Max Lan

415-264-6023 | ml248@uw.edu | linkedin.com/in/fulinlan20/ | github.com/FLinLan

Education

University of Washington Seattle, WA

Bachelor of Science in Electrical and Computer Engineering *Sep. 2022 – Jun. 2026*

- Relevant Coursework: Data Structures and Algorithms, Systems Programming, Database Management, Digital Circuits and Systems, Hardware/Software Interface, Computer Architecture, Embedded Systems

Experience

Firmware Engineer Sep. 2022 – Mar. 2023

Husky Robotics, University of Washington *Seattle, WA*

- Accelerated hardware development process through utilizing Altium Designer for PCB layout and design in robotic systems.
- Enhanced team collaboration efficiency by reducing code conflicts through implementing version control practices using GitHub to manage project codebases.
- Improved firmware-hardware alignment by reducing integration issues through interpreting and analyzing schematic sheets to understand and document hardware configurations.

Projects

Five-Stage Pipelined ARM CPU | *SystemVerilog, ModelSim, Intel Quartus, Git*

- Designed and implemented a 64-bit ARM CPU with a five-stage pipeline architecture (Fetch, Decode, Execute, Memory, Writeback) using SystemVerilog.
- Improved CPU performance by achieving a 20% increase in clock frequency compared to a single-cycle design, through implementing a 64-bit ARM CPU with a five-stage pipeline architecture .
- Enhanced instruction execution efficiency by reducing pipeline stalls by 40% through integrating data forwarding and hazard detection units.
- Ensured design reliability by achieving 40% test coverage through developing comprehensive testbenches to verify correct instruction execution and pipeline behavior.
- Optimized CPU design by improving timing performance by 50% utilizing ModelSim for simulation and debugging, and Intel Quartus for synthesis and timing analysis.

Plant Monitoring and Security System | *FreeRTOS, C/C++, ESP32, Arduino, Hardware Sensors*

- Developed a multi-functional IoT system capable of monitoring 3 environmental parameters simultaneously using ESP32 microcontroller and FreeRTOS, integrating various sensors.
- Improved system responsiveness by reducing task switching time by 80% through implementing concurrent task management using FreeRTOS, including two queues for data storage and multiple timers for task scheduling.
- Enhanced user experience by enabling 6 different monitoring modes through designing a user interface with LCD and button controls, allowing seamless switching between functionalities.
- Utilized hardware timers and implemented debouncing mechanisms for precise sensor data acquisition and responsive user input handling.
- Increased data acquisition accuracy by reducing sensor reading errors by 70% through utilizing hardware timers and implementing debouncing mechanisms.

Conway's Game of Life | *SystemVerilog, FPGA, ModelSim, Intel Quartus*

- Implemented a complex cellular automaton on an 8x8 grid with 8 possible states per cell using Verilog HDL on an FPGA-based DE1 board.
- Improved user interaction by enabling 8 different input methods through developing mechanisms for pattern creation and game initiation, incorporating debounced button inputs and an LED-based interface.
- Optimized timing management by reducing clock skew by 60% through utilizing a single clock source with a divider.
- Maximized FPGA resource utilization by fitting the design within 40% of available resources while ensuring accurate game behavior and system reliability through thorough testing.

Technical Skills

Languages: Python, C/C++, Java, SystemVerilog, ARM/x86 Assembly, Bash/Shell Script
Developer Tools: Git, GitHub, Visual Studio Code, Visual Studio, Xcode, ModelSim, Intel Quartus, Altium Designer, ESP-IDF
Hardware & Systems: DE1-SoC, ESP32