

PCI



SIG[®]



PCI-SIG® Architecture Overview

**Richard Solomon
IC Design Engineer
LSI Logic**



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, PCI Express
- Day 1 will progress through:
 - ✓ PCI Express Protocols...
 - ✓ ...PCI Express 2.0 5GT/sec Electricals...
 - ✓ ...and will close with exciting new work happening in the area of I/O Virtualization
- Day 2 devoted to members like you sharing their experiences implementing PCI-SIG Technologies

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**
 - ✓ ***PCI Express ECN adds “Trusted Configuration Space”***
 - *Just a fourth address space from a bus perspective*
 - *Enables system trust mechanisms*
- **Key configuration space regs/concepts**
 - ✓ **Base Address Registers (BARs)**
 - 64-bit vs 32-bit addressing
 - ✓ **Linked list of capabilities**

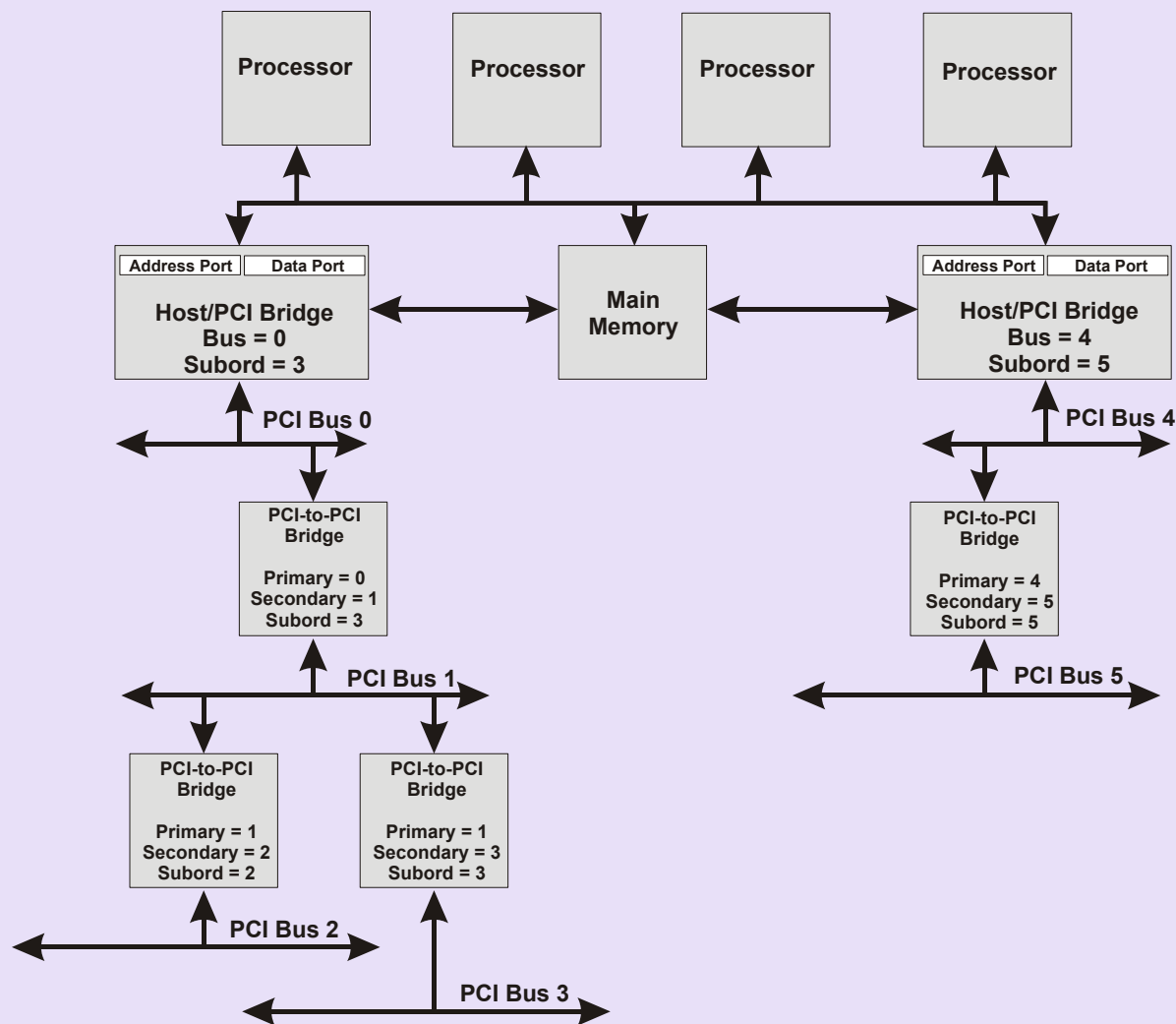
Address spaces – Memory & I/O

- **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)



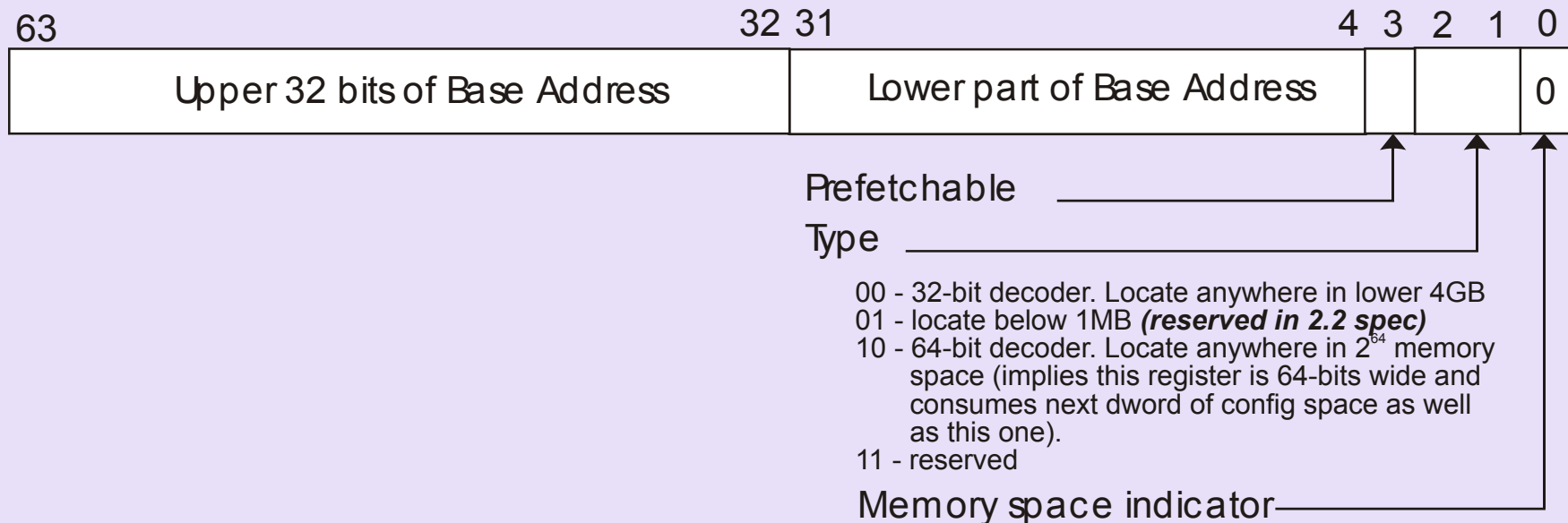
Using Configuration Space

- **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**

Byte				Doubleword Number (in decimal)
3	2	1	0	
Device ID		Vendor ID		00
Status Register		Command Register		01
Class Code			Revision ID	02
BIST	Header Type	Latency Timer	Cache Line Size	03
Base Address 0				04
Base Address 1				05
Base Address 2				06
Base Address 3				07
Base Address 4				08
Base Address 5				09
CardBus CIS Pointer				10
Subsystem ID		Subsystem Vendor ID		11
Expansion ROM Base Address				12
Reserved			Capabilities Pointer	13
Reserved				14
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	15

Using Configuration Space

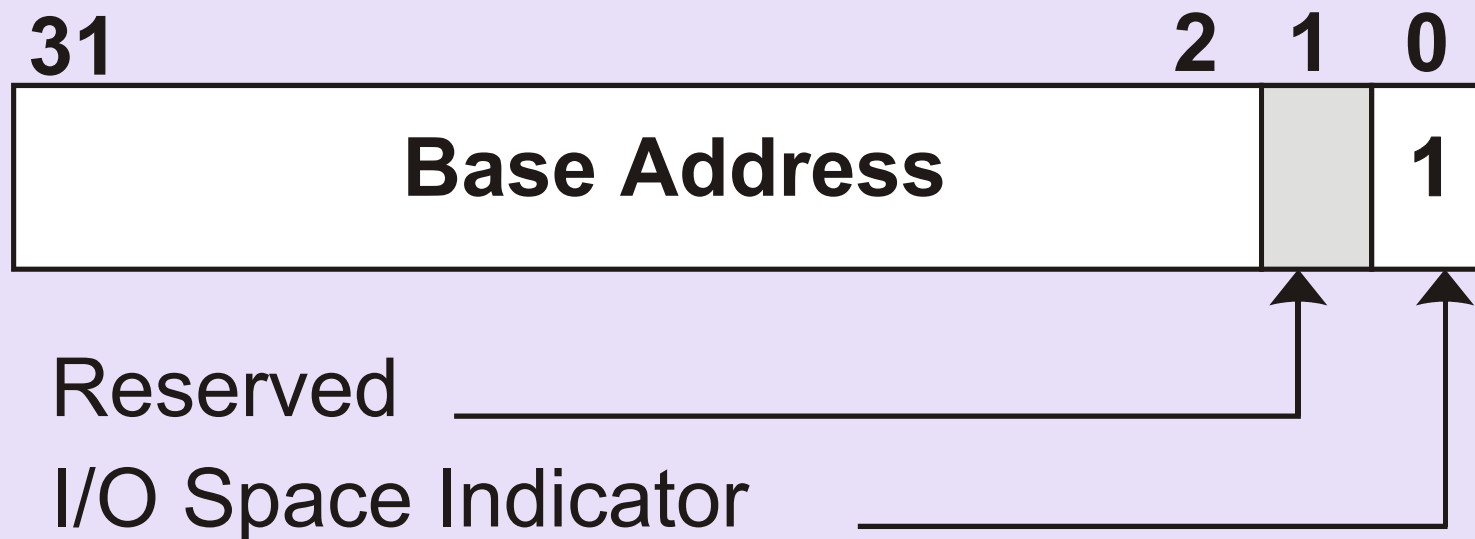
■ Memory BARs



- Note upper DWORD not present when 64-bit decode is not set

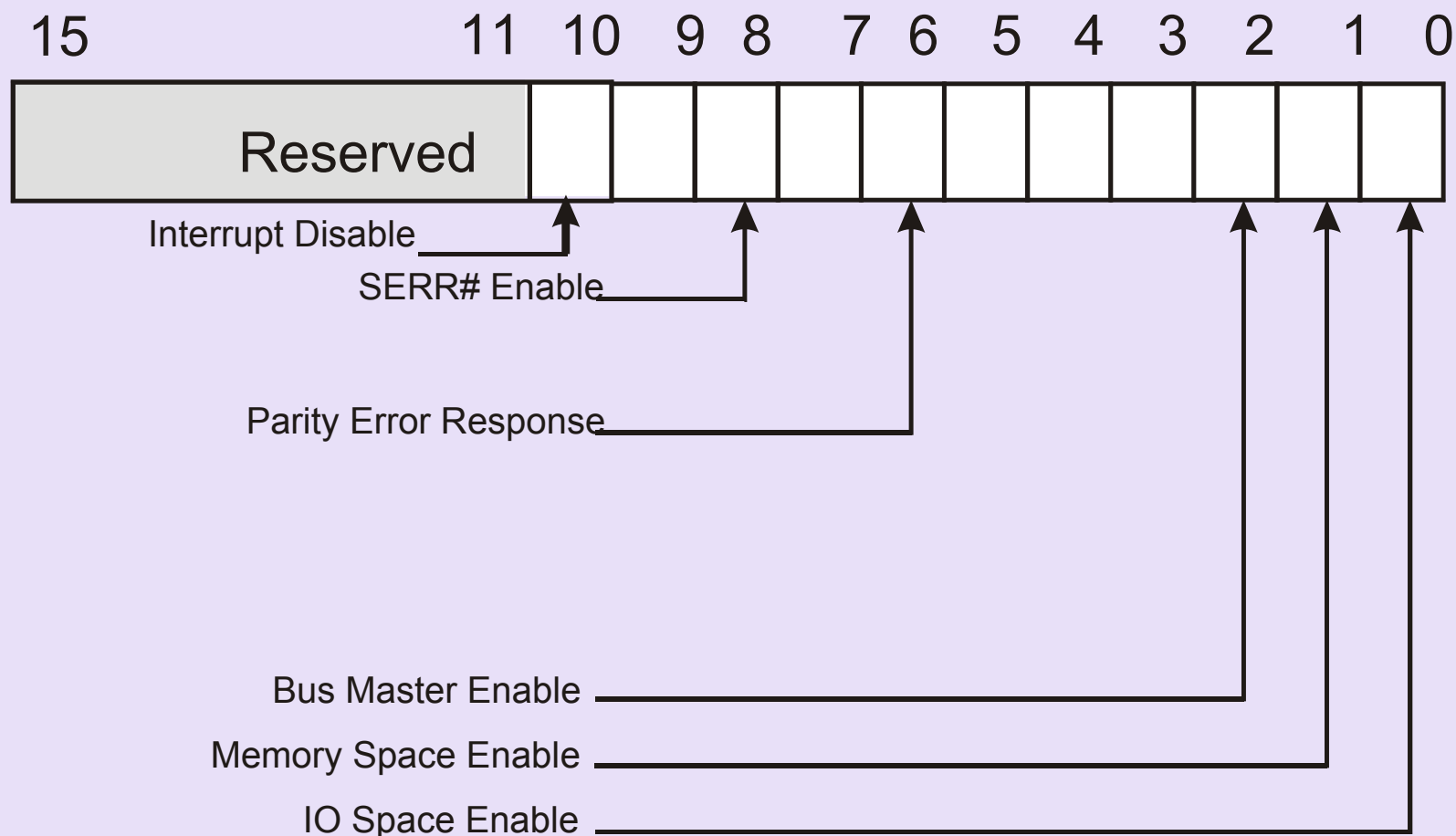
Using Configuration Space

- I/O BARs look similar to Memory
 - ✓ Bit 0 is “1” to indicate I/O
 - ✓ No upper DWORD
 - ✓ No other encoded bits



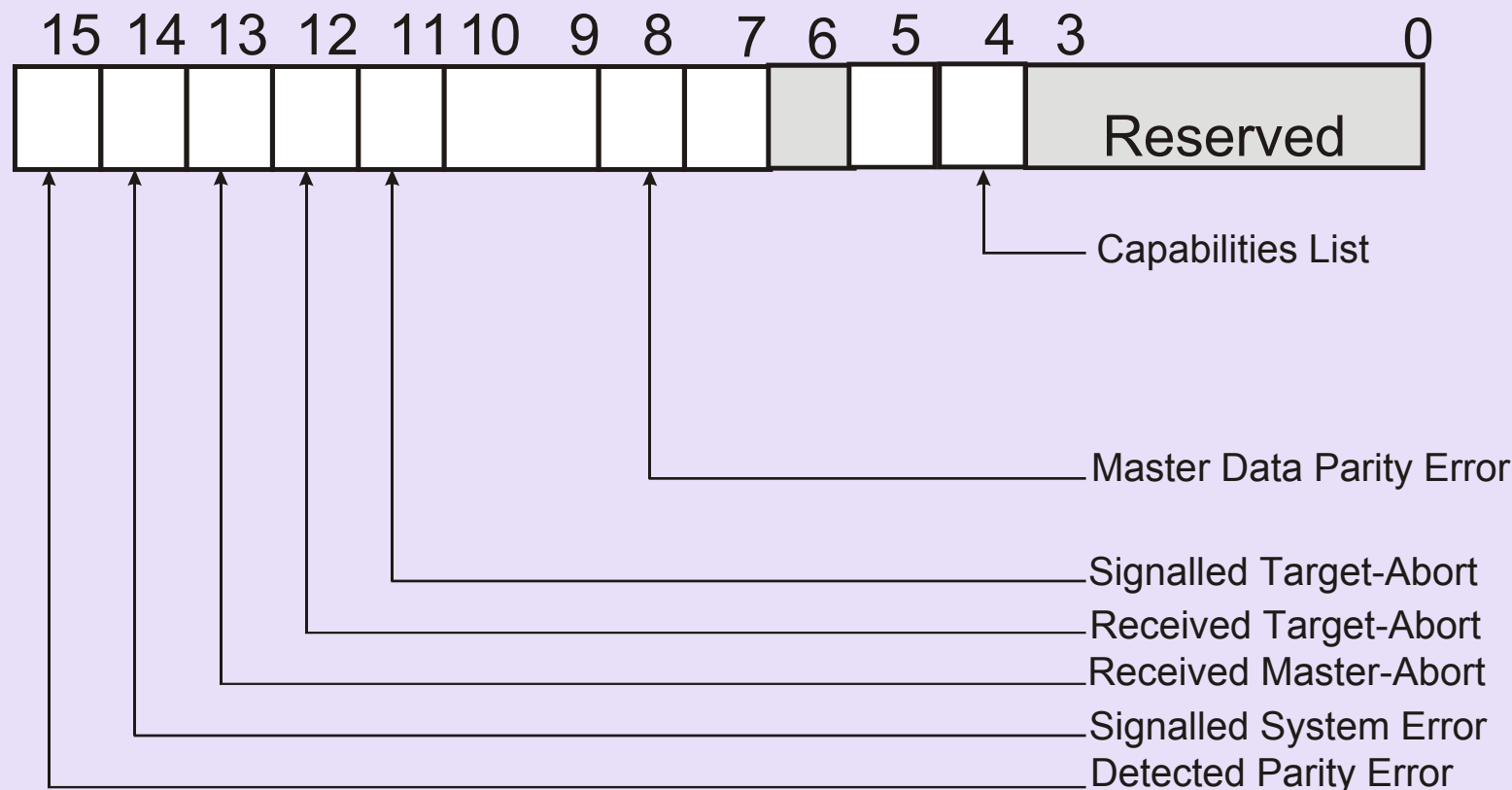
Using Configuration Space

■ Command Register (common fields)

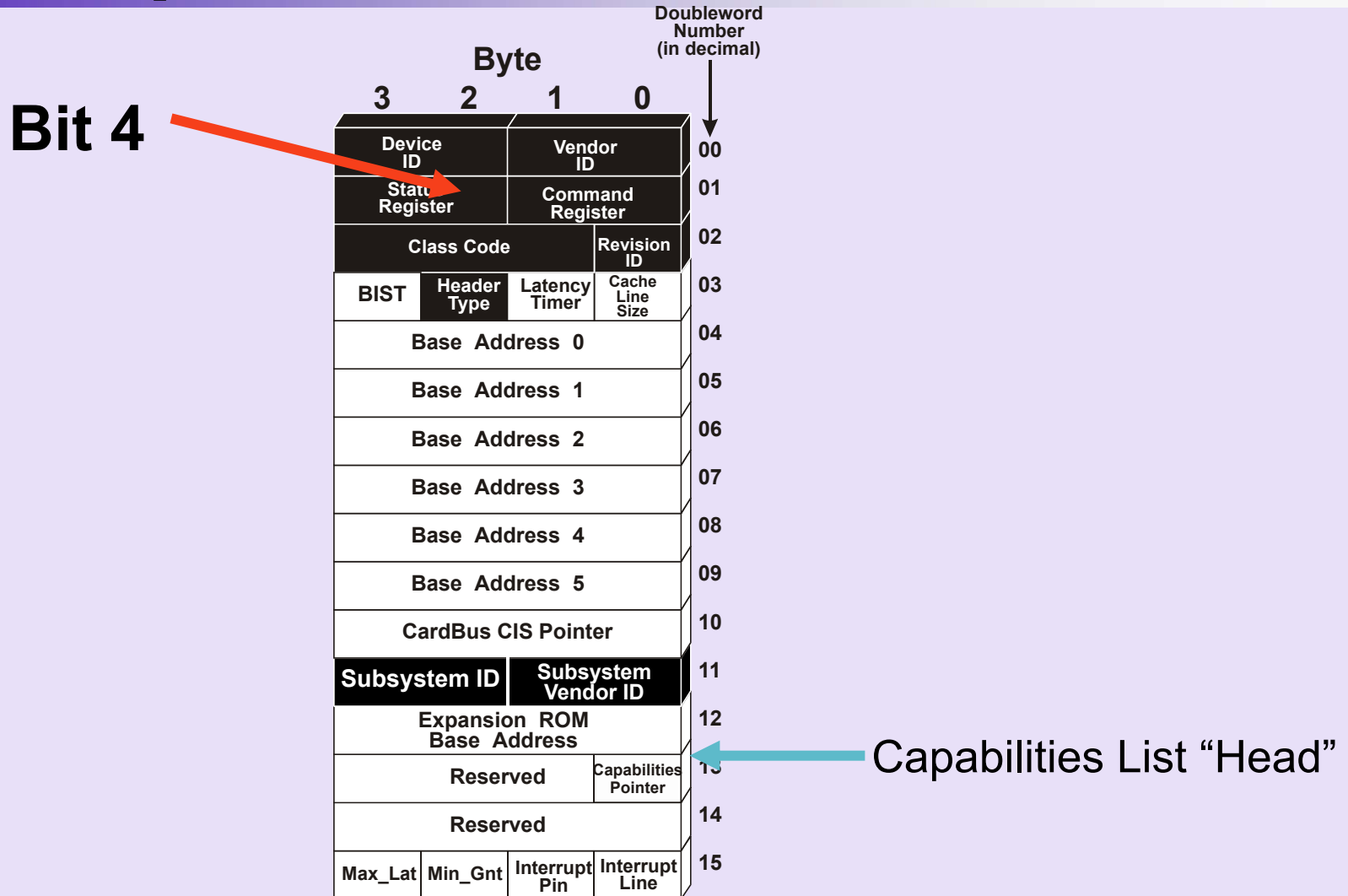


Using Configuration Space

■ Status Register (common fields)



Using Configuration Space – Capabilities List



Using Configuration Space – Capabilities List (cont'd)

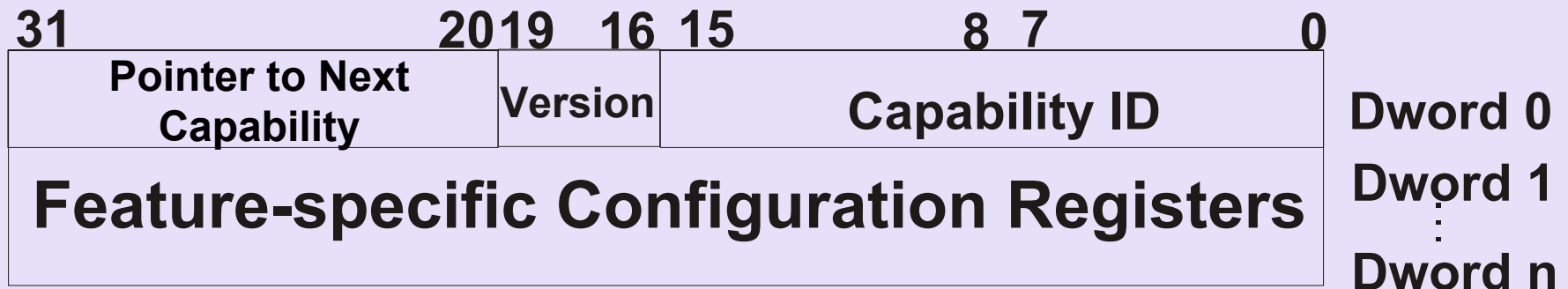
■ Linked list

- ✓ 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



What is “Trusted” Configuration Space?

- **Trusted Configuration Space (TCS) is an ECN for the PCIe 1.1 Base spec and will be included in PCIe 2.0 Base spec – does NOT apply to PCI or PCI-X**
- **TCS is a new PCIe address space**
- **TCS introduces 2 new PCIe Requests**
 - ✓ **Trusted Config Read & Trusted Config Write**
 - ✓ **System will control generation of these requests through means outside the PCIe spec**
- **TCS is optional for Endpoints, Switches, & Root Ports**
 - ✓ **Newly designed switches should include routing**

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**

MSI & MSI-X Explained



MSI & MSI-X Apply to ALL PCI-SIG Specifications

- **Implementation of MSI *or* MSI-X is mandatory in both PCI Express and PCI-X**
 - ✓ **Note recent ECNs allowing MSI-X instead of MSI**
- **Implementation of either MSI or MSI-X is optional in Conventional PCI**
- **Subsequent slides apply to any bus implementation of MSI and MSI-X**
 - ✓ **Same structures in PCI, PCI-X, and PCI Express**



Once Enabled, MSI or MSI-X Messages Replace INTx

- 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 and MSI-X Explained

■ MSI

- ✓ **Message Signaled Interrupts (MSI) is an optional feature that enables a device function to request service by writing a system-specified data value to a system-specified address (using a PCI DWORD memory write transaction).**
- ✓ **System software initializes the message address and message data (referred to as the “vector”) during device configuration, allocating one or more vectors to each MSI-capable function.**

MSI and MSI-X Explained

■ MSI-X

- ✓ **MSI-X defines a separate optional extension to basic MSI functionality.**
- ✓ **Many of the characteristics of MSI-X are identical to those of MSI.**
- ✓ **MSI-X additional capabilities include,**
 - **a larger maximum number of vectors per function**
 - **the ability for software to control aliasing, when fewer vectors are allocated than requested**
 - **the ability for each vector to use an independent address and data value, specified by a table that resides in Memory Space.**

MSI and MSI-X Explained

■ Per-vector masking

- ✓ Per-vector masking is managed through a *Mask* and *Pending* bit pair per MSI vector or MSI-X Table entry.
- ✓ An MSI vector is masked when its associated Mask bit is set.
- ✓ An MSI-X vector is masked when its associated MSI-X Table entry Mask bit or the MSI-X Function Mask bit is set.
- ✓ While a vector is masked,
 - the function is prohibited from sending the associated message,
 - and the function must set the associated Pending bit whenever the function would otherwise send the message.

MSI and MSI-X Explained

■ MSI-X ECN

- ✓ A function is permitted to implement both MSI and MSI-X, but system software is prohibited from enabling both at the same time.
- ✓ For the sake of software backward compatibility, MSI and MSI-X use separate and independent capability structures.
- ✓ On functions that support both MSI and MSI-X, system software that supports only MSI can still enable and use MSI without any modification.

MSI and MSI-X Explained

■ MSI Capability Structure

Capability Structure for 64-bit Message Address and Per-vector Masking

Message Control	Next Pointer	Capability ID	Capability Pointer
Message Address			Capability Pointer + 04h
<i>Message Upper Address (optional)</i>			Capability Pointer + 08h
Reserved	Message Data		Capability Pointer + 0Ch
<i>Mask Bits (optional)</i>			Capability Pointer + 10h
<i>Pending Bits (optional)</i>			Capability Pointer + 14h

MSI and MSI-X Explained

■ MSI-X Capability and Table Structures

31	16	15	8	7	0		
Message Control			Next Pointer		Capability ID		CP +00h
Table Offset					Table BIR		CP +00h
PBA Offset					PBA BIR		CP +00h

MSI-X Capability Structure

- ✓ Different from MSI, the MSI-X capability structure points to an MSI-X Table Structure and a MSI-X Pending Bit Array (PBA) structure, each residing in Memory Space.
- ✓ Each structure is mapped by a Base Address register (BAR) belonging to the function. A BAR Indicator register (BIR) indicates which BAR, and maps Memory space.

MSI and MSI-X Explained

■ MSI-X Capability and Table Structures

DWORD3	DWORD2	DWORD1	DWORD0		
Vector Ctrl	Msg Data	Msg Upper Addr	Msg Address	entry 0	Base
Vector Ctrl	Msg Data	Msg Upper Addr	Msg Address	entry 1	Base +1*16
Vector Ctrl	Msg Data	Msg Upper Addr	Msg Address	entry 2	Base +2*16
....
Vector Ctrl	Msg Data	Msg Upper Addr	Msg Address	entry (N-1)	Base +(N-1)*16

MSI-X Table Structure

MSI and MSI-X Explained

■ MSI-X Capability and Table Structures

63 62 61 2 1 0

Pending Bits 0 through 63	QWORD0 Base
Pending Bits 64 through 127	QWORD1 Base + 1*8
....	QWORD0 Base
Pending Bits $((N-1) \div 64) * 64$ through N-1	QWORD $((N-1) \div 64)$ Base + $((N-1) \div 64) * 8$

MSI-X PBA Structure

MSI and MSI-X Explained

- **Enabling and Sending Message Interrupts**
 - ✓ **Both MSI and MSI-X are disabled following reset.**
 - ✓ **System configuration software sets either the MSI Enable bit or the MSI-X Enable bit to enable either MSI or MSI-X, but never both simultaneously.**
 - ✓ **Once MSI or MSI-X is enabled, and one or more vectors is unmasked, the function is permitted to send messages.**
 - ✓ **To send a message, a function does a DWORD memory write to the appropriate message address with the appropriate message data.**

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

What is PCI-X 2.0?

- Revision 2.0 includes everything from Revision 1.0
 - ✓ PCI-X 66
 - Same clock speed as fastest conventional PCI
 - Easier timing and more efficient bus utilization
 - ✓ PCI-X 133
 - Twice as fast as conventional PCI
 - Easier timing and more efficient bus utilization
- Revision 2.0 introduces 2 new speed grades
 - ✓ PCI-X 266
 - “Double data rate” clocking for transfer rates up to 266MHz
 - Up to 2.13 Gigabytes per second of bandwidth
 - ✓ PCI-X 533
 - “Quad data rate” clock for transfer rates of up to 533MHz
 - Up to 4.26 Gigabytes per second of bandwidth
- All PCI-X devices are fully backward-compatible to:
 - ✓ 33 MHz conventional PCI (66 MHz support is optional)
 - ✓ All lower PCI-X speeds

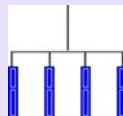
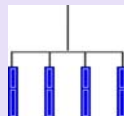
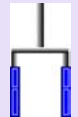
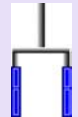
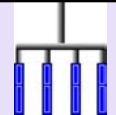
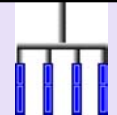
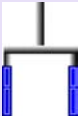
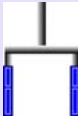






PCI-X Modes and Speeds



Mode 1

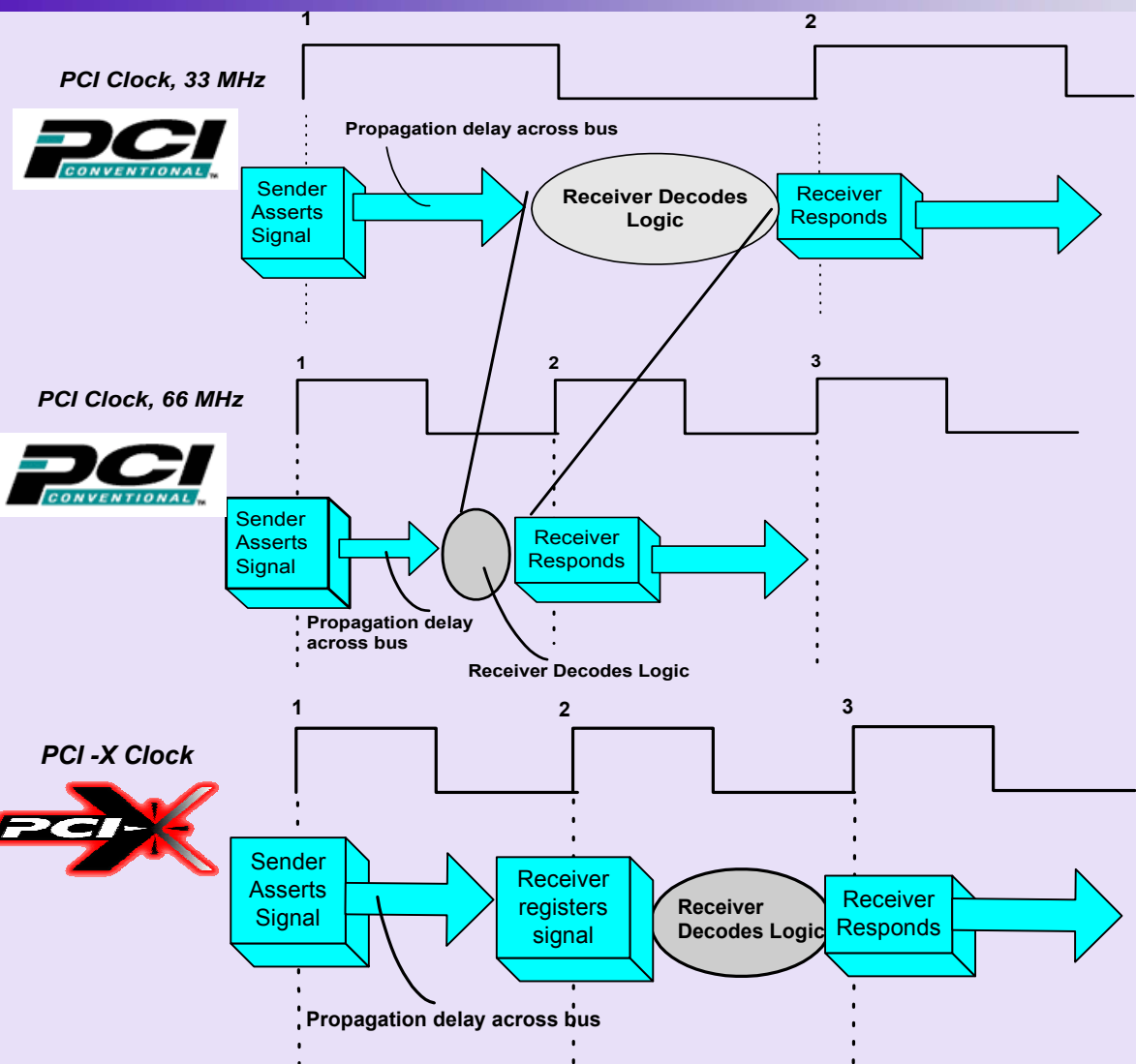


Mode 2

Mode	V _{I/O}	64-Bit		32-Bit		16-Bit	Error Prot	Conf Bytes	DIM
		Slots*	MB/s	Slots*	MB/s				
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

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

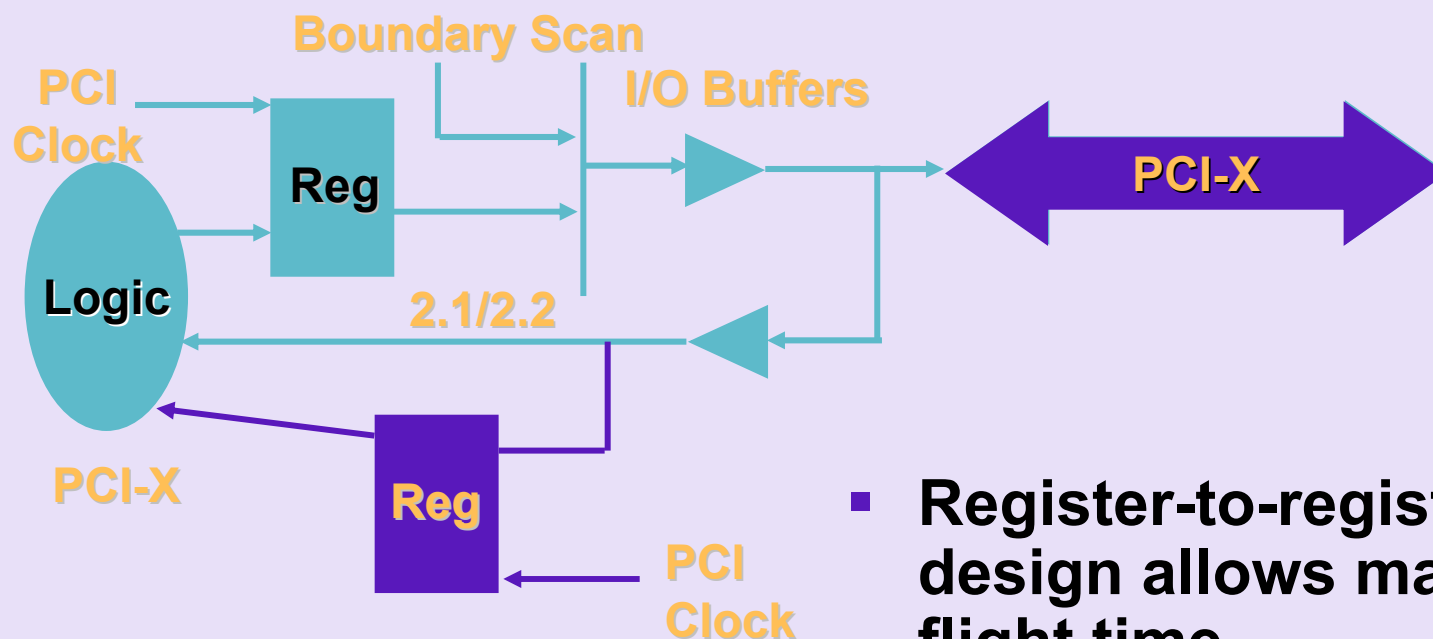
Registered Bus Protocol



- 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

Registered Bus Protocol

PCI-X protocol always takes 2 clocks to “turn around” a control event

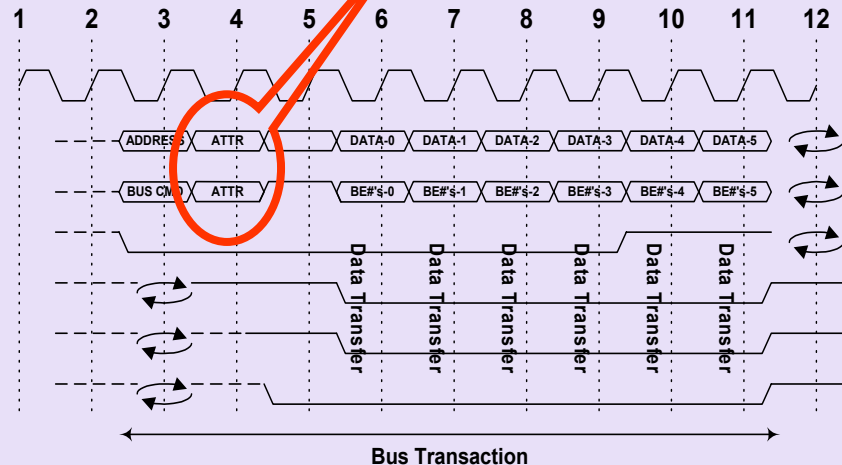
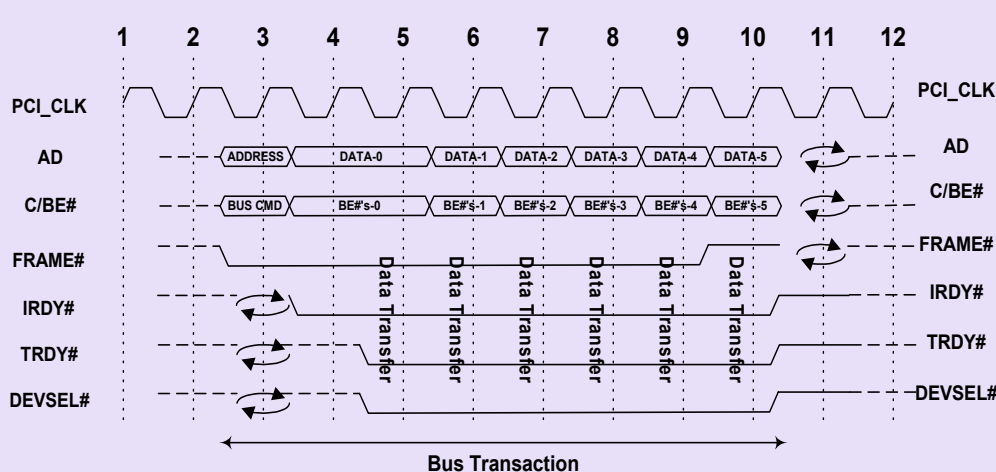


- Register-to-register design allows maximum flight time

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

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

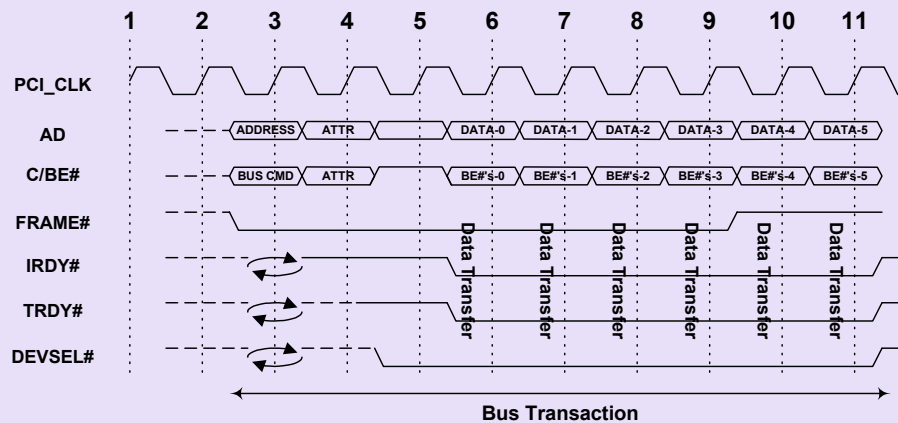
New “Attribute” phase for enhanced features



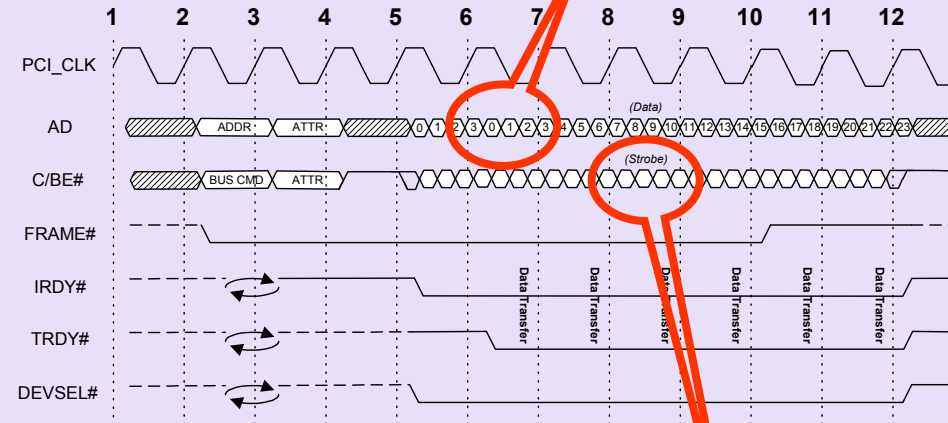


PCI-X 66/133 (Mode 1) vs. PCI-X 266/533 (Mode 2)

- 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)



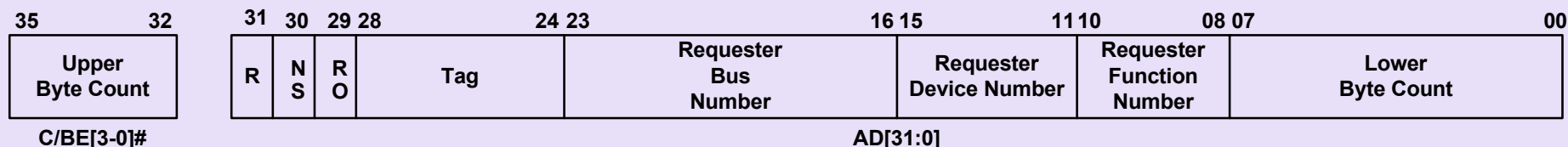
PCI-X 533 (Mode 2)

4 transfers per
clock cycle

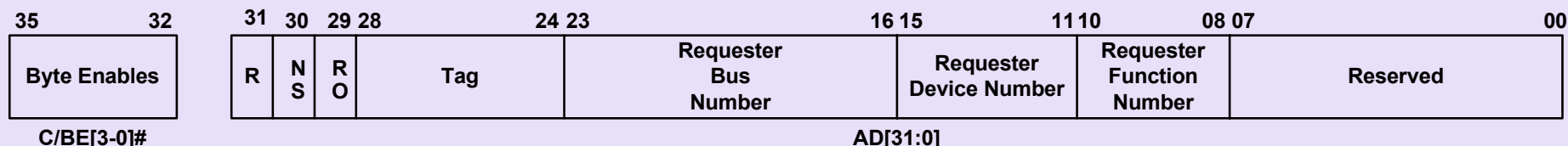
source-
synchronous
data strobes
share C/BE pins

Transaction Attributes

Requester Attributes for Burst Transactions



Requester Attributes for DWORD Transactions



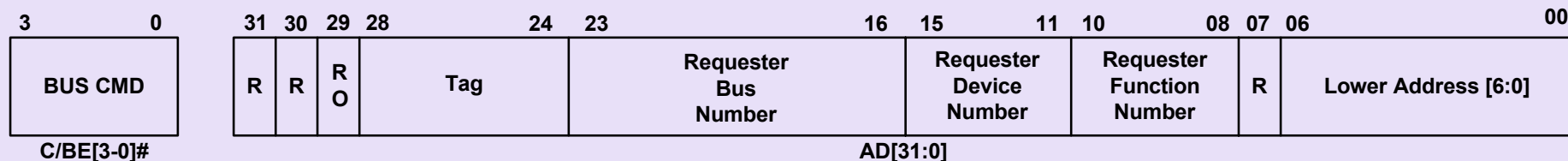
RO -- Relax ordering

NS -- No Snoop

R -- Reserved

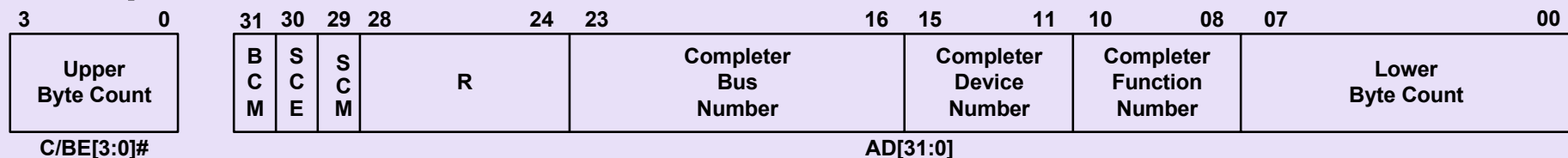
Transaction Attributes

Split Completion Address



RO -- Relaxed ordering

Completer Attributes



SCM -- Split Completion Message

SCE -- Split Completion Error

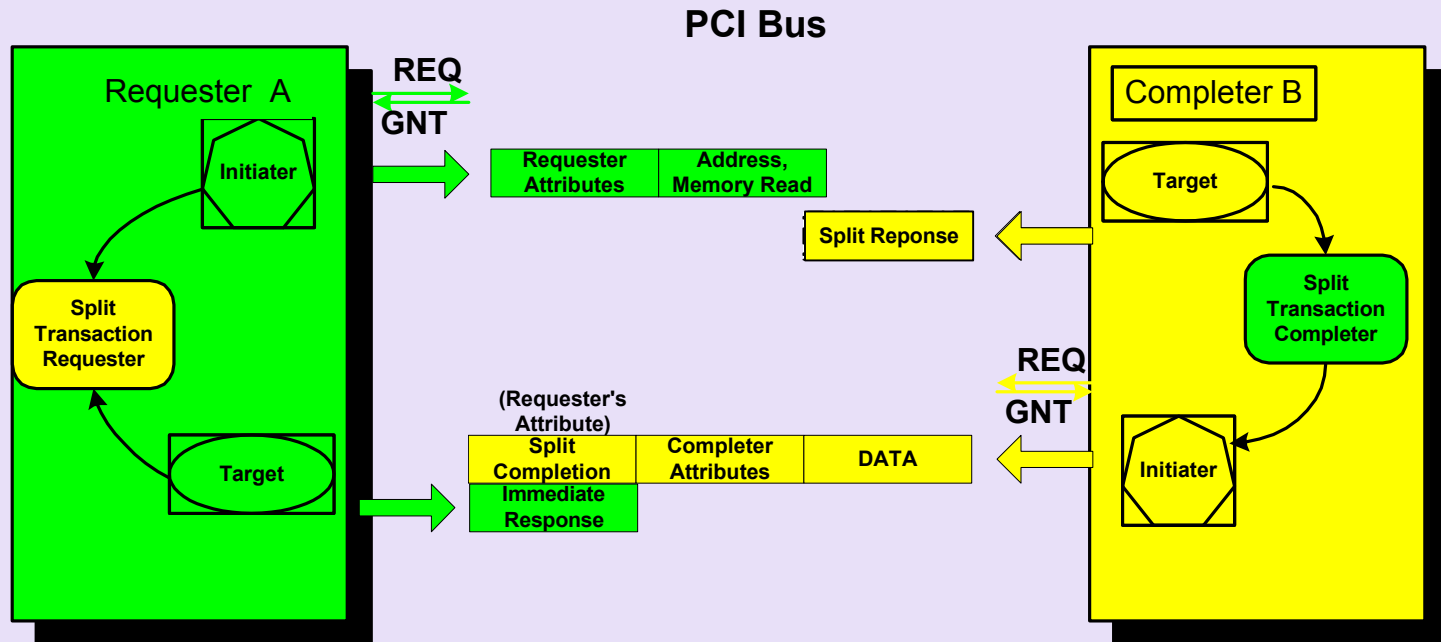
BCM -- Byte Count Modified

R -- Reserved

Split Transactions

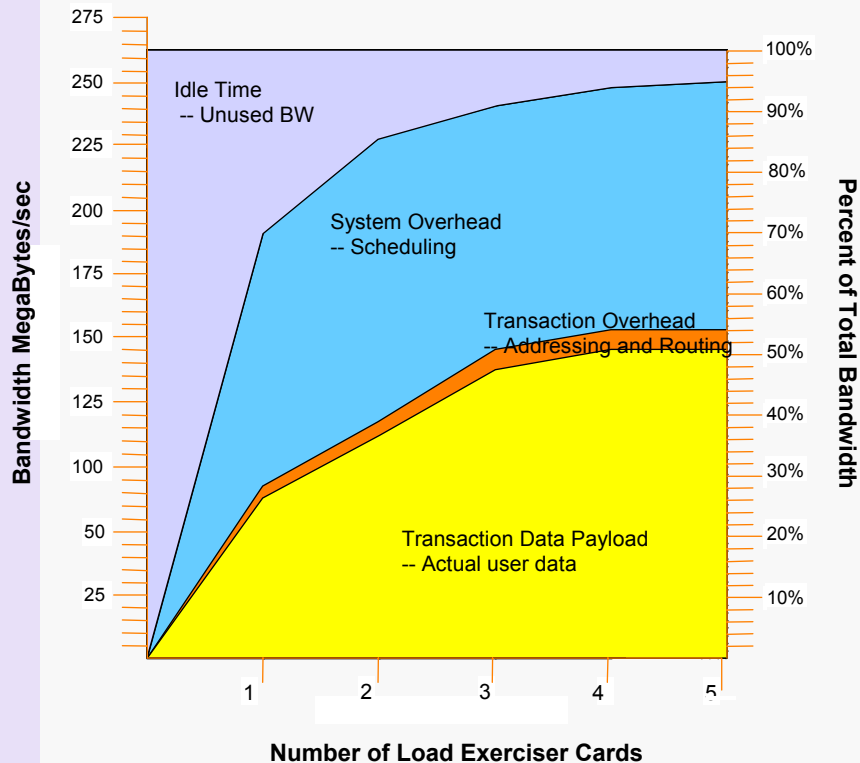
- Bus efficiency of Read almost as good as Write
- Split Completion routed back to requester across bridges using initiator's number and bus number
- Split Transaction components
 - ✓ Step 1. Requester requests bus and arbiter grants bus
 - ✓ Step 2. Requester initiates transaction
 - ✓ Step 3. Target (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

Split Transactions

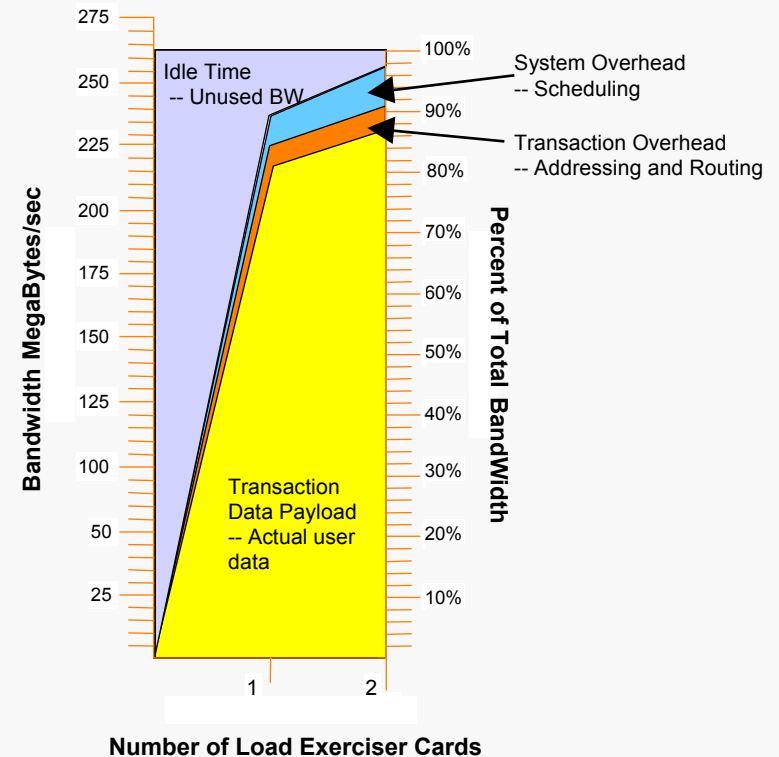


Efficient PCI-X Protocol

Bandwidth Usage with Conventional PCI Protocols



Bandwidth Usage with PCI-X Protocols, included in PCI-X 2.0



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

PCI-X I/O Signaling Voltages

- PCI-X 66 and 133
 - ✓ 3.3V signaling
 - ✓ Card-edge connector keyed for “3.3V” or “Universal” signaling
- PCI-X 266 and 533 use combination of 3.3V I/O and new 1.5V I/O
 - ✓ Control signals use 3.3V I/O
 - ✓ Data and strobe signals use 1.5V I/O
 - Faster signaling rates
 - Point-to-point and electrically terminated for improved noise immunity
 - New interface low-power state to manage interface power
 - I/O buffer change only
 - Same system supply voltages
 - Automatic selection by devices at power-up
 - ✓ Card-edge connector keyed for “3.3V signaling”



PCI-X 2.0 Offers Improved RAS Features

- Parity protection
 - ✓ Provides full compatibility with conventional PCI and PCI-X 1.0
- ECC protection new in PCI-X 2.0
 - ✓ Covers both header and payload
 - ✓ Provides automatic single bit error recovery
 - ✓ Detects all double bit errors
 - ✓ Detects all errors in single nibble
 - ✓ Detects phase errors (e.g. missed strobe or extra strobe)
 - ✓ Adds no additional latency over parity
 - ✓ Required for Mode 2; optional for Mode 1
- Enhanced data-error recovery options
 - ✓ Available both for Mode 1 and Mode 2



Compatible with Conventional PCI

- No OS or driver change required
 - ✓ New configuration registers default to functional values
 - ✓ Optional performance tuning registers
 - ✓ Other configuration registers unchanged
 - ✓ No device programming model changes required
- Optional improved error handling
 - ✓ Enables smart device and new driver to recover from PERR# event

PCI Express Overview

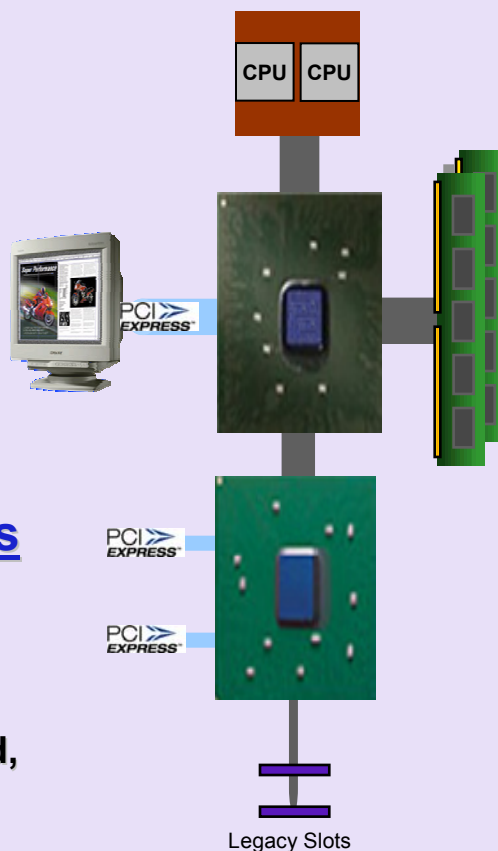
PCIe™ Architecture Primer

Scalable/Extensible I/O

- Scalable in performance and feature set
- Suitable for over 10-year horizon
- High-end and mainstream applications

Multiple Market Segments/Applications

- Mobile, desktop, server and communication devices
- Chip-to-chip, board-to-board, modules, docking, cables



Cost Effective

- PCI cost structure at system level
- Low power, no sidebands
- Commodity ingredients: FR-4 PCBs, simple connectors, low manufacturing costs

Compatibility & Smooth Migration

- Preserves investments in PCI ecosystem
- Path to future enhancements and proliferations

Serial, point-to-point interconnect of choice for all platform applications

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

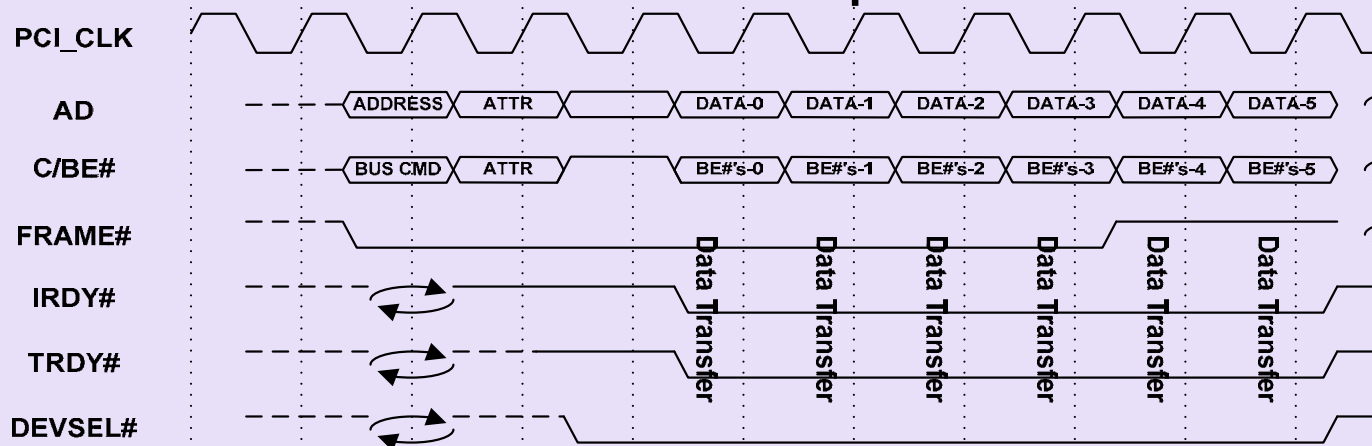
- ✓ CRC-based data integrity, hot plug, error logging

■ Enhanced Configuration Space

- ✓ Extensions and bridges into other architectures

PCIe Protocol Overview

■ PCI-X Address/Attribute phases:

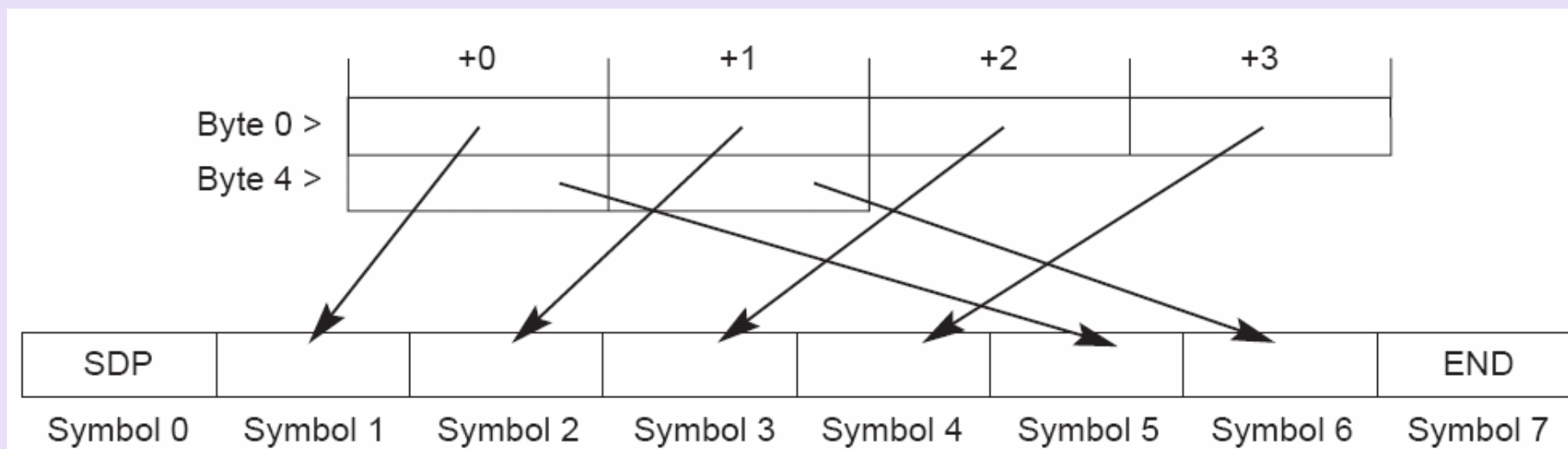


■ Evolved into the PCIe Packet Header:

	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					T D	E P	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

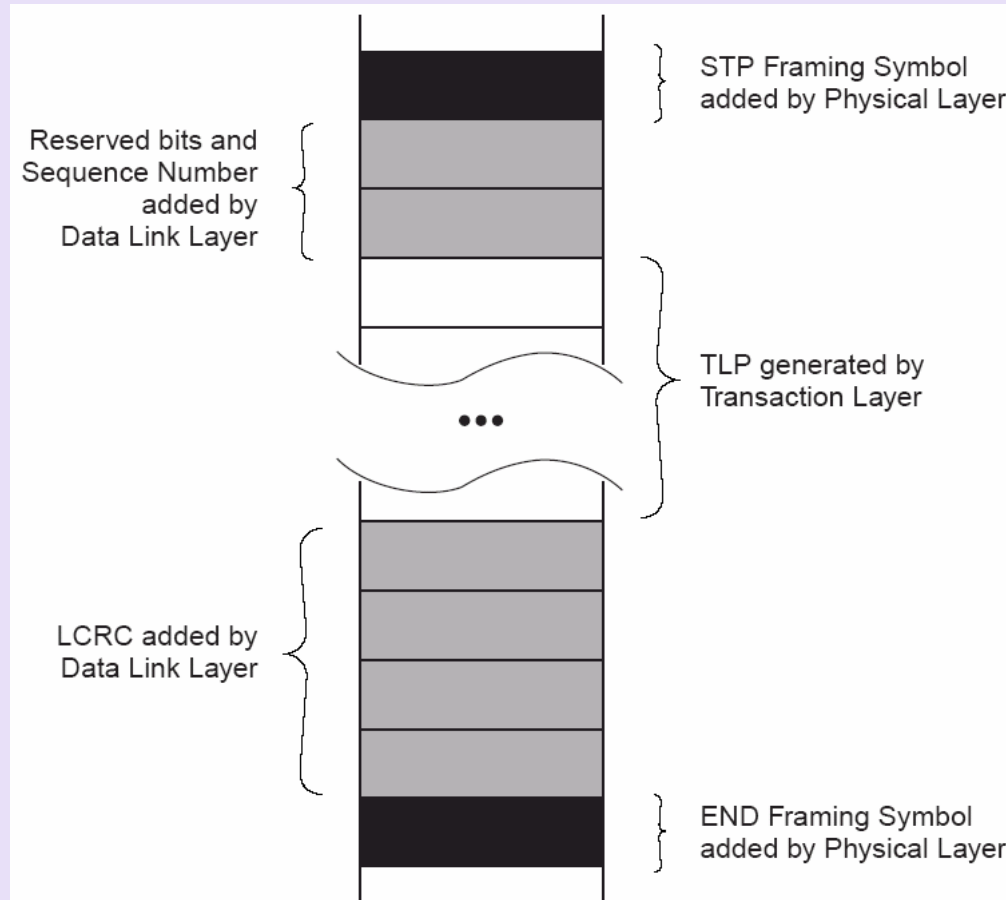
- The packet bytes get converted to 8b/10b and serialized



PCIe Protocol Overview

- Framing varies depending on link width

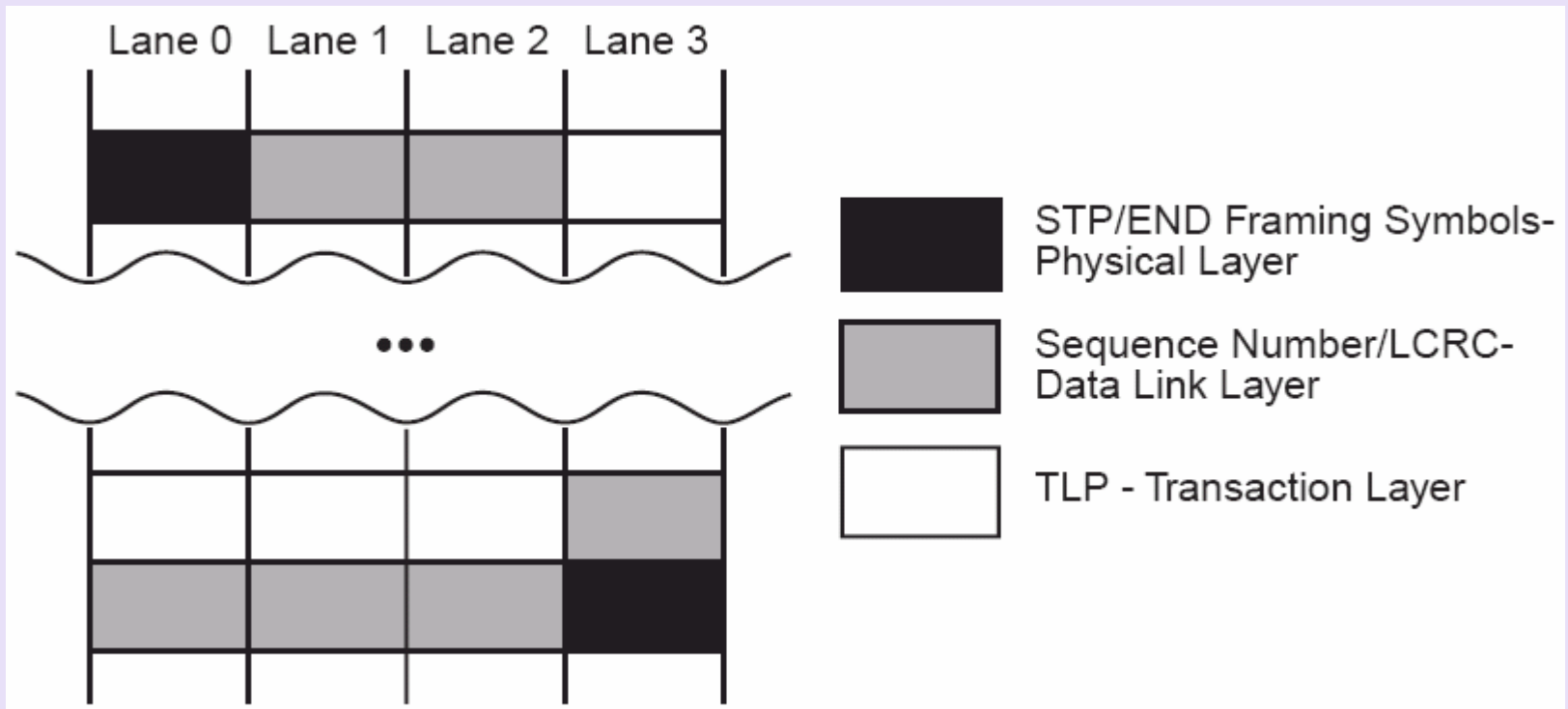
✓ x1



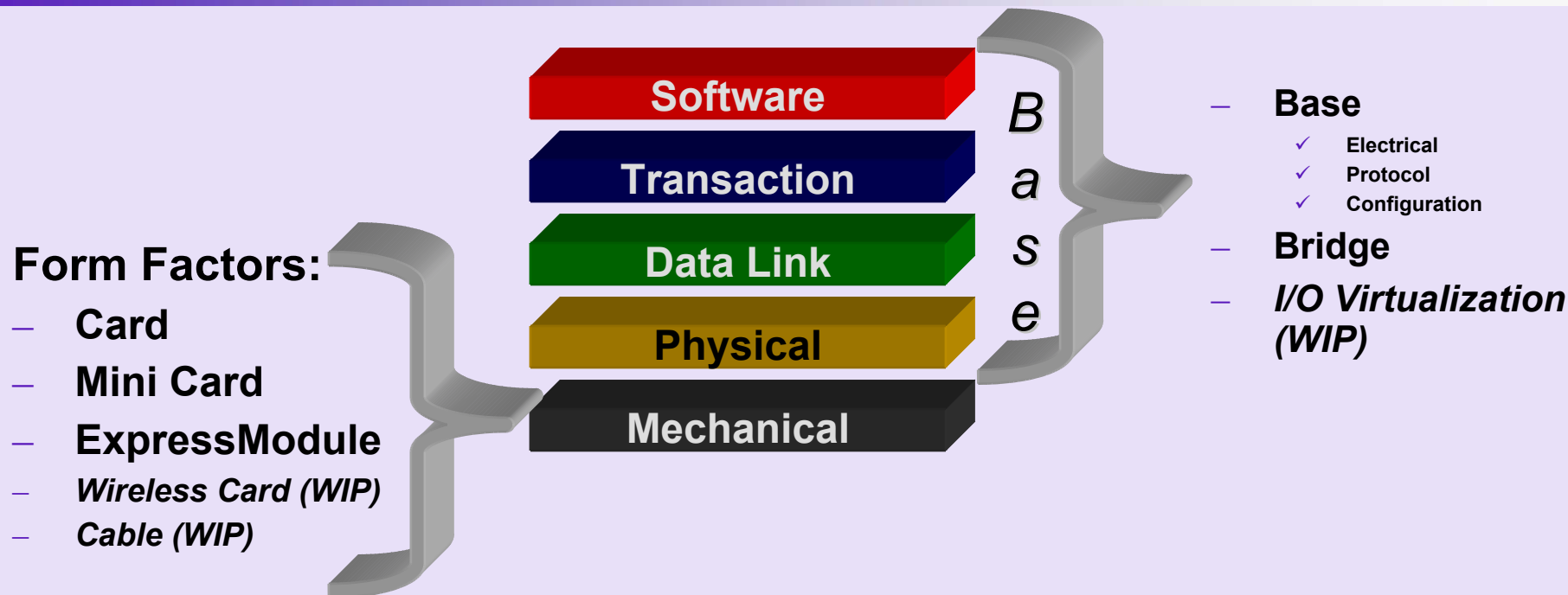
PCIe Protocol Overview

- Framing varies depending on link width

✓ x4



PCIe Architecture Specifications



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