Jack Toubes

jt1065@princeton.edu | (310) 270-3512 | Princeton NJ, 08540

Research Focus

Electrical and Computer Engineering major with research experience spanning the heart of the computing stack, from microarchitecture to systems, **looking to develop more sustainable computer systems as a graduate student**, to slow the growth of computing related emissions.

Education

UC Berkeley, PhD in Electrical Engineering and Computer Science — Starting Sep 2025

Princeton University, BSE in Electrical and Computer Engineering — Sep 2021 – May 2025

- 2025 NSF GRFP Honorable Mention
- Overall GPA: 3.96/4.0,
- **Relevant Coursework:** Computer Architecture, Embedded Systems, Design of VLSI Systems, Principles of Computer System Design (systems programming), Contemporary Logic Design

Research Experience

Senior Thesis: Sustainable Heterogeneous Architectures, Advised by Prof. Margaret Martonosi — Sep 2024 – Present

- My work on sharing memory for sustainable hardware accelerators has been submitted for publication at MICRO 2025.
- This work centered on identifying sustainable ways to incorporate hardware accelerators into SoCs, by considering the benefits and drawbacks of reconfigurability and specialization.
- Though highly specialized fixed-function accelerators emit less operational carbon (consume less power and energy) than reconfigurable ones, the latter may be more sustainable, as they decrease area when time-shared for many kernels, leading to a decrease in the embodied (manufacturing related) emissions of an SoC.
- Similarly, accelerators which share memory hardware may be able to reduce emissions if the embodied emissions decreased by sharing memory offset the embodied and operational emissions overheads required for sharing.
- To find the ideal design point, I analyzed the sustainability of a spectrum of accelerator architectures, from homogeneous Coarse Grained Reconfigurable Arrays (CGRAs) (completely reconfigurable) to completely fixed-function domain specific accelerators (fully-specialized), including middle-ground design points like *heterogeneous* CGRAs, and fixed function compute with memory sharing (FFSM).
- My results demonstrated that sharing memory is a significant sustainability win because of the dominant role of SRAM in a contemeporary SoC's area, whereas sharing compute hardware is generally a less sustainable choice.

ILA-PPA, Advised by Prof. Sharad Malik — May 2024 – Sep 2024

- ILA-PPA is my own independent work, performed over the Summer of 2024.
- ILA-PPA enables power, performance, and area (PPA) estimation from a behavioral description of a hardware accelerator called an Instruction Level Abstraction (ILA).
- ILA-PPA seeks to enable earlier decision making, saving time and money when compared to making changes later in the accelerator development process.
- The ILA-PPA methodology schedules an accelerator's operations and assigns them hardware blocks according to user-provided configuration parameters. This schedule leads to a reasonable estimate of the hardware required to implement an accelerator, and how it needs to be connected.
- This hardware estimate is integrated with highly configurable PPA models in order to produce PPA estimates for the accelerator.
- Early results show that by taking advantage of the parallel structure of the ILA, ILA-PPA runs faster than similar tools which use sequential behavioral models, and orders of magnitude faster than HLS and RTL synthesis tools, making design space exploration more efficient.

ILAng-Aladdin, Advised by Prof. Sharad Malik Nov 2023 – May 2024

- ILAng-Aladdin was my own, junior year independent work (ECE 398 on my transcript).
- ILAng-Aladdin was a predecessor to ILA-PPA in function (PPA estimation from an ILA) and motivation.
- I developed a tool to generate an architecturally accurate C-language simulation of an accelerator from it's ILA, which can be fed to Aladdin (a pre-RTL PPA analyzer) for PPA estimation.
- I updated Aladdin's core to be compatible with LLVM-18 and C23, it's scheduler, analyzer, and PPA models to better match the ILA model of an accelerator, and fixed many lingering bugs in the Aladdin codebase.
- I also integrated the PPA results from Aladdin back into the ecosystem surrounding the ILA (this ecosystem is called ILAng), enabling entire workload estimation when combined with the existing ILAng functional simulator.

Encapsulated Functions, Advised by Prof. Amit Levy Nov 2023 – Present

- Encapsulated Functions has been accepted for publication at OSDI 2025. I will be listed as the second author.
- Encapsulated Functions fortifies single-process, cross programming language interactions (like foreign function interfaces).
- This fortification is necessary since, even with completely correct code, differences in the expectations of two interacting languages (for example what valid values a given type can contain) can lead to undefined behavior.
- The Encapsulated Functions paradigm involves using a novel type system and a swappable memory protection mechanism to carefully mediate cross-language interactions, in order to prevent one language's invariants from being invalidated by the other language's actions.
- I played a role in the development of our sample implementation's core, including it's dynamic library loading, and it's custom memory allocator. I was also responsible for most of the application based evaluation.
- I co-wrote a significant portion of the the Encapsulated Functions paper, and helped edit/revise the rest.

Industry Experience

Embedded Systems Research Intern, NASA's Jet Propulsion Laboratory (JPL) May 2023 – Aug 2023

- I developed the bare metal embedded software infrastructure for a JPL designed low-power single board computer (SBC) to prepare it for use in future flight projects.
- I designed, programmed, and tested hardware drivers, implemented a bare metal nand-flash filesystem, and interfaced the SBC with Nasa's F-prime (F') framework, for ease of future development.
- I built a proof of concept flight application demonstrating my software's capabilities which could be used as a template or a building point for future flight projects.

Engineering Intern, Mercury Systems Jun 2022 – Aug 2022

- I supported the systems, software, and electrical engineering teams with a variety of technical tasks, including writing software for internal metrics, tending to system requirements, and writing internal tests for transport layer networking software.

Teaching Experience

Teaching Assistant, Electronic Circuit Design, Analysis and Implementation Feb 2025 – May 2025
Feb 2024 – May 2024

- I helped students through technical lab assignments, including circuit construction, measurement, and analysis.
- I worked with students to understand circuit analysis using weekly practice and homework problems.

Teaching Assistant, Robotics and Autonomous Systems (Carlab) Sep 2024 – Dec 2024

- "Carlab" is Princeton's flagship junior year ECE class. In Carlab, students build a small, autonomously controlled car which is expected to perform certain tasks, such as controlling its speed and navigating around a course.
- I help students with both hardware and software debugging of their cars, brainstorming solutions to problems, and thinking of ideas for final projects. I also generally provide guidance to the ECE juniors.