ODSA PoC Software development outline

ODSA: Seeds of future silicon innovation

Sharing contributions from ODSA PoC team by Purush Gupta L B
Agenda

● What is PoC?
● PoC Hardware overview
● Software stack for PoC
● Applications being targeted / Use Case
● Long term roadmap
● Call to Action
Background

Open Domain Specific Architecture is an new open definition for Programmable devices optimized for specific applications or class of applications and made of modular pChiplet [collection of die in a single package]

Goal for ODSA to apply these highly integrated, multi functional devices to meet the demands of high-intensity workloads in the data center/edge/enterprise application – e.g. machine learning, video processing, accelerator, etc
Proof-Of-Concept Hardware

ODSA POC Pchipllets
- Broadcom NIC ASIC via OCP 3.0 interposer Pchipllet
- Lattice FPGA based Pchipllet for Management
  - 16G 1PCIE, Unisoc Host, 4x4 processing/Stacking, Configuration, Data Monitoring etc
- NXP Multi-core ARM Processor + Accelerator Pchipllet
Proof-Of-Concept Software Stack

- **Kernel/Bootloader** for the embedded processor
- **Peripheral Device Drivers:** The main SOC needs to communicate and interact with various IP subsystems part of other PChiplets
- **Host Device Drivers:** The Platform host device drivers interface needs to be lightweight and should be available for most devices.
- **Application:** In the end applications make or break the platform and test the capabilities of the frameworks. Fortunately we have identified quite a few that can highlight the unique capabilities
## Proof-Of-Concept Applications/Use-cases

<table>
<thead>
<tr>
<th>Use-cases</th>
<th>Demo highlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network traffic management</td>
<td>Demonstrate ability to pipe traffic directed to x86 to the NXP-LX2 and potentially scan for malicious packets</td>
</tr>
<tr>
<td>FTP (Data Transfer Node: DoE Content Distribution)</td>
<td>Build an dedicated FTP client/appliance that can be used to stage/accelerate the traffic between two network connections</td>
</tr>
<tr>
<td>Computational Storage + OVS Offload</td>
<td>Demonstrate a computational storage controller</td>
</tr>
<tr>
<td>NVMe + RoCE Demonstrator</td>
<td>Demonstrate a NVMeOF Target controller</td>
</tr>
<tr>
<td>HCI Controller</td>
<td>Demonstrate encapsulation of the storage and networking components to host system in a direct attached or network attached configuration.</td>
</tr>
</tbody>
</table>
Proof-Of-Concept Bring up plan

1. **Kernel / Bootloader**
   a. Start with NXP LX2 open source Layerscape SDK (LSDK)
   b. Update the DTS
   c. Configure the bootloader, Kernel with appropriate modules and enable trusted firmware

2. **Peripheral Device Drivers**
   a. Review the portability of the drivers for the peer adapters so they can enumerate under ARM (NXP LX2)
   b. Exercise the interfaces to validate the functionality

3. **Host Device Drivers**
   a. Review enumeration of the devices on the x86 host

4. **Application**
   a. Review the libraries required to stitch the application
**Proof-Of-Concept SW Milestones**

- **Phase 1**: Feature List / Requirements definition
- **Phase 2**: SW Architecture Definition
- **Phase 3**: Baseline SW / Linux bring-up on NXP PChiplet
- **Phase 4**: Other Hardware bring-up and attach to x86
- **Phase 5**: End to end functional validation, performance characterization
- **Phase 6**: Demo for SC20

* Tentative proposal based on volunteers availability/commitments. We need help on every stage of this project!
Call for action

- Looking for teams and individuals who are passionate about new HW, FW and SW concepts and participate in paradigm shifts on how Silicon is integrated and deployed.
- In need of Kernel, device-driver, hardware debug folks with lots of curiosity, eagerness to help
- We welcome varied experiences, perspectives and vision to make this a reality! Please join and spread the word.