

# Vincent Thai

763-283-6596 | [thaiv47@stanford.edu](mailto:thaiv47@stanford.edu) | <https://www.linkedin.com/in/vincent-thai-6aa919376/> | <https://github.com/Vigithai>

## EDUCATION

### Stanford University

GPA: 3.95/4.00

*B.S. Electrical Engineering, Music (Tau Beta Pi) & M.S. Electrical Engineering*

Graduating June '25 & June '27

VLSI, Baremetal Programming, Data Structures, Dynamical Systems, Circuits

## EXPERIENCE

### Embedded Engineering Intern

Mar 2025 – May 2025

*Kos AI, Inc*

- Analyzed and reverse-engineered TI AFE49I30 firmware in C to map biosignal sampling architecture and mode-specific register behavior for ECG/PPG acquisition, reducing afe power consumption and extending battery
- Produced onboarding documentation to assist firmware contractor with STM32CubeIDE project recovery and register-level control structure

### Research Assistant

Jan 2022 – Mar 2025

*Kwabena Boahen, Boris Murmann, Erin Mordecai*

Stanford Dept. of Electrical Engineering & Dept. of Biology

- Implemented 5 generalized mathematical methods of analyzing complexity of 130+ dynamical systems using Python and Julia based algorithm benchmarking platform Dynadojo (URL: <https://github.com/DynaDojo/dynadojo/>)
- Redesigned and transferred 40+ hierarchical circuitry schematics/netlists of RRAM testbench array chip library from Cadence Virtuoso into open source foundry using Skywater 130nm CMOS process
- Cleaned, analyzed, and visualized publicly available dengue case data across 30 years and 5,500+ municipalities in Brazil with R ggplot to probe for thermal dependence of dengue transmission

### Electrical Engineering Course Assistant

Sep 2023 – Present

*ENGR 40M, EE101A (Intro to EE, Circuits I)*

Mark Horowitz, Juan Rivas

- Teaching and grading of EE classes with 100+ students in lectures, labs, office hours, and 1 on 1 tutoring
- Debugging student projects with Arduino, waveforms, soldering PCBs, CMOS gates and filtered op amps

## PROJECTS

### Simulated GPU Rasterization Accelerator

Intro to VLSI w/ M. Horowitz, T. Tambe, Fall 2024

- Implemented various stages of the Reyes Image Rendering Architecture algorithm with SystemVerilog + EDA tools to create parameterized design generators for flexible optimization of performance-power-area
- Bounding box generation optimized with backface culling + Sample testing with edge equations in fixed-point arithmetic, with clock gating trials. Verified functionality through C++ gold model and RTL implementation
- Optimized power (27.698 to 23.653 mW), area (0.0418 to 0.0323 mm<sup>2</sup>, and rasterizer units (15 to 9) for overall figure of merit improvement of 300% (296 to 93.7) constrained to 2ns/triangle throughput and 1.2ns clock period
- Basic exposure to CAD designing 6T SRAM cell layout in Cadence Virtuoso with Synopsys Libraries

### Embedded Game Controller

Baremetal Programming w/ C. Gregg, Fall 2022

- Built a primitive working OS in C and ARM assembly using Vim and git in 10 weeks from scratch on Model A+ Pi
- OS library with interrupts, shell, graphics (fonts, glyphs, shapes, etc.), ps2 mouse + keyboard, printf and malloc
- Reverse engineered Nintendo Wii controller and used i2c Salae logic analyzer to interface into custom embedded game controller project. Coordinated private github repo with project partners to design custom UI/graphics, and develop game code without external libraries in C to run arcade games using the Pi OS

### Boombox

Circuits 1+2 w/ E. Pop, B. Murmann Winter 2022

- Designed and soldered filters, AC-DC converters, and oscillators for boombox that plugs to wall + 3.5mm jack
- Includes full-wave rectifier, potentiometer, source followers and voltage regulator
- Performed analysis of analog gain stages, frequency response, and feedback of various stages and components, using benchtop multimeter, oscilloscope, function generator, MATLAB, and LTSpice to verify functionality

### FPGA Songplayer

Digital Design w/ S. Mitra Fall 2023

- Implemented a programmable song player on FPGA with Verilog and Vivado, using combinational logic, FSM, timing and synchronization. Met timing despite heavy arithmetic using dff pipelines and approximating division with bitshift-add logic + precomputed divisor ROM of shift coefficients. ROM generated via Python script
- Song player combines user-specified amount of sinewaves to construct notes and chords with harmonics, and with attenuation controlled by an ADSR envelope for toggleable dynamics.