Low-Power Contest 19/20

Synthesis and Optimization of Digital Systems

Description

Write a command for PrimeTime that implements a post-synthesis power minimization procedure. The new command, written in TCL, runs a leakage-constrained *dual-Vth cell assignment* with *gate re-sizing*. The only input argument is the leakage savings to be reached after the dual-Vth assignment process; it is measured as follows:

$$savings = \frac{start\_power - end\_power}{start\_power}$$

where $start\_power$ is the leakage power consumption of the original circuit, while end_power is the leakage power consumption of the circuit after the optimization. The allowed values for the input argument $savings$ may range from 0 (no leakage minimization) to 1 (maximum leakage savings).

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

SYNOPSIS

```
dualVth -leakage $savings$
```

EXAMPLE

```
dualVth -leakage 0.5 ;#50% of leakage savings w.r.t. the loaded design
```

Evaluation

The best algorithm is the one that matches the leakage savings constraint while reaching the smallest slack and the minimal dynamic power penalty using the lowest amount of CPU time. The evaluation metrics are as follows:

1. compliance to the input constraint $savings$;
2. slack penalty due to leakage minimization (lower is better):

$$slack\_penalty = \begin{cases} 0 & if\ final\_slack \geq 0 \\ -final\_slack & if\ final\_slack < 0 \end{cases}$$

3. dynamic power penalty (lower is better):

$$dp\_penalty = \frac{final\_dp - starting\_dp}{starting\_dp} + 1$$

4. execution time, that is the difference between *start-time* and *end-time* - tcl **clock** command (lower is better).

Note: the algorithm must be general also in terms of timing constraint, that means the original circuit can be synthesized with any clock period. It is possible to assume all the paths of the original circuit have slack >= 0.

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, just after the script `pt_analysis.tcl`
3. The benchmark is first synthesized under a fixed timing constraint (e.g., `clockPeriod`= 3.0 ns) using the `synthesis.tcl` with a single-VT target library, the `CORE65_LP_LVT`.
4. All the groups are invited (mandatory) to use the template 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 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 **antonio.cipolletta@polito.it** (in cc) using as subject <**SODS20 GroupN**> (N the ID of the group). Attached with the mail the following two (2) files:

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

***** DEADLINE July 5 (hh 23:59) 2020*****

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