GABRIEL D. WEST

518-986-1173 | gwest9@illinois.edu | github.com/gw12343 | gabrielwest.dev

Computer architecture, microarchitecture, hardware-software co-design, FPGA synthesis, and systems tooling.

EDUCATION

University of Illinois Urbana-Champaign Expected May 2029
B.S. Computer Engineering Overall GPA: 4.0
National Merit Finalist · AP Scholar with Distinction

TECHNICAL SKILLS

Languages: C++, C, Verilog, SystemVerilog, Python, Assembly, Java
Hardware / EDA: Vivado, Logisim, Nexys A7 FPGA, GTKWave
Tools: Git, CMake

SELECTED INDEPENDENT PROJECTS · portfolio: gabrielwest.dev

fpga-builder: Graphical node editor with synthesizable Verilog generator May 2026 – July 2026

C++ / ImGui / Verilog / SystemVerilog / Vivado | github.com/gw12343/fpga-builder

- Built a drag-and-drop node editor with a **graph-to-Verilog compiler** backend (const-fold, dead-code strip, combinational-loop reject), hierarchical modules, and live RTL preview.
- Validated the compiler on UART, a complete VGA controller, and the 32-bit CPU described below. **Generated RTL compiled unmodified in Vivado and ran on a Nexys A7 FPGA.**
- Verified **bit-exact agreement among generated RTL, C emulator, and Logisim** on Fibonacci and IEEE-754 soft-float using identical microcode and program ROMs.

32-bit CPU: custom ISA, two-pass assembler, cycle-accurate emulator, FPGA bring-up May 2024 – Jan 2025

Logisim / Java / C / Vivado / FPGA | github.com/gw12343/32-bit-cpu · assembler · emulator

- Designed a **from-scratch 32-bit ISA and microcoded CPU** in Logisim (8 GPRs plus PC/SP/BP/AR/IR, 32-bit ALU, 5 flags, 4K×29-bit control ROM, interrupts, memory-mapped I/O).
- Built a Java two-pass assembler with label resolution, operand type-checking, instruction validation, and matched program/microcode ROMs shared by Logisim, the emulator, and FPGA.
- Implemented a cycle-accurate C emulator (Nuklear + SDL2 GUI) as the golden model: same ROM files, register/flag/memory views, step and continuous execution.
- Ported with fpga-builder (2026): **closed timing at 50 MHz on a Nexys A7-100T (16 logic levels, WNS +0.09 ns)** and ran Snake over UART on the board.

cpp-engine: Modular 3D OpenGL game engine May 2025 – Jan 2026

C++ / OpenGL / Lua | github.com/gw12343/cpp-engine

- Built a C++17 3D game engine with ECS, a GUID asset pipeline, hot reload, prefab/scene serialization, Lua gameplay scripting, and a docked ImGui editor (hierarchy, inspector, scene view, play mode).
- Implemented deferred OpenGL rendering: G-buffer, 4096×4096 cascaded shadows with PCF, SSAO, mip-chain bloom, and HDR IBL at **~120 FPS on the exterior Bistro scene (1.8M triangles) at 4K on an RTX 3060**.
- Integrated Jolt physics (rigid bodies, mesh/heightfield, character controller), ozz skeletal animation with crossfade, and hot-reloadable Lua scripts (Start/Update, collisions).

LEADERSHIP

Design Lead, FIRST Robotics Competition (FRC Team 20) Sept 2025 – June 2026

- Led the mechanical design subteam for an 80+ member competitive robotics team, coordinating with electrical and programming on hardware–software integration.
- Iterated the robot intake, reducing weight and improving cycle time (and scoring) by 30%.