ESD 2019 Seminar Topics

1. SDSoC on Zybo/Ultra96 Development Board (1-2 persons)
   a. HLS-based accelerator: you can choose any domain you want such as lane detection, object detection, AES encryption/description, machine learning, etc.
   b. Expectations
      i. Understand the hardware architecture generated by the tool
      ii. Understand the HW/SW interface and hardware driver generated by the tool
      iii. Perform design space exploration for the system by changing the HLS optimization parameters for performance/resource trade-off
   c. Language and tool
      i. HLS-compatible (high-level synthesis) C/C++
      ii. OpenCV library (if needed)
      iii. SDSoC tool with Vivado-HLS
      iv. Xilinx Deep Neural Network Development Kit (DNNDK) (if needed)
   d. Ref
      iii. https://github.com/Xilinx/SDSoC-Tutorials

2. PYNQ on Ultra96 board (1-2 persons)
   a. Binarized Neural Network (BNN) or Quantized NN (QNN) implementations on Ultra96 board
   b. Expectations
      i. Understand BNN/QNN operations and structure
      ii. Understand the HW/SW interface and hardware driver on PYNQ
      iii. Analyze the existing implementation variances of BNN/QNN in Ultra96 in terms of power, performance and resources consumption
   c. Language and tool
      i. HLS-compatible (high-level synthesis) C/C++
      ii. Python/Jupyter Notebook
      iii. SDSoC or Vivado/Vivado-HLS tool
   d. Ref
      i. https://github.com/Xilinx/BNN-PYNQ
      iii. https://github.com/Xilinx/PYNQ
      iv. https://github.com/Avnet/Ultra96-PYNQ
      vi. https://github.com/fpgasystems/spooNN
3. System simulation with Gem5 (2 persons)
   a. Simulate heterogeneous system, ARM Big.little architecture, using Gem5
   b. Expectations
      i. Understand Gem5
      ii. Understand the ARM Big.little architecture
      iii. Implement the ARM Big.little architecture simulation framework using Gem5
   c. Language and tool
      i. Python, C/C++
      ii. Gem5
   d. Ref
      i. [http://gem5.org/wiki/images/0/0e/ASPLOS2017_gem5_tutorial.pdf](http://gem5.org/wiki/images/0/0e/ASPLOS2017_gem5_tutorial.pdf)

4. Identifying Multiplier used in the HDL code using Yosys (1-2 persons)
   a. Given a list of HDL files (VHDL/Verilog), identify where the multipliers are implemented using Yosys
   b. Expectations
      i. After synthesizing, the multipliers might be optimized by the tool which is hard to identify, the goal is to parse the HDL files at the semantic level in order to extract the arithmetic operation represented by the “*” sign with the sizes (width) of the inputs
      ii. If time permit, replace each operation by a compatible approximate multiplier.
   c. Language and tool
      i. C/C++
      ii. Basic HDL
      iii. Yosys
   d. Ref
      i. [http://www.clifford.at/yosys/](http://www.clifford.at/yosys/)

5. System simulation with SystemC/SoCLib (2 persons)
   a. Investigate the TSAR (Tera-Scale ARchitecture) architecture and simulation environment using SoCLib.
   b. Expectation
      i. Understand SystemC/SoCLib simulation library
      ii. Understand TSAR architecture
      iii. Execute the TSAR project to understand the process and analyze the system with different configurations such as number of cores, cache, etc.
      iv. If time permits, replace the MIPS in TSAR by the ARM core
   c. Language and tool
      i. SystemC
      ii. SoCLib
   d. Ref
      i. [https://www-soc.lip6.fr/trac/tsar/](https://www-soc.lip6.fr/trac/tsar/)
      ii. [http://www.soclib.fr/trac/dev](http://www.soclib.fr/trac/dev)