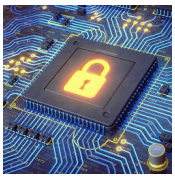

Security Closure of Physical Layouts

ISPD 2022 Contest

Q&A

Last updated: Wed Mar 30 00:16:03 EDT 2022

This page will be updated with Q&A once they come in; check back regularly. Important Q&A feedback will also be announced as well as shared via email to registered participants.

Q&A: Scoring, Evaluation, and Benchmarks

Q44: May I ask if we need to keep the best submission only in each test case's folder (deleting other submissions)?

> **A44:** No, you don't need to do that. We will consider all past submissions and extract the best from there.

Q43: May I ask for the public final benchmarks, will there be a submission limit?

> **A43:** No, neither for the public nor the blind benchmarks there's a limit. But we're limiting the number of parallel runs per team, and if the backend gets too flooded still we might need to take action as well.

Q42: What is the difference between a score of 0.00001 and 0.00000?

> **A42**: All zero means perfect score on all score components, whereas anything like 0.00001 or 0.00002 means that some score component is not perfect yet (see the scores.rpt for details). This will impact the ranking accordingly. Remember that design quality is included for the overall score as well, so

achieving perfect scores for ti and fsp_fi may still lead to this residue for the overall score.

Q41: I could get my score on bench openMSP430_2, but I couldn't get the score of bench SPARX for a long time. Is there a problem with the backend or just taking a long time?

A41: No, the backend works fine — that's part of the challenge for SPARX, where you'd want to look into the exploitable regions first (at high level is good enough, i.e., understand what's going on for regions here versus other benchmarks) and then take it from there.

Q40: In the new probing result, I find the PG net is not contribute to the exposed area. But the detail score evaluation in the website don't exclusive PG net. So is it a bug of the probing scripts or a new demand?

A40: Good observation, thanks. This is fixed by now (March 24th).

Q39: We noticed that when we submit the baseline PRESENT design, it will report check failure on metal2.

A39: This issue is fixed by now (March 20th). Also note that the PG constraints have been relaxed recently.

Q38: we want to know if the evaluation platform will be closed when the hidden cases are released and if the alpha round results will be updated based on the new scoring metric.

A38: The platform will remain open for some time after the blind benchmarks are released, but we might limit how many submission each team has. We'll announce once that's decided.

Yes, we'll update the alpha-round results considering the new scoring, but would await some more pending fixes for the evaluation backend before doing that.

Q37: It seems that sometimes the evaluation platform will not unzip correctly. It takes a zip file as the processed file and report an error.

A37: We don't keep logs for the subroutines like unzip, so I'm not sure what went wrong exactly. It might be that the ZIP file is not properly packed — can you check to unzip at your end? Another issue could be that the file was still uploading to Google Drive when it was downloaded, but I'm not sure right now

if that can really happen (as that would mean Google Drive lists files already when they're still uploading, which would not be good practice from their end.) You could also double-check the MD5 from the processed_files report, to see if there's any mismatch.

Q36: From my past experience in detailed routing, a via consists of three parts, including two via metals on two metal layers and cut shapes on the cut layer. In a most trivial case where a via appears at the end of a wire, its via metal will normally extend out a bit than its connected wires. Therefore, I think those via metals should be considered as well since they are not always completely covered by a wire. And I am not sure whether in the evaluation backend something similar is done.

A36: We do consider those vias metals but with our issue of lack of accuracy, it wouldn't make a difference currently. For cell exposure, all these metals will contribute to blocking, i.e., reduce exposure. Keep in mind that only the blockage itself is relevant, so any overlap of metals wouldn't be accounted for multiple times. For net exposure, that also holds true: if some area at the top metal is already exposed, the via and bottom metal in that particular via area is blocked by the net itself, so won't be counted again.

Q35: it seems that those design issues like "Undriven pins issues", "Open output ports issues", "Net output floating issues" are not very related to "physical layout" since you mentioned that they are actually from logic synthesis.

A35: These issues are originally arising from synthesis, that's correct, but they also directly relate to and impact the physical layout. (Layout optimization will play a role as well, as to which degree each issue is carried over in the DEF or not.) All these issues are relevant for both Trojan insertion and frontside probing, in terms of scoring some defense or rather penalizing some trivial defenses.

Q34: we have started to work on the final designs and noticed something strange with the MSP benchmark. These pins on the lower left corner appear to have no connections to the core. Their names are dbg_i2c_broadcast[*] , dbg_i2c_addr[*] , dbg_i2c_scl. Is this behavior intended?

A34: Right, that's something I had noticed as well that these pins are not connected. FYI dbg_i2c_sda_in is another one. I think that's related to the default synthesis settings provided by this opencores design, where not all dbg features are included. Since these pins are not related to any cell or net assets, we didn't think of fixing those issues, so I guess you could say it's loosely intended.

Q33: Could you tell me how to evaluate the final overall score for all benchmarks? I mean, the score for all benchmarks are going to be simply averaged or multiplied? How do you decide the winner for the contest?

> **A33**: The score for all benchmarks would be averaged, and most likely using a simple non-weighted average. Alternatively we might use some weights to emphasize on complex design, but we'd announce that in time to all if so.
>
> There might also be considerations for emphasizing design cost further the moment we see very good submissions w/ security metrics optimized to 0 (as indicated in the recent email to all).

Q32: About the timing report, we tried to learn about the update_io_latency command. May I ask why it is called before "set_db timing_analysis_type ocv" and "set_db timing_analysis_cppr both" instead of being called after we set the timing_analysis attributes? I noticed that when we ran update_io_latency, it will print the current analysis mode. That is why I guess the behavior of update_io_latency might be affected by the timing_analysis_type. And if we have decided the timing analysis-related setting, why don't we let innovus know it earlier (this is the reason why I am wondering if the commands should be swapped)?

> **A32:** Our understanding is that update_io_latency is related to clock propagation and evaluation, and is not related to timing analysis as such, so I think the order we have is correct. Note that update_io_latency is a wrapper command performing multiple things related to latency, and setting/initializing the timing is only one.
>
> In general, the order of commands plays a major role, but this is not always easy to figure out and decide what's best, especially w/ different commands doing similar but not exactly the same things as well as different behaviour/results across different versions. For example, in the Innovus reference it's also mentioned that we shouldn't use set_propagated_clock again, but we do this within the legacy script we use (shared by some team) and I don't see any difference for results w/ or w/o that command being called after update_io_latency.
>
> With the flow we currently have, we have observed the smallest differences/mismatches for the DEF-based evaluation and the original evaluation during design runs, but there's still some mismatch.

Q31: Since we only have a very brief summary (checks_summary.rpt) returned from the evaluation system, may I ask that if we want to do the equivalence checking on our side to know more about those issues (we have Cadence Conformal installed),

how can we do that? Like what files are needed and what commands are used to report those design issues?

A31: Sure, this is the LEC/Conformal script which we employ:

```
//// lec_64 -nogui -xl -dofile lec.do
////
//// template derived from "Sample Dofile" from "Conformal Equivalence
Checking User Guide"

// setup
set parallel option -threads 4

read library -both -liberty NangateOpenCellLibrary.lib
read lef file NangateOpenCellLibrary.lef

read design -golden _design_original.v
read design -revised design.v

// Enter LEC mode but w/o auto mapping
set system mode lec -nomap

// To specify pipeline retiming, requires Conformal XL license
analyze retiming

// Map key points automatically
map key points

// Analyzes the netlists and sets up the flattened design for accurate
comparison
analyze setup

// To specify datapath analysis, requires Conformal XL license
analyze datapath -merge

// To run key point comparison
add compare point -all
compare

// reports
report verification > check_equivalence.rpt
echo >> check_equivalence.rpt
report statistics >> check_equivalence.rpt
echo >> check_equivalence.rpt
report unmapped points >> check_equivalence.rpt
echo >> check_equivalence.rpt
report compare data >> check_equivalence.rpt
// NOTE redundant report but helps for parsing
report unmapped points >> check_equivalence.rpt.unmapped

// mark done; exit
date > DONE.lec
exit -force
```

Note that some issues are also evaluated and reported by Innovus. For more details, note the following mapping of terms/issues of ours to LEC and Innovus issues:

- Ours -> LEC
  - Equivalence issues -> Non-equivalent points, as reported by “report compare data” command
  - Unreachable points issues -> Unreachable points, as reported by “report unmapped points” command
  - Undriven pins issues -> “Warning: (RTL2.13) Undriven pin is detected” as reported during parsing by “read design” command
  - Open output ports issues -> “Note: (HRC3.5b) Open output port connection is detected” as reported during parsing by “read design” command
  - Net output floating issues -> “Warning: (RTL14) Signal has input but it has no output” as reported during parsing by “read design” command
- Ours -> Innovus
  - Basic routing issues -> As reported into *.conn.rpt file by “check_connectivity” command
  - Module pin issues -> As reported into *.checkPin.rpt file by “check_pin_assignment” command
  - Unplaced components issues -> As reported by “check_design -type route” command
  - Placement and/or routing issues -> As reported by “check_design -type route” command
  - DRC issues -> As reported into *.geom.rpt by “check_drc” command

Q30: (from organizers) We noticed that, for probing of cells, there might be cases where cells have >100% of their area exposed.

> **A30:** This issue is fixed now (Mar 8th). FYI this relates to some trade-off for evaluation runtime and accuracy, and we’re further looking into this issue in general. Thus, numbers for probing evaluation may still change/alternate; we will update everyone once that’s settled.

Q29: We have submitted a version of MISTY where we fixed all des_issues, but the scoring system returns 87 issues (same as in the baseline). We verified that if we unfix one of the issues, the scoring system is able to catch it and returns 1. But something goes wrong if there are no issues at all. Please have a look.

> **A29:** This issue should be fixed now (Mar 4th).

Q28: (from organizers) We noticed sometimes processing errors as follows:

```
ERROR: process failed for evaluation of nets probing -- **ERROR: (IMPSYT-
6692): Invalid return code while executing 'probing_nets.tcl' was returned
```

```
and script processing was stopped. Review the following error in
'probing_nets.tcl' then restart.
**ERROR: (IMPSYT-6693): Error message: probing_nets.tcl: divide by zero.
```

A28: This error is an indication that routing is incomplete or missing altogether in the DEF, especially related to the net assets. Check your DEF file for such issues when you see this particular error.

Q27: (from organizers) We noticed sometimes processing errors as follows:

```
ERROR: process failed for evaluation of cells probing -- **ERROR: (IMPVL-
209): In Verilog file 'design.def.v', check line 50760 near the text '0x0'
for the issue: 'syntax error'. Update the text accordingly.
```

A27: These are sporadic runtime issues with Innovus. We don't know what's the root cause for those, but we noticed that just re-running the same design already helps. Thus, please re-upload whenever you see this particular issue about '0x0' syntax errors.

Q26: (from organizers) We noticed sometimes processing errors as follows:

```
ERROR: process failed for evaluation of nets probing -- **ERROR: (IMPSYT-
6692): Invalid return code while executing 'probing_nets.tcl' was returned
and script processing was stopped. Review the following error in
'probing_nets.tcl' then restart.
**ERROR: (IMPSYT-6693): Error message: probing_nets.tcl: can't read "xnext":
no such variableBad return code in "dbForEachNetWire".
```

A26: This issue should be fixed by now (Feb 25th).

Q25: (from organizers) We noticed sometimes processing errors as follows:

```
ERROR: process failed for evaluation of exploitable regions -- **ERROR:
(IMPSE-110): File 'exploit_regions_metal1--metal6.tcl' line 437: divide by
zero.
**ERROR: (IMPSE-110): File 'exploit_eval_metal1--metal6.tcl' line 163: divide
by zero.
```

A25: This issue should be fixed by now (Mar 4th).

Q24: (from organizers) We noticed sometimes processing errors as follows:

```
ERROR: process failed for evaluation of exploitable regions -- **ERROR:
(IMPSE-110): File 'exploit_regions.tcl' line 47: can't read "dist": no such
variable.
```

A24: This issue should be fixed by now (Mar 4th).

Q23: When we try to report the Timing, we notice some very large difference compared with the reference timing.rpt. What's the main reason about that?

A23: This is most likely due to the recent change in the backend which considers post-route timing and clock propagation now.
Consider the following commands before reporting timing, to mimic the same setup we have in the backend. There might still be mismatches due to different versions, but probably only very small ones.

```
####
# settings
####
set_multi_cpu_usage -local_cpu 2
set_db design_process_node 45

set mmmc_path mmmc.tcl
set lef_path NangateOpenCellLibrary.lef
set def_path design.def
set netlist_path design.v

####
# init
####
read_mmmc $mmmc_path
read_physical -lefs $lef_path
read_netlist $netlist_path
read_def $def_path

init_design

####
# clock propagation
####
set_interactive_constraint_modes [all_constraint_modes -active]
reset_propagated_clock [all_clocks]
update_io_latency -adjust_source_latency -verbose
set_propagated_clock [all_clocks]

####
# timing
####
set_db timing_analysis_type ocv
set_db timing_analysis_cppr both
time_design -post_route

####
# basic eval
####
set fl [open area.rpt w]
puts $fl [get_db current_design .bbox.area]
close $fl
report_power > power.rpt
report_timing_summary -checks setup > timing.rpt
```

```
## NOTE no exit here, as this is supposed to be sourced along with other
scripts
```

Q22: We get the dangling wire violations in metal 1 (AES_1), the following steps are what we do:
1. run innovus and import design
2. read_def desing_oringinal.def
3. check_connectivity -nets {VDD VSS} -type special (no violations)
4. write_def -floorplan -no_std_cells design_original.fp.def
5. create new innovus and import design
6. read_def design_original.fp.def
7. check_connectivity -nets {VDD VSS} -type special (get violations)

Is this a bug, or do we do something wrong?

> **A22:** Not sure exactly how that happened, but we think these dangling M1 rail reports should be fine. Innovus may show these rails as dangling even if they are connected to straps and not to rings around the core.

Q21: We also noticed some results took more than 1 hour to process. Is it normal? And we would like to know if it is possible to speed up the evaluation process.

> **A21:** Since the evaluation is entirely based on tcl scripting, there is some runtime cost. The different evaluation stages have trade-offs for accuracy and runtime, and we've already opted for relatively short runtime with acceptable accuracy. We also already have parallel processing. For the baseline layouts, we've seen runtimes of max ~50 minutes. For any submission, runtimes will vary, depending on the placement and routing and timing paths in general and that for assets in particular, and also on the workload of the backend in general.
> We understand that it's not ideal to wait for hours for some result in worst case, but there is little room for improvement at this point.
> **Update Feb 22th:** we managed to speed up parts of the processing. Still, the runtime largely depends on the level of resilience of the submitted layout (i.e., the effort required for evaluation for resilience), as well as on the workload of the backend in general.
> **Update Mar 8th:** we have managed to further speed up parts of the processing. But, at the same time, we've seen that the accuracy for probing evaluation is not good. Thus, we currently looking into this issue again, and you may experience even longer runtimes now for evaluation. We'll update everyone once that's settled.

Q20: we have some questions about the sample benchmark provided. There is a command at line 100 in file “post_impl_aes_70_shield.final/inn.cmd.gz/inn.cmd”:

```
read_metric -id current
/home/jg6476/flow_dev/out/aes_out/_genus_xfer.metrics.json
```

We hope to know what is the purpose of this “_genus_xfer.metrics.json” file and whether this file can be provided to us.
we have another question regarding “inn.cmd” file, many commands in that file end with “…”, and we hope to know the meaning of “…”.

> **A20**: These are largely auto-generated commands from some version of our example flow. The _genus_xfer file is for legacy purposes, I think, and describes some configuration of default metrics. The “…” are placeholder for longer lists. In general, the sample benchmarks are provided as is. They are not really meant to be reimplemented, but rather for providing some more or less trivial examples, which can be fully understood from the netlist and DEF itself.

Q19: Comment from organizers: We’ve heard from multiple teams now that processing may already kick in even though one file is not fully uploaded yet.

> **A19**: FYI that is because our backend does continuous check and pull of all your submission folders. Making the interval longer would limit those hiccups, but also delay everyone’s processing so that’s not a good option.
> The best way forward is to submit only zip files of your DEF, netlist submissions — as “monolithic wrapper” these zip files would only be processed once fully uploaded from your end.
> Also note the following:
> – DEF and netlist files that go together *must* use the same basename, e.g., trial1.def and trial1.v. You may still upload multiple trials at once; they will be handled in separate runs. This also applies to the new feature of subfolder submission.
> – For the new feature of zip submission, you *cannot* put multiple trials into one archive; each trial must go into a separate zip.

Q18: What do the blue lines mean in the net_assets_wo_routing.gif? What’s the exact mean about the net assert highlighted w/o layer in the zip file? And what’s the mean about notion “x” and notion “o” in the gif file.

> **A18**: I think you mean cells_assets.gif? There, the blue lines are flylines representing the connectivity of the highlighted cell assets. For the net_assets_wo_routing.gif files, there should only be white routing patterns, with ‘O’ representing sink pins and ‘X’ source/driver pins.
> In general, these gif files are more for illustration purposes and only capture the high-level view. Net assets w/o routing layers would make it easier to un-

derstand the routing paths of those net assets; w/ routing layers, one cannot really see much difference between the assets' routing and other nets' routing.

Q17: About the exploit_regions.rpt file, can i assume the total number of regions and sites count in certain region may change with different result (since exploitable regions are only defined and evaluated within a exploitable distance of cells related to security assets)?

A17: Correct, these metrics will change depending on the placement and routing and timing paths of the layout.

Q16: What do the "angle: 0" and "side: N" mean in the cells_ea.rpt file and how do you calculate the Area_Exposed? Similar, how do you calculate TotalAreaNet and PercentageExposedAreaNet in the nets_ea.rpt file?

A16: Angle and side relate to angled attack vectors, which we don't consider for this contest. Here, we only consider a perpendicular attack vector (0 degrees angle) and only top-view. Area_Exposed is (Perc_Exposed * area of macro); Perc_Exposed is calculated by polygon-level operations, where we 1) consider all wire polygons which overlap the cell asset (from 0 degrees, top-view only) and 2) derive which area of the cell is still exposed. nets_ea.rpt is calculated similarly, using polygon-level operations considering all wires which are above the net of interest's wiring. (Above the net as we are evaluating probing, fault injection from the frontside, where the attack comes from the top metal down the metal stack.)

Q15: Related to Q12. In the corresponding DEF file, there is a NONDEFAULTRULE saying "LAYER metal7 WIDTH 480". Is it the reason that there are 45 Minimum Width violations in the design? What is the purpose of those NONDEFAULTRULES?

A15: These NONDEFAULTRULEs (NDRs) are indeed related to the DRC violations. But, you'll notice that not all nets with such NDR are leading to a violation. We're using NDRs here for some exemplary protection against probing. When you compare the exposed areas of cell and net assets of this AES_2 layout to AES_1, which is of the same target utilization, you should see the differences there.

Q14: When the report is generated on the google drive, will there be some kinds of time stamps in the folder name so that we can better distinguish different submissions?

A14: Yes, the numbers in the uploaded "results_*" folders are actually the UNIX timestamps. We have also recently added MD5 checksums for the report as

well as the initial email once runs are started. The initial email also contains the UNIX timespace in the subject, so all that together should help you to pin-point which report folder belongs to which submission file.Going forward, we could also just re-include the submission files in the report folder as well, to make it easier.

Q13: When we try to report the power, we can also notice some very minor difference compared with the reference power.rpt. Is it because of the floating point error or the different Innovus version (we are using Innovus 18.12-s106_1)?

A13: Yes, that's most likely due to floating point inaccuracies and different tool versions. You should probably even observe slightly different numbers whenever you run the very same design at your end multiple times through the evaluation at your end. In any case, such minor differences wouldn't really impact the score.

Q12: Is it normal that when we load the design and call the command "report_timing_summary -checks setup", there are many error/warning messages like

a) `**ERROR: (IMPEXT-2826): Net write_data[10] has wire width (480) with NONDEFAULT RULE smaller than using WIDTH defined in technology LEF (800). Update the NONDEFAULT RULE section of the technology LEF & read it back in`

b) `**WARN: (IMPEXT-2882): Unable to find the resistance for via 'via2_5' in Cap table or LEF or OA files. The default value of 4.0 ohms is being assigned. To avoid this, check the Cap table and LEF and OA files, provide the resistance and read the files again`

Despite those errors and warnings, we can get a timing result that is very close to that in the reference timing.rpt.

A12 a): These warnings should only appear for one of the AES benchmarks. This is introduce some DRC violations for some baseline layout, just for the sake of having DRC violations. So, an additional task to obtain good score for this benchmark would be to fix these violations by re-routing the particular nets (and/or nets in general).

A12 b): These warnings should appear for all benchmarks, and they are due to the simplified nature of the Nangate library. So, you can ignore those warnings.

Q11: For assessment, different tool may has different result about PPA according to different assessment method. So I want to know the environment.

A11: We're using Cadence Innovus v17 and v18. Correct, different tools may provide different PPA results. But, since we're running both baseline evalua-

tion and submission evaluation on the same environment, results will be comparable.

Q10: There is a question about Exploitable Region. If the timing of a pair of Flip Flop(FF) is too tight to insert any route segment or gate, but there are lots of placement sites and routing tracks around them, is that regard as *exploitable region*? For example, The timing slack of FF A and FF B is 0 after routing, So there is no timing budget for NAND gate and additional routing segment timing. Am I right?

A10: Correct, in case timing slack is 0, even if there are placement sites and routing tracks available nearby, these are not considered as exploitable region then.

Q9: do we need to upload the solution files for all the benchmarks for the solutions to be evaluated? Can we just submit the solutions for part of the benchmarks and get them evaluated?

A9: You can submit solutions for any benchmark at any time; you don't need to upload solutions for all benchmarks at once. You need to upload the DEF and the post-layout netlist.

Q8: Following question 7, may ask the meaning of "Metal 1 – 50/230.0"? Does it mean that 50/230 of the routing resources in the open region are used or are free?

A8: 50/230 would mean 50 out of 230 tracks are used. To avoid such confusion, we'll change this to open tracks and also add the label in the report files.

Q7: About the trigger_space_aes_90.rpt, may I ask what is the meaning of "Open Region" and "Number of sites" in line 145? If we sum up the number of sites of all open regions, why is the result much larger than the "Unique fully vulnerable sites in 10u radius from key registers" in the vulnerable_sites_aes_90.rpt file? Does it mean that those open regions can overlap with each other?

A7: "Open Region" is the same as exploitable region. We'll streamline wording with the upcoming revision of report files as well. Yes, open/exploitable regions can overlap with each other.

Q6: It is mentioned that free track will be calculated. May I ask the definition of free tracks of a metal layer? Is it the summation of available length on each of the tracks in a 2D region? Besides, is there a Innovus command to report this?

A6: The tracks are reported per region, which covers multiple sites. So, yes, it is summed across the region.

Q5: Will an asset cell plus a horizontal/vertical exploitable distance form a 2D rectangular region where we will identify exploitable regions?

> **A5:** In principle yes, but we aim for finer granularity by looking at the actual sites, not only the 2D rectangular region. Please find the update reports for actual site coordinates, along with plots.

Q4: I want to confirm that if every submission is record as a part of final score? If it is, the initial submission may has a low score, but the final score can be improved by more times submit. For example, first score is 2, second core is 10. The average score is 6. But as submit times increase, second is 10, 3nd is 10. Then the final average score is (2+10+10)/3 = 7.3 > 6. So is it not reasonable for record every submission to the final score?

> **A4:** No, we won't record the initial or any other intermediate submissions for final scores. We only record submissions internally, and also publish the current scores during the alpha round, so that every team knows how they are currently ranking against others. Also recall that the requirement to advance to the final round is merely to submit any valid solution (good or bad score doesn't matter) until the alpha round deadline. Then, for the final round, we will publicly release few more benchmarks but also keep some benchmarks undisclosed until the end. The final scoring will be based only on the final evaluation on the final benchmarks at the end of the contest.

Q3: this contest is more focus on physical design, i.e. placement and route. But for some team, it is time consuming and may has a deviation with organizer's real intention. So is it possible to release the script or code to determine the exploitable region?

> **A3:** All evaluation scripts are implemented in Cadence Innovus. Since we don't expect teams to use specifically only Innovus, we are not releasing the scripts. (Besides, as of now, the scripts are part of an ongoing research project where code releases are not approved yet.) However, we also don't want to impose any additional burden as in suggesting to re-implement the evaluation.
> The best strategy is to submit solutions regularly and see the evaluation results. Note that you'll also get feedback in particular on the regions through the report files returned along with the final score.

Q2: It is mentioned that there will be an idea called exploitable distance which is calculated by delay. Will this distance be provided as a constant for each benchmark? If not, should we get this by some Innovus command?

A2: The exploitable distances are not a single constant per benchmark; they are rather derived for each cell asset individually, through the related timing paths. If you haven't seen the updated scoring yet, please check to see if it makes sense? Also, we may provide some pseudo-code later on, but we don't plan to release the actual scripts for the contest.

Q1: As is described on the website, exploitable region is to insert a NAND to the path. In the final score for hardware trojan insert, placement resource is sites number in the exploitable region. So if the site don't used but the routing resource is used, (i.e. metal 1 has no place to connect the pin of NAND gate), Is that considered as unused placement resource, or can consider as insert a trojan?

A1: That's a good observation. For exploitable regions, note that we consider both placement sites (sites_*) and routing resources (fts_*) in the metrics.
You're right that metal1 is particular relevant for connecting the pins of some Trojan cells. However, free tracks in other layers would also be relevant, in order to tap into/connect with sensitive nets of interest or to realize intra-Trojan routing, both across the exploitable region. Thus, we have simplified fts_* to account for free tracks across the sum of all metal layers, not for layers individually, as we cannot say in advance which layers are most relevant for each exploitable region.
Besides, note that Trojans are not necessarily only a single NAND gate; that NAND gate is rather hypothetically assumed in order to determine a worst-case estimate of exploitable region.

Q&A: Scope and Approach in Detail

Q5: we seek some clarification on what is the definition of functional equivalence that the designs will be checked against. Do the submitted designs have to maintain cycle accuracy? For instance, if the design is pipelined with one additional barrier of flip-flops (inserted by us), would the design remain "equivalent"?

A5: This is a good question. The overall scope and context for the contest is closure stages, where it would actually be more appropriate to maintain cycle accuracy. Also, you'll notice that some benchmarks have very loose timing, so there should be sufficient margin/potential for security-design co-optimization while still maintaining cycle accuracy. So, yes, submissions have to maintain cycle accuracy.

Q4: Are we allowed to insert dangling wire segments?

A4: Yes, you may insert dangling wire segments. But, such segments could easily be removed again by an attacker; hence, like filler cells, they wouldn't contribute to security and would be considered as non-existent during scoring.
Update Feb 18th: dangling wires are captured during the initial checks for valid layouts now, so are not allowed anymore.

Q3: In the constraint part on the contest website, it says we must "maintain the sensitive components". Does it mean that we are not allowed to

1. replace a sensitive cell (e.g., AOI21_X1) with a different implementation (e.g., AOI21_X4)
2. change the placement of the sensitive cells
3. change the routing solution of sensitive nets?

A3: No, you may actually do all that of a), b), and/or c) as you wish. It's a good comment — our wording of "maintain" is too vague. We meant that you cannot simply remove or restructure the assets but you can fully revise the physical design of asset components (not the cells' definition themselves though).

Q2: Since you mentioned that we also need to upload the post-layout setlist, does it mean that we are allowed to replace the implementation of certain design component (e.g., replace a AOI21_X1 with AOI21_X4)?

A2: Yes, you may rework the implementation of any component, including assets. For assets, however, anything going beyond your example of driver strength will become complicated, given that you must a) maintain the sets of assets as such, i.e., do not remove or restructure them, and, of course, b) maintain functional equivalence for the overall design.

Q1: We noticed the netlist (.v) file had included the cts information (buffers/nets). So if we perform from placement to routing again, we may insert more buffers during the cts optimization stage.
We think you should give us the post-sim netlist files, not the post-route netlist, so that we could get the floorplan and PG network information from the original def files then perform our own placement/routing method.

A1: You're free to re-run CTS optimizations, as we don't imposed constraints on that, aside from the target frequency.
Please remember that this contest is focused on security *closure*, so mainly targeting for post-layout efforts. In particular, note that cells and nets assets are defined based on the post-layout files — these assets are not to be removed or restructured, which might well happen in case you'd restart from post-synthesis netlists.

> Thus, the provided DEF is the reference baseline, and the provided netlist is rather for informative purpose, e.g., to ease timing analysis in case you want/need this for your approach. That being said, as long as you maintain the assets as well as the overall functional correctness, you are still free to revise the physical layout and the logic of the design — so we're giving some flexibility here and do not enforce, let's say, only minor ECO tweaks.

Q&A: Threats and Approach in General

Q3: My thought is that the "right" way to solve these things is to build a placer and router from the ground up, that can factor security in directly. Writing a good placer is hard, and writing a good router is even harder, though.

> **A3:** The idea for this contest is rather to leverage any existing tools you have, could be commercial tools, OpenROAD tools, or your own tools. Without such basic tool setup, we agree that tackling the contest will require a lot of efforts for ramp-up. As for integrating defenses into your tool setup, you want to 1) fully understand the scope in general and the threats in particular, 2) fully understand the way the threats are considered and scored for this contest, and 3) be as creative as possible to use placement and routing and other design techniques for introducing defenses, but don't really "re-invent the wheel" for the basic techniques.

Q2: Related to probing. If I understand correctly, if there's "line of sight" on to the diffusion or poly of a critical gate, an attacker can probe/disrupt. How much space has to be exposed for the probe? If a probe can see between minimum-spacing wiring, then multiple metal layers have to be stacked and aligned to block off the line of sight.

> **A2:** Correct, any line of sight down to the active layer is considered vulnerable. How much space is required for an attacker to actually succeed with their probing or fault injection tools depends largely on the IC's technology node as well as the resolution of the probing, fault injection tools. To simplify for this contest, we ask to minimize the exposed area of assets in general. It is correct that spacing of wires and routing patterns will play key roles to defend against these threats.

Q1: Related to Trojans. Say we have a register cell that contains a security-critical bit. If the cell has lots of open area around it, a hacker could insert a trojan next to the cell, and then use that trojan to leak the bit in some manner(?). In terms of cell sites, is there a distance threshold we need to worry about? A number of (contiguous) nearby cell sites?

A1: There is indeed a notion of relevant distances, assuming that designs with Trojans inserted must still meet timing, hence Trojans cannot be placed arbitrarily. More details are provided in the scoring description.
