

## Objective & Summary

Contribute to high-quality ASIC design by building novel in-house design automation tools and methodologies. Past experience includes C++ software development, performance architecture, and VLSI physical design automation. Built automated design methodologies by designing algorithms and implementing high-performance tools. Drove end-to-end RTL-to-GDSII ASIC implementation (academic project) yielding functional silicon.

## Experience

### NVIDIA

AUSTIN, TEXAS

**VLSI CAD Architect and Developer -> Software R&D Engineer, VLSI Physical Design** 2020-present

Applied research role developing, maintaining, and deploying an in-house C++ library and tools suite used by flow, methodology, and implementation teams across the company.

Past projects scope include top level floorplanning and design integration, netlist, layout, timing, ECO, reuse-aware routing, bus planning, readers, writers, pinning, GUI, bridging architecture and physical design.

Built new design methodologies by collaborating across teams, implementing algorithms, and supporting deployment into flows across multiple projects. Embedded agentic AI under a "zero-token architecture" discipline.

### Intel

AUSTIN, TEXAS

**Silicon Architecture Engineer** 2017-2020

Integrated multiple C++ based simulation infrastructure. Accelerated SoC performance analysis methodology.

**Design Automation Engineer** 2016-2017

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

### North Carolina State University (NCSU)

RALEIGH, NORTH CAROLINA

**Graduate Research Assistant** 2010-2016

Led design and implementation of a 3D-IC processor fabricated on IBM/GF + Tezzaron TSV die-stacking process.

Developed and executed RTL to GDSII flow; includes synthesis, floorplanning, timing closure, DRC/LVS fixes.

### Qualcomm Research

SAN DIEGO, CALIFORNIA

**Research Intern** Summer 2013

Performed mixed-signal circuit design verification, post-silicon measurements, and FPGA prototyping.

## Skills

| Programming                          | VLSI Implementation                | Dev Tools                        |
|--------------------------------------|------------------------------------|----------------------------------|
| C++, Python, Tcl.                    | SystemVerilog/Verilog, SystemC.    | CMake, Pybind11, CI/CD, ninja.   |
| Basic: Java, Perl, SQLite, R,        | ICC2, Innovus, PrimeTime, Calibre. | Pandas, PyTorch, build systems.  |
| HTML, JavaScript/TypeScript, Svelte. | Place-Route, DFT, Timing, DRC/LVS. | Git, Perforce, WebUI, Qt, ImGUI. |

## 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 34<sup>th</sup> in USA, IEEEExtreme 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. 2010 - 2016

Dissertation: Three-Dimensional Integration of Heterogeneous Multi-Core Processors.

Taped out 3 chips; developed and executed end-to-end 3D-IC backend flow including architectural feedback, RTL verification, and delivery of final GDSII layout. **Teaching Assistant:** 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 Sys. 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

Visiting Scholar: coursework, research collaboration

Bandung Institute of Technology

B.S. in Electrical Engineering

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

Oita University

Research Exchange Student: Reinforcement Learning Lab (Prof. Shibata), Engineering & Coursework

Implemented face following on a panning camera in C using neural networks; humanoid robot control in C++.

DURHAM, NORTH CAROLINA

2013

WEST JAVA, INDONESIA

2004 - 2009

JAPAN. '07-'08

## Project Experience (Academic/Side)

### Machine Learning (PyTorch)

Implemented quantized LeNet model with feedback alignment training.

Experiments on developing new training algorithms for quantized neural networks.

### End-to-end Silicon Implementation / Tape-outs

Successful academic tape-out (functional fabricated 3D-IC processor chip) of a heterogeneous multi-core processor system with thread migration features at NCSU. Developed visualizations and fully automated back-end flow. Performed final physical verification (DRC/LVS) and layout fixes (e.g. antenna rules violation) for signoff.

Silicon implementation has two stacked dies of 5.25 mm x 5.25 mm on a 130 nm process.

### RTL Design (Verilog), FPGA Prototyping, bare-metal programming (C/asm)

Implemented "Sokoban" (moving box puzzle game) on FPGA: coded the game in MIPS assembly by hand (prototyped in C). Wrote MIPS processor Verilog RTL, features include pipelining, data forwarding, Kogge-Stone adder (1 GHz clock in commercial 180 nm process). Wrote Verilog code for debouncing FPGA buttons and rendering graphics through VGA interface. Created game sprites.

### Memory Systems

Performed modelling and performance comparison between ideal and non-ideal block placement policy for multi-core systems. 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 & ESL modelling of an SoC design that consists of an ARM Cortex core, DRAM model, and AMBA bus. Performed physical design optimizations, signal integrity/power/timing analysis. Tools: SystemC, Mentor Graphics Vista, Catapult, Python, C++, UML, Encounter, PrimeTime.

### Parallel Computer Architecture

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

### Computer Design and Technology (C++)

Implemented cache, branch target buffer, and Tomasulo superscalar processor simulators.

Implemented a checkpoint recovery mechanism for 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, found design bugs in 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 4GHz, power consumption 19.9  $\mu$ W, setup time 13.5ps, hold time 86ps,  $t_{DQ}$  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 3x3 arbiter-crossbar CMOS unit, 2nd best performance and energy\*delay-squared metric out of 27 teams. Customized power delivery network and clock tree design. Created custom standard cell library and top-level integration. Achieved 5.5 GHz clock frequency, 0.19 nW power, with FreePDK45 technology library. Tools: Cadence Virtuoso, HSPICE, Calibre DRC-LFD.

### ASIC Design (Verilog)

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