RAMTIN SOLEYMANI

(647) 986-8582 | soleymani.ramtin@gmail.com | LinkedIn | Github | Portfolio | Project Deck

EDUCATION

Toronto Metropolitan University Sep. 2021 – May 2026

Bachelor of Engineering, Electrical Engineering (Co-op) *Toronto, ON*

- **Relevant Coursework:** SoC Design, Computer Architecture, Digital Systems, Low-Power ICs, Embedded Systems
- **Dean's List; Final Year GPA:** 3.67/4.33

TECHNICAL SKILLS

Languages & Scripting: SystemVerilog, Verilog, Python, C/C++, SQL, Assembly, Bash/C Shell
Tools & Technologies: Synopsys VCS, ModelSim, Vivado, Intel Quartus II, Cadence Virtuoso, Git, Keil μVision5
Verification & RTL: UVM, SVA, Functional Coverage, Scoreboards, Testbenches, Directed/Constrained-Random Testing, Formal Verification, AMBA AXI/AHB/APB Fundamentals, UART/I2C/SPI, RTL Debug, Regression Testing

EXPERIENCE

Electrical Engineering Intern (Transmission Lines Engineering) Aug. 2024 – Aug. 2025

Hydro One Networks Inc. *Toronto, ON*

- Wrote a **20+ page technical report** using Hydro One/IEEE standards for PM safety/design decisions.
- Coordinated with Stations/PM teams in a **fast-paced** setting, sharing updates and tracking open projects.
- Automated PMP reporting/tracking workflows, improving organization and reducing manual update time by **50%**.
- Ran **induced-current/EMF checks** in TLW-GEN2 to confirm safety limits near HV transmission lines.
- Modelled HV lines in **PLS-CADD** and checked sag, tension, clearance, and conductor assumptions.
- Checked redesigned HV line models against **standards**, leading to **line re-stranding**.

Part-Time Electrical Engineering Intern (Electrical Design Engineer) May 2023 – Aug. 2024

R.J. Burnside & Associates Limited *Pickering, ON*

- Completed electrical design tasks **independently** while managing deadlines and peer reviews with the team.
- Checked electrical drawings against **client/city standards** before peer review and submission.
- Updated a street-lighting design, solving a cost issue and cutting construction cost by **25%**.

PROJECTS

Multi-Cycle MIPS Custom CPU | *SystemVerilog, STA, Assertions, Python* *GitHub*

- Designed a **32-bit multi-cycle CPU** in **SystemVerilog** with Harvard memory, a 19-state FSM, and datapath reuse.
- Validated **30+ ISR** using directed tests, waveform debug, FPGA bring-up, and a Python assembly-to-hex tool.
- Used synthesis and **STA** to debug timing, improving worst slack from **-0.742 ns** to **+3.275 ns**.
- Achieved stable **100 MHz timing closure** in Vivado after optimizing the critical datapath and control logic.
- Reduced hardware area by **30%** through datapath reuse while preserving correct instruction execution.

Camera Image Pipeline Verification | *SystemVerilog, UVM, CDC, UART* *GitHub*

- Built a real-time **OV7670 camera pipeline** in **SystemVerilog** with SCCB, RGB565, VGA, Sobel, and UART capture.
- Compared **FIFO + framebuffer** vs **single-framebuffer** architectures, testing tearing, latency, frames, and tradeoffs.
- Verified both paths with **UVM scoreboards**, **SVA**, coverage, and corner-case tests for stream behaviour.
- Checked **async FIFO CDC**, overflow/underflow, dropped-frame, and mismatch cases, achieving **0 errors**.
- Reduced tear severity by **32.91%** across a **25-frame** comparison test under BRAM limits.
- Scripted a **Python UART capture tool** to read framebuffer pixels and save PPM images for **debug review**.

FPGA Stopwatch on Basys 3 | *SystemVerilog, UVM, Waveform, Testbench* *GitHub*

- Developed a stopwatch in **SystemVerilog** on **FPGA** with separate control and timing blocks for a cleaner design.
- Engineered **debounce** and **edge-detection** logic for push-button control, preventing accidental state changes.
- Verified the design with a self-checking **UVM environment** and **scoreboard**, achieving a **100% pass rate**.

Embedded Media Center | *C++, OOP, Firmware* *GitHub*

- Created a real-time media centre on an **ARM Cortex-M3** with LCD graphics, audio playback, joystick input, and games, demonstrating a strong **embedded C and C++ development** across multiple interactive features.
- Modelled game logic with **classes and objects**, reducing memory usage by **60%**.
- Used **C++** and **OOP** to organize firmware more cleanly, making menus and input handling easier to manage.

Open to relocation. Canadian Citizen.