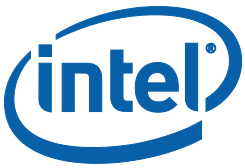


Energy-Efficient Platforms – Considerations for Platform hardware

Whitepaper

December 2011

Revision 1.01

Document Number: 322304-002

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
	1.01	• Updated formatting for help file conversion • Removed Section 1.2	December 2011
	1.0	• Initial Release	November 2011

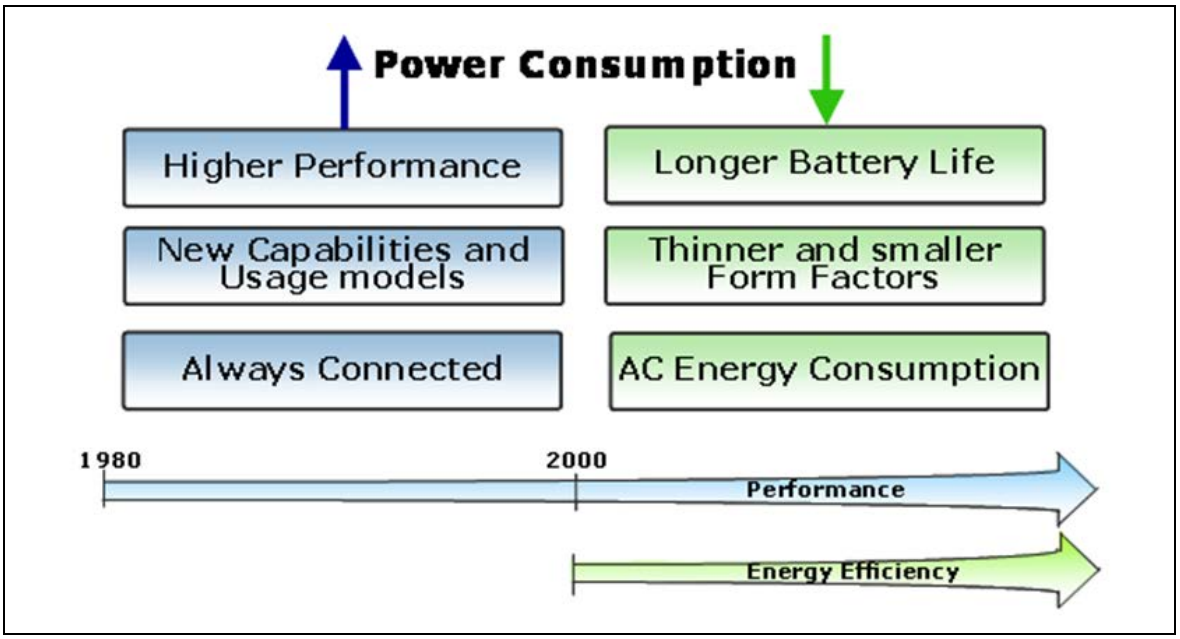
1 Introduction

1.1 Overview

Ever since the personal computing (PC) platforms were launched in the 1980s, industry focus has been on meeting consumer 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.

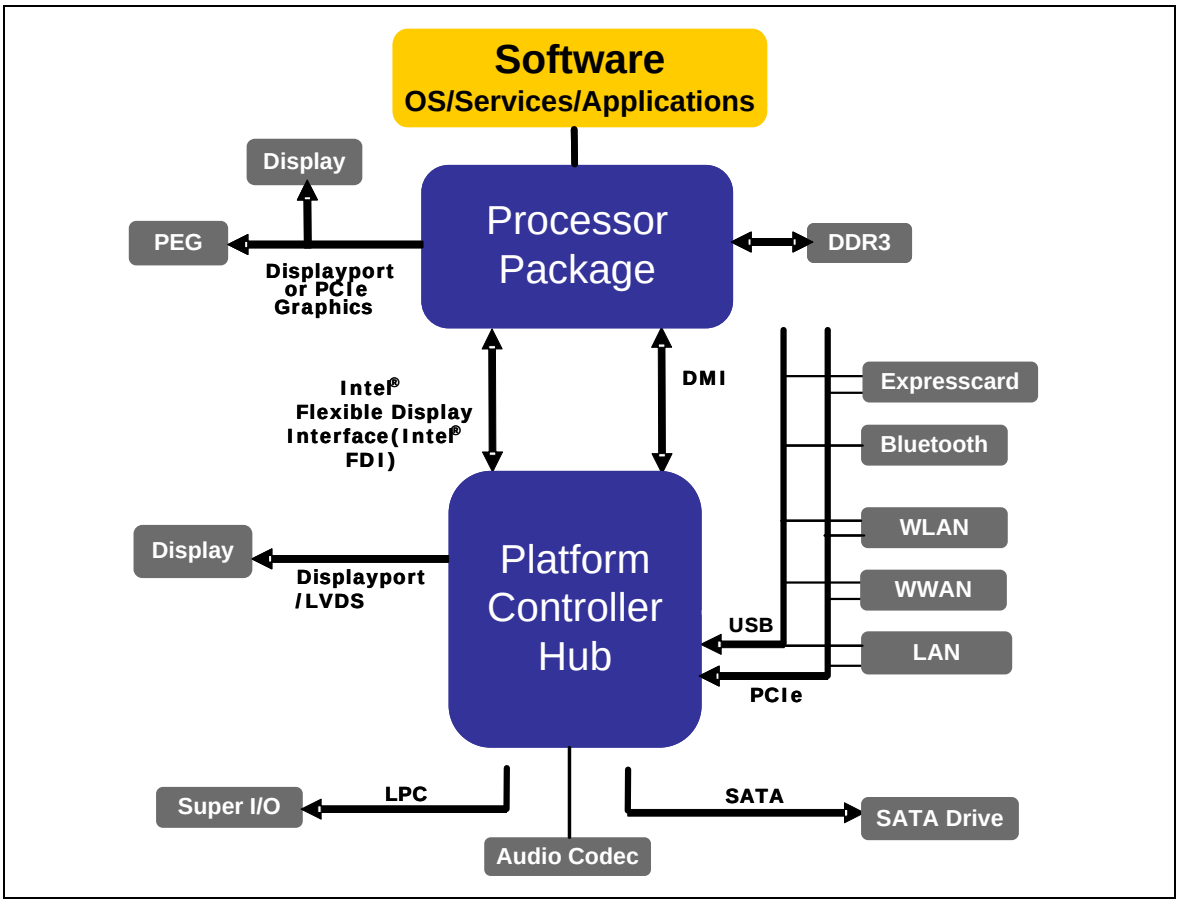
Starting around 2000, adoption of Mobile platforms of all form factors – Notebooks, Netbooks, Mobile Internet Devices, etc. have been steadily increasing and longer battery life is consistently ranked as one of the top requirements by consumers. Consumers of mobile platforms demand the same performance as desktops but also view battery life and small form factors as very important usability factors. Continuous network connectivity enhances the mobile usage model, but increases power consumption, thereby requiring improved energy efficiency. Mobile platforms must also meet the energy efficiency regulatory requirements such as the US Environmental Protection Agency (EPA) Energy Star program. To make the vision of "All day Battery Life" a reality, the average platform power consumption must go down.

Figure 1: Importance of Performance and Energy-Efficiency is growing



Intel® Architecture (IA) platforms such as depicted in Figure 2 are open systems where operating systems, software applications and services and hardware devices are created and sold by various 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 software ingredient can impede all these benefits by preventing the platform components from residing in low power states. Platform level energy efficiency requires all components in the platform ecosystem to cooperate.

Both innovative performance and power management features are being added to successive generations of Intel® Architecture (IA) platforms to provide performance on demand and save power at other times. Peripheral devices need to be designed with energy efficient power management features as per the industry specifications of the respective interconnects.

1.2 Mobile Platform Power

What is Mobile Platform Power? Power is typically the amount of energy consumed by the platform over time. This involves understanding the workload of the platform and the energy used over a given time for that workload.

Mobile Platform Power is typically broken into three categories:

- Thermal Design Power (TDP)
- Platform Average Power – Average platform power measured over some time when a workload is executing
- Platform Idle Power – Average platform power measured over some time when no workload is executing and the system is idle

The TDP power is defined as being the hardest workload the mobile platform should ever see under normal operating conditions, and is what the mobile platform thermal cooling system is designed to handle. TDP largely defines the power level that the system should be designed to cool, and is in general, not directly related to normal platform battery life.

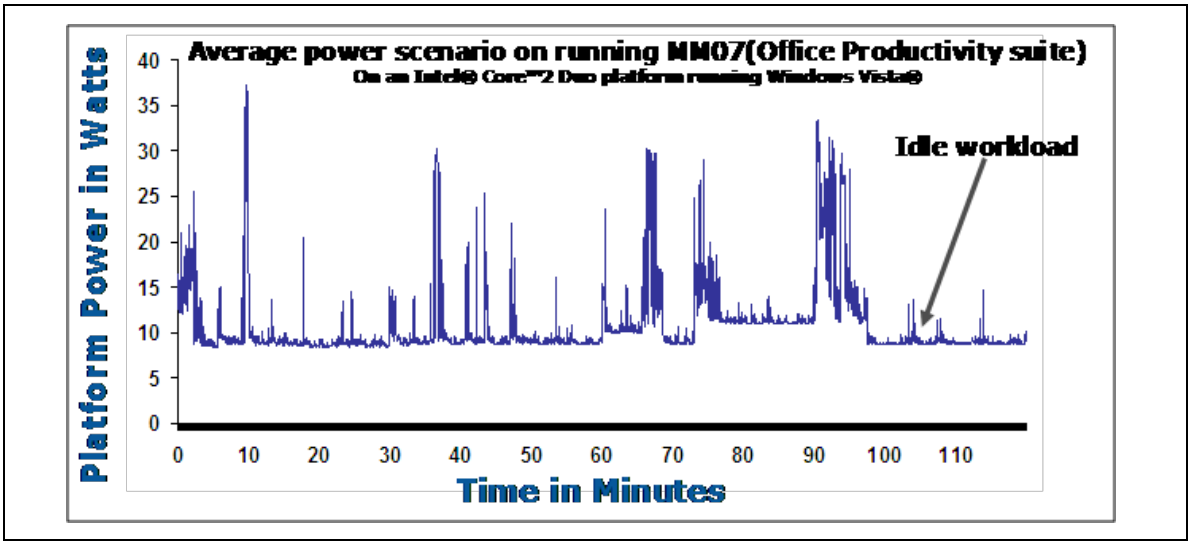
Average power for mobile platforms is defined as the average power consumption of the platform and is modeled by benchmarks such as Mobile Mark ’07 (MM07) that are representative of real end user usage patterns where the machine is idle between bursts of activity.

Idle power for mobile platforms is defined as being the power a platform would consume when the system is running in the ACPI S0 state, and software applications and services may be running but are not actively executing workloads and there is minimal background activity.

1.3 Typical Power Profile

Usage analysis has shown that a mobile platform in the ACPI S0 working state is typically idle for about 90-95% of the time as measured by the CPU C-state residency. The platform in this idle state still consumes about 7-10W of power due to large portions of system resources being kept powered up for best performance. Figure 3 below shows an example power graph over a period of time with a typical benchmark running.

Figure 3: Typical Mobile Platform Power Profile in ACPI S0 State



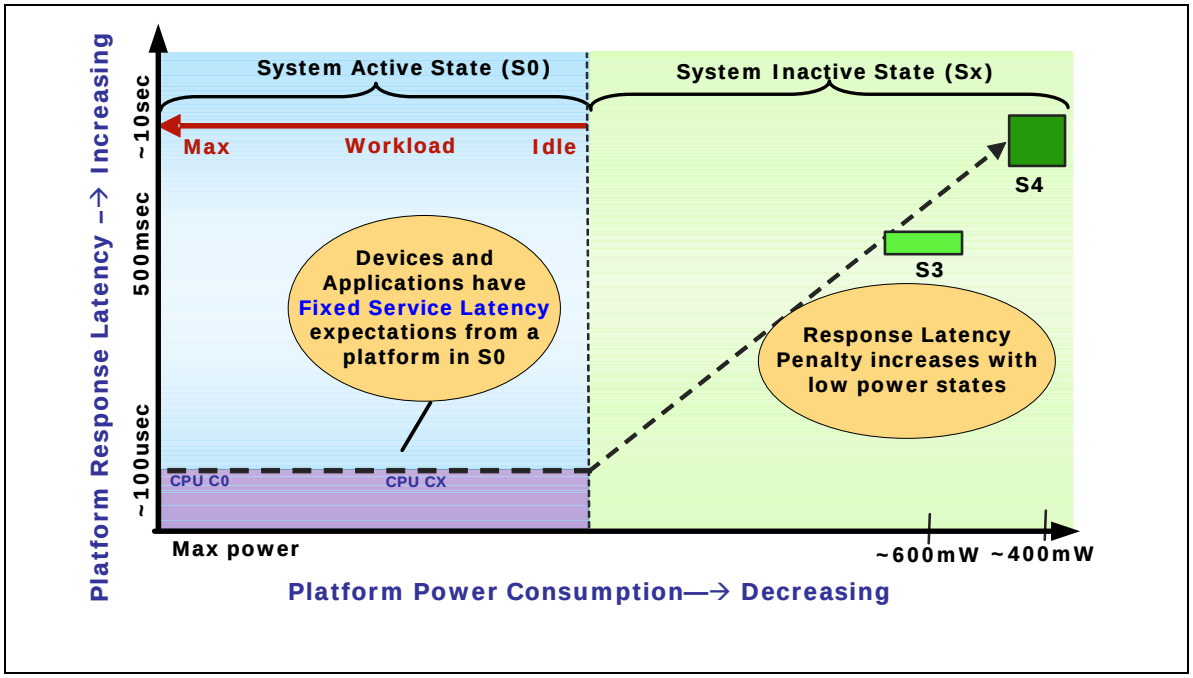
There can be significant platform power savings with a scalable architecture. Keeping system resources powered on when doing useful work and powered down when idle gives a good balance between performance and energy consumption. The vision is to increase energy efficiency by dynamically powering off 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 (~7-10 W) and sleeping power (~400 mW).

Since a mobile platform predominantly resides in the idle state, it is crucial to lower the platform idle power consumption for a significant increase in battery life. This also benefits the average power scenarios, and helps all but the most demanding (TDP-like) workloads. When actively executing workloads, improving computational and data efficiency so that the job can get done quickly, thereby increasing idle state residency improves platform energy-efficiency.

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 below shows the behavior of platform with fixed service latency.

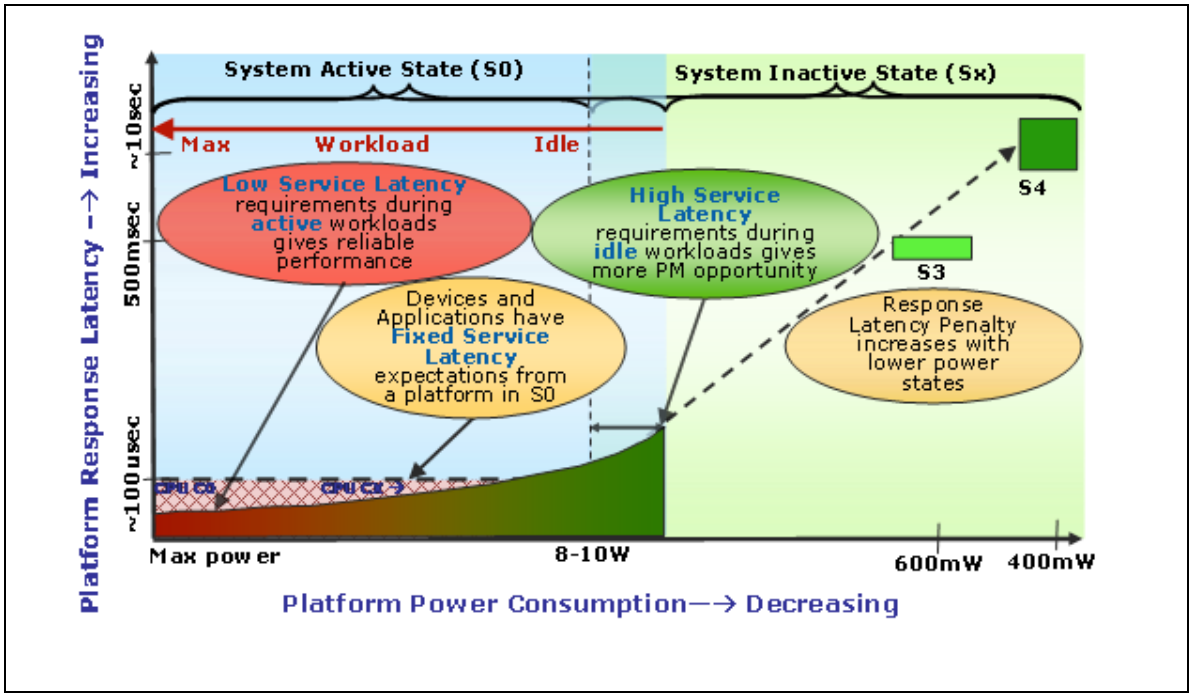
Figure 4: Fixed Service Latency expectations on today's platforms



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 us for mobile, < 5 us for desktop and < 1 us 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 on workload



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

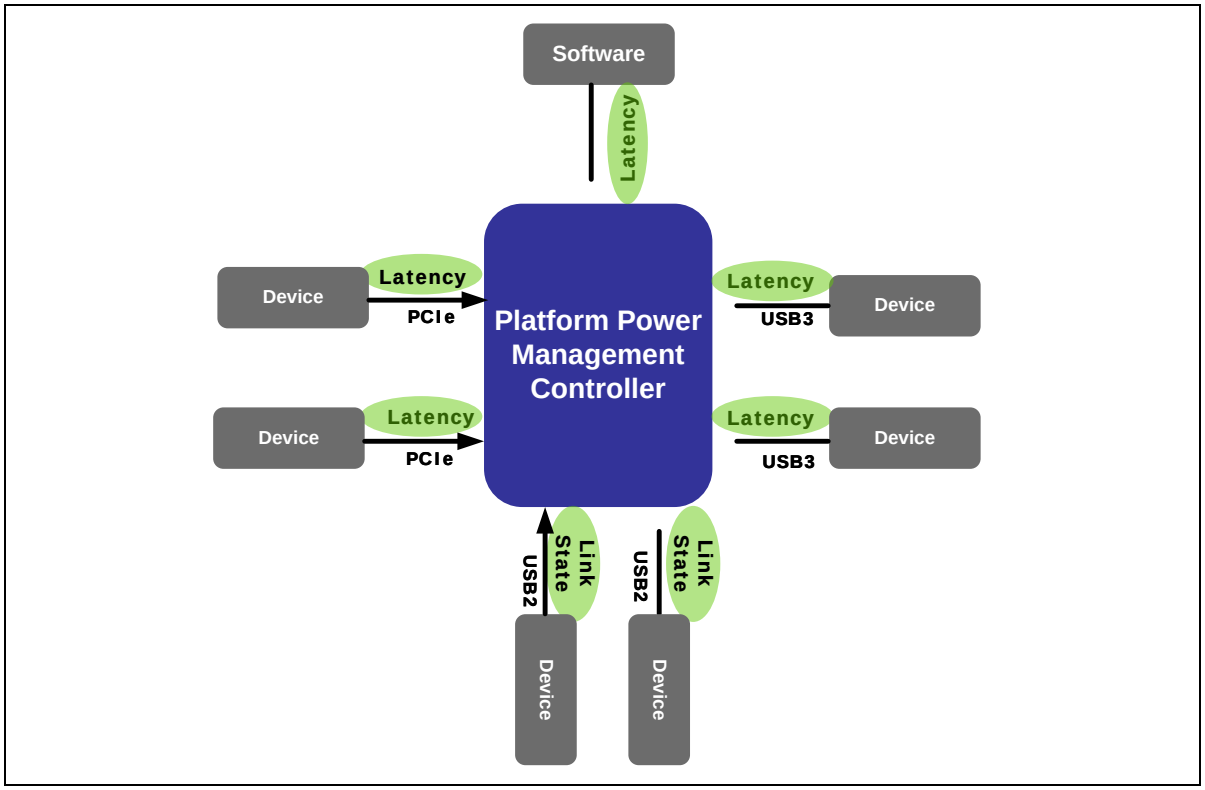
As shown in the figure 5, this enables the platform to lower its power consumption much more than what is possible in today's platforms during idle workloads without sacrificing performance.

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 the 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.

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

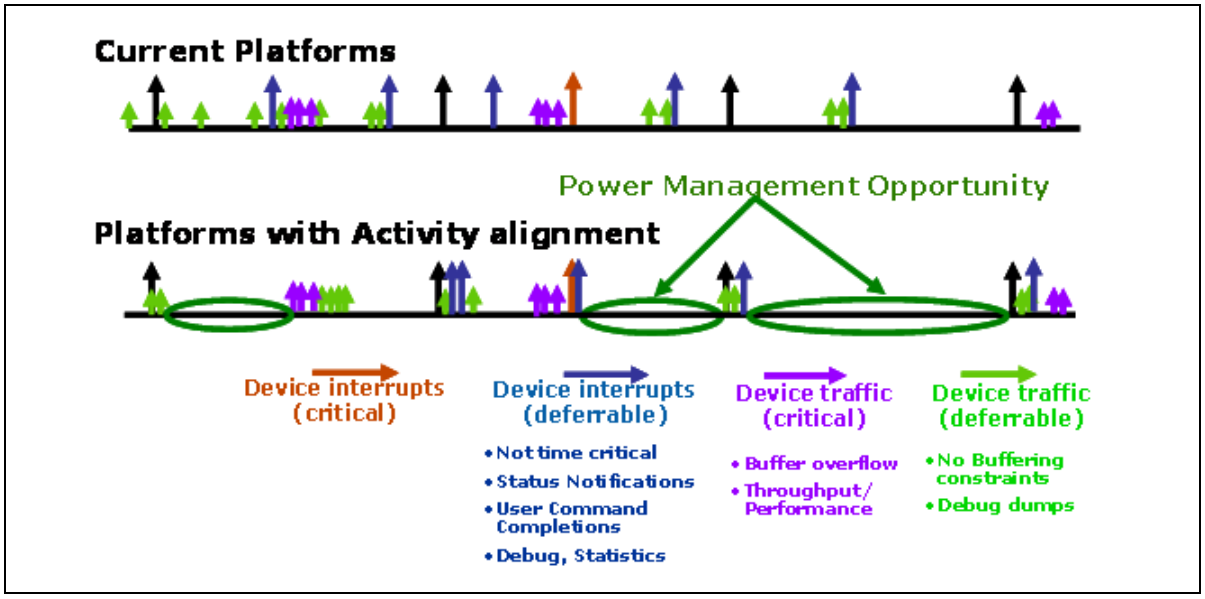
Two general methods have been defined for devices to convey their latency tolerance requirements:

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.

2.1.2 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

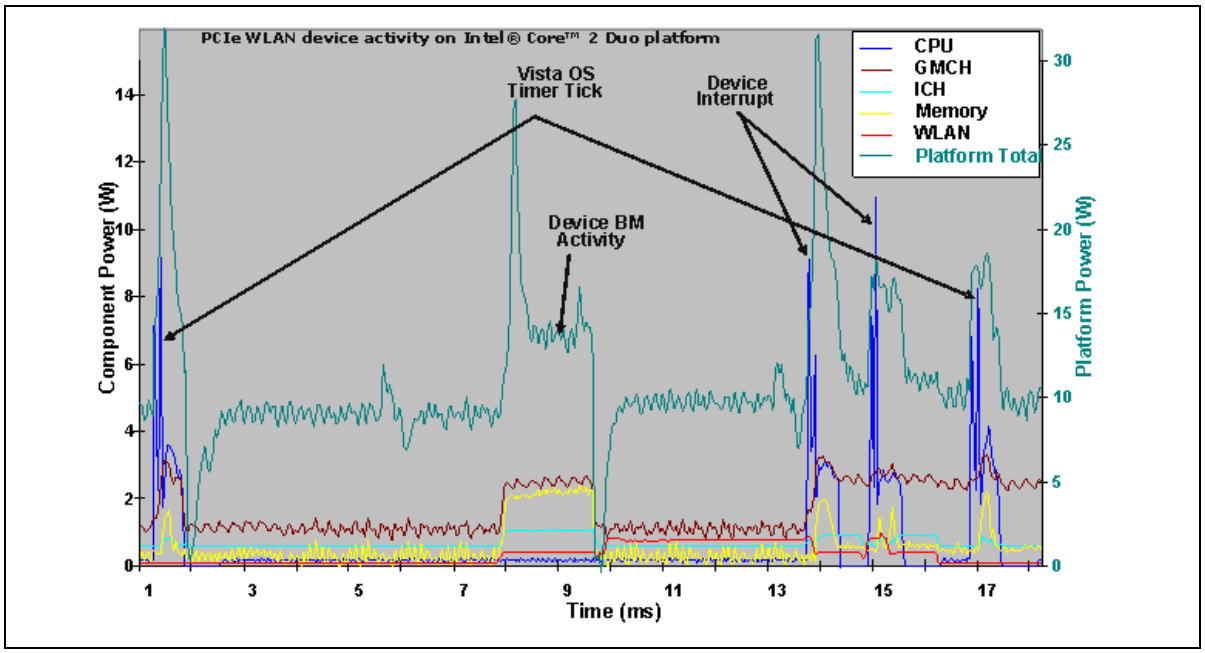


Information about optimal platform activity windows for device bus mastering and interrupt activity would be broadcast throughout the system providing an opportunity for devices to shape their traffic to these windows. The primary purpose of this technique is to present activity to the processor, memory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

As the 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 PM states during these idle periods.

2.1.3 Designing Devices for Platform Energy-Efficiency

Figure 8: Impact of Device Activity on Platfrom 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 300usec 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

PCI Express power management defines two major areas of support:

- PCI compatible power management. PCI Express power management is based upon hardware and software compatible with the PCI Bus power management interface specification, revision 1.1 and ACPI revision 2.0.
- Native PCI Express Extensions. These extensions define autonomous hardware-based Link Power Management, mechanisms for waking the system, a Message transaction to Power management Events (PME), and low power to active latency reporting and calculation.

PCI Express Link states are not visible directly to legacy bus driver software, but are derived from the power management state of the components residing on those Links. Defined Link states are L0, L0s, L1, L2, and L3. The power savings increase as the Link state transitions from L0 through L3.

Components may wakeup the system using a wakeup mechanism followed by a PME Message. PCI Express systems may provide the optional auxiliary power supply needed for wakeup operation from states where the main power supplies are off.

Another distinction of the PME mechanism is its separation of the following two tasks:

- Reactivation (wakeup) of the associated resources (i.e., re-establishing reference clocks and main power rails to the PCI Express components)
- Sending a PME Message to the Root Complex

Active State Power Management (ASPM) is an autonomous hardware-based, active state mechanism that enables power savings even when the connected components are in the D0 state. After a period of idle Link time, an ASPM Physical-Layer protocol places the idle Link into a lower power state. Once in the lower-power state, transitions to the fully operative L0 state are triggered by traffic appearing on either side of the Link. ASPM may be disabled by software.

3.1 PCIe Active State Link Power Management (ASPM)

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 7usec. 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.

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.

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

- L0 – Active state.L0 support is required for both ASPM and PCI-PM compatible power management. All PCI Express transactions and other operations are enabled.
- L0s – A low resume latency, energy saving "standby" state. L0s support is required for ASPM. It is not applicable to PCI-PM compatible power management. All main power supplies, component reference clocks, and components' internal PLLs must be active at all times during L0s.
- L1 – Higher latency, lower power "standby" state. L1 support is required for PCI-PM compatible power management. L1 is optional for ASPM unless specifically required by a particular form factor. All main power supplies must remain active during L1. All platform-provided component reference clocks must remain active during L1, except as permitted by Clock Power Management (using CLKREQ#) when enabled. A component's internal PLLs may be shut off during L1, enabling greater power savings at a cost of increased exit latency. The L1 state is entered whenever all Functions of a Downstream component on a given Link are programmed to a D-state other than D0. The L1 state also is entered if the Downstream component requests L1 entry (ASPM) and receives positive acknowledgement for the request.
- L2/L3 Ready – Staging point for L2 or L3. L2/L3 Ready transition protocol support is required. L2/L3 Ready is a pseudo-state that a given Link enters when preparing for the removal of power and clocks from the Downstream component or from both attached components. This process is initiated after PM software transitions a device into a D3 state, and subsequently calls power management software to initiate the removal of power and clocks.
- L2 – Auxiliary-powered Link, deep-energy-saving state. L2 support is optional, and dependent upon the presence of Vaux. A component may only consume Vaux power if enabled to do so. In L2, the component's main power supply inputs and reference clock inputs are shut off. When in L2, any Link reactivation wakeup logic (Beacon or WAKE#), PME context, and any other "keep alive" logic is powered by Vaux.
- L3 – Link Off state. When no power is present, the component is in the L3 state.

Figure 9: Link Power Management State Flow Diagram

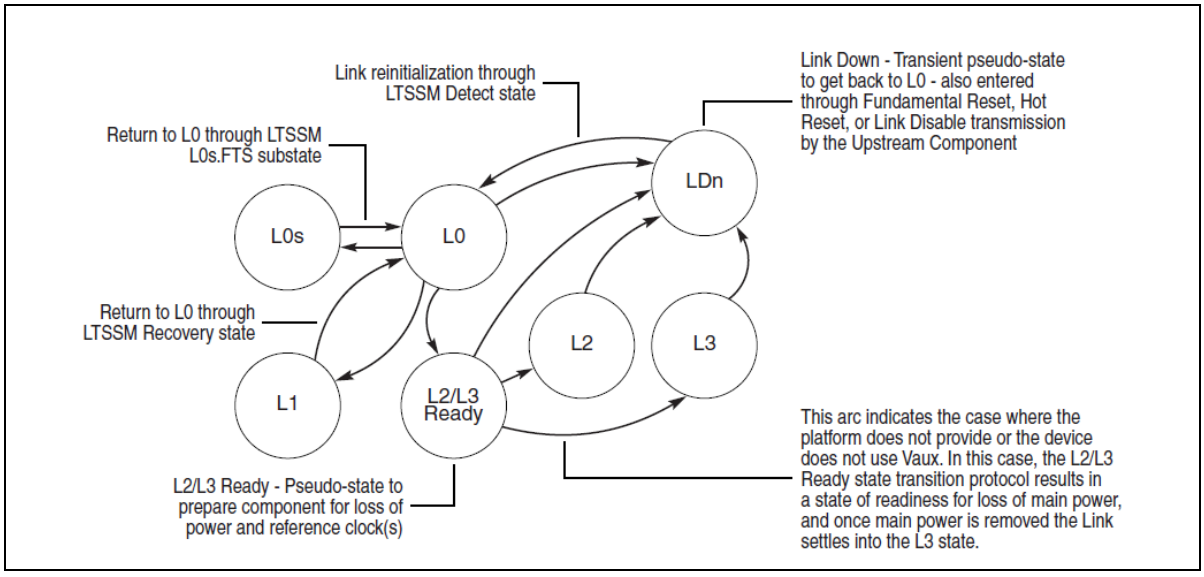


Table 1: Summary of PCIe Link Power Management States

	L-State Description	Used by S/W Directed PM	Used by ASPM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux
L0	Fully active Link	Yes (D0)	Yes (D0)	On	On	On	On/Off
L0s	Standby state	No	Yes [1] (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
L2/L3 Ready (pseudo-state)	Staging point for power removal	Yes [4]	No	On/Off [7]	On	On/Off	On/Off
L2	Low power sleep state (all clocks, main power off)	Yes [5]	No	Off	Off	Off	On [6]
L3	Off (zero power)	n/a	n/a	Off	Off	Off	Off
LDn (pseudo-state)	Transitional state preceding L0	Yes	N/A	On	On	On/Off	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.
3. L1 exit latency will be greatest for components that internally shut off their PLLs during this state.
4. L2/L3 Ready entry sequence is initiated at the completion of the PME_Turn_Off/PME_TO_Ack protocol 5 handshake . It is not directly affiliated with either a D-State transition or a transition in accordance with ASPM policies and procedures.
5. Depending upon the platform implementation, the system's sleep state may use the L2 state, transition to fully off (L3), or it may leave Links in the L2/L3 Ready state. L2/L3 Ready state transition protocol is initiated by the Downstream component following reception and TLP acknowledgement of the PME_Turn_Off TLP Message. While platform support for an L2 sleep state configuration is optional (depending on the availability of Vaux), component protocol support for transitioning the Link to the L2/L3 Ready state is required.
6. L2 is distinguished from the L3 state only by the presence of Vaux. After the completion of the L2/L3 Ready state transition protocol and before main power has been removed, the Link has indicated its readiness for main power removal.
7. Low-power mobile or handheld devices may reduce power by clock gating the reference clock(s) via the "clock request" (CLKREQ#) mechanism. As a result, components targeting these devices should be tolerant.

3.1.1 Recommendation 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.

3.2 PCI Express 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.

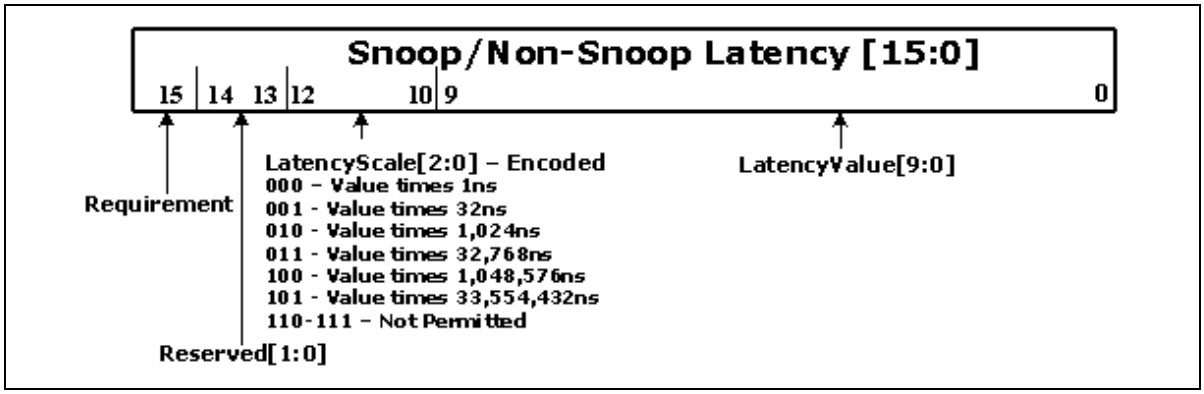
The LTR message is used by devices to convey their service latency requirements to upstream entities (switches, root complex, etc.). The latency values specified entail a budget for the entire path between the device and main memory – excluding link exit latencies and any device induced latencies that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

3.2.1 LTR Guidelines for Client Platforms

Figure 10: LTR Latency Field



- 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 500usec 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* 5µs latency tolerance at all times to ensure correctness, where a minimum of 100µs is strongly recommended.

Table 2: LTR Recommendations

Device Phase	Latency Tolerance	Notes
Low-Latency (Active)	5 to 20µs	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 5µs latency tolerance is required by all devices at all times, but higher values (e.g. 20µs) are obviously preferred. Devices which consistently operate at stringent values will significantly impact (increase) platform power consumption.

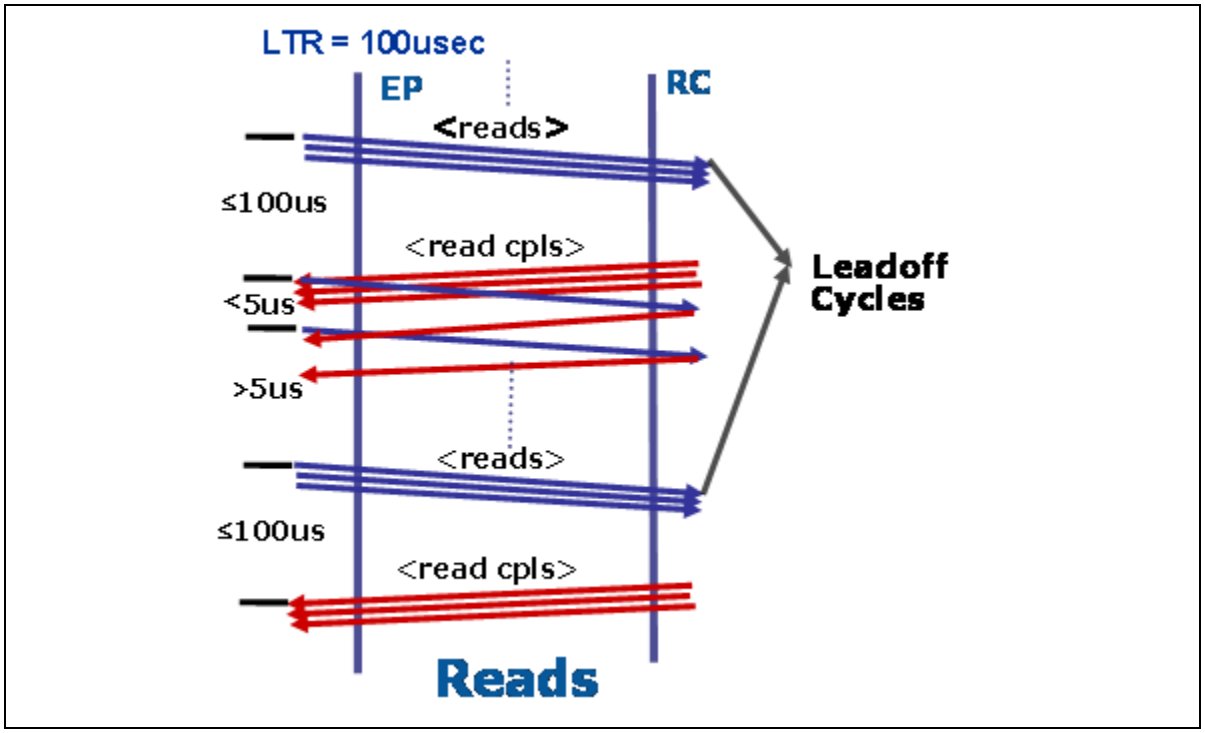
Normal (Active, Light Idle)	100µs	The recommended latency tolerance for most devices under active to pseudo-idle workloads. Allows the platform to employ relatively deep PM states.
Pervasively Idle (Idle)	Max Platform Latency Value (<1msec)	The recommended latency tolerance when a device is pervasively idle, generally characterized as the absence of meaningful activity for many milliseconds to seconds. Facilitates the use of the deepest platform idle PM states.

3.2.2 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 (<5usec). For transactions that are not leadoff cycles, no power management delays will be introduced by the platform.

3.2.2.1 Endpoint initiated Memory Reads and Completions

Figure 11: Endpoint initiated Memory Reads

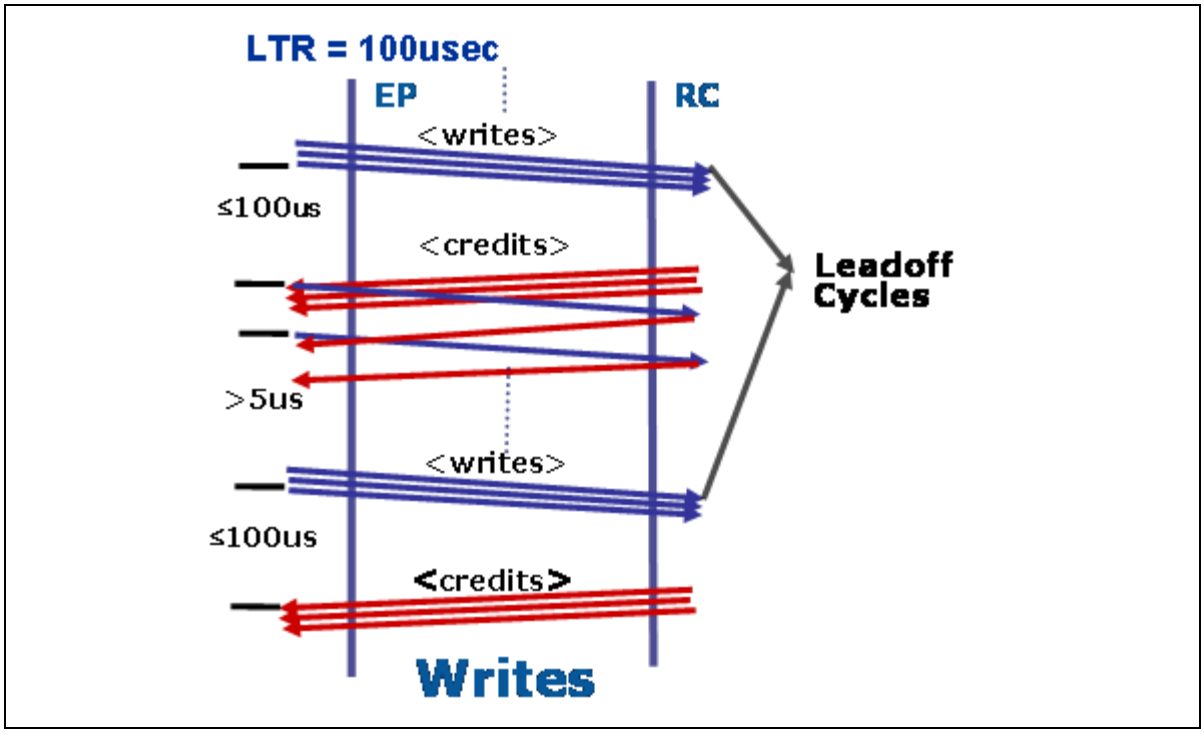


If an endpoint initiates a memory read transaction that is a leadoff cycle, there might be a delay in platform response with the delay not exceeding the latency value sent by the device in the last LTR message. If subsequent requests are pipelined, these transactions would also see the delays incurred 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.

3.2.2.2 Endpoint initiated Memory Writes and Flow Control

Figure 12: Endpoint initiated Memory Writes

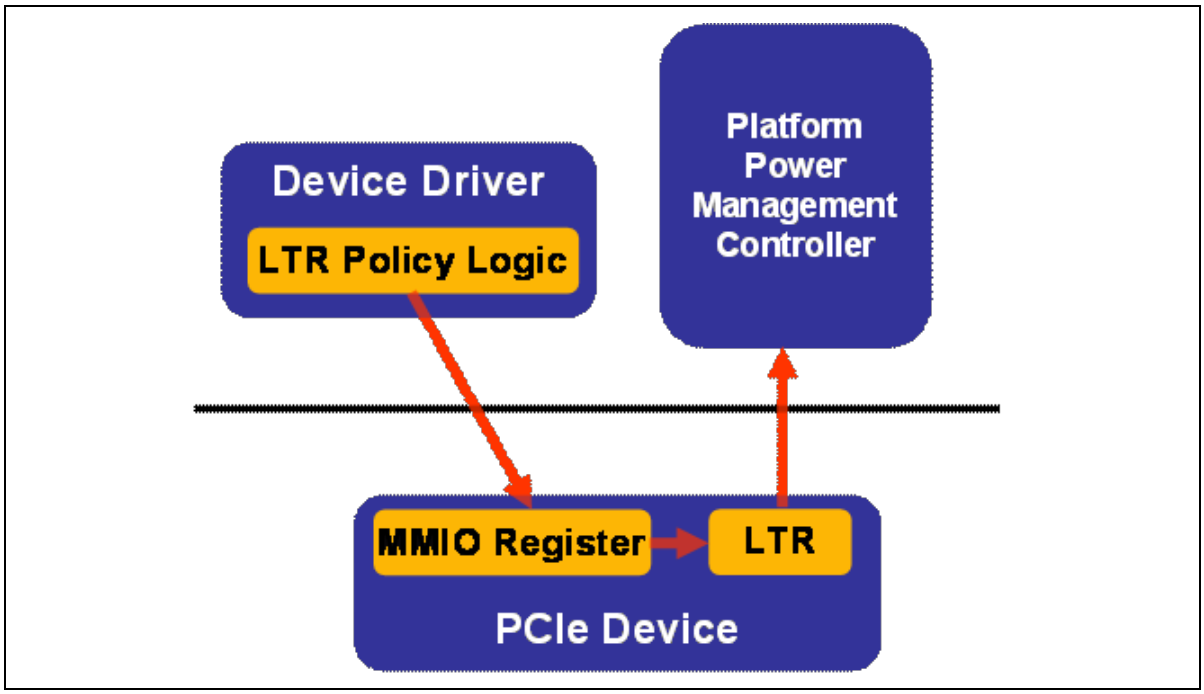


If an endpoint initiates a memory write transaction that is a leadoff cycle, there might be a delay in platform response with the delay not exceeding the latency value sent by the device in the last LTR message. For subsequent endpoint initiated transactions to avoid any power management latency, the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

3.2.3 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 the figure below may be preferable.

Figure 13: Software guided LTR message



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.

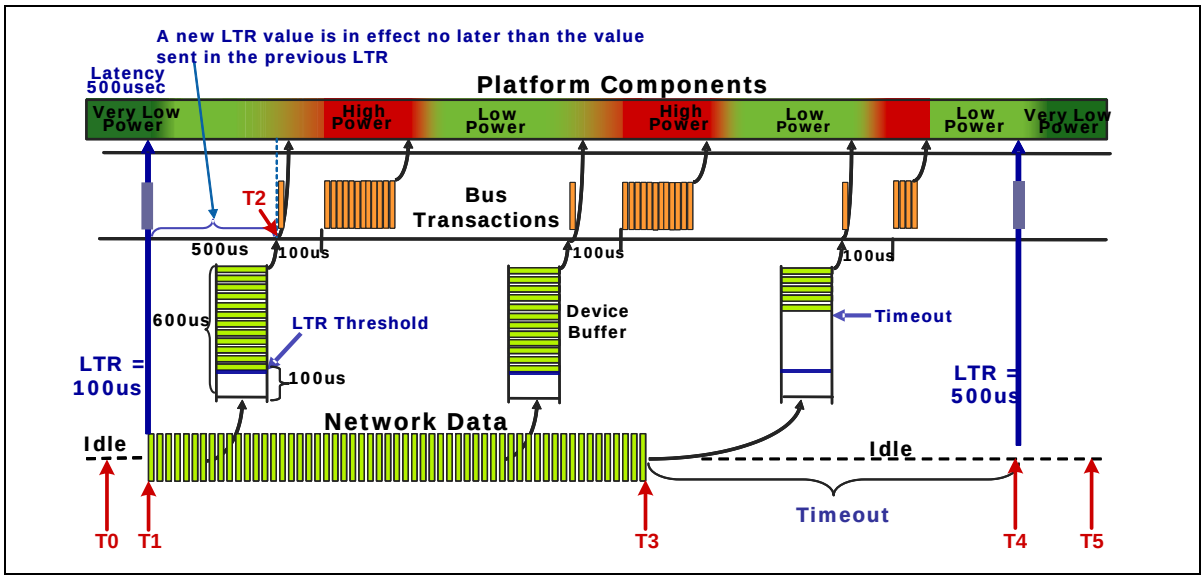
Static and slow dynamic devices may choose to implement the LTR policy logic in software. The device hardware LTR logic will send LTR messages upstream based on the guidance given by software. One way to implement this would be to have a memory-mapped IO (MMIO) register in the device. A write to this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

3.2.4 LTR Usage Examples

3.2.4.1 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 14: Ethernet adapter in ACPI D0 state sending LTR message



The figure above 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 100usec. 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 100usec 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 500usec which allows the platform to go back into a very low power state.

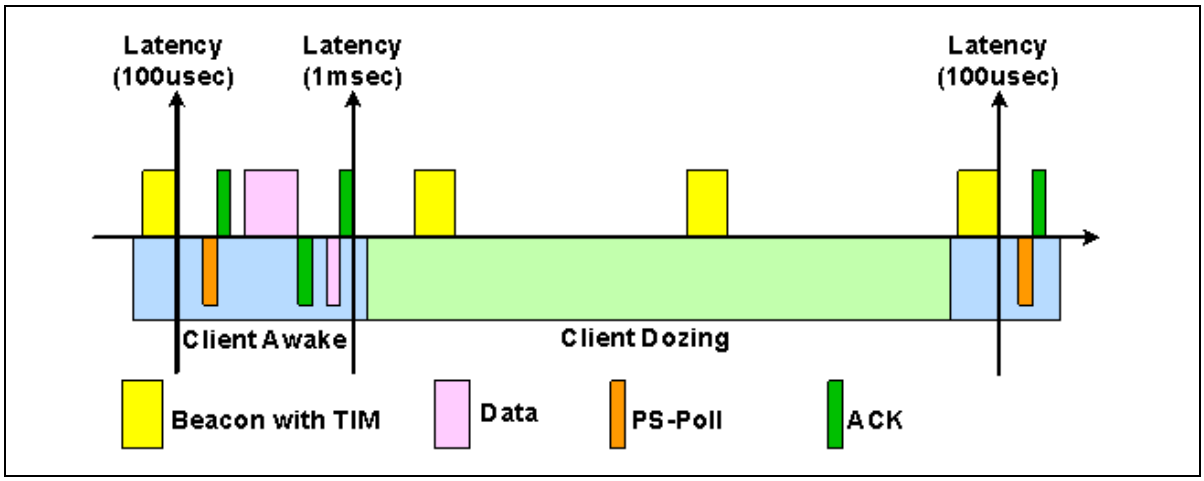
In the above example, the Ethernet LAN adapter uses its buffering as one of the metrics to give latency guidance. The latency values will also depend on data rate – 10Mbps, 100Mbps or 1Gbps. Higher latency values can be sent at lower data rates. The device/device driver may also take into account the 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.

3.2.4.2 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 the figure below, an LTR message of 100usec is sent when coming out of low power state and an LTR message of 1msec is sent when going into low power state.

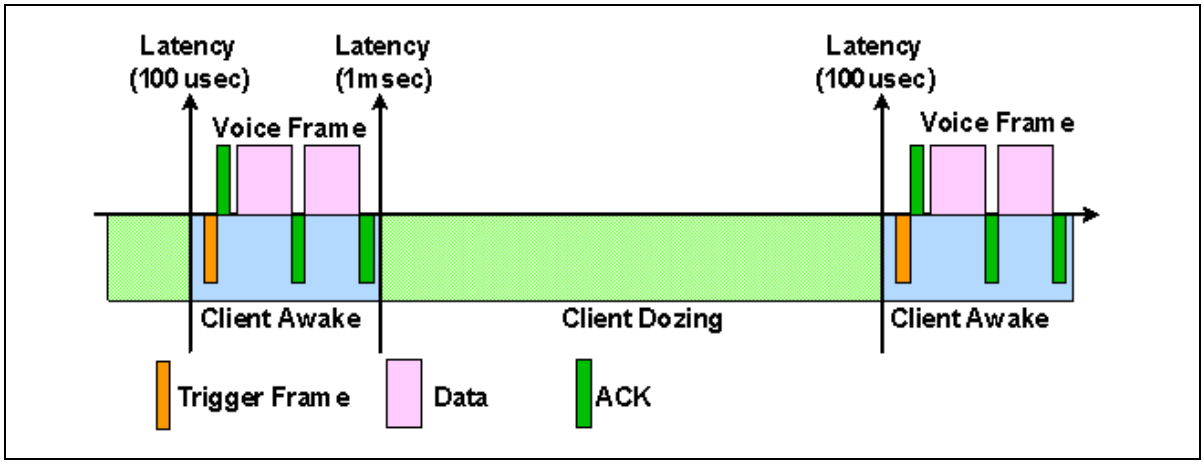
Figure 15: LTR messagefrom WLAN device using Wi-Fi Legacy Power Save



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

Figure 16: LTR messages from WLAN device using WMM power save



3.3 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.

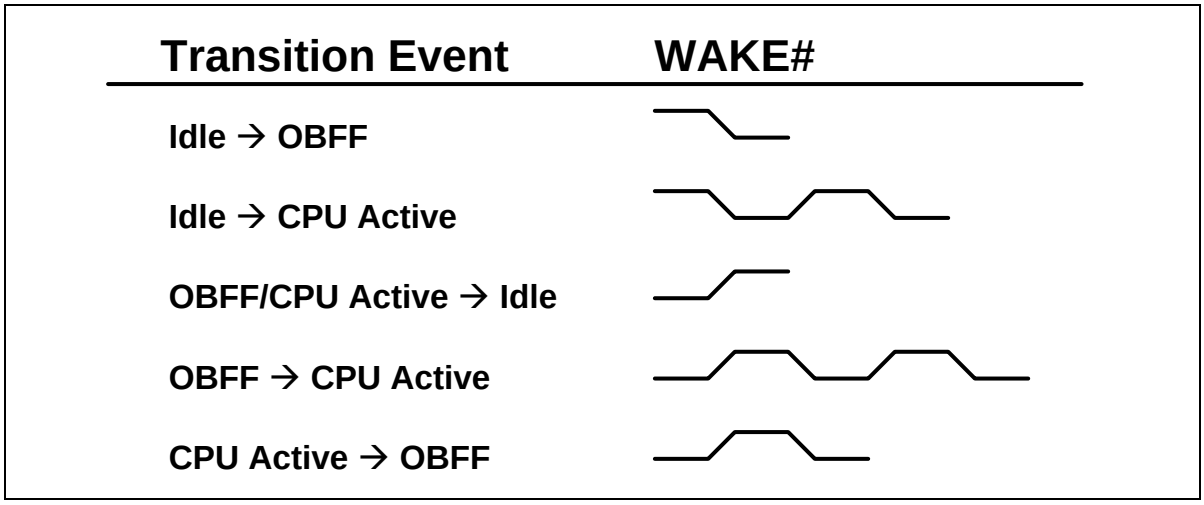
An OBFF indication is a hint – devices are still permitted to initiate bus mastering or interrupt traffic outside the optimal windows. But this will not be ideal for platform power and should be avoided as much as possible. The OBFF events are signaled using the WAKE# signal as this prevents needless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

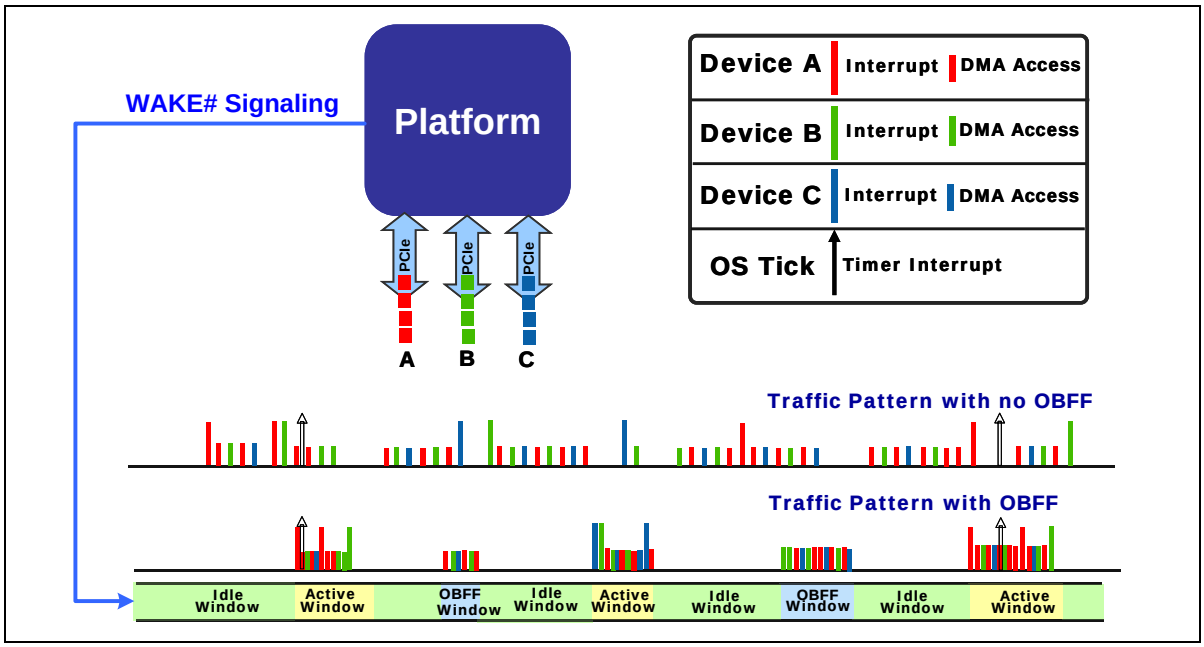
Figure below shows the WAKE# signaling for the OBFF transition events

Figure 17: WAKE# pin 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 18: Example of PCIe OBFF



4 SATA Link Power Management

4.1 Overview

SATA Link Power Management (LPM) puts the physical layer (PHY) of the link into a low-power state. This PHY layer Link Power Management is independent of the ATA protocol power state of the disk, and as such complements the existing power management capabilities provided by the ATA command set. For example, the ATA command set reduces the power consumption of the attached device by issuing ATA protocol-level power state change requests to the disk. These requests typically instruct the device to spin down the media to save power. The rotational state of the media is completely independent of the state of the link.

Independent intelligent PHY power management has shown a significant reduction in the overall power consumption of the SATA subsystem, both in the platform and in the SATA device itself.

4.1.1 Link Power Management States

SATA provides two link power management states, in addition to the “Active” state. These states are “Partial” and “Slumber,” that, by specification, differ only by the command sent on the bus to enter the low power state, and the return latency. Partial has a maximum return latency of 10 microseconds, while Slumber has a maximum return latency of 10 milliseconds. Typical host and device hardware implementations should be able to realize greater power savings in Slumber with its longer specified return latency. Partial, therefore, is designed to allow link power state transitions continually with minimal impact on performance. Slumber is designed to be used only when the link is expected to be idle for an extended period of time. It is not possible to transition the link directly from Partial to Slumber without passing through the Active state.

4.1.2 Host- and device- Initiated Power Management

SATA Link Power Management can be broken up into two basic types: Host-Initiated Link Power Management (HIPM) and Device-Initiated Link Power Management (DIPM). SATA Link Power Management requires cooperation between the host and the device. Either the host or the device can request the link to enter a low-power state, but the corresponding host or device must accept or reject the link state change request. Each of these provides power savings by themselves; maximum power savings, however, are achieved when both are implemented together.

Host-initiated power management can be implemented either in the host hardware or the host software. In the first case, the host controller requests a link power management transition immediately after all outstanding commands to the SATA device have been completed. This allows the link to enter a low-power state immediately upon completion of the commands to the

disk. Since the host has the best knowledge of what commands have been posted, or will be posted to the device, the host is able to make an immediate link power state change without invoking a time-out period.

The host controller on the host can automatically put the link into either Slumber or Partial after the command completes. Typically, for performance reasons, this will be Partial. However, after some period of idleness, it is generally assumed that the link will be inactive for an extended period of time and it is desirable to transition the link from Partial to Slumber. This can be done either by the host software or the device.

Device-initiated power management is implemented by the SATA device. The SATA device knows best how long a specific command might take to complete, and is best equipped to request a link power management state change while processing the command.

Since the host is best equipped to manage the PHY between commands and the device best within a command, the best power management is obtained when the host and device cooperate.

4.1.3 Link Power Management and Device State

The operating system can put devices into a D-state as defined by the Advanced Configuration and Power Interface (ACPI) specification. The link, however, can be put into its power management state independent of the D-state of the host controller or the device.

4.2 Power Management Protocol

The protocols used for entering and exiting Partial and Slumber states are different. To enter one of the low-power states, the standard communication protocol between the host and device can be used. Once the PHY has been placed into a low-power state, the standard communication protocol between the host and SATA device cannot be used; and another Out of Band (OOB) mechanism is required.

4.2.1 Entry Signaling Protocol

Media application programs often change the timer resolution to avoid glitches during Either the host or the device can initiate a request to enter into the Partial or Slumber power state. The request is signaled by transmitting either the PMREQ_P or PMREQ_S primitive. PMREQ_P is used for a request to transition to the Partial power state, and PMREQ_S is used to transition to the Slumber power state. The corresponding host or device must then respond with a PMACK acknowledge or PMNAK negative acknowledge primitive. If the request is acknowledged with a PMACK, both the host and device transition into the corresponding power state. To ensure the PMACK received is reliable without having to handshake the handshake, the target sends several PMACK's before entering a low-power state. If a PMNAK is received, no power state change occurs.

4.2.2 Exit Signaling Protocol

Once the PHY is in a low-power state, SATA specifies a low-level signaling mechanism to bring the interface back to active state. An OOB signal “COMWAKE” is sent to the device that acts as a wake-up call and causes communication to be reestablished.

4.2.3 Hardware/Software Protocol

Link Power Management is only enabled when the host controller and the device report that they are capable of issuing or receiving Link Power Management requests, and the OS software driver is capable of enabling the host hardware and SATA device. The host controller reports this capability to software in the Capabilities Register of the AHCI host controller. The SATA device reports the ability to support these commands in the IDENTIFY_DEVICE data structure returned by the device during device enumeration. The host must specifically enable DIPM on the SATA device via the ATA SET_FEATURES command upon initialization to enable DIPM on the SATA device. See the ATA and SATA specifications for details.

4.2.4 Listen Mode

The AHCI specification allows host software to put a port into listen mode when no device is connected to the port. A port in Listen mode has the equivalent power consumption on the host as a port in slumber, but allows the port to detect device insertions on that particular port.

4.2.5 Automatic Slumber to Partial (APS)

Automatic partial to slumber allows the host and the SATA device to do an automatic or implicit transition to slumber from partial without requiring the link to transition to active. An implicit transition means that the host or SATA device can take up to the 10 ms specified for slumber to return from the partial state. No explicit PMREQ_S commands are sent on the bus to transition it to slumber and no transition to the active state is required. As with DIPM, the SATA device sets a bit in the IDENTIFY DEVICE data structure to indicate that the device is capable of transitioning the link on the device. The device sets another bit in the IDENTIFY DEVICE data structure indicates that the SATA device allows the host to auto transition the link.

As with DIPM the host must specifically enable APS on the device via the ATA SET_FEATURES command.

4.3 Host vs. Device Link Control

There are two fundamental times the link can be put into slumber. These are between commands, or between command bursts, and within a command or command burst on the bus.

Between commands, the host is generally best equipped to put the link into a low power state. The host and host software has a wide variety of information available to it that it can use to determine the timing frequency of the transition into the two low power states.

Within a command or command burst, the SATA device is best equipped to transition the link into a low power state. The SATA device has advanced knowledge of the time it will take to respond to any specific command or command burst.

4.4 Host/Device Design Recommendations and Interaction

Between commands, the host hardware should transition the link into Partial after every command or command burst. Partial provides the best tradeoff between power savings and performance.

In the absence of a host-initiated power management transition, the device should attempt to transition the link into Partial after some appropriate short timeout.

After a longer period of timeout, it can be assumed that the link will remain idle for a longer period of time. Either the host, through the host software, or the device should at this point transition the link to Slumber.

Within a command, the SATA device should attempt to put the link into partial or slumber based on the time the SATA device will take to respond to the command.

4.5 Device Removal during Power Management

SATA Link Power Management disables the PHY on the SATA TX/RX transmit and receive lines between the host and device. Because the PHY is disabled, a device removal cannot be immediately detected without waiting for the next command to be sent to the SATA device. The AHCI specification provides the capability for a second electrical signal to be provided to the host controller to notify the host that the SATA device has been removed. Typically this will be implemented as a Mechanical interlock switch on the mobile platform.

4.6 Recommended Host and Device Behavior

Future IA platforms will contain additional power savings features in the core subsystem that require the SATA links to be in the low power state. The following recommendations will help OS and device vendors to take advantage of these capabilities.

4.6.1 Recommended Host Behavior

Host Software should enable Host Initiated power management on the host with the following capabilities listed below.

If the device supports Device Initiated Link Power Management, the host should enable device Initiated link power management on the device.

4.6.1.1 Host Behavior between Commands

The host software should enable Aggressive Link Power Management on the host controller, and set the host controller to auto transition to partial. Or the host SW should have the link enter partial as quickly as possible within 1-2 micro seconds of the command completion.

Host software should attempt to put the link into slumber after 10ms of inactivity delay.

If the device supports APS the host should enable APS on the device, and do an implicit (no PM_REQ) transition of the link into slumber after a 10ms timeout.

APS support in the host is optional.

4.6.1.2 Host Behavior between Commands

The device is best equipped to manage the link within a command. The host software should not put the link into partial or slumber within a command.

4.6.2 Recommended Device Behavior

Because not all currently shipping OS's and OS policies support HIPM slumber transitions, the device should support Device Initiated Power Management slumber transitions in the device. Device support of APS is optional.

4.6.2.1 Between Commands

The device firmware should put the link into partial very quickly (ideally in a few micro seconds after the command completion). This delay allows the host to put the link into partial if HIPM is enabled in the system.

The device should also transition the link into slumber after 10ms of inactivity.

In devices that support APS, and in systems where the host has enabled APS on the device, the device should do an implicit (no PM_REQ) transition to slumber after 10 ms.

Devices should not NAK host link requests at any time

4.6.2.2 Device Behavior within a Command

Within a command the device should put the link into partial or slumber immediately after the command is received, potentially taking into account the expected response time of the device and the expected response time of the system.

Within a command, the policy device uses to put the link into partial or slumber is determined by the device implementation. A simple implementation that puts the link into partial immediately after receiving a command is probably adequate for most device implementations. The suggestions below are designed to assist a device vendor in implementing a best of class power and performance implementation.

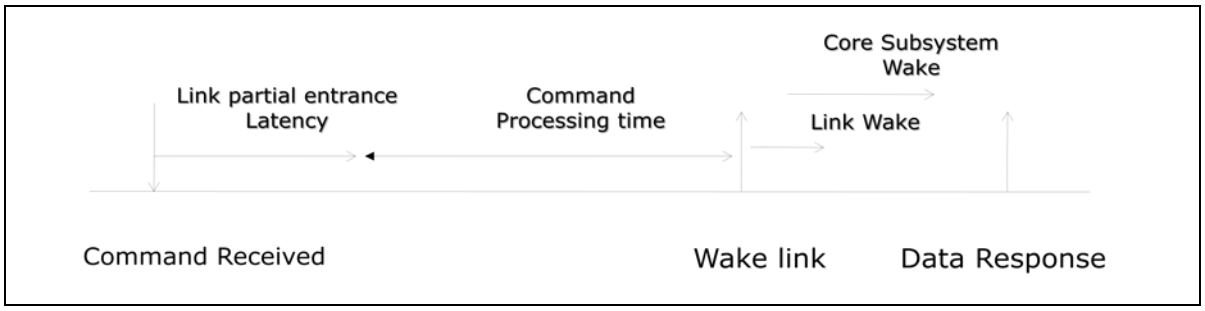
Max resume latency of the core system and the link combined is expected to be 1 millisecond for slumber, and 100 µ second for partial.

Partial is recommended, slumber is optional and for many devices the power savings may not be worth the implementation cost of slumber within a command.

Devices that support APS and slumber should support an explicit slumber transition using PM_REQ on the bus, rather than an implicit (no PMREQ) transition on the bus.

The following diagram shows a typical device response to a command. The diagram shows 3 major events: 1) when the command is initially received by the device, 2) when the device should begin to wake up the link and 3) when the device attempts respond to the host with data.

Figure 19: Typical Command Response by Device



After the command is received, the link takes some time (typically 1 microsecond) to enter into a low power state. Then there is some time the device requires processing the command. There is some time to wake up the link, and then there is some time that the core subsystem may take above and beyond the link wake up time before data can be transferred from the device to the host.

The device should enter a low power state within a command when the link entrance latency plus the combined link wake latency and core subsystem wake latency are less than the time the device will take to respond. The device may attempt to wake the link some time before the data is ready to transfer to ensure the host system is ready to receive the data when the device is ready to send it.

4.6.3 Summary of Recommended Host and Device Behavior

Table 3: Link Recommendations between Commands (No APS)

No APS	Host Initiated	Device Initiated	Comments
Slumber Timeout	10 ms	10 ms	Explicit Slumber transition using PMREQ_S

Partial Timeout	<1 µsec (ALPM) 1-2 µsec (SW initiated)	As soon as possible (within µsecs)	5 µsec allows host to transition to partial first

Table 4: Link Recommendations between Commands (APS enabled)

APS	Host Initiated	Device Initiated	Comments
Slumber Timeout	Implicit – 10 ms	Implicit – 10 ms recommended device time out	Implicit Slumber transition
Partial Timeout	<1 µsec (ALPM)	As soon as possible (within µsecs)	

Table 5: Link Recommendations within Commands (w/wo APS)

	Host Initiated	Device Initiated	Comments
Slumber Timeout	None	Optional. Use and duration of slumber calculated by device assuming 1 millisecond max system resume latency. Devices that enable APS should use explicit slumber transition.	Power savings probably not worth the implementation effort
Partial Timeout	None	Calculated by device assuming a maximum 100 µsec system resume latency	Partial timeout is recommended

4.7 Debugging SATA LPM Issues

4.7.1 SATA Link States

On an idle system (one with very little activity and very little SATA traffic) run a capture and look at the link states. You should run this test on a machine with the OS and drivers installed, but very few applications loaded on the system, and very little traffic to the disk. In this configuration all ports should be in slumber most of the time.

Hard drives used as the boot drive should show a > 90% slumber residency when the machine is idle and the disk not accessed. Hard drives not used as a boot drive should show 100% slumber.

Optical drives on OS's that poll the optical drive should show ~90% slumber on that port when the drive is unused. Optical drives on systems that support AN should show ~100% slumber when the drive is unused.

If the slumber state residency for the port is less than these numbers, then the following recommendations will help you achieve the maximum power savings.

4.7.2 Port Recommendations

ESATA ports cannot have SATA LPM enabled on them. A port that is not physically connected to an ESATA port cannot be marked ESATA.

Hot Plug Capable Ports cannot have SATA LPM enabled on them unless there is an interlock switch configured to the port as described in this document. Only Ports connected to drives that are physically removable should have the hot plug capable bit set. Ports connected to drives that are not physically removable should NOT have the bit set.

If the port is physically removable, and interlock switch should be configured in the system as described in this paper.

4.7.3 Device Recommendations

Make sure the device supports both HIPM and DIPM.

4.7.4 DIPM not Enabled on the Device

Make sure the device has DIPM enabled on it.

Intel® RST in its default configuration will enable DIPM when the drive supports it and when the bits in the Port Recommendations above are configured to allow LPM.

Other OS's or drivers may have different policies. For instance at the time of the development of this paper Microsoft Windows 7* will support Link Power Management differently based on the OS Power Policy setting. In particular Microsoft Windows 7 will support only HIPM in the balanced and high performance power policies. It will support HIPM and DIPM in the power saver power policy.

4.7.5 Device Behavior

If the platform complies with the above recommendations, and the link is still not showing significant slumber state residencies, then it is possible that the device is not effective in

transitioning the link into a low power state, or is not allowing the host to put the link into a low power state. There are two things to check for in driver design

1. The device claims to support DIPM and does not initiate a slumber transaction, or does not do so quickly enough. If the device does not initiate a slumber transaction, or does so after a very long timeout, the link may not transition as efficiently into the slumber state as otherwise. We recommend that the drive transition the link into slumber after about 100 ms in the 2011-2012 timeframe and 10 ms in the 2013 timeframe.
2. The device NAKs requests from the host to go into either the partial or slumber state. We recommend that the drive not NAK any of the partial or slumber requests from the host.

Contact the device vendor for more information on the behavior of the device. If necessary, debug of these problems will require a SATA line analyzer.

5 USB2 Link Power Management

5.1 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 (3ms+OS overhead for entry and 30msec+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.

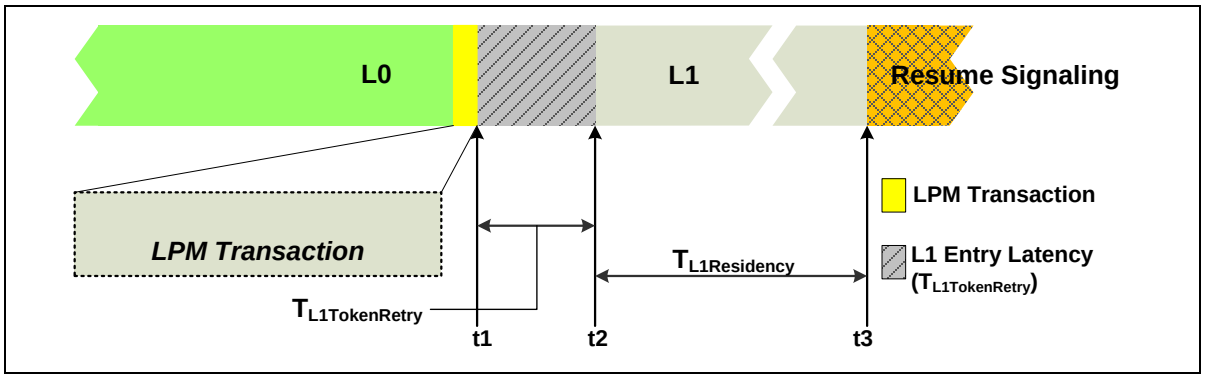
Supporting the LPM L1 state requires modifications to both USB host controllers and devices. It is backward compatible in that the new host can determine whether a device supports L1. L1 will only be used if the device acknowledges support for this feature.

Table 6: Comparisons of LPM L1 and LPM L2

	L1 (Sleep)	L2 (Suspend)
Entry	Explicitly entered via LPM extended transaction	Implicitly entered via 3ms 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; device-initiated resumes can be (optionally) enabled/disabled by software
Signaling	Low and Full- speed idle	Low and Full-speed idle
Latencies	*Entry*: ~10µs *Exit*: ~70µs to 1ms (host-specific)	*Entry*: ~3ms *Exit*: >30ms (OS-dependent)
Link Power Consumption	~0.6mW (data line- pull-ups)	~0.6mW (data line- pull-ups)
Device Power Consumption	Device power consumption level is application/implementation specific	Device consumption is limited to: ≤500 µA or ≤2.5mA
Hot Removal	Natively detected per USB2 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 20: 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 = 50µs, each increment adds 75µs (50µs, 125µs, 200µs…1.2ms).

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 7: USB2.0 latency Tolerance Support

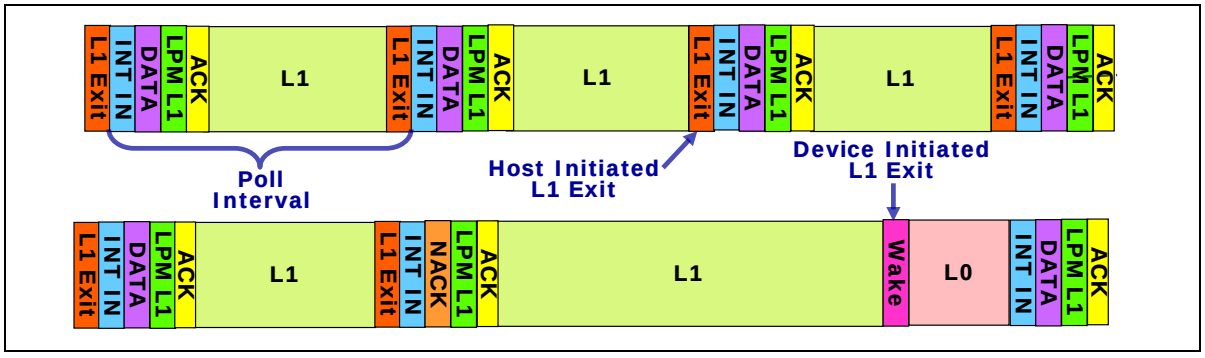
Type	Mechanism	Description
Implicit	L1 State	The latency indicated by the HIRD value can be tolerated by devices when their upstream link resides in the newly defined L1 state. Long latencies can also be tolerated for L2 state.
Explicit	–	There are currently no plans to support explicit latency tolerance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

5.2 LPM L1 Usage guidelines

5.2.1 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 21: LPM L1 usage for USB2.0 devices with Periodic Endpoints

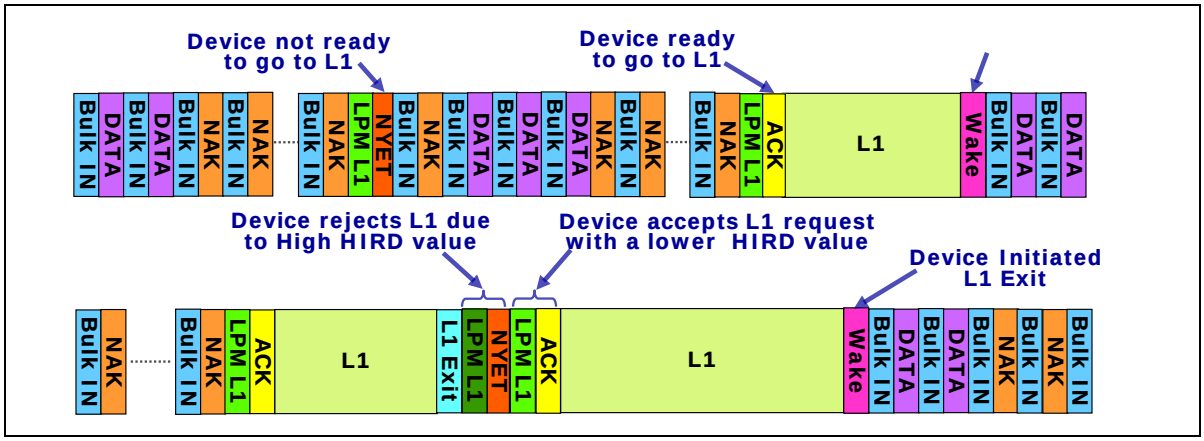


For Interrupt endpoints that are idle and do not have data to transfer, the host controller will put the link into L1 state after the NAK transaction and the endpoint will not be polled at the next poll interval. The link will continue to stay in the L1 state till the device initiates an L1 exit. This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

5.2.2 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 22: LPM L1 usage for 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.

5.3 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	

6 USB3 Link Power Management

6.1 Overview

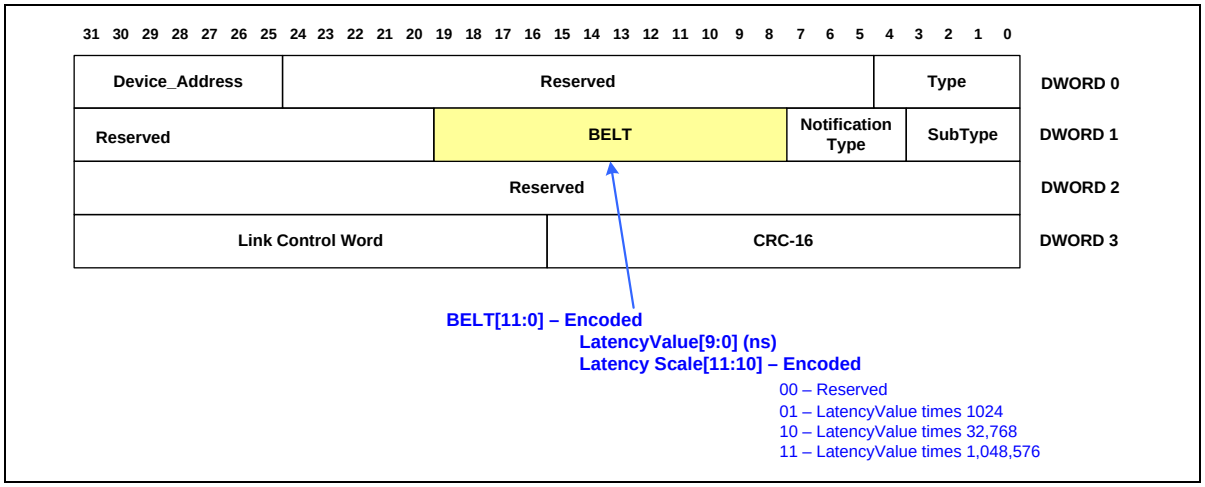
SATA Link Power Management (LPM) puts the physical layer (PHY) of the link into a low-power state. This PHY layer Link Power Management is independent of the ATA protocol power state of the disk, and as such complements the existing power management capabilities provided by the ATA command set. For example, the ATA command set reduces the power consumption of the attached device by issuing ATA protocol-level power state change requests to the disk. These requests typically instruct the device to spin down the media to save power. The rotational state of the media is completely independent of the state of the link.

Independent intelligent PHY power management has shown a significant reduction in the overall power consumption of the SATA subsystem, both in the platform and in the SATA device itself.

6.2 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 23: USB 3.0 Latency Tolerance Messaging



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.

6.3 LTM reporting guidelines for client platforms

Figure 24: 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		us

- All devices are assumed to support a BELT value of 1msec 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 125usec.
- 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 1msec 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.

6.4 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.

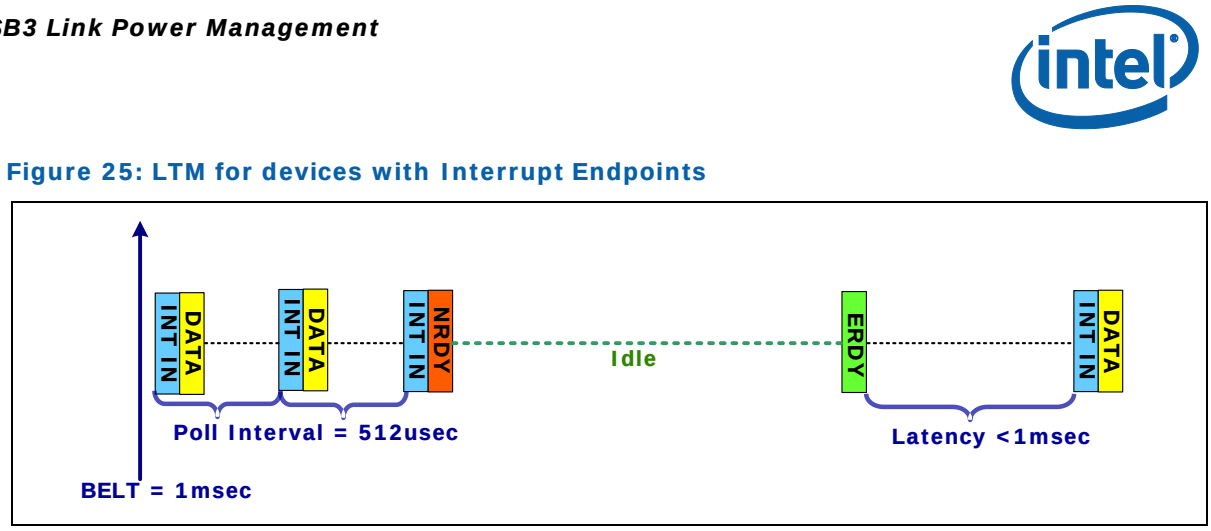


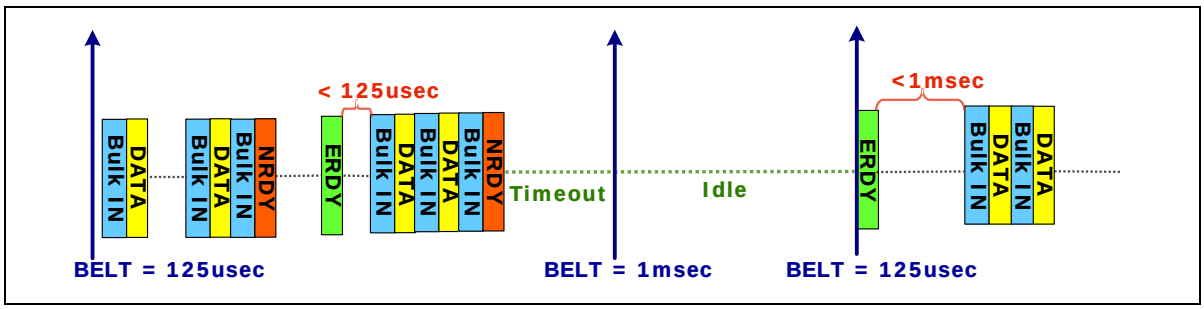
Figure 25: LTM for devices with Interrupt Endpoints

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 512usec. As long as there is a data transfer in every interval, the device will be polled every 512usec. 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 512usec, the next poll may not happen up to 1msec later as indicated by the BELT value.

6.5 LTM for Devices with Bulk Endpoints

Figure 26: LTM for devices with Bulk Endpoints



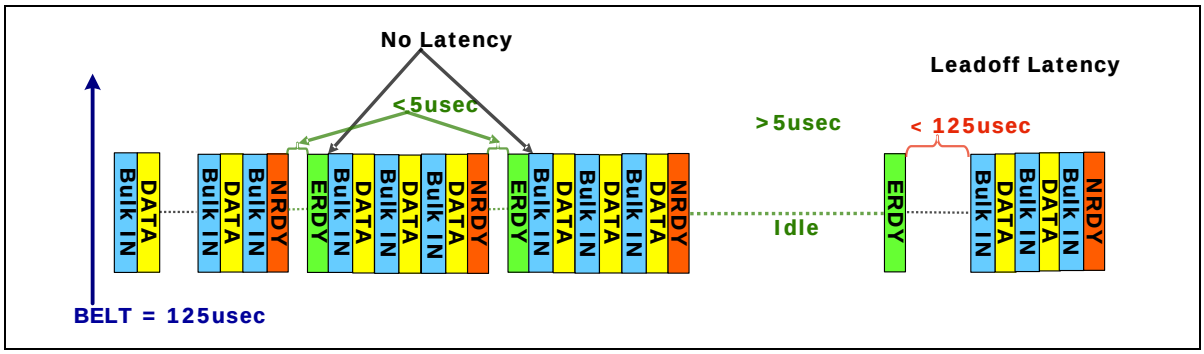
The figure above shows an example of a device with bulk endpoints using LTM messages. When the device moves to an active state, it sends an LTM message with a BELT value of 125usec. When there is no data to move, the endpoint sends an NRDY packet and the endpoint is placed in the flow control state. 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 125usec.

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 1msec 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 125usec. Since the previous BELT value was 1msec, the platform may take up to 1msec 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 5usec or longer, as shown in the figure below.

Figure 27: 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 ~300usec 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.

6.6 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 8: USB 3.0 Link Power Management States

Link State	Description
	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 usec 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 usec range.
U3 (Suspend)	U3 is a deep power saving state where portions of device (e.g., Physical Layer) power may be removed. V_{BUS} 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.

6.7 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

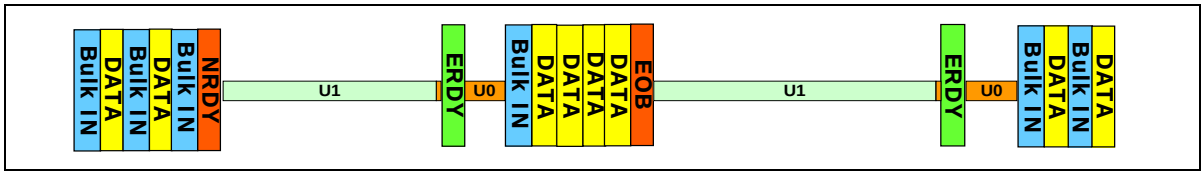
6.7.1 Devices with Bulk Endpoints

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 28: Link Power Management for Devices with Bulk IN Endpoint



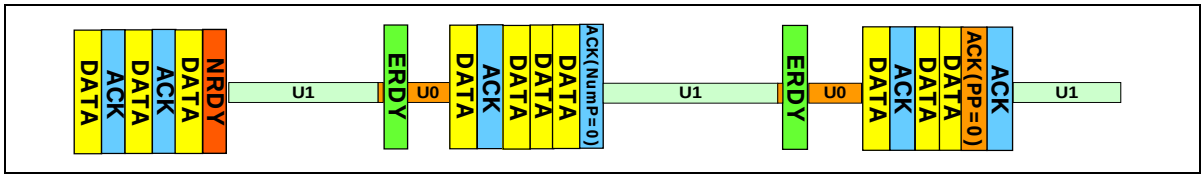
Typically, a device would put a link in U1 when it goes idle. When a device is aware of long periods of idleness as in a hard disk device spinning up a spindle, or a wireless device going into sleep mode, it may choose to put the link immediately in U2 instead of U1 for higher power savings.

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 29: 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.

6.7.2 Devices with Interrupt Endpoints

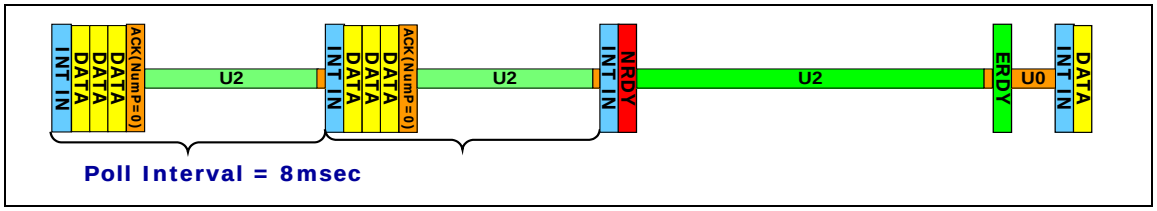
The interrupt transfer type is used for infrequent data transfers with a bounded service interval. Interrupt transactions are limited to a burst of three data packets in each service interval.

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 30: Link Power Management for Devices with Interrupt IN Endpoint

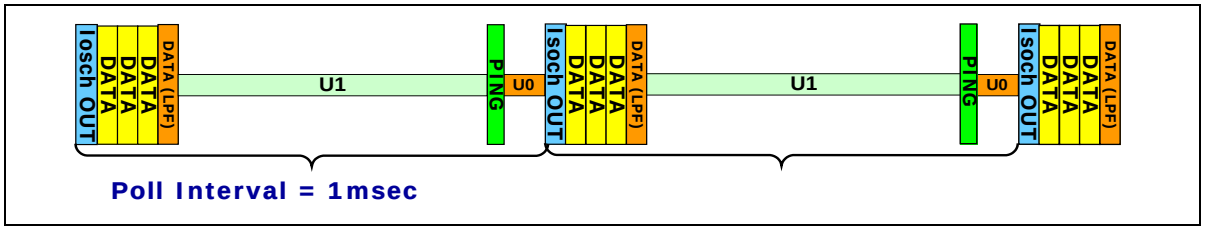


The device may choose to put the link directly into U2 instead of U1 if the service interval is sufficiently large.

6.7.3 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 31: 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.

6.8 Power Management Checklist for USB 3.0 devices

	Description	Yes / No
1	Initiate U1 Entry	
2	Initiate U2 Entry	
3	Support 1msec latency if no LTM support	
4	If latency <1msec, 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	

7 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-Efficiency will be crucial for the computing industry in the future both to increase battery life for mobile platforms and to reduce energy expenses for desktop and server platforms. Software behavior can have a significant effect on platform power consumption and battery life.

In typical usage models, the mobile platform is idle (as measured by CPU C0 residency) for about 90-95% of the time. It is important to reduce power consumption for idle and semi-idle workloads. Energy-efficient applications when idle should have a minimal impact on platform power consumption. Frequent background activity should be avoided.

During active workload execution, applications and services should improve computation efficiency, maximize multi-threaded execution and coalesce activity to increase idle residency. This will allow the platform to go into deeper low power states, reduce C-state transitions, and thereby amortizing the power cost of transitioning the platform into and out of low power states.

8 References

8.1 Tools

- Intel Battery Life Analyzer requests, questions and feedbacks
  mailto:BatteryLifeAnalyzer@intel.com
- Intel® VTune™ Amplifier XE
  http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/
- Microsoft Windows Performance Toolkit (included in Windows SDK)
  http://msdn.microsoft.com/en-us/windows/bb980924
- Sysinternals Process Monitor
  http://technet.microsoft.com/en-us/sysinternals

8.2 Documents

- "Energy-Efficient Platforms: Designing Devices Using the New Power Management Extensions for Interconnects"
  http://www.intel.com/technology/mobility/notebooks.htm
- "Maximizing Power Savings on Mobile Platforms"
  http://software.intel.com/en-us/articles/maximizing-power-savings-on-mobile-platforms/
- "Mobile Battery Life Solutions for Windows 7"
  http://download.microsoft.com/download/7/E/7/7E7662CF-CBEA-470B-A97E-CE7CE0D98DC2/mobile_bat_Win7.docx
- Selective Suspend in USB Drivers
  http://www.microsoft.com/whdc/driver/wdf/USB_select-susp.mspx
- "Designing Power-Friendly Devices"
  http://download.intel.com/technology/EEP/designing_power_friendly_devices.pdf
- "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
- "Latency Tolerance Reporting ECN", PCI Express 2.0 ECN, August 14, 2008
  http://www.pcisig.com/specifications/pciexpress/specifications
- "Optimized Buffer Flush/Fill ECN", PCI Express 2.0 ECN, April 30, 2009
  http://www.pcisig.com/specifications/pciexpress/specifications

- "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/

- "Universal Serial Bus Revision 3.0 Specification" http://www.usb.org/developers/docs/

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