Changed Feb-8-2013

# ISPD 2013 Contest Evaluation

Last modified: February 8, 2013

Changed Feb-8-2013

# 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.

# Benchmarks

- For the final evaluations, we are planning to use a subset of the released benchmarks with some modifications (e.g. different timing constraints and/or parasitics, modifications in the netlists). We may also use additional benchmarks that are not released.
- The final cell library file (.lib) that we plan to use in evaluations is available on the contest website.

* The exact subset will be decided based on the number of final submissions

# Output file

- When your sizer is run, it is expected to produce a <benchmark>.sizes file
  - The .sizes file format is defined in "ISPD_2013_Contest_Details.pdf" presentation on "Sizer Output (.sizes) File" slide
  - Each line is defined as:

    <full-instance-name>  <library-cell-name>
- Logic transformations are not allowed
  - **WARNING: Only cells with the same cell_footprint name can be swapped. For more details about swapping group, refer to slide "contest.lib File Example" in "ISPD_2013_Contest_Details.pdf"**
  - **WARNING: You must NOT use the function field in .lib file to determine which cells can be swapped.**

# Contest Evaluation

- Two separate rankings:
  - Primary ranking: Solution quality will be the main metric. Runtime will be used for tie-breaking.
  - Secondary ranking: Both solution quality and runtime will be important. Multi-core implementations are encouraged!
- Each submission will be run once for each ranking.
  - For the secondary ranking the submissions will be run with the "-fast" option after the mandatory arguments.
  - Submissions have to support "-fast" command line option, otherwise they will not be considered for the secondary ranking.
- There will be a hard runtime limit for each benchmark
  - The runtime limit will be  smaller for the secondary ranking runs.

# Violations

- Violations are the primary evaluation criteria for both rankings
- Violations are divided into three different types
  - Negative slack (ps)
  - Maximum capacitance (fF)
  - Slew (ps)
- All violations are added together into a single number
- All benchmarks can be sized without any violations

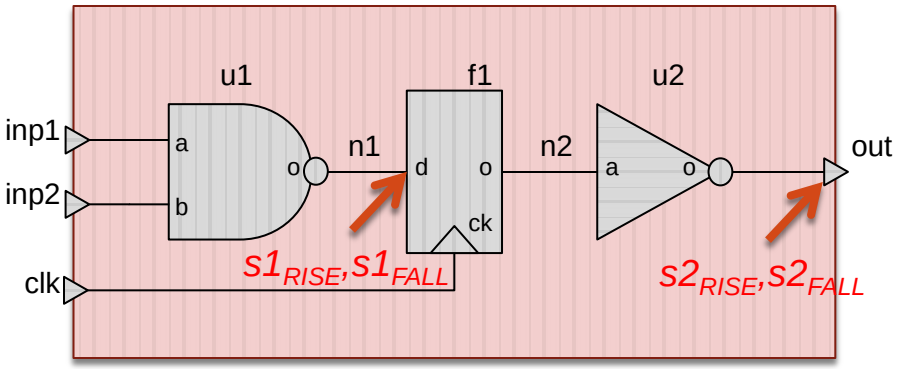
# Violations

- Negative slack violations:
  - Measured for both rising and falling transitions at the primary outputs and sequential inputs
  - See contest details slides for slack computation
  - In the simple.v example the slack variables that have to be observed are:

$s1_{RISE}$ = 10ps
$s1_{FALL}$ = -3ps
$s2_{RISE}$ = 1ps
$s2_{FALL}$ = -5ps

Total slack violation = ∑negative slacks
Total slack violation = 3+5 = 8 ps

Simple.v circuit



# Violations

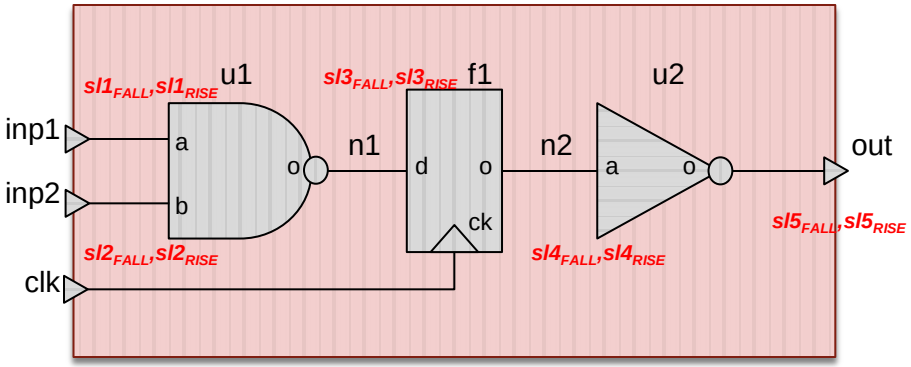
- Slew violation:
  - Measured on input pins for all cell instance and primary outputs (for both rise and fall transitions)
  - Slew limit is defined by the default_max_transition field in the .lib file

Slew violations:
Slew_limit= 100 ps

| | |
|---|---|
| $sl1_{FALL}$ = 97ps | $sl1_{RISE}$ = 95ps |
| $sl2_{FALL}$ = 99ps | $sl2_{RISE}$ = 103ps |
| $sl3_{FALL}$ = 101ps | $sl3_{RISE}$ = 112ps |
| $sl4_{FALL}$ = 75ps | $sl4_{RISE}$ = 89ps |
| $sl5_{FALL}$ = 42ps | $sl5_{RISE}$ = 52ps |

Total slew violation = 16ps

Simple.v circuit

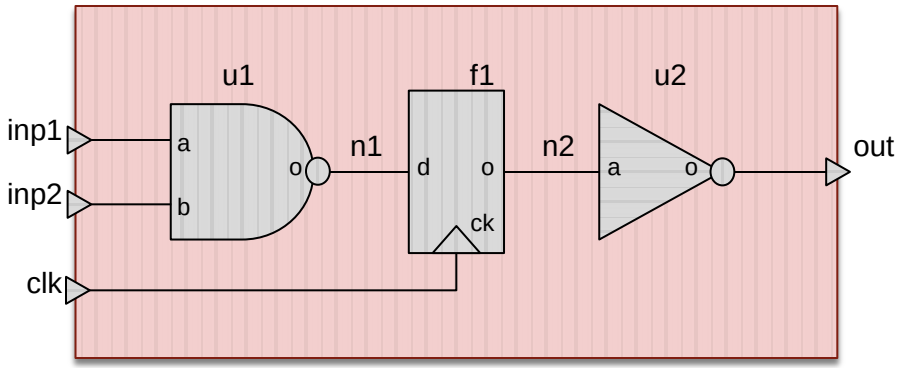


# Violations

- Output capacitance per cell:
  - Will be measured once per cell instance output and for each circuit input
    - Includes driving cells for primary inputs as defined in the .sdc file
  - The capacitance at the output of a gate is given by the effective capacitance (Ceff) seen at that pin. Ceff is computed by PrimeTime®, please check “ISPD_2013_Contest_Details.pdf” for information on how to compute it.
  - *Violation = max(0, (ceff(cell.output_pin) - maxcap(cell.output_pin))*

**Ceff per output pin**
cap(inp1)=27fF
cap(inp2)=20fF
cap(f1.o)=126fF
cap(u1.o)=151fF
cap(u2.o)=53fF
**Maximum capacitance allowed per cell:**
Max_cap(inp1)=60fF
Max_cap(inp2)=60fF
Max_cap(f1.o)=100fF
Max_cap(u1.o)=150fF
Max_cap(u2.o)=150fF
**Maximum Capacitance violations:**
violation(inp1) = max(0,(27fF-60fF) = 0
violation(inp2) = max(0,(20fF-60fF) = 0
violation(f1.o) = max(0,(126fF-100fF) = 26fF
violation(u1.o) = max(0,(151fF-150fF) = 1fF
violation(u2.o) = max(0,(53fF-150fF) = 0
**Total max_cap violation = 27fF**

Simple.v circuit

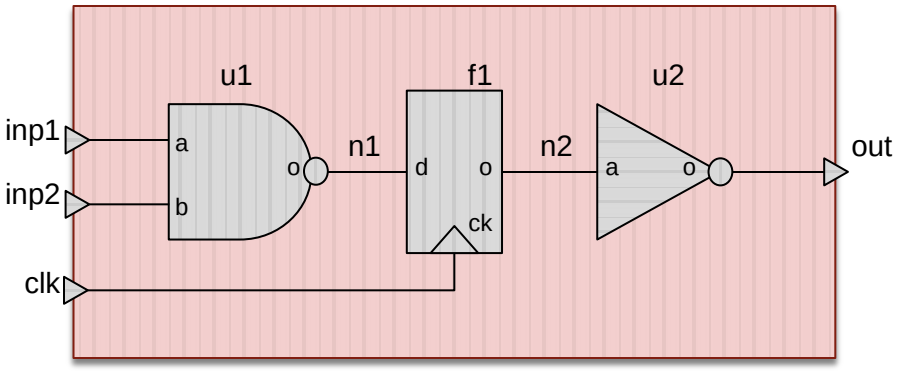


# Power

- Only leakage power is considered
- The leakage power value for each cell is given in the .lib file
- Total leakage power value is given by the sum of the leakage power for each cell

Total Power computation
Power(u1)=143uW
Power(f1)=313uW
Power(u2)=50uW
**Total Power = 134uW + 313uW + 50uW**
**Total Power = 497uW**

Simple.v circuit



# Sizer/Timer Interaction

- The teams must communicate the organizer what model option Sizer/Timer communicatino they are going to use
- If the organizers are not informed that PrimeTime® is needed it will not be started prior to the sizer execution.
- Teams who choose to use “Options 3” are responsible for starting/closing PrimeTime® and loading their own scripts.
  - The runtime will be measured from the time sizer execution starts to the time the last process started by the user is terminated.
  - Failure to terminate user process will increase measured runtime.

Changed Feb-8-2013

# Runtime

- The runtime is computed for each benchmark
- Time to load the design in PrimeTime® in the beginning **will be included** in the runtime measurement.
  - Runtime is the wall clock time from the beginning to the end of the execution of the submitted executable
  - Runtime includes not only the submission execution time but the time taken to load the design in PrimeTime® as well as the time to complete any PrimeTime® calls.
    - For the teams who select “Option 2” for sizer/timer interaction, the circuit will be loaded in PrimeTime® prior to the binary execution. The runtime measurement will include this time.
  - If no calls are made to PrimeTime®, runtime will include only the submission execution time
- All jobs running after the runtime limit is reached will be killed

# Runtime Limit

- The runtime limit will be defined **per benchmark**
  - Runtime limit is defined based on the number of cells for each benchmark according to the equation:

$$Runtime_{PRIMARY} = 3h + 1h \times Roundup\left(\frac{\#\,gates}{40K}\right)$$

  - Runtime limit is determined by 3 hours plus 1 hour per each 40K gates for each design
  - The number of gates divided by 40K will be rounded up to the next integer number
    - i.e., 2.6 is rounded up to 3, 4.1 is rounded up to 5
  - Runtimes are given in hours

# Runtime Limit: Secondary Metric

- The runtimes for the secondary metric runs will be equal to 1/5 of the runtime used in the primary metric:

$$Runtime_{SECONDARY} = Roundup\left(\frac{Runtime_{PRIMARY}}{5}\right)$$

- Runtime for the secondary metric will be rounded up to the nearest **minute**.
  - For example, if runtime primary is equal to 7 hours the runtime secondary will be 84 minutes.

# Primary ranking: Quality

- The ranking metric for a benchmark is defined in lexicographic order as:
  - First: ∑violations
  - Second: ∑power (when violations are tied)
  - Third: Runtime (when violations and power are tied)
- Sum of the ranks for each benchmark will define the final score for each team
  - If there is a tie at the end, it is broken using the sum of violations, power and runtime over all benchmarks in the same lexicographic order (example in next slides)

# Primary ranking: Quality

- Example: 3 teams, 2 hypothetical benchmarks
- Execution summary for each team on each benchmark
  - benchmark01:

| Team | Violations | | | | Total Power | Runtime |
|---|---|---|---|---|---|---|
| | Max Cap | Slew | Negative Slack | Sum | | |
| Team01 | 0 | 0 | 0 | 0 | 12.6 mW | 31m 42s |
| Team02 | 0 | 0 | 0 | 0 | 11.1 mW | 1h 12m 21s |
| Team03 | 12fF | 130ps | 10ps | 152 | 7 mW | 10s |

  - benchmark02:

| Team | Violations | | | | Total Power | Runtime |
|---|---|---|---|---|---|---|
| | Max Cap | Slew | Negative Slack | Sum | | |
| Team01 | 10fF | 0 | 0 | 10 | 5.4 mW | 13m 14s |
| Team02 | 0 | 0 | 0 | 0 | 7 mW | 15m 55s |
| Team03 | 0 | 10ps | 0 | 10 | 5.4 mW | 1m 49s |

# Primary ranking: Quality

- Below are the ranking for each benchmark and final ranking, the criteria used to decide on the ranking is highlighted on each case
  - Rankings for each benchmark

| benchmark01 | | | | |
|---|---|---|---|---|
| **Team** | **Violations** | **Power** | **Runtime** | **Rank** |
| Team01 | 0 | 12.6 mW | 31m 42s | #2 |
| Team02 | 0 | 11.1 mW | 1h 12m 21s | #1 |
| Team03 | 152 | 7 mW | 10s | #3 |

| benchmark02 | | | | |
|---|---|---|---|---|
| **Team** | **Violations** | **Power** | **Runtime** | **Rank** |
| Team01 | 10 | 5.4 mW | 13m 14s | #3 |
| Team02 | 0 | 7 mW | 15m 55s | #1 |
| Team03 | 10 | 5.4 mW | 1m 49s | #2 |

  - Final ranking

| Final | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Team** | **benchmark01** | **benchmark02** | **Rank Sum** | **Total Violations** | **Total Power** | **Total Runtime** | **Final Rank** |
| Team01 | #2 | #3 | 5 | 10 | 18 mW | 44m 56s | #2 |
| Team02 | #1 | #1 | 2 | 0 | 18.1 mW | 1h 28m 16s | #1 |
| Team03 | #3 | #2 | 5 | 162 | 12.4 mW | 1m 59s | #3 |

# Secondary ranking: Quality/Runtime

- The secondary ranking evaluates the solution that presents the best quality/runtime trade-off
- Violations are still the primary metric, all the solutions with the same number of violations are ranked by:

$$cost = Power \times \left( (1-\gamma) + \gamma \frac{Runtime}{Runtime_{REF}} \right)$$

Changed Jan-30-2013

- This metric trades quality by runtime improvement with respect to a reference runtime value
- If there are ties those will be broken using the same criterion applying to the Primary metric (Violations, Power, Runtime)

# Secondary ranking: Quality/Runtime

- **$\text{Runtime}_{\text{REF}}$ is fixed** and defined by half of the runtime limit.

$$Runtime_{REF} = \frac{Runtime_{SECONDARY}}{2}$$

Changed Jan-30-2013

- Gamma ($\gamma$) is 0.05
  - e.g., Using the full runtime limit will represent an increase in 5% in the solution cost. A runtime close to 0 represents a reduction of close to 5% in the final cost.

Changed Jan-30-2013

# Secondary ranking: Quality/Runtime

- Rankings for each benchmark. Orange cells indicate reference values used for trade-off computation. Yellow cells are the values used to decide ranking positions.

Changed Jan-30-2013

**Benchmark01 – 1h Runtime$_{REF}$**

| Team | Violations | Power | Runtime | Cost | Rank |
|---|---|---|---|---|---|
| Team01 | 0 | 12.6 mW | 31m 42s | 12.30285 | #2 |
| Team02 | 0 | 11.1 mW | 1h 12m 21s | 11.21285 | #1 |
| Team03 | 152 | 7 mW | 10s | 6.650972 | #3 |

Changed Jan-30-2013

**Benchmark02 – 10min Runtime$_{REF}$**

| Team | Violations | Power | Runtime | Cost | Rank |
|---|---|---|---|---|---|
| Team01 | 10 | 5.4 mW | 13m 14s | 5.4873 | #3 |
| Team02 | 0 | 7 mW | 15m 55s | 7.207 | #1 |
| Team03 | 10 | 5.4 mW | 1m 49s | 5.17905 | #2 |

**Final**

| Team | benchmark01 | benchmark02 | Rank Sum | Total Violations | Total Power | Total Runtime | Final Rank |
|---|---|---|---|---|---|---|---|
| Team01 | #2 | #3 | 5 | 10 | 18 mW | 44m 56s | #2 |
| Team02 | #1 | #1 | 2 | 0 | 18.1 mW | 1h 28m 16s | #1 |
| Team03 | #3 | #2 | 5 | 162 | 12.4 mW | 1m 59s | #3 |

# Evaluation rounds

- The organizers reserve the right to do evaluations in rounds.
  - Evaluations may be run for all teams on small benchmarks
  - For top qualifying teams, evaluations will be run on large benchmarks.
- Evaluation rounds will only be performed if there is not enough time to run all benchmarks for every team.

# Same ranking

- If 2 teams have the same ranking:
  - Both will have the rank equal to the number of teams that have a better ranking + 1
  - The teams following those will also be ranked according to the number of teams that have a better ranking

Example:
TeamB and TeamC are tied
TeamE, TeamF and TeamG are also tied

All tied teams have the same ranking
The ranking of a team is always equal to the number of teams in front of it in the ranking

| Rank | Team |
|---|---|
| 1 | TeamA |
| 2 | TeamB |
| 2 | TeamC |
| 4 | TeamD |
| 5 | TeamE |
| 5 | TeamF |
| 5 | TeamG |

# Error handling

- During timing iterations when using PrimeTime®, any invalid cell swapping will be ignored by PrimeTime® (i.e. cell size from previous iteration will continue to be used)
- If the .sizes files is missing, the final evaluation will be done using the original cell sizes from verilog file
  - Any missing cell in the .sizes file will have its size unaltered
    - If cell C is not included in the final .sizes file, then the original size of C in the verilog file will be assumed
- For final evaluation, any invalid cell swapping will be ignored and the original cell size from the verilog file will be used
  - Some examples of invalid cell swapping are:
    - An invalid cell size in the .sizes file
    - Swapping cells with different cell_footprint names

# Scripts

- Evaluation scripts will be provided.
- Scripts will:
  - Compute #violations
  - Compute power
  - Check solution correctness

# Submission Files

- Each team is allowed to submit at most 3 files based on the timer/sizer interaction model.
  - 1 binary executable
  - 1 shell script (bash, csh, sh or tcsh)
  - 1 tcl command file
- No data files are allowed
- The shell script can be user as a wrapper to load the tcl command file on PrimeTime® and to start the sizer
  - In this case the shell script has to be named sizer and it has to be able to be run as specified on the "ISPD_2013_Contest_Details.pdf" slides.
  - It will also be required to handle the –fast option for the Secondary metric evaluation

# Submission Files

- The required names for each file depends on the timer/sizer communication options:
  - Option 1 & Option 2
    - 1 binary executable: file name must be “sizer”.
    - The shell script and the TCL script are not allowed.
  - Option 3
    - 1 binary executable: file name must be “sizer” **if no shell script is submitted.**
    - (Optional) 1 shell script (bash, csh, sh or tcsh): file name must be “sizer” and the binary should be renamed to “sizerbin”.
    - (Optional) 1 tcl command file: file name must be “sizer.tcl”.
- In spite of the option you choose your submission will be run as:

```
sizer $ISPD_CONTEST_ROOT <benchmark>
```

`sizer $ISPD_CONTEST_ROOT <benchmark> -fast` (for fast mode)

Added Feb-8-2013

# PrimeTime® Licenses

- 5 PrimeTime® licenses will be available for the teams choosing Option 3.
  - 5 PrimeTime® runs can be launched in parallel
- Teams choosing Option 2 will use a single PrimeTime® license.

# Alpha submissions

- It is the contestant responsibility to guarantee that the submission works in the contest evaluation environment.
  - We will allow alpha executable submissions for that purpose
- Alpha binaries will be run by the contest organizers on a single benchmark
  - The teams may request a specific benchmark on which the executable will be run
  - Alpha runs will not be performed on benchmarks with a runtime limit larger than 7 hours

# Alpha submissions

- No debugging will be done by contest organizers.
- Execution log files will be sent to the contestants.
  - We are unable to send the log files to the teams that did not sign the Synopsis NDA.
- Each alpha submission period comprises **a single run.**
  - Each team can submit at most one alpha executable in a submission period.
  - Contest organizers will do their best to send back the alpha executable results within 3 business days

# Alpha submission calendar

February

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| | | Alpha submission round 1 | | | | |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | Alpha submission round 2 | | | | |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | Alpha submission round 3 | | | | |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 * | 25 | 26 | 27 | 28 | 1 | 2 |

Changed Feb-8-2013

* Final submission deadline

# Mandatory alpha submission

- All teams must submit at least 1 alpha executable
- Teams that do not submit any alpha executable by February 16th **will be removed from the contest.**

# Final submission deadline

- Final submission deadline is February 24th.
- **Final submissions will be accepted until February 24th, 11:59pm (PST)** Changed Feb-8-2013
- Teams that do not submit the final executable **will be removed from the contest.**
  - Alpha submissions will not be considered as the final submission.

# Machine Specification

- Evaluations will be run on a machine with the following configuration:
  - 64 bits machine
  - #cores: 16
  - Memory: 49152Mb
  - Processor speed: 2.93GHz

# Intel Legal Agreement

- Contestants should have received the Intel Legal Agreement by now.
- To run your binaries we need your agreement to the terms in the Legal Agreement.
- The first time you send an executable to us, please attach the legal form and explicitly state that you agree to the terms in the attached document.