



Revolutionize 3D Chip Design with Open3DFlow, An Open-source AI-enhanced Solution

Dr. Lei Ren

RISC-V International Open Source Laboratory (RIOS)
Tsinghua University



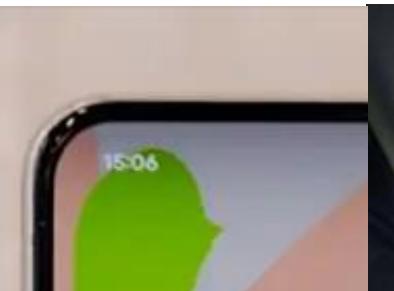
Introduction

- Applications drive system performance demand

HPC



Mobile Devices

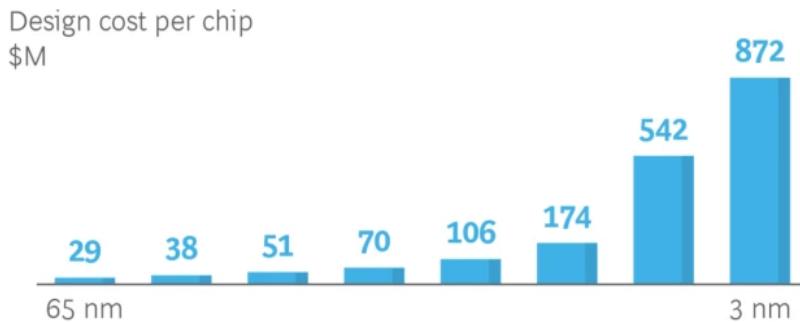


Autonomous Car

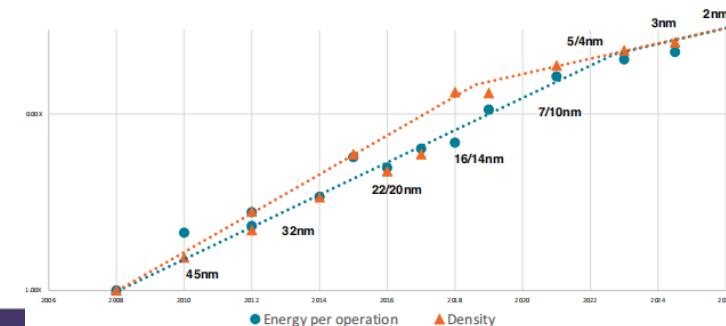


...

- Moore's Law slows down & Design costs explode as nodes get smaller



Technology Trends for Logic

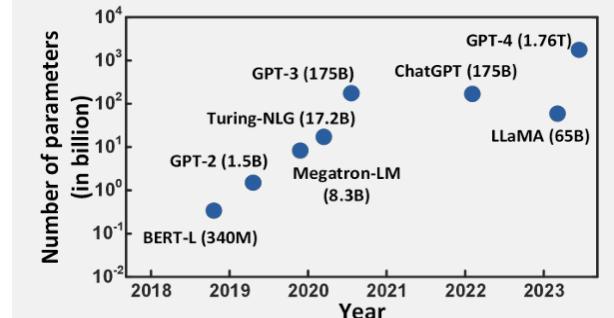
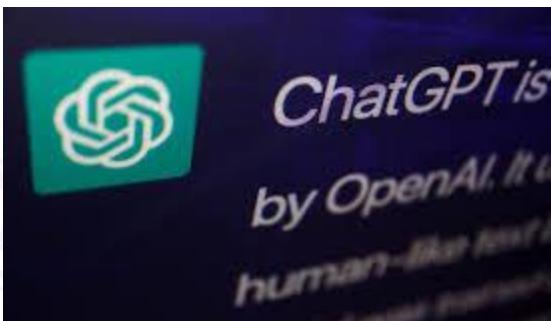




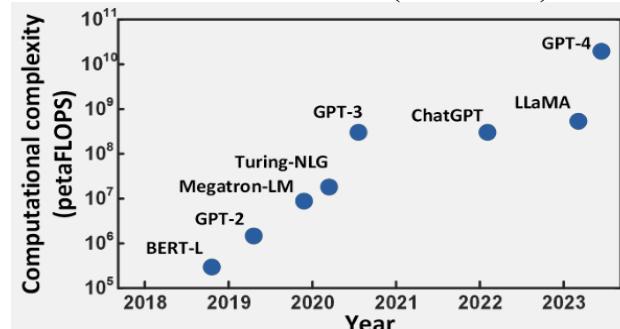
Introduction

- Generative AI Drive Compute and Bandwidth Performances

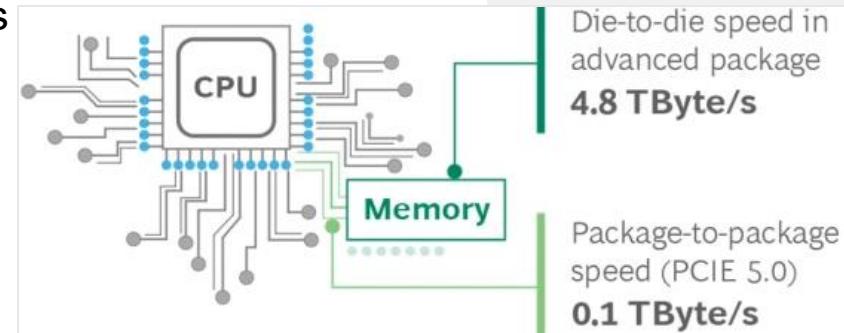
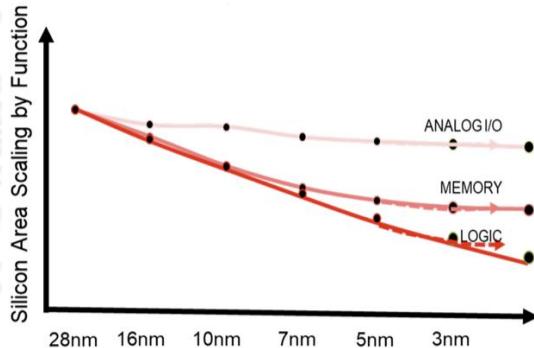
Model parameters: 340M → 1.76T (5000X)



Model compute throughput: 105 → 1010 Peta FLOPS (100000X)



- Limited and Divergent Scale Factors



Source: Wei Li, VLSI2024

- Integrating these components into a single SOC may not be cost-effective !



Chiplets & 3D IC

- Chiplets: Heterogeneous Integration

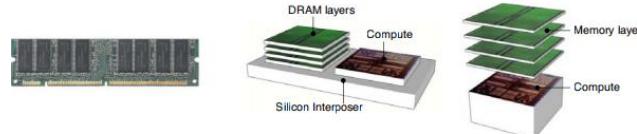
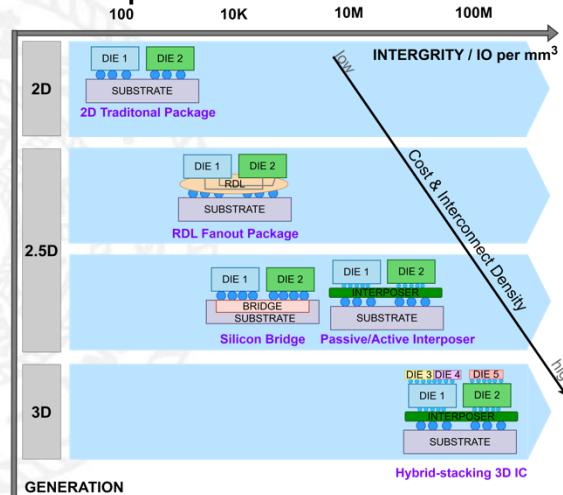
Small -> yield rate

Disaggregate -> low cost

IP reuse

- Bandwidths and energy efficiency between chiplets continues to increase!

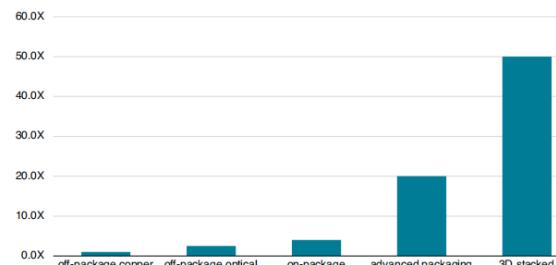
- Chiplet Interconnect Trace



Integration Enables Higher Bandwidth at Lower Power

pJ/bit	DIMMs	2.5D Micro-bumps (HBM)	3D Hybrid Bond
	~12	~3.5	~0.2

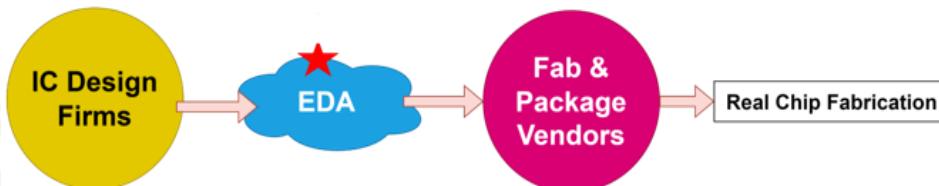
Relative Bits/Joule



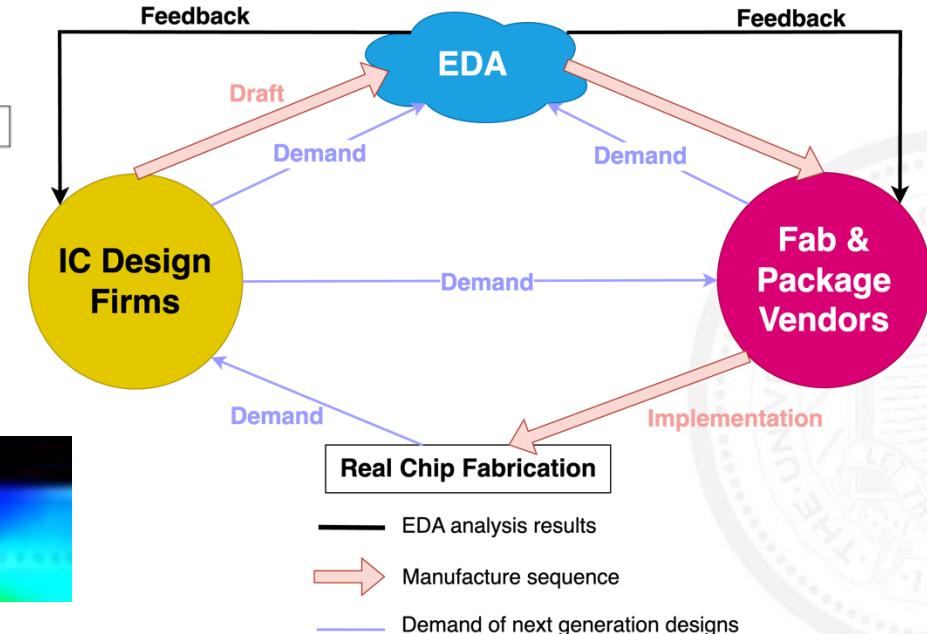
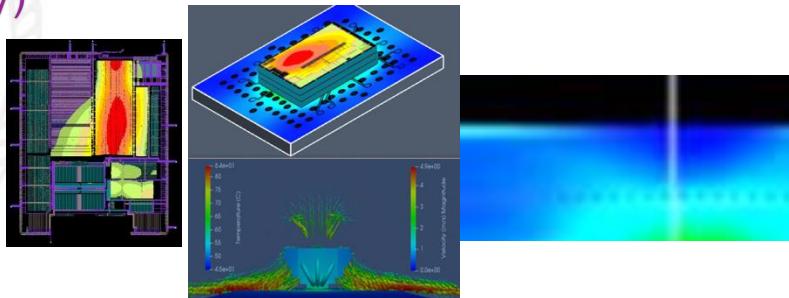
Source:
Wang et al, VLSI 2024

3D EDA Challenges

- How to realize 3D IC ?
- Electronic Design Automation (EDA): the glue between chip design and fabrication



- Toolchains before fabrication (code -> entity)

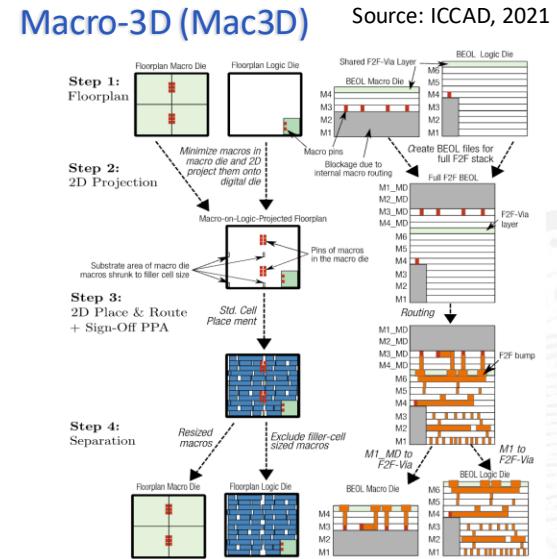
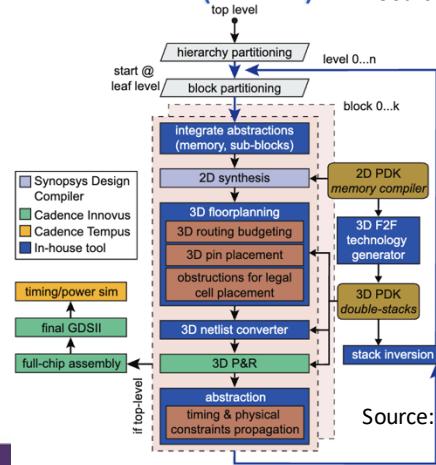
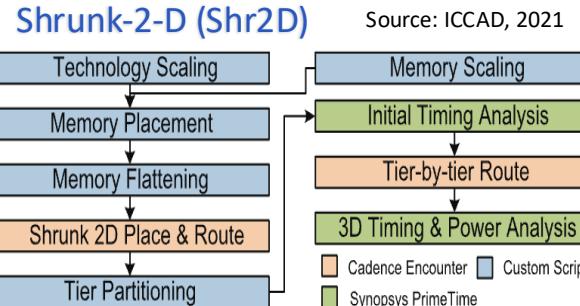
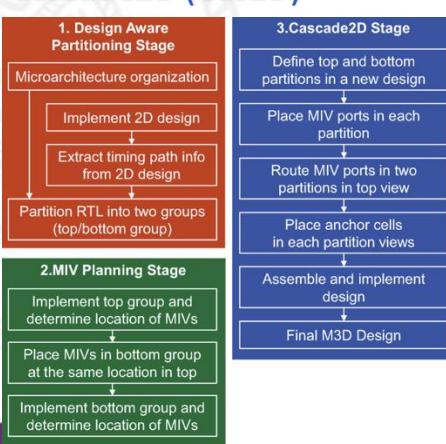
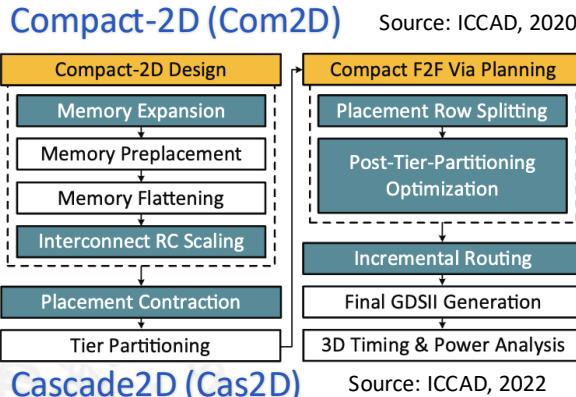


- Mutual iterative adaptation: closed loop ecosystem



3D EDA Challenges

- Academic 3D Flow



Monolithic-3D design flow

- Gap between real fabrication
 - Proprietary 2D EDA tool -> hard to reproduce
 - Without multi-physics post-process analysis

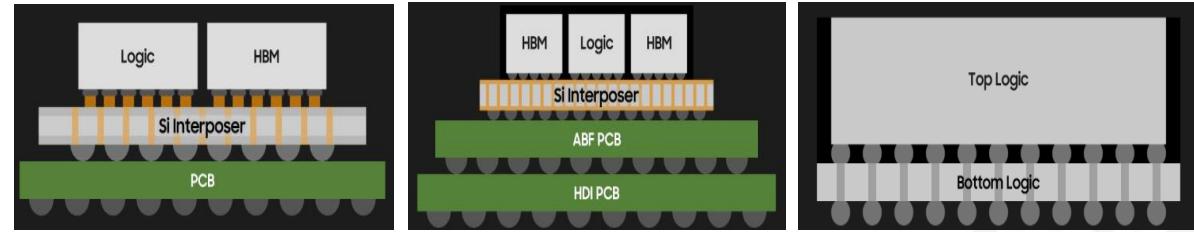


3D EDA Challenges

- Industrial 3D EDA Flow
 - Synopsys



3DIC Compiler
The Industry's Only Unified Exploration-to-Signoff Platform for 2.5D and 3D Multi-Die Designs



- TSMC 3DFabric™

TSMC 3DFabric™

- Cadence



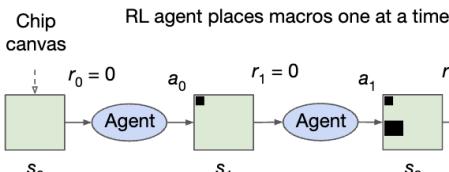
- Immature, Expensive, Blackbox, ...
- No “off-the-shelf” 3D chiplets -> hinder collaboration

Chip Stacking (FE 3D)

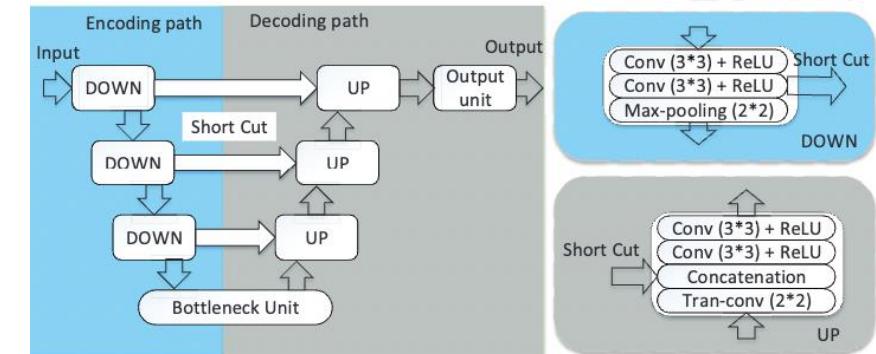
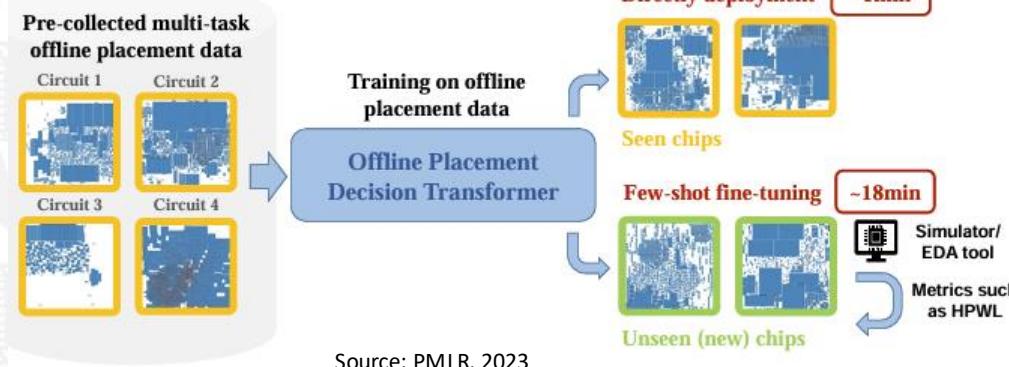
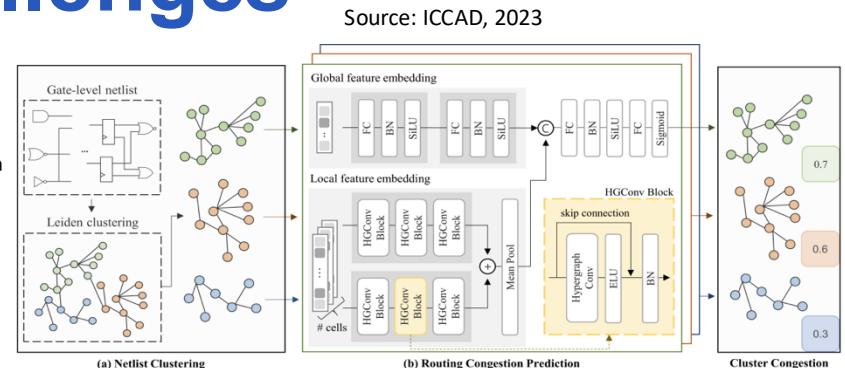
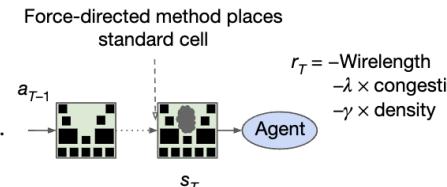


3D EDA Challenges

- Artificial Intelligence (AI) in EDA



Source: Nature, 2019



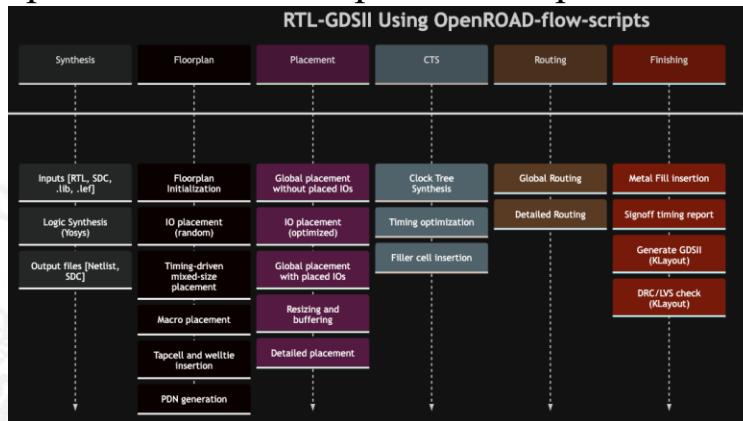
❖ When converting to 3D cases ...

- New considerations: Performance/Power/Area/Thermal (PPAT^T)
- New process method when adding a new dimension



Open EDA Ecosystem

❖ OpenEDA toolchain platform: OpenRoad

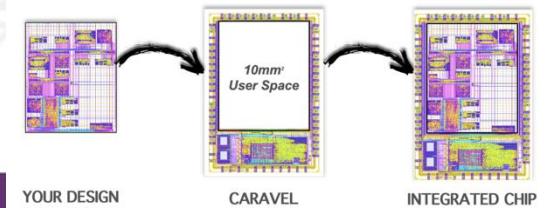


<https://github.com/The-OpenROAD-Project/OpenROAD>

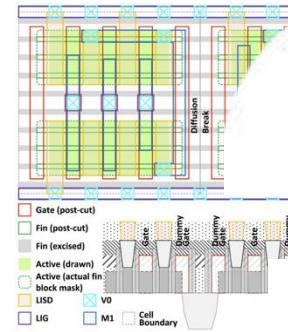
❖ Open Tapeout Project

Caravel - Full Chip Open-source Design

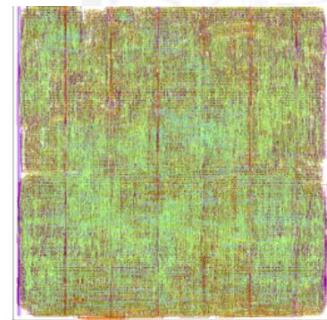
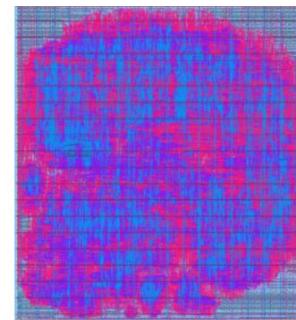
Integrated you project into full chip providing power, configurable IOs and a RISC-V management SoC.



❖ OpenPDK

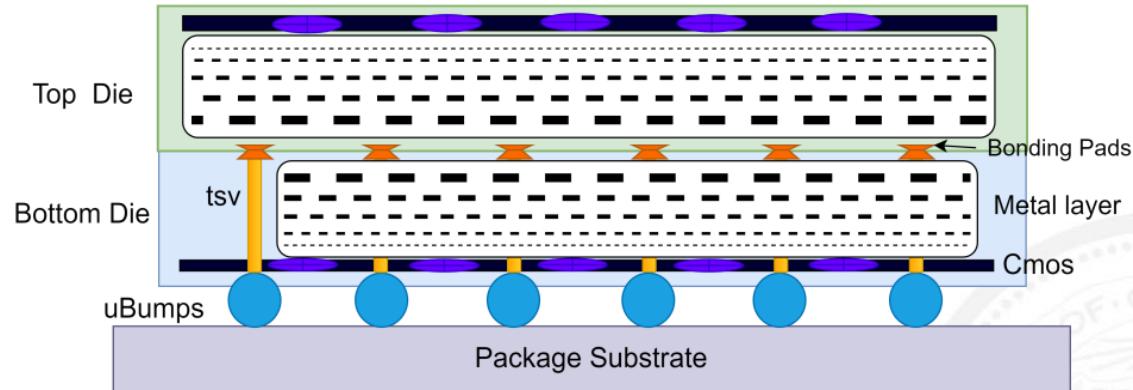


- Comparative results



Open 3D EDA Platform: Open3DFlow

- Target Design
 - Hybrid bonding (HB) F2F
 - TSVs for power & signal transmission
 - Thermal & SI analysis
 - Industrial-like strategy



	Tier Partition	Target Architecture	Holistic BEOL	Physical Modeling	Other Analysis	EDA Tools
Industrial Flow	First	3D IC with HB and TSVs	X	✓	Thermal, Stress, PI, SI	Proprietary
Shrunk-2-D	Last	M3D	X	X	\	Proprietary
Cascade2D	First	M3D	X	X	\	Proprietary
Compact-2D	Last	TSV-based or M3D	X	X	\	Proprietary
Macro-3D	First	3D IC with F2F Bumps	✓	X	\	Proprietary
Sequential-2D	First	3D IC with HB	X	X	\	Proprietary
Hier-3D	First	3D IC with F2F Bumps	✓	X	\	Proprietary
Open3DFlow	First	3D IC with HB and TSVs	X	✓	Thermal, SI	Open-source

Fully open-source: <https://github.com/b224his1/Open3DFlow>

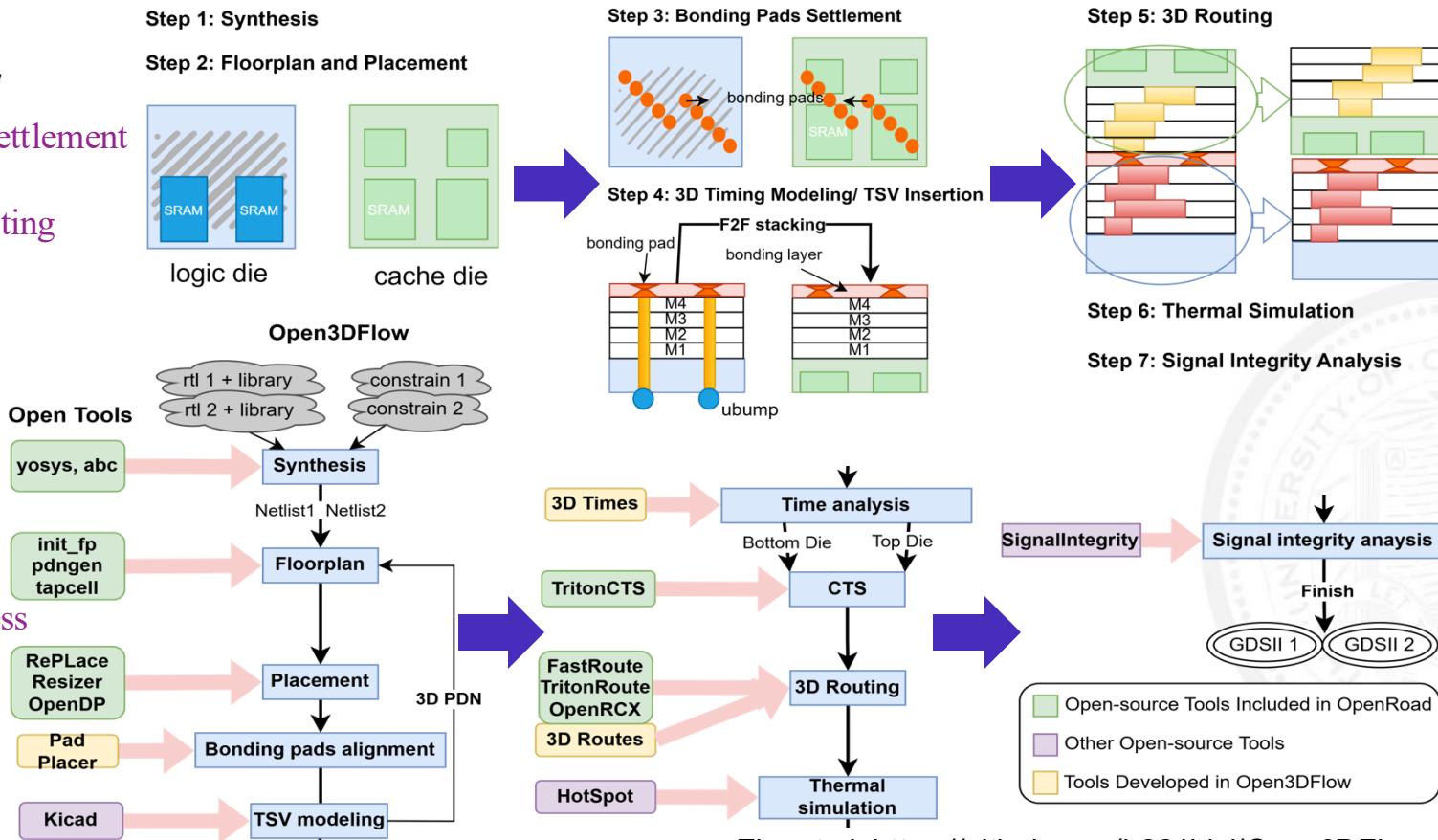


Open 3D EDA Platform: Open3DFlow



- 7-step workflow

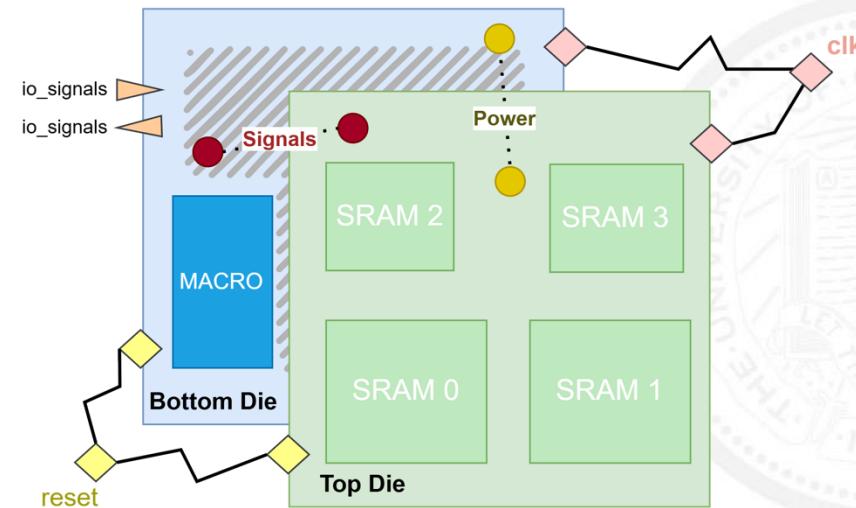
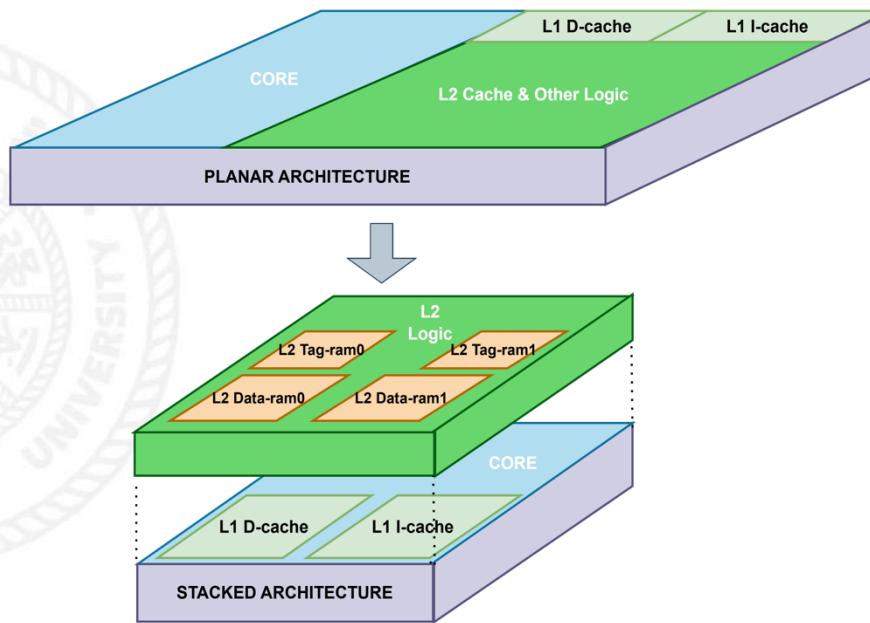
- HB bonding pads settlement
- TSV Modeling
- Simultaneously routing
- Thermal
- SI





Open 3D EDA Platform: Open3DFlow

- Proof-of-concept Design
 - OpenPDK: GF180 & Sky130
 - Top Die: 1KB unified L2 Cache
 - Bottom Die: 32-bit 5-stage CPU Main Logic + 512byte L1 D&I-Cache



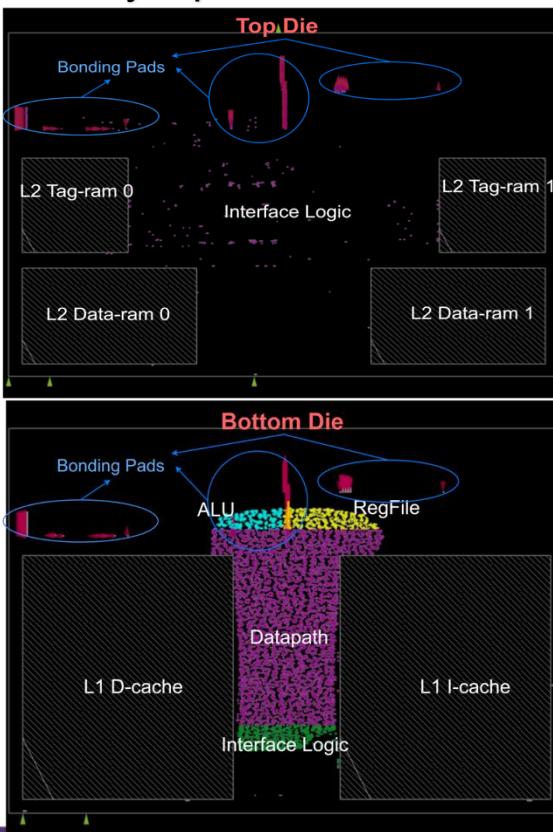


Open 3D EDA Platform: Open3DFlow



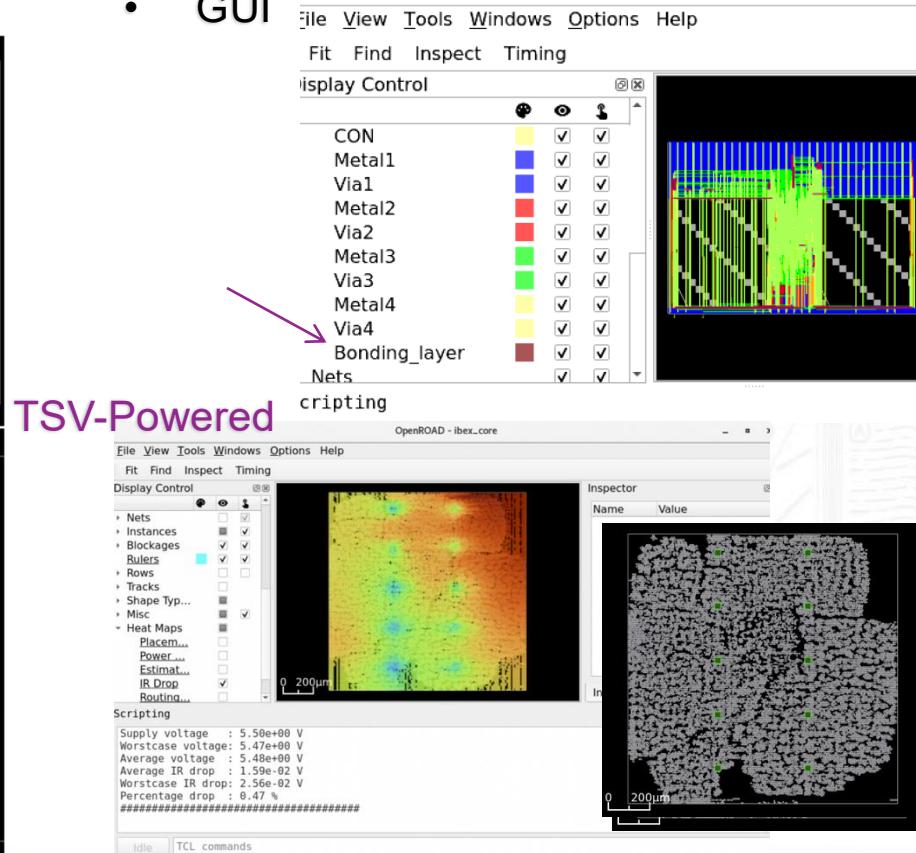
- Die images generated by Open3DFlow

DRC-clean GDS



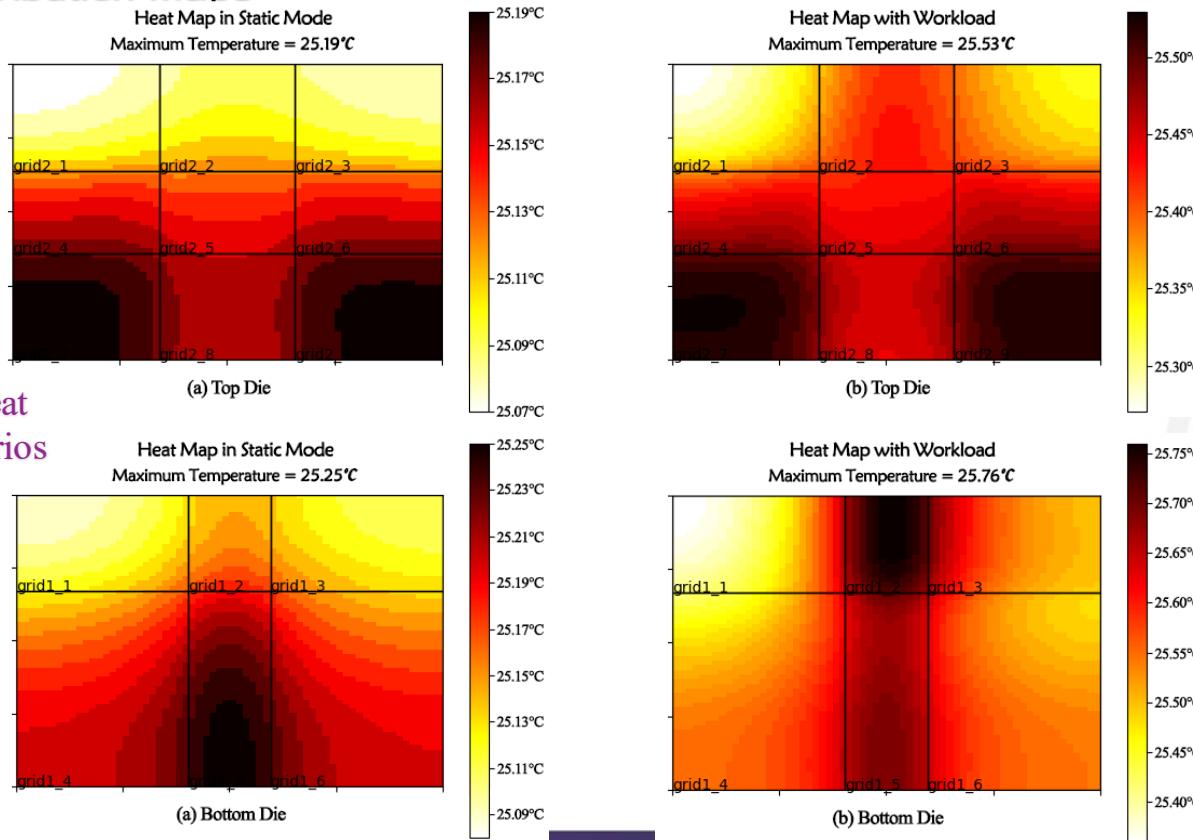
Zhu et al,
<https://github.com/b224hsli/Open3DFlow>

- GUI



Open 3D EDA Platform: Open3DFlow

- Temperature Distribution Maps



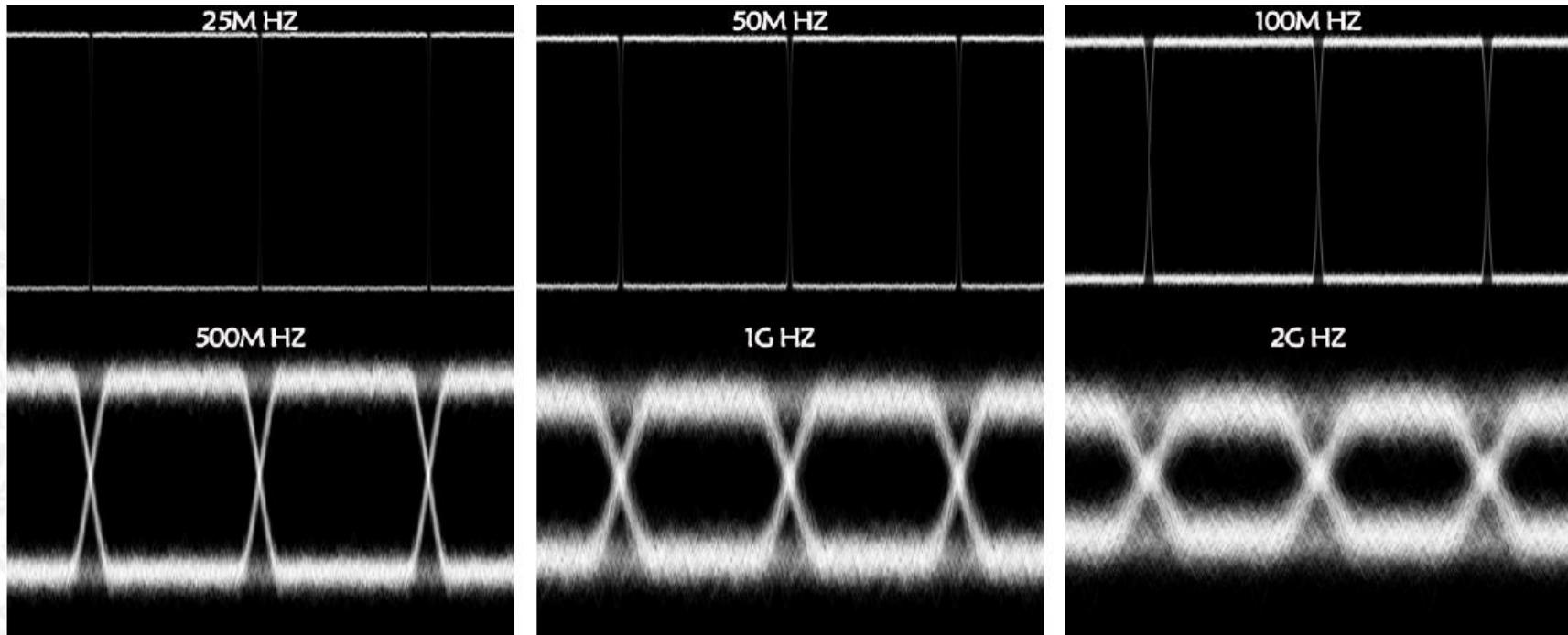
- The bottom die has more heat accumulation in both scenarios

Zhu et al,
<https://github.com/b224his1/Open3DFlow>



Open 3D EDA Platform: Open3DFlow

- Eye Diagrams



• frequency eye



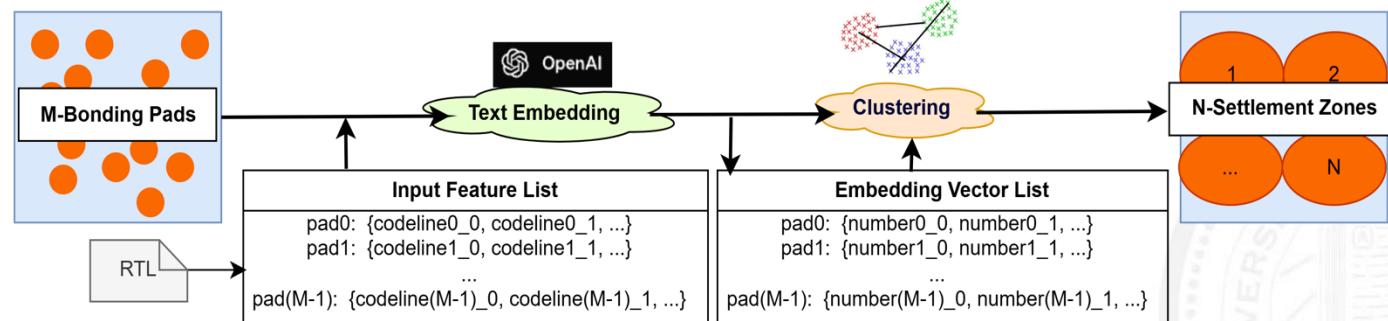
Zhu et al, <https://github.com/b224hisl/Open3DFlow>

AI Leverage: Harnessing the Power of LLM

- AI-aided HB pads placement

Normal cases: Logic synthesis → netlist connectivity → manually adjust → **debugging**

- Routable for both dies
- Fast convergence



- Resources utilization
- Performance

Zhu et al, <https://github.com/b224hisI/Open3DFlow>

	DRC Clean ?		Routing Time (s)		Peak Memory (MB)		Wirelength (m)		Worst Slack (ns)	
	Top Die	Bottom Die	Top Die	Bottom Die	Top Die	Bottom Die	Top Die	Bottom Die	Top Die	Bottom Die
ioPlacer	No	Yes	\	540.63	3227.31	5592.44	0.34	1.29	16.14	5.59
AI-aided	Yes	Yes	27.04	697.52	2866.88	5504.31	0.33	1.38	16.28	5.53
Comparison	\	\	\	29% ↑	11% ↑	2% ↑	3% ↑	7% ↓	0.9% ↑	0.7% ↑



AI Leverage: Harnessing the Power of LLM

- LLM-based Tcl generation: nature language to design 3D IC with Open3DFlow

01 NekoCoder-Tiny

当然！下面是一个使用Tcl编写的快速排序算法的示例：

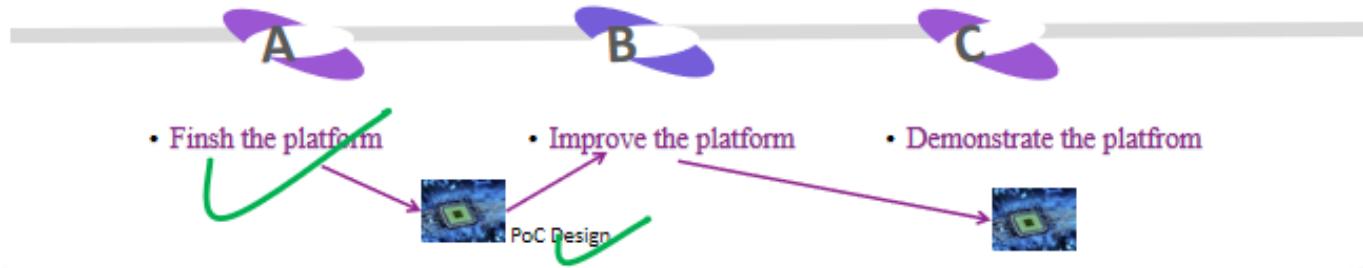
```
tcl
1 proc quicksort {array} {
2     set n [llength $array]
3     if {$n < 2} {
4         return $array
5     }
6
7     # 选择一个基准元素，这里我们选择数组的第一个元素
8     set pivot [lindex $array 0]
9
10    # 创建两个子数组：小于基准和大于基准的元素
11    set less [list]
12    set greater [list]
13
14    for {set i 1} {$i < $n} {incr i} {
15        if {[lindex $array $i] < $pivot} {
16            lappend less [lindex $array $i]
17        } else {
18            lappend greater [lindex $array $i]
19        }
20    }
21
22    # 递归地对小于基准和大于基准的子数组进行排序，并将它们合并
23    return [quicksort $less] [list $pivot] [quicksort $greater]
24 }
25
26 # 测试快速排序算法
27 set unsorted_array {10 7 8 9 1 5)
28 puts "Unsorted array: ($unsorted_array)"
29 set sorted_array [quicksort $unsorted_array]
30 puts "Sorted array: ($sorted_array)"
```

```
{  
    "instruction": "What should be in raw/ASIC-design-example-RISC-CPU/  
Layout-and-post-simulation/scripts/floorplan.tcl?",  
    "input": "",  
    "output": "source ..\\rm_setup\\common_setup.tcl\\nsource -echo ..\\rm_setup/  
common_setup.tcl\\nsource ..\\rm_setup\\icc_setup.tcl\\nsource -echo ..\\rm_setup/  
icc_setup.tcl\\nsource ..\\rm_setup\\lcrm_setup.tcl\\nsource -echo ..\\rm_setup/  
lcrm_setup.tcl\\nopen_mw_lib control_pad.mw\\ncopy_mw_cel -from data_setup -to  
floorplan\\nopen_mw_cel  
floorplan\\n#####\\n#####\\n#####\\n#Create a Rectangular  
Block\\n#####\\n#####\\n#####\\n#Create corner cells and supply  
cells\\ncreate_cell {cornerll cornerrl cornerul cornerur} PCORNER\\ncreate_cell  
{vss1left vss1right} PVSS1; #core ground\\ncreate_cell {vdd1left vdd1right} PVDD1;  
#core supply \\ncreate_cell {vss2top vss2bottom} PVSS2; #pad ground\\ncreate_cell  
{vdd2top vdd2bottom} PVDD2;#pad_supply\\n\\n#constrain the  
corners\\nset_pad_physical_constraints -pad_name \"cornerul\" -side  
1\\nset_pad_physical_constraints -pad_name \"cornerur\" -side  
2\\nset_pad_physical_constraints -pad_name \"cornerrl\" -side
```

Future Directions



- Demonstrate this platfrom through silicon
- Scalability



Partial Proprietary PDK Support (GF22)



Thank You!