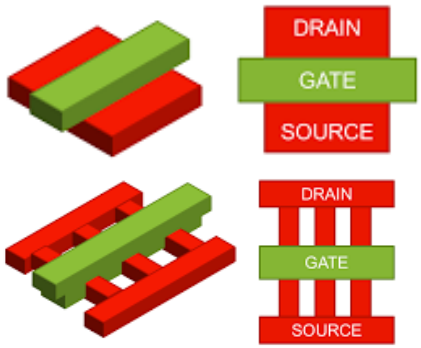
Automated system partitioning based on hypergraphs for 3D stacked integrated circuits

Integrated circuits: Let’s go 3D

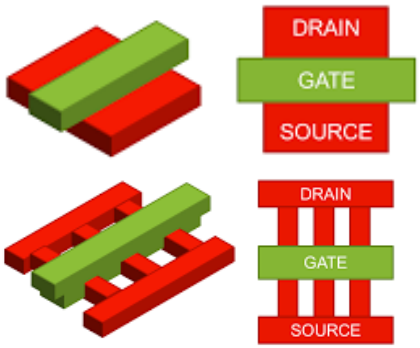
Building an Integrated Circuit (IC)

Transistors to build gates

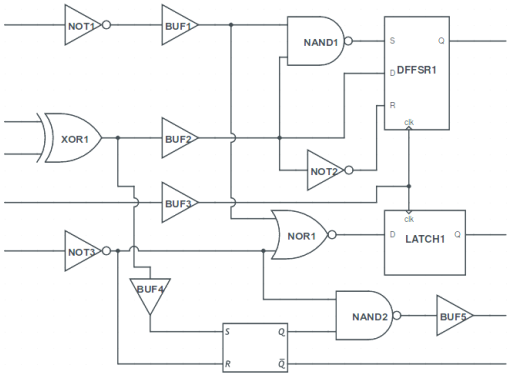


Building an Integrated Circuit (IC)

Transistors to build gates

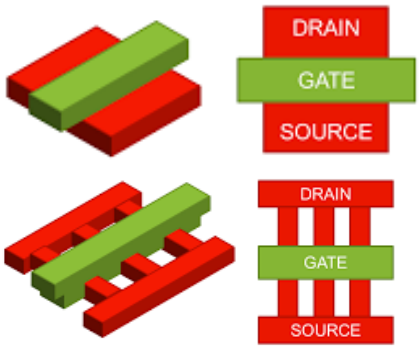


Gates to build logic functions

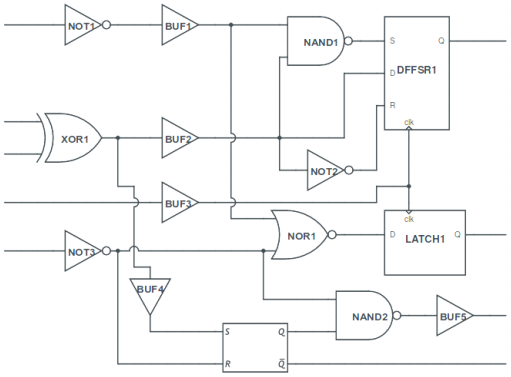


Building an Integrated Circuit (IC)

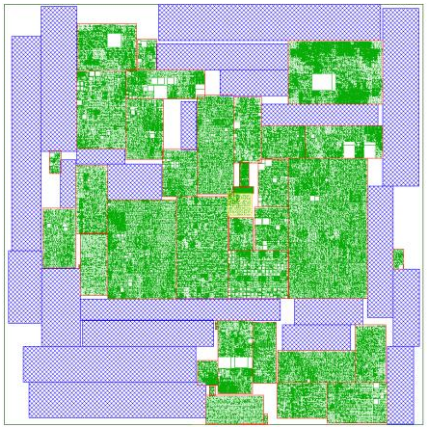
Transistors to build gates



Gates to build logic functions



Logic functions build ICs

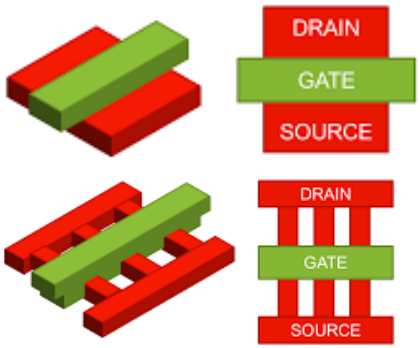


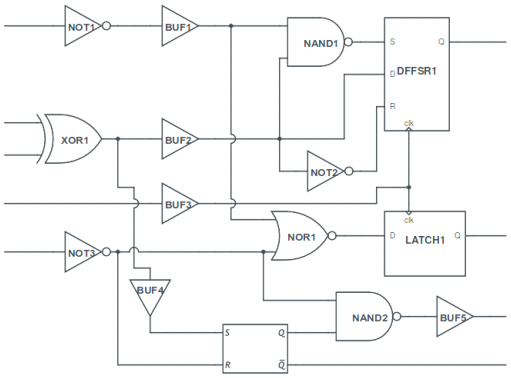
Building an Integrated Circuit (IC)

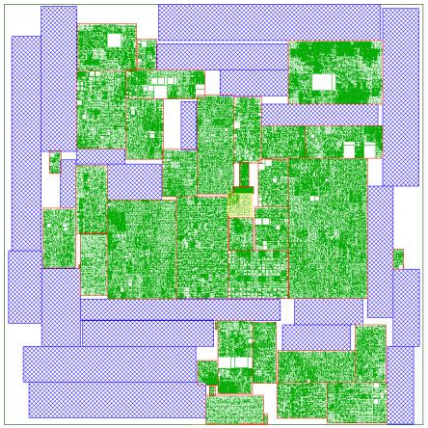
Transistors to build gates

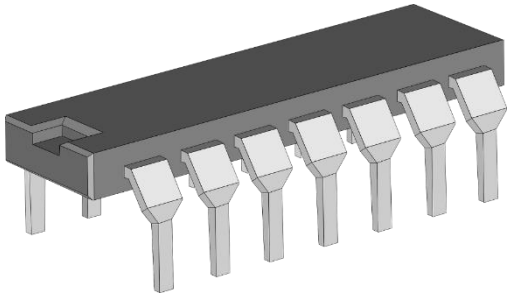
Gates to build logic functions

Logic functions build ICs



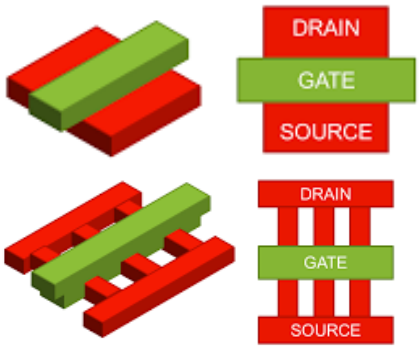




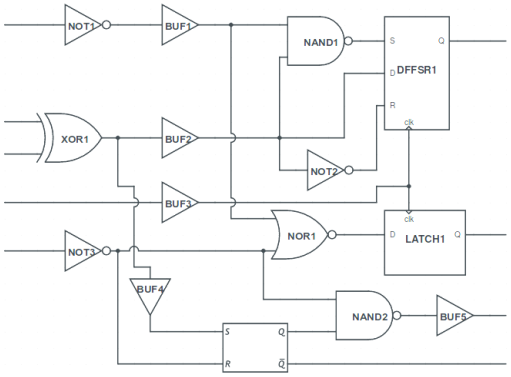


Building an Integrated Circuit (IC)

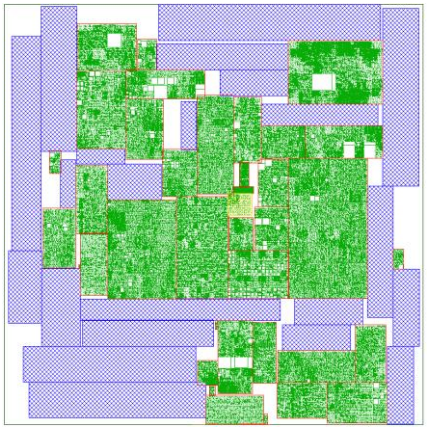
Transistors to build gates



Gates to build logic functions

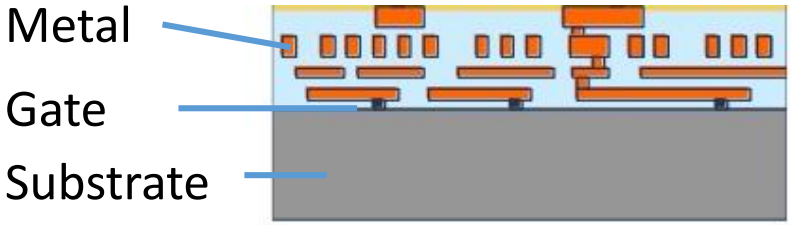


Logic functions build ICs



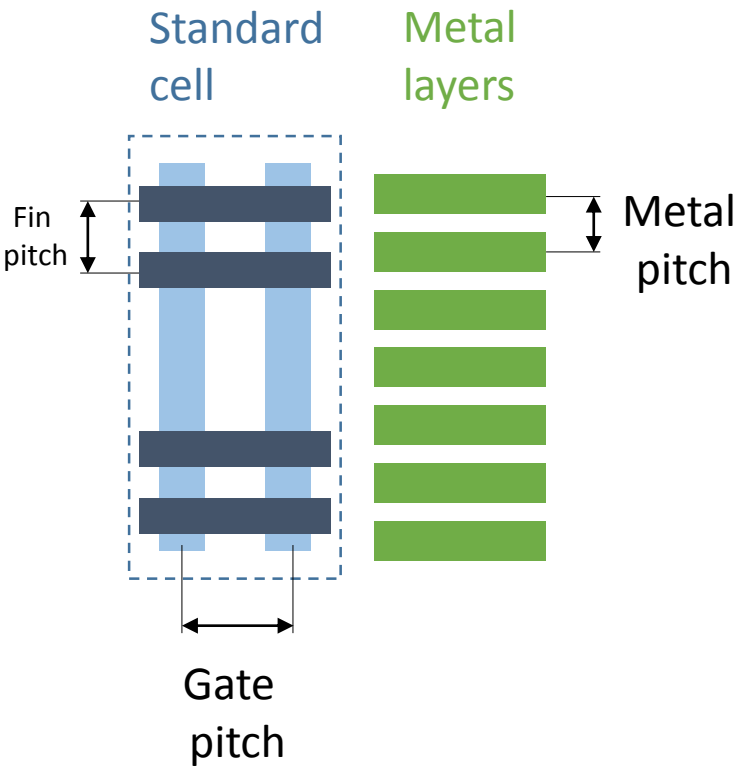
System performance depends on transistor performance and the quality of the system interconnect

What does a 2D IC look like?



Planar 2D IC: only one transistor layer

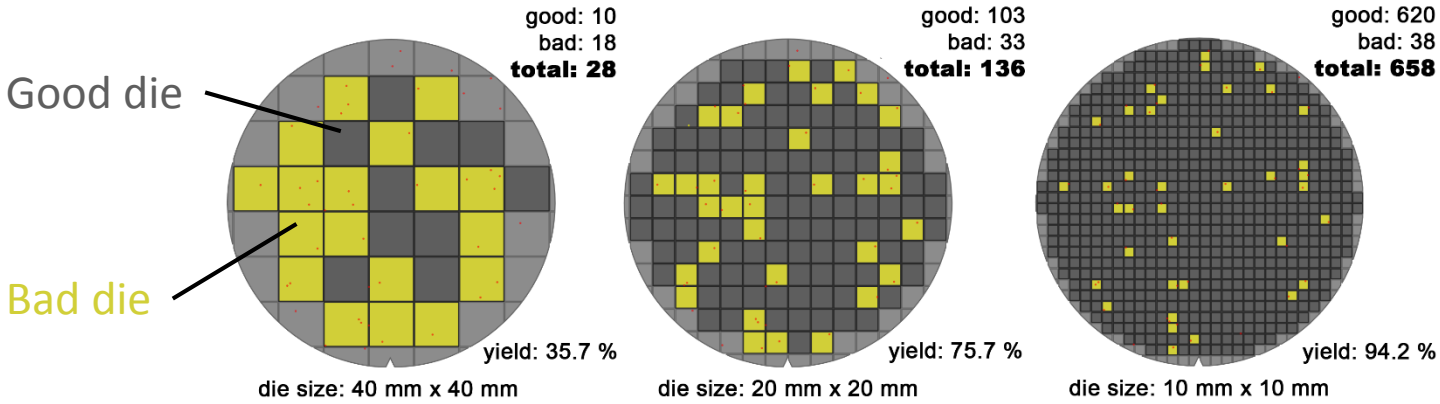
What are its limitations?



If you want **more** of them, you need them **smaller**.

But scaling has **physical** and **financial** limitations.

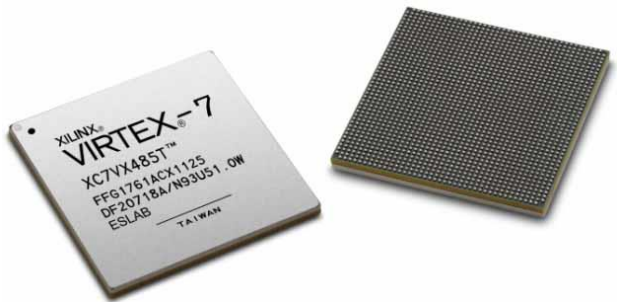
You can’t simply make it bigger



Constant amount of defects per wafer

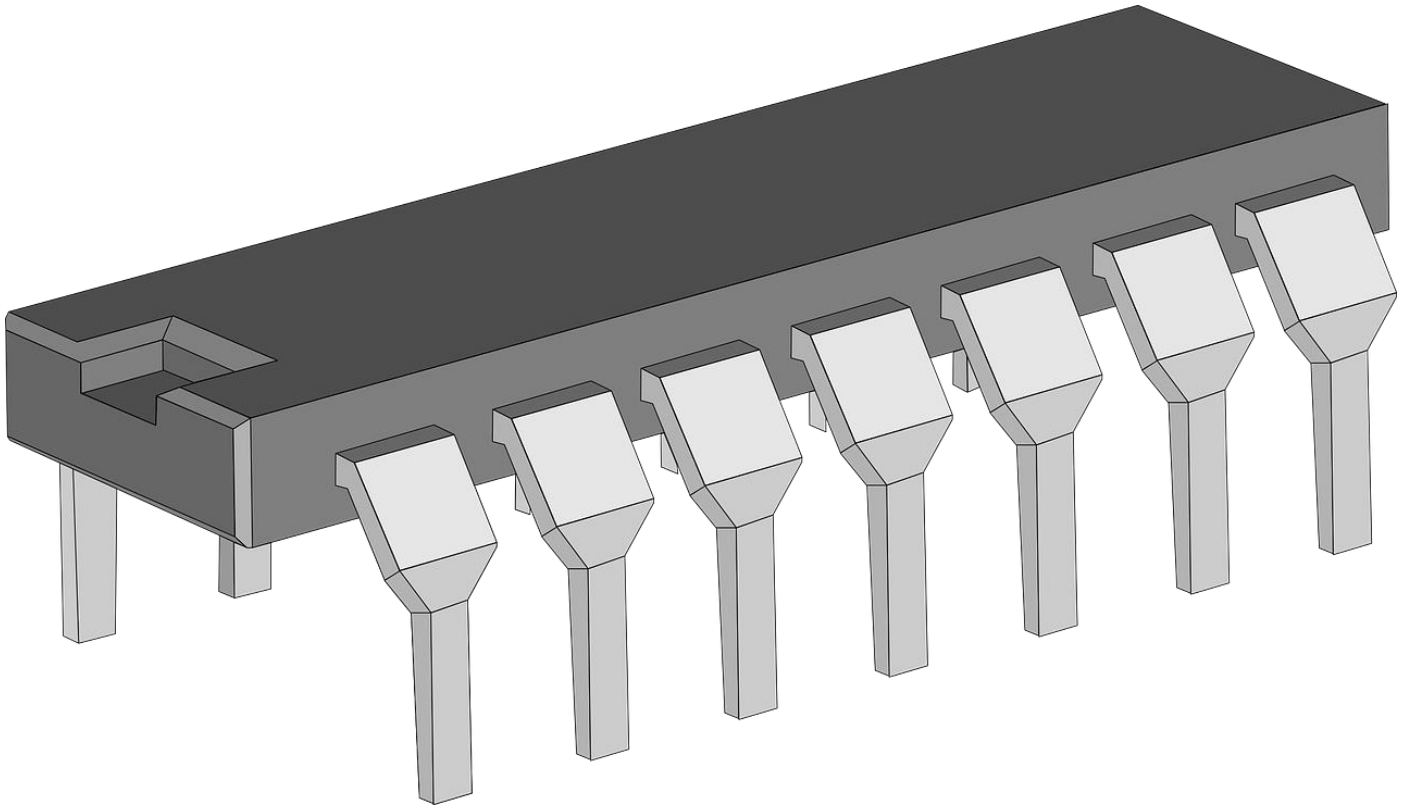
Larger IC means lower yield

Split the IC to keep it small

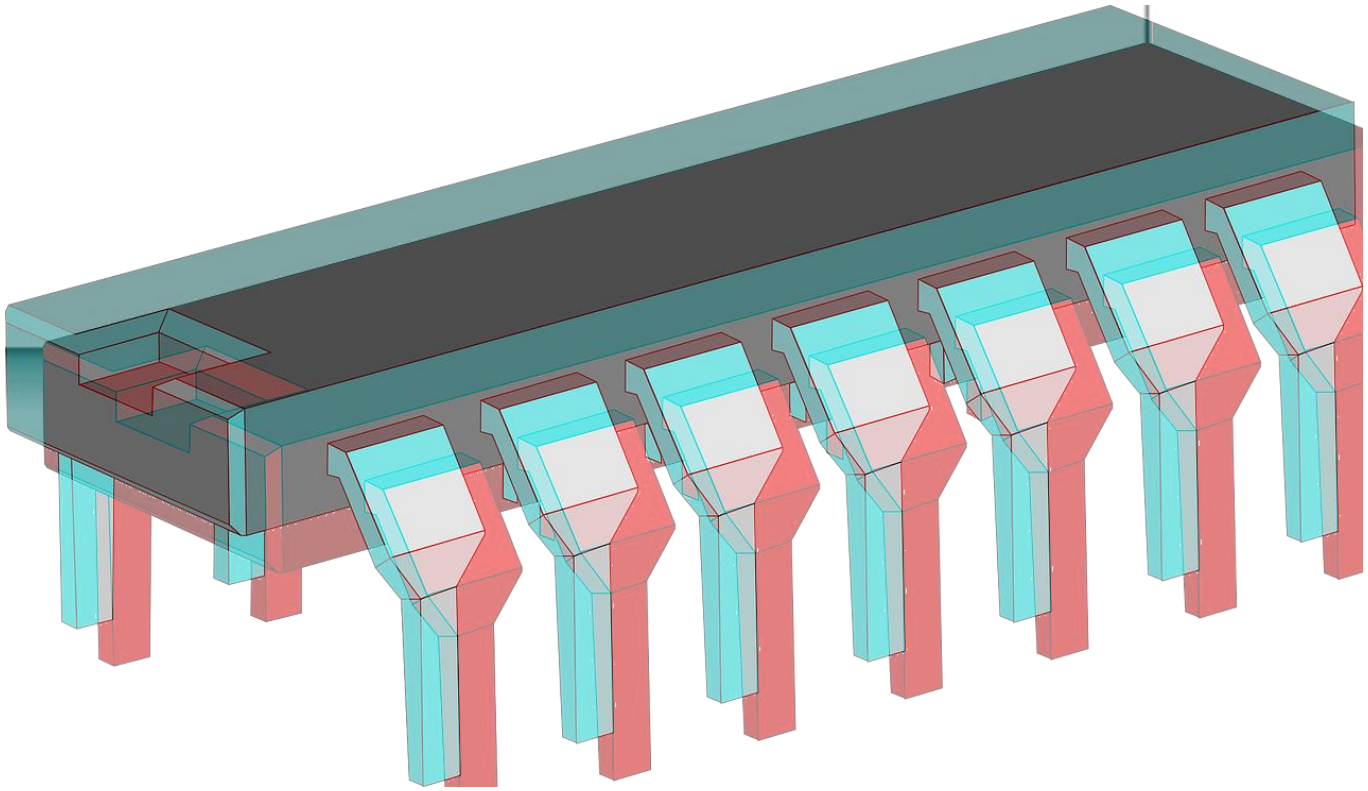


Xilinx split its latest node to keep it relatively affordable.

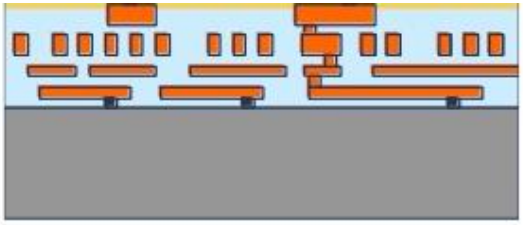
What is a 3D IC?



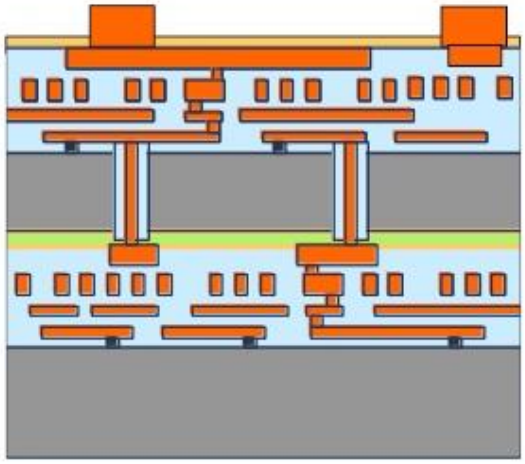
What is a 3D IC?



What is a 3D IC?

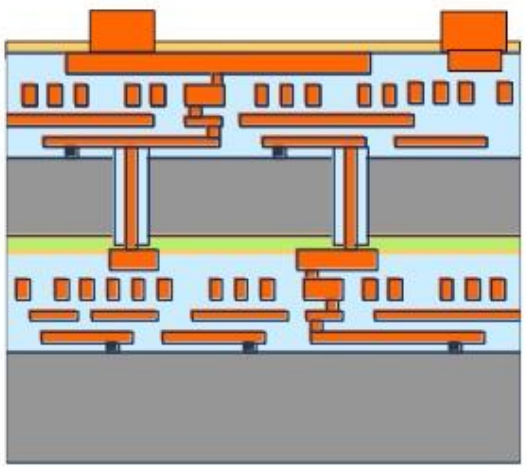


What is a 3D IC?

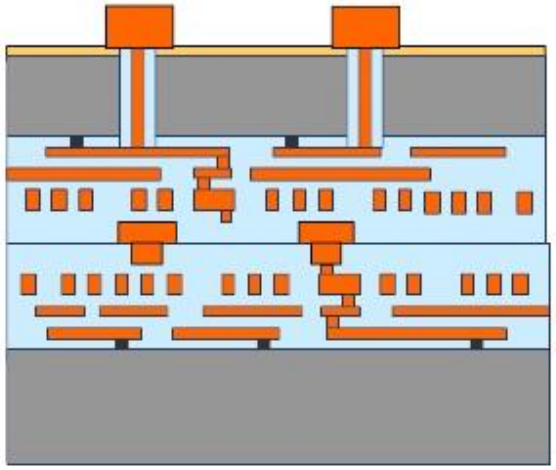


Face-to-Back (past)

What is a 3D IC?

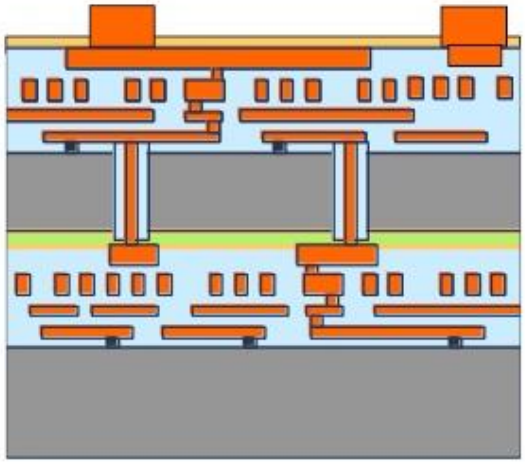


Face-to-Back (past)

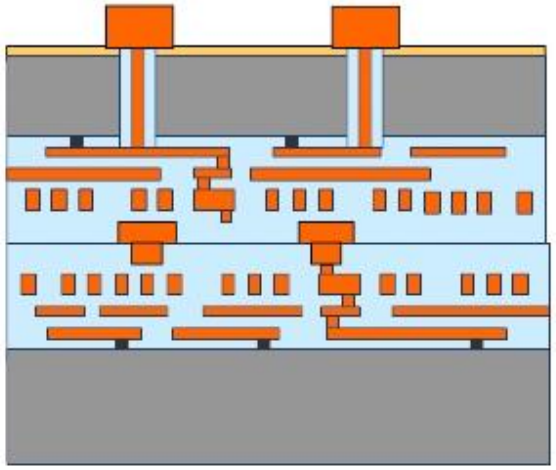


Face-to-Face (present)

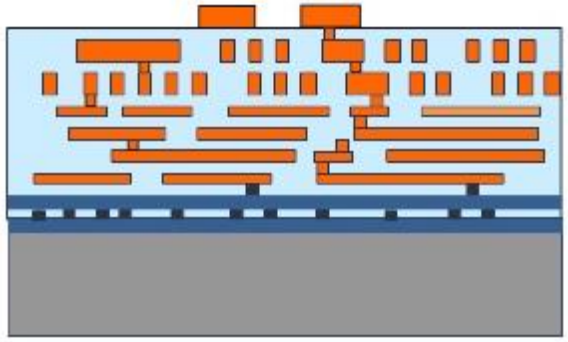
What is a 3D IC?



Face-to-Back (past)

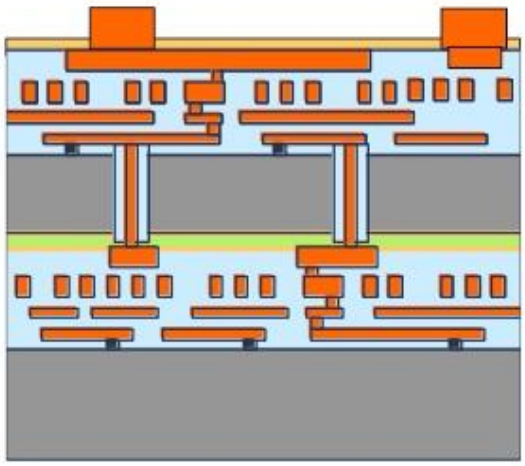


Face-to-Face (present)

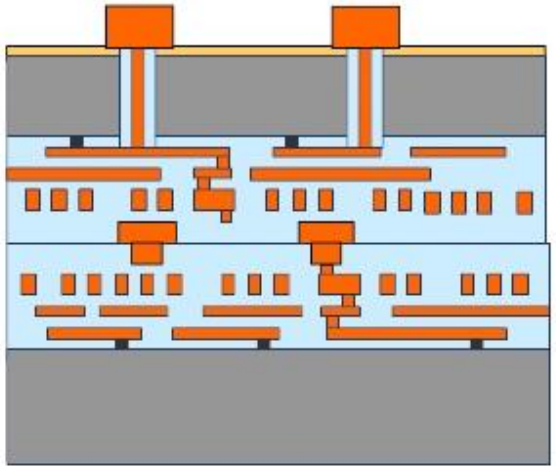


Transistor-on-transistor (future)

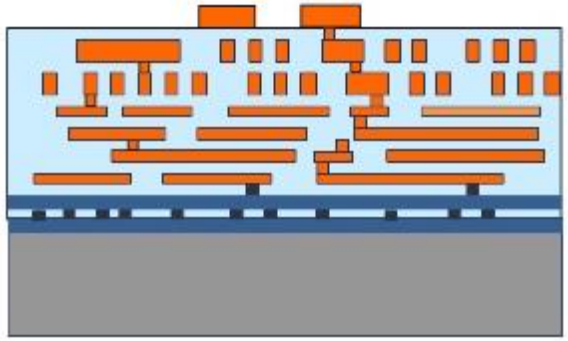
What is a 3D IC?



Face-to-Back (past)



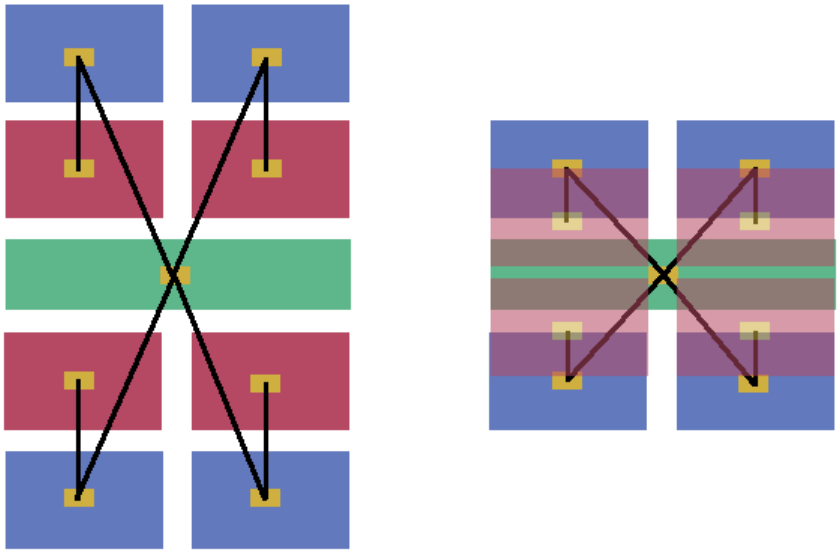
Face-to-Face (present)



Transistor-on-transistor (future)

Somebody needs to decide what goes where

3D benefit: shorter connections



Increased performance

Decreased system power consumption

Improved area utilisation

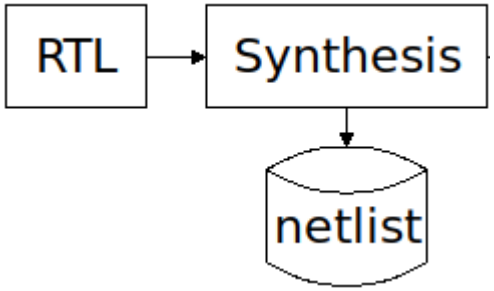
2D flow...

Hardware description: Verilog, VHDL, ...

2D flow...

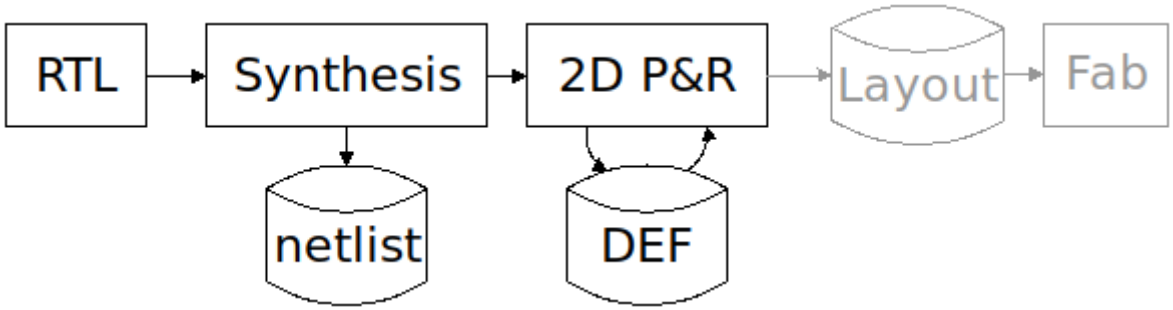
Synthesis: Yosys, ODIN-II, ABC, ...

Yields a netlist.



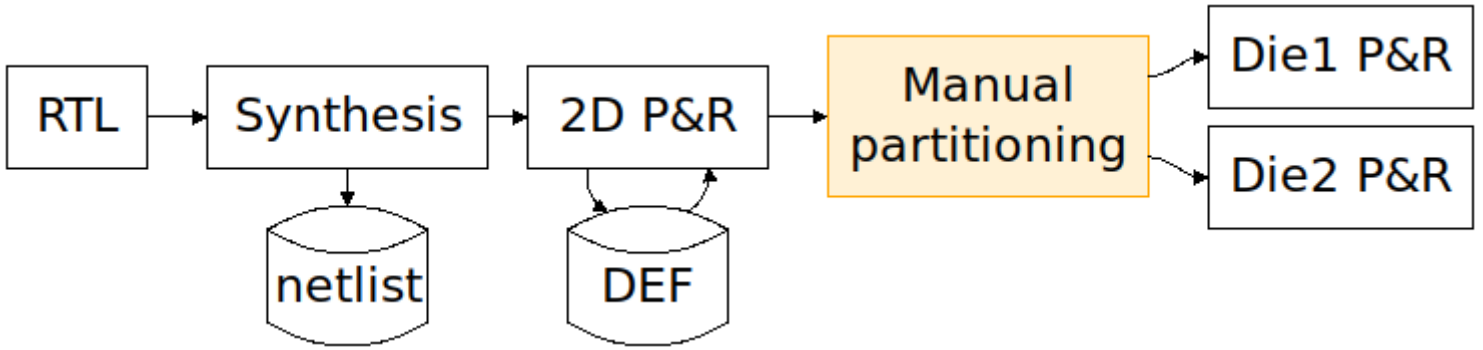
2D flow...

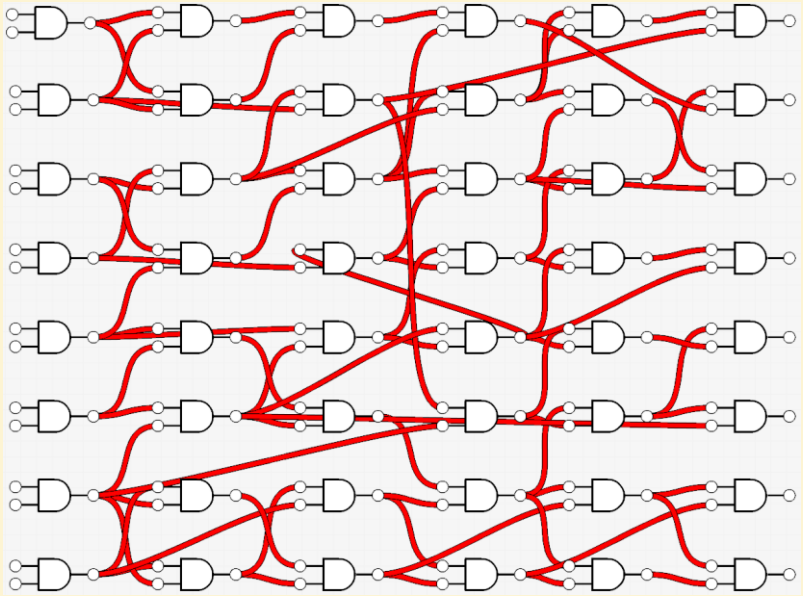
Place and route (P&R): QRouter, Graywolf, FGR, ...



… Extended to 3D

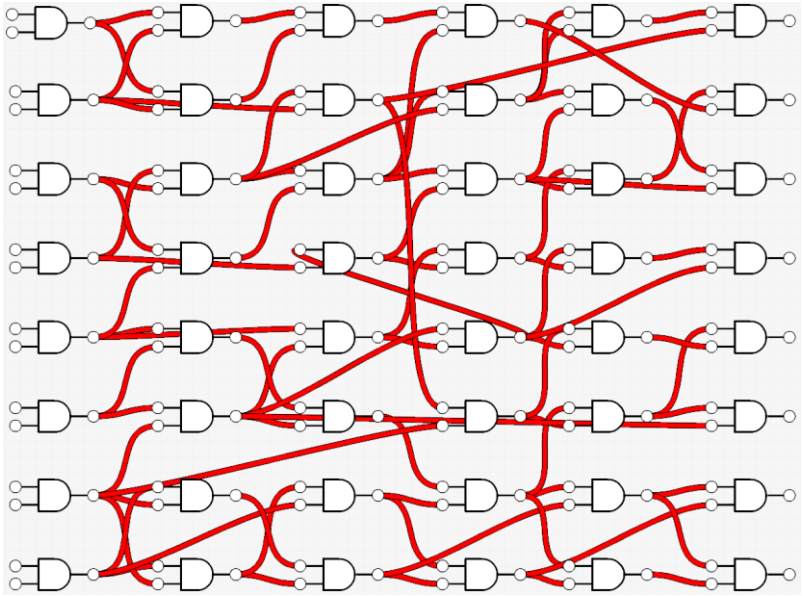
Pick which standard cell or module goes where





This is not a design.

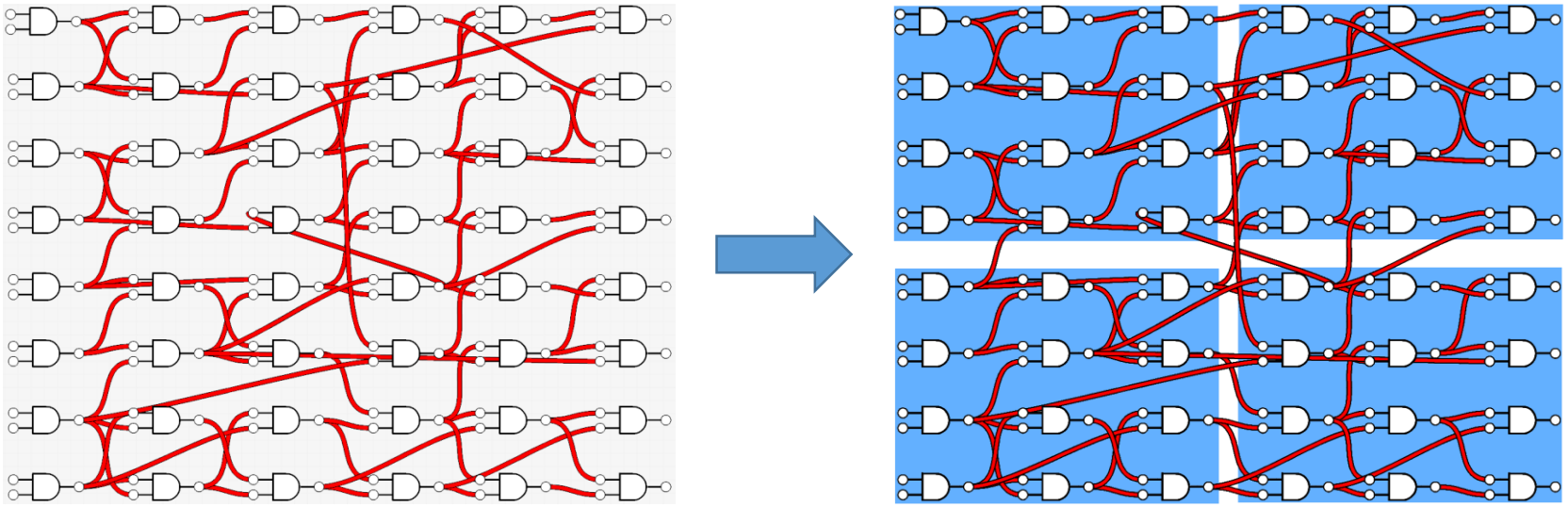
Bipartition this system



Objectives:

- Area balance
- Limit 3D interconnectivity

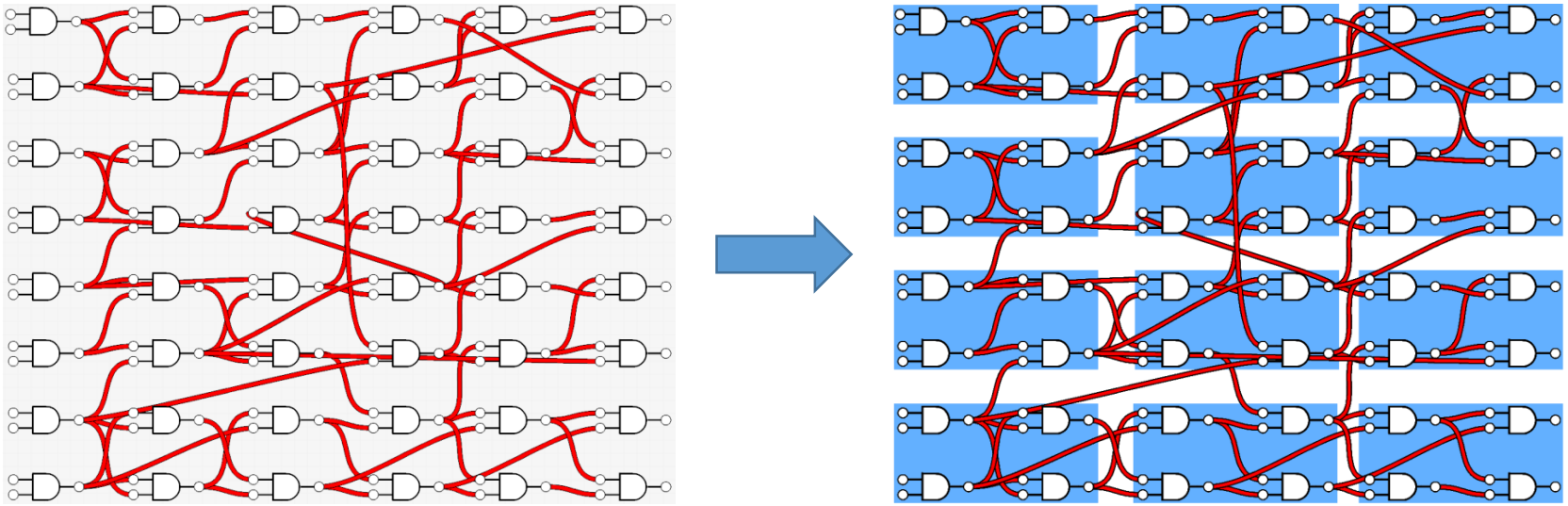
Clustering: hide the shortest nets



Big clusters: Few nets ☺

Long nets hidden ☹

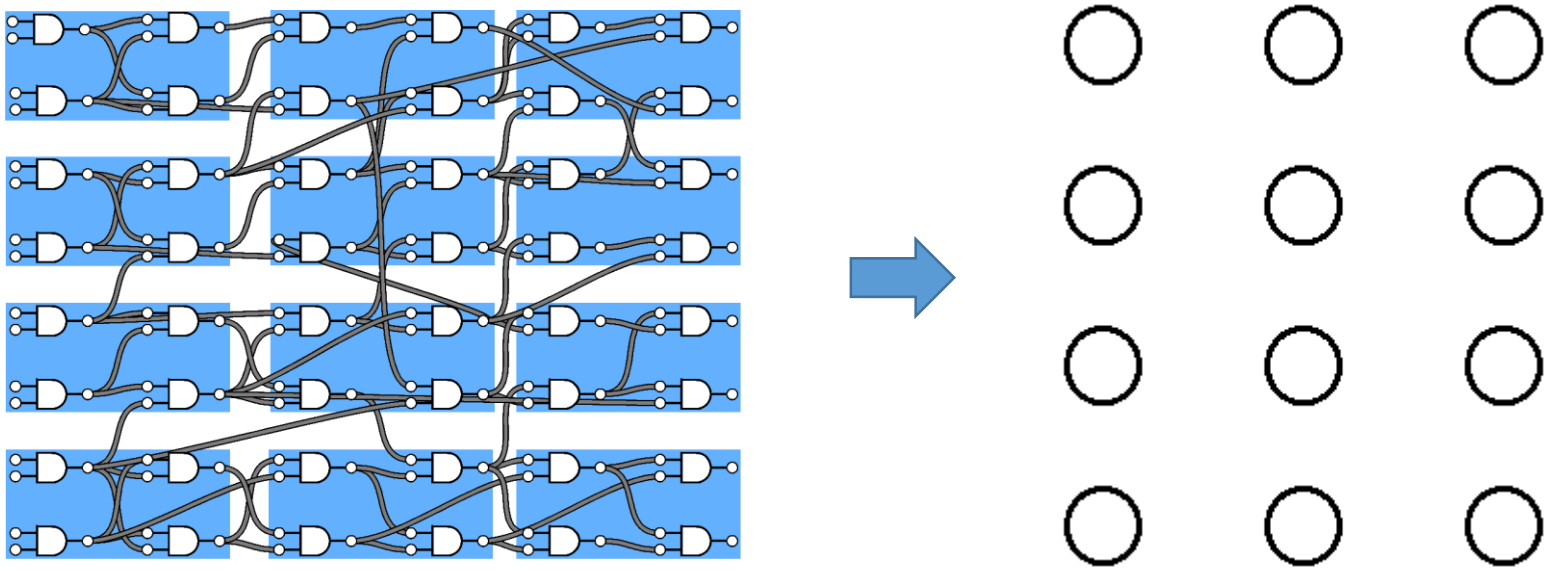
Clustering: hide the shortest nets



Small clusters: Lots of nets ☹

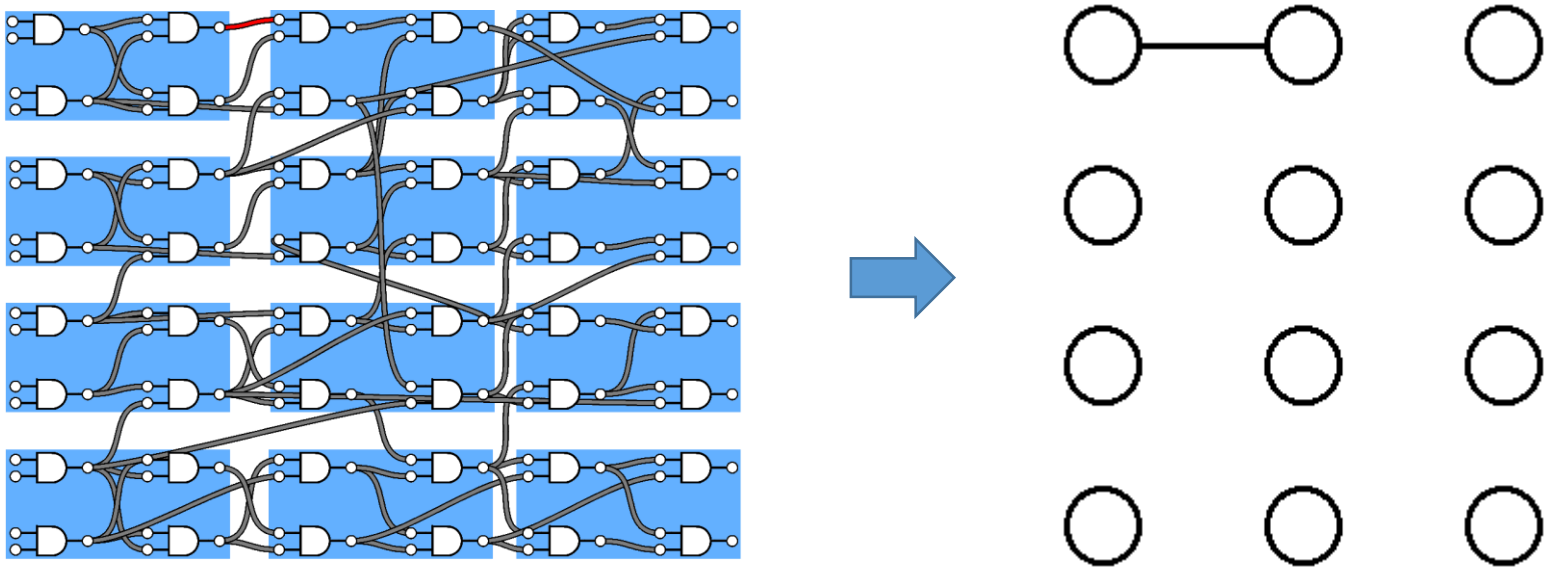
Long nets apparent ☺

Graph extraction



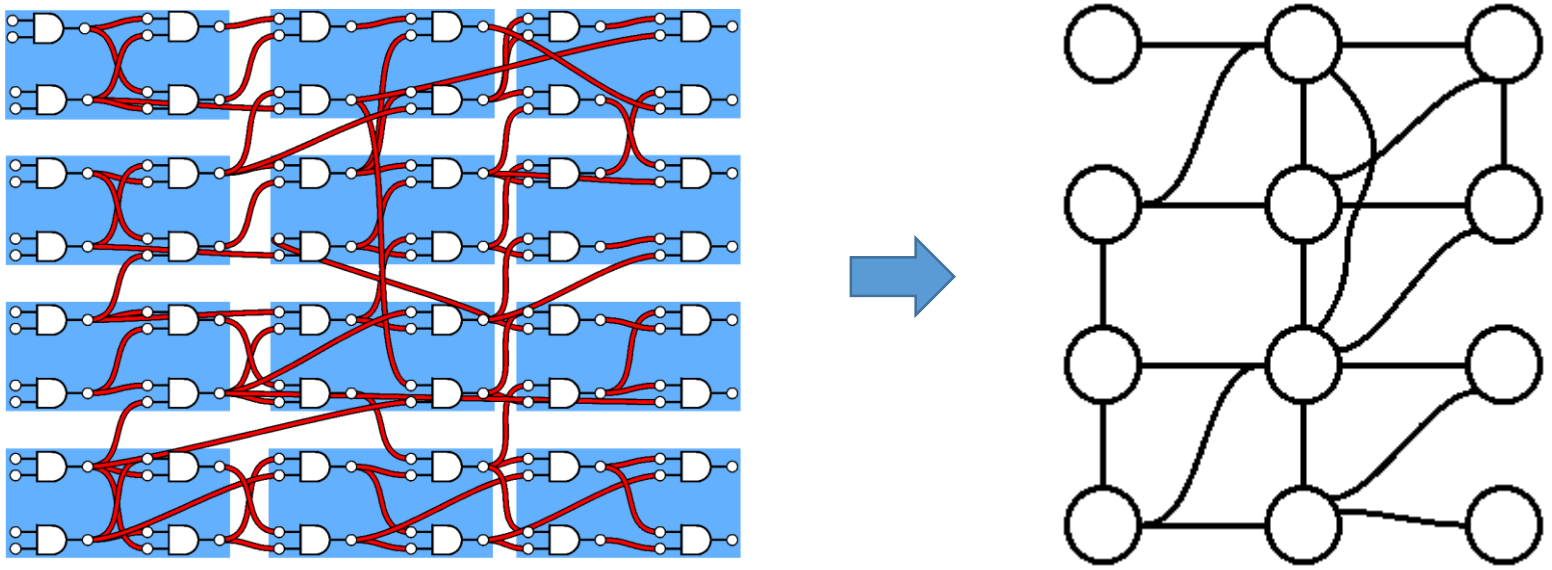
Clusters become vertices

Graph extraction



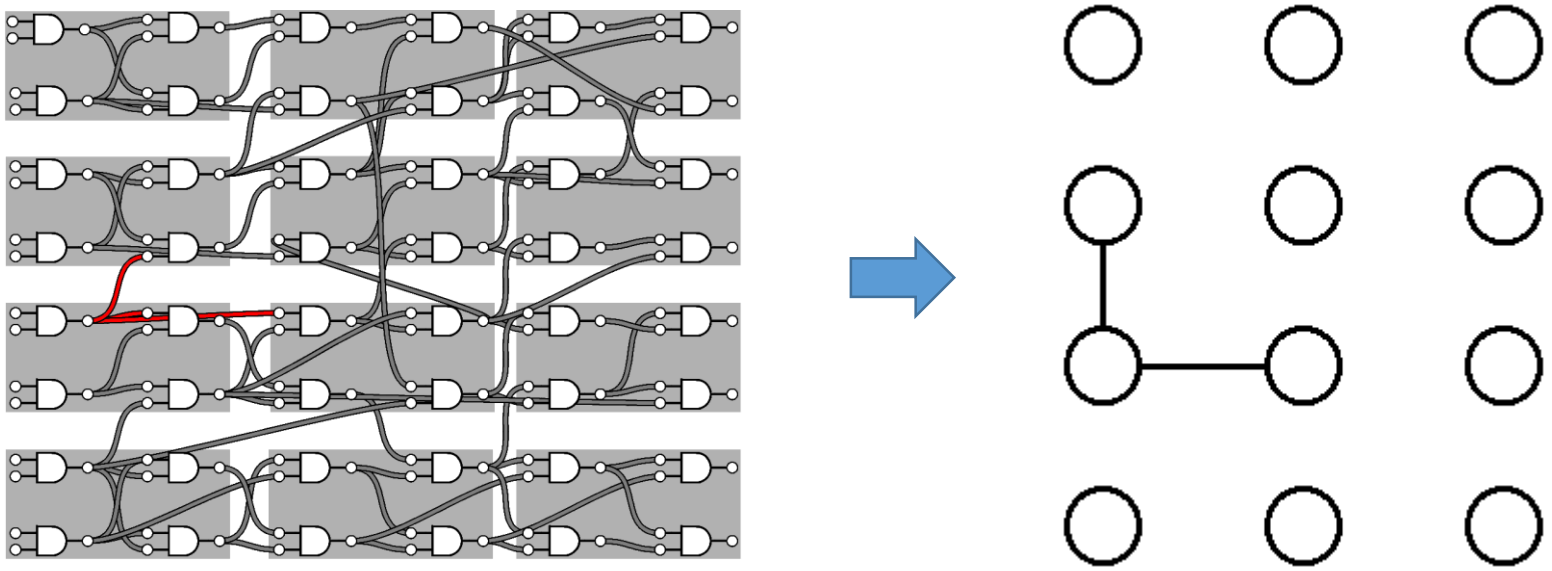
Interconnections become edges

Graph extraction



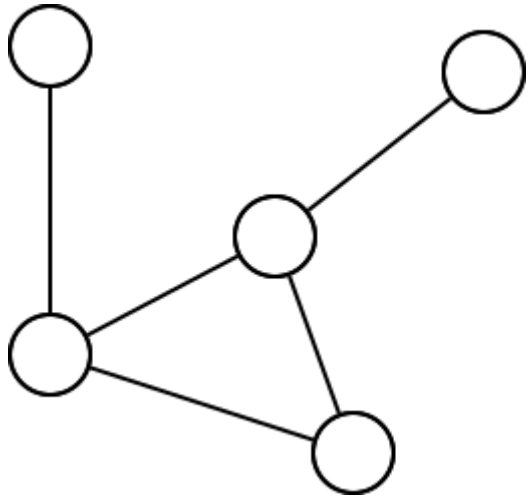
Extraction complete

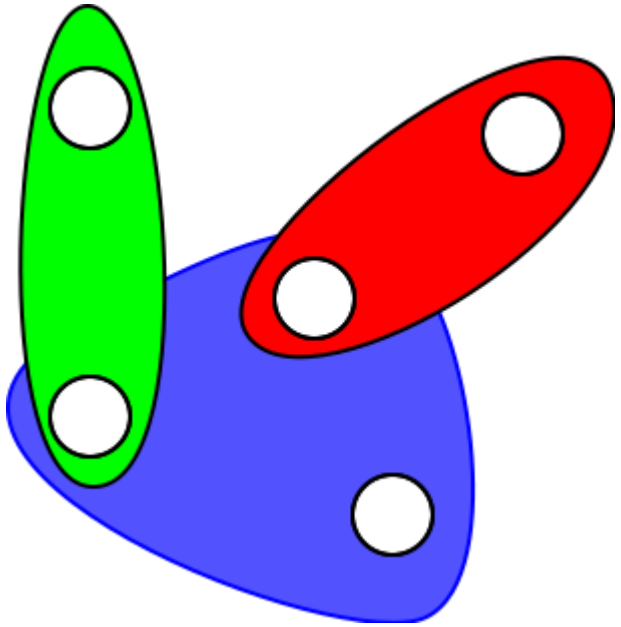
Graph extraction



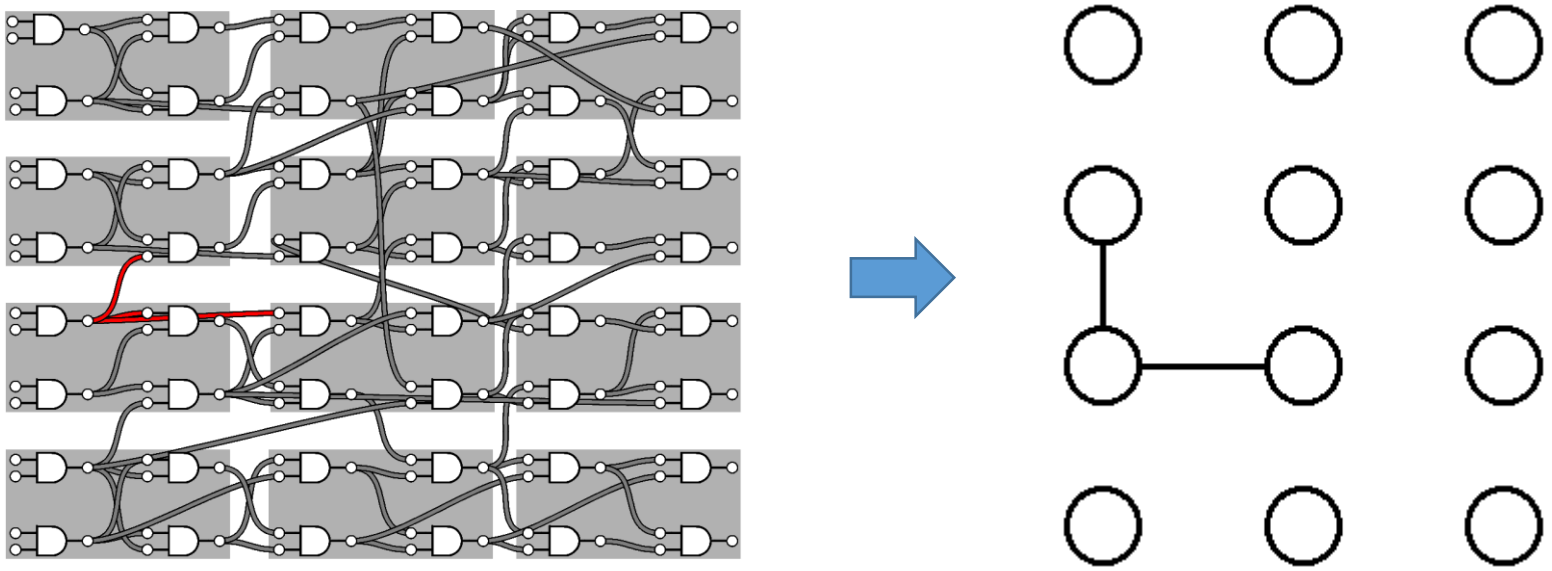
Single nets are split into different edges

From graph to hypergraph



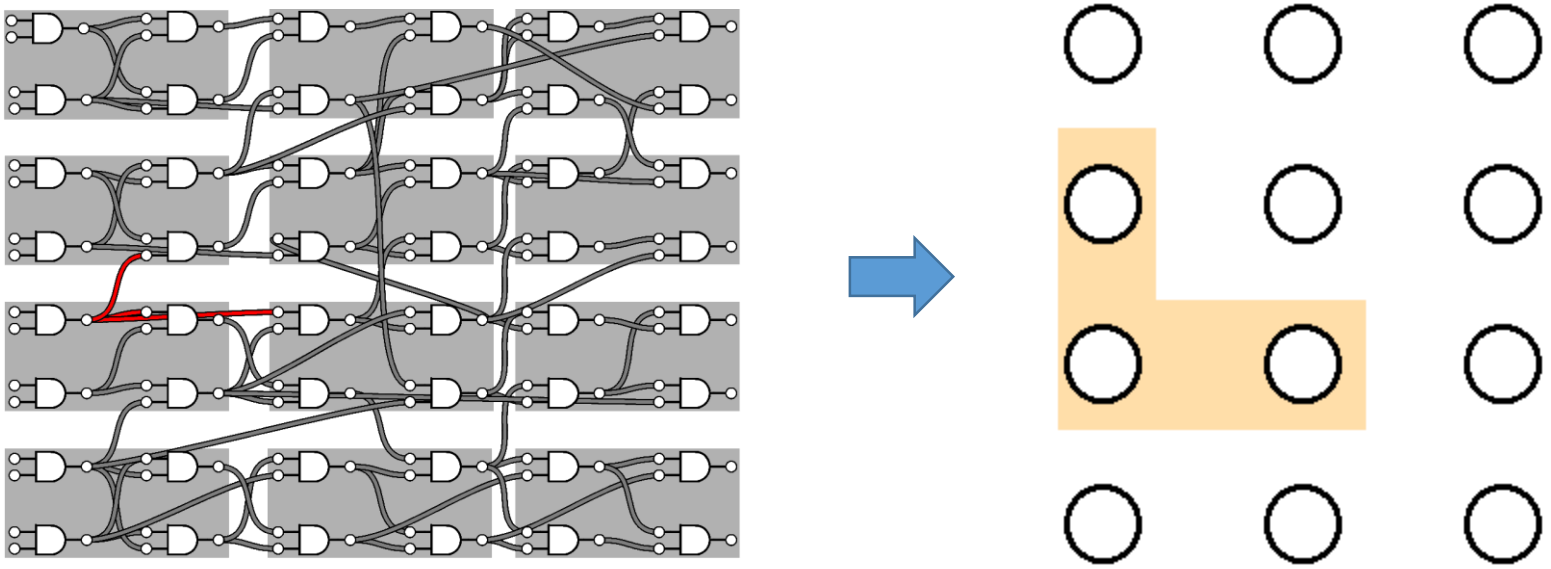


Graph extraction



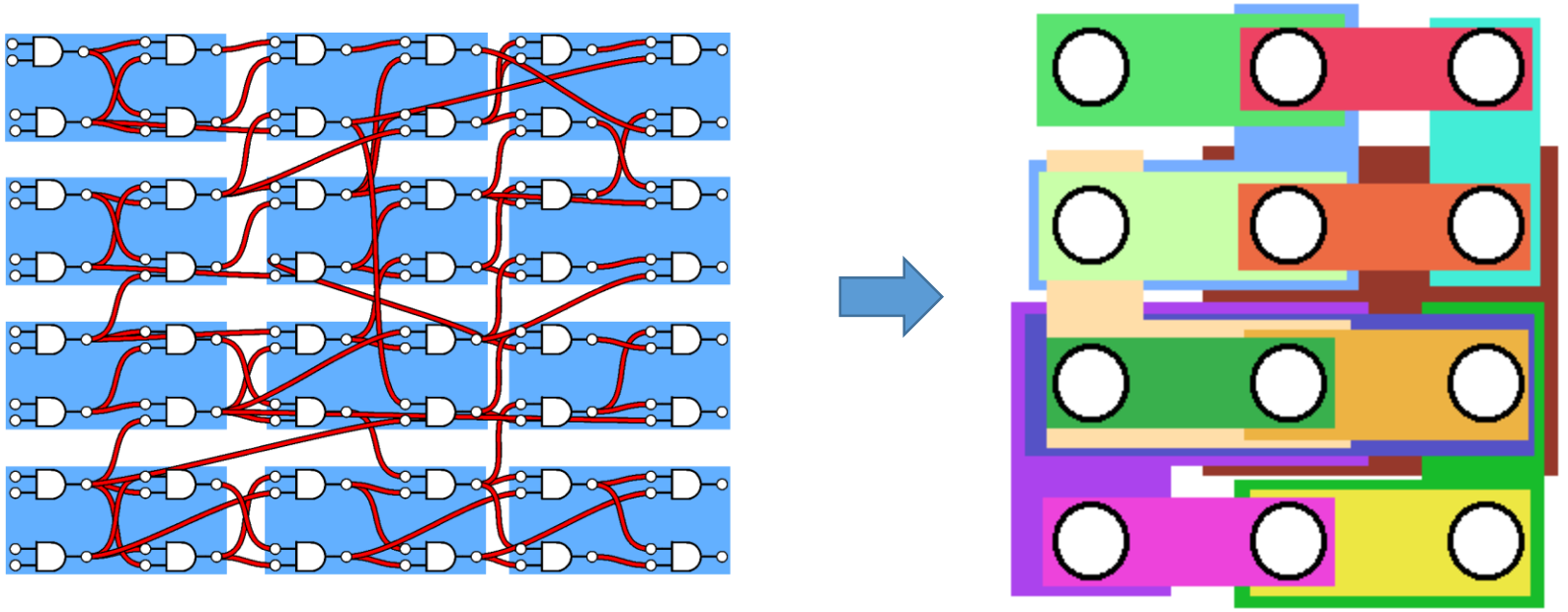
Single nets are split into different edges

Graph extraction

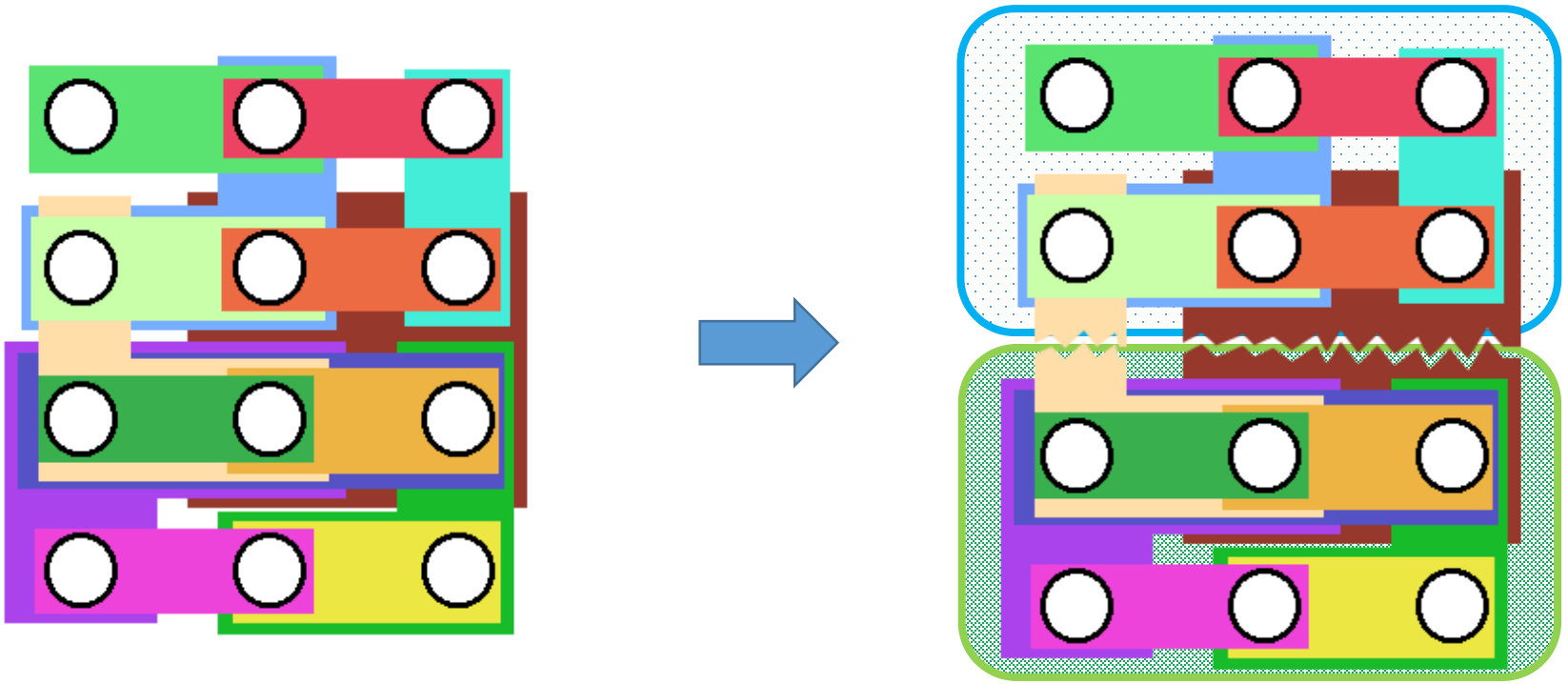


Hyperedges maintain their integrity

Hypergraph extraction



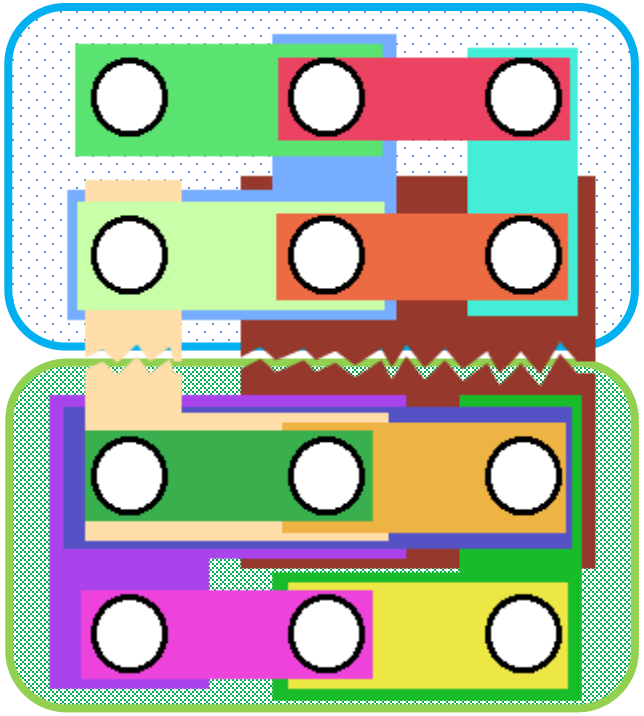
Partitioning



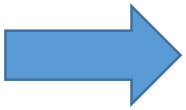
Minimize the crossing nets and maintain area balance

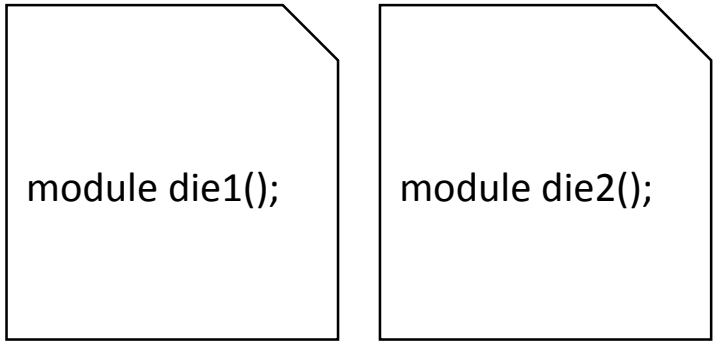
Split the netlist

Die 1

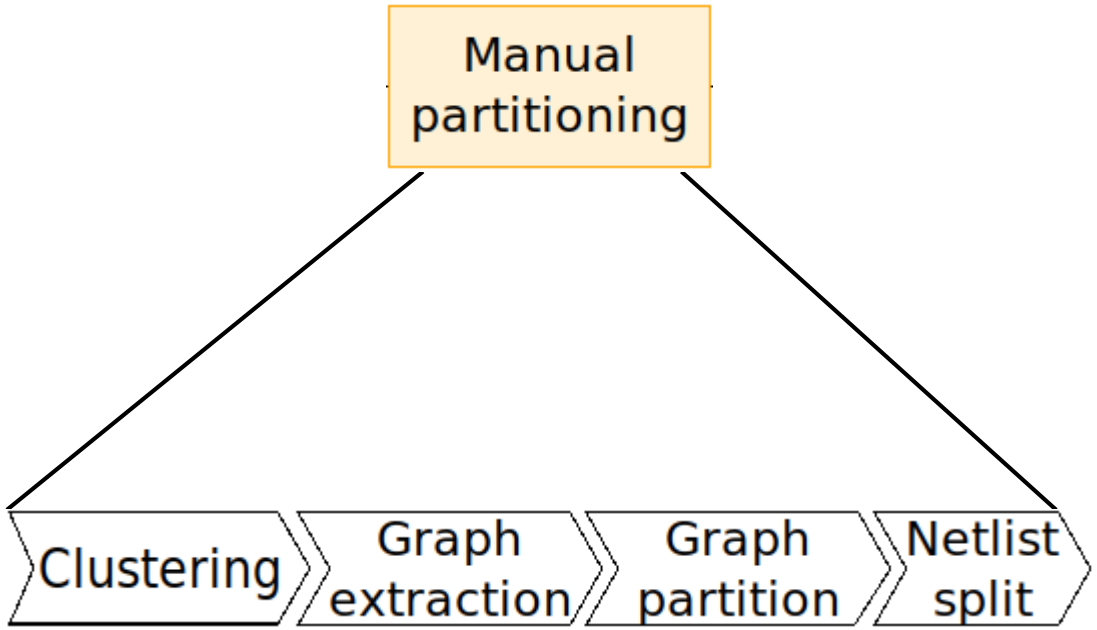


Die 2

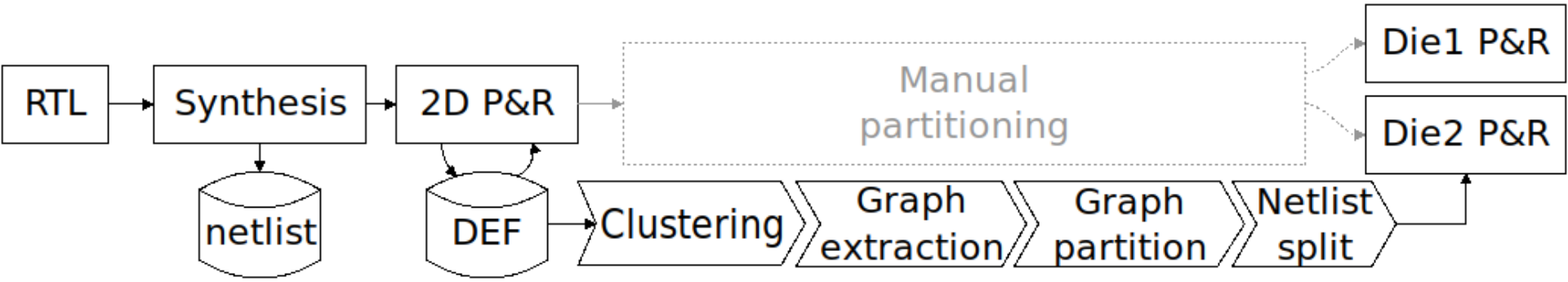




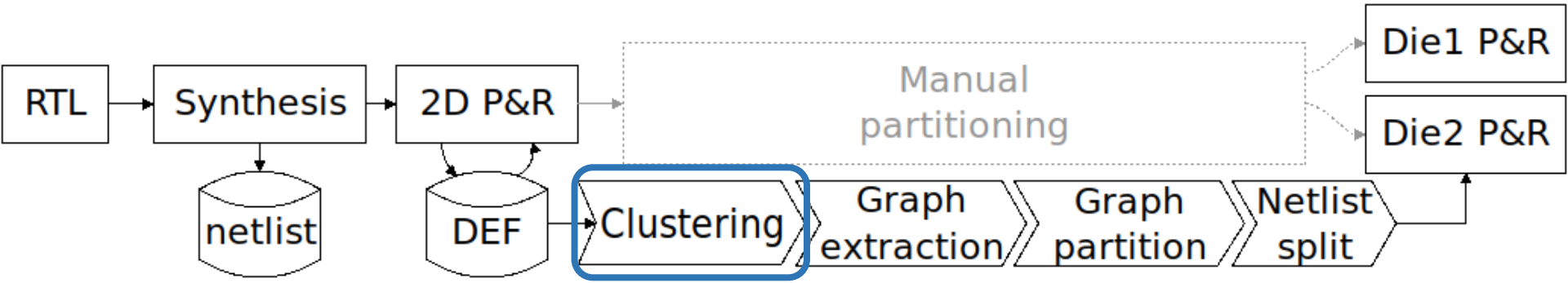
Replace the manual partitioning



Automated 3D flow

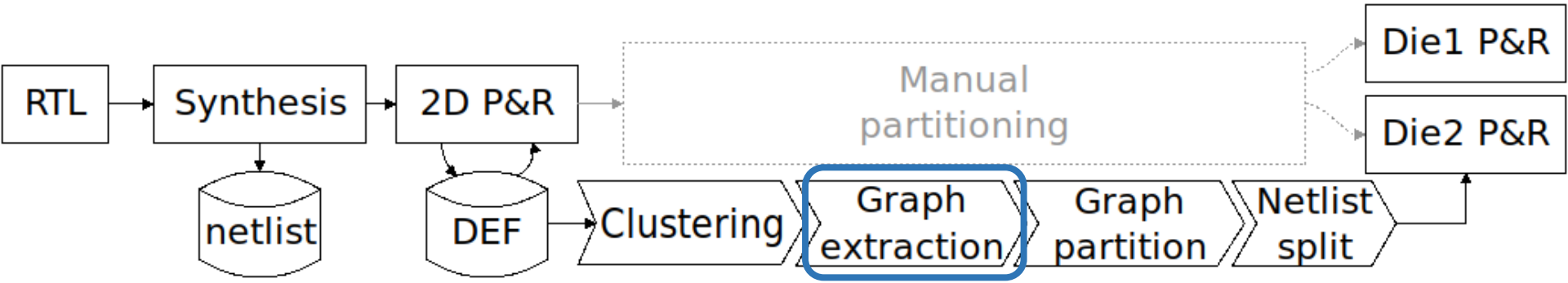


Automated 3D flow



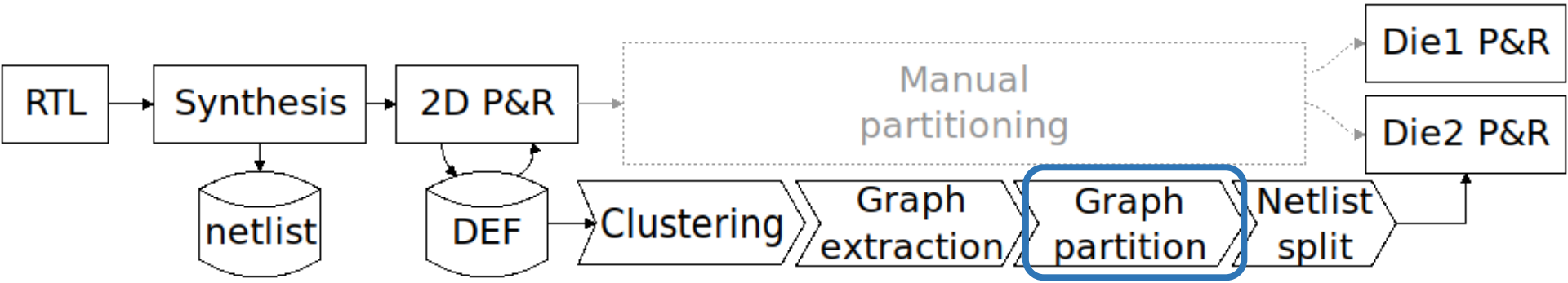
- Export design properties
- Design clustering
- Export clusters connectivity

Automated 3D flow



- From the clustering outputs
- Build hyperedges and merge identical ones
- Compute graph weights
- Format graph and call partitioning tool
- Export cut data and partition directives

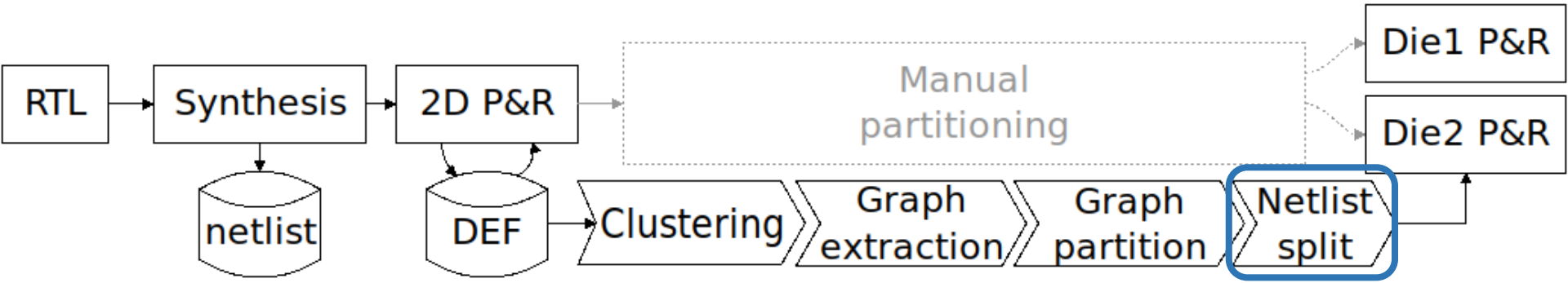
Automated 3D flow



hMETIS Karypis Lab, University of Minesota

PaToH Ümit Catalyürek, Bilkent University

Automated 3D flow



- Early development stage
- Based on 2D netlist and partitioning directives

Designs tested

LDPC

RISC V (BoomCore)

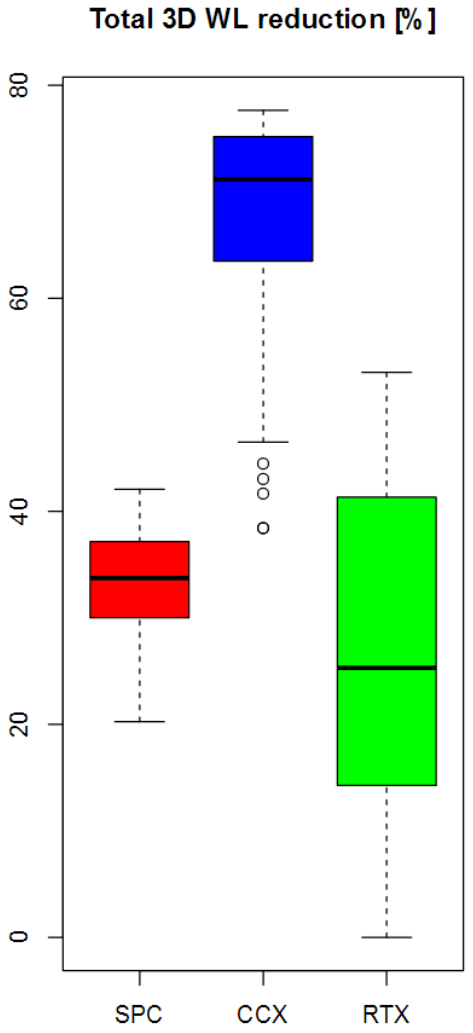
OpenSparc T2: SPC, CCX and RTX

3D: up to 77% less total wire length

Experiments using different functional blocks from OpenSPARC T2 SoC

Different partitioning schemes

Different clustering options

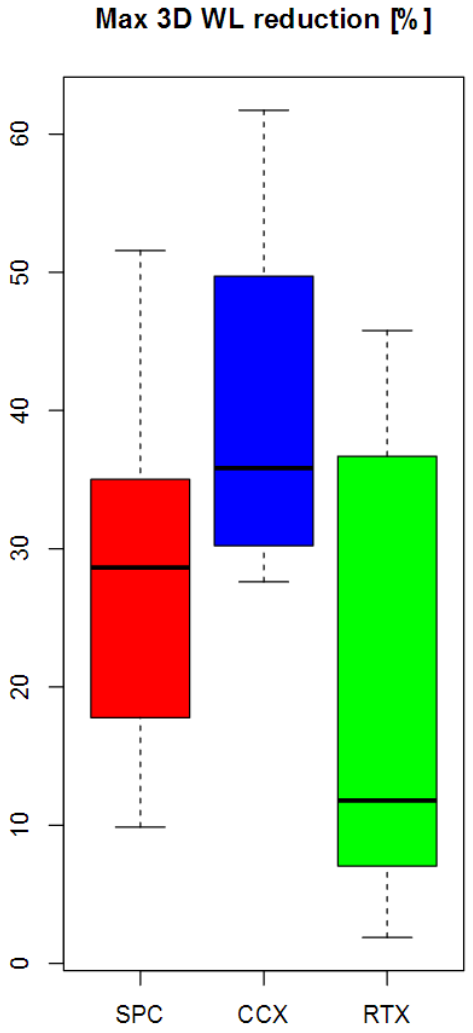


3D: Up to 61% shorter critical path

Experiments using different functional blocks from OpenSPARC T2 SoC

Different partitioning schemes

Different clustering options



Open questions

What is the best clustering?

Does the clustering method have a significant impact?

Can we predict the “partitionability” of a design?

Links: https://fosdem.org/2018/schedule/event/cad_3d_asic/