Guneev Dhillon

(236)332-5804 | guneevd@student.ubc.ca | www.linkedin.com/in/guneev-dhillon | guneevdhillon.com (Portfolio + GitHub)

Education

University of British Columbia Vancouver, BC
Bachelor of Applied Science in Computer Engineering CGPA: 3.80

- Dean's List (2024–25)

Student Design Team Experience

Signal Processing Engineer Oct 2024 - Present
MindTap, UBC Biomedical Engineering Student Team *Vancouver, BC*

- Served as the primary contributor on a 7-person signal processing subteam within a 16-person design team, owning the design and implementation of the EEG signal processing pipeline
- Designed a real-time EEG signal preprocessing pipeline in Python to clean and structure raw biosignals for downstream classification, supporting sustained streaming across 8 channels
- Implemented a ring buffer and batch-based processing to reduce data loss by approximately 50% during streaming
- Developed a user-facing simulation to collect realistic signals in a controlled setting, enabling consistent and labeled data capture across recording sessions
- Built the EEG data acquisition pipeline using BrainFlow to stream timestamped signals into structured CSV datasets, used to train ML models and designed for future integration into the signal translation pipeline
- Evaluated and iterated on ML classification models to enhance EEG command accuracy by 28% (F1 Score)
- Published in the CJUR × MURC booklet following a UBC presentation, demonstrating technical communication

Technical Work Experience

Lead Code Instructor Sept 2022 - Sept 2024
Code Ninjas *Surrey, BC*

- Mentored 150+ students in designing and building games and robots using Python, C#, and JavaScript
- Guided students through the full development cycle, from ideation to deployment, emphasizing modular design
- Adapted lesson plans and project scaffolding to support a wide range of skill levels, improving student engagement

Projects

RISC-V Labs, Software Construction (CPEN 211) | C, RISC-V, SystemVerilog, FPGA 2024–2025

- Completed a sequence of embedded systems labs involving design of RISC-V hardware and firmware on FPGA platforms, repeatedly reasoning about datapaths, memory-mapped I/O, interrupts, and timing constraints
- Designed a single-cycle RISC-V processor using SystemVerilog, validating datapath and control logic around instruction fetch, decode, execute, memory access, and write-back
- Implemented interrupt-driven input handling using hardware timers and edge-capture registers, eliminating polling
- Designed and built a clocked up–down counter on a breadboard using SR latches, reasoning explicitly about state encoding, reset behavior, and illegal latch states in physical hardware
- Debugged hardware–software timing and state issues by analyzing register state and signal transitions in Questa simulations and on FPGA hardware

Combination Lock and Keypad (FPGA) | SystemVerilog, DE10-Lite, Quartus 2025

- Designed a clocked finite state machine implementing lock, unlock, error, and reset behavior from sequential input provided by an externally polled keypad module, on a seperate FPGA
- Synchronized and debounced external input signals to ensure state transitions under FPGA timing constraints

ARM64 Reverse Engineering Lab | ARM Assembly, ROP Programming, Linux, Git 2025

- Analyzed ARM64 binaries at the machine level to understand and manipulate control flow by reasoning about calling conventions, stack frame layout, and return addresses in the absence of source code
- Utilized return-oriented programming and stack frame injections to reliably redirect execution flow

Technical Skills

Languages: C, C++, SystemVerilog, ARM Assembly, RISC-V Assembly, Python, Java, C#
Embedded & Hardware: FPGA (DE10-Lite), digital logic, interrupts, timers, memory-mapped I/O, VGA, UART
Tools: Quartus Prime, Git, Linux, VS Code, ModelSim, JTAG debugging
Software: Flask, React, REST APIs, Docker