BARDIA ZADEH

M: 07780009536 **E**: bardia.mzad@gmail.com **L**: London

Online Portfolio : https://bardia01.github.io/

EDUCATION

Imperial College London, January 2026

UKRI Funded PhD in Neural Network acceleration

In progress, estimated finish date is summer 2029.

Imperial College London, September 2020 – June 2024

MEng Electronic and Information Engineering

Graduated with First Class Honours (77% overall grade and 80% in final year).

CAREER ASPIRATIONS / PERSONAL PROFILE

After roughly 2 years in industry as a hardware engineer, I have returned to Imperial College London for a PhD in neural network acceleration.

PROGRAMMING LANGUAGES AND TECHNICAL SKILLS

Programming languages / methodologies:

- Verilog / SystemVerilog
- UVM (Universal Verification Methodology)
- Python, PyTorch
- C++
- Pandas, Matplotlib
- SVA (SystemVerilog Assertions)

Skills:

- Computer Architecture
- Digital Design and Design Verification with UVM
- Timing and area analysis
- Object-Oriented Programming

WORK EXPERIENCE

Apple GPU, August 2024 – January 2026

Design Verification Engineer

Using SystemVerilog and UVM, I work to verify part of a complex GPU frontend design.

- Owning testbenches for multiple units
- Verifying the functionality of micro-architectural features
- Developing testbenches and stimulus which are maintainable and scalable

Apple GPU, April 2023 – September 2023

Design Verification Intern – Interconnect team

- Built models based upon architectural specification which aided in bring-up of new features
- Improved testbench performance
- Implemented coverage collection
- Created a packet tracking and packet timeout tool in UVM to aid debug
- Debugged failure signatures using logs and waveform viewers
- Built versatile stimulus using UVM

Graphcore, June 2022 – September 2022

Logical Design Intern – Silicon team

- Design, verification, and synthesis of multiple modules in SystemVerilog with the use of C++ and Python for infrastructure.
- Used SystemVerilog and EDA tools to design, synthesize and verify different Floating Point arithmetic unit hardware modules. These included implementing multiple advanced optimizations such as clock gating and forced parallelism.
- Optimised designs using physical synthesis tools
- Created custom testbenches to verify the designs using SystemVerilog

PERSONAL / ACADEMIC PROJECTS

- Designed an arithmetic accelerator deployed on an FPGA to accelerate computations written in C. The designs were iterated and optimised for timing and area. The final design incorporated DMA accesses and a custom hardware block including a CORDIC implementation.
- Designed an image processor on FPGA hardware. This included hardware implementations of filtering, color space conversions and gradient detection.
- Built a self-driving rover which communicated with servers to upload a map of its surroundings which it built through scanning objects. The image processor provided instructions for the rover's movement and data about the objects found by analyzing the video feed from a camera.
- Built, optimized, and verified a double precision floating point multiplier with multiple rounding modes which met IEEE754 standards and was heavily optimized for speed.
- Created scripts for use in trial synthesis of digital modules. These included writing area and timing budgets.
- Used industry standard EDA tools to analyze physical builds of digital modules. This involved timing and area analysis.
- Built a live multiplayer game hosted on AWS using the Unity game engine, which used a DE10-Lite FPGA as a controller with tilt, button, and switch control inputs. I used python to create an API to integrate the controller inputs with Unity.
- Built a C compiler to the ANSI-C standard using C++ which output MIPS-32 5.4 assembly language.
- Designed and simulated a Turing-complete dual core CPU with floating point arithmetic and a serial input port based on a reduced ARM ISA