Disclaimer

Some of the information in this presentation refers to specifications still in the development process. This presentation reflects the current thinking of various PCI-SIG® workgroups, but all material is subject to change before the specifications are released.
Agenda

- PCIe® 3.1 Base Spec
- Newer PCIe 3.0 Protocol ECNs
- Protocol ECRs Under Development
Protocol Content Included In the PCIe 3.1 Base Spec

- Older Protocol ECNs Against the PCIe 3.0 Base Spec (not covered here)
  - Process Address Space ID (PASID)
  - 8.0GT/s Receiver Impedance
  - Lightweight Notification (LN) Protocol
  - Downstream Port Containment (DPC)
  - Enhanced DPC (eDPC)

- Newer Protocol ECNs Against the PCIe 3.0 Base Spec (covered here)
  - Separate Refclk Independent SSC (SRIS)
  - Precision Time Measurement (PTM)
  - L1 PM Substates with CLKREQ#
  - Readiness Notifications (RN)
  - NOP DLLP
  - Extension Devices
  - Enhanced Allocation (EA)
  - M-PCIe (PCIe Over M-PHY)

- Errata
Separate RefClk Independent SSC (SRIS)
Motivation

- Requirement: Need in-box low cost cabling to support existing physical partitionings (e.g. in desktop)

- Challenge: PCIe spec does not support independent clock with spread spectrum

- PCIe Base Spec changes needed to enable feature:
  1) Requires use of larger elasticity buffer
  2) Requires more frequent insertion of SKIP ordered set
  3) Requires overall system jitter budget modifications

New Terms: SRIS (5600ppm) and Separate RefClk With No SSC – SRNS (600ppm)
Precision Time Measurement (PTM)
PTM Motivation

- Precise Time Synchronization is a foundational technology enabling a broad array of applications
  - Continues to gain momentum in IEEE 1588-2008, 802.1AS, 802.11v …

- Required in many applications:
  - Synchronized A/V streaming (802.1 AVB)—studios, autos, and home
  - Instrumentation—distributed data acquisition, logic analyzers
  - Telecom—cell towers synchronized for seamless handoff
  - Industrial Automation/control—robots mustn’t crash into each other or hit people
  - Many others…

- PCIe is the ubiquitous “local bus” technology interconnecting IO controllers / Hosts
L1 PM Substates with CLKREQ#
L1 Power Management (PM) Substates Motivation

- Link idle power ~10% of Active
  - On the order of 10 mW per lane
- PCIe L1 Power does not meet stand-by power requirements for emerging thin and light form factor markets
- Regulatory requirements are driving down idle power across multiple market segments
- Platforms require idle power near zero
- Retain backwards compatibility
- Add minimum cost
## Power and Latency Solutions

<table>
<thead>
<tr>
<th>Sub-State</th>
<th>PLL</th>
<th>Rx/Tx</th>
<th>Common-Mode Keepers</th>
<th>x1 Port Power</th>
<th>Exit Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 (unmodified)</td>
<td>ON</td>
<td>off/idle</td>
<td>ON</td>
<td>25mW</td>
<td>2µs (retrain)</td>
</tr>
<tr>
<td>L1+CLKREQ (unmodified)</td>
<td>off</td>
<td>off/idle</td>
<td>ON</td>
<td>10mW</td>
<td>20µs (PLL)</td>
</tr>
<tr>
<td>L1.1</td>
<td>off</td>
<td>off</td>
<td>ON</td>
<td>300µW</td>
<td>20µs (PLL)</td>
</tr>
<tr>
<td>L1.2</td>
<td>off</td>
<td>off</td>
<td>off</td>
<td>10µW</td>
<td>70µs (Common mode restore + other delays)</td>
</tr>
</tbody>
</table>

**Solution:** Turn circuits off

*Note: Power savings will provide near linear scaling for multi-lane links.*

*These are targets for power and latency, not specified results.*
L1 PM Substates
High-Level Overview

- New power management substates of L1 Link state
  - Enables dramatically lower idle power by removing power from high-speed circuits
  - Applicable to both ASPM & PCI-PM L1 Link states
  - Entire ECN & various aspects of it are optional normative

- ECN history
  - Original ECR utilized CLKREQ# sideband signal for new purpose to manage L1 PM Substates
  - ECR later expanded to define a Low Frequency Activate Signaling (LFAS) inband mechanism to manage L1 PM Substates for platforms where CLKREQ# is not available
  - ECR later pulled back to CLKREQ# only, and eventually released as an ECN
  - The LFAS ECR has not progressed
Readiness Notifications (RN)
RN Motivation & Approach

- Avoid relatively long architected fixed delays following various forms of reset before software is permitted to perform its first Configuration access
  - **1 second** if Configuration Retry Status (CRS) is not used
  - **100ms** for most cases if CRS is used

- Avoid the complexity of using the CRS mechanism, potentially polling periodically up to 1 second following reset

- Specific cases for device or Function to become ready:
  - **Device** becoming ready following DL_Down to DL_Up
    - Exit from Cold Reset (initial power-up, hot-add, or D3cold)
    - Exit from Warm Reset, Hot Reset, Loopback, or Disabled
    - Exit from L2/L3 Ready
  - **Function** becoming ready following:
    - D3hot/D0 transition
    - FLR (Function-Level Reset)
    - Setting or Clearing of VF Enable in a Physical Function (SR-IOV)

- Approach: have the device or Function send a Message in these cases when it’s “Configuration-Ready”
FRS & DRS Messages

- Two distinct types of Messages
  - Device Ready Status (DRS)
  - Function Ready Status (FRS)
  - Both are PCI-SIG defined Type 1 VDMs with no payload
  - Receivers of these Messages ignore them if unrecognized

- DRS Message: uses local routing
  - Sent following DL_Down to DL_Up transition when device becomes ready for 1st Config access
  - In an MFD, all Functions (other than VFs) must be ready
  - Being “ready” also means device won’t send any CRS Completions

- FRS Message: uses route-to-root routing
  - Sent when an individual Function becomes ready
  - Requester ID indicates which Function became ready
  - FRS Reason field in Message indicates why Function became ready
RN Functionality in Different Components

- Functionality in Switch Upstream Ports & Endpoints
  - Ability to send **FRS & DRS Messages** Upstream for cases described earlier
  - **Readiness Time Reporting** capability indicates how long before a device or Function becomes Configuration Ready under certain cases

- Functionality in both RPs & Switch Downstream Ports
  - **Downstream Component Presence** field – indicates presence and whether a DRS Message has been received
  - **DRS Message Received** bit – indicates this event
  - **DRS to FRS Signaling Control** bit – enables Downstream Port to signal when it receives a DRS Message, by sending its own FRS Message or generating an interrupt

- Additional Functionality in RPs and RC Event Collectors
  - A queuing mechanism for received FRS Messages
  - Queue depth between 1 & 4095
  - Records Function ID & Reason Field for each FRS Message
  - Supports both interrupt and polling models
NOP DLLP
**NOP DLLP**
*(No Operation) DLLP*

- Assigns a reserved DLLP Type Encoding for “No Operation”
  - Existing transmitters should never transmit this encoding under normal operation, and receivers are required to silently discard any properly formed DLLPs received with this encoding.
  - Useful for Link Testing. Link tests can send a DLLP that’s required to be ignored by the receiver.

- Consolidates assigned DLLP Type Encodings into a single location
  - Documents MR-IOV encodings in Base spec

- Adds recommendations on the use of Vendor Specific DLLPs

---

**Table 3-1: DLLP Type Encodings**

<table>
<thead>
<tr>
<th>Encodings</th>
<th>DLLP Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>Ack</td>
</tr>
<tr>
<td>0000 0001</td>
<td>MRInit – See the MR-IOV Specification¹</td>
</tr>
<tr>
<td>0001 0000</td>
<td>Nak</td>
</tr>
<tr>
<td>0010 0000</td>
<td>PM_Enter_L1</td>
</tr>
<tr>
<td>0010 0011</td>
<td>PM_Enter_L23</td>
</tr>
<tr>
<td>0010 0100</td>
<td>PM_Active_State_Request_L1</td>
</tr>
<tr>
<td>0011 0000</td>
<td>PM_Request_Ack</td>
</tr>
<tr>
<td>0011 0001</td>
<td>Vendor Specific – Not used in normal operation</td>
</tr>
<tr>
<td>0100 0v2vYv0</td>
<td>InitFC1-P (v[2:0] specifies Virtual Channel)</td>
</tr>
<tr>
<td>0101 0v2vYv0</td>
<td>InitFC1-NP</td>
</tr>
<tr>
<td>0110 0v2vYv0</td>
<td>InitFC1-Cpl</td>
</tr>
<tr>
<td>0111 0v2vYv0</td>
<td>MRInitFC1 (v[2:0] specifies Virtual Link) – See the MR-IOV Specification²</td>
</tr>
<tr>
<td>1100 0v2vYv0</td>
<td>InitFC2-P</td>
</tr>
<tr>
<td>1101 0v2vYv0</td>
<td>InitFC2-NP</td>
</tr>
<tr>
<td>1110 0v2vYv0</td>
<td>InitFC2-Cpl</td>
</tr>
</tbody>
</table>
Extension Devices (Retimer)
Problem Statement

- At 8.0GT/s the channel loss budget does not satisfy all use cases
  - Cable
  - Backplane
  - Etc.

- PCI Express 3.0 TX EQ negotiation protocol makes extension device design complex – with significant potential for interoperability issues without a specification

- Solution: PCIe 3.0 ECN to define an extension device architecture that will guarantee interoperability with existing PCIe 3.0 compliant silicon (and up to worst case base spec compliant channels on either side of the extension device)
Extension Device Types

- Retimer (protocol aware)
  - Limited LTSSM device with no config space that drops out of the protocol after participating in TX EQ negotiation (end of phase 3)
  - Supports a full PCIe 3.0 Base Spec worst case channel in both directions

- Re-driver
  - Does not participate in TX EQ negotiation
  - Lower latency, lower active power
  - Channel supported in each direction is specific to the device silicon involved (MB and AIC), re-driver characteristics, and typically a full end-to-end channel less than twice a full PCIe 3.0 base spec worst case channel
    - Does not reset random jitter (adds some)
  - CEM TX EQ preset and dynamic change requirements cannot be met in all cases
Retimer Supported Topologies

Upstream Component (RC or Switch)
Downstream port

Upstream pseudo port
Retimer Component
Downstream pseudo port

Upstream port
Downstream Component (End point or switch)

Copyright © 2015, PCI-SIG, All Rights Reserved
Retimer Brief Overview

- Largely transparent to software
  - No type 0/1 header
  - Side band typical, but outside scope of ECR proposal
  - Retimer Presence Detected bit in Link Status 2 register
  - SW shall not enable L0s on any Link where a retimer is present
- Transparent to Data Link Layer/Transaction Layer
- Both ports of Retimer are always at same data rate
- Forwarding Mode (normal mode)
  - Ordered sets, TLPs, DLLPs, logical idle, electrical idle; are forwarded on each path
  - Some fields of Ordered sets are modified before forwarding
- Execution Mode (exceptions)
  - Retimer pseudo ports act as regular ports
  - Main example is phase 2/3 of the link EQ procedure
Enhanced Allocation
PCI Configuration Architecture

- Originally developed to support user-friendly “plug and play” – hardware self-description is now understood to have significant value at many levels
  - Simplifies and speeds system integration
  - Reduces system firmware/software cost/complexity
  - Improves system debug and robustness
  - And much more …

- The PCI Configuration Architecture has “stood the test of time”
  - We have been building on this base >20 years!
  - Other technologies just starting to provide level of capability PCI has provided from the start

- Q: How to retain value and extend the life of this architecture for 20 more years?
Problem Statement

- PCI Address Decode architecture is based on requirements of shared-bus physicals, and a comprehensive plug-n-play architecture
  - Decode is distributed
  - Addresses can be re-assigned at any time

- This does not map well to integrated environments such as SoCs or multi-chip modules
  - No value or requirement for arbitrary re/programmability
  - Many existing IP blocks for SoCs do not expect/support movable MMIO
  - Hardware and software must be designed for flexibility that is not required

- Result: Increased costs and complexity → As a technology, PCI is at a disadvantage in this space
Enhanced Allocation (EA) Capability Brief Overview

- An optional Conventional PCI Capability structure
  - Applicable to both Type 0 and Type 1 Functions
- Entries can indicate fixed I/O & memory ranges
  - Prefetchable or non-prefetchable memory ranges
  - Can relax historical 32b/64b associations with memory types
  - Can serve as a replacement for traditional BARs
- Supports new resource “type” definitions
- Future extensibility to support reprogrammable allocations
- Entries may correspond to “equivalent BARs”, but are not required to
- Functions relying on Enhanced Allocation typically will not operate correctly without the use of new software
M-PCle
Introduction – Changes

- Preserves the higher PCIe layers (TL, DLL)
  - Maintains PCIe programming model
  - 8b10b encoded
    - Special symbols like PCIe 2.5GT/s & 5GT/s mode

- Replaces PCIe PHY w/M-PHY
  - Modifications to Link Layer
  - New PHY interface RMMI
M-PCle LINK

- **LANE**
  - One TX pair connected to one RX pair.

- **SUB-LINK**
  - Consists of one or more LANEs in the same direction.
  - All the LANEs within a SUB-LINK operate at the same data rate.

- **LINK**
  - Consists of 2 SUB-LINKs, 1 in each direction.
  - SUB-LINKs operate at the same data rate.
  - SUB-LINKs are permitted to support different widths.
Introduction – Changes

PCle LTSSM

New M-PCle LTSSM
Configuration

- The M-PHY MODULE capabilities are discovered and MODULE attributes are configured
  - Support for software assisted discovery & configuration is supported (Configuration.Software)
- The Upstream component must initiate the Initial LINK Discovery, Configuration and Effectuate process
- Only TX-LANE(0) and RX-LANE(0) are operational and only in LS-MODE(PWM-G1)
- Local Register Configuration (LRC) is implementation specific
- Remote Register Configuration (RRC) must use Remote Register Access Protocol (RRAP) as defined in the ECN
Thank you for attending the PCI-SIG Developers Conference
Israel 2015

For more information please go to www.pcisig.com