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, Sustainable Energy *Toronto, ON*
Relevant Coursework: Circuits & Electronics; Signals & Systems; Semiconductor Devices

Experience

Radio Systems Lead - Electrical Team | GitHub **Jan 2025 – Present**
Blue Sky Solar Racing *Toronto, ON*

- **Led a 4-member** subteam to redesign the vehicle's wireless communication hardware using STM32WL-based LoRa transceivers, migrating from a tethered link to a multi-user, low-power wireless network; achieved >**2 km** range, **80%** TX power reduction, and real-time streaming of **50+** telemetry parameters.
- Debugged array and MPPT Relay PCB in Altium Designer using bench power supply and multimeter; reduced sequencing failures by **30%** through iterative testing.
- Designed multi-layer PCB with STM32 MCU and buck converters; performed PDN transient analysis in LTspice, achieving <**50 mV** droop under **5 A** load step with <**2 ms** recovery.

FPGA Design Intern – Embedded AV Systems | GitHub **May 2025 – Sep 2025**
Korea University *Seoul, Korea*

- Developed Verilog-based FPGA piano on DE1-SoC (Cyclone V) with PWM audio, VGA UI, and PS/2 input, achieving **100 MHz** timing closure with <**10%** LUT and <**5%** BRAM utilization via pipelining.
- Automated Quartus synthesis and ModelSim waveform simulation with Tcl/Bash; authored subsystem testbenches for reproducible, low-latency verification.

Projects

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

- Designed a **5-stage RV32I** pipeline CPU in Verilog with hazard detection, data forwarding, and a 2-bit static branch predictor to minimize pipeline stalls.
- Built a **SystemVerilog** verification environment with assertions (hazard/forwarding/mispredict/PC-align/x0-protect) and functional coverage (instruction mix, branch outcomes, forwarding, load-use); ran **Tcl**-driven **ModelSim** regressions with Python/shell automation, achieving **98%** instruction / **96%** branch coverage and resolving **12** RTL bugs.
- Synthesized in Intel Quartus Prime, achieving **500 MHz (2 ns)** with <**10%** slack violation; optimized RTL for <**0.2 W** dynamic power and documented timing and waveform reports.

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

- **Led a 3-member team** to engineer and validate a 2-layer PCB in Altium Designer with ATMEGA324PB MCU for RX/TX switching and TXEN logic, integrating with Quadrature Mixer and PA subsystems.
- Achieved **±1 kHz LO** stability and **90° I/Q** phase accuracy through SMD/through-hole assembly and lab validation using oscilloscope and function generator.
- Developed **embedded C** firmware (Si5351A over **I^{2}C**, **UART** CAT protocol) with **Python/shell** automation covering **70%** of validation steps, reducing test time by **65%** and resolving UART misresponse errors.

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

- **Led a 2-member team** to develop a real-time hearing loss/aid simulator on the DE1-SoC FPGA using **Embedded C** (Nios V softcore); integrated PS/2 keyboard, VGA UI, LED indicators, mic input, and stored audio. Achieved **8 ms** latency, **12% LUT**, **5% BRAM** usage, and timing closure at **100 MHz** with <**1 ns** slack.
- Implemented gain, noise injection/suppression, echo, filtering, and distortion in C; designed Verilog FSM for low-latency mode switching and verified performance on hardware via oscilloscope and live audio testing.

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

- Designed 2-layer Altium PCB with design-for-test points; simulated in LTspice to meet ±**20 mV** regulation, then soldered and bench-validated at **5.000 V ±20 mV**, <**100 mVpp** ripple, and <**20 mV** AC RMS noise under 33 Ω load.

Technical Skills

RTL & Verification: Verilog, SystemVerilog testbenches (ModelSim, EDAPlayground), UVM-style methodology
FPGA Design & Prototyping: Intel Quartus Prime (DE1-SoC), timing analysis & closure, pipelined architectures
Board Bring-up & Lab Validation: Oscilloscope, multimeter, bench PSU, SMD/through-hole soldering, SI/PI analysis
Embedded Systems & Scripting: C, C++, Python, TCL/Perl, Bash; UART, I^{2}C, SPI; automated regression scripting
Tools & Workflow: Git, Linux, VS Code