Hardware-Software Co-Design Assignment Report

Thierry Moreau, Luis Ceze
CSE548 (Computer Architecture)
June 7th 2017

Assignment Objectives & Constraints

Objectives:

- Explore hardware-software co-design methodologies for FPGA-equipped systems
- Understand and exploit accuracy/performance tradeoffs
- Identify system performance bottlenecks and tackle them accordingly

Constraints:

- Assignment can be completed under two weeks
- No prior FPGA experience or ML background is required

Assignment Overview

Part 1: Pipeline Optimization

- Familiarization with HLS pragmas, and identifying metrics that guide the optimization process
- Understanding optimization steps that improve design throughput

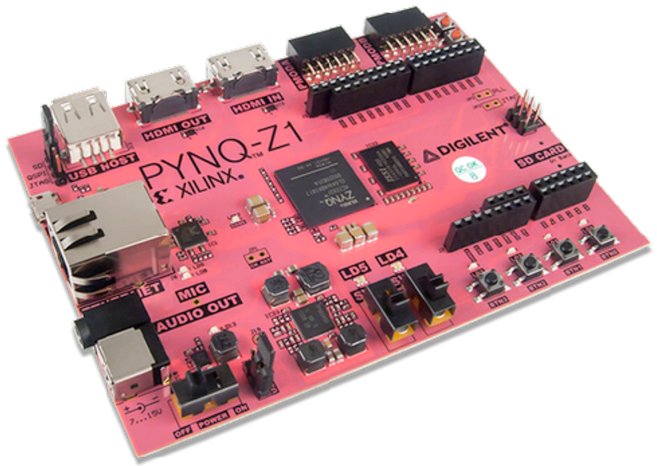
Part 2: Fixed-Point Optimization

- Identify performance/accuracy tradeoffs going from FP32 down to INT8
- Familiarize with data packing techniques to improve bandwidth
- Modification of a floating-point algorithm to work on fixed point computation

Part 3: Open-Ended Design Optimization

- Students have free range to explore software optimizations (improved training), hardware optimizations (more narrow fixed point types, input feature compression), or co-design techniques (different classifier algorithm that requires new hardware)

FPGA Hardware Platform



Target Platform: PYNQ $65 FPGA board
OS: Linux with Python libraries
Power: less than 2W
Storage: 630kB of on-chip storage
Memory: 4x64-bit channels to DRAM
Compute: 220 DSPs and 53k LUTs

Problem Statement

“Implement an FPGA-based inference accelerator that is both fast and accurate”

MNIST hand-written digit recognition dataset
http://yann.lecun.com/exdb/mnist/

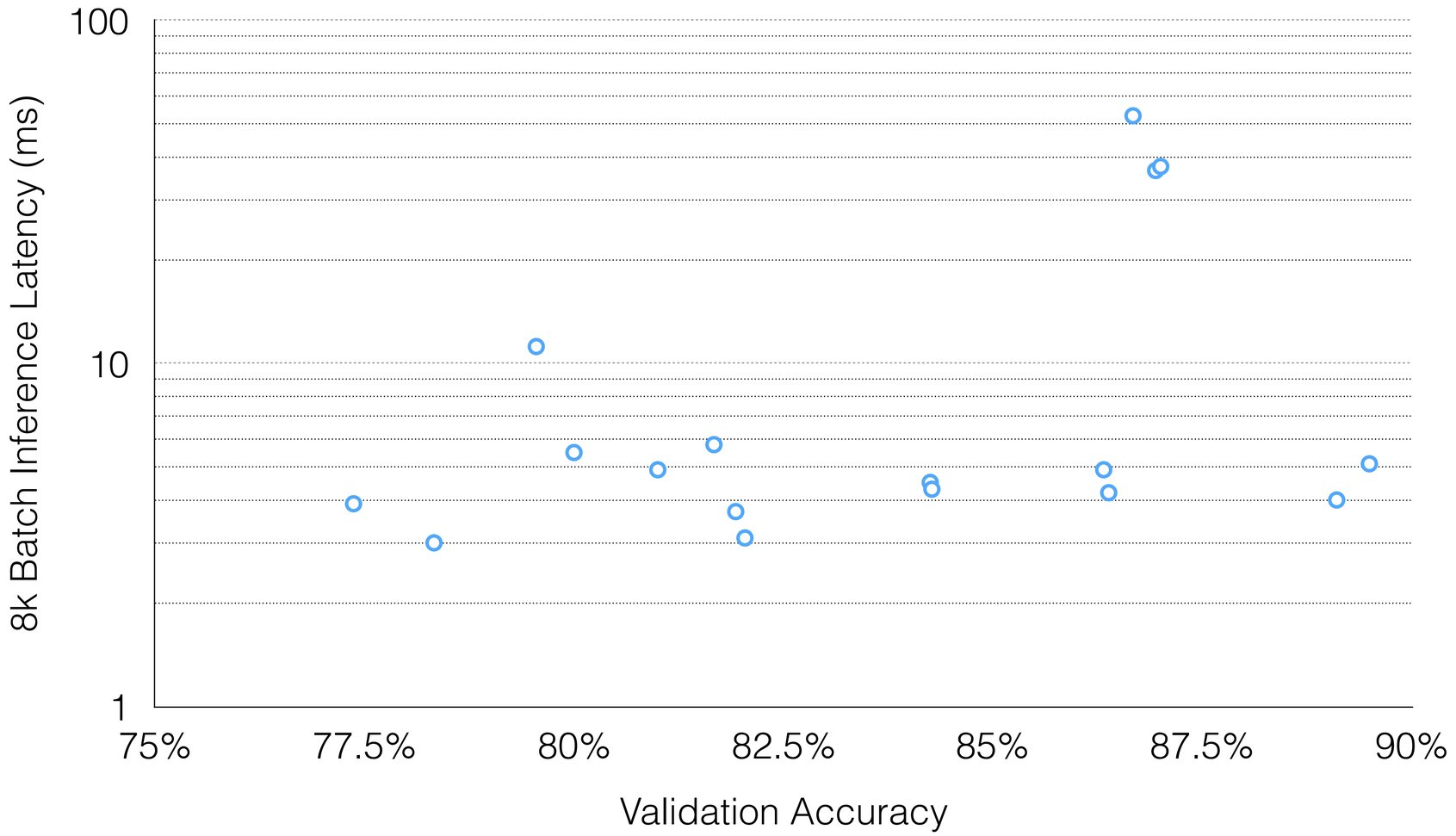
Submissions include artifacts to reproduce results.
Runtime and accuracies are measured on HW, on validation data

Source Overview

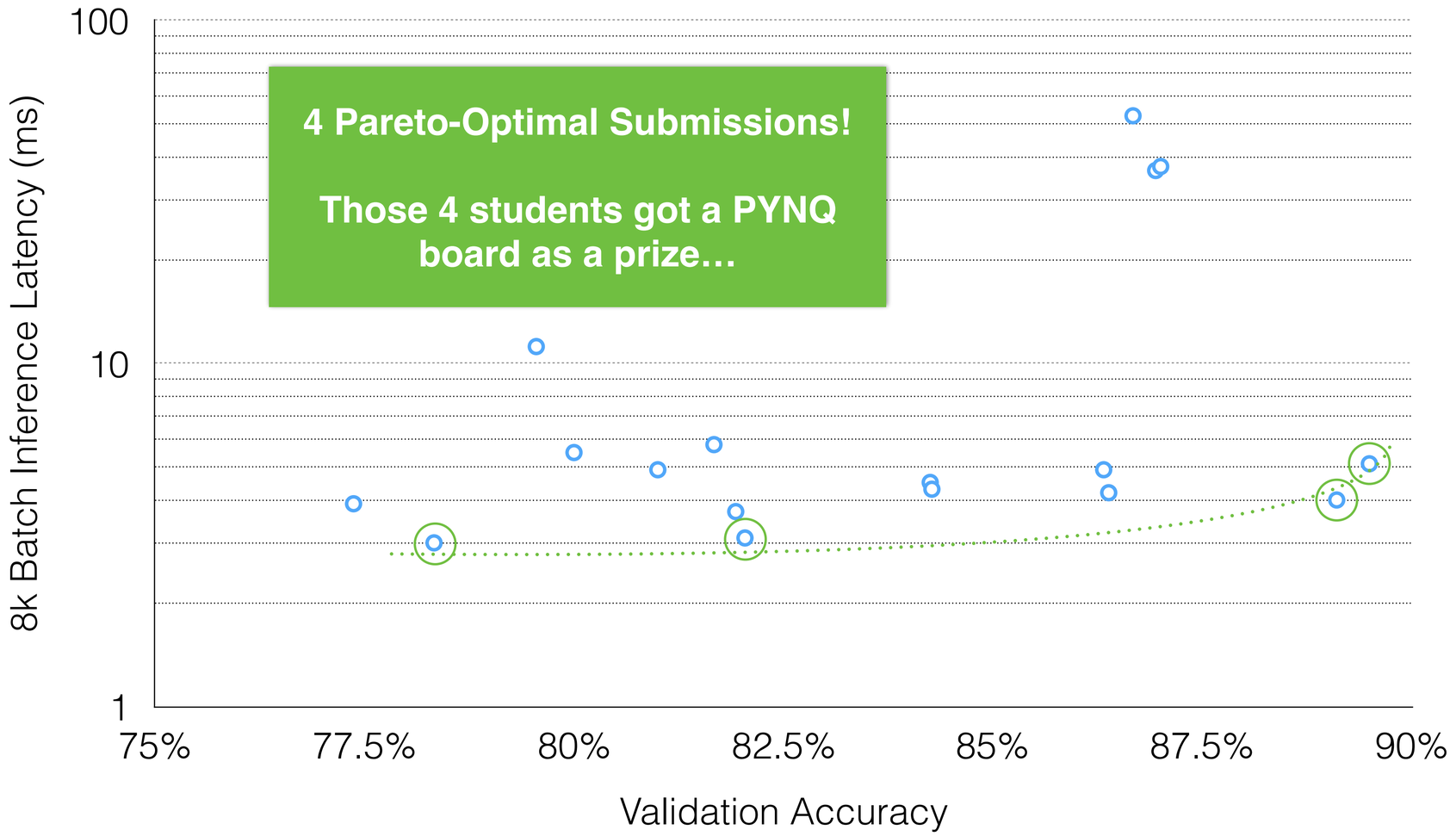
Instructions and source are all available on GitHub!

`zynq/hls:`	Accelerator c++ source and HLS compilation + test script
`zynq/tcl:`	Automated FPGA compilation script
`zynq/python:`	Customizable MNIST training script
`zynq/jupyter:`	iPython notebook files to run on the PYNQ to test FPGA classifier

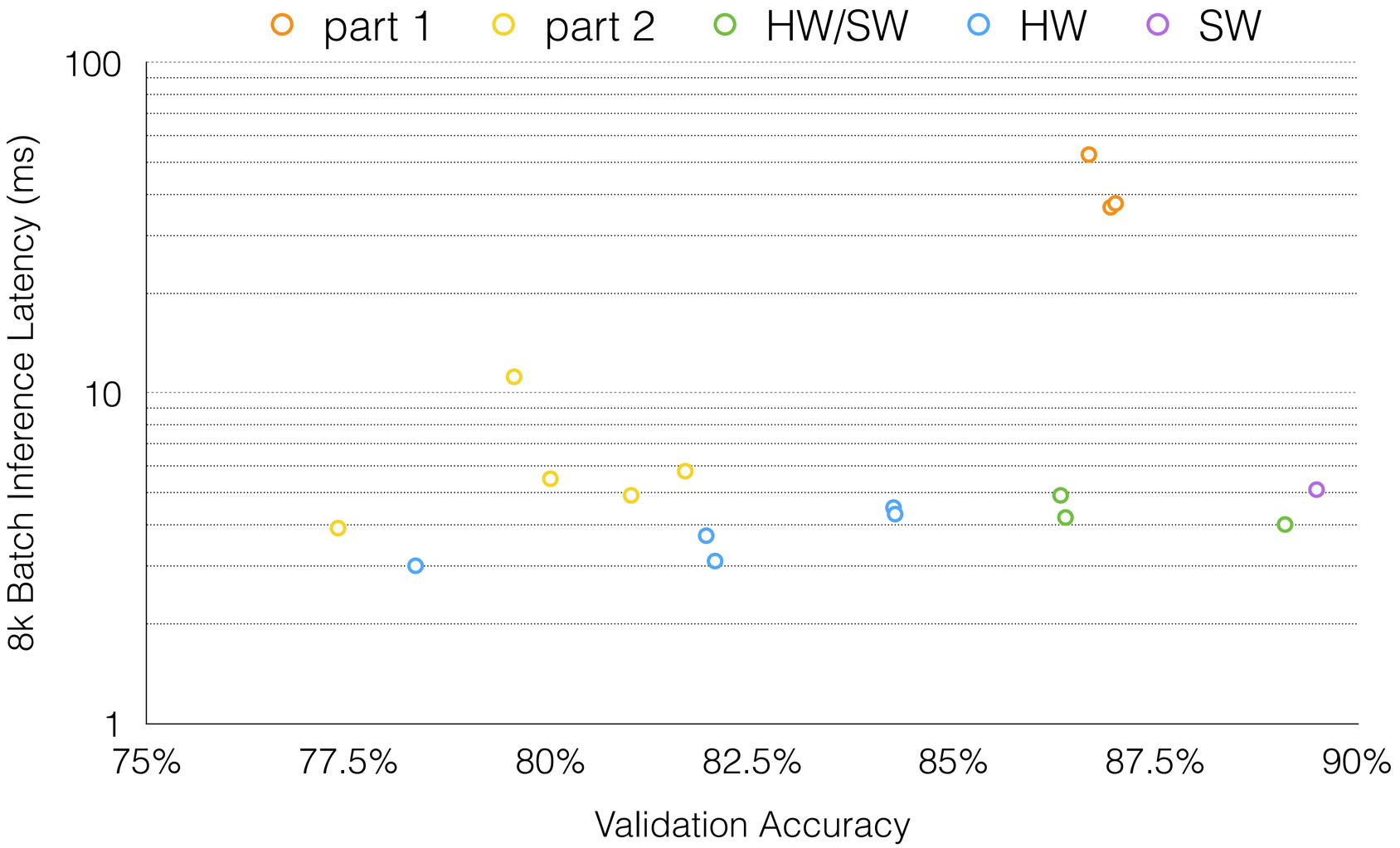
Submission Results



Pareto Line


4 Pareto-Optimal Submissions!
Those 4 students got a PYNQ board as a prize…
100
10
1
8k Batch Inference Latency (ms)
75%
77.5%
80%
82.5%
85%
87.5%
90%
Validation Accuracy

Optimization Attempts



Optimization Attempts

- [Part 1 & 2] Some students due to lack of time only attempted Parts 1 & 2 of the homework.
- [HW] Others managed to implement HW optimization only without changing the algorithm. Using `int4` or compressing the input from 256 down to 144 features was a common optimization.
- [SW] Another student attempted to implement a different learning algorithms on top of the same hardware. They replaced a linear classifier with a more accurate SVM.
- [SW+HW] Finally more ambitious students tried changing both the hardware and the classifier to implement simple multi-layer perceptrons. One MLP used XNOR net layers.

Next Steps

We are distilling large amount of feedback and will be updating into our GitHub repo

If you wish to deploy a similar lab in your class let us know!

moreau@cs.washington.edu
luisceze@cs.washington.edu