Design Automation of Analog to Digital Converters based on Geometric Programming

by

Purushothaman A

(200721004)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

in

Information and Communication Technology

to

Dhirubhai Ambani Institute of Information and Communication Technology

DEC 2012

Declaration

This is to certify that

1. the thesis comprises my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,
2. due acknowledgment has been made in the text to all other material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled "Design Automation of Analog to Digital Converters based on Geometric Programming " has been carried out by Purushothaman A (200721004) for the degree of Doctor of Philosophy in Information and Communication Technology at this Institute under my supervision.

Prof. Chetan D. Parikh

Abstract

The trend towards digital processing of analog signals in an increasing number of application fields has stimulated significant research efforts in the area of data converters implemented in CMOS technologies. Designing data converters for a given specification takes an unacceptably large amount of designer's time. The primary objective of the work reported here is to map human (expert) knowledge into the implementation of an Analog to Digital Converter (ADC) design system to speed up the overall ADC design process.

This work presents a Geometric Programming(GP) based design automation of ADCs. This automation gives a design with minimum power dissipation while taking other specifications as constraints. GP is an optimization problem, whose accuracy depends on the accuracy of equations used in it. Since using a square law model for short channel transistors runs into large errors, we have used a new transistor model to increase the accuracy of GP. Among the data converters, the design of Successive Approximation Register (SAR) ADC and Pipelined ADC are presented in this work.

As the first step of designing ADCs, different types of comparator architectures, namely OpAmp based comparator, static latch based and dynamic latch based comparators are designed. Both static latch based and dynamic latch based comparators employ positive feedback. For getting an accurate delay model from a positively fed back latch, we have developed a large signal model.

Capacitor array digital to analog converters (DAC) are an integral part of ADCs. The performance of ADCs depend upon the performance of the DACs employed in it. Different switching schemes are employed to enhance the performance of the DAC. To get a better performance we have developed a novel capacitor array based DAC. In this architecture three unit capacitors replaces the Most Significant Bit(MSB) capacitor of the conventional DACs, which reduces the area, power dissipation and settling time. The novel DAC neither requires an additional reference voltage nor an additional switching circuit. The architecture is verified by drawing a layout and doing post layout simulations.

A GP based automation algorithm for SAR ADC is written in MATLAB. According to the given specifications, the algorithm chooses the sub-blocks and integrates it to get a required SAR ADC. SPICE simulation results of a single ended 12 bit 1 MS/s SAR ADC in a 0.8 μm CMOS technology employing the proposed automation shows that the ADC consumes 0.8 mW power and thus justifies the proposed methodology.

This work also presents a systematic approach to the design of a multi bit per stage pipelined ADC. An algorithm based on geometric programming using MATLAB tool is developed to design pipelined ADC. The proposed algorithm computes the required design variables in a predefined ADC topology, for a specified process technology.Design variables includes number of stages, number of bits per stages, capacitor sizes in multiplying DAC and transistor sizes and biasing voltages in residue amplifier. Two additional methods, namely capacitor scaling and gain error improvements are adopted to reduce the power further. Using the proposed approach a 14 bit, 30 MS/s pipelined ADC with 3 V as supply voltage is designed in CADENCE with 0.18 μm technology.

The contributions of this work are:

1. A more accurate large signal delay model is proposed for both static and dynamic latch based comparators.

2. A novel Capacitor Array based DAC is developed which reduces power, area, noise and settling time.

3. Automation algorithms each for SAR ADC and Pipelined ADC are proposed to design the ADC for the given specifications.

Acknowledgments

It is with great pleasure that I acknowledge several of the many people who have supported me on my journey through the Ph.D. program. They provided me with so many of the things that I needed: from some, I received technical support; from some, professional and academic advice; from some, friendship; and from some, a combination of these things. To the people mentioned here, and to many others who are not, I extend my sincere, deepest, heartfelt gratitude.

My first and foremost acknowledgement is to my advisor, Prof. Chetan D. Parikh, without whose faith and encouragement this work would not have happened. I really appreciate his love for scientific enquiry, academic honesty, desire to share knowledge and teach through example, his expression of excitement in gaining new scientific insights and his passion to discuss new ideas in and beyond our field. Yet another thanks for being a wonderful co-author on the many papers we wrote together.

I was delighted to interact with Prof. Dipankar Nagchoudhuri by attending his classes and having him as my thesis committee member. I would like to thank Prof. Subhajit Sen for introducing me CADENCE and motivating me to use the tool for my thesis. I would like to thank Prof. Gautam Dutta, with whomIspent lot of evenings discussing about technical and non-technical things. Prof V.P. Sinha and Prof. Samaresh Chatterji deserve special thanks for enriching me with their entertaining classes.

A special thanks to Prof. Rahul Dubey, under whomIworked as Teaching Assistant for most of my stay at DAIICT, and who motivated me wheneverIfelt down.

I like to thank Tatu for the nice Tea talks, discussion regarding topologies and space, to help me to understand and derive the adomian polynomials.

I thank my fellow lab mates : Ratnik, Pratik, Sunil, Divyang, Nilesh, Jignesh, Gunesh, Bhavesh, Shubham and Gannu for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years.

I wish to extend my thanks to Arun and Khaja for making my hostel stay pleesent and entertaining.

Last, but not least, I would like to dedicate this thesis to my family, my wife Sharmi and daughter Shivani, for their love, patience, and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

All the real world signals are analog signals. Processing of real world signals hence require analog to digital interface and similarly digital to analog interface. Data converters form an integral part of these interfaces and hence assume greater significance. As shown in Fig. 1, there exist many different Analog to Digital Converter(ADC) architectures for different applications. For high speed, low resolution A/D conversions, flash and pipeline ADCs are mainly used. The successive approximation ADC provides moderate conversion speed and moderate resolution. The delta-sigma ADC is used for medium or low speed and/or high-resolution A/D conversions. The dual-slope ADC is good for very high-resolution and very low-speed applications. The incremental ADC can be seen as a delta-sigma ADC with periodical reset. It can achieve higher resolution and faster speed than the dual-slope ADC.

1.2 Previous work

Much work has been done to optimize both the digital and analog circuits. Automation algorithms used to optimize the circuits can be broadly classified into

- Evolutionary algorithms, generic population based meta heuristic optimization algorithms like Genetic algorithms and genetic programming for ring oscillator, pipelined ADC [2] and VCO design [3] , Artificial intelligence [4], Simulated annealing [5] and Particle swarm optimization for two stage amplifier [6],

- Linear constrained optimization like Integer programming for basic analog cell design [7],

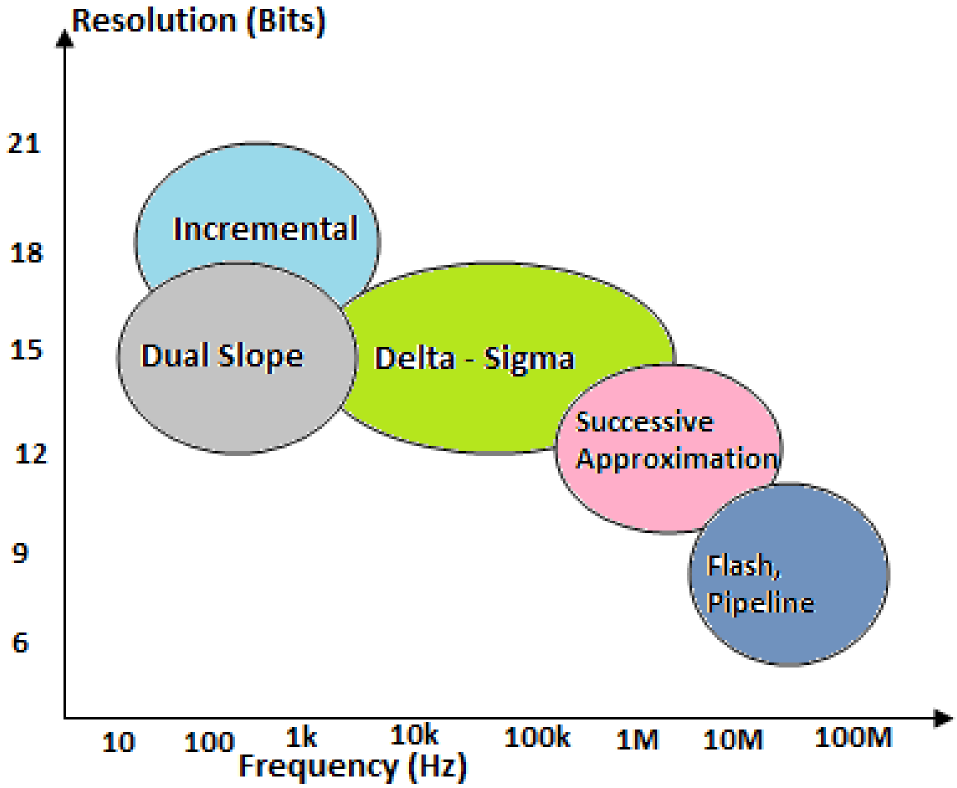

Figure 1.1: ADC architectures for different applications [1].

- Stochastic pattern search for designing two stage and cascoded amplifier [8], which incorporates probabilistic elements and
- Nonlinear constrained optimization methods like geometric programming [9,10].

Most of the above mentioned methods except geometric programming use a simulation tool as a part of the optimum loop which consumes a lot of time and can also get stuck in local optima. There is a scarcity of literature for automating mixed signal circuits.

This work uses geometric programming for automating the design of analog to digital converters, mainly for Successive Approximation Register(SAR) ADC and pipelined ADC. We also developed a novel capacitor array based digital to analog converter.

1.3 Thesis organization

The thesis is organized as follows. The review of geometric programming optimization and its application to circuit sizing is explained in chapter 2. The third chapter describes the design procedures of different comparator architectures. Fourth chapter proposes a novel DAC architecture and discusses the simulation results and comparison with the conventional architectures. The geometric programming based design procedure and an example for SAR architecture is presented in the fifth chapter. The sixth chapter analyzes the issues that would arise up during the design of pipelined ADC with capacitor scaling and gain error improvement using GP and proposes ways to solve them. The seventh chapter concludes the thesis.

Chapter 2

Geometric Programming

2.1 Basic Geometric Programming

Geometric Programming(GP) based circuit optimization has been in use since the 1980s [11-15]. Digital circuit design problems that were formulated as GPs can be found in [12,13]. In recent years, GP has been used for the optimization of analog and mixed-signal circuit problems [14,15]. A geometric program is an optimization problem of the form

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

$$\text{such that } f_i(x) \leq 1, i = 0, 1, 2...m$$

$$\begin{aligned} g_i(x) &= 1, i = 0, 1, 2...p \\ x(i) &> 0, i = 0, 1, 2...n. \end{aligned} \tag{2.1}$$

where $f_0(x)$ is the objective function and $f_i(x)$ and $g_i(x)$ are inequality and equality constraints respectively. In these constraints $f_0, ..., f_m$ are posynomial functions and $g_1, ..., g_p$ are monomial functions. A posynomial function has the form

$$f(x_1, ..., x_m) = \sum_{k=1}^{t} c_k x_1^{a_{1k}} x_2^{a_{2k}} ... x_m^{a_{mk}} \tag{2.2}$$

where $c_j \geq 0$ and a_{ij} are real. When there is only one term in the sum, f is called a monomial function. A geometric program can be converted into a convex optimization problem by changing variables and considering the logs of the function involved. The guaranty of a globally optimal solution is provided by the convexity property [19].

2.2 Solutions for Geometric Programming

The problem as presented in (2.1) is not a convex optimization problem. However, it can be transformed into a convex optimization problem by introducing variables $y_i = log\, x_i$ for all $i = 1, \ldots, n$. For any monomials g defined in (2.1),

$$g(x_1, x_2, \ldots, x_n) = g(y_1, y_2, \ldots, y_n) = ce^{y_1\alpha_1}e^{y_2\alpha_2}\ldots e^{y_n\alpha_n} = e^{\tilde{a}^T\tilde{y}+b} \tag{2.3}$$

where $\tilde{a}^T = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and $b = log\, c$. Similarly, for any posynomial defined in (2.2),

$$f(x_1, x_2, \ldots, x_n) = f(y_1, y_2, \ldots, y_n) = \sum_{k=1}^{m} c_k e^{y_1\alpha_{1_k}}e^{y_2\alpha_{2_k}}\ldots e^{y_n\alpha_{n_k}} = \sum_{k=1}^{m} e^{\tilde{a_k}^T\tilde{y}+b_k} \tag{2.4}$$

where $\tilde{a_k}^T = [\alpha_{1_k}, \alpha_{2_k}, \ldots, \alpha_{n_k}]$ and $b_k = log\, c_k$ for all $k = 1, 2, \ldots, m$.
The geometric programming in (2.1) is rewritten as an optimization in the variable $y \ni \Re^n$ as

$$\begin{aligned} &\text{minimize } \textstyle\sum_{k=1}^{m} e^{a\tilde{0}_k{}^T\tilde{y}+b_{0_k}} \\ &\text{such that } \textstyle\sum_{k=1}^{m} e^{a\tilde{i}_k{}^T\tilde{y}+b_{i_k}} \leq 1;\, i = 1, \ldots, m \\ &e^{a\tilde{i}_k{}^T\tilde{y}+b_{i_k}} = 1;\, i = 1, \ldots, p \end{aligned} \tag{2.5}$$

Taking the logarithms of the objective and constraints functions in (2.5),

$$\begin{aligned} &\text{minimize } f_0(y) = log\left(\textstyle\sum_{k=1}^{m} e^{a\tilde{0}_k{}^T\tilde{y}+b_{0_k}}\right) \\ &\text{such that } f_i(y) = log\left(\textstyle\sum_{k=1}^{m} e^{a\tilde{i}_k{}^T\tilde{y}+b_{i_k}} \leq 1\right);\, i = 1, \ldots, m \\ &h_i(y) = \tilde{a_{i_k}}^T\tilde{y} + b_{i_k} = 1;\, i = 1, \ldots, p \end{aligned} \tag{2.6}$$

In the above formulation functions f_i are convex functions in their argument y. h_i is an affine function. Thus, (2.6) is a convex optimization problem.

The formulation as presented in (2.6) is equivalent to the standard geometric programming in (2.1) and can be solved using algorithms developed for general convex optimization problems, such as interior point methods. Details of these methods can be found in [16]. The general idea behind these methods is to start from a feasible solution of the optimization problem, and iteratively update the feasible solution by taking steps in the directions that minimize the objective function.

Since the objective function in (2.6) is convex, the interior-point methods are guaranteed to find the global optimal solution within a given degree of accuracy and do so independently of the initial starting point. In addition, these methods can solve large scale problem efficiently. An example to illustrate the above is given in Appendix B.

Chapter 3

Comparators

Comparators are basic building blocks in many analog and mixed-signal systems. Design and synthesis of comparator architectures largely remains an analog designers art. In this chapter, we present a systematic methodology for designing and comparing comparator architectures using constrained optimization. The circuits are optimized for power, under constraints on gain, dimensions, and delay. Since most of the objective functions and constraints are either monomials or posynomials, geometric programming is chosen for optimization. Comparison with Cadence UMC 0.18 μm results show that this is a fast and efficient method for the optimization of a mixed-signal circuit. By using geometric programming, we are sizing the transistors in three different comparator architectures such as op amp based, static latch based and dynamic latch based comparators. Here sizing can be done for minimum power for the given constraints, like gain, delay, dimension and area. Unlike operational amplifier, which is basically a linear circuit, applying geometric programming in comparator is having its own challenges like forming posynomial equations, making proper approximations, deciding the proper dimensions etc. Another difficulty while designing comparators is the latch part, the latch part should be designed such a way that the current through the latch and sizes of the transistors will lead to a positive feedback. Making these latch constraints as posynomial equations is a more challenging one. Then we are systematically comparing the three comparator architectures for complexity, power, speed and area. Since in this work, the comparison of three architectures is done with the same algorithm and with same specifications and constraints, the comparison is a more justified one, more than that no literature compares different comparators architectures with an automation algorithm.

3.1 Opamp based Comparator

This section describes the geometric programming model for an OpAmp-based comparator, the constructions of its objective function, constraints and simulation of optimization results.

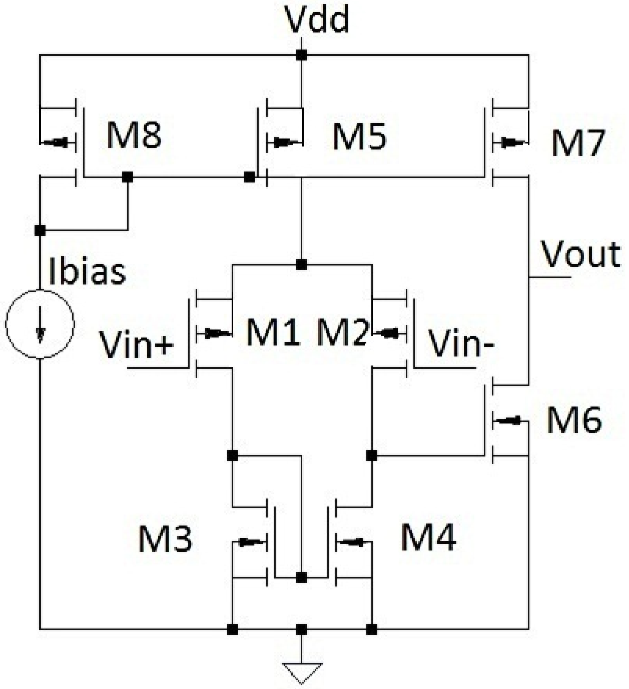

Figure 3.1: An OpAmp-based comparator.

The comparator satisfies the requirement that it should result in a logic high voltage if $V_{in}^{+} > V_{in}^{-}$, and a logic low otherwise. The limitations of using an OpAmp as a comparator are that its input voltage range is limited by the common mode input range of the OpAmp, and a slow response time, which is limited by the slew rate [17]).

3.1.1 GP Model

For improving the accuracy of geometric programming, we are using a slightly variant model based on [18], as shown below,

$$\begin{aligned} I_{Dn} &= k_1 W^{a1} L^{b1} V_{ov}^{c1} & I_{Dp} &= k_2 W^{a2} L^{b2} V_{ov}^{c2} \\ V_{ovn} &= k_3 I_D^{a3} W^{b3} L^{c3} & V_{ovp} &= k_4 I_D^{a4} W^{b4} L^{c4} \\ g_{mn} &= k_5 I_D^{a5} W^{b5} L^{c5} & g_{mp} &= k_6 I_D^{a6} W^{b6} L^{c6} \\ g_{dsn} &= k_7 I_D^{a7} L^{c7} & g_{dsp} &= k_8 I_D^{a8} L^{c8} \end{aligned} \tag{3.1}$$

Where a_1, b_1, c_1,... etc are the constants estimated by the Least Square Error (LSE) fitting method. GP formulation for open loop comparator is the same as that proposed by M.Henderson et. al [2]. This work uses the same objective functions and constraints with the above mentioned transistor model. The difference in our model is the absence of compensation network.

3.1.2 Simulation Results:

We now discuss the simulation results for the optimization of an open loop comparator. Simulations were done in MATLAB, utilizing the *ggplab* package [21]. The comparator of Fig. 3.1 was optimized for minimum power consumption, under six independent sets of constraints on the voltage gain, and propagation delay. table 3.1 shows the results of the optimizations. The first row of the table shows the optimized value of the power consumption (which was the objective function). The second and third rows show the constraints imposed on gain and delay, during the optimization. The next fifteen rows show the optimized design variables. The next two rows show the values of the gain and delay obtained by the optimization algorithm. And the last four rows give the values of the power consumption, the voltage gain, propagation delay, and estimated area, obtained from circuit simulations for the optimized design variables.

In the first two columns, the gain was kept constant and the Delay was decreased. This resulted in the (W/L) ratio of M_1, M_3, M_5, M_6, M_7 and M_8 increasing, because for a lower delay, the current drive required will be higher, and thus higher (W/L) ratios are necessary. The increase in overdrive voltages and bias current can also be explained by the same reason. The higher current drive is also reflected in the power consumption being higher in the second column.

In the first and third columns, the delay was kept constant and the gain was varied. When the gain is increased, the channel length of M_1, M_2, M_3, M_4, M_5 and M_7 increased. For maintaining the delay constant the width of M_1, M_2, M_3, M_4, M_5 and M_7 also increased. The PMOS overdrive

Objective Function						
Power (μW)	7	9	7	15	146	243
Constraints						
Gain (dB)	70	70	84	75	60	54
Delay (ns)	50	25	50	15	10	5
Design Variables						
W1=W2 (μm)	1.62	1.89	1.62	2.07	2.97	2.70
W3=W4 (μm)	0.99	1.26	0.99	1.46	4.41	3.96
W5 (μm)	1.17	0.99	1.17	0.99	0.99	0.99
W6 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W7 (μm)	0.99	0.99	0.99	0.99	1.35	0.99
W8 (μm)	2.25	1.08	2.25	0.99	0.99	0.99
L1=L2 (μm)	3.15	2.79	3.24	2.79	3.42	2.70
L3=L4 (μm)	5.04	4.05	5.04	3.60	3.87	3.60
L5 (μm)	0.99	0.63	0.99	0.63	0.45	0.45
L6 (μm)	2.79	1.26	2.79	0.81	0.27	0.18
L7 (μm)	0.99	0.63	0.99	0.63	0.45	0.45
L8 (μm)	0.99	0.63	0.99	0.63	0.45	0.45
Vodn (V)	0.10	0.10	0.10	0.11	0.18	0.23
Vodp (V)	0.11	0.14	0.11	0.15	0.48	0.69
Ibias (μA))	1.00	1.00	1.00	1.33	13.35	26.70
Matlab Results						
Gain (dB)	86	81	85	77	62	56
Delay(ns)	50	25	50	15	10	5
Cadence Simulation Results						
Power (μW)	6	9	6	12	135	212
Gain (dB)	84	79	83	76	62	58
Delay (ns)	62	28	59	17	12	7
Area (μs^2)	17	13	18	14	29	25

Table 3.1: Open Loop Comparator Simulation Results

voltage changed to obtain minimum possible power consumption. Increasing the gain does not directly cause any changes in current (unless otherwise it is violating dimension constraints;), hence there is not much change in current and thus in power. In this comparison we note an interesting point that 84 dB is the maximum possible gain for a 50 ns delay. Increasing the gain further cause the optimization routine to not provide a converged solution. This is because the channel lengths of all transistors were limited to $5\mu m$ by dimension constraints.

In the next three columns, we further decrease the delay. Reduction in delay increases the (W/L) of transistors, the overdrive voltages and the bias current, thus in turn increasing the power. In last two columns, we can see that the increase in power is exponential with reduction in the delay. This is because reduction in delay reduces the sizes of the transistors, but for maintaining the gain constant, the delay cannot be reduced below a point, because the bias current must be increased to reduce the output resistance. Increase in total current due to reduction in length and increase in biasing current increases the power exponentially.

The last four rows of table 3.1 are CADENCE-Spectre simulations, done on 0.18 μm technology MOSFETs with UMC model parameters, with the geometry parameters and bias voltages obtained from the optimization algorithm. Thus the Cadence results are primarily useful in determining whether the MOSFET transistor model used in the GP-optimization was reasonably accurate or not, compared to the more accurate BSIM3v3 model utilized in Cadence. Comparison of the Cadence results with the Matlab results in table I indicates that the transistor model used is very reasonable.

3.2 Static Comparator

A static comparator, shown in fig. 3.2, can be divided into two stages: The input stage and the decision stage. The input stage converts the input voltages to currents, which are then used to drive the decision stage. The decision stage is a cross coupled latch circuit which has two stable states; the positive feedback of the cross-coupling yields high speed switching [22].

3.2.1 GP Model

The problem is formulated separately for preamplifier and decision stage, because doing it a whole is quite complicated and it is not necessary to do so.

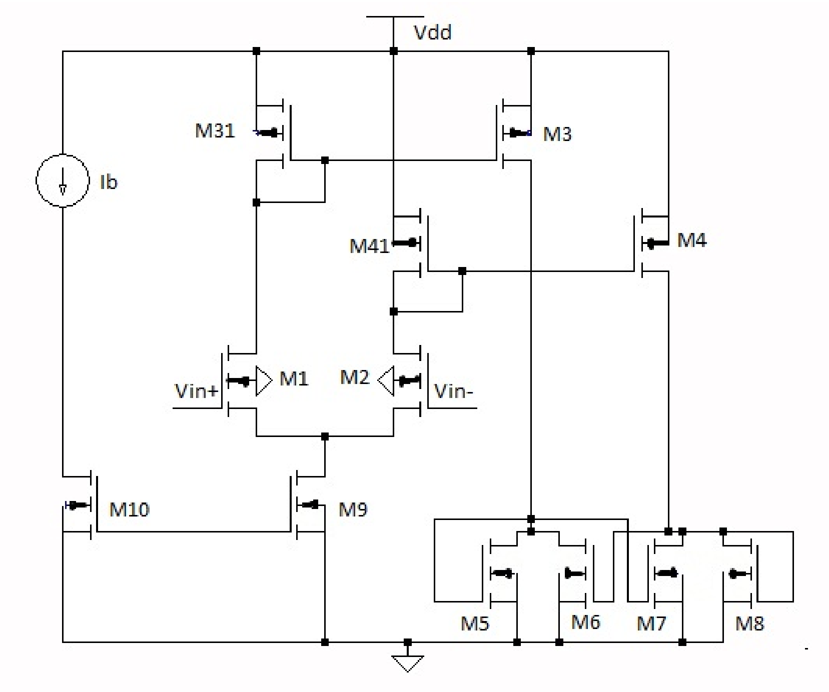

Figure 3.2: Schematic diagram of Static comparator.

Preamplifier

Preamplifier is a differential amplifier with diode connected loads. Transistors M_1, M_2, M_{31}, M_{41} and M_9 constitute preamplifier in Fig 3.2.

Objective function:

The function to be minimized is power dissipation. The power taken by the preamplifier is given by

$$P = V_d(I_1 + I_2) \tag{3.2}$$

where I_1 and I_2 are the currents in M_1 and M_2, respectively. P is a posynomial function of the design parameters, and hence we can use it as the objective to be optimized.

Constraints:

Dimension, symmetry, biasing, open loop gain and slew rate are considered as the constraints. These constraints define the design space for the optimum objective function. The mathematical models for the above mentioned constraints are derived in this section.

Dimension Constraints:

The following five equality constraints would satisfy the symmetry and matching conditions.

$$W_1 = W_2, W_3 = W_4, L_1 = L_2, L_3 = L_4 \tag{3.3}$$

The current mirror transistors M_9 and M_{10} must have the same length

$$L_9 = L_{10} \tag{3.4}$$

The five equality constraints in (3.3) and(3.4) have monomial expressions on the left and right hand sides, hence are readily handled by geometric programming.

Bias Constraints:
To make the transistors M_1, M_2, M_{31} and M_{41} remain in saturation region, for all possible values of the input common-mode voltage and the output signal swing, bias constraints are applied. The satisfaction of symmetric and matching conditions are assumed to derive bias constraints.
Transistors M_{31} and M_{41} are always in saturation region. For the transistors M_1 and M_2 to be ib saturation region,the condition to be satisfied is

$$k_3 I_1^{a3} W_3^{b3} L_3^{c3} \leq V_{cm,min} - V_{SS} - V_{tp} - V_{tn} \tag{3.5}$$

Gain and Delay constraints:
Open loop gain: Open loop gain for preamplifier can be derived as [22]

$$A_v = \frac{g_{m1}}{g_{m3}} \tag{3.6}$$

can easily fit in a monomial model as

$$\frac{k_5 W_1^{a5} L_1^{b5} I_1^{c5}}{k_6 W_3^{a6} L_3^{b6} I_3^{c6}} \leq A_v \tag{3.7}$$

Delay: Delay can be derived as [39]

$$t_{pd} = 0.693 C_L (\frac{1}{g_{m3}} || r_{o3} || r_{o1}) \tag{3.8}$$

where $C_L = (C_{dp1} + C_{dn1}) + (C_{gp3} + C_{gn3}) + C_W$ is a posynomial. Delay expression is a perfect gp constraint.

Decision Stage

A pair of PMOS transistors and a cross coupled latch, work as decision stage. Latch speeds up the decision operation, by employing positive feedback.
Objective Function:
The objective function is power and can be derived as
power:

$$Power = V_{dd}(I_{d3} + I_{d4}) \tag{3.9}$$

Constraints:
Here, dimension, open loop gain and propagation delay are considered as constraints.
Dimension constraints:
The sizes of M_3 and M_4 should be proportional to M_{31} and M_{41} of preamplifier. The ratio will be according to the current requirement of decision stage.

$$\frac{W_3}{L_3} = k\frac{W_{31}}{L_{31}}\frac{W_4}{L_4} = k\frac{W_{41}}{L_{41}} \tag{3.10}$$

and

$$\frac{W_4}{L_4} = k\frac{W_{41}}{L_{41}} \tag{3.11}$$

Where k = $\frac{I_{d31}}{I_{d3}}$.
The sizes of latch transistors are as follows:

$$\frac{W_5}{L_5} = \frac{W_8}{L_8} \tag{3.12}$$

$$\frac{W_6}{L_6} = \frac{W_7}{L_7} \tag{3.13}$$

$$\frac{W_6}{L_6} \leq \frac{W_5}{L_5} \tag{3.14}$$

and

$$\frac{W_7}{L_7} \leq \frac{W_8}{L_8} \tag{3.15}$$

Constraints (3.10), (3.11), (3.12) and (3.13) are equality constraints and monomials. Constraints (3.14) and (3.15) are posynomials. The constraints (3.14) and (3.15) are necessary to avoid infinite delay in latch because of large negative resistance.

Gain and Delay constraints:

Open loop gain:

$$A_v = \frac{g_{m3}}{g_{m5} - g_{m6}} \tag{3.16}$$

can try to fit as a posynomial with some modification as

$$A_{v1} \leq \frac{g_{m5} + g_{m6}}{g_{m5} - g_{m6}} \tag{3.17}$$

and

$$A_{v2} \leq \frac{g_{m3}}{g_{m5} + g_{m6}} \tag{3.18}$$

but still these are not perfect gp constraints (a perfect gp constraint should not have a posynomial in RHS, and it should not have a minus sign). Hence we make following approximations

$$A_{v2}(g_{m5} + g_{m6}) \leq g_{m3} \tag{3.19}$$

and

$$(A_{v1} - 1)g_{m5} = (A_{v1} + 1)g_{m6} \tag{3.20}$$

So constraint (3.16) is converted into two constraints (3.19) and (3.20).The total gain is the product of eqns 3.19 and 3.20, i.e., $A_v = A_{v1} * A_{v2}$. To get higher gain, either A_{v1} or A_{v2} or both has to be increased. According to equation 3.19 increasing A_{v2} increases the size of M_3 which results in increase in area and power. The other option to increase the gain is to increase A_{v1}. Since A_{v1} contains the positive feedback, it is easier to increase the gain, but positive feedback leads to exponential current to flow through the latch which may make the design unpredictable. In normal practice M_5 and M_6 are considered to be of the same size. Since GP is a equation based modeling, assuming M_5 and M_6 to be of the same size leads to infinite gain which would halt the optimization loop. Equation 3.20 provides a small difference between g_{m5} and g_{m6} and provides for the required gain.

To escape from the exponential behavior of positive feedback, a small design space is created (by choosing the upper and lower limits of W's and L's for the transistors which constitutes the latch). Within the design space the behavior of g_{m6} and g_{m5} w.r.t W's and L's are predicted with a sample set. With that information the gain is calculated.

Propagation delay :

Since the transconductance of latch transistors changes with the current, sizing the transistors for the delay with small signal model will be inaccurate. Hence We use large signal analysis, with a predefined current model as the input to the latch. The current input of the latch has an exponential nature. Giving an exponential input to the latch makes it more unstable and difficult to realize. So we are approximating the current input of the latch as a linear ramp expression.

The resultant equations are a form of Non-linear coupled differential equations. We are applying a new decomposition method known as Adomian Decomposition Method (ADM) to solve these equations [23, 24, 25, 26]. The resultant output equations for the latch are give as

$$\begin{aligned} v_o^+(t) &= \frac{C_2}{C_L}\frac{t^2}{2} - \frac{\beta_1}{C_L}\frac{C_2^2}{C_L^2}\frac{t^5}{120} - \frac{\beta_2}{C_L}\frac{C_2^2}{C_L^2}\frac{t_1 t^3}{12} \\ &- \frac{\beta_2}{C_L}\frac{C_2^2}{C_L^2}\frac{t^5}{30} + \frac{4\beta_2}{C_L}\frac{C_2^2}{C_L^2}\frac{t_1 t^4}{20} - \frac{\beta_2}{C_L}\frac{V_{sat}^2 t}{2} \\ &- \frac{1}{3}\frac{\beta_2}{C_L}\frac{C_2 t_1 V_{sat} t^2}{C_L} + v_{th} \end{aligned} \tag{3.21}$$

and

$$\begin{aligned} v_o^-(t) &= \frac{C_2 t_1 t^1}{C_L} - \frac{C_2}{C_L}\frac{t^2}{2} + V_{sat} - \frac{\beta_2}{C_L}\frac{C_2^2}{C_L^2}\frac{t^5}{120} \\ &- \frac{\beta_1}{C_L}\frac{C_2^2}{C_L^2}\frac{t_1 t^3}{12} - \frac{\beta_1}{C_L}\frac{C_2^2}{C_L^2}\frac{t^5}{30} + \frac{4\beta_1}{C_L}\frac{C_2^2}{C_L^2}\frac{t_1 t^4}{20} \\ &- \frac{\beta_1}{C_L}\frac{V_{sat}^2 t}{2} - \frac{1}{3}\frac{\beta_1}{C_L}\frac{C_2 t_1 V_{sat} t^2}{C_L} + v_{th} \end{aligned} \tag{3.22}$$

Where C_2 is the slope of the ramp signal, C_L is the load capacitance, t_1 is the time in which the ramp is attaining its maximum value, V_{sat} is the saturation voltage and v_{th} is the threshold voltage. More details regarding Adomian Polynomial based decomposition method is added in Appendix C.

Due to the constraints in posynomial formation, we cannot directly use eq. (3.21) and (3.22) in a GP model. Approximations are done and some of the terms are eliminated due to its insignificance and the resultant equation for delay is given in eq. (3.23).

$$t = \frac{C_L}{a_1} ln\left(\frac{a_1^2.\Delta v_0 + a_1 I_{slope} + 2I_{slope}C_L}{a_1^2.\Delta v_{in} + a_1 I_{slope} + 2I_{slope}C_L}\right) \tag{3.23}$$

where $a_1 = (\mu n C_{ox} W_5 - \mu n C_{ox} W_6)$, I_{slope} is the slope of the input ramp current and C_L is the load capacitance which includes Drain diffusion capacitance of M_3, M_5 and M_6 and the input capacitances of output stage. The value of a_1 will be negative to impose positive feedback.

The comparison between the proposed delay model and the CADENCE UMC $0.18\mu m$ simulation is shown in fig 3.3. It can be seen that the proposed delay model more or less follows the cadence

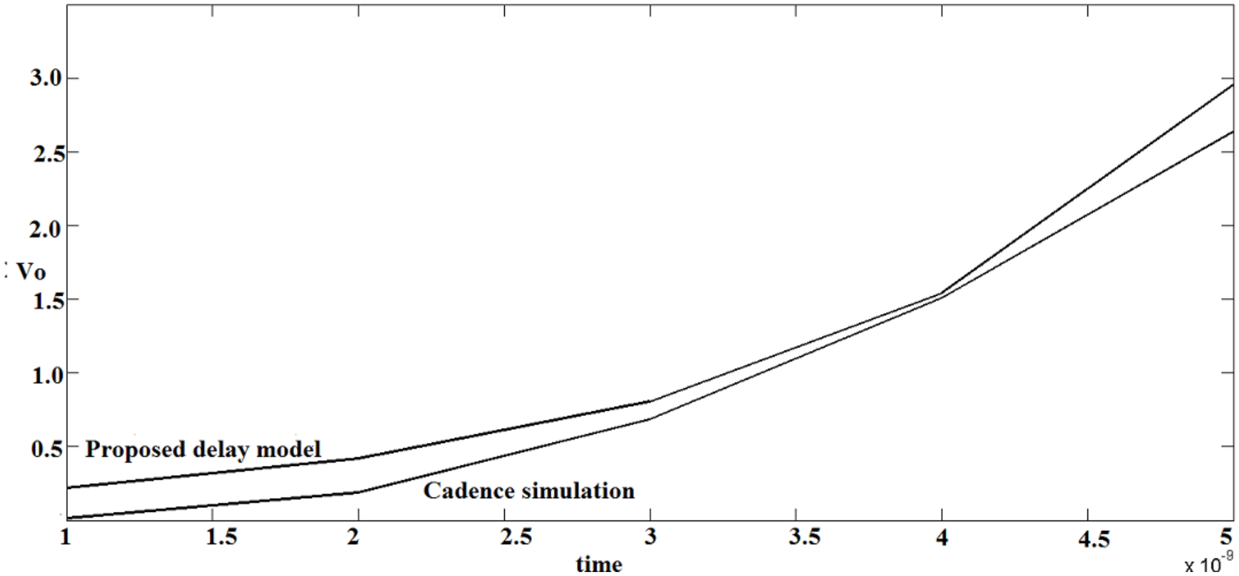

Figure 3.3: comparison between the proposed delay model and CADENCE simulation.

simulation result.

3.2.2 Simulation Results

Given below are the simulation results of a Static latch based comparator using geometric programming. The simulations were done in MATLAB, utilizing the $ggplab$ package.

table 3.2 shows the comparison of the objective function with six sets of constraints, for a static latched comparator. The first row of the table shows the optimized value of the objective function (namely, the dc power consumption). The second and third rows show the constraints (voltage gain and delay) imposed during the optimization. The rest of the rows show the optimized values for all the variables, Matlab results and Cadence simulation results. In the first two columns the gain was kept constant and the delay was decreased. When the delay is decreased, the (W/L) ratios of M_3,M_4,M_{31} and M_{41} get adjusted such that the current through the latch is high.

As the current through the latch is directly proportional to the (W/L) ratio of M_3 and inversely proportional to the (W/L) ratio of M_{31}, reducing the delay results in increase in the (W/L) ratio of M_3 and decrease in the (W/L) ratio of M_{31}. The other way to decrease the delay is to enhance the positive feedback by adjusting the value of a_1 in (3.39). GP uses both the above mentioned ways to keep the power optimum. To keep the gain constant the length of M_5 and M_7 get adjusted slightly. Further reduction in delay increases the bias current, thereby reducing the length of M_5, M_6, M_7 and M_8. The same argument holds good for the fourth and sixth columns. In the first and

Objective Function						
Power (μW)	5	7	5	18	35	35
Constraints						
Gain (dB)	70	70	84	60	54	60
Delay (ns)	50	25	50	10	5	5
Design Variables						
W1=W2 (μm)	1.17	1.17	1.17	1.17	1.17	1.17
W31=W41 (μm)	1.71	1.35	1.71	0.99	0.99	0.99
W3=W4 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W5=W8 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W6=W7 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W9 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W10 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
L1=L2 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
L31=L41 (μm)	1.44	1.71	1.44	1.80	2.43	2.43
L3=L4 (μm)	4.23	3.78	4.23	2.70	1.80	1.80
L5=L8 (μm)	3.51	3.60	3.51	0.81	0.54	0.54
L6=L7 (μm)	3.69	3.69	3.78	0.90	0.63	0.72
L9 (μm)	0.54	0.54	0.54	0.54	0.54	0.54
L10 (μm)	0.54	0.54	0.54	0.54	0.54	0.54
Vodn (V)	0.10	0.10	0.10	0.12	0.17	0.17
Vodp (V)	0.12	0.13	0.12	0.23	0.37	0.37
Ibias (μA))	1.00	1.00	1.00	2.12	3.24	3.24
Matlab Results						
Gain (dB)	80	81	85	64	58	63
Delay (ns)	50	25	50	10	5	5
Cadence Simulation Results						
Power (μW)	5	7	8	17	32	32
Gain (dB)	79	80	84	62	58	61
Delay (ns)	38	25	41	9	4	4
Area (μs^2)	16	15	16	8	7	8

Table 3.2: Static Latch Comparator Simulation Results

third columns, the delay was kept constant and the gain was increased. It could be seen that all the values are same except the length of transistors M_5 and M_8. Though the gain considered for optimization is $A_v = \frac{g_{m3}}{g_{m5}-g_{m6}}$, the actual gain is $A_v = \frac{g_{m3}}{g_{m5}-g_{m6}+g_{ds3}+g_{ds5}+g_{ds6}}$. Since all the parameters depend on the length of the transistors, a negligible change in length will make a significant difference in gain. The fifth and sixth columns also show a similar trend. Since we are going for an approximation in gain which is significantly large, a post Matlab process is necessary to meet our specifications. So the design variables in table II are not exactly the design variables arrived from optimization tool, but are obtained from post Matlab process.

Since the variation in length of transistor M_5 is negligible and there are no major variations in other design variables, power and delay remains constant.

3.3 Dynamic Comparator

This section describes the dynamic comparator, its GP model and simulation results. The dynamic comparator shown in fig 3.4, has a preamplifier and a dynamic latch.

The basic difference between the static and dynamic comparator is the clock circuitry that the dynamic comparator employs. The clock circuitry prevents excessive power dissipation with slight increase in area.

A preamplifier is used to convert an input voltage to a current, and the current is mirrored to positive feedback latch by a pair of current mirror transistors $M31$ and $M41$. The positive latch regenerates its input voltage difference and gives the output as either high or low. The latch circuit operates as follows: In the reset phase V_{lat} is high and outputs are pre-charged to the gnd. The switch transistors $M7$ and $M8$ are off and prevents the current flow into the latch. When $Vlat$ goes low, regeneration phase starts and $M7$ and $M8$ are turned on. The differential currents are mirrored to latch. Because of the difference in the current, a voltage difference at the output exists, which makes the positive feedback latch to start regeneration and produce binary output.

3.3.1 GP Model

The construction of objective function and constraints are explained in this section.

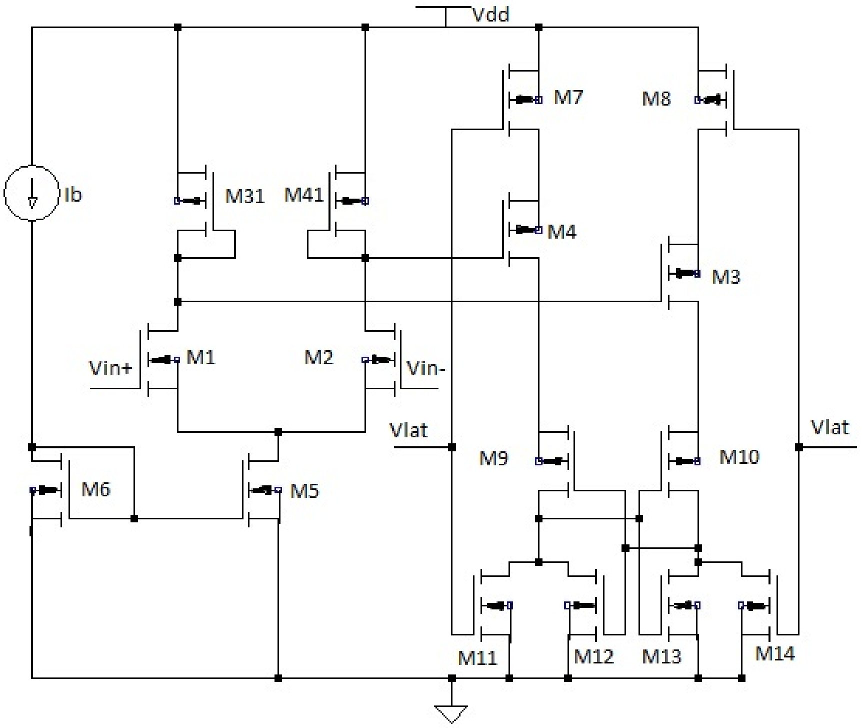

Figure 3.4: Dynamic comparator

Preamplifier

The problem formulation for preamplifier is similar to the problem formulation of preamplifier of a static comparator.

Dynamic Latch

Dynamic latch is nothing but a back to back inverter with a clock signal to avoid continuous current flow.
Objective Function:
Power dissipation is the objective function considered. Power dissipation includes static power dissipation during regeneration phase and dynamic power dissipation due to clock signals.

$$power = V_{dd}^2 F_{clk} C_L + V_{dd} I_{d4} \tag{3.24}$$

where C_L is the load capacitance which includes output capacitance and parasitic capacitances. F_{clk} is the clock frequency. I_{d4} is the current through the transistor M_4 during regeneration phase, which would be zero during reset phase.

Constraints:

The major constraints are gain and delay. Dimension, biasing and symmetry are also considered.

Dimension Constraints:

For proper differential operation transistor M_3 and M_4 must be identical. As we are using back to back inverter for latching , transistors M_9 must be identical to M_{10} and also M_{12} must be identical M_{13}. For avoiding clock skew, the transistors M_7, M_8, M_{11} and M_{14} must be identical. These conditions translates into the following constraints.

$$W_4 = W_5, L_4 = L_5 \tag{3.25}$$

$$W_7 = W_8 = W_{11} = W_{14}, L_7 = L_8 = L_{11} = L_{14} \tag{3.26}$$

$$W_9 = W_{10}, W_{12} = W_{13}, L_9 = L_{10}, L_{12} = L_{13} \tag{3.27}$$

Gain and Delay constraints:

DC Gain: DC gain of the latch can be expressed as

$$A_v = \frac{g_{m4}}{(g_{m12} + g_{gds4}) - \frac{g_{ds12}(g_{m9} - g_{ds12})}{g_d s12 + (g_{m9} - g_{ds12})}}; \tag{3.28}$$

Propagation delay:

Here we employ the same method as that in static latch comparator. The only difference being that the the input current is assumed to be a step signal. With this assumption and with large signal analysis, we get the following delay equation.

$$t = \frac{C_L}{a_1} ln \left(\frac{a_1 \Delta v_0 + 2I_{max}}{a_1 \Delta v_{in} + 2I_{max}} \right) \tag{3.29}$$

where $a_1 = (\mu n C_{ox} \frac{W_9}{L_9} - \mu n C_{ox} \frac{W_{12}}{L_{12}})$, I_{max} is the maximum value of the input current and C_L is the load capacitance which includes Drain diffusion capacitance of M_9,$M_{11}$and M_{12}. The value of a_1 will be negative to impose positive feedback.

The comparison between the proposed delay model and the CADENCE UMC $0.18 \mu m$ simulation is shown in fig 3.5, which indicates that the proposed delay model matches with the Cadence simulation results.

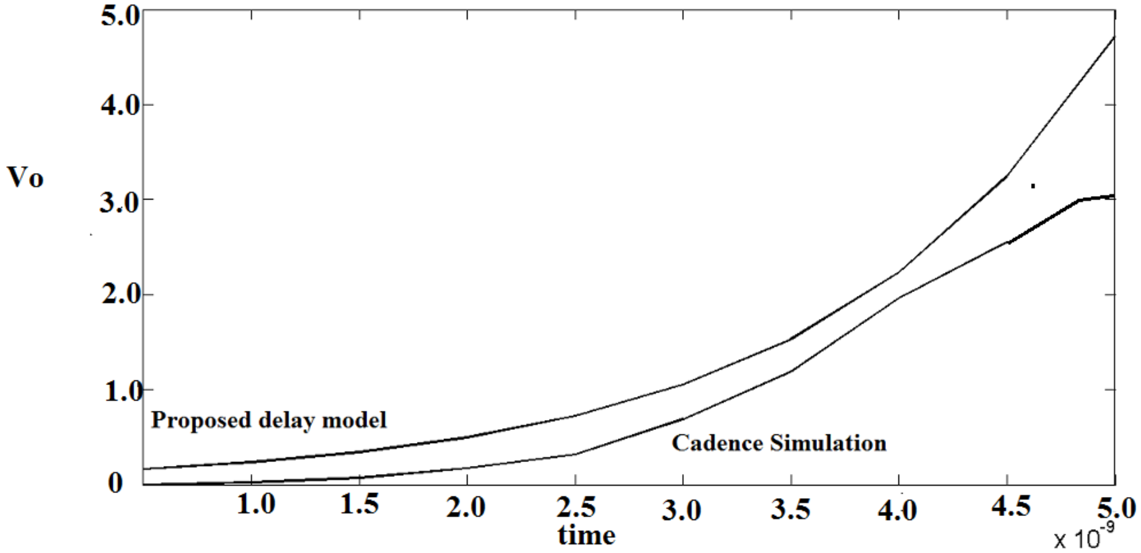

Figure 3.5: comparison between the proposed delay model and CADENCE simulation.

3.3.2 Simulation Results

Given below are the simulation results of the Dynamic latch based comparator using geometric programming. The simulations were done in MATLAB, utilizing the $ggplab$ package, and compared with Cadence 0.18 μm technology.

To have a fair comparison, the constraints are kept similar to that of a static latch based comparator, and the results are also same, but with minimum power.

In the first two columns, the gain was kept constant and the delay was decreased, this adjust the value of a_1 in (3.29) to get the required delay. Reduction in delay also reduces the channel length of M_{31}, M_3, M_9 and M_{12}. As a result the drive current would be higher and the parasitic capacitance would be lower. The fourth and sixth columns also show the same trend.

In the first and third columns the delay was kept constant and the gain was increased. Increasing the gain increases the channel length of M_{31} and M_{12} and decreases the channel length of M_3 and M_9. The same argument holds good for fourth and fifth columns.

The power dissipation in dynamic comparator is much lower than the static latch and op amp based comparator, because the dynamic latch circuit does not have static current. The power in dynamic latch circuit is only due to the charging and discharging of the output capacitor.

The last four rows show that our design variables gives closer results while simulating with Cadence 0.18 μm technology.

Objective Function						
Power (μW)	6	9	8	12	14	17
Constraints						
Gain (dB)	70	70	84	54	60	54
Delay (ns)	50	25	50	10	10	5
Design Variables						
W1=W2 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W31=W41 (μm)	1.08	1.17	3.60	1.26	1.35	1.44
W3=W4 (μm)	3.96	3.96	3.96	3.96	3.96	3.96
W7=W8 (μm)	1.98	1.98	1.98	1.98	1.98	1.98
W9=W10 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
W12=W13 (μm)	0.99	1.28	1.00	0.99	1.15	0.99
W5 (μm)	1.44	1.44	2.25	1.44	1.44	1.44
W6 (μm)	2.07	1.26	2.07	0.99	0.99	0.99
L1=L2 (μm)	5.04	5.04	2.25	5.04	5.04	5.04
L31=L41 (μm)	3.96	1.17	5.04	4.41	4.86	4.86
L3=L4 (μm)	1.98	1.53	1.08	1.44	1.62	1.62
L7=L8 (μm)	0.99	0.99	0.99	0.99	0.99	0.99
L9=L10 (μm)	1.08	0.18	0.99	0.18	0.18	0.18
L12=L13 (μm)	0.99	0.18	2.61	0.18	0.18	0.18
L5 (μm)	0.54	0.54	0.54	0.54	0.54	0.54
L6 (μm)	3.51	3.51	2.43	3.51	3.51	3.51
Vodn (V)	0.10	0.13	0.10	0.17	0.18	0.19
Vodp (V)	0.11	0.15	0.10	0.26	0.32	0.32
Ibias (μA))	1.00	1.00	1.00	1.06	1.32	1.53
Matlab Results						
Gain (dB)	74	73	85	56	62	56
Delay(ns)	50	25	50	10	10	5
Cadence Simulation Results						
Power (μW)	6	10	8	12	15	17
Gain (dB)	73	72	84	55	61	55
Delay (ns)	51	27	52	12	12	7
Area (μs^2)	35	26	40	28	30	31

Table 3.3: Dynamic Latch Comparator Simulation Results

Architecture	Complexity	Power	Speed	Gain	Area
Op amp based	Low	High	Low	Medium	Medium
Static Latch	Medium	Medium	High	High	Low
Dynamic Latch	Medium	Low	High	High	High

Table 3.4: Comparison between the comparator architecture

3.4 Comparison of Comparator Architectures

In this section we compare the three architectures for power, speed, area and complexity. Brief results are given in table 3.4.

By complexity we mean, the effort and time required to model the corresponding architecture into a geometric programming model. This includes converting ordinary equations into posynomials and monomials, and the corresponding approximations involved. From the perspective of complexity, we clearly see that the op amp based comparator is the better one. The equations in op amp based comparators can be directly written in a posynomial form without any approximations, which is not the case with static and dynamic comparators. Static and dynamic comparator's gain and delay equations need approximations so that they can be converted in to posynomial form.

The second comparison is based on power. For a given delay, dynamic latch based comparator consumes minimum power. The reasons are positive feedback and absence of continuous current through the latch circuit. Positive feedback provides minimum delay for static latch based comparator and dissipates medium power. For the same delay, op amp based comparator requires more current drive, hence dissipates more power.

Using similar arguments (positive feedback), we conclude that static and dynamic latch based comparators are faster than op amp based comparators.

For a given area, delay and power, the static and dynamic comparators give higher gain compared to op amp based comparators.

Dynamic comparators employ clocking circuitry and therefore occupy a large area. Static and op amp based comparators use low number of transistors, thereby occupy less area. Since the static comparators use minimum sized transistors, it uses up the least area amongst the three. Comparators are the most integral part of any analog to digital converters (ADCs). In many applications that require ultra low power, high speed and ultra high portability, ADCs are extensively used. Naturally comparators which are used in ADCs should fit into these specifications. Nowadays ADCs that require ultra low power with medium size and high speed find immense usage in biomedical applications [27,28]. According to the comparison made earlier, dynamic comparator perfectly fits the bill (refer table 3.4).

The other major area where ADCs are extensively used is sensor networks [28]. Sensor networks require medium power, less area and high speed. Based on the above comparisons static latch comparators are the best choice.

This chapter presented the design and comparison of open loop comparator, static latch based comparator and dynamic latch based comparators using constrained optimization. Since most of the objective functions and constraints are either monomials or posynomials, geometric programming was used for optimization. Geometric programming always gives global optimum values, so that if the given objective functions and constraints are properly defined, we get globally optimum values. The design of a comparator, using this method, is seen to be a very fast, efficient, and accurate optimization technique. Comparison with Cadence UMC 0.18 μm technology simulation results show good agreement. This method thus provides a new technique for the optimization and synthesis of analog and mixed-signal circuits.

Chapter 4

A novel Capacitor Array based Digital to Analog Converter

We present a novel capacitor array digital to analog converter(DAC) architecture. This DAC architecture replaces the large MSB (Most Significant Bit) capacitor of the conventional capacitor array DAC with 3 unit capacitors, thus reducing the area, power dissipation and the settling time. The novel DAC does not demand an additional reference voltage, nor an additional switching circuit. The architecture is verified by drawing a layout and doing post layout simulations using a CADENCE UMC180 technology. An 8-bit DAC with this architecture occupies 26 μm^2 area, consumes 6.34 nW at 3 V and has a settling time of 69 ns, which makes it smaller, faster, and less power consuming than a conventional capacitor array DAC. In UMC 180 nm technology, there are built-in NMOS(nmos3V) and PMOS(pmos3V) devices which can be operated at 3 V. This work employs those devices.

The new architecture is shown in Fig 4.1. A conventional capacitor array DAC uses a total capacitance of $2^n C_0$ for converting an n bit digital input into its equivalent analog value, where C_0 is the unit capacitance value. The new architecture uses only $2^{n-1} C_0$ for converting an n-bit digital input, and additional $3C_0$ capacitors. Thus the saving in the capacitance is $2^{n-1} C_0 - 3C_0$. For large n, this amounts to a significant saving in chip area. Also, since the settling time of the DAC is proportional to the largest capacitance used, the settling time of the proposed architecture is also reduced. The power consumption of the DAC depends upon the switching schemes used [29]. The switching schemes employed in conventional architectures can be used for this new architecture. Since the power consumption is directly proportional to the total capacitance, power consumption in this architecture is less than the conventional DAC architecture, for any given switching scheme. Moreover, in this architecture, the LSB switches from GND to a value less than VDD, hence the switching power will be less than the corresponding switching in conventional circuit.

For a higher number of bits, the saving in power, area and settling time are also higher. In a conventional architecture, optimization is done by changing the switching schemes, which require more clock phases and switches and thus increase the area and reduce speed. Since the new architecture uses a conventional switching scheme, there is no sacrifice in either area or speed. If we employ the other switching schemes in this architecture, power consumption can be reduced further.

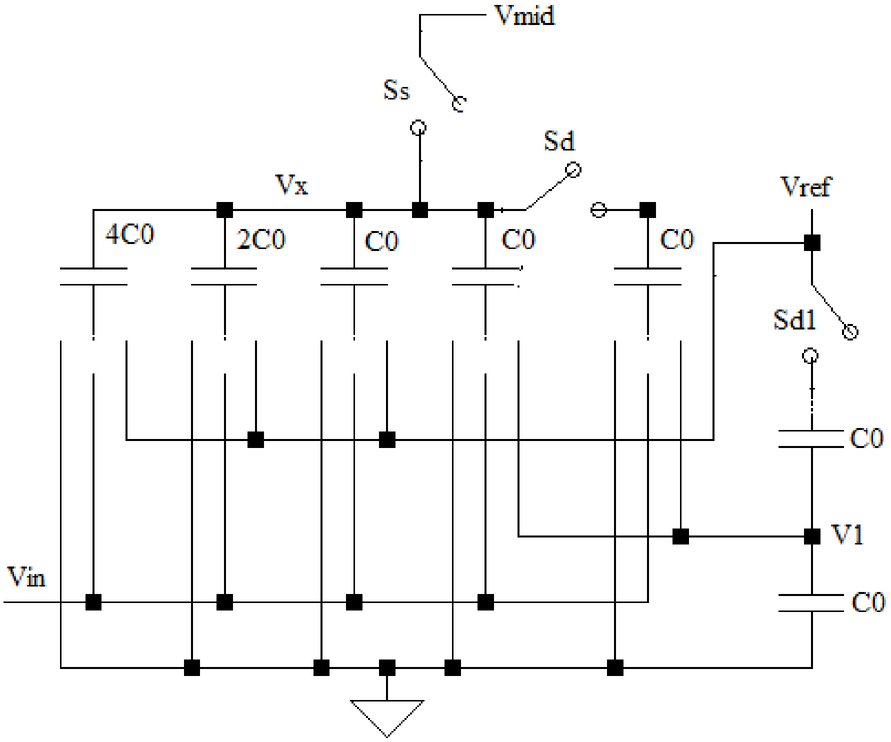

Figure 4.1: Novel 4-bit capacitor array DAC Architecture.

4.1 Working Principle

This section explains the difference between the conventional capacitor array DAC and the novel architecture, and also explains how the $3C_0$ capacitors replace the MSB capacitor. In an n-bit conventional capacitor array DAC, which requires $2^n C_0$, whenever an input bit is 1, the corresponding capacitor's bottom plate would be connected to V_{ref} and all other capacitors would be connected to ground. The difference between an n bit and (n-1) bit DAC in terms of output voltages is that an (n-1) bit DAC generates voltages in the range of $\frac{mV_{ref}}{2^{n-1}}$ where m = 0, 1, 2,...,n-2, whereas the n-bit DAC generates voltages in the range of $\frac{mVref}{2^n}$, where m = 0,1,2...,n-1. So the output voltage of an (n-1) bit DAC is a subset of the output of an n bit DAC. Hence instead of an n-bit DAC with the LSB as zero, we could use an (n-1) bit DAC by ignoring the LSB bit. For example, for a 4-bit DAC the output voltages for the input sequence of 0000 to 1111 are 0, $\frac{V_{ref}}{16}$, $\frac{2V_{ref}}{16}$, $\frac{3V_{ref}}{16}$, $\frac{4Vref}{16}$,. . . , $\frac{15V_{ref}}{16}$. whereas the output voltages for a 3-bit DAC, for the input sequence 000 to 111 are 0, $\frac{Vref}{8}$, $\frac{2Vref}{8}$, ..., $\frac{7V_{ref}}{16}$. Half the number of output voltages of a 4-bit DAC (like $\frac{2V_{ref}}{16}$ = $\frac{V_{ref}}{8}$, and $\frac{4V_{ref}}{16} = \frac{2V_{ref}}{8}$) could be generated using a 3-bit DAC.

In such a scheme, when LSB = 1, we need extra capacitors to generate the other n/2 outputs. In a conventional capacitor array DAC the MSB capacitor does this job. In the novel architecture we use 3 unit capacitors to generate those outputs. The working principle of the novel DAC can be explained as follows.

During the sampling phase, the bottom plates of all the capacitors are connected to V_{in} and the top plates are connected to V_{mid}. The voltage across the capacitors then is

$$V_x = V_{mid} - V_{in} \tag{4.1}$$

During this phase, switch S_d is ON and S_{d1} is OFF (see Fig. 4.1). When the LSB bit is high and all other bits are low, switches S_{d1} and S_d are both on and the last two unit capacitors are connected to V_1. The equivalent circuit is as shown in Fig. 4.2. The corresponding analog output will be

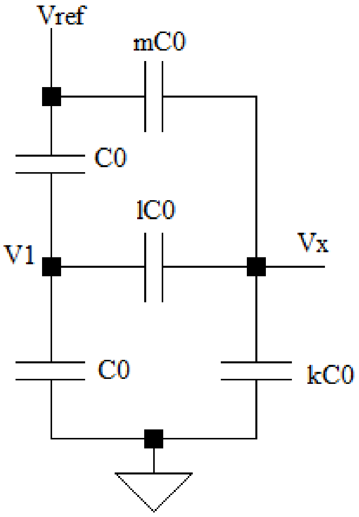

Figure 4.2: Generalized equivalent circuit for any bit pattern.

$$\begin{aligned} V_x &= V_{mid} - V_{in} \\ &+ \frac{(2m + lm + l)}{2(m + l + k) + l(m + k))} V_{ref} \end{aligned} \tag{4.2}$$

where m is the sum of capacitances connected to V_{ref}, and is given as

$$m = 2^{n-2} D_{n-1} + 2^{n-3} D_{n-2} + \cdots + D_1 \tag{4.3}$$

k is the sum of capacitances connected to ground:

$$k = 2^{n-2} \bar{D}_{n-1} + 2^{n-3} \bar{D}_{n-2} + \cdots + \bar{D}_1 + \bar{D}_0 \bar{S}_d \tag{4.4}$$

and

$$l = D_0 + D_0 S_d \tag{4.5}$$

where n is the number of bits, $D_{n-1} \ldots D_0$ are the digital inputs, $\bar{D}_{n-1} \ldots \bar{D}_0$ are their complements, and S_d is 1 when the switch S_d is closed, and it is zero when the switch is open. By substituting the values of l,m,k, equation (4.2) can be reduced to

$$V_x = \left(\frac{2^{n-2} D_{n-1} + 2^{n-3} D_{n-2} \ldots D_1}{2^{n-1}} + \frac{D_0}{2^{n-1}(1 + D_0)} \right) V_{ref} \tag{4.6}$$

Note that switches S_d and S_{d1}, are controlled by D_0. Switch S_d provides an extra capacitance, necessary when the LSB is HIGH. Switch S_d takes care that the node V_1 is charged accordingly. For example, if the input bit sequence is 1010, then $l = 0$, and $V_x = V_{mid} - V_{in} + \frac{m}{m+k} V_{ref} = V_{mid} - V_{in} + \frac{5}{8} V_{ref}$, which is the output for a conventional 3-bit DAC with input bits as 101. When the input is 1011, the equivalent circuit would be as shown in Fig. 4.3. For which $l = 1$ and $V_x = V_{mid} - V_{in} + \frac{(2m+lm+l)}{2(m+l+k)+l(m+k))} V_{ref} = V_{mid} - V_{in} + \frac{11}{16} V_{ref}$.
As the above discussion explains, V_x in Fig 4.1 is the analog equivalent voltage of the digital input signal, thus demonstrating that this circuit is, in fact, a digital-to-analog converter.

4.2 Simulation results

The circuit was laid out and simulated in a 180nm technology, with a V_{DD} of 3 V and a unit capacitor of 100 fF. The layout for an 8-bit DAC is shown in Fig 4.3. Post-layout simulation results are shown in Fig 4.4: A monotonically decreasing digital bit sequence was applied, and the analog output plotted as a function of time.

The magnified version of the simulation results with trace markers is shown in Fig 4.6. From the

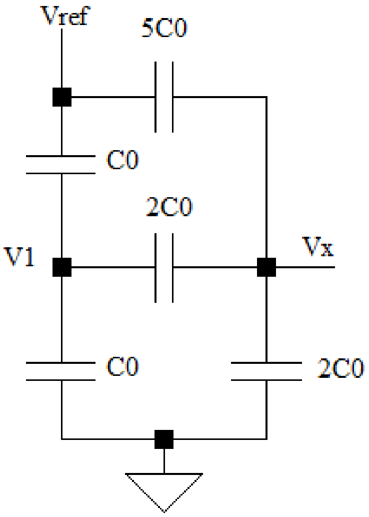

Figure 4.3: Equivalent circuit for input 1011.

trace markers, it is evident that the step sizes are uniform and their value is $V_{DD}/2^8$.
This 8-bit DAC architecture occupies 26 μm^2 area, consumes 6.42 nW at 3 V and has a settling time of 69 ns as opposed to 51 μm^2, 11.8 nW and 136 ns with a conventional DAC architecture with single step switching [31].
The simulated differential non-linearity (DNL) and integral non-linearity (INL) versus input codes are shown in Fig 4.7 and Fig 4.8, respectively. It can be seen that both INL and DNL are smaller than 0.68 LSB for the entire range of the input code.
The dynamic characteristics of the proposed DAC is shown in Fig 4.9. For measuring the dynamic characteristics such as Signal to Noise Ratio (SNR), Spurious Free Dynamic Range (SFDR) and Signal to Noise And Distortion (SINAD), an output from an 8-bit ADC is given as the input. SNR and SFDR were simulated with a 240 Hz input sine wave. The corresponding SNR and SFDR were 48.1 dB and 52.3 dB.

4.3 Comparison with conventional DAC for different switching schemes

Table 4.1 shows a comparison of the novel architecture with conventional DACs for power, area and settling time, with different switching schemes discussed in [32]. In the table, a_{sw} is the area of switch and t_{sw} is the switching time for a single transition. The first four rows in the table show the conventional architecture with different switching schemes, employed for reducing

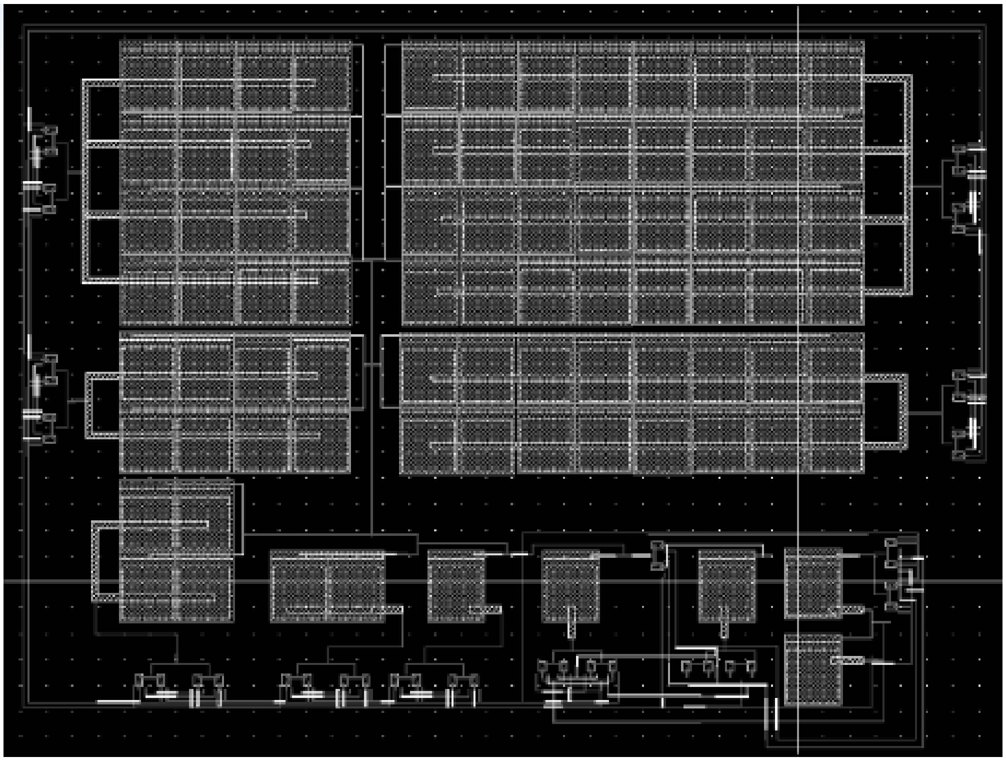

Figure 4.4: Layout of novel DAC Architecture.

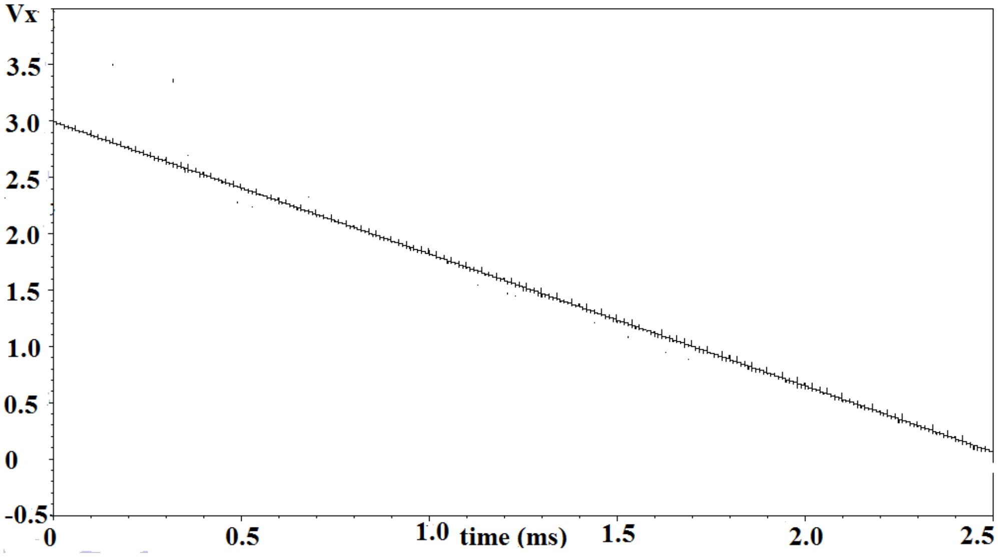

Figure 4.5: Post Layout Simulation results.

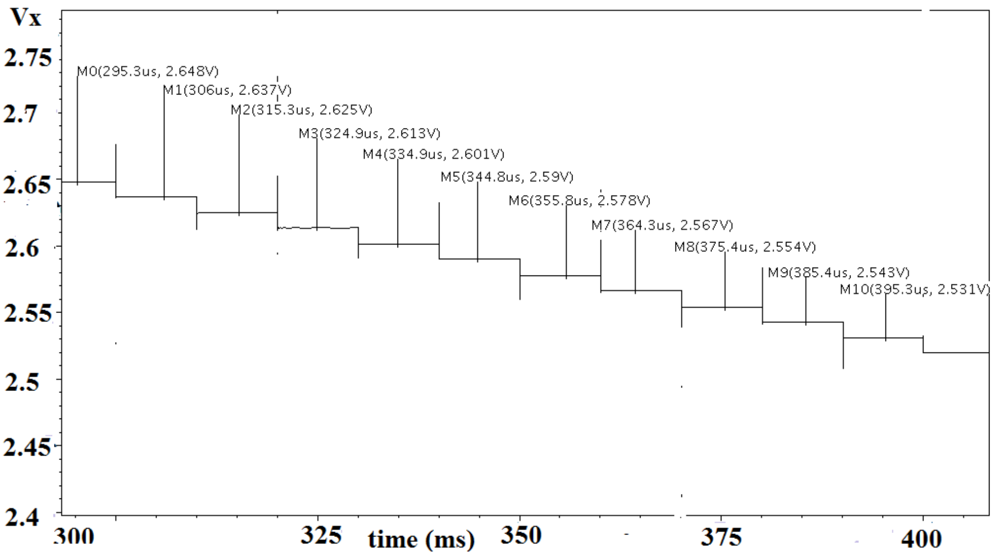

Figure 4.6: Magnified version of simulation results.

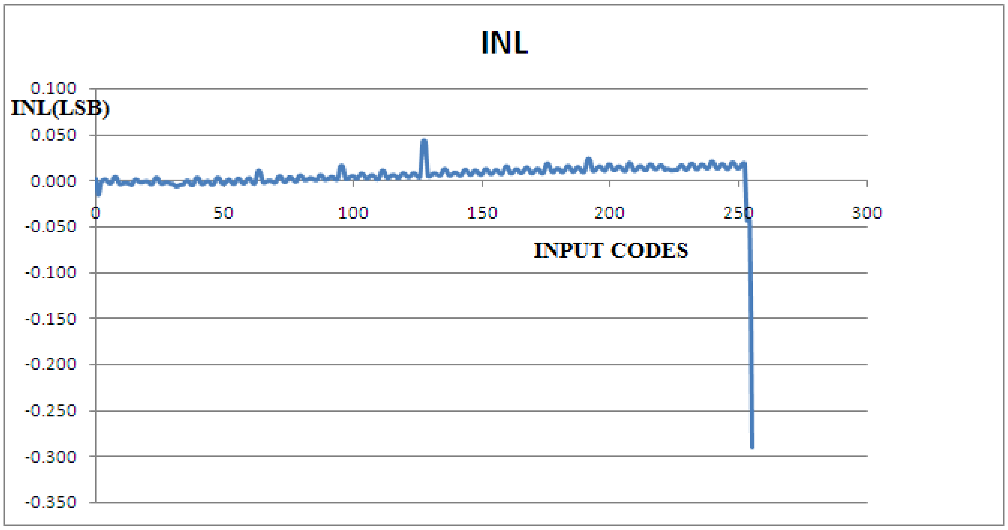

Figure 4.7: Simulated INL versus input codes.

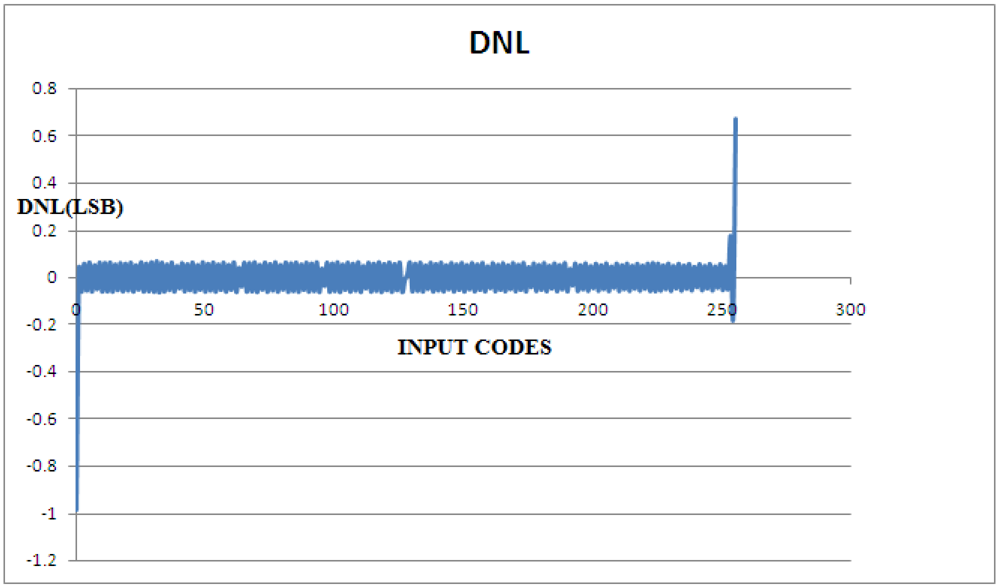

Figure 4.8: Simulated DNL versus input codes.

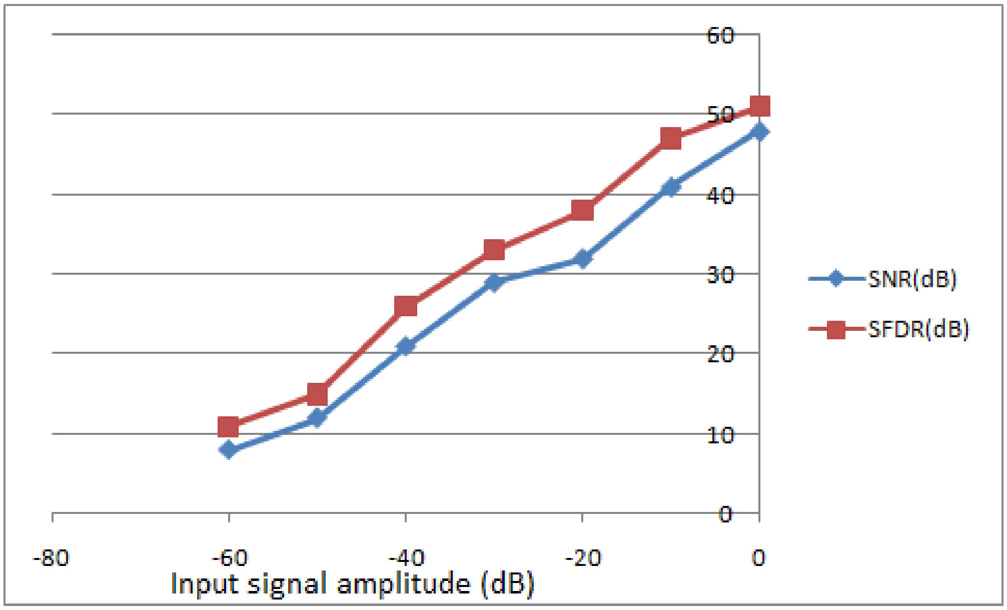

Figure 4.9: Simulated dynamic performance vs. input.

Switches schemes	Power	Delay	Area
1 step	$n(\frac{C_0V_{REF}^2}{4} + \frac{5}{4}C_0V_{REF}^2)$	$\tau ln(2^{n+1})$	$(2^nC_0 + (b+1)a_{sw}$
2 step	$n(\frac{C_0V_{REF}^2}{4} + \frac{3}{4}C_0V_{REF}^2)$	$\tau ln(2^{n+1}) + t_{sw}$	$2^nC_0 + (b+1)a_{sw}$
Charge sharing	$n(\frac{C_0V_{REF}^2}{4} + \frac{7}{12}C_0V_{REF}^2)$	$\tau ln(2^{n+1}) + t_{sw}$	$2^nC_0 + (2b+1)a_{sw}$
Capacitor splitting	$n(\frac{C_0V_{REF}^2}{4} + \frac{C_0V_{REF}^2}{4})$	$\tau ln(2^{n+1})$	$2^nC_0 + (2b+1)a_{sw}$
Novel DAC	$n(\frac{C_0V_{REF}^2}{8} + \frac{3C_0V_{REF}^2}{8})$	$\frac{\tau}{2}ln(2^{n+1})$	$(2^{n-1}+3)C_0 + (b+2)a_{sw}$

Table 4.1: Comparison between the conventional switching schemes and the novel architecture

power dissipation. The last row of the table shows the power dissipation, area and speed of the new architecture. The table clearly shows that the new architecture reduces power, area, as well as settling time.

A conventional single step switching has been used with the novel architecture. Other switching methods have not been tested with this architecture. Employing other switching methods may improve its performance further.

4.4 Comparison with the Existing capacitor Array DAC Architectures

Three architectures, namely Conventional Binary Weighted Capacitive Array(CBW), Binary Weighted Capacitive Array with attenuation Capacitor(BWA) and Split Binary Weighted Capacitive Array(SBW), taken from [ref 1] are compared with the proposed architecture.

To have a better comparison between the architectures, the value of the average power consumption, maximum standard deviations of the INL and DNL, and the total capacitance required for each architecture are summarized in table.

Standard deviation of INL and DNL for the proposed architecture is calculated using [30] and [47].

Architectures	Average Power	Standard Deviation of INL	Standard Deviation of DNL	Total capacitance required
CBW	$0.662^{N}\frac{f_{clk}C_0V_{ref}^2}{N+1}$	$\left(2^{\frac{N}{2}-1}\right)\frac{\sigma_0}{C_0}$	$\left(2^{\frac{N}{2}}\right)\frac{\sigma_0}{C_0}$	$2^N C_0$
BWA	$1.252^{N/2}\frac{f_{clk}C_0V_{ref}^2}{N+1}$	$\left(2^{\frac{3N}{4}-1}\right)\frac{\sigma_0}{C_0}$	$\left(2^{\frac{3N}{4}}\right)\frac{\sigma_0}{C_0}$	$2^{N/2}C_0$
SBW	$0.412^{N}\frac{f_{clk}C_0V_{ref}^2}{N+1}$	$\left(2^{\frac{N}{2}-1}\right)\frac{\sigma_0}{C_0}$	$\left(2^{\frac{N}{2}}\right)\frac{\sigma_0}{\sqrt{2}C_0}$	$2^N C_0$
Proposed DAC	$0.59\left(2^{(N-1)}+3\right)\frac{f_{clk}C_0V_{ref}^2}{N+1}$	$\approx\left(2^{\frac{N}{2}-1}\right)\frac{\sigma_0}{C_0}$	$\approx\left(2^{\frac{N}{2}}\right)\frac{\sigma_0}{C_0}$	$(2^{N-1}+3)C_0$

Table 4.2: Comparison between the conventional architectures and the novel architecture

4.4.1 Derivation For Standard deviation of Integral Non-linearity(INL) and Differential Non-linearity(DNL)

For calculating the linearity characteristics of the proposed scheme, each of the capacitor is modeled as the sum of the nominal capacitance plus the error.

$$C_n = 2^{n-2}C_0 + \delta_n \tag{4.7}$$

$$C_{n-1} = 2^{n-3}C_0 + \delta_{n-1} \tag{4.8}$$

$$C_0 = C_0 + \delta_0 \tag{4.9}$$

The error term $= \delta_n$ has zero mean and a standard deviation of

$$E[\delta_n^2] = 2^{n-2}\sigma_0^2 \tag{4.10}$$

If there are no initial charges, then V_x can be expressed as

$$V_x = \left(\frac{\sum_{n=1}^{b-1}(2^{(n-2)}C_0 + \delta_n)D_n}{2^{(b-1)}C_0} + \frac{(C_0+\delta_0)D_0}{2^{(b-1)}(C_0 + (C_0+\delta_0)D_0)} \right) V_{ref} \tag{4.11}$$

Subtracting the nominal value yields the error term

$$V_{error} = \left(\frac{\sum_{n=1}^{b-1}\delta_n D_n}{2^{(b-1)}C_0} + \frac{\delta_0 D_0}{2^{(b-1)}(C_0 + (C_0+\delta_0)D_0)} \right) V_{ref} \tag{4.12}$$

which can be approximated as

$$V_{error} = \left(\frac{\sum_{n=1}^{b-1}\delta_n D_n}{2^{(b-1)}C_0} + \frac{(C_0+\delta_0)D_0}{2^{(b)}C_0} \right) V_{ref} \tag{4.13}$$

which can be written as

$$V_{error} = \left(\frac{\sum_{n=1}^{b}(\delta_n)D_n}{2^{(b)}C_0} \right) V_{ref} \tag{4.14}$$

with variance

$$E[V_{error}^2(y)] = \frac{y}{2^{2b}} \frac{\sigma_0^2}{C_0^2} V_{ref}^2 \tag{4.15}$$

From that the worst case INL can be derived as

$$\sigma_{INL} = 2^{(b/2)-1}\frac{\sigma_0}{C_0}LSB \tag{4.16}$$

and the worst case DNL be

$$\sigma_{DNL} = 2^{(b/2)}\frac{\sigma_0}{C_0}LSB \tag{4.17}$$

The novel architecture can also be a part of split capacitor array dac, so that the savings in capacitor improves further compared with the normal split capacitor array dac. For example, an 8 bit digital to analog converter with binary weighted capacitor array would require 256 unit capacitors,whereas a split capacitor array would require 31 unit capacitors. If we employ the proposed novel architecture in the split capacitor array, the resultant architecture requires 22 unit capacitors. Fig 4.10, fig 4.11 and fig 4.12 explain the above mentioned concepts. From fig 4.10 it is evident that it requires 31 unit capacitors to implement an 8-bit Digital to Analog converter. Fig 4.11 shows the same

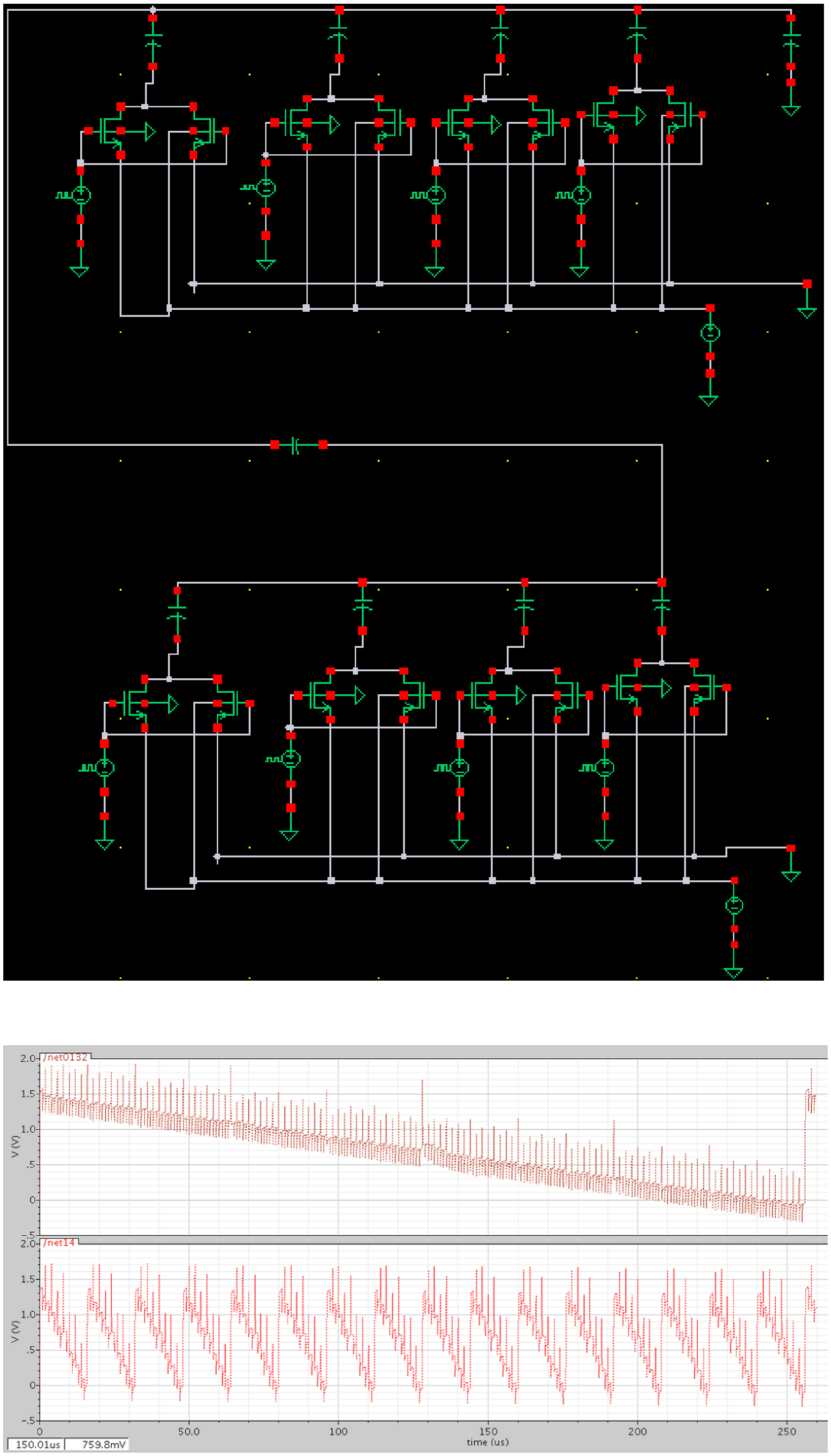

Figure 4.10: Split-capacitor array DAC and its simulation

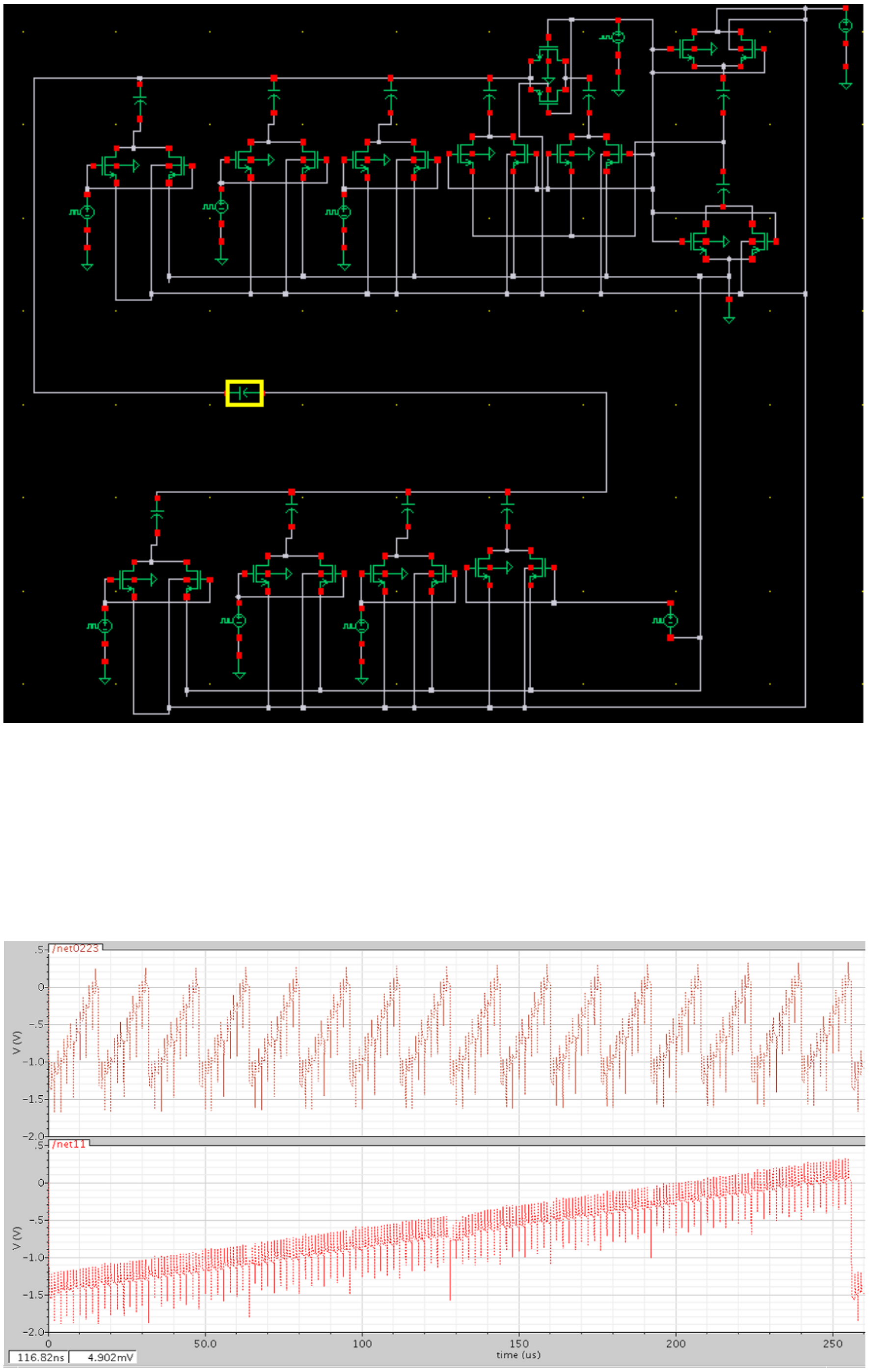

Figure 4.11: Split-Capacitor DAC with proposed Scheme on LSB sub-DAC and its simulation

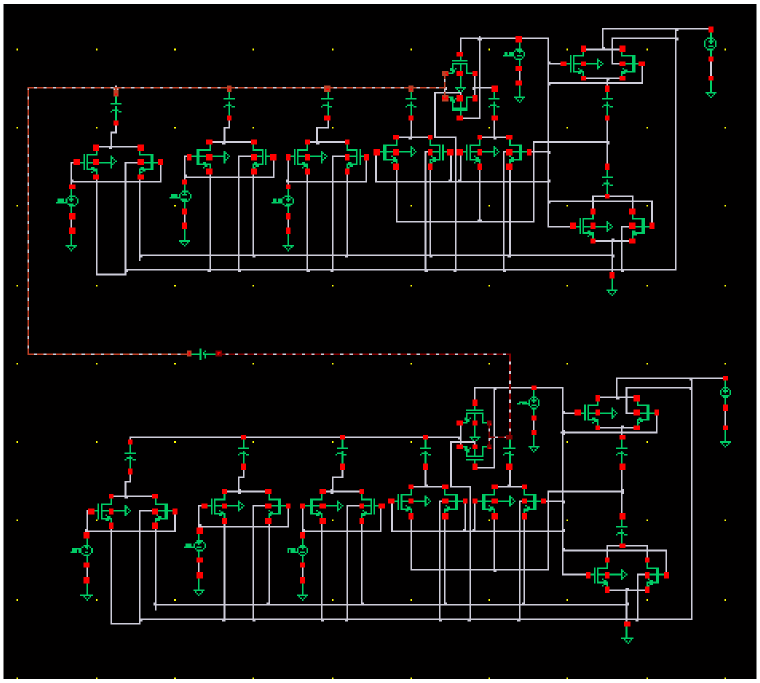

Figure 4.12: Split-capacitor DAC with proposed scheme on both the sub-DACs

8- bit digital to analog conversion, but uses only 26 unit capacitor. It employs the proposed DAC architecture for its LSB sub-DAC. The simulation results show that both the architectures behave in the same way. Fig 4.12 shows the 8-bit conversion with 21 unit capacitors and it employs the proposed architecture for both the LSB and MSB sub-DACs.

4.5 Design Considerations

In this section we consider the effects of Capacitor parasitics and Capacitor mismatches.

4.5.1 Capacitor Parasitics

Case 1 : A Parasitic Capacitance (C_p) at V_1, would result in the following circuit. Using the Star-Delta conversion an equivalent would be generated as shown in fig 4.13.

Where

$$C_1 = \frac{mC_0 + mlC_0 + lC_0}{l} \tag{4.18}$$

$$C_2 = \frac{mC_0 + mlC_0 + lC_0}{m} \tag{4.19}$$

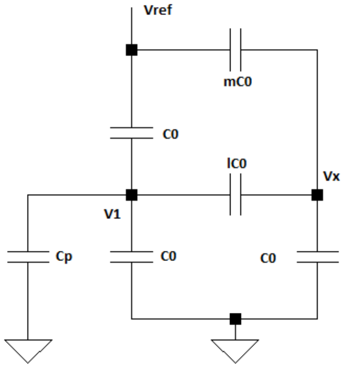

Figure 4.13: Generalized equivalent circuit with parasitic capacitance.

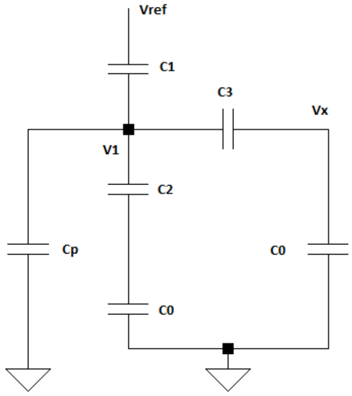

Figure 4.14: Equivalent circuit with a Delta-Star conversion

$$C_3 = mC_0 + mlC_0 + lC_0 \tag{4.20}$$

From Fig 4.14, with equations (4.18), (4.19) and (4.20), V_1 can be written as

$$V_1 = \frac{V_{ref}C_1}{C_1 + \frac{C_2C_0}{C_2+C_0} + C_p + \frac{C_3C_0}{C_3+C_0}} \tag{4.21}$$

while substituting (4.18), (4.19) and (4.20) in equation (4.21)

$$V_1 = \frac{V_{ref}\left(\frac{mC_0+mlC_0+lC_0}{l}\right)}{\left(\frac{mC_0+mlC_0+lC_0}{l}\right) + \frac{\left(\frac{mC_0+mlC_0+lC_0}{m}\right)C_0}{\left(\frac{mC_0+mlC_0+lC_0}{m}\right)+C_0} + C_p + \frac{(mC_0+mlC_0+lC_0)C_0}{(mC_0+mlC_0+lC_0)+C_0}} \tag{4.22}$$

Node voltage V_x can be written as

$$V_x = \left(\frac{m+ml+l}{m+ml+l+1}\right) V_1 \tag{4.23}$$

V_x can be written as in (4.24), while substuting (4.22) in (4.21)

$$V_x = \left(\frac{m+ml+l}{m+ml+l+1}\right)\left(\frac{V_{ref}\left(\frac{mC_0+mlC_0+lC_0}{l}\right)}{\left(\frac{mC_0+mlC_0+lC_0}{l}\right)+\frac{\left(\frac{mC_0+mlC_0+lC_0}{m}\right)C_0}{\left(\frac{mC_0+mlC_0+lC_0}{m}\right)+C_0}+C_p+\frac{(mC_0+mlC_0+lC_0)C_0}{(mC_0+mlC_0+lC_0)+C_0}}\right) \tag{4.24}$$

The following example would illustrate the effect of C_p at V_x.

Consider a bit pattern of ”11101” for a 5-bit digital to analog conversion with m = 14, l=2. Now V_x can be calculated as

$$V_x = \left(\frac{1276C_0}{1408C_0 + 58C_p}\right) V_{ref} \tag{4.25}$$

$$V_x = \left(\frac{29C_0}{32C_0 + 1.3C_p}\right) V_{ref} \tag{4.26}$$

For the next bit pattern ”‘11110”

$$V_x = \frac{15}{16} V_{ref} \tag{4.27}$$

The difference in these two generated voltage is

$$V_{x1} - V_{x2} = \frac{15}{16} V_{ref} - \left(\frac{29C_0}{32C_0 + 1.3C_p}\right) V_{ref} \tag{4.28}$$

In the above equation, if C_p = 0, then the difference is $(1/32)V_{ref}$ which is $V_{ref}/2^n$. If $C_p/C_0 = 0.5$ then the difference in two consecutive generated voltage is $(1/22)V_{ref}$ which is well within $0.5LSB$. This explanation shows that the Parasitic capacitance at V_1 will not affect the static characteristics of the proposed DAC.

Case 2 : If there is an additional parasitic capacitance from V_{DD} to V_1 : The generalized equivalent diagram would be as shown in fig 4.15. The generalized voltage at V_x is

$$V_x = \frac{m(C_0+C_p)+ml(C_p+C_0)+lC_0}{m(C_p+C0)+ml(C_p+C_0)+lC_0+C_0} \tag{4.29}$$

$$\frac{V_{ref}\left(\frac{m(C_0+C_p)+ml(C_0+C_p)+lC_0}{lC_0}\right)}{\left(\frac{m(C_0+C_p)+ml(C_0+C_p)+lC_0}{lC_0}\right)+\frac{\left(\frac{m(C_0+C_p)+ml(C_0+C_p)C_0+lC_0^2}{m(C_0+C_p)}\right)C_0}{\left(\frac{mC_0(C_0+C_p)+ml(C_0+C_p)C_0+lC_0^2}{m(C_0+C_p)}\right)+C_0}+C_p+\frac{(m(C_0+C_p)+ml(C_0+C_p)+lC_0)C_0}{(m(C_0+C_p)+ml(C_0+C_p)+lC_0)+C_0}}$$

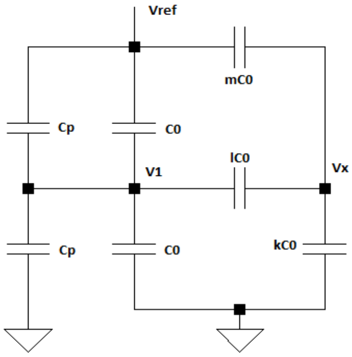

Figure 4.15: Equivalent circuit with parasitic capacitance

For a five bit digital to analog conversion, with a bit pattern of ”11101”, m = 14, l=2 and for $(C_p/C_0) = 0.5)$. From these values V_x can be calculated as

$$V_{x1} = \frac{29.33}{32} V_{ref} \tag{4.30}$$

For ”‘11110”’, V_x can be generated as

$$V_{x2} = \frac{15}{16} V_{ref} \tag{4.31}$$

The difference between the two subsequently generated voltage is

$$V_{x2} - V_{x1} = \frac{0.67}{32} V_{ref} \tag{4.32}$$

Which is well within $0.5LSB$.

Fig. 4.16 and 4.17 plots the impact of Cp on DAC linearity for an 8-bit DAC. The peak-to-peak INL of the DAC is plotted against the ratio Cp/Cu and the corresponding DNL and INL plots for when Cp/Cu is 0.1 and 0.5 are also plotted. As the plots show, in order to get an INL below +/-0.5 LSB, the unit capacitance, C_0, should be about 2x larger than the parasitic cap at the output node.

4.5.2 Capacitor mismatches

The novel architecture employs 3 additional capacitances, which are the same as the unit capacitance. So the required matching ratio compared with the unit capacitor is 1:1. This makes the

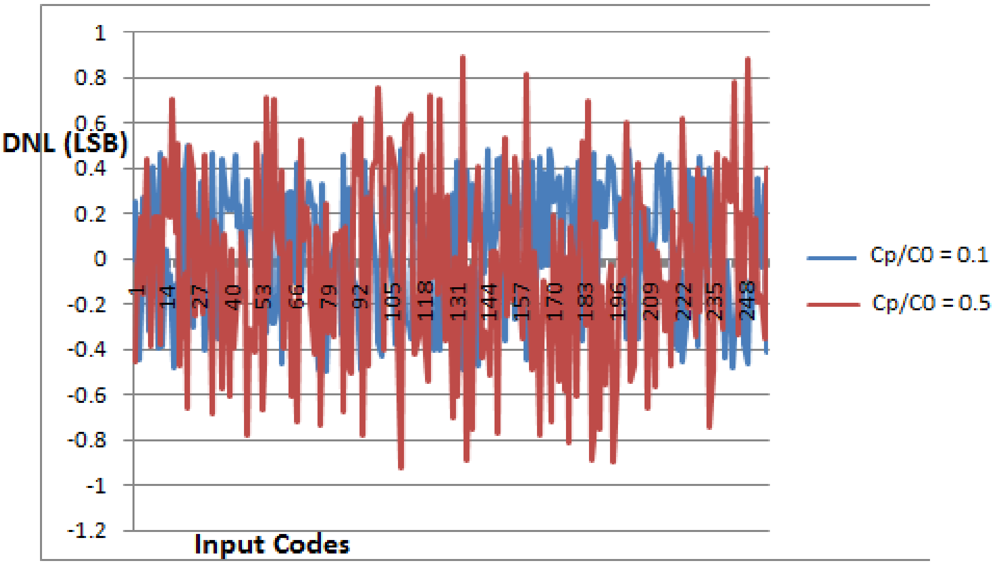

Figure 4.16: DNL for capacitor parasitics

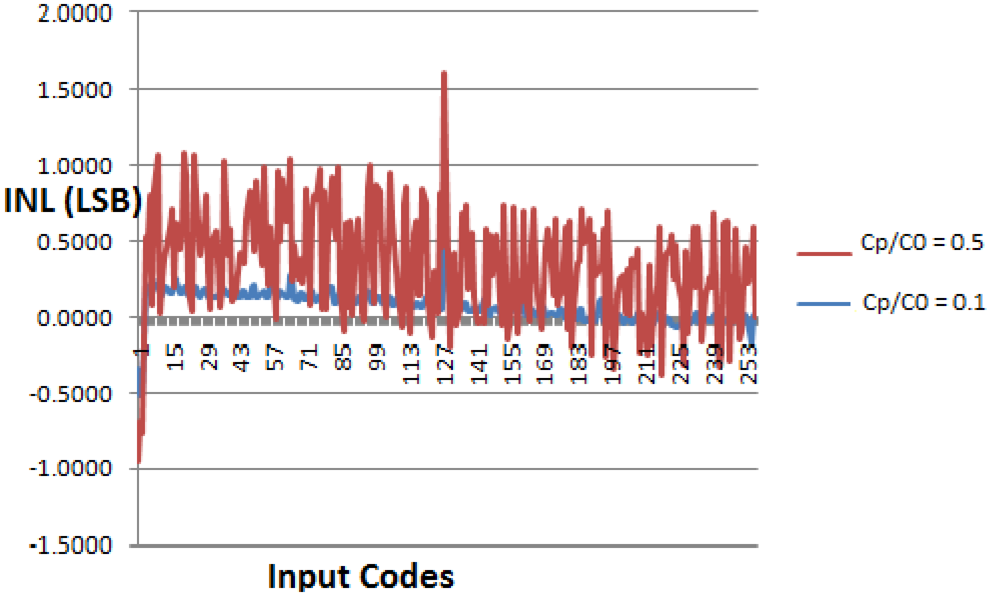

Figure 4.17: INL for capacitor parasitics

design less prone to capacitance mismatches. The impact of these random variations in captured in Fig.4.18 which plots the INL of an 8-bit DAC for $3\sigma C_0$ variation of 5 %. As the plot shows, the DAC shows a fairly muted response even to significantly large random per unit capacitor variations.

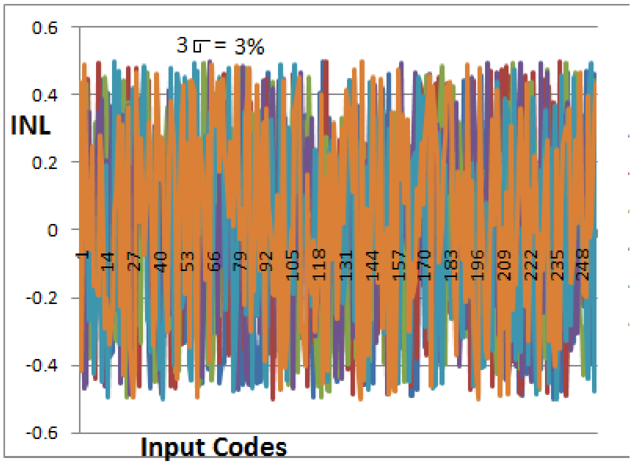

Figure 4.18: INL for randomly mismatched capacitor

Chapter 5

Design Automation for Successive Approximation Register Analog to Digital Converter (SAR ADC)

5.1 Successive Approximation Analog to Digital converter

SAR ADC has three levels of hierarchy: top-level, circuit-level and transistor-level. This paper works on top level model's specifications, find out the corresponding circuit level models and optimize the circuits. A more accurate transistor level model based on [18] is used in optimization. The transistor level model is presented in Appendix A.

The principle block diagram, for which the automation algorithm is used is shown in fig 5.1. The ADC consists of a capacitor DAC, comparator, and SAR logic. The charge redistribution technique provides an inherent sample and hold stage.

5.1.1 Top Level SAR ADC Model

This section discusses the specifications to be considered in the top level. These specifications can be optimally divided among the ciruict level components. Total power dissipation, conversion rate (speed), area and noise are the major specifications considered in this section.

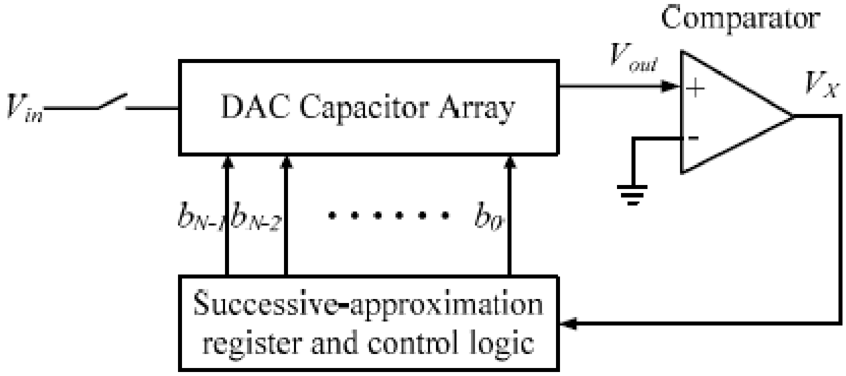

Figure 5.1: SAR ADC architecture

Total Power Dissipation

Total power dissipation is the sum of each component's power, i.e.,

$$P_{tot} = P_{sar} + P_{DAC} + P_{comp} \tag{5.1}$$

Where P_{sar} is the power consumed by the digital part, P_{DAC} is the power taken by DAC and P_{comp} is the comparator's power.

Area

The total area occupied by SAR ADC is given by

$$A_{tot} = A_{sar} + A_{DAC} + A_{comp} \tag{5.2}$$

Where A_{sar} is the area occupied by the digital part, A_{DAC} is the area taken by DAC and A_{comp} is the comparator's area.

Speed

An N-bit SAR ADC needs at least (N+1)-clock cycles to complete the conversion. From the speed specifications, time period for the clock can be decided. The duty cycle of the clock should be sufficient for the DAC and comparator output to settle and the SAR logic to produce the corresponding logic. So the speed of SAR ADC decides the settling time for comparator and DAC. The clock period can be calculated as

$$T_{ON} = \frac{1}{2 * speed * N} \tag{5.3}$$

This T_{ON} should be greater than the sum of DAC settling time and comparator settling time, i.e.,

$$k_1\tau_{comp} + k_2\tau_{DAC} + \leq T_{ON} \tag{5.4}$$

k_1 and k_2 in equation (5.4) are the constants derived from the settling error. τ_{comp} and τ_{DAC} are the time constants for comparator and DAC respectively.

Input-referred noise

Digital to Analog converter is the major contributor of noise, both in sampling phase and conversion phase, the noise can be expressed as

$$P_{noise} = \frac{kT}{C_{tot}} \tag{5.5}$$

where k is the Boltzman constant, T is the absolute temperature and C_{tot} is the total capacitance of DAC.

5.1.2 Successive approximation registers

The logic control block of the ADC has two important aims. It sets the switches in function of the current state of the conversion and the output of the comparator; secondly it computes and stores the digitally converted values that is to be given as output at the end of the conversion. A two layer flip-flop based SAR logic is shown in fig 5.2. For the given specifications, the proposed algorithm estimates the sizes of the flip-flops.

GP Model for flip flop

A good number of literature are available which explain the optimization of digital circuits using geometric programming [11-15].

A detailed study of Digital circuit optimization via geometric programming is proposed by Boyd et al.[13]. A slightly modified version of that work [13] is used in this paper to optimize the digital part.

Total power dissipation, speed and area are the major specifications to be considered. This section

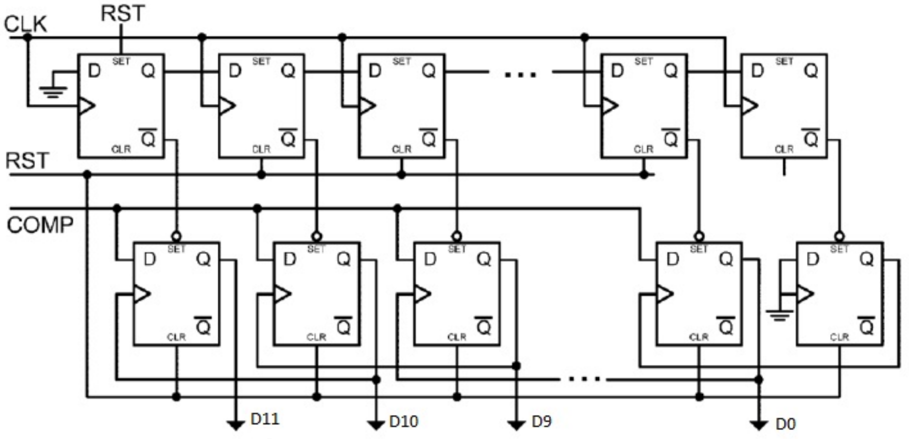

Figure 5.2: SAR schematic

explains the generalized posynomial models for those specifications.

Generalized Posynomial Power Model

The total dynamic power dissipated can be derived as.

$$P_{dyn} = \sum_{i=1}^{n} f_i (C_i^L V_{dd} + C_i^{int}) V_{dd}^2 \tag{5.6}$$

This is evidently a posynomial of the device sizes and supply voltages.

The total (average) static power is given by the formula

$$P_{static} = \sum_{i=1}^{n} I_i^{leak} V_{dd} \tag{5.7}$$

The total power is the sum of dynamic and static power.

$$P_{tot} = P_{static} + P_{dyn} \tag{5.8}$$

Generalized Posynomial Delay Model

For calculating the delay, we are using RC delay model; The delay for a functional block can be divided into 1) setup time 2) Hold time and 3) Clock to Q delay.

$$D_{total} = t_{setup} + t_{logic} + t_{clk \to Q}$$

Generalized Posynomial Area Model

Area can be calculated as

$$Area = \alpha_1 W_1 L_1 + \alpha_2 W_2 L_2 + \cdots + \alpha_n W_n L_n \tag{5.9}$$

Where α_i is process dependent variable, Ws and Ls are the Width and Length of transistors used. While giving the values for total power dissipation, speed and area, the GP model of SAR logic gives the sizes of the transistors used in the flip flops.

5.1.3 Capacitor array Digital to Analog Converters

The basic architecture, the working principle of Capacitor array DAC is already discussed in chapter 4, which can be referred here.

GP Model for DAC

The top level model allocates the input specifications for the DAC . The input specifications are the maximum area the DAC can occupy, the maximum time DAC can take to settle and the noise floor for DAC. The objective function is the total power dissipation.
According to the given specifications, the automation algorithm chooses the corresponding switching schemes for the digital to analog conversion. The GP model gives the design variables for the selected switching scheme.

Design variables and Trade off analysis

The unit capacitor's and the sizes of the switches are the major design variables. An upper and lower limits are imposed on the size of the capacitor. The size of the switches are limited by the technology constraints.
Identifying the proper size of the unit capacitor is the main motto of this design. The performance of the DAC is vastly dependent on the unit capacitor. For a fair thermal noise, the capacitor size should be as large as possible, but large capacitor increases the settling time, area and power dissipation.
We could neither decrease the capacitor size beyond a limit, because the technology constraints impose a lower limit for the capacitor's size. The top plates of the capacitors in DAC are connected to the input of comparator, so the capacitor size should be such that the comparator's input parasitic capacitance should not affect the DAC's performance.

GP takes all the above mentioned constraints into consideration and provides an optimized unit capacitor size and also the switch sizes.

5.1.4 Comparators

GP for comparators has been dealt in greater detail in chapter 3, which can be referred here.

5.2 Design Automation for SAR ADC

A systematic automation design methodology is proposed to optimize the power of successive approximation analog to digital converter (SAR ADC). SAR ADC is a mixed signal circuit having both analog and digital sub-blocks. Different varieties of sub-blocks of SAR ADC are thoroughly analyzed and designed using geometric programming as optimization tool. According to the specification, the corresponding sub-blocks are chosen and integrated as the whole ADC. The specifications for the entire ADC is cautiously budgeted among sub-blocks, so that we will get an optimized and accurate design in hand. A computer Aided Design tool is developed to implement the whole optimization process. For a given specification, this tool will give the corresponding design variables for minimum power. This paper, thus gives an abstract level automation tool for SAR ADC which can be improved further.

5.3 Automation algorithm

For a given specification, the systematic automation problem is to allocate the specifications(inputs) for the sub-block models, and to give the optimized design variables.

From the literature[31,32,33,34,35], we can understand that in power perspective, analog parts will take half the total power and the remaining power will be taken by digital circuits. The settling time of DAC and the delay of comparator are the major factors which decide the speed of ADC. Area wise, the major part of the chip is occupied by capacitor array, in the remaining area the digital parts (2(n+1) flip flops and interconnects) will take more area and then the comparator. The total noise floor of SAR ADC is LSB/2, and great part of the noise is contributed by the DAC.

The optimization model for comparator will generate the dimensions of transistors it employs and the value of biasing current. The DAC optimization model will give the dimension of unit capacitor and the sizes of switches. The model for SAR algorithm will give the sizes of gates employed in the flip flops. This optimization tool also gives back the specifications corresponding to the optimized design variables.

5.3.1 Flow Chart

The flow chart of the proposed algorithm is shown in fig 5.4.

5.3.2 Optimization Example

To check the validity of the proposed algorithm, a 8-bit, 1 MS/s, $0.8\mu W$ SAR ADC is designed by using the proposed automation algorithm. The comparison of this architecture with the existing architecture [35] with the same specifications shows the merits of the proposed algorithm.The design details, layout, static and dynamic characteristics of the SAR architectures and the comparison are discussed in this section.

Design details

The top-level SAR ADC specifications are, power consumption - 1.5mW, speed should be greater than 1 MS/s, area should be less than 400x400 μm^2 and thermal noise for DAC should be less than $LSB/6$.
The allocated specifications for comparator are, power dissipation - 0.6 mW, delay - $0.2\mu s$, gain - 10,000 and area - 40x40 μm^2. The optimization tool gives the design variables as shown in table 3. The resultant power consumption - 0.32 mW, delay - 0.19 μs and area is approximately 15x15 μm^2, corresponding to the optimized design variables.

Table 5.1: Comparator Simulation Results

W1/L1	W31/L31	W3/L3	W5/L5	W6/L6	Ib(μA)	Vb(V)
1/5	2/5	2/4.4	1.2/4.3	1.2/4.4	3.4	1.1

The specifications given to DAC are Power dissipation - 0.3 μW, settling time - 0.7 μs area 250x250 μm^2 and noise - $LSB/6$. For 3 supply voltage, the noise floor - 0.134 mV. The unit capacitor can be calculated as 105.5 fF. With this unit capacitor, the settling time - 0.68 μS, which is well within the constraints. Area of the DAC will be approximately 230x225 μm^2.
For the SAR logic, the assigned specifications are power dissipation - 0.7 μW, delay - 0.1 μs and area - 80x80 μm^2. The optimization tool gives the design variables as The total area for the digital part corresponding to this design variables - 42x42 μm^2, total power - 0.53 μW and the total delay

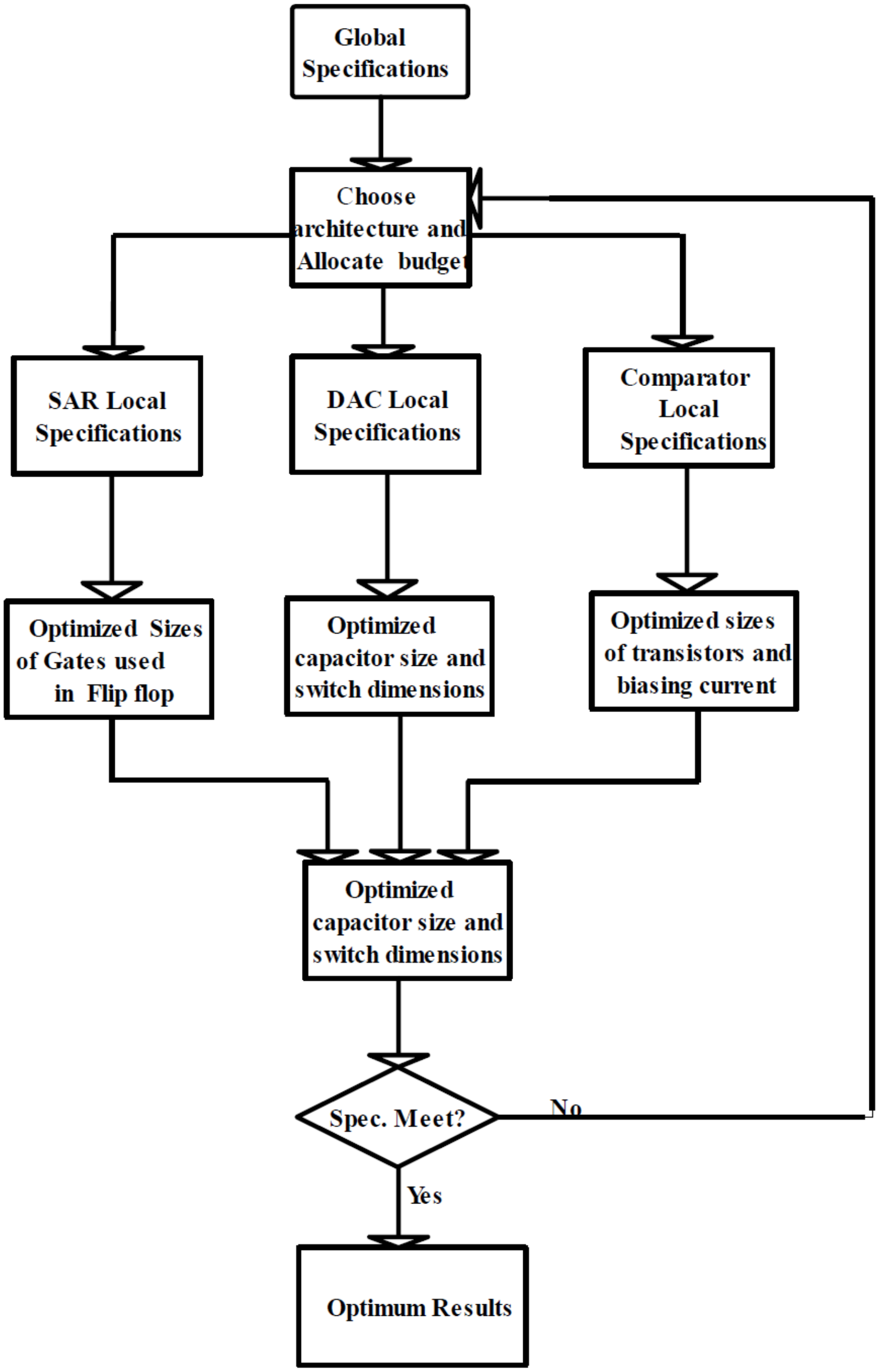
Global Specifications
Choose architecture and Allocate budget
SAR Local Specifications
DAC Local Specifications
Comparator Local Specifications
Optimized Sizes of Gates used in Flip flop
Optimized capacitor size and switch dimensions
Optimized sizes of transistors and biasing current
Optimized capacitor size and switch dimensions
Spec. Meet?
No
Yes
Optimum Results

Figure 5.3: Flowchart of Automation Algorithm.

Table 5.2: Digital part Simulation Results

Components	Sizes(μm)	Power(nW)	Delay(ns)
CLKinv	$Wp = 4.8Wn = 2.4$	21	0.03
Nand1	$Wp = 2.4Wn = 2.4$	26	0.07
Nand2	$Wp = 2.4Wn = 2.4$	26	0.07
Inv1	$Wp = 4.8Wn = 1.6$	21	0.03
Inv2	$Wp = 4.8Wn = 2.4$	21	0.03
Inv3	$Wp = 3.2Wn = 2.4$	19	0.01

Table 5.3: Comparison between existing design

Bits	Supply voltage(V)	Power(mW)	Speed(MS/s)	Area(μm^2)
8	3.3	1.56	1.23	450x315
8	3.3	0.8	1.04	254x254

- 0.1 μs.

Layout consideration

With the design variables given by the automation algorithm the layout is drawn using UMC 0.18 μm technology. By sweeping the analog values over the complete input range with a ramp generator the static measurements are performed. Differential non-linearity (DNL) and integral non-linearity (INL) plots are calculated.
In fig. 5.5, DNL plot is depicted. The maximum deviation is $+0.4/-0.6LSB$. Figure 5.6 shows the INL plot, INL is within $\pm 0.7LSB$

Comparison with the existing design

These results can be compared with the results in [35]. The comparison is shown in table 5. The first row indicates the results from the existing design, and the second row shows the results from the proposed methodology. Comparison shows that the proposed methodology gives minimum power and minimum area for the same number of bits, supply voltage and same speed.
An optimization methodology is proposed for the systematic automation of Successive Approximation Analog to Digital converter. A deterministic model based on linear programming is used to optimally divide top-level specifications into the specifications for the sub-blocks. Geometric programming based models for the sub-blocks are derived which gives the optimized design vari-

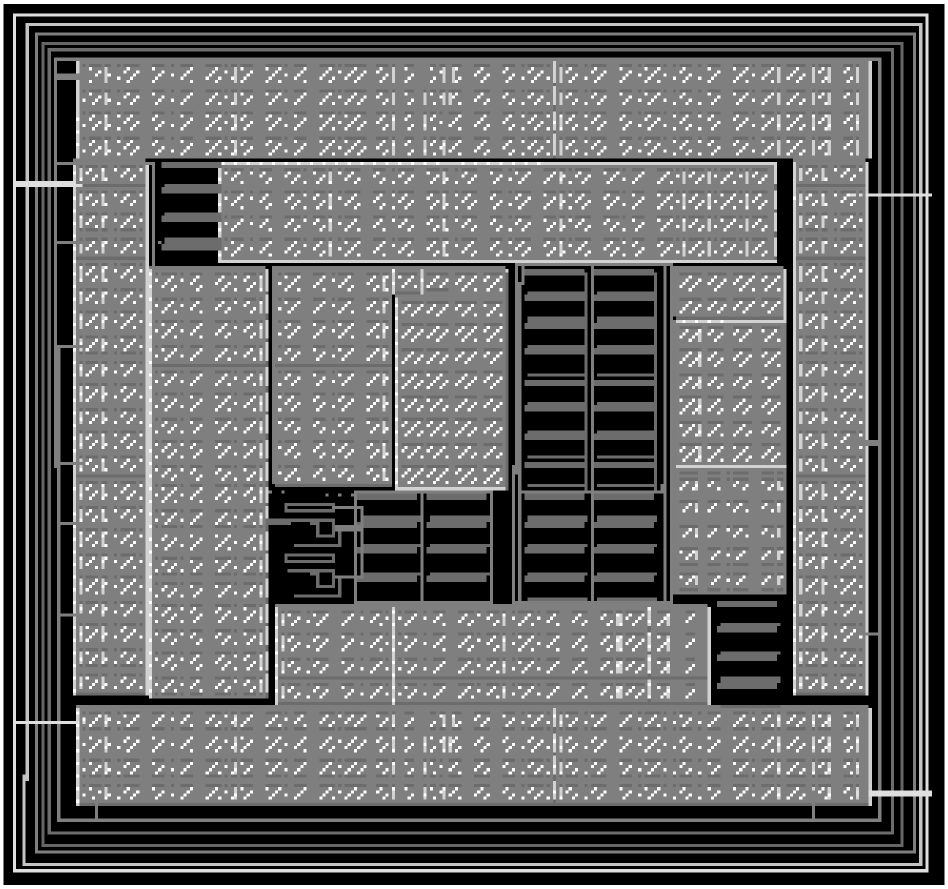

Figure 5.4: Layout view

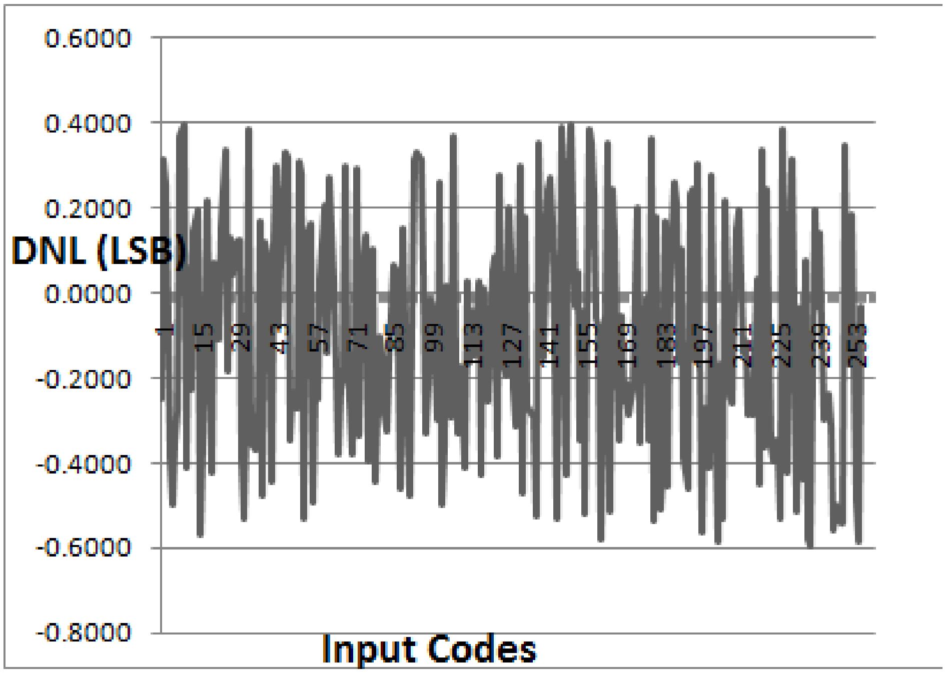

Figure 5.5: Simulated DNL

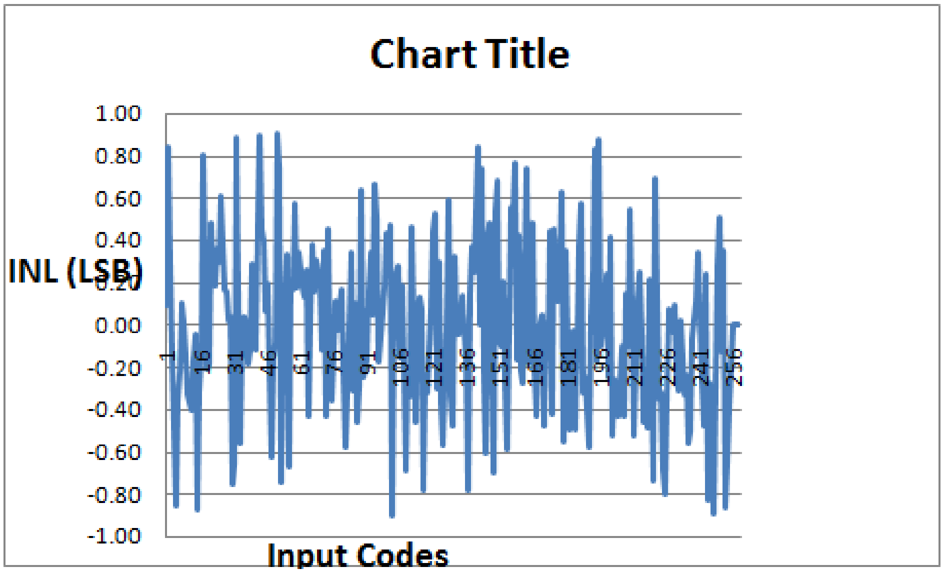

Figure 5.6: Simulated INL

ables. A 8-bit, 1 Ms/s, $0.18\mu W$ SAR ADC is designed and compared with an existing design. Comparison justifies the proposed automation method.

Chapter 6

Pipelined Analog to Digital Converter

An automation design methodology based on geometric programming for designing low power pipelined analog to digital converter is proposed in this paper. An equation based optimization is presented, which can be easily converted into convex optimization problem. This paper presents a systematic approach to design a multi bit per stage pipelined analog to digital converter, for the given specifications with power optimization. This tool computes the design variables in a predefined ADC topology, for a desired process technology. This paper considers capacitor scaling to make nonidentical stages to save power. An example is presented to justify this work.

6.1 Pipelined ADC Architecture

The conventional pipeline ADC [36] is shown in Fig. 6.1. Each stage of a pipeline architecture consists of a sample and hold, an analog to digital converter, a digital to analog converter, a subtracter, and an amplifier. A single circuit, the multiplying digital to analog converter (MDAC), can be realized by using switched capacitor techniques [36 - 40]. Which implements the functions of the S/H and DAC, but also the subtracter and amplifier. In each stage, MDAC and the sub-ADC are the major power dissipation components. The power contributions from other components are typically much smaller. As a result it is reasonable to consider only the MDAC and sub-ADC for power analysis.

6.2 Design Methodology for Pipelined ADC

This paper presents a two layer optimization problem for Pipelined ADC. The first stage of the optimization finds out the optimum number of stages required and the bits per stage. The second

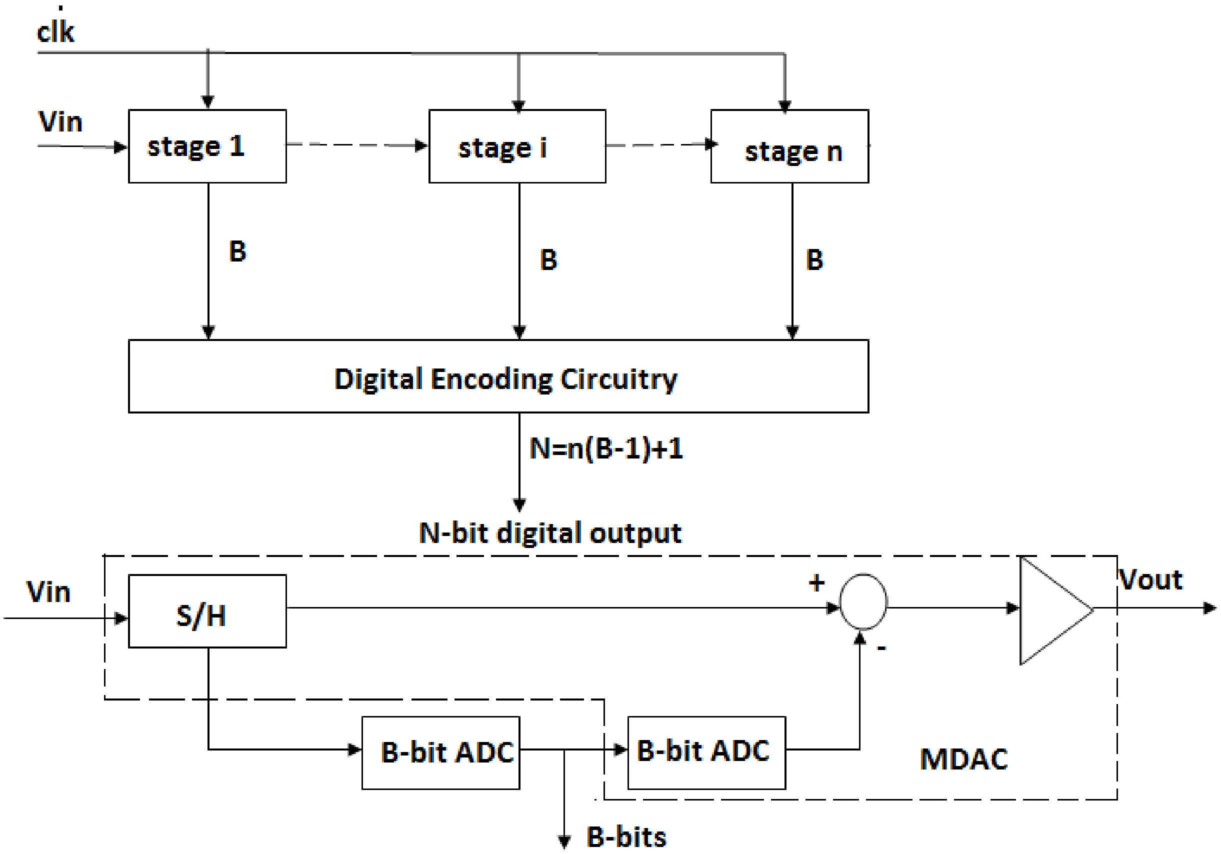

Figure 6.1: Conventional Pipelined ADC Architecture

stage estimates the design variables for the number of stages given by first stage with minimum power.

6.2.1 Optimizing the Number of Stages and Bits/stage:

The Number of stages and the number of bits per stage is limited by [41]

1. Thermal noise.

2. The gain and non-linearity errors produced by the capacitor mismatches.

3. Amplifier's non-ideal effects.

Appropriate sizing of the capacitor reduces the thermal noise. By employing the self calibration techniques and proper amplifier design gain and non-linearity errors in the MDAC can be reduced. Digital correction techniques can be employed to reduce the error from the sub-ADC and MDAC.

Design Space

To explore the design space, in which we can find an optimal design of Pipelined ADC, the optimization to be followed is based on the following steps.

1. The ADC specifications such as resolution, reference voltage, minimum speed of operation, minimum area requirements and Peak-to-peak analog input swing.

2. The effective resolution per stage is upper limited by 3 to avoid high resolution ADC (High resolution a need more number of comparators which results in high power dissipation) and lower limited by 1.

3. The first few stages, this paper is taking first two stages, need more gain and low thermal noise requirements (because the inaccuracies of the stage shifts the residue output from its ideal value, and its cumulative effect over the stages turns into linearity degradation of the overall transfer characteristic), so we are fixing the first two stages resolution as 2. The last stage does not require a MDAC and the resolution can be fixed as 2.

4. According to the gain and noise requirement, we can decide the MDAC specifications.

(a) Thermal noise consideration:

The on-resistance of the switches and the op amps are the main sources of thermal noise[40, 41]. The rms value of the total thermal noise(TTN) is given by

$$TTN = \sqrt{\frac{N_{SH}^2}{1} + \frac{N_{MDAC2}^2}{1.G_1^2} + \cdots + \frac{N_{MDAC(NS-1)}^2}{1.G_1^2 \dots G_{NS-1}^2}} \tag{6.1}$$

Where N_{MDAC} and N_{SH}, respectively are the output referred RMS noise contributions of the $MDAC_i$ and of the S&H and G_i represents the closed loop gain of the $MDAC_i$ during residue amplification.

The output referred mean square noise introduced by each MDAC can be approximately as [42]

$$\begin{aligned} N_{MDAC_i}^2 &= V_{S\&H}^2 + V_1^2 + V_2^2 \\ &= \xi[\frac{kT}{C_{MDAC_i}} \\ &\quad +(4kT.R_{ON} + 4kT.R_{eq})B_{eq}]G_i^2 \end{aligned} \tag{6.2}$$

Where C_{MDAC_i} is the total input capacitance of the $MDAC_i$, k is the Boltzmann's constant, and T is the absolute temperature.The constant ξ can assume the values of either one or two depending on, respectively, whether the circuit is implemented in a single ended or fully differential configuration. R_N and R_{eq} are the on resistance of the switch and output resistance of the residue amplifier respectively.

Equation (6.1) and (6.2) will give as a constraint which define a design space for switch sizes, maximum input capacitance for MDAC and sizes of the transistors in amplifier used in MDAC.

(b) Settling Time constraint

Settling time of the MDAC depends upon the total capacitance used and the on-resistance of the switches. Settling time decides the speed of the stage, in turn speed of the ADC.

$$T_{settle} = k_1\tau \tag{6.3}$$

Where k_1 is the constant, which can be derived from the noise floor within which the MDAC will settle and τ is the time constant which is proportional to the total capacitance of the DAC and on-resistance of the switches.

Equation (6.3) will give an another design space with a lower limit for capacitor and dimension for the switches used.

(c) Amplifier Requirements

The gain and slew rate are the major constraints for the amplifier. The gain can be defined by the accuracy required by the stage, if an accuracy of M bits required during the residue amplification in an N_i-bit stage, then the finite amplifier dc gain A_0 must satisfy

$$A_0 \geq 2^{M-N_i}.\left(2^{N_i}+1+\frac{C_{pi}}{C_i}\right) \tag{6.4}$$

Slew rate(SR), can be defined as the ratio between the current and output capacitance of the amplifier, i.e.,

$$SR = I/C_L \tag{6.5}$$

Slew rate is a deciding factor for amplifier's settling time. Both SR and Gain are depend on the tail current. So optimized tail current can produce optimized slew rate and gain. Equation (6.4) and (6.5) will give an another design space for MDAC's amplifier design. The intersection of design spaces, discussed in (a), (b) and (c) will give an optimized design space for MDAC.

5. FLASH ADC Design

A high speed ADC which employs a resistive divider and 2^N-1 comparators, where N is the resolution of FLASH ADC. Since the comparators and resistive divider draw more power, and the comparator's offset limits the minimum input, resolution for FLASH ADC is limited by 4.

The main specifications to be determined in the ADC are the value of the resistance and the characteristics characteristics of the comparator.

(a) Deciding the unit capacitor in the resistive divider: The dominant time constant determines the optimum value of resistance. It can be shown that the dominant time constant can be given as [43]

$$\tau = \left[\frac{2^{N_i}(2^{N_i}-1)}{\pi^2}\right]RC \tag{6.6}$$

where R is the value of the unit resistor in the resistive divider and C represents the capacitance associated with each tap during the auto zero of the comparator, which includes the input capacitor of the comparator and all the parasitics in that node.

(b) Choosing the Comparator: Comparator is the heart of FLASH ADC, which decides ADC's characteristics. Three different comparator architectures, namely open loop, static and dynamic comparators are thoroughly analyzed and compared. The details are given in Chapter 3. table 6.1 shows the comparison between the three architectures.

Architecture	Complexity	Power	Speed	Gain	Area
Op amp based	Low	High	Low	Medium	Medium
Static Latch	Medium	Medium	High	High	Low
Dynamic Latch	Medium	Low	High	High	High

Table 6.1: Comparison between the comparator architectures

According to the input ADC specifications, we can choose the corresponding comparator architecture.

6. Capacitor Scaling:
In a pipeline ADC, the thermal noise contribution of a later stage is effectively attenuated by the gain of the previous stages [42]. So by relaxing the thermal noise constraint for the later stages demands small capacitors. On the other hand, MDAC's power dissipation is directly proportional to the capacitor value, by reducing the capacitor value, we can reduce the power dissipation. As a result, the capacitors of later stages can be scaled down without increasing the thermal noise significantly.

7. Once we have decided the design space for S&H, Comparator and MDAC, then we can decide the number of stages and number of bits per stage.

8. We have already decided that we will fix the first two stage's resolution as 2, and last stage resolution as 2. A MATLAB algorithm is written, which does the following things

   (a) It generates the possible interstage resolutions.

   (b) For each interstage resolutions, assigns specifications for S&H, MDAC and ADC.

   (c) Chooses the perfect comparator architecture for ADC.

   (d) Using Geometric Programming optimizes the interstage designs.

   (e) Compares each interstage combination and chooses the one which takes minimum power.

6.3 Equation based ADC Model

For an optimization problem, the necessary inputs are the objective function and the constraints. Pipelined ADCs are high resolution ADCs (12-18 bits), but as the resolution increasing, power dissipation is also increasing. Reducing power is thus an objective function for the design of Pipelined ADCs. From the specifications given, such as resolution, speed and area, we can provide the constraints to the optimization problem.

6.3.1 Objective function

By deciding the factors, decided in the previous section, the total power dissipation in a pipelined ADC can be given as

$$P_{tot} = \sum_{i=0}^{NS-1} P_{MDACi} + \sum_{i=1}^{NS} P_{FLASHi} + \sum_{i=1}^{NS} P_{sci} + \sum_{i=1}^{NS} P_{synci} \tag{6.7}$$

Where $P_{MDAC_i}$is the power dissipated in i^{th} MDAC, which is the sum of the static power dissipated in the residue amplifier and a dynamic power contributed by the switching of the capacitors at the sampling frequency.

P_{FLASHi} represents the power dissipation in FLASH ADC, which can be determined by the static contribution of resistive divider, dynamic contribution of the latch and the output capacitors, and the static contribution of preamplifier in comparator.

P_{sc} and P_{sync} are the power dissipated by the self-calibration circuit and synchronizing circuit respectively. Comparing to MDAC and Flash ADC power dissipation, these dissipation is negligible.

6.3.2 Constraints

Constraints are the deciding factors for the optimized design space. Constraints are derived from the input specifications. The settling time for the MDAC and ADC, Noise floor for MDAC, Area for the MDAC and ADC are the major constraints, the first two constraints can be derived for the speed and resolution of the ADC, the third constraint is a straightforward one. In addition to the above mentioned constraints, the equation for capacitor scaling and the algorithm for deciding the number of stages and resolution per stages are fed in to this equation based model.

Settling Time

The settling time for MDAC and ADC should be less than the ON-period of the clock used in that stage.

$$k_1 \tau_{MDAC} + k_2 \tau_{FLASH} + t_{slack} \leq \frac{1}{2f_s} \tag{6.8}$$

Where k_1 and k_2 are constants depend upon the noise floor and resolution. For a stage of 3 bit resolution and the noise floor for MDAC is LSB/6, then k_1 is calculated as 2.08, for a supply voltage of 3. τ_{MDAC} and τ_{FLASH} are the time constant for MDAC and ADC respectively. τ_{ADC} is given in (6.6), τ_{MDAC} can be derived as

$$\tau_{MDAC} = \frac{1}{\beta} \frac{C_L}{G_{eff}} \tag{6.9}$$

where β is the feedback factor, C_L is the load capacitance and G_{eff} ia the gain of MDAC during amplification phase.

Noise

As we have seen already, the dominant noise component is thermal noise. Total thermal noise (TTN) is given in equation (6.1). Very stringent thermal noise constraint is given to the first two stages and then the constraint is relaxed for the further stages.

Area

The total area occupied by the ADC can be derived as

$$A_{tot} = \sum_{i=0}^{NS-1} A_{MDACi} + \sum_{i=1}^{NS} A_{FLASHi} + \sum_{i=1}^{NS} A_{sci} + \sum_{i=1}^{NS} A_{synci} \tag{6.10}$$

Where A_{MDAC_i} is the area occupied by MDAC and can be give as

$$A_{MDAC} = A_{RA} + A_{C_{tot}} + A_{switches} \tag{6.11}$$

A_{RA},$A_{C_{tot}}$ and $A_{switches}$ are the area taken by residue amplifier, capacitance and switches in MDAC respectively.

A_{FLASH} in (6.10) is the area occupied by sub-ADC, and can be given as

$$A_{FLASH} = (2^{N_i} - 1)A_{comp} + A_{res} \tag{6.12}$$

A_{comp} is the area of the comparator used in sub-adc and A_{res} is the area occupied by the resistive divider.

Gain and Slew rate of the Residue amplifier

The gain and slew rate of the residue amplifier can be derived from the resolution and noise specifications of the ADC and are given in equations (6.4) and (6.5).

6.4 Design Example

Using the proposed approach a 14 bits, 30 MSPS pipelined ADC with 3 V as supply voltage is designed. The details of design approach is discussed in this section.
For the above specifications, and using the rules discussed in section IV, the exploration of design space shows that there are 68 possible combinations with the number of stages ranging from 6 to 11. Some of the possible combinations and the comparison is shown in table 6.2. table 6.2 shows the possible combinations which is having near optimum characteristics. Out of the combinations given above, if we are comparing the characteristics, we can choose the architecture with combination

Bits/stage	No.of Opamp	No. of Comparators	Power Dissipation (mW)	Speed (MSPS)	Area (mm^2)
22111111112	11	17	25.9	33	0.724
222111122	9	19	23.6	36	0.694
223111112	9	20	25.2	36	0.722
22211122	8	19	18.6	39	0.686
22311122	8	22	26.6	39	0.760
2222222	7	21	22.4	41	0.714
2233112	7	24	29.6	41	0.798
223322	6	26	30.3	43	0.836

Table 6.2: Possible combinations of Resolutions per stage

	Error w/o Capacitor scaling	Error with capacitor scaling
$error_0$	$0.125LSB_{14b}$	$0.1LSB_{14b}$
$error_1$	$0.125LSB_{12b}$	$0.1LSB_{14b}$
$error_2$	$0.125LSB_{10b}$	$0.125LSB_{14b}$
$error_3$	$0.125LSB_{7b}$	$0.15LSB_{14b}$
$error_4$	$0.125LSB_{6b}$	$0.15LSB_{14b}$
$error_5$	$0.125LSB_{5b}$	$0.15LSB_{14b}$
$error_6$	$0.125LSB_{4b}$	$0.175LSB_{14b}$
$error_7$	$0.125LSB_{2b}$	$0.175LSB_{14b}$
$Power_{total}(mW)$	18.6	12.4

Table 6.3: Comparison for Capacitor Scaling

	Uniform Gain error	Gain error optimization
$error_0$	$0.125LSB_{14b}$	$0.2LSB_{14b}$
$error_1$	$0.125LSB_{12b}$	$0.2LSB_{14b}$
$error_2$	$0.125LSB_{10b}$	$0.125LSB_{14b}$
$error_3$	$0.125LSB_{7b}$	$0.1LSB_{14b}$
$error_4$	$0.125LSB_{6b}$	$0.1LSB_{14b}$
$error_5$	$0.125LSB_{5b}$	$0.1LSB_{14b}$
$error_6$	$0.125LSB_{4b}$	$0.075LSB_{14b}$
$error_7$	$0.125LSB_{2b}$	$0.075LSB_{14b}$
$Power_{total}(mW)$	18.6	15.3

Table 6.4: Comparison for Gain error optimization

[22211222] as the optimum one.

Now for the chosen architecture, we can do the capacitor scaling. table 6.3, shows the difference between uniform stages and the stages with capacitor scaling. Stringent constraints have been given to the initial stages, so that the error does not propagate to the subsequent stages, but a relaxed constraint is given to the later stages. This results capacitor scaling in later stages and power dissipation. This reduces the error propagation from initial stages to the subsequent stages.

Jintae Kim et. al [44], proposed a optimization method to reduce the power dissipation by optimally distributing the gain error of residue amplifier. By following that method, gain error distribution for an 14-bit Pipelined ADC with the stage combination as [22211222] is given in table 6.4. This work combines both capacitor scaling and gain error optimization, so that we are getting a power of 10.9 mW, as compared to 18.6 mW, thus saving a power of 42 %.

6.5 Experimental result

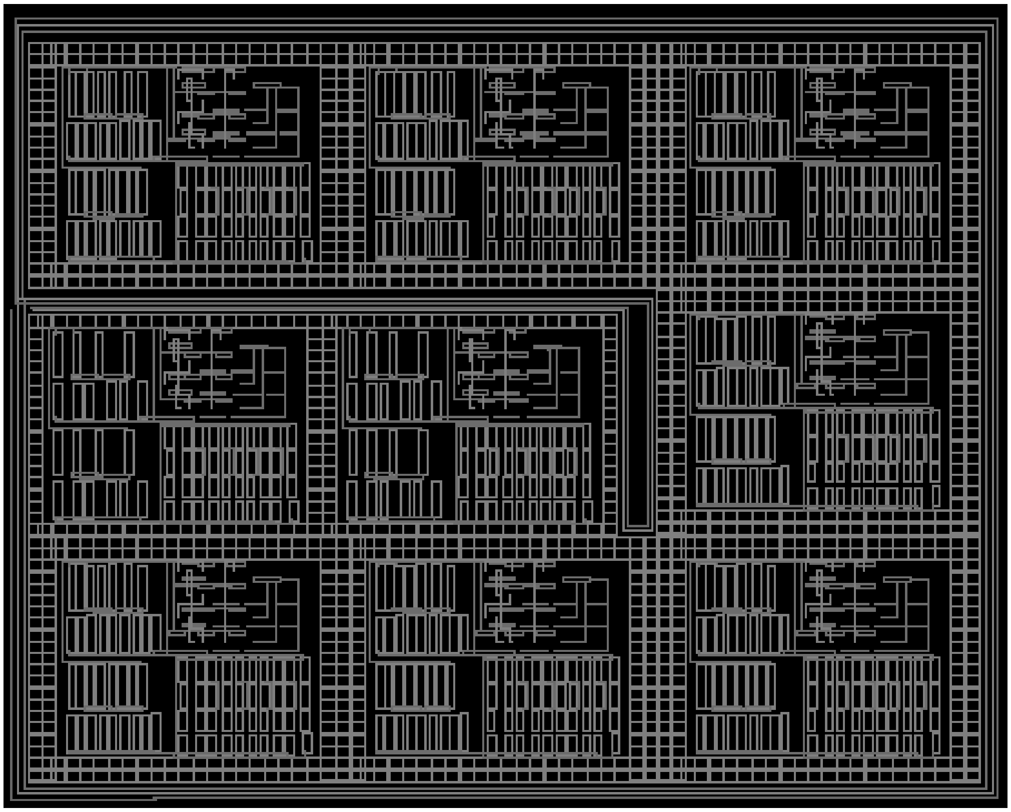

Figure 6.2: Layout Design

The Differential Non linearity (DNL) and the Integral Non Linearity (INL) for the chosen architecture are presented respectively in fig. 6.2 and 6.3. The DNL is almost $\pm 1LSB$, and the INL is $\pm 4LSB$.

The SNDR and SFDR with different sampling frequencies up to 60 at a 6 MHz input are plotted in Fig. 6.4. SNDR and SFDR are respectively 57.1 dB and 69.1 dB at 40 MS/s .

Automation design for a high speed high resolution pipelined ADC is discussed. Optimum values of number of stages and number of bits per stages are identified. Power dissipation in Pipelined ADC is optimized, by considering area, speed, noise as constraints. Using capacitor scaling and optimal distribution of gain error, power dissipation is further reduced. A detailed design example shows the effectiveness of proposed methodology.

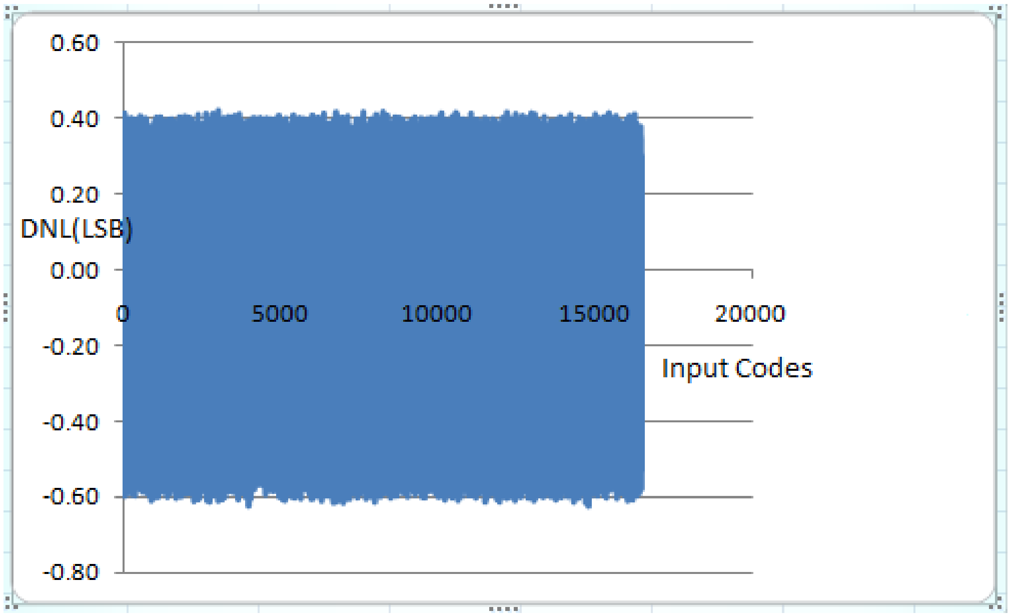

Figure 6.3: Simulated DNL vs Input codes.

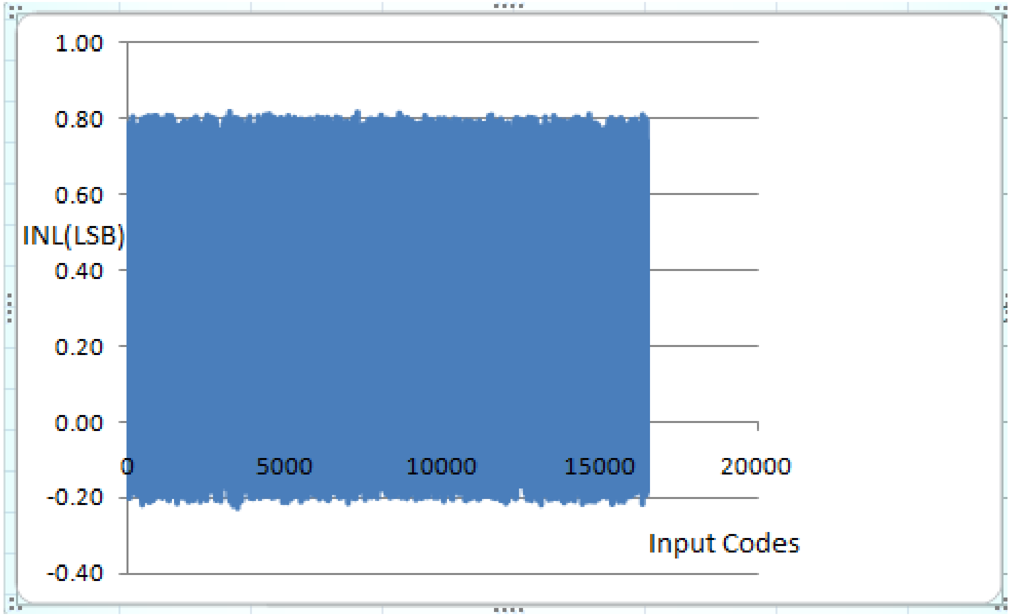

Figure 6.4: Simulated INL vs Input codes.

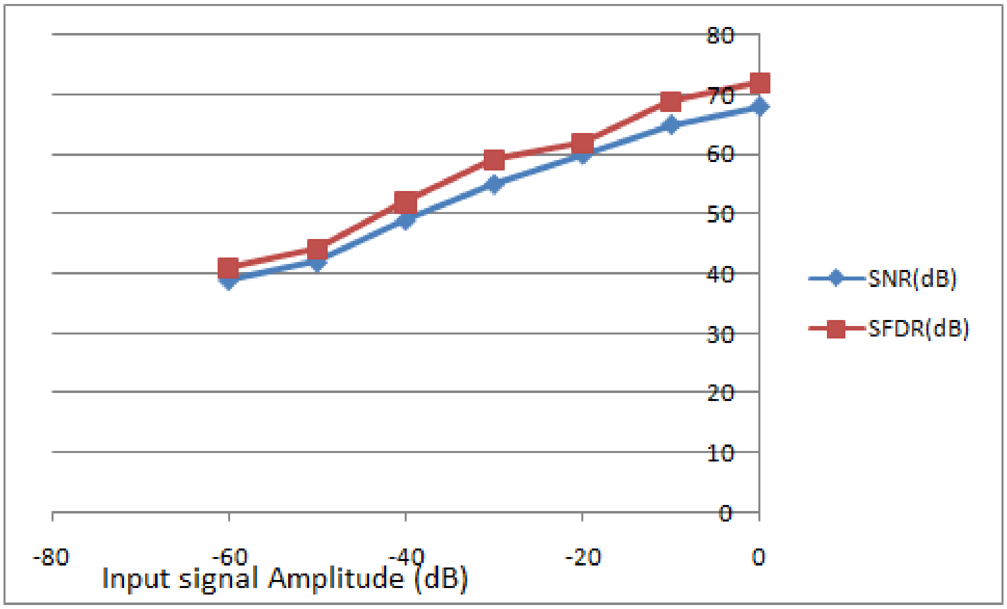

Figure 6.5: Simulated SNDR and SFDR vs sampling frequency.

Chapter 7

Conclusion

An analysis has been done to choose geometric programming as the suitable method for circuit sizing. Since the accuracy of GP depends on the accuracy of the equations used, an accurate transistor model is used.

The analysis and design automation for three different comparator architectures, namely Op-amp based, static latch based [45] and dynamic latch based comparators have been done. New delay models for static and dynamic latch based comparators are developed using a specialized decomposition method called Adomian decomposition method. A detailed analysis of simulation results are presented.

A novel capacitor array based DAC architecture is developed and a comparison with the conventional architectures has been done to justify the merits of the novel DAC. Layout for an 8-bit novel DAC is drawn and its post-layout simulation is verified. The DNL, INL, SNR and SFDR characteristics are analyzed.

An automation algorithm using GP for SAR ADC is developed [46]. The accuracy of the proposed automation algorithm is illustrated with an example.

A GP based automation algorithm for a pipelined ADC is presented. Capacitor scaling and Gain error corrections are included in the automation algorithm to get more accurate design values. The proposed automation algorithm is explained with an example.

Appendix A

Local and Global Optima

A fundamental property of convex optimization problems is that any locally optimal point is also globally optimal. To see this, suppose that x is locally optimal for a convex optimization problems, i.e., x is feasible and

$$f_0(x) = inf\ \{f_0(z)\,|\|z - x\|_2 \leq \Re\}\ \text{(A.1)}$$

for some $\Re > 0$.
Now suppose that x is not globally optimal, i.e., there is a feasible y such that $f_0(y) < f_0(x)$.
Evidently $||y - x||_2 > \Re$, since otherwise $f_0(x) \leq f_0(y)$.
Consider the point z is given by

$$z = (1 - \theta)x + \theta y$$

then

$$\theta = \frac{\Re}{2||y-x||_2}$$

Then we have $||z - x||_2 = \Re/2 < \Re$, and by convexity of the feasible set, z is feasible. By convexity of f_0, we have

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y) < f_0(x)$$

which contradicts (1). Hence there exists no feasible y with $f_0(y) < f_0(x)$, i.e., x is globally optimal.

Appendix B

How to convert geometric programming into Convex optimization problems

- Step 1:
  Write the problem in a GP form, such that For the function

$$f(\mathbf{x}) = \sum_{1}^{t} c_k x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$$

  Minimize $f_0(\mathbf{x})$
  subject to $f_i(\mathbf{x}) \leq 1$,
  $g_i(\mathbf{x}) = 1$,
  where $f_i(\mathbf{x})$ are posynomial function, and $g_i(\mathbf{x}) = 0$ are monomial function.

- step 2:
  Change variable

$$y_i = logx_i$$

$$x_i = e^{y_i}$$

  Then the optimization problem will be Minimize $f_0(y)e^{a^T y+b_k}$
  subject to $\sum_{k=1}^{k_i} e^{a_{ik}^T y+b_k} \leq 0$,
  $e^{g_i^T y+h_i} = 0$,

- step 3:
  Now take log Minimize $f_0\bar{(}y) = log[e^{a^T y+b_k}]$
  subject to $log[\sum_{k=1}^{k_i} e^{a_{ik}^T y+b_k}] \leq 0,$
  $\quad\quad g_i^T y + h_i = 0,$
  This is in convex form.

- step 4: Use interior point method to solve this convex optimization problem.

Appendix C

Adomian Polynomial based decomposition for cross-coupled differential equations

By applying KCL at node 1 and 2 we get

$$C_1 \frac{dv_0^+}{dt} + \beta_1 (v_0^+ - v_{th})^2 + \beta_1 (v_0^- - v_{th})^2 = I_1(t) \tag{C.1}$$

$$C_1 \frac{dv_0^-}{dt} + \beta_1 (v_0^- - v_{th})^2 + \beta_1 (v_0^+ - v_{th})^2 = I_2(t) \tag{C.2}$$

Substituting the values of $I_1(t)$ and $I_2(t)$, equations (1) and (2) becomes

$$C_1 \frac{dv_0^+}{dt} + \beta_1 (v_0^+ - v_{th})^2 + \beta_1 (v_0^- - v_{th})^2 = \frac{I_{max}}{t_1} t \tag{C.3}$$

$$C_1 \frac{dv_0^-}{dt} + \beta_1 (v_0^- - v_{th})^2 + \beta_1 (v_0^+ - v_{th})^2 = \frac{I_{max}}{t_1} (t_1 - t) \tag{C.4}$$

Let $u = u_0^- - v_{th}$ and $v = v_0^- - v_{th}$ then

$\frac{du}{dt} = \frac{dv_0^+}{dt}$ and
$\frac{dv}{dt} = \frac{dv_0^-}{dt}$.

Equations (C.3) and (C.4) becomes

$$C_1 \frac{du}{dt} + \beta_1 u^2 + \beta_2 v^2 = \frac{I_{max}}{t_1} t \tag{C.5}$$

$$C_1\frac{dv}{dt}+\beta_1 v^2+\beta_2 u^2=\frac{I_{max}}{t_1}(t_1-t) \tag{C.6}$$

Applying Laplace transform on both the sides

$$C_1 sU(s)+L[\beta_1 u^2+\beta_2 v^2]=\frac{C_2}{s^2} \tag{C.7}$$

$$C_1 sV(s)+L[\beta_1 v^2+\beta_2 u^2]=\frac{C_2 t_1}{s}-\frac{C_2}{s^2} \tag{C.8}$$

Rearranging the equations

$$U(s)=\frac{1}{C_1 s}\left[\frac{C_2}{s^2}-L[\beta_1 u^2+\beta_2 v^2]\right] \tag{C.9}$$

$$V(s)=\frac{1}{C_1 s}\left[\frac{C_2(t_1 s-1)}{s^2}-L[\beta_1 v^2+\beta_2 u^2]\right] \tag{C.10}$$

while taking laplace inverse transform:

$$u(t)=\frac{c_2}{c_1}\frac{t^2}{2}-L^{-1}\left[\frac{1}{s}L\left[\frac{\beta_1}{c_1}u^2+\frac{\beta_2}{c_1}v^2\right]\right] \tag{C.11}$$

$$v(t)=\frac{c_2}{c_1}t_1 t-\frac{c_2}{c_1}\frac{t^2}{2}+V_{sat}L^{-1}\left[\frac{1}{s}L\left[\frac{\beta_1}{c_1}v^2+\frac{\beta_2}{c_1}u^2\right]\right] \tag{C.12}$$

Equation (C.11) can be written as

$$u_0=\frac{c_2}{c_1}\frac{t^2}{2} \tag{C.13}$$

and

$$u_{n+1}=-L^{-1}\left[\frac{1}{s}L\left[\beta_1\sum_{n=0}^{\infty}A_m(u)+\beta_2\sum_{n=0}^{\infty}B_m(v)\right]\right] \tag{C.14}$$

In the same manner, equation (C.12) also can be written as

$$v_0=\frac{c_2}{c_1}t_1 t-\frac{c_2}{c_1}\frac{t^2}{2}+V_{sat} \tag{C.15}$$

and

$$v_{n+1}=-L^{-1}\left[\frac{1}{s}L\left[\beta_1\sum_{n=0}^{\infty}C_m(v)+\beta_2\sum_{n=0}^{\infty}D_m(u)\right]\right] \tag{C.16}$$

Where A_m,B_m,C_m and D_m are adomian polynomials and given as

$$A_m = D_m = \frac{1}{m!}\frac{d^m}{d\lambda^m}\left[N\left[\sum_{m=0}^{\infty}\lambda^i u_i\right]\right]_{\lambda=0}, m = 0,1,2... \tag{C.17}$$

$$A_0 = u_0^2 = \frac{c_2^2}{c_1^2}\frac{t^4}{4} = C_0 \tag{C.18}$$

similarly

$$B_m = C_m = \frac{1}{m!}\frac{d^m}{d\lambda^m}\left[N\left[\sum_{m=0}^{\infty}\lambda^i v_i\right]\right]_{\lambda=0}, m = 0,1,2... \tag{C.19}$$

$$B_0 = v_0^2 = \frac{c_2^2}{c_1^2}\frac{t^4}{4}(2t_1 - t)^2 + V_{sat}^2 + \frac{c_2}{c_1}t(2t_1 - t)V_{sat} = D_0 \tag{C.20}$$

$$u_1 = -L^{-1}\left[\frac{1}{s}L\left[\frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}\frac{t^4}{4} + \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t^2t_1^2 + \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t^4 - \frac{4\beta_2}{c_1}\frac{c_2^2}{c_1^2}t_1t^3 + \frac{\beta_2}{c_1}V_{sat}^2 + 2\frac{\beta_2}{c_1}\frac{c_2}{c_1}tt_1V_{sat} - \frac{\beta_2}{c_1}\frac{c_2}{c_1}t^2V_{sat}\right]\right] \tag{C.21}$$

While taking laplace transform, we get

$$u_1 = -\left[\frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{120} - \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t_1^2\frac{t^3}{12} + \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{30} - 4\frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t_1\frac{t^4}{20} + \frac{\beta_2}{c_1}V_{sat}^2\frac{t}{2} + \frac{1}{3}\frac{\beta_2}{c_1}\frac{c_2}{c_1}t_1V_{sat}t^2\right] \tag{C.22}$$

while substituting these results in equation (C.11), we would get

$$u(t) = \frac{c_2}{c_1}\frac{t^2}{2} - \left[\frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{120} - \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t_1^2\frac{t^3}{12} + \frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{30} - 4\frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}t_1\frac{t^4}{20} + \frac{\beta_2}{c_1}V_{sat}^2\frac{t}{2} + \frac{1}{3}\frac{\beta_2}{c_1}\frac{c_2}{c_1}t_1V_{sat}t^2\right] \tag{C.23}$$

similarly, v(t) can be derived as

$$v(t) = \frac{c_2}{c_1}t_1t - \frac{c_2}{c_1}\frac{t^2}{2} + V_{sat} - \left[\frac{\beta_2}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{120} - \frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}t_1^2\frac{t^3}{12} + \frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}\frac{t^5}{30} - 4\frac{\beta_1}{c_1}\frac{c_2^2}{c_1^2}t_1\frac{t^4}{20} + \frac{\beta_1}{c_1}V_{sat}^2\frac{t}{2} + \frac{1}{3}\frac{\beta_2}{c_1}\frac{c_2}{c_1}t_1V_{sat}t^2\right] \tag{C.24}$$

Bibliography

[1] J.Markus, “Higher-order incremental delta-sigma analog to digital converters,” PhD dissertation, Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2005.

[2] Qiao Yang, and Xiaobo Wu, “Power Optimization for Pipeline ADC Via Systematic Automation Design,” in *proc. Int. MultiConference of Engineers and Computer Scientists*, vol II, pp. 34-38, March 2010.

[3] T. R. Dastidar, P. P. Chakrabarti, and P. Ray, “A Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse,” *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 2, pp. 211-224, April 2005.

[4] F. El-Turkey and E. E. Perry, “BLADES: An artificial intelligence approach to analog circuit design,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 8, no. 6, pp. 680-692, Jun. 1989.

[5] G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, “Analog circuit design optimization based on symbolic simulation and simulated annealing,” *IEEE J. Solid-State Circuits*, vol. 25, no. 3, pp. 707-713, Jun. 1990.

[6] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, P. Siarry, “Analog circuit design optimization through the particle swarm optimization technique,” *Analog Integr Circ Sig Process*, vol. 63, pp. 71-82, May 2010.

[7] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, “Integer programming based topology selection of cell-level analog circuits,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 14, no. 4, pp. 401-412, Apr. 1995.

[8] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums, "Anaconda: Simulation based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 19, no. 6, pp. 703-717, Jun. 2000.

[9] M. Hershenson, S. Mohan, S. Boyd, and T. Lee, "Optimization of inductor circuits via geometric programming," in *Proc. 36th Design Automation Conference*, pp. 994-998, June 1999.

[10] M. Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 317-324, May 2002.

[11] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization.* New York: John Wiley and Sons, 1996.

[12] A. Mutapcic, K. Koh, S. Kim, and S. P. Boyd, *ggplab version 1.00 A Matlab Toolbox for Geometric Programming*. [On-line]. Available: www.stanford.edu/ boyd/cvx/.

[13] S. Boyd, S.-J. Kim, and S. Mohan, *Geometric programming and its applications to EDA problems*, Design and Test in Europe (DATE) 2005 tutorial. [On-line]. Available: www.stanford.edu/ boyd/date05.html.

[14] J. Fishburn, and A. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. IEEE International Conference on Computer-Aided Design (ICCAD-85) Digest of Technical Papers*, pp. 326-328, Nov. 1985.

[15] C. Chu and D. Wong, "Closed form solutions to simultaneous buffer insertion/sizing and wire sizing," *ACM Transactions on Design Automation of Electronic Systems*, vol. 6(3), pp. 343-371, May 2001.

[16] K. Kortanek, X. Xu, and Y. Ye, "An infeasible interior-point algorithm for solving primal and dual geometric programs," in *Math Programming*, vol.76, New York: Springer-Verlag, pp. 155-181, Apr 1996.

[17] D. A. Johns and K. Martin, *Analog Integrated Circuit Design.* New York:John Wiley and Sons, 1997.

[18] S. DasGupta and P. Mandal, "An Automated design approach for CMOS LDO regulators," in *Proc. Asia and South Pacific Design Automation Conf.,*, pp. 510-515, Jan. 2009.

[19] W. Daems , G. Gielen, and W. Sansen, "Simulation-based Generation of Posynomial Performance Models for the Sizing of Analog Integrated Circuits," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 22(5), pp. 517-534, May 2003.

[20] T. R. Dastidar, P. P. Chakrabarti, and P. Ray, "A Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse," *IEEE Tran. on Evolutionary Computation*, vol. 9, no. 2, pp.211-224, April 2005.

[21] R. J. Baker, *CMOS: Mixed-Signal Circuit Design.* Wiley-IEEE, 2nd Edition, 2009.

[22] S. Somali and G. Gokmen, "Adomian Decomposition method for non-linear starm-liouville problems," *Surveys in Mathematics and its applications*, vol.2, pp. 11-20, 2007.

[23] D.J. Evans and K. R. Raslan, "The Adomian Decomposition method for solving delay differential equations," *Int. Journal of Computer Mathematics*, vol. 00, pp. 1-6, June 2004.

[24] F. M. Allan, "Derivation of the Adomian Decomposition method using the homolopy analysis method," *Elsevier-Applied Mathematics and Computation*, vol. 190, pp. 6-14, Jan. 2007.

[25] E. Celik, M. Bayram and T. Yeloglu, "Solution of Differential- Algebraic Equations by Adomian Decomposition Method," *Int. Journal of Pure and Applied Mathematical Sciences*, vol. 3, pp. 93-100, June 2006.

[26] H.-C. Chow and Y.-H. Chen, "1V 10-bit Successive Approximation ADC for Low Power Biomedical Applications," in *Proc. 18th IEEE European conf. on circuit theory and design*, pp. 196-199, Aug. 2008.

[27] M. E. M. A. Yakoub, M. Sawan, and C. Thibeaulty, "A Neuromimetic Ultra low-power ADC for Bio-Sensing Applications," in *Proc. IEEE North-west Workshop on Circuits and Systems and TAISA Conf.*, pp. 1-4, July 2009.

[28] M. D. Scott, B. E. Boser, and K. S. J. Pister, "An Ultralow-Energy ADC for Smart Dust," *IEEE J. Solid-State Circuits,* vol. 38, no. 7, pp. 1123-1129, July 2003.

[29] F. Chen, A. P. Chandrakasan, and V. Stojanovi, "A Low-power Area-efficient Switching Scheme for Charge-sharing DACs in SAR ADCs," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 1-4, Sept. 2010.

[30] B. P. Ginsburg and A. P. Chandrakasan, "500-MS/s 5-bit ADC in 65-nm CMOS With Split Capacitor Array DAC," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 737-749, April 2007.

[31] B. P. Ginsburg and A. P. Chandrakasan, “An Energy-Efficient Charge Recycling Approach for a SAR converter With Capacitive DAC,” in *Proc. IEEE Int. Symp. Circuit and systems(ISCAS’05)*, pp. 184-187, May 2005.

[32] J. Park, H.-J. Park, J.-W. Kim, S. Seo, P. Chug, “A 1mW 10-bit 500KSPS SAR A/D Converter,” in *Proc. IEEE Int. Symp. on Circuits and Systems,*, pp. 581-584, May 2000.

[33] S. Mortezapour and E. K. F. Lee, “A 1-V, 8-Bit Successive Approximation ADC in Standard CMOS Process,” *IEEE J. Solid-State Circuits*, VOL. 35, NO. 4, pp. 642-646, April 2000.

[34] M. Davidovic, G. Zach, and H. Zimmermann, “A 12-bit Fully Differential 2MS/s Successive Approximation Analog-to-Digital Converter with Reduced Power Consumption,” in *Proc. IEEE 13th Symp. on Design and Diagnostics of Electronic circuits and systems(DDECS),* , pp. 399-402, April 2010.

[35] E. Culurciello, and A. G. Andreou, “An 8-bit 800-W 1.23-MS/s Successive Approximation ADC in SOI CMOS,” *IEEE Trans. on circuits and systemsII,* vol. 53, no. 9, pp. 858-861, Sept 2006.

[36] S. H. Lewis, “A 10-b 20-MS/s analog-to-digital converter,” *IEEE J. Solid-State Circuits,* vol. 27, pp. 351-358, Mar. 1992.

[37] D. Cline and P. Gray, “A power optimized 13-b 5-Msamples/s pipelined analog-digital converter in 1.2 um CMOS,” *IEEE J. Solid-State Circuits,* vol. 31, pp. 294-303, Mar. 1996.

[38] T. Cho and P. Gray, “A 10 b 20 Msamples/s, 35mW pipeline A/D converter,” *IEEE J. Solid-State Circuits,* vol. 30, pp. 166-172, Mar. 1995.

[39] J. Goes, J. Vital, and J. Franca, “Systematic design for optimization of high-speed self-calibrated pipelined A/D converters,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.,* vol. 45, no. 12, pp. 1513-1526, Dec. 1998.

[40] J. Goes, J. C. Vital and J. E. Franca, *Systematic Design for Optimization of Pipelined ADCs.* Kluwer Academic Publishers, 2001.

[41] M. Nakaya, T. Kumamoto, T. Miki and Y. Horbia, “Analysis of reference-tap-voltage fluctuation in flash A/D Converters,” *Trans. Electrical and Computer Engineering of Japan,* vol. J69-C, no. 3, pp. 237-244, Mar. 1986.

[42] P. T. F. Kwok, H. C. Luong, “Power optimization for Pipeline Anaog-to-Digital Converters,” *IEEE Tran. Circuits and Systems - II: Analog and Digital Signal Processing,* vol. 46, no. 5, pp. 549-553, May 1999.

[43] T. Cho and P. Gray, “A 10 b 20 Msamples/s, 35mW pipeline A/D converter,” *IEEE J. Solid-State Circuits,* vol. 30, pp. 166172, Mar. 1995.

[44] J. Kim, S. Limotyrakis, and C.-K. K. Yang, “Multilevel Power Optimization of Pipelined A/D Converters,” *IEEE Tran. VLSI Systems*, vol.19(5), pp. 832-845, May 2011.

[45] A. Purushothaman, C. D. Parikh, “Design of static latch based comparator using power constrained optimization,” *Indian Journal of VLSI and Electronic system Design,* pp.212-215, July 2011.

[46] A. Purushothaman, C. D. Parikh, “Automating the design of Successive Approximation Register Analog to Digital Converters,” *Indian Journal of VLSI and Electronic system design,*, pp. 156-163, July 2012.

[47] M. Saberi, R. Lofti, K. Mafinezhad and W. A. Serdign, “Analysis of Power Consumption and linearity in Capacitive Digital to Analog Converters Used in Successive Approximation ADCs,” *IEEE Transactions on Circuits and Systems - I*, vol.58, no.5, pp. 1736-1748, Aug. 2011.