JOINT INSTITUTE
交大密西根学院

Lab #4

Getting Started with the Automated ASIC Design Flow-based on OpenRoad

Nov. 4, 2024

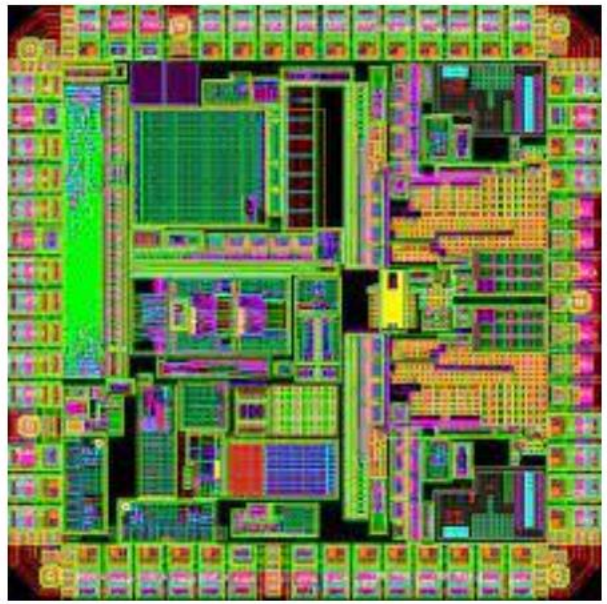
What is ASIC design flow

- How to design a chip from concept to silicon?

```
module PE (clock, R, S1, S2, S1S2mux, newDist, Accumulate, Rpipe);
input clock;
input [7:0] R, S1, S2;// memory inputs
input S1S2mux, newDist;// control inputs
output [7:0] Accumulate, Rpipe;
reg [7:0] Accumulate, AccumulateIn, Difference, Rpipe;
reg       Carry;

always @(posedge clock) Rpipe <= R;
always @(posedge clock) Accumulate <= AccumulateIn;

always @(R or S1 or S2 or S1S2mux or newDist or Accumulate)
  begin  // capture behavior of logic
   difference = R - S1S2mux ? S1 : S2;
   if (difference < 0) difference = 0 - difference;
// absolute subtraction
    {Carry,AccumulateIn} = Accumulate + difference;
    if (Carry == 1) AccumulateIn = 8'hFF;// saturated
    if (newDist == 1) AccumulateIn = difference;
// starting new Distortion calculation
  end
endmodule
```

Motion Estimator Processing Element (PE).

RTL (.v)



Layout

What is ASIC design flow & EDA tool

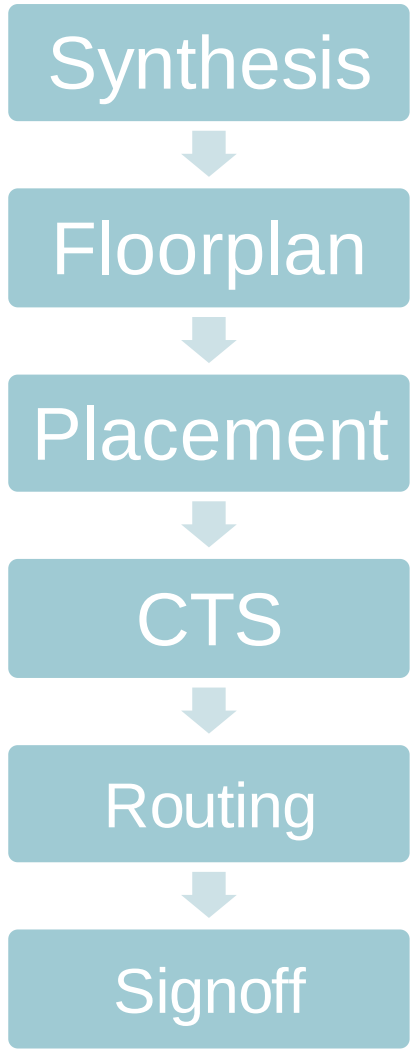
- Electronic Design Automation (EDA) Tool
  - enable billion-traistor level chips
  - use lots of algorithms to enable the design automation
  - divide the whole flow into small steps
- Commercial Tool & Open-source Tool

XILINX cadence

SYNOPSYS® KEYSIGHT TECHNOLOGIES

ANSYS Mentor Graphics®

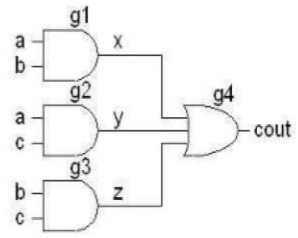
OpenROAD



Synthesis

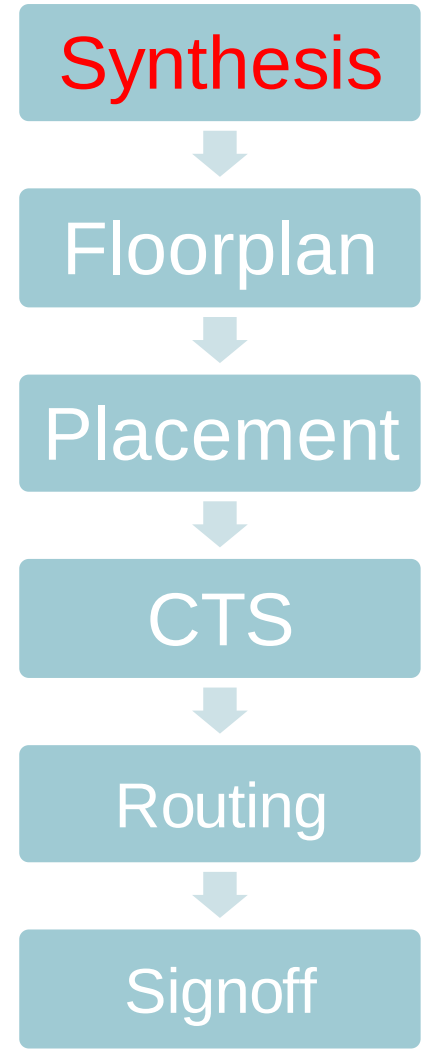
- Input
  - RTL design (Verilog/SV code)
  - .sdc file (design constraints file)
- Output
  - gate level netlist

```
module carry(input  a, b, c,
             output cout)

standard cell name
      wire   x, y, z;

      and g1(x, a, b);
      and g2(y, a, c);
      and g3(z, b, c);
      or  g4(cout, x, y, z);
endmodule
```



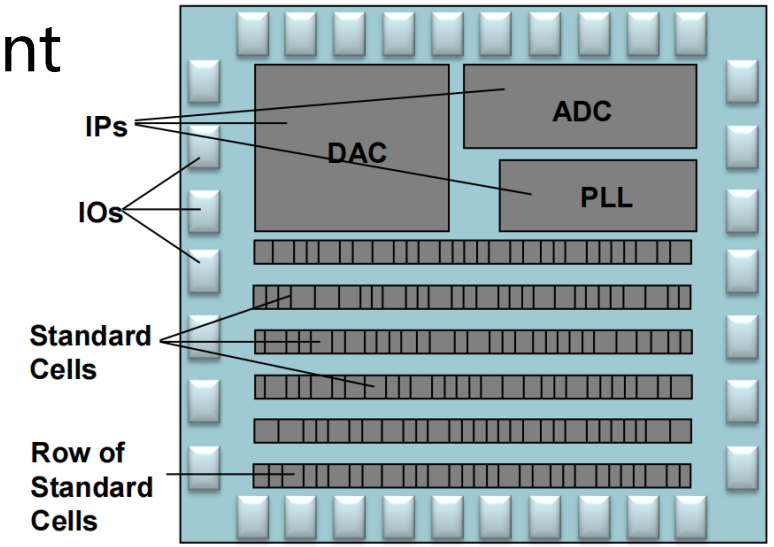
For a same logic, it may have multiple standard cells with different size etc.

For example:
NAND2_X1
NAND3_X1
NAND4_X1
NAND4_X2



Floorplan

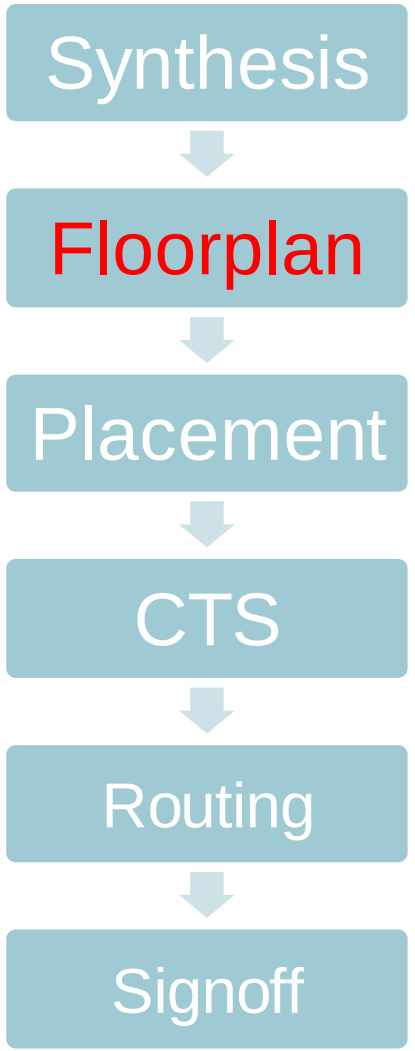
- In floorplan, we guide the tool...
  - IO Placement
  - Die Size & Aspect Ratio
  - Special Cell Pre-placement
  - IP/Macro Pre-placement
  - Power grid generation
  - Blockage deginition
- Concept
  - Core Utilization



(expected used slots)/(all available slots)

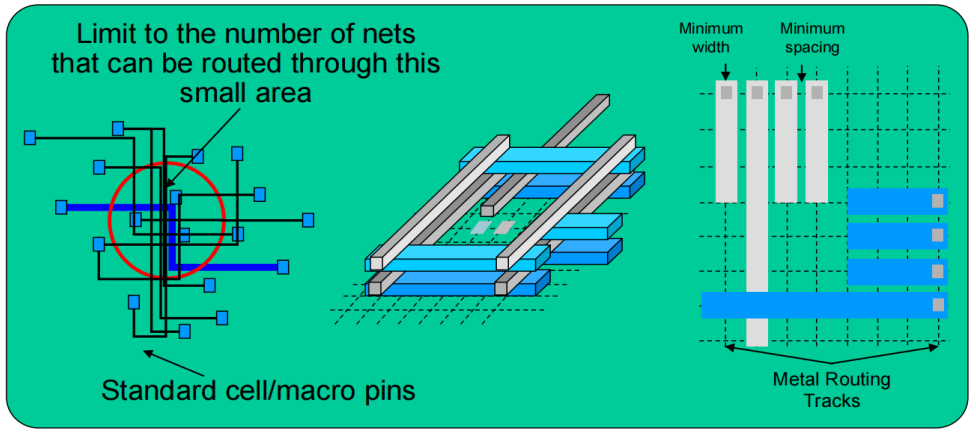
To high -> hard to route & too much heat

To low -> chip will be unnecessarily large



Placement

- The tool will try to place all the cells into the legal position.
  - lots of algorithms involved
  - global placement + detailed placement
- Congestion issue!
  - our routing resource is limited
  - so care about the place density

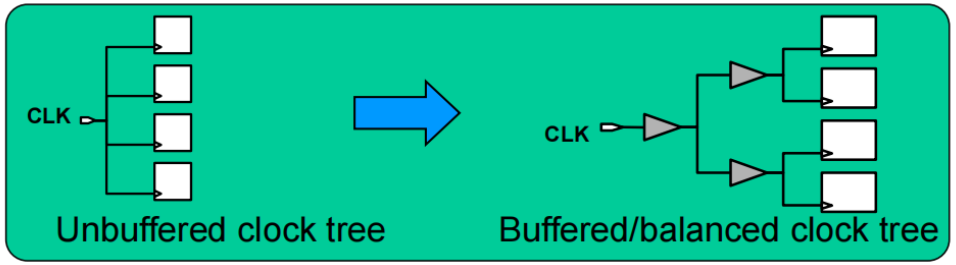


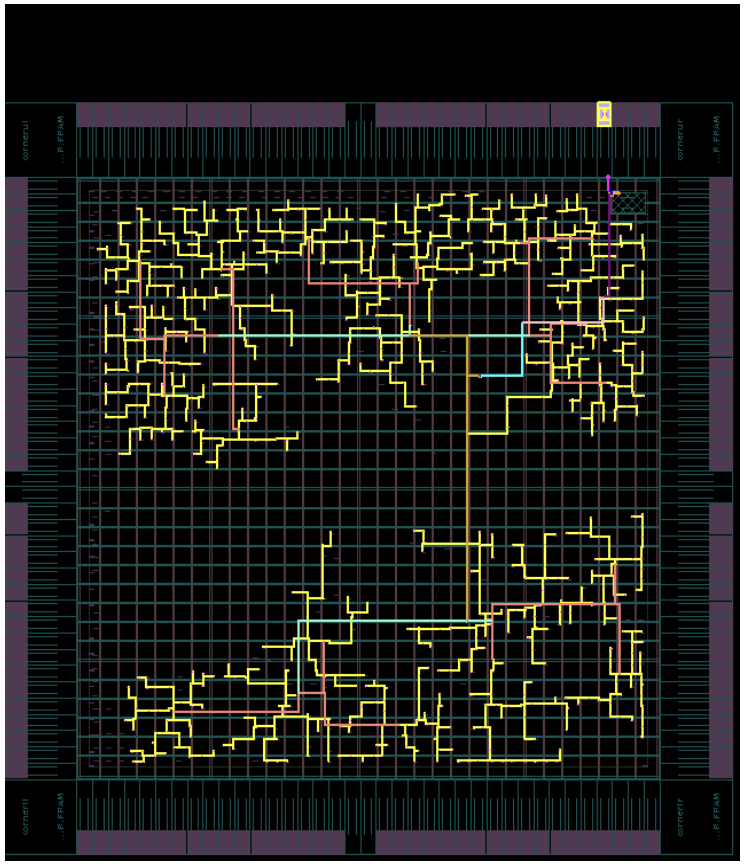
When this limit is approached or exceeded, this area is said to be congested.

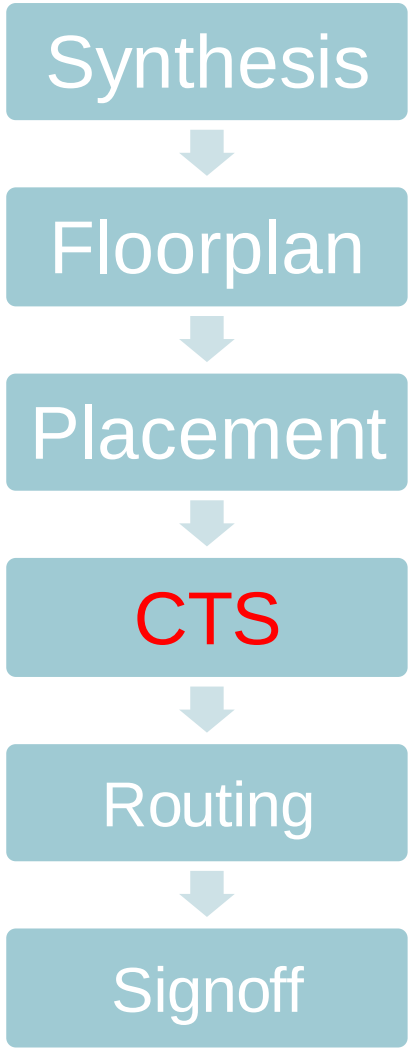
Synthesis
Floorplan
Placement
CTS
Routing
Signoff

Clock Tree Synthesis (CTS)

- Clock signal is important for timing, so route the clock signal first.

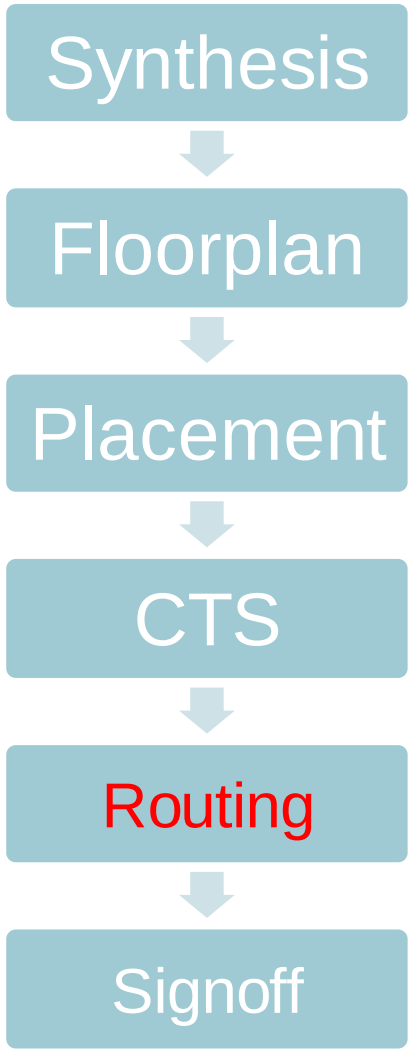






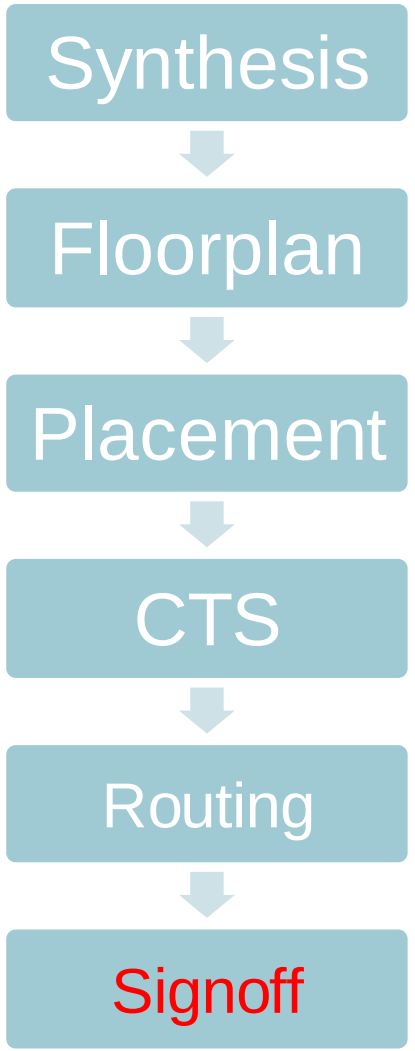
Routing

- Route other signals and try to meet the timing.
  - also involves lots of algorithms
  - global routing + detailed routing
- After this step, a complete GDS layout will be generated.



Signoff

- Do lots of checks & ECO
- DRC, LVS, STA...
  - DRC: Design Rule Check
    - check if the design satisfies the design rule
  - LVS: Layout vs Schematic
    - check if the layout is the same as the original design
  - STA: Standard Timing Analysis
    - check if the design have timing violations
  - Important to know how to read the reports
- ECO (Engineering Change Order)
  - Manual work, fix things in the last minute...



Debugging Strategy

- INFO
  - Report data, status, or current progress
- WARNING
  - Unexpected situation, but tools will do best to continue
  - Designer should fix warnings or validate they are benign
- ERROR
  - Unexpected situation, tools cannot work around issue
- CRIT
  - openroad must exit immediately (rare)
  - All segfaults / asserts / crashes are bugs :)

Debugging Strategy

- Review error which caused flow to abort
- Check warnings and errors starting from beginning of flow
  - Early warnings can be cause of later errors
- Try to identify root cause of issue
  - Design problem?
  - Tool problem?
  - Unrealistic expectations?

Common Problems and Solutions

- Utilization too high - fails placement
  - Increase die area or decrease core utilization
- Utilization too high - fails resizing
  - Check for proper SDC constraints
  - Check that user-generated macros have reasonable constraints (e.g. good .lib files)
- Congestion too high - fails global routing
  - Try previous fixes
  - Try decreasing layer adjustment
- Congestion too high - fails detail routing
  - Try previous fixes
  - Try adding cell padding to space cells further apart
  - If violations always occur on same cell(s), try marking those cells as dont_use
- Design too small - fails PDN generation
  - Try increasing design size or reducing power grid pitch

Common Problems and Solutions

- Design runtime too long
  - Increase utilization if too low
  - Relax timing constraints
  - Reduce design complexity
  - Faster machine :)
- Failing setup time
  - Hard problem - may just need to reduce constraints
  - Change architecture: more pipelining, reduce complexity
- Failing hold time
  - Check that user cells (e.g. SRAM) are properly constrained
  - Check design constraints are valid (SDC)
    - Designs with multiple clocks are tricky!
  - Check that your PDK has properly correlated parasitics

Lab #4

- Due at 11:59 PM, Nov. 18th, Friday
- The content is not much, but it is important to understand every step and report, and also how this flow works. Lab4 ~ Lab5 are all based on this flow, so recommend everyone can run through all the steps.
- The debug part may cost lots of time if you are not familiar with how design flow works.
- Contact me if you have problem when logging in the EDA platform.
- Also, submit through this platform, not through canvas. (All deliverables with your individual report).

Reference

- Yichen Cai, 2023FALL Lab 4.