Chapter 1

A Quick Tourfr om Gates to Layout

If you have no prior knowledge of Place and Route (P&R), this chapter is designed to bootstrap you into the design of digital layout. Later in the book, you will be able to master the rest of the materials. This chapter covers the essential steps of a Gate to Layout flow. The details of the flow will be addressed in subsequent chapters. If you are experienced in P&R, you still might want to browse through this chapter to get a feel for the technical aspects covered in the other chapters.

The basic steps of a Gate to Layout flow include the following:

Step 1	**Load library**
Step 2	**Import gate netlist**
Step 3	**Specify design constraints**
Step 4	**Floor planning**
Step 5	**Power planning**
Step 6	**Physical synthesis**
Step 7	**Clock tree synthesis**
Step 8	**Routing**
Step 9	**Physical verification**
Step 10	**Post-layout verification**

Most P&R projects have to move through these steps. However, the steps are not necessarily executed in the order listed. Below are examples of the variation that may occur in the execution of P&R:

- Physical verification (step 9) of the pad ring during floor planning (step 4).
- Physical synthesis (step 6) to assess the feasibility of a floor plan (step 4) without power planning (step 5).
- Some P&R flow perform physical synthesis (step 6) and clock tree synthesis (step 7) concurrently.

Not all the steps are necessary for a P&R project. Depending on the logic design requirements and fabrication process technology, some steps can be omitted, or some steps can be added. For example,

- Clock tree synthesis might not be necessary at 0.6um process technology.
- Crosstalk noise violation analysis and fixing is a must for layout using 0.18um or smaller process technology.

Step 1: Load Library

A P&R library contains two types of information

- technology library
- cell library

Step1.1 Technology Library

Wires that satisfy all layout design rules must be put in place by the router. The timer engine requires accurate parasitic capacitances and resistances for static timing analysis, crosstalk analysis and power analysis. Information regarding the layout design rules and capacitance

look-up table are nested in the technology library. The following table lists some of the P&R tasks and their corresponding roles from the technology library that are necessary to perform the tasks.

P&R Task	Technology library
Congestion driven P&R	Process design rules.
Timing driven P&R	Routing Parasitics.
Cross-talk aware P&R	Routing parasitics with coupling capacitance.
Electro-migration (EM) sign-off	Electro-migration limit for each metal layer.
Metal and via density filling	Metal and via density requirement

Step 1.2 Cell Library

The cell library holds both logical and physical information of the logic cell. The logical information is similar to the contents of a synthesis library. Several different types of information exist and cannot all be listed here. The following are some of the common contents of the cell library.

- cell type (e.g. combinational, sequential, pad, timing model, etc.)
- Pin-to-Pin delay
- slew for output pin
- capacitive loading on input pin
- leakage and dynamic power consumption
- design rule (e.g. maximum input slew for input pins and maximum load for output pins)
- maximum allowable noise for input pins, and holding resistance for output pins

The full physical layouts of the cells are too complicated to be used in a P&R environment. Hence, the physical information in the cell library contains a simplified version of the layout commonly known as an "abstract". An abstract contains the following information:

- cell name
- size of the cell
- allowable orientation
- pin names and their layout geometries
- routing blockages
- process antenna areas for the pins

Power and ground pins are typically excluded from the logical library but they must be included in the abstract library. The following graphic (Fig. 1.1) depicts an example of a standard cell abstract.

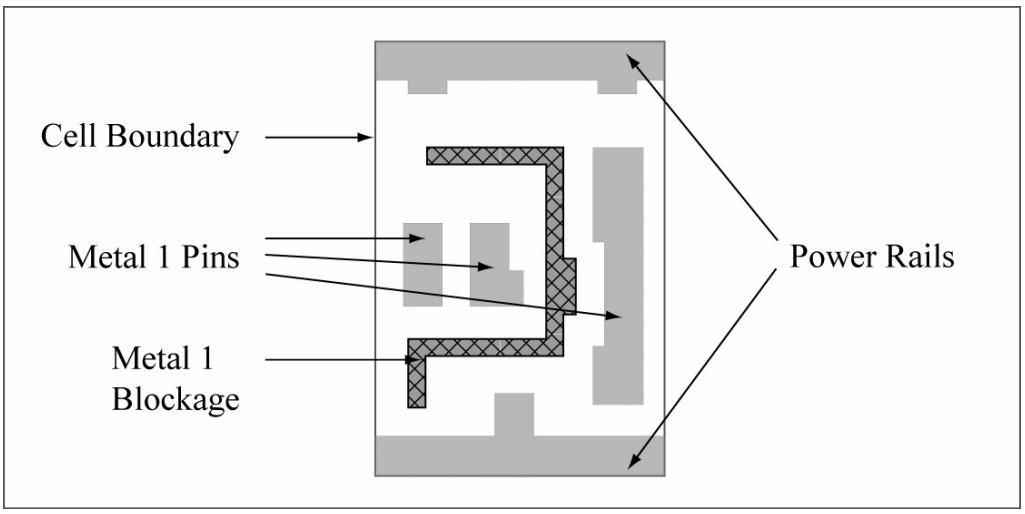


Figure 1.1 : An example of a standard cell abstract.

A popular format for abstract library is Layout Exchange Format (LEF). LEF format supports both the technology library and the cell library. The technology defines the name of the layers (e.g. metal layers and via layers) that are used in the cell libraries, therefore it is mandatory to load the technology library before the cell library.

Step 1.3 Three Types of Cell Libraries

There are three types of cell libraries:

- Standard cell library
- Pad cell library
- Macro library

Standard cells must be placed in the "core" and on the "cell row". "Core" and "Cell row" are described in the latter part of this chapter. Pads and macros do not have this restriction and can be placed anywhere on the layout. However, pads are typically placed on the peripherals of the layout. All P&R tools have specific functions to automate the placement of the pads.

Step 2: Import Gate Netlist

Verilog is the most popular gate-level netlist format. It is also the preferred netlist format for most P&R tools. An alternative is the use of the VHDL gate-level netlist.

After loading the netlist into the P&R tool, the logic gate in the design bind to their cell master in the cell libraries. This process seems trivial, but it is necessary to cater to all situations where cells from different libraries have the same cell name.

Step 2.1 Cell and instance

A cell library is a collection of cells. Standard cell refers to a logic gate. I/O[1] cells are usually called I/O pads. Hard macros refer to the layout of the IP[2]. An IP without a layout implementation is called a soft macro.

An instance refers to a cell in the design. Rather than saying "adding the cell AND2D1 to the design", the common term used is "instantiating the cell AND2D1 to the design". Every instance in the same design hierarchy must have a unique instance name. The following example is an instantiation of a 2-input AND gate:

---

[1] I/O stands for Input/Output.
[2] **I**ntellectual **P**roperty.

```
AND2D1 inst0 ( .A1(net1), .A2(net2), .Z(net3) );
```

The instance name of the 2-input AND gate is “inst0”, and the master name (or cell name) is AND2D1.

Figure 1.2 summaries the definition of cell and instance.

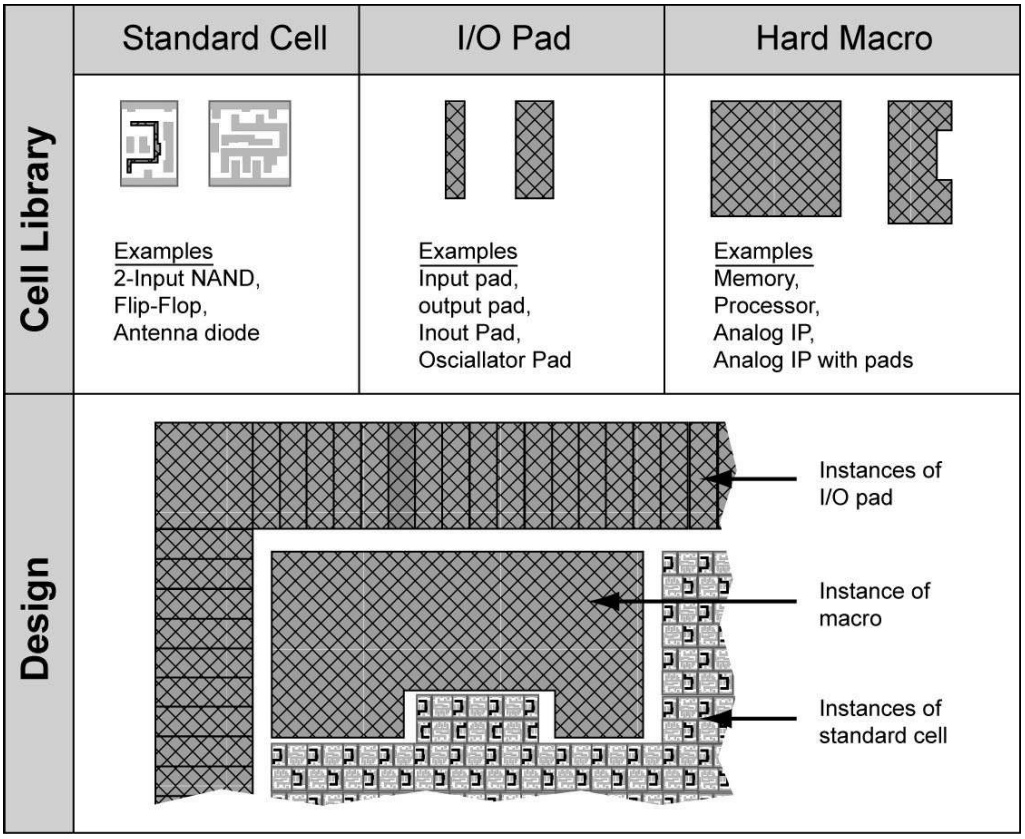


Figure 1.2 : Cell and instance.

Step 2.2 Define power and ground connections

Refer to the instantiation of a 2-input AND gate.

```
AN2D1 inst0 ( .A1(net1), .A2(net2), .Z(net3) );
```

Gate level netlist does not typically include the connection to the power and the ground supplies. However, these connections have to be defined

before layout implementation. All P&R tools use global nets[1] and wildcards to define these connections. For example, the following Design Exchange Format (DEF) statements create two new nets named VDD and VSS. It also connects all the pins with name "vdd" to net VDD, and all the pins with name "vss" to net VSS. The "*" in the commands is the wildcard that matches to any instance name.

```
-VDD (* vdd)
-VSS (* vss)
```

If VDD and VSS are the only two power nets in the design, and all the power pins of the cell instances are named "vdd" and "vss", then these two commands are sufficient to define all the power supply connections.

Tie-high and tie-low refer to the connection of the input pins to the power supply. Tie-high and tie-low are represented as 1'b1 and 1'b0 in the Verilog netlist. There are two ways to physically connect tie-high and tie-low nets. These nets can either connect directly to the power and the ground nets, or connect using tie-high and tie-low cells. Whichever the case, it is the design of the pad library and the standard cell library that determines the appropriate type of connection.

Insertion of Tie Cell into the design

Can you suggest the advantages of inserting the tie cells into the design after placing the instances, rather than inserting them into the design during logic synthesis?

Step 2.3 Instance name and hierarchical instance name

A design can contain many hierarchical levels. The design name is the same as the top-level design hierarchy name. An instantiation can be a cell or a sub-hierarchy.

[1] The net is global because it can be reference directly by its name at any design hierarchy without the use of the full hierarchical name.

A netlist can instantiate the same sub-hierarchy several times. However, in order to allow the layout of the sub-hierarchy to be implemented differently for each instance, the sub-hierarchy must be "uniquified" by duplicating the sub-hierarchy with different hierarchy cell names. Figure 1.3 illustrates the "uniquification" process. Note that instance names are preserved during uniquification.

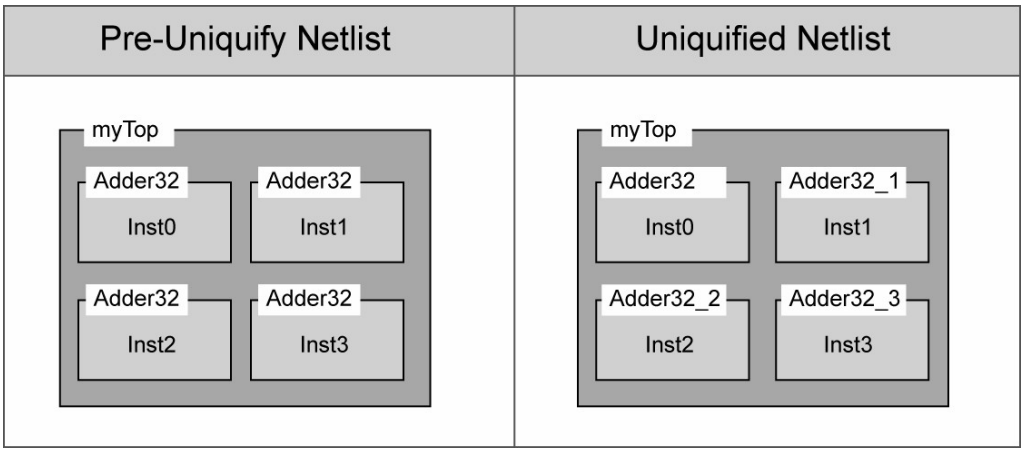


Figure 1.3 : Uniquify the netlist.

A full hierarchical name is required to refer to a particular instance. Refer again to Figure 1.3. Assume there is a flip-flop with instance name "U1" in "Adder32". As "Adder32" is instantiated four times before uniquification, we can refer to that flip-flop in the four instances of "Adder32" as Inst0/U1, Inst1/U1, Inst2/U1 and Inst3/U1. As instance names are preserved during uniquification, the instance names of the four flip-flops remain unchanged after uniquification.

Step 3: Specify Design Constraint

P&R is a constraint-driven process. In the absence of constraints, P&R optimization is purely congestion-driven. A timing constraint is an important part of the design constraint. Timing constraints specify the timing goals of the design. In order to perform timing-driven placement or physical synthesis, timing constraints must be available. Timing constraints are most likely to be specified in SDC format.

The timing constraints should be specified at the top-level of the design. It is necessary to specify a complete top-level timing constraint for a timing-driven flow. Any unconstrained timing paths will not be optimized for timing performance. Hence, unconstrained paths might be implemented with logics that are very slow or with extraordinarily large slew.

On the other hand, over-constraining the timing requirement is undesirable. An over-constrained design can result in an implementation that is unnecessarily large in area and the P&R will possibly consume a much longer run-time.

In addition to timing constraints, there are constraints not related to timing performance. There is no standardized name for this type of constraint, so it will be termed a "non-timing constraints" in this book.

There are many different types of non-timing constraints. The following list describes some of them.

- design rules which include maximum fan-out, maximum slew and maximum capacitance loading
- scan-chain re-ordering and re-partitioning
- selective hierarchy flattening
- buffering of inputs and outputs with user-specified cells
- identification of cells that the tool cannot modify or can only resize
- identification of nets that must be preserved during logic optimization
- disallow the use of certain cells
- assign higher priority to certain nets so as to achieve shorter wiring length
- restriction in the area that certain cells can be placed
- among others

Non-timing constraints can be employed to ensure the physical implementation meets the design requirements, improvement of layout quality and turn-around time, as well as to work-around the limitations of the P&R tools.

Step 4: Floor Planning

Floor planning is the first step of physical layout implementation. A floor plan should include the following decisions:

- size of the layout
- core area
- placement of I/O pads and I/O pins
- placement of the hard macros

A floor plan should include the placement of all the pads (or pins) and the hard macros. However, the standard cells are not placed yet and no routing is performed at this stage[1].

Step 4.1 Size of the layout

The first step in floor planning is to define the outline of the layout. If the layout is rectangular, only the length and the width of the layout are required. More co-ordinates are needed to define the outline of a rectilinear layout, such as an L-shape layout. Most of the P&R tools do not alter the size of the layout specified by the user.

Step 4.2 Core Area

The core area is usually defined by specifying the distance between the edge of the layout and the core, as shown in Figure 1.4.

[1] It is possible to pre-place some standard cells, and also to pre-route wires at this stage of the flow. It is also possible that smaller macros are left unplaced.

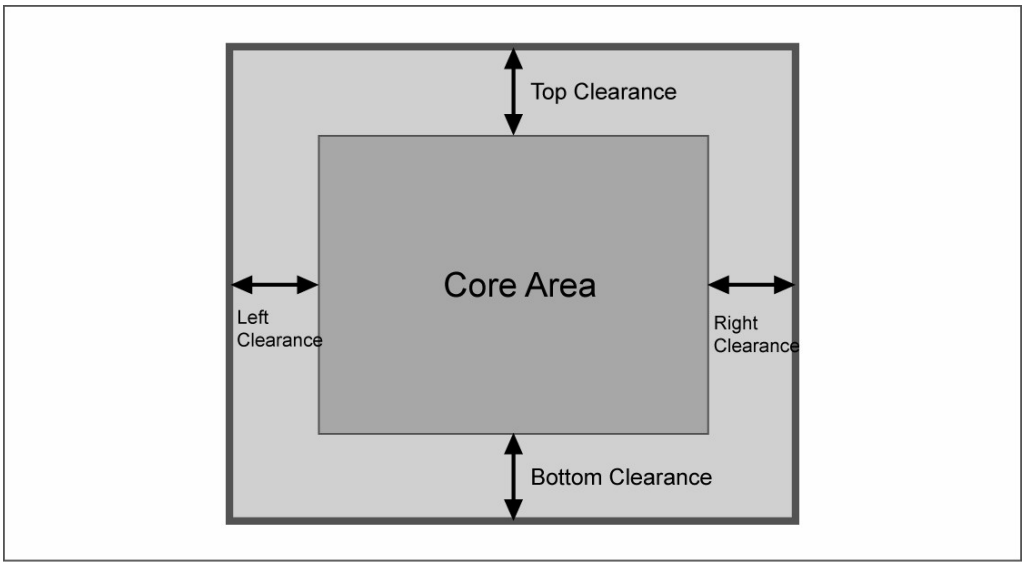


Figure 1.4 : Definition of the core area.

All standard cells must be placed in the core area. I/O pads and macros do not have this restriction although it is common to place macros in the core area. The area outside the core can be used to place the I/O pads, the I/O pins, and the core power rings.

Standard cells are placed in rows, similar to placing books on a book shelf. The rows are called “cell rows” and are drawn inside the core area. All cell rows have the same height. There are three common ways to arrange the cell rows (Figure 1.5).

The most common approach for layout with more than three metal layers is to flip every other cell row which does not leave a gap between the cell rows.

The second configuration is to flip every other cell row, but leave a gap between every two cell rows. The purpose of the gaps is to allocate more resources for the inter-connect routings.

The last configuration is to leave a gap between every cell row, and not flip the cell rows. This configuration is useful when only two or three metal layers are available for routing.

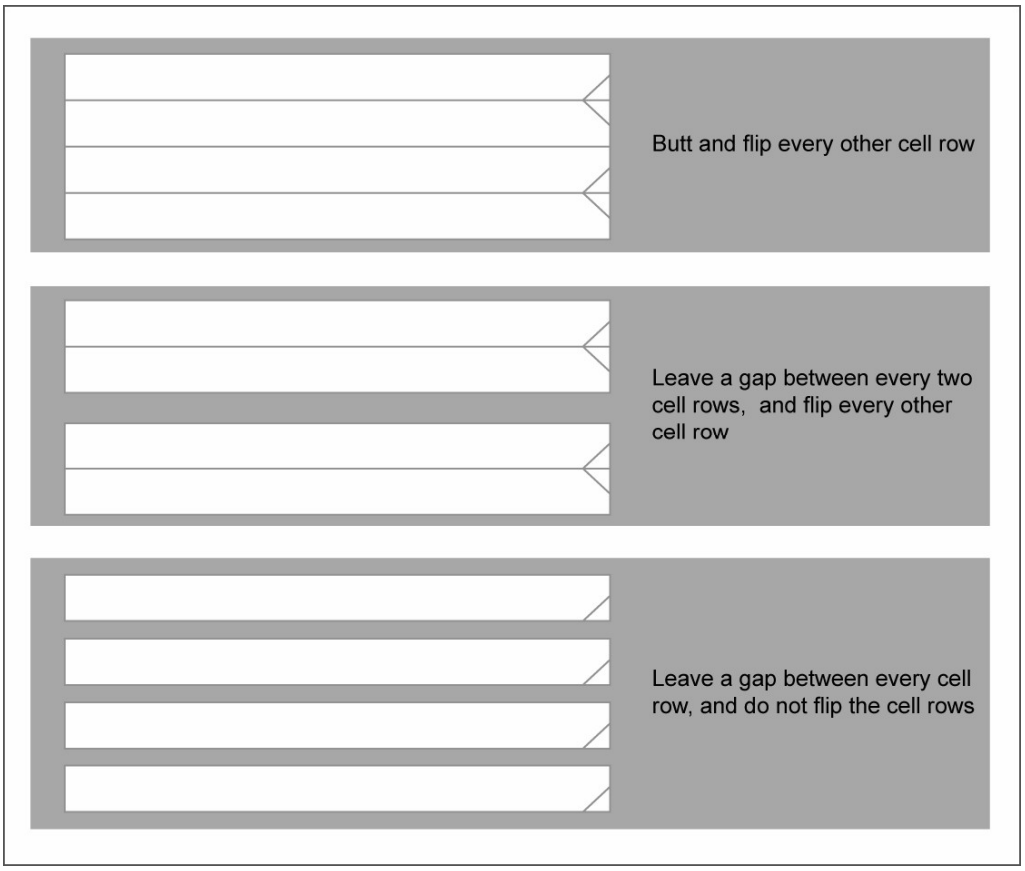


Figure 1.5 : Three types of cell row configuration.

The "slanting lines" on the right of the cell rows in Figure 1.5 denote the orientation of the cell rows. Modern P&R tools will fill the core area with cell rows automatically. Some P&R tools require the user to specify the areas in the core where the cell-row should be created.

Arrangement of the cell rows

What is the reason for flipping the cell rows?

When you need to leave a gap in between the cell rows, how do you determine the height of the gap?

If all the standard cells are in the same power domain, then only one core area is required. In a multiple core power P&R flow, more than one core area must be defined, and every core area must associate itself with a power domain.

Step 4.3 Placements of IO Pads and IO Pins Geometries

For a chip-level layout, the next step is to place the IO pads. The P&R tool can fill the gaps between the pads with pad filler cells and corner cells. For a block-level layout, the user needs to define the location and geometries (size and metal layer) of every IO pin.

Step 4.4 Placements of the Hard Macros

The floor plan is complete if the design does not contain any hard macro[1]. Otherwise, the next step is to place the hard macros. Placing the hard macros may not be a simple task. A good macro placement has the following qualities:

- provides a compact layout
- does not cause routing congestion
- does not make timing closure difficult
- allows robust power routing between the power pads and the macros

The biggest challenge in placing the macros is in assessing the quality of the floor plan, which cannot be achieved without executing the rest of the flow. Thus, floor planning is an iterative and time consuming process. The trick in performing floor planning is to shorten the turn-around time of the iterations, and to reduce the number of iterations. The following figure (Figure 1.6) depicts a simple floor plan of a chip-level layout with only one macro.

[1] Examples of hard macro are memory, analog layout or a complete layout of a CPU. Basically, any instant that is not a standard cell is a hard macro!

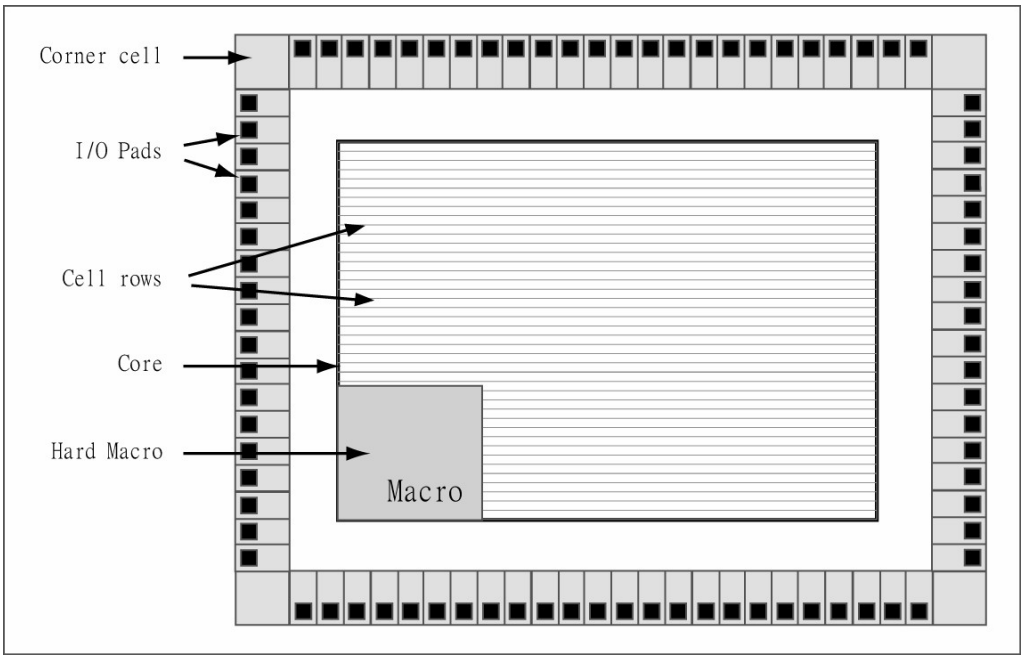


Figure 1.6 : A floor plan with one macro and I/O pads

Step 5: Power Planning

All connections to the power and ground nets are routed during power planning. The only exception is the tie-high and the tie-low nets. Most P&R tools use a dedicated router to route the power nets. All power routings created by the power router are considered pre-routes, and are not modified by the detailed router when the signal nets are routed.

The key consideration for power planning is:

- an acceptable IR-drop from the power pads to all power pins
- meeting electro-migration requirements
- does not result in routing congestion
- compact layout

A power plan consists of several types of power structure. Figure 1.7 illustrates a typical sequence to construct the power structures.

1. core power rings are routed first
2. core power pads are connected to the core power rings

3. the power rings are added around the macros where necessary
4. vertical stripes and horizontal stripes are added to reduce the IR-drop at the power rails of the standard cells and the macros
5. the power pins of the hard macros are tapped to the core rings or the power stripes
6. if tie-high and tie-low cells are not used, the tie-high and tie-low inputs to the hard macros and IO pads are tapped to the power structures
7. the power rails for the standard cell are added to the power plan

The power rails can tap the power from the core power rings, the power stripes and the macro power rings.

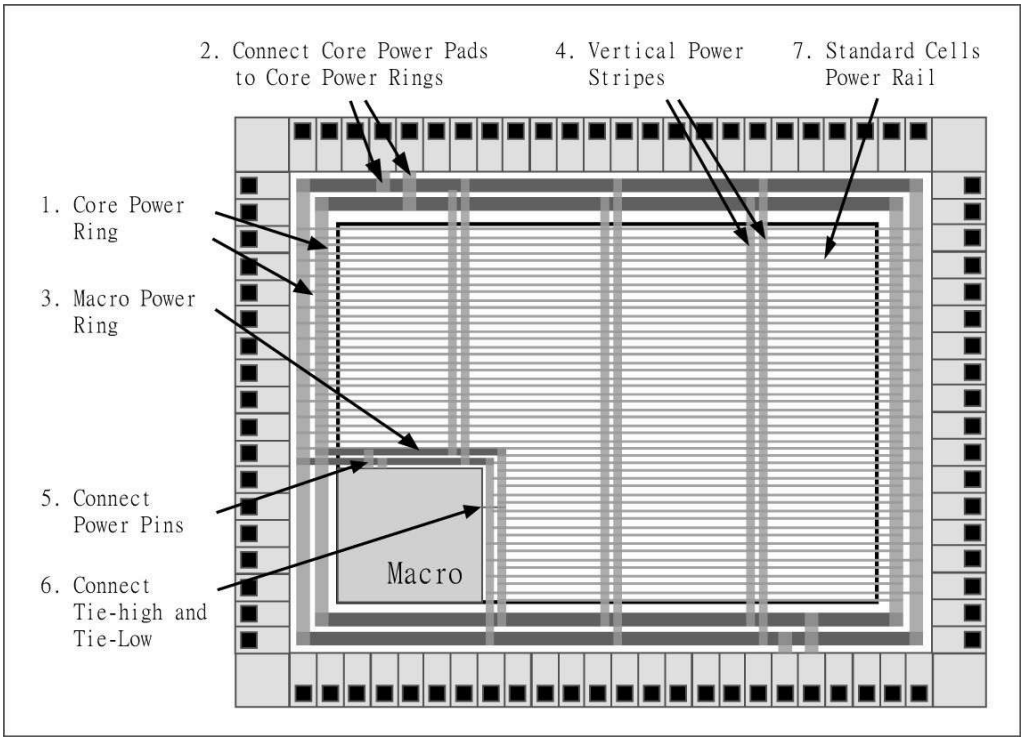


Figure 1.7 : Steps in building a power plan.

Priorities of the power structures

In the 7 steps illustrated above to build the power structures, can you suggest a reason why "step 2" should not be executed last?

If you are to use both vertical and horizontal stripes, what are the considerations to decide which one should be added to the power plan first?

Step 6: Physical Synthesis

Physical synthesis refers to the placement of the standard cells and the optimization of the layout base on the design constraints.

Physical synthesis consists of a few typical phases:

- global placement
- global routing
- physical optimization
- detailed placement
- further physical optimization

After physical synthesis, all standard cells are placed on the cell rows. The placement should be legalized, which means that the standard cells are on the cell row, on the placement grid, non-overlapping, and the power pins of the standard cells are properly connected. The placement should be routable, meeting the timing goal, and satisfy the placement constraints specified by the user.

In order to meet the timing goal, the tool might need to optimize the netlist. Different tools have different capabilities given the type of optimization it can perform. The user can also configure the type of optimization the tool can utilize. Some of optimization techniques a P&R tool can employ are listed below. The optimization techniques are

listed in an increasing order of structural change relative to the original netlist.

- gate sizing
- buffer insertion and removal
- pin swapping
- cloning
- logic restructuring
- architecture retargeting

Physical synthesis becomes essential when the IC industry started to adopt process technologies that are 0.25um and smaller. The following table summaries the evolution of the placement methodology.

Process Technology	**Physical Optimization Techniques**
0.6um and larger	Placement is congestion driven. Manual insertion of buffers to long nets after routing.
0.35um	Placement is timing driven. Physical optimization is restricted to gate sizing and buffer insertion.
0.25um and smaller	Physical synthesis is fully adopted.

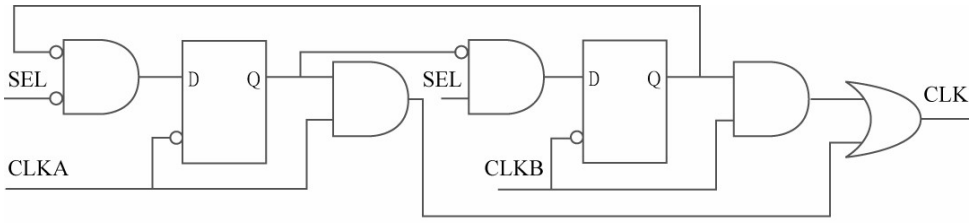
Table 1.1: Evolution of placement methodology with process technology.

Most P&R flows will not attempt to restructure the logic in the clock network. Some P&R tools have the ability to size the cells in the clock network during physical synthesis.

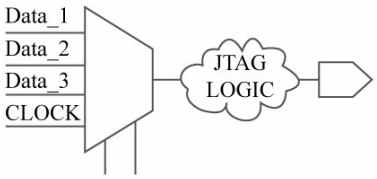
Mixing clock and data

A clock network can contain combinatorial logic cells. Clock signals can also be used as data signals. The first example below is a glitch-less multiplexer where the output clock is one of the two input clocks. This example illustrates that there can be combinatorial gates in the clock network.

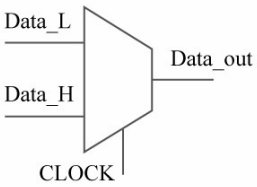
The second example shows multiplexing of a clock signal with data signals. The third example shows the uses of the clock to select the output of a multiplexer.



Example 1 : Glitchless clock multiplexer



Example 2 : Multiplex Clock with Data



Example 3 : Doubling data rate

As physical synthesis preserves the clock network, the combinatorial gates in the clock network may not be sized or buffered appropriately. Thus, a huge negative slack might be expected after physical synthesis if no additional steps are taken to avoid these issues.

To achieve a good estimation of the inter-connect parasitics, global routing is performed during physical synthesis. It is assumed that detailed routing will match global routing closely so that physical synthesis is optimizing on the real critical paths.

The routing congestion map can be derived from global routing. Any routing congestion at this stage should be resolved as much as possible by reiterating the placement with additional controls, or by improving the floor plan and power plan.

Before proceeding further to the layout design, the layout is now ready to perform IR-drop analysis. Two main objectives of IR-drop analysis are to ensure

- all power pins of the hard macros and the standard cells are connected to the power structures
- the voltage drops in the power structures are within acceptable limit.

Step 7: Clock Tree Synthesis

After all the standard cells are placed, the clock nets are buffered. The following list provides the additional requirements for synthesizing clock trees when compared to the buffering of the high fan-out non-clock nets:

- clock latency
- clock skew
- restriction on the type of buffer and inverter the clock tree can use
- stricter signal slew requirements on the clock nets

Clock latency is the delay of the clock signal from the clock source to the clock pin. Clock skew is the difference between the clock latencies and the two clock pins.

It is straight forward to specify the clock tree requirements to the P&R tool. If the clock tree starts from one source and fans-out only to the clock pins of the flip-flops and the macros, the clock tree meets the requirements of the P&R tool. Unfortunately, this is not always the case. For example, the clock tree schematic shown below has the following additional requirements:

- the clock latencies of flip-flops div_reg* do not have to be balanced with the clock latencies of the other flip-flops
- a small clock skews between the flip-flops div_reg*
- a small clock skews for the rest of the flip-flops in both functional mode and test mode
- optimize for shorter clock latencies during functional mode

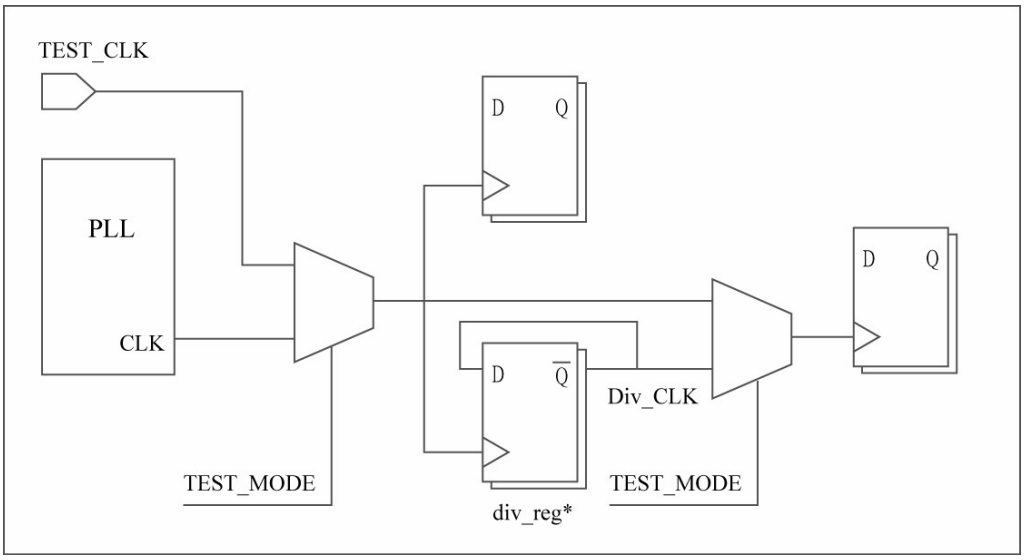


Figure 1.8 : Clock tree schematics example

Strategy for clock tree synthesis

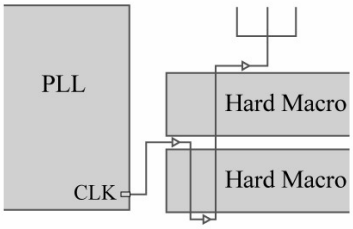
The strategy to synthesize the clock tree is highly dependent on the features of the P&R tool you are using.

As a P&R engineers, how do you find out all the requirements of the clock tree?

It is not uncommon for the clock tree synthesis algorithm to generate a poor clock tree when complicated blockages exist in the layout. The following is an example of a poor clock tree:

Poor Clock Tree

Here is an example where the algorithm for the clock tree synthesis cannot efficiently bring the clock signal from the clock source to the clock pins. This is due to the presence of many large blockages. The clock latency is unnecessarily long. Poor slews in the clock nets resulted in a large clock skew.



Can you suggest any approaches to alleviate this problem?

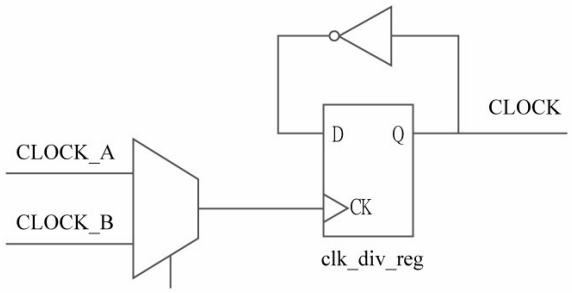
Additional routing requirements are often applied on the clock nets. In order to reduce noise coupling, the route spacing to the clock net can be doubled. Shielding the clock net is another option to reduce noise coupling. For a clock with very high clock frequency, it may be necessary to use multiple-via on the clock routing to meet the electro-migration rules.

Before clock trees are inserted, the tool uses ideal clock latencies and ideal clock skews for timing analysis. After the clock trees are synthesized, the tool should use the actual clock latencies and clock skews for the timing analysis.

Ideal Clock Latency

The default ideal clock latency is 0. How about the ideal clock latency of a generated clock? The divided clock "CLOCK" in the schematic below has the following specification:

```
create_generated_clock -name CLOCK \
  -divide_by 2 \
  -source [get_pins clk_div_reg/CK] \
  [get_pins clk_div_reg/Q]
```



Generated clock with multiple sources

What should be the clock latency of "CLOCK"? Also, what is the frequency of the "CLOCK"; should it be half the frequency of "CLOCK_A" or halve the frequency of "CLOCK_B"?

It is now possible to analyze and fix hold violations using computed clock latencies associated with every clock end-point after clock tree synthesis. Hold violations should be fixed first in "best corner" operating condition, and then in the "worst corner" operating condition. Current P&R tools fix hold violations by adding delay to the data path. The tool will not attempt to make changes to the clock path. It is advisable to analyze the buffers added by the tool for hold fixing. If the result is not satisfactory, clock trees might need to be re-synthesized using different approaches.

Timing nightmare after clock tree synthesis!

Imagine this scenario. The timing requirement of a design is met after the physical synthesis step. Clock tree synthesis is then performed and all the clock trees meet the skew and latency specifications. However, static timing analysis (STA) shows that there are many timing paths with very poor timing slack.

Can you suggest a few reasons for the poor timing slack?

Step 8: Routing

The process of routing the non-power signals is called "detailed routing". After clock tree synthesis, empty space still exists in the cell rows. These empty spaces in between the standard cells will eventually be filled by filler cells. Fillers can be inserted either before or after detailed routing. If the fillers contain metal routing other than the power rail, then the fillers should be inserted before routing. Otherwise, it is best to add the fillers after routing. Figure 1.9 shows the results before and after filler cell insertion.

Filler cells for a standard cell library

Filler cells come in different widths. The width of the smallest filler cell is the size of the placement grid. The widths of all the standard cells are multiples of the placement grid size.

Do you know the purposes of the filler cells?

The filler cells usually have widths that are given as 1x, 2x, 4x, 8x, 16x, 64x, etc. The next bigger filler cell always has its width doubled. Do you know why?

Why is it better to insert the filler cell after detailed routing?

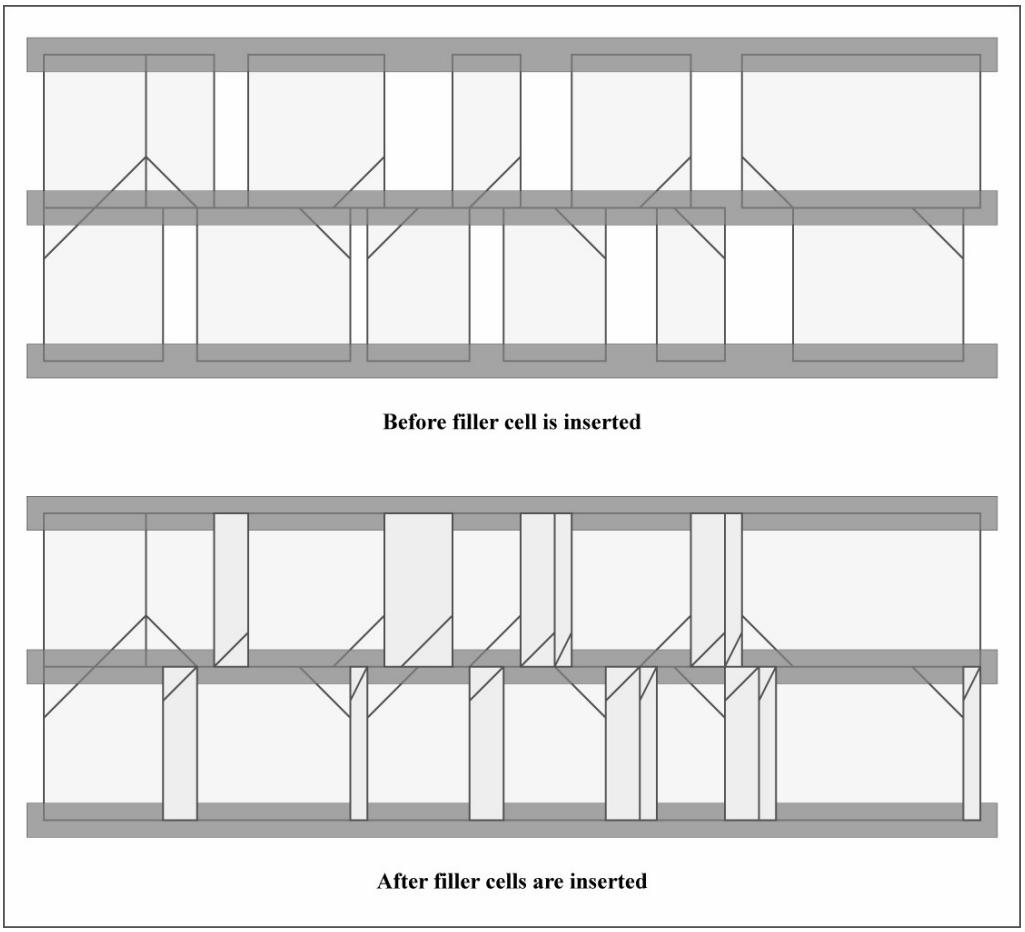


Figure 1.9: Inserting the filler cells

Routing is performed throughout P&R flow. Powers are routed during power planning by a dedicated router. During physical synthesis, the signal nets are “globally routed” using the global router. The routings are called global routes. Global routes allow the P&R tool to resolve routing congestion and estimate the routing parasitics. After the clock trees are synthesized, the clock trunks are routed with actual metal geometries. This occurs before the clock trees are re-optimized. This is then followed by detailed routing.

Detailed routing is carried out in stages. The detailed routing stages are different in various P&R tools, but the methodology used for detailed routing is similar. The stages are outlined below.

1. Track routing: Global routing uses a very coarse routing grid. Track routing assigns the global routes to the actual routing tracks.

2. Detailed routing with only metal one: Connections between cells that are placed side-by-side are possible candidates.
3. Connecting the rest of the signal nets by following the result of track routing: The aim is to connect all the routings so that there are none "Open". The routings can be full of routing violations (e.g. short) and design rule violations (e.g. metal to metal spacing violations).
4. Resolve routing violations iteratively: The detailed router divides the layout into regions and works within each region to clean up the routing violations. This process iterates with a region of different sizes and aspect ratios. The iterative process continues until there are no more routing violations or the limit on the number of iterations has been reached.
5. Iterating between fixing antenna violations and cleaning up routing violations: New routing violations can be introduced during the process of fixing the antenna violations.
6. Optimizing the detailed routes: The types of optimization to be performed will depend on the user's specifications and the tool's capability. The optimization can include minimizing wire jog and switching of routing layers, or even out the spacing between the routes, and replacing single via with redundant via.

All P&R tools have the functionality to perform cell-level DRC and LVS on a routed design. However, it is mandatory to perform a full-layer (or sometimes called full-GDS) physical verification.

How about the flows for cross-talk reliability, electro-migration, power sign-off and yield enhancement?

These topics are beyond the scope of this chapter. The methodology to address each of these concepts is rooted in the complete P&R flow. They will be discussed in the later chapters.