The First FPGA Backend for the HEIR Compiler: Evaluating Boolean vs. Arithmetic Pipelines for Practical FHE Acceleration

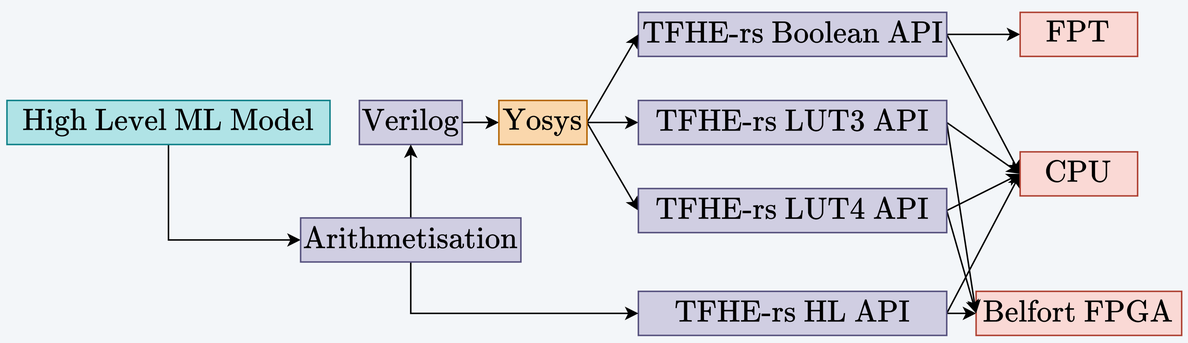
Wouter Legiest[1], **Thomas de Ruijter**[1], **Asra Ali**[2], **Jeremy Kun**[2] **and Ingrid Verbauwhede**[1]

[1]**COSIC, KU Leuven** – [2]**Google LLC**

HEIR: Homomorphic Encryption Intermediate Representation

- Universal MLIR-based FHE compiler [1]
- Goals: conversion of high level non-FHE code to a secure and efficient FHE program
- Support for all FHE schemes and HW platforms
- Provide a platform for benchmarking

CGGI Pipelines

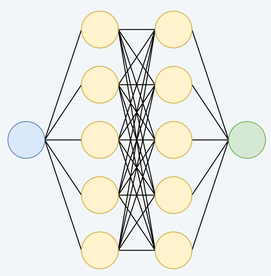


1. **High-Level Program**: Python-like definition
2. **Arithemetisation**: Convert to Verilog integration for Yosys
3. **Logic Synthesis**: Boolean or LUT generation using ABC synthesis
4. **Flexible Mapping**: Native support for 3-bit and 4-bit LUT architectures
5. **IR Integration**: Automated back-conversion to the HEIR dialects
6. **Backend Emission**: Optimised Rust code generation for targeted backends
7. **Hardware Support**: Ready for FPT [2] and Belfort FPGA architectures

Use Case

Simple Neural network inference with a an 8b integer input and no activation function:

	Nodes	Weights [bit]	Acc. [bit]
First FC	5	8	16
Second FC	5	8	32
Third FC	5	8	32



Batching: Vectorisation Passes

- *'StraightLineVectorizerPasses'* and *'CggiBooleanVectorizerPasses'*
- Core of finding parallelisation
- Based on Topological (Level) sorting of the execution graph
- Find the LUT or Boolean gates that can be done in parallel

CPU - FPGA Results

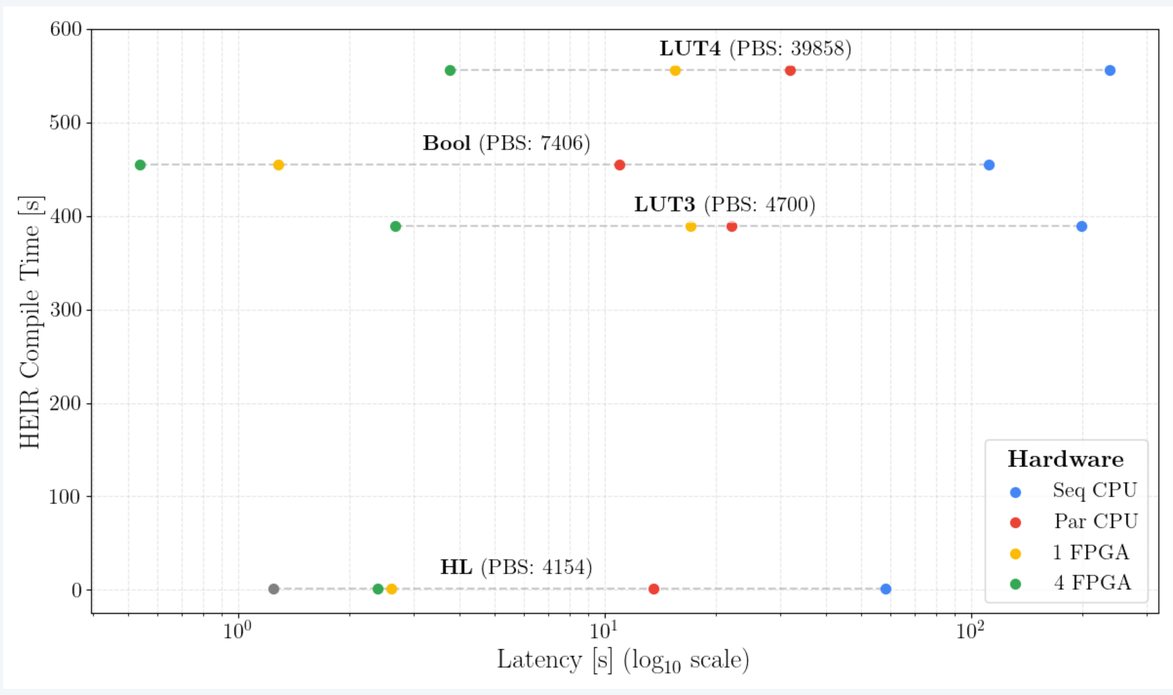


Figure 1: Latency of a neural network inference using different software generations and hardware platforms. The grey HL dots represent the parallel execution of a single operation across multiple FPGAs

Scheme: (Belfort) TFHE-rs v1.4.1 [3]
System: 2× Intel Xeon Silver 4208 – 8 cores – 2.1 GHz
4× AMD Alveo™ U55C FPGA Card

Conclusion:

- Possible to run different FHE operations on different FPGAs.
- Boolean has the highest utilisation of the FPGA Hardware, but has a higher compiler cost

References & Funding

[1] A. Ali et al., "HEIR: A universal compiler for homomorphic encryption," *CoRR*, vol. abs/2508.11095, 2025.

[2] M. V. Beirendonck, J. D'Anvers, F. Turan, and I. Verbauwhede, "FPT: A fixed-point accelerator for torus fully homomorphic encryption," in *CCS*, ACM, 2023, pp. 741–755.

[3] Zama, *TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data*, `https://github.com/zama-ai/tfhe-rs`, 2022.

ERC Adv. Grant No. 101020005; CSF No. VR20192203; FWO PhD SB No. 1S57125N

`wouter.legiest@esat.kuleuven.be`