Mercury Mcindoe

236-513-2840 | mercurymcindoe@gmail.com | linkedin.com/in/maplesyruphg06 | github.com/maplesyrup-0606
mercurymcindoe.notion.site

Education

University of British Columbia Vancouver, BC
Bachelor of Applied Science in Computer Engineering, ***CGPA : 4.33/4.33, 90.1%*** *Sept. 2021 – May 2025*

- Minor in Honours Mathematics
- Dean's Honour List/Roll: 2021, 2022, 2023
- Martin Sikes Memorial Scholarship in Electrical and Computer Engineering, 2023

Experience

Software Developer Jan. 2024 – Present
UBC Uncrewed Aircraft Systems *Vancouver, BC*

- Developed a wrapper API for **MissionPlannerScripts** using **Python** to automate drone control through scripts.
- Performed unit and integration tests with **Pytest** and **Poetry**, and conducted 10+ flight tests to ensure firmware-software connectivity.
- Containerized server environments through **Docker**, enhancing consistency across devices.

Mathematics Undergraduate Teaching Assistant Sept. 2023 – Present
University of British Columbia, Department of Mathematics *Vancouver, BC*

- Led weekly **Calculus I** and **Calculus II** discussions to enhance student comprehension.
- Conducted weekly 3-hour office hours, providing personalized assistance to address individual student needs.
- Consistently maintained a 90% average favorable rating, reflecting high student satisfaction and effective teaching support.

Projects

RC4 Cracker | *System Verilog, RTL Design, On-Chip Memory*

- Constructed an RC4 decryption circuit using **System Verilog**, leveraging on-chip memory and a modular design.
- Implemented a quadra-core system for brute-force key cracking by leveraging parallel processing, reducing a 32-bit password decryption time from **10 minutes** to approximately **3 minutes**.

OS/161 Operating System Development | *C Programming, Operating Systems*

- Implemented synchronization primitives (locks) in **C** to ensure thread safety.
- Extended kernel with system calls and a robust file system, enhancing **process control** and **file management**.
- Used **GDB** for debugging, boosting performance and stability.

RISC Machine | *System Verilog, FPGA*

- Designed a 16-bit CPU using a simplified instruction set architecture in **System Verilog**
- Constructed an **Finite State Machine** to handle instructions such as data storage, branching, and arithmetic operations.
- Developed a **5-stage pipeline** for instruction processing, including components like an instruction decoder and arithmetic logic unit.

LectureLink | *Node.js, Firebase, MongoDB*

- Developed the backend system using **Node.js**, **Firebase**, and **MongoDB** to connect students based on their course enrollments, enabling them to find and collaborate with classmates easily.
- Developed server-side logic for course matching with **email.js**, enabling faster user matching.
- Implemented secure user sign-in and session management with **Firebase** authentication services.

Technical Skills

Languages: Verilog/System Verilog | C/C++ | ARM Assembly | Java | Python
Others: FPGA | Quartus | ModelSim | Bash | GDB | Git | Unix/Linux | RESTful API | Flask | HTTP | Jira
Related CourseWork: Computing Systems | Micro Computer Systems | Operating Systems | Computer Communications | Object-Oriented Programming | Algorithms and Data Structures | Relational Databases | Software Construction