# Low-Power Contest: Slack-Driven Dual-Vth Assignment

## Synthesis and Optimization of Digital Systems
## A.Y. 2015/16

Write a new TCL command to be integrated within Design Compiler that performs a Slack-driven Dual-Vth post-synthesis optimization. Such a command reduces leakage power by means of dual-Vth assignment while forcing the number of quasi-critical paths below a user-defined constraint. The figure below pictorially describes the behavior of the procedure.

Main arguments of the command are:

- `-arrivalTime` → the actual timing constraint the circuit has to satisfy after dual-Vth assignment [ns]
- `-criticalPaths` → the total number of timing paths that fall within a given slack window after the dual-Vth assignment [integer]
- `-slackWin` → is the slack window for critical paths [ns]

The command returns the list `resList` containing the following 4 items

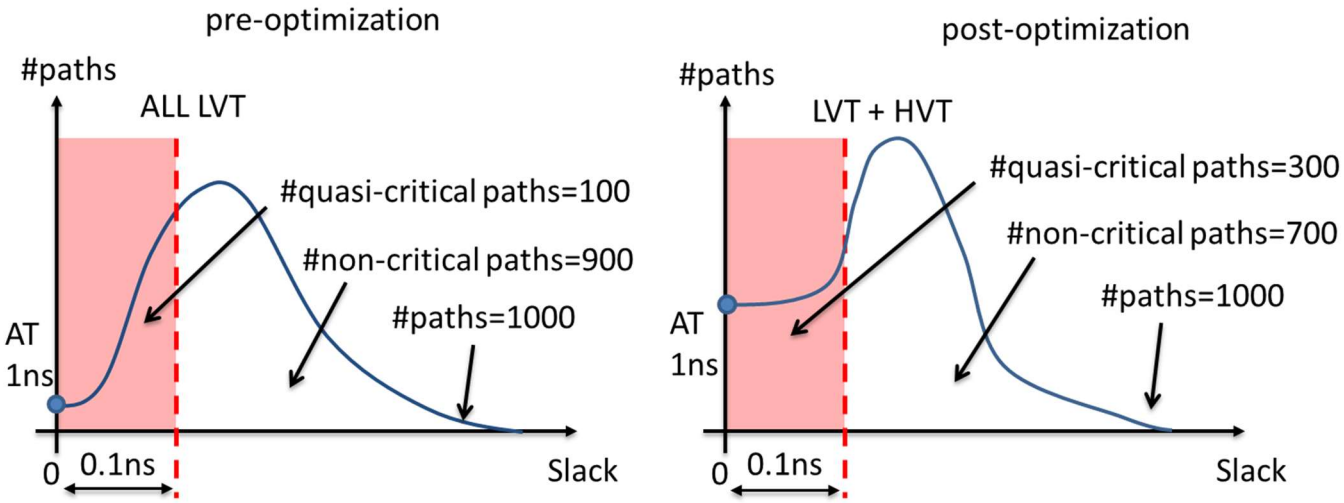
item 0--> `power-savings`: % of leakage reduction w.r.t. the initial configuration;
item 1--> `execution-time`: difference between starting-time and end-time [seconds]*.
item 2--> `lvt`: % of LVT gates
item 3--> `hvt`: % of HVT gates

**Note: use the tcl* ***`clock`*** *command*

### *SYNOPSIS*

```
$resList$ leakage_opt -arrivalTime $at$ -criticalPaths $num$ -slackWin $time$
```

```
res leakage_opt -arrivalTime 1 -criticalPaths 300 -slackWin 0.1
```



**Basic Rules for the Competition**

1. Combinational circuits used as benchmarks: `c1908.v c5315.v`.
2. The command will be executed under Design Compiler, just after the script `synthesis.tcl`
3. The benchmark is first synthesized under a fixed timing constraint (e.g., `clockPeriod` = 3.0 ns) using the `compile` command; the obtained netlist is used as starting point for the `leakage_opt` command. Please notice `clockPeriod` and `arrivalTime` may differ (the script should identify and report unfeasible conditions).
4. Only the numbers returned by the command, through the `resList`, will be considered for the competition.
5. The structure of the script *must* satisfy the synopsis described above (name of the in/out parameters and units of measure).
6. Scores:
   - groups that deliver a working script (slack constraint satisfied) will get 3 points;
   - the algorithm getting the largest power-savings with the minimum CPU time will get **3** extra points;
   - fake (and/or cut&pasted) scripts will get **-3** points.
7. All the scripts will be evaluated using the following cost function

$$S = \mathrm{sqrt}[(1-X)^2+Y^2]$$

with `X` the percentage of power savings and `Y` the CPU time [seconds].

Each group must send a mail to **andrea.calimera@polito.it** and **valerio.tenace@polito.it** (in cc) using as mail subject <**SODS16 GroupN**> (where N is the ID of the group, see attached .xls file). Attached with the mail the following 2 files:

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

***** DEADLINE July 17 (hh 00:00) *****

(late messages will be automatically discarded)