ISPD 2012 Evaluation Program

Last modified: January 29, 2012

Disclaimer

- The information published in this presentation is subject to change.
- It is the contestants responsibility to check the website frequently to check for any updates until the submission deadline.
- Although we did our best to test it, the provided program may still contain errors. Please let us know ASAP if you discover any issues with this program. Please continue to check the announcements in case there are any fixes.

Evaluation Program

- A .cpp program is being provided to help contestants to compute the number of violations in each solution
  - The same program will be used to compute violations for the final evaluation
  - The results obtained from the evaluation program are golden
  - The .timing file will be generated using PrimeTime as explained before

Compilation

- Evaluation program is composed of three files:
  - parser_help.cpp
  - parser_help.h
  - validate.cpp
- "parser_helper.h" and "parser_helper.cpp" are the same files that were released before with the benchmarks
- Compilation is made by the following command:
  - g++ parser_help.cpp validate.cpp –o validate

Command line

- ./validate <benchmark folder> <library>
- Validate program will look for the required benchmark files inside the benchmark folder, e.g.:
  - Command line:
    - ./validate simple contest.lib
  - Required files:
    - ./simple/simple.v
    - ./simple/simple.sdc
    - ./simple/simple.spef
    - ./simple/simple.sizes
    - ./simple/simple.timing
    - ./contest.lib

Rounding

- Violations are rounded to the second decimal place
- Each violation is rounded individually. Total violation is the sum of each rounded violation, e.g.:
  - Slew violation list:
    - U1.a -10.4142 is rounded to -10.41
    - U2.b -12.1047 is rounded to -12.10
    - Total slew violation is -22.51

Output

- The evaluation program will print on the screen the sum of slack, slew and capacitance violations as well as the total power and total sum of all violations
- A different file is created to store the detailed report of each type of violation. Inside each file all individual violations are reported
- “./validate simple contest.lib” generates:
  - simple/simple.cap_violations
  - simple/simple.slew_violations
  - simple/simple.slack_violations