Keevan Harold

437-344-3036 | keevan.harold@gmail.com | Toronto, ON
keevanharoldportfolio.pages.dev

Education

University of Waterloo Waterloo, ON

Bachelor of Applied Science in Electrical Engineering | Expected 2029

Experience

Hitachi Rail May 2026 - August 2026

Software Analyst (Co-op) | Toronto, ON

- Developed C++ functionality for an LDC Viewer to organize automated-test data from the Environment Simulator, Wayside Simulator, and WinSCLTT, with updates guided by CBTC system requirements.
- Implemented data filtering and event timelines to help engineers investigate discrepancies in simulator outputs; debugged and validated updates with senior developers through Git-based code-review workflows.

Hitachi Rail September 2025 - December 2025

Software Analyst (Co-op) | Toronto, ON

- Built and maintained Python automated tests for train-operator commands against CBTC requirements; incorporated hardware-team input into realistic end-to-end scenarios.
- Integrated REST API requests using Postman to verify command execution and system responses.

Swarmio Inc April 2024 - August 2024

Junior Data Scientist | Toronto, ON

- Supported gaming latency and edge-computing tests; prepared Python datasets for CreateML experiments using an existing CoreML model.

AE Electrical LTD. April 2023 - August 2023

Administrative Assistant | Toronto, ON

- Maintained expense records and data entry; designed directory advertisements and documented material-sourcing tradeoffs.

Projects

Streaming Sobel Edge-Detection Accelerator

SystemVerilog, Python, Icarus Verilog, Yosys

- Designed synthesizable streaming RTL with two line buffers and a moving 3x3 window, explicit signed arithmetic, saturation, and ready/valid backpressure; used width-proportional storage instead of a full-frame buffer.
- Verified 6,822 arithmetic vectors and 17,376 flow-control output transfers against Python; achieved zero mismatches across 152,720 real-image output comparisons with continuous input and randomized gaps/stalls.
- Synthesized the 256x256 configuration with Yosys and validated an 8x6 synthesized netlist; measured one output per clock during unstalled interior runs in RTL simulation.

Single-Cycle RISC-V Processor

SystemVerilog, Icarus Verilog, Make, Surfer

- Implemented an RV32I subset with modular ALU, control, registers, memories, and branch logic; supported aligned LW/SW, conditional branches, JAL/JALR, LUI, and AUIPC.
- Built seven self-checking testbenches; verified arithmetic, memory, control flow, reset, and a simulation-only EBREAK halt in a 14-cycle CPU smoke test, with VCD waveform inspection.

4-Layer STM32 Buck-Converter PCB

KiCad, PCB Design, Power Electronics

- Designed and routed a 50-component STM32 board with 3.3 V buck regulation, USB, SWD, I2C/UART, and an external crystal; used local decoupling and dedicated ground/power planes.
- Completed routing with zero saved DRC violations, unconnected items, or schematic parity issues; documented stackup and routing constraints for the board review.

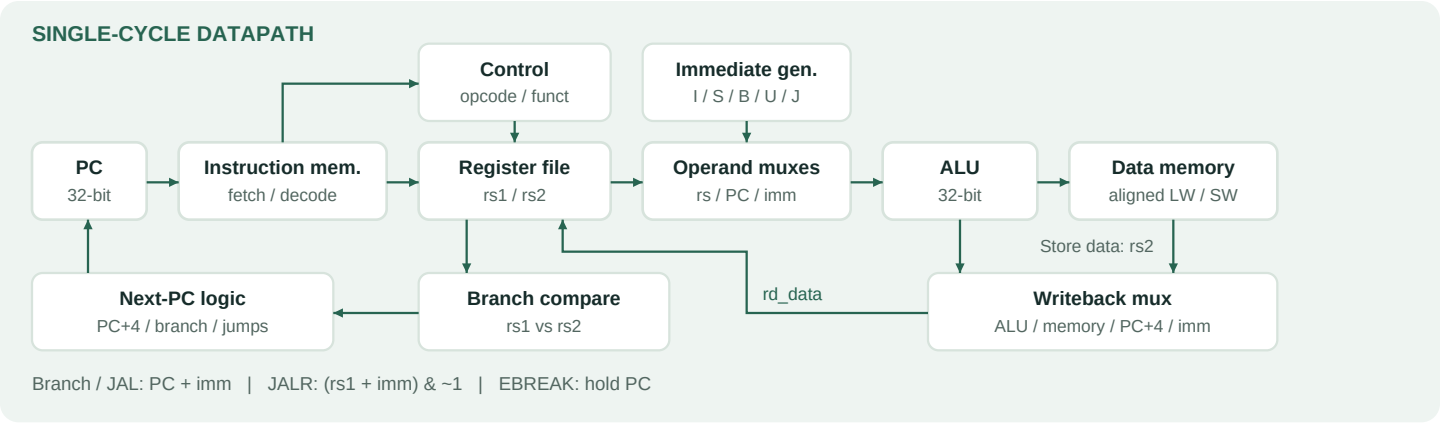
Technical Skills

Languages: SystemVerilog, VHDL, C++, Python **| Hardware:** RTL Design, PCB Design, KiCad, Digital Logic, Power Electronics, Quartus Prime
Tools: Icarus Verilog, Yosys, Make, Surfer, Git, GitHub, Bitbucket, Postman, VS Code
Other: NumPy, Pillow, VMware, CreateML, CoreML

01 Single-Cycle RISC-V Processor

SYSTEMVERILOG / ICARUS VERILOG / MAKE / SURFER

Designed a 32-bit educational processor implementing an RV32I subset, with modular RTL and a repeatable verification workflow spanning individual logic blocks and complete program execution.



Functional overview from cpu.sv; selected control and auxiliary connections are summarized.

IMPLEMENTATION

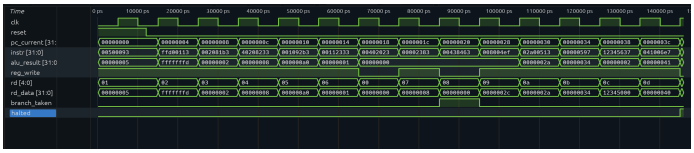
Processor Implementation

- **Modular datapath:** Connected the program counter, instruction memory, register file, immediate generator, ALU/control, branch unit, data memory, and writeback mux into a single-cycle core.
- **Arithmetic and decode:** Implemented ten register ALU operations plus immediate arithmetic, logic, shifts, and signed/unsigned comparisons; reconstructed I, S, B, U, and J immediates with sign extension.
- **Control flow:** Supported all six conditional branches, JAL, and JALR; selected sequential or redirected PC updates, returned PC + 4 link addresses, and cleared the JALR target's least-significant bit.
- **Memory and writeback:** Implemented aligned 32-bit LW/SW, LUI, and AUIPC; selected ALU, memory, link, or upper-immediate results for register writeback and restricted memory control to supported word accesses.
- **Repeatable execution:** Added parameterized instruction-image loading through IMEM_FILE and IMEM_LAST, NOP initialization for unused words, deterministic register/data-memory reset, and a simulation-only EBREAK halt.

VERIFICATION

Testing and Development Tools

- **Automated regression:** Built seven self-checking testbenches for the ALU, immediate generator, branch unit, register file, data memory, control unit, and integrated CPU; ran them with Icarus Verilog through make test.
- **Unit-level checks:** Tested ALU operations and zero detection, immediate reconstruction, signed/unsigned branch conditions, immutable x0, register reads/writes, memory enables, reset, and unsupported load-width rejection.
- **End-to-end program:** Verified a 14-cycle smoke test covering negative immediates, arithmetic, a store/load round trip, taken-branch instruction skipping, JAL/JALR links, LUI/AUIPC, reset, and the final halt state.
- **Debugging and artifacts:** Used $fatal checks for automatic failure reporting; retained readable assembly and matching hex input, a verification matrix, and make waves to regenerate VCD traces for signal inspection in Surfer.



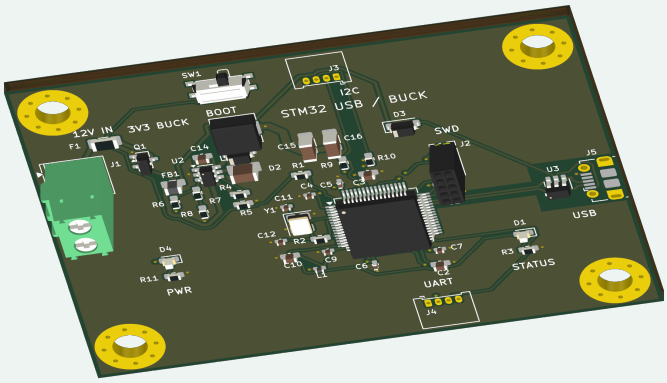
Surfer waveform from the CPU smoke test.

Scope: Simulated RV32I subset; full ISA compliance and FPGA implementation remain future work.

02 STM32 Buck-Converter PCB

KICAD / STM32F405 / USB / POWER ELECTRONICS

Developed a four-layer STM32 USB control-board design with 3.3 V buck regulation, peripheral access, and local support circuitry. Learning project based on Phil's Lab #11.



4 LAYERS

Ground + 3.3 V planes

50 COMPONENTS

489 tracks / 54 vias

0 DRC VIOLATIONS

Saved KiCad checks:
0 unconnected items
0 schematic parity issues

3D CAD view of the routed board; USB, I2C, and UART connector models are unavailable.

CIRCUIT DESIGN

Schematic and Supporting Circuitry

- **Controller and interfaces:** Integrated an STM32F405RGT6 with a Micro-USB connector, SWD programming/debug access, and separate I2C and UART headers for peripheral connectivity.
- **Buck power stage:** Incorporated the MP2359 regulator, switching inductor, Schottky diode, feedback divider, and input/output capacitors to provide the board's intended 3.3 V supply.
- **MCU power support:** Included local supply decoupling and bulk capacitance, plus a ferrite-bead-filtered analog supply; placed support components near the associated MCU power pins.
- **Clock and boot circuitry:** Added a 16 MHz external crystal with 12 pF load capacitors, boot-selection circuitry, and power/status LEDs with current-limiting resistors.
- **USB and input protection:** Incorporated USBLC6-2SC6 protection on the USB data interface and fuse, MOSFET, and diode components in the input-power circuitry; retained power-path review as a pre-fabrication task.

PCB ENGINEERING

Board Layout and Project Outputs

- **Four-layer organization:** Routed USB and critical local signals on F.Cu, used In1.Cu as the ground reference and In2.Cu for 3.3 V distribution, and placed selected crossing signals on B.Cu with outer-layer ground pours.
- **Placement and power routing:** Kept the regulator switching connections, crystal, and decoupling local; used 0.55-0.80 mm input/switch tracks and 1.0 mm inductor-to-output-capacitor tracks, with 0.20 mm general signals.
- **USB routing:** Kept D+ and D- entirely on front copper over the ground reference; used a 0.28 mm width and 0.22 mm paired-trunk gap. Saved route measurements show a 0.643 mm total length difference.
- **Connectivity and checks:** Completed the 50-component layout with 489 track segments and 54 vias. Saved KiCad checks report zero DRC violations, zero unconnected items, and zero schematic parity issues.
- **Reviewable outputs:** Retained native KiCad schematic, PCB, and project files, copper-layer exports, board previews, routing notes, and machine-readable verification reports documenting dimensions and remaining review items.

Current status: Routed review draft. Fabrication and bench testing are pending; final component, power-path, and manufacturer impedance checks remain.

03 Sobel Edge-Detection Accelerator

SYSTEMVERILOG / PYTHON / ICARUS VERILOG / YOSYS

Designed and verified a streaming image-processing accelerator that converts grayscale pixels into edge strength, combining bounded storage, signed arithmetic, and robust ready/valid flow control.



256 x 256 input



254 x 254 RTL output

64,516 PIXELS

Exact Python agreement in each 256 x 256 image run.

4,096 BITS

Two 256 x 8 row buffers.
No full-frame storage in RTL.

1 OUTPUT / CLOCK

During unstalled interior runs.

Actual RTL simulation output. Image: NASA / scikit-image; grayscale conversion and resize in Python.

IMPLEMENTATION

Datapath and Streaming Control

- **Streaming architecture:** Accepted 8-bit raster pixels through a parameterized interface; formed moving 3x3 neighborhoods with two row buffers and nine local pixel registers, using storage proportional to image width.
- **Arithmetic correctness:** Explicitly widened operands before shifts and sums; computed 11-bit signed gradients, combined absolute values in 12 bits, and saturated to 8 bits to prevent overflow and wraparound.
- **Backpressure control:** Used a shared pipeline enable to freeze data, validity, coordinates, and memories when output is blocked; allowed pending results to drain when input stops.
- **Frame and reset handling:** Counted accepted pixels to track frame boundaries. Valid-only cropping excludes stale rows without clearing memory arrays; reset discards pending work and restores frame alignment.
- **Measured cycle behavior:** Observed one-cycle output availability after a window's bottom-right pixel is accepted, and 254 consecutive outputs in the unstalled 256x256 demonstration.

ENGINEERING EVIDENCE

Verification and Synthesis

- **Independent golden model:** Built reproducible Python/SystemVerilog checks covering 6,822 arithmetic vectors and 502 exact windows, with directed patterns, signed-gradient cases, saturation, and seeded random inputs.
- **Protocol stress testing:** Matched 17,376 accepted output transfers across six sizes and two seeds; exercised random gaps/stalls, 30-cycle stalls, consecutive frames, reset while blocked, and final pipeline draining.
- **Real-image demonstration:** Matched all 152,720 output transfers across 256x256 and 128x96 inputs in continuous and stalled modes, with zero mismatches and zero maximum absolute error.
- **Synthesis and netlist checks:** Synthesized four parameter configurations with Yosys, including 256x256. The 8x6 synthesized netlist passed 2,568 output comparisons against Python.
- **Reproducible evidence:** Retained commands, fixed seeds, image outputs, source hashes, tool versions, architecture notes, and machine-readable verification and synthesis reports.

Scope: RTL simulation and generic synthesis completed. FPGA deployment, device utilization, maximum clock frequency, and power remain unmeasured.