Embedded Systems Programming

PCIe – An Introduction
(Module 11)

Yann-Hang Lee
Arizona State University
yhlee@asu.edu
(480) 727-7507

Summer 2014
PCI Challenges

- **Limited Bandwidth**
  - PCI-X and Advanced Graphics Port (AGP) for higher frequency
  - Reduction of distance
- **Bandwidth shared between all devices**
- **Limited host pin-count**
- **Lack of support for real time data transfer**
- **Stringent routing rules**
- **Lack of scaling with frequency and voltage**
- **Absence of power management**
- **PCI-X** -- an enhancement of the 32-bit PCI Local Bus for a higher bandwidth demand.
  - a double-wide version of PCI, running at up to four times the clock speed
Inter-Networking Driving Demand

- Multimedia applications drive the need for fast, efficient processing of data over wired or wireless media
- CPU performance doubles about every 18 months while PC Bus performance doubles about every 3 years

Source: Intel

Real-time Systems Lab, Computer Science and Engineering, ASU
PCI Express Basics

- Serial, point-to-point, Low Voltage Differential Signaling
- 2.5GHz full duplex lanes (2.5Gb/s)
  - PCIe Gen 2 = 5Gb/s
- Scaleable links - x1, x4, x8, x16
- Packet based transaction protocol
- Software compatible but with higher speeds
- Built-in Quality of Service provisions
  - Virtual Channels
  - Traffic Classes
- Reliability, Availability and Serviceability
  - End-to-End CRC (Cyclic redundant checking)
  - Poison Packet
  - Native Hot Plug support
- Flow Control and advance error reporting

PCI Express Device 1

PCI Express Device 2

x4 Link Example

Ref Clock

Lane
## PCI Express Performance

<table>
<thead>
<tr>
<th>Link Width</th>
<th>X1</th>
<th>X2</th>
<th>X4</th>
<th>X8</th>
<th>X12</th>
<th>X16</th>
<th>x32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth in Gbits/s (Tx and Rx)</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>60</td>
<td>80</td>
<td>160</td>
</tr>
<tr>
<td>Throughput in GB/s (Tx and Rx)</td>
<td>.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Throughput in GB/s (per direction)</td>
<td>.25</td>
<td>.5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

**Raw:** Assuming 100% efficiency with no payload overhead.

- **PCI 32/66**
- **PCI or PCI-X 64/66**
- **PCI-X 64/133**
PCiE Layers

- Layered architecture
- Application Data transferred via packets
  - Transaction Layer Packet (TLP)
- PCiE core usually implement the lower three layers
- Protocol handling
  - connection establishing
  - link control
  - flow control
  - power management
  - error detection and reporting
PCIe TLP Structure

Information in core section of TLP comes from Software Layer / Device Core

- **Start Sequence**: 1B
- **Header**: 2B, 3-4 DW
- **Data Payload**: 0-1024 DW
- **ECRC**: 1DW
- **LCRC**: 1DW
- **End**: 1B

Created by Transaction Layer

Appended by Data Link Layer

Appended by Physical Layer

PCI-SIG Developers Conference  
Copyright © 2007, PCI-SIG, All Rights Reserved
Transaction Types, Address Spaces

- Request are translated to one of four transaction types by the Transaction Layer:
  - Memory Read or Memory Write. Used to transfer data from or to a memory mapped location
    - also supports a locked memory read transaction variant.
  - I/O Read or I/O Write. Used to transfer data from or to an I/O location
    - restricted to supporting legacy endpoint devices.
  - Configuration Read or Configuration Write – Used to discover device capabilities, program features, and check status in the 4KB PCI Express configuration space.
  - Messages. Handled like posted writes. Used for event signaling and general purpose messaging.
Programmed I/O Transaction

Requester:
- Step 1: Root Complex (requester) initiates Memory Read Request (MRd)
- Step 4: Root Complex receives CplID

Completer:
- Step 2: Endpoint (completer) receives MRd
- Step 3: Endpoint returns Completion with data (CplID)

PCI-SIG Developers Conference

Copyright © 2007, PCI-SIG, All Rights Reserved
Supplementary Slides
## Transaction Layer Packet Types

<table>
<thead>
<tr>
<th>Description</th>
<th>Abbreviated Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Read Request</td>
<td>MRd</td>
</tr>
<tr>
<td>Memory Read Request – Locked Access</td>
<td>MRdLk</td>
</tr>
<tr>
<td>Memory Write Request</td>
<td>MWr</td>
</tr>
<tr>
<td>IO Read Request</td>
<td>IORd</td>
</tr>
<tr>
<td>IO Write Request</td>
<td>IOWr</td>
</tr>
<tr>
<td>Configuration Read Request Type 0 and Type 1</td>
<td>CfgRd0, CfgRd1</td>
</tr>
<tr>
<td>Configuration Write Request Type 0 and Type 1</td>
<td>CfgWr0, CfgWr1</td>
</tr>
<tr>
<td>Message Request without Data Payload</td>
<td>Msg</td>
</tr>
<tr>
<td>Message Request with Data Payload</td>
<td>MsgD</td>
</tr>
<tr>
<td>Completion without Data (used for IO, configuration write completions and read completion with error completion status)</td>
<td>Cpl</td>
</tr>
<tr>
<td>Completion with Data (used for memory, IO and configuration read completions)</td>
<td>CplD</td>
</tr>
<tr>
<td>Completion for Locked Memory Read without Data (used for error status)</td>
<td>CplDLk</td>
</tr>
<tr>
<td>Completion for Locked Memory Read with Data</td>
<td>CplDLk</td>
</tr>
</tbody>
</table>