Wesley Wu

Naperville, IL | (630) 946-4126 | wesleywu2002@gmail.com | LinkedIn | GitHub

Education

University of Illinois Urbana-Champaign
Master of Science in Electrical and Computer Engineering | Focus: Computer Architecture | GPA: 3.85/4.0 | Dec 2025
Bachelor of Science in Computer Engineering | GPA: 3.78/4.0 | Dec 2023

Relevant Coursework: Parallel Computer Architecture, Advanced Computer Architecture, Accelerator Architectures, Digital Systems Lab (FPGA), VLSI System Design, Operating Systems, Operating System Design, Networking, IC Device Theory & Fabrication

Technical Skills

- **Languages**: SystemVerilog (RTL Design), UVM, C, C++, TCL, CUDA, Python, Java, x86 ASM
- **Tools:** Synopsys VCS, Verdi, DVE, Altera Quartus, ModelSim, Cadence Virtuoso, Innovus, vManager, Linux
- **Workflow**: Git, GitHub, Gerrit, Jira, Agile, ClearCase

Work Experience

Arm — **Chandler, AZ**

System IP Engineer Intern — *May 2025-Aug 2025*
- Collaborating with RTL team to identify and validate critical CoreLink RTL bug fixes, improving system stability
- Porting in-house coverage methodology for new CoreLink microarchitectural feature to enable targeted testing

SoC Validation Engineering Intern — *May 2024-Aug 2024*
- Architected a remote thermal device controller API, streamlining user interactions and simplifying test execution
- Developed a tkinter GUI with live temperature plotting, enhancing user control and real-time monitoring
- Built a Python library for automation, enabling efficient batch/overnight testing with minimal manual effort
- Implemented a feedback loop to ensure precise maintenance of thermal testing device and DUT temperatures
- Designed a local buffer for temperature readings, allowing simultaneous live monitoring by multiple users

Qualcomm — **San Diego, CA**

SoC APSS Integration DV Intern — *May 2023-Aug 2023*
- Incorporated functional coverage using SystemVerilog/UVM, enhancing quality of mobile SoC design verification
- Developed verification plans for functional coverage, ensuring the SoC design meets end user requirements
- Identified coverage holes in test suites, accelerating SoC design exploit detection and back-end design processes

Motorola Solutions — **Schaumburg, IL**

Embedded Android Engineer Co-Op — *Nov 2022-May 2023*
- Implemented dump function in C for fastboot protocol, enabling user to check any partition image for corruption
- Added special fastboot commands, enabling developers to change flag bits in system images without remounting
- Updated fastbootd user-mode display, giving developers more useful information about their current device
- Enabled flashing lock/unlock for fused devices, allowing devs to flash various OS images in fastbootd user mode

Android Platform Software Engineering Intern — *May 2022-Aug 2022*
- Devised logic in C to prevent rollback to old, vulnerable firmware or maliciously modifying vital image data
- Upgraded pre-existing preflash validation in C, ensuring the device doesn't brick in fastboot mode or during boot
- Prevented device bricking in bootloader stage, allowing user to recover device in the case of image data corruption

Projects

oPOSsum - SAT Solver Accelerator | *SystemVerilog, Synopsys VCS + Verdi, Cadence Virtuoso + Innovus*
- Architecting a control-driven SAT solver accelerator with parallel functional units and custom interconnect
- Driving full RTL-to-GDSII ASIC flow targeting TSMC 65nm process on a 1mm^2 die, with planned tapeout

Homomorphic Encryption Accelerator | *SystemVerilog, Synopsys VCS + Verdi, C*
- Designed a hardware accelerator to speed up encrypted computation, focusing on encrypted-space multiplication
- Drafted top-down design of accelerator, from block design to data flow to architecture of low-level functional units
- Observed 16x speedup over pipelined RISC-V CPU, tested using a workload kernel written in C compiled to binary
- Implemented each layer of the design hierarchy in SystemVerilog and verified layers using Synopsys VCS & Verdi

Pipelined RISC-V Processor | *SystemVerilog, Synopsys VCS + Verdi*
- Created 5-stage in-order pipeline with data forwarding, split L1 cache, cache arbiter, and static branch prediction
- Designed, verified, and integrated a two-way set-associative cache in SystemVerilog for split L1 cache
- Verified pipelined CPU design functionality using RVFI and Synopsys toolchain, including Synopsys VCS & Verdi

Custom RISC-V CPU Layout | *Cadence Virtuoso (Layout XL), Innovus, TCL*
- Performed custom layout of the datapath of a bitsliced RV32I-compliant single-cycle CPU, using only up to metal6
- Built and used custom standard cell library layouts to design all datapath components, resulting in lower area
- Automated layout of the CPU's control unit by effectively utilizing TCL scripting and Cadence Innovus