JOVAN KOLEDIN

206.504.4184 | jovank@uw.edu | https://www.linkedin.com/in/jovan-koledin/

SKILLS

Software Languages & HDLs: C, C++, Python, MATLAB, Java, SystemVerilog, Verilog, VHDL
Operating Systems and Build Automation: Linux family, Windows, Make, CMake, Bash, Shell scripting, Batch
EDA & Simulation Tools: Xilinx Vivado, ModelSim, iVerilog, OrCAD Capture, KiCad, Zuken CR-8000
Lab Equipment: Logic Analyzer, Oscilloscope, Multimeter, Signal Generator, Soldering Iron, High Voltage Power Supplies

RELEVANT EXPERIENCE

Embedded Software Engineer | Lockheed Martin*,* Bothell, WA July 2023 - Present

- Wrote C user space Linux device driver to bring up FTDI USB-to-SPI RF signal generator boards through a Linux CLI. Operators used to need multiple third-party GUIs and dozens of minutes to setup boards, but CLI reduced setup to one command, and saved $20,000+ of licensing costs. Tested with Logic Analyzer, O-Scope, and Multimeter
- Completed design of Linux C++ DSP engine for radar threat detection simulator. Led team consisting of three engineers, educated everyone on proper version control (Gitlab/Git) and documentation practices
- Streamlined Xilinx Dev Kit operation for operators by developing a C optical sensor data streaming application for Linux that communicated over TCP/IP with a host GUI. Also developed the accompanying MATLAB GUI to communicate with the on board FPGA via TCP/IP DMA and for viewing data from optical sensors
- Designed VHDL SPI interface IP blocks for Xilinx FPGAs, enabling communication with multi-DAC peripheral devices; validated with VHDL testbenches in Vivado and oscilloscope measurements of DAC signals
- Integrated VHDL glue logic between Xilinx IP components in Vivado for DSP RF FPGA applications, including clock domain crossing FIFOs and AXI4-Lite buses. Verified designs with Vivado synthesis toolchain and testbenches
- Created FPGA ModelSim build pipeline by developing complete VHDL testbench suite and .bat automation scripts

Graduate Research Assistant | P^N Computer Engineering Lab (PNCEL), UW January 2025 - Present

- Collaborating with Prof. Ang Li, to develop domain-optimized hardware accelerators and compilers
- Developed LLVM C++ compiler that converts LLVM IR to Data Flow Graph (DFG) ISA for reconfigurable dataflow unit processor. Engineered Python simulator GUI to visualize DFG execution steps and verify their memory interactions
- Designed and verified SystemVerilog reconfigurable vector processing elements that executed variable-precision unsigned addition operations. Also developed and verified scalar processing elements that executed memory load/store operations, integer multiplication operations, and comparison operations. Used SystemVerilog assertions in testbenches for verification
- Designed SystemVerilog memory modules that contained a register file and configuration memory, along with custom ports for NoC communication, processing element communication, and config network programming

PROJECTS

Raspberry Pi 5 LED Driver May 2024 - July 2024

- Developed a Linux C driver for controlling the popular WS2812b LEDs with a Raspberry Pi 5 using the MOSI GPIO pin
- Raspberry Pi broke all backwards compatibility with existing OSS packages for the LEDs because of a completely new peripheral chip introduced with the Pi 5. This package fixes the issue by allowing a user to toggle the SPI MOSI pin of the Pi 5 at the frequency needed to specify colors for the cascaded ICs that drive the LEDs

GPU SystemVerilog Implementation on FPGA January - April 2025

- Designed and implemented a RISC-V single core GPU in SystemVerilog that supported the rv32i ISA and a limited vector extension. Went above and beyond on class project by adding a custom vector unit that provided a 3x latency reduction and a 15.5x clock cycle count reduction compared to baseline GPU core for a 32x32 matrix multiplication
- Developed a custom cross-compilation pipeline to translate matrix multiplication RISC-V assembly with vector instructions into hardware-executable hex files that ran on the GPU

Crossy Road ASIC Design November 2024 - January 2025

- Created a fully playable version of *Crossy Road* on an ASIC using Verilog, leveraging the SkyWater 130 nm toolchain and Yosys/OpenLane for GDS generation, with a scheduled tape-out in early 2026
- Designed the core game logic, VGA monitor communication, and input handling, all in Verilog. Optimized original design with 5,000+ logic elements down to ~1,150 logic elements without removing any game features

EDUCATION

University of Washington, Seattle September 2024 - June 2026
M.S. Electrical and Computer Engineering, focus on Computer Architecture
Coursework: Computer Architecture, GPU hardware design with FPGAs, Semiconductor Physics, Data Structures and Algorithms

Gonzaga University August 2019 - May 2023
B.S. Electrical Engineering, Computer Science minor
Coursework: Operating Systems, VLSI, Microcomputer Arch & Assembly, Parallel & Cloud Computing, Digital Signal Processing