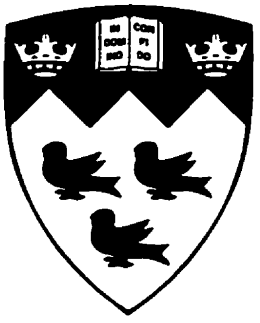
A Top-Down Approach to Delta-Sigma Modulator Design

Naveen Chandra,

(B.Eng 1999)

Department of Electrical Engineering

McGill University, Montréal



June 2001

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

This work presents the use of a new design methodology for the creation of analog integrated circuit components featuring optimization with Matlab and Simulink. This procedure allows circuit design to take place at the highest level of abstraction, and has the added advantage that designs can be implemented with currently used, and widely available tools. It results in building block requirements being specified prior to the undertaking of transistor level simulations, thereby saving much valued design time.

Also presented are the issues behind the design of an audio-band, single bit switched capacitor delta-sigma modulator with 16 bits of performance. This begins with the filtering function design, continues with the realization in a low voltage standard CMOS process, and concludes with experimental measurements to gauge performance. The creation of the modulator was carried out using the proposed top-down design methodology, and its experimental performance is used to help validate the procedure.

In addition, the design and fabrication of a multibit $\Delta\Sigma$ modulator including a novel internal DAC was carried out. The multibit DAC is based on encoding DC levels into digital PDM bitstreams, which are then decoded by a single analog filter. The use of a single filtering path reduces mismatch effects in the DAC. The functionality, performance, and limitations are explored by examining the experimental results of a 2.5 V, audio-band, 2^{nd} order, 3-bit $\Delta\Sigma$ modulator in a 0.25 μm CMOS process.

Résumé

Ce travail présente l'utilisation d'une nouvelle approche de conception pour la création des composants analogiques de circuit intégré comportant l'optimisation avec Matlab et Simulink. Ce procédé permet à la conception de circuit d'avoir lieu à un niveau d'abstraction plus élevé et a l'avantage que la conception peut être effectuée avec les outils actuels disponibles et largement utilisés. Il a comme conséquence conditions de module qui sont définies avant le début des simulations au niveau des transistors, reduisant consideralement le temps de conception.

Les differents aspects de la conception d'un modulateur delta-sigma avec 16 bits de résolution pour une application audio sont également présentés. Ceci commence par la conception des fonctions de filtrage, la réalisation dans une technologie CMOS à basse tension s'en suit, et finit avec des mesures expérimentales pour mesurer l'exécution. La création du modulateur a été effectuée en utilisant la méthodologie de conception hierarchique (haut vers le bas) proposée, et son exécution expérimentale est utilisée pour valider le processus.

De plus, la conception et la fabrication d'un modulateur $\Delta\Sigma$ multibit comprenant une nouvelle topologie de DAC interne ont été fabriqué. Le DAC multibit est basé sur l'encodage des niveaux DC dans les trains d'onde digitaux de PDM, qui sont alors décodés par un filtre analogique. L'utilisation d'une voie d'accès de filtrage simple attenue l'impact des erreurs liées au DAC. La fonctionnalité, l'exécution et les limitations sont explorées en examinant les résultats expérimentaux du modulateur $\Delta\Sigma$ de deuxième ordre avec 3-bits fonctionnant à 2.5 V conçu pour une application audio et fabriqué dans un procédé CMOS de 0.25 μm.

Acknowledgements

The work presented herein was made possible by the unselfish contribution of knowledge, expertise, and especially support of many people.

To start, I would like to express my gratitude to my supervisor, Professor Gordon Roberts. His boundless enthusiasm, excitement, optimism, and drive are largely responsible for the success of this work. Furthermore, his unique outlook and approach to analog circuit design has reinforced my desire for a career in this exciting field.

I would also like to express my gratitude to the many members of the MACS lab for providing a stimulating and supportive research environment. In particular, I would like to thank Nazmy, Arshan, Geoffrey, Yang, Ian, Lige, Mohamed, and Sebastien not only for the many useful technical discussions we engaged in, but also for their friendship. I can truly say that I will miss working with this group, as they have helped make my degree a truly rich and rewarding experience.

My appreciation also extends to my closest friends Ronald, Gayle and Jessica whose support shone through with welcome distractions and much needed diversions. In times of personal hardship they were always there, and more often than I can count, restored some balance to my life.

Mostly, I would like to thank my parents Prakash and Brunhild, and my brother Philip, for their love and constant support throughout my entire education. Their encouragement was vital, and their belief in my abilities was often stronger than my own.

Finally, I would like to acknowledge the Natural Science and Engineering Research Council (NSERC) and MICRONET for their funding assistance, Texas Instruments for providing a mixed-signal tester for the lab, and the Canadian Microelectronics Corporation (CMC) for making available their integrated circuit fabrication resources and for providing much needed equipment in the lab.

Table of Contents

List of Figures

CHAPTER 1 Introduction

This chapter will discuss the motivation for the work that will be presented in the thesis. Featured among this work is a new top-down design methodology using Matlab and Simulink, which will be validated through the fabrication and testing of an integrated circuit. Also featured is a review of the issues behind the design of $\Delta\Sigma$ modulators, including the design of their filtering functions, the implementation of a modulator in a low voltage process, and other associated problems.

1.1 Motivation

A top-down design methodology consists of the division of labour in a progressive manner, among several levels of abstraction. On the conception of an *idea* and the setting of performance goals, design at the *system* level can commence. It is at this stage that the functionality of the proposed system is tested, the design or selection of a high level architecture is made, and the analog building blocks necessary for implementation are determined. Following this, the implementation of the building blocks at the *circuit* transistor level is performed, followed by the *layout* implementation of the transistors and other components, and this is then completed by *fabricating* the design in silicon. A top-down design cycle is illustrated in Figure 1.1.

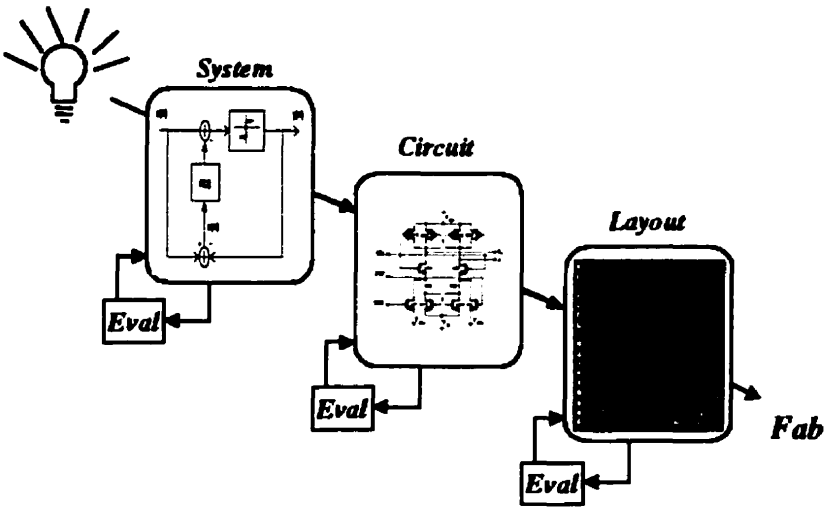


Figure 1.1: Top-Down Design Flow

In particular, the design of analog systems primarily consists of three obstacles:

1. The selection of an architecture.
2. The determination of the specifications for the analog building blocks necessary to implement the chosen architecture.
3. The minimization of the effects due to circuit non- idealities.

These are usually done separately, and more often than not, most of the building block specifications and non-idealities are explored through simulation at the transistor level using circuit simulation programs such as SPICE. To verify the circuit with respect to changes in passive component (resistor, capacitor, etc....) and transistor characteristics requires simulation over the full range of these variations. Obtaining performance curves, when considering a single architecture, let alone multiple architectures, also requires multiple simulations. Attempting to perform this type of analysis at a low level can be problematic due to long simulation times [1]. Most system simulations require days to run for each case given the fastest workstations. As a result, the design cycle will take too long to practically meet the market demands for the technology, especially given the rapid progression seen today. Table 1.1, shows the Semiconductor Industry Association (SIA) predictions made in

Feature	1999	2002	2005	2008	2011	2014
Line Width (μm)	0.18	0.13	0.10	0.07	0.05	0.035
Transistors/Chip (Millions)	23.8	47.6	190	539	1523	4308
Frequency (MHz)	1200	1600	2000	2500	3000	3600
Metal Layers (max)	7	8	9	9	10	10
Supply Voltage (volts)	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.3-0.6

Table 1.1: Long-term Technology Roadmap [2]

1999. As is evident, Very Large Scale Integrated circuit (VLSI) technology feature sizes are expected to shrink, re-enforcing the need for shorter design cycles.

On top of this problem, the results obtained from SPICE may be limited by rounding and truncation errors which accumulate over the large amount of time steps needed for accurate simulation. Furthermore, it is often difficult to focus on key design parameters when working with many integrated devices at the transistor level. In addition to this, the entire procedure must be carried out for each architecture selected, necessitating even further simulation. Therefore, the designer has to be very careful when choosing an architecture for fear of losing much time in the initial design stages.

In order to reduce the number of design iterations and to better explore the design options, it is beneficial to perform the analysis at the system architectural level before starting transistor level design. This allows for a feasibility analysis in which all the design considerations are treated at the highest level of abstraction. The ultimate goal of this procedure is to have the low level circuit parameters dictated by the selected architecture and desired performance. Therefore, a top-down design procedure using optimization will be presented which makes use of Simulink modeling and Matlab optimization [3].

1.2 Background

In order to provide some perspective on where this work fits in, a brief high-level survey on the state of Computer-Aided Design (CAD) for analog and mixed-signal integrated circuits is carried out.

It is widely accepted that analog circuit design is a knowledge intensive task that involves a significant amount of time including multiple iterations. In addition, successful design usually involves the participation of designers with a large skill base gained through much experience. As a result, the techniques required to build good quality analog circuits seem to primarily exist in the form of expertise held by individual design engineers [4]. Due to this fact, it is the analog sections of the designs that most negatively impact the turnaround or cycle time of analog and mixed-signal integrated circuits [5]. Furthermore, most analog systems are ported by hand as they continually move from one fabrication process to another.

The lack of easily transferable expert design knowledge, and the length of the design cycle needed for quality analog design, illustrates just a couple of the needs that motivate the development of CAD tools for analog circuits. In fact, CAD tools which deal with analog integrated circuit design can benefit and improve the design process in multiple ways and have made analog design automation appealing for several decades. Some of the most obvious and important benefits of analog CAD tools are [6, 7]:

1. Reduction in design time, and as a result design cost.

2. Retention of expert design knowledge by capturing the relevant information through the CAD tools themselves.

3. Improvement in error-free designs on the first fabrication run as error prone tasks become automated.

4. Ability to easily design circuits for different fabrication processes with minimal additional effort (top-down approaches only).

The first attempts at analog design automation involved linear circuit synthesis. It was motivated by the growing demand for cheap passive analog filters for use in the telephone industry between 1920 and 1940 [8]. The main focus was on the use of simple techniques designed to describe circuits as a series of equations, which could then be directly solved

to obtain the design values [9]. Although these methods were relatively fast, they proved to be difficult to update, and were also found to be inaccurate [10].

As time progressed through the 1970's, CAD tools for generating analog circuit design equations, such as SNAP [11], were developed. Their major drawback included the inability to generate anything other than exact expressions, and as such could only be used in specific situations or cases. It was not until the middle 1980's that the desire and need for self-contained CAD tools gained prominence.This was primarily motivated by the then recent success of digital CAD tools, which allowed for most components to be designed quickly and efficiently [5, 12]. As a result, much work and thought was put into developing more diverse and comprehensive analog tools [6, 13, 14].

From this renewed interest in CAD tools, two broad categories of approaching analog designs developed [4, 5]:

1. Bottom-up layout based approach.

2. Top-down knowledge based approach.

The bottom-up approach derives its origins from digital design ideas, and is carried out by fixing some aspect of the lower level of design [4, 5]. There are three main approaches that fall into this category:

1. Transistor arrays [15-17]. These arrays are analogous to digital gate arrays. They simplify layout, but have a very poor layout density. They also limit the design to a maximum number of transistors.

2. Analog library cells [18-20]. These cells provide a higher layout density, but are not very flexible because the designer is limited to a fixed set of circuit blocks.

3. Fixed-topology module generators [21, 22]. These modules fix some aspect of the circuit's topology and parameterize's the rest. This leaves a restricted number of choices for the designer.

All of these methods help to facilitate layout, but they do not help the designer to create a circuit from a set of performance specifications, and as a result, they are only effective for medium performance designs [5].

Top-down design approaches are based on transforming performance goals and specifications into circuit schematics, and have already been encountered in section 1.1. In general, this design approach can be divided into two groups:

1. Optimization-based design. This type of design models the circuit or system as a large set of coupled non-linear equations, and relies on numerical techniques to determine a solution. Many of the available CAD tools involve simulated annealing [23] such as DELIGHT.SPICE [24], ASTRX/OBLX [25], MAELSTROM [10], and ADSA[26]. Alternatively some tools use a search algorithm similar to gradient based searches such as the vertically integrated tool for $\Delta\Sigma$ modulators does in [27].

2. Knowledge-based design. This type of design is based on translating a set of performance specifications into circuit schematics using detailed analog domain knowledge [28]. This has been accomplished in CAD tools such as BLADES [8], IDAC [29], and OASYS [4]. It also extends to neural network approaches that can be seen in [30, 31].

Top-down design approaches have demonstrated some success, but for the most part each tool is limited to a small, fixed number of schematics or components. These tools, which are not designed to be reproduced, remove the designer from the process, and do nothing to increase the designer's knowledge. Furthermore the techniques used in these programs are hidden and cannot be applied to other designs. As a result, this thesis will in part present an easily reproduceable top-down design approach using widely available tools.

1.3 Thesis Overview

Chapter 2 begins by reviewing issues behind the design of $\Delta\Sigma$ modulator filtering functions, and deals with the problems of mapping them onto a viable architecture. A lowpass,

third order Lossless Discrete Integrator (LDI), audio-band (bandwidth of 24 KHz), single bit, delta-sigma modulator is used as an example to introduce some design, simulation, and implementation issues involved in designing these filters.

In Chapter 3, a new design methodology for the creation of analog or mixed signal integrated circuit components is presented. It features the use of top-down design techniques in conjunction with an optimization process, so that circuit design can take place at the highest level of abstraction. As a result, it allows for the requirements of the building blocks to be specified prior to the undertaking of transistor level simulations, thereby saving much valued design time. This method has the added advantage that designs can be implemented with currently used, and widely available tools such as Matlab and Simulink. For illustration, a design example featuring a third order, switched capacitor delta-sigma modulator will be presented.

Chapter 4 will continue by introducing the issues behind the design of a low voltage switched capacitor CMOS $\Delta\Sigma$ modulator, starting with the design procedure and realization in a low voltage (2.5 V) standard CMOS process (0.25 μm), and concluding with experimental measurements to gauge performance. The objective of this chapter is to design a lowpass, third order lossless discrete integrator, audio-band (bandwidth of 24 KHz), single bit, delta-sigma modulator with a desired 16 bits of performance. The design of the modulator was carried out using the methodology proposed in Chapter 3, and its experimental performance will be shown to validate this procedure.

Chapter 5 includes the design and fabrication of a multibit $\Delta\Sigma$ modulator, along with the design of a novel internal DAC. The novel multibit DAC is based on encoding DC levels into digital PDM bitstreams, which are then decoded by a single analog filter. The use of a single filtering path reduces mismatch effects in the DAC. The functionality, performance, and limitations of this method are explored by examining the experimental results of a lowpass, 2.5 V, audio-band, 2^{nd} order, switched capacitor, 3-bit $\Delta\Sigma$ modulator in a 0.25 μm CMOS process.

Chapter 6 will then summarize the work presented throughout this thesis.

CHAPTER 2 Design and Mapping of the Filter Functions For Lowpass ΔΣ Modulation

This chapter introduces the issues behind the design of ΔΣ modulator filtering functions, and deals with the problems of mapping them onto a viable architecture. The primary objective is to design a lowpass, third order Lossless Discrete Integrator (LDI), audio-band (bandwidth of 24 KHz), single bit, delta-sigma modulator. In order to achieve the desired 16 bits of performance, an oversampling ratio of 128 was used, necessitating a system clock run at 6.144 MHz.

2.1 Introduction

With the ever growing importance of digital technology in signal processing and data communications, the need for accurate high speed data converters becomes a pressing issue. Traditional Analog to Digital converters (ADC) and Digital to Analog converters (DAC) are often a source for non-linear distortion in the conversion process, due to the properties of quantization components and mismatches in circuit elements. These problems can be remedied through expensive means such as custom laser trimming, or averaging techniques that can effectively compensate for systematic distortion, but slow down conversion speeds [1]. The use of ΔΣ modulators provides for an alternative form of data conversion, whereby the principles of noise shaping and Pulse Density Modulation (PDM) are

utilized to arrive at simple yet effective data conversion structures. The characteristics of conventional A/D converters and the associated quantization process involved must be examined in order to appreciate the alternatives offered by ΔΣ modulators.

A quantizer is an integral component of any ADC. It is virtually impossible to represent an infinite multitude of input analog voltages, since a digital word of fixed length is the resulting output of the conversion process. This fixed word length limits the total number of voltage levels that can be represented, and as such an element that approximates sampled analog voltages to the nearest representable voltage is required for an ADC to be rendered functional. It is the role of the quantizer to perform such a function.

The quantizing action introduces errors in signal representation since the output voltage levels are merely approximations of the input voltage levels in each sample interval. Therefore, no ADC operation is exact. The greater the number of quantization levels an ADC possesses, the more accurately the input analog waveform may be described.

The system that is the subject of discussion, is a one bit lowpass A/D ΔΣ modulator. Figure 2.1 represents a simplified conceptual schematic of a basic ΔΣ modulator [32]. The schematic indicates that in the feed forward path of the system, a discrete time signal, *x(n)* (which is a sampled version of an analog signal) is converted directly into a digital signal by the use of a one bit ADC. For a one bit converter, there are only two output voltage levels. Any voltage higher than zero volts is represented by positive unit value voltage, while any voltage below this threshold is represented by a negative unitary voltage.

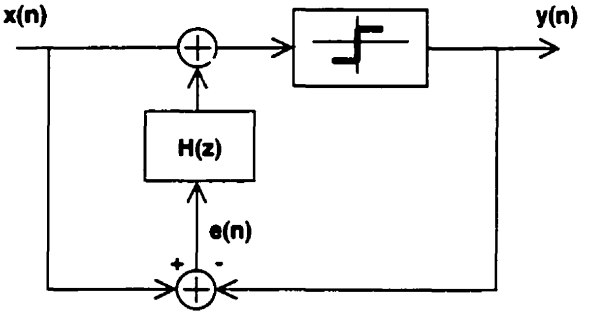


Figure 2.1: Conceptual Schematic Model of a ΔΣ Modulator

In the feedback path of the system the digital signal is converted back to an analog signal, and the original input analog signal is subtracted from the recovered signal. This operation inherently produces the quantization error, which is a measure of the amount of error introduced by the conversion process. This error signal, $e(n)$, which represents a prediction of the offset that will be produced by the analog to digital conversion operation for a particular sample is then filtered by the transfer function $H(z)$, and then subtracted from the input. The aim is to render the effect of the error negligible once it is re-introduced in the conversion process. The aim of filtering the error is to produce what is called noise shaping, which will be examined in greater length.

Due to the fact that practical discrete time components are being utilized, there is invariably a delay between the time a sampled analog input is presented to the system, and the calculation of the error signal. As such, the error signal that is generated is not entirely representative of the sample being presented at the input, but rather it is that of the previous sample. If the adjacent samples of the input signal are correlated, then the error produced by previous samples can be representative enough to be utilized as an estimate for the error produced by the present sample. This correlation is often accomplished by oversampling the input signal. In this manner, a more accurate data conversion can be attained, since a significant amount of the error can be accounted for in advance.

The noise shaping operation mentioned previously refers to the fact that in $\Delta\Sigma$ modulators the signal power of the noise (the discussion will now shift its focus on a frequency domain analysis) can be shaped or filtered. This approach allows for the noise to be processed in such a way such that its power is less concentrated in the band occupied by the signal of interest, and is shifted over to a region unoccupied by the signal. Given that the noise has now been shifted out of band, a filter can be used at the output of the $\Delta\Sigma$ modulator to recover the original analog signal with minimal interference from the quantization noise.

In order to better appreciate the advantages that this conversion technique possesses, the spectral characteristics of a conventional quantizer must first be examined. Figure 2.2(a) represents the Fast Fourier Transform (FFT) of a quantized sine wave. A sin-

gle tone representing the input signal is evident amongst the quantization noise floor. From the plot it is evident that the spectral components of the noise are spread out over a large band, including the region that contains the signal of interest. Figure 2.2(b) represents the FFT of a sine wave that has been processed by a one - bit lowpass ΔΣ modulator. From the plot it is evident that the input signal lies in a band relatively unaffected by noise, and that the noise is predominately concentrated in the high frequency range. By applying a lowpass filter to the output of the ΔΣ modulator, a relatively accurate replica of the input signal can be recovered.

The manner in which the signal and the noise are affected by the circuit is determined by the structure of the modulator. Specifically, a Noise Transfer Function (*NTF*), which describes the behavior of the noise in the system, and a Signal Transfer Function (*STF*), which describes the behavior of the input signal in the system, can be derived for each modulator. Although ΔΣ modulation devices are highly non-linear due to the presence of a quantizer in the ADC, linear models that describe its behavior can be obtained by accurately

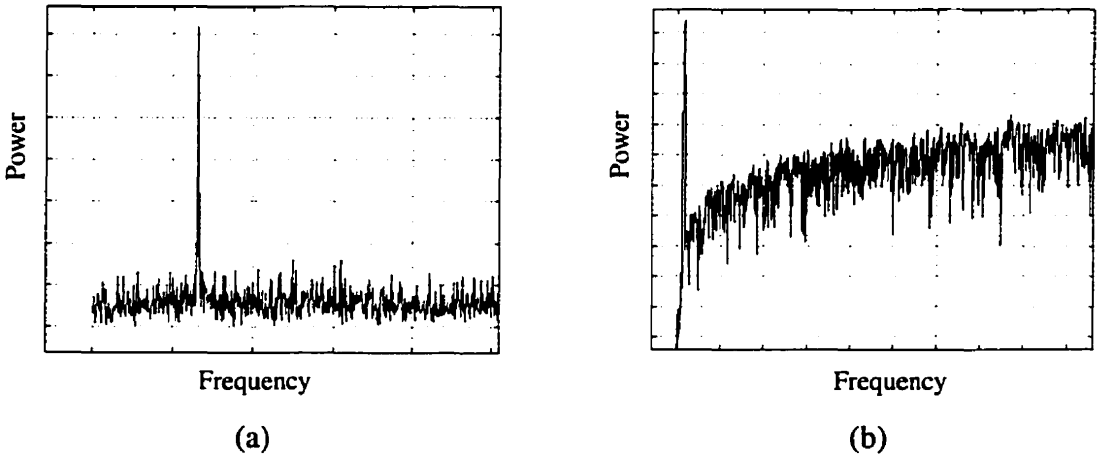


Figure 2.2: Frequency Response of Quantized (a) Sine wave (b) ΔΣ PDM Stream

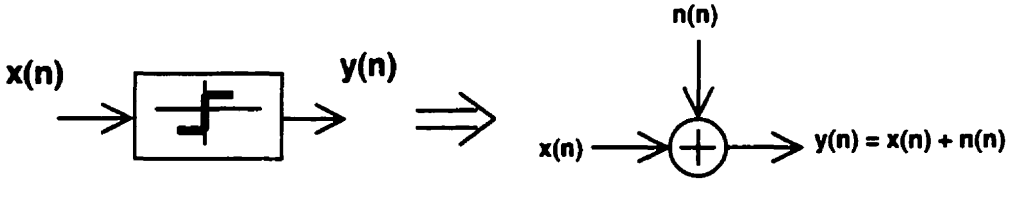


Figure 2.3: Linearized Model of the Quantizer

modeling it as a linear device [33], and can be seen in Figure 2.3. In essence a quantizer takes the input signal and superimposes noise. Therefore, it can be modeled as an adder that computes the summation of the input analog sample, and a noise component.

Figure 2.4 represents the conceptual schematic of the ΔΣ modulator after the quantizer has been replaced with its equivalent linear model. From this, two transfer functions can be derived. The *STF* which represents how the input signal will be affected, and the *NTF* which will determine in which regions the quantization noise will be suppressed or amplified. For the particular organization of the modulator presented in Figure 4, the *STF* and the NTF are given by

$$STF = 1 \tag{2.1}$$

$$NTF = \frac{1}{[1 + H(z)]} \tag{2.2}$$

For this particular structure, the input signal remains relatively unaffected by the modulator, while the noise is filtered by the *NTF*. By altering the characteristics of *H(z)* (the loop filter), different noise shaping characteristics can be attained. This gives designers flexibility in terms of which bands the noise can occupy, and the issue of designing a ΔΣ modulator breaks down to a filter design problem. There are of course other issues that designers must keep in mind, such as stability criteria and practical implementation issues.

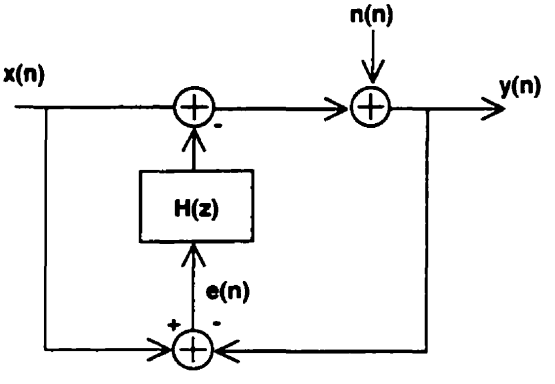


Figure 2.4: Schematic Model of ΔΣ Modulator with Linearized Quantizer Model

None the less, the design of the loop filter predominates since it dictates the *NTF*, and in some cases the *STF*.

2.2 Designing the Signal and Noise Transfer Functions

For low voltage low power applications, a single loop ΔΣ modulator topology is preferable over a cascaded one. This is because aside from the first stage, it has more relaxed design requirements on the amplifiers used. In order to achieve the desired performance, a third order LDI topology is selected, and can be seen in Figure 2.5. The LDI topology contains feed forward paths from the input, as well as internal feedback. The presence of many signal branch paths in the structure provides greater design flexibility concerning the characteristics of both the signal and noise transfer functions that can be produced. Merely altering some of the gain coefficients or eliminating various feedback paths can dramatically affect its signal processing characteristics. This allows for a certain degree of flexibility in its design. The intention is to acquire through various optimization algorithms, appropriate coefficients needed for the implementation of this topology as a lowpass ΔΣ modulator. It is important to note that the topology does not force the *STF* to be unity, and as a result, this fact must be accounted for in the design process.

The STF is represented by a transfer function that is composed of a combination of the circuit coefficients, which makes deriving the *STF*, as well as the *NTF*, a complex task. As a result, the Intermediary Function Analysis Program (IFAP) [34] and Maple, two CAD tools, were used to derive symbolic equations describing the transfer functions.

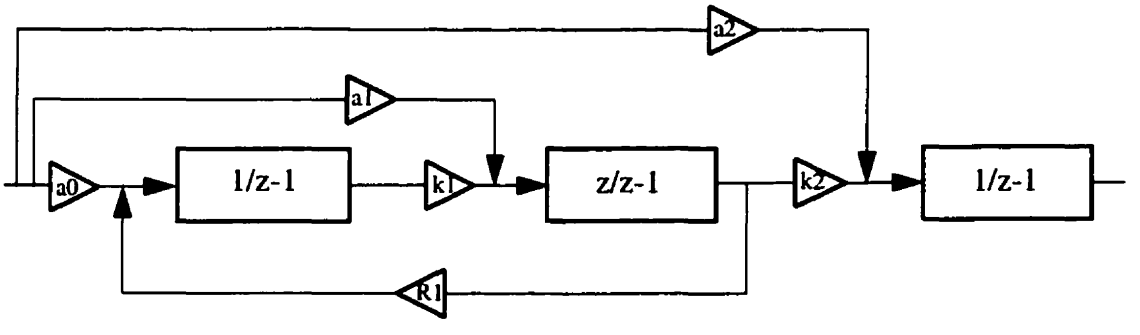


Figure 2.5: Signal flow Graph of 3rd Order LDI Topology

IFAP parses a hardware description of a circuit, and produces a Maple file that sets up a matrix describing a set of linear of equations representing it in a mathematical form. This file is then compiled in Maple, and the *STF* and *NTF* are produced in symbolic form. Figure 2.6 represents a block diagram of the process required to obtain the *STF* and *NTF*.

The signal and noise transfer functions expressed in terms of circuit coefficients, were evaluated in the aforementioned manner, and are presented as follows:

$$STF = \frac{(a2 + a1k2)z^2 + (-a1k2 - a2k1R1 - 2a2 + a0k1k2)z + a2}{z^3 - (b1k2 + k1R1 + b2 + 3)z^2 + (k1R1 + b1k2 + 2b2 + b2k1R1 - b0k1k2 + 3)z - (b2 + 1)} \quad (2.3)$$

$$NTF = \frac{z^3 - (k1R1 + 3)z^2 + (k1R1 + 3)z - 1}{z^3 - (b1k2 + k1R1 + b2 + 3)z^2 + (k1R1 + b1k2 + 2b2 + b2k1R1 - b0k1k2 + 3)z - (b2 + 1)} \quad (2.4)$$

It is important to note that the denominators of the *NTF* and *STF* are identical, indicating that both functions share the same poles. This fact places a large restriction on the design process, as it is necessary to derive both the *STF* and *NTF* such that they share the same denominator.

Given the existence of the symbolic *STF* and *NTF*, a design strategy must be established to allow for the selection of numerical values of those coefficients such that the *NTF* and *STF* exhibit the characteristics of a lowpass $\Delta\Sigma$ modulator. In order to properly design these functions, a set of criteria must be established to determine what traits they must posses in order to produce the necessary results.

Since the goal is to produce a lowpass modulator, the *NTF* must be a highpass function in order to suppress the quantizer noise as much as possible in the lowpass region where

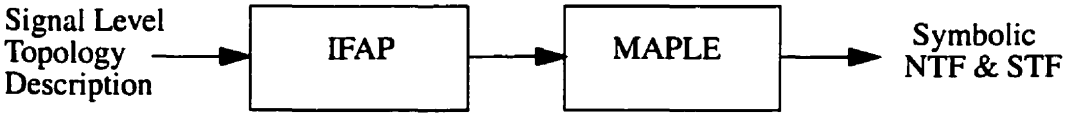


Figure 2.6: Methodology Employed to Derive the *NTF* & *STF*

the input signal is located. In contrast, the *STF* must be able to allow the relevant spectral components of the input signal to propagate through the circuit with minimal distortion and attenuation. Since the relevant spectral band is in the lowpass region, the *STF* can be chosen to resemble a lowpass filter. The behavior of the *STF* in any other region is not of much concern in the design process, since no input signal components will lie outside the lowpass region. Figure 2.7 displays the proposed characteristics for both the *STF* and *NTF*.

Given that the general form of the transfer functions have been proposed, more specific restrictions must be placed upon them in order to ensure proper operation of the ΔΣ modulator, and to maintain the integrity of the applied signal. The proposed solution involves designing the transfer functions, and then mapping them onto the topology. This is advantageous because it allows for the design of an *STF* that can be mapped onto any topology. This in essence breaks down the design procedure into two clearly distinct problems that can be solved, which include the transfer function design, and obtaining the circuit coefficients.

2.2.1 NTF Design

The pole - zero plot presented in Figure 2.8 summarizes the constraints enforced on the *NTF*, and should be used as a graphical reference for each of the following design criteria:

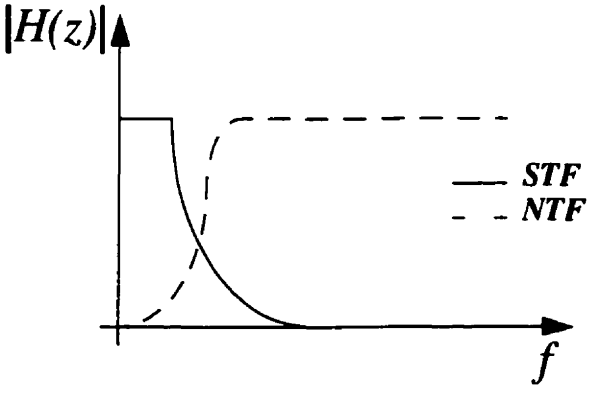


Figure 2.7: Proposed *NTF* and *STF* characteristics

i) Conjugate pole / zero criteria:

In order to eventually obtain the circuit coefficients in a realizable form, it is necessary that the poles and zeros of the *NTF* appear in conjugate pairs. If not, it would necessitate the use of imaginary circuit coefficients, which would obviously be unrealizable.

ii) Stability criteria:

In order to have a stable circuit, it is necessary that the poles of the system be present inside the unit circle. Since the design will make use of an optimization algorithm, it will terminate when a certain degree of precision or tolerance is reached. If that tolerance is too large, it is conceivable that the poles could be placed outside the unit circle without violating the constraint. As a result, it was necessary to build in a safeguard to prevent the optimizer from misplacing the poles. Therefore, the poles are restricted to fall within a circle of radius 0.98. This serves the dual purpose of not allowing the optimizer to misplace the poles, and also, it prevents the poles from being placed on or outside the unit circle. If the poles were on the unit circle, any small change in the circuit conditions could result in instability, by shifting the poles slightly outside the unit circle.

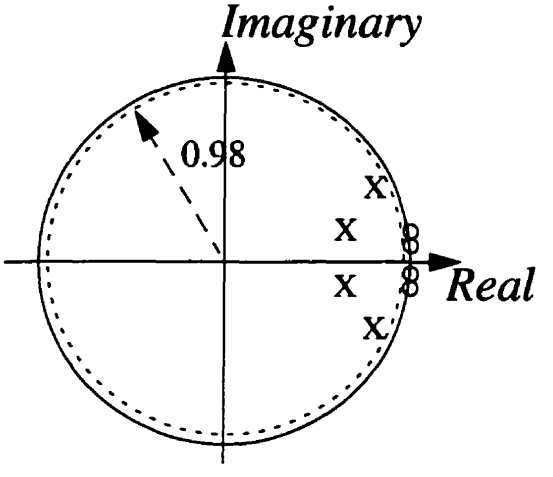


Figure 2.8: *NTF* criteria P/Z plot

iii)NTF bound criteria

A requirement for stability which is absolutely necessary is that the magnitude of the *NTF* must be less than 2.0 at every frequency [35]. As a result, this bound was integrated as a constraint in the *NTF* design procedure. However, there is no guarantee that the modulator will be stable just because this constraint is adhered to. Therefore, this bound was lowered to 1.6 to avoid problems.

iv) Zeros on unit circle criteria

Another constraint imposed, was that the *NTF* zeros be placed on the unit circle. This is desirable because it provides maximum noise attenuation in the signal passband region.

Computer aided design software can serve as powerful tools that provide designers with a means of improving design procedures and efficiency. One such tool that aids in the design of $\Delta\Sigma$ modulators is the Delta Sigma Modulator and Oscillator Designer (DSMOD) [36], which produces an *NTF* that is a compromise between SNR and stability, and adheres to the conditions imposed above. It solves for the *NTF* based on the model presented in Figure 2.4.

The *NTF* designed in DSMOD will be used as the starting point for the *NTF* eventually created for the modulator under construction. Since the design of the *STF* and *NTF* are not independent of one another, some flexibility has to be left for altering the *NTF* in order to aid in the design of the *STF*. In Figure 2.9, the DSMOD report on the design of the *NTF* can be seen, and will be used as initial conditions for the real *NTF* design.

In Figure 2.10, the ideal *NTF* plots that are output by the program, including the pole zero plot and a close-up of the passband can be seen. The three complex conjugate poles and zeros can clearly be seen in this plot (Figure 2.10(c)), and the *NTF* magnitude response (Figure 2.10(a)) directly demonstrates the low frequency attenuation that the noise undergoes.

NTF DESIGN REPORT

File Name:
Description:

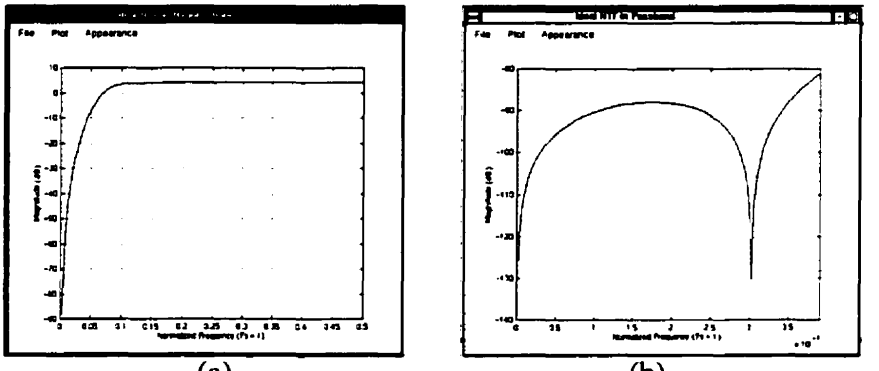
Specifications

Order:	3
Passband Type:	Lowpass
Oversampling Ratio:	128.000
Passband Edge (Hz):	0.003906
Transfer Function Type:	Butterworth + Optimal Zeros (Schreier)
Bound on NTF:	1.60000
Maximum NTF Magnitude:	1.59875
Avg. Inband Power Att.:	89.472
Expected SNR (estimate):	113.533

Power of Z	Numerator of NTF	Denominator of NTF
3	1.00000e+00	1.00000e+00
2	-2.99964e+00	-2.07743e+00
1	2.99964e+00	1.53492e+00
0	-1.00000e+00	-3.91095e-01

Zeros of NTF		Poles of NTF	
Magnitude	Phase	Magnitude	Phase
1.00000	0.00	0.79607	23.48
1.00000	-1.09	0.79607	-23.48
1.00000	1.09	0.61713	0.00

Figure 2.9: DSMOD Report on *NTF* Design



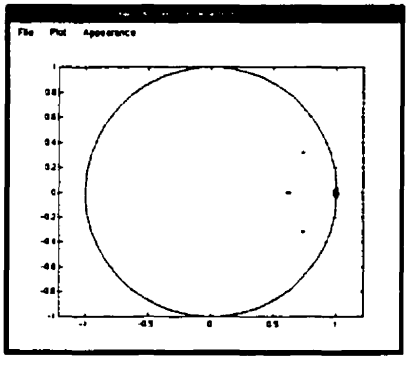


Figure 2.10: DSMOD Designed (a) Ideal *NTF* (b) Ideal *NTF* Passband (c) Pole Zero Plot

Although DSMOD provides powerful features that allow for the design of ΔΣ modulator noise transfer functions, it lacks the ability to design signal transfer functions. This limitation is a direct result of the theoretical model it utilizes to represent these modulators, which assumes that the signal transfer function is equivalent to unity. As a result, it is restricted to designing modulators with certain fixed topologies. Other topologies do not necessarily have an *STF* equivalent to unity, and in order to convert them into the form that DSMOD uses, it would be necessary to use an extra summer in most circuits, which effectively requires the presence of an extra operational amplifier. Most designers find this impractical, and wish to implement the circuit without the use of this extra component. In order to achieve this, an *STF* must be designed.

2.2.2 The STF Design

The schematic presented in Figure 2.11 represents the logical flow of operations required to implement a *STF* solving algorithm. In this instance, the pole and zero locations of a pre-designed *NTF* were obtained from DSMOD and used as initial conditions in the optimization procedure. The pole and zero locations of the *NTF* as well as the *STF* were manipulated under the following conditions: the *NTF* should closely resemble the original *NTF* designed by DSMOD while adhering to the criteria established in section 2.2.1, and the *STF* should abide by restrictions that are presented later in this section. This program then solves for the *NTF* and *STF* poles and zeros simultaneously. Once the optimized pole and zero locations are obtained, they are converted into transfer function form through a multiplicative process. The numerical *STF* and *NTF* obtained from the optimizer are then

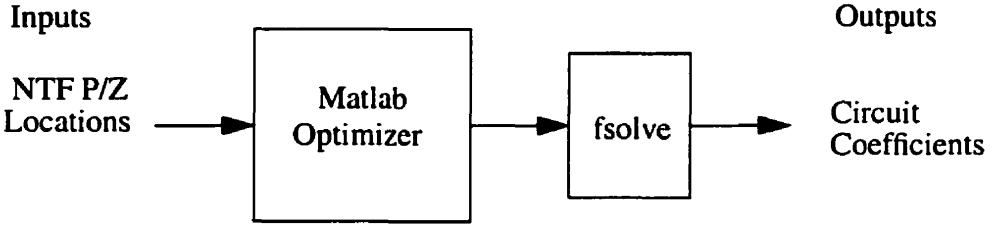


Figure 2.11: Pole / Zero Optimization Methodology

equated to the symbolic forms of the equations that are expressed in terms of the circuit coefficients. As such, a set of non-linear equations result. When solved, these equations will provide for the circuit coefficients. This task was accomplished through the use of *fsolve*, another Matlab optimization routine that employs numerical algorithms to solve sets of non-linear equations. The following is a list of criteria that were used in the optimization process for the purpose of designing the *STF*.

i) Conjugate criteria

The *STF* zeros must appear in conjugate pairs. The reason for this is identical to the argument presented previously for the *NTF* poles and zeros.

ii) Unity gain in passband

Another condition placed on the *STF* forces its mean to be 0 dB in the passband region. This provides the *STF* with unity gain in the passband region, and ensures that the essential signal components are not attenuated.

The method of error determination for the optimization routine consists of a combination of three factors which along with the two constraints discussed above determine the optimization strategy. The three factors of error determination are the ripple in the *STF* passband, linear phase in the passband of the *STF*, and the distance that the optimized poles are from the poles determined by DSMOD.

iii)Minimize STF passband ripple

In order to help preserve the input signal, it is important that the gain of the *STF* be constant in the passband region. Although this is not possible to achieve since a ripple is always present, the problem now becomes to minimize this ripple, in order to achieve a passband region that is as flat as possible. The approach used to set up this condition in the optimizer, was to minimize the maximum ripple number, defined as the maximum gain divided by the minimum gain in the passband region.

iv) Linear phase in passband of STF

Phase distortion (which is the departure from linear phase) is an extremely difficult error to compensate for. Therefore, it is necessary to guarantee that the signal will remain the same no matter where it is transmitted inside the passband. As a result, it was necessary to formulate an error condition to minimize the effects of phase distortion on the input signal. This condition was formulated in the optimizer by working with the group delay. The group delay is a measurement of the rate of change of phase with respect to frequency. It is desired that this quantity be constant to ensure linear phase in the passband region. Furthermore, if it is desired that the group delay be constant, it follows that the rate of change of group delay with respect to frequency should be zero. Therefore, the error condition setup in the optimizer specifies that the rate of change of group delay with respect to frequency be minimized.

v) System Poles and their Proximity to the DSMOD Poles

One of the goals of this algorithm was to maintain the *NTF* designed by DSMOD. As a result, the third error condition that was implemented strived to minimize the distance between the optimizer designed poles, and the poles designed by DSMOD.

After creating and running the routine, the *NTF* and *STF* transfer functions were mathematically found and can be seen as follows:

$$NTF = \frac{z^3 - 2.99964z^2 + 2.99964z - 1}{z^3 - 2.07743z^2 + 1.53492z - 0.391095} \tag{2.5}$$

$$STF = \frac{0.0175278z^2 + 0.033176z + 0.0156986}{z^3 - 2.07743z^2 + 1.53492z - 0.391095} \tag{2.6}$$

It is important to note that the *NTF* was not yet forced to change from the DSMOD ideal.

The *fsolve* portion of the routine was then used to map the transfer functions into the LDI circuit coefficients seen in Figure 2.12, which illustrates the system level representation of the LDI topology.

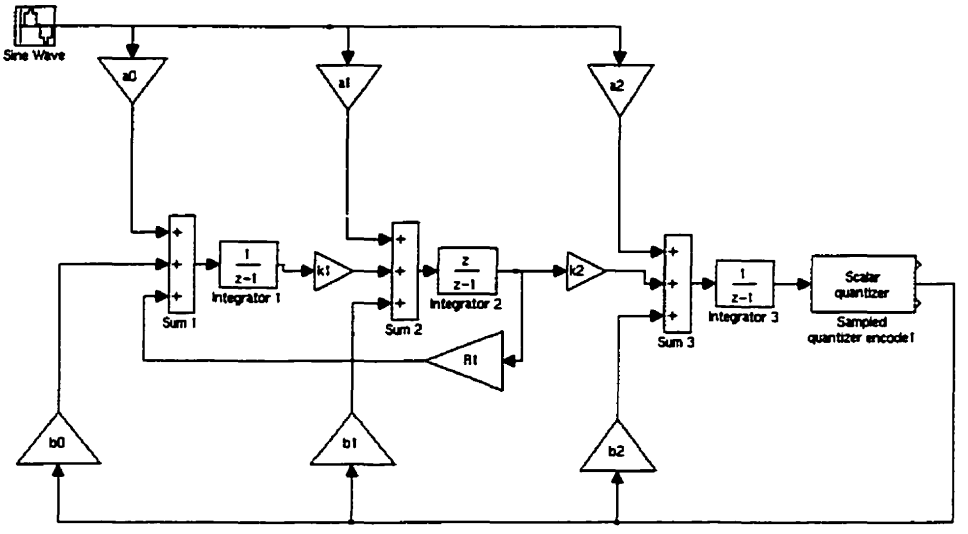


Figure 2.12: Mapping onto the LDI Topology

2.3 Designing with Simulink

Once the mapping had taken place, the system was simulated using Simulink, and it was verified to be stable. The next task was to manipulate the coefficients. In order to obtain an easily implementable system, ideally one would prefer coefficients that are rational fractions. This allows for very simple common centroid layout, which improves the accuracy of matching.

2.3.1 Dynamic Range Scaling and Coefficient Modification

Dynamic range scaling is an essential step in choosing the coefficients of the modulator. In order to avoid distortion due to the clipping of one of the amplifiers, it was necessary to modify the coefficients so that the outputs of all of the integrators peak at the same level as that of the overall *STF*. To do so, the spectral peaks of all of the integrators were found using Simulink, and dynamic range scaling was performed in the manner described in [37]. The result was that the coefficients **a0**, **a1**, **a2** and **R1** (which can be seen in Figure 2.12) were very small.

Since the coefficients represent the capacitor ratios that will be implemented in the modulator (i.e.- $\mathbf{a0} = C_{a0}/C_{i1}$), a large ratio is undesirable for two reasons:

1. It is harder to match capacitors whose values are spread far apart in magnitude. For the worst case **a** and the **R1** coefficients above, the spread was 145 and 1000 respectively.

2. A large ratio necessitates the use of a huge capacitor which can be impractical due to area considerations. In the case of this modulator, the unit size capacitor used was 250 fF. If a ratio of 1000 is needed, a 250 pF capacitor would be required, and its presence would use up the entire area allocated for the design of the modulator.

It is important to consider that the *STF* can be designed with many restrictions. However, the more restrictive the *STF* is, the larger the amount of total capacitance is needed. This is due to the huge capacitor spread involved in the designed *STF*. Due to this, a re-evaluation of the design method was necessitated.

After some examination, it was noticed that the **R1** coefficient controls the spread of the zeros in the *NTF*. In order to stagger the zeros across the signal passband, an extremely small **R** is needed. Practically this translates into a huge total capacitance. In order to remove this problem, the coefficient was set to zero, thereby collapsing all of the zeros of the *NTF* to DC. The removal of the **R** coefficient more than halved the total capacitance needed, reducing it to 780 units of capacitance. It is important to note that this did nothing to solve the large spread in the **a** coefficients.

It was determined that the huge capacitor spreads in the **a** coefficients were due to the placement of the *STF* zeros. In order to get excellent phase characteristics, the *STF* zeros need to be placed at high frequencies, which in turn necessitated the feedforward coefficients to be small. As a result, the *STF* zeros were moved in from high frequencies towards DC, where the trade-off in phase versus capacitor spread was examined.

An optimization routine in Matlab was employed in order to allow at most 1 degree of deviation from linear phase in the *STF*, for the purpose of reducing the capacitor spread. Following the optimization, the non-linear equations resulting due to the mapping of the

Coefficient	Ratio
a0	1/4
a1	1/8
a2	0
k1	3/8
k2	5/8
b0	1/4
b1	4/8
b2	5/8
R1	0

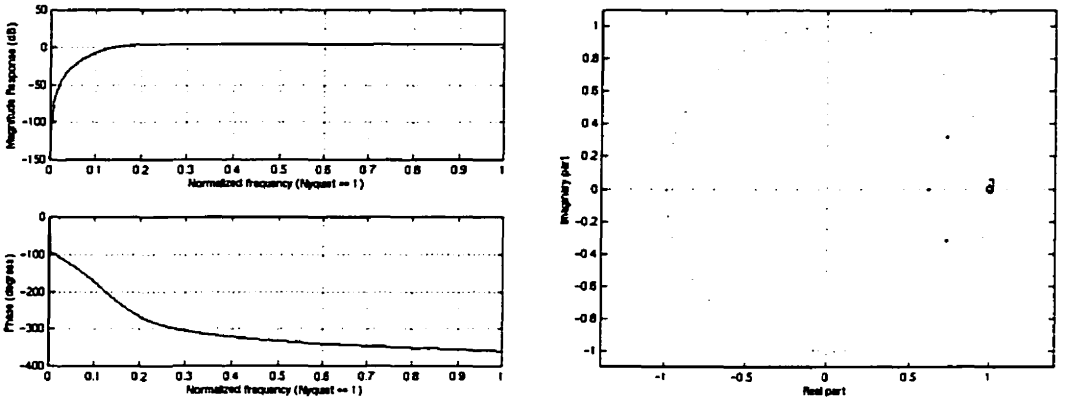
Table 2.1: Absolute Value of Coefficients used in Modulator

transfer function onto the coefficients of the topology, were solved in a semi-automatic manner such that the size of the coefficients could be guided through the use of initial conditions. The result was that the total amount of capacitance was significantly reduced to 71 units, and the capacitor spread was reduced to 8. The absolute values of the coefficients used can be seen in Table 2.1. All of the ratios are rational fractions whose denominators are powers of 2. This lends itself to a fairly simple and effective common centroid capacitor layout strategy which will be discussed in section 4.3.1.

In order to achieve easily implementable coefficients, some trade-offs in performance were made. For example, the coefficient 3/8 was used instead of the value 0.368 at an expense of 2 dB of dynamic range.

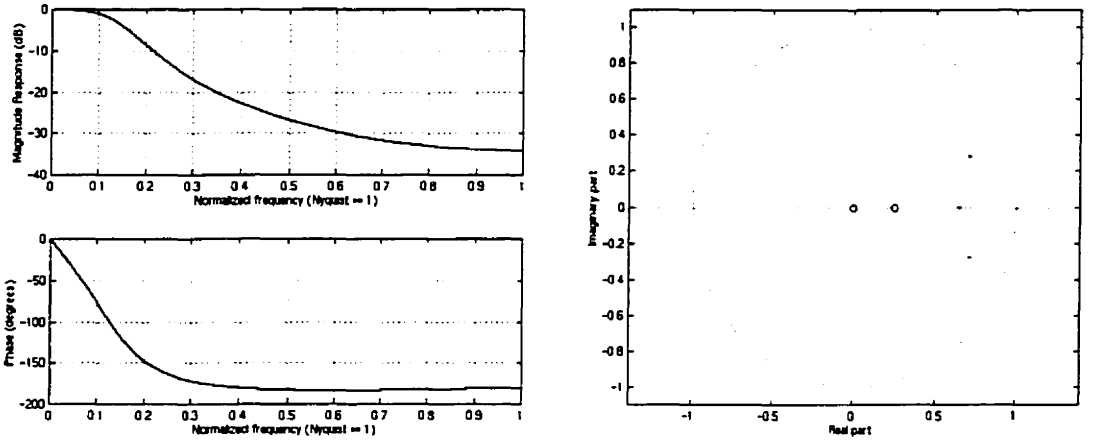
2.3.2 Simulink Extracted Transfer Functions

Following the determination of the coefficients, the noise and signal transfer functions were extracted from the Simulink system. The *NTF* transfer function can be seen in Figure 2.13(a), and its corresponding poles and zeros plot can be seen in Figure 2.13(b). Accompanying these figures are the *STF* transfer function (Figure 2.14(a)) and its poles and zeros plot (Figure 2.14(b)).



(a) (b)

Figure 2.13: Simulink Extracted *NTF* (a) Transfer Function (b) Pole/Zero Plot



(a) (b)

Figure 2.14: Simulink Extracted *STF* (a) Transfer Function (b) Pole/Zero Plot

If we examine the STF more closely, we can see that the system is approximately unity in the passband as seen in Figure 2.15. The STF appears to exhibit perfectly linear phase (Figure 2.16), however it has some deviation. The deviation in group delay, seen in Figure 2.17 was calculated to be 795 psec, which translates into a 6.87 mili-degree deviation from linear phase, which was deemed acceptable according to industrial standards [38]. As a result, we can see that the manipulation/truncation of the coefficients has still allowed us to preserve the important characteristics of both the *NTF* and *STF*.

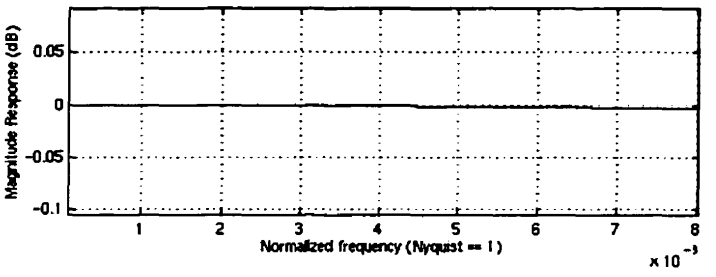


Figure 2.15: Magnitude of the *STF* in the Passband

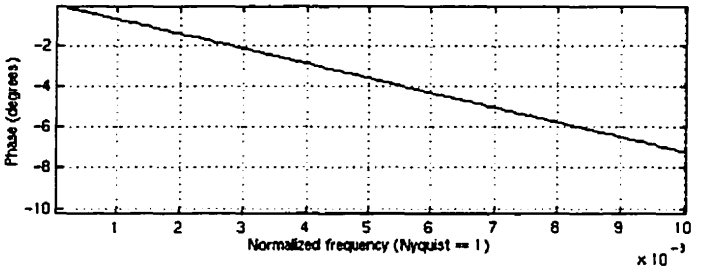


Figure 2.16: Linear Phase in the Passband of the *STF*

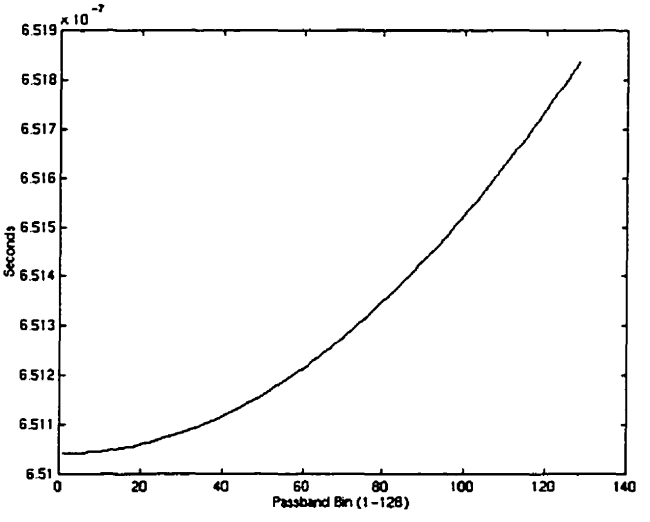


Figure 2.17: Simulink Extracted Group Delay in the Passband

2.3.3 Ideal Simulink Simulation Results

The system's signal flow graph representation, which was seen in Figure 2.12, was then simulated with its finalized ratios (Table 2.1) in Simulink. An audioband coherently sampled sinusoid [39] was input to the system, and the output was interpreted using a 2^{15} point FFT, and a Kaiser window with a β of 30 [40].

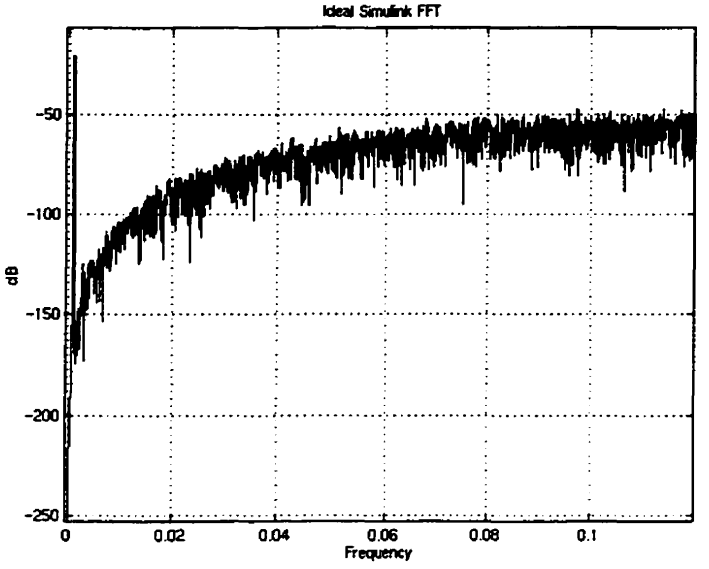
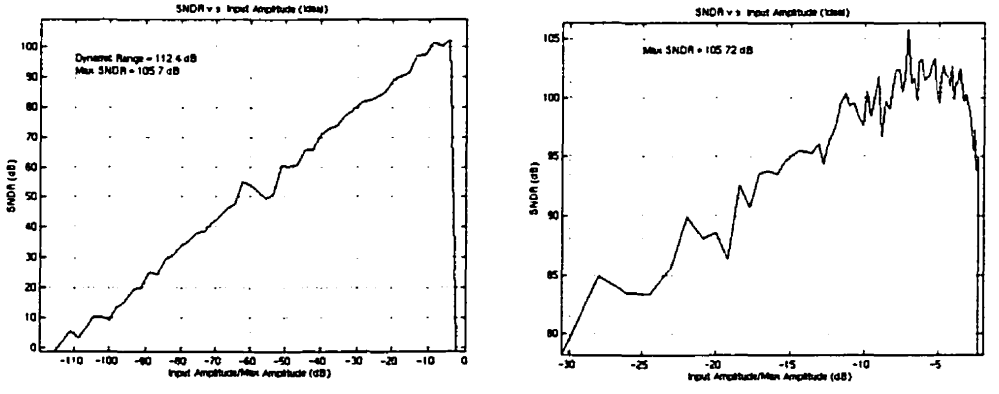


Figure 2.18: FFT of Ideal System in Simulink (60% at 5812.5 Hz)

A simulation was run with an input amplitude that is 60% of the rails at a frequency of 5812.5 Hz. The resulting SNDR was 101.8 dB, and the resulting FFT can be seen in Figure 2.18.The amplitude of the sinusoid input to the modulator was then swept, and an SNDR curve was obtained. In Figure 2.19(a) and Figure 2.19(b) below, we can see the SNDR plots of the modulator. It is important to note, that this is an simulation of the ideal system as no circuit component non-idealities have yet been considered.



(a) (b)

Figure 2.19: SNDR v.s. Input Power (Ideal System) (a) Dynamic Range (b) Close-up

From these plots one can note that the maximum SNDR is 105.72 dB at 44% of the maximum amplitude (1.25 volts). The dynamic range of the ideal system was found to lie between the input signal power levels of -2.9 dB and -114.9 dB, which translates into 112 dB, or 18.3 bits according to [41].

2.4 Conclusion

A method for designing and mapping of filter functions (NTF and STF) onto ΔΣ modulators was presented. Guidelines for designing both the STF and NTF were highlighted, and multiple CAD tools were used in order to obtain functions that adhere to them. Matlab optimization routines were then employed to map the filter functions onto a specific LDI topology. Simulink was then used to validate and verify correct operation.

CHAPTER 3 Top-Down Design Methodology For Analog Circuits Using Matlab And Simulink

This chapter presents a new design methodology for the creation of analog or mixed signal integrated circuit components. Through the use of top-down design techniques in conjunction with an optimization process, circuit design can take place at the highest level of abstraction. As a result, the requirements of the building blocks will be specified prior to the undertaking of transistor level simulations, thereby saving much valued design time. This method has the added advantage that designs can be implemented with currently used, and widely available tools. For illustration, a design example featuring a third order, switched capacitor delta-sigma modulator will be presented.

3.1 Design Methodology Motivation

Choosing the specifications for an analog circuit given a set of design objectives such as minimal area and power, is potentially a very complicated and time consuming process. This task is further complicated by the fact that the number of specifications to deal with is usually very large, and vary over wide ranges from one application to another.

One of the main drawbacks to analog design is the uncertainty that arises due to a change in technology. As a result, it is more amenable to consider a design process that can

begin without a complete dependence on a specific technology. This would allow for a large portion of the design to be carried out independently of the technology, and could lead to a high degree of reusability. High level optimization geared design avoids the reliance on a specific technology the longest, and in this case provides the designers with values for familiar parameters (for example g_m and r_{out} in the case of an Operational Transconductance amplifier or OTA). These are quantities that are very familiar and understood by designers and provide excellent guidelines for the construction of a device. Furthermore, symbolic analysis is a method by which designers can obtain an understanding of a circuit's behaviour. Therefore relating the behaviour of the circuit to several descriptive equations or expressible rules of thumb, and moreover to a few key parameters, can reduce the complexity of a problem and lends itself to a solution through optimization.

The main focus of this chapter is to provide designers with a means of tackling the design problem by presenting a simple to implement design methodology that involves widely used and available tools. As a result, the procedures can be implemented and reused with little difficulty or expense. In order to make use of widely available tools which are already known to many designers [42, 43], Simulink is used to implement the system architecture and model non-idealities, while the Matlab optimization toolbox [44] is used to create routines to optimize the circuit parameters.

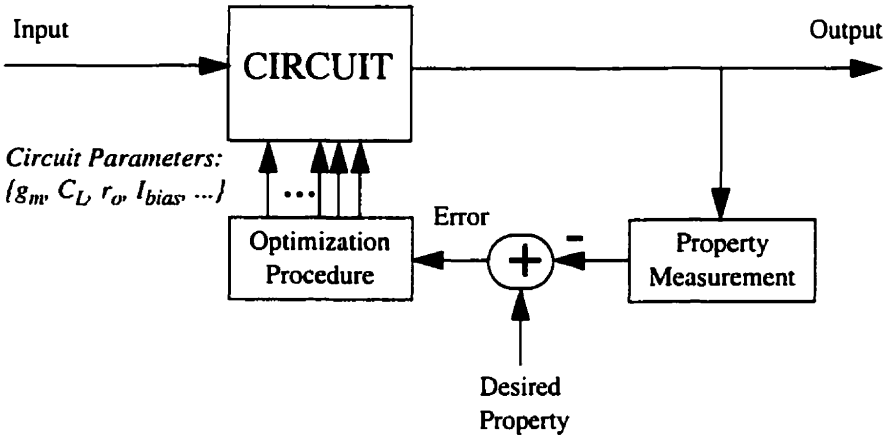


Figure 3.1: Optimization Procedure

3.1.1 Optimization Procedure

Optimization in general concerns the minimization or maximization of a function. In the case of analog systems, the objective is to maximize or minimize a given set of performance criterion, such as the Signal to Noise and Distortion Ratio (SNDR), power, area or settling time. To do so, one must establish a trackable set of variables (g_m, r_o, I_{bias} or C_{load}) that can be used to model the system while simultaneously providing a link to the performance criterion. An illustrative flow of this procedure can be seen in Figure 3.1. It consists of four main components: a property measurement block, a summing block with the desired property as one input, an optimization procedure, and finally the circuit that is under development. The property measurement block detects particular attributes of the output signal from the circuit and compares it with the desired behaviour. The error signal is then used to adjust the key parameters of the circuit such that the error is minimized. It is using this very simple principle that the proposed top-down methodology is executed.

Due to its ability to model nonlinear devices, there already exists much work in a variety of fields, which use Simulink modeling, and as such it is not difficult to find pre-existing models, or to create new models specific to a brand new design [45-49]. In order to best explain the procedure, the methodology will be explained using a Delta-Sigma ($\Delta\Sigma$) analog-to-digital converter (ADC). It is important to realize that this methodology is not limited to $\Delta\Sigma$ modulators. It may be used to help design any device or system, in which key parameters, formulas, or rules of thumb can be identified.

3.2 Switched Capacitor Delta-Sigma Design Procedure

$\Delta\Sigma$ ADCs are widely used and well suited for high resolution conversion. One dominant factor in their popularity stems from their high tolerance for component mismatch and circuit non-idealities. Despite its high tolerance for non-idealities, the $\Delta\Sigma$ modulator is still governed by the limitations of its analog building blocks. In particular, switched capacitor (SC) $\Delta\Sigma$ modulators are sensitive to circuit non-idealities at the input stage where no noise-shaping has yet taken place [50]. Specifically, sampled capacitor (kT/C) noise, switch on-

resistance, and OTA characteristics (noise, clipping, finite gain, finite bandwidth, and slew rate) at the front end limit the achievable dynamic range and therefore performance. A high level diagram of a Simulink realization of a 3rd Order ΔΣ single loop modulator can be seen in Figure 3.2. It is important to realize that upon implementation, each component block has to be translated into its circuit equivalent (analog building blocks) and then implemented.

In order to use an optimization procedure, it is first necessary to isolate the key parameters that influence the system's performance. In the design of high resolution SC ΔΣ modulators, it can appear that there is a large set of parameters to optimize. However, if one examines the building blocks of the system, it can be seen that most of the factors that affect the system's performance (kT/C noise, switch on-resistance, OTA noise, voltage clipping due to a finite output range, finite gain, finite bandwidth, and slew rate) can be related to a small set of parameters. These parameters will now be explored, and since ΔΣ modulators are sensitive to circuit non-idealities at the input stage (the first integrator seen in Figure 3.2) where no noise-shaping has yet taken place, this section of the modulator will be the focus of the analysis.

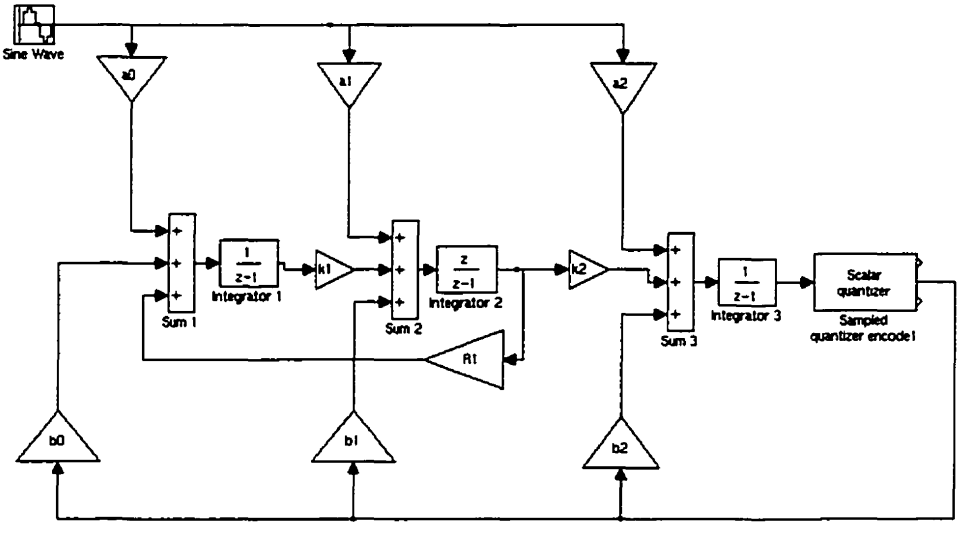


Figure 3.2: High Level Diagram of the Modulator

3.2.1 Sampled Capacitor (kT/C) Noise and Switch On-Resistance

A critical source of noise, is the kT/C noise injected into the first integrator of the modulator because it directly increases the noise level at the output. The kT/C noise component results from the sampled noise stored on the capacitor due to switch on-resistances. As a result, the input capacitors must be large enough to counter this effect. Therefore the first key parameter, which can be seen in Figure 3.3, is the input sampling capacitor C_S.

The presence of the on-resistance in a switch is also a source of distortion in SC ΔΣ modulator implementations. As with the kT/C sampling noise, the distortion caused by the switches has its most predominant effect in the first stage of the modulator. This distortion can be quite significant if not properly accounted for, as a result the maximum on-resistance allowable, R_{MAX}, is also a key parameter.

3.2.2 OTA Parameters

The OTA within the modulator is the most critical component, as the non-idealities that it exhibits causes an incomplete transfer of charge leading to non-linearities. By briefly examining this device, key parameters that govern its non-ideal behaviour can be isolated and examined. Shown in Table 3.1 are the corresponding non-idealities that must be considered in an OTA.

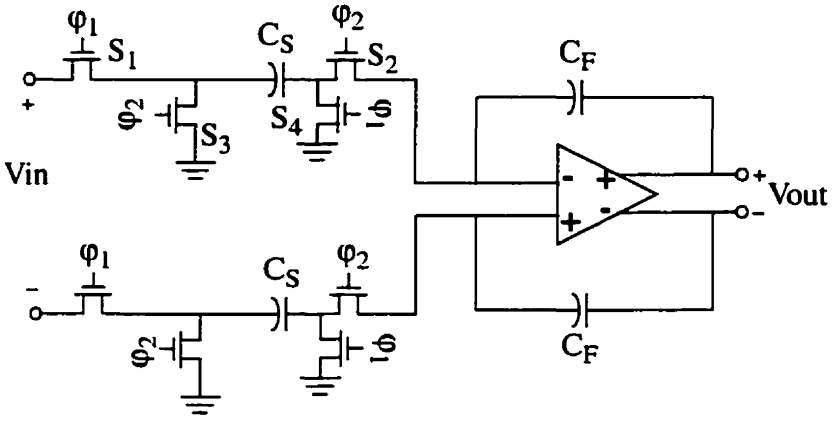


Figure 3.3: Switched Capacitor Implementation of the First Integrator

There are three OTA topologies commonly used in the design of ΔΣ modulators [51], the Folded Cascode amplifier (Figure 3.4(a)), the Two-Stage Class A amplifier (Figure 3.4(b)), and the Two-Stage Class AB amplifier (Figure 3.4(c)). Table 3.2 highlights the key properties that govern the operational performance of these OTAs. These properties allow for behavioural modeling of the device, and more importantly, can be dissected to obtain the key parameters of the device. It is important to note that r_{o1} and r_{o2} refer to the output resistance of the first and second stages. We can see from this table, that the main sources for non-idealities in the modulator can be related to a few key OTA parameters (g_m, r_o, C, and I_{bias}). As a result, the key parameters for each of the OTA topologies that can be used in the modeling of each structure are listed in Table 3.3.

Non-ideality	**Result**
Finite DC Gain	Rise in quantization noise
Saturation	Possible harmonic distortion
Bandwidth Limiting	Noise due to incomplete settling
Slew Rate Limiting	Noise due to incomplete settling and harmonic distortion
Thermal Noise	Added white noise

Table 3.1: OTA Non-idealities

Property	**Folded Cascode**	**Two-Stage Class A**	**Two-Stage Class AB**
DC Gain	$g_{m5}*r_o$	$g_{m1}*g_{m6}*r_{o1}*r_{o2}$	$g_{m5}*r_{o1}$
Unity Gain Frequency	g_{m5}/C_L	g_{m1}/C_C	g_{m5}/C_C
Slew Rate	I/C_L	$\min\{I_{bias}/C_C, I/(C_L+C_C)\}$	I_{bias}/C_C

Table 3.2: Key Properties in OTAs

Folded Cascode	**Two-Stage Class A**	**Two-Stage Class AB**
g_{m5}, r_o, C_L, I_{bias}, C_S	g_{m1}, r_{o1}, C_L, C_C, I_{bias}, I, C_S	g_{m5}, r_{o1}, C_C, I_{bias}, C_S

Table 3.3: Key Parameters for Modeling

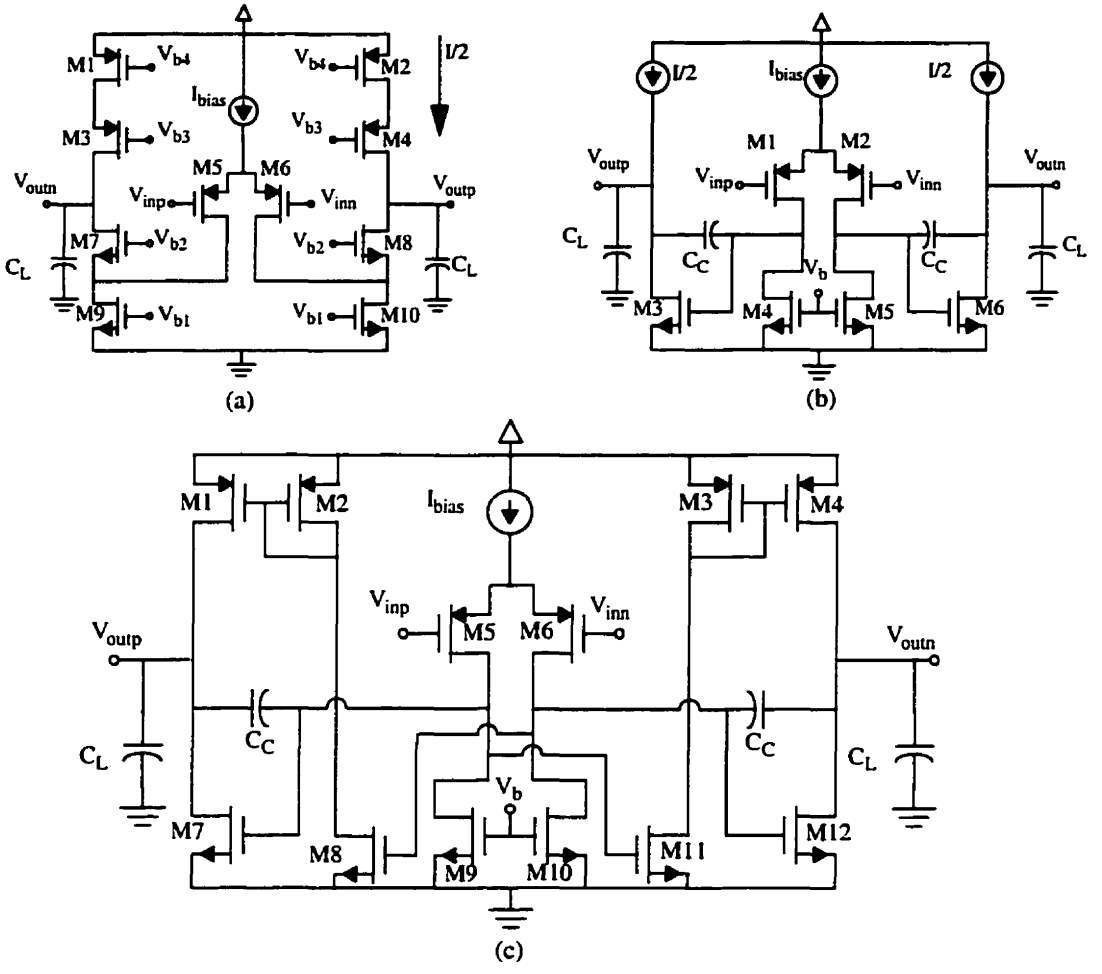


Figure 3.4: OTA Topologies: (a) Folded Cascode (b) Two-Stage Class A (c)Two-Stage Class AB.

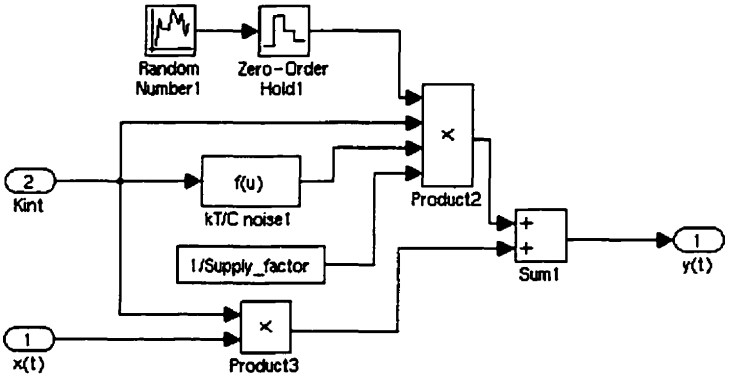


Figure 3.5: kT/C Noise Model in Simulink

3.3 Modeling of ΔΣ Modulators in Simulink

In order to perform behavioural simulations of ΔΣ modulators while taking into account most of the non-idealities, it is necessary to create models for them. A set of models have been proposed that attempt to simulate the non-idealities in Simulink [49]. The following work is intended as modifications and extensions to those models for the purpose of providing reference to the key parameters highlighted in Table 3.3, and to make them more suitable for an optimization environment.

3.3.1 Sampled Capacitor (kT/C) Noise

The thermal noise associated with the switching due to input sampling onto the first integrator can be modeled as follows:

$$y(t) = k_{int} \cdot [x(t) + n_{switch}(t)] \tag{3.1}$$

where

$$n_{switch}(t) = \sqrt{\frac{2kT}{C_S}} \cdot RN(t), \tag{3.2}$$

k_{int} is the integrator gain, C_S is the input sampling capacitor, k is Boltzman's constant, T is temperature, and $RN(t)$ is a Gaussian random number with zero mean and unity standard deviation. This has been implemented in a Simulink diagram shown in Figure 3.5. It is important to note that the formula used in the *f(u)* block with its input k_{int}, implements the $n_{switch}(t)$ formula. It is important to note that a 2 appears in the equation for $n_{switch}(t)$. This is due to the fact that the capacitor is sampled twice in the SC circuit (once of each of the two phases). A *Supply_factor* block can also be seen in the diagram. This is present for Simulink simulations in which it is desired to normalize the supply voltage. If no such normalization is desired, then *Supply_factor* should be set to 1.

3.3.2 Distortion Due to Switches

In Figure 3.3, a SC integrator was presented to illustrate the input stage of the ΔΣ modulator. On ϕ_1, one can see that the input is sampled through switches S_1 and S_4 onto capac-

itor C_S. However, since the on-resistance is dependent on the voltage across the switch, it can be seen that the resistance of S_1 depends on the time varying input, and therefore becomes a source of distortion. The resistance of switch S_4 has one terminal fixed to a constant voltage, and its source and drain experience a very small voltage difference, making it a lesser source of distortion. The same holds true for switches S_2 and S_3 on ϕ_2, which also have one end fixed to a constant voltage. As a result, the distortion mechanism introduced through S_1 and S_4 will be the focus of the analysis, and its circuit model representation is illustrated in Figure 3.6.

In order to model the effects of the distortion created by these switches, one could solve the following differential equations which are based on the ones presented in [52]:

$$\frac{d}{dt}v_o(t) = \frac{v_i(t) - v_a(t)}{C_S \cdot R_{s1}(v_i(t), v_a(t))} \tag{3.3}$$

and

$$\frac{v_i(t) - v_a(t)}{R_{s1}(v_i(t), v_a(t))} = \frac{v_a(t) - v_o(t) - V_{CM}}{R_{s4}}. \tag{3.4}$$

This set of equations can be numerically solved for each clock period, provided that the voltage dependent R_{s1} can be quantified.

The model in Figure 3.7 represents a way of implementing the above equations in Simulink. From that diagram, it can be seen that a look-up table is used in order to represent the on-resistance values of switch S_1 at different voltage levels. It can also be seen that a Matlab function is used. This function implements the solution to Eq. (3.3) and Eq. (3.4).

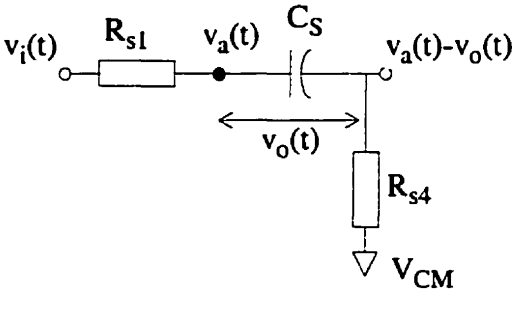


Figure 3.6: Model of Sampling Operation [52]

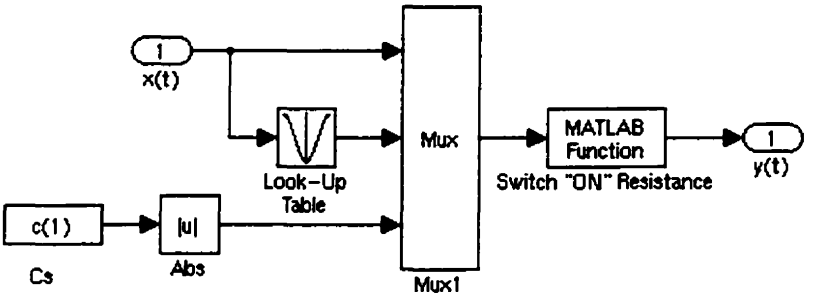


Figure 3.7: Switch Distortion Model Using Look-Up Table

The main drawback of this model is that a complex relationship for the on-resistance with respect to the input voltage must be derived or obtained. This can be done through simulations in Spice or through the use of appropriate equations [53]. This however may be impractical, and a simpler alternative was sought out.

As a result, a method for finding a bound on the Total Harmonic Distortion (THD) of the switch was developed out in order to predict the worst case distortion, and thereby provide a worst case bound for the on-resistance.

The distortion associated with the sampled capacitor network can be quantified by a THD measure. A procedure for doing this using a switched current integrator was presented in [54], and has been modified for use here. According to [54], a bound on the THD is given by

$$THD \leq \left[\frac{\left\| V_{OUT} - V_{OUT1} \right\|_2^2}{\left\| V_{OUT1} \right\|_2^2} \right]^{\frac{1}{2}}, \tag{3.5}$$

where *THD* refers to the distortion level relative to the first harmonic, V_{OUT1} represents the power of the first harmonic (the input signal), and V_{OUT} represents the total power associated with all of the harmonics present. It is important to note that $V_{OUT} - V_{OUT1} = \Delta V_{OUT}$.

In order to get a measure for these components, one needs to refer to the difference equation that describes the sampled capacitor network shown in Figure 3.6. The difference equation is given by

$$V_{OUT}(n) = (1-\gamma(n)) \cdot V_{IN}(n), \tag{3.6}$$

where $\gamma(n)$ represents the time varying settling error that occurs on the n^{th} sample, and is equal to $e^{\frac{-T}{2 \cdot \tau}}$, in which T is the clock period, and τ is the settling time constant of the switch.

The settling error $\gamma(n)$ can be divided into two components γ_{LIN} and $\Delta\gamma(n)$. These values represent a time invariant settling error and a time varying settling error respectively, such that

$$\gamma(n) = \gamma_{LIN} + \Delta\gamma(n). \tag{3.7}$$

Furthermore, the output voltage can also be divided into two components: a linear component V_{OUTl} which can be associated with γ_{LIN}, and ΔV_{OUT} which can be associated with $\Delta\gamma(n)$, meaning that

$$V_{OUT}(n) = V_{OUT1}(n) + \Delta V_{OUT}(n). \tag{3.8}$$

If this equation is substituted into Eq. (3.6), and decomposed into a time invariant linear equation and a time varying non-linear equation, the following relationships arise:

$$V_{OUT1}(n) = (1-\gamma_{LIN}) \cdot V_{IN}(n) \tag{3.9}$$

and

$$\Delta V_{OUT}(n) = -\Delta\gamma(n) \cdot V_{IN}(n). \tag{3.10}$$

In order to place an upper bound on the THD, it is necessary to ensure that Eq. (3.5) is maximized. This is done by maximizing ΔV_{OUT} and minimizing V_{OUTl}. If γ_{MAX} corresponds to the maximum settling error produced by the network, it follows that both γ_{LIN} and $\Delta\gamma(n)$ should be set to $\gamma_{MAX} = e^{\frac{-T}{2 \cdot \tau_{MAX}}}$, where τ_{MAX} represents the longest time constant possible.

Making these substitutions and taking the z-transforms of Eq. (3.9) and Eq. (3.10) results in the following transfer functions:

$$\frac{V_{OUT1}(z)}{V_{IN}(z)} = 1 - \gamma_{MAX} \tag{3.11}$$

and

$$\frac{\Delta V_{OUT}(z)}{V_{IN}(z)} = -\gamma_{MAX} \cdot \tag{3.12}$$

Assuming that the system is excited by a sine wave with amplitude A, it can be shown that the power associated with the output signal is $\frac{A^2}{2}$ [55].

As a result,

$$\|V_{OUT1}\|_2^2 = \frac{A^2}{2} \cdot (1 - \gamma_{MAX})^2, \tag{3.13}$$

$$\|\Delta V_{OUT}\|_2^2 = \frac{A^2}{2} \cdot (-\gamma_{MAX})^2, \tag{3.14}$$

and the bound on the THD can be given as

$$THD \leq \frac{\gamma_{MAX}}{1 - \gamma_{MAX}}. \tag{3.15}$$

Recalling attention to Figure 3.6, one can determine that $\tau_{MAX} = 2 \cdot R_{MAX} \cdot C_S$, meaning that the bound on the THD is given by

$$THD \leq \frac{e^{\frac{-T}{4 \cdot R_{MAX} \cdot C_S}}}{1 - e^{\frac{-T}{4 \cdot R_{MAX} \cdot C_S}}}, \tag{3.16}$$

where C_S is the sampling capacitance, and R_{MAX} is the maximum allowable on-resistance of the switch.

Since the system being modeled is differential in nature, the second harmonic is relatively small, therefore the dominant source of distortion will be the third harmonic [1]. As a result, the distortion bound on the switches found in Eq. (3.16) can be implemented by directly introducing a third harmonic into the simulink system as illustrated in Figure 3.8, and can be given by $THD = \frac{H_3|_{RMS}}{A/\sqrt{2}}$, where $H_3|_{RMS}$ is the RMS value of the third harmonic, and A is the system input sinewave amplitude.

From the diagram, a distortion generator and a gain block can be seen. The distortion generator is simply a sampled sinewave that has an amplitude of one, and a frequency equal to three times that of the system input sinewave. The gain block, *THD_factor*, represents the means of properly achieving the amplitude of the third harmonic. It is given by

$$THD\_factor = H_3|_{peak} = THD \cdot A \tag{3.17}$$

where A is the system input sinewave amplitude, $H_3|_{peak}$ is the peak 3rd harmonic distortion, and *THD* is the upper bound on the distortion caused by the switches and is given by the equality portion of Eq. (3.16).

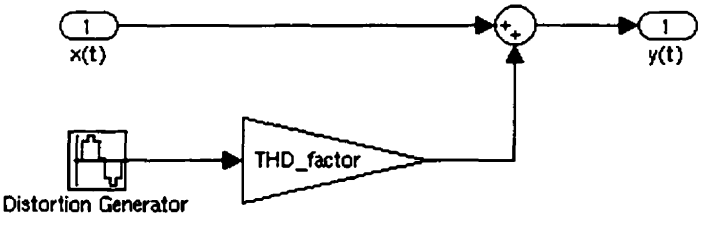


Figure 3.8: Simulink Model for Switch Distortion

3.3.3 OTA Noise

The input referred noise of an OTA can be modeled as follows:

$$y(t) = k_{int} \cdot [x(t) + n_{OTA}(t)] \tag{3.18}$$

where

$$n_{OTA}(t) = V_n \cdot RN(t), \tag{3.19}$$

k_{int} is the integrator gain, V_n is the OTA RMS noise voltage, and $RN(t)$ is a Gaussian random number with zero mean and unity standard deviation. The model can be seen in Figure 3.9.

3.3.4 Switched Capacitor Integrator Non-Idealities

Switched Capacitor integrators are composed of non-ideal analog building blocks, the foremost of which is an OTA. As a result, a model needs to be created that takes into account the noise and/or distortion generated by these components.

One of the main non-idealities is caused by having finite gain in the OTA. As a result, the transfer function of the integrator changes depending on the amount of DC gain provided by the amplifier. The ideal transfer function is given by

$$H(z) = \frac{k_{int}}{z-1} \tag{3.20}$$

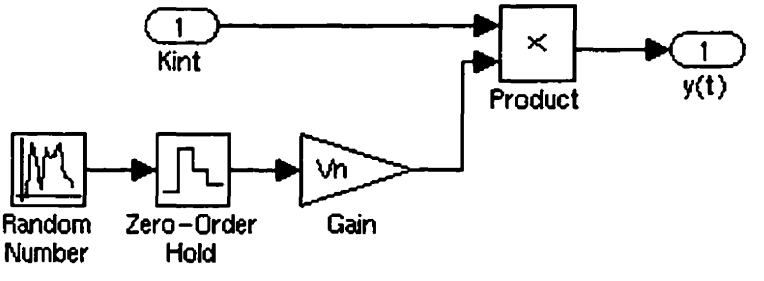


Figure 3.9: OTA Noise Model in Simulink [49]

In reality, this model is not sufficient for the complete description of the integrator's characteristics. Examining Figure 3.10 reveals that the transfer function of the integrator changes depending on the amount of DC gain provided by the amplifier. As a result, the transfer function takes on a gain dependency given by:

$$H(z) = \frac{k_{int}}{z-\alpha}, \tag{3.21}$$

where k_{int} is the integrator gain, $\alpha = \frac{g_m \cdot r_o - 1}{g_m \cdot r_o}$, and g_m as well as r_o are the parameters in the amplifier that define its gain. The deviation caused by finite gain, and represented by α, must be accounted for in any modeling.

Another source of non-ideality comes about because an OTA is subject to clipping. Clipping occurs when the OTA is asked to produce an output voltage higher than the voltage supply rails. Should this occur in the system, the output would cease to follow the ideal output voltage waveform, and instead a distorted waveform would be produced. This is illustrated in Figure 3.11, where V_H and V_L are the upper and lower power supply rails. This effect must be integrated into the model as well, and is accomplished through the use of a saturation block in Simulink.

A final source of non-ideality that will be dealt with is slew and bandwidth limiting. The slew and bandwidth behaviour of an OTA were modeled based on analysis in which their effects in a switched capacitor integrator are interpreted as a nonlinear gain [56].

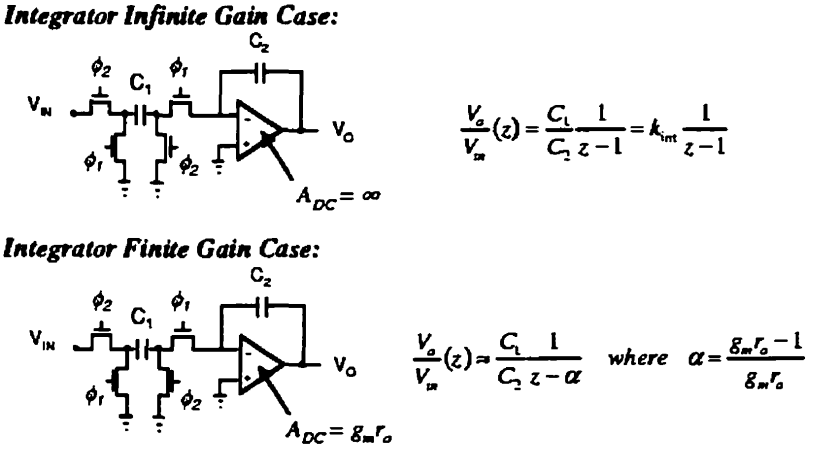


Figure 3.10: Transfer Function of Integrator with Finite Gain

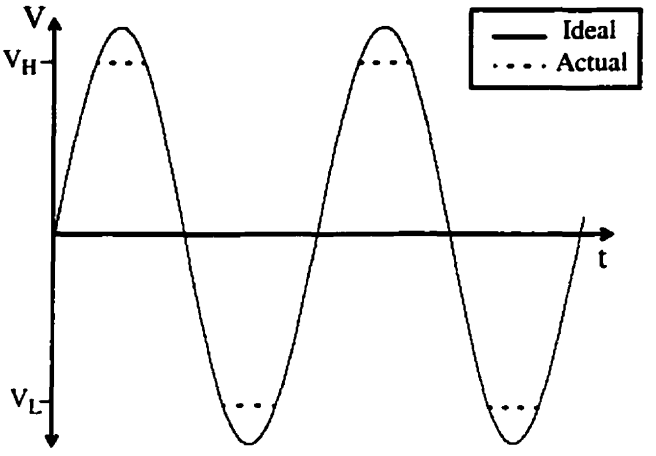


Figure 3.11: Distortion Effect of Clipping

There are three basic mutually exclusive conditional states that can occur, based on the present input voltage and the previous output voltage:

1. Complete Bandwidth Limiting,
2. Complete Slew Rate Limiting,
3. First Slew Rate Followed by Bandwidth Limiting.

In order to understand and model this effect properly, it will now be explored in some detail. The primary goal is to capture the input-output transient behaviour of the circuit in mathematical form. Figure 3.12(a) shows the single-ended SC integrator, while Figure 3.12(b) reveals the circuit representation of the integrator on the driving phase ϕ_2. It is important to recall, that in a sampled data system such as this, there is a finite amount of time given for all the phases to execute. In a two phase system, that time corresponds to *T/ 2* (where *T* corresponds to the clock period). For the purposes of this analysis, assume that capacitor C_3 is a lumped capacitor containing the output parasitics and the loading of the next stage. It is also important to realize that the analysis was done on a single-ended amplifier to simplify the understanding of the procedure. If one takes advantage of the symmetry present in a differential circuit, the same analysis is valid [57].

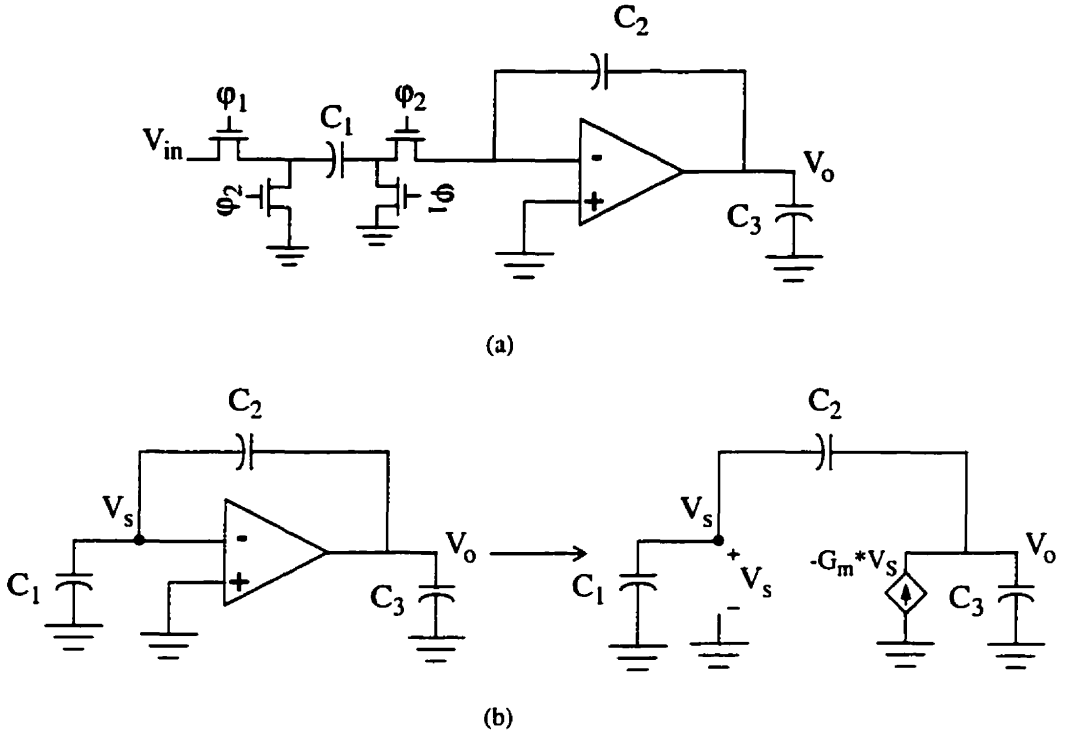


Figure 3.12: Analysis of Transient Integrator Behaviour

The transient associated with the output voltage can be represented in the following form:

$$v_o(t) = v_o(0^-) - G \cdot v_S(0) \cdot \left(1 - e^{\frac{t}{\beta \cdot w_t}}\right) \tag{3.22}$$

where $\beta = \frac{C_2}{C_1 + C_2}$, $w_t = \frac{G_m}{C_L}$, $C_L = C_3 + \frac{C_1 C_2}{C_1 + C_2}$, $G = \frac{C_1}{C_2}$, $v_S(0)$ is the input OTA node voltage at the beginning of the cycle, and $v_o(0^-)$ refers to the output voltage present at the end of the previous cycle. This means that effectively, the charging/discharging time constant τ of the circuit can be defined as:

$$\tau = \frac{1}{\beta w_t} = \frac{C_1 C_2 + C_2 C_3 + C_1 C_3}{G_m C_2} \tag{3.23}$$

It is important to note that $v_S(0) = -v_{in}(0)$, and as a result, Eq. (3.22) can be written as

$$v_o(t) = v_o(0^-) + G \cdot v_{in}(0) \cdot \left(1 - e^{\frac{t}{\beta \cdot w_t}}\right) \tag{3.24}$$

The system behaves as prescribed by Eq. (3.24) when limited purely by the charging/ discharging time constant, and is commonly referred to as being bandwidth limited.

The system may also suffer from Slew Rate (SR) limitations. This is illustrated in Figure 3.13. Since the amplifier can only supply a maximum finite current (I_{bias}) to the capacitive load (C_L), an inherent limit to the instantaneous change in output voltage is present. Therefore, if the change in voltage is too large, this limit is exceeded, and the output will be subject to a linear charging (with slope SR) described by

$$SR = \frac{d}{dt} v_o(t)\Big|_{max} = \frac{I_{bias}}{C_L}. \tag{3.25}$$

If it is assumed that the system is slew rate limited for a portion of the time t_o, the output voltage can be written as

$$v_o(t) = \begin{cases} v_o(0^-) + SR \cdot t, & if \ t \le t_o \\ v_o(0^-) + SR \cdot t_o + (G \cdot v_{in}(0) - SR \cdot t_o) \cdot \left(1 - e^{\frac{t_o - t}{\tau}}\right), & if \ t_o \le t \end{cases} . \tag{3.26}$$

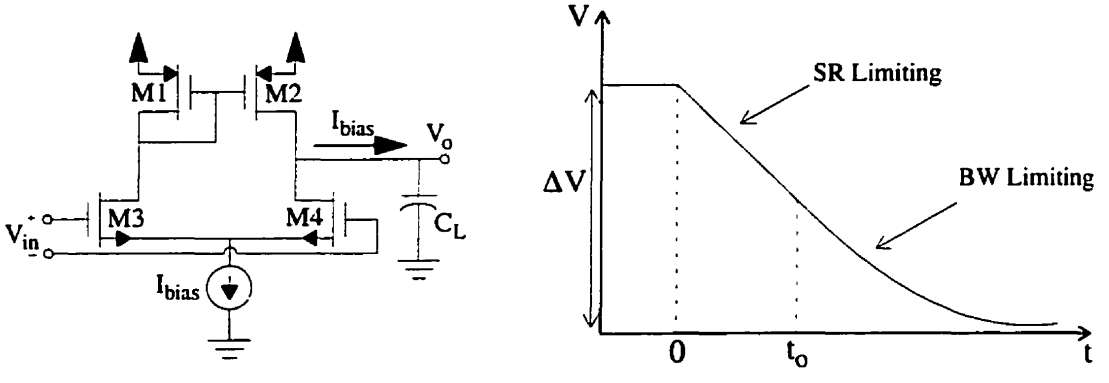


Figure 3.13: The Presence and Effect of Slew Rate Limiting

At time t_o, the system ceases to be slew rate limited and returns to its "natural" charging/discharging state and becomes bandwidth limited. It follows that at only the point corresponding to t_o, the bandwidth and slew rate limiting will be identical. This arises because bandwidth limiting produces a monotonically increasing/decreasing exponential voltage function. Such a response has a different slope at every point, and thereby will be equal to SR only once (i.e. at $t = t_o$, $\frac{d}{dt}v_o(t) = SR$). As a result, one can solve for the time the system remains slew rate limited by evaluating the slope of the bandwidth limited response at time t_o, and equating it with the slope of a slew rate limited system, i.e.,

$$\frac{d}{dt}v_o(t)\Big|_{t_o} = \left(\frac{-SR \cdot t_o}{\tau} + \frac{G \cdot v_{in}}{\tau}\right) \cdot e^{\frac{t_o - t}{\tau}}\Bigg|_{t_o} \tag{3.27}$$

After evaluating the function at time t_o, one is left with the following equation,

$$SR = \frac{-SR \cdot t_o}{\tau} + \frac{G \cdot v_{in}}{\tau} \tag{3.28}$$

This can be easily re-arranged to provide,

$$t_0 = \frac{G \cdot v_{in}}{SR} - \tau \tag{3.29}$$

Therefore, slew rate limiting will occur if $\frac{d}{dt}v_o(t) > SR$ and last for the length of time given by t_o.

When creating a model involving slew rate and bandwidth limiting, the actual transient behaviour of the integrators is not crucial since SC circuits depend only on the output at the end of each cycle. Therefore, it is important to obtain an expression for the output voltage of the integrator at time *T/2*. The appropriate behaviour at this time can be summarized as follows:

$$v_o\left(\frac{T}{2}\right) = \begin{cases} v_o(0^-) + G \cdot v_{in} \cdot \left(1 - e^{-\frac{T}{2\tau}}\right), & if\ t_o \le 0 \\ v_o(0^-) + SR \cdot t_o + (G \cdot v_{in} - SR \cdot t_o) \cdot \left(1 - e^{-\frac{t_o - \frac{T}{2}}{\tau}}\right), & if\ 0 < t_o \le \frac{T}{2} \\ v_o(0^-) + \frac{SR \cdot T}{2}, & if\ t_o > \frac{T}{2} \end{cases} \quad (3.30)$$

The finite gain, the clipping levels, the finite bandwidth and the slew rate limiting of the OTA are incorporated into a Simulink model of the SC integrator as shown in Figure 3.14. The formulas used in this block depend on the type of OTA implemented. In this case, the formulas for the folded cascode amplifier were used, hence parameters C_S, g_m, r_o, and I_{bias} are required (Table 3.3). From the diagram, it can be seen that a Slew & Bandwidth Matlab function block exists that interfaces with a multiplexer. Due to its conditional nature, the modeling was much easier accomplished through the use of a Matlab function, which implemented the previously described equations. Furthermore, it was necessary due to the nature of slew rate limiting, to create a block that is capable of retaining "memory" of its previous state. This is accomplished by feedback through the multiplexor, which also takes in the key parameter values (C_S, g_m, r_o, I_{bias}) fed to it by the optimization routine.

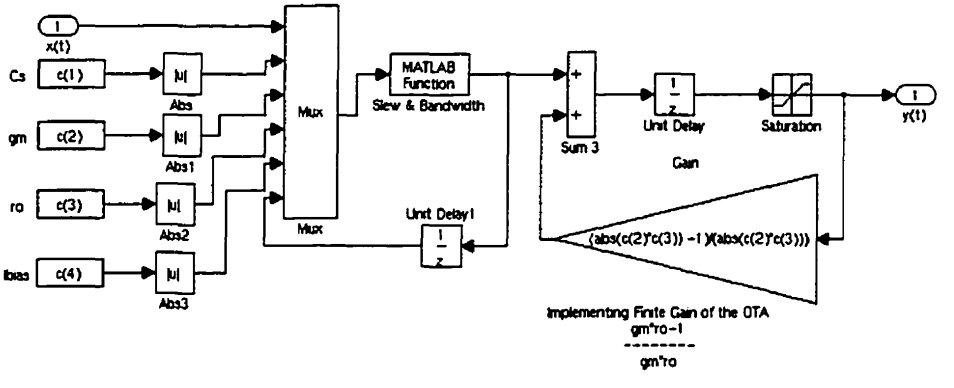


Figure 3.14: Non-Ideal Integrator Model in Simulink

3.4 Optimization Setup

In this section, a set of guidelines that clarify the Matlab design space will be presented to help explain and carry out the design methodology. Some familiarity with the Matlab optimization toolbox will be assumed.

The optimizer is essentially split up into three parts, and is illustrated in Figure 3.15. Each block is described below:

1. The Matlab Optimization Block, which receives an error value and accordingly adjusts the key parameters given to it in order to minimize any further error. On the first run, the optimizer begins by using the initial conditions supplied to it for the key parameters.

2. The Simulink System, with its built in non-idealities, is called by the optimizer so that data for property measurements can be obtained (i.e., getting the required data to produce SNDR calculations).

3. The Matlab Evaluation Block receives data from the Simulink System, evaluates the results based on constraints (bounds) on the key parameters, and produces a value for the error based on the designer's criteria. When the error is low enough the process will stop and deliver its results.

For the specific case of the ΔΣ Modulator, the folded cascode OTA was chosen, and as a result, the key parameters that necessitated optimization were C_S, g_m, r_o, and I_{bias}. It is important to note that although R_{MAX} is used, it need not be optimized. From Eq. (3.16), it can be inferred that R_{MAX} is dependent on the parameter C_S, therefore its bound can be calculated at the end of the procedure and then reported. The next step is to define both the error measurement criteria and the constraints. It is important to note that these criteria can be as simple or as complicated as the designer wishes. The more criteria that are given translates into a longer optimization routine, but a final result that is closer to the overall goal. In the case given here, the following three criterion were used:

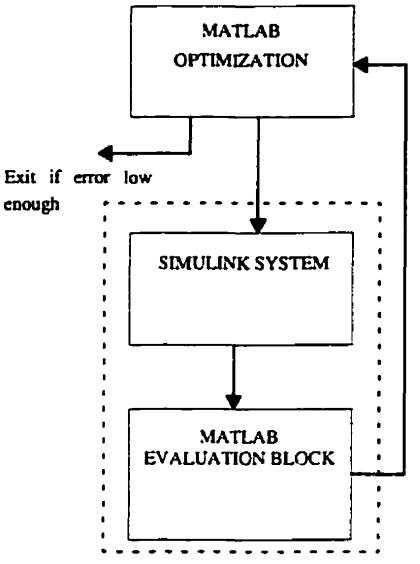


Figure 3.15: Optimization Setup

1. Minimization of Area. This was formulated on the basis of capacitance. Since the capacitance required in single loop modulators usually accounts for at least half of its area, the area constraint placed an emphasis on minimizing the capacitance.

2. Minimization of Power. The main power consumption in the modulator comes from the OTA used in the first stage integrator. As a result, constraints were placed on the bias current in order to place an upper bound on power consumption, and to minimize the current used.

3. Maximization of SNDR. Since SNDR is a dominant measure of a modulator's performance, lower limits were placed on its value, with an emphasis placed on its maximization.

The Simulink system was used in order to get the data for the SNDR measurements. All these measurements were affected by the key parameters, as the modeling for the system was based around them. An example of a $\Delta\Sigma$ simlink system, with all of the non-ideality modeling present, can be seen in Figure 3.16. This is the same system that was presented in Figure 3.2, except that in this diagram, the Simulink models have been added

to account for kT/C and OTA noise, switch distortion, and to account for the use of a non-ideal integrator in the first stage of the modulator.

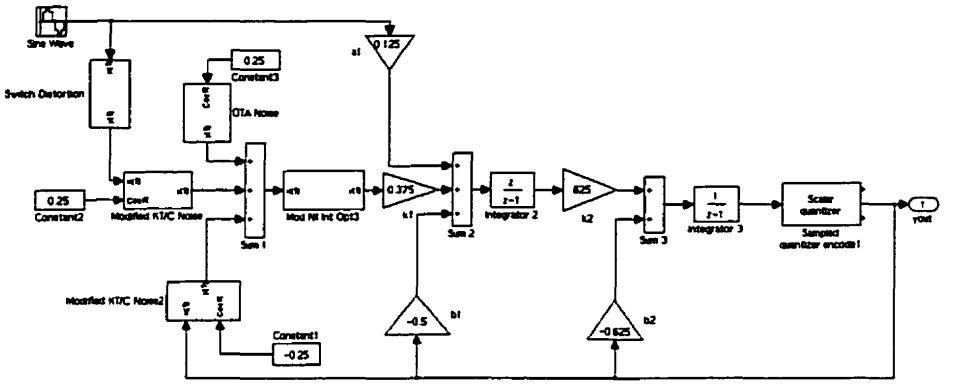


Figure 3.16: Non-ideal System in Simulink

```
% File: optim.m

function [c,Rmax] = optim(Npoints,M,Supply_factor,fs,bw,simtime1,SNDR_goal);

% Calculation of Constants

A = ampli;
SNDR_weight = 10^(SNDR_goal/10);
number_of_tau = -log(10^(-1*(SNDR_goal/20)));
maxtime = 1e9*(1/fs)*(1/2)/number_of_tau;
window1 = kaiser(Npoints,30);
band_edge = floor((bw/fs)*Npoints)+ 1;
THD = 10^(-SNDR_goal/20);
THD_factor = A*THD;

% Starting values of Coefficients
% Cs    1e-12 (pF)
% gm    1e-4  (100's uV/Amp)
% ro    1e6   (Mohm)
% Ibias 1e-4  (100's uAmps)
coef(1)= 4.5;
coef(2)= 33;
coef(3)= 1;
coef(4)= 5;

% The optimizer will try to minimize Cs, gm, ro, and Ibias

% Lower and Upper Bounds on the Coefficients
VLB = [3.5 10 .2 .5];
VUB = [5 34 1 10];

% Set up Optimization Options: Used to Define Tolerances and Step Sizes
options = foptions;
options(1) = 1;
options(2) = .1;
options(3) = .01;
options(4) = .1;
options(14) = 1000
options(16) =.1;
options(17) = 1;

[C,dummy]=constr('param',coef,options,VLB,VUB,[],Npoints,M,Supply_factor, ...
                 THD_factor,simtime1,SNDR_weight,maxtime,SNDR_goal,window1,band_edge);

 Rmax = -1/(4*fs*c(1)*log(THD/(THD+1)));
```

Figure 3.17: Matlab Optimization Block: *optim.m*

3.4.1 Implementation in Matlab

Example code for the third order ΔΣ modulator is presented in Figure 3.17 and Figure 3.18. The first function called "optim.m" (Figure 3.17), basically implements the Matlab Optimization Block seen in Figure 3.15. This function takes in performance specifications from the user (such as the sampling frequency (f_S), and the desired SNDR (SNDR_goal)) and is also used to set up the optimization options such as the step size and

```
% File: param.m

function [L2,g] = param3o(coeff,Npoints,M,Supply_factor,THD_factor,simtime1,SNDR_weight, ...
                          maxtime,SNDR_goal,window1,band_edge);

% Running Simulink to get the stream
  disp('Simulating...');

inputs = simset('SrcWorkspace','current','DstWorkspace','base');

%Creating the Varaibles for Interface with Simulink

c = [(coeff(1)*1e-12) (coeff(2)*1e-4) (coeff(3)*1e6) (coeff(4)*0.5e-4)];
Cs = c(1)
gm = c(2)
ro = c(3)
Ibias = c(4)

[tsim,xsystem,yout]=sim('Non_ideal_modulator',simtime1,inputs);

simtime = length(yout);
hy =(2*(abs(fft(yout(simtime-Npoints+1:simtime).*window1,Npoints))).^2)/((Npoints)*(Npoints-1));


% Additional Factors for use in Time Constant Calculation

Cp1 = 1e-12;
Ck1 = .75e-12;
Clp1 = 2.346e-12;

% Time Constant on Each Phase

tau11 = ((4*c(1)*Cp1) + Clp1*(4*c(1) + Cp1))/(c(2)*4*c(1));
tau12 = (Clp1 + Ck1 + 4*c(1)*(2*c(1) + Cp1)/(6*c(1) + Cp1))/((c(2)*4*c(1))/(6*c(1) + Cp1));

tau = 1e9*max(tau11,tau12);

% Calculate SNDR with Kaiser Window

noise1 = sum(hy(1:(M+1-floor(19/2)-1)));
noise2 = sum(hy((M+1+floor(19/2)+1):band_edge));
noise = noise1 + noise2;
noise = noise./((.402)^2);
peak = (hy(M+1))/(.2276^2);
sndr = 10*log10(peak/noise)
mag = peak/noise;

L2 = (abs(1/mag))*SNDR_weight

disp('Done.');

% ******* set up constraints **********
% constr <= 0

constr1 = tau - maxtime        % Tau is less than the maximum allowable time
constr2 = -tau                 % Tau is greater than zero
constr3 = -sndr + SNDR_goal    % SNDR is greater than the SNDR goal

g = [constr1 constr2 constr3];
```

Figure 3.18: Matlab Evaluation Block: *param.m*

tolerance of the program. This program strives to minimize the coefficients that are specified, in this case C_S, g_m, r_o, and I_{bias}. By minimizing these parameters, the minimal requirements for the OTA will result, and power should also be minimized. It is in this function where the initial conditions along with the bounds on the variables are set, which will be discussed in sections 3.4.2 and 3.4.3. It is also in this function where all preliminary calculations are done. For example, the windowing function for the Fast-Fourier Transform (FFT) used to calculate the SNDR, and the *THD_factor*, are created here. This is created only once and the variables are passed to the Matlab Evaluation Block everytime an SNDR is calculated. By creating them in this function rather than recalculating them in the evaluation block, valuable CPU time is saved.

It is also in this function that R_{MAX} is calculated. This is done once at the end of the program, as R_{MAX} depends on C_S.

The function "param.m" (Figure 3.18) basically implements the Matlab Evaluation Block and the Simulink System Block. In fact, the Simulink System Block is implemented through a function call to the Simulink file "Non_ideal_modulator" seen in Figure 3.16. The Matlab Evaluation Block, which is the remainder of the file, receives the information gathered from the Simulink simulation, and calculates an error based on how far this is from the desired result. Furthermore, several constraints can be placed on any of the variables or values derived from them. For example, the settling time of the integrator is calculated in this function. A constraint is placed upon it, forcing the optimization to continue until the settling time is less than the maximum settling time allowable to still achieve the desired performance. Further constraints in this function force the settling time to be greater than zero (a condition that prevents the use of negative numbers), and force the optimizer to continue until the desired SNDR performance is achieved.

The final function "slew.m" (Figure 3.19), represent the code that was used to implement the Matlab function which accounts for bandwidth and slew rate limiting. This code is specifically tailored to the 3rd order $\Delta\Sigma$ modulator seen in Figure 3.16, however, it can be easily adapted for other designs. This function takes in the inputs C_S, g_m, r_o, I_{bias}, u, v (as seen in Figure 3.14), where u is the current input to the integrator, and v is the previous

```
function [out]= slew(u,Cs,gm,ro,Ibias,v)

% u is current input
% v is previous output

% Sampling Frequency
Fs = 6.144e6;
Ts = (1/Fs);
Tsn = 1;              % All Equations are Normalized to Ts

% Capacitors Used in the Integrator
% Note: Gain of integrator is 1/4

Clp = 2.4e-12;        % OTA Output Capacitance
Cp  = 1e-12;          % OTA Input Capacitance
Ck1 = .75e-12;        % Capacitance of Next Stage

% Effective Load
CL = Clp + Ck1 + (4*Cs*(2*Cs + Cp))/(6*Cs + Cp);

% Slew Rate of the OTA (V/sec)
Sr = Ibias/CL;
SR = (Sr/Fs);

% Unity Gain Bandwidth of the OTA
ugbw = gm/(2*pi*CL);
UGBW = (ugbw/Fs);

% Scale Input to Integrator
d = u - v;

% Time Constant
tau = 1/((2*pi*UGBW*4*Cs)/(6*Cs + Cp));

% Slew Time: Implement Slewing Equations Here
to = ((abs(d))/SR) - tau;

f = abs((d/tau));
if abs((d/tau)) <= SR

    outpt = v + d*(1 - exp(-Tsn/(2*tau)));
else

  if to < (Tsn/2)

      outpt = v + d + (SR*to - d)*exp((to-(Tsn/2))/tau);

  else
      if d > 0
         outpt = v + (Tsn/2)*SR;
      else
         outpt = v - (Tsn/2)*SR;
      end

  end
end


out = outpt;
```

Figure 3.19: Matlab Function: *slew.m*

output of the integrator. This function is basically used to properly implement Eq. (3.30) in its entirety. It is important to note that the input u is already scaled by a factor of 1/4 (as seen in Figure 3.16), and therefore the quantity G in Eq. (3.30) is set to one.

3.4.2 Initial conditions

A good set of initial conditions, or an adequate bounding of the optimized variables is needed, for improved optimization efficiency, and to increase the chances of obtaining a convergent solution.

The best set of initial conditions and bounds will usually come from the designer, as a designer usually has a better intuitive understanding, and more knowledge concerning the feasibility of certain values. For example, a simulator may solve for a capacitance of 1 nF, but a designer knows that such a value is infeasible or undesired. The task then shifts to the designer for obtaining a good set of initial conditions and bounds.

In addition, the knowledge of transistor parameters such as r_o and g_m can help to obtain all of the required information necessary for setting bounds on variables. Preliminary simulations in SPICE can solve this problem quickly and efficiently. For example, the measurement of r_o for a single transistor is a trivial matter, and based on this value a range of values for the output resistance of an OTA can be constructed. Furthermore, feasible bounds for g_m can be found by simulating a differential pair, and sweeping the bias current. This process is very simple and quick, and only need be done once for each new technology. Bounds can also be created based on the designer's preference. For example, if a capacitance is desired, then upper and lower bounds can be set so that no solution will contain too low (put the lower bound equal to that of the minimum sized matchable capacitor allowable in the technology) or too high a capacitance for the designer's liking. Once bounds have been chosen for each of the key parameters, choosing the initial conditions is not so arduous a task, as any values within those bounds can be taken as a starting point.

3.4.3 Additional Factors

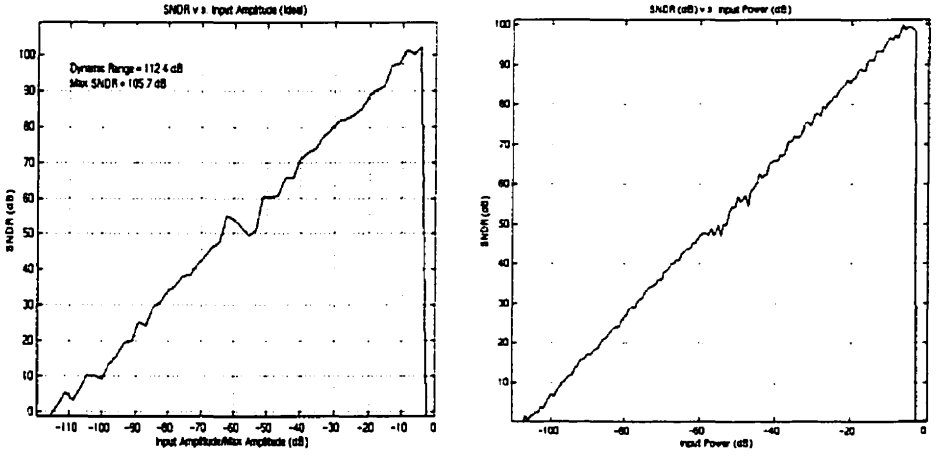
There are additional factors that can alter the behaviour of the system. As examples, the input and output capacitances of the amplifier along with the loading capacitance of the common-mode feedback circuit (CMFB) can affect the settling times of the integrators. These parameters can be accounted for by lumping them in with other key parameters. For example, C_L can be altered to reflect all the additional loading mentioned. The inclusion of

these additional factors can be left to the discretion of the designer, or can be part of a larger iterative design procedure.

3.5 Summary of Simulation Results

This optimization procedure was applied to the design of a 16 bit (SNDR = 98 dB), 3rd order ΔΣ ADC, in which the primary objective was to design a lowpass, audio-band (bandwidth of 24 KHz), single bit, modulator, with an oversampling ratio of 128. The choice of coefficients used in the Simulink diagrams is purely an architectural design consideration and will not be discussed here (see Chapter 2 for more details). The focus remains on the performance of the system with the modeled non-idealities.

An ideal Simulink simulation of this modulator was run in order to determine the maximum achievable performance of such a system. The simulation yielded a peak SNDR of 105.72 dB, and a dynamic range of 112 dB, which can be seen in Figure 3.20(a). After the addition of the models that were presented in section 3.3, the Simulink diagram was constructed (Figure 3.16), and the optimization routine was run. The process resulted in feasible design values (g_m = 3.3 mA/V, r_o = 963 kΩ, I_{bias} = 458 μA, C_S = 4.5 pF, and R_{MAX} = 785Ω). An SNDR curve which reflected the modeled non-idealities was produced, and can



(a) Ideal (b) Nonideal

Figure 3.20: SNDR Plots for 3rd Order Modulator in Simulink

be seen in Figure 3.20(b). The simulation of the non-ideal modulator yielded a peak SNDR of 99.45 dB, and a dynamic range of 102 dB. This system was verified in SPICE through a series of simulations.

3.6 A Fully Coded ΔΣ Modulator Design Example

A brief example will be presented in order to further clarify the design process. This example is fully reproduceable in Matlab and Simulink, and can be used to get an understanding of the methods needed for the design. The same 3rd order ΔΣ modulator will be used to design for kT/C noise requirements. The modulator will operate at 6.144 MHz, with a bandwidth of 24 kHz, have 16 bit performance, with a power supply of 2.5 V.

In order to begin, one can break the process down into simple steps to be followed:

1. Identify/Select the parameters to be optimized.
2. Create/Use Simulink models to implement the non-idealities caused by the chosen parameters.
3. Select the performance criteria, and formulate relevant constraints.
4. Choose the initial conditions, upper and lower bounds, and integrate any relevant additional factors.

For this design, the effect that will be examined is kT/C noise. This noise relates to the input capacitors C_S (as explained in section), and as such is the parameter that will be optimized.

The step that follows involves designing the kT/C noise block for the integrator. The model used is based on the one seen in Figure 3.5, but requires some customization for this design. The first issue to deal with concerns the *f(u)* block, and its variables (see section). For this example only one temperature point will be examined, so a fixed value of 300 degrees Kelvin can be used for T. On top of this, C_S is needed in the formula. For the purposes

of this optimization routine, C_S will be defined as *c(1)*. Therefore, *c(1)* must appear in the equation. This allows for the optimization routine to directly interface with Simulink. Furthermore, since a normalized system was constructed in Simulink (i.e. the power supply is normalized to +1 V and -1 V), a value for *Supply_factor* should be chosen. Assuming that the power supply is 2.5 V leads to a selection of 1.25 for *Supply_factor*. Upon the customization of the Simulink model, a full system diagram should be built. For this example, the diagram can be seen in Figure 3.21, and includes the Simulink kT/C modeling

The next step that must be taken involves deciding what performance criteria will be used to determine a successful completion of the optimization routine. For simplicity, SNDR will be used as the sole performance indicator. In order to calculate this value, a Simulink system must be run, its output data points collected, and an FFT taken. For this modulator, 16 bits of performance is required (98 dB), and as a result, a constraint is placed on SNDR forcing it to be above 98 dB.

Finally, it is necessary to determine the initial conditions and bounds on the parameter C_S. Depending on the technology used, there is a minimum value of capacitance that can be manufactured with a certain pre-known degree of accuracy (usually 20%). Therefore, this value should be used for the lower bound on the capacitance. In this case 250 fF was chosen. The upper bound is usually limited by area constraints. In order to minimize area,

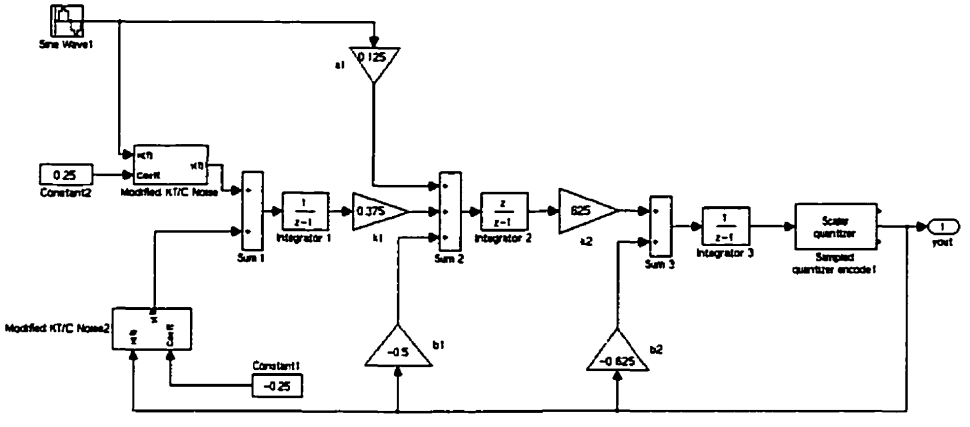


Figure 3.21: Simulink System Diagram with kT/C Noise Models *('KTC_optim')*

one wants to use the smallest capacitance possible. In this case, the upper bound on capacitance was limited to 5 pF. A final decision remains concerning the selection of an initial condition. Basically, one can choose any value within the previously determined bounds.

At this point, there is enough information to create the optimization files. In the first file, seen in Figure 3.22, the system specifications are listed, the Kaiser window for the FFT is calculated, the initial conditions and finally the upper and lower bounds are set. The values listed correspond to numbers that are needed to make various calculations. For example *Supply_factor* is needed for the kT/C noise block. *Npoints* refers to the number of points to be used in the FFT, while *fs* and *bw* correspond to the sampling frequency and bandwidth of the modulator. The value *band_edge* is a constant used to determine the last in-band point to be considered when calculating the FFT. The value *M*, determined by coherency, represents the bin in which the input signal lies. In this case a frequency of 4312.5 Hz corresponds to an *M* of 23. Also defined in this file are the optimization options. The *options* vector contains the parameters used in defining the display format, coefficient accuracy, termination accuracy, constraint accuracy, the maximum number of iterations, and the

```
% File: optim_kTC.m

function [c] = optim_kTC();

warning off;
% Calculation of Constants

M = 23;              % Frequency of 4312.5 Hz
Npoints= 2^15;
Supply_factor = 1.25;
fs = 6.144e6;
bw = 24e3;
window1 = kaiser(Npoints,30);
band_edge = floor((bw/fs)*Npoints)+ 1;

% Starting values of Coefficient
% Cs    1e-12 (pF)
coef(1)= 3.0;

% Lower and Upper Bounds on the Coefficients
VLB = [0.25];
VUB = [5.0];

% Set up Optimization Options: Used to Define Tolerances and Step Sizes
options = foptions;
options(1) = 1;
options(2) = .1;
options(3) = .01;
options(4) = .1;
options(14) = 1000;
options(16) =.1;
options(17) = 1;

[C,dummy]=constr('param_kTC',coef,options,VLB,VUB,[],Npoints,M,Supply_factor,window1,band_edge);
```

Figure 3.22: File 'optim_kTC'

maximum step-size. Definitions and guidelines for choosing these values can be found in [44].

The second file, which is called in the last line of the previous one, can be seen in Figure 3.23. This file contains the call to Simulink model 'KTC_optim', which was seen in Figure 3.21, the error calculation, and the constraints. The error condition is chosen to get the SNDR as close to 98 dB as possible. This is done by dividing the V/V representation of 98 dB by the current SNDR in V/V does this. Ideally, this will give a value of 1. As a result, 1 is subtracted to indicate that there is no error when this is achieved. This formulation penalizes SNDRs greater than 98 dB, but this is necessary to ensure that the smallest C_S possible is used. The constraint forcing the SNDR to be greater than 98 dB can also be seen in this file.

```
% File: param_kTC.m

function [L2,g] = param_kTC(coeff,Npoints,M,Supply_factor,window1,band_edge);

% Running Simulink to get the stream
  disp('Simulating...');

inputs = simset('SrcWorkspace','current','DstWorkspace','base');

%Creating the Variables for Interface with Simulink

c(1) = [(coeff(1)*1e-12)];
Cs = c(1)

[tsim,xsystem,yout]=sim('KTC_optim',45700,inputs);

simtime = length(yout);
hy =(2*(abs(fft(yout(simtime-Npoints+1:simtime).*window1,Npoints))).^2)/((Npoints)*(Npoints-1));


% Calculate SNDR with Kaiser Window

noise1 = sum(hy(1:(M+1-floor(19/2)-1)));
noise2 = sum(hy((M+1+floor(19/2)+1):band_edge));
noise = noise1 + noise2;
noise = noise./((.402)^2);
peak = (hy(M+1))/(.2276^2);
sndr = 10*log10(peak/noise)
mag = peak/noise;

L2 = abs(((10^(98/10))/mag) - 1)

disp('Done.');

% ******* set up constraints ***********
% constr <= 0

constr1 = -sndr + 98    % SNDR is greater than the SNDR goal

g = [constr1];
```

Figure 3.23: File 'param_kTC'

By creating these two files along with the simulink diagram, this system can be simulated in Matlab. Once run, the optimization routine should quickly find a value for the minimum capacitance needed at the input stage. In this case, the result should be approximately 3.25 pF, and should take approximately 130 seconds of CPU time.

Based on this example involving kT/C noise along with the information gathered from the discussion in earlier sections, one has the necessary tools and background to easily expand upon these principles. Further parameters that influence the non-idealities of the system, along with additional constraints to more thoroughly define performance, can be easily be added to develop a more accurate and comprehensive design.

3.7 Conclusion

A new design methodology for analog or mixed signal integrated circuit components was presented, along with the benefits of a top-down optimization procedure. Foremost among these benefits was a shorter design cycle, along with ease of implementation and reproducibility. A major advantage of having adopted such a design strategy was its universal applicability to any design problem, provided that one has the ability to obtain formulas or rules of thumb to help guide the process. Once these formulas and rules of thumb have been obtained or decided upon, Matlab and Simulink could be used to model them, and an optimization procedure could be conceived, as per the guidelines presented in this chapter. In order to more easily understand and apply this procedure, Simulink modeling along with several design procedures and considerations were presented. Furthermore, the design of a $\Delta\Sigma$ modulator using this methodology was carried out to more concretely illustrate the benefits of a top-down design methodology using Matlab and Simulink.

CHAPTER 4 Low Voltage SC CMOS Single Bit Lowpass ΔΣ Modulation

This chapter introduces the issues behind the design of a low voltage switched capacitor CMOS ΔΣ modulator, starting with the design procedure and realization in a low voltage (2.5 V) standard CMOS process (0.25 μm), and concluding with experimental measurements to gauge performance. The primary objective is to design a lowpass, third order lossless discrete integrator, audio-band (bandwidth of 24 KHz), single bit, delta-sigma modulator. In order to achieve the desired 16 bits of performance, an oversampling ratio of 128 was used, necessitating a system clock run at 6.144 MHz.

4.1 Introduction

A ΔΣ modulator may be realized in one of several ways, including active RC, switched opamp and switched capacitor implementations. For low frequency applications, an active RC implementation requires large capacitors to create the long time constants necessary, which is undesirable due to the heavy area consumption of capacitors. Also, it relies on the absolute precision of both capacitors and resistors, which due to process variations can change by as much as 20%, making it a problematic method of implementation [1, 33, 55]. Recently, several designs have been attempted using a switched opamp realization [58-60].

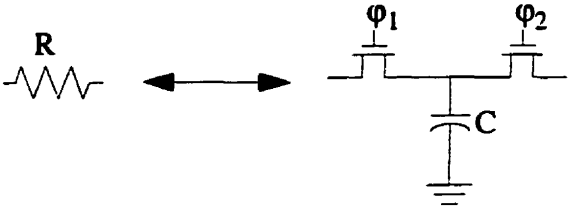


Figure 4.1: Switched-Capacitor Equivalent Resistor

The main reason for this is to design without the use of voltage multipliers to drive the switches [61]. However, there have been successful designs using a SC implementation without the use of voltage multipliers, such as the ones seen in [52, 62]. Furthermore, the performance of switched opamp implementations seem to be somewhat low, with the highest recorded performance to date being 77dB [59].

Switched capacitor circuits have come about based on the fact that if one is to switch a capacitor between two nodes at a high clocking rate, it is equivalent to placing a resistor between the nodes such that $R = \frac{1}{C \cdot f}$, and can be seen in Figure 4.1 [63]. As a result, the time constant of the system ($\tau = R \cdot C$), is now determined by a ratio of capacitors and the clocking frequency. Typically, in a standard CMOS process with good layout matching techniques the mismatch in ratios can be controlled to about 0.1%. Therefore, the large time constants needed can be more easily created in a SC realization, resulting in its selection for implementation.

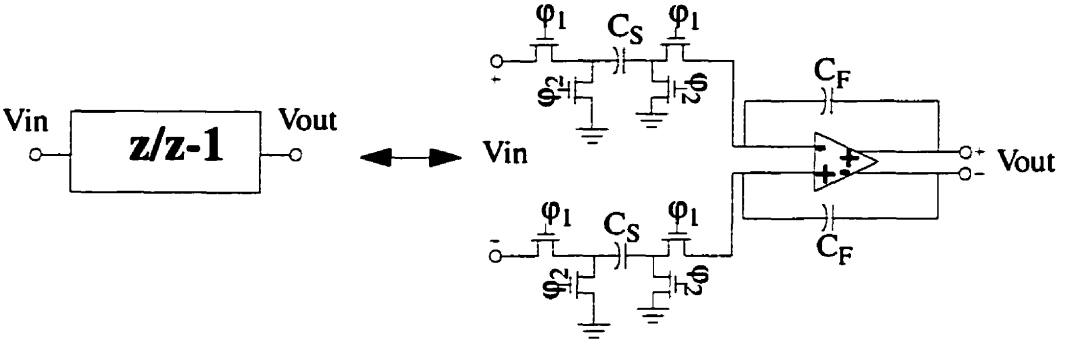


Figure 4.2: Realization of a Non-Delayed Integrator in SC

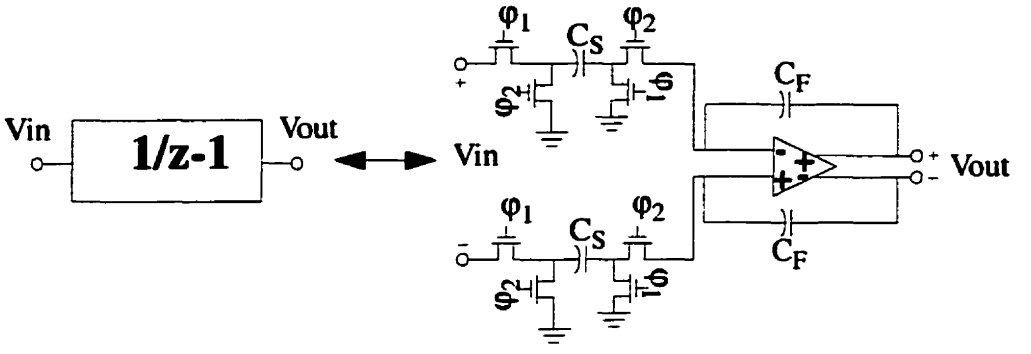


Figure 4.3: Realization of a Delayed Integrator in SC

Since SC circuits are sampled data circuits, they can be designed using z-transform methods. The integrators used in the LDI topology can be converted to SC integrators as seen in Figure 4.2 (a non-delayed integrator) and Figure 4.3 (a delayed integrator). Using these conversions, a SC implementation was used, and a fully differential realization of the modulator can be seen in Figure 4.4.

4.2 Design Verification in SWITCAP2

As the modulator will be a SC implementation, it was deemed necessary to create and simulate the circuit in SWITCAP2 [64], which is a discrete time simulator. This would serve as both a validation of the Simulink results of Chapter 2, and moreover, allow for the verification of the clock phasing that is involved.

The $\Delta\Sigma$ modulator under the exact same conditions used in section 2.3, was simulated in SWITCAP2, and the resulting FFT can be seen in Figure 4.5. The result matched closely with what was previously seen in Figure 2.18, and the SNDR was calculated to be 102.2 dB. Several more simulations were run, at various signal amplitudes, and the deviations from Simulink were found to be at most 0.5 dB.

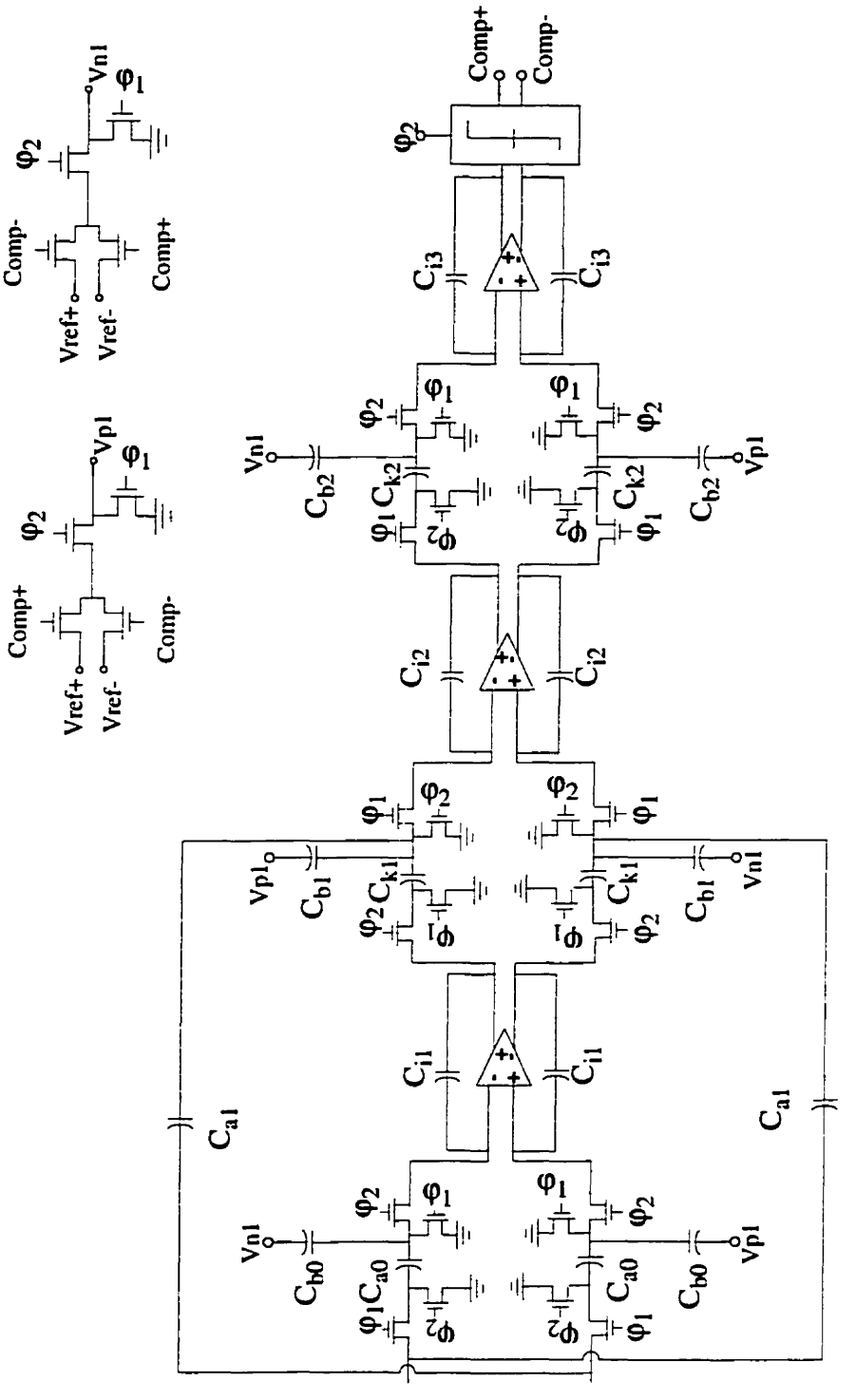


Figure 4.4: Switched Capacitor Implementation of the ΔΣ Modulator

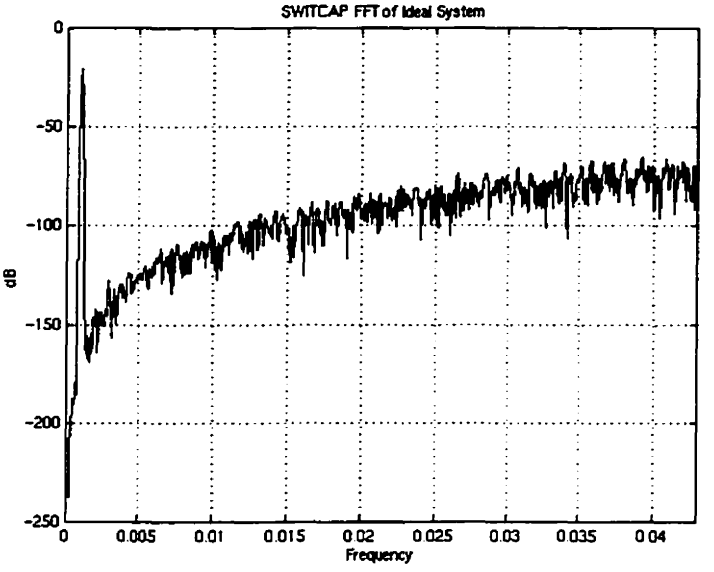


Figure 4.5: FFT of Ideal System in SWITCAP (60% at 5812.5 Hz)

4.3 Switched Capacitor Design Issues

The following section will describe several key issues that had to be dealt with as a result of using a SC implementation.These issues include capacitor layout, sampled capacitor noise, and settling time.

4.3.1 Capacitor Layout

As described in section 4.1, the coefficients of the modulator determine the capacitor ratios needed in the design. As a result, the ability to have linear and matched capacitors is crucial. Since a 0.25 μm standard CMOS, single poly, 5 metal process was used, three capacitor composition options were available:

1. Five Layer Metal Capacitors. These capacitors have the lowest density of the three structures resulting in a larger area of implementation. However, they also are the most linear. This is because the metal layers are the furthest from the substrate, meaning that the parasitic capacitance to the substrate has the least effect.

2. Poly-Metal Capacitors. These capacitors are more dense than the 5 layer metal ones, but since polysilicon is resistive, the capacitors have a lower Q and in essence are worse capacitors [63]. The distance between the polysilicon and the substrate is smaller than that of metal to substrate, therefore its parasitic capacitance is larger.

3. MOS Capacitors. Although much denser than the other two structures, these capacitors are highly non-linear [65]. This is mainly due to the fact that the bottom plate of the capacitor is a doped area, and as a result forms a parasitic diode that predominantly contributes to high non-linearity in the structure [66].

Although the 5 layer metal capacitors have a low density, its other advantages outweigh this fault. As such, the capacitors for the modulator use this composition.

A problematic source in the realization and fabrication of capacitors are etching errors. These errors are unavoidable in the manufacturing process. In order to minimize its effects on the capacitor ratios, the perimeter to area ratio should be kept the same for all the capacitors that are supposed to be matched [33]. This will cause every section to be affected in the same manner, and thereby preserve the ratio of capacitance between each one. This implies that all of the capacitors should be composed of smaller unit sized elements.

In order to obtain the best matching possible, the capacitors from each of the three sections were grouped together and implemented in a square array. One of the three arrays is depicted in Figure 4.6, where capacitors were matched to the ratios of 5/8, 5/8, 5/8, 5/8, 8/8 and 8/8. The capacitor sections with an *a* subscript denote capacitors that appear on the bottom half of the differential structure, and *D* represents a dummy structure.

Each of the capacitor arrays was placed in a large well biased to the appropriate voltage. This was done to help isolate the capacitors from any stray charge in the substrate that might enter into a signal path through parasitics formed between it and the bottom plate.

Dummy capacitors (5 layer metal structures, left unconnected to the circuit) were used to complete missing components of the arrays. This was done to ensure that all capacitor

D	D	D	D	D	D	D	D
D	3	3a	2	2a	3a	3	D
D	3a	3	1a	1	3	3a	D
D	2a	1a	2	2a	1	2	D
D	2	1	2a	2	1a	2a	D
D	3a	3	1	1a	3	3a	D
D	3	3a	1a	1	3a	3	D
D	D	D	D	D	D	D	D

Figure 4.6: Example of a Matched Capacitor Array

edges would see the same thing, and as such make more uniform the etching error, which depends heavily on what surrounds the structure, across the capacitor array.

4.3.2 Sampled Capacitor Noise

One of the most serious limitations of low voltage ΔΣ modulators is the strain imposed by sampled capacitor noise (kT/C) as a design consideration. As the supply voltage decreases, the input voltage range to the modulator is shrunk. A reduced input range can be seen to directly translate into large sampling capacitor sizes for the first integrator as a result of the need to reduce the kT/C noise below the noise floor of the modulator.

The noise from the capacitors results from the sampled noise stored on the capacitor generated by the switch resistances. The mean-squared noise voltage contribution of the sampling capacitors that falls in-band in an oversampled system due to a single switch is given in [1] as

$$Vrms^2 = \frac{k \cdot T}{M \cdot C} \tag{4.1}$$

where k is Boltzmann's constant, T is temperature in Kelvins, and M is the oversampling ratio.

Examining a switched capacitor arrangement (Figure 4.4), shows that each capacitor is sampled through two sets of switches, thereby doubling the mean-squared noise. Furthermore, an extra multiplicative factor of two should be added due to the differential nature of the circuit. This comes from the fact that twice as many switching components are used.

Errors or noise that is introduced at the input node are added to the signal and degrade the modulator performance directly. As a result, the first stage of the modulator is the most sensitive to noise. This means that in order to maximize performance, the sampling noise of the capacitors generated there should be the dominant contributor to the overall sampling noise. As a result, the input referred inband noise power of the capacitors have been estimated as follows:

$$Vrms^2 = \frac{4kT}{MC_{a0}} + \frac{4kT}{MC_{b0}} \tag{4.2}$$

Since both C_{a0} and C_{b0} were chosen to be 4.5pF (see sections 3.5 and 4.4), V_{rms} can be calculated to be 7.583μV, and provide a noise power of -102.4 dB.

4.3.3 Settling Time

A key factor in the modulator design is the settling of the OTAs used to create the integrators needed in the system. This ΔΣ modulator is clocked at 6.144MHz, but since the clock is split into several phases (which is a SC requirement), in actuality the amplifiers have only half of the period to settle (or 81.38 ns).

Furthermore, the precision to which the integrator in the first stage settles should match the overall resolution of the modulator, thereby forcing it to settle to 11.28τ (or 7.21 ns) in order to achieve a precision of 98 dB.

In order to examine this more closely, an analysis [37] on the switched capacitor circuit was carried out. Since the system being analyzed is a SC integrator, the time constant can be given as

$$\tau = \frac{C_L}{Gm \cdot \beta} \tag{4.3}$$

where C_L is the effective load capacitance at the amplifier output, *Gm* is the amplifier transconductance, and β is the feedback factor of the integrator. Based on this, the formulas for the all the integrator time constants of the system seen in Figure 4.4 were calculated and are listed below:

$$\tau_{11} = \frac{C_{i1} \cdot Cp_1 + Clp_1(C_{i1} + Cp_1)}{Gm_1 \cdot C_{i1}} \tag{4.4}$$

$$\tau_{12} = \frac{C_{i1}(C_{a0} + C_{b0} + Cp_1) + (Clp_1 + C_{k1}) \cdot (C_{i1} + C_{a0} + C_{b0} + Cp_1)}{Gm_1 \cdot C_{i1}} \tag{4.5}$$

$$\tau_{21} = \frac{C_{i2}(C_{a1} + C_{b1} + Cp_2 + C_{k1}) + (Clp_2 + C_{k2}) \cdot (C_{i2} + C_{a1} + C_{b1} + Cp_2 + C_{k1})}{Gm_2 \cdot C_{i2}} \tag{4.6}$$

$$\tau_{22} = \frac{C_{i2} \cdot Cp_2 + Clp_2(C_{i2} + Cp_2)}{Gm_2 \cdot C_{i2}} \tag{4.7}$$

$$\tau_{31} = \frac{C_{i3} \cdot Cp_3 + Clp_3(C_{i3} + Cp_3)}{Gm_3 \cdot C_{i3}} \tag{4.8}$$

$$\tau_{32} = \frac{C_{i3}(C_{a2} + C_{b2} + Cp_3) + Clp_3(C_{i3} + C_{b2} + Cp_3 + C_{k2})}{Gm_3 \cdot C_{i3}} \tag{4.9}$$

In these equations, Cp_x and Clp_x represent the input and output capacitances respectively of amplifier x, and τ is interpreted as follows $\tau_{integrator,phase}$.

Although only required for the first integrator, the settling times for all the integrators were constrained to be within 7.21 ns. This allowed for a high confidence design. The time constants for the first integrator is taken care of in Chapter 3 and in section 4.4. In order to obtain the capacitor values for the remaining integrators, the formulas in Eq. (4.6) to Eq. (4.9) were used. Since some of the capacitors appeared in more than one of the equations, a Matlab optimization program was written to minimize the time constants while keeping the capacitor ratios unchanged, and while maintaining discrete capacitance steps of 250 fF in order to facilitate layout.

Capacitor	Value
C_{a0}	4.5 pF
C_{a1}	250 fF
C_{b0}	4.5 pF
C_{b1}	1 pF
C_{b2}	1.25 pF
C_{k1}	750 fF
C_{k2}	1.25 pF
C_{i1}	18 pF
C_{i2}	2 pF
C_{i3}	2 pF

Table 4.1: Final Capacitor Values

It is important to note that this optimization was carried out in conjunction with the design of the OTAs, and therefore the fixed values for the various *Cp*'s and *Clp*'s (which included the capacitance of the common mode feedback circuit) were known. The final capacitor values can be found in Table 4.1, and the time constants for each integrator can be seen in Table 4.2

Time Constant	Value (nsec)
τ_{11}	1.1602
τ_{12}	4.9407
τ_{21}	6.1587
τ_{22}	1.9457
τ_{31}	2.8281
τ_{32}	6.1652

Table 4.2: Time Constants for the Integrators on each Phase

4.4 Non-Ideal Simulation in Simulink

In order to design the system while accounting for non-idealities, and to establish multiple design guidelines such as input stage capacitor size, OTA parameters, and switch resistance, Simulink simulations were carried out. The modeling of the non-idealities and an optimization were done according to the guidelines explored in Chapter 3.

The non-ideality model's key parameters were optimized, and the results can be seen in Table 4.3. The addition of these models significantly affected the performance, and the resulting changes can be seen in Table 4.4. A comparison of the ideal and non-ideal simulation results can be seen in Figure 4.7(a), with a close-up of the passband being shown in Figure 4.7(b). From the figures, the simulations with non-idealities included show a noise floor along with a 3rd harmonic which is not present in the ideal case. This effectively limits the achievable SNDR to 99.45 dB.

Key Parameter	Value
C_S	4.5 pF
g_m	3.3 mA/V
r_o	963 kΩ
I_{bias}	458 μA
R_{on}	785 Ω

Table 4.3: Optimized Parameter Values

Measure	Ideal Simulation	Non-Ideal Simulation
Dynamic Range	112 dB	102 dB
Peak SNDR	105.72 dB	99.45 dB

Table 4.4: Ideal v.s. Non-Ideal Characterization Results

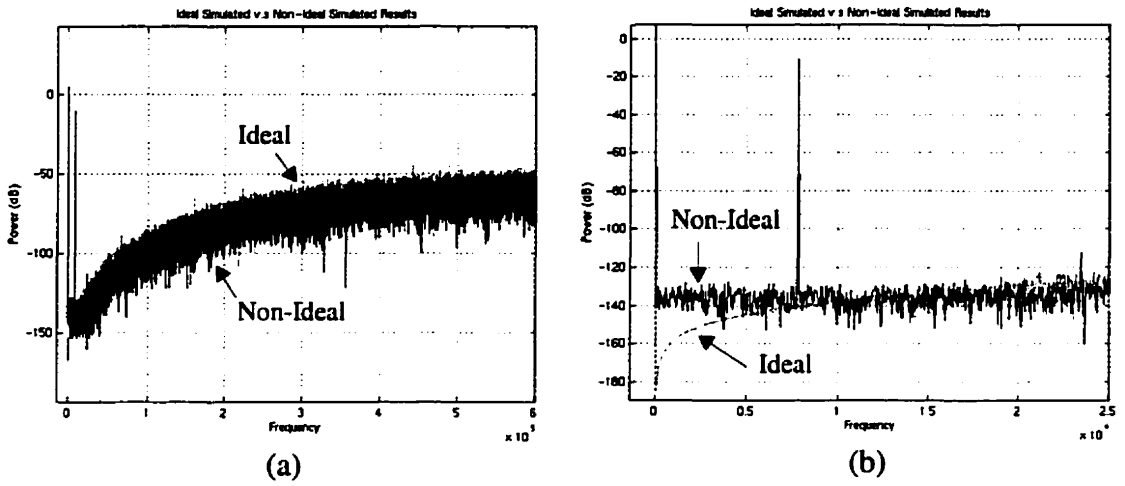


Figure 4.7: Ideal v.s. Non-Ideal Simulink Simulation Results

4.5 Investigation of Process Variation

In order to ensure that this design is suitable for fabrication in a standard CMOS process, some investigation concerning the effects of process variation was conducted on the architecture. This was also accomplished through the use of Matlab and Simulink. It is important to note that the circuit non-idealities were taken into account (Chapter 3 and section 4.4) during the process variation trials in order to account for as many factors as possible.

In this analysis, the absolute capacitor values and capacitor ratio matching were modeled separately. This was done after some consideration about the nature of process variation. Since actual data (other than worst case situations used to characterize a process) is not generally available, some assumptions have to be made in order to carry out this analysis.

For the absolute capacitor values, the following assumptions were made:

1. The worst case tolerances are used as bounds on the amount of process error that will occur during manufacturing. This is to say that an error that exceeds the worst case bounds is considered a catastrophic defect and does not produce valid data for this analysis.

2. Since no real information is known as to the actual distribution of error, a cautious evaluation of the situation dictates that any values within the given bounds are equally manifestable.

As a result of these guidelines, a uniform random variable was used to model the effects of process variation on the absolute capacitance values. The mean of the random variable was set to the desired capacitor value, and the bounds on the variable were set to values corresponding to a 20% deviation, which should model a 20% tolerance in its actual value.

For the capacitor ratio matching, the following assumptions were made:

1. The matching of two entities relies both on the process and on the designer's ability to properly layout a device. As a result, most matching tolerances given assume proper layout. As a result the distribution of error will be dependent on the designer's prowess.

2. Since no process guidelines are given concerning matching, no absolute bounds can be set to determine acceptable error.

As a result of the less stringent conditions, a gaussian random variable was chosen to model matching. Assuming competent layout matching techniques defined the random variable's characteristics. The mean was set to the desired ratio, and the deviation was set to 1%, which should correspond to most of the error falling within a 1% deviation in the ratios. It is important to note that poor matching (> 1%) is feasible and is considered.

The system was then run 1000 times in order to get a fairly good sample set. Histograms were then constructed based on the data collected. A histogram representing the performance of the system through SNDR can be seen in Figure 4.8. The results indicate that the architecture and system design is fairly tolerant of the capacitor errors. The mean was determined to be 97.19 dB, and the standard deviation was 1.12 dB.

Another important feature that was examined was the signal transfer function. The variation in the passband signal gain should be as close to 0 dB as possible. Figure 4.9 rep-

resents that gain over process variation. The mean was determined to be -48.8μ dB, with a standard deviation of 0.1215 dB.

The results of this analysis confirmed that the structure would perform adequately under process variation and is suitable for implementation.

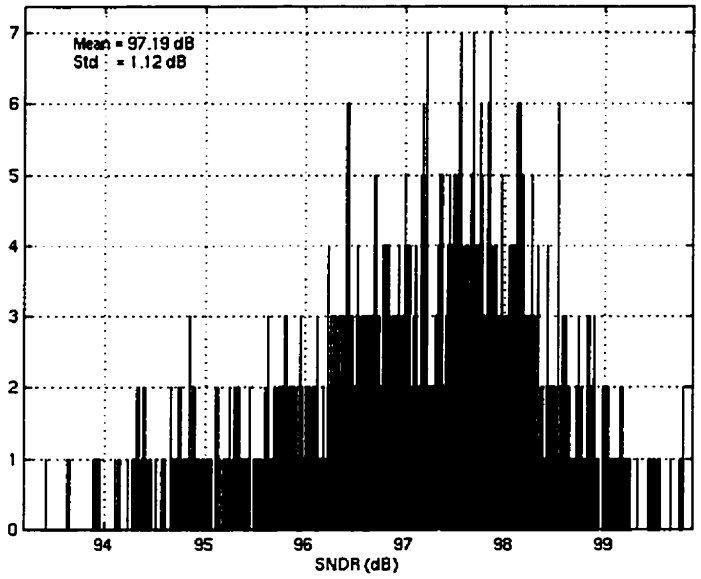


Figure 4.8: SNDR Performance under Process Variation

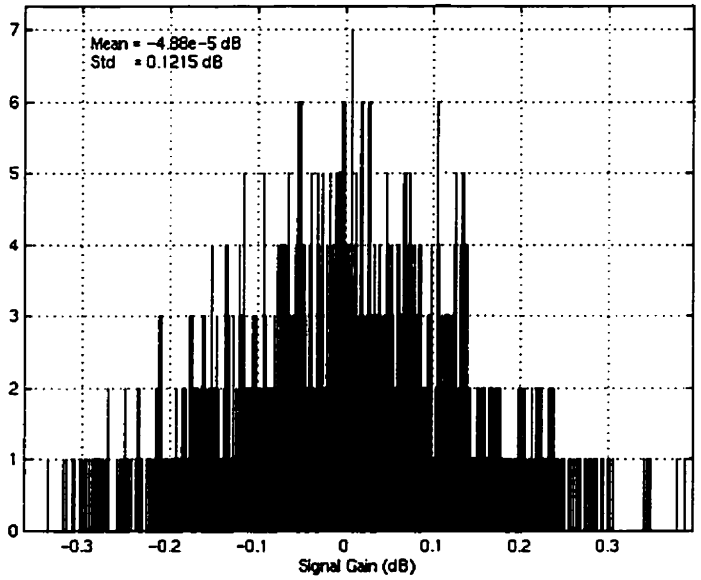


Figure 4.9: Passband Signal Gain under Process Variation

4.6 Design Components

The down scaling of submicron CMOS technology complicates the design of circuits due to reduced headroom (reduced power supply voltage) resulting from the lowering of the technology's maximum allowable voltage. Since the threshold voltages are relatively large with respect to the shrinking headroom, the analog building blocks that make up SC circuits will encounter severe problems when operating at low voltage conditions. The following section will describe the diverse low voltage components used in the design of the ΔΣ modulator. This section will feature the clocking circuitry used, the design of the OTAs, the SC common mode feedback circuitry, the comparator, the switches and the internal DAC.

4.6.1 Clocking Circuitry Requirements

Traditionally, SC circuits are run by a two-phase non-overlapping clock, in which the small time where both phases are low is needed to avoid charge sharing [33]. A two phase clock however is not intolerant to signal dependent charge injection. To help alleviate this, a four phase clocking scheme [32, 67] can be used. This type of system is illustrated in Figure 4.10(a) and Figure 4.10(b), which illustrates the four phase clocking waveforms and corresponding integrator setup.

In a four phase scheme, the amplitude dependent charge injection from switches *S1* & *S2* is stored on the parasitic capacitance C_p, and is dissipated to ground. This occurs

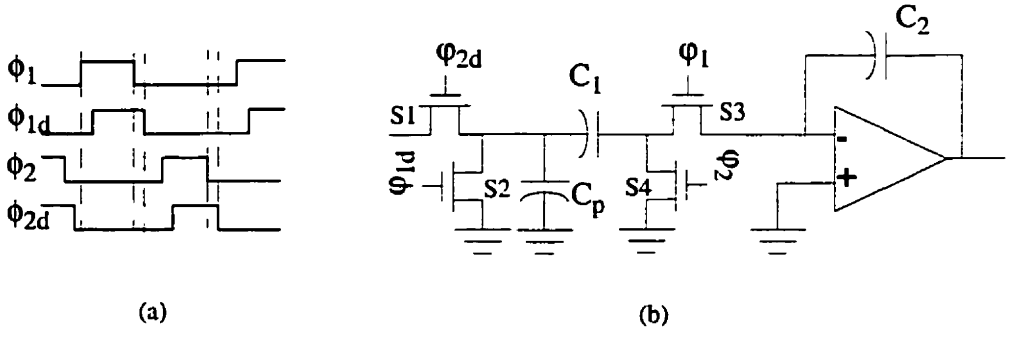


Figure 4.10: Four Phase Clock (a) Waveforms (b) Integrator Setup

because when *S1* opens, *S4* is already open, and therefore the input voltage dependent charge stored in the channel can not be injected onto C_1. Instead, it is stored onto the parasitic capacitance C_p (C_p is a lumped capacitor which represents all of the parasitics at that node). A similar analysis holds true for switch *S2*.

There is however some charge injected when switches *S3* and *S4* are opened. The charge injected from these switches are not signal dependent as both ends of the switch are held at a voltage close to analog ground, and as a result will not contain signal dependent charge which would lead to harmonic distortion.

As a result of the four phase requirements, digital clocking circuitry was built to take in a single clock source and split it into the four required signals.

4.6.2 Operational Transconductance Amplifiers

The choice of an OTA topology plays a crucial role in low voltage and low power design. First of all, it was decided than an OTA rather than an operational amplifier would be used. This was done primarily for the reason of power and output swing conservation. Since an operational amplifier necessitates the use of an output stage, which itself consumes a fair amount of power and limits the output swing by a V_t, it was felt that an OTA could be used without compromising the system.

A fully differential single stage OTA was designed using a folded cascode topology with one level of cascoding. Using the top-down design methodology discussed in Chapter 3, the OTA was required to meet the specifications seen in Table 4.5. The level of cascoding was deemed necessary based on the r_o requirements seen in the table.

The complete CMOS amplifier schematic can be seen in Figure 4.11. Transistors M_5 and M_6 are the pmos input transistors. Transistors M_9 and M_{10} provide loading to the differential pair by pulling a current controlled by the common mode feedback circuit (CMFB). Transistors M_7 and M_8 are the cascode transistors for the folded cascode stage.

Parameter	Value
gm	3.3 mA/V
r_o	963 kΩ
I_{bias}	458 μA

Table 4.5: OTA Design Parameters

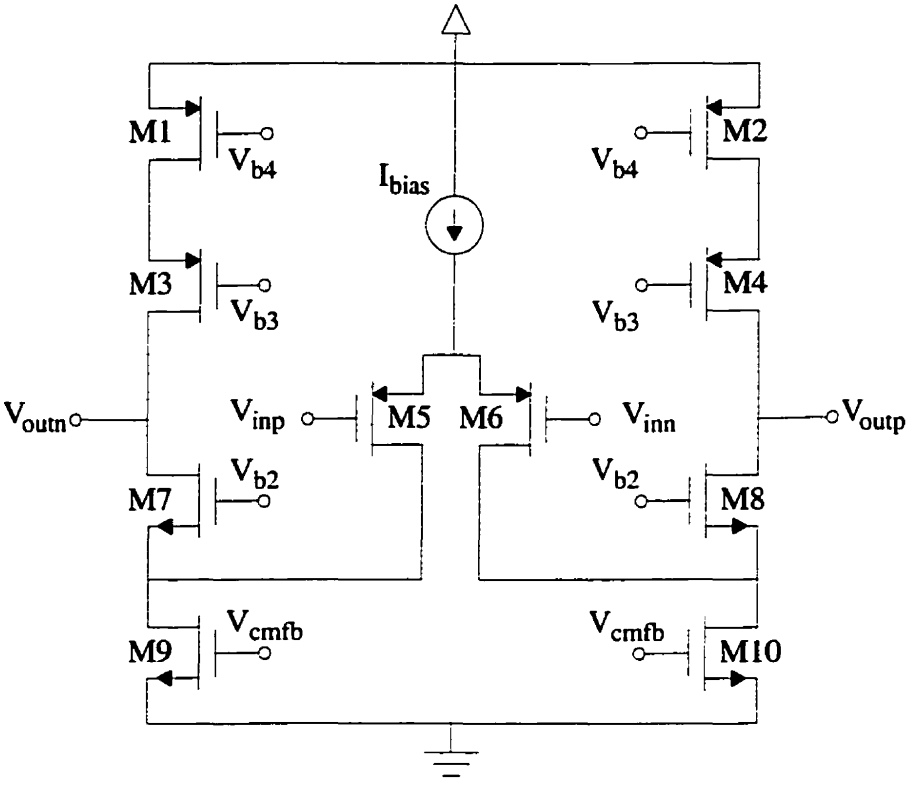


Figure 4.11: Schematic of OTA

The differential output of the amplifier are at the drain of transistors M_4/M_8 and M_3/M_7. Transistors M_1, M_2, M_3, and M_4 are used to provide current to the active load. Pmos transistors were used for the input differential pair in order to provide lower flicker noise [33]. Cascoded nmos transistors were used as the active load in order to present a higher output resistance and thereby increase the gain. The amplifier uses a switched capacitor common mode feedback circuitry which is discussed in section 4.6.3.

Specification	Integrator 1	Integrator 2	Integrator 3
Gain	6557 V/V	8024 V/V	8024 V/V
UGBW	61.5 MHz	72.4 MHz	72.6 MHz
Slew Rate	46 V/μsec	44 V/μsec	44 V/μsec
Phase Margin	78 degrees	72.4 degrees	72.1 degrees
I_{bias}	500 μA	200 μA	200 μA
Power	2.5 mW	1.0 mW	1.0 mW
Load	8.2 pF	3.45 pF	3.27 pF

Table 4.6: Characterization of the OTAs

Building the OTA to adhere to the specifications in Table 4.5, resulted in the characterization values listed in Table 4.6. It is important to note that some of the values from that table were exceeded in the design process, if it facilitated the design. Since the ΔΣ modulator requires three integrators, and therefore three OTAs, the table characterizes all three of them. The same topology was used for each of them, but the effective loading seen on each differed. In an attempt to save power, integrators 2 and 3 were designed to meet the same specs as integrator 1 while using the minimum power possible. Also, the same bias circuit was shared between all three integrators. The values in the table were found through simulation of the layed out (including parasitics) version of the OTA.

4.6.3 Switched Capacitor Common Mode Feedback Circuitry

As described in section 4.6.2, a fully differential OTA structure was used in the design of the modulator, and as such required a common mode feedback loop in order to set the output DC level properly. Since OTAs are able only to drive capacitive loads, it is not suitable to use a continuous time resistive CMFB circuit without the use of a buffer. As a result, in order to save power and maintain the output swing, it is advantageous to capitalize on the fact that a SC system is being built, and use a SC CMFB circuit [68] as seen in Figure 4.12.

The switches used are nmos transistors, with the exception of the switches connected to the output of the amplifier, which were transmission gates with dummy structures (this was done in order to preserve the output swing and reduce charge injection).

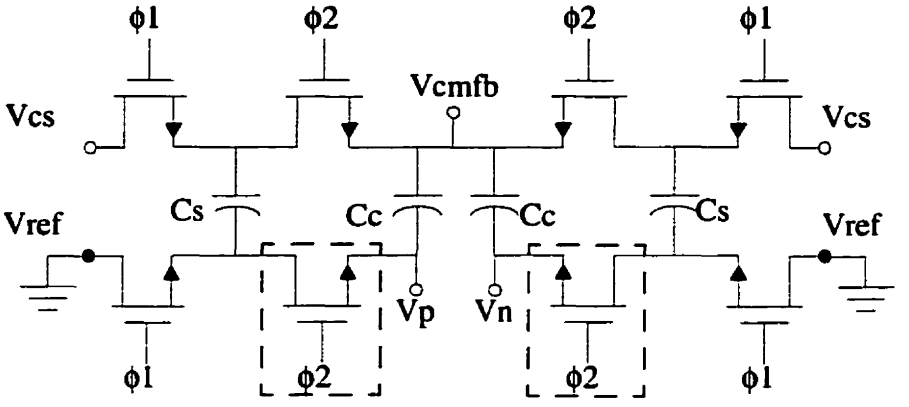


Figure 4.12: Switched Capacitor Common Mode Feedback System

Examining Figure 4.12, one can determine the operation of the CMFB circuit. It is important to note that ideally the voltage $V_{cmfb} - V_p = V_{cs} - V_{ref}$. On ϕ_1, the difference between V_{cs} and V_{ref} is stored on C_s. On ϕ_2, ignoring the start-up transient, the voltage V_{cs} minus V_{ref} is stored across C_c. If V_p and V_n increase, the voltage V_{cmfb} increases, since there is a fixed voltage drop across C_c.

Directing attention to Figure 4.11 for a moment, it can be seen that if V_{cmfb} decreases, then the V_{GS} of M_{10} is decreased. Since a constant current is supplied by the current source, the V_{DS} of transistor M_{10} must increase and thereby raise the output voltage. Therefore, a change in V_{cmfb} has the opposing effect on the output voltage.

As a result, it can be seen that if V_{outp} (V_p) and V_{outn} (V_n) increase, the voltage V_{cmfb} is directly increased, which will cause V_{outp} and V_{outn} to decrease, as prescribed by the corrective loop.

It is important to note that there is a start-up transient in the system, and it occurs as follows: On ϕ_1, assume that C_c has no charge across it, and that C_s acquires the difference between V_{cs} and V_{ref}. On ϕ_2, a weighted averaging occurs based on the size of capacitors C_c and C_s. This procedure is continually repeated throughout the operation of the circuit, so eventually the voltage across both capacitors will be V_{cs} - V_{ref}. This analysis holds true for any initial charge stored on C_c.

By examining this effect and the operation of the CMFB circuit, one can deduce the following:

1. If $C_s > C_c$, the circuit will converge faster, meaning that the CMFB will converge to the right output level quickly.

2. A large C_c will allow the CMFB circuit to track the output level better. This arises because there is less charge injection associated with large capacitors, so the voltage stored across it will be closer to the ideal when C_c is large.

3. C_c directly loads the OTA, and must be accounted for during its design. The CMFB circuit is attached and directly loads the output at all times, however the OTA only drives the circuit on one phase. Therefore ϕ_1 should correspond to the phase in which the driving occurs as it provides less loading.

As a result of these factors, appropriate capacitor sizes ranging from 250 fF to 750 fF, were chosen for the CMFB circuits of the integrators.

4.6.4 Comparator

The purpose of the comparator in this circuit is to implement the one bit quantizer, which provides the output of the modulator. Since the comparator is present after all three of the integrator/gain stages, the non-idealities introduced by it undergo the full noise shaping of the modulator, just like the quantizer noise it produces. As a result, its non-idealitites are reduced the most, thereby easing its design constraints [1, 33].

The choice of comparator architecture was examined in detail. Since there has been no quantitative study done on the requirements of a comparator in a $\Delta\Sigma$ modulator, a conservative approach was taken during its selection. This was done because it was important to ensure that the comparator was not the limiting factor in the modulator's performance.

The two main selection criteria for the comparator were low power, and the ability to resolve a signal with a resolution of 12 bits. Although many comparators were looked at,

only two were seriously considered for the modulator. The trade-offs of each will now be summarized and discussed.

The first comparator that was tested was a dynamic latch comparator. In Figure 4.13, one can see the comparator circuit diagram used. It was modeled after the latch stage found in [69].

A similar comparator was shown to work in a ΔΣ modulator [51], and therefore validated its choice. This comparator was designed and layed out, and the main trade-offs can be seen in Table 4.7. The circuit was simulated with a 10% deviation in the lengths of all the transistors. This resulted in a large drop in its ability to resolve signals (its performance slipped from 14 to 10 bits), which indicated that it was highly sensitive to process variation.

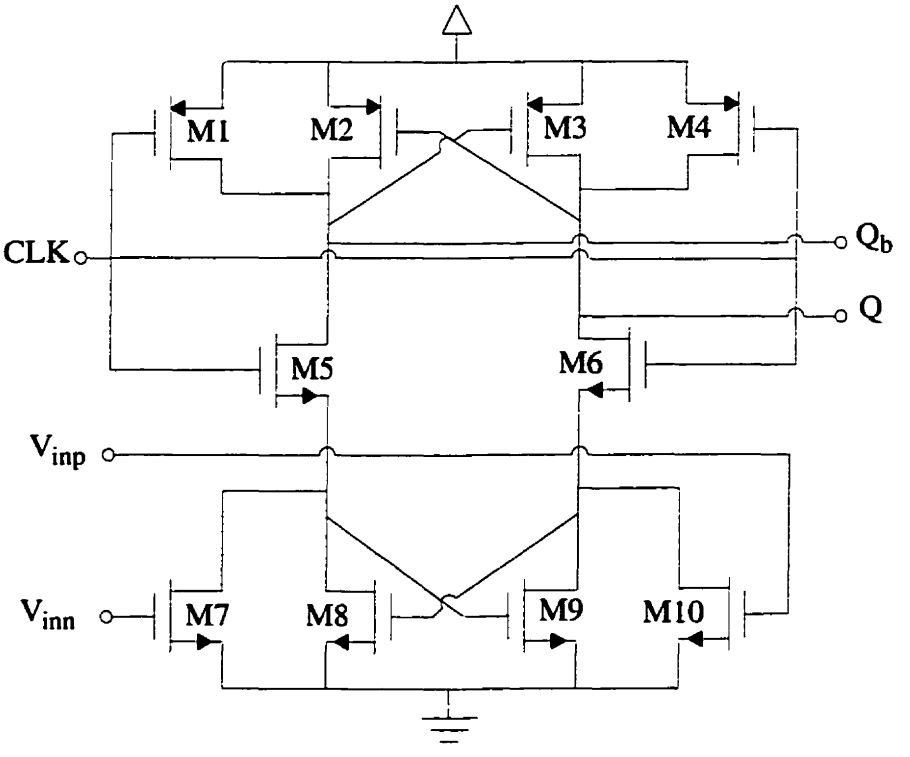


Figure 4.13: Dynamic Latch Comparator

Advantages	Disadvantages
Low Power (180 μW)	High Sensitivity to Parasitics
Simulation before layout had a resolution > 14 bits	Simulation after layout had a resolution = 8 bits
Easy to Build	Highly Sensitive to Process Variation

Table 4.7: Dynamic Latch Comparator Trade-offs

The comparator was actually layed out twice, in an attempt to improve its performance. The first time it was layed out it achieved only 6 bits of performance. This was deemed unacceptable in light of the criteria established earlier. It was determined that the circuit was highly sensitive to the mismatch of parasitics between the corresponding nodes on each side of the circuit (i.e.- the mirrored nodes). As a result, the circuit was re-layed out in an attempt to maintain the same parasitics at each mirrored node. This improved the performance to 8 bits of resolution, which was still below the desired performance. The main advantage of this circuit was its extremely low power consumption of 180μW.

The second comparator that was examined was the latched comparator found in [70], which can be seen in Figure 4.14. The circuit operates with two different phases, the tracking phase and the latching phase.

1. *Tracking Phase:* While the latch is disabled, the pre-amplifier tracks the difference in the input and amplifies it.

2. *Latching Phase:* The pre-amplifier is disabled. The latch when enabled immediately amplifies the output of the pre-amplifier it sees to a digital logic level and holds it.

The presence of the pre-amplifier alleviates the effects of offset caused by process or mismatch in the latch, as the offset is effectively divided by the gain of the pre-amplifier [69]. This was the largest source of error in the previously examined comparator.

Furthermore, the layout of this comparator was carried out, and simulation after layout revealed that it could meet the 12 bits of performance.

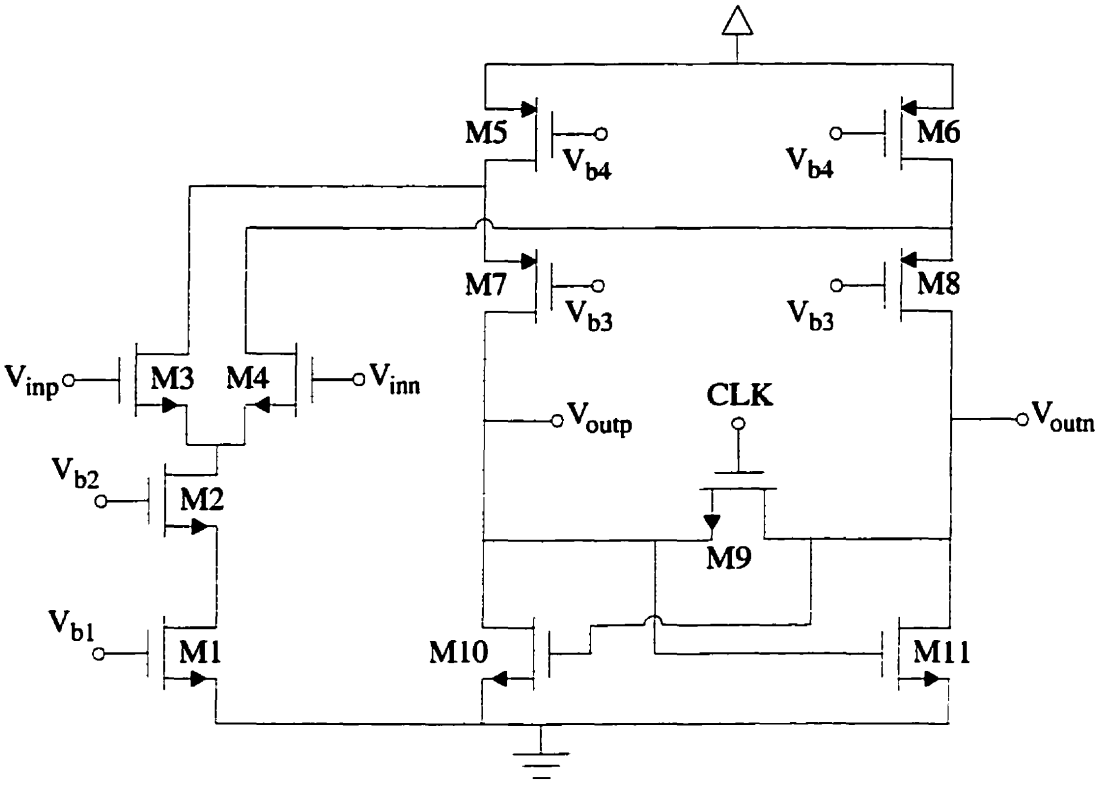


Figure 4.14: Latched Comparator

The use of a pre-amplifier stage necessitated the creation of bias voltages. However, since the architecture is that of a folded cascode, careful designing allowed for the sharing of the bias circuitry with the OTAs already present in the system. As a result, no extra power consumption was needed for a bias circuit. The main trade-offs are summarized in Table 4.8.

Since the power consumption was still considered low in this design, a conservative decision was made and this comparator was chosen to ensure the modulator would not have limitation caused by this comparator.

Regardless of which comparator was used, further digital circuitry was required to complete the circuit. In the case of the comparator that was chosen, buffering (a series of inverters), and an RS latch was required to make the circuit compatible with the system it was integrated into. The buffers were necessary to guarantee that the digital voltage levels

Advantages	Disadvantages
Folded Cascode Architecture allows for shared used of OTA Bias Circuit	Higher Power Consumption (750 μW)
Simulation after layout had a resolution > 12 bits	Harder to Design

Table 4.8: Latched Comparator Trade-offs

are as close to optimal as possible, and the RS latch was used to hold the value of the latch phase during the next tracking phase of the comparator.

4.6.5 Switch Selection

One of the more fundamental building blocks in a SC system is the switch. Its design will be the focus of this discussion.

The main non-ideality associated with a switch is its on-resistance (R_{on}). The on-resistance can be controlled by changing the W/L aspect ratio of the switch, such that R_{on} is minimized by increasing W/L. A large on-resistance can lead to settling problems when a signal is sampled through the switch.

Low voltage design of switches becomes challenging because a reduction in supply voltage increases the switch resistance, since the gate overdrive voltage is reduced for a fixed signal amplitude [52]. Furthermore, effects such as the variation in R_{on}, charge injection, and clock feedthrough complicate the design.

The variation in R_{on} of a switch, can introduce distortion into a SC ADC, mainly through the first integrator. In order to reduce the variation in R_{on}, transmission gates with nmos and pmos transistors in parallel should be used [71]. In order to minimize the on-resistance variation, W/L of the nmos and pmos transistors must be chosen carefully. Traditionally, this is accomplished by selecting a pmos ratio roughly 2-3 times larger than the nmos [53]. Despite this improvement in linearizing R_{on}, the corresponding non-idealities in the two transistors are not completely matched, and variations in R_{on} still exist.

Another non-ideality associated with a switch is clock feedthrough, which is the coupling of the gate (clock) waveform into the output of the switch, due to parasitic capacitance [53]. Since the nmos and pmos transistors are in parallel and require opposing clock signals, the feedthrough due to the clock can be reduced through cancellation. As a result, in order to cancel clock feedthrough, the aspect ratios of the nmos and pmos transistors should be chosen to be equal. It is important to note that this cancellation is not complete because the feedthrough capacitances (gate-to-source and gate-to-drain) in the nmos and pmos are not exactly equal. Also, the turn-on delay is not the same for each type of transistor, meaning that the conductance in the channels do not always track each other when turning on or off.

In a SC system, two switches are used in conjunction with a capacitor, and implements a sampling system. This can be seen in Figure 3.6. In order to reduce clock feedthrough and minimize both the on-resistance and its variation, a trade-off has to be made. Equally sized nmos and pmos were chosen to minimize feedthrough, at the expense of using larger switches to achieve the same maximum R_{on}.

To select a target R_{on} for the design of the switches, an analysis on the distortion due to settling was carried out in section 3.3.2. After some manipulation of Eq. (3.16), a formula for R_{MAX} can be derived to be

$$R_{MAX} = \frac{-T}{4 \cdot C_S \cdot \ln\left(\frac{10^{\frac{-\Sigma}{20}}}{1 + 10^{\frac{-\Sigma}{20}}}\right)} \tag{4.10}$$

where Σ is the precision expressed in dB to which the system must settle, T is the clock period, C_s is the sampling capacitor, and R_{MAX} is the largest on-resistance tolerable. By using R_{MAX}, a conservative estimate is made. For the switches used in the $\Delta\Sigma$ modulator under construction, R_{MAX} was found to be 785 Ω to keep the THD below 100 dB.

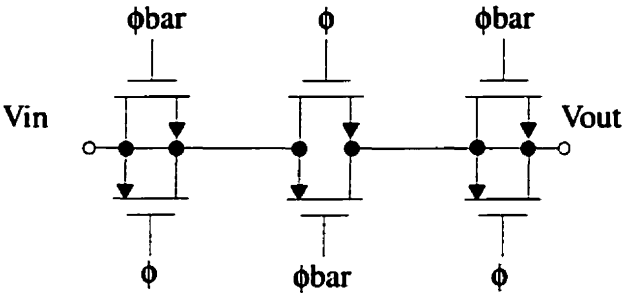


Figure 4.15: Switch Configuration: Transmission Gate with Dummy Structures

Charge injection (discussed in section 4.6.1), is another source of distortion in the switch. The effects of charge injection have been reduced through the use of dummy switches. The switch implemented with dummy switch structures can be seen in Figure 4.15.

4.6.6 Internal DAC

The single bit modulator requires a 1 bit DAC to convert the digital output voltage into an analog feedback voltage. For this design it was achieved by switching in an external reference value from a voltage source, determined by the output of the comparator. The SC switching network used can be seen in Figure 4.4.

There are several important factors for a voltage reference: the absolute accuracy of the voltage, the drift of the voltage, the noise in the reference, power supply rejection, and its ability to drive loads. For a $\Delta\Sigma$ modulator, absolute accuracy and low drift are not required [1]. However, any ripple or noise on the reference will directly limit the overall resolution of the modulator. As a result, it is vital to have high quality/resolution sources.

The reference was implemented by an external DC source supplied from the A567 Mixed-Signal Teradyne tester in conjuction with an active filter. The filter was implemented to provide better noise suppression and further clean up the DC source. The filter used was a two-pole Butterworth filter and can be seen in Figure 4.16.

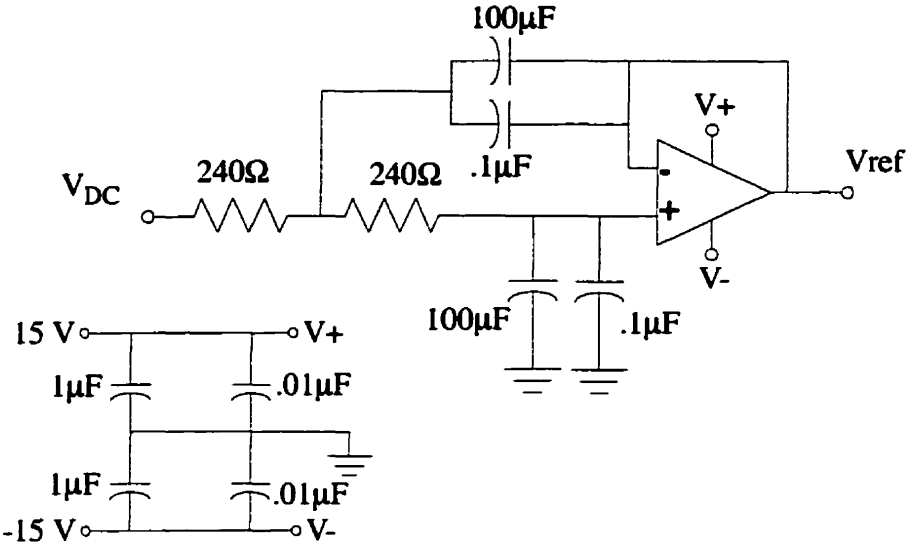


Figure 4.16: External Filters used to Provide Reference Voltages

4.7 Transistor Level Simulation

The simulation of analog sampled data circuits at the transistor level is a heavily time consuming process. The main reason for this is that the operating conditions of the circuit change dramatically depending on the switch configuration on each clock phase. This necessitates the use of transient analysis that lasts for many clock periods in order to get enough outputs points to perform an FFT. Since this system is complicated by many feed-forward and feedback paths, the computational power and time required to perform the simulation is daunting. For all practical purposes, a full transistor level simulation is not possible.

However, HSPICE was used to simulate each of the individual components (OTAs, switches, clock circuitry and comparator) to verify their functionality and performance.

Furthermore, the full system was simulated several times in HSPICE using a combination of idealized models for OTAs with finite bandwidth and gain, an idealized model of the comparator, the transistor level switches, and actual capacitor values. The four phase clocking was also integrated by using the actual transistor level four phase clock generator.

Simulations containing the actual transistor level OTAs and comparator were also attempted, but this exponentially lengthened even the shortest of simulations. In order to achieve a 4096 point FFT required well over 600 hours of simulation time on a Sparc Ultra 10 workstation.

The results of the simulations indicated that the modulator operated properly, however a complete performance verification could not be done. This was a result of the sheer amount of time necessary to conduct a simulation with enough points. Without an adequate amount of points, it is impossible to accurately distinguish the distortion tones from the noise floor of the FFT.

Although the performance of the entire modulator with transistor level devices could not be obtained through simulation, the various simulations conducted throughout the design process provided for a high degree of confidence, allowing the design to be submitted for manufacturing.

4.8 Experimental Results

The modulator has been implemented in a 2.5 volt, 0.25 μm standard CMOS process with a single poly layer and 5 metal layers. A single clock of 6.144 MHz is used to drive the converter. The non-overlapping and delayed clock phases are generated on chip.

The chip was fabricated with the ΔΣ modulator, several individual analog building blocks (test structures including comparators and switches), an unrelated experimental circuit, the pads and ESD protection in a 2.9 mm by 2.9 mm area. This can be seen from the actual chip photograph in Figure 4.17, where the modulator components are highlighted. The actual active area of the modulator and all on chip supporting circuitry (such as the non overlapping clock generator) is 1.44 mm^2, and takes up only slightly more than 17% of the chip area seen in the photograph. A close-up of the modulator can be seen in Figure 4.18.

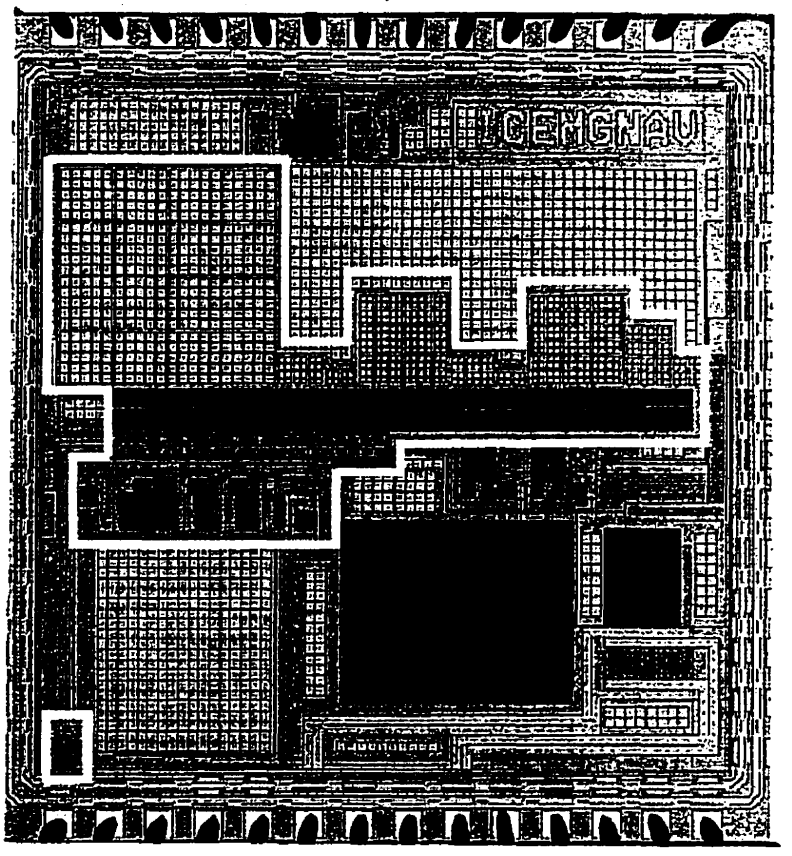


Figure 4.17: Chip Photograph

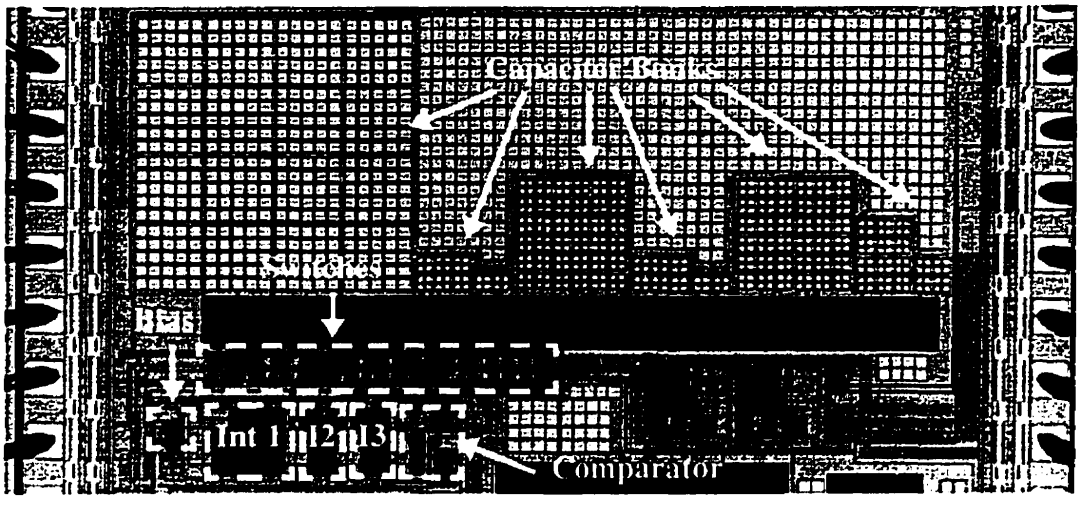


Figure 4.18: Close-up of Modulator

The testing of the Integrated Circuit (IC) was carried out using a Teradyne A567 mixed signal tester, as seen in Figure 4.19. In order to improve the resolution of the DC references and sinusoids output by the tester, some filtering was required. The tester's DC sources, used as references for the modulator were cleaned up using the two-pole Butterworth filter seen in section 4.6.6. The differential input signal was obtained from the tester's precision low frequency source. This input signal, once generated by the tester, was filtered using a Krohn-hite 3988 Filterbox that implemented a lowpass 8th order Butterworth filter with a programmable cutoff frequency. The test setup used can be seen in Figure 4.20.

To measure the performance of a high performance modulator, it is essential to implement proper shielding, grounding and decoupling of the device. As a result, a 6 layer fully custom Printed Circuit Board (PCB) seen in Figure 4.21, was manufactured. This was done

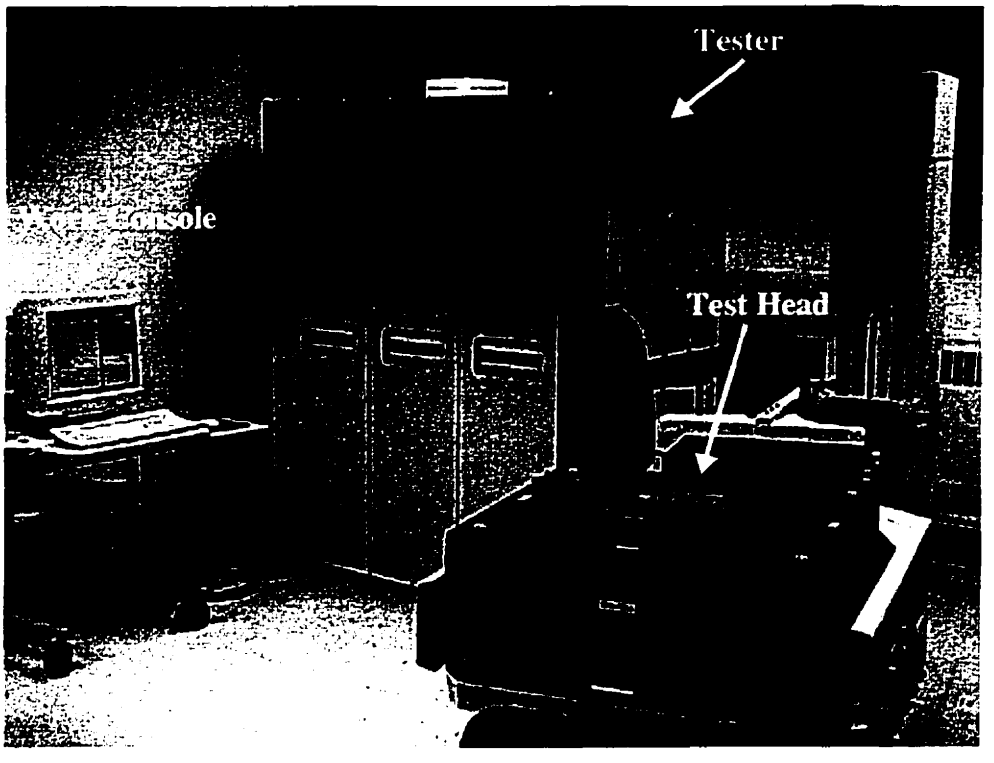


Figure 4.19: A567 Mixed Signal Tester

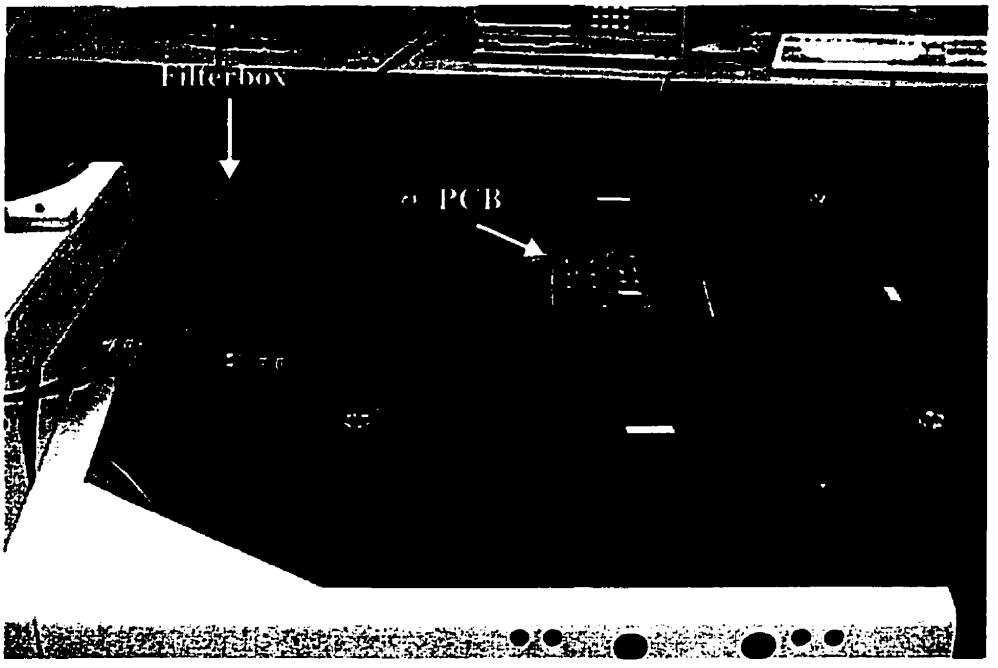


Figure 4.20: Test Setup

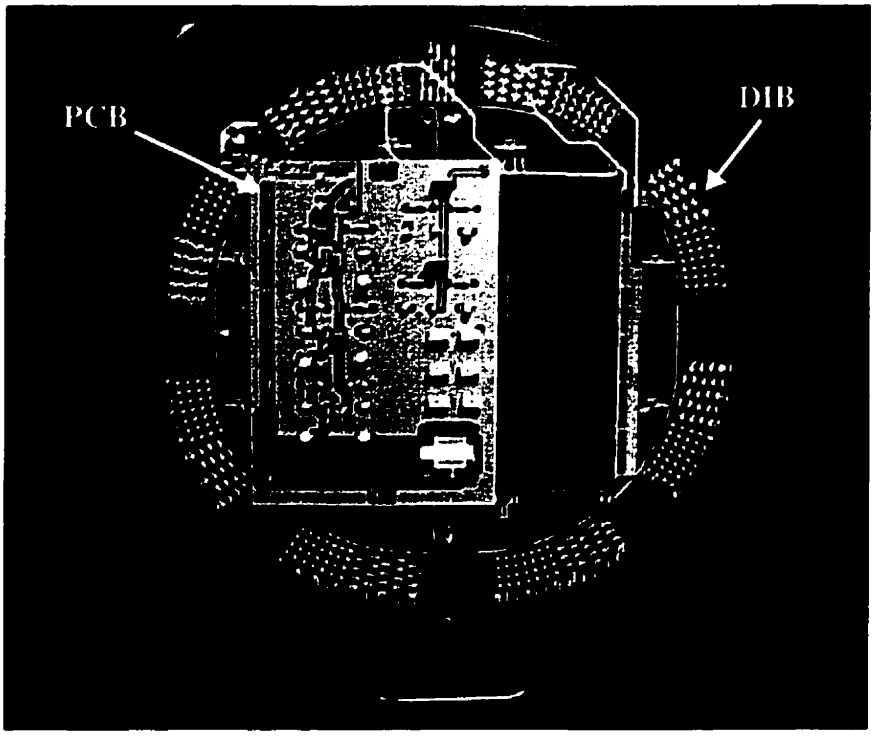


Figure 4.21: Six Layer PCB and Device Interface Board (DIB) Used

to help shield the circuit from external noise coupling, and to separate the analog and digital signals to avoid crosstalk problems.

To evaluate the performance of the modulator, 132 000 points of the output bitstream were captured and stored straight off the PCB by the A567 mixed signal tester's digital capture memory. These points were then exported to Matlab, a 2^{17} point FFT was taken, and the results recorded. This was done for each test run.

A selected sample of the FFTs captured can be seen in Figure 4.22 and Figure 4.23. From these figures, the noise shaping action of the modulator can be seen, along with the thermal noise limitations in the low frequency passband. Figure 4.22(a) and (b) correspond

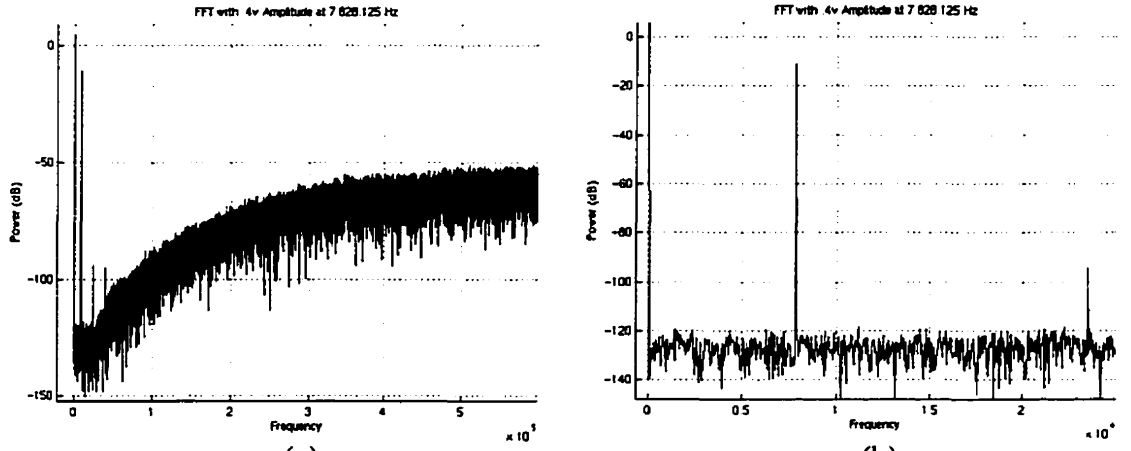


Figure 4.22: Measured Results with a 0.4V Amplitude at 7828.125 Hz

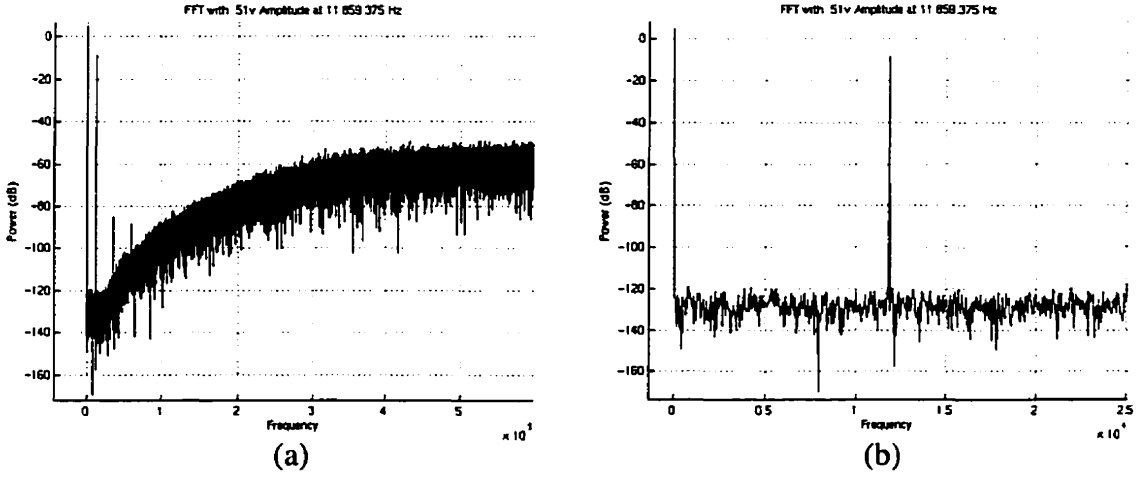


Figure 4.23: Measured Results with a 0.51V Amplitude at 11 859.375 Hz

to a 7828.125 Hz sinewave input with an amplitude of 0.4 volts. From these plots, it is fairly evident that the 3rd harmonic is the fundamental source of distortion and is the limiting performance factor. Figure 4.23(a) and (b) correspond to an 11 859.375 Hz sinewave of 0.51 volt amplitude. In this case, the 3rd harmonic falls out of band, and the system is limited by the thermal noise floor and its associated power.

The lowpass modulator has been designed to operate on signals within a 24 kHz band, which corresponds to the frequencies used in audio applications. In order to determine the useful range of operation for the modulator, the input signal amplitude was varied, and the output signal-to-noise ratio was measured. This was done both with and without the harmonic distortion components included in the useful signal band. The results of this characterization can be seen in Figure 4.24. The dynamic range of this modulator was measured to be approximately 90 dB, with a peak SNR of 94 dB, and a peak SNDR of 82 dB.

These numbers although close, do not meet the target design values. Figure 4.25 shows the non-ideal Simulink simulation that was obtained in section 4.4 overlaid onto the experimental results. From the plot, it is evident that the major difference lies with the presence

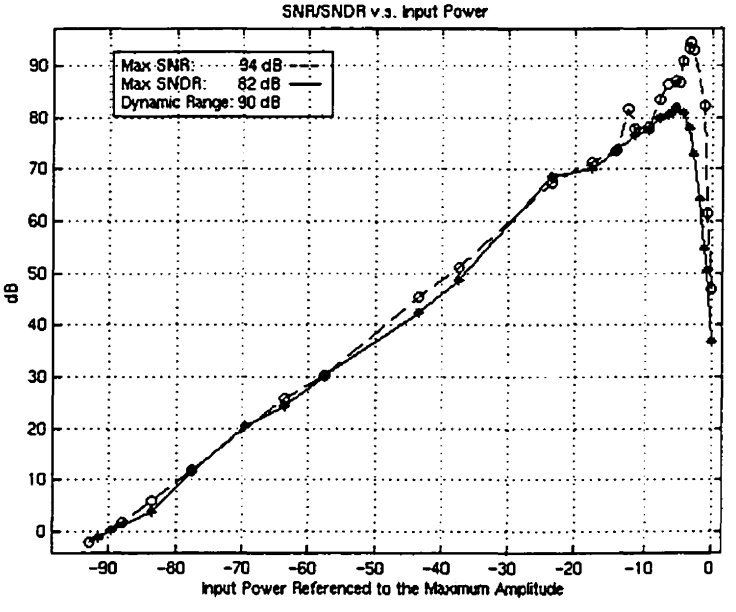


Figure 4.24: SNR/SNDR v.s. Input Power

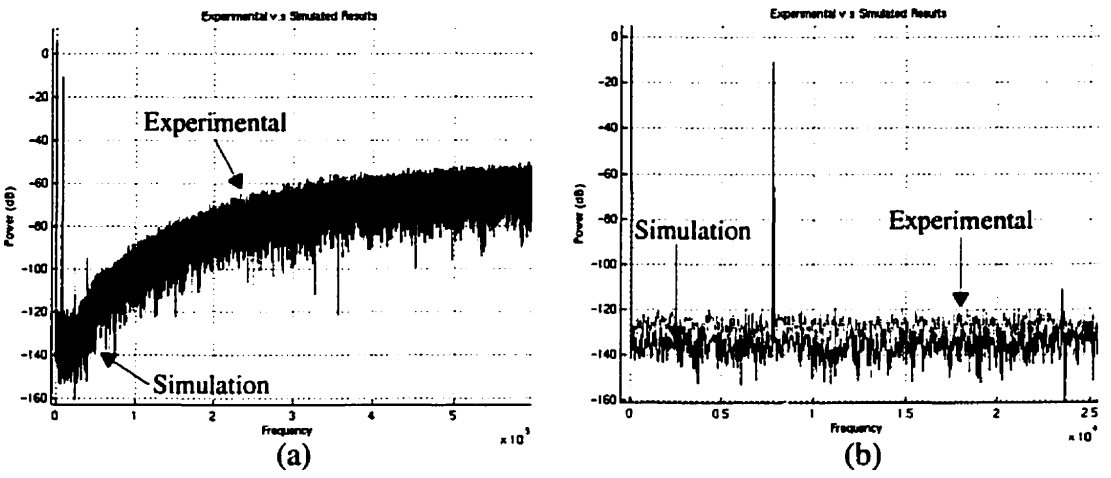


Figure 4.25: Experimental v.s. Non-Ideal Simulated Results

of a more dominant 3rd harmonic. The following discussion will try to address this issue in some detail.

After due consideration, it was hypothesized that the switches could be causing the harmonic distortion seen in the system. Since the IC was fabricated with test switch structures isolated from the modulator, the opportunity to test them was there. Preliminary tests indicated that the switches produced a level of distortion higher than expected. As a result, an HP4145B Semiconductor Parameter Analyzer was used in order to measure the on-resistance of the switches at different input voltages. The experimental setup for this can be seen in Figure 4.26. In order to get a good estimate of R_{on}, 10 different ICs were used and the results from them were averaged. The resulting on curve can be seen in Figure 4.27.

Now that the R_{on} curve has been obtained, it can be seen that the maximum R_{on} exceeds the target value of section 4.4. This immediately re-enforces the notion that the distortion is due to the incomplete settling of the input switches. In order to further quantify this effect, Eq. (3.16) can be used. Recalling that this equation sets a bound on the THD, it follows that when using it, one can get a better approximation of the actual amount of THD caused by the switch if the experimental data of Figure 4.27 is used.

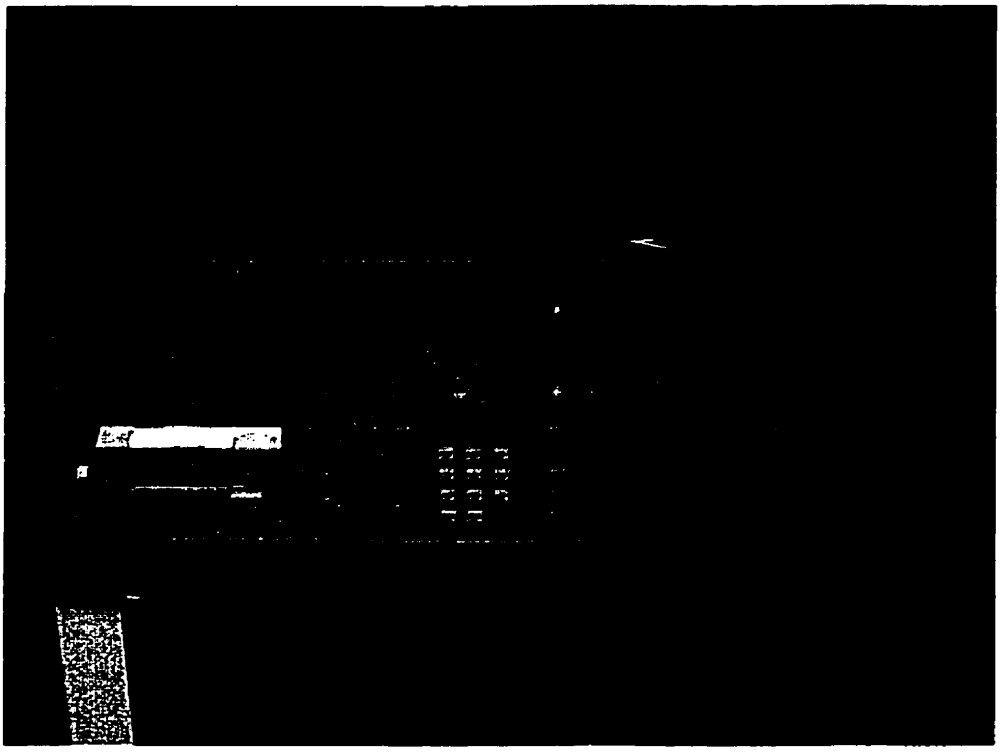


Figure 4.26: HP4145B Semiconductor Parameter Analyzer Used For R_{on} Measurements

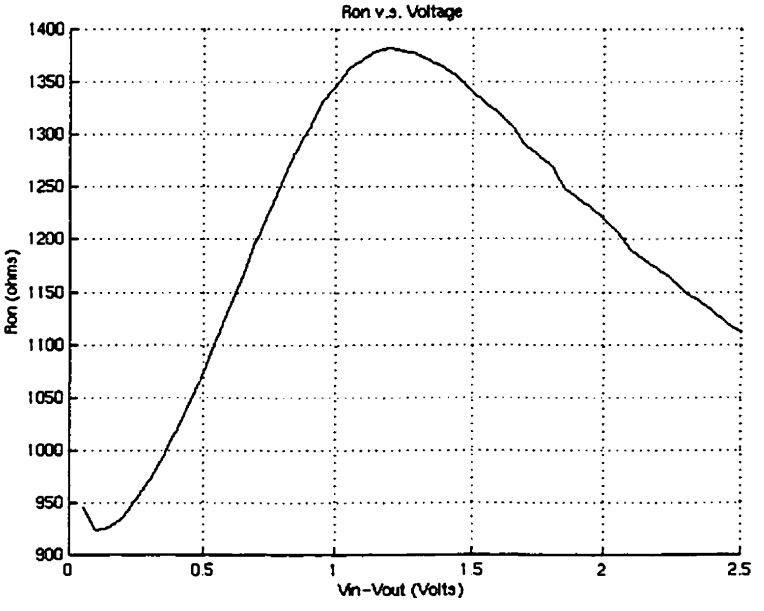


Figure 4.27: Measured R_{on} v.s. Voltage Plot

Examining Figure 3.6, and recalling the discussion in section 3.3.2, one knows that switch S_1 undergoes large signal dependent variations between its input and output. Switch S_4 on the otherhand, undergoes a small voltage variation, with its input set to a fixed voltage. As a result, instead of using R_{MAX} in Eq. (3.16), one can substitute R_{MAX} for $R_{on} = R_{S1} + R_{S4}$, where R_{SX} is the on-resistance of switch x. Since S_4 experiences a small voltage variation, one can use the R_{on} in Figure 4.27 closest to zero volts for R_{S4}. For S_1, the input-output voltage difference is indexed to the input voltage, and therefore the R_{on} for it can be selected accordingly from Figure 4.27 for R_{S1}.

Figure 4.28(a) shows the Simulink FFT output including all of the non-idealities with the more precise R_{on} calculations, along with the experimental results at an amplitude of 0.4 volts. It is evident that because of the unexpectedly high switch on-resistance distortion is expected, and a close-up of the distortion tone is seen in Figure 4.28(b).

The Simulink modeled system delivers a distortion value close to the one that was observed, and validates the design procedure of Chapter 3.

As a result of the preceding analysis, it follows that this problem with the switches can easily be corrected by decreasing the associated on-resistance. This can be done in two ways:

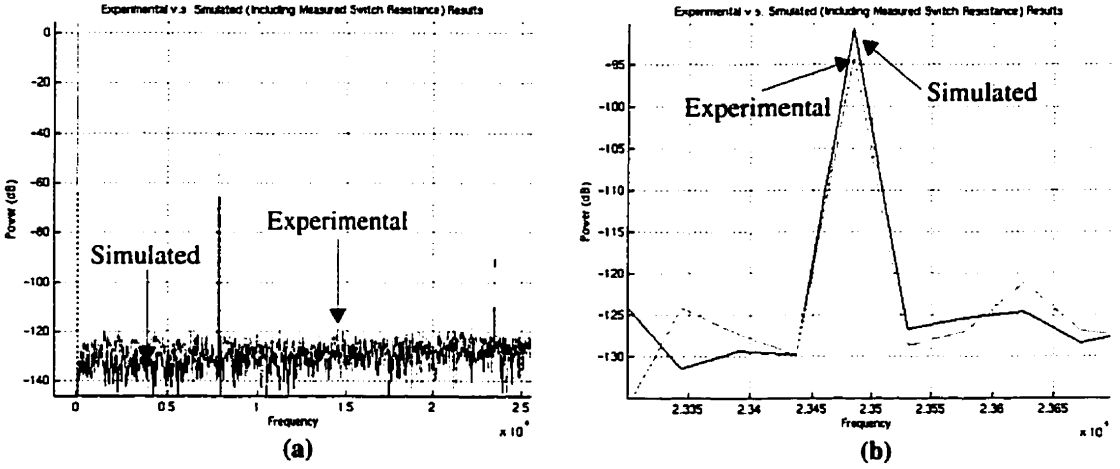


Figure 4.28: Experimental v.s. Simulink Simulated Results Accounting for Measured R_{on}

1. Increasing the W/L ratio of the switches. This has been shown to work in [52].

2. Bootstrapping the clock voltage used to drive the switch beyond the supply voltage range. This alternative has been shown to work in [51].

By implementing one of these two methods the on-resistance of the switches can be reduced, and as a result, the distortion seen can be reduced.

From Figure 4.28, it can also be noted that the measured noise floor is slightly higher than expected. Unfortunately, there is no real method to isolate the reason for this. The rise in the noise floor could be the result of any one, or combination of, the following effects:

1. Process variation on the input stage capacitors. As a result of process variation the capacitors in the first stage of the $\Delta\Sigma$ modulator can vary by as much as 20%. Since they are the dominant contributor to kT/C noise, this variation can lead to a rise in total noise power.

2. Process variation in the differential pair of the input stage OTA. This can cause a rise in the thermal noise associated with it, and as a result cause the noise floor to rise.

3. The test environment. Although external filters were used on all circuit inputs, and a 6 layer PCB was used, the signal quality can still be improved. Integrating the input sinewave filter onto the PCB would certainly reduce some input noise. Furthermore, integrating the fabricated IC directly onto the DIB rather than interfacing with the DIB through a PCB would also help to reduce noise.

Table 4.9 summarizes the experimental performance of the modulator presented in this chapter.

Dynamic Range	90 dB
SNR	94 dB
SNDR	82 dB
Oversampling Ratio	128
Sampling Rate	6.144 MHz
Signal Bandwidth	24 kHz
Input Range	3.0 V_{pp}
Reference Voltages	2.0, 0.5 V
Supply Voltage	2.5 V
Static Power Consumption	5.75 mW
Technology	0.25 µm CMOS
Active Area	1.44 mm^2

Table 4.9: Converter Performance Summary

4.9 Conclusion

This chapter focused on the low voltage design issues involved in the design and fabrication of a lowpass audio-band ΔΣ modulator. The design began by mapping and realizing the modulator created in Chapter 2 in a SC implementation. As a result of using this implementation, several issues were thoroughly investigated, including capacitor layout, sampled capacitor noise, and settling time. Next, Simulink was used to investigate the effects of process variation, used to simulate the non-idealities that most effect a SC system, and used to establish specifications on key design parameters such as C_S, g_m, r_o, I_{bias}, and R_{on}. Following this, design of the low voltage components was discussed. These included the clocking circuitry requirements, the OTA design, the design of the SC CMFB circuit, the comparator selection and design, and the design and selection of the switches. The strategy for design verification was presented throughout the chapter, showing that various successful simulations were conducted that warranted fabrication. The experimental performance of the modulator was then presented, revealing that it achieved 90 dB of performance in a 2.5 V, 0.25 µm standard CMOS process.

CHAPTER 5 Multibit Delta-Sigma Modulation Featuring a Novel Internal DAC

This chapter presents the design, fabrication, and testing of a multibit ΔΣ modulator and encompasses the design and verification of a novel DAC. The modulator uses a multi-bit DAC based on encoding DC levels into digital PDM bitstreams, which are decoded by a single analog filter. The use of a single filtering path reduces mismatch effects in the DAC. The functionality, performance, and limitations of this method are explored by examining the experimental results of a lowpass, 2.5 V, audio-band, 2^{nd} order, switched capacitor, 3-bit ΔΣ modulator in a 0.25 μm CMOS process.

5.1 Introduction

In the design of ΔΣ modulators, it is possible to increase the complexity of the internal quantizer through the introduction of multiple quantization levels. In essence, one can use a multibit over a single-bit quantizer. The use of this more complex quantizer in the realization is referred to as multibit ΔΣ modulation.

There are several advantages that make multibit ΔΣ modulators more attractive than their single-bit counterparts [1, 72]:

1. The quantization noise that results is actually more random and therefore is closer to

the ideal assumption that it can be modeled as white noise.

2. A higher resolution can be obtained for the same oversampling ratio (OSR). This means that in order to obtain a certain resolution, a smaller OSR can be used.

3. The use of a multibit quantizer enhances the stability of the modulator.

4. The reduced quantization noise results in lower out-of-band noise and therefore a less complex decimation filter can be used due to a lower required stop-band attenuation.

Despite these advantages, the use of a multibit DAC presents another, possibly more detrimental, source of distortion. This is due to the fact that any non-linearity in the internal DAC adds directly into the first stage of the modulator, making it the equivalent of input signal distortion. As a result, the linearity of the DAC has to be comparable to the overall linearity of the modulator [73].

The non-linearity in multibit DACs is due to the mismatches between the various analog components needed to produce the output levels. In order to improve the resolution of the internal DAC, several solutions have been proposed:

1. The use of external custom trimmed resistors to eliminate mismatch [74].

2. The use of architectures which are less sensitive to the accuracy of the internal multibit DAC [75].

3. The use of dynamic element matching (DEM) to randomize the DAC errors thereby converting harmonic distortion into flat wideband noise [76].

4. The use of first-order noise-shaping techniques on the noise caused by DAC mismatches [77].

5. The use of higher-order noise shaping for the noise caused by the DAC [78-81].

6. The use of a single-path multibit DAC to remove the component mismatch by using the same analog component for all reference levels [82].

This chapter will present a novel device based on the principle presented in [82] concerning a single-path multibit DAC, and features the use of a DC voltage generator similar to the one presented in [83]. Its performance and use will be evaluated in a lowpass, audio-band, multibit ΔΣ modulator.

5.2 The Multibit DAC

Through the use of the same analog component to produce all of the output levels of a multibit DAC, component mismatch errors can be removed, and the accuracy of the DAC improved. In order to implement this, the DAC input can be converted to a bitstream that encodes the appropriate DC level, which then can be extracted/filtered using a single analog component. An illustration of such a system can be seen in Figure 5.1.

The approach taken in [82] was to use Pulse Width Modulation (PWM) bitstream encoding in a digital system of length N, clocked much faster (350 times) than the modulator. As a result, a series of tones at multiples of F_s/N is produced at the output [84], and then lowpass filtered in order to extract the proper DC level.

This approach suffers from a serious speed limitation. Particularly, if the sampling frequency of the modulator is 6 MHz, a clock of 2.1 GHz, and correspondingly fast digital circuitry, would be needed to implement the system. Although possible to implement, this is not an easy task to undertake.

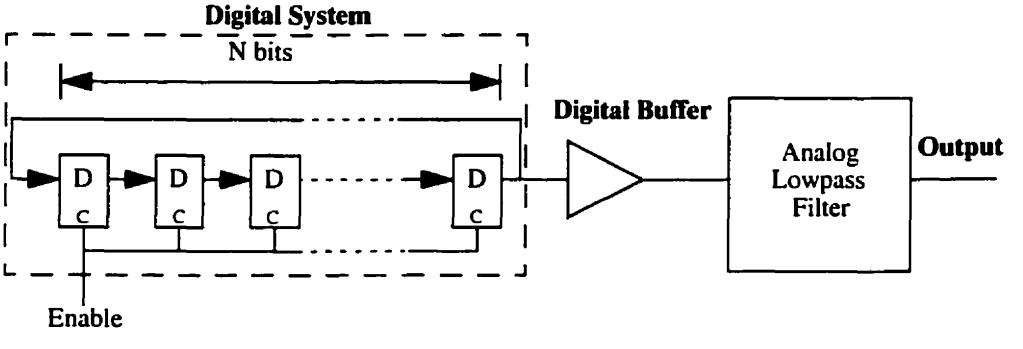


Figure 5.1: High Level Realization of Single-Path DAC

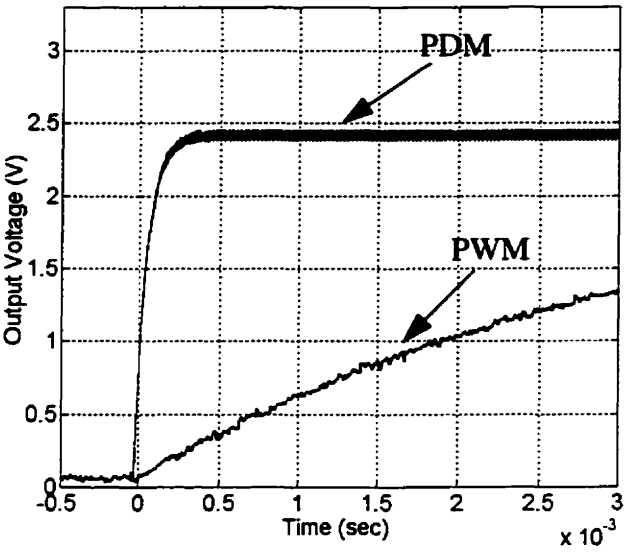


Figure 5.2: PDM v.s. PWM Settling

The multibit DAC proposed here combats that speed disadvantage in two ways. First, PDM instead of PWM bitstreams are used to encode the DC levels. PDM encoding benefits from an improved settling over its PWM counterpart. This can be seen in Figure 5.2, where the step response of a PDM versus a PWM stream can be seen for the same DC encoded voltage.

Secondly, it is beneficial to remove the clocking requirement on the digital circuitry. Instead of using the clocked D-flip-flop circuit of [82], an asynchronous cell can be used. This will allow the PDM bitstream to cycle at the fastest speed possible for a given technology. This means that the bitstreams can free-run in the GHz vicinity (for a 0.25 μm CMOS process) without the need for a dedicated clock. The higher in speed at which the bitstream is cycled helps translate into AC tones that are located further up in frequency. This means that the settling time requirement can be more easily met as less lower frequency attenuation must be provided by the filter to achieve the same resolution, allowing the poles to be pushed further out.

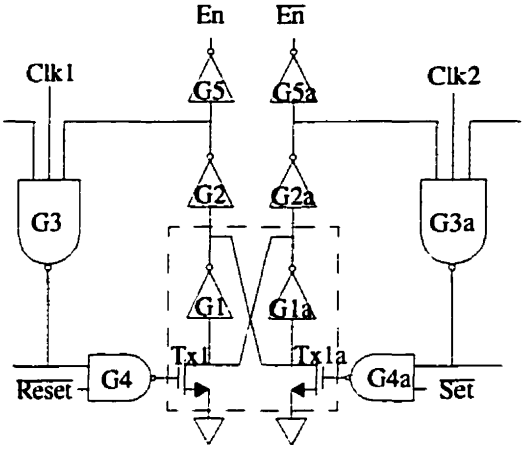


Figure 5.3: Asynchronous Cell Used in Bitstream Generator Implementation

5.2.1 Periodic Bitstream Generator

The bitstream generator is created by arranging a series of fully differential memory cells or asynchronous latches together in a circular manner thereby forming a scan chain, where the output of one of the memory cells is used to provide the bitstreams. In order to maximize the operating speed of the generator, an asynchronous cell found and designed according to the guidelines in [85] was used, and can be seen in Figure 5.3. Its free-running speed was determined to be approximately 1.2 GHz in simulation.

5.2.2 Filter Selection

In order to extract the analog level from the DC bitstream, a lowpass filter with a specific attenuation factor is needed. This is to remove unwanted high frequency noise and to scale the voltages to be within the linear range of the opamps in the circuitry that it will be interfacing with.

As digital signals are immune to mismatches in digital components, the only source of mismatch in the DAC is the analog lowpass filter used to convert the digital signal into a DC level. As a result, the accuracy of the DAC is dependent on the type of filter used. Three categories of filter were considered and their trade-offs examined.

i) Passive RC Ladder Filters

The main advantage of using a passive RC ladder is that it consumes no static power, making this approach very appealing. Also, since only passive components are used, no signal distortion is caused through its use. Furthermore, since only the DC tone is conserved, the higher the order of the filter, the smaller the size of resistors and capacitors needed to implement the RC ladder. This fact can be used to help meet area constraints.

Unfortunately, there are several disadvantages as well. Primarily, these filters are very difficult to design as the calculations needed to synthesize RC networks [86, 87] are quite intensive and require a great deal of effort. Furthermore, only real poles can be created with a ladder structure, thereby limiting the implementable frequency responses of the filter.

On top of this, because the ladder topology inherently has a unity gain, more latches are needed in the bitstream generator in order to create the required DC levels. For example, assume that in a 3-bit system the DC levels can be created with seven (2^b - *1*) latches in conjunction with a system gain of 1/5 about midrail. As a result, with a power supply of 2.5 V the DC values of 1.0 V, 15/14 V, 8/7 V, 17/14 V, 9/7 V, 19/14 V, 10/7 V, and 1.5 V can be easily created. Using an RC ladder network, 35 latches would be needed to create the same required DC levels.

Finally, RC ladders are greatly affected by loading and changes in component values. The relationship between the poles and component values is very complex. As a result, process variation can cause drastic changes in the pole locations and thereby alter the frequency response of the filter. Since the changes in each capacitor and resistor affect all of the poles, ensuring the proper filter response over the range of possible variations is tricky.

ii) Active RC Filters

The main advantage of this filter type is that the minimal amount of latches (2^b - *1*) can be used in the bitstream generator. This is because an active RC structure can easily be designed to deliver any desired gain. Furthermore, due to its ability to implement complex poles, an active RC filter can potentially obtain faster settling times than its equivalent or-

der passive RC ladder filter. In addition, it is less sensitive to process variation. Since each section of the active RC filter creates a pole and is buffered from the previous and following stages (due to the presence of an opamp), the effects of process variation are easier to quantify and consider in the design process.

The main drawback of this method is static power consumption. Since one opamp is needed per order of the filter, this method involves much higher static power consumption. Also, the presence of an opamp can introduce distortion, meaning that the DC levels produced must be inside its linear range. The opamp causes another problem as well. The quick cycling time of the bits in the flip-flops combined with the rail-to-rail digital levels provide a large, sharp transition at the input of the opamp. This can be very problematic and cause further distortion.

iii)Single Amplifier Biquadratic Filters (SAB)

A SAB filter is a method of implementing a 2^{nd} order filter section using an RC filter in combination with an opamp. As a result, it only uses half of the static power of its active RC equivalent. Furthermore, the fact that an RC section is present in the signal path before the opamp allows for the slowing of the transition edges, thereby facilitating its processing by the opamp. These factors make the SAB filter more appealing than a strictly active RC implementation. Although subject to distortion due to the active opamp components, it also benefits from the ability to implement any gain factor easily, and therefore use the minimum amount of latches for the bitstream generator.

After careful consideration of these factors, it was determined that the filter would be implemented using an SAB structure. To obtain 12 bits of accuracy it was necessary to design for 8.6τ settling. For a 1.536 MHz clocking frequency, this translated into a 37 nsec settling time. In order to achieve this, a 4^{th} order filter was needed, and can be seen in Figure 5.4.

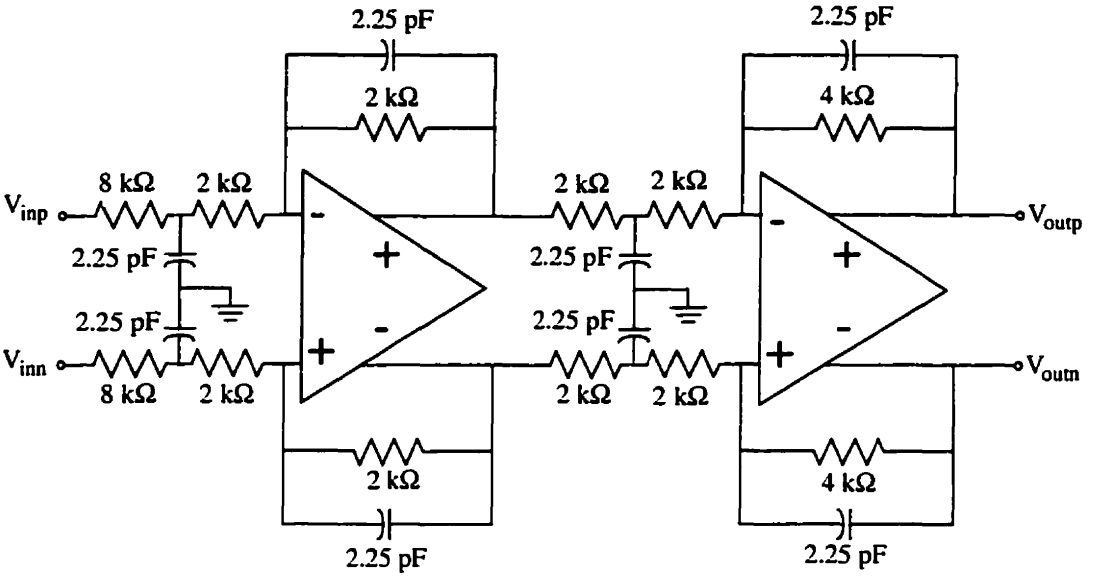


Figure 5.4: 4th Order SAB Filter

Figure 5.5 shows the step response of the designed filter. One can note that the maximum settling time τ was designed to be 21.9 nsec, and the maximum ripple which determines the accuracy of the signal was kept to within 12 bits. It is important to note that although the DAC will be operated at 1.536 MHz, in actuality it has been designed to operate at twice that frequency (3.072 MHz). This is necessary due to the nature of SC $\Delta\Sigma$ modulation, in which the reference voltage created by the DAC needs to be determined

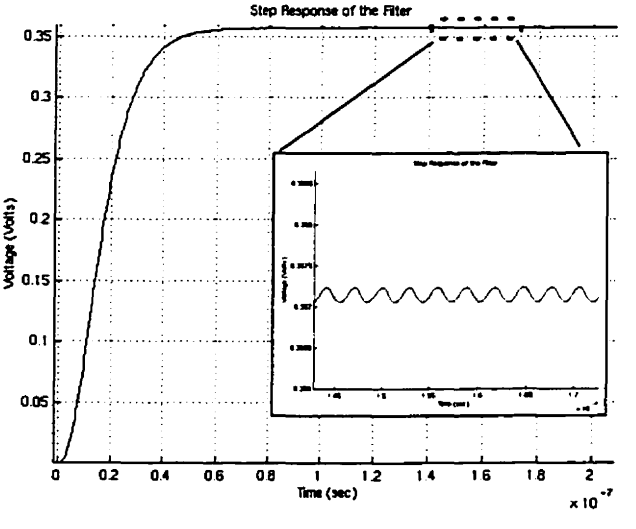


Figure 5.5: Step Response of the SAB Filter

in one clock phase. Consequently, the DAC delivers its results in one clock phase and holds the output on the other (essentially remaining idle). This means that the DAC is capable of digital conversion at a rate of 3.072 MHz.

5.3 Multibit ΔΣ Modulator Design

In order to apply and test the performance of the proposed DAC, a multibit, lowpass, audio-band, 2^{nd} order, ΔΣ modulator with an OSR of 32 and a clocking rate of 1.536 MHz, was designed to make use of the component. The architecture of the modulator was introduced in [72], and the high level signal flow graph can be seen in Figure 5.6.

The system was simulated over a wide range of amplitudes in Simulink using ideal components and an ideal DAC. The corresponding SNDR was recorded for each amplitude, and can be seen in Figure 5.7. From this plot, the dynamic range was determined to be 83.6 dB, and the maximum SNDR was 79.2 dB. A 2^{17} point FFT of this system with an input amplitude of 83.5% of the maximum allowable amplitude at a frequency of 1957.03125 Hz can be seen in Figure 5.8.

In order to more accurately determine the achievable performance of the modulator, Simulink was used to model the non-idealities. Furthermore, an optimization routine was

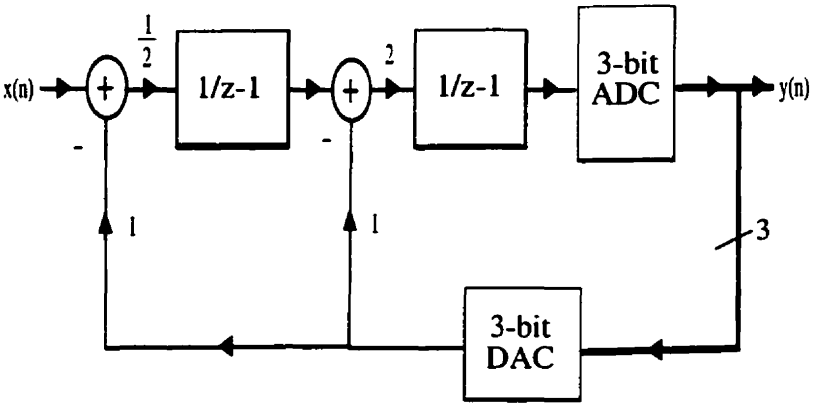


Figure 5.6: Signal Flow Graph of 2^{nd} Order Multibit Modulator

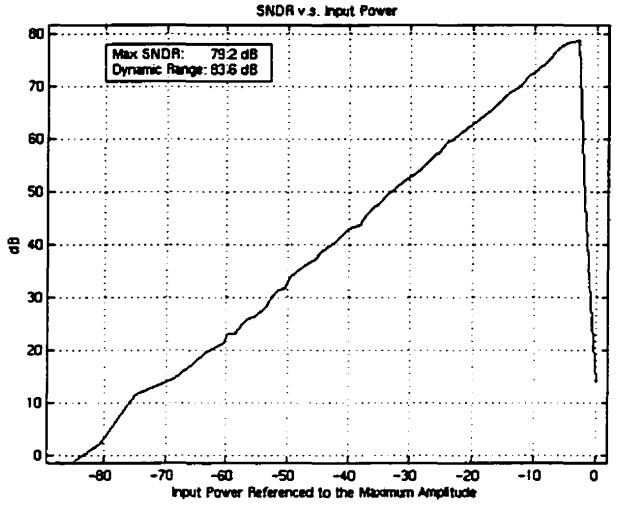


Figure 5.7: SNDR v.s. Input Power for the Ideal Simulink System

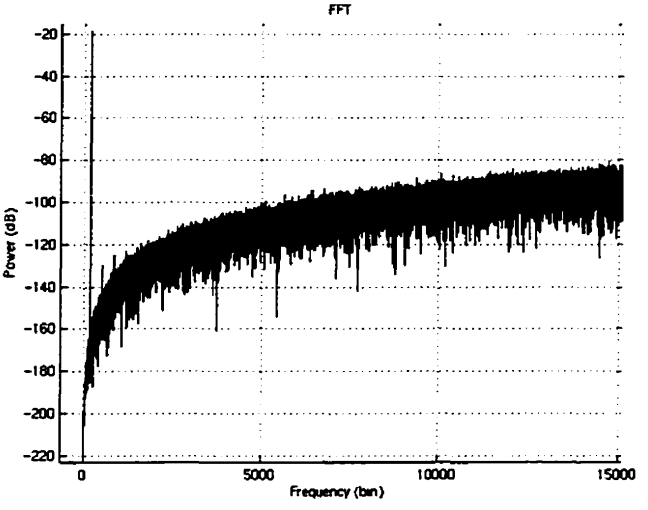


Figure 5.8: Frequency Response of the Modulator when SNDR is Maximized

Parameter	Value
C_s	2.5 pF
I_{bias}	300 μA
g_{m1}	0.59 mA/V
C_c	2.5 pF
$gm6*r_{o1}*r_{o2}$	4.24 MV/A

Table 5.1: Key Design Parameters

used in order to establish design guidelines on key circuit parameters as seen in Chapter 3. After optimization, the non-ideality model's key parameters were determined, and can be seen in Table 5.1.

Following this, Simulink simulations were run using these parameters. The results of the non-ideal simulations produced a dynamic range of 80.6 dB, and a maximum SNDR of 77.6 dB. A plot showing SNDR versus input power relative to the maximum amplitude can be seen in Figure 5.9. Based on the results obtained from Simulink, the modulator was implemented in a SC realization using the guidelines in section 4.1. The fully differential SC realization can be seen in Figure 5.10.

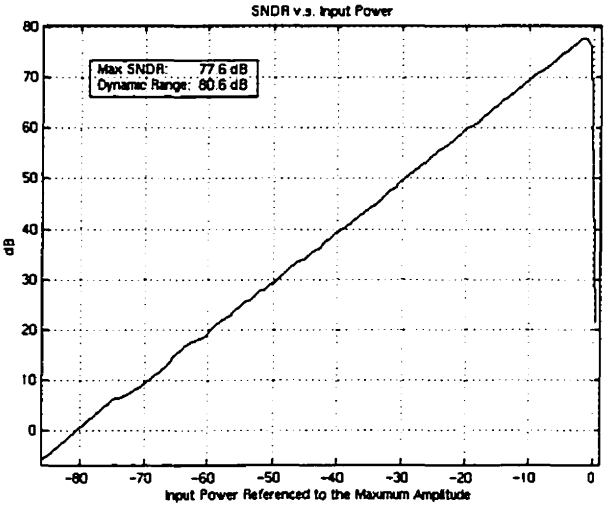


Figure 5.9: SNDR v.s. Input Power for the Non-Ideal Simulink System

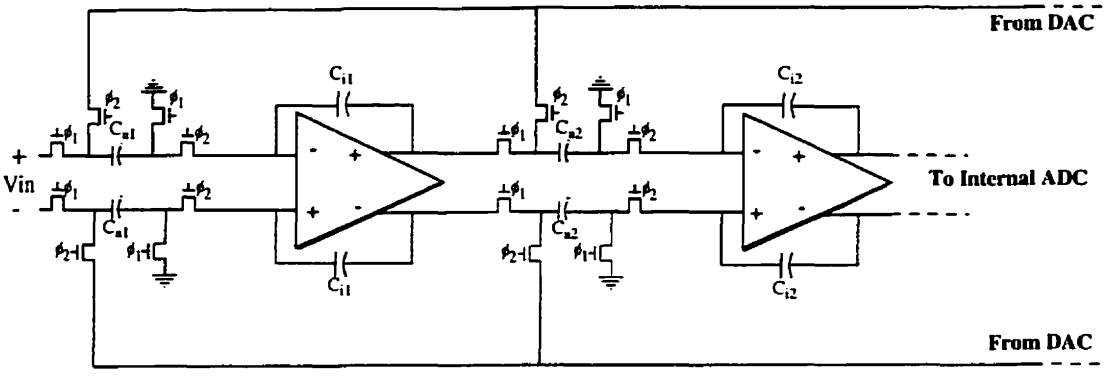


Figure 5.10: SC Implementation of the 2nd Order ΔΣ Modulator

5.4 Design Issues and Components

The design issues and components used in the implementation of this multibit ΔΣ modulator were the same as those discussed in sections 4.3 and 4.6. Equivalent analysis was carried out on the circuit of Figure 5.10 to determine suitable formulas and values where necessary. There were however a few exceptions such as the opamp, the internal ADC (quantizer), and the internal DAC. The design of these components will be summarized in this section.

5.4.1 Operational Amplifier

A fully differential two-stage class A operational amplifier with a source-follower output stage was designed and can be seen in Figure 5.11. The buffering output stage was used in order to drive the internal ADC. As a result of the already present output stage, a continuous time CMFB circuit was used, and can be seen in Figure 5.12.

The opamps were created based on the guidelines in Table 5.1, and the resulting specifications after layout can be seen in Table 5.2. It is important to note that some of the values from the table were altered in the design process if it resulted in an overall improvement in

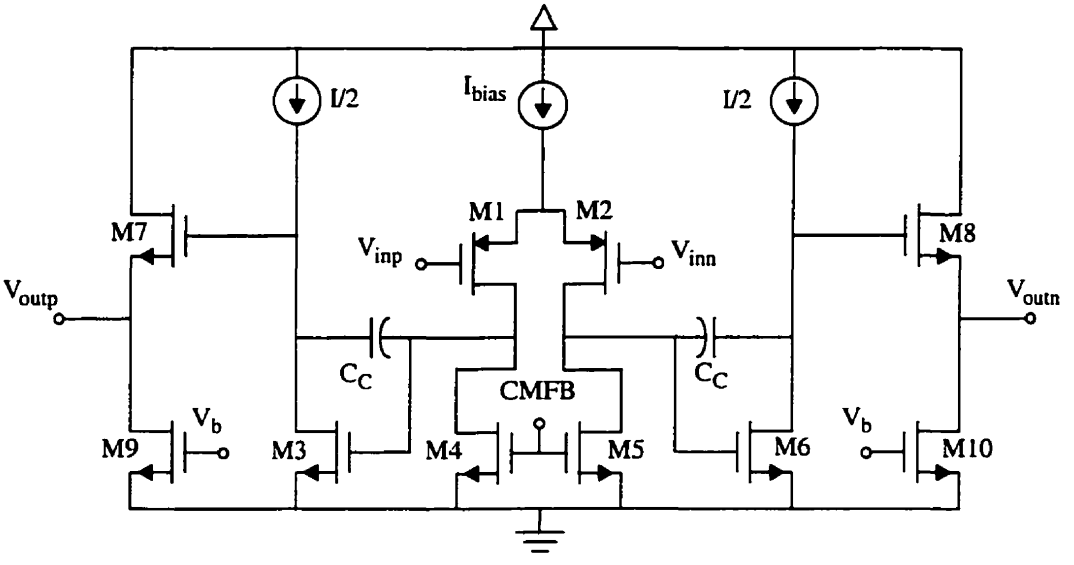


Figure 5.11: Two-Stage Class A Amplifier with Output Buffer

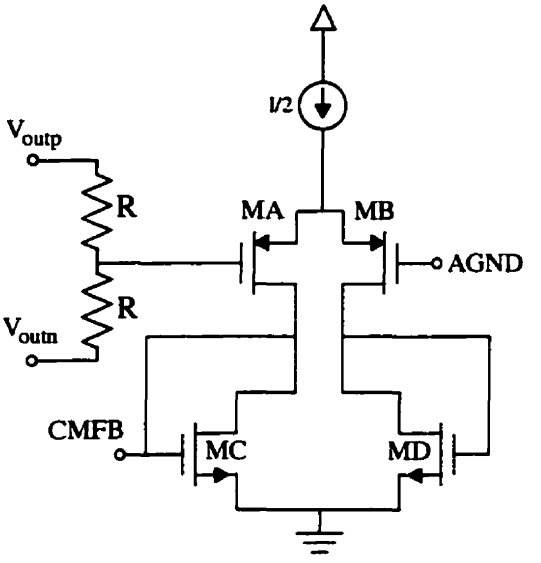


Figure 5.12: Continuous-Time Common Mode Feedback Circuit

Specification	Integrator 1
Gain	4849 V/V
UGBW	53.65 MHz
Slew Rate	182 V/μsec
Phase Margin	57.5 degrees
I_{bias}	250 μA
Power (with CMFB)	4.75 mW
C_c	1.75 pF
Load	2.5 pF

Table 5.2: Characterization of the Opamp

the opamp. In order to save power, the same bias circuit was shared among both integrator opamps and with the opamps used in the SAB filter.

5.4.2 Internal ADC (3-bit Quantizer)

The modulator's performance is very tolerant of the non-linearity and hysteresis in the internal A/D converter because these effects are reduced through second order noise shaping. As a result, there is no stringent design requirement on the 3-bit ADC.

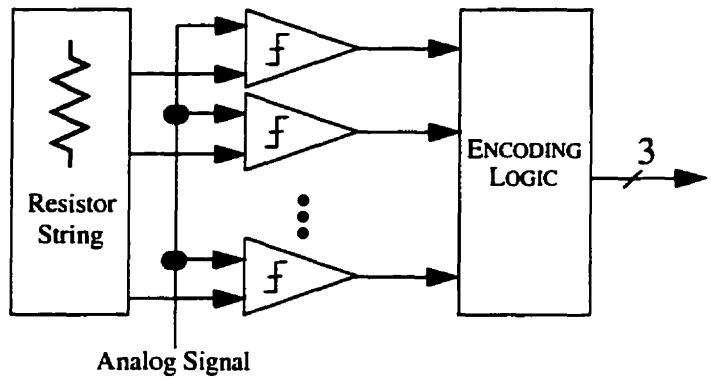


Figure 5.13: High Level Realization of the Internal 3-bit ADC

The 3-bit quantizer was implemented using a 3-bit flash ADC architecture, which makes use of a resistor string to set the comparison threshold levels, and seven differential comparators (see section 4.6.4). The ADC was modeled after the one presented in [72], and a high level model of the internal ADC can be seen in Figure 5.13. The output of the seven comparators is converted into a 3-bit word through the use of simple combinational logic. In order to encourage high accuracy of the ADC, the resistor string was carefully layed out. This was done in order to promote good matching properties, and thereby maintain a more linear ADC. The complete quantizer was simulated at the transistor level in HSPICE to verify its proper operation.

5.4.3 Internal DAC

The internal DAC was based on the principles outlined at the beginning of this chapter. Its components were described in sections 5.2.1 and 5.2.2. All that remains is its integration into the $\Delta\Sigma$ modulator.

In order to properly integrate the DAC into the modulator, an interface between the output of the internal ADC and the DAC needs to be implemented. Since a 3-bit code is produced at the output of the internal ADC, this digital signal is used as part of the combinational logic control system, which presents and loads the proper hardcoded inputs to the bitstream generator. These inputs are summarized in Table 5.3.

Bitstream	DC Level	Filter Gain	DAC Output (V)
0000000	0	1/5	1.0
0001000	1/7	1/5	1.0714
0010010	2/7	1/5	1.1428
0010101	3/7	1/5	1.2143
1101010	4/7	1/5	1.2857
1101101	5/7	1/5	1.3571
1110111	6/7	1/5	1.4286
1111111	1	1/5	1.5

Table 5.3: DAC Bitstream Encoding

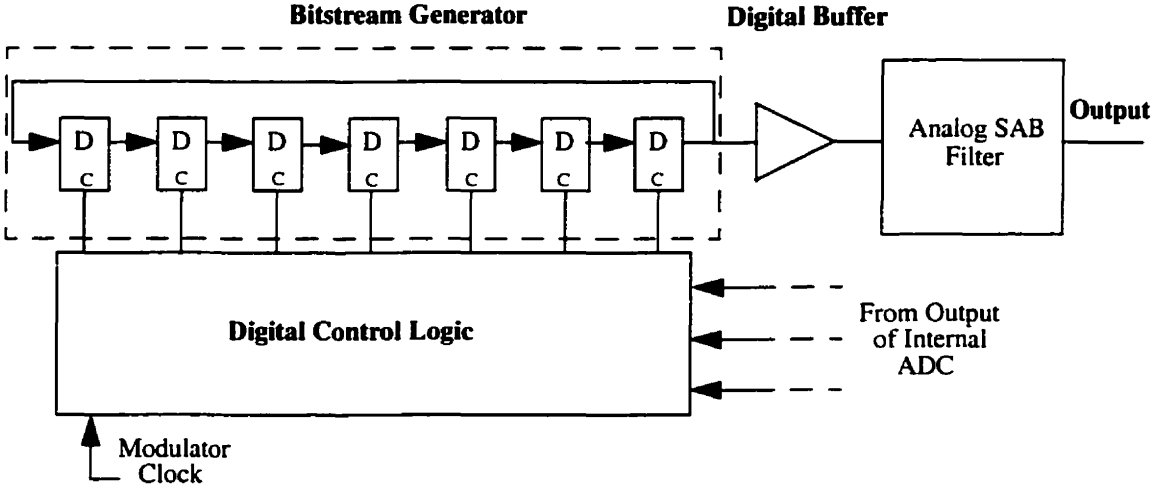


Figure 5.14: High Level Representation of the Internal DAC Using in the ΔΣ Modulator

A high level representation of the system can be seen in Figure 5.14. It is important to note that the delay through the control logic must be accounted for to ensure adequate settling time through the filter.

The differential outputs of the filter are directly used as the feedback references, and no interface is needed. Finally, the internal DAC was extensively simulated at the transistor level in HSPICE. It was verified to produce the desired reference voltages with the needed accuracy.

5.5 Experimental Results

The modulator including the DAC was implemented in a 2.5 volt, 0.25 μm standard CMOS process comprising of a single poly layer and 5 metal layers. A single 1.536 MHz clock was used to drive the converter, with the non-overlapping and delayed clock phases generated on chip.

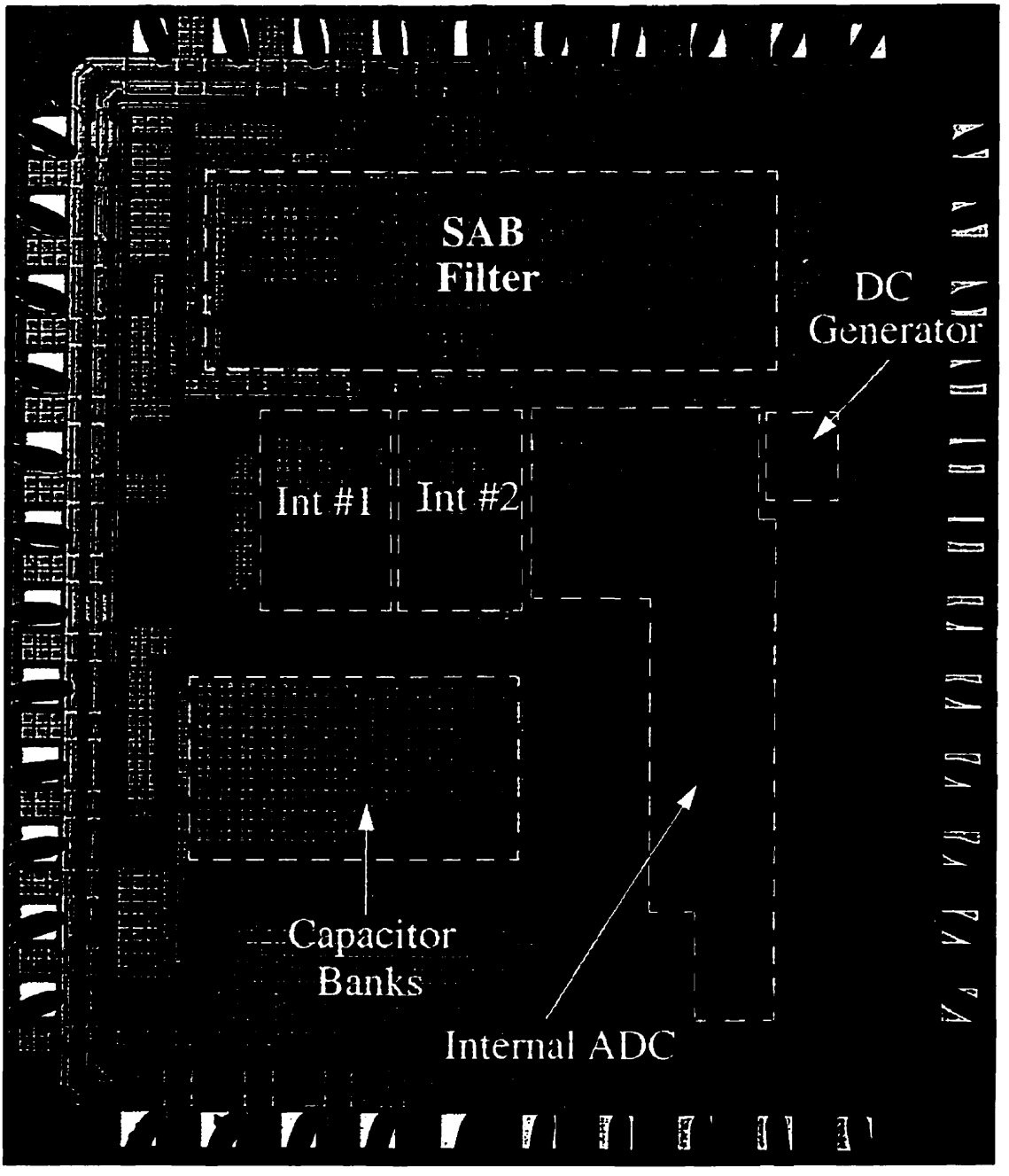


Figure 5.15: Chip Photograph

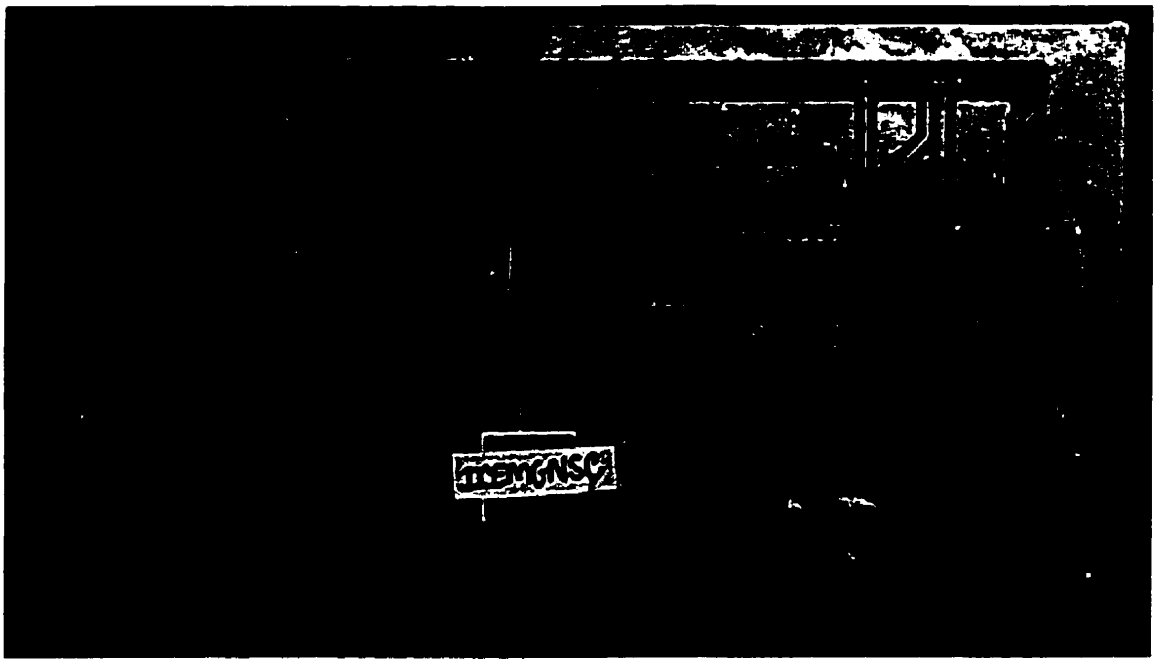


Figure 5.16: PCB Board Used to Interface with the A567 Mixed Signal Tester

The entire system including pads and ESD protection was fabricated in a 2.9 mm by 2.4 mm area. This can be seen from the actual chip photograph in Figure 5.15, where various modulator components are highlighted. the actual area of the modulator and all on-chip supporting circuitry is 2.7 mm^2, with the internal DAC taking up 0.65 mm^2.

The testing of the IC was carried out using a Teradyne A567 mixed signal tester, and a PCB board was built to interface with the chip. This can be seen in Figure 5.16. The testing procedure will be explained in the sections that follow.

5.5.1 DAC Testing

It is important to mention that due to chip space limitations, there was not enough area to replicate the DAC for testing purposes. Rather, some design for testability techniques were used to give access to several of the DAC's analog nodes by wiring them to pads through on-chip switches. These switches were then enabled while in test mode.

Testing the DAC independently from the modulator is rather problematic. This is due to the inherent difficulties of correctly extracting and reading an accurate analog DC level

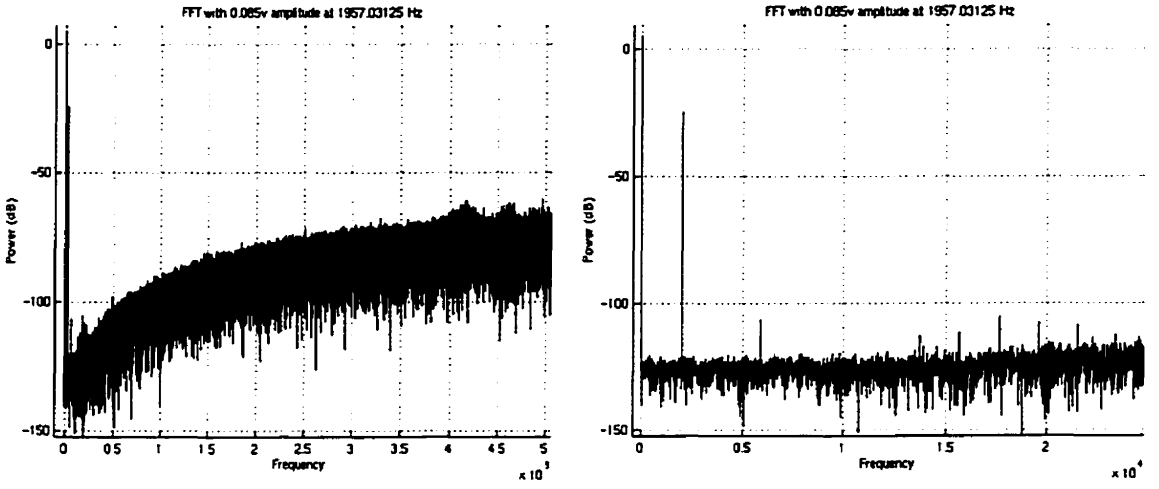
off of a chip. Although the A567 mixed signal tester can accurately measure a DC voltage to the precision needed to validate the design of the DAC built here, delivering the value actually produced by the DAC to the tester is quite challenging. Aside from the noise associated with the pads, the noise on the PCB traces used to carry the output voltages, and the offset and noise voltages of the buffers used on the PCB board, an additional source of noise and distortion was present when reading the output voltages. Namely, the on-chip switches used to deliver the signal to the pads introduced thermal noise onto the signal along with a voltage dependent error (or distortion). This distortion is caused by the voltage dependence of the on-resistance, and essentially corrupts the output voltages. More importantly, the voltage dependence affects the linearity of the measured results, thereby corrupting any linearity measurements. Consequently, since the linearity of the DAC has to be comparable to that of the overall modulator [73], the performance of the modulator will be used to evaluate the performance of the DAC.

5.5.2 Multibit ΔΣ Modulator Testing

The testing of the modulator was carried out in a procedure similar to the one used for a single bit modulator of Chapter 4. The performance of the modulator was evaluated by capturing 132 000 samples of the 3-bit parallel output with the A567 mixed signal tester. These points were then exported to Matlab where a 2^{17} point FFTs were taken for each test run.

The noise shaping action and thermal noise limitations of the modulator can be seen in Figure 5.17. Figure 5.17(a) and (b) show a 1957.03125 Hz sinewave input with an amplitude of 0.085 volts. It is quite evident that there are multiple harmonics which effectively limit the modulator's performance.

In order to determine the usefulness of the multibit modulator, its SNR and SNDR values were obtained for various input levels. This can be seen in Figure 5.18. The dynamic range was measured to be 75 dB, with a peak SNR of 72 dB, and a peak SNDR of 67 dB.



(a) (b)

Figure 5.17: Measured Results with 0.085V Amplitude at 1957.03125 Hz

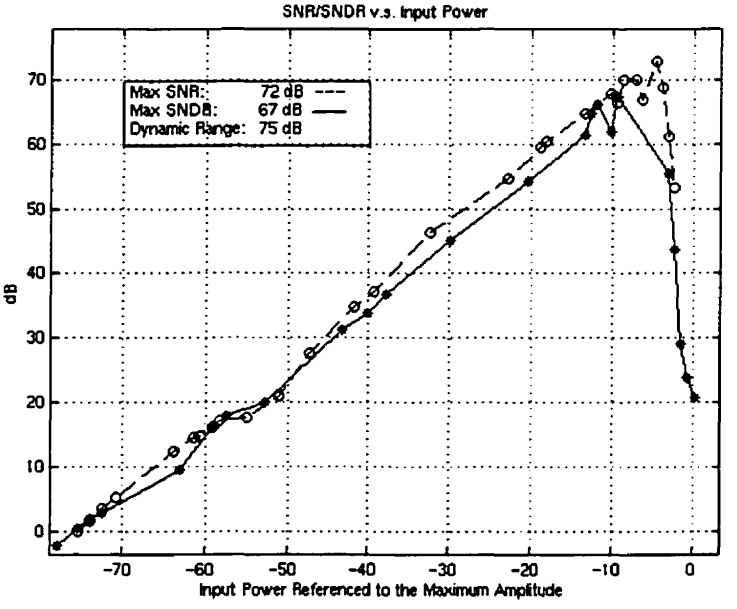


Figure 5.18: SNR/SNDR v.s. Input Power

Comparing these results with those found in simulation (Figure 5.9) reveal that the performance was lower than desired. This is due to the harmonics seen in Figure 5.17. Table 5.4 summarizes the experimental performance of the modulator presented in this chapter.

Dynamic Range	75 dB
SNR	72 dB
SNDR	67 dB
Oversampling Ratio	32
Sampling Rate	1.536 MHz
Signal Bandwidth	24 kHz
Input Range	1.0 V_{pp}
Supply Voltage	2.5 V
Power Consumption	44.1 mW
Technology	0.25 μm CMOS
Active Area	2.7 mm^2

Table 5.4: Converter Performance Summary

In order to determine the cause of the multiple harmonics, some tests aimed at eliminating possible sources were carried out. Firstly, the modulator was slowed down and run at a lower bandwidth. This was done to see if the distortion tones were dependent on settling time factors such as bandwidth or slew rate. The results of this test revealed no change in the amplitude of the harmonics. Secondly, the bias current for the modulator was tuned in order to see if the harmonics could be reduced. Tuning of the bias current did not lead to any reduction in the harmonic levels. Finally, the common mode level of the input signal was varied in order to determine if the distortion present was due to a mismatch in input signal and IC analog grounds. This procedure, like the previous two, seemed to have no beneficial effect on the distortion.

As a result, the most likely source of the distortion is the internal DAC, as its linearity directly limits the linearity of the modulator. Unfortunately, the modulator was not designed with the ability to override the internal DAC, so this could not be explicitly proven. However, given the elimination of the previously mentioned sources, the non-linearity of the DAC remains as the most likely candidate for the distortion.

Since the modulator was able to achieve 10.8 bits of accuracy, one can infer that the internal DAC was also able to achieve at least that amount of accuracy. Therefore, although 12 bits of accuracy was not achieved, the novel internal DAC was shown to work, and with

some modifications could present a new and viable means of implementing a multibit DAC.

5.6 Future Improvements

To improve the linearity of the multibit DAC presented in this chapter, an error compensation scheme is needed. This section will focus on a simple method of error correction that can be used to improve performance.

To implement an N-bit DAC with an accuracy of M bits (where M > N), in this chapter 2^N -1 latches were needed, and no ability to tune the output voltage was provided. This deficiency can be corrected, and will be briefly discussed.

In order to tune a voltage to an accuracy of M bits, it is necessary to be able to control that voltage in discrete steps of at most one LSB. As a result, one must be able to generate 2^M discrete voltage levels. For the type of DAC proposed here, this requires the use of (2^M -1) latches. However, since only an N-bit DAC is required, only 2^N of the 2^M levels are

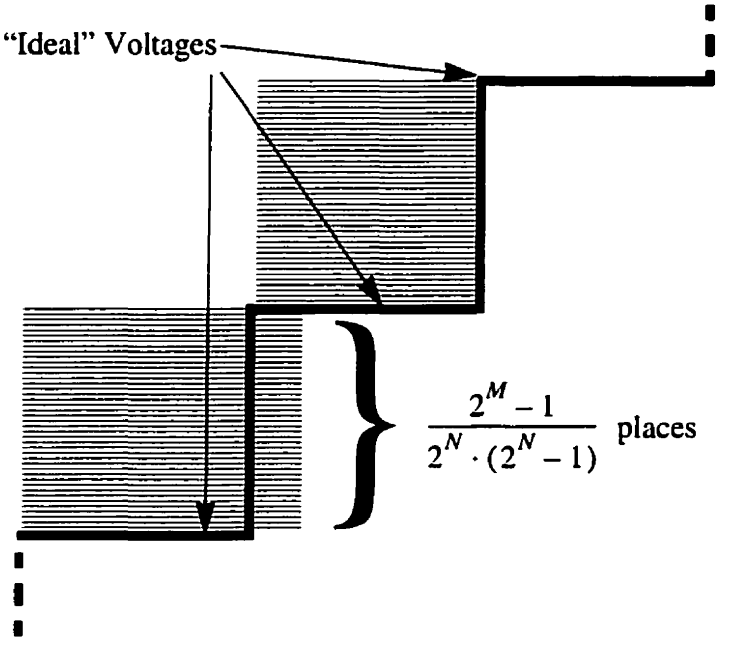


Figure 5.19: Ideal Voltage Levels for an N-bit DAC with 2^M Levels

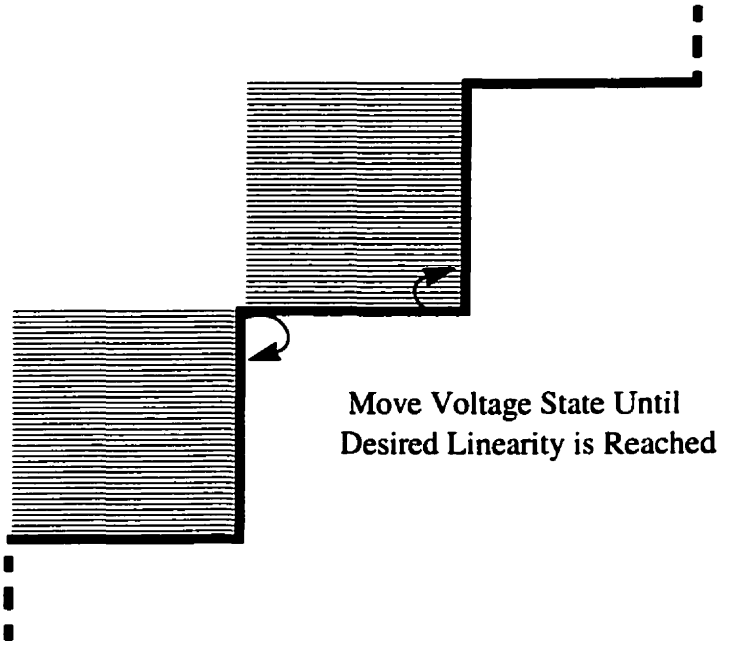


Figure 5.20: Shifting the Ideal Voltage Level to Improve Linearity

needed. As a result, each level needs to be separated by $\frac{2^M - 1}{2^N \cdot (2^N - 1)}$ places and is referred to as the "ideal" voltages. An illustration of this can be seen in Figure 5.19. This allows for the programming of the necessary levels, and also leaves $\frac{\left(\frac{2^M - 1}{2^N \cdot (2^N - 1)} - 1\right)}{2}$ empty states above and below the "ideal" voltage level.

Instead of hardcoding the bitstreams as was done on the DAC presented in this chapter, one can store an encoding of each of the 2^M states into ROM. Initially, the "ideal" voltages are sent to the DAC, and the modulator is run producing an FFT containing the undesired harmonics and a corresponding SNDR measurement. Next, the voltage level encoding sent to the bitstream generator is incremented so that it corresponds to one of the empty states above or below the "ideal" voltage (Figure 5.20). The modulator can then be run again, and the harmonics and SNDR examined. The procedure can then be repeated until the best linearity voltages for the DAC are found.

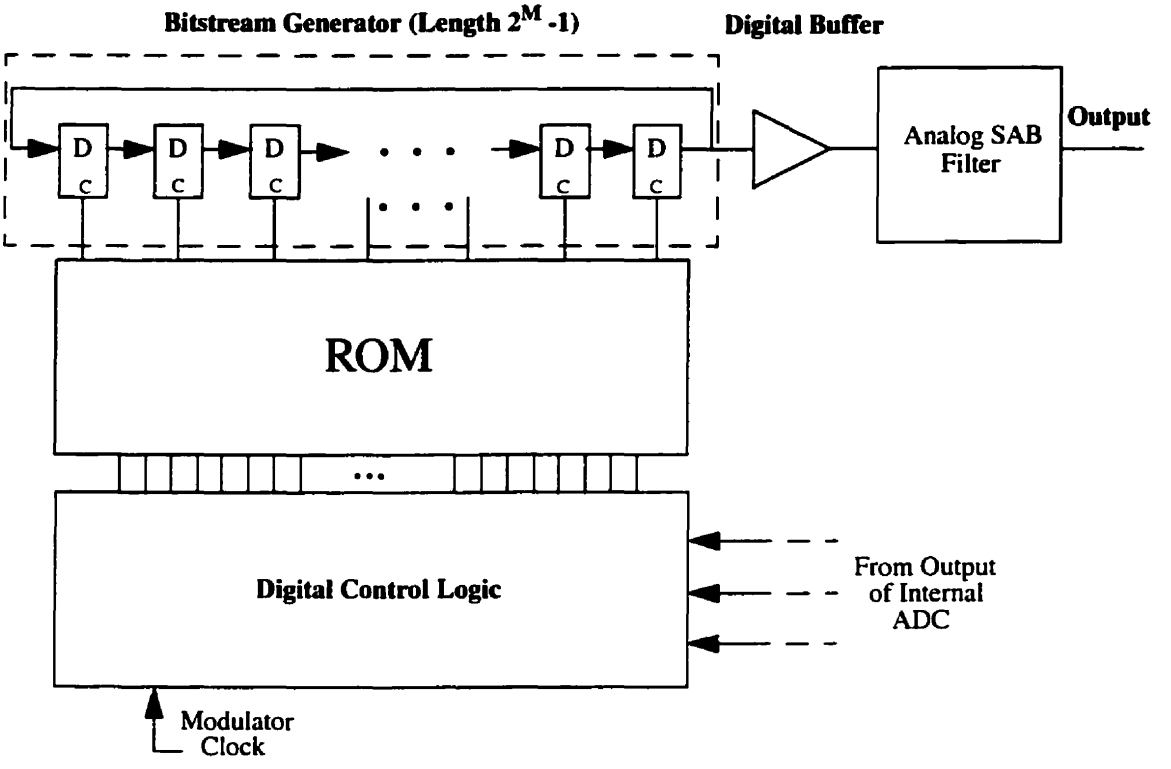


Figure 5.21: High Level Representation of the DAC Including Error Correction

A high level representation of the proposed system can be seen in Figure 5.21. It is important to note that not all the 2^M levels need to be encoded into ROM. Since the DAC is known to have 10 bits of linearity (section 5.5.2), only a fraction of the empty states corresponding to the difference in desired linearity (M) and the inherent linearity (10 bits) need be stored.

5.7 Conclusion

This chapter focused on the design and fabrication of a multibit ΔΣ modulator with a novel internal DAC. A multibit DAC based on encoding DC levels in PDM bitstreams and decoding them using a single analog filter was proposed. Simulation and experimental results of the DAC implemented in a 2nd order, lowpass, 3-bit, audio-band ΔΣ modulator validated its functionality. The experimental results however indicate that the linearity of the DAC must be improved upon to reach the simulated performance. As a result, an approach employing digital correction for the DAC was briefly discussed at the end of the chapter.

CHAPTER 6 Conclusion

This thesis began by introducing the noise shaping advantages gained by using ΔΣ modulation. This was followed by a discussion of the issues behind the design of ΔΣ modulator filtering functions, along with the problems of mapping them onto viable architectures.

Having introduced these issues, the transfer function design of a 3rd order, audio-band, single bit modulator with 16 bits of performance was carried out, and mapped onto a lossless discrete integrator topology.

Following this, the need for an easily reproduceable, high level, technology independent, top-down design methodology for the creation of analog or mixed-signal circuit components featuring Matlab, Simulink, and optimization was proposed. This method was then used to produce design values for the building blocks of a third order switched capacitor ΔΣ modulator.

Next, the issues behind the design of a low voltage, switched capacitor CMOS ΔΣ modulator were discussed. This was followed by a description of the design steps taken to arrive at the final layout, which featured the proposed top-down design methodology. The modulator was then fabricated in a 2.5 volt, 0.25 μm standard CMOS process and tested. The ex-

perimental performance of the modulator was then presented, revealing that it achieved 90 dB of performance, which in turn validated the top-down design approach.

Finally, a new multibit DAC based on encoding DC levels in PDM bitstreams and then extracting them through the use of a single path lowpass analog filter was proposed. The DAC was designed and inserted into a 2^{nd} order, 3-bit, audio-band $\Delta\Sigma$ modulator which was then fabricated. The experimental results validated the functionality of the novel DAC, but also indicated that there was room left for improvement.

In conclusion, this work has demonstrated the application of a simple to implement design methodology using Matlab, Simulink and an optimization routine to establish specifications on key design parameters, allowing for the design of high performance circuits. Furthermore, a novel multibit DAC was proposed and was shown to function properly.

References

[1]S.R. Norsworthy, R.Schreier and G.C. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, New York, 1997.

[2]Semiconductor Industry Association. *The National Technology Roadmap for Semiconductors: 1999 Edition.* Austin, Texas, International SEMATECH, 1999.

[3]*SIMULINK and MATLAB User's Guides*, MathWorks Inc., 1997.

[4]R.Harjani, R.A. Rutenbar, and L.R. Carley. "OASYS: A Framework for Analog Circuit Synthesis". *IEEE Transactions on Computer Aided Design*, vol.8, no. 12, pp.1247-1266, December 1989.

[5]R. Harjani. "OASYS: A Framework for Analog Circuit Synthesis". *Proceedings of the Second Annual IEEE ASIC Seminar and Exhibit*, pp.13-1.1 - 13-1.4, 1989.

[6]C. Toumazou and C.A. Makris. "Analog IC Design Automation: Part I- Automated Circuit Generation: New Concepts and Methods". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 2, pp.218-238, February 1995.

[7]B.A.A. Antao. "Trends in CAD of Analog ICs". *IEEE Circuits and Devices Magazine*, vol. 12, no. 5, pp.31-41, September 1996.

[8]F.El-Turky and E.E. Perry. "BLADES: An Artificial Intelligence Approach to Analog Circuit Design". *IEEE Transactions on Computer Aided Design*, vol.8, pp.680-692, June 1989.

[9]M.G.R. Degrauwe, B.L.A.G. Goffart, C. Meixenberger, M.L.A. Pierre, J.B. Litsios, J. Rijmenants, O.J.A.P. Nys, E. Dijkstra, B. Joss, M.K.C.M. Meyvaert, T.R. Schwarz and M.D. Pardoen. " Towards an Analog System Design Environment". *IEEE Journal of Solid-State Circuits*, vol.24, no. 3, pp.659-671, June 1989.

[10]M. Krasnicki, R. Phelps, R.A. Rutenbar and L.R. Carley. "MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells". *36th Proceedings of the Design Automation Conference*, pp.945-950, 1999.

[11]P.M. Lin and G.E. Alderson. *SNAP: A Computer Program for Generating Symbolic Network Functions*. Purdue University Technical Report TR-EE-70-16, August 1970.

[12]R.A. Rutenbar. "Analog Design Automation: Where are we? Where are we going?". *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp.13.1.1-13.1.8, 1993.

[13]S.J. Seda, M.G.R. Degrauwe and W. Fichtner. "A Symbolic Analysis Tool for Analog Circuit Design Automation". *IEEE International Conference on Computer-Aided Design*, pp.488-491, 1988.

[14]Y. Deval, J.B. Begueret, J. Tomas and P. Fouillat. "Toward Analog Circuit Synthesis: A Global Methodology Based Upon Design of Experiments". *Proceedings of the 13th Symposium on Integrated Circuits and Systems Design*, pp.295-300, 2000.

[15]A.L. Aita, S. Bampi and J.H. Choi. "A Generator of Trapezoidal Association of Transistors (TAT): Improving Analog Circuits in a Pre-Diffused Transistor Array". *Proceedings of the 13th Symposium on Integrated Circuits and Systems Design*, pp.99-104, 2000.

[16]P. Crolla. "An Advanced Array for Rapid Implementtion of Semi-Custom Integrated Circuit Designs". *Proceedings of the Custom Integrated Circuits Conference*, pp.475-478, 1985.

[17]P. King. "Subcircuits on Linear Arrays- A New Array Topology". *Proceedings of the Custom Integrated Circuits Conference*, pp.470-474, 1985.

[18]D.C. Stone, J.E. Schroeder, R.H. Kaplan and A.R. Smith. "Analog CMOS Building Blocks for Custom and Semicustom Applications". *IEEE Journal of Solid-State Circuits*, vol.19, no. 1, pp.55-61, February 1984.

[19]T. Teel and D. Wayne. "A Standard Cell Approach to Analog IC Design Utilizing Subthreshold Building Blocks". *Proceedings of the Custom Integrated Circuits Conference*, pp.484-490, 1985.

[20]T. Pletersek, J. Trontelj, L. Trontelj, I. Jones and G. Shenton. "High-Performance Designs with CMOS Analog Standard Cells". *IEEE Journal of Solid-State Circuits*, vol.21, no. 2, pp.215-222, April 1986.

[21]E. Hobekotte, B. Hoefflinger, H. Klien and M.A. Beunder. "State of the Art in the Analog CMOS Circuit Design". *Proceedings of the IEEE*, vol. 75, no. 6, pp. 816-828, June 1987.

[22]J. Kuhn. "Analog Module Generators for Silicon Compilation". *VLSI Systems Design*, pp.74-80, May 1987.

[23]S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi. "Optimization by Simulated Annealing". *Science*, vol. 220, no. 4598, May 1983.

[24]W. Nye, D.C. Riley, A. Sangiovanni-Vincentelli and A.L. Tits. "DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, pp.501-519, April 1988.

[25]E.S. Ochotta, L.R. Carley and R.A. Rutenbar. "Analog Circuit synthesis for Large, Realistic Cells: Designing a Pipelined A/D Converter with ASTRX/OBLX". *Proceedings of the Custom Integrated Circuits Conference*, pp.365-368, 1994.

[26]D. Enright, R.J. Mack, R.J. Binns and P. Hallam. "The Role of Designer Knowledge for Circuit-Level Optimization within an Analog Synthesis System". *42nd Midwest Symposium on Circuits and Systems*, vol.1, pp.60-63, 2000.

[27]F. Medeiro, B. Perez-Verdu, and A. Rodriguez-Vazquez. "A Vertically Integrated Tool for Automated Design of ΣΔ Modulators". *IEEE Journal of Solid-State Circuits*, vol.30, no.7, pp.762-772, July 1995.

[28]R.J. Bowman and D.J. Lane. "A Knowledge-Based System for Analog Integrated Circuit Design". *IEEE International Conference on Computer-Aided Design*, pp.210-212, 1985.

[29]M.G.R. Degrauwe, O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B.L.A.G. Goffart, E.A. Vittoz, S. Cserveny, C. Meixenberger, G. Van Der Stappen and H.J. Oguey. "IDAC: An interactive Design Tool for Analog CMOS Circuits". *IEEE Journal of Solid-State Circuits*, vol.22, pp.1106-1114, December 1987.

[30]A.H. Zaabab, Q.J. Zhang and M.S. Nakhla. "Device and Circuit-Level Modeling Using Neural Networks with Faster Training Based on Network Sparsity". *IEEE Transactions on Microwave Theory and Techniques*, vol.45, pp.1696-1704, October 1997.

[31]Y. Fang, M.C.E. Yagoub, F. Wang and Q.J. Zhang. "A New Macromodeling

Approach for Nonlinear Microwave Circuits Based on Recurrent Neural Networks". *IEEE Transactions on Microwave Theory and Techniques*, vol.48, no.12, pp.2335-2344, December 2000.

[32]G.W. Roberts. "*ΔΣ Modulation, Implementation and Applications*". McGill University, May 1997.

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

[34]G.W. Roberts, *IFAP: Symbolic Transfer Function Program*, McGill University.

[35]W.L. Lee, *A Novel Higher Order Iterpolative Modulator Topology for High Resolution Oversampling A/D Converters*, Master's Thesis, Massachusetts Institute of Technology, June 1987.

[36]X. Haurie, *Signal Generation Using High Order Delta Sigma Modulation*, Master's Thesis, McGill University, November 1996.

[37]G.W. Roberts, *Switched Capacitor Filter Theory and Design*, McGill University, 1997.

[38]Crystal Semiconductor Corporation. *CS4218 16-Bit Sterio Audio Codec Data Sheet.* Austin, Texas, 1996.

[39]M. Mahoney, *DSP Based Testing of Analog and Mixed Signal Circuits*, IEEE Computer Society Press, 1987.

[40]J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, Third Edition*, Prentice Hall, New Jersey, 1996.

[41]P.M. Aziz, H.V. Sorensen, and J. Van Der Spiegel. "An Overview of Sigma-Delta Converters: How a 1-bit ADC achieves more than 16-bit resolution". *IEEE Signal Processing Magazine*, pp.61-84, January 1996.

[42]T.E. Dwan, and T.E. Hechert. "Introducing SIMULINK into Systems Engineering Curriculum". *Proceedings of Frontiers in Education Conference*, 23rd Annual Conference, pp.627-671, 1993.

[43]A.Azemi, and E.E. Yaz. "Utilizing SIMULINK and MATLAB in a Graduate Non-linear Systems Analysis Course". *Proceedings of Frontiers in Education Conference*, 26th Annual Conference, vol.2, pp.595-598, 1996.

[44]A. Grace. *Optimization Toolbox for use with MATLAB*, MathWorks Inc., 1990.

[45]E.Baha. "Modeling of Resonant Switched-Mode Converters Using SIMULINK". *IEE*

Proceedings on Electronic Power Applications, vol. 145 3, pp.159-163, May 1998.

[46]A.S. Bozin. "Electrical Power Systems Modeling and Simulation Using SIMULINK". *IEE Colloquium on The Use of Systems Analysis and Modeling Tools: Experiences and Applications (Ref. No. 1998/413)*, pp.10/1-10/8, 1998.

[47]A. Dumitrescu, D.Fodor, T. Jokinen, M.Rosu, and S. Bucurenciu. "Modeling and Simulation of Electrical Drive Systems Using MATLAB/SIMULINK Environments". *International Conference IEMD '99*, pp.451-453, 1999.

[48]H. Hanselmann, U. Kiffmeier, L. Koster, M. Meyer, and A. Rukgauer. "Production Quality Code Generation from SIMULINK Block Diagrams". *Proceedings of the IEEE 1999 International Conference on Computer Aided Control System Design*, vol.1, pp.355-360, 1999.

[49]S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto, and F. Maloberti. "Modeling Sigma-Delta Modulator Non-Idealities in SIMULINK". *Proceedings of the IEEE Symposium on Circuits and Systems*, vol.2, pp. 384-387, ISCAS '99.

[50]B.E. Boser, and B.A. Wooley. "The Design of Sigma-Delta Modulation Anaolg-to-Digital Converters". *IEEE Journal of Solid State Circuits*, vol.23, no.6, pp.1298-1308, December 1988.

[51]S.Rabii, and B.A.Wooley. "A 1.8-V Digital-Audio Sigma-Delta Modulator in 0.8μm CMOS". *IEEE Journal of Solid-State Circuits*, vol.32, no.6, pp.783-796, June 1997.

[52]Y.Geerts, A.M. Marques, M.S.J. 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*, Vol.34, No.7, pp.927-936, July 1999.

[53]J.M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall, New Jersey, 1996.

[54]P.J. Crawley and G.W. Roberts. "Predicting Harmonic Distortion In Switched-Current Memory Circuits". *IEEE Transactions on Circuits and Systems*, Vol.41, No.2, pp. 73-86, February 1994.

[55]A.S. Sedra and K.C. Smith, *Microelectronics Circuits: Third Edition*, Oxford University Press, New York, 1991.

[56]F. Medeiro, B. Perez-Verdu, A. Rodriguez, and J.L. Huertas. "Modeling OpAmp Induced Harmonic Distortion for Switched-Capacitor ΣΔ Modulator Design". *Proceeding of the IEEE International Symposium on Circuits and Systems*, vol.5, pp.445-448, London, UK, June 1994.

[57]G.Temes and LaPatra, *Introduction to Circuit Synthesis and Design*, McGraw-Hill, New York, 1977.

[58]V. Peluso, M.S.J. Steyaert, and W. Sansen. "A 1.5-V-100-μW ΔΣ Modulator with 12-b Dynamic Range Using the Switched-Opamp Technique". *IEEE Journal of Solid-State Circuits*, vol.32, no.7, pp.943-952, July 1997.

[59]V. Peluso, P. Vancorenland, A.Marques, M.S.J. Steyaert, and W. Sansen. "A 900mV 40mW Switched Opamp ΔΣ Modulator with 77dB Dynamic Range". *Digest of Technical Papers 1998 IEEE International Solid-State Circuits Conference*, pp.68-69, 1998.

[60]A. Baschirotto, and R.Castello. "Low-Voltage Fully Differential Switched-Opamp Bandpass ΔΣ Modulator". *IEE Proceedings on Circuits, Devices and Systems*, vol. 146 5, pp. 249-253, October 1999.

[61]J. Crols, and M.S.J. Steyaert. "Swithched-opamp: An Approach to Realize Full CMOS Switched-Capacitor Circuits at Very Low Supply Voltages". *IEEE Journal of Solid-State Circuits*, vol.29, pp.936-942, August 1994.

[62]A.M. Marques, V. Peluso, M.S.J. Steyaert, and W. Sansen. "A 15-b Resolution 2-MHz Nyquist Rate ΔΣ ADC in a 1-μm CMOS Technology". *IEEE Journal of Solid-State Circuits*, vol. 33, no.7, July 1998.

[63]G.W. Roberts, *Course Notes: Analog Microelectronics*, McGill University, Fall 2000.

[64]K. Suyama, S.C. Fang, *User's Manual Fot SWITCAP2*, Version 1, Columbia University, 1990.

[65]D. Allstot, and W. Black. "Technological Design Considerations for Monolithic MOS Switched-Capacitor Filtering Systems". *Proceedings of the IEEE*, vol.71, pp.967-986, August 1983.

[66]J. McCreary. "Matching Properties, and Voltage and Temperature Dependence of MOS Capacitors". *IEEE Journal of Solid-State Circuits*, vol.16, pp.608-616, 1981.

[67]T. Choi, R. Kaneshiro, P. Gray, W. Jett, and M. Wilcox. "High Frequency CMOS Switched-Capacitor Filters for Communications Applications". *IEEE Journal of Solid-State Circuits*, vol. SC-18, pp.652-664, December 1983.

[68]H. Recoules, R.Bouchakour, P. Loumeau, F.T. Braz. "Two SC-CMFB Networks Used in Fully Differential OTA: Measurements and Improvements". *Proceedings of the 1998 Midwest Symposium on Circuits and Systems*, pp.352-355, 1999.

[69]A. Yukawa. "A CMOS 8-Bit High-Speed A/D Converter IC". *IEEE Journal of Solid-State Circuits*, vol. SC-20, no.3, pp.775-779, June 1985.

[70]S.H. Lewis, and P.R. Gray. "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter". *IEEE Journal of Solid-State Circuits*, vol. SC-22, no. 6, pp.954-962, December 1987.

[71]A.K. Ong, V.I. Prodanov, and M. Tarsia. "A Method for Reducing the Variation in "On" Resistance of a MOS Sampling Switch". *IEEE International Symposium on Circuits and Systems*, pp. V-437 to V-440, May 2000.

[72]M. Sarhang-Nejad, and G.C. Temes. "A High-Resolution multibit ΔΣ ADC with Digital Correction and Relaxed Amplifier Requirements". *IEEE Journal of Solid-State Circuits*, vol. 28, no. 6, pp.648-660, June 1993.

[73]J.C. Candy, and G.C. Temes. "Oversampling methods for A/D and D/A conversion". *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation, IEEE Press Collection of Papers*, Piscataway, New Jersey, pp.1-25, 1992.

[74]R.W. Adams. "Design and Implementation of an Audio 18-bit Analog-to-Digital Converter Using Oversampling Techniques". *Journal of Audio Engineering*, vol. 34, no. 3, pp.153-166, March 1986.

[75]T.C. Leslie, and B. Singh. "An Improved Sigma-Delta Architecture". *IEEE International Symposium on Circuits and Systems*, vol. 1, pp.372-375, 1990.

[76]L.R. Carley. "A Noise-Shaping Coder Topology for 15+ bit Converters". *IEEE Journal of Solid-State Circuits*, vol. SC-24, pp.267-273, April 1989.

[77]R.T. Baird, and T.S. Fiez. "Linearity Enhancement of Multibit ΔΣ A/D and D/A Converters Using Data Weighted Averaging". *IEEE Journal of Solid-State Circuits*, vol. 42, pp.753-762, December 1995.

[78]R. Schreier, and B. Zhang. "Noise-Shaped Multibit D/A Converter Employing Unit Elements". *Electronic Letters*, pp.1712-1713, September 28, 1995.

[79]A. Keady, and C. Lyden. "Tree Structure for Mismatch Noise-Shaping Multibit DAC". *Electronic Letters*, vol. 33, no.17, pp.1431-1432, August 14, 1997.

[80]I. Galton. "Spectral Shaping of Circuit Errors in Digital-to-Analog Converters". *IEEE Journal of Solid-State Circuits*, vol. 44, pp.808-817, October 1997.

[81]A. Yasuda, H. Tanimoto, and T. Iida. "A Third-Order ΔΣ Modulator Using Second-Order Noise-Shaping Dynamic Element Matching". *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp.1879-1886, December 1998.

[82]L. Louis, and G.W. Roberts. "A Single-path Multi-bit DAC for LP ΔΣ A/D Converters". *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, vol.1, pp.369-372, 1998.

[83]M.M. Hafed, S. Laberge, and G.W. Roberts. "A Robust Deep Submicron Programmable DC Voltage Generator". *The 2000 IEEE International Symposium on Circuits and Systems*, vol. 4, pp.5-8, 2000.

[84]B. Dufort, and G.W. Roberts. "Signal Generation Using Periodic Single and Multi-bit Sigma-Delta Modulated Streams". *Proceedings on the 1997 International Test Conference*, pp.396-405, 1997.

[85]S. Laberge, and R. Negulescu. "An Asynchronous FIFO with Fights: Case Study in Speed Optimization". *To be presented at the IEEE International Conference on Electronics, Circuits & Systems*, Kaslik, Lebanon, December 17-20, 2000.

[86]E.A. Guillemin. "Synthesis of RC Networks". *Journal of Mathematics and Physics*, vol.28, no.22, January 1949.

[87]A.D. Fialkow. "Two Terminal Pair Networks Containing Two kinds of Elements Only". *Proceedings of the Symposium on Modern Network Synthesis*, Polytechnic Institute of Brooklyn, Microwave Research Institute, New York, April 1952.