

# Synthesis and Optimization of Digital Systems

## (02LVNOV)

### Low-power contest

Simone Baratta (s208978)  
Michele Iacobone (s207617)  
Francesco Giancane (s211327)

#### Abstract

The main goal of this contest is to perform static power optimization of combinational circuits, using the two benchmark designs c1908 and c5315 using the Dual- $V_{th}$  technique.

Since an exhaustive search was not viable, the only possible solution was to develop a heuristic algorithm, so we chose an approach based on genetic algorithms: this approach has the advantage that the static power consumption of the single cells is not necessarily taken into account.

In order to manage the cell mapping problem using a genetic algorithm, the individuals were chosen to be strings of 1's and 0's; each digit represents the threshold voltage to be used with a specific cell of the network; the fitness function is computed as the total static power consumption of the network, which makes it unnecessary to know the static power of the single cells.

## 1 The algorithm

The algorithm is written in TCL, which makes it easy to handle strings of digits (such as the representation of our individuals as strings of 0 and 1 characters); several procedures were developed, which are responsible for manipulating the most promising genes in semi-random ways.

At every iteration, the main loop chooses the 3 individuals having the best fitness value (i.e. static power consumption), combines their genes into 17 new individuals using these procedures (so that we have a total of 20 individuals), and then sorts the individuals again, before starting the next generation, according to their fitness values.

The algorithm is implemented in both a low-effort and a high-effort mode: in the low-effort mode, a check is added that stops iterating if the cost function doesn't improve over a certain number of generations (i.e. recombining genes doesn't yield any individuals with better fitness); the high-effort mode, on the other hand, performs a predefined number of iterations, with higher chances of stepping out of local minima.

## 2 Overall performance

The proposed approach is not as fast as other heuristic approaches, but it obtains remarkable results in terms of power saving; additionally, the algorithm is easily adapted to similar problems, by only modifying the fitness function and tweaking the number of kept/generated individuals at every generation.

As an example, running the algorithm on the c1908 benchmark achieved an arrival time of 1.0625ns with a static power saving of 96% compared to the original design.