Randy Widialaksono

linkedin.com/in/randyhw • r@codify.dev

Summary

Advancing ASIC design quality and methodology by building novel, performant tools, flows, harnesses, and models. Electrical engineering background in computer architecture and VLSI design, with prior experience applying ML and driving end-to-end RTL-to-GDSII 3D-IC tape-outs to functional silicon. Recent work centers on developing a core C++ CAD library and tools for early floorplanning and shift-left initiatives, exploring applications of RL, Rust, and agentic AI.

Experience

NVIDIA — Austin, Texas

VLSI CAD Architect and Developer / Software R&D Engineer – ASIC Physical Design *2020–present*

- Developed, maintained, and deployed an in-house C++ library and tool suite for design and methodology teams, spanning floorplanning through DRC. Managed build pipelines, library dependencies, Python & Rust bindings generation, regression testing infrastructure, and cross-tool integration support.
- Eliminated production-flow bottlenecks in chip integration and physical design by building high-performance tools that cut manual work or slashed runtime, typically from hours to minutes. Built design flows to improve productivity and quality by conducting research, collaborating across teams, and implementing algorithms.
- Applied data-oriented and user-centric API design around standardized data exchange formats.
- Explored applications of agentic AI/LLM: developing a custom harness and applying a zero-token architecture.
- *Sample scope:* top-level reuse-aware bus route planning, feedthrough routing, floorplan graph extraction & traversal, ACO, spanning trees, GUI development, multithreading, chiplet RTL generation, ECO, netlist/layout querying & editing, readers/writers, computational geometry, pinning, constraint generation, and custom routing.

Intel — Austin, Texas

Silicon Architecture Engineer – SoC Performance Architect *2017–2020*

Integrated an x86 model into a C++ system-level simulator; applied machine learning for performance projections.

Design Automation Engineer – Physical Design *2016–2017*

Developed, configured, and deployed in-house EDA physical design flows for partition/block-level closure.

North Carolina State University (NCSU) — Raleigh, North Carolina

Graduate Teaching Assistant & Research Assistant *2010–2016*

Led design and implementation of a 3D-IC processor fabricated on an IBM/GF + Tezzaron TSV die-stacking process. Developed and executed the entire flow: synthesis, floorplanning, CTS, P&R, timing closure, and DRC/LVS fixes.

Qualcomm Research — San Diego, California

Research Intern: mixed-signal circuit verification and FPGA prototyping. *Summer 2013*

Skills

Programming	VLSI Implementation	Tools and Frameworks
C++, **Python**, Tcl.	SystemVerilog/Verilog, SystemC.	CMake, Pybind11, PyO3, CI/CD.
Basic: Java, Perl, SQLite, R,	ICC2, Innovus, PrimeTime, Calibre.	Git, Perforce, Qt, ImGui, PyTorch.
HTML, JS/TypeScript, Rust.	Place & Route, DFT, Timing, DRC/LVS.	Marimo, Pandas, Agentic Harnesses.

Select Publications/Awards

A Logic-on-logic 3D-stacked Heterogeneous Multi-core Processor. IEEE ICCD Conference 2017.
Physical Design of a 3D-stacked Heterogeneous Multi-core Processor. IEEE 3D-IC Conference 2016.
Reuse-Based Optimization for Pre-Bond and Post-Bond Testing of 3D-Stacked ICs. IEEE Transactions on CAD vol. 34.
Ranked 34th in the USA, IEEExtreme 24-hour Programming Competition, 2014. Team of 2.
Best FPGA Implementation at International LSI Design Contest, Japan, 2009. Xilinx Award. Team of 3.

Education

North Carolina State University — Raleigh, North Carolina

Ph.D. in Computer Engineering *3.98/4.0*

Dissertation: Three-Dimensional Integration of Heterogeneous Multi-Core Processors.
Built an automated 3D-IC flow optimizing paths across 3D vias; drove architectural feedback, RTL verification, and GDSII signoff. Taped out 3 chips on MPW runs. **TA**: Design of Digital Systems, Computer Design & Tech.

Graduate Coursework

Software Engineering	Advanced Microarchitecture	ASIC Design	Physical Design
Memory Systems	Parallel Computer Arch.	ASIC Verification	Electronic System Level Design
Computer Networks	Computer Design & Tech.	IC Technology & Fabrication	VLSI Systems Design
Embedded Systems Design	Digital Electronics	Modern Computer Algebra-AU	VLSI System Testing (Duke U.)

Duke University — Durham, North Carolina

Visiting Scholar: coursework, research collaboration on DFT and 3D-IC CAD. *1 year*

Bandung Institute of Technology — West Java, Indonesia

B.S. in Electrical Engineering *4 years*

Thesis: C implementation of neural networks and a Kohonen SOM (training/inference, floating/fixed-point) on a multi-core Parallax microcontroller. Teaching Assistant: Digital Systems, Microprocessor Lab.

Oita University — Oita, Japan

Research Exchange Student: Reinforcement Learning Lab (Prof. Shibata), with coursework *1 year*

Implemented C-based neural networks for face tracking on a panning camera and reinforcement learning algorithms for hardware control.

Project Experience (Academic/Side)

End-to-end Silicon Implementation / Tape-outs

Taped out a functional 3D-IC heterogeneous multi-core processor with thread migration at NCSU. Built visualizations and a fully automated back-end flow, and performed final physical verification (DRC/LVS) and layout fixes (e.g., antenna rule violations) for signoff.
Fabricated as two stacked 5.25 mm $\times$ 5.25 mm dies on a 130 nm process.

RTL Design (Verilog), FPGA Prototyping, Bare-Metal Programming (C/asm)

Implemented Sokoban (box-pushing puzzle game) on an FPGA: wrote game logic and sprites from scratch in C and MIPS assembly. Designed a pipelined MIPS processor in Verilog RTL with data forwarding and a Kogge-Stone adder, along with Verilog modules for button debouncing and VGA display rendering.

Machine Learning (PyTorch)

Implemented a quantized LeNet using feedback alignment and custom low-precision training algorithms.

Memory Systems

Performed modeling and performance comparison between ideal and non-ideal block placement policies for multi-core systems. Experimented with cache block placement policy: requestor core cache vs remote core cache. Analyzed experiment results from running SPEC2K benchmarks in SIMICS.

ESL & Physical Design (SystemC, HLS)

Performed TLM and ESL modeling of an SoC (ARM Cortex core, DRAM model, and AMBA bus), along with physical design optimization and signal integrity, power, and timing analysis. Tools: SystemC, Mentor Graphics Vista, Catapult, Python, C++, UML, Encounter, PrimeTime.

Parallel Computer Architecture

Implemented a cache coherence simulator in C++ with protocol enhancements to reduce off-chip memory accesses.

Computer Design and Technology (C++)

Implemented a cache, branch target buffer, and Tomasulo superscalar processor simulators.
Implemented a checkpoint recovery mechanism for a large fetch window processor within SimpleScalar simulator.

Advanced Microarchitecture (C++)

Implemented and compared thread migration (across cores) strategies within SimpleScalar simulator framework.

ASIC Verification (SystemVerilog)

Verified an out-of-order superscalar core for tape-out, uncovering design bugs in the load-store unit and issue queue.

Digital Electronics (CMOS circuit design)

Designed a low-power Hybrid Latch Flip-flop in academic 45 nm tech library. Operating clock frequency 4 GHz, power consumption 19.9 μW, setup time 13.5 ps, hold time 86 ps, t_{DQ} of 63.64 ps.
Designed a voltage-mode and current-mode differential transmitter circuit. Tools: HSPICE.

VLSI Systems Design (logic design, physical layout)

Designed a full-custom 3×3 arbiter-crossbar CMOS unit, achieving the 2nd best performance and energy-delay-squared metric out of 27 teams. Customized the power delivery network and clock tree design, created a custom standard cell library, and performed top-level integration. Achieved a 5.5 GHz clock frequency and 0.19 nW power with the FreePDK45 technology library. Tools: Cadence Virtuoso, HSPICE, Calibre DRC-LFD.

ASIC Design (Verilog)

Implemented a Viterbi decoder in RTL Verilog. Optimized throughput and the delay-per-unit-area metric by designing a fast floating-point unit, using dual-port memory, and pipelining.