Diana Jung

437-605-9273 | dianajunguoftece@gmail.com | linkedin.com/in/dianajung-uoft | github.com/hyeonjijung1 | dianajung.com

Education

University of Toronto **Expected May 2027**

B.A.Sc. in Electrical and Computer Engineering, minor in Engineering Business *Toronto, ON*

Relevant Coursework: Digital Logic, Computer Organization, Semiconductor Devices, Signals & Systems, Computer Architecture

Experience

FPGA Design Intern – Embedded AV Systems | GitHub **May 2025 – Sep 2025**

Korea University *Seoul, Korea*

- Designed Verilog RTL system (DE1-SoC, Cyclone V) with FSM-based control and multi-I/O datapaths; VGA rendered real-time PS/2 keyboard + piano keys, while PWM audio output generated tones. Closed timing at **100 MHz** with **<10%** LUT and **<5%** BRAM via pipelining.
- Built SystemVerilog testbenches with assertions and automated Quartus/ModelSim (Tcl/Bash), ensuring reproducible verification and low-latency performance.

Radio Systems Lead - Hardware Team | GitHub **Jan 2025 – Present**

Blue Sky Solar Racing *Toronto, ON*

- **Led 4-member subteam** to redesign STM32WL-based LoRa radio module enabling multi-node wireless link between solar + chase car; achieved **>2 km** range and **80%** TX power cut.
- Developed **embedded C** firmware (LoRa stack, UART/CRC handling) and validated end-to-end communication in lab; automated regression-style tests in **Python/shell** for repeatability.
- Owned hardware bring-up (Altium PCB, soldering, oscilloscope/PSU debug) and cut sequencing failures by **30%** through iterative validation.

Projects

5-Stage RV32I RISC-V Pipeline CPU RTL Design | GitHub **Jun 2025 – Present**

- Designed modular **5-stage RV32I pipeline CPU** in Verilog (SoC-ready) with hazard detection, data forwarding, and branch prediction; fully synthesizable and documented.
- Developed **SystemVerilog verification** (assertions, coverage) and automated ModelSim regressions (Tcl/Python), achieving **98%** instruction / **96%** branch coverage and resolving 12 RTL bugs.
- Synthesized in Quartus with timing/power analysis; achieved **500 MHz** Fmax (critical paths <10% slack) and optimized RTL to ∼<0.2 W dynamic power.

Controller Hardware for Software-Defined Radio (SDR) | GitHub **Jan 2025 – Apr 2025**

- **Led 3-member team** to design a 2-layer SDR controller PCB in **Altium** (ATMEGA324PB) for RX/TX switching and TXEN logic, integrated with Quadrature Mixer and PA subsystems.
- Owned full bring-up: **SMD/through-hole soldering**, power sequencing, and oscilloscope-based validation; achieved **±1 kHz LO** stability and **90° I/Q** phase accuracy.
- Developed **embedded C** firmware (Si5351A via **I^2C**, **UART** CAT protocol) with **Python/shell automation**, automating **70%** of validation and cutting test time by **65%**, while resolving UART misresponse errors.

FPGA–Nios V Hearing Loss & Aid Simulator | GitHub **Mar 2025 – Apr 2025**

- Developed real-time hearing loss/aid simulator on DE1-SoC FPGA with **Embedded C** (Nios V softcore); integrated PS/2, VGA, LEDs, mic, and audio I/O. Achieved **8 ms** latency, **12% LUT**, **5% BRAM**, and timing closure at **100 MHz** (**<1 ns** slack).
- Implemented gain, noise control, echo, filtering, and distortion in C; designed Verilog FSM for low-latency mode switching and **hardware-verified** functionality via oscilloscope and live audio tests.

5V Linear Voltage Regulator — Lab Validation Hardware | GitHub **Feb 2025**

- Achieved ±20 mV accuracy, <100 mVpp ripple, and <20 mV RMS noise by designing a 5V linear regulator PCB in Altium/LTspice, then soldering and validating with oscilloscope + bench PSU.

Technical Skills

RTL/Verification: Verilog, SystemVerilog (UVM, assertions, coverage), ModelSim/VCS
ASIC/SoC Flow: Synthesis, timing closure, power/perf analysis
Scripting & Automation: Python, Tcl, Bash
FPGA/Hardware: Quartus, pipelining, bring-up, lab validation (oscilloscope, PSU, multimeter); C/C++, UART, I2C, SPI