Low-Power Contest 22/23

Synthesis and Optimization of Digital Systems

Write a TCL command for PrimeTime that implements a post-synthesis leakage power minimization procedure.

SYNOPSIS

`dualVth slackThreshold maxFanoutEndpointCost`

ARGUMENTS

`slackThreshold`: Endpoints with slack < `slackThreshold` are defined as ***violating endpoints***. Allowed values may range from 0 to 0.1 ns.

`maxFanoutEndpointCost`: Maximum fanout endpoint cost for each cell in the circuit. The fanout endpoint cost for a specific cell must be calculated as the sum of the differences between the `slackThreshold` and the slack of the most critical path passing through each ***violating endpoint*** within the fanout of that cell. Allowed values are greater than the maximum cell fanout endpoint cost of the initial netlist.

DESCRIPTION

The `dualVth` command runs a Dual-Vth (LVT and HVT) cell assignment to minimize the leakage power consumption while meeting the following constraints:

- The slack of the most critical path of the circuit is positive (>= 0).
- For each cell of the circuit, the fanout endpoint cost is < `maxFanoutEndpointCost`.
- Logic gates must keep the same cell footprint, i.e., same size and area, during the optimization loop.

EXAMPLE

```
dualVth 0.05 0.4
```

Note: only feasible values will be used for testing, namely, do not implement any feasibility check.

Evaluation

The best algorithm is the one that matches the constraints with minimum leakage power using the lowest amount of CPU time. Specifically, the following cost function will be used to evaluate the script (lower is better):

$$\text{Optimization Cost} = \sqrt{\left(\frac{P_{\text{leakage\_final}}}{P_{\text{leakage\_initial}}}\right)^2 + \frac{(\text{CPU Time})^2}{2000}}$$

where $P_{\text{leakage\_initial}}$ is the leakage power before the optimization, $P_{\text{leakage\_final}}$ is the leakage power after the optimization, CPU time is the difference between start-time and end-time (using the TCL **clock** command).

Basic Rules for the Competition

1. Combinational circuits used as benchmarks: {`c1908.v`, `c5315.v`}
   Note: the algorithm must be general and will be tested on other benchmarks, too.
2. The command will be executed under PrimeTime, with the script `pt_contest.tcl`
3. The initial gate-level netlist contains only cells from the `CORE65_LP_LVT` library. It is possible to assume that all paths of the netlist have slack >= 0.
4. All the groups must use the template `dualVth_Group_N.tcl` available on the webpage of the course. Other additional procedures can be used only if invoked within the `dualVth` procedure.
5. Scores:
   - groups that deliver a working script (constraints met) will get **3** points;
   - the best algorithm (lowest optimization cost) will get **3** extra points;
   - fake (and/or cut&paste) scripts will get **-3** points.

Each group will send an e-mail to **andrea.calimera@polito.it** and **valentino.peluso@polito.it** (in cc) using as subject <**SODS23 GroupN**> (N the ID of the group). Attached with the mail the following 2 files:

1. one single TCL file, titled <**dualVth_Group_N.tcl**>, containing the script code
2. 1 (one) page pdf, titled <**Group_N.pdf**>, which gives a brief description of the algorithm

***** DEADLINE Jun 20 (hh 23:59) *****

(late messages, or messages not compliant with the above specs, will be automatically discarded)