GEETH SATHSARA WIJESEKARA

github.com/SathsaraGeeth — linkedin.com/in/geethsathsara — geethsathsara.com
Colombo, Sri Lanka — +94 71 326 4461 — geethwa03@email.com

EDUCATION

University of Moratuwa, Sri Lanka 2024 – Present

· B.Sc. (Hons) in Electronic and Telecommunication Engineering CGPA: 3.89 / 4.00

TECHNICAL SKILLS

Languages	SystemVerilog, Verilog, C, CUDA, OpenMP, Python, RISC-V Assembly, LLVM
Architecture	CPU Architecture, SoC Design, FPGA Design, Memory Systems, AXI, Hardware Accelerators
Verification	UVM, SystemVerilog Assertions (SVA), Formal Verification, Functional Coverage
Tools	AMD Vivado, Zynq, Verilator, Yosys, Cocotb, Git, Linux, perf, Nsight Compute, Nsight Systems, Jenkins, Altium Designer
Interests	Computer Architecture, RTL Design, Hardware Accelerators, HPC, CUDA, ML

PROJECTS

Custom Multicore SoC & Software Stack

· Designed a parameterizable 64-bit SystemVerilog SoC with barrel-scheduled multithreaded cores, a custom RISC-V-inspired ISA, CSRs, traps/interrupts, LR/SC atomics, and a dual-lane FPU.
· Built distributed scratchpad memory, DMA, a Network-on-Memory (NoM), and a 2D mesh NoC for scalable inter-core communication and message passing.
· Integrated an 8×8 output-stationary systolic-array accelerator with kernel driver and userspace API, sustaining 128 INT/FP operations per cycle while achieving 99.88% peak utilization and 1.97× lower memory movement than a conventional mapping on dense 256×256 GEMM.
· Extended LLVM's RISC-V backend and developed a bootable Unix-like OS in C/assembly supporting SMP, virtual memory, VFS, process management, and an SMPD programming model.
· Verified using **UVM** constrained-random verification, **formal verification** (SymbiYosys), DPI-C++ ISA scoreboarding, and full-system OS boot regression, automated by a Jenkins **CI** pipeline with JUnit reporting.

Lazy Graph Tensor Execution Runtime

· Built a demand-driven tensor runtime in C with lazy graph construction, a typed IR, an `.op` DSL, and LLVM ORC v2 JIT specialization from LLVM bitcode.
· Implemented subgraph discovery, dead-path elimination, pointwise fusion, specialization caching, and memory planning to minimize compilation, allocation, and tensor traffic.
· Demonstrated on a 240-frame RAW10 ISP pipeline with a 99.5% JIT cache hit rate under persistent execution and on end-to-end YOLOv5n inference.

Systolic Array GEMM Accelerator

· Designed a parameterizable output-stationary GEMM accelerator with pipelined MAC processing elements and double-buffered operand storage, supporting fully pipelined INT8/FP8 matrix multiplication.
· Integrated AXI4-HP (DMA/memory) and AXI4-GP (control) interfaces on a Xilinx Zynq-7020 FPGA, achieving timing closure at 100 MHz; verified against a NumPy golden model using Verilator.

UART with UVM and Formal Verification

· Designed a configurable SystemVerilog UART with independent RX/TX FIFOs, 16× oversampling, majority-vote sampling, configurable baud generation, and an MMIO register interface.
· Verified using **UVM** constrained-random verification and **formal verification** (SymbiYosys), proving reset behavior, FIFO correctness, baud generation, and TX/RX protocol safety.

LLM-Assisted C Optimization Framework

· Built an LLM-assisted optimization framework that profiles target hardware and optimizes performance-critical C kernels through an automated compile-benchmark-validate loop.

SELECTED COURSE PROJECTS

SDR Paging System: Duplex BladeRF link with QPSK, AES-128 encryption, and stop-and-wait ARQ.
Analog Computer: Op-amp-based arithmetic, integration, and differentiation circuits.
Analog PID Controller: Closed-loop motor speed and position control.
Campus Network: Hierarchical, VLAN-segmented backbone network design.
Wearable Safety Device: Fall and gas detection with GPS-tagged SMS alerts.