PCle® 7.0 Specification: 128 GT/s Bandwidth for Future Data-Intensive Markets

Presented by:
Dr. Debendra Das Sharma
Intel Senior Fellow and co-GM, Memory and I/O Technologies
Member, PCI-SIG® Board
Agenda

- Introduction: Evolution of PCI Express® Technology
- PCIe® 6.0 and PCIe 7.0 Specifications: A Deep Dive
- PCI Express Technology and Storage
- Form Factors
- Optical-friendly PCIe Technology
- Compliance
- Conclusions
Evolution of PCI Express® Specification

- PCIe® specification doubles the data rate every generation with full backwards compatibility every 3 years
- Ubiquitous I/O across the compute continuum: PC, Hand-held, Workstation, Server, Cloud, Enterprise, HPC, Embedded, IoT, Automotive, AI
- One stack / same silicon across all segments with different form-factors; a x16 PCIe 5.0 device interoperates with a x1 PCIe 1.0 device!
- PCIe 7.0 specification currently at Rev 0.3 level maturity – making good progress

<table>
<thead>
<tr>
<th>Revision</th>
<th>Max Data Rate</th>
<th>Encoding</th>
<th>Signaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe 1.0 (2003)</td>
<td>2.5 GT/s</td>
<td>8b/10b</td>
<td>NRZ</td>
</tr>
<tr>
<td>PCIe 2.0 (2007)</td>
<td>5.0 GT/s</td>
<td>8b/10b</td>
<td>NRZ</td>
</tr>
<tr>
<td>PCIe 3.0 (2010)</td>
<td>8.0 GT/s</td>
<td>128b/130b</td>
<td>NRZ</td>
</tr>
<tr>
<td>PCIe 4.0 (2017)</td>
<td>16.0 GT/s</td>
<td>128b/130b</td>
<td>NRZ</td>
</tr>
<tr>
<td>PCIe 5.0 (2019)</td>
<td>32.0 GT/s</td>
<td>128b/130b</td>
<td>NRZ</td>
</tr>
<tr>
<td>PCIe 6.0 (2022)</td>
<td>64.0 GT/s</td>
<td>1b/1b (Flit Mode*)</td>
<td>PAM4</td>
</tr>
<tr>
<td>PCIe 7.0 (2025)</td>
<td>128.0 GT/s</td>
<td>1b/1b (Flit Mode*)</td>
<td>PAM4</td>
</tr>
</tbody>
</table>

(*Flit Mode also enabled in other Data Rate with their respective encoding)

PCIe architecture continues to deliver bandwidth doubling for 7 generations spanning 3 decades! An impressive run!
## PCI Express® Specifications: Speeds and Feeds

### PCIe® Speeds/Feeds - Pick Your Bandwidth

- Flexible to meet needs from handheld/client to server/HPC
- Max Total Bandwidth = Max RX bandwidth + Max TX bandwidth
- 35 Permutations yielding 11 unique bandwidth profiles
- Encoding overhead and header efficiency not included

<table>
<thead>
<tr>
<th>Specifications</th>
<th>x1</th>
<th>x2</th>
<th>x4</th>
<th>x8</th>
<th>x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5 GT/s (PCIe 1.x +)</td>
<td>500 MB/S</td>
<td>1 GB/S</td>
<td>2 GB/S</td>
<td>4 GB/S</td>
<td>8 GB/S</td>
</tr>
<tr>
<td>5.0 GT/s (PCIe 2.x +)</td>
<td>1 GB/S</td>
<td>2 GB/S</td>
<td>4 GB/S</td>
<td>8 GB/S</td>
<td>16 GB/S</td>
</tr>
<tr>
<td>8.0 GT/s (PCIe 3.x +)</td>
<td>2 GB/S</td>
<td>4 GB/S</td>
<td>8 GB/S</td>
<td>16 GB/S</td>
<td>32 GB/S</td>
</tr>
<tr>
<td>16.0 GT/s (PCIe 4.x +)</td>
<td>4 GB/S</td>
<td>8 GB/S</td>
<td>16 GB/S</td>
<td>32 GB/S</td>
<td>64 GB/S</td>
</tr>
<tr>
<td>32.0 GT/s (PCIe 5.x +)</td>
<td>8 GB/S</td>
<td>16 GB/S</td>
<td>32 GB/S</td>
<td>64 GB/S</td>
<td>128 GB/S</td>
</tr>
<tr>
<td>64.0 GT/s (PCIe 6.x +)</td>
<td>16 GB/S</td>
<td>32 GB/S</td>
<td>64 GB/S</td>
<td>128 GB/S</td>
<td>256 GB/S</td>
</tr>
<tr>
<td>128.0 GT/s (PCIe 7.x +)</td>
<td>32 GB/S</td>
<td>64 GB/S</td>
<td>128 GB/S</td>
<td>256 GB/S</td>
<td>512 GB/S</td>
</tr>
</tbody>
</table>

+ = data rate supported by this and subsequent spec revisions.
Bandwidth Drivers for PCI Express® Specifications

- Device side: Networking (800Gb/s -> 1.6 Tb/s), Accelerators, FPGA/ASICs, Memory (need more memory b/w)
- Alternate Protocols (CXL™, proprietary SMP cache coherency protocols for multi-socket servers) on PCIe® architecture
- As compute capability grows exponentially, so does I/O bandwidth
  - Platform already has hundreds of lanes for I/O => speed has to go up
- But ... we need to meet the cost, performance, power metrics as an ubiquitous I/O with hundreds of Lanes in a platform

New usage models are driving bandwidth demand – doubling every three years (New Usage Models: Cloud, AI/Analytics, Edge)
PCle® 6.0 and PCle 7.0 Specifications:
A Deep Dive
## Key Metrics for PCIe® 6.0/ 7.0 Architecture: Requirements

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Expectations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>PCIe 6.0 data rate @ 64 GT/s -&gt; PCIe 7.0 data rate @ 128.0 GT/s, PAM4 signaling (double the bandwidth per pin every generation)</td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td>&lt;10ns adder for Transmitter + Receiver over 32.0 GT/s (including FEC) (We can not afford the 100ns FEC latency as networking does with PAM4)</td>
</tr>
<tr>
<td><strong>Bandwidth Inefficiency</strong></td>
<td>&lt;2 % adder over PCIe 5.0 specification across all payload sizes</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>0 &lt; FIT &lt;&lt; 1 for a x16 (FIT – Failure in Time, number of failures in $10^9$ hours)</td>
</tr>
<tr>
<td><strong>Channel Reach</strong></td>
<td>Similar to PCIe 5.0 specification under similar set up for Retimer(s) (maximum 2)</td>
</tr>
<tr>
<td><strong>Power Efficiency</strong></td>
<td>Better than PCIe 5.0 specification</td>
</tr>
</tbody>
</table>
| **Low Power**            | Similar entry/ exit latency for L1 low-power state  
Addition of a new power state (L0p) to support scalable power consumption with bandwidth usage without interrupting traffic |
| **Plug and Play**        | Fully backwards compatible with PCIe 1.x through PCIe 5.0/6.0 specifications                                                              |
| **Others**               | HVM-ready, cost-effective, scalable to hundreds of Lanes in a platform                                                                    |

Right trade-offs to meet each of these metrics!
PAM4 Signaling at 64.0 and 128.0 GT/s

- PAM4 signaling: Pulse Amplitude Modulation 4-level
  - 4 levels (2 bits) in same Unit Interval (UI); 3 eyes
  - Helps channel loss (same Nyquist as 32.0 GT/s)
- Reduced voltage levels (EH) and eye width increases susceptibility to errors
- Gray Coding to reduce errors in each UI
- Precoding to minimize errors in a burst
- Voltage levels at Tx and Rx define encoding

<table>
<thead>
<tr>
<th>Voltage Level</th>
<th>Tx Voltage</th>
<th>Rx Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-Vtx</td>
<td>V &lt;= Vth1</td>
</tr>
<tr>
<td>1</td>
<td>-Vtx/3</td>
<td>Vth1 &lt; V &lt;= Vth2</td>
</tr>
<tr>
<td>2</td>
<td>+Vtx/3</td>
<td>Vth2 &lt; V &lt;= Vth3</td>
</tr>
<tr>
<td>3</td>
<td>+Vtx</td>
<td>V &gt; Vth3</td>
</tr>
</tbody>
</table>

### Scrambled 2-bit aligned value

<table>
<thead>
<tr>
<th>Prior to Gray Coding</th>
<th>After Gray Coding</th>
<th>Unscrambled 2-bit as well TS0 Ordered Sets</th>
<th>Voltage Level</th>
<th>DC-balance Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>11</td>
<td>3</td>
<td>+3</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>+1</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>01</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>0</td>
<td>-3</td>
</tr>
</tbody>
</table>
## Error Assumptions and Characteristics w/ PAM4

Parameters of interest: FBER and error correlation within Lane and across Lanes

- **FBER – First bit error rate**
  - Probability of the first bit error occurring at the Receiver
  - Receiving Lane may see a burst propagated due to DFE
  - The number of errors from the burst can be minimized
    - Constrain DFE tap weights - balance TxEQ, CTLE and DFE equalization

- **Correlation of errors across Lanes**
  - Due to common source of errors (e.g., power supply noise)
  - Conditional probability that a first error in a Lane => errors in nearby Lanes

- **BER depends on the FBER and the error correlation in a Lane and across Lanes**

### Table: Error Correlation Across Lanes

<table>
<thead>
<tr>
<th>Lane</th>
<th>L0</th>
<th>L1</th>
<th>L2</th>
<th>...</th>
<th>L15</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Error</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Burst error in a Lane</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lane correlation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

©2023 SNIA. All Rights Reserved.
Our Approach: Light-weight FEC and Retry

- Light-weight FEC, strong CRC, and keep the overall latency (including retry) really low so that the Ld/St applications do not suffer latency penalty
- We are better off retrying a packet with 10\(^{-6}\) (or 10\(^{-5}\)) probability with a retry latency of 100ns vs having a FEC latency impact of 100ns with a much lower retry probability

Low latency mechanism w/ FBER of 1E-6 to meet the metrics (latency, area, power, bandwidth)
FLIT Encoding: Low-latency w/ High Efficiency

- FLIT (flow control unit) based: FEC needs fixed set of bytes
- Error Correction (FEC) in FLIT => CRC (detection) in FLITs => Retry at FLIT level
- Lower data rates will also use the same FLIT once enabled

- FLIT size: 256B
  - 236B TLP, 6B DLP, 8B CRC, 6B FEC
  - No Sync hdr, no Framing Token (TLP reformat), no TLP/DLLP CRC
  - Improved bandwidth utilization due to overhead amortization
  - FLIT Latency: 2ns x16, 4ns x8, 8ns x4, 16ns x2, 32ns x1
  - Guaranteed Ack and credit exchange => low Latency, low storage

- Optimization: Retry error FLIT only + existing Go-Back-N retry
- Other benefits of Flit Mode: scalability (future-proofing) with new TLP arrangement, making it easier to parse
- Once Flit mode is negotiated, it must be supported at all speeds

Low latency improves performance and reduces area
Retry Probability and FIT vs FBER Correlation

- Single Symbol Correct interleaved FEC plus 64-b CRC works well for raw FBER of 1E-6 even with high Lane correlation
- Retry probability per FLIT is $5 \times 10^{-6}$
- B/W loss is 0.05% even with go-back-n
- FIT is almost 0
- Can mitigate the bandwidth loss significantly by adopting retry only the non-NOP TLP FLIT

**Spec Requirement:** FBER of 1E-6 with a burst of $\leq 16$ to meet the performance goals with a light-weight FEC

### Retry Probability and FIT vs FBER Correlation

<table>
<thead>
<tr>
<th>Retry Time (ns)</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Burst Error Probability</td>
<td>1.0E-04, 1.0E-05, 1.0E-06, 1.0E-07</td>
</tr>
<tr>
<td>Correlation second Lanes</td>
<td>1.0E-03, 1.0E-03, 1.0E-04, 1.0E-05</td>
</tr>
<tr>
<td>Width of Link</td>
<td>16, 16, 16, 16</td>
</tr>
<tr>
<td>Frequency</td>
<td>64, 64, 64, 64</td>
</tr>
<tr>
<td>Bits per FLIT/ lane</td>
<td>128, 128, 128, 128</td>
</tr>
<tr>
<td>Prob 0 error/ Lane (no correlation Lanes)</td>
<td>0.98728094, 0.998720812, 0.999872088, 0.9999872</td>
</tr>
<tr>
<td>Prob 1 error / Lane (no correlation Lanes)</td>
<td>0.01272526, 0.001272857, 0.000127288, 0.0127289</td>
</tr>
<tr>
<td>Prob 2 errors/Lane (no correlation Lanes)</td>
<td>0.00526225, 0.0052675, 0.0052708, 0.0052742</td>
</tr>
<tr>
<td>Prob 3 errors/Lane (no correlation Lanes)</td>
<td>3.37135E-07, 3.4095E-10, 3.4333E-13, 3.4337E-16</td>
</tr>
<tr>
<td>Prob 4 errors/Lane (no correlation Lanes)</td>
<td>0.015365E-09, 0.106546E-13, 0.10667E-17, 0.10668E-21</td>
</tr>
<tr>
<td>Prob 0 errors in FLIT (w/ Lane correlation)</td>
<td>0.814081918, 0.0972872819, 0.097954095, 0.9997952</td>
</tr>
<tr>
<td>Prob 1 errors in FLIT (w/ Lane correlation)</td>
<td>0.0165450705, 0.019778713, 0.000204878, 0.00020473</td>
</tr>
<tr>
<td>Prob 2 errors in FLIT (w/ Lane correlation)</td>
<td>0.0018480047, 0.000048716, 5.01219E-06, 0.0164E-06</td>
</tr>
<tr>
<td>Prob 3 errors in FLIT (w/ Lane correlation)</td>
<td>0.0001203308, 0.002153E-06, 4.1132E-09, 0.4122E-12</td>
</tr>
<tr>
<td>Prob 4 errors in FLIT (w/ Lane correlation)</td>
<td>0.00427E-05, 0.04517E-08, 0.4716E-12, 4.7348E-16</td>
</tr>
<tr>
<td>Prob 0 errors all Lanes/ FLIT (w/ correlation)</td>
<td>0.0014600198, 0.0972872819, 0.097954095, 0.9997952</td>
</tr>
<tr>
<td>Prob of 1 error all Lanes/ FLIT</td>
<td>0.1664022747, 0.019766156, 0.00020474, 0.00020473</td>
</tr>
<tr>
<td>Retry Prob/ FLIT (&gt;1 error in all Lanes/ FLIT)</td>
<td>0.019747377, 0.000493096, 5.0272E-06, 0.0002047</td>
</tr>
<tr>
<td>Number of FLITs over retry window</td>
<td>100, 100, 100, 100</td>
</tr>
<tr>
<td>0 uncorrected FLIT errors over retry window</td>
<td>0.136822199, 0.095187476, 0.099497494, 0.09999949</td>
</tr>
<tr>
<td>1 uncorrected FLIT errors over retry window</td>
<td>0.272140195, 0.046959754, 0.00020475, 0.0037E-06</td>
</tr>
<tr>
<td>Retry prob over Retry time</td>
<td>0.865917801, 0.048112573, 0.00020476, 0.0037E-06</td>
</tr>
<tr>
<td>Time per FLIT (ns)</td>
<td>2, 2, 2, 2</td>
</tr>
<tr>
<td>FLITs per sec</td>
<td>500000000, 500000000, 500000000, 500000000</td>
</tr>
<tr>
<td>FLITs per 1E9 hrs</td>
<td>1.8E+21, 1.8E+21, 1.8E+21, 1.8E+21</td>
</tr>
<tr>
<td>CRC bits</td>
<td>64, 64, 64, 64</td>
</tr>
<tr>
<td>Aliasing Prob</td>
<td>5.4210E-20, 5.4210E-20, 5.4210E-20, 5.4210E-20</td>
</tr>
<tr>
<td>SDC/ FIT</td>
<td>2.950354E-24, 2.4892E-27, 2.55895E-31, 2.5667E-35</td>
</tr>
<tr>
<td>FIT (Failure in Time)</td>
<td>0.005310986, 0.48058E-08, 0.46072E-10, 0.46021E-14</td>
</tr>
<tr>
<td>Effective BER (Single Symbol Correct)</td>
<td>0.005310986, 0.48058E-08, 0.46072E-10, 0.46021E-14</td>
</tr>
<tr>
<td>Effective BER (Double Symbol Correct)</td>
<td>0.005310986, 0.48058E-08, 0.46072E-10, 0.46021E-14</td>
</tr>
<tr>
<td>Effective BER (Triple Symbol Correct)</td>
<td>1.70087E-07, 1.4349E-10, 1.4755E-14, 1.4796E-18</td>
</tr>
</tbody>
</table>

(Numbers get worse by 2x at 128.0 GT/s - still well within expectations)
PCIe® 6.0 Specification FLIT Mode Bandwidth at 64.0 GT/s

- Bandwidth increase = 2X (BW efficiency of FLIT mode) / (BW efficiency in non-FLIT mode)
- Overall we see a >2X improvement in bandwidth (benefits most systems)
  - Efficiency gain reduces as TLP size increases
  - Beyond 512 B (128 DW) payload goes below 1
- Bandwidth efficiency improvement in FLIT mode due to the amortization of CRC, DLP, and ECC over a FLIT (8% overhead) – works out better than sync hdr, DLLP, Framing Token per TLP, and 4B CRC per TLP overheads in PCIe 5.0 specification
- Expect 2X increase in bandwidth at 128.0 GT/s with PCIe 7.0 specification

Bandwidth Efficiency improvement causes > 2X bandwidth gain for up to 512B Payload in 64.0 GT/s FLIT mode
Latency Impact of FLIT Mode

- FLIT accumulation in Rx only (Tx pipeline)
- FEC + CRC delay expected to be ~ 1-2 ns
- Expected Latency savings due to removal of sync hdr, fixed FLIT sizes (no framing logic, no variable sized TLP/CRC processing) is not considered in Tables here
- With twice the data rate and the above optimizations, realistically expect to see lower latency except for x2 and x1 for smaller payload TLPs –worst case ~10ns adder
- Latency expected to improve at 128.0 GT/s as the accumulation time halves from 64.0 GT/s

<table>
<thead>
<tr>
<th>Data Size (DW)</th>
<th>TLP Size (DW)</th>
<th>Latency in ns for 128b/130b @ 32.0GT/s</th>
<th>Latency in ns in FLIT Mode @ 64.0 GT/s</th>
<th>Latency Increase due to accumulation (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>6.09375</td>
<td>18</td>
<td>11.90625</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>10.15625</td>
<td>20</td>
<td>9.84375</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>14.21875</td>
<td>22</td>
<td>7.78125</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>22.34375</td>
<td>26</td>
<td>3.65625</td>
</tr>
<tr>
<td>32</td>
<td>36</td>
<td>38.59375</td>
<td>34</td>
<td>-4.59375</td>
</tr>
<tr>
<td>64</td>
<td>68</td>
<td>71.09375</td>
<td>50</td>
<td>-21.09375</td>
</tr>
<tr>
<td>128</td>
<td>132</td>
<td>136.09375</td>
<td>82</td>
<td>-54.09375</td>
</tr>
<tr>
<td>256</td>
<td>260</td>
<td>266.09375</td>
<td>146</td>
<td>-120.09375</td>
</tr>
<tr>
<td>512</td>
<td>516</td>
<td>526.09375</td>
<td>274</td>
<td>-252.09375</td>
</tr>
<tr>
<td>1024</td>
<td>1028</td>
<td>1046.09375</td>
<td>530</td>
<td>-516.09375</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Size (DW)</th>
<th>TLP Size (DW)</th>
<th>Latency in ns for 128b/130b @ 32.0GT/s</th>
<th>Latency in ns in FLIT Mode @ 64.0 GT/s</th>
<th>Latency Increase due to accumulation (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>0.380859375</td>
<td>1.125</td>
<td>0.744140625</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.634765625</td>
<td>1.25</td>
<td>0.615234375</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>0.888671875</td>
<td>1.375</td>
<td>0.486328125</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>1.396484375</td>
<td>1.625</td>
<td>0.228515625</td>
</tr>
<tr>
<td>32</td>
<td>36</td>
<td>2.412109375</td>
<td>2.125</td>
<td>-0.287109375</td>
</tr>
<tr>
<td>64</td>
<td>68</td>
<td>4.443359375</td>
<td>3.125</td>
<td>-1.318359375</td>
</tr>
<tr>
<td>128</td>
<td>132</td>
<td>8.505859375</td>
<td>5.125</td>
<td>-3.380859375</td>
</tr>
<tr>
<td>256</td>
<td>260</td>
<td>16.63085938</td>
<td>9.125</td>
<td>-7.505859375</td>
</tr>
<tr>
<td>512</td>
<td>516</td>
<td>32.88085938</td>
<td>17.125</td>
<td>-15.75585938</td>
</tr>
<tr>
<td>1024</td>
<td>1028</td>
<td>65.38085938</td>
<td>33.125</td>
<td>-32.25585938</td>
</tr>
</tbody>
</table>

Meets or exceeds the latency expectations
# Key Metrics for PCIe® 6.0 Specification: Evaluation

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Expectations</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Rate</td>
<td>64 GT/s, PAM4 (double the bandwidth per pin every generation)</td>
<td>Meets</td>
</tr>
<tr>
<td>Latency</td>
<td>&lt;10ns adder for Transmitter + Receiver over 32.0 GT/s (including FEC) (We can not afford the 100ns FEC latency as n/w does with PAM-4)</td>
<td>Exceeds (Savings in latency with &lt;10ns for x1/ x2 cases)</td>
</tr>
<tr>
<td>Bandwidth Inefficiency</td>
<td>&lt;2 % adder over PCIe 5.0 specification across all payload sizes</td>
<td>Exceeds (getting &gt;2X bandwidth in most cases)</td>
</tr>
<tr>
<td>Reliability</td>
<td>0 &lt; FIT &lt;&lt; 1 for a x16 (FIT – Failure in Time, failures in 10⁸ hours)</td>
<td>Meets</td>
</tr>
<tr>
<td>Channel Reach</td>
<td>Similar to PCIe 5.0 specification under similar set up for Retimer(s) (maximum 2)</td>
<td>Meets</td>
</tr>
<tr>
<td>Power Efficiency</td>
<td>Better than PCIe 5.0 specification</td>
<td>Design dependent – expected to meet</td>
</tr>
<tr>
<td>Low Power</td>
<td>Similar entry/ exit latency for L1 low-power state</td>
<td>Design dependent – expected to meet; L0p looks promising</td>
</tr>
<tr>
<td></td>
<td>Addition of a new power state (L0p) to support scalable power consumption with bandwidth usage without interrupting traffic</td>
<td></td>
</tr>
<tr>
<td>Plug and Play</td>
<td>Fully backwards compatible with PCIe 1.x through PCIe 5.0 specification</td>
<td>Meets</td>
</tr>
<tr>
<td>Others</td>
<td>HVM-ready, cost-effective, scalable to hundreds of Lanes in a platform</td>
<td>Expected to Meet</td>
</tr>
</tbody>
</table>

Meets or exceeds requirements on all key metrics. Expect same results for 128.0 GT/s
Unordered I/O (UIO): QoS and path to Multi-Pathing

- PCI/PCIe enforce Producer/Consumer via fabric-enforced ordering rules
  - Problem: Limits performance
  - Problem: PCIe Posted Writes don’t match other SoC fabric semantics; Requester doesn’t (directly) know if/when the write has actually completed
  - Problem: Mismatched write performance to multiple destinations cause ~global stalls
- Relaxed Ordering (RO), and ID-Ordering (IDO) not commonly used
  - Problem: Still need “flag” operations to use PCI baseline ordering
  - Problem: RO/IDO not intended to support cases with multiple paths (see example at right)
- Goals:
  - Enable higher performance, esp. multiple-paths, via source-ordering
  - Fully backwards compatible with existing producer-consumer model
  - Simplest possible discovery/configuration

Example illustrating multiple paths between communicating devices future
UIO Details

- UIO on one or more dedicated non-VC0 channel
  - VC0 always for traditional ordering model
  - Reuse existing FC mechanisms: 5 transactions total – all new TTYPEs
    - “Posted”: UIO Memory writes (gets completion)
    - Non-Posted: UIO Memory Read
    - Completion: UIO Memory Read Completion with data, UIO Mem Rd Completion without data; UIO Memory Write completion (no data)

- Must be enabled end-to-end
- UIO and non-UIO don’t mix – different VCs also ensures QoS
  - E.g., Persistent Memory access vs regular memory access w/ UIO is two different VCs (non-0)
  - VCs are now easier/ cheaper to implement with shared credits
- No ordering – all ordering enforced at source (i.e., don’t do the flag write till all the data that is covered by it is completed)

<table>
<thead>
<tr>
<th>Row Pass Col?</th>
<th>UIO Write</th>
<th>UIO Read</th>
<th>UIO Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIO Write</td>
<td>Permitted</td>
<td>Permitted</td>
<td>Permitted</td>
</tr>
<tr>
<td>UIO Read</td>
<td>Permitted</td>
<td>Permitted</td>
<td>Permitted</td>
</tr>
<tr>
<td>UIO Completion</td>
<td>Yes</td>
<td>Yes</td>
<td>Permitted</td>
</tr>
</tbody>
</table>
PCI Express® Technology and Storage
PCIe® SSDs for Storage

- **PCI Express® architecture** is a great interface for SSDs
  - Stunning performance: 8 GB/s per lane/direction (PCIe 6.0 specification x1 @ 64.0 GT/s)
  - Lane scalability: 32/16 GB/s per device (x4/x2)
  - Lower latency: Platform + Adapter: 10 µsec down to 1 µsec
  - Lower power: No external SAS IOC saves 7-10 W
  - Lower cost: No external SAS IOC saves $
  - CPU-integrated PCIe lanes: Up to 128 PCIe 3.0 specification

- With NVM Express® and PCIe technology evolution, storage is no longer the bottleneck
Enterprise SSD Unit Shipment Forecast by Interface

RAS Features

- PCIe® architecture supports a very high-level set of Reliability, Availability, Serviceability (RAS) features
- All transactions protected by CRC-32 for non-Flit Mode and 6B FEC + 8B CRC for Flit Mode and Link level Retry, covering even dropped packets
- Error injection mechanism along with elaborate error logging in Flit Mode
- Transaction level time-out support (hierarchical)
- Well defined algorithm for different error scenarios
- Advanced Error Reporting mechanism
- Support for degraded link width / lower speed
- Support for hot-plug (planned and surprise)
DPC/ eDPC for RAS

- (enhanced) Downstream Port Containment (DPC and eDPC) for emerging usages
- Emerging PCIe® technology usage models are creating a need for improved error containment/recovery and support for asynchronous removal (a.k.a. hot-swap)
- Defines an error containment mechanism, automatically disabling a Link when an uncorrectable error is detected, preventing potential spread of corrupted data
- Reporting mechanism with Software capability to bring up the link after clean up
- Transaction details on a timeout recorded (side-effect of asynchronous removal)
- eDPC: Root-port specific programmable response to gracefully handle DPC downstream
**I/O Virtualization**

- Reduces System Cost and power
- Single Root I/O Virtualization Specification
  - Released September 2007
  - Allows for multiple Virtual Machines (VM) in a single Root Complex to share a PCI Express® (PCIe®) adapter
- An SR-IOV endpoint presents multiple Virtual Functions (VF) to a Virtual Machine Monitor (VMM)
  - VF allocated to VM => direct assignment
- Address Translation Services (ATS) supports:
  - Performance optimization for direct assignment of a Function to a Guest OS running on a Virtual Intermediary (Hypervisor)
- Page Request Interface (PRI) supports:
  - Functions that can raise a Page Fault
- Process Address Space ID enhancement to support Direct assignment of I/O to user space
PCIe® Specification Security Capabilities

- **Rationale:** Key assets warrant improved security
  - Consumers: data integrity, confidentiality
  - Businesses & suppliers: reputation, revenue-stream, intellectual property, business continuity
  - Governments: national security, defense, elections, infrastructure

- **Goal:** Define foundational security capabilities for a wide spectrum of systems / devices / components
  - PCIe technology has a broad reach: Smart phones, tablets, PCs, servers, switches / routers, processors, memory/storage/IO modules, IoT devices, vehicles and more
  - Build on industry developments to provide consistency across multiple technologies – PCIe, CXL, USB, etc.
    - Including DMTF’s (Distributed Management Task Force) Security Protocol and Data Model (SPDM) and Management Component Transport Protocol (MCTP) specifications
  - Build upon existing security standards (ISO, NIST, IEEE…) that are interconnect agnostic
  - Protect against multiple attacks: supply chain, physical, persistent, malicious components, etc.

Ack: Dave Harriman, Joe Cowan
PCI-SIG® & DMTF Specifications for Security

- **SPDM** defines a “toolkit” for authentication, measurement, and other security capabilities.
- **CMA/SPDM** defines how SPDM is applied to PCIe® devices/systems.
- **DOE** supports Data Object transport between host CPUs & PCIe components over PCIe technology.
- Various **MCTP** bindings support Data Object transport over different interconnects.
- **IDE** provides confidentiality, integrity, and replay protection for PCIe Transaction Layer Packets (TLPs).
- **TDISP** defines the security architecture and protocol device interface assignment to TEEs.

Ack: Dave Harriman
Form Factors
PCIe® Architecture: One Base Specification - Multiple Form Factors

- **BGA**: Smallest footprint (22mm x 30 to 110 mm): SSDs in boot slots, data center storage, WWAN
- **M.2**: Widely used in systems w/ 4 HL options. Higher Power. Robust compliance program
- **U.2 2.5in (aka SFF-8639)**: SSDs x4 or 2 x2 w/ hot-plug
- **CEM Add-in-card**: High B/W: hand-held, IoT, automotive
- **16x20 mm small and thin platforms**
- **Smallest footprint (22mm x 30 to 110 mm): SSDs in boot slots, data center storage, WWAN**
- **U.2 2.5in (aka SFF-8639)**: SSDs x4 or 2 x2 w/ hot-plug
- **CEM Add-in-card**: High B/W: hand-held, IoT, automotive

**E1.S (SFF-TA-1006)**
- (Up to 32 Modules)

**E1.L (SFF-TA-1007)**

**E3 Form-factors**
- E3.L (40W)
- E3.L 2T (70W)
- E3.S 2T (40W)
- E3.S (25W)

**Various Proprietary FFs for HPC Applications**
- Multi-KW cards

**Multiple Form-factors from the same silicon to meet the needs of different segments**
Cable Topology Support at Higher Speeds

Cable mitigates PCB loss limitations at 32+ GT/s and enables architectural flexibility
Internal and External Cable Topologies

Meg6-like PCB Loss at 16 GHz: ~ 1 dB/in
Cable assembly Loss at 16 GHz: ~0.1875 dB/in

A Typical Internal Cable Topology
(e.g., connecting a Riser/Backplane to the system board)

A Typical External Cable Topology
(e.g., connecting two boards within a rack)

Development of internal and external cable specs for 32 and 64 GT/s are work-in-progress

Acknowledgment: Debendra's Keynote at FMS 2022
(Rack level dis-aggregation with CXL/ PCIe® technology enabled by PCIe cables – Electrical and Optical)
Optical-friendly PCI Express® Technology
Problem Statement

- Optical has the promise of high bandwidth density and reach across a Rack/ Pod
  - Use case: Resource Pooling/ Sharing; Using PCIe® technology for developing composable systems with fabric topologies
  - Pros: Small (unlike copper cables which occupy more space) and reach (order of tens of meters vs Cu 1m/2m)
  - Cons: multiple technologies, cost vs copper – let this play out
- PCI-SIG® has launched an optical cable WG:
  - PHY Logical enhancements (comprehend sideband, mapping of mainband bit stream), including a Retimer-based approach
    - Enhancements to the Port / Pseudo-Port depending on the chosen implementation
  - Form-factor, if needed
  - Define in a way to enable on-package optics
- Assumptions:
  - Same type of Retimer is in both ends (optical technology neutrality)
  - EIC-PIC interface does not need to be defined
  - Complementary to copper cable work (different reach optimization)
Some Possible Implementations

Scope of PCI-SIG® Spec (for now)

SoC (RP)

PCIe Link

EIC

PIC

 optical Retimer

Platform Mgmt

SoC (EP/ Switch)

PCIe Link

EIC

PIC

 optical Retimer

Platform Mgmt

Optical Interconnect

Optical Retimer

MB

Riser card/ AIC

Optical

 SoC (RP)

PCIe® Link

Mgmt I/F

 Platform Mgmt

SoC (EP/ Switch)

PCIe Link

Mgmt SB

Platform Mgmt

MB

Riser card/ AIC

Optical

 Retimer

 Optical

 Can also have a copper cable connect the two Retimers (i.e., no optical)
 (Want no separate channel/ wire/ fiber for side-band after Retimer)
 (a: Retimer/ EIC/ PIC on riser/AIC)

RP

(CPO)

EP/Switch

RP

PIC

PIC

EP/Switch

(b: Retimer/ EIC/ PIC on package)

The optical or EIC-PIC signaling will not be spec’ed but the same Flit/ Ordered Sets defined can be used

SoC (EP/ Switch)

PCIe Link

Mgmt SB

Platform Mgmt

Board

MB

Riser card/ AIC

Optical

 Retimer

 Optical

 (Example System with discrete components)

Can also have a copper cable connect the two Retimers (i.e., no optical)
(Want no separate channel/ wire/ fiber for side-band after Retimer)
(a: Retimer/ EIC/ PIC on riser/AIC)

(c: Integrated EIC on package, Discrete PIC)

Same Base Spec ECN(s). Form-Factor can be looked at separately for various implementations

(Multiple permutations for on/off-package possible on either end: Port, Retimer/ EIC and PIC on 3, 2, 1 packages. Each side can have different levels of integration)
Compliance
PCI-SIG®: From Spec to Compliance

PCI-SIG® Specs

Describes
Requirements
Base specification and Form-Factor specs

C&I (Compliance and Interop) Test Specs
Describes Test criteria based on spec requirements
• Test Definitions
• Pass/Fail Criteria

Test H/W & S/W Validates Test criteria
• Compliance
• Interoperability

Test Tools, Fixtures And Procedures

Clear Test Output Maps
• Directly to Test Spec

Predictable path to design compliance

PASS
FAIL
Conclusions

▪ Single standard covering the entire compute continuum
▪ Predominant direct I/O interconnect from CPU with high bandwidth and used for alternate protocols with coherency and memory semantics
  ▪ Low-power, High-performance
▪ Currently working on 7th generation: 128 GT/s, PAM4, Same FEC/ CRC/ Retry mechanism as PCIe® 6.0 specification with full backward compatibility
  ▪ Expecting flat latency, high reliability, and improved power efficiency
▪ A robust and mature compliance and interoperability program
Please take a moment to rate this session.

Your feedback is important to us.