FAST PARASITIC-AWARE SYNTHESIS METHODOLOGY FOR HIGH PERFORMANCE ANALOG AND RF CIRCUITS

ABDULLAH AL IFTEKHAR AHMED

Fast Parasitic-Aware Synthesis Methodology for High Performance Analog and RF Circuits

by

© Abdullah Al Iftekhar Ahmed

A thesis submitted to the School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Engineering

Department of Electrical and Computer Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May, 2012

St. John's Newfoundland

ABSTRACT

In this work, a fast parasitic-aware synthesis approach of CMOS analog and RF circuits is presented. Traditionally in layout-aware synthesis approaches, the optimization of analog and RF circuits is attained by two separate stages. The circuit sizing stage which is mostly implemented by using some evolutionary algorithms along with certain commercial off-the shelf simulators is followed by layout generation, extraction and verification. This loop continues until convergence is found. In this thesis, a fast parasitic-aware method, which considers the circuit performance constraints and layout induced parasitics simultaneously within a concurrent phase of circuit synthesis by using convex optimization problem, is proposed. The proposed methodology is used to optimize and verify the performance of five high performance analog circuits and two RF circuits in two different CMOS technologies. The synthesis time is found to be under a few seconds and the experimental results demonstrate the high efficacy of this fast parasitic-aware synthesis approach.

ACKNOWLEDGEMENTS

I would like to convey my gratitude to my supervisor Dr. Lihong Zhang for his continuous guidance and financial support to carry on this research. I also want to thank the School of Graduate Studies and the Faculty of Engineering, Memorial University of Newfoundland for the graduate support and the research facilities. I am also thankful to my lab mates for their help, support and insightful discussions. I would like to specially thank my friends who were always beside me in difficult times and helped me to pass through those phases.

Finally, I am grateful to my parents and my sister for their unconditional love and care.

Table of Contents

List of Tables

List of Figures

List of Symbols

g_m = Transconductance of Transistor

β = Transconductance Parameter

μ = Mobility of the Channel of Transistor

C_{ox} = Capacitance per unit area of gate oxide

W = Transistor Width

L = Transistor Length

I_d = Transistor Drain Current

Vt = Threshold voltage

γ = Buld Threshold Parameter

λ = Channel Length Modulation Parameter

k = Boltzman's Constant

T = Temperature

C_{gs} = Gate to Source Intrinsic Capacitance

C_{ds} = Drain to Source Intrinsic Capacitance

C_{db} = Drain to Body Intrinsic Capacitance

g_{ds} = Transistor Channel Conductance

1. Introduction

In order to achieve high speed, low power operation, the complementary metal oxide semiconductor (CMOS) technology is continuously scaled. The scaling also enables designers to obtain a high packing density resulting in more chips per wafer. There is a decrease in the intrinsic device capacitances with scaling which reduces the switching time, resulting in faster circuit operation [1]. The newer technologies are also less error prone to fabrication defect [2]. For a 30% decrease in technology, the reported reduction in gate delay is 30%, increase in device density is 200%, and clock frequency increase is 43%, with a 50% decrease in active power dissipation [3]. But technology scaling is also associated with some drawbacks like impact of parasitics, short channel effect, gate di-electric tunnelling, heat dissipation, wiring and interconnection problem and packaging that may degrade the circuit performance. Especially for technologies newer than 32nm, the parasitic capacitances can drastically affect the circuit performance [4]. Although some intrinsic capacitance does decrease with the scaling of effective channel length and width, the extrinsic capacitance like gate to contact capacitance does not decrease proportionally [1]. Also with technology scaling, interconnects become thinner making the resistance an issue and closely spaced interconnections with an increase in the coupling capacitance introduces crosstalk [5]. All these parasitics can have an impact on delay, current driving capacity and switching time causing degradation to the overall circuit performance.

Motivation

In newer Silicon on Chip solutions, analog circuits occupy a part of the whole system along with the digital components. Even though, usually the portion containing analog circuitry is smaller than the digital one, the design of the analog part works as the bottleneck of the whole system design. The digital domain enjoys the mature hardware description language like VHDL or Verilog which makes it easy to automate the process. But automation of analog circuits is far from perfect. The analog circuit design is a creative and intuitive process and requires a clear understanding of the circuit components, their inter-relation and matching. Because the design is knowledge intensive and complex in nature, it is hard to find a single process that can solve all the analog requirements.

There are several approaches undertaken by the designers to automate the analog circuit synthesis process. Their design process can be broadly divided into several categories like

- Simulated Annealing: that either uses some numerical cost functions or circuit level simulation for verification [6]. The process is statistical and heuristic in nature and obtaining the solution is a tedious process.
- Topology Based Synthesis: that generates topology from basic building blocks and considering the tradeoffs between blocks, optimizes the selection [7].

- Sub-circuit Based Synthesis: that breaks down to smaller circuit components and structural constraints are developed for each sub-circuit to facilitate iterative steps to linearize the constraints [8].

Alongside the circuit sizing phase, several methods have been applied to include the layout induced parasitic effect during circuit synthesis. More details of some of the approaches are elaborated in Chapter 3.

Most of the traditional approaches are based on some sort of evolutionary algorithm and cost function evaluation that demands a high computational cost without any definite guarantee of convergence. The work in this thesis targets to solve this problem by automating the analog circuit synthesis, which can attain a very fast convergence and considers parasitics inside the circuit sizing phase. The details of the proposed methodology are also given in Chapter 3 and the formulation is given in the subsequent chapters.

Organization of the Thesis

The thesis is organized as follows. Chapter 1 gives a brief introduction of the problems of analog circuit synthesis and the importance of adding parasitics into circuit synthesis.

Chapter 2 includes test cases to point out the effects of parasitics on two dynamic comparator structures which are the Lewis-Gray Structure and the Differential Pair Structure and puts emphasis on the importance of considering parasitics during the circuit sizing.

Chapter 3 discusses some existing layout-aware circuit synthesis methodology and their advantages and drawbacks. Then, it discusses the proposed parasitic aware synthesis methodology in detail. The proposed parasitic aware synthesis methodology is compared with the existing methodologies and the advantages are shown.

Chapter 4 contains the detailed design, optimization and verification of three CMOS Operational Amplifier structures to demonstrate the applicability of the proposed methodology on high performance analog circuit.

Chapter 5 continues to apply the methodology by optimizing a gain based CMOS Analog Comparator and another latch based CMOS Dynamic Comparator and verifies the solution in Cadence simulation.

Chapter 6 contains the design of a Low Noise Amplifier (LNA) and a Cross Coupled LC Oscillator showing the effectiveness of the proposed methodology on RF circuits.

Finally, chapter 7 draws conclusions and discusses the future scope of this work.

2. Effect of Parasitics on Analog Circuit: Case Study

In this chapter, the significance of proper transistor sizing and the effect of the presence of parasitics at different nodes of analog circuits are examined, to understand the importance of proper circuit synthesis and inclusion of parasitic components inside the circuit synthesis phase. Two dynamic comparator structures namely Lewis-Gray structure [9] and Differential pair structure [10] were chosen to scrutinize the effect of parasitics on analog circuits because dynamic comparators are expected to be more susceptible to layout mismatch. For each circuit, changes in transistor width, length and aspect ratio are made and their performance is observed. Later, parasitic capacitances are put on different nodes of the two comparator structures that may arise from the layout and the effect is shown.

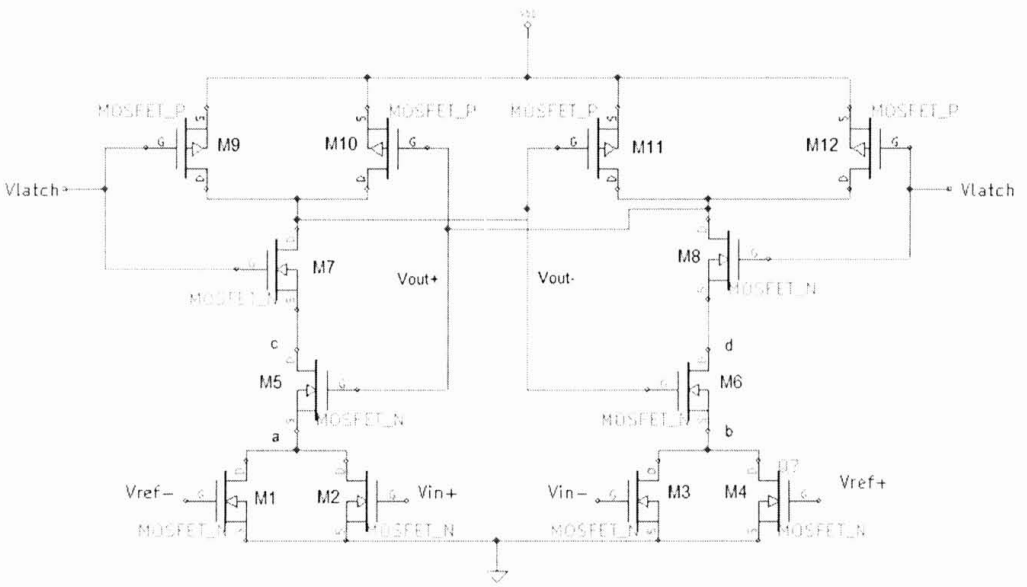


Figure 1: Dynamic Comparator (Lewis Gray Structure) [9].

2.1. Sizing Effect on Lewis Gray Structure

A common structure of dynamic comparator is the Lewis-Gray structure (also known as Resistive Divider structure) shown in Figure 1. The working principle is explained in [11]. There are two modes of operation, when $V_{latch} = 0$, the NMOS, M7 and M8 are in cut off region, whereas the PMOS M9 and M12 are ON. So both the differential output voltage is pre-charged to the supply voltage V_{dd}. When, $V_{latch} = 1$, the pre-charged output voltages causes the NMOS M5 and M6 to stay ON, whereas the NMOS M7 and M8 are turned ON by the high voltage at V_{latch}. The input and reference voltage of M1, M2, M3, M4 decide the resistance of each branch and the final differential output voltages and it stays the same until the comparator is reset by $V_{latch}=0$.

2.1.1. Case I: Testing with known transistor size

At the beginning, the circuit behaviour of Lewis-Gray structure is tested in *TSMC 0.18um* technology with a certain transistor size. The width of the transistors considered are, $(W)_{1,2,3,4} = 1.45um$ and $(W)_{5,6,7,8} = 3.5um$ and the length, L is made equal to $0.45um$. The two reference voltages, V_{ref+} & V_{ref-} are taken as $1.6V$ and $1.2V$ respectively. The clock has a frequency of 10MHz with $1.6V$ /$0V$ (High/Low). The condition of differential input of the circuit to detect properly is given in [11] which is,

$$V_{in}^{+} - V_{in}^{-} = \frac{W_B}{W_A}\left(V_{ref}^{+} - V_{ref}^{-}\right)$$

where $W_A = W_2 = W_4$ and $W_B = W_1 = W_3$. For any inputs that satisfies the equation, the positive output V^+ should go high and the negative output V^- should go low. The differential input, V_{in}^- is varied from 0.2~0.6V and the other input V_{in}^+ is varied from 0.3 to 1.6V. The circuit is simulated and the result is shown in Table I.

The input combinations for which the correct output logic is obtained are shown shaded inside the Table I. The negative input voltages (V_{in}^-) are placed in columns and the positive input voltages (V_{in}^+) are put in rows and the table shows the logic values of the corresponding positive (V^+) and negative output voltages (V^-). From the table, it is observed that for lower values of V_{in}^- like 0.2V, the minimum difference needed between the two inputs to make the circuit work is at least 0.7V whereas this difference should be 0.4V according to the equation (W_B=W_A). The reason for this is the negative input which is below the threshold voltage limits the total current flowing. This condition is satisfied for higher inputs like 0.5V and 0.6V, where the gate to source voltage is large enough to provide sufficient current for circuit operation.

Table I: Lewis Gray Structure: Case I

Vin-	0.2		0.3		0.4		0.5		0.6	
Vin+	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-
0.3	0	1	0	1	0	1	0	1	0	1
0.4	0	1	0	1	0	1	0	1	0	1
0.5	0	1	0	1	0	1	0	1	0	1
0.6	0	1	0	1	0	1	0	1	0	1
0.7	0	1	0	1	0	1	0	1	0	1
0.8	0	1	0	1	0	1	0	1	0	1
0.9	1	0	1	0	1	0	1	0	0	1
1	1	0	1	0	1	0	1	0	1	0
1.1	1	0	1	0	1	0	1	0	1	0
1.2	1	0	1	0	1	0	1	0	1	0
1.3	1	0	1	0	1	0	1	0	1	0
1.4	1	0	1	0	1	0	1	0	1	0
1.5	1	0	1	0	1	0	1	0	1	0
1.6	1	0	1	0	1	0	1	0	1	0

2.1.2. Case II: Reducing the channel length

In this case, the channel length and width of the individual transistors is reduced but the aspect ratio is kept as constant. The chosen length L is reduced from $0.45um$ to $0.35um$, and the widths, $(W)_{1,2,3,4}$ are made $1.15um$. The result is shown in Table II. From the table, it is observed that, for the same $V_{in}^{-} = 0.3V$, the minimum difference increases from 0.6 to $0.7V$. The same is noticed throughout the whole input range. Therefore, for this comparator, reducing the length and keeping the same aspect ratio, results in a need for higher input voltage difference, to make the circuit work.

Table II: Lewis Gray Structure: Case II

Vin-	0.3		0.4		0.5		0.6		0.7	
Vin+	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-
0.3	0	1	0	1	0	1	0	1	0	1
0.4	0	1	0	1	0	1	0	1	0	1
0.5	0	1	0	1	0	1	0	1	0	1
0.6	0	1	0	1	0	1	0	1	0	1
0.7	0	1	0	1	0	1	0	1	0	1
0.8	0	1	0	1	0	1	0	1	0	1
0.9	0	1	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0	0	1
1.1	1	0	1	0	1	0	1	0	1	0
1.2	1	0	1	0	1	0	1	0	1	0
1.3	1	0	1	0	1	0	1	0	1	0
1.4	1	0	1	0	1	0	1	0	1	0
1.5	1	0	1	0	1	0	1	0	1	0
1.6	1	0	1	0	1	0	1	0	1	0

2.1.3. Case III: Increasing aspect ratio

In this case, the aspect ratio is increased by making the initial transistor width, $(W)_{1,2,3,4}$ as $1.5um$, keeping the length, L same as $0.35um$. The result is similar to that in Table II, which does not change the operating input conditions. So, a little increase in the transistor aspect ratio is not enough to improve the circuit operating input condition.

Next, keeping the length L= 0.35um as constant, the device width is increased incrementally, starting with an initial value of 0.5um and a step size 0.25um and ending at 4um. The negative input voltage, V_{in}^- is taken as 0.3V in all cases. The result is shown in Table III up to W=2.75um. From the result, it is seen when W is equal to 2.75um, the minimum input difference needed reduces from 0.7V (For V_{in-} = 0.3V, Table II) to 0.5V. So it can be concluded that for this circuit increasing the transistor aspect ratio increases the current flow, which improves the operating range of the circuit. So proper sizing can result in a better performance of analog circuits.

Table III: Lewis Gray Structure: Case III

W	0.5um		0.75um		1um		1.25um		1.5um		1.75um		2um		2.25um		2.5um		2.75um	
$Vin+$	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-
0.3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0.4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0.5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0.6	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0.7	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0.8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
0.9	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0

2.2. Sizing Effect on Differential Pair Structure

The same performance analysis is done in STM 90nm technology with the Differential Pair structure shown in Figure 2. The operation of the differential-pair structure is explained in detail in chapter 5.

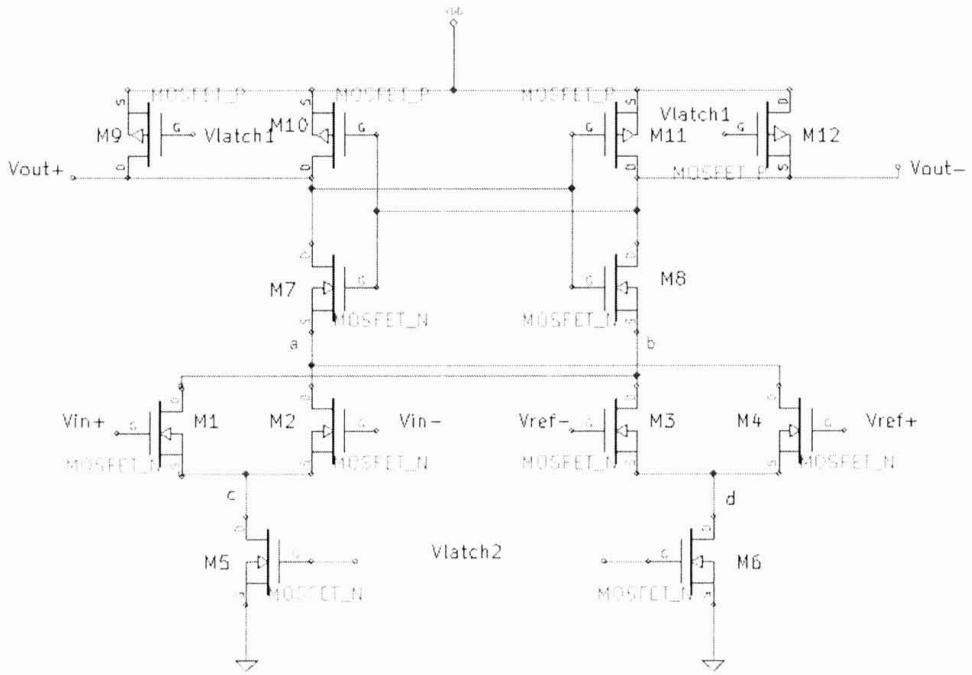


Figure 2: Dynamic Comparator (Differential Pair Structure) [10].

For the Differential pair circuit, the two reference voltages are taken as $V_{ref+} = 0.8V$ and $V_{ref-} = 0.6V$. There are two clock sources namely, $V_{latch1} = 1/0\ V$ (High/Low) with a frequency of $100MHz$ and $V_{latch2} = 0.4/0\ V$ (High/Low), again with a frequency of $100MHz$. The clocks have the same phase. The transistors has the following widths, $(W)_{1,2,3,4} = 0.32um$ and $(W)_{10,11} = 0.32um$, $(W)_{7,8} = 0.76um$, $(W)_{9,12} = 0.32\ um$, and a

minimum size length, $L = 0.1um$. The result is shown in Table IV. It is seen from the result that, for any negative input, $V_{in-} > 0.3V$, the minimum input difference is constant as $0.3V$.

Table IV: Differential Pair Structure

Vin-	0.2		0.3		0.4		0.5		0.6	
Vin+	V+	V-	V+	V-	V+	V-	V+	V-	V+	V-
0.3	0	1	0	1	0	1	0	1	0	1
0.4	0	1	0	1	0	1	0	1	0	1
0.5	0	1	0	1	0	1	0	1	0	1
0.6	0	1	0	1	0	1	0	1	0	1
0.7	1	0	1	0	1	0	0	1	0	1
0.8	1	0	1	0	1	0	1	0	0	1
0.9	1	0	1	0	1	0	1	0	1	0
1	1	0	1	0	1	0	1	0	1	0

Next, the circuit performance is measured by decreasing the length and keeping aspect ratio constant (like in case II) and increasing the aspect ratio (like in case III). But it didn't show any difference in the input operating condition up to $W = 4um$. So, although the differential pair structure needs lower input difference than the Lewis-Gray structure, it showed little variance with transistor sizing.

2.3. Effect of Parasitic Capacitance at Different Nodes:

The effect of parasitic capacitances is tested with both circuit structures. A capacitance of 1fF is put at different nodes of the circuit and the circuit behaviour is examined. The result for Lewis-Gray structure in *TSMC 0.18um* technology is shown in Table V. It can

be seen that, for the added parasitic capacitances of $1fF$ to the substrate, at almost all nodes, the minimum input difference is increased by at least $0.1V$. The output node V and the node at d (Figure 1) seem to be the most susceptible to the parasitic effects which shows a $0.2V$ increase in minimum input difference for $V_{in}^{-} = 0.4V$. On the other hand, a small 1fF capacitance at the positive output node may reduce the required minimum difference between the inputs and actually improves the performance. As can be seen from Table V, for $V_{in}^{-} = 0.6V$, the minimum input difference between $(V_{in}^{+} - V_{in}^{-})$ is $0.4V$ when there is no capacitance added. But if there is a capacitance of 1fF present at the positive output node, it actually decreases the minimum difference needed for the circuit to operate properly. The reason can be understood from the circuit diagram in Figure 1, as a capacitance at the positive output node, makes the pre-charged voltage at that node, harder to discharge than the negative output node. So a smaller input difference is sufficient enough to discharge the negative side and make the comparator decide correctly.

Table V: Parasitic Effect on Minimum Input Difference: Lewis Gray Structure

(The difference between $(V_{in}+) - (V_{in}-)$ is in the table, - means data not available)

Vin-	No Cap	$1fF$ at Node *V+*	$1fF$ at Node *V-*	$1fF$ at Node *c*	$1fF$ at Node *d*
0.4	0.5	0.5	0.6	0.6	0.5
0.5	0.4	0.4	0.5	0.5	0.4
0.6	0.4	0.3	0.5	-	-
0.7	0.3	-	0.4	-	-

The same result for the Differential Pair structure in *STM 90nm* technology is shown in Table VI. It is found that for $1fF$ capacitance at negative output node, V^- and at node b to the substrate, the comparator circuit doesn't work at all.

Table VI: Parasitic Effect on Minimum Input Difference: Differential Pair Structure

(The difference between (V_{in}+) – (V_{in}-) is in the table, - means data not available)

Vin-	No Cap	$1fF$ at Node $V+$	$1fF$ at Node $V-$	$1fF$ at Node a	$1fF$ at Node b
0.3	0.4	-	Doesn't work	-	-
0.4	0.3	-	Doesn't work	0.1	Doesn't work
0.5	0.3	0.1	Doesn't work	0.1	Doesn't work
0.6	0.3	-	Doesn't work	-	-

Doing a parametric sweep reveals that for an input conditions of $V_{in}+ = 0.8V$ and $V_{in}- = 0.4V$, a capacitance of $0.182fF$ at the negative output node to the substrate can throw the circuit out of operation. This capacitance can arise from very little mismatch in the layout and can result in erroneous result at the output.

To see the possibility of layout induced parasitic mismatch at the output nodes, a differential pair circuit is used to do post layout simulations in *TSMC 90nm* for two cases, one with matching floorplan and the other with unsymmetrical floorplan. In both cases, the same transistor sizes were taken. The lengths of transistors were taken as the minimum 0.1um and all the transistor widths were taken as $0.33um$ except for W_7 and W_8 which were taken as $1um$. This is done in *TSMC 90nm* instead of STM 90nm because parasitic extraction in *STM 90nm* is no longer supported by CMC.

At first, a layout is generated keeping the symmetry in transistor placement, intending to reduce the layout induced parasitic mismatch. The layout was extracted for a post layout simulation. The layout is shown in Figure 3 and the corresponding simulation result for V_{in}^{+} = 1.8V and V_{in}^{-} = 0.4V and a clock frequency of 10MHz is shown in Figure 4. Since, V_{in}^{+} − V_{in}^{-} = 1.4V, the positive output voltage (Voutpos) stays high during the evaluating phase of the clock and the negative output (Voutneg) goes to low. This is the expected result as was seen in Table IV.

Figure 3: The symmetric layout generated for the Differential Pair Comparator in TSMC 65nm with symmetric placement.

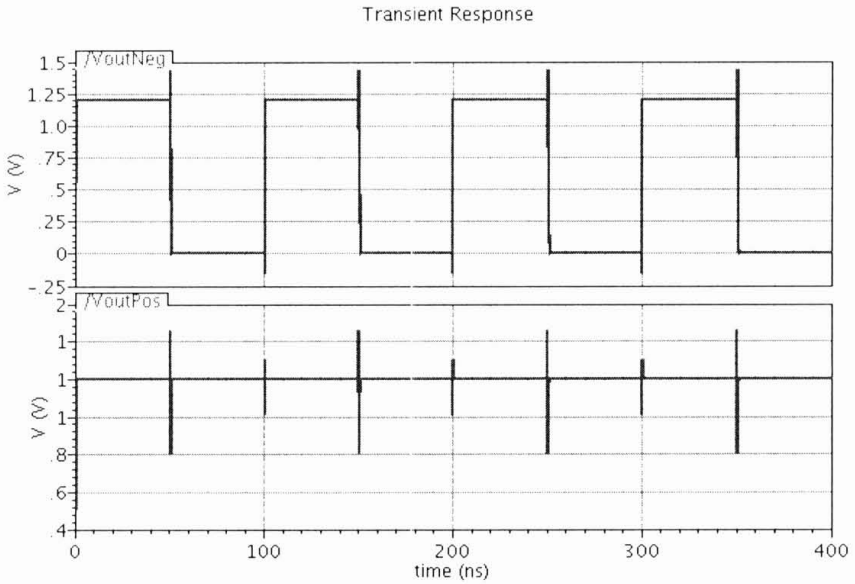


Figure 4: The symmetric Differential Pair Post-layout simulation in *TSMC 90nm* for V_{in}^{+}=0.8*V* and V_{in}^{-}=0.4*V*

From the Calibre PEX extraction, the total capacitance to the substrate at the positive output node is found to be 1.0789*fF* and at the negative output node is 0.8741*fF*. As the capacitance at the negative output node is less than the positive output node, the comparator compares correctly, this is in accordance with the Table VI which showed a small capacitance at the positive output node can actually improve the minimum input difference required to make the circuit.

Now, the same differential pair design is used to generate another layout with the same sizing but unsymmetrical placement. The layout is shown in Figure 5.

Figure 5: Layout generated using Layout-XL from Differential Pair Schematic in CMOS 90nm with unsymmetrical placement.

Since $V_{in+} - V_{in-} = 0.4V$, from the analysis in section II, the $V+$ (VoutPos) should stay high and $V-$ (VoutNeg) should go low at the evaluating phase. But, in Figure 6 that shows transient simulation result, gives the opposite result.

Figure 6: Post layout simulation result for unsymmetrical Differential Pair comparator in CMOS 90nm.

It is understood that the asymmetry in device placement was enough to provide a capacitive mismatch more than $0.182fF$ to throw the circuit out of operation for this input condition. From the extraction result obtained from Calibre PEX, the total capacitance to substrate at the positive output node is $0.79639fF$ and the total capacitance at the negative output node is $1.1985fF$. So the difference between the two capacitances is $0.4022fF$ which is indeed more than the capacitance ($0.182fF$), that was found to be able to throw the circuit out of operation, as observed in the post-layout transient simulation result. This confirms the sensitivity analysis performed on the schematic that a small capacitance at the negative output node can cause the circuit give erroneous result.

So it can be concluded that, not only a proper sizing of the transistors is important for obtaining acceptable performance of an analog circuit, the layout induced effects that can arise from any possible mismatch also needs to be considered for proper circuit operation.

2.4. Summary

In this chapter the effect of circuit synthesis and layout parasitics with two dynamic comparator circuits in two different technologies is investigated. The simulation obtained shows improvement in circuit operating ranges after circuit sizing. The parasitic capacitance effect showed more obvious effect, as the Differential Pair structure became non-operational for a small added parasitic capacitance. Because in newer CMOS

technologies, the contribution of the parasitic capacitance is more prominent due to the fringe and coupling capacitances, it is very important to take into account the parasitic effects during the phase of circuit synthesis. A fast and accurate parasitic aware synthesis method can counteract to the parasitic effects seen in high performance analog circuits.

In the following chapter, the traditional parasitic-aware synthesis along with some prevailing parasitic-aware synthesis approaches for circuit synthesis is reviewed. Then the proposed fast parasitic aware synthesis methodology in this thesis is presented with detail.

3. Parasitic-Aware Synthesis

3.1. Traditional Approaches

In a traditional layout-oriented synthesis process, the circuit sizing and optimization is done in two independent steps embedded inside a loop [12]. The loop continues until the desired performance is found. The circuit is first sized for given specification (front-end). The layout is obtained from the sized circuit (back-end). Then the parasitics are extracted from the layout and a netlist is created. Using that netlist the circuit performance is evaluated and the result is used to determine whether to terminate the process or to continue the loop. The process is given at the following flow chart in Figure 7.

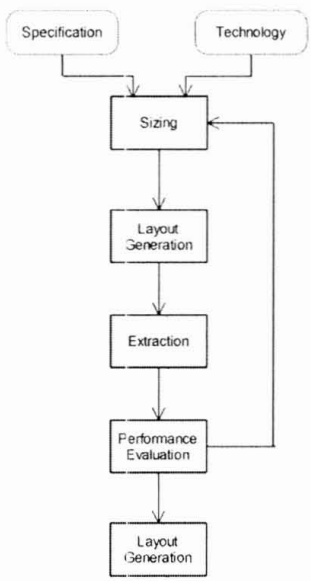


Figure 7: Traditional Approach for Layout Aware Synthesis [12].

In their proposed design, the authors at [12] proposed a knowledge based method using a tool called COMDIAC, which uses equations already defined from detailed knowledge of

the circuit. It uses the layout information during circuit sizing phase. The flow is presented in Figure 8. The sizing is based on trial and error and there is no demarcated terminating factor which determines how long it might take to converge. At each step, the layout tool is called even for multiple times to give the parasitic estimation based on the different constraints given to the layout tool. The circuit sizing responds to the parasitics by changing the transistor size and the process keeps running until there is no change in parasitics. So the transistor sizing and parasitic estimation still stays as a self-governing step. There is no guarantee that the loop will terminate because for even small change in sizing, the parasitics can deviate in newer CMOS processes. This is because as the technology scales, interconnects are more closely spaced and fringe capacitance contribution which is not linear to predict, becomes more significant as the size gets smaller.

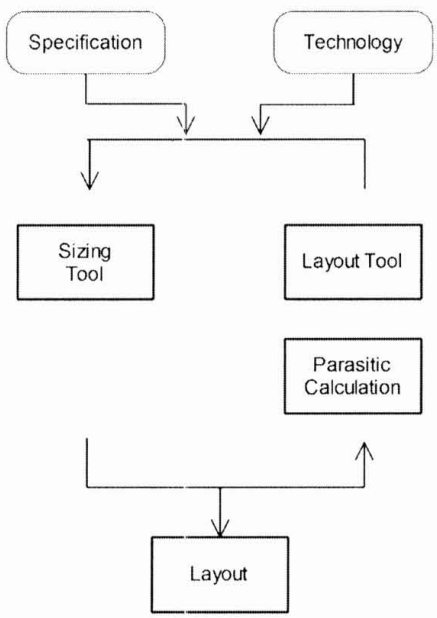


Figure 8: Layout-oriented synthesis proposed in [12].

In [13], another layout aware circuit synthesis process is proposed. For the circuit sizing part a genetic algorithm namely Differential Evolution is employed using a pre-formulated cost function that is generated using a circuit level simulator. Once the cost functions are found, they are used in numerical simulation on the right side instead of using device level simulation. Afterwards the layout generation uses a module generator for basic devices to generate the layout. It is only different to the traditional approach by using the generated cost functions to evaluate circuit performance instead of device level simulation and using module generator instead of layout generation and extraction. There is no guarantee that the system will find the global solution. The authors recommend using this to get an approximate result and then rectifying it by simulation. How the simulation will fine tune the circuit sizing is unknown as it also will need accurate layout information. Their flow is given in Figure 9.

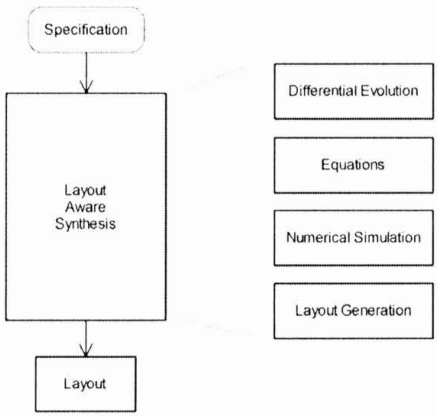


Figure 9: Parasitic Aware Synthesis Approach proposed in [13].

The authors in [14] show the importance of including layout information in circuit sizing by comparing the deviation in performance with and without parasitic consideration. It

removes the layout in the loop and presents a macro model for parasitic capacitance estimation. It divides the circuit into different modules and devises models for inter and intra module capacitance. It uses a procedural layout generator (PLG) called MSL to generate layout and off-the-shelf extractor is used to extract the layout parasitics. As the module size may change during the iteration, it creates a lookup table called Module Characterization Table (MCT) to estimate the values of parasitics during circuit sizing phase. It uses simulated annealing to size the circuit. The flow is shown in the following Figure 10.

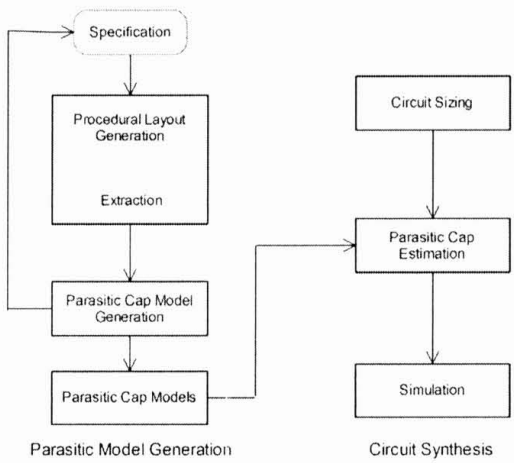


Figure 10: Fast parasitic estimation modeling proposed in [14].

There are several drawbacks of linear interpolation of the parasitics value. The change in capacitance in smaller technology, like fringe capacitance and coupling capacitance might not be linear. And it uses HSPICE with simulated annealing which gets global optimization but at the cost of CPU time. Although the time needed for parasitic extraction using Procedural Layout Generator is mentioned to be ~15minutes, it doesn't

mention the whole optimization time. Not to mention, the time needed to generate the Model Characterization Table for higher number of modules will increase linearly.

The same authors propose a slightly different approach in [15]. Instead of using simulated annealing as a sizing engine, it uses a symbolic performance models (SPM) that are generated using equations from small signal models. The layout information is obtained from MSL as before and an off-the shelf extractor is used to extract the parasitic values. The SPM can take all capacitances in the small signal model or take the few contributing parasitics that are identified from previous sample layouts.

Since it still uses an off-the shelf extractor and the time needed for SPM and complexity are unknown. The layout information is pre-calculated and not dynamic. It is valid for a fixed template layout and the SPM has the limitation of not handling very large circuits.

In the proposed method in [16] shown in Figure 11, at the beginning, the simulated annealing is used to size the circuit by simulating the netlist with presence of parasitics, which is obtained from Model Characterization Table generated by the Procedural Layout Generator discussed before. When it converges, it finds a feasible region and determines the worst case parasitics using the feasible region by a placer. Then it makes local changes to cope with the interconnect capacitances for the worst case scenario.

The methodology still suffers from the total cost of simulated annealing, off-the shelf extractor and the effect of making local changes which might result in a variation of capacitances with respect to those that was obtained at the first sizing step.

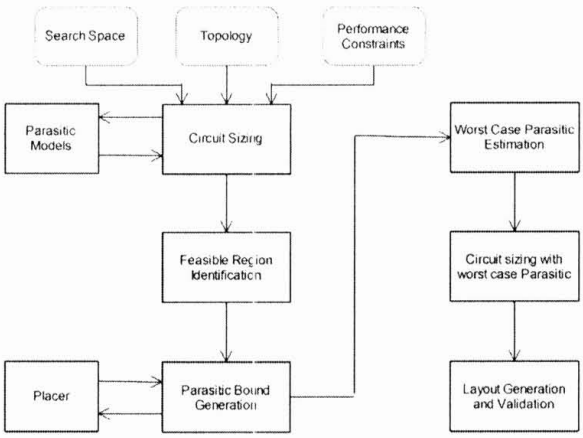


Figure 11: The proposed method in [16].

Another method proposed in [17] is close to the traditional approaches, except instead of taking single layout information; it considers the parasitics of the previous runs to resize the circuit. In each loop, the circuit is resized, placed, routed and extracted. An off-the shelf simulator and extractor is used to evaluate and generate the parasitic value. At each step, a best and worst case capacitance corner is generated to make it robust. The flow is shown in Figure 12.

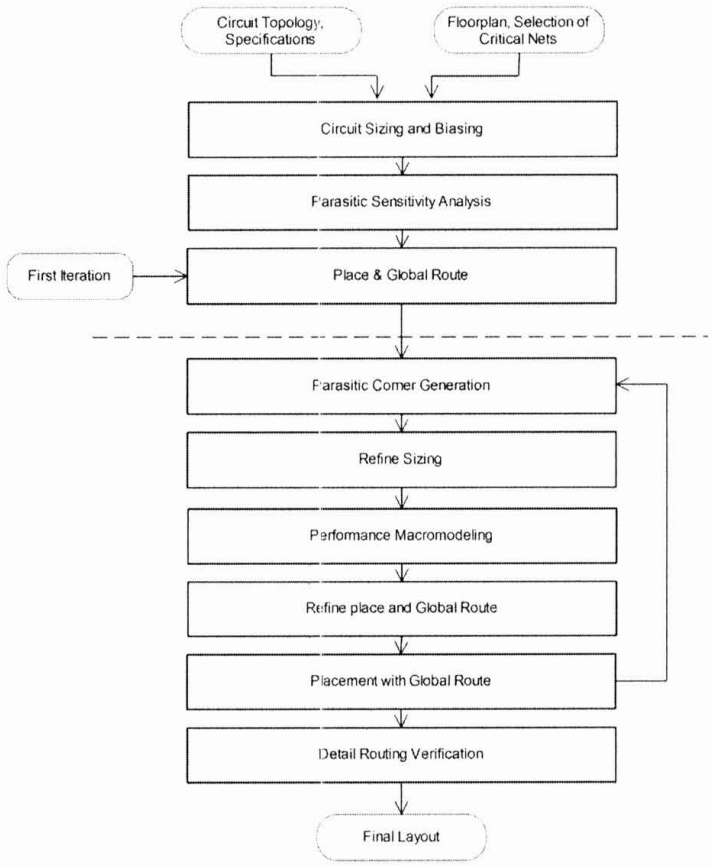


Figure 12: Proposed Synthesis Flow in [17].

The disadvantage is running the routing and floor-planning and extraction inside each loop as well as evaluating with a commercial simulator might take a long time to converge.

The method proposed in [18] includes circuit sizing by simulation. At each step a floorplan is generated and parasitics are estimated using the floorplan and transistor size. After convergence several floorplans are considered for performance and after the

floorplan is selected, layout is generated, extracted and verified. If not met, then the loop executed again. It again uses the simulation based method that is CPU costly and cannot guarantee the convergence.

Another method is proposed in [19]. In this method, a transistor level simulator (HSPICE) is used with simulated annealing technique for the first phase of sizing. In the second phase a deterministic method is used. Template based layout generation is used which takes a few seconds to generate the layout. They use Cadence PCELL and SKILL programming language. At first the sizing engine selects a set of random values inside a range. For those values the Geometric Constraint module selects from a set of layout styles and parasitics is extracted from the selected layout. Then the performance is evaluated. If not met, the loop is executed again.

It uses SA which is slow, use layout generation and extraction at each step which is also slow, and there is no definite guarantee of convergence.

The authors in [20] proposed a method that uses nonlinear optimization algorithm. It uses numerical simulation along with consideration of multiple layouts, placement and routing in each iteration. The parasitic contribution is obtained by integral field solver. The CPU time is reported to be 8 times that of a traditional circuit sizing.

Most of the parasitic aware synthesis used for RF synthesis mentioned above like [13], [16], [17] etc. uses a form of evolutionary algorithm and simulation based sizing method. The authors in [21] used particle swarm optimization and adaptive simulated annealing for parasitic aware RF circuit design. The process uses commercial simulator like HPSICE or SPECTRE along with curve fitting tool in MATLAB to provide parasitic aware synthesis. The detail of parasitic modeling is unclear. Although, the process looks to be two times faster than the conventional simulated annealing technique, the two examples provided shows the number of iteration to be more than 1000. Curve fitting the parasitics for modeling and using a commercial simulator with a high number of iterations can prove to be CPU costly. Other method like [22] uses optimization with recourse including ellipsoidal uncertainty to design a low noise amplifier in Si-Ge technology along with a ring oscillator. It does not elaborate the how to include the parasitics into synthesis and concerns with the circuit sizing only.

From the discussion about the prevailing parasitic aware sizing method above, most of the circuit synthesis methodologies can be divided into two steps. In the first step, an evolutionary algorithm like simulated annealing is used to size the circuit using either a commercial simulator or numerical cost function. This is followed by the second step, that is, from the obtained size; the parasitics are estimated either by actual layout generation and extraction using a commercial tool or by parasitic estimation and extrapolation. Then the circuit performance is evaluated considering those parasitics and the loop continues until convergence is found.

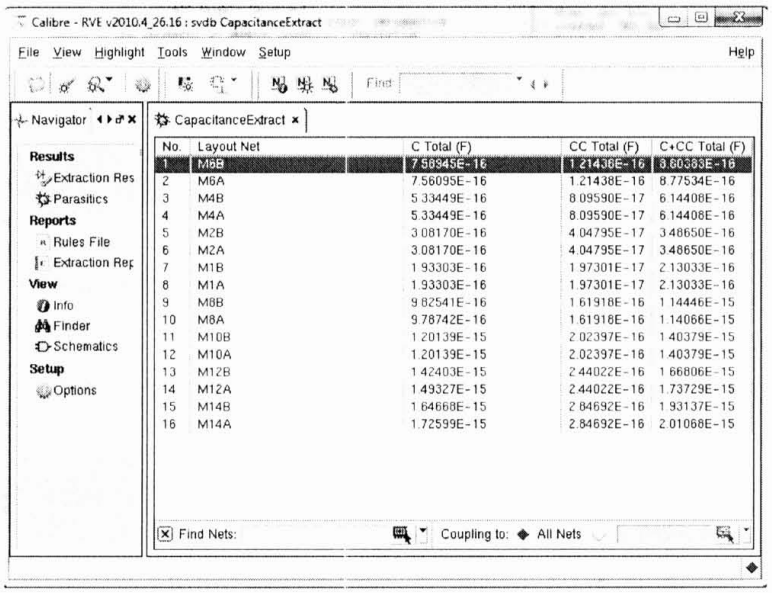


Figure 14: Calibre PEX extracted result for the Metal lines drawn.

The extracted capacitances in terms of the metal length of the parallel metal 1 lines are plotted in Figure 15. From Figure 15, it can be seen for *TSMC 0.18um* technology, the increase in capacitance can be characterized as linear after the metal length becomes 5um or more. This used to be the case in older CMOS technology and the parasitic aware synthesis which used interpolation to estimate capacitance was logical. But as can be seen for the new *TSMC 90nm* technology, with increasing metal lines, the parasitic capacitance increase is not exactly linear. As an example, for parallel metal lines of 1um, the total capacitance (C+CC Total (F) in Figure 14) is 2.1e-16 but for 2um, this value is 3.48e-16 which is only around 1.5 times of the previous, although the length is taken as twice. The reason for this is, as the size of the interconnect gets smaller, the fringe capacitance whose effect is not linear contributes more significantly compared to large interconnect

size. Even when the length is more than 5um, the value of the capacitances deviate from the one obtained from extrapolation of the previous two points.

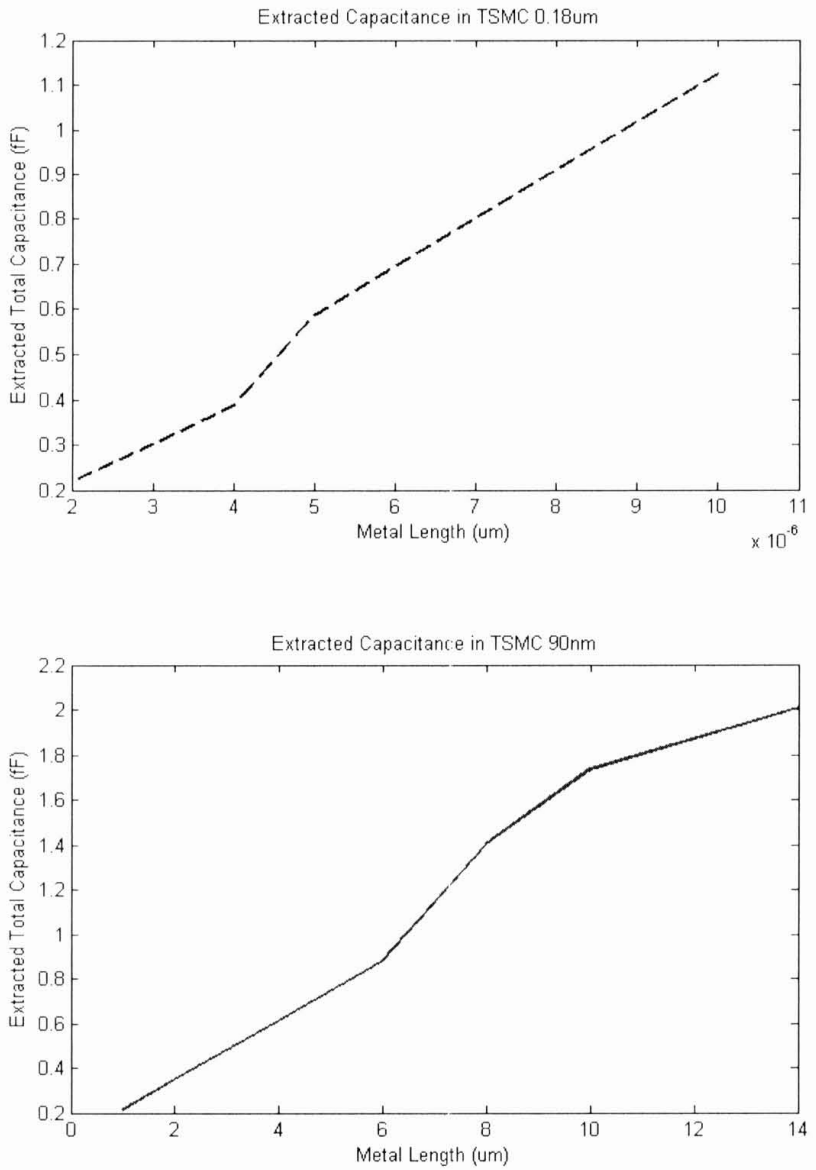


Figure 15: Extracted Total Capacitance plotted in terms of Metal length.

Therefore if the circuit sizing and layout generation and extraction is considered in separate steps, and the layout information of the previous run is used to estimate the parasitics inside the next sizing stage, the estimation may not give an accurate result. This can also be said for using any look up table with interpolation or extrapolation. So it is important to consider the sizing and parasitic estimation simultaneously.

Therefore, it gives the motivation to develop a design methodology that can conduct the circuit sizing and parasitic estimation simultaneously. Instead of using a sizing engine to find a set of transistor sizes followed by parasitic extraction, the proposed method models this as a convex optimization problem to concurrently perform transistor sizing taking parasitics into account. Figure 16 shows the proposed circuit synthesis flow.

A convex optimization problem called Geometric Programming (GP) [23] is utilized, which can include a set of performance constraints formulated from a given technology parameters and required performance specifications, as well as a set of symbolic interconnect parasitics formulated with geometrical requirements and floor-planning. The symbolic floor-planning and routing constraints enables to use a set of parasitic expressions for interconnect parasitics to be enclosed inside the circuit sizing phase. Finally, a GP solver [24] is deployed to provide a solution which not only attains the desired circuit performance but also concurrently considers layout induced parasitic effects. Such GP solvers, using standard interior point algorithm [24] and being able to solve large convex optimization problems very quickly and efficiently, can be readily accessed in the public domain.

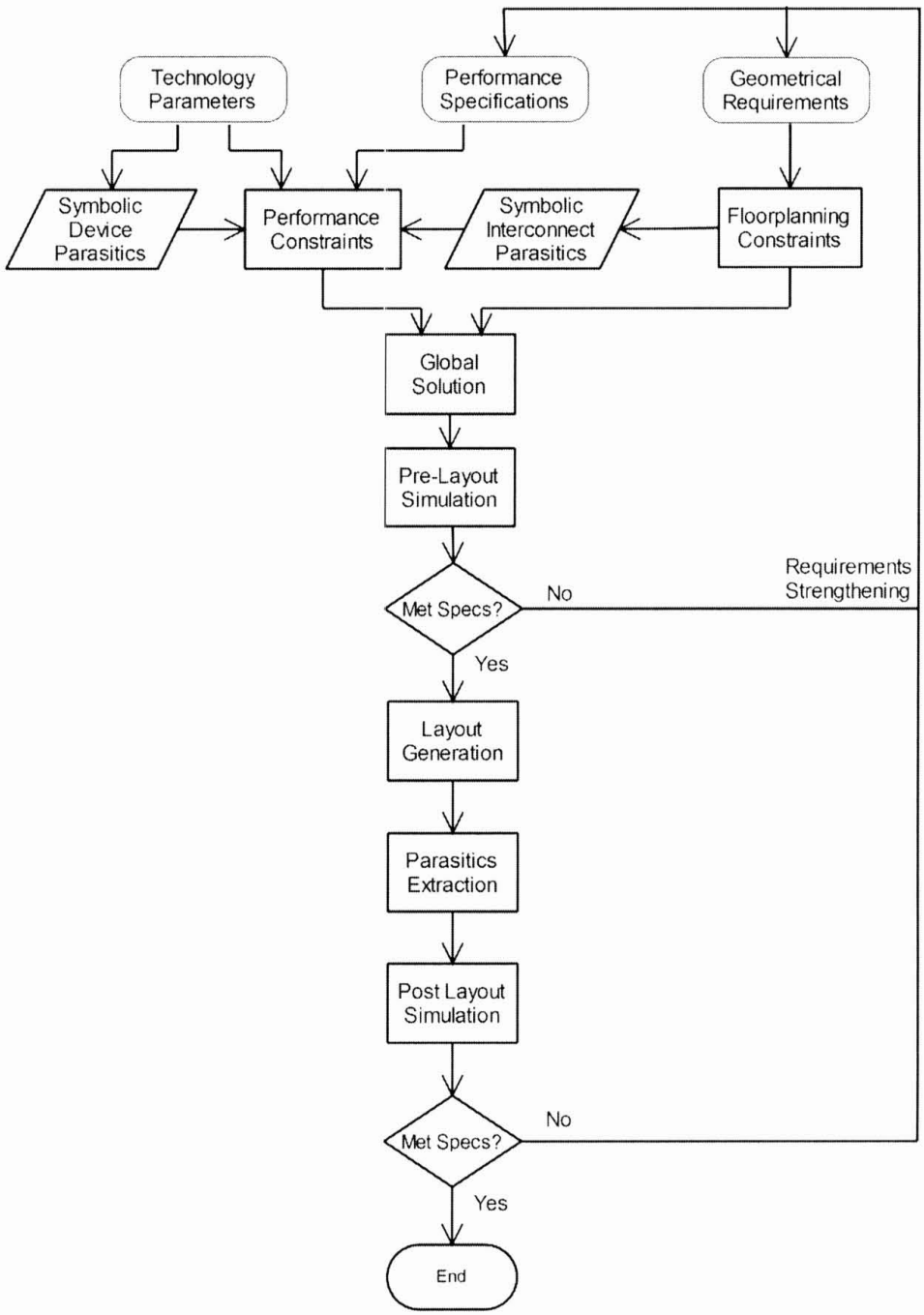


Figure 16: Proposed parasitic-aware circuit synthesis methodology.

After a successful run of GP (Geometric Programming) solver, the obtained sized circuit is tested in Cadence for a pre-layout simulation. If the simulation result meets the requirement, then a layout is created using Cadence XL. From the generated layout, Calibre parasitic extraction tool PEX is used to extract the post layout netlist. This netlist is used to verify the post layout simulation. If the specifications are not satisfied, then the requirements in GP formulation are changed accordingly and again the verification process is carried out.

The proposed parasitic-aware synthesis method has some notable advantages compared to other methods mentioned above.

- It uses very fast and efficient GP solver to do the synthesis which finishes within less than a few seconds. It is much faster than any computationally intensive simulators and evolutionary algorithms. In all the design examples, the GP solve was finished under 2 seconds.
- It does not require any commercial layout generator or extractor inside the optimization phase, so it does not rely on their efficiency. It calculates the layout parasitics simultaneously contributing to the sizing constraints.
- Using accurate device parasitics model for a particular CMOS technology, it gives an accurate result extremely quickly compared to any actual layout generator or procedural layout generators.

- This approach does not require any initial conditions to converge because GP always outputs a global optimum solution, if feasible, irrespective of the starting point [24].

The important blocks of the proposed parasitic-aware synthesis methodology are discussed below.

3.2.1. Circuit Sizing

Even though a form of evolutionary algorithms is the method of choice for sizing in most of the layout-inclusive synthesis stated above, in practice there is no assurance of the time of convergence to provide a global solution. Furthermore, since finding the empirical parameter values or lower bounds could be tricky [21], the termination condition and time are unknown. There are other circuit sizing methods like [25], which breaks the circuit into structural sub-blocks and structural constraints are developed for each sub-block. It is difficult to simultaneously determine the layout induced parasitics dynamically and add to the circuit sizing. Instead, they have to rely on the parasitic values of the previous run or use some interpolation from a pre-generated look up table.

That is the reason why it is proposed to apply geometric programming [24], which can take both performance constraints and floor-planning constraints simultaneously to determine an optimum global solution. It can also avoid intensive usage of computationally expensive commercial simulators or layout generators in order to provide a quick convergence. For any optimization problem $f_0(x)$, GP is formulated as

$$\text{minimize } f_0(x)$$

$$\text{subject to } f_i(x) \leq 1, i = 1, \ldots\ldots, p$$

$$g_i(x) = 1, i = 1, \ldots\ldots, m$$

where f_i are a set of constraints modeled in posynomial form and g_i are a set of optimization constraints modeled in monomial form with x being the GP variables. The posynomial and monomial forms can be put as follows:

$$f_i(x) = ax^b + cx^d + \ldots \leq 1 \qquad \text{(posynomial)}$$

$$g_i(x) = ax^b . cx^d \ldots == 1 \qquad \text{(monomial)}$$

For the targeted circuit, the performance constraints like symmetrical matching, device size, biasing conditions, open loop gain, unity gain bandwidth, phase margin, etc. can be modeled either in a monomial or posynomial form. For common analog circuits, these expressions are available in literature. For more complicated analog circuits, the circuit can be broken into several smaller blocks for which expressions are available and subsequently the performance can be modeled either in posynomial or monomial form. In addition, floor-plan can be constructed using transistor size, minimum allowable distance between transistors and matching of different components. All these constraints are represented in the GP equations/inequalities that are delivered to the GP solver for an optimum solution.

3.2.2. Inclusion of Layout Effects

An important part of the proposed synthesis approach is the inclusion of layout effects to be considered alongside the sizing constraints. To incorporate the accurate calculation of layout-induced capacitance and resistance, the circuit is floor-planned, routed and each device capacitance and interconnect capacitance and resistances are modeled to formulate a set of symbolic layout constraints.

Floorplanning:

To mitigate the layout-induced mismatch, the symmetry constraints are put inside the floorplanning constraints. Minimum allowable distance in the technology between two devices is used.

For example, for the circuit for Two Stage P input channel Operational Amplifier in Figure 17, with the presence of the parasitic capacitances (C_{int1} - C_{int4}) and interconnect resistances (R_1, R_2), the floorplan is shown in Figure 18. The transistors M1 & M2 and M3 & M4 are placed symmetrically, which is visible from the floorplan.

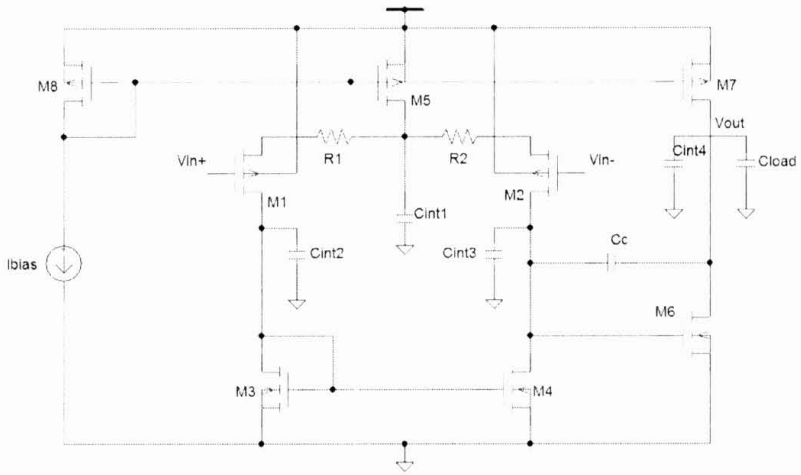


Figure 17: Two Stage P-input channel Op Amp.

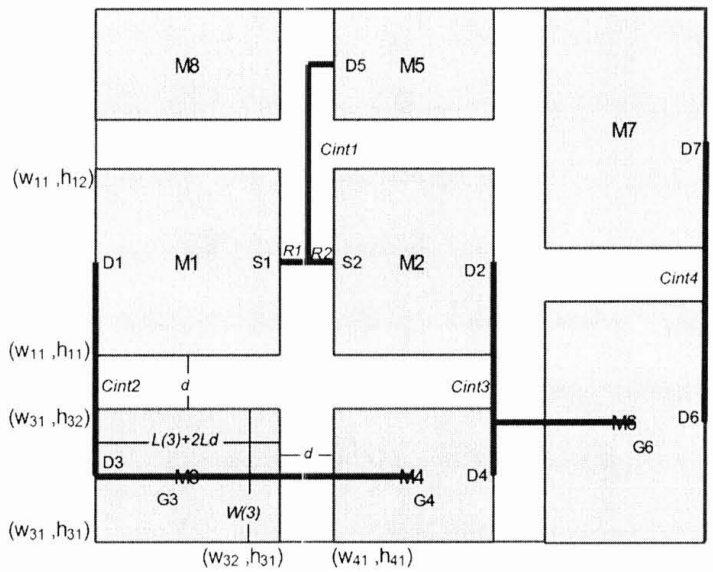


Figure 18: Floorplan of the circuit in Figure 17

Floorplanning constraints are formulated such that the total area required is minimized. Cartesian coordinates are used to denote the positions of the devices. For any transistor,

$$W(i) \leq h_{i2} - h_{i1} \tag{1}$$

$$d \leq h_{i1} - h_{j2}$$

$$L(i) + 2L_d \leq w_{i2} - w_{i1}$$

$$d \leq w_{i1} - w_{j2}$$

where $W(i)$ is channel width, $L(i)$ is channel length, h_{i2}, h_{i1} and w_{i2}, w_{i1} are the height and width coordinates of the i_{th} transistor, L_d is the lateral diffusion of source and drain regions, and d is the minimum allowable distance between two adjacent transistors.

Device Parasitics Modeling:

For any CMOS technology, an accurate capacitance and resistance model for all sub-circuit device capacitances, like C_{ds}, C_{gs}, and C_{db}, are available from the foundry. These models provide accurate device capacitances in terms of transistor width, length and number of fingers. Because the intrinsic capacitances are closely related to individual device sizes, any pre-calculated values like in [14] for estimating these capacitances might give an erroneous result. Using these capacitance models, accurate intrinsic device capacitance constraints are formulated in symbolic form and passed to GP.

Routing and Interconnect Parasitics modeling:

Using the floor-planning constraints, the shortest path for interconnecting the transistors is calculated. The Manhattan distance between two nodes is used to obtain the length of interconnects (shown as thick lines in Figure 18) in the symbolic form. Using the unit capacitances from metal layer to substrate and metal layer to active regions available in the technology parameters, the expression for interconnect capacitance is obtained in terms of Cartesian coordinates. These expressions are represented inside the device performance constraints and utilized to size the circuit.

Once all the performance and parasitics models are formulated the GP solver is employed to solve the optimization problem considering all performance and floor-planning constraints simultaneously. The solution obtained from the GP solver is then verified by pre-layout simulation. If the pre-layout simulation does not meet the requirement, the required performance parameters in GP are made stringent to seek for a better optimization. If the pre-layout simulation is successful, then a layout is generated using layout-XL in Cadence and the generated layout is extracted and a post-layout simulation is performed.

In case of all the circuits optimized in this methodology, the number of iteration between the GP solution and pre-layout simulation is found to be very low. And once the pre-layout simulation succeeded, all the post-layout simulation succeeded in the first run. So

the convergence of this parasitic aware methodology is very fast as the GP solution is obtained under a few seconds and the required number of simulations using a commercial simulator for pre and post layout verification, is significantly lower than any evolutionary based approach. Moreover, it does not need to generate the layout and extract it each time inside the loop, which can turn out to be extremely slow, when used with the evolutionary algorithms and a commercial extractor, as done in the other traditional approaches.

3.3. Summary

In this chapter, several contemporary layout aware analog circuit synthesis methodologies are reviewed and their features and comparison is made. Then the proposed fast parasitic-aware synthesis method is described in detail and the advantages of this method in comparison to the existing methods are listed.

In the next chapter, three Operational Amplifier circuits are designed and optimized following the above methodology and the obtained solution is verified in post layout simulation.

the convergence of this parasitic aware methodology is very fast as the GP solution is obtained under a few seconds and the required number of simulations using a commercial simulator for pre and post layout verification, is significantly lower than any evolutionary based approach. Moreover, it does not need to generate the layout and extract it each time inside the loop, which can turn out to be extremely slow, when used with the evolutionary algorithms and a commercial extractor, as done in the other traditional approaches.

3.3. Summary

In this chapter, several contemporary layout aware analog circuit synthesis methodologies are reviewed and their features and comparison is made. Then the proposed fast parasitic-aware synthesis method is described in detail and the advantages of this method in comparison to the existing methods are listed.

In the next chapter, three Operational Amplifier circuits are designed and optimized following the above methodology and the obtained solution is verified in post layout simulation.

4. CMOS Operational Amplifier Design

4.1. Two Stage N-channel Input Differential Amplifier

A widely used two stage N-channel input differential amplifier is made of a single-ended differential amplifier stage followed by a common source stage which is showed by Figure 19. This amplifier doesn't suffer from reduced voltage swing like a cascoded amplifier [26] but the unity gain frequency is not as good as the cascode amplifier. It also provides a high voltage gain with a relatively small number of transistors. In the differential stage, M1 and M2 work to take two differential inputs. The current through M3 is the same as M1 and M4 has the same mirror current. M5 works as the current sink of the two branches. In the second stage which is the common source amplifier, M7 works as the driver transistor and M6 is the current source load [26]. The interconnect capacitances C_{int1}, C_{int2}, C_{int3}, C_{int4} from different nets and two resistive components in the two sensitive branches *R1* and *R2* are also shown in the figure that were considered inside the parasitic aware synthesis.

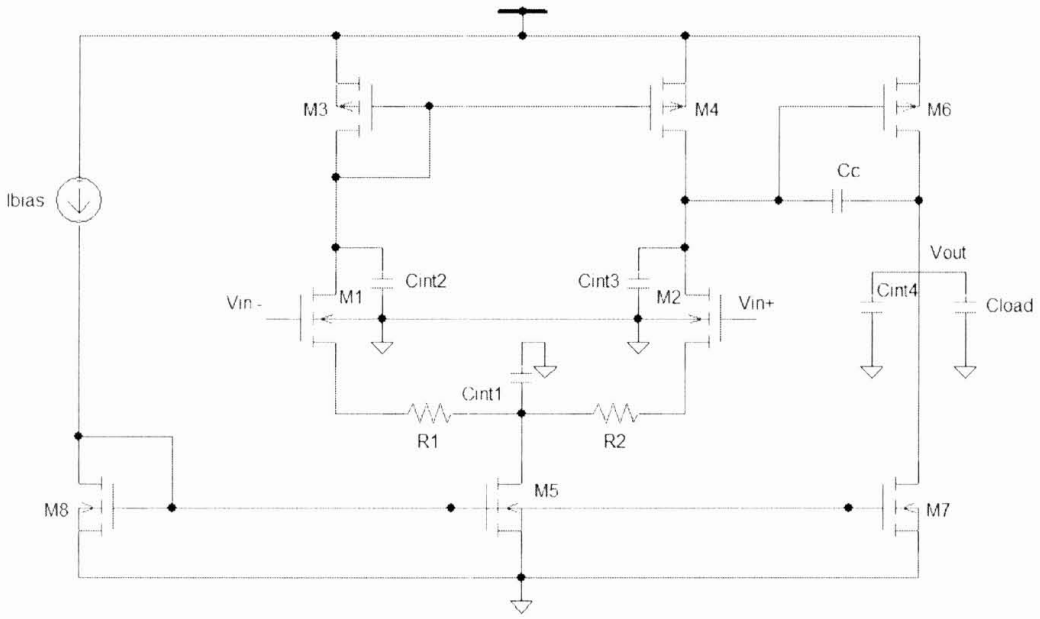


Figure 19: Two Stage N-input Operational Amplifier [27].

The design objectives are

minimize *Area*

while subject to

gain $\geq$ 65dB

Unity Gain Frequency $\geq$ 5MHz

Phase Margin $\geq 60^0$

CMRR $\geq$ 80 dB

Power $\leq$ 1mW

When optimizing the circuit, all the required specification of minimum gain, unity gain bandwidth, phase margin, input common mode range and slew rate was put as target constraint inside the geometric programming formulation. Also, for all the transistors, the device parasitic capacitances like C_{db},C_{gs}, C_{gd} etc are added to the constraint equations. A floor plan with minimum allowable distance between the transistors is created and the interconnect capacitances are calculated using their unit values and from the selected Cartesian coordinates of different transistors. This is also put together with device capacitances to take into account the parasitic effect. Since all these, parasitic capacitances are modeled in terms of transistor sizes and interconnect length, instead of a fixed constant value, when GP optimizes the design, all the parasitic effects are taken care of alongside the circuit sizing simultaneously. It is an obvious difference of the proposed approach from the traditional layout aware synthesis, where circuit sizing step is first completed and then a layout generation and extraction follows to validate the design. All the modeling details are shown here:

4.1.1. Gain Constraint

The gain of the two-stage amplifier is the product of the gains of individual stages. If we have M1 & M2 and M3 & M4 perfectly matched, the gain of the first differential stage is given as [27]:

$$A_{v1} = \frac{-g_{m1}}{g_{ds2}+g_{ds4}} = \frac{-2g_{m1}}{(\lambda_2+\lambda_4)\, I_5}$$

Where g_{m1} & g_{m2} are the transconductance of M1 & M2 which is found by,

$$g_m = \sqrt{2\mu C_{ox}\frac{W}{L}I_D}$$

and g_{ds2} & g_{ds4} are the small signal channel conductance, given as

$$g_{ds} = \frac{\partial i_D}{\partial v_{DS}} = \lambda I_D$$

The gain of the second stage is that of a common source amplifier obtained from [27],

$$A_{v2} = -g_m R_D = -\frac{-g_{m6}}{g_{ds6}+g_{ds7}} = \frac{-g_{m6}}{(\lambda_6+\lambda_7)I_6}$$

So the overall gain of the amplifier is given by,

$$A_v = A_{v1}.A_{v2} = \frac{2g_{m1}g_{m6}}{(\lambda_2+\lambda_4)I_5\,(\lambda_6+\lambda_7)I_6} \tag{2}$$

In GP formulation, this equation can be modeled as a monomial and if the minimum gain requirement is A_{vmin} , which is provided in the specification, then we can formulate the gain constraint as follows:

```
OpenGainConstr = [
      ((2*kn*W(1)/L(1)*I(1))^0.5)/gm1==1;
      ((2*kp*W(6)/L(6)*I(6))^0.5)/gm6==1;
      (2*gm1*gm6)/(I(5)*I(6)*(lamda2+lamda4)* (lamda6+lamda7))/Av==1;
      Avmin/Av<=1;
      ];
```

Here, *Av* is the calculated gain from the GP formulation, which must be greater than the specified minimum gain *Avmin*.

4.1.2. Pole Constraint

In order to maintain sufficient phase margin, the non-dominant poles of the circuit should be at least 10 times higher than the dominant pole which controls the unity gain bandwidth. To ensure that the compensating capacitance *Cc* is used to place the poles further away from each other. This method is known as pole splitting.

The dominant pole of the amplifier circuit is given by [27]

$$pole_1 = \frac{g_{m1}}{C_c}$$

If we consider the parasitic interconnect capacitance then,

$$pole_1 = \frac{g_{m1}}{(C_c + C_{db2} + C_{db4})} \tag{3}$$

If C_1 is taken as the gate capacitance of M6 then its expression is found from [23] :

$$C_1 = C_{gs6} + C_{db2} + C_{db4} + C_{gd2} + C_{gd4}$$

where C_{db} incorporates both intrinsic device drain to body capacitance and the interconnect capacitance. The total output capacitance C_{tl} is obtained from [23]

$$C_{tl} = C_{load} + C_{db6} + C_{db7} + C_{gd6} + C_{gd7}$$

The non-dominant output pole is given by [23]

$$pole_2 = \frac{g_{m6}}{C_1 C_c + C_1 C_{tl} + C_c C_{tl}} \tag{4}$$

A constraint is added to put this $pole_2$ at least 10 times away from $pole_1$.

$$\frac{10\, pole_1}{pole_2} \leq 1 \tag{5}$$

All the device capacitances, C_{db}, C_{gs}, C_{gd} etc. are modeled either in posynomial or monomial form in terms of the equations found in either the *TSMC 0.18um* or *TSMC 90nm* technology. Since the equations are the property of TSMC and are not allowed to be distributed, the equations are not shown here. But they can be easily found for the respective foundry located in /CMC/*kits*. The parasitic interconnect capacitances are also formulated using the floorplan and added to the sizing constraints.

The total drain-body capacitance has two parts, one comes from the device parasitic capacitance, C_{dbt} and another comes from the parasitic interconnect capacitance, C_{dbi}. Here, *m1ActAreaCap* and *m1FieldAreaCap* are unit capacitances from metal 1 to active region and diffusion region respectively obtained from technology files. The interconnect capacitance is modeled using the minimum size floorplan and finding the Manhattan distance from routing and multiplying with the unit capacitances. The floorplan is discussed in detail later and shown in Figure 20. In the formulation, *W(x)* is the width of transistor *x*, *d* is the minimum allowable distance between two transistors, *MetalWidth* is the width of the interconnect metal line which is taken as the minimum allowed by respective technologies. For example, the interconnect capacitance at the drain node of M2 can be modeled as

```
(Model for intrinsic device Cdb from foundry equation)<=Cdb2t;
((W(2)*m1ActAreaCap/2+d*m1FieldAreaCap+W(4)*m1ActAreaCap/2+d*m1Fie
ldAreaCap+Width*m1ActAreaCap/2)*MetalWidth*1e6)/(Cdb2i)<=1;

(Cdb2i+Cdb2t)/Cdb2<=1;
... ... ...
```

Similarly, the gate to source C_{gs} and gate to drain capacitance, such as C_{gd} can be modeled in *TSMC 0.18um* and *TSMC 90nm* technology in the same manner using the equations available from the foundry as,

After modeling all the intrinsic capacitances and interconnect parasitics, the pole constraints are added, to finish the formulation of the pole requirements.

```
((2*kn*W(1)/L(1)*I(1))^0.5)/gm1==1;
(Cc+Cdb2+Cdb4)<=tp(19);
pole1/(gm1/tp(19)/2/pi)<=1;

(C1*Cc+C1*Ct1+Cc*Ct1)/tp(18)<=1;
(gm6/tp(18))==pole2;
(10*pole1)/pole2<=1;
```

4.1.3. Unity Gain Bandwidth Constraint

The dominant pole controls the unity gain bandwidth of the amplifier. If the minimum required UGF is w_c which is given as a constant in specification, then, $pole_1$ is made equal to w_c .

```
pole1/wc==1;
```

4.1.4. Input Common Mode Range

It is the voltage up to which the amplifier can amplify the difference between the two differential inputs. Considering M5 works in saturation, the equations for positive and negative ICMR are taken from [27] which are as follows:

$$Positive\ ICMR = V_{DD} - \sqrt{\frac{I_5}{\beta_3}} - |V_{t,max}| + |V_{t,min}|$$

$$=> Positive\ ICMR + \sqrt{\frac{I_5}{\beta_3}} + |V_{t,max}| = V_{DD} + |V_{t,min}| \quad (6)$$

$$Negative\ ICMR = V_{SS} + \sqrt{\frac{I_5}{\beta_1}} + |V_{t,max}| + V_{DS5,sat} \quad (7)$$

where,

$$\beta = \mu C_{ox} \frac{W}{L}$$

Since GP solver doesn't support negative value, the equation for Positive ICMR is rearranged and implemented as follows:

```
ICMRconstr = [
    (kp*W(3)/L(3))/Beta3==1;
    ((I(5)/Beta3)^0.5)/ti11==1;
    (VicmrMax+ti11+VtMax)/(Vdd+VtMin)<=1; %Positive ICMR

    (kn*W(1)/L(1))/Beta1==1;
    ((I(5)/Beta1)^0.5)/ti22==1;
    (Vds5sat+Vss+ti22+VtMax)/(VicmrMin)<=1;%Negative ICMR

   (kn*W(5)/L(5))/Beta5==1;
    ((2*I(5)/Beta5)^0.5)/Vds5sat==1;
];
```

4.1.5. Slew Rate Constraint

Slew rate is the ability to sink current and puts a limit of voltage rate at the output.

The equation for slew rate for the circuit taken from [27] is given by,

$$SR = \frac{I_5}{C_c} \tag{8}$$

The equation is simply modeled as monomial using the specification of the accepted slew rate,

```
SlewConstr = [
     SRmin/(I(5)/Cc)<=1;
     ];
```

4.1.6. Power Constraint

There is a maximum allowable power which is dissipated in the circuit specified as P_{max}. The power consumed is calculated as the product of the total current flowing in all the branches to the supplied voltage and inserted as another constraint in GP.

```
PowerConstraint = [
    ((Vdd-Vss)*(Ibias+I(5)+I(7)))/P<=1;
    P<=Pmax;
    ];
```

4.1.7. Current Constraints

The current bias is taken as a design variable which has to be lower than a certain specified value $I_{biasMax}$. As the transistor dimension of M1 & M2 and M3 & M4 is identical, I_1 and I_2 have the same current. The same is true for I_3 and I_4 as they are in series with I_1 and I_2 respectively and connected as current mirror. The current of each branch is $^1/_2$ of the sink current which is I_5. I_5 is connected as current mirror with the bias current, so the same current flows.

The current constraints are modeled as follows:

```
gpvar Ibias;
CurConstraint = [

    Ibias/IbiasMax<=1;
    I(8)/Ibias==1;
    I(5)/Ibias==1;
    I(6)/I(7)==1;
    I(1)/(I(5)/2)==1;
    I(2)/(I(5)/2)==1;
    I(3)/I(1)==1;
    I(4)/I(2)==1;
    ];
```

4.1.8. Device Size Constraints

In the circuit W_1 and W_2 are identical as well as W_3 and W_4. Also all the devices are more than the minimum allowable transistor width supported by the technology like 0.22um for *TSMC 0.18um* technology and less than a pre-specified maximum transistor width. They are also put as a constraint to the GP solver.

```
SymMatchConstr = [
      W(1)/ W(2) == 1;
      W(3)/W(4) == 1;
      ];

DevSizeConstr = [
          W(1)/Wmax <= 1;
          W(2)/Wmax <= 1;
          .........
    ];
```

Also, the minimum allowable width of the transistor in 0.18um technology is 0.22um. Therefore another constraint is added to ensure that.

```
constr_x = ones(n,1)*0.22e-6 <= W;
```

4.1.9. Floorplanning Constraints

The floor-planning is done using a similar method to that discussed in chapter 3. A Cartesian coordinate is attributed to each corner of all the transistors and they are placed at a minimum allowable distance, *d* from each other. The schematic and the corresponding floorplan are shown in Figure 20.

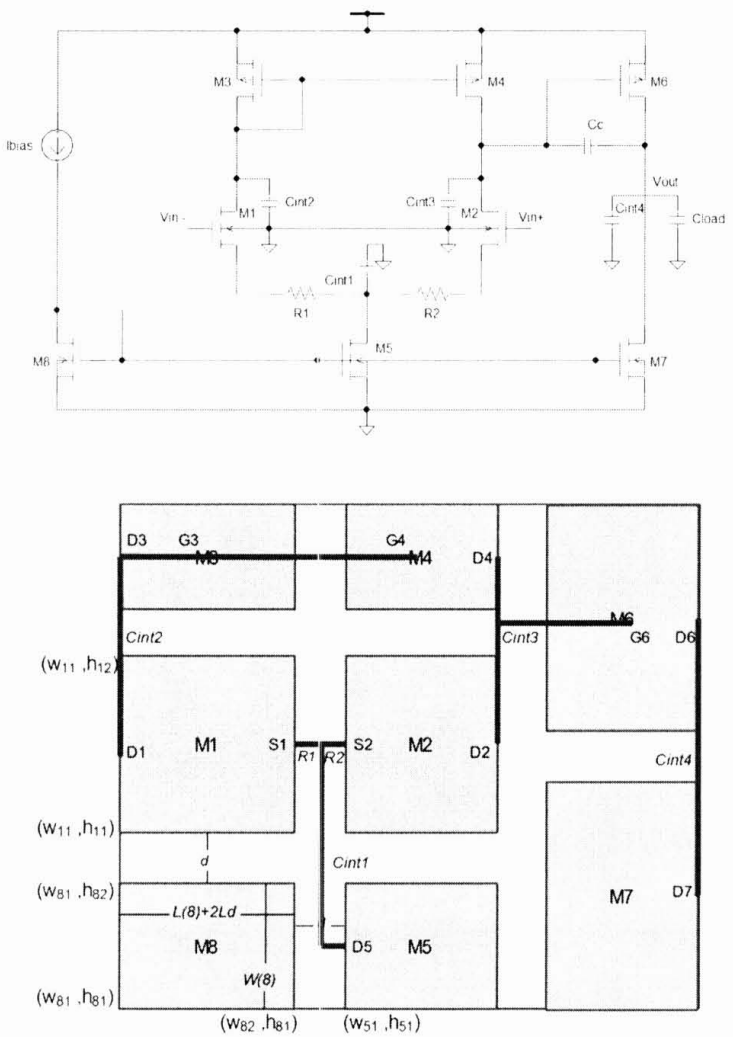


Figure 20: Schematic and Floorplan of Two Stage N-input Op Amp.

The symmetry is kept by placing the transistors with the same width side by side to remove the interconnect capacitance mismatch arising from the layout. The minimum allowable distance between two transistors is *d*. Here *W(i)* is the width of the *i*th transistor and the variable *width* is used to represent the sum of channel length of *i*th transistor *L(i)* and the lateral diffusion of the drain and source region L_d, so $width=L(i)+2*L_d$ as shown in the floorplan in the figure. The interconnect resistances and capacitances are also shown in the figure which were used inside the pole constraint.

The formulation of the floorplanning constraint is given below:

```
%Floorplanning

heightConstr = [

     (d+h82)<=h11;
     (d+h82)<=h21;
     (d+h52)<=h21;
      h11==h21;
     (d+h12)<=h31;
     (d+h22)<=h41;

     (d+h72)<=h61;
     (W(1)+h11)<=h12;
     (W(2)+h21)<=h22;
     (W(3)+h31)<=h32;
     (W(4)+h41)<=h42;
     (W(5)+h51)<=h52;
     (W(6)+h61)<=h62;
     (W(7)+h71)<=h72;
     (W(8)+h81)<=h82;

];

widthConstr = [

     (d+w82)<=w51;
     (d+w12)<=w21;
     (d+w32)<=w41;
     (d+w52)<=w71;
     (d+w22)<=w61;
     (d+w42)<=w61;
     (d+w42)<=w71;
```

```
    (w11+Width)<=w12;
    (w21+Width)<=w22;
    (w31+Width)<=w32;
    (w41+Width)<=w42;
    (w51+Width)<=w52;
    (w61+Width)<=w62;
    (w71+Width)<=w72;
    (w81+Width)<=w82;


];


% constraints
floor_constr = [
    heightConstr;
    widthConstr;
];
```

4.1.10. Area Constraint

The main goal of this optimization is to find the minimum size of the transistors so that the circuit still satisfies all the required performance specifications. So the area is calculated from the floorplan and an area constraint is used when calling GP solver.

```
AreaConstr = [
    (max(w72, w62)*max(h82,max(h52,h72)))/Area<=1;
    Area<=Areamax;
    ];
```

Finally we can call the GP solver and optimize the design in terms of area and finish this layout aware synthesis. We put all the constraints in a list and pass it to the *gpsolve* function to optimize for area.

```
constr = [
      SymMatchConstr;
      DevSizeConstr;
      InOffsetConstr;
      CurConstraint;
      SlewConstr;
      ICMRconstr;
      poleConstr;
      PowerConstraint;
      OpenGainConstr;
      AreaConstr;
      floor_constr;
      constr_x;
    ];

[min_Area solution status] = gpsolve(Area,constr);
assign(solution);
```

4.1.11. GP Solution

The two stage N-input channel op amp is synthesized in both TSMC 0.18μm and *TSMC 90nm* CMOS technology. The transistor length was treated constant as 1 μm and the transistor width and bias current were defined as design variables. To differentiate the importance of adding the floorplan and the layout induced capacitive and resistive effects, at first only the intrinsic device capacitances are passed to the GP solution ignoring the interconnect capacitance. After the GP solution, the obtained transistor sizes are used to floorplan and then calculate the interconnect capacitance and resistances. Then the circuit performance is simulated in Cadence.

In the next case, all the interconnect capacitances (C_{int1}, C_{int2}, C_{int3} & C_{int4} in the Figure 20) of the different nets are added to the GP constraints. Similarly, the obtained GP result is passed for pre-layout verification by simulation in Cadence. This design with

interconnect parasitics considered is then compared with the previous design to find the contribution from considering interconnect parasitics.

The optimized design variables after the GP solution for both *TSMC 0.18um* and *TSMC 90nm* technology is shown in Table VII. For the optimized device sizes, the resultant parasitic components are also shown in the table. For the synthesis without considering interconnect capacitance, the parasitic components are calculated after the GP solution is obtained. For the synthesis with interconnect capacitances, the obtained value from GP is directly shown.

Table VII: Optimal Design Variables for Two Stage N-input Op-Amp

Var.	Two Stage N-input Op-Amp			
	TSMC 0.18μ		*TSMC 90nm*	
	Without Interconnect Cap.	***With Interconnect Cap.***	***Without Interconnect Cap.***	***With Interconnect Cap.***
L1-L8 (μm)	1.00	1.00	1.00	1.00
W1=W2 (μm)	7.1	7.2	1.94	1.8
W3,W4 (μm)	1.4	1.37	5.11	5.50
W5 (μm)	0.5	0.51	3.35	3.60
W6 (μm)	69.6	69.6	69.6	69.6
W7 (μm)	13.0	13.0	22.8	22.7
W8 (μm)	1.4	1.38	1.83	1.85
Cc (pF)	0.5	0.5	1.2	1.2
Ibias (μA)	5.0	5.0	5.0	5.0
Cint1 (fF)	0.27	0.13	0.15	0.04
Cint2 (fF)	0.24	0.24	0.14	0.04
Cint3 (fF)	0.22	0.07	0.14	0.03
Cint4 (fF)	0.55	0.48	0.32	0.26
R1 (Ω)	11.6	2.03	21.3	4.49
R2 (Ω)	6.02	2.03	10.5	4.49
Cload(pF)	1	1	1	1
CPU time (s)	0.97	1.12	1.32	1.46

4.1.12. Pre-layout Simulation Result

The optimal design variables obtained from GP for both *TSMC 0.18um* and 90nm technology are simulated in Cadence simulation for a pre-layout verification and to observe the effect of considering the interconnect capacitance during design synthesis.

1. ***TSMC 0.18um* technology**

The resulting ac analysis for 0.18um technology is given in Figure 21. The solid line is the result obtained for the design considering the interconnecting parasitics and the dashed line is for the synthesis without considering the interconnect capacitance.

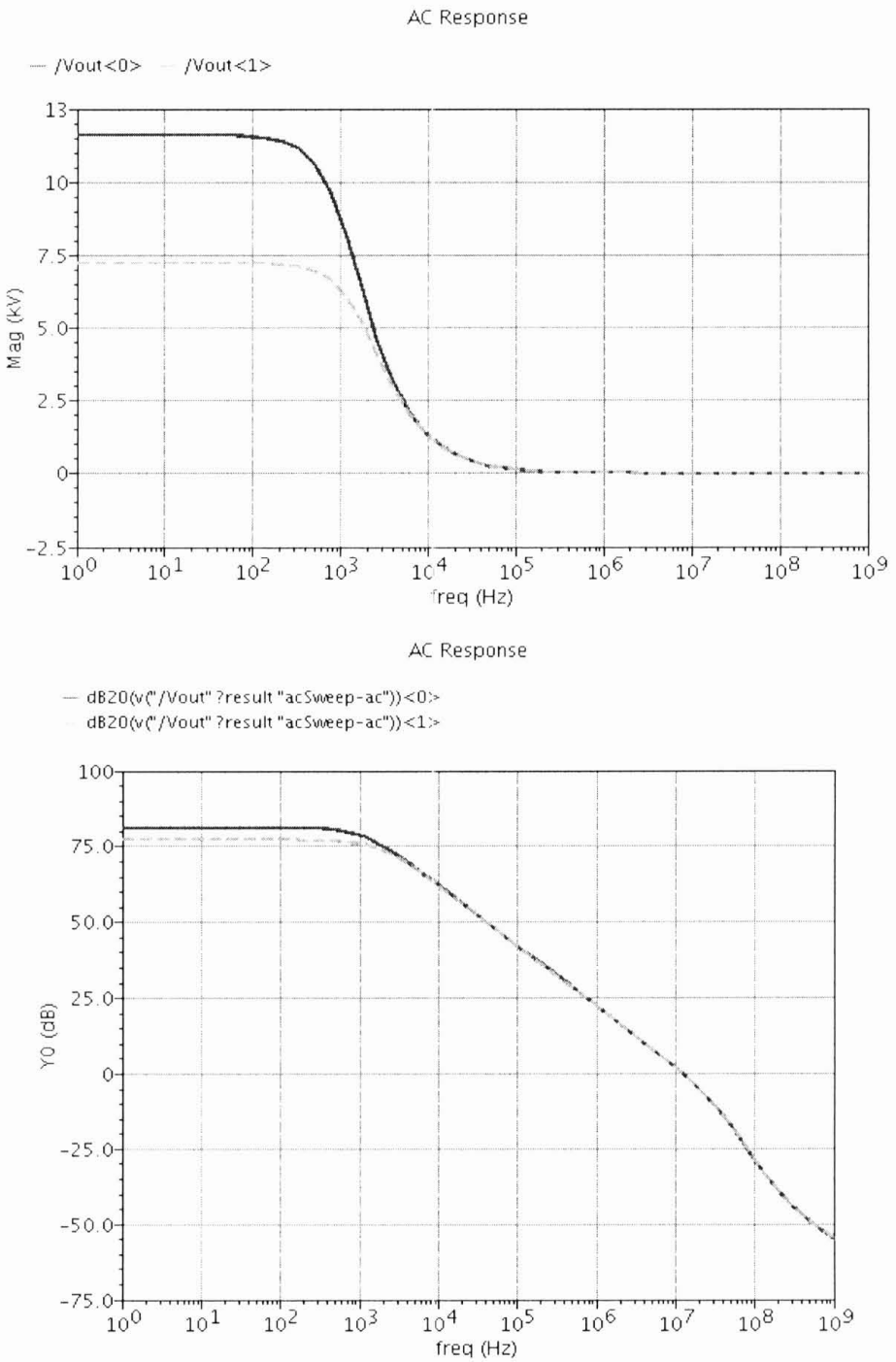


Figure 21: Pre-layout in Cadence for Two Stage NMOS Op Amp in CMOS 0.18um Technology. (Magenta-Solid: Design with interconnect parasitics, Red-Dashed: Design without interconnect parasitics)

The other performance measures are shown in Table VIII. Although both synthesis met the target specification, it is obvious from the simulation that the synthesis that considered interconnect capacitance and utilized a minimum size floor-plan keeping symmetry into consideration gives a much better result in terms of gain and unity gain bandwidth than the one without considering the interconnect parasitic effect. Because the synthesis considering the interconnect capacitance with minimum size floorplan results in a design which has low parasitic interconnect capacitance and resistance, the simulated result shows major improvement and shows of the advantage of including parasitic effects inside the circuit sizing phase.

Table VIII: Prelayout Simulation Result for Two Stage N-input Op Amp in *TSMC 0.18um* technology

Variable	Specification	TSMC 0.18 μm	
		Without Interconnect Cap.	*With Interconnect Cap.*
Gain	$\geq$ 65 dB	77.18	81.29
UGF	$\geq$ 5MHz	12.41	12.58
Phase Margin	$\geq 60^0$	63.46	62.78
CMRR	$\geq$ 80 dB	105.9	108.1
Power	$\leq$1000μW	244.6	245.6

2. *TSMC 90nm* technology

Similarly the simulation result for *TSMC 90nm* technology for both the synthesis process is given in the following Figure 22.

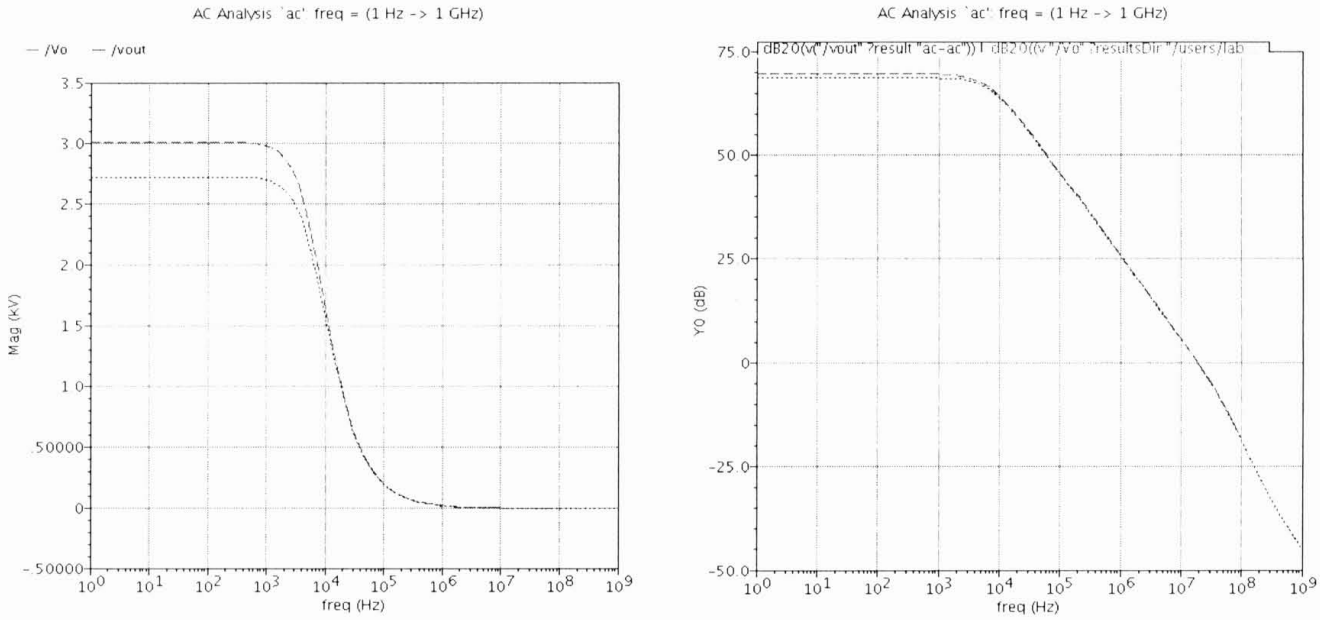


Figure 22: Pre-layout in Cadence for Two Stage NMOS Op Amp in *TSMC 90nm* technology. (Red-Dashed: Design with interconnect parasitics, Blue-Dotted: Design without interconnect parasitics)

Because the 90*nm* technology is constrained by a lower supply voltage of 1.2V instead of 1.6V in 0.18*um* technology, the simulation result did not show the same radical improvement for the synthesis with interconnect capacitance compared to the one without interconnect capacitance. Still improvements in both gain and unity gain frequency is found when interconnect parasitics is considered. The other performance comparisons are shown in Table IX. To facilitate comparison with post-layout simulation, the large transistors M6 and M7 of the design that considers interconnect parasitics, are divided into several smaller transistors. M6 which has a width of 69.6*um* is divided into six smaller transistors of 11.6*um* with a number of multiplier 6 and M7 which has a width of

22.7*um* is divided into four smaller transistors of width 5.675um with a multiplier number of 4. This will help to get a layout which is nice rectangular or square in shape. If a single large transistor were used, that would take a large space in the chip wafer.

Table IX: Simulation Result of Two Stage N-input Op-Amp

Variable	Specification	*TSMC 90nm*	
		Without Interconnect Cap.	*With Interconnect Cap.*
Gain	≥ 65 dB	68.69	69.57
UGF	≥ 5MHz	18.81	19.19
Phase Margin	$\geq 60^0$	63.42	62.85
CMRR	≥ 80 dB	82.58	83.43
Power	≤1000μW	52.84	49.89

4.1.13. Post Layout Simulation

After the pre-layout verification is successful, the next step in the design synthesis flow, is to verify the design with interconnect parasitics by a doing a post layout simulation and compare the result with the pre-layout simulation to test the accuracy of the parasitic estimation used in the proposed parasitic aware synthesis flow. The post layout simulation is obtained by creating a layout, extracting its parasitic components and simulating with the extracted view of the circuit.

At the moment in the available environment, *TSMC 0.18um* technology supports extraction using DIVA extract and *TSMC 90nm* supports extraction by Calibre from Mentor Graphics. In this case, the optimum design obtained in *TSMC 90nm* technology considering all interconnect parasitics, is used to verify the post layout simulation because

in the test the Calibre Extraction is found to be more detailed. The post layout simulation is shown in the following part.

The first step for post-layout simulation is to draw the amplifier raw circuit in Cadence schematic with appropriate pins connected to the supply voltage and ground and all the inputs and bias current. The raw schematic is shown in Figure 23.

Figure 23: Raw circuit for two stage miller OTA Amp in CADENCE.

The next step is to export the schematic to layout XL to generate the corresponding layout. All the transistors, the compensating capacitance and pins are placed manually. Then the Virtuoso Chip Assembly Router (VCAR) is used to automatically route between the internal nodes. The resultant routing is transferred to layout window which shown in Figure 24. The Figure shows all the transistors and the large compensating capacitance which takes most of the space in the layout. From the layout, Calibre Parasitic Extraction tool PEX is used to extract the netlist from this layout.

Figure 24: Layout generated from schematic using Layout XL for an 8-bit DAC.

A config file is created to simulate the op-amp with the extracted netlist and the circuit is simulated. It is then compared with the pre-layout simulation to see the actual layout induced effects compared to the proposed theoretical and previous calculations. Figure 25 shows the comparison between the pre and post-layout simulations. The solid blue (diagonal) shows the post-layout result for unity gain (red) line and the result for pre-layout simulation with the design considering interconnect parasitics. Table X lists the key performance values for comparison.

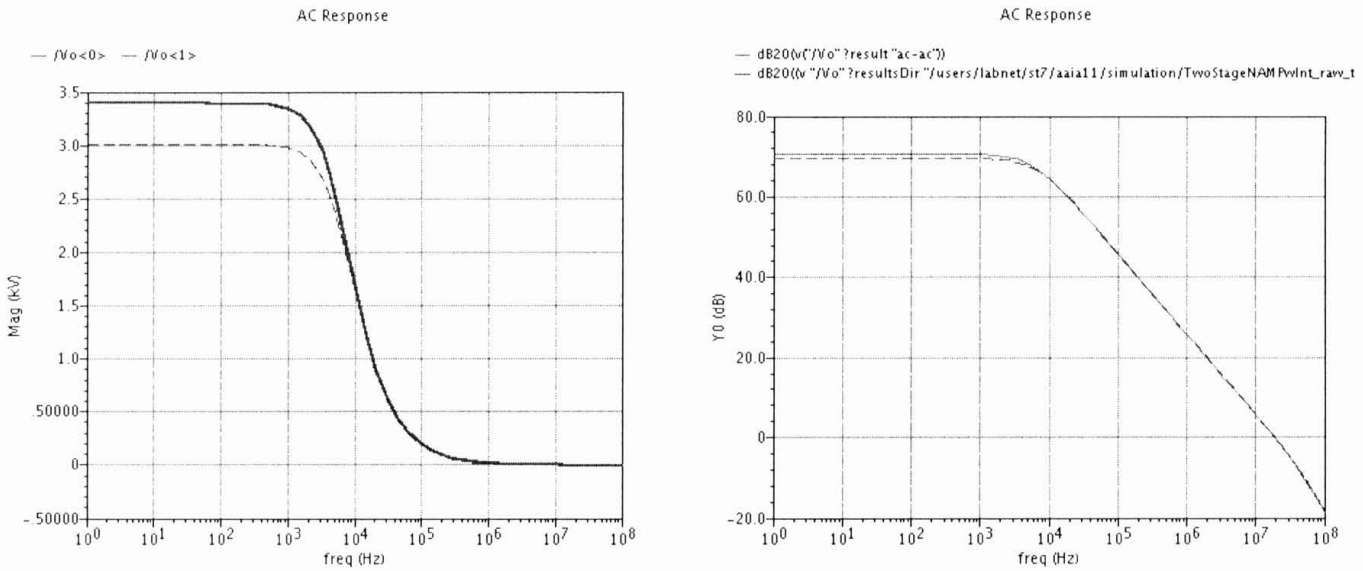


Figure 25: Comparison of Pre and Post Layout Simulation for Two Stage N-channel input Op Amp in *TSMC 90nm* technology (Magenta-Solid: Post Layout, Red-Dashed: Pre-layout).

Table X: Comparison between pre and post layout simulation result in *TSMC 90nm*

Variable	Specification	*TSMC 90nm*	
		Pre-layout Simulation.	*Post Layout simulation*
Gain	≥ 65 dB	69.57	70.61
UGF	≥ 5MHz	19.19	18.95
BW (3dB)	UnSpecified	67965.17	61503.65
Phase Margin	$\geq 60^0$	62.85	62.61

From the Figure 25 and the Table X, the post layout performance satisfies the initial target performance specifications. Although the unity gain bandwidth is slightly lower than the pre-layout simulation, it is fairly close and much higher than the target bandwidth and the slight increase gain indicates a similar total gain bandwidth product between the pre-layout simulation result and post-layout simulation. The reason for the

slight increase in gain is due to the increase in total resistance which contributes to the gain. The resistance of the Prelayout schematic and post layout extracted netlist is measured by adding an ac current source of magnitude 1 to the output node and removing all ac components from the circuit and keeping the all dc components for proper biasing. The voltage at the output gives the total resistance. The total resistance from the schematic for pre layout case is found to be 41.91kΩ whereas the total resistance from the extracted netlist for post layout case is found to be 46.28kΩ. As the gain is proportional to $g_m * R_{out}$, this explains the slight increase in gain.

So the design obtained considering the parasitics using the proposed parasitic aware synthesis methodology meets the entire performance requirement, therefore can handle the layout induced mismatch as seen by the post-layout simulation.

The next section discusses the design of two stage p-input Op Amp with the proposed methodology.

4.2. Two Stage P-Channel Input Differential Operational Amplifier

The structure of this two stage p-channel differential op amp is similar to that of the two stage n-channel op amp, which has a differential input stage followed by a common source stage shown in Figure 26.

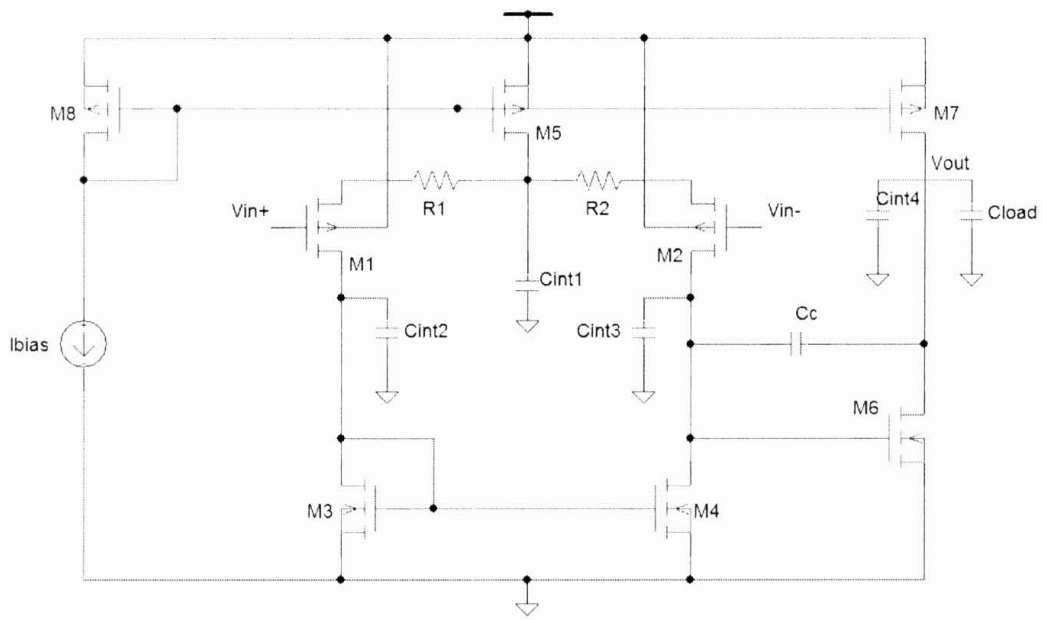


Figure 26: Two Stage P-input Amplifier [23].

The choice of the input stage type depends on several factors. There are a few advantages of using the p-channel input type over the n-channel mentioned in [28].

- PMOS-input op-amp has a better slew rate than the NMOS input op amp.
- If an output buffer is used to drive a resistive load, it is usually selected to be an n-type source follower. Having a PMOS as input and an NMOS at the output stage

causes an increase in the transconductance at the output stage which is favourable for unity gain frequency.

- The 1/f noise is lower for PMOS compared to the NMOS.

4.2.1. GP constraints

The circuit is designed following the same procedure as an n-input op amp. For a set of given specifications, the performance specifications are modeled either as posynomial or monomial form. A floorplan is created considering the minimum allowable distance. Interconnects are drawn as Manhattan distances and symbolic equations are formed for interconnect to substrate capacitances and the interconnect resistances. Also all the device capacitances are modeled in symbolic form using the distinct equations of the respective technology. These parasitics are included inside the pole constraint equations to lead to parasitic aware synthesis. The details of the formulation are provided as follows:

4.2.2. Gain Constraints

The voltage gain of the total amplifier circuit is given by [23],

$$A_v = \left(\frac{g_{m2}}{g_{02}+g_{04}}\right)\left(\frac{g_{m6}}{g_{06}+g_{07}}\right)$$

This can be written as

$$A_v = \frac{2C_{ox}}{(\lambda_n+\lambda_p)^2}\sqrt{\frac{\mu_n\mu_p W_2 W_6}{L_2 L_6 I_1 I_7}} \tag{9}$$

This can be formulated as a monomial constraint.

```
OpenGainConstr = [
    (Un*Up*(W(2)*W(6))/(L(2)*L(6)*I(1)*I(7)))/tempA1==1;
    (2*Cox/((lamdaN+lamdaP)^2)*(tempA1^0.5))/Av==1;
    Avmin<=Av;
    Av/Avmax<=1;
    ];
```

4.2.3. Pole Constraints

There are four poles in the circuit. The dominant pole can be written as the following equation taken from [23]

$$pole_1 = \frac{g_{m1}}{C_c}$$

The first non-dominant pole is the output pole which is given in [23]

$$pole_2 = \frac{g_{m6}}{C_1C_c+C_1C_{tl}+C_cC_{tl}} \tag{10}$$

Where C_1, the M6 gate capacitance, C_2, the M3 gate capacitance and C_{tl} are obtained from [23]:

$$C_1 = C_{gs6} + C_{db2} + C_{db4} + C_{gd2} + C_{gd4}$$
$$C_2 = C_{gs3} + C_{gs4} + C_{db1} + C_{db3} + C_{gd1}$$
$$C_{tl} = C_{load} + C_{db6} + C_{db7} + C_{gd6} + C_{gd7}$$

The second non-dominant pole is the mirror pole which is taken from [23]

$$pole_3 = \frac{g_{m3}}{C_2} \tag{11}$$

And the third non-dominant pole is

$$pole_4 = \frac{g_{m6}}{C_1} \tag{12}$$

The outline of the pole constraint for 0.18um technology is as follows:

```
PoleConstr = [

      (Using equation from foundry for intrinsic Cdb)==Cdb1t;
      ((W(1)*m1ActAreaCap/2+d*m1FieldAreaCap+W(3)*m1ActAreaCap/2+Width*m
      1ActAreaCap+d*m1FieldAreaCap+Width*m1ActAreaCap/2)*MetalWidth*1e6)
      /Cdb1i<=1;
      (Cdb1t+Cdb1i)/Cdb1<=1;
      ... ... ...
```

The intrinsic device capacitance from the drain to body of M1 is denoted as C_{db1t} and the interconnect contribution is denoted as C_{ab1i}. The intrinsic capacitance is formulated using the equation obtained from the foundry model and the interconnect capacitance is calculated using the floorplan shown in Figure 18, as discussed in chapter 3, by calculating the lengths of interconnects and the unit capacitance values, which are also available from technology model parameters. For example, the drain to body capacitance at the drain of M6 and M7 can be formulated as

```
      (Using equation from foundry for intrinsic Cdb)== Cdb6t;
      (W(7)*m1ActAreaCap/2+h71*m1FieldAreaCap+W(6)*m1ActAreaCap/2)*Metal
      Width*1e6)/Cdb6i1<=1;
      (Cdb6i2+h62*m1FieldAreaCap*1e6)<=Cdb6i1;
      (Cdb6t+Cdb6i2)/Cdb6<=1;

      (Using equation from foundry for intrinsic Cdb)== Cdb7t;
      (Cdb7t)/Cdb7<=1;% M6 & M7 drain same node so the interconnect cap
                    % added once.
```

Similarly all the other device capacitance and interconnect capacitances are formulated. Then using the equations above the pole constraint is modeled as follows

```
    (Cgs6+Cdb2+Cdb4+Cgd2+Cgd4)<=tp(5);
    C1/tp(5)==1;

    (Cgs3+Cgs4+Cdb1+Cdb3+Cgd1)<=tp(17);
```

```
C2/tp(17)==1;

(Cload+Cdb6+Cdb7+Cgd6+Cgd7)/Ctl<=1;
(gm6*Cc)*(C1*Cc+C1*Ctl+Cc*Ctl)<=tp(9);
pole2/tp(9)==1;

(gm1/Cc)/pole1<=1;
pole3/(gm3/C2)==1;
pole4/(gm6/C1)==1;

pole1/pole2/0.1<=1;
pole1/pole3/0.1<=1;
pole1/pole4/0.1<=1;

];
```

Like the n-input channel two stage op amp, the interconnect capacitance is put in symbolic form and are added when calculating the poles. To make there is sure sufficient phase margin, the non-dominant poles are placed at least 10 times away from the main pole.

4.2.4. Unity Gain Bandwidth Constraint

The dominant pole is made equal to the required unity gain frequency, w_c to satisfy the bandwidth requirement.

```
BwConstr = [
    wc/(pole1)==1;
    ];
```

4.2.5. Input Common Mode Constraints and output swing

The positive common mode range puts a constraint on M5 which is [23]

$$\sqrt{\frac{I_1 L_1}{\mu_p C_{ox}/2W_1}} + \sqrt{\frac{I_5 L_5}{\mu_p C_{ox}/2W_5}} \leq V_{DD} - V_{ICMR,max} + V_{tp} \tag{13}$$

The negative common mode range puts constraint on M1 which gives [23]

$$\sqrt{\frac{I_1 I_3}{\mu_n C_{ox}/2W_3}} \leq V_{ICMR,min} - V_{SS} + \sqrt{\frac{I_5}{\beta_1}} - V_{tp} - V_{tn} \tag{14}$$

The output swing puts two more constraint on M6 and M7 which are [23]

$$\sqrt{\frac{I_7 L_6}{\mu_n C_{ox}/2W_6}} \leq V_{out,min} - V_{SS} \tag{15}$$

$$\sqrt{\frac{I_7 L_7}{\mu_p C_{ox}/2W_7}} \leq V_{DD} - V_{out,max} \tag{16}$$

All these are put as bias constraint as follows:

```
BiasConstr = [
    (I(1)*L(3)/(kn/2*W(3)))/temp1==1;
    (I(1)*L(1)/(kp/2*W(1)))/temp5a==1;
    (I(5)*L(5)/(kp/2*W(5)))/temp5b==1;
    (I(7)*L(6)/(kn/2*W(6)))/temp6==1;
    (I(7)*L(7)/(kp/2*W(7)))/temp7==1;

    (temp1^0.5)/(VcmMin-Vss-Vtp-Vtn)<=1;
    (temp5a^0.5+temp5b^0.5)/(Vdd-VcmMax+Vtp)<=1;
    temp6/(VoutMin-Vss)<=1;
    temp7/(Vdd-VoutMax)<=1;];
```

4.2.6. Slew Rate constraint

The slew rate can be put as the following constraints [23]

$$\frac{C_c}{2I_1} \leq \frac{1}{SR_{min}} \qquad \frac{C_c+C_{tl}}{I_7} \leq \frac{1}{SR_{min}} \tag{17}$$

```
SlewConstr = [
      Cc/2/I(1)*SRmin<=1;
      (Cc+Ctl)/I(7)*SRmin<=1;
      ];
```

4.2.7. CMRR constraint

The CMRR constraint is put as [23]:

$$CMRR = \frac{2C_{ox}}{(\lambda_n+\lambda_p)\lambda_p}\sqrt{\frac{\mu_n\mu_p W_1 W_3}{L_1 L_3 I_5^2}} \tag{18}$$

```
CmrrConstr = [
    Un*Up*W(1)*W(3)/L(1)/L(2)/(I(5)^2)==tc11;
    2*Cox*(tc11^0.5)/(lamdaN+lamdaP)/lamdaP/CMRR<=1;
];
```

4.2.8. Power and Current Constraints

The power and current constraints are provided in similar manner as in the n-channel input op-amp discussed in the previous section of this chapter.

4.2.9. Area Constraints

The area constraint is modeled from the floorplan and is optimized during solving to find the minimum area.

```
AreaConstr = [
    (max(w72, w62)*max(h82,max(h52,h72)))/Area<=1;
```

```
    Area<=Areamax;
    ];
```

We put all the constraints in a constraints list and gpsolver is used to optimize the circuit in terms of performance targeting area minimization.

```
constr = [
    SymMatchConstr;
    DevSizeConstr;
    AreaConstr;
    InOffsetConstr;
    CurConstraint;
    BiasConstr;
    GateOverConstr;
    PowerConstraint;
    PoleConstr;
    OpenGainConstr
    floor_constr;
    BwConstr;
    DomPolConstr;
    PhaseConstr;
    SlewConstr;
    CmrrConstr;
    floor_constr;
    constr_x;
    ];

[min_Area solution status] = gpsolve(Area,constr);
assign(solution);
```

4.2.10. Results

The similar approach to the two stage n-channel input op amp is followed when synthesizing the p-channel two stage op amp. In the first case, the amplifier is designed both *TSMC 0.18um* and 90nm technology, without considering the interconnect capacitance and resistance. In the next case, to add the parasitics inside synthesis, the parasitic interconnect capacitance and resistance are added to see their contribution and both designs are verified in Cadence simulation. After the successful pre-layout

simulation, the design considering the parasitic effect is used to generate layout in Cadence layout XL in *TSMC 90nm* technology and post layout simulation is performed. The GP solution for the given specification is given below:

Table XI: Optimal Design variable for two stage p-input operational amplifier

Var.	Two stage P-input Op Amp			
	TSMC 0.18μ		*TSMC 90nm*	
	Without Interconnect Cap.	***With Interconnect Cap.***	***Without Interconnect Cap.***	***With Interconnect Cap.***
L1-L8 (μm)	1.00	1.00	1.00	1.00
W1=W2 (μm)	2.0	1.84	1.72	2.05
W3,W4 (μm)	1.4	1.63	0.55	0.71
W5 (μm)	3.38	3.73	1.75	1.75
W6 (μm)	41.6	49.0	16.7	21.3
W7 (μm)	50.7	56.0	26.2	26.2
W8 (μm)	3.3	3.73	1.75	1.75
Ibias (μA)	0.67	0.67	0.83	0.89
Cint1 (fF)	0.28	0.13	0.14	0.04
Cint2 (fF)	0.25	0.11	0.13	0.03
Cint3 (fF)	0.21	0.06	0.12	0.02
Cint4 (fF)	0.61	0.61	0.18	0.14
Cc	0.175	0.175	0.25	0.25
R1 (Ω)	12.9	2.27	20.6	3.04
R2 (Ω)	5.52	2.27	9.78	3.04
Cload (pF)	1	1	1	1
CPU time (s)	1.18	1.31	1.54	1.70

4.2.11. Pre-layout Simulation

Both design obtained in *TSMC 0.18um* and *TSMC 90nm* technology is verified for Pre-layout simulation.

1. *TSMC 0.18um* technology:

The simulation result for *TSMC 0.18um* technology for both synthesis is shown in Figure 27, where the red (solid) is synthesis taking parasitics into account and the blue (dashed) one is the synthesis without considering the interconnect parasitics.

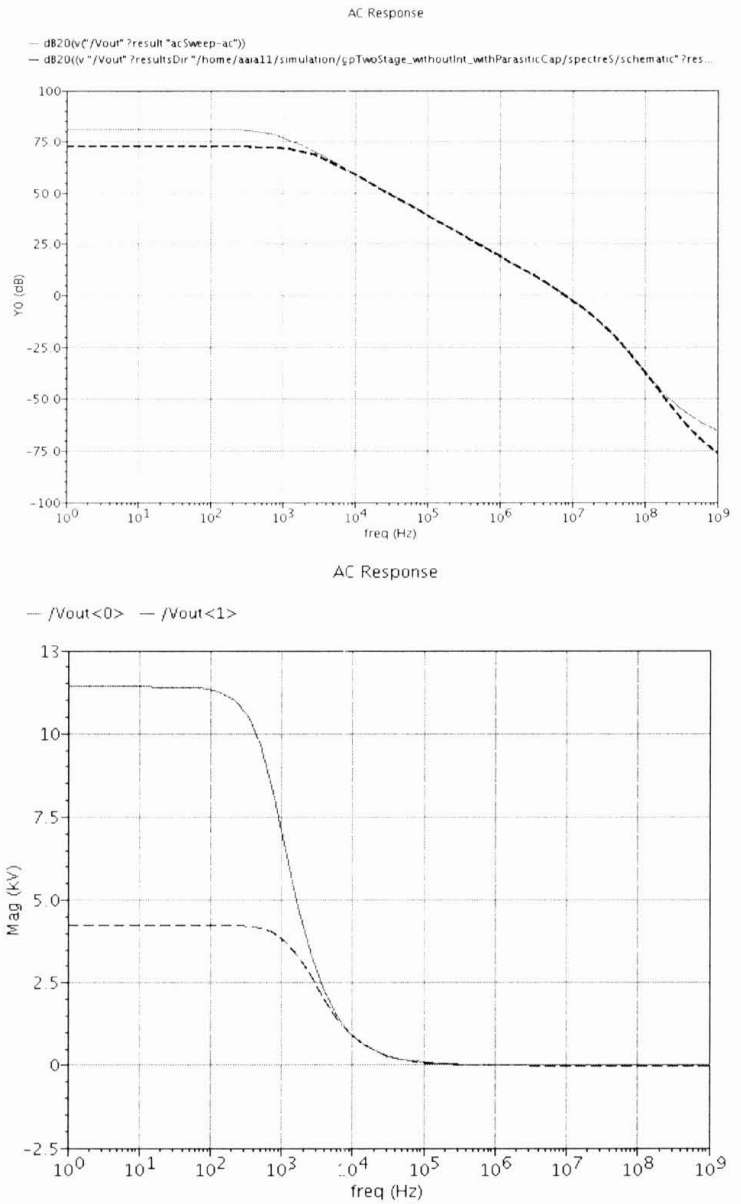


Figure 27: Pre-layout in Cadence for Two Stage PMOS Op Amp in CMOS 0.18um technology. (Red-solid: Design with interconnect parasitics present, Blue-dashed-Design without parasitics)

The other results are shown in the Table XII. Even though both met the performance requirements, the design with parasitics present is gives significantly better gain, unity

gain frequency and CMRR compared to the design without considering the interconnect parasitic effect.

Table XII: Pre-layout simulation result for two stage p-input op amp in *TSMC 0.18um* technology.

Variable	Specification	TSMC 0.18 μm	
		Without Interconnect Cap.	*With Interconnect. Cap.*
Gain	$\geq$ 65 dB	72.60	81.14
UGF	$\geq$ 5MHz	8.17	9.18
Phase Margin	$\geq 60^0$	65.01	61.11
CMRR	$\geq$ 80 dB	107.64	114.76
Power	$\leq$1000μW	20.57	20.57

2. *TSMC 90nm* technology:

The pre-layout simulation result for *TSMC 90nm* technology is given in the following Figure 28, where the red (dashed) line is the result with the design considering interconnect parasitics and the blue (dotted) line is the result obtained from design without considering interconnect parasitics. The design which considers interconnect parasitics are further considered for rectangular or square shaped layout. As a single large transistor is not convenient for obtaining a square or rectangular placement, therefore the large transistors M6 and M7 is divided into four smaller transistors.

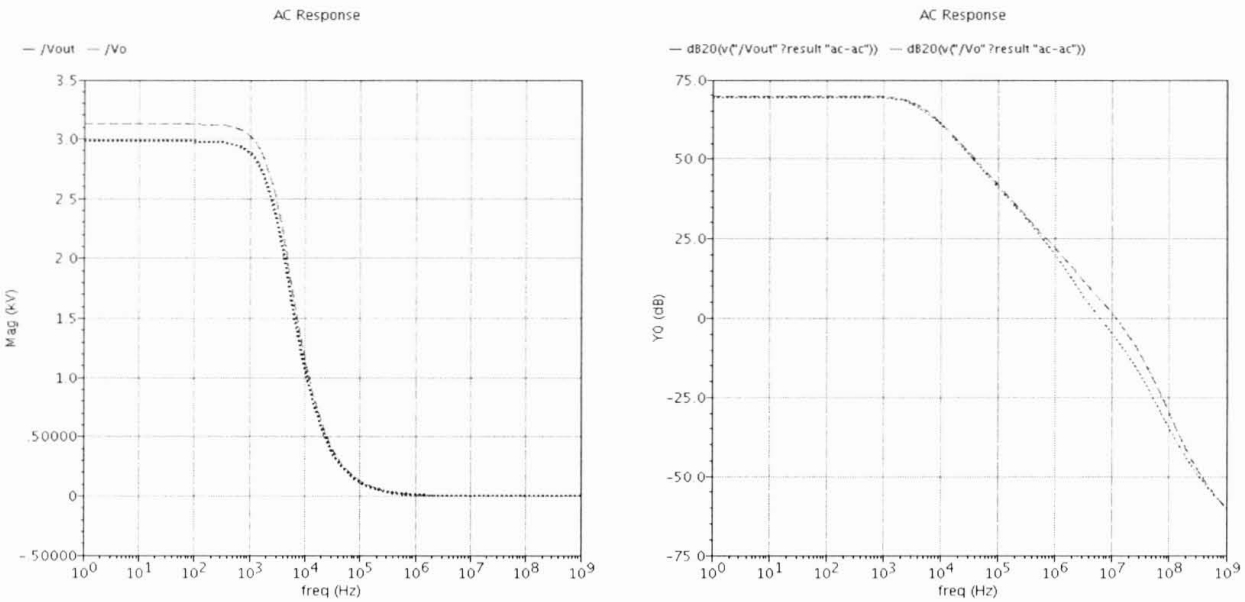


Figure 28: Pre-layout in Cadence for Two Stage PMOS Op Amp in *TSMC 90nm* technology. (Red-dashed: Design with interconnect parasitics in consideration, blue-dotted-without consideration)

Their performance is given in the following Table XIII. Again, as the 90nm technology is constrained by low 1.2V the achieved gain is lower than that of 0.18um technology, but both synthesis approaches satisfies all the performance requirements. The design considering interconnect parasitics shows much better unity gain frequency than its counterpart. The obtained gain is also slightly better than the design without considering interconnect parasitics.

Table XIII: Pre-layout simulation result for two stage p-input op amp in *TSMC 90nm* technology

Variable	Specification	*TSMC 90nm*	
		Without Inct. Cap.	*With Inct. Cap.*
Gain	≥ 65 dB	69.48	69.901
UGF	≥ 5MHz	6.18	11.427
Phase Margin	≥ 60^0	62.38	60.858
CMRR	≥ 80 dB	104.5	103.2
Power	≤1000μW	17.07	18.31

4.2.12. Post-layout Simulation

After a successful pre-layout simulation, the optimum result in TSMC 90nm technology which considers interconnect parasitics during design is used to do a post-layout simulation to find the efficacy of the methodology in actual layout. At first all the raw circuit shown in Figure 25, is drawn in Cadence with appropriate pins connected to the source and ground terminal.

Figure 29: The test schematic of two-stage p-input op amp in TSMC 90nm technology

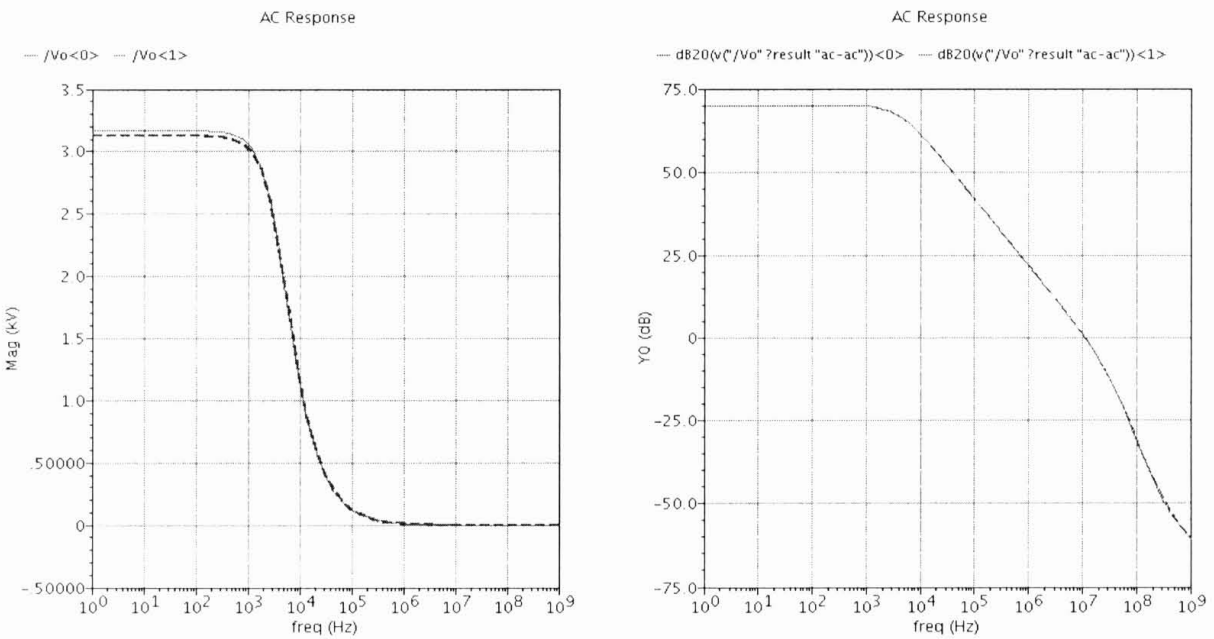
The test bench which is used to test this op amp is given in Figure 30 where the source and ground and all the input signals are driven and output load capacitance is included.

Figure 30: Test Bench used for the two stage p-input op amp

Similar to the n-input two stage op-amp layout-XL is used from Design Synthesis to generate a layout from the new schematic. The transistors and the contacts are manually placed keeping the symmetry in mind. Then the Virtuoso Chip Assembly and Routing is used to draw the interconnections between the nets. The resulting layout is shown in Figure 31.

Figure 31: Generated layout by XL of the two-stage p-input Op amp in AMI05 micro technology.

This generated layout is then extracted by the tool Calibre PEX and extracted netlist is created. The extracted netlist is used to simulate and obtain the post-layout simulation result. Figure 32 shows the post-layout simulation curve in magenta (solid) line and the pre-layout curve in red(dashed) line. Table XIV compares the overall performance of the



pre-layout schematic driven simulation with the post-layout simulation.

Figure 32: Pre and Post Layout Simulation for two-stage p-input Op Amp in *TSMC 90nm*. (Solid-Magenta: Post Layout simulation, Red-dashed: Prelayout simulation)

Table XIV: Pre and Post Layout performance comparison for two-stage p-input Op Amp

Variable	Specification	*TSMC 90nm*	
		Pre-layout Simulation.	*Post Layout simulation*
Gain	≥ 65 dB	69.901	70.01
UGF	≥ 5MHz	11.427	11.13
BW (3dB)	UnSpecified	42635.21	41617.24
Phase Margin	$\geq 60^0$	60.858	60.49

From the table, it is seen that the Cadence pre and post layout simulation is fairly consistent except the unity gain frequency which is a little lower in the post-layout case. But it still much higher than the initial targeted unity gain frequency. The gain at the post layout simulation is 70.01dB a little higher but very close to the 69.901dB obtained in pre-layout simulation. So the gain bandwidth product remains in the same range.

From the pre and post layout verification, it is confirmed that the design obtained using the proposed methodology achieves all performance requirement, while minimizing area. Furthermore, the pre and post layout simulation result is fairly consistent, which shows that the parasitic model used in the synthesis is fairly close to the actual layout induced parasitics.

The next section elaborates on the Cascode Two Stage Operational Amplifier in details.

4.3. Two Stage Operational Amplifier using Cascode Output Stage

The two stage amplifiers discussed before have a few problems like limited unity gain bandwidth and insufficient phase margin for high load capacitance. To improve the UGF some cascode structure can be used at the output stage. One such amplifier is shown in Figure 33 which has cascading second stage.

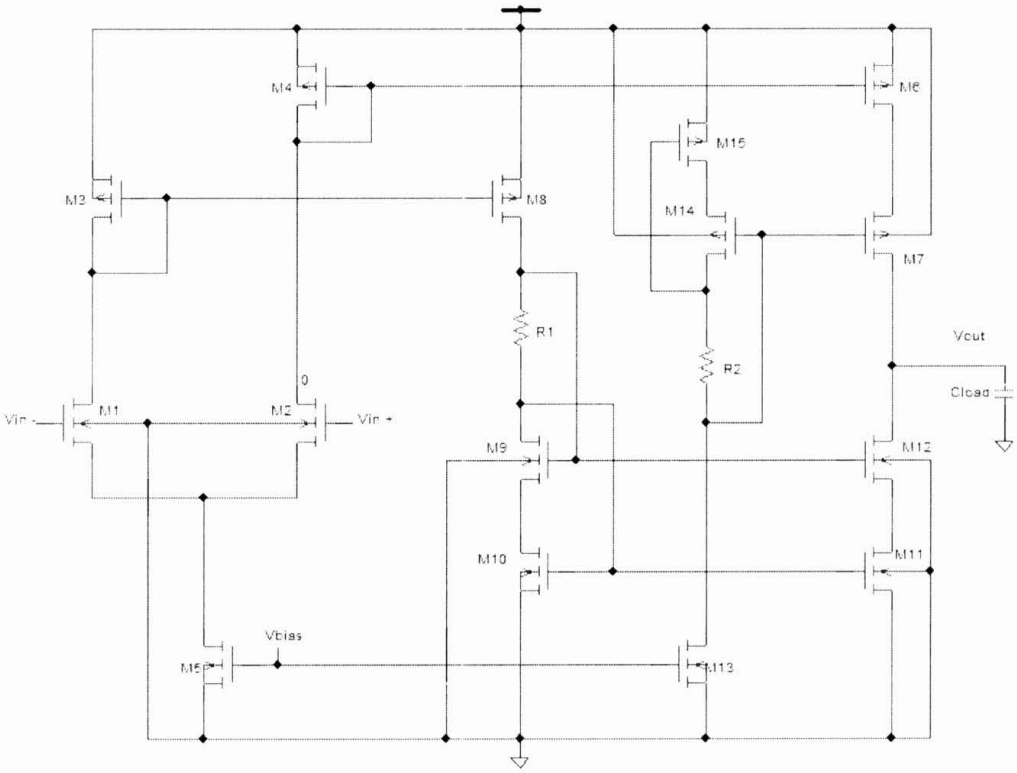


Figure 33: Cascode Two Stage Amplifier [27].

The design objectives are

minimize *Area*

while subject to

gain $\geq$ 65dB

Unity Gain Frequency $\geq$ 10 MHz

Phase Margin $\geq 60^0$

CMRR $\geq$ 80 dB

Power $\leq$ 10mW

Since for both two stage n-input channel and p-input channel op amp, the design considering all parasitics proved to be noticeably better than the design without parasitics considered, the design for cascode two stage amplifier was done considering all the parasitic components. To show that the proposed methodology is not limited to any certain technology, this cascode design is implemented targeting *TSMC 0.18um* technology. In the following chapters, the differential pair comparator and the RF circuits are designed in *TSMC 90nm* technology along with the two stage open loop comparator in *TSMC 0.18um* technology.

The design and optimization steps of this amplifier are as follows:

4.3.1. Gain Constraints

The gain of the first input stage of the cascode two stage amplifier is given by [27]

$$A_{v1} = \frac{g_{m1}}{g_{m3}}$$

The gain of the second stage can be written as [27]

$$A_{v2} = \frac{g_{m6}+g_{m8}}{2} R$$

Therefore the overall gain is expressed as

$$A_v = \left(\frac{g_{m1}}{g_{m3}}\right)\left(\frac{g_{m6}+g_{m8}}{2} R\right) \tag{19}$$

Where R is denoted by [27],

$$R = (g_{m7} r_{ds7} r_{ds6})(g_{m12} r_{ds12} r_{ds11})$$

In the formulation, if N is the number of finger than, $Nd = int(N+1)/2$. The outline of the gain constraint modeled in GP is as follows:

```
OpenGainConstr = [
```

The drain to source resistance is formulated using equation obtained from the foundry of *TSMC 0.18um* technology.

```
(Using equation from foundry for intrinsic Rd)<=rds6;
```

Similarly, other drain to source resistances are formulated using intrinsic foundry equation. Here, *tg* is a temporary GP variable used to store intermediate values. The gain constraint is modeled as,

```
((2*kp*W(6)/L(6)*I6)^0.5)/gm6==1;

((2*kn*W(1)/L(1)*I1)^0.5)/gm1==1;
.........

(gm7*rds7*rds6)==tg(1);
(gm12*rds12*rds11)==tg(2);
(tg(1)+tg(2))<=tg(3);
(tg(1)*tg(2)/tg(3))==Rii;

(gm1/2/gm4)==tg(4);
(gm6+gm8)<=tg(5);
(tg(4)*tg(5)*Rii)==Av;
 Avmin<=Av;
 Av<=Avmax;
];
```

4.3.2. Unity Gain Bandwidth Constraints

The dominant pole of this cascode two stage amplifier is located at the output node and can be written by the following equation [27],

$$pole_1 = \frac{1}{C_{out}R} \tag{20}$$

where C_{out} can be written as the sum of the load capacitance, the intrinsic drain to body capacitances of M7 and M12 and the interconnect capacitance,

$$C_{out} = C_{load} + C_{db7} + C_{db12} + C_{int1}$$

The interconnect capacitance C_{int1} is modeled by making a minimum sized floorplan shown in Figure 34.The transistor are placed keeping matching in mind and the routing is done trying to minimize the Manhattan distance. The thick black line is metal 2 and the

grey line corresponds to metal 1 in the floorplan. In the figure, *d* is the minimum allowable distance between two transistors and *WR2* is the width of the resistors *R1* and *R2*.

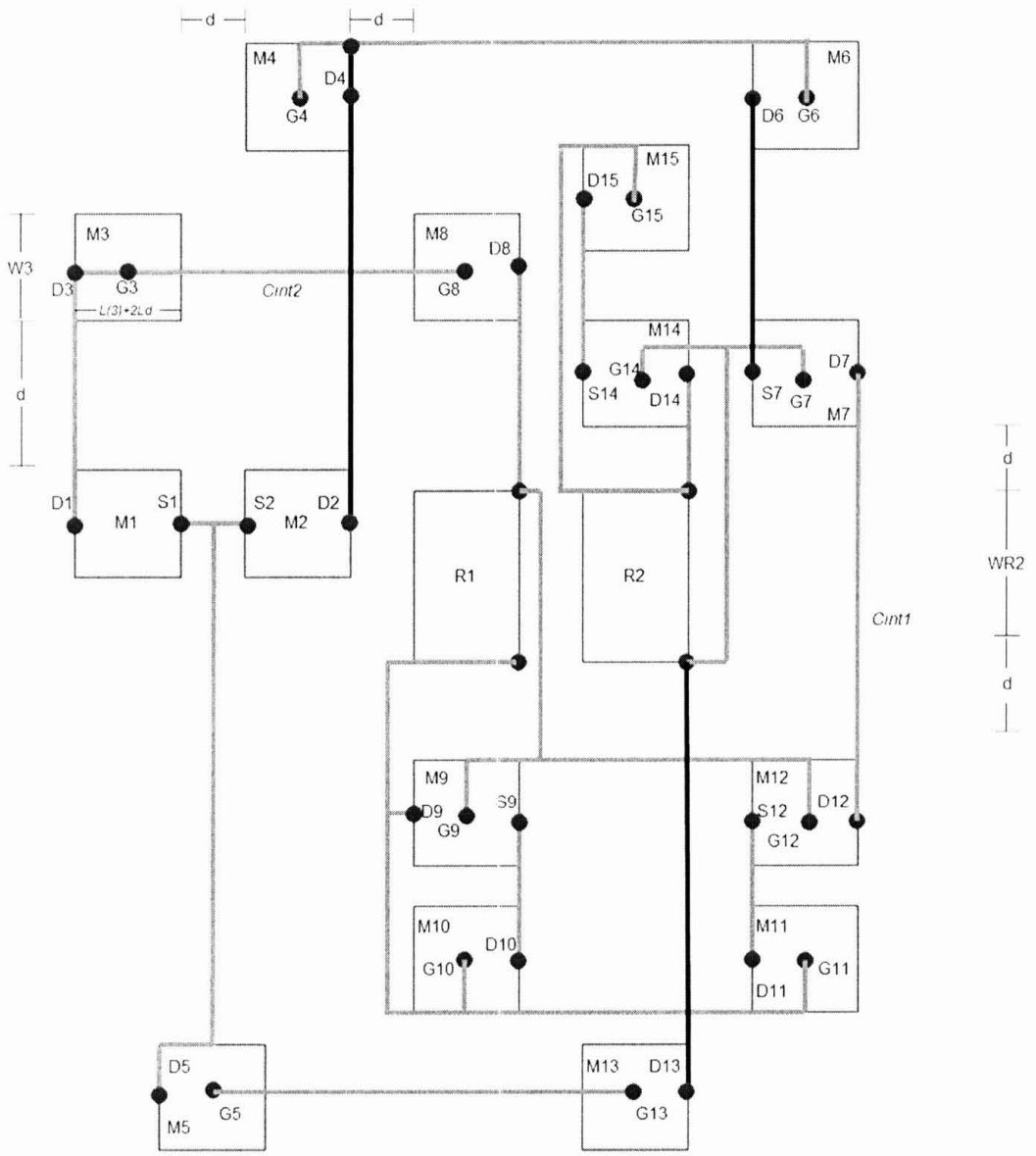


Figure 34: Floorplan for the Cascode Two Stage Amplifier.

The unity gain bandwidth can be written as [27] which is the multiplication of the gain and the dominant pole,

$$GB = \frac{g_{m1}(g_{m6}+g_{m8})k}{2g_{m4}C_{out}} \tag{21}$$

where k is given as

$$k = \frac{S_6}{S_4} \text{ and}$$

$$S = \frac{W}{L}$$

This is modeled as,

```
GBconstr= [

    Rdb7/(410*L(7)/W(7)/Nf/2)==1;
    Cdb7/(21.6/Rdb7)/1e-15==1;

    Rdb12/(2570*L(12)/W(12)/Nf/2)==1;
    Cdb12/(159/Rdb12)/1e-15==1;

(((W(7)/2+W(12)/2)*m1ActAreaCap+(c+WR2+d)*m1FieldAreaCap)*MetalWidth*1e6
)<=Cint1;

    (Cload+Cdb7+Cdb12+Cint1)<=Cout;

    (gm6+gm8)<=tempB1;
    (gm1*tempB1*k/2/gm4/Cout)==tempB2;
    GB<=tempB2;

];
```

4.3.3. Pole Constraints

In this two stage cascode amplifier, the dominant pole is located at the output node and its equation is shown in the previous page. The other non-dominant pole is located at the gate of M3 & M8 and can be written from [27] as the sum of the gate to source capacitances and the interconnect capacitances,

$$pole_3 = \frac{-g_{m3}}{C_{gs3}+C_{gs8}+C_{db3}+C_{ab1}+C_{int2}} \tag{22}$$

It is made equal to at least 10 times the required bandwidth to ensure stability.

The GP formulation is shown below. *tp1* is a temporary GP variable.

```
poleConstr=[

      (Using equation from foundry for intrinsic Cgs)<= Cgs3;
      (Using equation from foundry for intrinsic Cgs)<= Cgs8;
      (Using equation from foundry for intrinsic Cdb)<= Cdb1;
      (Using equation from foundry for intrinsic Cdb)<= Cdb3;
```

The interconnect capacitance is modeled using the floorplan shown before.

```
(((W(1)/2+W(3)/2+Width+Width/2)*m1ActAreaCap+(d+d+Width+d)*m1FieldAreaCa
p)*MetalWidth*1e6)<=Cint2;

    (Cdb1+Cdb3+Cgs3+Cgs8+Cint2)<=tp1(9);
    (gm3/tp1(9))==pole3;
    (10*GB)<=pole3;

];
```

4.3.4. Input Common Mode Range

The input common mode equations are the same as the one in n-input two stage amplifier, which are taken from [27]

$$Positive\ ICMR = V_{DD} - \sqrt{\frac{I_5}{\beta_3}} - |V_{t,max}| + |V_{t,min}| \tag{23}$$

$$Negative\ ICMR = V_{SS} + \sqrt{\frac{I_5}{\beta_1}} + |V_{t,max}| + V_{DS5,sat} \tag{24}$$

Using these equations, we model the ICMR constraint

```
ICMRconstr = [
    (kp*W(3)/L(3))/Beta3==1;
    ((I5/Beta3)^0.5)/ti11==1;
    (VicmrMax+ti11+VtMax)<=ti22;
    (Vdd+VtMin)/ti22<=1;

     (kn*W(1)/L(1))/Beta1==1;
     ((I5/Beta1)^0.5)/ti33==1;
     (Vss+ti33+VtMax+Vds5sat)/(VicmrMin)<=1;
     (kn*W(5)/L(5))/Beta5==1;
    ((2*I5/Beta5)^0.5)/Vds5sat==1;
];
```

If we consider M6, M4, M11 and M12 are in saturation and their saturated V_{ds} is 0.5V then we get,

$$V_{ds11} = \sqrt{\frac{2I_{11}}{k_n S_{11}}} \tag{25}$$

Similarly, we can find the drain to source voltage for M6, M7, M12.

4.3.5. GP solution

All the constraints are passed to GP solver and area is minimized as our objective constraint.

```
constr = [
                SymMatchConstr;
                DevSizeConstr;
                CurConstraint;
                outVoltageConstr;
                ICMRconstr;
                poleConstr;
                OpenGainConstr;
                GBconstr
                AreaConstr;
                constr_x;
    ];
[min_Area solution status] = gpsolve(Area,constr);
assign(solution);
```

The optimal transistor size for the given constraints is given in Table XV.

Table XV: Optimal solution obtained from GP for Two Stage Cascode Amplifier

Variable	*Cacode Two Stage in TSMC 0.18um technology*
L1-L15 (μm)	1.00
W1=W2 (μm)	22.58
W3,W4 (μm)	22.72
W5 (μm)	59.99
W6, W7, W8, W14, W15 (μm)	56.81
W9, W10, W11, W12 (μm)	11.65
W13 (μm)	74.99
R1, R2 (Ω)	2k
Cload (pF)	15
CPU time (s)	1.2852

4.3.6. Pre-Layout Simulation Result

The optimal design for the cascode two stage amplifier obtained from the solution of GP is passed to Cadence for simulation in *TSMC 0.18um* technology. The two input voltage is taken as 1V with 180^0 phase difference. The simulation result is shown in Figure 35.

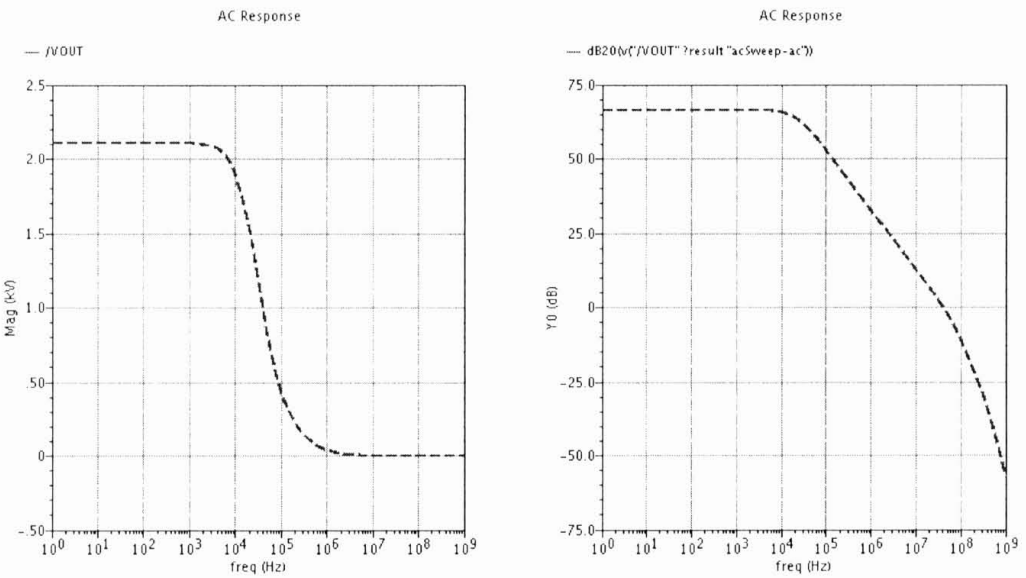


Figure 35: Simulation Result of Cascode Two Stage Amplifier in *TSMC 0.18um* technology

The obtained op amp performance is shown in Table XVI along with the target required specification.

Table XVI: Prelayout simulation result for cascade two stage amplifier.

Variable	Specification	Cascode Two Stage Amplifier With Interconnect Parasitics
Gain	≥ 65 dB	66.483
UGF	≥ 10MHz	41.027
Phase Margin	$\geq 60^0$	60.856
CMRR (dB)	≥ 80 dB	84.08
Power	≤10mW	2.25

The simulation result shows that the optimal design returned from GP has satisfied all performance requirements. Compared to both two stage n-input and p-input amplifier which were designed for a low load capacitance 1pF load with a unity gain frequency of 19.19 MHz for the n-input and 11.42 MHz for the p-input two stage op amp in *TSMC 90nm* technology, this two stage cascade amplifier supports a much higher load (15pF) and still provides a very high unity gain frequency of 41.02 MHz and a phase margin more than 60^0 (60.85^0). Although the achieved gain 66.483 dB is lower than that achieved for the two stage P input Op Amp (PAMP) which was 69.90 dB and N input Op Amp (NAMP) which is 69.57 dB shown in section 1 and 2, it is still more than the targeted required gain which is 65dB and is compensated by the high unity gain bandwidth. So the design obtained from GP passes the pre-layout simulation performance requirements.

4.3.7. Post Layout Simulation

After the successful pre-layout simulation, the next step is to verify using the post-layout simulation. Since this amplifier is designed in *TSMC 0.18um* technology, DIVA extraction tool is used to extract the layout and do the post-layout. The raw schematic is shown in Figure 36 and the corresponding test bench is shown in Figure 37.

The next step is to generate the layout using Layout-XL and then manually place the transistors and finally use Virtuoso Chip Assembly and Router to route the interconnections. The obtained layout view is shown in Figure 58.

Figure 58: Cadence XL layout for Cascode First Stage Amplifier in 0.18µm CMOS technology

The layout is extracted using the DIVA. This extracted view is simulated using the test bench shown in Figure 39 and the obtained result is shown in Figure 59. In the figure, the red-dashed is the Prelayout simulation result and the magenta-solid line is the post layout simulation result. From the figure, it is observed that the pre and post layout simulation gives almost identical result.

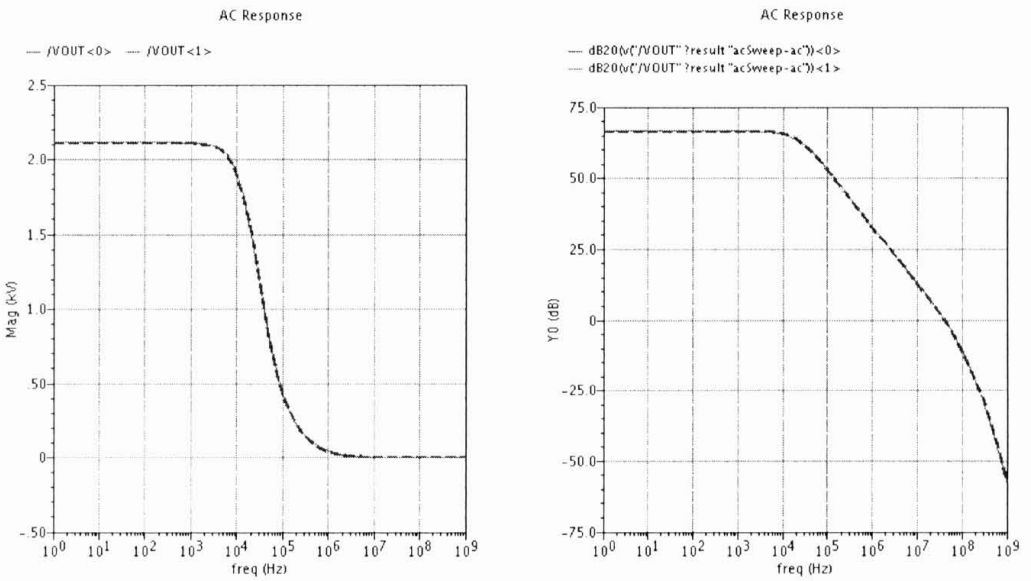


Figure 39: Post layout simulation result for Cascode Two Stage Amplifier in Cadence *TSMC 0.18um* technology. (Magenta-Solid: Post Layout Simulation, Red-Dashed-Prelayout Simulation)

The comparison between the pre and post layout simulation is given in Table XVII.

Table XVII: Pre and Post Layout Simulation Result for Cascode Two Stage Amplifier

Variable	Specification	*TSMC 0.18um*	
		Pre-layout Simulation.	***Post Layout simulation***
Gain	≥ 65 dB	66.483	66.501
UGF	≥ 5MHz	41.027	41.049
BW (3dB)	UnSpecified	198214.68	198088.26
Phase Margin	$\geq 60^0$	60.856	60.906

From Table XVII, it is observed that the pre and post layout simulation results give very consistent results and both satisfy the target performance requirement.

4.4. Summary

In this chapter, three operational amplifier topologies are synthesized with the proposed methodology and tested in two CMOS technologies, *TSMC 0.18um* and *TSMC 90nm* technology. The results obtained from pre and post layout simulation for all three topologies indicates that the optimum design found by proposed synthesis methodology has exceeded all the performance requirements. Furthermore, the simulation results for the designs considering all parasitics proved to be significantly better compared to the designs without parasitics for the two stage n and p input channel op amps. Also, in all cases the synthesis using GP took a time less than 2 seconds, which is remarkably faster than any traditional approaches that use some evolutionary algorithms along with commercial simulators for sizing and off-the shelf layout generator and extractor to incorporate layout effects.

In the next chapter, two high speed analog comparators are synthesized and tested with the proposed methodology and the result is shown.

.

5. High Speed Comparator Design

In this chapter, two analog comparator structures are synthesized in the proposed geometric programming based parasitic aware method. The first structure is based on the two stage amplifier and the second structure is called the differential pair structure that was used in the case study in chapter 2.

5.1. Two Stage Open Loop Comparator

Because the two stage p or n channel input op amp can have a high gain, the op amp can be used as a comparator. The compensating capacitance which is used to ensure stability is not needed in the comparator design because it then gives highest operational bandwidth [27]. The structure is shown in Figure 40.

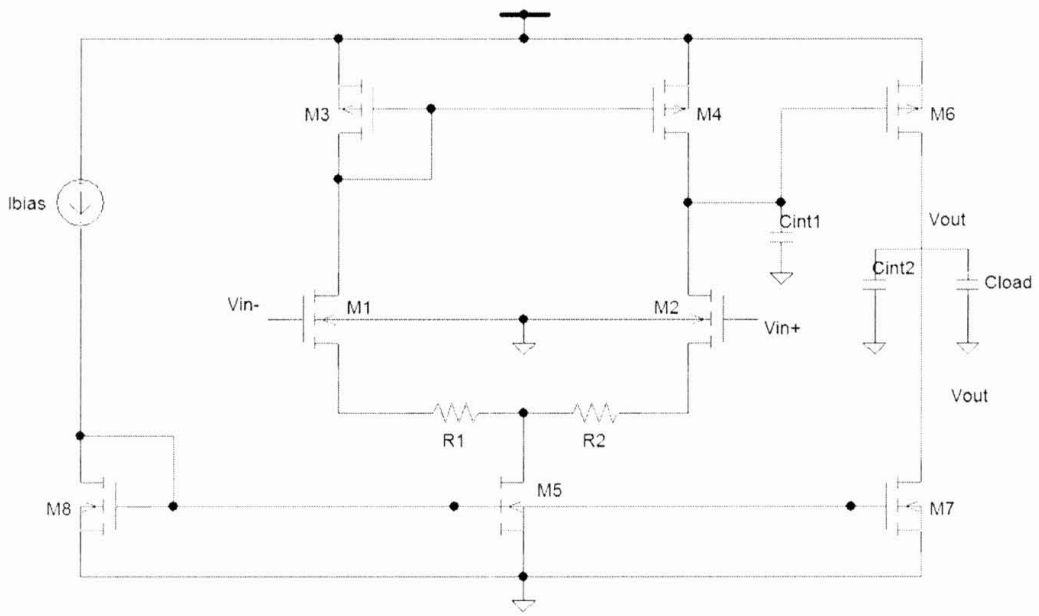


Figure 40: Two Stage Open Loop Comparator [27].

The comparator is designed to optimize for area for the constraints below:

minimize *Area*

while subject to

min input difference > = 10mV

propagation delay < = 50ns

5.1.1. Min Input Difference Constraint

The open loop gain determines the resolution of the comparator which a target specification of the comparator. The higher the gain, the smaller the input different needed between the two differential inputs to compare and get a correct compared result.

The open loop gain is given in [27] as:

$$A_{v(O)} = \left(\frac{g_{m1}}{g_{ds2}+g_{ds4}}\right)\left(\frac{g_{m6}}{g_{ds6}+g_{ds7}}\right) \tag{26}$$

Also, if the minimum resolution of the comparator is

$$V_{in(min)} = V_{IP} - V_{IN}$$

then, then relation with the open loop gain is given by,

$$A_{v(0)} = \frac{V_{OH}-V_{OL}}{V_{in(min)}} \tag{27}$$

In the design, the resolution $V_{in(min)}$ is taken to be 10mV.

The outline of GP formulation is given shown below. Here *tz* is used as temporary GP variable inside GP formulation.

```
Av0=(Voh-Vol)/VinMin;
GainConstr=[
```

The drain to source resistance is formulated using equation obtained from the foundry of *TSMC 0.18um* technology.

```
    (Using equation from foundry for intrinsic Rd)<=rds2;
    rds2*gds2==1;
    ..........

    (2*kp*W(6)/L(6)*I6)^0.5==gm6;
    (gds2+gds4)<=tz(17);
    (gds6+gds7)<=tz(18);


     (Av0*tz(17)*tz(18))/(gm6)==gm1;
     (gm1^2*L(1))/(W(1)*kn*I5)==1;   %2*I1=I5

];
```

5.1.2. Propagation Delay Constraint

The propagation delay is the time it takes for the comparator to respond to an input change and is given as a design specification. This is related to the poles of the comparator.

The comparator has two poles which are considered during the design. The first pole is located at the output of the first stage and the second pole is located at the output of the second stage. The pole equations are taken from [27] which are,

$$pole_1 = \frac{(g_{ds2}+g_{ds4})}{C_1} \tag{28}$$

and

$$pole_2 = \frac{(g_{ds6}+g_{ds7})}{C_{out}} \tag{29}$$

where the total capacitance at the output of first stage, C_1 is given in terms of the parasitic intrinsic capacitances and interconnect capacitance as

$$C_1 = C_{gd2} + C_{gd4} + C_{gs6} + C_{db2} + C_{db4} + C_{int1}$$

and the total capacitance at the output of the second stage is

$$C_{out} = C_{load} + C_{db6} + C_{db7} + C_{int2}$$

The propagation delay is related to the pole of the comparator by the following equation in [27],

$$pole = \frac{1}{t_p\sqrt{mk}} \tag{30}$$

Where m =1 if the two poles are considered equal for simplicity and k is defined as the ratio of the given input difference to the minimum required input difference, that is taken as 10 in the design.

```
PoleConstr=[
     (pole1*tprop*(m*k)^0.5)<=1;
     pole2==pole1;
];
```

The current through the output branch, I_6 is given in terms of the $pole_2$ as [27]

$$I_6 = \frac{pole_2 C_{out}}{\lambda_n + \lambda_p} \tag{31}$$

The sink current through the input branch I_5 is given as [27]

$$I_5 = I_6 \frac{2C_1}{C_{out}} \tag{32}$$

The outline of GP model is as follows:

```
Vsd6=Vdd-Voh;
Vds7=Vol-Vss;

Vsg3=Vdd-VicmPlus+Vtn;

CurConstraint = [

    (0.5*kp*W(6)/L(6)*(Vsd6)^2)==I6;

    (0.5*kn*W(7)/L(7)*(Vds7)^2)==I7;
    ... ... ...

    (Using equation from foundry for intrinsic Cgd)<=Cgd2;
    ... ... ...
```

The other gate to drain capacitances are formulated in similar fashion.

```
(Using equation from foundry for intrinsic Cgs)<=Cgs6;
```

The intrinsic drain to body capacitance, C_{db} is modeled using equations of intrinsic device capacitance. The interconnect capacitance C_{int1} and C_{int2} at the output node of the first and second stage are modeled using the floor plan which is similar to what shown in two stage n-input channel op amp in Figure 20. The outline of formulation is shown below:

```
    (Using equation from foundry for intrinsic Cdb)<=Cdb2;

((W(2)*m1ActAreaCap/2+d*m1FieldAreaCap+W(4)*m1ActAreaCap/2+d*m1FieldArea
Cap+Width*m1ActAreaCap/2)*MetalWidth*1e6)/(Cint1)<=1;

     (Using equation from foundry for intrinsic Cdb)<=Cdb4;
     (Using equation from foundry for intrinsic Cdb)<=Cdb6;
```

```
((W(7)*m1ActAreaCap/2+d*m1FieldAreaCap+W(6)*m1ActAreaCap/2)*MetalWidth*1
e6)<=Cint2;

    (Using equation from foundry for intrinsic Cdb)<=Cdb7;

      ... ... ...
```

Finally all the capacitances are added to find the total output capacitance of the first stage *C1* and a constraint is put between I5 and I7 following the equation shown above.

```
(Cgd2+Cgd4+Cgs6+Cdb2+Cdb4+Cint1)<=C1;

(Cload+Cdb6+Cdb7+Cint2)<=C2;

(pole2*C2/(lamdaN+lamdaP))==I7;
I6==I7;
(I5*C2)/(2*I7*C1)==1;
];
```

5.1.3. Input Common Mode Range Constraint

The input common mode range is governed by, [27]

$$V_{ds5,sat} = V_{icm,minus} - V_{gs1} - V_{ss} \tag{33}$$

Here, V_{ss} is taken as ground. The positive ICMR controls the V_{sg3} by [27], which is the highest input voltage the amplifier can amplify and is limited by the supply voltage and required gate to source voltage to ensure saturation.

$$V_{sg3} = V_{dd} - V_{icm,plus} + V_t$$

The gate to source voltage of M1 is found by,

$$V_{gs1} = V_t + \left(\sqrt{\frac{2I_1}{k_n {}^{W1}/_{L1}}}\right)$$

Here, the input positive and negative common mode is given as specification as 1.62V and 0.36V. All these are formulated as below,

```
Vsg3=Vdd-VicmPlus+Vtn;
ICMRconstr = [
    (kp*(Vsg3-abs(Vtp)^2)*W(3)/L(3))==I5; %2*I3=I5
    (I5/kn/(W(1)/L(1)))^0.5==tp(9);
    (tp(9)+abs(Vtp))<=Vgs1;
    (VicmMinus+Vgs1)<=(Vds5sat);
    (kn*Vds5sat^2*W(5)/L(5)/2/I5)==1;
    ];
```

where, $k_p=\mu_p Cox$, $k_n = \mu_n Cox$ respectively.

The other GP constraints like the minimum and maximum device size, transistor symmetry etc. are added in the same manner as the two stage n-input op amp design.

5.1.4. GP solution

Finally, the solution is found by calling the GP solver

```
constr = [
        SymMatchConstr;
        DevSizeConstr;
        CurConstraint;
        ICMRconstr;
        PoleConstr;
        GainConstr;
        floor_constr;
        AreaConstr;
        constr_x;
    ];
```

The optimal values obtained from GP are listed in Table XVIII.

Table XVIII: Optimal design for Two Stage Comparator

Var.	Two Stage Open Loop Comparator With Interconnect Parasitics
	TSMC 0.18μm
L1-L8 (μm)	1.00
W1=W2 (μm)	6.28
W3,W4 (μm)	3.94
W5 (μm)	0.579
W6 (μm)	7.66
W7 (μm)	1.57
W8 (μm)	0.22
Cload (pF)	1
R1 (Ω)	1.86
R2 (Ω)	1.86
Cint1 (fF)	0.012
Cint2 (fF)	0.511
Ibias (μA)	30u
CPU time (s)	0.972

5.1.5. Pre-Layout Simulation Result

The comparator is tested in Cadence in *TSMC 0.18um* technology for different cases. In the first case Vin^+ is taken more than 0.3V than Vin^- and the result is observed. In the second case, Vin^+ is taken to be less than 0.1V than Vin^-. In case 3, the minimum input difference of 10mV is tested. In case 4, a changing piecewise linear voltage is applied to the two inputs to see the propagation delay of the comparator.

Case 1:

For this case, Vin^+ is taken as 0.7V and Vin^- as 0.4V, so the applied input difference is 0.3V. Figure 41 shows the response of transient simulation for this case. Although, it does not reach the maximum supplied 1.8V because there is a voltage drop across MOSFET, but still it is high enough (1.35 V) to be decoded as logic 1.

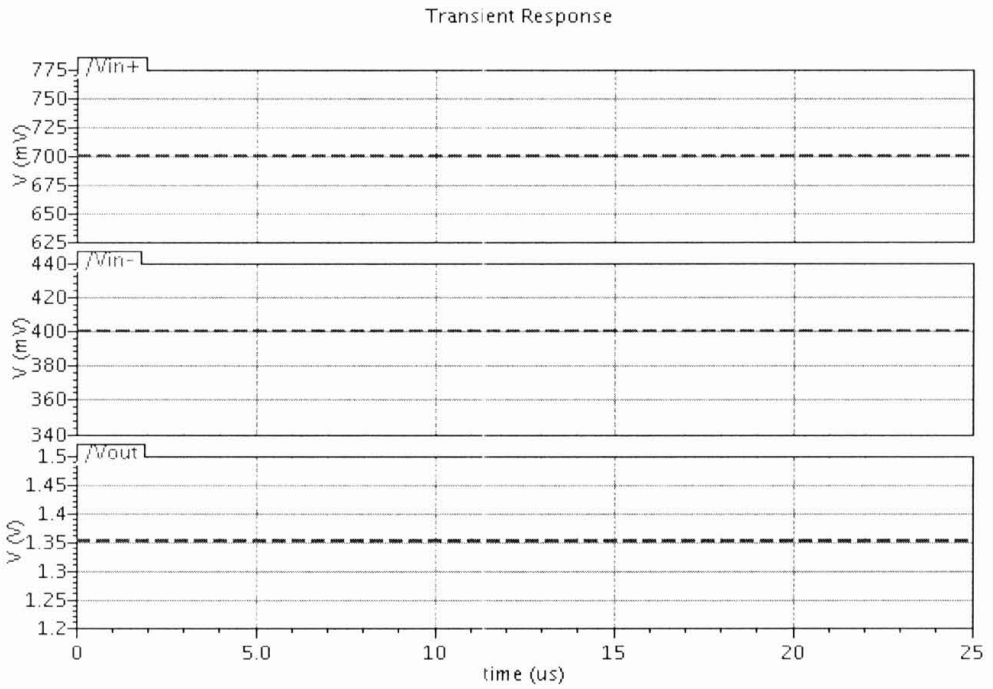


Figure 41: Transient Simulation Result for two stage open loop comparator with Vin+ = 0.7V and Vin- = 0.4V in *TSMC 0.18um* technology

Case 2:

For this case, the positive input Vin^+ is taken as 0.3V which is lower than the applied negative input Vin^- = 0.4V, so the input difference is -0.1V. As Vin^+ is less than Vin^-, the expected comparator output should go to zero. Figure 42 contains the response of transient simulation for this case. From the curve, the final settling voltage is found to be 10.42 ~ 10.47 nV which is decoded as logic 0.

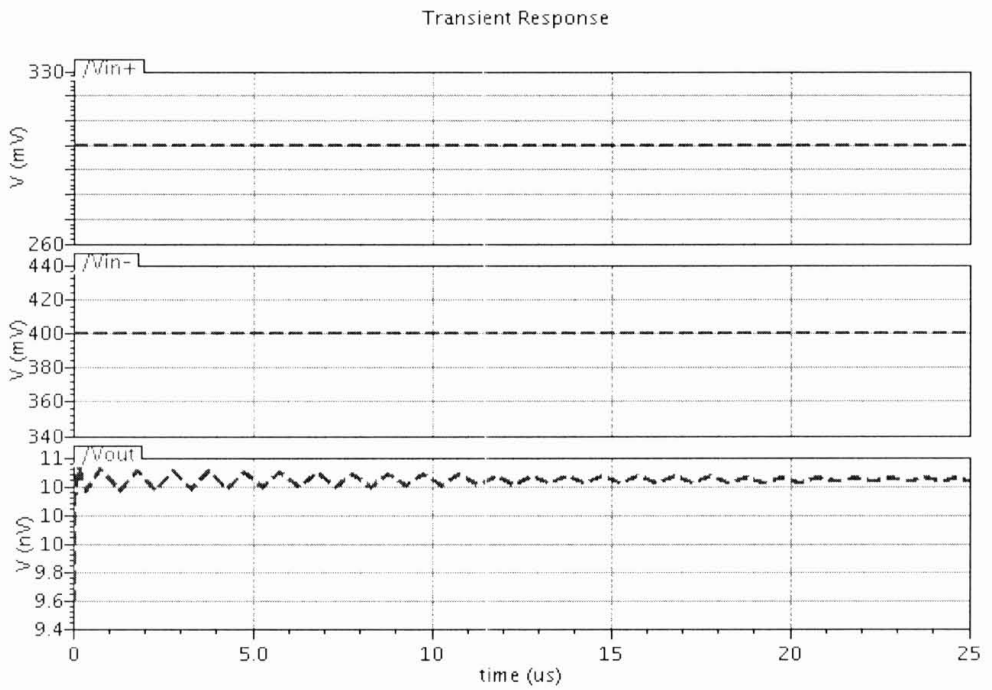


Figure 42: Transient Simulation Result for two stage open loop comparator with Vin+ = 0.3V and Vin- = 0.4V in *TSMC 0.18um* technology

Case 3:

In this case, the comparator is tested with a very small input difference which is 10mV by taking Vin^+ as 110 mV and Vin^- as 100 mV. Figure 43 shows the response of transient simulation for this case. The settling voltage is found to be 1.29 V which is logic 1. So the comparator can compare very small input difference and still can decode to correct logic.

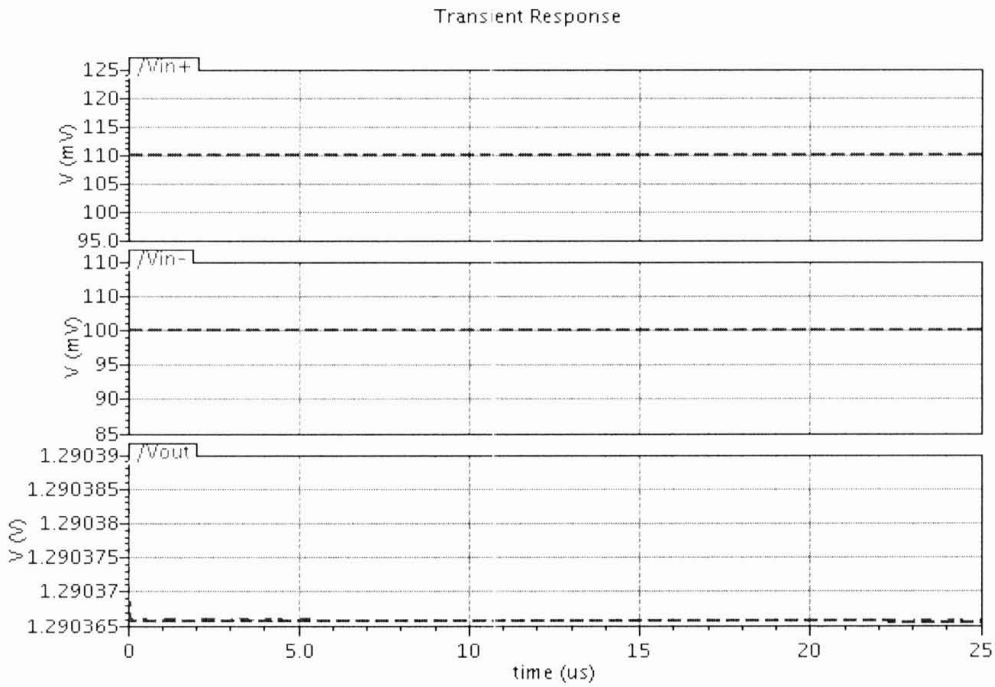


Figure 43: Transient Simulation Result for two stage open loop comparator with Vin+ = 110 mV and Vin- = 100 mV in *TSMC 0.18um* technology

Case 4:

In this case, the propagation delay, which is the time it takes for the output to respond to an input change, is checked. Two piece wise linear voltage sources are used as inputs with initially zero voltage. The positive input goes to 0.6V at 100ns and goes down to 0V at 200ns. The negative input rises to 0.6V at 200ns and falls to 0V at 300ns. The simulation result is shown in Figure 44.

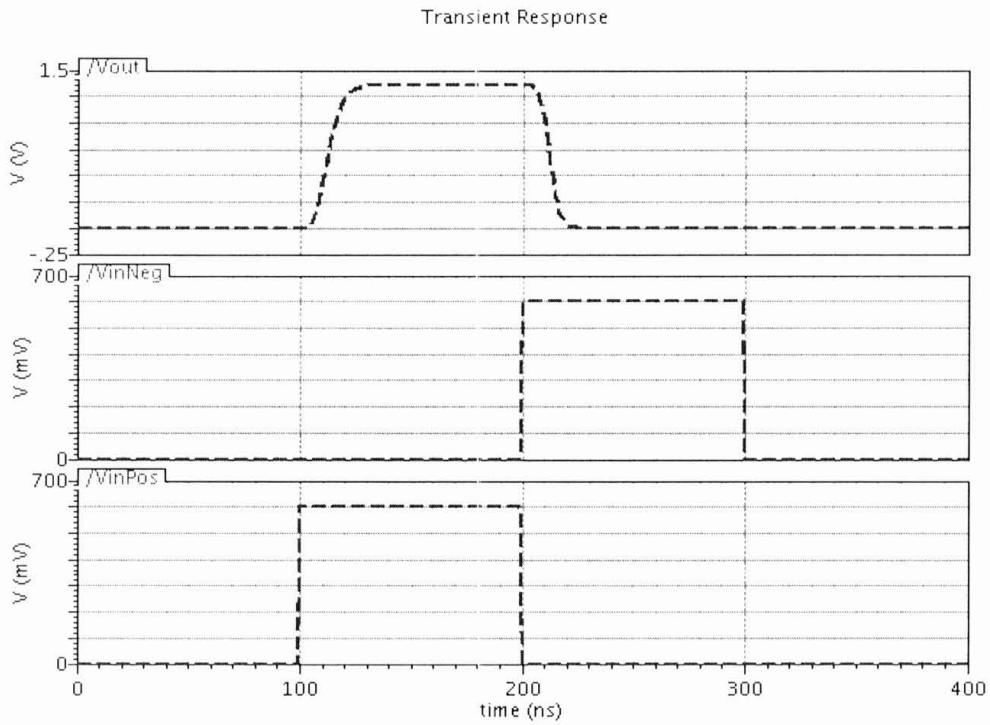


Figure 44: Transient Simulation Result for two stage open loop comparator with PWL sources

The high settling voltage of output is 1.3572V. The rise time is found to be 22ns and the fall time is found to be 20ns. The propagation delay of the comparator depends on the difference of input voltage [27] and the higher the input difference, the quicker is the

response from the comparator. The target propagation delay of the optimization was less than 50ns; therefore the design satisfies the propagation delay requirement.

All these cases show that the design comparator satisfies the requirement specification. So the optimal solution found from the proposed methodology is capable to design a two stage open loop comparator.

5.1.5. Post-Layout Simulation Results

After the pre-layout verification, the comparator design is then tested for post-layout simulation. A layout is generated using Layout XL and routed using VCAR in IBM 0.18um technology from the comparator schematic in Figure 44.

Figure 45: Open Loop Comparator post layout

Then the layout is extracted using DIVA and the extracted netlist is used for post layout simulation results. The layout is shown in Figure 46.

Figure 46: Open loop comparator generated layout

In the test case for post layout simulation, the positive input switches from 0V to 0.6V at 150ns and goes back to 0V at 250ns. Similarly, the negative input is made to switch to 0.6V at 300ns and goes back to 0V at 350ns. The transient simulation result is shown in Figure 47.

From the Figure 47, it is observed that when the positive input is more than the negative input, the output goes to logic high. The settling voltage for logic high is found to be 1.5924V. This is not the full supply voltage which is 1.8V as there is a drop of voltage across drain to source of the MOSFET, but is high enough to be considered as logic high.

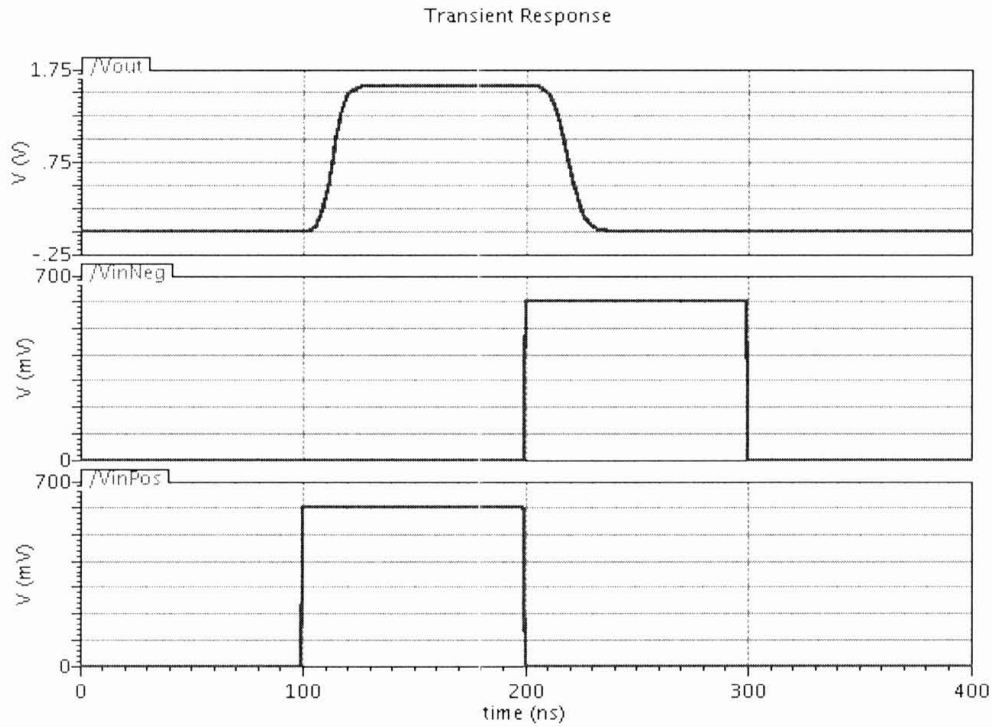


Figure 47: Post layout transient simulation result for Open Loop Comparator

The rise time which is the same as the propagation delay as the output change from one state to another, is found to be 23.9ns. When the negative is more than the positive input, the output drops to logic low. The fall time is found to be 30ns. Both rise and fall time is less than the specified propagation delay which is under 50ns.

In this section, a two stage open loop comparator is designed and tested with the presence of parasitics in *TSMC 0.18um* technology. The simulations show the design of open loop comparator obtained by the proposed parasitic aware synthesis methodology achieves all the performance requirements.

In the next section, a Differential-Pair Comparator is designed and tested with the proposed methodology considering all parasitics in *TSMC 90nm* technology.

5.2. Differential Pair Comparator

The structure and brief operation of Differential Pair Comparator [11] was introduced in chapter 2 which is shown again Figure 48. This dynamic comparator is faster than the gain based comparator designed above and consumes very little power as it is driven by clock signal.

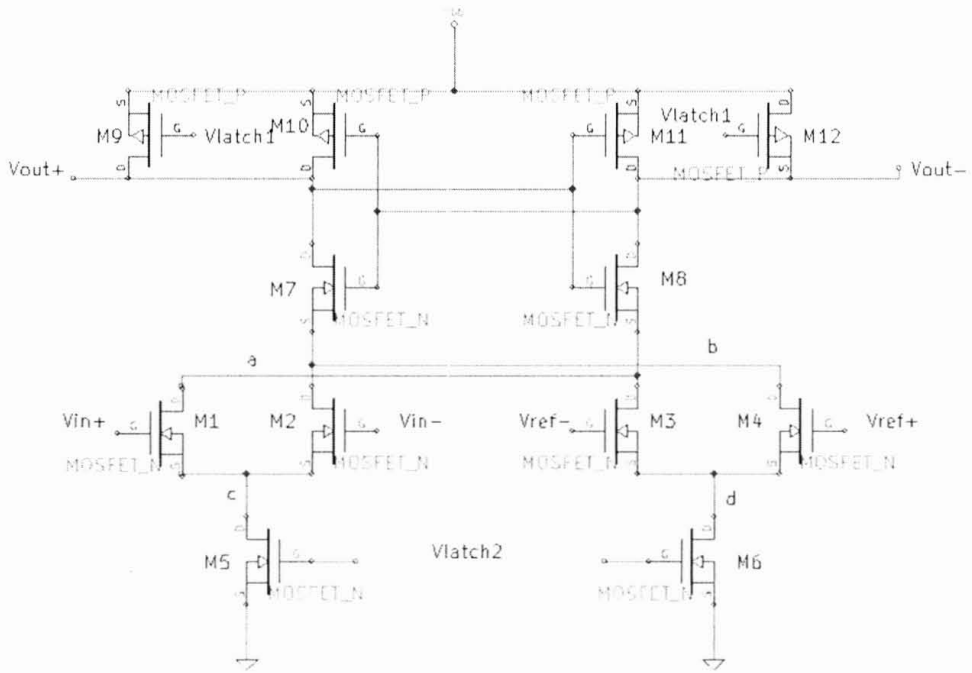


Figure 48: Differential Pair Comparator [10].

This comparator is based on latch configuration [27]. The transistors M10 and M11 work as the latch transistor. The comparator has two modes of operation. When $V_{latch1} = 0V$, the latch is disabled and both M9 and M12 is ON, making both positive and negative output

voltage pre-charge to the supply voltage. When V_{latch1} = 1V, it enables the latch and turns off the transistors M9 and M12. As the gates of M7 & M10, are connected to the negative output V_{out}^-, and the gates of M11 and M8 are connected to the positive output, V_{out}^+, the NMOS transistors M7 and M8 makes a positive feedback path for the latch. As both outputs were pre-charged to supply voltage when V_{latch1} = 0, when V_{latch1} becomes 1, they don't instantly discharge and keep M7 and M8 still ON. Then depending on the voltages of the positive and negative inputs, one branch provides more resistance than another and the positive feedback of the latch makes one output to stay at supply voltage and causes another output to discharge to logic 0. The positive output stays high if the positive input is more than the negative input and vice versa. A separate clock of 0.4V/0V is used instead of using the same V_{latch1} with the same phase to make sure that the two tail transistors M5 and M6 works in saturation region [29] and provide a constant current for the latch. The comparator is designed for *TSMC 90nm* technology. All parasitics are considered during formulation of the sizing constraints as they are proved to offer better performance in pre and post layout verification from previous examples.

The design formulation of the comparator consists of the following GP constraints:

5.2.1. Propagation Delay Constraint

The Differential Pair Comparator discussed is a regenerative comparator based on latch. The latch time constant can be expressed as [27],

$$\tau_l = \frac{C_{out}}{g_m} \tag{34}$$

Where $C_{out,p}$ and $C_{out,n}$ are the total capacitances at the positive and negative output nodes. Their expressions can be written as,

$$C_{out,p} = C_{load,p} + C_{db7} + C_{db10} + C_{gs11} + C_{gs8} + C_{intp}$$
$$C_{out,n} = C_{load,n} + C_{db8} + C_{db11} + C_{gs10} + C_{gs7} + C_{intn}$$

where C_{intp} and C_{intn} are the interconnect capacitances that can be modeled in symbolic form using the minimum size floorplan. The propagation delay of the latch, t_{prop} which is a given as a target specification can be written from [27] in terms of the final low and high output voltages, V_{oh} and V_{ol} as,

$$t_{prop} = \tau_l ln\left(\frac{V_{oh}-V_{ol}}{2\Delta V_{in}}\right) \quad (35)$$

where ΔV_{in} is the difference between the two latch output voltage before the latch is enabled, which is always less than V_{oh} - V_{ol} . The target maximum allowable propagation delay is taken as 1ns.

These equations are modeled as

```
uln=log((Voh-Vol)/2/DelVi);
PropDelayConstr = [

    (2*kn*W(7)/L(7)*Id)^0.5==gm7;
    (2*kn*W(8)/L(8)*Id)^0.5==gm8;

    (Using equation from foundry for intrinsic Cgs)<=Cgs7;
```

Similarly C_{gs8}, C_{gs10}, C_{gs11} are modeled. The drain to body intrinsic capacitance of Cdb7 is modeled as,

```
    (Using equation from foundry for intrinsic Cdb)<=Cdb7;
```

Similarly, the other drain to body capacitances are modeled using the equations for calculating intrinsic device capacitance as shown before. The two interconnect capacitance at the positive and negative output nodes are modeled using the floorplan shown in Figure 49. The thick black line for routing is taken as metal 2 and the grey line is metal 1. The minimum distance between any two transistors is d, and $Width = L+2L_d$, is a variable which is the sum of transistor gate length, which taken as minimum size 0.1um and lateral diffusion of the source and drain of the transistors, L_d.

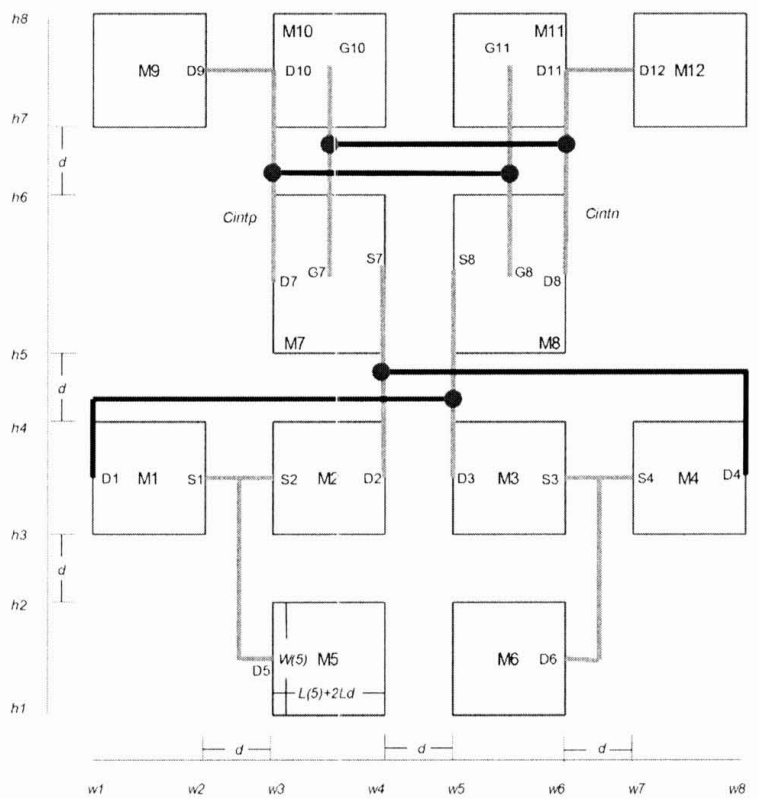


Figure 49: Differential Pair Comparator floorplan used to model interconnect parasitics

The two interconnect capacitances are modeled as below,

```
(((d+W(10)/2+d+W(7)/2+W(11)/2+d+W(8)/2)*m1MaskCap+(Width+d+Width/2)*m2Ma
skCap)*MetalWidth*1e6)<=Cintp;

(((d+W(11)/2+d+W(8)/2+W(10)/2+d+W(7)/2)*m1MaskCap+(Width+d+Width/2)*m2Ma
skCap)*MetalWidth*1e6)<=Cintn;
```

The total capacitances at the two output nodes and the final propagation delay constraint are formulated as follows,

```
(Cloadn+Cdb8+Cdb11+Cdb12+Cgs10+Cgs7+Cintn)<=C8;
(Cloadp+Cdb7+Cdb9+Cdb10+Cgs11+Cgs8+Cintp)<=C7;

(C7/gm7)==Tau7;
(C8/gm8)==Tau8;

Tau7*uln<=tprop;
Tau8*uln<=tprop;
```

5.2.2. Resistance Constraint

When the two clocks are at evaluative phase, the latch is enabled and depending on the resistances of the two branches, the latch decides which output will stay high and which one will go low. As the transistors M1, M2, M3 and M4 works in triode region, the MOSFET on resistance of the two branches can be written as [27],

$$\frac{1}{R_p} = k_n \left[\frac{W_1}{L_1}(v_{in}^{+} - v_t) - \frac{W_3}{L_3}\left(v_{ref}^{-} - v_t\right)\right] \tag{36}$$

and

$$\frac{1}{R_n} = k_n \left[\frac{W_2}{L_2}(v_{in}^{-} - v_t) - \frac{W_4}{L_4}\left(v_{ref}^{+} - v_t\right)\right] \tag{37}$$

So, as W_2 & W_4 and W_1 and W_3 are considered equal, the conductance of the two branches can be written as,

$$G_1 = k_n \frac{W}{L}\left(V_{in} - V_{ref} - 2V_t\right)$$

$$\Rightarrow G_1 = k_n \frac{W}{L}(V_{ov}) \tag{38}$$

where the effective overdrive voltage is

$$V_{ov} = \left(V_{in} - V_{ref} - 2V_t\right)$$

The resistances, which must be equal, for the same applied inputs voltages and reference voltages, in order to ensure proper matching between two branches, are found to be the inverse of this conductance and the resistance is taken under a certain specified value R_{max} to ensure sufficient speed. This is modeled as,

```
gpvar G1 Rest;
ResConstr = [
     kn*(W(1)/L(1))*Vov==G1;
     Rest*G1==1;
     Rest<=Rmax;
];
```

5.2.3. Capacitance Constraints

As analyzed and identified in chapter 2, when doing the sensitivity analysis of the Differential Pair Comparator, a capacitive mismatch between the two output nodes can throw the comparator out of operation. So another constraints is put so that the difference

between the interconnect capacitances, modeled from the floorplan, is lower than a certain specified value. In this case the value is selected to be 0.18fF, which was obtained from the case study in chapter 2.

```
gpvar tc; % temporary variable
CMaxDiff=0.12e-15;
CapConstr=[
    %C8-C7<=CmaxDiff
    C7+CMaxDiff<=tc;
    C8<=tc;
];
```

5.2.4. Other Constraints

Here, the input NMOS W_1 - W_6, is considered equal and the PMOS $W_7 - W_{12}$ are also considered equal. Those are given in symmetric match constraint. All the devices are less than the maximum allowable device size which is given in Device size constraint. Finally Area constraint is defined in terms of circuit area area and GP solver is called to obtain the optimal result.

```
SymMatchConstr = [
      W(1)/ W(2) == 1;
      W(3)/W(4) == 1;
      ... ... ...

      W(7)/W(8) ==1;
      W(9)/W(10)==1;
      ... ... ...
];

DevSizeConstr = [
      W(1)/Wmax <= 1;
... ... ...
    ];
```

5.2.5. GP solution

GP solution is obtained by passing all the constraints to the GP solver which is shown below.

```
constr = [
                SymMatchConstr;
                DevSizeConstr;
                PropDelayConstr;
                ResConstr;
                CapConstr;
                AreaConstr;
                constr_x;
    ];
[min_Area solution status] = gpsolve(Area,constr);
assign(solution);
```

The optimal solution given by GP is listed in Table XIX.

Table XIX: Optimal design for Differential Pair Comparator obtained after synthesis.

Var.	**Differential Pair Comparator**
	TSMC 90nm
L1-L12 (μm)	0.10
W1-W6 (μm)	5.00
W7,W8 (μm)	1.85
W9-W12 (μm)	5.00
CPU time (s)	1.15

5.2.6. Pre-layout Simulation

The design is tested in pre-layout simulation in *TSMC 90nm* technology. Figure 50 shows the transient simulation result when V_{in}^{+} = 0.8V and V_{in}^{-}=0.4V. The two references voltages are taken as 0.8V and 0.6V. As the positive input is more than the negative input

by 0.4V, for the correct comparison, the positive output should stay high at the evaluate phase and the negative output should go low, which is observed in the figure.

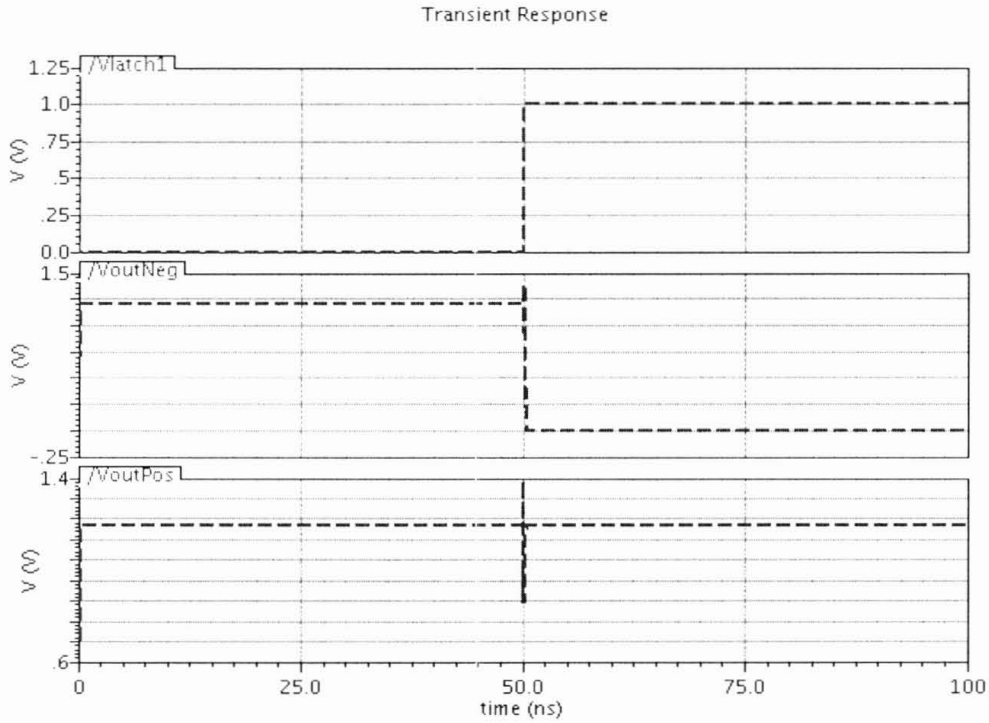


Figure 50: Differential Pair Pre-layout simulation in *TSMC 90nm* for Vin^{+}=0.8V and Vin^{-}=0.4V

The pre-layout simulation result for V_{in}^{+}=0.8V and V_{in}^{-}=0.4V is shown in Table XX.

Table XX: Prelayout simulation result for V_{in}^{+}=0.8V and V_{in}^{-}=0.4V

Variable	Differential Pair Comparator *Vin^{+}=0.8V and Vin^{-}=0.4V*
Propagation Delay (Fall time of Negative Output)	0.435 ns
Positive Output Overshoot	1.39 V
Negative Output Overshoot	1.388 V
Settling Voltage (Positive Output)	1.2 V

For Vin^+=0.3V and Vin^-=0.4V, the expected result is to have the negative output staying high and the positive output should go low which is shown in Figure 51.

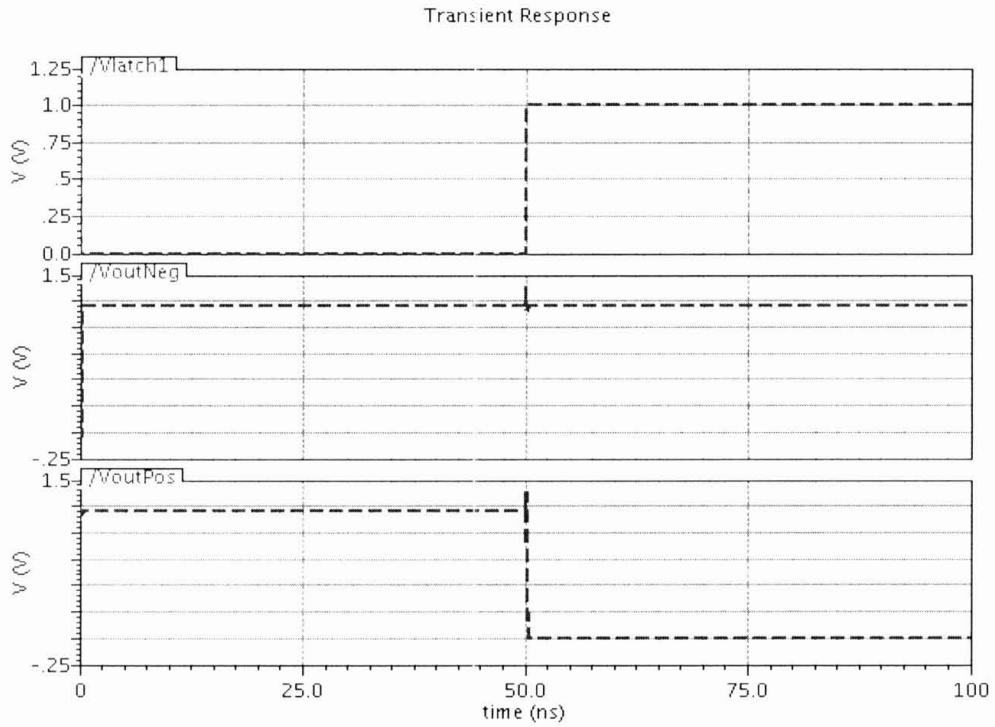


Figure 51: Differential Pair Pre-layout simulation in *TSMC 90nm* for Vin^+=0.3V and Vin^-=0.4V

The pre-layout simulation result for Vin^+=0.3V and Vin^-=0.4V is shown in Table XXI.

Table XXI: Prelayout simulation result for V_{in}^+=0.3V and V_{in}^-=0.4V

Variable	Differential Pair Comparator *Vin^+=0.3V and Vin^-=0.4V*
Propagation Delay (Fall time of Positive Output)	0.48 ns
Positive Output Overshoot	1.388 V
Negative Output Overshoot	1.3877 V
Settling Voltage (Negative Output)	1.2 V

In both cases, the settling voltage is found to be equal to the supply voltage and the propagation delay is found to be less than 1ns which is target maximum allowable propagation delay.

5.2.7. Post-layout simulation

The differential pair comparator structure in Cadence shown in Figure 52 is used to generate layout using layout-XL which is shown in Figure 53. During the layout generation, all the transistors with equal size are placed symmetrically to reduce the layout induced effects.

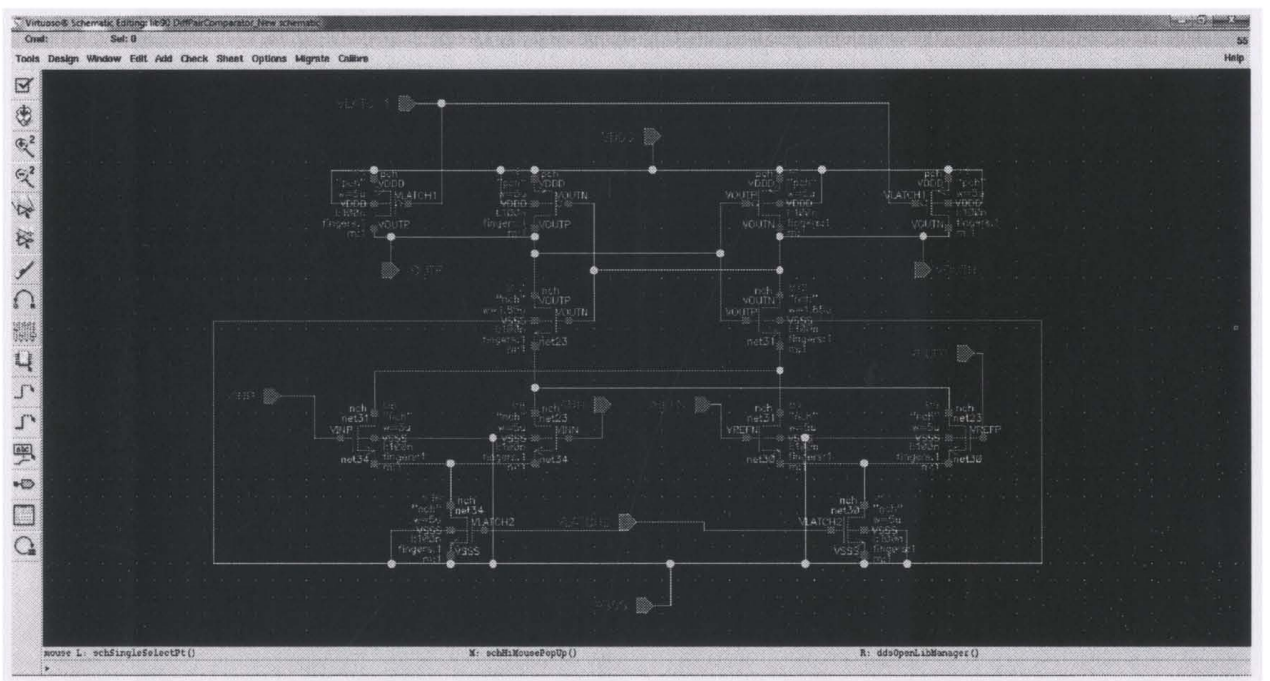


Figure 52: Schematic of raw Differential Pair Comparator in *TSMC 90nm* technology.

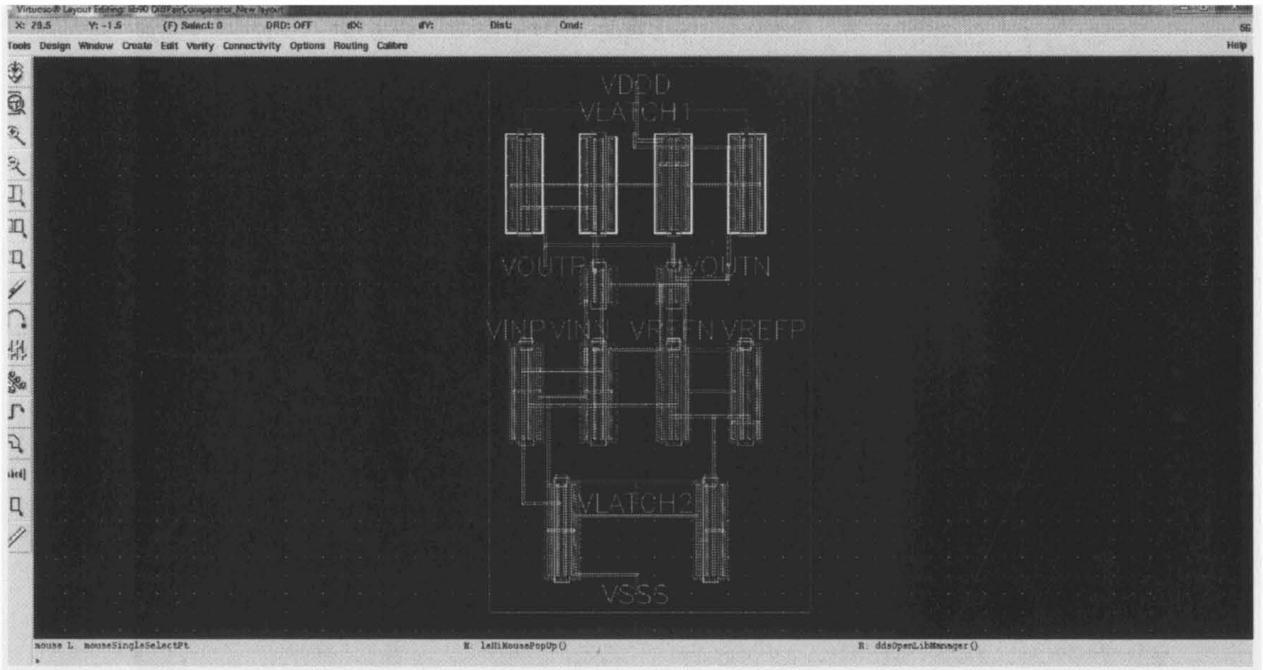


Figure 53: The generated layout from the schematic of differential pair comparator using layout-xl and after routing using VCAR in *TSMC 90nm* technology.

Figure 54 shows the post layout simulation result when the positive input Vin^{+}=0.8V and the negative input Vin^{-}=0.4V. Table XXII contains the other results in post layout simulation for this input condition.

Table XXII: The performance obtained from post-layout simulation in *TSMC 90nm* for Vin^{+}=0.8V and Vin^{-}=0.4V

Variable	Differential Pair Comparator *Vin^{+}=0.8V and Vin^{-}=0.4V*
Propagation Delay (Fall time of Negative Output)	0.642 ns
Positive Output Overshoot	1.41 V
Negative Output Overshoot	1.4018 V
Settling Voltage (Positive Output)	1.2 V

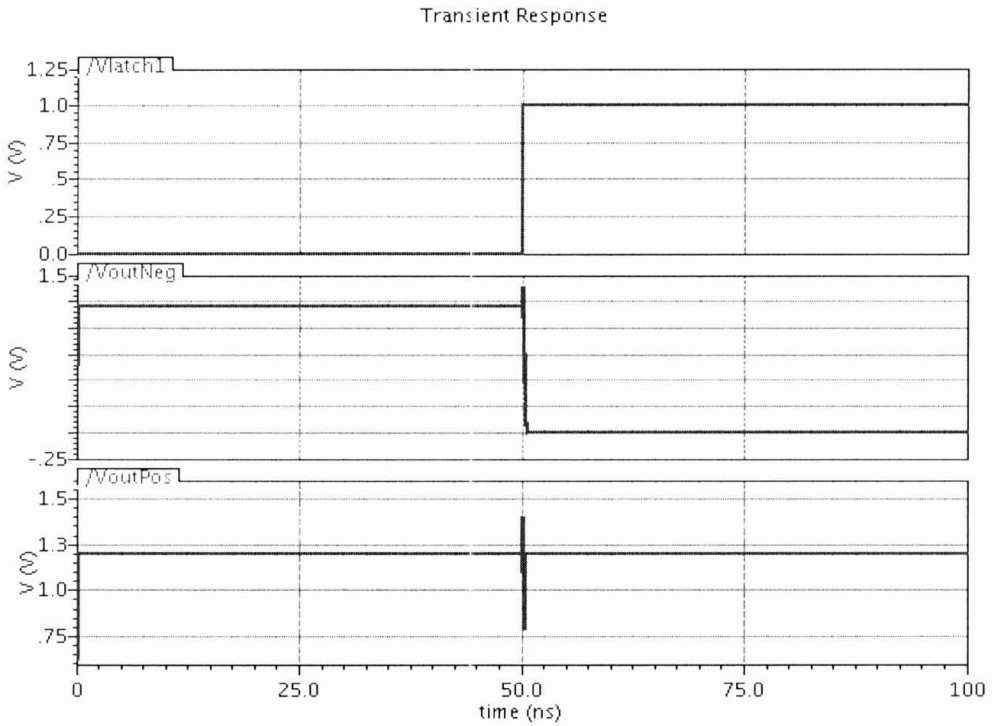


Figure 54: Differential Pair Post-layout simulation in *TSMC 90nm* for Vin^{+}=0.8V and Vin^{-}=0.4V

Figure 55 shows the post layout simulation result when the positive input Vin^{+}=0.3V is less than the negative input Vin^{-}=0.4V.

Table XXIII contains the other results in post layout simulation for this input condition.

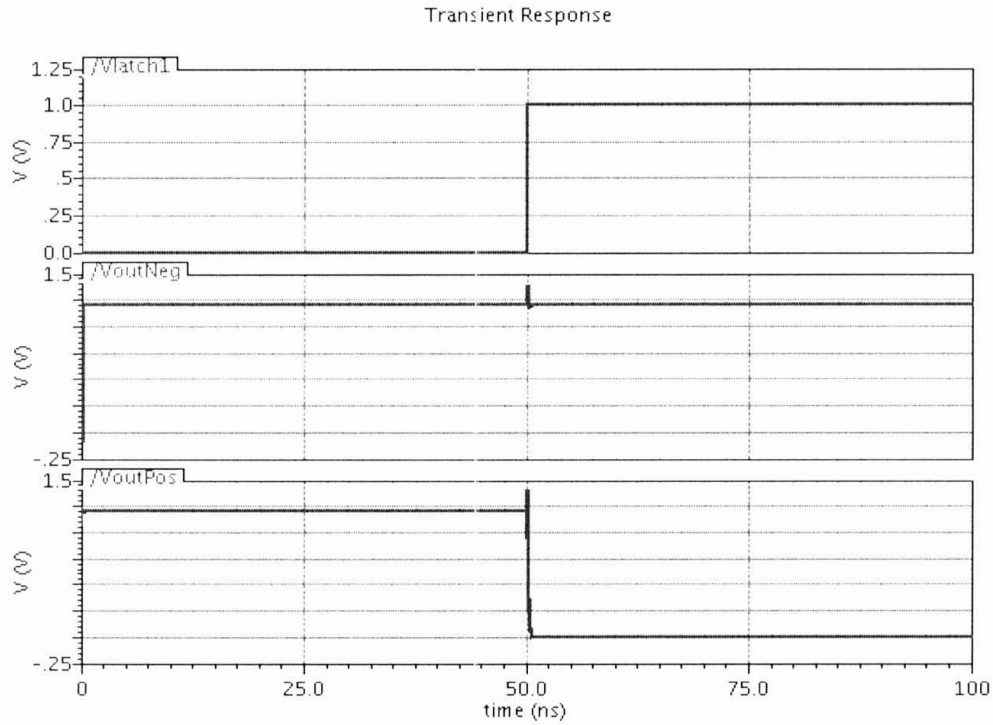


Figure 55: Differential Pair Post-layout simulation in *TSMC 90nm* for Vin^{+}=0.3V and Vin^{-}=0.4V

Table XXIII: The performance obtained from post-layout simulation in *TSMC 90nm* for Vin^{+}=0.3V and Vin^{-}=0.4V

Variable	Differential Pair Comparator *Vin^{+}=0.3V and Vin^{-}=0.4V*
Propagation Delay (Fall time of positive output)	0.524 ns
Positive Output Overshoot	1.4061 V
Negative Output Overshoot	1.4018 V
Settling Voltage (Negative Output)	1.2 V

The post layout simulation for the cases when positive input is higher than negative input and when negative input is more than positive input shows that the comparator performs as expected with lower than the target 1ns maximum propagation delay. The Calibre

extraction result of the layout reveals that the total output capacitance to substrate at the positive node is 1.7521fF and at the negative output node is 1.8786fF. So the difference is 0.1265fF which is less than the value (0.182fF) obtained in sensitivity analysis in chapter 2 that can make the comparator non-operational for the input condition. So, the symmetrical placement of the transistors along with proper sizing using the proposed synthesis approach based on GP, can handle the parasitic mismatch that can arise from layout and operates correctly.

5.3. Summary

In this chapter, two analog comparators are designed with the proposed parasitic-aware methodology by Geometric Programming. They are first tested in pre-layout simulation and later tested in post-layout simulation by layout generation and extraction. Two different technologies are used for the two comparators to show that the approach is not limited to any certain technology. The performance obtained from the simulation confirms that the comparator design obtained by the proposed methodology works as expected in post layout simulation.

In the last two chapters, five analog circuits are designed and tested in the proposed method. This ensures that the proposed methodology is adequate for analog designs that are sensitive to layout mismatch. The synthesis time has been found to be very fast and the designs met all the initial performance requirements that were put into consideration.

In the next chapter, the proposed method is applied to two RF circuits in *TSMC 90nm* technology and their performances are measured.

6. RF Circuit Design

In this chapter, the proposed parasitic aware circuit synthesis methodology is applied to two RF circuits namely a low noise amplifier (LNA) and a voltage controlled oscillator (VCO) and their performance is simulated in *TSMC 90nm* technology. The design of LNA targets several key factors like input impedance match with the 50Ω input resistance for maximum power transfer and a sufficient gain with low noise figure, so that it can overpower the noise at a targeted resonant frequency. Several LNA structures are shown in [30] like the common source LNA with inductive load, common source LNA with resistive feedback, common source LNA inductively degeneration; common gate LNA etc. The common source LNA with inductive load requires a large inductor to match the impedance for the targeted frequency range, causing a large parasitic capacitance that degrades the performance [30]. The common source LNA with resistive feedback suffers from a high noise figure arising from the resistance. The common gate supports a better input match but it has a relatively higher noise figure compared to the cascode common source LNA with inductive degeneration. Because the latter gives an acceptable gain with low noise figure, it is selected as the design structure of the LNA.

6.1. Cascode Common Source LNA with Inductive Degeneration

The cascode common source LNA with inductive degeneration is shown in Figure 56. The target is to attain more than 15dB gain with a noise figure less than 2dB.

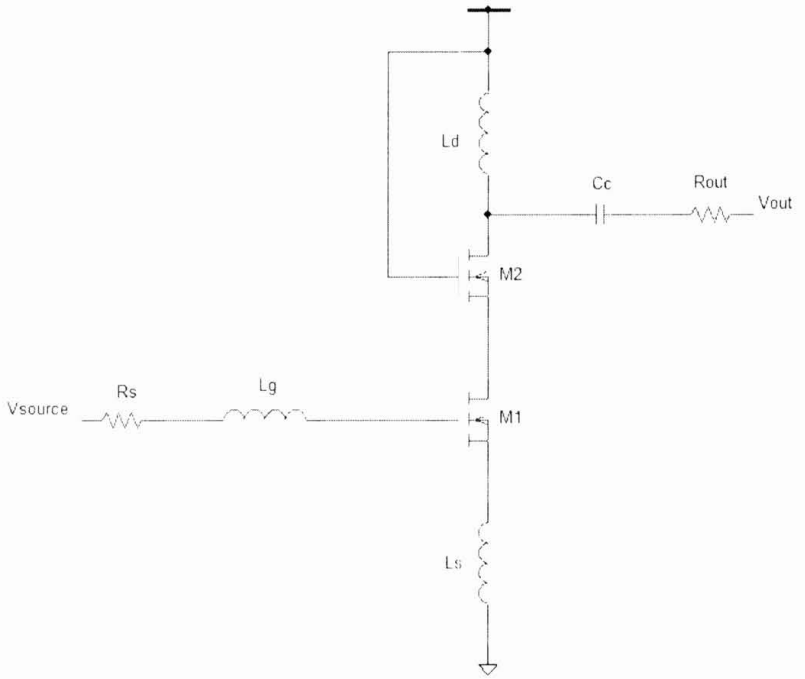


Figure 56: Cascode Common Source LNA with inductive degeneration [30].

The design constraint can be put as follows.

minimize *Area*

subject to:

Gain ≥ 15 dB

Noise Figure, NF ≤ 2 dB

Target frequency, f = 5GHz

The several design constraints of the LNA design are as follows:

6.1.1. Interconnect Constraint

The interconnect resistance R_{int} shown in the floorplan in Figure 57, from the inductor Lg to the gate of M1, influences the input side matching and the noise figure of the LNA. Two separate optimizations are performed for the target specification of the LNA. In the first case, the minimum allowable width of interconnect is taken and R_{int} is calculated. This is used during simulation to observe the effect of using minimum width interconnect on the circuit performance. In the next case, the interconnect resistance is optimized by controlling the maximum allowable interconnect capacitance and resistance R_{int} by optimizing the interconnect width, *MetalWidth*. This optimized interconnect resistance is also passed to the input matching constraint to give parasitic aware synthesis. The advantage of optimizing the interconnect width is also observed during simulation.

The interconnect constraint is formulated as follows:

```
gpvar Rint Cint;

InterconnectConstr=[
     MetalWidth<=MaxWidth;
     MinWidth<=MetalWidth;
     (MetalWidth*MetalThick)==MetalArea;
     (IntLength*1e6*rho/MetalArea)==Rint;
     MetalWidth*IntLength*1e6*m1FieldAreaCap==Cint;
     Cint<=CintMax;
     Rint<=RintMax;
];
```

6.1.2. Resistance Constraint

The first target of the LNA design is to match the source resistance which is usually taken as 50 Ω. As mentioned earlier, two separate designs were performed of this LNA, the first design doesn't consider the parasitics during input matching and the next design takes into account the parasitics. For the first case, the input impedance is matched with only the source resistance which is 50 Ω. But as there are several parasitic resistances like parasitic gate resistance, R_G present at the gate of the input transistor M1, parasitic series resistance of the gate inductor L_G that is denoted as RL_G and also an interconnect resistance R_{int} from the gate inductor L_G to the gate of M1, they are added to the source resistance for input impedance match in the second case, resulting in a parasitic aware synthesis. The gate resistance R_G is modeled with the model equation obtained from *TSMC 90nm*. The interconnect length determined using the floorplan shown in Figure 57 and calculated using the unit resistance available from the technology files.

```
ResConstr=[

    (W/nr)==wr;    %nr=no of finger used
    (Using equation from foundry for intrinsic Rg)<=Rg;

    (Lg+Ls)<=LgPlusLs;
    (w0*Lg/Q)==RLg;

    (Rsource+Rg+Rint+RLg)<=Rs;
    ];
```

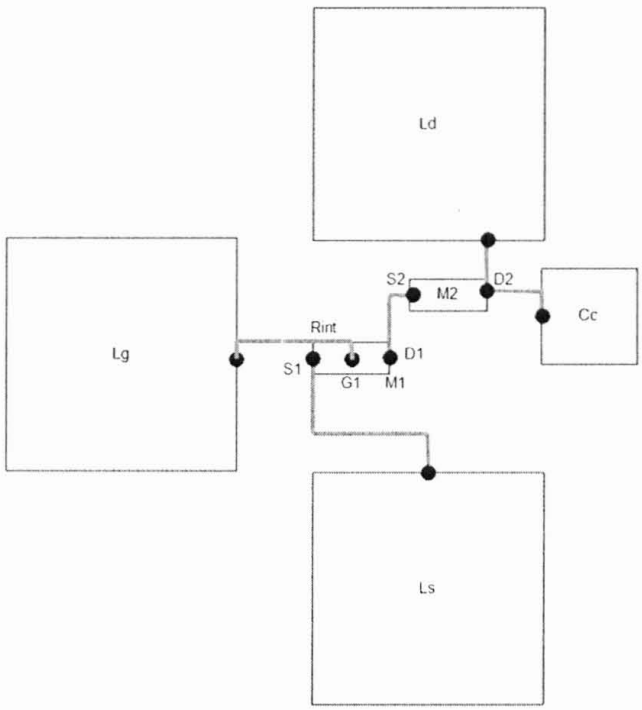


Figure 57: Floorplan for LNA to estimate parasitics

6.1.3. Matching Constraint

The cut off frequency of the LNA is given in [31] which is taken as gp variable,

$$w_T = \frac{g_m}{C_{gs}} = \frac{R_s}{L_s} \tag{39}$$

The optimal quality factor of the gate inductor, Q is found by [32],

$$Q = \sqrt{1 + \frac{1}{p}} \tag{40}$$

where, $p = \frac{\delta \alpha^2}{5\gamma}$ and $\delta = 4$, $\gamma = 2$, α=0.9

The equation for the gate inductor is found from [31], where w_0 is the targeted resonant frequency which is 5GHz

$$L_g = \frac{QL_s}{w_0} - L_s \tag{41}$$

The gate to source capacitance of the NMOS in terms of frequency is found by [32],

$$C_{gs} = \frac{1}{w_0^2(L_g+L_s)} \tag{42}$$

Also, the gate to source capacitance with respect to transistor size is found by [33] ,

$$C_{gs} = \frac{2}{3}C_{ox}WL_{min} \tag{43}$$

where W and L_{min} are the width and minimum length of the transistor respectively. All these constraints are modeled as RFconstr in GP as follows:

```
RFconstr=[
    (Rs/Ls)==wT;
    ((delta*alpha^2)/(5*gamma))==p;

    p*tmp(1)==1;
    (1+tmp(1))<=tmp(2);
    tmp(2)^0.5==QL;

    ((QL*Rs)/w0)==(LgPlusLs);% The optimal quality factor QL

    (Cgs*(w0^2*LgPlusLs))==1;
    (3/2*(Cgs/(Cox*Lmin)))<=W;
    ];
```

The inductor L_s is taken between a range of 0.3nH and 1nH and put as inductor constraint. To increase gain, the drain inductor is taken as at least 5 times of the source degenerative inductor. A lower and upper limit of L_d is also put so that it is picked inside a certain range of 1~2nH.

```
IndConstr=[
    Ls<=Lsmax;
    Lsmin<=Ls;
    (5*Ls)<=Ld;
    LdMin<=Ld;
    Ld<=LdMax;
];
```

6.1.4. Gain Constraint

The gain of the source degenerated LNA is given in [30] by the following equation

$$\frac{V_{out}}{V_{source}} = \frac{w_T}{2w_0}\frac{R_p}{R_s} \tag{44}$$

where R_p is the parallel model of the series loss resistance of the drain inductor. The values of the series and parallel loss resistance are found from [34] [35] respectively, where $Q.F.$ is the quality factor of the drain inductor.

$$R_{series} = \frac{w_0 L_d}{Q.F.} \tag{45}$$

$$R_p = \frac{(w_0 L_d)^2}{R_{series}} \tag{46}$$

```
GainConstr=[
      (w0*Ld/Q)==Rseries;
      ((w0*Ld)^2/Rseries)==Rp;
      (Rp/2/Ls/w0)==tg;
       Avmin<=tg;
];
```

6.1.5. Noise Figure Constraint

The equation to predict the noise figure is found from [30] given by,

$$NF = 1 + g_m R_s \gamma \left(\frac{w_0}{w_T}\right)^2 \tag{47}$$

which is modeled as,

```
NoiseConstr=[
    (kn*W/Lmin*Vov)<=gm;
    (1+gm*Rs*gamma*(w0/wT)^2)<=tn;
    tn<=NFmax;
];
```

6.1.6. GP solution

Finally all the constraints are passed to the GP solver and the optimal design parameters for the LNA is obtained.

```
constr=[
   InterconnectConstr;
   IndConstr;
   ResConstr
   RFconstr;
   GainConstr;
   NoiseConstr;
];
[min_Area solution status] = gpsolve(Area,constr);
assign(solution);
```

The values of the design variables for the two cases are given in Table XXIV. The first column lists the design that doesn't consider the parasitic effects during optimization and the second column is the optimal design from GP considering the parasitics.

Table XXIV: Optimal Design found by proposed methodology for the Source Degenerated LNA

Var.	Cascode Common Source LNA with Inductive Degeneration	
	Without Parasitics Considered	With Parasitics Considered
L1-L2 (μm)	0.1	0.1
W1-W2 (μm)	320	250
Ls (nF)	0.311	0.319
Lg (nF)	2.90	3.73
Ld (nF)	1.78	1.807
Rint (Ω)	16.66	3.52
CPU time (s)	0.74	0.76

6.1.7. Simulation Result

Both the design in simulated in *TSMC 90nm* technology. The value of the coupling capacitance C_c is taken as 260fF so that it gives resonance at the output node at the targeted frequency of 5GHz. Figure 58 shows the simulation result for the first case where the gate inductor parasitic resistance and interconnect resistance were not considered. In the figure, *S21*, *S11*, *S22* and the *NF* are shown in dB in terms of frequency. From the simulated result, the gain is found to be 15.6 dB which is above the required gain of 15dB. But the noise figure is found to be 2.62dB at 5GHz, which is more than the acceptable noise figure given at the specification. Both *S11* and *S22* are found to be less than 0dB, so the LNA is stable according to Stern Stability Criterion [30].

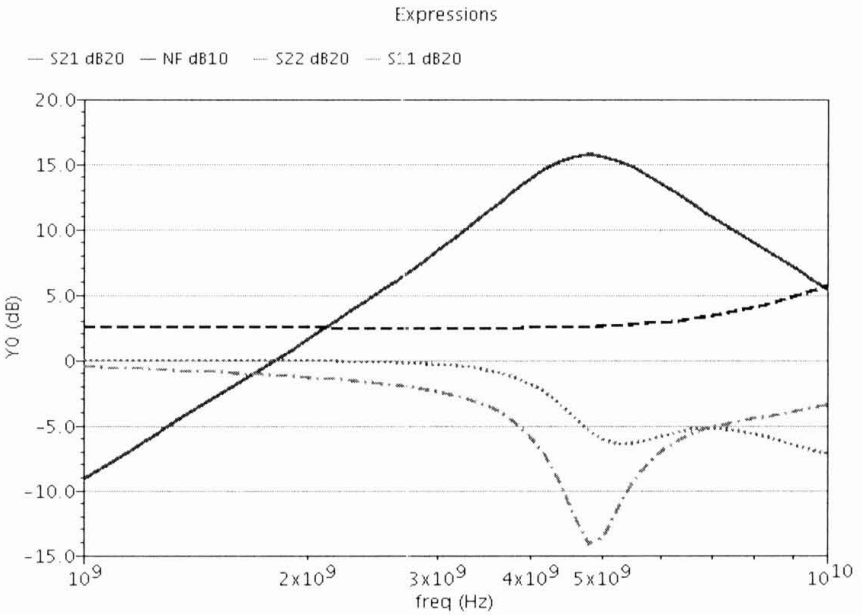


Figure 58: Simulation result of CS inductively degenerate LNA design without considering parasitics. (Red-solid: S21 (gain), Blue-dashed: NF, Green-dot dash: S11, Pink-dotted: S22)

So even though, the LNA design obtained without interconnects parasitics satisfies some of the performance requirements, the noise is found to be higher than desired.

Next, the design obtained with interconnect parasitics considered is simulated similarly in *TSMC 90nm* technology and the result is shown in Figure 59.

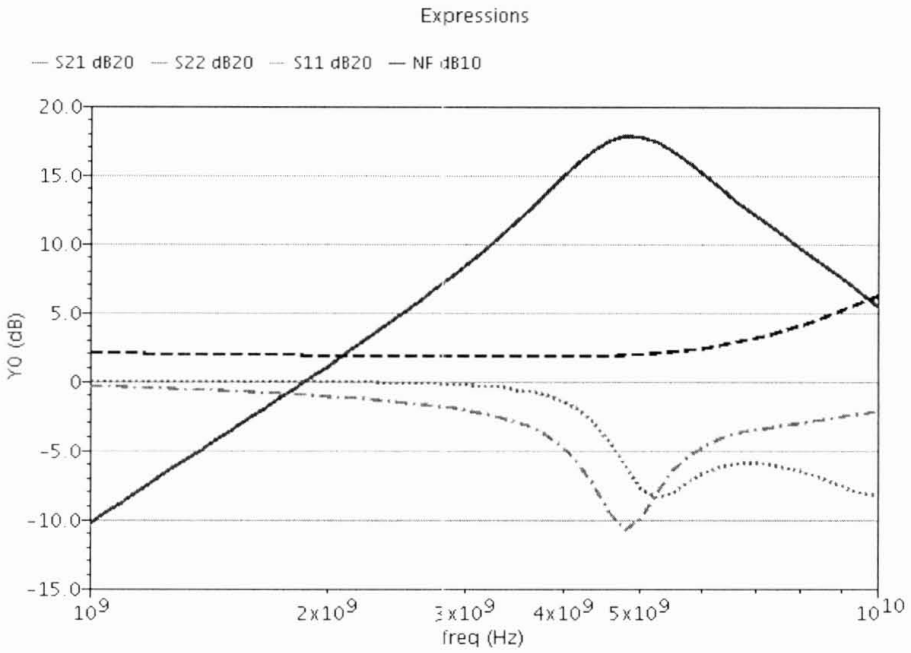


Figure 59: Simulation result of CS inductively degenerate LNA design with considering parasitics. (Red-solid: S21 (gain), Blue-dashed: NF, Green-dot dash: S11, Pink-dotted: S22)

In this case the gain (*S21*) is found to be 17.78dB, which is higher than the required 15dB and also higher than that of the design without considering parasitics. The noise figure is found to be 1.99dB which is less than the required 2dB specification. The *S11* and *S22* are found to be less than 0dB, which ensures the stability of the amplifier.

The comparison of *S21* parameters between the designs with and without considering parasitics is shown in Figure 60. In the figure the red (solid) is the design with parasitics considered and the blue (dashed) is the design without parasitics considered. From the

figure, it is noticed that incorporating parasitics to the formulation gives a better gain (*S21*) with still an acceptable noise figure from specification.

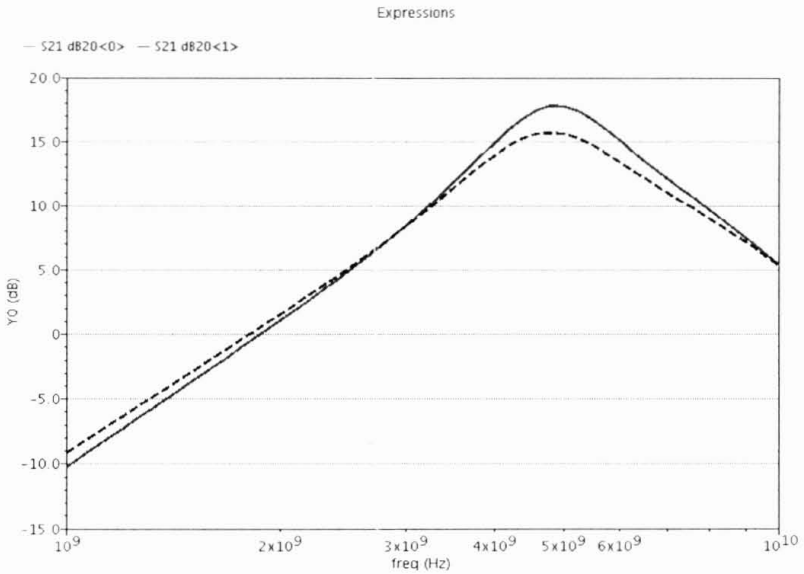


Figure 60: Gain Comparison of Source Degenerated LNA design with and without considering parasitics. (Solid-Red: With parasitics, Dashed-Blue: Without parasitics)

The comparison of the results in two cases is shown in Table XXV.

Table XXV: Comparison of CS LNA Design with and without considering parasitics

Variable	Specification	*TSMC 90nm*	
		Without Parasitics Considered	*With Parasitics Considered*
Gain, S21 (dB)	≥ 15 dB	15.603	17.78
S11 (dB)	≤ 0 dB	-13.52	-9.94
S22 (dB)	≤ 0 dB	-6.08	-7.71
NF (dB)	≤ 2 (dB)	2.62	1.995

From the comparison, the design with parasitics considered provides better gain of 17.78dB compared to the design without parasitics which gave 15.603dB, though both satisfy the requirement. The *S11* and *S22* which are the reflection coefficients are less than 0 in both cases; therefore according to Stern Stability Criterion [30], both LNA are

unconditionally stable. The noise figure of the design with considering parasitics satisfies the performance requirements and significantly better than the one without parasitics considered.

Therefore, after applying the proposed method in RF LNA, it can be concluded that the proposed parasitic aware method is applicable to optimize high frequency RF circuits in the same manner as the analog circuits optimized in the previous chapters. Furthermore, adding parasitics and optimizing interconnect simultaneously to the LNA design provides a significantly better result in terms of gain and noise performance.

In the next section, one more RF circuit, a cross coupled LC oscillator is designed and simulated in *TSMC 90nm* technology.

6.2. Cross Coupled LC Oscillator

In this section a cross coupled oscillator [36] is designed using the proposed methodology shown in Figure 61.

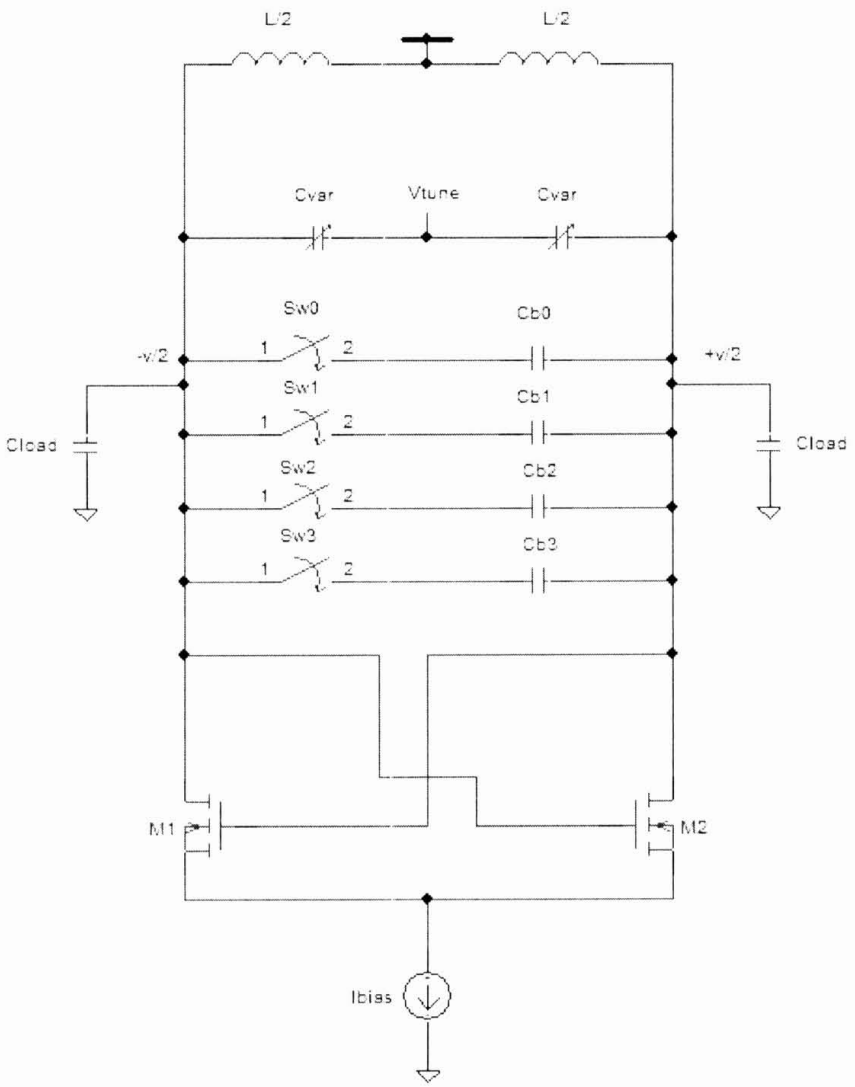


Figure 61: Cross Coupled LC Oscillator [36].

The LC oscillator is considered to have two cross coupled NMOS transistors, an inductor L, a branch with variable capacitors, C_{var} and 2^B number of switched capacitors where B is taken as 2. The branches of the switched capacitor are controlled by binary signals and shown as switches S_{w0} to S_{w3}. These branches are used to coarse tune the oscillator [36] by the capacitances C_{b0} to C_{b3}. The variable capacitor is used to fine tune the oscillator.

The design objectives of the oscillator are:

minimize *Area*

while subject to:

Centre frequency, $f = 5GHz$

Tuning Range = $\pm$ 10%

Phase Noise $\leq$ - 110*dBc/Hz* at phase noise offset frequency of 600 KHz

The GP design constraints for the LC oscillator are as follows:

6.2.1. Transistor Constraint

The transistors have to compensate the parasitic loss of the inductor and branch capacitors and provide a negative resistance to make up the loss. The condition given in [30] to compensate the loss is

$$g_m R_p \geq 1 \tag{48}$$

where R_p is parallel parasitic resistance of the inductor obtained from inductor model.

The equation is obtained from [35] as

$$R_p = \frac{(w_{osc}L)^2}{R_{series}}$$

and R_{series} is obtained from [34] in terms of quality factor, Q

$$R_{series} = \frac{w_{osc}L}{Q}$$

To make sure the transistors flow enough current to compensate the loss, the $g_m R_p$ is taken more than 6 times. Also, the minimum and maximum allowable transistor size is also included as constraint. The transistor length is taken as 0.1um which is the minimum in *TSMC 90nm* technology. The design is started by picking a suitable value of the inductor for the target technology. All these are formulated in GP as follows:

```
gpvar L Rp Rs;
indConstr=[
    L==1.33e-9;
    (w*L/Q)==Rs;
    ((L*w)^2/Rs)==Rp;
];

gpvar gm tmp W Lmin;
 tranConstr=[
    0.12e-6<=W;
```

```
    W<=320e-6;
    Lmin==0.1e-6;%Minimum channel Length
    (2*kn*W/Lmin*Ibias/2)^0.5==gm;
    gm*Rp==tmp;
    6/tmp<=1;
];
```

6.2.2. Oscillation Constraint

The equation to obtain the oscillating frequency is taken from [30] as

$$w_{osc} = \frac{1}{\sqrt{L(C_{gs}+C_{db}+4C_{gd}+C_{int}+C_d+C_{load})}} \tag{49}$$

where C_{int} is the total interconnect parasitic capacitances in the positive and negative output node and C_d is the total branch capacitance added for oscillation. The other device capacitances C_{gs}, C_{db}, C_{gd} are modeled as the equations obtained from the *TSMC 90nm* technology file. The C_{load} is taken as 200fF in the specification.

As the tuning range is considered to be ±10%, three separate capacitances C_{dMax}, C_d, C_{dMin} are calculated as the highest, minimum and lowest capacitance to oscillate at *fMin* = 4.5 GHz, *f*=5*GHz* and *fMax*=5.5*GHz* frequency respectively. A higher capacitance results in a lower oscillating frequency as can be seen from the equation. These give the range of the branch capacitance required to obtain tuning throughout the whole range.

The design is done with the width of interconnects are put as an optimization variable to give rise to parasitic aware synthesis. The reason for this is, if the interconnect width is smaller, the interconnect capacitance is reduced but the interconnect resistance is

increased. So, if the interconnect width is modeled as GP target variable and a balance is found between the total interconnect capacitance and resistance so that both are under an acceptable maximum specified value, that would ensure the best performance for the LC cross coupled oscillator in terms of oscillating frequency and phase noise. The maximum acceptable interconnect capacitance is put as $30fF$ and the maximum allowed resistance is put as 5Ω.

The GP formulation is as follows:

```
 InterconnectConstr=[
      MetalWidth<=MaxWidth;
      MinWidth<=MetalWidth;
      (MetalWidth*MetalThick)==MetalArea;
      (IntLength*1e6*rho/MetalArea)==Rint;
      MetalWidth*IntLength*1e6*m1FieldAreaCap==Cint;
      Cint<=CintMax;
      Rint<=RintMax;
 ];


OscConstr=[
    (Using equation from foundry for intrinsic Cgs)<=Cgs;

    (Using equation from foundry for intrinsic Cgd)<=Cgd;

    (Using equation from foundry for intrinsic Cdb)<=Cdb;

    (Cint+Cgs+Cdb+4*Cgd+Cload+CdMin)<=CtotalMin;
    (Cint+Cgs+Cdb+4*Cgd+Cload+CdMax)<=CtotalMax;
    (Cint+Cgs+Cdb+4*Cgd+Cload+Cd)<=Ctotal;

    (wMin^2*L*CtotalMax)==1;
    (wMax^2*L*CtotalMin)==1;
    (w^2*L*Ctotal)==1;
];
```

6.2.3. Phase Noise Constraint

A constraint is set to make the phase noise of the LC oscillator less than a maximum allowed phase noise, PN_{max} . This value is taken to be less than -110dBc/Hz.

The phase noise can be written from [36] as

$$PN = \frac{1}{16\pi^2 f_{off}^2 C_{tl}^2 V_{osc}^2}\left(\frac{4kTR_{series}}{4\pi^2 f^2 L^2} + 0.5 * 4kT\gamma g_m\right) \tag{50}$$

where f_{off}^2 is the offset frequency of phase noise taken as 600KHz, C_{tl} is the total capacitance parallel with the loop inductor L, and f is the centre frequency. The GP formulation is done in the following way

```
PhaseNoiseConstr = [
     (4*k*T*Rs/4/pi^2/f^2/L^2)==loopNoise;
     (4*k*T*gamma*gm)==tranNoise;
     (16*pi^2*fOffset^2*Ctl^2*Vosc^2)*tn1==1;
     (tn1*(loopNoise+0.5*tranNoise))<=PN;
     PN<=PNmax;];
```

6.2.4. GP Solution

Finally *gpsolve* is used to obtain the solution of this design problem

```
constr=[
    indConstr;
    tranConstr;
    OscConstr;
    PhaseNoiseConstr
    AreaConstr;
];
```

```
[min_area solution status] = gpsolve(Area,constr);
assign(solution);
```

The solution from GP is shown in Table XXVI

Table XXVI: Design Variables obtained from GP for cross coupled LC oscillator

Variable	*Cross Coupled LC Oscillator In TSMC90nm*
L1=L2 (μm)	0.1
W1=W2 (μm)	245.8
L (nH)	1.33
CdMin for 5.5GHz (fF)	101.67
Cd for 5GHz (fF)	256.17
CdMax for 4.5GHz (fF)	478.10
Ibias (mA)	2.2
CPU Time (s)	0.5435

Because there are four branches of parallel capacitor with switch, and the lowest and highest capacitance is between the range of $100fF$ to $500fF$, the values of the branch capacitor *Cb0* to *Cb3* are taken as $50fF$, $100fF$, $150fF$, $200fF$ respectively. The variable capacitance C_{var} is used as *nmoscap* which is available in *TSMC90rf* library for using as varactor. The value of the length and width of the varactor is taken as 2um and 5.5um respectively which gives the varactor range as $102.59fF$ (at 0V) and $160.99fF$ (at full voltage). As two varactors are in series, the total contribution lies between $50fF$ - $80fF$ which is used to fine tune the oscillating frequency. Using the combination of switches of parallel capacitors and the tuning voltage of the varactor, the obtainable range of capacitance is from $50fF$ (all switches open, $V_{tune} = 0V$) - $580fF$ (all switches closed, $V_{tune} = 1.2V$).

6.2.5. Simulation Result

The simulated result in *TSMC 90nm* is shown in Table XXVII. From the table, it is seen that for a value of switched capacitance of $200fF$ and a varactor capacitance of $51.29fF$, the total capacitance is $251.29fF$ and the oscillating frequency if $5GHz$ which is the target frequency. This capacitance is close to what was obtained from GP as C_d which was $256.17fF$. For other combinations, the oscillating frequency can be obtained from $4.2GHz$ to 5.8 GHz which more than the ±10% tuning ranges. The total power consumption for the current $I_{bias} = 2.2mA$, is $2.64mW$.

Table XXVII: Simulation result for different capacitances of the cross coupled LC oscillator

Total Switched Capacitance, Cb (fF)	Vtune (V)	Approx. Variable Cap, Cvar/2 (fF)	Ctotal (fF) =Cb+Cvar/2	Oscillating Frequency (GHz)
0	0	51.29	51.29	5.8
50	0	51.29	101.29	5.6
100	0	51.29	151.29	5.4
150	0	51.29	201.29	5.2
200	0	51.29	251.29	5
250	0	51.29	301.29	4.8
300	0	51.29	351.29	4.8
350	0	51.29	401.29	4.6
400	0	51.29	451.29	4.4
450	0	51.29	501.29	4.4
500	0	51.29	551.29	4.2
500	1	80.54	580.54	4.2

Figure 62 shows the result for the total capacitance of 251.29*fF* that gives the oscillating frequency of 5*GHz* which is very close to the result 256.17*fF* that was initially obtained by synthesis with interconnect optimization.

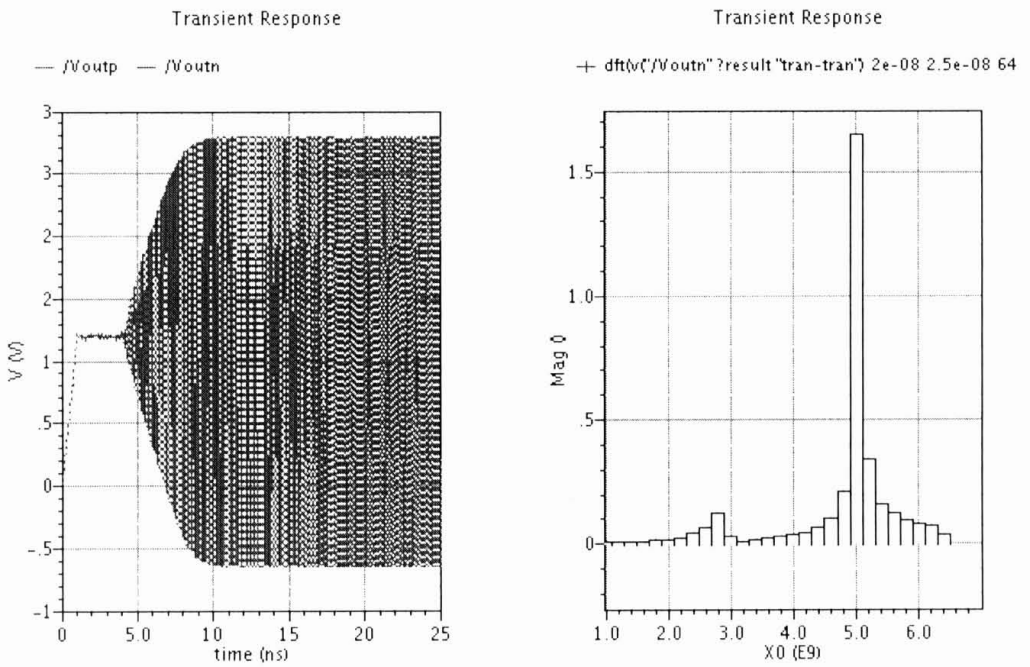


Figure 62: Simulation Result for approximated total capacitance of 251.29fF for LC oscillator in *TSMC 90nm.*

The corresponding phase noise curve is obtained from pss and pnoise simulation shown in Figure 63.At 600 *KHz*, the phase noise is found to be -121.6 *dBc/Hz*, which is less than the required maximum specified phase noise of -110 *dBc/Hz*.

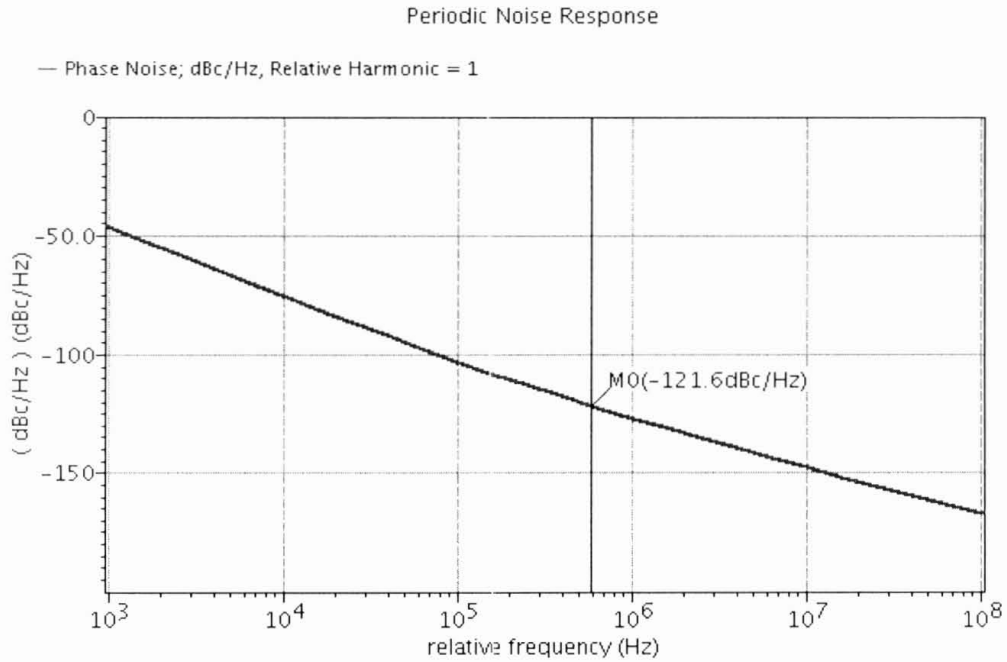


Figure 63: Phase noise at 600KHz for approx. total capacitance of 251.29fF for LC Oscillator in *TSMC 90nm* technology that yields oscillation 5GHz.

So the designed cross coupled LC oscillator obtained by the proposed methodology attains all the performance requirements, tuning range and phase noise requirements.

6.3. Summary

In this chapter, two RF circuits, a source degenerated LNA and a cross coupled LC oscillator are designed and tested using the proposed methodology based on Geometric Programming. In both cases, the obtained solution satisfied all the performance requirements. The solution time is also found to be very low. So it can be concluded that,

the proposed methodology can be successfully applied to optimizing RF circuits as well as the analog circuits as shown in previous chapters.

Since using the traditional approaches that use commercial simulators and generating actual layout and extraction using off-the shelf extractor might take a long time to converge depending on multiple runs, it is convenient to have a fast parasitic aware method that can simultaneously estimate the parasitics and give a solution in quick time. From the two RF design examples shown above, this parasitic aware synthesis method can serve that purpose and provide a solution in quick time.

The next chapter provides the conclusion and future scope of this work.

7. Conclusions and Future Work

In this thesis, a fast parasitic aware circuit synthesis methodology for high performance Analog and RF circuits is proposed and tested. The proposed method utilizes a convex optimization problem called Geometric Programming by modeling the circuit performance constraints and parasitic contribution and provides a global solution. The methodology is applied to five analog circuit including three Operational Amplifiers and two Analog Comparators and two RF circuit which includes RF Low Noise Amplifier and a Cross Coupled LC Oscillator. The obtained result is simulated in both Pre and Post Layout Condition and the results show the efficacy of the proposed methodology.

7.1. Contribution of the Thesis

The contribution of the thesis is listed below:

- The Thesis introduces a high speed parasitic aware method which is not dependent on initial conditions and provides a global solution in very quick time. The method does not use any off-the shelf simulator inside the optimization phase and uses a very fast and efficient GP Solver.
- At the beginning, the thesis performs a sensitivity analysis on dynamic comparator structures and emphasizes on the importance of parasitic aware synthesis methodology (Chapter 2).

- Some initial setup work includes working with CMC to identify and fix errors to run transistor level simulation in *STM 90nm* and *TSMC 90nm* technology and create tutorials for them. Those are added in the Appendix. Furthermore, the parasitics extraction process using Calibre in *TSMC 90nm* is also formulated and presented as well.

- The thesis proposes a fast parasitic-aware synthesis methodology that can be applied to both Analog and RF circuit synthesis, as a convex optimization problem using Geometric Programming. The thesis implements the design of a two stage N-input Op Amp using Geometric Programming and introduces a simultaneous floorplan design and adding parasitic components inside the circuit synthesis phase (Section 4.1). It adds the parasitics by modeling the device parasitics from respective technologies e.g. *TSMC 0.18um* and *TSMC 90nm* and creates symbolic interconnect parasitics model be generating a minimum sized floorplan with matching constraints. These parasitics models are then added to performance constraints and results in a parasitic aware synthesis. The thesis also designs the same problem without considering the interconnect parasitics and by comparing both results, the optimization that is obtained with considering all parasitic effects using the proposed methodology is found to provide much better performance result.

- The proposed parasitic aware methodology is then applied on two other high performance analog amplifiers (Section 4.2, 4.3), two analog comparators (Chapter 5) and two RF circuits (Chapter 6). A total of seven analog and RF

circuits are modeled and optimized with the proposed parasitic aware synthesis. The designs are tested in pre-layout Cadence simulation and a noticeable improvement is found in all cases when parasitics are considered.

- The optimal designs obtained for the Analog Circuits are then used to generate layout by Layout-XL. The layout are extracted and post-layout simulation is performed which show fairly consistent result between the post-layout and the pre-layout simulation. This verifies the accuracy of the implemented parasitics modeling proposed in this thesis.

7.2. Future Scope of This Work

Future scope of this work includes testing this methodology on other Analog and RF circuits. A library can be created with all the building blocks of Analog and RF system and the whole system can be designed in fairly quick time using this fast parasitic aware synthesis approach. The approach can also be used as an excellent starting point for other evolutionary based approaches whose performance relies greatly on the initial starting point. As any methodology that uses commercial simulator and layout generator and extractor is CPU intensive, it will be greatly beneficial to start from a point which is close to the solution space and fine tune the solution. Compared to other approaches that use a pre-generated look up table and extrapolates the values for parasitic estimation, this method uses a very accurate intrinsic device parasitics equations available from foundry

for symbolic modeling and in the same way, can easily incorporate fringe capacitances model if it becomes available.

7.3. List of Publications

Three publications have been made from the partial portion (up to Chapter 4.2) given in this thesis. They are

[1] Abdullah Al Iftekhar Ahmed and Lihong Zhang, "A Fast Parasitic Aware Synthesis Method of High Performance Analog Circuits," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012.

[2] Abdullah Al Iftekhar Ahmed and Lihong Zhang, " Sensitivity Analysis of Layout Parasitics, Circuit Sizing and Topology Variance on Analog Circuits," in *the Twentieth Annual Newfoundland Electrical and Computer Engineering Conference (NECEC),* 2011.

[3] Abdullah Al Iftekhar Ahmed and Lihong Zhang," Dynamic Comparators and Parasitics," in *the Nineteenth Annual Newfoundland Electrical and Computer Engineering Conference (NECEC)* 2010.

Bibliography

[1] J. Mueller, R. Thomaa, E. Demircanc, C. Bernicotb and A. Juge, "Modeling of MOSFETparasiticcapacitances, and their impact on circuitperformance," *Solid-State Electronics,* vol. 51, no. 11–12, p. 1485–1493, 2007.

[2] B. Sviličić and A. Kraš, "Cmos Technology: Challenges for Future Development," Pomorstvo, god. 20, br. 2 (2006), str. 97-104, 2006.

[3] M. White and Y. Chen, "Scaled CMOS Technology Reliability Users Guide," Jet Propulsion Laboratory Publication, NASA Electronic Parts and Packaging (NEPP) Program, 2008.

[4] L. Wei, F. Boeuf, T. Skotnicki and H.-S. P. Wong, "CMOS technology roadmap projection including parasitic effects," in Proceedings of *International Symposium on VLSI Technology, Systems, and Applications*, pp. 78-79, 2009.

[5] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSIs," *IEEE Transactions on Electron Devices,* vol. 40, no. 1, pp. 118-124, 1993.

[6] R. A. Rutenbar, "Simulated annealing algorithms: an overview," *IEEE Circuits and Devices Magazine,* vol. 5, no. 1, pp. 19-26, 1989.

[7] T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Simultaneous multi-topology multi-objective sizing across thousands of analog circuit topologies," in Proceedings of the *44th annual Design Automation Conference*, pp. 944 - 947, 2007.

[8] H. Graeb, S. Zizala, J. Eckmueller and K. Antreich, "The sizing rules method for analog integrated circuit design," in Proceedings of *IEEE/ACM International Conference on Computer Aided Design*, pp. 343 - 349, 2001.

[9] J. He, S. Zhan, D. Chen and R. Geiger, "Analyses of Static and Dynamic Random Offset Voltages in Dynamic Comparators," *IEEE Transactions on Circuits and Systems,* vol. 56, no. 5, pp. 911-919, 2009.

[10] L. Sumanen, M. Waltari and K. Halonen, "A mismatch insensitive CMOS dynamic comparator for pipeline A/D converters," in Proceedings of *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 32 - 35 vol.1, Jounieh, Lebanon, 2000.

[11] L. Sumanen, M. Waltari, V. Hakkarainen and K. Halonen, "CMOS dynamic comparators for pipeline A/D converters," in Proceedings of *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. V-157 - V-160 vol.5, 2002.

[12] M. Dessouky, M.-M. Louerat and J. Porte, "Layout-oriented synthesis of high performance analog circuits," in Proceedings of *The Design, Automation, and Test in Europe (DATE)*, pp. 53 - 57, 2000.

[13] P. Vancorenland, G. V. d. Plas, M. Steyaert, G. Gielen and W.Sansen, "A layout-aware synthesis methodology for RF circuits," in Proceedings of *The International Conference on Computer-Aided Design (ICCAD)*, pp. 358 - 362, 2001.

[14] A. Agarwal, H. Sampath, V. Yelamanchili and R. Vemuri, "Fast and accurate parasitic capacitance models for layout-aware synthesis of analog circuits," in Proceedings of *The Design Automation Conference (DAC)*, pp. 145 - 150, 2004.

[15] M. Ranjan, W. Verhaegen, A. Agarwal, H. Sampath, R. Vemuri and G. Gielen, "Fast, layout-inclusive analog circuit synthesis using pre-compiled parasitic-aware symbolic performance models," in Proceedings of *The Design, Automation, and Test in Europe*, pp. 604 - 609 Vol.1 2004.

[16] A. Agarwal and R. Vemuri, "Layout-aware RF circuit synthesis driven by worst case parasitic corners," in Proceedings of *International Conference on Computer Design (ICCD)*, pp. 444 - 449, 2005.

[17] G. Zhang, A. Dengi, R. Rohrer, R. Rutenbar and L. Carley, "A synthesis flow toward fast parasitic closure for radio-frequency integrated circuits," in Proceedings of *The Design Automation Conference*, pp. 155-158, 2004.

[18] X. Wang, S. McCracken, A. Dengi, K. Takinami, T. Tsukizawa and Y. Miyahara, "A Novel Parasitic-Aware Synthesis and Verification Flow for RFIC Design," in Proceedings of *European Microwave Conference*, pp. 664 - 667, 2006.

[19] R. Castro-Lopez, O. Guerra, E. Roca and F. Fernandez, "An Integrated Layout-Synthesis Approach for Analog ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 7, pp. 1179 - 1189, 2008.

[20] H. Habal and H. Graeb, "Constraint-Based Layout-Driven Sizing of Analog Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* , vol. 30, no. 8, pp. 1089-1102, 2011.

[21] K. Choi and D. Allstot, "Parasitic-aware design and optimization of a CMOS RF power amplifier," *IEEE Transactions on Circuits and Systems,* vol. 53, no. 1, pp.

16-25, 2006.

[22] Y. Xu, K.-L. Hsiung, X. Li, L. Pileggi and S. Boyd, "Regular Analog/RF Integrated Circuits Design Using Optimization With Recourse Including Ellipsoidal Uncertainty," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 28, no. 5, pp. 623-637, 2009.

[23] M. Hershenson, S. Boyd and T. Lee, "Optimal design of a CMOS op-amp via geometric programming," *Computer-Aided Design of Integrated Circuits and Systems,* vol. 20, no. 1, 2001.

[24] S. Boyd, S.-J. Kim, L. Vandenberghe and A. Hassibi, "A Tutorial on Geometric Programming," 2007. [Online].

[25] R. Schwencker, J. Eckmueller, H. Graeb and K. Antreich, "Automating the sizing of analog CMOS circuits by consideration of structural constraints," in Proceedings of *The Design, Automation, and Test in Europe (DATE)*,pp. 323 - 327, 1999.

[26] M. C. Schneider and C. Galup-Montoro, CMOS Analog Design Using All-Region MOSFET Modeling, Cambridge University Press, 2010.

[27] P. E. Allen and D. R. Holberg, CMOS Analog Circuit Design, Oxford University Press, 2002.

[28] D. Johns and K. Martin, Analog Integrated Circuit Design, John Wiley & Sons, Inc., 1997.

[29] V. Katyal, "A New High Precision Low Offset Dynamic Comparator for High Resolution High Speed ADCs," in Proceedings of *IEEE Asia Pacific Conference on Circuits and Systems*, pp. 5 - 8, 2006.

[30] B. Razavi, RF Microelectronics, Prentice Hall, 2012.

[31] J. P. Silver, *MOS COMMON-SOURCE LNA Design Tutorial, http://www.odyseus.nildram.co.uk/RFIC_Circuits_Files/MOS_CS_LNA.pdf.*

[32] I. C. Design Group Universitat Politècnica de Catalunya, *Low Noise Amplifiers, http://weble.upc.es/rfcs/Material/Tema5_LNA.pdf.*

[33] T. H. Lee, The Design of CMOS Radio-Frequency Integrated Circuits, Cambridge University Press, 2003.

[34] C. Bowick, RF Circuit Design, Newnes, 2007.

[35] B. Razavi, "A 1.8 GHz CMOS voltage-controlled oscillator," in Proceedings of *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 388 - 389, 1997.

[36] S. Boyd, S. J. Kim and S. S. Mohan, "Geometric Programming and Its Applications to EDA Problems," in Proceedings of the *Design, Automation and Test in Europe (DATE) Tutorial*, 2005.

Appendices

Appendix A

Tutorial for Running Transistor Level Simulation in Cadence for STM 90nm Technology

This tutorial is for the two *Solaris* servers *Panther* and *Jaguar* in the university. For new *linux* servers, please contact system administrator for updated license and file locations.

1. Login to Panther
2. Create a new directory.
3. Make sure you don't have .cdsinit in your home directory
4. Copy *cds.lib* from /nfs/CMC/mnt/kits/cmos90nm.3.0/CMCdir in the new directory (If this file has been relocated to a new directory in some other server, contact system administrator)
5. Type *startCds –t cmos90nm* in the terminal.
6. Your script will fail.
7. Now, logout from panther and login to Jaguar and go to your newly made directory.
8. Run the *startCds* script again. It will be successful this time.
9. Create your schematic e.g. if you want an inverter, use *nlvt* and *plvt* for NMOS and PMOS transistors from cmos90nm library.

10. Open Analog Environment
11. Setup your simulations (DC, AC, transient, etc.). For transient analysis, select analysis->choose->tran and give stop time.
12. Go Tools->Setup Corners
13. Under 'Corners' select TT, under 'Library' select DK
14. Click on 'Save Model File' button
15. Netlist and Run
16. If everything is successful, you will get the simulation result.

Appendix B

Tutorial for Running Transistor Level Simulation in *TSMC 90nm* Technology

1. In *.cshrc* at your home directory use cadence 2009, if it is setup as 2007 (in *Panther*).

2. Create a new directory.

3. Execute the following commands. More detailed instructions can be found in the pdk usage guide.

```
cp <pdk_install_direcotry>/display.drf .
ln –s <pdk_install_directory>/models .
ln –s <pdk_install_directory>/stream .
cp -f <pdk_install_directory>/assura_tech.lib .
ln –s  <pdk_install_directory>/Assura .
ln –s  <pdk_install_directory>/Calibre .
```

In *panther*, <pdk_install_directory> is located at the following directory */CMC/kits/tsmc_90nm/CRN90G/T_N90_CM_SP_013_K1_V1.0A/*. For *Cadlamss (linux server),* find the similar directory.

4. Create a cds.lib and add the following line

INCLUDE /CMC/kits/tsmc_90nm/CRN90G/T_N90_CM_SP_013_K1_V1.0A/cds.lib

5. Execute *icfb&*, it will open Cadence.

6. Draw the schematic of your circuit.

7. Open tools > analog environment

8. Set transient analysis, select outputs to be plotted and run the simulation (you don't need to change the simulator like in *0.18um*, the *spectre* is selected by default)

9. The transient response will be shown after simulation.

Appendix C

Running Calibre Design Rule Check (DRC) and Parasitic Extraction (PEX) for TSMC90nm Technology

Calibre DRC:

1. In your directory run, (*Panther*)

source /CMC/scripts/setenv.calibre.csh

2. Open Cadence

3. In *CIW* window type

load("setupCalibre.il")
setupRVE()

If you just created symbolic link, and *setupCalibre.il* is not in your directory you will have to copy it from */nfs/CMC/mnt/tools/mentor/calibre_2009.4_31.27/CMCdir*

(Again, contact system administrator, if they are relocated in the new servers).

3. Draw the layout.

4. Open Calibre->DRC

5. At the rules tab, select both the rule file in your directory */Calibre/drc*

6. Set the TOPCELLNAME as your layout name

7. Click Run DRC, you will see DRC summary report.

Parasitic Extraction from Layout (Calibre PEX):

1. In your directory run,

source /CMC/scripts/setenv.calibre.csh

2. Open Cadence

3. Copy "calibre.rcx" file from ../Calibre/rcx/ to your home directory and tsmc90nm directory. Also copy the 'rules' file.

Edit the 'calibre.rcx' as following:

```
LAYOUT PRIMARY "layouttest"
LAYOUT PATH "layouttest.gds"
LAYOUT SYSTEM GDSII
//LAYOUT PATH "layout.net"
//LAYOUT SYSTEM SPICE

//SOURCE PRIMARY "ts_allcelllvs"
//SOURCE PATH "ts_allcelllvs.cdl"
//SOURCE SYSTEM SPICE
```

4. In CIW window type

load("setupCalibre.il")

```
setupRVE()
```

5. To extract capacitance on some metal lines, create shape pin on those metal lines in the following way:

a. Select layer M1 (pin) in LSW window

b. In Virtuoso select Create - > Pin

c. In the opened form, select shape pin

d. Draw rectangle for pin and add a terminal name

6. Create label using the same layer in LSW window with its origin over the pin shape.

As an example, some metal lines are shown with their labels

Figure: Metal lines for Capacitance Extraction

7. Open Calibre-PEX

8. In the rules tab, select the edited "rules.rcx" file and click load

9. Run PEX

10. When it finishes, close start RVE to view the capacitance.

Figure: Extracted Capacitance in Calibre RVE

c. In the opened form, select shape pin

d. Draw rectangle for pin and add a terminal name

6. Create label using the same layer in LSW window with its origin over the pin shape.

As an example, some metal lines are shown with their labels.

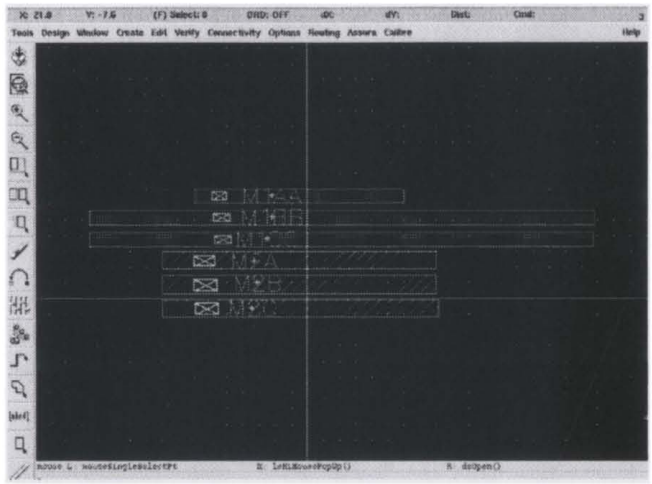


Figure: Metal lines for Capacitance Extraction

7. Open Calibre->PEX

8. In the rules tab, select the edited 'calibre.rcx' file and click load.

9. Run start PEX

10. When it finishes, click start RVE to view the capacitance.

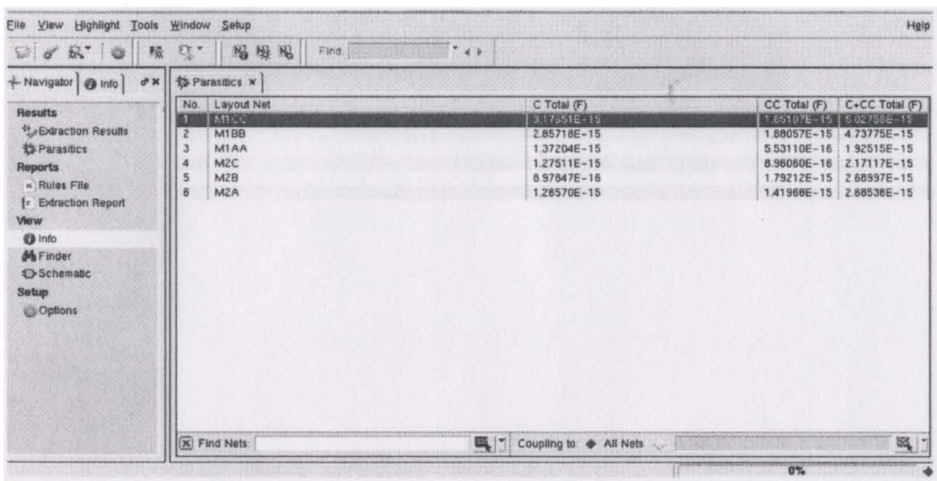


Figure: Extracted Capacitance in Calibre RVE

Extracting resistance:

To find resistance of an interconnect, add two ports across metal line in the following way:

1. Select the appropriate pin layer (e.g. M1 (pin)) in LSW.
2. In Virtuoso window select Create->Pin.
3. In create pin form select shape pin (not symbolic, which is the default) and enter a terminal name in the form.
4. Draw the two corners of the pin on your metal line.
5. In Virtuoso select Create > Label, and again use the terminal name you specified
6. Repeat for your second pin.

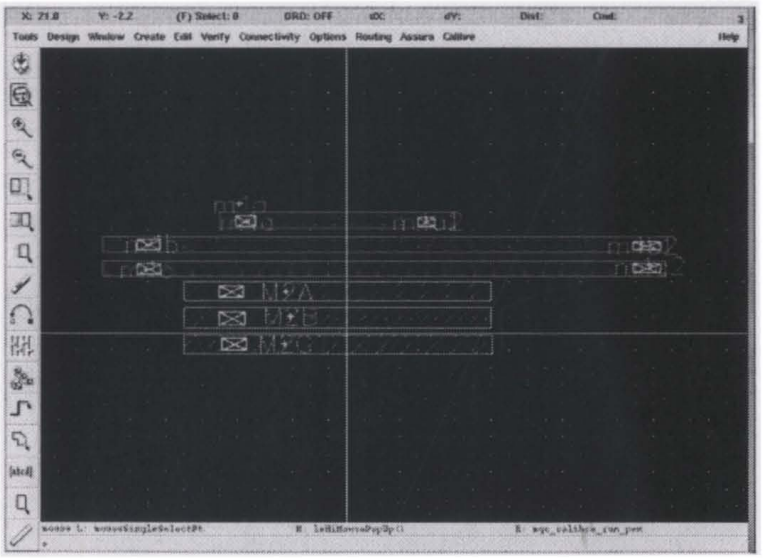


Figure: Metal lines for Resistance Extraction

After running Calibre PEX, click the *start RVE* to view the extracted resistances.

Extracting resistances

To find resistance of an interconnect, add two pins across metal line in the following steps:

1. Select the appropriate pin layer (e.g. M1_pin) in LSW.
2. In Virtuoso window select Create->Pin.
3. In create pin form select shape pin (not symbolic, which is the default) and enter a terminal name in the form.
4. Draw the two corners of the pin on your metal line.
5. In Virtuoso select Create > Label, and again use the terminal name you specified.
6. Repeat for your second pin.

Figure [illegible]: Pins for Resistance Extraction

After running Calibre PEX, select the sheet RES to view the extracted resistances.

Figure: Extracted Resistance in Calibre RVE