

# Ari Manjikian

US Permanent Resident | 404-654-0033 | [amanjikian3@gatech.edu](mailto:amanjikian3@gatech.edu) | [linkedin.com/in/arimanjikian](https://www.linkedin.com/in/arimanjikian) | [github.com/arimanjikian](https://github.com/arimanjikian)

## EDUCATION

---

### Georgia Institute of Technology

*B.S. Electrical Engineering, GPA: 4.0*

Atlanta, GA

Aug 2025 – May 2028

### State College of Florida

*A.A. Physics, GPA: 4.0*

Sarasota, FL

Aug 2024 – May 2025

## EXPERIENCE

---

### FPGA Undergraduate Researcher

Jan 2026 – Present

*SharcLab @ Georgia Tech*

- Extended the open-source HLSFactory framework, used for design-space exploration, by implementing 10 new Spector-HLS hardware designs, enabling more comprehensive evaluation of synthesis and optimization strategies.
- Added and validated backend support for Xilinx Vivado and Vitis toolchains from 2023 to 2025, enabling HLSFactory designs to synthesize and run across multiple FPGA toolchain versions.
- Used HLSFactory's vendor-agnostic OptDSL frontend to define optimization directive combinations and expand the benchmark suite from 10 to 22 synthesizable HLS designs.

### Digital Design Engineer

Aug 2025 – Present

*SiliconJackets @ Georgia Tech*

- Contributed to the design of a superscalar RISC-V processor supporting the RV32I, M, and F extensions, developed by a 200+ member student team and taped out on the SkyWater 130nm open-source ASIC process.
- Implemented SystemVerilog modules within the commit and system functional group and developed CSR logic to expose pipeline performance signals, enabling visibility into module behavior and identification of bottlenecks.
- Collaborated with the physical design team during implementation and signoff, validating clocking, reset strategy, and timing closure for silicon operating frequencies approaching 200 MHz.

## PROJECTS

---

### FPGA VGA Pong Engine | *SystemVerilog, Lattice iCE40HX1K*

Jan 2026

- Implemented a cycle-accurate 640×480@60Hz VGA controller generating HSYNC/VSYNC and pixel coordinates via 800×525 timing counters and driving a 25 MHz streaming pixel pipeline.
- Designed a combinational hardware renderer computing paddle, ball, and boundary pixels per cycle without framebuffer storage, reducing memory usage to 0 BRAM while sustaining continuous video output.
- Synthesized to 847 LUTs and 389 flip-flops on a Lattice iCE40HX1K FPGA (73% logic utilization) and met timing at 99 MHz, providing ~4× margin over the VGA pixel clock.

### Advent of FPGA Challenge by Jane Street | *Hardcaml, OCaml, RTL Design*

Dec 2025

- Participated in Jane Street's Advent of FPGA challenge, implementing a hardware solution in Hardcaml by modeling registers, combinational datapaths, and control logic to solve a logic-driven streaming problem.
- Constructed cycle-accurate RTL by utilizing Hardcaml primitives for state, arithmetic, and conditional logic, and generated synthesizable Verilog directly from the functional design.
- Developed a verification harness using Hardcaml's testing framework to perform cycle-by-cycle simulation, generate waveforms, and validate signal-level behavior against expected outputs.

### RISC-V Single-Cycle CPU Core | *SystemVerilog, cocotb, Verilator*

Nov 2025

- Designed and implemented a single-cycle RISC-V processor core supporting the full RV32I instruction set, including ALU ops, branching, and memory access.
- Built a cocotb verification framework and ran comprehensive instruction-level testing in Verilator, reaching 99.4% code coverage and uncovering multiple decode and control-logic edge cases.

## TECHNICAL SKILLS

---

**Programming & Scripting Languages:** Python, C, C++, Tcl, Perl, Golang, JavaScript (ES6+)

**Hardware Design:** SystemVerilog, Verilog, VHDL, RISC-V, ASIC Design, VLSI, FPGA Design

**Verification:** UVM, Coverage-Driven Verification, cocotb, Cycle-Accurate Simulation, Waveform Debugging

**Developer Software:** Git, Linux, Xilinx, Vivado, Vitis, Innovus, SimVision, Verdi, MATLAB

**Timing & Implementation:** Static Timing Analysis, Clocking & Reset Strategies, Floorplanning Collaboration