PCI Express over IP - Accelerated

We are
a Silicon Valley based
technology company with
Offices in Germany. We are
Partner of leading electronic
device and solution providers
and have been enabling key
innovators in the automotive,
industrial, test & measurement
markets to build better
Embedded Systems, faster.

Our Mission is
To develop and market
technology solutions for
Embedded Systems
Realization via pre-validated IP
and expert application support,
and to combine off-the-shelf
FPGA devices with Open-Source
Software for dependable,
configurable Embedded
System platforms

Our Expertise is
I/O connectivity and
Acceleration of data
communication protocols,
additionally opening up FPGA
technology for analog applications,
and the integration and optimization
of Open Source Linux and
Android software stacks on
modern extensible processing
architectures.

Dr. Endric Schubert, Univ. Ulm & Missing Link Electronics
Andreas Braun, Missing Link Electronics
Ulrich Langenbach, Fraunhofer HHI
Introduction and Motivation

PCIe to Ethernet Throughput Matching

Max Throughput [Gb/s]

PCIe Generation

* Source: Adee Ran, Intel
† Reflects nominal data rates, not traffic patterns
PCI Express

- PCIe replaces the PCI Local Bus (backwards compatible)
- Full-duplex serial transmission
- At 8GT/s line rate (Gen3) on up to 32 lanes
- Packet-based protocol with four layers

**PCIe Layer**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Start</th>
<th>Seq.</th>
<th>Header</th>
<th>Payload</th>
<th>ECRC</th>
<th>LCRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Link</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Data Link layer, physical layer: Reliable transport on the link
- Transaction layer:
  - Transport of application data, device configuration, interrupts, Quality of service
  - TLP categories: Memory, I/O, configuration, message
PCI Express

- PCIe is point-to-point. Hierarchical system topologies via switches

- ID based routing (bus/device/function number) and address based routing
- Transactions require completion (split-transaction) or posted transactions possible
PCI Express

- PCI Local Bus from software view (addressing, driver, configuration, ...)
- PCIe devices implement a set of registers (configuration space)
- PCIe topology needs to be explored at the beginning of system start-up
- Enumeration of devices by completing Configuration-TLPs

**Range problem:** Physical line length of PCIe on PCBs is limited to the centimeter range
State-of-the-art

PCle external cabling:
- Standard for copper cables

InfiniBand:
- Standard HPC interconnect

ExpEther:
- PCle Gen2 over 10GbE networks by NEC.

Source: Nec Corporation

Source: Mellanox
TCP / IP

• 10 GigE will soon push from data center into embedded markets

Transporting 1 bit per second needs 1 Hz
• 1 GigE $\Rightarrow$ 1 CPU at 1 GHz
• 10 GigE $\Rightarrow$ 4 CPUs at 2.5 GHz
Network Processing

SoC FPGA as (yet) another computer

<table>
<thead>
<tr>
<th></th>
<th>Intel i7-4770</th>
<th>Xilinx Zynq 7045</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>~100 GFLOPS</td>
<td>5 GFLOPS (PS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>778 GFLOPS (PL)</td>
</tr>
<tr>
<td>TDP</td>
<td>84 W</td>
<td>&lt;20 W (typ)</td>
</tr>
</tbody>
</table>

SOC FPGA has 4x more compute
With ¼ the power dissipation!

Network protocol processing at application layer (ISO Layer 7) can more efficiently be implemented via a programming approach (in C or C++) than by digital circuit design (in VHDL or Verilog).
**PCIe over IP**

**PCIe Layer**
- Application
- Transaction
- Data Link
- Physical

**Transaction Layer Packet (TLP)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>4 B</td>
</tr>
<tr>
<td>Seq.</td>
<td>2 B</td>
</tr>
<tr>
<td>Header</td>
<td>12/16 B</td>
</tr>
<tr>
<td>Payload</td>
<td>128/256 B</td>
</tr>
<tr>
<td>ECRC</td>
<td>4 B</td>
</tr>
<tr>
<td>LCRC</td>
<td>4 B</td>
</tr>
</tbody>
</table>

**Network Layer**
- Application
- TCP
- IP
- MAC
- Ethernet

**TCP/IP Packet**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>8 B</td>
</tr>
<tr>
<td>SFD</td>
<td>1 B</td>
</tr>
<tr>
<td>Dst.MAC</td>
<td>6 B</td>
</tr>
<tr>
<td>Src.MAC</td>
<td>6 B</td>
</tr>
<tr>
<td>Type/Len</td>
<td>2 B</td>
</tr>
<tr>
<td>TLP</td>
<td>46 B – 1500 B</td>
</tr>
<tr>
<td>FCS</td>
<td>4 B</td>
</tr>
</tbody>
</table>
Concept of PCIe-over-IP

Distributed PCIe Switch based on “XPressRICH3” PCIe IP Core from PLDA
Implementation

FPGA1 – Upstream Side

Main-board

- PCIe PHY
- UP
- DN port edit
- Interface adapter + TLP-aggr.
  - Sideband con.
  - TLP con.
- NPAP
- 10GbE PHY

Server config

PERST#

FPGA2 – Downstream Side

Device

- PCIe PHY
- DP
- Config + Completer + Device edit
- Interface adapter + TLP-aggr.
  - Sideband con.
  - TLP con.
- NPAP
- 10GbE PHY

LAN
Implementation

Setup

Local

Remote

CPU

PCIe

RC

10GbE

ZC706

UP Port

FMC-to-PCIe

ZC706

DN Port

NVMe SSD

EP

PCIe
Implementation

**TLP aggregation**

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>02 00 00 00 00 11 02 00 00 00 00 55 08 00 45 30</td>
</tr>
<tr>
<td>00000010</td>
<td>05 c8 9a fa 40 00 ff 06 56 e6 c0 a8 01 b9 c0 a8</td>
</tr>
<tr>
<td>00000020</td>
<td>01 d5 ca 05 ca 06 fc 5a b4 d2 b5 45 28 44 50 18</td>
</tr>
<tr>
<td>00000030</td>
<td>01 e6 d3 79 00 00 00 00 00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>00000040</td>
<td>00 02 11 c7 4a 00 4b 05 c1 18 50 32 37 45 a9 a0</td>
</tr>
<tr>
<td>00000050</td>
<td>72 80 e9 d9 cb 1d 15 d4 b9 df 03 bb 23 05 82 ba</td>
</tr>
</tbody>
</table>

Legende:
- **Ethernet II Header**
  - [0000-0005] Dst. MAC: (02 00 00 00 00 11) → 02:00:00:00:00:11
  - [0006-0011] Src. MAC: (02 00 00 00 00 55) → 02:00:00:00:00:55
- **Internet Protocol Header**
  - [0017] Protocol: (06) → TCP
  - [001a-001d] Src. IP: (c0 a8 01 69) → 192.168.1.105
  - [001e-0021] Dst. IP: (c0 a8 01 65) → 192.168.1.101
- **Transmission Control Protocol Header**
  - [0022-0023] Src. Port: (ca 05) → 51717
  - [0024-0025] Dst. Port: (ca 06) → 51718
- **PCIe TLP Header**
  - [0036] FMT/Type: (60) → 64-bit Memory Write Request
  - [0038-0039] Length: (00 40) → 64 Doublewords (32-bit) → 256 Byte
  - [003a-003b] Requester ID: (03 00) → 03:0.0
  - [003e-0045] Address1: (00 00 00 02 11 c7 4a 00) 015e-0165
  - [0027-0028] Address2: (00 00 00 02 11 c7 4b 00)
  - [0039-0040] Address3: (00 00 00 02 11 c7 4c 00)
  - [003e-0045] Address4: (00 00 00 02 11 c7 4d 00)
  - [04be-04c5] Address4: (00 00 00 02 11 c7 4e 00)
- **Data**
- **Padding**
Implementation

Local switch

Distr. switch with TLP aggregation

Bandwidth [MB/s]

Blocksize [KiB]

Bandwidth [MB/s]

Blocksize [KiB]

Local

Remote

(c) 2016 MLEcorp.com

EW2016-Session18-MLE-PCIeOverIP 15
Conclusion

● Reliable “tunneling” of PCI Express via TCP/IP
● Fully transparent to PCIe Root Complex and Operating System
● FPGA processing enables bandwidth at 10 GigE line rates

● Based on “XPressRICH3” PCIe IP Core from PLDA

● Please visit us at the show:
  - HHI – Hall 4 Booth
  - MLE – Hall 2 Booth 2-421
  - Xilinx – Hall 1 Booth 1-205