Armaan Mathur

647-219-3651 | a65mathu@uwaterloo.ca | linkedin.com/in/armaan-mathur | github.com/armaanmxt

Education

University of Waterloo **Sep. 2025 – 2030**

Bachelor of Applied Science in Nanotechnology Engineering | President's Scholarship *Waterloo, ON*

Technical Skills

RTL & Digital Design: SystemVerilog, Verilog, Synthesizable RTL, FPGA Design, Digital Logic, FSMs

Verification: cocotb, Self-Checking Testbenches, SystemVerilog Assertions, RTL Simulation, Waveform Debugging

Programming & Automation: Python, Git, RTL Regression Automation

EDA Tools: AMD Vivado, Yosys, Icarus Verilog, GTKWave

Interfaces & Hardware: Ethernet, SPI, PWM, Clock Domain Crossing (CDC), Basys 3 (AMD Xilinx Artix-7)

Experience

RTL Design & Verification Member **Apr. 2026 – Present**

ASIC Chip Design Team (UWASIC) *University of Waterloo*

- Developed **synthesizable SystemVerilog RTL** for a **10 Gigabit Ethernet Physical Coding Sublayer (PCS)** encoder, implementing datapath and control logic to convert 64-bit XGMII data and control inputs into 66-bit blocks.
- Implemented **64b/66b encoding** across **14 data and control block formats**, including sync headers, control-code mapping, start/terminate handling, and deterministic error handling for invalid inputs.
- Built **self-checking SystemVerilog testbenches** with directed and boundary-case tests to verify encoder outputs and **encoder–decoder round-trip behavior**, including terminate-lane positions, byte alignment, and malformed inputs.
- Debugged **RTL integration** across PCS, MAC, and packet-parser blocks using **RTL simulation and waveform analysis**, identifying data-alignment, control-signal, and interface mismatches; reviewed SystemVerilog changes through Git.

Design Engineering Intern **May 2026 – Aug. 2026**

Ontario Power Generation (OPG) *Pickering, ON*

- Reviewed and **redlined electrical schematics** and mechanical drawings for components associated with pressure tubes, fuel bundles, and steam generators, supporting the Pickering Refurbishment.
- Developed and closed a **Temporary Modification (TMOD)** through OPG's **Engineering Change Control** process, preparing installation and restoration documentation and coordinating reviews and approvals.
- Conducted **field walkdowns** to verify equipment against drawings; documented discrepancies and helped resolve them.

Projects

Python RTL Regression Tool | *Python, Icarus Verilog, Yosys, JSON* **Jul. 2026**

- Built a **Python RTL regression tool** that reads JSON configurations, compiles SystemVerilog modules, runs automated simulations with Icarus Verilog, and generates **consolidated pass/fail reports**.
- Created **20+ directed and parameterized tests** across 3 RTL modules and injected reset, arithmetic, timing, overflow, and underflow defects to confirm the regression flow correctly detected failures.

L1 Cache Controller | *SystemVerilog, RTL Design, Computer Architecture* **Oct. 2026 – Present**

- Designed and implemented a **1 KB direct-mapped L1 data cache** with 64 cache lines, 128-bit cache blocks, and 32-bit addressing, incorporating tag comparison, valid-bit tracking, and **hit/miss detection**.
- Developed a finite-state machine (FSM) for blocking **cache refills, managing CPU memory requests**, cache line replacement, and 128-bit transfers from main memory.
- Synthesized and deployed the cache controller on a **Xilinx Artix-7 FPGA (Basys 3)** using Vivado, integrating on-chip block RAM for cache storage and evaluating FPGA resource utilization.

Parameterized Synchronous FIFO | *SystemVerilog, Python, cocotb, Icarus Verilog, GTKWave* **Sep. 2026**

- Designed a parameterized SystemVerilog synchronous FIFO with **configurable data width and depth**, circular read/write pointers, **occupancy tracking**, and full, empty, almost-full, and almost-empty status flags.
- Defined deterministic **simultaneous read/write**, overflow, and underflow behavior, preserving FIFO ordering through **pointer wraparound** and back-to-back transactions.
- Verified data integrity and flag timing with a **cocotb scoreboard** backed by a Python reference queue, directed boundary cases, randomized traffic, **SystemVerilog assertions**, and waveform inspection.

Certifications & Courses

Simulation and Hardware Debugging for FPGA Designs | *Microchip* **Oct. 2026**

- Introduction to RTL simulation, HDL testbenches, QuestaSim, FPGA hardware debugging with SmartDebug and Synopsys Identify.