PCI Express® Controller
Design Challenges at 16GT/s

Richard Solomon
Synopsys
Disclaimer

Presentation Disclaimer: 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®.
Agenda

PCI Express® 4.0 Overview
PCI Express 4.0 Challenges
Controller Design Action Plans
Summary/Conclusions
PCI-SIG Announces PCIe® 4.0

- Announced **Nov 29, 2011**
- 16GT/s bit rate
  - Double of 3.0 (8GT/s)
  - No protocol changes expected
- Specification Releases (*my estimates*)
  - 0.5 spec: Q1’15
  - 0.7 spec target: Q3’15 to Q4’15
- First industry products expected in late 2016; mainstream mid-late-2017
PCI Express 4.0

- Backwards compatible with prior generations
  - Connector pinout same, discussing footprint/design changes but preserving card backwards compatibility

- Signaling rate doubled to 16GT/s
  - Transition like 2.5GT/s->5GT/s, not disruptive like 5GT/s->8GT/s!
  - No encoding changes (16GT/s still 128/130)

<table>
<thead>
<tr>
<th>PCIe Architecture</th>
<th>Raw Bit Rate</th>
<th>Interconnect Bandwidth</th>
<th>Bandwidth Lane Direction</th>
<th>Total Bandwidth for x16 link</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe 1.1</td>
<td>2.5 GT/s</td>
<td>2 Gb/s</td>
<td>~250 MB/s</td>
<td>~8 GB/s</td>
</tr>
<tr>
<td>PCIe 2.0</td>
<td>5.0 GT/s</td>
<td>4 Gb/s</td>
<td>~500 MB/s</td>
<td>~16 GB/s</td>
</tr>
<tr>
<td>PCIe 3.0</td>
<td>8.0 GT/s</td>
<td>8 Gb/s</td>
<td>~1 GB/s</td>
<td>~32 GB/s</td>
</tr>
<tr>
<td>PCIe 4.0</td>
<td>16.0 GT/s</td>
<td>16 Gb/s</td>
<td>~2 GB/s</td>
<td>~64 GB/s</td>
</tr>
</tbody>
</table>
## Target Markets

- **GPUs** – e.g. 4K Gaming?!?!
- **Enterprise Networking**: Multi-Gb Enet
- **Enterprise Storage**: PCIe SSD RAID

### Target Applications

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Storage</th>
<th>Storage (SSD)</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Applications</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Super Computers</td>
<td>• NAS</td>
<td>• PCIe SSD NVM Express, SCSI</td>
<td>• Network processors</td>
</tr>
<tr>
<td>• Servers (incl. ARM)</td>
<td>• Disk Controllers</td>
<td>Express, SATA Express</td>
<td>• Routers</td>
</tr>
<tr>
<td>• Processors</td>
<td></td>
<td></td>
<td>• Switches</td>
</tr>
<tr>
<td>• Chipsets</td>
<td></td>
<td></td>
<td>• Ethernet chips</td>
</tr>
<tr>
<td>• Interconnect</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Port Types

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Storage</th>
<th>Storage (SSD)</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port Types</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Lanes

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Storage</th>
<th>Storage (SSD)</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lanes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• x8, x16</td>
<td>• x8, x16</td>
<td>• x1, x2, x4</td>
<td>• x8, x16</td>
</tr>
</tbody>
</table>

### 16GT/s

<table>
<thead>
<tr>
<th>Data Center</th>
<th>Storage</th>
<th>Storage (SSD)</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16GT/s</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 0.5 Design Start</td>
<td>• 0.5 Design Start</td>
<td>• 0.7 Design Start</td>
<td>• 0.5 Design Start</td>
</tr>
</tbody>
</table>

---

Copyright © 2015, PCI-SIG, All Rights Reserved
Channels Shrinking

- PCI-SIG specified retimers & re-drivers to extend reach and to achieve today’s 16” channel lengths at 16GT/s
  - Extension Devices ECN
  - Additional specification in PCIe 4.0 Base
Channels Shrinking

- PCI-SIG specified retimers & re-drivers to extend reach and to achieve today’s 16” channel lengths at 16GT/s
  - Extension Devices ECN
  - Additional specification in PCIe 4.0 Base

NO REQUIRED CONTROLLER CHANGES!
Re-drivers & Retimers

- **Re-driver**
  - “A non-protocol aware, software transparent, analog only, Extension Device.”*

- **Retimer**
  - “A Physical Layer protocol aware, software transparent, Extension Device that forms two separate electrical Link Segments.”*

- **Impact to devices**
  - “There are no required changes for existing PCI Express 3.0/3.1 devices. New PCI Express 3.0/3.1 devices can optionally implement features to detect Retimer presence, report presence through configuration space and/or optionally adjust other parameters based on Retimer presence.”*

*All quotes from “Extension Devices ECN” 6-October-2014
Re-drivers & Retimers

- **Re-driver**
  - ✓ “A non-protocol aware, software transparent, analog only, Extension Device.”*

- **Retimer**
  - ✓ “A Physical Layer protocol aware, software transparent, Extension Device that forms two separate electrical Link Segments.”*

- **Impact to devices**
  
  “There are no required changes for existing PCI Express 3.0/3.1 devices. New PCI Express 3.0/3.1 devices can optionally implement features to detect Retimer presence, report presence through configuration space and/or optionally adjust other parameters based on Retimer presence.”*

*All quotes from “Extension Devices ECN” 6-October-2014
Retimer Detection

- Retimers set a reserved bit in the TS2s they forward
- Downstream Ports (switches and RCs) are allowed to detect this and set new configuration space bits
  - Retimer Presence Detect Supported
- Endpoint designers have NO change
- Switch and Root Complex designers have option to detect and report retimer presence
Retimer Detection

- No spec on what to do when retimer present:
  “It is recommended that designers of Upstream and Downstream Components consider Retimer latency when determining the following characteristics:
  - Data Link Layer Retry Buffer size
  - Transaction Layer Receiver buffer size and Flow Control Credits
  - Data Link Layer REPLAY_TIMER Limits
  Additional buffering (replay or FC) may be required to compensate for the additional channel latency.”

- Reality: retimers constrained by existing devices
  ✓ It’s possible PCIe 4.0 will account for retimers

- Nonetheless – erring on the side of slightly larger values may help you in the future
Agenda

PCI Express 4.0 Overview
PCI Express 4.0 Challenges
Controller Design Action Plans
Summary/Conclusions
PCI Express 4.0 Challenges

- Link Equalization
- PHY Interface
- Multiple Packets per Clock Cycle
- Higher Bandwidth

- Degrees of difficulty vary with
  - Design team’s familiarity with PCIe
  - Current bandwidth
  - Target bandwidth

Copyright © 2015, PCI-SIG, All Rights Reserved
LINK EQUALIZATION
Link Equalization

- PCIe 3.0 added Link Equalization
  - Four phase process - exchange presets at 2.5GT/s then switch to 8GT/s and perform equalization
- PCIe 4.0 will extend that to 16GT/s
  - Likely to require 2-steps
  - Expect to bring link to 8GT/s and then repeat four phases to go to 16GT/s
- Possible LTSSM changes
  - May at least need to tweak some arcs to repeat EQ if that’s chosen
Link Equalization

- This is a possible open area of specification

- Low-Moderate risk of design changes
  - Changes fairly isolated, so overall risk of executing early is low

- Many bigger challenges ahead so either
  - Implement what comes out in 0.5  *OR*
  - Hold off and wait it out
Link Equalization

- This is a possible open area of specification

- Low-Moderate risk of design changes
  - Changes fairly isolated, so overall risk of executing early is low

- Many bigger challenges ahead so either
  - Implement what comes out in 0.5 *OR*
  - Hold off and wait it out

Monitor PCIe 4.0
Drafts Closely!
PHY INTERFACE
PHY Interface

- Width & Frequency
  - Obviously first-order concern
  - Common practice has been 16-bits / lane
  - Common maximum frequency has been 500MHz
  - Variable width or variable frequency when changing speeds?

- Interface signaling details
  - Same or new controls for equalization?
  - Clocking changes?
  - Other?
PHY Interface

- **Width & Frequency**
  - Obviously first-order concern
  - Common practice has been 16 bits / lane
  - Common maximum frequency has been 500MHz
  - Variable width or variable frequency when changing speeds?

- **Interface signaling details**
  - Same or new controls for equalization?
  - Clocking changes?
  - Other?
MULTIPLE PACKETS PER CLOCK CYCLE
Multiple Packets per Clock

- Issue showed up with PCIe 3.0 at 8GT/s
- Several ways to describe it … Let’s try simply
- Smallest PCIe TLP is 3 DWORDs + LCRC (=128 bits)
- At 8GT/s 16-bit PHY runs 500MHz
  - ✓ x1-x8 at most a single complete packet per clock
    - 16 bits / lane * 8 lanes = 128 bits
  - ✓ x16 two complete packets per clock possible
    - 16 bits / lane * 16 lanes = 256 bits
Multiple Packets per Clock

- At 16GT/s 32-bit PHY runs 500MHz
  - ✔ x1-x4 single complete packet per clock
    - 32 bits / lane * 4 lanes = 128 bits
  - ✔ x8 two complete packets per clock possible
    - 32 bits / lane * 8 lanes = 256 bits
  - ✔ x16 four complete packets per clock possible
    - 32 bits / lane * 16 lanes = 512 bits
Multi-packet Solutions
Option 1

- Limit interface width to 128-bit
  - Requires increased clock frequency
    - 8GT/s x16 = 1GHz
    - 16GT/s x8 = 1GHz
    - 16GT/s x16 = 2GHz
  - Easier architectural/RTL design
    - For both controller designer & user (SoC designer)
  - “Challenging” gate-implementation
    - Closing timing flop-flop
    - RAM availability
Multi-packet Solutions Option 2

- Provide multiple packet paths
  - Easy out for controller designer
  - Forces user to multi-thread to handle multiple packets per clock
  - Forces user to maintain ordering among threads
  - Theoretical best performance
Multi-packet Solutions Option 3

- Serialize data stream
  - Controller designer guarantees never multiple packets per clock
  - Easy for user
  - Forces controller designer to duplicate logic
  - Forces controller designer to maintain ordering
  - Theoretically lower performance in pathological cases
    - 100% minimum packets has to fall off somewhere

Copyright © 2015, PCI-SIG, All Rights Reserved
Multi-packet Solutions Option 4 (2+3 = 4?)

- Widen controller datapath and serialize data stream
  - Decreases clock frequency
  - Requires more routing resources
    - May be easier for user than upping clock frequency
  - Forces controller designer to duplicate logic
  - Forces controller designer to maintain ordering
  - Theoretically lower performance in pathological cases
    - 100% minimum packets has to fall off somewhere

<table>
<thead>
<tr>
<th></th>
<th>128-bit</th>
<th>256-bit</th>
<th>512-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>16GT/s x8</td>
<td>1GHz</td>
<td>500MHz</td>
<td>250MHz</td>
</tr>
<tr>
<td>16GT/s x16</td>
<td>2GHz</td>
<td>1GHz</td>
<td>500MHz</td>
</tr>
</tbody>
</table>
HIGHER BANDWIDTH
16GT/s Introspection

- Which speed/architecture do you **least dislike**?
  - ✓ 512 bit data bus running @ 500MHz **OR**
  - ✓ 256 bit data bus running @ 1GHz?

- Do you have concerns about
  - ✓ 256-bit @ 1GHz
    - Interfacing to the controller e.g. application side at 1GHz?
    - Providing memories at 1GHz?
  - ✓ 512-bits @ 500MHz
    - Do you anticipate routing issues with a 512-bit data bus?
16GT/s Introspection Silicon Technology

- What geometry do you intend to use?
  - 28nm, 20nm or lower?
- What process do you intend to use?
  - HPP, HPM, HPL, LP, other?
- Will you use ULVT/LVT cells or at least mixed VT ASIC libraries?
- High speed or high density tech libraries?
- What are your worst case for
  - Voltage
  - Temperature
  - Inverted temp
Agenda

PCI Express 4.0 Overview
PCI Express 4.0 Challenges
Controller Design Action Plans
Summary/Conclusions
Action Plan
Evaluate Background & Goals

- Designers’ Background
  - First PCIe controller implementation
  - Have 2.5GT/s or 5GT/s implementation but no 8GT/s implementation
  - Have 8GT/s implementation of x1-x8 link
  - Have 8GT/s implementation of x16 link

- Implementation Target
  - 16GT/s x1-x4
  - 16GT/s x8
  - 16GT/s x16
## Action Plan Migration Challenges

<table>
<thead>
<tr>
<th></th>
<th>x1-x4</th>
<th>x8</th>
<th>x16</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>8GT/s x16</strong></td>
<td>Bandwidth Lower!</td>
<td>Same bandwidth</td>
<td>Bandwidth doubles</td>
</tr>
<tr>
<td></td>
<td>New EQ</td>
<td>Same 2 packets/clk</td>
<td>4 packets per clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New EQ</td>
<td>New EQ</td>
</tr>
<tr>
<td><strong>8GT/s x1-x8</strong></td>
<td>Same bandwidth?</td>
<td>Bandwidth 2X</td>
<td>Bandwidth 4X</td>
</tr>
<tr>
<td></td>
<td>New EQ</td>
<td>2 packets per clock</td>
<td>4 packets per clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New EQ</td>
<td>New EQ</td>
</tr>
<tr>
<td><strong>2.5 or 5GT/s</strong></td>
<td>Equalization 130b-</td>
<td>2 packets per clock</td>
<td>4 packets per clock</td>
</tr>
<tr>
<td></td>
<td>Encoding</td>
<td>Equalization 130b-Encoding</td>
<td>Equalization 130b-Encoding</td>
</tr>
<tr>
<td><strong>First</strong></td>
<td>PCIe Protocol 10b-</td>
<td>PCIe Protocol 10b-</td>
<td>PCIe Protocol 10b-</td>
</tr>
<tr>
<td><strong>implementation</strong></td>
<td>Encoding Equalization</td>
<td>2 packets per clock</td>
<td>4 packets per clock</td>
</tr>
<tr>
<td></td>
<td>130b-Encoding</td>
<td>Equalization 130b-Encoding</td>
<td>Equalization 130b-Encoding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Action Plan

## Migration Challenges

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>x1-x4</th>
<th>x8</th>
<th>x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>8GT/s x16</td>
<td>SLACKER!!! Go buy beer for the rest. 😊 Follow the spec but relax. Start “later”. Double your b/w and multi-packet logic. Start soon.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8GT/s x1-x8</td>
<td>Follow the spec but relax. Start “later” Follow the spec but relax. Start “later”. Double your b/w, solve multi-packet. Start now! Quadruple your b/w, solve multi-packet. Start now!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5 or 5GT/s</td>
<td>More b/w, new protocol and encoding. Start now! Lots more b/w, solve multi-packet. Start yesterday! Lots &amp; lots more b/w, solve multi-packet. Start last week!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First implementation</td>
<td>Tons of new stuff, you’re either good or suicidal! Start last week! When did you tell management you’d have this thing done? Spiff up resume! “Would you like to super-size that combo meal?”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

PCI Express 4.0 Overview
PCI Express 4.0 Challenges
Controller Design Action Plans
Summary/Conclusions
Summary / Conclusions

- Start work on 16GT/s now to intercept PHYs and specification
  - Monitor PHY interface specifications
  - Consider retimer implications, especially for Switch & Root Complex designs

- Make datapath and clocking decisions now
  - Non-trivial architectural changes
    - Driven by data rate, mostly immune to spec changes
  - May ripple into application logic
    - Especially for multi-packet per clock cycle cases

- Get physical design involved early
  - Especially for x16 implementations
Thank you for attending the PCI-SIG Developers Conference
Israel 2015

For more information please go to
www.pcisig.com
PCI Express® Controller
Design Challenges at 16GT/s

Richard Solomon
Synopsys