

**Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign**

**ECE 498 NSU - Deep Learning in Hardware  
Mini Design Project – Fixed Point 5-tap Linear Predictor**

**Assigned: 10/23/20**

**Due: 11/06/20**

**Fall 2020**

**Professor N. Shanbhag**

**Background:**

So far, we have talked about various topics in the lectures, including finite precision analysis and DNN training, such as LMS (least mean square) and SGD (stochastic gradient descent). In Homework 1, you used Wiener-Hopf method to compute the optimal weights and MSE for a 3-tap linear predictor. In Homework 2, you used SGD to train your weights and observed that they approached the optimal weights and so was the MSE. You also pipelined your architecture such that it achieves lower critical path delay. In Homework 3, you experimented with quantized gradient update, where you limited the precision used for backpropagation. You should have built background knowledge on how to build and train a network with finite precision and how to construct hardware mapping such that it has better performance.

**Project Description:**

In this mini project, your goal is to implement the 3-tap linear predictor from Homework 2 in hardware. Your task is to design the 3-tap linear predictor in fixed-point such that its MSE is within 0.5dB of the optimal MSE. You will be using Python as your software prototyping language and Verilog as your hardware prototyping language.

You will then have to implement this 3-tap fixed-point predictor in Verilog or SystemVerilog using Vivado, or any other applicable software that you want. All your implementation should be in fixed point and you need to be careful about the precisions in your design, namely the bit growth during computation and setting your output back to the desired precision. Once you finish coding and verify your functionality through simulation, i.e., making sure the Verilog output matches the output of the fixed-point Python code, you need to synthesize it and report the resource utilization, design latency, design energy, and schematic. Interpret the numbers available to you in the context of your design.

**Deliverables:**

- **Type report** for you hardware and software summary, including
  - Optimal solution and software design strategy, i.e, quantization.
  - Hardware design strategy and report of the resource utilization, timing, design power, and schematic.
- **Python Code** for your software prototyping
- **HDL Code** for your 3-tap fixed-point predictor (Verilog/SystemVerilog)
- **Behavioral simulation** that demonstrates your design correctness. You should show sufficient time steps and values to show how your design works.

**Timeline:**

- **Release:** 10/23/20
- **Due:** 11/06/20

**Grading:**

- **10%** Software Prototyping
- **60%** Hardware Prototyping
- **30%** Clarity and exposition of your written report.