

FloorPlanning

How to Plan your own chip

Ahmed Abdelazeem

Faculty of Engineering
Zagazig University

RTL2GDSII Flow, February 2022

Table of Contents

1 Introduction

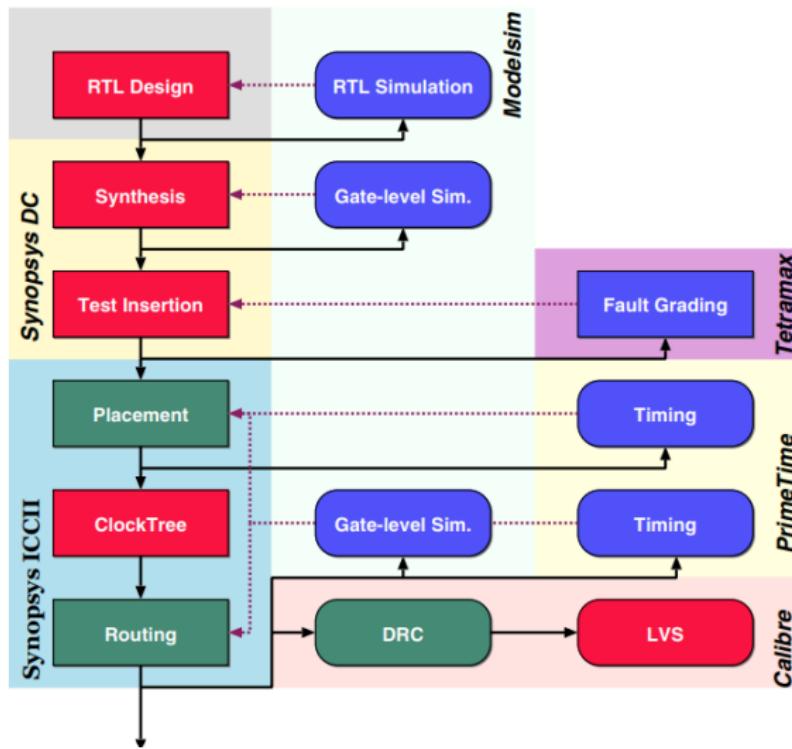
2 Floorplanning

Table of Contents

1 Introduction

2 Floorplanning

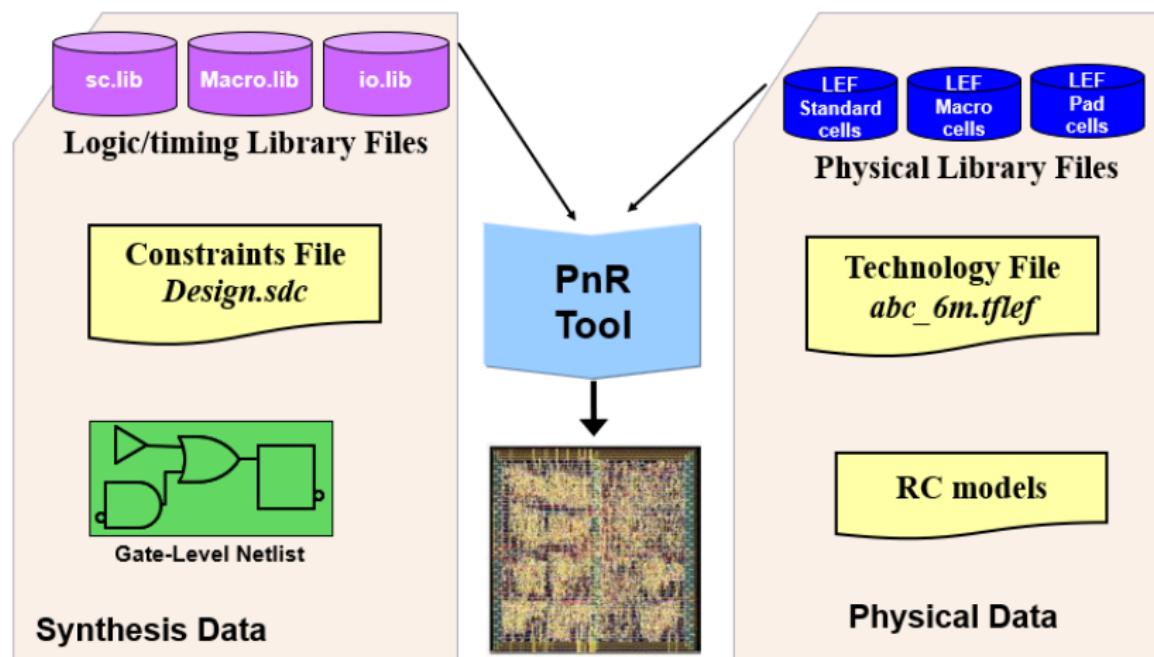
Overall Design Flow



The Big Picture...

b

Data Setup



Logical Libraries

- Provide **timing** and **functionality** information for all standard cells (and, or, flipflop, ...)
- Provide **timing** information for hard macros(IP, ROM, RAM, ...)
- Define drive/load design rules:
 - Max fanout
 - Max transition
 - Max/Min capacitance
- Are usually the same ones used by Synthesis during synthesis

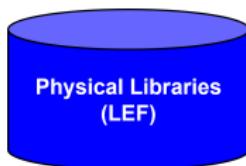


Timing Constraints

- “Timing Constraints” are required to communicate the design’s timing intentions to PnR Tool
- They should be the same ones used for synthesis with Synthesis Tool (preferably SDC)

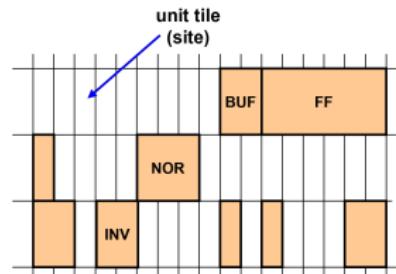
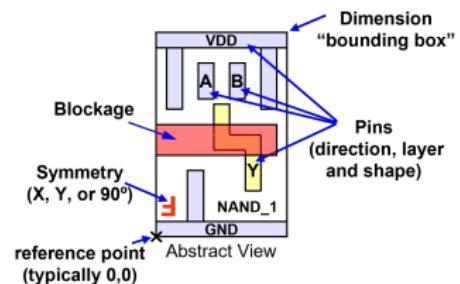
```
create_clock -period 10 [get_ports clk]
set_input_delay 4 -clock clk \
    [get_ports sd_DQ[*]]
set_output_delay 5 -clock clk
    [get_ports sd_LD]
set_load 0.2 [get_ports pdevsel_n]
set_driving_cell -lib_cell buf5 \
    [get_ports pdevsel_n]
...
...
```

Physical Libraries



- Contain physical information of standard, macro and pad cells, necessary for placement and routing
- Define placement unit tile

- Height of placement rows
- Minimum width resolution
- Preferred routing directions
- Pitch of routing tracks



Technology File (.tflef file)

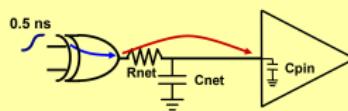
- Tech File is unique to each technology
- Contains metal layer technology parameters:
 - Number and name designations for each layer/via
 - Dielectric constant for technology
 - Physical and electrical characteristics of each layer/via
 - Design rules for each layer/Via (Minimum wire widths and wire-to-wire 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"  
    lengthPrecision     = 1000  
    gridResolution      = 5  
    unitVoltageName     = "v"  
}  
  
...  
  
Layer "m1" {  
    layerNumber         = 16  
    maskName           = "metall1"  
    pitch               = 0.56  
    defaultWidth        = 0.23  
    minWidth            = 0.23  
    minSpacing          = 0.23  
}  
...
```

abc_6m.tf

Timing is Based on Cell and Net Delays



$$\text{Cell Delay} = f(\text{Input Transition Time}, C_{\text{net}} + C_{\text{pin}})$$

$$\text{Net Delay} = f(R_{\text{net}}, C_{\text{net}} + C_{\text{pin}})$$

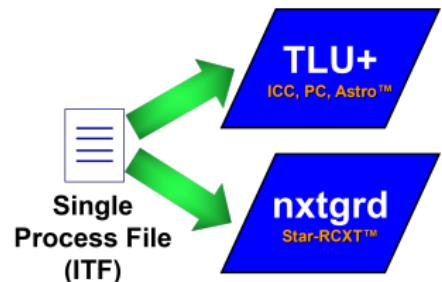
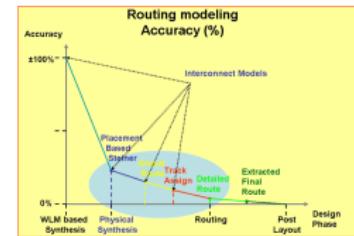
- PnR tool calculates delay for every cell and every net
- To calculate delays, tool needs to know each net's parasitic Rs and Cs

RC Models

- Tool calculates C and R using the net geometry and the TLU+ look-up tables
- UDSM process effects modeled

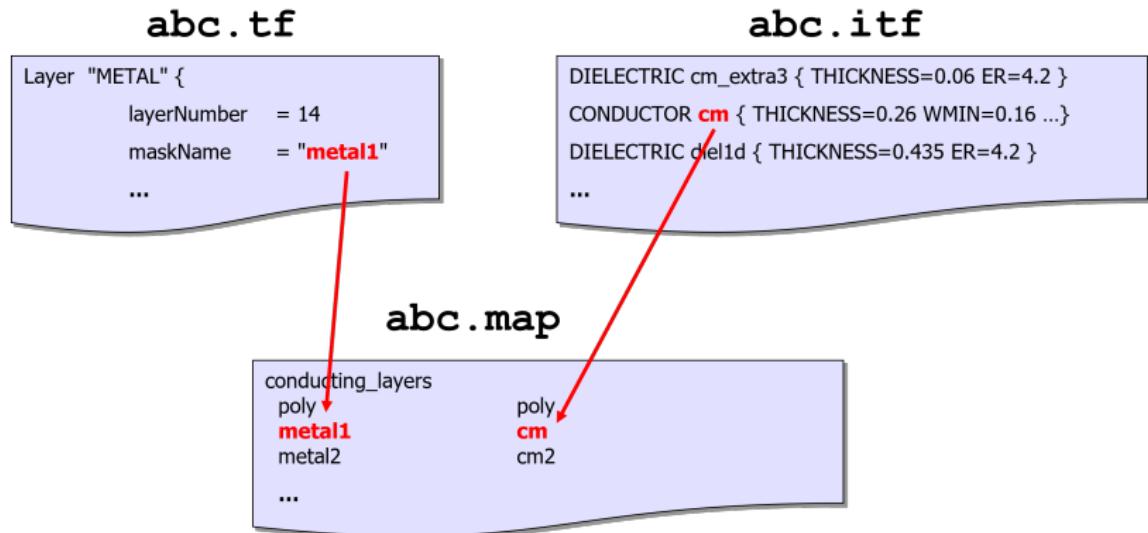
UDSM Process Effects

- Conformal Dielectric
 - Metal Fill
 - Shallow Trench Isolation
 - Copper Dishing:
 - Density Analysis
 - Width/Spacing
 - Trapezoid Conductor
- See Appendix B for details



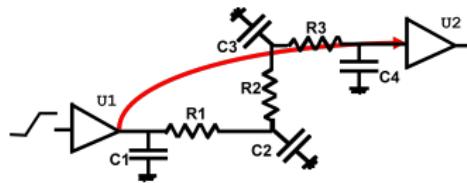
Mapping file

The Mapping File maps the .tf (lef technology file) layer/via names to Star-RCXT .itf layer/via names.



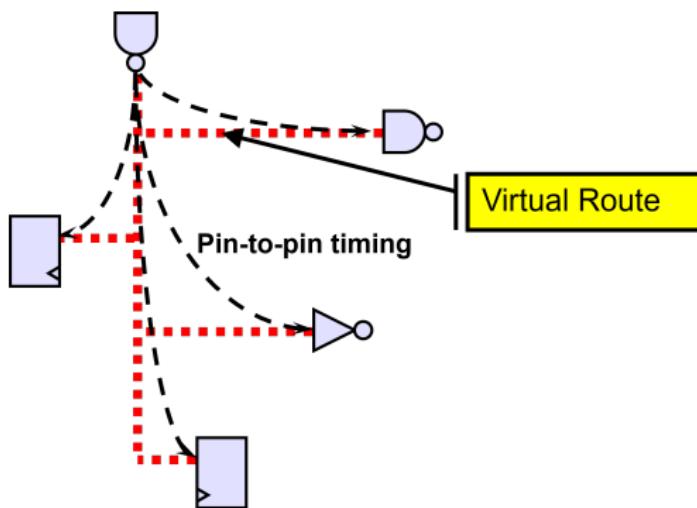
Calculating Cell and Net Delay

- Now that **R** and **C** are known from TLU+, the delays can be calculated
- For Cell Delays, only C_{total} / C_{eff} is needed



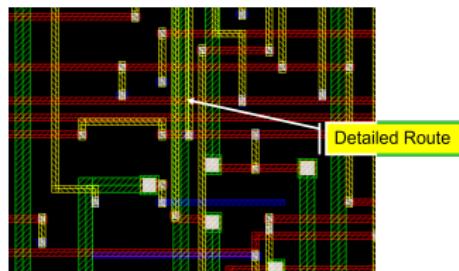
- Calculating **Net Delay** is done using Delay Calculation algorithms: Elmore, Arnoldi

PreRoute Delay Calculation Algorithm



- Prior to routing, net geometry is estimated based on a Virtual Route
- Since Virtual Routing is only an estimate, an Elmore model is used for delay calculation

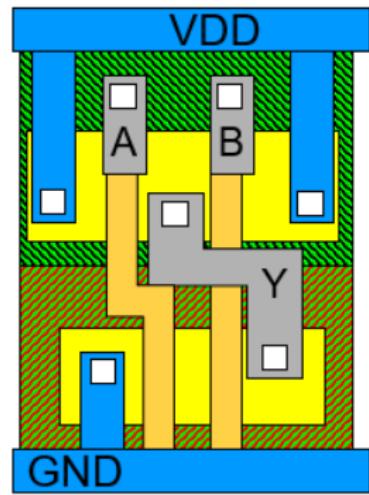
PostRoute Delay Calculation Algorithms



- After routing, detailed nets are available and extraction can be more accurate
- By default, Elmore is still used
- **Arnoldi** can be turned on for postroute calculations

What is a Standard Cell Library?

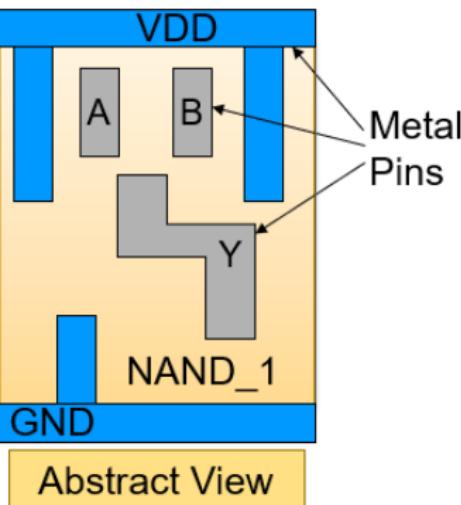
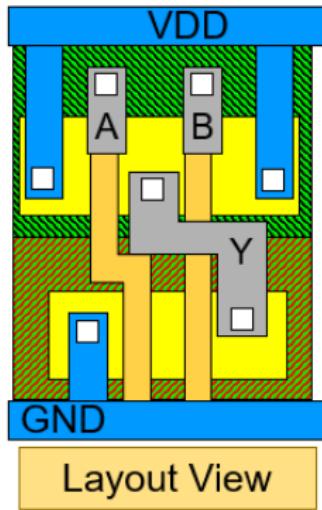
- A Standard Cell is a predesigned layout of one specific basic logic gate
- Each cell usually has the same standard height.
- A Standard Cell Library contains a varied collection of standard cells
- Libraries are usually supplied by an ASIC vendor or library group



Layout View
2-Input NAND Gate

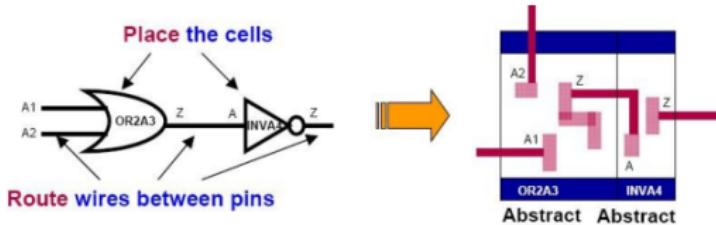
“Layout” vs. “Abstract” Views

- A standard cell library also contains a corresponding abstract view for each layout view
- Abstract views contain only the minimal data needed for Place & Route



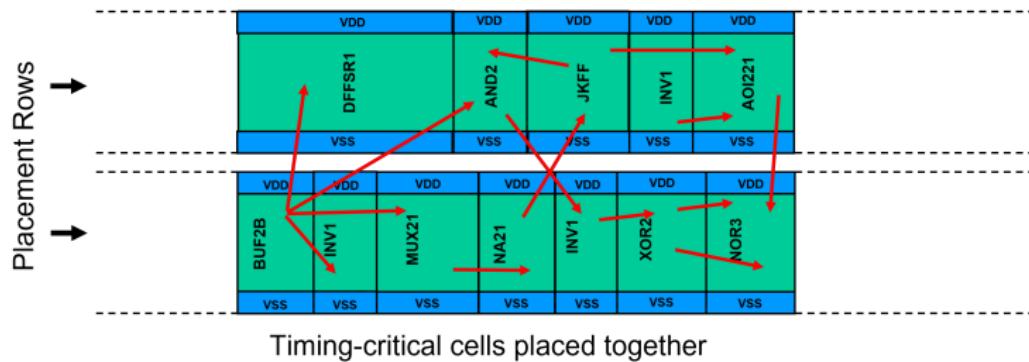
What Does “Place and Route” Do?

- Layout is built with three types of reference cells:
 - Macro cells (ROMs, RAMs, IP blocks)
 - Standard cells (nand2, inv, dff, ...)
 - Pad cells (input, output, bi-dir, Vdd, Vss pads)
- You have to define Macro and Pad cell locations during the Floorplanning stage, before Placement and Routing
- Location of all Standard Cells is automatically chosen by the tool during Placement, based on routability and timing
- Pins are then physically connected during Routing, based on timing



Timing-Driven Placement

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



Abutted Rows

- Placement rows are commonly abutted to reduce core area
- Cell orientations in abutted rows are flipped

VDD	VDD	VDD	VDD	VDD	VDD
DFFSR1	AND2	JKFF	INV1	AOI221	
VSS	VSS	VSS	VSS	VSS	

VDD	VDD	VDD	VDD	VDD	VDD
BUF2B	INV1	MUX21	NA21	INV1	XOR2
VSS	VSS	VSS	VSS	VSS	VSS

Non-abutted Rows



VDD	VDD	VDD	VDD	VDD	VDD	VDD
DFFSR1	AND2	JKFF	INV1	AOI221		
VSS	VDD	VDD	VDD	VDD	VDD	VDD

VDD	VDD	VDD	VDD	VDD	VDD	VDD
BUF2B	INV1	MUX21	NA21	INV1	XOR2	NOR3
VSS	VSS	VSS	VSS	VSS	VSS	VSS

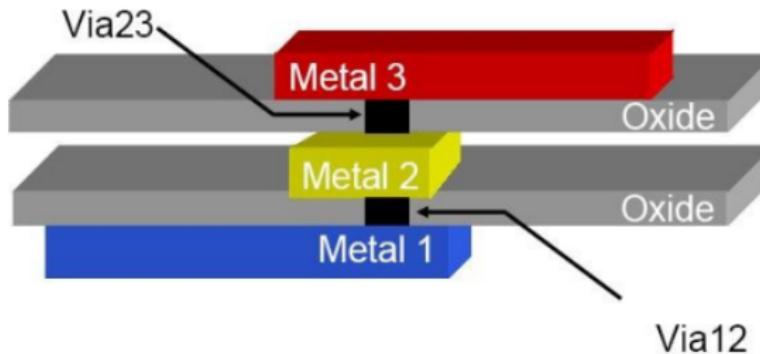
abutted Rows

Vias: Connecting Metal to Metal

- Connecting between metal layers requires one or more vias.

Example

Connecting a signal from Metal 1 to Metal 3 requires two vias and an intermediate Metal 2 connection



Preferred Routing Directions

- Metal layers have preferred routing directions
- Default preferred direction:
 - Metal 1 – Horizontal
 - Metal 2 – Vertical
 - Metal 3 – Horizontal, etc
- Why is this beneficial?



Preferred Routing Directions

- Metal layers have preferred routing directions
- Default preferred direction:
 - Metal 1 – Horizontal
 - Metal 2 – Vertical
 - Metal 3 – Horizontal, etc
- Why is this beneficial?

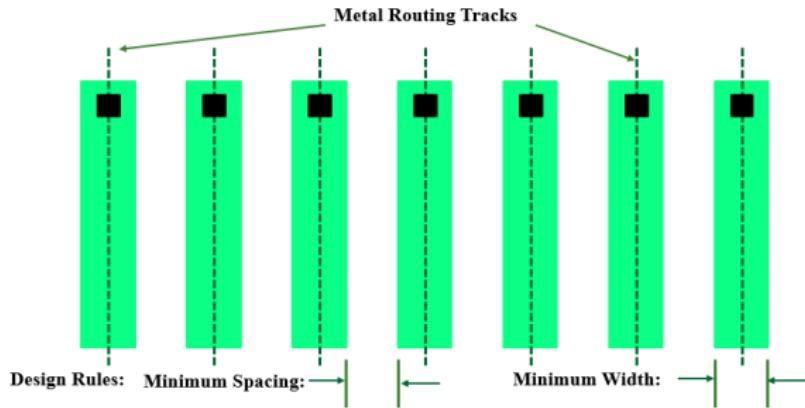


preferred routing directions

Having preferred routing directions greatly reduces the amount of metal layer “jumping” the router may need to do to connect any two pins, which reduces resistance and therefore propagation delay, as well as run time.

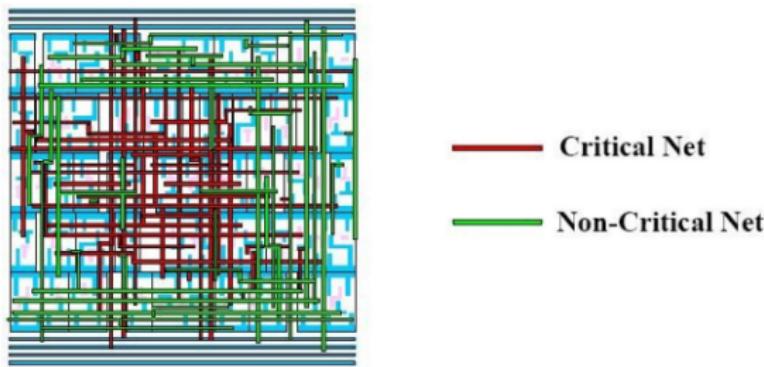
Routing Tracks

- Metal routes must meet minimum width and spacing “design rules” to prevent open and short circuits during fabrication
- In gridded routers these design rules determine the minimum center-to-center distance for each metal layer, a.k.a. grid or track spacing
- Congestion occurs if there are more wires to be routed than available tracks



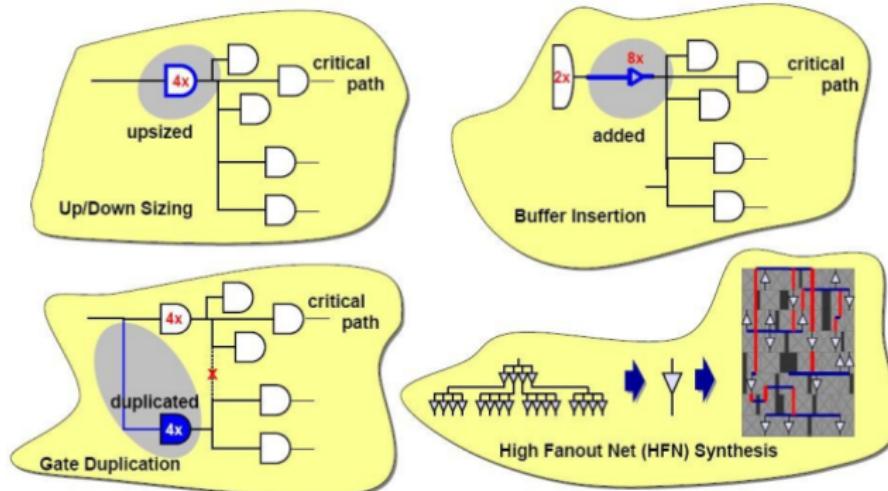
Timing-Driven Routing

- Routing along the timing-critical path is given priority:
 - Creates shorter, faster connections
- Non-critical paths are routed around critical areas:
 - Reduces routability problems for critical paths
 - Does not adversely impact timing of non-critical paths



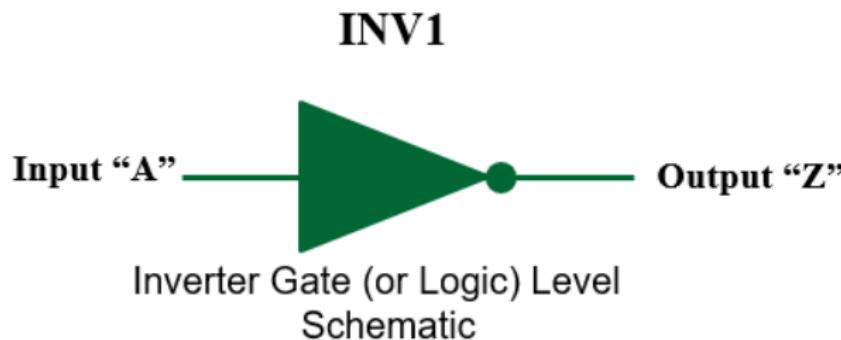
Logic Optimization

What if critical paths do not meet timing/drive requirements, even with timing-driven placement or routing?



What is a Gate in “Gate-level Netlist”?

Gate: Basic Logic Component



- Other Gates:
Buffer, Nand, Nor, Xor, AOI, Mux, D-FF, Latch, etc

Transistor or Device Representation

CMOS Inverter Example



Gate Schematic

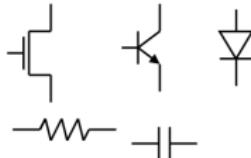


Transistor or Device View

Gates are made up of active devices or transistors.

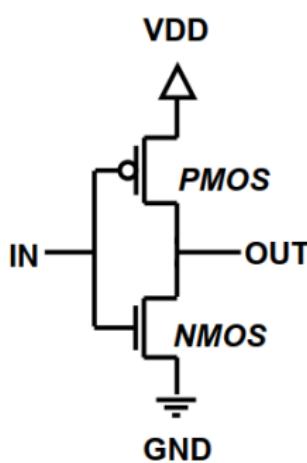
Basic Devices and Interconnect

- Integrated circuits are built out of active and passive components, also called devices:
 - Active devices:
 - Transistors
 - Diodes
 - Passive devices:
 - Resistors
 - Capacitors
- Devices are connected together with polysilicon or metal interconnect:
 - Interconnect can add unwanted or parasitic capacitance, resistance and inductance effects
- Device types and sizes are process or technology specific:
 - The focus here is on CMOS technology

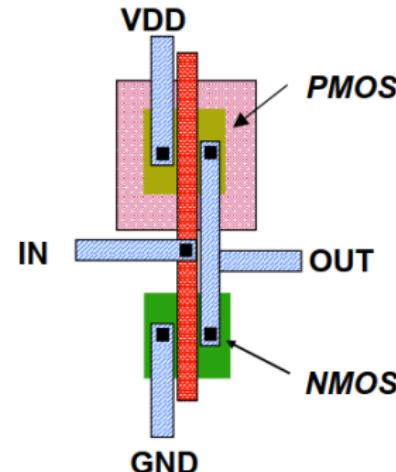


What is “Physical Layout” ?

CMOS Inverter Example



Transistor or Device View

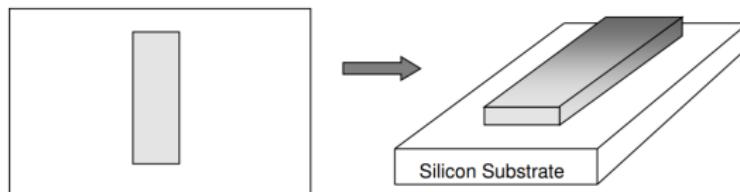


Physical or Layout View

Physical Layout – Topography of devices and interconnects, made up of polygons that represent different layers of material.

Process of Device Fabrication

- Devices are fabricated vertically on a silicon substrate wafer by layering different materials in specific locations and shapes on top of each other
- Each of many process **masks** defines the shapes and locations of a specific layer of material (diffusion, polysilicon, metal, contact, etc)
- Mask shapes, derived from the layout view, are transformed to silicon via photolithographic and chemical processes

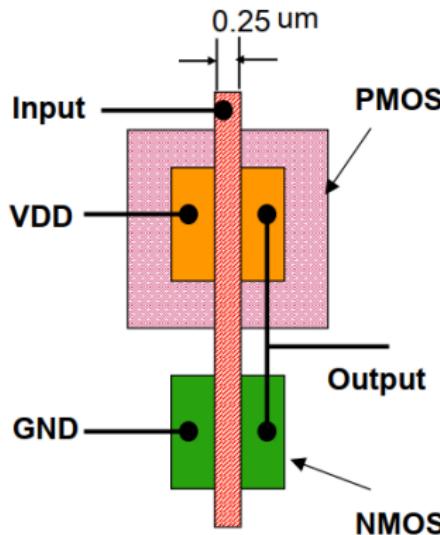


Layout or Mask (aerial) view

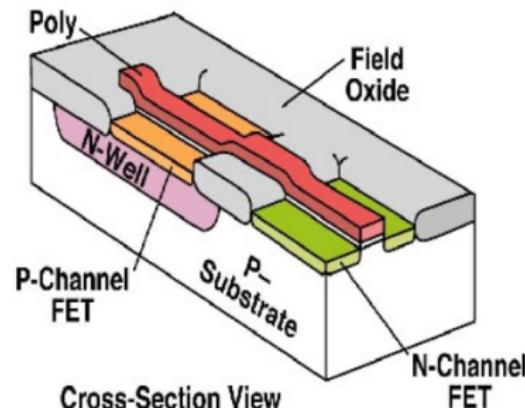
Wafer (cross-sectional) view

Wafer Representation of Layout Polygons

Example of complimentary devices in 0.25 μm CMOS technology or process.



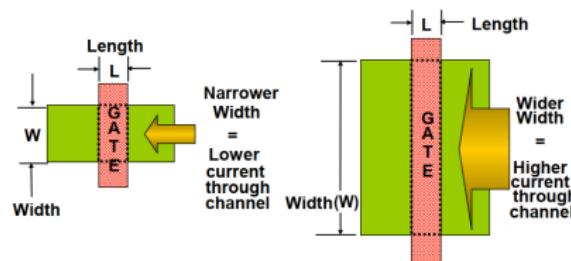
Aerial or Layout View



Wafer Cross-sectional View

What is Meant by “0.xx um Technology”?

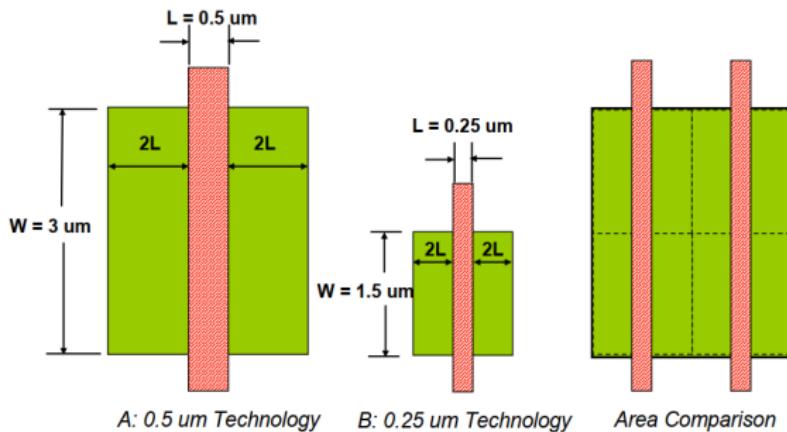
Gate or Channel Dimensions (L and W)



0.xx um Technology

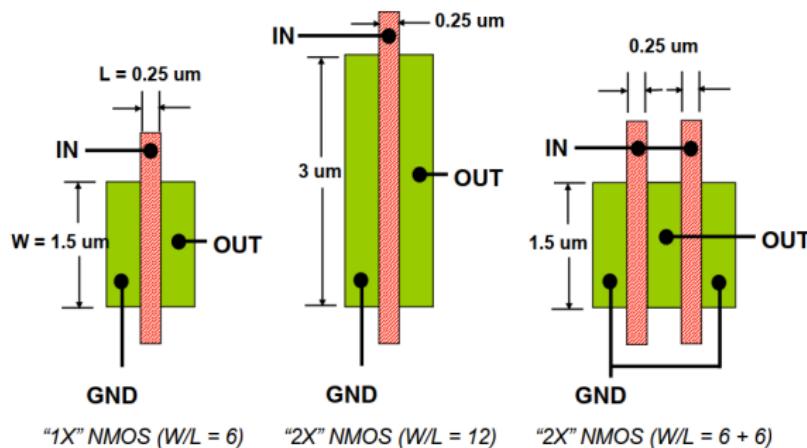
- In CMOS Technology the um or nm dimension refers to the channel length, a minimum dimension which is fixed for most devices in the same library.
- Current flow or drive strength of the device is proportional to W/L ; Device size or area is proportional to $W \times L$.

Comparing Technologies



The drive strength of both devices is the same: $W/L = 6$.
The diffusion area ($5 \times L \times W$) of A is 4x that of B.
Which is preferred?

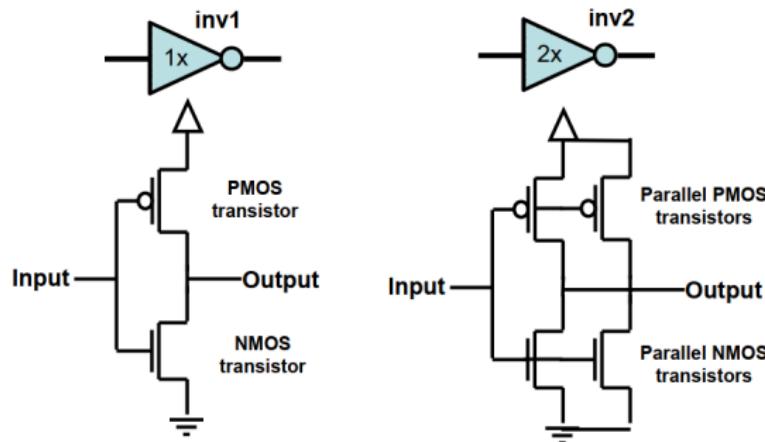
Relative Device Drive Strengths



Drive Strengths

To double the drive strength of a device, double the channel width (W), or connect two 1X devices in parallel. The latter approach keeps the height at a fixed or “standard” height.

Gate Drive Strength Example



Gate Drive

Each gate in the library is represented by multiple cells with different drive strengths for effective speed vs. area optimization.

Drive/Buffering Rules: Max Transition/Cap

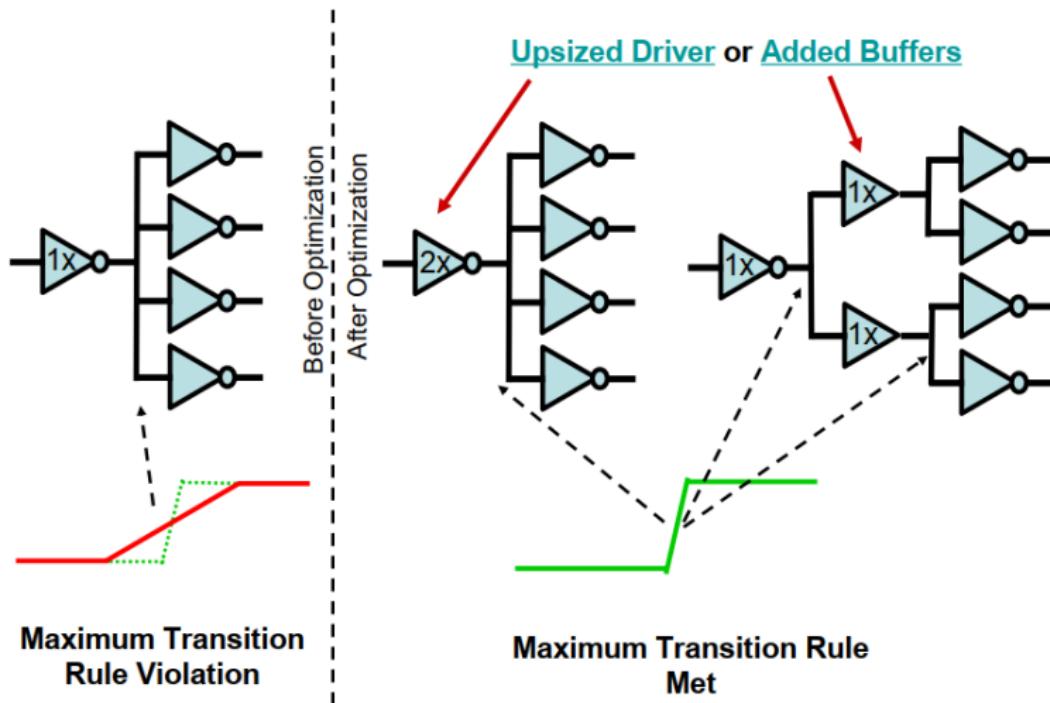


Table of Contents

1 Introduction

2 Floorplanning

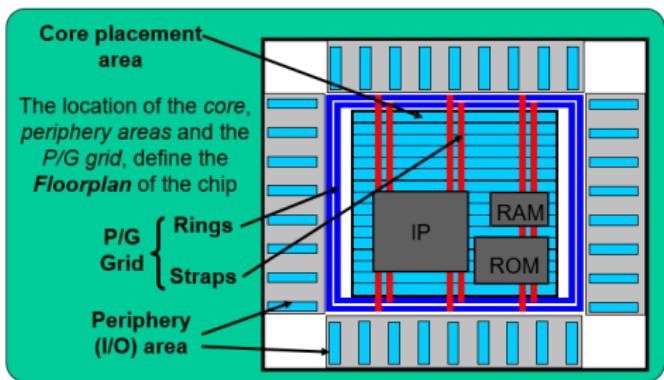
What Is Floorplanning?

Definition

Floorplanning is the process of deriving the die size, allocating space for soft blocks, planning power, and macro placement.

With a top-level netlist, you can start to floorplan the chip.

- Define Die Size.
- Place the IOs.
- Perform macro placement.
- Perform power planning.
- Power domain definition
- Flip-chip bump placement



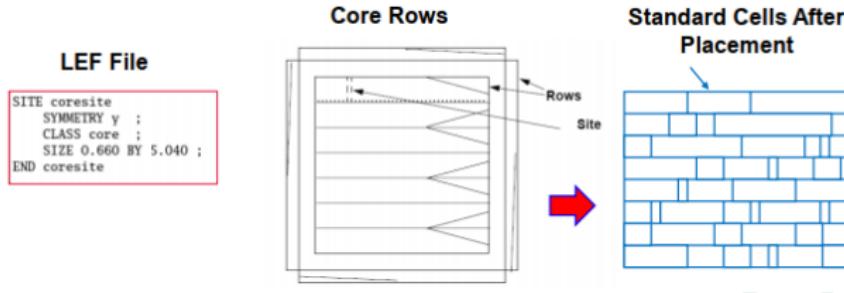
Floorplanning

- Floorplanning consists of defining the **Core Placement area**, the **Periphery area** and the **Power/Ground grid**.
- The Core Placement area consists of **Placement Rows** where standard cells and macro cells are placed.
- A placement row consists of a row of “**unit tile**” cells (part of the standard cell library, width defined by the minimum metal pitch). Standard cells are placed in the core of a chip and occupy specific tile(s) within the placement rows. **A standard cell may occupy a single or multiple tiles**.
- The pad cells (**input, output, bi-dir, power and ground pads**) are placed in the Periphery Area, which is defined by the area around the outside boundary of the core, usually separated by a “core-to-pad” spacing distance.

What Are Sites and Rows?

A site is the minimum unit of placement. It represents a slot where a cell can be placed. Rows are multiples of sites and define locations where the placement tool places cells.

Placement tools place cells in locations defined by the cell's description in the LEF file. If a cell is of type CORE, then it can only be placed in a CORE rows. If a cell is of type IO, it can only be placed in IO rows



Core and IO Region

- **From:**

- A gate level netlist
- Relevant physical libraries
- Default or user specified aspect ratio and utilization.

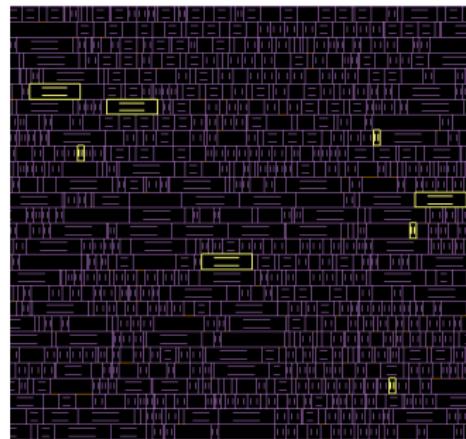
- Calculate the area of all macro cells and leaf cells
- Generate bounding shapes and cell placement rows
- Place IO PADS

- Signal pads
- Filler and corner pads
- Bump or flip-chip IO pads

$$\text{Utilization} = \frac{(\text{Total Std Cell} + \text{Macro Cell Area})}{\text{Core Area}} * 100\% \quad (1)$$

Utilization: A Factor in Determining Core Size

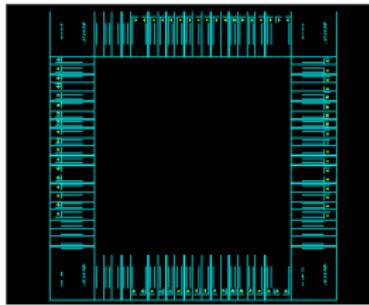
- Core “utilization” is the percentage of the core that is used by placed std cells and macros
- Ideally would like to achieve 100% utilization at tape-out. In practice range is 80-85
- Recommended starting netlist utilization should not exceed 60-75 optimizations and DFM



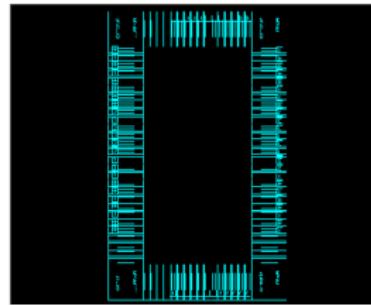
$$\text{Utilization} = \frac{(\text{Total Std Cell} + \text{Macro Cell Area})}{\text{Core Area}} * 100\% \quad (2)$$

Aspect Ratio: A Factor in Determining Core Size

- **Aspect ratio** is the ratio between vertical routing resources to horizontal routing resources.
- If you specify a ratio of 1.00, the height and width are the same and therefore the core is a square.
- If you specify a ratio of 2.00, the height is two times the width.



Floorplan With No Options Specified



Floorplan With Aspect Ratio of 2.0

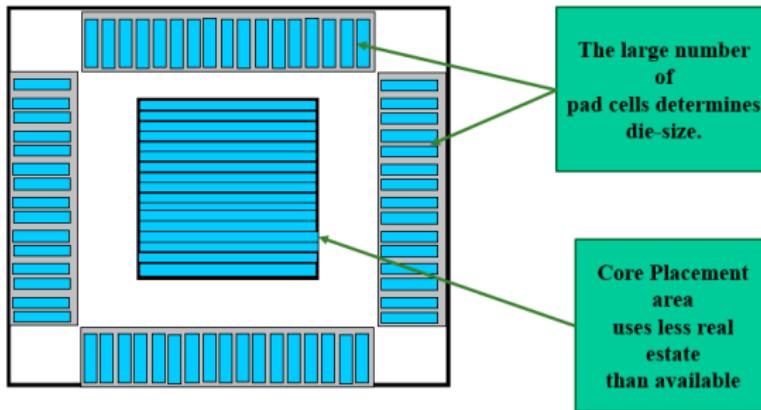
Balance Routing Resources

- If less vertical routing resources are available, make floorplan wider (aspect ratio < 1) if possible, to increase vertical routing resources
- If less horizontal routing resources are available, make floorplan taller (aspect ratio > 1) if possible, to increase horizontal routing resources

Note

Balancing vertical/horizontal routing resources reduces overall congestion

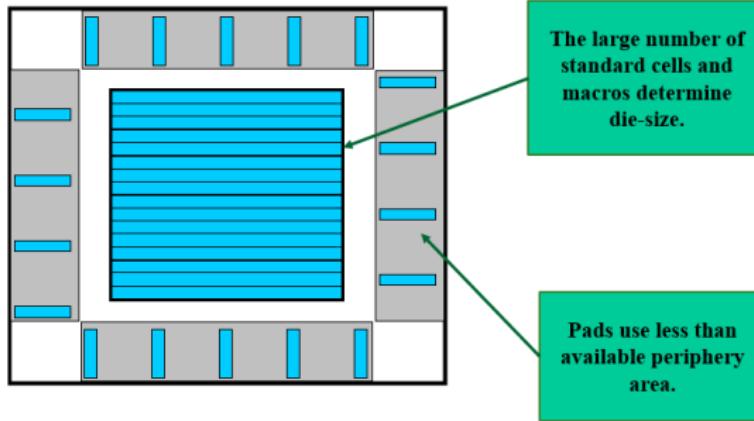
Pad-Limited Design



Question

If the utilization of a pad-limited design is too high during floorplanning will reducing it affect die size?

Core-Limited Design

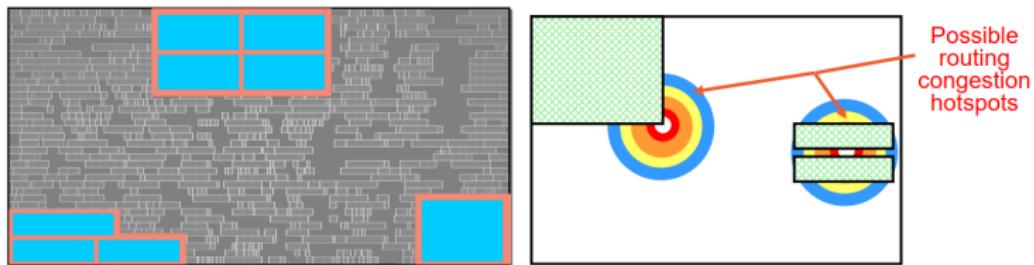


Question

If the utilization of a core-limited design is too high during floorplanning will reducing it affect die size?

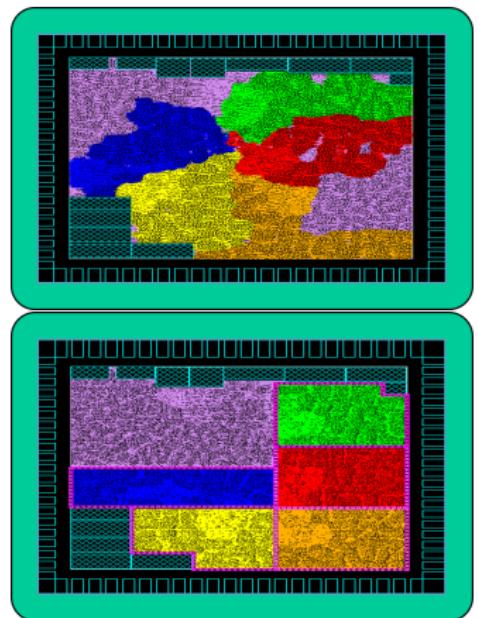
Hard Macro Placement

- When placing large macros we must consider impacts on routing, timing and power. Usually **push them to the sides** of the floorplan.
 - Placement algorithms generally perform better with a **single large rectangular placement area**
 - For wire-bond place power hungry macros **away from the chip center**
- After placing hard macros, mark them as **FIXED**.



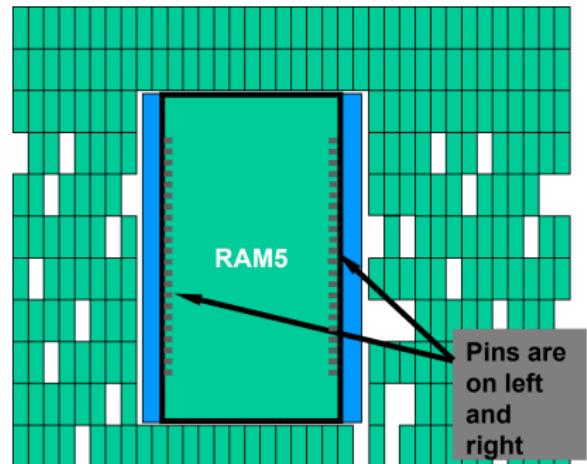
Placement Blockages and Halos

- Sometimes, we want to “help” the tool put certain logic in certain regions or cluster them together.
- Place and Route tools define several types of placement bounds:
 - **Soft move bounds** specify placement goals, with no guarantee that the cells will be placed inside the bounds.
 - **Hard move bounds** force placement of the specified cells inside the bounds.
 - **Exclusive move bounds** force the placement of the specified cells inside the bounds. All other cells must be placed outside the bounds.



Placement Blockages and Halos

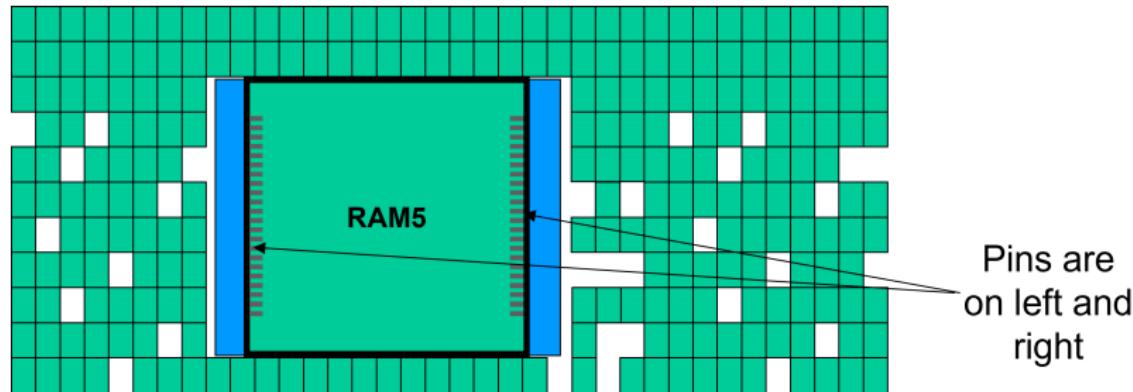
- Placement blockage **halos** are areas that the tools should not place any cells.
- These, too, have several types:
 - **Hard Blockage** – no cells can be placed inside.
 - **Soft Blockage** – cannot be used during placement, but may be used during optimization.
 - **Partial Blockage** – an area with lower utilization.
 - **Halo (padding)** – an area outside a macro that should be kept clear of standard cells



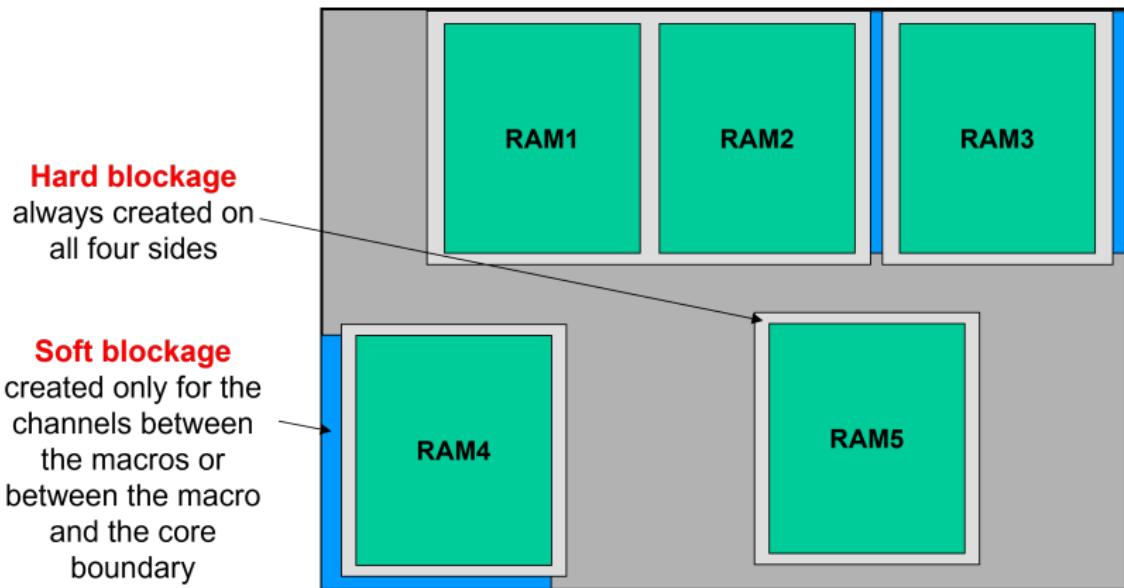
Placement Blockages: Macro Keepout Margin (Padding)

Padding

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

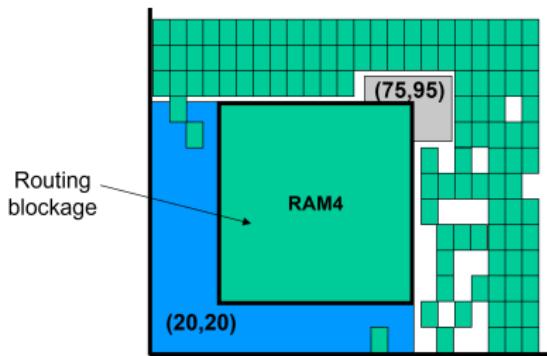


Placement Blockages: Adding or Modifying Global Placement Blockages

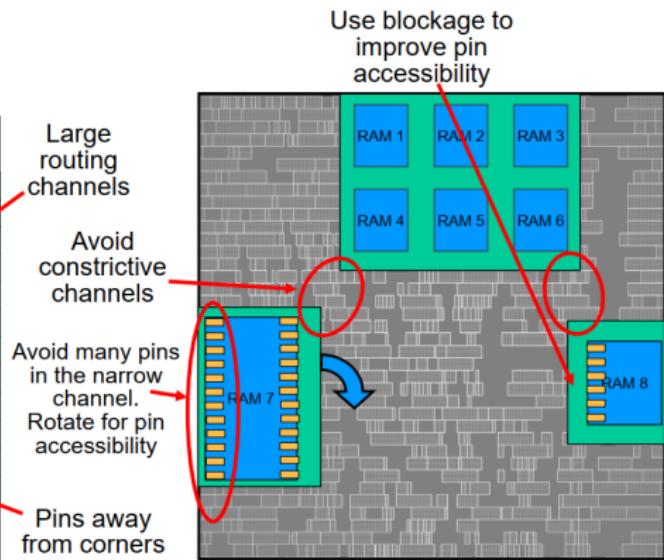
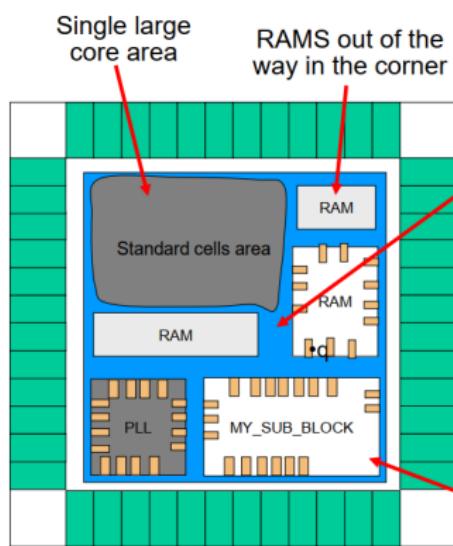


Placement Blockages: Routing Blockage (Route Guide)

- **Routing blockages** are used to prevent the route in a particular area for the specific metal layer for all nets or only signal nets or PG nets.
- **Routing blockages** – areas where routing is not allowed



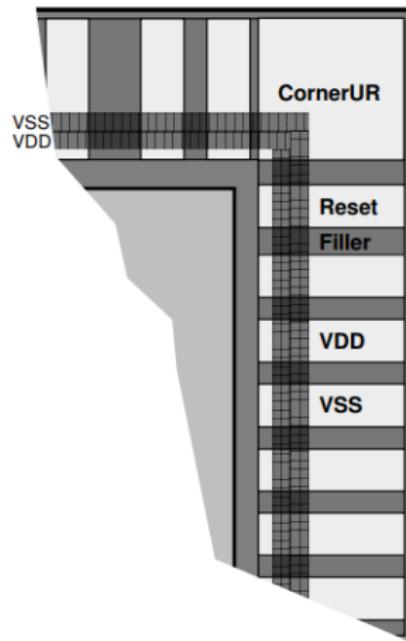
Guidelines for a good floorplan



Pad Area

■ Pad area consists of:

- Input/Output/InOut pads
- Power pads and corner pads
- Pad fillers
- P/G rings

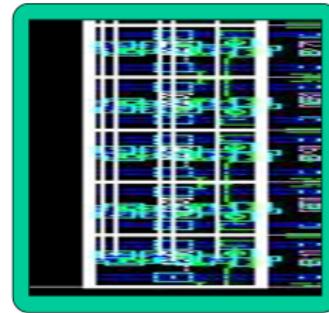


Pin Alignment

- Pin alignment is not less important than macros alignment.
- In the case of macros, wire length is reduced, and the area is increased. In the case of Pin alignment, wire length is reduced, and the area remains the same



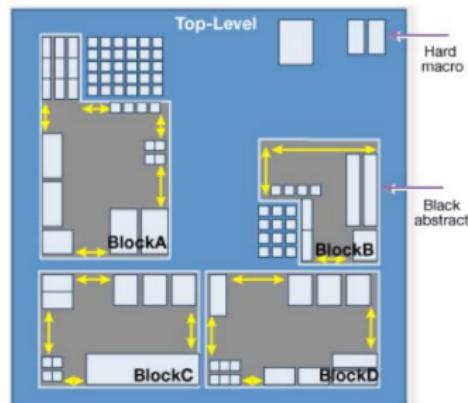
Before



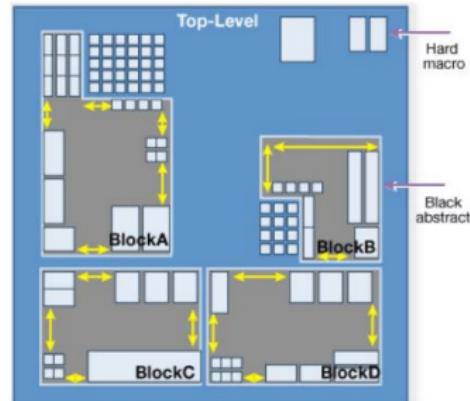
After

Hierarchical Approach

- Chip is partitioned into smaller blocks
- Each block is P&R'ed individually
- Blocks are integrated back into the chip



- **Chip has:**
 - Pads (signal and P/G)
- **Block has:**
 - Pins (signal and P/G)
- **Blocks can be rectangular or rectilinear in shape.**



....

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
وَمَا أُوتِيْتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا