PCI Express* 3.0 Technology: Device Architecture Optimizations on Intel Platforms

Mahesh Wagh
IO Architect

TCIS006

Sponsors of Tomorrow.™ intel

Agenda

- Next Generation PCI Express* (PCIe*) Protocol Extensions Summary
- Device Architecture Considerations
  - Energy Efficient Performance
  - Power Management
- Software Development
- Summary
- Call to action

PCI Express* (PCIe*) 2.1 Protocol Extensions Summary

Extensions	Explanation	Benefit
Transaction Layer Packet (TLP) Processing Hints	Request hints to enable optimized processing within host memory/cache hierarchy	Reduce access latency to system memory. Reduce System Interconnect & Memory Bandwidth & Associated Power Consumption Application Class: NIC, Storage, Accelerators/GP-GPU
Latency Tolerance Reporting	Mechanisms for platform to tune PM	Reducing Platform Power based on device service requirements, Application Class: All devices/Segments
Opportunistic Buffer Flush and Fill	Mechanisms for platform to tune PM and to align device activities	Reducing Platform Power based aligning device activity with platform PM events to further reduce platform power Application Class: All devices/Segments
Atomics	Atomic Read-Modify-Write mechanism	Reduced Synchronization overhead, software library algorithm and data structure re-use across core and accelerators/devices. Application Class: (Graphics, Accelerators/GP-GPU))
Resizable BAR	Mechanism to negotiate BAR size	System Resource optimizations - breakaway from “All or Nothing device address space allocation” Application Class: – Any Device with large local memory (Example: Graphics)
Multicast	Address Based Multicast	Significant gain in efficiency compared to multiple unicasts Application Class: (Embedded, Storage & Multiple Graphics adapters)

Continued…

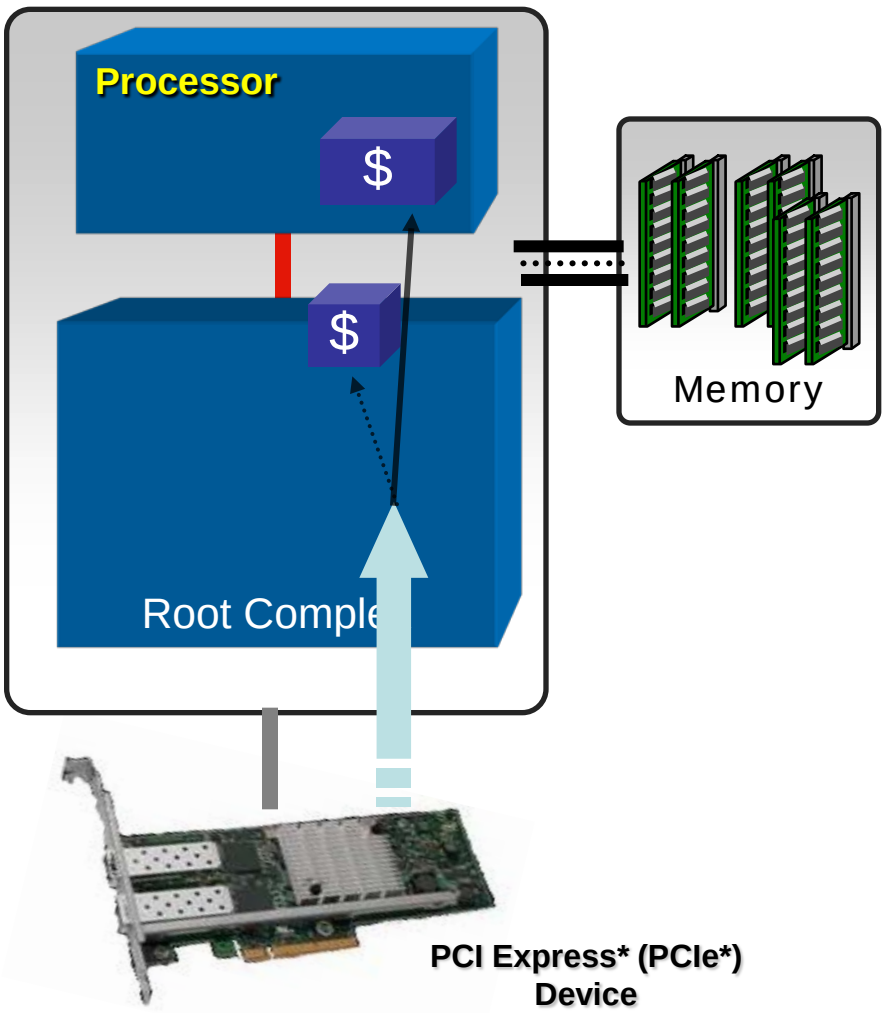
Extensions	Explanation	Benefit
I/O Page Faults	Extends IO address remapping for page faults – (Address Translation Services 1.1)	System Memory Management optimizations Application Class: Accelerators, GP-GPU usage models
Ordering Enhancements	New ordering semantic to improve performance	Improved performance (latency reduction) ~ (IDO) 20% read latency improvement by permitting unrelated reads to pass writes. Application Class: All Devices with two party communication
Dynamic Power Allocation (DPA)	Mechanisms to allow dynamic power/performance management of D0 (active) substates.	Dynamic component power/thermal control, manage endpoint function power usage to meet new customer or regulatory operation requirements Application Class: GP-GPU
Internal Error Reporting	Extend AER to report component internal errors (Correctable/ uncorrectable) and multiple error logs	Enables software to implement common and interoperable error handling services. Improved error containment and recovery. Application Class: RAS for Switches
TLP Prefix	Mechanism to extend TLP headers	Scalable Architecture headroom for TLP headers to grow with minimal impact to routing elements. Support Vendor Specific header formats. Application Class: MR-IOV, Large Topologies and provisioning for future use models

New Features with Broad Applicability

Device Architecture Considerations

- TLP Processing Hints (TPH)
- Power Management Extensions

TLP Processing Hints (TPH)



TLP Processing Hints (PCIe Base 2.1 specification)

- Memory Read, Memory Write and Atomic Operations

System Specific Steering Tags (ST)

- Identify targeted resource e.g. System Cache Location
- 256 unique Steering Tags

Benefits

Effective Use of System Resources

- Reduce Access latency to system memory
- Reduce Memory & system interconnect BW & Power

Improves System Efficiency

Effective use of System Fabric and Resources

Requirements Checklist

Ecosystem

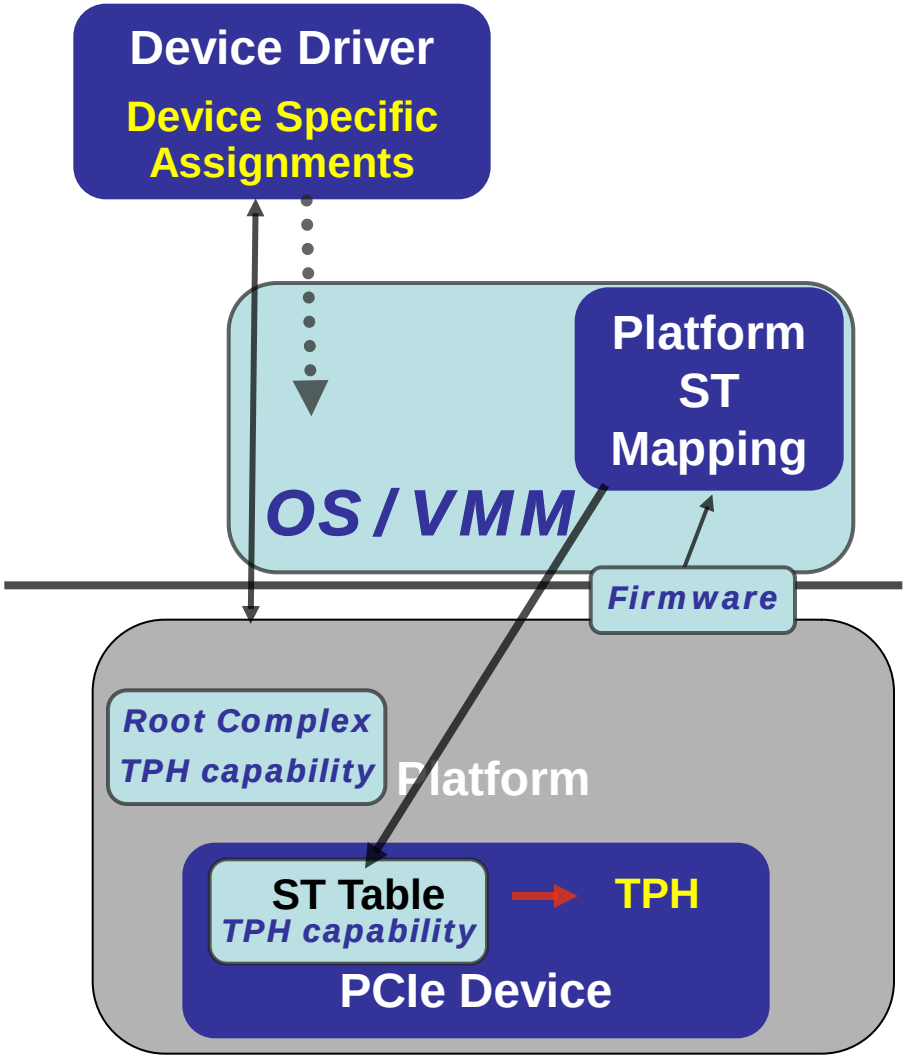
- *Platform support (Root Complex, Routing Elements)*
- *System specific Steering Tag advertisement*

Device Architecture

- *Characterize workloads*
- *Application processor Affinity*
- *Steering Tag to Workload association*
- *Select Modes of Operation*

Software Development

- *Basic Capability Discovery, Identification and Management*
- *System specific Steering Tag advertisement and assignment*
- *Device Driver enhancements*



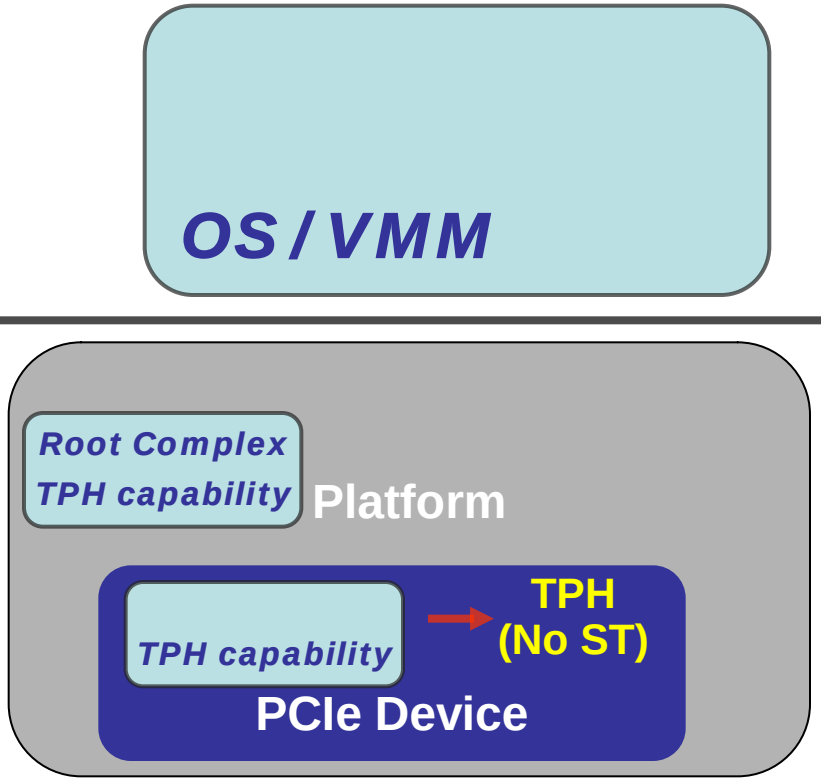
No ST Mode

No Steering Tags used

Request Steering is Platform Specific

Basic Capability Enablement

Minimal Implementation cost and complexity



Basic support

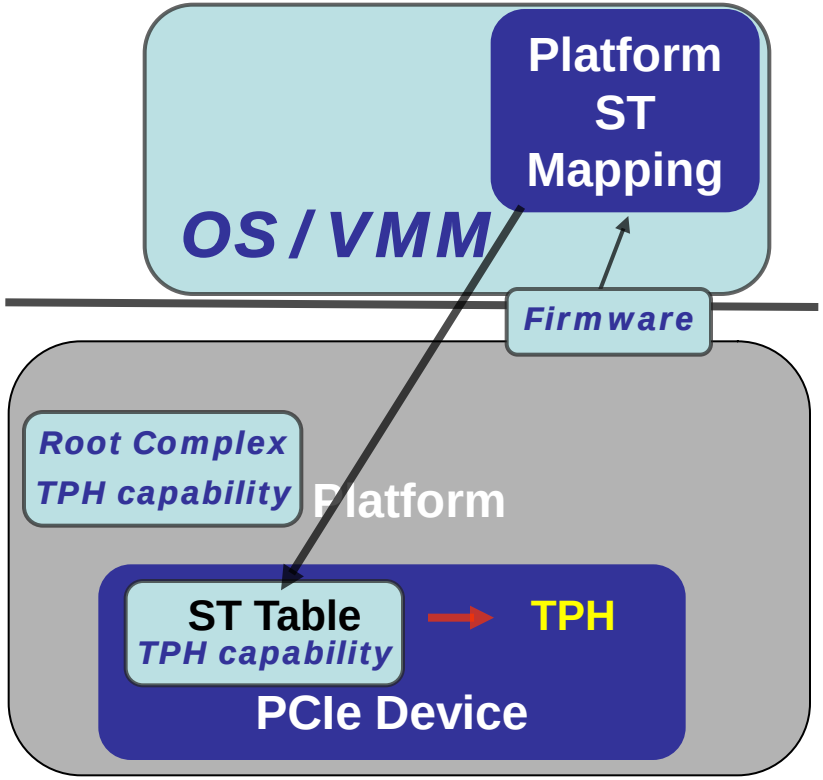
Interrupt Vector Mode

ST associated with Interrupt (MSI / MSI-X)

Firmware provides Platform specific ST information to OS / Hypervisor

OS / Hypervisor assigns ST along with Interrupt vector assignment

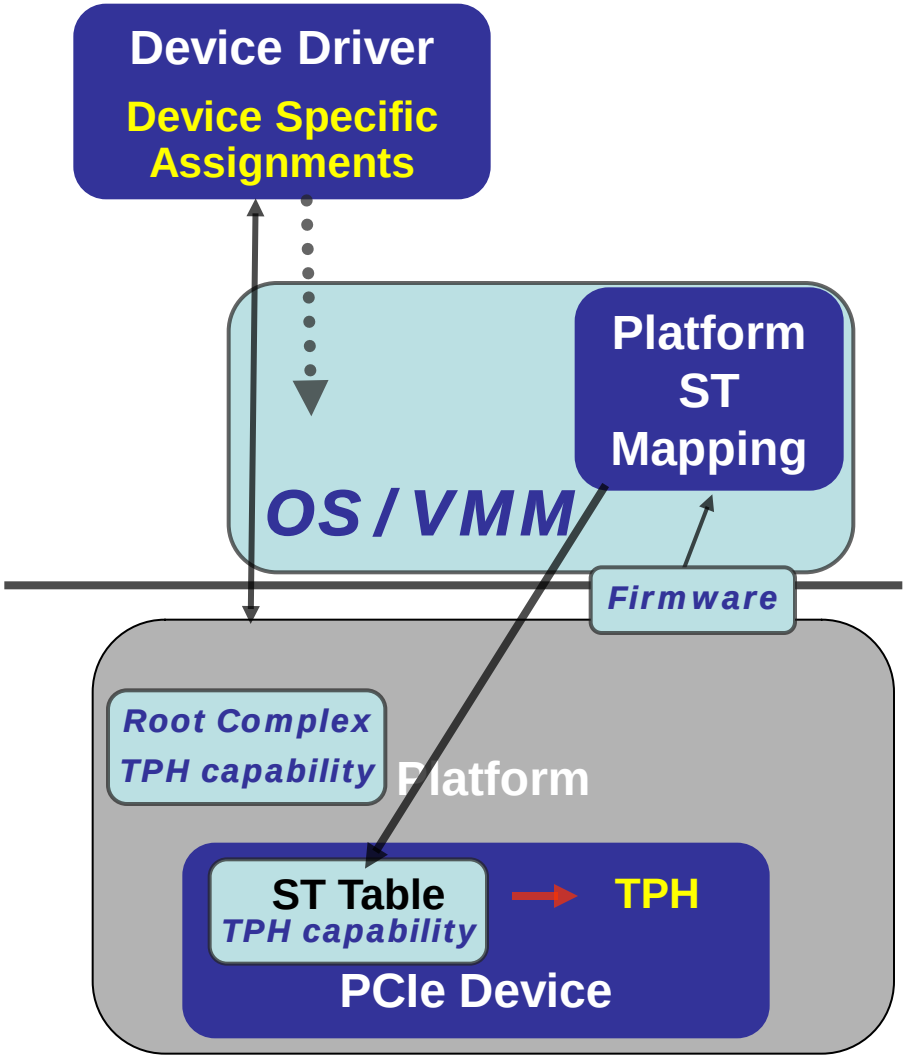
Suitable for devices with workload / Interrupt affinity to cores



TTM Advantage

Device Specific Mode

- Device Specific ST association
- Builds upon Interrupt vector mode s/w support
- Device Driver determines processor affinity
- New API required to request ST assignments
- Independent of Interrupt association



Scalable & Flexible Solution

TPH Aware Device Architecture

Classify device initiated transactions:

Bulk vs. Control

- **Select hints based on Data Struct. Use models**

Control Struct. (Descriptors)
Headers for Pkt. Processing
Data Payload (Copies)

Steering Tag Modes

- ✓ **No ST:** Basic Hints only, No ST used
- ✓ **Interrupt Vector Mode**: Faster TTM with Interrupt association
- ✓ **Device Specific Mode:** Scalable, Flexible & Dynamic, can provide TTM advantage

Software Development

- ✓ **Basic TPH Capability Identification, Discovery and Management**
- ✓ **Firmware Support to advertise ST assignments**
- ✓ **OS/Hypervisor ST assignment support**
- ✓ **Optional API support**

TPH permits Device Architecture Specific Trade-Offs

Device Architecture Considerations

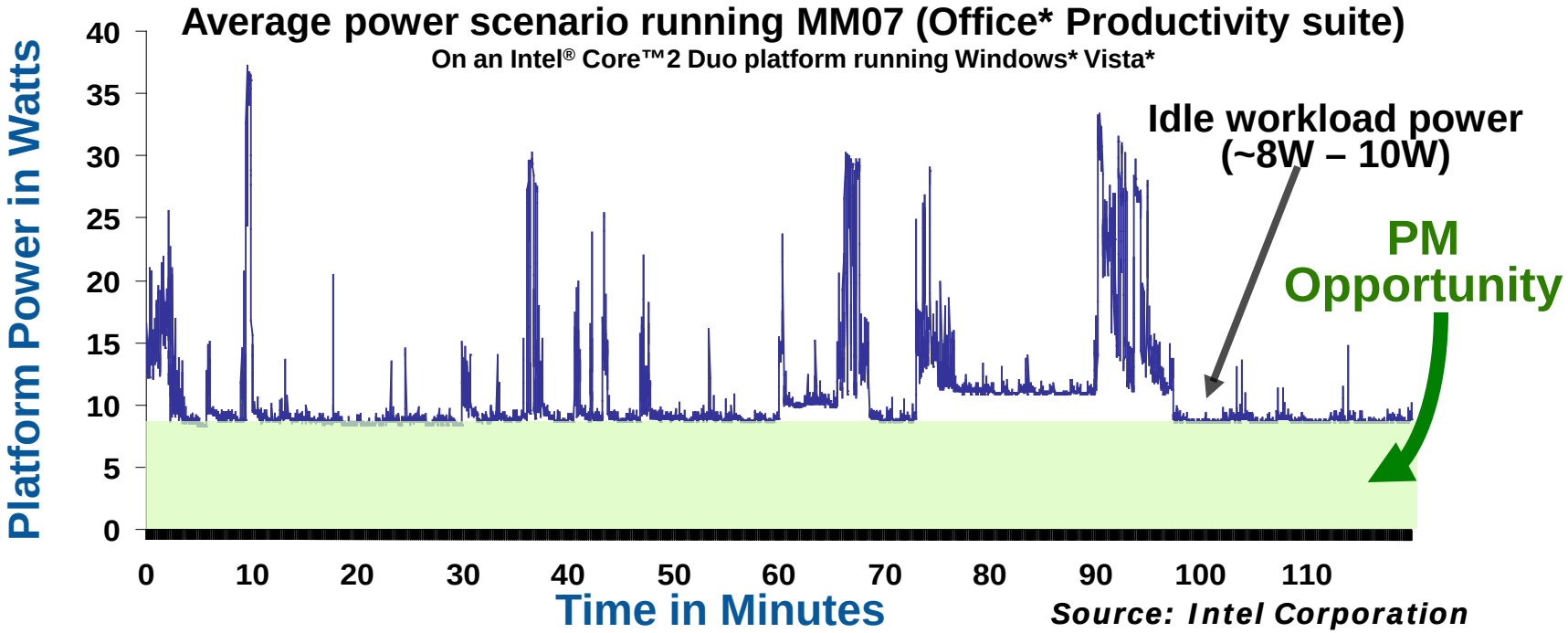
- TLP Processing Hints
- Power Management Extensions

System Perspective

- **Device behavior impacts power consumption of other system components**
  - Devices should consider their system power impact, not just their own device level power consumption
    - Extreme example: Enhanced Host Controller Interface (EHCI)
- **Systems (and devices) are idle most of the time**
  - There's a big opportunity for devices and systems to take advantage of that
- **Latency Tolerance Reporting (LTR) enables lower power, longer exit latency system power states *when devices can tolerate it***
- **Optimized Buffer Flush/Fill (OBFF) enables platform activity alignment, resulting in system power savings**

Opportunity for Devices to Differentiate on Platform Power Savings

Platform Power Savings Opportunity



- Usage Analysis: Typical mobile platform in S0 state is ~90% idle
- When idle, platform components are kept in high power state to meet the service latency requirements of devices & applications

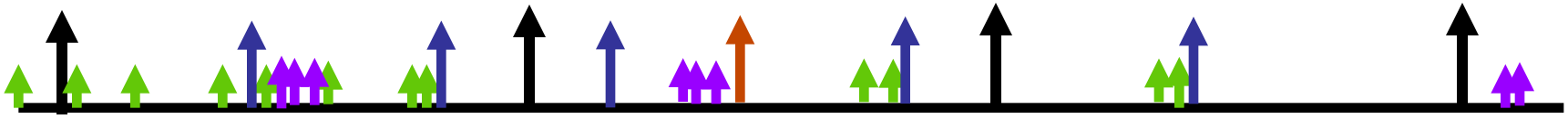
Power consumption for idle workload is high

Optimized Buffer Flush / Fill

- Next several foils describe:
  - Platform activity alignment
  - PCI Express* (PCIe*) OBFF mechanisms
  - OBFF within context of platform activity alignment
  - Device implementation impacts for OBFF

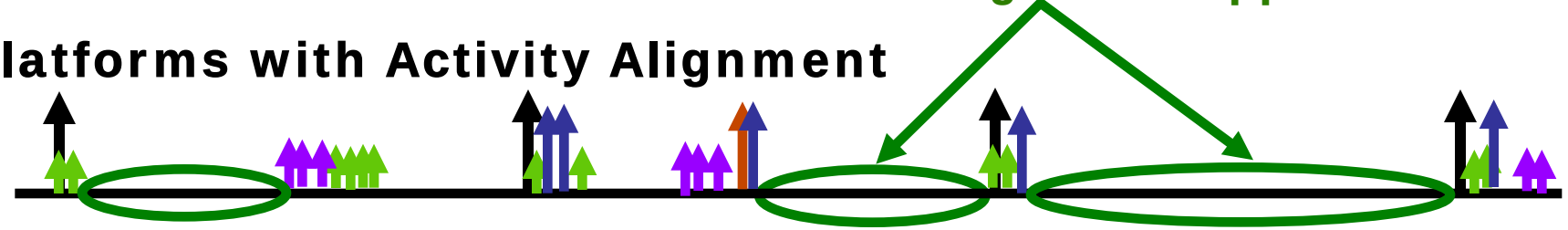
Platform Activity Alignment

Current Platforms



Power Management Opportunities

Platforms with Activity Alignment



OS tick inter-rupts

Device interrupts (critical)

- Time Critical
- Buffer replenish
- Performance / Throughput

Device interrupts (deferrable)

- Not time critical
- Status Notifications
- User Command Completions
- Debug, Statistics

Device traffic (critical)

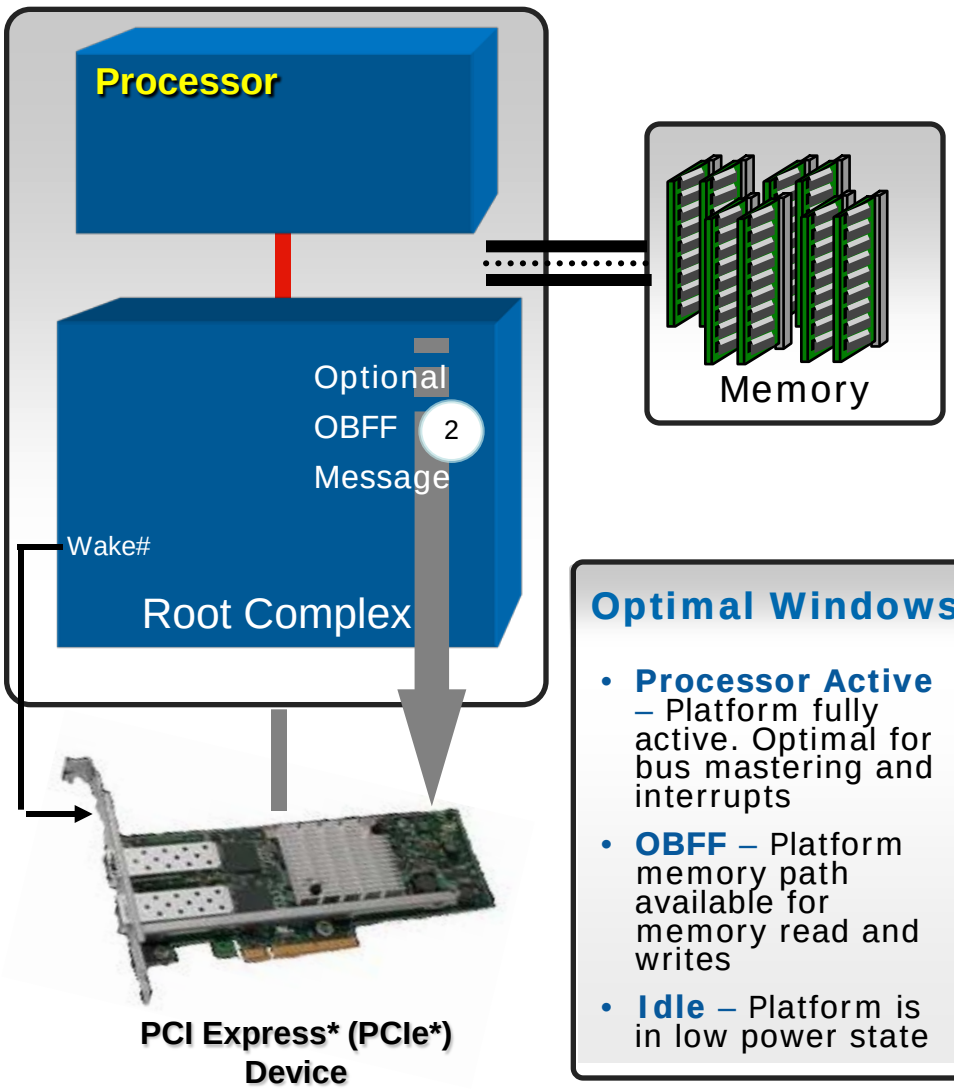
- Buffer over-(under-) run
- Throughput / Performance

Device traffic (deferrable)

- No Buffering constraints
- Debug dumps

Creates PM Opportunities for Semi-active workloads

Optimized Buffer Flush / Fill (OBFF)



OBFF

- Notify all Endpoints of optimal windows with minimal power impact

Solution1: When possible, use WAKE# with new wire semantics

Solution2: WAKE# not available – Use PCIe Message

Optimal Windows

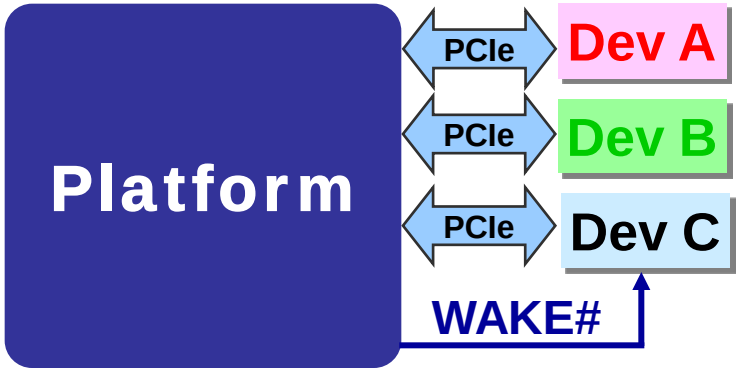
- **Processor Active** – Platform fully active. Optimal for bus mastering and interrupts
- **OBFF** – Platform memory path available for memory read and writes
- **Idle** – Platform is in low power state

WAKE# Waveforms

Transition Event	WAKE#
Idle → OBFF	
Idle → Proc. Active	
OBFF/Proc. Active → Idle	
OBFF → Proc. Active	
Proc. Active → OBFF	

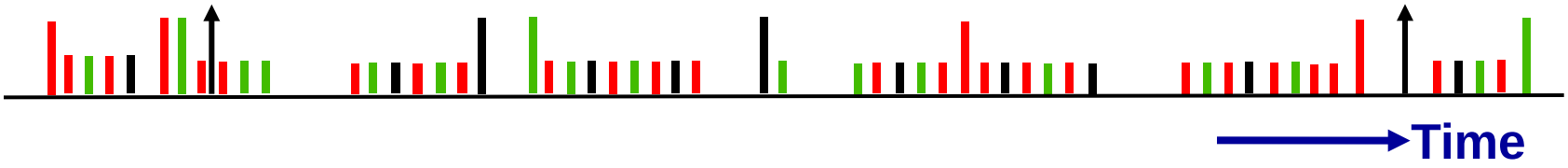
Greatest Potential Improvement When Implemented by All Platform Devices

OBFF and Activity Alignment

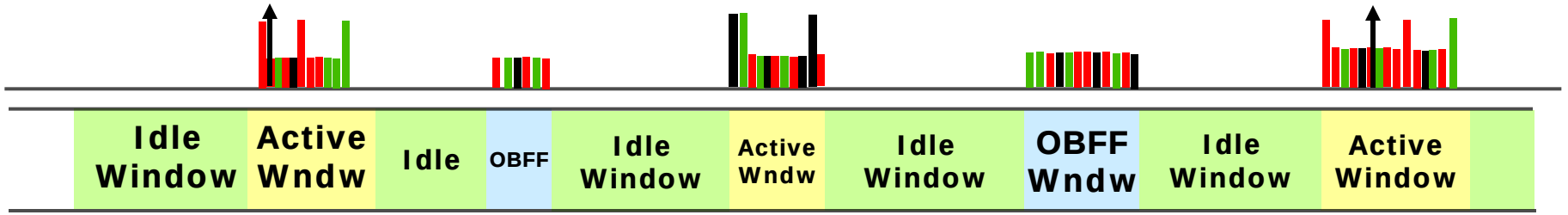


	Interrupts	DMAs
Device A		
Device B		
Device C		
OS Timer Tick (intrpt)		

Traffic Pattern with No OBFF



Traffic Pattern with OBFF



OBFF Device Implementation Impacts

Maximize idle window duration for platform

- Align transactions with other devices in system
- Coalesce transactions into groups where possible
  - Perform groups of transactions all at once, don't trickle all the time

Classify device initiated transactions: critical vs. deferrable

- Perform critical transactions as necessary
- Defer other transactions to align with platform activity
  - Decode platform idle / active / OBFF window signaling

Select data buffer depths to tolerate platform activity alignment

- 300µs of deferral buffering recommended for Intel mobile platforms

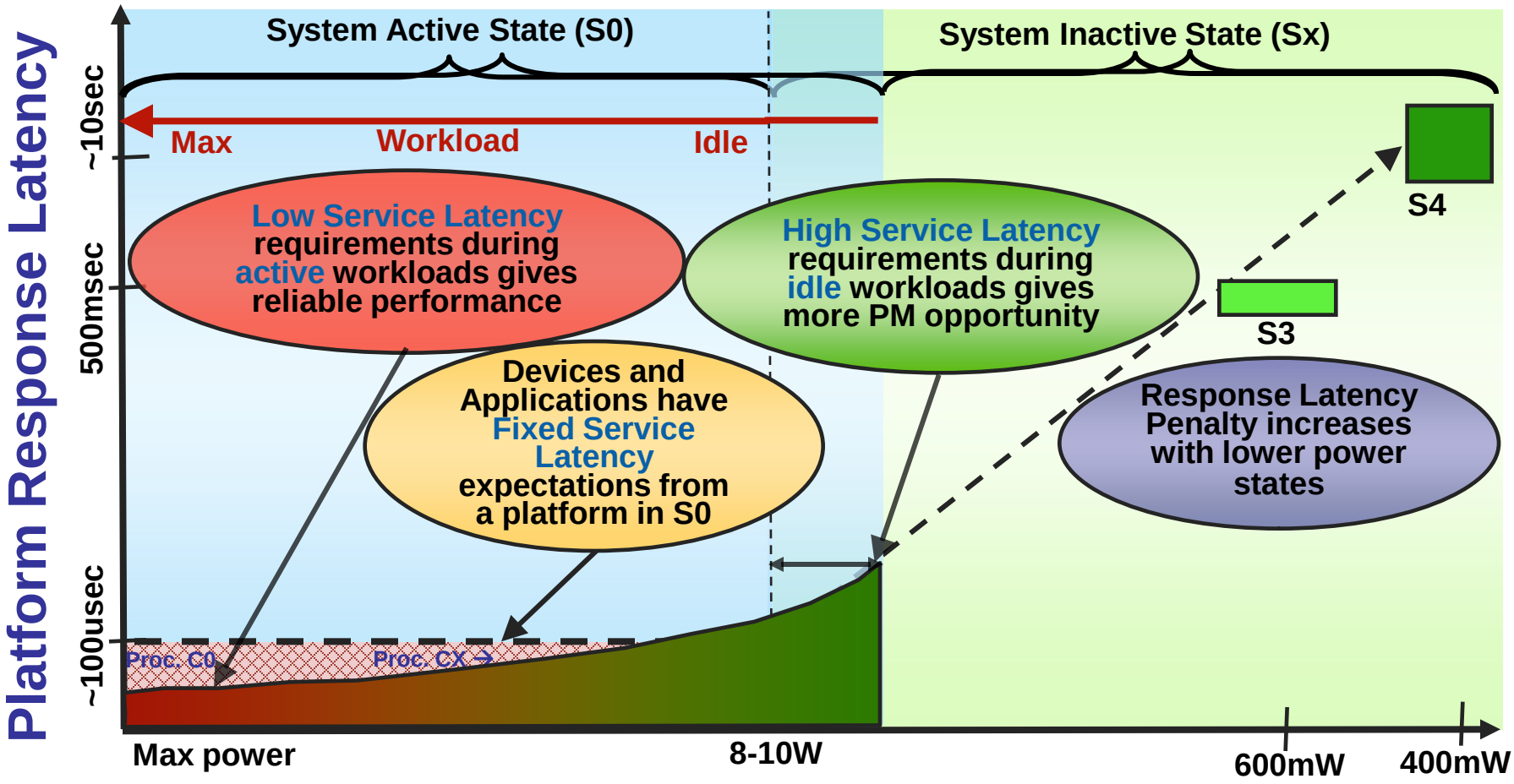
19

Latency Tolerance Reporting

- The next several foils describe:
  - Power vs. response latency
  - PCI Express* (PCIe*) LTR mechanisms, semantics
  - Examples of device implementation schemes
    - Application state driven LTR reporting
    - Data buffer depth driven LTR reporting
    - Software guided LTR reporting
  - Device implementation impact summary for LTR

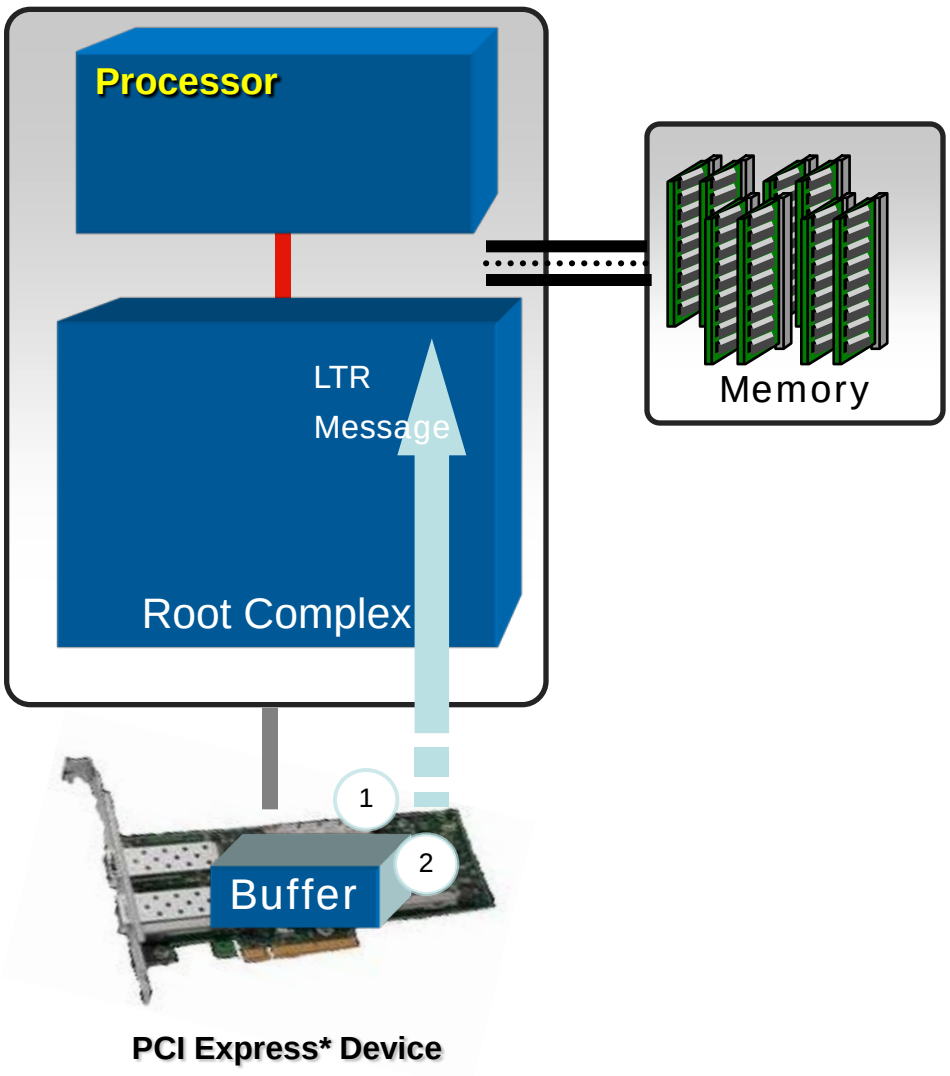
20

Power Vs Response Latency (Mobile)



Variable Service Latency requirements in S0 is Optimal

Latency Tolerance Reporting (LTR)



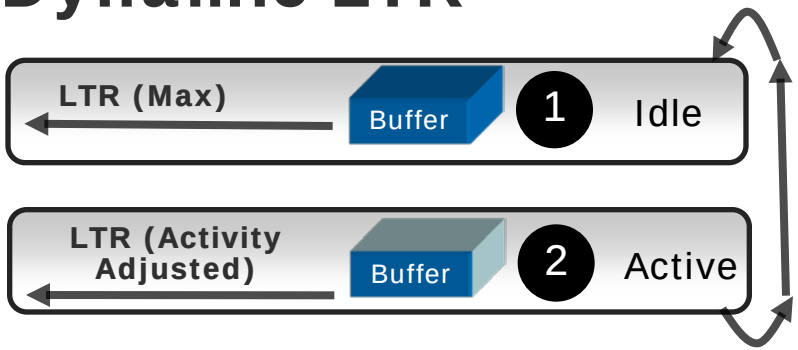
LTR Mechanism

- PCI Express* (PCIe*) Message sent by Endpoint with tolerable latency
  - Capability to report both snooped & non-snooped values
  - "Terminate at Receiver" routing, MFD & Switch send aggregated message

Benefits

- Provides Device Benefit: Dynamically tune platform PM state as a function of Device activity level
- Platform benefit: Enables greater power savings without impact to performance/functionality

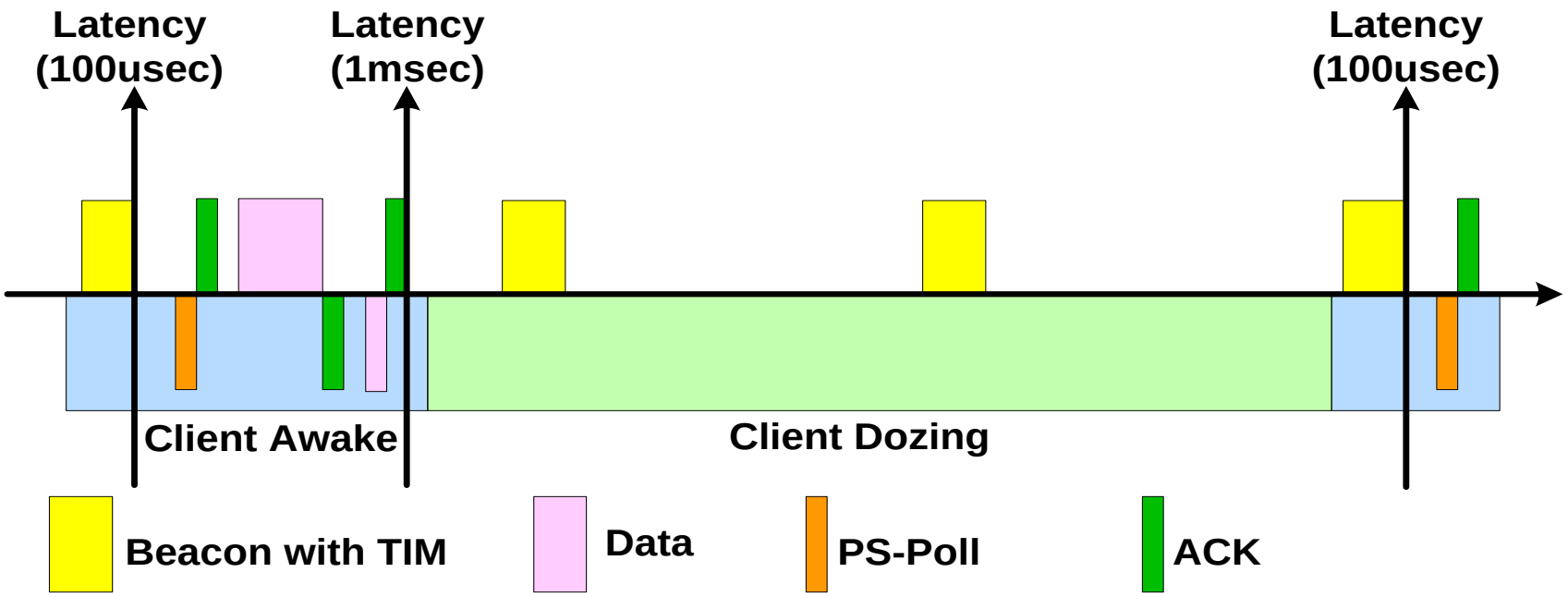
Dynamic LTR



LTR enables dynamic power vs. performance tradeoffs at minimal cost impact

Application State Driven LTR

Example: WLAN Device Sending LTR

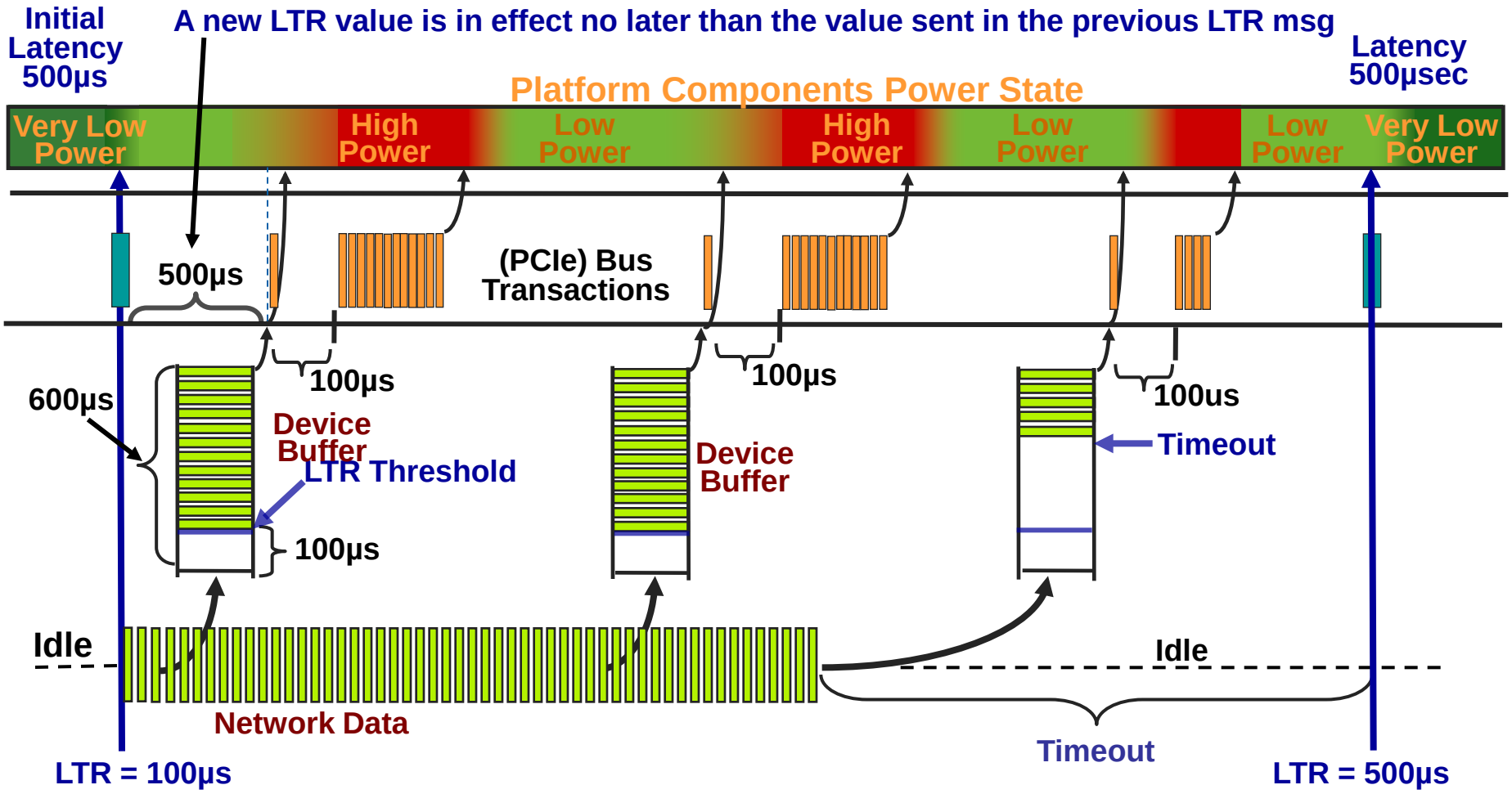


Latency information with Wi-Fi Legacy Power Save

Example use of device PM states to give latency guidance

Data Buffer Utilization Guided LTR

Example: Active Ethernet NIC Sending LTR

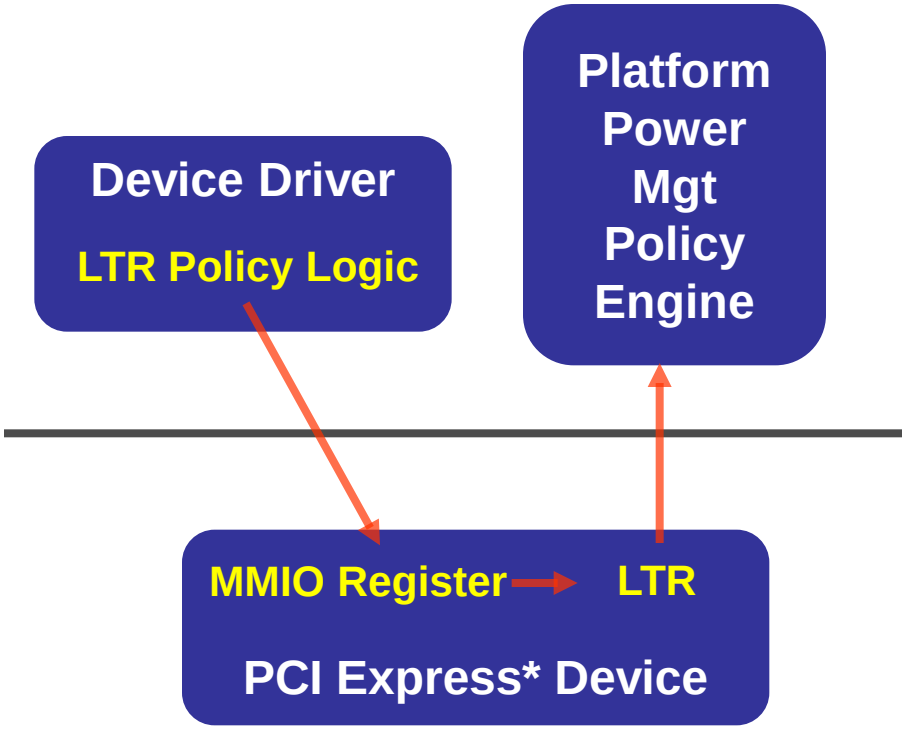


Example use of buffering to give latency guidance

Software Guided Latency

SW Guided Latency

- Three device categories
  - ✓ **Static:** Device can always support max platform latency
  - ✓ **Slow Dynamic**: Latency requirements change infrequently
  - ✓ **Fast Dynamic:** Latency requirements change frequently
- Static and Slow Dynamic types of devices may choose SW guided messaging
  - ✓ Policy logic for determination of when to send latency messages (and what values) in software
  - ✓ E.g. use an MMIO register
    - ➢ A write to the register would trigger an LTR message



25

LTR Device Implementation Impacts

When idle, let platform enter deep power saving states

- Use MaxLatency (LTR Extended Capabilities field) when idle
- Require low latencies only when necessary – don't keep platform in high power state longer than necessary

Dynamic, hardware driven LTR

- Leverage application based opportunities to tolerate more latency
  - E.g. WLAN radio off between beacons
- Implement data buffering mechanism to comprehend LTR

Software guided LTR

- Implement simple MMIO register interface
  - Register writes cause LTR message to be sent

26

Software Enabling

Features requiring basic software support

Capability Discovery, Identification and Management

- 8GT/s speed upgrade
- Atomics
- Transaction Ordering Relaxations
- Internal Error Reporting
- TLP Prefix

Features requiring additional support

Above and beyond capability enablement

Resource Allocation, Enumeration & API

- Transaction Processing Hints
- LTR & OBFF
- Resizable BAR
- IO Page Faults
- Dynamic Power Allocation
- Multicast

Summary

- Next Generation PCI Express* (PCIe*) Protocol Extensions Deliver Energy Efficient Performance
  - Protocol Extensions with Broad Applicability
- Ecosystem Development is essential
  - Platform Support
  - Device architectures optimized around protocol features
  - Software support and Enabling

Call to Action

- Device Architecture Considerations
  - Develop Device Architecture to make the most of the most of proposed protocol extensions
- Differentiate products utilizing TPH
  - Select Hints/ST modes based on device/market segment requirements
- Differentiate products utilizing LTR and OBFF
  - Can differentiate by platform power impact not just device power
  - Power Savings opportunity is huge
- Keep track of Next Generation PCI Express* Technology development
  - PCI-SIG www.pcisig.com
- Engage with Intel on Next Generation PCI Express product development
  - www.intel.com/technology/pciexpress/devnet

Acknowledgements / Disclaimer

- I would like to acknowledge the contributions of the following Intel employees
  - Stephen Whalley, Intel Corporation
  - Jasmin Ajanovic, Intel Corporation
  - Anil Vasudevan, Intel Corporation
  - Prashant Sethi, Intel Corporation
  - Simer Singh, Intel Corporation
  - Miles Penner, Intel Corporation
  - Mahesh Natu, Intel Corporation
  - Rob Gough, Intel Corporation
  - Eric Wehage, Intel Corporation
  - Jim Walsh, Intel Corporation
  - Jaya Jeyaseelan, Intel Corporation
  - Neil Songer, Intel Corporation
  - Barnes Cooper, Intel Corporation

All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG*.

Additional Sources of Information on This Topic

- Other Sessions
  - **TSIS007** –PCI Express* 3.0 Technology: PHY Implementation Considerations on Intel Platforms
  - **TSIS008** – PCI Express* 3.0 Technology: Electrical Requirements for Designing ASICs on Intel Platforms
  - **TCIQ002:** Q&A: PCI Express* 3.0 Technology

- www.intel.com/technology/pciexpress/devnet

Legal Disclaimer

- 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. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards 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.
- Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- Intel, Intel Inside, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the third quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the corporation's expectations. Ongoing uncertainty in global economic conditions pose a risk to the overall economy as consumers and businesses may defer purchases in response to tighter credit and negative financial news, which could negatively affect product demand and other related matters. Consequently, demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including conditions in the credit market that could affect consumer confidence; customer acceptance of Intel's and competitors' products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Additionally, Intel is in the process of transitioning to its next generation of products on 32nm process technology, and there could be execution issues associated with these changes, including product defects and errata along with lower than anticipated manufacturing yields. Revenue and the gross margin percentage are affected by the timing of new Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; and Intel's ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on changes in revenue levels; capacity utilization; start-up costs, including costs associated with the new 32nm process technology; variations in inventory valuation, including variations related to the timing of qualifying products for sale; excess or obsolete inventory; product mix and pricing; manufacturing yields; changes in unit costs; impairments of long-lived assets, including manufacturing, assembly/test and intangible assets; and the timing and execution of the manufacturing ramp and associated costs. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. The current financial stress affecting the banking system and financial markets and the going concern threats to investment banks and other financial institutions have resulted in a tightening in the credit markets, a reduced level of liquidity in many financial markets, and heightened volatility in fixed income, credit and equity markets. There could be a number of follow-on effects from the credit crisis on Intel's business, including insolvency of key suppliers resulting in product delays; inability of customers to obtain credit to finance purchases of our products and/or customer insolvencies; counterparty failures negatively impacting our treasury operations; increased expense or inability to obtain short-term financing of Intel's operations from the issuance of commercial paper; and increased impairments from the inability of investee companies to obtain financing. The majority of our non-marketable equity investment portfolio balance is concentrated in companies in the flash memory market segment, and declines in this market segment or changes in management's plans with respect to our investments in this market segment could result in significant impairment charges, impacting restructuring charges as well as gains/losses on equity investments and interest and other. Intel's results could be impacted by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. A detailed discussion of these and other risk factors that could affect Intel's results is included in Intel's SEC filings, including the report on Form 10-Q for the quarter ended June 27, 2009.