HIGH PERFORMANCE ADAPTIVE SIGMA DELTA MODULATOR DESIGNS

by

Yiğit Dağhan GÖKDEL

B.S., Electrical and Electronics Engineering, Bilkent University, 2004

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

Graduate Program in Electronics Engineering
Boğaziçi University

2007

HIGH PERFORMANCE ADAPTIVE SIGMA DELTA MODULATOR DESIGNS

APPROVED BY:

Prof. Günhan Dündar …….
(Thesis Supervisor)

Assoc. Prof. Arda Yurdakul ……

Asst. Prof. Arda D. Yalçınkaya ……

DATE OF APPROVAL: 29.01.2007

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

Firstly, I would like to thank my supervisor, Prof. Günhan Dündar for his positive encouraging attitude, guidance and suggestions. I could not have imagined having a better advisor and mentor for my thesis.

I would also like to give my special thanks to Selçuk Talay for teaching me about the subject and for answering every single question that I asked throughout the preparation of this thesis.

I wish to thank Baykal Sarıoğlu for his suggestions and helps in the programming part of this thesis. He helped me immensely by giving me encouragement and friendship.

My sincere thanks are due to Merve Kasarcı for supporting me throughout all my graduate education. Without her encouragement and understanding it would be have been impossible for me to finish this work.

I am also especially grateful to my beloved family, Şule, Erkan and Damla Gökdel whom I owe so much for their continuous support, patience and love. They always encouraged me to do my best. They are the true reason of my success.

Finally, I wish to express my sincere thanks to my grandmother, Melek Erkoçlar, who is passed away in the course of the preparation of this thesis, for caring and loving me all along her whole life.

ABSTRACT

HIGH PERFORMANCE ADAPTIVE SIGMA DELTA MODULATOR DESIGNS

In this work two different solutions to the saturation problem of the integrator building block of Sigma Delta modulators are presented. Saturation causes an error that limits the dynamic input range of the Sigma Delta A/D converters. Therefore, integrator must be prevented from entering saturation in order to get correct and meaningful data out of the modulator.

In the first proposed solution, when the output voltage of the integrator reaches the edge of saturation, designed circuit properly adapts the gain of the integrator to prevent it from clipping by simply adding an extra capacitor to the feedback path of the opamp and eventually decreases the summing step size of the integrator. Conversely, when the integrator output moves away from the saturation region, the added capacitor is removed from the circuit. This process is accomplished with help of a control circuit consisting of Schmitt Triggers and logic circuits.

Second method tries to solve the problem by using a non-uniform multi-bit quantizer architecture. One bit SD modulators are more resistant to the clipping effect of the saturation compared to multi-bit architectures. However it is also known that multi-bit SD architectures have much better SNR properties than one bit modulators. The proposed circuit tries to take advantage of both architectures by acting like a multi-bit modulator when output of the integrator is far away from saturation region and like a one bit modulator when the circuit enters into saturation.

ÖZET

YÜKSEK PERFORMANSLI UYARLAMALI SIGMA DELTA MODÜLATÖR TASARIMLARI

Bu çalışmada, Sigma Delta modülatörlerde bulunan tümlev alıcı devrenin doyum problemine değişik iki tane çözüm önerilmiştir. Doyum, Sigma Delta modülatörlerinin dinamik giriş alanını sınırlamaktadır. Bu nedenle, tümlev alıcı devrenin doyuma ulaşması modülatör çıkısının doğruluğunu ve geçerliliğini koruması açısından kesinlikle engellenmelidir.

Önerilen ilk sistem, tümlev alıcı devrenin çıkışı doyum sınırına yaklaştığında durumu önceden algılayıp devrenin kazancını yeni duruma geri besleme yoluna fazladan bir kondansatör ekleyerek uyarlamakta ve böylelikle tümlev çıkışında doyum yüzünden oluşan kırpılmaları engellemektedir. Yapılan bu işlem sonucunda tümlev üstünde yapılan toplama işleminin adım büyüklüğü düşürülmektedir. Tümlev, doyum bölgesinden uzaklaştığında ise eklenen kondansatör devreden çıkarılmaktadır. Önerilen bu yöntem, tasarlanan bir kontrol devresi sayesinde hayata geçirilmektedir.

Önerilen ikinci yöntem, aynı probleme değişen şekilli çok bitli bir kuantalayıcı kullanarak çözüm bulmaya çalışmaktadır. Tek bit kullanan Sigma Delta modülatörleri çok bit kullananlara kıyasla doyum problemine karşı daha çok dayanıklılık göstermektedir. Buna karşın çok bitli Sigma Delta modülatörlerin daha iyi işaret-gürültü oranı olduğu bilinmektedir. Önerilen devre, bahsedilen doyum bölgesinden uzakta çok bitli, doyum bölgesinde ise tek bitli bir modülatör gibi davranarak bu iki sistemin iyi özelliklerinden aynı anda faydalanmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVATIONS

C	Capacitance
dB	Decibel
e_{rms}	Rms voltage
$f_{baseband}$	Baseband frequency
$f_{Nyquist}$	Nyquist frequency
$f_{sampling}$	Sampling frequency
N	Number of bits
N_{TF}	Noise transfer function
P	Power
S_{TF}	Signal transfer function
Δ	Quantization step size
Φ_1, Φ_2	Clock signals
A/D	Analog to Digital
ADC	Analog to Digital Converter
BW	Bandwidth
DAC	Digital to Analog Converter
FSR	Full Scale Range
LSB	Least Significant Bit
OSR	Oversampling Ratio
PSD	Power Spectral Density
SC	Schmitt Trigger
SD	Sigma Delta
SNR	Signal to Noise Ratio
VLSI	Very Large Scale Integration

1. INTRODUCTION

It is a well known fact that the world that surrounds us and the signals in the real world are all analog. However, contrary to their analog essences, signals are generally preferred to be processed and used digitally thanks to the great advances recently made in digital technologies. This situation creates the absolute necessity of the conversion of signals from analog to digital domain and vice versa.

Throughout the last two decades, various analog to digital and digital to analog conversion designs and techniques are explored and used. Regarding their conversion speeds and used bandwidths, these data converters can be classified as “Oversampling Converters” or “Nyquist Rate Converters”. Nyquist rate converters typically require more precise analog circuitry which is very hard to realize due to the necessity to maximize the usage of the available bandwidth. Conversely, oversampling converters, besides their lower circuit complexity, circuit area and cost, use simpler anti-aliasing filters and quantizers as an exchange for lower operating speeds.

Sigma Delta A/D converters are examples of oversampling converters. Higher sampling frequencies along with noise shaping techniques are utilized as the characteristic principles in SD A/D conversion in order to achieve higher resolutions and SNR values. SD modulation and its principles were first introduced in 1962 [2]. Yet, these types of converters have not become widely known and attractive until the recent advances in VLSI technologies which resulted in the implementation of densely packed digital circuits and the augmentation of the usage of SD modulation. Today, digital audio processing technology can be shown as the primary application area of SD converters and the examples of the commercial products which use this technique have been available in markets for more than a decade now.

Several different problems have been solved and advances have been made concerning SD modulation owing to the thorough investigations done for years. Yet, there are still unsolved problems concerning these converters. One of these problems is the saturation of the integrator building block that limits the dynamic range of the SD

modulator and that eventually disrupts the output. Proposing possible solutions to this problem of the integrator building block of the modulator is the primary aim of this thesis.

Following this chapter, chapter two provides an overview of SD converters. It will briefly review the basic types of data converters which are followed by a discussion of SD converters in general. Thereafter, oversampling, noise shaping and quantization noise will be explained in detail. The chapter concludes with the discussion of decimation concept.

The problem that is tried to be solved and objectives are mentioned in chapter three. Thereafter, in chapter four, system level architectures are proposed. Additionally, circuit implementations are described along with the challenges in the design of the SD modulator and decimation filter. Finally, chapter five concludes the thesis.

2. OVERVIEW OF SIGMA DELTA CONVERTERS

2.1. Data Converters

Various analog to digital conversion methods are present in the literature. The sampling rate of the conversion is used as the essential categorization element that ensures the classification of analog to digital converters. The two main converter groups classified according to their sampling rates are “Nyquist Rate Converters” and “Oversampling Converters”.

Nyquist rate converters are able to attain very high conversion speeds whilst they need complex analog circuitry to perform. Flash and Successive Approximation A/D converters can be classified as Nyquist rate converters. Conversely, oversampling converters function at more moderate speeds and have more relaxed analog circuit complexity comparing to Nyquist rate converters. SD A/D converters can be given as an example of oversampling converters.

2.2. Sigma Delta Converters

SD converters exchange the loss of accuracy peculiar to analog circuits for faster signal processing and more digital circuitry. The loss in magnitude is compensated by resolution in time. Thus, they capitalize the speed of analog circuits and the accuracy of digital circuits [3].

As it is evident from Fig 2.1., SD A/D converters attain the highest resolution for a relatively low signal bandwidth [1]. If both the advantages and disadvantages of SD A/D converters are considered, it can be shown that the applications that require high resolutions but smaller bandwidths are the most suitable ones. Therefore, sigma-delta techniques are often used in speech applications where the signal bandwidth is only 4 kHz and where up to 14 bits of resolution may be needed [3]. Additionally, audio applications, where the signal bandwidth is 20-24 kHz and where high fidelity audio requires 16-18 bits

of resolution can be stated as another field in which SD A/D modulators are utilized frequently.

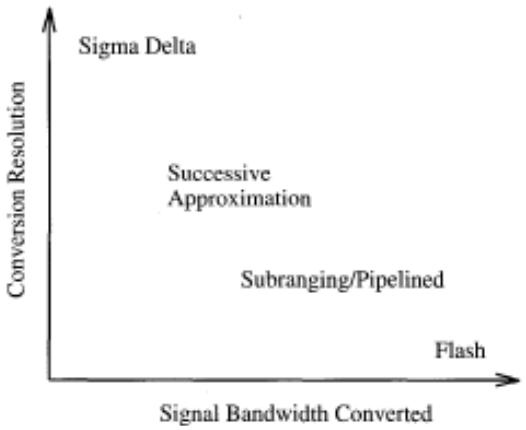

Figure 2.1 Bandwidth resolution tradeoffs [1]

2.2.1. General Structure of SD A/D Converters

SD A/D converters use the results of two different principles named “oversampling” and “noise shaping” in order to lower the quantization noise and improve the resolution. These two principles are explained in a more detailed fashion throughout the chapter two.

A generic first order SD A/D converter consists of an anti-aliasing filter at the input followed by the SD modulator and a digital decimator part at the output of the converter as depicted in Figure 2.2.

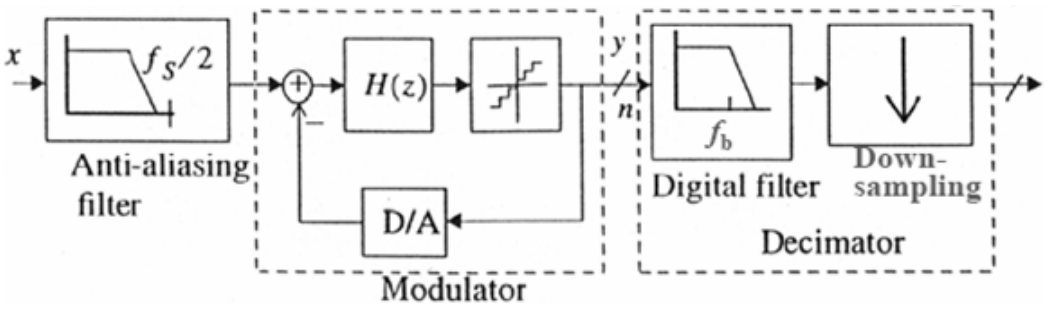

Figure 2.2 Block diagram of SD A/D Converter

At the input, the signal is passed through an anti-aliasing filter to ensure that the original signal can be reacquired after the reconstruction of the signal. After passing through the filter, the signal is fed to the SD modulator part which is depicted in Figure 2.3. Using this circuit topology, the noise created by quantization process is shaped into higher frequencies.

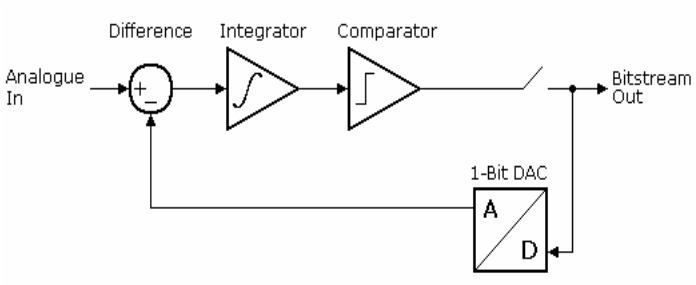

Figure 2.3 Block diagram of first order SD modulator

Following the modulator, the bit stream created at the output, is low pass filtered in a digital way to attenuate the noise previously shaped into higher frequencies and then down-sampled to Nyquist frequency. This digital side of the SD A/D converter which is what makes it inexpensive to produce, is more complex than the analog side and performs crucial filtering and decimation parts.

Briefly, the principle of the SD architecture is to make rough evaluations of the signal, to measure the error, integrate it and then compensate for that error. The mean output value is then equal to the mean input value if the integral of the error is finite [4]. The density of ones at the modulator output is proportional to the input signal. For an increasing input the comparator generates a greater number of ones and vice versa for a decreasing input. Thus, it can be said that the average value of the output tracks the input value.

To understand the functioning of SD A/D converters in depth, a first order SD modulator as depicted in Figure 2.3 was modeled using Simulink and its simulation outputs were processed in MATLAB [5]. The model for the first order SD modulator is given in Figure 2.4. The m-files of the quantizer and DAC building blocks of this model can be found at Appendix section A.1.

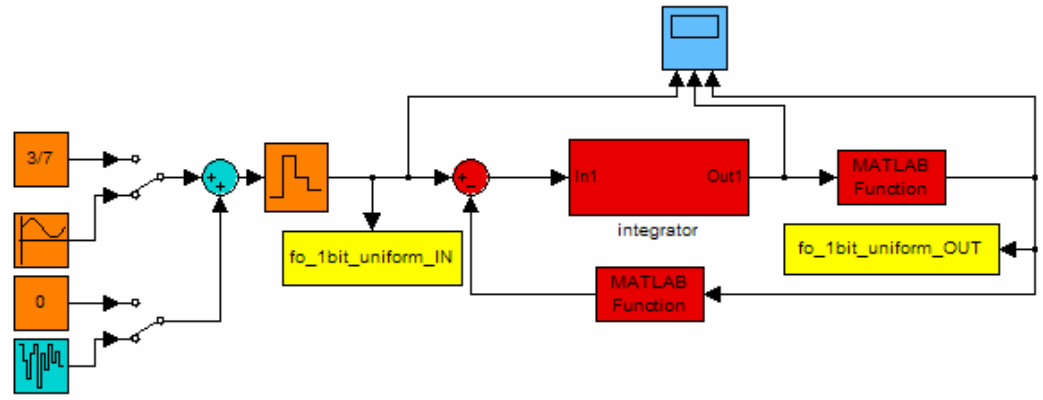

Figure 2.4 First order SD modulator model in time domain

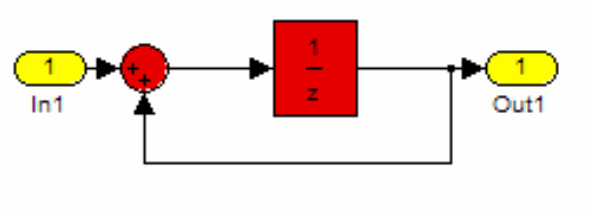

Figure 2.5 Integrator model used in Simulink model

A popular example of showing how SD modulators work is to use a constant 3/7V signal as the input to the modulator [4]. If the necessary simulation is carried out, three bits of every seven bit sequence take the value of 1V whilst the rest stays at 0V as shown in Figure 2.7. The first plot is the input of the modulator which is a constant of 3/7. The latter one is the modulators output.

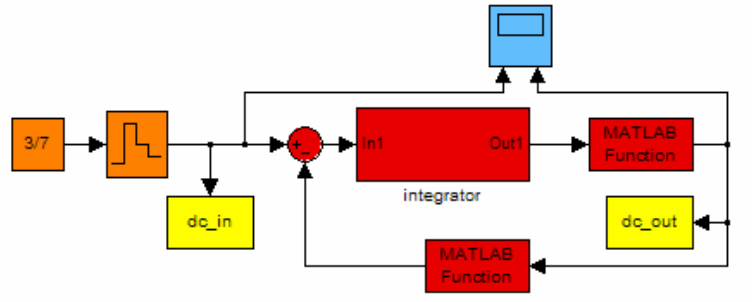

Figure 2.6 First order SD modulator for DC input

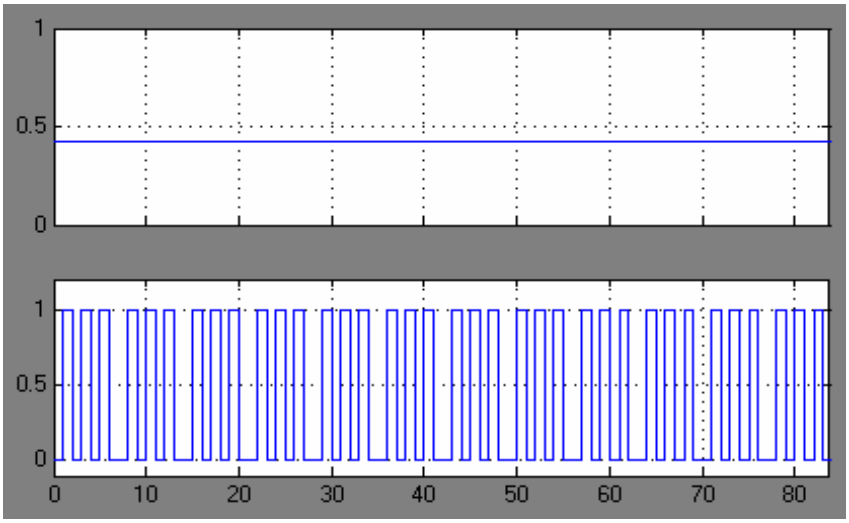

Figure 2.7 Output of SD modulator for 3/7V constant input

After doing small changes in the quantizer and DAC codes, a simulation is carried out with an input sinusoidal signal of 0.45V and 1.65V bias. The simulation creates the following waveform in Figure 2.8. The related codes can be found in Appendix A.1.2.

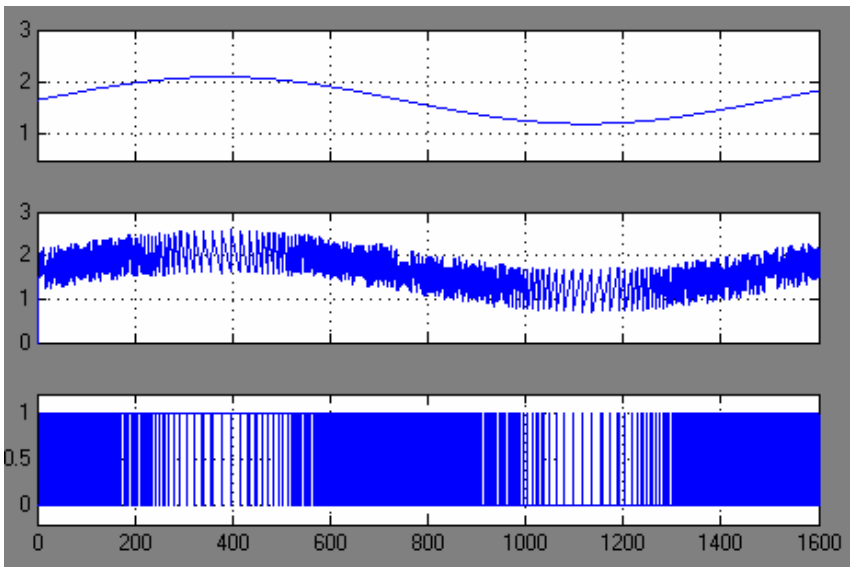

Figure 2.8 Output of SD modulator for a sinusoidal input

As previously described, Figure 2.8 shows that for one bit quantization, as the input voltage increases, more consecutive ones appear at output and vice versa more consecutive zeros are formed while the input decreases. Eventually, the average of the digital output tracks the input voltage.

2.2.2 Quantization Noise

In order to be able to investigate the principles of oversampling, one should have to understand clearly the concept of quantization noise. Briefly, a continuous amplitude signal is mapped to a finite number of discrete levels. Hence, quantization noise can be depicted as the difference between the digital and analog data, once the digital one is reconstructed in the analog domain [3]. The quantization error decreases with the number of the quantization levels, consequently the resolution of the A/D converter increases. Figure 2.9 represents a uniform quantization process. The interval between the successive levels is called the quantization step Δ and can be given as:

$$\Delta = \frac{FSR}{2^{N-1}} \tag{2.1}$$

In this formula, N represents the number of bits that are used in quantization and FSR represents the full scale range of the signal. As it can be seen from Figure 2.9, quantization rounds off a continuous amplitude signal to integer numbers in the range previously determined. The error is defined by the input and has equal probability of lying anywhere in the range of $\pm\Delta/2$. In other words, the quantization error is restricted to Δ located between plus and minus one half LSB. If it is assumed that the noise is uniformly distributed over Δ, then the probability density function becomes:

$$P(y) = \frac{1}{\Delta} \tag{2.2}$$

If the quantization error e has the equal probability of lying anywhere in the range $\pm\Delta/2$, its mean square value becomes

$$e_{rms}^2 = \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} e^2 de = \frac{\Delta^2}{12} \tag{2.3}$$

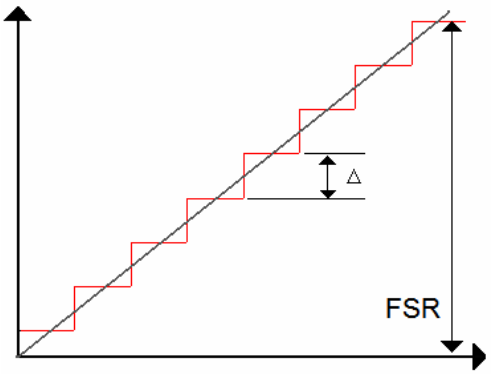

Figure 2.9 Uniform quantization

Taking the square root of the above expression gives the quantization noise voltage.

$$e_{rms} = \frac{\Delta}{\sqrt{12}} \tag{2.4}$$

To be able to compute the signal to noise ratio of an ideal N bit quantized sine wave, in addition to the quantization noise voltage that is just computed as equation 2.4, the sine wave rms magnitude must be also evaluated. The rms voltage of sine wave is equal to:

$$s_{rms} = \frac{2^N \Delta}{2\sqrt{2}} \tag{2.5}$$

Hence;

$$SNR = \frac{s_{rms}}{e_{rms}} = \frac{\frac{2^N \Delta}{2\sqrt{2}}}{\frac{\Delta}{\sqrt{12}}} = 2^N \sqrt{1.5} \tag{2.6}$$

If the logarithm of the above equation is taken, the dB value of SNR is found as below.

$$(SNR)_{dB} = 6.02N + 1.76 \tag{2.7}$$

It is important to keep in mind that equation 2.7 creates a link between the SNR and the number of bits used in quantization. However, all the formulas derived in this section, until this point is valid for Nyquist rate converters. As previously mentioned SD A/D converters use oversampling technique to lower the quantization noise and consequently improve the resolution. Oversampling principle changes the e_{rms} formula by taking the sampling frequency knowledge into account and eventually improves the SNR.

2.2.3. Oversampling

According to the Nyquist theorem, the higher sampling rate does not add information to the sampled signal but unnecessarily widens the spectrum. Since the quantization noise power remains unchanged regardless of the sampling rate, the noise power density must go down as the spectrum widens [3]. Previously, the error voltage depends only to the number of bits used in quantization but if the bandwidth of the oversampled signal is restricted to the baseband and if the sampling frequency effect is also considered, the equation 2.3 becomes:

$$e^2{}_{rms} = \frac{\Delta^2}{12\,OSR} \tag{2.8}$$

OSR abbreviation means oversampling ratio and it is equal to:

$$OSR = \frac{f_{sampling}}{2 f_{baseband}} = \frac{f_{sampling}}{f_{Nyquist}} \tag{2.9}$$

Eventually, oversampling provides smaller noise in the band of interest and smaller noise ensures a bigger SNR assuming the quantization noise spectrum is white. For instance, using equation 2.9, when an oversampling ratio of two is used, sampling frequency and the bandwidth of the quantization noise becomes four times larger, whereas the magnitude also is divided by four. Hence, if the bandwidth of oversampled signal is restricted through a filter to the baseband, the quantization noise power becomes four times smaller. In other words, four times decrease in quantization noise power means a 6 dB

improvement in SNR if equation 2.10, an altered version of equation 2.7 for oversampling converters, is considered.

$$(SNR)_{dB} = 6.02N + 1.76 + 3.01(OSR) \qquad (2.10)$$

As can be seen from the equation above, every doubling of OSR improves SNR by 6dB, which is the same as adding one bit to the quantized signal. This principle consequently creates an increase in the accuracy of the converter which is a very important characteristic of SD A/D converters. Finally, the improvement resulting from oversampling is shown if Figure 2.10.

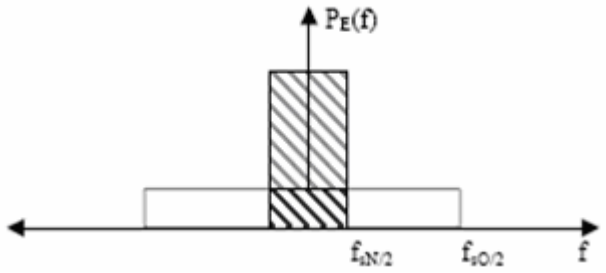

Figure 2.10 Nyquist rate vs. oversampled A/D converter

2.2.4. Noise Shaping

Noise shaping is a filtering operation that shifts a part of the noise to higher frequencies to leave less noise power in the baseband. This method supplements the benefit acquired using oversampling principle and enables to improve SNR without further compromising the bandwidth [5].

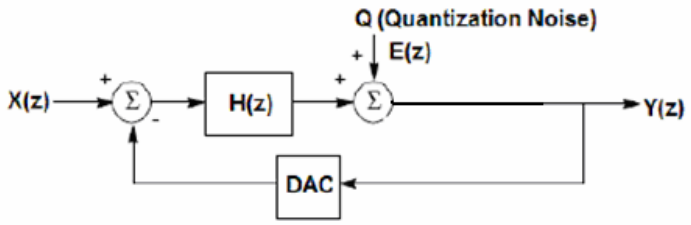

Figure 2.11 Linearized block diagram of a first order SD modulator

In the block diagram depicted above, it is assumed that the quantization noise is generated by an input e which is independent of the circuit input. The transfer function of the circuit becomes:

$$Y(z) = S_{TF}.X(z) + N_{TF}.E(z) \tag{2.11}$$

where;

$$S_{TF} = \frac{H(z)}{1+H(z)} \tag{2.12}$$

and

$$N_{TF} = \frac{1}{1+H(z)} \tag{2.13}$$

In the above equations, S_{TF} and N_{TF} represent the transfer function of the signal and the noise respectively. A loop filter $H(z)$ that acts like an all pass filter to the input signal and that acts like a high pass filter to the quantization noise is required in order to shape the quantization noise away from the desired frequency range. Ideally, the noise and the signal transfer functions intend to be 0 and 1 respectively. Hence $H(z)$ must approach to infinity for z=1. The desired effect can be created using an integrator instead of the loop filter $H(z)$ in the first order SD modulator. The transfer function of an integrator is:

$$H(z) = \frac{z^{-1}}{1-z^{-1}} \tag{2.14}$$

Hence, equations 2.12 and 2.13 become equations 2.15 and 2.16 respectively.

$$S_{TF} = z^{-1} \tag{2.15}$$

$$N_{TF} = 1-z^{-1} \tag{2.16}$$

Equation 2.10 also changes and becomes:

$$Y(z) = X(z)z^{-1} + E(z)(1 - z^{-1}) \tag{2.17}$$

It can also be represented as;

$$y[n] = x[n-1] + e[n] - e[n-1] \tag{2.18}$$

Eventually, as planned before, SD modulator behaves as a low pass filter to the input signal and as a high pass filter to the quantization noise. The circuit, at the output, makes the modulation error equal to the differences of quantization noise while leaving the input signal untouched, except for a delay as in equation 2.17. The combined effect of oversampling and noise shaping can be depicted as in Figure 2.12.

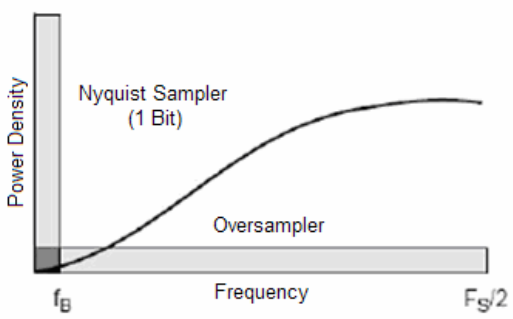

Figure 2.12 Spectrum of a first order SD modulator

Increasing the order of the modulator as depicted in Figure 2.13 also increases noise shaping. An increase in the order of the SD modulator by one ensures a 20dB/decade improvement in the shaping of the quantization noise.

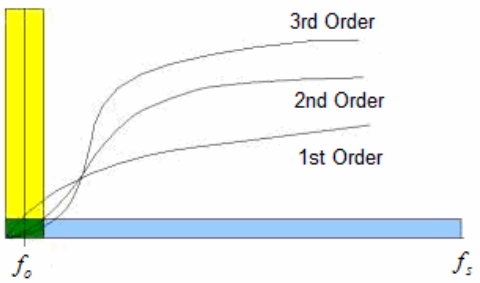

Figure 2.13 Noise shaping spectrum for different orders of modulations

2.2.5. Decimation

The digital side of the SD A/D converter which is what makes it inexpensive to produce, is more complex than the analog side. It performs low pass filtering and down sampling. Digital low pass filter attenuates the out of band noise created using noise shaping as in Figure 2.14.

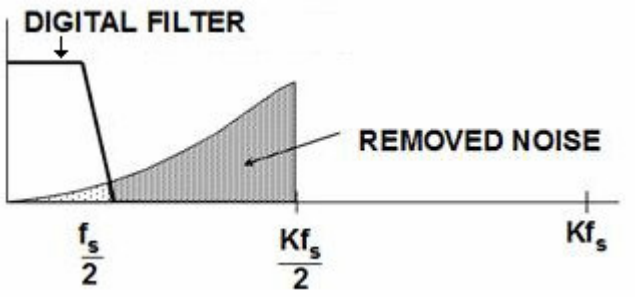

Figure 2.14 The function of the digital filter in decimation

After digital low pass filtering, sampling rate is reduced by downsampling so the Nyquist sampling theorem is satisfied.

3. PROBLEM STATEMENT

The integrator plays a crucial role in SD modulation. The incoming input signal accumulated on the integrator and transmitted to the comparator block. Actually, the integrator block is an opamp with a capacitor on its feedback path as it can be seen from Figure 3.1.

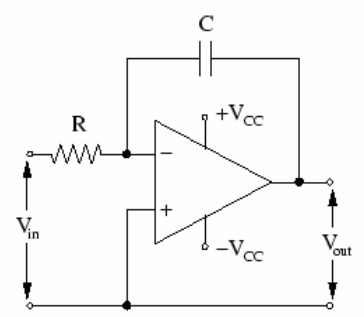

Figure 3.1 Simple integrator

The incoming signal is accumulated on the feedback capacitor in every clock cycle. Hspice simulations using an ideal behavioral opamp are carried out in order to depict the behavior of the integrator as shown in Figure 3.2.

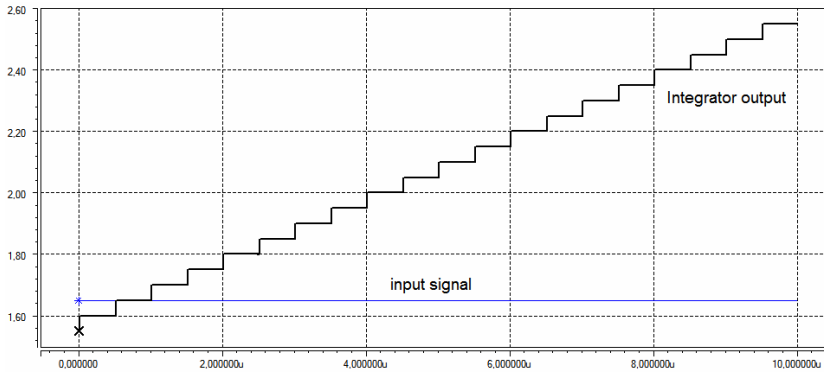

Figure 3.2 The behavior of the integrator implemented with an ideal opamp

In the above figure, the output of the integrator continues to increase consistently in every clock cycle with a step size equals to the input voltage and never goes into the saturation. Yet, if a non-ideal behavioral opamp is used in Hspice simulations, the following waveform in Figure 3.3 is acquired at the output of the integrator.

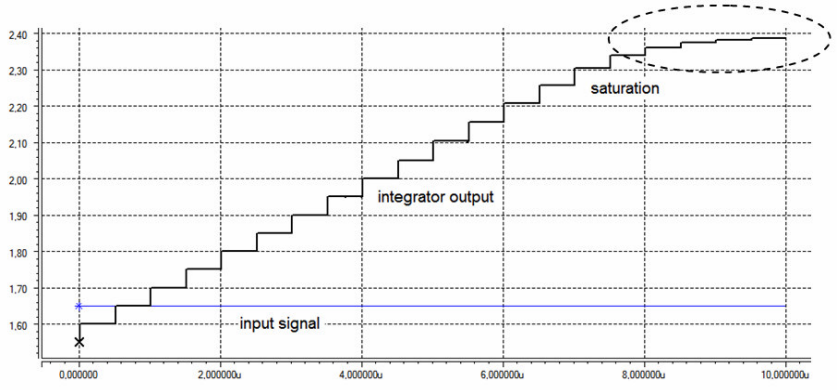

Figure 3.3 The Behavior of the integrator implemented with an non-ideal opamp

Since the output of the integrator begins to saturate approximately at 2.3V, contrary to the ideal case, the output does not increase consistently and eventually ceases to increase completely after a small interval of time with gradually reducing step sizes. Thus saturation of the integrator causes loss of information at the output along with the degradation of dynamic range and SNR. Besides, it also spoils the output waveforms as depicted below in Figure 3.4 and 3.5.

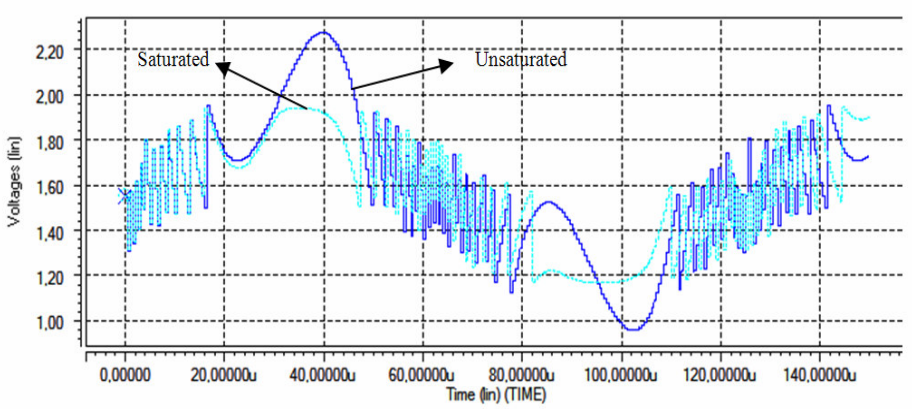

Fig 3.4 Integrator output waveforms with/without saturation

The related simulations of both Figure 3.4 and 3.5 are carried out with the same sinusoidal input voltages using an ideal behavioral opamp with and without saturation. Hspice simulation tool is employed for the simulations. It is clearly seen that saturation changes the waveform of the quantization outputs severely. Namely, it changes the digitized data and makes it completely different from the analog one depending on the degree of the saturation and deterioration.

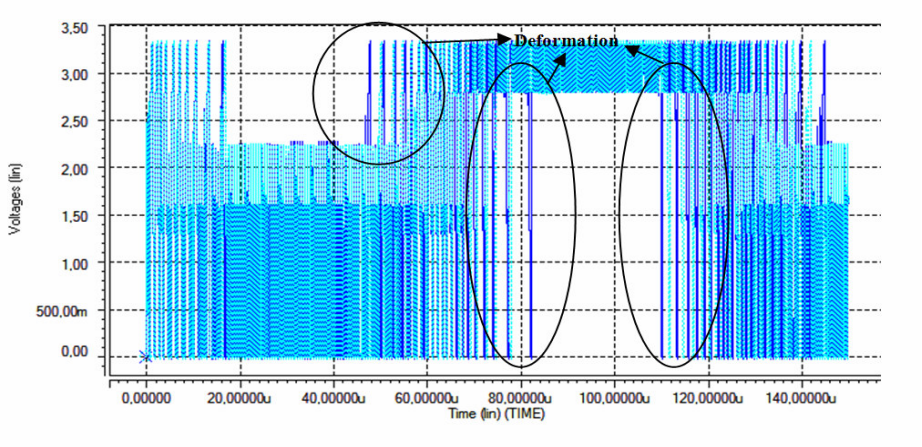

Fig 3.5 Deformation of quantizer output waveforms when there is saturation

Several different solutions have been proposed on this problem with saturation in the literature. For instance, with the help of additional hardware, a structure that uses input feed-forward signals and an additional extra quantizer can make the swing of integrator 85% smaller and can reduce the influence of the harmonic distortion coming from the opamp non-linearity [15].

One another method is to use mixed-mode integrators that consist of analog and digital integrators in order to increase the dynamic range [8]. With this method it is possible to achieve 12 dB better dynamic range than the conventional structures with the same oversampling ratio. This method makes use of an overload detector that keeps track of only the integrator output. If the output of an integrator exceeds a certain level, it is likely to exceed the allowed maximum voltage in the next time step. In order to prevent integrator from being saturated, the local feedback loop is activated and the digital counterpart of the feedback signal is applied to digital compensators.

Alternatively, a structure that is adaptively controlling the feedback coefficient of the integrator can extend the unclipping input level to full scale while that of the standard design is usually 3dB lower [6]. In this method, integrator gains together with the feedback coefficients serve as the scaling factors to avoid clipping. Variations in the gain of integrators change the signal ranges of the integrator outputs without affecting the functionality of the modulator.

Much other work also has been done to tighten the upper bounds on the output of the integrators of standard second-order SD modulator [7] [30] [31]. According to these bounds, designers can estimate the clipping level and therefore determine the scaling factors to avoid clipping. However, so far all the bounds that have been developed are based on constant inputs and specific initial conditions which are necessary but not sufficient stability criteria.

One another possible and more delicate solution to this problem is that when the integrator is at the edge of saturation, the situation can be sensed in advance and through the system that will be developed, integrator gain can be adjusted accordingly.

The gain may be made variable by making the components that determine the gain of the circuit, capacitor and resistor, adjustable. Since changing the value of the capacitor is more feasible than changing the value of the resistor, an extra capacitor on the feedback path may be utilized to make the gain changeable. Additionally, a control circuit that automatically senses the saturation and that change the value of the feedback capacitance without ever stopping the conversion process and without causing charge loss is needed. Afterwards the output acquired from the modulator can be processed and filtered accordingly in digital domain to get the correct bit stream.

Another proposed solution to saturation problem of SD modulators is that to use a non-uniform multi-bit SD A/D architecture which acts like multi-bit SD modulator when the output of the integrator is far away from the saturation region and like a one bit uniform SD modulator when the output is at edge of the saturation. By being so, the proposed architecture benefits from the resistance of one bit SD modulator against saturation along with the SNR characteristics of multi-bit SD architectures.

In this proposed work, different solutions to the saturation problem of SD converters are also thoroughly investigated in the hope of finding a better and simpler technique.

4. ADAPTIVE SYSTEM ARCHITECTURES

4.1 Adaptive Gain Controlled SD Modulator

In order to be able to compare the performance of the proposed system, an ordinary first order SD modulator is designed and tested using Simulink, Matlab and Hspice respectively.

4.1.1 Simulink Model of First Order SD Modulator

Firstly, a Simulink model of a first order SD modulator is created as in Figure 4.1 in order to be able to do exhaustive behavioral simulations. The designed model uses a one bit uniform quantizer and a one bit digital to analog converter. m-files of used blocks can be found in Appendix A.1.2.

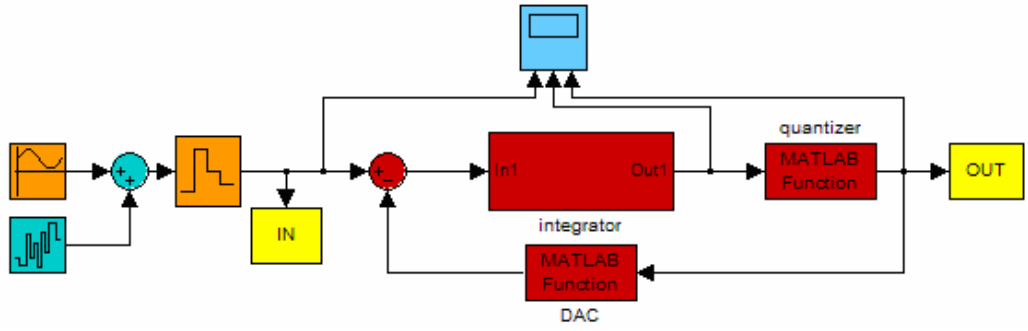

Figure 4.1 Simulink model of one bit first order SD modulator

The output waveforms of the designed modulator are shown together with the integrator outputs for given input signals in Figure 4.2. Simulations are done for a sinusoidal input signal of 0.45V on 1.65V bias and for a duration of 1600secs. However, this time, different from Figure 2.8, the modulator accepts an input with dither, a deliberately added erroneous signal, in order to make sure that the quantization noise is smaller and independent from the input signal [5].

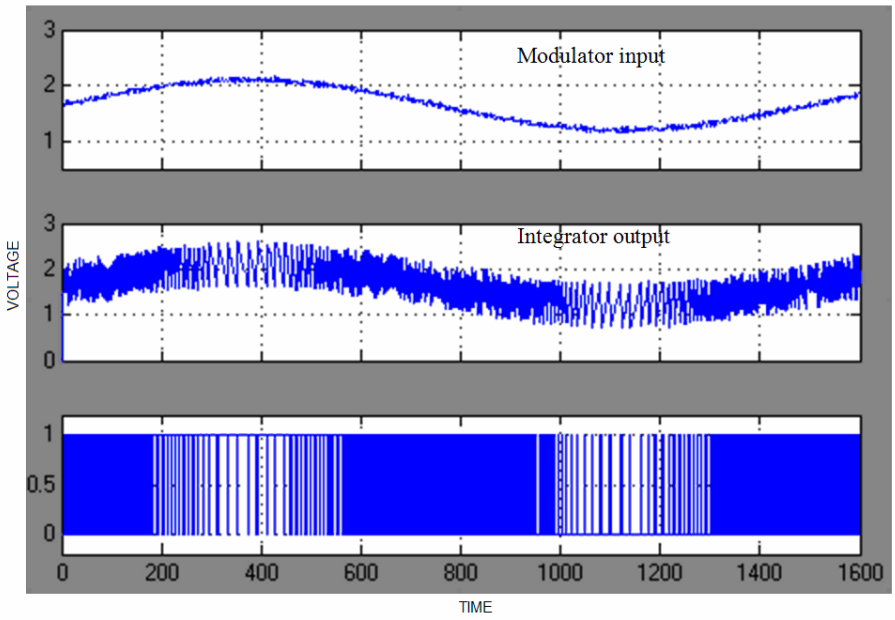

Figure 4.2 Output waveforms for an input with dither

The usage of dither is absolutely necessary in order to get valid PSD plots. As it appears in Figure 4.3, simulations that are carried out without using dither at the input cause additional noise in PSD plot and eventually give wrong results.

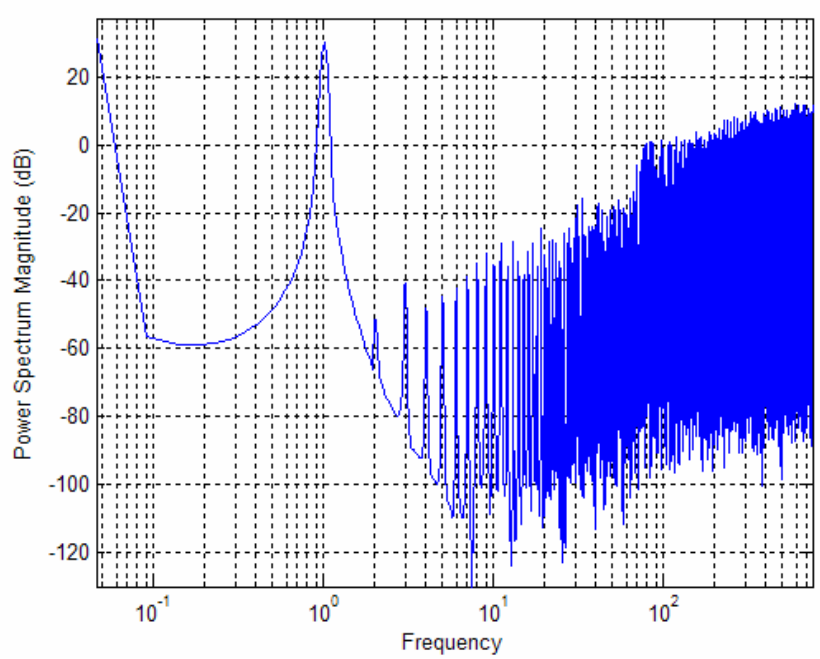

Figure 4.3 PSD for an input without dither

To check the validity of designed model, both the input and the output of the modulator are separately 8:1 decimated digitally using the code given in Appendix A.2.1 and compared. The following waveform is acquired.

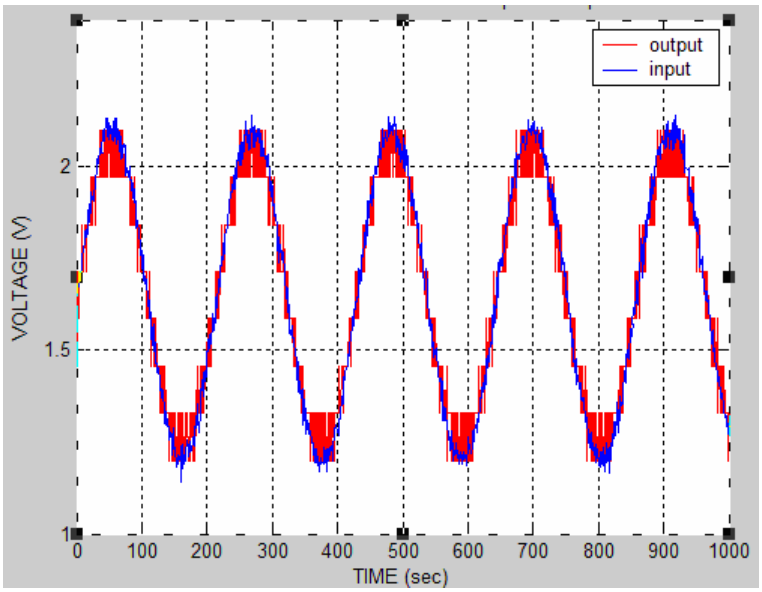

Figure 4.4 Waveforms of 8:1 decimated input and output signals

It is obvious that decimated input and output perfectly matches on top of each other in Figure 4.4 and that the created Simulink model functions correctly. Eventually, PSD plot of the model appears as follows.

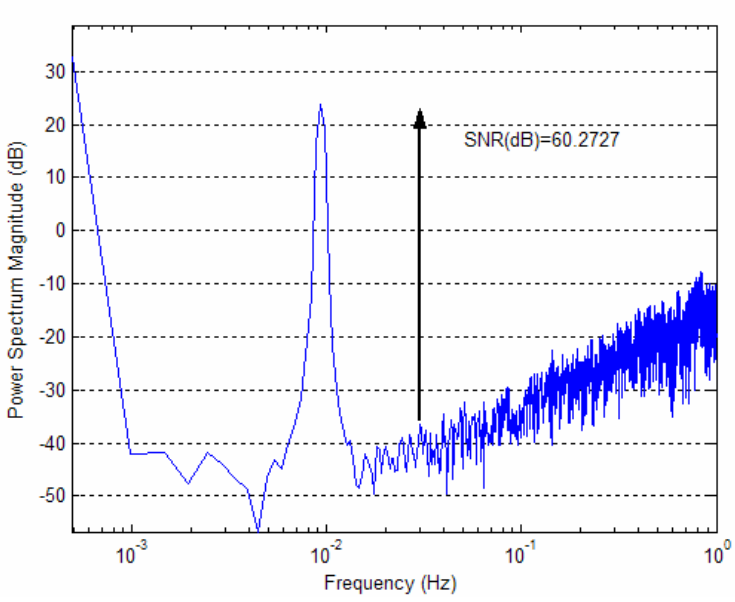

Figure 4.5 PSD plot of one bit first order SD modulator

Approximately, a 60dB of SNR is attained using the model. However, it is essential to keep in mind that all the Simulink simulations made until now are saturation free. To see the effect of saturation more clearly and to observe how it corrupts the conversion, a second Simulink model is designed. This model uses two bit quantizer and DAC. The related m-files of these blocks are in Appendix A.1.3.

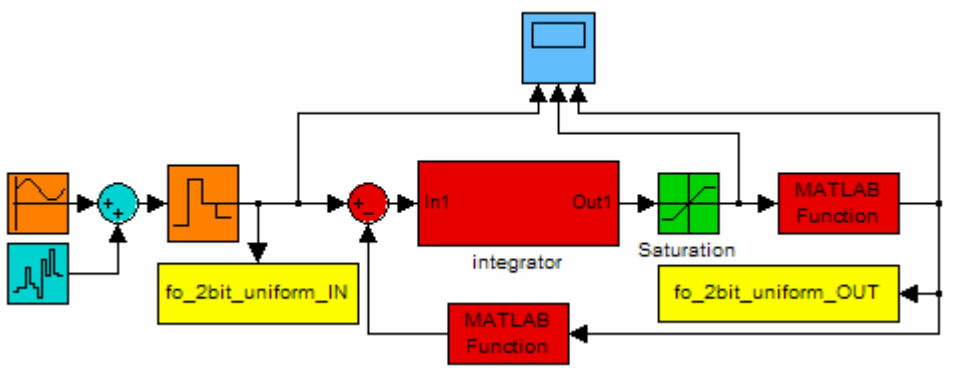

Figure 4.6 Simulink model of two bit first order SD modulator

Designed modulator is simulated with non-saturating input signal and has the following waveforms in Figure 4.7. This time, unlike one bit modulator, four different output values are acquired.

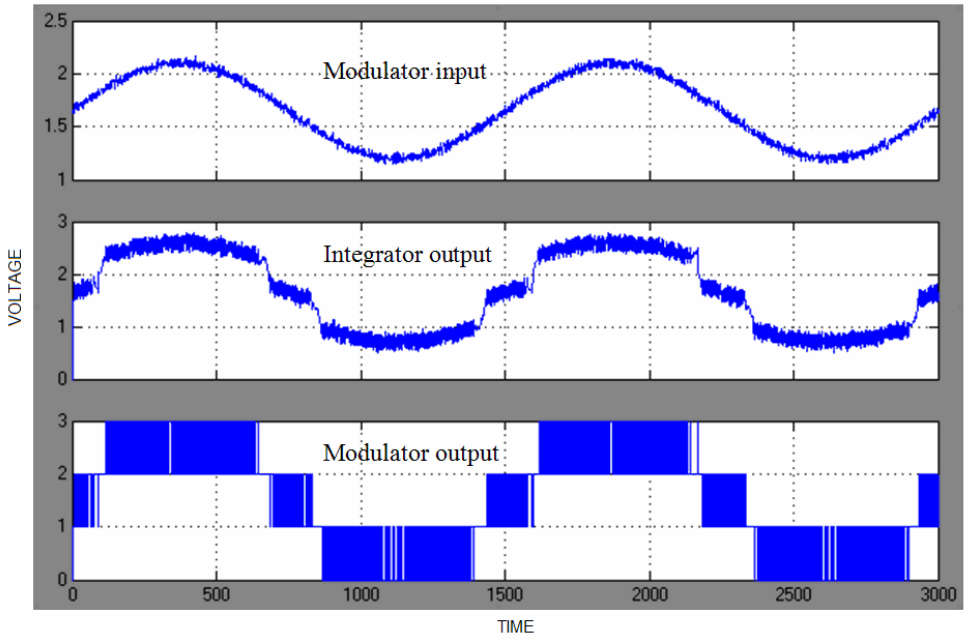

Figure 4.7 Output waveform of two bit first order SD modulator

In the model, integrator is designed to be saturated at 2.3V and 0.8V. Therefore, if the previous simulation is repeated for the same configuration but this time with saturating input signal. The output waveforms in Figure 4.8 are obtained.

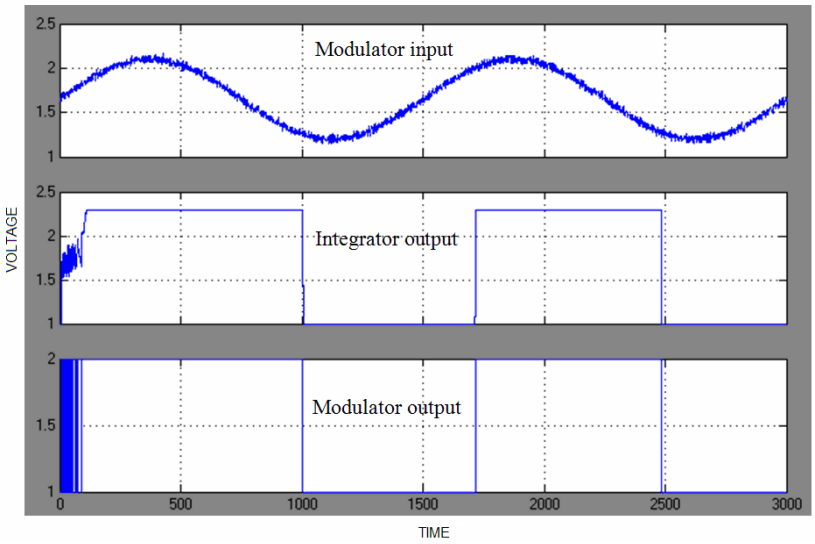

Figure 4.8 Output waveform when there is saturation

Consequently, saturation vastly distorts the output waveforms and this can be observed with ease by comparing Figure 4.7 and 4.8. Additionally, designed two bit model has the following PSD plots for both saturating and non-saturating inputs.

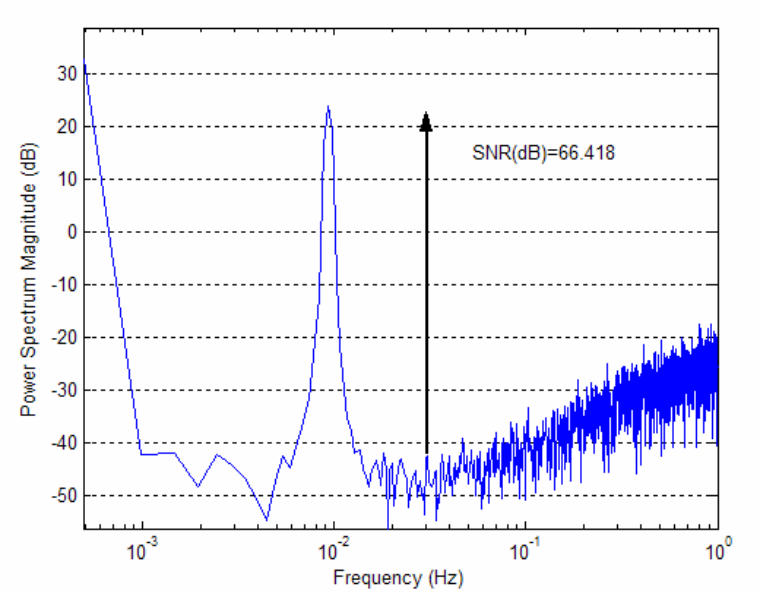

Figure 4.9 PSD plot of two bit first order SD modulator without saturation

The SNR of the two bit first order SD modulator is approximately 66.4dB. Initially, we have found that the SNR of one bit modulator was 60dB. It means that there is a 6dB SNR difference between one bit and two bit SD modulators. This result satisfies the equation 2.7 that is previously derived in quantization noise section. The combined PSD plot of both one bit and two bit modulators is depicted in Figure 4.10.

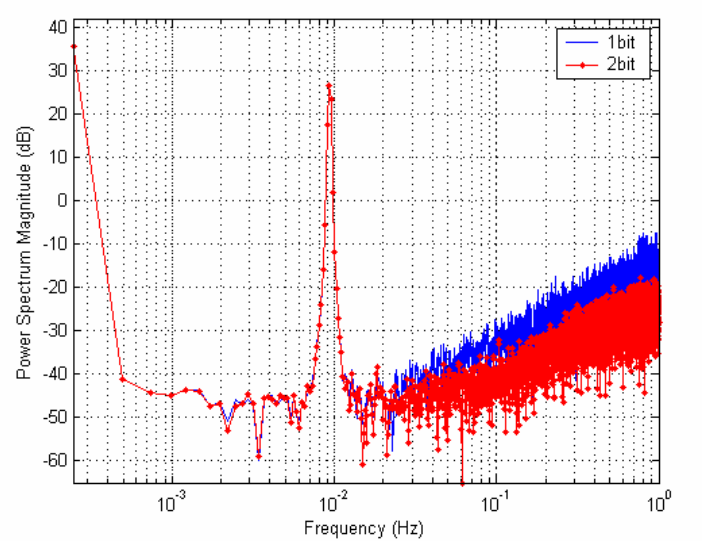

Figure 4.10 PSD plots of one bit vs. two bit modulators

When an integrator block that becomes saturated at 2.3V and 0.8V is used in the model, PSD plot differentiates as in Figure 4.11. Saturation causes 50dB drop in SNR.

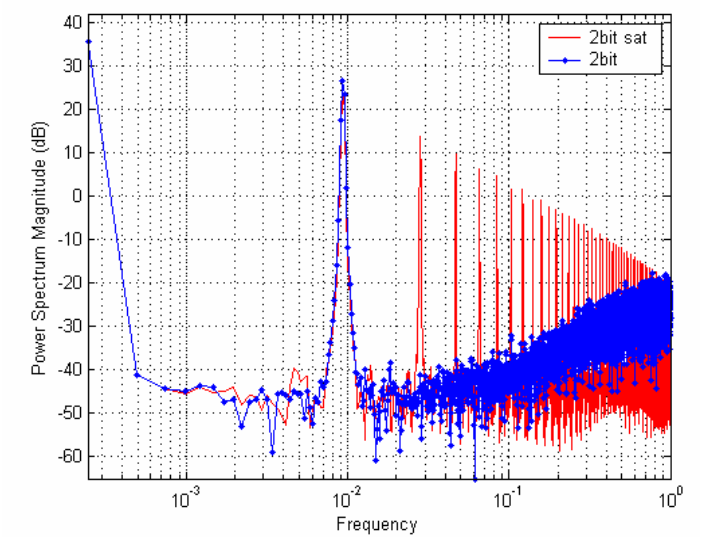

Figure 4.11 PSD plots of saturated vs unsaturated two bit modulators

4.1.2. Hspice Simulations of First Order SD Modulator

The schematic diagram of the designed one bit first order SD modulator is given in Figure 4.12.

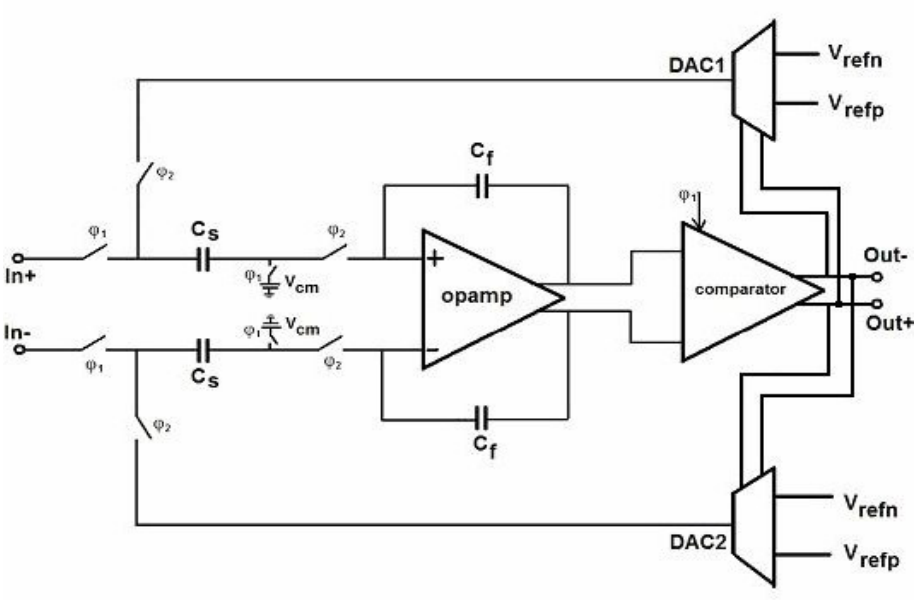

Figure 4.12 The schematic diagram of the first order SD modulator

A fully differential circuit topology is preferred in the design of this SD modulator. Integrating block is implemented as a switched capacitor integrator which can be seen in Figure 4.13.

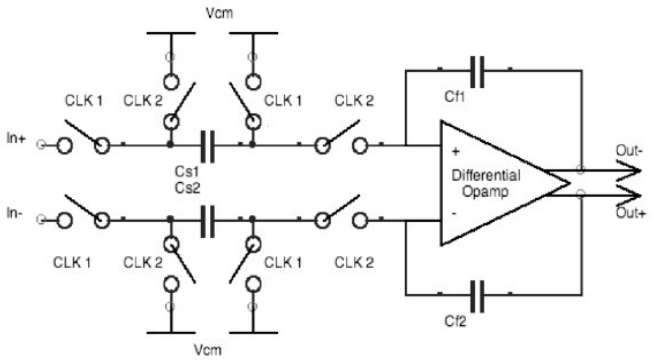

Figure 4.13 A switched capacitor integrator

The fully differential folded cascode opamp used in integrator has the following circuit topology [9].

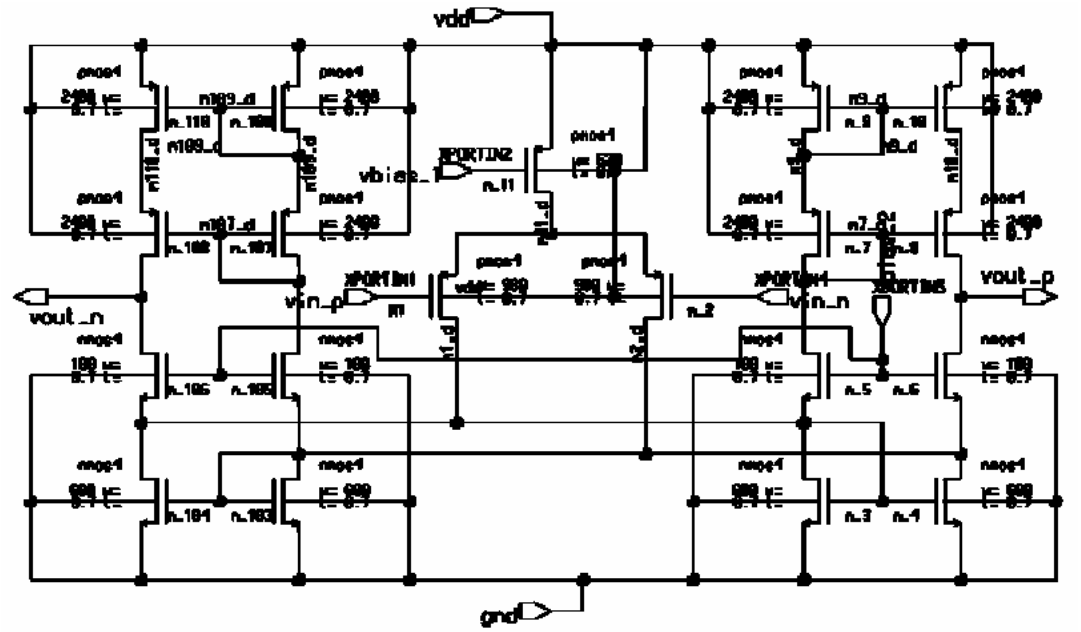

Figure 4.14 Fully differential folded cascode opamp

The comparator is designed as a one bit quantizer and has the following circuit topology [11].

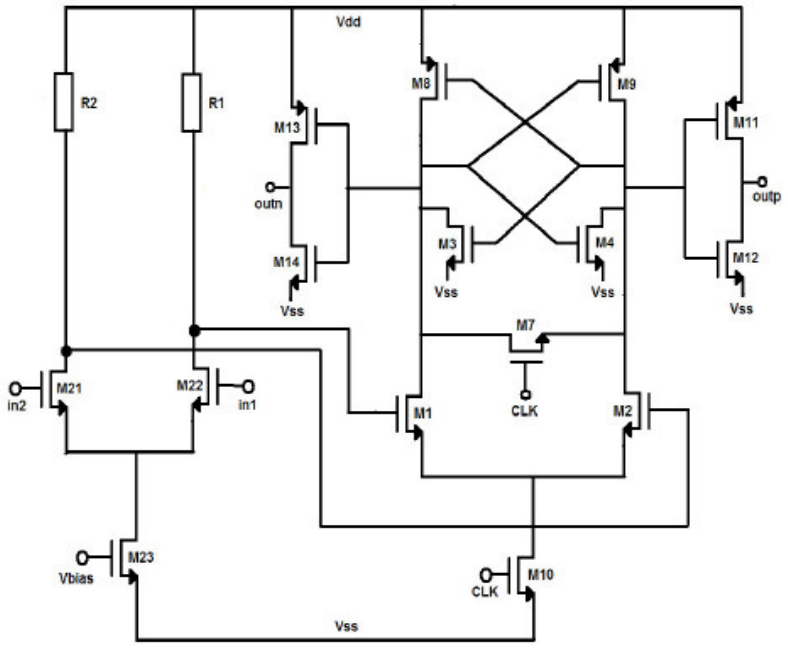

Figure 4.15 The schematic diagram of the comparator

DAC1 and DAC2 are the outputs of digital to analog converters that are used in the feedback path. They are implemented by using multiplexers which take two reference voltages as inputs.

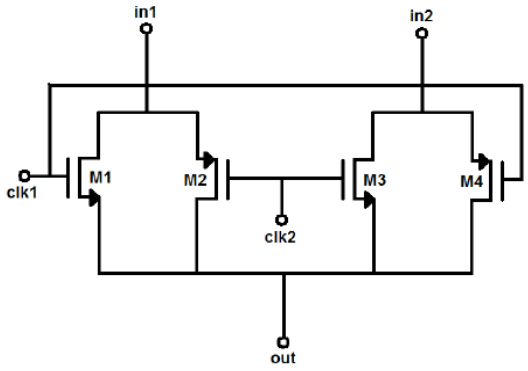

Figure 4.16 The Schematic Diagram of the Multiplexer

φ_1 and φ_2 are non-overlapping clock signals and used in the circuit as follows;

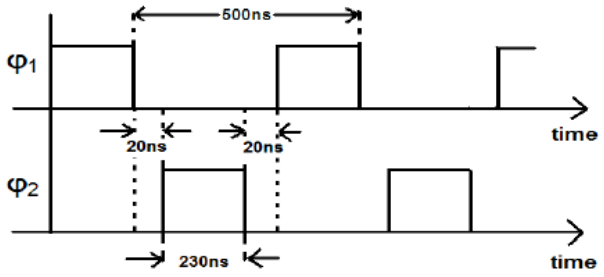

Figure 4.17 Timing Diagram of the Clock Signals

Finally, it can be specified that the ratio C_f / C_s is equal to two and V_{CM} is the common mode signal of value 1.55V. The Hspice code of the designed circuit together with its subcircuits can be found in Appendix B.1.

If the necessary simulations are carried out for both saturated and unsaturated ideal opamp, the output waveforms are acquired as previously given in Figure 3.4 and 3.5. To get the PSD plot of the circuit, the output data of a sufficiently long simulation is processed with a C code and prepared to Matlab for acquisition (Appendix C.1). Later, the prepared

data are digitized by using a specific code (Appendix A.2.2) in Matlab to get rid off the metastates formed during the spice simulation. Eventually, PSD is plotted as in Figure 4.18.

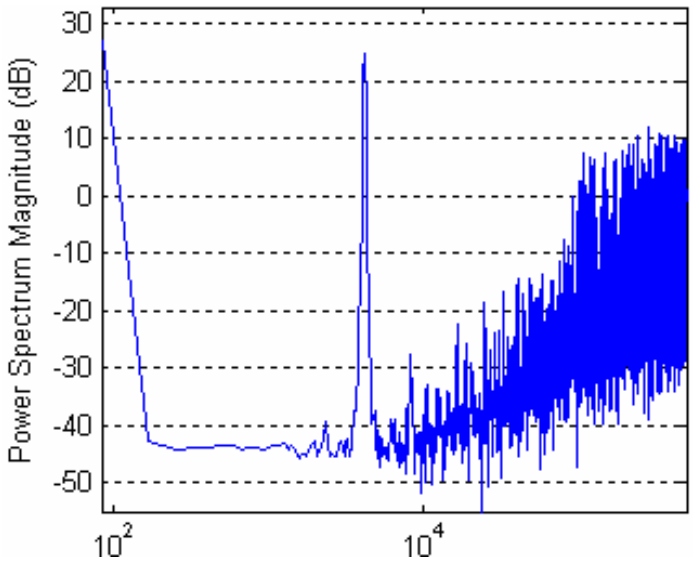

Figure 4.18 PSD of an ordinary first order SD modulator without saturation

SNR of generic first order SD modulator is computed using data acquired after Hspice simulations and has a value of approximately 60 dB. This is an expected result if we recall that the SNR of one bit first order SD modulator gathered from Simulink simulations is also 60 dB.

4.1.3. Variable Gain SD Modulator

The circuit depicted in Figure 4. 19 is proposed in order to solve the problem with saturation. The related spice codes are in Appendix B.2. Firstly, it is known that the transfer function of a switched capacitor integrator that appears in Figure 4. 13 is as follows:

$$\frac{Out(z)}{In(z)} = \frac{C_s}{C_f} \times \frac{z^{-1}}{1 - z^{-1}} \tag{4.1}$$

As can be seen, the gain of the integrator depends on the ratio of the sampling and the feedback capacitors. Thus, when the integrator comes to the edge of saturation, we have to decrease its gain in order to prevent it from entering saturation.

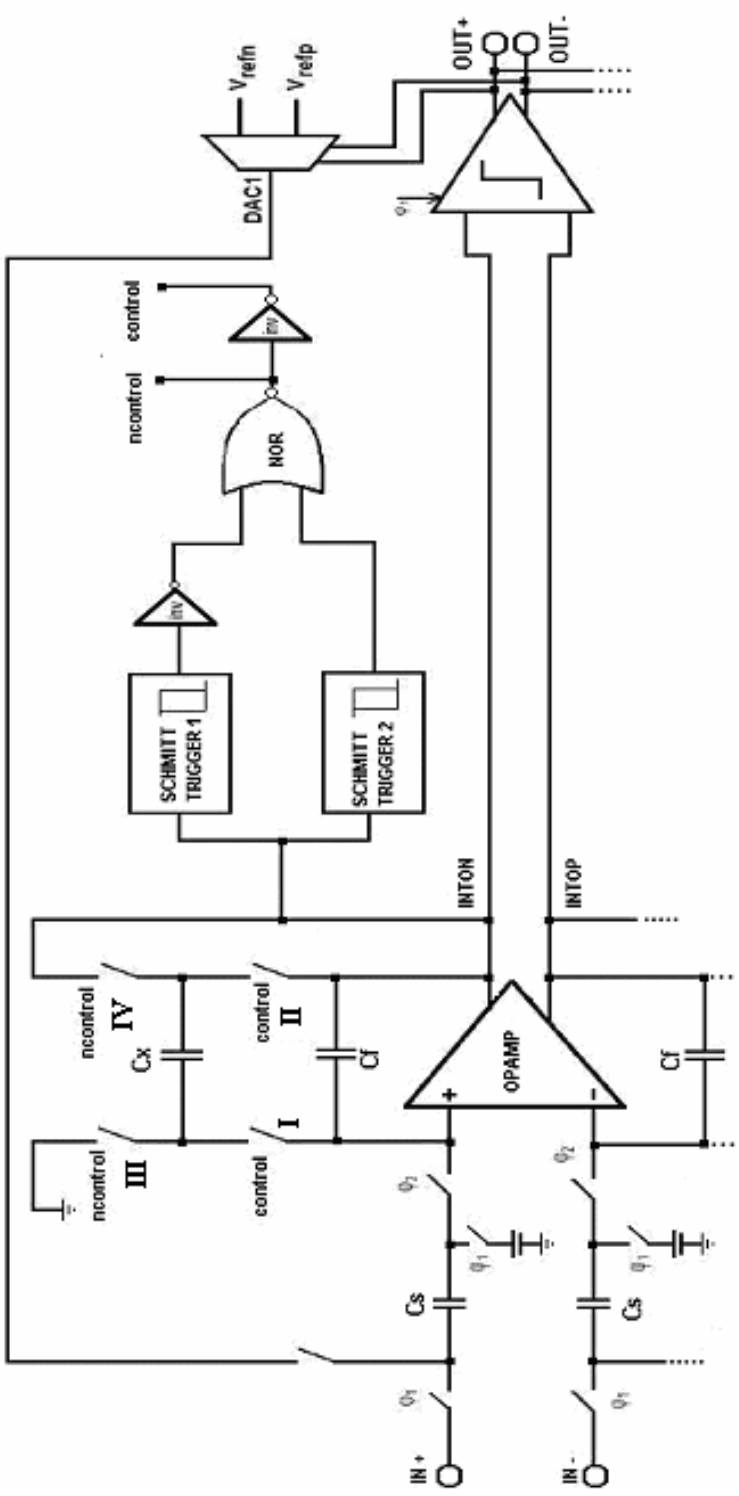

Figure 4.19 Adaptive first order SD modulator

According to the equation 4.1, the gain of the integrator can be decreased by increasing the value of the feedback capacitor. Then this value is augmented by adding a parallel capacitor onto it when the integrator comes to the edge of saturation. The edge of the saturation can be described as the voltage value at the output after which the integrator could not integrate correctly. In other words, it is the voltage level at the output of the integrator block after which the integrator could not add up the charges correctly on the feedback capacitor.

However, when the integrator moves away from the saturation region, the gain of the integrator must be increased again. Thus, the added capacitor must come out of the circuit. As a result, when the integrator comes to the edge of saturation an extra capacitor will be added to the feedback capacitor and when it is far away from saturation region, the added capacitor will be removed from the circuit.

In the proposed architecture, a control circuit, appearing in Figure 4.20, which consists of schmitt triggers and logic circuits, uses the integrator outputs as inputs in order to automatically sense the situation in advance, before the integrator goes into saturation. When the integrator is far away from the saturation region, the switches one and two are open to disconnect the extra capacitor C_X from the capacitor C_f . So, the proposed circuit functions as a generic first order SD modulator. While C_X is disconnected from C_f , it becomes connected to the output of the integrator and ground in order to follow the output voltage of the integrator.

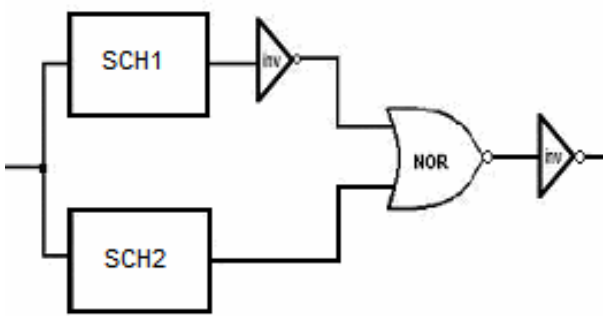

Figure 4.20 Control circuit

As soon as the control circuit senses the coming saturation, it sends the appropriate signal to do the necessary switching. Switches one and two becomes closed, three and four becomes open. So again, in other words, control circuit connects the extra capacitor onto the feedback capacitor and decreases the gain of the integrator.

In order to be able to design the control circuit, the threshold values of the Schmitt Triggers must be defined. Therefore, 100mV DC input is applied to the integrator. Circuit goes into the saturation for outputs of 900mV and 2.3V as it can be depicted in Figure 4.21 after Hspice simulations.

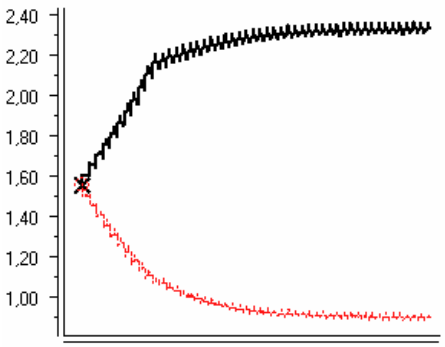

Figure 4.21 The saturation levels of the designed integrator

Thus, to prevent the integrator reaching these specific voltages and consequently to get rid off the corruption of the output bit stream, control circuits should send the proper switching signals before these saturation voltages. To accomplish this task, control circuit executes the following function in Figure 4.22.

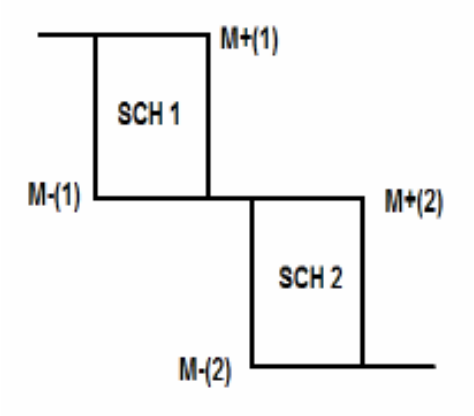

Figure 4.22 Schmitt trigger function

In order to leave some margins, values lower than saturation levels are selected as threshold values of Schmitt Triggers. The first schmitt trigger circuit has the high and low threshold voltages of 1.85V and 1.70V respectively. The second one has the values of 1.40V and 1.25V. It means that, with help of the logic circuits that follows the schmitt triggers, control circuit connects the extra capacitor C_X =4pF to the feedback capacitor C_f for voltages bigger than 1.85V or smaller than 1.25V. Afterwards, when the integrator output signal moves away from the saturation region, that is to say becomes smaller or bigger than 1.70V and 1.40V respectively, everything turns back into the way it was in an ordinary first order SD modulator.

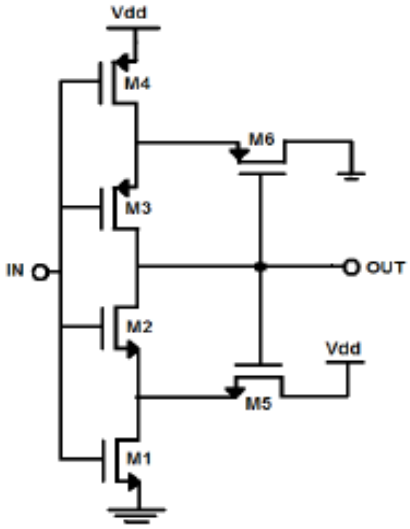

Figure 4.23 Schematic diagram of Schmitt Trigger

C_s and C_f are chosen to be 1pF and 2pF respectively. That means that by adding C_X =4pF, the gain of the integrator becomes three times smaller. The moment when the control signal adds the extra capacitor into the circuit, the integrator output will again continue to increase but this time with three times smaller gain. In other words, the integrator will sum the incoming voltages with three times smaller step sizes. At this point, it is essential to realize that when the saturation is far away owing to the proper switching that makes the extra capacitor to follow the output voltage of the integrator, the integrator output will continue to increase but from where it was before the connection. Thus, there will be no charge lost on this extra capacitor.

The output waveforms of the integrators of both the adaptive and generic first order SD modulators depicting the phenomena explained above can be seen in Figure 4.24.

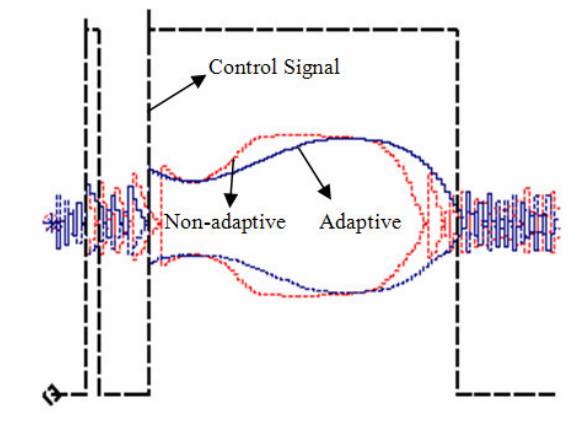

Figure 4.24 Integrator outputs of adaptive and non-adaptive circuits

It is obvious from Figure 4.24 that when the signal becomes bigger than 1.85V, the control signal is pulled up to the bias voltage which is 3.3V and then when the output signal decreases back below 1.70V, it turns back to its initial value of 0V. When the control signal is on, the integrator output begins to augment with smaller step sizes and eventually does not go into saturation whereas the non-adaptive one becomes clipped at the saturation voltage as it can be seen form above figure.

By courtesy of the differential and symmetrical structure of the circuit, a similar functioning is again valid for the lower side of the circuit which doesn't show up in Figure 4.19. In case of a decrease of voltage below 1.25V, the control signal becomes 3.3V to put C_X into the circuit and to decrease the step size of the integrator. Then, conversely, whilst the signal grows back to a level bigger than 1.40V, C_X becomes disconnected from the feedback loop.

More detailed and comprehensive views of the explained phenomena above can be seen below in Figure 4.25, 4.26 and 4.27. In Figure 4.25, it is obvious that due to the three times decrease in gain, the integrator output increases with three times smaller step sizes, 166mV instead of 500mV.

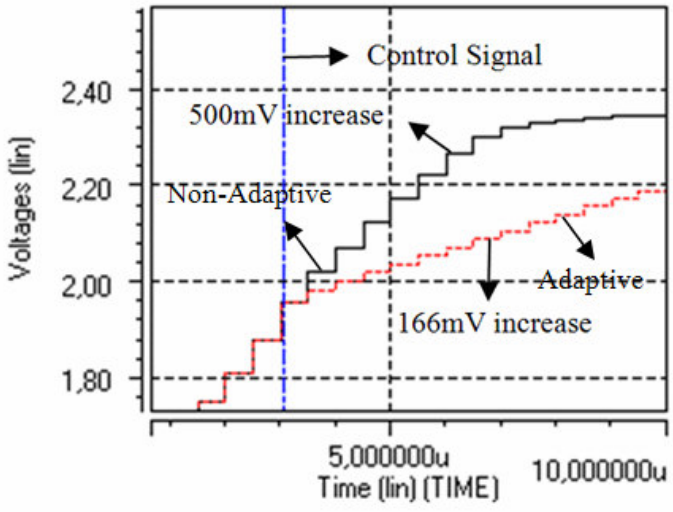

Figure 4.25 A zoomed view of the integrator outputs

In Figure 4.26, we see the behavior of the adaptive and non-adaptive circuits when a DC input is applied. The non-adaptive circuit comprising ideal non-saturated opamp continues to increase consistently with a step size equals to the DC input voltage whereas the one with non-ideal opamp enters saturation at 2.3V. However, adaptive circuit decreases its step size. Thus, the point where it enters saturation is not in the figure.

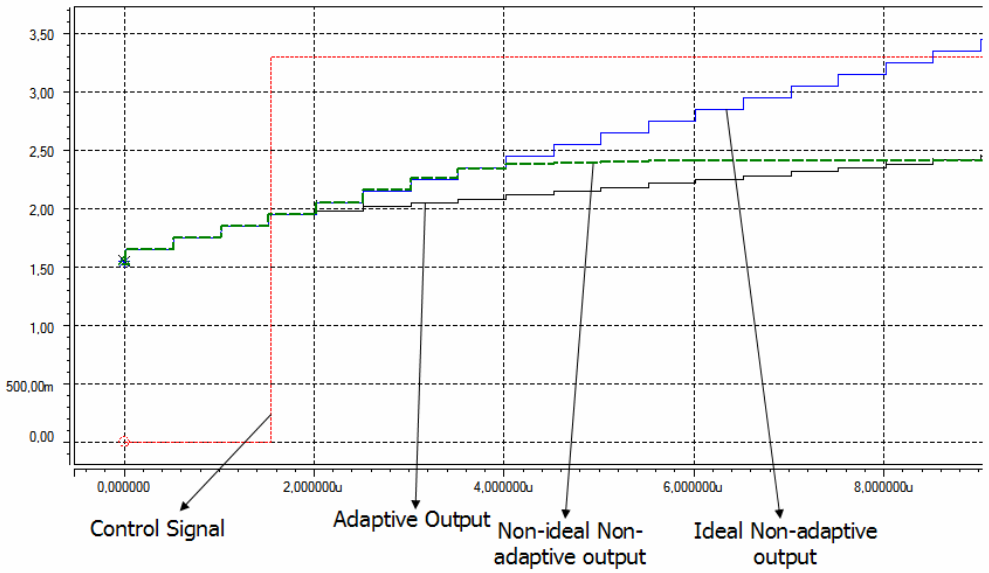

Figure 4.26 Integrator outputs of adaptive vs. non-adaptive circuits for DC input

The same analysis done with a DC signal is repeated with a sinusoidal input and has the following output waveform for one period of input sine wave. Again, adaptive circuit prevents the integrator from reaching saturation.

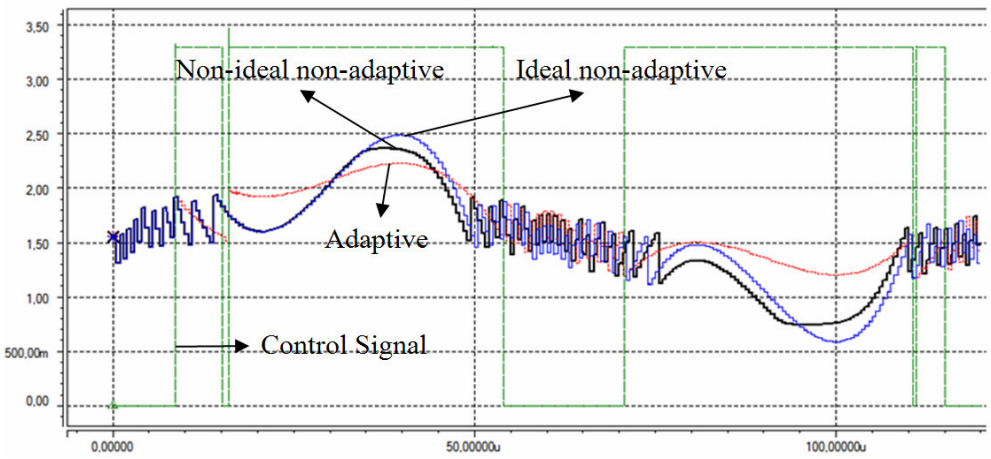

Figure 4.27 Integrator outputs of adaptive vs. non-adaptive circuits for sine input

4.1.4. Decimation Filter

When the gain of the integrator becomes three times smaller, the transfer function of the integrator changes accordingly. Then, this change causes a variation in the transfer function of the whole system. As we shall recall, linearized block diagram of a first order SD modulator is like this:

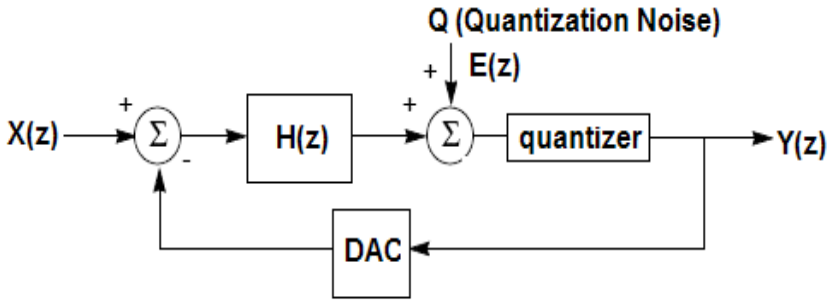

Figure 4.28 Linearized block diagram of a first order SD modulator

The transfer function of the feedback circuit that appears in Figure 4.28 may be computed as:

$$\frac{H(z)}{1+H(z)} \tag{4.2}$$

If the transfer functions of both the adaptive and non-adaptive circuits are computed according to the above formula, we derive the table 4.1. According to these computations, in order to make the circuits signal transfer function once again the same as the non-adaptive one especially during the interval of time in which the extra capacitance is connected to the feedback loop, we pass the output signal of the adaptive modulator through a specifically designed filter.

So, the aim is to make the signal transfer function equals to z^{-1} once again, By passing it through the filter in the intended interval of time, we simply multiply the output data of adaptive SD modulator with $3-2z^{-1}$ just when the extra capacitance is connected to circuit.

The block diagram of the mentioned filter that multiplies the incoming data by $3-2z^{-1}$ can be seen in Figure 4.29. The related code of the decimation filter can be found in Appendix A.2.3.

Table 4.1 Transfer functions of adaptive and non-adaptive circuits

	NON-ADAPTIVE	ADAPTIVE
INTEGRATOR	$H(z)=\frac{z^{-1}}{1-z^{-1}}$	$\frac{H(z)}{3}=\frac{z^{-1}}{3-3z^{-1}}$
CIRCUIT	$\frac{H(z)}{1+H(z)}$	$\frac{H(z)}{3+H(z)}$
SIGNAL	z^{-1}	$\frac{z^{-1}}{3-2z^{-1}}$
NOISE	$1-z^{-1}$	$\frac{3(1-z^{-1})}{3-2z^{-1}}$

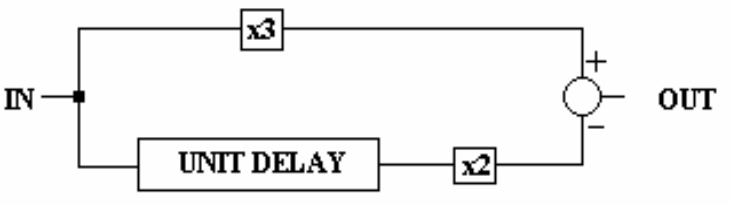

Figure 4.29 Block diagram of the decimation filter

4.1.5 Results, Conclusion and Drawbacks

After doing the necessary simulations using saturating inputs for both adaptive and non-adaptive SD modulators, following power spectral density plot in Figure 4.30 is acquired in the bandwidth of interest. It shows the PSD difference between the proposed adaptive circuit and the designed generic first order SD modulator. The output data is 8:1 decimated, passed through the decimation filter and low pass filtered in order to get this final waveform. Spice simulations are done for OSR=128. An increase of approximately 4dB in SNR is observed using this proposed architecture.

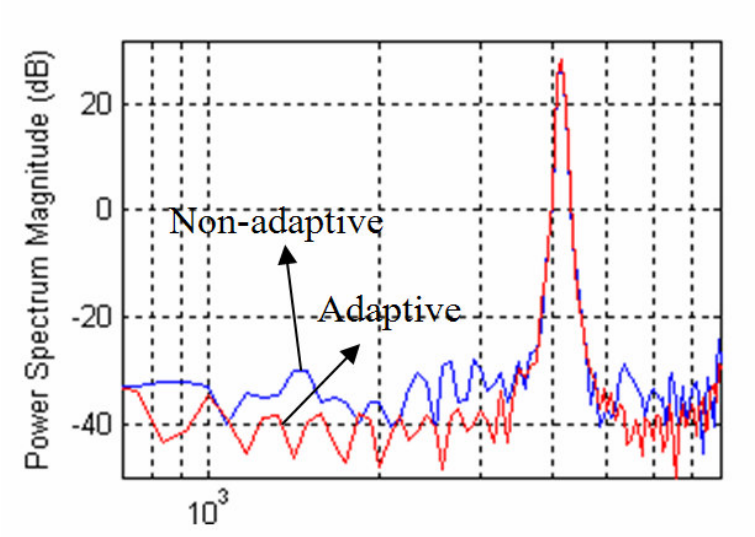

Figure 4.30 Output PSD plots of adaptive vs. ordinary SD modulator

Throughout the design and simulation part of this proposed architecture, generally sinusoidal signals are used as the input signals to the modulators. One main reason of this is to be able to compare the SNR of different architectures. Yet, speech signals are also

used as input signals to the modulators in the hope of getting better simulated results because, as mentioned above, one of the main application areas of the SD modulators is the processing of audio and speech signals where the required bandwidth is up to approximately 25 KHz at most. Additionally, with speech signals in which very high and very low signals reside, this proposed adaptive architecture will perform better.

Thanks to the structure of speech signals, input signal to the modulator will take high voltage values for very short intervals of time and therefore the extra capacitor will be rarely used in proposed architecture. Unfortunately, connecting the extra capacitor to the feedback path for a long time is no different than just to use a bigger feedback capacitance value. So, the amount of time, during which the control circuit decreases the gain of the integrator by adding extra capacitance, should be restricted to be approximately between 20 and 50 per cent of the input signal period in order to make the architecture meaningful and valid.

Additionally, having extremely large input values even for very short intervals makes the generation of a meaningful PSD plots almost impossible because when the input looks like a rectangular wave, its Fourier transform creates tones at higher frequencies.

Speech or speech like signals satisfies these requirements by definition. Therefore, they appear to be perfect input signal to test the proposed adaptive first order SD modulator. The simulation result of the proposed circuit using a speech like signal is shown in Figure 4.31 and 4.32.

A SNR improvement of approximately 4.9dB is computed between the adaptive and non-adaptive SD modulators using a speech like input signal. SNR is computed compare to the noise floor. The related Matlab and Hspice code can be found at Appendix A.2.4 and B.3 respectively.

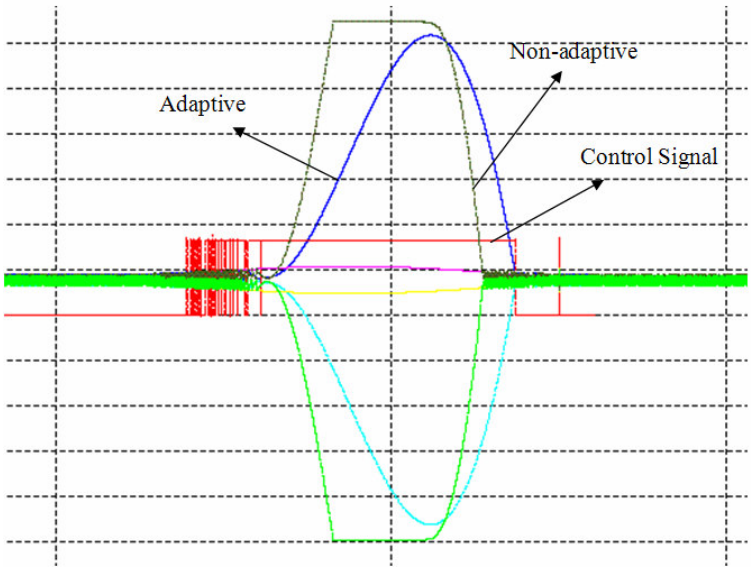

Figure 4.31 Integrator outputs of adaptive vs. ordinary SD modulator using a speech like input signal

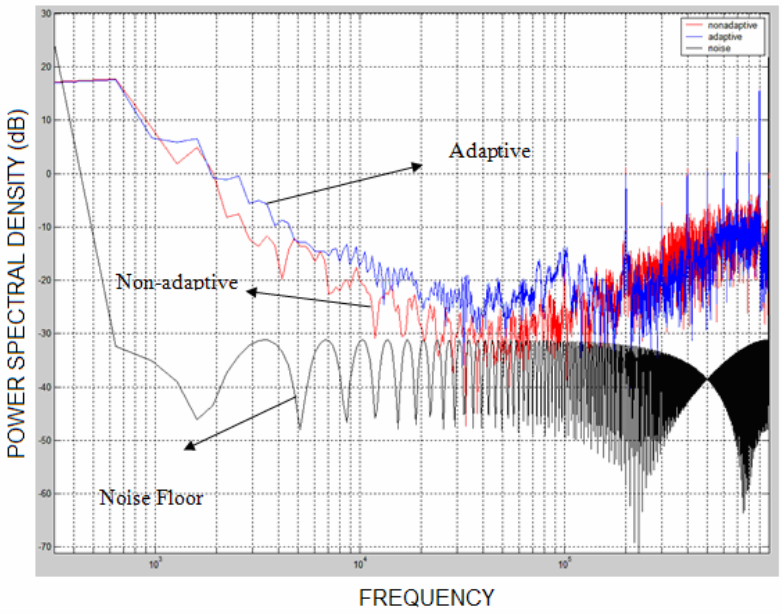

Figure 4.32 PSD plots of adaptive vs. ordinary SD modulator using a speech like input signal

Besides its power spectral density behavior, for general behavioral study, it is assumed that there is no clipping involved and all the results are analyzed in comparison to those of designed ordinary first order SD system.

The maximum integrator output voltage is one of the most critical parameter that avoids clipping. Figure 4.33 shows the maximum integrator output voltage vs. the input level of both the ordinary non-adaptive SD modulator and the adaptive one. The output voltage of the adaptive architecture is consistently well bounded throughout the full scale of the input, while the output voltage of the non-adaptive SD modulator starts to increase dramatically at large inputs. It indicates that the adaptive SD modulator architecture can avoid clipping at large inputs whereas the non-adaptive one does not. The effective bounding of the integrator output voltage enables the adaptive design to maximize the non-clipping input level to its full scale. Therefore, the adaptive architecture has a better dynamic range and can attain bigger SNR values.

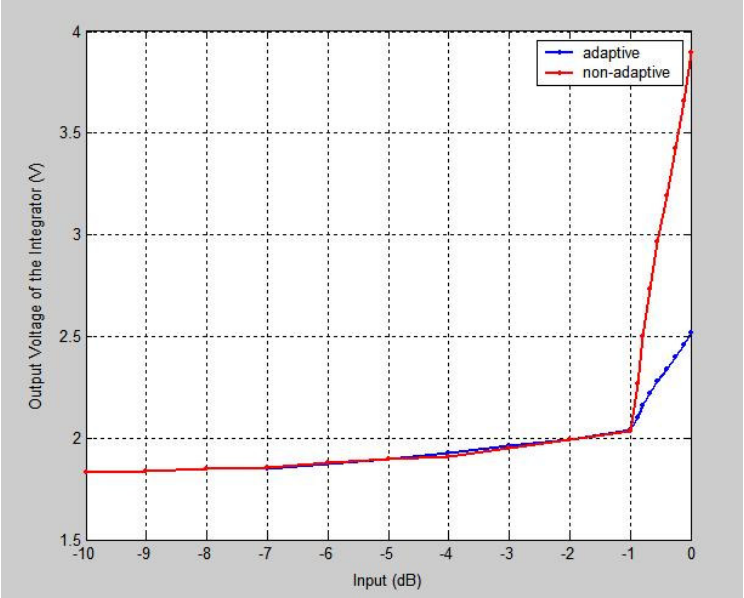

Figure 4.33 Maximum integrator output voltages of adaptive and non-adaptive SD Modulators

In conclusion the proposed variable gain architecture senses the saturation in advance and by automatically changing the gain of the integrator adapts itself to the current situation in order to prevent saturation completely or to decrease the time spent in the saturation region without ever stopping the conversion process. Circuit also corrects the deviations originated from changing gain thanks to a digital correction filter following the modulator output.

The proposed architecture finds a solution to the problem with saturation simply by just adding an extra capacitor to the circuit and achieves a 4dB SNR improvement. Furthermore, it performs better on speech or speech like signals by granting approximately 5 dB SNR improvement. Additionally, proposed variable gain architecture maximizes the non-clipping input level to its full scale. Therefore, it has a better dynamic range.

Nevertheless, it also has some major drawbacks. The proposed circuit doesn't perform well with the high sinusoidal input voltages and loses its perspective and validity when the extra capacitor stays connected to the feedback path for a long time. Additionally, using one bit SD modulator architecture prevents bigger SNR improvement.

4.2 Multi-bit Non-Uniform SD Modulator

4..2.1 Limitations of One Bit Architecture

Saturation of the integrator building block limits the dynamic range of the SD modulator and eventually distorts the output bit stream. The effect of saturation was exhaustively investigated above.

It was realized that even with the same simulation configuration and same input signals, multi-bit SD modulators show the effects of saturation much better than one bit SD modulator. The reason is that while using one bit SD modulator, even we have saturation levels very close to the common voltage level as in Figure 4.34, there will be no too big differences in the modulator output bit stream compare to unsaturated version of the modulator. As thoroughly explained in previously proposed architecture, one bit SD modulator proves this by not letting us to improve SNR furthermore.

The one bit quantizer functions with a simple principle. If it accepts a signal bigger than 1.65V as the input, it produces a '1' at the output. Conversely, for input voltages lower than 1.65 V, quantizer produces a '0'. So, no matter how the saturation level is close to 1.65V, the output value will not change and will be equal to '1' as long as the input grows bigger than V_{cm} and vice versa.

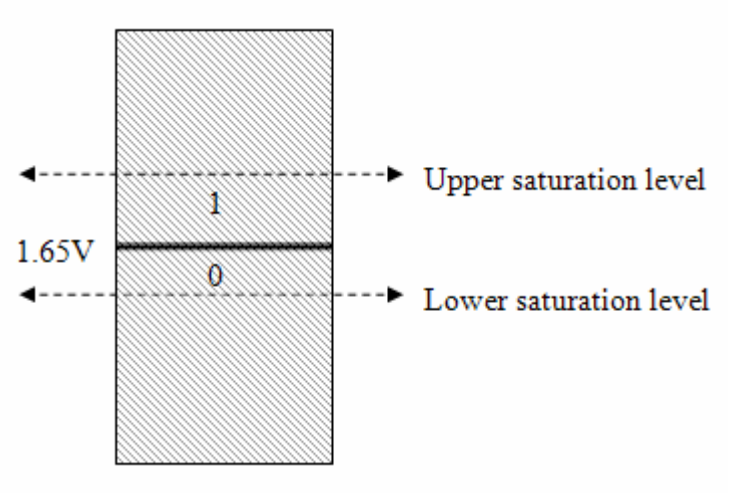

Figure 4.34 Saturation of one bit quantizer

However, for instance, if a two bit SD modulator and saturation levels as in Figure 4.35 are used, we observe a much bigger distortion due to the saturation as depicted previously in Figure 4.8. The reason is that saturation clips the quantizer input before it reaches to 2.475V and 0.875V. Therefore, potential '11' and '00' outputs never occur. The output of the modulator becomes distorted and bounded between values 01 and 10 instead of properly producing four different values.

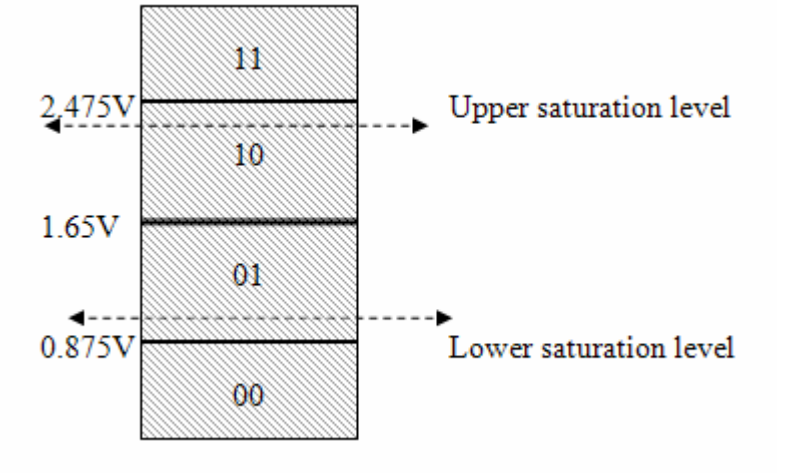

Figure 4.35 Saturation of two bit quantizer

It is important to point out that while comparing one bit and two bit SD modulators, same biggest and smallest DAC reference voltages are used as depicted in Figure 4.36.

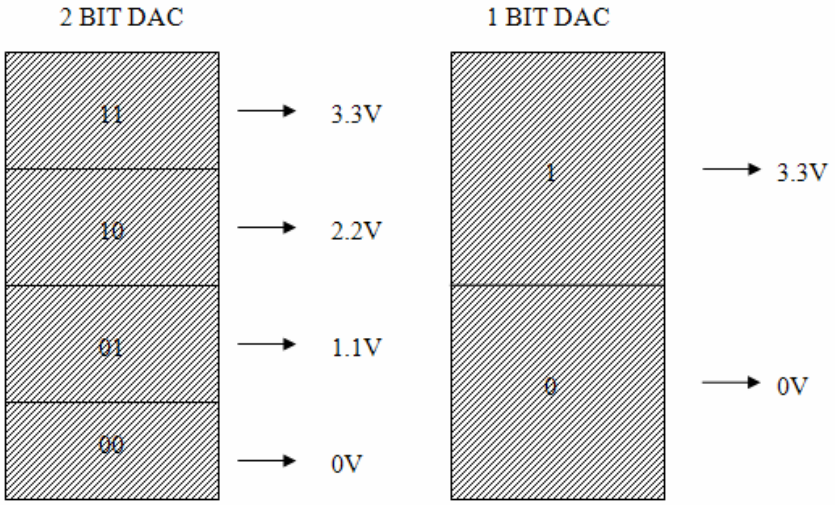

Figure 4.36 DAC structures of one bit and two bit SD modulators

Other DAC reference voltages are uniformly interpolated between 3.3V and 0V in multi-bit SD modulators.

To conclude the discussion above, by properly adjusting the quantization levels according to the saturation levels of the integrator, one can smooth the distortion created by saturation. Additionally, by using multi-bit configuration, one can also benefit from 6 dB SNR improvement per bit according to the formula 2.7. Therefore, a non-uniform multi-bit SD modulator may perform better compare to ordinary first order SD modulator.

4..2.2 Uniform Multi-bit SD Modulators

In order to be able to compare and measure the performance of the non-uniform multi-bit modulator, we have to first investigate the behavior and performance of uniform multi-bit SD modulators. Therefore, multi-bit SD modulators up to four bit are designed and tested using Matlab and Simulink respectively. The quantization levels that are used in modulators appear in Figure 4.37. DAC reference voltages are interpolated between 3.3V and 0V according to the used bit number. M-files of three and four bit quantization and DAC blocks are in Appendix A.1.4 and A.1.5 respectively.

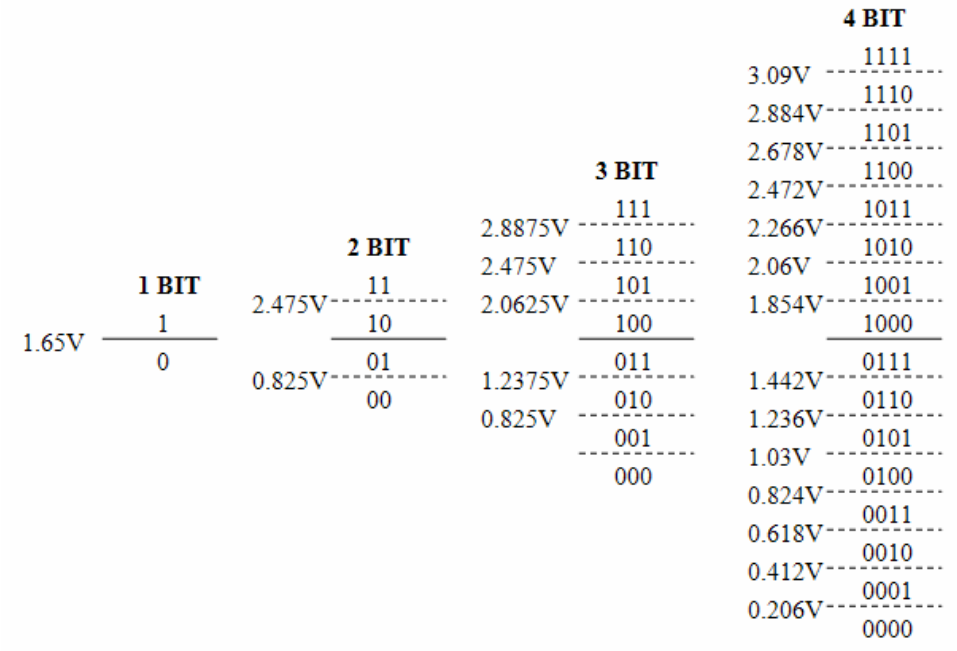

Figure 4.37 Multi-bit quantization levels

The waveforms of two bit first order SD modulator were plotted previously in Figure4.7. Below, in Figure 4.38 and 4.39, the waveforms of three and four bits are depicted. First plots are the inputs to the modulators. Second plots are integrator outputs and the last ones are outputs of the modulators. Simulations are done for the same input levels.

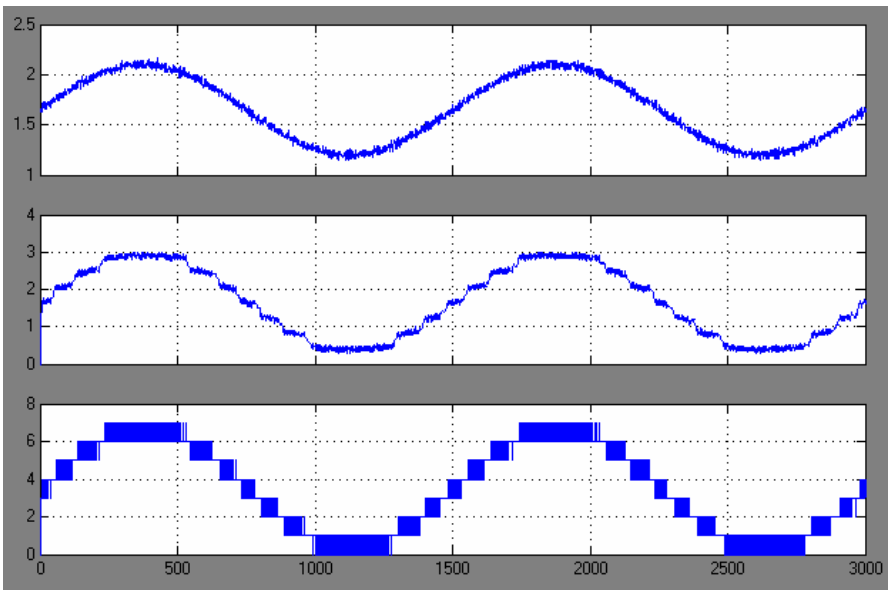

Figure 4.38 Three bit first order SD modulator outputs

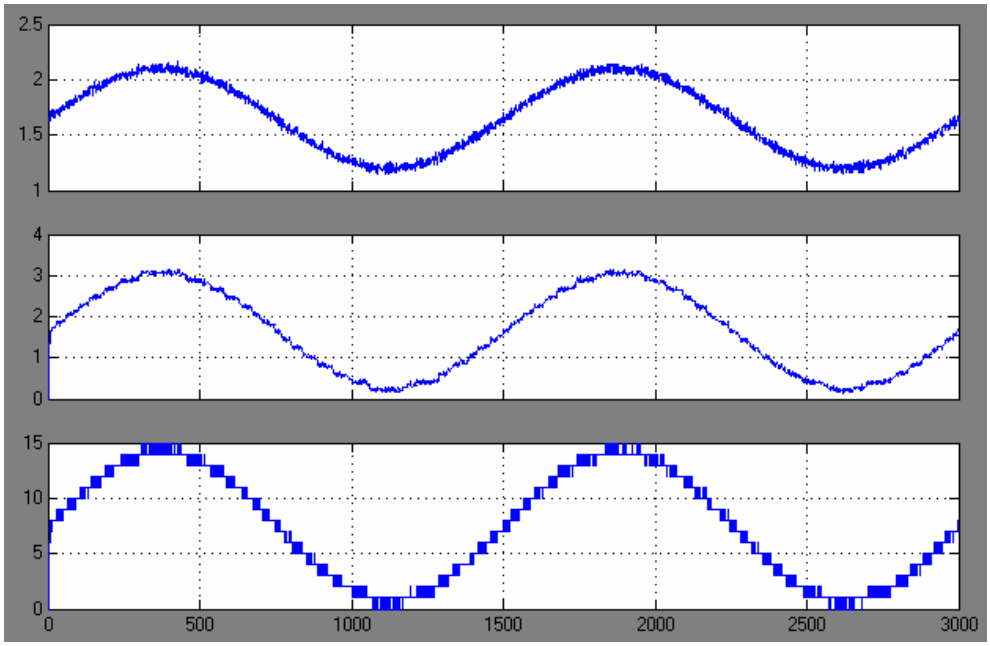

Figure 4.39 Four bit first order SD modulator outputs

Finally, the comparison of output waves of all the designed multi-bit SD modulators can be seen below. The waves that are plotted are the ones just before the decimation.

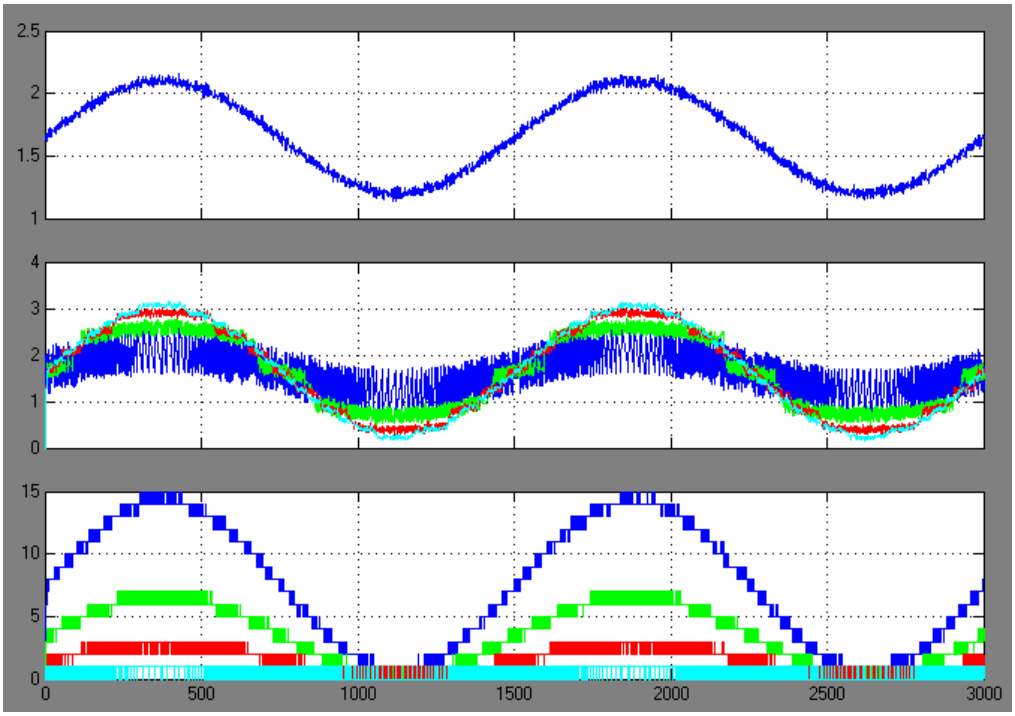

Figure 4.40 Comparison of multi-bit first order SD modulators outputs

PSD comparison of different multi-bit SD modulators is depicted in Figure 4.41.

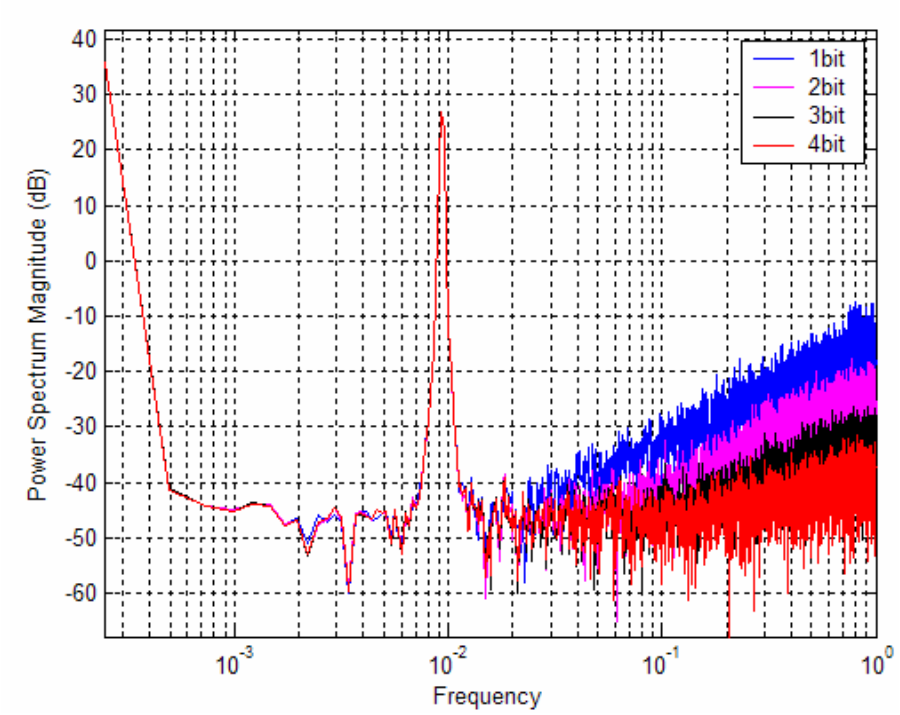

Figure 4.41 PSD comparison of multi-bit first order SD modulators outputs

As expected, it is clear from the above plot that by increasing the bit number used in quantization, we also increase the SNR of the modulator.

4..2.3 Structure of Non-Uniform Multi-bit SD Modulator

Usage of multi-bit SD modulators grants us 6 dB SNR improvements per bit but also makes the system more vulnerable against the clipping created by saturation as formerly showed in two bit uniform first order SD modulator example.

Therefore, to get a system that is more resistant to the effects of saturation and that has a bigger SNR, both the advantages of one bit and multi-bit modulators can be integrated in one non-uniform multi-bit SD modulator. For instance, one bit modulator is more resistant to the clipping of the integrator output than any other multi-bit SD modulator. The reason of this behavior was formerly explained. Therefore, one may design a four bit quantizer that turns and behaves like a one bit quantizer when there is clipping due to saturation

The idea above is depicted in Figure 4.42 below. Non-uniform modulator uses just one quantization level above and below the predefined saturation levels different from four bit quantizer that uses four different quantization levels again above and below the saturation levels. Thus, for an increasing voltage from V_{cm}, the non-uniform modulator acts like four bit modulator. Yet, right before the clipping of the integrator outputs due to the saturation, it passes a new quantization level and stays there as a one bit modulator in spite of the growing voltage.

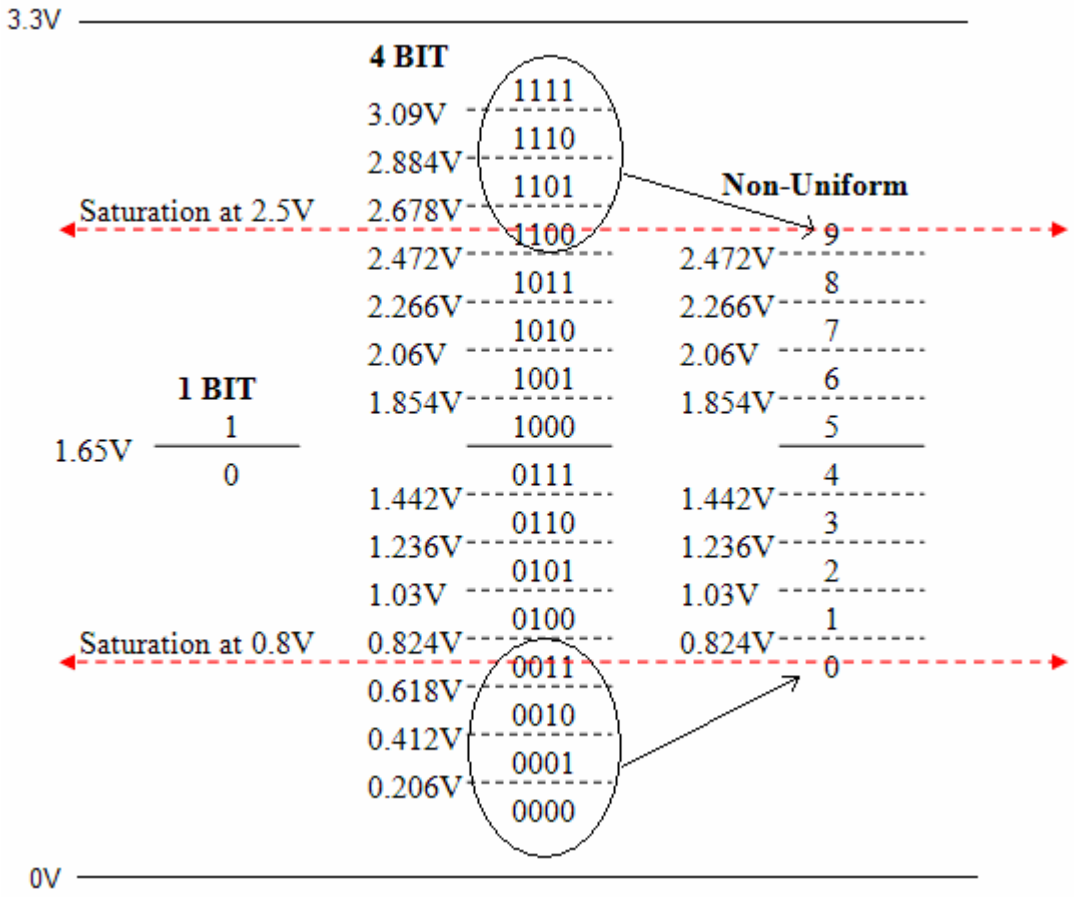

Figure 4.42 The structure of non-uniform quantizer

The digital to analog converters of the non-uniform SD modulator is again designed as in four bit modulator. The reference voltages are uniformly interpolated between 3.3V and 0V for ten different values. Below, in Figure 4.43 and 4.44, the behaviors of one bit and four bit SD modulators are shown. As it can be easily realized, four bit structure enters into saturation and has its output bit stream deformed while one bit structure does not.

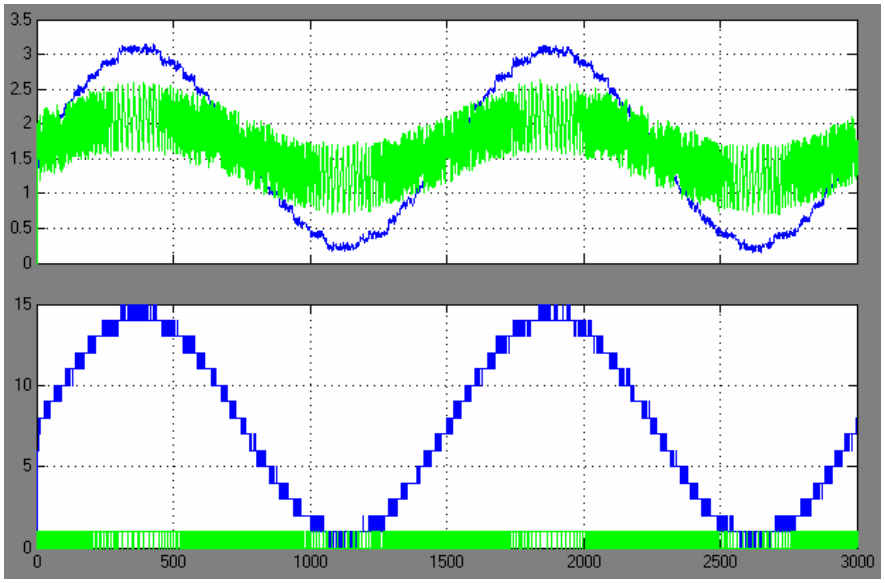

Figure 4.43 Comparison of one bit vs. four bit modulators without any saturation

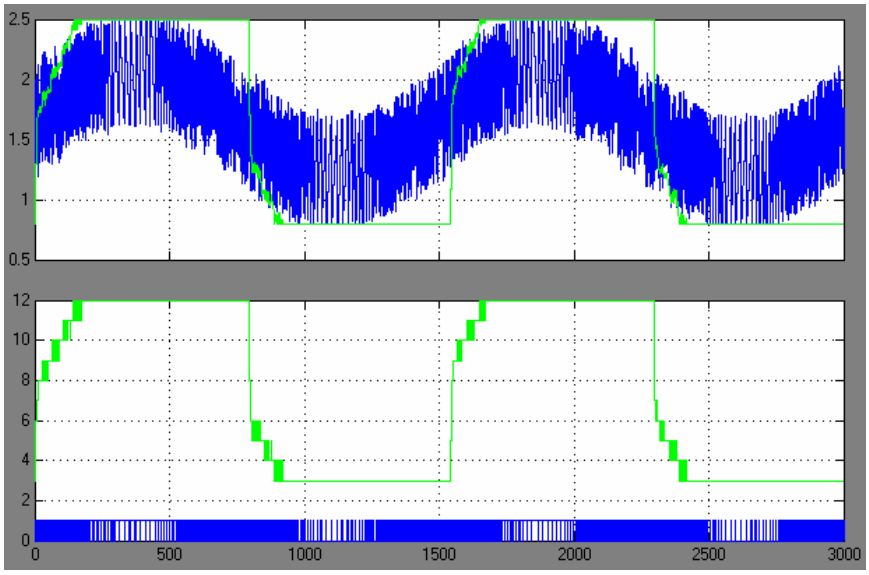

Figure 4.44 Comparison of one bit vs. four bit modulators with saturation

4..2.4 Results and Conclusion

When the performance of the non-uniform modulator is tested using the same simulation configuration of one and four bit first order SD modulators, following results are collected. M-files of the quantizer and DAC blocks of non-uniform multi-bit SD modulator can be found in Appendix A.1.6.

Below, we see the waveform comparison of the modulators. As intended, non-uniform multi-bit SD modulator is not affected from the clipping of the integrator output due to the saturation as it can be seen from Figure 4.45.

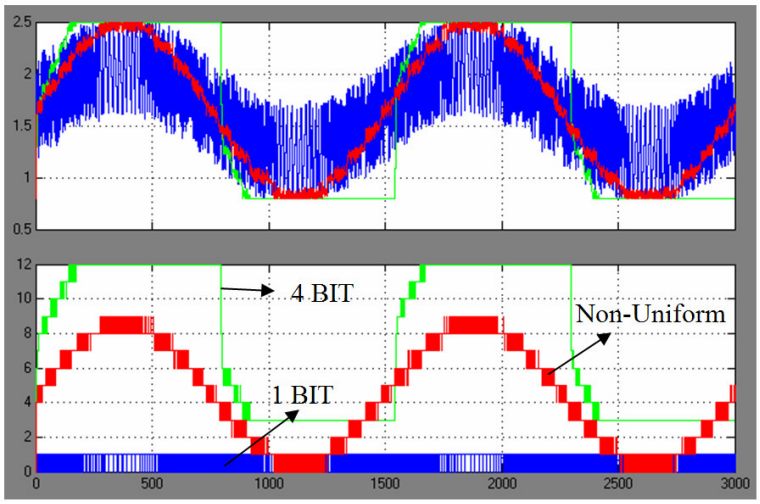

Figure 4.45 Waveforms of 1 bit, 4 bit and non-uniform first order SD modulators

The PSD comparison of one bit first order SD modulator vs. non-uniform multi-bit one is depicted in Figure 4.46. The plots are acquired after 8:1 decimating the modulator outputs for a highly saturated input.

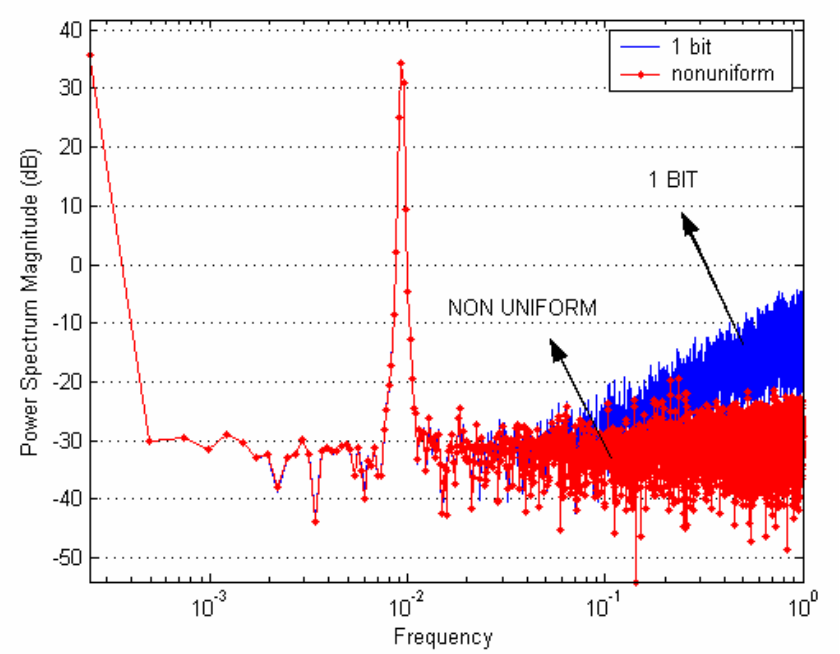

Figure 4.46 PSD plots of one bit SD modulator vs. proposed architecture

It can be easily said that non-uniform multi-bit first order SD modulator has better SNR properties compare to one bit architecture. Finally, PSD plots of one four bit uniform first order SD modulator vs. non-uniform multi-bit one is depicted in Figure 4.47.

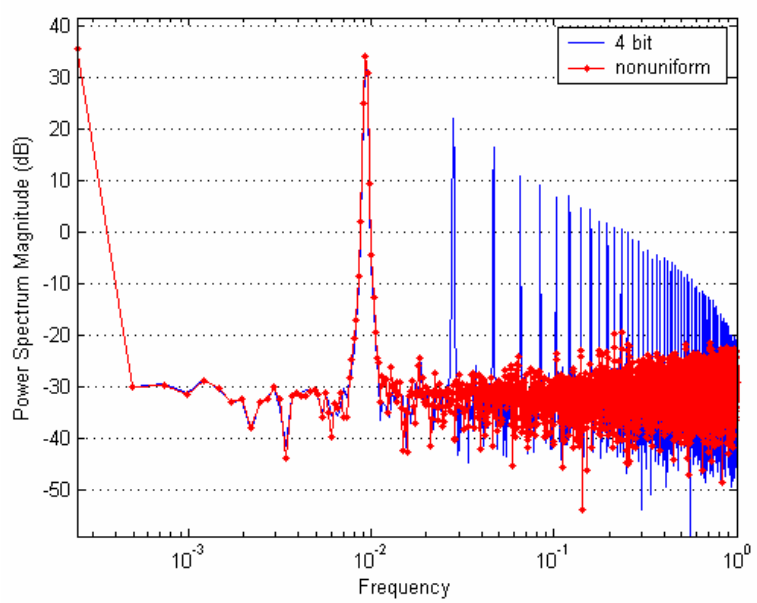

Figure 4.47 PSD plots of four bit SD modulator vs. proposed architecture

SNR of saturated four bit architecture is approximately 10 dB whereas SNR of the proposed architecture is approximately 60 dB. Dynamic range plot of both the proposed circuit vs. four bit uniform first order SD modulator can be found in Figure 4.48.

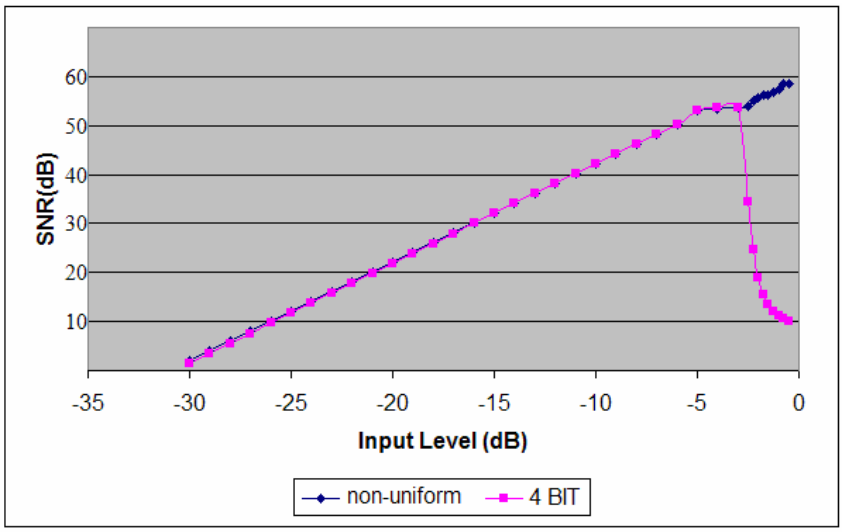

Figure 4.48 Dynamic range plot of the proposed architecture

It has been shown that a non uniform multi-bit sigma-delta modulator improves the input signal range over its one-bit and four bit counterparts. Therefore, a non-uniform multi-bit SD data converter can achieve a given signal-to-noise ratio performance objective with a lower oversampling ratio and/or lower modulation order than an one-bit converter, resulting in significant reduction of power dissipation and noise coupling from digital to analog circuits as well as ease of circuit design in the overall system. The main problem with multi-bit sigma-delta converters is also valid for the proposed circuit which is the internal multi-bit D/A process which leads to instability, distortion and noise effects in the overall converter.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

Sigma Delta Analog to Digital converter has come into widespread use as high resolution converter in recent years with the improvements in silicon technology. Even though, it was first presented in the early 1960s, it still has some unsolved problems. One of these problems is the saturation of the integrator building block of the modulator.

Integrator must be prevented from going into saturation in order to get correct and meaningful data out of the modulator. In this work, two different solutions to this problem are proposed. Each of them is exhaustively investigated and tested in the course of the preparation of this thesis.

First method solves the problem by changing the gain of the integrator. Proposed architecture automatically senses the saturation in advance. When the output voltage of the integrator reaches the edge of saturation, the designed circuit properly adapts the gain of the integrator to prevent it from clipping by simply adding an extra capacitor to the feedback path of the opamp and eventually decreases the summing step size of the integrator. Conversely, when the integrator moves away from the saturation region, the added capacitor is removed from the circuit. The operation is done with help of a control circuit consisting of Schmitt Triggers and logic circuits. As a result approximately 4 dB SNR improvements for sinusoidal inputs and 5 dB SNR improvements for speech and speech like signal are achieved. Yet, this method has some limitations and drawbacks. The amount of time in which the extra capacitor is connected to the circuit must be carefully checked in order to keep the proposed circuit still meaningful and valid. Additionally, for very large inputs signals, proposed system doesn't perform well.

Second method tries to solve the same problem by using non-uniform multi-bit quantizer architecture. One bit SD modulators are more resistant to the clipping effect of the saturation compare to multi-bit architectures. However, it is also known that multi-bit

SD architectures have much better SNR properties than one bit modulators. The proposed circuit tries to take advantage of both architectures by acting like a multi-bit modulator when output of the integrator is far away from saturation region and like a one bit modulator when the circuit goes into saturation. The proposed circuit has better SNR characteristics compared to ordinary one bit first orders SD modulator and four bit uniform first order SD modulator.

5.2 Future Work

All the work until now is done using first order Sigma Delta data converter architecture. As a future work, the proposed systems can be implemented for higher order SD modulator architectures and their performance may be evaluated accordingly.

Additionally, in order the check the validity of the design and evaluate the performance of the proposed variable gain architecture correctly, running simulations that consists of both the analog and the digital parts at the same time using VHDL-AMS is a must as future work. Thereafter, the proposed systems can be designed and produced as integrated circuits in order to see and measure their real life performance.

Finally, one other future work that can be done is to combine the principles used in the two proposed architectures. A SD modulator that adapts its gain by automatically sensing the saturation in advance and that uses a non-uniform multi-bit quantizer resisting the clipping due to the saturation of the integrator, can be created.

APPENDIX A: M FILES

A.1. Simulink Codes

A.1.1. One Bit Quantization and DAC for Constant Input

```
function[y]=dc_quan(u)
if(u>0)
   y=1;
else
   y=0;
end

function[y]=dc_DAC(u)
if(u==1)
   y=1;
else
   y=0;
end
```

A.1.2. One Bit Quantization and DAC for Sine Input

```
function[y]=quan_1bit(u)
if(u>1.65)
   y=1;
else
   y=0;
end

function[y]=DAC_1bit(u)
if(u==1)
   y=2.15;
```

```
else
   y=1.15;
end
```

A.1.3. Two bit Quantization and DAC for Sine Input

```
function[y]=DAC_2bit(u)
if(u==0)
   y=1.15;
end
if(u==1)
   y=1.4833;
end
if(u==2)
   y=1.8166;
end
if(u==3)
   y=2.15;
end

function[y]=quan_2bit(u)
if(u<0.825)
   y=0;
end
if(u>=0.825 & u<1.65)
   y=1;
end
if(u>=1.65 & u<2.475)
   y=2;
end
if(u>=2.475)
   y=3;
end
```

A.1.4. Three bit Quantization and DAC for Sine Input

```
function[y]=quan_3bit(u)
if(u<0.4125)
   y=0;
end
if(u>=0.4125 & u<0.825)
   y=1;
end
if(u>=0.825 & u<1.2375)
   y=2;
end
if(u>=1.2375 & u<1.65)
   y=3;
end
if(u>=1.65 & u<2.0625)
   y=4;
end
if(u>=2.0625 & u<2.475)
   y=5;
end
if(u>=2.475 & u<2.8875)
   y=6;
end
if(u>=2.8875)
   y=7;
end

function[y]=DAC_3bit(u)
if(u==0)
   y=0;
end
if(u==1)
```

```
    y=0.471428;
end
if(u==2)
    y=0.942857;
end
if(u==3)
    y=1.41428;
end
if(u==4)
    y=1.885714;
end
if(u==5)
    y=2.3571;
end
if(u==6)
    y=2.82857;
end
if(u==7)
    y=3.3;
end
```

A.1.5. Four bit Quantization and DAC for Sine Input

```
unction[y]=quan_4bit(u)
if(u<0.206)
    y=0;
end
if(u>=0.206 & u<0.412)
    y=1;
end
if(u>=0.412 & u<0.618)
    y=2;
end
```

```
if(u>=0.618 & u<0.824)
   y=3;
end
if(u>=0.824 & u<1.03)
   y=4;
end
if(u>=1.03 & u<1.236)
   y=5;
end
if(u>=1.236 & u<1.442)
   y=6;
end
if(u>=1.442 & u<1.65)
   y=7;
end
%--------------------------------
if(u>=1.65 & u<1.854)
   y=8;
end
if(u>=1.854 & u<2.06)
   y=9;
end
if(u>=2.06 & u<2.266)
   y=10;
end
if(u>=2.266 & u<2.472)
   y=11;
end
if(u>=2.472 & u<2.678)
   y=12;
end
if(u>=2.678 & u<2.884)
   y=13;
```

```
end
if(u>=2.884 & u<3.09)
   y=14;
end
if(u>=3.09)
   y=15;
end

function[y]=DAC_4bit(u)
if(u==0)
   y=0;
end
if(u==1)
   y=0.22;
end
if(u==2)
   y=0.44;
end
if(u==3)
   y=0.66;
end
if(u==4)
   y=0.88;
end
if(u==5)
   y=1.1;
end
if(u==6)
   y=1.32;
end
if(u==7)
   y=1.54;
end
```

```
%-----------------------
if(u==8)
   y=1.76;
end
if(u==9)
   y=1.98;
end
if(u==10)
   y=2.2;
end
if(u==11)
   y=2.42;
end
if(u==12)
   y=2.64;
end
if(u==13)
   y=2.86;
end
if(u==14)
   y=3.08;
end
if(u==15)
   y=3.3;
end
```

A.1.6. Non-uniform multibit quantization and DAC for Sine Input

```
function[y]=quan_nonuniform(u)
if(u<0.824)
   y=0;
end
if(u>=0.824 & u<1.03)
```

```
    y=1;
end
if(u>=1.03 & u<1.236)
    y=2;
end
if(u>=1.236 & u<1.442)
    y=3;
end
if(u>=1.442 & u<1.65)
    y=4;
end
if(u>=1.65 & u<1.854)
    y=5;
end
if(u>=1.854 & u<2.06)
    y=6;
end
if(u>=2.06 & u<2.266)
    y=7;
end
if(u>=2.266 & u<2.472)
    y=8;
end
if(u>=2.472)
    y=9;
end
function[y]=DAC_nonuniform(u)
if(u==0)
    y=0;
end
if(u==1)
    y=0.3666;
end
```

```
if(u==2)
   y=0.733;
end
if(u==3)
   y=1.09;
end
if(u==4)
   y=1.46;
end
if(u==5)
   y=1.83;
end
if(u==6)
   y=2.19;
end
if(u==7)
   y=2.56;
end
if(u==8)
   y=2.93;
end
if(u==9)
   y=3.3;
end
```

A.2. Matlab Codes

A.2.1. Input and Output Decimation Codes

```
function[out]=inputdecimation(in,bit)
%decimate the input according to the given bit number
m=size(in,1);
i=1;
```

```
j=1;
r=1;
count=0;
data=2^bit;
while(i<=m)
   count=count+1;
   r=mod(count,(data-1));
   if(r==0 | i==1)
      out(j,1)=in(i,1);
      j=j+1;
   end
   i=i+1;
end

function[out1,out2]=outputdecimation(in,bit,offset,input,inbit)
%decimates the output data
%inbit represents the quantization bit of the input
m=size(in,1);
i=1;
j=1;
num=0;
rem=0;
count=1;
data=2^bit;
min=offset-input;
vpp=input*2;
while(i<=m)
   rem=mod(count,(data-1));
%-------------------------------------------------------------------
%detects the data according to the bit number of the input
   if(inbit==1)
      if(in(i,1)==1)
         num=num+1;
```

```
        end
    else
        num=num+in(i,1);
    end
%-----------------------------------------------------------------------

    if(rem==0)
        if(inbit==2)
            num=num/3;
        end
        if(inbit==3)
            num=num/7;
        end
        if(inbit==4)
            num=num/15;
        end
        if(inbit==5)
            num=num/9;
        end
        val=num/(data-1);
        out1(j,1)=num;
        out2(j,1)=(vpp*val)+min;
        j=j+1;
        num=0;
    end
    count=count+1;
    i=i+1;
end
```

A.2.2. Quantization of Spice Output

```
%quantization of non-adaptive circuit
function[out]=quan2(in)
```

```
close all;
i=1;
j=1;
max1=size(in,1);
max2=floor((size(in,1)/50)-1);

%raw data check
while(i<=max1)
   if(in(i,2)>3.5)
      sprintf('%s','there is a value bigger than 3.5V')
   end
   if(in(i,2)<-0.5)
      sprintf('%s','there is a value smaller than -0.5V')
   end
   i=i+1;
end

%raw data plot
figure;
plot(in(:,2));
grid on;
title('output plot before quantization');
%quantization by checking
while(j<=max2)
   out(j,1)=j;
   if(in(15+(50*j),2)>3.25 & in(15+(50*j),2)<3.35)
      out(j,2)=1;
   elseif(in(15+(50*j),2)>-0.05 & in(15+(50*j),2)<0.05)
      out(j,2)=0;
   else
      %if the data is not in the band
      sprintf('%s','an error occured during quantization')
   end
```

```
   if(in(15+(50*j),3)>1.55)
      out(j,3)=1;
   elseif(in(15+(50*j),3)<1.55)
      out(j,3)=0;
   else
      %if the data is not in the band
      sprintf('%s','an error occured during quantization of control signal')
   end

   j=j+1;
end

%figure;
plot(out(:,2));
grid on;
title('output plot after quantization');
```

A.2.3. Decimation Filter Code

```
%filter to correct the sigma delta output
function[out]=correctionfilter(x)
m=size(x,1);
i=1;
count=0;
while(i<=m)
   out(i,1)=x(i,1);
   out(i,3)=x(i,3);
   if(out(i,3)==1)
      count=count+1;
      if(count==1)
         out(i,2)=0;%3*x(i,2);
      else
```

```
            out(i,2)=(3*x(i,2))-(2*x(i-1,2));
        end
    end
    if(out(i,3)==0)
        count=0;
        out(i,2)=x(i,2);
    end
    i=i+1;
end
```

A.2.4. SNR Computation of Adaptive Circuit with Speech Like Signal

```
%SL input SD converter SNR computation
function[]=snrSL(nonadaptive,adaptive,noise,fs,limit,begin)
close all;
w=1;
count=0;
sumNN=0;
sumGG=0;
sumAA=0;

n=quantize(nonadaptive);
a=quantize(adaptive);
g=quantize(noise);

gg=g(:,2);
nn=n(:,2);
aa=a(:,2);

m=floor(size(nn,1)/2);

[ampN,fN]=PSD(nn,m,fs);
[ampA,fA]=PSD(aa,m,fs);
```

```
[ampG,fG]=PSD(gg,m,fs);

dBampN=10*log10(ampN);
dBampA=10*log10(ampA);
dBampG=10*log10(ampG);

figure                ;
                      semilogx(fN,dBampN,'r');
                      hold on;
grid on;
semilogx(fN,dBampA,'b');
hold on;
semilogx(fN,dBampG,'m');
axis tight;
legend('nonadaptive','adaptive','noise');


r=1;
py=size(fN,1);
while(w<=py)
   if(fN(w,1)<=limit && fN(w,1)>=begin)
      dBampNN(r,1)=dBampN(w,1);
      sumNN=sumNN+dBampNN(r,1);

      dBampAA(r,1)=dBampA(w,1);
      sumAA=sumAA+dBampAA(r,1);

      dBampGG(r,1)=dBampG(w,1);
      sumGG=sumGG+dBampGG(r,1);

      flimited(r,1)=fN(w,1);
      r=r+1;
      count=count+1;
```

```
    end
    w=w+1;
end

signalN=sumNN/count
signalA=sumAA/count
noise=sumGG/count

snrN=signalN-noise
snrA=signalA-noise
snr=snrA-snrN

figure;
semilogx(flimited,dBampNN,'r');
hold on;
grid on;
semilogx(flimited,dBampAA,'b');
hold on;
semilogx(flimited,dBampGG,'m');
axis tight;
legend('nonadaptive','adaptive','noise');
```

A.2.5. Miscellaneous Codes

```
%does what the adaptive circuit does
function[y]=circuitfunction(x)
%x=input vector
i=1;
m=size(x,1);
while(i<=m)
    y(i,1)=x(i,1);
    y(i,3)=x(i,3);
    if(x(i,3)==1)
```

```
        y(i,2)=(x(i-1,2)+(2*y(i-1,2)))/3;
    else
        if(i==1)
            y(i,2)=0;
        else
            y(i,2)=x(i-1,2);
        end
    end
    i=i+1;
end
```

APPENDIX B: H-SPICE CODES

B.1. First Order Non-Adaptive Sigma Delta Converter

B.1.1. Top Level Code

```
**********************************************************************
SD Converter with sinusoidal input
vin1 in1 0 sin(1.55 400m 8000
vin2 in2 0 sin(1.55 -400m 8000)
**********SWITCHS*****************************************
xsw1 in1 2  clk idswt
xsw2 in2 4  clk idswt
xsw3 3 cm  clk idswt
xsw4 5 cm  clk idswt
xsw5 3 intp clk2 idswt
xsw6 5 intn clk2 idswt
xsw7 DAC2 2 clk2 idswt
xsw8 DAC1 4 clk2 idswt
**********CIRCUIT*****************************************
x1 intp intn intop inton cm opamp
xcomp1 bias 0 intop inton clk outp outn comparator
x3 refn refp outp outn DAC1 bias 0 0 mux
x4 refp refn outp outn DAC2 bias 0 0 mux
**********CAPACITORS*****************************************
c1 2 3 1p
c2 4 5 1p
c3 intp inton 2p
c4 intn intop 2p
**********VOLTAGES*****************************************
Vclk clk 0  pulse(3.3 0 0n  1n 1n 270n 500n)
Vclk2 clk2 0  pulse(0 3.3 20n 1n 1n 230n 500n)
Vcm cm 0 1.55
```

```
Vbias bias 0 3.3
Vrefp refp 0 2.05
Vrefn refn 0 1.05
**********INCLUDES**********************************************
.include iddifopamp.sp
.include idswt.sp
.include comparator4.sp
.include mux.sp
.include modn.mod
.include modp.mod
*******************************************************************
.options post nomod probe
.ic v(inton)=1.55 v(intop)=1.55
.ic v(outp)=0 v(outn)=0
.ic v(intp)=1.55 v(intn)=1.55
.tran 10n 375u
.probe v(outn) v(intop) v(inton) v(in1) v(in2)
.print v(outn)
.end
```

B.1.2. Subcircuits

```
********Comparator*********************************************
.subckt comparator 1  101  25  26   99  outn  outp
vbias 27 101 0.750
r2 1 24 24000
r1 1 23 24000
M21 24 26 22 101 modn w=100E-6 l=2E-6 as=2.700000e-10 ad=2.700000e-10
+ps=1.830000e-04 pd=1.830000e-04 nrs=4.166667e-03 nrd=4.166667e-03
M22 23 25 22 101 modn w=100E-6 l=2E-6 as=2.700000e-10 ad=2.700000e-10
+ps=1.830000e-04 pd=1.830000e-04 nrs=4.166667e-03 nrd=4.166667e-03
M23 22 27 28 101 modn w=50E-6 l=2E-6 as=1.800000e-10 ad=1.800000e-10
```

```
+ps=1.230000e-04 pd=1.230000e-04 nrs=6.250000e-03 nrd=6.250000e-03
vdummy 28 101 0
**********************************************************************
M1 2 23 5 101 modn w=60E-6 l=3E-6 as=1.800000e-10 ad=1.800000e-10
+ps=1.230000e-04 pd=1.230000e-04 nrs=6.250000e-03 nrd=6.250000e-03
M2 3 24 5 101 modn w=60E-6 l=3E-6 as=1.800000e-10 ad=1.800000e-10
+ps=1.230000e-04 pd=1.230000e-04 nrs=6.250000e-03 nrd=6.250000e-03
M8 2 3 1 1 modp w=60E-6 l=3E-6 as=6.750000e-11 ad=6.750000e-11
+ps=4.800000e-05 pd=4.800000e-05 nrs=1.666667e-02 nrd=1.666667e-02
M9 3 2 1 1 modp w=60E-6 l=3E-6 as=6.750000e-11 ad=6.750000e-11
+ps=4.800000e-05 pd=4.800000e-05 nrs=1.666667e-02 nrd=1.666667e-02
M3 2 3 101 101 modn w=20E-6 l=3E-6 as=6.750000e-11 ad=6.750000e-11
+ps=4.800000e-05 pd=4.800000e-05 nrs=1.666667e-02 nrd=1.666667e-02
M4 3 2 101 101 modn w=20E-6 l=3E-6 as=6.750000e-11 ad=6.750000e-11
+ps=4.800000e-05 pd=4.800000e-05 nrs=1.666667e-02 nrd=1.666667e-02
M7 2 99 3 101 modn w=25E-6 l=1.5e-6 as=7.500000e-11 ad=7.500000e-11
+ps=5.300000e-05 pd=5.300000e-05 nrs=1.500000e-02 nrd=1.500000e-02
M10 5 99 101 101 modn w=6E-6 l=6e-6 as=1.800000e-11 ad=1.800000e-11
+ps=1.500000e-05 pd=1.500000e-05 nrs=6.250000e-02 nrd=6.250000e-02
**********************************************************************
M11 outp 3 1 1 modp w=30E-6 l=1.2e-6 as=9.000000e-12 ad=9.000000e-12
+ps=9.000000e-06 pd=9.000000e-06 nrs=1.250000e-01 nrd=1.250000e-01
M12 outp 3 101 101 modn w=12E-6 l=1.2e-6 as=3.600000e-12 ad=3.600000e-12
+ps=5.400000e-06 pd=5.400000e-06 nrs=3.125000e-01 nrd=3.125000e-01
M13 outn 2 1 1 modp w=30E-6 l=1.2e-6 as=9.000000e-12 ad=9.000000e-12
+ps=9.000000e-06 pd=9.000000e-06 nrs=1.250000e-01 nrd=1.250000e-01
M14 outn 2 101 101 modn w=12E-6 l=1.2e-6 as=3.600000e-12 ad=3.600000e-12
+ps=5.400000e-06 pd=5.400000e-06 nrs=3.125000e-01 nrd=3.125000e-01
.ends
```

```
**********************************************************************
*Fully differential ideal comparator
**********************************************************************
.subckt iddifcomp inp inn outp outn
.PARAM vohigh=3.3v volow=0v rbval=1k rfval=1k
Ecomp1 outn 0 PWL(1) inp inn -1u,vohigh 1u,volow
Ecomp2 outp 0 PWL(1) inp inn -1u,volow  1u,vohigh
Rb inn 0 rbval
Rf inn out rfval
Cb inn 0 1ff
.ends
**********************************************************************
*Fully differential ideal opamp with VCM
*by Daghan Gokdel 10/09/2006 Sunday
**********************************************************************
.subckt opamp inp inn outp3 outn3 cm
eopm1 outp cm inp inn 5k
eopm2 cm outn inp inn 5k
rin1 inp cm 50meg
rin2 inn cm 50meg
rout1 outp outp1 5
rout2 outn outn1 5
d1 outp1 m1 dmod
v1 m1 0 0.8
d2 outn1 n1 dmod
v2 n1 0 0.8
r1 outp1 outp2 3
r2 outn1 outn2 3
r3 outp2 outp3 3
r4 outn2 outn3 3
d3 m2 outp2 dmod
v3 m2 0 2.3
d4 n2 outn2 dmod
```

```
v2 n2 0 2 3
.model dmod d
.ends
*********************************************************************
*ideal multiplexer for refn=2.05 and refp=1.05
*Daghan GOKDEL
*28/09/2006
*********************************************************************
.subckt idmux inp inn out
vch ch 0 .5
vcm cm 0 1.5
*inp=+-, outp=+-1
Ecomp out cm value='sign(v(ch),v(inp)-v(inn))'
.ends
*********************************************************************
*ideal switch
.subckt idswt vin vout phi
g1 vin vout VCR pwl(1) phi 0 0,100g 3.3,1m
.ends
*********************************************************************
*multiplexer
.subckt mux 1 2 clk1 clk2 out 100 0 101
m1 1 clk1 out 101 modn w=wmin l=0.6u
m2 1 clk2 out 100 modp w=wmin l=0.6u
m3 2 clk2 out 101 modn w=wmin l=0.6u
m4 2 clk1 out 100 modp w=wmin l=0.6u
.param wmin=3.6u
.ends
*********************************************************************
```

B.2. First Order Adaptive Sigma Delta Converter

B.2.1. Top Level Code

```
**********************************************************************
*Adaptive SD converter
vin1 in1 0 sin(1.55 530m 8000)
vin2 in2 0 sin(1.55 -530m 8000)
****************SWITCHS*********************************************
xsw1 in1 2  clk idswt
xsw2 in2 4  clk idswt
xsw3 3 cm  clk idswt
xsw4 5 cm  clk idswt
xsw5 3 intp clk2 idswt
xsw6 5 intn clk2 idswt
xsw7 DAC2 2 clk2 idswt
xsw8 DAC1 4 clk2 idswt
****************CIRCUIT*********************************************
x1 intp intn intop inton cm opamp
xcomp1 bias 0 intop inton clk outp outn comparator
x3 refn refp outp outn DAC1 bias 0 0 mux
x4 refp refn outp outn DAC2 bias 0 0 mux
****************CAPACITORS******************************************
c1 2 3 1p
c2 4 5 1p
c3 intp inton 2p
c4 intn intop 2p
***************CONTROL CIRCUIT**************************************
xcontrol1 intop pcont pconti bias 0 control
xcontrol2 inton ncont nconti bias 0 control
xcontrol3 bias pconti nconti controli nor
xcontrol4 bias controli control 0 inverter
```

```
cz1 z1 z2 4p
cz3 z3 z4 4p

xs1 intp z1 control idswt
xs2 inton z2 control idswt
xs3 intn z3 control idswt
xs4 intop z4 control idswt

xs5 z1 0 controli idswt
xs6 z2 inton controli idswt
xs7 z3 0 controli idswt
xs8 z4 intop controli idswt
**********VOLTAGES*****************************************
Vclk clk 0  pulse(3.3 0 0n  1n 1n 270n 500n)
Vclk2 clk2 0  pulse(0 3.3 20n 1n 1n 230n 500n)
Vcm cm 0 1.55
Vbias bias 0 3.3
Vrefp refp 0 2.05
Vrefn refn 0 1.05
**********INCLUDES*****************************************
.include iddifopamp.sp
.include idswt.sp
.include comparator4.sp
.include mux.sp
.include modn.mod
.include modp.mod
.include inverter.sp
.include control.sp
.include control2.sp
.include nor.sp
*************************************************************
.options post nomod probe
.ic v(inton)=1.55 v(intop)=1.55
```

```
.ic v(outp)=0 v(outn)=0
.ic v(intp)=1.55 v(intn)=1.55
.tran 10n 2m*375u
.probe v(outn) v(intop) v(inton) v(in1) v(in2) v(control)
.print v(outn) v(control)
.end
```

B.2.2. Subcircuits

```
*********************************************************************
*Control circuit
*********************************************************************
.subckt control in cont conti bias gnd
x1 in m1 bias gnd schmitt_p
x2 bias m1 m2 gnd inverter
x3 in m3 bias gnd schmitt_n
x4 bias m2 m3 cont nor
x5 bias cont conti gnd inverter
.include nor.sp
.include schmitt.sp
.include inverter.sp
.include modn.mod
.include modp.mod
.ends
*********************************************************************
*inverter
*D.Gokdel 05/06/2006
.subckt inverter bias in out gnd
m1 out in bias bias modp L=0.35u W=1.4u
m2 out in gnd gnd modn L=0.35u W=10.4u
.include modn.mod
.include modp.mod
.ends
```

```
*nor gate
*D.Gokdel 05/06/2006
*******************************
.subckt nor vdd A B out
m1 p1 A vdd vdd modp W=37.8u L=2.8u
m2 out B p1 vdd modp W=37.8u L=2.8u
m3 out A 0 0 modn W=15.4u L=2.8u
m4 out B 0 0 modn W=15.4u L=2.8u
*******************************
.include modn.mod
.include modp.mod
.ends
*********************************************************************
*schmitt Triggers
.subckt schmitt_p in out vdd gnd
M1 4 in gnd gnd modn W=.7u L=.35u
M2 out in 4 gnd modn W=1.05u L=.35u
M3 out in 2 vdd modp W=4u L=.35u
M4 2 in vdd vdd modp W=4u L=.35u
M5 gnd out 2 vdd modp W=.35u L=.35u
M6 vdd out 4 gnd modn W=.35u L=.35u
.ends

.subckt schmitt_n in out vdd gnd
M1 4 in gnd gnd modn W=2.85u L=.35u
M2 out in 4 gnd modn W=1.4u L=.35u
M3 out in 2 vdd modp W=4u L=.35u
M4 2 in vdd vdd modp W=4u L=.35u
M5 gnd out 2 vdd modp W=2.8u L=.35u
M6 vdd out 4 gnd modn W=.35u L=.35u
.ends
*********************************************************************
```

B.3. Adaptive Sigma Delta Converter with Speech Like Input

```
**********************************************************************
*Adaptive SD converter with soundlike input
**********************************************************************
*for 6.25m transient analysis
vg1 g1 0 1.6
xg1 g1 g2 clkin idswt
vg2 g2 g3 sin(0 550mv 4e2)
xg3 g3 in1 clkin idswt
xg2 g1 in1 clkinN idswt
vg4 g4 0 1.5
xg4 g4 g5 clkin idswt
vg5 g5 g6 sin(0 -550mv 4e2)
xg5 g6 in2 clkin idswt
xg6 g4 in2 clkinN idswt
vclkin clkin 0 pwl(0 0 1.24999m 0 1.25m 3.3 2.5m 3.3 2.501m 0 3.749m 0 3.75m 3.3
5m 3.3 5.01m 0)
vclkinN clkinN 0 pwl(0 3.3 1.24999m 3.3 1.25m 0 2.5m 0 2.501m 3.3 3.749m 3.3
3.75m 0 5m 0 5.01m 3.3)
**********SWITCHS**********
xsw1 in1 2  clk idswt
xsw2 in2 4  clk idswt
xsw3 3 cm  clk idswt
xsw4 5 cm  clk idswt
xsw5 3 intp clk2 idswt
xsw6 5 intn clk2 idswt
xsw7 DAC2 2 clk2 idswt
xsw8 DAC1 4 clk2 idswt
**********CIRCUIT**********
x1 intp intn intop inton cm opamp
xcomp1 bias 0 intop inton clk outp outn comparator
x3 refn refp outp outn DAC1 bias 0 0 mux
```

```
x4 refp refn outp outn DAC2 bias 0 0 mux
**********CAPACITORS****************************************
c1 2 3 1p
c2 4 5 1p
c3 intp inton 2p
c4 intn intop 2p
**********CONTROL CIRCUIT*********************************
xcontrol1 intop pcont pconti bias 0 control
xcontrol2 inton ncont nconti bias 0 control
xcontrol3 bias pconti nconti controli nor
xcontrol4 bias controli control 0 inverter

cz1 z1 z2 4p
cz3 z3 z4 4p

xs1 intp z1 control idswt
xs2 inton z2 control idswt
xs3 intn z3 control idswt
xs4 intop z4 control idswt

xcontrol5 intop pcontG pcontiG bias 0 controlG
xcontrol6 inton ncontG ncontiG bias 0 controlG
xcontrol7 bias ncontiG pcontiG controlG nor

xs5 z1 0 controlG idswt
xs6 z2 inton controlG idswt
xs7 z3 0 controlG idswt
xs8 z4 intop controlG idswt
**********VOLTAGES****************************************
Vclk clk 0  pulse(3.3 0 0n  1n 1n 270n 500n)
Vclk2 clk2 0  pulse(0 3.3 20n 1n 1n 230n 500n)
Vcm cm 0 1.55
Vbias bias 0 3.3
```

```
Vrefp refp 0 2.05
Vrefn refn 0 1.05
**********INCLUDES**********************************************
.include iddifopamp.sp
.include idswt.sp
.include comparator4.sp
.include mux.sp
.include modn.mod
.include modp.mod
.include inverter.sp
.include control.sp
.include control2.sp
.include nor.sp
*******************************************************************
.options post nomod probe
.ic v(inton)=1.55 v(intop)=1.55
.ic v(outp)=0 v(outn)=0
.ic v(intp)=1.55 v(intn)=1.55
.tran 10n 6.25m
.probe v(outn) v(intop) v(inton) v(in1) v(in2) v(control)
.print v(outn) v(control)
.end
*******************************************************************
```

APPENDIX C: C CODES

C.1. Code preparing data for acquisition in Matlab

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>

int main(void)
{
FILE *data;
FILE *data2;
char tdata;
	int i=0;

	if((data=fopen("temp.txt","r"))==NULL)
	{
		printf("Cannot open temp.txt \n");
	}
		if((data2=fopen("temp2.txt","w"))==NULL)
	{
		printf("Cannot open temp2.txt \n");
	}
		i=0;
	printf("control 1");
	while (!feof(data))
	{
		fscanf(data,"%c",&tdata);
		if(tdata=='n'||tdata=='m'||tdata=='u')
			{
			printf("bulundu\n");
```

```
                if(tdata=='n')
                        {
                        fprintf(data2,"e-9");
                        }
                if(tdata=='m')
                        {
                        fprintf(data2,"e-3");
                        }
                if(tdata=='u')
                        {
                        fprintf(data2,"e-6");
                        }
                if(tdata=='f')
                        {
                        fprintf(data2,"e-15");
                        }
                if(tdata=='p')
                        {
                        fprintf(data2,"e-12");
                        }
                }
        else
                {
                fprintf(data2,"%c",tdata);
                }
                i++;
        }
        fclose(data);fclose(data2);
}
```

REFERENCES

1. Aziz, P. M., H. V. Sorensen and J. V. D. Spiegel, "An Overview of Sigma-Delta Converters", *IEEE Signal Processing Magazine*, Vol. 13, No.1, pp. 61-84, January 1996.

2. Inose, H. and Y. Yasuda, "A Unity Bit Coding Method by Negative Feedback", *Proceedings of IEEE*, Vol. 51, No.11, pp. 1524-1535, November 1963.

3. Jespers, P. G., *Integrated Converters: D to A and A to D Architectures, Analysis and Simulation,* Oxford University Press, New York, 2001.

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

5. Brigati, S., F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto and F. Maloberti, "Modeling Sigma-Delta Modulator Non-idealities in SIMULINK", *ISCAS'99*, Orlando, USA, 30 May-2 June 1999, Vol. 2, pp. 384-387, 1999.

6. Sun, X. and K.R. Laker, "Adaptive Integrator-Output Bounding (AIB) for Second Order Sigma-Delta ADC", *IEEE International Conference On Signal Processing Proceedings*, Beijing, China, August 21-25 2000, Vol. 1, pp. 631-634, 2000.

7. Farell, R. and O. Freely, "Bounding the Integrator Outputs of Second-Order Sigma-Delta Modulators" *IEEE Transactions on Circuits and Systems*, Vol. 45, pp. 691-702, 1998.

8. Shim J.H., I. Park and B. Kim, "A third order Sigma-Delta modulator in 0.18 μm CMOS with calibrated mixed-mode integrators" *IEEE Journal of Solid State Circuits*, Vol. 40, No.4, pp. 2329-2329, April 2005.

9. Saglamdemir, M. O., O. Yetik, S. Talay and G. Dundar, "A Coefficient and Architecture Selection Tool for SD Modulators Considering Component Non-Idealities", *GLSVLSI'07*, Stresa-Lago Maggiore, Italy, March 2007.

10. Turgay, K. *Design of Second Order SD with Feedforward Path,* Senior Project, Bogazici Univesity, June 2005.

11. Meriç, A., *Variable Gain Sigma Delta Analog to Digital Converter Design*, Senior Project, Bogazici University, June 2005.

12. Chiang, J. S., T. H. Chang and P. C. Chou, "A Low Distortion and Swing Suppression Sigma-Delta Modulator with Extended Dynamic Range", *IEEE Asia Pacific Conference on ASIC*, Taipei, Taiwan, August 6-8, pp.9-12, 2002.

13. Aldajani, M. A., A.H. Sayed, "Stability and Performance Analysis of an Adaptive Sigma Delta Modulator", *IEEE Transactions on Circuits and Systems*, Vol.48, No.3, pp. 233-244 March 2001.

14. Sun, X. and K. R. Laker, "Cascaded SD ADC with Adaptive Interstage Coupling", *ISCAS 2000*, Geneva, Italy, May 28-31 2000, Vol.5, pp. 621-625, 2000.

15. Kwon, S. and F. Maloberti, "Opamp Swing Reduction in Sigma Delta Modulators", *ISCAS 2004*, Vancouver, Canada, May 23-26 2004, Vol.1, pp. 525-528, 2004.

16. V. Quiquempoix, P. Deval, J. Markus and G. C. Temes, "A Low Power 22 bit Incremental ADC", *IEEE Journal of Solid State Circuits*, Vol.41, No.7, pp. 1562-1571, July 2006.

17. S. Talay, "High Level Modeling of Sigma Delta A/D Converters", Guided Reseach, Bogazici University, June 2002.

18. Malcovati, P., S. Brigati, F. Francesconi, F. Maloberti, P. Cusinato and A. Baschirotto, “Behavioral Modeling of Switched-Capacitor Sigma-Delta Modulators”, *IEEE Transactions on Circuits and Systems*, Vol. 50, Issue 3, March 2005.

19. Sun, X. and K. R. Laker, “A New Design for Cascaded Sigma Delta Modulators”, *ISCAS 2001*, Sydney, Australia, May 6-9 2001, Vol. 4, pp. 104-107, 2001.

20. Sun, X. and K. R. Laker, “Tonal Behavior Analysis of an Adaptive Second-Order Sigma-Delta Modulator”, *ISCAS 2002*, Scottsdale Arizona, USA, May 26-29 2002, Vol. 4, pp. 277-280, 2002.

21. Shim, J. H., I. Park and B. Kim, “Hybrid Sigma Delta Modulators with Adaptive Calibration”, IEEE Transactions of Circuits and Systems, Vol. 52, No.5, pp. 885-893, May 2005.

22. Aldajani, M. A., A H. Sayed, “SNR Performance of an Adaptive Sigma Delta Modulator”, *ISCAS 2001*, Sydney, Australia, May 6-9 2001, Vol. 1, pp. 392-395, 2001.

23. Aldajani, M. A., A H. Sayed, “An Adaptive Structure for Sigma Delta Modulation with Improved Dynamic Range”, *MWSCAS 2000*, Lansing Michigan, USA, August 8-11 2000, Vol.1, pp. 390-394, 2000.

24. Yu, J., M. B. Sandler and R. E. Hawken, “Adaptive Quantization for One-Bit Sigma Delta Modulation”, *IEEE Proceedings of Circuits, Devices and Systems*, Vol.139. No.1, pp. 39-44, February 1992.

25. Ramesh, M. C. and K. S. Chao, “Sigma Delta Analog to Digital Converters with Adaptive Quantization”, *MWSCAS 1997*, Sacramento, USA, August 3-6, Vol.3, pp. 22-25, 1997.

26. Cusinato, P., F. Pasolini, F. Stefani and A. Baschirotto, "Saturation Recovery Technique for High Order Bandpass Switched Capacitor SD Modulators", *ICECS 2001*, Malta, September 2-5 2001, Vol.2, pp. 1085-1088, 2001.

27. Williams, L. A. and B. A. Wooley, "A Third Order Sigma Delta Modulator with Extended Dynamic Range", *IEEE Journal of Solid State*, Vol. 29, No.3, pp. 193-202, March 1994.

28. Ferguson, P. F., A. Genesan and R.W. Adams, "One Bit High Order Sigma Delta A/D Converters", *ISCAS 1990*, New Orleans, USA, May 1-3 1990, Vol.2, pp. 890-893, 1990.

29. Fujcik, L., J. Haze, R. Varba and T. Mougel, "Modeling and Design of Novel Architecture of Multibit Switched Capacitor Sigma Delta Converter with Two Step Quantization Process", *ICN/ICONS/MSL 2006,* April 23-29 2006, pp. 186-187, 2006.

30. Motamed, M., "The Double Loop Sigma Delta Modulator with Unstable Filter Dynamic: Stability Analysis and Ton Behavior", *IEEE Transactions on Circuits and Systems*, Vol. 43, pp. 549-559, 1996.

31. Wang, H., "A Geometric view of Sigma Delta Modulations", ", *IEEE Transactions on Circuits and Systems*, Vol. 39, pp. 402-405, 1992.