Chapter 2

# FLOORPLANNING

*"Design is neither a form alone or function alone, it is aesthetic synthesis of the both." Ferdinand Porsche*

Floorplanning is the art of any physical design. A well thought-out floorplan leads to an ASIC design with higher performance and optimum area.

Floorplanning can be challenging in that it deals with the placement of I/O pads and macros as well as power and ground structures. Before one proceeds with physical floorplanning one needs to make sure that the data used during the course of physical design activity is prepared properly. Proper data preparation is essential to all ASIC physical designs in order to implement a correct-by-construction design.

Entire physical design phases may be viewed as transformations of the representation in various steps. In each step, a new representation of an ASIC is created and analyzed. These physical design steps are iteratively improved to meet system requirements. For example, the placement or routing steps are iteratively improved to meet the design timing specifications.

Another challenge commonly faced by physical designers is the occurrence of physical design rule violations during ASIC design verification. If such violations are detected, the physical design steps need to be repeated to

correct the errors. Sometimes these error corrections have a direct impact on the ASIC timing and may require a re-time of the design to meet timing specifications.

Most of the time, the corrections are very time-consuming. Therefore, one of the objectives of physical design is to eliminate or reduce the number of iterations during each step of the design. One of the important keys in reducing the number of iterations is the use of high quality and well-prepared data.

The types of data that are required to start a physical design are:

- Related technology and library files
- Circuit description of the design in the form of netlist representation
- Timing requirements or design constraints
- Floorplan

## 2.1 Technology File

Almost all physical synthesis and place-and-route tools operate based on the technology file. Technology files contain information or commands that are used to configure structures, parameters (such as physical design rules and parasitic extractions), and limits of an ASIC design targeted to specific process technology.

These commands are used at different stages of ASIC design implementation by physical design tools. One of the objectives is to make sure that all parameters in the technology files are set correctly. Once the initial technology file is created, several trial runs need to be performed and results, such as standard cell placement, routing quality, and accuracy of parasitic extraction, should be carefully analyzed. Based upon the final inspection, technology files may require further refinement for optimal performance.

Technology rule basics are as follows:

- Manufacturing grid

- Routing grid
- Standard cell placement tile
- Routing layer definition
- Placement and routing blockage layer definition
- Via definition
- Conducting layer density rule
- Metal layer slotting rule
- Routing layer physical profile
- Antenna definition

Manufacturing grid is determined by the smallest geometry that a semiconductor foundry can process. All drawn geometries during physical design must snap to this manufacturing grid.

Routing grids or tracks are used by physical synthesis and place-and-route tools during detail routing. The routing tracks can be grid-based, gridless-based, or subgrid-based.

Standard cell placement tile is used during the placement phase. The placement tile is defined by one vertical routing track and the standard cell height.

Routing layer definition is used to define the layers that are used to route the design. These definitions include wire width, routing pitch, and preferred routing direction such as vertical, horizontal, or diagonal.

Placement and routing blockage layer definitions are internal to physical design tools and are used to define “keep-out” regions for standard cell placement and routing.

Via definition defines the layer, size, and type for connection between overlapping geometries of conductor for different conductive layers. This cut layer, or via, can be a single via, stacked via, or array of via.

Conducting layer density rule defines the percentage of area of the chip that is required for processes that are using Chemical Mechanical Polishing (CMP) for each physical layer in the design.

The chemical mechanical polishing process requires limited variation in feature density on conducting layers. This dictates that the density of layout

geometries in a given region must be within a certain range. With new silicon processes (i.e. metallization), violation of this rule can have negative impacts on the yield.

Configuration of metal layers slotting rule defines the minimum layer width that may need to have slotting features (i.e. a cut inside a wide routing layer). This rule varies between foundries and is used to limit mechanical stress for a given conducting layer.

Physical profile for each layer is used to define and include conductor thickness, height, and interlayer dielectric thickness. Definition of the electrical interconnect profile includes resistance and dielectric constants.

Antenna definition for each layer configures the physical design tools for automatic antenna repair. Antenna phenomena occur during the metallization process when some wires connected to the polysilicon gates of transistors are left unconnected until the upper conducting layers are deposited. A long wire connected to the gate of MOSFT can act as a capacitor or antenna that collects charges during the plasma-etching step. If this energy build-up on the floating transistor gate is suddenly discharged, the transistor could suffer permanent damage due to gate oxide breakdown. Discussion of how to fix antenna phenomena is presented in Chapter 4.

## 2.2 Circuit Description

In the early days of ASIC design, logic designers used schematic capture, or circuit editing tools, to implement the design. Once the design was captured, the circuit description was imported from capturing tools in some sort of format for physical design. During this time, the rapid emergence of Computer Aided Engineering (CAE) along with Computer Aided Design (CAD) technology led to the development of a broad range of interchangeable data formats and hardware description languages.

Unfortunately, most of these circuit descriptions were limited to specific companies and data types. ASIC and physical designers who wished to use a combination of different CAD tools were forced to convert among various

formats in order to complete the design. This cumbersome and time-consuming translation process drove the need for a standard electronic design interchange format.

To address this issue, the first electronic industry standard format (Electronic Design Interchange Format, or EDIF) was introduced. EDIF is very rich in format and is capable of representing connectivity information, schematic drawings, technology and design rules, and Multi Chip Module (MCM) descriptions, as well as allowing transfer of documentation associated with physical layouts.

During the same time, two major formats were introduced–Very High Speed Integrated Circuit (VHSIC) or Hardware Description Language VHDL and the Verilog description language (commonly referred to as Verilog).

From an ASIC design perspective, VHDL is not a circuit design tool but merely creates an accurate model of circuit designs. This language, however, has been a phenomenal success because it has the utility and ability to meet circuit and system design requirements. VHDL is independent of design tools and is a powerful language for the modeling of hardware timing.

With the introduction of the first logic synthesis tools, the Verilog model that represented the functionality of the circuit designs could be synthesized. This was a major event, as the top-down design methodology could now be used effectively. The design could be modeled down at the Register Transfer Level (RTL) and could then be translated into the gate using synthesis tools. With this event, the use of Verilog modeling increased dramatically.

Use of Verilog simulation for sign-off certification by ASIC designers was the next major trend to emerge. As Verilog became popular with semiconductor vendors' customers, they began to move away from their own proprietary simulators and to use Verilog simulators for timing certification.

However, Verilog remained a closed language and the pressures of standardization eventually caused the industry to shift to VHDL. Once this was recognized, the Open Verilog International (OVI) committee was formed which brought standardization to Verilog. With standardization, Verilog simulators are now available for most computers, with a variety of performance characteristics and features.

The Verilog language is growing faster than any other hardware description language and is more heavily used than ever. It has truly become the standard hardware description language.

In today's ASIC design flow, the design is generated in RTL format, along with design constraints being synthesized and final results in the form of gate level description, and is imported to the physical synthesis tools for physical design implementation.

Verilog gate-level netlists are widely used owing to their ease of understanding and clear syntax. Although the behavioral Verilog language has a vast number of keywords, there are only a few that may be used in structural Verilog to represent the entire circuit function and connectivity.

A structural Verilog netlist consists of keywords, names, literal comments, and punctuation marks. A Verilog structural netlist is case-sensitive, and all its keywords, such as *module, endmodule, input, output, inout, wire,* and *assign* are lowercase.

The most basic element in Verilog is a *module* definition with corresponding input and output ports [1]. It represents a logical entity that is usually implemented in a piece of hardware. A module can be a simple gate or a complex network of gates. The ports in modules can be a single-bit or multiple bits wide, and each port can be defined as an *input*, *output,* or *inout* (i.e. bidirectional) port. The nets that connect the elements inside a module are described by wire statements.

For improved readability, spaces, tabs, and new lines can be used. A single line comment can start with "//". For multiple line comments, start with "/*" and end with "*/".

A module in the Verilog language starts with the keyword *module* followed by the module name, then the list of inputs and outputs. It ends with the keyword *endmodule*. Each module name must be unique.

Figure 2-1 represents a logical entity implemented using the Verilog circuit modeling style.

```
module bottom_level ( D, E, Y1, Y2, Y3, Y4);
    input      D[3:0], E;
    output     Y1,Y2,Y3;
    inout      Y4;

    assign           D[2] = Y2;
    OR_type          I0 ( .A(D[0]), .B(D[1]), .Z(Y1) );
    AND_type         I2 ( .A(D[2]), .B(Y4), .Z(Y3) );
    TRIBUF_type      I3 (. A(D[3]), .ENB(E), .Z(Y4) );
endmodule

module top_level (IN0,IN1,IN2,IN3,EN,OUT0,OUT1,BIDIR);
    input      IN0,IN1,IN2,IN3,EN;
    output     OUT0, OUT1;
    inout      BIDIR;
    wire       NET1,NET2,NET3,NET4,NET5,NET6;

    bottom_level  bottom_level_instance ( .D{net4,net2,IN1,
          net1}, .Y1(net5), .Y2(), .Y3(net6), .Y4(BIDIR) );
    BUF_type      I0 ( .A(IN0), .Z(net1) );
    BUF_type      I1 ( .A(IN2), .Z(net2) );
    BUF_type      I2 ( .A(EN), .Z(net3) );
    BUF_type      I3 ( .A(IN3), .Z(net4) );
    BUF_type      I4 ( .A(net5), .Z(OUT0) );
    BUF_type      I5 ( .A(net6), .Z(OUT1) );
endmodule
```

**Figure 2-1** A Logical Entity in Structural Verilog Format

The entity that is shown in Figure 2-1 contains simple OR, AND, TRISTATE BUFFER, and BUFFER gates.

The gate level representation or schematic of a structural Verilog netlist, as shown in Figure 2-1, is illustrated in Figure 2-2.

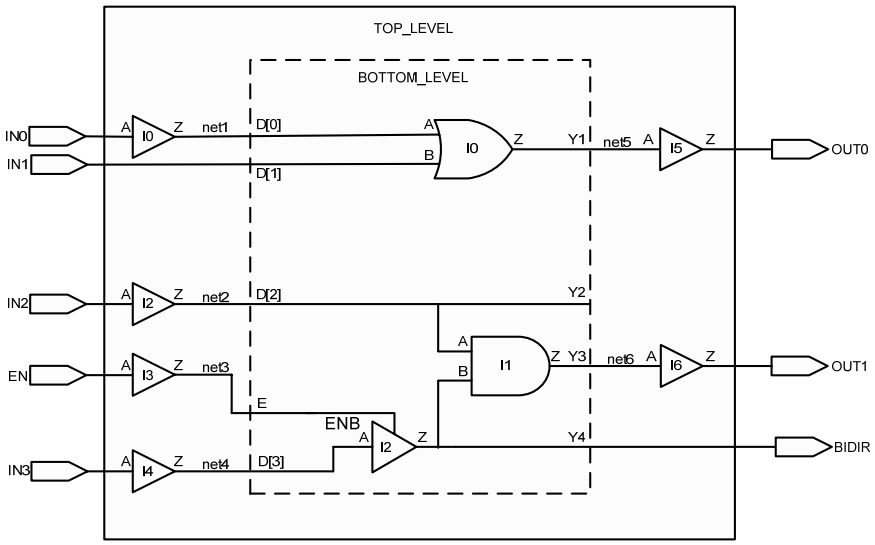


**Figure 2-2** Gate Level Representation

Figure 2-1 and its corresponding schematic presentation of that in Figure 2-2 show there are two module definitions – **bottom_level** and **top_level** and their corresponding input and output ports. The structural module **top_level** has one instance of **bottom_level**. In addition, both **top_level** and **bottom_level** modules contain gate types (e.g. OR_type, AND_type, TRIBUF_type, and BUF_type) that are considered cell types. These cell types are built into a standard cell library and have predefined functions.

The **bottom_level** module has a single continuous assignment statement, indicated by the keyword *assign*. This keyword assigns the value of the right-hand side of the "=" to the value on the left-hand side of the "=".

Although the *assign* statement is considered to be a behavioral statement, it is widely used in today's structural Verilog netlists and presents a physical short between two names. This implies having a wire which has two distinct names attached at each end. When dealing with the *assign* keyword, special care must be taken during the physical ASIC verification to avoid misleading results. One of the most frequently used techniques is to replace the *assign* keyword with a buffer.

After processing the imported netlist, the next step is to apply design constraints as explained in the next section.

## 2.3 Design Constraints

Design constraints are ASIC design specifications that are applied during logic and physical synthesis. Each tool attempts to meet two general design constraints:

- Timing constraints
- Design rule constraints

Timing constraints are user-specified and are related to speed, area, and the power consumption of the ASIC design.

Timing constraints utilized by physical design tools are performance related. The most basic timing constraints are as follows:

- System clock definition and clock delays
- Multiple cycle paths
- Input and output delays
- Minimum and maximum path delays
- Input transition and output load capacitance
- False paths

System clocks, and their delays, are extremely important constraints in ASIC designs. System clocks are typically supplied externally, but can be generated inside an ASIC. All delays, especially in a synchronous ASIC design, are dependent upon the system clocks.

Most logic synthesis tools consider the clock network delays to be an ideal (i.e. a clock with fixed latency and zero skew) and are used during design synthesis. The physical design tools use the system clock definition to perform what is known as Clock Tree Synthesis (CTS) and try to meet the clock networks' delay constraints.

Multiple cycle paths are for ASIC designs that have a non-single cycle clock timing requirement. This directs the physical design tools to avoid optimization of data paths that have non-single clock behavior.

Input and output delays are used to constrain the boundary of external paths in an ASIC design. These constraints specify point-to-point delays from external inputs to the first registers and from registers to the outputs of an ASIC design.

Minimum and maximum path delays provide greater flexibility for physical synthesis tools that have a point-to-point optimization capability. This means that one can specify timing constraints from one specific point (e.g. pin or port) in the ASIC design to another, provided such a path exists between the two specified points.

Input transition and output capacitance loads are used to constrain the input slew rate and output capacitance of an ASIC device input and output pins. These constraints have a direct effect on the final ASIC design timing.

The values of these constraints are set to zero during physical design and place-and-route activity to ensure that the actual ASIC design timing is calculated independent of external conditions and to make sure register-to-register timing is met. Once that is achieved, these external conditions can be applied to the design for input and output timing optimization.

False paths are used to specify point-to-point non-critical timing either internal or external to an ASIC design. Properly identifying these noncritical timing paths has a great impact on physical design tools' performance.

Design rule constraints are imposed upon ASIC designs by requirements specified in a given standard cell library or within physical design tools.

Design rule constraints have precedence over timing constraints because they have to be met in order to realize a functional ASIC design. There are four types of major design rule constraints:

- Maximum number of fan-outs
- Maximum transitions
- Maximum capacitance

- Maximum wire length

Maximum number of fan-outs specify the number of destinations that one cell can connect to for each standard cell in the library. This constraint can also be applied at the ASIC design level during the physical synthesis to control the number of connections one cell can make.

Maximum transition constraint is the maximum allowable input transitions for each individual cell in the standard cell library. Apart from each element in the standard cell library, this constraint can be applied to a specific net or to an entire ASIC design.

Maximum capacitance constraint behaves similarly to maximum transition constraint, but the cost is based on the total capacitance that a particular standard cell can drive any interconnection in the ASIC design. It should be noted that this constraint is fully independent of maximum transition, and therefore, it can be used in conjunction with maximum transition.

Maximum wire length constraint is useful for controlling the length of wire to reduce the possibility of two parallel long wires of the same type. Parallel long wires of the same type may have a negative impact on the noise injection and may cause crosstalk.

These design rule constraints are mainly achieved by properly inserting buffers at various stages of physical design. Thus, it is imperative to control the buffering in an ASIC design during place-and-route to minimize area impact.

## 2.4 Design Planning

Efficient design implementation of any ASIC requires an appropriate style or planning approach that enhances the implementation cycle time and allows the design goals such as area and performance to be met. There are two style alternatives for design implementation of an ASIC–flat or hierarchical. For small to medium ASIC's, flattening the design is most suited; for very large

and/or concurrent ASIC designs, partitioning the design into subdesigns, or hierarchical style, is preferred.

The flat implementation style provides better area usage and requires effort during physical design and timing closure compared to the hierarchical style. The area advantage is mainly due to there being no need to reserve extra space around each subdesign partition for power, ground, and resources for the routing. Timing analysis efficiencies arise from the fact that the entire design can be analyzed at once rather than analyzing each subcircuit separately and then analyzing the assembled design later. The disadvantage of this method is that it requires a large memory space for data and run time increases rapidly with design size.

The hierarchical implementation style is mostly used for very large and/or concurrent ASIC designs where there is a need for a substantial amount of computing capability for data processing. In addition, it is used when subcircuits are designed individually. However, hierarchical design implementation may degrade the performance of the final ASIC. This performance degradation is mainly because the components forming the critical path may reside in different partitions within the design thereby extending the length of the critical path. Therefore, when using a hierarchical design implementation style one needs to assign the critical components to the same partition or generate proper timing constraints in order to keep the critical timing components close to each other and thus minimize the length of the critical path within the ASIC.

In the hierarchical design implementation style, an ASIC design can be partitioned logically or physically.

Logical partitioning takes place in the early stages of ASIC design (i.e. RTL coding). The design is partitioned according to its logical functions, as well as physical constraints, such as interconnectivity to other partitions or subcircuits within the design. In logical partitioning, each partition is place-and- routed separately and is placed as a macro, or block, at the ASIC top level.

Physical partitioning is performed during the physical design activity. Once the entire ASIC design is imported into physical design tools, partitions can be created which combine several subcircuits, or a large circuit can be partitioned into several subcircuits. Most often, these partitions are formed

by recursively partitioning a rectangular area containing the design using vertical or horizontal cut lines.

Physical partitioning is used for minimizing delay (subject to the constraints applied to the cluster or managing circuit complexity) and satisfying timing and other design requirements in a small number of subcircuits. Initially, these partitions have undefined dimensions and fixed area (i.e. the total area of cells or instance added to the partition) with their associated ports, or terminals, assigned to their boundaries such that the connectivity among them is minimized. In order to place these partitions, or blocks, at the chip level, their dimensions as well as their port placement must be defined.

One method that is suggested to estimate the perimeter of a macro instance is to use the number of terminals or ports allowed for each block and their associated spacing between each terminal [2]. The relationship between the perimeter of each partition and the number of associated terminals is given by

$$P = NS, \tag{2.4.1}$$

where $P$ is the perimeter of the physical partition block, $N$ is the number of terminals or ports, and $S$ corresponds to spacing between terminals.

The perimeter estimate given by Equation 2.4.1 determines an appropriate width and height for each partition in the hierarchy, based on the wire demand in both vertical and horizontal directions. However, in order to fit each macro instance at the chip top level in an effective manner, the automatic floorplan algorithm needs to have a range of legal shapes that is derived from aspect ratio bounds for each partition in the design.

The aspect ratio bounds that are generated by the hierarchical floorplan algorithm must have the flexibility to ensure that each macro instance shape can be reshaped for optimum placement. Because one requirement for an ASIC design is to fit into the smallest available die size, during the partitioning process for each partition several layout alternatives are considered by modification of the dimensions and terminal placements along the boundary of each partition in the design such that the amount of unused and routing area between each partition is minimized.

Both flat and hierarchical physical design implementation begin with importing technology files, library files, the netlist, and design constraints into the physical design tools. Once this data is imported, the physical design tool performs binding operations on the entire netlist for flat design implementation or for each sub-netlist for hierarchical design implementation.

During the binding, or linking, process, the incoming netlist is flattened and all entities are analyzed to determine their available models. A variety of checks is performed automatically to determine whether the physical synthesis or place-and-route internal data structure is ready to proceed with the rest of the design implementation flow.

Often the checks that detect problems with the physical database are related to the netlist, such as unconnected ports, mismatched ports, standard cell errors, or errors in the library and technology files. Because of these checks, a log file that contains all errors and warnings will be generated. It is important to review the log file and make sure that all reported errors and warnings are resolved before proceeding to the next phase.

Regardless of the physical design implementation style, after physical database creation using the imported netlist and corresponding library and technology files, the first step is to determine ASIC device and core width and height. In addition, standard cell rows and I/O pad sites are created. The rows and I/O pad sites are for standard cell and I/O pad placement. Figure 2-3 shows an initial ASIC design floorplan.

The height of a row is equal to the height of the standard cells in the library. If there are any multiple-height standard cells in the library, they will occupy multiple rows.

Most of the time, standard rows are created by abutment. The standard rows are oriented in alternating 180-degree rotation or are flipped along the X-axis so that the standard cells can share power and ground busses. If the ASIC core has routing congestion owing to the limited number of routing layers, one solution is to create routing channels between rows. These all can be separated individually or as pairs.

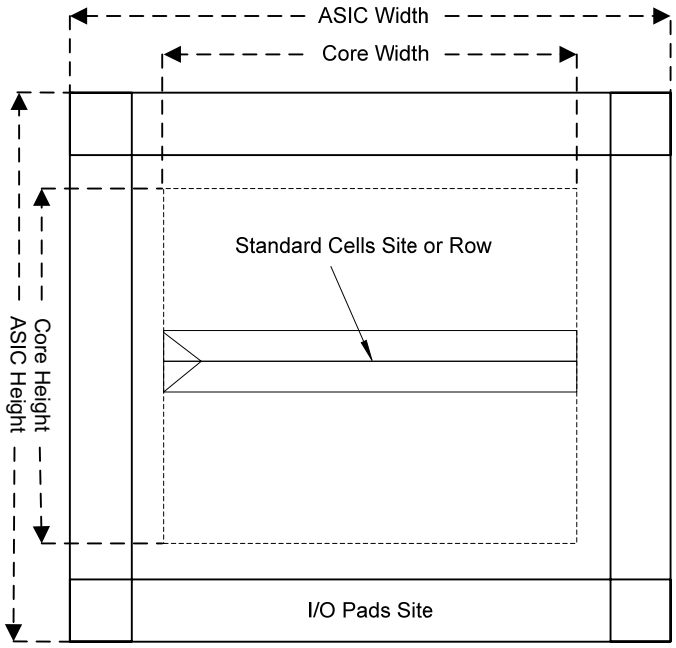


**Figure 2-3** ASIC Design Initial Floorplan

## 2.5 Pad Placement

Correct I/O pad placement and selection is important for the correct function of any ASIC design. As mentioned in Chapter 1, for a given ASIC design there are three types of I/O pads. These pads are power, ground, and signal.

It is critical to functional operation of an ASIC design to insure that the pads have adequate power and ground connections and are placed properly in order to eliminate electromigration and current-switching noise related problems.

Electromigration (EM) is the movement or molecular transfer of metal from one area to another area that is caused by an excessive electrical current in

the direction of electron flow (electron "wind"). Electromigration currents exceeding recommended guidelines can result in premature ASIC device failure. Exceeding electromigration current density limits can create voids or hillocks, resulting in increased metal resistance, or shorts between wires, and can impair ASIC performance. Figure 2-4 shows electromigration damage due to excess current captured by Electron Scanning Microscopy (ESM) with 10K magnification.

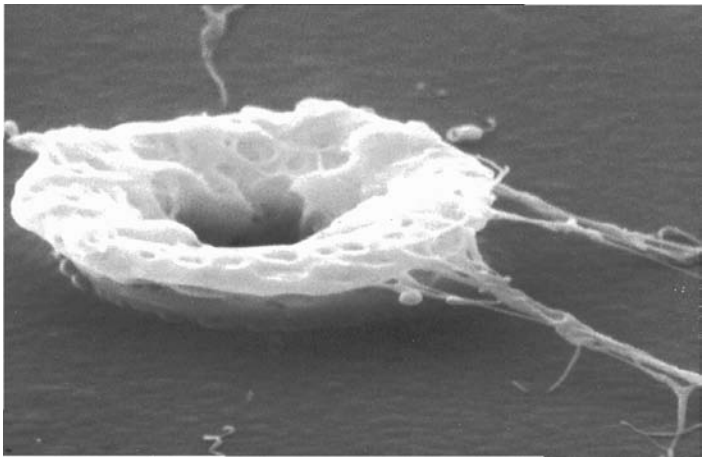

**Figure 2-4** Electromigration Damage

Using Equation 2.5.1, one can determine the minimum number of ground pads required to satisfy the electromigration current limits. The required number of power pads is equal to the required number of ground pads and is given by

$$N_{gnd} = \frac{I_{total}}{I_{\max}}, \tag{2.5.1}$$

where $N_{gnd}$ is the number of ground pads; $I_{total}$ is the total current in an ASIC design (which is the sum of its static and dynamic currents in amperes); and $I_{\max}$ is the maximum EM current in amperes per ground pad.

Switching noise is generated when ASIC outputs make transitions between states. An inadequate number of power and ground pads will lead to system data errors due to these switching noise transients. There are two types of mechanisms that can cause noise:

- $dv/dt$ caused by a capacitive coupling effect
- $di/dt$ caused by an inductive switching effect

The capacitive coupling effect is the disturbance on the adjacent package pin caused when switching transients inject pulses via parasitic coupling capacitance.

The maximum $C(dv/dt)$ noise occurs when the ASIC output current nears the maximum current for a given capacitive output load of $C$. This noise problem can be resolved by proper pad placement, package pin selection, ASIC output pad type and drive current, and input pad type.

To reduce or eliminate capacitive coupling effects during I/O pad placement and selection, one may consider the following guidelines:

- Isolate sensitive asynchronous inputs such as clock or bidirectional pins from other switching signal pads with power or ground pads
- Group bidirectional pads together such that all are in the input or output mode
- Group slow input pads together positioning them on higher inductance package pins
- Use input pads with hysteresis as much as possible

The inductive switching effect is related to simultaneous switching ASIC outputs that induce rapid current changes in the power and ground busses. The inductance in the power and ground pins cause voltage fluctuations in the ASIC internal power and ground level relative to the external system.

These rapid changes in current can change the ASIC input pads' threshold and induce logic errors or may cause noise spikes on non-switching output pads that affect signals connected to other systems.

The maximum $L(di/dt)$ occurs when ASIC output starts to make the transition to another voltage level and its absolute current increases from zero through a wire with inductance of $L$. Factors such as process, ambient

temperature, voltage, location of output pads, and number of simultaneous switching output pads determine the magnitude of inductive switching noise.

To control inductive switching noise, enough power and ground pads must be assigned and placed correctly. This way the noise magnitude will be limited. This noise reduction will prevent inputs of ASIC design from interpreting the noise as valid logic level.

Successful reduction of inductive switching noise can be accomplished by the following:

- Reduce the number of outputs that switch simultaneously by dividing them into groups with each group having a number of delay buffers inserted into their data paths
- Use the lowest rated sink current or low-noise output pads as long as speed is not an issue
- Place the simultaneously switching output or bidirectional pads together and distribute power and ground pads among them according to their relative noise rating
- Assign static and low frequency input pads to higher inductance package pins
- Reduce the effective power and ground pin inductance by assigning as many power and ground pads as possible

## 2.6 Power Planning

The next step is to plan and create power and ground structures for both I/O pads and core logic. The I/O pads' power and ground busses are built into the pad itself and will be connected by abutment.

For core logic, there is a core ring enclosing the core with one or more sets of power and ground rings. A horizontal metal layer is used to define the top and bottom sides, or any other horizontal segment, while the vertical metal layer is utilized for left, right, and any other vertical segment. These vertical and horizontal segments are connected through an appropriate via cut. The next consideration is to construct the standard cell power and ground that is

internal to the core logic. These internal core power and ground busses consist of one or two sets of wires or strips that repeat at regular intervals across the core logic, or specified region, within the design. Each of these power and ground strips run vertically, horizontally, or in both directions. Figure 2-5 illustrates these types of power and ground connections.

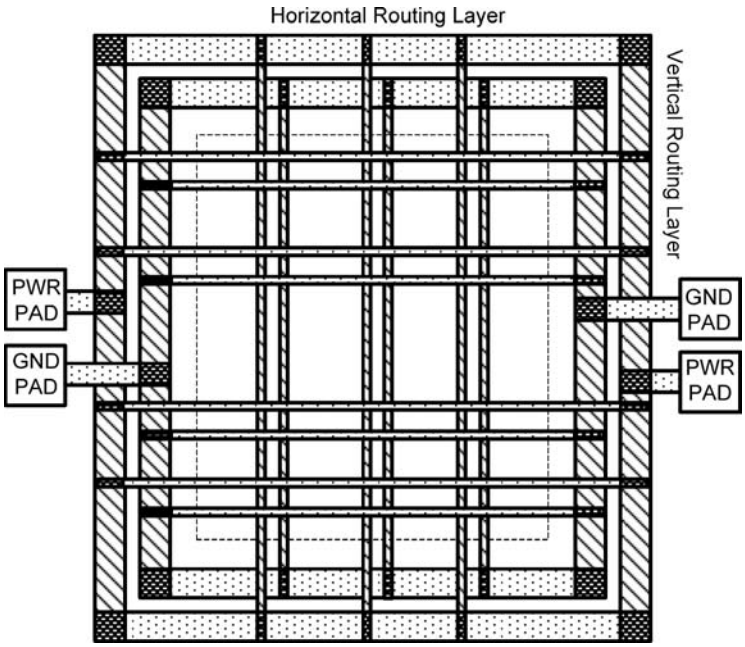


**Figure 2-5** Ring and Core Power and Ground

If these strips run both vertically and horizontally at regular intervals, then the style is known as power mesh. The total number of strips and interval distance is solely dependent on the ASIC core power consumption.

As the ASIC core power consumption (dynamic and static) increases, the distance of power and ground strip intervals increases. This increase in the power and ground strip intervals is used mainly to reduce overall ASIC voltage drop, thereby improving ASIC design performance.

In addition to the core power and ground ring, macro power and ground rings need to be created using proper vertical and horizontal metal layers. A macro ring encloses one or more macros, completely or partially, with one or more sets of power and ground rings.

Another important consideration is that when both analog and digital blocks are present in an ASIC design, there is a need for special care to insure that there is no noise injection from digital blocks or core into the sensitive circuits of analog blocks through power and ground supply connections.

Much of this interference can be minimized by carefully planning the power and ground connections for both digital core and analog blocks. There are several methods to improve the noise immunity and reduce interference.

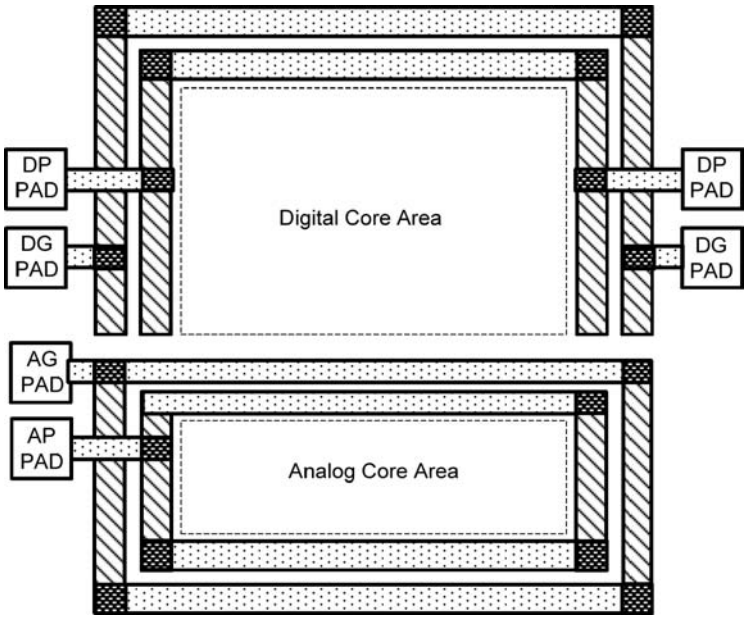


**Figure 2-6** Decoupled Analog and Digital Core Power Supply

The most effective method is to decouple the digital and analog power and ground by routing the digital power/ground (DP and DG) and analog power/ground (AP and AG) supply connections separately [3] as shown in

Figure 2-6. However, this decoupling will not be complete if there is ground connectivity from the die substrate through standard cells' ground (i.e. source).

In order to make sure that analog circuits are completely decoupled from digital circuits, one needs to separate the substrate from the ground in the standard cells (e.g. NWELL Process). This is not mandatory but depends on how sensitive the analog circuit is with respect to noise injection from the digital core area.

It is strongly recommended to check for power and ground connectivity and/or any physical design rule violations after construction of the entire power and ground network.

Depending on the ASIC design's power supply requirements, the width of the core macro power and ground ring could exceed the maximum allowable in order to reduce their resistance during floorplanning and power analysis. This increase in power and ground width could be problematic from manufacturing point of view.

The main problem with wide metal (i.e. exceeding manufacturing limits) is that a metal layer cannot be processed with uniform thickness, especially when applied over a wide area. The metal becomes thin in the middle and thick on the edges causing yield and current density problems. To solve this metal density problem, one can either use multiple power and ground busses or use metal slotting.

Most of today's physical design tools are capable of performing slotting on a wire-by-wire or segment-by-segment basis by cutting those wires or segments with widths greater than the specified maximum into multiple, thinner segments of the same type.

These maximum specified widths for different metal layers are determined by semiconductor manufacturers as a slot-width and need to be included in the physical design tool technology file.

## 2.7 Macro Placement

Typically, the placement of macros takes place after I/O placement, and after power and ground bus structure has been defined. Macros may be memories, analog blocks, or in the case of hierarchical style, an individually placed and routed subcircuit. Proper placement of these macros has a great impact on the quality of the final ASIC design.

Macro placement can be manual or automatic. Manual macro placement is more efficient when there are few macros to be placed and their relationship with the rest of the ASIC design is known. Automatic macro placement is more appropriate if there is not enough information on which to base the initial macro placement and/or the number of macros is large.

During the macro placement step, one needs to make sure that there is enough area between blocks for interconnections. This process (commonly known as channel allocation or channel definition) can be manual or can be accomplished by floorplan tools. The slicing tree is used by the floorplan algorithm for slicing floorplan during macro placement and to define routing channels between the blocks.

Most of today's physical design tools use a global placer to perform the automatic initial macro placement based on connectivity and wire length reduction. Wire length optimization is the most prevalent approach in automatic macro placement.

With increases in the number of embedded blocks such as memories, placing macros of varying sizes and shapes without a good optimization algorithm can result in fragmentation of placement and routing space that can prevent a physical design from being able to complete the final route.

One of the basic algorithms used for automatic macro placement considers that macros are connected to each other by nets and are supposed to exert attractive forces on each other by means of wire length proportional to the distance between these macros.

Automatic macro placement is an iterative process. During the macro placement process, macros are free to move until the equilibrium or optimum wire length is achieved. It is interesting to note that in this algorithm, if there is no

relationship between the macros, they tend to repel each other and their placement result may not be optimum.

To improve the placement quality of macros that are not related to each other, one may consider simultaneous standard cell and macro placement provided the physical design tool can deal with both macro and standard cell placement at once. In terms of algorithms, while commercial physical design tools have considerably improved in the past few years, automatic macro placement is still in the early stages of development compared to standard cell placement.

The implementation challenge associated with macro placement is conceptually a time and space problem that needs to be solved simultaneously.

A well-developed macro placement algorithm must be able to handle widely differing shapes and sizes, macro orientation, congestion, and timing-driven placement. Although many improvements have been incorporated into the macro placement algorithms to insure the quality of their placement, one might need to modify the resulting location and/or orientation in order to achieve an optimum floorplan based on the physical quality of measurement.

When it is not an easy task to measure the macro placement quality of an ASIC design containing a large number of blocks, there are some basic physical measurements that one can adopt.

Given a placement solution, the physical measurements could be wire length, data flow direction (e.g. the macro placement relative to each other as well as to standard cell placement), or macro, port accessibility and related timing.

The total wire length for a given placement is a good indicator when comparing different placements of the same physical design. In order to decrease overall wire length, ensure that the chip area is not segmented by the macro's placement.

To avoid area segmentation, macros should be positioned such that the standard cell area is continuous. An area with close to 1:1 aspect ratio is recommended as it allows standard cell placers to utilize the area more efficiently and thereby reduce total wire length.

The segmented floorplan leads to an excess of wire length interconnections from the standard cells located at the bottom of the die to those at the top of the die. Thus, it is necessary that the macros be kept along the ASIC core area in order to avoid a floorplan segmentation problem.

Figure 2-7 shows a problematic segmented floorplan that may lead to long interconnections between the bottom and top of the die.

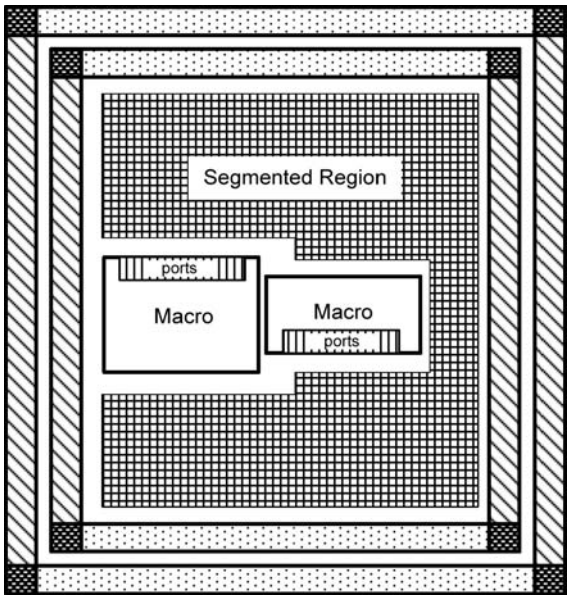


**Figure 2-7** Segmented Floorplan

Another aspect of increased wire length is related to macro placement with respect to their orientation and pin placements. Depending on the macro orientation and their actual pin location, the length of the nets connecting to the macros can be different, and can have significant impact on the routing optimization process.

With respect to wire length reduction, macros should be oriented such that their ports are facing the standard cells, or core area, and their orientation should match the available routing layers. Thus, any macro placement

algorithm requires computing the macro's interconnect distance by including proper orientation and pin positions.

Figure 2-8 shows a floorplan with macro ports facing the standard cell region, thereby minimizing localized increase in wire length.

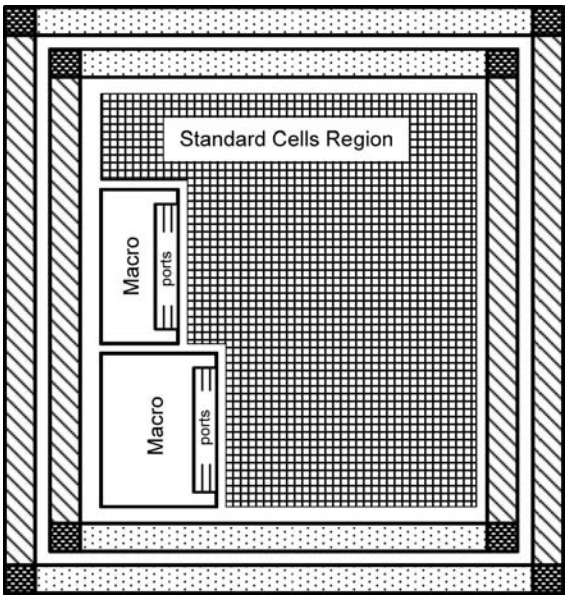


**Figure 2-8** Floorplan with Macros Facing Standard Cells Region

Another aspect of physical measure is that of macro placements relative to each other, as well as to the standard cell placement, and the macro port accessibility has a direct impact on the chip's final routing. The macro placement, and thereby the quality measure, can be determined by the analysis of routing congestion produced by a global router.

Most global routers are capable of producing both graphical and text statistical reports. The graphical report, also referred to as the congestion map, provides a visual aid to see where routing congestion exists (e.g. hot spots). The statistical report is a good indication of how much a physical design is congested.

The most common scenarios that cause physical design routing congestion are: there may not be enough space between the macros to provide routing channels–especially for I/O connections and macros (if these macros are placed at the ASIC periphery and over the macro); routing is prohibited; or standard cell trap pockets may be around the edges of macros or within the corners of the floorplan.

Standard cell trap pockets are long, thin channels between macros. If many cells are placed in these channels, routing congestion can result. Therefore, these channels need to be kept free for most standard cells and should be available for repeater or buffer insertion (if this type of insertion is supported by the physical design tool). Figure 2-9 shows a floorplan with a standard cell trap pocket.

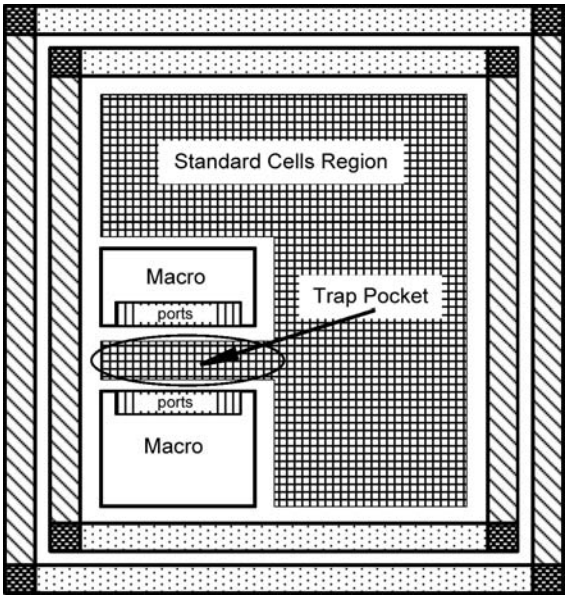


**Figure 2-9** Floorplan with Standard Cells Trap Pocket

After macro placement and before performing global routing, most physical designs require keep-out or buffer-only regions to be defined by drawing a

blockage layer over an area containing macros to prevent the placer from moving any standard cells into those regions.

Naturally, the wires that are used in keep-out regions have a tendency to be long. By allowing buffer insertion in those areas by using a buffer-only region (or blockage), the placer will taper these long nets and thus avoid the long transition times associated with them.

These blockage layers are created over pre-placed macros such that their power and ground rings are covered. Blockage layers are also used to relieve routing congestion around the macro's corners. When a macro is blocked on many routing layers, wires have a tendency to detour around corners and connect to nearby standard cells thereby creating routing congestion at the corner of the macros.

To reserve more resources for the router, one can draw a blockage layer at these corners. These blockage regions can be simple or gradual as illustrated in Figure 2-10.

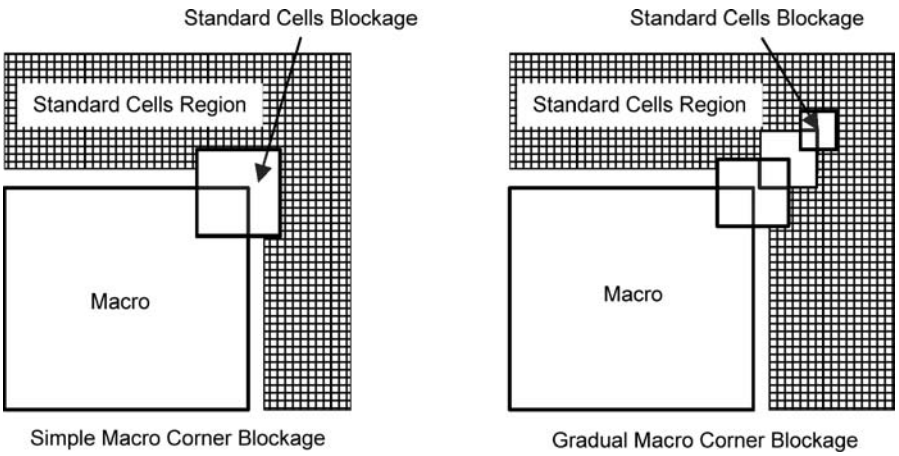


**Figure 2-10** Macro Corner Standard Cells Blockage

After refinement of floorplan and macro placement, standard cells are placed and connectivity analysis is performed. Connectivity analysis is the process of studying the connections between macro pairs, macro, I/O pads, and

related standard cell instances. Connectivity analysis is also used to identify macros that have substantial direct connectivity and to refine their locations accordingly.

This analysis is conducted by using what is known as fly lines. When fly lines are activated through physical design or place-and-route tools, Graphic User Interface (GUI) displays the lines that mark the connections between currently selected instances (e.g. standard cells, macros, or I/O pads). Using fly lines, one can analyze and identify situations where moving or rotating macros will yield shorter wire lengths that improve the overall ASIC routability during the floorplanning stage of the physical design cycle.

## 2.8 Clock Planning

Clock planning is considered another part of floorplanning. The idea of the implementation of clock distribution networks is to provide clock to all clocked elements in the design in a symmetrically-structured manner.

Although most ASIC designs use clock tree synthesis, clock tree synthesis may not be sufficient for very high-performance and synchronized designs. In this case, one needs to implement the distributed clock networks manually in order to minimize the skew between communicating elements due to their line resistance and capacitance.

The basic idea of manual implementation of clock distribution networks is to build a low resistance/capacitance grid similar to power and ground mesh that covers the entire logic core area as shown in Figure 2-11.

It is important to note that this type of clock grid does not rely on the matching of components such as clock buffers. However, it can present a systematic clock skew.

In order to minimize such clock skew, a clock tree that balances the rise and fall time at each clock buffer node should be utilized during clock planning. This minimizes the hot-electron effect. The problem of hot-electron occurs

when an electron gains enough energy to escape from a channel into a gate oxide.

The presence of hot-electron effect in the gate oxide area causes the threshold voltage of the device to change and thus alters the delay of the clock buffers. This in turn leads to an additional skew. Hence, balancing the rise and fall times of all clock buffers means that hot-electron effect influences the clock buffers at the same rate, minimizing unpredictable skew.

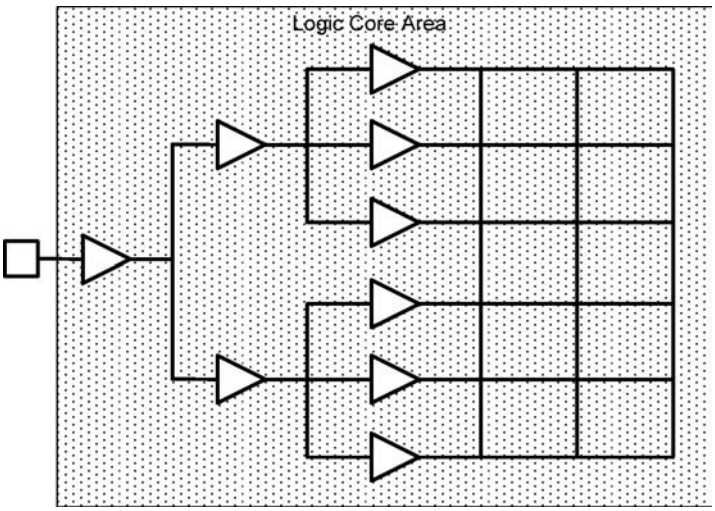


**Figure 2-11** Clock Distribution Network

It is essential to realize that clock grid networks consume a great deal of power due to being active all the time and it may not be possible to make such networks uniform owing to floorplanning constraints (e.g. to spread the power dissipation evenly across the chip).

Another aspect of clock planning is that it is well suited to hierarchical physical design. This type of clock distribution is manually crafted at the chip level, providing clock to each sub-block that is place-and-routed individually.

To minimize the clock skew among all leaf nodes, the clock delay for each sub-block must be determined and the design of the clock planned accordingly.

Figure 2-12 illustrates typical hierarchical clock planning.

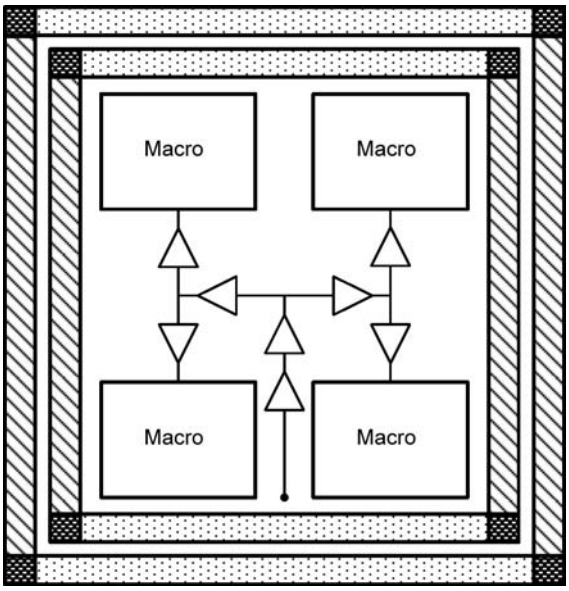


**Figure 2-12** Hierarchical Clock Planning

## 2.9 Summary

In this chapter, we have explained various basic aspects of physical design data preparation and ASIC physical design and floorplanning alternatives.

In the area of data preparation, we have provided a general idea of the technology files that are required by physical design tools, and have provided an example of a Verilog structural netlist with the most common description of its syntax and keywords.

In the design constraint section, we have discussed several important timing and design constraints and their impact on the quality of the ASIC physical design.

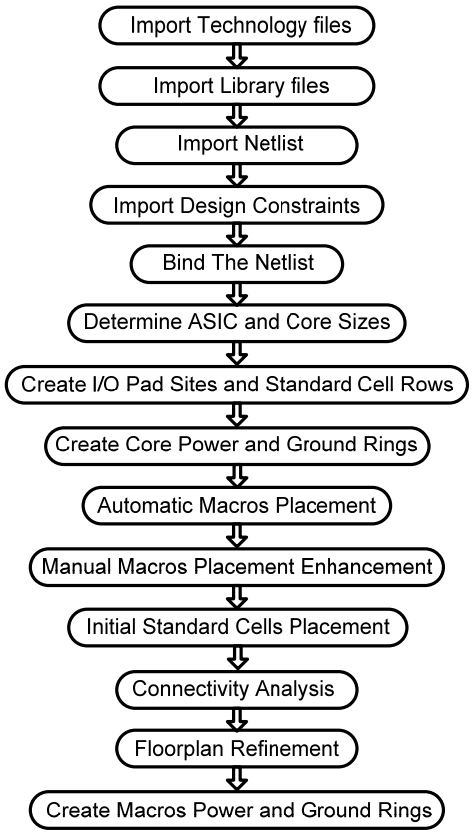


**Figure 2-13** Basic Floorplanning Steps

In the design implementation section, we have outlined the fundamentals of different floorplanning styles, mainly flat and hierarchical, and their advantages, and we have explained basic floorplanning techniques.

In addition, we have outlined the importance of automatic partitioning of a design, and port optimization and minimization for ASIC design. We should

note that selection of a floorplanning style depends upon many factors such as the type of ASIC, area, and performance, and relies heavily on one's physical design experience.

In the input-output section, we have explained placement and given basic guidelines with respect to coupling capacitance and inductive switching.

In the power and ground section, we have shown several styles of creating power and ground connections. Again, depending upon the floorplanning style, the power and ground connections need to be designed to meet the ASIC power requirements and can vary from design to design.

In the macro placement section, we have illustrated various macro placements based on industry practices. In addition, we should note that regardless of what style of macro placement is used during floorplanning, a well thought-out floorplan and macro placement leads to a higher quality of the final ASIC design with respect to performance and area.

Finally, in the clock planning section we have briefly discussed the manual clock planning topology and its importance for high-speed design applications.

Figure 2-13 exhibits the basic steps that are involved during the physical design floorplanning phase.

## References

[1] Eli Sternheim, Rajvir Singh, Rajeev Madhavan, and Yatin Trivedi, *Digital Design and Synthesis with Verilog HDL*, Automata Publishing Company, 1993.

[2] Naveed Sherwani, *Algorithms for VLSI Physical Design Automation, SECOND EDITION,* Kluwer Academic Publishers, 1997.

[3] R. Jacob Baker, Harry W. Li, and David E. Boyce, *CMOS, Circuit Design, Layout, and Simulation,* IEEE Press Series on Microelectronic Systems, 1998.

Springer

http://www.springer.com/978-0-387-36642-5

Physical Design Essentials
An ASIC Design Implementation Perspective
Golshan, K.
2007, XX, 212 p., Hardcover
ISBN: 978-0-387-36642-5