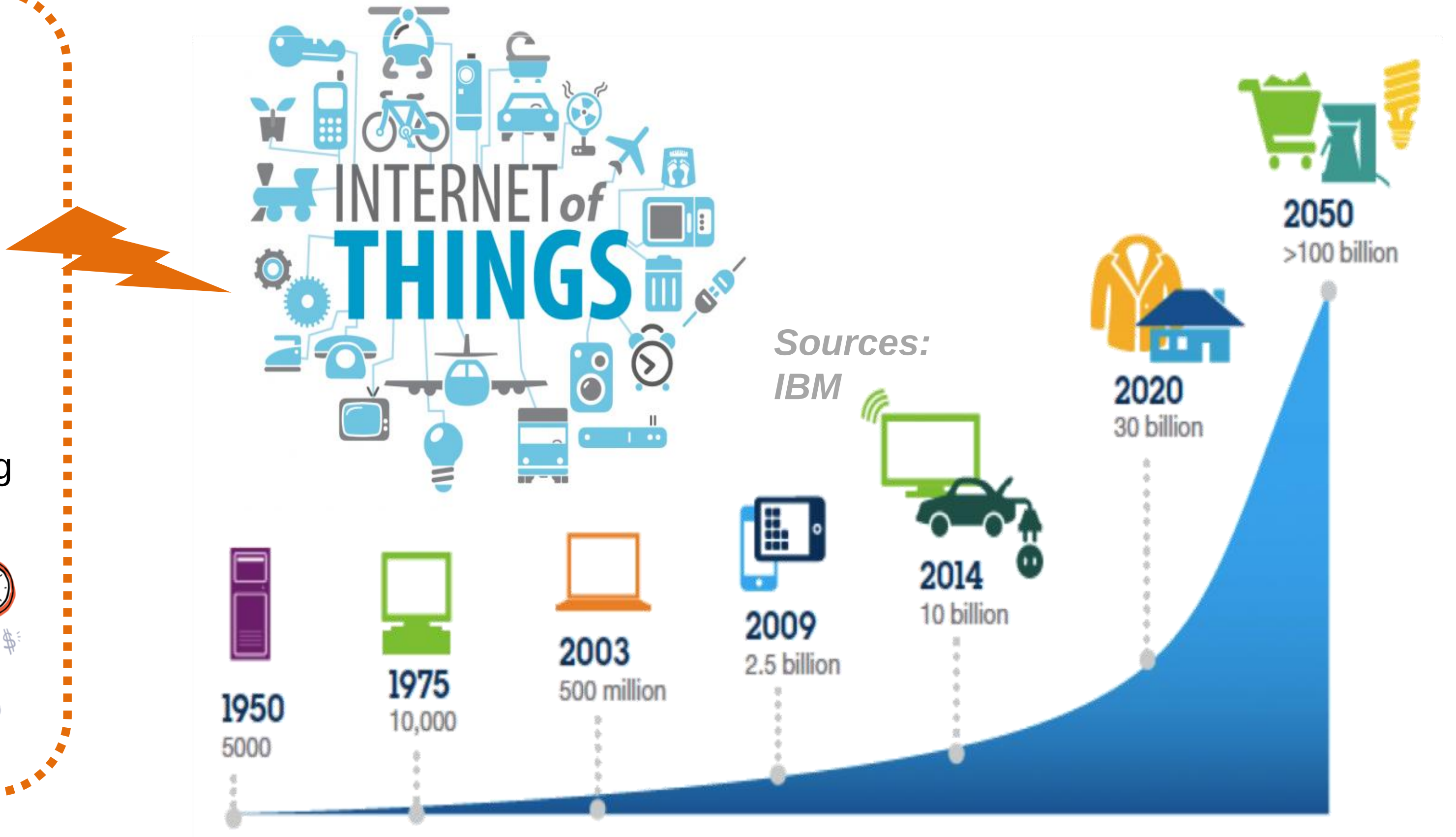
TEXAS
The University of Texas at Austin

Generative-Adversarial-Network-Guided Well-Aware Placement for Analog Circuits

Keren Zhu, Hao Chen, Mingjie Liu, Xiyuan Tang, Wei Shi, Nan Sun and David Z. Pan
ECE Department
The University of Texas at Austin
This work is supported in part by the NSF under Grant No. 1704758, and No. 2112665 and the DARPA ERI IDEA program

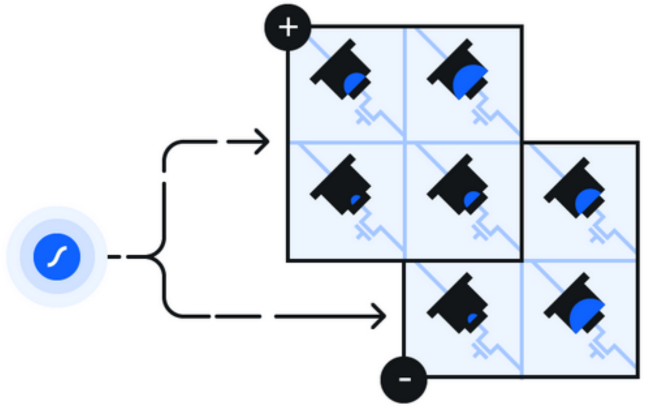
Background: Automating AMS Layouts

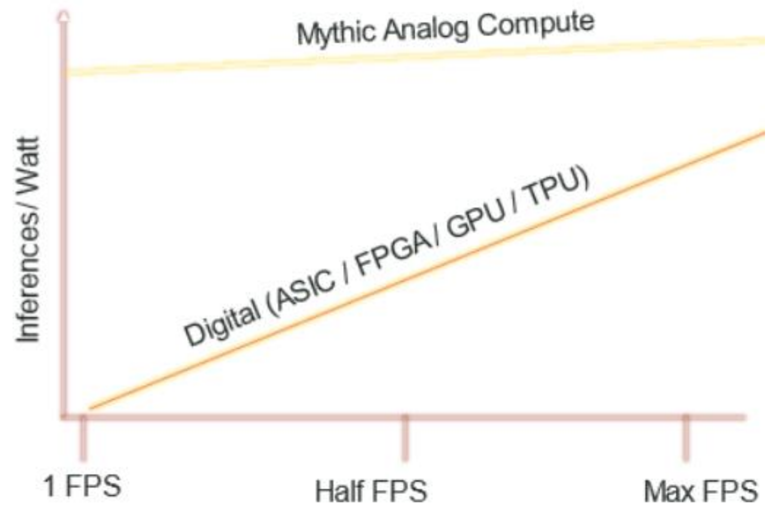
- There are high demand for analog and mixed signal (AMS) circuits
- Drawing AMS layouts are still manual and cost time



Why AMS Matters

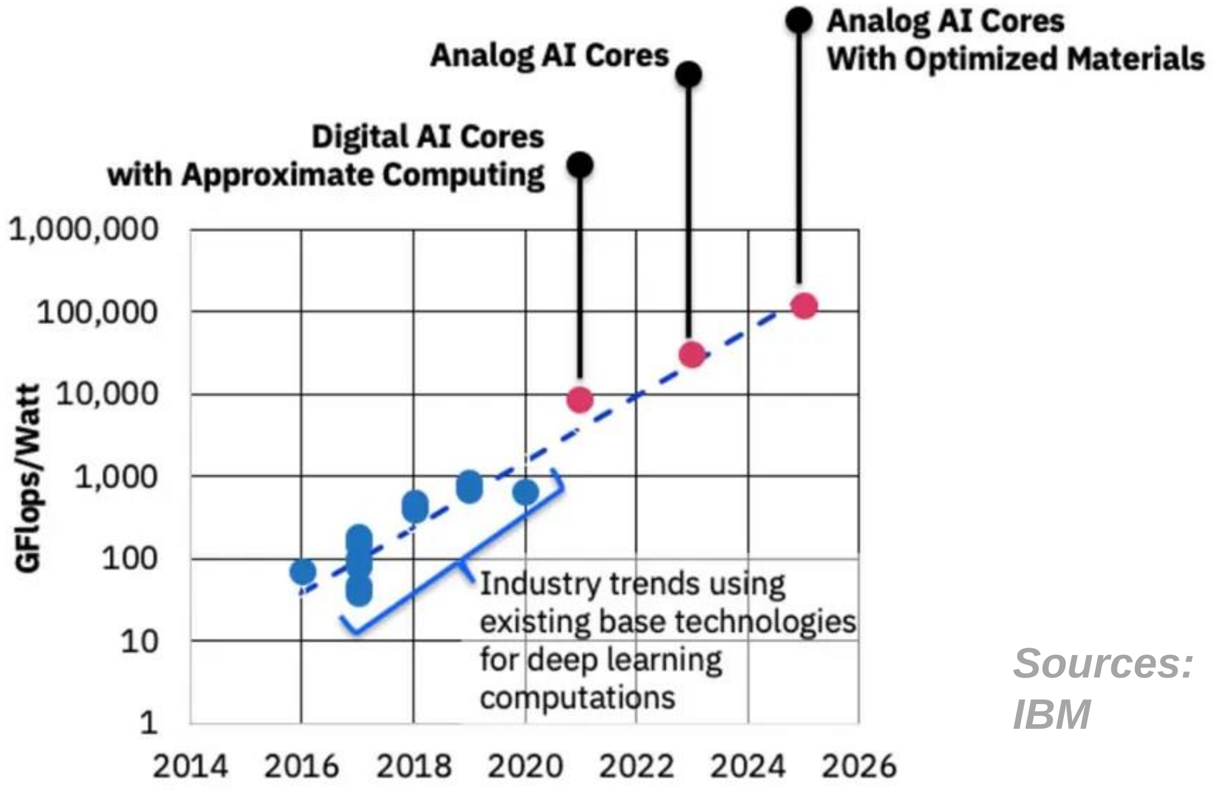
- Analog computing for AI chip





Sources: Mythic

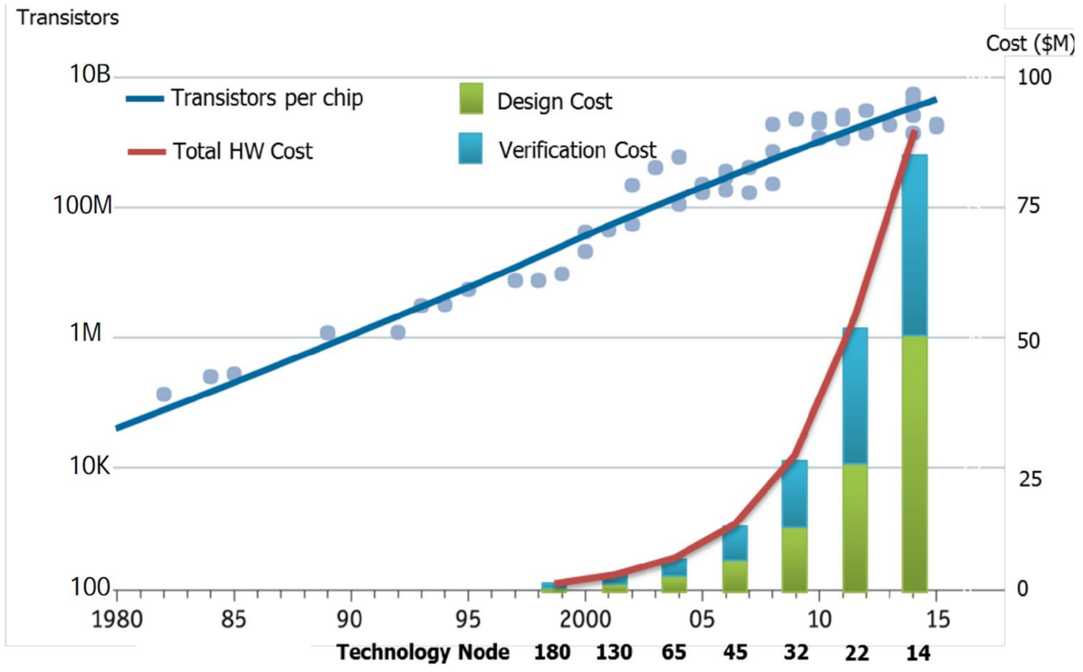
RAIN NEUROMORPHICS
ANALOG INFERENCE
MYTHIC
Hewlett Packard Enterprise
IBM



Sources: IBM

The Mixed-Signal Design Problem

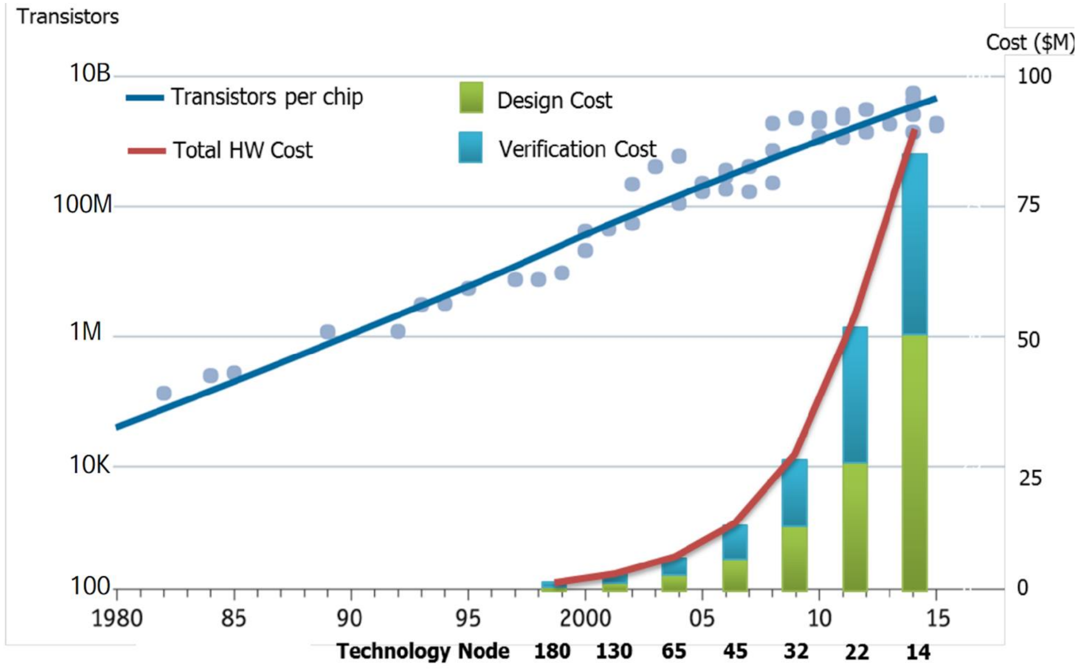
- ASIC design cost is keep increasing.
- EDA is a key.



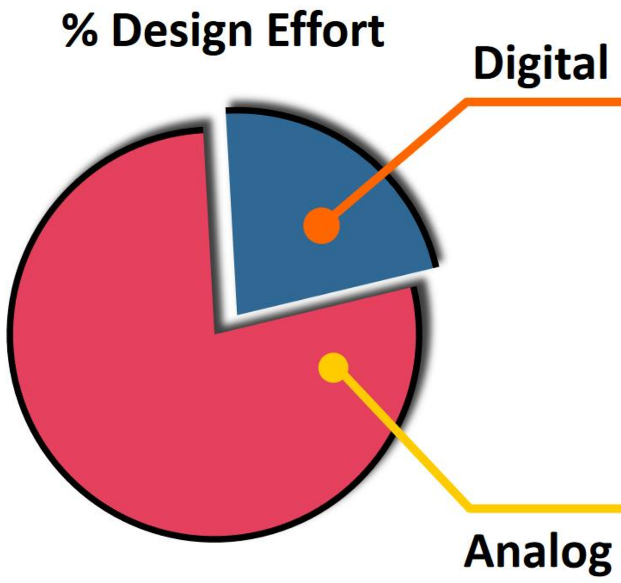
[Olofsson, *2018*]

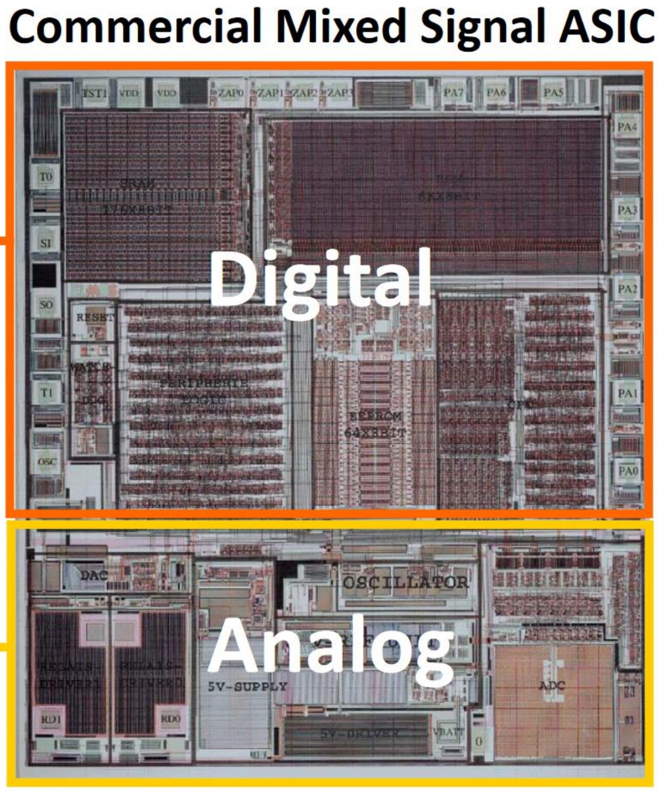
The Mixed-Signal Design Problem

- Analog design is mainly manual!



[Olofsson, *2018*]

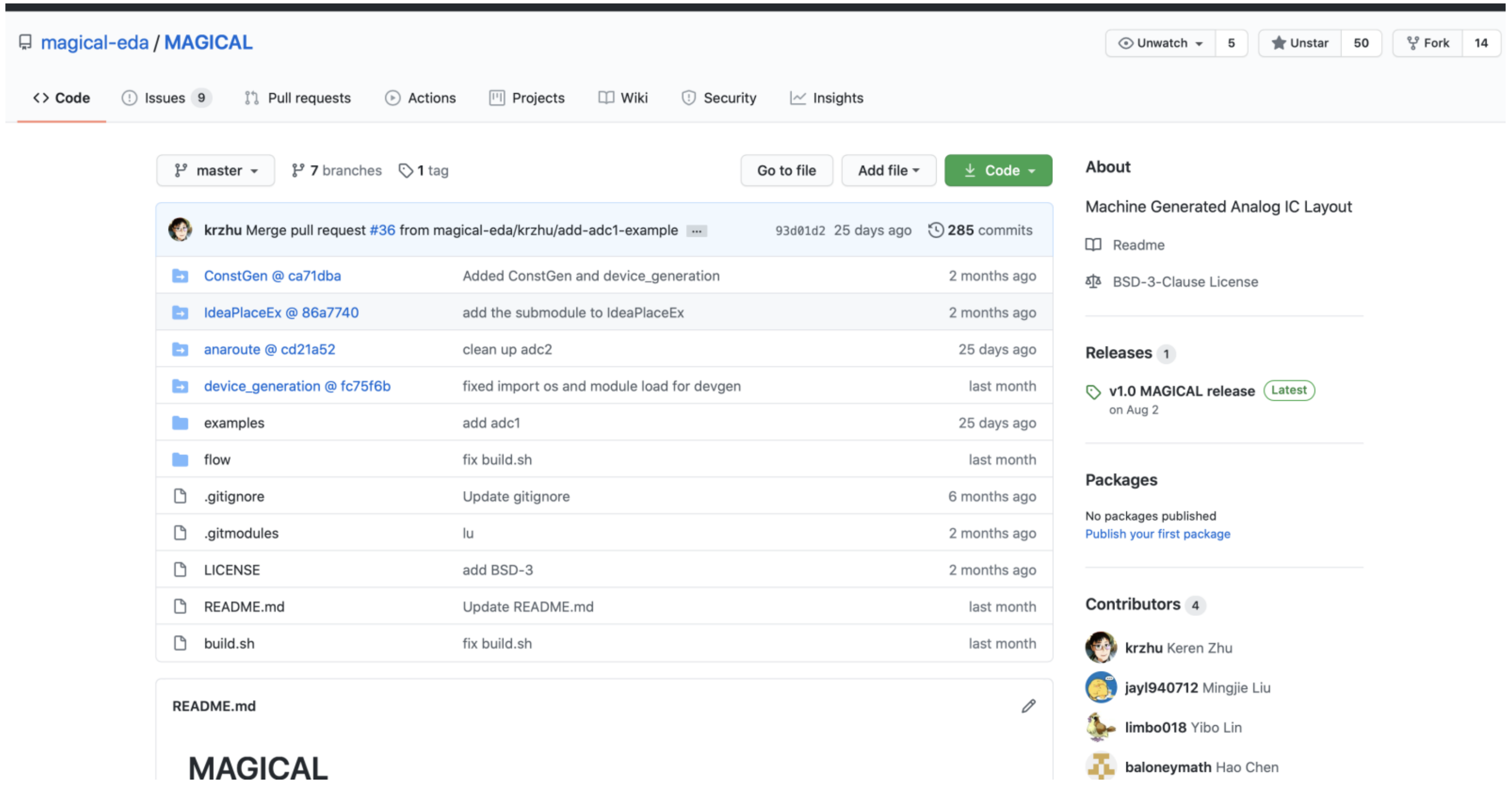




[Rutenbar, *2010*]

Background: MAGICAL

- This work is based on MAGICAL https://github.com/magical-eda/MAGICAL
- Machine Generated Analog IC Layout



magical-eda / MAGICAL

Unwatch 5 | Unstar 50 | Fork 14

Code | Issues 9 | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 7 branches | 1 tag | Go to file | Add file | Code

krzhu Merge pull request #36 from magical-eda/krzhu/add-adc1-example … 93d01d2 25 days ago 285 commits

ConstGen @ ca71dba	Added ConstGen and device_generation	2 months ago
IdeaPlaceEx @ 86a7740	add the submodule to IdeaPlaceEx	2 months ago
anaroute @ cd21a52	clean up adc2	25 days ago
device_generation @ fc75f6b	fixed import os and module load for devgen	last month
examples	add adc1	25 days ago
flow	fix build.sh	last month
.gitignore	Update gitignore	6 months ago
.gitmodules	lu	2 months ago
LICENSE	add BSD-3	2 months ago
README.md	Update README.md	last month
build.sh	fix build.sh	last month

README.md

MAGICAL

About

Machine Generated Analog IC Layout

Readme

BSD-3-Clause License

Releases 1

v1.0 MAGICAL release Latest
on Aug 2

Packages

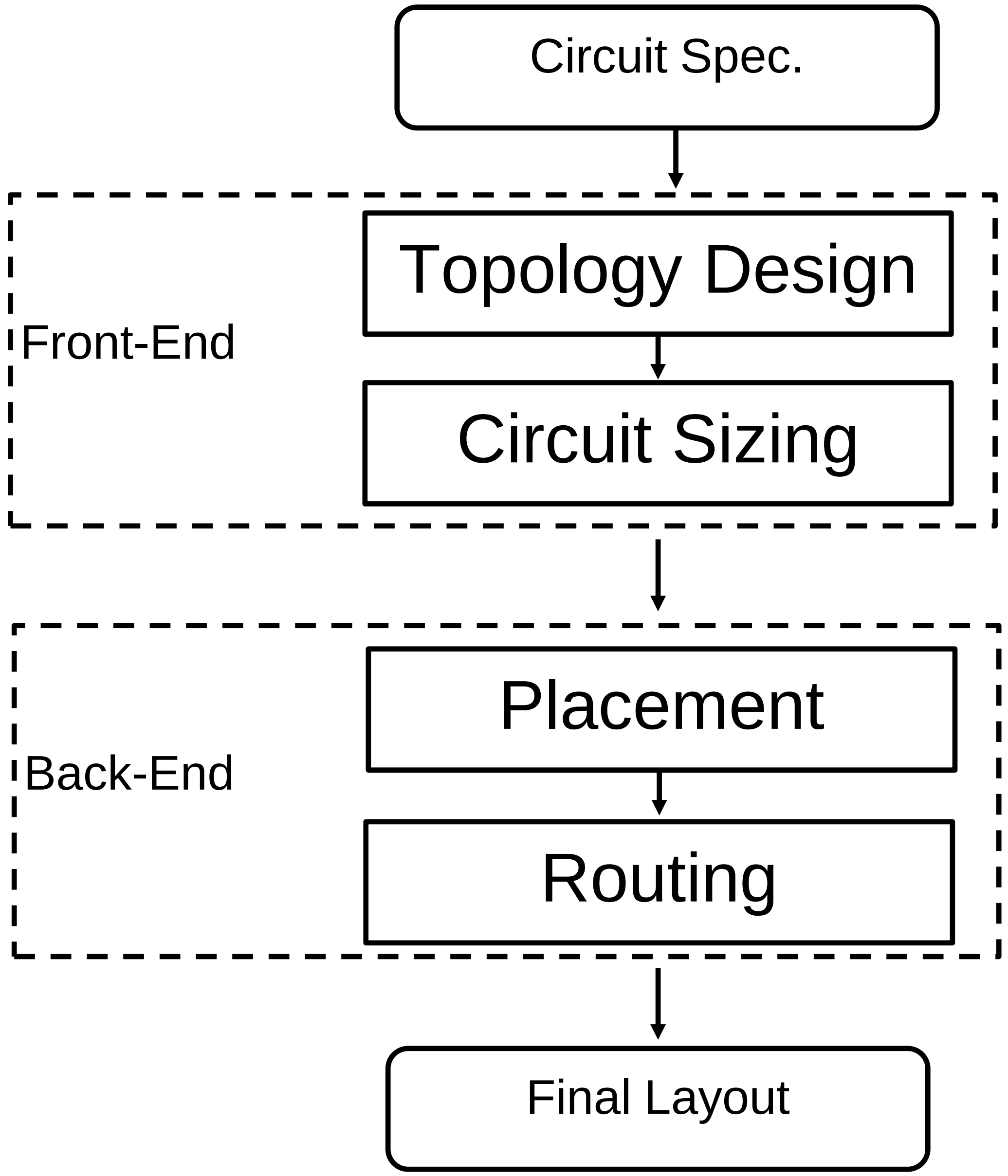
No packages published
Publish your first package

Contributors 4

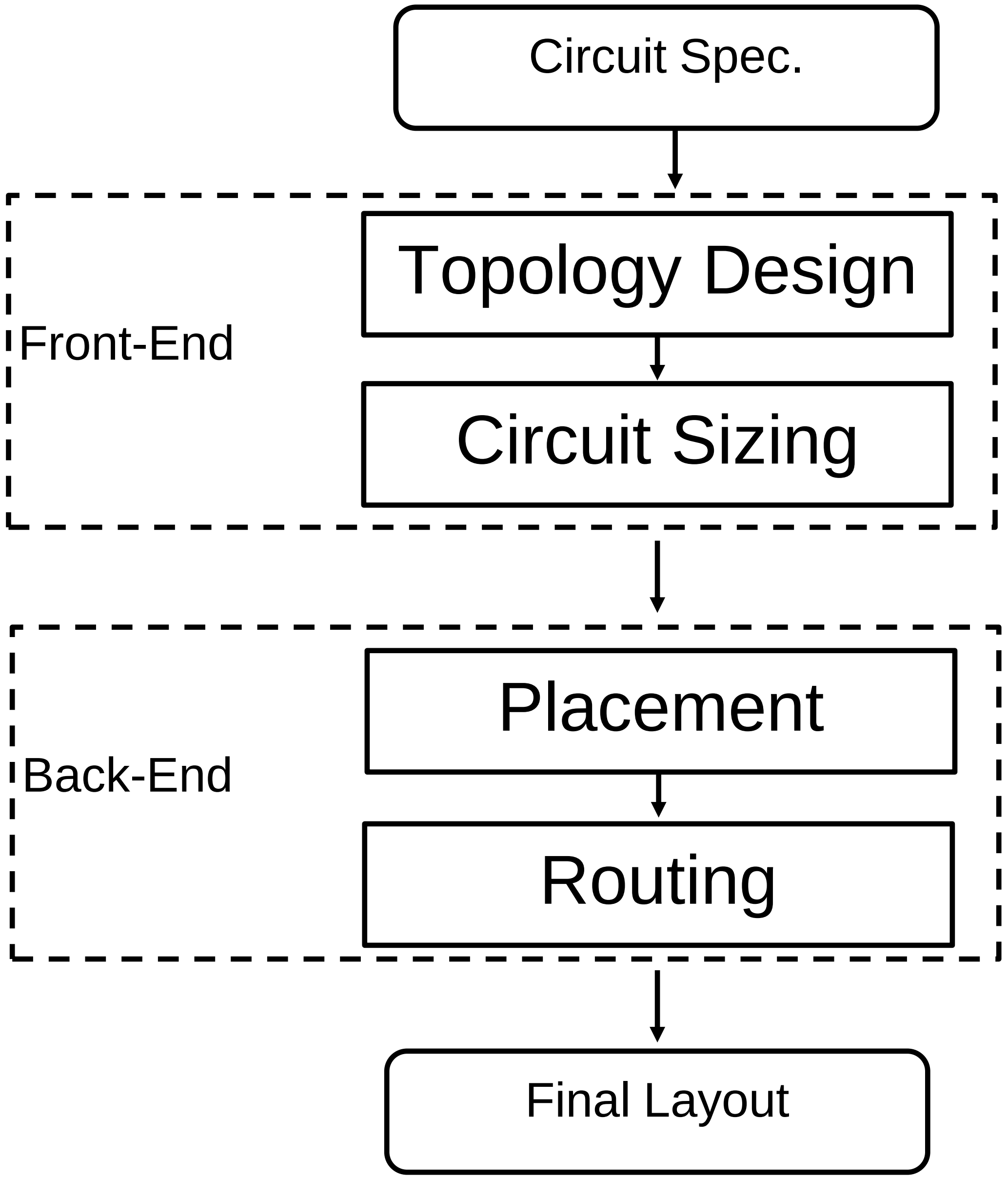
krzhu Keren Zhu
jayl940712 Mingjie Liu
limbo018 Yibo Lin
baloneymath Hao Chen

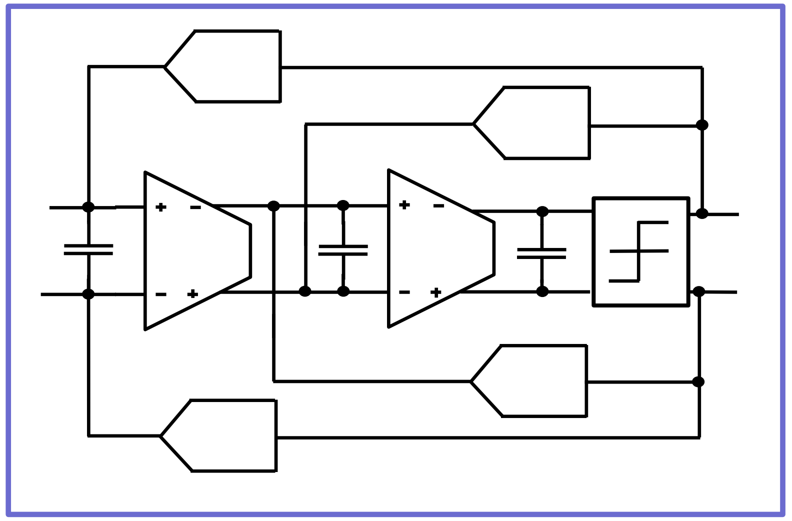
- Open-sourced
- Silicon-proven
- Fully-automated
- End-to-end

AMS Design Automation 101

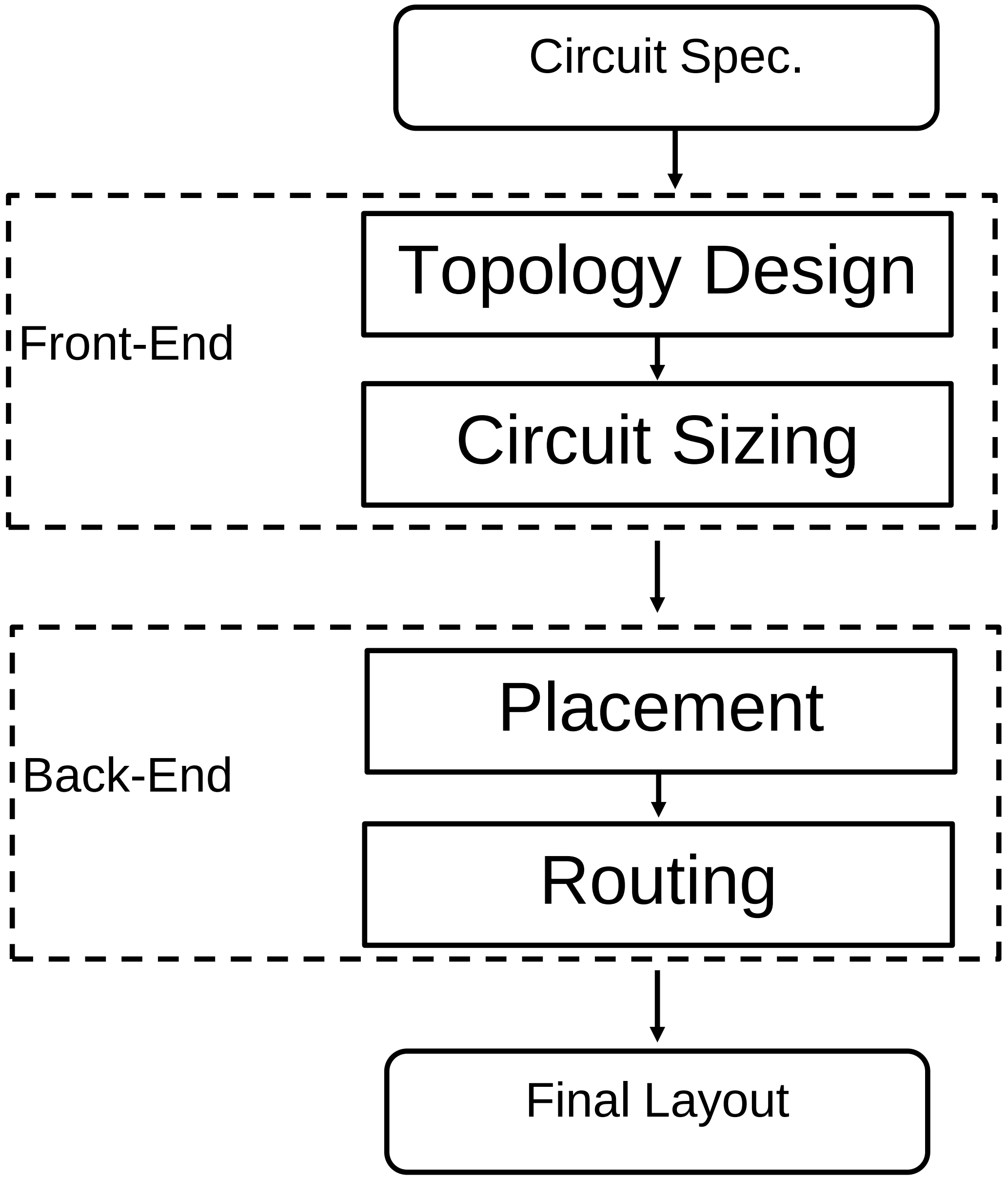


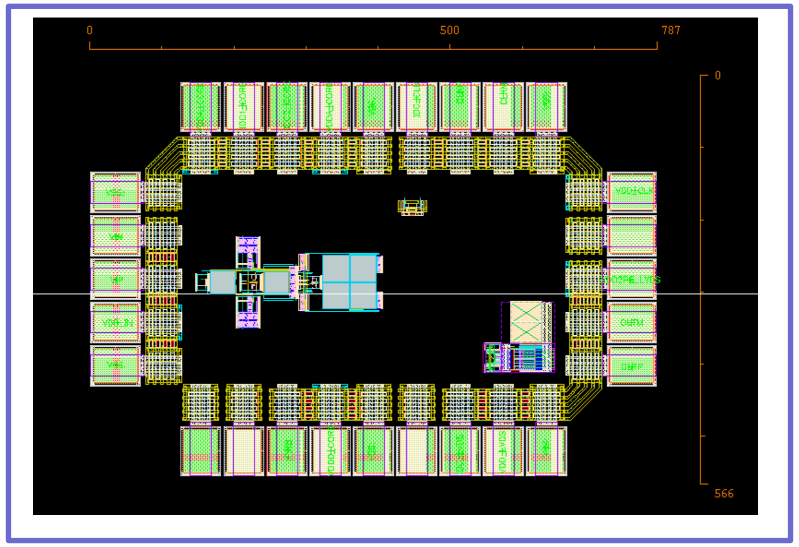
AMS Design Automation 101





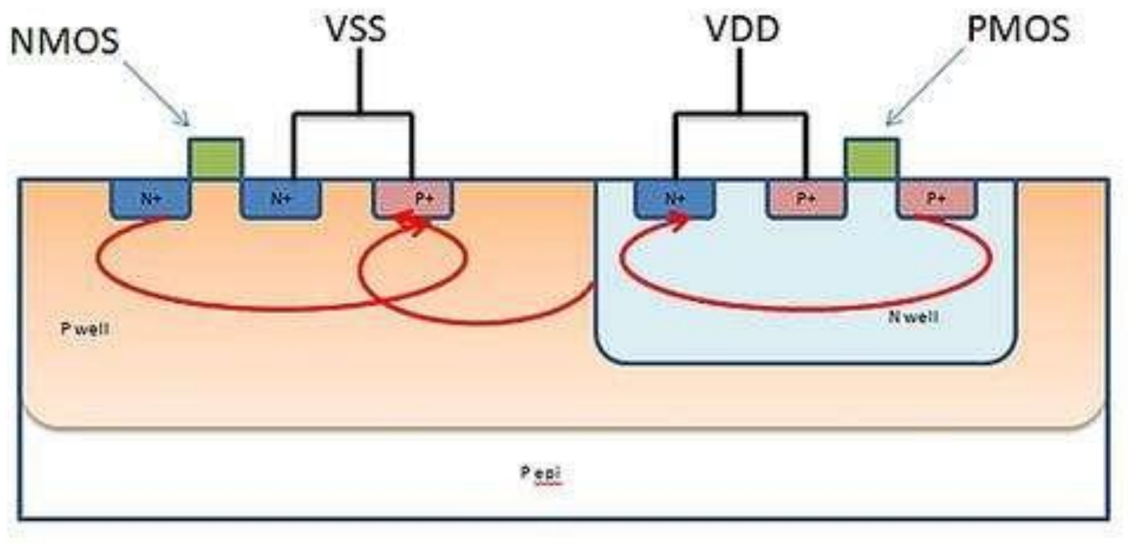
AMS Design Automation 101

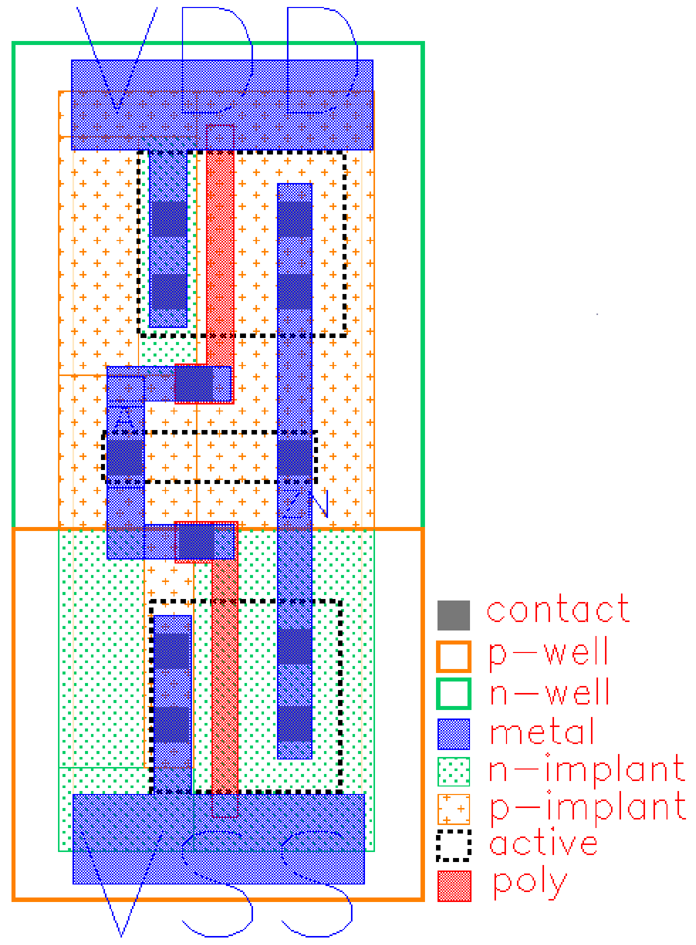




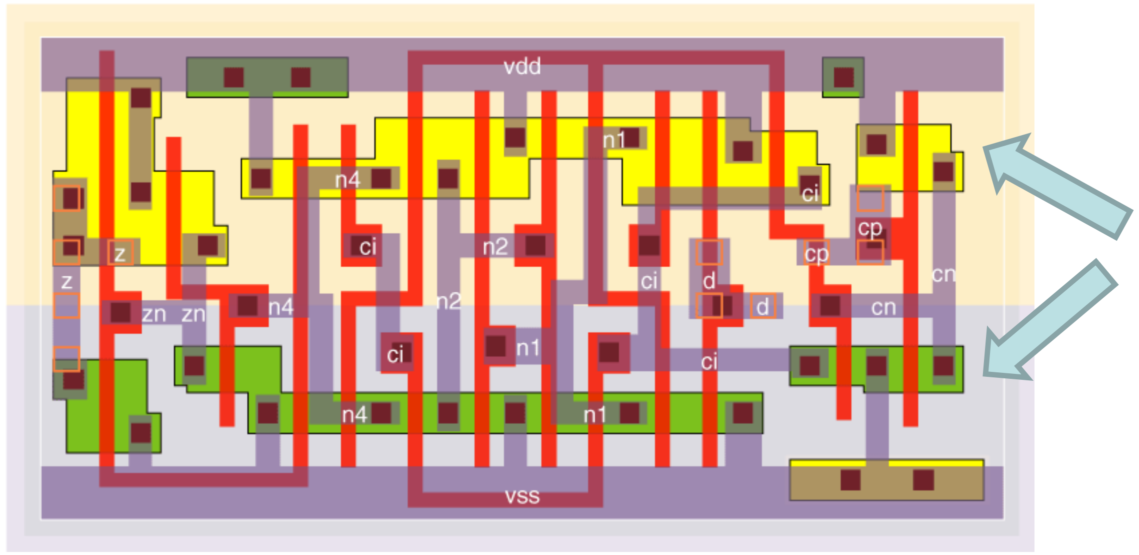
Background: Wells In Layouts

- Wells are essential to make transistor working





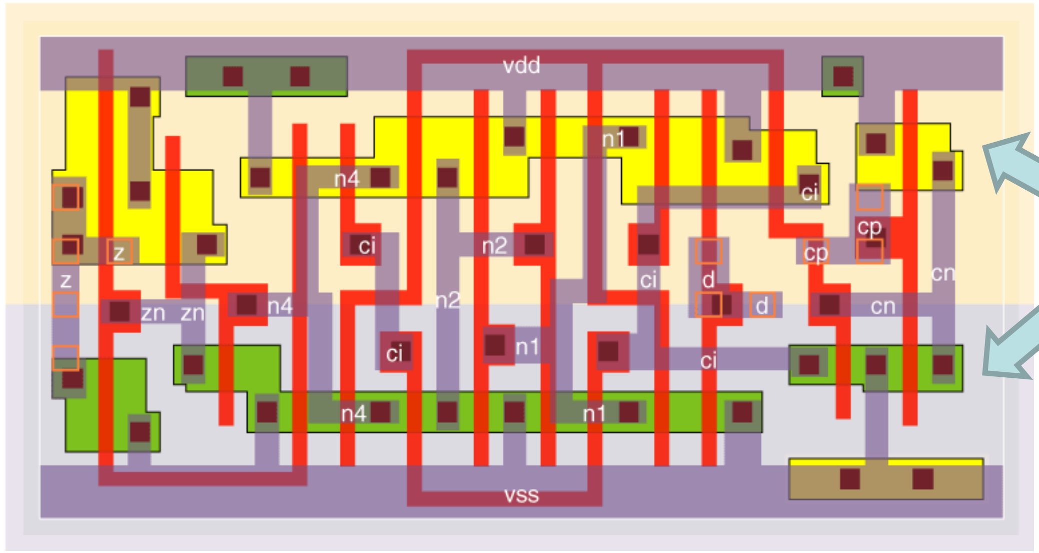
Background: Wells In Layouts



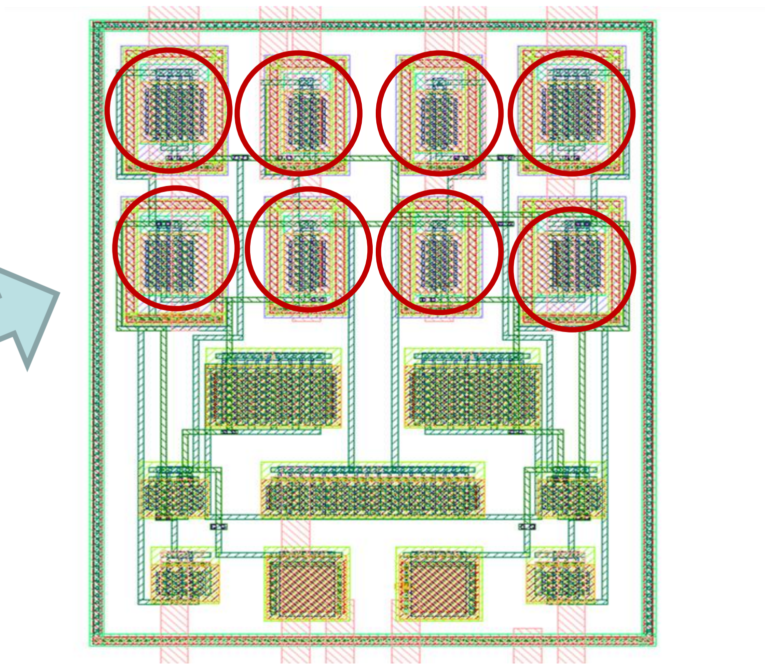
A digital stdcell

[Petley, *2008*]

Background: Wells In Layouts



Well



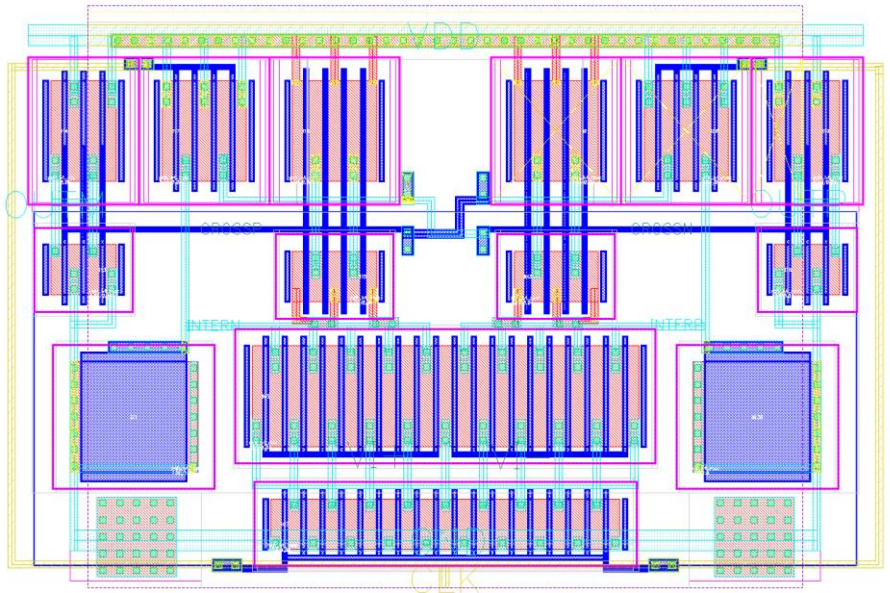
A digital stdcell

[Petley, *2008*]

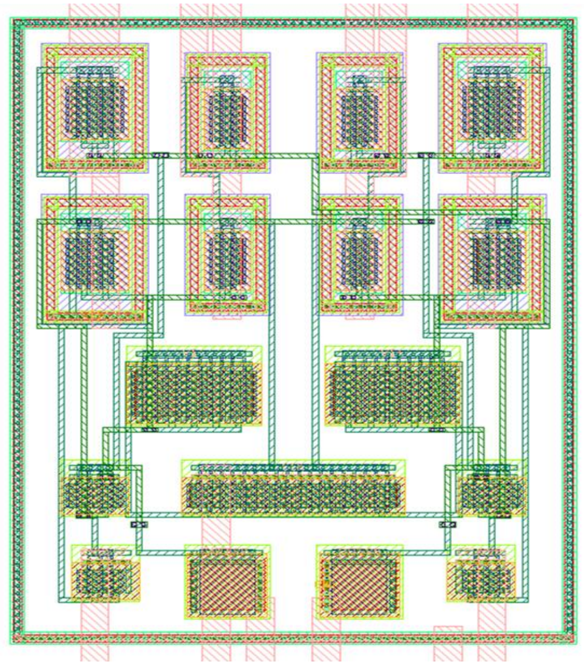
A comparator layout generated with MAGICAL

[Chen, *2021*]

Background: Wells In Layouts



The manual comparator layout



A comparator layout generated with MAGICAL

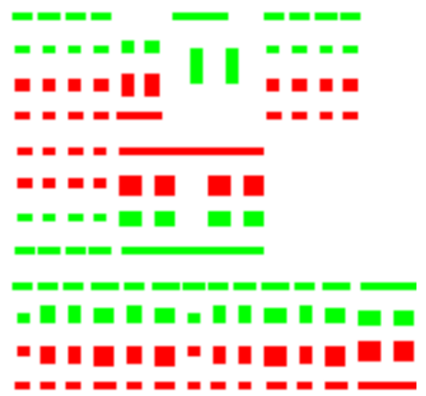
[Chen, *2021*]

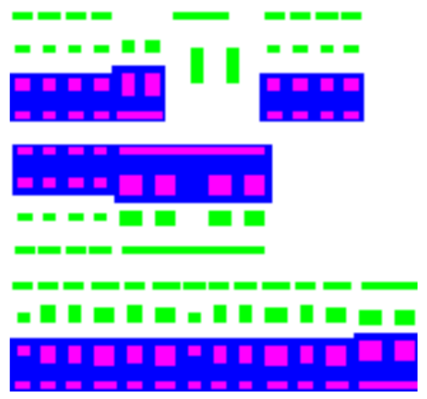
Background: WellGAN

- Generative adversarial network (GAN)-guided well generation

Background: WellGAN

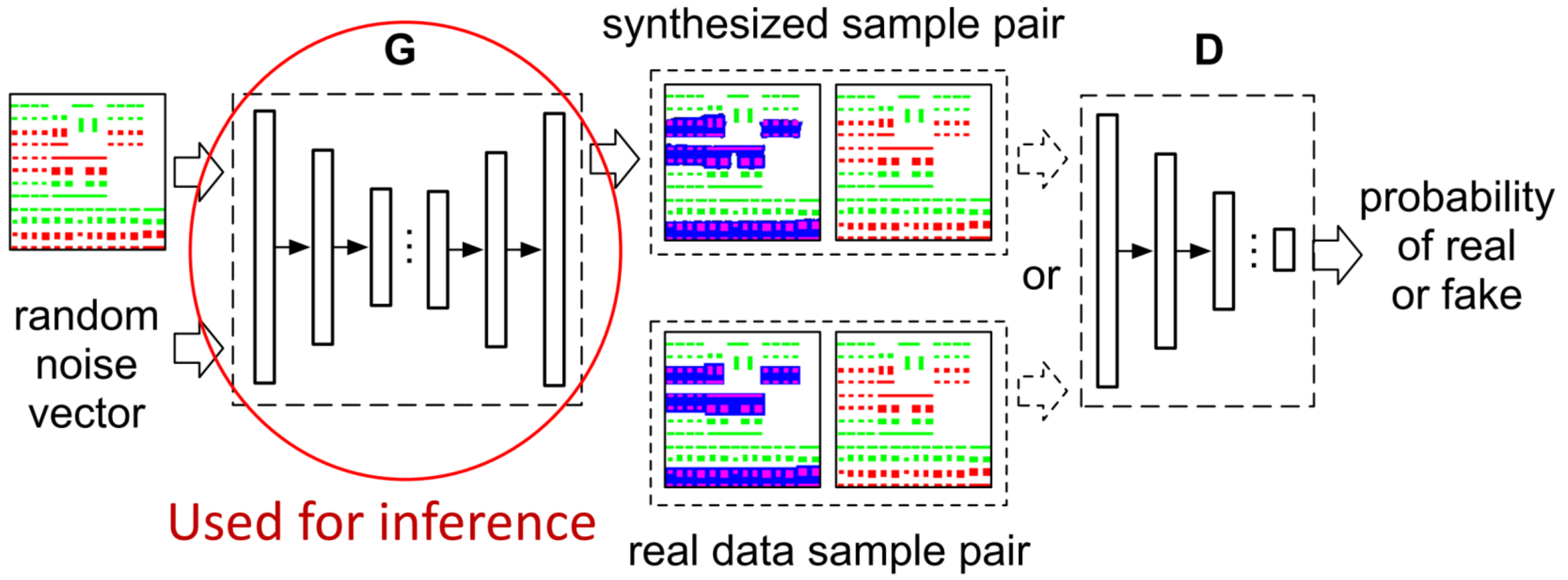
- Generative adversarial network (GAN)-guided well generation


Input


Output

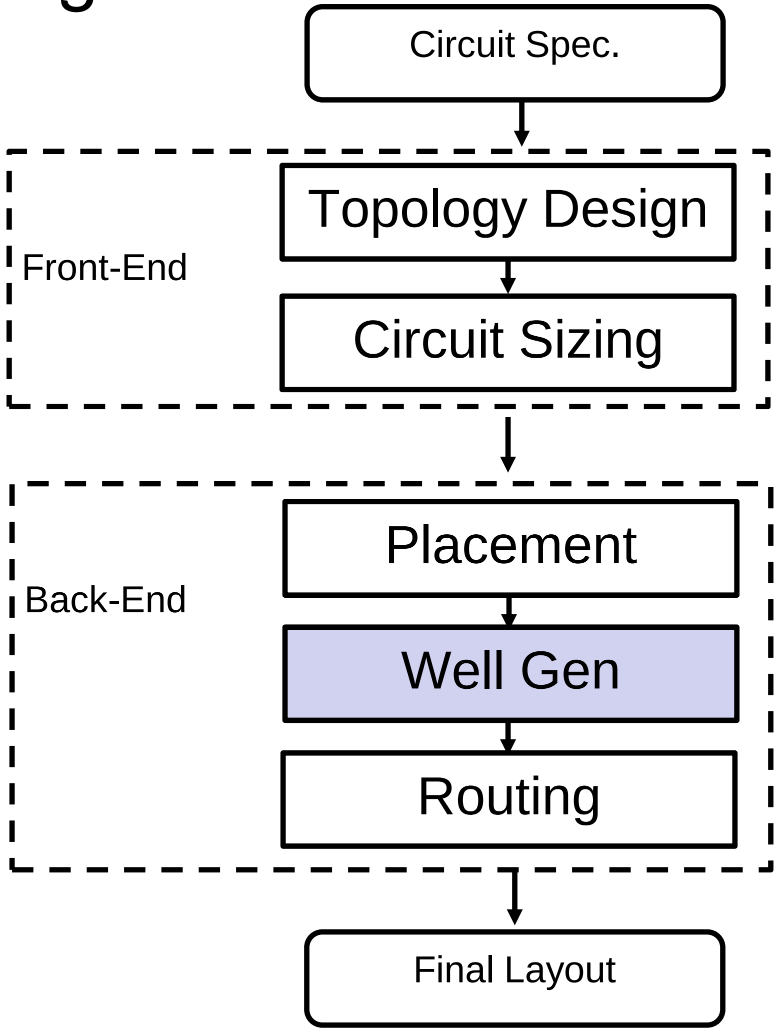
Background: WellGAN

- Generative adversarial network (GAN)-guided well generation



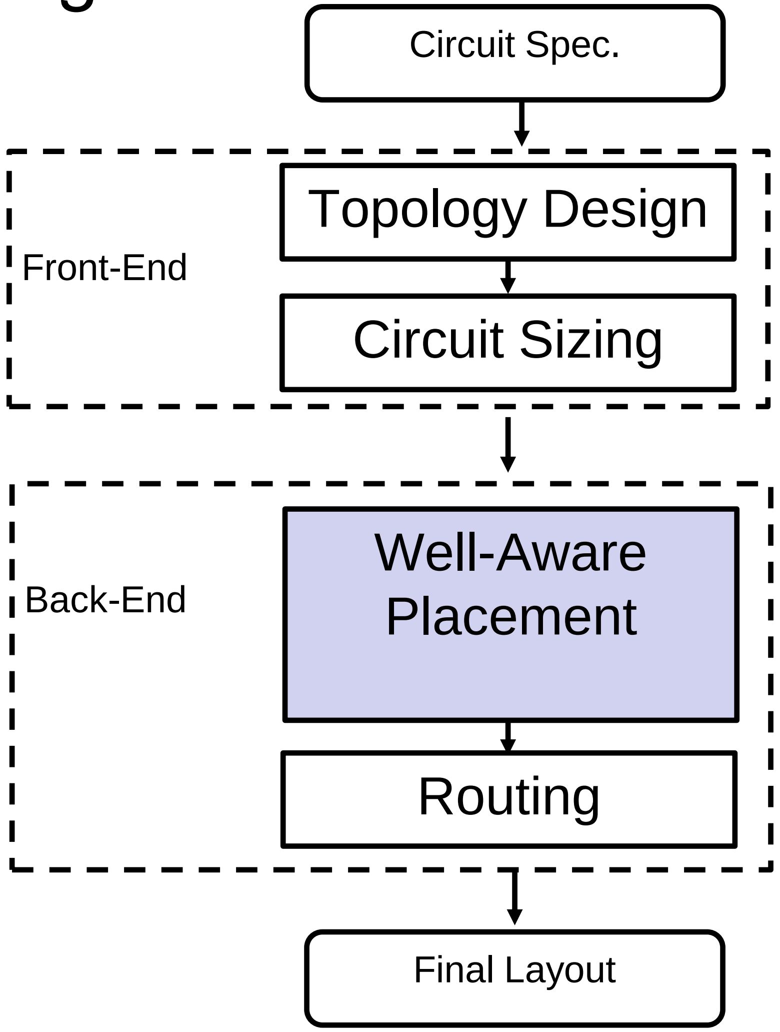
Background: WellGAN

- Making well generation as a separate stage…
  - Loss optimization opportunities
  - Might be infeasible

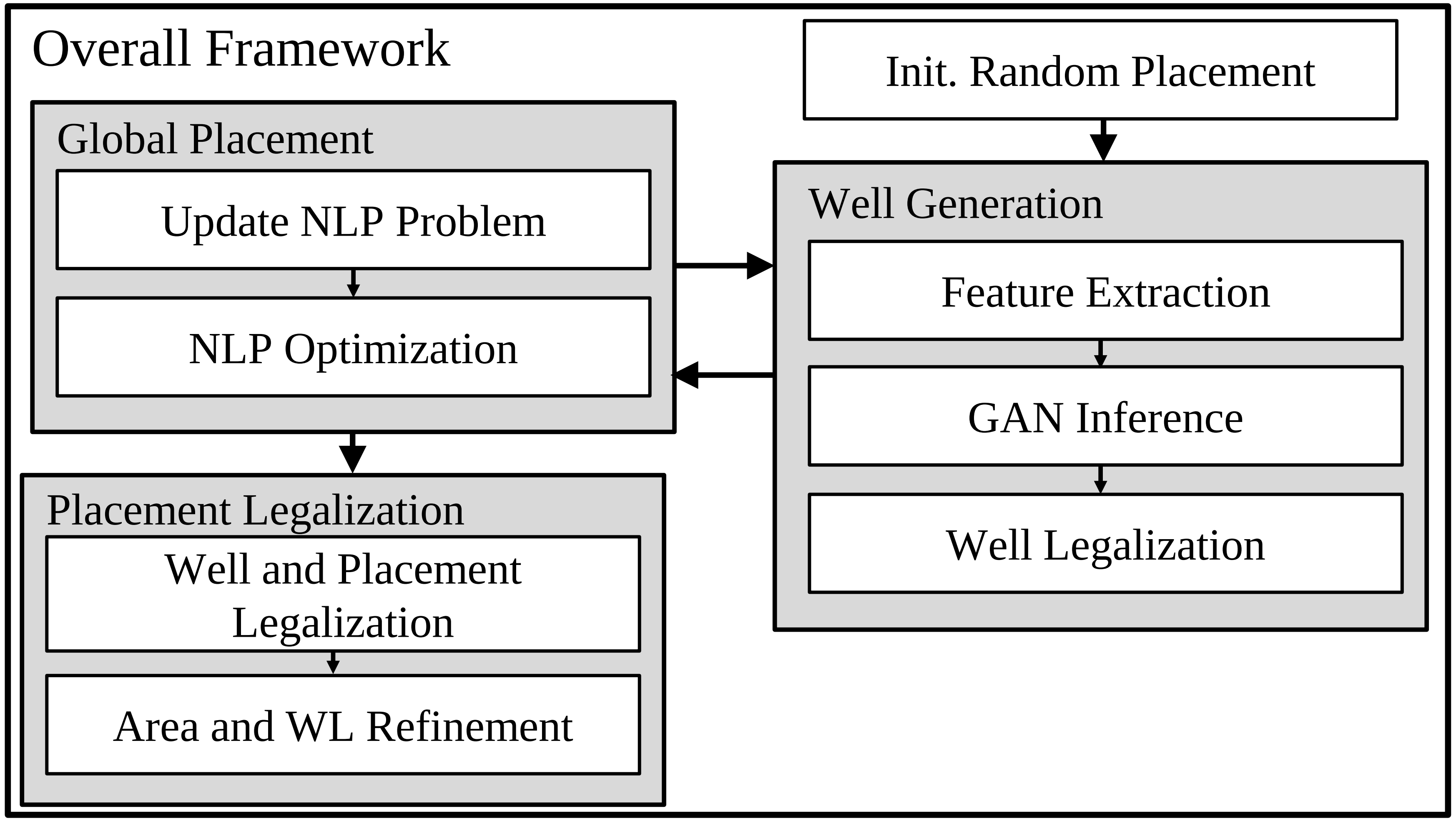


Background: WellGAN

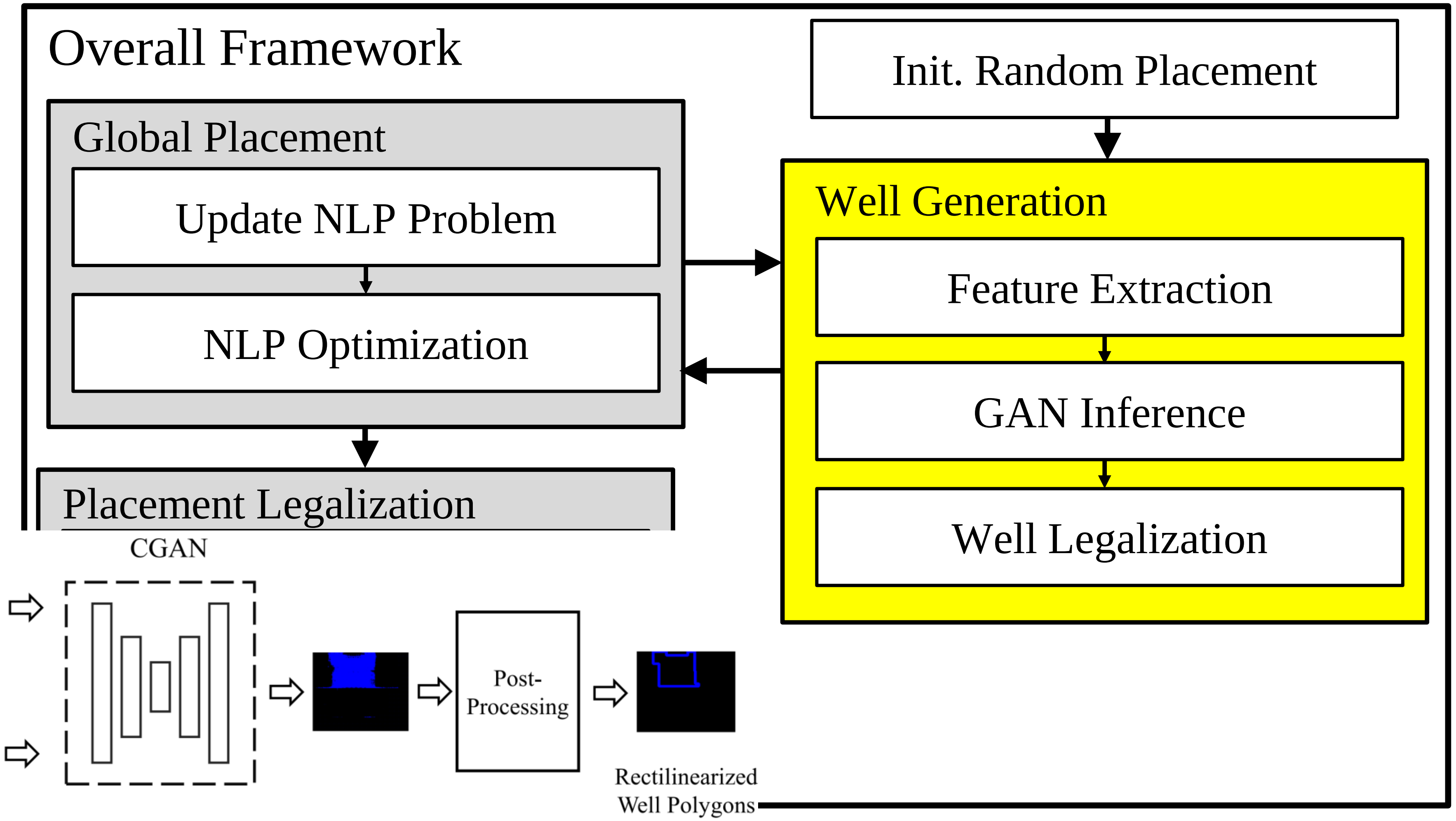
- Making well generation as a separate stage…
  - Loss optimization opportunities
  - Might be infeasible
- ML-guided well-aware placement
  - Fuse placement and well-generation
  - Optimization meets ML

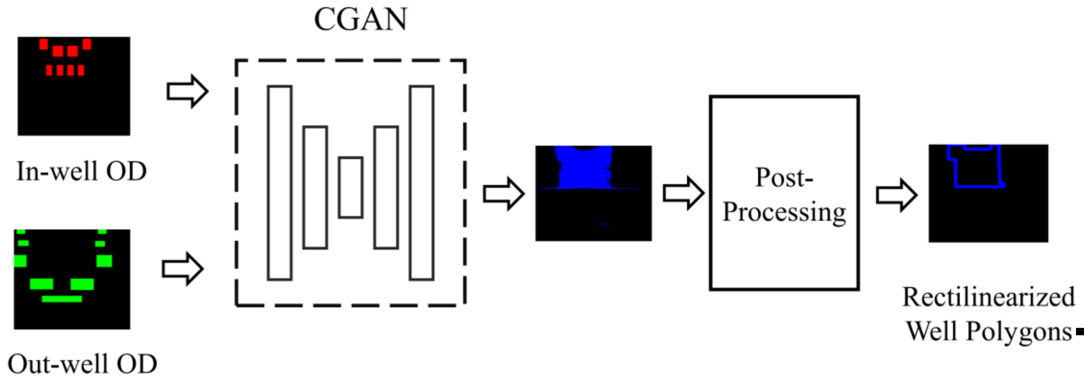


Well-Aware Placement Flow

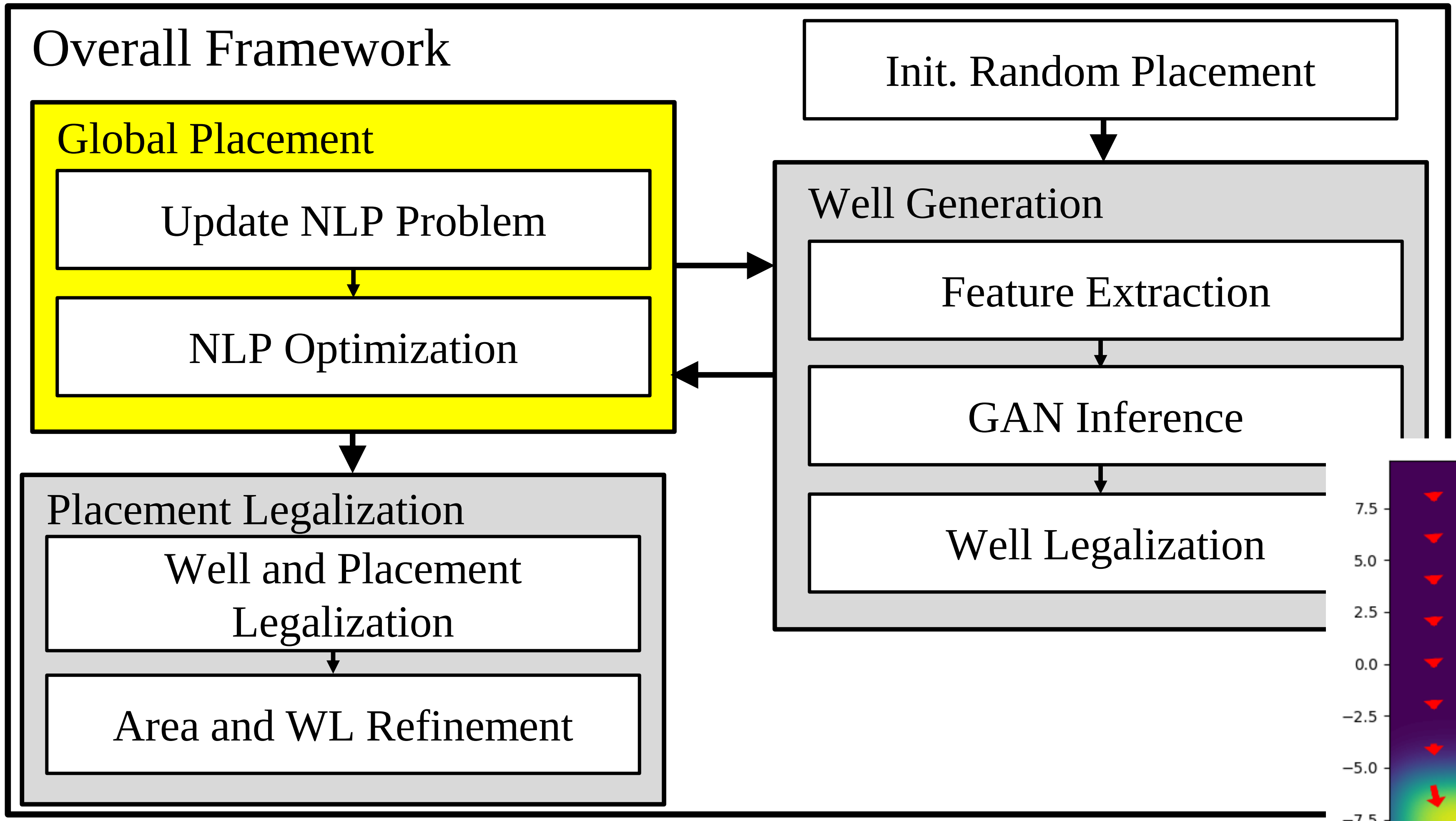


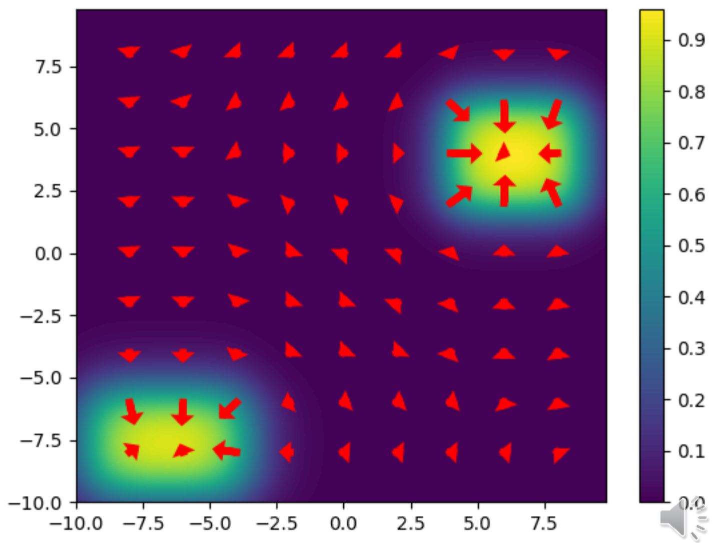
Well-Aware Placement Flow



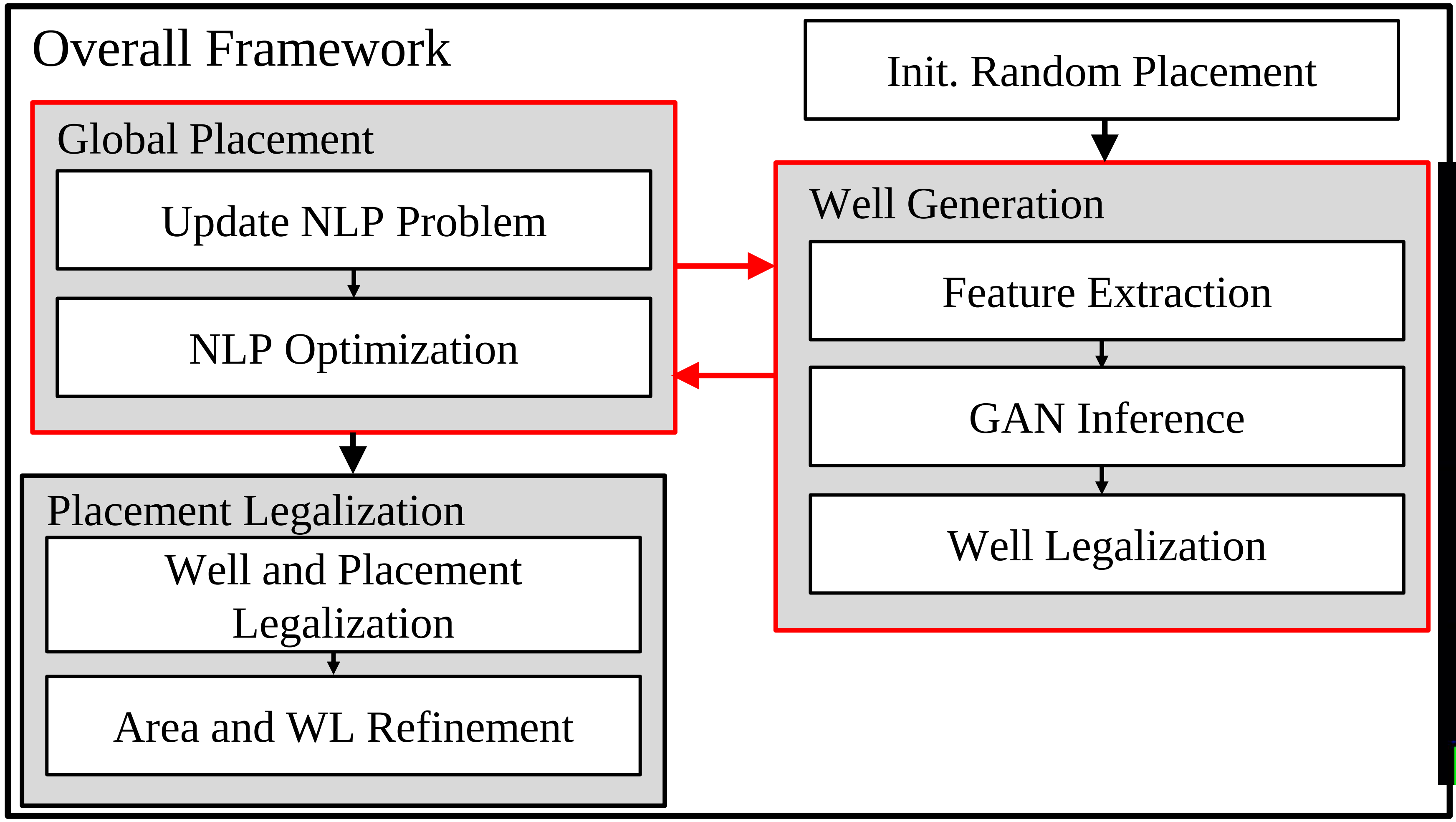


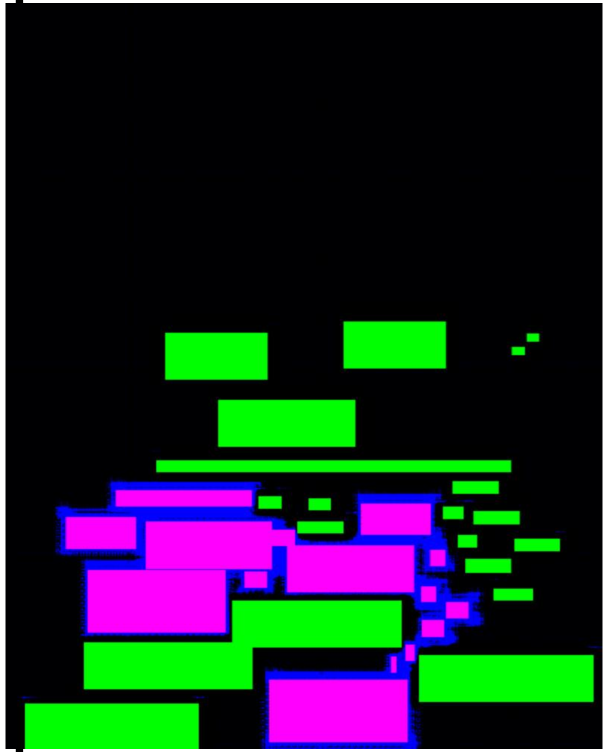
Well-Aware Placement Flow



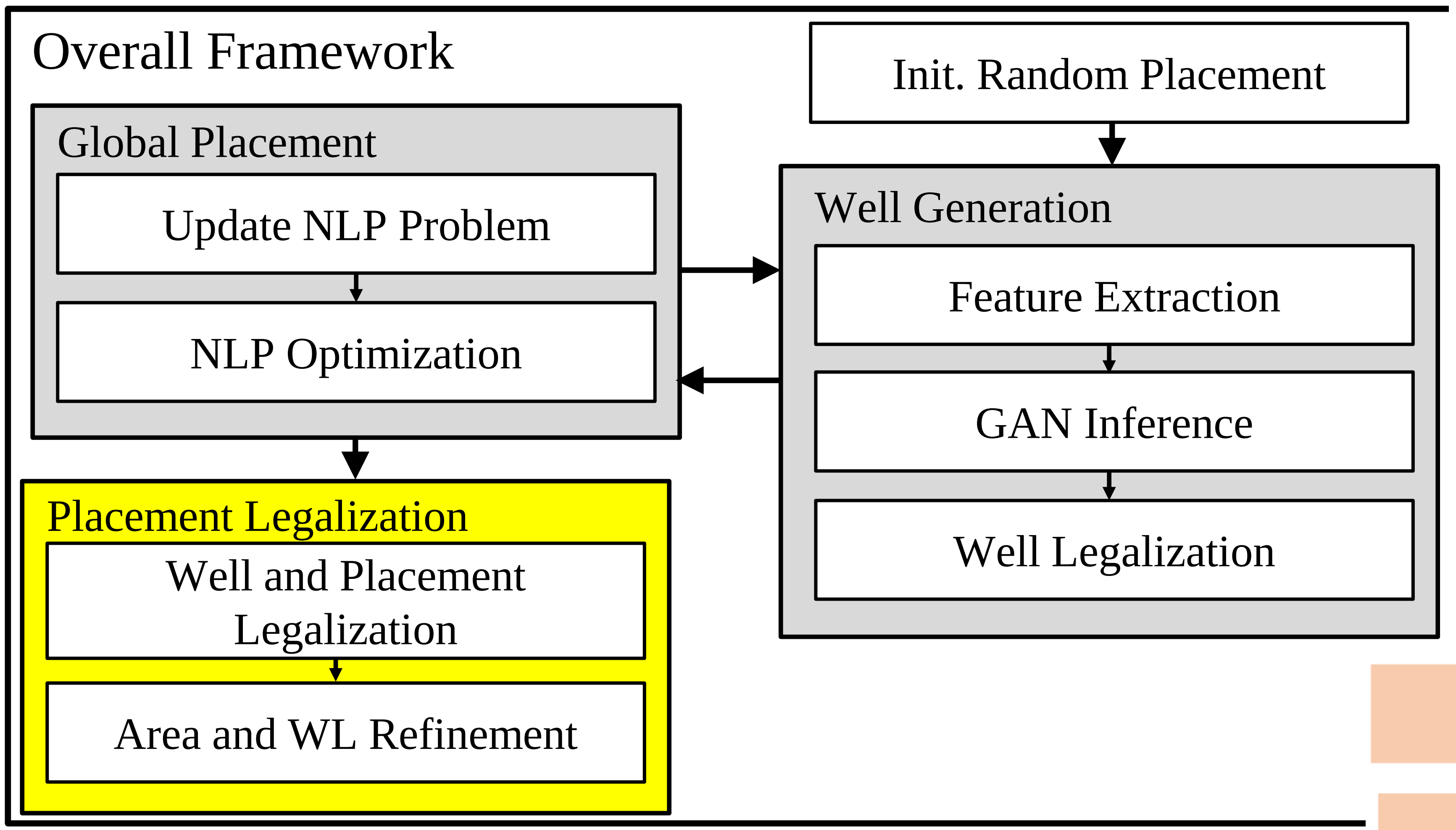


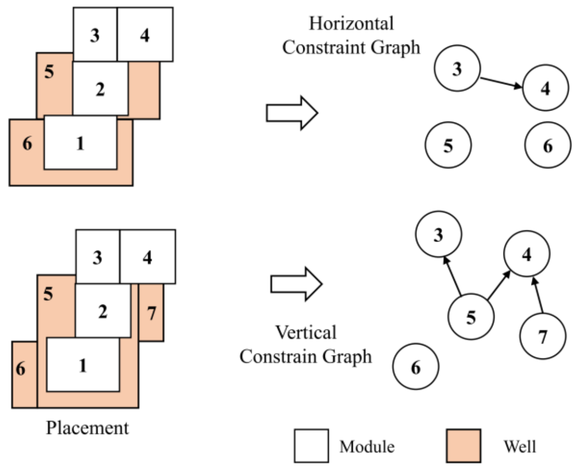
Well-Aware Placement Flow

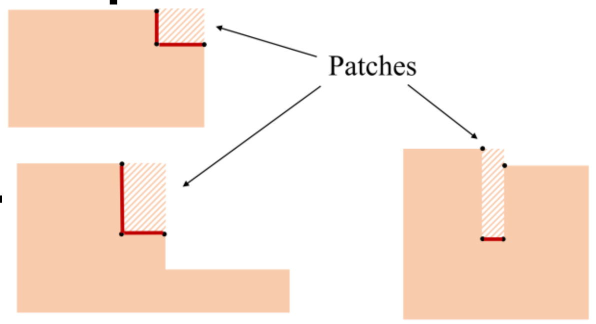




Well-Aware Placement Flow



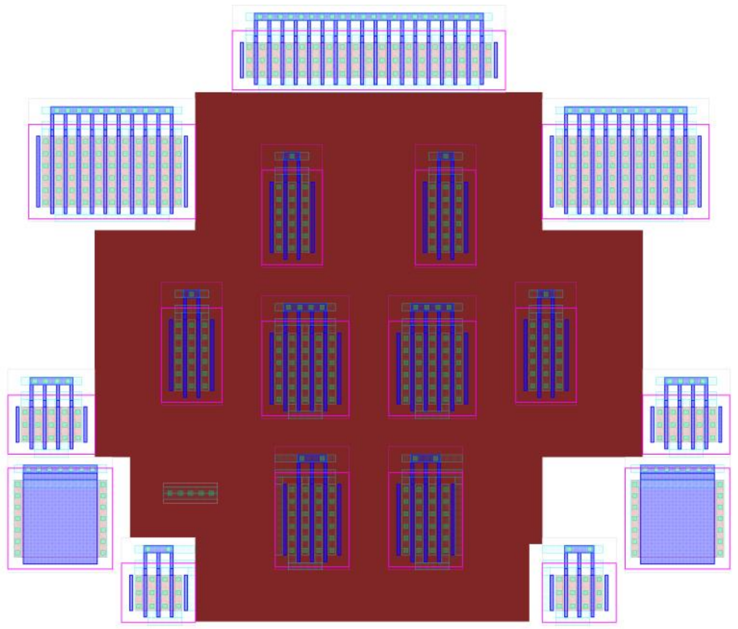


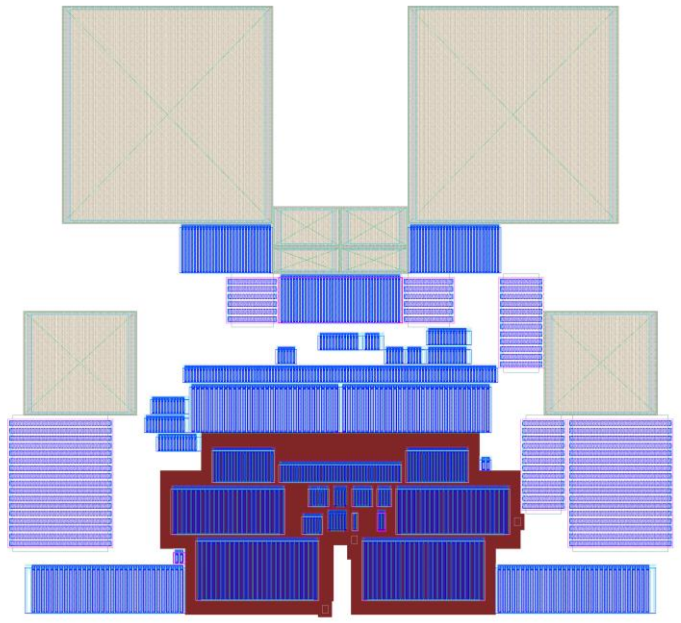


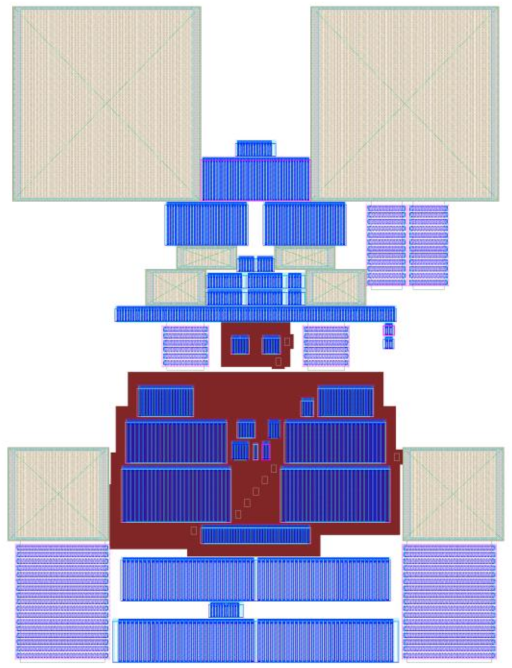
Experimental results on area, HPWL and Runtime

	Individual Wells			DAC’ 19			This Work		
Circuits	Area	HPWL	Runtime	Area	HPWL	Runtime	Area	HPWL	Runtime
OTA1	360.2	72.3	1.3	318.0	68.7	3.2	290.3	60.3	3.6
OTA2	756.2	234.7	4.8	750.7	203.1	7.9	599.0	205.2	10.6
OTA3	1055.4	586.6	48.9	1325.6	559.5	43.2	965.6	651.3	34.1
OTA4	3255.2	837.1	39.7	3313.6	799.6	40.1	3033.7	866.0	42.6
COMP1	175.1	78.8	2.0	144.4	95.1	6.6	82.2	61.8	3.5
COMP2	192.2	93.1	3.0	194.2	105.0	5.6	84.7	48.1	3.6
BOOTSTRAP	177.9	64.5	2.0	130.8	83.4	5.0	97.5	63.2	4.8
RDAC	361.5	209.2	12.4	370.4	287.0	30.2	144.3	137.0	23.7
Norm.	1.82	1.26	0.64	1.74	1.46	1.33	1.00	1.00	1.00

Experimental result: layouts







Conclusion

- A new ML-guided well-aware placement framework
- Integration of machine learning guidance with physical optimization techniques
- Experimental results show significant reduction in area and HPWL

Thanks
Q&A