AUTOMATION OF RTL TO GDS II DESIGN FLOW USING SCRIPT'S METHODOLOGY

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

VLSI DESIGN AND EMBEDDED SYSTEMS

Submitted by:

VISHESH JAIN

2K18/VLS/18

Under the supervision of

Dr. GURJIT KAUR

Department of Electronics and Communication Engineering,

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

July 2020

Department of Electronics and Communication Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I **Vishesh Jain**, Roll **no 2K18/VLS/18** of M.Tech VLSI Design and Embedded Systems, hereby declare that the dissertation tilted, "**Automation of RTL to GDS II flow using script's methodology"** which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Masters of Technology is original and not copied from any source without proper citation. This work has not previously formed for the award of any degree, diploma associateship, fellowship or other similar tile or recognition.

Place: Delhi VISHESH JAIN

Date:

Department of Electronics and Communication Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project titled "**Automation of RTL to GDS II flow using script's methodology**" which is submitted by **Vishesh Jain**, Roll No 2K18/VLS/18, Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the him under my supervision. To the best of my knowledge this work has not been submitted in part or full for any degree elsewhere.

Place: Delhi

Date:

(Dr. Gurjit Kaur)

SUPERVISOR

ACKNOWLEDGEMENT

A successful project can never be prepared by the efforts of the person to whom the project is assigned, but it also demands the help and guidance of people who helped in the completion of the project.

I would like to thank all those people who have helped me during my study. With profound sense of gratitude, I thank Dr. Gurjit Kaur, my Research Supervisor, for his encouragement, support, patience and his guidance in this research work. I take immense delight in extending my acknowledgement to my family and friends who have helped me throughout the journey.

VISHESH JAIN

Roll no: 2K18/VLS/18

M.TECH. (VLSI Design and Embedded System)

Department of E.C.E, Delhi Technological University

Delhi – 110042

ABSTRACT

As technology is advancing from submicron technologies to nano-meter technologies, design complexities and challenges also increased proportionally. As a result, engineers have lot of things to concentrate and the design needs a lot of attention so that it can be closed in the expected time.

Also engineer needs to well take care of the things like firing the Runs, choosing correct inputs, creating command files, custom reporting, and the other routine things. And, the project implementation team should ensure that the project specifications like corners derates, HVT Cells, Metal Scheme etc., are well maintained in all blocks at all levels. All these things have given rise to the deployment of a robust automated system called RTG Flow.

RTG is acronym for RTL 2 GDS which has a very robust automation that helps in the smooth flow of implementation of Library qualification, physical design flows and verification flows i.e. from the Synthesis to Tape-Out.

Basically, RTG Flow involves running lot of scripts and wrappers to do a specific task like Firing the runs, controlling them in both sequential and parallel manner, parsing the reports, comparing the runs, displaying them in web etc.,

In simple, it can do most of the tasks where the human intervention is not needed. It has been developed with the scripting languages like Make, Shell, Perl, Tcl-TK, Python. The thesis is an effort of automation efforts involved in design of an ASIC.

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS & ABBREVIATIONS

RTG	RTL to GDSII
RTL	Register transfer level
GDSII	Graphic design system
ASIC	Application specific integrated circuit
GPU	Graphical processing unit
SOC	System on chip
FPGA	Field programmable gate array
HDL	Hardware description language
DFT	Design for testability
STA	Static timing analysis
CTS	Clock tree synthesis
DRC	Design rule checks
EDA	Electronic design automation
PnR	Placement and Routing
UF	Utilization factor
IC	Integrated circuit
CLK	Clock
IP	Intellectual property
HVT	High Vt cells
PPA	Power, performance and area
Lib Gen	Library Generation

UDM	Unified database management
CI	Continuous Integration
CD	Continuous deployment
GUI	Graphical user interface

CHAPTER 1
INTRODUCTION

ASIC's are chips that are used to perform an application, e.g., GPU for running specific gaming applications only, standard applications like portable devices like micro SD card, hard disk drives ,which contains controllers that are dedicated for storage applications, special chips for automobile applications , and certain SOC used to perform complex computations. ASIC design in initial stages started with few thousands of gates only. As technology advanced the number of gates increased from few thousands to millions, following Moore's law of doubling the number of transistors. ASIC's which consists of multiple blocks like memory elements and other elements are called SoCs or system on a chip. These SOCs will be integrated further and will form a more complex SOC. The increasing complexity demands for a structured design flow and methodology for its implementation ,which can be scaled and provides flexibility to the designers for a seamless integration.

FPGA are another paradigm shift in VLSI design .FPGA are used for ASIC implementation before they are fabricated to verify the functionality of the design.The difference between FPGA and asic is that FPGA can be re-programmed multiple times unlike a design on ASIC ,which cannot be reprogrammed. Thisability to field-program the IC doesn't restrict the user to any predetermined hardwarefunction, and IC can be tweaked based on changing standards providing reduced non recurring engineering cost and significant time to market advantages over ASICs, although taking a hit on the performance and power consumption.

ASIC design can be classified as logical design and physical design.Logical design starts with HDL specification and its architecture's design captures the

HLL,power requirements and timings criteria for the chip design i.e. speed of design operation .This specification is then converted into RTL level design i.e. Register Transfer Level, which provides the abstract behavior of the design in terms of its logical operation on its signal between flops in design. This behavior is captured using hardware description languages (Verilog or VHDL).Once the functionality is written, it is verified using simulation by providing various inputs to the design and the response is captured. The objective of simulations is to validate the resulting output matches the desired outputs of the design.

For example, in adder implementation it includes two inputs and single output, the test vector would give various input combination and check whether output is sum of the input combinations. At this stage design is ready for synthesis also.In synthesis stage the RTL is converted into equivalent logical gates, which are hardware equivalent of the description. Consider the following code written in Verilog.

```
module flipflop (d, clk, rst, q)

input d, clk, rst;

output q;

reg q;

always @(posedgeclk)

if (rst)

q <= 1'b0;

else

q<= d;
```

end module

The above code will be synthesized to a positive edge triggered flip flop with synchronous reset. An HDL is synthesizable RTL, if it can be mapped to unique hardware implementation which is unambiguous implementation.In this step designers also tries to capture the design and timing parameters ,which represents high level design objectives set at chip architecture phase. Theseparameters include frequency of clock, delay and target library ,which helps synthesis tool to optimize the design to meet the requirements.

Synthesis is followed by design for testability(DFT).DFT is a technique that makes sure that that significant number of test pins are inserted to perform post silicon validation of the IC,to check it's functionality in order to make sure that faulty parts are not supplied in maket. DFT step is followed by verification of the RTL,to verify that design intent is correctly served. This process is called equivalence checking and it uses formal verification techniques.This stage design is also ready for timing analysis and timing checks are performed on the design.STA is a process of checking whether the design meets the intended timing requirements,timing analysis can be performed in two ways static and dynamic timing analysis ,in STA the input stimulus is not applied.STA engines require the timing constraints to model the chip requirement and assumptions to be made in design to meet timing requirements set at chip architecture phase.STA step is where logical design is completed, and chip is ready to go for physical design.

Physical design starts with Floor planning stage ,after primary timing analysis ,the blocks are placed with aim of optimizing area ,aspect ratio and how signals between the blocks will interact with each other. The main aim of floor

planning is to make sure that there is minimum interaction between the blocks and does not posses any difficulty at the time of routing.

These factors impact all the important timing parameters like power,area and timing and overall performance of the design. The optimal floor planning is followed by routing stage, in routing the connections between the blocks is routed. Floor planning is followed by CTS (clock tree synthesis) ,where clock distribution is done.

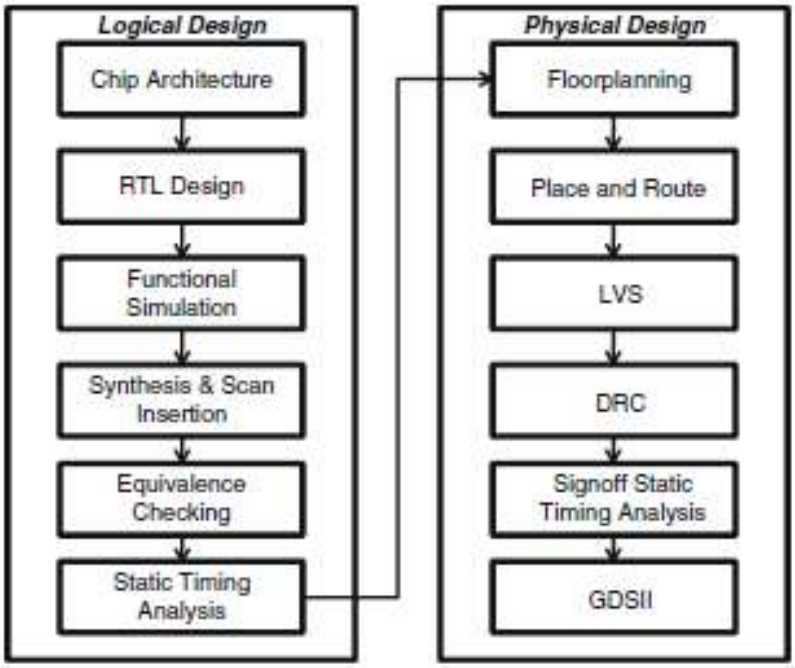


Fig 1.1 ASIC Design flow

The purpose of clock tree synthesis is even clock distribution and minimizing the skew between the different part of the design. These steps are collectively called layout design of the chip.STA checks are performed multiple times on chip while layout design to obtain actual timing analysis and assumptions made at initial synthesis phase are qualified and accordingly the assumptions are corrected. Fig 1.1 shows a typical ASIC design flow.

At layout stage of IC ,some verification needs to be ensured to ensure the design criteria parameters are met.

1.The rules laid down by foundry, where the manufacturing will be done ,are met. These rules are very exhaustive , called Design rule checks(DRC) checks

2.The Layout is matched with the netlist after synthesis stage. This stage is called LVS(layout versus schematic),where the layout is verified with post synthesis netlist.

Once DRC checks and LVS checks are clean, signoff STA is performed. It is not necessary that design will meet the timing criteria so further rigorous timing checks needs to be performed. Once post layout STA checks are passed ,GDSII of design is generated. It is geometric representation of the polygons which has actual description of the layout containing all the connectivity.The chip is manufactured according to the GDSII file being released to them. This whole flow of chip design is called RTG (RTL to GDSII) and process of releasing the GDSII file is called ***tape out*** of the design.

1.1 Motivation & Problem Overview

Physical design is a very complex and challenging design ,with evolving technologies eveyday the complexity of the design keeps on increasing manifold times.Designers are always in search of the complex challenges ahead of them in design issues.With every reduction of nanometer at technology front the complexity of design closure for implementation flowas and methodology becoe more and more complex.On top of it time to market is constant metric for success of any ASIC product keeping high pressure to meet Tapeouttimelines.While implementation teams has to focus on understanding and fixing design amd methodology issue due to heavy depencence on continuously evoving costly and

resource intensive EDA tools their majority of times goes in understanding technology from foundry ,executing runs ,waiting for results and analyzing quality metric from large log and report files to identify fixes and provide feedback while continuously fighting for machine and liscenceresources.This loop is even more complex for technologies like 16FF or lower where most of signoff methodologies and flows are not standalone checks.

The thesis is an effort on part of design automation to reduce the overall burden on design challenges and automate the test that are being run manually by the designers and consume a significant bandwidth of the design.The design quality tests are automated and the data's are extracted from logs report to ensure quality checks of the design and pass through a standard quality cheks.

1.2 Objectives

The thesis has following objectives:

1. To understand the physical design flow,various stages of design and literature study of design stages, challenges and solutions.
2. To understand the challenges involved in automated design flow, study the problems, identify the gaps and propose automation platform for the proposed solutions .
3. Study manual regression running mode of design tests, identify the gaps and provide the automated solutions to the manual running of test ,which helps in reducing the lifetime of project.
4. Creating common platform for all tests and a qualified automated test platform.
5. Analyze the results achieved by automation platform and future scope of enhancement in automation.

1.3 Thesis Organization

The organization of thesis is in five major chapters.

Chapter 1 introduces the overall design cycle of an ASIC from RTL to GDSII phase and the problem that a designer faces in a ASIC design.

Chapter 2 is literature survey of the physical design cycle. In this chapter, the physical design cycle is discussed in detail from Floor planning to DRC checks review. It gives the overview of all the design phases of the physical desing and what all the design is comprised of. The chapter has schematic visualtization of the physical desing a chip to have a clear understanding of design.

Chapter 3 introduces to the RTG (RTL 2 GDSII) automation of a design, the challenges at the design side.It discusses in detail how automation environment is created around physical design cycle to reduce the overall burden and bandwidth of a design. A highly qualified Lib Gen flow is being discussed ,which reduces the overall cycle time of the project and provides a streamline flow to the project.

Chapter 4 LibGen flow has many components of a desing qualification phase,out of many qualification, sanity test are one of the major quality parameters to check that desingers run to ensure that the design is free from any functionlaity related error and all design fucntionlaity are met before physical design of a ASIC starts.This chapter discusses about the automation achieved in sanity regression and the environment is created for automation of sanity checks. These checks are done to ensure RTL is clean and block level functionalities are met.

Chapter 5 This chapter discuss about the automation results,future scope in design automation and the impact the design automation puts on a design cycle.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

"Physical Design Flow" is the backend part of RTL to GDSII flow of VLSI design cycle. It deals with the design of the circuit on the chip. The design has a fixed flow and the series of steps are performed in an order to design the chip.

The design steps to be followed are:

1) Floor Planning

2) Placement and Routing

3) Optimize Placement

4) Clock Tree Synthesis

5) Routing

6) DRC check.

7) Parasitic Extraction

These are the main seven steps that are followed in physical design cycle of a chip design. Each step is an exhaustive design and carries of series of steps. For an overview the above seven steps are being discussed in section 2.1.

2.1.1 Physical design flow

1. Floor Planning:

The fig 2.1 depicts the basic chip layout after the floor planning has been performed. It carries a series of steps that will be discussed in section 2.2.

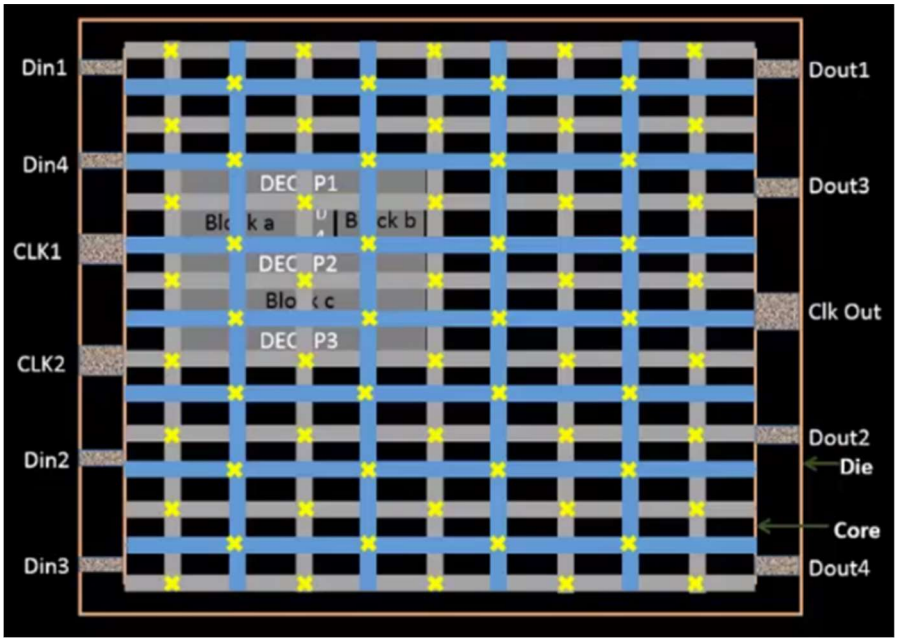


Fig 2.1 Floor Planning of a chip

2. Placement and Routing:

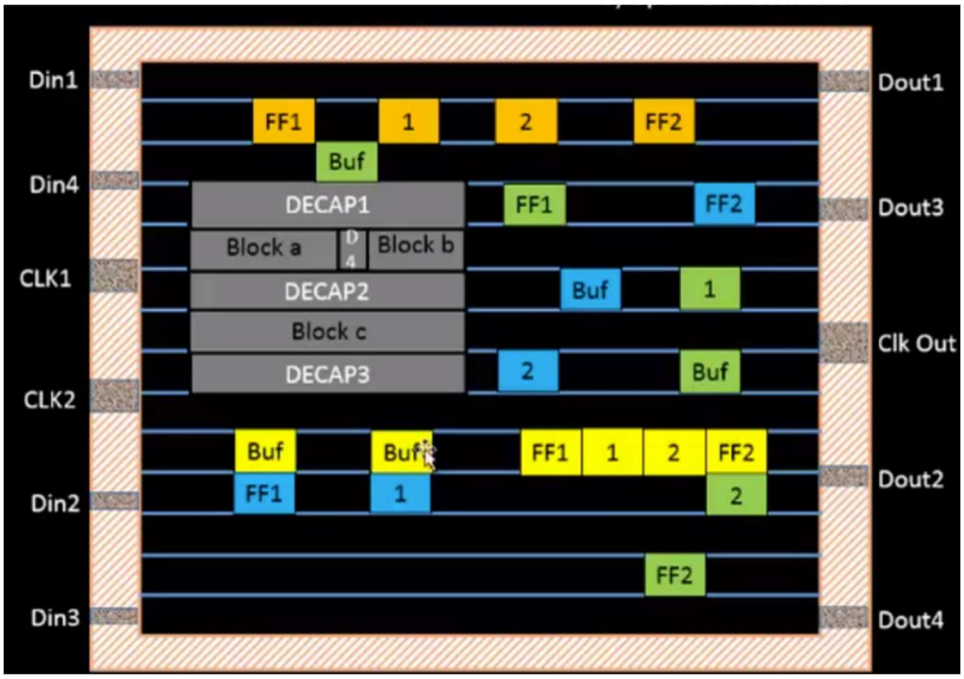


Fig 2.2 Layout of chip after placement

Fig 2.2 shows the chip layout after placement. In this a sequence of steps are followed to place the logical blocks on the chip in such a way that they are best placed on the chip. Detailed process is discussed in section 2.3.

3. Routing and Clock Tree Synthesis:

Fig 2.3 shows the layout of chip after Routing between cells and CTS has been performed.

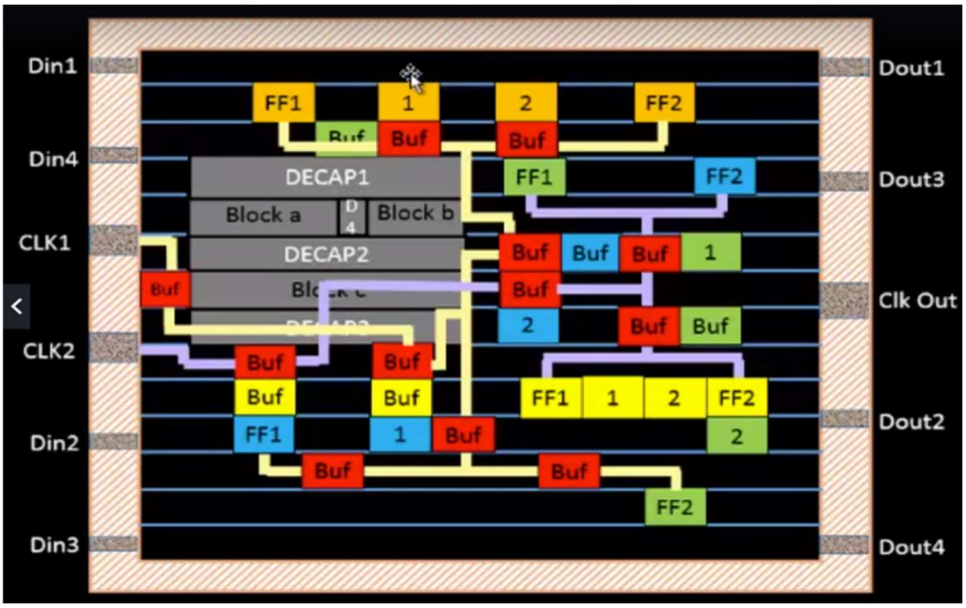


Fig 2.3 Layout of cell after routing and CTS.

1) Section 2.4 discusses about the steps in detail, various challenges and solutions in exixting literature.

2) Clock net shielding clean check and final design: After performing all the design steps clean step is done to check whether all rules are followed, and final parasitic extraction is performed. Fig 2.4 shows the layout of chip after a complete Physical Design flow.

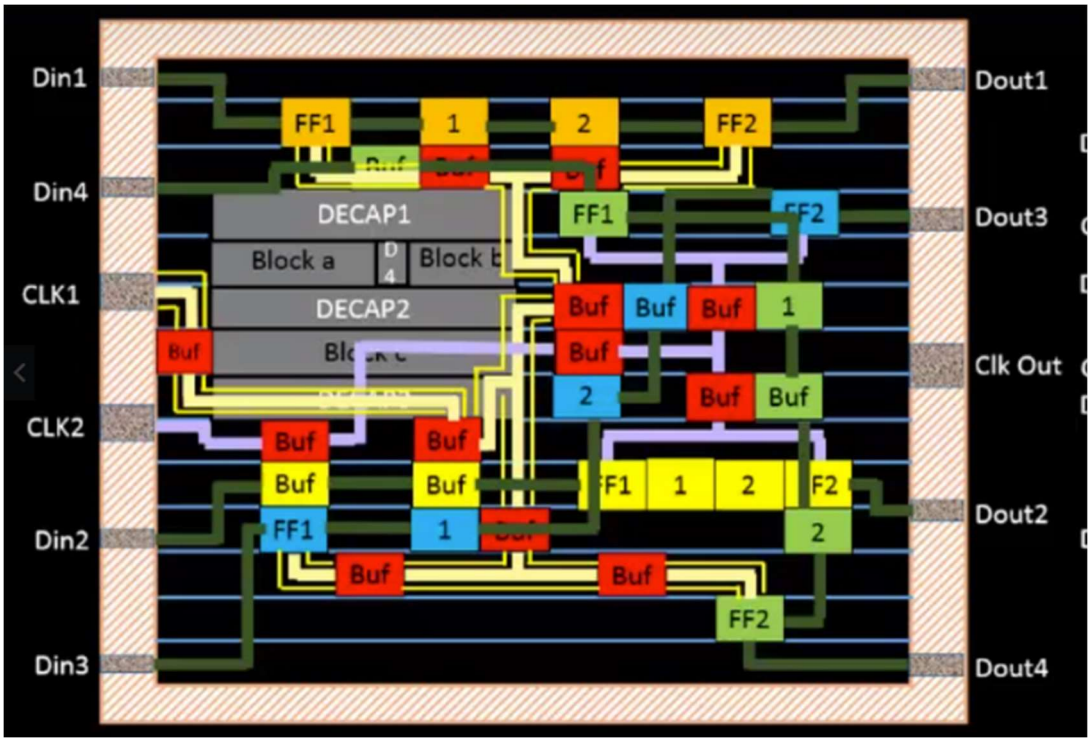


Fig 2.4 Final layout of the Cell

2.2 Floor planning

Floor planning is the first step that is being performed while designing the layout of the chip. Floor planning deals with the steps a to be followed to prepare a floorplan of chip, keeping in consideration all the design requirement.

There are series of steps that are followed in floor planning. The sequence of steps that are followed are:

1) Define width and height of core and die.

2) Define the location of preplaced cells.

3) Surround the preplaced cells with a decoupling capacitor.

4) Power Planning

5) Pin placement

6) Logical Cell Placement Blockage

1) Define the Width and Height of core and die:

Depending upon the size of logical cells to be placed on the chip and their size, the width and height of Core must be decided. It defines the area required by the chip for its design. To find the ratio of area occupied utilization ration is calculated. It defines the portion of the area being occupied.

$$\text{Utilization Factor} = \frac{Area\ occupied\ by\ the\ netlist}{Total\ Area\ of\ the\ core} \qquad (2.1)$$

Ideally Utilization ration should be 1.

Higher the value of UF ,better is the core utilization.The design is always made such that utilization factor is near to 1 ,but due to several isssues the core has to be left untilised. Fig 2.5 (a-c) shows the floor planning part of design.

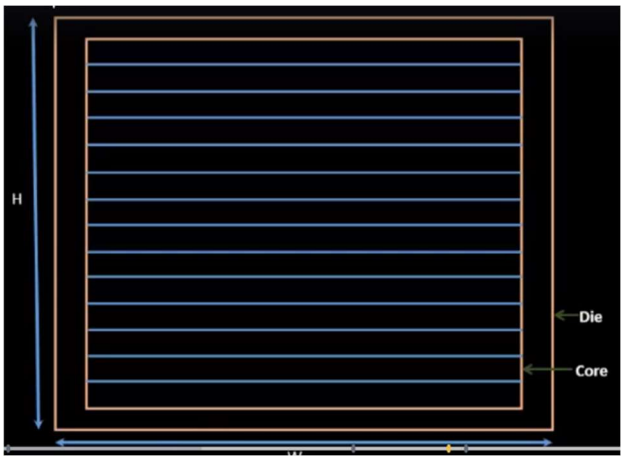


Fig2.5(a)Defining the width and height

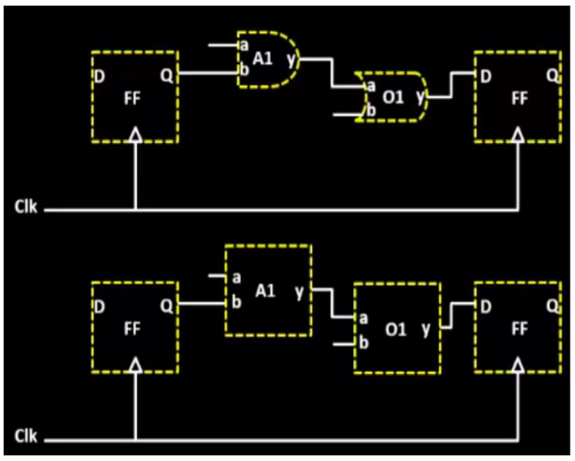


Fig 2.5(b) Binding the netlist to calculate the effective area

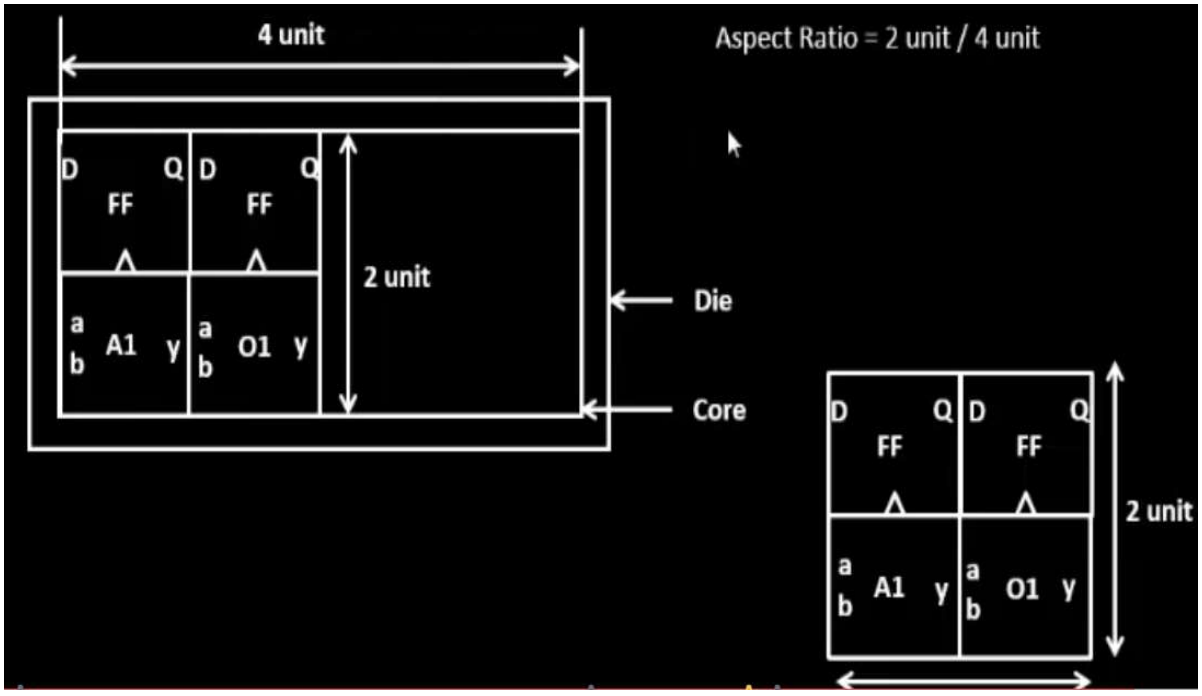


Fig 2.5(c) Finding the Utilization factor of Chip.

2) Defining the location of preplaced cells:

Preplaced cells are the cells that are already available to be placed and only their connections need to be taken care of. Already available cells are also called IP's or Macros, their functionality isfixed and cannot be altered.

The IP includes standard blocks like memory block or other stanard IP bein imported by other vendor or desing. These IP's can be reused again where ever required. Fig 2.6 (a-c) depicts the location of Macros on IC. The arrangement of these IP's on block is referred to as Floorplan.

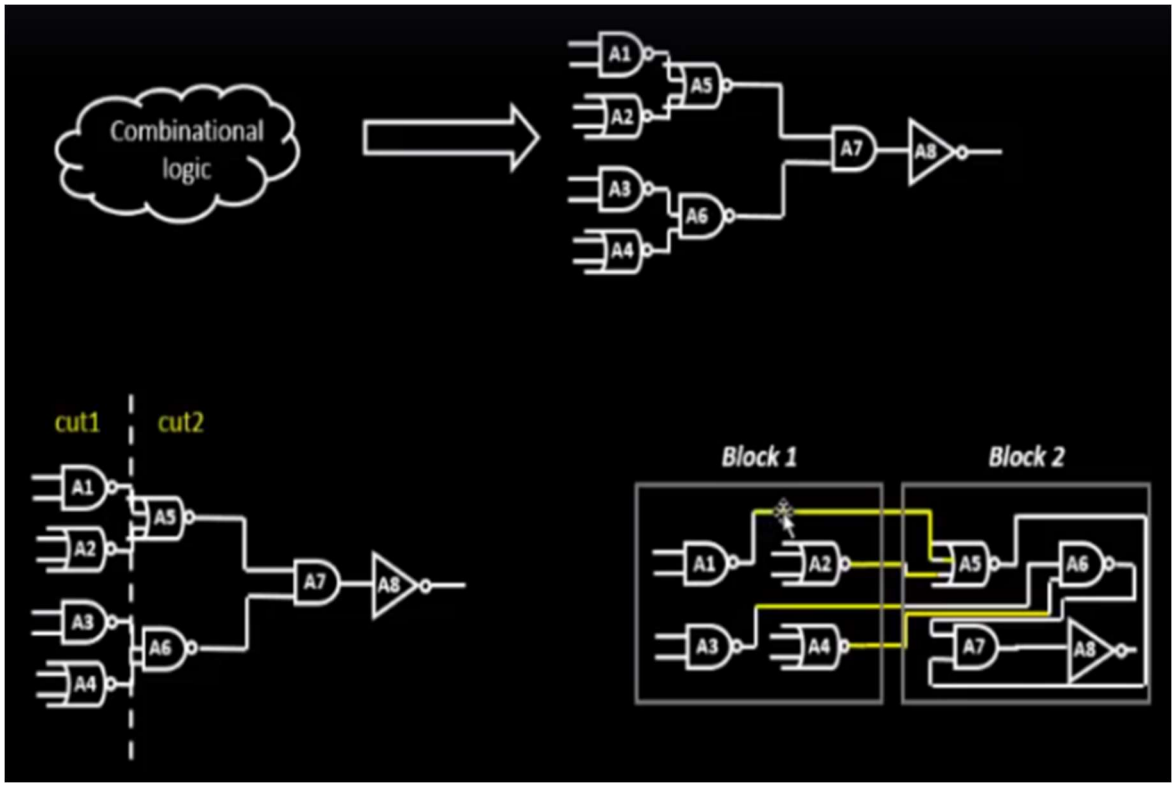


Fig 2.6(a) Making Interconnections between IP's

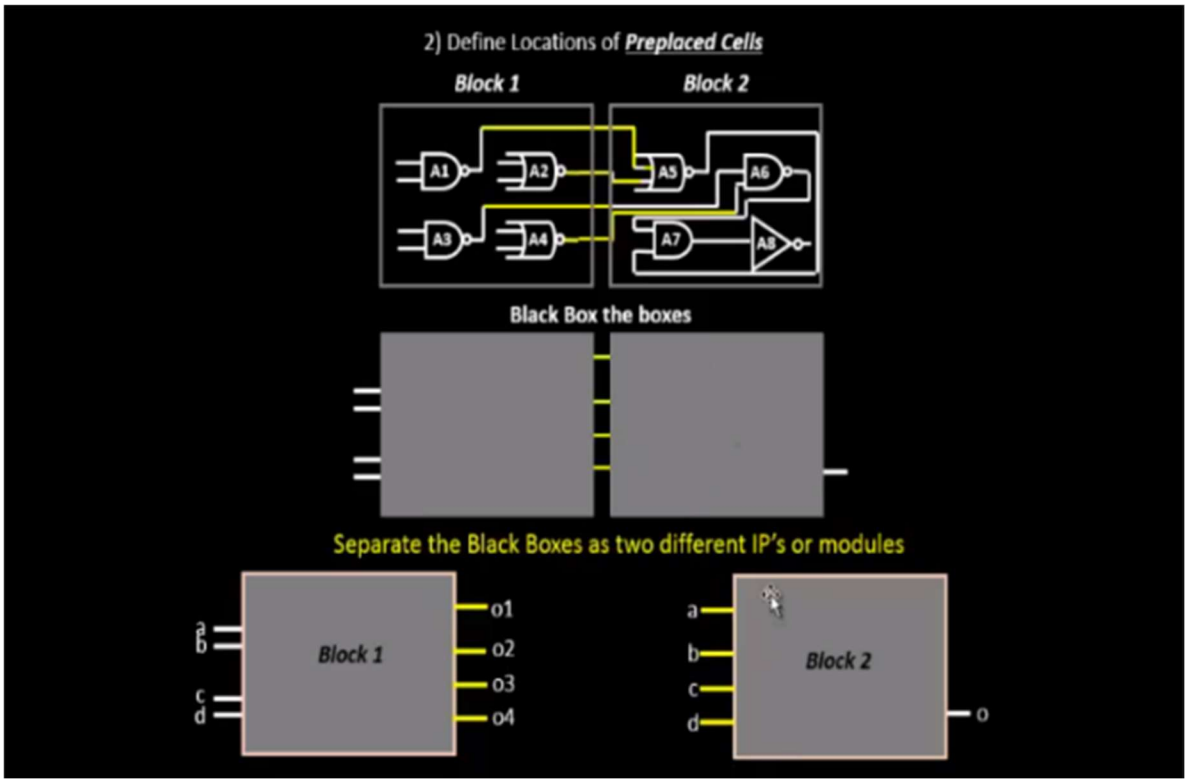


Fig. 2.6(b) Asuming Macros as box

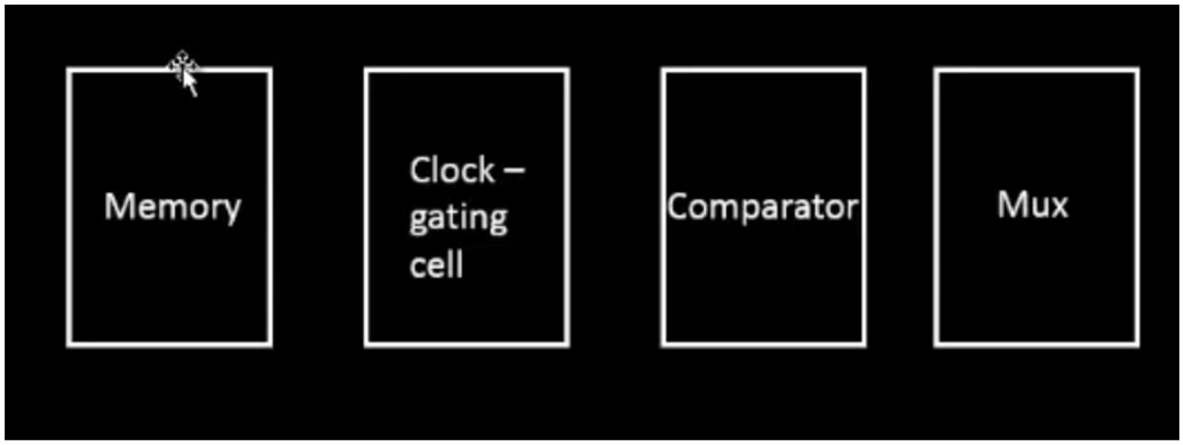


Fig2.6 (c) Commonly available Macros

3) Surround the preplaced cells with decoupling Capacitor:

In Fig 2.7 (a-c)

(i) Consider the capacitance to be zero Rdd,Idd,Iss are well defined values

(ii) During the switching operation circuit requires switching current I peak

(iii) Now due to presence of Rdd, Ldd there will be a voltage drop across the inductor and resistor

Hence. the voltage available to the blocks will not be Vdd rather a voltage less than Vdd. To provide the cells with full voltage Vdd, a capacitor is placed in parallel to the blocks, which charges to the value Vdd and in absence of supply voltage or any decrease in voltage drop this capacitor Cd discharges and provide full voltage for operations of block.

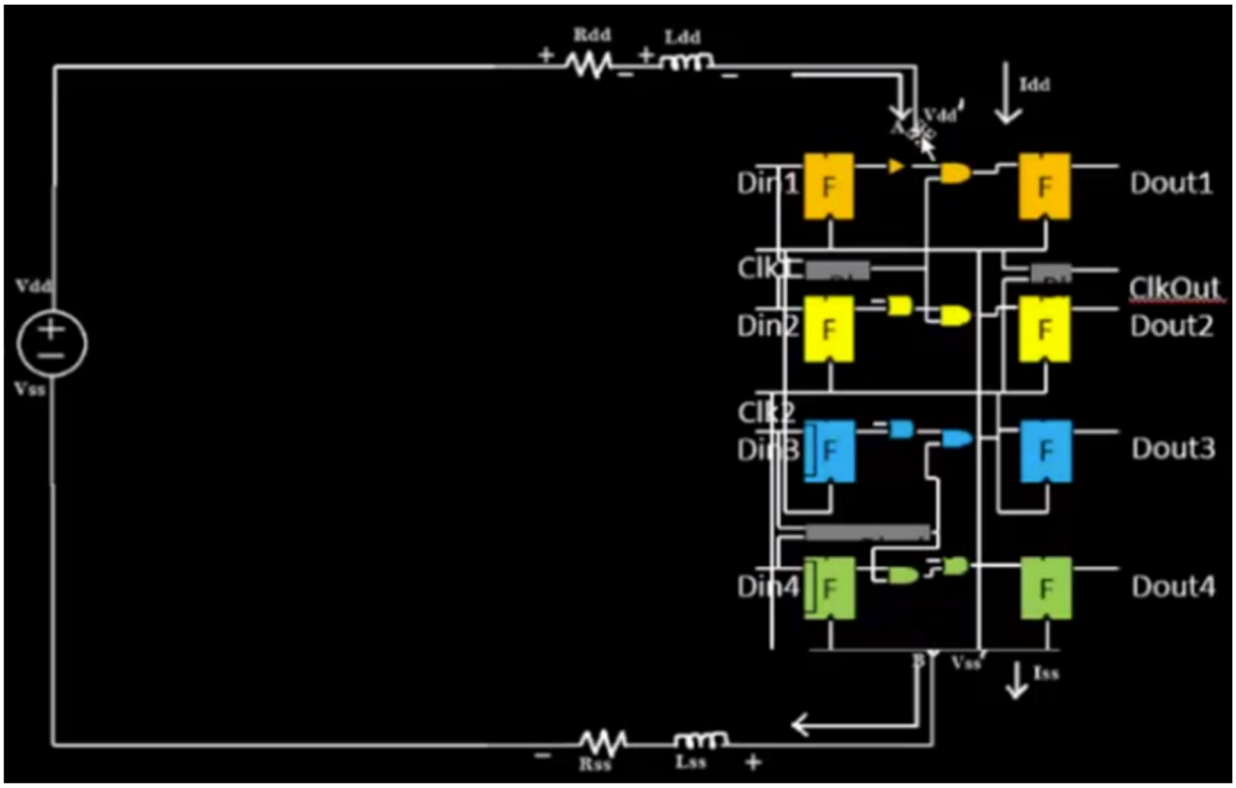


Fig 2.7(a) Cells without Cd

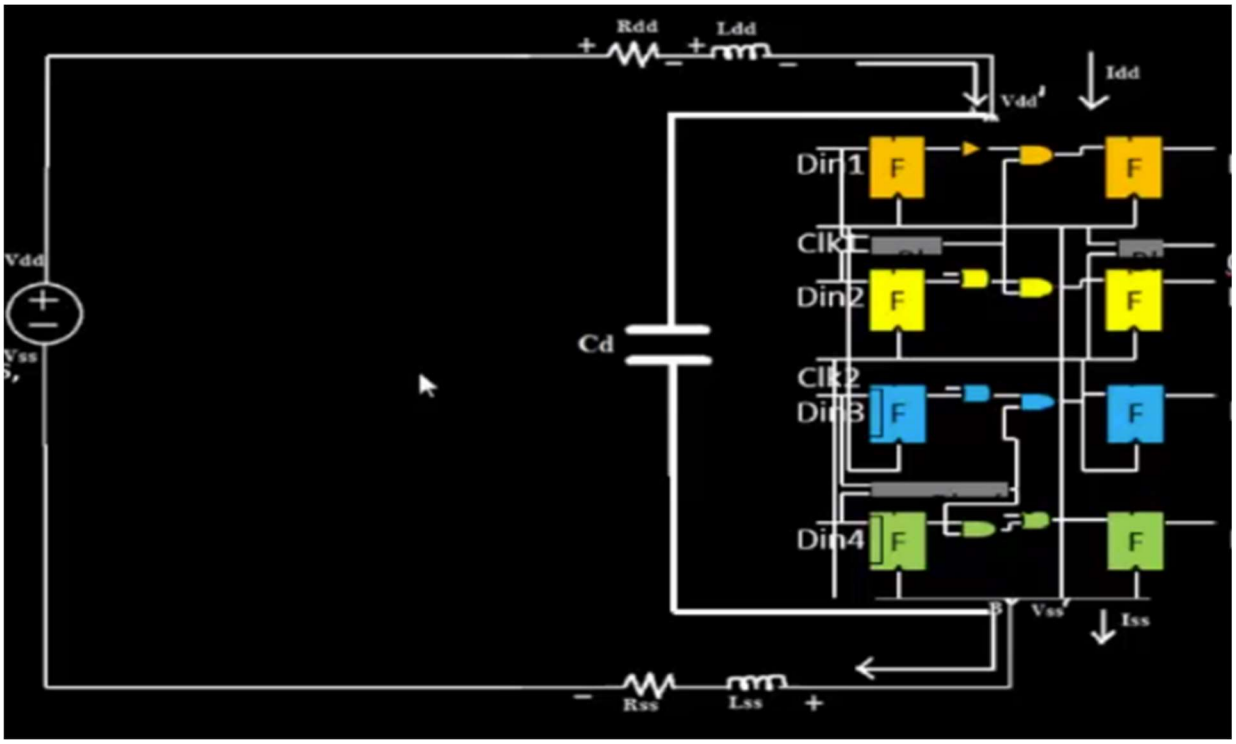


Fig. 2.7 (b) Cells with Cd

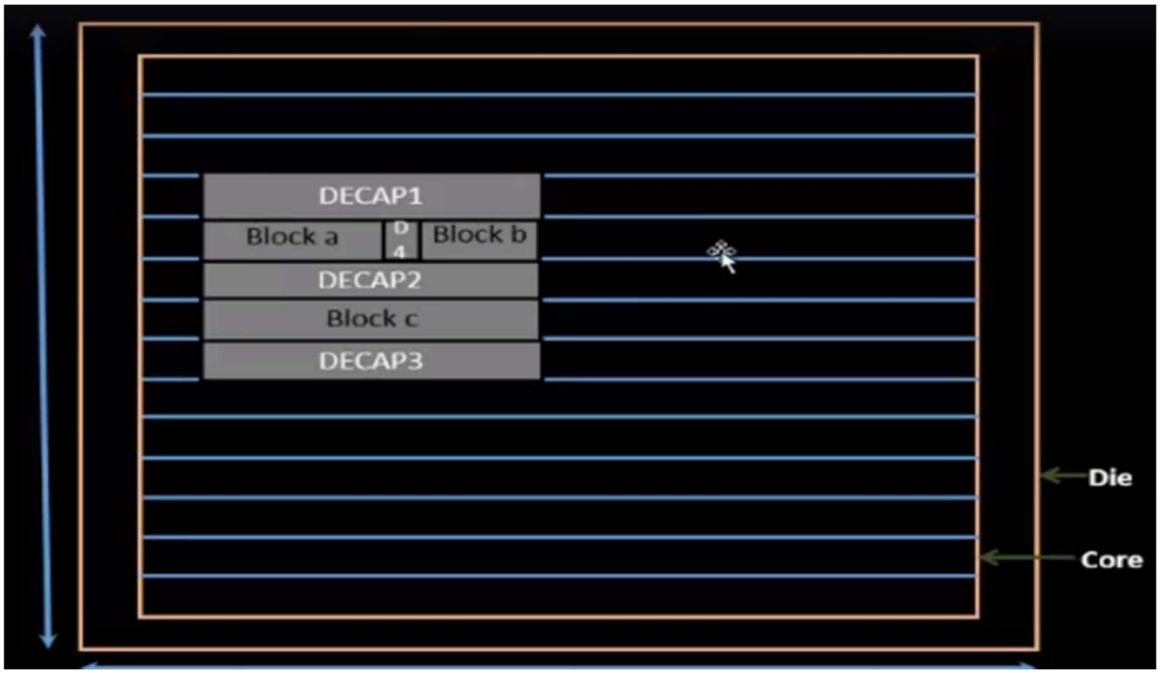


Fig 2.7(c) Placement of decoupling capacitors on IC

Power Planning: The problem associated with central power supply is that it does not reaches as exact Vdd at the inputs of pins due to parasitic capacitance formed and it leads to the problem of Voltage Droop and Ground Bounce.

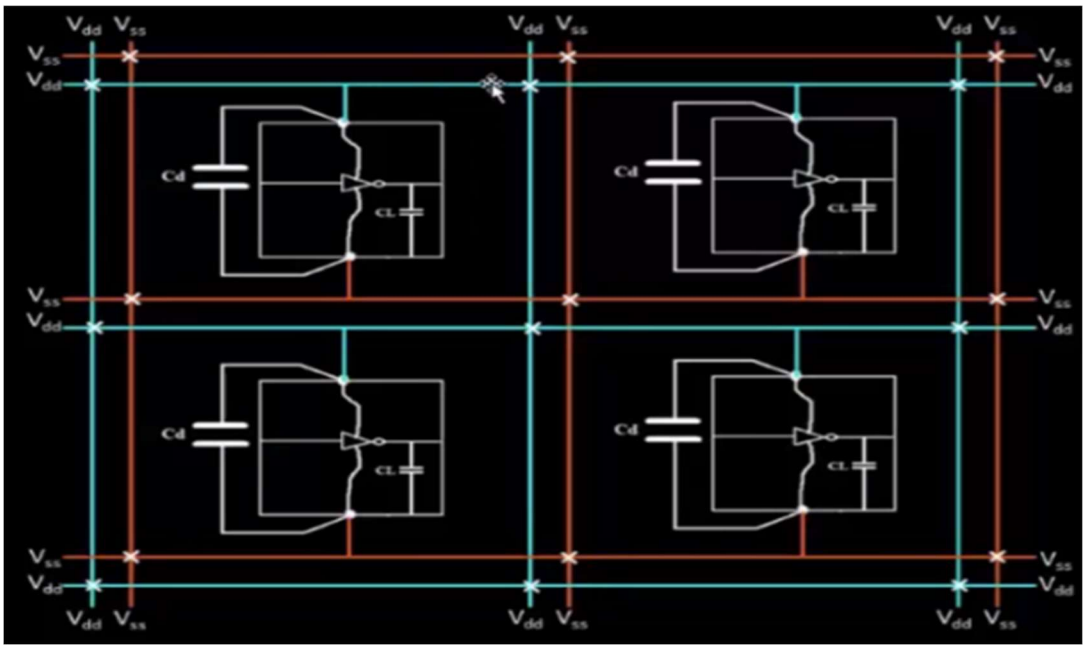


Fig 2.8(a) Power Supply being Created

To avoid such problem, instead of having a single supply voltage a mesh of supply voltage is created on chip and power is tapped from nearest Vdd point so that the voltage does not drops. Fig 2.8(a) and (b) shows the arrangement of power planning on the Chip.

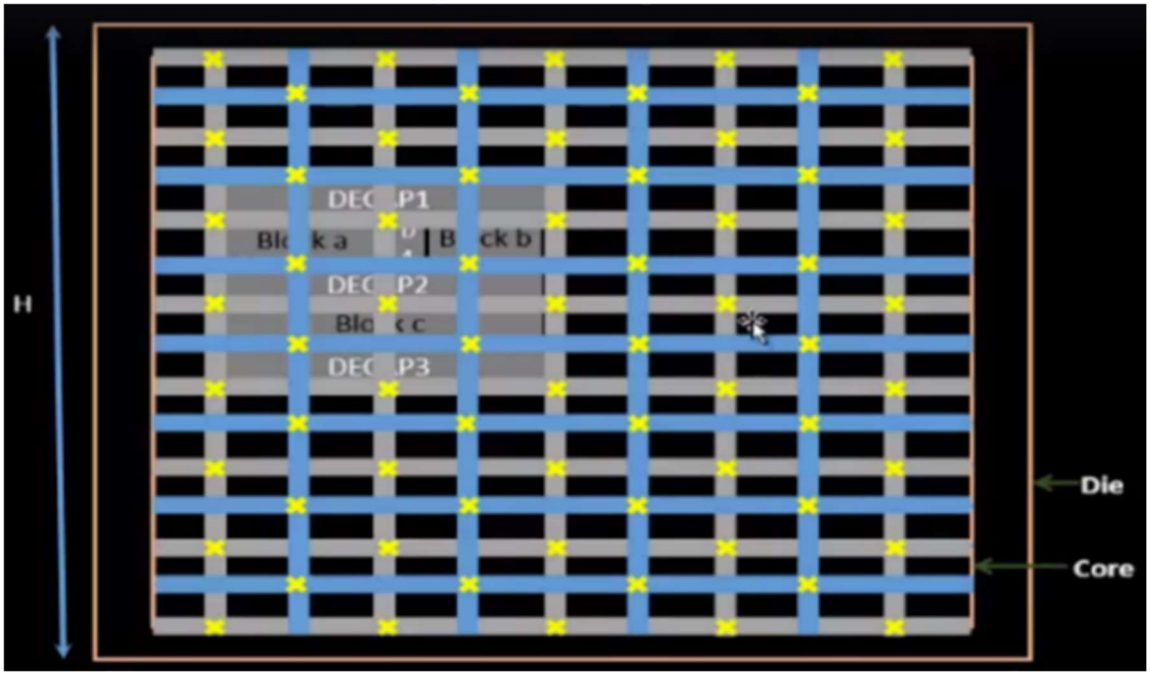


Fig 2.8(b) Mesh arrangement of power Supply

4) Pin placement and logical cell placement blockage:

After power planning, the pin placement is done.Consider the circuit diagram in fig 2.9(a)-(c) that is supposed to be placed on the chip. The input and output pins are placed on the chip such that all input pins are on left side and all output pins on right side. Pin placement is followed by Logical cell blockage,where logical cells placement are blocked and nothing can be placed in the place.

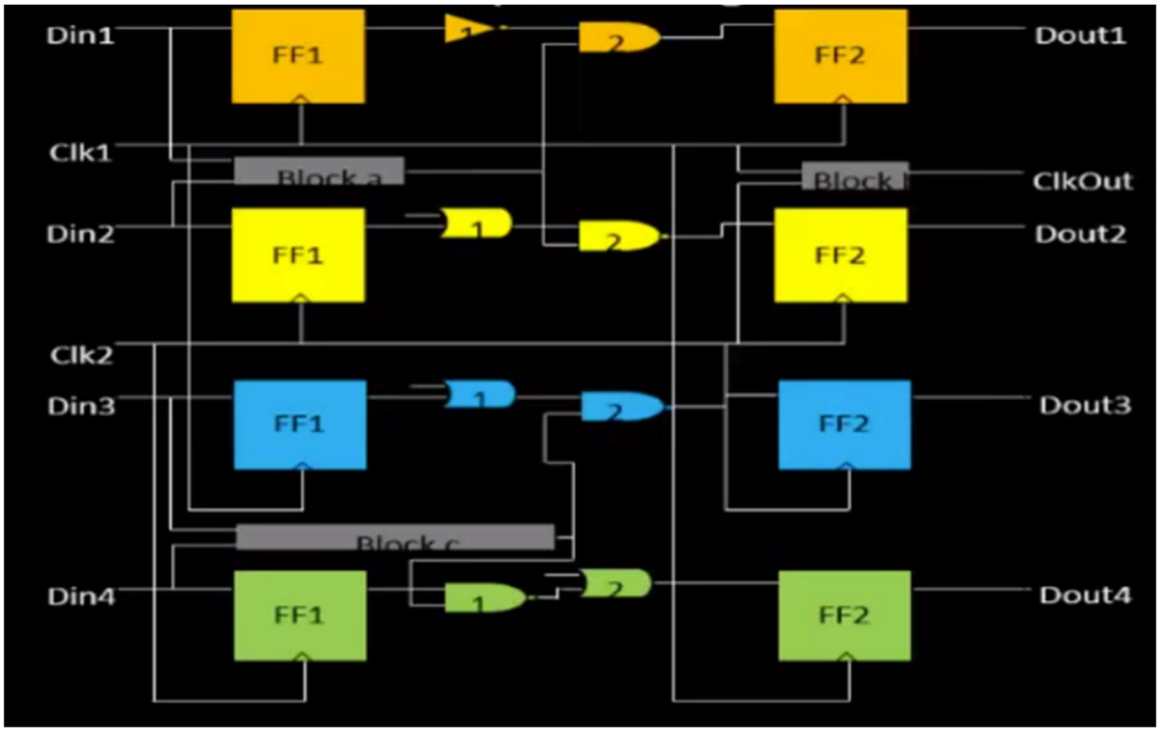


Fig 2.9 (a) Circuit whose pin placement has be done.

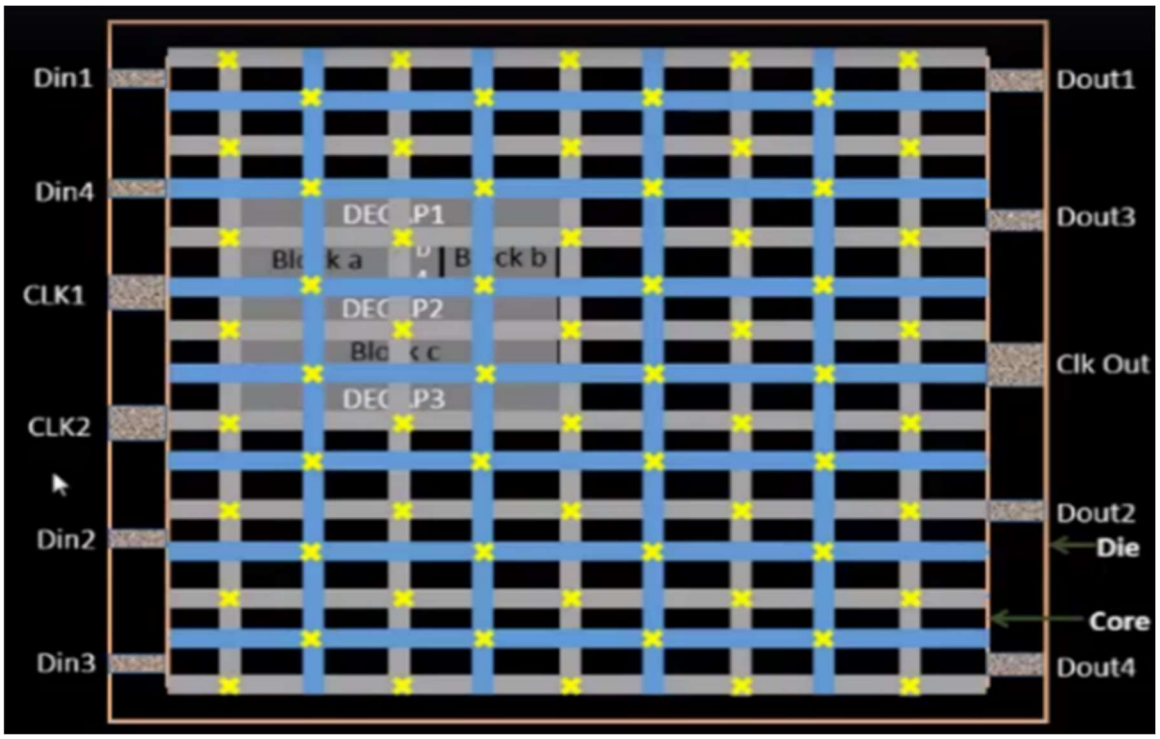


Fig 2.9(b)Pin placement of the circuit

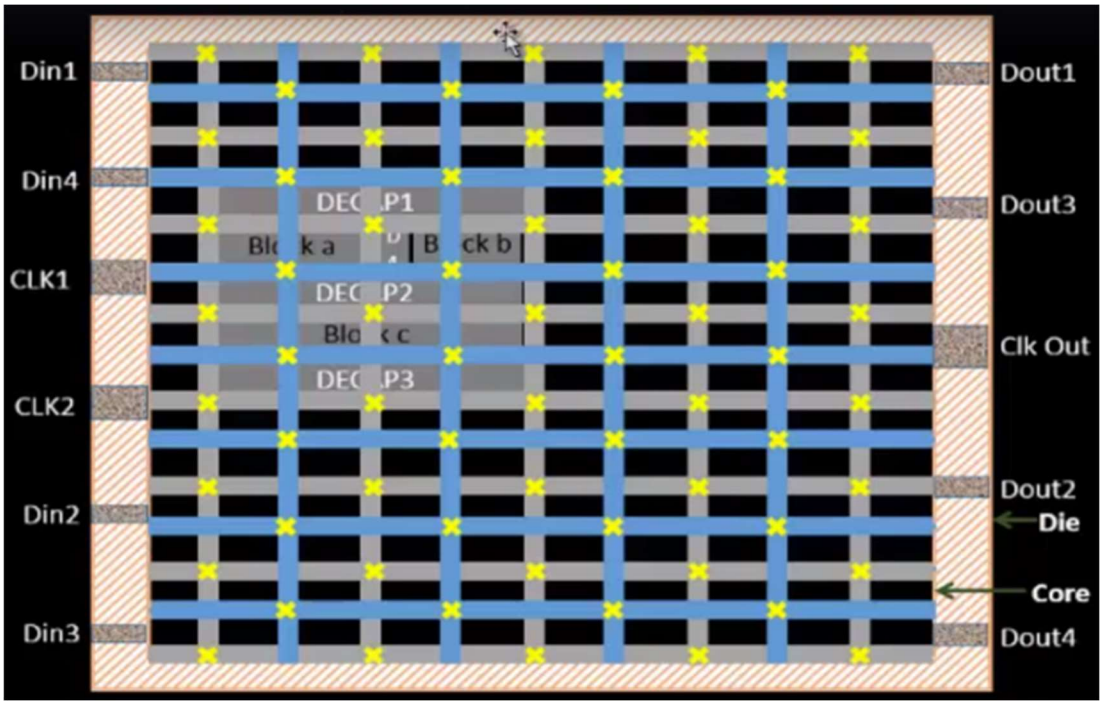


Fig 2.9 (c) Logical Cell Blockage

The paper [1] ,discusses about the HS algorithm ,which is inspired by music phonmonenon of musucians to achieve the placement area in floorplanning design.The algorithm discusses in [2] details about area reduction methods and advantages over other algorithms.The code is witten in MATLAB and results are simulated and validtated through MCNC benchmark circuits. In paper [3], a newly proposed algorithim B*tree crossover is proposed,which works on the principle of simulated annealing algorithm (BCSA),which works to minimize the dead space for optimizing the area on the chip.The algorithm shows significant results in area optimization

2.3 Placement and Routing

Floorplanning decides where the logical blocks, IP's and any other element has to be placed. After the placement of these bocks interconnections between the blocks has to be done. The way interconnections has to be done is decided by

placement and Routing tool as shown in fig 2.10(a)-(c). Follwing steps are performed in Placement and Routing

1) Bind the netlist with Physical Cells.

2) Placement

3) Optimize Placement

1) Bind The Netlist with Physical Cells:The netlist consists of the gates and other logical elements that are there in the circuit.These cells has to be attached with a library,which has all these cells, Library also contains the timing details of these cells. Beside this it also contain infromation of required condition on which the cells works. The library contains different size of the cells to be placed,which are used according to speed and power requirements.Based on the timing requirements and design need ,these cells are chosen from the library.

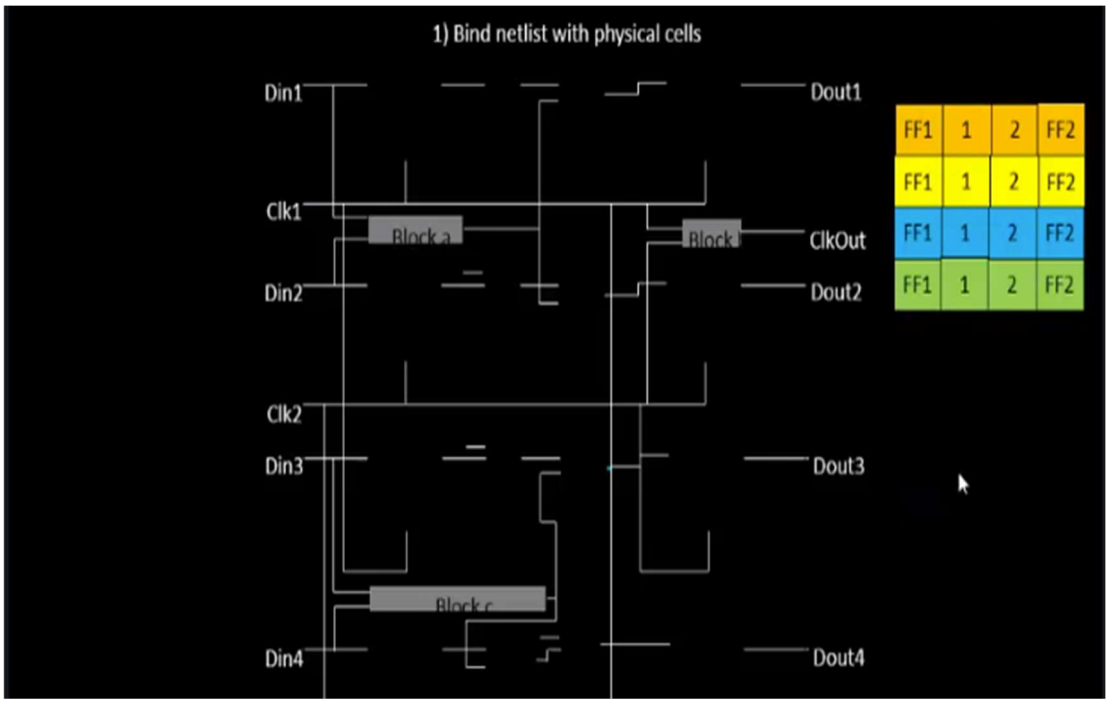


Fig 2.10 (a) Mapping the Cells into Library

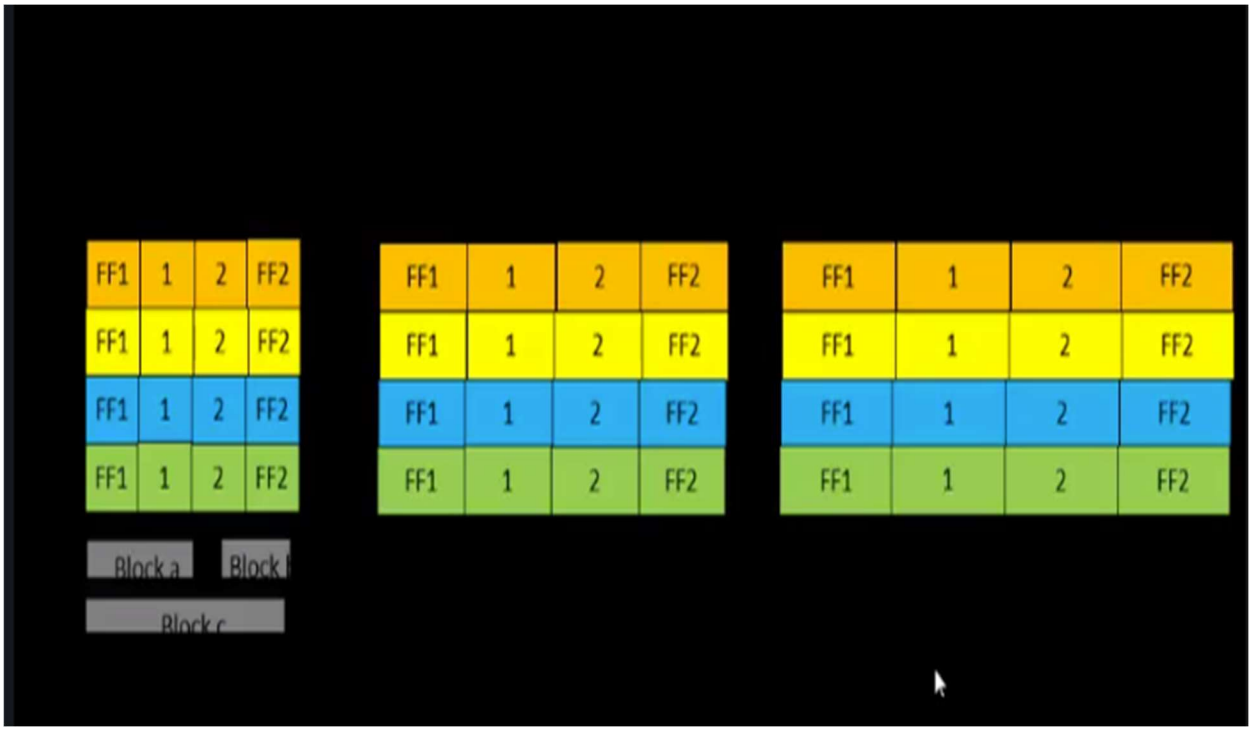


Fig 2.10(b) Cells of different Sizes into Library

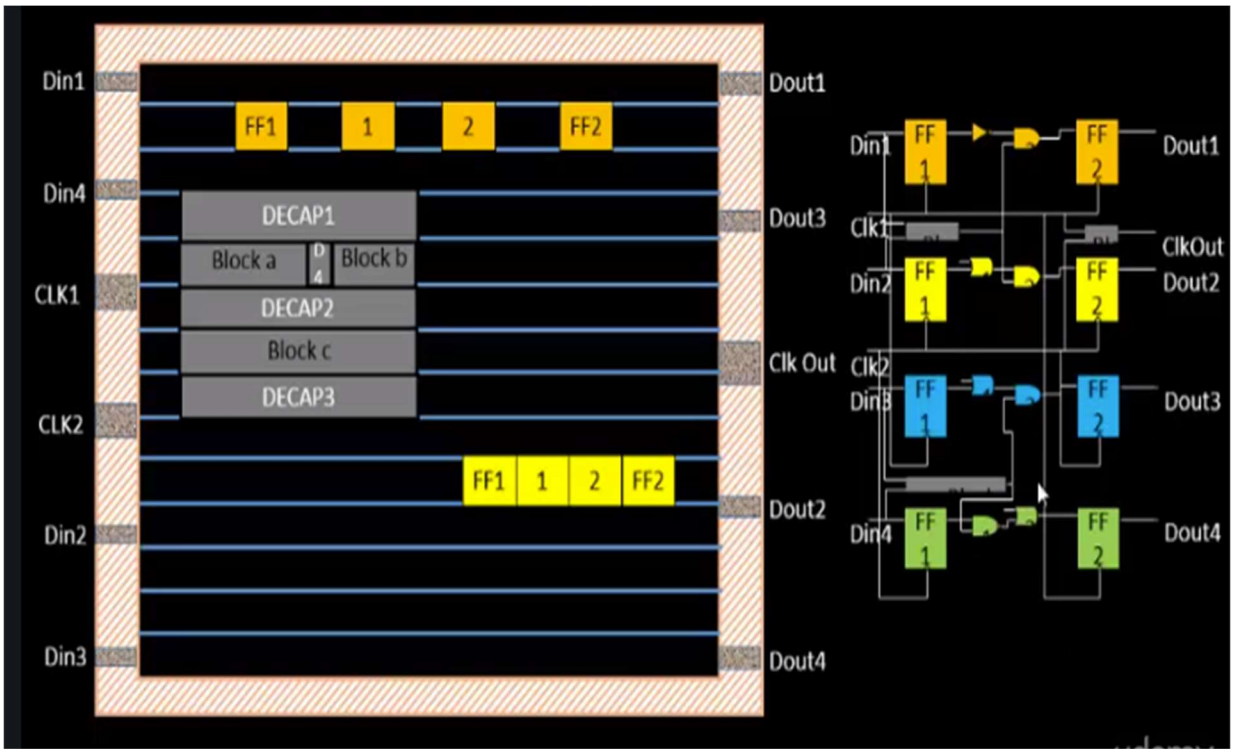


Fig 2.10(c) Placement of Cells on the Chip

2) Placement and Optimize Placement: In optimize placement stage, we estimate the wire length and capacitance and based on that repeaters are inserted.Repeaters are inserted to maitain the same signal strength from IC input to that of Cell input.Fig 2.11(a-c) shows the optimze placement of logic blocks with repeaters inserted in between them.

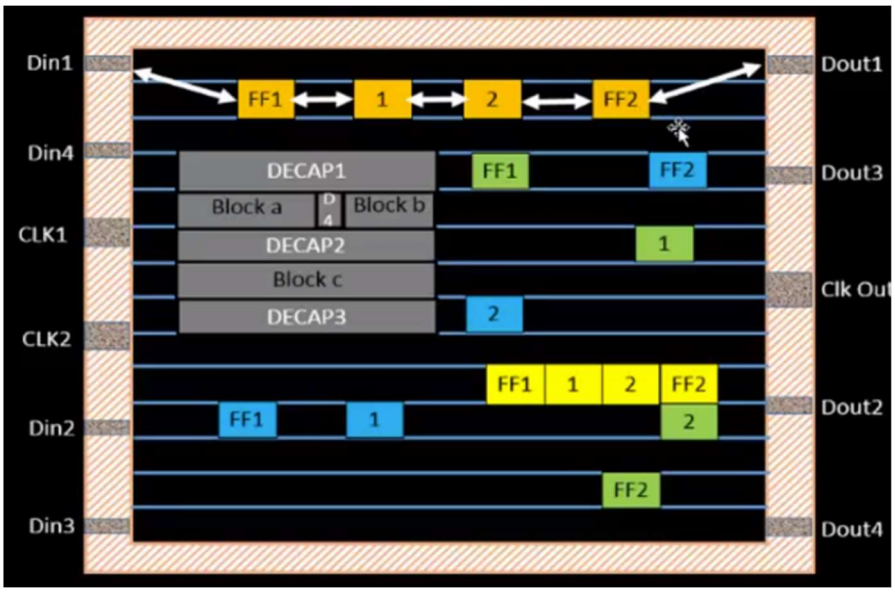
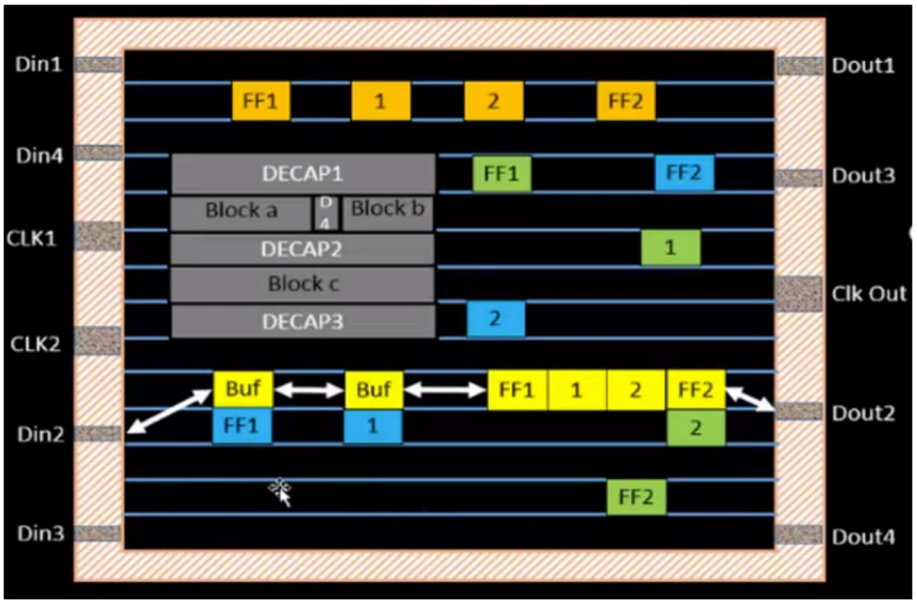


Fig 2.11(a)



2.11(b)

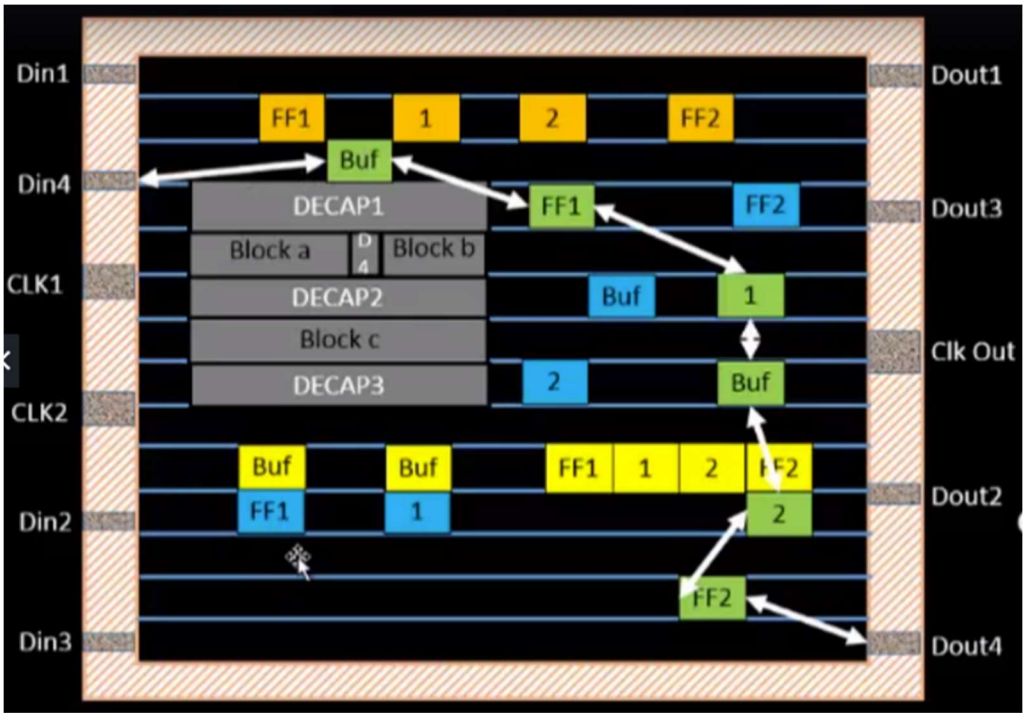


Fig 2.11(c)

Fig 2.11 Optimize placement of blocks with repeaters inserted.

The paper [3] discusses about the conventional VLSI physical design techiniques and the overall methodology adapted for various cell placement,routing techniques and physical synthesis of curcuits. Paper [4] discusses about thr macro cell placement algorithms and the optimizations achieved through these algorithms.The paper also discuss how the alogithmachives the area optimization also.The algorithm described in [6] is iterative and incremental ,which allows flexibility in design flow.The ALGORITHM also gives better results in terms of delay in path.

2.4 Clock Tree Synthesis

In physical design flow after routing and placement process, timing analysis using ideal clocks is done to check whether the circuit is meeting the timing requirements or not.Timing anslysis done is static in nature since its independent

of input parameters.Timing analysis helps to identify the frequency of the clock at which circuit will run.Obtaining the ideal clock for circuit is a diffiicult task and engineering is done to make the clock to near ideal. Clock is generated with the help of crystal oscillator, availabe at the CLK pin of the chip. The clock is now supposed to be made available at the cells input ,which are placed inside the chip, keeping clock paramters in a stable and according to constraints in an important task.Routing of clock to cells,with proper timing parameters, inside the chip is called Clock Tree Synthesis.

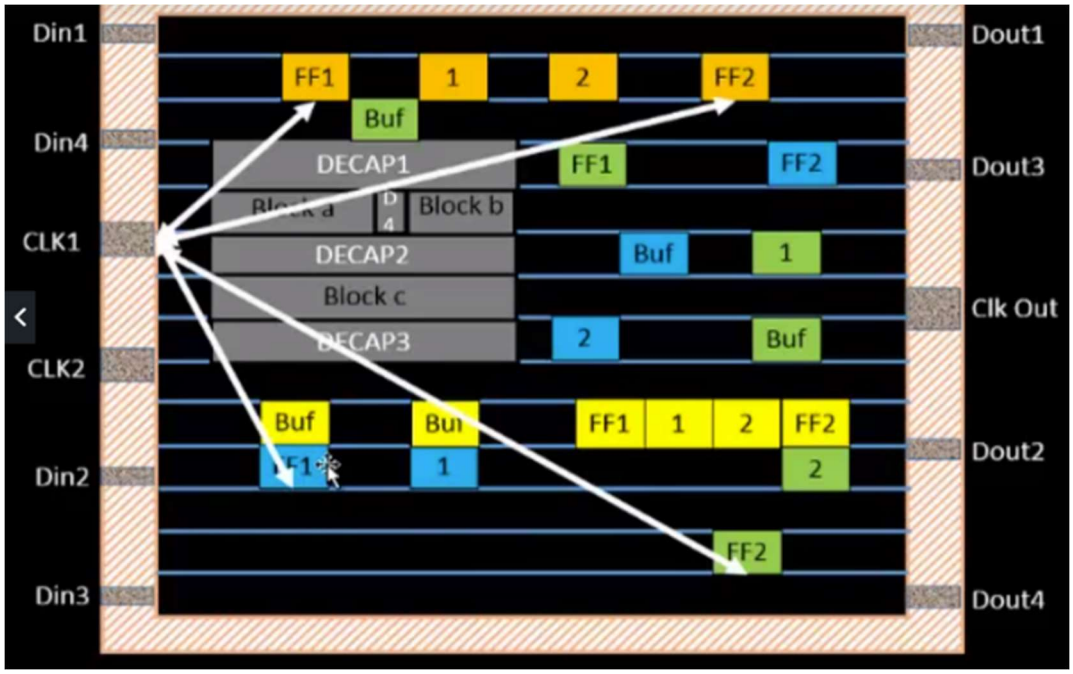


Fig 2.12(a) Clock availabe at CLK1

For CLOCK there are six quality parameters check that are supposed to be performed.The six quality parametrs checks are :

1) SKEW
2) PULSE WIDTH
3) DUTY CYCLE
4) LATENCY

5) CLOCK TREE POWER

6) SIGNAL INTEGRITY AND CROSS TALK

Above quality paramters are to be fulfilled while routing the clock from CLK pin to Cell input. Failing to obtain the above parameters would lead to malfunctioning of the chip,which is not desirable.

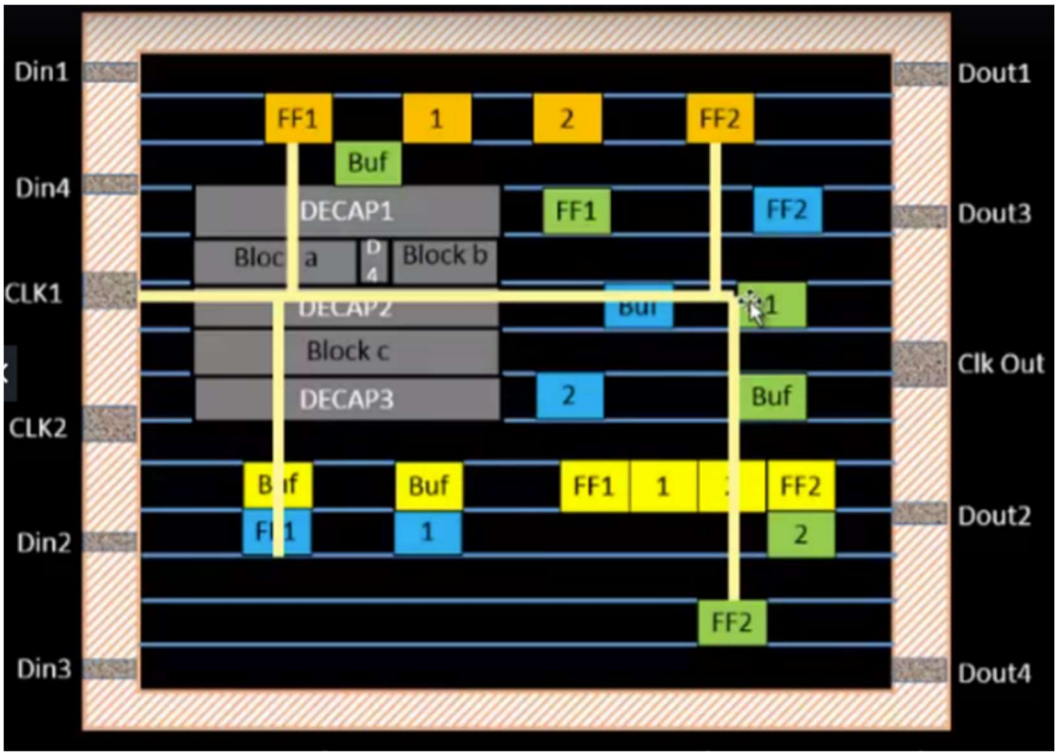


Fig 2.12(b) CLK being routed to cells input

There are well defined algoritms built and EDA tools uses these algorithm to Route the clock to the cell's input. H-Tree algorithm is one of the famous algorithm used for routing the clock.In H tree algorithm,the clock is routed in H fashion ,till the input of the cell is reached.H-Tree algorithm Minimizes the Parasatic capacitance and ressintance between the pin and input terminal so that all the 6 quality parameters are met.Fig 2.13(a)-(c) shows the clock being routed with the help of H-Tree algorithm.

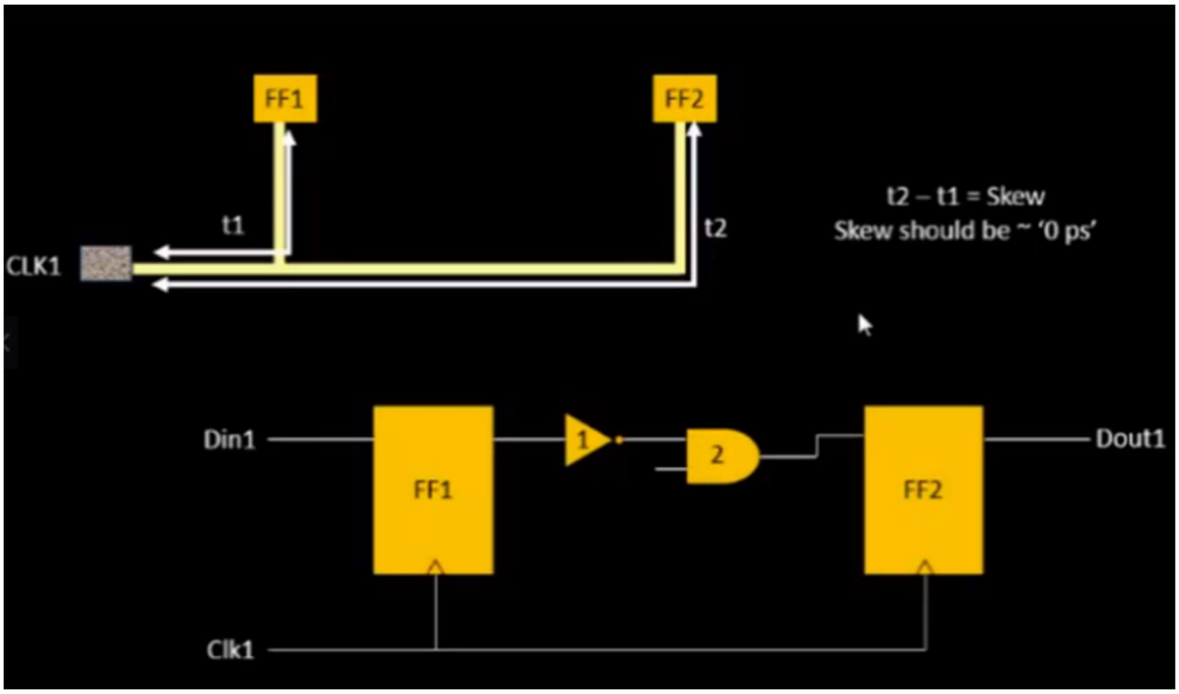


Fig 2.13(a) Routing of CLK to cell input.

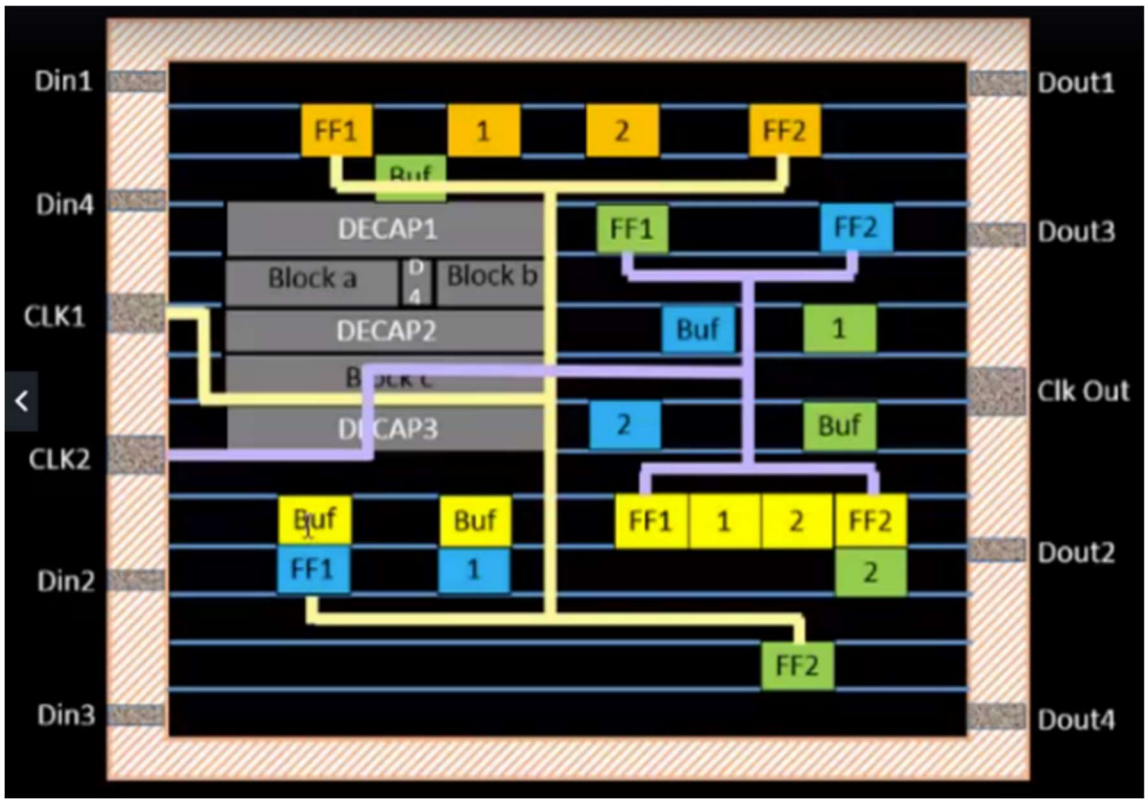


Fig 2.13(b) CLOCK routed by CTS algorithm

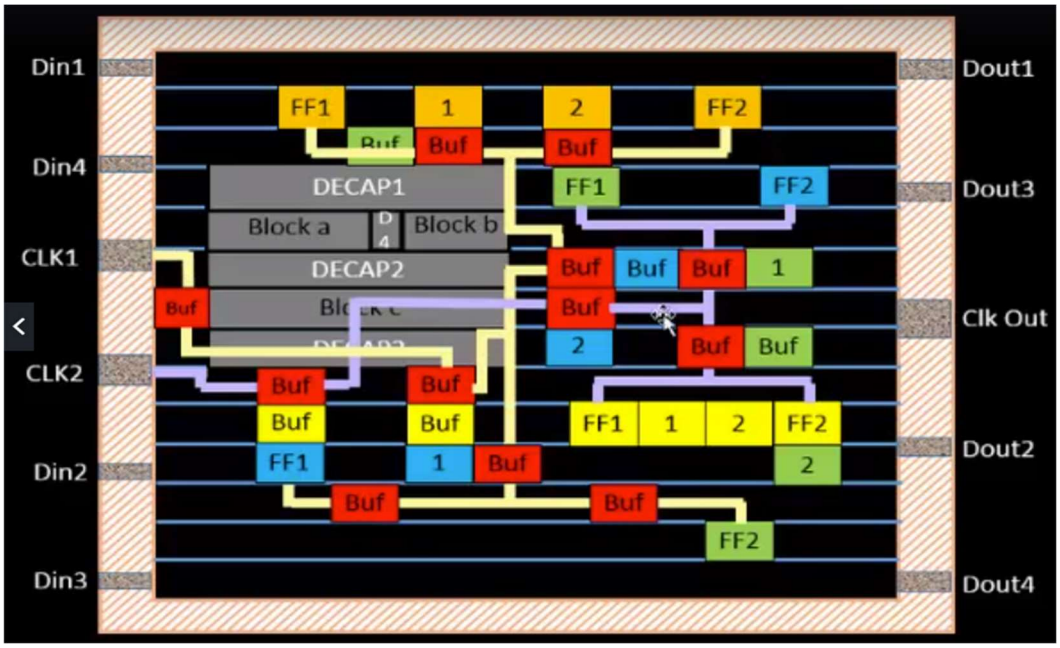


Fig 2.13(c) CLOCK routed with buffers to maintain the strength

Due to parasitic Ressistance and Capacitance formed due to wiring between different interconnects, the srength of CLK may degrade,which is not which is not desirsble.Buffers are used in CLOCK path,which increases the strength of the CLK.Fig 2.13(c) shows Routing of CLOCK after inseritng buffers in the path.

The paper [7], gives details about amethod which aids in low clock skew applicable to the industry standard CTS design flow.The alogirthm details about how clock root is partitioned into several pseudo clock soures a gate level [8].The work in paper [9],[10],[11] investiges about the different methods for minimizing the impact of on chip variations in clock by the use of clock biffers and wire sizing.The OCV varaitons has been disraged in clock path ,these paper discussed in details about the clock and data path variations and provides three different algorithm that allows clock skew to be corelated with path sensitivities to timing violations.

2.5 Routing

Till now in Physical Design cycle All the IP's and logical cells have been placed on the Chip and clock has been Routed to all the cell inputs ,which requires CLK as one of its inputs.Connection between the logical cells has to also done,this process of connection is called Routing.There are several algorithms that are used to do Routing between th cells.The Routing has to be done in such a way that it choses the shortest path between the two cells,such thef formation in parasitic ressistance and capacitance is minimum.There are several algorithm for routing.One such algorithm,Lee's algorithm is described.

Lee's Routing Algorithm

The steps followed in Lee's Algorithm are :

1) Routing grid is developed in the Backend of floorplan.
2) Source and Terminal are decided.
3) All grids around source are level 1,all grids around level 1 are level 2 and so on
4) Levelling is increased from 1→2→3→4→5…,until we reach the destination.
5) L shape routing is preferred over zig-zag.
6) This process is repeated for all the cells.

Fig 2.14(a-e) shows the various steps of Lee's algorithm beingfollowed.

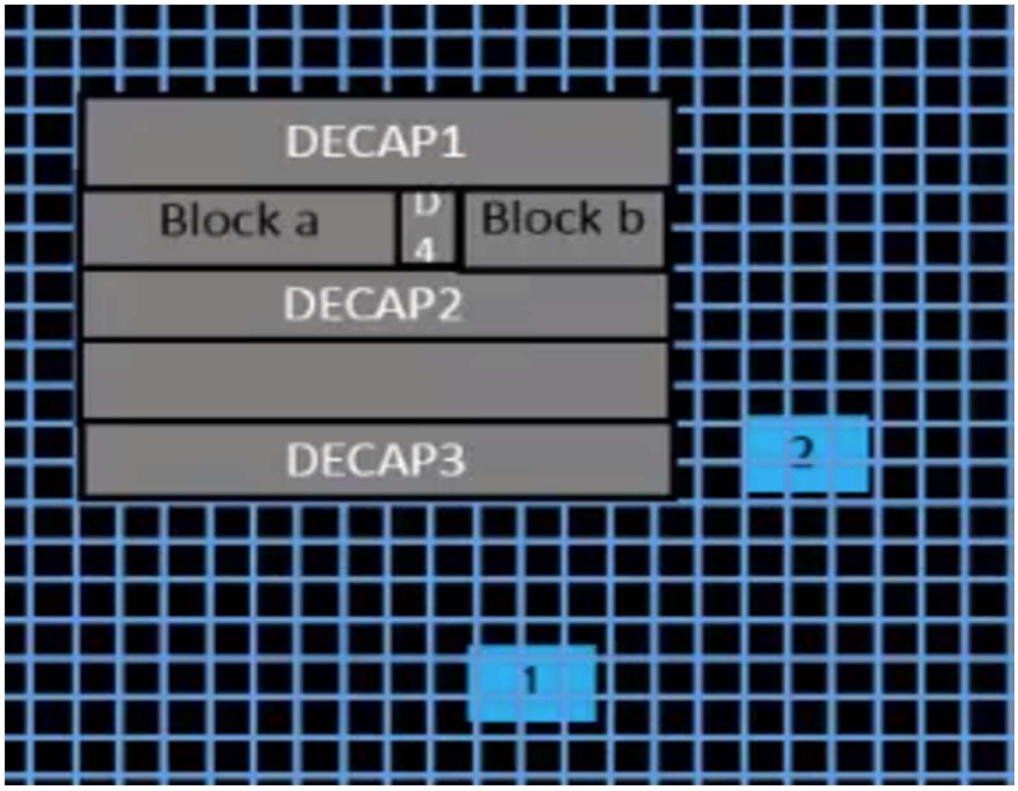


Fig 2.14(a)Mesh Grid on

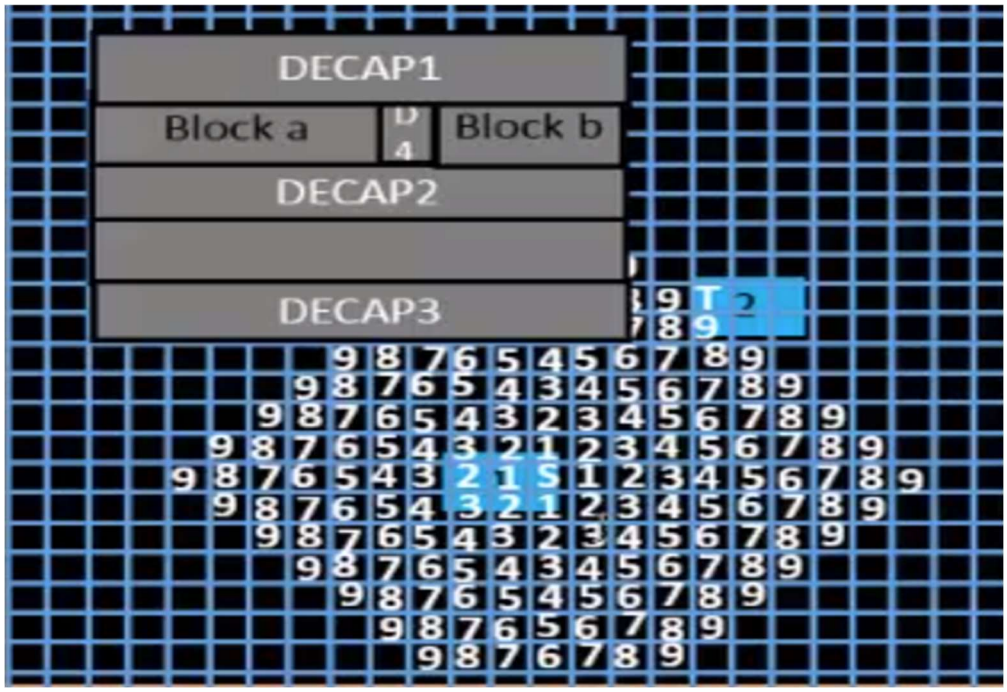


Fig 2.14(b) Numbering the girds around the Numbeirng

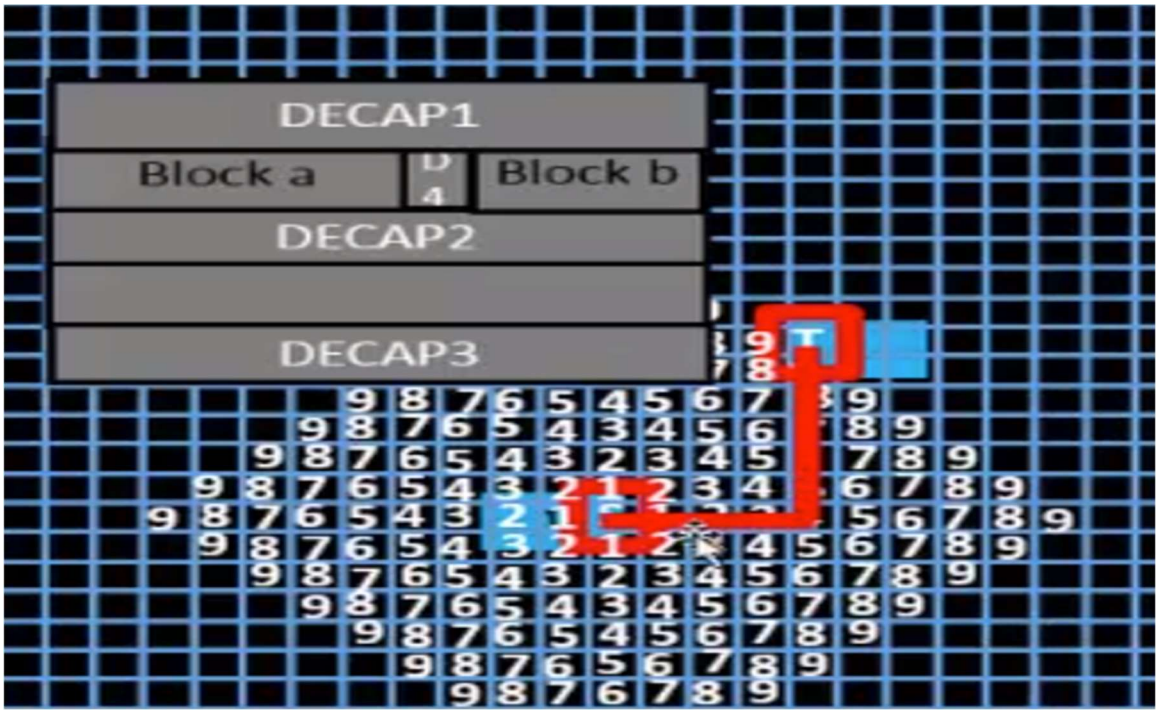


Fig 2.14(c)

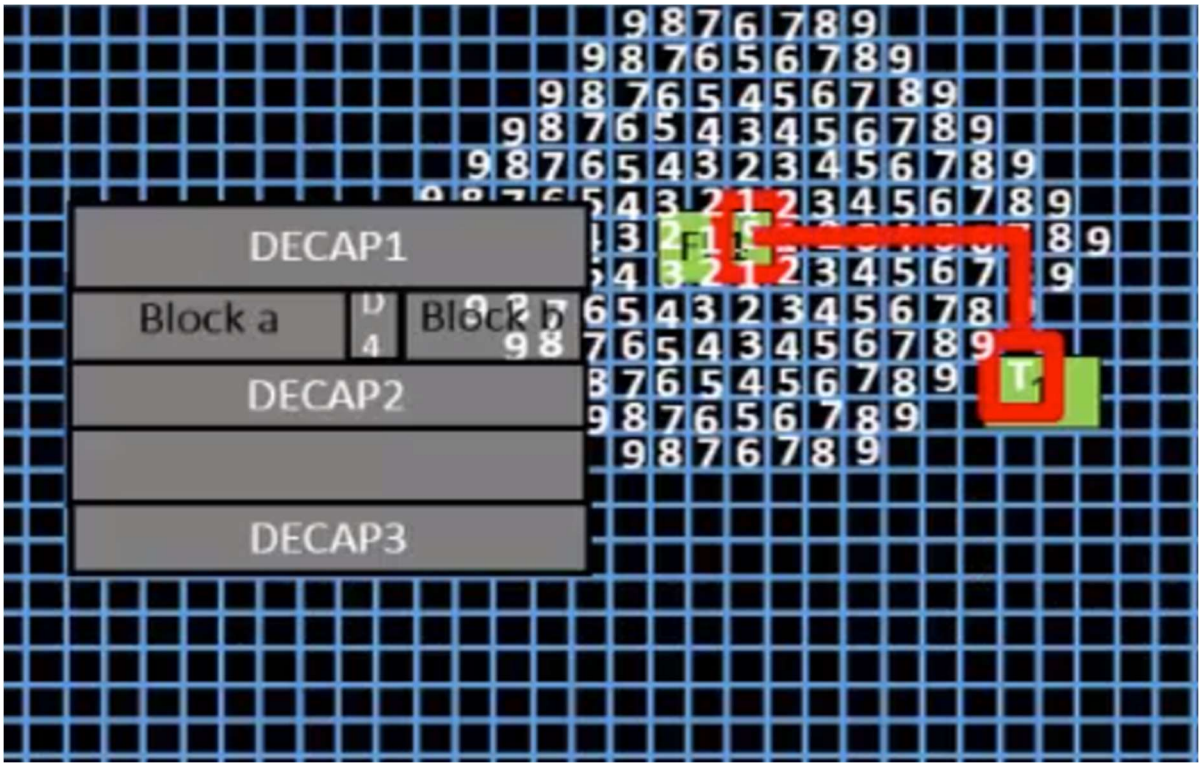


Fig 2.14(d) Routing wire

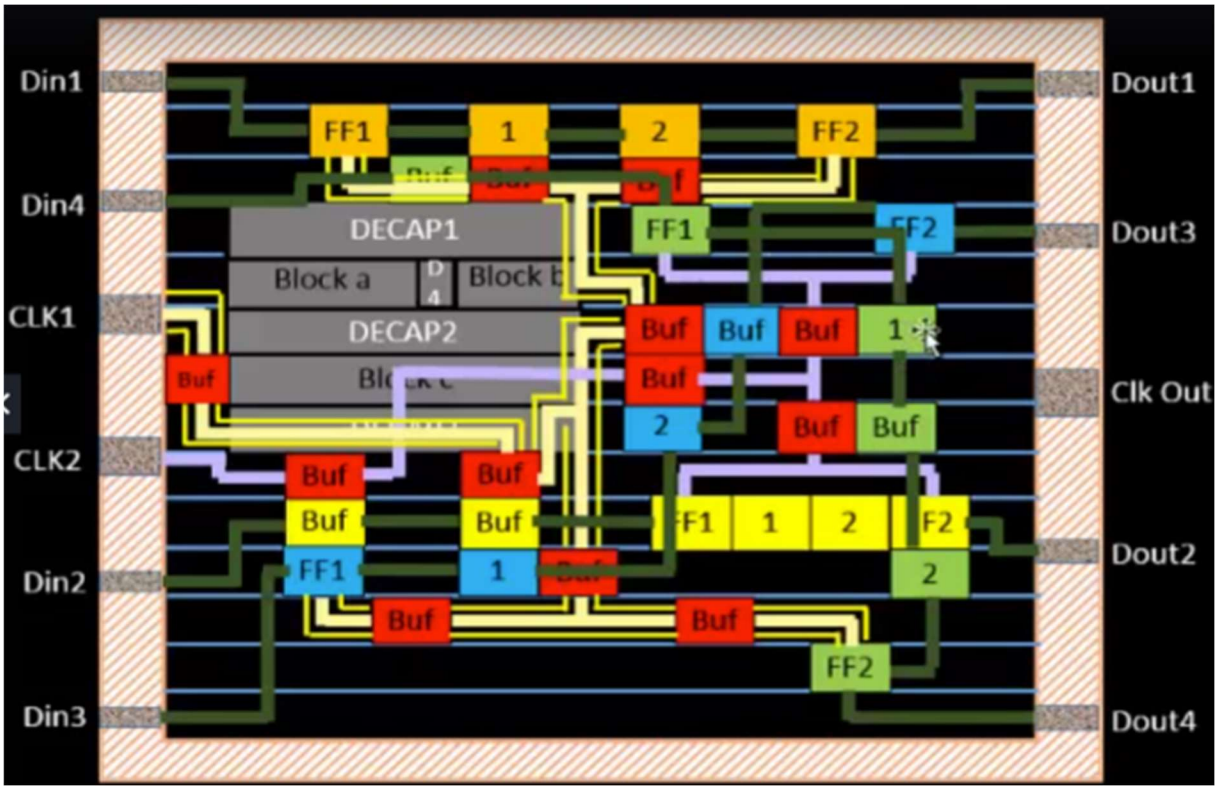


Fig 2.14(e) Circuit with complete Routing

2.6 DRC check and Parasatic Extraction:

DRC check and parasitic extraction are the last steps of Physical Design Flow,the chip is ready and all the Design Rules checks are supposed to be performed The checks are performed to ensure that the chip after fabrication does not produce erronous results due to the connection between the components.There are several rules laid out in EDA tools,which automatically checks the design.There are more than 10,000 DRC rules that are checked by EDA tools.

After the DRC rule check the chip is ready and send to FAB for its manufacturing.Before sending to FAB, parasitic extraction is done. In Routing design process ,there were physical wires laid on the chip,so parasitic extraction

says that each wire has got finite resistance and capacitance and we have to extract the capacitance and provide it to analysis tool. The extracted Capacitance and Ressistance need to be represented in file format.There are many standard way to represent Resistance and Capacitance, one way is SPEF format. SPEF stands for Standard Parasitic Exchange Format.This format is laid by IEEE.This format gives RC value of the wire.

This chapter gave an overall view how design cycle works,the details of design cycle and various steps assciated in a physical design.The next two chapters provides the details of automation of designcycle, challenges that were encountered and possible solution to the problems.

CHAPTER 3

RTG Automation

3.1 Introduction

As technology is advancing from submicron technologies to nano-meter technologies, design complexities and challenges also increased proportionally. As a result, engineers have lot of things to concentrate and the design needs a lot of attention so that it can be closed in the expected time.Also engineer needs to well take care of the things like firing the Runs, choosing correct inputs, creating command files, custom reporting and the other routine things. And, the project implementation team should ensure that the project specifications like corners, derates, HVT Cells, Metal Scheme etc., are well maintained in all blocks at all levels. All these things have given rise to the deployment of a robust automated system called RTG Flow.

RTG is acronym for RTL to GDS which has a very robust automation that helps in the smooth flow of implementation of Library qualification and Physical Design Flows i.e from the Synthesis to Tape-Out.Basically, RTG Flow involves running lot of scripts and wrappers to do a specific task like firing the runs, controlling them in both sequential and parallel manner, parsing the reports, comparing the runs, displaying them in web etc.In simple, it can do most of the tasks where the human intervention is not needed.

It has been developed with the scripting languages like Make, Shell, Perl, Tcl-TK, Python and Java for Web Integration.With every reduction of nanometer at technology front the complexity of design closure for implementation flows and methodology become more and more complex. On top of it time tomarket is

constant metric for success of any ASIC product keeping high pressure to meet Tape Out timelines. While implementation teams has to focus on understanding and fixing design and methodology issues due to heavy dependence on continuously evolving costly and resource intensive EDA tools their majority of time goes in understanding technology from foundry vendors, executing runs, waiting for results and analyzing quality metric from large log and report files to identify fixes and provide feedback while continuously fighting for machine and license resources. This loop is even more complex for technologies like 16FF or lower where most of signoff methodologies and flows are no longer standalone checks.The cycle execution flow is shown in fig 3.1

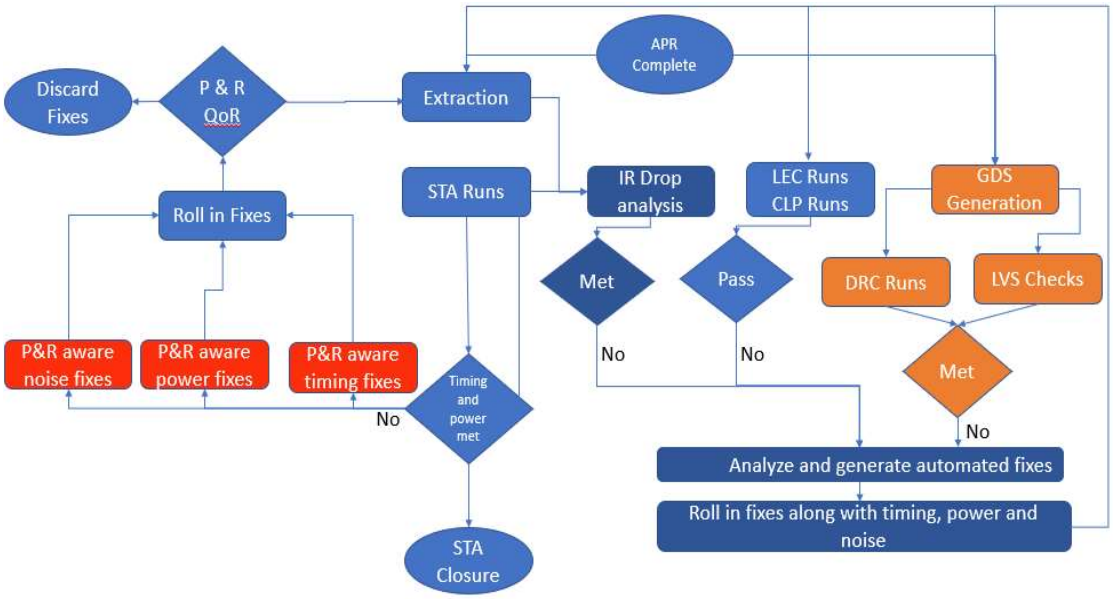


Fig 3.1 Implementation of Cycle execution flow

The RTG platform is an intelligent GUI based automation plaform named RTL2GDS(RTG), which not only manage technology and methodologies for all implementation flows starting from synthesis, DFT to all the way up to sign-off checks and ATPG generation but also manages all infrastructure resources like disk,CPU's, computation time etc.

3.2 RTG Architecture

RTG architecture is a five layer architectutre build over the RTG shell which holds all the flows and methodologies required for the design implementation cycle where all flows do automated signoff via pre defined release mechanism to avoid kind of data pick up errors.The layer 1 holds all the technology and methodology features for design with required configurations using simple variable equal to value format.,which can be easily configured by the user.The system configuration and run generation layers read the setup files and provide required setup and dependencies to user which gets execute automatically.The user will have a GUI where they can monitor their runs analyze tables and graphs,create branch runs from specific intermediate node.The flow also includes an inbuild data processing and analysis(DPA) which gets executed automatically.The users will have a GUI where they can monitor their runs analyze results and graphs.

The DPA engine where user can run multiple experiments with different PPA (power,performance and area) targets and all it ca tell which once worked best and stop other rusn to save on resources.

For example to evaluate CTS quality in PnR flow different VT type users need to tell type of VT with single varaibles and start all branch experiments from place optimized databases with right PPA priorities and RTG wiil not only execute all the runs automatically but also give tabular and graphical using DPA engine to make quick decision which in current bases system takes anywhere from few hours to couple of days based on number of experiments and clocks In design,the DPA engine also has capabilities to execute the lower dependency nodes only of specific QoR is achieved by previous nodes.

3.3 Execution Flow

Execution Flow:Setup files are the simple input text files with which the users can interact with the flow. Setup are the settings having a specific task to do in the flow as per the value they have been .They are used to tweak/update/maintain the flow.They are defined with the syntax.

variable=value.

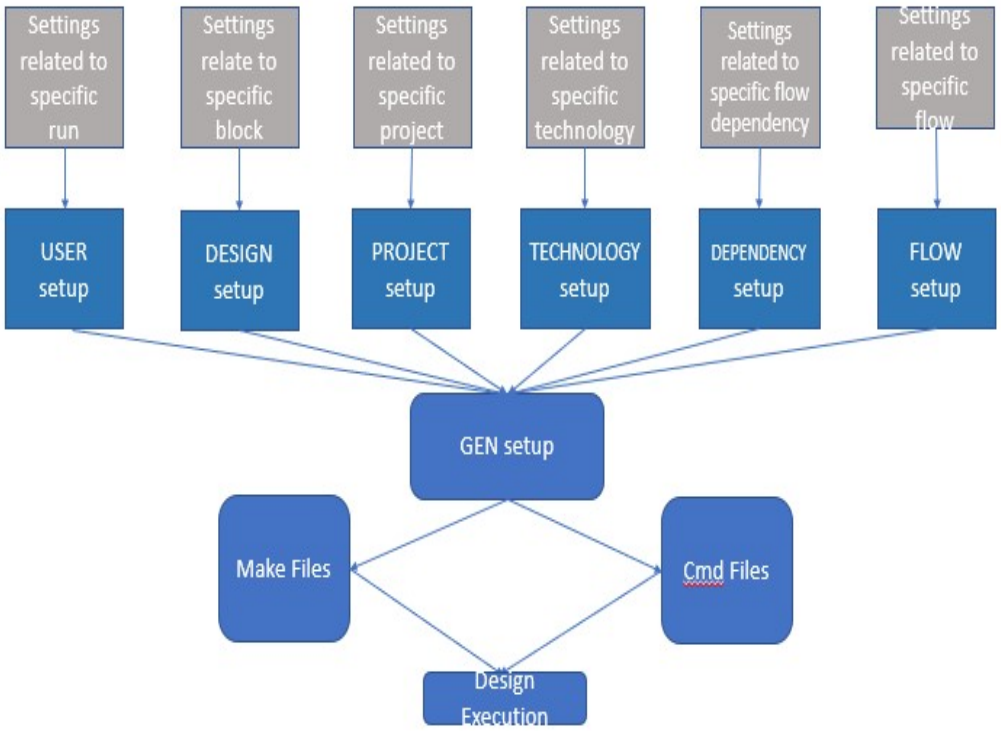


Fig 3.2 Flow Execution Diagram

Different setup files used in RTG flow

1. USER setup
2. DESIGN setup
3. PROJECT setup
4. TECHNOLOGY setup

5. FLOW setup
6. DEPENDENCY setup

The hierarchy setup of setup files is shown in fig 3.2,the files and under which purpose they are used.

3.4 Execution Flow Files

User.setup have the following information

1. Design Name -→ top module name of the block running.
2. Branch Name→Type of flow e.g. PnR ,ECO,FCIERM.
3. DIR_TO_RUN→Directory in which RTG flow should start.
4. RUNNMAE→User name to differentiate from other runs.
5. TREE_NAME→Impl or FCHIP

Illustration of user.setup file is as:

DESIGN NAME	=	sldpc top
BRANCH NAME	=	PnR
DIR TO RUN	=	/pwd
RUN NAME	=	RunOne
TREE_NAME	=	Impl

There are several users that are running the automation platform and running different runs according to the use.To have a common platform and unified structure across all users and hierarchy is maintained, a standard directory strucucture is created.This standard structure creates a uniform flow of thd design even in a huge engineering group also.

Fig 3.3 shows a typical directory structure of the design.

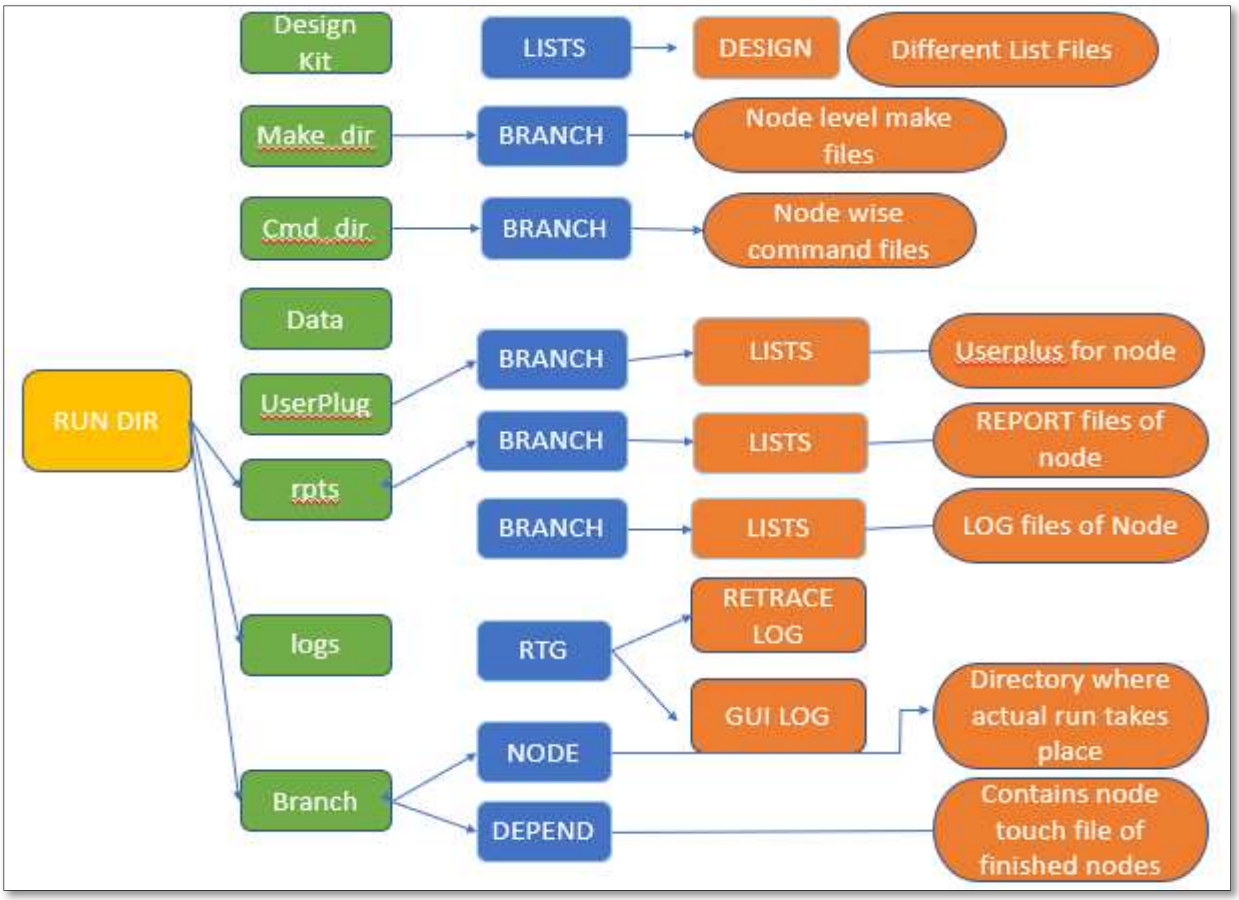


Fig 3.3 Directory Stucture of RTG Automation

3.5LibGeb Flow

Lib Gen Flow:A complete ASIC library mangement engine.

ASIC contains several building blocks like digital block,analog memory,memoryIP's,IO's and other custom macros.All these building blocks are delivered as a library input for a particular project.It makes library inputs as critical elements throughout the ASIC design cylclelibrary consist of several types of modeling of IP s which are used in the design. For example timing models, physical models, power model. DFT models, behavioral models etc.

This library data can go up to 100s of GB with 90-100 different views spread across 1000's of text and binary files required for each stage of ASIC development cycle.Managing such a huge data is a big challenge and manual method of qualifying them is a nightmare.Figure3.4 shows aconventional design execution considering the scenario explain in above diagram, it is obvious that the accuracy and availability of views is one of the most important aspect of ASIC design cycle.

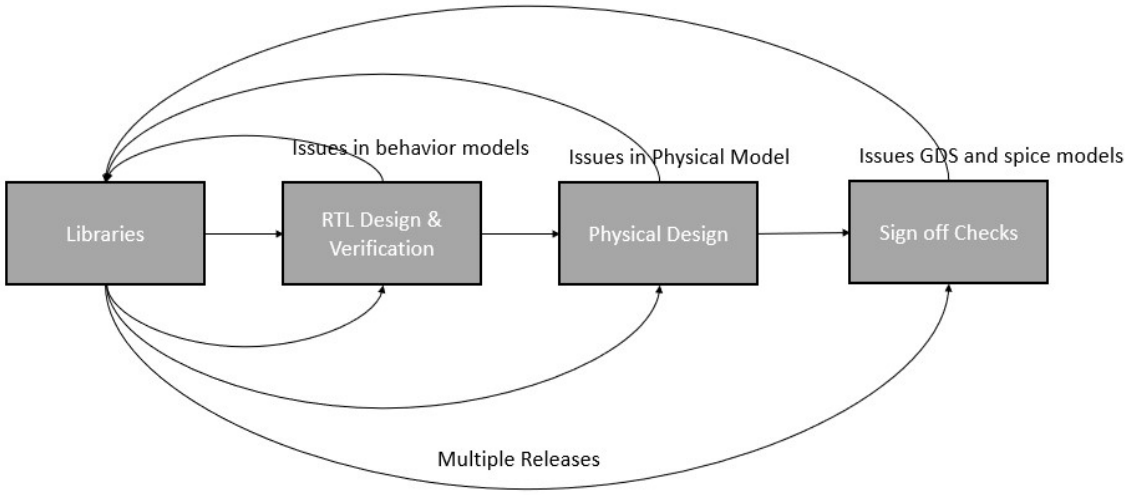


Fig 3.4 Conventional Design Execution

It is impossible for any individual or team to manually verify every aspect of delivered libraries. Any issues in library like naming mismatch, missing views, missing cells, missing or incorrect details can lead to serious consequences from loss of man hours to schedule impact to Chip failure.As a solution to this critical problem,LibGen Flow - simple and highly effective library management engine for all ASIC development projects.

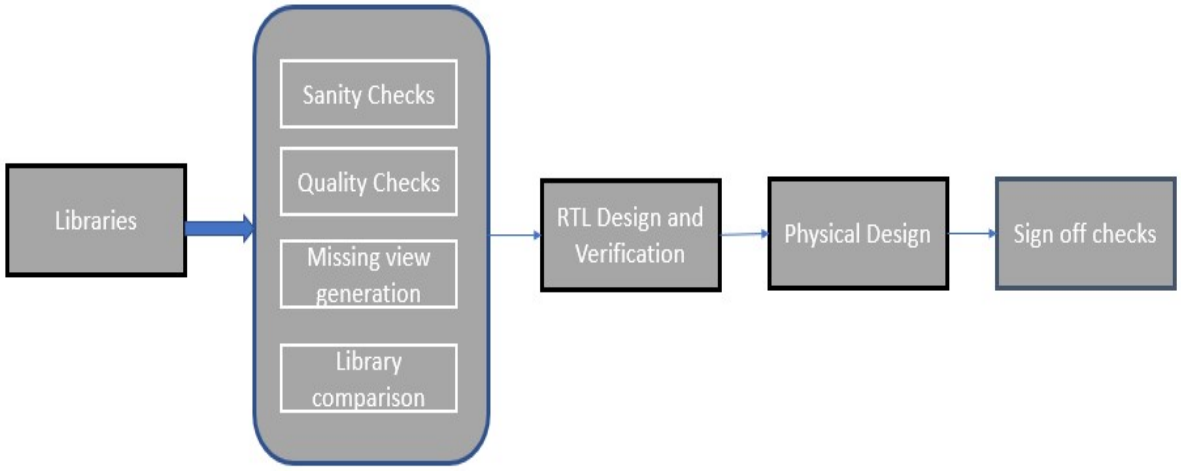


Fig 3.5 Lib Gen Flow

LibGen flow as shown in Figure 3.5 takes library inputs from different sources and run exhaustive analysis and segregates them based on lib view, VT type, lib corner, technology and link them in a design kit with a standard naming convention. It also have pre identified sanity and quality checks which will be performed before making any IP/foundry libraries available for execution flows.

Advantages of LibGen flow in Design cycle is as follows:

1. Provides more time margin to concentrate on design issues by fixing all library issues ahead of the design cycle Standard naming convention across all the libraries across all the vendors Integrated Sanity Checks. For example missing of libs, missing lefs etc.
2. Integrated Quality Checks. For exampleLefVsGds, LibVsLef, VerilogsLibetc.

3. Integrated missing library models generation engine.

4. GUI based Review and publish window with easy to read email with adequate summary and waiving/reject mechanism.

5. Eliminates last minute glitches due to incorrect libraries and helps to predict the design cycle Benchmarking between technologies by comparing critical parameters like leakage, area, timing etc. Which speeds up the IP selection process.

6. Benchmarking between technologies by comparing critical parameters like leakage,area,timingetc.Which speeds up the IP selection process.

The LibGen flow as shown in fig 3.5 contains Sanity checks in the design execution flow and these checks are run before the release of RTL for the synthesis.It is the first time that RTL is attached to the technology node at which it has to be fabricated eg 16FF,7nm technology node.To ensure that the RTL design is functionally correct and free from any errors.Various sanity checks need to be performed on the ASIC design.The problem with these sanity checks is that it is huge set of test's, and every test need be run.The tests contains different set of conditions and scenario.To make sure all the test's are passing, designer and verification has to run these test and maintain a report of important information from the logs of test.For correct decision on test and codes are corrected to perform correct fucntionlaity.Manual running of these test's takes several hours and some time even days to obtain result.

To eliminate the manual running of the tests and reduce the excution time,automated running of tests helps in avoiding tedious and long running jobs.More automated test running provides a regular reporting structure to the designer with all the important information from huge log reports.The next chapter discusses in detail about the Automation of test and how the automation helps in a unified quality checks on the design.

CHAPTER 4

UDM Implementation

4.1 UDM Overview

UDM-Unified database management system is a standalone database management system to bring out the consistency across all ASIC design groups.The UDM provides a standard quality checks platform to the SOC design and verification teams, so that the rigor of quality checks are maintained in the design and verification process of SOC.Unified database management system is a unified continuous integeration flow and automation of quality checks in design verification in RTL LibGen'sflow.The UDM structure is defined by the user ,what the quality of regression for the block under verification should be,before the RTL codes are merged in main branch of Bit Bucket repository and RTL is synthesised.

There are various quality checks being performed on an RTL and different methodologies are being adapted in a project for RTL code merges.Some of the methodologies that are being performed in a verification of design process are as follows.

1.Tagged release:In tagged release process, tags are released by respective user and updated on a common platform by indivisualusers,by manually running the test and after a decision is taken whether a regression is successful or unsuccessful in its run.The codes are then merged in main branch repositories depending upon how many tags were successful and overall regression has passed the minimum threshold criteria of quality checks.

2.Bit Bucket and Jenkins:It is a two staged process of merging codes in the repositories ,this kind of approach is applied in cases where the ASIC has more complex functionality and many sub branches of a tree (main branch) are created and code merger requires a more stringent checks.In this two staged process,the regression is first run through Jenkins (automated test running),by taking the latest RTL codes from the git repository and then status of test results are displayed to user. Depending upon whether the Jenkins build was passed or failed the code merges in bit bucket is allowed accordingly.

Jenkins is a CI and CD (continuous Integertation and continuous Deployment) tool,which is used for test automation and code mergers in RTL code repository.Jenkins reads the latest code from bit bucket ,runs the test in local machine and exit with a criteria 0 or 1. Jenkins exiting with 0 means the regression has passed and code merger in main branch whereas if the exit value is non zero then regression has failed and code merger is not allowed until the regression runs are clear.

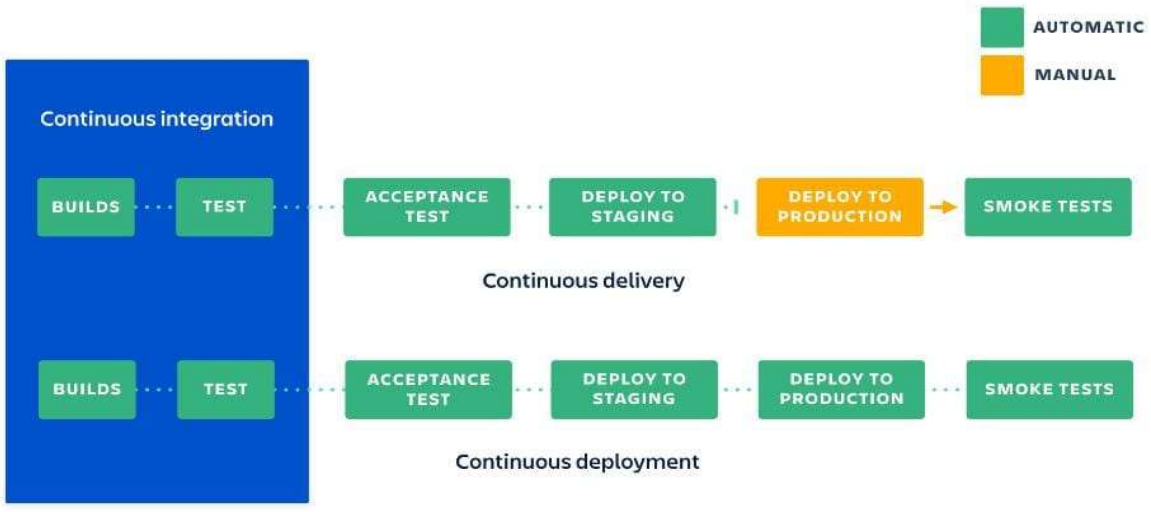


Fig 4.1 CI and CD execution Cycle

4.2 Manual and Automated Test Runnig

In SOC design cycle from RTL2GDS phase, RTL design has to undergo for verification process,in design verification process the block under verification has to undergo through various TEST conditions to ensure the desing functionality and purpose of design is achieved.As discussed in LibGen flow the Sanity Test's are the test's ,which used to ensure that SOC RTL design is clean and the basic functionality of design is met.SOC deisgn is simulated,emualated on various industry standard platforms to verify the design like FPGA, Palladium which are different emulation boards to verify the design functionality.

To verify deisgn functionality 100's of TEST cases are written in using Verilog,Sytem Verilog methodologies and as the design specification changes,these TEST cases need to be re written /modified according the specifications of design .The design and verification engineers need to continuously run these test on a specified interval of time and maintain the quality check parameters as discussed in introduction.Manually running of these tests is a long process and consumes more man hours ,which increses the project time to market and cost of the product.Instead of running these tests manually ,test automation is performed ,where these tests are triggered automatically and status of these test reports is communicated to the end user, and test's results are available to them.

Fig 4.2 explains ,in the design verification process the sanity regression are runs for ASIC, FPGA and Palladium ,each of these runs have different number of tests in its TEST list depending upon the functionality need to be verified.These tests are run manually by the designer and they wait for the test reults ,whether they are pass or failed ,depending upon the status of result they need to make the

necessary changes in the design.Doing all these runs manually takes hours,in some cases days so it increases the wait time for the user.

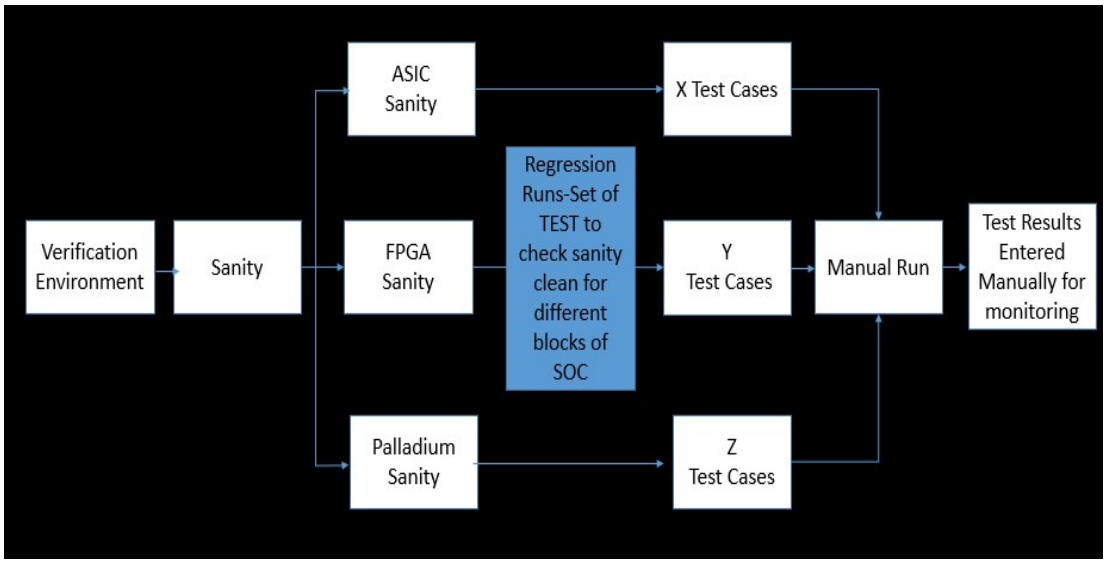
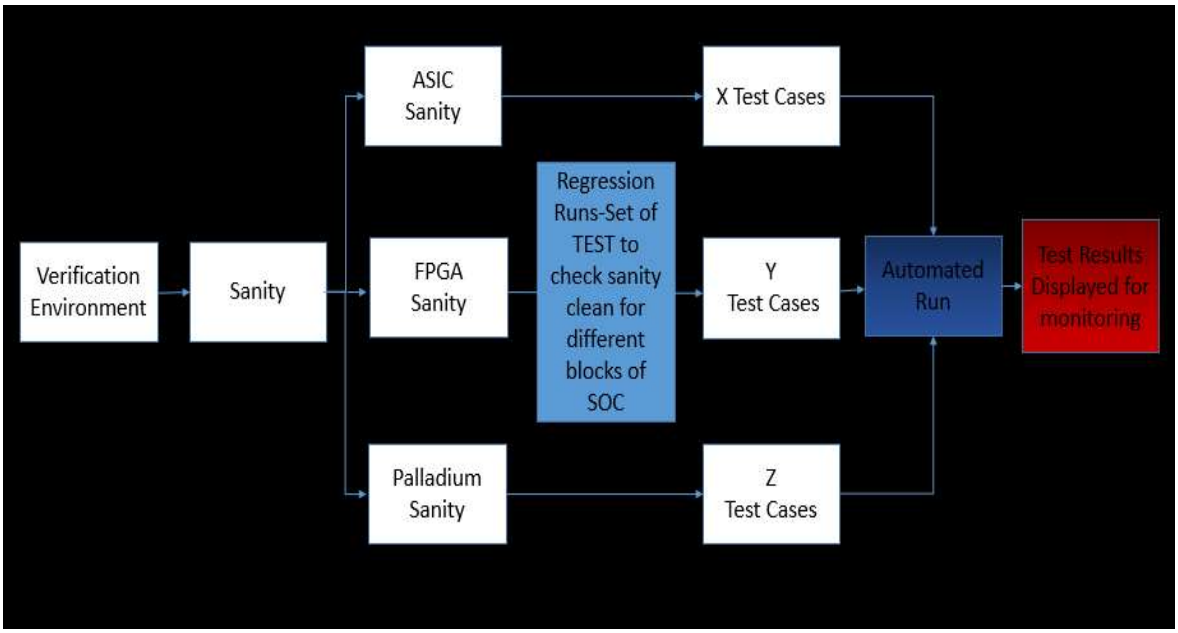


Fig 4.2 Manual regression run Cycle

The above process involves manual test running and manual test reporting at a common platform,this process however can be automated and automated running of these thesetest's can be done with the help of CI nad CD platforms like Jenkins and internal scripting in local to ensure the correct functionality of automation in design.The fig 4.3 shows the TEST automation methodology being adapted ,bringing down the manual test running and a standard plaform for quality checks in ASIC design.The automation of TEST's are achieved with the help of scripting written in Perl,Shell and python ,with the help of which well methodology is defined for the automation.



4.3 Automated regression Run

4.3 Work Flow of Test Automation

Test Automation is achieved using python and shell scripting.Thework flow of automation is defined in fig 4.4. The shell script is the main script which contains all the terminal instructions to for running the test's.Shell script is defined as the main file(.csh file) it has all the commands to run the test's .

The *.Csh creates a parent directory (Project specific directory eg. Project A) ,in parent directory Run Directory(Directory of a particular time instant eg. RUN_DIR_2020:06:20_*time),in Run directory the test regressions are launched.These Run directory launches multiple test's at the same time in separate Directories(Dir 1,Dir 2…..Dir N),each directory has the details logs of each test.These log files has all the relevant information of the TEST result e.g. status of the TEST,simulation results,compilation errors if any, JOB ID's,memoryoccupied,time taken for the result and CPU usages of the test .These test results are accumulated by the help of Python script and an Email is sent to

the user containing the relevant information of the regressions.The regression has set of test,whose results are accumulated and then sent to user.The overall scenario is controlled by Jenkins,which triggers the test's by envoking the Main file at a regular interval of time set by the designer.The environment created around the verification process are called wrappers.The wrapper work to automate the Verification and Design process flow.

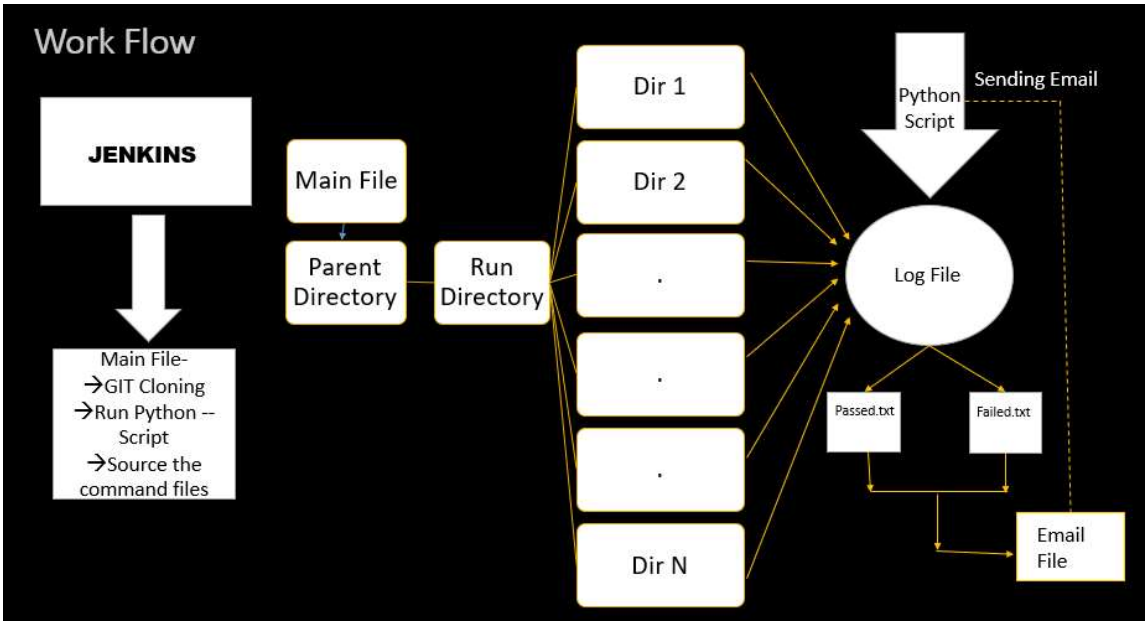


Fig 4.4 Test Automation Flow

4.5 Scripts Functionality

The scripts written to support verification work flows are called wrapper scripts.The wrapper scripts are combination of one or more platforms to support the overall functionality.The functionality created for UDM Implementation are combination of Shell and Python scripts.The detail functionality of Scripts are discussed below.

4.5.1 Shell Script

Shell script(*.csh) script is the main engine of the Santiy TEST automation.The test that are running are verification test to verify the functionality of the Block under Verification.

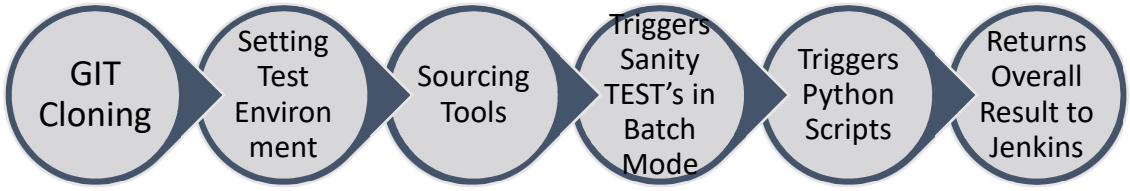


Fig 4.5 Pipeline of Shell Script working

The detail functionality of script is as:

1. Create Parent directory and Run Directory for running the regression in local environment of the user.
2. The shell script then invokes the lates RTL from the central repository and stores it in the local.
3. It then sets the TEST bench environment for running the test's it includes

a) Sourcing the Tools file(*.csh) file which contains the instruction to set the tools required to run theTest.These tools include compilation,simulations,debugtools.All these tools perform their the indivisual functionality

4. This Script also triggers the Pyhton Script integerated with the Shell script to carry its functionality
5. Returns overall results to Jenkins after the regression has been run completely to mark the build as successful or failure by exiting with value 0 or 1.

6. The scripts passes the relevant information to the Python module containing the details specific to the Sanity eg ASIC Sanity or FPGA Sanity and the number of test's that are supposed to be launched in regression.Fig 4.5 shows the pipeline view of the SHELL scripts functionality.

4.5.2 Python Script

Python script is another integerated script in wrapper tools environment which carries out the main functionality to collect the all the relevant information from the Logs and display it to the user for the decision making on the Test ,whether the test needs to be removed,RTL needs correction etc.

Major functionlaity of Python Script Includes

1. Some scripts have to be made Soft coded so that it is more modular In its design and can carry out major functionality of the project under different platforms like if it used for ASIC, FPGA or Palladium Tests running.
2. The scripts is invoked as soon as the Tests are launched in the grid and now test monitoring need to be done.It counts and Track the status of Test's result.Whether all test result have been reported in the Log report if not it need to wait until all the results are not logged.At the same time the script need to continuously keep a record of number of failed and passed test and their total matches the actual number of Tests.

If (Total count =Passed Count + Failed Count)

Next module

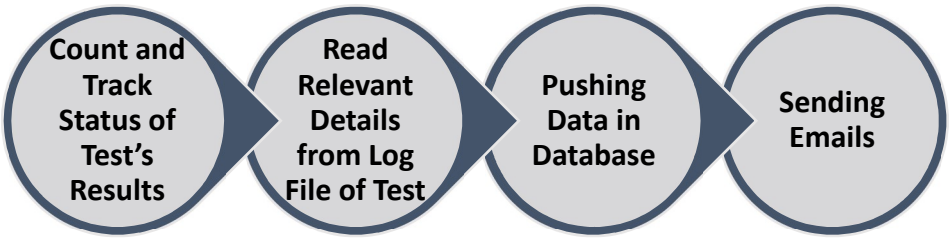
Else

Wait

3. The scripts reads all the relevant information from the log report of each Test.

4. The data for future enhancement and modification has to be pushed into central database that is achieved with the help of MongoDb module of python script.

5. The other major function of Python script is to Send Email with all the relevant information to the user.Thepipleline of the tasks achieved by the python wrapper is shown in fig.



4.6 Pipeline of Python Script Working

The flow diagram in fig. shows the all relevant information that is being displayed, it also shows the purpose of the each information being displayed.

- Test Name→Name of the test eg FIM BLOCK Regs
- Test Status→Passed/Failed
- Memory occupied→It is the memory that was occupied by the test at the time of exection of run.

•Memory Reserved→Before the regressions are run ,ceratain amount of fixed memory is being set for the TEST to run,there are cases when test takes more amount of memory than reserved for it.

To monitor which test took more memory for running,this data is being dispalyed .The test that takes more than reserved memory,they need to be optimsed and re-written.

- Time taken→The duration of time the indivisual test ran need's to be monitored .The test taking more amount of time ,they also need optimisation and their test benches need to be re written and optimised.

Fig 4.7(a-b) shows the flow diagram of the data being displayed to the user and their relevance.

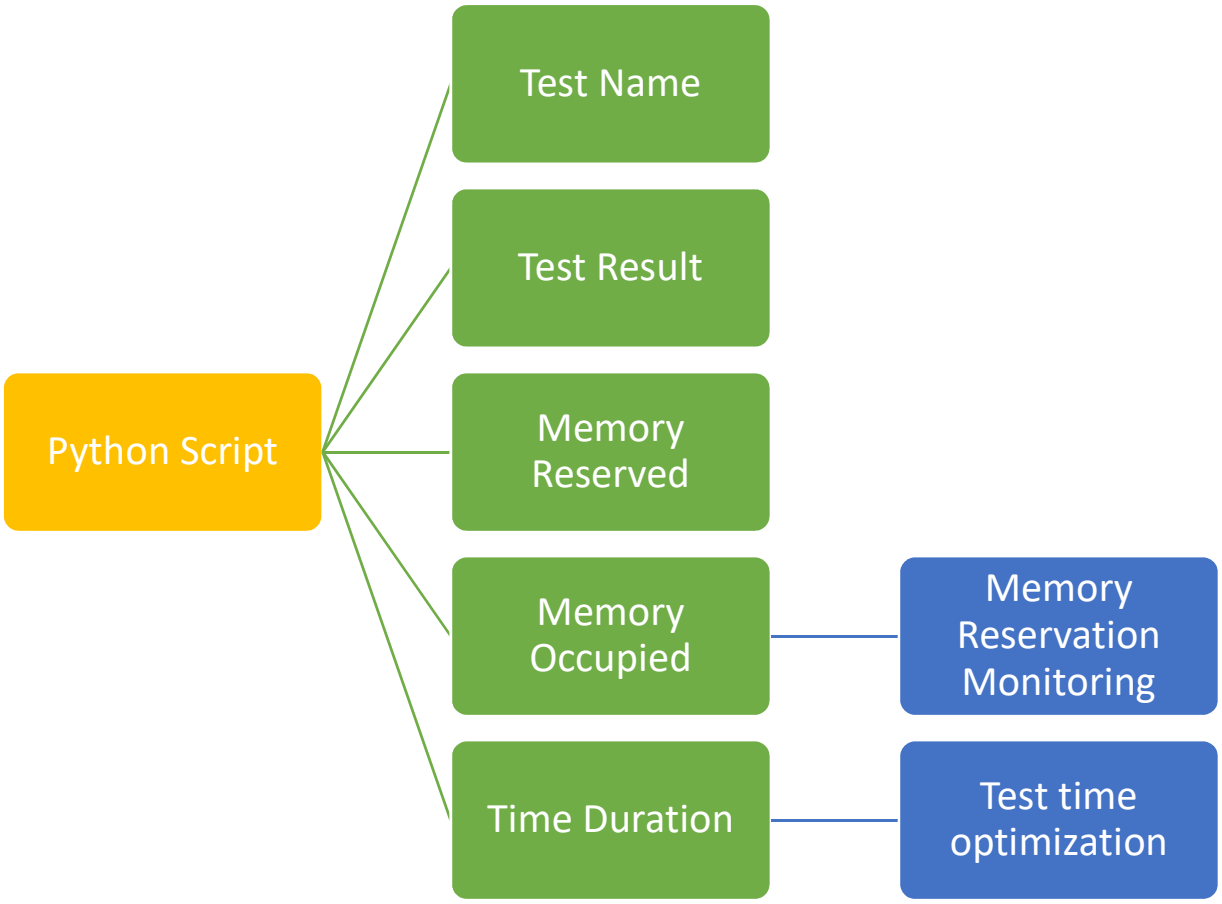


Fig 4.7(a)

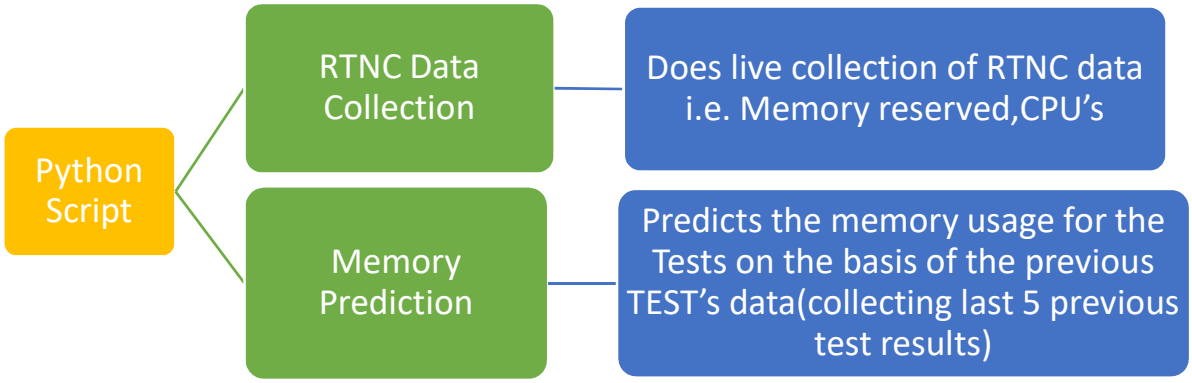


Fig 4.7(b) Python script flow execution of data

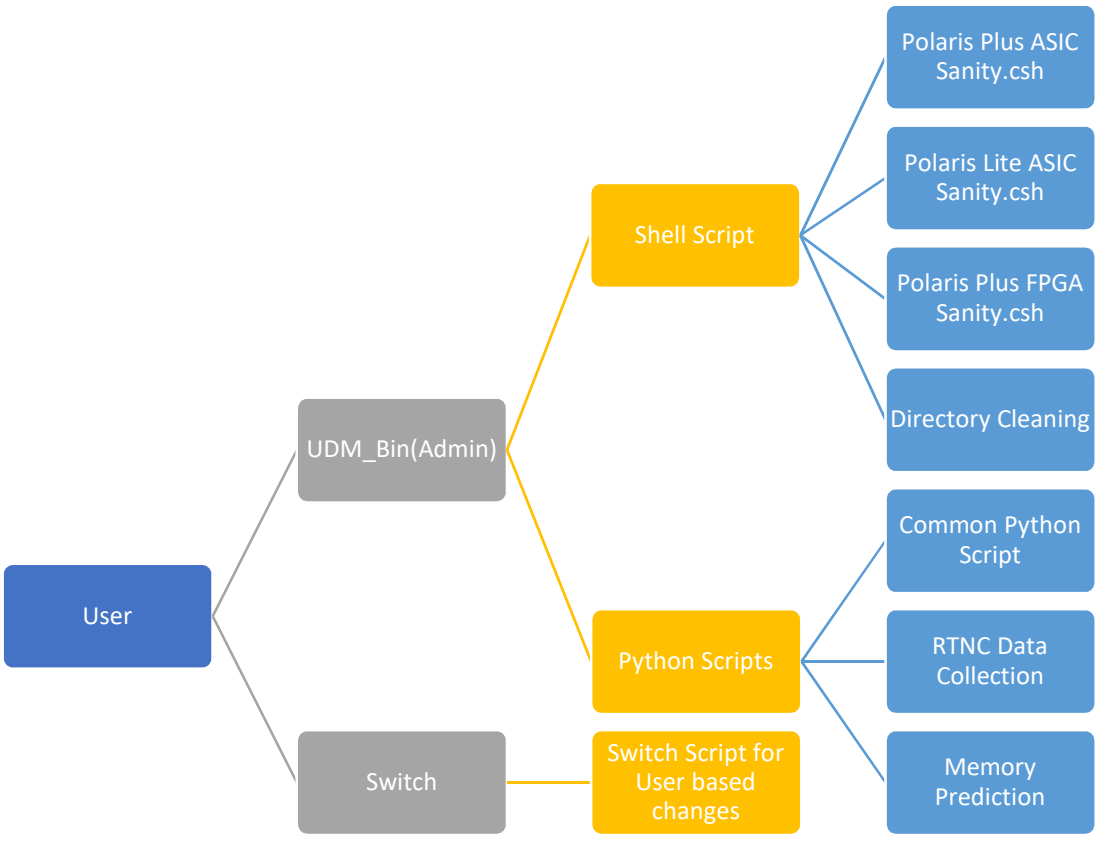


Fig 4.8 Directory structure for execution

The directory structure for each regression for the wrapper setup is shown in fig 4.8. The directory structure is such that its can be downloaded to any user from SVN repository and can be run into local account of the user.As discussed in previous chapter uniform directory structure is important to maintain a uniform flow of design for all the users and the same infrastructure can be re used across multipe projects and ASIC's.The fig 4.8 shows the directory structure being created for Sanity checks .It is broadly classified into Shell and Pyhton bins ,where each bin contains the relevant scripts ans can be directly run to obatain the functionality.

CHAPTER 5

CONCLUSION

RTG automation provides a wide range of advantages and reduces the overall life cycle of a complex ASIC design.RTG is a fully automated GUI Based platform, easy to configure , prioritize tasks, auto schedule runs ,maximize license and compute utilization.Automation provides minimal errors Automated QoR and progress reports enable predictability It provides faster convergence ,ensure efficiency and consistent implementation across design.The uniform data base structure enable resource mobilization across ASICS's.It enable smooth and error free hand-off. RTG automation isadaptable,hasintegrated methodologies for multiple foundries and technology tool agnostic,dynamic node generation and resource reservation.

5.1 Results and Future scope

The table 5.1 shows a the typical reporting structure of the test's results whose relevant information is being displayed to the user.The information includes the Test number, status of the test result (passed or failed), memory it took to execute in background.The memory occupied details is important for the user to ensure that the memory occupied is under the permisable reserved limits.

The accumulation of the memory occupied for previous few runs would help in predicting the future usage of the memory and the memory usage could be made dyanamic instead of static usage,which is currently being done.Memory occupied also helps to optimize the RTL codes so that it takes less memory to run and resources could be optimally utilized.The other important parameter is being

shared to the user is the time taken by the test to run,time is one of the most important parameter which every designers and verification engineer want to reduce,which not only saves the resouces but also the time to market and giving a leading edge to the product,hence time optimization is also a very important key parameter.

Table 5.1 Regression reporting structure

Test	Test Status	Memory occupied	Time Taken	Memory Reserved
Test 1	PASSED	885	389.5	1000
Test 2	PASSED	1025	2233.7	1000
Test 3	FAILED	885	2369	1000
Test 4	FAILED	850	974	1000
Test 5	PASSED	950	1950	1000
Test 6	PASSED	1000	440	1000
Test 7	PASSED	1250	89.9	1000
Test 8	PASSED	880	124	1000
Test 9	PASSED	870	322	1000
Test 10	PASSED	870	949	1000
Test 11	PASSED	1600	1101	1000
Test 12	PASSED	860	540	1000
Test 13	PASSED	872	411	1000
Test 14	PASSED	950	974	1000
Test 15	PASSED	850	1063	1000
Test 16	PASSED	850	540	1000
Test 17	PASSED	870	52.5	1000
Test 18	PASSED	860	453	1000
Test 19	PASSED	852	84.5	1000
Test 20	PASSED	1120	58.3	1000

The fig 5.1 shows the memory and time being utilized by each test.

It is important to note that the memory utilization is constant in maximum of test cases ,where as the time utilization by certain tests are more then few several thousands.These test decides the over all time of a regression it takes to run.The less the time takes,lesser the duration for which resources are occupied and thus reducing the overall quality check duration of a SOC.

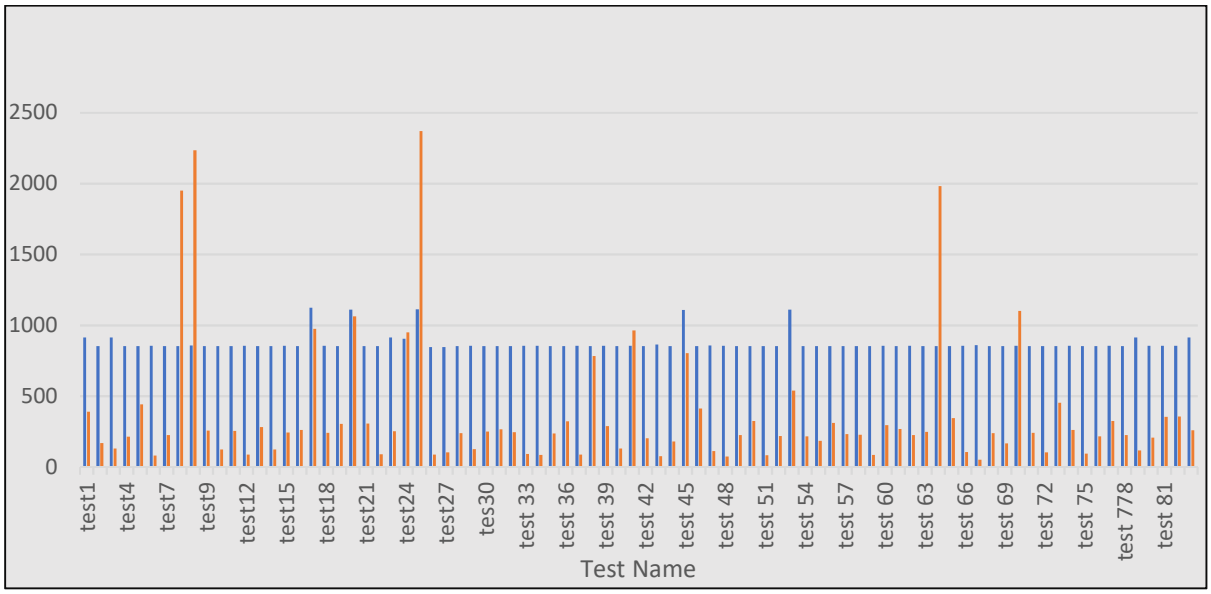


Fig 5.1 Time and Memory utilization

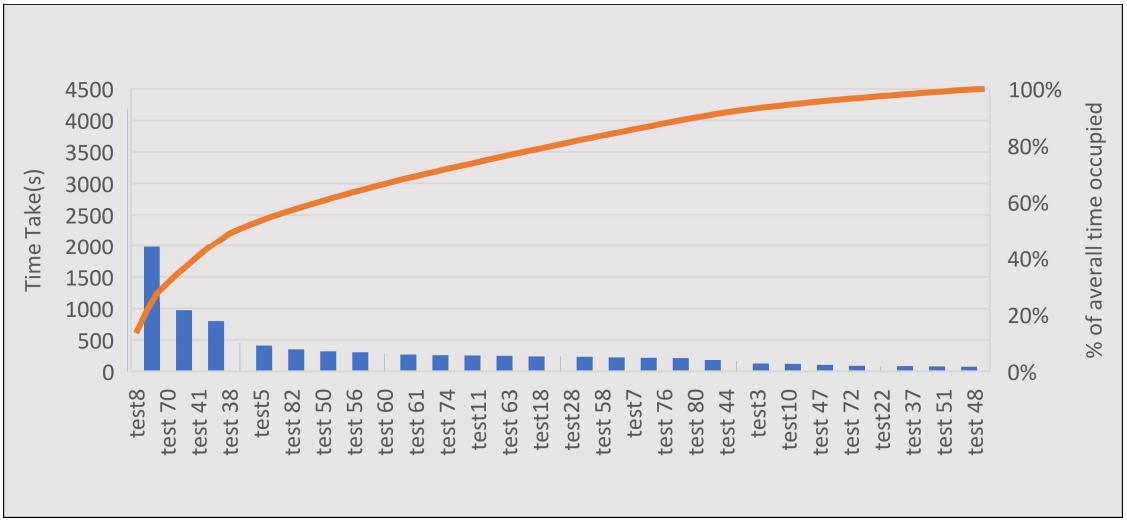


Fig 5.2Time utilization of Test's

Fig 5.2 and 5.3 shows the time and memory utilization indivisually.The time graph gives important insights about the test's that take 80% of the quality check time and are only 20% of all the total tests being run.Optimizing the 20% test would lead to reducing the overall desing cycle of the project and less resource

utilization.Fig 5.3 shows the memory utilization of the test's,it is observed that certain test's take more than the specified limit of the reservation.

The reason forthe large memory utilization is these tests are run a large Blocks and has various complex scenarios in test assertions. Running complex assertions requires more resources to run and more time to run.

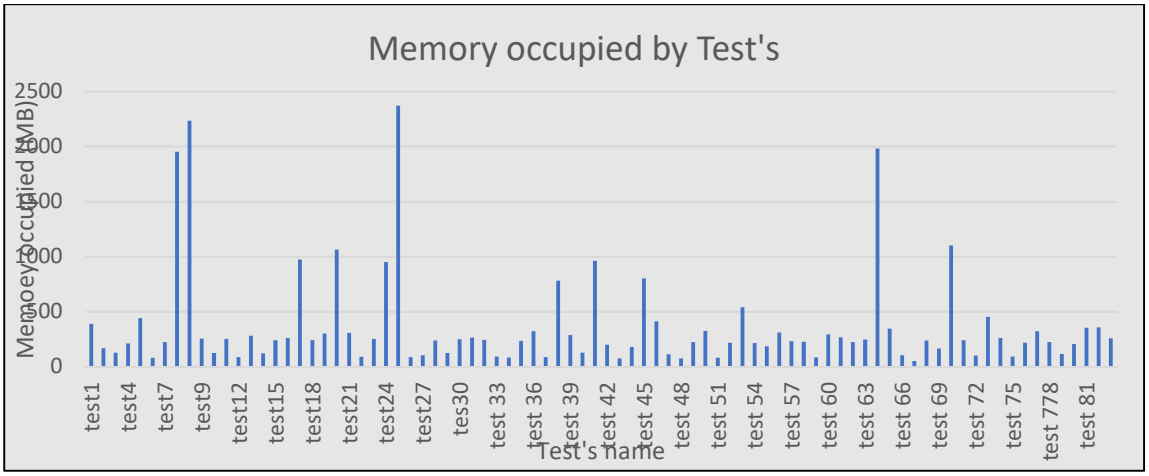


Fig 5.3Memory occcupied

In order to reduce the time and memory utilization ,it is important to write multiple number of test assertion in different test beches and reducing the complexity of design.However reducing the complex test benches is not an easy task.Reducing the complex verification test's requires to segregate the codes in various functional module ,the dividing of the tests is simple modules can be done with the help of machine learning algorithms .The algorithms identifies the repeated modules in different test benches and makes independent separate module ,which can be run once for all blocks.These are the future enhcnacment where automation with machine learning works to improve the life cycle of project.

REFRERENCES

[1] S. N. Adya, M. C. Yildiz, I. L. Markov, P. G. Villarrubia, P. N. Parakh and P. H. Madden, “Bench-marking for Large-Scale Placement and Beyond”,IEEE Transactions on Computer-Aided Design ofIntegrated Circuits and Systems23(4) (2004), pp. 472-487..

[2] Arnold S. Tran ; Richard A. Forsberg ; Jack C. Lee , “A VLSI Designverification strategy”,IEETrasn. IBM Journal of Research and Development.

[3] Robert Brayton ; Luca P. Carloni ; Alberto L. Sangiovanni-VincentellizianoVilla , “Design Automation of Electronic Systems: Past Accomplishmentsand Challenges Ahead”, Proceedings of the IEEE (Volume: 103 , Issue: 11 , Nov. 2015)

[4] TrpimirAlajbeg, MladenSokele, "Implementation of Electronic design Automation software tool in the learning process", Information and Communication Technology Electronics and Microelectronics (MIPRO) 2019 42nd International Convention on, pp. 532-536, 2019.

[5] ChammireddyS,KothariP,BajaniP,SinghG , “LibGen- A complete ASIC library managemt engine”, ISTC ,Sandisk ,2017

[6] A.E. Ruehli ; G.S. Ditlow , “Circuit analysis, logic simulation, and design verification for VLSI”,Proceedings of the IEEE (Volume: 71 , Issue: 1)

[7]R. Aitken, G. Yeric, B. Cline, S. Sinha, L. Shifren, I. Iqbal and V. chandra,“Physical Design andFinFETs”,Proc. ACM International Symposium onPhysical Design, 2014, pp. 65-68.

[8] Guirong Wu, Song Jia, Yuan Wang and Ganggang Zhang, "An efficient clocktree synthesis method in physical design," 2009

IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC), Xi'an, 2009, pp. 190-193, doi: 10.1109/EDSSC.2009.5394159.

[9] U. Brenner and J. Vygen, “Faster Optimal Single-Row Placement with FixedOrdering”,Proc.Design, Automation and Test in Europe,2000, pp. 117-121.

[10]F. Brglez and H. Fujiwara, “Recent Algorithms for Gate-Level ATPG with faultSimulation and their PerformanceAssessment”,Proc. IEEE InternationalSymposium on Circuits and Systems,1985, pp. 663-698

[11] R. Carden and C.K. Cheng, “A Global Router with a Theoretical Bound on theOptimal Solution”,IEEE Trans. on CAD15(2) (1996), pp. 208-216.

[12] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze and M. Xie, “mPL6: Enhanced Multilevel Mixed-SizePlacement”,Proc. ACM International Symposium on Physical Design, 2006, pp. 212-214.

[13] C. Condrat, P. Kalla and S. Blair, "A methodology for physical design automation for integrated optics," 2012 IEEE 55th International MidwestSymposium on Circuits and Systems (MWSCAS), Boise, ID, 2012, pp. 598-601,doi: 10.1109/MWSCAS.2012.6292091

[14] M. R. Guthaus, D. Sylvester and R. B. Brown, "Clock tree synthesis with data-path sensitivity matching," 2008 Asia and South Pacific Design AutomationConference, Seoul, 2008, pp. 498-503, doi: 10.1109/ASPDAC.2008.4484001.

[15] M. R. Guthaus, D. Sylvester and R. B. Brown, "Clock tree synthesis with data-path sensitivity matching," 2008 Asia and South Pacific Design AutomationConference, Seoul, 2008, pp. 498-503, doi: 10.1109/ASPDAC.2008.4484001.

Visesh Thesis report

ORIGINALITY REPORT

7%	2%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Sridhar Gangadharan, Sanjay Churiwala. "Chapter 1 Introduction", Springer Science and Business Media LLC, 2013 Publication	3%
2	escholarship.org Internet Source	1%
3	Submitted to Newham Sixth Form College, London Student Paper	<1%
4	Guirong Wu, Song Jia, Yuan Wang, Ganggang Zhang. "An efficient clock tree synthesis method in physical design", 2009 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC), 2009 Publication	<1%
5	ethesis.lib.cycu.edu.tw Internet Source	<1%
6	hdl.handle.net Internet Source	<1%

7	dblp.uni-trier.de Internet Source	<1%
8	Submitted to University of Edinburgh Student Paper	<1%
9	Submitted to University of Nottingham Student Paper	<1%
10	Ajay Kumar G C, K N Subrahmanya, Sajeesh Ammikkallingal, Sai Anudeep Polisetti. "Physical Design, Power and Area Optimization of High Frequency Block at Smaller Technology Node", 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2019 Publication	<1%
11	Submitted to Birla Institute of Technology Student Paper	<1%
12	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
13	Eric Schneider, Hans-Joachim Wunderlich. "SWIFT: Switch-Level Fault Simulation on GPUs", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018 Publication	<1%
14	scholar.lib.vt.edu Internet Source	<1%

15	www.freepatentsonline.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		