

# Matthew Ngaw

mngaw@andrew.cmu.edu · +1 978 831 0111 · [mattngaw.github.io](https://github.com/mattngaw) · [linkedin.com/in/mattngaw](https://linkedin.com/in/mattngaw)

## EDUCATION

M.S. in Electrical and Computer Engineering, Carnegie Mellon University  
Integrated Master and Bachelor Program

Expected May 2025  
Pittsburgh, PA

B.S. in Electrical and Computer Engineering, Carnegie Mellon University

May 2024  
Pittsburgh, PA

Relevant Courses    [Modern Computer Architecture](#)    [Logic Design & Verification](#)    [Compiler Design](#)  
\* Denotes a current course.    [Performance Analysis & Modeling](#)    [Reconfigurable Logic](#)    [Digital IC Design\\*](#)  
[Hardware Security\\*](#)    Non-Von Neumann Architecture\*

## RESEARCH EXPERIENCE

SPIRAL Research Group, Carnegie Mellon University

Aug 2023 - Present

Undergraduate Research Assistant, Advisors: Larry Tang and Prof. Franz Franchetti

Pittsburgh, PA

- Developing fault-tolerant and radiation-hardened accelerators in SystemVerilog
- Hardening floating-point units using interval arithmetic, mixed precision, and error analysis
- Implemented bit-precise floating-point arithmetic models in both SystemVerilog and C++

SPIRAL Research Group, Carnegie Mellon University

May 2022 - May 2023

Undergraduate Research Assistant, Advisors: Het Mankad and Prof. Franz Franchetti

Pittsburgh, PA

- Studied and tested algorithms for fast large-integer arithmetic (on the order of billions of digits)

## PROFESSIONAL EXPERIENCE

Arm Ltd.

May 2024 - Present

Hardware Architecture Intern

Austin, TX

- Modeling, benchmarking, and improving AMBA CHI

Siemens EDA

May 2023 - Aug 2023

Software Engineering Intern

Boston, MA

- Worked with team of 20 people to develop a distributed compiler in C++ for a Verilog emulation platform

## PROJECTS

[RISC-V Superscalar Core](#)    *Computer Architecture, SystemVerilog, VCS*

Spring 2023

- Designed and simulated a synthesizable 5-stage pipelined, 2-way superscalar CPU core implementing RV32IM

- Implemented hazard detection, data forwarding, branch prediction, and dual-issue logic
- Ranked 2nd among 24 lab groups in both categories (performance and performance per watt)

**Design Space Exploration of Systolic Array Multiplication** *Vitis HLS, FPGA, Hardware Design* Fall 2023

- Designed a systolic array matrix multiplication kernel on an AMD Zynq UltraScale+ using high-level synthesis
- Explored the design space across three dimensions to find the optimal parameters for the given hardware platform
- Achieved peak performance of 12.7 GOPS computing the product of two 4096x4096 matrices with 32-bit elements

**C0 Compiler** *OCaml, Compiler Design* Spring 2024

- Built from scratch a compiler that translates programs from C0 to x86-64 assembly
- C0 is a large subset of C including loops, functions, typedefs, structs, arrays, memory allocation
- Implemented analysis and optimization passes yielding execution time and code size improvements
- Implemented a mark-and-sweep garbage collector that precisely recollects all dead objects from the heap

**Branch Predictor Simulation & Analysis** *C++, Performance Analysis* Fall 2022

- Implemented simulations for 5 different branch predictors to explore cost-benefit tradeoffs
- Analyzed performance against the SPEC CPU 2017 Benchmark Suite using dynamic binary analysis

**Simple Network-on-Chip** *SystemVerilog, Handshaking, VCS* Spring 2024

- Designed a synthesizable NoC system consisting of nodes communicating through routers
- Implemented a round-robin arbiter for fairness, FIFO queues to optimize performance/cycle count

## PUBLICATIONS

Z. Gong, N. Zhu, M. Ngaw, J. Rivera, L. Tang, E. Tang, H. Mankad, F. Franchetti, “Interval Arithmetic-based FFT for Large Integer Multiplication”, IEEE High Performance Extreme Computing Conference (HPEC), 2022, Poster with extended abstract

## TEACHING EXPERIENCE

Undergraduate Teaching Assistant, Carnegie Mellon University

18-344: The Hardware-Software Interface

Fall 2023

18-213: Intro to Computer Systems

Fall 2022, Spring 2023

## SKILLS

Software: C, C++, Rust, OCaml, Python, Bash, GDB/LLDB

Hardware: SystemVerilog, Verilog, gem5, Synopsys VCS, Vitis HLS, Fusion 360 (PCB)

Misc: Git, Linux, LaTeX, vim, tmux