Scalable Logic Gate Sizing using ML Techniques and GPU Acceleration

Bing-Yue Wu[1], Rongjian Liang[2], Geraldo Pradipta[2], Anthony Agnesina[2],
Haoxing Ren[2], and Vidya A. Chhabria[1]
[1]Arizona State University [2]NVIDIA Research

0. Version History

02/01/2024: Initial draft of the problem statement sent for review
02/15/2024: Feedback from the review. Minor changes.
05/18/2024: Updated to include SPEF file. Modification from post-placement to post-routing. Update the number of benchmarks. Add links to the scripts.
5/25/2024: Updated document to include lef files as input; added a link to .size files as sample outputs; updated k1 and k2 values to be pin-specific;
6/13/2024: Updated evaluation script to include legalization and global routing. Updated testcases to be post-placement instead of post-routing. SPEF was removed due to a discrepancy between OpenROAD and commercial tool global routing. Updated evaluation script to run legalization (detailed placement) post-sizing, global routing post sizing, and updated timing numbers post-sizing to make the problem more realistic. Added four benchmarks with post-placement def on the GitHub repo.
6/21/2024: Updated the evaluation score and specified the values of the penalty terms in the document and on GitHub evaluation script with the correct units and penalty terms. Updated OpenSTA sub module due to bug in OpenROAD/OpenSTA on equivCells. Added responses to more questions. Added that SRAM cells cannot be sized.
6/23/2024: Update NV_NVDLA_partition_p testcase by changing clock period from 0.4ns to 0.5ns.
6/27/24: Updated details on the submission process. Fixed a bug in the evaluation script. Updated the docker file.
7/11/24: Added a new visible testcase on github.
7/20/24: Updated the scoring function for more details on runtime and updated the scoring function to penalize those participants who do not really size the design.

1. Introduction

With scaling and the slowdown in Moore's law, EDA tools must work increasingly harder to achieve power, performance, and area (PPA) specifications. A critical step for PPA improvement is timing optimization, which, today, is heuristic-based and may not result in optimal solutions. An integral part of timing optimization is the NP-hard logic gate sizing problem, which involves selecting a size for every netlist instance from a set of choices in the standard cell library, each with a different delay/area/power. The goal is to assign gate sizes to each instance to minimize the area/power of the circuit while satisfying delay constraints. The problem is challenging due to (i) the non-convexity of the delay models, (ii) the discrete space of gate sizes, (iii) the number of near-critical paths, (iv) the tradeoff between power and timing, and (iv) the resource availability constraints which prevent the exploration of the complete design space. Logic gate sizing has been studied for over three decades now. In early work, the gate sizes were assumed to be continuous, and the timing models were either derived from RC Elmore delay models, which can be transformed into a convex function of sizes or approximated by a convex function [1]. Later, for discrete gate sizing, several heuristics emerged that used sensitivity-based methods [2], convex programming [3], and Lagrangian relaxation (LR)-based [4], [5] methods. LR-based methods have found tremendous success, but under realistic nonconvex timing models, they use heuristics to choose the order of gate sizing and work over a small local neighborhood of the circuit [4]. In the recent past, machine learning techniques have been leveraged to speed up LR [6], reinforcement learning (RL) has been applied to address the gate sizing problem [7], and generative AI-based techniques have been used for netlist optimization [8]. However, ML-LR is still based on heuristics, and RL techniques are not scalable.

The contest aims to overcome these challenges of gate sizing techniques and drive academic research in timing optimization. In this context, the contest serves three purposes:

1) Lower the barrier to entry for non-EDA experts by converting traditional EDA problems to an ML-solvable problem through ML-friendly data representation formats.
2) Explore the state-of-the-art algorithms for gate sizing to drive academic research to generate scalable gate-sizing algorithms using GPU acceleration or machine learning (supervised, unsupervised, or reinforcement learning-based techniques).
3) Release updated benchmarks in a FinFET technology node with evaluation scripts to accelerate research in logic gate sizing.

Contest goal: The goal of the contest is to develop a scalable sizer that solves a constrained optimization problem, which is formulated as:

$$minimize \sum_{i \in I} LeakagePower_{c_i}$$

$$subject\ to\ slack\ (pin_i)\ \geq\ 0;\ slew\ (pin_i)\ \leq\ k_1(pin_i);\ load(pin_i)\ \leq k_2(pin_i)\ \forall\, i\ \in\ I$$

$$c_i\ \in C_i\ \forall\, i\ \in I$$

where I is a set of instances; C_i is the set of choices for instance i in the library; $c_i \in C_i$ is the library cell assigned to instance i; $Power_{c_i}$ is the power of instance i; $slack\ (pin_i)$ and is slack at all pins of instance i; $slew\ (pin_i)$ and is the transition time at all pins of instance i, $load(pin_i)$ is the load capacitance seen at the **output** pins of instance i, and $k_1(pin_i)$, and $k_2(pin_i)$ are constants and obtained from the library file. The goal is to minimize the generated solution's leakage power while meeting slack, max slew, and max load constraints.

2. Input/Output Files and Formats

Input Formats

The contest will provide inputs in two formats. The first is standard EDA files, and the second is CircuitOps' intermediate data representation format. Both formats are explained below:

(i) Standard EDA file formats:

- .v file: Gate-level verilog netlist
- .def file: Post-placement design cell locations and route locations
- .sdc file: Constraint file provided in TCL to specify input port delay and transition, output port capacitance, and specified clock period
- .lib file: Library file that consists of the look-up-tables for delay, slew, and power computation with the area of each cell
- .lef file: Technology lef and cell lef

If participants choose to work with these files, they are expected to work with them as input and develop parsers. If participants choose to develop their sizer in C/C++, they can leverage OpenROAD [9] and OpenSTA (submodule in OpenROAD) [10] header files to parse liberty and netlist. If participants choose to develop their sizer in Python, they can use OpenROAD Python APIs to query the necessary information, or they can reuse existing Python packages such as liberty-parser [11] and verilog-parser [12]. Participants can also work with their parsers if they cannot extract the information they need from existing APIs. We have provided example APIs in scripts for querying information from OpenROAD DB if you choose to use OpenROAD as the parser and work within the OpenROAD Python interpreter (link).

(ii) CircuitOps data representation format:
The CircuitOps data representation format provides an ML-friendly format consisting of graphs and annotated features. The graph is a labeled property graph that is backed by relational tables. The graph utilizes a node for a net, pin, and cell of a circuit. It has four types of edges to represent connections between pin-to-pin nodes (pin-pin), pin-to-cell nodes (pin-cell), pin-to-net nodes (pin-net), and cell-to-net nodes (cell-net). Each node and edge has associated relational tables with properties associated with the node and edge. The relational tables will be provided to the contestants, which can easily be parsed into their sizer using standard CSV readers in C++/Python. Details on the list of properties and their description and types of nodes and edges can be found in the Appendix, the CircuitOps paper [13], and the GitHub repository [14]. **These files will be in a CSV format.**

The choice of the input format is left to the participants. The first EDA file-based format will provide you with maximum information to develop your gate sizer and the ability to extract custom features for ML models. The second will be limited to the features and properties available in CircuitOps, but it will allow you to skip the tedious parsing tasks and focus on the ML/GPU acceleration aspects. We will release translators from standard EDA file format to CircuitOps format (the translators will leverage OpenROAD APIs and database queries). Please note that the translators are slow and cannot be done iteratively within your sizing algorithm. Some of the properties in CircuitOps can be computed fast from OpenROAD (the properties that come in the intermediate files), but not all properties.

Intermediate Files
In case you are using CircuitOps and do not want to build your independent timing engine, we have provided you with a script (link) that takes the temporary output file (see format below), the input EDA files, and generates the following outputs. The runtime for this script is about one minute for the "NV_NVDLA_partition_m" benchmark. These files will provide timing, power, load capacitance, and power information from the OpenROAD timer. The timing information (.csv) will be stored in a file with the format below:

```
<pinNam or ortName>, <maxcap>, <maxtran>, <pin_tran>, <pin_slack>,
<pin_rise_arr>, <pin_fall_arr>, <input_pin_cap>, <output_pin_cap>

<instanceName>, <leakagePower>
```

Output File Format
The output file (.size) should have the following format:

```
<instanceName> <libCellName>
```

3. Benchmarks

We have released benchmarks in the ASAP7 technology node (link). The benchmarks will vary in size from 10K-1M instances. We will release five visible benchmarks and five hidden benchmarks on which we will evaluate your solutions. For the five visible benchmarks, we will also provide reasonably good quality (.size) output files for the ability to use supervised learning-based machine learning techniques. Here is one example.

4. Developing the Gate Sizer: Rules, assumptions, timer, and support scripts.

Assumptions: The contest will focus on optimizing setup time only. Sequential gate sizing will not be permitted. **Sizing of macros or sram cells will not be permitted.** All sequential gates in the benchmark will be rising edge triggered. The benchmarks will assume an ideal clock. There will be no clock buffers in the circuit (zero skew). The contest will be set up on post-routed benchmarks. We have provided SPEF, and certain properties of the SPEF have been annotated with CircuitOps format (refer to the appendix).

Timing Engine: You can implement your timer and power estimator within your sizer. However, the final evaluation will use OpenROAD's timer (OpenSTA). You may also use OpenROAD's timer and power estimator as part of your sizer. We will provide examples of Python APIs interacting with the OpenROAD timer (link). The APIs will be in Python and white-boxed. If you are developing your code in C++, you can utilize the corresponding C++ APIs that the Python APIs wrap around within OpenROAD.

Support Scripts: We will provide two types of support scripts:

(i) It will use standard EDA tool files as an example and operate with OpenROAD Python APIs to update gate sizes and obtain updated timing and power numbers. We will provide scripts that operate with this format with either File I/Os (input: standard EDA files and .sizes file and outputs: timing and .power files) or directly with function calls returning standard Python data structures (input: python list of sizes and output: python list for timing and power values) (link).

(ii) It will use CircuitOps as input, it will read the design files, update gate sizes using OpenROAD Python APIs (based on an input .size), and provide updated timing and power numbers, and update the CircuitOps format with either File I/Os (input: relational tables CSVs and EDA file formats and output: table CSV same as intermediate files) or directly with pandas data frames as inputs and outputs (link).

6. Evaluation: Metrics, Platforms, Scripts, and Submission

Metrics: The sizer will be evaluated on the quality of the generated solution and runtime. The following cost function will be used to score the submissions:

$$\begin{aligned}
Score = (\delta\ + runTimeF) \\
* \Big((Power_{tot} - OrignialBenchmarkPower_{tot}) \\
+ \alpha * abs(TNS) \\
+ \sum_i \beta_i * (worstSlew(pin_i) - k_1(pin_i)) \\
+ \sum_i \gamma_i * (maxLoad(pin_i) - k_2(pin_i)) \\
\Big)
\end{aligned}$$

where $\alpha = \{0\ if\ worstSlack \geq 0,\ 10\}$

$$\beta_i = \left\{0\ if\ worstSlew(pin_i) \leq k_1(pin_i),\ else\ 20\right\}$$

$$\gamma_i = \left\{0\ if\ maxLoad(pin_i) \leq k_2(pin_i),\ else\ 20\right\}$$

$$runTimeF = maximum\{log_2(runTime/medianRunTime),\ \lambda\}$$

Units: The $Power_{tot}$ and $OrignialBenchmarkPower_{tot}$ are in micro Watt. The TNS is in nanoseconds. The $(worstSlew(pin_i) - k_1(pin_i))$ is in nanoseconds and $(maxLoad(pin_i) - k_2(pin_i))$ is in femto Farad. δ is 6, and λ is -5. The runs that error out shall not be considered when calculating the $medianRunTime$. **The cutoff runtime is 2 hours.** If you have generated a solution before, those solutions will be used. If there is no solution after 2 hours, you will not be evaluated for any metrics.

The $runTime$ is the runtime of the benchmark being evaluated, and the $medianRunTime$ is the median of the runtimes of all teams submission's on the benchmark being evaluated. The

$OrignialBenchmarkPower_{tot}$ is the power of the released benchmark (unoptimized and downsized solution) and $Power_{tot}$ is the size of the generated solution. Note the evaluation will perform placement legalizing post sizing, rerun global routing, and update parsitics, then estimate the above score. The values of k_1 and k_2 are specific to the pins of each liberty cell and are defined in the liberty file. They can be extracted using the OpenROAD APIs as shown in this script. These values are also provided as a part of the IR tables and can be queried using OpenROAD. Each participating team will be given a score for each benchmark.

In the final evaluation, the score for each testcase will be scaled to 100 using the following equation: $100 \times \frac{Best\ least\ score\ across\ all\ team}{contestant's\ score}$. No points will be given for any failed cases, even if the .size file is successfully generated but global routing fails to complete within 2 hours. In the process of calculating the median runtime, we will not take the runtime of failed cases into account. The scores will be added across all testcases, and the top three teams with the **highest** scores will win the contest.

Due to the limited computing resource we have, for the global routing stage, we will set a two-hour limit for evaluating the .size file. If the global routing takes more than two hours to evaluate the .size file due to high overflow, we will consider it a failed case.

Platform: The submitted sizers will run on a computing platform equipped with four A100 GPU, which has a memory capacity of 80GB. The utilization of up to 8 CPU threads will be supported. We will supply a Docker image preconfigured with CUDA and some popular ML libraries to facilitate a standardized development environment. We encourage participating teams to capitalize on the potential of GPU acceleration and explore the integration of ML techniques. However, the usage of the GPU is optional, and teams are free to choose their preferred approach, whether it involves leveraging the GPU or not.

Scripts: We have released the first version of our evaluation scripts (link) so teams can test their scores on visible benchmarks. The scores reported are only for the power, worst slack, worst slew, and max load components of the score with equal weightage to α, β, and γ. The values of the weights will be updated shortly in this document and the script. The runtime-related components will be evaluated on the platform and compared with other participants.

Submission Process: Teams are **required to build a Docker image on top of the provided Dockerfile**. The Dockerfile **must** be built on this existing provided Dockerfile. More details on how to submit the Docker image (including the Google Drive upload link, the timeframe for uploading, and instructions on accessing Google Drive) will be provided by the Contest Chair through separate emails.

Please archive your Docker image to a tar file and use gzip to compress it to a **tar.gz** file (refer to https://docs.docker.com/engine/reference/commandline/save/). Then upload the tar.gz file to the Google Drive upload link shared by the Contest Chair in separate emails. Within the Docker environment, please **create a directory named “sizer” under the existing “/app” folder and place the binary/scripts in this directory (/app/sizer)**. The submission should contain a top-level “ICCAD24_sizer.sh” in the /app/sizer directory which takes the design name as the ONLY input argument and must name the output .size file using the design name. For example, if the design name is NV_NVDLA_partition_m, the output .size file should be named NV_NVDLA_partition_m.size.

Please name the Docker image and the final tar.gz file using the team ID. For example, if the team ID is cadc1234, then use cadc1234 for the Docker image and cadc1234.tar.gz for the submitted file. Please DO NOT upload other files in the beta and final submission. **Submissions not following the submission process instructions will not be graded in the beta or final submission.**

Example ICCAD24_sizer.sh file (**You are allowed to modify them, such as adding more flags, as long as the top-level "ICCAD24_sizer.sh" script only takes the design name as the only input.**):

Example 1: In case the sizer is a binary file named sizer

design_name = $1
```
/app/sizer/sizer                                                -lef
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lef -lib
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lib -def
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/design/$design_name/$de
sign_name.def                                             -IR_Table
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/IR_Tables/$design_name/
-output /app/sizer/output/$design_name.size
```

Example 2: If the sizer is a Python script file around OpenROAD or even a general Python script:

design_name = $1
```
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/build/src/openroad
-python                 /app/sizer/sizer.py                     -lef
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lef -lib
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lib -def
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/design/<design_name>/$d
esign_name.def                                            -IR_Table
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/IR_Tables/$design_name/
-output /app/sizer/output/$design_name.size
```

OR
design_name = $1

```
/app/sizer/sizer.py ./sizer -lef
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lef -lib
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/platform/ASAP7/lib -def
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/design/$design_name/$de
sign_name.def -IR_Table
/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/IR_Tables/$design_name/
-output /app/sizer/output/$design_name.size
```

The Docker images will be pulled and executed on an NVIDIA platform equipped with 4 NVIDIA A100 GPUs during the evaluation process. Specifically, we will read information from the "/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark" directory. Hidden testcases will be added to the /app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/design and /app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/IR_Tables folders.

The evaluation script will be executed using the generated .size solution. to run the submitted sizer and evaluate the generated solutions.

Please archive your Docker image to a tar file and use gzip to compress it to a tar.gz file (refer to https://docs.docker.com/engine/reference/commandline/save/). Subsequently, upload the tar.gz file to the Google Drive upload link. The sizer binary/scripts are expected to be stored in /app/sizer within the Docker image. Therefore, uploading the sizer binary/scripts separately in the submission is unnecessary. If you have any questions regarding this, please feel free to ask. Thank you!

Resource limit:

RAM: 200 GB
CPU Cores: 8 cores
GPUs: 4 NVIDIA A100 GPUs

References

[1] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," in Proc. ICCAD, 1985.
[2] J. Hu, et al., "Sensitivity-Guided Metaheuristics for Accurate Discrete Gate Sizing," in Proc. ICCAD, 2012.
[3] K. Kasamsetty, et al., "A New Class of Convex Functions for Delay Modeling and its Application to the Transistor Sizing Problem," IEEE T. Comput. Aid. D., vol. 19
[4] A. Sharma, et al., "Fast Lagrangian Relaxation-Based Multithreaded Gat, no. 7, pp. 779–788, 2000.e Sizing Using Simple Timing Calibrations," IEEE T. Comput. Aid D., vol. 39, no. 7, pp. 1456–1469, 2020.
[5] C.-P. Chen, et al., "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," IEEE T. Comput. Aid D., vol. 18, no. 7,pp. 1014–1025, 1999.

[6] X. Zhou et al., "Heterogeneous Graph Neural Network-based Imitation Learning for Gate Sizing Acceleration," in Proc. ICCAD 2022,
[7] Y. -C. Lu, S. Nath, V. Khandelwal and S. K. Lim, "RL-Sizer: VLSI Gate Sizing for Timing Optimization using Deep Reinforcement Learning," in Proc DAC, 2022
[8] S. Nath, G. Pradipta, C. Hu, T. Yang, B. Khailany and H. Ren, "TransSizer: A Novel Transformer-Based Fast Gate Sizer," in Proc. ICCAD, 2022.
[9] https://github.com/The-OpenROAD-Project/OpenROAD
[10] https://github.com/The-OpenROAD-Project/OpenSTA
[11] https://pypi.org/project/liberty-parser/
[12] https://pypi.org/project/verilog-parser/
[13] R. Liang, A. Agnesina, G. Pradipta, V. A. Chhabria and H. Ren, "Invited Paper: CircuitOps: An ML Infrastructure Enabling Generative AI for VLSI Circuit Optimization," in Proc. ICCAD 2023.
[14] https://github.com/NVlabs/CircuitOps

Appendix

CircuitOps format description:

Cell properties:
Cell_name: name of the cell
Is_seq: whether the cell is sequential (register, flip flop, etc.)
Is_macro: whether it is a macro or a standard cell
Is_in_clk: whether the cell is in clock net
X0, y0, x1, y1: the smallest x-coordinate, the smallest y-coordinate, the largest x-coordinate, and the largest y-coordinate of the cell bounding box
Is_buf, is_inv: whether the cell is a buffer or and inverter
Libcell_name: the libcell name of the cell
Cell_static_power, cell_dynamic power: the static and dynamic power of the cell at the current timing corner

Libcell_properties:
Libcell_name: name of the library cell
Func_id: The functional ID of a libcell is a unique identifier assigned based on its functionality. Libcells that perform the same function will share the same functional ID, while those with different functionalities will have distinct IDs.
Libcell_area: area of the libcell
Worst_input_cap: the largest input capacitance
Libcell_leakage: leakage power of the libcell
Fo4_delay: gate delay considering a load capacitance four times larger than the largest input capacitance, while the input slew remains a predefined constant
Libcell_delay_fixed_load: gate delay considering a predefined constant load capacitance and a predefined constant input slew

Net properties:
Net_name: name of the net
Net_route_length: total routing path length of the net
Net_steiner_length: total length of the Steiner tree of the net
Fanout: fanout of the net
Total_cap: total capacitance of the net, including wire capacitance and the sink pin capacitance
Net_cap: total wire capacitance
Net_coupling: coupling capacitance
Net_res: wire resistance

Pin properties:
Pin_name: name of the pin
X, y: coordinates of the center of the pin
Is_startpoint: whether it is a timing path starting point
Is_endpoint: whether it is a timing path ending point
Dir: direction of the pin, i.e., whether it is an input pin or output pin
Maxcap: capacitance constraint of the pin
Maxtran: transition time constraint of the pin
Num_reachable_endpoint: count of reachable endpoints
Cell_name: name of the corresponding cell
Net_name: name of the corresponding net
Pin_tran: transition time of the pin
Pin_slack: slack of the pin
Pin_rise_arr: arrival time of the pin at rising edge
Pin_fall_arr: arrival time of the pin at falling edgeq
Input_pin_cap: input pin capacitance

Pin_pin_edge:
Src: driving pin name
Tar: sink pin name
Is_net: whether the pin2pin edge is a net connection
Arc_delay: timing arc delay from the src to tar

Q&A

5/18/24

1. Q: Does the cell library provide the area information?
   A: Yes, you can find the area information in the lib files.

2. Q: I'd like to ask whether the technology LEF, LEF, and SPEF files are also part of the input during the evaluation stage, because the problem description on the contest website specifies that the input format only consists of V, DEF, SDC, and LIB files. I'm curious do I have the technology LEF, LEF, and SPEF files during the final evaluation

stage, or these files are provided solely for reference and won't be included as input files.
A: The question formulation has been updated.

3. Q: In the document of Problem C, it is stated that reasonably good quality (.size) output files will be provided. I want to ask when will the output files be released, thank you.
A: The sample .size file has been provided. (Example: design.size)

4. Q: The command "sudo apt-get install -y python3-graph-tool" ,python graph tool can not be installed under linux environment(using Ubuntu 22.04),and it says "Unable to locate package python3-graph-tool"
A: Run “sudo apt-get update” first should solved the problem. Feel free to submit issues to the github repository if you have further questions or reach out to the contest chairs.

5. Q: The data "NV_NVDLA_partition_m" seems to be different every time I download from the github.
A: We have only updated the benchmark once on May 13, 2024 and once on June 14 (updated on June 14).

6. Q: I have installed python3-graph-tool ,and its version is 2.57. However, when I run the commands "pip3 install -r requirements.txt",there is a ERROR in the picture below
A: Please follow our steps in the README file, we did not provide the requirements.txt file. Feel free to submit issues to the github repository if you have further questions.

7. Q: First, in the provided GitHub benchmark, we only see one test case. Will there be only one test case in the final test as well? Second, should we replace the cells in the given Verilog file using the five provided libraries (asap7sc7p5t_AO_RVT_FF_nldm_201020.lib, asap7sc7p5t_INVBUF_RVT_FF_nldm_201020.lib, asap7sc7p5t_OA_RVT_FF_nldm_201020.lib, asap7sc7p5t_SEQ_RVT_FF_nldm_201020.lib, asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib)? Will the hidden libraries follow a similar format?
A: Yes, more testcases will be available by Jun 13 (Please look at GH for the new testcasses). The hidden benchmarks will also use the same cell library as the public benchmarks. The following benchmarks will be released within a week.

8. Q: I recently followed option 2 from the benchmark GitHub repository to build the OpenROAD and CircuitOps environments. Although the build process for the OpenROAD binary file completed without errors, I encountered differences in the evaluation results when comparing them to the original benchmark statistics. Here's the command I used for evaluation:
```
../../OpenROAD/build/src/openroad -python evaluation.py --file_path ../../result/NV_NVDLA_partition_m/baseline/final.size --design_name NV_NVDLA_partition_m
```

The execution record is as follows:

```
OpenROAD v2.0-13992-g38a3b37e1
Features included (+) or not (-): +Charts +GPU +GUI +Python
This program is licensed under the BSD-3 license. See the LICENSE file for details.
Components of this program may be licensed under more restrictive licenses which must
be honored.
[WARNING ORD-0039] .openroad ignored with -python
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13177, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13210, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13243, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13276, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13309, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13342, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 13375, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 14771, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 14804, timing group from output port.
[WARNING  STA-1212]  ../../platform/ASAP7/lib/asap7sc7p5t_SIMPLE_RVT_FF_nldm_201020.lib
line 14837, timing group from output port.
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/asap7_tech_1x_201209.lef, created 24
layers, 9 vias
[WARNING ODB-0176] error: undefined layer (OVERLAP) referenced
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/sram_asap7_64x256_1rw.lef, created 1
library cells
[WARNING ODB-0176] error: undefined layer (OVERLAP) referenced
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/sram_asap7_16x256_1rw.lef, created 1
library cells
[WARNING ODB-0176] error: undefined layer (OVERLAP) referenced
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/sram_asap7_64x64_1rw.lef, created 1
library cells
[WARNING ODB-0176] error: undefined layer (OVERLAP) referenced
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/sram_asap7_32x256_1rw.lef, created 1
library cells
[INFO  ODB-0227]  LEF  file:  ../../platform/ASAP7/lef/asap7sc7p5t_27_R_1x_201211.lef,
created 212 library cells
[WARNING ODB-0217] duplicate VIARULE (Pad_M9) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M9_M8) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M8_M7) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M7_M6) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M6_M5) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M3_M2widePWR0p936) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M4_M3widePWR0p864) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M5_M4widePWR0p864) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M6_M5widePWR1p152) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M7_M6widePWR1p152) ignoring...
[WARNING ODB-0217] duplicate VIARULE (M2_M1) ignoring...
[INFO ODB-0227] LEF file: ../../platform/ASAP7/lef/asap7_tech_1x_201209.lef
[INFO             ODB-0127]             Reading             DEF             file:
../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.def
[INFO ODB-0128] Design: NV_NVDLA_partition_m
```

```
[INFO ODB-0130]     Created 359 pins.
[INFO ODB-0131]     Created 23856 components and 123288 component-terminals.
[INFO ODB-0132]     Created 2 special nets and 47712 connections.
[INFO ODB-0133]     Created 24119 nets and 75342 connections.
[INFO               ODB-0134]               Finished               DEF               file:
../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.def
===================================================
WNS: 0.000000 ns, worst Slew: 0.001190 ns
worst load capacitance: 0.000060 pF, total leakage power: 1.521694 uW
Score: 1.521694
===================================================
>>>
```

The result differs from those posted on GitHub. I have tried another machine and the results matched those on GitHub. However, the build process for OpenROAD was identical on both machines. I'm curious about the reason behind this discrepancy. Thanks for any insights you can provide.

A: We updated the evaluation script on May 14, 2024. Please run git pull again if you encounter such a situation, as we have already updated the evaluation script once. Feel free to submit issues to the github repository if you have further questions.

6/13/24

9. The document mentions that an A100 docker image could be provided, but it has not been released yet. Our team's method uses Python to call a C++ kernel (written with "Pybind" and uses GPU acceleration). We need to ensure that our program runs smoothly on the test machine. Could we use our own environment and provide a docker image like in previous contests?
   A. Yes you can use your own environment and provide the docker image.We will provide more details re. submission on June 18th to the updated document.

10. It appears that the timing units have been updated from "ns" to "ps" in the latest update. However, the unit is not mentioned explicitly in the SDC file. For Problem C, are all SDC timing units still "ps"?
   A. OpenROAD/OpenSTA uses the units of the first read liberty file (different liberty files have different units). So, if you read the liberty files in the exact same order as the provided scripts, then the units are in ps. So we modified the sdc to reflect these units to ps. Yes, all SDC timing units are in ps. So we recommend you read the liberty files in the same order as the reference files to maintain the consistency of the units. Note here that the evaluation script has a divide by 1000.

11. Additionally, we noticed the use of "set_false_path -from …" in SDC. Since we developed our own parser, we currently only support the "-from" based false path. Will other benchmarks use different types of false path settings?
   A. All benchmarks at least have one line of "set_false_path -from [get_ports {}]". No other type of false path setting is used across both visible and hidden benchmarks. So only set_false_path -from

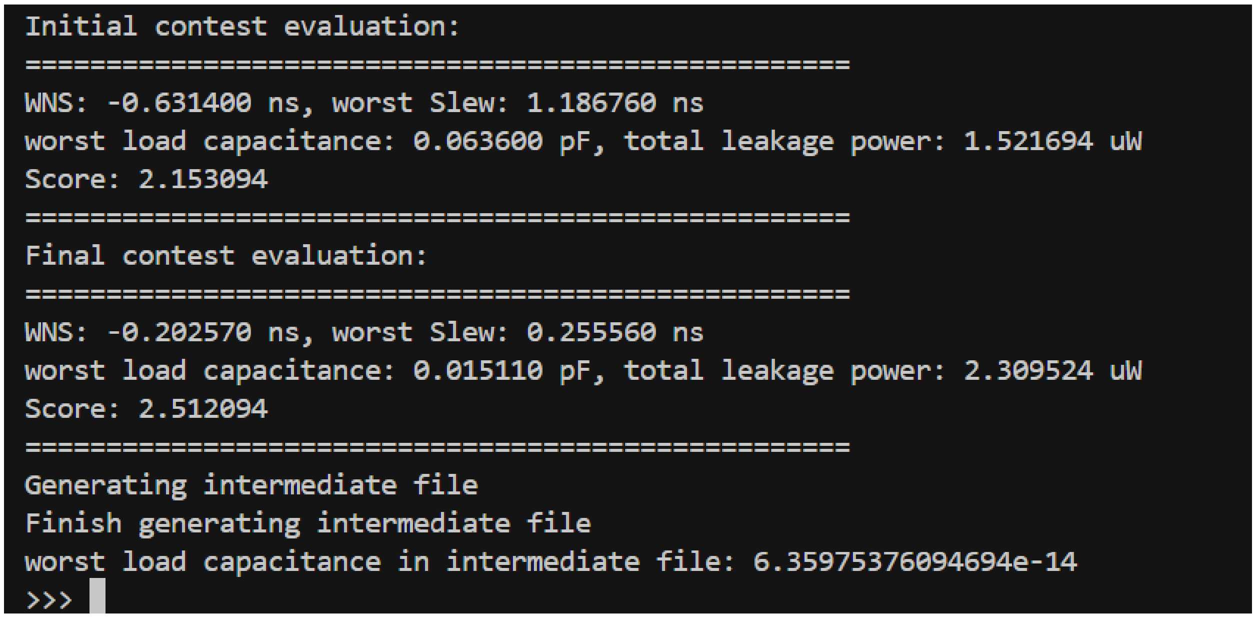
12. In Q2 of the contest Q&A, it states, “Our evaluation will be **primarily** through OpenROAD and OpenSTA (within OpenROAD).” What does "primarily" mean? If other evaluators can be used, participants should be informed.
    A. OpenROAD and internal OpenSTA will only be used for evaluation. Primarily can be deleted.

13. Given the reference to "... The final testing will not be conducted on the provided virtual machine as it is only for file submission. ..." in the previous letter,
    I would like to ask if the final uploaded results only need to be submitted as a .size file?
    Or do I need to submit other files (such as a replacement logic gate specification)?
    If it's only a .size file that needs to be submitted, i.e. the logic gate has already been replaced, then how do you test the hidden test data?
    A. Participants must provide all files required to reproduce the results in the .size files. This could be either a binary or script. The scripts/binary should generate the .size file. More details on the submission process will soon be updated in the submission section of the contest document by June 18th.

14. I would like to ask how to obtain the load capacitance, one of the competition indicators, through "update_circuitops_properties.py". By referring to the code in "OpenROAD_example.py", it seems that the load capacitance of each pin can be obtained by adding the following line (timing.getNetCap(pin.getNet(), corner, timing.Max) if pin.isOutputSignal() else "None") to "update_circuitops_properties.py", like this:

```
pin_entry = pd.DataFrame({
  "pin_name": [design.getITermName(pin)],
  "maxcap": [-1],
  "maxtran": [-1],
  "pin_tran": [timing.getPinSlew(pin)],
    "pin_slack":  [min(timing.getPinSlack(pin,  timing.Fall,  timing.Max),
timing.getPinSlack(pin, timing.Rise, timing.Max))],
  "pin_rise_arr": [timing.getPinArrival(pin, timing.Rise)],
  "pin_fall_arr": [timing.getPinArrival(pin, timing.Fall)],
     "input_pin_cap":   [timing.getPortCap(pin,   corner,   timing.Max)   if
pin.isInputSignal() else "None"],
    "output_pin_cap":  [get_output_load_pin_cap(pin,  corner,  timing)  if
pin.isOutputSignal() else "None"],
   "output_load_cap": [timing.getNetCap(pin.getNet(), corner, timing.Max)
if pin.isOutputSignal() else "None"]
})
```

However, when I analyze the load capacitance in the generated CSV file, I find that the maximum load capacitance is different from the results shown in the evaluation. I would like to ask why the load capacitance did not change accordingly and remain the same as the initial solution, or is it actually not possible to obtain the load capacitance through the method I mentioned above?



```
Initial contest evaluation:
==================================================
WNS: -0.631400 ns, worst Slew: 1.186760 ns
worst load capacitance: 0.063600 pF, total leakage power: 1.521694 uW
Score: 2.153094
==================================================
Final contest evaluation:
==================================================
WNS: -0.202570 ns, worst Slew: 0.255560 ns
worst load capacitance: 0.015110 pF, total leakage power: 2.309524 uW
Score: 2.512094
==================================================
Generating intermediate file
Finish generating intermediate file
worst load capacitance in intermediate file: 6.359753760946946e-14
>>>
```

The program I used for testing is as follows: I used the function provided by the competition to read the design, and then evaluated the initial statistical results. After that, I used the "swap_libcell" provided by the competition to swap the library cells according to the reference provided by the competition, and then evaluated the statistical results again. Finally, I called the "update_circuitops_properties.py" provided by the competition to generate the intermediate file.

```python
import os
import sys
import openroad as ord
from openroad import Tech, Design, Timing
import pandas as pd
import subprocess

def evaluation_copy(design: Design, timing: Timing):
    WNS, maxSlew, maxCap, totalLeakagePower = 0, 0, 0, 0
    # Penalties are subject to change
    WNSPenalty, maxSlewPenalty, maxCapPenalty = 1, 1, 1

    capLimit = 0

    # Get all timing metrices
    design.evalTclString("report_wns > evaluation_temp.txt")
    with open("evaluation_temp.txt", "r") as file:
        for line in file:
            WNS = float(line.split()[1]) / 1000
                design.evalTclString("report_check_types    -max_cap    >
evaluation_temp.txt")
    with open("evaluation_temp.txt", "r") as file:
        for line in file:
            if len(line.split()) != 0:
                maxCap = line
        maxCap = float(maxCap.split()[2]) / 1000
                design.evalTclString("report_check_types    -max_slew    >
evaluation_temp.txt")
    with open("evaluation_temp.txt", "r") as file:
        for line in file:
            if len(line.split()) != 0:
                maxSlew = line
                capLimit = line
        maxSlew = float(maxSlew.split()[2]) / 1000
        capLimit = float(capLimit.split()[1]) / 1000
    os.remove("evaluation_temp.txt")
    # We only have one corner in this contest
    corner = timing.getCorners()[0]
    for inst in design.getBlock().getInsts():
        totalLeakagePower += timing.staticPower(inst, corner)
    totalLeakagePower *= 1000000
    # Adjust penalties
    WNSPenalty = 0 if WNS >= 0.0 else WNSPenalty
    maxSlewPenalty = 0 if 320 >= maxSlew else maxSlewPenalty
    maxCapPenalty = 0 if capLimit >= maxCap else maxCapPenalty

    # Compute score
     score = totalLeakagePower + WNSPenalty * abs(WNS) + maxSlewPenalty *
abs(maxSlew) + maxCapPenalty * abs(maxCap)
    print("====================================================")
    print("WNS: %f ns, worst Slew: %f ns" % (WNS, maxSlew))
     print("worst load capacitance: %f pF, total leakage power: %f uW" %
(maxCap, totalLeakagePower))
    print("Score: %f" % score)
    print("====================================================")

def generate_intermediate_file(size_path):
    print("Generating intermediate file")
      command = ["../../OpenROAD/build/src/openroad", "-python", "-exit",
"../intermediate_file_generator/update_circuitops_properties.py",
"--design_name",   "NV_NVDLA_partition_m",   "--file_path",   size_path,
"--dump_csv", "--dump_path", "./"]
    result = subprocess.run(command, text=True, capture_output=True)
    print("Finish generating intermediate file")

if __name__ == "__main__":
    sys.path.append("../example/")
    from OpenROAD_helper import load_design
    ord_tech, ord_design = load_design("NV_NVDLA_partition_m", False)
    timing = Timing(ord_design)

    # Initial contest evaluation
    print("Initial contest evaluation:")
    evaluation_copy(ord_design, timing)
    # Use reference to swap libcell
    sys.path.append("../evaluation/")
    from evaluation import swap_libcell
                                                            size_path           =
"../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.size"
    swap_libcell(size_path, ord_design)

    # Final evaluation
    print("Final contest evaluation:")
    evaluation_copy(ord_design, timing)

    # Generate intermediate file
    generate_intermediate_file(size_path)
    pin_update_df = pd.read_csv('./pin_properties_update.csv')
    pin_update_df_filled = pin_update_df.fillna({'output_load_cap': -1})
            print(f"worst   load   capacitance   in   intermediate   file:
{max(pin_update_df_filled['output_load_cap'])}")
```

Thank you!

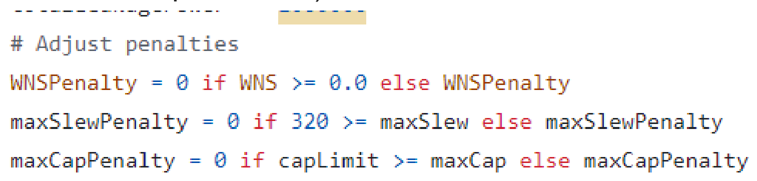
A. Your approach is correct. The getNetCap property provides the load capacitance at the output pin. It adds the sink capacitance and the wire capacitance.

You need to compare the “output_load_cap” in the intermediate file with the “maxcap” property to determine if it violates the timing constraint, same thing applies to the slew property. Please refer the updated evaluation script. The previous output message in the evaluation script showed figures that did not violate the timing constraint. The latest evaluation script, which we have modified on 6/14/2024 has fixed the displayed message.

15. Sorry to bother again. We noticed that in the provided .size file, there are some sequential gates that are resized. But in the document of problem C, it is stated that "Sequential gate sizing will not be permitted.". If we are not allowed to resize sequential gates, we think that using the .size file as label for training might slightly mislead the model. And we might not be able to achieve the same quality as the .size file under the constraint that we cannot resize sequential gates. Thus, we want to ask if it is possible for you to provide the .size file without resizing sequential gates. Thank you very much.
A. This was a mistake. Thanks for pointing this out. We have now updated file .size for the testcases.

16. The triggering conditions for penalty in the evaluation seem to be incorrect. Why are the slew limits all set to 320? The task mentions it should be determined based on the pin, but here it's a fixed value, and moreover, the units are also set incorrectly. The cap and slew limits should be divided by 1000 to get the correct value(320/1000 and caplimit/1000).

在evluation的時候penalty 的觸發條件是否有錯，為什麼slew的limit都是320題目說是根據pin決定但這裡是固定值，而且單位也設置錯了，cap和slew的limit應該要/1000才是正確值(320/1000 and caplimit/1000)



```
# Adjust penalties
WNSPenalty = 0 if WNS >= 0.0 else WNSPenalty
maxSlewPenalty = 0 if 320 >= maxSlew else maxSlewPenalty
maxCapPenalty = 0 if capLimit >= maxCap else maxCapPenalty
```

A. 320ps is the fixed value for the ASAP7 library. However, we have now fixed this to use the API-fetched values.

6/18/2024

17. I am writing to seek clarification regarding the existence of the five hidden benchmarks mentioned in Problem C. Specifically, I would like to confirm if these hidden benchmarks defined by ASAP7 indeed exist. Additionally, I have a few questions about these hidden benchmarks:
    a. Will the number of cell types increase in these hidden libraries, or will there simply be more sizes available?
    A. The hidden benchmarks use ASAP7 PDK itself. So the number of cell types will not increase. Only those that are available in the liberty file.
    b. Will the naming conventions for the hidden libraries be consistent with those of the currently provided visible libraries?
    A. Yes the naming conventions will be identical since we are using the same technology/PDK. The platform folder will remain the same.

18. I cloned the latest github of problem C and run the evaluation.py. But it cannot get the same results with the statistics on github, such as the benchmark "mempool_tile_wrap", I run the evaluation.py with command "../../OpenROAD/build/src/openroad -python evaluation.py --file_path ../../design/mempool_tile_wrap/mempool_tile_wrap.size --design_name mempool_tile_wrap". The results I get are as following:

```
Legalizing...
Run Global Routing...
====================================================
WNS: -14.519220 ns
worst Slew: 1538.320362 ns, Limit: 0.227000 ns
worst load capacitance: 29.004316 pF, Limit: 0.368640 pF
total leakage power: 2.692525 uW
Score: 1584.536423
```

please help to check with this! Thank you.

We are not able to reproduce the results on the GitHub repository. So it is challenging for us to see what error you are facing. Please ensure you have the latest OpenROAD submdule. Please ensure you have the latest GH repository. Please follow the thread in GitHub Issue of the ICCAD repository.

19. Due to the recent update that includes the Legalization, Global Route and Estimate Global Route RC sections, I would like to ask if the routing will be affected by different gate sizing each time. Will the routing time vary depending on the gate replacements? Will there be instances where routing cannot be completed or where gates cannot be placed properly, leading to a deadlock issue? If routing or gate placement cannot be completed successfully, is it possible to terminate and return an unsuccessful completion message?
    Based on our analysis, we sized the visible benchmarks to be the largest possible size for all instances. Three of the four visible testcases pass legalization in OpenROAD. The only one that fails is NV_NVDLA_partition_m which says [ERROR DPL-0036] Detailed placement failed. For the global routing stage, we will set a two-hour limit for the evaluation. If global routing takes more than two hours due to high overflow, we will consider it a failed case.

    Since this scenario is very unlikely to happen in both the visible testcases and hidden testcases due to the low utilizations in OpenROAD, we will not be considering this in the optimization function and considering that a failed legalization as an invalid solution and that testcase will not be counted. Our scoring system will give 100 points to the team with the best solution (least score as defined in the evaluation section of the document) and all other teams will be graded relative to that out of 100. So for the benchmark that fails then it would result in a 0/100.

    The routing is global routing and the testcases have very low utilization. So the hope is that legalization is fully possible. In case routing does not complete, OpenROAD will throw an error saying unsuccessful completion.

20. We have a question regarding the consideration of “logical equivalence.” Is logical equivalence sufficient when the functions are the same? We found that "O2A1O1Ixp33_ASAP7_75t_R" and "O2A1O1Ixp5_ASAP7_75t_R" are considered logically different by the contest evaluation script. This discrepancy appears to be caused by the differences in the number of timing arcs and schematics between the two gates. (You can follow up this using the provided links -> link1, link2) What is meant by “logical equivalence” in the context of this contest? Do you think the evaluation script needs to be modified?
    A. Thanks for pointing this out. This is super helpful. Since this is a bug in OpenROAD and we are uncertain when it will be fixed, we have created a workaround by using a

modified version of OpenSTA submodule. Please see the pull request on the contest GitHub. **Please use this submodule of OpenSTA only. Note that sram cells and sequential cells cannot be sized.**

6/20/2024

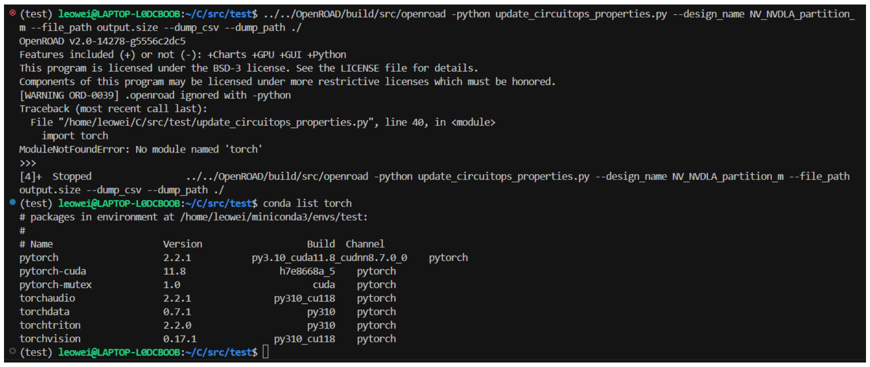
21. I am a participant in the ICCAD Problem C, and I have a few questions regarding the benchmark. I would like to confirm if the hidden benchmarks mentioned in Problem C refer to five additional hidden Verilog files or to five ASAP7 libraries. Additionally, when submitting our code, are we allowed to call the organizers' evaluation function and obtain the results?
The hidden testcases use the same libraries as the visible testcases. So the only difference is that the hidden testcases are different designs. The platform folder remains the same. Yes you may call the evaluation function. Please note the change in the evaluation function on June 21 in GH.

22. We have a question regarding the new description of the contest. We would like to know how the scores are calculated if it fails to legalize using the .size file provided by our framework. Is there any penalties when the .size solution is not legalizable?
Based on our analysis, we sized the visible benchmarks to be the largest possible size for all instances. Three of the four visible testcases pass legalization in OpenROAD. The only one that fails is NV_NVDLA_partition_m which says [ERROR DPL-0036] Detailed placement failed. For the global routing stage, we will set a two-hour limit for the evaluation. If global routing takes more than two hours due to high overflow, we will consider it a failed case.

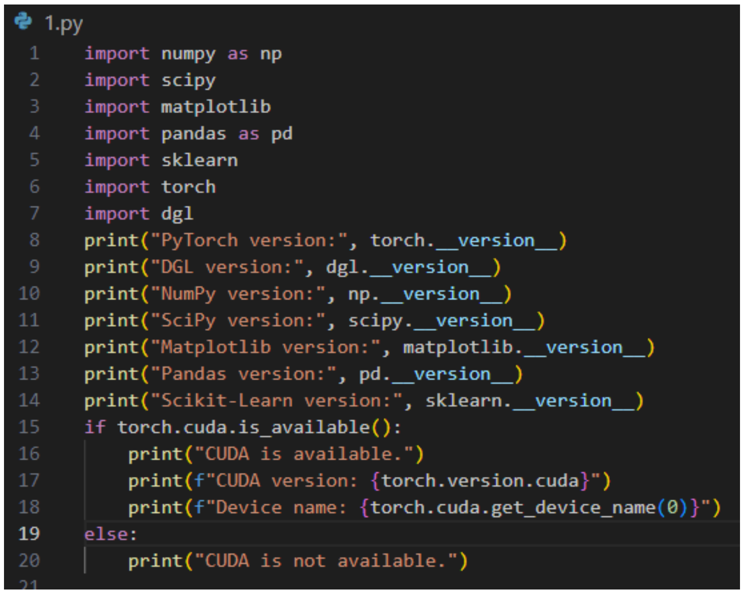
    Since this scenario is very unlikely to happen in both the visible testcases and hidden testcases due to the low utilizations in OpenROAD, we will not be considering this in the optimization function and considering that a failed legalization as an invalid solution and that testcase will not be counted. Our scoring system will give 100 points to the team with the best solution (lowest score as defined in the evaluation section of the document), and all other teams will be graded relative to that out of 100. So for the benchmark that fails, then it would result in a 0/100.

6/23/2024

23. In the picture below, I import torch in my python code, but the openroad raise an error as ModuleNotFoundError.However,I run another code to check the version of my packages,which can be execute successfully.So I wonder if it's Openroad's problem. please tell me about the version of your pytorch,dgl or any other packages in your environment.thank you

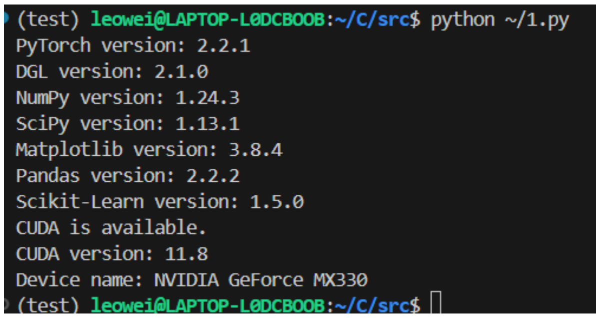


```
(test) leowei@LAPTOP-L0DCBOOB:~/C/src/test$ ../../OpenROAD/build/src/openroad -python update_circuitops_properties.py --design_name NV_NVDLA_partition_
m --file_path output.size --dump_csv --dump_path ./
OpenROAD v2.0-14278-g5556c2dc5
Features included (+) or not (-): +Charts +GPU +GUI +Python
This program is licensed under the BSD-3 license. See the LICENSE file for details.
Components of this program may be licensed under more restrictive licenses which must be honored.
[WARNING ORD-0039] .openroad ignored with -python
Traceback (most recent call last):
  File "/home/leowei/C/src/test/update_circuitops_properties.py", line 40, in <module>
    import torch
ModuleNotFoundError: No module named 'torch'
>>>
[4]+  Stopped                 ../../OpenROAD/build/src/openroad -python update_circuitops_properties.py --design_name NV_NVDLA_partition_m --file_path
output.size --dump_csv --dump_path ./
(test) leowei@LAPTOP-L0DCBOOB:~/C/src/test$ conda list torch
# packages in environment at /home/leowei/miniconda3/envs/test:
#
# Name                    Version                   Build  Channel
pytorch                   2.2.1           py3.10_cuda11.8_cudnn8.7.0_0    pytorch
pytorch-cuda              11.8                 h7e8668a_5    pytorch
pytorch-mutex             1.0                        cuda    pytorch
torchaudio                2.2.1               py310_cu118    pytorch
torchdata                 0.7.1                   py310    pytorch
torchtriton               2.2.0                   py310    pytorch
torchvision               0.17.1              py310_cu118    pytorch
(test) leowei@LAPTOP-L0DCBOOB:~/C/src/test$
```

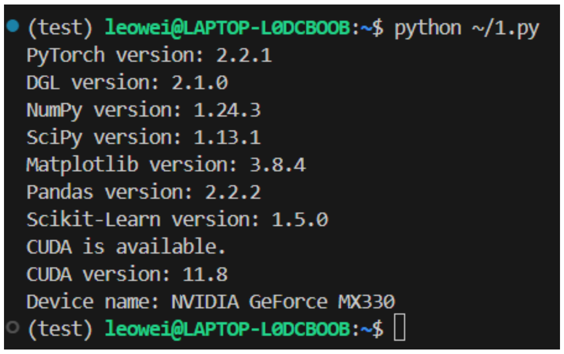


1.py

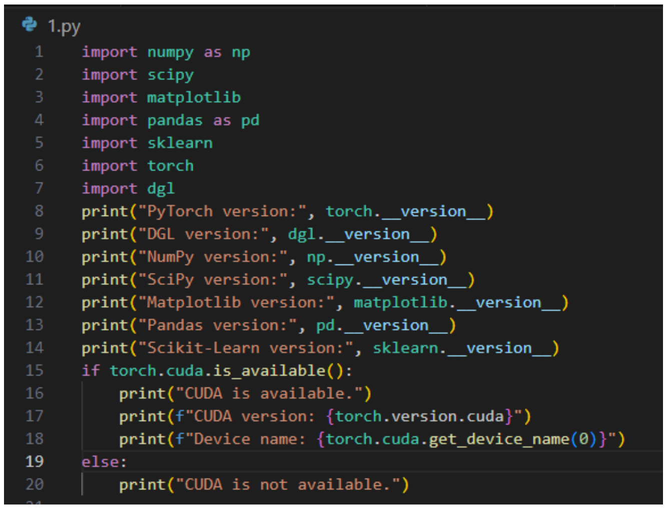
```python
import numpy as np
import scipy
import matplotlib
import pandas as pd
import sklearn
import torch
import dgl
print("PyTorch version:", torch.__version__)
print("DGL version:", dgl.__version__)
print("NumPy version:", np.__version__)
print("SciPy version:", scipy.__version__)
print("Matplotlib version:", matplotlib.__version__)
print("Pandas version:", pd.__version__)
print("Scikit-Learn version:", sklearn.__version__)
if torch.cuda.is_available():
    print("CUDA is available.")
    print(f"CUDA version: {torch.version.cuda}")
    print(f"Device name: {torch.cuda.get_device_name(0)}")
else:
    print("CUDA is not available.")
```



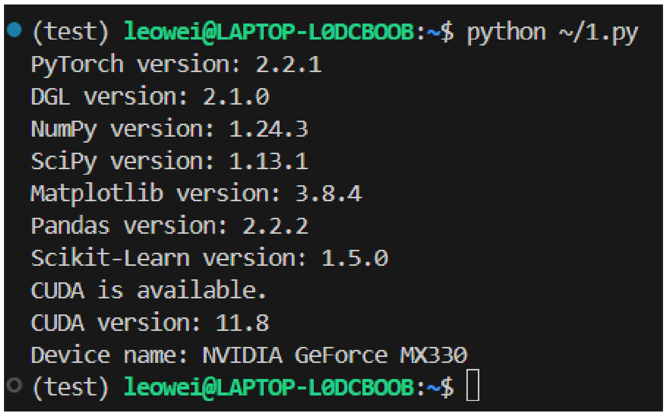
```
(test) leowei@LAPTOP-L0DCBOOB:~/C/src$ python ~/1.py
PyTorch version: 2.2.1
DGL version: 2.1.0
NumPy version: 1.24.3
SciPy version: 1.13.1
Matplotlib version: 3.8.4
Pandas version: 2.2.2
Scikit-Learn version: 1.5.0
CUDA is available.
CUDA version: 11.8
Device name: NVIDIA GeForce MX330
(test) leowei@LAPTOP-L0DCBOOB:~/C/src$
```



```
(test) leowei@LAPTOP-L0DCBOOB:~$ python ~/1.py
PyTorch version: 2.2.1
DGL version: 2.1.0
NumPy version: 1.24.3
SciPy version: 1.13.1
Matplotlib version: 3.8.4
Pandas version: 2.2.2
Scikit-Learn version: 1.5.0
CUDA is available.
CUDA version: 11.8
Device name: NVIDIA GeForce MX330
(test) leowei@LAPTOP-L0DCBOOB:~$
```



```
1.py
import numpy as np
import scipy
import matplotlib
import pandas as pd
import sklearn
import torch
import dgl
print("PyTorch version:", torch.__version__)
print("DGL version:", dgl.__version__)
print("NumPy version:", np.__version__)
print("SciPy version:", scipy.__version__)
print("Matplotlib version:", matplotlib.__version__)
print("Pandas version:", pd.__version__)
print("Scikit-Learn version:", sklearn.__version__)
if torch.cuda.is_available():
    print("CUDA is available.")
    print(f"CUDA version: {torch.version.cuda}")
    print(f"Device name: {torch.cuda.get_device_name(0)}")
else:
    print("CUDA is not available.")
```



```
(test) leowei@LAPTOP-L0DCBOOB:~$ python ~/1.py
PyTorch version: 2.2.1
DGL version: 2.1.0
NumPy version: 1.24.3
SciPy version: 1.13.1
Matplotlib version: 3.8.4
Pandas version: 2.2.2
Scikit-Learn version: 1.5.0
CUDA is available.
CUDA version: 11.8
Device name: NVIDIA GeForce MX330
(test) leowei@LAPTOP-L0DCBOOB:~$
```

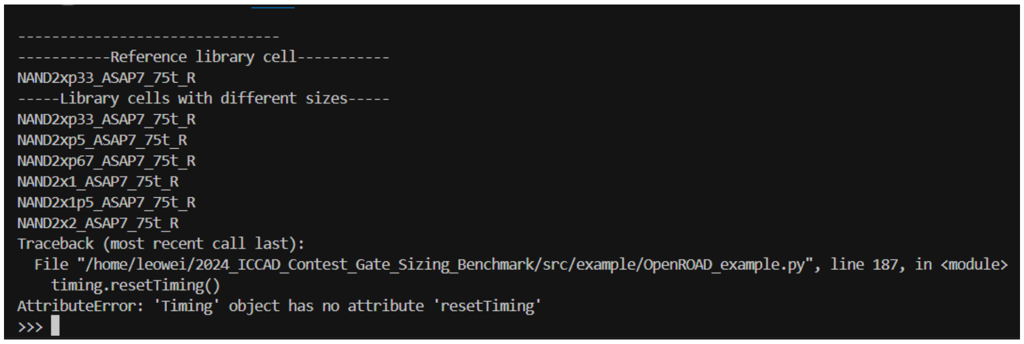
Please make sure the package is installed under the same Python path that OpenROAD is using.

24. Regarding the latest update mentioned in the email about the Timing::resetTiming() API, I would like to ask how to build STA network? Second, if building the STA network and calling the swapMaster() API is equivalent? Additionally, I would like to inquire if there is a time limit for running the contest code. If so, what is the maximum allowed time?
For the first and second questions, the STA network is built when OpenROAD calls the underlying C++ getSta() function. Since most Timing APIs call the getSta() function under the hood, it's better to call Timing::resetTiming() every time you perform swapMaster and have called Timing APIs before. For the evaluation, we will allow a 2-hour limit for performing global routing, and for running the gate sizing program, we will determine this after the alpha submission.

25. I wanna know if I use the OpenROAD in my code, am I going to submit the whole OpenRoad directory to the sever or maybe there are other way to do it? Besides , you mention that "More details on submission process and commands to run will be provided by June 20, 2024." but I don't see it. Could you tell me where to find it ?
Your submission should contain everything we need to run your code and evaluate the testcases easily. We need everything that is needed to execute the command listed in

the Submission Process part of the document. We have now updated the submission process.

26. In the latest updated testbench, the new API resetTiming() can't work

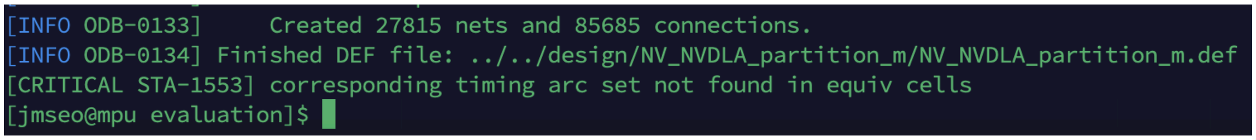


```
------------------------------
----------Reference library cell----------
NAND2xp33_ASAP7_75t_R
-----Library cells with different sizes-----
NAND2xp33_ASAP7_75t_R
NAND2xp5_ASAP7_75t_R
NAND2xp67_ASAP7_75t_R
NAND2x1_ASAP7_75t_R
NAND2x1p5_ASAP7_75t_R
NAND2x2_ASAP7_75t_R
Traceback (most recent call last):
  File "/home/leowei/2024_ICCAD_Contest_Gate_Sizing_Benchmark/src/example/OpenROAD_example.py", line 187, in <module>
    timing.resetTiming()
AttributeError: 'Timing' object has no attribute 'resetTiming'
>>>
```

Please run "git pull --recurse-submodules" or reclone the GitHub repo and remake the openroad binary again.

6/27/2024

27. For the newly committed evaluation script, I can see the changed error log like below.



```
[INFO ODB-0133]     Created 27815 nets and 85685 connections.
[INFO ODB-0134] Finished DEF file: ../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.def
[CRITICAL STA-1553] corresponding timing arc set not found in equiv cells
[jmseo@mpu evaluation]$
```

Is this the desired operation for the contest?

I just tested with the NV_NVDLA_partition.size file slightly modified. (changed line 13831 from “g43328__4319 O2A1O1Ixp33_ASAP7_75t_R” to “g43328__4319 O2A1O1Ixp5_ASAP7_75t_R”)

We have added the timing.resetTiming() api in the sample evaluation script to fixed the issue.

28. I encountered the error "[CRITICAL STA-1553] corresponding timing arc set not found in equiv cells" when attempting to swap cells, and I'm unsure of the cause. Could you please explain the circumstances under which this error occurs?
Please execute timing.resetTiming() before the sizing operation if the STA network is built.

7/11/2024

29. Hi, I have a few questions about order in the python api.
    a. Does equiv_cells get from timing.equivCells(inst_master) sort in power(smallest to largest)?

b. Does pins and insts sort in topological order? If they are not in the sorted order, is there any flag which can force it to sort in these order?

a. No, they are not sorted using power.
b. No, they are not sorted in any order, please refer to : https://github.com/The-OpenROAD-Project/OpenROAD/blob/f8f2b729ce1f72d1160050149d589f830db16dbb/src/odb/include/odb/dbSet.h#L94

30. Here are some questions we had from the team:
    a. Does the size file need to have every gate in the circuit or only the ones that changed?
    b. Do you have an example script to show how to load the design using tcl commands only? We keep getting different values after running an evaluation on the benchmarks.
    c. Do any of the designs being tested have more than one source clock?
    d. Is there a possibility that there are false paths not defined by the .sdc files or are all the false paths already set? Can our algorithm work assuming every false path has been declared/defined in the constraints?

    a. The final .size file only needs to have the gates with updated sizes. So no, the .size file does not need to have all gates in the circuit.
    b. We have included a tcl proc for loading design in tcl. Please refer to "https://github.com/ASU-VDA-Lab/2024_ICCAD_Contest_Gate_Sizing_Benchmark/blob/main/src/example/OpenROAD_helper.tcl"
    c. One of the hidden cases has two clock signals.
    d. All false paths are defined in the sdc file.

31. I would like to ask if the interchangeable library cells for the same library cell can be different in different designs. Since the timing object is created from the design object, and we use the timing object's function timing.equivCells(master) to check which library cells are interchangeable, I would like to know if this has any impact.

    Please refer to this link, the function timing.equivCells(master) will return the equivalent cells from the library and not the design only. So all the interchangeable library cells will be the same across all designs.

32. which version of CUDA is used on the test machine? The 2.2.0 Pytorch can only support 11.8 or 12.1 CUDA. Does the version of torch must be 2.2.0 in the docker image? Can we change the Dockerfile to install other version of torch?

    Here's the configuration of the evaluation server:
    CUDA version:12.2, Driver version: 535.161.07

The command “RUN pip install --no-cache-dir torch==2.2.0” in the provided Dockerfile will install the “torch==2.2.0+cu121”: version, which includes cuda 12.1 runtime dependencies. We state, 'Teams are required to build a Docker image on top of the provided Dockerfile,' to ensure that all submitted docker images can run correctly on our systems. If you modify the settings, you must be prepared to accept responsibility for any issues that prevent your submission from running correctly during the beta and evaluation stages.

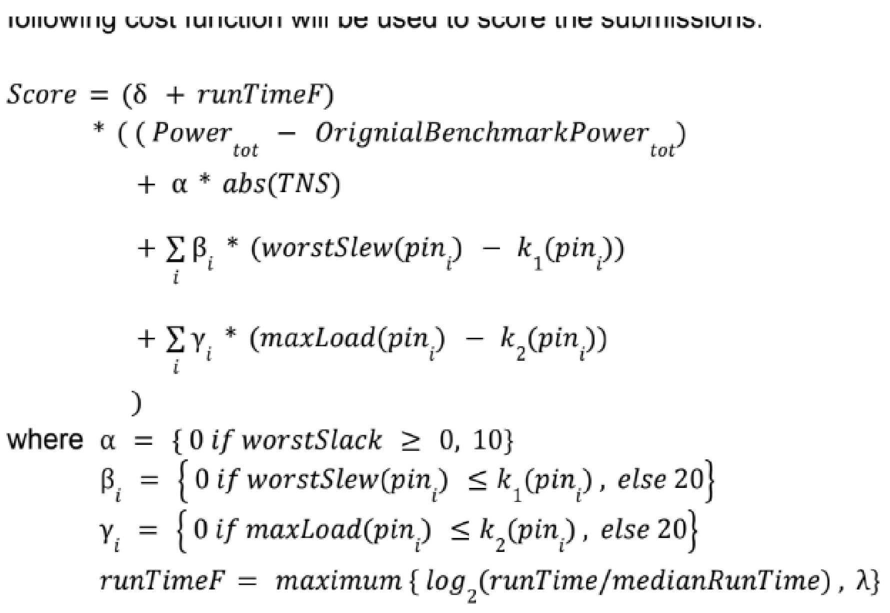
7/19/2024

33. I found that there exists circuit loop in the benchmark. For example, in "ariane136", pin "g1405522/B1" is connected with pin "g1404610/Y" and pin "g1404610/A" is connected with pin "g1405522/Y". Please help to check with this! Thank you.

    I am not sure I fully understand the question. But OpenSTA can time this logic.
    It does not affect the logic, and OpenSTA will handle breaking the combinational feedback loop, so the number you are getting will not be affected.

7/25/2024

34.



following cost function will be used to score the submissions.

$$Score = (\delta + runTimeF) * ((Power_{tot} - OrignialBenchmarkPower_{tot}) + \alpha * abs(TNS) + \sum_i \beta_i * (worstSlew(pin_i) - k_1(pin_i)) + \sum_i \gamma_i * (maxLoad(pin_i) - k_2(pin_i)))$$

where $\alpha = \{0\ if\ worstSlack \geq 0,\ 10\}$

$$\beta_i = \{0\ if\ worstSlew(pin_i) \leq k_1(pin_i),\ else\ 20\}$$

$$\gamma_i = \{0\ if\ maxLoad(pin_i) \leq k_2(pin_i),\ else\ 20\}$$

$$runTimeF = maximum\{log_2(runTime/medianRunTime),\ \lambda\}$$

What does "Runtime" mean? Is it the time that we generate the .size file or we evaluate the score from the .size file?

The runtime is the time it takes for your sizer to generate the .size files.

35. We would like to ask about the result of our alpha test, since we failed to generate results in some cases. The description below shows "Time out for NV_NVDLA_partition_p, ariane136, mempool_tile_wrap and other hidden designs", but in the Q&A No.24 it is mentioned that "for running the gate sizing program, we will determine this(runtime limit) after the alpha submission." Thus we expected there is no runtime limit on the alpha test. Would you please check if our program reports errors other than exceeding the time limit? And we also want to ask if it is possible to test our program on the machine that is used for testing before beta submission. Thank you!

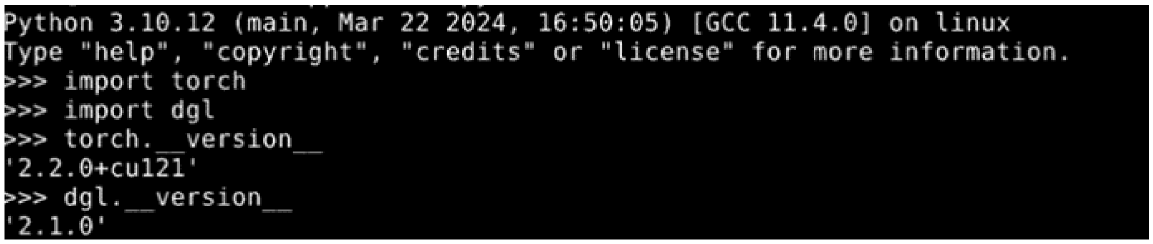
    Yes we did mention we would have no runtime constraint. However, we were bound by compute resources and time to grade all evaluation. For us to finish all the submission and return them back in time required us to stop evaluation after 7200s. There is no log to trace, so it's just a timeout. No, the evaluation is run on the internal Nvidia server, and we cannot grant access to the participants..

36. I would like to ask will the performance on the visible benchmark be included in the ranking, or will only the performance on the hidden benchmark be considered?

    Both visible and hidden cases will be considered in ranking. Since we use both hidden and visible for evaluation, we do not permit reading the reference .size file as input. The reference .size file is only provided for those who wish to train supervised learning models. So the .size file cannot be used as input as a starting point solution. When we do our beta evaluations, we will check for this.

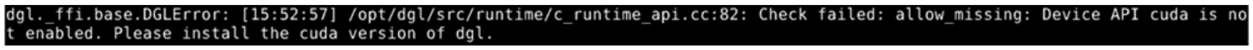
7/30/2024

37. We are a participant of Problem C and we have a question regarding the required packages installation. I have installed the provided version of torch (2.2.0+cu121) and dgl (2.1.0).



```
Python 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import dgl
>>> torch.__version__
'2.2.0+cu121'
>>> dgl.__version__
'2.1.0'
```

    However when I tried to put the DGL graph onto GPU, I got the following error.



```
dgl._ffi.base.DGLError: [15:52:57] /opt/dgl/src/runtime/c_runtime_api.cc:82: Check failed: allow_missing: Device API cuda is no
t enabled. Please install the cuda version of dgl.
```

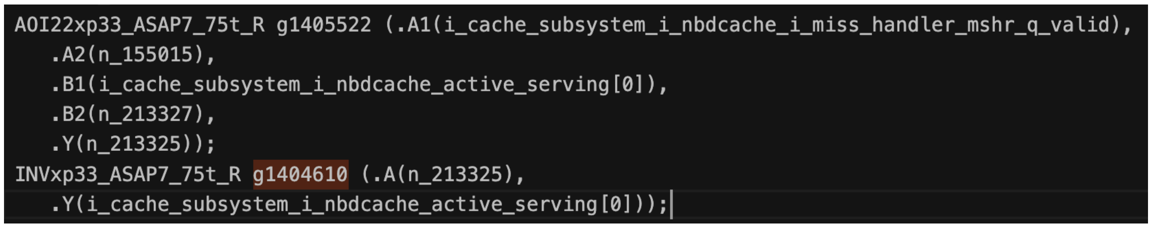
    I would like to ask what is the command for getting the cuda version of dgl 2.1.0. Thank you very much.

    We have added another dockerfile for dgl-cuda support, please refer to this file.

38. I would like to inquire about the CPU configuration in the competition. Is the competition environment providing 8 CPUs, and does each CPU have 8 cores for multiprocessing?

We are providing 8 cpu threads (4 cpu cores and 2 threads per core) for the contest.

39. Hi, I have a question about benchmark ariane136. Since we develop our own timing engine, we found there is a structure that we don't know how to deal with. The structure is as follow:



```
AOI22xp33_ASAP7_75t_R g1405522 (.A1(i_cache_subsystem_i_nbdcache_i_miss_handler_mshr_q_valid),
   .A2(n_155015),
   .B1(i_cache_subsystem_i_nbdcache_active_serving[0]),
   .B2(n_213327),
   .Y(n_213325));
INVxp33_ASAP7_75t_R g1404610 (.A(n_213325),
   .Y(i_cache_subsystem_i_nbdcache_active_serving[0]));
```

But it seems that openroad can still calculate tns, I wonder how openroad deal with this case.

OpenSTA will disable the edge closing the loop to break the combinational feedback loop. Please refer to the following link: https://github.com/The-OpenROAD-Project/OpenSTA/blob/9a490b1b13187f6e19822bfb1d409b2c9d9d1032/search/Levelize.cc#L207

8/1/2024

40. there will be 4 A100 GPUs on the test machine as you mentioned, are the additional codes in my sizer is needed to use these 4 GPUs?

You are welcome to include methods in your sizer, such as machine learning techniques and GPU acceleration (cuda programming), to fully utilize the GPU resources for the contest.

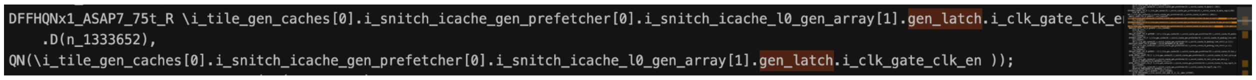
8/8/2024

41. I was rereading the file that introduced the problem, and I got a question. There is "There will be no clock buffers in the circuit (zero skew)" in document. I am curious if this can limit the resizing of cells with the same func_id, especially for cells with func_id 34 and 35. As an example, I am asking whether it is possible to change an HB type cell to a BUF or an INV type cell to a CKINV, specifically for cells with func_id 34 and 35. The reverse case should also be considered. In both cases, I believe they have the same func_id, but their roles in the circuit might be slightly different. Please help me resolve my question.

OpenSTA will evaluate the functionality of cells to grant the sizing operation. Since we are using OpenSTA for evaluation, we will leave it to OpenSTA to permit the sizing operation. Please refer to the code to understand how OpenSTA grants the gate sizing operation: https://github.com/bingyuew/OpenSTA/blob/a037f46d5062a65837411359705f8225186ee4b3/liberty/EquivCells.cc#L258

8/14/2024

42. We found that some port in mepool_tile_wrap.v is not in correct form. According to following picture, some QN should be .QN.



```
DFFHQNx1_ASAP7_75t_R \i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[0].i_snitch_icache_l0_gen_array[1].gen_latch.i_clk_gate_clk_e
    .D(n_1333652),
QN(\i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[0].i_snitch_icache_l0_gen_array[1].gen_latch.i_clk_gate_clk_en ));
```

Thanks for the notice. It has been fixed.

43. Since the public cases will be scored, and you provide reference results for these cases. I have a question that how can you prevent the competitors using some "hard code" to output the reference results for these cases quickly by something like case name recognition or other methods.

We have noticed it in the alpha stage, thus, in the beta and final stages, the reference .size files are removed in the evaluation process. If the team has used hard codes within their program based on testcase we will be sure to check this for all teams. For example, we can change the name of the testcase e.g., NVDLA_partition_m to arbitrarty_design_name to prevent case name-based recognition. We are also ensuring that the runtimes between hidden benchmarks and viable benchmarks are not *significantly* different. If the results of the teams are extremely good on the visible, but there is a large discrepancy with hidden benchmarks, we will take a closer look at the teams' approach in that case. Please avoid hard codes and such approaches and let us work towards better algorithms!

8/16/2024

44. We found that in all cases, the cell name in the .def file is not the same as .v file, and in mempool_tile_wrap case there is some cell in .v that is not exist in .def. Since we read the verilog as our input file, we need a correct .v file.

The format of the DEF and the netlist file is correct. We have tested it by reading both into the commercial tool, and the tool can match all cells in the DEF and netlist files. Any format discrepancies, such as the presence of '[' in the netlist file and '\[' in the DEF file, should be handled by the contestant. Since we are using the DEF file in the evaluation process, the cell name in the .size file should be the same as in the DEF file. Perhaps you can disregard the '\' character when reading the cell name from the netlist and add it in front of the bracket characters when dumping the .size file?

We have also double checked the files for the mempool_tile_wrap design, and all cells can be found in both the DEF and the netlist files. If you can point us exactly to the names of the cells that will make the question better and more concrete.

8/19/2024

45. I would like to clarify whether the 2-hour limit means that if the program doesn't terminate within 2 hours, it will be considered that we failed to generate the .size file. In our tests, the .size file is always created within 2 hours, but the program may not fully complete its

run within that time. In this case, would it still be considered as not having generated the .size file? Also, should the .size file be placed in the /app/sizer directory? Thank you for your clarification.

For generating the .size file: If the .size file can be generated within the 2-hour limit, then your .size file will be put into evaluation, regardless of whether the program is killed due to the runtime limit or not. If the program is terminated due to the runtime limit, then the runtime for your sizer is considered as 7200 seconds.

For evaluating the .size file: Even if you successfully generate a .size file, you must still manage to complete the evaluation process within the 2-hour limit. If your .size file fails to pass the legalization process or fails to complete the global routing stage due to high overflow, then it is still considered a failed case.

The .size file should be put under the /app/sizer/output directory.

46. I would like to ask why my runtimes are all 7200 seconds. When I tested it myself, I could almost always get the answers within 5 minutes. The difference is quite significant, so I'm a bit confused.

    You did not exit the OpenROAD Python shell after generating the .size file. As a result, the program is still running. Please add these lines in your program:
    ```
    >>> from openroad import Design, Tech
    >>> design = Design(Tech())
    >>> design.evalTclString("exit")
    ```
    We have reevaluated your solution and we are updating the beta evaluation results with the exit added. Please remember to add it on your own for the final submission.

47. Thanks for evaluating the beta submission. But the results show that the execution of case "hidden4" got "segment fault". We would like to know what caused this. Could you please provide us with more information about this run, e.g., error logs or the logs print by our algorithm, such that we can debug based on it and avoid these issues later. And for the case "hidden3", result shows that it failed to complete evaluation within the 2-hour limit, but we cannot estimate this situation before the final evaluation, maybe you can provide some public API for us to only evaluate whether our results can get the global routing within 2 hours for the hidden case. Otherwise, it seems like a black box and is unfair. BTW, we still don't know how to evaluate the 100 points for each cases.

    1) We are now sending your submission back to Nvidia's server to display everything on the screen for you.
    2) We are using the script provided here: [2024_ICCAD_Contest_Gate_Sizing_Benchmark/src/evaluation/evaluation.py at main · ASU-VDA-Lab/2024_ICCAD_Contest_Gate_Sizing_Benchmark (github.com)]() for evaluation. The Nvidia server has 8 A100 GPUs and 64 AMD EPYC 7742 CPU cores.

However, for the evaluation, we will restrict usage to only 4 GPUs, 8 CPU threads, and up to 200GB of RAM. Since the evaluation script is provided, it should not be considered a black box.
3) We will score each team using a relative approach on a per-testcase basis. We will use the following equation to compute the final score for each testcase:
$100 \times \frac{\textit{Best least score across all team}}{\textit{contestant's score}}$. If a team successfully generates the .size file but fails to pass the evaluation process within the 2-hour limit due to high overflow, the team will not be given points. It will be considered an invalid solution. We understand your challenge but however, there is no use of finding a solution that cannot be routed. Hence legalization and routing have been added as a part of the evaluation. We cannot give partial points at this time as its too late in the contest and it would be unfair to teams who have actually generated a solution that can go through global routing.

48. Thank you very much for evaluating our beta submission. We have some questions concerning the errors that occurred during the evaluation of the hidden cases. The provided libcell property IR tables ('libcell_properties.csv' files) are identical across the five visible cases, so we had assumed that these tables contained information for all 196 libcells in the library and would remain consistent for the hidden cases as well. However, based on the errors encountered during our beta test, we suspect that there may be fewer libcells in the hidden case libcell property tables (most likely XOR or XNOR libcells). We wonder if these tables only display libcell choices for a specific design. Could you please verify if there are any differences in the 'libcell_properties.csv' files for the hidden cases, particularly for hidden1, hidden2, hidden3, and hidden4 (where our code encountered errors)? Specifically, are there fewer or more libcells, and are the func_ids numbered differently compared to the visible cases? If differences do exist, would it be possible to make the hidden libcell property tables visible to all contestants or to release a comprehensive libcell property table with all libcells and fixed func_ids for us to use as an input file? This would greatly assist us in correcting our code and would likely benefit other contestants facing similar issues. Alternatively, would it be acceptable for us to create our own libcell property table and include it in our submitted "sizer" directory as an input file? We appreciate your assistance and look forward to your reply on this matter.

All public and hidden cases have 196 library cells in the libcell_properties.csv table. The IDs might be different, but the categories are correct. We do not encourage you to hard code this in your program, however, we have unified the libcell_properties table across both the public and hidden cases and resubmitted it to Nvidia's server to rerun the scores. It will be sent to you within a week.

49. We have observed a significant discrepancy between our local measurements and the beta submission runtime. Our beta submission was tested on a DGX machine equipped with 4 A100-SXM4-80GB GPUs. However, the runtime on the beta submission environment was approximately 5 times slower than our local measurements. We have attached a table below highlighting the runtime differences for several benchmarks:

Benchmark	Beta Submission Runtime (seconds)	DGX Measurement Runtime (seconds)
NV_NVDLA_partition_m	121.26	22.3
NV_NVDLA_partition_p	593.94	191.47
ariane136	1002.2	339.3
mempool_tile_wrap	798.73	283.5
aes_256	445.34	177.72

We kindly request your assistance in understanding the reasons behind this substantial performance gap. Could there be any configuration differences or environmental factors in the beta submission environment that might be contributing to this issue? Any insights or suggestions you can provide would be greatly appreciated. Thank you for your time and consideration. We look forward to your response.

We have sent your submission back to Nvidia's server to rerun it, the result should be sent back to you within a week.
We are using the following Docker command to run the container:
docker run -d -w /app -it --cpus=4 --gpus="device=0,1,2,3" -m 200g --name cadcxxxx_con cadcxxxx.
We are limiting to 4GPUs and 8 CPU threads due to computation challenges.

50. I hope this email finds you well. I am writing to report an issue we encountered during the beta testing phase of the contest. Specifically, we have observed significant anomalies in our results for the hidden_3 test case. To better understand and resolve the issue, we kindly request access to the corresponding log file for hidden_3. This will greatly assist us in diagnosing the problem and ensuring that our submission meets the contest's standards. Thank you for your attention to this matter. We appreciate your support and look forward to your response.

    We are now sending your submission back to Nvidia's server to display everything on the screen for you. The result will be sent to you within a week.

8/21/2024

51. We have a question regarding the submission process: Besides sizer.py, can other Python script files or .dll files that sizer.py needs to use be stored in /app/sizer within the Docker image?

    Yes, all sizer-required files can be stored under the /app/sizer directory. Please do not put any sizer-related files under the benchmark directory, as we have to re-clone the benchmark directory during the evaluation to prevent contestants from modifying the public benchmark and the OpenROAD module used for evaluation.

8/22/2024

52. Hello, there is an open issue on GitHub as regards the evaluation results output of the reference sizing provided by the contest. I am trying to include the evaluation script in my sizing code, and I feel I need to ask if it's okay to use the same evaluation code even if it doesn't give out the correct result it is meant to give for the reference sizes.

    You are free to use the evaluation code in your sizing code. But we will evaluate using the script provided in the GitHub repository. Please do not read .size files into your sizing solution as these will not exist for the hidden testcases.

53. Is it possible to extend the cut off time of the evaluation for the hidden case to 4 hours even longer? Or taking some measures to shorten the runtime of the global routing part of the evaluation script? Otherwise, we cannot estimate the run time for evaluating the results of hidden cases, and control the run time for evaluating with 2 hours by our algorithms is much more impossible. And the main objective of this contest should be the "gate sizing" instead of the routing, I believe that by doing these we can concentrate more on the main objective instead of dealing with the minor and uncertain points.

    The execution time for running your sizer and performing the evaluation are separate. You get two hours for running your gate sizer and another two hours for running the evaluation. Since there are teams that can successfully perform the evaluation within the two-hour limit, it does not make sense if your gate sizer runs quickly but your solution is impossible to route.

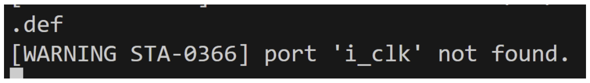
8/25/2024

54. We noticed that the overall score calculation was modified. Each case’s score totals the overall score. So we're wondering how you handle those failed cases, including routing time excess or something else.

    The overall score calculation method has remained the same since the alpha submission. We will use the following equation to compute the final score for each testcase: $100 \times \frac{Best\ least\ score\ across\ all\ team}{contestant's\ score}$. No points will be given for any failed cases, even if the .size file is successfully generated but global routing fails to complete within 2 hours. In the process of calculating the median runtime, we will not take the runtime of failed cases into account.

8/27/2024

55. Hello, I just want to know if this warning could be an issue for my sizing contest.



```
.def
[WARNING STA-0366] port 'i_clk' not found.
```

It says clock not found. This can or may be a serious warning and can impact the timing. We cannot debug this for you. Please look into the warning. Contestants must solve their bugs on their own.

56. Thank you for your response, I however want to point your attention to the fact that the evaluation code does not give the same output as what is given on the contest github page for the benchmark sizes in the design folder.

    We are unable to reproduce the issue on our end or on the evaluation platform. We can guarantee that the timing number is correct on the evaluation platform.

57. Hello , I am trying to run the evaluation script on sizing output but it was indicating that an instance was missing which was actually present in the .size file

    Without an error and a testcase we cannot debug the evaluation script. FYI: We don't allow gate sizing on flip-flops and macros. We cannot help you fix bugs in your sizers. Please refrain from asking us to look into your code.

58. I was trying to find some details of the cells in the libcell_id.csv, and I realized that it was deleted can I use the info from any of the .lib files?

    It's the same as the 'func_id' in the libcell properties table. The libcell_id.csv was deprecated and now removed.

59. We want to know which hidden cases are the same in both the alpha and beta tests. Also, will the final test use the same hidden cases as in the beta test? Many thanks.

    Hidden1-4 are the same in both the alpha and beta evaluations. For beta we added hidden5 which was not in alpha. The final will also use the same testcases. The testcases will not change.

60. Can you provide the median runtime for each case in the beta test?
    Feel free to back-calaculate it based on your score. All the information to back-calculate your score is available to you. Note that the mediate runtimes are not representative as we see drastic changes between alpha and beta submission and drastic changes across teams as well.

61. Thank you very much for your reply! May I ask if you have the updated beta submission result statistics of team cadcxxxx?

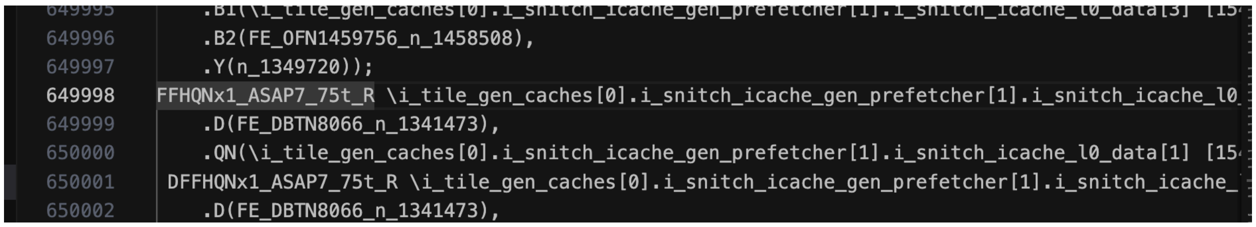
It will take another week to rerun the beta evaluation on the evaluation platform. We have rerun the evaluation on the ASU server, and the result remains the same except for the runtime. So your beta evaluation numbers are not going to be updated.

9/2/2024

62. Hello, I am a participant in Problem C and our team number is cadcxxxx. I have a few questions regarding the final submission that I would like to ask. Regarding Problem C, we are required to compress our Docker file into a file.tar.gz. During the Beta submission, we noticed that the compressed file size was 7GB. Is this normal? Additionally, could you provide the commands you use to decompress and run the Docker file? This will allow us to test our compressed file in our environment to ensure it works correctly and runs smoothly. Thank you for your assistance.

1) Please refer to the link (https://docs.docker.com/engine/reference/commandline/save/) in the Submission Process section of this document for procedures to compress your docker image. And yes, 7GB is normal.
2) We will not decompress the docker image. We will use “docker load < cadcxxxx.tar.gz” to load your docker image and use “docker run -d -w /app -it --cpus=4 --gpus="device=0,1,2,3" -m 200g --name cadcxxxx_con cadcxxxx” to run the docker container. Please refer to QA NO.49 for details.

63. We found that in your test case mempool_tile_wrap, there is a missing D in line 649998.



```
649995        .B1(\i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[1].i_snitch_icache_l0_data[3] [15
649996        .B2(FE_OFN1459756_n_1458508),
649997        .Y(n_1349720));
649998    FFHQNx1_ASAP7_75t_R \i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[1].i_snitch_icache_l0_
649999        .D(FE_DBTN8066_n_1341473),
650000        .QN(\i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[1].i_snitch_icache_l0_data[1] [15
650001     DFFHQNx1_ASAP7_75t_R \i_tile_gen_caches[0].i_snitch_icache_gen_prefetcher[1].i_snitch_icache_
650002        .D(FE_DBTN8066_n_1341473),
```

Sorry for accidentally deleting it. It has been fixed.

64. Could you help to run our submission on the hidden case before the final submission so as we can know whether our results can pass the global routing evaluation?

We have already rerun your beta submission and sent the results back to you. If you are asking us to run your final submission early, we cannot do that.

9/3/2024

65. Since we want to predict the final runtime on the evaluation machine, Can the contest provide us the information of Cpu?

Please refer to QA No. 47 for the platform information. The Nvidia server has 8 A100 GPUs and 64 AMD EPYC 7742 CPU cores. We will restrict to using only 8 cpu threads

(4 cpu cores with 2 threads for each core) and 200 GB of memory. We will use “docker run -d -w /app -it --cpus=4 --gpus="device=0,1,2,3" -m 200g --name cadcxxxx_con cadcxxxx” to run the docker container.

9/4/2024

66. Since the final score of each case will be scaled to 100 points, and I believe that the final evaluation will be the top three teams with the highest scores will win the contest instead of the least scores? May be the document about this should be updated?

    The score mentioned in the original document refers to the original score before scaling. And yes, after scaling to 100, the top three teams with the highest combined scores will win the contest. We have updated the scoring section in the document to reflect this.

9/5/2024

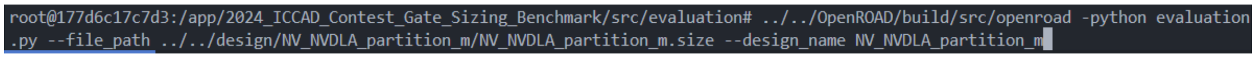
67. We followed the instruction to build the docker container, but the evaluation.py doesn't work properly. We used the benchmark .size file as the input of evaluation.py, and it returned a wrong score.

    We tried to evaluate the output/NV_NVDLA_partition_m.size(which is generated by our model in docker) in a local built environment. The result meets our expectations.

    Do you know how to fix this problem?

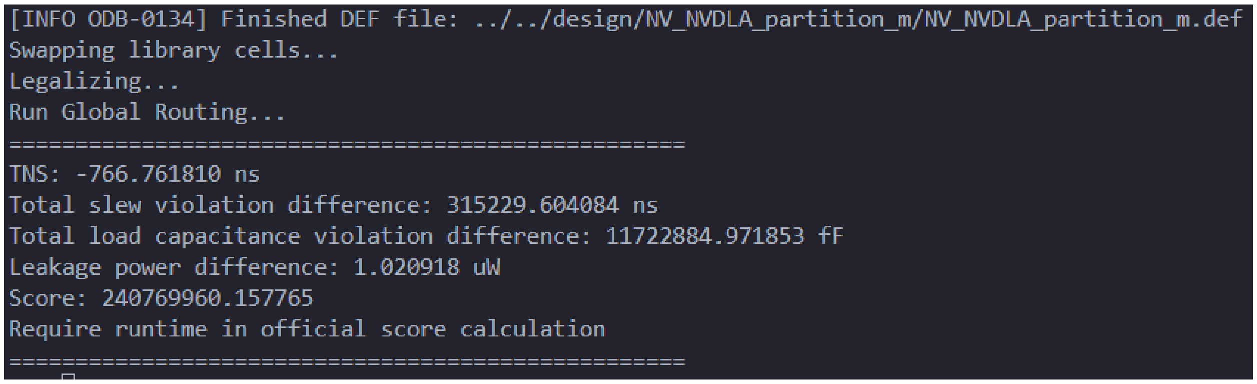
    Besides, We wonder if you will run the evaluation in our submitted docker container or use your own evaluation script?

    The following are the evaluation results by using benchmark NV_NVDLA_partition_m.size in docker container and local built environment.
    command:
    root@177d6c17c7d3:/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/src/evaluation# ../../OpenROAD/build/src/openroad -python evaluation.py --file_path ../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.size --design_name NV_NVDLA_partition_m



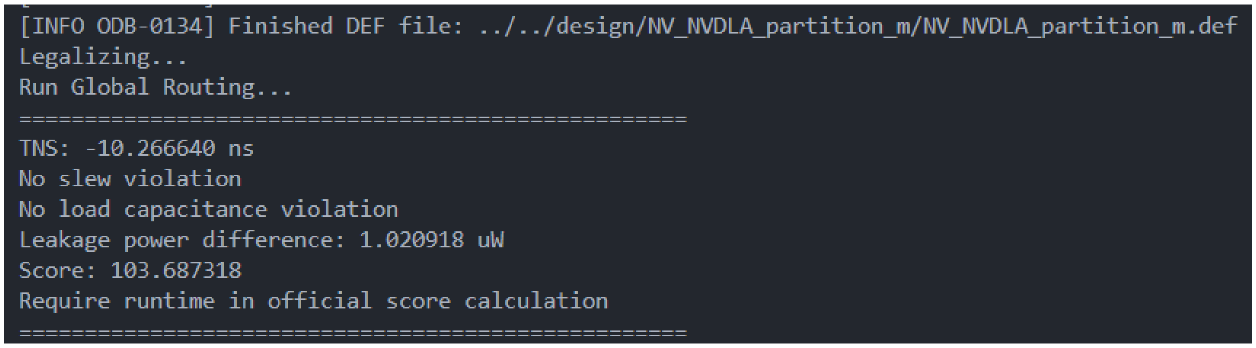
```
root@177d6c17c7d3:/app/2024_ICCAD_Contest_Gate_Sizing_Benchmark/src/evaluation# ../../OpenROAD/build/src/openroad -python evaluation.py --file_path ../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.size --design_name NV_NVDLA_partition_m
```

    1. result in docker



```
[INFO ODB-0134] Finished DEF file: ../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.def
Swapping library cells...
Legalizing...
Run Global Routing...
===================================================
TNS: -766.761810 ns
Total slew violation difference: 315229.604084 ns
Total load capacitance violation difference: 11722884.971853 fF
Leakage power difference: 1.020918 uW
Score: 240769960.157765
Require runtime in official score calculation
===================================================
```

2. result in local built env



```
[INFO ODB-0134] Finished DEF file: ../../design/NV_NVDLA_partition_m/NV_NVDLA_partition_m.def
Legalizing...
Run Global Routing...
===================================================
TNS: -10.266640 ns
No slew violation
No load capacitance violation
Leakage power difference: 1.020918 uW
Score: 103.687318
Require runtime in official score calculation
===================================================
```

Thanks for your help!

1) The log you provided, which you claimed is from the docker container, has additional log that do not belong to the evaluation script on github. Please confirm that you are running the exact same script we provided. You can refer to QA 18 and 56 and the github issue for this particular question. We guarantee that the numbers will be correct on the evaluation platform.

2) We will run the evaluation in the docker container using our own evaluation script, which is identical to the one in the github repository.