Divyam Arora

divyama3@illinois.edu | divyamarora22 | DivyamArora22 | (217) 305-0860

Education

University of Illinois at Urbana-Champaign Champaign, IL
Bachelor of Science in Computer Engineering (GPA: 3.71/4.0) *Aug 2021 – May 2025*

- Emma L. Marshall Scholarship, James Scholar, Dean's List
- Coursework: Computer Architecture, Operating Systems, VLSI Design, Digital Systems Lab, Analog Signal Processing, Data Structures and Algorithms, Database Systems, IOT and Cognitive Computing, Linear Algebra, High Frequency Trading.

Skills

- **Languages**: SystemVerilog, VHDL, Assembly x86, C, C++, TCL, Python, Go, Shell Scripting, SQL, CUDA
- **Tools**: Synopsys VCS, Design Compiler, Cadence Virtuoso, Innovus, Xilinx Vivado, Quartus, Verdi, GDB, Lint, Linux, UVM

Work Experience

CPU RTL Design Engineer Champaign, IL
Independent Project *Oct 2024 – Dec 2024*

- Designed a high-performance, out-of-order RISC-V CPU (RV32IM) in SystemVerilog, implementing 4-way set-associative cache, explicit register renaming, and a GShare branch predictor, achieving 75% branch prediction accuracy and optimizing latency, power, and area trade-offs.
- Integrated a timing-accurate SDRAM controller and developed a UVM-based verification environment with constrained random testing, assertion-based verification, and functional coverage using Synopsys VCS.
- Optimized microarchitecture for ASIC implementation, performing timing analysis, synthesis optimization, and RTL debugging, while applying formal verification methodologies to enhance design scalability and performance.

Research Assistant Champaign, IL
XLab at UIUC *May 2024 – Aug 2024*

- Adapted Linux kernel memory management for Elastic Cuckoo Page Table (ECPT), improving latency, bandwidth, and system-level cache efficiency while analyzing performance using DynamoRIO and SST Simulator to evaluate memory-level parallelism and address translation efficiency with C++ kernel-level testing and debugging.
- Evaluated non-tree page table designs on memory management, including transparent huge pages and page table isolation, ensuring efficient address translation and memory-level parallelism.

Projects

RISC-V CPU Physical Design and PnR Automation *May 2025*

- Designed and laid out a standard cell library using Cadence Virtuoso, exported Liberty and LEF files, and used them for full-chip PnR of a single-cycle RISC-V32I processor.
- Implemented and integrated a bit-sliced datapath and synthesized controller using Synopsys Design Compiler and Cadence Innovus, optimizing for area and power.
- Automated PnR flow using Tcl scripting, handled floorplanning, placement, clock tree synthesis, and timing analysis to achieve clean layout and timing closure.

RTL Design Automation Framework for SoCs *Sep 2024*

- Developed a Python-based RTL design automation tool to streamline linting, Clock Domain Crossing analysis, and syntax checks for SystemVerilog-based SoC designs, reducing manual review time by 40% and improving debugging efficiency.
- Integrated the framework with Synopsys VCS and Verilator, automating testbench validation, coverage collection, and static code analysis for ASIC and FPGA workflows.
- Enhanced RTL verification flow by implementing automated reporting scripts for CDC issues, assertion-based rule violations, and synthesis constraints validation, ensuring compliance with design best practices and reducing verification bottlenecks.

FPGA Based Mortal Kombat *Apr 2024*

- Engineered a custom IP core utilizing the AXI protocol for high-speed memory subsystem communication, implemented in SystemVerilog, ensuring efficient data transfer and interconnect scalability for compute-intensive applications.
- Integrated and synthesized a MicroBlaze soft processor on a Xilinx Spartan-7 FPGA to manage game logic execution and peripheral interfacing, optimizing system responsiveness while reducing compile time to under 90 seconds.
- Performed RTL verification and functional validation, conducting timing analysis, synthesis optimization, and debugging using Verilog simulation tools, ensuring stable performance across varying workload conditions.

Linux-Like Operating System *Dec 2023*

- Designed and implemented a Linux-like operating system core for the x86 architecture, incorporating segmentation (GDT, IDT), paging, and device initialization, aligning with Intel IA-32 specifications to manage memory protection and isolation.
- Developed low-level kernel functionalities, including a terminal driver, Ext2 file system, real-time clock synchronization, and system calls for file operations, enabling efficient inter-process communication and resource management.
- Built a round-robin task scheduler with process control block (PCB) management, optimizing context switching and CPU utilization, while integrating hardware interrupts and exception handling for robust system stability.