Jungi Lee

+82 10-5651-6428 ◇ jungi.lee@snu.ac.kr ◇ https://jungi-lee.github.io

RESEARCH INTERESTS

Computer Architecture, Accelerator for Emerging Workloads, Efficient AI Systems, Hardware-Software Co-Design

EDUCATION

Seoul National University, Electrical and Computer Engineering *Seoul, Republic of Korea*

M.S./Ph.D. in Electrical and Computer Engineering *03/2023 - Present*

- Computer Architecture and Systems Lab, advised by Prof. Jaewoong Sim.

B.S. in Electrical and Computer Engineering *03/2017 - 02/2023*

- GPA: 3.87/4.30, Major GPA: 3.98/4.30

RESEARCH EXPERIENCE

Graduate Research Assistant *03/2023 - Present*

Seoul National University *Seoul, Republic of Korea*

- Advisor: Prof. Jaewoong Sim
- Worked on extending the microscaling (MX) formats to enhance model performance for LLM inference.
  - Proposed **MX+**, a non-intrusive extension that represents the largest magnitude values in each block with higher precision while using the same bits as others.
  - Integrated MX+ into CUTLASS and the Triton compiler and demonstrated that it significantly improves model performance with a negligible slowdown.
  - Further reduced the performance overhead of software-based MX+ integration by incorporating architectural support into acceleration units such as Tensor Cores in NVIDIA GPUs.
- Worked on designing a key-value cache management framework for offloading-based LLM serving systems.
  - Designed an efficient attention speculation algorithm that amplifies a few channels in a matrix and uses them to speculate dot product results with almost no overhead.
  - Participated in designing **InfiniGen**, a key-value cache management framework that provides scalability under long-text generation by significantly reducing prefetching overhead.
- Worked on algorithm-hardware co-design to efficiently execute LLM inference.
  - Co-design channel grouping algorithm and hardware architecture to minimize accuracy and performance loss when using low-bit integer datatypes for LLMs.
  - Designed **Tender**, an algorithm-hardware co-design technique that quantizes activations and weights of LLMs down to a 4-bit integer with a much smaller accuracy loss than state-of-the-art solutions.
  - Implemented Tender and other accelerators using SystemVerilog and achieved significant area reduction over state-of-the-art accelerators.
- Worked on architectural support to accelerate an emerging graphic application.
  - Participated in designing the **GSCore** accelerator architecture that efficiently executes the rendering pipeline of 3D Gaussian Splatting.
  - Implemented an end-to-end C++ simulator with DRAM timing for functional *and* timing simulation and achieved a large speedup compared to the original GPU implementation with software optimizations.

PUBLICATION

* indicates equal contribution.

Wonbeom Lee*, **Jungi Lee***, Junghwan Seo, and Jaewoong Sim, InfiniGen: Efficient Generative Inference of Large Language Models with Dynamic KV Cache Management, in Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), 2024

Jungi Lee*, Wonbeom Lee*, and Jaewoong Sim, Tender: Accelerating Large Language Models via Tensor Decomposition and Runtime Requantization, in Proceedings of the 51st ACM/IEEE International Symposium on Computer Architecture (**ISCA**), 2024

Junseo Lee, Seokwon Lee, **Jungi Lee**, Junyong Park, and Jaewoong Sim, GSCore: Efficient Radiance Field Rendering via Architectural Support for 3D Gaussian Splatting, in Proceedings of the 2024 International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**), 2024

Junseo Lee, Kwanseok Choi, **Jungi Lee**, Seokwon Lee, Joonho Whangbo, and Jaewoong Sim, NeuRex: A Case for Neural Rendering Acceleration, in Proceedings of the 50th ACM/IEEE International Symposium on Computer Architecture (**ISCA**), 2023

TEACHING EXPERIENCE

Graduate Teaching Assistant *03/2023 - Present*
Seoul National University *Seoul, Republic of Korea*

- ECE 315.A - Digital Systems Design and Experiments Fall 2023
- ECE 322 - Computer Organization Spring 2023

Led lab sessions, answered questions about the class material, and graded labs/exams (ECE 315.A, ECE 322). Guided course project on building a CNN accelerator on FPGA (ECE 315.A).

SKILL

Programming Language: C/C++, Python, CUDA, Verilog, System Verilog, Unix/Linux

Tools: Pytorch, Intel Pin, TVM