PCI Express™
Protocol Overview

Carl Jackson
Hewlett-Packard
Member, PCI Express Protocol WG
carl.jackson@hp.com
Agenda

- PCI Express Features Summary
- Transactions and Packets
- Transaction Ordering
- Flow Control
- Virtual Channels
- Data Integrity
- Error Management / Classification
- Bridging to PCI / PCI-X
- Additional Features
- Specification Errata
- Summary
PCI Express™ Features Summary

- **Physical Interface:**
  - Point-to-point full-duplex interconnect
  - Differential low voltage signaling
  - Embedded clocking
  - Supports connectors and cables

- **Performance:**
  - Scaleable frequency (2.5 Gb/sec initially)
  - Scalable width (1, 2, 4, 8, 12, 16, 32)
  - Low latency and high utilization

- **PCI Compatibility:**
  - Configuration model and PCI software driver model
  - PCI PM software compatible

- **Protocol:**
  - Fully packetized split-transaction
  - Credit-based flow control
  - Hierarchical topology support
  - Virtual channel mechanism

- **Advanced Capabilities:**
  - Enhanced configuration and power management
  - RAS: CRC-based data integrity, hot plug, advanced error logging / reporting
Transaction Basics

- Full split-transaction packet protocol
  - Request Packet (e.g. 1R)
  - Completion Packet (e.g. 1C)
- Transactions flow between two ends
  - Switches are transient elements
  - Subject to Ordering, Flow Control and Data Integrity mechanisms

Link (local) = Between two components
End to End = Between Requester/Completer
Packet Sources and Types

- **Transaction Layer Packet (TLP)**
  - Memory, IO, Configuration, and Message transactions
  - Request and Completion packets

- **Data Link Layer Packet (DLLP)**
  - Link Data Integrity Ack/Nak Support
  - Flow Control exchange
  - Low-level Power Management
Packet Formation – TLP & DLLP

Core of TLP not modified by Data Link or Physical Layers
Core of DLLP not modified by Physical Layer
<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Payload Indicator and 12B header flag</td>
<td>2b</td>
<td>Fmt</td>
</tr>
<tr>
<td>Type</td>
<td>5b</td>
<td>Memory, Configuration or IO Completion</td>
</tr>
<tr>
<td>Traffic Class</td>
<td>3b</td>
<td>Reserved for future expansion</td>
</tr>
<tr>
<td>E2E Data Integrity</td>
<td>2b</td>
<td>Attributes: Snoop, Ordering</td>
</tr>
<tr>
<td>Payload Size</td>
<td>10b</td>
<td></td>
</tr>
<tr>
<td>Completer ID</td>
<td>16b</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>3b</td>
<td></td>
</tr>
<tr>
<td>Attr</td>
<td>2b</td>
<td></td>
</tr>
<tr>
<td>Byte Count</td>
<td>12b</td>
<td></td>
</tr>
<tr>
<td>Requestor ID</td>
<td>16b</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>8b</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>1b</td>
<td></td>
</tr>
<tr>
<td>Lower Address</td>
<td>7b</td>
<td></td>
</tr>
<tr>
<td>Bus#, Device#, Function#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completion Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transaction Tag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byte Count Modified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byte Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Bridges to PCI-X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DLLP - Detail

DLLP Types:
- Ack/Nak
- InitFC/UpdateFC
- PM DLLP

Type (8b)

Type Dependent Format (24b)

CRC (16b)

DLLP Data Integrity

Ack/Nak:
- Reserved (12b)
- Sequence # (12b)

InitFC/UpdateFC:
- R(2b)
- HdrFC(8b)
- R(2b)
- DataFC(12b)

PM DLLP:
- Reserved (24b)
Message Transaction Support

- Message transactions are a type of Request
- Miscellaneous “side-band” signals replaced by in-band messages
- Applications for In-Band “Virtual Wire” Signaling
  - Error Signaling
  - Interrupt Signaling (INTx)
  - Power Management
  - Hot-Plug Signaling
  - Slot Power Limit
  - Lock Support
  - Vendor-specific Messages
Transaction Ordering

- Ordering rules cover transactions end-to-end
- Data Link Layer
  - Transactions serialized i.e. no reordering
- Transaction Layer
  - Baseline = PCI Ordering
  - Ordering relaxation via Transaction Attributes
  - Virtual Channels are ordering independent
# Ordering Rules

<table>
<thead>
<tr>
<th>Row Pass Column?</th>
<th>Posted Request</th>
<th>Non-Posted Request</th>
<th>Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Write or Message Request (Col 2)</td>
<td>Read</td>
<td>I/O or Configuration Request (Col 3)</td>
<td>Read</td>
</tr>
<tr>
<td>Read Completion (Col 4)</td>
<td>Write Request (Col 5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **a)** No
- **b)** Y/N

<table>
<thead>
<tr>
<th>Posted or Message Request (Row A)</th>
<th>Y/N</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a)</strong> No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>b)</strong> Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Posted Request (Row B)</th>
<th>I/O or Configuration Write Request (Row C)</th>
<th>Y/N</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Request (Row B)</td>
<td>No</td>
<td>Y/N</td>
<td>Y/N</td>
</tr>
<tr>
<td><strong>a)</strong> No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>b)</strong> Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completion (Row D)</th>
<th>I/O or Configuration Write (Row E)</th>
<th>Y/N</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Completion</td>
<td>a) Y/N</td>
<td>b) Y/N</td>
<td></td>
</tr>
<tr>
<td><strong>a)</strong> No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>b)</strong> Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td></td>
</tr>
</tbody>
</table>
Flow Control

- Transaction to Transaction Layer across one Link
- Prevents overflow of receive buffers
- Enables compliance with ordering rules
- Credit-based scheme
  - Transmitter throttles according to its supply of credits
  - Separate FC information for:
    - Posted, Non-posted, Completion
    - Header vs. Data
  - Requesters cannot use FC to throttle completions

Flow control and ordering rules enable deadlock-free operations
Flow Control Example

1. \textbf{A} advertises buffer space for two Non-Posted Requests (NPH)
2. \textbf{B} sends two Memory Read Requests
3. \textbf{A} consumes one of the Non-Posted Requests
4. \textbf{A} advertises the released buffer space to \textbf{B}
5. \textbf{B} sends another Memory Read Request
Virtual Channels – Key Aspects

- Independent traffic paths – ordering and flow control
- Traffic Class (TC) labeling for traffic differentiation
- Up to 8 Virtual Channels with associated service rates
  - Default TC0/VC0 = baseline (“Best effort”)
- Mapping of TCs to Virtual Channels for platform flexibility
- Two stage arbitration: Port Arbitration and VC Arbitration
- Configuration of TC/VC mapping and VC arbitration by software

VC Mechanism = Support for QoS
TC/VC Mapping Example

TC/VC mapping provides platform flexibility
Port and VC Arbitration

Routing of Traffic:
- Determine Egress Port based on address/routing info
- Determine target VC within Egress port based on TC/VC map

Port Arbitration: Arbitration between traffic targeting same VC/Egress Port
- Vendor defined fixed arbitration, e.g., Round Robin (RR); Programmable Weighted RR; Programmable Time-based WRR

VC Arbitration: Arb. between traffic from different VCs targeting same Link
- Strict priority; Round Robin; Weighted RR
Data Integrity

Framing with 8b/10b

Explicit error forwarding mechanism

Optional End-to-end 32b CRC coverage

Transaction Layer Packet protected with 32b CRC

False/missed start correction

Both local and end-to-end robust error coverage

False/missed termination correction
Link Data Integrity – Retry Example

1. Three TLPs sent from A to B
2. Packet P2 corrupted
3. B detects corruption and issues Nak DLLP
4. A resends Packet P2 and P3
5. B acknowledges successful receipt of Packets
End-to-End Data Integrity – ECRC

- Component internal errors are critical
  - Header errors → TLP misrouting
  - Data corruption → application and system failure
- End-to-end data integrity using ECRC
  - Protecting from system-wide errors
  - Enabling upper layers error recovery
- ECRC basics:
  - Optional Capability – additional 32 bit field (part of TLP)
  - Generated by the source component – applies to all invariant TLP fields
  - Switches must pass ECRC unchanged
  - Checking in the destination component – resulting behavior is device specific
- Advanced Error Reporting support is also required when ECRC is supported
Error Management

- Consistent mechanism for managing PCI Express errors
  - Signaling using Completion Status
  - Signaling using Error Messages
  - Control and Logging using configuration registers
  - Baseline and Advanced Error Reporting

- Error Classification and Mapping
  - Correctable, Uncorrectable (Fatal, Non-Fatal)
  - Mapping of Physical, Data Link, and Transaction Layer Errors

- Advanced Error Reporting expands management / reporting capability and number of managed events

Robust Data Integrity Complemented With Standardized Error Signaling / Logging
Error Classification

- **Correctable Errors (Msg: ERR_COR)**
  - Hardware corrects the error without software impact beyond performance
  - Useful for link integrity profiling
  - Examples: Invalid Symbol in packet; CRC error

- **Uncorrectable – Non-Fatal (Msg: ERR_UNC)**
  - Hardware cannot correct the error
  - Transaction lost; impacts the software
  - Link otherwise fully functional
  - Examples: Unsupported Request; Completer Abort

- **Uncorrectable – Fatal (Msg: ERR_Fatal)**
  - Hardware cannot correct the error
  - Link unreliable
  - Likely component / hierarchy reset required
  - Examples: Physical Layer Training Failure; Malformed Packet
Bridging to PCI / PCI-X

- PCI / PCI-X connectivity provided via bridges
- Internal “virtual” PCI-to-PCI bridges provide software compatibility

Diagram:
- CPU
- Root Complex
- PCI Express
- Memory
- PCI-X Bridge
- PCI-X Bridge
- PCI-X Device
- PCI-X Device
- PCI-X
- PCI-X

- Narrow High-Bandwidth Pipe
- Wide IO Fanout

Platform Conference
Additional PCI Express Features

- Active State Power Management
- Native Hot-Plug support
- Power Budgeting / Limiting Control
- Isochronous Traffic Support
- Extended Configuration Mechanism
- Unique Device Identification (EUI-64)
PCI Express Spec Errata

- Roughly 50 items plus typos; mostly clarifications
  - Important to read the Errata document
- TLP Digest (TD bit) carries ECRC only
  Error Forwarding uses EP bit
- Incorrect values in Byte Count calculation table
- Removed PCI Express Bridge and Advanced Switching material due to changes in respective specifications
- Restricted use of discontiguous byte enables
- Clarifications of cold/warm/hot reset and wakeup functionality
- New restrictions on use of Configuration Retry Status
- Data Link Layer clarifications (C27, C31, C33)
- Additional errata applicable to the physical layer

Spec Errata currently posted on PCI SIG website for 30 day membership review. See http://www.pcisig.com/specifications/pciexpress
Summary

- PCI Express advances overall platform capabilities while preserving PCI architecture and software investments.

- Layered approach and scaleable features provide stable foundation for continued growth.

- New capabilities such as VCs and Native Hot-Plug enable important emerging applications and modular form factors.
For Additional Information…

- PCI Express 1.0 Specifications and Errata are available from the PCI-SIG
  - Visit www.pcisig.com
  - Compliance and interoperability testing will be based on the current revisions *plus the errata*

- Engage with PCI-SIG Serial Technology Communications Workgroup
  - Take advantage of technical enabling
  - Participate in compliance and interoperability forums