Logic-Level Synthesis Contest 20/21

Claudia Golino s282847
Mattia Mirigaldi s288224
Stefano Palmieri s281677

September 2021

1 DualVth group10

This project consist in the implementation of a command written in TCL that run a post-synthesis power minimization procedure.
The command takes as input a negative slack and through dual-Vth cell assignment and gate re-sizing optimize the design by violating the slack up to maximum the negative slack provided as input.

SYNOPSYS COMMAND : *dualVth –allowed_slack allowed_slack_value*

The constraint that has to be met :

$$slack \geq allowed\_slack, allowed\_slack \leq 0 and total time < 15min \quad (1)$$

The optimization consist on maximize the score function :

$$Score = (\frac{area\_initial}{area\_final} + \frac{Pleakage\_initial}{Pleakage\_final} + \frac{Pdynamic\_initial}{Pdynamic\_final}) * (\frac{1 - cpu\_time}{900}) \quad (2)$$

The algorithm starts by identifying the HVT and LVT cells and save them respectively in two lists called *"LVT_cells"* and *"HVT_inital_cells"* and then :

- The function *"change_cells_to_HVT"* reorder the list LVT_cells according to a priority function called "compare_priority" that associate a priority to each cell in the list, priority that depends on the product between the slack and leakage power associated to the cell. The first element of the sorted list is the cell with highest priority that is then swapped to its HVT variation that gives best local score and met the slack constraint.
  The locale score is evaluated as :

$$Score = (\frac{C\_area\_initial}{C\_area\_final} + \frac{C\_Pleakage\_initial}{C\_Pleakage\_final} + \frac{C\_Pdynamic\_initial}{C\_Pdynamic\_final}) \quad (3)$$

Where C_area_initial, C_Pleakage_initial, C_Pdynamic_initial are the initial values associated to the LVT cell, while C_area_final, C_Pleakage_final, C_Pdynamic_final are the values of its HVT variation. This process is iterated until there are cells in the LVT_cells list or if the remaining cells have not a HVT variation that met the slack constraint.

- The remaining LVT cells and intial HVT cells are resized by calling the function *"resize_cell"* that chose the version of the cell with lower size than the initial one and that achieve higher local score.