Introduction to Physical Design / APR

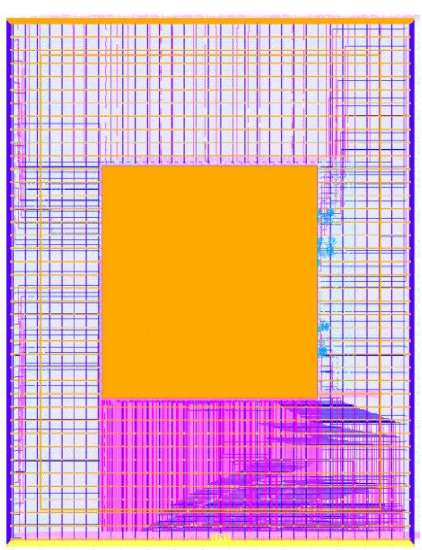
Workshop Session 1

Presented by
Hafiz Wajeh ul Hasan

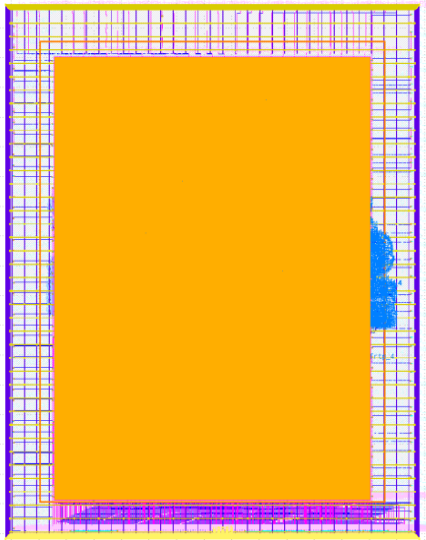
https://github.com/merledu
https://www.merledupk.org

Our Tape-outs

- We have taped-out 5 designs on Google shuttle program



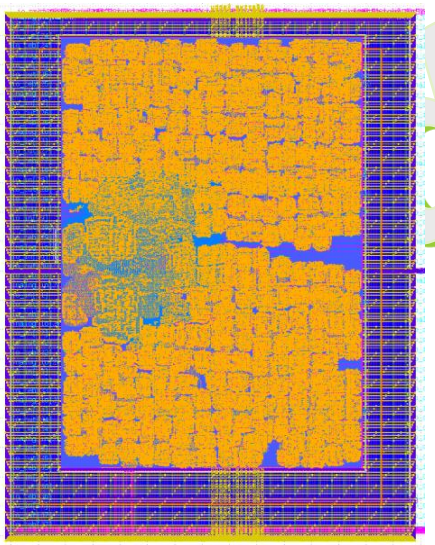
Ibtida SoC



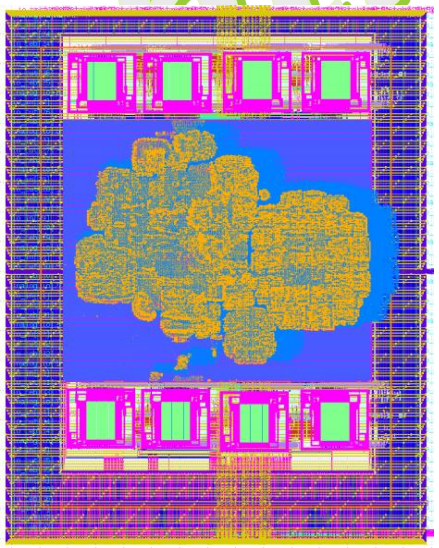
Ghazi SoC

MPW-1 | Nov 2020

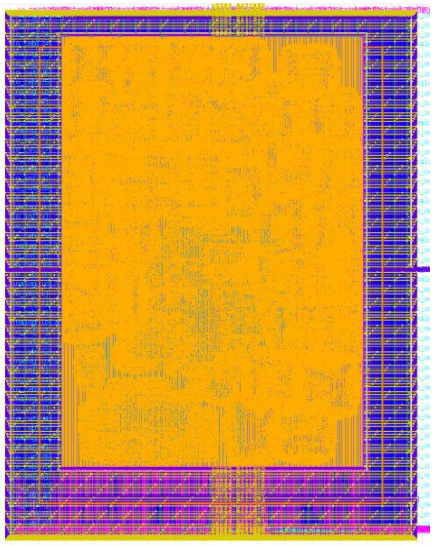
STATUS: In the midst of fabrication



Azadi SoC



Lexicon SoC



BrqRV SoC

MPW-2 | June 2021

STATUS: In the midst of verification before fabrication

GitHub Links

1.https://github.com/hadirkhan10/caravel_ibtida_soc
2.https://github.com/merledu/caravel_Ghazi_soc
3.hhttps://github.com/merledu/caravel_azadi_soc
4.hhttps://guthub.com/wajehulhasan/caravel_Lexicon
5.https://github.com/HamzaShabbir517/caravel_BrqRV_EB1

Instructions

- Post the questions in the comment box
- You can post the questions on slack workspace as well, the link is provided in the description
- You should also install VM in your PCs/laptops with the Sky130 PDK installed using the links provided on the slack
- In case you are encountering any problems during installation, post them on the slack workspace

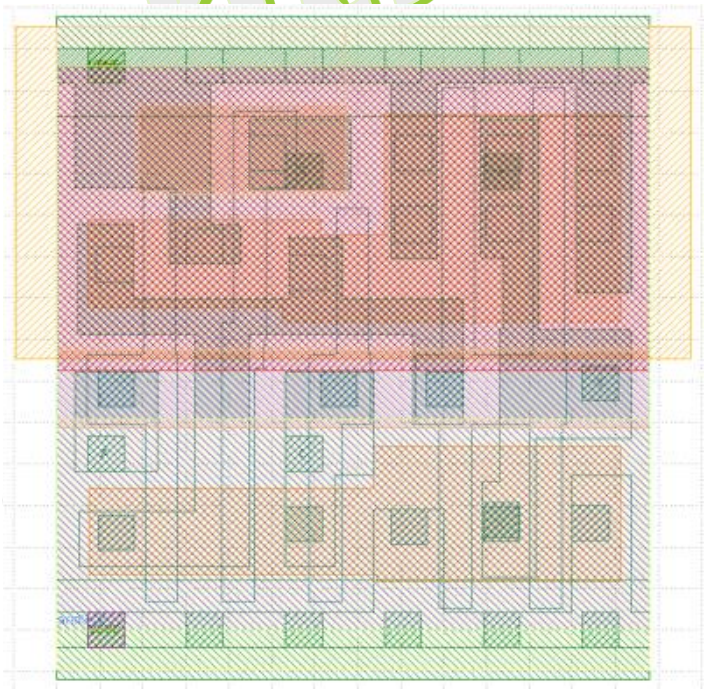
Contents

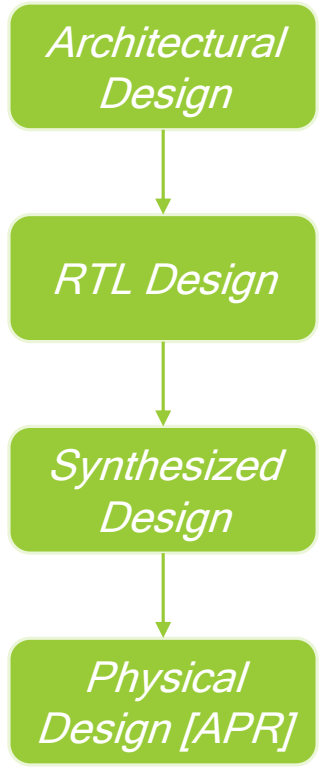
- Design Flow
- Physical Design
- Standard Cells
- Stick Diagrams
  - Inverter
  - NAND
  - NOR
- APR
  - APR Flow

Design Flow

An overview of the design flow

- Step 1: Defining the architecture of your design
- Step 2: Writing the design abstraction for the architecture
  - This could be done using VHDL, Verilog or System Verilog
- Step 3: This code is then synthesized to Gate Level Circuit Design/Netlist
  - The circuit design level can also be referred to transistor level, inside every Gate/Standard Cell we have some transistors
- Step 4: **Physical Design**, this is the process of transforming a circuit description into the physical layout.



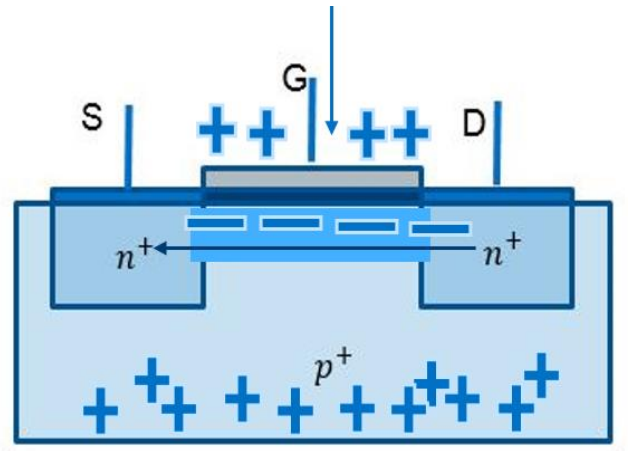


Physical Design

- The physical design is where the circuit level is converted to a complete physical geometric representation. This geometric representation is called integrated circuit layout

- During the Physical Design, we place and route Logic Gates[Standard Cells]. So the basic building block for the complete design are the Standard Cells

- Standard Cells comprises of Transistors[CMOS]

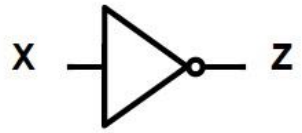


Physical view of a simple planar transistor

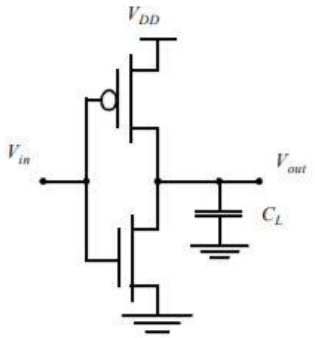
Standard Cells

- The basic Standard Cells are:

Inverter



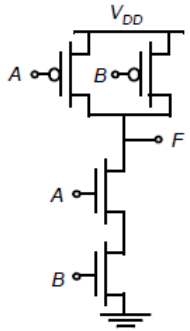
Symbolic View



Schematic View

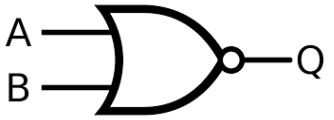
NAND

Symbolic View

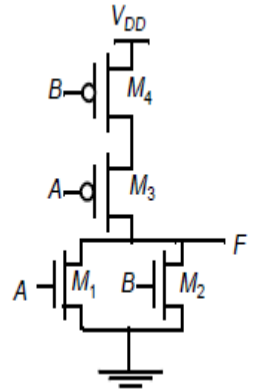


Schematic View

NOR



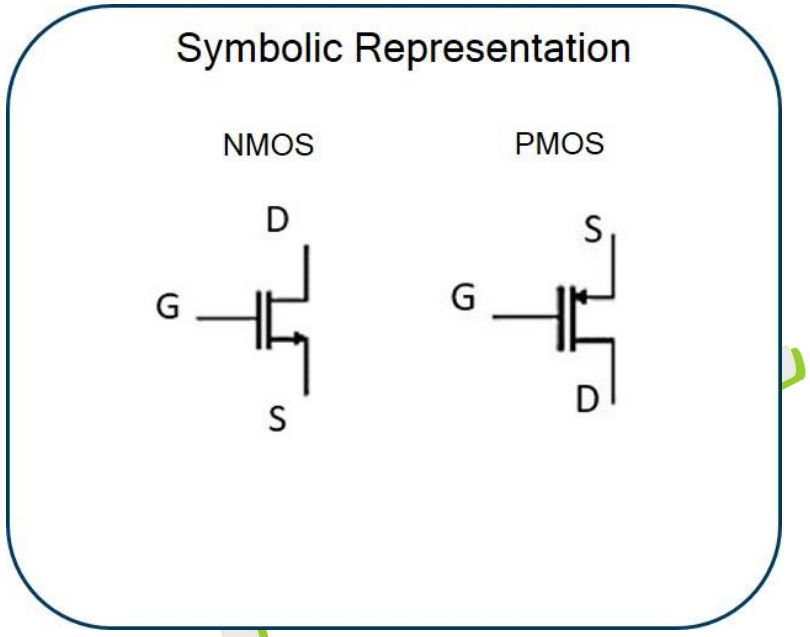
Symbolic View

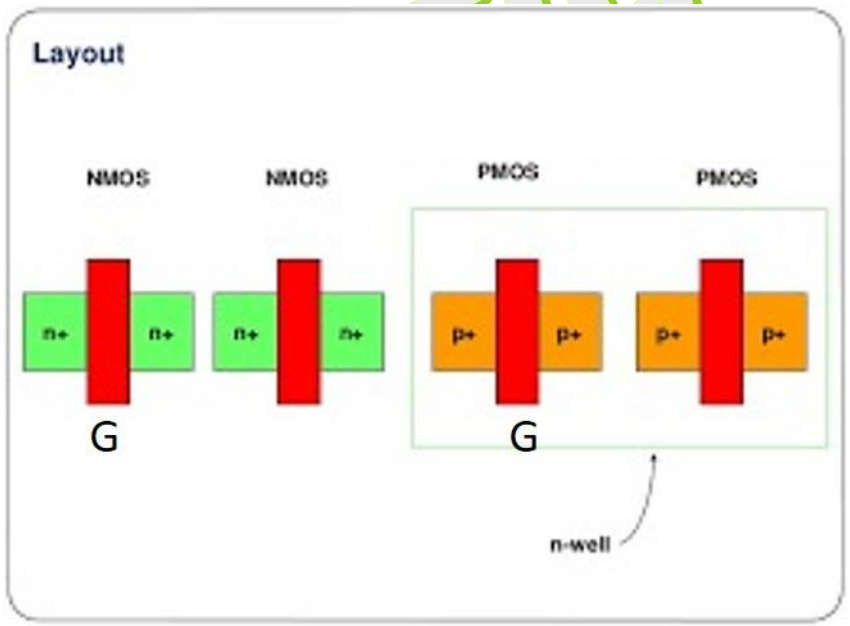


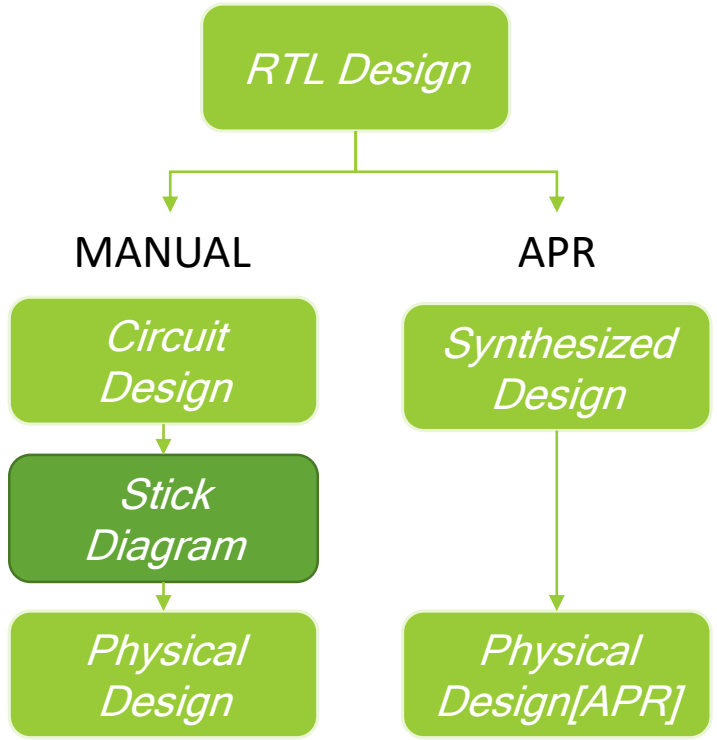
Schematic View

Stick Diagram

- It is basically an interface between symbolic circuit and the actual layout.
- Stick diagrams convey layer information through color codes (or monochrome encoding).



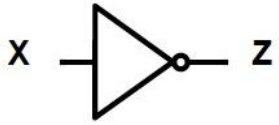


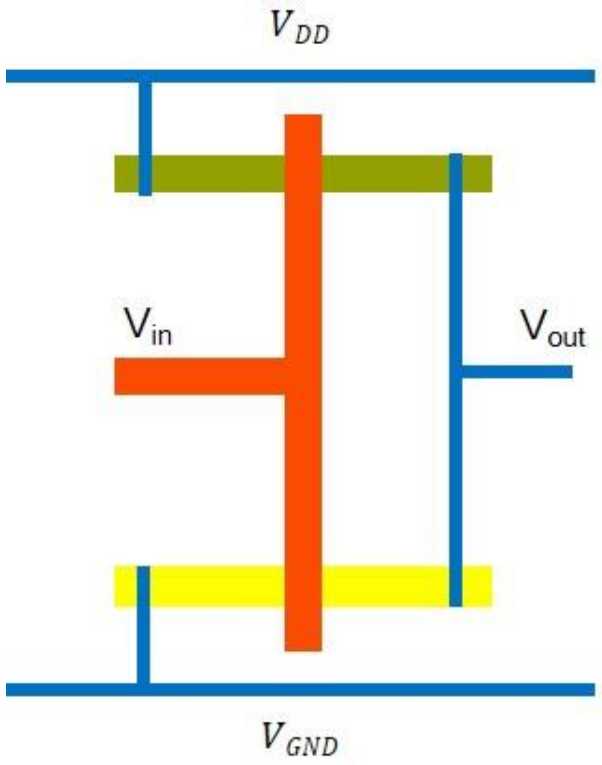


Inverter

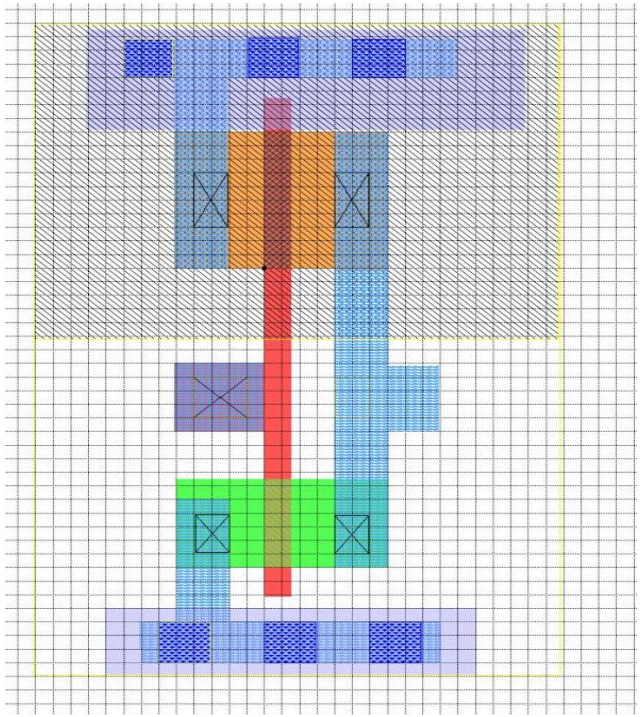
- Step 1: RTL code
- Step 2: Logic design
- Step 3: *Schematic* level design

$$Z = \sim X$$





Actual Layout



METAL

POLY

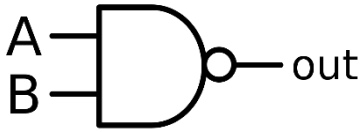
P Diff

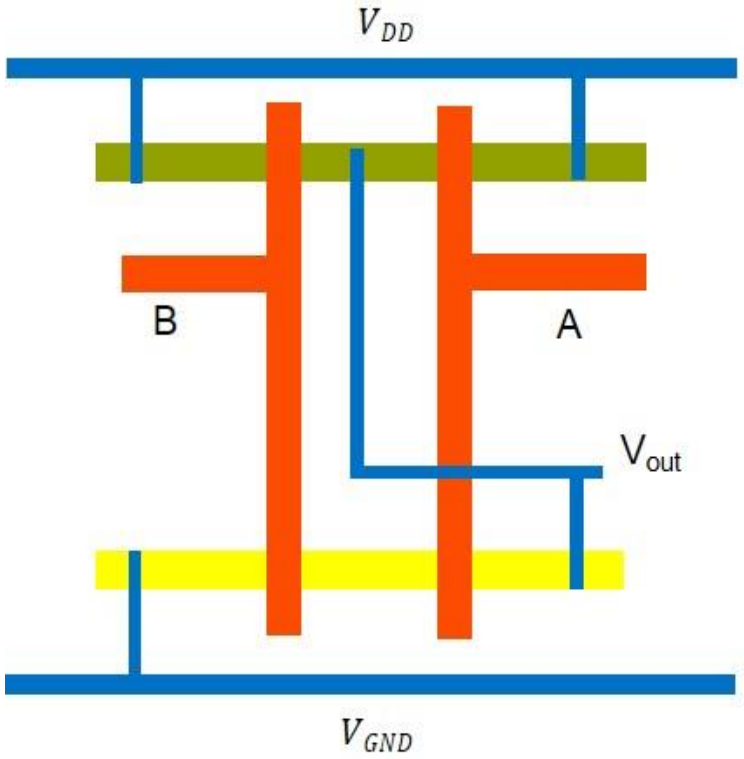
N Diff

NAND

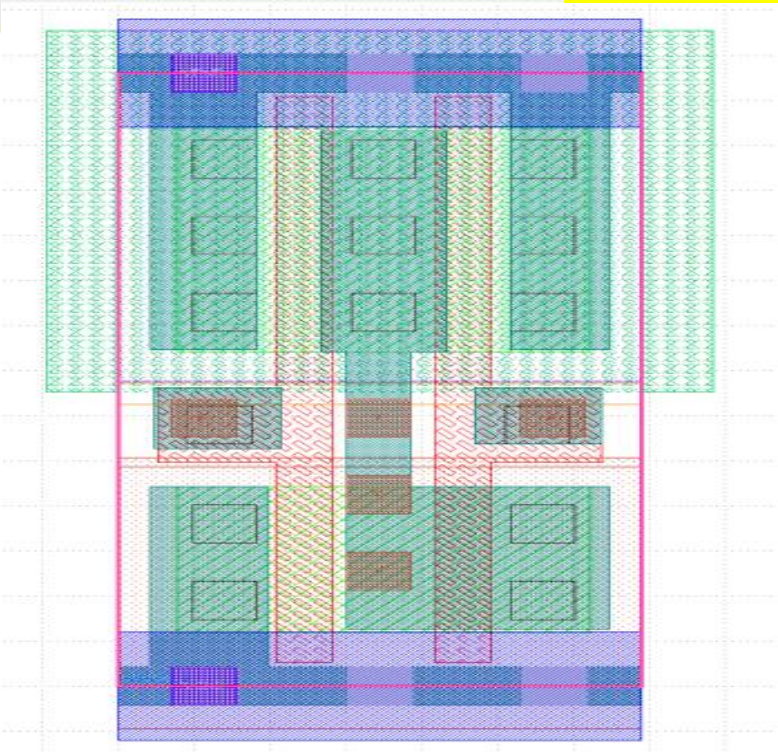
- Step 1: RTL code
- Step 2: Logic design
- Step 3: *Schematic* level design

$$X = \overline{A.B}$$





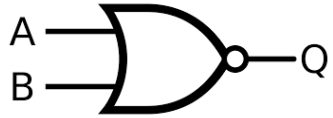
Actual Layout

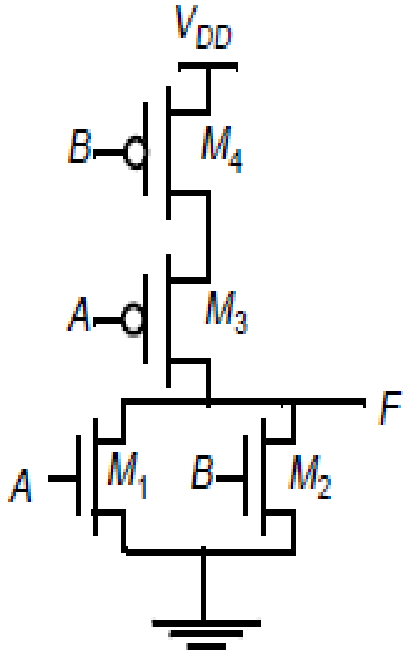


NOR

- Step 1: RTL code
- Step 2: Logic design
- Step 3: *Schematic* level design

$$X = \overline{A + B}$$



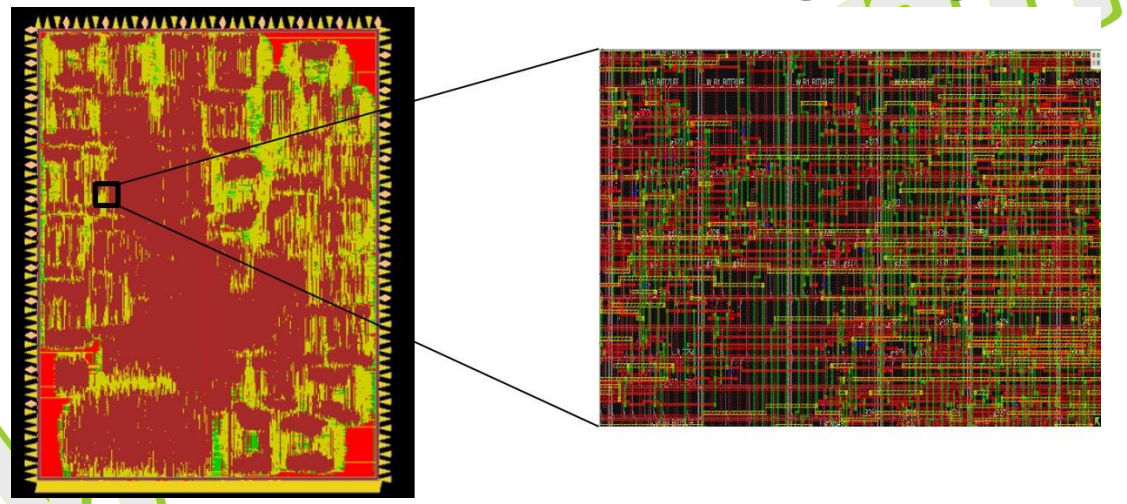


Stick Diagram

?

APR

- What is APR
  - APR is Auto Place and Route
    - Place the standard cells
    - Route the standard cells together
- Why APR?
  - Lets have a look at the following design



Note: The last design we did had **279995** Gates

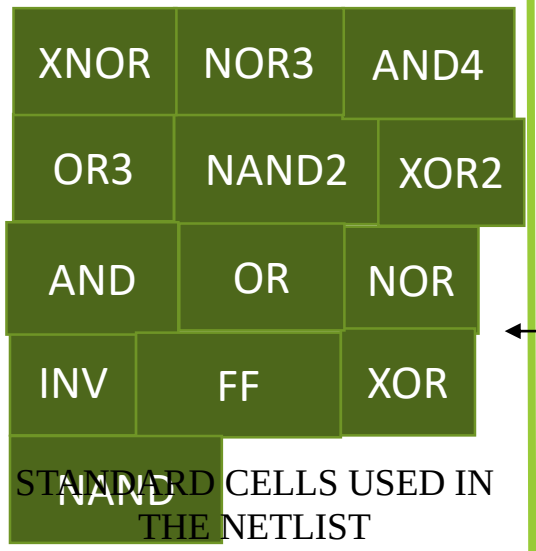
Imagine yourself placing and routing near 300,000 Gates

Now let's have an overview of the APR flow

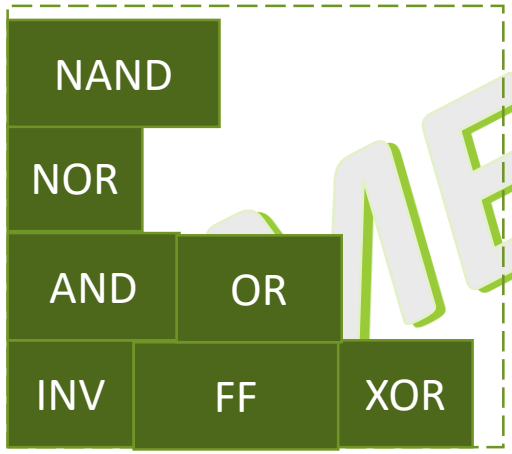
  - When the designs are too big, it is tedious to draw the layout manually

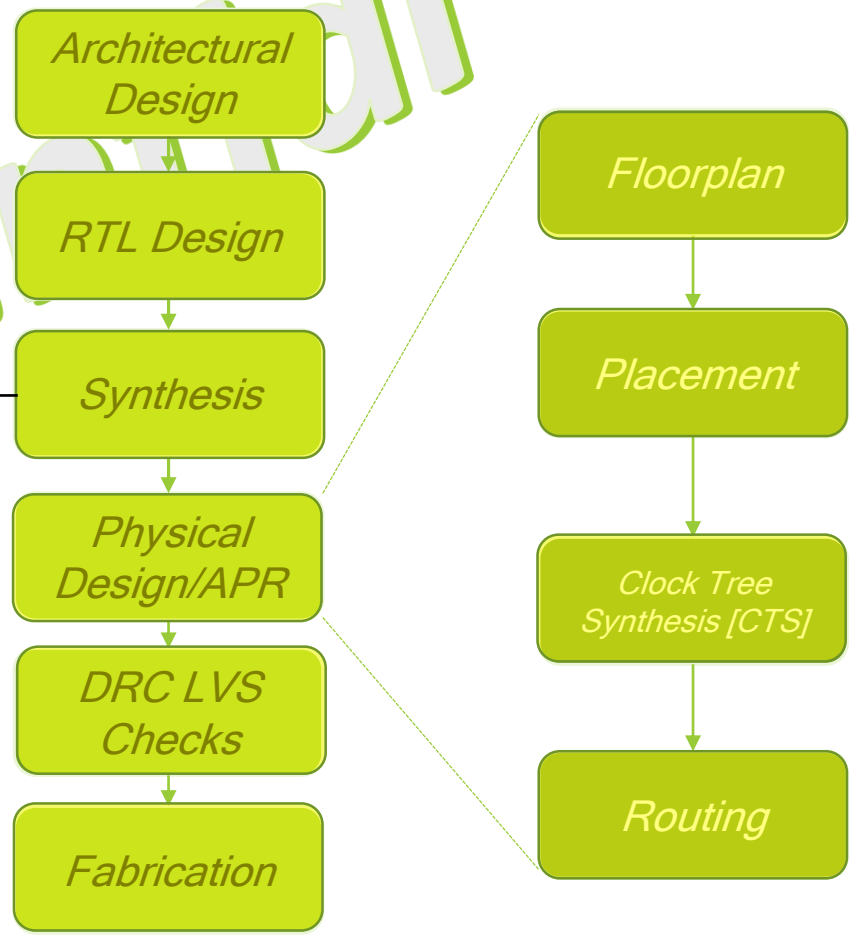
APR Flow

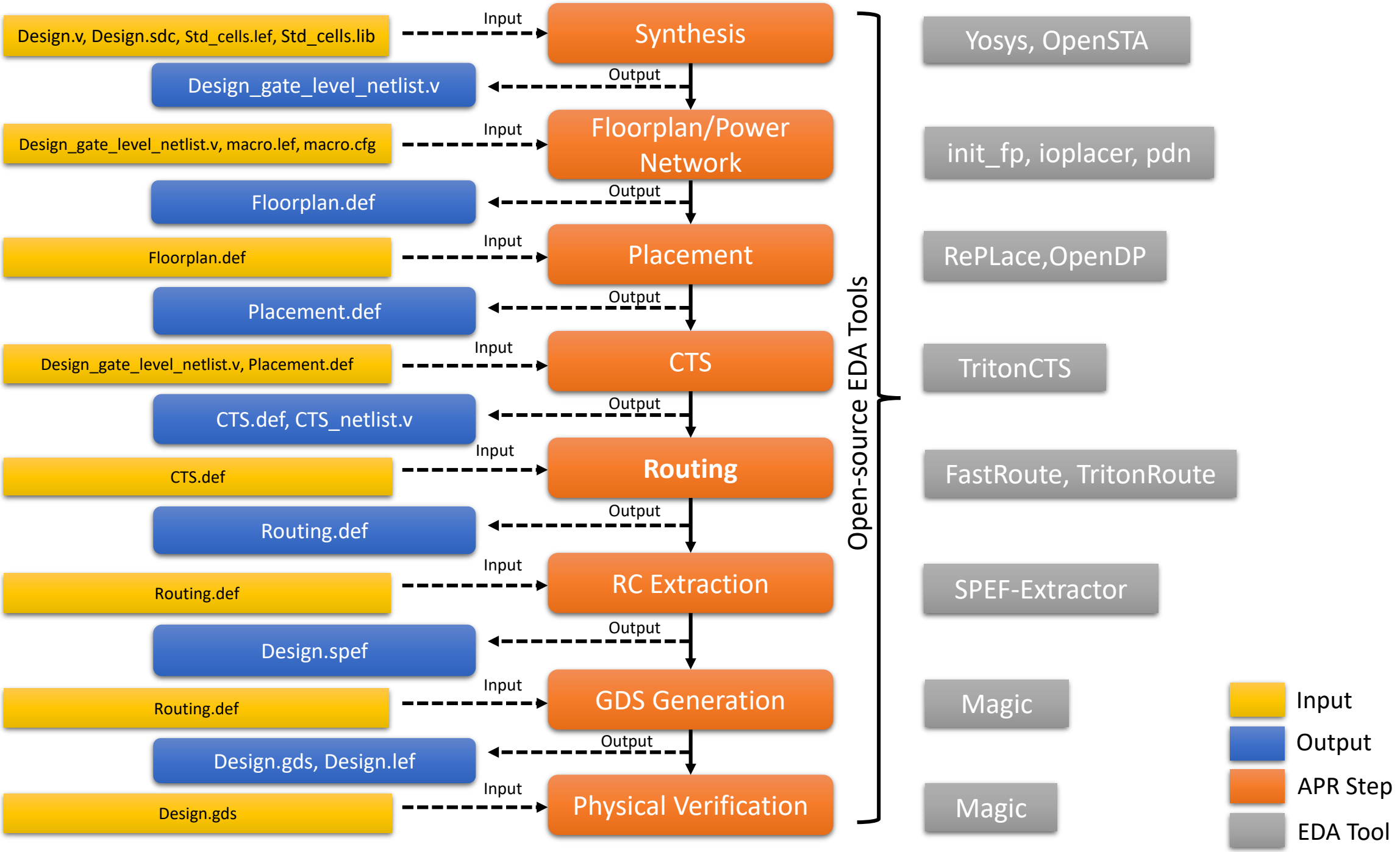
LIBRARY CONTAINING STANDARD CELLS



STANDARD CELLS USED IN THE NETLIST






Design.v, Design.sdc, Std_cells.lef, Std_cells.lib
Input
Synthesis
Output
Design_gate_level_netlist.v
Design_gate_level_netlist.v, macro.lef, macro.cfg
Input
Floorplan/Power Network
Output
Floorplan.def
Floorplan.def
Input
Placement
Output
Placement.def
Design_gate_level_netlist.v, Placement.def
Input
CTS
Output
CTS.def, CTS_netlist.v
CTS.def
Input
Routing
Output
Routing.def
Routing.def
Input
RC Extraction
Output
Design.spef
Routing.def
Input
GDS Generation
Output
Design.gds, Design.lef
Design.gds
Input
Physical Verification
Open-source EDA Tools
Yosys, OpenSTA
init_fp, ioplacer, pdn
RePLace,OpenDP
TritonCTS
FastRoute, TritonRoute
SPEF-Extractor
Magic
Magic
Input
Output
APR Step
EDA Tool

Thank you