Nathan Williams

nathanwilliams@utexas.edu | https://nate.town | linkedin.com/in/sizeru
Austin, TX

Education

The University of Texas at Austin *May 2027*
Bachelors / Masters | *Computer Science* *Austin, TX*
Minor | *Robotics*
Relevant Coursework: Advanced Computer Architecture | Advanced Operating Systems | Embedded System Design Lab | Energy-Efficient Computing | Virtualization | Theory of Computation | Graphics: Honors | Human-Computer Interaction | Wireless Networks | Programming for Performance | Data Management | Neural Nets

Experience

Operating Systems / Embedded Intern | *FUTO* *May 2025 - Present*
- Reverse-engineered chip microarchitecture optimizations on Intel Ultra, ARM Broadcom, and ARM Rockchip PEs.
- Board bringup and peripheral initialization using a custom WIP language for Raspberry Pi 4B, 3B, and Zero 2 W.
- Handwrote assembly to improve performance on critical loops improving language's memset by 1.9x on x64.
- Benchmarked language performance to compare against theoretical max to determine optimization candidates.
- Removed all external dependencies of new programming language by writing cross-architecture syscall library.

Advanced Computer Architecture TA | *UT Austin* *(Semesters) Aug 2023 - May 2025*
- Evaluated the performance of different modern CPU microarchitectural advancements.
- Verified novel RTL designs for CPUs on FPGAs.

Network Admin & Software Engineer | *Redi-Mix Concrete* *(Summers) May 2020 - August 2024*
- Crafted a full-stack web server written in Rust and client daemon written in C# to monitor concrete batch quality and provide customer insights which has recorded 6000+ concrete batches since 2022.
- Modernized a legacy POS program (written in 1991 using dBASE III+ on MS-DOS) which receives ~$100k in weekly transactions to allow for networking, syncing with the Sage accounting software, and automatic backups.

Research

Sumparators | *UT Austin* *May 2025*
A digital circuit for performing comparisons of sums with less latency than the state-of-the-art
- Synthesized, routed, and performing static timing analysis on a custom circuit design using open-source tools

The PANv0 ISA | *UT Austin* *Mar 2024*
A novel ISA which is portable between computers of any word size and maximizes performance without recompilation
- Designed an ISA with the goal of guaranteeing portability, backwards-compatibility, and forwards-compatibility
- Built an assembler for the ISA using Python which assembled a program hand-written in PANv0 assembly
- Benchmarked a PANv0 program running in a C simulator against other existing ISAs such as x86 and ARM.

Projects

Digital Piano | *Final Project at UT Austin* *August 2024*
A piano made from plywood, hall sensors, magnets, a custom PCB, two micro-controllers, and a custom RTOS

The 'Ozone' Processor | *Final Project at UT Austin* *April 2024*
A built-from-scratch out-of-order processor designed using SystemVerilog to run a subset of ~30 ARMv8 instructions

SaraScript | *Personal Project* *August 2023*
A declarative, secure-by-default scripting language written in Rust as a lightweight alternative to PHP (4K+ downloads)

Project Omelas | *Personal Project* *July 2023*
A project which advocates for digital privacy and ownership
- Self-host a website (built using SaraScript on OpenBSD) which holds all of the contents of Project Omelas.
- Provide publicly accessible tutorials on building cheap, low-maintenance, and secure personal web services

Skills

Programming Languages: C | C++ | Rust | Verilog | x86 | ARM | JavaScript | HTML | CSS | SQL | GLSL | Python
Tools / Frameworks: AWS | FPGA | NGINX | Vulkan | OpenGL | OpenBSD | .NET | Qt | CMake | dBASE III+ | Godot