O-DU Low

General Information

<table>
<thead>
<tr>
<th>O-RAN Distributed Unit Low Layers (ODULOW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTL</td>
</tr>
<tr>
<td>Zhimin Yuan</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Focus on L1 functional blocks development according to WGs specification.</td>
</tr>
<tr>
<td>Project</td>
</tr>
<tr>
<td>o-du, o-du/l2, o-du/phy, o-du/sim</td>
</tr>
<tr>
<td>Weekly Meetings</td>
</tr>
<tr>
<td>TBD, anyone is interested in O-DU Low project, would like to get more information, would like to know how to participant, please no hesitate to contact zhimin.yuan@intel or raise any questions through this wiki page.</td>
</tr>
</tbody>
</table>

Quick Explore:

Below link will help you have a quick explore

View current code contribution through web:

https://gerrit.o-ran-sc.org/r/gitweb?p=o-du/phy.git;a=tree

Explore the Documentation:

https://docs.o-ran-sc.org/en/latest/projects.html#o-ran-distributed-unit-low-layers-odulow

General Architecture

Current L1 functional development mainly focus on the open interface development, three interfaces will be involved to interact with L1 as below:

- Interface between L1 and Front haul, it will adopt the WG4 specification for the CUS plane communication.
- Interface between O-DU Low and O-DU High, FAPI interface will be adopt according to WG8 AAL specification.
- Interface between O-DU Low and accelerator, it's the hot topic in WG6 now, DPDK BBDev was adopted as previous design, it will follow the WG6 definition after WG6 specification is finalized.

For amber release, O-RAN FH library was contributed by Intel as seed code, BBDev API was already open source in DPDK community, please find detail API information in http://doc.dpdk.org/api/rte__bbdev_8h.html, FAPI API is under development.

O-RAN FH Lib is developed according to the O-RAN WG4 specification. It performs communication between the O-DU and RU and consists of multiple HW and SW components. Logical representation of HW and SW components is shown as below diagram:
From the hardware perspective, two networking ports are used to communicate to the Front Haul and Back (Mid) Haul network as well as to receive PTP synchronization. The system timer is used to provide “sense” of time to the gNB application.

From Software perspective the following components are used:

- Linux PTP provides synchronization of system timer to GPS time.
  - Ptp4l is used to synchronize oscillator on Network Interface Controller (NIC) to PTP GM.
  - Phc2sys is used to synchronize system timer to oscillator on NIC.
- DPDK to provide interface to Ethernet port.
- O-RAN FH library is built on top of DPDK to perform U-plane and C-plane functionality according to the O-RAN WG4 specification.
- 5G NR PHY(L1) uses the O-RAN FH library to access interface to RU. The interface between O-RAN FH library and 5G NR PHY is defined to communicate TTI event, symbol time, C-plane information as well as IQ sample data.
- 5G NR PHY communicates with L2+ application using the set of MAC/PHY APIs and the shared memory interface defined as WLS.
- L2+ in turn can use Back (Mid) Haul networking port to connect to CU unit in context of 3GPP specification.

Weekly meeting:

Now it's mainly ad-hoc meeting, record as below:

<table>
<thead>
<tr>
<th>date</th>
<th>Mar.16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendees</td>
<td>Intel: Zhimin</td>
</tr>
<tr>
<td></td>
<td>NTUST: Ray, Prof. Chung-An Shen, Prof. Chin-Ya Huang</td>
</tr>
</tbody>
</table>
Minutes:  
Action Items:
- Zhimin will provide
  - Model number of FPGA card such that Ray can invite the company to join this project
  - A list of job items for DPDK xRAN. Ray can pass the information to his colleague, Prof. Shen, for evaluation
- Sam has to check the supporting of NIC (and confirm it with Zhimin) and IEEE 1588
- Ray’s team will figure out how to reuse the API document generated by Doxygen and convert it to RST format

date: Mar. 8

Attendees: Intel: Zhimin
            NTUST: Ray, Prof.
            AAEON: SamChen; FredyHsu; EyckNi;

Minutes: Action Items:
1. Zhimin will provide
2. Sam has to check the supporting of NIC (and confirm it with Zhimin) and IEEE 1588
3. Ray’s team will figure out how to reuse the API document generated by Doxygen and convert it to RST format