuous Bulk EP Polling – even when idle	
2	Minimize polling rate for Interrupt Endpoints	
3	Use Isochronous Endpoints for Streaming devices	
4	Use Selective Suspend dynamically for long periods of inactivity	
5	Use LPM L1 aggressively	

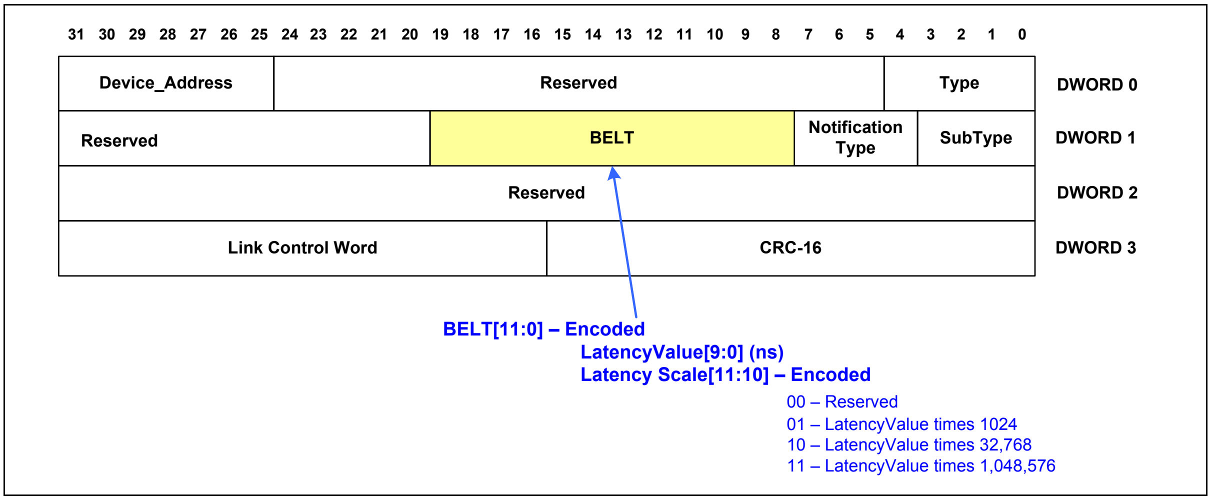
§

5 USB 3.0 Devices

5.1 Latency Tolerance Messaging (LTM)

The USB 3.0 specification defines a Latency Tolerance Messaging (LTM) transaction packet which enables a device to indicate its service latency requirements to the platform.

Figure 21. USB 3.0 Latency Tolerance Messaging (LTM)



The Best Effort Latency Tolerance (BELT) field in the transaction packet defines how much platform response latency the device can tolerate. It represents the time between when a host receives an ERDY packet and when it responds by initiating an IN or an OUT transaction associated with that ERDY packet. Devices update their BELT value based on activity level – a higher BELT value when idle and a lower BELT value when active. Devices indicate whether they are capable of sending LTM packets via a device capability descriptor. The feature is enabled by software.

5.2 LTM Reporting Guidelines for Client Platforms

Table 5. USB 3.0 BELT Values

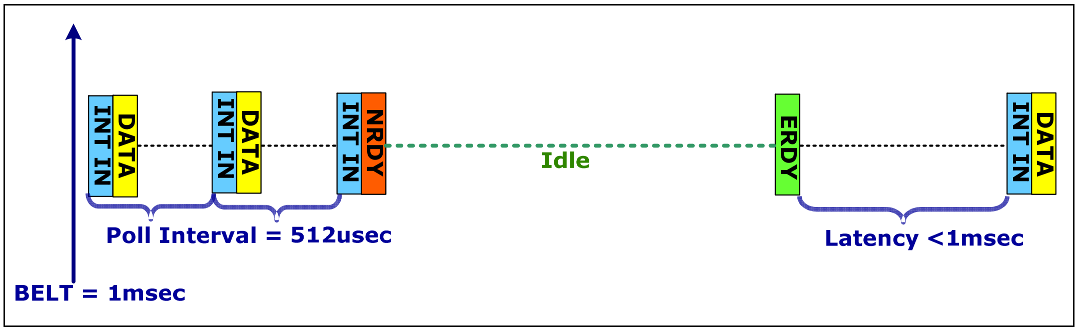
Name	Description	Min	Max	Units
tBELTdefault	Default value for BELT	1		ms
tBELTmin	Minimum value of BELT allowed in a Latency Tolerance Message	125		µs

- All devices are assumed to support a BELT value of 1 ms by default. If a device cannot tolerate the default BELT value, it will send an LTM message indicating its requirements.
- The minimum BELT value allowed is 125 µs.
- The BELT value applies to the entire device. Devices report the lowest value across all functions and endpoints.
- It is recommended that devices not send more than 2 LTM messages within a 1-ms period. Receiving LTM messages from multiple devices at high frequency will not be useful for platform power management and may cause power state trashing and hub buffer overflows.
- A device only sends a new LTM message if there is a change in service latency requirement. Each successive LTM message from a given device must have a different BELT value.
- The BELT value does not include the link exit latencies. The end to end link exit latencies are provided to the device in the U1SEL (U1 System Exit Latency) and U2SEL (U2 System Exit Latency) fields. Devices should take into account these link exit latencies along with the BELT values when considering their total latency tolerance.

5.3 LTM for Devices with Periodic Endpoints

LTM is not applicable to isochronous endpoints. Service interval is guaranteed for isochronous endpoints and the host controller will handle all power management related latencies. When a device is aggregating latency requirements across all its endpoints, it must not place any requirements for isochronous endpoints.

Figure 22. LTM for Devices with Interrupt Endpoints



For interrupt endpoints the platform will provide a response latency that depends on whether the endpoint is in a flow control state. A flow control state is entered when the endpoint sends an NRDY and remains in effect until the endpoint sends an ERDY. If the endpoint is not in a flow control state then the endpoint service interval applies. If the endpoint is in a flow control state then the larger of the endpoint service interval and the BELT value applies.

In the example above, the service interval for the endpoint is 512 µs. As long as there is a data transfer in every interval, the device will be polled every 512 µs. When there is no data to transfer, an NRDY is sent and the endpoint is placed into a flow control state. At a later time when the endpoint is ready to resume data transfer, it will indicate to the host controller to resume polling by sending an ERDY. Even though the poll interval is 512 µs, the next poll may not happen up to 1 ms later as indicated by the BELT value.

5.4 LTM for Devices with Bulk Endpoints

Figure 23. LTM for Devices with Bulk Endpoints

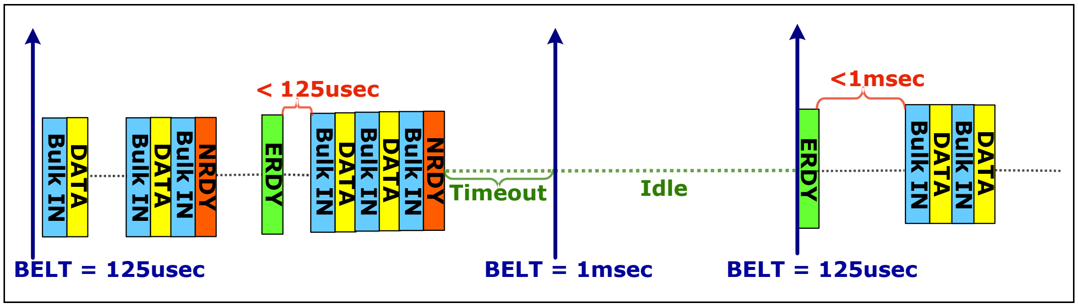
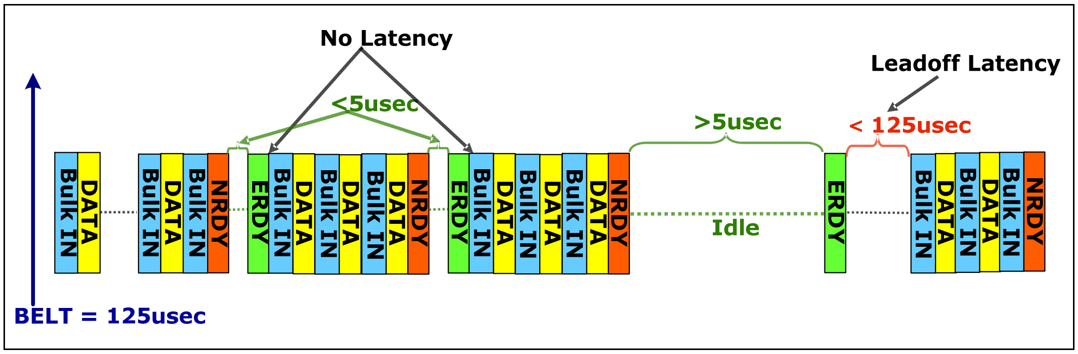


Figure 23 shows an example of a device with bulk endpoints using LTM messages. When the device moves to an active state, it sends an LTM message with a BELT value of 125 µs. When there is no data to move, the endpoint sends an NRDY packet and the endpoint is placed in the flow control state. When the endpoint is ready to move data again, it sends an ERDY packet. The host controller will respond with a Bulk IN packet no later than 125 µs.

When the device is predominantly idle (as determined by a timeout in the example above) or is aware that it has completed all data transfers (Packets pending flag not set by host), it sends an updated LTM message (BELT value of 1 ms in example above). This enables the platform to go into very low power states. When the device moves to an active state again, it sends an LTR message with BELT value of 125 µs. Since the previous BELT value was 1 ms, the platform may take up to 1 ms to respond with a BULK IN packet.

An endpoint may be placed into a flow control state for very short periods of time due to the bursty nature of data traffic. The platform may not go into low power states during these periods. The BELT values in the LTM messages are only applicable to leadoff transactions – the first transaction after the endpoint has been in the flow control state for 5 µs or longer, as shown in Figure 24.

Figure 24. BELT Value Applicable to Leadoff Transaction after Idle Period



For platform energy efficiency, it is important for devices to understand that they should burst data with idle periods in between (of ~300 µs or larger) where the endpoint is in the flow control state, the link is in low power state and the platform can go to low power states. It will not be good for platform energy-efficiency if devices burst small amounts of data frequently with small gaps between bursts where the platform just starts going into low power states and is then brought out of this state.

5.5 Link Power Management (LPM)

USB 3.0 supports multi-level link power management. The link power state may be driven by the device or by the downstream port inactivity timers that are programmable by host software. This is different from USB 2.0 LPM where the transition to low power link states is always initiated by the host. Enabling devices to initiate entry to low power link states allows for more aggressive power management of the links as the devices can put the link into lower power state immediately after data transfer completion. After U1 and U2 link states are enabled by software during configuration, the transitions in and out of these link states is handled by hardware and hence there are no additional software related latencies.

Table 6. USB 3.0 Link Power Management States

Link State	Description
U0 (On)	U0 is the normal link operational state. All packet communication, whether for control or data transfer, occurs in this state.
U1 (Idle, Fast Exit)	U1 is a low exit latency standby state (device D0). The electrical characteristics of this state allow substantial power savings in comparison with U0. Exit latency is dominated by the time to achieve receiver symbol lock and the link training process. The latency to exit this state is in the µs range.
U2 (Idle, Slow Exit)	U2 is a low to medium range exit latency standby state (device D0). Exit latencies are the same as for U1, plus clock generation (e.g., PLL) startup time if the clock generation circuitry is quiesced during U2. The latency to exit this state is typically in the msec range, but can be in the µs range.
U3 (Suspend)	U3 is a deep power saving state where portions of device (e.g., Physical Layer) power may be removed. VBUS remains active during U3. Devices may remove power from most circuitry while retaining power for circuitry needed during suspend (reset detection, host wakeup detection and remote wakeup). The latency to exit this state is in the msec range. U3 entry may only be initiated by host software.

5.6 Recommendations for Link State Transitions

Power savings resulting from the effective use of link power management can have a significant impact on platform power consumption. The link power state may be driven by the downstream port inactivity timers that are programmable by host system software or by the device based on its knowledge of traffic patterns. The USB fabric will propagate the lowest link state upwards.

When inactivity timer values are programmed by system software, the values may not be aggressive as a common value which will not impact performance for many types of devices (with different endpoints) may be selected. The USB 3.0 specification provides the following information to devices to assist with U1/U2 entry initiation when idle:

- Packets Pending (PP) Flag, used with Bulk Endpoints
- End of Burst Flag, used with Interrupt Endpoints
- Last Packet Flag, used with Isochronous Endpoints
- U1 and U2 device-to-host exit latencies

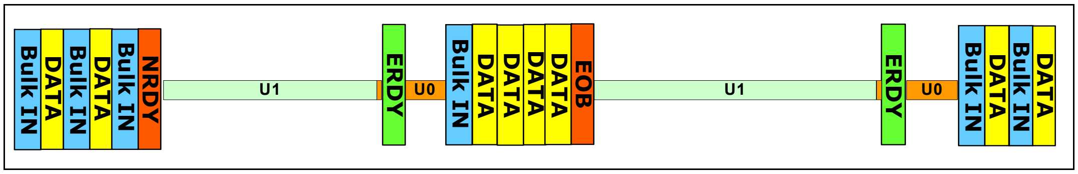
5.6.1 Devices with Bulk Endpoints

Bulk IN Endpoints

A bulk IN endpoint is in a flow control state when it returns one of the following responses to an ACK TP:

- Responding with an NRDY TP
- Sending a data packet with EOB bit set to 1 in the data packet header

Figure 25. Link Power Management for Devices with Bulk IN Endpoint



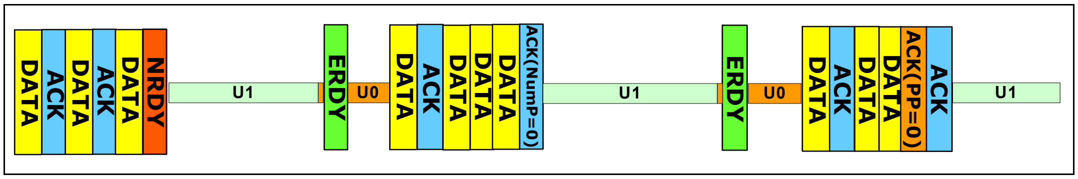
Typically, a device would put a link in U1 when it goes idle. When a device is aware of long periods of idleness as in a hard disk device spinning up a spindle, or a wireless device going into sleep mode, it may choose to put the link immediately in U2 instead of U1 for higher power savings.

Bulk OUT Endpoints

A bulk OUT endpoint is in a flow control state when it returns one of the following responses to a data packet:

- Responding with an NRDY TP
- Sending an ACK TP with the NumP field set to 0

Figure 26. Link Power Management for Devices with Bulk OUT Endpoint



The Packets Pending (PP) flag in the ACK TP is set to 1 by the host to indicate that another packet is available for the endpoint. Devices with Bulk OUT endpoints should also use this flag to determine when to put the link in a low power state.

5.6.2 Devices with Interrupt Endpoints

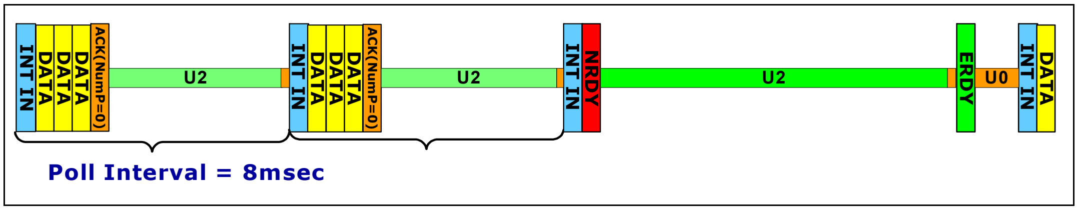
The interrupt transfer type is used for infrequent data transfers with a bounded service interval. Interrupt transactions are limited to a burst of three data packets in each service interval.

Interrupt IN Endpoints

An interrupt endpoint is in an idle state when one of the following happens:

- All the data transfer for the service interval has successfully completed
- The endpoint has no data and responds with an NRDY. The host shall not perform any more transactions to the endpoint in subsequent service intervals till the endpoint responds with an ERDY.

Figure 27. Link Power Management for Devices with Interrupt IN Endpoint

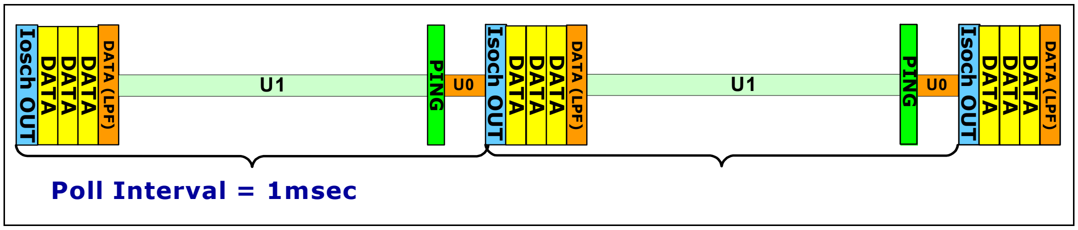


The device may choose to put the link directly into U2 instead of U1 if the service interval is sufficiently large.

5.6.3 Devices with Isochronous Endpoints

An isochronous endpoint is idle when all transfers for a given service interval have been completed, as indicated by the Last Packet Flag. Depending on the service interval and the amount of data moved, the device may choose to put the link in U2 if there is time for sufficient U2 residency.

Figure 28. Link Power Management for Devices with Isochronous OUT Endpoint



To ensure that the service requirements are met, the host will send a PING packet ahead of the transfer to bring all the links between the host and the device out of the low power state.

5.7 Power Management Checklist for USB 3.0 Devices

	Description	Yes/No
1	Initiate U1 Entry	
2	Initiate U2 Entry	
3	Support 1-ms latency if no LTM support	
4	If latency <1 ms, support LTM messages	
5	When LTM used, measure platform power when device active	
6	No or minimal impact to platform power when device is idle	

§

6 Conclusion

The world is moving toward 'Green technologies' and consumer demand for 'Extended Battery Life' is always increasing. The need for higher performance and new usage models will also keep increasing as they have done in the past couple of decades. Energy Efficient performance will be crucial for the computing industry in the future. Achieving this requires cooperation from all the components in the platform ecosystem. A framework that provides a mechanism for devices and applications to provide their service latency requirements based on workload to the platform power management controllers, allows for aggressive power management without sacrificing performance or reliability.

§

7 References

1) "Designing Power-Friendly Devices"

   http://download.intel.com/technology/EEP/designing_power_friendly_devices.pdf

2) "Making USB a More Energy-Efficient Interconnect"

   http://download.intel.com/technology/itj/2008/v12i1/2-usb/2-Making_USB_a_More_Energy_Efficient_Interconnect.pdf

3) "Latency Tolerance Reporting ECN", PCI Express 2.0 ECN, August 14, 2008

   http://www.pcisig.com/specifications/pciexpress/specifications

4) "Optimized Buffer Flush/Fill ECN", PCI Express 2.0 ECN, April 30, 2009

   http://www.pcisig.com/specifications/pciexpress/specifications

5) "USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification as of July 16, 2007"

   http://www.usb.org/developers/docs/

6) "Universal Serial Bus Revision 3.0 Specification"

   http://www.usb.org/developers/docs/

7) "Appendix C – Power Management" section of the USB 3.0 Specification

§