Che (Randy) Chang

Email : che.chang@wisc.edu
Personal Website : https://randy1005.github.io/

Education

- **University of Wisconsin, Madison** — Madison, WI
  PhD in Electrical and Computer Engineering — *Aug. 2023 – Present*
  - **Research**: High-Performance Computing, Graph Algorithms
- **University of Utah** — Salt Lake City, UT
  Master of Entertainment Arts and Engineering — *Aug. 2020 – May. 2022*
- **National Cheng Kung University** — Tainan, Taiwan
  Bachelor of Science in Computer Science — *Sept. 2015 – June. 2019*

Experience

- **Student Researcher** — Google, Mountain View, CA
  AI Chips Team — *May 2024 – Aug. 2024*
  - Designed algorithms that parallelize SPICE simulation while pruning unnecessary input patterns to improve in-house timer performance.
  - Achieved up to **20× speedup** when analyzing timing on various designs.
- **Software Engineer Intern** — Intel Corporation, Santa Clara, CA
  Design Methodology and Automation Team — *May 2022 – Aug. 2022*
  - Utilized a constraint analyzer to automate FPGA designs debugging.
- **Software Testing Intern** — Collins Aerospace, Salt Lake City, UT
  Software Verification Team — *Feb. 2021 – Dec. 2021*
  - Built a regression test tracking system to automatically generate test plans for QA engineers.

Projects

- **G-PathGen: An Efficient GPU-parallel Critical Path Generation Algorithm**
  - Designed efficient GPU kernel algorithms for parallel critical path generation and dynamically adjusted the generated path count to maximize GPU utilization while minimizing redundant work.
  - Achieved **1.6–243.8× speedup** and **100% accuracy** compared to a state-of-the-art GPU solution when generating one million paths on large designs.
- **PathGen: An Efficient Parallel Critical Path Generation Algorithm**
  - Designed a multi-level queue scheduler to efficiently schedule parallel exploration of similar-priority paths.
  - Achieved **2.7–7.4× speedup** with 16 threads and **nearly 100% accuracy** compared to a state-of-the-art single-threaded timer when generating one million paths on large designs.
- **Ink: Efficient Incremental k-Critical Path Generation**
  - Designed an algorithm that efficiently identifies a set of paths to reuse for the next query and effectively prunes the path search space.
  - Achieved **5.2–22.4× speedup** and **20–31% less memory usage** compared to a state-of-the-art timer when generating one million paths on large designs.

Publications

- **PathGen: An Efficient Parallel Critical Path Generation Algorithm** — **ASP-DAC 2025**
  - **Che Chang**, Boyang Zhang, Cheng-Hsiang Chiu, Dian-Lun Lin, Yi-Hua Chung, Wan-Luan Lee, Zizheng Guo, Yibo Lin, Tsung-Wei Huang
  - **Best Paper Nominee (13 out of 169 accepted papers)**
- **Ink: Efficient Incremental *k*-Critical Path Generation** — **DAC 2024**
  - **Che Chang**, Tsung-Wei Huang, Dian-Lun Lin, Guannan Guo, Shiju Lin

Skills

- C++, CUDA