Kunal's **Portfolio**

University of Waterloo

B.A.Sc Honours Electrical & Computer Engineering

chandan.one
github/kunalchandan
kchandan@uwaterloo.ca
814-807-7652
linkedin/kunal-chandan

Awards

- Baylis Medical Capstone Design Award
- NSERC Undergraduate Student Research Award

Certifications

- 2023 - Ignition Core Certified
- 2022 - QNFCF Cleanroom Certification
- 2022 - G2N Cleanroom Certification

Interests

- Cycling
- Rock Climbing
- Juggling

Hardware Projects

BEAMFORMING HEARING AID 🔗

- Designed 4 channel microphone array PCB in **KiCAD**, PCB does active analog bandpass filtering, differential amplification, and multichannel **ADC** over **SPI** to Raspberry Pi
- R-Pi does compression and sends audio over **Flask** server for further digital filtering and beamforming
- **Pytorch** to create quantized voice isolation model and minimize latency and maintain performance
- Used **multiprocessing**, **asyncio**, and **websockets** to maximize throughput and performance

PIPELINED RISC-V CORE

- Designed 5-stage pipelined **RISC-V** 32-bit core in **Verilog** using only synthesizable constructs
- Core synthesized on FPGA and successfully ran programs. Testbenches used to ensure cycle accuracy

REALIZABLE ANALOG FILTERS 🔗

- Generated optimal schematics and realizable parts for analog filter given cutoff frequency and roll-off
- Used **sympy** for circuit analysis and **pandas** as a parts database backend

Software Projects

RAY TRACING ENGINE 🔗

- Implemented 3D recursive path-tracing for arbitrary materials on basic geometric shapes
- Used **nalgebra** for arbitrary rotations & positions of camera & objects
- Parallel processing of ray-tracing using **rayon** yielding **~10X** performance speed-up

MULTIPLE DNA SEQUENCE ALIGNER 🔗

- Perform multiple sequence alignement on DNA or amino acid sequences, **dynamic programming** and **graph theory** used to generate optimal sequence
- Used Go threads to improve performance, parallelization improved performance 8x

COMPILER FOR NOVEL LANGUAGE 🔗

- Written in **C++** to support basic arithmetic functions, arrays, maps, and functions as first order members
- Used **CMake** to manage project and dependencies, **Catch** for unit and end-to-end testing
- Used **Boost** to manage graph datastructures and vizualization of Abstract Syntax Tree (AST)
- Targetting **LLVM IR** to allow for cross-platform compatability

GPU MANDELBROT GENERATOR 🔗

- Fractal generator written in **C++** using **CUDA**
- Parallleized code ran **56,160%** faster compared to single threaded CPU program