SAMPLE SYLLABUS

## New York University Tandon School of Engineering

Electrical & Computer Engineering

ECE-GY 9413

Spring / 2021

Brandon Reagen

| Course Meeting Days | Course Meeting Times | Course Meeting Room |
|---|---|---|
| | | |

| | | | | |
|---|---|---|---|---|
| **To contact professor:** | bjr5@nyu.edu | Professor Phone | **Office:** | Professor Office |
| **Office Hours:** | Office Hours | | | |

Additional Office Hour Info

## Course Co/Prerequisites:

## Course Description:

This course will cover recent advances to computer design in response to the demise of Dennard's scaling. Since 2005, processor design has evolved to favor parallelism (multi-core CPUs and GPUs) and specialization (i.e., specialized hardware accelerators like the TPU) to combat the heat and power restrictions imposed by advanced process technology. A selection of papers will be read and discussed in detail to prepare students for this new era of computing designs.

## Course Objectives:

Understand the current issues facing high-performance computer design and the solutions being put forth. Be able to program parallel machines effectively (i.e., GPUs) and understand specialized hardware

## Course Structure:

Course will meet once a week and discuss three papers centered around a theme. Students will submit summaries of the paper before each course. The papers will first each be presented then their merit debated in a program committee like scenario. The professor will provide insights and summaries around the papers' influence.

## Readings:

### Required Text:

N/A

### Optional and Recommended Text:

## Grading Criteria & Course Requirements:

Paper summaries: 25 points

Discussion lead: 25 points

Programming project: 50 points

**Course Topic Outline**

[Date] Limits of ILP
1) “Limits of Instruction-Level Parallelism”, David Wall
2) “The Alpha 21246 Microprocessor”, Robert Kessler.
3) Optimizing Pipelines for Power and Performance”, Viji Srinivasan

[Date] Parallel Architectures
1) “Niagra: A 32-way Multithreaded SPARC Processor”, P. Kongetira
2) “How GPUs Work”, David Luebke
3) “The Vector-Thread Architecture”, Krste Asanovic

[Date] The Power Wall
1) “Conservation Cores: Reducing the Energy of Mature Computations”, Ganesh Venkatesh
2) “Near Threshold Computing: Overcoming Performance Degradation from Aggressive Voltage Scaling”, Ron Dreslinski
3) “Dark Silicon and the End of Multicore Scaling”, Hadi Esmaeilzadeh

[Date] The Need for Customization
1) “Understanding sources of inefficiency in general-purpose chips”, Rehan Hameed
3) “Aladdin: A Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures”, Sophia Shao
[
Date] Application Specific Architectures
1) “Anton, a Special-purpose Machine for Molecular Dynamics Simulation”, David Shaw
2) “Q100: The Architecture and Design of a Database Processing Unit”, Lisa Wu

[Date] CGRAs
1) “The Distributed Microarchitecture of the TRIPS Prototype Processor”, Karthikeyan Sankaralingam 2)
“Executing a Program on the MIT Tagged-Token Dataflow Architecture”, Arvind
3) “Dynamically Specialized Datapaths for Energy Efficient Computing”, Venkatraman Govindaraju

[Date] HW for ML: Part1
1) “Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks”, Yu-Hsin Chen
2) “Understanding Reuse, Performance, and Hardware Cost of DNN Dataflows: A Data-Centric Approach using MAESTRO”, Hyoukjun Kwon

[Date] HW for ML: Part 2
1) “In-Datacenter Performance Analysis of a Tensor Processing Unit”, Norman Jouppi.
2) “DNN Engine”, Paul Whatmough.

[Date] HW for ML: Part 3
1) “EIE: Efficient Inference Engine on Compressed DNN”, Song Han
2) “SCNN: An Accelerator for Compressed-sparse CNNs”, Angshuman Parashar
3) “MASR: A Modular Accelerator for Sparse RNNs”, Udit Gupta

[Date] Enabling Privacy
1) “Cheetah: Optimizing and Accelerating Homomorphic Encryption for Private Inference”, Brandon Reagen
2) “HEAX: An Architecture for Computing on Encrypted Data”, M. Sadegh Riazi

[Date] Code Review
Students will present their code for an industry-style review.

[Date] Performance Results
Students will present profiling data to understand the performance of their code.

# COURSE ASSESSMENT STATEMENT

**Course:** ECE-GY **Department:** Electrical & Computer Engineering

| | | | | |
|---|---|---|---|---|
| 1 | **OBJECTIVES**<br>*What will students learn?* | Understand the current issues facing high-performance computer design and the solutions being put forth. Be able to program parallel machines effectively (i.e., GPUs) and understand specialized hardware design. | | |
| 2 | **OUTCOMES**<br>*What, specifically, will the students know or be able to do upon completion of this course?* | Students will have read the most important papers in the field. They will also know how to program GPUs. | | |
| 3 | **EDUCATIONAL OPPORTUNITIES** (Pedagogy: Lecture, Lab, etc.)<br>*How will you accomplish each objective?* | They will be assigned a significant reading load each week and be expected to actively participate each week in the discussion.<br>The course project will be to implement homomorphic encryption kernels in CUDA to run on GPUs. Their performance results will indicate their mastery. | | |
| 4 | **ASSESSMENT MEASURES**<br>*How will you measure each of the objectives & outcomes in rows 1 & 2? (homework, exam question, project, presentation, etc.)* | Weekly writeups will be collected for each paper. They will have to lead discussions, showing deep understanding of papers. Finally, they'll have to implement GPU code and profile it. This will count for the majority of the grade. | | |