

Physical Design Optimization of an IP Core

Major Project Report

*Submitted in partial fulfillment of the requirements
for the degree of*

**Master of Technology
in
Electronics & Communication Engineering
(Embedded Systems)**

By

**Ashish Kumar
(17MECE09)**



Department of Electronics & Embedded Systems

Institute of Technology

Nirma University

Ahmedabad-382481

May 2019

Physical Design Optimization of an IP core

Major Project Report

*Submitted in partial fulfillment of the requirements
for the degree of*

**Master of Technology
in
Electronics & Communication Engineering
(Embedded Systems)**

By
Ashish Kumar
(17MECE09)

Under the guidance of

External Project Guide:

Mr Vishal Katba,
Manager,
Central IP Hardening Group,
Intel Technology India Pvt. Ltd., Bangalore

Internal Project Guide:

Prof. B.D.Fataniya,
Assistant Professor in EC Engineering,
Institute of Technology,
Nirma University, Ahmedabad



Department of Electronics & Communication Engineering

Institute of Technology

Nirma University

Ahmedabad-382481

December 2018

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and Intel Technology India Pvt. Ltd. and has not been submitted elsewhere for a degree.
2. Due acknowledgment has been made in the text to all other material used.

- **Ashish Kumar**

17MECE09

Disclaimer

“The content of this thesis does not represent the technology, opinions, beliefs, or positions of Intel Technology India, its employees, vendors, customers, or associates”.



Internal Certificate

This is to certify that the Major Project entitled “**Physical Design Optimization of an IP Core**” submitted by **Ashish Kumar (17MECE09)**, towards the partial fulfillment of the requirements for the degree of Masters of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Project, to the best of our knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Prof. B.D.Fataniya

Internal Guide,
Assistant Professor in EC Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr. N.P.Gajjar

Program coordinator,
Professor in EC Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr. D. K. Kothari

Professor and Head,
EC Engineering Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Alka Mahajan

Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

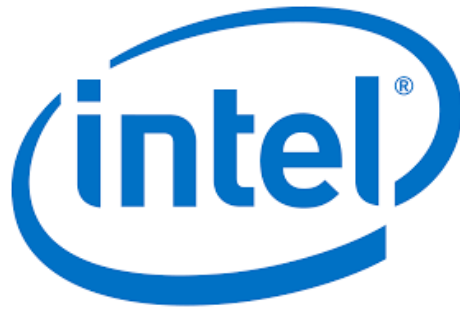
I, **Ashish Kumar, 17MECE09**, give undertaking that the Major Project entitled ”**Physical Design Optimization of IP Core**” submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Electronics and Communication Engineering (EMBEDDED SYSTEMS)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by
Prof. Bhupendra D Fataniya
(Signature of Guide)



External Certificate

This is to certify that the Major Project entitled “**Physical Design Optimization of an IP Core**” submitted by **Ashish Kumar (17MECE09)**, towards the partial fulfillment of the requirements for the degree of Masters of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Project, to the best of our knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Bengaluru

Mr. Vishal Katba

External Guide,

Intel Technology India,

Bengaluru.

Acknowledgement

I would like to express my gratitude and sincere thanks to Dr. N.P.Gajjar, PG Coordinator of M.Tech Embedded Systems and Prof. B.D.Fataniya for providing guide-lines during the review process.

I take this opportunity to express my profound gratitude and deep regards to Prof. B.D.Fataniya, guide of my internship project for his exemplary guidance, monitoring and constant encouragement.

I would also like to thank Mr. Vishal Katba, external guide of my intern-ship project from Intel India Technologies Pvt. Ltd., for guidance, monitoring and encouragement regarding the project.

- Ashish Kumar

17MECE09

Contents

Declaration	iii
Disclaimer	iv
Internal Certificate	v
Statement of Originality	vi
External Certificate	vii
Acknowledgement	viii
Abstract	xiii
Abbreviation Notation and Nomenclature	xv
1 Introduction	1
1.1 Objective	1
1.2 Motivation	1
1.3 Problem Statement	2
1.4 Approach	2
1.5 Scope of Work	2
1.6 Outline of Thesis	3
2 Literature Review	4
2.1 VLSI Design Flow[1]	4

2.1.1	System Specification[1]	4
2.1.2	Architectural Design[1]	5
2.1.3	Behavioural or Funtional Design[2]	6
2.1.4	Logic Design[2]	6
2.1.5	Physical Design[2]	6
2.1.6	Fabrication	9
2.1.7	Packaging Testing and Debugging	9
2.2	Stages in Physical Design:	9
2.3	Sign-off Checks in Physical Design:	17
3	How we optimized the Design	23
3.1	Optimization in each Steps	23
3.1.1	Floorplan	23
3.1.2	Placement	24
3.1.3	Clock Tree Synthesis	26
3.1.4	Route	27
4	Results	29
5	Conclusion	32
	Bibliography	33

List of Figures

2.1	VLSI Design Flow [1]	5
2.2	Physical Design Methodology [1]	7
2.3	Main Stages in Physical Design Flow [1]	10
2.4	Floorplan [3]	11
2.5	Floorplan	12
2.6	Placement [3]	13
2.7	Clock Network Before CTS [3]	14
2.8	Clock Tree after CTS [3]	14
2.9	Clock source and sink [3]	15
2.10	Power Consumption in Clock Net [3]	15
2.11	Routing [3]	16
2.12	VCLP Flow at RTL[4]	19
2.13	VCLP Flow at Netlist Level[4]	19
2.14	VCLP Flow at PG Level[4]	20
2.15	Flow Diagram for FEV[5]	21
3.1	Floorplan with final constraints	24
3.2	Congestion after placement	25
3.3	Clock Tree after CTS	26
4.1	Results after Partial Placement of 40 percent	29
4.2	Results after Keepout Margin of 1 1 1 1	30
4.3	Final Congestion in design	31

4.4 Final Congestion numbers 31

Abstract

Aggressive scaling down of technology in last five decades, the integrated circuit design has entered in the nanometer scale era. As the scaling down may help in production of more powerful chips, at the very same time designers face a lot of challenges. The physical design stage encounter the most of the challenges. Firstly, with the reduction in the die-size the placement and routability becomes an issue for the design. It may lead to DRC errors or may lead to an unroutable design. Secondly, with the evolution in the fabrication technology the total utilization of the die-size has increased. So insertion of more standard cells in the same die area is possible, with this an issue of congestion has come out. If we insert more cells in less area, than there will be less space for the wires to run, and it will cause the congestion. Thirdly, the processors of today are being clocked over 3 GHz, so timing becomes an issue, so timing of the chip has to be met according to the timing-budget. Fourthly, the power management of the chip, today most of the chip are multi-voltage design, so they have multiple power domains. So the design should be such that it must have power rails of that power domain and also the global power rails, with the power gates or power switches.

Also the impact of process variations is also increasing with the shrinking size of the chip. Several important process variations effects show strong dependency on the underlying patterns of the die, these problems can be addressed through appropriate physical design techniques. The common technique is the addition of the derates in the design, which will give some percentage of relaxation in terms of power, timing and wire delays.

In this report, the ideas for solving the routability, congestion, timing, multi-voltage design and several sign-off checks will be given. The sign-off checks will be for Logical Equivalence Checks (FEV), low power checks, timing estimation (STA) checks. The Logical Equivalence Check (FEV) checks that all the cells have been inserted by the tool or not, sometimes the tool may even optimize the logic, those checks are also incorporated in the Logical Equivalence Check (FEV). The concepts of Unified Power Format will be very helpful for low power checks. The usage of isolations, level shifters, retention cells, power domains is the basic of low power checks. This check

determines that the isolations or level-shifters are inserted at proper places or not. For the timing estimation (STA) check the concept of timing i.e. setup and hold timing are very important. If there is a setup violation then according to the type of cells we bound them together and bring all the cells closer so there is no setup violations. Also the setup time of a SoC determines the operational frequency of the chip.

Abbreviation Notation and Nomenclature

HDL	Hardware Description Language
VHDL	Very High Speed Integrated Circuit Hardware Description Language
SoC	System on Chip
UPF	Unified Power Format
FEV	Formal Equivalency Verification
STA	Static Timing Analysis
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
ALU	Arithmetic and Logic Unit
I2C	Inter-Integrated Circuit
PCI	Peripheral Component Interconnect
PCB	Printed Circuit Board
DIP	Dual In-line Package
QFP	Quad Flat Package
BGA	Ball Grid Array
PGA	Pin Grid Array
MCM	Multi Chip Modules
DRC	Design Rule Check
LEC	Logical Equivalence Check
VCLP	Verdi Conformal Low Power
LVS	Layout Versys Schematic
PDN	Power Delivery Network
ECO	Engineering Change Order
GRC	Global Route Congestion
GDSII	Graphic Data System
CTS	Clock Tree Synthesis
NDR	Non Default Rule
IP	Intellectual Property

NIU Network Interface Unit
SPICE Simulation Program with Integrated Circuit Emphasis
EDIF Electronic Design Interchange Format
OVM Open Verification Methodology
UVM Open Verification Methodology
WLM Wire Load Model
ASIC Application Specific Integrated Circuits

Chapter 1

Introduction

1.1 Objective

The objective of this thesis is to optimize an IP in terms of area, power, timing, performance, reduce battery consumption for the chip with the help of modern day Physical Design Tools and signoff techniques

1.2 Motivation

The rule proposed by Co-founder of Intel Gordon Moore, which says that, number of transistors per square inch on a chip will get doubled every 18 months, and this rule become standard of nearly every silicon manufacturing industry.

Now, as continuous attempts are being made to reduce the size, improve the performance, reduce the power consumption, due to this the transistor size is decreasing with the lithography, and due to this the leakage power becomes an issue.

Also when reducing the size of the core area of a chip and increasing the number of transistor, congestion among the cells increases, cross-talk increases, area utilization also increases and sometimes routing becomes nearly impossible for a design.

At one side we are reducing the area and accommodating more cells, we need to make sure that there should be enough space to route the design. For these problems to not to occur, we optimize the designing of our design.

1.3 Problem Statement

The recent trend of small, handy and power full devices has led the chip manufactures, to go below 5nm technology. Packing of more and more standard cells for the added functionality and improve in performance, while still keeping power dissipation in check. Therefore the problem faced during the back-end physical design stage are discussed and the solutions available to us are also discussed. The work includes, the starting form the floorplan of the design. Starting from the port, macro placement and bounds, blockages creation. Then moving to placement stage, checking for timing, congestion, health checks, further going to the clock tree synthesis, correcting timing there and proceeding to routing, checking for DRCs. Meanwhile after every stage, performing some sign-off checks for the low power design, functional equivalence checks, static timing analysis, power delivery network and changing the design as highlighted by any of the sign-off checks failure.

The output is a chip, which is clean in terms of timing, congestion and all sign-off checks and can be sent to the fabrication unit. At the fabrication unit, the chip is tested rigourously on the software and few samples are fabricated for testing, before the mass production starts.

1.4 Approach

The suggested approach is that, performing some experimental runs on the given inputs and constraints, checking for the correctness of the design, checking the health checks, reports. Making some changes to clean the reports and health checks. Knowledge of floorplan is required for the kick start of the design process. A good floorplan can make the deisgn routable very easily, whereas a bad floorplan can make an easy design unroutable. Methods to reduce the congestion, timing violations, low power implementation are very helpful for the design closure.

1.5 Scope of Work

To implement the RTL to a chip, we first need to synthesize it to get the netlist. To synthesize we use Design Compiler by Synopsys. After netlist is obtained, we perform few sign-off checks, which

will inform for missing instances or wrongly placed instances. After correction of all issues, we move to the most important part, placement and routing. This stage will do floorplan, placement, clock tree synthesis, routing and standard filling on the chip and generate the GDSII file, which can be sent to the fabrication unit.

1.6 Outline of Thesis

This dissertation comprises of five chapters. In the first chapter the objective, motivation, problem statement, approach of work and outline of the report is given. In the second chapter, basic knowledge of VLSI design flow, Physical Design methodology, stages in physical design is provided and along with that few sign-off check are also discussed. In the third chapter, methods of optimizing the design are discussed with respect to the floorplan, placement, congestion, timing, clock tree synthesis, routing, DRCs, ECOs. In the fourth chapter, the results of the methods for the optimizing the design are listed. In the fifth chapter, work done in 1 year is concluded. And the sixth chapter gives the references.

Chapter 2

Literature Review

2.1 VLSI Design Flow[1]

The design cycle of a chip starts with the specifications, which are obtained from the customer or the marketing team, then the design and RTL team will meet and selects all the possible and practical specifications. After final specifications has been picked out, the RTL team will start writing the HDL(Hardware Description Language) code of the chip or SoC. There will be UPF team, which will decide the power intent of the chip. After many iterations of RTL ,UPF ,physical design and sign-off checks the final chip is sent to fabrication unit and we get a chip which can do stuffs.

2.1.1 System Specification[1]

This is the first step in the development of a chip. It is some sort of high level representation of a system. The main factors considered during system specification are :

- Functionality
- Performance
- Physical Dimensions

The specification can be anything, but it is always a compromise between the available technology and economical viability.

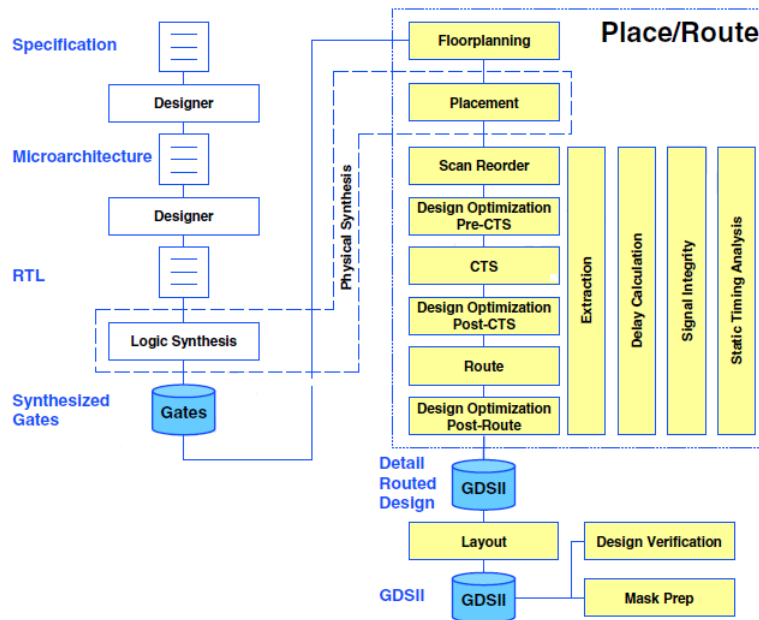


Figure 2.1: VLSI Design Flow [1]

2.1.2 Architectural Design[1]

During this step the selection of architecture/micro-architecture takes place. The very famous architectures are RISC (Reduced Instruction Set Computer) or CISC (Complex Instruction Set Computer). The architecture/micro-architecture is selected on the basis of number of:

- ALU
- Floating Point units
- Number and structure of pipelines
- Size of cache
- Size of address and data buses
- Communication among blocks i.e. I2C, PCI etc.

By the selection of architecture for a design, the power intent, performance and die size can be predicted.

2.1.3 Behavioural or Funtional Design[2]

During this step main functional units of the system are defined also, the type of communication between the units is also defined. The area, power, time budgeting, performance of each unit is also estimated in this step only.

The behavioral aspects of the system are considered without implementation specific information. For example, it may specify that a multiplication is required, but exactly in which mode such multiplication may be executed is not specified. We may use a variety of multiplication hardware depending on the speed and word size requirements. The key idea is to specify behavior, in terms of input, output and timing of each unit, without specifying its internal structure .

2.1.4 Logic Design[2]

In this step the control flow, word widths, register allocation, arithmetic operations, and logic operations of the design that represent the functional design are derived and tested[1].

This description is called Register Transfer Level (RTL) description. RTL is expressed in a Hardware Description Language (HDL), such as VHDL or Verilog. This description can be used in simulation and verification. This description consists of Boolean expressions and timing information. The Boolean expressions are minimized to achieve the smallest logic design which conforms to the functional design. This logic design of the system is simulated and tested to verify its correctness. In some special cases, logic design can be automated using high level synthesis tools. These tools produce a RTL description from a behavioral description of the design.

2.1.5 Physical Design[2]

During this process a geometric representation or we can also say it as "layout" is plotted out of the circuit representation from netlist. In the netlist the gates can be imagined as the shape of the gates that we used to see, but during layout they get a specific height and width.

The details for the layout is dependent on the design rules, which are defined on the basis of quality of fabrication and the electrical properties of the fabricating materials. As the physical design step is a complex process, so it is divided into number of sub-steps. During physical design a number of varification and validation checks are performed over the layout.

Physical Design Methodology

First of all the specifications are collected from the customer or within the company, for a specific design according to the needs. Then the specifications are compared with the, what can be possible for the design team, with the current tools, current fabrication technologies. There every possible specification will be taken and some of the specification will get discarded. Secondly, the selection of micro-architecture will take place it can be Harward micro-architecture, Von-Newman micro-architecture, modified Harward micro-architecture or any newly developed architecture according to the need. Figure 2.2 shows the Physical Design Flow for the ASICs.

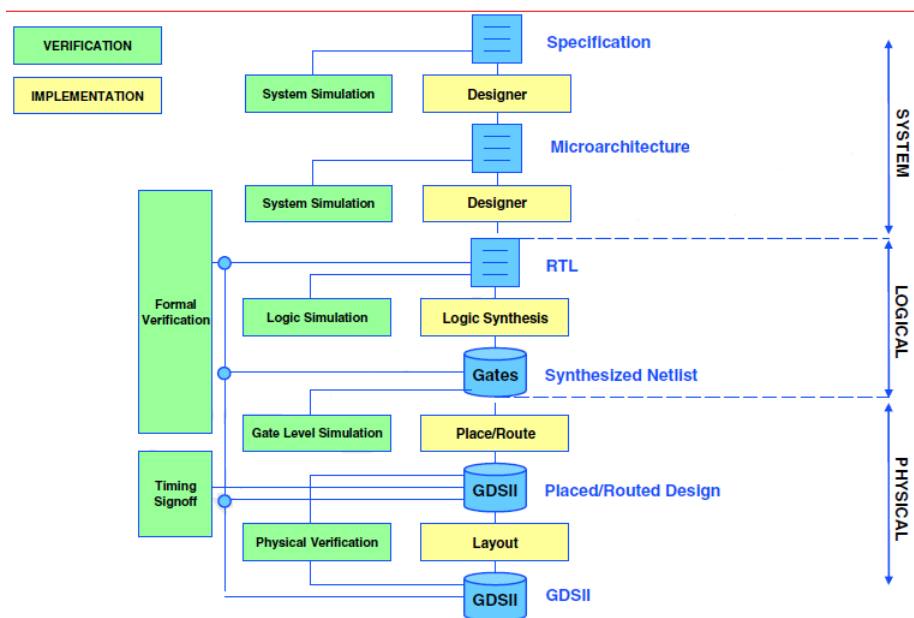


Figure 2.2: Physical Design Methodology [1]

Thirdly, according to micro-architecture, the RTL coding will take place. RTL is generally written

in HDL (Hardware Description Language), it can be VHDL, Verilog, System Verilog, System C. With the RTL the coding for UPF (Unified Power Format) also takes place, which contains the power intent for the design. UPF contains information for power domains, isolation cells, level shifter cells, retention cells and always-on buffers. After the completion of coding of RTL and UPF, formal verification of both takes place. After the successful completion of verification the logical simulation takes place, which will depict the block of SoC for which, the RTL and UPF are written for. Again if any discrepancies are found then the correction of RTL and UPF takes place. After many iterations, RTL comes out and send to the Back-end teams for further processes.

After the RTL comes to the Back-end team, first and foremost thing is the synthesis of RTL to make netlist. Netlist is also in Verilog. At the synthesis stage, several sign-off checks will be done so that, RTL should be clean before it reaches the Physical Design team. FEV will be performed to check the logical connection of the cells as per netlist, it will also take care of the optimizations done by the tool. VCLP (Verdi Conformal Low Power) will check for the multi-voltage design attributes, such as power domains are correct or not, isolations are placed correct or not, level shifters are having correct primary and secondary supplies or not, likewise. STA tool will check for the timing for setup and hold parts. Timing should be clean before actual design process starts.

After synthesis is done, the netlists are send for the actual design process, called as Place and Route (PnR). Here multiple iterations are performed to efficiently place the cells, macros, IPs and power domains so that there should be no congestion, timing violations, DRC errors, health check errors and routing errors. The main aim is to make design routable while meeting all the factors of a good chip which can fullfill the needs of the customer. After successful designing of the block, several blocks are combined and integration of all the blocks takes place. In this process the ports of adjacent blocks are connected for power, ground, signal and clocks. After successful integration, the final GDSII (Graphic Database System) file is generated which will be sent to the fabrication unit. The machines at the fabrication only understands this database, and according to this data, the floorplan, placement, clock tree synthesis, routing for power, clock and signal will take place on the respective metal layers. At the end physical checks are performed on the chip and it is shipped to the customers.

2.1.6 Fabrication

Once the layout and verification of the design are clean, the design can be sent to fabrication unit. The layout data is given as input to the fabrication in a tape, the event at the exit of the layout is called as Tapeout. The data obtained in the tape from the layout is converted into photo-lithography masks for each and every layer. Masks identify spaces on the wafer, where certain materials need to be deposited, diffused or even removed. Silicon crystals are made and sliced to wafers. Near to perfect kind to polishing of the wafer is required as the modern day VLSI devices have very small dimensions. There are several steps in fabrication process, mainly involving diffusion and deposition of various materials on the silicon wafer. One mask is used in each step, and several dozens of masks are used in fabrication process.

A wafer can be as large as 20 cm in diameter and hundreds of chips can be made out of it. Before the mass production of a chip, several prototypes are made and tested.

2.1.7 Packaging Testing and Debugging

Finally, the wafer is fabricated and diced into individual chips in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in Printed Circuit Boards (PCBs) are packaged in Dual In-line Package (DIP), Pin Grid Array (PGA), Ball Grid Array (BGA), and Quad Flat Package (QFP). Chips used in Multi-Chip Modules (MCM) are not packaged, since MCMs use bare or naked chips.

2.2 Stages in Physical Design:

The main steps in ASIC physical design flow are:

- Design Netlist
- Floorplanning
- Placement
- Clock-Tree Synthesis (CTS)

- Routing
- Physical Verification
- GDS II Generation

The technology libraries used in ASIC physical design process are provided by the fabrication unit, which are classified as the minimal feature size. The sizes are 2 μ m, 1 μ m, 0.5 μ m, 90nm, 45nm, 18nm, 14nm, etc. they are also classified on the manufacturing process, which can be : n-well, twin well process and SOI process

Following are the steps in physical design flow, shown in the figure 2.3.

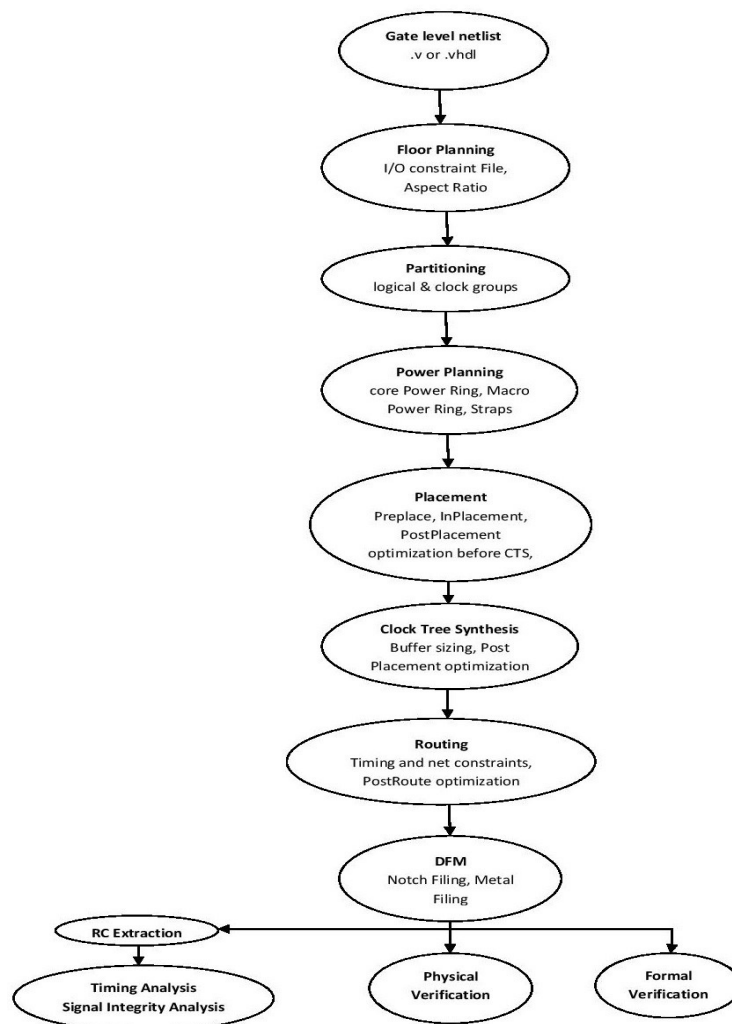


Figure 2.3: Main Stages in Physical Design Flow [1]

- **Design Netlist :** Netlist is generated after the synthesis is done. The netlist has the information of the cells being inserted in the design, connection between the cells, connection with the power supplies. Tools used for synthesis are:
 - Cadence RTL Compiler/Build Gates
 - Synopsys Design Compiler

During synthesis the timing constraints, area constraints, location of macros, ports, voltage areas are applied, so that the design should meet the speed, area and functionality . once all these thing are verified in synthesis and synthesis is done without any error, physical design stage starts.

- **Floorplanning :** This is the first step in physical design. Here, the structures which should be placed together, so that the timing and the functionality should meet, are identified. During floorplanning, halo cells, power switches are inserted and power straps are also created. The power switches are connected in the daisy chain fashion. The basic concept of putting everything close to everything is followed. Floor planning is defined as taking account of macros used in the design, memory, other IP cores and their placement needs, the routing possibilities and also the area of the entire design. There is a trade-off between the area and speed. If the design is optimized for minimum area, that means we are using fewer resources and thus higher speed of the system can be attained.

The floorplanning looks like this as shown by figure 2.4 and figure 2.5 .

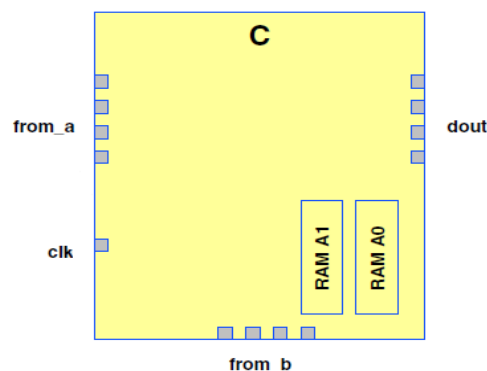


Figure 2.4: Floorplan [3]

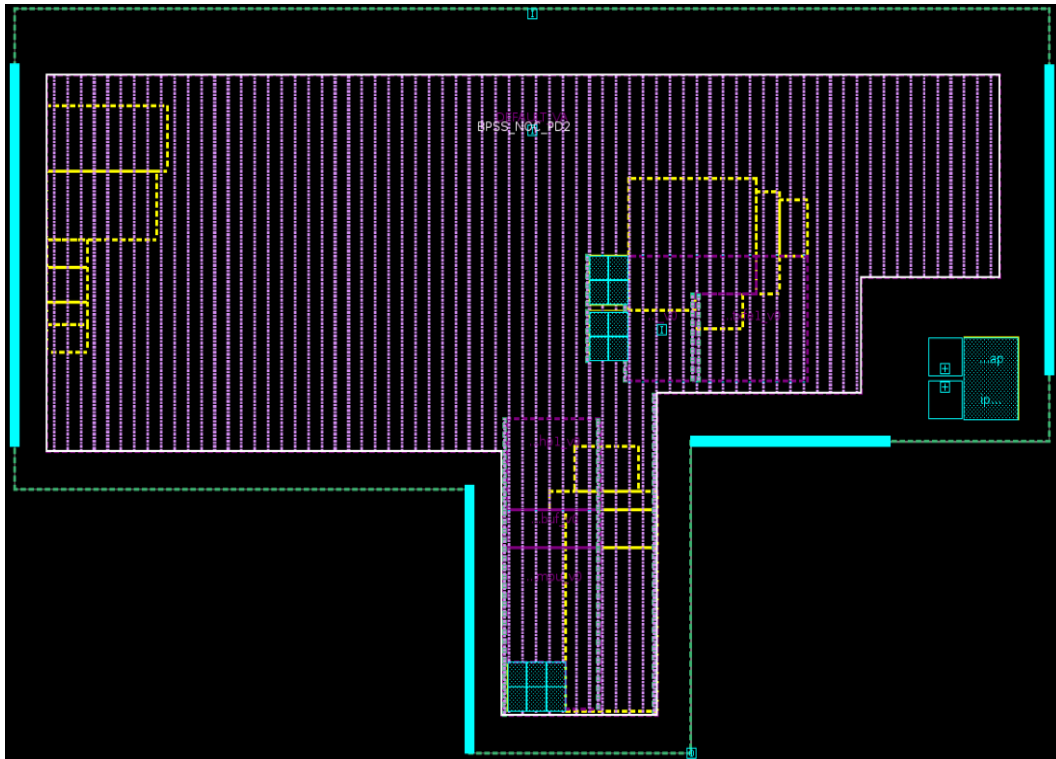


Figure 2.5: Floorplan

As a general rule, data-path benefit most from floor planning, and other logics like state machines, or some random logic are placed to the left section of the place and route software.

- **Placement :** Before the placement, all Wire Load Models(WLM) are removed. Placement uses RC values from Virtual Route(VR) to calculate timing. Here the actual cells will be placed in the design. During this stage, tap cells, fiducial cells, spare cells and bonus cells will be placed. After the cells have been placed, congestion, timing, area utilization, cell density, pin density are checked.

The cells can be seen as placed within the site row as shown in figure 2.6 .

Placement can be done in three phases:

- Pre-placement optimization : In this process optimization happens before netlist is placed. In this process high-fan out nets are collapsed downsizing the cells.
- In placement optimization : In this process logic is re-optimized according to the VR. Cell bypassing, cell moving, gate duplication, buffer insertion, etc. can be performed

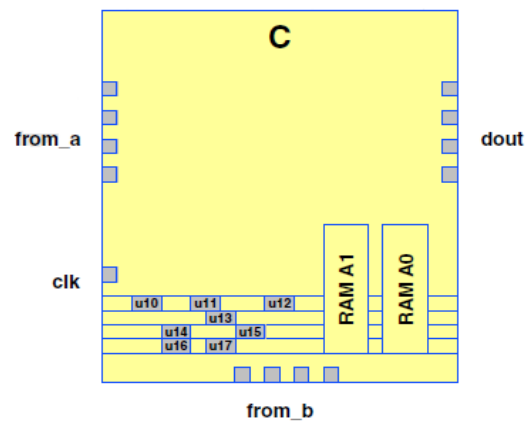


Figure 2.6: Placement [3]

in this step.

- Post Placement optimization : Netlist is optimized with ideal clocks before CTS. It can fix setup, hold violations. Optimization is done based on global routing.
- **CLOCK TREE SYNTHESIS (CTS)** : Till placement the delays which are considered for timing are ideal clock delays, which means that the delay of wires is considered as 0. Now, in placement the cells have been placed, and in CTS, the clock is implemented, buffer insertion, gate sizing and other optimization techniques will be implemented on the data and clock paths. Clock net is same for all the instances. A single clock net will connect all the synchronous elements in the design, no matter what is the count of instances. The clock network looks like this till CTS with ideal clocks as shown in figure 2.7

As we can see, only 1 driver is driving so many loads, which is impractical because one driver can drive so many flops. The ability of a driver is determined by its drive strength which is also related to the distance between the driver and driving buffer and the number of loads. Apart from the drive strength, balancing of the clock tree is also very important to make the clock skew value as 0. After the clock tree synthesis, clock tree along with buffers looks like this as shown in figure 2.8.

Few parameters that need to be in mind during CTS are:

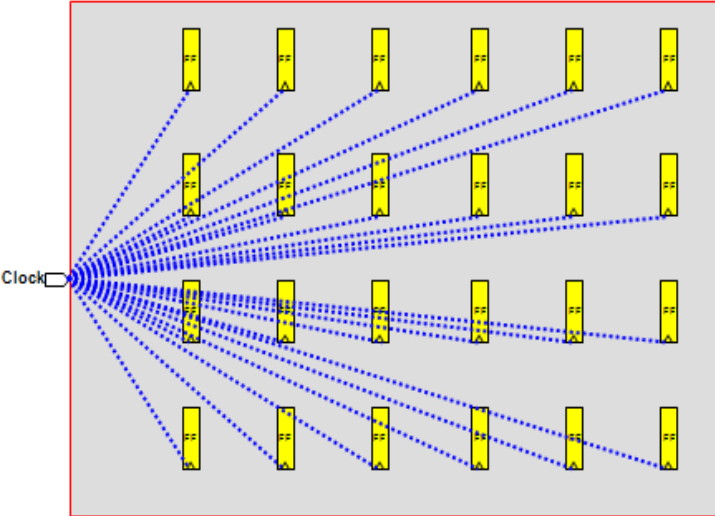


Figure 2.7: Clock Network Before CTS [3]

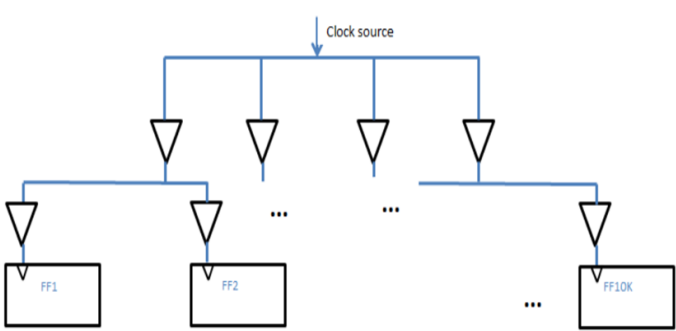


Figure 2.8: Clock Tree after CTS [3]

– **Skew** : Skew is defined as the delay in the clock network delay in a clock tree. This is the most important goal of the CTS, if skew will be there, loss of information will be there and purpose of the chip will not be fulfilled.

- * Clock Source : It is the point from where the clock propagation will start in our circuit.
- * Clock Sink : Clock sinks are the point which will receive the clock, generally they are the clock pins of our synchronous elements such as flops.

In the above picture, the delay from the clock source to the clock sink are shown. The skew is the difference between the maximum delay to the flop/instance from

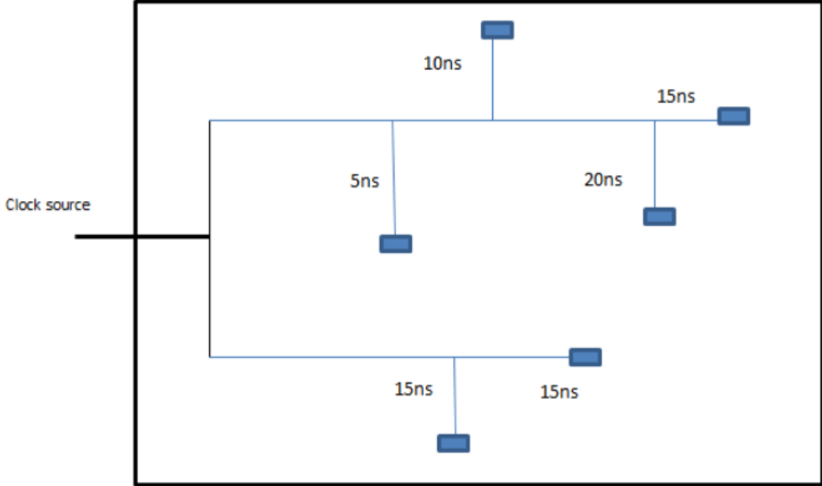


Figure 2.9: Clock source and sink [3]

the source and minimum arrival time from the source and the sink.

In this case $Skew = 20ns - 5ns = 15ns$

The aim of the CTS is to reduce the skew in the circuit to be close to zero, which means that the every instance of the clock tree should get the clock at the same time.

- **Power :** Major part of the design power is consumed by the clock, as the power consumed in the clock network is determined by the wire length, width and the switching activity (which is high as clock toggles everytime). To reduce clock power dissipation, clock gating is used.

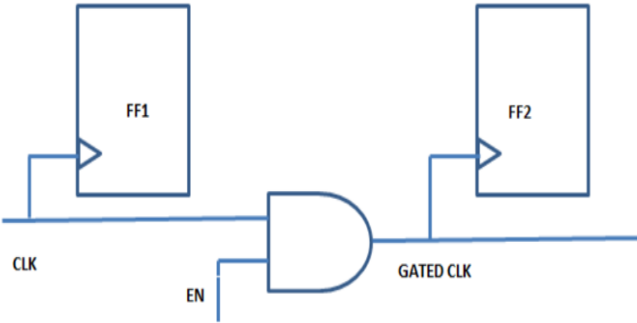


Figure 2.10: Power Consumption in Clock Net [3]

In the figure 2.10 the flip flop FF1 get the ungated clock and the FF2 get the gated clock, the EN(enable) signal is controlling the duration of the clock provided to the FF2 flip flop.

- **Routing :** The process of routing will trace the precise path for interconnection between the pins of standard cells, macros, boundary cells and pad cells. The EDA tool has the information of the location of the blocks, pins of blocks, cells , macros and I/O pads at the chip boundary. The netlist has the logical connectivity, and the physical connection is established in the routing stage through the routing of metal layers and inserting vias. The electrical connection is established by some rules known as “Design Rules”. It is essential that :

- a. Tool should make all connection defined in the netlist i.e. the design should be cent percent routable. (no LVS errors)
- b. No Design Rules should be violated while routing (no DRC errors)
- c. Timing of the design should be met.

Figure 2.11 shows scenario after routing.

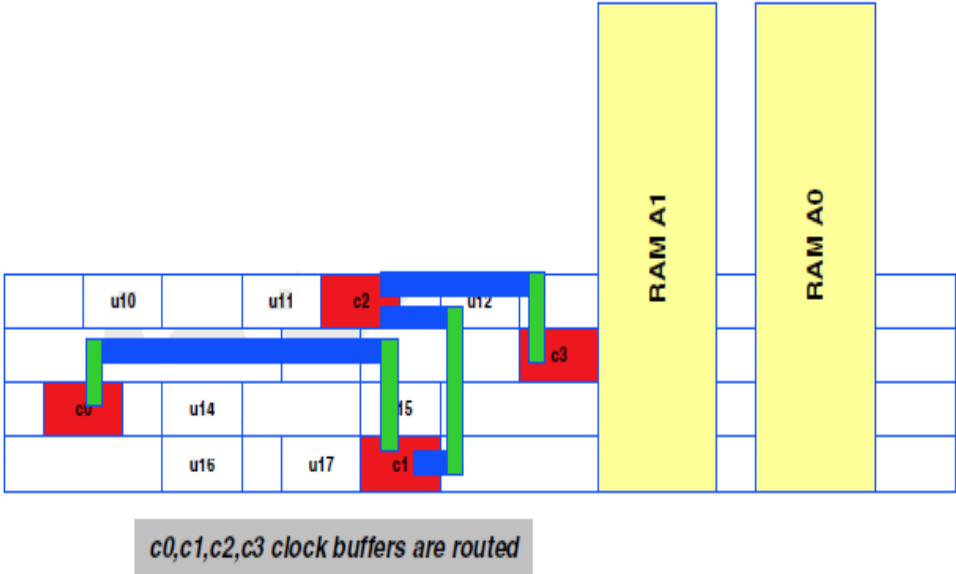


Figure 2.11: Routing [3]

The techfile which is provided by the fabrication unit, contains parameters for each layer, for each layer Minimum spacing, minimum width, minimum area are defined, which vias can be placed between each layer. If any of these parameters like spacing, width, via size etc are violated for any routing the tool does, you will get a DRC error.

2.3 Sign-off Checks in Physical Design:

Signoff (also written as sign-off)[6] checks is the collective name given to a series of verification steps that the design must pass before it can be taped out. This implies an iterative process involving incremental fixes across the board using one or more check types, and then retesting the design. There are two types of sign-off's: front-end sign-off and back-end sign-off. After back-end sign-off the chip goes to fabrication. After listing out all the features in the specification, the verification engineer will write coverage for those features to identify bugs, and send back the RTL design to the designer. Bugs, or defects, can include issues like missing features by comparing the layout to the specification, errors in design. When the coverage reaches a maximum then the verification team will sign it off. By using a methodology like UVM, OVM, or VMM, the verification team develops a reusable environment. Nowadays, UVM is more popular than others.

After the apr (automatic place and route) stage a lots of check are made on the design to validate the design in terms of timing, functionality, low power implementation, layout checks, design rule checks, power distribution checks etc. lot of third party tools are used for this process.

Here we will have a look on the various sign-off checks which are common to all the technology nodes in physical design of an SoC.

- **VCLP** - Verdi Conformal Low Power [4]

VC LP is a multi-voltage, static low power rule checker that allows engineers to rapidly verify designs that use voltage control based techniques for power management. VC LP is part of the Synospys Eclipse Flow[4].

VC LP also helps in pipe-cleaning the power intent of the design that is captured in IEEE 1801 Unified Power Format (UPF) before such intent is used as a golden reference for implementation and other verification tools. Further, VC LP verifies the implemented power-intent

later in the design flow.

VC LP is integrated with Verdi to provide designers and verification engineers access to the combined power of low power specific debug features and use Verdi's de facto industry-standard workflow, interface and powerful debug capabilities.

Features of VCLP

The key features and benefits of using VC LP for low power static verification in a typical design flow are as follows[4]:

- Power Intent Consistency Checks Performs syntax and semantic checks on the UPF that help validate the consistency of the UPF before starting with the implementation.
- Signal Corruption Checks Detects the violating power architecture at the gate-level netlist.
- Structural Checks Validates insertion and connection of special cells used in low power design such as isolation cells, power switches, level shifters, retention registers, and always-on cells through out the implementation flow.
- Power and Ground (PG) Checks Check the PG consistency against the UPF specification for power network routing on physical netlists.
- Functional Checks Validates the correct functionality of isolation cells and power switches.

Flow for VCLP checks

- **RTL Level Low Power Checks** : At RTL-level, the VC LP UPF checks help in identifying power intent issues early in the design lifecycle and enable us to arrive at a clean UPF before starting the design flow. The VC LP UPF checks ensure that the UPF is complete and the design conforms to all the isolation and level shifter rules for all power-modes[4].

The flow at RTL level looks like as mentioned in figure 2.12

- **Netlist Level VCLP Checks** : At Netlist level, the VC LP UPF and functional (architectural) checks ensure that the netlist is consistent with UPF in structure and function.

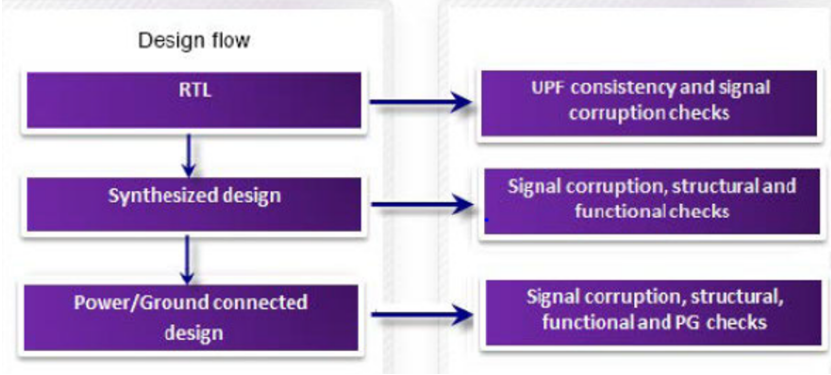


Figure 2.12: VCLP Flow at RTL[4]

The UPF checks ensure the design instances are consistent with UPF. The architectural checks ensure that the implemented design is functionally correct. There might be cases where the design is structurally correct but functionally incorrect. The VC LP architectural checks identify these low power functional issues even though the implemented design might be structurally correct. The VC LP UPF and Structural checks also help in identifying implementation issues by verifying if the low power cells (ISO/LS/RET) inserted in the design is consistent with the UPF and the library. The other VC LP functional checks include Analog checks, Inout checks, Bias checks, Diode checks and so on[4].

The flow at netlist level can be summarized as shown in figure 2.13

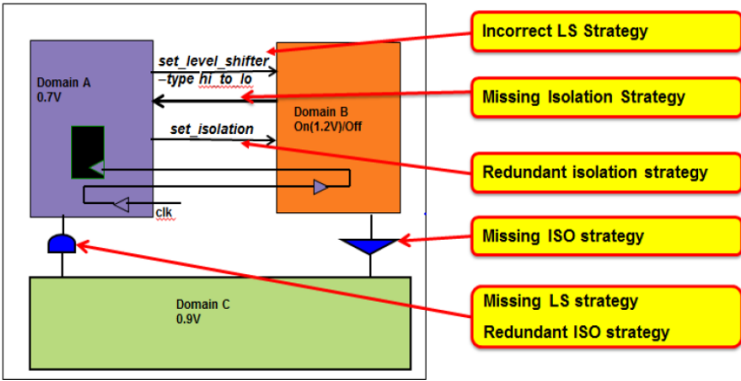


Figure 2.13: VCLP Flow at Netlist Level[4]

- **PG Level VCLP checks** : The VC LP Power Ground checks help validate the power network implementation by verifying if the Power/Ground pin connectivity in the post-

layout design is consistent with UPF and cell library[4].

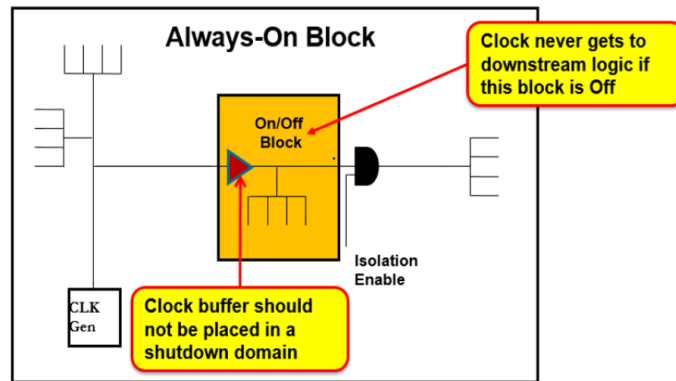


Figure 2.14: VCLP Flow at PG Level[4]

- **FEV/LEC** Functional Equivalence Check/Logical Equivalence Check.

The FEV is a tool that verify RTL, gate, or transistor-level designs. As part of the functional verification platform, FEV gives us the complete equivalence checking solution available for verifying complex system-on-a-chip (SoC) designs from RTL to layout. It verifies the widest variety of circuits, including complex arithmetic logic, datapath, memories, and custom logic. Conformal has high-performance, high-capacity, and excellent debugging capabilities. These features are combined in an integrated environment. Figure 2.15 shows the flow diagram of the FEV[5]

- **FEV Features[5]**

FEV incorporates many features that streamline and authenticate the design process, while giving you flexibility.

- * Supports Full-Chip Verification

FEV has excellent processing speed that significantly reduces verification time for high capacity, high-complexity, full-chip designs.

- * Supports Multiple Design Formats

FEV supports Verilog®, VHDL, SPICE, EDIF, and NDL design formats.

- * Supports Standard Library Formats

FEV supports Verilog simulation libraries and the Synopsys® Liberty™ Format libraries.

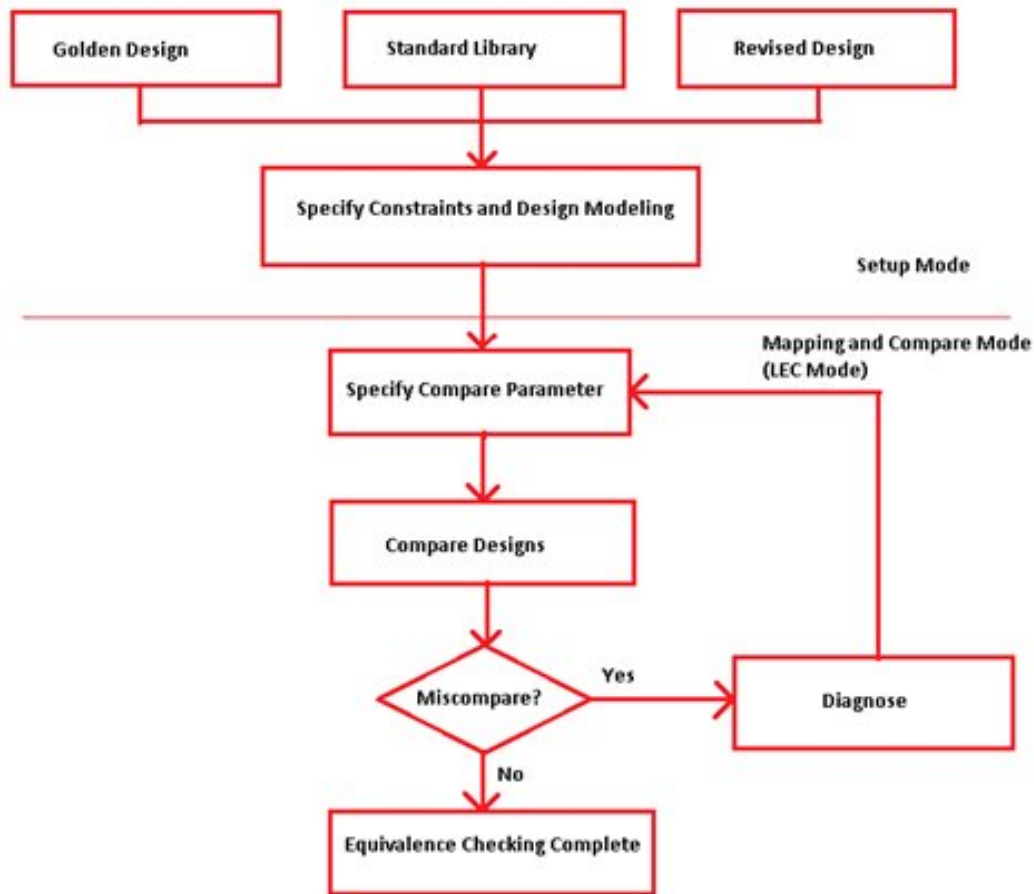


Figure 2.15: Flow Diagram for FEV[5]

- * Employs Verilog/VHDL-RTL and Transistor Function Abstraction Conformal has a built-in Verilog/VHDL-RTL and transistor function abstraction engine that lets you verify Verilog/VHDL-RTL, gate, or transistor level designs.
- * Employs Advanced, Automatic Mapping Conformal contains advanced and proprietary sequential element mapping algorithms that identify corresponding sequential elements automatically with minimal user resources. This feature relieves you of the tedious job of specifying corresponding flip-flops and latches.
- * Employs an Efficient and Effective Comparison Engine Conformal has a superior formal comparison engine to ensure successful verification of non-similar designs with different hierarchical structures. Conformal contains a unique correlation learning technology that effectively explores both structural and functional relationships of the logic in two designs and dramatically reduces the verification run

time. This technology does not require high memory use and is very effective for both similar and dissimilar designs.

- * Includes Automatic Diagnosis When a logic mismatch is found, designers find that it is absolutely essential to be able to quickly locate the source of functional differences. Conformal automatically diagnoses functional differences, narrowing them to a small number of possible locations in the design. This feature helps you identify and effectively correct problems and reduce debugging time.
- * Includes Integrated Debugging Conformal has extensive gate reporting integrated with the schematic viewer. This feature gives you flexibility and immediate feedback for debugging and diagnosis.

Chapter 3

How we optimized the Design

3.1 Optimization in each Steps

For the explanation of optimization of our design, we have divided the optimization in different stages in physical design.

3.1.1 Floorplan

Adjustment in the boundary of the design, creation of voltage areas, placement of macros, analog cells, ports, changes in the boundary of the design. Checking of the health check reports, insetion of power switches, insertion of power rails on the power switches.

The floorplan is the very important step in the physical design, a good floorplan can help the design to be congestion free, makes it routable and helps in meeting the timing.

After creating voltage areas, bounds and the final co-ordinates of the floorplan of our design looked like this as shown in figure 3.1.

Issues faced in floorplan and their solution :

- **Issues related to Libraries** - Librarires are provided by the fabrication units. During the design process, libraries are updated on routine basis, if we take the older libraries then the cells, nets to which that library belongs will not be inserted properly and it may have incorrect pin connections. So it is a good practice to take the latest libraries for all the cells, macros, IP, ports etc.

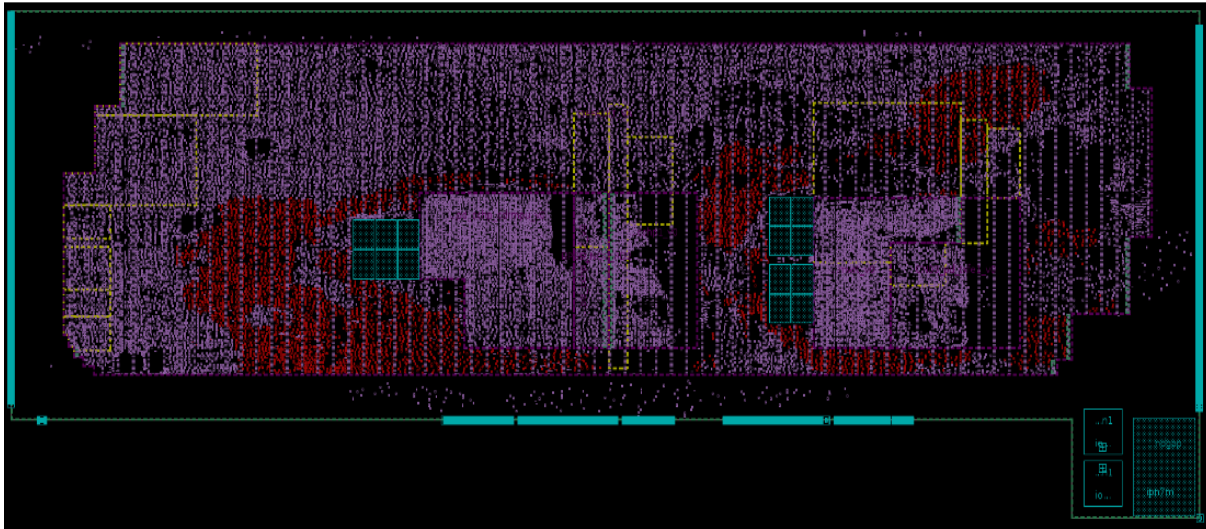


Figure 3.1: Floorplan with final constraints

- **Die area, voltage area, bounds, ports issues** – The die area that is provided by the top level integration team is used in floorplanning, it also takes 1 or 2 iterations. Voltage area is created by the block owner, based on the location of the ports, Network Interfacing Units (NIU), clock adapters, congestion in the design. Bounds are placed in the design for the purpose of keeping all the cells, which are communicating to a specific block via specific ports. It helps in meeting the timing, as all the cells between the starting point and the ending point are close to each other in a small area.

3.1.2 Placement

Checking the placement of the standard cells, power switches, tap cells. After the cells have been placed, the congestion of the design has to be checked, cells density, pin density and timing has to be checked.

After the placement of the standard cells the placement and the congestion in the design looks like as shown in figure 3.2

Issues faced during placement:

- **Timing** - The timing of the design is checked, according to that cell placement and port placement is done to meet the timing. For the cells falling in one hierarchy and all talking to some specific cells, ports, macros the bounds are created. The bound will keep all the cells

- Copy.png

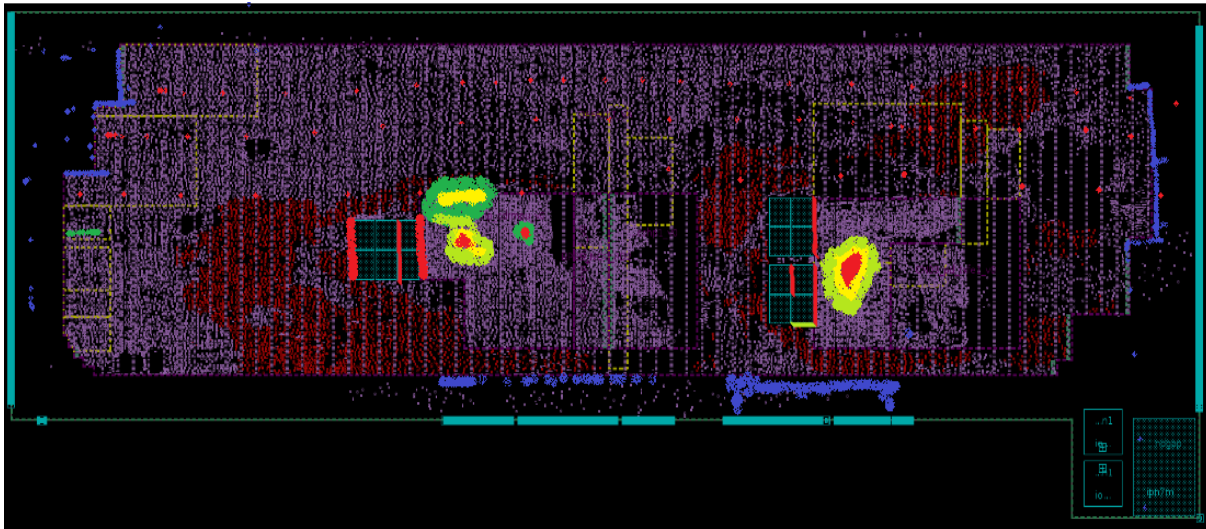


Figure 3.2: Congestion after placement

in one area and help in meeting the timing. But if the bound is small and the cell density in that area is high, the congestion will be observed there. So make the bound such that, it has around 50 percent utilization.

- **Congestion** - It is a situation when the total number of wires/nets that are run through an area is more than the actual capacity of that area. In this case, the nets cannot be routed as nets are more than the capacity of that area. If this problem is not solved in placement stage, it will show the shorts during route stage. So the solution is, first analyse the cause of the congestion, whether it is high cell density, high pin density, congestion is observed at sharp edges.
 - If the cell and pin density is high, implementing keepout margins, cell padding, partial placement blockages, limiting the cell density at that place will help in reducing the congestion.
 - If the congestion is observed at sharp edges or L shaped edges, then we have to limit the cell density there, as all the net which are coming will bend and the vias at that place will occupy more space. So if we place hard or partial placement blockage at that place, it will reduce the cell density, so pins of those cells will also be less, and hence congestion is reduced at that place.

- In this design, the red dots which we see, is the congestion overflow of 7, they are in straight line and they were observed because the feedthrough nets and the always on buffers were placed in the gated domain(power domain which is not always on). Due to that also congestion was observed. So for that, the feedthrough nets and always on buffers were placed outside, in the channel left alover the design. This helped in congestion reduction.

3.1.3 Clock Tree Synthesis

This stage will create the clock tree, clocks will be provided to the flip-flops, cells, buffers, inverters etc. After CTS, the actual timing of the design will come out as till CTS, the delays of clocks are considered as ideal, only the clock uncertainty is the factor which is giving some pessimistic results. The clock tree used to look like as shown in figure 3.3.

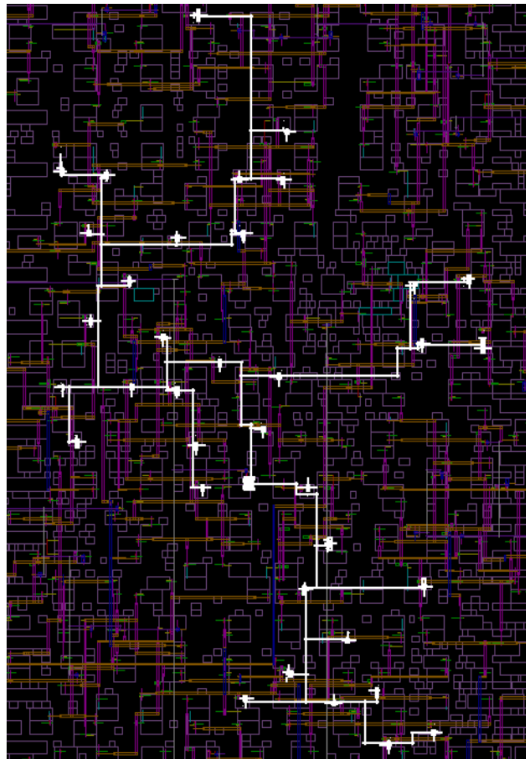


Figure 3.3: Clock Tree after CTS

Problems faced during CTS:

- **Bad Timing** If we are happy to look at timing till placement, and when we look at timing after CTS, then we might get shocked, as the CTS will show the exact timing after clock has been implemented in the design. There we will see a lot of setup , hold , data transition , clock transition, data capacitance, clock capacitance violations in the design.
 - The main reason for that bad timing will be mainly the unbalanced clock skew.
 - For hold downsizing the buffers in the path will in reducing the hold violations, for setup we have to make the data to come fast at the pin before the clock could arrive. So for that we have to remove some buffers in the data paths so that the delay can be reduced[7].
 - For the transition delays, the rise time and the fall time of the cells, flip-flops will be affected. To retain the originality of the signal, we have to add buffers as per the distance of the clock source and the destination to solve this issue[7].
 - For Clock and Data capacitance, the driving strength of the cell reduces as the distance increases, so we have to add buffers to provide the drive strength to the driving cell to drive the load[7].

3.1.4 Route

Till the CTS the power nets, clock nets were routed in the design, but after the routing is done the whole design will have the nets including data nets, signal nets, clock nets and the power nets. More clear picture of the design will come to us. The function of the route engine is to do routing in such a way that the clock tree is not touched to do the signal and the data net routing. During routing all the NDRs (Non Default Rule) will be implemented such as, moving some nets to the higher metal layer, double wide spacing, clock shielding etc. But the NDR must not create more DRCs.

Problems faced during route:

- **DRC** Design Rule Checks are physical checks of metal widths, spacing, pitch between the metals, shorts, opens, via spacing rules, minimum spacing rules. If any of these violations are being reported, then the design cannot be taken to the other sign-off checks, as we will

be playing with the nets. Some nets will be detoured, so timing can be affected, sometimes even connecting and disconnecting may even hamper the functionality. So after every iteration of DRC fixing, LEC (Logical Equivalence Check) to check the functionality and STA (Statistical Timing Analysis) must be performed to check whether the tool have not messed up with the signal and timing nets.

- **Timing** Once again if timing is still not met after route stage, then the violations of the timing must be fixed as mentioned in CTS stage.

After doing all these fixes and analysis, we can assume that our design is having very good floorplan, design is free from congestion, timing of the design is met as mentioned as per the max operational frequency, clock tree has as much low skew as close to zero, design is routable, passing the LEC checks STA checks, VCLP checks, LVS checks and PDN (Power Delivery Network) checks design is ready to generate the GDSII (Graphic Design System), which will be sent to the fabrication unit for the chip to be brought out as a product.

Chapter 4

Results

Here, in one project we were facing huge congestion issues of about 10 percent, the results and inferences are as follows.

- Partial Placement blockages

After keeping the blockage percentage of 40 percent, in the area where we were seeing congestion it reduced to the maximum overflow of 3(green), where it was earlier overflow of 7(red).

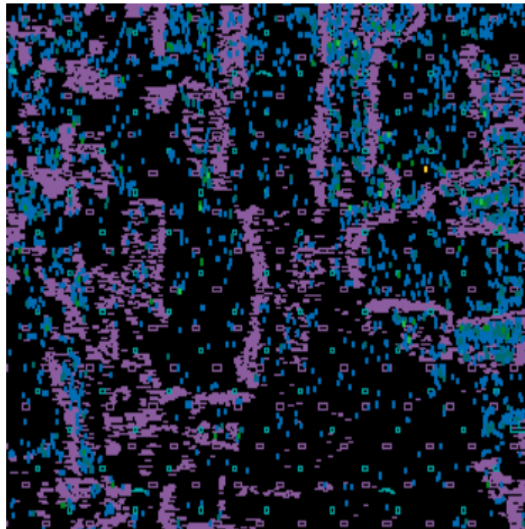


Figure 4.1: Results after Partial Placement of 40 percent

- Keepout Margins

1 experiment was done with the same congestion, with applying keepout margins, it was due to high pin density in 1 area and overflow was of 7, after applying keepout margins/cell padding of 1 1 1 1, overflow reduced to 3, but the overflow density increased as shown in figure 4.2

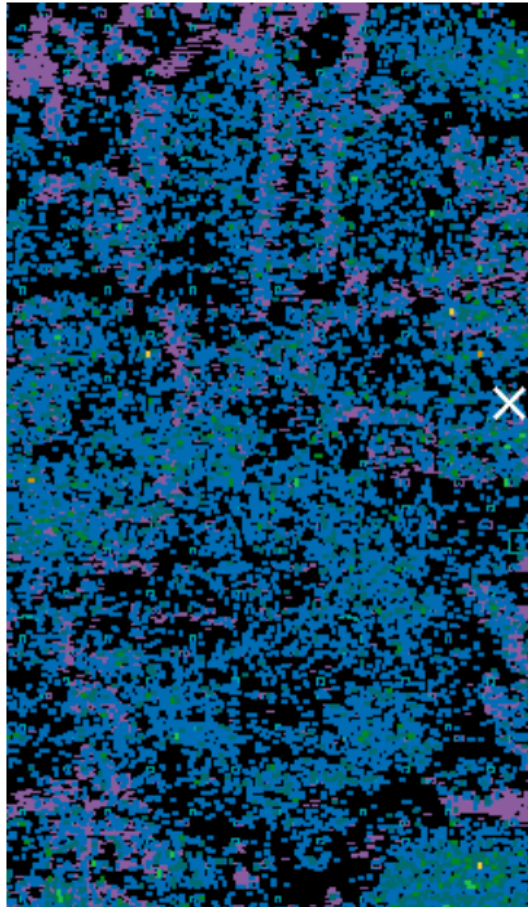


Figure 4.2: Results after Keepout Margin of 1 1 1 1

- Increased Congestion Effort Increasing the congestion effort is usually not recommended, because it places the cells so far apart that it messes up with the timing. Unless and until all the solution are not helping out, do not use increase congestion effort.

So after implementing these solutions and routing of feedthrough nets and always on buffer in the default power domain, we got the congestion of our design in control of about 0.14 percent, as shown in figure 4.3. The blue colour dots which we can see are the overflow of 1, and the actual numbers are shown in figure 4.4.

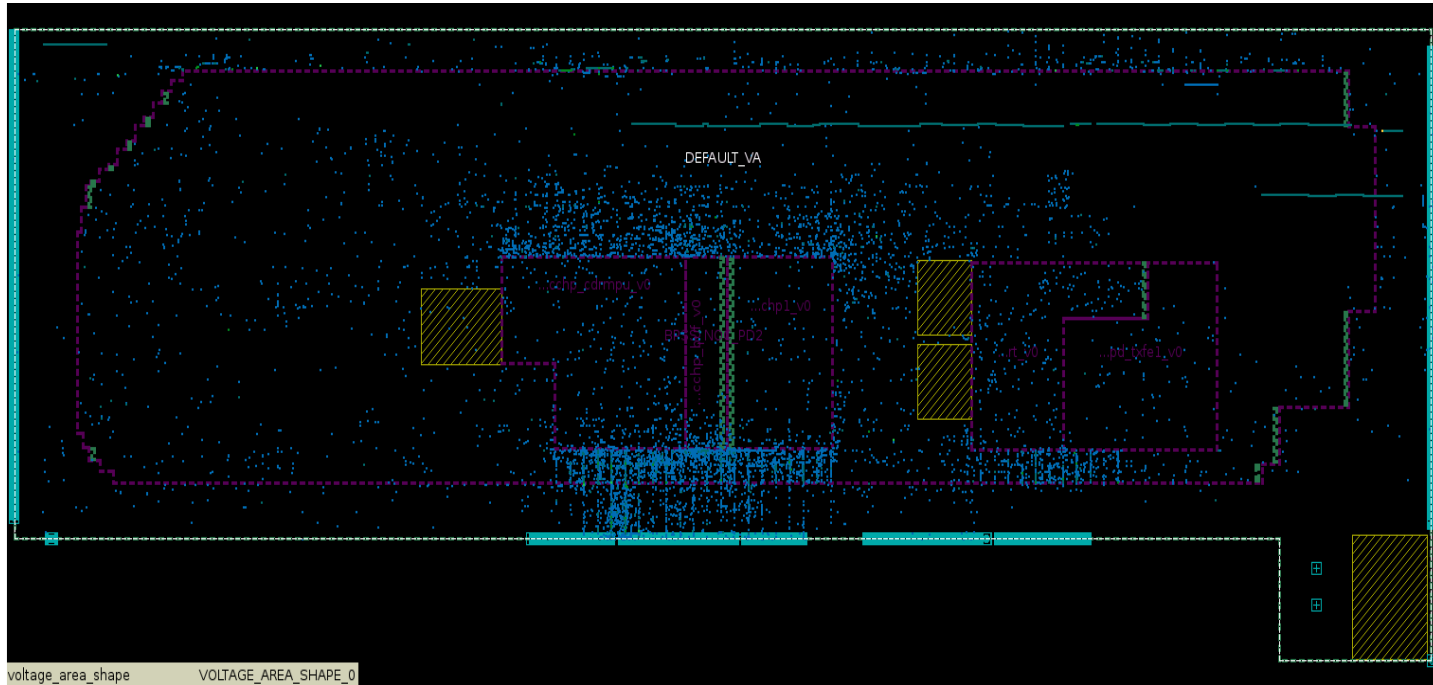


Figure 4.3: Final Congestion in design

Layer Name	overflow		overflow (%)	# GRCs has	max overflow
	total	max			
Both Dirs	14493	4	11871 (0.14773%)	3	
H routing	11411	3	8928 (0.22221%)	32	
V routing	3082	4	2943 (0.07325%)	3	

Figure 4.4: Final Congestion numbers

Chapter 5

Conclusion

Technology is scaling down very fast, designs are getting more and more complicated, so as their validation, implementation and verification. More and more sophisticated designing tools are needed to take account of all the rules related to the lower technology nodes. With the increase in complexity of the design, floorplanning, timing, congestion, routability are main concerns, which are growing very fast and difficult to resolve.

A good floorplan will be very help full for the rest of the stages in the design, it will also help to reduce the time to implementation as ports placement, macro placement, bounds and blockage creation will be according to the nearby partitions. This will help in meeting the timing, once timing is met most of the issue can be resolved easily.

Congestion will also not be a big problem, using methods to resolve congestion such as placement blockages(soft, partial, hard), keepout margins, limiting cell density, proper pin spacing between macros, keeping the feedthrough nets and always on buffers in the default domain(if they are not timing critical).

As for the DRC's most of them can be resolved by small detours. Once good results are obtained, then the ECO's will start in the design for timing, functional ECOs, Power Delivery Network(PDN), Layout versys Schematic(LVS), Low power implementation (VCLP).

With the proper knowledge of the problems and solutions and little bit of experience a successful tapeout of the design can be done.

Bibliography

- [1] Pravriti, *VLSI Design Flow*, 2018.
- [2] N. A. Sherwani, “Algorithms for vlsi physical design automation,” in *Test Conference, 2005. Proceedings. ITC 2015. IEEE International*, pp. 10–pp, IEEE, 2015.
- [3] Synopsys, *Physical Design Flow*, 2016.
- [4] Synopsys, *VCLP Low Power Signoff and Static Verification*. Synopsys, 2018.
- [5] Cadence, *Conformal LEC User Guide*. Cadence, 2018.
- [6] Signoff-scribe, *Sign-off Checks*, 2017.
- [7] J. Bhasker and R. Chadha, *Static timing analysis for nanometer designs: A practical approach*. Springer Science & Business Media, 2009.