
FloorPlan

Before starting with the floorplan we will perform **import design**, **sanity checks** and **partitioning**.

Import Design

- The following input files information are loaded to the PnR tool
 - Netlist (.v/ .vhd/ .edif)
 - Physical Libraries (.lef)
 - Timing Libraries (.lib)
 - Technology Files
 - Constraints (.sdc)
 - IO Info. File (optional)
 - Power Spec. File (optional)
 - Optimization Directives (optional)
 - Clock Tree Spec. File (optional at floorplan stage)
 - DEF/ FP (optional if floorplan is not done)
 - Core area is approximately calculated by the tool from the Netlist
 - While Importing, first we have to load the LEF files and then LIB files
-

Sanity Checks

- Sanity Checks mainly checks the quality of netlist in terms of timing.
- It also consists of checking the issues related to Library files, Timing Constraints, IOs and Optimization Directives

1. Library checks
 - Missing cell information
 - Missing pin information
 - Duplicate cells
2. Design checks
 - Inputs with floating pins
 - Nets with tri-state drivers
 - Nets with multiple drivers
 - Combinational loops
 - Empty modules
 - Assign statements
3. Constraint checks
 - All flops are clocked or not
 - There should not be unconstraint paths
 - Input and output delays

Partitioning

Physical Design Netlist

- All Ports must be defined and should be present
- No Assignment Statements (1'b0 or 1'b1 statements): Assignment statements causes feed-through (i/p directly to o/p) and can be avoided by adding buffers
- No Unmapped Cells
- No Combinational Timing Loops

Styles of Implementation

Flat

- Small to Medium ASIC
- Better Area Usage Since no reserve space around each sub-design for power/ground

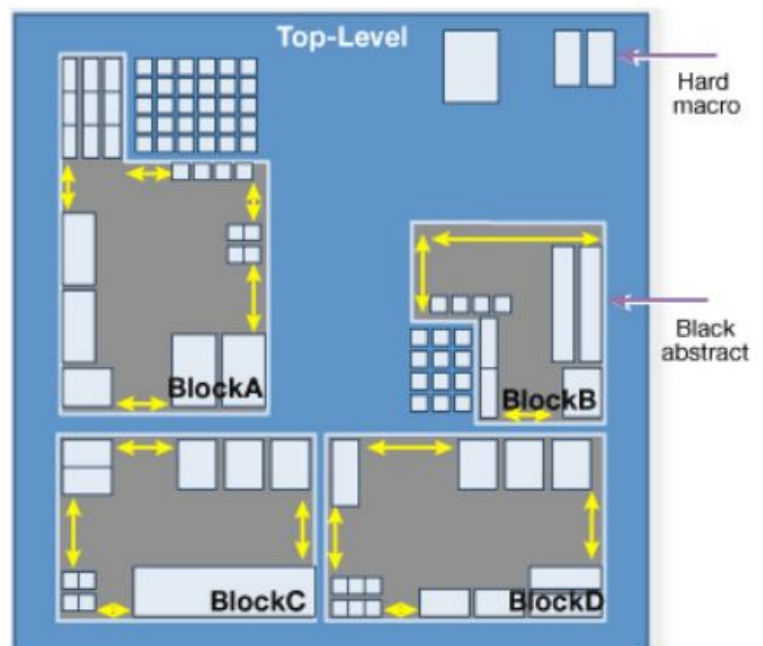
Hierarchical

- For very large design
- When sub-systems are design individually
- Possible only if a design hierarchy exist

- The Hierarchical Partitioning is done prior to Floorplan

- **Partition can be done based on**

- Design Hierarchy
- Timing Criticality
- Functionality
- Clock Domain
- Design Files
- Block Size



- Partitioning Inputs and Outputs by Registers
- Minimize Cross-Partition-Boundary IO
- For Sub-block designs, the Partitioning is not required
- For Full Chip only we need to design with Partitioning

FloorPlanning

Floorplan is one the critical & important step in Physical design. Quality of your Chip / Design implementation depends on how good is the Floorplan. A good floorplan can be make implementation process (place, cts, route & timing closure) cake walk. On similar lines a bad floorplan can create all kind issues in the design (congestion, timing, noise, IR, routing issues). A bad floorplan will blow up the area, power & affects reliability, life of the IC and also it can increase overall IC cost (more effort to closure, more LVTs/ULVTs).

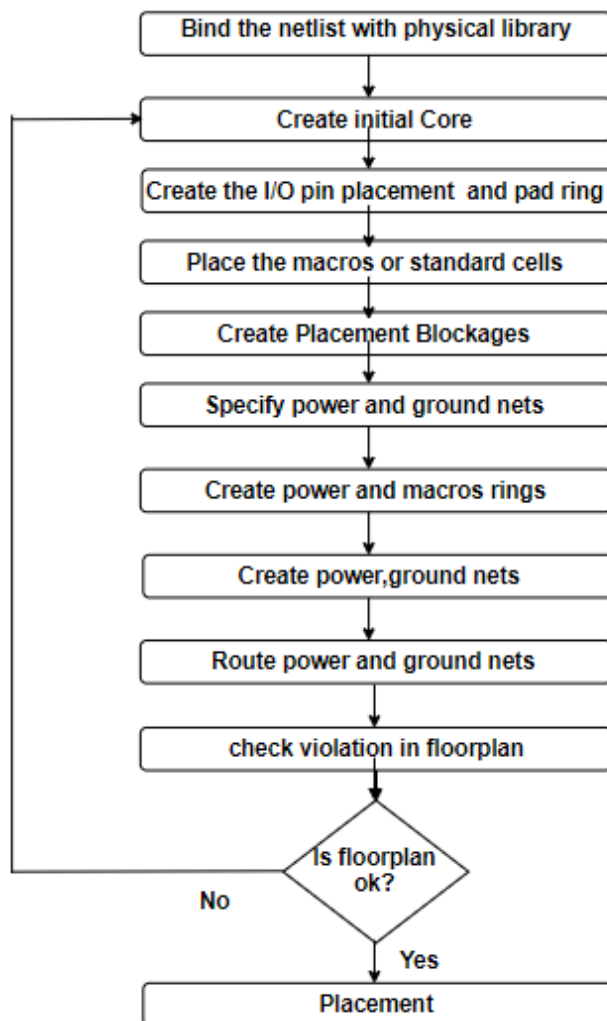
Objectives of Floorplan

- minimize the area
- minimize the Timing
- Reduce the wire length
- Making routing easy
- Reduce IR drop

Inputs of Floorplan

- Technology file (.tf)
- Netlist
- SDC
- Library files (.lib & .lef) & TLU+ file

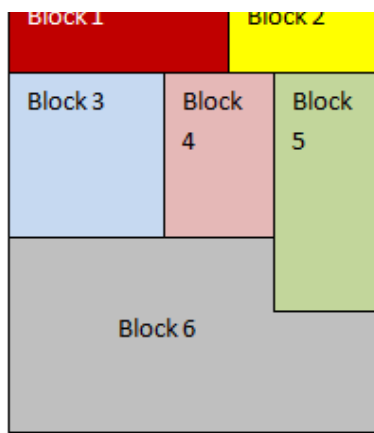
Floorplan Flowchart



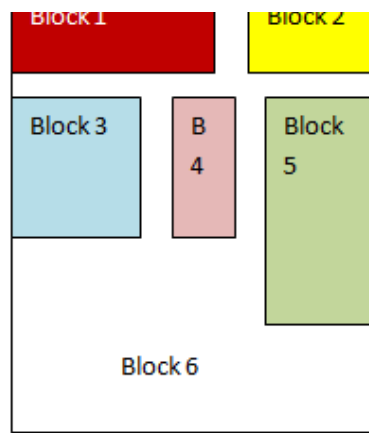
Types of Floorplan Techniques

- Abutted floorplan : Channel less placement of blocks.
- Non-Abutted Floorplan : Channel based placement of blocks.
- Mix of both: partially abutted with some channels.

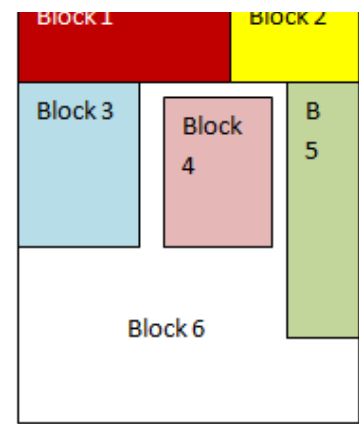




Abutted



Non abutted



Mixed

Terminologies and Definitions

Utilization

- Area of the core that is used by placed Standard Cells and Macros expressed in percentage

Manufacturing Grid

- The smallest geometry that semiconductor foundry can process or smallest resolution of your technology process (e.g. 0.005)
- All drawn geometries during Physical Design must snap to this grid
- While Masking fab. use this as reference lines

Standard Cell Site/ Standard Cell Placement Tile/ Unit Tile

- The minimum Width and Height a Cell that can occupy in the design
- The Standard Cell Site will have the same height as Standard Cells, but the width will be as small as your smallest Filler Cell
- It's one Vertical Routing Track and the Standard Cell Height
- All Standard Cells must be multiple of Unit Tile

Standard Cell Rows

- Rows are actually the Standard Cell Sites abut side by side and then Standard Cells are placed on these Rows
- Cells with the equal no. of Track definition will have same height

Placement Grid

- Placement Grid is made up of Standard Cell Site
- Its always a multiple of Manufacturing Grid
- Placement Grid is made up of the Rows which are composed of Sites

Routing Grid and Routing Track

- Horizontal and Vertical line drawn on the layout area which will guide for making interconnections
- The Routing Grid is made up of the Routing Tracks
- Routing Tracks can be Grid-based, Gridless based or Subgrid-based

Flight-line/ Fly-line

- Virtual connection between Macros and Macro or Macros and IOs

Macro

- Any instances other than Standard Cell and is as loaded as black box to the design is Macro
- Intellectual Property (IP) e.g. RAM, ROM, PLL, Analog Designs etc.
- Hard Macro: IP with Layout implemented
- Soft Macro: IP without Layout implemented (HDL)

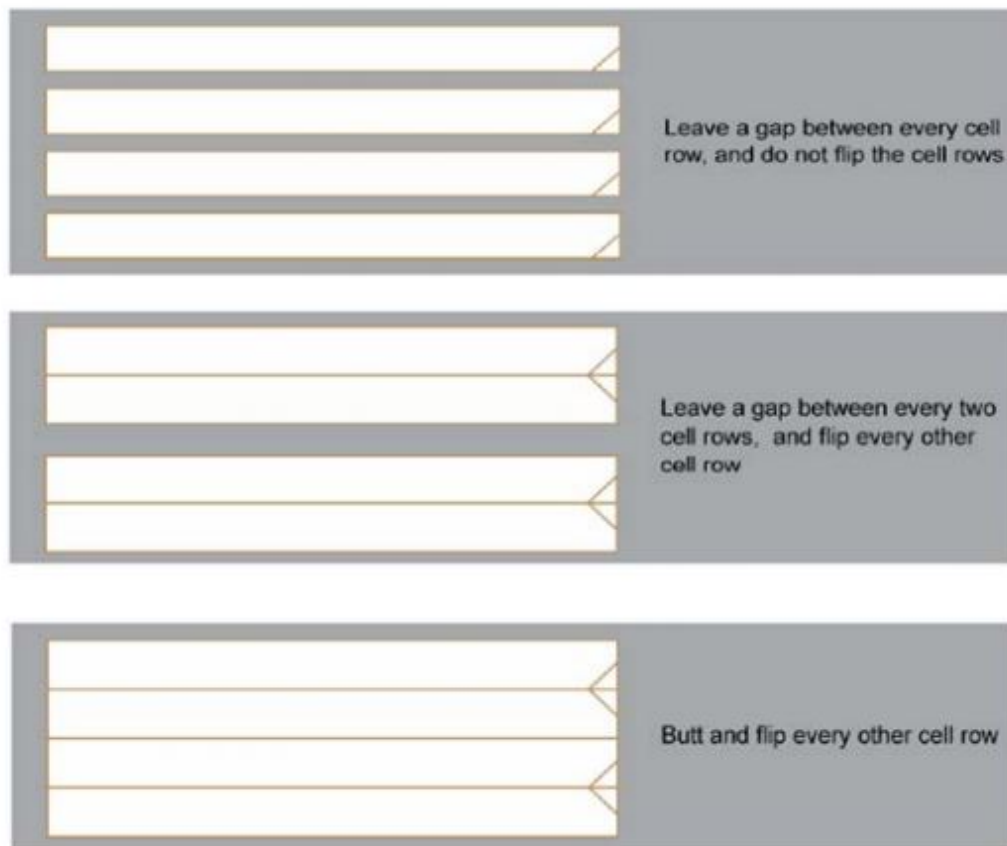
Steps in Floorplan

- Initialize with Chip & Core Aspect Ratio (AR)
- Initialize with Core Utilization
- Initialize Row Configuration & Cell Orientation
- Provide the Core to Pad/ IO spacing (Core to IO clearance)
- Pins/ Pads Placement
- Macro Placement by Fly-line Analysis
- Macro Placement requirements are also need to consider
- Blockage Management (Placement/ Routing)

Utilization

Row Configuration

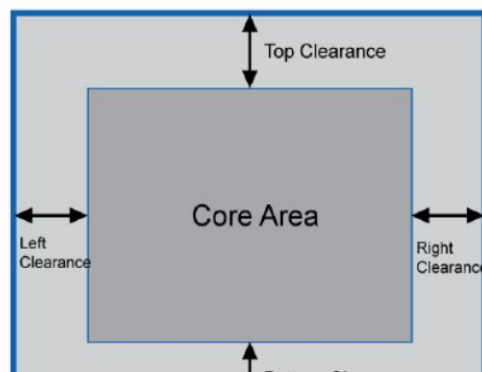
- Slanting lines in the side of the cell rows denote the Cell Orientation



Butt and flip is Most common because of better space utilization

Core to Pad/ IO spacing

- Core to IO clearance
- Used for Placing IOs and Power Ring

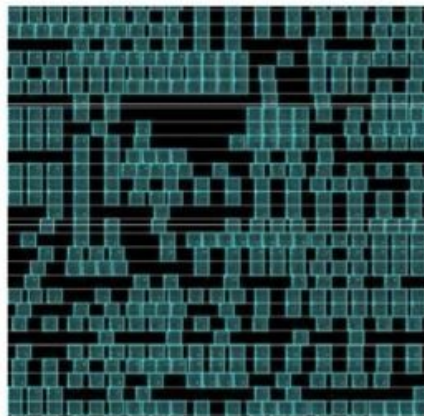




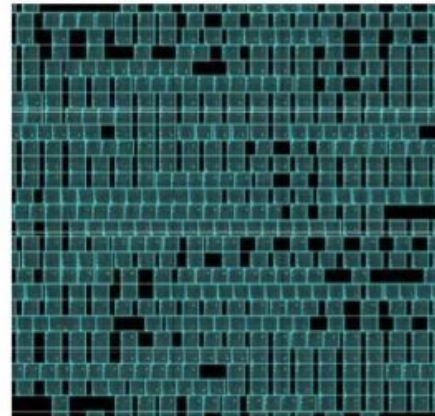
$$\text{Utilization} = \frac{\text{Total Standard Cell Area} + \text{Macro Area}}{\text{Total Core Area}} \times 100 \%$$

$$\text{Aspect Ratio} = \frac{\text{Horizontal Routing Resource}}{\text{Vertical Routing Resource}}$$

- or simply Height/Width
- Aspect Ratio decides the shape
- Full chip Aspect Ratio can have a maximum value of 1.25



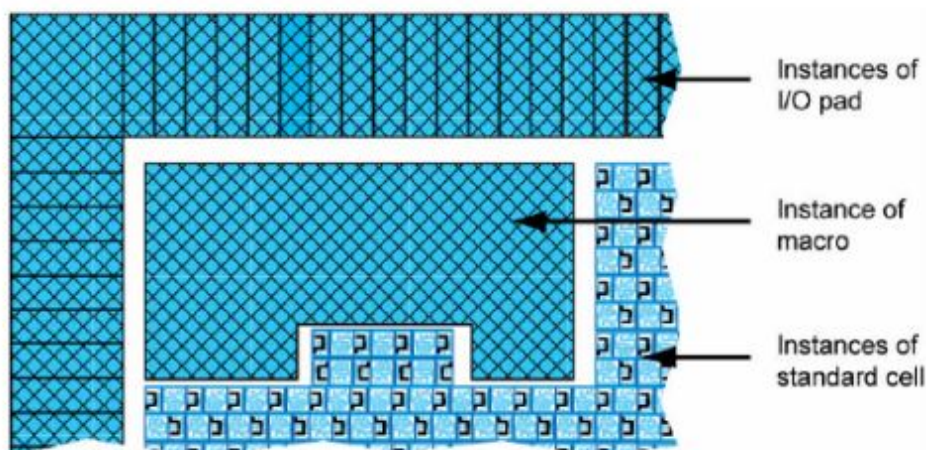
Low standard-cell utilization



High standard-cell utilization

IO Placement

- Chip Level its IO Pads and Block Level its IO Pins
- Pin is a logical entity and is a property of a Port
- Port is a physical entity and a Port have only 1 Pin associated with it
- Netlist will have Pins and Layout will have Ports
- Unplaced Port is not represented in the Layout



- Different types of IOs
- Signal Pads/Pins
- Core Power Pads/Pins
- IO Power Pads/Pins

- IO Power Pads
- Corner Pads (Doesn't hold any logic, provides IO Pad Ring connectivity)
- Filler Pads (Fill the gaps between IO pads to get the Ring Connectivity)
- Physical-only pads that are not part of the input Gate level Netlist need to be inserted prior to reading IO constraints
- IO Pads enables the design to operate at different voltages with the help of Level Shifters, Pre-Drivers (at Core Voltage) Post-Drivers (at IO Voltage)
- No of Core Power Pads needed:

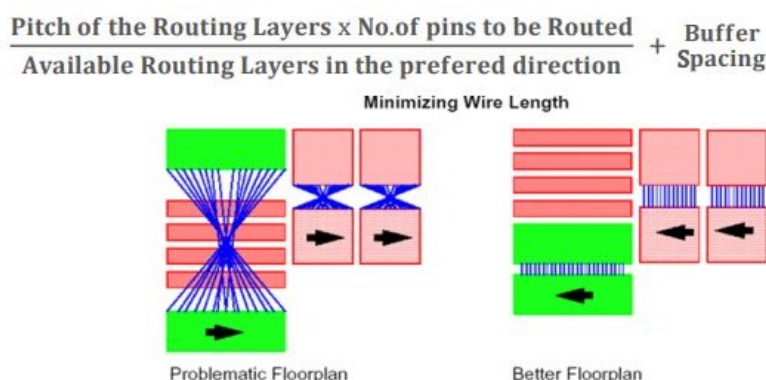
$$\frac{\text{Total Dynamic Power}}{\text{No. of Sides} \times \text{Core Voltage} \times \text{Maximum allowable Current of IO Pad}}$$

- There will be 1 Core GND Pad along with every Core Power Pad
- No. of IO Power Pads needed:
Thumb Rule: 1 pair of IO power pads for every 4 to 6 signal pads.

Macro Placement

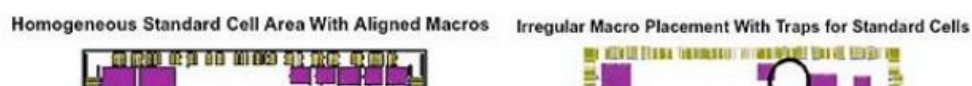
macro placement is done based on connectivity information of macro to IO cell and macro to macro. macro placement is very critical for congestion and timing. macro placement should result in uniform std cell area.

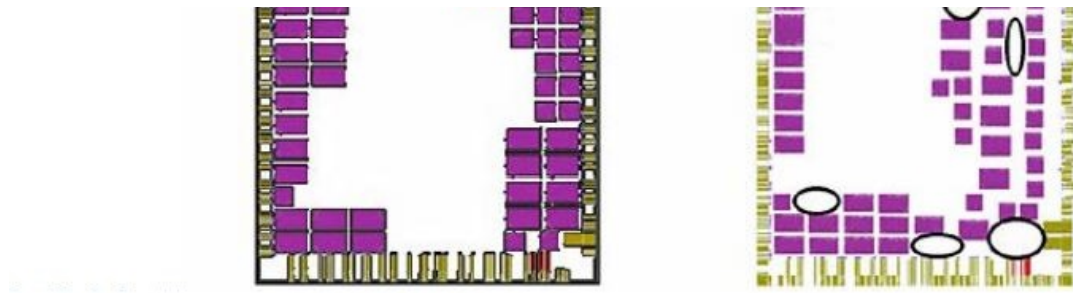
- Fly-line Analysis (For Connectivity information)
- Macro keep-out (For Uniform Standard Cell Region)
- Channel Calculation (Critical for Congestion and Timing)
- Avoid odd shaped area for Standard Cells
- Funnel shaped Macro Placements are preferred
- Fix the Macro locations, so that tool wont alter during Optimization
- Spacing between Macro:



Macro Placement Tips

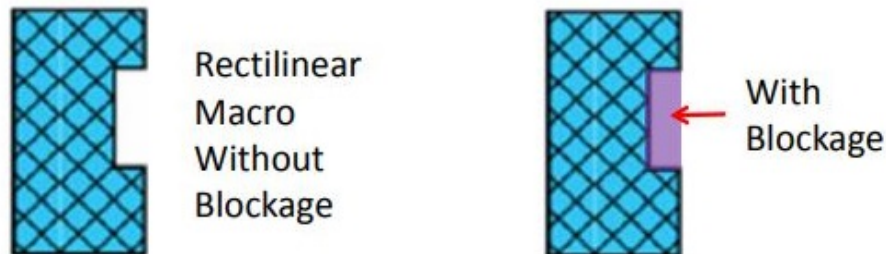
- Place macros around chip periphery, so that core area will be clustered
- Consider connections to fixed cells when placing Macros
- In advanced Technology Nodes Macro Orientation is fixed since the Poly Orientation can't vary, so there will be restrictions in Macro Orientation
- Reserve enough room around Macros for IO Routing
- Reduce open fields as much as possible
- Provide necessary Blockages around the Macro





Blockages

- blockages are specific location where placing of cells are blocked.
- blockages acts like guidelines for placement of std cells.
- blockages will not be guiding the tool to place the std cells at some particular area, but it won't allow the tool to place the std cell in the blocked area.



Placement Blockage & Routing Blockage

- Both of the Blockages can again be classified as-
 - Hard, Soft and Partial Blockages
- Hard Blockage
 - Complete Standard Cell Blockage
 - std cell blockages are mostly used to
 - avoid routing congestion at macro corners
 - Restrict std cells to certain regions in the design
 - control power rail generation at macro cells
- Soft Blockage
 - Non-Buffering Blockage, only buffers can be placed and std cells cannot be placed
- Partial Blockage
 - Partial Standard Cell Blockage and is used to avoid congestion
 - We can Block Standard Cells as per the required percentage value

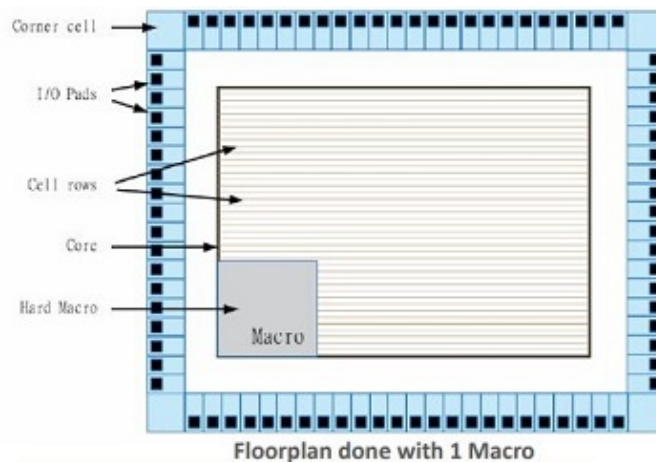
Keep-out / Halo



- Halo is similar to Soft Blockage meaning it allows placement of buffers and inverters in its area. (Terminology in Cadence EDI)
- Its basically a keep-out Macro margin
- Halo respects Macro while other Blockages respect location
i.e., even if Macro is moved Halo also moves along with it
- Halos of adjacent macros can overlap.

Issues arises due to bad Floorplan

- Congestion near Macro Pins/ Corners due to insufficient Placement Blockage
- Std. Cell placement in narrow channels led to Congestion
- Macros of same partition which are placed far apart can cause Timing Violation



Floorplan Qualification:

- No I/O ports short
- All I/O ports should be placed in routing grid
- All macros in placement grid
- No macros overlapping
- Check PG connections (For macros & pre-placed cells only)
- All the macros should be placed at the boundary
- There should not be any notches. If unavoidable, proper blockages has to be added
- Remove all unnecessary placement blockages & routing blockages (which might be put during floor-plan & pre-placing)

Floorplan outputs

- IO ports placed
- cell rows created
- macro placement final
- core boundary and area
- pin position
- floorplan def

