

# High Level Synthesis Using Genetic Algorithm

Andrew Zakhary  
*Electronics Engineering*  
*Hochschule Hamm-Lippstadt*  
Hamm, Germany  
2210009

*Abstract—*

*Index Terms—*component, formatting, style, styling, insert

## I. INTRODUCTION

High level synthesis is a method of generating digital circuits automatically from a behavioral model. This behavioural model could describe a mathematical equation, the movement of a robot or a home appliance. the output of such process is split into four different categories [1]

- **Functional units (FUs)** these are responsible for implementing the logical functions.
- **Registers and memory units** these are the spaces where the data would be stored.
- **Switchable connections** these the busses and multiplexers that connect the memory units with functional units.
- **Controller** this is is the unit that is responsible for switching the switchable connections.

During the process of High level synthesis a lot of constraints have to managed. Some are fixed like number of pins, space available or time required for each operation while some others need optimization like power consumption and costs. Due to the complexity of the synthesis process process which arises because of the complexity of the model, multiple scheduling algorithms have been developed that try to optimize different resources like reducing the number of FUs or registers or reducing power consumption among many others. Since optimizing all factors at the same time is nearly impossible, algorithms must prioritize some over others. For example an algorithm can prioritize execution time and therefore may decide to sacrifice power consumption.

Genetic algorithm in this regard a more flexible solution as the priority of the fitness function can be changed to serve different functions. The way genetic algorithms are implemented usually is with population containing individuals that are created randomly. This population is then graded on a specific fitness function and the top performing individuals are selected for the next phase. The algorithm then creates a new population based on mixing "traits" of the successful ones to try to improve the overall fitness of the population from one generation to another. Some algorithms also include

## REFERENCES

- [1] E. Torbey and J. Knight. High-level synthesis of digital circuits using genetic algorithms. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 224–229, 1998.