Aadi Desai

aadi.desai@yahoo.com +447916412721 aadidesai London, UK 8bitsqu.id (Portfolio)

Profile

MEng graduate of Electronics and Information Engineering at Imperial College London, looking to start their career in Software and Hardware Engineering. Experience with embedded and systems programming and architecture design.

Education

Imperial College London *Oct 2019 - Jun 2023*

Master of Engineering - MEng, Electronic and Information Engineering

Award: Provisional 2.1 - Upper Second Class Honours - *Marks delayed due to marking and assessment boycott*

Course Modules: Advanced Computer Architecture • Analysis and Design of Circuits • Coding Theory • Communication Networks • Communications • Control Systems • Cryptography Engineering • Digital and Computer Architecture • Digital Systems Design • Discrete Maths • Embedded Systems • Hardware/Software Verification • High Level Programming • Information Processing • Instruction Architectures and Compilers • Mathematics • Music Technology • Network and Web Security • Principles of Distributed Ledgers • Program Analysis • Programming for Engineers • Signals and Systems • Software Reliability • Software Systems • Type Systems for Programming Languages

Queen Elizabeth's School *Sep 2018 - Aug 2019*

A-Level Subjects: Maths (A*), Further Maths (A), Physics (A*)

Skills

Used in Academics: C++ • Python • F# (Basic Familiarity) • Intel Quartus Prime • LiteX • SystemVerilog

Used in Personal Projects: C++ • Rust • Python • Docker + Compose • Cloudflare DNS + Pages + Workers

Soft Skills: Teamwork • Problem Solving • Analytical Skills • Leadership • Communication • Creativity

Experience & Accomplishments

Digital Design Engineer – Nordic Semiconductor *Apr 2022 - Sep 2022*

Led research into open-source tooling to improve the testing and development workflow of RTL within the company, focusing on Verilator for its speed in simulation and robust linting options.

Integrated low-overhead syntax highlighting for SystemVerilog within a VSCode extension to improve the designer experience and increase productivity, including custom rules for syntax and style linting on open files.

VEX EDR International Robotics Competition *Sep 2017 - May 2018*

Won awards in regional competitions and progressed to the UK national event, securing a finalist place, and qualifying our team for the International Event held in Kentucky, USA, where we represented the UK.

Goals: Design, build and test structural components of the VEX EDR robotics system, including compensating for known unreliable DC motors and encoders using PID controllers on the central computer. Programming the robot to compete in the assigned game and adjust for unknows such as opponent position and target location. Testing and debugging of the various modules that made up the robot.

Store Volunteer – St Luke's Hospice *Nov 2014 - Apr 2019*

Worked on the sales floor and in the storage room, initially as part of the Duke of Edinburgh Award. Continued volunteering for multiple years as the experience working with new team members and customers was rewarding. Progressed to lead volunteer and managed shop and other volunteers when needed.

Role requirements included: layout and ticketing of store displays and decorations, sales and processing purchases, aiding customers with any questions or specific items and supervising / training new volunteers on the various roles around the sales floor and sorting area.

Projects

Final Year Project: FPGA Accelerator for StackSynth *Full Year, 4th Year University, 2023*

StackSynth is an educational synthesiser platform for real-time programming, used in the 3rd Year Embedded Systems module of the Electrical Engineering course at Imperial College London. This project greatly increased the audio capabilities and performance of the platform using a custom OrangeCrab FPGA accelerator board. During the project, the following designs were developed in LiteX (Python-based Domain Specific Language) and SystemVerilog:

- LiteX wrapper: to contain and connect the individual SystemVerilog modules and provided IP.
- CAN receiver: to communicate (receive only) with the existing StackSynth platform correctly.
- 48kHz sample generator: 64 individual oscillators, each with waveform shape and frequency control
- C++ demo software for the integrated RISC-V CPU (VexRiscV): replaced by student-written software in the updated module coursework.

Home Lab / Server *Summer Break, 2nd Year University, 2021*

Installed a local server as a space for other projects to be deployed to and a sandbox to experiment with different tools or languages. Initially driven by my desire to automate more devices within the house, avoiding the need to buy expensive hardware that could result in vendor lock-in or incompatibilities.

- Installed a Home Assistant instance on a Raspberry Pi 4 Model B, initially using Dynamic DNS (DuckDNS), port forwarding and matching TLS certificates for remote access, but since upgrading to Cloudflare Tunnels, which avoid network security and DNS issues entirely.
- Recompiled the RPi4B 64-bit kernel with the AppArmor Linux Security Module enabled to meet Home Assistant server hardening recommendations and achieve “Officially Supported” status.
- Upgraded bedside, bedroom and office lights to smart lights using Shelly One hardware running ESPHome.
- Expanded to support extra applications, e.g., uptime monitors and self-hosted apps, also exposed using Cloudflare Tunnels via subdomains.

EE2Rover Group Project *Summer Term, 2nd Year University, 2021*

Goal: Design and build a rover with the ability to operate in a remote location without direct supervision, including reacting to and navigating through obstacles identified using an onboard camera. Design a remote-control interface with asynchronous instruction delivery so the rover could continue in the case of temporary loss of communication.

- Programmed the central ESP32 which hosted the control interface and provided communication channels to the other modules on the rover: Power, Motion, Vision.
- Identified communication standards for transfer of status information and commands between Rover modules, using 2-wire UART and the ArduinoJSON library for consistent implementation in all modules.

MIPS-Compatible CPU Group Project *Autumn Term, 2nd Year University, 2020*

- Designed a synthesizable MIPS-compatible CPU using a memory-mapped interface for peripherals.
- Developed a custom Avalon-compatible communication module to convert the original bus design.
- Migrated the interface design, which used separate bus interfaces for instruction and data memories, to the more complex Intel Avalon standard. This enabled evaluation of the CPU design alongside other standard Avalon-compatible modules.

Custom CPU Group Project *Summer Term, 1st Year University, 2020*

- Created a custom instruction set and matching CPU, with the goal of running three benchmark algorithms:
  - Calculate Fibonacci numbers using recursion: requiring a stack.
  - Generate pseudo-random integers: requiring an efficient method of multiplication.
  - Traverse a linked list to find a particular value: requiring efficient indirect addressing.
- Researched existing implementations of standard IP blocks to aid decisions on custom IP design choices.
- Planned the bit layout and opcodes of instructions within the 16-bit word size.