

KIRRITHAN SATHANANTHAN

647-768-1438 | k3sathan@uwaterloo.ca | [linkedin.com/kirrithan](https://www.linkedin.com/kirrithan) | kirri.xyz | github.com/skirrithan

Passionate about designing ultra **low-latency**, **high-performance** systems at the intersection of **hardware** and **software**

Education

University of Waterloo

Bachelor of Applied Science in Engineering

Waterloo, ON

Skills

Hardware: SystemVerilog/Verilog, RTL Design & Verification, FPGA Design, ASIC Synthesis, STA, CDC

Software: Python, C++, Rust, C, Bash

Experience

Apple - Silicon Engineering Group (SEG)

Jan 2026 – Apr 2026

SoC Integration Engineer

Cupertino, CA

- Built a synthesis debug framework that analyzed **10K–50K line** logs across **11 stages**, reducing manual triage from a multi-day workflow to **under 15 minutes**
- Automated failure classification across RTL, constraints, timing, and physical synthesis, surfacing failing blocks/stages and root-cause summaries for RTL engineers
- Analyzed **gate-level netlists** and **SDC timing constraints** to identify logic-depth bottlenecks, synthesis violations, and **CDC issues** across large-scale SoC designs before physical implementation.
- Implemented regression intelligence across **131 SoC blocks**, aggregating **7,786 errors** by milestone, partition, severity, and synthesis stage, and generating **1,979** waiver entries across **129 blocks**

Silicon Hardware Engineer Intern

May 2025 – Sep 2025

“Excellent work ethic. High amount of learning over the course of the internship” – Dr. Stevenson, CEO

PAX Markets (YC25)

Palo Alto, CA

- Reduced trading latency by **20%** (50 ns to 40 ns) by collaborating with CEO and CTO to **architect** and **validate** FPGA infrastructure, and delivering the first silicon **on-chip** co-located exchange
- Identified and resolved **15+ critical logic bugs**, improving system reliability by developing a cross-language unit test framework (**Python**, **Rust**, **Verilog** RTL) to validate concurrency and edge cases across models, cutting regression debug time by **25%**
- Implemented and debugged SystemVerilog **pipeline optimizations**, resolving signal misalignment across pipeline stages and ensuring timing closure on critical datapaths resulting in a **15% reduction** in design errors
- Built event-driven test generation and golden-model validation pipelines to study correctness under concurrency, backpressure, and timing skew

ASIC CAD Design Engineer

Sep 2024 - Dec 2024

Infinera Corporation (Acquired by Nokia, 2024)

Ottawa, ON

- Built Perl/TCL CAD automation and Jenkins health-check flows, reducing simulation and synthesis setup time by **15%**
- Worked across **ASIC RTL, synthesis, and verification flows** using Synopsys tools (Design Compiler, VCS, SpyGlass), gaining exposure to timing, power, and linting constraints

Projects

Shard - KV Cache Compression for LLM Inference | Python, PyTorch, CUDA, Hugging Face

May 2026

- Built a **Triton/PyTorch KV cache compression** runtime for **Llama-3.1-8B-Instruct** on **NVIDIA B200**, reducing decode-time memory traffic by addressing **KV-cache data-movement bottlenecks** in long-context inference
- Achieved **10.0x KV compression** at 8K and **11.2x** at 32K, reducing 8K stored KV from **~1.07 GB** to **~102 MB**
- Preserved quality with **1.00 NIAH recall** across 4K–32K, **LongBench-E avg Δ -0.05**, and **WikiText-2 PPL +0.26%**
- Wrote **Triton compressed-attention kernels**, validated kernel correctness against PyTorch references, and used **torch.profiler** CUDA traces to reduce decode loop time from **15.1s** to **3.05s**

Clkwise - Chipdesign Co-Pilot | Flask, Python, Verilog, Cerebras SDK, HTML/CSS

Sept 2025

- Built a **full-stack web platform** that parses FPGA/ASIC Vivado timing reports and automatically generates Verilog code fixes to resolve critical path violations
- Integrated **Cerebras API SDK** for ultra-fast structured LLM analysis of Vivado timing data, extracting root causes and producing optimized RTL patches in seconds
- Developed a clean **diff visualization UI** to highlight RTL edits (additions/removals), enabling engineers to validate fixes with a single click