Topic 7.3

ASIC Design Flow III

Xinfei Guo
xinfei.guo@sjtu.edu.cn

November 11th, 2021

T7 learning goals

- How to design a Chip (SoC) from concept to silicon?
  - **Full design flow from RTL to Layout**
  - **How to make decisions at each step**

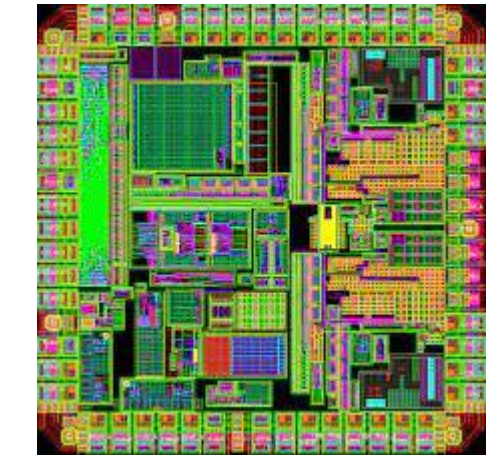
```
module PE (clock, R, S1, S2, S1S2mux, newDist, Accumulate, Rpipe);
input clock;
input [7:0] R, S1, S2;// memory inputs
input S1S2mux, newDist;// control inputs
output [7:0] Accumulate, Rpipe;
reg [7:0] Accumulate, AccumulateIn, Difference, Rpipe;
reg       Carry;

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



Layout

Figure: Synopsys

Placement

What should we place cells on the floorplan?

Physical Synthesis Flow

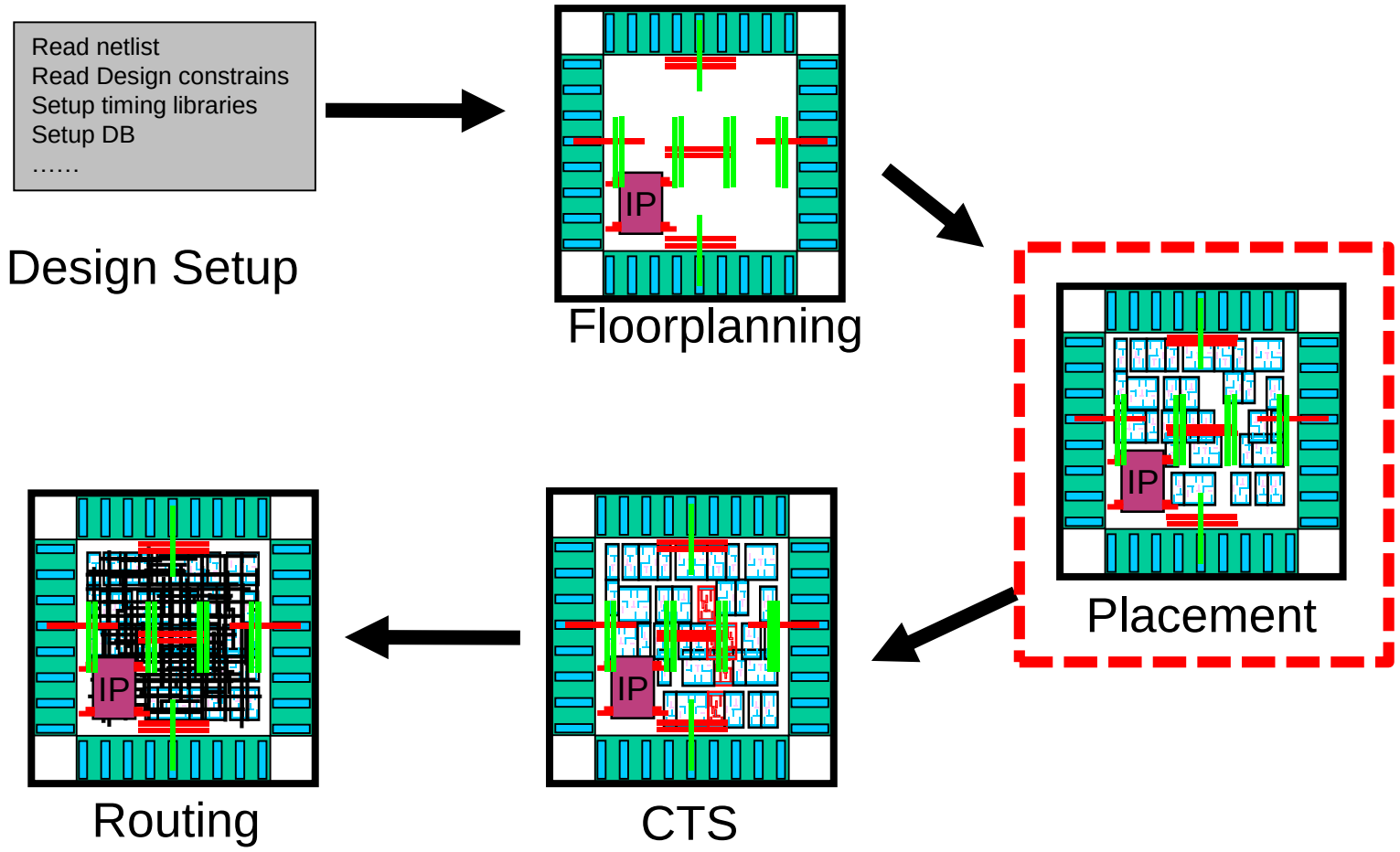
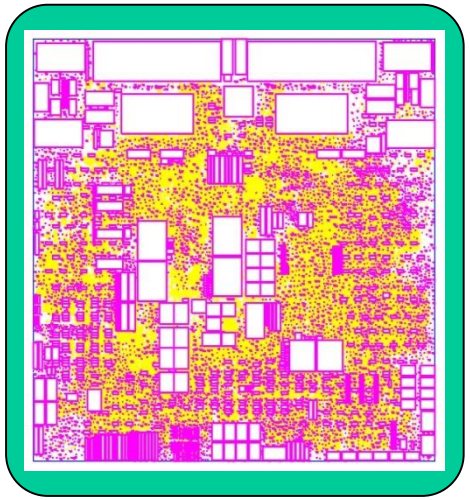


Figure: Synopsys

Placement Problem

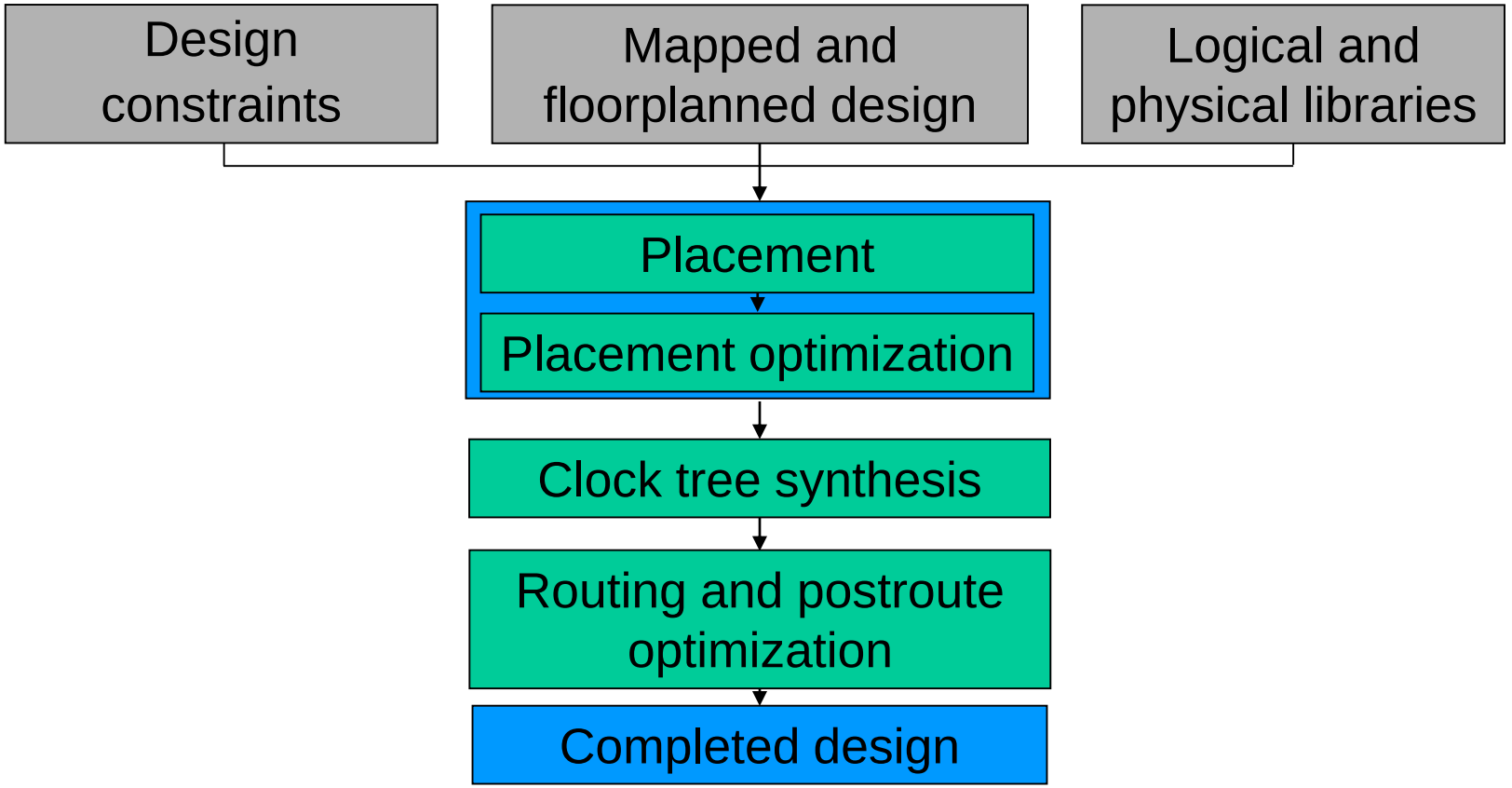
- The goal of placement is to **minimize the total area and interconnect cost**.



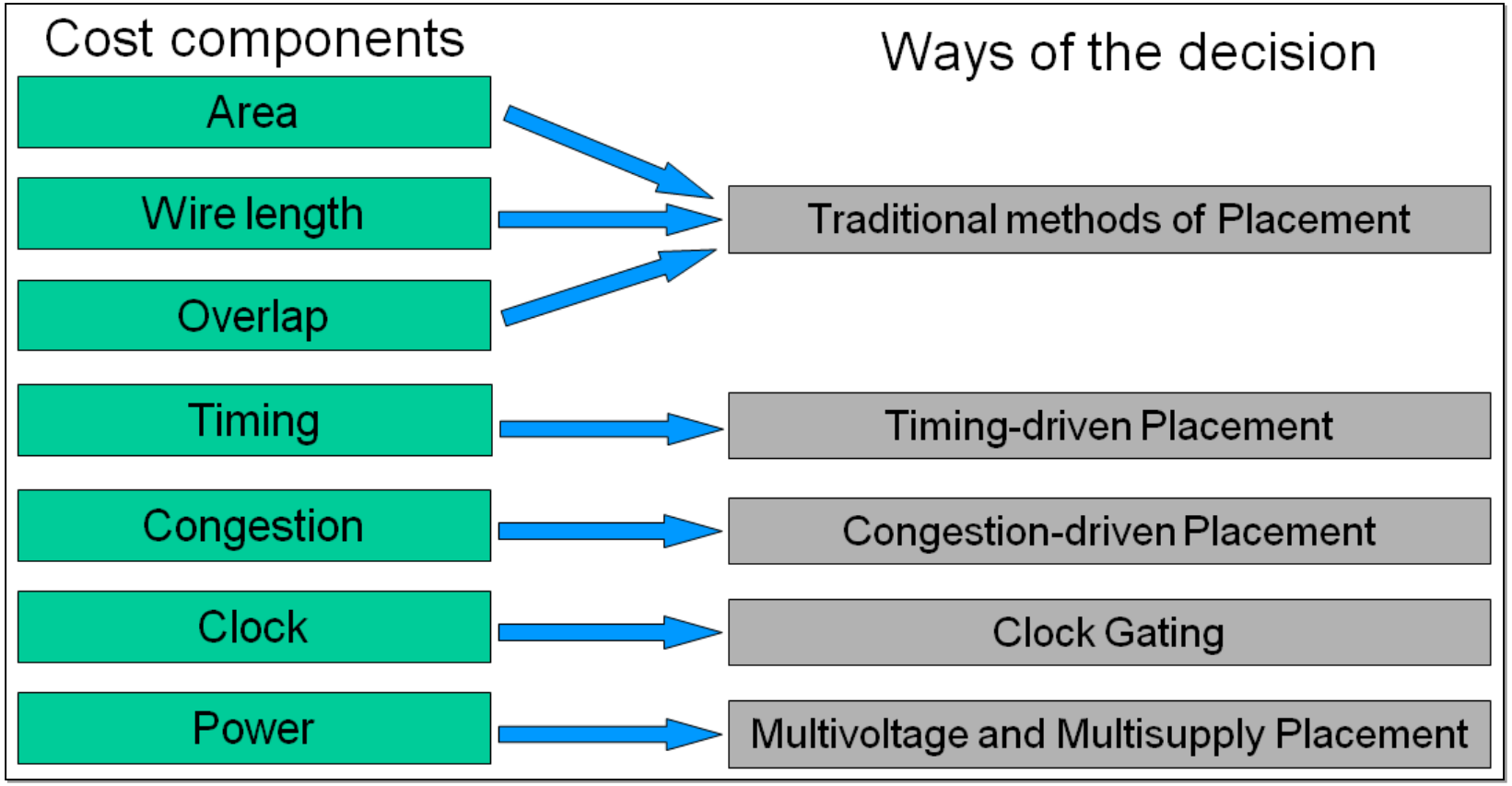
The quality of the attainable routing is highly determined by the placement.

Circuit placement becomes very critical in 90nm and below technologies.

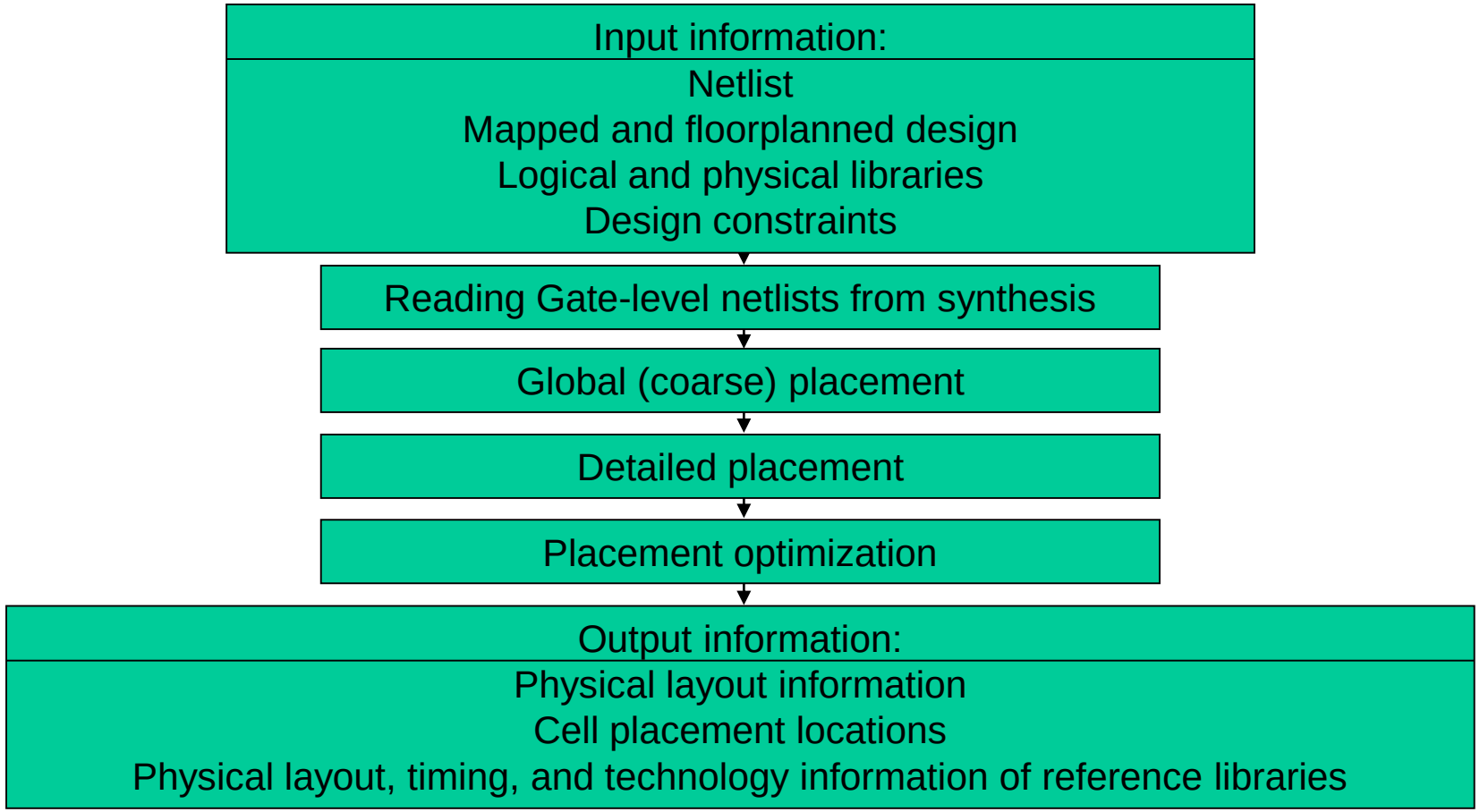
Location of Placement in a Typical IC Physical Design Flow



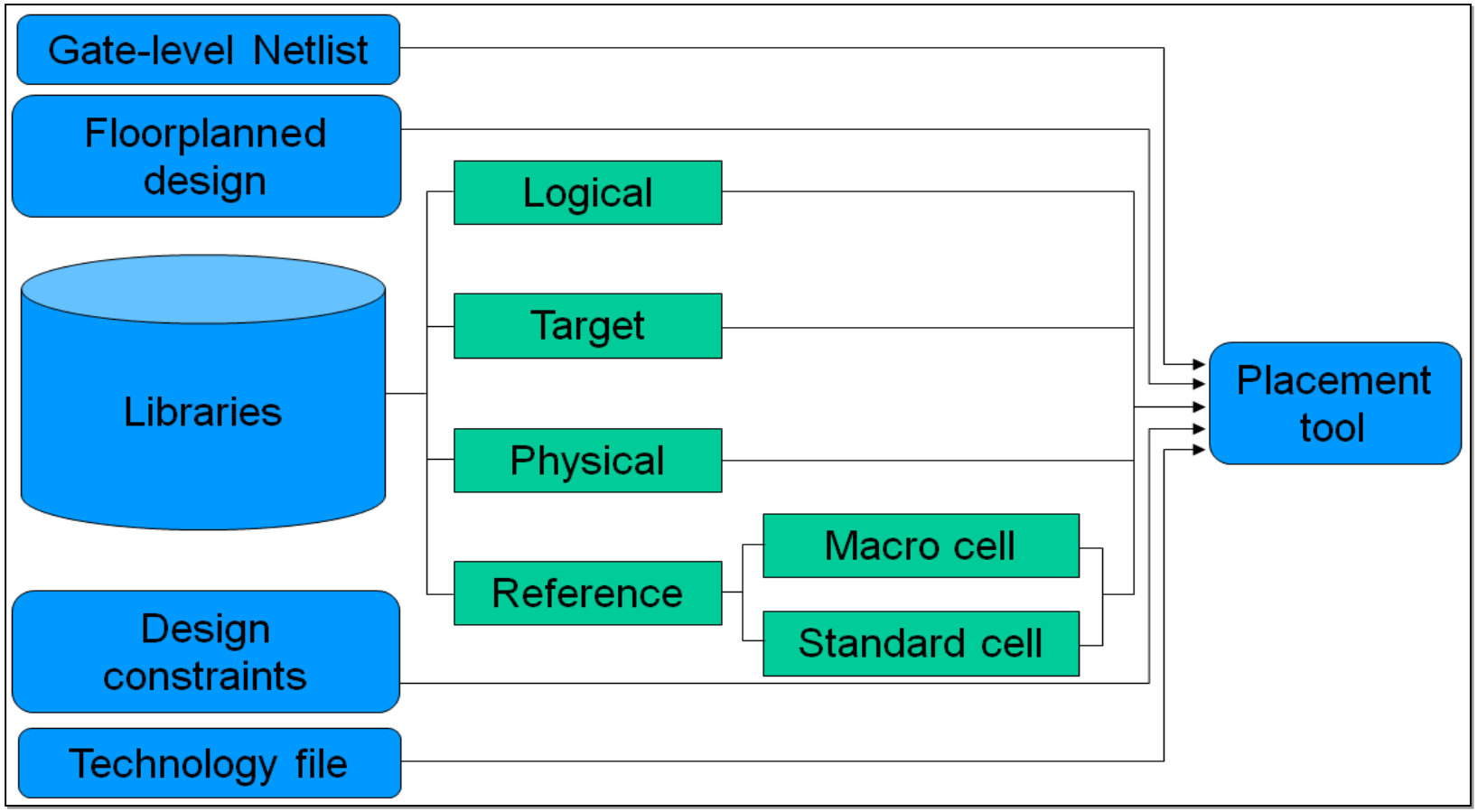
New Tendencies of Physical Designing and Placement Cost Components



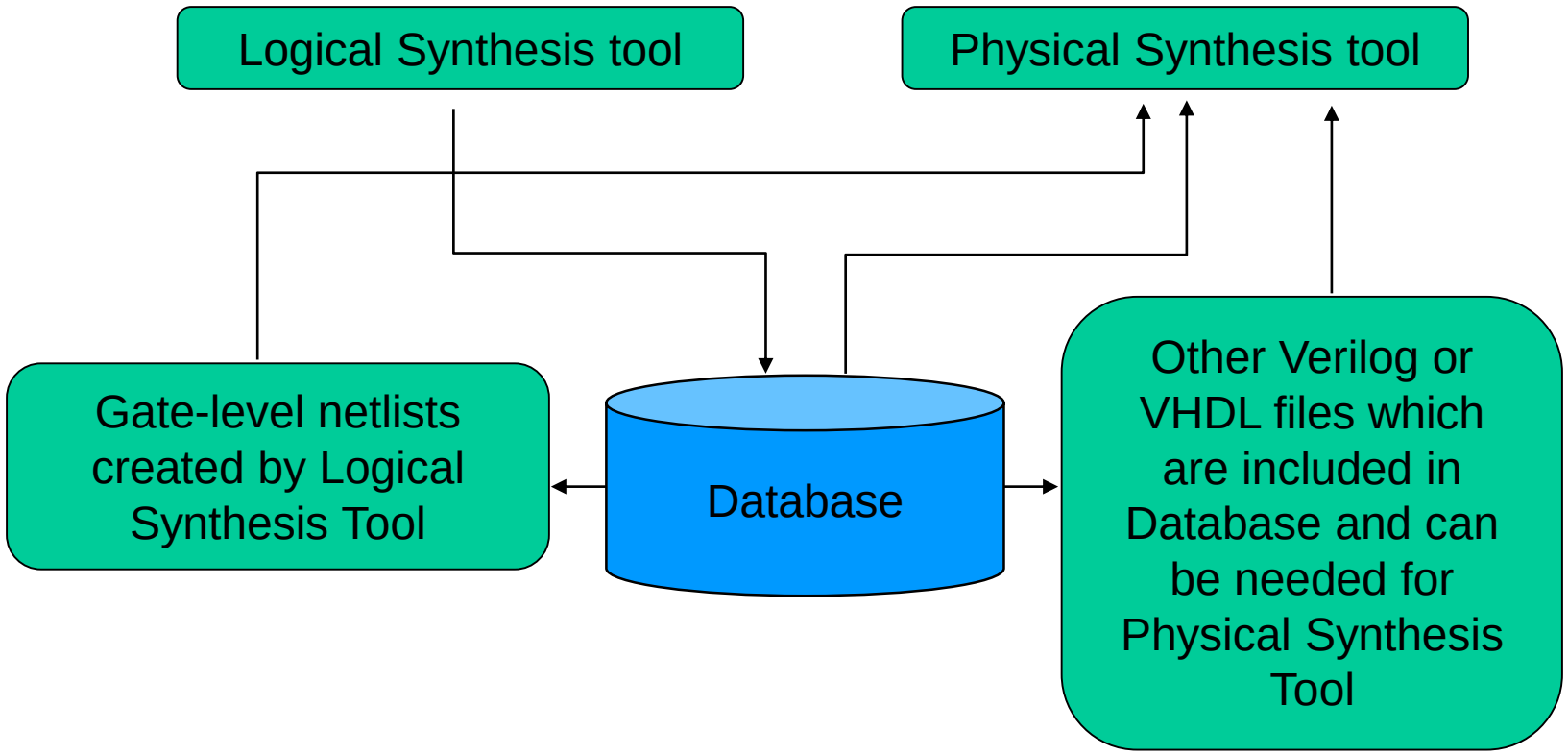
Placement Steps



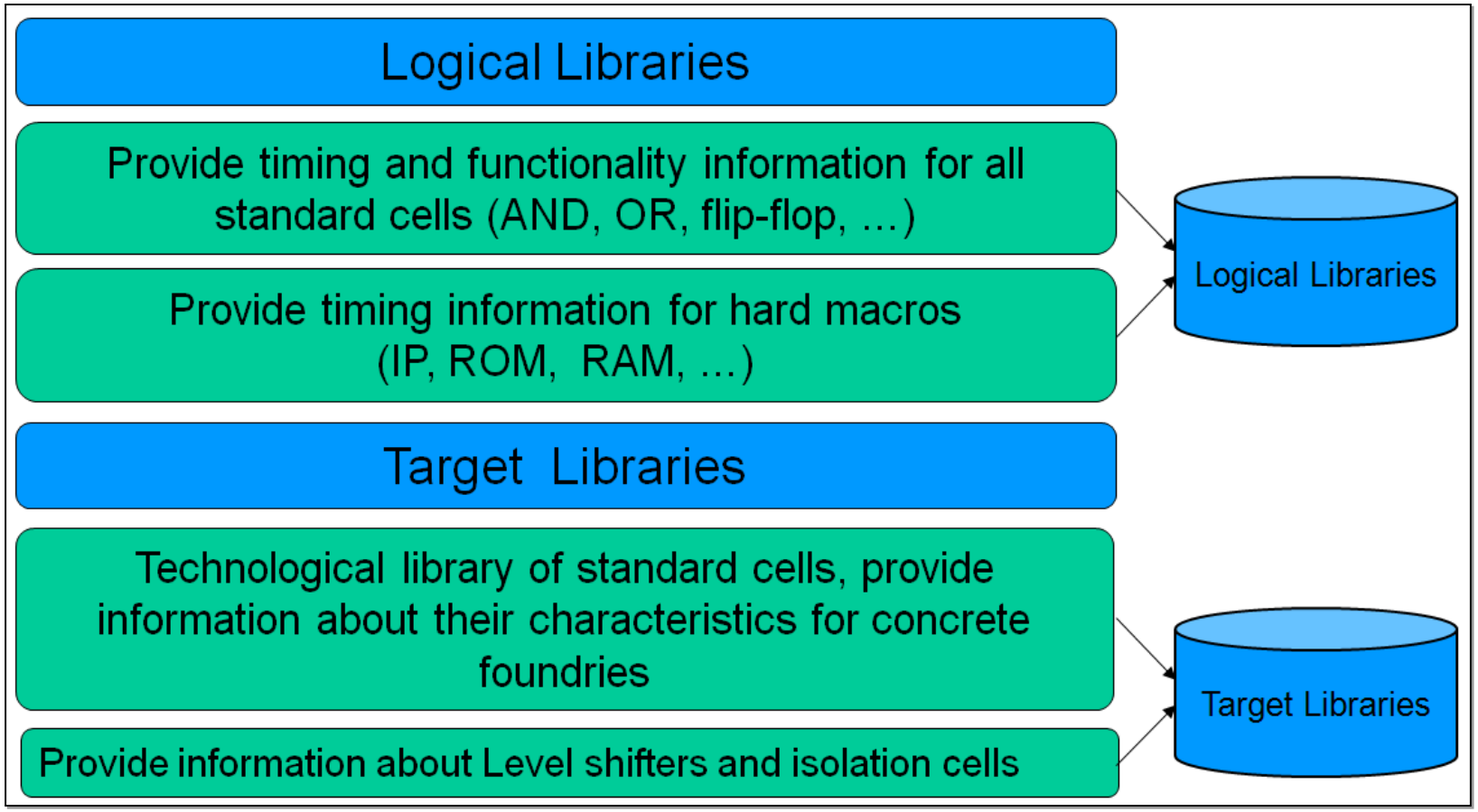
Input Information for Placement Tool



Reading Gate-Level Netlists from Synthesis

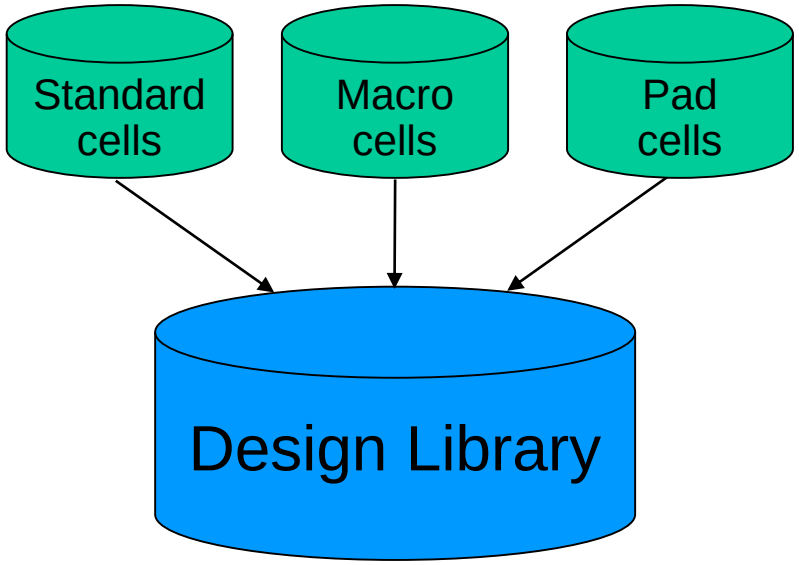


Logical and Target Libraries



Reference Libraries

- Contain subdesigns or cells used by many other designs
- Referenced by pointers in the design library for memory efficiency



Technology File

- Tech File is unique to each technology
- Contains metal layer technology parameters
  - Number and name designations for each layer/via
  - Dielectric constant for technology
  - Physical and electrical characteristics of each layer/via
  - Design rules for each layer/Via (Minimum wire widths and wire-to-wire spacing, etc.)
  - Units and precision for electrical units
  - Colors and patterns of layers for display
  - . . .

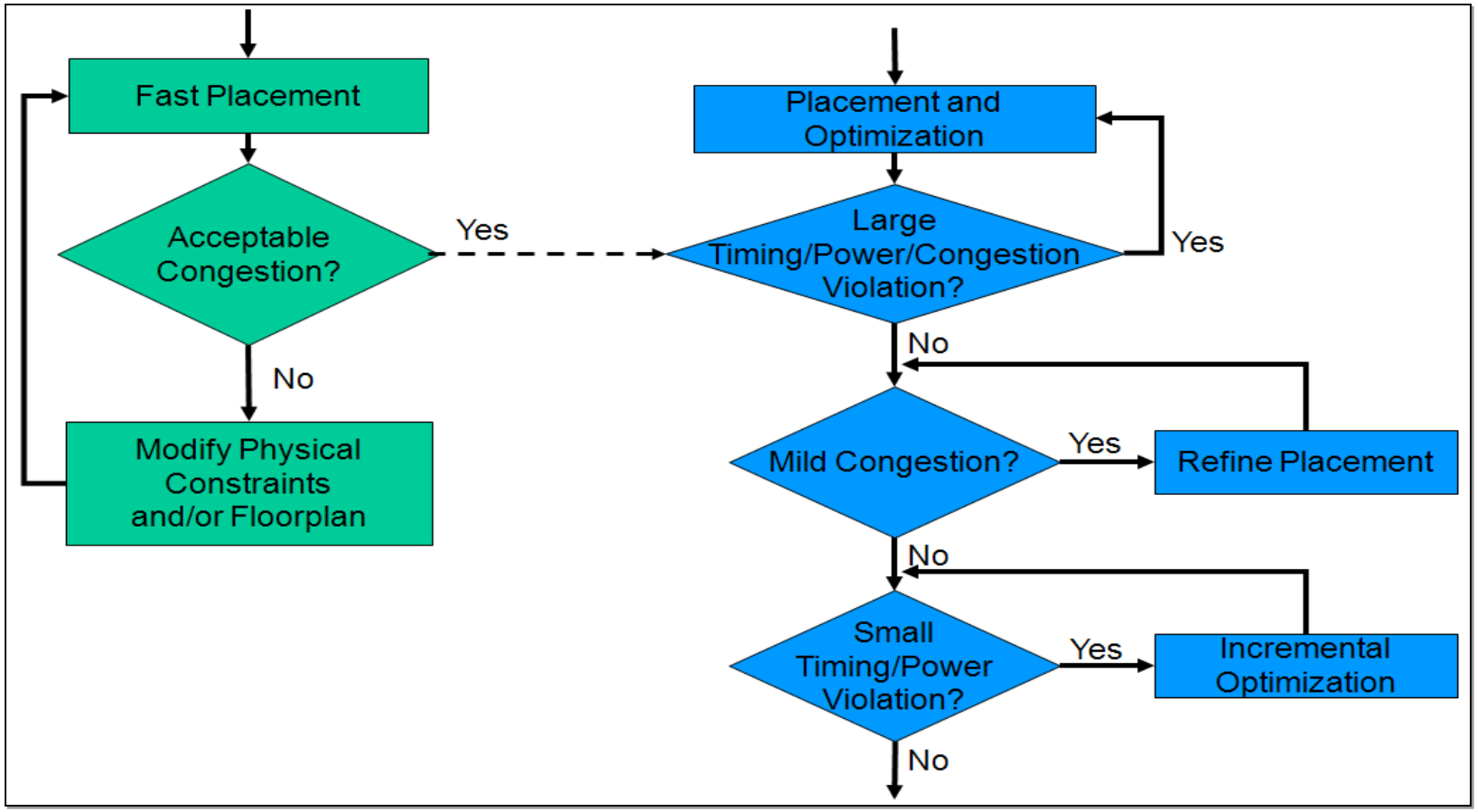
Example of a Technology File

```
Technology  {
dielectric      = 3.7
unitTimeName = "ns"
timePrecision = 1000
unitLengthName = "micron"
                    }
...
Layer  "m1" {
layerNumber = 16
maskName    = "metal1"
pitch = 0.56
defaultWidth = 0.23
minWidth = 0.23
```

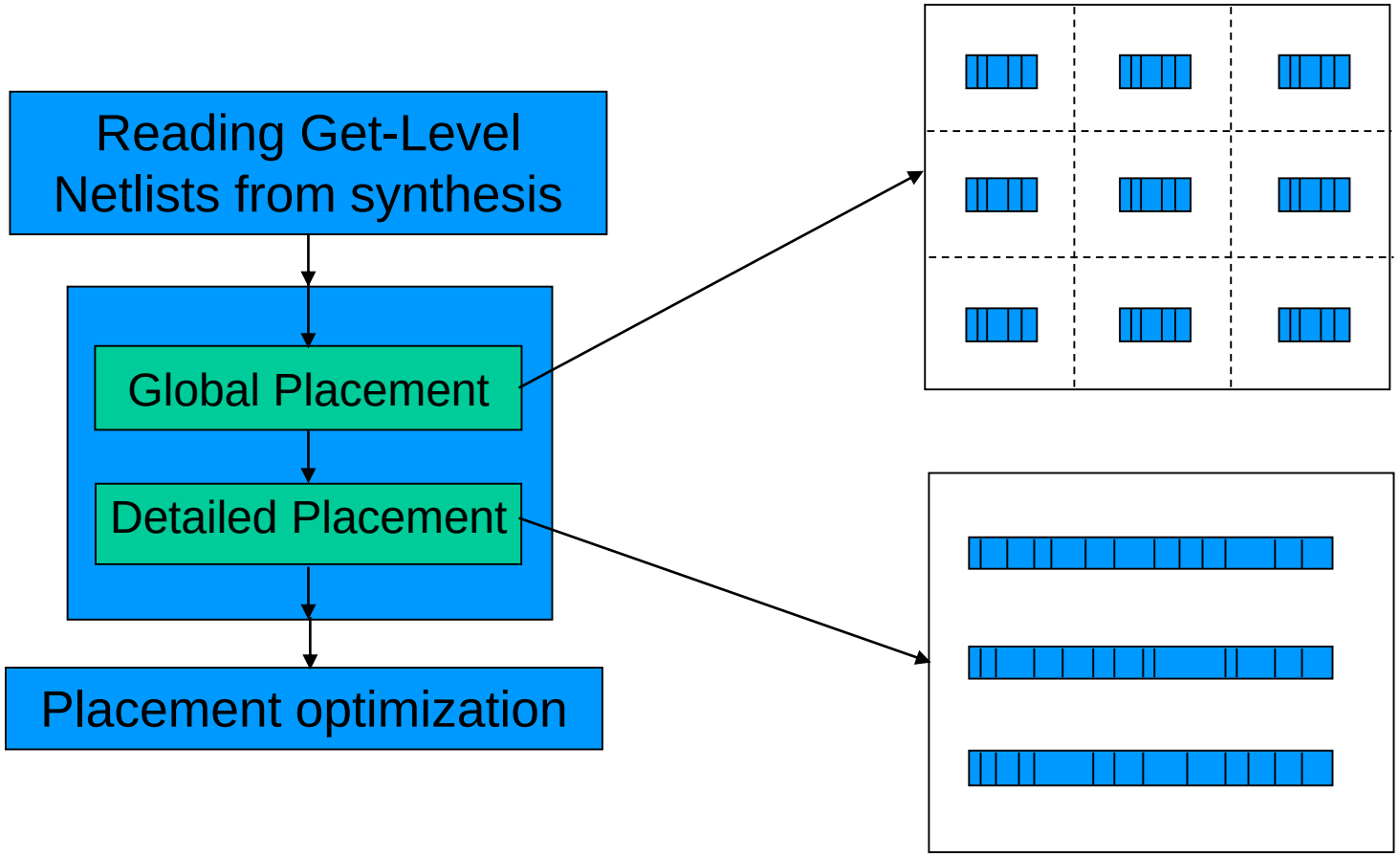
Placement and Optimization Attributes

Typical Attributes	Coarse placement	Detailed placement	Optimization
Fixed	Cannot move cells	Cannot move cell	Cannot move, rotate, or resize cells
Imposed on clock buffers	Cannot move cells	Cannot move cells	Cannot move, rotate, or resize cells
Soft fixed	Cannot move cells	No restrictions	No restrictions
Size only	No restrictions	No restrictions	Can only resize cells
In place size only	Cannot move cells	No restrictions	Can resize cells only if there is room
Imposed on clock sinks	No restrictions	No restrictions	Can resize cells only if there is room
Don't touch	No restrictions	No restrictions	Cannot move, rotate, or resize cells

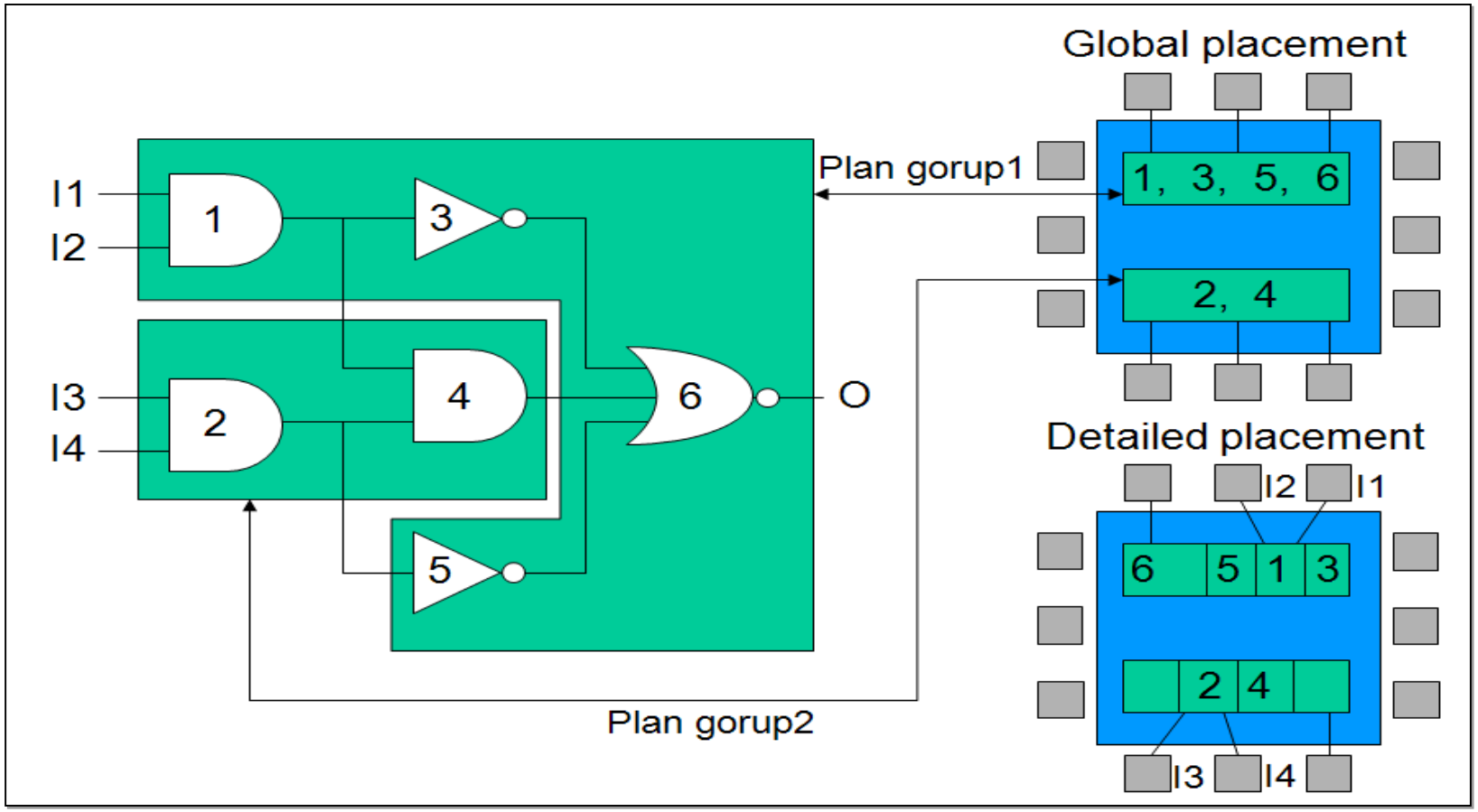
Placement Methodology



Global and Detailed Placement



Partitioning-Based Placement

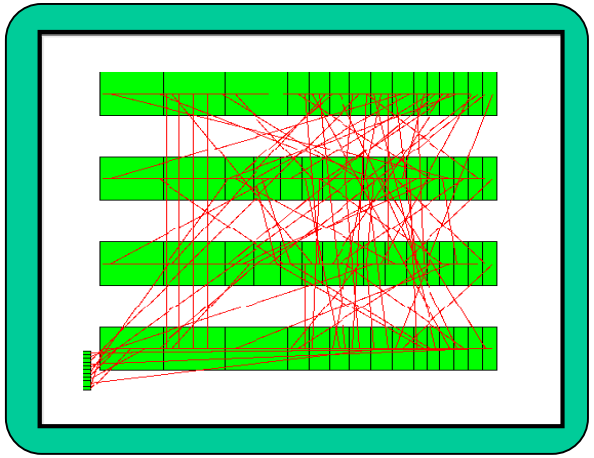


Global Placement

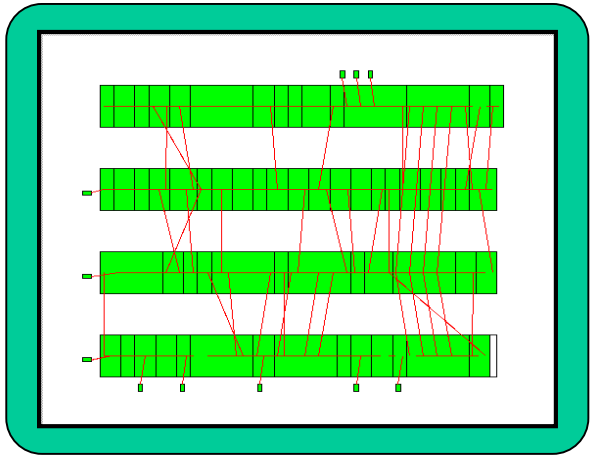
Standard cells must be in groups in such a way that the number of connections between groups is minimum

This issue is solved through circuit partitioning

As a basic criterion, the minimum is taken among group connections

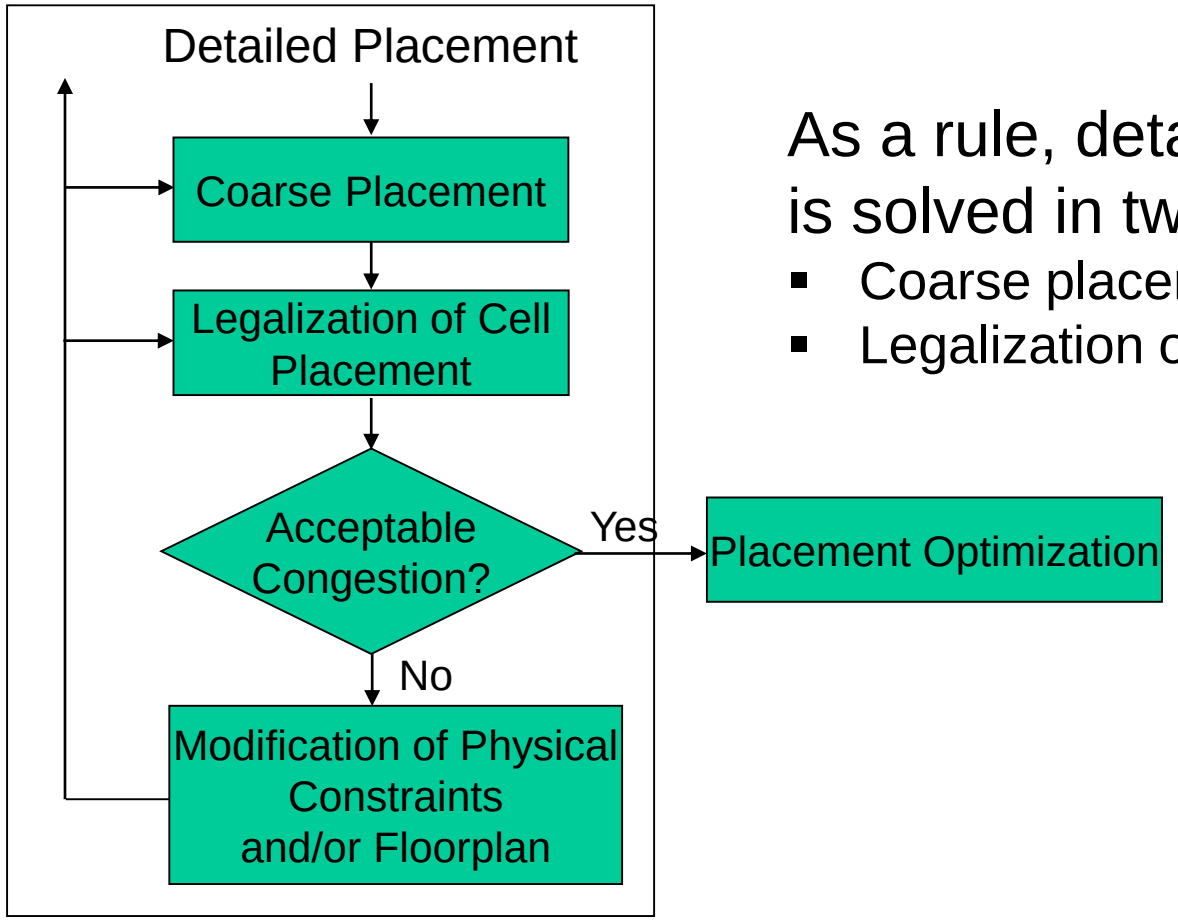


Bad Placement



Good Placement

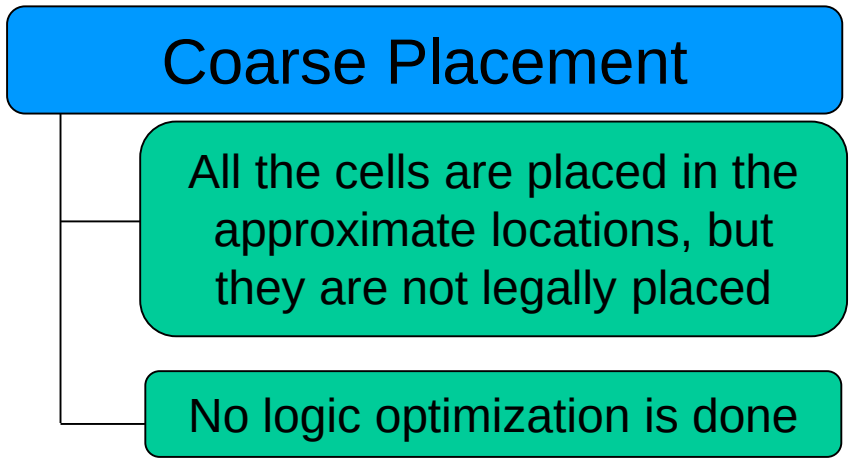
Detailed Placement

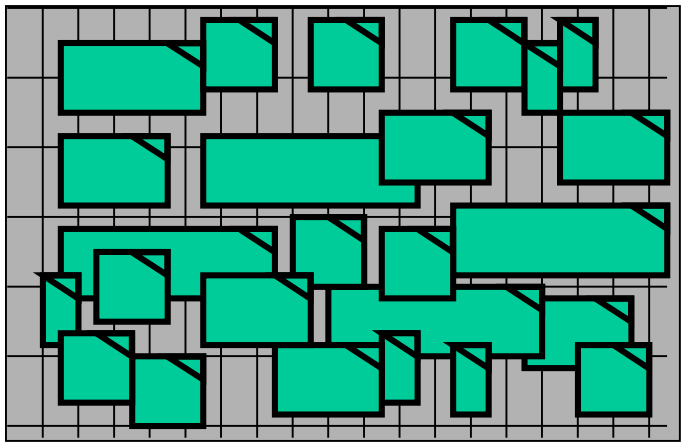


As a rule, detailed placement is solved in two stages:

- Coarse placement
- Legalization of cell placement

Coarse Placement

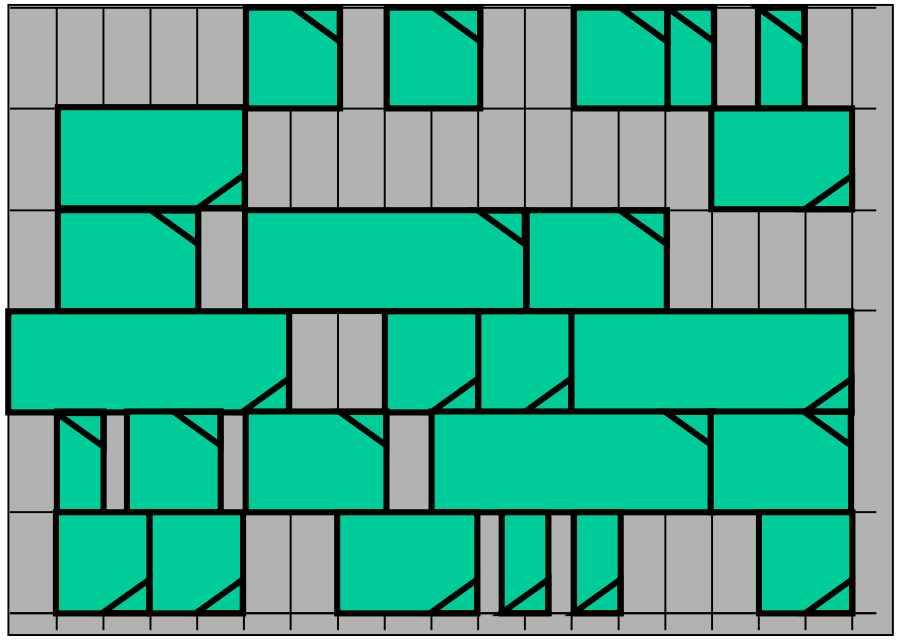




- In a coarse placement all the cells are placed in the approximate locations but they are not legally placed.
- Cells overlap and are not on-grid.
- Large cells (e.g. RAMs) form large placement blockages for other smaller leaf cells.
- Power routing forms routing layer blockages that will also be checked and avoided if specified.

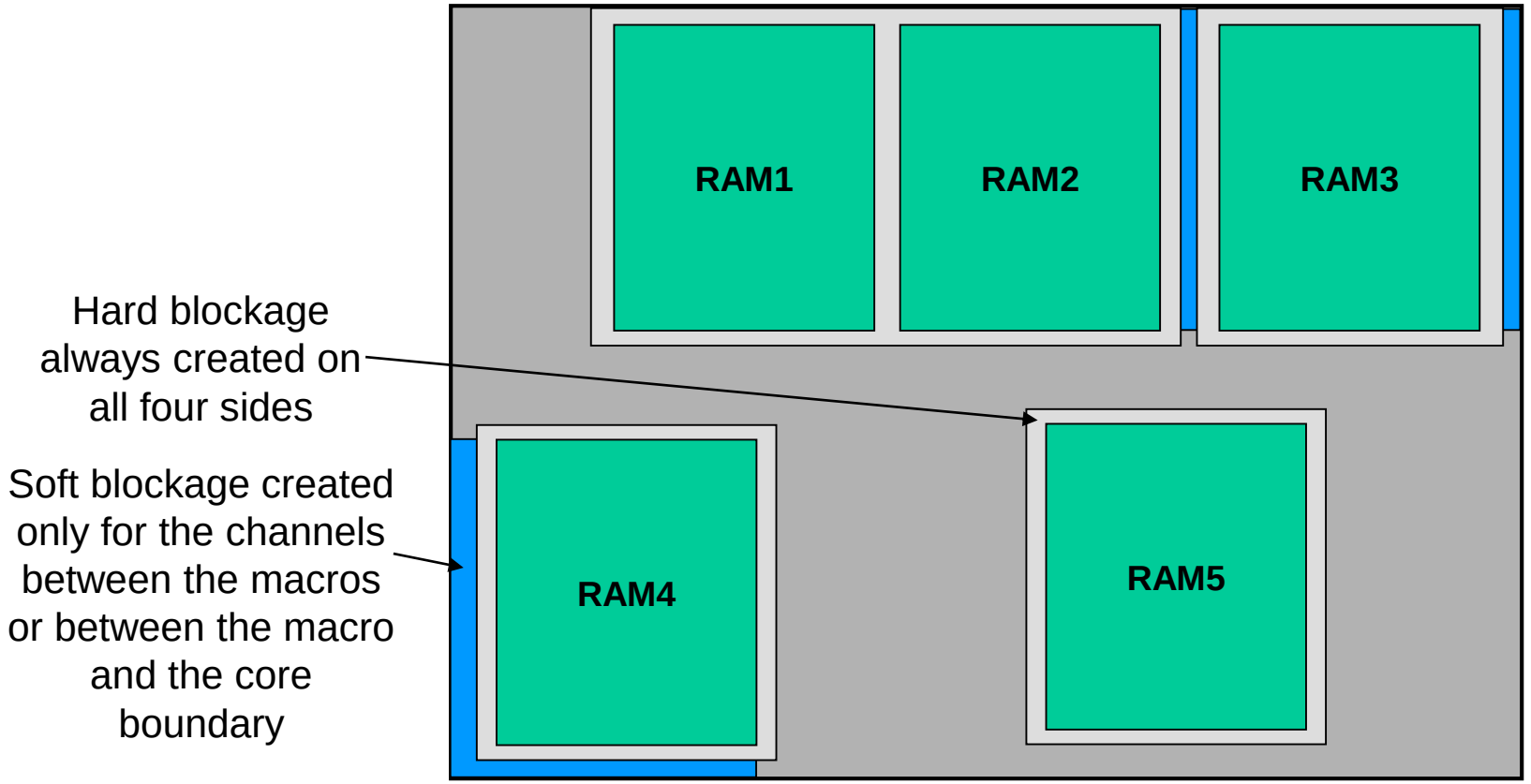
Legalize Cell Placement

Ensure that legal placement is done before saving the design.



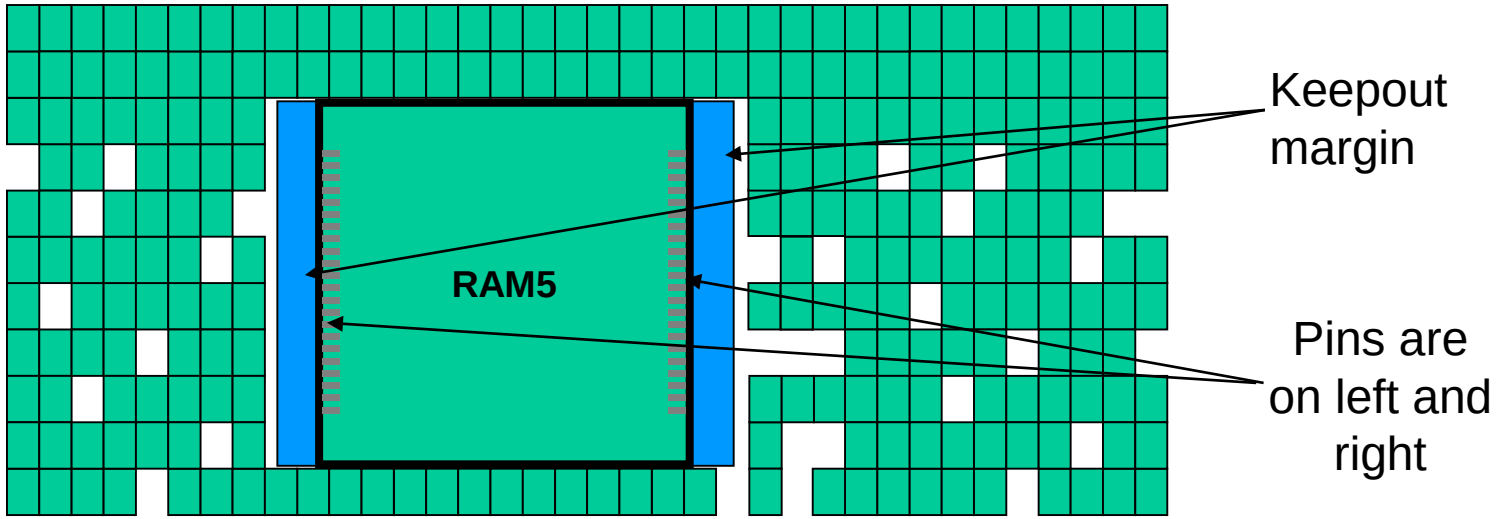
Legal placement of cells is not required for analyzing routing congestion at an early stage

Placement Blockages: Adding or Modifying Global Placement Blockages



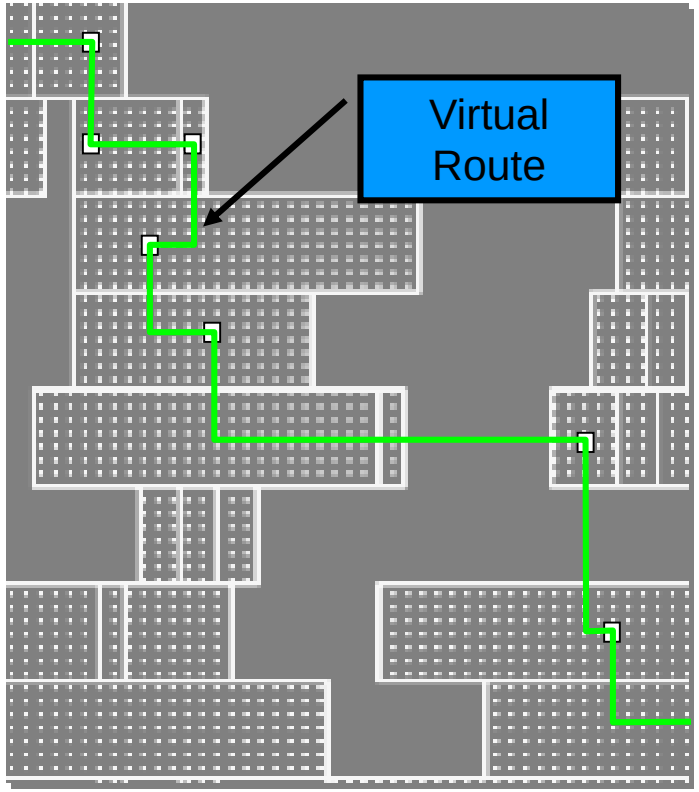
Placement Blockages: Macro Keepout Margin (Padding)

A keepout margin is a region around the boundary of fixed macros in the design in which no other cells are placed.



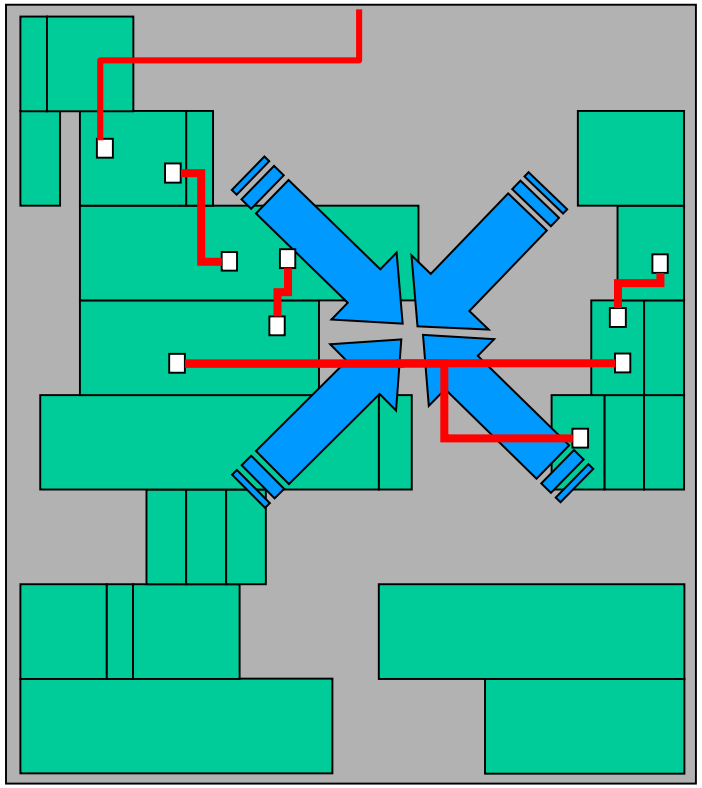
Timing-Driven Placement (1)

- All steps including placement are timing-driven
- Timing-driven placement tries to place critical path cells close together to reduce net RCs and to meet setup timing
- RCs are based on Virtual Route (VR)



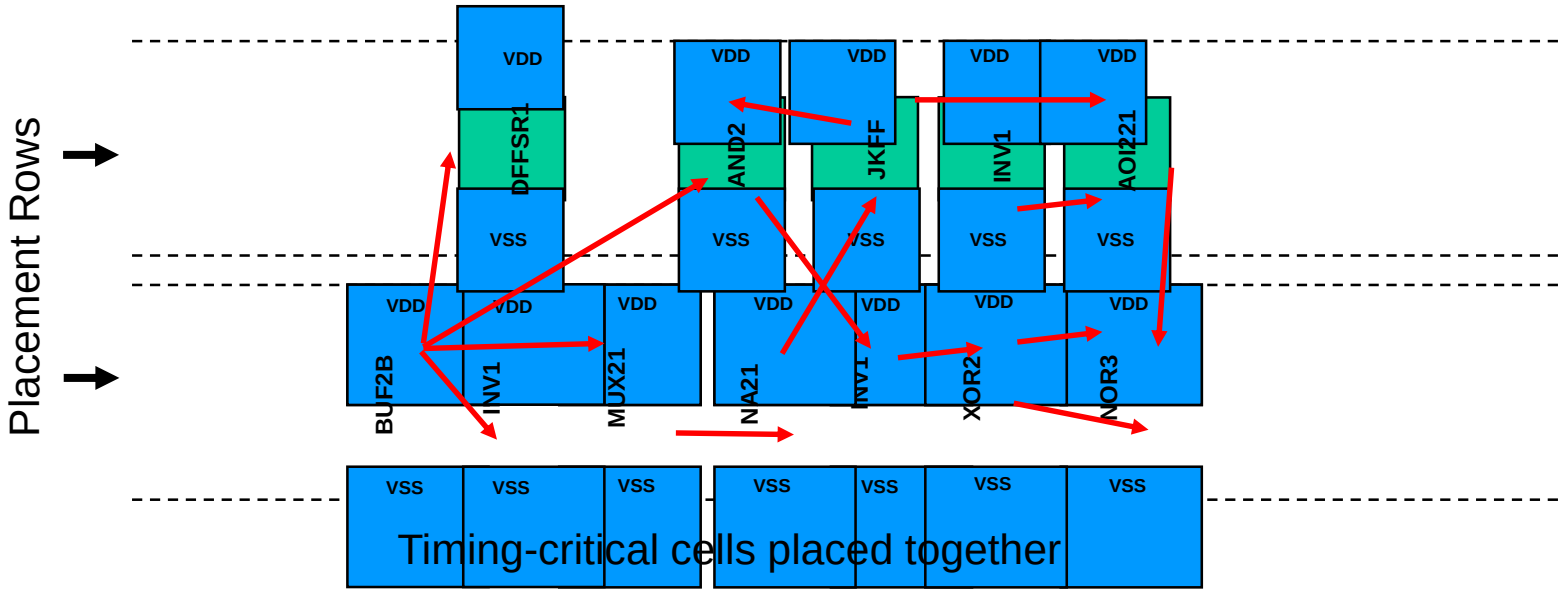
Timing-Driven Placement (2)

- Timing-driven placement based on Virtual Route
  - Tries to place cells along timing-critical paths close together to reduce net RCs and meet setup timing
  - Net RCs are based on Virtual Routing (VR) estimates

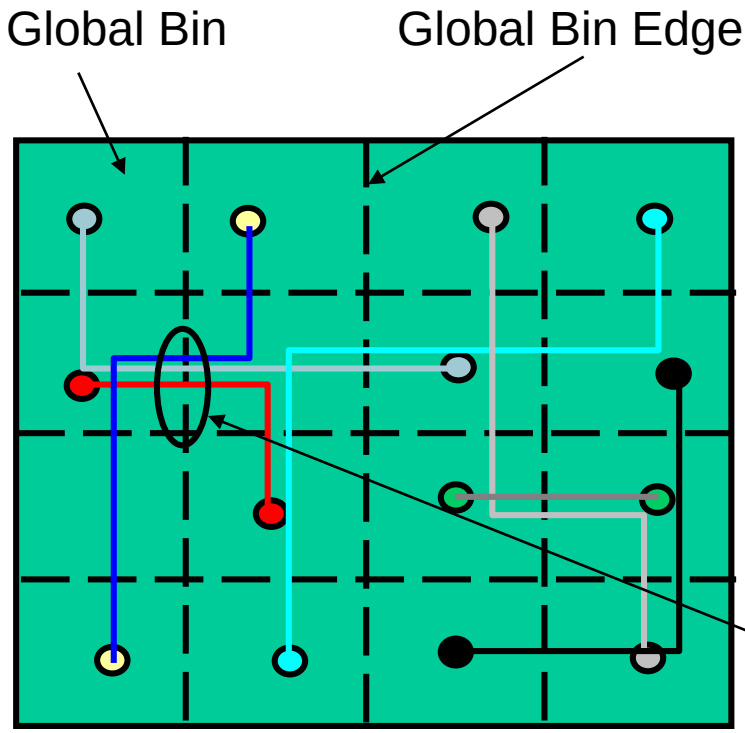


Timing-Driven Placement (3)

- Standard cells are placed in “placement rows”
- Cells in a timing-critical path are placed close together to reduce routing-related delays → Timing-Driven Placement



Congestion-driven Placement: Congestion



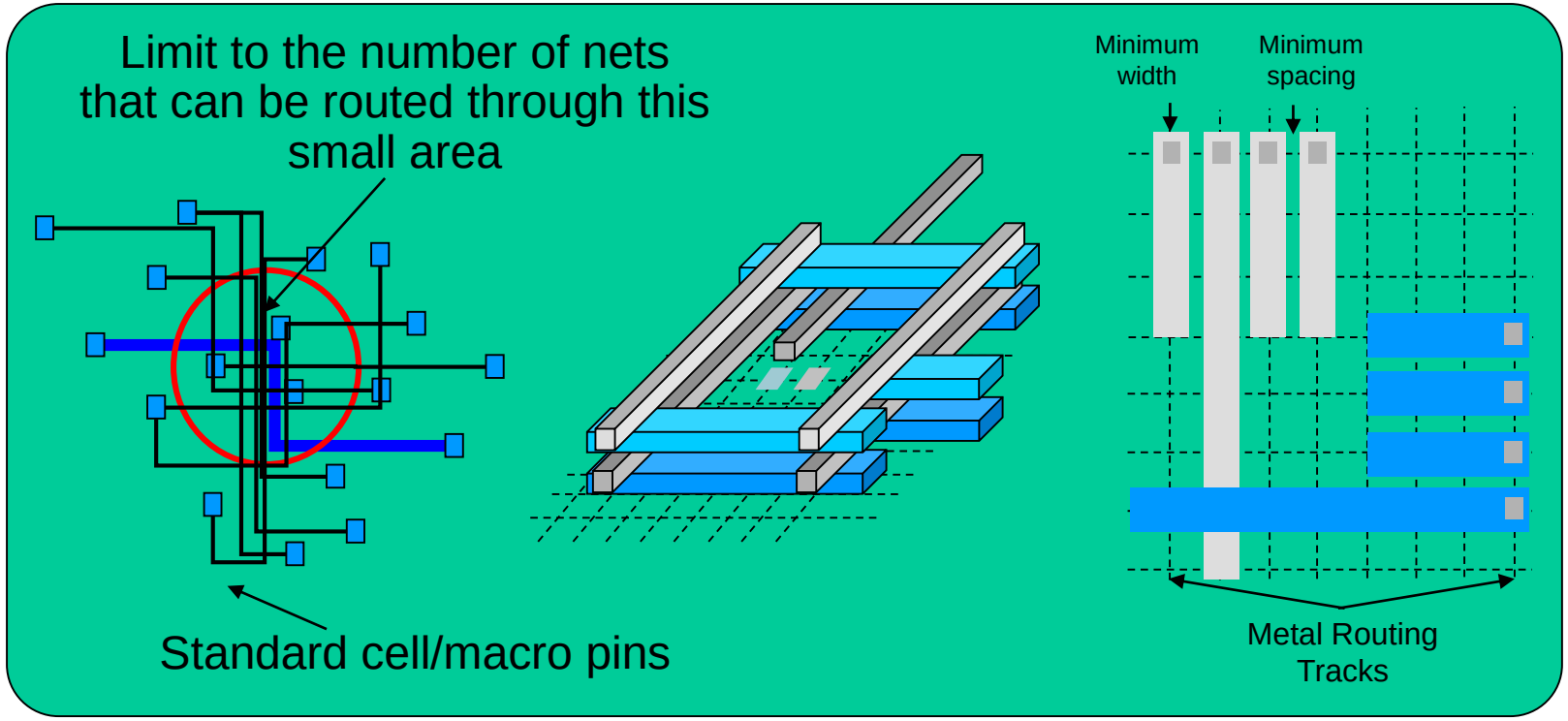
Overflow on each edge =

$$\begin{cases} \text{Routing Demand - Routing Supply} \\ 0 \quad \text{(otherwise)} \end{cases}$$

$$\text{Total Overflow} = \sum_{\text{all edges}} \text{overflow}$$

Routing demand = 3
Assume routing supply is 1,
overflow = 3 - 1 = 2 .

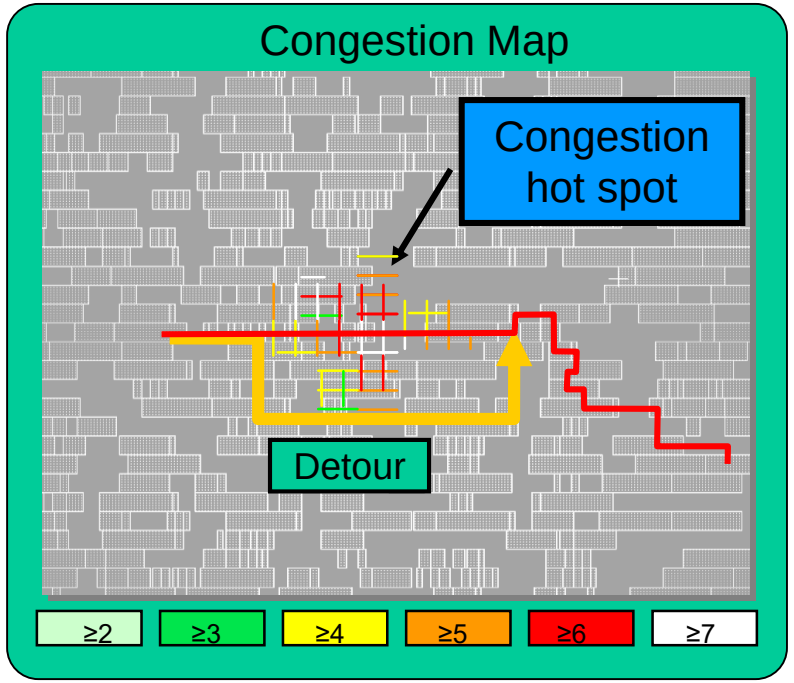
Congestion-driven Placement: Routing resource



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

Placement Issues with Congestion

- If congestion is not too severe, the actual route can be detoured around the congested area
- The detoured nets will have worse RC delay compared to the VR estimates



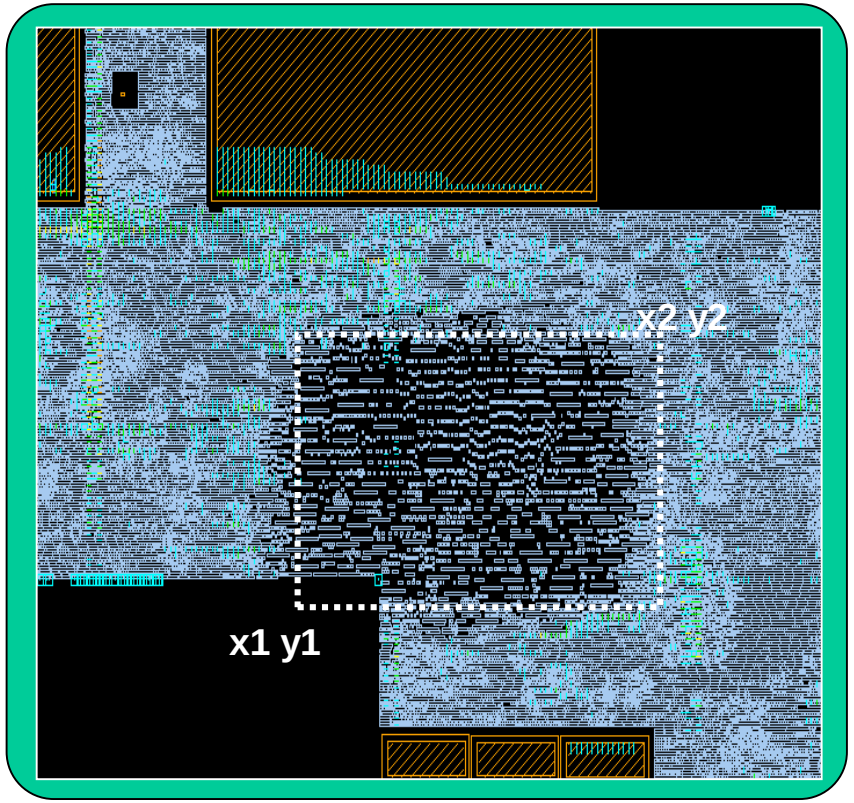
In highly congested areas, delay estimates during placement will be optimistic.

Fix Congestion: Modify Floorplan

- Top-level ports
  - Changing to a different metal layer
  - Spreading them out, re-ordering or moving to other sides
- Macro location or orientation
  - Alignment of bus signal pins
  - Increase of spacing between macros
- Core aspect ratio and size
  - Making block taller to add more horizontal routing resource
  - Increase of the block size to reduce overall congestion
- Power grid: Fixing any routed or non-preferred layers

Modifying Physical Constraints: Cell Density

- Cell density can be up to 95% by default
  - Density level can also be applied to a specific region
- Lower cell density in congested areas using coordinate option



Clock Tree Synthesis (CTS)

How to route the clock?

Physical Synthesis Flow

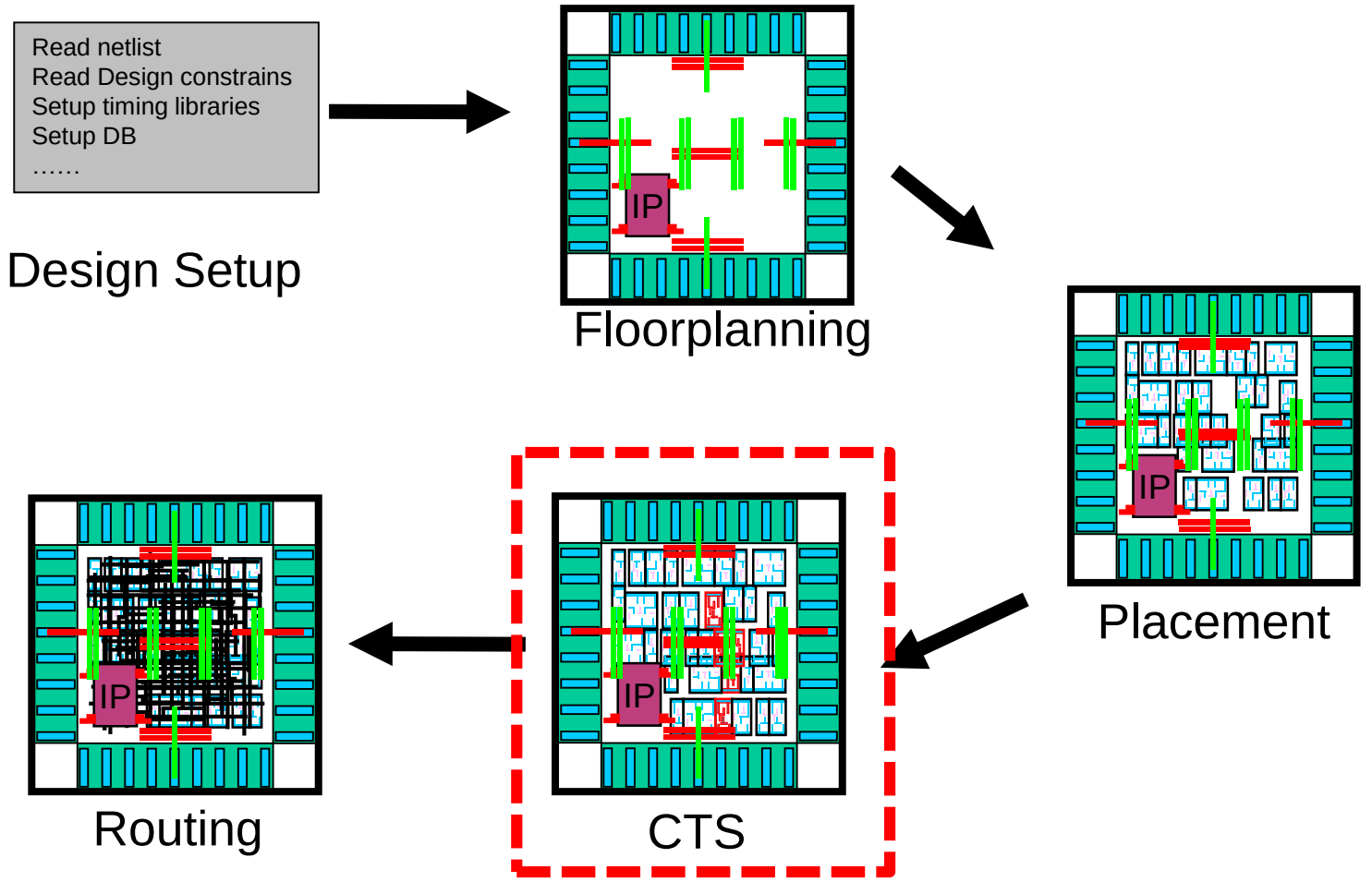
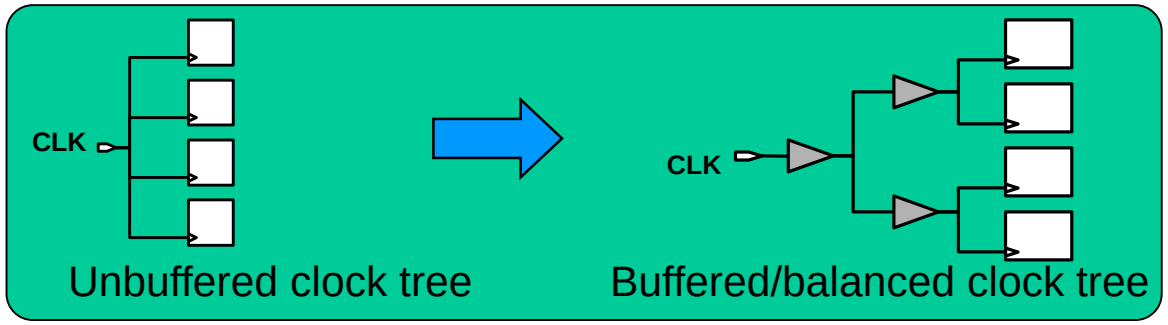


Figure: Synopsys

Clock Tree Synthesis (CTS) Problem

- CTS Problem
  - **CTS is the process of distributing clock signals to clock pins based on physical/layout information**
  - After placement of cells the tree of synchronization is synthesized
  - Balanced clock tree is synchronized with the addition of buffers
  - After routing CT optimization is made

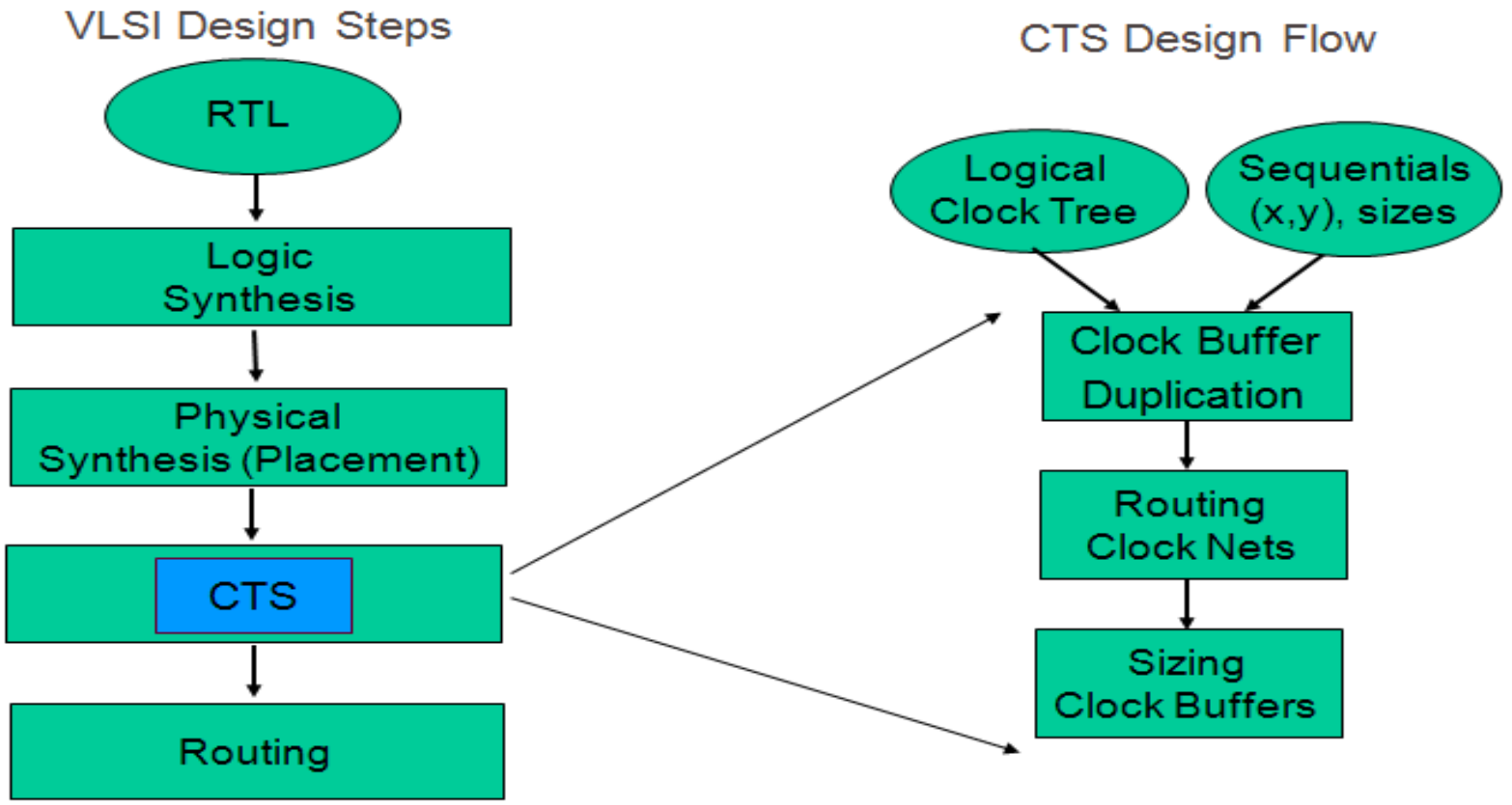


CTS Problem: Clock Tree Synthesis Goals

- Meeting the clock tree design rule constraints
  - Maximum transition delay
  - Maximum load capacitance
  - Maximum fanout
  - Maximum buffer levels
- Meeting the clock tree targets
  - Maximum skew
  - Min/Max insertion delay

Constraints are upper bound goals. If constraints are not met, violations will be reported.

CTS Problem

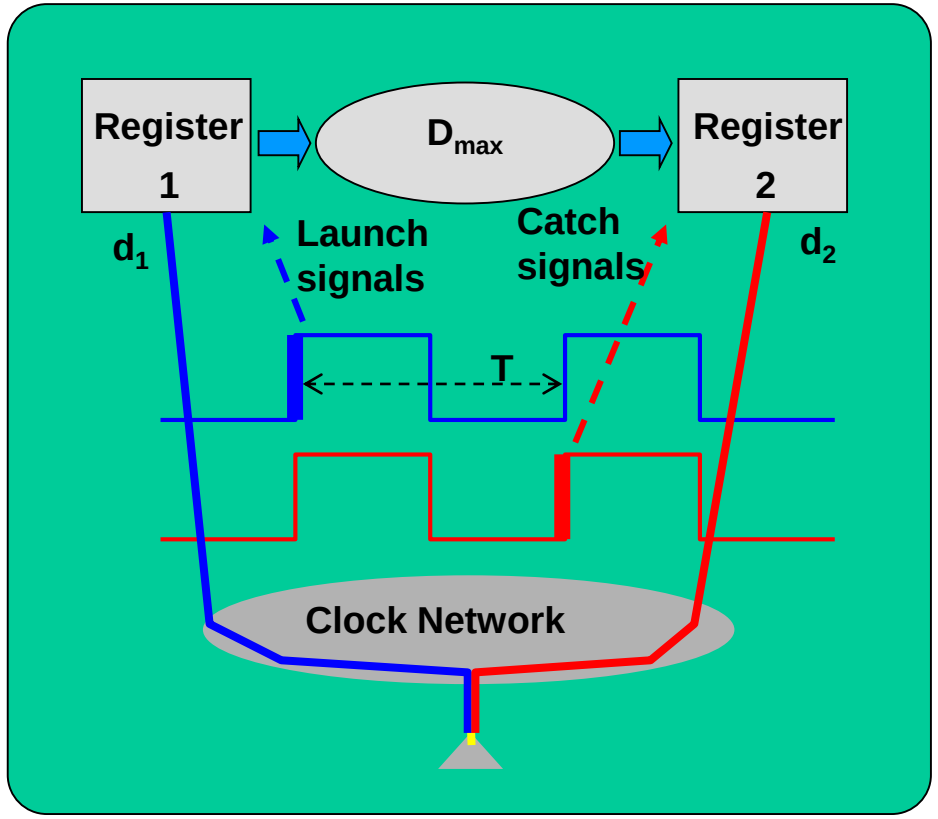

VLSI Design Steps
RTL
Logic Synthesis
Physical Synthesis (Placement)
CTS
Routing
CTS Design Flow
Logical Clock Tree
Sequentials (x,y), sizes
Clock Buffer Duplication
Routing Clock Nets
Sizing Clock Buffers

Clock Tree: General Concepts: Clock Distribution Network

Skew = $d_1 - d_2$

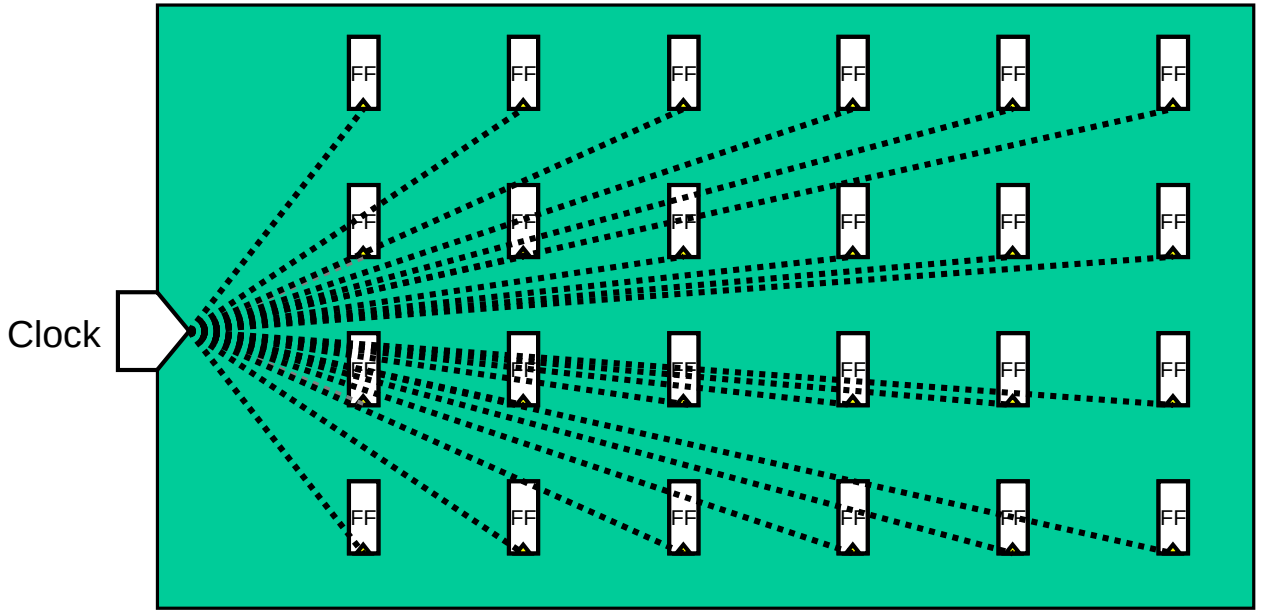
Zero skew: $d_1 = d_2$

Useful skew, $d_1 - d_2 = \delta_{12}$

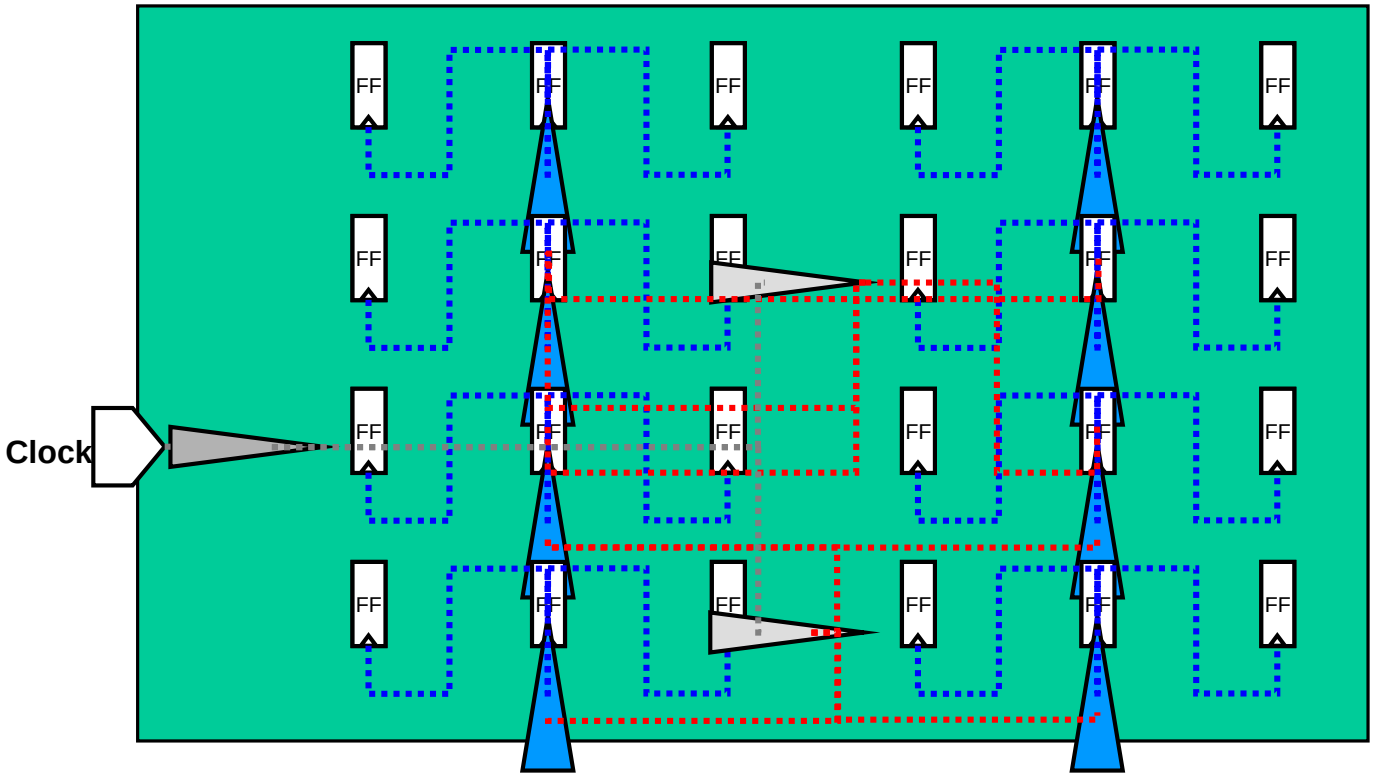


CTS Problem: Starting Point Before CTS

- All clock pins are driven by a single clock source
- All clock pins are from a source of clock pulses in various geometrical distances

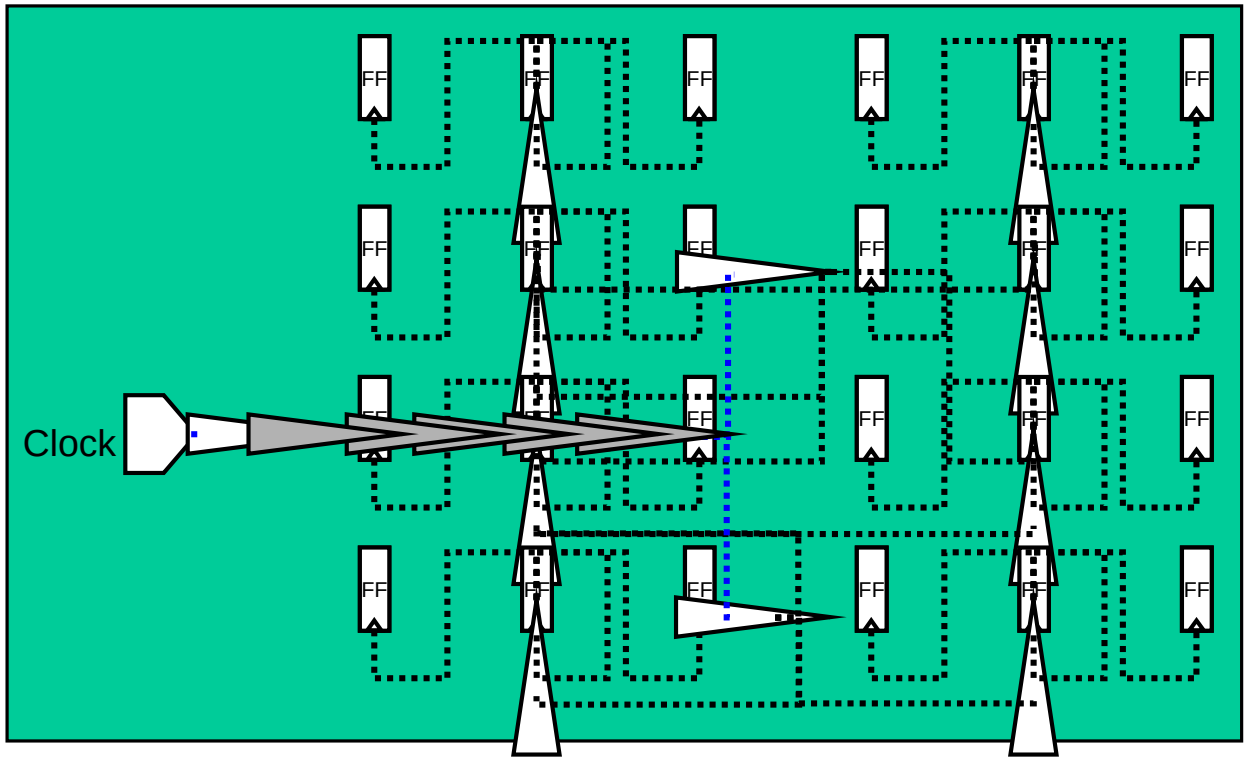


Generated and Gated Clocks: CTS (1)



A buffer tree is built to balance the loads and minimize the skew

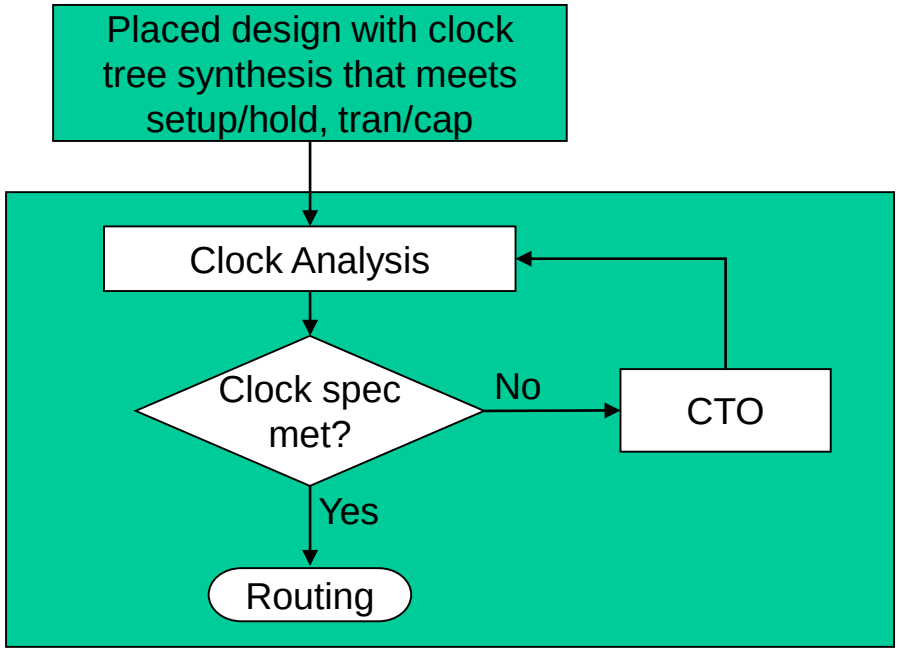
Generated and Gated Clocks: CTS (2)



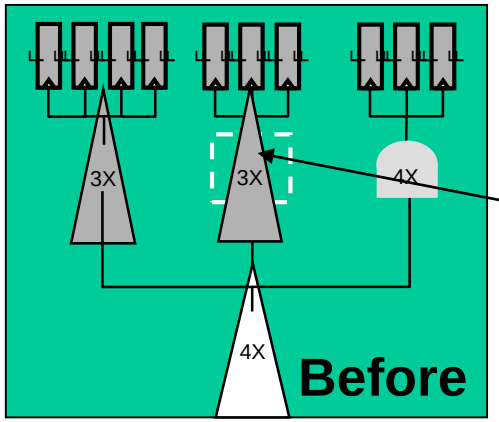
A delay line is added to meet the minimum insertion delay

Clock Tree Optimization (CTO)

Performing additional Clock Tree Optimization as necessary to further improve clock skew



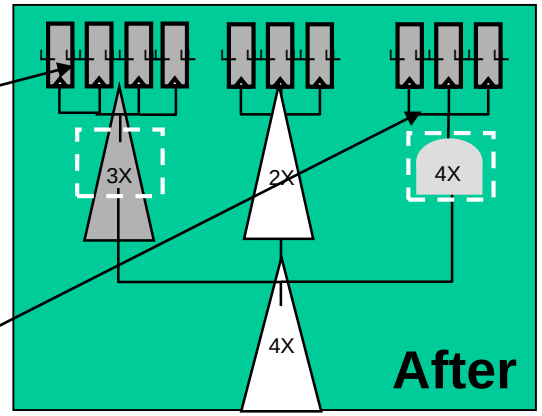
Clock Tree Optimization Options

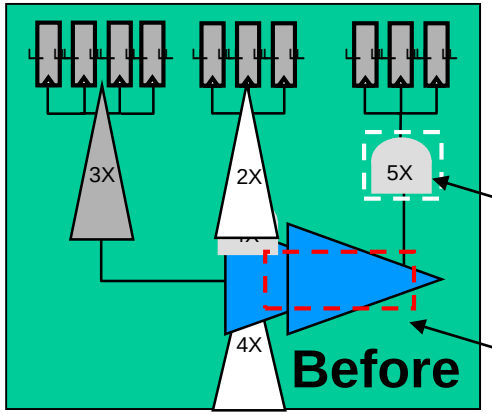


Buffer sizing

Buffer relocation

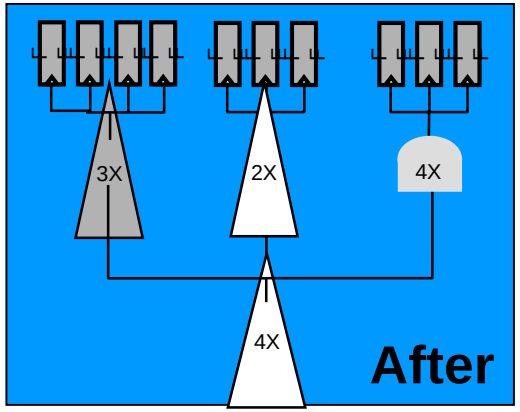
Gate relocation





Gate sizing

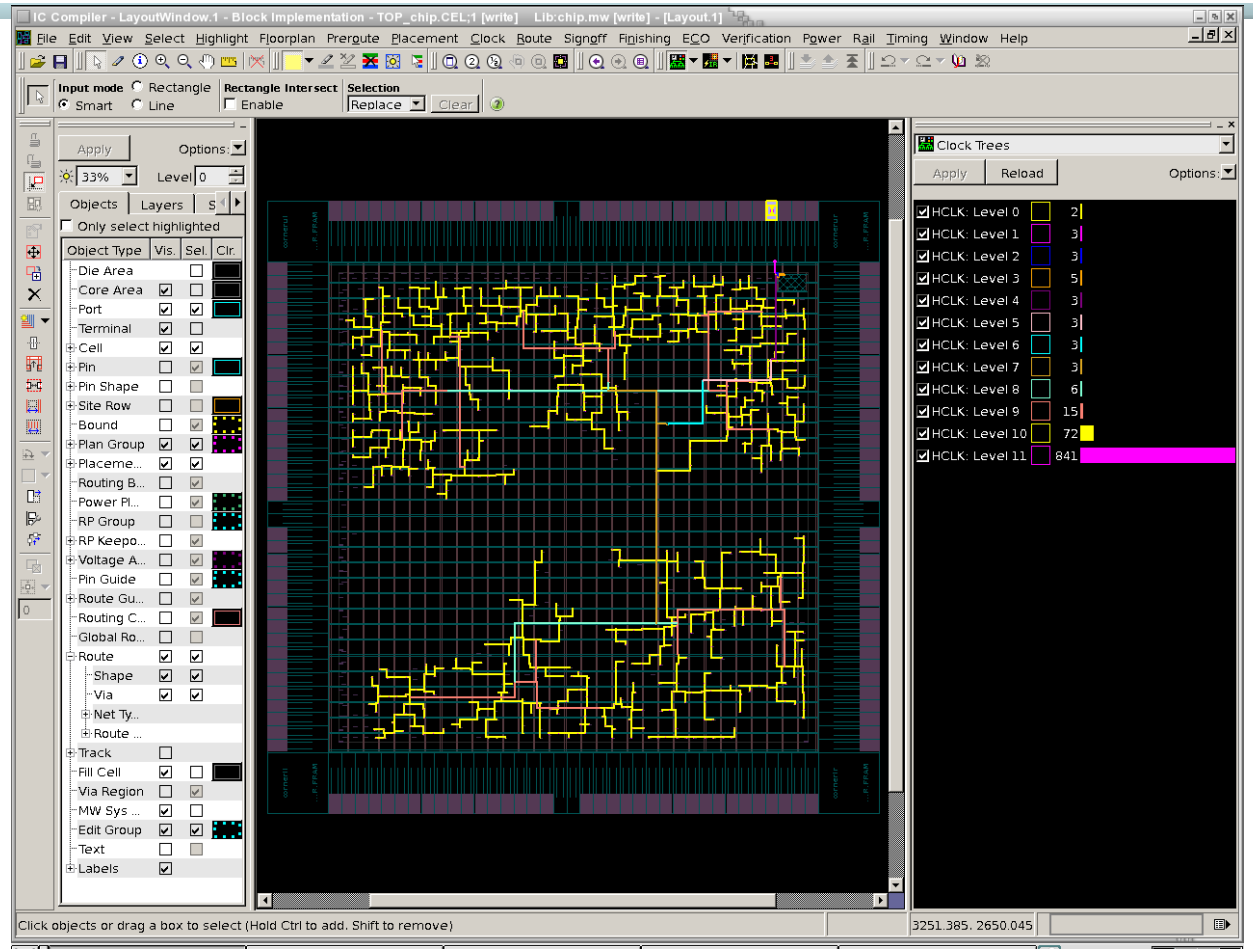
Delay insertion



CTS Problem: Prerequisites for Clock Tree Synthesis

- Before running CTS, the design must meet the following requirements:
  - The design should be placed and optimized
  - Placement – completed; Power and ground nets – prerouted
  - Estimated congestion – acceptable
  - Estimated timing – acceptable (~0ns slack)
  - Estimated max cap/transition – no violations
  - High fanout nets
    - Reset, Scan Enable synthesized with buffers
    - Clocks are still not buffered

Clock Tree



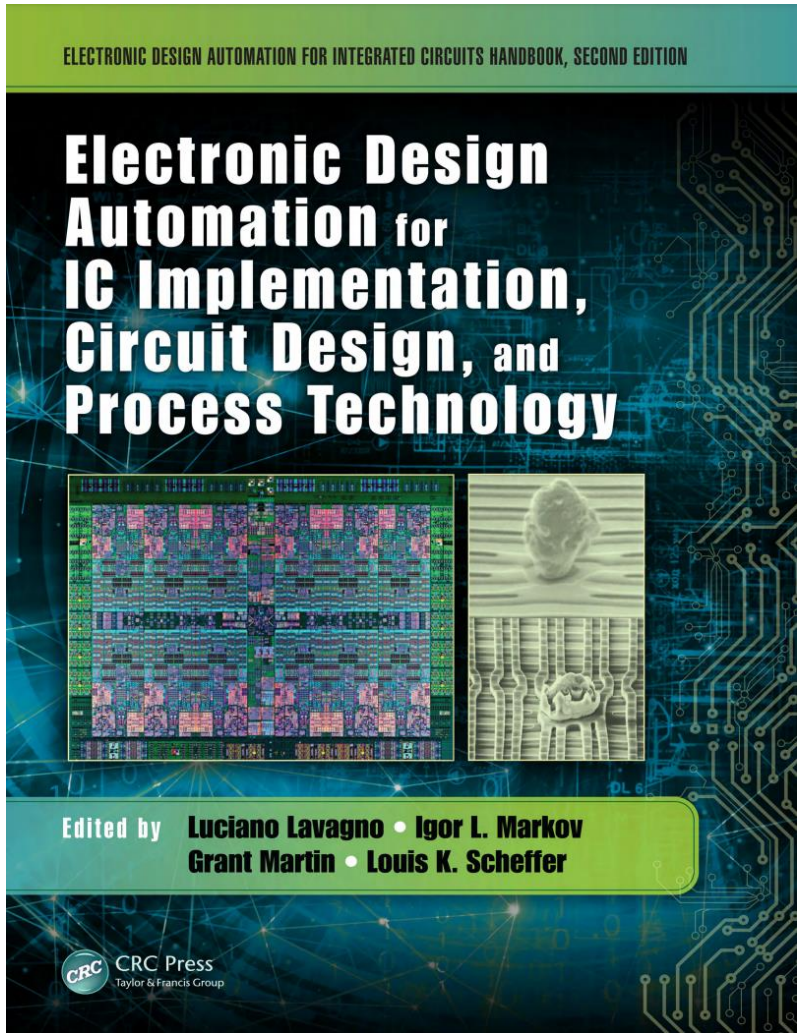
Design view after of CTS

Where are we Heading?

- ASIC Design Flow IV

Action Items

- HW#4 is coming!
- Reading Materials
  - Slides



Acknowledgement

Slides in this topic are inspired in part by material developed and copyright by:

- Synopsys Courseware