PCI-SIG® Architecture Overview

Betty Luk
AMD

What's all this PCI stuff anyway?

- **Presentation will cover basic concepts and their evolution from PCI™ through PCI-X™ to PCI Express®**
  - ✓ **Specs written assuming designers have these key background concepts**
  - ✓ **High level overview of PCI, PCI-X, and PCI Express**

PCI Background

Revolutionary AND Evolutionary

- PCI
  - ✓ Revolutionary
    - Plug and Play jumperless configuration (BARs)
    - Unprecedented bandwidth
      - 32-bit / 33MHz – 133MB/sec
      - 64-bit / 66MHz – 533MB/sec
    - Designed from day 1 for bus-mastering adapters
  - ✓ Evolutionary
    - System BIOS maps devices then operating systems boot and run without further knowledge of PCI
    - PCI-aware O/S could gain improved functionality

Revolutionary AND Evolutionary

- **PCI-X**
  - ✓ **Revolutionary**
    - **Unprecedented bandwidth**
      - **Up to 1066MB/sec with 64-bit / 133MHz**
    - **Registered bus protocol**
      - **Eased electrical timing requirements**
    - **Brought split transactions into PCI "world"**
  - ✓ **Evolutionary**
    - **PCI compatible at hardware *AND* software levels**
    - **PCI-X 266/533 added as "mid-life" performance bump**
      - **2133MB/sec at PCI-X 266 and 4266MB/sec at PCI-X 533**

Revolutionary AND Evolutionary

- **PCI Express (aka PCIe®)**
  - ✓ **Revolutionary**
    - **Unprecedented bandwidth**
      - **x1: 250MB/sec in *EACH* direction**
      - **x16: 4000MB/sec in *EACH* direction**
    - **"Relaxed" electricals due to serial bus architecture**
      - **Point-to-point, low voltage, dual simplex with embedded clocking**
  - ✓ **Evolutionary**
    - **PCI compatible at software level**
      - **Configuration space**
      - **Power Management**
      - **Of course, PCIe-aware O/S can get more functionality**
    - **Transaction layer familiar to PCI/PCI-X designers**
    - **System topology matches PCI/PCI-X**

PCI Concepts

PCI Concepts

- **Address spaces**
  - ✓ **Memory – 64-bit**
  - ✓ **I/O – 32-bit (non-burstable since PCI-X)**
  - ✓ **Configuration ("Config") – Bus/Device/Function**

- **Key configuration space regs/concepts**
  - ✓ ***B*ase *A*ddress *R*egisters (BARs)**
    - – **64-bit vs 32-bit addressing**
  - ✓ **Linked list of capabilities**

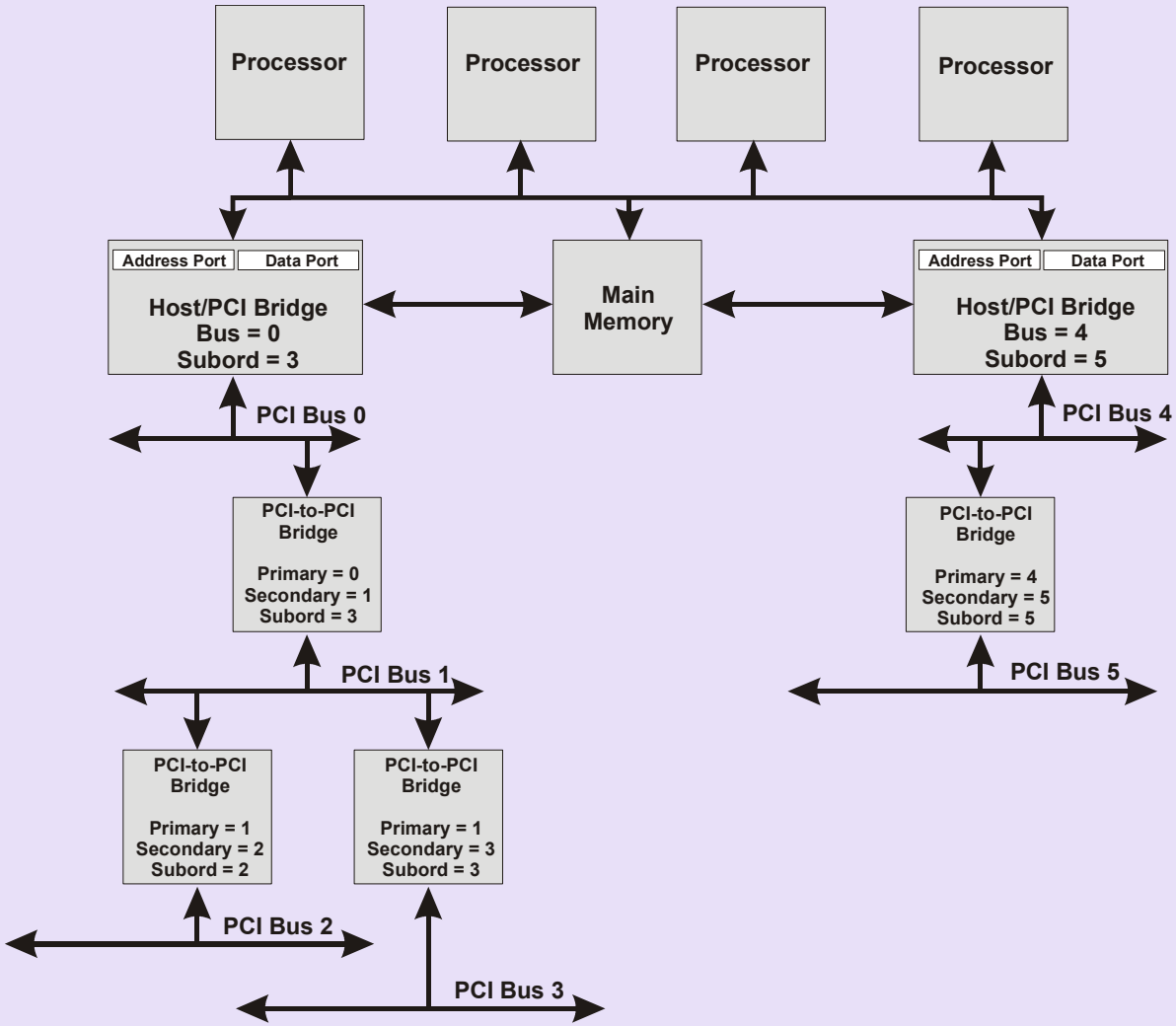
Address spaces – Memory & I/O

- **Memory space mapped cleanly to CPU semantics**
  - ✓ **32-bits of address space initially**
  - ✓ **64-bits introduced via Dual-Address Cycles (DAC)**
    - – **Extra clock of address time on PCI/PCI-X**
    - – **4DWORD header in PCI Express**
  - ✓ **Burstable**
- **I/O space mapped cleanly to CPU semantics**
  - ✓ **32-bits of address space**
    - – **Actually much larger than CPUs of the time**
  - ✓ **Non-burstable**
    - – **Most PCI implementations didn't support**
    - – **PCI-X codified**
    - – **Carries forward to PCI Express**

Address spaces – Configuration

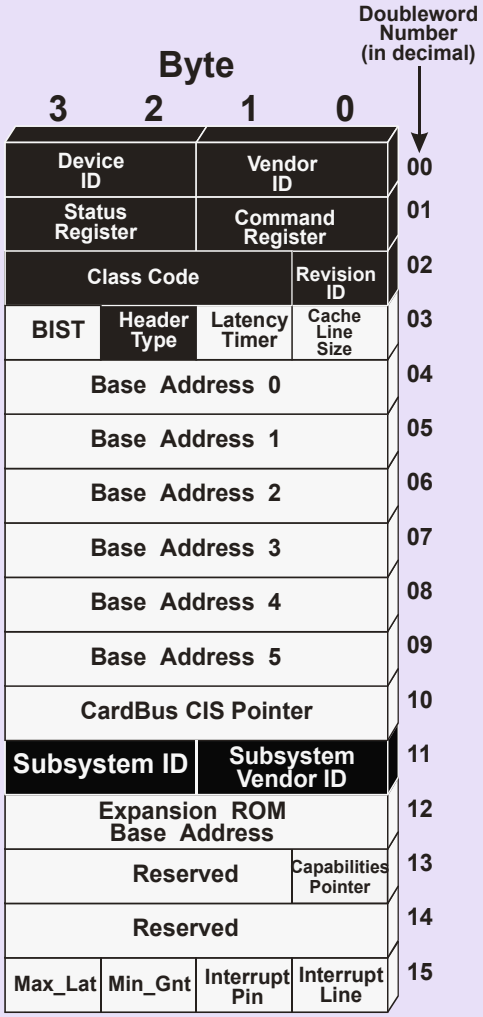
- Configuration space???
  - ✓ Allows control of devices' address decodes without conflict
  - ✓ No conceptual mapping to CPU address space
    - – Memory-based access mechanisms introduced with PCI-X and PCIe
  - ✓ Bus / Device / Function (aka BDF) form hierarchy-based address
    - – "Functions" allow multiple, logically independent agents in one physical device.
      - • E.g. combination SCSI + Ethernet device
      - • 256 bytes or 4K bytes of configuration space per device
    - – PCI/PCI-X bridges form hierarchy
    - – PCIe switches form hierarchy
      - • Look like PCI-PCI bridges to software
  - ✓ "Type 0" and "Type 1" configuration cycles
    - – Type 0: to same bus segment
    - – Type 1: to another bus segment

Configuration Space (cont'd)

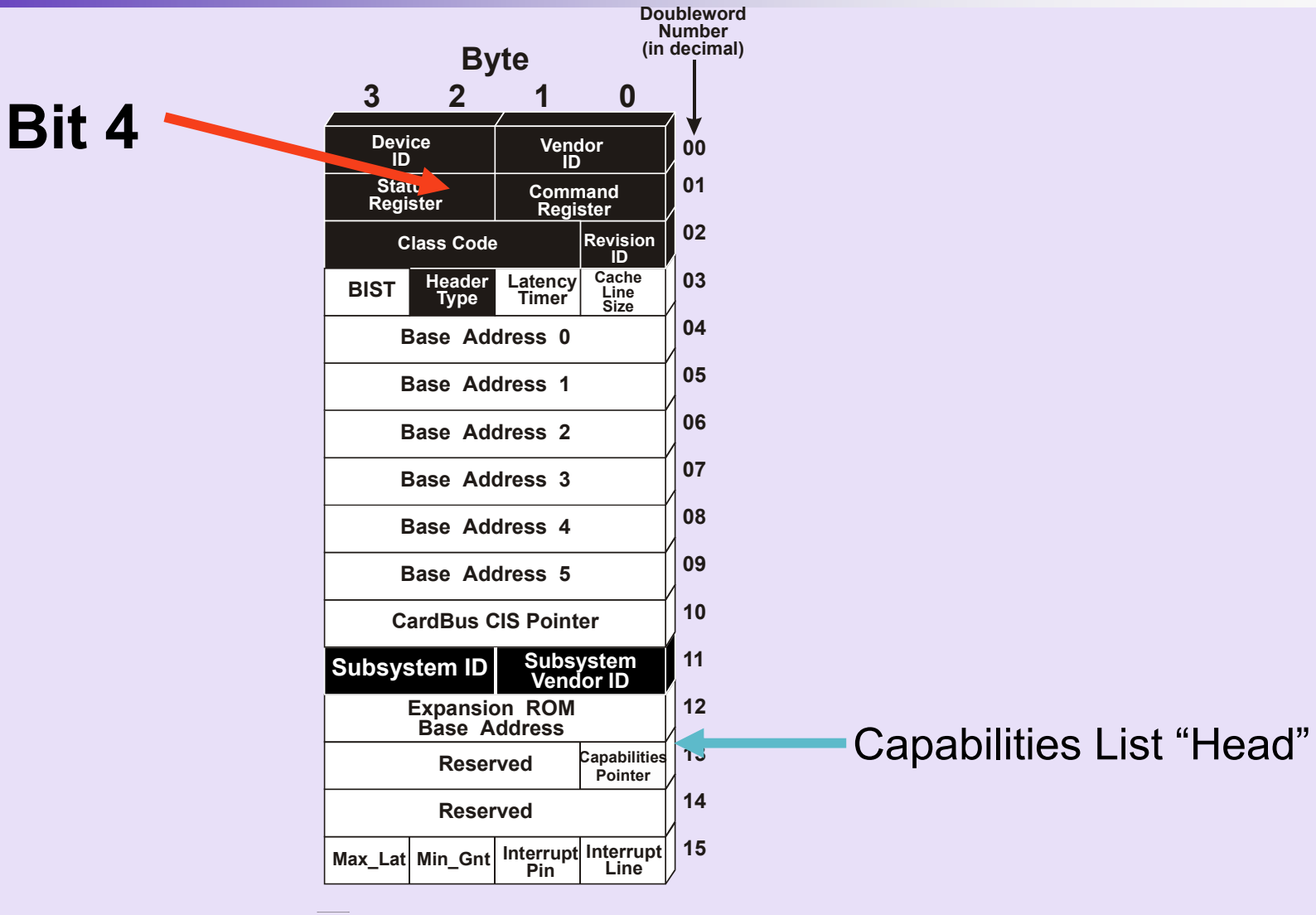


Using Configuration Space

- **Device Identification**
  - ✓ VendorID: PCI-SIG assigned
  - ✓ DeviceID: Vendor self-assigned
  - ✓ Subsystem VendorID: PCI-SIG
  - ✓ Subsystem DeviceID: Vendor
- **Address Decode controls**
  - ✓ Software reads/writes BARs to determine required size and maps appropriately
  - ✓ Memory, I/O, and bus-master enables
- **Other bus-oriented controls**

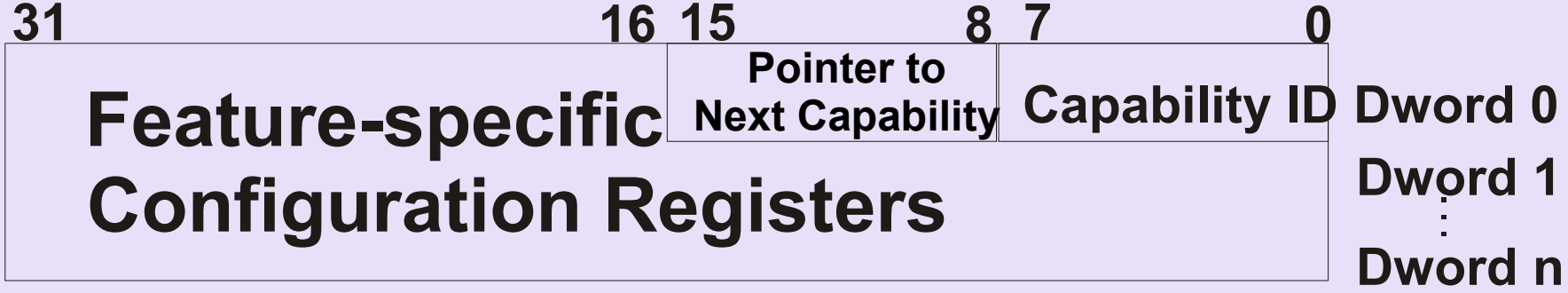


Using Configuration Space – Capabilities List



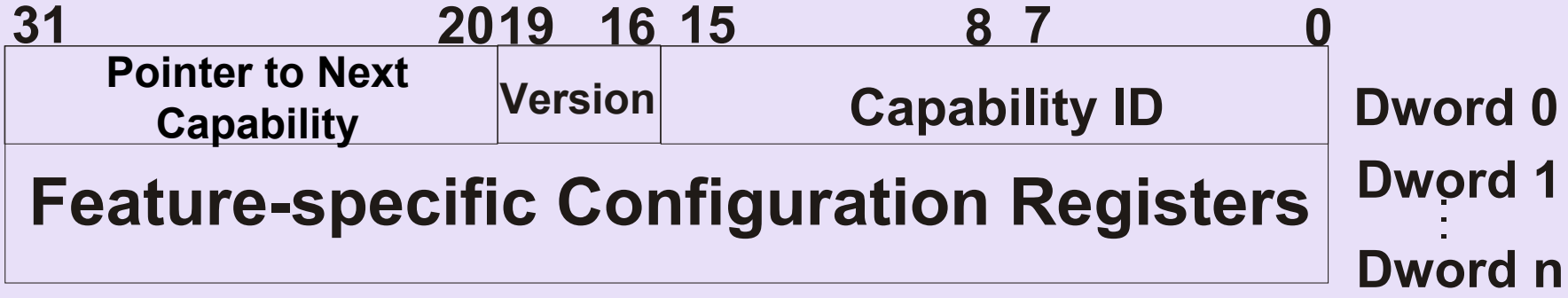
Using Configuration Space – Capabilities List (cont'd)

- **Linked list**
  - ✓ **Follow the list! Cannot assume fixed location of any given feature in any given device**
  - ✓ **Features defined in their related specs:**
    - **PCI-X**
    - **PCIe**
    - **PCI Power Management**
    - **Etc…**



Using Configuration Space – Extended Capabilities List

- PCI Express only
- Linked list
  - ✓ Follow the list! Cannot assume fixed location of any given feature in any given device
  - ✓ First entry in list is *always* at 100h
  - ✓ Features defined in PCI Express specification



Interrupts

- PCI introduced INTA#, INTB#, INTC#, INTD# - collectively referred to as INTx
  - ✓ Level sensitive
  - ✓ Decoupled device from CPU interrupt
  - ✓ System controlled INTx to CPU interrupt mapping
  - ✓ Configuration registers
    - report A/B/C/D
    - programmed with CPU interrupt number
- PCI Express mimics this via “virtual wire” messages
  - ✓ Assert_INTx and Deassert_INTx

What are MSI and MSI-X?

- **Memory Write replaces previous interrupt semantics**
  - ✓ **PCI and PCI-X devices stop asserting INTA, INTB, INTC, INTD once MSI or MSI-X mode is enabled**
  - ✓ **PCI Express devices stop sending Assert_INTx and Deassert_INTx TLPs once MSI or MSI-X mode is enabled**

- **NOTE: *Boot devices* and any device intended for a non-MSI operating system generally must still support the appropriate INTx signaling!**

MSI vs MSI-X

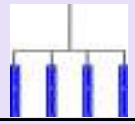
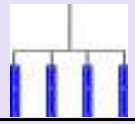
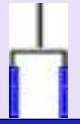
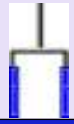
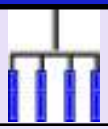
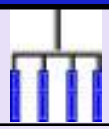
- MSI uses one address with a variable data value indicating which “vector” is asserting
- MSI-X uses a table of independent address and data pairs for each “vector”
  - ✓ Allows software to control aliasing (when fewer vectors are allocated than requested)
  - ✓ Table size supports more vectors than MSI structure allowed

PCI-X Explained

What is PCI-X?

- "PCI-X is high-performance backward compatible PCI"
  - ✓ PCI-X uses the same PCI architecture
  - ✓ PCI-X leverages the same base protocols as PCI
  - ✓ PCI-X leverages the same BIOS as PCI
  - ✓ PCI-X uses the same connector as PCI.
  - ✓ PCI-X and PCI products are interoperable
  - ✓ PCI-X uses same software driver models as PCI
- PCI-X is faster PCI
  - ✓ PCI-X 533 is up to 32 times faster than the original version of PCI
  - ✓ PCI-X protocol is more efficient than conventional PCI

PCI-X Modes and Speeds

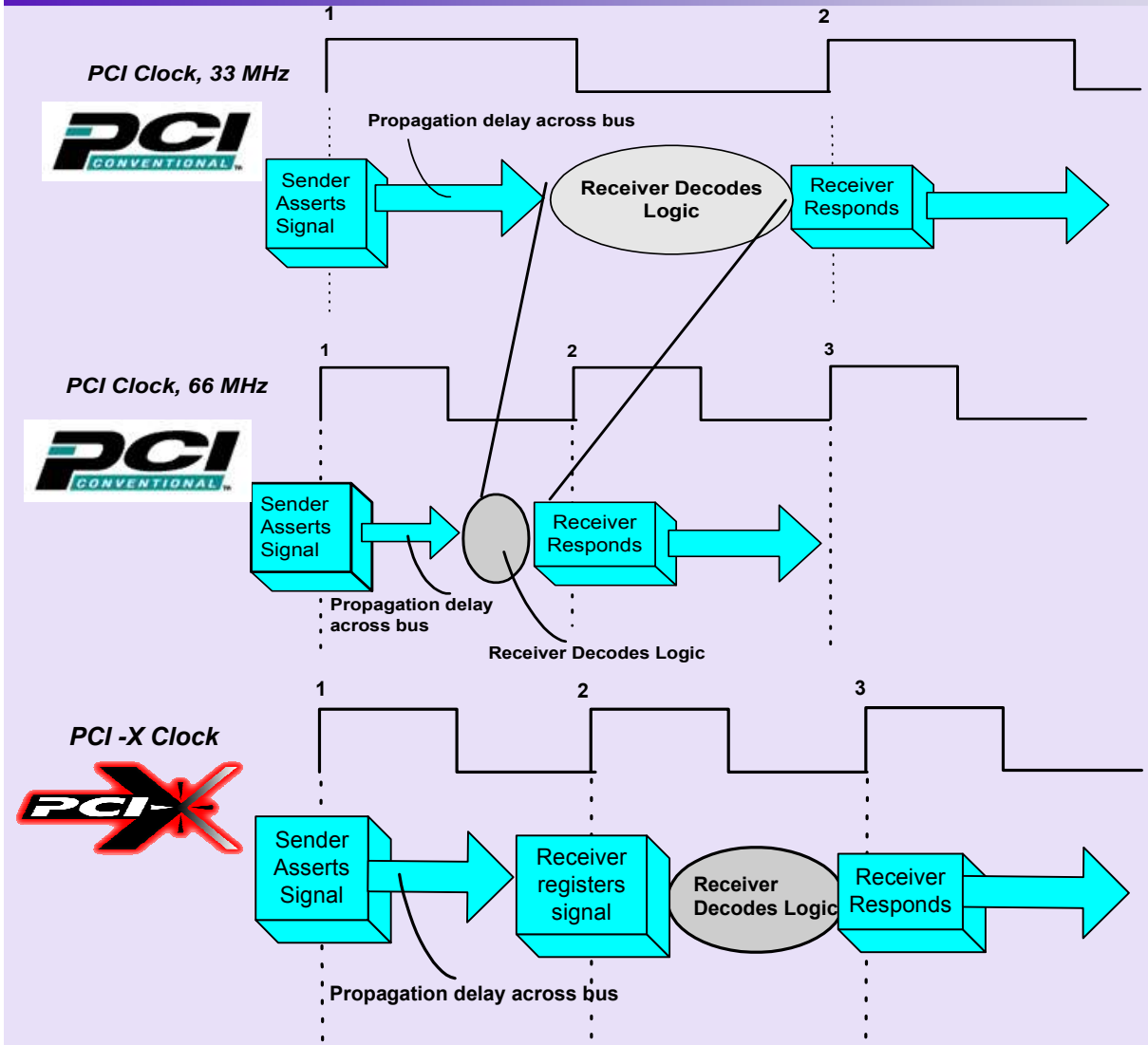
Mode	$V_{I/O}$	64-Bit Slots*	64-Bit MB/s	32-Bit Slots*	32-Bit MB/s	16-Bit	Error Prot	Conf Bytes	DIM
PCI 33	5V/3.3V		266		133	N/A	par	256	N/A
PCI 66	3.3V		533		266	N/A	par	256	N/A
PCI-X 66	3.3V		533		266	N/A	par or ECC	256	yes
PCI-X 133 (operating at 100 MHz)	3.3V		800		400	N/A	par or ECC	256	yes
PCI-X 133	3.3V		1066		533	N/A	par or ECC	256	yes
PCI-X 266	1.5V		2133		1066	533	ECC	4K	yes
PCI-X 533	1.5V		4266		2133	1066	ECC	4K	yes

Mode 1

Mode 2

* For lower bus speeds, # slots / bus is implementation choice to share bandwidth

Registered Bus Protocol

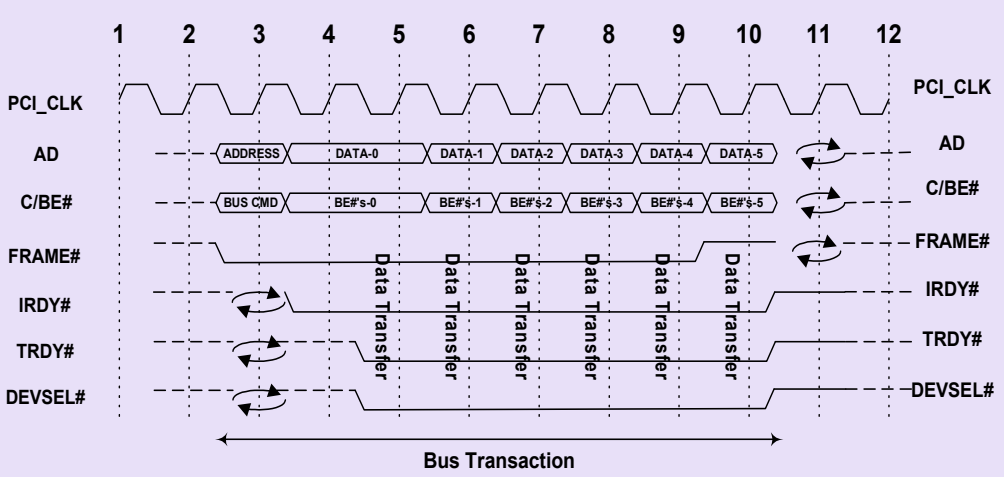


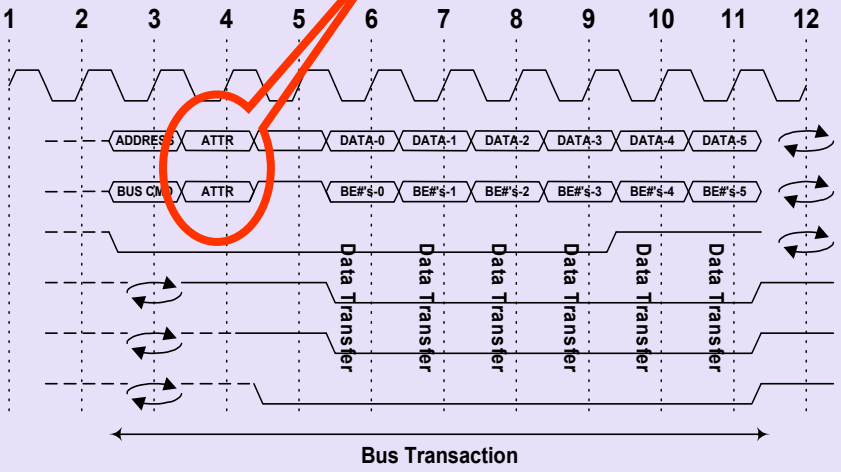
- PCI @ 33MHz
  - ✓30 ns period
  - ✓7 ns setup time
- PCI @ 66MHz
  - ✓ 15 ns period
  - ✓ 3ns setup time
- PCI-X registered protocol allocates a full clock period for logic decision
  - ✓ @ 66MHz - 15ns
  - ✓ @ 133MHz - 7.5ns

PCI 2.x/3.0 vs. PCI-X Mode 1

- Same bus and control signals
- Evolutionary protocol changes
- Clock frequency up to 133 MHz

New "Attribute" phase for enhanced features

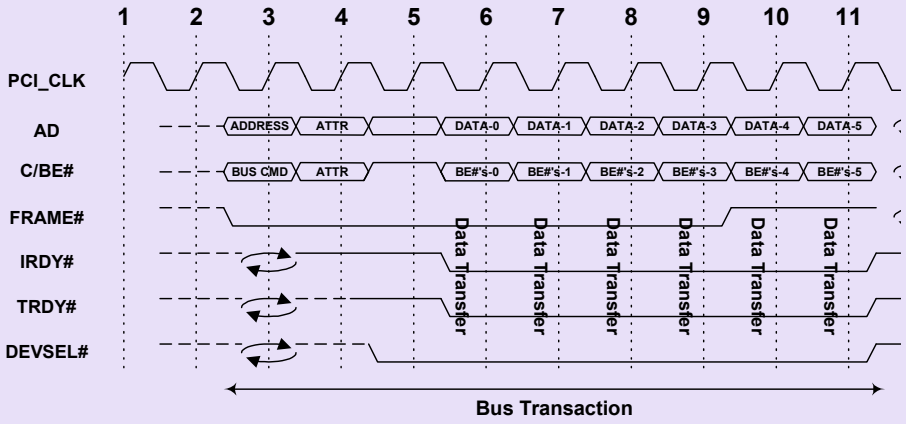




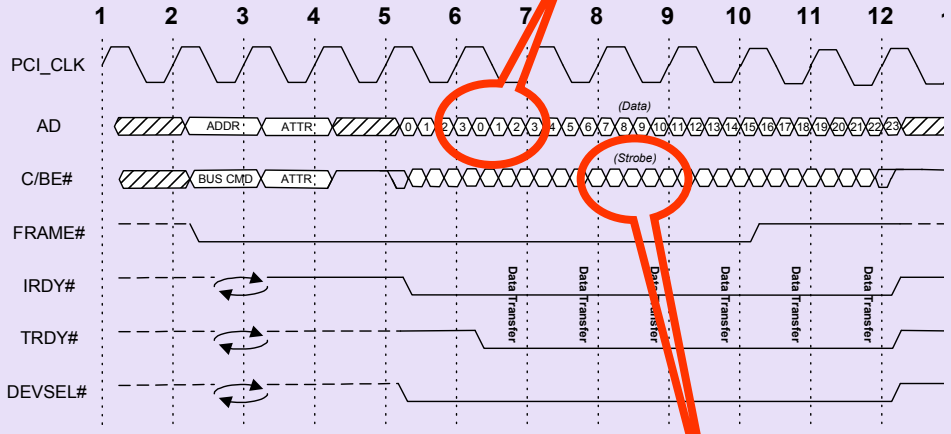
(Common clock)

PCI-X 66/133 (Mode 1) vs. PCI-X 266/533 (Mode 2)

- Same bus and control signals
- PCI-X 266 moves 2x the data
  PCI-X 533 moves 4x the data
- Clock frequency up to 133 MHz



PCI-X 66/133 (Mode 1)



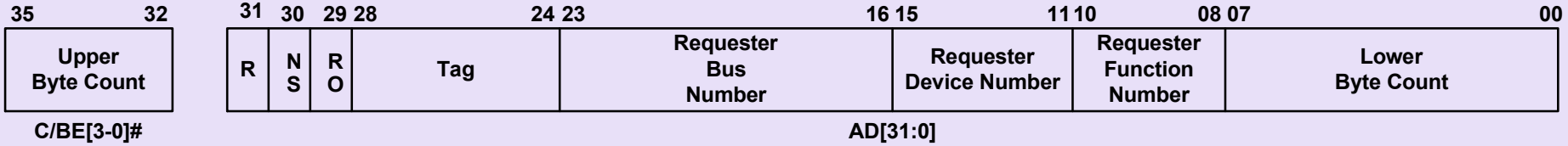
PCI-X 533 (Mode 2)

4 transfers per clock cycle

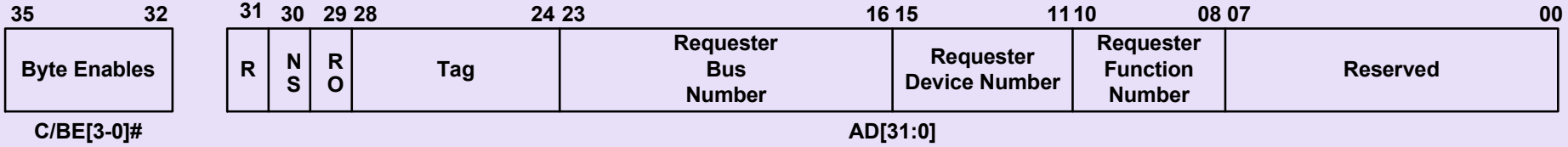
source-synchronous data strobes share C/BE pins

Transaction Attributes

Requester Attributes for Burst Transactions



Requester Attributes for DWORD Transactions



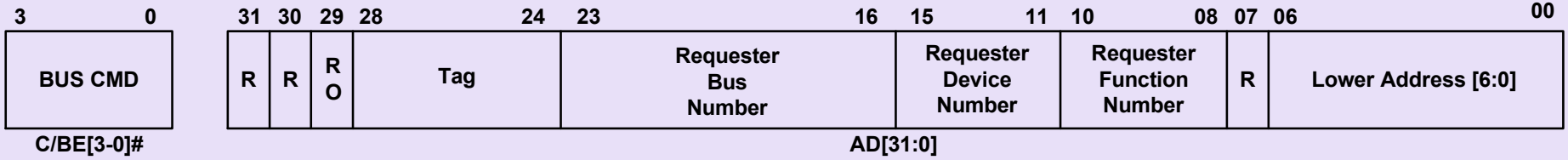
RO -- Relax ordering

NS -- No Snoop

R -- Reserved

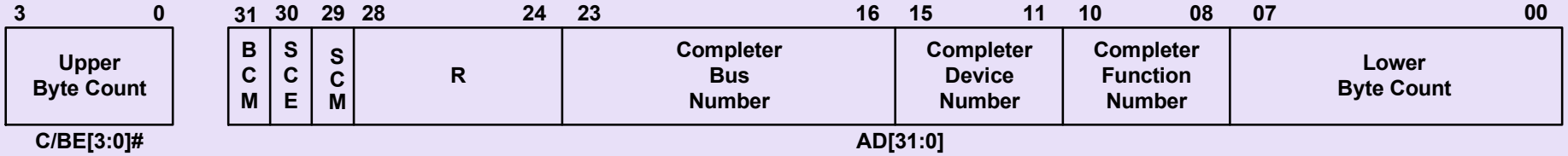
Transaction Attributes

Split Completion Address



RO -- Relaxed ordering

Completer Attributes



SCM -- Split Completion Message

SCE -- Split Completion Error

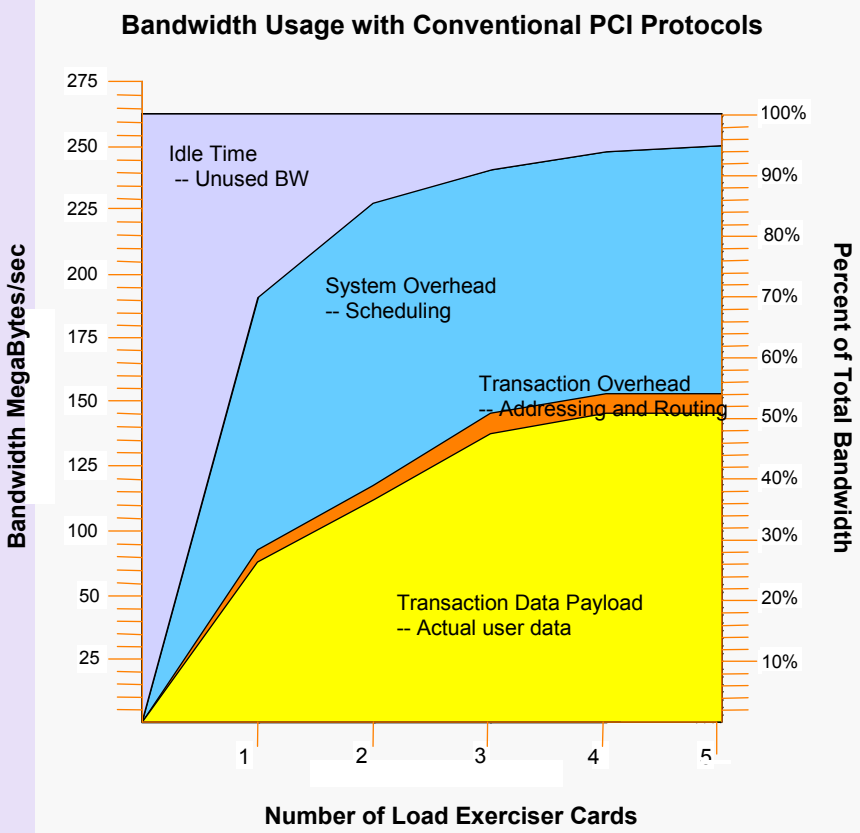
BCM -- Byte Count Modified

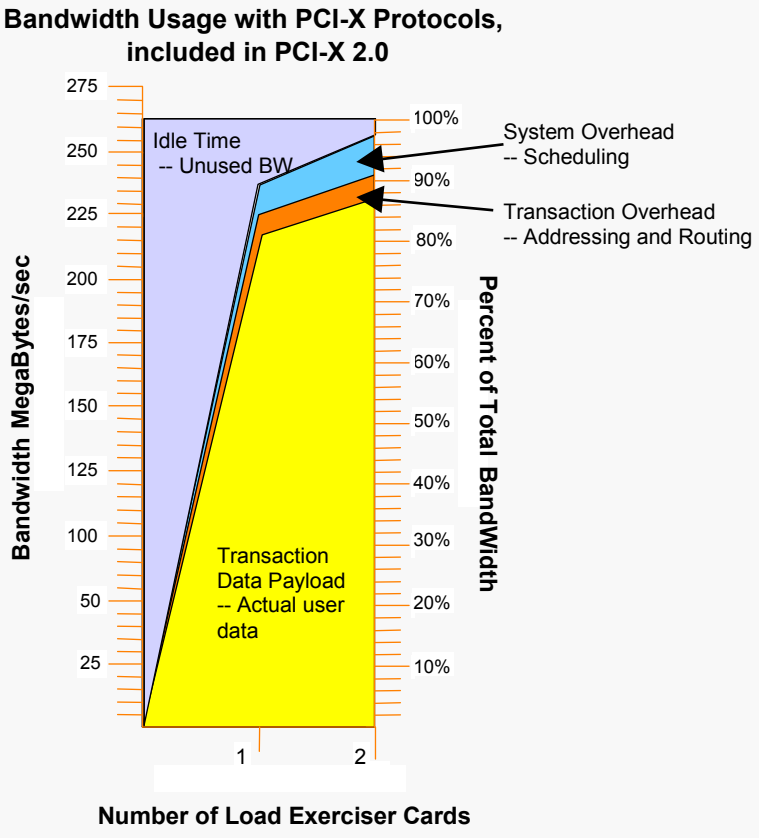
R -- Reserved

Split Transactions

- Bus efficiency of Read almost as good as Write
- Split Completion routed back to requester across bridges using initiator's number and bus number
- Split Transaction components
  - ✓ Step 1. Requester requests bus and arbiter grants bus
  - ✓ Step 2. Requester initiates transaction
  - ✓ Step 3. Target (completer) communicates intent with new target termination, Split Response
  - ✓ Step 4. Completer executes transaction internally
  - ✓ Step 5. Completer requests bus and arbiter grants bus
  - ✓ Step 6. Completer initiates Split Completion

Efficient PCI-X Protocol





The PCI-X protocol is more efficient than traditional PCI.

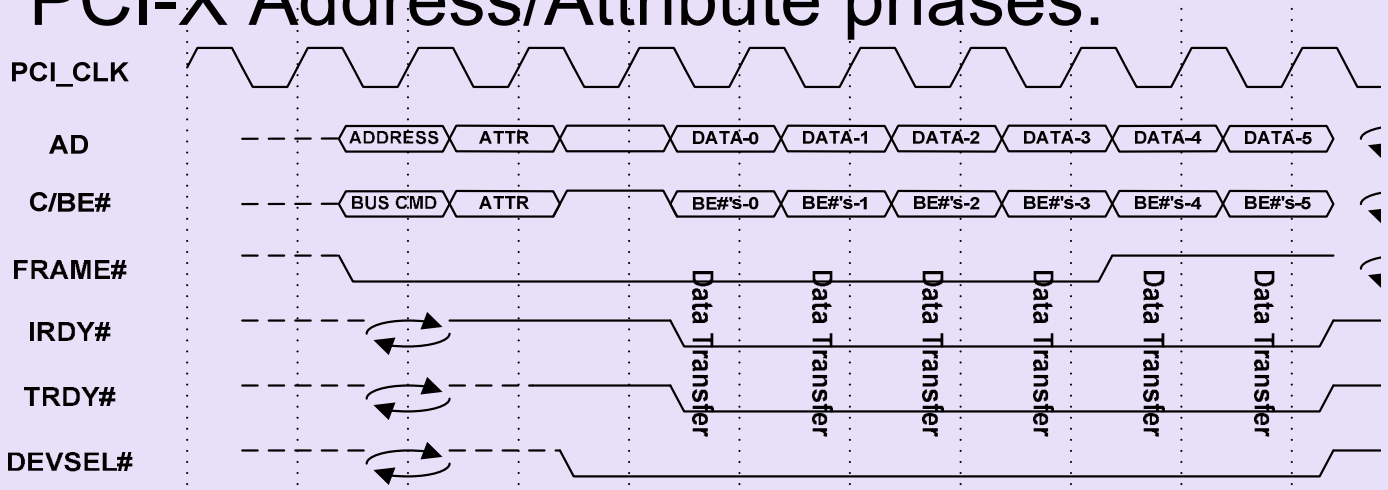
PCI Express Overview

PCIe Architecture Features

- **PCI Compatibility**
  - ✓ Configuration and PCI software driver model
  - ✓ PCI power management software compatible
- **Performance**
  - ✓ Scalable frequency (2.5-5GT/s)
  - ✓ Scalable width (x1, x4, x8, x16)
  - ✓ Low latency and highest utilization (BW/pin)
- **Physical Interface**
  - ✓ Point-to-point, dual-simplex
  - ✓ Differential low voltage signaling
  - ✓ Embedded clocking
  - ✓ Supports connectors, modules, cables
- **Protocol**
  - ✓ Fully packetized split-transaction
  - ✓ Credit-based flow control
  - ✓ Hierarchical topology support
  - ✓ Virtual channel mechanism
- **Advanced Capabilities**
  - ✓ CRC-based data integrity, hot plug, error logging
- **Enhanced Configuration Space**
  - ✓ Extensions and bridges into other architectures

PCIe Protocol Overview

- PCI-X Address/Attribute phases:

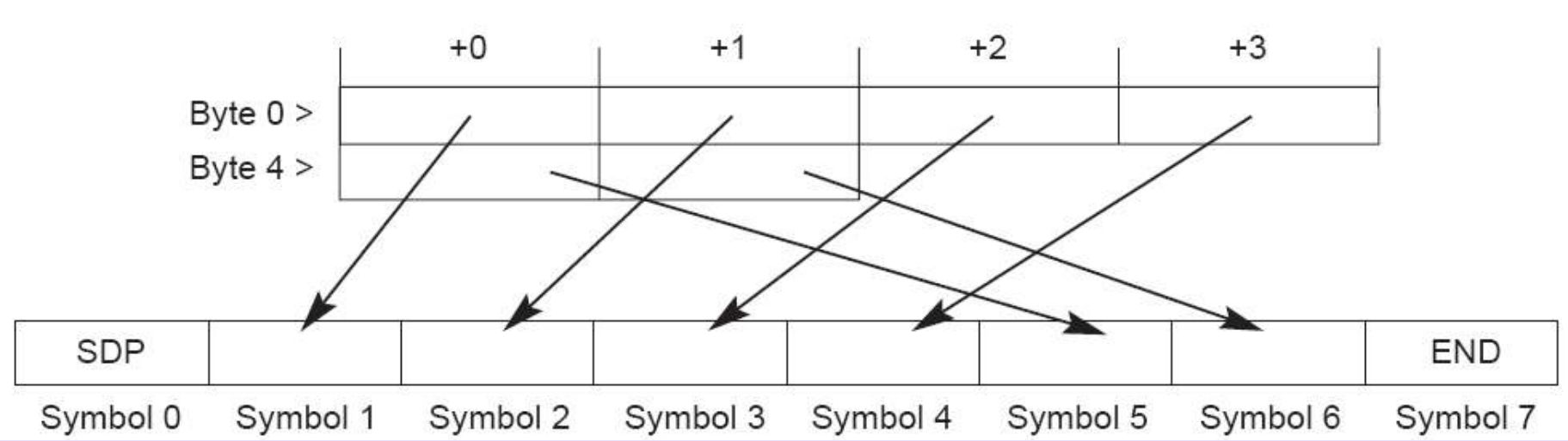


- Evolved into the PCIe Packet Header:

	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x 1			Type				R	TC			Reserved				TD	EP	Attr		R				Length							
Byte 4 >	Requester ID																Tag								Last DW BE				1st DW BE			
Byte 8 >	Address[63:32]																															
Byte 12 >	Address[31:2]																								R							

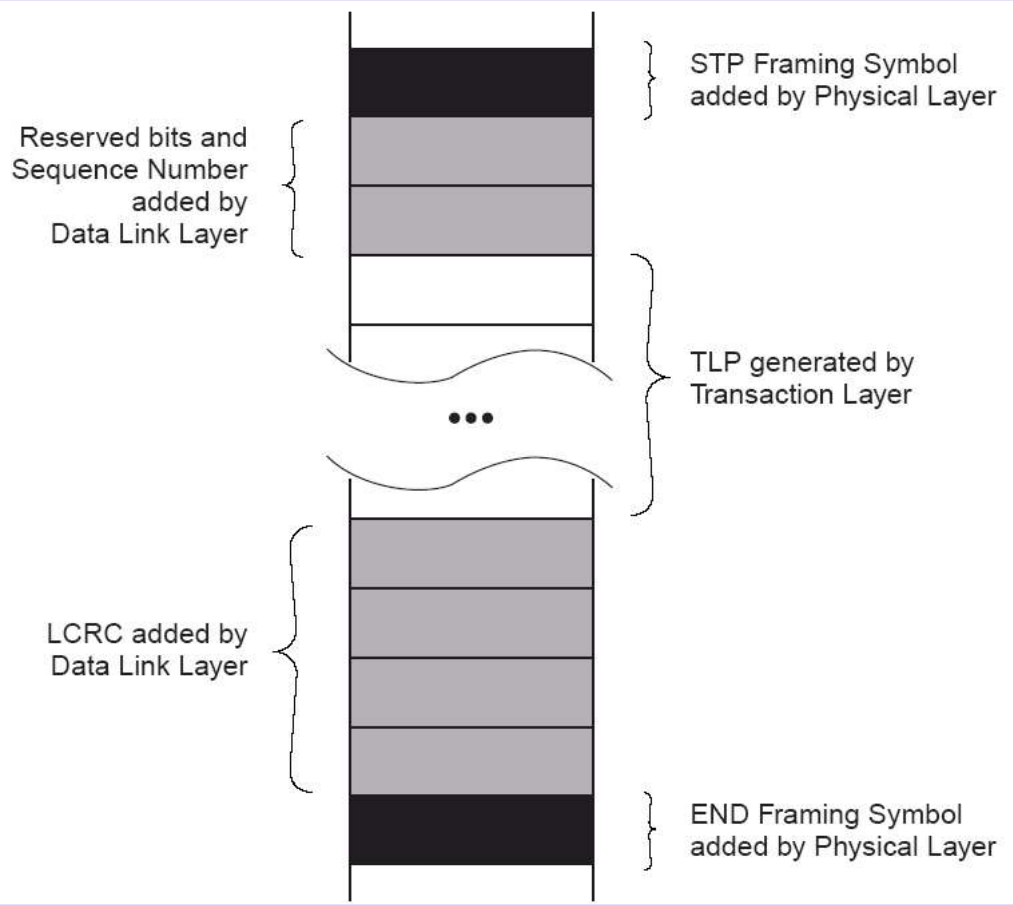
PCIe Protocol Overview

- The packet bytes get converted to 8b/10b and serialized



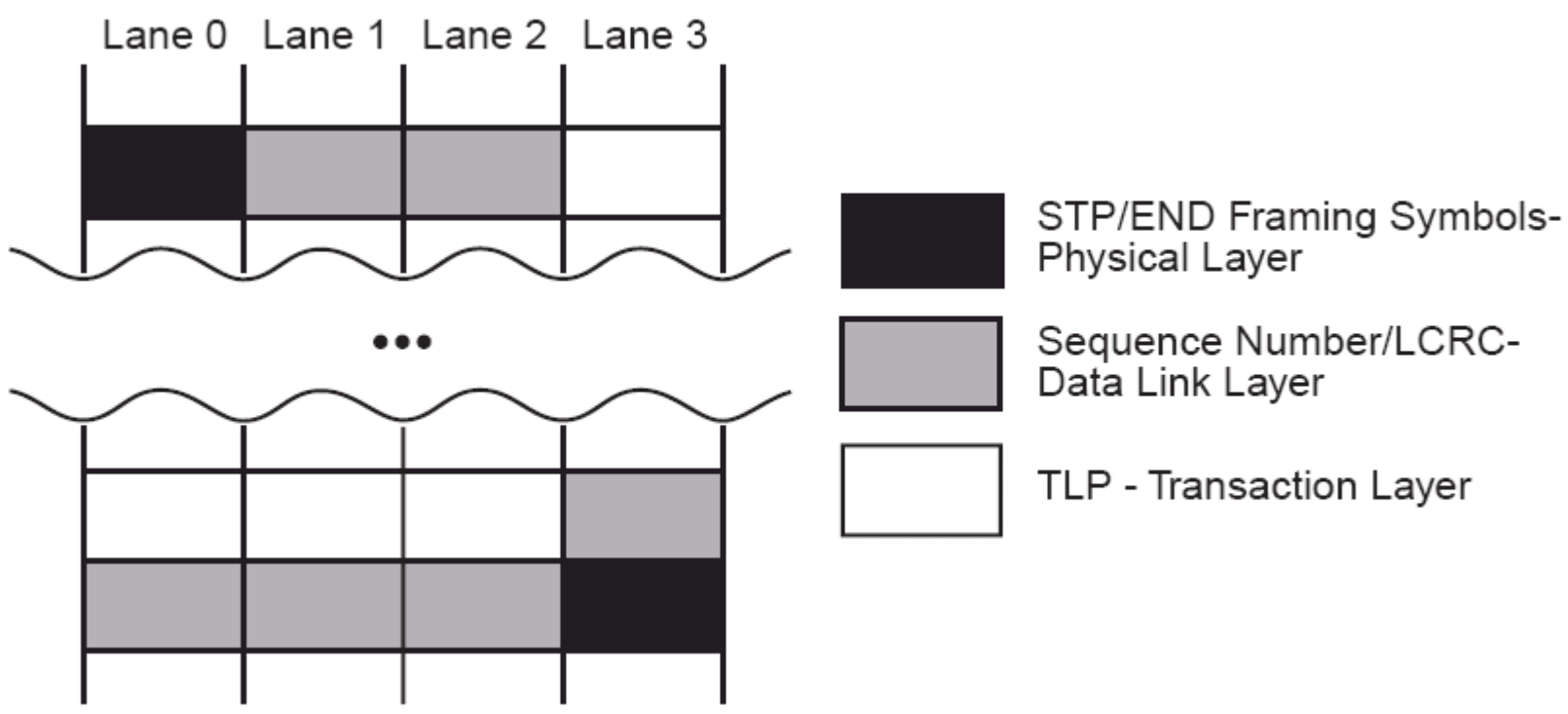
PCIe Protocol Overview

- Framing varies depending on link width
  - ✓ x1

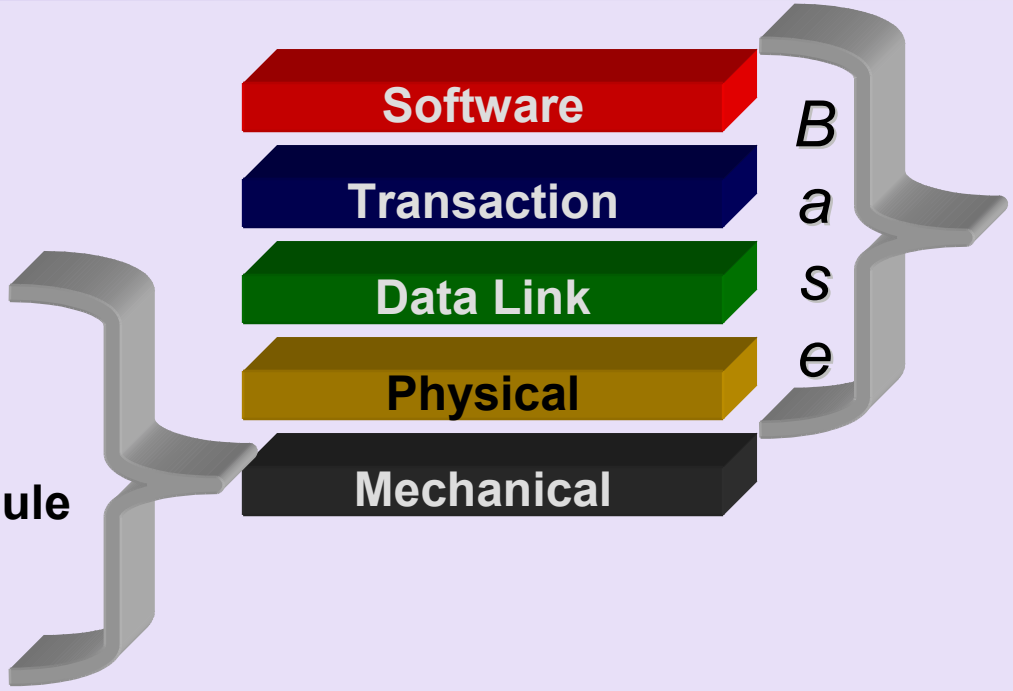


PCIe Protocol Overview

- Framing varies depending on link width
  - ✓ x4



PCIe Architecture Specifications



Form Factors:

- Card
- Mini Card
- ExpressModule
- Cable

- Base
  - ✓ Electrical
  - ✓ Protocol
  - ✓ Configuration
- Bridge
- *I/O Virtualization (WIP)*

- **Layered, scalable architecture**
- **Performance matched to applications**
- **Innovative form factors**

PCI Express 2.0

PCI Express 2.0

- Backwards compatible with PCI Express 1.1
- Speed bump
  - ✓ 2.5GT/s to 5GT/s
  - ✓ Logical extensions to the physical layer only
- Incorporates errata and ECN since PCIe 1.1
  - ✓ Function Level Reset
  - ✓ Access Control Services
  - ✓ Alternate Routing ID
  - ✓ Completion Timeout
  - ✓ Power budgeting
  - ✓ Trusted Configuration Space (removed)

Logical Extensions Summary

Extensions	Explanation	Benefits
Speed Negotiation	Capability to upgrade or downgrade link speed	RAS (improved link uptime), dynamic link speed optimization, power savings
Compliance Speed	Programmable as well as inband mechanism to select compliance pattern speed	Flexibility to perform compliance testing at multiple speeds with low cost
Electrical Idle Entry and Exit	Protocol changes to facilitate circuit design	Enhanced robustness, yield, power savings, ease of design (TTM)
Link width Upconfigure	Dynamic Link width change	Power savings
Testability Enhancements	Receiver Compliance Transmitter Margining Loopback Enhancements	Cost effective way to test the transmitter and receiver Assumption based loopback for test equipment.

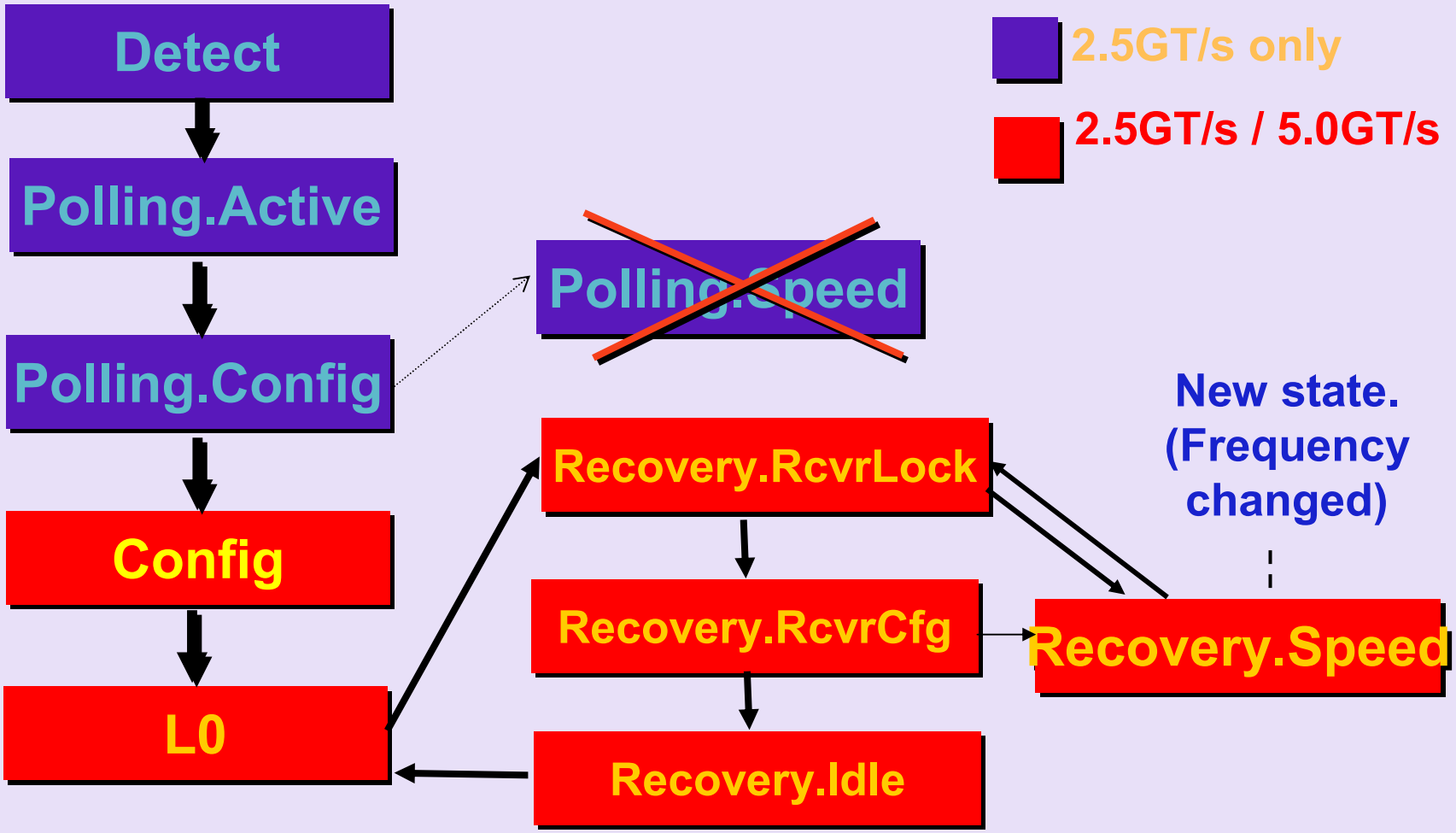
Speed Negotiation: Philosophy

- Dynamic link speed change
  - ✓ Link stays up
  - ✓ Reliability
    - 2.5 GT/s data rate as fall back speed
    - Revert back data rate if speed change unsuccessful
  - ✓ Power improvements
    - Lower frequency when demand drops
- Notify software on bandwidth change
  - ✓ Autonomous
  - ✓ Reliability / Software induced

Speed Negotiation

- Initially link trains to L0 in 2.5GT/s data rate
- Supported speeds advertised in TS ordered sets
  - ✓ Supported speeds by the other component noted in Config.Complete and Recovery.RcvrCfg
- Speed change handshake in Recovery
  - ✓ New substate: Recovery.Speed
  - ✓ Speed changed in Recovery.Speed
- Polling.Speed state obsolete

Relevant LTSSM States



Config Register Changes

- Added encoding for 5GT/s speed (Link Capabilities)
- New control field: Target Link Speed
  - ✓ Sets target & upper limit on link operational speed
  - ✓ Set only in upstream component
  - ✓ Speed change occurs by setting link retrain bit
  - ✓ Also sets speed for software initiated Compliance
- Hardware Autonomous Speed Control
  - ✓ Controls ability of hardware to reduce speed dynamically
  - ✓ Does not affect speed reduced for reliability reasons
- Link Bandwidth change notification mechanism
  - ✓ Link B/W Management Status
  - ✓ Link Autonomous B/W Status
- Software initiated Compliance
  - ✓ Enter Compliance and Enter Modified Compliance

Speed of Polling.Compliance

- Two ways to set the speed
  - ✓ Inband Method: Compliance Load Board (Modified)
  - ✓ CSR method (new method)
    - – Link Control2 Register Fields
      - Target Link Speed, Enter Compliance, Enter Modified Compliance, Compliance De-emphasis, and Transmit Margin
    - – Procedure:
      - Link trains to L0
      - CSR bits written on both sides to enter compliance
      - SBRE upstream.
      - LTSSM:Hot Reset -> Detect -> Polling.Active -> Polling.Compliance
- Speed, de-emphasis, transmit margin determined prior to entering Polling.Compliance

Electrical Idle Exit Challenges

- Exit from Electrical Idle a challenge in 5.0GT/s speed
  - ✓ Receiver sensitivity in 5GT/s speed: 120 mV
  - ✓ Idle detection threshold in 5GT/s speed: 175 mV
- Expect exit EI to be detected by circuits in 2.5GT/s speed
- Need a low-frequency recurring pattern for 5.0GT/s:
  - ✓ Electrical Idle Exit Sequence (EIES) defined
    - – Effectively 5 1's followed by 5 0's 13 times => low frequency
  - ✓ Requirements relaxed to detect exit EI in *non-2.5GT/s* speeds
  - ✓ One EIES sent after every 32 TS1/TS2'es
  - ✓ Sent in Recovery and Config.Linkwidth.Start
  - ✓ Mix of low frequency (EIES) for EI exit detection and high frequency (TS1) for symbol lock.
- L0s Exit in 5GT/s speed
  - ✓ Send 4 to 8 K28.7 symbols to help with EI exit detection

Electrical Idle Entry

- EI entry detection difficult. Required in:
  - ✓ L0, Loopback.Active, and Recovery
- Have the logical layer do the *inference* to alleviate circuit design
- New *infer* EI as an alternative to detecting EI
  - ✓ L0: no SKP Ordered Sets or UpdateFC in 128us.
  - ✓ Recovery.RcvrCfg and Recovery.Speed on successful negotiation:
    - no TS ordered sets in any configured lane in 1280 UI interval
  - ✓ Recovery.Speed with unsuccessful negotiation:
    - absence of exit from EI
      - 16000 UI in 5GT/s
      - 2000 UI in 2.5GT/s
  - ✓ Loopback Active
    - 2.5GT/s: no exit EI in 128 usec
    - 5GT/s: rely on EIOS only

Link Width Upconfigure

- New capability to upconfigure link width
- Power savings during low demand
- Up to the width initially negotiated
  - ✓ E.g.: x8 -> x4 -> x1 -> x2 -> x8
- Capability advertised in the TS2 ordered sets in Config.Complete state (Symbol 4, Bit 6 of TS2)
- Link width upconfigure and planned downconfigure:
  - ✓ L0 -> Recovery.RcvrLock -> Recovery.RcvrCfg -> Recovery.Idle -> Config -> L0
- Link width is changed in Config State
  - ✓ Initiator of upgrade delays sending link number till the other side wakes up on those lanes and sends TS ordered sets

Testability Enhancements

- Receiver Compliance:
  - ✓ A new mode in Polling.Compliance
  - ✓ Receiver sends out modified compliance patterns includes receiver error count
  - ✓ Intended to margin receiver
  - ✓ Two mechanisms: inband and CSR
- Transmitter Margining:
  - ✓ Multiple voltage levels for Tx; CSR setting
- Loopback:
  - ✓ Speed change (de-emphasis) in Config -> LB
    - assumption-based training
  - ✓ Ability to loopback even without achieving lock

Enhancements for Robustness

- Changes to Polling.Compliance for High Availability (HA)
  - ✓ Entry condition relaxed from any lane that detected a receiver but did not get an exit from Electrical Idle to multiple lanes
  - ✓ Must go to Polling.Compliance if all lanes that detected a receiver did not get an exit from electrical idle
- Electrical Idle Ordered Set extension for >2.5GT/s speeds:
  - ✓ Two consecutive sets of COM, IDL, IDL, IDL
- Electrical Idle detection (optional):
  - ✓ Received signals switching at a frequency greater than 125MHz

Other PCIe 2.0 Changes

Feature	Explanation	Benefit
Function Level Reset	Ability to reset registers specific to a function in endpoint	Enables software to reset endpoint hardware with function-level granularity. Virtualization friendly
Access Control Services	Control peer-to-peer traffic. Applicable to RCs, switches and multi-function devices	Control which components can talk to one another
Alternative Routing ID	5 bit device number + 3 bit function number becomes new 8 bit function number	Increase number of functions allowed per multi-function device
Completion Timeout Control Capability	Allow system software to set completion timeout value or disable mechanism	Reduce likelihood of false triggering
Power Budgeting	Adds 250W, 275W and 300W power limits	Enable future higher power future PCIe products

Acknowledgements

Special thanks to:

Richard Solomon

Debendra Das Sharma

Wesley Shao

Marc Wells

Thank you for attending the PCIe Technology Seminar

For more information please go to www.pcisig.com