Problem B:

Power and Timing Optimization Using Multibit Flip-Flop

Sheng-Wei Yang, Jhih-Wei Hsu, Yu-Hsuan Cheng, Cindy Chin-Fang Shen
(Synopsys, Inc.)

Q&A

Q1. For the given input test cases, only TNS will be provided. Based on our understanding of DEF/LEF, it seems that the slack for each cell will not be explicitly indicated in the input data. Could you confirm whether the input will provide the initial slack for each cell at the initial placement, or is there another way to obtain the slack for each cell?

A1. We will include an SDC file with the input.

Q2. As an extension of the first point, I would like to confirm whether the slack, power, and area information of an individual flip-flop will be included in the flip-flop library, or if the library follows the same format as last year.

A2. We will include an SDC file with the input. User can utilize open source or self-implemented Timer to help parse it.

Q3. In the Evaluation section, it is mentioned that banking & debanking must follow the scan chain rules. Does this imply that the scan chains are predefined at the initial placement, specifying the sequence of cells in each scan chain? If so, when performing optimization, should we consider the order of the scan chain and ensure that banking & debanking operations adhere to its topological order?For example, if the initial placement shows that the topological order of a certain scan chain is flip-flops A → B → C, would it be valid to bank only A and C, or must all flip-flops within the chain be banked together in order?

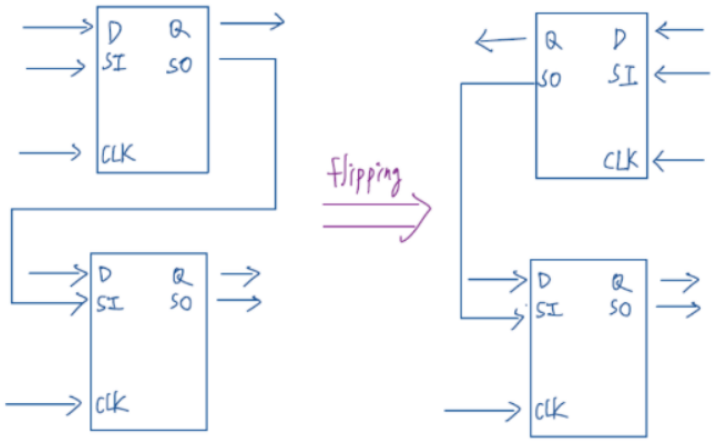
A3. If there is scan def in the def file, contestants should follow. If there is no scan def file, contestants don’t need to follow it.

Q4. Is banking across different scan chains prohibited? Additionally, will every flip-flop necessarily belong to a scan chain? Lastly, does the SI/SO mapping need to be included in the output?

A4. If there's scan def, contestants should follow its connection before and after multibit moves. If there's no scan def, we do not constrain it.

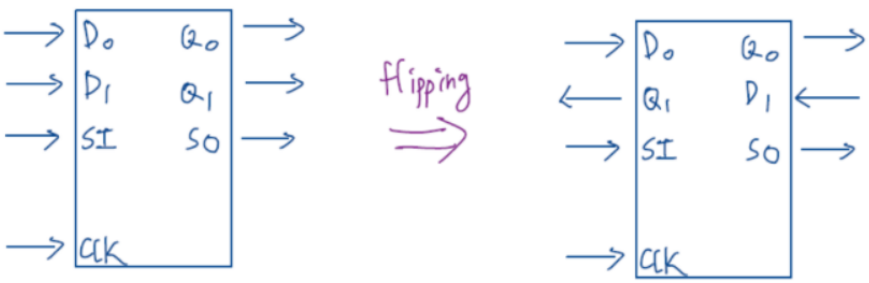
Q5a. Can the pins of the flipflop be flipped as shown in the examples below?

Case 1: flip a single bit FF (the top one)



Case 2: flip all the pins of a MBFF

Case 3: flip the partial pins of a MBFF



A5a. We allow cell flipping as long as cells could be placed legally.

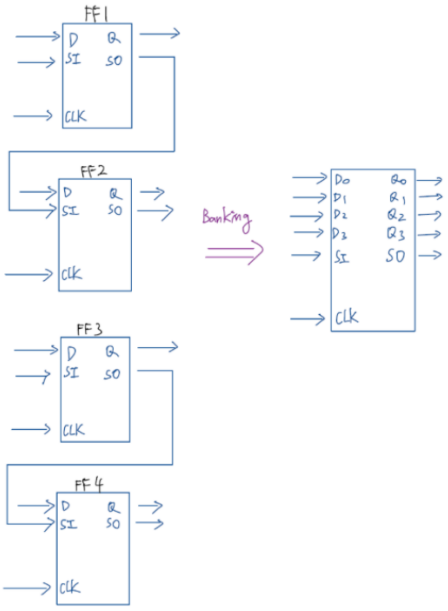
Q5b. Does it mean that the pin locations relative to a flip-flop from the input file can be modified (i.e., the output pin locations can differ from the input) to implement "flipping"?
Or is there an attribute to indicate whether the FF has been flipped in the output file?

A5b. Yes. Please refer to LEF DEF format on how to reflect a flipped cell solution.

Q6. If the scan chain order of the single-bit flip-flops is FF1 → FF2 → FF3 → FF4, does it mean that the mapping must follow the order (i.e. the D pins of FF1,FF2,FF3,FF4 must be mapped to D0,D1,D2,D3 of the MBFF respectively) after banking? Or can the D pins of FF1, FF2, FF3, and FF4 be mapped to D3, D1, D0, and D2, for example?

A6. If there is scan def in the def file, contestants should follow its precedence in their submission results. That is to say, the order precedence should remain the same.
If there is no scan def file, contestants don't need to follow it.

Q7. How should the pins of single-bit FFs be mapped to a MBFF after banking in the case shown below (where the scan chain orders are FF1 → FF2 and FF3 → FF4)? If I map the SI of FF1 to the SI of the MBFF, does that imply the SO of FF4 should be mapped to the SO of the MBFF? Do I understand correctly?



A7. If that's why defined in the scan def, yes. FF1/SI should map to the banked cell's SI and FF4/SO should be mapped to the banked SO.

Q8. According to the updated problem description, it is mentioned that an SDC file will be provided.
However, as far as I understand, the SDC file typically only contains information such as the clock period.
To compute slack values for each path, it seems that additional data would be required — such as path delays derived from cell delays, wire lengths, and setup/hold times of flip-flops.
Could you kindly clarify where or how we are expected to obtain the slack information for each path?

A8. Gate delays, loads, input delays and output delays are being provided for each clock loads. We are providing library information of each cells. Please utilize the given input information to optimize the circuit.

Q9. According to the problem description, "A placement region is divided into several bins, each with a defined utilization rate threshold." However, I could not find the actual threshold value or details on how this constraint should be applied.
Could you please clarify:
What is the specific utilization rate threshold for each bin?
Will this threshold be provided as part of the input for each testcase, or is it a fixed value?
How strictly must this threshold be enforced (e.g., soft vs. hard constraint)?

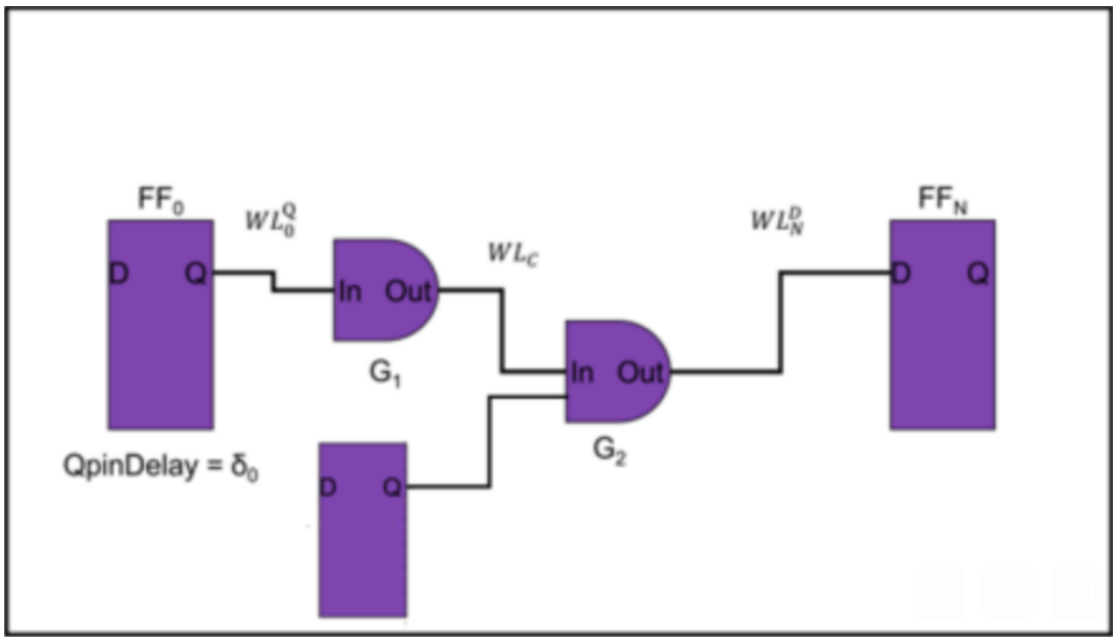
A9. Thank you very much for your inquiry. The utilization rate threshold constraint is not enacted in 2025 ICCAD CAD Contest Problem B. Please refer to the updated Problem Formulation.

Q10. The problem statement says "All instances must be placed on-site and within the die region." However, it is not explicitly stated whether the die region size or placement row/site information will be included as part of the input.
Could you please confirm:
Will the die area and legal placement sites be provided via the LEF/DEF input files?
Is there any additional information about site dimensions or die boundaries we should be aware of?

A10. Yes. The die area and legal placement sties will be provided via the input files. The legality of contestants' submission, along with the cost metrics, would be evaluated with the latest version of Synopsys Fusion Compiler.

Q11.

1. Could you give a formulated definition about the function TNS(), like the relation between TNS() and DisplacementDelay, QpinDelay, TimingSlack.
2. How to calculate the WLq if the Q pin connects to more than one pin?
3. How do we determine the slack when a FF has multiple ancestor FFs? As depicted in the figure.How to know which one is critical path?



4. How will the contest provide the slack-related information? Will it list the slack value of each individual pin, or only provide the overall TNS of the original circuit?

A11. Please refer to the latest Problem Formulation of 2025 ICCAD CAD Contest Problem B.

Q12. Is the organizer considering providing a complete .lib file (such as Nangate45_typ) for contestants to test timing and power consumption using open source tools?

A12. We will provide .lib file format to contestants.

Q13. I would like to clarify a detail regarding the DIEAREA specification in the .def file. According to the LEF/DEF Language Reference Manual (Reference [5]), particularly at the bottom of page 251, an example is provided that illustrates an L-shaped polygon for DIEAREA. However, I would like to confirm whether, in this contest, the DIEAREA is always guaranteed to be a rectangle.
Additionally, if the DIEAREA is always a rectangle, is it guaranteed that the coordinates are always specified using exactly four points, as shown in testcase1?

A13. The DIEAREA would be a shape descriptive by LEF DEF format.

Q14. I would like to ask if there's a file with the information of all kinds of gates and flipflops?

A14. LEF DEF file as well as .lib file will be provided along with updated testcase.

Q15. I just downloaded the testcase. When I start to check the input files, I cannot locate where the Flip-Flop Library Cells are. There is a weight file and a .swp file that correspond to it. There are also a .sdc file and a .def file, which contain the placement results. However, those files have no information regarding the flip-flop library cell. There are also no instructions regarding how the .sdc file name will be sent to the program in the problem instruction on the ICCAD contest webpage, which was updated on May 8.
On the other hand, there are other files in the testcase1.tar too. A Verilog file represents the circuit's topology, and a .tf file contains the technology information. But again, there is no information regarding the cell library.
I would like to know if the provided testcase1 is intended to be like this, or if something is wrong.

A15. LEF DEF file as well as .lib file will be provided along with updated testcase.

Q16. For problem B, lots of folders and files are added in the updated testcase. I would like to inquire whether there's a description file explaining the new-added folder and file in the testcase, inlcuding SNPSHOPT25, SNPSLOPT25, SNPSROPT25, SNPSSLOPT25, and all folders within them.

A16. These are for the .lef and .lib files describing the library of testcase1.

Q17. By the information on spec The contestant's submitted program should be able to execute as follows ./cadb_0000_alpha <inputFile1.txt> <inputFile2.txt> <inputFile3.txt>… <outputName>
I would like to ask the input file include all the files in testcase1 file(.sdc、.v、.tf、.def、testcase1_weight)?

A17. Yes.

Q18. I would like to ask the spec says we have to solve the location of MBFF as a result and we have to submit the def. However, the placement of wire and net information are in the def, should we handle this by ourselves?

A18. Yes.

Q19. I am currently working on Problem B and would like to perform static timing analysis (STA). I would like to use Cadence Tempus for this purpose.
I have tried sourcing the license files with the following commands:

source /project/cad/cad50/cadence/CIC/license.csh
source /project/cad/cad50/synopsys/CIC/license.csh

However, it seems that the tempus (or pt_shell) command is not available in my current environment.

May I kindly ask if Tempus is available on this system, or if there is any recommended way for performing STA under the current setup?

A19. Cadence Tempus would not be used as evaluation metrics for this contest.

Q20. We have also noted that the current file set does not include essential implementation data such as cell width/height, power, and timing information. Should we expect additional files (e.g., .lef, .lib or .ndm, .upf, etc.) to be provided?

A20. .lef and .lib has been sent with the latest release.

Q21. I'd like to confirm the meaning of the program requirements description "./cadb_0000_alpha <inputFile1.txt> <inputFile2.txt> <inputFile3.txt>... <outputName>
inputFile1.txt represents the input syntax data from sections 3.1.1 to 3.1.5.
inputFile2.txt and all subsequent files, up to but not including the last parameter, represent design data in LEF/DEF format."
For the testcase1, I would like to confirm whether inputFile1 refers to testcase1_weight.
Additionally, do inputFile2 through the last input file (excluding the outputName) include not only LEF/DEF files but also other design files such as testcase1.sdc, testcase1.v, testcase1.tf, and *.lib?

A21. The format would be like this:
cadb_0000_alpha testcase1_weight testcase1_lib1.lef testcase1_lib2.lef testcase1_lib1.lib testcase1_lib2.lib testcase1.tf testcase1.v testcase1.def testcase1.sdc outputName
Please note that the number of .lib and .lef may be design-dependent and therefore tangible.

Q22. Can the .lib and .db files under liberty provide relevant format descriptions? Reference only has format descriptions for .sdc and lefdef.

A22. We would encourage contestants to refer to Liberty File Format: What is Library Characterization? – How it Works & Techniques | Synopsys (https://www.synopsys.com/glossary/what-is-library-characterization.html)

Q23. I have tried to use Library Compiler to convert all .lib files in the liberty folder into .db files, and tried to use ICC2 Library Manger to convert all .db files into .ndm files so that ICC2 can read them, but it seems that such conversion on ICC2 Library Manager seems to have errors. Therefore, I would like to ask the organizer if there are some problems with this test case, or is it simply that ICC2 does not support this operation?

A23. We encourage contestants to continue research with our testcase. However, we could not guarantee that every file format that contestants converted this design into would result in success.

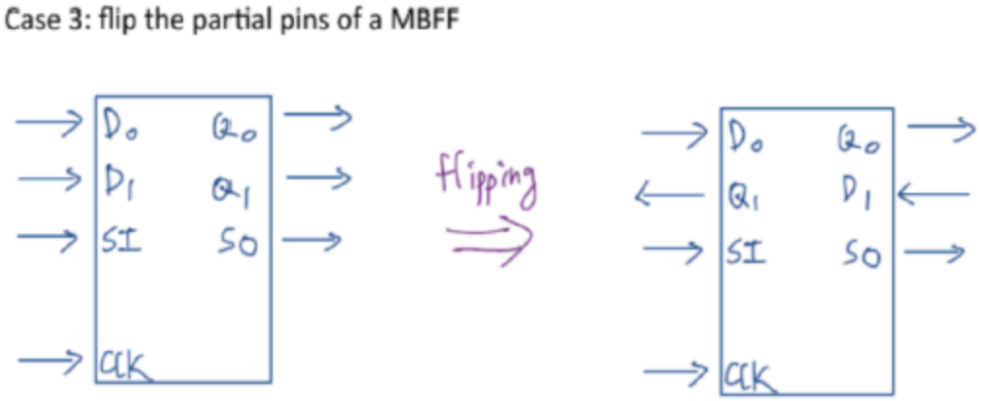
Q24. Is the placement result in the input data guaranteed to be legal? In addition, we noticed that in the currently released testcase1, some non-flip-flop cells are not on-site or overlapping. Is this expected behavior, or could we be misunderstanding the provided data?

A24. We expect contestants could optimize the design regardless of initial legality.

Q25. Will there be any tools provided to check whether our generated output is legal (e.g., specific Fusion Compiler commands or other checkers)?

A25. We are evaluating the content of evaluation tool to be released public. Meanwhile, check_legality from Fusion Compiler could be applied for legality check for now.

Q26. The A5a of the QA response mentioned the possible flipping shown in the attached image. But this doesn't seem possible because DEF LEF format only supports mirroring and rotation. Could you clarify on how what possible cases are possibly present in the testcase?



A26. Please follow LEF/DEF file format where cell flippings are allowed.

Q27. After reviewing the most recent Q&A, I noticed that the response to Q11 lacks sufficient detail. Specifically, A11 refers us to the latest Problem Formulation to address the four sub-questions, but I was unable to locate any relevant information therein. I would like to confirm whether the timing model used for this year's

competition is the same as that of ICCAD 2024 Problem B. If that is the case, I kindly ask that this be explicitly stated in either the Q&A or the Problem Formulation. Otherwise, I respectfully request a thorough explanation of the timing model adopted this year.

A27. 2025 ICCAD CAD Contest Problem B is an extension from 2024 ICCAD CAD Contest Problem B. The similarity is that both aim for power/performance/area optimization. The differences is that instead of using simplified and artificial data format, in 2025 we have adopted standard industrial format for contestants to participate in industrial-scale optimization problem.

Q28. I find it unusual that only the .sdc file has been provided, as it contains constraints primarily used for synthesis. It would be more appropriate to directly provide the initial timing slack of each flip-flop (FF). Based on the current setup, it appears that we are expected to perform timing analysis using EDA tools to obtain this information. However, such pre-processing seems unrelated to the core objectives of this EDA problem, and it would be unreasonable to require participants to run a synthesis tool like Design Compiler simply to retrieve initial slack values. I therefore request that you either clearly explain how to obtain the initial slack using the provided resources or supply the necessary data files directly.

A28. We do not require contestants to run Design Compiler. The SDC file is to help contestants understand the timing information of the current design.

Q29. I would like to raise concerns about the contents of testcase1. It appears that various files, possibly outputs from Design Compiler, have been included without explanation, particularly the .lib files. These files are highly repetitive and collectively occupy approximately 29 GB of storage. It is unclear whether we are expected to parse all of them, or if only a subset is relevant. Clear guidance on how to utilize these files effectively would be greatly appreciated.

A29. As explained in the Problem Formulation, we would like to provide a full industrial format design for contestants to participate in industrial-level optimization problem.

If you are not aware of what liberty file format is, we recommend you to refer to some materials on what liberty file format works: What is Library Characterization? – How it Works & Techniques | Synopsys (https://www.synopsys.com/glossary/what-is-library-characterization.html)

Q30. As stated in problem description 5.1, the score function requires the contestant to optimize both TNS and total Power. Reviewing the provided testcase, we found that related information is only present in either *.lib or *.db files. Although it might be possible for the contestants to attempt a calculation on all the simulation parameters and derive the required information to calculate the score function, such task is clearly outside the bound of the contest.

A30. Please refer to Liberty File Format and LEF/DEF Language Reference. There are several online references that you can review to understand the physical timing analysis of these library settings. Below are some examples that you could refer to:
https://people.eecs.berkeley.edu/~alanmi/publications/other/liberty07_03.pdf
https://www.ispd.cc/contests/18/lefdefref.pdf

For this Problem B contest, applying physical design tool on the testcase is not a requirement to participate in this contest. We have provided the testcase using industrial standard so that contestants could parse and optimize using standard file format. We would encourage contestants to study the Liberty File Format and LEF/DEF Language Reference and utilize the well-formulated testcase information for optimization.

Q31. We would like to clarify whether it is allowed to invoke Synopsys tool commands (e.g., ICC2, Design Compiler) within the final submitted program — for example, by calling icc_shell -f script.tcl during execution.
To ensure we fully comply with the contest rules, could you please confirm if such usage is permitted in the final submission?

A31. We allow contestants to use open source parser or open source software to help contestants parse and analyze the testcase. However, please note that contestants are responsible for the copyright legality of the submissions. A submission including third party commerical EDA tools, including but not limited to Synopsys technology, is subject to the copyright claim from the included third party tool. We would advise

contestants to implement based on open source code and refrain from copyright infringement.

Q32a. According to QA21, all the files in testcase one are needed, including .tf, .v, and .lib files. However, the purpose of those files and their explicit formats are not mentioned in the problem introduction, which was last updated on 5/29. I'd like to know whether we should figure out how to use the files or if an updated version of the instructions is coming.

b. Is there a way to tell whether a cell is a flip-flop, or should we judge it by checking the information of the cell name, pins, etc?

A32a. The release file for .tf, .v, and .lib are techfile format, Verilog file format, and Liberty library format. The purpose of these files is to represent design status.

b. You could tell if a cell is a flip-flop by reading its .lib file.

Q33. Because there are many files in the testcase 1, I would like to know that how should we test testcase 1. Or more specifically, which libraries (the lib or the db files) and files (lef, def, etc.) should be used to test testcase 1. I think that it is unlikely to just pass all the .lib files to our program.

It would be even better if you can give the terminal script that should be used to test testcase (something like ./your_exexutable weight_file lib1_file.lib lib2file.lib ... Lib.lef place.def out.def)

A33. Please refer to Liberty File Format. If you are not familiar with how Liberty File Format works, there are some online resources that you can utilize. For instance: Synopsys' Open-Source Liberty Format to Incorporate On-Chip Variation Extensions - Sep 5, 2012 (https://news.synopsys.com/home?item=123415)

If there's any update to the finalization of the evaluation script we will update to contestants.

Q34a. In testcase1.v, all SI and SE pins of the flip-flops are unconnected.
Does this imply that scan chains are not required to be considered for this testcase?

b. In testcase1, there are four types of macros: SNPSHOPT25, SNPSLOPT25, SNPSROPT25, and SNPSSLOPT25.
Is it valid to use different types of single-bit flip-flops (FFs) to construct a multi-bit flip-flop (MBFF)?
Or does the single_bit_degenerate attribute in the .lib file means the exact single-bit FFs that are allowed to be grouped into a specific MBFF?
For example, is SNPSHOPT25_FSDN4_V2_2 restricted to being formed only from SNPSHOPT25_FSDN_V2_1?

```
cell(SNPSHOPT25_FSDN4_V2_2) {
  area : 5.2392 ;
  cell_leakage_power : 1.433e+04 ;
  ecsm_vtp : 0.3374 ;
  ecsm_vtn : 0.3193 ;
  single_bit_degenerate : "SNPSHOPT25_FSDN_V2_1" ;
```

A34a. Yes. The scan connection should be logically equivalent. If the testcase has no scan connection, the submission is expected to have them stay logically equivalent.

b. single_bit_degenerate describes the single bit FF that are allowed to be grouped into a specific MBFF.

Q35. I would like to apply for the design compiler because I want to integrate the design compiler into the workflow of my solution towards problem B. Therefore, I will need to make sure that the software can be installed and executed under the testing environment. And judging from my experience on the virtual machine you provided, it is almost impossible to install anything on the platform.

I would like to know that if this solution flow is allowed (i.e. cooperate with existing design compiler to format the lib files into a simpler form), or we have to write or find a parser on our own and integrate it directly into our solution.

A35. We allow contestants to use open source parser or open source software to help contestants parse and analyze the testcase. However, please note that contestants are responsible for the copyright legality of the submissions. A submission including third party commercial EDA tools, including but not limited to Synopsys technology, is subject to the copyright claim from the included third party tool. We would advise

contestants to implement based on open source code and refrain from copyright infringement.

Q36. We would like to ask for clarification regarding several issues we have encountered while working on testcase1:

We observed that after relocating the flip-flop and gate cells in the .def fille, the timing information reported by report_qor -summary, report_timing, and other timing-related commands does not change. Regardless of how we modify the cell placements, the timing metrics such as TNS remain unchanged. This behavior seems inconsistent with what we would expect in a physical design flow, and we would like to confirm whether there may be any misconfiguration on our part.

Based on the response to Q24, we understand that the input data may be illegal and may contain overlaps even between gate cells. However, in the output format section of the problem description, it states "a flip-flop placement solution with its Verilog of netlist connections." We would like to clarify: are we allowed to move gate cells as well, or only flip-flops?

A36. It could be misconfiguration or incompatibility between your timing information file reading and design description.

We expect contestants to move only flip-flops. Contestants does not need to move or consider the legality status of combinational cells.

Q37. There are so many .lib files in testcase1. Which of them should we use to test testcase 1, or should we use all of them (there are like 50+ .lib files) when testing? It would be better if could provide a clearer file dependency regarding the testing of testcase1.

A37. We expect contestants' submitted program could work like this:

./cadb_0000_alpha

-weight <weightFile>

-lef <lefFile1> <lefFile2> ...

-db <dbFile1> <dbFile2> ...

-tf <tfFile1> <tfFile2> ...

-sdc <sdcFile1> <sdcFile2> ...
-v <verilogFile1> <verilogFile2> ..
-def <defFile1> <defFile2> ...
-out <outputName>

However, for Alpha testing, submissions without following this format are still eligible for testing as long as a clear description of README file is included with the submission to describe how to run and evaluate.

For file dependency, we will input testcase using the aforementioned format. We expect contestants to have the ability to handle the input file, but whether the input file should be applied for the optimization is up to contestants' decision.

In another words, we expect the submission to be able to handle the input with the given format, but we do not require contestants to use every information and files from the input.

For more details regarding .lib files and their file dependency, the input file format is following Liberty File Format. Please refer to related documentations on how what Liberty File Format represents.

Every .db file under the subfolder of testcase1 could be dbfile*. For example: snps25lopt_base_ff0p715v125c.db or snps25lopt_base_ff0p715v25c

Q38. We are writing to request your guidance on a persistent technical issue we've encountered while working on the testcase1 design case. After extensive and systematic debugging, we have observed some unusual phenomena that we are unable to resolve and would like to ask for your clarification.

Our Observations:
* File Content and Naming Confirmed: We have manually verified that the cell names in the Verilog netlist (.v)—for example, SNPSSLOPT25_OR2_MM_2—perfectly match the corresponding cell definitions within the provided .lib library files. This confirms our basic data mapping is correct.
* Tool Linking Failure: However, regardless of the Tcl script flow used (including robust methods like creating separate tech/design libraries), the link_design command in Fusion Compiler consistently fails to find these confirmed-to-exist cells, reporting

unresolved references.
* An Accompanying Anomaly: As a control experiment, we discovered a peculiar behavior in the provided Linux environment. Standard command-line utilities that need to parse files (like less and grep) fail to read the full content of the .lib files. However, simple-stream commands like cat can display the files correctly. This leads us to suspect that the library files may have a special format or encoding that requires a specific handling method.

Our Questions:
Based on these observations, we suspect our understanding of the provided kit's characteristics may be incomplete. We would be grateful if you could clarify the following:
* Do these educational library (.lib) files have a known special format, or do they require a specific tool version/setting to be parsed correctly? (We are currently using Fusion Compiler version V-2023.12-SP4).
* Is there a pre-processing or conversion script that we might be unaware of, which needs to be run on the netlist (.v) or libraries (.lib) before executing the standard Place & Route flow?
* If possible, would you be able to provide a minimal, known-good Tcl script example that successfully completes the link_design step? This would serve as an invaluable baseline for our comparison.

A38. Regarding the .lib file format and tool compatibility
The educational library (.lib) files provided in the contest kit are standard Liberty format. As far as we understand, we don’t know if the library (.lib) files have any special format.
However, in some cases, they may be compressed, encoded, or have Unix line endings or character encodings that differ from typical ASCII text. This can cause issues with some command-line tools (like less or grep) if the files contain non-printable or special characters.
Fusion Compiler V-2023.12-SP4 is compatible with standard Liberty files. However, if you observe that only cat displays the file correctly while less/grep do not, it is possible the files are: Compressed (e.g., gzip, bzip2, or similar), Encoded (e.g., UTF-16 or with a BOM), or Contain long lines (which can break some tools)
As for the pre-processing or conversion requirements, there is no special pre-processing or conversion script required for the provided .v or .lib files in the standard contest flow.
Concerning your request for minimal known-good tcl script for link_design, we could

do some further analysis to evaluate if providing a baseline for all contestants is feasible.

Q39. Is there any timing information ommited in the testcase?

A39. The timing information has been given for the timing analysis of this contest. Please clarify on what you meant by omitted timing information.

Q40. Based on Q34b, it seems that mbffs can only debank into its single_bit_degenerate. Is it allowed to debank into the same sbff under different optimization? For example, debank SNPSHOPT25_FSDN2_V2_1 into SNPSLOPT25_FSDN_V2_1?

A40. We allow debanking from higher-bit multibits into lower-bit multibits.

Q41. Is it allowed to change a sbff into another sbff with better performance?

A41. Yes. As long as they are compatible and functionally equivalent.

Q42. In Q34b, it says single_bit_degenerate restricts the banking/debanking between sbff and mbff. However, there's some mbffs without single_bit_degenerate such as SNPSHOOPT25_SSRRDPQ4_2 Iis there any banking restriction for these mbffs?

A42. single_bit_degenerate describes the single bit FF that are allowed to be grouped into a specific MBFF. Those with no single_bit_degenerate means they don't have exact equivalent compatible single bits.

Q43. We have noticed that the libraries provided in the testcase include a wide range of PVT corners. Could you kindly clarify which specific corners we are allowed to use? Additionally, for timing and power analysis, which PVT corner should we consider?

A43. You are allowed to use any corner as long as the registers are compatible and placed legally within compatible voltage area.

Q44. It appears that the testcase does not include any files related to wire RC information. As a result, when we move flip-flops and re-run timing analysis using Fusion Compiler, we observe no timing changes. Could you please confirm whether any additional files are needed?

A44. We do not think any further files are needed for now.

Q45. Regarding the banking/debanking constraints of degenerate flip-flops, would it be possible to provide more detailed documentation or clarification? Specifically, we would like to understand whether a degenerate flip-flop can be replaced with a cell from another library before or after the banking/debanking process, and whether cross-library transformations are supported under these operations.

A45. You are allowed to use any corner as long as the registers are compatible and placed legally within compatible voltage area.

Q46. We would like to analyze the detailed characteristics of each component using Synopsys Fusion Compiler. According to the User Guide, it appears that we need to use the create_lib command, which requires an .ndm file. Will .ndm files be provided for the testcases, or is there an alternative recommended method for obtaining accurate component data?

A46. This contest is meant for contestants to design their own multibit optimization technology to deliver a best PPA solution for the given testcase. Fusion Compiler is the tool that is being used for metrics evaluation like TNS, power, and area.

Q47. In Section 3.1.5 of the specification, does this imply that new component types may be introduced via the weight file? Or are we expected to infer component types solely based on information from standard input files such as DEF?

A47. We are not planning for any new component type for now.

Q48. In Section 5.2, it is stated that the following files will be read: weight, lef, db, tf, sdc, v, and def. Will .lib files also be used during evaluation? In Testcase 1, the db folder contains only one subdirectory named SNPSLOPT25. We would like to better understand how library files are accessed and used by the evaluation system.

A48. Thank you for your inquiry. .lib is being used during evaluation. Please refer to the latest Problem Description.

Q49. Additionally, may we ask when Testcase 2 is expected to be released?

A49. New announcements will be made after Alpha testing.

Important note from topic chairs:

Fusion Compiler, IC Compiler II, or any other commercial Physical Design tools are not a prerequisite tool to participate in this contest. This contest is meant for contestants to design their own multibit optimization technology to deliver a best PPA solution for the given testcase. Fusion Compiler is the tool that is being used for metrics evaluation like TNS, power, and area.

Q50. 1. How to know the operation mode for each testcase. I mean for example, there are many kinds of operation mode, such as ff0p88v25c, ff0p88v125c, ss0p88vm40c, tt0p65v25c, ss0p585vm40c ... Where can I find which operation mode is used in input file?

2. For Q39, I mean where can I optain the delay data such as Q_pin delay in 2024 input file?

3. For Q40 and 41, does it mean I can change a sbff to another kind of sbff and bank into mbff other than its? Is single_bit_degenerate this contradicted?

A50.

1. Please expect every operation mode to be used.
2. Please refer to the .lib file for each library cell characteristics and .sdc file for design timing information.

3. Every sizing or swapping between single bit flip-flops need to be functionally equivalent.

Q51. 1. Based on a previous response, we understand that we are only allowed to move flip-flops and do not need to consider the legality of combinational cells. We would like to confirm: is it expected that some of the provided testcases may contain illegal combinational cell placements, and that the final legality check will only consider flip-flops?

2. Based on a previous response, it was mentioned that we can use Fusion Compiler's check_legality command to validate legality. However, in the context where combinational cells are not considered, are there any other recommended ways to check legality specific to flip-flops?

3. We have also observed that Fusion Compiler's check_legality enforces additional constraints beyond the problem's definition — for example, direction constraints (FS & N) when flip-flops are placed in certain rows. Could you clarify whether we should follow these tool-specific constraints or only the legality rules defined in the contest?

4. Regarding a previous issue we raised — the timing report not reflecting changes in cell placement — we have now identified that this is likely due to missing wire RC information, such as .itf or .tluplus files. As a result, Fusion Compiler appears to be computing timing using zero interconnect delay. Would it be possible for the organizers to provide the relevant RC-related files?

5. Despite the zero interconnect delay, we still observe that the initial TNS reported by Fusion Compiler matches the value provided in the testcase1_weight file. Is this an expected behavior under the current setup?

We would also like to ask some follow-up questions regarding flip-flop variants and unconnected wires in the netlist:

6a. In testcase1, scan chains are not used. As a result, the flip-flops are connected to sequentially named unconnected wires in the .v file. Given this, are we allowed to replace the flip-flop cells with types that do not have SI and SO pins?

6b. Similarly, some flip-flop cells include a QN pin that is unused. If this pin is not connected, are we allowed to substitute the flip-flop with one that lacks the QN pin?

6c. Is there any naming rule for the unconnected wires in the .v file (e.g., must follow a certain order, or avoid reusing previously used names)?

6d. If extra unconnected wires remain after banking, should we remove their declarations from the .v file?

Lastly, we would appreciate clarification regarding the updated program requirements document:

7a. In the updated version of the 5.2 Program Requirements document released on June 25, we noticed that it lists .db rather than .lib. However, the contest-provided files are mostly in .lib format. Could you clarify whether .lib will be supported as input?

7b. The same updated document states that .tf, .sdc, .def, and .v may have multiple files. Should we expect that some testcases will contain multiple files for these formats?

We greatly appreciate your support and look forward to your clarification.

A51.

1. The final legality check will only consider flip-flops.
2. There are some attributes to distinguish whether a cell is combination or sequential. For example, is_combinational. For further information, please refer to the manual.
3. We expect no-overlapping and cells are on-site.
4. Please refer to the updated testcase.
5. It is expected behavior.

6a. Yes.
6b. Yes.
6c. No.
6d. No.
7a. .lib will be supported. Please refer to the latest spec.
7b. It is unlikely but possible.

Q52. 1. Will the evaluation be based on a single PVT corner (e.g., TT at 0.8V 25°C), or does it involve multiple corners (e.g., SS, TT, FF under various voltage and temperature conditions)?

In the directory testcase1/SNPSSLOPT25/liberty/nldm/, we found many .lib files corresponding to different PVT settings. Could you please clarify which specific corner(s) we should use for evaluation?

2. If multi-corner analysis is involved, could you kindly indicate which input file(s) specify the corner setup?

A52.

1. It will involve multiple corners.
2. It will be the worst case analysis.

Q53. 1. In the "5. Evaluation" section, it is stated that the results will be evaluated using Synopsys Fusion Compiler. However, we noticed that the 2025 CAD Contest Competition Host User Manual, distributed via email on May 31, does not include Fusion Compiler in the list of available tools. Does this mean that Fusion Compiler is indeed not accessible to participants, or was this an oversight in the manual?
2. In the updated “5.2 Program Requirements” section (as of June 24), we observed that only .db files are provided, with no .lib files. Does this imply that only Synopsys .db files will be used in future testcases? If so, when will the updated testcases be released? If not, what is the expected input file format? Additionally, we noticed that in order to use Fusion Compiler, it is necessary to convert .lib files to .db files using Synopsys Library Compiler. However, this conversion process can take more than one hour, which would directly exceed the contest's time limit for the submitted binary. Given this, does it mean that using Fusion Compiler is effectively prohibited under the current constraints?

A53.

1. All the commands used in this Contest are mutual commands between Fusion Compiler and IC Compiler II.
2. Both .db and .lib are provided. Contestants could decide if their program is optimizing based on .db input or .lib input.

Q54. 1. We are developing mostly with Python, do we have to strictly bundle the entire program into only one executable? Or we can just submit a folder containing all our Python code along with a script for running testcases?

2. It is mentioned in the QA that you will provide the naming format for submission. But so far, I cannot find any related information, neither on your website nor in the provided documents. Could you ensure that it will be released before Beta submission?

A54.
1. The evaluation process is automated and based on the Program Requirements. Failure to follow the Program Requirement would possibly result in no score.
2. The naming convention is already mentioned in the spec. cadb_0000_alpha for alpha submission, cadb_0000_beta for beta submission, and cadb_0000_final for final submission, where 0000 is the placeholder for contestants' team number.

Q55. I am currently running the provided testcase using Synopsys Fusion Compiler (fc). However, I noticed that there is no RC extraction model file (such as .tluplus or ITF), nor any layer/tech map files included in the materials.

When I try to run place_opt or timing analysis, Fusion Compiler reports:

> No valid parasitic for any corner

This seems to indicate that the RC models or tech maps are missing.

May I ask:
1. Are there any RC model files (such as tluplus or ITF) or layer/tech map files available from the contest organizers?

2. If these files are not provided, is there any recommended workflow or workaround to proceed with the Fusion Compiler flow?

I have checked all the files provided and only found ndm, lef, def, lib, db, and tf files, but nothing related to RC models or maps.

A55. Both testcase2 and testcase3 already have that information.

Q56. While reviewing Section 3.1.5 (*Flip-Flop Library Cells*) of the Problem B specification, we noticed that the _weight file in **Testcase 1** does not contain any cell names or components.

We would like to ask:

1. Will the _weight file include a list of **standard cells actually used in the design**, including flip-flops and other logic gates?
2. If so, what **format or syntax** should we expect for how these used cells will be presented?

This clarification would be greatly appreciated, as it will help us correctly parse and utilize the provided files during implementation.

A56. That's not in _weight file. Please refer to .def and .v file.

Q57. 1. Is there .spef file such that we can get the wire RC information and calculate the switching power?
2. For the problem this year, do we need to consider displacement delay?
3. For Q40 and 41, does it mean I can change a sbff to another kind of sbff and bank into mbff other than its single_bit_degenerate? Is this contradicted? For example, can we change the ff type from SNPSHOPT25_FSDN_V2_2
to SNPSROPT25_FSDN_V2_2?

A57.
1. We do not plan to release .spef file at this moment.
2. Yes.
3. Every sizing or swapping between single bit flip-flops need to be functionally equivalent.

Q58. 1. Based on Q43, different PVT conditions will result in different TNS and TPO. I wonder that how the final score will be calculated. For example, will each PVT have a weighting factor that is used to sum the results, or will a specific PVT be used for evaluation?

2. I noticed that some cells consist of both a flip-flop and a latch. Is this type of cell viewed as a flip-flop that we need to manipulate (e.g., banking / legalization) in the

contest?

For example, the following cell, SNPSHOPT25_SSRRDPQ_1.

```
cell(SNPSHOPT25_SSRRDPQ_1) {
  area : 1.332 ;
  cell_leakage_power : 3508 ;
  ecsm_vtp : 0.3392 ;
  ecsm_vtn : 0.3244 ;

  ff(IQ,IQN) {
    clocked_on : "(CK&SR)" ;
    next_state : "D" ;
    clear : "(CK|SR)" ;
    preset : "(CK|SR)" ;
    clear_preset_var1 : L ;
    clear_preset_var2 : L ;
  }

  latch(IQ2,IQ2N) {
    enable : "(!SR)" ;
    data_in : "IQ2" ;
  }
```

3. How can we check whether flip-flops still maintain compatibility and functional equivalence after banking or type switching? Can this be done using Fusion Compiler, or will the organizer provide the script for verification ?

A58.

1. The evaluation would be based on the worst case scenario.

2. Yes.

3. By functionally equivalent it is expected that there is no bad logic resulted in the optimization process.

Q59. I would like to raise a concern regarding the constraints described in Section 5 (Evaluation), specifically points A and B:

A. All instances must be placed on-site and within the die region.

B. No overlap is allowed.

We believe these constraints, as currently stated, are **not realistically satisfiable** due to the presence of **overlaps between combinational cells in the original testcase**, such as in testcase1. Since these cells are already placed and overlaps exist **prior to any modification**, it is **impossible** for contestants to meet constraints A and B if they are strictly applied to **all instances**.

However, in **Q36**, you clarified:

"We expect contestants to move only flip-flops. Contestants does not need to move or consider the legality status of combinational cells."

This clarification is appreciated, but it seems to contradict the current wording in the Evaluation section. If the evaluation script or judging criteria were to blindly enforce the legality of **all instances**, it could lead to undeserved disqualifications — even for teams who followed your instructions to only modify flip-flops.

Therefore, we respectfully ask:

1. Could you **revise the language of 5.A and 5.B** to explicitly state that these constraints apply **only to the flip-flops placed or modified by contestants**?
2. Or alternatively, clarify whether combinational cell overlaps in the original DEF will **not** be evaluated as part of these constraints?

Such clarification (or revision) would prevent confusion and ensure a fair and realistic evaluation environment for all participants.

Thank you very much for your time and support. We truly appreciate your efforts in hosting this meaningful competition and look forward to your guidance on this matter.

A59.
1. We'll update accordingly to make the testcase and spec in alignment.
2. Combinational cells overlaps in original DEF will not be evaluated as part of the constraints.

Q60. Is there any official checker or script available to verify the correctness of the program output before submission?

A60. We'll broadcast to contestants if possible.

Q61. 1. Do we need to consider the DRC violations of nets?

2. How can we check functional equivalence between the input and result .def and .v files ?

Will the organizers provide a checker?

3. Previously, it was only mentioned that there might be multiple .lef, .lib, or .db files. Does the Program Requirements section imply that there can be multiple .tf, .sdc, .v, and .def input files as well? And is it possible to have more than one module, not just the "top" module?

Thank you very much for your help. Any clarification would be greatly appreciated.

```
module top ( clk , in1 , out1 , in2 , out2 , in3 , out3 , in4 , out4 , in5 ,
    out5 , in6 , out6 , in7 , out7 , in8 , out8 , in9 , out9 , in10 , out10 ) ;
```

```
./cadb_0000_alpha
    -weight <weightFile>
    -lib <libFile1> <libFile2> ...
    -lef <lefFile1> <lefFile2> ...
    -db <dbFile1> <dbFile2> ...
    -tf <tfFile1> <tfFile2> ...
    -sdc <sdcFile1> <sdcFile2> ...
    -v <verilogFile1> <verilogFile2> ...
    -def <defFile1> <defFile2> ...
    -out <outputName>
```

```
Timing
------------------------------------------------------------------------
Context                              WNS           TNS           NVE
------------------------------------------------------------------------
Design            (Setup)          -0.43       -1054.86          6100

Design            (Hold)           -0.10        -458.92          6400
------------------------------------------------------------------------

Miscellaneous
------------------------------------------------------------------------
Cell Area (netlist):                        18300.35
Cell Area (netlist and physical only):      18300.35
Nets with DRC Violations:    11682
```

A61.
1. No.
2. We'll broadcast to contestants if possible.
3. It is unlikely but possible.

Q62. I have a question about the unit of TNS, power and area in cost metric, as I saw that in the alpha test result, the power score value is so so so big (even divided by weight 1000). Also the given weight file will contain TNS, area, TPO, so what is the meaning of these value, as the TPO is so so so small in the weight file. Thanks.

A62. We expect contestants submission could optimize overall score based on cost metrics.

Q63. 1. Flip-Flop Identification Method
Is the identification of flip-flops (FFs) based on specific keywords such as "ff(" or "ff_bank(" in the .lib files?
If so, we searched for "SNPSHOPT25_SSRRDPQ4_2" (as referenced in Q42) in the .lib file but found that it does not contain either of these keywords.
Does this imply that "SNPSHOPT25_SSRRDPQ4_2" is not considered a flip-flop?

2. Replacement Rules Between MBFFs and SBFFs
Based on the previous Q&A, can we assume that as long as the net connections and total bit count are correct, it is valid to freely replace one flip-flop with another—regardless of whether it's SBFF <-> SBFF or SBFF <-> MBFF?
For example, is it acceptable to replace the 1-bit flip-flop "SNPSHOPT25_FSDPSYNRBQ_V2LP_0P5" with another 1-bit flip-flop "SNPSHOPT25_FDPQB_V2LP_2", even though they have different pin counts?

3. Units in the Initial Score Formula
In the formula for calculating the initial score, are the units for TNS, power, and area given in nanoseconds (ns), microwatts (μW), and square micrometers (μm²), respectively?

A63. 1. It is based on cell attribute.
2. The swapping needs to be functionally equivalent.
3. It is based on Fusion Compiler's or ICC2's default unit.

Q64. I'm currently analyzing the impact of flip-flop banking on power consumption using the standard cell library provided in the contest. According to general expectations, banking (e.g., replacing two 1-bit flip-flops with one 2-bit flip-flop) should help reduce power—particularly clock tree power—by lowering clock pin capacitance or reducing the number of sinks. However, from the library data, the internal power and CK capacitance of the 2-bit flip-flops seem higher than those of two 1-bit flip-flops combined, and leakage is almost the same. As a result, the total power appears to increase instead of decrease.

I would like to ask:

1. Is there any explanation for why the power does not seem to decrease with banking in the current library?
2. Are there any other factors we should consider to observe the power reduction effect (e.g., changes in the synthesized clock tree or other net-level power savings)?
3. Is it possible to measure or estimate the power savings from banking more accurately using the provided files or tools?

A64.
1. There could be multiple reasons. From library design to cell selection, each part could affect the quality of result. Please refer the the .lib for cell characteristics.
2. Yes. With each optimization move it comes with changes to quality of result.
3. The evaluation will be based on report_power command from Fusion Compiler or ICC2.

Q65. Does it have hidden cases in problem B this year or not?
If so, how should we take scan chain into consideration because there's no scan chain in testcase1?

A65. Yes. There will be hidden cases for Problem B this year.

Q66. We would like to seek clarification on two issues we have encountered while working on testcase2 and testcase3:

1. Request for SPEF files:

We noticed that .tluplus and .map files are provided for testcase2 and testcase3, but .spef files are not. We would like to understand the reason behind this.
The .tluplus files appear to contain a large amount of encrypted data, making it difficult to extract RC information directly. As far as we understand, it is only possible to convert .tluplus to .spef using Synopsys tools such as Fusion Compiler. However, since Synopsys tools are not allowed in this contest and with only about one month remaining, it would be extremely difficult for us to develop a custom parser and RC extraction tool from .tluplus(.itf) + .map within the competition timeframe.
Furthermore, we believe that such effort would deviate from the core focus of this contest. Therefore, we sincerely hope the organizers could consider providing the .spef files.

2. Clarification on “worst case scenario”:
In A58.1, it is stated that "The evaluation would be based on the worst case scenario." However, the TNS value given in testcase2_weight is 3988.15, which differs from what we computed under what we believe to be the true worst case PVT corner. After testing various PVT settings, we observed that the actual worst-case TNS for testcase2 should be 6645.68. Similar discrepancies also appear in testcase1 and testcase3.
Could you please clarify what is meant by “worst case scenario” in this context, and which corner (or setup) we are expected to evaluate our solution against?

A66. 1. After careful consideration, we must respectfully decline your request for .spef file generation. This decision aligns with the contest's established evaluation framework, which utilizes standardized default settings to ensure a level playing field for all participants.

We would like to clarify that the .tluplus and .map files included with the updated testcases are provided for reference purposes only. These files are not intended to serve as the definitive source for RC extraction data in the contest evaluation process. Generating and distributing .spef files could potentially mislead contestants regarding the actual evaluation criteria. The evaluation procedure would be based on default environment scenarios. We did not endorse taking the RC extraction data given along with the testcase as golden.

2. The worst case scenario is derived via Fusion Compiler / IC Compiler II’s default mode. In our evaluation environment we derive the value through default scenario

calculation. We have reviewed our settings and we do not think TNS is anything near 6645.68.

Q67. May I ask whether the LEF and LIB files used in the contest are fixed across all testcases?
Specifically:
· Can we assume that no additional LEF or LIB files will appear in the final evaluation testcases?
· Will the content of the current LEF and LIB files remain unchanged throughout the contest?

A67. There will be no further LEF and LIB files used in this contest.

Q68. We have some questions regarding the alpha test results and the flip-flop pin naming convention in the provided library:

1. **Alpha test results clarification**:
We noticed several discrepancies when comparing the alpha test results with the initial values provided in testcase1_weight. Specifically:

o For **area**, it seems that the alpha test only accounts for flip-flops, while testcase1_weight includes both flip-flops and gates. Could you confirm which definition should be used?

o For **power**, the score appears to be calculated using picowatts (pW) directly, rather than the 10^9 pW unit specified in testcase1_weight for TPO. This leads to the power score being scaled by 10^9, effectively dominating the overall score. Could you clarify which unit should be used?

o Additionally, the **TPO** initial value in testcase1_weight seems to consider only the register power from the report_power output, while the alpha test appears to include the *clock network* as well. Could you confirm which value (register only or including clock network) should be used for TPO calculation?

2. **Flip-flop pin naming in the library**:

After reviewing the .lib file of testcase1, we would like to confirm if our observations regarding flip-flop pin naming are correct:

o For **1-bit flip-flops**, the data and output pins are always named D and Q.

o For **multi-bit flip-flops (MBFFs)**, the data and output pins follow the pattern D1, D2, ..., and Q1, Q2, ...

o When banking flip-flops into a new MBFF, the **SI** pin of the new MBFF corresponds to the SI of the flip-flop originally placed at D1/Q1, and similarly, the SO pin corresponds to that of the last flip-flop in the group.
Could you please confirm whether this understanding is correct?

A68. 1-1 Please refer to our release for public calculator script.

1-2 Please refer to our release for public calculator script.

1-3 Please refer to our release for public calculator script.

2-1 The pin naming would be library-dependent.

2-2 The pin naming would be library-dependent.

2-3 Not necessarily. Banked SI pin should have corresponding pin according to compatibility. Banked multibit may have independent SO pin or referring to the highest Q bit of the multibit, given that the banked multibit has sequential scan chain.

Q69. In testcase1, there are four macro types: SNPSHOPT25, SNPSLOPT25, SNPSROPT25, and SNPSSLOPT25.
Each macro type's Liberty file contains multiple kinds of single-bit flip-flop cells. Some multi-bit flip-flops in the Liberty files include the single_bit_degenerate attribute.

Is it allowed to replace one single-bit flip-flop cell with a different one within the same macro type?

For example, is it valid to replace SNPSSLOPT25_FSDN_V2_2 (a 1-bit flip-flop cell) with SNPSSLOPT25_FSDN_V2_1 (another 1-bit flip-flop cell of the same macro type)?

A69. We allow sizing between single bit flip-flops if the move does not disrupt its logic equivalence.

Q70. The verilog file in testcase2 has nested modules, I would like to ask if I should follow the module structure, it means I cannot merge two flip flops with two different modules. Besides, should I output the verilog file with the same module structure or I can just use a "top" module like testcase1.

A70. Cross-hierarchy banking would not be logically consistent.
Please output the Verilog with the same module structure.

Q71. While reviewing the provided standard cell library for Problem B, I noticed that all flip-flop cells with scan functionality contain SI and SE pins, but I could not find any SO (scan-out) pins in the LEF/LIB definitions.
May I confirm whether the SE pin should be interpreted or treated as the SO pin in this problem?

A71. SE pin is input pin. SO pin is output pin. We do not think an input pin can be logically equivalent to an output pin.

Q72. 1. According to the LEF/DEF Language Reference, the scan chain should be represented by statements like those shown in Pic. 1. However, testcase 2 and 3 do not contain such statements; instead, they just use the net-VSS to connect all the scan pins. Does this mean that there is no scan chain rule in this contest?
Pic 1.

```
SCANCHAINS 2 ;
- the_chain
  + COMMONSCANPINS ( IN PA1 ) ( OUT PA2 )
  + START PIN scanpin
  + STOP  cell4 PA2
  + ORDERED
     cell2 ( IN PA0 )
     cell1 ( OUT P10 )
  + FLOATING
     scancell1 ( IN PA0 )
     scancell2 ( OUT P10 )
  ;
- chain2
  + START comp1 scanpin
  + FLOATING float1 ( IN P1 ) float2 ( OUT P2 ) float3 ( IN P1 ) ( OUT P2 )
  + ORDERED C1 ( IN P1 ) ( OUT P2 )
    C2 ( IN P1 )
    C3 ( OUT P2 )
    C4 ( IN P1 ) ( OUT P2 )
    C5 ( OUT P2 )
  + STOP PIN
  ;
END SCANCHAINS
```

2. In testcase 2, can SBFFs from different hierarchies be banked into an MBFF? If so, does this mean we need to modify the netlist information (such as net names, ports, and wires) in the .def and .v files to implement it? For example, would we need to create new ports and wires in the .v file within certain module hierarchies to implement the MBFF connections?

If the original net information in the .def file is:

- a/b/net_1
  (a/b/c/c_1 X)
  (a/b/c/d/FF_1 D)

and it becomes:

- a/b/net_1
  (a/b/c/c_1 X)
  (a/MBFF_1 D)
  should we change the net name to a/net_1?

A72. 1. There will be hidden testcases for this contest.
2. Cross-hierarchy banking would not be logically consistent.

Q73. May I confirm whether all LIB files from the provided library folders (SNPSHOPT25, SNPSLOPT25, SNPSROPT25, SNPSSLOPT25) will be included as inputs for every testcase?
Or will each testcase only include a subset of these LIB files?

A73. All LIB files are provided for each testcase.

Q74. Thank you for clarifying the difference between the SE and SO pins. However, after reviewing all the provided LIB and LEF files, I could not find any flip-flop cell that contains an SO pin definition.
May I confirm whether this is expected for the contest?

A74. That's the case for the given testcase.

Q75. Regarding the final evaluation:
· Will there be hidden testcases used for the final scoring? If so, could you indicate approximately how many there are?
· Will the public testcases be included in the final score calculation?
· For the Alpha result, is it evaluated solely on testcase1, or are other cases also involved?
· For determining the final ranking and the winner, how are scores aggregated across all official and hidden testcases? For example, is it based on the total sum, the average, or a weighted metric across all testcases?

A75. Yes. There will be hidden testcases.
Yes. Public testcases are included in the final score calculation.
Alpha only evaluated testcase1, Beta evaluated a total of 4 testcases.
The score for each testcase will be normalized and the result will be weighted by difficulty.

Q76. We have a few questions regarding the provided libraries, netlist handling, and module notation:
1. In a previous response, you mentioned that although there are hidden cases, no new libraries will be provided. However, after reviewing the entire library, we found that

none of the flip-flops have an SO pin. Does this mean that scan chains will never appear in the contest cases, since no flip-flop contains an SO pin?

2. or unused pins, we noticed that some are connected to VSS while others are connected to SYNOPSYS_UNCONNECTED_xxxx. When handling unconnected pins, are we allowed to connect all of them to VSS? Alternatively, can we replace connections originally tied to VSS with newly created unconnected wires? For example:

- Original:
- SNPSSLOPT25_FSDN_V2_2 mid12__448 (.D (qo_foo12) , .SI (VSS) , .SE (VSS) , .CK (clk) , .Q (\q_mid12[447]) , .QN (SYNOPSYS_UNCONNECTED_1207) , .VDD (VDD) , .VSS (VSS));
- Modified to tie QN to VSS:
- SNPSSLOPT25_FSDN_V2_2 mid12__448 (.D (qo_foo12) , .SI (VSS) , .SE (VSS) , .CK (clk) , .Q (\q_mid12[447]) , .QN (VSS) , .VDD (VDD) , .VSS (VSS));
- Or modified to connect to newly defined unconnected wires:
- SNPSSLOPT25_FSDN_V2_2 mid12__448 (.D (qo_foo12) , .SI (UNCONNECTED_1) , .SE (UNCONNECTED_2) , .CK (clk) , .Q (\q_mid12[447]) , .QN (UNCONNECTED_3) , .VDD (VDD) , .VSS (UNCONNECTED_4));

3. Will you provide a checker later in the contest?

4. Are we allowed to use "/" to represent submodules and directly declare them inside the top-level top module? For example, can we directly declare a wire in top that belongs to a submodule deep in the hierarchy, such as:
wire\hier_top_mod_5/hier_top_mod_4/hier_top_mod_3/hier_top_mod_2/hier_top_mod_1/hier_top_mod_0_1/hier_mod_5/SYNOPSYS_UNCONNECTED_2219;
We have confirmed that this notation can be read by Fusion Compiler.

A76.

1. No. If you could not find any independent pin named SO pin, it means the flip-flops are not using independent SO pin.
2. The connection must be logically equivalent.
3. Please refer to the updated public_cost.tcl for how testcase scores are evaluated.
4. We allow optimization moves as long as your results are maintaining logical and functional equivalence.

Q77. We have a few questions regarding flip-flop behavior, logical equivalence, and banking:

1. We found that for some flip-flops, the relationship between D and Q is not a simple one-to-one mapping. Instead, multiple data inputs (e.g., D0 and D1) drive a single Q, with additional control signals determining which D value is assigned to Q. For example:

Q <= ((((D0 & (!S)) | (D1 & S) | (D0 & D1)) & (!SE)) | (SI & SE) | (((D0 & (!S)) | (D1 & S) | (D0 & D1)) & SI));

In theory, such a flip-flop cannot be replaced with another SBFF or be used for banking directly since the functional behavior differs. However, if SE, S, and SI are all tied to VSS, then the behavior reduces to Q <= D0. In this case, would it be considered logically equivalent if we replace this flip-flop with another Q <= D flip-flop, or perform banking using such a replacement?

2. We also found some flip-flops whose output is QN instead of Q. Can these flip-flops be used for banking?

3. If the answer to 2. is yes, can a flip-flop whose output is QN be banked together with a flip-flop whose output is Q? For example:

```
SNPSHOPT25_FSDN2_V2_1 C1 (
      .D1   (FF1_d),
      .Q1   (VSS),            // Unused → tied to VSS
      .QN1 (FF1_qn1),      // ← Key point: using QN1 output
      .CK   (clk),
      .D0   (FF0_d0),
      .Q0   (FF0_q0),       // ← Key point: using Q0 output
      .QN0 (VSS),            // Unused → tied to VSS
      .SI   (VSS),
      .SE   (VSS),
      .VSS (VSS),
      .VDD (VDD)
);
```

4. Many flip-flops have additional pins besides D, Q, SI, VDD, and VSS, such as S, SR, VDDR, RD, and RS. In all cases (including hidden cases), will these pins always be connected either to "VDD", "VSS" or "SYNOPSYS_UNCONNECTED_xxxx"?

5. If the answer to 4. is yes, are we allowed to replace these flip-flops with other SBFFs or perform banking with them?

A77.

1. If the optimization move does not compromise its logic and function equivalence under the current connection, it is considered viable.

2. Contestants are only allowed to perform optimizations on sequential cells (single-bit/multibit flip-flops), and the result has to be logically and functionally equivalent to the input design.

3. The banking is allowed if the optimization move results in logical and functional equivalence.

4. In most cases it does.

5. The banking is allowed if the optimization move results in logical and functional equivalence.

Q78. We would like to clarify several points regarding the contest specifications:

1. In this contest, can we assume that there will be no test cases where a single module definition is instantiated multiple times in the design?

2. If a module definition has multiple instances and we modify the composition (cells and netlist) of one instance, such a change would inherently modify the corresponding module definition. Would this imply that all other instances based on the same module definition must also be modified in the same way?
For example, suppose the following m1 module definition has two instances, a1 and a2:

```
module m1 (clk, in1, ...);
   input clk;
```

```
    input in1;
    ...
    sbff_lib1 sbff1 (...);
    sbff_lib1 sbff2 (...);
    ...
endmodule

module top (clk, in1, ...);
    input clk;
    input in1;
    ...
    m1 a1 (...);
    m1 a2 (...);
    ...
endmodule
```

If we bank a1/sbff1 and a1/sbff2 into a single a1/mbff1, this would require modifying the m1 module definition as follows:

```
module m1 (clk, in1, ...);
    input clk;
    input in1;
    ...
    mbff_lib1 mbff1 (...);
    ...
endmodule

module top (clk, in1, ...);
    input clk;
    input in1;
    ...
    m1 a1 (...);
    m1 a2 (...);
    ...
endmodule
```

In this case, since a2 is also an instance of m1, would a2/sbff1 and a2/sbff2 also be required to be banked into a multi-bit flip-flop in the same way?

3. In the output Verilog file, is it acceptable to leave unused pins unconnected, or should they be connected to a dummy net (e.g., SYNOPSYS_UNCONNECTED_X) instead?

4. If unconnected pins are allowed, can we also omit the connection for that pin entirely in the instance declaration?
For example, consider the following module:

```
module example (input clk, input in1, input in2);
   ...
endmodule
```

If in2 is unused, can we instantiate it like this:

```
example u1 (.clk(clk), .in1(sig1));
```

instead of explicitly writing:

```
example u1 (.clk(clk), .in1(sig1), .in2());
```

5. Regarding the libraries provided, are all multi-bit flip-flops (MBFFs) implemented as parallel scan bits rather than serial scan chains? We observed that in the current MBFFs, each bit's scan input is connected to the same SI rather than to the scan-out of the previous bit.

6. If the answer to (5) is yes (i.e., MBFFs are parallel scan bits), we also noticed that there is only a single SI pin, and all bits share this same scan input. Does this mean that in the case where scan chains exist (i.e., when SE is not tied to VSS or SYNOPSYS_UNCONNECTED_X), only SBFFs that share the same SI can be banked together, and furthermore, they cannot belong to the same scan chain (since in a scan chain the scan input of the next stage must be the scan output of the previous stage)? We would like to confirm if our understanding is correct.

7. We also observed that some flip-flops provide only a Q pin or only a QN pin, with one serving as the scan output and the other as the inverted scan output. Is it possible for both types of flip-flops (with Q vs. with QN) to appear in the same scan chain?

8. If the answer to (7) is yes, and in fact the Liberty library contains MBFFs that implement serial scan chains, would it then be possible to perform banking in such cases?

A78.

1. No two modules can share the same name.
2. I think you get the causal relationship wrong. If your optimization resulted in a2/sbff1 and a2/sbff2 getting banked, you could reuse them with the same module. If not, two separate module definitions are still needed.
3. The contestant's output is expected to be logically and functionally equivalent to the input design.
4. Please remain logical and functional equivalent. Differences in unconnected pin load and pin connected to a flag wire may lead to logic discrepancy.
5. If a multibit is connected to only one scan input. It is called a multibit flip-flop with serial scan chain. There could be both parallel scan bits and serial scan chain multibit flip-flops.
6. No. Your understanding is incorrect. If a multibit is connected to only one scan input. It is called a multibit flip-flop with serial scan chain, and its compatible SBFFs should have scan chain connected in sequential order. Please refer to Figure 5 from the Problem Description.
7. It is possible.
8. Banking would be only possible if the result is logically equivalent. If the library does not have any possibility to negate the QN scan chain into a banked multibit, it would not be possible.

Q79. Could you provide a testcase with scan chain rules which we can use to verify our program?

A79. Currently only testcase1, testcase2, and testcase3 are public testcases.

Q80. I have a question regarding the set_scenario_status settings in public_cost.tcl. Currently, the hold status for cold.BC, cold.WC, and norm.BC is set to false. After running original tests, I observed that in the generated report_qor_summary.rpt, the Timing (Hold) section shows WNS, TNS, and NVE values as either zero or --. Could you please clarify:

1. Should I modify all set_scenario_status -hold false statements to true in order to obtain meaningful hold timing results?

2. Does cold.WC indeed refer to the "worst case" scenario?

A80.
1. You could modify scenario status for your own testing.
2. The (Design) row refers to the worst case scenario. Please refer to parse_report_qor_summary.py for more information.

Q81. While testing the provided public_cost.tcl script in Fusion Compiler, It seems that all the testcase.sdc files provided are in SDC 2.2 format, while the script public_cost.tcl requires SDC 2.1 in Fusion Compiler.

We would like to confirm: Should we manually convert the .sdc files to version 2.1 for compatibility? Or will the organizers provide updated .sdc files that match the required version?

A81. Contestants are allowed to parse the sdc file with on their own. Contest Organizer does not intend to release any other version of sdc file as in terms of content there is no difference between each version, except for the version number itself.

Q82. I would like to ask for the clarification of library_name in {instance_name, library_name, bit_width}, for example, would it be “snps25lopt_base_ff0p88v25c” or "snps25lopt_base_ff0p88v25c.lib" or "SNPSLOPT25" or anything else?

A82. “library_name” stands for “library cell name”. It’s none of your mentioned names. You could find them at the .lef file.
Below is an example creation:
create_multibit
{ {hier_top_mod_5/hier_top_mod_4/hier_top_mod_3/hier_top_mod_0_2/hier_mod_4/mid12__86 SNPSLOPT25_FSDN_V2_4 1}
{hier_top_mod_5/hier_top_mod_4/hier_top_mod_3/hier_top_mod_0_2/hier_mod_4/mid12__278 SNPSLOPT25_FSDNQ_V3_4 1}

{hier_top_mod_5/hier_top_mod_4/hier_top_mod_3/hier_top_mod_0_2/hier_mod_4/ newCell SNPSLOPT25_FSDN2_V2_1 2} }

Q83. In 3.2.1, for the list of pin mappings between input flip-flop pins and output flip-flop pins, is it only necessary to print the mappings for pins such as "D", "Q", "QN", and "CLK"?

A83. No. We expect every pin mapping is included.

Q84. I would like to ask a few questions regarding Problem B:

1. Scan Chain: In the provided LEF and LIB files, FF cells have SI as Scan Input, SE as Scan Enable, and Q as Scan Output (SO). Is this correct? The problem description mentions that scan chain connections are SI and SO, but in Q71 it is stated that SE is an input pin and SO is an output pin. This creates some confusion. Could you please clarify?

2. in Mapping in .list File: Related to the above, in the output .list file, the pin mapping is required to include D, Q, and CLK. Do we also need to include QN, SI, and SE pins?

3. Blockages in Testcase 2: In Testcase 2, the DEF file contains blockages. When placing flip-flops, should we avoid overlapping with these blockages?

4. Banking Constraints: Regarding the banking_compatible.rpt file: Apart from the Single-bit FF libraries mentioned in this report, instances of other FF libraries cannot be banked into Multi-bit FFs, correct? Also, any Multi-bit FF library not listed in the report cannot be used as the library for banking operations, correct?

5. Placement in Testcase 1: In Testcase 1, the original instances are not placed according to the placement rows. Do we need to adjust the placement of all FFs, even those that are not banked into Multi-bit FFs?

6. Library Optimization (Q51 6a & 6b): Q51 mentions that 1-bit FFs that do not use scan chains or QN pins could be replaced with libraries that do not include scan

chains or QN, instead of banking them into 4-bit FFs. Would using this approach to save area and power violate the intended rules of the problem?

A84.
1. The “scan chain connection” itself refers to the SO-SI connections from one register to another. SE is the enablement of scan mode.
Site note: please refer to the latest Problem Spec as scan chain consideration is relaxed.

2. All pin mappings are needed.

3. Yes. Failure to consider the blockages would result in legality issues.

4. No. The list is provided to help contestants get a head start on compatibility. Those that are considered compatible in the list could be banked/debanked accordingly with functional equivalence. Instances of other FF libraries could also be banked into Multi-bit FFs as long as it is logically and functionally equivalent. It does not mean any Multi-bit FF library not listed in the report cannot be used as the library for banking operations.

5. Yes. Contestants are in charge of the legality all the output flip-flops, albeit the input is legal or not.

6. We are not sure about which intended rules are you referring to. The goal of this contest is to allow optimization moves on sequential cells that could improve timing, power, and area in a composite cost function while maintaining logically and functionally equivalent. Hence, we allow optimization moves, including but not limited to banking, debanking, cell sizing, as long as the moves remain logical and functional equivalent.

Q85. I would like to ask about flip-flop banking. If we have the same submodule instance under different hierarchies (e.g., hier1/hier2/A and hier3/hier2/A), is it allowed to bank one but not the other?
If yes, how can we preserve module in verilog file?

A85. No.

Q86. I would like to ask if a pin is originally UNCONNECTED, does it need to be included in the pin mapping output?"

Following up on my previous question, I would like to ask: if a pin is originally VDD or VSS, should these also be included in the pin mapping output?

A86. Yes. Yes.

Q87. As you mentioned earlier, there are four 1-bit flip-flops, each with SI and SO pins. However, the 2-bit and 4-bit flip-flops may have only one pair of SI and SO pins, or in some cases, no SI/SO pins at all.
In both the netlist and the DEF, the SI/SO information is included. So, if we group (or bank) four 1-bit flip-flops with SI/SO into 4-bit flip-flops without SI/SO, and assuming scan chain constraints are not considered as previously mentioned — is this approach still valid?
Additionally, if we group four 1-bit flip-flops with SI/SO into a 4-bit flip-flop that has only one pair of SI/SO pins, and again assuming scan chain constraints are not considered would this also be a valid approach?

A87. Since there's no scan constraint, the scan mapping is not needed and hence would not impede logic & functional equivalence. Hence, the mismatch of the dangling SI/SO pins would not cause functional inequivalence.
When the design has no SI/SO pin connections at all, the banking would not lead to functional inequivalence due to SI/SO mismatch.

Q88. Since the contest deadline is approaching and we highly value this competition, we would like to confirm that our output format meets the official requirements. Therefore, we kindly ask for your clarification on the following questions regarding Problem B:
1. In the contest specification, it is mentioned that no scan chain will appear. Do we still need to include SE/SI mapping information in the output list? If yes, when four single-bit FFs are banked into one multi-bit FF, there will only be one pair of SE/SI pins, while originally there were four pairs. Could you provide an example of how this should be represented?

2. Following the above, suppose we have a cell named FF1 (pin list: CLK, D, Q, QN) where QN is unconnected. If we replace it with another library cell and rename it to FF2 (pin list: CLK, D, Q), how should the mapping information for the now-absent QN pin be handled?
3. Based on the official rules for legal placement and the QA responses, is it correct that all sequential elements must follow the placement rules (i.e., conform to placement rows, not overlap with other cells or blockages), whereas other gates do not need to follow these rules and should not be moved even if their original positions are illegal in the testcase?

A88. 1. We expect every pin mapped, if possible. In your case where scan pins are present in either input or output and void in another, the mapping is not needed.
2. We expect every pin mapped, if possible. In your case every resultant output pin should be mapped.
3. Contestants should only optimize/move/transform sequential gates, and the submission result of all sequential gates should be placed on site with no sequential cells overlapped. Modification to combinational cells would lead to violation.

Q89. I have a quick question about the required output format when flip-flops in the testcase include a QN pin.

Example from the Verilog (for clarity):

```
SNPSHOPT25_FSDN_V2_1 foo1__268 (
    .D    ( \qo_in1[67] ),
    .SI ( SYNOPSYS_UNCONNECTED_667 ),
    .SE ( SYNOPSYS_UNCONNECTED_668 ),
    .CK ( clk ),
    .Q    ( q_foo1[267] ),
    .QN ( SYNOPSYS_UNCONNECTED_669 )
) ;
```

When generating the “list” file (the mapping/list output required by the contest), should we include an explicit mapping entry for the QN pin (for example: foo1__268/QN map foo1__268/QN or similar), or is it sufficient to only list the mapped Q pin? Concretely:

1. If QN is connected to a net in the testcase, should we include that QN → net mapping in the list file?

2. If QN is unconnected (or connected to a SYNOPSYS_UNCONNECTED_* net), do you prefer we omit it from the list, or include it (and if included, how should SYNOPSYS_UNCONNECTED_* be represented)?

3. When banking single-bit FFs into MBFFs, do you expect QN pins to be mapped/handled explicitly in the same way as Q pins (i.e., should MBFF pin mapping preserve both Q and QN mapping semantics)?

If there is a canonical example file or a short format spec for the list/mapping output that shows whether auxiliary pins (like QN) must be listed, that would also be very helpful.

A89. 1. Yes.
2. If the output Verilog has QN pin, we expect contestant to include the mapping in the .list file.
3. We expect contestants to handle logic and functional equivalence of every data connection, including all QN pins that have net connections.

Q90. I have another question regarding the pin mapping output. Suppose a Flip-Flop initially has a pin connected to UNCONNECTED, for example: .QN(UNCONNECTED). Later, I replace it with a Flip-Flop that does not have a QN pin. How should this be represented in the pin mapping output?

A90. We expect every pin mapped, if possible. In your case every resultant output pin should be mapped.

Q91. We will like to ask if the following operations are valid:

1. split_multibit { {bar2__173 SNPSROPT25_FSDNQ_V3_2 1} {dummy782 dummy_lib_cell 1} }
(1 bit ff -> 1 bit ff debanking)

2. create_multibit { {dummy3421 dummy_lib_cell 1} {foo1__118 SNPSSLOPT25_FSDNQ_V3_1 1} }
(1 bit ff -> 1bit ff banking)

A91. 1. No
2. No.

Q92. 1. Could you **update the** testcaseX_weight **files, including all initial scores and weights**, so that we can confirm our current TCL scripts and corresponding versions are correct? In addition, for testcase1, the TPO value was originally given as **9.3** with a corresponding weight of **100**. The unit used here seems to differ from those in testcase2 and testcase3. Could you confirm whether this will be corrected or updated?

2. Regarding your reply in **QA83** ("A83. No. We expect every pin mapping is included."), we would like to confirm whether the following examples and operations are considered correct for the **pin mapping list**:

Consider the following flip-flop instances:

1-bit FF1_1, C1 (.SI (SYNOPSYS_UNCONNECTED_1) , .SE (SYNOPSYS_UNCONNECTED_2))
1-bit FF1_2, C2 (.SI (SYNOPSYS_UNCONNECTED_3) , .SE (SYNOPSYS_UNCONNECTED_4))
1-bit FF1_3, C3 (.SI (VSS) , .SE (VSS))
1-bit FF1_4, C4 (.SI ( ) , .SE ( ))
1-bit FF2_1, C5 with no SI&SE pins
2-bit FF3_1, C6 (.SI (SYNOPSYS_UNCONNECTED_5) , .SE (SYNOPSYS_UNCONNECTED_6))
2-bit FF3_2, C7 (.SI (VSS) , .SE (VSS))
2-bit FF3_3, C8 (.SI ( ) , .SE ( ))
2-bit FF3_4, C10 has SI&SE pins but not written in the Verilog
2-bit FF4, C9 with no SI&SE pins

a. By substituting library cells, when replacing **C1** with **C5**, how should the SI and SE pin mapping be described?
b. By substituting library cells, when replacing **C5** with FF1_1~5 (which have SI and SE pins), how should SI and SE be mapped? Should they simply be omitted?

In **QA78.4**, it was mentioned: *"Differences in unconnected pin load and pin connected to a flag wire may lead to logic discrepancy."* In this case, which of the approaches in **C1–C4** should be used to represent SI and SE?
c. By substituting library cells, when replacing **C3** with **C5**, how should the SI and SE pin mapping be described?
d. By substituting library cells, when replacing **C4** with **C5**, how should the SI and SE pin mapping be described?
e. By banking {C1, C2} into **C6** and creating a new instance **C6**, should the pin mapping be written as:

C1/SI → C6/SI; C1/SE → C6/SE;
C2/SI → C6/SI; C2/SE → C6/SE;

(i.e.,
mapping SYNOPSYS_UNCONNECTED_1 and SYNOPSYS_UNCONNECTED_3 both to SYNOPSYS_UNCONNECTED_5, and
mapping SYNOPSYS_UNCONNECTED_2 and SYNOPSYS_UNCONNECTED_4 both to SYNOPSYS_UNCONNECTED_6)?
f. Are **C6** and **C8** functionally equivalent? If they are, is it valid to write C6 in the same way as C8? If so, in the case where {C1, C2} are banked into C8, how should the pin mapping be written?
g. We have the same question as in (f), but for **C8** and **C10**, and would like to ask for clarification.

Consider also the following instances:

1-bit FF, C11 (.Q (net_1))
1-bit FF, C12 (.Q (net_2))
2-bit FF, C13 (.Q0 (net_1), .QN0 ( ), .Q1 (net_2), .QN1 ( ))

h. By banking {C11, C12} into **C13**, should pin mappings
for **C13/QN0** and **C13/QN1** also be listed? If so, how should they be described?
i. By debanking **C13** into {C11, C12}, should pin mappings
for **C13/QN0** and **C13/QN1** also be listed? If so, how should they be described?

3. In the beta test, logical equivalence was not explicitly checked. May we ask if you plan to provide a **checker** that allows us to verify logical equivalence? Otherwise, it is difficult for us to ensure that what we consider logically equivalent fully matches your expectations. We would also like to confirm whether the formats of the **pin mapping list** and the **operation log list** will be formally validated against your rules.

4. In testcase2 and testcase3, VDD and VSS are declared using supply1 and supply0. Should we retain these declarations in our output files?

5. If the answer to (4) is yes (i.e., they are required in order to maintain logical equivalence), can we instead remove the VDD and VSS input declarations entirely, omit all .VDD and .VSS pin connections in instances, and directly replace their usage with 1'b1 and 1'b0? We ask this because when using Fusion Compiler's write_verilog command, it automatically generates Verilog in this format.

6. The problem is about handling the flip-flop banking/ debanking issue, but we noticed that there are latches included in the "the compatible cell list". Does this mean we also need to handle latches?

7. In our understanding, $ff in public_cost.tcl seems to be selected using the command "get_cells -hierarchical -physical_context -f is_sequential".
We would like to ask: Does this also include other sequential cells that are not flip-flops?
Similarly, when checking legality, do you also use the is_sequential attribute for cell selection?
Does this mean that in all our operations, we need to consider not only flip-flops, but all sequential cells?

8. For cell sizing, is there a need to output the new cell and old cell pin mapping since their instance names are the same? If we need to output the list of pin mapping, what is the format?

9. For the flip-flops that are not banked or changed, do we need to output the list of the pin mapping?

A92. 1. We're reviewing it and see if necessary update is needed.

2a. We expect every pin mapped, if possible. In this case C5 has no SI and SE pin hence they would not be mapped
b. The void of SI and SE pin makes mapping the two pins impossible. They should be omitted. Whichever amongst C1 to C4 was used to size C5, the SI and SE should remain unconnected.
c. We expect every pin mapped, if possible. In this case C5 has no SI and SE pin hence they would not be mapped
d. We expect every pin mapped, if possible. In this case C5 has no SI and SE pin hence they would not be mapped

e. You should map the preceding bit's SI to the banked SI. In this instance, if you are banking {C1, C2} and it is C1 precedes C2, then C1/SI -> C6/SI; C1/SE -> C6/SE.
f. You are only listing SI and SE pins and the two pins are not inequivalent. It is valid to write Verilog description in the same fashion. The preceding bits should map to the resultant SI & SE pin.
g. For .list file we are expecting every pin mapped, if possible. Consider the intersect between the input pins and the output pins and enumerate every pin in the intersect.
h. No.
i. No.

3. We will use commercial tool for formal verification. The operation log list will be used to verify if contestants move qualify and pass formal verification and the pin mapping list would supplement the context.

4. Please retain the declarations.

5. We expect contestant could produce the result using their own programming. Outsourcing the program to Fusion Compiler is not permitted. Please note that contestants are responsible for the proprietary ownership of their submissions.

6. The input testcases would be flip-flops and the contest is focused on flip-flop optimization.

7. This command does include all sequential cells but in contests domain the sequential cells means flip-flops. We use this parameter to check legality. Contestants need to handle all flip-flops.

8. Yes. Please follow the description from 3.2.1.

9. Yes.