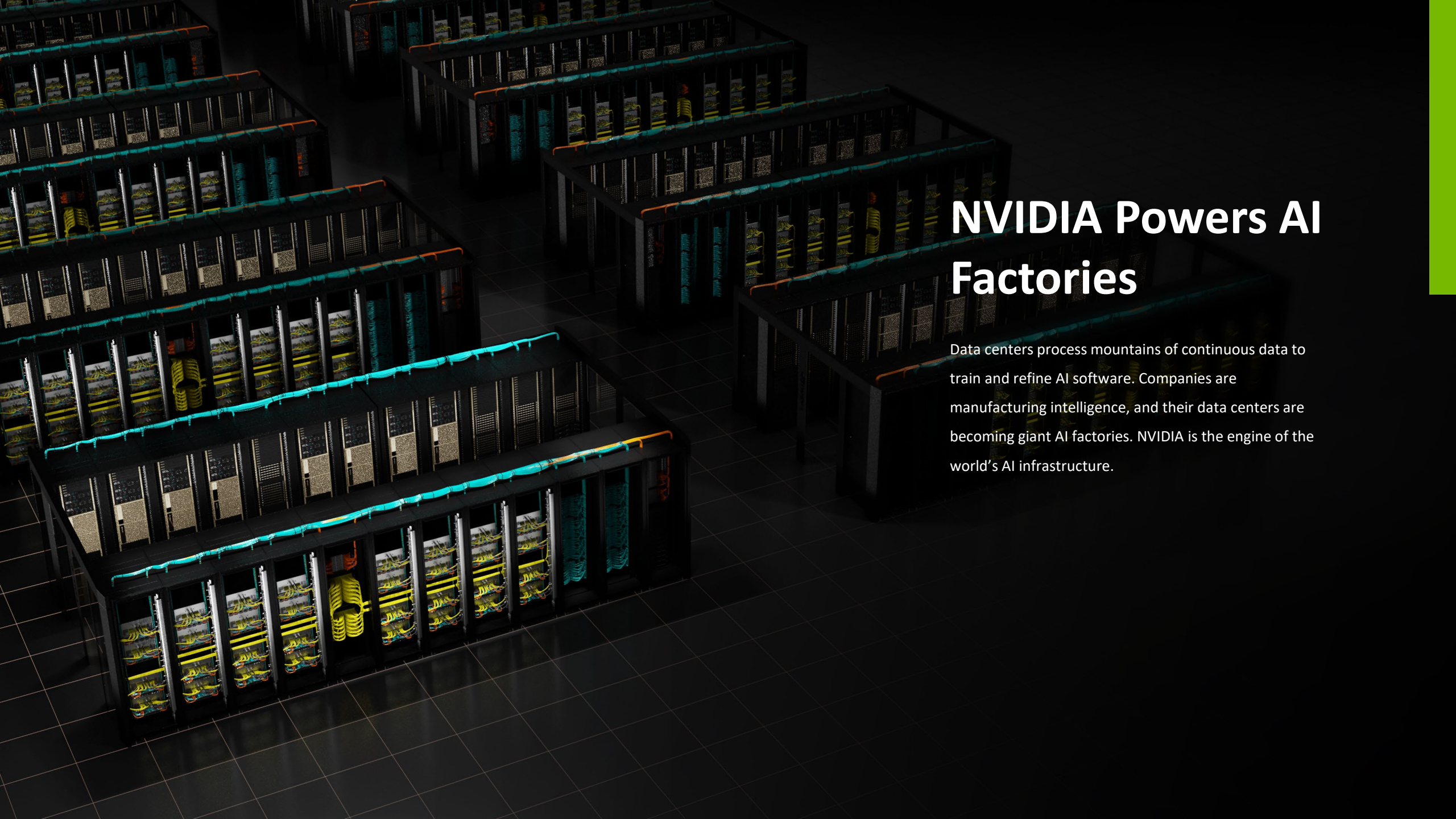
Catching unwanted synthesis optimization on RTL anchor points

Presenter: Sanjana Dhiran
Authors: Sanjana Dhiran, Rahul Shivananda, Alon Berenshtein
Nvidia

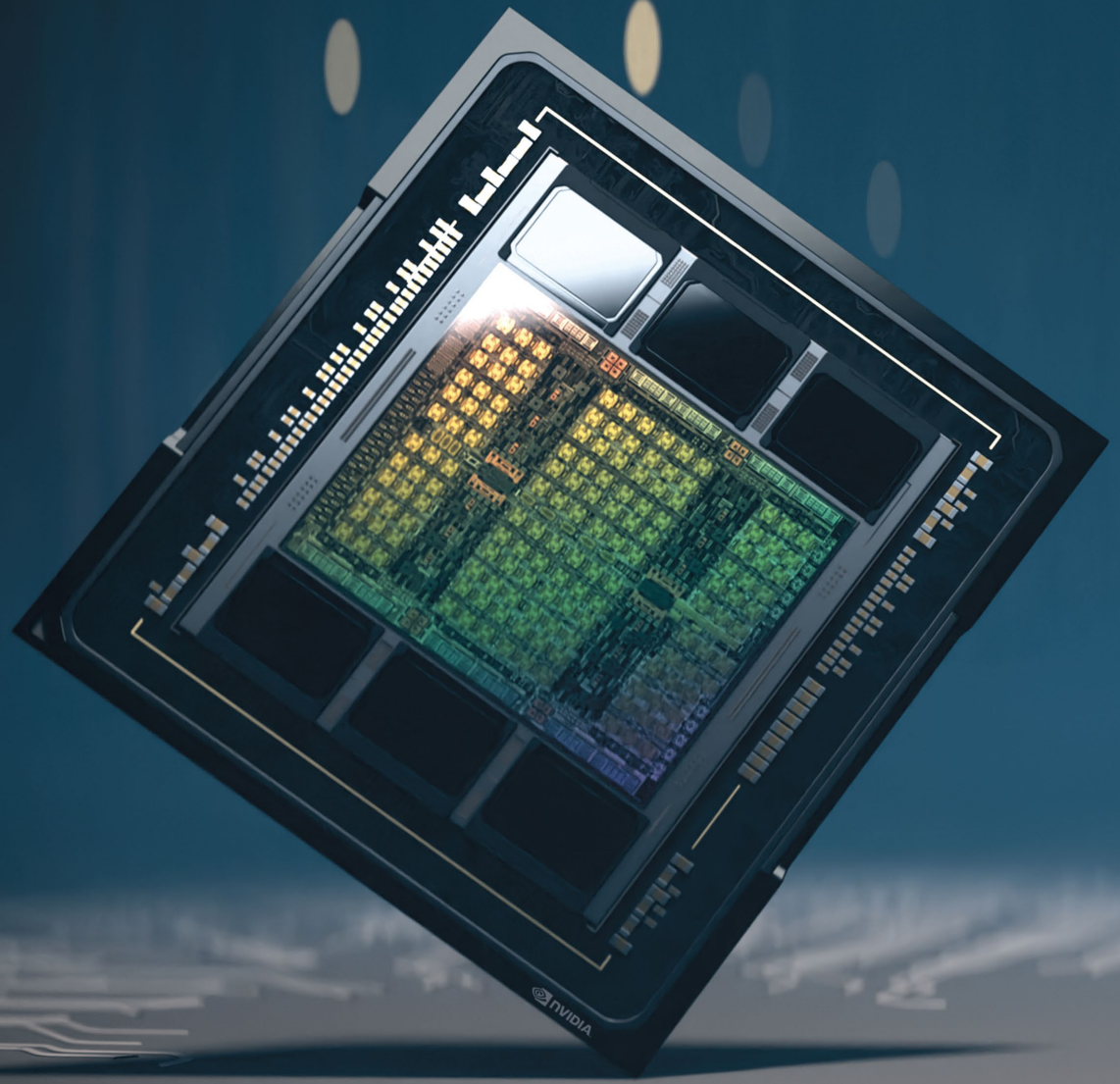
Agenda

- About Nvidia
- Problem Statement
- Proposed Flow and Challenges
- Final Implemented Flow
- Future Enhancements



NVIDIA Powers AI Factories

Data centers process mountains of continuous data to train and refine AI software. Companies are manufacturing intelligence, and their data centers are becoming giant AI factories. NVIDIA is the engine of the world’s AI infrastructure.



Sparking the iPhone Moment of AI

The acceleration of deep learning ignited the big bang of AI. ChatGPT, a large language model powered by an NVIDIA DGX™ AI supercomputer, reached 100 million users in just two months. Its magical capabilities have captured the world's imagination. Generative AI is a new computing platform, like the PC, internet, and mobile-cloud. Accelerated computing and AI have fully arrived.

What's the definition of a large language model?

A large language model is a type of artificial intelligence system that has been trained on massive amounts of text data and can generate human-like language responses to input.

Problem Statement

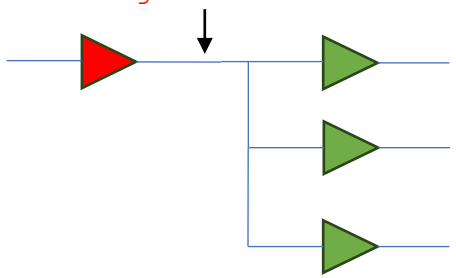
BACKGROUND

Why we need anchor cells?

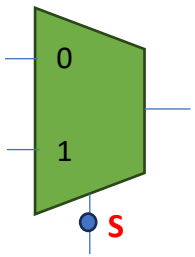
- Anchor cells are the cells that sticks through synthesis and are expected to be intact in both rtl and netlist.
- These anchor cells have **`dont_touch`** and **`size_only`** constraints as these may be used in the downstream flows to apply specific constraints.
- For example:

MCP (multi cycle path) for STA

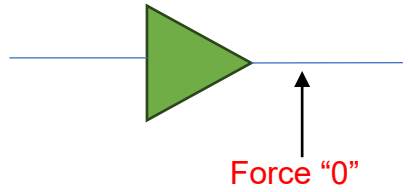
`set_multicycle_path -through`



"**`set_case_analysis 0`**" for DFT at select line "S" mode

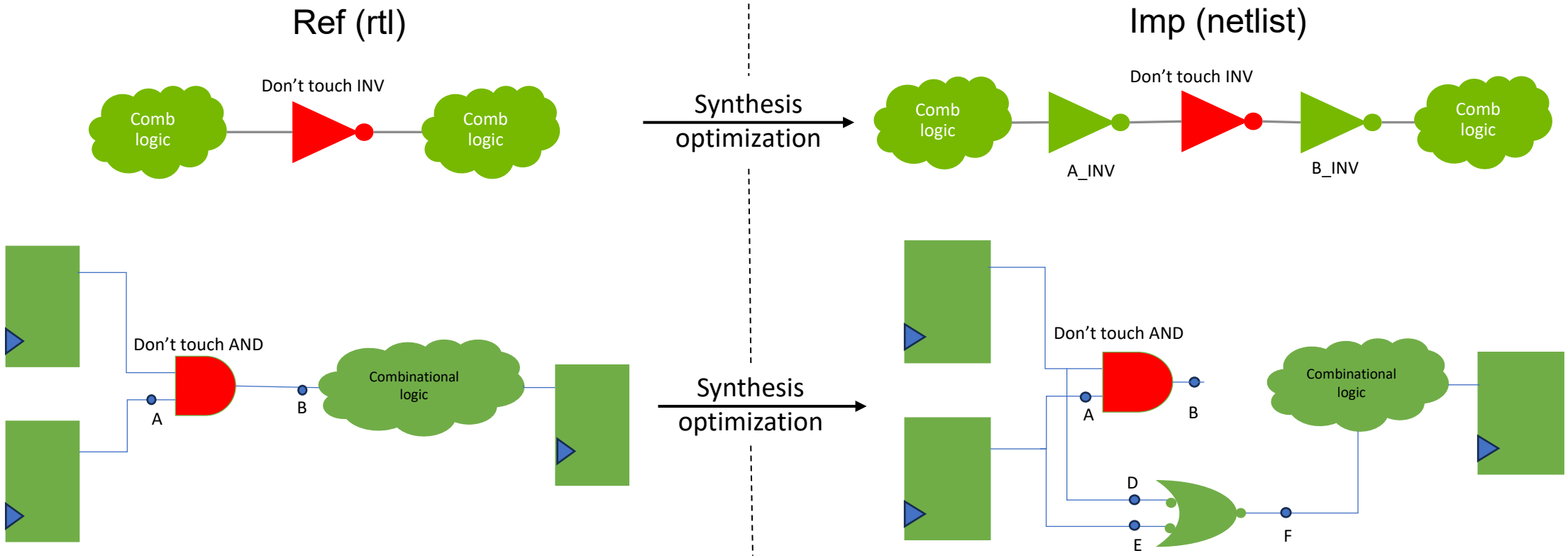


Force "0" at o/p of certain cells for CDC analysis



PROBLEM STATEMENT

- Sometimes synthesis inadvertently optimizes **`dont_touch`** and **`size_only`** cells.
- This is not currently caught by formality if the 2 designs are logically equivalent.

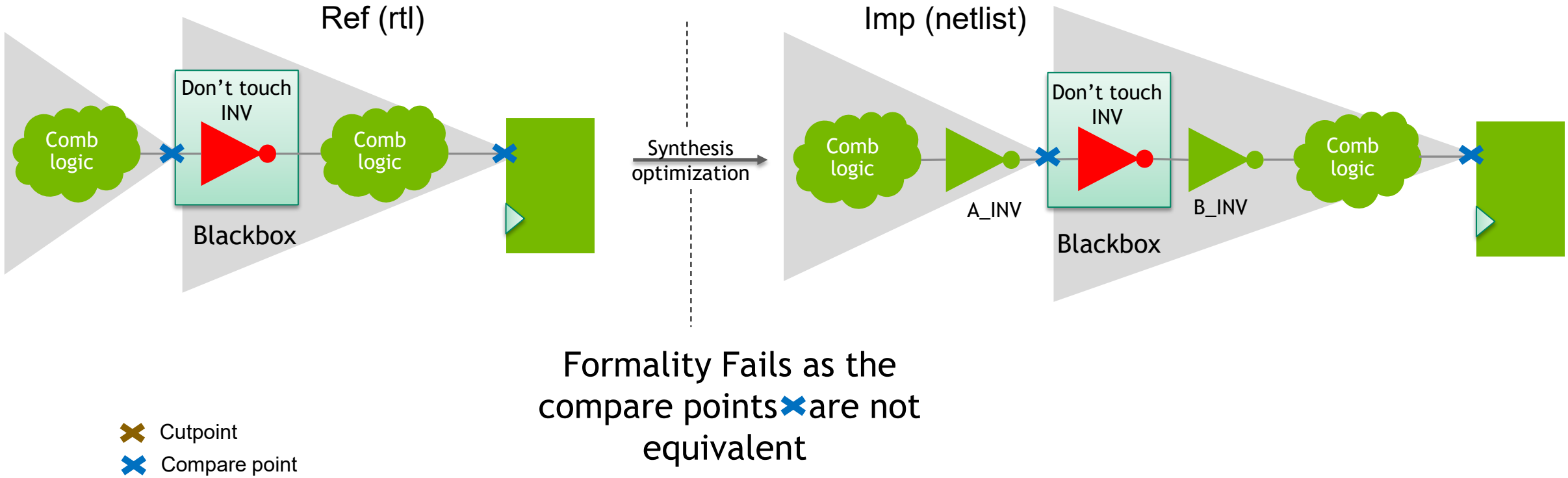


Formality passes as the logic is equivalent

Proposed Flow and Challenges

PRIOR ART

- Bbox cells (**`set_black_box`**) that should not be optimized by synthesis



INITIAL PROPOSED FLOW

- Black Box related setting
  - To perform an identity check between two comparable Blackboxes

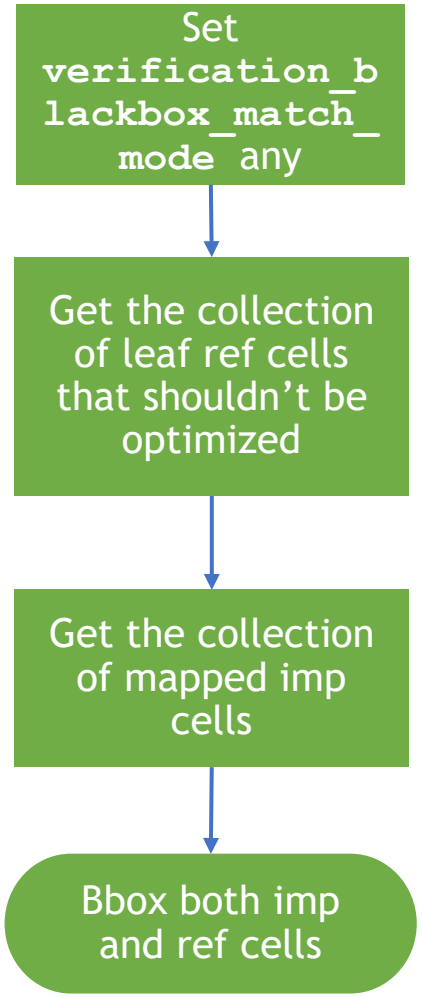
`set verification_blackbox_match_mode` **`any`**

Two Blackboxes are matched regardless of the library or design names

`set verification_blackbox_match_mode` **`identity`**

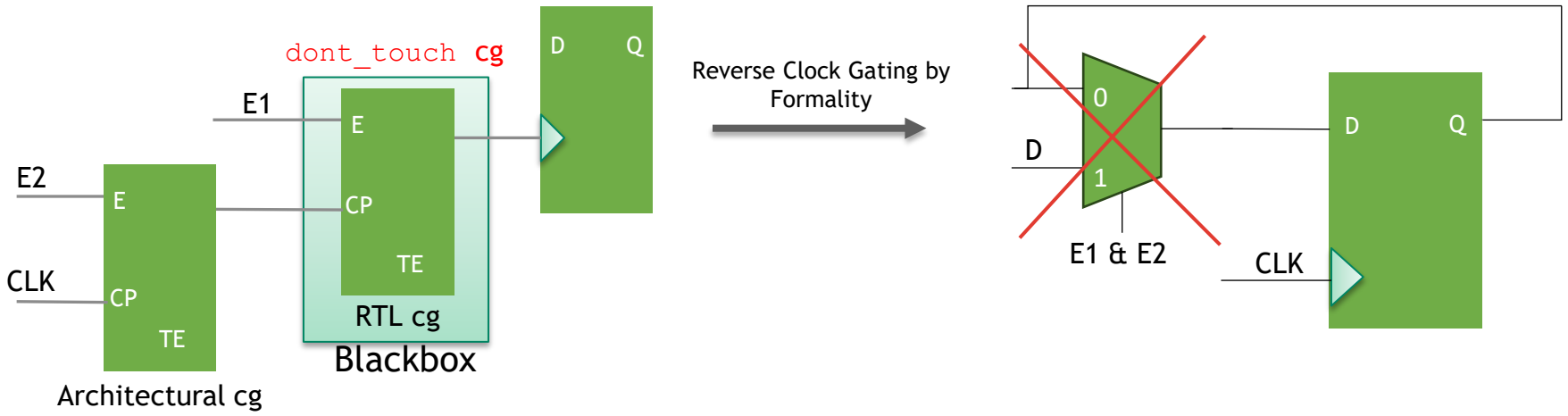
Two Blackboxes are matched only if they have the same library and design names

  - Example: Different drive strength **`size_only`** cells are matched with **`any`** and unmatched with **`identity`**.



CHALLENGES WITH BLACKBOX FLOW

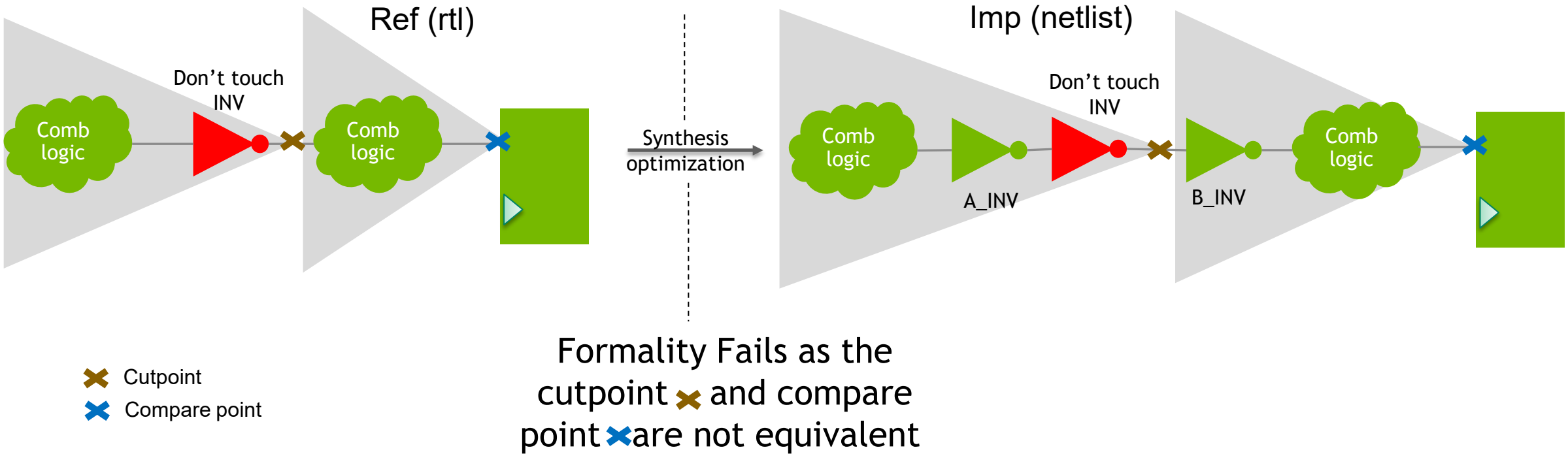
- Setting **`verification_blackbox_match_mode`** to **any** can be dangerous because if synthesis tool incorrectly used a different function for a cell, that would go uncaught because the cell is blackboxed.
- The "**reverse_clock_gating**" flow does not work, if we blackbox **`size_only/don't_touch`** clock gate (cg) anchor points.



Final Implemented Flow

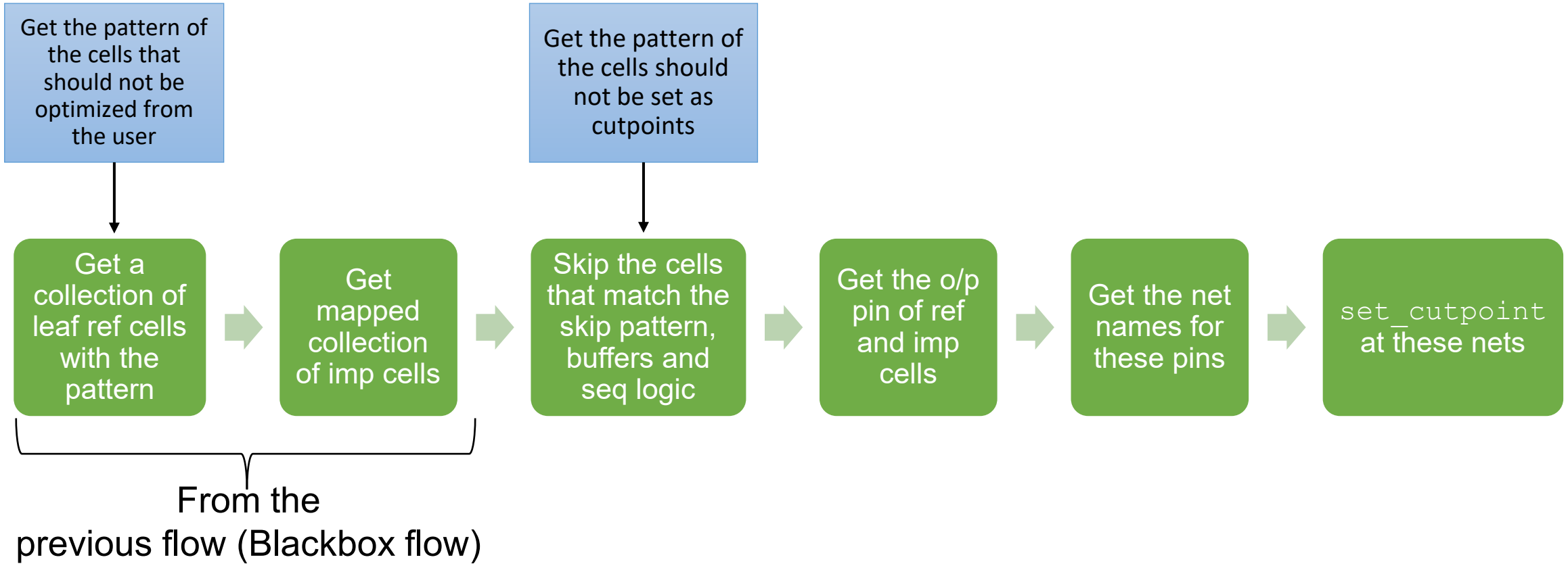
MODIFIED SOLUTION

- The end goal is to have the required cells as compare points.
- Set them as **cutpoints** (`set_cutpoint`) instead of Bbox.



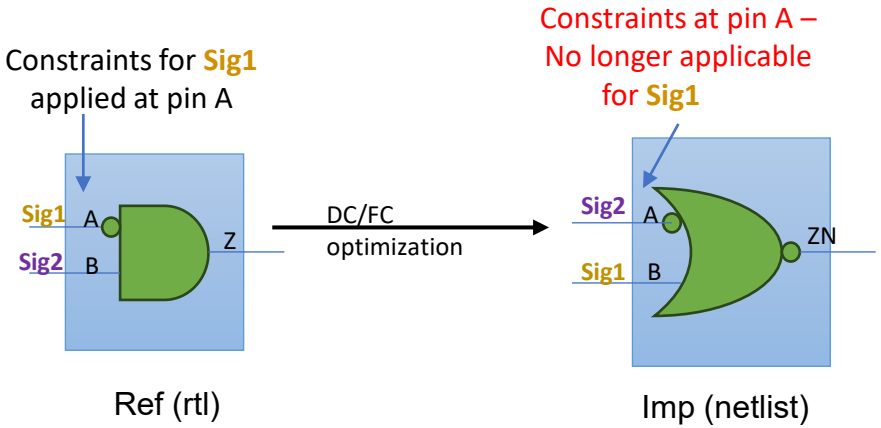
IMPLEMENTED FLOW WITH CUTPOINTS

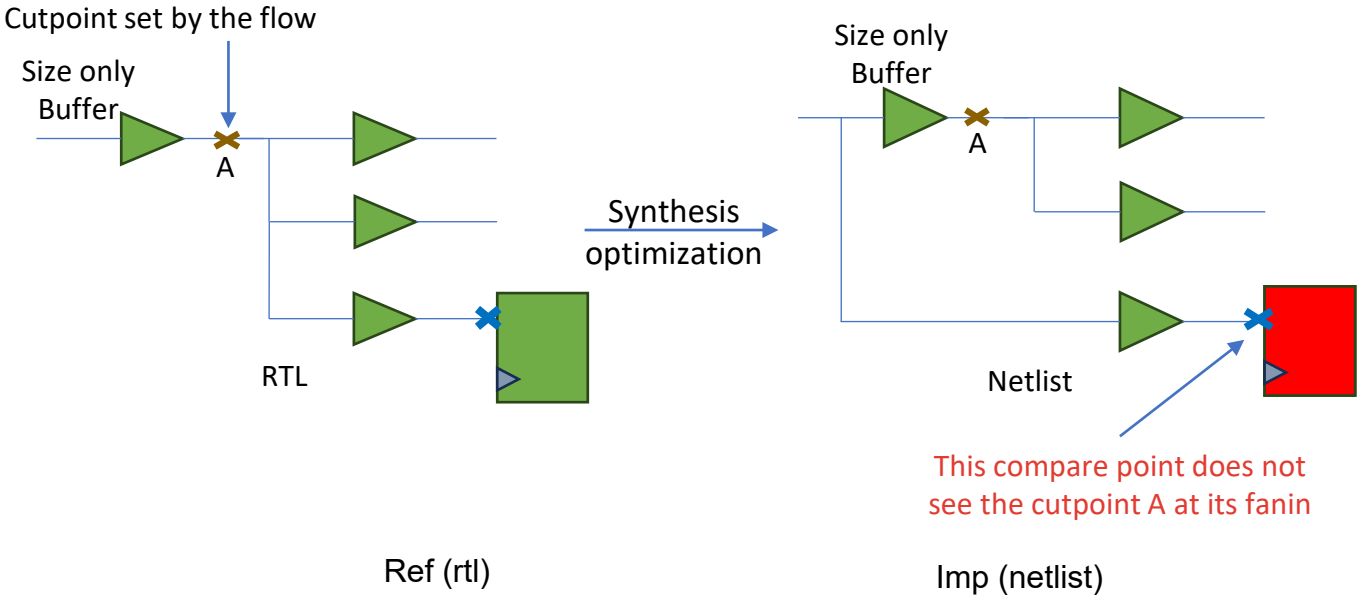
- The logic does not get hidden from the tool
- No need to deal with the Bbox verification settings



REAL ISSUES CAUGHT WITH THE CUTPOINT FLOW

- Caught the swapping of one AND gate lib cell in rtl to a logically equivalent NOR gate lib cell in netlist.
- Movement of fanouts across buffers/inv in the netlist.
- The flow was able to catch certain don't touch cells that were optimized away in the netlist.
- Caught instance name changes done by synthesis leading to potential loss of constraints applied to rtl instance names





Cutpoint

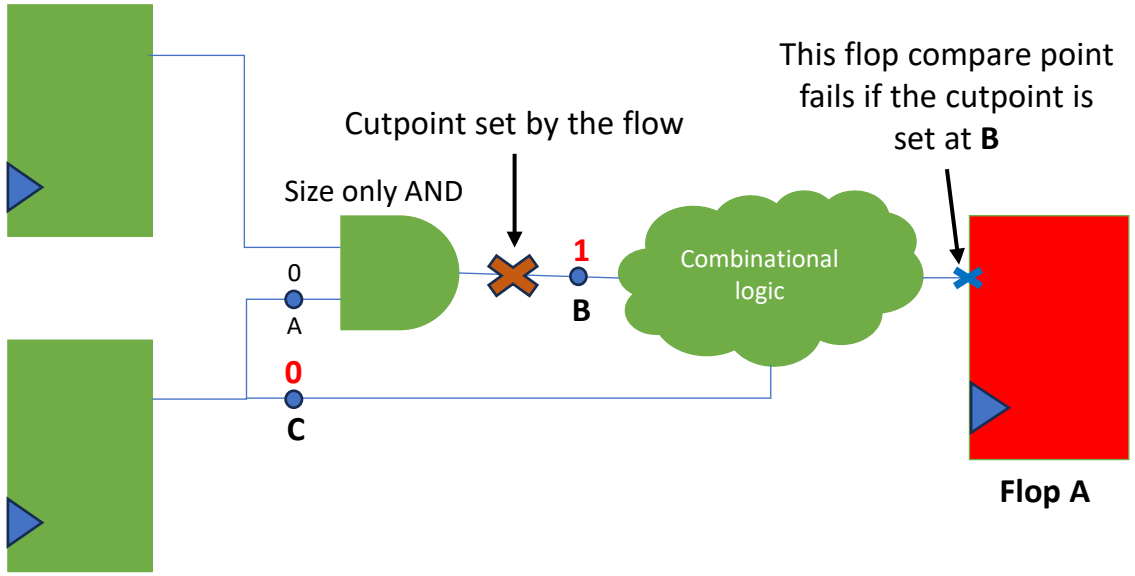
Compare point

CHALLENGES WITH THE CUTPOINT FLOW

- Tool does not allow setting cutpoints at leaf cell pins.
  - Hence had to set them at corresponding nets, which don't always have a predictable name.
- Had to apply the cutpoints after the SVF has been processed by the tool.
- Tool does not allow setting cutpoints at net objects (**Fixed**)
  - Need to apply the cuts on net names instead. There can be conflicts on hier and leaf net objects.
- Had to skip this flow for buffers/inverters as both FC and DC move fanouts across certain buffers. (Fixes for DC/FC are **WIP**)

LIMITATION WITH THE CUTPOINT FLOW

- With the flow in place Formality was throwing false verification failures due to illegal pattern at the cutpoints when there were reconvergent paths bypassing the anchor in the design.
- Example:
  - Illegal i/p pattern to the combinational cloud as B can be 1 and C can be 0. But w/o cutpoint this pattern can never be in place.



Future Enhancements

FUTURE ENHANCEMENTS

- Improvements in Formality
  - It should be able to set cutpoints at tech lib instance pins instead of nets for improved debug ability.
  - We need a pattern legalizer in Formality, to avoid the false failures on reconvergent logic.

- Improvements in DC/FC
  - The tool should not move logic around **`dont_touch`** and **`size_only`** buffers/inv.
  - We keep discovering new cases of unwanted optimizations in newer synthesis tool releases.


THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY