

# PHILIP WOJCICKI

filip.wojcicki18@imperial.ac.uk  
linkedin.com/in/filip-wojcicki

+48 600 800 121  
filipwojcicki.com

## EDUCATION

IMPERIAL COLLEGE LONDON '22 - '27  
PhD in Reconfigurable Accelerators, advised by:

- ◆ Prof. Wayne Luk — Custom Computing
  - ◆ Prof. Paul Kelly — Software Perf. Optimisation
- Research on **FPGAs**, **AI acceleration**, **compilers**

IMPERIAL COLLEGE LONDON '18 - '22  
MEng Computer Engineering (= *summa cum laude*):

- ◆ Distinguished Final Year Project (scored 88%)
- ◆ Y4: Ranked #1 of 65, 1<sup>st</sup> Class Hons (84%)
- ◆ Y3: Ranked #4 of 65, 1<sup>st</sup> Class Hons (80%)
- ◆ Y2: Ranked #1 of 74, 1<sup>st</sup> Class Hons (83%)
- ◆ Y1: Ranked #1 of 74, 1<sup>st</sup> Class Hons (80%)

## RESEARCH EXPERIENCE

I have presented my work at:

- ◆ FCCM'26, Flatiron Institute, REACH'25, FDF'25, ScalPerf'25, ACACES'25, CICSIC'23, NANDA'23, Tianjin University, CERN FastML workshops, FPT'22,

I have served as a reviewer for:

- ◆ FPGA, FCCM, CF, IMPACT, PACT, Euro-Par

## TEACHING EXPERIENCE

Since 2020, I have been a teaching assistant:

- ◆ I develop and maintain labs and coursework about creating a C90 to RISC-V compiler in C++.
- ◆ That includes automation for testing, benchmarking, reviewing and environment deployment, with experience report planned for SIGCSE TS'27.
- ◆ I provide support during lab sessions, mark coursework, offer 1-to-1 tutorials. I have been awarded **Top Undergraduate Teaching Assistant** prize.

## SKILLS

### PROGRAMMING

- ◆ Python, C++, SystemVerilog, RISC-V ASM

### TOOLS & TECHNOLOGIES

- ◆ ML frameworks (PyTorch, HLS4ML)
- ◆ Quant. Aware Training (Brevitas, QPyTorch)
- ◆ Hardware design/verification (Vivado, Quartus)
- ◆ Compiler infrastructure (MLIR, LLVM)
- ◆ Linux environment, version control (Git, Bash)
- ◆ Software profiling (Intel VTune Profiler, perf)

## INDUSTRIAL EXPERIENCE

FRACTILE Nov '24 – Jan '25  
Low-latency Researcher

- ◆ Evaluated latency and area trade-offs of block floating-point formats (**Microscaling**) for an analog in-memory-compute AI-accelerator ASIC.
- ◆ Implemented and benchmarked low-latency paths as **ASM RISC-V (RVV)** kernels and **SystemVerilog** blocks to guide HW/SW partitioning.

JUMP TRADING Sep '20 – Oct '20, Mar '21 – Sep '21  
Software & Hardware Engineer

- ◆ Created a library in **Python** for exploring topology, configuring, connecting and graphing multi-device (ASIC, FPGA, CPU, GPU) hardware systems
- ◆ Built an efficient, distributed arbitrage trading **C++** application composed of several parallel processes running on ASICs, FPGAs, and x86 machines
- ◆ Configured and benchmarked formal verification using **SystemVerilog** and a novel custom **Python** tool for use in ASIC and FPGA development
- ◆ Built an ultra low latency ASIC validation test platform for floating point calculations on x86 and RISC-V architectures using **C++**, **C** and **Python**

ARM Jun '20 – Sep '20  
Systems Architect

- ◆ Improved autonomous driving platform's verification in **SystemVerilog**, overhauled its documentation, added support for formal (**JasperGold**)

## RESEARCH PROJECTS

MX ATTENTION ON FPGA [FCCM, code]

- ◆ Built a parametrised **SystemVerilog** FPGA attention architecture with operator-wise MX-style BFP formats and configurable accumulation.
- ◆ Developed a **Python** DSE framework, finding designs with **29% fewer LUTs**, **7% fewer FFs**, and **0.42 lower perplexity** than baseline.

QUANTISED TNN ON FPGA [FPT, code]

- ◆ Developed a novel TNN architecture for GPUs (PyTorch) and FPGAs (HLS) for High Energy Physics experiments in collaboration with CERN
- ◆ FPGA solution outperformed SoTA models on GPU by **~1000 times** thanks to software/hardware-aware optimisations, without accuracy loss
- ◆ Experimented with quantisation-aware training (QAT), developed a quick FPGA-friendly post-training quantisation (PTQ) scheme for HLS4ML

QUBIT VISION TNN ON FPGA [IEEE TQE - pending]

- ◆ Designed a low-latency FPGA pipeline (CameraLink) for real-time qubit-state classification in collaboration with experimental quantum physicists
- ◆ Optimised the ViT architecture to achieve millisecond-scale end-to-end detection with up to **~120× lower latency** than a GPU baseline

ZERO-SHOT LEARNING [ACM TECS - pending]

- ◆ Developed a lightweight zero-shot learning framework with attribute knowledge graphs, reducing parameters by **~100×** while retaining accuracy
- ◆ Designed an accelerator on FPGA for CNN feature extraction and attribute recognition, achieving **~67×** speedup over a software-only baseline

SYSTOLIC ARRAY COMPILER [ICCD - pending, code]

- ◆ Developed an MLIR-based HLS framework for **C/C++** and **PyTorch** programs to systolic-array accelerators using polyhedral optimisations.
- ◆ Achieved SoTA performance while enabling **MLIR** pass interchangeability.