



# Achieve Dataflow and Global Placement Coherence using Virtual Object Constraint

## Author | Email:

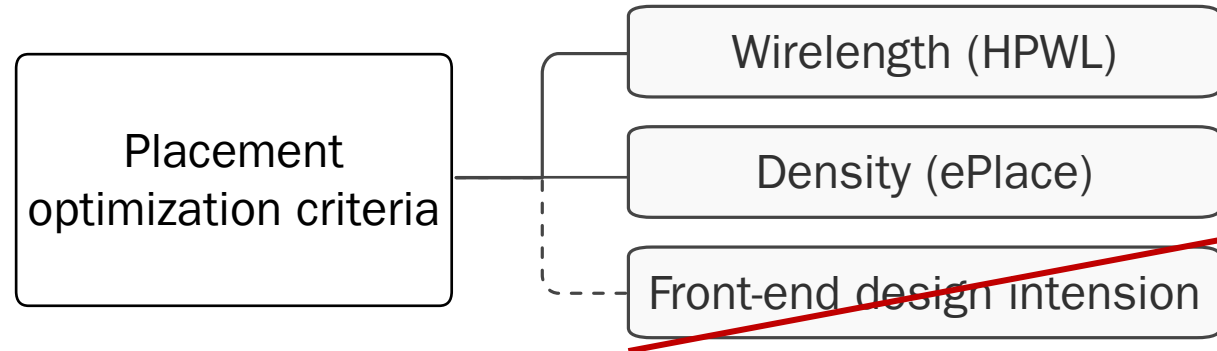
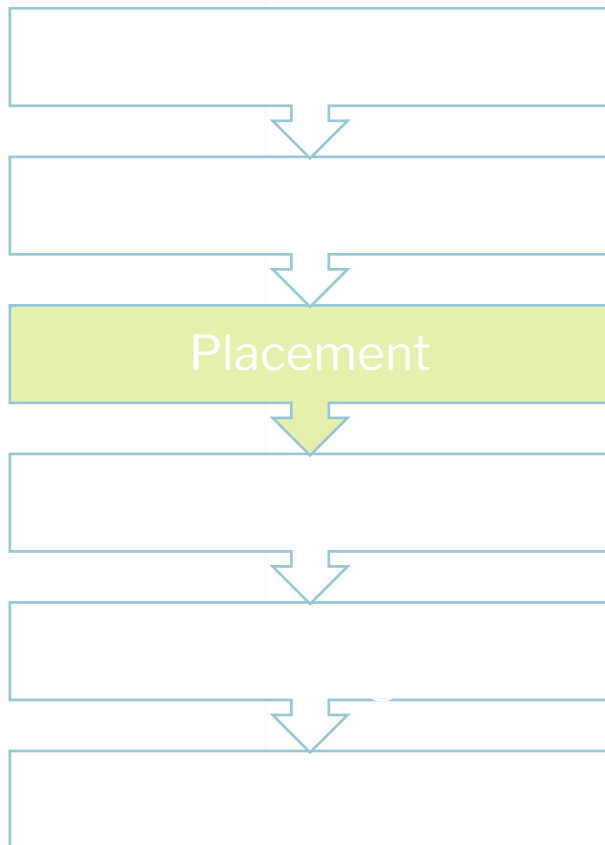
- Jing Wang [wangjing509@hisilicon.com](mailto:wangjing509@hisilicon.com)
- Yang Liu [liuyang169@hisilicon.com](mailto:liuyang169@hisilicon.com)
- Zixiao Wang [zixiaowang@hisilicon.com](mailto:zixiaowang@hisilicon.com)
- Qiuling Zeng [zengqiuling@hisilicon.com](mailto:zengqiuling@hisilicon.com)

## Presenter:

- Dmitry Yakimets [dmitry.yakimets@huawei.com](mailto:dmitry.yakimets@huawei.com)



# Introduction



Constraints are applied to modules or critical paths:

- Placement bounds
- Pre-placed cells
- ...

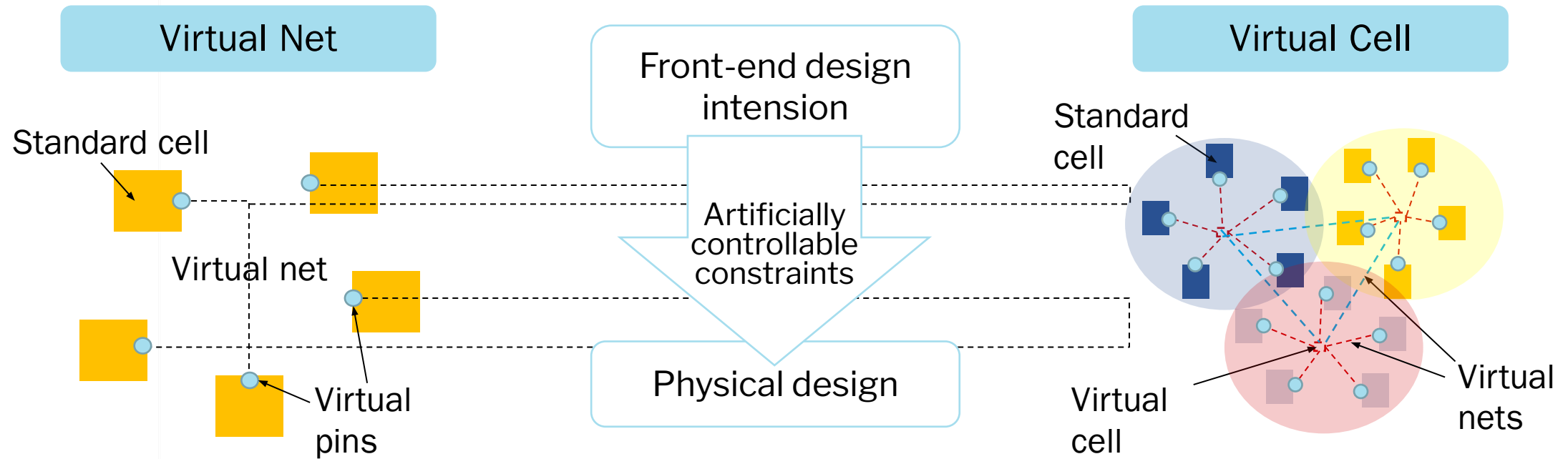


Limited flexibility  
Multiple iteration  
Degraded PPA

How to guide global placement to achieve coupling between front-end logical information and physical design process without PPA loss?



# Virtual Object Constraint

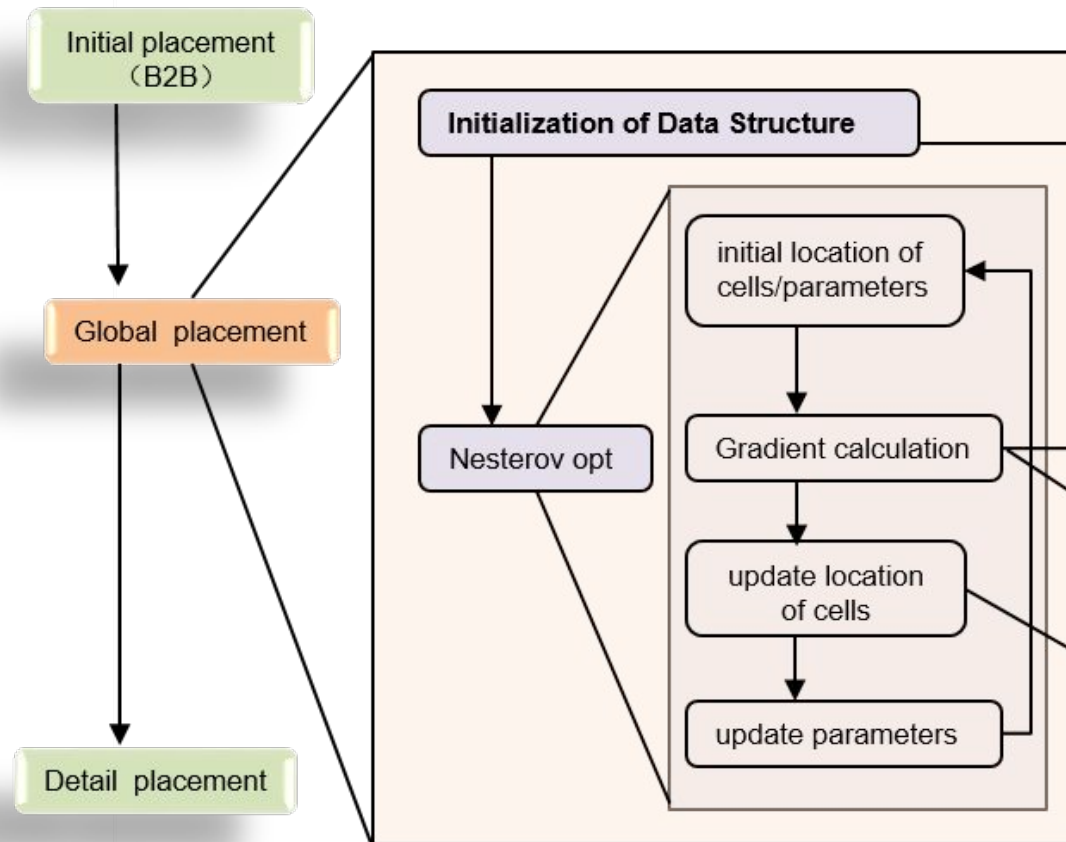


- Clusters cells together even if they are not connected in a netlist
- Optimized just like real nets

- Enables guided placement of submodules



# Implementation



Additional operations w.r.t. original flow:

- Introduce virtual objects (cell, pin, net) in data structure according to pin list specified by users.
- Initialize weight of virtual net based on user settings

● wirelength gradient of virtual net

● density gradient of virtual cell

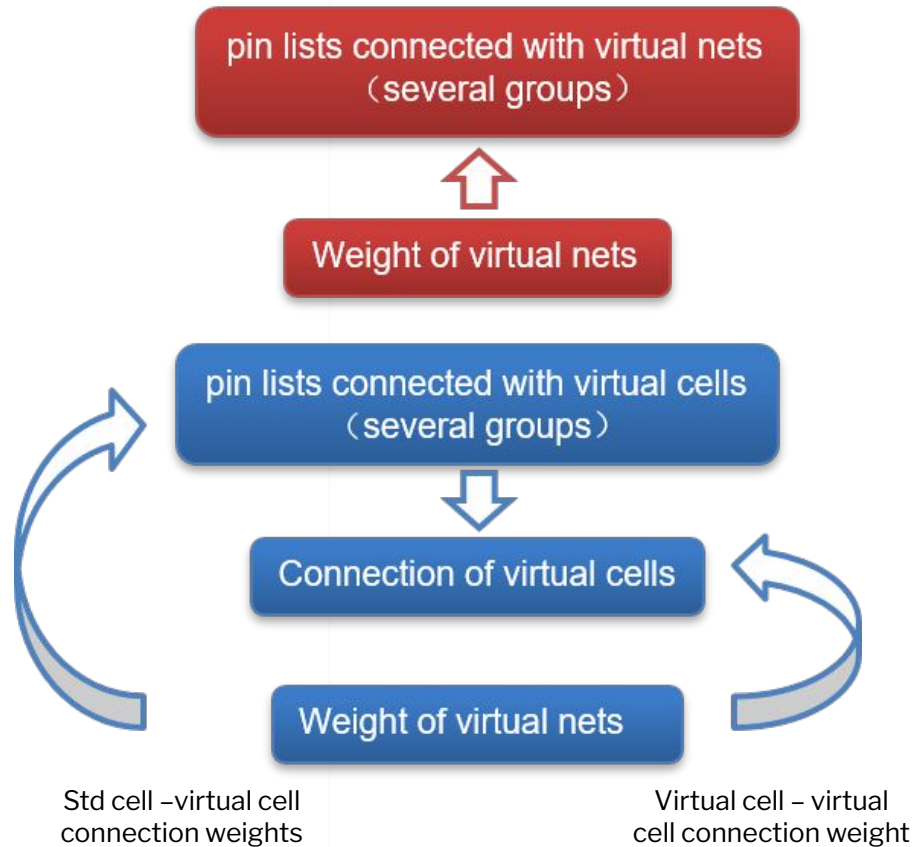
● calculate location of virtual cell

New artificially controllable constraints are implemented under framework of OpenROAD





# User Interface

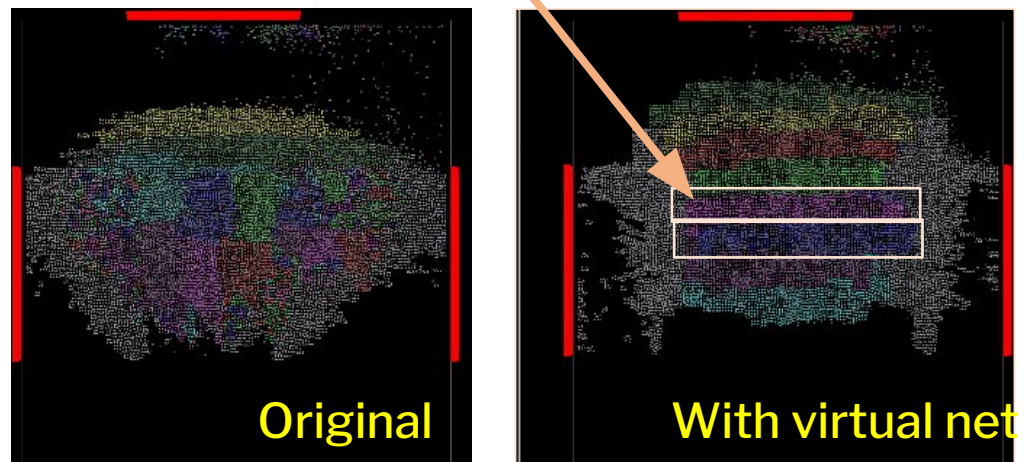
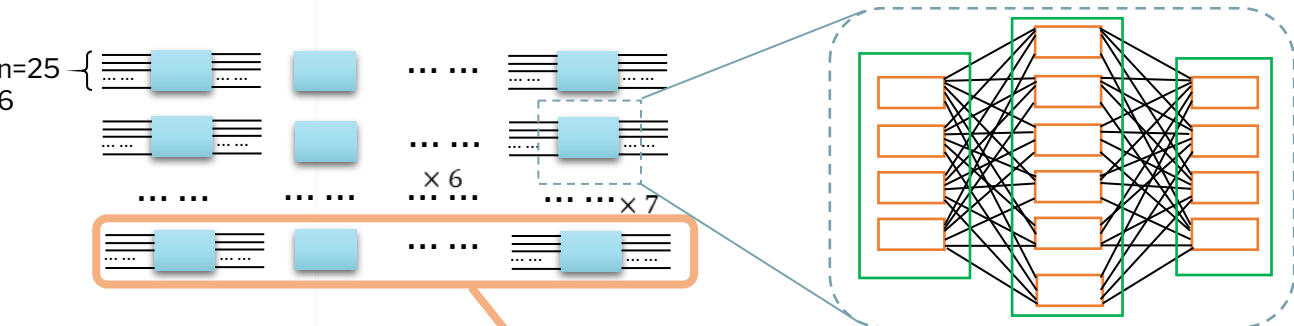


```
tcl
1 set pin_list_for_virtual { {pin1_1 pin1_2 pin1_3 ...} {pin2_1 pin2_2 pin2_3 ...} ...}
2 -user_virtual -with_virtual_cell $pin_list_for_virtual
3 -apply_virtual_net
4 set net_weight {weight_1 weight_2 ...}
5 -virtual_net_weight $net_weight
```

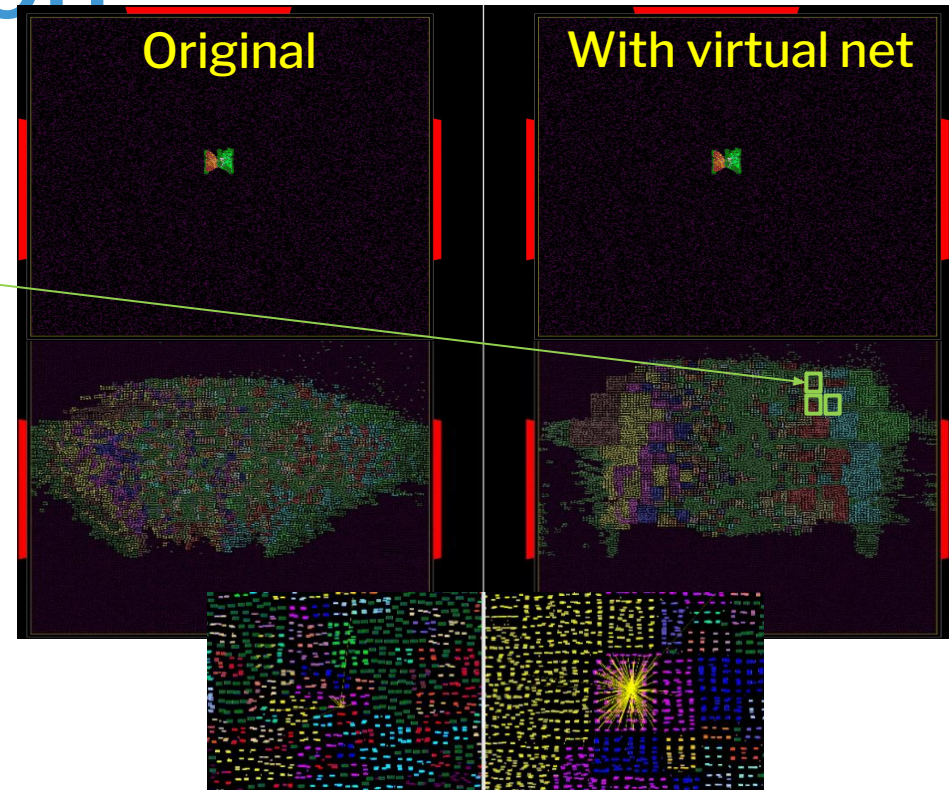
```
tcl
1 ## pin_list
2 set pin_list_for_virtual { {pin1_1 pin1_2 pin1_3 ...} {pin2_1 pin2_2 pin2_3 ...} ...}
3 -user_virtual -with_virtual_cell $pin_list_for_virtual
4
5 ## net weight
6 set net_weight {weight1 weight2 weight3 ...}
7 -virtual_net_weight $net_weight
8 set net_weight2 { {weight1_1 weight1_2 weight1_3 ...} {weight2_1 weight2_2 weight2_3 ...} ...}
9 -virtual_net_weight2 $net_weight2
10
11 ## virtual cell connection
12 set virtual_connect_list { {virtual_cell_index1 virtual_cell_index2 ...} {virtual_cell_index3 virtual_cell_index4 ...} ...}
13 -virtual_cell_connection $virtual_connect_list
14 set connection_weight {weight1 weight2 ...}
15 -virtual_connection_weight $connection_weight
```



# Clos Network Optimization



Same color for standard cells of one layer



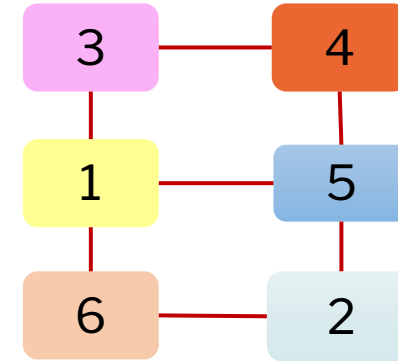
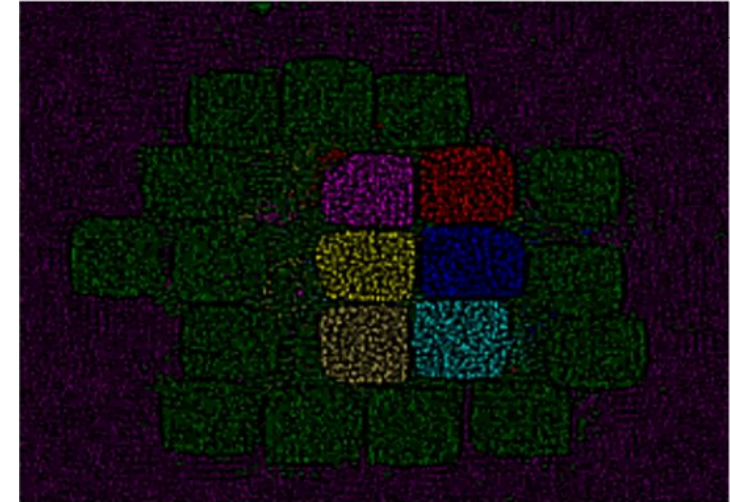
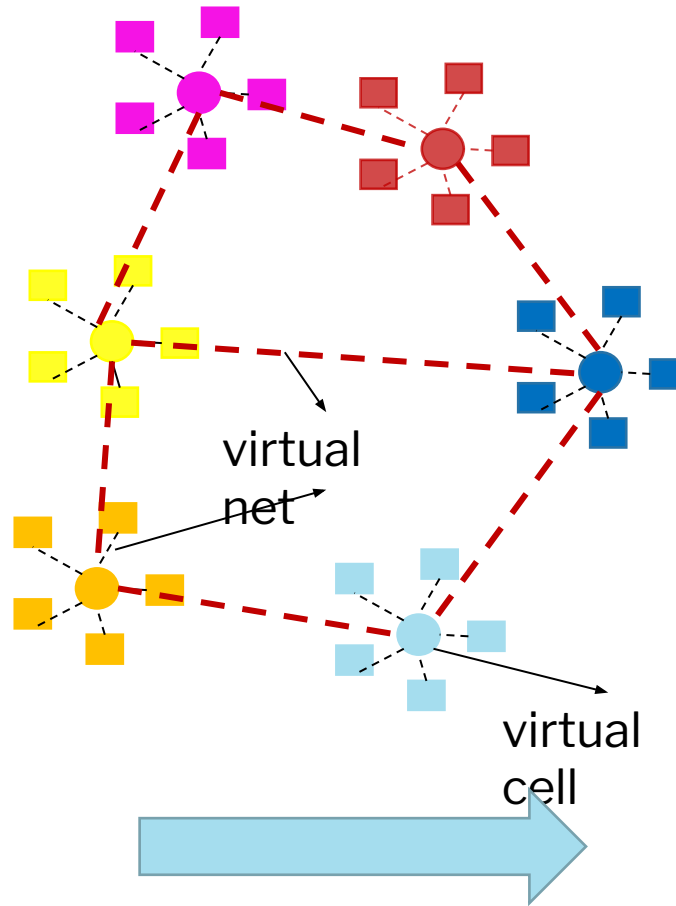
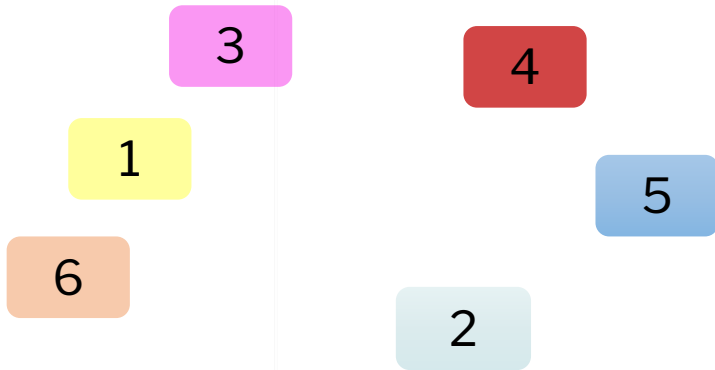
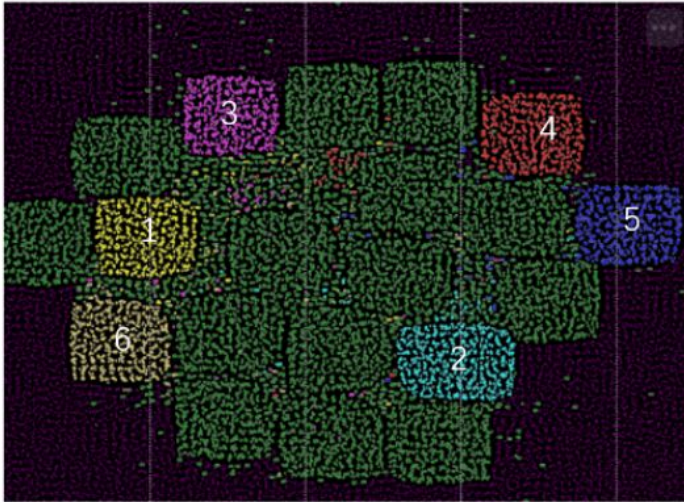
Same color for standard cells of a single crossbar

Thanks to virtual nets, placement optimization behavior is indirectly guided to be consistent with the front-end





# Optimization of Submodules Placement



# Summary

- Placement is a step of paramount importance in chip physical design
- Traditional placer driven by objectives of minimizing total wirelength and density cause mismatch between actual dataflow and design intention
- This work introduces new artificially controllable constraints: virtual objects for global placement.  
They allow to guide the placement in a more intuitive and simple way, coherent with design intension
- This work promotes coupling of front-end logical information and physical design process







**JULY 9-13, 2023**

**MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA**

# Thank you

