Low-Power Contest Project Report, Group 27
Synthesis and Optimization of Digital Systems
June 2023

Nicolas Abril S313391

Pouyan Asgari S289607

Andrés Zambrano S311270

1 Explanation of the algorithm

We used a greedy statistical approach for selecting which cells to swap.

We start by changing all cells to HVT, which reduces power to the minimum possible, but also makes many cells violate timing constraints. To fix this we switch the most critical cells back to LVT in batches. Having all the cells start in HVT means that the moment that the timing constraints are met

The criteria for selecting which cells to swap in each iteration is based on the average slack of the most critical path of each HVT cell. If a cell has a slack lower than the average plus some margin then it is swapped to its LVT version.

After some tests we noticed that the slowest part was calculating the slacks for all cells, especially at the beginning when most of the cells are HVT and so we need to find the slack for many paths. This is why we decided to go with the average, the number of swapped cells starts large and then tapers to avoid overshooting and swapping cells that we didn't need to.

The margin comes to control what is the percentile that we're changing. Although we don't know the shape of the distribution of the slacks, by adding a fraction of the standard we can choose to have a larger or smaller percentage of cells be swapped at each step. The values that worked reasonably well in all circuits was average plus 30% of standard deviation on the first iteration and then average minus 30% of standard deviation for all the next ones. This approach doesn't necessarily work well for any design since the slack distribution changes from circuit to circuit, but it should work on a wide range since it's reasonably generic.

2 Pseudocode

```
dualVth (slackThreshold, maxFanoutEndpointCost):
swap all cells to HVT
while timing constraints are not met:
  get list of HVT cells
  get slack of most critical path of each cell
  get average and standard deviation of those slacks
  if first iteration:
    for each hvt cell:
      if cell slack < (average slack + 0.3*standard deviation):
        swap cell back to LVT
  else:
    for each hvt cell:
      if cell slack < (average slack - 0.3*standard deviation):
        swap cell back to LVT
```