

Machine Learning in Digital Circuit Design

Author: Ondřej Galeta

Supervisor: prof. Ing. Lukáš Sekanina, Ph.D.

CAN A COMPUTER DESIGN ITS OWN CIRCUIT?

Digital circuits process information using logic operations and form the basis of modern computing hardware. Automated circuit design can explore thousands of possible implementations, but evaluating each candidate is computationally expensive.

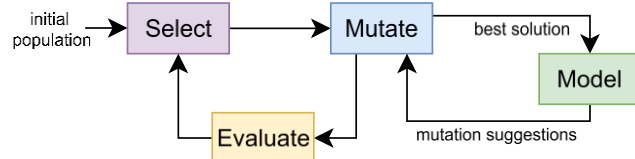
This thesis investigates whether **machine learning** can make **evolutionary algorithms for automated circuit design more efficient**. The results of this thesis were published in [1], presenting a transformer-based mutation operator for genetic programming.

CURRENT STATE

The circuit design process is formulated as an optimization problem. An **evolutionary algorithm**, such as **genetic programming**, starts with circuit designs and **mutates them to create new candidate designs**. These designs are evaluated, and the most promising ones are retained and mutated again in subsequent iterations. This approach is inspired by natural evolution: generate variations, evaluate them, keep the better solutions, and repeat. The main limitation is the number of designs that need to be tested. Circuit evaluation can be computationally expensive, and blindly exploring the design space wastes time.

TRANSFORMER-BASED MUTATION

The proposed approach adds **machine learning to the evolutionary process**. The proposed model is a **custom architecture** inspired by the Bidirectional Encoder Representations from Transformers (BERT) architecture [2] originally developed for large language models. It **learns from previously evaluated circuits** and uses this knowledge to **predict promising mutations**. Instead of exploring the design space blindly, the evolutionary algorithm can focus its effort where good solutions are more likely to be found.



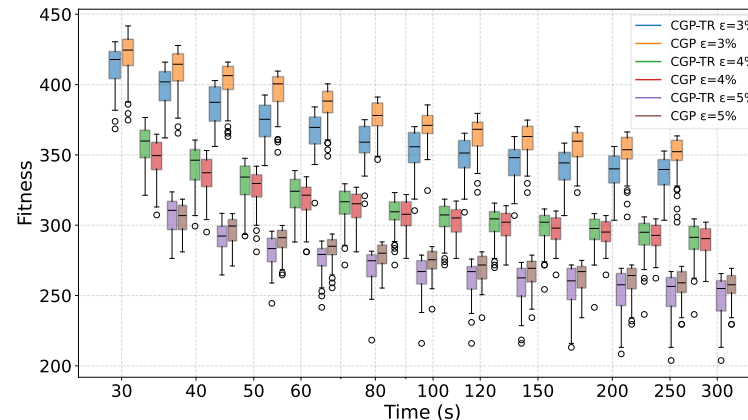
CASE STUDY: APPROXIMATE MULTIPLIERS

The approach was tested on **8-bit approximate multipliers**. Multiplication is a basic operation used throughout digital systems. An exact multiplier produces the mathematically correct result, but it can require considerable hardware resources.

Approximate computing takes a different approach: a small amount of error is deliberately allowed in exchange for lower hardware cost.

In this work, the focus was placed on **minimizing the on-chip area** of the multiplier while maintaining the **Worst-Case Error (WCE)** below the specified threshold.

This can be useful when a small numerical error has little effect on the final result, for example, in **deep neural networks**, **image** or **signal processing**. Especially for deep neural networks, this can significantly reduce computational costs, making **deep neural networks**, such as **LLMs**, **more affordable and accessible** to a wider part of society.



The plot above shows how the proposed CGP-TR method improves fitness, represented by the required on-chip area, faster than standard CGP for a given Worst-Case Error (WCE) threshold. Each box plot shows the fitness distribution across independent runs at a given point in time. For each method, 10% of the 576 runs are shown in the plot. CGP-TR generally reaches good circuit designs faster by guiding mutations towards promising regions of the design space. The improvement in convergence speed is statistically significant for WCE thresholds of 3% and 5%.

RESULTS

The experiments show that machine learning can effectively guide evolutionary circuit design. The **proposed approach reduces search time** by up to almost **60%** compared with conventional evolutionary search. For selected error constraints, it also achieves **comparable or slightly better trade-offs between circuit area and accuracy** than the reference designs [3], which are well established in academia.

This demonstrates that machine learning does not have to replace evolutionary algorithms. Instead, it can act as a **guide**, helping them decide which parts of the enormous design space are worth exploring.

WHY DOES IT MATTER?

Today, hardware is often designed for general-purpose use. Automated design opens another possibility: **circuits tailored to a specific application and its requirements**.

A designer could specify what the circuit should achieve, such as its required accuracy and maximum size, and an algorithm could search for a suitable implementation.

The thesis is a step towards combining **machine learning and evolutionary computation for automated hardware design**.

[1] Ondřej Galeta, Lukáš Sekanina, Genetic Programming with Transformer-Based Mutation for Approximate Circuit Design. In Proc. Of IEEE Congress on Evolutionary Computation (IEEE CEC), Maastricht, Netherlands, 2026.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proc. of NAACL-HLT'19, pages 4171–4186, 2019.

[3] Vojtěch Mrazek, Radek Hrbáček, et al. Evoapprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. In Proc. of DATE'17, pages 258–261, 2017.