**Laboratoire d'Architecture des Processeurs**
IN-F Ecublens
CH-1015 Lausanne
http://lap.epfl.ch

**Semester Project**
**Summer 2023**

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Benchmarking Dynamatic against modern HLS tools

**Dirren / Elija-Angelo** **Computer Science**

## Introduction

High level synthesis tools present a more agile approach for designing digital systems than classical workflows. By foregoing the necessity of writing RTL code in favour of defining the behavior of our desired circuit with high-level languages, we leave the task of designing the architecture in the hands of these tools and as such, want them to be able to produce circuits which run as efficiently as possible. Dynamic scheduling, used by the LAP's in-house HLS tool *Dynamatic*, allows for major performance gains in this respect; comparing its performance against industry-standard tools should give us insight on how this academic tool stacks up against its commercial counterparts.

## Goals

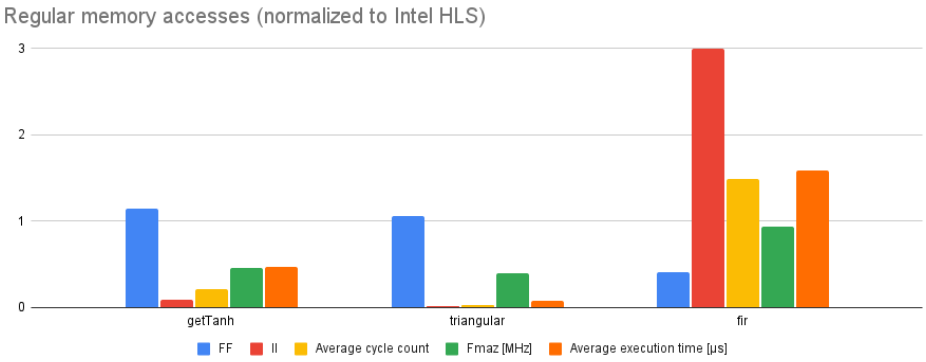
The goal of this project is to define a common ground on which these tools with differing architectures can be compared, in the form of a set of benchmarks. Once the data from each tool has been extracted on the basis of these benchmarks, the aim is to analyse the approaches the different tools take and what performance they achieve, to be able to make qualitative judgements on the advantages and drawbacks of each.

## Summary

The main outcomes from this project are:

- A set of 16 benchmarks designed to test different forms of circuits and optimizations, adapted to three HLS tools: Dynamatic, Intel HLS and Vitis
- Procedures on how to build circuits with almost identical behavior over these three platforms, and onto AMD and Intel FPGAs
- A basic characterization script for buffer placement, based on an existing script designed for Vivado, and adapted to Quartus and Intel FPGAs
- A collection of data collected from the benchmarks from the three aforementioned tools

Data snippet:




## Future considerations

Due to time constraints, this project still leaves a lot of questions unanswered which would merit further study. The essential points are:

- More in-depth study of benchmarks where Dynamatic is outperformed by other tools
- Completing the characterization script to generate full delay tables
- Adapting the existing components in Dynamatic to use Quartus' built-in libraries for better compatibility
- Extending the benchmarks to other HLS tools, like Catapult HLS and Stratus HLS
- Adding benchmarks closer to real-world use cases