Outline FPGA summer school

Day 1, Getting started FPGA day:

Keynote

Morning session (2 hrs): Getting started with FPGAs

- Hardware design on FPGA, starting with blinky
  - Synthesis, Implementation, Bitstream generation, demonstrating the FPGA design flow.
  - Performance estimation.
- Alternate approach of mapping the same design on FPGA (using Microblaze - V processors)

Lunch

Hands-on session(2 hrs):

- Hardware design session, where half of them pick hardware design (RTL + IP), other half pick the alternate approach.
- Comparison of the results, in terms of design time, effort, area, execution time.
- Bonus: Integrating a custom IP with Microblaze-V.

Day 2, Pynq day:

Keynote

Morning session (2 hrs): Introduction to custom overlays in Pynq

- What is PS, PL and SoC architecture, comparison to basic boards.

- What does python mean here, what does Pynq’s base.bit already have
- How to customise overlay (sobel.bit):
  - Designing the IP.
  - Making vivado connections, synthesis, loading to the board.
  - Performance benefits

Lunch

Hands-on session(2 hrs):

- Familiarity with setup - use default overlay
- Replicate and configure custom overlay (rest of the session)

Day 3, Kria day:

Keynote

Morning session (2 hrs):

- Comparison of Pynq, Kria applications and benefits.
- YOLO: https://github.com/Xilinx/Vitis-AI-Tutorials/blob/1.4/Design_Tutorials/07-yolov4-tutorial/readme.md
- LIDAR + Camera Fusion on KV260: https://github.com/Xilinx/Vitis-AI-Tutorials/tree/2.0/Tutorials/kv260_lidar_cam_fusion/

Lunch

Hands-on session(2 hrs):

- KV260
  - Built in applications (Smart camera, Defect detection, NLP SmartVision, AI Box with ReID)
  - FIR filter

    - YOLO: license plate detection, Fire detection, ppe detection
    - NES emulator for Xilinx KV260 FPGA board
    - DPU projects: Eye State Detection, Pose detection, 4d detection
    - Smart security camera (lidar might be needed)
- Exploration of KR260
    - Human Detection LIDAR

Day 4, Ultrascale+ day:

Keynote

Morning session (2 hrs): HLS on ZCU104

- Comparison of Pynq, Kria and Zynq ultrascale+, applications and benefits.
- High level synthesis of Digital Beamformer:
    - https://github.com/Xilinx/Vitis-Tutorials/tree/2024.1/Vitis_HLS/Design_Tutorials/02-Beamformer
    - Baseline synthesis.
    - Code analyzer to show the bottlenecks in the digital beamformer: https://github.com/Xilinx/Vitis-Tutorials/tree/2024.1/Vitis_HLS/Feature_Tutorials/01-using_code_analyzer
    - Micro-optimization involving pipeline, partition, unrolling: https://github.com/Xilinx/Vitis-Tutorials/tree/2024.1/Vitis_HLS/Feature_Tutorials/02-Beamformer_Analysis
- Comparing with x86 or ARM using the same C/C++ code.
- Note: This session can be followed by students as well

Lunch

Hands-on session(2 hrs):

- Introduction to AI profiler:

    - https://github.com/Xilinx/Vitis-AI-Tutorials/blob/1.4/Design_Tutorials/16-profiler_introduction/README.md
    - Run Resnet for image detection: https://github.com/Xilinx/Vitis-AI-Tutorials/tree/3.5/Tutorials/RESNET18/
    - Deep Processing Unit (DPU) and its usefulness.
    - Layer by layer profiling and analysis
- If resnet is run in Kria, Pynq, we compare the performance of zcu104