

ASIC Layout Flows

98-154/18-224/18-624: Intro to Open-Source Chip Design

Physical Design Automation

- Manually laying out digital logic by drawing transistors is possible but very inefficient
- The vast majority of the physical design flow can be entirely automated, lots of internal checks to verify manufacturability and correctness
- Known as “RTL-to-GDS flow”

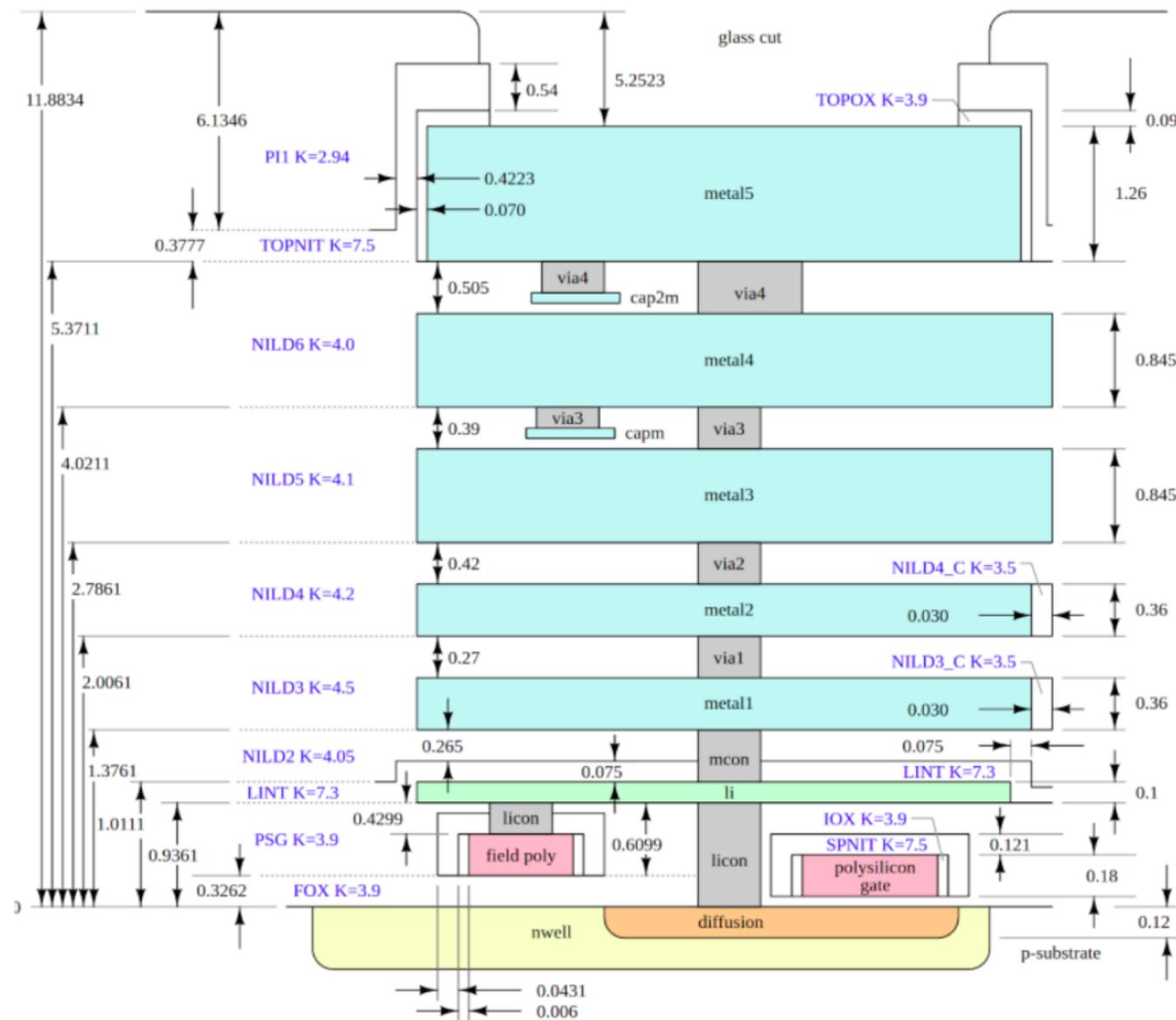
SKY130 Process

- First recent open-source manufacturable process
 - 130nm process node, includes *everything* to design a chip and be able to fabricate it at the Skywater Foundry

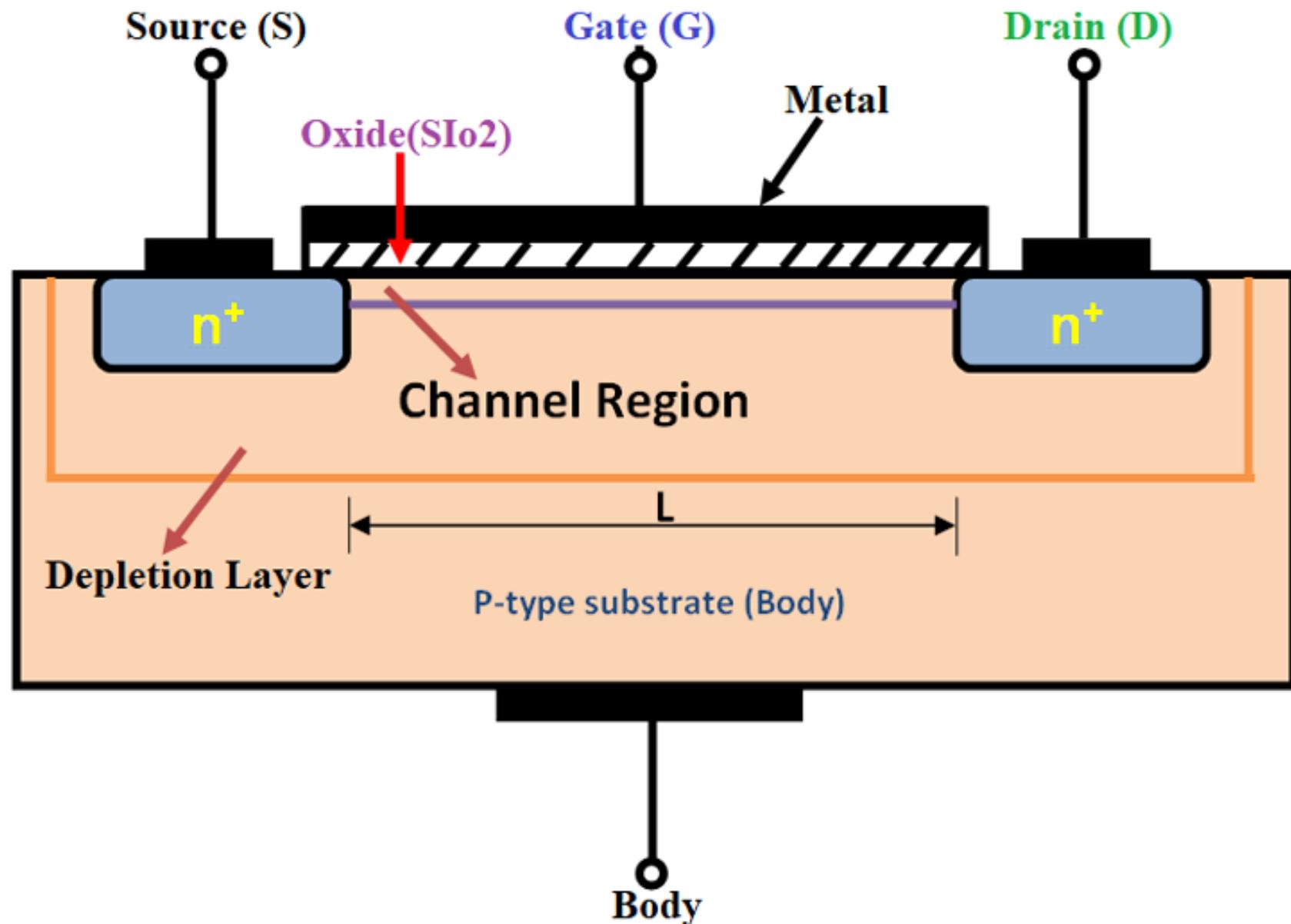


98-154/18-224/18-624: Intro to Open-Source Chip Design

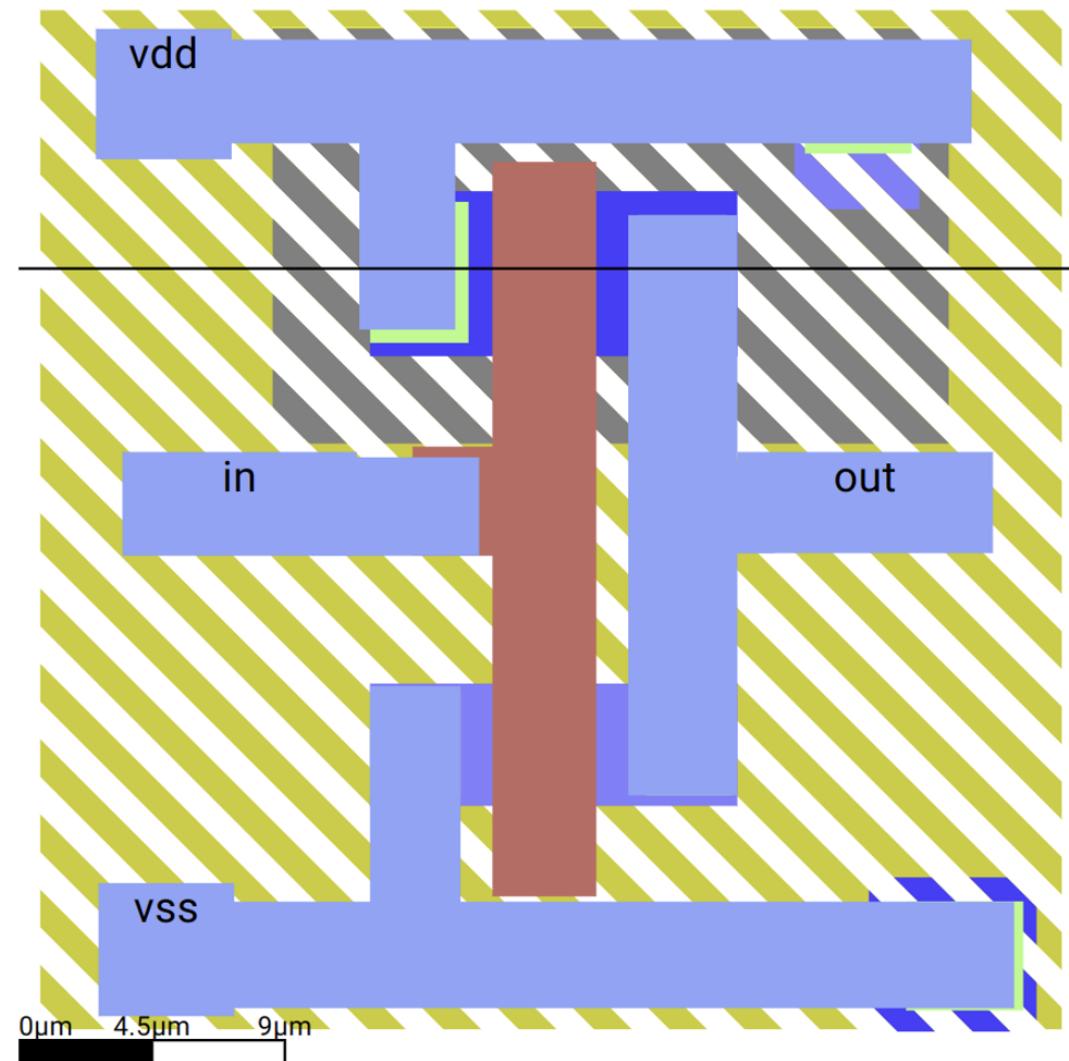
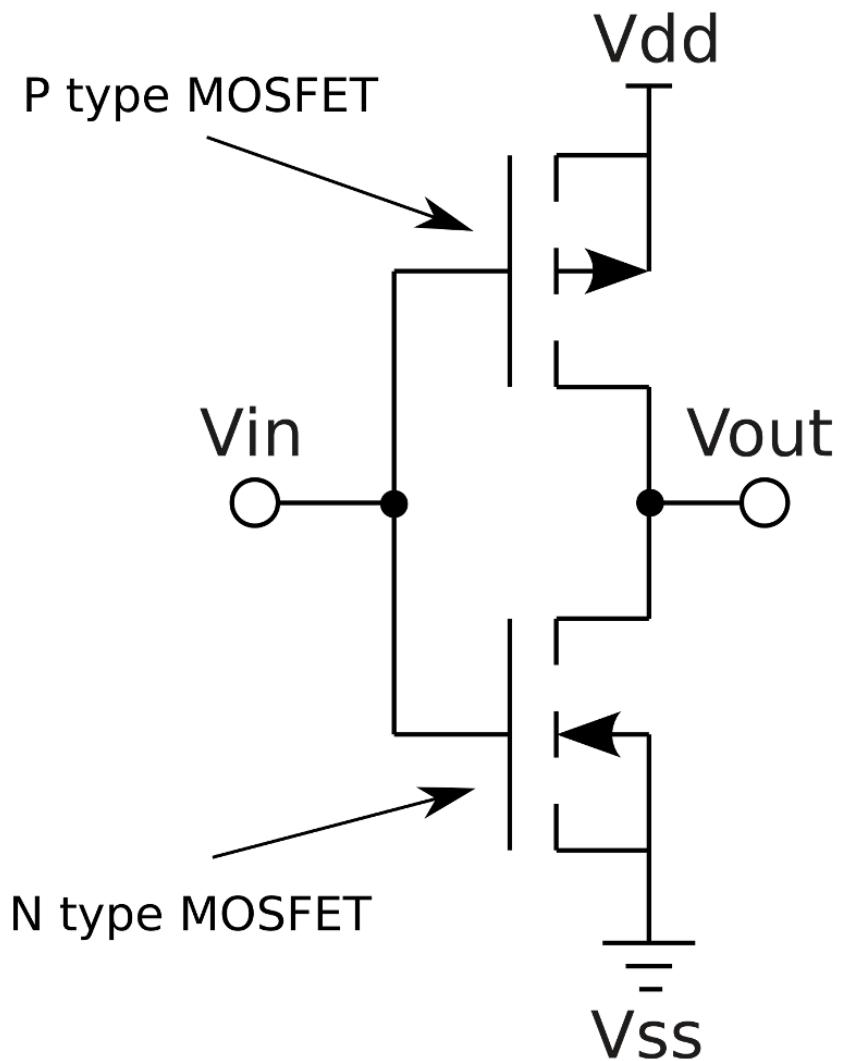
SKY130 Stackup



MOSFET



Inverter

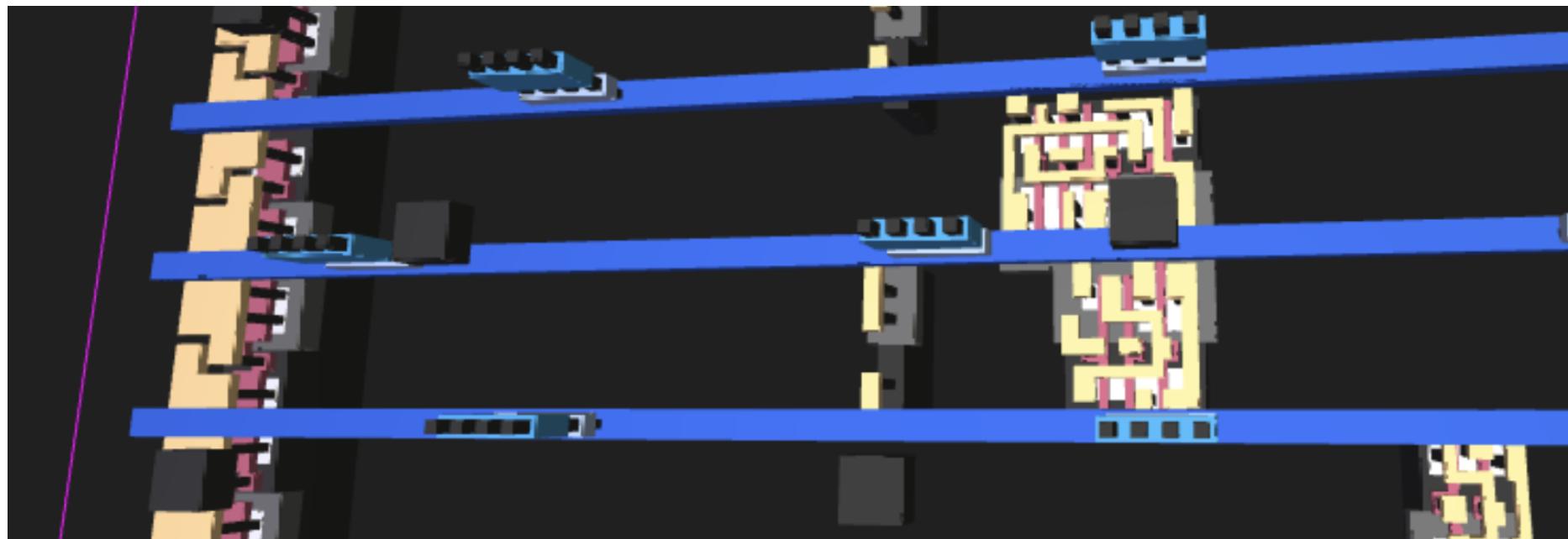


Standard Cells

- Making circuits out of individual transistors is very inefficient, because of how much variability there is
- “Standard Cell Library” (which is made by hand) contains common logic gates, flops, etc. with a standardized shape so that they can be automatically synthesized and laid out by the routing software
- [SKY130 Standard Cells](#)

Standard Cells: Power Rails

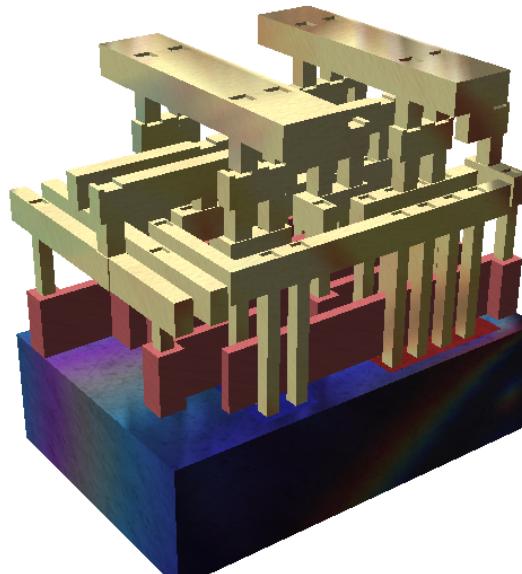
- All standard cells are a fixed height and have PWR at the top and GND at the bottom
- Hence, can lay out alternating power and ground rails and place standard cells in alternating orientations



Physical Design Representation Formats

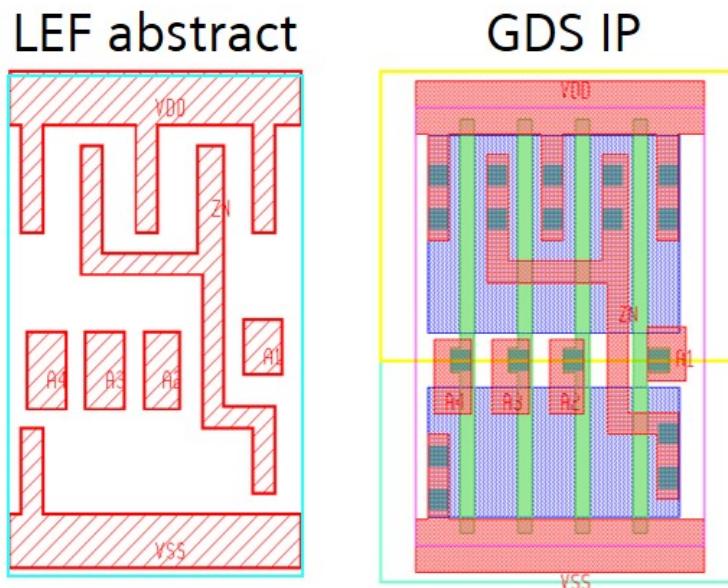
.GDS

- GDSII (Graphical Design System file)
- Hierarchical file format representing geometric shapes and layout in a design
- Usually what you provide to the fab to be taped out



.LEF

- Library Exchange Format
- Abstracts away the internals of cells but maintains geometry
- Can also be used to abstract “macros” (harden a set of standard cells into one LEF “cell”)



.DEF

- Design Exchange Format
- Used in conjunction with LEF files to lay out designs at a standard-cell level

.MAG

- Magic VLSI file format
- Used by a tool called Magic which handles a lot of file-generation and processing steps as part of the open-source VLSI flows

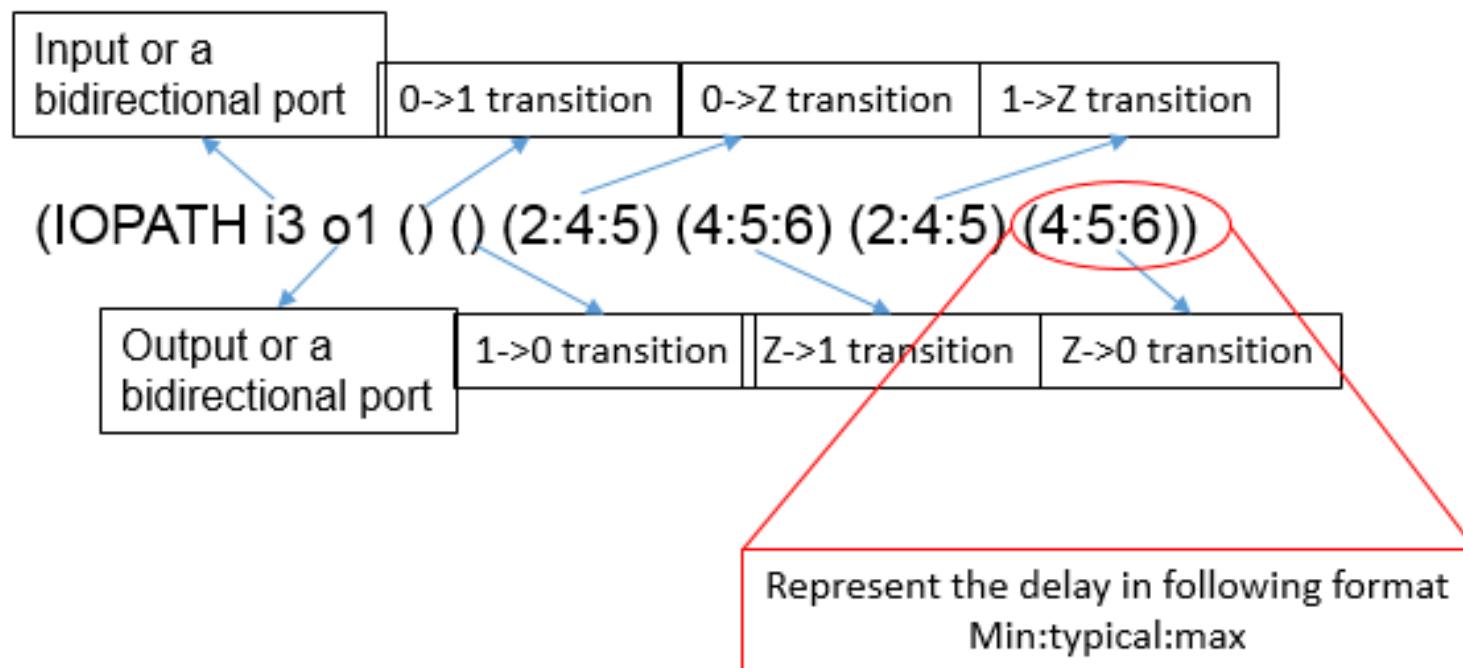
.SDC

- Synopsys Design Constraints
- Used to represent timing and drive-strength constraints on I/Os

```
create_clock -name __VIRTUAL_CLK__ -period 24.0000
set_clock_uncertainty 0.2500 __VIRTUAL_CLK__
set_input_delay 4.8000 -clock [get_clocks {__VIRTUAL_CLK__}]
    -add_delay [get_ports {io_in[9]}]
set_load -pin_load 0.0729 [get_ports {io_out[1]}]
set_driving_cell -lib_cell gf180mcu_fd_sc_mcu7t5v0_inv_1 -
    pin {ZN} -input_transition_rise 0.0000 -
    input_transition_fall 0.0000 [get_ports {wbs_sel_i[3]}]
```

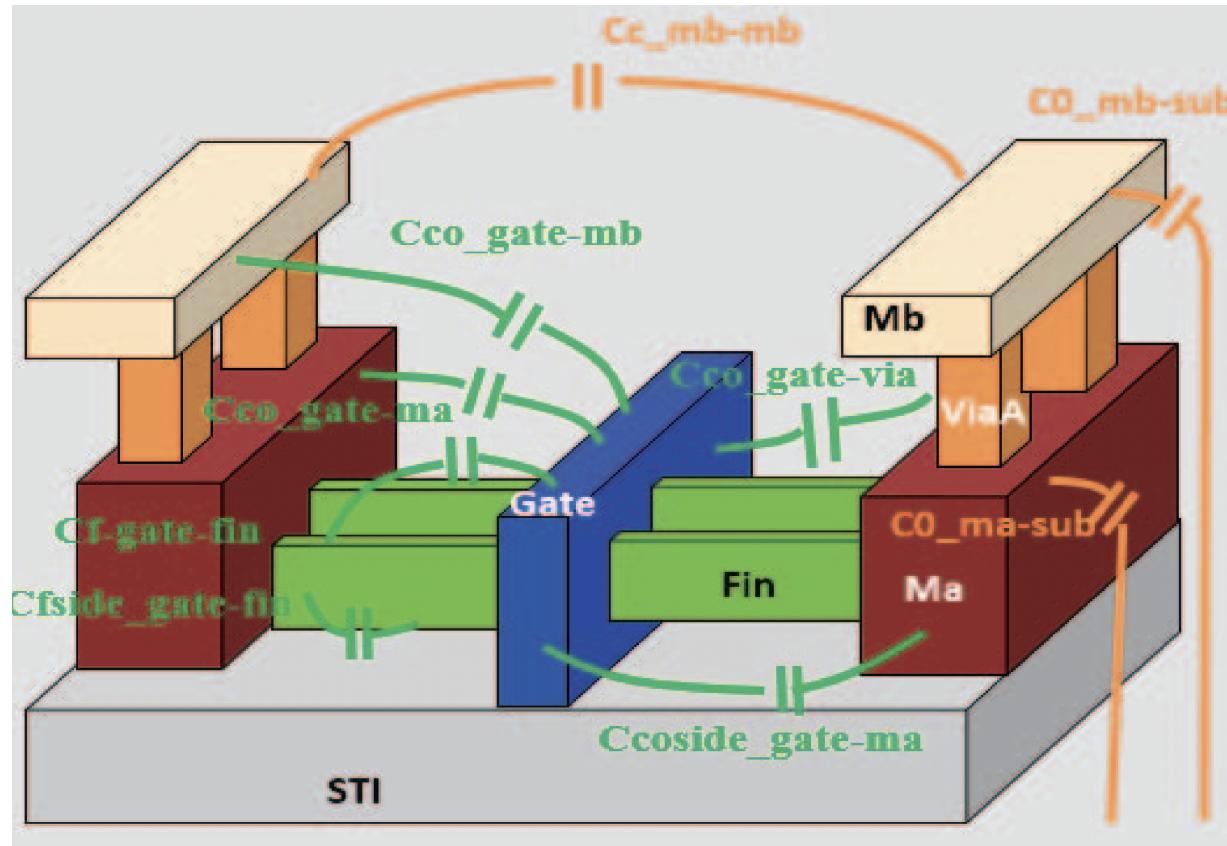
.SDF

- Standard Delay Format
- Used to represent timing and delay measurements of components of a design



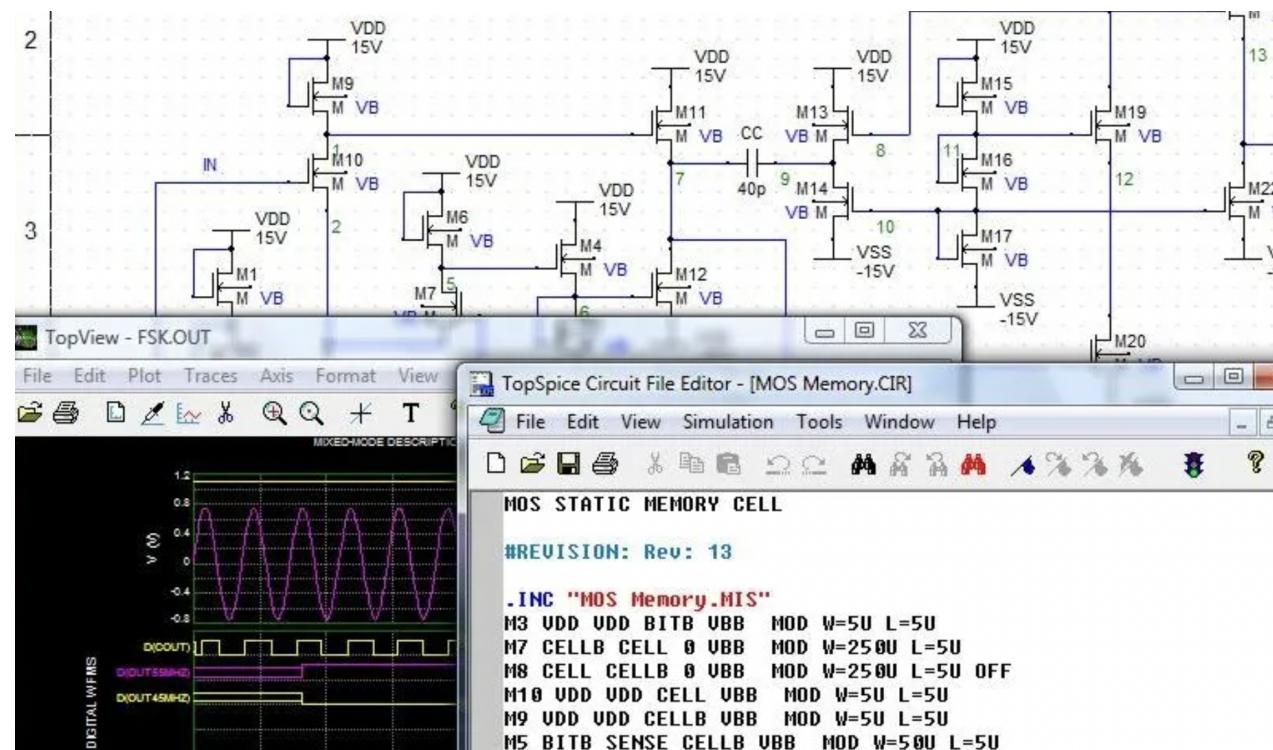
.SPEF

- Standard Parasitic Exchange Format
- Used to represent result of parasitic extraction of a design



.SPICE

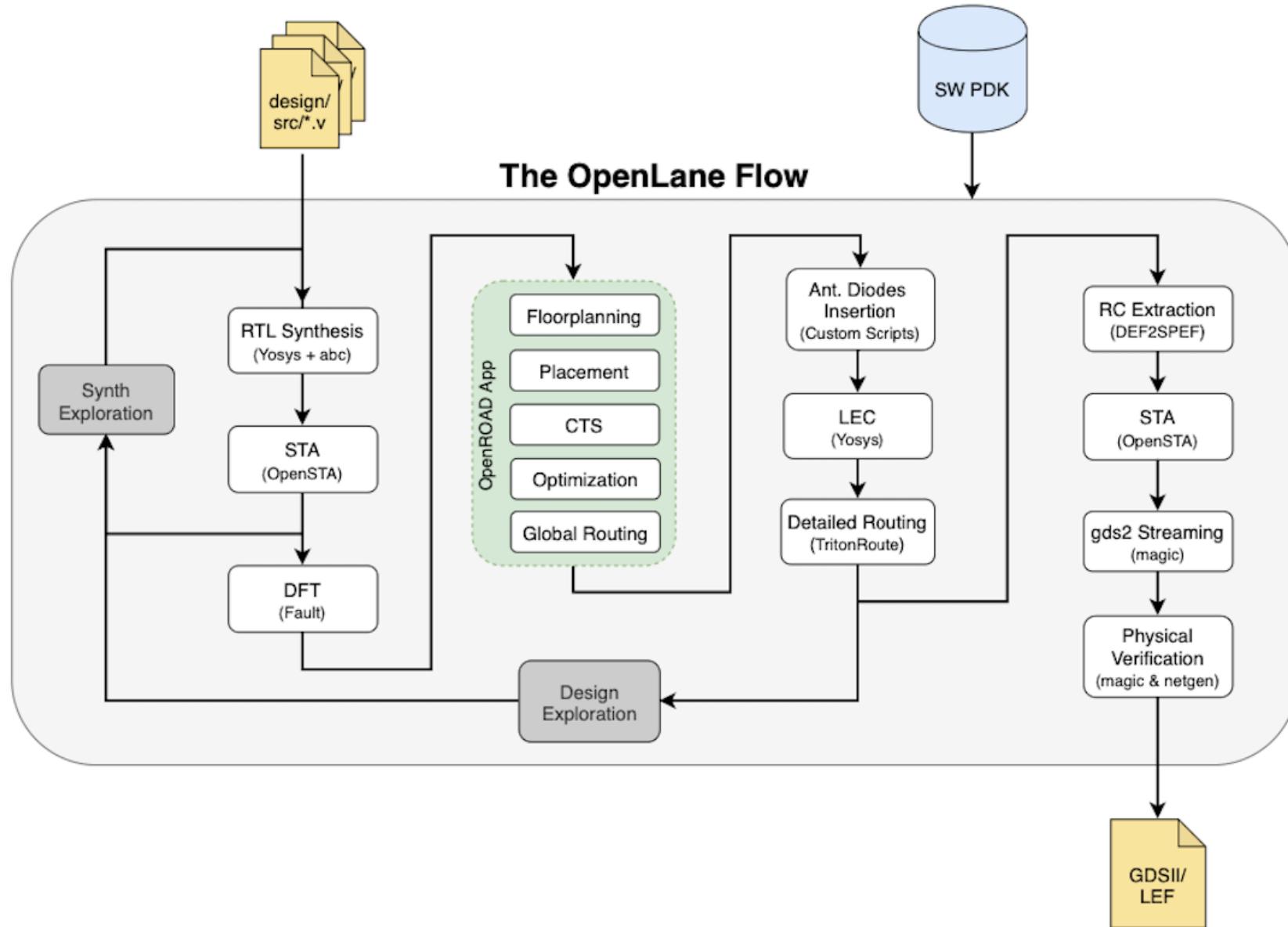
- Simulation Program with Integrated Circuit Emphasis
- SPICE is a file format, a specific software, and a general category of softwares; focused on analog-level simulation



OpenLane

- Currently the most feature-complete open-source ASIC design flow
- Has been used for successful tapeouts on the SKY130 process node
- Released in 2020 alongside the SKY130 process, more recently added support for GF180

OpenLane Flow



Synthesis

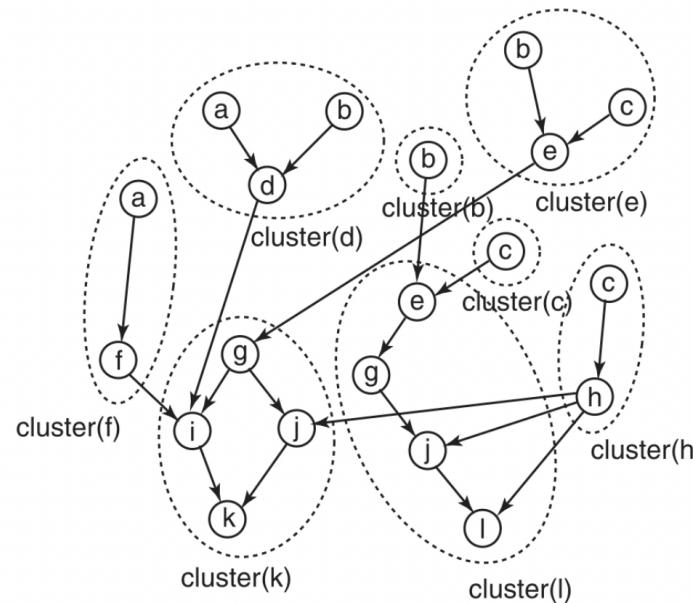
- Yosys targeted to the SKY130 standard cell library as a backend

Design for Test

- Tools such as Fault can be used for scan-chain insertion to introspect design after tapeout
- Will have full lecture on DFT and bringup later in the semester

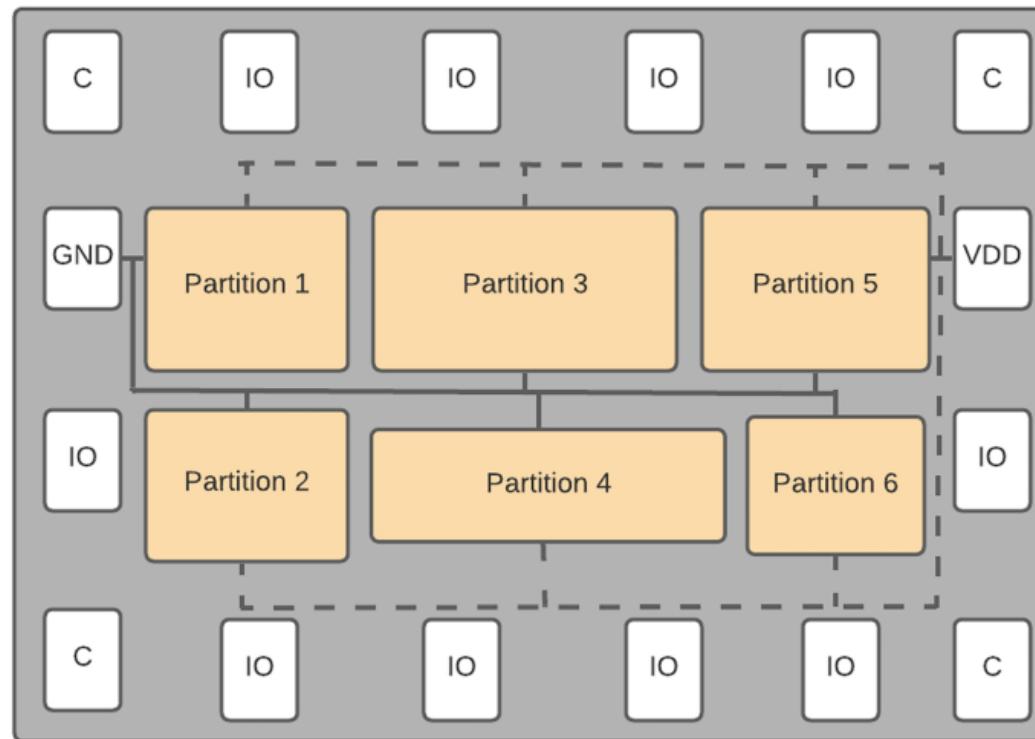
Partitioning

- Split up the synthesized netlist into interconnected components (graph clustering)
- Overpartitioning and underpartitioning can both be bad for floorplanning

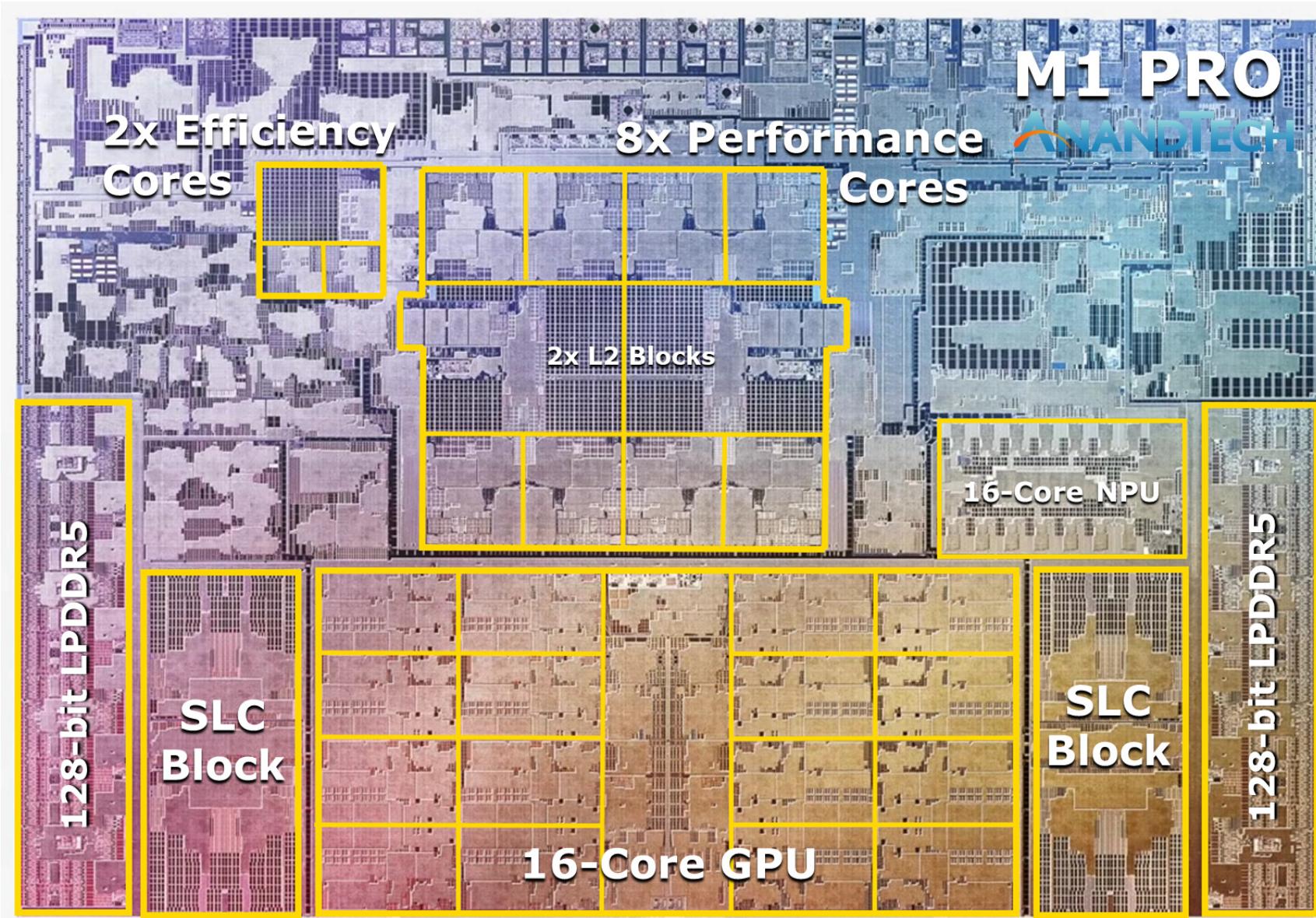


Floorplanning

- Lay out partitioned components in the 2D space of the chip
- Often done manually (for particularly complex designs) but can also be done automatically by optimizing wirelength



A more familiar example

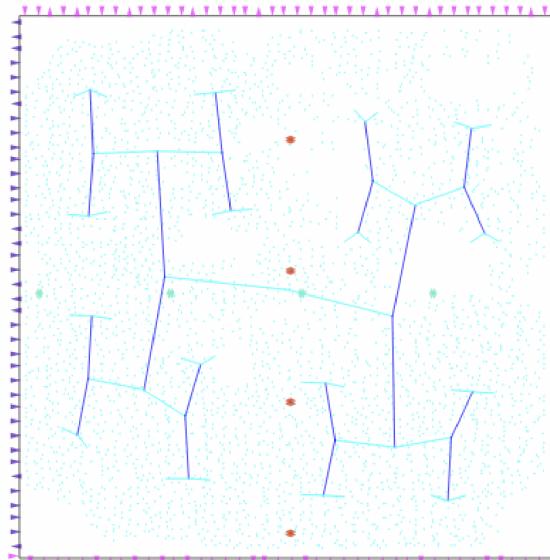


Placement

- Given a set of standard cells, place them on the 2D space of the chip while minimizing wire length
- Best-case wire length (Steiner tree) for a net is an NP-hard problem to compute, heuristics like HPWL and MST are used
- Timing estimations can be used as well (maximize positive timing slack)
- Can run into issues with congestion if the heuristic does not account for spreading out cells

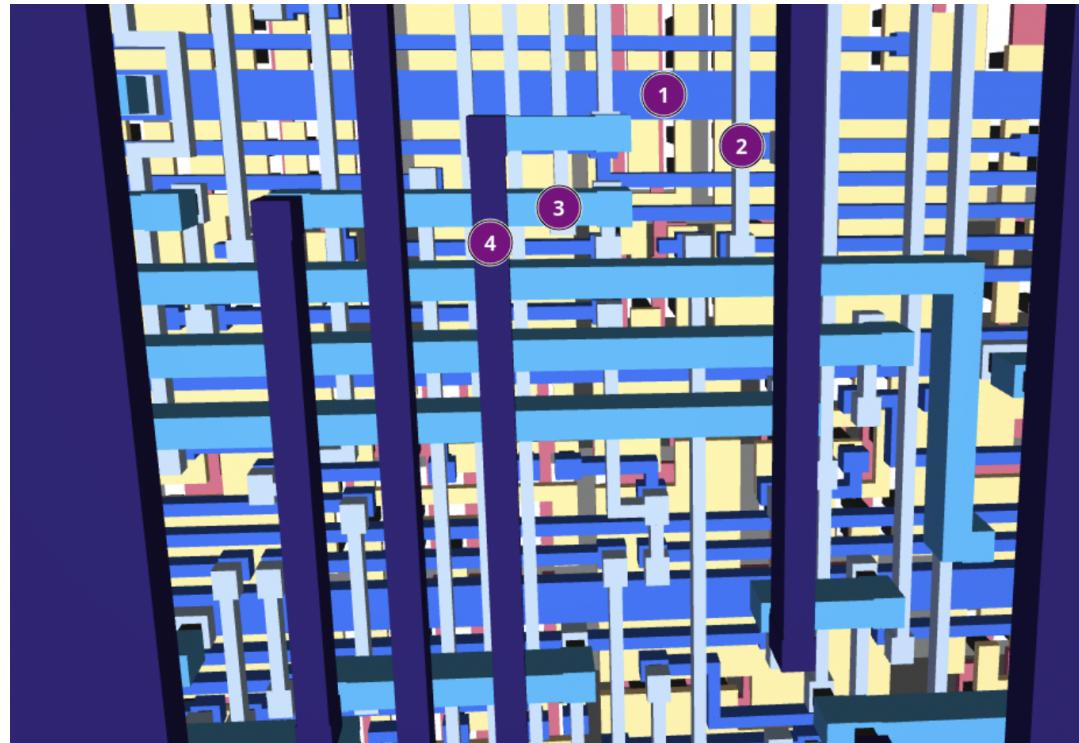
Clock Tree Synthesis

- Every sequential element relies on clocks
- Minimize clock skew; ideal case is a H-tree fractal but wire-length is high
- Insert clock buffers where needed

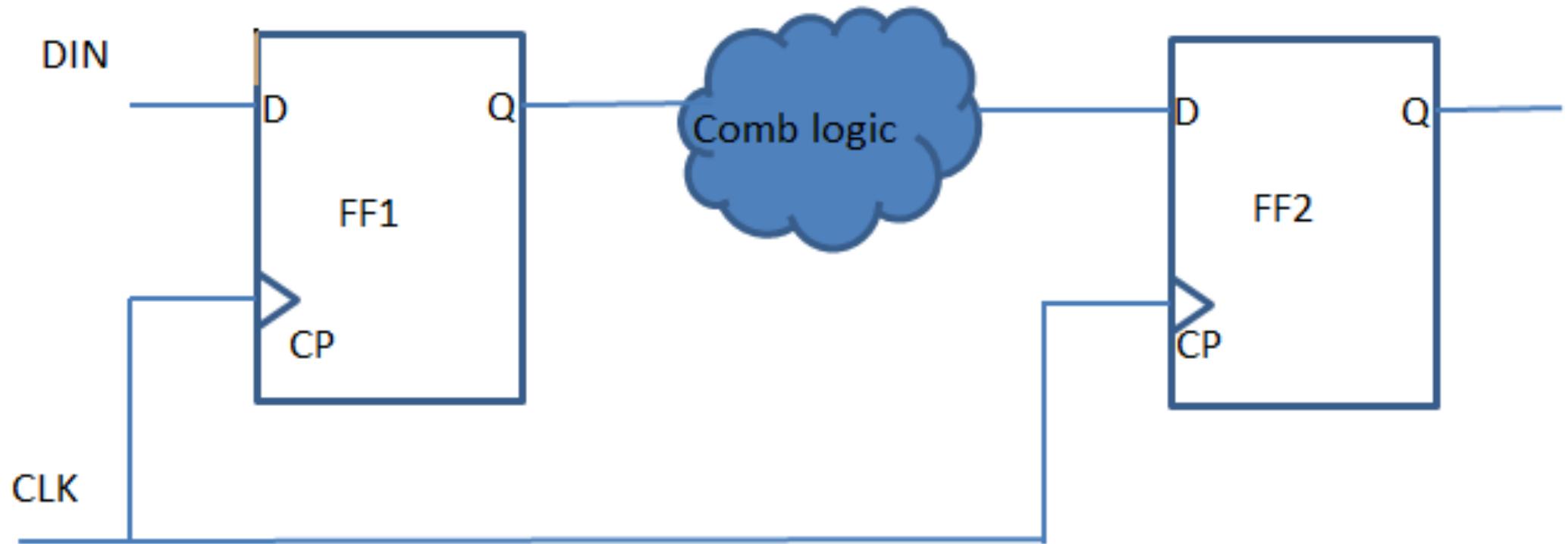


Routing

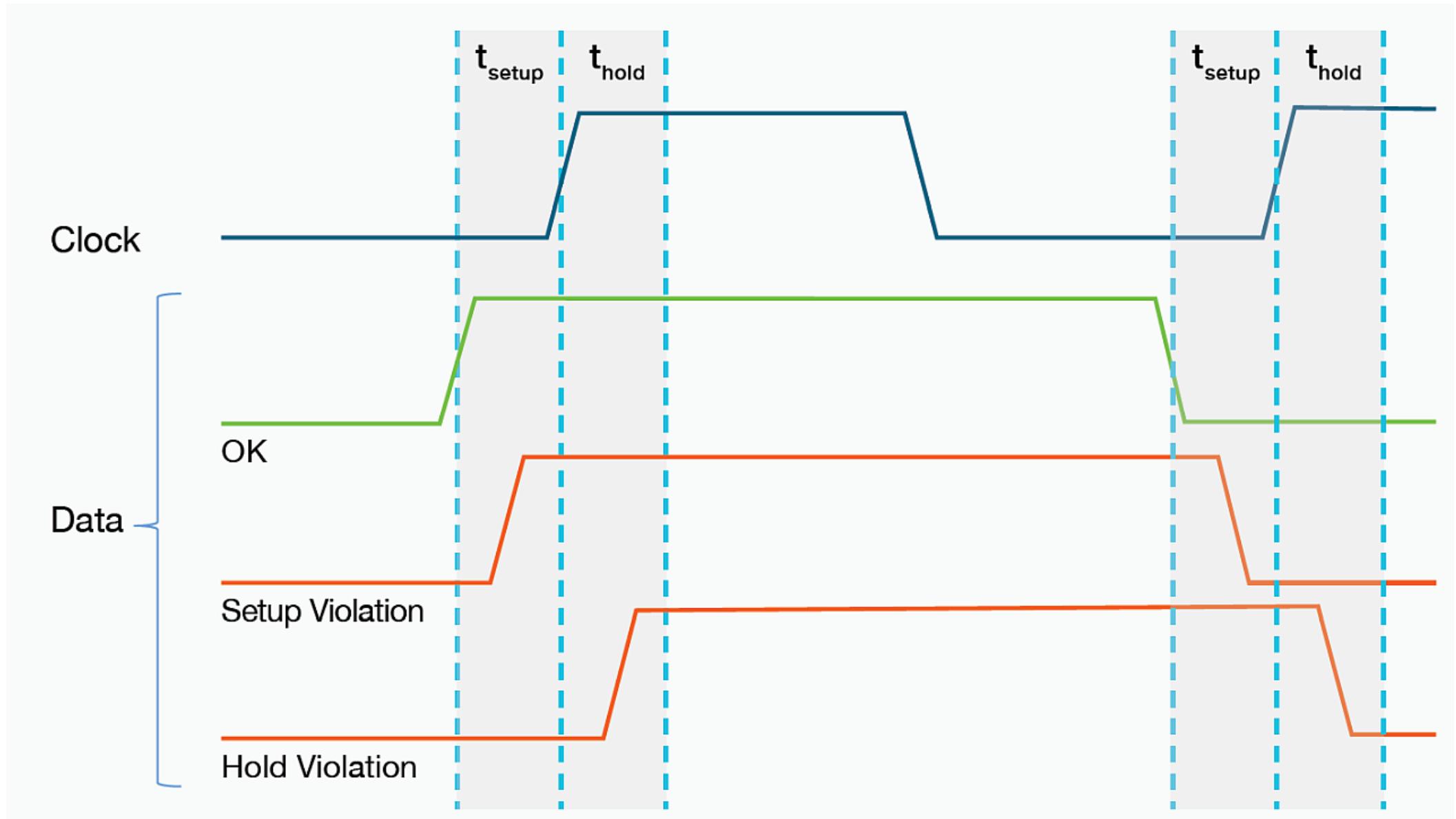
- Find optimal path to connect together all the nets
- Based on search algorithms such as A*, Mikami-Tabuchim, Hightower



Timing Violations



Timing Violations



Static Timing Analysis

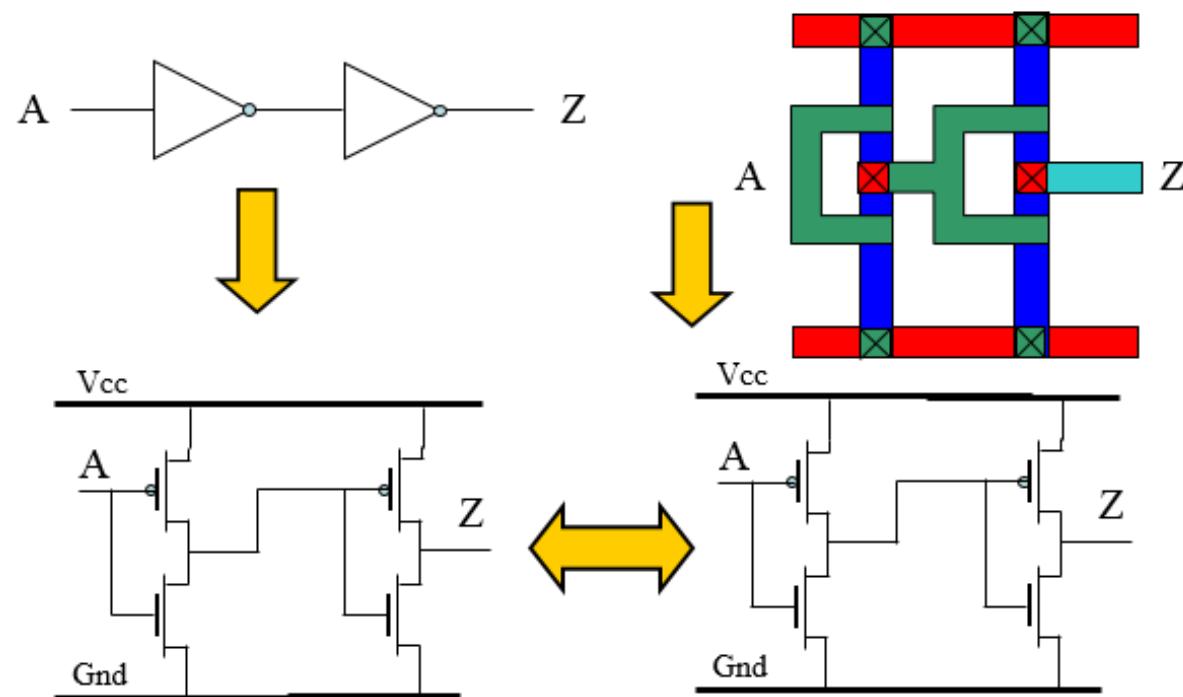
Ensure that the design meets timing constraints

The propagation delay between FF in a functional circuit has both an upper and lower bound defined by the hold and setup times, desired frequency, and clock skew. Formally, with $t_{cycle} = \frac{1}{f}$:

$$t_{cycle} \geq t_{delay} + t_{setup} + t_{skew}, t_{delay} \geq t_{hold} + t_{skew}. \quad (5.2)$$

LVS

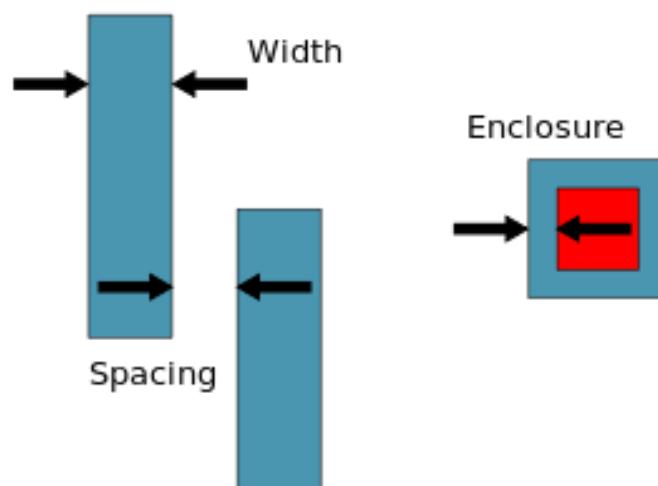
- Layout vs Schematic - extract the logic from the generated layout and use it to verify that there were no changes to the underlying logic as a result of software bugs



DRC/ERC

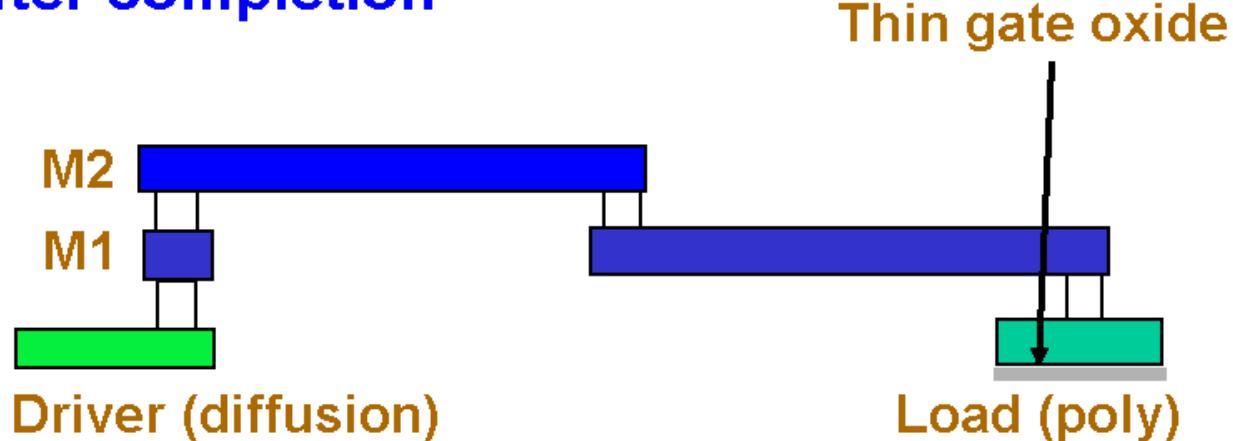
- Design Rule Checking - ensure there are no issues that would cause the circuit to be unmanufacturable (spacing, etc.)
- Electrical Rule Checking - ensure there are no electrical issues such as shorts

The three basic DRC checks

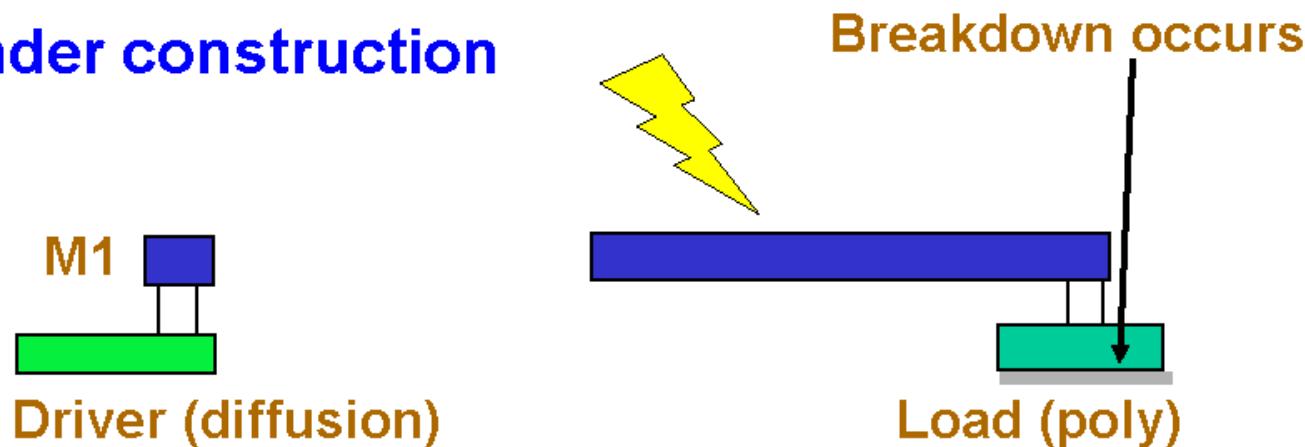


Antenna Diodes

(a) After completion



(b) Under construction



Interactive Notebook

<https://colab.research.google.com/gist/proppy/5ca7ad39f7c72464playground-conda.ipynb>