Vighnesh Iyer

vighneshiyer.com
vighnesh.iyer@berkeley.edu

Research Interests

Microarchitecture simulation; sampled simulation; RTL verification and debugging methodologies/tools; stimulus generation techniques; specification mining; machine learning for DV; RTL fuzzing; DSLs for hardware construction / hardware-specific IRs; FPGA-accelerated emulation; FPGA-accelerated RTL-level power/performance estimation; RTL power modeling; domain-specific accelerators; hardware resiliency

Education

2018 – current	UC Berkeley: PhD EECS, Advisor: Prof. Borivoje Nikolic
2014 – 2018	UC Berkeley: BS EECS

Experience

May 2024 – current **Google** — Student Researcher

- Investigating trace-driven sampled simulation for multi-threaded workloads

August 2023 – current **UC Berkeley** — Graduate Student Researcher

- High throughput, low latency, high accuracy microarchitecure simulation.
  - Combine functional (ISA-level), uArch trace-driven models, and RTL simulation to exploit the best traits of each simulation methodology
  - Demonstrate that “multi-level simulation” enables fast microarchitecture iteration cycles with evaluation on realistic workloads
  - Github: TidalSim
- Applying software parametric fuzzing techniques to RTL verification.
  - Development of a parametric stimulus generator for RISC-V programs that also emits instrumentation to identify the influence of each byte of the parametric bytestream that serves as generator input
  - Leverage bytestream instrumentation to perform guided mutation
  - Applying hardware fuzzing to stimulus generation for microarchitectural metric targeting
- Methodologies for RTL coverpoint / bug synthesis using specification mining infrastructure.
  - Break the limitations of using open-source RTL for verification research by synthesizing microarchitecturally-interesting temporal properties
  - Demonstrate the usage of synthesized properties to evaluate different dynamic verification techniques
- Applying machine learning to RTL-level dynamic verification.
  - Investigating coverage extrapolation via GNNs to overcome the limitations of supervised learning for coverage prediction
  - Leveraging generator instrumentation for stimulus embedding
- Leveraging RTL-level formal-driven trace generation for power macromodel construction.
  - Mitigate the issue of low training dataset diversity by using formal tools to generate short and diverse traces that cover microarchitecturally-relevant (and power-relevant) design states and trajectories

May 2023 – August 2023 **Jane Street** — FPGA Engineering Intern

- FPGA infrastructure work

August 2018 – May 2023 **UC Berkeley** — Graduate Student Researcher

- Worked on a functional API for random stimulus generation that decouples the description of constraints from the interpreter that generates legal stimulus. In doing so, we enable automatic extraction of the randomization graph for stimulus embedding and coverage for the generator itself as well as the generated stimulus.
- Worked on a monadic simulation API for high-performance testbench fork/join threading.
- Worked on power modeling techniques that use selective signal sampling and event traces to estimate energy. Leveraged formal methods for trace generation for power model training.
- Worked on verification libraries for Chisel circuits with 2 MS students to create an API for constrained random stimulus generation, assertion based verification, transaction-level testing with VIPs, and cosimulation coupling a functional simulator with the RTL simulation of an accelerator written in Chisel
- Investigated the usage of specification mining for RTL bug localization by mining LTL properties from simulation waveforms and checking properties on failing simulations
- Worked on a systolic array based, dataflow configurable, GEMM accelerator generator (Gemmini) tightly coupled to a RISC-V core, designed for ML inference workloads
- Worked on the physical design and verification of a multicore RISC-V chip taped out in TSMC16
- TA'ing EECS 151/251A (Digital Design and ICs); led students through FPGA labs and the design of a pipelined RISC-V processor; teach discussion sections

May 2021 – August 2021 **Apple (CPU Verification Tooling)** — Intern

- Investigated the usage of machine learning to guide random stimulus generation for coverage targetting
- Designed a framework to evaluate various predictive models from data collected during stimulus generation to impact in RTL simulation

May 2020 – August 2020 **NVIDIA (ASIC and VLSI Research Group)** — Research Intern

- Developed models to predict RTL-level structural coverage from functional simulation features with the intention to accelerate coverage closure and guide stimulus generation

Jan 2018 – August 2018 **NVIDIA (ASIC and VLSI Research Group)** — Research Intern

- Emulated a ML inference accelerator testchip with a RISC-V controller on a VCU118 board; found bugs and workarounds pre and post silicon
- Developed a Microzed-based stimulus board for driving a testchip during radiation beamtesting
- Developed an FPGA-accelerated deterministic fault injection framework for simulating transient fault effects in single-clock RTL

Jan 2017 – Jan 2018 **Berkeley Wireless Research Center** — Undergraduate Researcher

- Developed FPGA RTL and simulation framework to interface between an ASIC and a host machine.
- Debugged and tested SERDES links connecting a RISC-V core's memory backend to an FPGA's DDR backing store.

June 2016 – August 2016 **Analog Devices** — Digital Verification Intern

- Developed a UVM testbench using SystemVerilog from scratch to stress test a NVM controller to be integrated into the digital portion of a mixed-signal chip
- Designed 2 verification IP blocks to be used in block-level and system/chip-level testbenches
- Wrote suite of coverpoints and ran simulations to find bugs and achieve full coverage on basic NVM operations

Jan 2016 – Dec 2017 **UC Berkeley** — TA: EECS 151 (Digital Design and Circuits)

- Designed six FPGA labs to introduce students to fundamental concepts of digital design encompassing FSMs, serial I/O, and chip-to-chip communication
- Expanded the class FPGA project consisting of a RISC-V core to include AC97 audio and DVI link video components

June 2015 – May 2016 **Guidewire Software —** Software Engineering Intern

- Developed a SPA using AngularJS on the front end and Spring + Jersey on the backend, that enabled users to perform administrative tasks; deployed to internal admins and Guidewire customers
- Wrote comprehensive unit and integration tests using Protractor and Karma/Jasmine; standardized CSS across all admin apps

June 2014 – April 2015 **Zurich North America —** iOS/Web Application Developer

- Created an iOS mobile app and REST API service to disseminate information regarding Zurich's IT Security Standards throughout the entire organization
- Deployed the mobile app to hundreds of IT architects throughout the enterprise
- Created a SPA and API for the internal distribution of security research

Awards

1. 2018 - Outstanding Graduate Student Instructor Award (UC Berkeley)

Publications

- Google Scholar: Vighnesh Iyer

1. RTL Bug Localization Through LTL Specification Mining. *Proceedings of the 17th ACM-IEEE International Conference on Formal Methods and Models for System Design.* No. 5, 2019. **Vighnesh Iyer**, Donggyu Kim, Borivoje Nikolic, Sanjit A. Seshia
2. A Dual-Core RISC-V Vector Processor With On-Chip Fine-Grain Power Management in 28-nm FD-SOI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* Volume: 28, Issue: 12, Dec. 2020. John Wright, Colin Schmidt, Ben Keller, Daniel Palmer Dabbelt, Jaehwa Kwak, **Vighnesh Iyer**, Nandish Metha, Pi-Feng Chiu, Stevo Bailey, Krste Asanovic, Borivoje Nikolic
3. Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration. *DAC 2021*. Hasan Genc, Seah Kim, Alon Amid, Ameer Haj-Ali, **Vighnesh Iyer**, Pranav Prakash, Jerry Zhao, Daniel Grubb, Harrison Liew, Howard Mao, Albert Ou, Colin Schmidt, Samuel Steffl, John Wright, Ion Stoica, Jonathan Ragan-Kelley, Krste Asanovic, Borivoje Nikolic, Yakun Sophia Shao
4. SimCommand: A High-Performance RTL Testbench API. *Open-Source Computer Architecture Research (OSCAR) Workshop at ISCA 2022*. **Vighnesh Iyer**, Kevin Laeufer, Koushik Sen, Borivoje Nikolic
5. Simulator Independent Coverage for RTL Hardware Languages. *ASPLOS 2023*. Kevin Laeufer, **Vighnesh Iyer**, David Biancolin, Jonathan Bachrach, Borivoje Nikolic, Koushik Sen
6. Mixed-Abstraction HDLs and A Discussion on Other Aspects of HDL Design. *Programming Languages for Architecture (PLARCH) Workshop at ISCA 2023*. **Vighnesh Iyer**, Borivoje Nikolic
7. New Embedded DSLs for Hardware Design and Verification. *Programming Languages for Architecture (PLARCH) Workshop at ISCA 2023*. **Vighnesh Iyer**, Kevin Laeufer, Young-Jin Park, Rohit Agarwal, Lixiang Yin, Bryan Ngo, Oliver Yu, Koushik Sen, Borivoje Nikolic

Skills

- FPGA emulation, Xilinx FPGAs, Vivado
- Verilog, SystemVerilog, Chisel/FIRRTL, UVM, verification metholodogy
- Java, Python, C, C++, Scala, OCaml, Rust, Typescript

Coursework

CS 61A/B/C (Data Structures, Computer Architecture)
CS 152 (Computer Architecture and Engineering)
CS 162 (Operating Systems and Systems Programming)
CS 188 (AI)
CS 294 (Recent Topics on Program Synthesis, Compilation, and Debugging)
CS 294 (Architectures and Systems for Warehouse-Scale Computers)
EE 16A/B (Designing Information Devices and Systems)
EE 105 (Microelectronic Devices and Circuits)
EE 120 (Signals and Systems)
EE 123 (DSP)
EE 128 (Feedback Control Systems)
EE 140 (Analog ICs)
EE 142 (RF Circuits)
EE 219C (Formal Methods)
EE 227 (Convex Optimization)
EE 240C (VLSI Analog-Digital Interface ICs)
EE 241B (Advanced Digital ICs)
EE 290C (DSP Circuits)
EE 290 (Hardware for Machine Learning)
EECS 149 (Embedded Systems)
EECS 151 (Digital Design, Digital ICs)
Math 53/54 (Multivariable Calculus, Linear Algebra, Differential Equations)
Math 128A (Numerical Analysis)
Math 221 (Advanced Matrix Computations)
Stat 133 (Concepts in Computing with Data)
Stat 134 (Concepts of Probability)
Stat 135 (Concepts of Statistics)
Stat 150 (Stochastic Processes)