International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 6-13 Jan 2019-June 2019 www.csjournals.com

Constraints Based Approach of Floorplanning for ASIC Development

Rajendra Patel, Sachin Pandya
[12]Marwadi University, Department of Electronics and Communication Engineering, Rajkot, India
[1]rajendrakumar.patel@marwadieducation.edu.in, [2]pandya.sachin92@gmail.com

Abstract: Due to the increase in the complexity of the VLSI designs, the back end designing, especially floorplanning has become an important task. In this paper, a constraints-based approach is proposed that helps to place the macros at appropriate locations in the core area to get better timing results, power consumption and routability of the design. The required data to calculate the dimensions of the core and the die are fetched from the library files using TCL scripts. The proposed method resizes the die as per the requirements to place and route the hard macros and standard cells in the design, calculates the dimensions for the rings and stripes for better power distribution in the chip and finds the locations for the macros in the core area by analyzing their inter-connectivity with IOs, other macros and standard cells. The experimental results show that the floorplanning done with the help of the constraints gives better timing, power consumption, and routability as compared to the floorplanning done by the tool using its default method.

Keywords:ASIC**,** Back-end Design Flow, Floorplanning, Physical Design.

Introduction

In last decade, an improvement has been seen in the semiconductorindustry and now, the size of the transistors is reduced from Micrometers to Nanometers range. The progress of the VLSI technology has increased the complexity of the design where there are millions of layouts on a chip. As the high complexity of the chip, to provide improved chip performance, these tools have a role that is important. In VLSI back-end design, also called as physical design, the quality of the layout depends on how good the floorplan is. Hence, floorplanning is a crucial stage. The estimation of total area for the layout, delays, and congestion can be done after floorplanning stage. The objective of the floorplanning is to locate the circuit modules on a chip and decide their orientation in such a way that they do not overlap one another.

Overview of VLSI Design Process

The term VLSI is an abbreviation for “Very Large-Scale Integration”. Millions of transistors are fabricated on a single chip. The flow of VLSI design consists of different sequential steps: system specification, architectural description, logic description, circuit description, physical design, fabrication, packaging, and testing [6]. Fig. 1 shows the design steps of VLSI flow.The flow begins with system specifications. In this step the specifications such as functionality, physical shape and dimensions, and performance, design techniques, fabrication technology etc. are determined. The specifications such as functionality, power, and area are decided by the market demand and economic perspective [8].

Floorplanning

The floorplanning has become a challenging task because of the growth in the VLSI system density and complexity. It has an important role to play in the performance of the system. A VLSI circuit is made of a number of rectangular or rectilinear modules which are having variable dimensions and are interconnected by nets [9]. Floorplanning is the starting point of VLSI physical design flow [1]. It determines the locations and dimensions of blocks in a chip so as to minimize the chip area and interconnect wirelengths [1].The Floorplanning is nothing but a kind of placement that is having embedded flexible modules. The modules are such that their area is fixed but the height and the width can be varied according to the aspect ratio constraints [10].

A. Floorplanning Description

The objective is to decide the positions and dimensions of the rectangular or rectilinear modules or Intellectual Property modules on the core area in such a way that the performance of the circuit is optimized. The following description explains a complete representation of the floorplanning problem [11].

Generally, the inputs for a floorplanning problem are given as follows:

- a set of n rectangular modules B = {1 2. . . n) with a list of areas ai, $1 \leq i \leq n$;
- a partition of S into sets B1 and B2 representing the modules with fixed and free orientations respectively;

- an interconnection matrix $C_{n \times n} = [C_{ij}]$, $1 \leq i,j \leq n$, where Cij captures the connectivity between modules i and j. (we assume C is symmetric, i.e.,$C_{ij} = C_{ji}$, given by netlist);
- values a_i for tile area of each module i;
- bounds R_{ilow} andR_{iup} on the aspect ratio R_i of each module i;
- bounds w_{Flow}, and w_{Fup}, h_{Flow}, and h_{Fup} on the width and height respectively of the floorplan, for an instance of outline free floorplanning
- values w_F and h_F for the width and height of the floorplan, for an instance of fixed-outline floorplanning.

The output of the floorplan gives the information the dimensions of and location of each module on the core. This output should fulfill the following:

- There is no overlap between the modules.
- Coordinates (xi, yi), height hi and width wi for each module such that wi × hi , $1 \leq i \leq n$.
- Rilow$\leq$ hi / wi $\leq$ Riup or Riup$\leq$ hi / wi $\leq$ Rilow for every module i with fixed orientation (i$\in$ S1).
- Rilow $\leq$ hi / wi $\leq$ Rilow for every module i with free orientation (i$\in$ S2).
- The modules are enveloped in the floorplan, and the total wirelength is minimized.

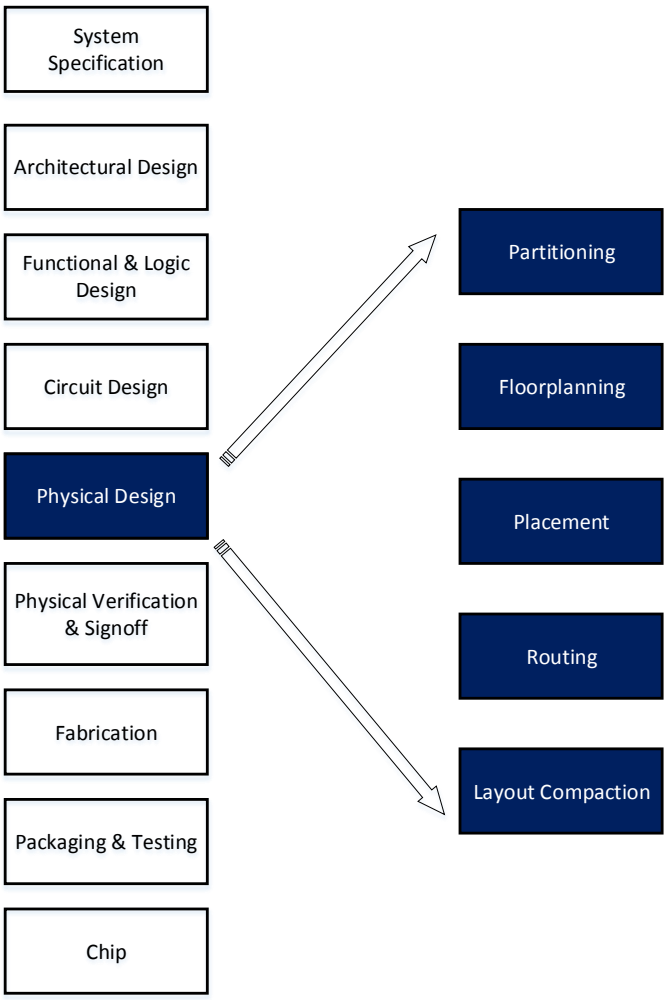

Figure 1 VLSI Design Flow [7]

Sutanthavibul [11] formulated the floorplanning problem as a mixed integer linear program. Chen [2] has solved the floorplanning problem using particle optimization which is a population-based evolutionary algorithm. Young [3] in his paper derived a method to handle all kinds of placement constraints at the same time including preplace constraints, range constraints, boundary constraints alignment constraint, abutment,

clustering etc. Tang and Wong [4] discussed fixed frame floorplanning where the problem is addressed by handling alignment constraint which arises in the bus structure. In [5] it is observed that when two modules are put together, at least one of which has range constraint, the combined supermodule will also have range constraint.

Proposed Design Flow

The task of floorplanning requires some inputs. These inputs are described below.

A. Library Exchange Format (LEF)

While transforming the design into layout form, these files needed as they have the sizes and numbers about the width, requirements of the Macros and Pads. Also they contain the physical data metal layers used.

B. Liberty Files (.lib)

While placing the cells and routing the nets, the objective is to optimize the layout in terms of area, timing and power. For that, some information about the timing and power dissipated by the cells and macros is needed.The information is provided by .lib files.

C. Verilog Netlist

The blocks or modules in the designs are connected with other to get the desired functionality. The interconnection between blocks is described by the netlist. It does not have any information about the parasitic, capacitance, resistance or any other physical information.

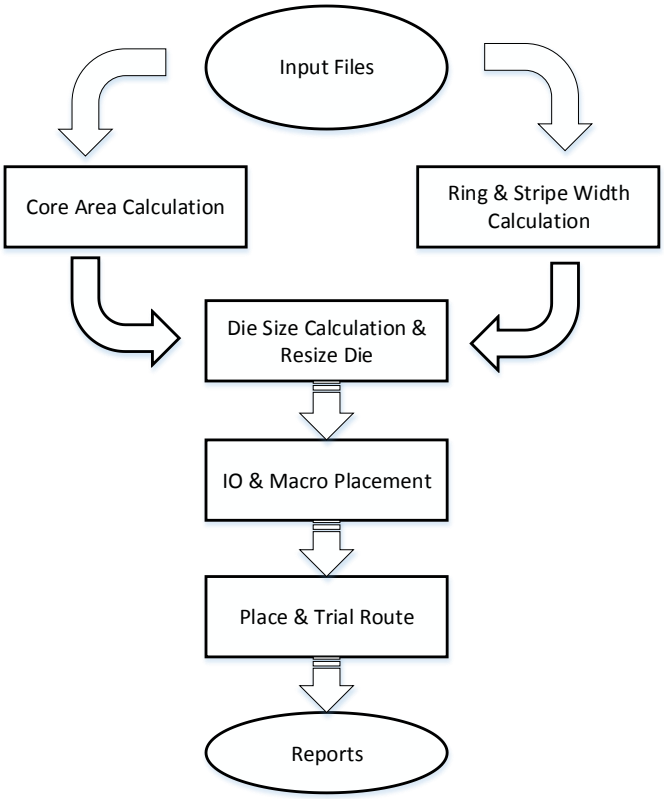

Figure 2 Proposed Design Flow

For effectivefloorplan, a design flow is proposed as shown in Fig. 2.To compute the area of the die, core, the width of the stripes and the rings are obtained by extracting the information from files. The below-given expressions are used to do the computation.

Standard cell area = S

Extrapolation to 0.7 = S_e = S*0.7

Macro area (including Halo) = M

Total Area for core, C= S_e + M

Core Edge (aspect ratio 1:0), $E_c = (C)^{1/2}$

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 6-13 Jan 2019-June 2019 www.csjournals.com

Voltage = E

Number of power pads = 4

Current Rating of Pad = I

Chip current, T= 4 * I

Side current = S_i = T/4

Density of current for Layers = G

Width of the ring = R_w = Si/G

S2S Distance = y

Total Stripes = N = E_e / y

Current in each layers = M_i = T/2

Width W_{st} of stripe required to carry M_i current = M_i / Metal Current Density

Width of stripe = S_w = W_{st} / N

The total size of the die is computed using the achieved area for core and the widths of the ring. After computing the die length and widths, the die is again given the achieved length and widths.

Ring Spacing = z

Core to Ring Distance = D_{cr}

Ring to IO Distance = D_{rio}

IO width = W_{io}

Core to IO distance = D_{cio} = z + D_{cr} + D_{rio} + R_w×2

Die Edge, D = E_c + 2×(D_{cio}) + 2×(W_{io})

When the die size has been modified, the IOs and Macros are put into the core area as per connection among them. The power pads are spread onto the four sides.Then the output results of the chip are taken after the placement and routing.

Simulation

The scripts have been developed using TCL scripting language and the tool used for sourcing the scripts and performing floorplanning, placement, and routing is Cadence Encounter 14.14.The computer system of VEDA IIT, Hyderabad with Intel Core I3-4160T processor of 3.10 GHz, 4 GB RAM with RedHat OS has been used as the host machine.The work has been done on Dual Tone Multi-Frequency (DTMF) chip which has 42838 gates, 5906 standard cells, 4 Hard Macros, 71 IO pads. The layout has been done using six routing layers. The operating frequency is considered to be 150 Mhz, technology used is 180 nm.Fig. 3 shows the initial design when loaded first time in the tool.When the tool driven floorplanning is done, the tool gives a layout which has the macros placed in the core area and IO pads around the periphery. Fig. 4 shows the tool driven floorplanning.Fig. 5 shows the floorplanning done by the proposed script which give better performance in terms of timing and power.Here, in constrained Floorplan, the die has been resized according to the extrapolated standard cell area and the power ring calculations. This Floorplan also contains space around core for the power rings to draw the power from power pads to standard cells and macros. The IO pads have been placed in such a way that on each side there is one pair of VDD and VSS pads.The Fig. 6 and Fig. 7 show the placement done with the tool drive Floorplan and the constrained Floorplan respectively. They contain the white patches which are the Global Cell Overflows.

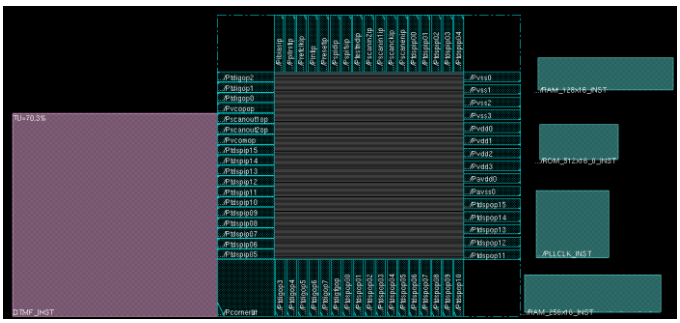

Figure 3 Initial Design

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 6-13 Jan 2019-June 2019 www.csjournals.com

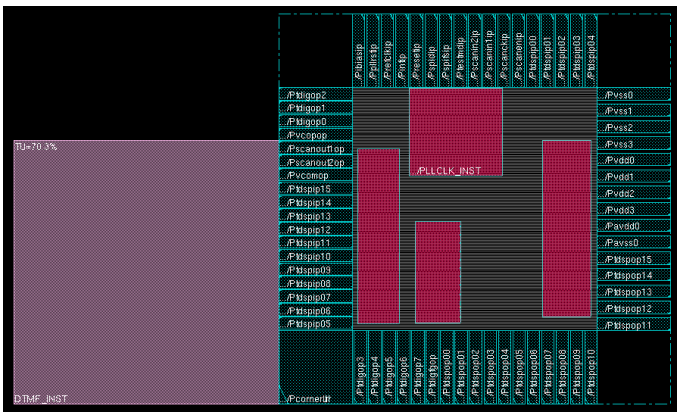

Figure 4 Tool Driven Floorplan

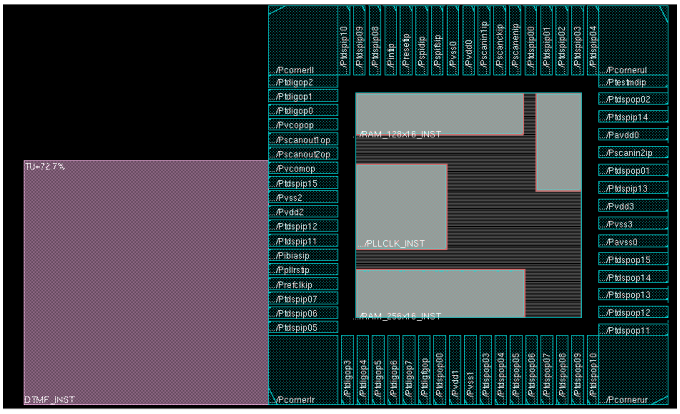

Figure 5 Constrained Floorplan

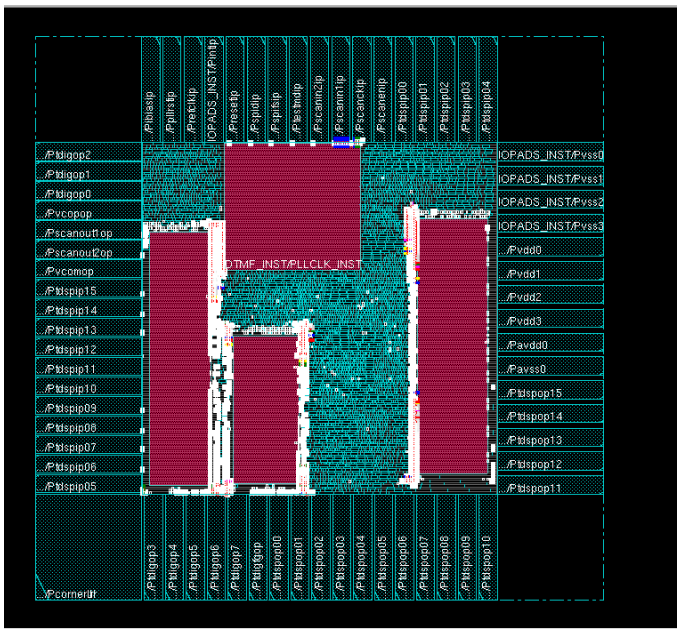

Figure 6 Placement (Tool Driven Floorplan)

A. Global Cell Overflow (GC Overflow)

It is the area where the available routing resources are less than the required routing resources. It is generally around the macro boundaries.From the figures, it can be seen that the GC overflow is comparatively less in the constrained Floorplan than the tool driven Floorplan. The reason behind that are the techniques used for building the Floorplan that are mentioned below.The Fig. 8 and 9 show the trial routing done on the tool driven floorplan and the constrained floorplan respectively.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 6-13 Jan 2019-June 2019 www.csjournals.com

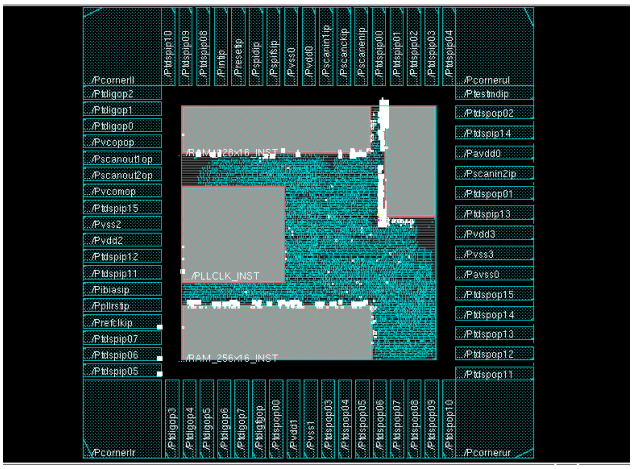

Figure 7 Placement done on Constrained Floorplan

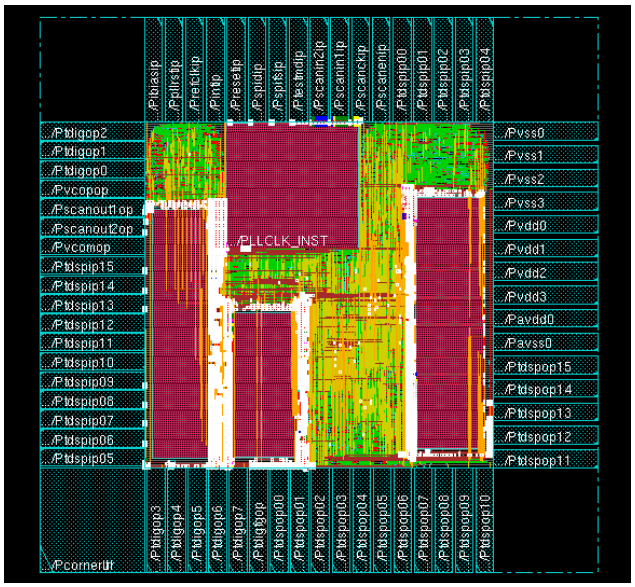

Figure 8 Trial Route (Tool Driven Floorplan)

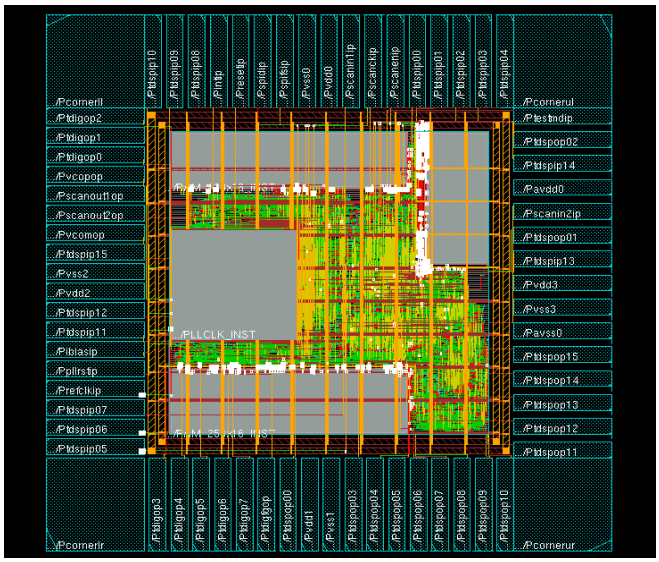

Figure 9 Trial Route (Constrained Floorplan)

Results

After performing the placement and trial routing on both the floorplans, the timing, Design Rule Violation (DRV) and power reports have been generated. They are shown in Table 1, 2 and 3 respectively.

Table 1 Timing report (Tool driven floorplan)

Setup Mode	All	Reg2Reg	Default
WNS (ns)	-1.576	1.576	0.000
TNS (ns)	-2.134	-2.134	0.000
Violating Paths	4	4	0
All Paths	367	367	0

Table 2 DRV report (Tool driven floorplan)

DRVs	Nr Nets Violated	Worst Violation	Nr Total Nets
Max Cap	28	-0.476	32
Max Tran	170	-8.638	170
Max Fan Out	125	-525	127

Table 3 Power report (Tool driven floorplan)

Power Type	Power (mW)	Percentage
Internal Power	22.02499872	52.3455%
Switching Power	20.04949286	47.66504%
Leakage Power	0.00174762	0.0042%
Leakage Power	42.07623906	

From the Tables 1,2 and 3, it is inferred that there are certain design violations in the tool driven floorplan. The Tables 4, 5 and 6 show the reports for the constrained floorplan design that has improved performance in every aspect.

Table 4 Timing report (Constrained floorplan)

Setup Mode	All	Reg2Reg	Default
WNS (ns)	0.098	0.098	0.000
TNS (ns)	0.000	0.000	0.000
Violating Paths	0	0	0
All Paths	367	367	0

Table 5 DRV report (Constrained floorplan)

DRVs	Nr Nets Violated	Worst Violation	Nr Total Nets
Max Cap	0	0	0
Max Tran	0	0	0
Max Fan Out	133	-59	135

Table 6Power Report (Constrained floorplan)

Power Type	Constrained Floorplan	Percentage
Internal Power	21.31113466	58.1436%
Switching Power	15.33971032	41.8517%
Leakage Power	0.00171712	0.0047%
Total Power	36.65256194	100.00%

Table 7 shows the summaries of the comparison of both the approaches. It can be seen that the constrained floorplan gives the layout which is better than the tool driven floorplan in terms of timing, power, and routability of the design.

Table 7 Result Summaries

Parameter	Tool Driven Floorplan	ConstrainedFloorplan	Improvement
WNS (ns)	-1.576	0.098	106.29%
TNS (ns)	-2.134	0	Reduced to 0
ViolatingPaths	4	0	Reduced to 0
Routing Overflow (H)	0.99%	0.04%	95.95%
Routing Overflow (V)	6.12%	6.12%	96.24%
Total Power	42.076 mW	36.65 mW	12.89%

Conclusion

The proposed approach performs the task of floorplanning in such a way thatafter trial routing the total negative slack of the design is reduced to zero and thepower consumption is reduced by 12.89% as compared to thelayoutdeveloped by the tool with default settings. Also, the horizontal routing overflowand vertical routing overflow are reduced by 95.95% and 96.24%, respectively. It implies that compared to the tool driven floorplanning, the constrained floorplanning gives improved layout in terms of performance, power,androutability.

Future Work

After performing the floorplanning, there are GC overflow patches on the core areathat gives the errors of routability. The techniques can be modified to improvethe design further in terms of GC overflow. The constraints are given with the help of TCL scripts that are sourced into thetool. For thebetter user interface, the scripts can be embedded into a Graphic UserInterface using ToolKit(TK).

References

[1]. R. B. Singh, A. S. Baghel, and A. Agarwal, "A review on VLSI floorplanning optimization using metaheuristic algorithms," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 4198-4202.

[2]. Chen Guolong, Guo Wenzhong, and Chen Yuzhong, “A PSO-based intelligent decision algorithm for VLSI floorplanning,” Soft Computing Journal, 2010, pp. 1329-1337.

[3]. Young, Evangeline F. Y. Chu, Chris C. N. and Ho, M. L., “Placement Constraints in Floorplan Design,” IEEE Trans. Very Large Scale Integr. Syst. , July 2004, vol. 12, pp. 735–745.

[4]. Xiaoping Tang, and D. F. Wong, "Floorplanning with alignment and performance constraints," Proceedings 2002 Design Automation Conference (IEEE Cat. No.02CH37324), 2002, pp. 848-853.

[5]. Young, F. Y. and Wong, D. F. and Yang, H. H. , “Slicing Floorplans with Range Constraint,” Trans. Comp.-Aided Des. Integ. Cir. Sys. November 2006, vol. 19, pp. 272–278.

[6]. Hu. TC &Kuh. ES 1985. VLS1 circuit layout: Theory and design’. IEEE Press. New York. USA.

[7]. http://neeasic.blogspot.in/2011/03/vlsi-design-flow.html nee-ASIC Design: All About VLSI Engineering (BackEnd-Physical Design-Place and Route)

[8]. Sherwani. NA. 1999. A1gorithms for VLSI physical design automation. Kluwer Academic Publishers. Boston. USA.

[9]. Wong, DF & Liu, CL 1986, ‘A new algorithm for Floorplan design’. Proceedings of ACM/IEEE Design Automation Conference. pp. 101-107.

[10]. Sait, SM & Youssef, H 1995, ‘VLSI physical design automation: Theory and practice’. IEEE Press, New York. USA. ISBN 978-0-7803-1141-1, pp. 1-19. 1-426.

[11]. Sutanthavibul. S. Shragowitz. E & Rosen. JB 1991. An analytical approach to Floorplan design and optimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. vol.10. no. 6. pp. 761-769.