HIGH LEVEL MODELING OF SIGMA-DELTA ANALOG TO DIGITAL CONVERTERS

by

Selçuk Talay

B.S., Electronics and Communication Eng., Istanbul Technical University, 1998

M.S., Electrical and Electronics Engineering, Bogazici University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisor Professor Günhan Dündar for his guidance, endless support and patience throughout my thesis. He provided the motivation to me in my research and has always been available to advise me. I am very grateful for his mentoring throughout years, which is not limited with research topics.

I would like to thank Professor Franco Maloberti for his valuable contribution and the time he devoted to my thesis. Also, I would like to express my sincere gratitude to Professor Ömer Cerid, Professor Avni Morgül and Professor Cem Ersoy for their kind interest and contribution to my research throughout my thesis

I also want to thank Assistant Professor Alper Demir for serving on my qualifying exam committee.

I would like to thank Engin Deniz, Orkun Sağlamdemir and Ömer Yetik for their support and contribution.

I am very grateful to all the members of BETA for their help and support. My special thanks goes to Faik Başkaya, Dağhan Gökdel, Baykal Sarıoğlu and Okan Zafer Batur for making our research environment such an enjoyable place.

The financial support provided by Science and Technology Research Council of Turkey (TÜBİTAK) via project 103E039 is gratefully acknowledged.

I am forever indebted to my parents, my parents in law, my brothers and my brother in law for their understanding, endless patience and encouragement when it was most required. I could not have completed this thesis without their love and support.

Finally, this thesis is dedicated to my dearest wife, Defne Kayrak Talay. Without her, I would never find enough encouragement and motivation for this theses.

ABSTRACT

HIGH LEVEL MODELING OF SIGMA-DELTA ANALOG TO DIGITAL CONVERTERS

This thesis discuss high-level modeling of Sigma-Delta ADC's, development of a design automation tool which operates at system level, and other related software which are required for accurate operation of this tool. There are some models available in the literature which are suitable for a design automation system. These models are taken as a starting point in the modeling effort in this thesis. Also, a Sigma-Delta ADC design automation tool was developed which involves novelties such as performance estimator and architecture selection. With the contribution of the performance estimator module, the tool provides an extensive design environment to the users for the design of Sigma-Delta analog-to-digital converters. The developed models and their effects are clearly presented with examples. Design examples for 0.5μm and 0.35μm AMS technologies are provided for proving the accuracy and flexibility of the design automation tool. In addition, a Sigma-Delta ADC IC was designed and fabricated.

ÖZET

SİGMA-DELTA ANALOG SAYISAL DÖNÜŞTÜRÜCÜLERİN YÜKSEK SEVİYEDE MODELLENMESİ

Bu tez, Sigma-Delta analog-sayısal çeviricilerin (ADÇ) yüksek seviyede modellemesi, bu modellerin kullanıldığı, sistem seviyesinde çalışan bir tasarım otomasyon yazılımının geliştirilmesi ve bu otomasyon yazılımının işleyebilmesi için gerekli olan yardımcı araçların geliştirilmesini kapsamaktadır. Literatürede tasarım otomasyonu için daha önceden geliştirilmiş bazı modeller bulunmaktadır. Bu tezde, bu modeller başlangıç noktası olarak alınmış ve gerekli modeller geliştirilmiştir. Geliştirilen Sigma-Delta ADÇ tasarım otomasyon yazılımı literatürde mevcut olmayan yenilikler ve özellikler ile donatılmıştır. Bunların en önemlileri performans öngörücüsü ve yapı seçicisidir. Performans öngörücüsünün de yardımıyla , geliştirilen yazılım, kullanıcılara Sigma-Delta ADÇ tasarımı konusunda son derece kapsamlı bir tasarım ortamı sunmaktadır. Otomasyon yazılımı için gelitirilen modeller ve etkileri açık bir şekilde bu tezde sunulmuştur. Ayrıca, 0.5μm ve 0.35μm AMS teknolojileri için örnekler sunulmuş ve bu yolla doğrulukları ve esnek çalışma şekilleri gösterilmiştir. Bunlara ek olarak bir adet Sigma-Delta test tümdevresi de tasarlanmış ve üretilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

C_f	Feedback capacitor
C_f	Sampling capacitor
e_T	Switch thermal noise voltage
f_d	Nyquist frequency
f_in	Input signal frequency
f_s	Sampling frequency
m	Offset error
M	Oversampling ratio
P_Q	In-band noise power
r_o	is the OPAMP output resistance
R_S	are switch on resistances
$S_E(f)$	Noise power spectral density
t_1	Charging time
t_2	Transfer time
V_{cm}	Common-mode voltage
V_{in}	Input voltage
V_n	Total RMS noise voltage referred to op-amp input
v_o	Output voltage
α	Integrator leakage
δ	Error in the sampling time instance / Sampling time uncertainty
δ_1	Charging error
δ_2	Charge transfer error
δ_3	The error coming from the finite gain of the amplifier
Δ	Quantizer stepsize
$\Delta\tau$	Standard deviation of the sampling time uncertainity
Φ_1	Clock phase 1
Φ_2	Clock phase 2
τ	Integrator time constant

ADC	Analog-to-Digital Converter
BTS	Basic Two Stage
CAD	Computer Aided Design
CMFB	Common-Mode Feedback
CMOS	Complementary Metal Oxide Semiconductor
CT	Continuous-Time
DAC	Digital-to-Analog Converter
DR	Dynamic Range
DT	Discrete-Time
ENOB	Effective Number Of Bits
FPGA	Field Programmable Gate Array
FS	Full Scale
HD	Harmonic Distortion
IC	Integrated Circuit
INL	Integral Non-Linearity
LSB	Least Significant Bit
MASH	Multi-Stage Noise-Shaping
MSE	Mean Square Error
MSB	Most Significant Bit
NTF	Noise Transfer Function
OSR	Oversampling Ratio
PE	Performance Estimator
PSD	Power Spectral Density
SNDR	Signal-to-Noise and Distortion Ratio
SNR	Signal-to-Noise Ratio
SR	Slew-Rate
STR	Signal Transfer Function

1. INTRODUCTION

Analog-to-digital converters (ADC) provide the link between the analog world and the digital world of signal processing [1], computing, and other digital data collection or data processing systems. Nowadays, electronic signals are processed increasingly in the digital domain. Since most signals are analog in the world we live, the importance of ADC's has been increasing considerably. We can safely say that ADC's will keep on being key components in large systems.

In the past, ADC's have been implemented by using clever analog circuits that consume considerable time to design. Although the digital processing hardware was not fast and strong enough for carrying out some of the desired tasks in the past, digital processing power has become cheaper and easier to implement with advances in CMOS technology. Hence, this has increased the interest in analog-to-digital conversion techniques and circuits. Processing in digital domain, however, imposes various requirements on the ADC. These include conversion rate, resolution, power consumption, and dynamic range. Unfortunately, none of the available ADC circuits can excel in all specifications; hence, different ADC circuits have been developed for this purpose.

There are various types of ADC's available both in the market and in the literature. These architectures differ from each other by their speed, accuracy, power, cost, etc. The selection of the most appropriate type is a hard problem. Although there are some other ADC architectures such as successive approximation or Subranging ADC's, the most important architectures are flash, pipeline and Sigma-Delta ADC's. These three cover most solutions on the solution space described on resolution vs. speed as illustrated in Figure 1.1 [2]. As a result, for most of the designs, a solution with these architectures can be found.

The fastest of all types of high-speed analog-to-digital converters is the flash or parallel type converter whose block diagram is given in Figure 1.2. The flash converter is considered to be the fastest ADC because conversion takes place in a single cycle;

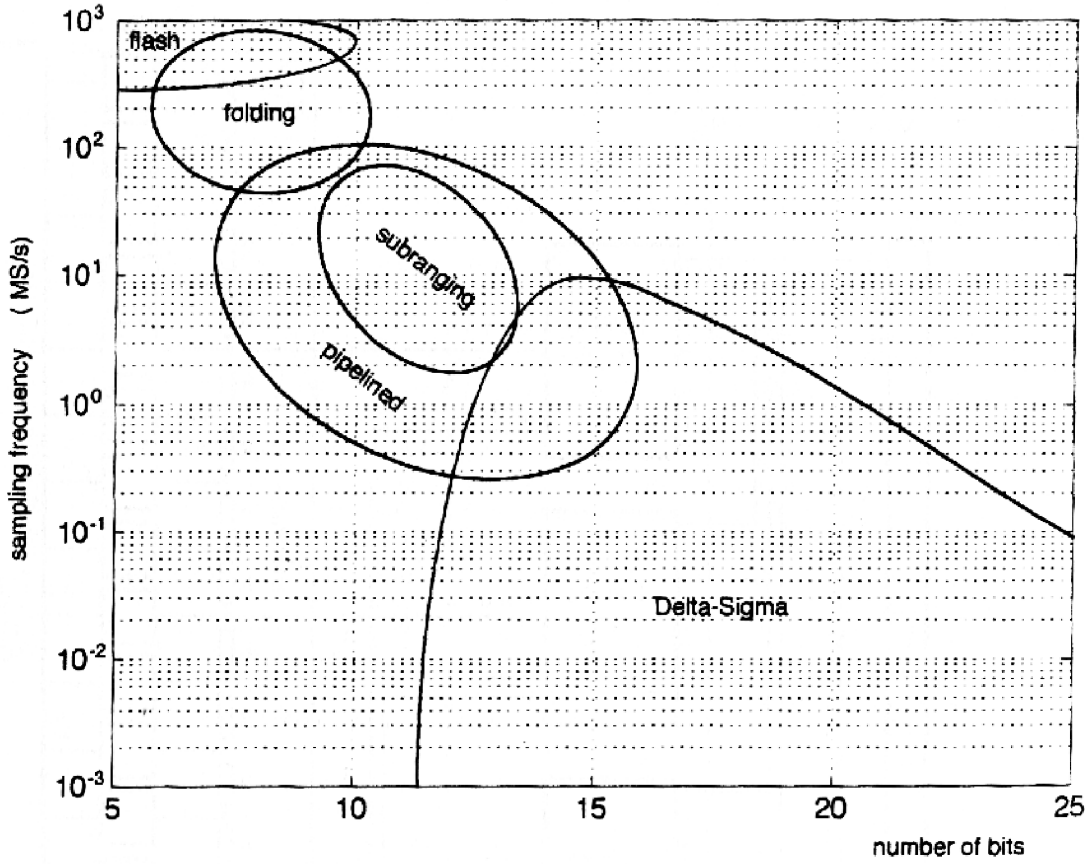

Figure 1.1. Resolution vs sampling rate

hence the name, "Flash". The resolution of flash converters tends to be limited to around eight bits due to the fact that the amount of circuitry doubles every time the resolution is increased by one bit. Another drawback of this architecture is that the design of flash converters demands a high level of matching between the parallel comparators and resistors in the resistor ladder, which is a challenging task.

The pipelined ADC consists of a number of consecutive stages. A general pipelined architecture is given in Figure 1.3. The stages are similar in their functionality, where each stage generally resolves only one or two bits. Each individual stage consists of a sample and hold, a low resolution flash ADC, a low resolution digital-to-analog converter (DAC), and a summing stage including an interstage amplifier for providing gain. The outputs of each stage are combined in the output latch. Stage one takes a sample of the input voltage and makes the first coarse conversion. The result is the most significant bit (MSB) and its digital value is fed to the first latch. As the residue

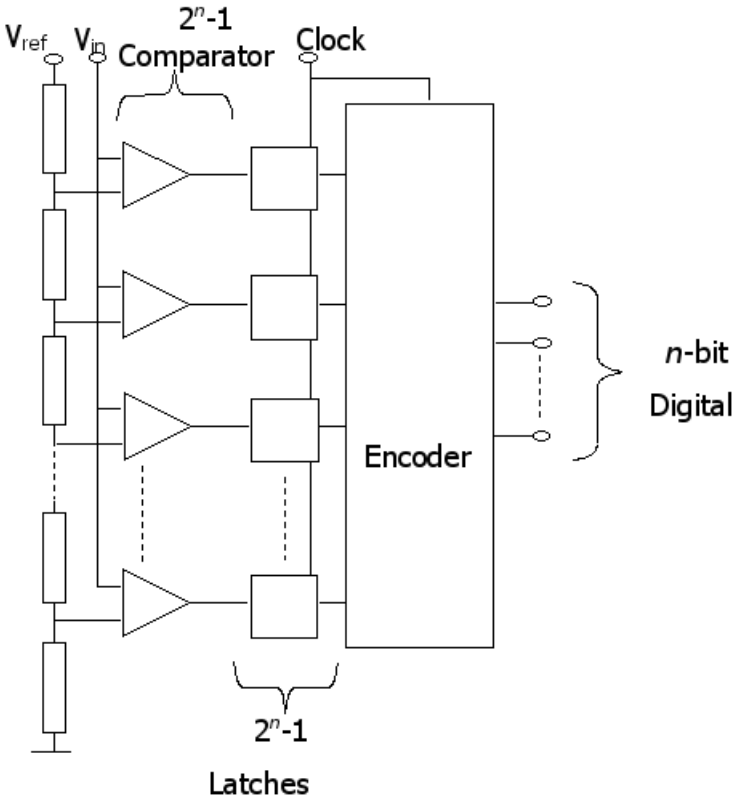

Figure 1.2. Flash ADC block diagram

of the first stage gets resolved in the subsequent n-stages, the MSB value is rippled through n latches in order to coincide with the end of the conversion of the last stage. Then, all data bits are latched in the output register and the output becomes available. Since a pipeline design strategy is used, the throughput of the ADC is high. Also, the number of blocks used in the ADC is significantly decreased as compared to the Flash ADC by dividing the ADC into stages.

Another alternative for A/D conversion is Sigma-Delta ADC's. They are very suitable to achieve resolutions of up to 20 bits. They exchange the loss of accuracy inherent to analog circuits in digital processes for faster signal processing and more digital circuitry. In other words, more accurate ADC's can be achieved by complex and fast digital circuitry. Sigma-Delta modulation is a robust technique for analog-to-digital conversion. Other ADC architectures require precisely matched components in order to attain high accuracy. Also, it can easily be said that the resolution of the ADC is linearly proportional with the number of components that have to be matched

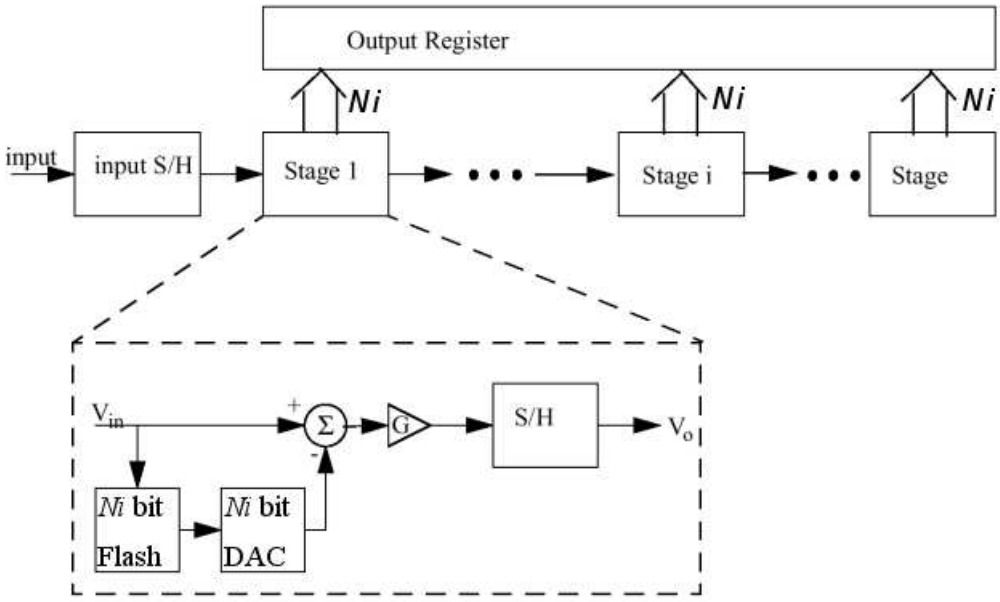

Figure 1.3. Pipeline ADC block diagram

accurately. The advantage of the Sigma-Delta architecture is that the matching requirement is very relaxed as compared to other architectures. The use of feedback allows the utilization of imprecise components.

Sigma-Delta ADC's have been enjoying increasing interest through the last decade. The main reason of interest for this architecture is its ability to achieve high resolutions with a decrease in conversion rate [3]. Also, efficient Sigma-Delta ADC designs can transfer complexity from analog circuitry to digital, allowing designers to design low-power ADC's. These features make them a good candidate for various applications especially in telecommunication [4, 5]. However, efficient design of Sigma-Delta ADC's is still not a completely solved problem. Although the number of distinct blocks in Sigma-Delta ADC's is small, which is a desired attribute, they can be connected in many different configurations, thus rendering the choice of an optimum architecture and the design parameters of each block quite difficult. Designers select architectures for specific performances, but the optimality of the selection is not guaranteed. The existence of numerous possibilities for different connection schemes of building blocks introduces a design choice at the very beginning of the design flow; the selection of the appropriate connection scheme or configuration. The design space has many dimensions and being able to find a near optimal solution has crucial importance. One

of the novelties of the presented work in this thesis is the architecture selection feature, which eliminates configurations with low performance. Thus, the designer may overcome many problems or eliminate most candidate configurations in the beginning.

Regarding all design choices and possible iterations made in absence of an assisting design automation tool, the time spent for designing a Sigma-Delta ADC is very large compared to the time required for a digital design, which has a well-defined design flow and design tools. The increasing complexity of Sigma-Delta ADC's and time to market limitations create a demand for Sigma-Delta ADC design automation tools. However, there are no complete top-down design automation tools available for Sigma-Delta ADC design. Some Sigma-Delta ADC design automation tools have been presented [6, 7, 8, 9, 10, 11, 12, 13] in the literature, most of them having been developed in the academia and having found limited use in the industry. A successful commercial tool (MIDAS) is reported in [14]. Also, there are many reported studies in the literature which are focused on selection of the appropriate Sigma-Delta configuration [15, 16, 17, 18, 19]. Finding the optimal or near optimal configuration for the desired specifications is a very complex problem. The reason is not only that the design may contain different architectures, but also that the architectures themselves may have different configurations. Finding the optimal component values for these configurations is a different problem. In other words, even if the ADC type has been chosen a priori (i.e., Sigma-Delta), the architecture will have different possible configurations (e.g. 4^{th} order 2-2, 4^{th} order 2-1-1, 4^{th} order single loop, parallel) which increases the complexity of choosing the ADC.

Synthesis environments or design space search tools mostly contain dedicated tools for behavioral simulation. They are also equipped with some optimization capabilities at system level. However, a design automation tool that works from the system level down to the layout level and that can model and optimize multiple architectures has not been yet reported. The available tools are mostly focused on a selected architecture and try to generate transistor level data for these blocks for which the schematic is already known or given earlier. Even so, the presented approaches demonstrate that the design effort decreases with the assistance of tools or models developed

for designing Sigma-Delta ADC's [20, 21]

In order to develop such systems, accurate models of the architectures are required. There are tools reported in the literature that are capable of designing ADC's down to circuit level. However, these tools can only design a single ADC architecture, and are more suitable for parameter optimizations. Also, there are some behavioral simulators for very limited number of ADC architectures such as flash and pipeline architectures [22]. The tools developed for Sigma-Delta ADC's are generally behavioral simulators. In addition, these simulators are developed for a limited number of Sigma-Delta architectures, since accurate models of every possible architecture are not available.

1.1. Motivation

There is an ongoing research at several institutions, including our research laboratory, whose aim is to develop a complete analog design automation tool, which will cover all levels through system level to layout level. However, this is a challenging task since there is no well-defined design flow available for analog circuits similar to the well established flow for digital circuits. For this reason, the initial task is the definition of the design flow. The design flow proposed by our group is given in Figure 1.4. There are three abstract levels: system level, circuit level, and layout level. For general analog design, another level, the functional level, can be inserted between the two top levels [23]. However, these levels should be supported by libraries or performance estimators. Another general analog design flow is given in Figure 1.5 [24] which has more levels. The design system presented in Figure 1.4 is under development and many tools have already been implemented [25]. The aim of this thesis is to develop a Sigma-Delta ADC design tool at system level. Some architectures (Flash and Pipeline) have been implemented so far for the available design automation system and the Sigma-Delta ADC tool will cover the remaining sections in ADC design space defined by resolution vs. conversion rate as given in Figure 1.1.

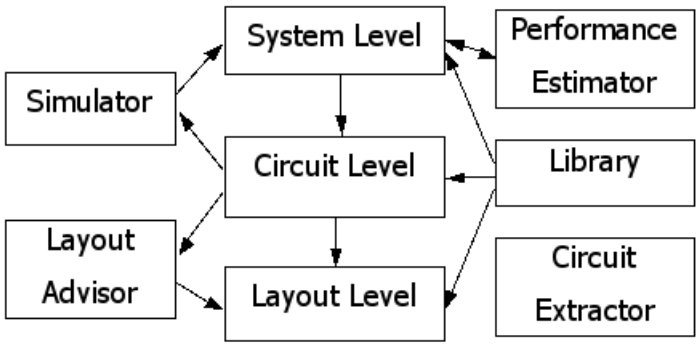

Figure 1.4. Analog design automation flow

Generating parameters from the system level tool to be used in the circuit level is a challenging task. Also, the designer should have enough flexibility for choosing the level of interaction throughout the design process. Thus, many different approaches should be made available to the user at system level. Three different usage models were planned for the ADC design tool discussed in this thesis. First one is the automation mode, which completely works with automation software. Another mode will be semi-automatic, which is suitable for designs where some design blocks are available and the rest are to be designed. The last one will be for designs where an overall design is available and performance verification is desired.

The tool should also be flexible regarding the choice of architecture for the ADC, which means that the new architectures can also be used with this tool.

As a summary, the goal of this thesis is to develop accurate models for Sigma-Delta ADC's and a tool that takes advantage of these models in designing a Sigma-Delta ADC, which we can process in a complete analog design automation tool. This task requires modeling of architectures, developing the methodology in order to perform an architecture selection, and optimization of parameters within a given or selected architecture.

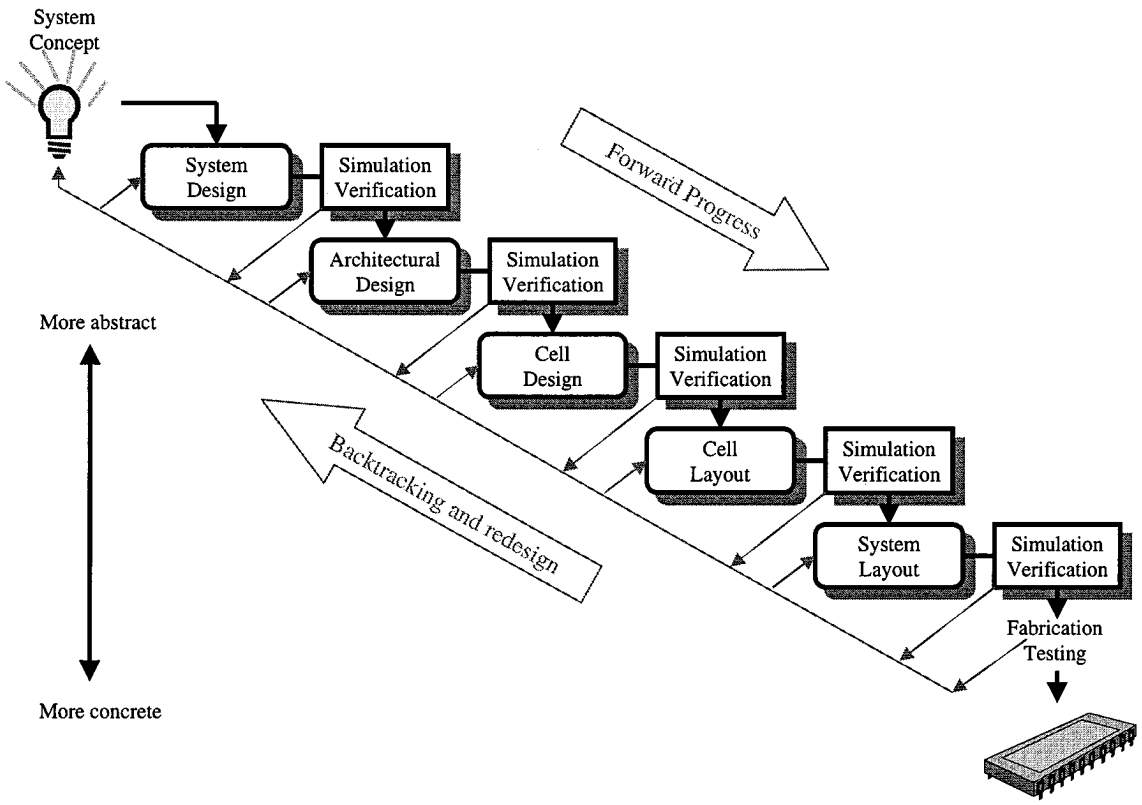

Figure 1.5. High-level view of design process

1.2. Methodology

In order to achieve the goal of the thesis, several milestones were defined. These milestones include modeling of Sigma-Delta ADC's, development of a design automation system for Sigma-Delta ADC's, and verification of developed models and software.

There are some models for Sigma-Delta ADC's available in the literature. These models were developed for mainly two approaches available in the literature: analytical modeling and behavioral modeling. Hence, for modeling a Sigma-Delta ADC, one of them should be selected. Both approaches were investigated by means of developing both primitive behavioral and analytical models. Behavioral models were developed in MATLAB and Simulink environment. Analytical models were developed and coded in C++. The main problem of the behavioral approach, despite the fact that developing models is relatively easy, is that it requires iterations when used in the design automation system. Hence, searching a solution in the design space would take too much time. Since one of the objectives of this thesis is to provide the designer with

a solution from a solution space which has many dimensions, this approach was not preferred and analytical approach was used. However, MATLAB models were used to compare the results for model verification. Also, for architecture synthesis module, MATLAB was used because of necessity since symbolic toolbox is required for the operation. After the selection of the analytical approach, each error source was defined that contributes to noise in calculation of the main performance metric, SNDR. Then, these models were analytically defined and models were developed such that they may be incorporated into the design automation system.

The next step is the development of a design automation system which utilizes the models discussed above. The tool should be developed in such a way that it can provide the desired level of interaction to the designer depending on the needs of the designer. The designer may need to verify a design or would like to use previously designed blocks. In other words, the tool should be an assisting software to the designer, who chooses the level of interaction throughout the design, rather than a block box synthesizer. Hence, three modes of operation were defined and the tool was developed considering these modes. Since there is no iteration process, the result may be achieved in a couple of minutes.

Verification of models by means of test circuits and IC implementation should be also performed. Results can be compared with available examples in the literature. Thus, a test IC was designed by utilizing the models developed. Hence, model accuracy was verified by physical implementation. The developed tool was used to calculate performance values of some reported test IC's to compare the accuracy. Also, this calculated performance was compared with the output of other tools in the literature .

1.3. Key Contributions

This section summarizes the contributions made within the framework of this thesis.

- Development of models for Sigma-Delta ADC's. The achievements were reported

in [26, 27, 28]

- Development of a Sigma-Delta ADC design automation tool based on the developed models. The achievements were reported in [29]
- Development of a performance estimator tool and integration of the tool into the Sigma-Delta ADC design automation tool. The achievements were reported in [30]
- Development of an architecture synthesis tool. The achievements were reported in [31, 32]
- Development of an optimization tool in MATLAB for architecture selection.
- Design of a test IC. The achievements were reported in [29, 33]

1.4. Thesis Organization

The thesis is organized as follows:

Chapter 2 introduces Sigma-Delta ADC's and available implementation methods in the literature. Hence, a background is provided for Sigma-Delta ADC's with in the chapter.

Chapter 3 is dedicated to the modeling effort of the Sigma-Delta ADC's. The approaches available and models developed within the framework of this thesis as well as models available in the literature were presented. These models are utilized in the following chapters.

Chapter 4 presents the design automation system developed for the Sigma-Delta ADC design. The methods and algorithms used are presented and design flow with the developed tool is also provided in this chapter

Chapter 5 introduces the performance estimator module and provides examples for the developed tool.

Chapter 6 provides examples for the developed design automation system

Chapter 7 presents an integrated circuit which contains two Sigma-Delta ADC's. Design details are provided in this chapter.

Chapter 8 concludes the thesis.

2. SIGMA-DELTA ADC'S

Although the architecture targeted is known as Sigma-Delta converter, it is still referred as Delta-Sigma converters in some sources. In addition to "Sigma-Delta ADC"or "Delta-Sigma ADC", other terms often found in the literature are oversampling and noise shaping ADC's. The last two terms in fact show the main principles of Sigma-Delta ADC's. The former spreads the quantization noise power over a bandwidth equal to the sampling frequency, which is much greater than the signal bandwidth. In the latter, the modulator behaves as a low-pass filter on the signal, and as a high-pass filter on the noise, thus "shaping"the quantization noise so that most of the energy will be above the signal bandwidth. A digital low-pass filtering stage then greatly attenuates out-of-band quantization noise, and finally, downsampling brings the sampled signal to the Nyquist rate.

Sigma-Delta modulation is a robust technique for analog-to-digital conversion. Other ADC architectures require precisely matched components or in other words precise ratio of components in order to attain high accuracy. Also, it can easily be said that the resolution of the ADC is linearly proportional with the number of components that have to be matched accurately. The advantage of the Sigma-Delta architecture is that it does not require that close matching. The use of feedback allows the utilization of imprecise components.

One way of improving the resolution of a Nyquist rate converter is oversampling the input as in Sigma-Delta ADC's. Although the total quantization noise power does not change, its frequency spectrum does. The same amount of noise is spread over a wider range leaving a small amount in the baseband. Figure 2.1 shows the noise spread over the spectrum. The shaded area is the noise power of a Nyquist rate converter.

Increasing the sampling rate reduces the quantization noise power by a factor OSR, which is the oversampling ratio. This analysis assumes that the quantization

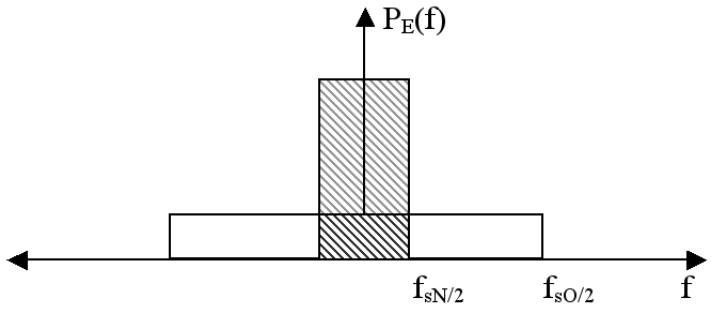

Figure 2.1. Noise power of an oversampling converter

noise spectrum is white; however, this is not the case in practical systems.

Although commercial Sigma-Delta ADC and DAC's have been available for more than a decade, the primary application of these converters has been in digital audio. The low bandwidths in digital audio applications have made oversampled converters reasonable to use. It is only recently, as we benefit from the increased speed of submicron devices, that Sigma-Delta modulators are exploited for wider band systems such as wireless RF communications.

Sigma-Delta modulation was first introduced by Inose, Yasuda and Murakami in 1962 [34]. During the following years, especially the last decade, a considerable effort has been devoted to this subject. A very complete review of these efforts including different methods and theory used for Sigma-Delta ADC and DAC is presented in [3]. A more recent version of this book was published in 1998 [35]. The number of different configurations and the difficulty to compare these configurations with each other has increased the motivation on the modeling of this architecture. In addition to this, the development efforts of an ADC design automation system met with the absence of an accurate model of the architecture. Thus, despite the efforts to design various Sigma-Delta ADC's with better performance values, much research has concentrated on the modeling of Sigma-Delta ADC's in order to have an insight of the architecture.

The reader is referred to [3] especially for a very extensive list of references, mostly papers published in technical journals or presented at conferences. Many of the papers concentrate on theoretical issues such as stability, limit cycles, quantization

noise spectra, analysis of nonlinear behavior, etc. Also, a number of papers discuss in detail the structure of new architectures and techniques with the development of new devices as of 1992. However, many books were published since then, which cover different aspects of Sigma-Delta ADC's [7, 36, 37, 38, 39, 40, 41, 42, 43, 44].

The above paragraphs sketch previous research activities about Sigma-Delta ADC's. However, the research has been multiplied in the last few years [45, 46, 47, 48, 49, 50, 51, 52]. Although there is a deep theory behind the operation of Sigma-Delta ADC's, the operation principles are very basic. In Figure 2.2, the operation of a Sigma-Delta ADC is illustrated. Constant input 3/7 is a popular example for showing how the Sigma-Delta ADC works. If the iterations are carried out via simulation, the result should be 3 bits with a value of 1 out of every 7 bits. This simulation is shown in Figure 2.2. The first plot is the input signal and the last one is the output. However, it is not easy to design a high-performance Sigma-Delta ADC most of the time, which involves many design decisions.

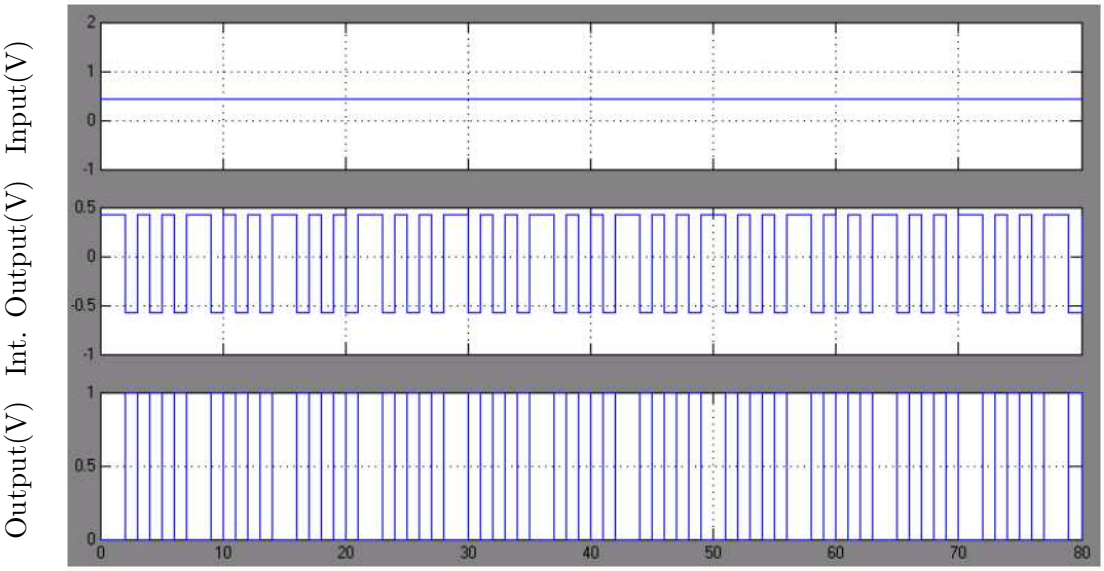

Figure 2.2. Output of Sigma-Delta for 3/7 as an input

The implementation of a Sigma-Delta ADC for specific performance criteria is not unique. That is, there are many approaches available for implementing the theory. Most popular and easiest way is to use sampling for the input signal and use a discrete

time loop filter (integrator). One other popular method is called continuous-time implementation where the integrator is defined in s-domain and quantizer and DAC are clocked. The following sections present these methods mainly remaining in the block level. In addition, it should be noted that these methods can be implemented both in voltage-mode or in current-mode. Combination of these is also possible as in [53] where the input is current and output is voltage.

2.1. Discrete Time Sigma-Delta ADC's

The most popular implementation method for Sigma-Delta ADC's is defining the loop-filter in z-domain and using switched capacitor integrators for the implementation. The model of a first order discrete time Sigma-Delta modulator is given in Figure 2.3.

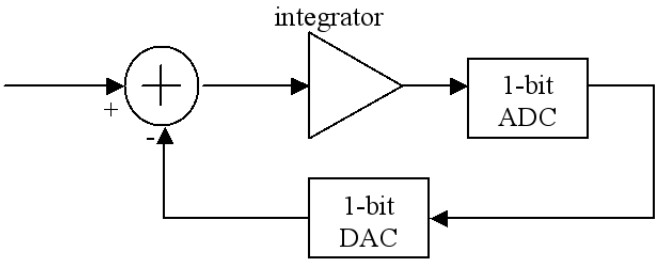

Figure 2.3. First order Sigma-Delta modulator

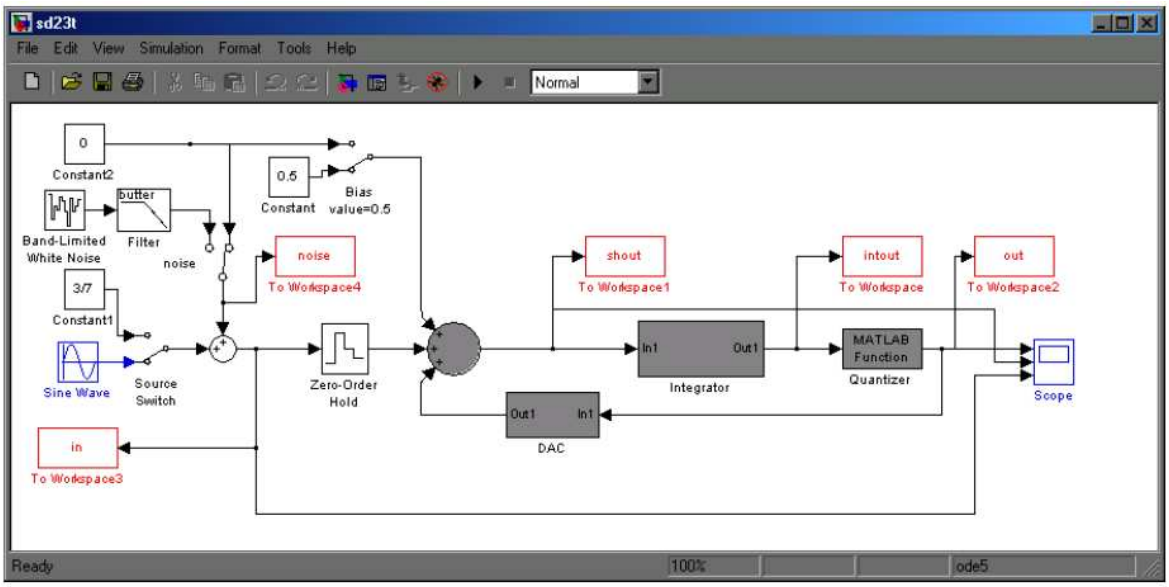

Figure 2.4. First order Sigma-Delta MATLAB output

A first order ideal Sigma-Delta converter was modeled by using Simulink and the output was processed in MATLAB. Hence, it is possible to observe the behaviour of the ADC. The model for the first order Sigma-Delta ADC is given in Figure 2.4. Figure 2.4 depicts the subblocks: integrator and DAC. Building the model using subblocks in Simulink eases the building of the multibit version.

Simulink allows the simulation of z-domain transfer functions with MATLAB functions; thus, the z-domain equivalents of the blocks may also be used. Time domain descriptions of some blocks shown in Figure 2.5. The utilization of both descriptions has been used in the literature [54] for modeling the system and both models can be used in Simulink.

The output of a Sigma-Delta ADC is similar to an oscillating output. The actual information lies in the average of the output signal. Hence, the rate of logical high or low values shows the value of output signal. This relation is depicted in Figure 2.6. In Figure 2.6, a sine wave with 0.45V amplitude is given to the first order Sigma-Delta ADC. The lower curve shows the Sigma-Delta output. The first two plots are the input and the integrator output.

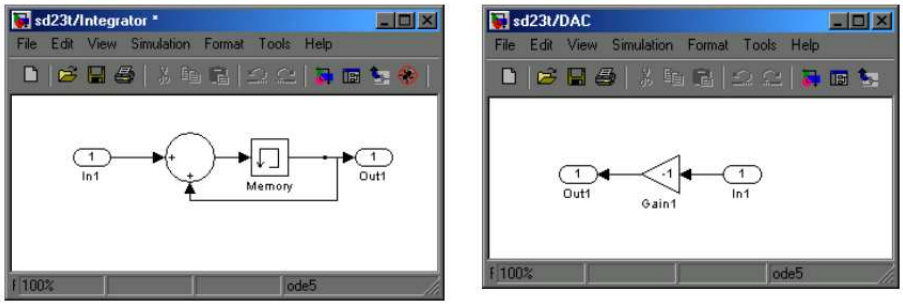

Figure 2.5. Integrator and 1 bit DAC model in time domain

The model given in Figure 2.3 can be defined as given in Figure 2.7. The transfer

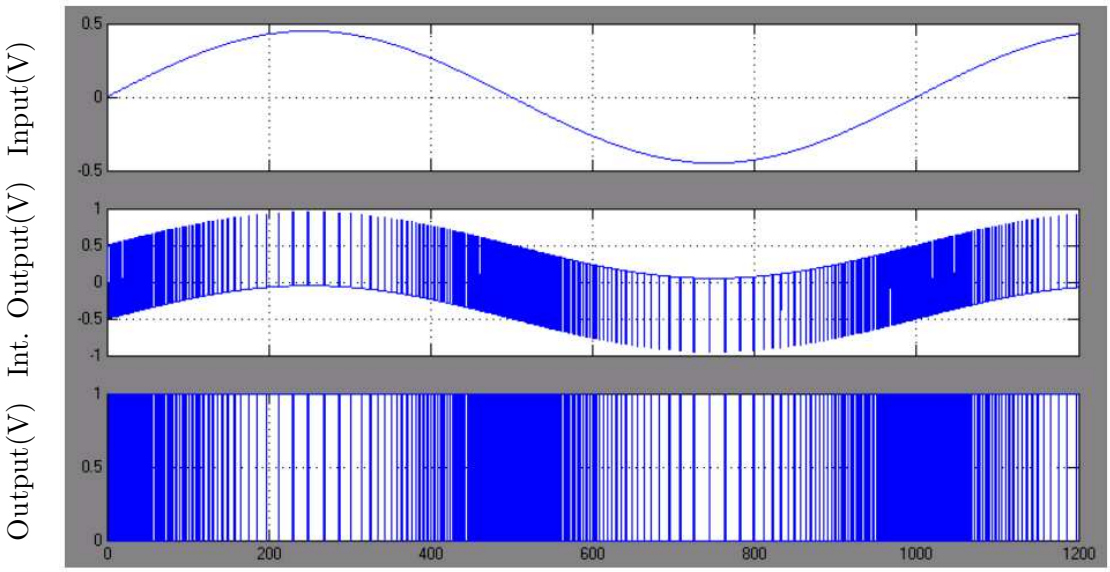

Figure 2.6. Output, Integrator output and input of 1 bit 1st order Sigma-Delta ADC

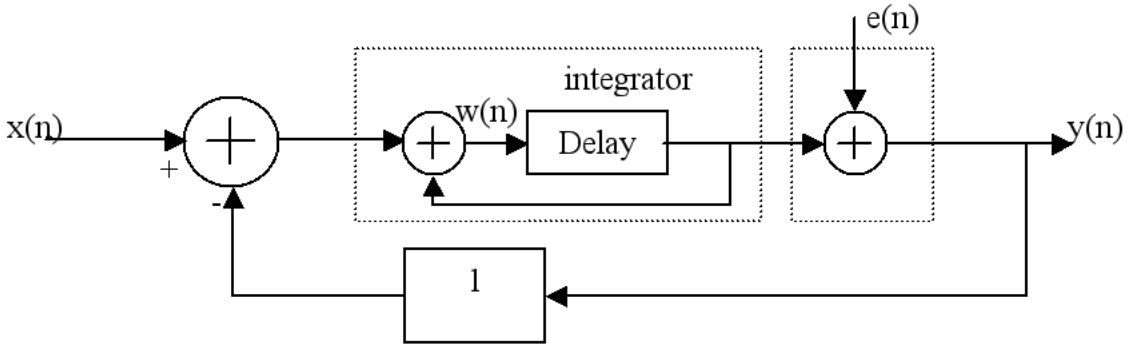

Figure 2.7. First order Sigma-Delta modulator

function of the ideal block model given in this figure can be expressed as:

$$\begin{aligned} w(kT) &= x(kT) - y(kT) + w(kT-T) \\ y(kT) &= w(kT-T) + e(kT) \\ y(kT) &= x(kT-T) + e(kT) - e(kT-T) \end{aligned} \tag{2.1}$$

Then we can write:

$$Y(z) = z^{-1}X(z) + (1-z^{-1})E(z) \tag{2.2}$$

which leads to a transfer function

$$H(z) = \frac{z^{-1}}{(1 - z^{-1})} \tag{2.3}$$

In (2.2) the term preceding $X(z)$ is defined as the signal transfer function (STF) and the term preceding $E(z)$ is defined as the noise transfer function (NTF). Hence, signal and quantization noise have always been modified by different transfer functions. This is one of the various advantages of Sigma-Delta ADC's, since the in-band quantization noise can be reduced due to this feature of the Sigma-Delta ADC. This reduction is also called noise shaping. It can be seen from (2.2) that the input can be observed at the output node with only a delay, whereas quantization noise is shaped by a high-pass filter. Hence, the STF and NTF can be designed for desired performance values.

The transfer function in (2.3) can be inserted as a block in the Simulink model. Although Figure 2.7 shows that there is a delay on the path from the input to the output, some researchers prefer to use a transfer function without a delay in the signal path and place the delay on the feedback path. However, this structure is not easy to implement in practice. Standard switched capacitor implementation already has a delay in the signal path. Hence, it is more convenient to use the model with delay.

One other advantage of Sigma-Delta ADC's is their modular structure. In order to have better noise shaping, the order of the NTF can be increased. The addition of a second integrator to the signal path is sufficient for this improvement. This structure called second order since the NTF is second order. The block diagram of the second order Sigma-Delta ADC is given in Figure 2.8. In this architecture, we have two integrators on the signal path. Also, feedback is modified. Although the DAC output is directly fed to the summation nodes in this research, the results of [7] show that lower gain in the feedback loop can improve performance. This also prevents the saturation of the integrator.

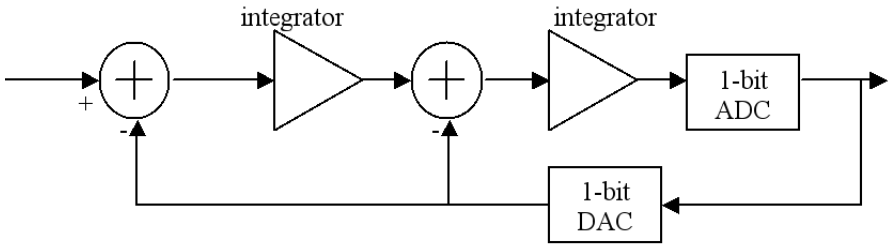

Figure 2.8. Second order Sigma-Delta modulator

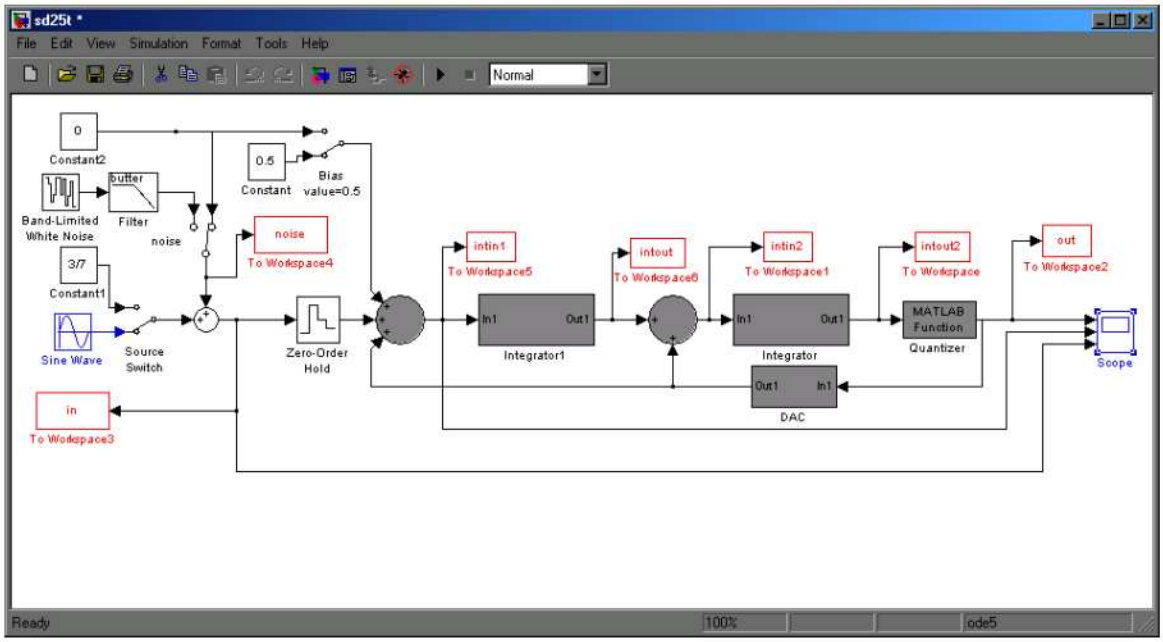

Figure 2.9. Second order Sigma-Delta modulator model in time domain

With the STF and NTF given below, there is only one delay between the input and the output. It is easier to show the transfer function with one delay unit in the signal path. However, in standard implementation, there is one delay for each integrator. Thus, (2.5) does not hold and it is modified to (2.6). The only difference is that the there are two unit delays between the input signal and the output signal. This does not affect the behaviour of the system significantly. The important point in all cases is the order of NTF.

The transfer function of a second order system can be expressed as:

$$\begin{aligned} z(kT) &= x(kT) - y(kT) + z(kT-T) \\ w(kT) &= z(kT) - y(kT) + w(kT-T) \\ y(kT) &= w(kT-T) + e(kT) \\ y(kT) &= x(kT-T) + e(kT) - 2e(kT-T) + e(kT-2T) \end{aligned} \tag{2.4}$$

Then we can write:

$$Y(z) = z^{-1}X(z) + E(z)(1-z^1)^2 \tag{2.5}$$

for one unit delay and

$$Y(z) = z^{-2}X(z) + E(z)(1-z^1)^2 \tag{2.6}$$

for two unit delays.

It should be stated that the coefficient of the DAC output that was fed to the input of the second integrator in (2.6) should be two. Figure 2.10 shows the output of a signal, which has amplitude of 0.2V. The lowest plot in the figure shows the bit stream at the output.

Discrete time Sigma-Delta ADC architectures can have even higher orders. With the selection of adequate coefficients, each order increase results in increasing the power of terms in STF and NTF. However, order values greater than two may cause stability problems and require simulations for stability check.

Another method for designing high performance Sigma-Delta ADC's is to use cascading. Thus, stability can be guaranteed because only first and second order structures are used. An example of cascaded Sigma-Delta ADC's is given in Figure

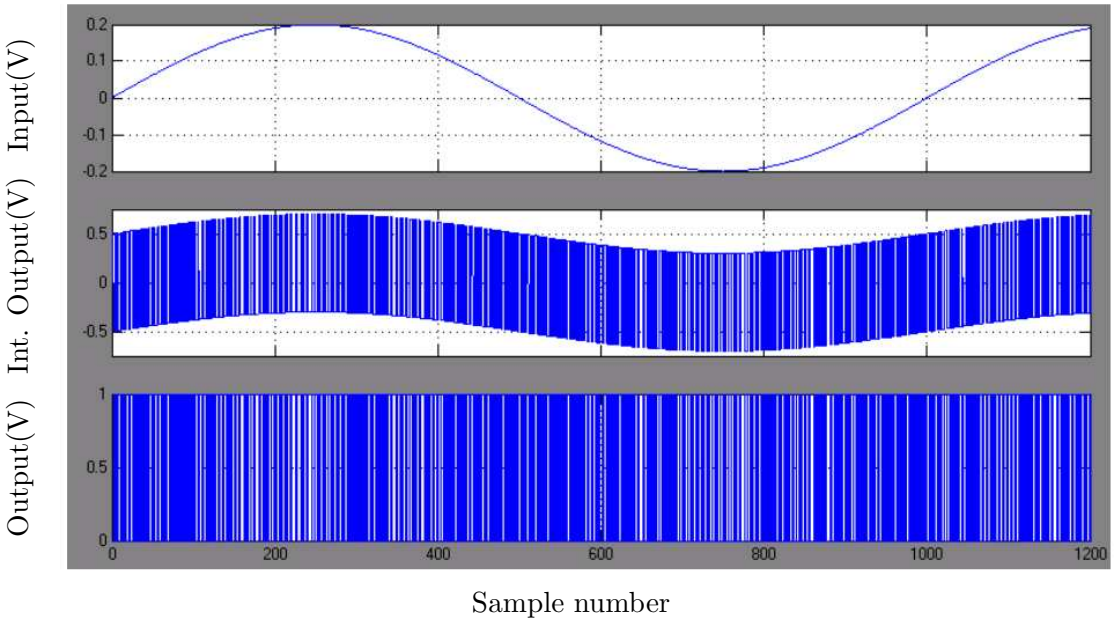

Figure 2.10. Output, integrator output and input of a second order Sigma-Delta ADC

2.11. Cascade architectures are more flexible but they require digital filters at the outputs to achieve the correct STF and NTF.

2.2. Continuous Time Sigma-Delta ADC's

One other popular method for Sigma-Delta ADC design is to use s-domain filters for implementation. Although the name implies a "continuous"operation, it is not fully continuous in reality. There is still a clock signal given to the ADC and DAC. The term continuous relates to the type of loop filter used in a Sigma-Delta ADC. Hence, the main difference from DT Sigma-Delta ADC's is the filter. Continuous-time (CT) filters introduce many advantageous features compared with ordinary DT filters. The most important advantage is that they may be designed for low-power, the main reason being the absence of switching activity involved in the operation. However, other problems arise in CT Sigma-Delta ADC's which are not so critical in their discrete time counterparts such as jitter.

Despite their disadvantages, CT ADC's have been gaining more and more interest in the last few years [55, 56, 57, 58, 59, 60, 61, 62]. Similar to discrete time Sigma-

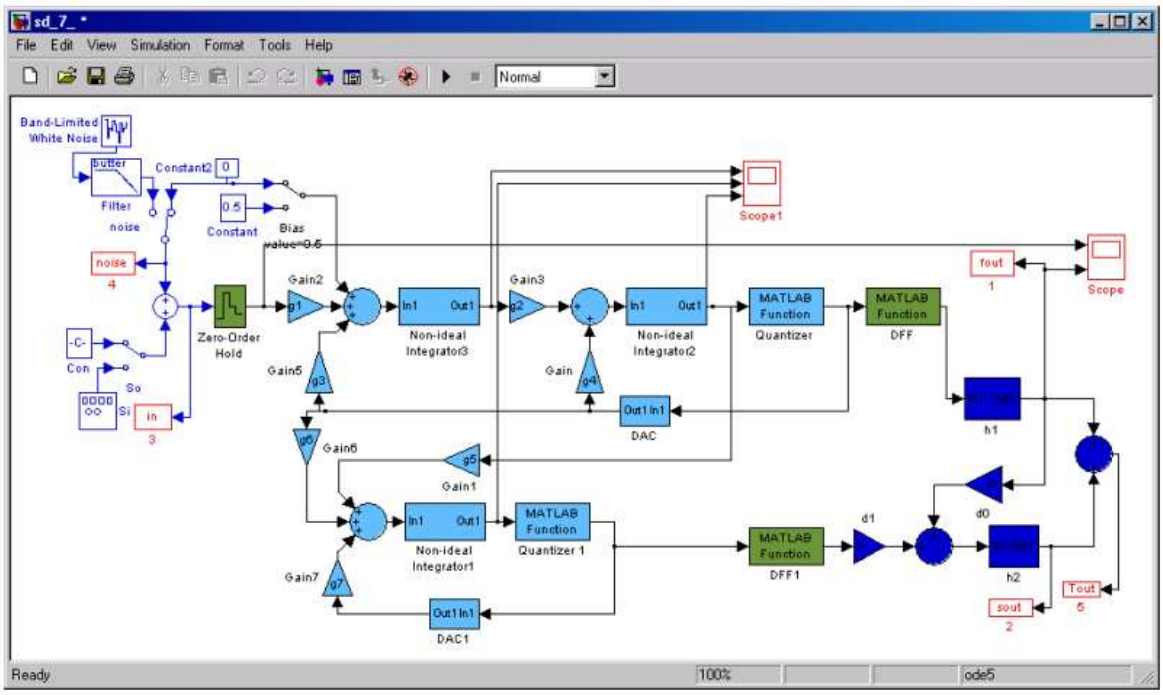

Figure 2.11. 2-1 cascade configuration

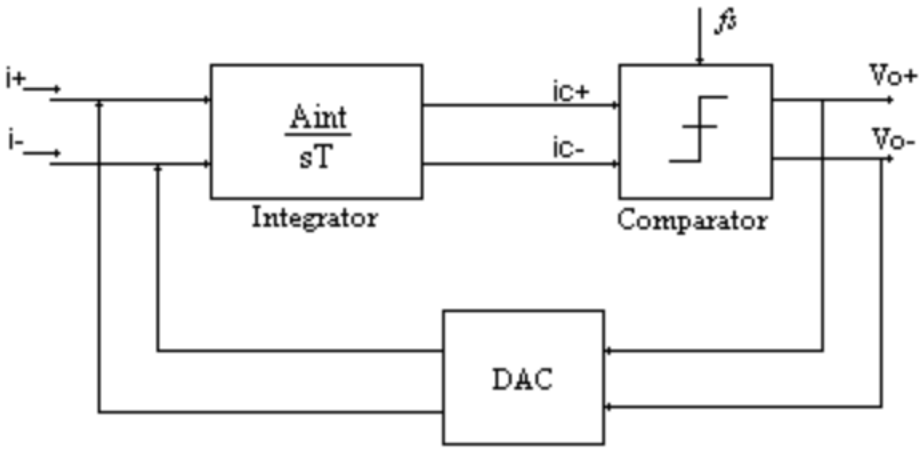

Figure 2.12. Continuous Time Sigma-Delta ADC Block Diagram

Delta ADC modeling efforts, continuous-time model development has also been a focus of research for the last few years. The main approach for modeling CT Sigma-Delta ADC's is the behavioral modeling approach [55, 13]. However, there is also reported work on analytical modeling approach in the literature [63, 64, 65].

CT Sigma-Delta ADC's can also be designed with different order values. Also cascading is possible but most of the time it is not desired. The main reason is the de-

viation in the coefficients used to realize cascading. The tolerances shold be extremely low in order to prevent the performance loss.

A simulation of CT Sigma-Delta ADC is given in Figure 2.13. The ADC is current input voltage output type. However, it can be observed from the figure that the output signal is very similar to DT Sigma-Delta ADC's.

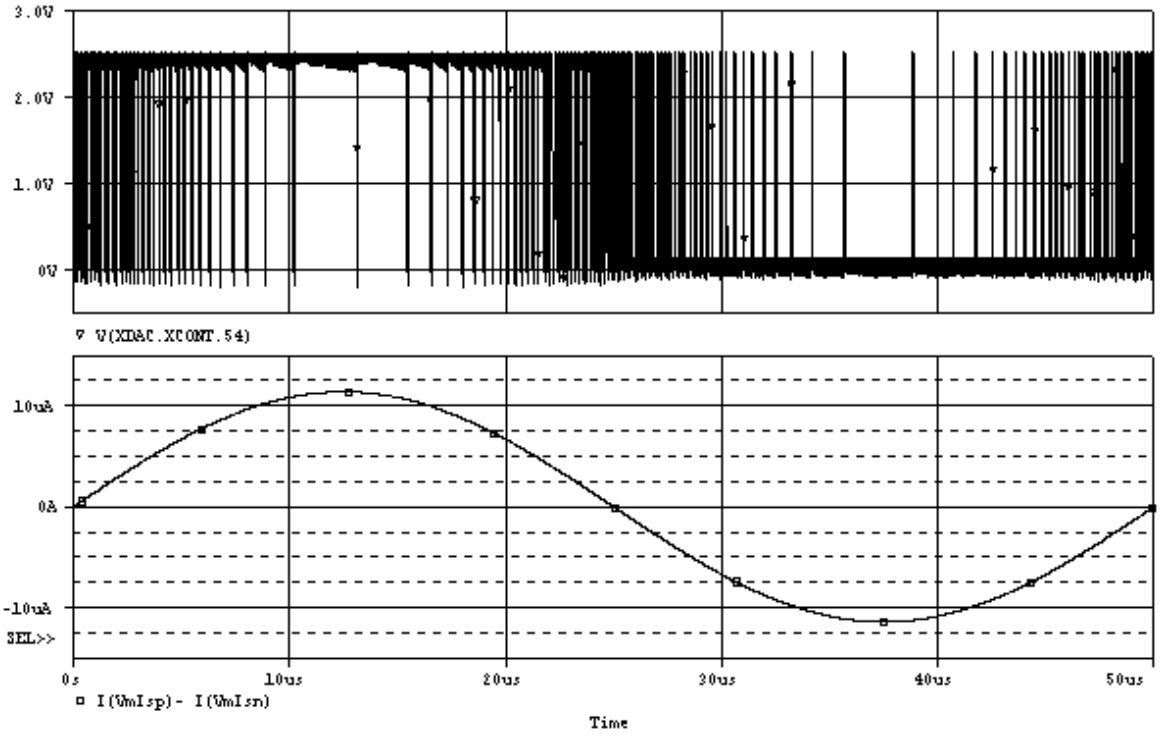

Figure 2.13. Circuit simulation of current-mode CT Sigma-Delta ADC

3. MODELING OF SIGMA-DELTA ADC'S

Development of design automation systems for Sigma-Delta ADC's requires accurate models of the architecture. Recent studies about modeling the Sigma-Delta architecture are given in the following paragraphs. The aim of this section is to define the types of errors in the blocks that a Sigma-Delta ADC contains and to present the effects of these errors via the developed models. Since Sigma-Delta ADC's have building blocks that operate in both digital and analog domains, it is difficult to model dynamic and static errors of the architecture accurately. In addition, the number of possible configurations makes the problem more challenging. Although the number of distinct blocks is limited, their interconnection configurations may affect the overall performance significantly.

In order to model Sigma-Delta ADC's, many different approaches have been tried. Some approaches concentrate on the speed of evaluation, in which case the accuracy of the models is degraded. More accurate results can be achieved by device level simulation tools [24], but they require huge amount of CPU time. Also, some researchers have tried to develop custom models [66], but they cannot achieve the required flexibility whereby the tools can only be used for one or two configurations [10].

One of the milestones in modeling efforts of Sigma-Delta ADC was a project (called AMPHIS) in the third framework of CORDIS named "ESPRIT III", which aims to develop necessary CAD algorithms for a complete ADC design automation. A group at the Instituto de Microelectrónica de Sevilla-Centro Nacional de Microelectrónica in Spain focused on Sigma-Delta ADC's as a contributor to this project. They published their research [7, 10], which successfully models the behavior of Sigma-Delta ADC for some configurations. They developed three tools: SDOPT for modulator sizing, FRIDGE for basic cell sizing, and ASIDES for behavioral simulation. Some other primitive design tools are also available [14, 67, 68] for limited configurations of Sigma-Delta ADC's. There is also another research group in Italy, which mostly focuses on

behavioral modeling of the Sigma-Delta architecture [54, 8, 69].

In the literature there are many other reserach activities reported previously. There were also some projects in EU framework that aimed to model the Sigma-Delta architecture, such as Anastasia+ [70], which is a name given to the project with the subject "Analogue enhancements for a system-to-silicon automated design"(Part of MEDEA+ [71]). The list of project partners contains university of Pavia and many corporations such as Siemens, Anacad and Analog Microelectronics. The group working on the Anastasia+ project [70] mainly focused on modeling the Sigma-Delta architecture. This project has ended in December 2004.

There are also some other groups working on modeling of Sigma-Delta ADC's [6, 11, 12, 16, 17]. However, they are mostly focused on developing models of a single architecture. The architecture selection procedure is left to the designer. Although there are methods for calculating the performance metrics reported in the literature, they are not used for architecture selection [11, 18, 72, 73, 74]. The designer is forced to perform several behavioral simulations by their simulation tools in order to select the architecture. However, even if the behavioral simulations may point to an optimum solution [13], this optimum point may change when the actual circuitry in the blocks is designed. A few behavioral simulations [8, 9] are not enough for such a design. The system level parameters should be bound to lower level parameters in order to achieve optimum design.

The following paragraphs basically summarize the modeling methods available in the literature. These efforts include device level models which can be simulated in simulators such as SPICE [75], or finite-difference equations for simulators such as SWITCAP [76, 77], or custom numerical models generally developed in C language [78]. However, these approaches have disadvantages. Some of them have speed advantage but lack accuracy, where some are more accurate, but the flexibility of the model to adapt to new architectures is poor.

SPICE is a common simulator mainly used for device level model simulations.

However, they are not suitable for Sigma-Delta ADC's. Its accuracy causes extremely long simulation times. The models may be simplified by using time-domain macro models [24]. In this case, the accuracy and the flexibility cannot meet the desired levels. Also, since the designer would perform more than one simulation in order to optimize the performance, the total simulation time dedicated to an overall architecture simulation would be still long.

One other approach is to use the z-domain descriptions [79] of Sigma-Delta architectures. There are simulators like SWITCAP that use this type of models. Some of them are developed in such a way that they can handle only one or two architectures, like MIDAS [14]. They have a speed advantage but flexibility and accuracy are lacking most of the time.

There are also some other approaches that are table-lookup based. However, this approach is not suitable for modeling various Sigma-Delta architectures. Especially for large tables the speed degradation is huge.

There are some approaches which lie between the ones given above. However, the results are not promising for such efforts when the architecture selection procedure is taken into account.

Two studies [6, 69], are related with this thesis regarding the modeling efforts. Both of these are focused on the modeling effort of the Sigma-Delta ADC's. The first one presents a tool called DAISY (Delta-Sigma Analysis and Synthesis). The tool developed contains an inner block which performs a behavioral simulation and tries to optimize the parameters regarding the performance evaluation of the behavioral simulator. The advantage of this approach is the easy implementation of the models. The calculations for slew-rate and jitter errors for each sample is much more easy and these models are commonly used in behavioral simulators.

The drawback of this approach is that if the initial assignment in the solution space is far from the solution, then the number of iterations, which should be performed

by the behavioral simulator, may be high. In [6], the approach is compared with device level simulations are reported as advantages to the available tools. However, our approach is more advantagous than the reported work in the literature since our approach does not contain any behavioral simulation cycle. Although our MATLAB models are very similar with the models in this paper and they are capable of performing any performance evaluation.

The work reported in [69] utilizes a Simulink model in order to estimate the performance of second order systems. This paper uses a MATLAB model to perform behavioral simulations. The advantages of our approach to the one given in [6] are again valid for this work. The study in [69] aims to show the behavior of some of the non-idealities available in high order systems and it does not claim to be a synthesis environment for Sigma-Delta ADC's.

3.1. Error Sources

The error sources in the system were modeled in order to estimate the SNR performance of the ADC. The SNR value for the Sigma-Delta ADC represents the effective number of bits of the ADC. Then, the resolution of the system can be calculated. The error sources of the architecture may be collected under four different headings. Three of the error sources are related to the blocks of the ADC: Integrator, quantizer and DAC. The remaining one is the clock jitter. All the error sources can be modeled either as noise sources or transfer functions [7, 3]. As a result, from the transfer function, the noise shaping can be obtained. From the noise sources, the in-band noise can be obtained [80]. Then, SNR and other performance parameters can be calculated. In order to achieve faster models, calculations performed for all error sources must be carried out without behavioral simulations.

3.1.1. Integrator

The integrator designs used in Sigma-Delta ADC's are commonly switched capacitor integrators. Most of the errors inherited in these integrators can be modeled in

the transfer function. The ordinary integrator has the following transfer function [3].

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{3.1}$$

The transfer function of a non-ideal integrator can be expressed as:

$$H'(z) = \frac{B \times z^{-1}}{1 - Cz^{-1}} \tag{3.2}$$

where B represents the gain error and C represents the leakage at the simplest level. However, the expressions of B and C are much more complex and composed of time constants, OPAMP gain, switch resistances, etc. A complete model can be given as [81]

$$H''(z) = \frac{C_s}{C_f} \frac{(1 - \delta_1) \times (1 - \delta_2) \times z^{-1}}{(1 - (1 - \delta_3)z^{-1})(1 - \delta_1\delta_2)z^{-1}} \tag{3.3}$$

In this equation δ_1, is charging error, δ_2, is charge transfer error and δ_3, is the error coming from the finite gain of the amplifier. These parameters are as follows:

$$\delta_1 = e^{-\frac{t1}{2R_sC_s}} \tag{3.4}$$

$$\delta_2 = e^{\frac{A_v t2}{C_s + 2A_v R_s C_s}} \tag{3.5}$$

$$\delta_3 = \frac{C_s(1 - \delta_2)}{r_o C_f A_v + C_s + C_f} \tag{3.6}$$

where R_S are switch on resistances, r_o is the OPAMP output resistance, t_1 and t_2 are charging and transfer times.

Although the transfer function of the integrator involves non-idealities, there are more error sources, which should be added separately. These are among the major differences and contributions of our work from other works reported.

Slew-rate of the OPAMP causes an error which is additive to the output at signal band. Some previous researchers try to estimate slew-rate by analytically calculating

[7] or by performing behavioral simulations [54]. For fast design automation systems, slew-rate can be calculated in a different way. The details for slew-rate model is given in following sections.

There are also errors caused by parasitic capacitances (C_p), which can be added to the capacitor C_S in transfer function. If these parasitic capacitances are known or can be estimated during the design, these capacitor values can be used to calculate C_S.

Capacitance mismatch is also important. The mismatch modifies the gain in transfer function, such as (1-σ) where, σ represents the mismatch value.

The input referred noise coming from switches and OPAMP can be taken as an additive noise source at the input. The effect of this error results in an increase in the noise floor. The offset of the OPAMP is an additive noise source in the feedback loop. This noise source can be added to the quantizer error as explained in the next section.

3.1.2. Quantizer

The error sources of the quantizer are the offset error, and the voltage error caused by settling time or slew rate. It can be easily seen that the offset error (m) results in an error as expressed below, where Δ represents the voltage steps in quantization.

$$e_q^2 = (\frac{\Delta}{12} + m^2) \tag{3.7}$$

The other errors can be taken as additional noise to the signal band. However, the error given in the above expression is shaped by the noise transfer function. The in-band noise power is given as P_Q, where OSR is the oversampling ratio.

$$P_Q = (\frac{\Delta}{12} + m^2)\frac{\pi^2}{3.OSR^3} \tag{3.8}$$

The noise source in the quantizer is in the feedback loop thus, the noise contribution is shaped by the noise transfer function. As it can be seen from the equation, the in

band noise can be reduced by increasing the OSR.

3.1.3. Digital-to-Analog Converter

For single bit architectures the errors of the digital-to-analog converter (DAC) do not have significant impact on the whole system. However, for multibit systems the linearity of the DAC is crucial and actually the limiting factor in the performance since it is directly added to the input. Especially for ADC's designed by using switched capacitor integrators, the only noise contribution of the DAC is the noise of the switches which is connected to the integrator. This error can be modeled as additive noise to the input signal, which will raise the noise floor of the system.

The equation database in the developed tool has the DAC non-linearity model as given in [82]:

$$\begin{aligned} Y(z) = z^{-1}X(z) + d_{2N-3}(1-z^{-1})^{L}E_N(z) \\ +d_{2N-3}(1-z^{-1})^{L-L_N}E_D(z) \end{aligned} \tag{3.9}$$

In (3.9) E_D is the non-linearity error of the last stage and it is clear that it will be shaped by order $L-L_N$. The inband error power for a two stage cascaded architecture can be expressed as:

$$P_Q \approx d_1^2\left(\sigma_Q^2\frac{\pi^8}{9OSR^9} + \sigma_D^2\frac{\pi^4}{5OSR^5}\right) \tag{3.10}$$

3.1.4. Slew-Rate

The developed slew-rate model is one of the novelties of this thesis [27]. The research in the literature shows that slew-rate errors have only recently been taken into account for the Sigma-Delta design tools. Also, these errors are not modeled but rather inherited from the behavior simulators for functional verification. That is, the designer does not care about slew-rate while designing the Sigma-Delta ADC, but then

performs a behavioral simulation to check if the design satisfies the given slew-rate limits. This approach is inefficient since the designer may overestimate the slew-rate error, which may result in a larger silicon area or higher power consumption. Also, the designer may underestimate the slew-rate error, which in this case results in functional error and the ADC should be redesigned. The model developed should estimate the slew-rate error from the input signal so that the designer may know the magnitude of the error with a given input characteristic. This approach had been used to derive the slew-rate model.

Previous research about slew-rate [8, 83] has mainly focused on calculating the distortion caused by the slew-rate. However, calculating the distortion is not sufficient. The effects should be estimated for an ADC with a given input signal. Thus, more efficient designs can be possible.

Slew-rate is effective when the amplifier in the integrator cannot supply sufficient current to the output. As a result, for a switched capacitor integrator, a wrong value of charge can be transferred to the feedback capacitance.

The voltage at the input node of the modulator and at the input of the integrator may not have similar amplitutes. Slew-rate is related with the voltage at the input of the integrator and small changes in the input signal do not mean small changes at the input of the integrator.

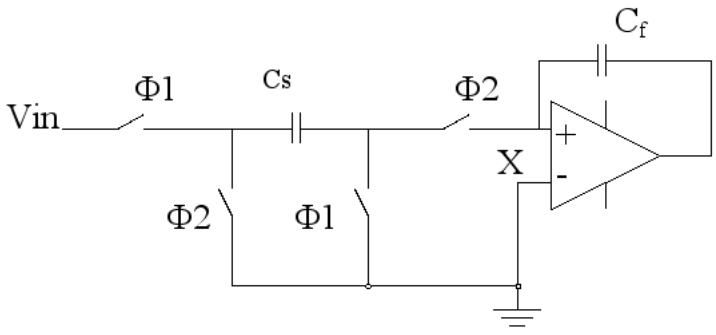

Figure 3.1. First order switched capacitor integrator

Figure 3.1 shows the diagram of the switched capacitor integrator. The symbols ϕ_i, show two non-overlapping clock signals. The charge is collected on the feedback

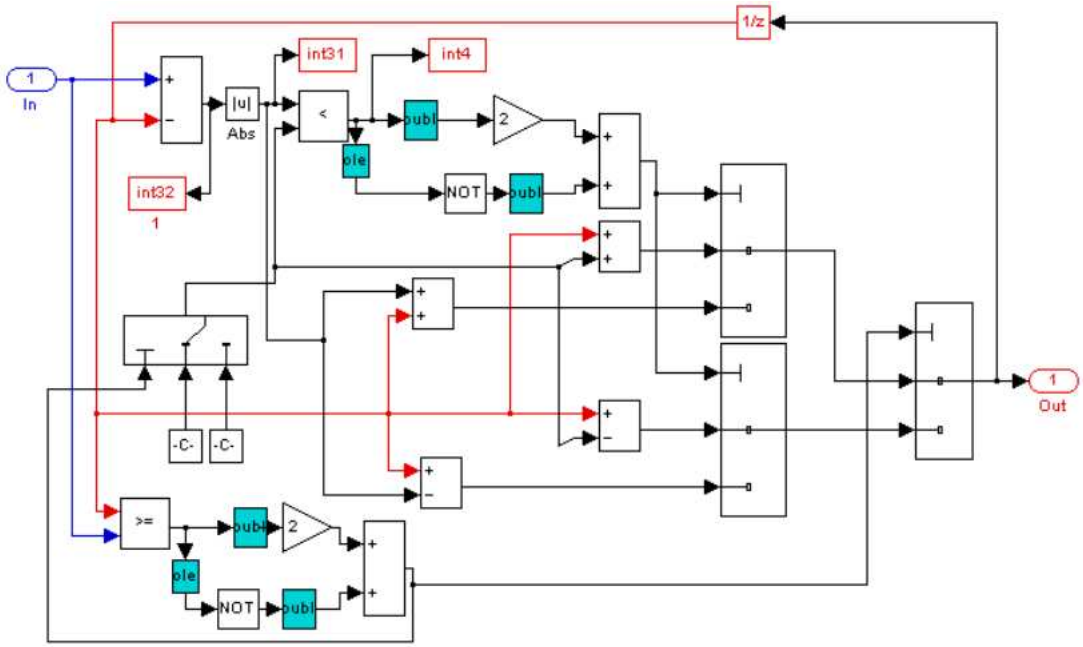

Figure 3.2. Slew-Rate model in MATLAB

capacitor C_f. The amplifier should be "strong" enough to transfer the charge to this capacitor. The input of the amplifier, node marked with X, is an important node. The values at this node determine whether the ADC enters a non-slewing condition or not. In order to analyze the behavior of the system a MATLAB-Simulink model has been developed. Also, the model of the non ideal integrator developed earlier has been integrated into Simulink which represents the non-idealities such as capacitor mismatch, finite amplifier gain, gain error and etc. The subblock that represents the slew rate of this integrator is given in Figure 3.2. The block in Figure 3.2 uses a linear model for slew-rate. The block can be modified and the linear model can easily be replaced by its more complex model such as tanh(x).

The developed MATLAB model was used to analyze the behavior of the SD ADC with different input signals that differ by their histogram and frequency spectrum. The results showed that if the value of the input signal is nearly half of the maximum input voltage, possibility of non-slewing is maximum. In other words, if the number of non-slewing conditions has been observed for various different DC voltages, it can be seen that the values close to the mid level of the input range results in higher number of non-slewing conditions. Table 3.1 summarizes some of these results for an input range

from -1 V to 1 V.

Another important observation from the analysis is the dependence on the frequency of the signal. Although the slew-rate definition seems to be directly related to the frequency, this not the case in the analysis. The frequency of the signal does not have significant effect in the total error produced by the slew-rate. Different input signals have been given as an input to the model. Results showing the effect of the frequency are given in Table 3.2.

Table 3.1. Input level effect on slewing conditions

Input level	Percent of non-slewing conditions
0.9V	5.0%
0.1V	89.9%
-0.2V	79.8%
-0.9V	5%

Table 3.2. MSE for two different frequencies

Frequency	Mean square error
20Hz	2.773e-006%
10Hz	2.698e-006%

Figure 3.3 shows the effect of the frequency. Even though the frequency has been doubled the MSE has only changed less than 3%. These results are achieved when the slewing condition is symmetric for rise and fall of the signal. The results show that the frequency of the signal does not effect the slew-rate distortion significantly. Figure 3.4 shows the effect of slew-rate with different non-slewing conditions. The increased error at high frequencies is due to low sampling since signal frequencies are high.

Figure 3.4 shows the MSE of the output for one totally slewing and four non-slewing systems with different slew-rate ratios. The horizontal axis shows the number of samples taken in one period. The vertical axis represents the calculated MSE value. The number of non-slewing conditions is very important. The distortion at the output of an ADC can be represented by MSE of the error, where error is the difference

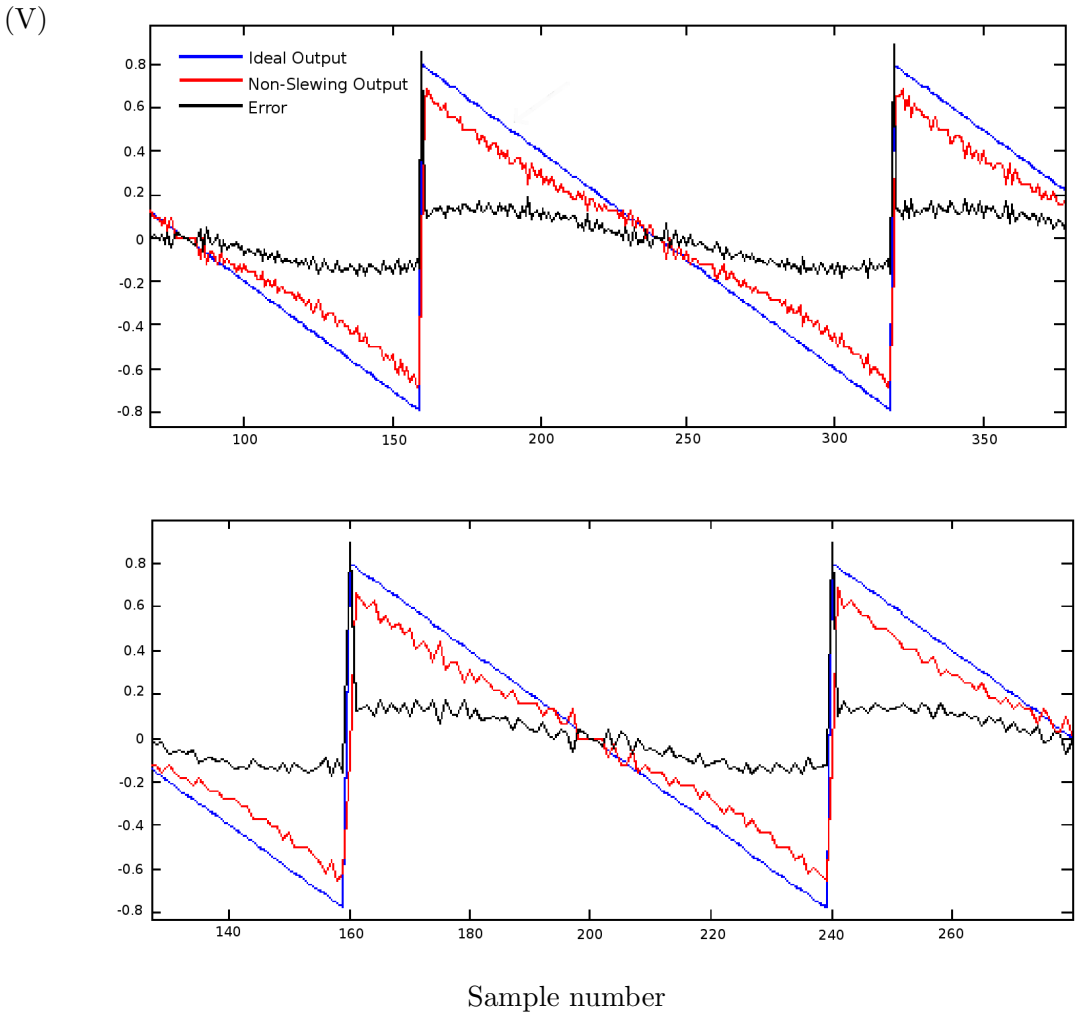

Figure 3.3. Ramp input for two different frequencies showing the ideal input, ADC output with non-slewing condition and the error between

between the ideal signal and the converted signal as illustrated in Figure 3.3. On the other hand it is easier to estimate the slewing condition from the model. The square of the number of non-slewing conditions is actually linearly proportional to MSE, which can be observed from Figure 3.5. The horizontal axis shows the square of the ratio that is the number of non-slewing condition occurred over the total number of samples. Three different curves represent different slew-rate conditions.

Previous paragraphs show the simulation-based information. This information can be obtained from analytical expressions. For a system where the reference levels are -1V and 1V, the DAC output can be expressed by a signum function. Thus, by

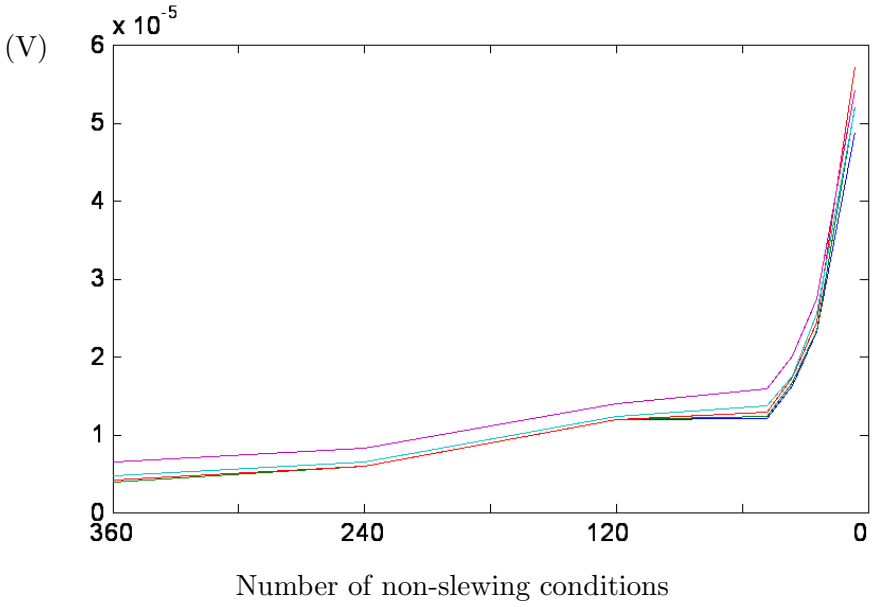

Figure 3.4. MSE of output for different non-slewing conditions for increasing frequency.

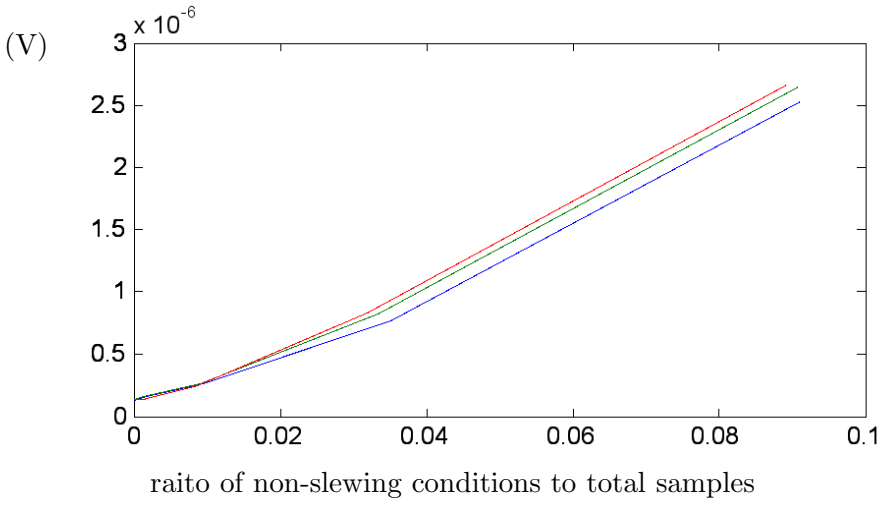

Figure 3.5. MSE vs. square of the ratio of slewing conditions to the total sample number.

using the discrete time definitions, an expression for the node X in Figure 3.1 can be found. However, this expression should be in terms of input such that it allows us to estimate the non-slewing condition from the input. If the histogram of the input signal is known, the total number of slewing occurrences can be estimated which can lead to the estimation of total error that will add up at the output. If the input is x[n], output y[n], input of the quantizer w[n] for a first order Sigma-Delta ADC, the

following equations can be written:

$$y[n] = sgn(w[n])w[n] = w[n-1] + x[n-1] - y[n] \tag{3.11}$$

which leads to:

$$w[n] - w[n-1] = x[n-1] - sgn(w[n]) \tag{3.12}$$

The left side of the first expression shows the difference of the current output of the integrator with the previous output. If the difference is larger than the corners defined in linear slew-rate characteristic, non-slewing condition occurs. This equation also represents the reason why mid level inputs may cause more slewing conditions. The output of the DAC, which has been represented by the signum function, will have nearly equal number of -1 and +1 for medium level signals and is similar to a square wave oscillator. Thus, smaller x[n] values will bring large number of slewing conditions in a system which has lower slew-rate limits. However, if input has values closer to the limits the DAC output will be either -1 or +1. Since the negative feedback tries to compensate, the input and the DAC output will have opposite signs most of the time. Thus, the difference between them will generally bring smaller differences. However, the difference may be large for a small amount of time. The simulations have shown that it was 5% of the total samples as it was stated in Table 3.1. Also, slew-rate requirement becomes more stringent. This is the case when both input and the DAC output have the same sign.

The information presented in this thesis shows that for a system which has a signal histogram having peak values at center and which has a lower slew-rate level will have large number of non-slewing conditions. This type of input signals requires careful amplifier design. The slew-rate capacity should large enough. Also capacitor values should be selected as small as possible to relax the slew-rate requirements. On the other hand, if the signal histogram has values spread all over the input range or having its peak values near the limits, the system may tolerate smaller slew-rate values. Although, this approach will bring error to the system it can be tolerated for mid level

resolutions. Depending on the application, this approach may relax the tight limits on the amplifier specifications.

Another important result of this analysis is that, the sinusoidal inputs may not be the correct input for testing the slew-rate effect on SD ADC. Since the histogram of the sinusoidal input is mainly collected at the limits of the input spectrum, possibility of non-slewing conditions is very small. More suitable input will be a signal with Gaussian like histogram.

As a summary, the number of non-slewing conditions were defined in terms of input signal in order to estimate the number of non-slewing conditions. This information is then converted to the MSE value with the conversion presented in Figure 3.5. Although the expression given earlier summarizes the relation, some numerical results will clarify the situation.

For an input signal limited by -1V to +1V, the maximum difference at the node X is basically the maximum of the input signal ± 1; that is, if the maximum value of the input signal is 0.4V, the maximum difference will be -1.4V and +1.4V. For an example design, which allows up to 1.12V per each transition, following values can be obtained:

$$\begin{aligned} 1.12V = x_{lim} \pm 1 = 2.12V or 0.12V \\ -1.12V = x_{lim} \pm 1 = -2.12V or -0.12V \end{aligned} \tag{3.13}$$

These limits determine where the slewing starts. For input signal values exceeding 0.12V and -0.12V, the slewing starts, which gives the coarse estimate of the number of slewing conditions. For input samples with values greater than ± 0.12V such as 0.3V, there are two possibilities according to the digital output. One is 1.3V and the other is -0.7V. Only 1.3V will introduce a slewing condition. For values such as 0.1V there will be no slewing. If the output node has equal possibility of taking -1 or +1 values, which is a coarse estimation, the number of slewing conditions will be equal to the half of the number of samples having values greater than ± 0.12V. For our example, the total number of samples is 8192, and number of samples with greater values than

±0.12V is 2604. If the possibility of the output to take the value, which causes the slewing condition, is 50%, the total number of slewing conditions should be 2794. The MATLAB simulations showed that the real value for slewing conditions is 2301. The error in this approach is nearly 20%. Although this value may give an opinion on the order of magnitude of the problematic situations, the number should more accurate. The reason of this problem is basically taking the possibility of the digital output as 50%. If the percentage is 50%, this means that the average of the input is 0, since the digital output is actually shows the average of the input signal. However, this cannot be true since values closer to the 0 do not introduce any slewing condition is their value is smaller than ±0.12V. This result shows that the output should have different possibility values for different input signals. In the given example, it can be observed that the possibility of values causing slewing conditions is less that 50%, actually 40%. Since the limit of the effective range has been defined and the average value of the input signal can be calculated in order to get the ratio of ±1 at the output, an accurate estimation can be calculated. For the example presented in this section the input signal is a sawtooth signal with ±0.4V of maximum values. The possibility of taking the value that cause slewing with in the input range affecting the slewing is 39%. This is very close to the real value, 40%.

This example shows that this approach is suitable for estimations of slewing conditions, which may be used by designers. Since this approach is only related with the input histogram of the signal, the calculation time is quite small.

3.1.5. Jitter

Another problem for a Sigma-Delta ADC is the error caused by the jitter. Since the architecture of the Sigma-Delta modulator uses oversampling, the error that may come from jitter at high sampling rates cannot be ignored. Sigma-Delta ADC's generally have high sampling ratios, which may require jitter smaller than a few picoseconds for wideband applications. Thus, the effect of jitter should be modeled accurately. Another contribution of this thesis is the model developed for jitter error [28]. There are some behavioral simulators that use jitter in their model [8]. However, these sim-

ulators only calculate an error for each sample depending on the current value of the signal and a random function which modifies the sample time. This approach can be acceptable for a simulator. However, for a design automation tool, the error caused by the jitter should be estimated with given information. This information should be the input signal characteristics and the jitter characteristics. Since behavior of the jitter may vary for different circuit topologies, the characteristics for the jitter should be available. For some systems, input may couple to the clock signal, which causes it to be modified by the input signal. There are some other systems, which have clock variation independent of the input signal but possess variations which have Gaussian distributions. In order to develop a general tool, the characteristics of the jitter should be known.

The effect of jitter can be modeled by an error added to the signal depending on the signal and clock. The most convenient way of defining this error is to consider the first derivative of the signal and the possible shift in the sample time. The result will add an error to the signal. This approach is similar to the work presented in [8]. However, the model presented in [8] assumes a sinusoidal type input and the calculations for each sample may vary with other types of input. Also, the jitter activation function was taken as white, which does not represent the reality most of the time.

In our model, the signal at the input of the first integrator with clock jitter error is defined as:

$$Xnew[nT] = X[nT + d] = X[nT] + X'[nT].p(t) \tag{3.14}$$

where, X[nT] is the input signal, X'[nT] is the first derivative of the signal and p(t) is the Gaussian probability function representing the deviance from the sample time. Figure 3.6 shows the jitter error. The linear approach is feasible since Sigma-Delta ADC employs high oversampling ratios values where the difference between two input samples does not change much most of the time.

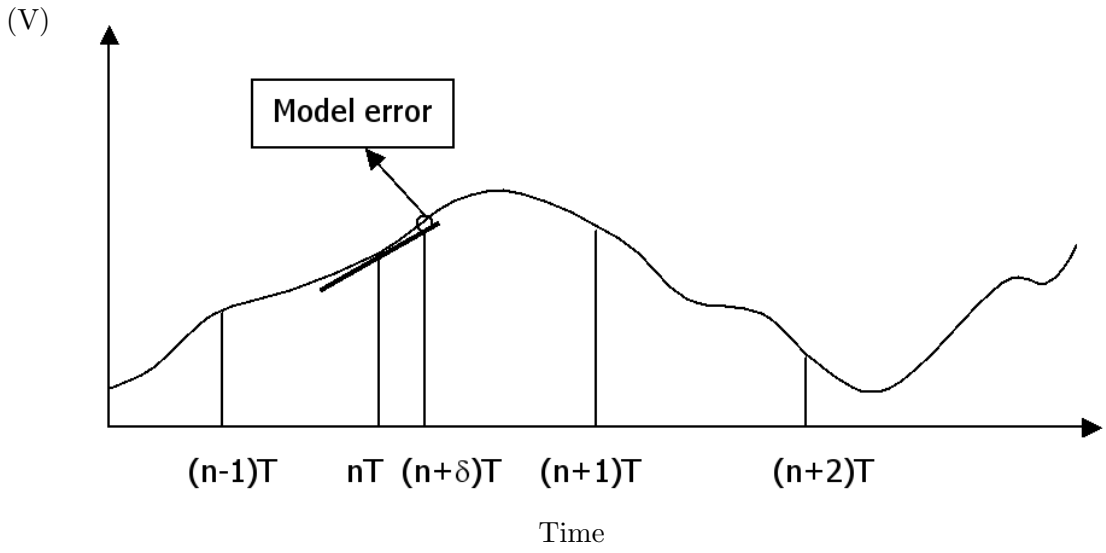

Figure 3.6. Linear approach to jitter error

The deviance from original sampling time is shown with (n+δ)T in Figure 3.6. The error of the linear model is also shown in the figure. Otherwise, the first derivative would not give accurate results. For higher accuracy it is possible to include second derivative at the expense of evaluation time. In our simulations, it was observed that the linear model gives satisfactory results for applications where using Sigma-Delta ADC's is advantageous.

The jitter error was modeled as an additive input noise where $X'[nT] \times p(t)$ term is an addition to the signal. The derivative can be expressed as $X[nt] - X[nT - 1]$. Since the input signal characteristics is available, this difference can be calculated. The $p(t)$ term is just a Gaussian function and its characteristics must be given for reasons explained above.

This model was implemented in MATLAB and model tests were performed. The signal used as an input is given in Figure 3.7. The signal is sum of three sinusoidal signals. The signal is sampled with a clock containing jitter. The histogram of the jitter in the sample clock is shown in Figure 3.8. The distribution of the jitter is a normal distribution, which is true for most Sigma-Delta applications [6]. MATLAB already samples signals even though a continuous signal is chosen as input. This sampling process prevents the modification of sampling in order to realize the jitter. Therefore,

the sampling rate was increased by 50 times and new samples were taken for every 50 samples. The x-axis shows the deviance of the sample time where the difference between two samples is 50.

The results for the input signal given in Figure 3.7 are very promising. The error (MSE) between system using sampling modified with Gaussian distribution and the system using our model is 0.5%. The plot of the error is given in Figure 3.9.

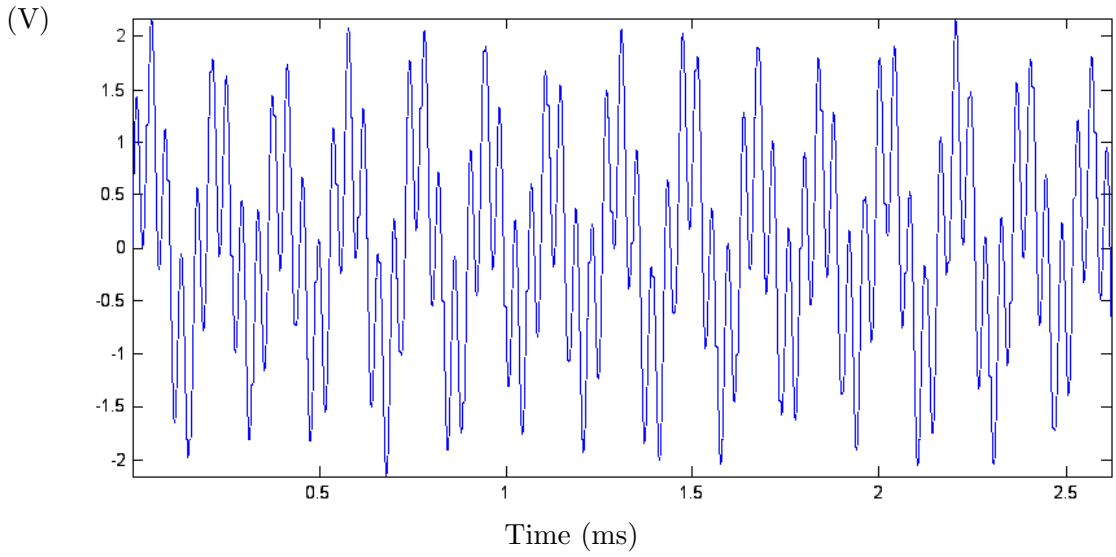

Figure 3.7. Input signal

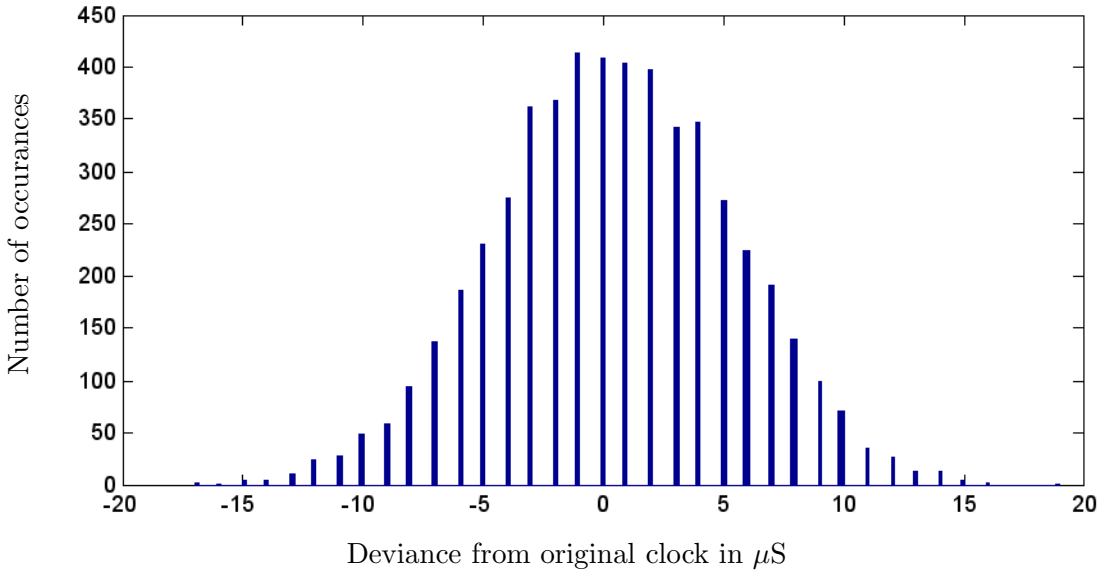

Figure 3.8. Histogram of the clock jitter

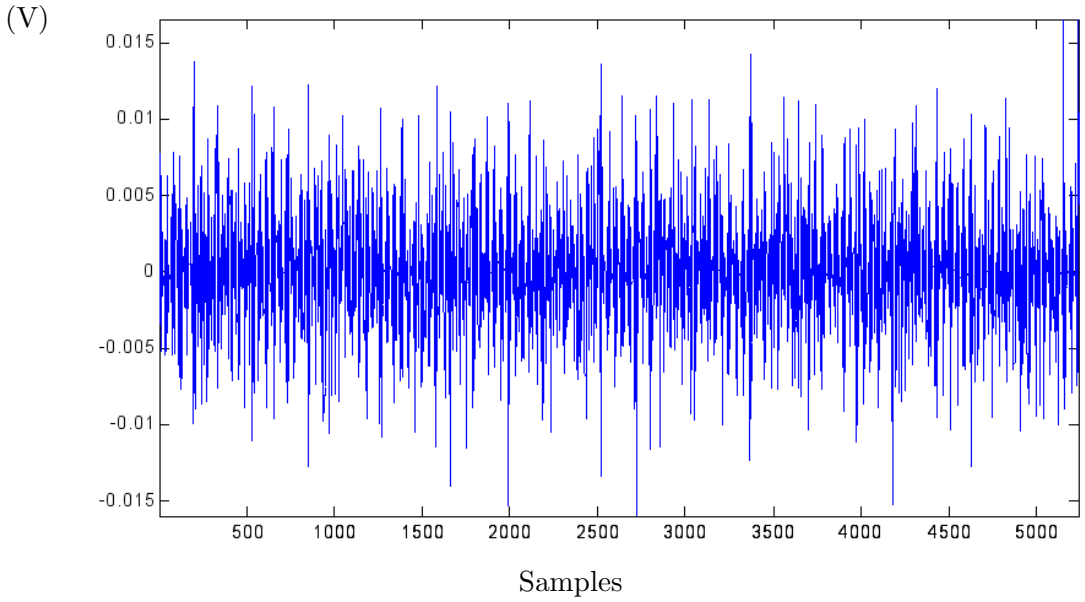

Figure 3.9. Jitter error at the input

3.1.6. Matlab models

MATLAB-Simulink simulations were performed in order to see the effects of the noise sources explained earlier. The input was taken as 0.5V peak-to-peak sine wave in Simulink model. The simulation results show the characteristics of the system for noise sources. The noise sources of our approach correspond to input referred noise, which is modeled as an additive noise source at the input. Since this noise is added to the input signal, the noise is not shaped with the noise transfer function, NTF. The input referred noise shaped by the signal transfer function, which is z^{-1}, or a unit delay, directly added to the output node. This result can be seen from Figure 3.10 and Figure 3.11. The noise at the input increases the noise floor. In Figure 3.12, the noise shaping also can be seen at high frequencies.

3.2. Multi Stage Architectures

Multi stage architectures are commonly preferred for performance improvement on NTF. Each order shapes the noise better by modifying the NTF such that the slope of NTF increases 20dB/decade. NTF for first and second order systems is given in Figure 3.13.

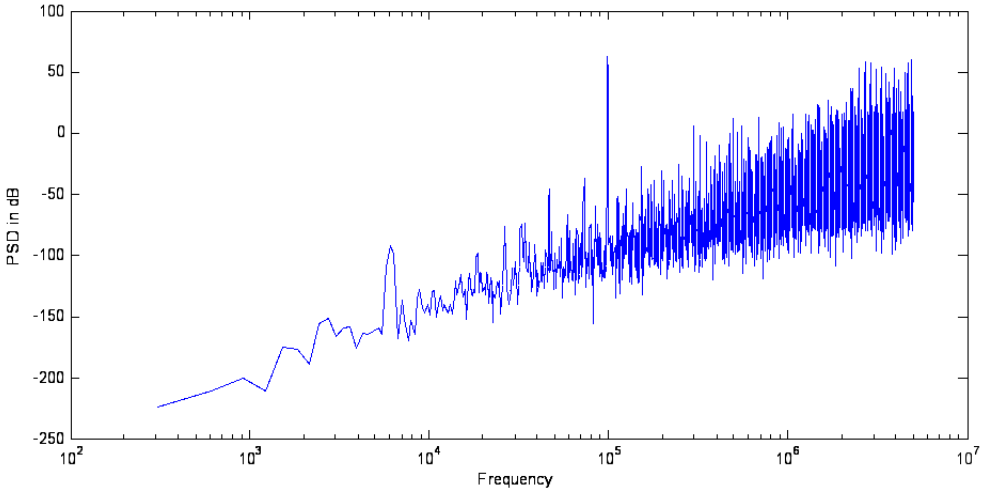

Figure 3.10. PSD of the output of noiseless Sigma-Delta modulator

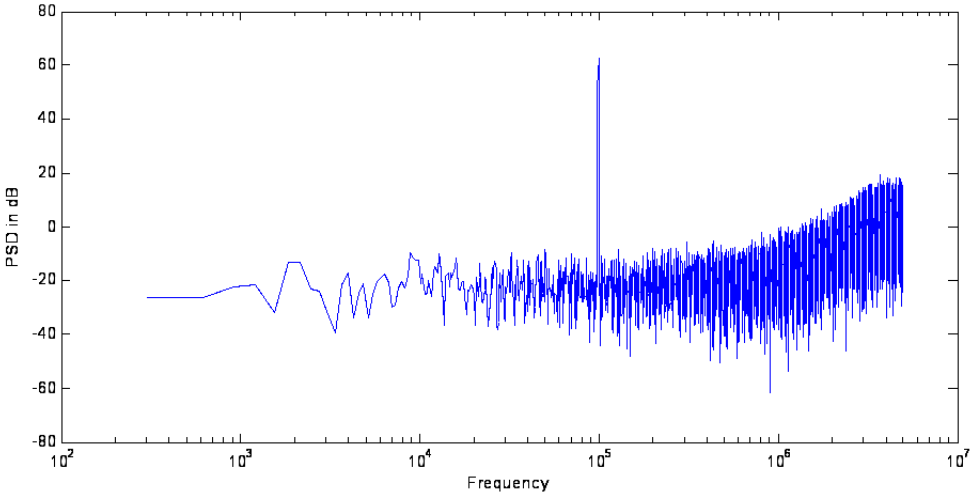

Figure 3.11. PSD of the output of noisy Sigma-Delta modulator

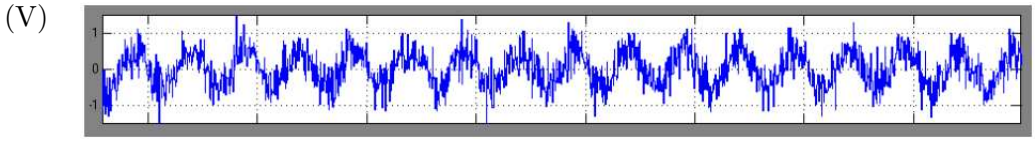

Figure 3.12. Noisy input given to the system

The major error sources in a typical higher order Sigma-Delta architecture regarding all cascaded or higher loop structures, are the ones which are introduced by the first loop in the architecture. The error sources can be classified into two groups in general. These can be defined as the error sources which are modified by STF and the ones which are modified by NTF. Since NTF shapes the error sources and moves them

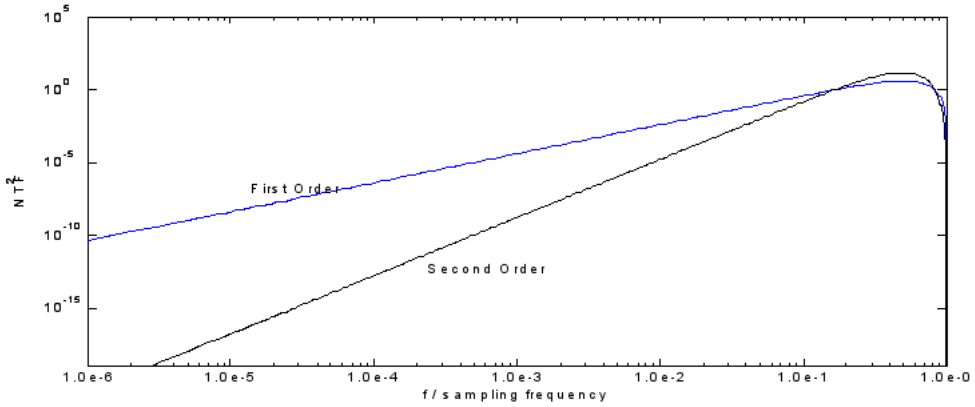

Figure 3.13. NTF of first and second order system

out of band, some error sources do not degrade the system performance compared with the others. The error sources, such as input referred noise sources, which are modified by STF directly increase the noise floor. However, in multistage architectures, the error sources which are referred to the input of an integrator other than the first one, are modified at least with first order NTF. This means that, the similar error sources which are modeled as error sources at the input of the first integrator and at the input of the other integrator inputs effect the system performance differently. The former directly affect the performance, whereas the effect of the latter would be degraded by at least one order of NTF.

Thus, the developed system calculates the performance of the system in a similar fashion. However, almost in all cases, it was observed that the first integrator loop is the major contributor in noise calculation.

3.3. Multibit Architectures

In the design space of a Sigma-Delta ADC, one dimension is the resolution of the quantizer in the ADC architecture. In [3] it was shown that, increasing order, OSR and quantizer resolution value improves the performance of the ADC design. Increasing order of the design will add extra integrator blocks. Hence, the area and power consumption will grow and may exceed the allowed limits. A better approach for improving the design may be lying in the other design dimension: quantizer resolution

[84, 85, 86, 87, 88, 89, 90]. The increased quantizer resolution not only improves SNR but also relaxes the slew-rate and settling time requirements. However, multibit quantizers generally require multibit DAC's. Thus, DAC performance becomes important, since the error generated by DAC will be fed directly to the input and may deteriorate the desired performance [7, 29]. A very clever way to overcome this difficulty is to use cascaded architectures and use multibit quantizer in latter stages. This approach will improve performance and the effect of DAC errors will be limited [5]. Thus, there are multibit designs available in the literature in which the only quantizer [84] or one of them in cascaded architectures [7, 5, 82] was designed with more than one bit resolution. These designs present a performance improvement compared with the classical Sigma-Delta configurations.

Unfortunately, despite their advantage to increase the SNR ratio, they also introduce DAC non-linearity to the system. Since DAC non-linearity is a major error and non-linearity source for the Sigma-Delta ADC, the designs require more expertise. Some designers overcome this problem by using moderate resolutions in the quantizer but using only single bit DAC in the feedback path [7].

The equation database in the developed tool has the DAC nonlinearity model as given in [82]:

$$\begin{aligned} Y(z) = z^{-1}X(z) + d_{2N-3}(1-z^{-1})^{L}E_N(z) \\ + d_{2N-3}(1-z^{-1})^{L-L_N}E_D(z) \end{aligned} \tag{3.15}$$

In (3.15), E_D is the non-linearity error of the last stage and it is clear that it will be shaped by $L - L_N$ order. The inband error power for a 2 stage cascaded architecture can be expressed as:

$$P_Q \approx d_1^2 \left(\sigma_Q^2 \frac{\pi^8}{9OSR^9} + \sigma_D^2 \frac{\pi^4}{5OSR^5} \right) \tag{3.16}$$

3.4. Continuous-time models

The main error sources that deviate the performance of CT Sigma-Delta ADC from ideal behaviour are amplifier finite gain, integrator non-idealities, time constant error, non-linearity, additive noise, jitter, and excess loop delay.

The ideal transfer function of CT integrator is given in (3.17). However this behaviour is ideal and more reasonable model is given in (3.18) which involves nonidealities.

$$H(s) = \frac{1}{\tau \times s} \tag{3.17}$$

$$H(s) = \frac{B/\tau}{C + s} \tag{3.18}$$

In (3.18), B is $\frac{Av}{1+Av}$ and C is $\frac{1/\tau}{1+Av}$. Av is the amplifier gain and τ is the time constant of the integrator. The time constant may have different form regarding the implementation of the integrator. There are four structures which are widely used in CT integrators [63]. The type of integrators are RC, MOSFET-C Gm-C and Gm-MC filters. Regarding the finite amplifier gain, the noise power can be calculated as given in (3.19).

$$P_A = \frac{\Delta^2}{12K}\left[\frac{1}{A^{2L}M} + \sum_{m=1}^{L}\frac{\pi^{2l}L(L-1)\ldots(L-m+1)}{(2m+1)M^{2l+1}A^{2(L-m)}m!}\right] \tag{3.19}$$

An advantage of using CT integrator is that the settling error is less critical compared with the DT integrator. However, it should not be discarded and may be effective in some high-speed applications.

Another source of error is the influence of gain bandwidth product (GBW) [63].

Small values of GBW introduce another pole to the system which degrades the system performance considerably [64].

Mismatch errors also cause significant errors such as instability in high-order single-loop systems. Also matching has a very important effect in cascade architectures. Especially for CT Sigma-Delta ADC circuits for which coefficients are calculated by transforming DT transfer functions, there are many coefficients and matching is crucial for these coefficients. There are methods proposed in the literature [65] such as direct synthesis method where there are less components and their sensitivity is less to element tolerances.

Integrator time constant is also a very important parameter in ADC performance. Parasitic capacitances may introduce a deviation in the pole location.

$$H(s) = \frac{A}{1 + sC(1 + \varepsilon_\tau)/g_0} \tag{3.20}$$

In (3.20) ε is the error term introduced because of parasitics and g_0 is the output conductance. The ε term may have different forms regarding four different types of integrators defined above. Also the time constant τ should be selected carefully for optimum performance [65]. The deviation from optimum value degrades system performance severely.

Non-linearity has also great impact on the system performance. Especially in voltage to current transformation in integrators (such as Gm-C or RC), non-linearity is introduced to the system. The result is an increase in harmonic distortion which may dominate maximum SNR value that may be achieved. Other source of non-linearity is the multibit DAC. The linearity of the DAC should be lower than the required resolution which is a very hard objective to achieve generally. Some correction mechanisms are available such as dynamic element matching (DEM) and calibration.

Additive noise sources model the noise contribution of blocks such as offset error

of integrator and DAC. Thermal noise also is very important and may deteriorate the system performance. Noise power of thermal noise is given in (3.21)

$$P_{th} = \frac{KT}{4C} \times \frac{\pi^{2L}}{(2L+1)M^{2L+1}} \tag{3.21}$$

Comparator metastability is reported to be another error source and it may be modeled as jitter noise [91]. There are some methods proposed to overcome this problem as given in [92]

Excess loop delay is one of the major error sources which causes instability. The performance of the ADC does not degrade, but rather the ADC becomes non-functional. In this case, additional poles are introduced to STF and NTF and cause instability. However, this instability may be avoided with design techniques [91, 93, 94, 95] such as utilization of extra feedback path or digital compensation.

Clock jitter is the major error source in CT Sigma-Delta ADC's. The sampling process occurs at the input of the quantizer block and at the DAC. The jitter error caused by the sampling at the input node of the quantizer is shaped by the NTF [7] and hence the impact on the system performance is limited. On the other hand, any jitter error occuring in the DAC is directly added to the signal path and affects the system performance. The effect can be observed as the increase in the in-band noise power. There are different approaches to model the jitter error. First approach is white noise approximation [91]. This approach models the SNR degradation as:

$$SNR_J = 10log(\frac{1}{16MB^2\sigma_j^2}) \tag{3.22}$$

The other approach is given in [96] where jitter noise includes both white and shaped components. The most popular approach to overcome the jitter error is to use multibit NRZ DAC's [97]. Thus, the system becomes less sensitive to clock variations.

The error sources for CT Sigma-Delta modulator are defined above and the effect of these error sources are examined via behavioral models. These models were defined as macromodels in HSpice. The Simulink environment did not perform well regarding the simulation time and since both CT and DT signals are present in simulations, convergence problems arise. HSpice is inherently continuous and it is easier to define DT components by utilizing a clock signal. Also, parametric HSpice simulation is closer to the device-level simulation and blocks can be replaced with real circuits for observing their performances.

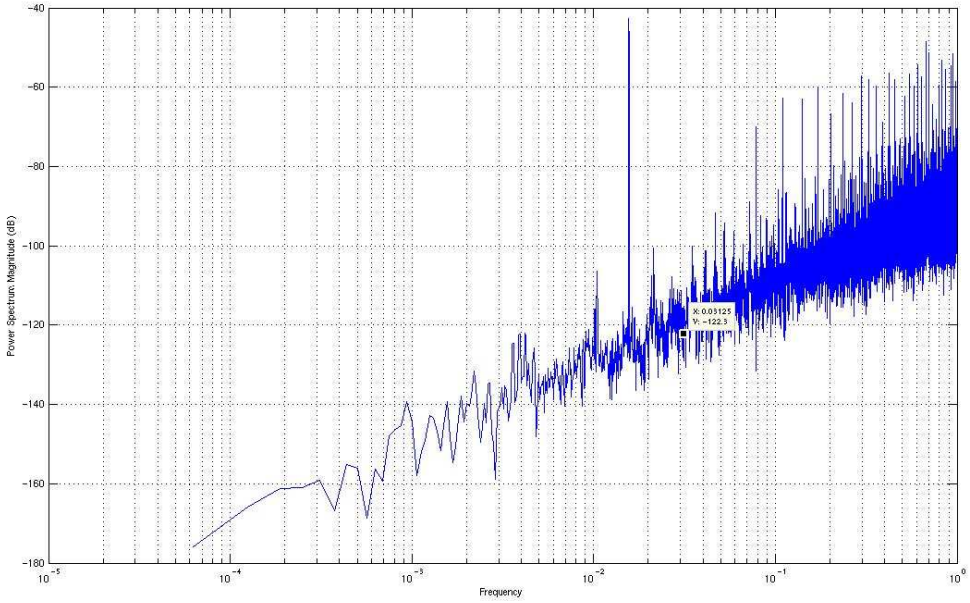

Figure 3.14. PSD of First Order CT Sigma-Delta Modulator

In Figure 3.14, The PSD of a non-ideal CT Sigma-Delta modulator output is given. The ideal output provide an SNR of 76dB. A small deviation in the pole of the CT integrator, results in a considerable performance loss approximately 25dB. Such simulation results were performed and the results were compared with the models. The behavioral simulations provide verification for developed models as well as very valuable information about other aspects in the design process that are not modeled. Since the integrator is a CT filter, the pole positions may introduce error to the system which will reduce the system performance. This error was modeled and verified through various simulations. Two of the other noise sources are integrator and comparator offset values.

These errors can be modeled as additive noise sources at the input node and at the input of the comparator input. The effects of these errors were observed as in Figure 3.15 and Figure 3.16. They were compared with analytical models. There are some other issues in the design process which are not modeled before in the literature such as antisymmetric character of the comparator output. As it is presented earlier, the linearity of the DAC is very important in the ADC performance and if DAC structure is not designed to avoid errors occurred due to deviations in its input, the system performance becomes very sensitive to comparator output voltages. If DAC circuit is using the voltage provided by the comparator as a reference, then this sensitivity would pass DAC circuit and reach to the input node where all errors combined to raise the noise floor at the output. Hence, the symmetry of the comparator output may be improtant for some designs and it should be considered within the design process. In Figure 3.17 the effect of the error on system performance which is caused by non linear characteristic of the comparator circuit, is illustrated. The percentage of deviation in the final value of the comparator output level result in a very large SNR drop. This effect was not modeled in the literature before.

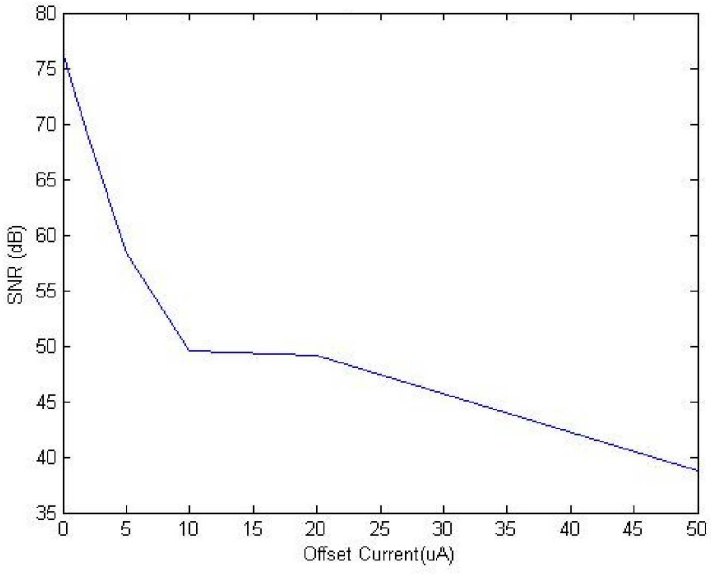

Figure 3.15. SNR vs. Integrator offset voltage

The CT models with examinations performed by the results of the behavioral simulation provide insight to the CT Sigma-Delta ADC design.

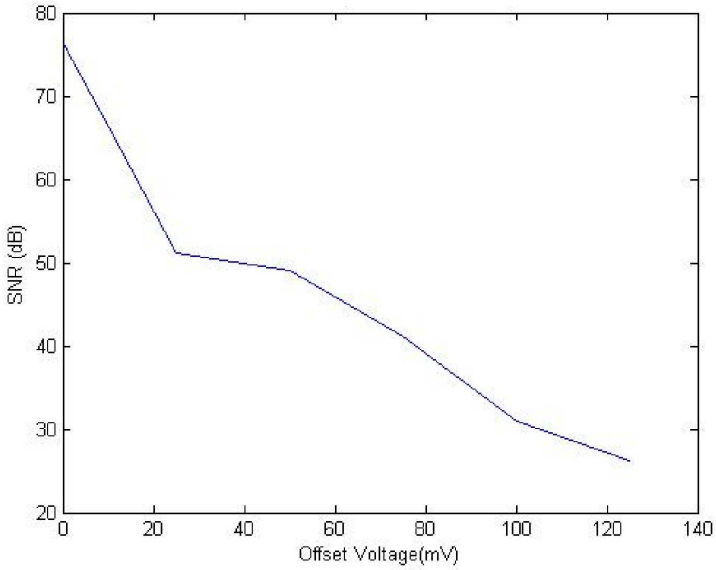

Figure 3.16. SNR vs. Comparator offset voltage

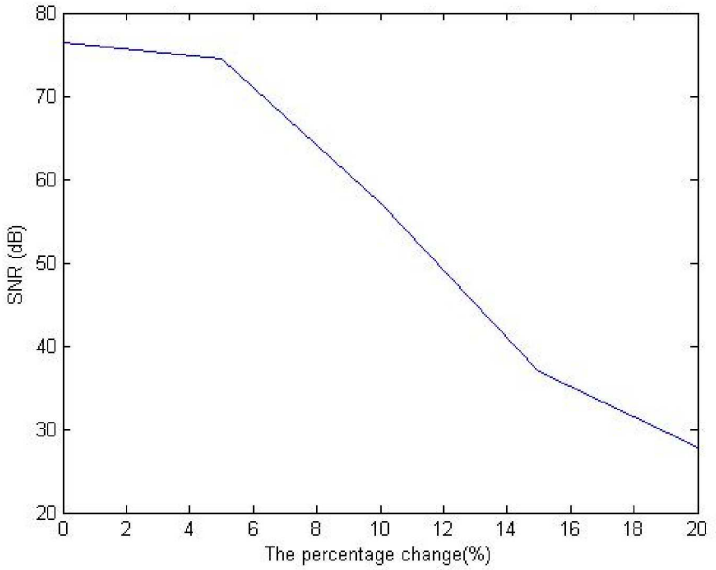

Figure 3.17. SNR vs. The percentage change of the comparator negative output

4. DESIGN AUTOMATION OF SIGMA-DELTA ADC'S

In a Sigma-Delta ADC design, the architecture should be determined before the design can proceed further. In the approach presented in this thesis, a coarse SNDR calculation is carried out, as well as area and power consumption values, in order to estimate the performance of the ADC. The area and power values are estimations which are provided by Performance Estimator module of the system. On the other hand, SNDR can be calculated using ideal components in the earliest stage and then can be modified by including non-linearities. SNDR of the ideal system can be calculated as given in [6]. Non-ideality contributions to overall noise value can be calculated with the models and approaches described in earlier sections.

Stability is another important issue in Sigma-Delta ADC design [3, 36]. Analytically, it is not easy to analyze stability of every possible sigma-Delta Architecture. Thus, the preferred method for stability check is to perform a behavioral simulation or stick to structures that were proven to be stable [82]. In a design automation system, in which many candidate solutions are available, it is not feasible to perform stability analysis for each candidate solution. It is not even feasible to check it at the end of the design, since stability check may fail and force the designer to make new iterations. Therefore, it is not a suitable approach for the design automation tool. The approach in the design automation system presented here is to follow a flexible approach. The user may select to guarantee the stability [29], then the tool uses a cascaded system where at most second order modulators are used. The reason for this is that first and second order systems are guaranteed to be stable. Then, for higher order systems like a third order Sigma-Delta modulator, the system forms a MASH structure denoted as 2-1. The user may wish to discard the stability restrictions and search for a solution from a wider design space. In that case, the user should utilize an add-on tool developed in MATLAB to check the stability. This tool outputs the transfer function of the Sigma-Delta modulator with non-idealities, for which the user may check the stability.

There are some input parameters for ideal SNDR calculation at the beginning.

These are oversampling ratio (OSR) and order of the system. Depending on the operation mode of the design system, these parameters may be provided by the user or they may be chosen by the design system. It is related with the operation modes of the design system. These operation modes provide the flexibility to the users of the developed system. The algorithm of the design automation system is given in Figure 4.1.

The algorithm uses a structure called ‘ worm ’ which is a solution for the ADC. The initial number of worms depends on the number of elements in the library. The worm contains all information such as OSR, supply voltage, configuration, noise values, SNDR, capacitor values, and the block parameters such as amplifier gain, comparator offset. These worms are passed to functions in order to modify their relevant value, such as quantization noise or SNDR.

4.1. Modes of Operation

There are three different modes of operation in the tool. The first one is to gather the block parameters from the user and check the feasibility of the design. This mode can be considered as a quick validation of the system. This mode is useful compared with time consuming behavioral simulations. The user provides all the required design parameters such as OSR, supply voltage, configuration, capacitance values. The tool calculates the achievable performance values. Hence, the user may not only verify the candidate design but also gather information about the performance of the system. Thus, even if the design may not succeed in providing the required performance, the user may have an idea how far the candidate design is far away from the desired performance values.

The other operation mode is the semi-custom mode, where the user specifies a library in which the characteristics of the blocks (which may have multiple versions) in an ADC are given. Then, the system estimates the performance of the ADC realized by the elements of these libraries and provides a solution where the design is optimized within the candidate designs with respect to an adjustable cost function containing area and power information. If only some blocks are available or only some of the blocks are

suitable for the desired performances, the system searches the design space to calculate the required parameters for the other blocks. For example, if only comparators are available in the library, the system searches for amplifier parameters, which satisfy the desired ADC specifications. However, to estimate amplifier parameters such as area and power, a PE is required. Performance estimation is much more accurate compared with primitive macromodels of the blocks, but, it should be fast because it is inside the optimization loop.

The last mode of operation is useful when there is no solution with the given libraries. The system can perform parameter sweep for the selected design parameters and utilizes the PE. Then, the designer can find the block, which should be replaced or improved in terms of that parameter. The ideal way for the design is to use device level tools [25] in order to realize the blocks and finally the converter. The blocks marked with "*"in Figure 4.1 show that these blocks are optional and may be skipped by user.

4.2. Architecture Selection

One of the most important feature of a design automation tool should be the ability of the finding best architecture for provided specifications. The design automation systems which have to work in a design space with many dimensions should be capable of selecting an architecture and there may exist many solutions which satisfy all specifications. These solutions differ from each other by their area and power consumptions. Hence, even if there may be many solutions, there exists an optimum configuration regarding the area and power consumptions.

It is not convenient to select an architecture in the system level if the cost of the selection can not be determined accurately. Thus, in order to compare the performances of the architectures, SNDR can not be the only metric for performance calculation. Power and area consumptions should also be considered. However, these values require knowledge of device level information and cannot be determined for a solution candidate at system level. It has to be passed to device level tools for sizing and optimization; hence, area and power consumptions can be calculated. However,

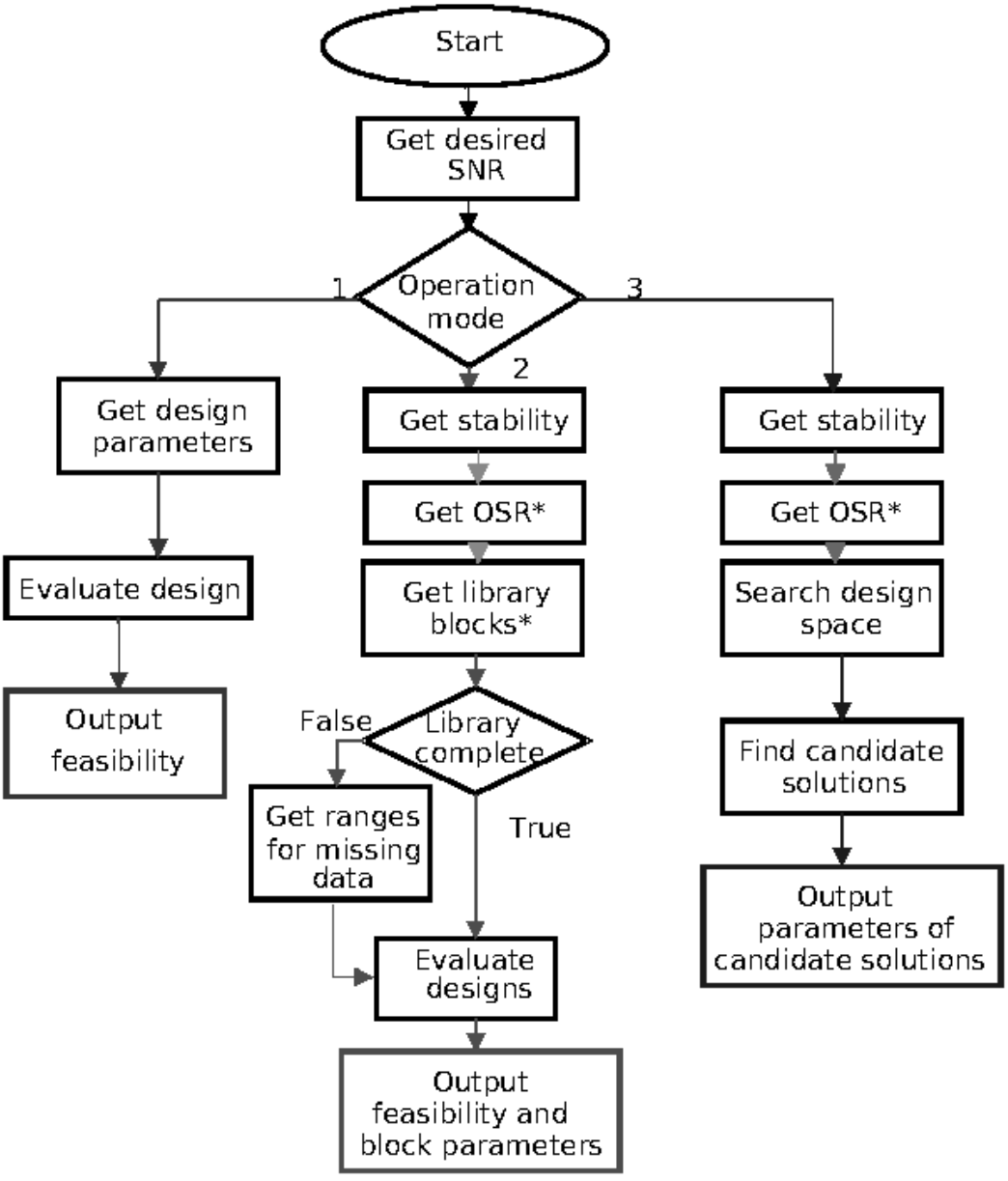

Figure 4.1. Algorithm

this approach requires integration of system level and device level design automation tools.

In our approach, these levels are separated and should not be merged in a design automation system. Otherwise, calculating performance of candidate solution will require more effort and even iterations may be introduced. In order to overcome this problem, a performance estimation tool which works between system level and circuit level and communicates with the system level tools was implemented. This tool provides device level information to system level tools in fast manner. With the utilization of the performance estimator, candidate solutions may be evaluated at the system level and accurate architecture selection becomes possible. This approach was used in design automation system.

The developed tool as a first step, generates all possible configurations. The generation is closely related with the selections of the user. The user may select stability to be guaranteed. In that case the tool only generates cascaded architectures and use only up to second order for each stage. In other words, there will not be any third order loop in the candidate solutions. Considering the selections of the user, the tool searches for possible configurations. Most of these are configurations already reported in the literature [98, 99, 84, 100, 101, 102, 103, 104, 105, 106, 85, 107, 108, 109]. The tool generates possible configurations as solution candidates. The number of candidates is related with the specifications that the user provides. Some performance metrics, such as relative SNDR values, may be achieved not only with second order configurations but also with third and fourth order architectures. Although it may not seem reasonable to select a fourth order architecture for a performance value that may be achieved by a second order system at first sight, this way is not the case sometimes. There may be more adequate solutions which has an order of four but lower OSR values and more relaxed amplifier specifications. On the other hand, second order architecture may require higher OSR values which may result in unreasonable clock frequencies. Hence, the tool has more relaxed limits for the candidate selections. Then, after the generation of the candidate configurations, these configurations passed to error calculation routines and performance of each candidate is calculated. An example for

the process is provided in the section six. The performance parameters are SNDR, area and power consumptions for the candidate. As a last step, the most appropriate architecture is selected by a cost function which takes these performance metrics as input.

One step further than the available approach is not to limit the design space with the available architectures. The design space is defined by single loop or cascaded architectures previously known, such as 2-1-1 cascaded or third order single loop configurations. However, a more innovative way of selecting an architecture is to consider all possible connection schemes [32] and then search for a candidate architecture.

The second approach may be called "architecture generation"since the result may not be previously known or in other words previously available in the literature. In order to achieve this goal, a tool was developed in MATLAB [32, 110]. The aim of this tool is to provide different configurations with similar performances. Then, an additional optimizer was developed which selects the best architecture from these.

4.2.1. Architecture Generation

The developed MATLAB tool works independent of the modulator order and finds all possible Sigma-Delta modulator topologies satisfying a desired system response with the minimum number of signal paths which in turn leads to minimum complexity. The operation of the tool may be divided into two basic parts: *(1) Generation of the symbolic transfer functions (STF and NTF) for any given topology of any order and with any complexity, (2) Generation of all possible topologies with minimum number of signal paths that is, minimum complexity.*

First section of the tool uses the symbolic analyzer functions of the MATLAB. A SPICE like netlist is provided to the system which defines all the connections. A generic netlist may be used for all designs. However, the tool allows the user to modify the available connections. The benefit of this approach is that, some connections may be removed, which the user may not wish to implement. For example, the user may

prefer to use only feedback paths and may remove the feedforward paths. The tool provides the transfer function at every node. Hence for each node, including the output node, a parametric transfer function may be obtained. Also, the user may use not only ideal blocks but also other non-ideal models. The following blocks have been defined in the tool. However, this list is not exclusive and other blocks may be added if necessary.

- *INTEGRATOR:* An integrator block with the z-domain transfer function of $\frac{1}{1-z^{-1}}$. It has no delay.
- *INTEGRATORD:* An integrator block with the z-domain transfer function of $\frac{z^{-1}}{1-z^{-1}}$. It has one unit delay for which the capital letter D stands for.
- *NONIDEAL_INTEGRATOR:* An integrator block with the z-domain transfer function of $\frac{b}{1-cz^{-1}}$ where the parameters b and c stand for the integrator non-idealities that have been explained earlier.
- *NONIDEAL_INTEGRATORD:* An integrator block with the z-domain transfer function of $\frac{bz^{-1}}{1-cz^{-1}}$ where the parameters b and c again stand for the integrator non-idealities. It has one unit delay for which the capital letter D stands for.
- *ADDER:* That is a dynamic block that is, it works independent of the number of inputs to it. It can add or subtract any number of signals simultaneously if the nodes are defined as positive or negative in the netlist.
- *GAIN:* This is a one-input one-output block used to model the SD modulator architecture coefficients or path gains.

A generic second order SD modulator architecture is shown in Figure 4.2. It comprises all the possible feedback and feedforward paths from g_1 up to g_{15}. The symbolic node equations can be expressed in terms of the path gains g_1 to g_{15} and thus the STF and the NTF in terms of path gains can be generated automatically by the tool. For the specific architecture of Figure 4.2, where x_{22} is the output node, x_1 is the signal input node and x_{21} is the noise input node. The resulting STF and NTF in

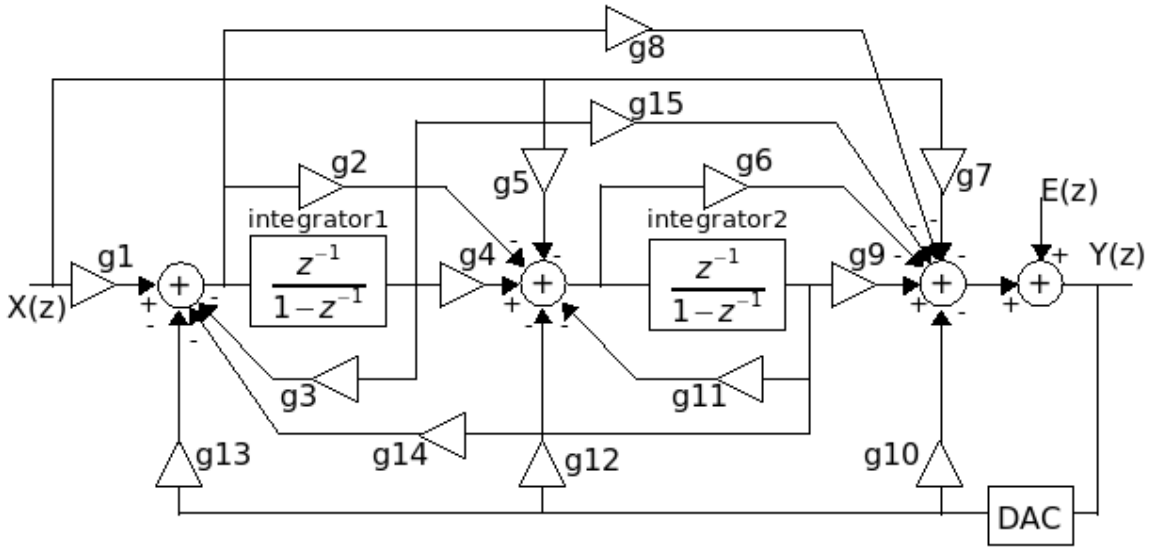

Figure 4.2. The generic standard second order SD modulator architecture comprising all possible feedback and feedforward paths

terms of path gains for the given architecture are as follows:

$$\begin{aligned} STF(z) &= \frac{n_{2_{STF}}z^2 + n_{1_{STF}}z + n_{0_{STF}}}{d_{2_{STF}}z^2 + d_{1_{STF}}z + d_{0_{STF}}} \\ NTF(z) &= \frac{n_{2_{NTF}}z^2 + n_{1_{NTF}}z + n_{0_{NTF}}}{d_{2_{NTF}}z^2 + d_{1_{NTF}}z + d_{0_{NTF}}} \end{aligned} \tag{4.1}$$

where

$$
\begin{aligned}
n_{2_{STF}} &= -g_7 + g_6\, g_5 + g_1\, g_6\, g_2 - g_1\, g_8 \\
n_{1_{STF}} &= -g_1\, g_9\, g_2 + g_{14}\, g_2\, g_7 - g_1\, g_8\, g_{11} - g_1\, g_{15} \\
&\quad -g_{14}\, g_5\, g_8 - g_7\, g_1\, 1 + 2\, g_1\, g_8 - g_3\, g_7 - g_1\, g_6\, g_4 \\
&\quad +g_3\, g_6\, g_5 - g_9\, g_5 + 2\, g_7 - 2\, g_1\, g_6\, g_2 - 2\, g_6\, g_5 \\
n_{0_{STF}} &= -g_7 + g_3\, g_7 + g_1\, g_{15} - g_1\, g_8 + g_9\, g_5 + g_7\, g_{11} \\
&\quad -g_3\, g_6\, g_5 + g_6\, g_5 + g_1\, g_6\, g_2 + g_1\, g_6\, g_4 + g_1\, g_9\, g_4 \\
&\quad +g_1\, g_8\, g_{11} + g_1\, g_9\, g_2 + g_{14}\, g_5\, g_8 - g_{14}\, g_5\, g_{15} \\
&\quad -g_{14}\, g_4\, g_7 - g_1\, g_{15}\, g_{11} - g_3\, g_7\, g_{11} - g_3\, g_9\, g_5 - g_{14}\, g_2\, g_7 \\
n_{2_{NTF}} &= 1 \\
n_{1_{NTF}} &= -2 + g_3 + g_{11} - g_{14}\, g_2 \\
n_{0_{NTF}} &= 1 - g_{11} + g_{14}\, g_2 - g_3 + g_{14}\, g_4 + g_{11}\, g_3 \\
d_{2_{STF}} = d_{2_{NTF}} &= -g_8\, g_{13} + g_{10} + g_6\, g_{13}\, g_2 - g_6\, g_{12} + 1 \\
d_{1_{STF}} = d_{1_{NTF}} &= -2 + g_3 + g_{11} - g_9\, g_{13}\, g_2 + g_{10}\, g_3 + g_9\, g_{12} \\
&\quad +g_{10}\, g_{11} - g_{14}\, g_2 - g_8\, g_{11}\, g_{13} - g_{14}\, g_{10}\, g_2 + 2\, g_8\, g_{13} \\
&\quad -2\, g_{10} - g_{15}\, g_{13} + 2\, g_6\, g_{12} - g_6\, g_3\, g_{12} + g_{14}\, g_8\, g_{12} \\
&\quad -g_6\, g_{13}\, g_4 - 2\, g_6\, g_{13}\, g_2
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
d_{0_{STF}} = d_{0_{NTF}} &= 1 - g_3 + g_{10} - g_{11} + g_6\, g_{13}\, g_2 + g_{14}\, g_2 + g_8\, g_{11}\, g_{13} \\
&\quad -g_{15}\, g_{11}\, g_{13} + g_{14}\, g_{15}\, g_{12} + g_{14}\, g_{10}\, g_2 + g_{14}\, g_4 \\
&\quad +g_9\, g_3\, g_{12} + g_{10}\, g_{11}\, g_3 + g_{11}\, g_3 + g_9\, g_{13}\, g_4 + g_9\, g_{13}\, g_2 \\
&\quad +g_6\, g_3\, g_{12} - g_{14}\, g_8\, g_{12} + g_6\, g_{13}\, g_4 + g_{14}\, g_{10}\, g_4 + g_{15}\, g_{13} \\
&\quad -g_9\, g_{12} - g_6\, g_{12} - g_{10}\, g_{11} - g_{10}\, g_3 - g_8\, g_{13}
\end{aligned}
$$

The second phase of the tool is to use the generated symbolic STF and NTF to match with user provided desired STF and NTF. The default for desired STF and NTF are the ideal ones. In order to achieve this, the coefficients of the symbolic STF and NTF given in Eq. 4.2 are matched with those of the numeric STF and NTF defined by the user respectively.

If the numeric transfer functions to be realized are in the form;

$$STF(z) = \frac{p_2 z^2 + p_1 z + p_0}{q_2 z^2 + q_1 z + q_0}$$
$$NTF(z) = \frac{r_2 z^2 + r_1 z + r_0}{q_2 z^2 + q_1 z + q_0} \tag{4.3}$$

where p_i, q_i, r_i are all real numbers, then the tool generates the following set of equations in terms of fifteen path gain parameters from g_1 to g_{15}:

$$\begin{aligned} n_{2_{STF}} &= p_2, \quad n_{2_{NTF}} = r_2, \quad d_{2_{STF}} = d_{2_{NTF}} = q_2 \\ n_{1_{STF}} &= p_1, \quad n_{1_{NTF}} = r_1, \quad d_{1_{STF}} = d_{1_{NTF}} = q_1 \\ n_{0_{STF}} &= p_0, \quad n_{0_{NTF}} = r_0, \quad d_{0_{STF}} = d_{0_{NTF}} = q_0 \end{aligned} \tag{4.4}$$

Then, this set of equations is solved simultaneously to generate a set of solutions. Each element of this set corresponds to a set of path gains from g_1 to g_{15} which in turn corresponds to a different Sigma-Delta modulator architecture. Since the number of equations is less than the number of variables, it introduces a freedom to the user in generating many different topologies all realizing the same desired response [110]. Then the resulting set of equations are solved by invoking the MATLAB's symbolic toolbox which is constructed on the MAPLE kernel.

The MATLAB tool has some important advantages with respect to other similar tools such as [111]. They search for feedforward or feedback configurations that already available in the literature. On the other hand, this tool can find different topologies since occurrence of both several feedforward and several feedback paths is allowed.

A flowchart summarizing the operation of the tool is given in Figure 4.3.

In order to illustrate the new architecture generation feature an example is provided. For the desired transfer functions given in 4.5 which define the standard second order system response, the tool has generated 70 different architectures some of which

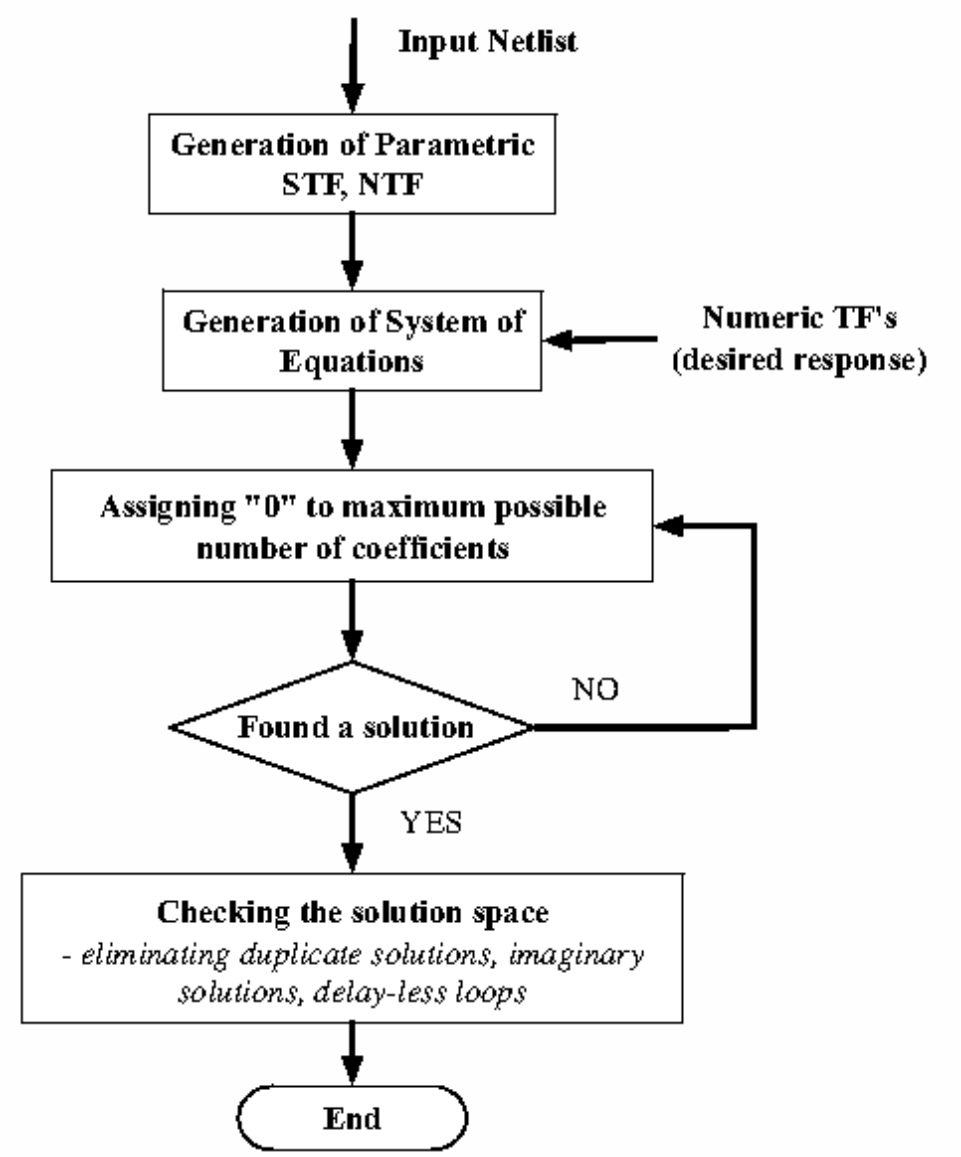

Figure 4.3. The flowchart of the automatic architecture generator tool

are given as examples in Table 4.1.

$$\begin{aligned} STF(z) &= z^{-2} \\ NTF(z) &= (1-z^{-1})^2 \end{aligned} \tag{4.5}$$

The solution in the first column of Table 4.1 is the standard second order Sigma-Delta modulator architecture shown in Figure 4.4. This can be verified by making $g_9 = g_{13} = 1$.

Table 4.1. Architectures Found for the example

Coeffs.	Different Solutions				
	1	2	3	4	5
g_1	g_{13}	$\frac{1}{g_4 g_9}$	0	$\frac{4}{g_8}$	g_1
g_2	0	0	0	0	0
g_3	0	0	-0.5	0.5	0.5
g_4	$\frac{1}{g_9 g_{13}}$	g_4	g_4	$\frac{1}{4g_{14}}$	$\frac{1}{g_1 g_9}$
g_5	0	$\frac{2}{g_9}$	$-\frac{4}{g_6}$	0	0
g_6	0	0	g_6	0	0
g_7	0	0	-4	-4	0
g_8	0	0	0	g_8	0
g_9	g_9	g_9	$-\frac{g_6}{2}$	0	g_9
g_{10}	0	0	0	0	0
g_{11}	0	0	0.5	-0.5	-0.5
g_{12}	$\frac{2}{g_9}$	0	0	$\frac{2}{g_8 g_{14}}$	$\frac{2}{g_9}$
g_{13}	g_{13}	$\frac{1}{g_9 g_{14}}$	$-\frac{2}{g_4 g_6}$	0	0
g_{14}	0	0	$\frac{1}{4g_4}$	g_{14}	$\frac{g_1 g_9}{4}$
g_{15}	0	$-2g_4 g_9$	0	$\frac{g_8}{2}$	0

The architecture generation feature may be observed by the solution given in the third column. This architecture is extremely interesting in the sense that the input signal is not directly fed to the first integrator but given to the second one in the loop as given in Figure 4.5. By selecting $g_3 = -0.5,\ g_4 = 1,\ g_5 = -2,\ g_6 = 2,\ g_7 = -4,\ g_9 = -1,\ g_{11} = 0.5,\ g_{13} = -1,\ g_{14} = 0.25$ a very interesting topology can be achieved.

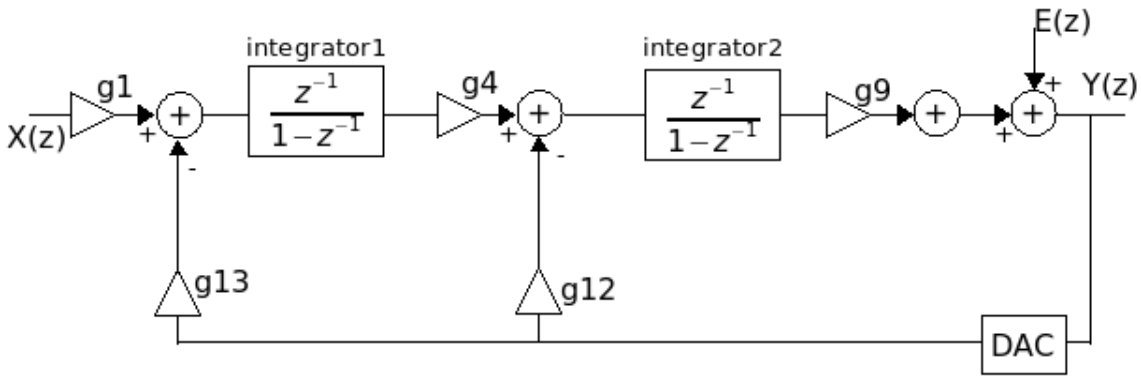

Figure 4.4. The standard second order SD modulator

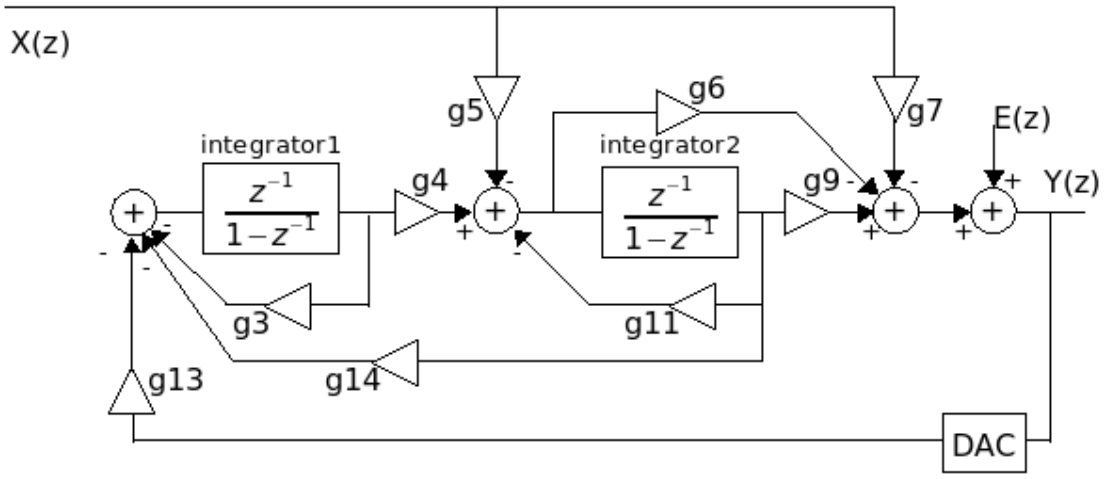

Figure 4.5. The solution proposed in column 3 of Table 4.1

The architectures shown in Figure 4.4 and 4.5 have been implemented and simulated in MATLAB Simulink with ideal blocks and the PSD plots for these architectures are shown in Figure 4.6 [32]. It can be observed that both architecture provide the same SNR value.

4.2.2. PERFORMANCE CONSIDERATIONS

In the previous section, it was demonstrated that different architectures may provide same performance values. However, these candidate solutions, which realize the same STF and NTF can differ from each other by means of some criteria; namely, area occupied by the circuit, power consumed by the circuit, and the sensitivity of the circuit performance. Hence, considering these metrics, the optimal solution should

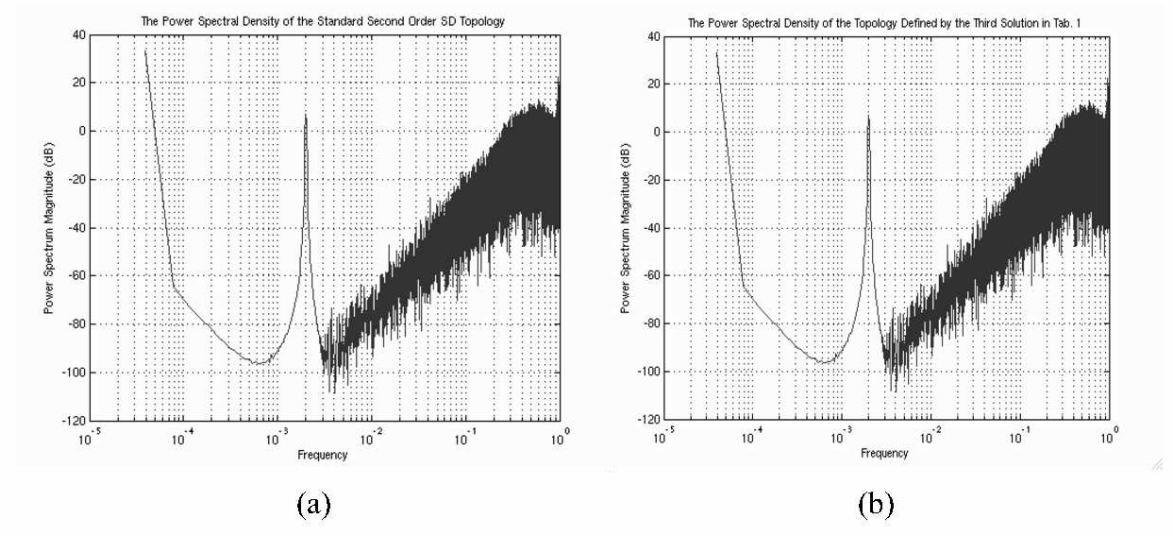

Figure 4.6. The PSD plot for the solution given for both solutions

have to be identified.

4.2.2.1. Area. In integrated circuit design, area is an important criterion for comparing different implementations since silicon area directly related with the cost of the design. The area for an ADC with a specific order may be different even the number of blocks remain the same. The reason is that, the connection scheme can be different as well as number of coeficients. Since these coefficients are realized by switched-capacitor circuitry, coefficient value and number of coefficients directly defines the area and the number of capacitors. Hence, same order Sigma-Delta ADC's can be different regarding the total silicon area. As a result, sum of the values of the capacitors used in the implementation gives a variable to compare different topologies. Calculated values of this variable for the implemented topologies are tabulated as follows:

Table 4.2. Capacitor values used in implementation for each architecture

	Architectures												
	Std.	1	2	3	4	5	6	7	8	9	10	11	12
Capacitance (pF)	16	18	18	22	16.5	20	18	25	26	29	20	36	22

4.2.2.2. Sensitivity. When performance of a ΣΔ modulator is considered, SNR can be a parameter of interest. In order to describe a modulator as a high performance one, its SNR value must be better than most of the others. The relation governing the SNR of a modulator was given in Equation (4.6). P_Q, which is the in band noise power, has the formula given in (4.7)

$$SNR(dB) = 10log_{10}\left(\frac{A^2/2}{P_Q}\right), \tag{4.6}$$

$$P_Q = \int_{-f_d/2}^{f_d/2} S_Q(f)df = \int_{-f_d/2}^{f_d/2} S_E(f)|NTF(z)|^2 df \cong \frac{\Delta^2}{12}\frac{\pi^2}{3M^3}. \tag{4.7}$$

In this formula, $S_Q(f)$ is the power spectral density of the shaped quantization noise, whereas $S_E(f)$ is the PSD of unshaped quantization noise. Also, Δ is the quantization step and M is the oversampling ratio. Since PSD of the unshaped quantization is generally considered as uniform distributed white noise, the parameter that affects the SNR appears to be $|NTF(z)|^2$ term. The parameters that make the topologies different are the coefficients and their values. All in all, sensitivities of the SNRs of different architectures to the path gains is the parameter to dealt with. Since SNR is directly related to $|NTF(z)|^2$, sensitivity of this term to the path gains is important in comparing the sensitivities of SNRs of the implemented architectures.

There are 15 coefficients in the second order ΣΔ modulator. Sensitivity of $|NTF(z)|^2$ to each coefficient can be calculated. However, the problem is that $|NTF(z)|^2$ can be 0 for some cases. Hence, a different definition of sensitivity which is called *Semi-relative Sensitivity* is applied, which is discussed in [112] further. The equation for this sensitivity definition is

$$Q_x^{P(x)} = x\frac{dP}{dx}. \tag{4.8}$$

By using (4.8), sensitivity for the cases when $P(x) = 0$ can be accounted for. So $Q_{g_i}^{|NTF(z)|^2}$ will be the parameter of interest, which has the following equation:

$$Q_{g_i}^{|NTF(z)|^2} = g_i \frac{\delta |NTF(z)|^2}{\delta g_i}. \tag{4.9}$$

Having semi-relative sensitivity metric in hand, it is possible to reach a value that gives an idea about the sensitivity of architecture's SNR to the path gains[113]:

Below NTF and STF sensitivities of each architecture is tabularized for a sample run:

Table 4.3. STF and NTF sensitivities of each architecture

	Architectures												
	Std.	1	2	3	4	5	6	7	8	9	10	11	12
STF	16.2	17.7	15.8	15.4	22.6	27.7	17.7	27.7	41.8	41.8	23.8	66.2	16.6
NTF	13.7	9.4	11.8	34.2	28	13.7	9.4	13.7	34.2	34.2	15.4	15.4	35.1

As it can be observed from Table 4.3, 12^{th} architecture has the highest sensitivity to the coefficients. As an example to see this effect, SIMULINK simulations have been done for 12^{th} architecture using models for each block. For 12^{th} architecture, g_{14} has the highest sensitivity value, so effect of a change in that coefficient was observed. For a 10 percent change in g_{14}, SNR decreased by 16 dB. Also, in order to see the success of approach, effect of the change in a gain for an architecture with low sensitivity has been observed. For this aim, sixth architecture has been chosen since it has the lowest sensitivity. g_5 which has the lowest sensitivity value has been selected for the path. When its value is changed 10 percent SNR value decreased by only 1.1 dB. As it can be observed, change is small which proves that approach is consistent.

4.2.2.3. Power. Power consumption is another important aspect of IC design. Hence, architectures with lower power consumptions should be found. In a $\Sigma\Delta$ modulator,

most of the power is consumed by the integrator. A switched-capacitor integrator includes operational amplifier, capacitors and switches. High portion of this power is due to operational amplifier. Therefore, $\Sigma\Delta$ modulator power is highly correlated with the operational amplifier power. If the current flowing through the single output stage is I_{branch} and the current supplied to the single input differential stage is I, power consumed by the operational amplifier is

$$P = (4I_{branch} + 2I)V_{DD}. \tag{4.10}$$

For a typical operational amplifier used in Sigma-Delta ADC's, the power of the operational amplifier is $P = 6.8V_{DD}I_{branch}$. I has the equation

$$I = \frac{g_{meff}V_{ov}}{2}, \tag{4.11}$$

where g_{meff} is the transconductance of the differential amplifier's single transistor having the equation [114]

$$g_{meff} = Bln(2)C_{L,eff}2f_s, \tag{4.12}$$

where f_s is the sampling frequency, $C_{L,eff}$ is the effective load capacitance, B is the number of bits resolution. $C_{L,eff}$ is defined as a multiple, N_C, of the sampling capacitor, C_s. N_C is called excess capacitance factor and has the equation

$$N_C = 1 + \frac{(C_L)(C_s + C_I)}{C_s C_I}, \tag{4.13}$$

where C_L is the load capacitance, C_I is the feedback capacitance. B is

$$B = log_2(\sqrt{\frac{2}{3}DR^2} + 1), \tag{4.14}$$

where dynamic range (DR) is $DR^2 = \frac{P_S}{P_N}$, in which P_S is the maximum signal power

for a sinusoidal signal having equation ${V_s}^2/2$. P_N is the noise power, which is $\Delta^2/12$, Δ being the step size of the quantizer.

A MATLAB tool was developed for the automatic calculation of power consumption [113]. For a sample run, the estimated power values are tabulated in Table 4.4.

Table 4.4. Power Values

	Architectures												
	Std.	1	2	3	4	5	6	7	8	9	10	11	12
Power(mW)	0.8	1.2	0.83	0.58	0.37	1.79	1.2	2.24	1.26	1.42	0.87	1.79	0.58

It can be observed from Table 4.4 that the architectures differ from each other by means of lowest power consumption value.

4.2.3. Coefficient optimization

The MATLAB tool only generates the equations as given in 4.1. The user should select the appropriate numbers for the coefficients. However, this process should be automated and thus, new routines were developed for this purpose. The optimization approach for this kind of equations can be linear programming with non-linear constraints. In addition, the constraints contain division which increase the complexity of the problem. There are some solutions for this kind of problems such as Lagrange relaxation methods [115]. However, the nonlinear constraints are generated automatically and it is not feasible to implent this method in an automation system. The constraints may have various forms and may include division and exponential terms which cannot directly modified to standart nonlinear constraint forms. Hence, automatic generation of nonlinear constraints that may be solved by standard tools is not possible. However, a dedicated tool can be developed for this purpose.

Since the coefficients are zero for most of the connections, it is more efficient

to use brute-force search method. The developed functions search for independent coefficients and evaluate them with a given limit. The other coefficients are either zero or a function of these independent coefficients and this decreases the number evaluations. As the coefficients actually represent the capacitance values for switched capacitor implementation, the evaluations are based on area, power, and sensitivity [113].

As a summary, the developed MATLAB tool was improved by adding optimization functions and added to the design automation system. The user may select to use the known architectures or prefer to utilize the add-on module for better performance optimization.

4.3. Area and Power Estimation

The power and area calculation methods are similar to the one given in [22]. The static power can be separated into three parts: comparator, amplifier, and digital circuit. It is obvious that most of the power will be dissipated by the amplifier in the switched capacitor integrator. This value is calculated by PE. The power consumed by the digital part can be estimated in a coarse manner. The order of the system and the structure of the cascade can be used to estimate the number of latches and blocks of the filter. Thus, a macro model can be formed which defines the power consumption for different orders of filters. This macro model is developed by performing simulations. Then, the measured data are fitted to the macro model. The power of the other blocks are estimated via PE. The area estimation is done as described in [22]. The major problem here is to estimate the area of the blocks dynamically. For this reason, the integration of a performance estimator to the system is required. Utilizing the PE and library, the estimation of the area for all modes of operation described above is quite accurate.

Total area can be calculated by adding the areas of the amplifiers, comparators, digital circuitry and switches. The results will be more accurate by adding a constant, which represents the routing cost. This constant can be present in the technology file

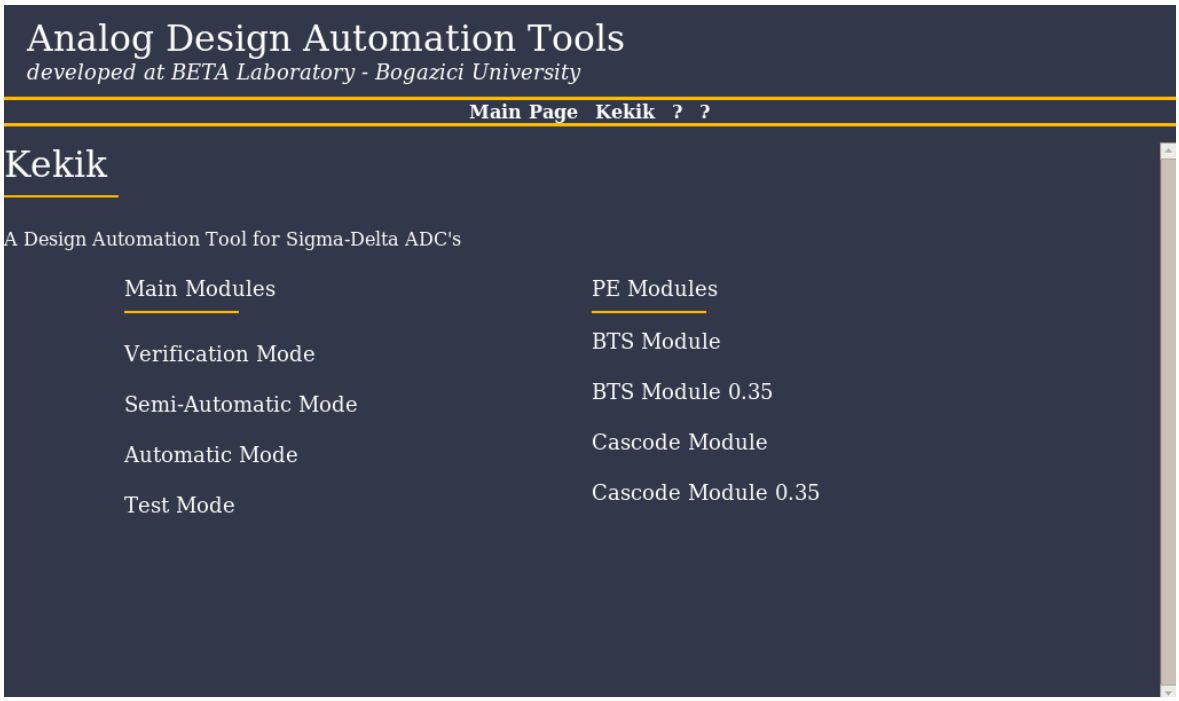

Figure 4.7. Web interface

of the tool.

4.4. Sigma-Delta ADC Designer Web Interface

A web interface was developed in order to make the system easily accessable by other researchers very easily. The common gateaway interface (cgi) was used to connect web interface to the developed system. Cgi can use C and C++ systems, thus the developed system can be easily modified for web use. However, the output functions were modified and rewritten for providing html format output. The web interface and sample output screen are shown in Figure 4.7 and Figure 4.8.

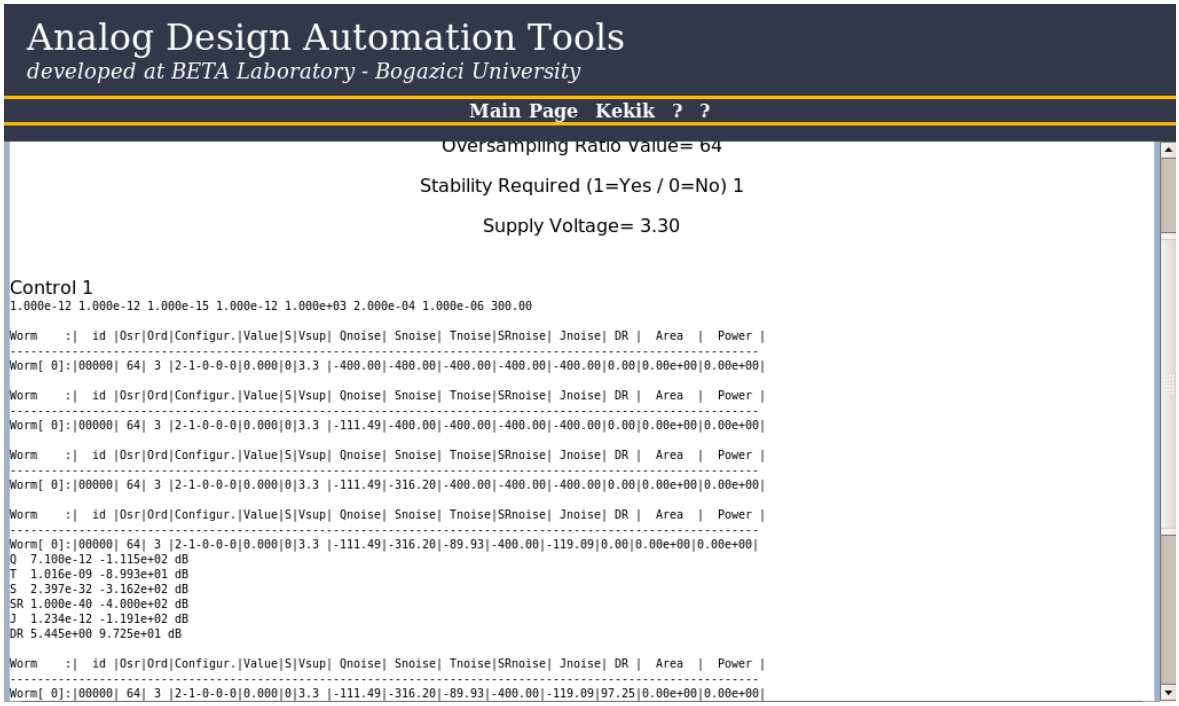

Figure 4.8. Output for verification mode.

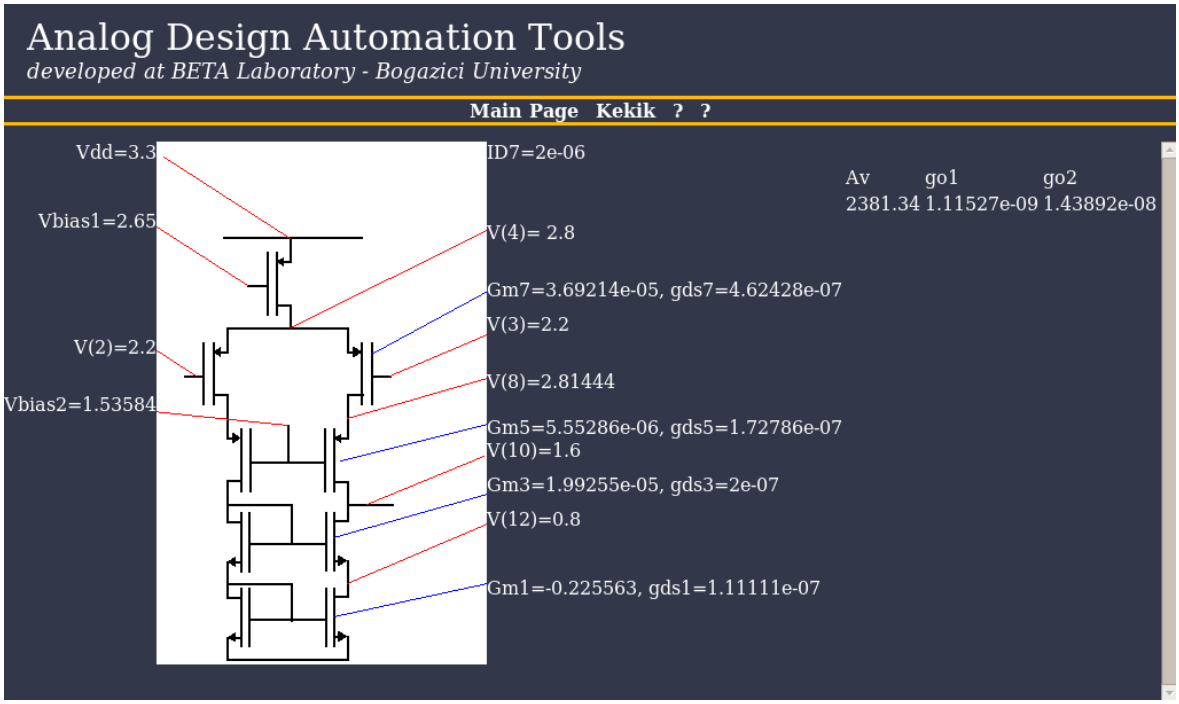

Figure 4.9. Example output for PE cascode module.

5. PERFORMANCE ESTIMATOR

The utilization of a performance estimator (PE) module in system level Sigma-Delta ADC design automation tool is one of the major novelties of this thesis which provides the designer with new opportunities in the process of automatic ADC design. Although PE concept was previously reported in the literature, their functionality and integration into a design flow has not been demonstrated till now. In this thesis, the utilization of the PE is important to speed up the operation of the Sigma-Delta ADC synthesis tool by allowing evaluation of design alternatives without having to synthesize each circuit. Thus, infeasible specifications can be eliminated straight away and other alternatives can be evaluated much faster and the best architectures at the system level can be obtained.

The PE works on a pre-defined circuit topology, taking in some performance parameters as variables, and estimating the remaining parameters. One can denote P_i as a performance criterion of any analog circuit such as gain, bandwidth, slew rate, output resistance, etc. and I_i as an independent circuit parameter determining the performance such as the W or L of the MOSFET, etc. Performance of the circuit can be defined by (5.1) where f_i are the functions linking performance parameters to the independent variables.

$$\begin{aligned} P1 &= f1(I1, I2, ...Ij), \\ P2 &= f2(I1, I2, ...Ij), \\ &\vdots \\ Pi &= fi(I1, I2, ...Ij). \end{aligned} \tag{5.1}$$

The problem is to find an expression or at least an estimate for Pi in terms of each other without having to calculate I_j [25]. In general, the functions f_i are non-linear and most of the time implicit. From this definition, the difficulty of the problem is obvious. One solution would be to form a look-up table for all combinations of I_j once

and the required P_i must be searched from the table every time.

The EKV MOSFET model is a fully analytical model dedicated to the design and analysis of low-voltage, low current analog circuits [116]. The EKV model provides the advantage of continuity for all large and small signal variables in all regions of transistor operation including weak inversion, moderate inversion and strong inversion. Using classical BSIM models, one is faced with the disadvantage of three equations for three different operating regions and discontinuities as well as the sheer size of the model.

In the EKV model, I_D is derived using the charge sheet model described in [116].

$$I_D = 2N\mu C_{ox} Ut^2 \left[ln^2 \left(1 + e^{\left(\frac{V_P - V_S}{2Ut}\right)} \right) - ln^2 \left(1 + e^{\left(\frac{V_P - V_D}{2Ut}\right)} \right) \right] \tag{5.2}$$

IC is the inversion coefficient given by I_D/I_S where I_S is the normalization current. Large values of IC indicate strong inversion, whereas small values correspond to weak inversion. Using IC, design parameters such as output conductance, bias voltages, width of the transistor, current etc can be derived. The EKV model provides a simple expression for the g_m/I_D ratio of the MOS transistor which is precise and continuous throughout all the regimes. Experimental results show that g_m/I_D ratio is valid for all MOSFETs in a wide range of technologies [117].

The 'divide and conquer' method can be utilized for performance estimation. Modeling each analog block with its simple equations analytically, it is easier to translate the input specifications of the main analog system to its blocks so that the problem will be partitioned to its lowest parts. For example, a typical basic two stage (BTS) OPAMP is composed of four analog building blocks; namely, differential input pair, current mirror, common source driver and a current source. Shifting the focus to one higher level, only two blocks are present: a differential input pair and a push-pull output stage. The methodology when designing a large circuit will be to start estimation from the last block and move backwards until the first one. Our PE models several versions of common source gain stages, current mirrors, differential pairs, source followers, and cascode structures. Based on these blocks, the performances of many typical

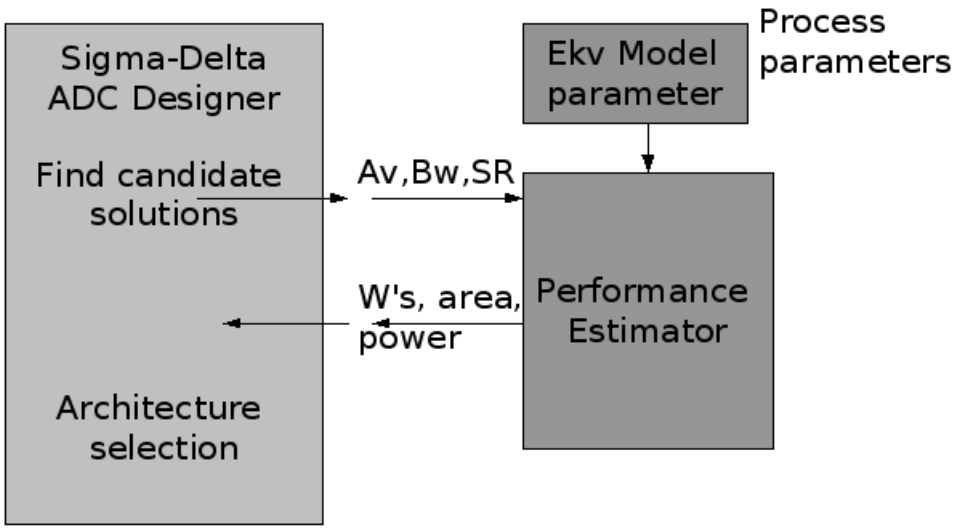

Figure 5.1. Communication diagram of blocks

circuits can be estimated. By use of the manually derived equations and estimation recipe, the PE takes in performance specifications from ADC designer like SR, gain BW, etc. and provides the power and area requirements of the best design fulfilling those specifications (Figure 5.1). In other words, PE generates pareto-optimal curves for candidate solutions. Sample PE design surfaces for a sample OPAMP are depicted in Figure 5.2.

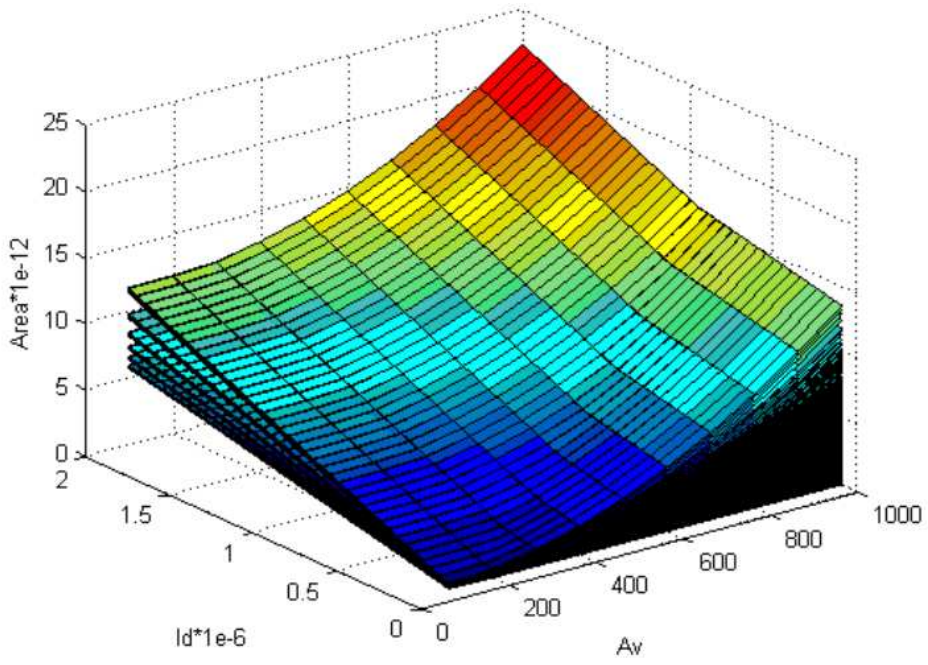

Figure 5.2. Design surface defined by PE for BTS Opamp

The PE may use any technology defined by a technology file which contains many EKV parameters. BSIM models are much more popular than EKV models, hence, it is sometimes difficult to find EKV models of every technology. This bottleneck of the EKV based designs may be overcome by some approaches available in the literature [118],[119]. If the foundry does not provide the EKV models, they can be derived from various measurements of fabricated transistors. However, this approach is not suitable most of the time, since it is time and budget consuming. A more practical way is to use BSIM models to derive EKV models [118]. BSIM models are known to provide accurate results under certain conditions. For these conditions, certain transistor sizes and inversion types for transistors should be selected. If these conditions can be created for certain simulations, the results may be utilized like real measurement results. However, it is clear that the results will not be as accurate as foundry provided parameters. The important point to note in this respect is that, the tolerance of PE is large enough to accept these variations. Since, the PE is not a synthesizer but an "estimator", the error coming from the model can be tolerated.

A parameter extraction approach can be carried out similar to [118, 119]. In this approach the BSIM models are used to find EKV model parameters via results achieved from a large number of simulations. For each parameter, specific conditions should be selected. The next step is to perform related measurements from SPICE simulations. This approach provides results in short time but the number of parameters calculated is limited. However, it is sufficient for a rapid calculation of parameters to see their order of magnitudes. Also, it is sufficient to gather parameters which are used in g_m/I_d design methodology. The parameters which can be gathered by this approach are: VTO, GAMMA, PHI, LETA, WETA, Q0, LK, KP, THETA, DL, DW, Q0 and LK. Also it should be stated that, some parameters can be taken from BSIM parameters directly like TOX, CJ and XJ.

However, for more precise results for EKV simulations, all parameters should be found. In [118], curve fitting approach as well as least square optimization algorithms are used for gathering EKV parameters. Thus, more parameters with better accuracy can found. These parameters include, intrinsic model parameters, temperature pa-

rameters, noise parameters, overlap capacitances and junction capacitance parameters which are sufficient for almost all types of circuits. In Figure 5.3 and Figure 5.4, it can be observed that BSIM simulation results are very similar to similation results by using EKV model parameters gathered from BSIM parameters. The small deviation from measurement results may be observed in Figure 5.5. This small deviation proves the accuracy of EKV models gathered from BSIM models.

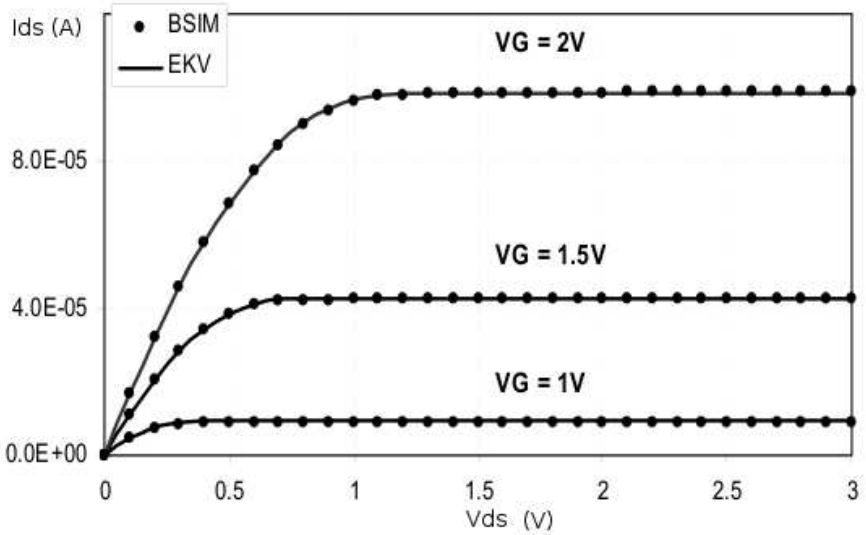

Figure 5.3. I_{DS} vs V_{DS} graphs of BSIM and EKV simulations for 0.5μm

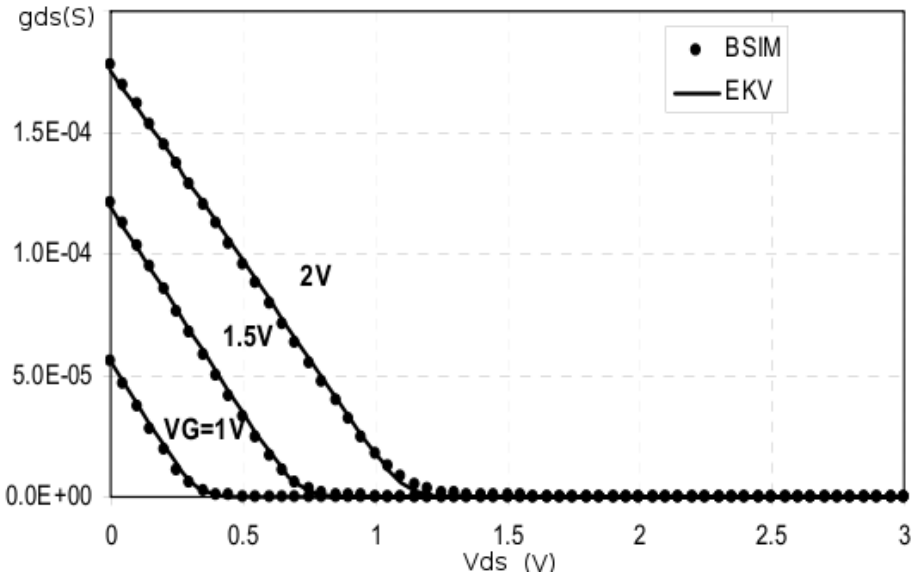

Figure 5.4. g_{DS} vs V_{DS} graphs of BSIM and EKV simulations for 0.5μm

The PE may use any given technology if the technology parameters are provided via a file. This technology file contains 16 parameters. PE may use one technology

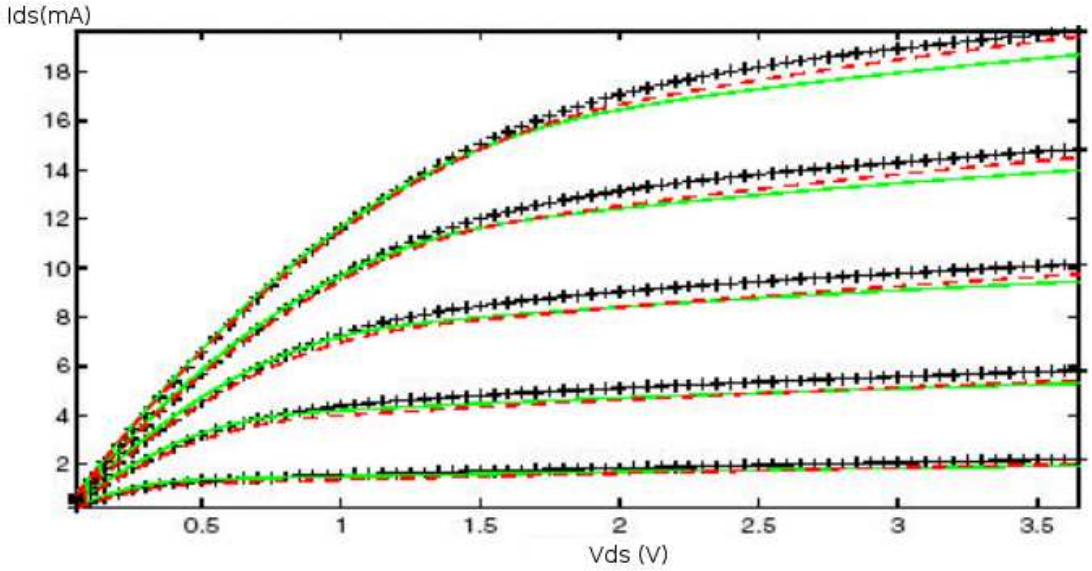

Figure 5.5. I_{DS} vs V_{DS} graphs of measurement results with BSIM and EKV simulations for 0.35μm

file at a time, in other words, user may only search for specific technology and can not search two technologies simultaneously.

The error between estimated values and simulation results is small and thus, results prove that PE provides accurate calculations. However, real technology parameters would present the effectiveness of the developed methodology. Thus, BSIM AMS 0.35μm parameters were converted to EKV parameters and examples are provided in the following sections.

5.1. Basic Two Stage Amplifier

Basic two stage (BTS) amplifier is a popular amplifier structure. The structure may be decomposed to 3 basic sub-structures: current source, differential amplifier and class-A power amplifier. In order to realize PE tool, some common building structures have previously modeled. Hence, these building structures has their own models available and they should be combined together for PE.

BTS amplifier has a power amplifier output stage with a current source. One

of the design specification for BTS amplifier is to have a large swing at the output which means that the output should be fixed at the half of the supply voltage. This specification could be achieved with sizing of the transistors.

The gain of the BTS amplifier can be divided into two parts which represent the gain contribution of output stage and differential amplifier. The percentage of the gain contribution may be calculated heuristically or by sweeping the possibilities. Either way, the contribution of the output stage should be calculated. The inversion coefficient can be swept in order to calculate the related gain value. However, the current flowing in the output stage should be calculated previously. This value is calculated from slew-rate requirement. This calculation provides the minimum value of the current. Higher values may also be possible depending on the other specifications met. Hence, current with sizing information provide partial area and power consumption values. The bandwidth provided to the BTS module provides information about the dominant pole which is related with compensation capacitance. This capacitance can be used to calculate the required current in order to satisfy the bandwith requirements. Also gain can be calculated with the current value and associated inversion coefficients. This coefficient can be swept in order to fulfil the gain requirements.

The most important calculation in the process is the calculation of the g_{ds}. This calculation generally very problematic since the deviation in the calculations is very high. Utilization of the EKV models introduce an advantage at this point. The g_{ds} values can be calculated via derivation of current equation. Also another possibility to use a pseudo Early Voltage in order to calculate a g_{ds} term as a function of transistor length. Both methods were incorporated in the developed tool. In the regions where derived equation does not hold due to simplifications, the other method is utilized.

Another important process is the verification of the stability in the presence of the second pole. This verification performed for all candidate solutions.

As a result, power consumption can be estimated by current calculations and together with inversion coefficient value area of the amplifier can be calculated.

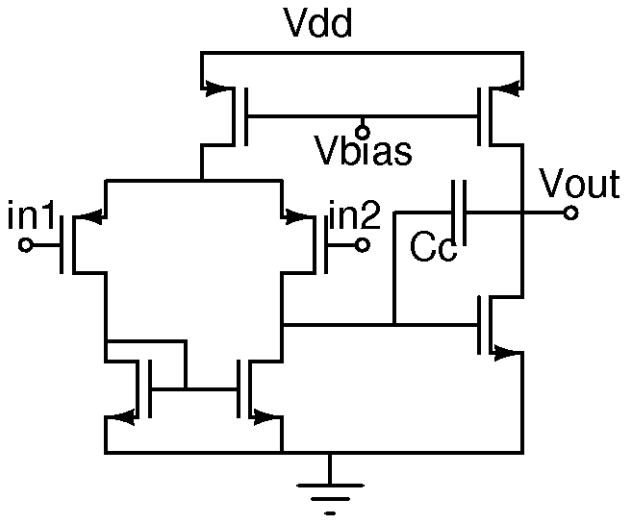

Figure 5.6. Basic two stage OPAMP

5.2. Cascode Amplifier

Cascode amplifier is a single stage amplifier and contains many difficulties due to stacking of transistors. The calculation of DC biasing values is also very difficult and small changes in widths of the transistors may lead to unwanted operation regions. The cascode module developed is capable of calculating the correct bias voltages and sizing in order to achieve the desired performance. The DC value of the output node can be set to half of the supply voltage. In cascode amplifier design the design equations are simple since the structure is single stage. The structure is composed of a current source and cascode structures. Hence for each stack, the equations are nearly same.

A similar approach is used in cascode amplifier. The output stage current is calculated by slew-rate requirement and inversion coefficients are swept for finding the optimal solution. Some overdrive values for transistors may be defined in terms of inversion coefficients and current mirrors decrease the number of inversion coefficient to be swept. Hence, the result is achieved in a very short time.

5.3. Amplifier with Custom Model

There may be other amplifier structures that are available and the user may wish to utilize this available circuit in ADC design. In such cases, the tool should be

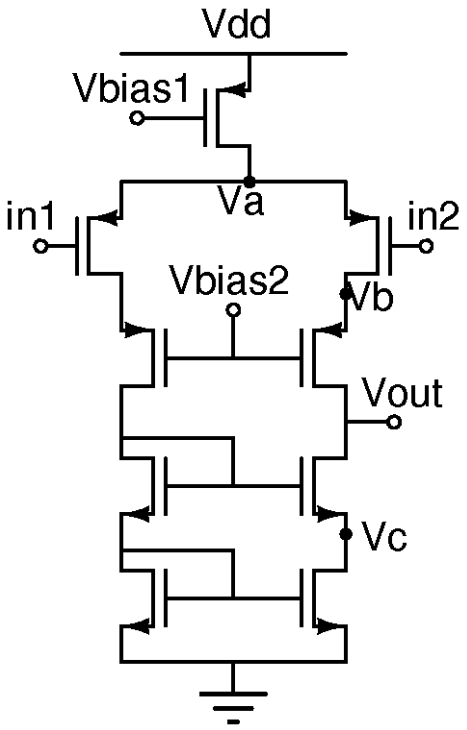

Figure 5.7. Cascode OPAMP

capable of utilizing these structures. The approach used for this purpose is to develop a customizable module which links output parameters (area, power) with the input parameters(gain, bandwith, load capacitance and slew-rate) Hence, any previously designed circuit may be mapped to a model that can be used by PE. This approach provides flexibility to user by mapping the input parameters to output parameters where polynomial equations were used to define the relations. Hence, better mapping is possible by this approach. Although, area and power parameters may be defined by simpler equations, this approach may provide better mapping.

$$\begin{aligned} power \;=\; & a_0 + (a_1\,A_v + a_2\,A_v^2 + a_3\,A_v^3)\,(a_4\,A_v + b_4\,Cl + c_4\,f_{3db} + d_4\,SR)^t \\ & +(b_1\,Cl + b_2\,Cl^2 + b_3\,Cl^3)\,(a_5\,A_v + b_5\,Cl + c_5\,f_{3db} + d_5\,SR)^x \\ & +(c_1\,f_{3db} + c_2\,f_{3db}^2 + c_3\,f_{3db}^3)\,(a_6\,A_v + b_6\,Cl + c_6\,f_{3db} + d_6\,SR)^y \\ & +(d_1\,SR + d_2\,SR^2 + d_3\,SR^3)\,(a_7\,A_v + b_7\,Cl + c_7\,f_{3db} + d_7\,SR)^z \end{aligned} \tag{5.3}$$

$$
\begin{aligned}
area \;=\; & e_0 + (e_1\,A_v + e_2\,A_v^2 + e_3\,A_v^3)\,(e_4\,A_v + f_4\,Cl + g_4\,f_{3db} + h_4\,SR)^t \\
& +(f_1\,Cl + f_2\,Cl^2 + f_3\,Cl^3)\,(e_5\,A_v + f_5\,Cl + g_5\,f_{3db} + h_5\,SR)^x \\
& +(g_1\,f_{3db} + g_2\,f_{3db}^2 + g_3\,f_{3db}^3)\,(e_6\,A_v + f_6\,Cl + g_6\,f_{3db} + h_6\,SR)^y \\
& +(h_1\,SR + h_2\,SR^2 + h_3\,SR^3)\,(e_7\,A_v + f_7\,Cl + g_7\,f_{3db} + h_7\,SR)^z
\end{aligned}
\tag{5.4}
$$

In (5.3) and (5.4), the polynomials are given which define the relation of area and power with input parameters. These equations are sufficient to produce custom design surfaces, similar to the surface illustrated in Figure 5.2.

Area and power may also be related with input parameters by their electrical relation such as relating power with slew-rate parameter by using appropriate coefficients. Although the actual relation would require the current rate information, the slew-rate with a user defined coefficient would be sufficient to define the power of the amplifier with input parameters. Hence, this simple definition may be easier to develop a mapping between circuit definition that are readily available and the area and power consumptions. The given equations in (5.3) and (5.4) allow user to define a propotion between area and GBW or a relation between power and $\frac{SR}{Cl}$.

It should be noted that, if user has only very few amplifiers, providing their area and power values directly to the ADC automation tool will be easier than deriving the relations. However, if an amplifier is not designed but rather the design performance for a specific technology can be roughly defined by means of simple equations, then the custom amplifier model would be utilized to find the design parameters.

5.4. Example Performance Estimator Designs

In order to test the PE, two technologies were used. The first one is the 0.5μm technology which is delivered with EKV 2.6 [120].

The below examples present results for two PE blocks: basic two stage (BTS) OPAMP and cascode OPAMP. PE provides area and power consumption estimations of a candidate solution. While PE is trying to achieve performance values such as gain

and bandwidth, it also calculates DC voltage and current values. Although in this work these results are not feed into the device level optimization tools, they can be used by device level tools as initial values in optimization.

Table 5.1. Design specifications

Gain	2000
Load Capacitance	1pF
Slew-Rate	20V/μs
Cut-off Frequency	10kHz
Supply Voltage	3.3V
Technology	Ekv 0.5μm

Table 5.2 presents performance values of two candidate solutions for a BTS OPAMP in 0.5 μm technology. The specifications are given in Table 5.1. Although these requirements may not be stringent, they are appropriate for many Sigma-Delta ADC designs. The BTS PE module, generated 29645 solution candidates and provided 125 solutions for the given specifications. The generated solutions were examined by Hspice simulations and their accuracy was compared with the estimated values. These parameters were presented in Table 5.2. Another parameter is the DC value of the output node, which was set to 1.65V. Simulation results show that the variation in this value is bounded by $\pm$ 50mV.

The specifications in Table 5.1 were provided to BTS module which uses 0.35μm AMS technology data. The results show similar accuracy with the previous example as given in Table 5.3.

Other structures may be modeled and added to PE database. Another common structure for OPAMP is the cascode OPAMP. The main complexity of the cascode design is to find bias voltage points and to size the transistors appropriately such that the output node has the desired DC value. Although most of these systems require common-mode feedback structure to ensure the correct operation, such sizing of transistors will ease the addition of common mode feedback elements.

Table 5.2. BTS OPAMP dynamic performance results for 0.5μm

		Solution	7059		Solution	6749
	Calc.	Sim.	Err.	Calc.	Sim.	Err.
Av	68.0 dB	67.3 dB	1.3%	66.3 dB	66.5	0.3%
Gm1	121μS	128μS	5.5%	40.9μS	38.4μS	6.5%
Gm6	44.1μS	42.3μS	4.3%	109μS	109μS	0%
I1	7.2μA	6.7μA	7.4%	7.2μA	6.69μA	7.6%
I6	2.6μA	2.1μA	23%	2.7μA	2.22μA	21%
gds1	720nS	827nS	13%	720nS	752nS	4.2%
gds2	1.66μS	1.59μS	4.4%	1.66μS	1.46μS	13.7%
gds4	270nS	310nS	13%	270nS	292nS	7.5%
gds6	624nS	586nS	6.5%	624nS	553nS	12.8%
f1	9.4kHz	9.02kHz	4.2%	10.4kHz	10.7kHz	2.8%
f2	38MHz	30.5MHz	24%	34.7MHz	30MHz	15.6%

Table 5.3. BTS OPAMP dynamic performance results for 0.35μm

		Solution	38347		Solution	45172
	Calc.	Sim.	Err.	Calc.	Sim.	Err.
Av1	72.5 dB	72.8 dB	0.4%	69.1 dB	69.3 dB	0.3%
Gm1	108μS	112μS	3.7%	193μS	188μS	2.6%
Gm6	93.1μS	81.5μS	12.4%	143μS	145μS	1.4%
I1	9.5μA	9.9μA	4.2%	16.5μA	16.1μA	2.5%
I6	4.14μA	4.4μA	6.2%	11.0μA	11.2μA	1.8%
gds1	458nS	452nS	1.3%	871nS	833nS	4.6%
gds2	1.05μS	1.25μS	19.0%	2.65μS	1.9μS	39%
gds4	256nS	269nS	5.0%	847nS	723nS	17.1%
gds6	757nS	953nS	25.8%	1.9μS	1.41μS	34%
f1	8.8 kHz	10.2 kHz	15.9%	10 kHz	8.7 kHz	15%
f2	32MHz	38 MHz	18%	54 MHz	44 MHz	22.7%

Table 5.4 presents the DC values calculated by the estimator and the simulation results performed by Hspice.

Table 5.4. Cascode OPAMP DC performance results for 0.5μm

	Solution	404	Solution	428
	Calc.	Sim.	Calc.	Sim.
Va	2.80	2.85	2.80	2.86
Vb	1.74	1.76	1.74	1.80
Vout	1.60	1.64	1.60	1.66
Vc	0.80.	0.82	0.80	0.83
Id	3μA	3.1μA	3μA	2.9μA

Similar accuracy for DC values were achieved as a result of simulations for 0.35μm technology. Hence, they are not presented here in details.

The results prove the accuracy of the PE in DC calculations. In these examples, the calculated values are same for different candidate solutions The difference is the sizes of transistors as well as bias voltages provide to the circuitry. These voltages vary form 0.8V to 2.4V for *Vbias2* and 2.5 to 2.9 for Vbias1. The bias voltage values are actually the outputs of the PE as a result of IC value sweep of transistors. Thus, PE may find solutions with different values of IC for which most of the designers would not look for. Hence, unusual voltages may be calculated as bias voltages. However, simulation results show that these solutions still satisfy the given specifications.

Table 5.5 and Table 5.6 presents the dynamic parameters of the cascode structure. These results are achieved for specifications given in Table 5.1 with a difference that the cutoff frequency is not set. Although some parameters deviate from the desired values, they stay on the safe side.

PE modules developed provide enough accuracy for the ADC designer. They may be even used as initial conditions for device level optimizers. The results provided show that the ADC designer may search the design space efficiently. The discussion

Table 5.5. Cascode OPAMP dynamic performance results for 0.5μm

	Solution	156		172		43
	Calc.	Sim.	Calc.	Sim.	Calc.	Sim.
Av (dB)	74.6	74.9	74.9	74.8	78.4	80.2
f3db	1.05 kHz	981 Hz	1.00 kHz	1.11 kHz	347 Hz	426 Hz

Table 5.6. Cascode OPAMP dynamic performance results for 0.35μm

	Solution	77		598		1115
	Calc.	Sim.	Calc.	Sim.	Calc.	Sim.
Av (dB)	73.8	72.6	76.3	77.3	75.6	78.6
f3db	1.3 kHz	1.7 kHz	428 Hz	350 Hz	896 Hz	760 Hz

so for has covered only modules. ADC designer examples that utilize PE modules are presented in next section.

6. EXAMPLE RUNS OF DESIGN AUTOMATION TOOL

In this section, different example runs are provided in order to illustrate how the developed design automation tool can be used by a designer. The presented runs differ from each other not only by means of desired specifications but also with different operation modes and design perspectives for a design. The simplest utilization of the developed design automation system is to use verification mode, by which the system performance can be observed for a specific design where all design parameters have been previously defined. The designers may require their design to be verified by calculating the performance of the ADC, which is SNR of the system, most of the time. Another interesting utilization of this mode is that the designer may check the increase of the performance with modifications to the block parameters. As an example to this approach, the designer may use the design automation tool to calculate the performance deviation if GBW of the amplifiers are a little bit higher or lower. An adequate run can be achieved by selecting a second order system with high SNR. The specifications given in [7] were used and given to the design automation system. The calculated performance was 100.1dB which is very accurate compared with the given measurement result, 100.2 dB and simulation result 98.3 dB. The user may explore other solutions by modifying the provided parameters to the design automation tool. The effect of using a 3-bit quantizer can be observed by this way which would provide better performance regarding the SNR. In such example design, the performance increase would be 13.4dB for a perfect DAC circuit which has a very low INL value. The increase in INL of the DAC circuit to 10 percent of the full scale would reduce the performance increase to only 6.7dB. Also, designer may use this verification mode to examine the sensitivity of the design to some circuit parameters. Hence, the user may observe the effects of using an amplifier with lower gain values. If the amplifier is designed initially with a gain higher than 60dB, the gain reduction can be tolerated up to %60 of its original value, since the performance loss even with this reduction is negligible.

The verification mode provides all the information that the designer can gather from a behavioral simulation. The advantage of the devaloped tool can be observed

better with the utilization of other modes of operation. The Sigma-Delta ADC design tool also utilizes the PE depending on the selected operation mode, in order to find adequate solutions for given specifications. The given specifications may be in different forms depending on the desired operation mode. In the second example, the semi-custom mode of the Sigma-Delta ADC design tool was chosen because it contains both library oriented design and automated design. The amplifiers were designed automatically by utilizing the PE and the comparators were chosen from the library. The amplifier types utilized by the PE throughout this example are BTS and cascode OPAMPs. In the design example, the desired specifications are a SNDR of 86 dB, a supply voltage of 5V, and an OSR of 64. The order of the modulator was not set. The desired SNDR value corresponds to nearly 14 bits of resolution. The solutions were restricted to be single bit.

The algorithm searches for candidate architectures with the given specifications. Since behavioral simulation was not performed for checking the stability of the design, the tool advised the user to use cascaded architectures, which utilize at most second order modulators. As a result the candidate architectures were 2-1, 2-2 and 2-2-1. However, lower order architectures can be achieved by assigning the OSR as a free variable. In this case, the algorithm searches for OSR values which provide adequate performance. This search is performed on ideal Sigma-Delta ADC equations and the real performance is calculated later by adding noise contributions.

The candidate architectures which satisfy the desired SNDR performance for OSR of 64 were sent to the PE and area and power values were estimated. For this example, it was assumed that only BTS OPAMP's were available. Comparison of BTS module with cascode is presented in next example.

For the semi-custom operation mode that is used in this example, worms were created for 2-1, 2-2 and 2-2-1 configurations. The amplifier library was not used in the example, thus the comparator library defines the number of initial worms. The user may select the range of parameters for the amplifier and other blocks in the design automation tool. For our run, ranges for the gain of the amplifier and capacitor values

in the integrator were set.

The worms were sent to noise estimation functions who calculate the SNDR of the worm. As a result, the user has many solutions which satisfy the specified SNDR performance. The algorithm generates 846 candidate solutions. Performance estimator could find a solution for 567 of them. Thus, 67% of the candidate solutions were implementable results. The solutions are listed according to their SNDR values by the software. Several solutions may exist for the same SNDR with different structures in which case the optimum solution (in terms of area, power, or a combination of these two) should be selected by the user.

In another example run, both cascode and BTS modules were used in PE. Thus, the selected architectures may contain different types of amplifiers for different stages. The results observed show that topologies with higher gains are selected for first integrator and for the other amplifiers lower gain solutions are selected. Thus, power consumption for the first stage is higher than the following ones. Slew-rate limitation is very important in amplifier selection. Since, the current charging the load capacitor is calculated via minimum slew-rate, the power dissipation is directly proportional with the current. Hence, the selected amplifiers most of the time have large gain values although ADC designer is looking for much less. The main reason is that the gain is actually related with the length of the transistors and these modules do not sweep the length of the transistor which may provide solutions with lower gain values. For example, ADC designer is searching for an amplifier with a gain of 2000 and slew-rate of $20V/\mu s$. PE searches for a candidate solution and the solutions generated have at least gain of around 5000. The solutions of a behavioral optimization approach presented in [34] have small gain values but large or medium slew-rate values. Hence, solutions for system level specifications were achieved. However, the block level requirements for these solutions were actually not connected to any device level block and their implementation as such is not simple. However, in our work, even W/L ratios as well as bias voltage values are estimated. Also, power and area values are calculated more

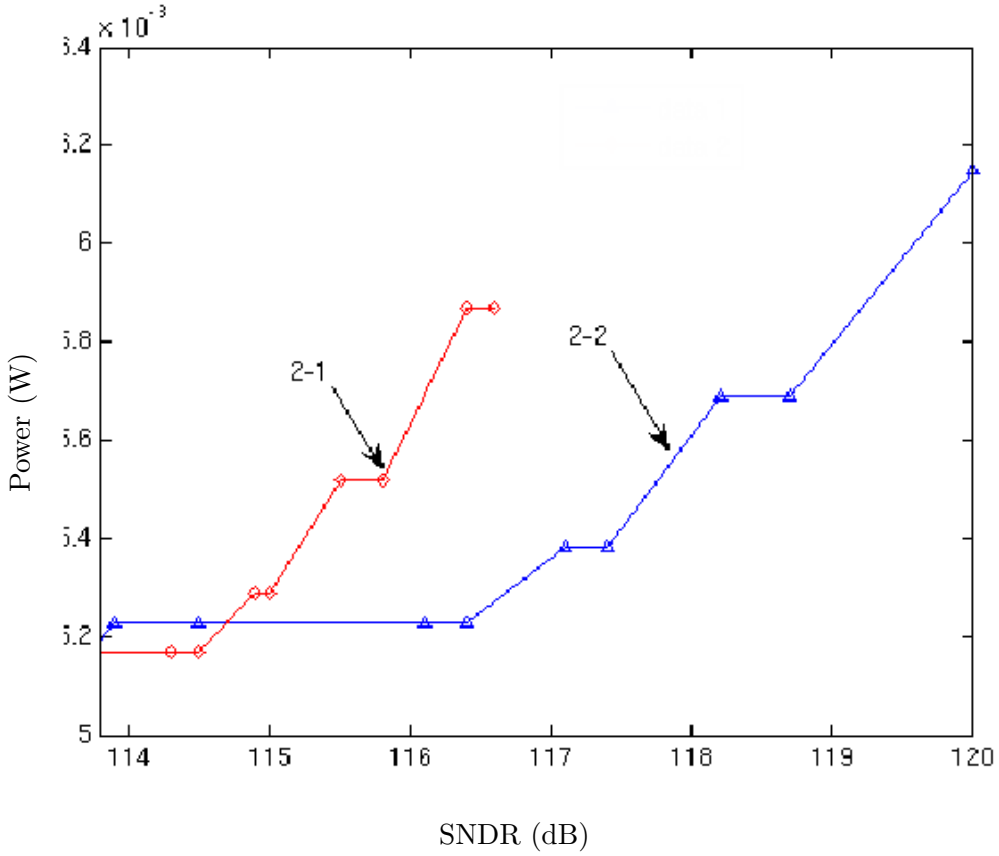

Figure 6.1. SNDR vs power

accurately.

In Figure 6.1, SNDR vs. power and area results for some cascode solutions are presented. In these solutions the OSR is 128. Here, the PE only uses cascode OPAMP structure. It is also interesting to observe from this figure that 2-1 structure can achieve the same SNDR specs with lower power as opposed to the 2-2 structure for low SNDR. For more demanding SNDR values, the OPAMP is pushed to its limits in the 2-1 structure and power increases sharply. In addition, the solutions by utilizing BTS architectures are more advantages in both power and area values.

The developed tool uses PE modules to calculate SNDR value and estimate its power and area. Since power consumption values can be estimated accurately, pareto-optimal curves can be generated. In Figure 6.2, a SNDR vs power curve is presented as an example. In this example, OSR is limited between 20 and 64. BTS and cascode

modules were enabled and SNDR values calculated for different configurations. Lowest power consumption value was achieved for 2-1 cascaded configuration with OSR value of 20. The tool selects cascode OPAMP for first stage and BTS for the rest. The highest SNDR value is achieved for 2-1-1 cascaded configuration.

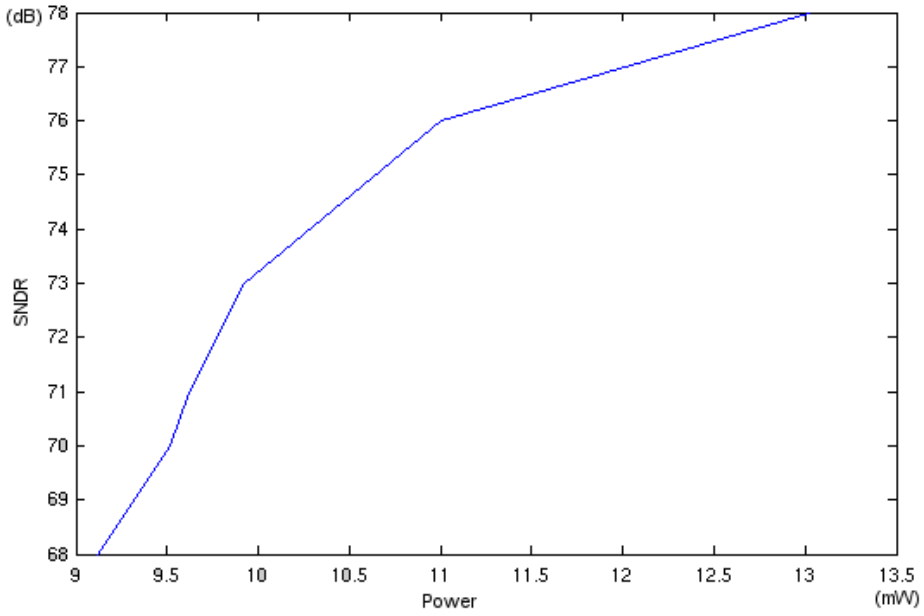

Figure 6.2. SNDR vs power

Table 6 is another example which illustrates that similar performances can be achieved with different configurations. In such examples power estimation has greater importance on configuration selection. Hence, macro models may not be accurate enough.

Table 6.1. Different cfigurations with similar performances

SNDR	Configuration	OSR	Power	Area of active devices
73	2-1	32	9.92mW	$1115\mu m^2$
73	2-2	20	9.99mW	$1540\mu m^2$
78	2-1	64	10.32mW	$1745\mu m^2$
78	2-1-1	32	13.02mW	$2060\mu m^2$

The developed tool also explores possible multibit solutions. In Table 6, the comparison of two solutions which have similar performances is given. The increased

performance with the utilization of mulbit quantizers, observed in Table 6 is enough for making them eligible for GSM applications.

Table 6.2. Different cfigurations with similar performances

SNDR	Configuration	Bit	OSR
81	2-1	3	32
81	2-2	3	20
78	2-1	1	64
78	2-1-1	1	32

The examples show that the developed tool may be used to generate pareto-optimal curves for desired specifications. The design space from which these curves are generated has many dimensions and the PE connects performance values to real circuit parameters. Hence, not only optimization through SNDR vs power curve is possible, but also area can be incorporated to select the most appropriate configuration.

7. INTEGRATED CIRCUIT DESIGN

One of the milestones in this thesis is to design a Sigma-Delta ADC that can be used to verify the models developed through out the thesis. The fabrication of Sigma-Delta ADC test chip was made with the grant of TUBITAK, project EEEAG-101E039. The design process gave some idea about the effects of error sources and allowed the models to be used through the design process. The models were used to calculate the tolerances of blocks that will be used in the chip. The amplifier gain, amplifier BW, the offset of the flash adc, effects of on resistances of switches are some of the parameters that are calculated by utilizing the developed models. Then, this values lead us in the process of block (amplifier,comparator) design. The mismatch values that are available in the technology data were used with the tolerable ofset value to calculate the sizes of the transistors in the comparator.

The designed ADC contains two parts as it can be seen in Figure 7.1. One is a third order, 2-1 cascaded Sigma-Delta ADC. The other one is first order single bit Sigma-Delta ADC. The first module has two outputs. One output is taken at the end of the second order modulator and the other one is taken from the output of the first order modulator. The connection scheme of the blocks for 2-1 cascaded system can be different from Figure 7.1. The first input of the first module, then, will be the quantization noise of the second order system. The difference between two different configurations results in different digital filter function. However, even though the connection is not the same, the same performance can be achieved, at least theoretically. However, the implementation of the circuit in silicon given in Figure 7.1 is much simpler compared with the other connection scheme.

The opportunity to take these two outputs separately enables us to observe the performance of both second and third order systems. The other module which is a first order system, can be used to test a first order system. As a summary, these two modules enable us to test first, second, and third order systems. However, the fabricated IC has the analog part only. This means that the output of the IC is just a

bit stream. The outputs must be fed to a digital filter in order to get parallel binary data. An FPGA kit was used for this purpose. Using such a system enables us to test the performances of different type of filters. The digital filters were compiled and loaded to the FPGA.

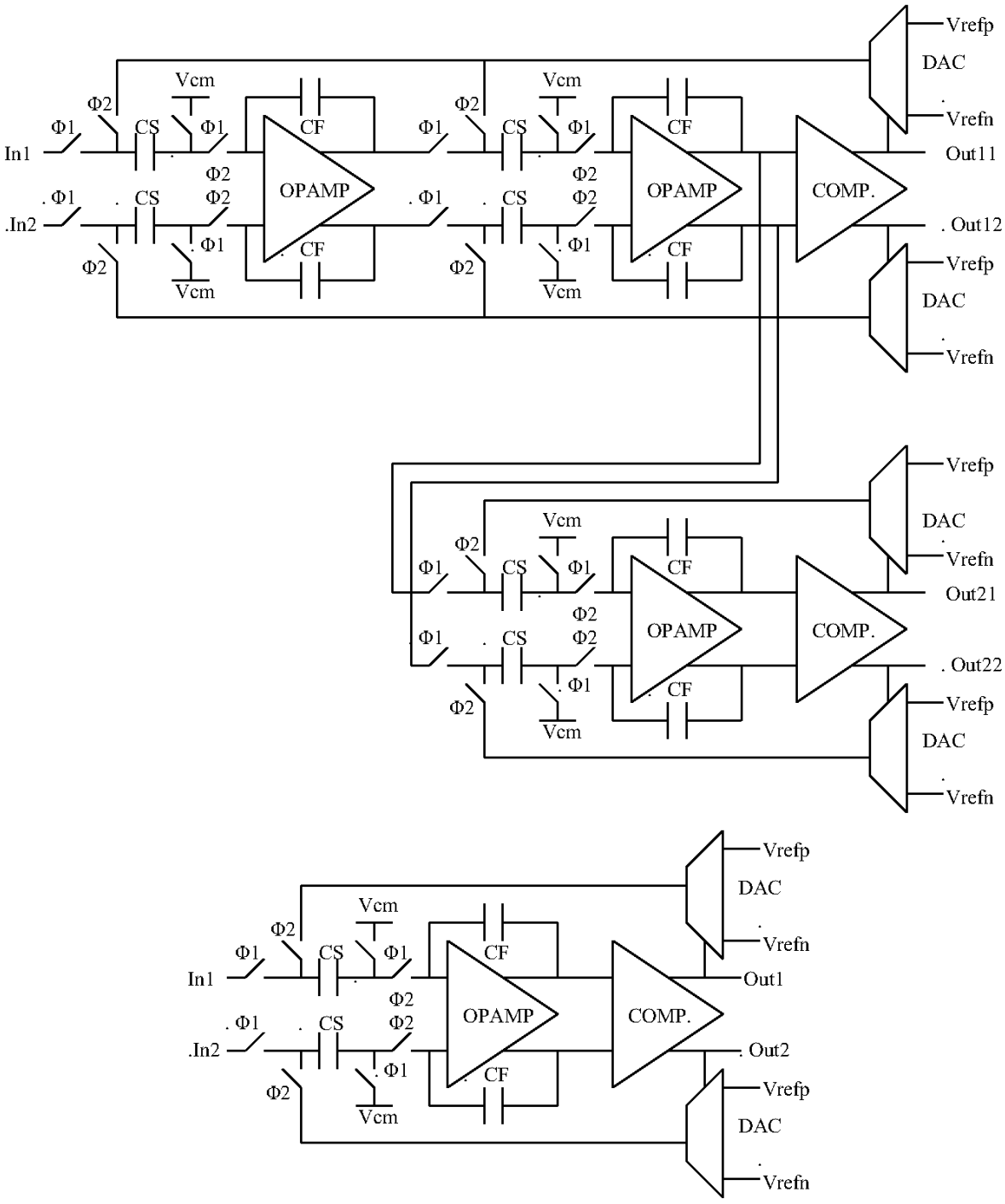

Figure 7.1. Block diagram of the IC containing two modules

The design of the Sigma-Delta ADC started with defining the modules, which had been implemented. In order to have more than one ADC to test the models, the Sigma-Delta architecture described above had been selected.

The next step is to determine the specifications and perform functional verification of the ADC. These can be achieved by using MATLAB models developed earlier. The Simulink model of the Sigma-Delta architecture is given in Figure 7.2. The behavioral simulations performed with the model gave important information. One of them is dynamic range value. The technology allows 5V supply and this value defines the saturation limit of the integrator. The input must be smaller than a limit value, which may result in a saturation situation. Although, the coefficients can be calculated from the models, functional verification can be performed. A coefficient such as 1 can be assigned for all gain stages. However in that case, the charge accumulated in the integrator will be high. In order to overcome this problem coefficients can be made smaller. In our design we select the coefficients as 0.5 to achieve higher input range (-1V to 1V).

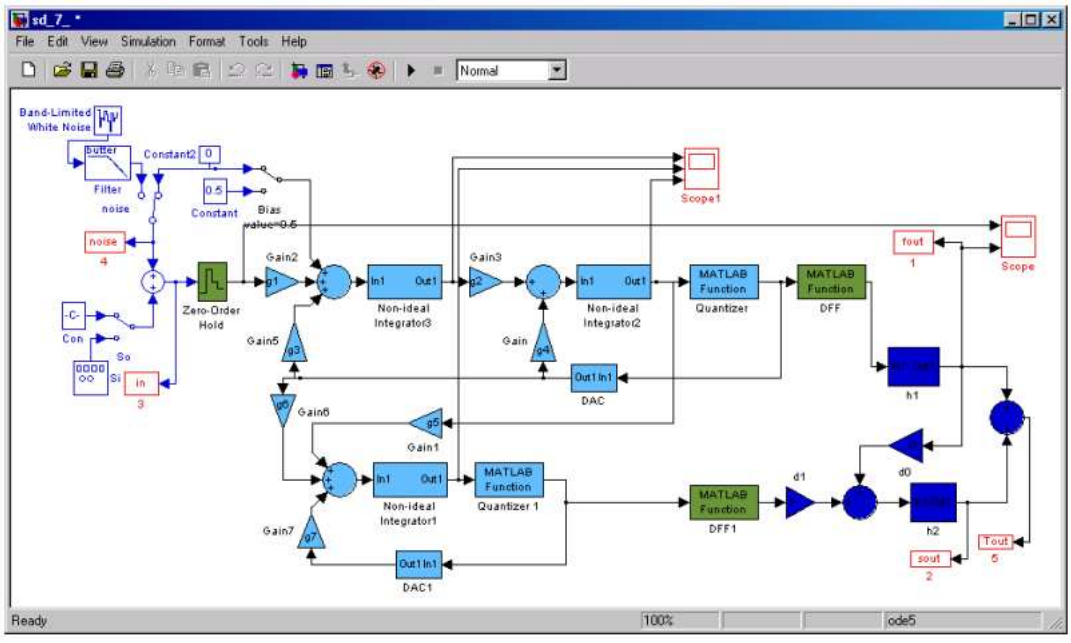

Figure 7.2. Simulink model of the 2-1 cascaded Sigma-Delta

The next step in the design flow is to determine the specifications of the building blocks. The building blocks for such system are switched capacitor integrator and

quantizer. These blocks can easily be implemented by OPAMP's, transmission gates and comparators. In order to estimate the performance of the system and determine these specifications, MATLAB models were used. For example, one of the problems for a switched capacitor integrator is the offset of the OPAMP. The offset is added in each cycle and causes a wrong value of charge at the input of the quantizer. This problem can be solved by using a differential architecture which is the case in our design. However, there may still be some offset, which will still cause problems. The offset value, which can be tolerable, had been observed with the model developed on MATLAB. The designed OPAMP has much smaller offset value than it was required. Other effects, which may cause errors, were also checked with the model. One of them is the capacitor mismatch. The results obtained from the models show that the minimum matching should be higher than 99%. This information was used with the technology data to determine the correct width and length values of the capacitors.

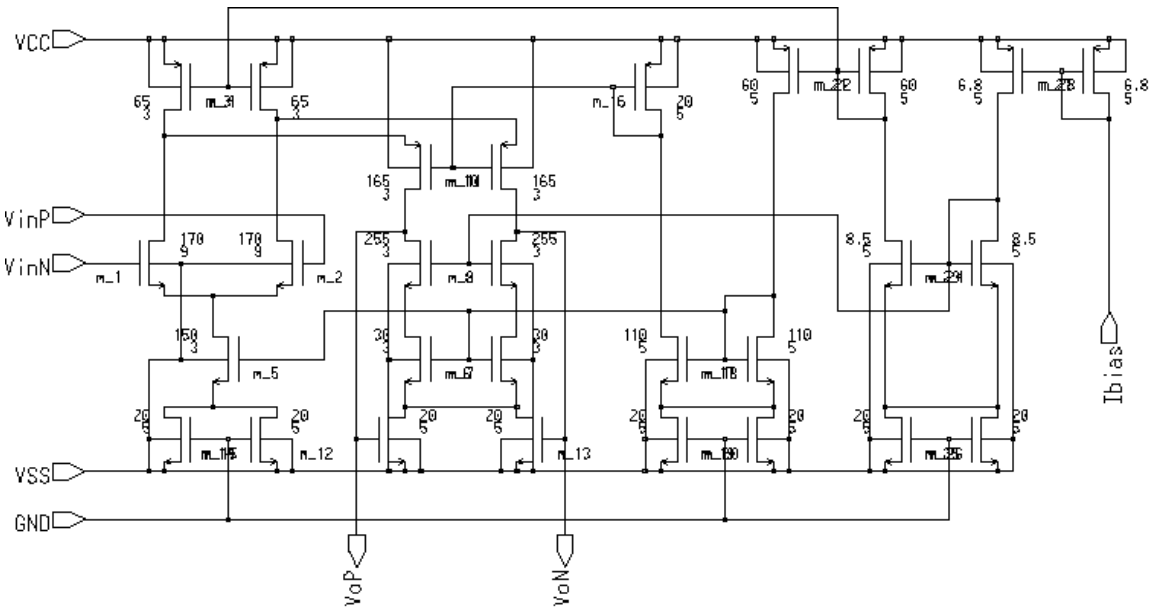

Figure 7.3. OPAMP

The designed OPAMP is given in Figure 7.3. The advantage of this OPAMP is that it only requires one bias current, which also reduces the number of pins. The design is fully differential and the circuit employs common-mode feedback in order to stabilize the output nodes. The designed amplifier provides 68dB gain.

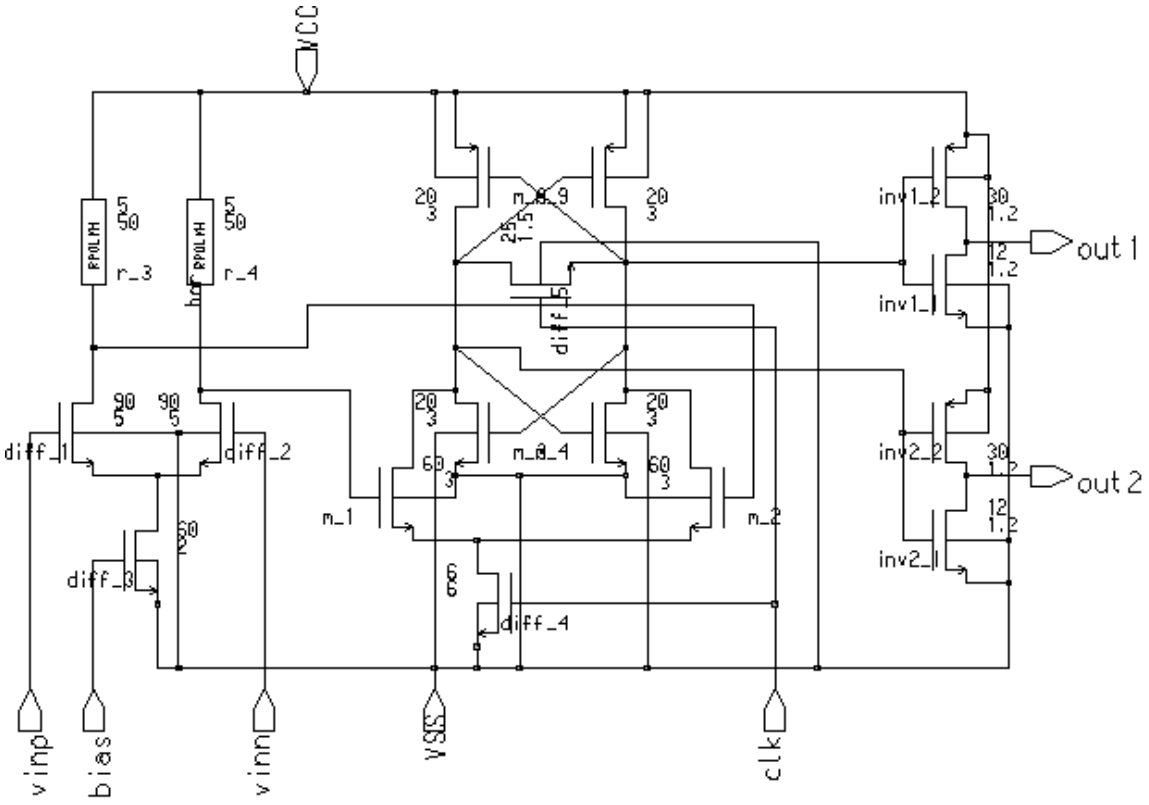

Figure 7.4. Comparator

The circuit given in Figure 7.4 is the comparator circuit used in our ADC design. This circuit had been used in previous flash and pipeline ADC designs [22]. MATLAB models showed us that this comparator used in a Sigma-Delta ADC could tolerate more errors such as offset, which may result from mismatched transistors. Hence, transistor sizes can be smaller and comparator may consume smaller area. There is a preamplifier as an input stage, which also reduces the offset value and kick-back noise.

After having performed behavioral simulations and having designed the building blocks, circuit level simulations were carried out. First of all, blocks were simulated separately. After observing the functional correctness of the block, the blocks were joined together to form the ADC. The following step was to simulate the overall ADC. The output for the sinusoidal input can be observed in Figure 7.5. The output stream must have two levels (-2.5V and 2.5V) and the average value must represent the input for certain bit lengths. In Figure 7.5, we may observe this situation and recognize two high peaks and one low peak. Also these results were compared with the ones obtained

from behavioral simulations performed with MATLAB.

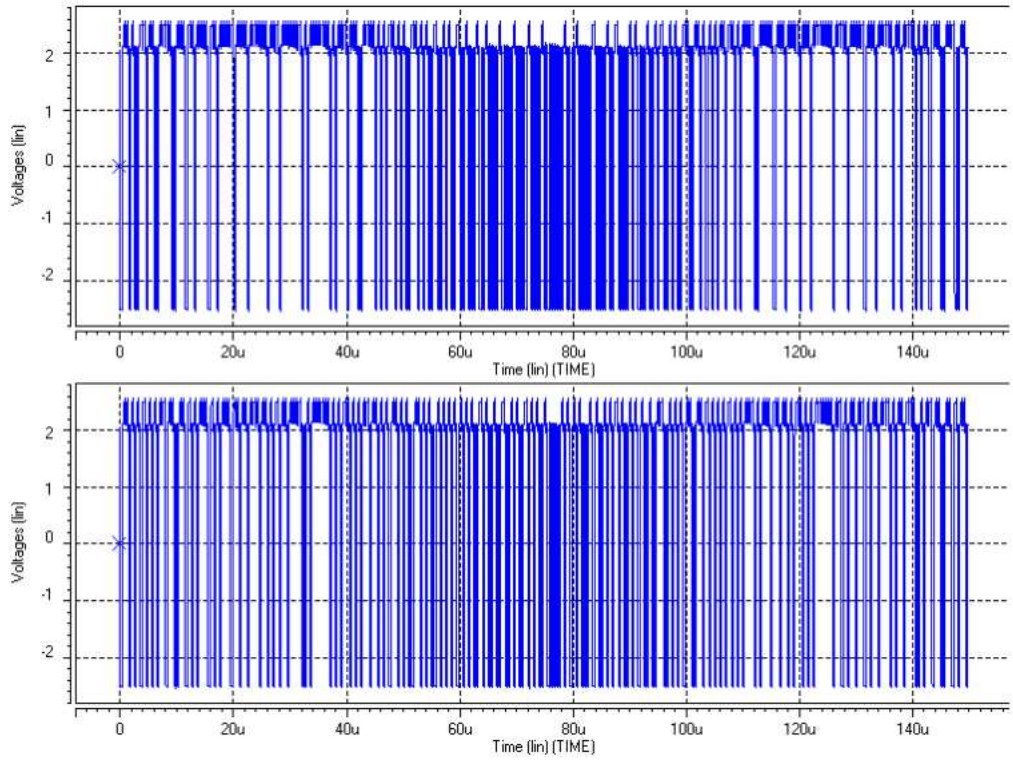

Figure 7.5. Simulation results

The layout of the test IC was drawn carefully considering advanced layout techniques to increase matching of the devices. The symmetry was taken into consideration whereever possible. Dummy circuits and shapes were added also for better matching. Devices were drawn in an interdigitized manner and common centroid capacitor structures were used for capacitors. The layouts of amplifier and capacitor circuits are given in Figure 7.6 and 7.7.

The IC shown in Figure 7.8 was fabricated in Austriamicrosystems 0.6μm technology. The test environment was prepared for testing and a PCB was designed for the fabricated Sigma-Delta ADC chip. There are two biasing and reference voltage generators on the PCB. Also non-overlapping clock generator circuit was put on the PCB.

In addition to the PCB design, the digital filter that should be connected at the

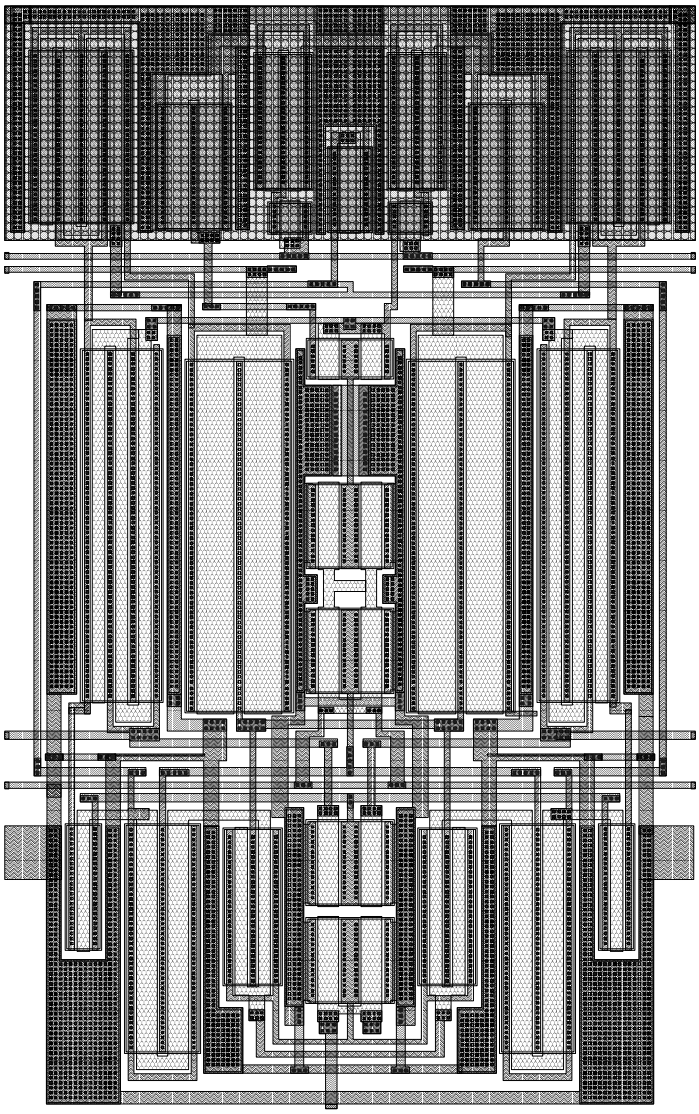

Figure 7.6. Amplifier layout

output was also designed. The filter is a decimation filter and basically converts serial bit arrays to parallel bit streams. The VHDL codes were written for implementing the filter. The filters implemented were basic and triangular decimators. Then these filters were written to a FPGA system for hardware implementation. However, only basic filter can be implemented on an FPGA system. The reason for this is that the triangular filter has coefficients which can be realized with more logic blocks in an FPGA. However, this is not a limitation for testing since the oututs can be acquired by a data acquisition card (DAQ) and transfered to the PC, the output may be fed into many different decimation filters defined in the MATLAB environment. Since the performance of the ADC circuits are measured via PSD of the circuit, decimation filter

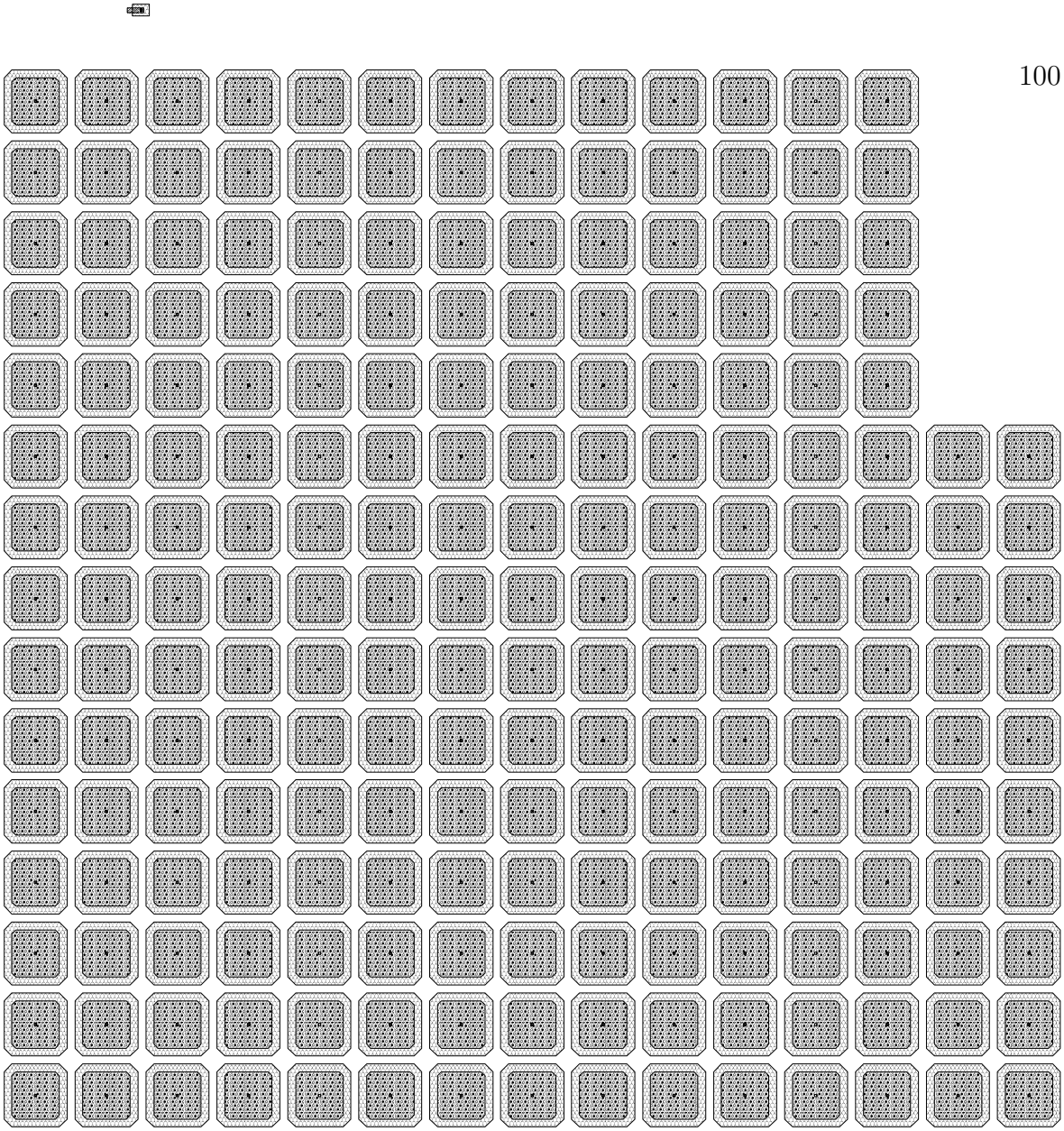

Figure 7.7. 1pF Capacitor layout

is not necessary. However, the advantage of such decimation filter is that since, the output rate is downscaled, it is easier to capture the output data via data aquisition system. Many sinc decimation filters were designed and even their performances were compared [33]. One example simulation is shown in Figure 7.9.

Our aim to realize the digital filter of the Sigma-Delta chip is to process outputs faster and capture them as binary outputs. Thus, the output rate will be decreased and faster capture of the output with our DAQ card can be possible. Without the filter, the output serial bit array can be captured also but the clock rate cannot be increased above a certain level, because it was not possible to synchronize the clock

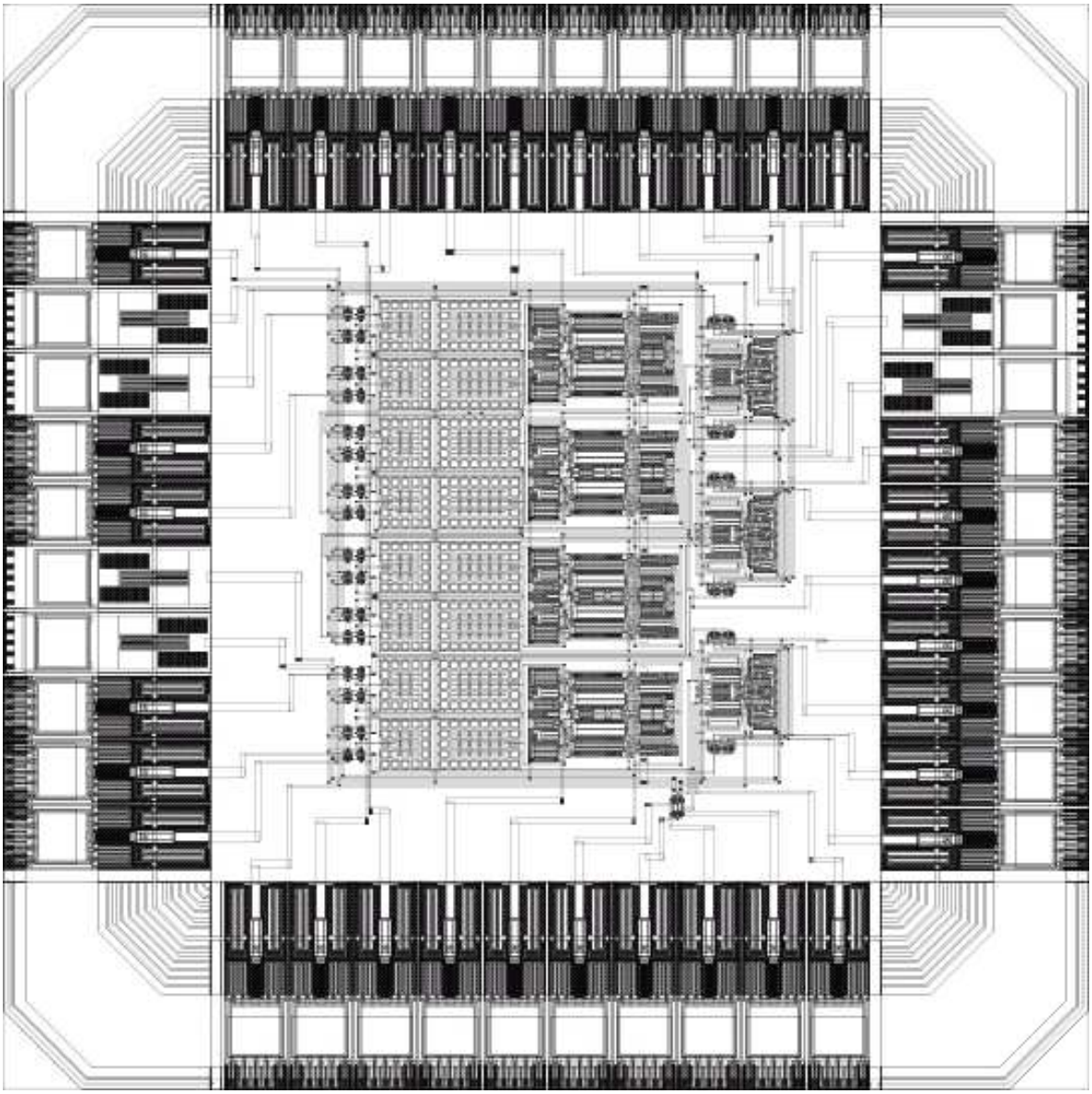

Figure 7.8. Layout of the test IC

signals of Sigma-Delta and DAQ.

The whole test system is given below in the figure. A data acquisition system was used to capture the output of the test chip.

The analog part was tested for verification of the PCB. The buffering problem with the design, which prevents operating in higher frequencies, was solved by adding a buffer circuit to the outputs. These buffer circuits also have capacitance loading which in fact again limit the speed. However, the value of the loading capacitors are

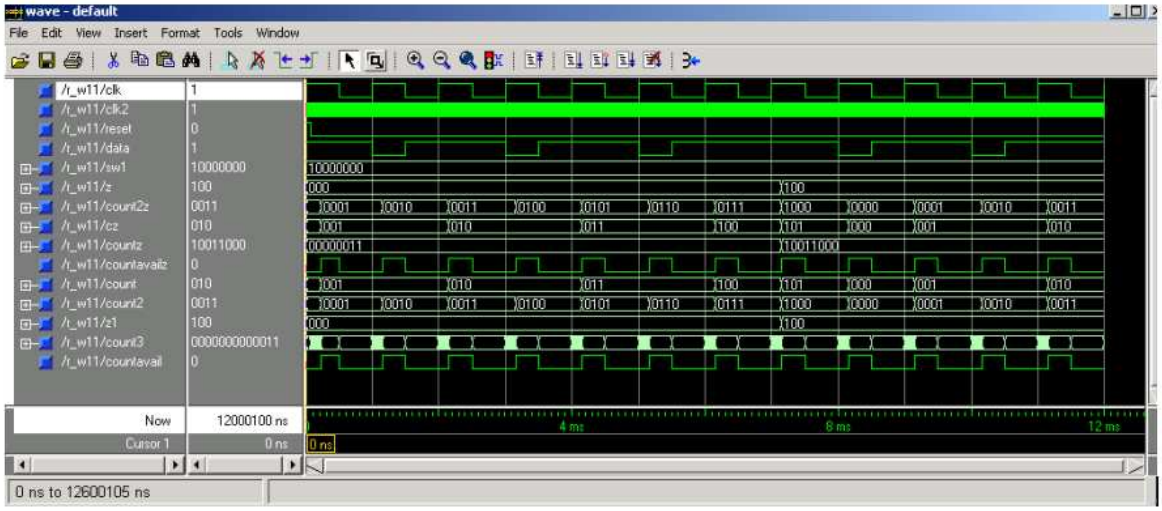

Figure 7.9. FPGA simulation result

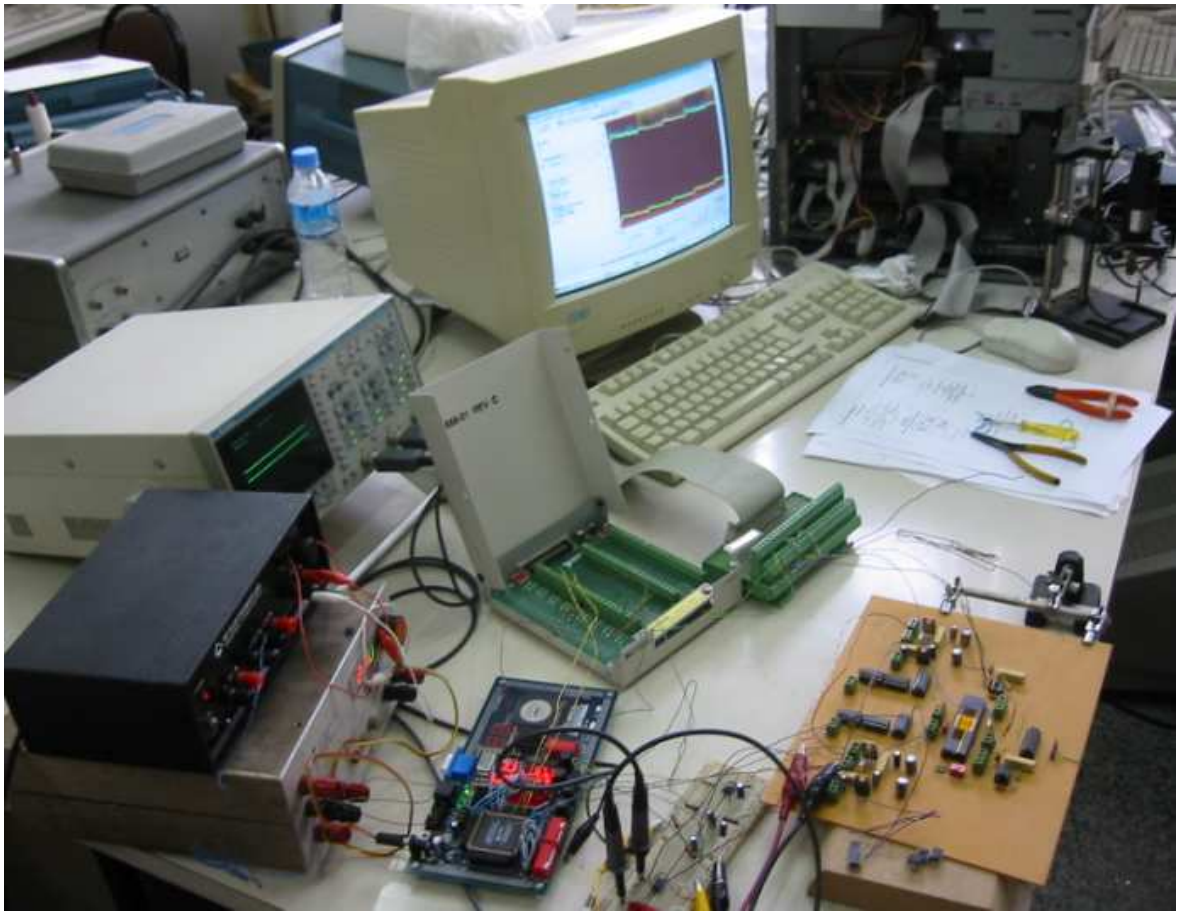

Figure 7.10. Test system

smaller. Thus higher speeds were achieved even with loading effects of the measurement equipment. With the previous PCB design, 25KHz was the limit for the clock. However with new PCB nearly 700KHz clock rate was achieved. Figure 7.11 shows the captured

output of the chip output with 60mV offset in the input. Measured offset value is presented in 7.12.

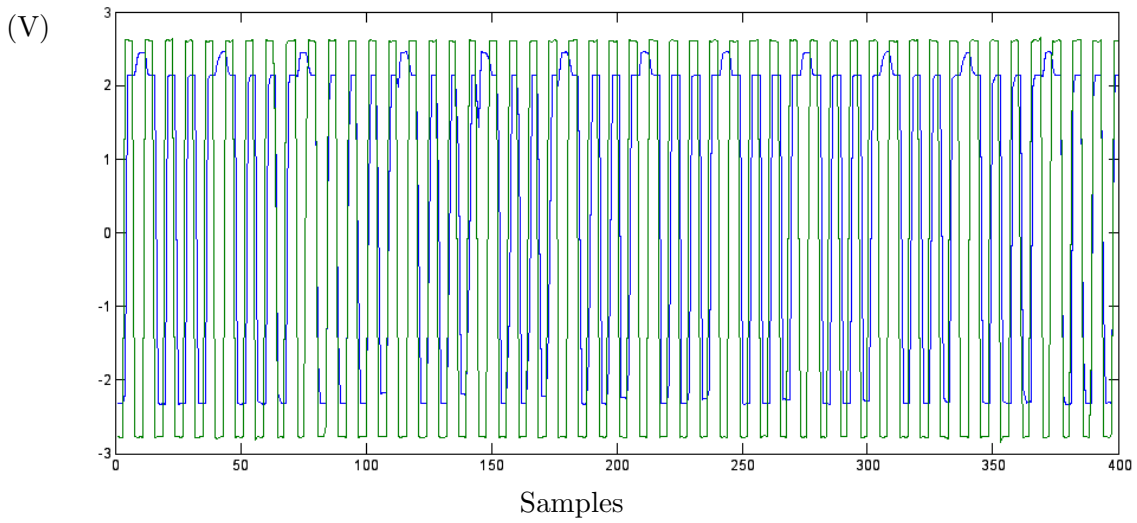

Figure 7.11. Reconstructed output

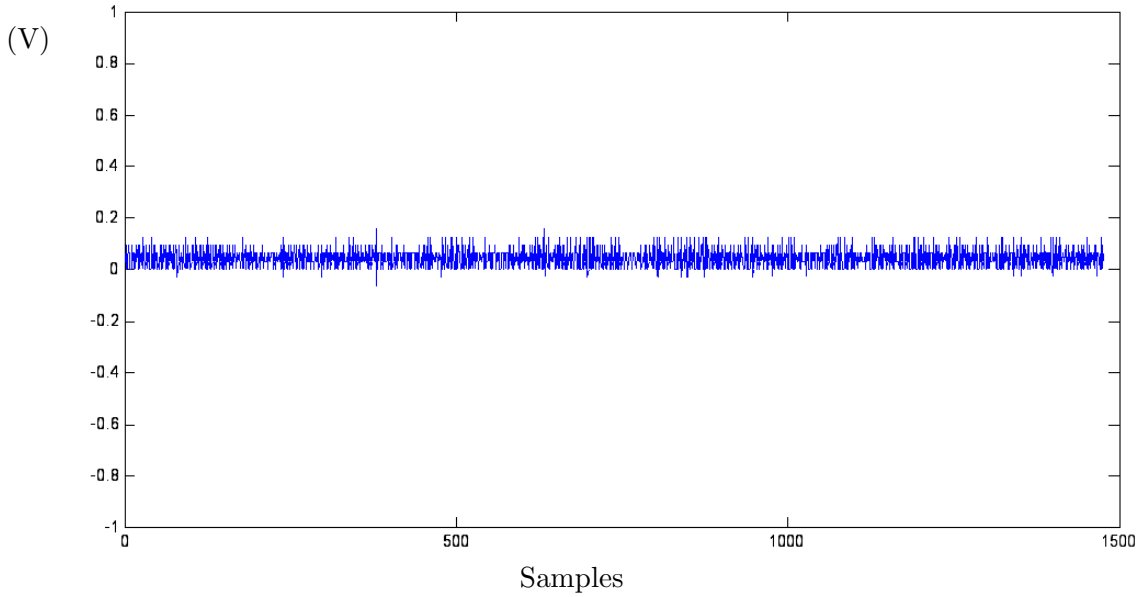

Figure 7.12. Measured offset value

The first order Sigma-Delta ADC modules in five of the chips were tested and their characteristics were measured as shown in Figure 7.15. The results showed that all designed chips are working correctly. One undesired result is that, all modules had offset value, that is the bias values of the differential inputs should be modified in order to achieve correct result at the output. In other words, the output digitally equals to

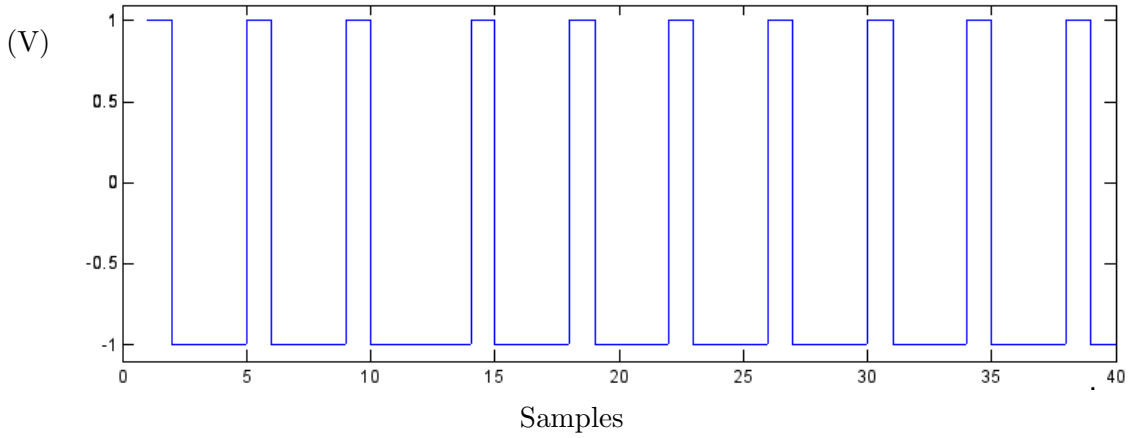

Figure 7.13. Digital equivalent of Figure 7.11

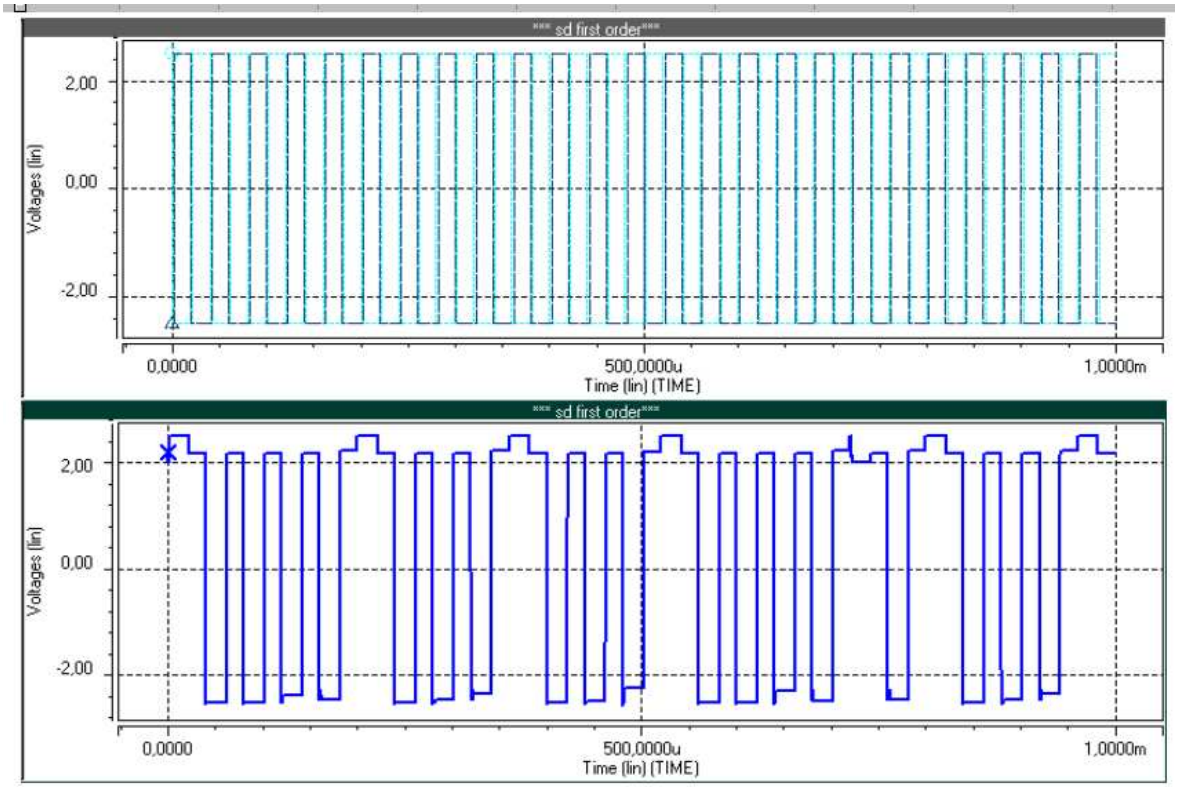

Figure 7.14. Hspice simulation results

zero when one input is 0V and the other is 60mV. The offset value for the chips varies between 58mV to 62mV.

The input-output characteristics of chips are given in Figure 7.17. The x-axis shows the analog input and y-axis shows the averaged analog value gathered from the output bit array. The input range is ±750mV and these values are mapped to ±1V.

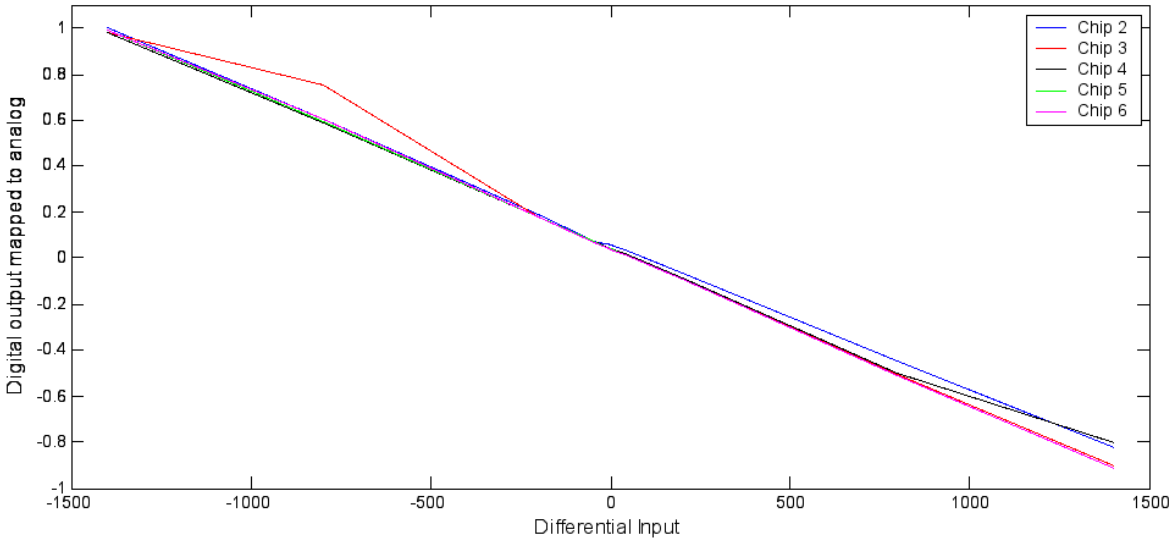

Figure 7.15. Measurement results for 5 IC

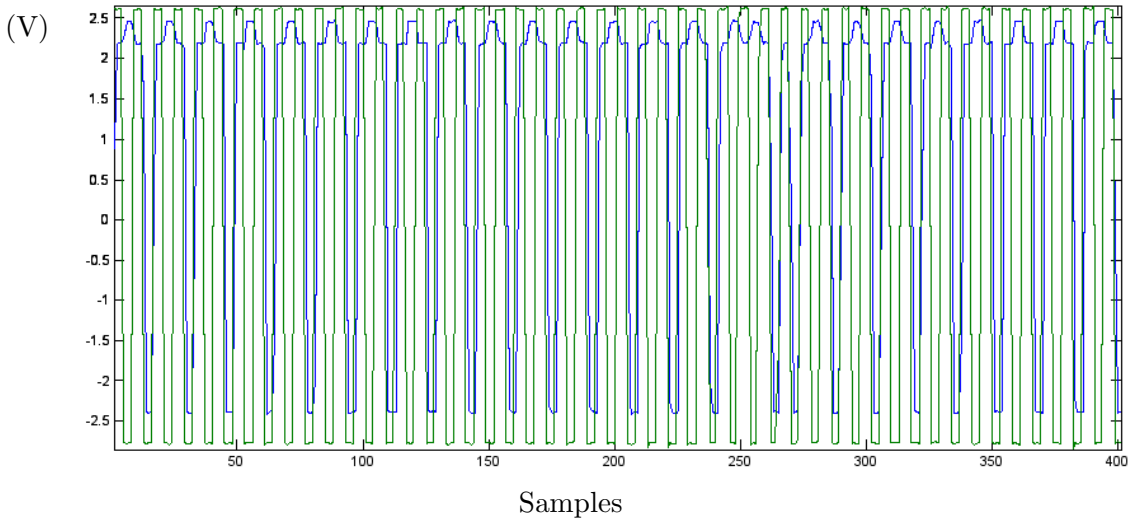

Figure 7.16. Modulator output with offset

The digital FPGA system is given in Figure 7.18. This digital filter implementation was connected to the analog output. The filter is basic decimation (sinc) filter, which converts the serial input to three bit digital output.

The whole ADC was also tested with static inputs. An input of 160mV was provided to the system. The digital output of the system was captured and can be seen in Figure 7.19. The average code for the input is 4.8 in 0-7 scale. With reference voltage of -750mV and 750mV, the output corresponds to 163mV.

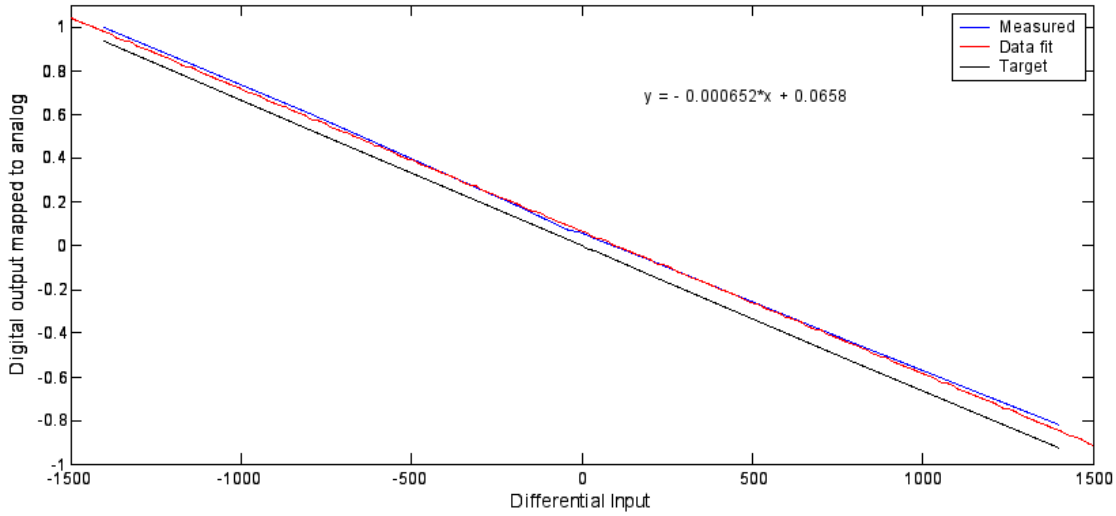

Figure 7.17. Input vs. output characteristics of Chip2

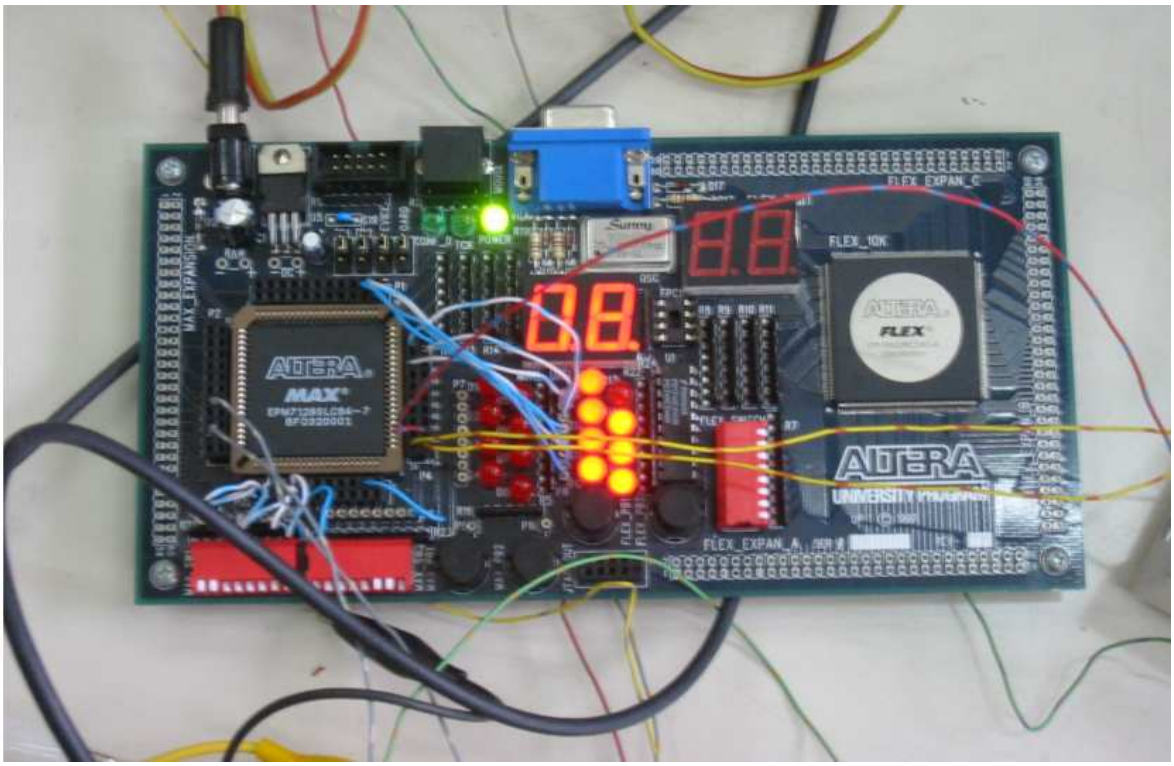

Figure 7.18. FPGA system

The static testing is not sufficient to test the system. Dynamic testing should also be carried out. The sine wave is a better input to test the dynamic behaviour. Then, the power spectral density of the system can be measured and the real SNR value can be found. This value is used to calculate effective number of bits, which is actually the real resolution of the system. However, the first setup is not sufficient to capture enough data point to plot PSD of the ADC. The data aquisition system has limited memory

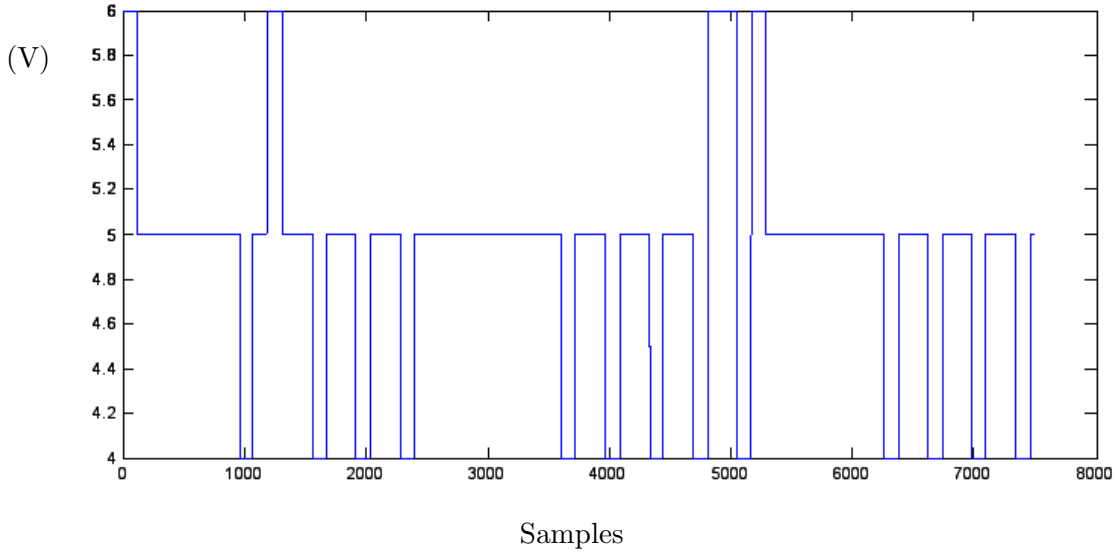

Figure 7.19. FPGA system output

and it is not possible the store enough points. Hence a new testing environment is prepared as shown in Figure 7.20. The main equipment for this setup is the DSP development kit of ALTERA. This FPGA kit has on board memory and it can be used to store samples. At the same time another advantage of this kit is that it is possible to use external clock for triggering or the internal clock may be used as clock signal by the device under test (DUT). The stored data is then transferred to PC via uart(RS232) interface. MATLAB environment read the stored data and PSD can be obtained as shown in Figure 7.21

The measurement results show the correct operation of the test IC. The IC was not designed for achieving a very high performance but rather to verify the models, hence the SNR of the IC is not very high. The measurement results are very similar to the post layout simulation results, which proves the accuracy of the developed models.

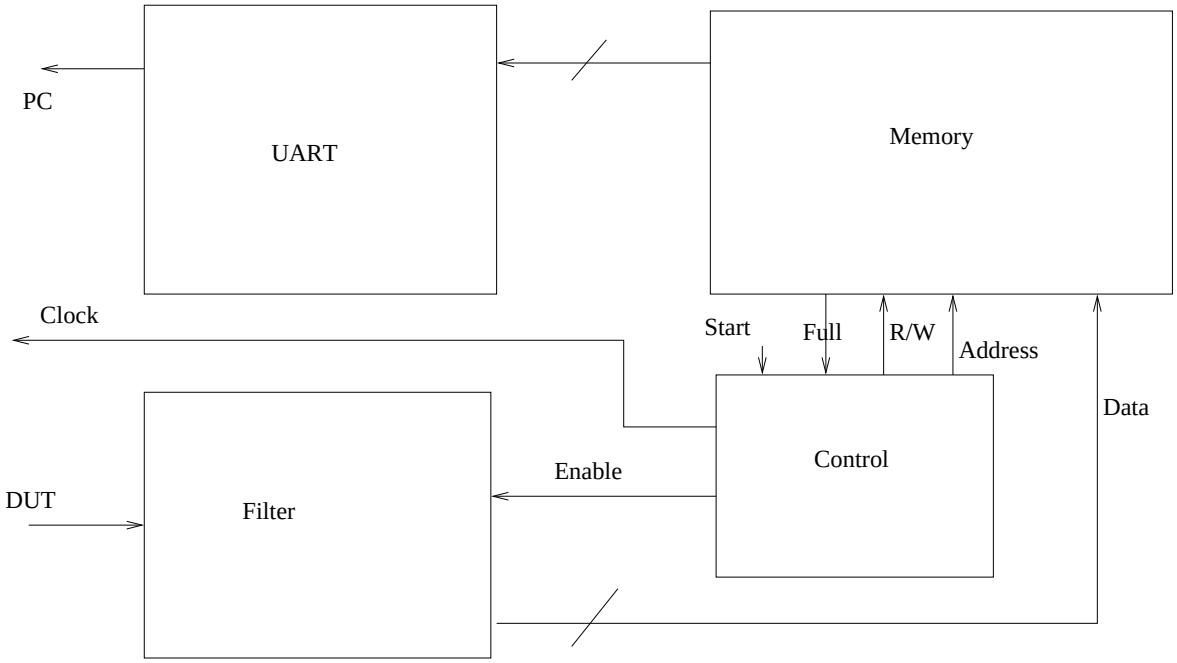

Figure 7.20. Block diagram of test structures in FPGA kit

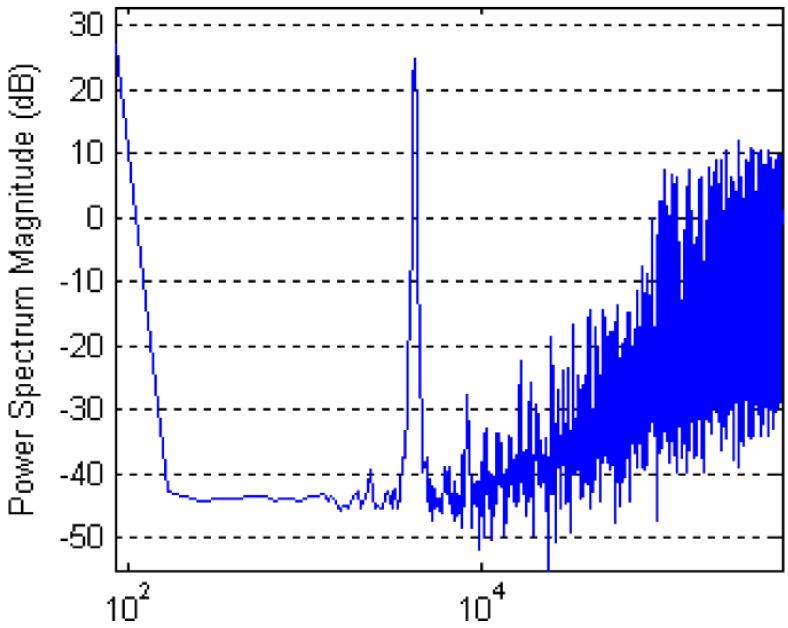

Figure 7.21. PSD of the test IC

8. CONCLUSION AND FUTURE WORK

In this thesis, high-level analytical models for Sigma-Delta ADC's were developed. Thus, the performance of different configurations of Sigma-Delta ADC'S can be calculated. In Sigma-Delta ADC design, the most difficult problem for a designer is to choose the most appropriate configuration for the desired specifications; in other words, optimize the ADC architecture. There are typically many known possibilities that can be a candidate for a specific performance criteria. However, there may also be many other possibilities or configurations that were not examined before. Therefore, high-level models are required for examining the available configurations and possible configurations. Hence, with the developed models, designer may have better understanding of parameters that degrade the performance.

With the help of developed models a design automation environment was developed. Thus, the designer may utilize the developed tool for performance calculation and architecture selection. The flexibility of the developed tool allows the designer to use different modes of operation for various needs. The designer may verify a candidate architecture or perform a design space search for only some part of the design or may use an automatic designer for larger design space search. Then, the designer may choose the most suitable configuration. Also, the designer can use the developed tool via a web based user interface. One of the novelties of the developed automation tool is that it utilizes a performance estimator. Hence, for architecture selection, not only the SNDR value is used but also area and power consumption can be considered. Therefore, better performances can be achieved regarding area and power. The performance estimator module can be used stand alone via web interface.

The thesis contains the development of models for Sigma-Delta ADC's and a design automation tool that utilizes these models. Also, model verification was performed by comparing results with the ones available in the literature. Matlab models were used to examine the behaviour of error sources and they are used to developed analytical models. These models form an equation database that the design automa-

tion tool can utilize. Especially jitter and slew-rate models are novel compared with similar research in the literature. Another novelty of the presented work is the ability to use a MATLAB tool designed for generating new architectures. The optimization section of the MATLAB tool was developed and merged with the automation system

In addition an IC was designed and fabricated in order to test model accuracy. The IC was designed from system level specifications down to layout level. All the blocks were designed according to the specifications or tolerances that was achieved form developd models. A testing environment was designed and tests were performed which show that the IC is working as it was expected.

As a summary, a design automation tool for Sigma-Delta ADC's working at the system level was developed. New models were developed, by the introduction of a performance estimator, innovative operation of the tool is achieved. Optimization functions were developed for achitecture generation module and an IC was designed for testing purposes. The developed tool provides an automated assistance to the designers with its flexibility and user interface. The novelties of the achievements in the thesis were confirmed by journal and conference papers.

8.0.1. Future Work

In this thesis, many aspects of Sigma-Delta ADC modeling, design and design automation are presented. However, there are also many research topics remain that warrant further study.

- The developed MATLAB module for exploring the design space for new architectures aim to find a solution which definitely satisfy the provided STF and NTF functions. However, the main aim should be the satisfaction of the STF and the NTF with in a given tolerance limit, which may ease the circuit design and may result in lower power and area consumptions. Since, the STF should be defined not as a all-pass filter but rather as a bandpass filter which covers the the band of interest, the performance reduction of STF which does not lie within the

in-band is not important. A better approach can be the utilization of optimization algorithms for minimizing the error between the STF of the "synthesized" architecture and the provided STF defined in a given bandwith.

- Performance estimator module should contain more structures which are preferred by Sigma-Delta ADC designers. Such as folded-cascode amplifier structure, there are other amplifier structures left that are not modeled in this thesis.
- The performance estimator module does not consider the length of the transistor as a design parameter and hence, does not sweep the lengths. Therefore, the solution space is limited and most of the time the results achieved do not differ from each other as much as it is expected.

REFERENCES

1. Razavi, B., *Principles of Data Conversion System Design*, IEEE Press, New York, 1995.

2. Jespers, P., *Integrated Converters*, Oxford University Press, 2001.

3. Candy, J. C. and G. C. Temes (editors), *Oversampling Delta-Sigma Data Converters Theory, Design and Simulation*, IEEE Press., New York, 1992.

4. Yao, L., M. Steyaert, and W. Sansen, "A 1-V 140W 88-dB Audio Sigma-Delta MOdulator in 90-nm CMOS", *IEEE J. Solid-State Circuits*, , No. 11, November 2004.

5. Rio, R. *et al.*, "Highly Linear 2.5-V CMOS $\Sigma\Delta$ Modulator for ADSL+", *IEEE Trans. Circuits Syst. I*, Vol. 51, No. 1, pp. 47–62, January 2004.

6. Francken, K. and G. E. Gielen, "A High-level Simulation and Synthesis Environment for Delta Sigma Modulators", *IEEE Trans. Computer-Aided Design*, Vol. 22, pp. 1049–1061, August 2003.

7. Medeiro, F., B. Perez-Verdu, and A. Rodriguez-Vazquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, Netherlands, 1999.

8. Malcovati, P. *et al.*, "Behavioral Modeling of Switched-Capacitor Sigma-Delta Modulators", *IEEE Trans. Circuits Syst. I*, Vol. 50, No. 3, pp. 352–364, March 2003.

9. Zhang, H. and A. Doboli, "Fast Time-Domain Symbolic Simulation For Synthesis of Sigma-Delta Analog-Digital Converters", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. 125–128, Vancouver, Canada, May 2004.

10. Medeiro, F., B. Perez-Verdu, A. Rodriguez-Vazquez, and J. Huertas, "A Vertically Integrated Tool for Automated Design of ΣΔ Modulators", *IEEE J. Solid-State Circuits*, Vol. 30, No. 7, pp. 762–772, July 1995.

11. Bajdechi, O., G. Gielen, and J. Huijsing, "Systematic Design Exploration of Delta-Sigma ADC's", *IEEE Trans. Circuits Syst. I*, Vol. 51, No. 1, pp. 86–95, January 2004.

12. Malcovati, P. *et al.*, "Behavioral Modeling of Switched-Capacitor Sigma-Delta Modulators", *IEEE Trans. Circuits Syst. I*, Vol. 50, No. 3, pp. 352–364, March 2003.

13. Ruiz-Amaya, J., J. Rosa, F. Medeiro, F. Fernandez, R. Rio, B. Perez-Verdu, and A. Rodriguez-Vazquez, "An Optimization-Based Tool for The High-Level Synthesis Of Discrete-Time and Continuous-Time ΣΔ Modulators in The MATLAB/SIMULINK Environment", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. 97–100, Vancouver, Canada, May 2004.

14. Williams, L., "MIDAS-A Functional Simulator for Mixed Digital and Analog Sampled Data Systems.", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. 2148–2151, San Diego, USA, May 1992.

15. Marques, A., V. Peluso, M. Steyaert, and W. Sansen, "Optimal Parameters for ΣΔ Modulator Topologies", *IEEE Trans. Circuits Syst. II*, Vol. 45, No. 9, pp. 1232 – 1241, September 1998.

16. Lauwers, E. and G. Gielen, "Power Estimation Methods for Analog Circuits for Architectural Exploration of Integrated Systems", *IEEE Trans. VLSI Syst.*, Vol. 10, No. 2, pp. 155–162, April 2002.

17. Schreier, R., "An Empirical Study of High-Order Single-Bit Delta-Sigma Modulators", *IEEE Trans. Circuits Syst. II*, Vol. 40, No. 8, pp. 461–466, August 1993.

18. Bajdechi, O., G. Gielen, and J. Huijsing, "Fast Exploration of ΣΔADC Design Space", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. II–49– II–52, Arizona, USA, May 2002.

19. Vogels, M. and G. Gielen, "Figure of merit based selection of A/D converters", *Design, Automation and Test in Europe Conference and Exhibition*, pp. 1090 – 1091, Munich, Germany, March 2003.

20. Brigati, S., F. Francesconi, P. Malcovati, and F. Maloberti, "A Fourth-Order Single-Bit Switched-Capacitor Σ-Δ Modulator for Distributed Sensor Applications", *IEEE Trans. Instrum. Meas.*, Vol. 53, No. 2, pp. 266–270, April 2004.

21. Morizio, J. *et al.*, "14-bit 2.2-MS/s Sigma-Delta ADC's", *IEEE J. Solid-State Circuits*, Vol. 35, No. 7, pp. 968–976, July 2000.

22. Talay, S. and G. Dündar, "High Speed Design Tool for Flash and Pipeline ADC's", *Proc. Eur. Conf. on Cir. Systems.*, Krakow, Poland, September 2003.

23. der Plas, G. V., G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts, "AMGIEA synthesis environment for CMOS analog integrated circuits", *IEEE J. Technol. Computer Aided Design*, Vol. 20, p. 10371058, September 2001.

24. Gielen, G. and R. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", *Proc. of the IEEE*, pp. 1825–1852, December 2000.

25. Balkır, S., G. Dündar, and A. Öğrenci, *Analog VLSI Design Automation*, CRC Press, 2003.

26. S.Talay and G.Dundar, "Sigma-Delta A/D eviricilerinin MATLAB ile Modellenmesi", *Proceedings of the IEEE 11th Signal Processing and Communications Applications Conference (SIU),(In Turkish)*, April 2003.

27. Talay, S. and G. Dündar, "Slew-Rate Effect in First Order Sigma-Delta

ADC's", *Proc. 12th IEEE Mediterranean Electrotechnical Conference*, pp. 95–98, Dubrovnik, Crotia, May 2004.

28. Talay, S. and G. Dündar, "Jitter Model of Sigma-Delta A/D Converters", *Proc. Signal Proc. and Comm. Applications Conf.,*, pp. 371 – 374, Kuşadası, Turkey, April 2004, (in Turkish).

29. S.Talay and G.Dundar, "A sigma-delta adc design automation tool", *PhD. Research in Microelectronics and Electronics (PRIME)*, pp. 62–65, July 2005.

30. S.Talay, E. Deniz, and G.Dundar, "A sigma-delta ADC design automation tool with embedded performance estimaton", *13th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, 2006.

31. Sağlamdemir, O., Ö. Yetik, S. Talay, and G. Dundar, "A Coefficient Optimization and Architecture Selection Tool for SD Modulators Considering Component Non-Idealities", *The 17th edition of ACM Great Lakes Symposium on VLSI*, pp. 423–428, March 2007.

32. Yetik, Ö., O. Sağlamdemir, S. Talay, and G. Dundar, "A Coefficient Optimization and Architecture Selection Tool for SD Modulators in MATLAB", *Design, Automation & Test in Europe Conference & Exhibition, 2007*, pp. 1–6, April 2007.

33. O.Gursoy, O.Saglamdemir, M.Aktan, S.Talay, and G.Dundar, "Low-Power Decimation Filter Architectures For Sigma-Delta ADC's", *4th International Conf. on Electrical and Electronics Eng.,(ELECO)*, December 2006.

34. Inose, H., Y. Yashuda, and J. Murakami, "Telemetering system by code modulation- SD modulation", *IRE Trans. Space Electron. Telemetry*, Vol. SET-8, pp. 204–209, September 1962.

35. Norsworthy, S. R., R. Schreier, and G. C. Temes (editors), *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE press, 1998.

36. Norsworthy, S. R., R. Schreier, and G. C. Temes (editors), *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE press, 1998.

37. Norsworthy, S. R., R. Schreier, and G. C. Temes (editors), *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE press, 1998.

38. Breems, L. and J. H. Huijsing (editors), *Continuous-time Sigma-Delta modulation for A/D conversion in radio receivers*, Kluwer Academic Publishers, 2001.

39. Rabii, S. and B. A. Wooley (editors), *The design of low-voltage, low-power, Sigma-Delta modulators*, Kluwer Academic Publishers, 1999.

40. Geerts, Y., M. Steyaert, and W. Sansen (editors), *Design of multi-bit delta-sigma A/D converters*, Kluwer Academic Publishers, 2002.

41. Kozak, M. and zzet Kale (editors), *Oversampled delta-sigma modulators : analysis, applications, and novel topologies*, Kluwer Academic Publishers, 2003.

42. de la Rosa, J. M., B. Prez-Verd, and A. Rodrguez-Vzquez (editors), *Systematic design of CMOS switched-current bandpass sigma-delta modulators for digital communication chips*, Kluwer Academic Publishers, 2002.

43. Yao, L., M. Steyaert, and W. Sansen (editors), *Low-Power Low-Voltage Sigma-Delta Modulators in Nanometer CMOS*, The Springer International, 2006.

44. Gerfers, F. and M. Ortmanns (editors), *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, The Springer International, 2005.

45. Silva., P. G. R., L. J. Breems, K. A. A. Makinwa, R. Roovers, and J. H. Huijsing, "An IF-to-Baseband $\Sigma\Delta$ Modulator for AM/FM/IBOC Radio Receivers With a 118 dB Dynamic Range", *IEEE J. Solid-State Circuits*, pp. 1076–1089, May 2007.

46. Chalvatzis, T., E. Gagnon, and M. R. andS. P. Voinigescu, "A Low-Noise 40-GS/s

Continuous-Time Bandpass ΣΔ ADC Centered at 2 GHz for Direct Sampling Receivers", *IEEE J. Solid-State Circuits*, pp. 1065–1075, May 2007.

47. Thandri, B. K. and J. Silva-Martinez, "A 63 dB SNR, 75-mW Bandpass RF ΣΔ ADC at 950 MHz Using 3.8-GHz Clock in 0.25-μm SiGe BiCMOS Technology", *IEEE J. Solid-State Circuits*, pp. 269–279, February 2007.

48. Schoofs, R., M. Steyaert, and W. Sansen, "A Design-Optimized Continuous-Time DeltaSigma ADC for WLAN Applications", *IEEE Trans. Circuits Syst. I*, pp. 209–217, January 2007.

49. Kyehyung, L., G. Temes, and F. Maloberti, "Noise-Coupled Multi-Cell Delta-Sigma ADCs", *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 27–30, May 2007.

50. Ouzounov, S., R. van Veldhoven, C. Bastiaansen, K. Vongehr, R. van Wegberg, G. Geelen, L. Breems, and A. van Roermund, "A 1.2V 121-Mode CT ΔΣ Modulator for Wireless Receivers in 90nm CMOS", *Proc. of the IEEE International Solid-State Circuits Conference*, pp. 242–600, February 2007.

51. Breems, L., R. Rutten, R. van Veldhoven, G. van der Weide, and H. Termeer, "A 56mW CT Quadrature Cascaded ΔΣ Modulator with 77dB DR in a Near Zero-IF 20MHz Band", *Proc. of the IEEE International Solid-State Circuits Conference*, pp. 238–599, February 2007.

52. Maloberti, F. and Y. Choi, "86 dB DR Cross-Coupled Time-Interleaved xx ADC for Audio Signal Band with 322 μA Current Consumption", *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 2148–2151, May 2007.

53. Yazkurt, U., N. Beilleau, G. Dundar, H. Aboushady, S. Talay, and L. de Lamarre, "Scaling Input Signal Swings of Overloaded Integrators in Resonator-Based Sigma-Delta Modulators", *IEEE International Conf. on Electronic Circuits and Systems*, December 2006.

54. Brigati, S., F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto, and F. Maloberti, "Modeling Sigma-Delta Modulator Non-idealities in SIMULINK", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. 384–387, 1999.

55. Aboushady, H., L. de Lamarre, N. Beilleau, and M. Louerat, "A mixed equation-based and simulation-based design methodology for continuous-time sigma-delta modulators", *47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS '04.*, Vol. 1, July 2004.

56. Hernandez, L., A. Wiesbauer, S. Paton, and A. D. Giandomencio, "Modelling and optimization of low pass continuous-time sigma delta modulators for clock jitter noise reduction", *Proc. of the IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 1072–5, May 2004.

57. Ortmanns, M., F. Gerfers, and Y. Manoli, "A continuous-time sigma-delta modulator with switched capacitor controlled current mode feedback", *Proc. of the 29th European Solid-State Circuits Conference,(ESSCIRC)*, pp. 249–252, September 2003.

58. Ortmanns, M., F. Gerfers, and Y. Manoli, "Fundamental limits of jitter insensitivity in discrete and continuous-time sigma delta modulators", *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. I–1037 – I–1040, May 2003.

59. Breems, L., R. Rutten, and G. Wetzker, "A cascaded continuous-time $\Delta\Sigma$ Modulator with 67-dB dynamic range in 10-MHz bandwidth", *IEEE J. Solid-State Circuits*, Vol. 39, pp. 2152 – 2160, December 2004.

60. Paton, S., A. D. Giandomenico, L. Hernandez, A. Wiesbauer, T. Potscher, and M. Clara, "A 70-mW 300-MHz CMOS continuous-time $\Delta\Sigma$ ADC with 15-MHz bandwidth and 11 bits of resolution", *IEEE J. Solid-State Circuits*, Vol. 39, pp. 1056–1063, July 2004.

61. Dorrer, L., F. Kuttner, P. Greco, P. Torta, and T. Hartig, "A 3-mW 74-dB SNR 2-MHz continuous-time delta-sigma ADC with a tracking ADC quantizer in 0.13μm CMOS", *IEEE J. Solid-State Circuits*, Vol. 40, pp. 2416 – 2427, December 2005.

62. Ortmanns, M., F. Gerfers, and Y. Manoli, "A Case Study on a 2-1-1 Cascaded Continuous-Time Sigma-Delta Modulator", *IEEE Trans. Circuits Syst. I*, Vol. 52, pp. 1515–1525, August 2005.

63. Gerfers, F., M. Ortmanns, and Y. Manoli, "A 1.5-V 12-Bit Power-Efficient Continuous-Time Third Order $\Sigma\Delta$ Modulator", *IEEE J. Solid-State Circuits*, Vol. 38, pp. 1343–1352, August 2003.

64. Ortmanns, M., F. Gerfers, and Y. Manoli, "Compensation of Finite Gain-Bandwith Induced Errors in Continuous-Time Sigma-Delta Modulators", *IEEE Trans. Circuits Syst. I*, Vol. 51, pp. 1088–1099, June 2004.

65. Tortosa, R., J. de la Rosa, and F. F. andA. Rodriguez-Vazquez, "A New High-Level Synthesis Methodology of Cascaded Continuous-Time Sigma-Delta Modulators", *IEEE Trans. Circuits Syst. II*, pp. 739–743, August 2006.

66. Zhang, H. and A. Doboli, "Fast time-domain symbolic simulation for synthesis of sigma-delta analog-digital converters", *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. V–125–V–128, May 2004.

67. Degrauwe, M. G. R. *et al.*, "IDAC: An Interactive Design Tool for Analog CMOS Circuits", *IEEE J. Solid-State Circuits*, Vol. 22, pp. 1106–1114, December 1987.

68. Mar, M. F. and R. W. Brodersen, "A Design System for On-Chip oversampling A/D interfaces", *IEEE, Transactions on VLSI systems*, Vol. 3, pp. 345–354, September 1995.

69. Koe, W. and J. Zhang, "Understanding The Effect of circuit Non-idealities on Sigma-Delta Modulator", *Proc. of IEEE Int. Workshop on Behavioral Modeling*

and Sim., pp. 94–101, Santa Rosa, California, October 2002.

70. Sommer, R., "Results form Anastasia+: Top-Down Design Methods for Mixed-Signal Applications", *MEDEA+ Design Automation Conference*, October 2002.

71. "MEDEA+", `http://www.medeaplus.org/`.

72. EPFL, "INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS 2005 EDITION", `http://www.itrs.net/Links/2005ITRS/SysDrivers2005.pdf`.

73. Goodenough, F., "Analog Technologies of all Varieties Dominate ISSCC", *Electronic Design*, Vol. 44, pp. 96–111, 1996.

74. Rabii, S. and B. A. Wooley, "1.8V Digital-Audio SigmaDelta Modulator in 0.8μm CMOS", *IEEE J. Solid-State Circuits*, Vol. 38, pp. 783–796, June 1997.

75. Nagel, L. W., "SPICE2: A Computer Program to Simulate Semiconductor Circuits", *ERL-M520, University of California, Berkeley*, 1975.

76. Fang, S., Y. Tsividis, and O. Wing, "SWITCAP: A switched-capacitor network analysis programPart I: Basic features", *IEEE Circuits Systems Mag.*, Vol. 5, pp. 4–10, September 1983.

77. Fang, S., Y. Tsividis, and O. Wing, "SWITCAP: A switched-capacitor network analysis programPart I: Advanced applications", *IEEE Circuits Systems Mag.*, Vol. 5, pp. 41–46, September 1983.

78. Dias, V., V. Liberali, and F. Maloberti, "TOSCA: A user-friendly behavioral simulator for oversampling A/D converters", *Proc. of the IEEE International Symposium on Circuits and Systems*, p. 26772680, 1991.

79. Vandewalle, J., H. D. Man, and J. Rabaey, "Time, frequency, and Z-domain modified nodal analysis of switched-capacitor networks", *IEEE Trans. Circuits*

Systems, Vol. 28, p. 186195, March 1981.

80. Medeiro, F., R. del Rio, J. de la Rosa, B. Perez-Verdu, and A. Rodriguez-Vazquez, "Architectures and Design Considerations for Wireline $\Delta\Sigma$ Modulators Beyond ADSL", *Journal of Measurement*, Vol. 37, p. 328343, March 2005.

81. Robertini, A. and W. Guggenbuhl, "Errors in SC Circuits Derived from Linearly Modeled Amplifiers and Switches", *IEEE Trans. Circuits Syst. I*, Vol. 39, No. 2, pp. 93–101, February 1992.

82. Medeiro, F., B. Perez-Verdu, J. Rosa, and A. Rodriguez-Vazquez, "Fourth-Order Cascade SC $\Sigma\Delta$ Modulators: A Comparative Study", *IEEE Trans. Circuits Syst. I*, Vol. 45, No. 10, pp. 1041–1051, October 1998.

83. R. Castro-Lopez, B. P.-V., F. Medeiro and A. Rodriguez-Vazquez, "Behavioral Modelling and Simulation of $\Sigma\Delta$ Modulators Using Hardware Description Languages", *Proc. of Design and Test in Europe*, March 2003.

84. Geerts, Y., M. Steyaert, and W. Sansen, "A High-Performance Multibit $\Sigma\Delta$ CMOS ADC", *IEEE Journal of Solid-State Circuits*, pp. 1829–1840, December 2000.

85. Fujimori, I., L. Longo, A. Hairapetian, K. Seiyama, S. Kosic, J. Cao, and S.-L.Chan, "A 90-dB SNR 2.5-MHz Output-Rate ADC Using Cascaded Multibit Delta-Sigma modulation at 8x Oversampling Ratio", *IEEE Journal of Solid-State Circuits Conference*, pp. 1820–1828, December 2000.

86. Brandt, B. and B. Wooley, "A 50-MHz Multibit Sigma-Delta Modulator for 12-b 2-MHz A/D Conversion", *IEEE J. Solid-State Circuits*, Vol. 26, pp. 1746–1756, December 1991.

87. Brooks, T., D. Robertson, D. Kelly, A. D. Muro, and S. W. Harston, "A Cascaded Sigma-Delta Pipeline A/D Converter with 1.25 MHz Signal Bandwidth and 89

dB SNR", *IEEE J. Solid-State Circuits*, Vol. 32, pp. 1896–1906, December 1997.

88. Medeiro, F., B. Perez-Verd, and A. Rodrguez-Vzquez, "A 13-bit, 2.2-MS/s, 55-mW Multibit Cascade $\Delta\Sigma$ Modulator in CMOS 0.7-μm Single-Poly Technology", *IEEE J. Solid-State Circuits*, Vol. 34, pp. 748–760, June 1999.

89. Hamoui, A. A. and K. Martin, "High-Order Multibit $\Delta\Sigma$ Modulators and Pseudo Data-Weighted-Averaging in Low-Oversampling $\Delta\Sigma$ ADCs for Broad-Band Applications", *IEEE Trans. Circuits Syst. I*, Vol. 51, pp. 72–85, Jannuary 2004.

90. Y.Cheng, C. Petrie, D. Comer, and D. Comer, "Multibit Delta-Sigma Modulator with Two-Step Quantization and Segmented DAC", *IEEE Trans. Circuits Syst. II*, Vol. 53, pp. 848–852, September 2006.

91. Cherry, J. and W. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*, Kluwer Academic Publishers, 2000.

92. Dagher, E., P. A. Stubberud, W. K. Masenten, M. Conta, and T. V. Dinh, "A 2-GHz Analog-to-Digital Delta- Sigma Modulator for CDMA Receivers With 79-dB Signal-to-Noise Ratio in 1.23-MHz Bandwidth", *IEEE J. Solid-State Circuits*, Vol. 39, pp. 1819–1828, November 2004.

93. Olieai, O., "Design of Continuous-Time Sigma-Delta Modulators With Arbitrary Feedback Waveform", *IEEE Trans. Circuits Syst. II*, Vol. 50, pp. 437–444, August 2003.

94. Fontaine, P., A. N. Mohieldin, and A. Bellaourar, "A Low-Noise Low-Voltage CT $\Sigma\Delta$ Modulator with Digital Compensation of Excess Loop Delay", *Proc. of the IEEE International Solid State Circuits Conf.*, pp. 498–499, 2005.

95. Yapar, U. and G.Dundar, "Current-Mode Circuits for Sigma-Delta Converters", *Proc. of the IEEE European Conference on Circuit Theory and Design*, August 2007.

96. Olieai, O., "State-Space Analysis of Clock Jitter in Continuous-Time Oversampling Data Converters", *IEEE Trans. Circuits Syst. II*, Vol. 50, pp. 31–37, January 2003.

97. Tortosa, R., J. de la Rosa, A. Rodriguez-Vazquez, and F. Fernandez, "Analysis of Clock Jitter Error in Multibit Continuous-Time ΣΔ Modulators with NRZ Feedback Waveform", *Proc. of the IEEE International Symposium on Circuits and Systems*, May 2005.

98. Miao, G., H. Yang, and T. Pushan, "An Oversampled A/D Converter Using Cascaded Fourth Order Sigma-Delta Modulation and Current Steering Logic", *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 412–415, 1998.

99. Marques, A., V. Peluso, M. Steyaert, and W. Sansen, "A 15-b Resolution 2-MHz Nyquist Rate ΣΔ ADC in a 1-μm CMOS Technology", *IEEE Journal of Solid-State Circuits*, pp. 1065–1075, July 1998.

100. Kuo, T.-H., K.-D. Chen, and H.-R. Yeng, "A Wideband CMOS Sigma-Delta Modulator With Incremental Data Weighted Averaging", *IEEE Journal of Solid-State Circuits*, pp. 11–17, January 2002.

101. Reutemann, R., P. Balmelli, and Q. Huang, "A 33mW 14b 2.5MSample/s ΣΔ A/D converter in 0.25μm digital CMOS", *IEEE Journal of Solid-State Circuits Conference*, p. 316, 2002.

102. Balmelli, P. and Q. Huang, "A 25MS/s 14b 200mW ΣΔ Modulator in 0.18μm CMOS", *ISSCC Dig. Tech.Papers*, pp. 74–75, February 2004.

103. Jiang, R. and T. S. Fiez, "A 14-bit ΣΔ ADC With 8 OSR and 4-MHz Conversion Bandwidth in a 0.18-μm CMOS Process", *IEEE Journal of Solid-State Circuits Conference*, pp. 63–74, January 2004.

104. Feldman, A., B. Boser, and P. Gray, "A 13-Bit, 1.4-MS/s Sigma-Delta Modulator for RF Baseband Channel Applications", *IEEE Journal of Solid-State Circuits Conference*, pp. 1462–1469, October 1998.

105. Geerts, Y., A. Marques, M. Steyaert, and W. Sansen, "A 3.3-V, 15-bit, Delta-Sigma ADC with a Signal Bandwidth of 1.1 MHz for ADSL Applications", *IEEE Journal of Solid-State Circuits*, pp. 927–936, July 1999.

106. Morizio, J., M. Hoke, T. Kocak, C. Geddie, C. Hughes, J. Perry, S. Madhavapeddi, M. Hood, G. Lynch, H. Kondoh, T. Kumamoto, T. Okuda, H. Noda, M. Ishiwaki, T. Miki, and M. Nakaya, "14-bit 2.2-MS/s Sigma-Delta ADC's", *IEEE Journal of Solid-State Circuits Conference*, pp. 968–976, July 2000.

107. del Ro, R., F. Medeiro, J. de la Rosa, B. Prez-Verd, and A. Rodrguez-Vzquez, "Correction-Free Multi-Bit Sigma-Delta Modulators for ADSL", *in Analog Circuit Design Structured Mixed-Mode Design, Multi-Bit Sigma-Delta Converters, Short Range RF Circuits (M. Steyaert, A.H.M. van Roermund, and J.H. Huijsing, Editors). Kluwer Academic Publishers,*, 2002.

108. Lee, K.-S. and F. Maloberti, "A 1.8V, 1MS/s 85dB SNR 2+2 Mash $\Sigma\Delta$ Modulator with 0.9V Reference Voltage", *Proc of the Symposium on VLSI Circuits*, pp. 71–74, June 2003.

109. Park, Y.-I. *et al.*, "A 16-bit, 5-MHz multi-bit sigma-delta ADC using adaptively randmoized DWA", *in Proc. IEEE Custom Integrated Circ.Conf*, pp. 7–2–17–2–4, September 2003.

110. Yetik, O., *An Automatic Architecture Generator For Sigma-Delta Mmodulators Considering Component Non-idealities*, Master's thesis, Bogazici University, 2007.

111. Schereier, R., "The Delta-Sigma Toolbox 7.1 (Online)", December 2004, `http://www.mathworks.com/matlabcentral/fileexchange`.

112. Schaumann, R., M. Ghausi, and K. Laker, *Design of Analog Filters*, Prentice Hall Series in Electrical and Computer Engineering, 1990.

113. sağlamdemir, O., *Implentation and Performance Evaluation of Sigma-Delta Modulators*, Master's thesis, Bogazici University, 2007.

114. Peluso, V., M. Steyaert, and W. Sansen, *Design of Low–Voltage Low–Power CMOS Delta–Sigma A/D Converters*, Kluwer Academic Publishers, 1999.

115. Held, M. and R. Karp, "The traveling-salesman problem and minimum spanning tress", *Operations research*, pp. 1138–1162, 1970.

116. Enz, C., F. Krummenacher, and V. E.A., "An Analytical MOS Transistor Model Valid in All Regions of Operation and Dedicated to Low-Voltage and Low-Current Applications", Vol. 8, pp. 83–114, 1995.

117. Bucher, M., "Analysis of Transconductances in Deep Submicron CMOS with EKV 3.0", *MOS Models Parameter Extraction Workgroup*, , No. XFAB, Erfurt, Germany, October 2002.

118. Kayal, D. S. M., "BSIM2EKV: BSIM3.3 to EKV 2.6 Model Library File Automatic Conversion", *MOS Modeling and Parameter Extraction Working Group Workshop*, Lausanne, November 2004.

119. Mawet, P., "Low-power circuits and beyond; a designer's prospective on the EKV model and its usage", *MOS Modeling and Parameter Extraction Working Group Workshop*, Montreux, September 2006.

120. EPFL, "EKV model for 0.5μm", `http://legwww.epfl.ch/ekv/`.

121. Fischer, G. and A. Davis, "Alternative Topologies for Sigma-Delta Modulators - A Comparative Study", *IEEE Trans. Circuits Syst. II*, Vol. 44, No. 10, pp. 789–797, October 1997.

122. Dong, Y. and A. Opal, "An Overview on Computer-Aided Analysis Techniques for Sigma-Delta Modulators", *Proc. IEEE Int. Symp. Circuit Syst.*, pp. 423 – 426, Sydney, Australia, May 2001.

123. Geerts, Y., M. Steyaert, and W. Sansen, "A High-Performance Multibit $\Sigma\Delta$ CMOS ADC", *IEEE J. Solid-State Circuits*, pp. 1829–1840, 2000.

124. O.Saglamdemir, O. Yetik, S. Talay, and G. Dundar, "A Coefficient Optimization and Architecture Selection Tool for SD Modulators Considering Component Non-Idealities", *The 17th edition of ACM Great Lakes Symposium on VLSI*, March 2007.

125. Dias, V., G. Palmisano, P. O'Leary, and F. Maloberti, "Fundamental Limitations of Switched-Capacitor Sigma-Delta Modulators", *IEE Proceedings*, pp. 27–32, February 1992.

126. Maloberti, F. (editor), *Analog Design for CMOS VLSI Systems*, Kluwer Academic, 2001.

127. Martens, E. and G. Gielen, "Classification of analog synthesis tools based on their architecture selection mechanisms", *Integration VLSI J.*, p. doi:10.1016/j.vlsi.2007.06.001, 2007.