DESIGN AND FPGA IMPLEMENTATION OF AREA OPTIMIZED 16-BIT DELTA-SIGMA DIGITAL TO ANALOG CONVERTER FOR PORTABLE AUDIO APPLICATIONS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Semiconductor Devices & VLSI Technology)

By

SHIVAKUMAR POTTAVATHINI

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2009

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being reported in this dissertation report, entitled **"Design and FPGA Implementation of Area Optimized 16-Bit Delta-Sigma Digital to Analog Converter for Portable Audio Applications"**, is being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Semiconductor Devices and VLSI Technology**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work, carried out from June 2008 to June 2009, under guidance and supervision of **Dr. S. Dasgupta**, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date : 30/06/09

Place : Roorkee

P Shivakumar

Shivakumar Pottavathini

CERTIFICATE

This is to certify that the statement made by the candidate is correct to best of my knowledge and belief.

Date: 3rd/06/09

Place: Roorkee

Dr. S. Dasgupta Assistant Professor,
Department of Electronics & Computer,
Indian Institute of Technology Roorkee

ACKNOWLEDGEMENT

At the outset, I express my heartfelt gratitude to Dr. S. Dasgupta, Assistant Professor, Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for his valuable guidance, support, encouragement and immense help. I consider myself extremely fortunate for getting the opportunity to learn and work under his able supervision. I have deep sense of admiration for his innate goodness and inexhaustible enthusiasm. It helped me to work in the right direction to attain desired objectives. Working under his guidance will always remain a cherished experience in my memory and I will adore it throughout my life.

My sincere thanks are also due to all the Research Scholars of my VLSI Design Lab without whom availing of lab facilities during non-working hours wouldn't have been possible. I also extend my sincere thanks to rest of the faculty in the Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for the technical knowhow and analytical abilities they have imbibed in us which have helped me in dealing with the problems I encountered during the project.

I am greatly indebted to my family and all my friends, who have willingly applied themselves to the task of helping me with ample morale support and valuable suggestions. I would also like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work. Finally, I would like to thank God, whose gracious presence strengthened my faith which helped me get through troublesome situations during my work.

Shivakumar Pottavathini

M. Tech. (SDVT)

ABSTRACT

The evolution of Digital Signal Processing (DSP) has revolutionized the field of communications and signal processing which allows high speed processing of input data. However, it is required to convert these processed digital data into analog domain for real world applications. Digital to Analog Converters (DACs), which are a mixture of analog and digital circuits are employed to perform this task of digital to analog conversion. In-order to fully utilize the advantages provided by DSP, it is very much required to implement these DACs completely using digital circuits, which is possible only by using Delta-Sigma Converters.

This thesis presents the design of an area optimized, oversampled, 16-bit Delta-Sigma Digital to Analog Converter (DAC) for portable audio applications. A new style of architecture without multipliers is presented to save hardware and improve system throughput. CIC interpolator and Error Feedback modulator structures have been employed to optimize and achieve low area, thus leading to low power consumption. Most of the implementation is done using digital blocks which greatly reduce the effect of process variations in ultra deep sub-micron technology.

With recent improvements in the density of Field Programmable Gate Array (FPGA) devices, it is becoming possible to integrate systems with an increasingly higher level of functionality. The entire Delta-Sigma DAC is initially modeled and simulated in MATLAB. The Designed DAC is implemented on Spartan 3E (xc3s500e-4fg320) FPGA kit employing the latest Xilinx System Generator model based design methodology. The results obtained were satisfactory and showed reduction in area in terms of the number of slices used on FPGA.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
ADC	Analog to Digital Converter
ABE	Audio Back End
AFE	Audio Front End
CD	Compact Disc
DSP	Digital Signal Processing
DAC	Digital to Analog Converter
ENOB	Effective Number of Bits
LSB	Lowest Significant Bit
MSB	Most Significant Bit
INL	Integrated Non-Linearity
NTF	Noise Transfer Function
OSR	Over Sampling Ratio
SNR	Signal to Noise Ratio
SNR	Signal to Noise Ratio
STF	Signal Transfer Function

Chapter 1

INTRODUCTION

1.1. Introduction

In modern electronic circuits, a lot of signal processing takes place in the digital domain. Digital circuits are preferable in many applications due to their robustness and accuracy. Less effort is required to implement digital systems which are generally not susceptible to the drift or tuning problems commonly found in analog circuitry. People interact with machines only by means of analog signals. Also, most of the signals in the physical world are fundamentally analog in nature (heat, light, light waves, etc.). Consequently, Analog-to-Digital (A/D) and Digital-to-Analog (D/A) Converters are inevitable as interfaces between the digital worlds of integrated systems and the analog world of speakers, video screens, keyboards, etc. Most modern audio signals are stored in digital form (for example CDs and MP3) and in order to be heard through speakers they must be converted into an analog form. Digital to Analog Converters (DACs) thus find a place in every CD player, digital music player and PC sound card available in the market.

A complete digital audio chain can be described by the block diagram shown in Fig.1.1, The operation flow can be explained as follows. An instrument like guitar emits sound which is then captured by a microphone and converted into an electric signal. This signal is then amplified and filtered by Audio Front End (AFE) equipment. It is then converted to digital data by an Analog to Digital Converter (ADC) and stored onto a Compact Disc (CD) or other digital audio medium. During playback, the medium is read and output data is transformed back to an analog signal by a DAC, then amplified and filtered by Audio Back End (ABE) equipment and finally converted to real world audio sound by the loudspeaker.

Such digital audio systems as the one shown in Fig.1.1, place high performance demands on Digital to Analog Converters (DACs) because of wide dynamic range requirements [2,3]. A resolution of 16-bits is required for CD quality sound. CD audio is a high quality audio standard that supports 16-bit linear Pulse Code Modulation (PCM) with sampling rates as high as 44.1 KHz [4].

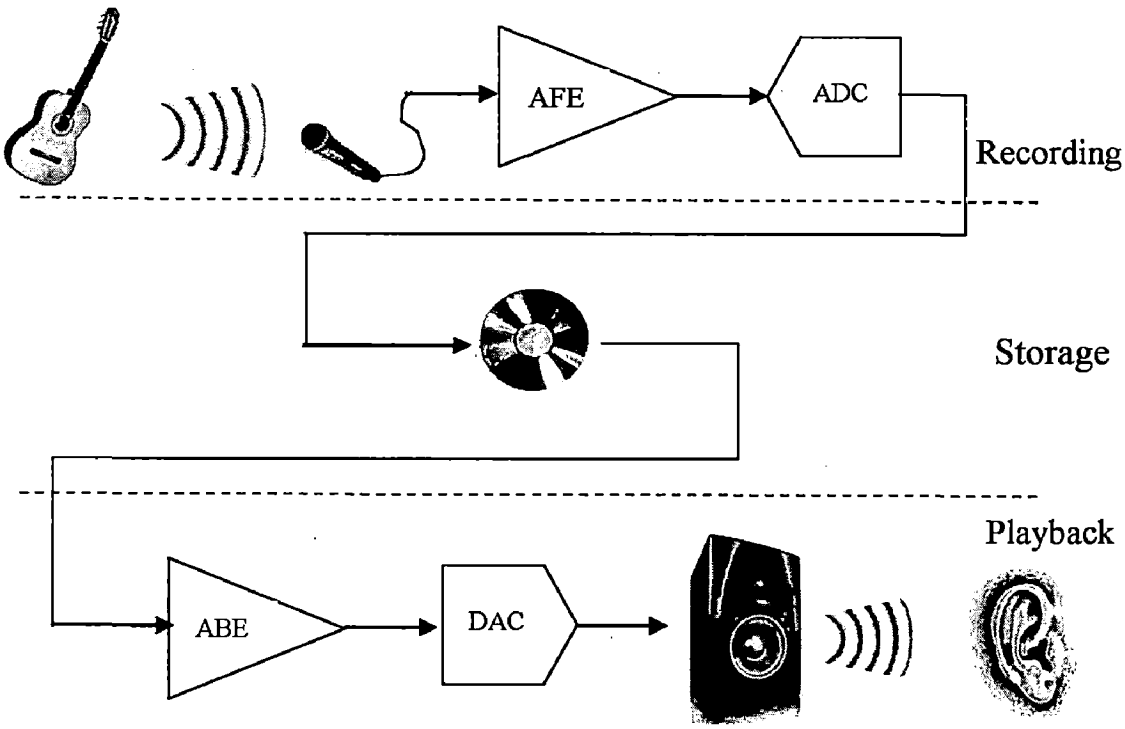


Figure 1: Digital Audio Recording and Playback Chain [1]

In most cases, the linearity and accuracy of Nyquist-rate converters is determined by the matching accuracy of the analog components (resistors, current sources or capacitors) used in the implementation. For example, in an N-bit resistor-string DAC, the resistors must have a relative matching error of less than 2^{-N} to guarantee an Integral Non-Linearity (INL) less than 0.5 Least Significant Bit (LSB). For16 bits of accuracy using 1.2 V reference voltage, the voltage corresponding to the permissible one-half least-significant-bit (LSB) error is then $2^{-17} \times 1.2$ V, or about 9.15μV. This is a very small value; it is the voltage generated by about a dozen electrons stored in a 0.1pF capacitor. It is also comparable to the thermal noise present at the input of a typical MOS op-amp. Thus, practical conditions restrict the matching accuracy to about 0.02%, resulting in maximum Effective Number of Bits (ENOB) to be about 12, for such converters [5-7]. In such a situation, Delta-Sigma DAC's structure is often preferred because of their advantages such as high resolution, insensitivity to circuit non-idealities and low cost.

Delta-Sigma DAC is primarily based on the two principles, namely, Oversampling and Noise Shaping. The expanded operating bandwidth afforded by oversampling allows the shaping of quantization noise such that most of quantization noise power falls outside the signal band while the input signal is unaltered. The result is a converter with high Signal to Noise Ratio (SNR) within the frequencies of interest. A key point with Delta-Sigma

DAC is that its implementation is predominantly digital, its architecture transfers the complexity from analog to the digital domain, reduces the stringent requirements on analog circuitry found in conventional Nyquist-rate DACs. In Previous works [7-16], different kinds of Delta-Sigma DACs have been presented of which most of them were implemented as Application Specific Integrated Circuits (ASICs). In this thesis, a 16-Bit Delta-Sigma DAC has been designed for portable audio applications where special attention is paid to area optimization thereby also reducing the power requirements.

1.2. Thesis Contribution

Area and power are critical considerations for portable systems. There is thus every need of developing a DAC which consumes less area and power which can be implemented with low complexity.

The thesis contributes a new architecture for 16-bit Delta-Sigma DAC for audio applications without using multipliers. Two facts have been used here to decrease the design complexity. Firstly, human ear works as a crude Low Pass Filter (LPF) and can't hear above 20 KHz. Secondly, the noise shaper itself works as a LPF for signal and as High Pass Filter (HPF) for the noise.

These points allowed the use of only CIC filters at the input end for interpolation. For noise shaper, a 'single-bit second order digital noise-shaper looped with error feedback' has been chosen. In order to cope with the present day competition scenario, and low time to market requirements, the implementation of the designed DAC was done on a Field Programmable Gate Array (FPGA) employing the latest Xilinx System Generator model based design methodology.

1.3. Thesis Organization

The six chapters of this thesis have been organized in the following structure:

Chapter 2 briefly reviews the fundamentals of Delta-Sigma Audio DACs. Nyquist sampling and oversampling are compared. The technique of Noise shaping is described in detail. Finally, an overview of the designed audio DAC structure is presented which gives an idea of the complete system.

In Chapter 3, Principle of Interpolation Operation is discussed. Implementation of Interpolation is presented. Basic CIC Interpolation Filter and Pipelined Structure is

Presented .The simulation results performed in MATLAB of the designed Interpolation are also provided.

Chapter 4 discusses the different architectures for the implementation of the modulator are discussed. Design oh the Second Order 1- bit Delta-Sigma Modulator using Error Feedback architecture is presented and also MATLB simulation results of the designed modulator are

In chapter 5, FPGA architecture and its design flow using System Generation tool are discussed and the FPGA implementation results of the designed DAC are presented .

Chapter 6 concludes this thesis.

Chapter 2

A REVIEW ON DELTA-SIGMA DATA CONVERTERS

Digital-to-Analog converters are one of the major blocks of systems which require interfacing of digital components to the real-world analog signals. Digital-to-Analog converters transform digital binary representation (input word) into a corresponding analog representation. This is illustrated by the black box view of the DAC as shown below in Fig.2.1.

The Digital Signal Processing (DSP) block produces a sequence of bits representing a digital signal which is converted to analog levels by means of a Discrete-Time-to-Continuous-Time (DTCT) converter. The resulting analog representation of the signal is then low pass filtered in a reconstruction filter to remove the out-of-band noise components. The next subsection discusses different available topologies of DTCT Converters.

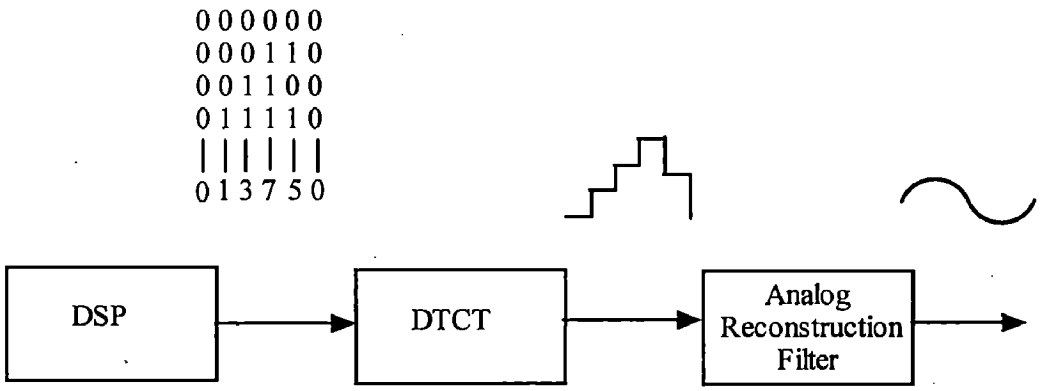


Figure 2.1: Block Diagram of a Digital to Analog Converter .

2.1 Different Topologies of DTCT Conversion

The core element of the Digital to Analog Converter shown in Fig.2.1 is the Discrete-Time-to-Continuous-Time (DTCT) converter, which takes input as digital values, which are sampled at discrete instants of time, then, converted into analog levels that are continuous in time. These analog levels can be generated using voltage, charge, or current, resulting in three different topologies for DTCT conversion: voltage division, charge redistribution, and current steering. In the following sub-sections, we discuss these topologies in detail.

2.1.1. Voltage Division

A given reference voltage V_{ref} can be divided into N equal segments using a ladder composed of N identical resistors, where N is typically a power of 2. Fig.2.2(a) shows a 2-bit example of such a converter. An m-bit DTCT converter would require a ladder with 2^m resistors, manifesting the exponential growth of the number of resistors as the function of resolution [17]. The analog output corresponding to the digital input is generated by connecting one of the voltage levels created by the ladder to the output through a simple switch network.

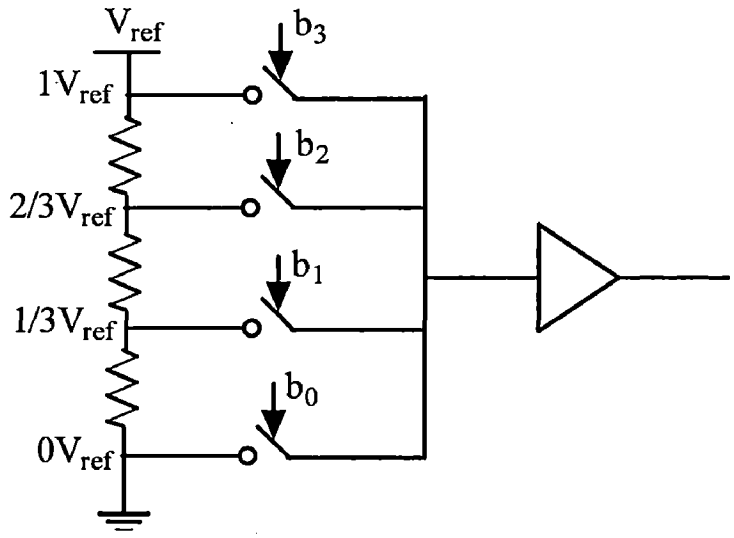


Figure 2.2: DTCT Voltage Division Topology

A disadvantage of voltage division is the need for linear resistors, which are not available in many standard CMOS process technologies and requirement of large die area when implementing a large number of output levels. Moreover, a resistive ladder dissipates considerable amounts of static power.

2.1.2 Charge Redistribution

Another DTCT conversion technique is charge redistribution using an array of identical capacitors that share the same top plate, as illustrated in Fig.2.3. The bottom plates of the capacitors are switched between ground and a reference voltage, V_{ref}, according to the input thermometer code. In other words, each capacitor can inject a charge equal to CV_{ref} into the output node, thereby producing an output voltage proportional to the height of the input thermometer code [17].

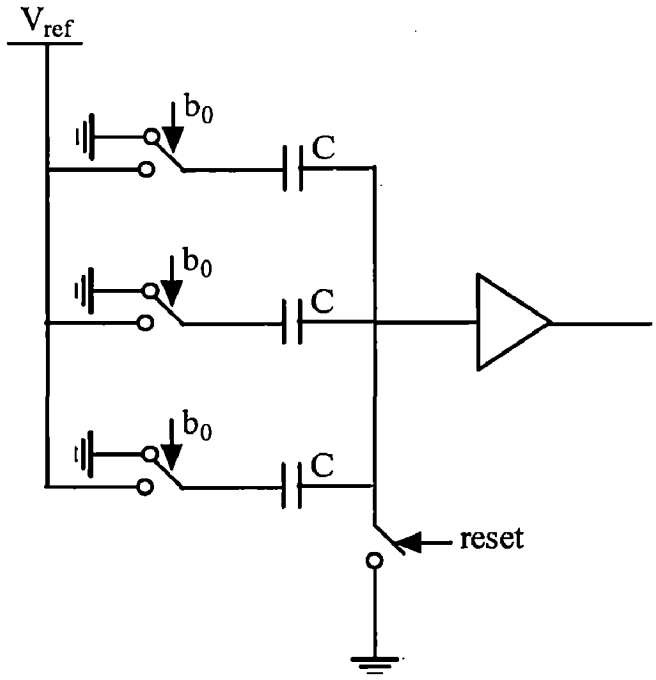


Figure 2.3: DTCT Charge Redistribution Topology

A disadvantage of charge redistribution is the need for linear capacitors, which can consume considerable area when generating a large number of output levels.

2.1.3. Current Steering

In a current-steering DTCT converter, an array of identical current sources are used to generate analog output levels on a resistive load, as shown in Fig.2.4 for a 2-bit example. The analog output voltage is a function of the number of current cells that are switched on according to the digital input code [17].

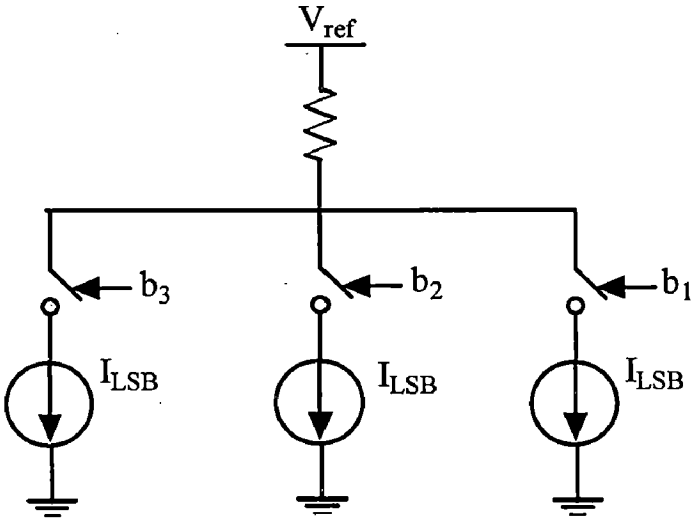


Figure 2.4: DTCT Current Steering Topology

A disadvantage of the current-steering architecture is the amount of static power dissipation in the current sources.

Therefore Nyquist Rate architectures present significant challenges when implementing higher resolution Digital-to-Analog Converters.

An alternative architecture for Digital-to-Analog Conversion that is frequently used in digital audio applications (where the target resolution exceeds 16 bits) is the oversampling Delta-Sigma modulation which has been followed in this thesis. In the next section, the Delta-Sigma Data Conversion system is introduced.

2.2. Delta-Sigma Digital-to-Analog Converter

Fig.2.5 shows the basic block diagram of an oversampled Delta-Sigma DAC. The input to this converter is an N-bit digital signal sampled at the Nyquist rate $2f_0$. This signal is oversampled at f_s by the oversampling ratio R ($f_s = Rf_0$). Digital interpolator is used to remove spectral images of the baseband signal centred at multiples of $2f_0$. The resulting oversampled N-bit data is then reduced into K-bit (K<N) code by means of a digital Delta-Sigma modulator. The K-bit output of the digital modulator is converted into an analog signal using a DTCT converter. Finally, the analog output is low pass filtered to remove the out-of-band quantization noise and remaining spectral images at multiples of f_s.

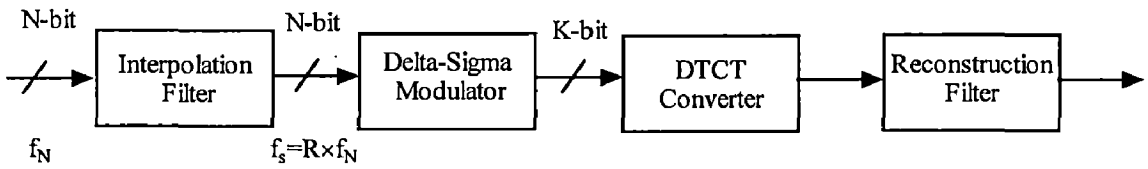


Figure 2.5: Block Diagram of a Delta-Sigma D/A Conversion System

Two fundamental concepts are necessary for proper understanding of the advantages gained by using Delta-Sigma DACs. They are: Oversampling and Noise shaping. Inorder to appreciate the advantages of oversampling, a brief review on Nyquist rate sampling is essential which is presented below.

2.2.1. Nyquist-Rate Sampling

According to the sampling theorem, as long as an analog signal is sampled at a frequency f_s that is at least twice the signal bandwidth f_0, which is called Nyquist rate, the signal can be completely represented by and recoverable from the sampled values [5]. A DAC

working at Nyquist rate, called Nyquist rate DAC, needs an analog reconstruction filter with a very small transition band because the image of the signal is just next to the signal itself. In addition to the stringent requirements on the smoothing filter, Nyquist-rate DACs can only achieve moderate accuracy because the matching problems among the analog components directly limit the precision of these DACs.

The quantization error e(n) is the difference between the input and output values. It is calculated by modelling the quantizer as shown in Fig.2.6. Digital signal x(n) contains the quantization errors due to its finite N-bit resolution. The output signal, y(n) is equal to the closest quantized value of x(n).

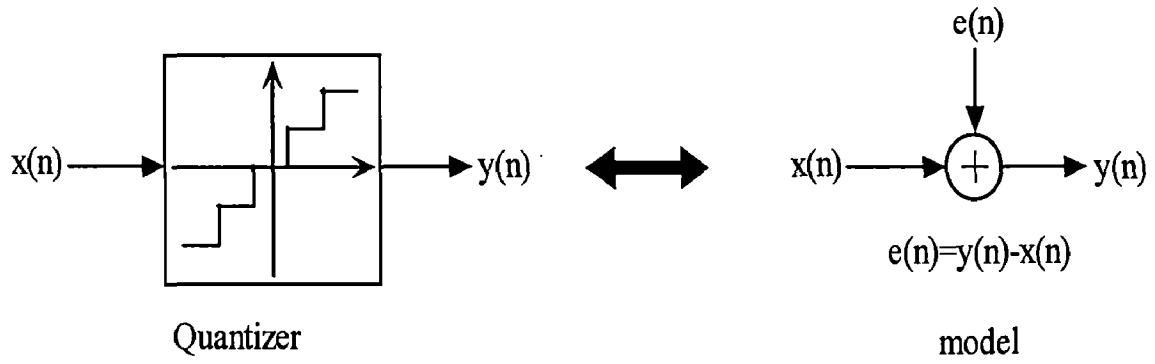


Figure 2.6: Quantizer and its Linear Model

If x(n) is very active, e(n) can be approximated as an independent random number uniformly distributed between $\pm\Delta/2$, where Δ equals the difference between two adjacent quantization levels. The quantization noise power can be found out to be equal to $\Delta^2/12$ which is independent of the sampling frequency, f_s. Also the power spectral density of noise, $S_e(f)$ is assumed to be white (i.e., a constant over frequency) and all its power is limited within $\pm f_s/2$. Mathematically it can be written as follows [7]:

$$\int_{-f_s/2}^{f_s/2} S_{e(x)} = \int_{-f_s/2}^{f_s/2} k_x^2 df = k_x f_s = \frac{\Delta^2}{12} \qquad [2.1]$$

Where k_x is equivalent to

$$k_x = \left(\frac{\Delta}{\sqrt{12}}\right)\sqrt{\frac{1}{f_s}} \qquad [2.2]$$

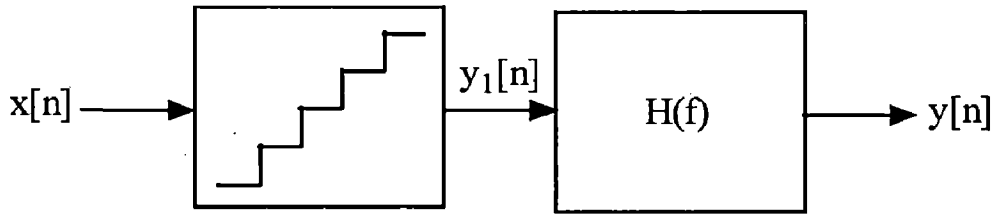
In the following sub-section, the advantage due to oversampling is described.

2.2.2. Oversampling Advantage

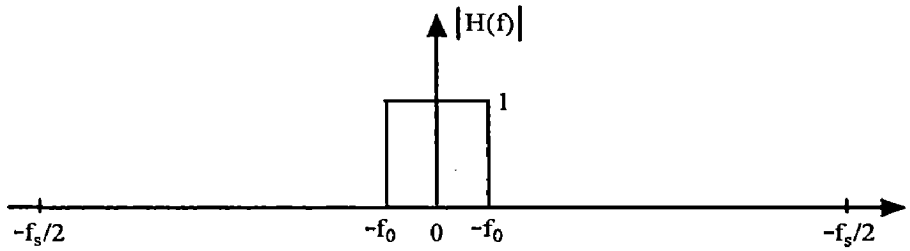
Oversampling is defined at the point where $f_s > 2f_0$, where f_0 is the input signal bandwidth and f_s is the sampling rate. The Over Sampling Ratio (OSR) can be given as:

$$OSR = \frac{f_s}{2f_0} \qquad [2.3]$$

A typical oversampling system is shown in Fig.2.7 (a). After quantization, since the signals of interest are all below f_0, $y_1[n]$ is filtered by H(f) to create the signal y[n]. This filter shown in Fig.2.7 (b) eliminates quantization noise (together with any other signals) greater than f_0.



(a)



(b)

Figure 2.7: (a) Oversampling System

(b) The Brick Wall Response of the Filter to Remove Out of Band Quantization Noise

The power spectral density, $S_e(f)$, of the quantization noise for Nyquist rate sampling with rate f_{s1} and oversampling f_{s2} are shown in Fig.2.8. For Nyquist rate sampling with signal band, $f_0=f_{s1}/2$, all the quantization noise power, represented by the area of the tall shaded rectangle, occur across the signal bandwidth. In the oversampled case, the same noise power, represented by the area of the unshaded rectangle has been spread over a bandwidth equal to the sampling frequency f_{s2} which is much greater than the signal bandwidth f_0. Only a relatively small fraction of the total noise power falls in the signal band $[-f_0, f_0]$.

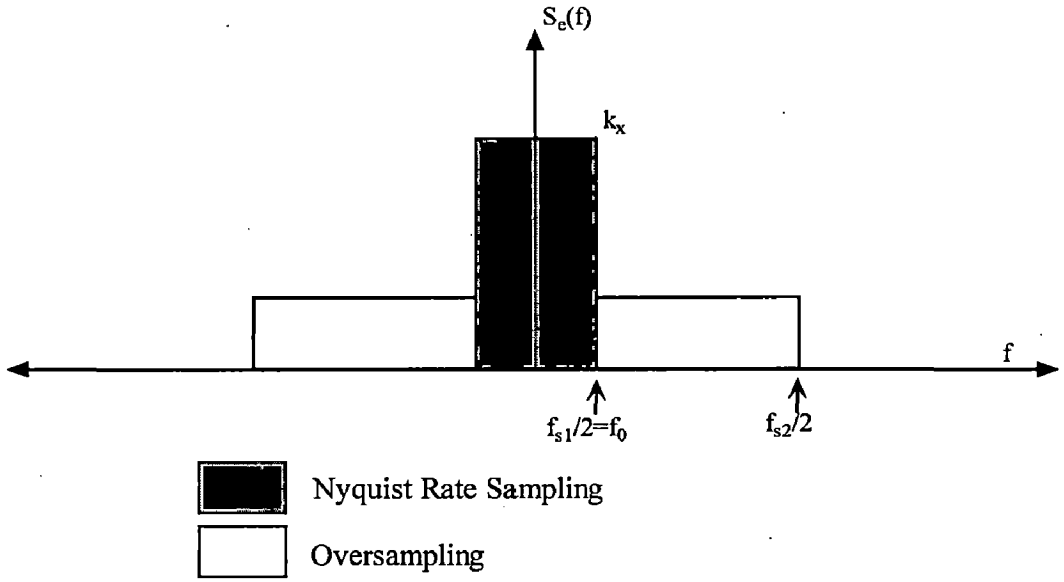


Figure 2.8: Quantization Noise Power Spectral Density for Nyquist Rate and Oversampling Conversion

If we assume the input signal to be to be a sinusoidal nature, its maximum peak value without clipping is at $2^N(\Delta/2)$. For this maximum sinusoidal wave, the signal power, P_s, is given by

$$P_s = \left(\frac{\Delta 2^N}{2\sqrt{2}}\right)^2 = \frac{\Delta^2 2^{2N}}{8} \qquad [2.4]$$

The power of the input signal within $y_2(n)$ remains the same as before since the signal's frequency content is below f_0. However, the quantization noise power is reduced to

$$P_e = \int_{-f_0}^{f_0} S_e(f)|H(f)|^2\, df$$

$$= \int_{-f_0}^{f_0} k_x\, df = \frac{\Delta^2}{12}\frac{2f_0}{f_s} = \frac{\Delta^2}{12}\left(\frac{1}{OSR}\right) \qquad [2.5]$$

Therefore, doubling OSR (i.e., sampling twice the rate) decreases the quantization noise power by one-half or, equivalently, 3 dB (or equivalently 0.5 bits).

The maximum Signal to Noise Ratio of maximum sinusoidal power to the quantization noise in the signal is given by:

$$SNR = 10\log\left(\frac{P_s}{P_e}\right) = 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log(OSR) \qquad [2.6]$$

The above equation can be simplified as:

$$SNR = 6.02N + 1.76 + 10\log(OSR) \qquad [2.7]$$

The first term of (2.7) is the SNR due to N-bit quantizer while the OSR term is the SNR enhancement obtained from oversampling. Here from (2.7), it clear that oversampling gives an SNR improvement of 3 dB/octave or, equivalently, 0.5 bits/octave.

In contrast to Nyquist sampling, the cost for oversampling is that the digital circuits in the system are running at a higher speed which is practical with the development of VLSI technology. In today's applications, oversampling technique is widely used to achieve high resolution in data converters.

2.3. Noise-Shaping Modulators

As shown in Fig.2.8, in the oversampling case, the noise in the signal band is low. For a given OSR, the in-band noise can be further reduced by using delta-sigma modulation. The modulator is designed to perform filtering functions. In the simplest case, the input signal should be low-pass filtered through the modulator. The truncation noise introduced by the modulator should be high pass filtered. A general noise shaped delta-sigma modulator and its linear model are shown in Fig.2.9 as given in [7]. The name of delta sigma comes from the fact that the difference (delta) between the input signal u(z) and the delayed output $z^{-1}y(z)$ is added (sigma) together during the operation.

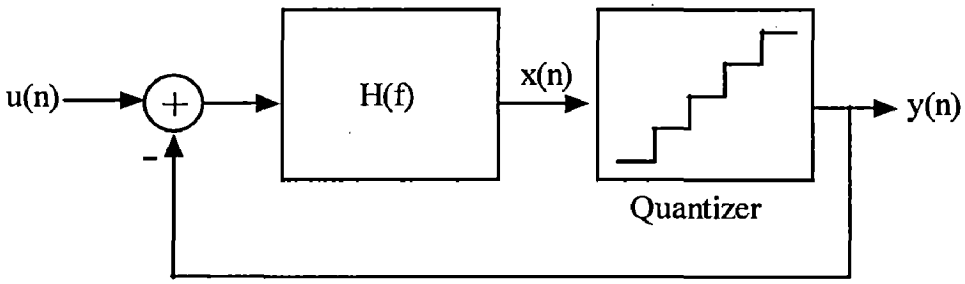


Figure 2.9 (a): A General Delta-Sigma Modulator

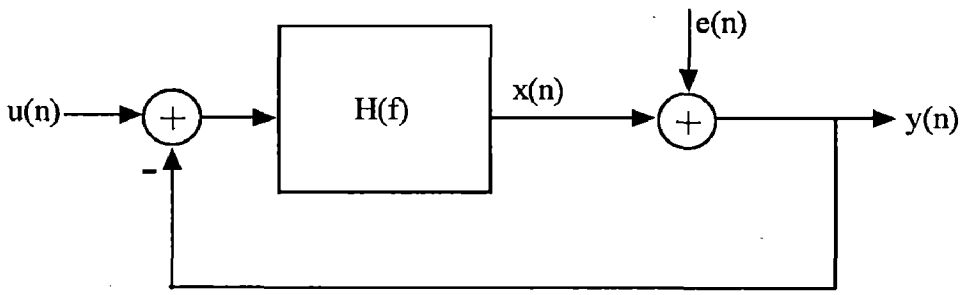


Figure 2.9 (b): Linear Model of the Modulator Showing Injecting Quantization Noise

From the Fig.2.9, the output of the first order Delta-Sigma modulator is given by

$$Y(z)=\frac{H(z)}{1+H(z)}U(z)+\frac{1}{1+H(z)}E(z) \qquad [2.8]$$

This can be written in the form

$$Y(z)=STF(z)U(z)+NTF(z)E(z) \qquad [2.9]$$

Here, STF(z) is called Signal Transfer Function and NTF(z) is called Noise Transfer Function.

In order to shape the quantization noise, H(z) is chosen in a way that its magnitude is large from 0 to f_0 (i.e over frequency band of interest). With this choice, the signal transfer function (STF) will be unity in the required frequency band of interest. The Noise Transfer Function (NTF) will be approximately zero over the same band. Thus, the quantization noise is reduced over the frequency band of interest and the signal is not affected at all. High frequency noise is not reduced by feedback as there is little loop gain at high frequencies. However, analog filtering can remove the out-of-band quantization noise with very less effect on the desired signal.

In order to realize first order noise shaping, the noise transfer function, NTF(z) should have a zero at dc (i.e., z = 1) so that quantization noise is high-pass filtered. Since the zeros of the noise transfer function NTF(z) are equal to the poles of H(z), we obtain first order noise shaping by letting H(z) be a discrete time integrator (i.e., have a pole at z =1). The transfer function of the accumulator is given by,

$$H(z)=\frac{z}{z-1} \qquad [2.10]$$

A block diagram for such a choice is shown in Fig.2.10.

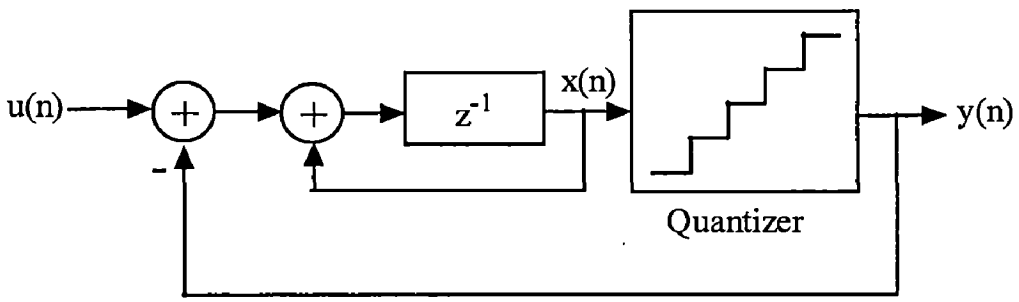


Figure 2.10: First Order Noise Shaping Modulator

The Signal Transfer Function (STF) is given by,

$$STF(z) = \frac{Y(z)}{X(Z)} = \frac{\frac{1}{Z-1}}{1+\frac{1}{Z-1}} = Z^{-1} \qquad [2.11]$$

We can observe that the Signal Transfer Function is simply a delay and the Noise Transfer Function (NTF) is given by,

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1+\frac{1}{Z-1}} = 1 - Z^{-1} \qquad [2.12]$$

The Noise Transfer Function is a first order discrete time differentiator (i.e. a high pass filter). This function can be viewed as present sample minus previous sample. The magnitude spectrum of a first order Delta-Sigma Noise Transfer function (NTF) is shown in Fig.2.11.

NTF (z) contains a zero at $z = 1$, i.e., at DC frequency on the unit circle of the Z-plane, therefore NTF provides the zero gain or infinite attenuation at DC frequency. From Fig.2.11, it is clear that there is a large attenuation at lower frequencies and relative amplification at higher frequencies. For comparison, the oversampled NTF, which has unity gain, is also shown in Figure. 2.7. The vertical bar demonstrates the extent of the signal band, f_0, where $f_0 = 0.05fs$. Quantization noise to the left of the bar that contributes to the finite resolution of the modulator is greatly attenuated while noise to the right of the bar is not attenuated as much or is actually amplified. However, noise to the right of the bar can mostly be removed with analog reconstruction filter.

The magnitude of the noise transfer function is found by putting $z = e^{jwT} = e^{j2\pi f/fs}$.

$$NTF(f) = 1 - e^{-j2\pi\frac{f}{f_s}} = 2je^{-j\pi\frac{f}{f_s}} \frac{e^{j\pi\frac{f}{f_s}} - e^{-j\pi\frac{f}{f_s}}}{2j}$$

$$= 2je^{-j\pi\frac{f}{f_s}} \sin\left(\frac{\pi f}{f_s}\right) \qquad [2.13]$$

Taking the magnitude on both sides,

$$|NTF(f)| = \left|2\sin\left(\frac{\pi f}{f_s}\right)\right| \qquad [2.14]$$

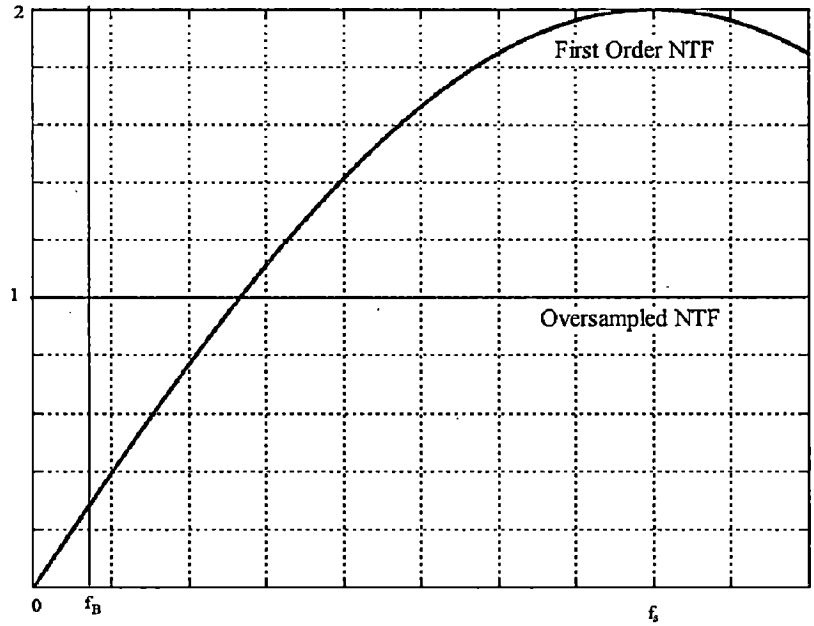


Figure 2.11: First Order Noise Transfer Function

The spectral density of the quantization noise at the output of the modulator is given by,

$$S_{ve}(f) = S_e(f)NTF(f) \qquad [2.15]$$

Now, the quantization noise power over the frequency band from 0 to f_0 is given by

$$P_e = \int_{-f_0}^{f_0} S_e^2 |NTF(f)|^2 df = \int_{-f_0}^{f_0} \left(\frac{\Delta^2}{12}\right)\frac{1}{f_s}\left[2\sin\left(\frac{\pi f}{f_s}\right)\right] df \qquad [2.16]$$

Making approximations, $f_0 << f_s$ (i.e. OSR >> 1), so that we can approximate $\sin(\pi f/f_s)$ to be $\pi f/f_s$, we have

$$P_e = \left(\frac{\Delta^2}{12}\right)\left(\frac{\pi^2}{3}\right)\left(\frac{2f_0}{f_s}\right)^3 = \frac{\Delta^2\pi^2}{36}\left(\frac{1}{OSR}\right)^3 \qquad [2.17]$$

Assuming maximum signal power to be the same as that obtained as before in (2.4), the maximum SNR for this case can be given by,

$$SNR = 10\log\left(\frac{P_s}{P_e}\right) = 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log\left[\frac{3}{2\pi}(OSR)^3\right] \qquad [2.18]$$

or equivalently,

$$SNR = 6.02N + 1.76 - 5.17 + 30\log(OSR) \qquad [2.19]$$

From the above equations, it is quite evident that doubling the OSR gives an improvement for a first order modulator of 9 dB or, equivalently a gain of 1.5 bits/octave. This result can be compared to the 0.5 bits/octave when no noise shaping is used. The next sub-section briefly discusses higher order noise shaping modulators.

2.4. Higher Order Modulators

The modulator shown in Fig.2.12 realizes second order noise-shaping (i.e., the noise transfer function, NTF(z) is a second order high-pass function). For this modulator, the signal transfer is given by

$$STF(f) = z^{-1} \qquad [2.20]$$

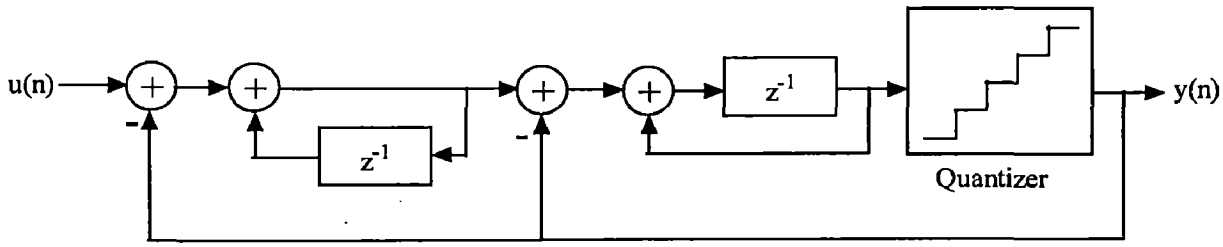


Figure 2.12: Second Order Delta-Sigma Modulator

and the noise transfer function is given by

$$NTF(f) = \left(1 - z^{-1}\right)^2 \qquad [2.21]$$

also the magnitude of the transfer function can be given by

$$\left|NTF(f)\right| = \left[2\sin\left(\frac{\pi f}{f_s}\right)\right]^2 \qquad [2.22]$$

resulting in the quantization noise power over the frequency band of interest is given by

$$P_e = \frac{\Delta^4 \pi^4}{60}\left(\frac{1}{OSR}\right)^5 \qquad [2.23]$$

also if we again assume that signal power is obtained by the eq. 2.4.

$$SNR = 10\log\left(\frac{P_s}{P_e}\right) = 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log\left[\frac{5}{2\pi}(OSR)^5\right] \quad [2.24]$$

or equivalently

$$SNR = 6.02N + 1.76 + 1.76 - 12.9 + 50\log(OSR)(dB) \quad [2.25]$$

One interesting point to note here is that doubling the OSR improves the SNR for a second order modulator by 15 dB or equivalently, a gain of 2.5 bits/octave.

2.4.1. Lth-Order Modulator

An Lth order modulator based on a straight forward extension of the first order Delta-Sigma realizes a STF given by z^{-1} and a NTF given by = $(1-z^{-1})^L$. This contains L zeros at $z = 1$ or at DC frequency on the unit circle. The ideal in-band SNR achieved by an Lth order modulator is given

$$SNR_{\max} = 6.02N + 1.76 - 10\log\left(\frac{\pi^{2L}}{2L+1}\right) + (20L+10)\log(OSR)(dB) \quad [2.26]$$

Thus, for every doubling of the oversampling ratio, this modulator provides an extra (6L+ 3) dB of SNR, or an extra (L+ 1/2) bits of resolution [7].

In Fig.2.13, the power spectral density (PSD) for the 1st, 2nd, 3rd order modulators are shown. We see that for lower frequencies the attenuation of noise is higher for higher modulators orders.

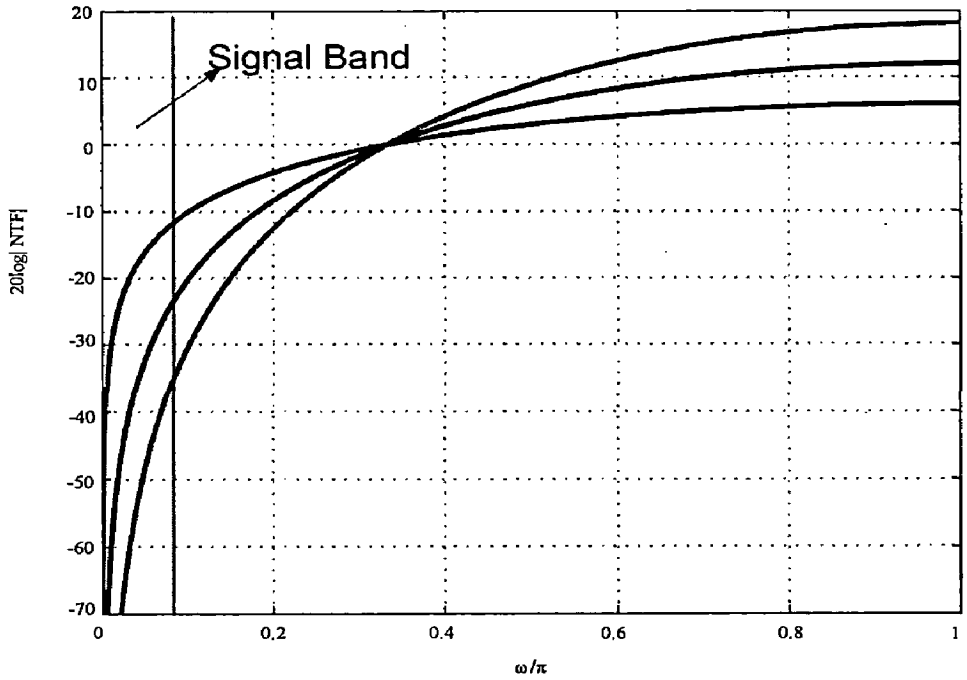


Figure 2.13: Noise Transfer Functions of 1st, 2nd and 3rd Order Modulators

2.5. Designed System Overview

Fig.2.13 shows the basic block diagram of designed audio Delta-Sigma DAC. The DAC accepts 16-bit digital input data at sampling rates of 44.1 KHz. The interpolation ratio of the interpolator can be configured to 176X.

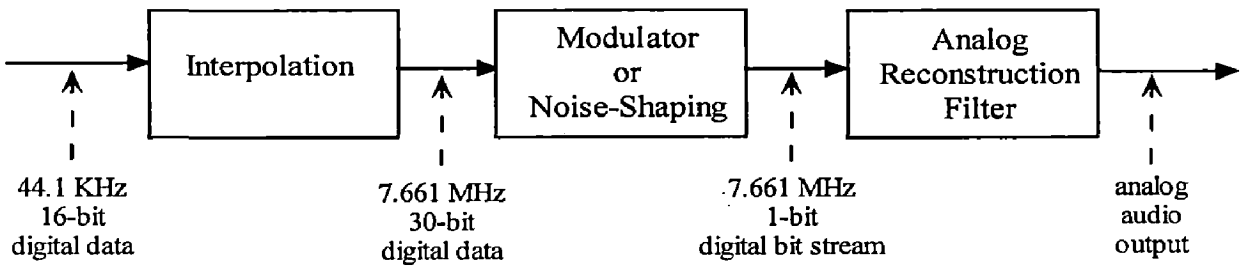


Figure 2.13: Block Diagram of the Audio DAC

As the employed modulator is of one-bit resolution, there is no requirement of the DTCT block which is mostly analog in nature as discussed in section 2.1. This helps in reducing the area required by the DAC. The modulator used in this design is of 2nd order. Higher order modulators are not used due to the fact that more aggressive noise shaping increases the gain of the noise shaping high pass filter at higher frequencies, resulting in a decrease in the stable input range of the modulator thus affecting the stability of the system [18]. The OSR value is calculated as per (2.26) which is found to be equal to 176.

For 44.1 kHz input signals, the interpolator gives the output data rate of 7.7616 MHz by setting the interpolation ratio as 176X. The main function of the digital-to-analog conversion is performed by the delta-sigma modulator, which produces one-bit output and spectrally shapes the quantization noise to high frequencies. The modulator is clocked at 7.7616 MHz. The output of the Delta-Sigma modulator which is 1bit is given to the 1bit DAC which can be easily implemented without requirement of complicated analog circuits [19]. This stage is followed by an analog reconstruction filter. The output of this filter gives the equivalent analog signal for the given digital input. We now move on the design of interpolation filter in the next chapter.

Chapter 3

INTERPOLATION FILTER DESIGN

As shown in Figure 2.10, the first block in a delta sigma audio DAC is an interpolation filter followed by a delta sigma modulator. In this chapter, the design of a digital interpolator with a 176 up-sampling factor is described.

The objective of an interpolator is to provide oversampling to the input of ΔΣ modulator. Its main function is to multiply the input sampling rate by a factor known as the Over Sampling Ratio (OSR). By using an interpolator, we can estimate the intermediate points between every two input sampling instants. For example, if the interpolator is a 64X interpolation filter, then the input to ΔΣ modulator is increased by 64 times.

The interpolation filter used in the delta sigma DAC is built by an up-sampler and a low-pass filter as shown in Figure 3.1[20].

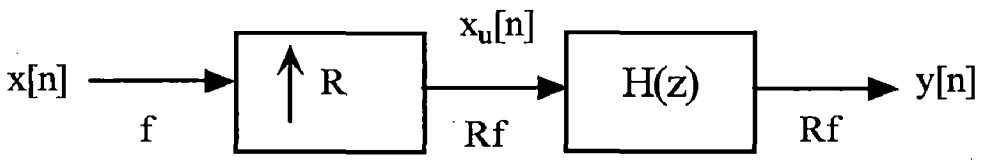


Figure 3.1: Block Diagram of the Interpolation Filter

3.1 Up-Sampling

An up-sampler with an up-sampling factor R, develops an output sequence $x_u[n]$with a sampling rate that is R times larger than that of the input sequence x[n]. The up-sampling operation is implemented by inserting R-1, equidistant zero-valued samples between two consecutive samples of the input sequence x[n] according to the relation

$$x_u[n] = \begin{cases} x\left[\frac{n}{R}\right], & n = 0, \pm R, \pm 2R \\ 0, & otherwise \end{cases} \qquad [3.1]$$

The up-sampling operation is shown in the Figure.3.2,

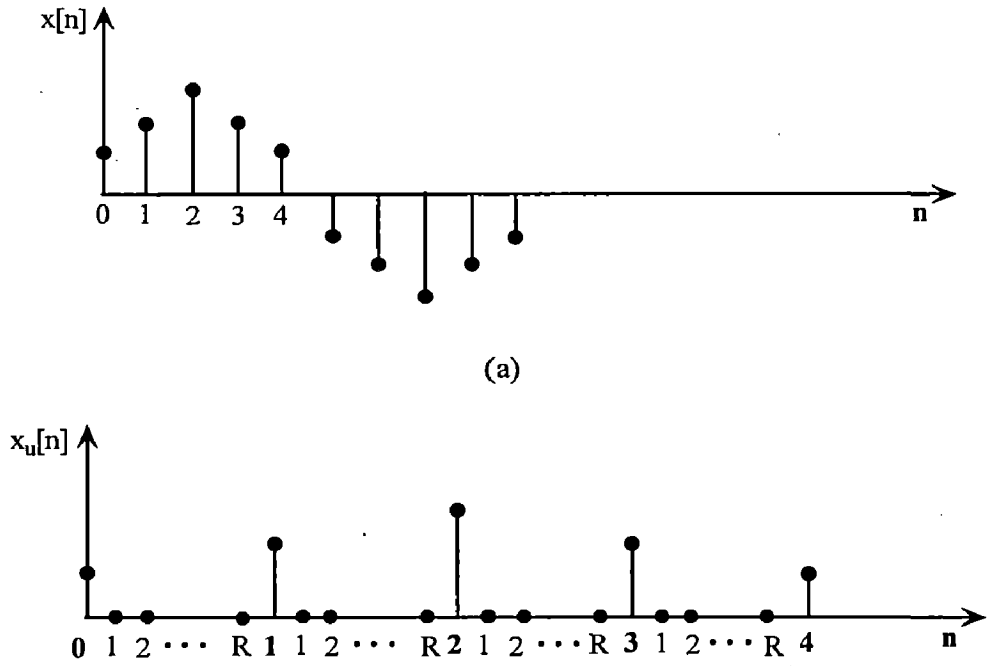


Figure 3.2: Up-Sampling Operation (a) Input Signal (b) R Times Up-Sampled Signal

3.1.1 Spectrum of the Up-Sampled Signal

In terms of the z-transform, the input-output relation is given by

$$X_u(z) = \sum_{n=-\infty}^{\infty} x_u[n]z^{-n} = \sum_{\substack{n=-\infty \\ \text{when } n \text{ is integer} \\ \text{multiples of } R}}^{\infty} x_u[n/R]z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x[m]z^{-Rm} = X(z^R) \qquad [3.2]$$

Putting $z = e^{j\omega}$,in the above equation becomes

$$X_u(e^{j\omega}) = X(e^{j\omega R}) \qquad [3.3]$$

Fig.3.3 shows the spectrum of the original signal and up-sampled signal

3.1.2 Imaging Effects

As shown in the Fig.3.3 (b), a factor of R sampling rate expansion leads to the R-fold repetition of $X(e^{j\omega})$. This process is called imaging because there occurs additional images of the input spectrum. Thus, a spectrum $X(e^{j\omega})$ bandlimited to the low-frequency region does not look like a low-frequency spectrum after up-sampling,

because of the insertion of zero valued samples between the nonzero samples of $x_u[n]$. The low-pass filter following the up-sampler removes the R-1 images of $x_u[n]$ in effect the low-pass filter fills in the zero valued samples in $x_u[n]$ with interpolation sample values.

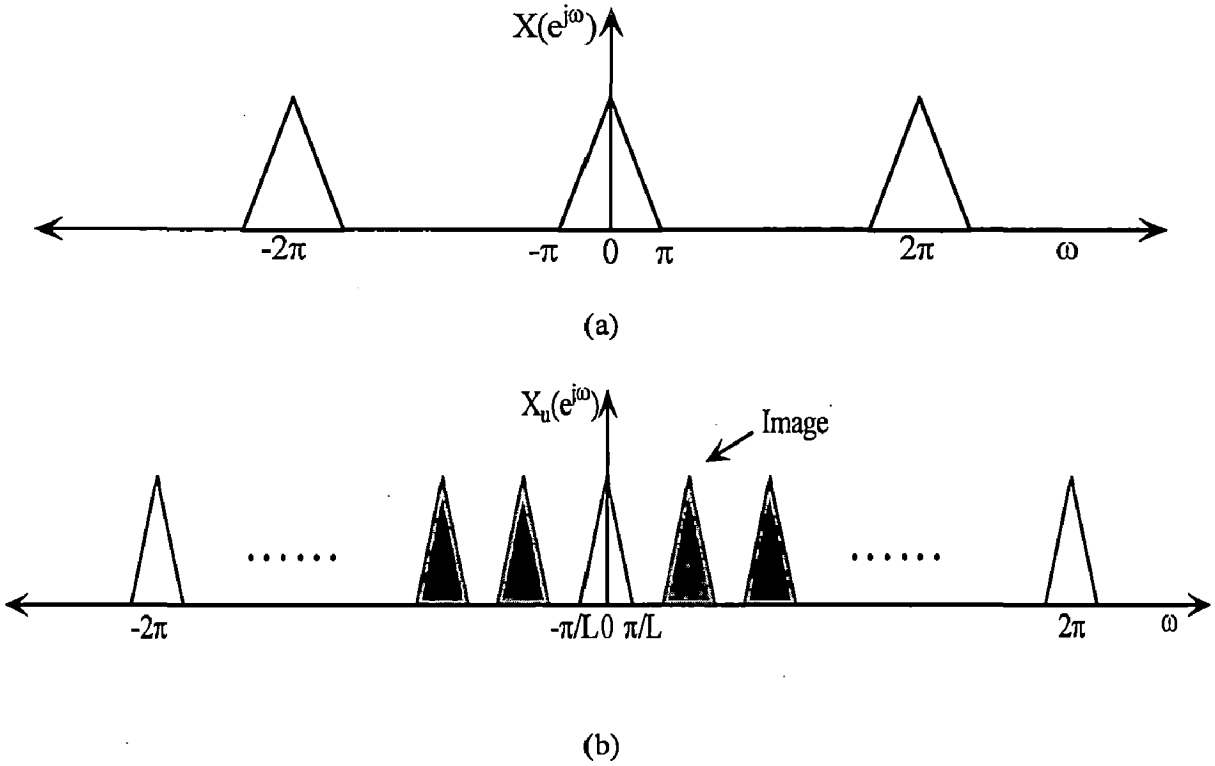


Figure 3.3: Effects of Up-Sampling in the Frequency Domain

(a) Input Signal, **(b)** Output Signal

3.2 Conventional Interpolation Filter Structure

A low-pass filter H(z), called the interpolation filter as shown in Fig.3.1, is required to remove the unwanted images in the spectra of the up-sampled signal $x_u[n]$. Fig.3.4 shows the traditional architecture for the implementation of an interpolation filer. A direct form FIR filter of length M is implemented after the up-sampler. This filter has M multipliers, M two-input accumulators, and 3M storage registers. Assuming the sampling rate of the input signal is f_s then $M \times f_s$ multiplications are excuted every second [21].

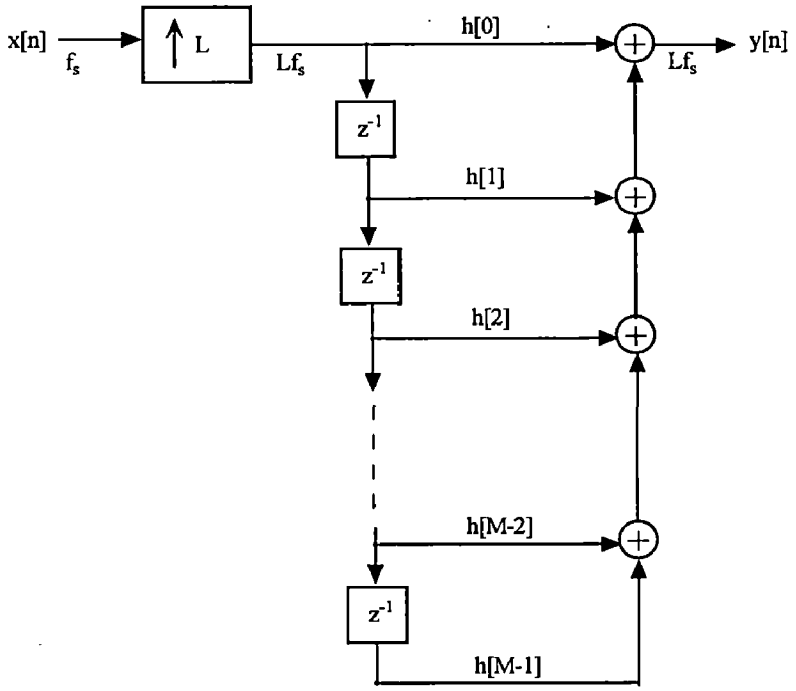


Figure 3.4: Direct Form Structure for Interpolation Filter

Improvement can be made to this structure for example, a poly phase structure could be used to increase the throughput [22], but there is a cost of greater implementation.

The designed interpolation filter is needs to perform a 1:176 up conversion and out filtering with a cut-off frequency of 20 KHz. Since our DAC is to be used in mobile systems, the design needs to be implemented carefully, with design targets that include small hardware and low power consumption. The structure used in design consists of a two stage Cascaded Integrator Comb (CIC) filter as shown in Fig.3.5.

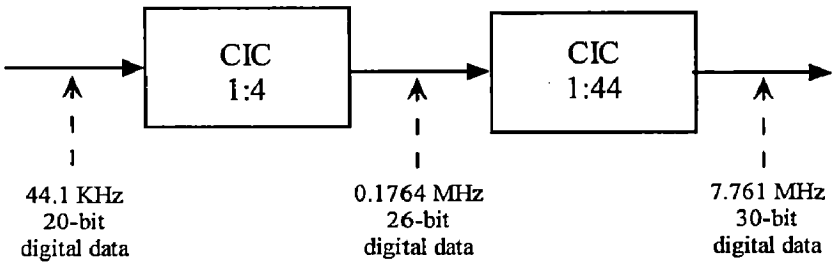


Figure 3.5: Block Diagram of Interpolation Filter

In the next subsection, we first describe the basic CIC filter and later present its modified structure as described in [23,24], which increases the throughput of the filter.

3.3 CIC Interpolation Filter

The choice of CIC filter in this implementation has been inspired from the fact that the number of adders and registers required are very less as compared to the conventional structures given in [20,21]. This can be verified by studying the basic structure of CIC Interpolation filter introduced by Hogenauer in [25]. It is shown in Fig.3.6.

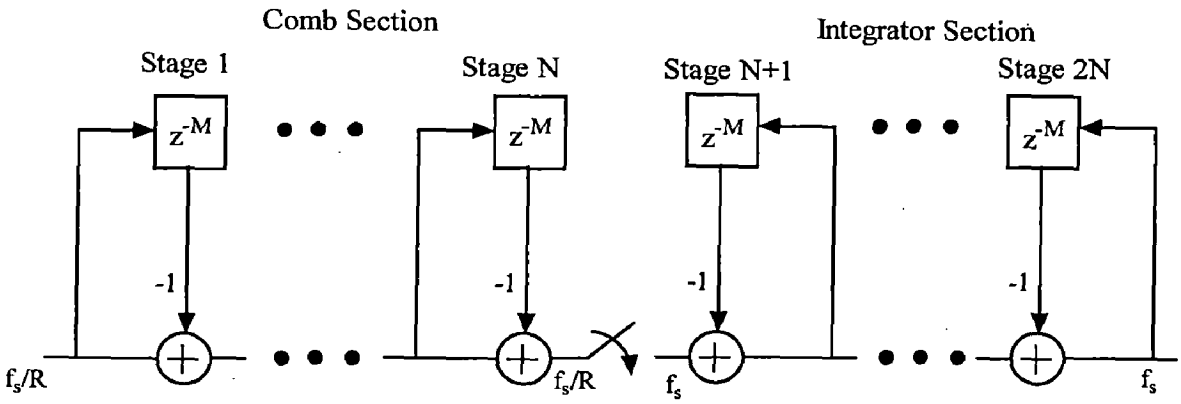


Figure 3.6: CIC Interpolation Filter

In the above figure, the comb section operates at the input sampling rate F_s/R where 'R' is the integer rate change factor. This section consists of 'N' comb stages with a differential delay of 'M' samples per stage. The differential delay is a filter design parameter used to control the filter's frequency response. In practice, M is usually equal to 1 or 2. The system function for a single comb stage referenced to the high sampling rate is given as:

$$H_c(z) = 1 - z^{-RM} \qquad [3.4]$$

The integrator section of CIC filters consist of N ideal digital integrator stages operating at the high sampling rate, i.e., F_s. Each stage is implemented as a one-pole filter with a unity feedback coefficient. The system function for a single integrator is given by:

$$H_I(z) = \frac{1}{1 - z^{-1}} \qquad [3.5]$$

There is a rate change switch between the two filter sections. For interpolation, the switch causes a rate increase by a factor of R by inserting R – 1 zero valued samples between consecutive samples of the comb section output. It follows from (3.4) and (3.5)

that the system function for the composite CIC filter referenced to the high sampling rate, F_s, is:

$$H(z) = H_I{}^N(z) H_C{}^N(z) = \frac{\left(1-z^{-RM}\right)^N}{\left(1-z^{-1}\right)^N} \qquad [3.6]$$

$$= \left[\frac{\left(1-z^{-1}\right)\left(1+z^{-1}+.....+z^{-(RM-1)}\right)}{\left(1-z^{-1}\right)}\right]^N$$

$$= \left[\sum_{k=0}^{RM-1} z^{-k}\right]^N \qquad [3.7]$$

As can be observed from the CIC structure, an adder exists for each stage, two registers are needed to store the two input signals for each stage, and 2 additional registers store the integrators' and combs' outputs. Thus, only 2N adders and 4N+2 registers are needed to implement the above transfer function [3.6].

If this same transfer function is realized by conventional direct-form I structure, then N cascaded low pass filters would be required. Thus, a total of N*R*M adders and N*(2*R*M+1) registers are required. Comparing the CIC implementation with the traditional implementation for large R, we find that, many adders and registers are hence saved. For this reason, the CIC interpolation filter falls under efficient class of digital filters. Like CIC filters, some of the filters described in [21] do not require multipliers, however, these filters are restricted to a rate change factor of two and have limited attenuation in the imaging bands.

In order to attain the highest possible throughput, a pipelined structure is used in this work. As can be seen from Fig.3.6 the CIC structure has two parts, namely comb and integrator. The integrator part of this structure can be modified by introducing a pipeline between the two adders [23]. This is shown in Fig.3.7.

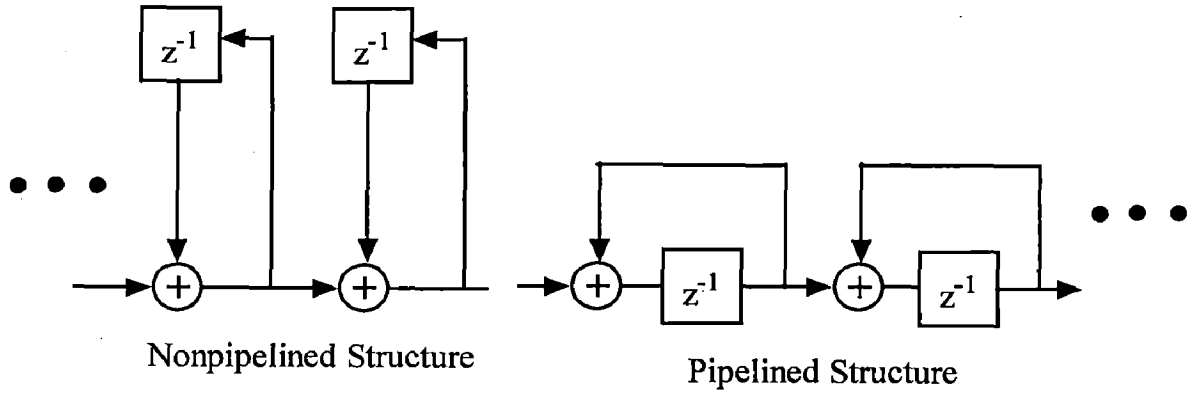


Figure 3.7: Pipelined and Nonpipelined Structures

As can be seen from the above figure, in the pipelined structure, no additional pipeline registers are used, so that the hardware requirement is the same as in the non-pipelined structure. The delay between registers determines the maximum clock rate. If the delay is short, a higher maximum clock can be used; if the delay between registers is long, only a slower frequency clock signal can be used. In the non-pipelined structure, there are two adders between two adjacent registers, while in the pipelined structure there is only one. Thus, a pipelined integrator section is preferred for higher throughput. In the comb section, a pipelined structure is not necessary as the clock rate is much slower than the integrator section.

3.3.1 Frequency response characteristics

The frequency response is obtained by evaluating Equation 3.6 at:

$$z = e^{j2\pi f/R} \qquad [3.8]$$

where f is the frequency relative to the input sampling frequency interpolation filter. The magnitude frequency response is given by:

$$|H(f)| = \left[\frac{\sin(\pi M f)}{\sin\left(\frac{\pi f}{R}\right)}\right]^N \qquad [3.9]$$

In the design process of a CIC filter implementation, the parameters R, M and N are selected to provide adequate pass-band characteristics over the frequency from zero to a predetermined cut off frequency f_c. Here the pass-band frequency range is the bandwidth of the audio signal range i.e 0 to 20 KHz. From the equation 3.9 it is clear that the nulls

of the magnitude response occur at integer multiples of f =1/(M). Thus, the differential delay M, and Oversampling ratio R are used to control the placement of nulls. For CIC interpolation filter imaging occurs in the regions around these same nulls. Specifically, these imaging bands are

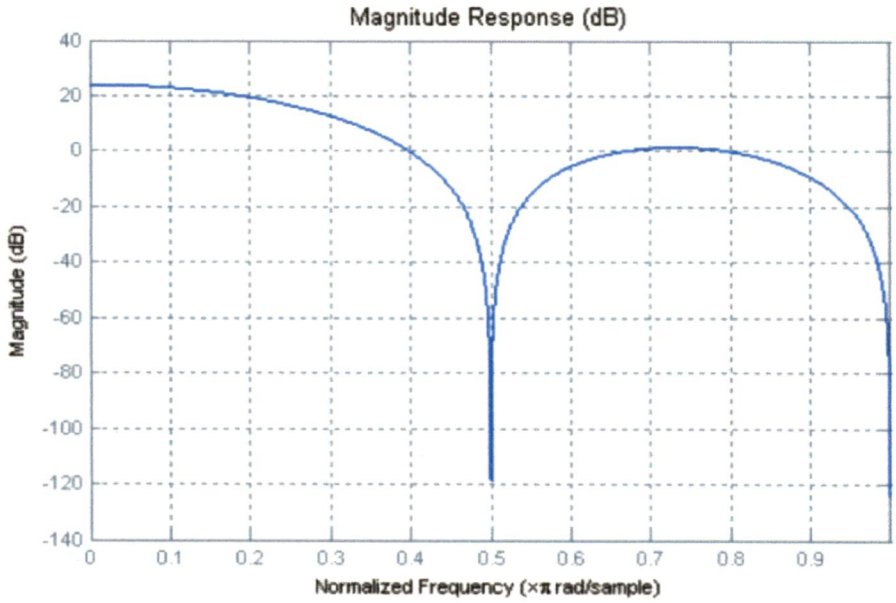
$$(i - f_c) \leq f \leq (i + f_c) \qquad [3.8]$$

for $f \leq \frac{1}{2}$ and i = 1, 2, ..., $\lfloor R/2 \rfloor$ where $\lfloor x \rfloor$ is the largest integer not greater than x.

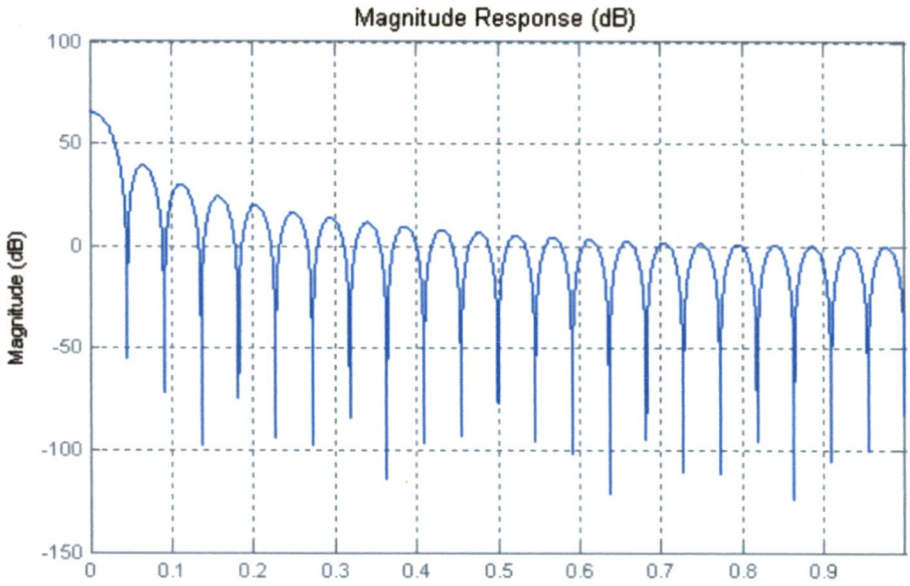
The rate change parameter R can also be used to control the frequency response of the CIC filter. In essence, increasing the rate change increases the length of the cascaded unit-amplitude, rectangular window of length R*M. This results in an increase in attenuation and decrease of the width of the response side lobes. The number of stages parameter, N can also be used to affect the CIC filter magnitude response. This effect can be understood from the fundamental concept of a cascaded of N filtering stages each with an impulse response of a unit-amplitude, rectangular window. The larger the number of cascaded stages, the more attenuated the magnitude response side lobes become.

3.3.2 Simulation Results

The 176 of interpolation of the input signal was done in two stages. First stage increases the input sample rate 4 times and then the input signal again 44 times oversampled by the second stage. As frequency response of the CIC filter is not flat in the pass band, the parameters of the CIC filter were chosen such that there was less attenuation at the edge of audio signal band by. By using MATLAB simulations we have selected R=4, M=1 and N=2 for first stage CIC filer, and R=44, M=1 and N=2 for second stage CIC filter. The frequency responses of first stage and second stage CIC are shown in the Fig.5 (a) and Fig.5 (b) respectively.



(a)



(b)

Figure 3.8: (a) Magnitude Response of CIC 4x Interpolator

(b) Magnitude Response of CIC 44x Interpolator

Fig.3.8 shows a drop in the magnitude response at the pass band edge. This leads to a slight distortion at the high end of the audio signal band. This is tolerable since the hearing sensitivity of the human hear falls with increasing frequency and is almost negligible above 20 KHz [26, 27].

Chapter 4

Delta Sigma Modulator Design

A Delta-Sigma modulator, illustrated in Fig.2.10 is basically a negative feedback system, where the large gain at low frequencies forces the output of the loop to follow the input at those frequencies. The quantizer in the forward path truncates the number of output bits to K, which is much smaller than the number of input bits, N. The truncation introduces large quantization noise into the output of the modulator. Fortunately, this large noise has a high pass transfer function to the output. Thus, it is shaped away from the signal band, and most of the noise power resides in the extra frequency spectrum outside the signal band that is created by oversampling. Delta-Sigma modulators are also referred to as noise shapers owing to the high pass filtering (shaping) of the quantization noise.

The task in this thesis is to design an area optimized noise shaper unit for achieving quality results. Noise-shapers can be classified by the order, by the architecture (single-loop or cascaded) and by the type of the feedback signal (MSB or LSB/error-feedback) [5]. We here are interested in the classification based on feedback which is given as:

- Noise-Shaper loops with Signal Feedback
- Noise-Shaper loops with Error-Feedback

4.1 Different Architectures of Delta-Sigma Modulators

There are several different structures that can be used to build a Delta-Sigma modulator.

4.1.1 Error-Feedback Modulator

In the Error-Feedback Delta Sigma modulator the quantization error is fed back to the input of the circuit as illustrated in Fig.4.1, where Q is the quantizer that reduces the number of bits in a binary word, i.e., performs truncation.

The input signal x[n] consists of N-bits and the output signal y[n] consists of K-bits. The quantization noise introduced by the truncator can be modelled as white Gaussian noise as discussed in the section 2.2.1 it is again given in the Fig.4.2. The quantization noise consists of bits N-K.

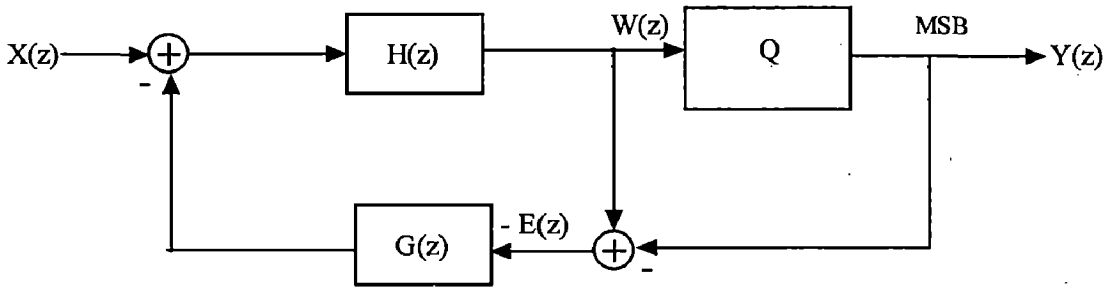


Figure 4.1: The Structure of an Error-Feedback Delta-Sigma Modulator

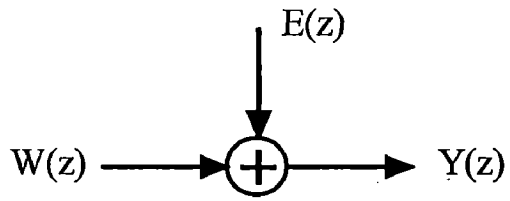


Figure 4.2: Simplified Model of the Quantizer

Therefore the quantization noise is given by

$$E(z) = W(z)-Y(z) \qquad [4.1]$$

The transfer function of the error feedback structure from the input X(z) and the quantization noise E(z), to the output Y(z) is given by

$$Y(z) = H(z)X(z) + [1 - H(z)G(z)]E(z) \qquad [4.2]$$

where H(z) and G(z) are the filters in Fig.4.1. From (4.2) the Signal Transfer Function (STF) and Noise Transfer Function of the modulator can be found to be:

$$STF(z) = H(z) \qquad [4.3]$$

and

$$NTF(z) = 1 - H(z)G(z) \qquad [4.4]$$

An important requirement on the impulse response g[n] of the filter G(z) , is

$$g(0) = 0$$

This implies that the filter G(z) must contain a delay. Otherwise the modulator is not realizable in the digital domain since the output value is depending on its own value.

4.1.2 Signal-Feedback Modulator

In the signal-feedback sigma-delta modulator the output signal is fed directly back to the input of the circuit as shown in Fig.4.3.

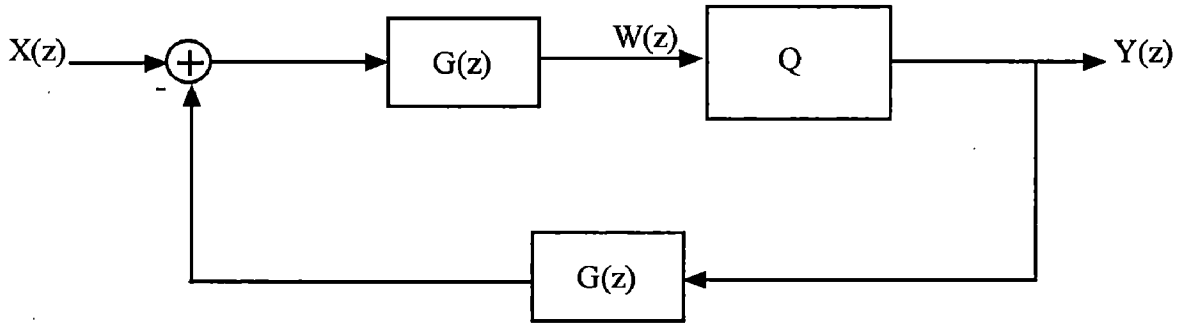


Figure 4.3: The Structure of a Signal Feedback Delta-Sigma Modulator

The transfer function of the signal feedback structure is:

$$Y(z) = \frac{H(z)X(z)}{1 + H(z)G(z)} + \frac{E(z)}{1 + H(z)G(z)} \qquad [4.5]$$

Therefore the STF and NTF of the modulator are

$$STF(z) = \frac{H(z)}{1 + H(z)G(z)} \qquad [4.6]$$

and

$$NTF(z) = \frac{1}{1 + H(z)G(z)} \qquad [4.7]$$

In this structure it is somewhat more difficult to choose G(z) and H(z) to achieve the desired noise transfer function and signal transfer function, compared to the error-feedback structure.

4.1.3 Multiple-Feedback Modulator

In the multiple-feedback sigma-delta modulator both the output and other intermediate signals are fed back to the input of the circuit, e.g. as in Fig.4.4.

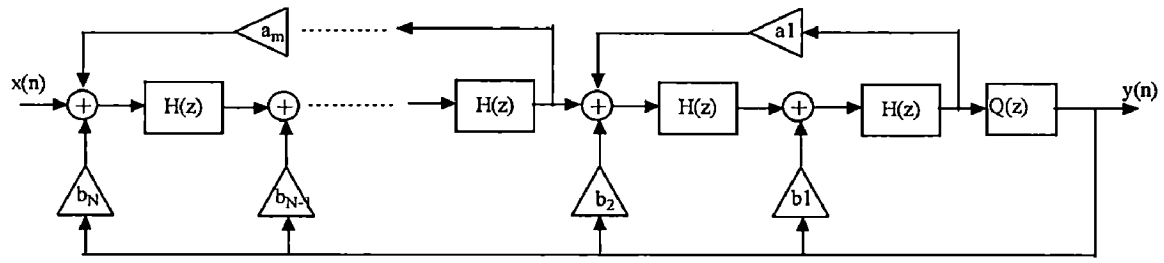


Figure 4.4: The Structure of a Multiple Feedback Delta-Sigma Modulator

The transfer function of the modulator is :

$$Y(z) = STF(z)X(z) + NTF(z)Q(z) \qquad [4.8]$$

The Signal Transfer Function (STF) and Noise Transfer Function (NTF) of the multiple feedback modulators are given by [23].

$$NTF(z) = \frac{\prod_{i=1}^{m}(1+a_i H^2)}{\prod_{i=1}^{m}(1+a_i H^2) + \sum_{i=1}^{m}\left[(b_{2i}H + b^{2i}H^2)\prod_{j=i+1}^{m}(1+a_j H^2)H^2\right]} \qquad [4.9]$$

$$STF(z) = \frac{\prod_{i=1}^{m} H^2}{\prod_{i=1}^{m}(1+a_i H^2) + \sum_{i=1}^{m}\left[(b_{2i}H + b^{2i}H^2)\prod_{j=i+1}^{m}(1+a_j H^2)H^2\right]} \qquad [4.10]$$

Where 'n' is the filter order and m is $\lfloor n/2 \rfloor$.

In this thesis Error Feedback architecture has been chosen for the design of the Delta-Sigma modulator because it leads to simple realization than the signal feedback structures.

- The difference between quantizer input and output can be done by simply taking the LSBs of a digital number; no hardware required
- The feedback FIR filter is very easily implemented as the coefficients are simple power of 2 only (discussed in the next section). An equivalent loop filter designed with conventional integrators requires several three-input adders, as each integrator must sum its input signal as well as its feedback signal to generate a new output sample.
- It does not require multiplication of multi bit words, only delays, additions, and shifts of the binary point are needed.

In the next section the design of the Delta-Sigma noise shaping loop with the Error Feedback architecture is presented.

4.2. Design of Error Feedback Noise Shaper

The desired function of the modulator is that the noise is to be high-pass filtered, so it is attenuated in the signal band, and the input signal is to be all-pass filtered or unaffected,

to make the output as close to the input as possible. The filters H(z) and G(z) of Error-Feedback structure are obtained as follows.

The Signal Transfer Function (STF) and the Noise Transfer Function (NTF) of the Error Feedback Modulator as given in (4.3) and (4.4) are respectively $STF(z) = H(z)$ $NTF(z) = 1 - H(z)G(z)$.

We have chosen H(z)=STF(z)=1 to all pass filter the input signal. If we do not truncate the signal, the output signal is equal to the input signal.

therefore NTF becomes

$$NTF(z) = 1 - G(z) \Rightarrow G(z) = 1 - NTF(z) \qquad [4.11]$$

NTF (z) is a high pass FIR filter. A general second order FIR filter is shown in Fig.4.5

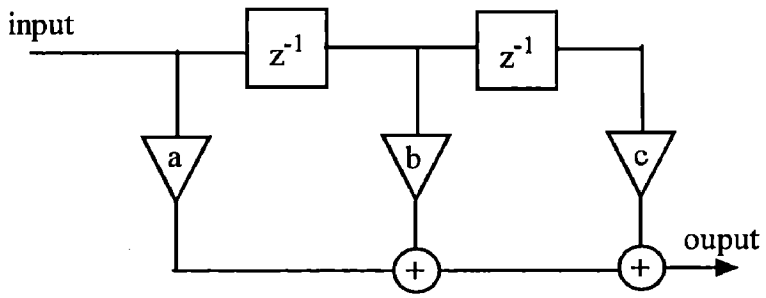


Figure 4.5: An Example of an FIR Filter

The zeros of the transfer function should be placed in a high-pass configuration, i.e., the zeros should have comparatively small angles with respect to the real axis, in order to attenuate the quantization noise in low frequencies and they should lie on the unit circle, to attenuate as much as possible. The frequency axis is placed on the unit circle in the complex plane and the closer the zeros are to the unit circle the more they attenuate. The zeros must also be conjugated, i.e., each zero is mirrored in the real axis and are therefore paired. This is due to the frequency axis being mirrored in the frequency $f_s/2$, the Nyquist frequency, where f_s is the sampling frequency.

Therefore the transfer function of the second order NTF in pole zero configuration is given by

$$NTF(z) = \frac{\left(z - e^{j\alpha}\right)\left(z - e^{-j\alpha}\right)}{z^2} \qquad [4.12]$$

Where 'α' is the angle of zero with respect to real axis

Polynomial expression of the above transfer function is obtained as follows

$$NTF(z) = \left(z^{-1} - e^{j\alpha}\right)\left(z^{-1} - e^{-j\alpha}\right)$$

$$= z^{-2} - 2z^{-1}\cos\alpha + 1 \qquad [4.13]$$

As the input to the modulator is oversampled the audio signal range(0-20KHz) becomes very narrow compared to the sampling frequency (7.761MHz) and is located around the DC frequency therefore the zeros of the noise transfer function are chosen around DC frequency such that it attenuates the quantization noise in audio frequency region. Therefore making α=0 we get,

$$NTF(z) = z^{-2} - 2z^{-1} + 1 \qquad [4.14]$$

The pole-zero diagram and magnitude response of the designed second order Error Feedback NTF are given in Fig.4.6 and Fig. 4.7 respectively.

From the Fig. 4.7 it can be observed that as discussed earlier the noise transfer function acts as high filter, it shapes the quantization noise introduced by the truncator such that quantization noise is low in the band of interest i.e audio signal band.

The shaped quantization and its power spectral density are given by

$$E_2(z) = E(z)\left(1 - 2z^{-1} + z^{-2}\right) \qquad [4.15]$$

$$S_{E_2}\left(e^{j\omega}\right) = \left|1 - 2e^{-j\omega} + e^{-j2\omega}\right|^2 S_E\left(e^{j\omega}\right) \qquad [4.16]$$

Fig.4.8 shows the corresponding noise-shaping loop, with 1-bit truncator in accordance with design specifications of the system as discussed in the section 2.5.The output signal of the modulator consists of the Most Significant Bit (MSB) of input signal and the feedback signal consists of the N-1 Least Significant Bits (LSBs).

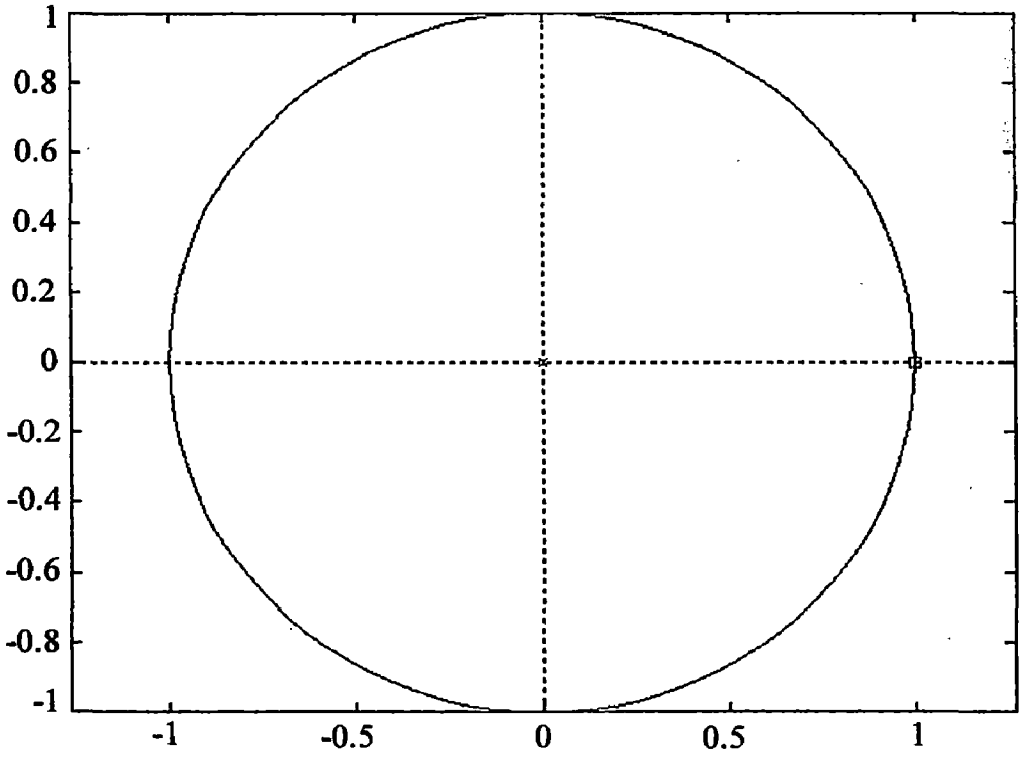


Figure 4.6: Pole-Zero Diagram of the Designed Second Order Error Feedback Modulator

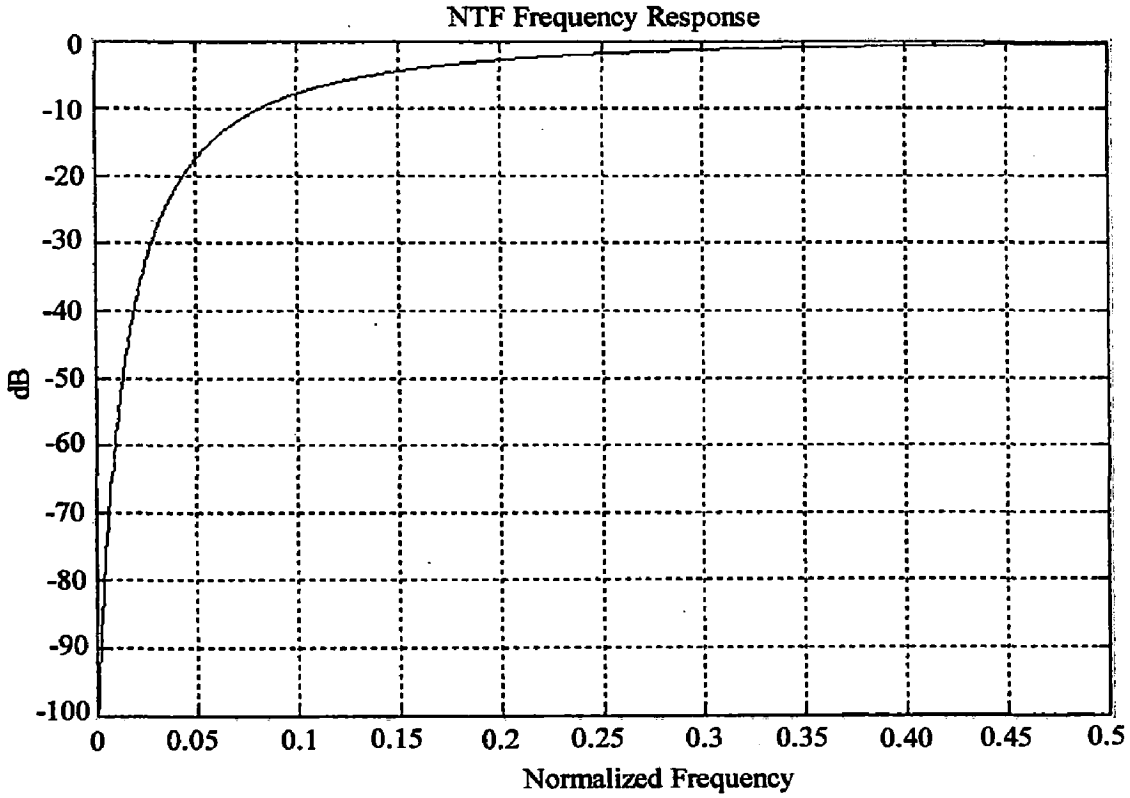


Figure 4.7: Noise Transfer Function of the Second Order Error Feedback Modualtor

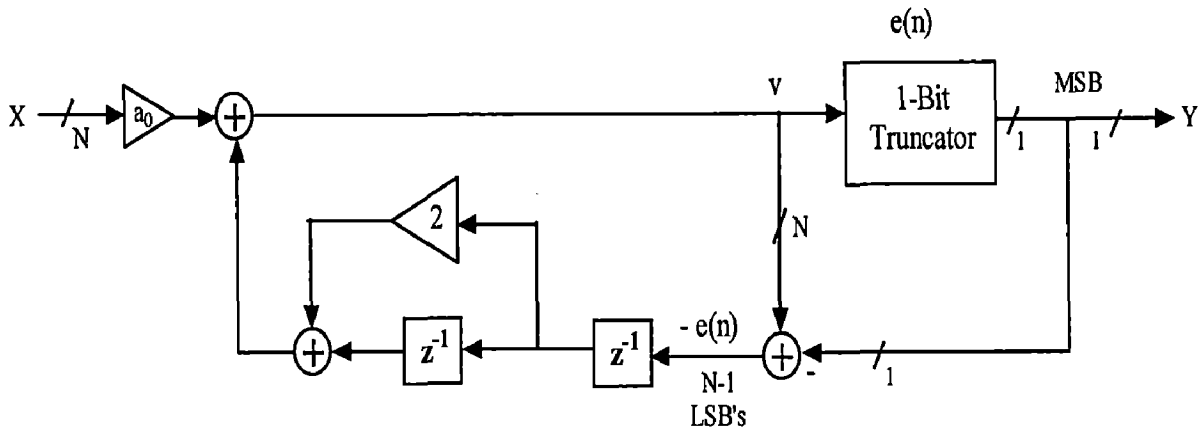


Figure 4.8: Second-Order Noise Shaper with Restricted Input Signal

In the Delta-Sigma loop, instability is caused when the input signal v(n) of the quantizer (here, the truncator) grows beyond the operating range of the employed digital logic bits. Depending on the arithmetic used, this may just cause the saturation of y(n) at its largest possible value, or it may cause a wrap around, where the output y(n) suddenly decreases with increasing v(n) at overflow. While saturation is usually acceptable, wrap-around causes large errors, and hence must be prevented.

We can note that the above structures do not require multiplication of multi-bit words; only delays, additions and shifts of the binary point are needed. This leads to a great saving in area required for implementation.

The next sub-section shows the simulation results conducted in MATLAB for the complete designed DAC.

4.2.1. MATLAB Simulations

This sub-section presents the MATLAB simulation results of the designed Delta-Sigma DAC. A 12 KHz sinusoidal input of amplitude -6dBFS is applied which is sampled at 44.1 KHz. It is then interpolated to 7.661 MHz sampling rate using two stage CIC interpolation filter. The output of the interpolation filter is then applied to the Delta-Sigma modulator, the input and output waveforms of the modulator is shown in the Fig.4.9, the output wave form is 1-bit waveform. As this figure shows, the only similarity between the input and output waveform in the time domain is that when the input is positive, the output is usually either +1 or 0, and that when the input is negative, the output is usually either -1 or 0. Finally, the quantized 1-bit DAC output is passed through an FFT to obtain the output frequency spectrum, the obtained power spectral density of the modulator is

shown in Fig.4.10. The simulation of the SNR as a function of the input amplitude is shown in Fig.4.11.

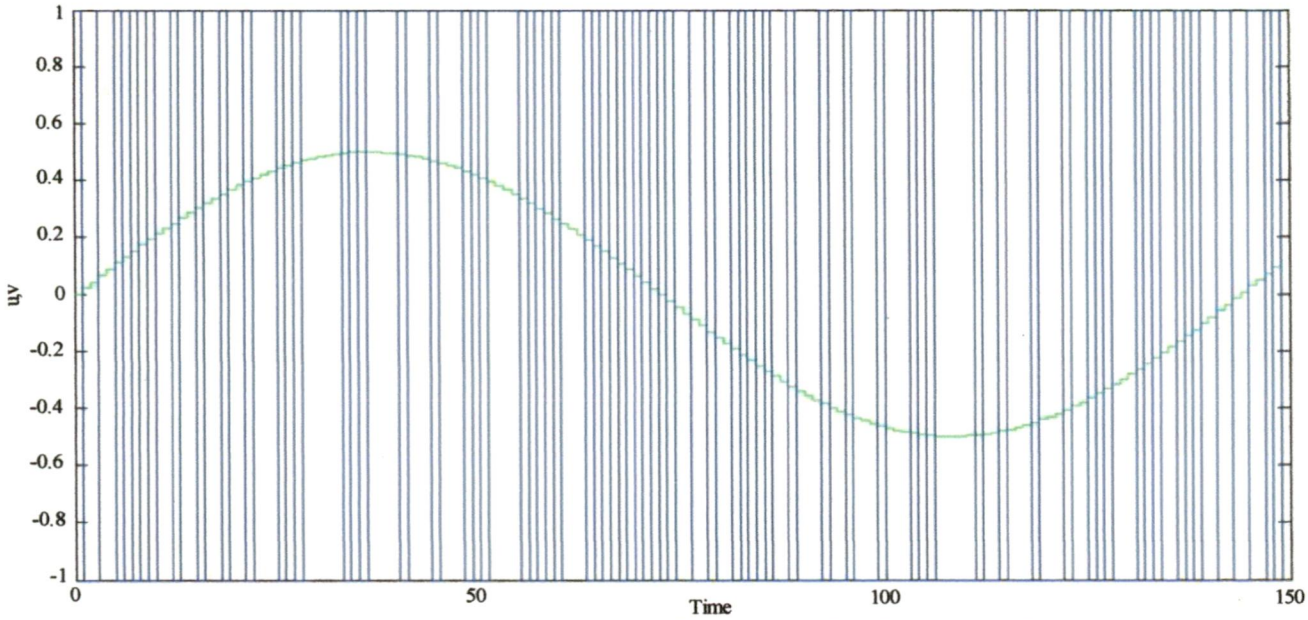


Figure 4.9: Input and Output Waveforms of Delta-Sigma Modulator

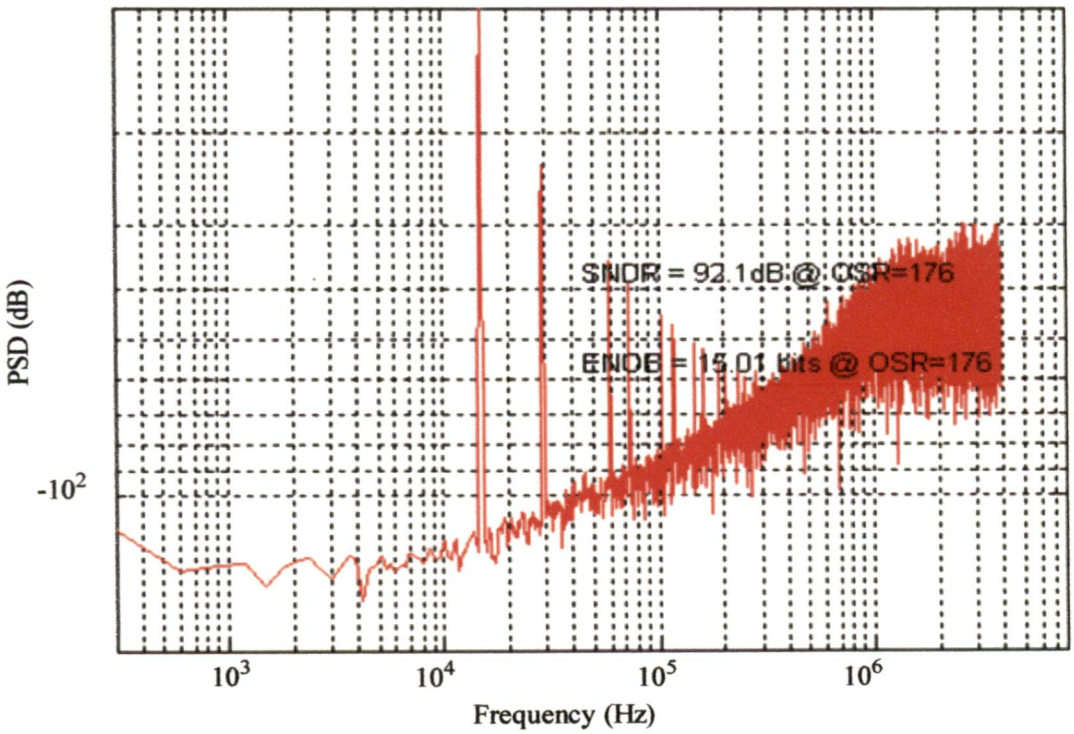


Figure 4.10: Output Power Spectral Density of Designed Delta-Sigma DAC

It can be seen that the maximum SNR obtained is92.1dB, which corresponds to an ENOB of 15.01 bits. The system level simulation thus verifies the correctness of the DAC architecture.

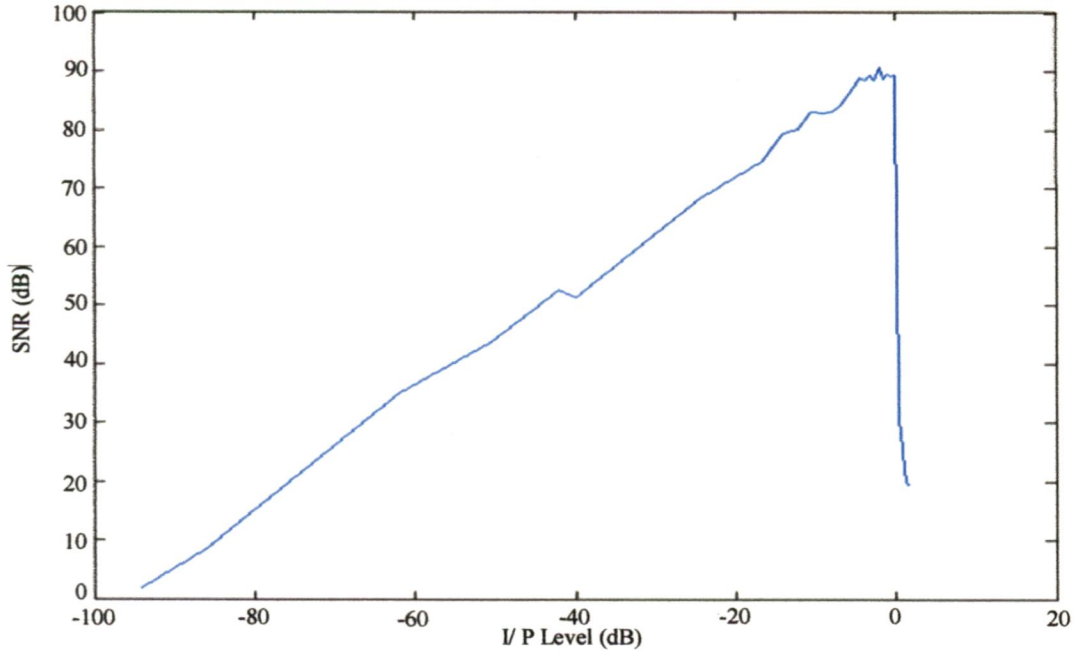


Figure 4.8: SNR vs Input Level

The next chapter discusses the Implementation details and results of the designed Delta-Sigma DAC on FPGA. The Xilinx based system generator design flow is also discussed.

Chapter 5

FPGA IMPLEMENTATION

Field Programmable Gate Arrays (FPGAs) are able to offer advantages over traditional VLSI technology in terms of time to market, lower costs for small quantities and dramatically reduced development times. As Moore's law continues to improve device density, the trend is to integrate increasingly higher levels of functionality into FPGA designs.

Although FPGA technology might at first seem to not be suitable for the implementation of analog components such as a DAC, their architectures turn out to be very suitable for Delta-Sigma converters which are primarily digital. Having the flexibility to incorporate DACs into FPGA designs allow for higher levels of integration, reducing cost, board area and possibly power consumption. FPGAs also make an excellent prototyping environment for Delta-Sigma converter designs [29].

In this work Spartan 3E (xc3s500e-4fg320) FPGA kit has been used for the implementation of designed DAC. Spartan 3E FPGA kit is most widely used FPGA kit because of its advantages like low power, low cost etc [30]. In this thesis, The FPGA implementation was developed using the Xilinx System Generator for DSP_{TM}, a Simulink-based tool for FPGA design. Design flow based on System Generator tool derives a hardware realization directly from the system model via automatic code generation. This methodology is sometimes referred to as 'Model-Based Design' [31], such high level design approaches aim to increased productivity (from higher levels of abstraction) and reliability (from automatic code generation and more robust test methodologies).

The following sections briefly explain the Spartan 3E FPGA architecture and FPGA Implementation using Xilinx System Generator tool [32]. Finally FPGA implementation results of the Designed DAC are presented.

5.1. Architectural Overview

The Spartan-3E family architecture consists of five fundamental programmable functional elements:

- **Configurable Logic Blocks (CLBs)** contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.
- **Input /Output Blocks (IOBs)** control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Support a variety of signal standards, including four high-performance differential standards. Double Data-Rate (DDR) registers are included.
- **Block RAM** provides data storage in the form of 18-K-bit dual-port blocks.
- **Multiple Blocks** accept two 18-bit binary numbers as inputs and calculate the product.
- **Digital Clock Manager (DCM)** provides self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

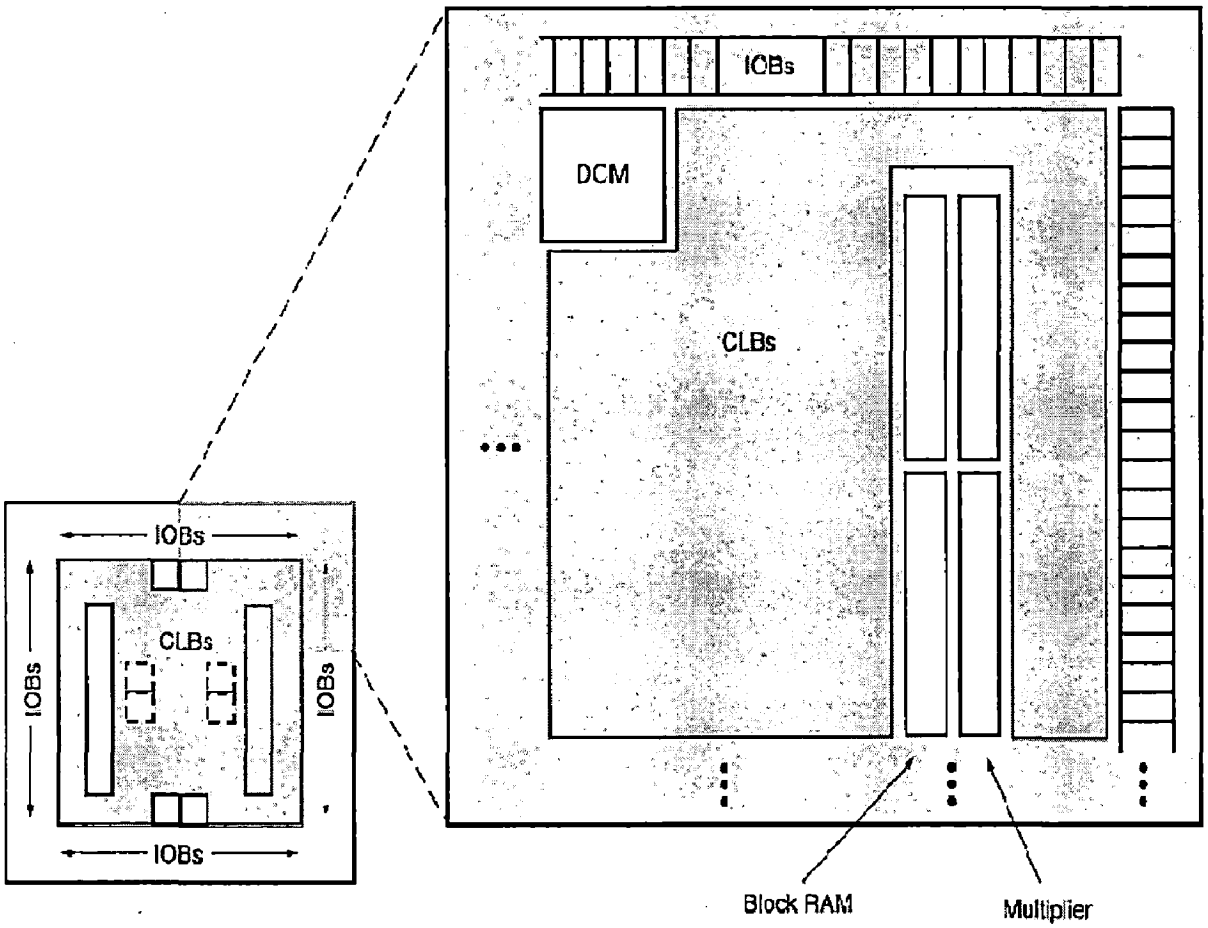


Figure 5.1: Spartan 3E Family Architecture

These elements of Spartan 3E family architecture are shown in Fig.5.1. A ring of IOBs surrounds a regular array of CLBs. Each device has two columns of block RAM except for the XA3S100E, which has one column. Each RAM column consists of several 18-Kbit RAM blocks. Each block RAM is associated with a dedicated multiplier. The DCMs are positioned in the center with two at the top and two at the bottom of the device.

5.2. FPGA Design Flow using Xilinx System Generator

In order for an FPGA to carry out a set of instructions, it must be first programmed by loading a design via a bit-stream file into the FPGA internal configuration memory. The bit-stream file contains all of the configuration information from the physical design defining the internal logic and interconnections of the FPGA, as well as device-specific information from other files associated with the target device.

Simulink/Matlab has been created as a system-level design tool that enabled us to exploit visual data flows. Simulink also offers a path to automatically generate an FPGA bit stream (program the FPGA) on the target development board through third-party tool such as Xilinx System Generator . Therefore, Xilinx System Generator software has been chosen to exploit a fast route to programming the FPGA.

Fig.5.2 shows the FPGA design flow methodology used in this work. It consists on a basic four-step methodology for implementing FPGA applications [33]: 1) system design, 2) verification and simulation, 3) implementation and 4) configuration.

- **System Design** This first step describes the model design that has to be implemented onto the FPGA device. In our case, we create the designs using System Generator which is a visual programming environment.
- **Verification and Simulation** Functional verification and simulation checks the logical correctness of FPGA design. Once any form of design entry has represented a design, it is necessary to verify if such a description satisfies the design specifications.
- **Implementation** This step includes the synthesis and place & route processes. Synthesis converts design entry into actual gates/blocks specified in FPGA devices.

- This is a very important step of the whole design procedure. A lot of efforts have been put into the improvement of the synthesis algorithms. On the other hand, the place & route tool selects the optimal position for elementary design blocks and minimizes length of interconnections on FPGA devices.
- **Configuration** This final step implies downloading bit stream codes onto the FPGA device. The Xilinx system development provides with the Project Navigator software which includes the iMPACT tool. iMPACT works as the programming interface between a PC (host) and the FPGA.

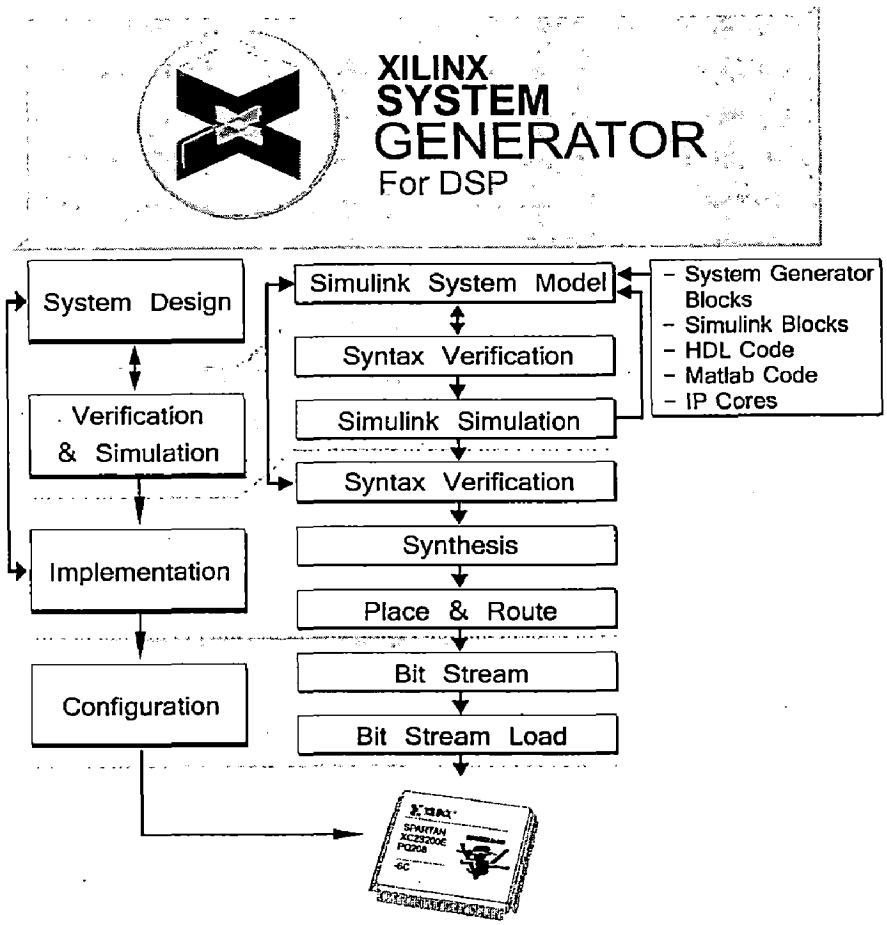


Figure 5.2: FPGA Based Platform Design Flow

5.3. Implementation

Having reached the desired performance throughout the verification and simulation processes, source design can be implemented. Basically, this is accomplished by translating the block model into an HDL code or a .bit configuration file. This step is the

combination of several process chained automatically. First, double-click is pressed on the System Generator token in the top level of the model to open the hardware generation Graphical User Interface (GUI) that lets to specify FPGA family and device, net list type, Simulink clock rate, and whether a test bench is needed. Clicking Generate produces a cycle and bit accurate HDL net list that can be synthesized and placed-and-routed using Xilinx ISE Foundation FPGA implementation software. Figure 5.3 illustrates the configuration options environment from the System Generator block. This block generates all necessary files for the FPGA application:

- Verilog file of the circuit, optimized for the target FPGA family.
- Synthesis scripts for different synthesizers (XST, Leonardo Spectrum, Synplify).
- Scripts for Xilinx Place & Route.

5.4. FPGA Configuration

This section describes the process of loading the configuration bit-stream (.bit file) into the FPGA. To download the .bit file we will be using iMPACT (part of the Xilinx tools installation). The following five steps outline this process

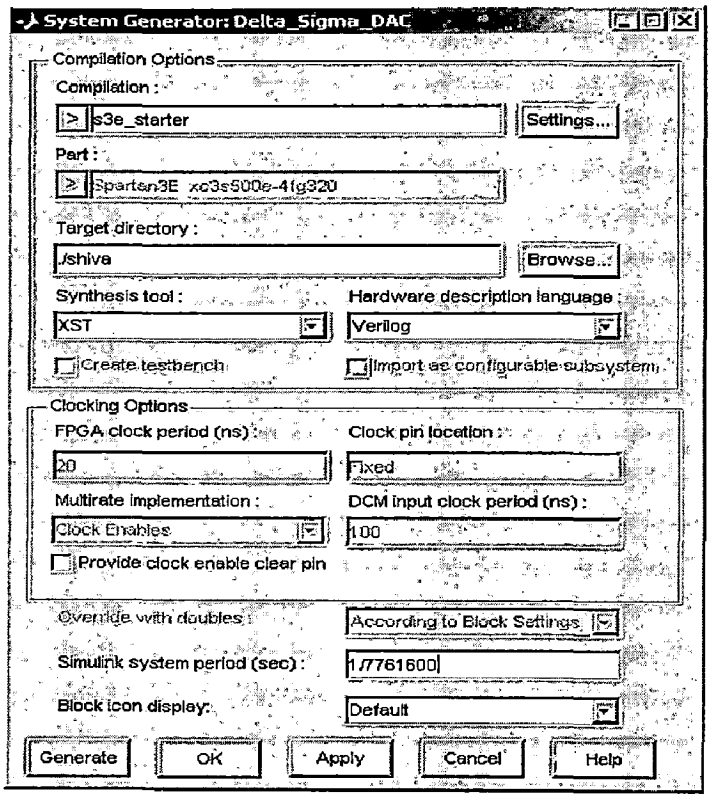


Figure 5.3: System Genarator Token

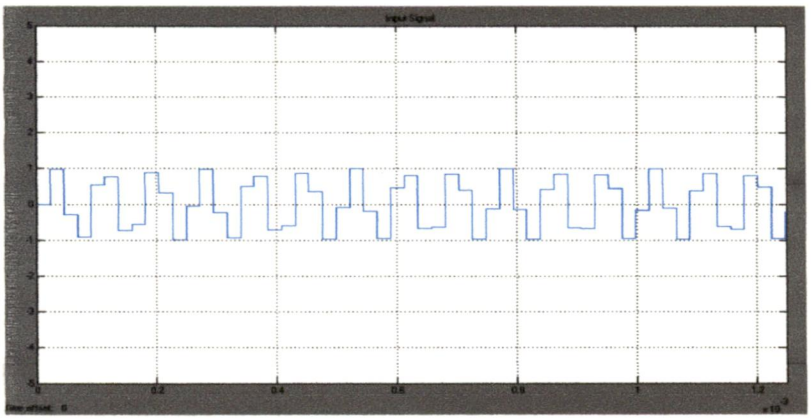
1. When iMPACT starts up, we select the default choice Configure Devices in the first dialog. Next, select Boundary-Scan Mode, and the Automatically connect to cable and identify Boundary-Scan chain.
2. iMPACT will then connect to the JTAG adapter cable, and search the JTAG chain for devices. If this fails, there is a communication problem with the parallel port on the PC, a problem with the JTAG cable, or a problem with the JTAG scan chain on the board. It will not be possible to configure the board until these problems are resolved.
3. Once iMPACT has identified all devices on the JTAG scan chain, we have to assign the programming file to each of the devices in the chain, in turn.
4. In the Assign New Configuration File dialog box for each device, we select the configuration file for the type of device and click on Open, or click on Bypass to put that device into bypass mode, causing it not to be configured. When prompted to select a programming file for the FPGA, we select the configuration bit file for the Spartan-3e (choosing the .bit file from the project directory) device.
5. After configuration files have been assigned to all devices, a device is configured by right clicking on its icon in the iMPACT window and choosing Configure from the menu. Appropriate options for the device being programmed must be selected from the Program Options dialog box. iMPACT will configure the device and then indicate whether configuration was successful or not. If HDL code has been created, the Xilinx Integrated Software Environment (ISE) can be used to convert a collection Verilog or VHDL files into a configuration bit-stream (.bit file) for the FPGA application.

5.5. Results

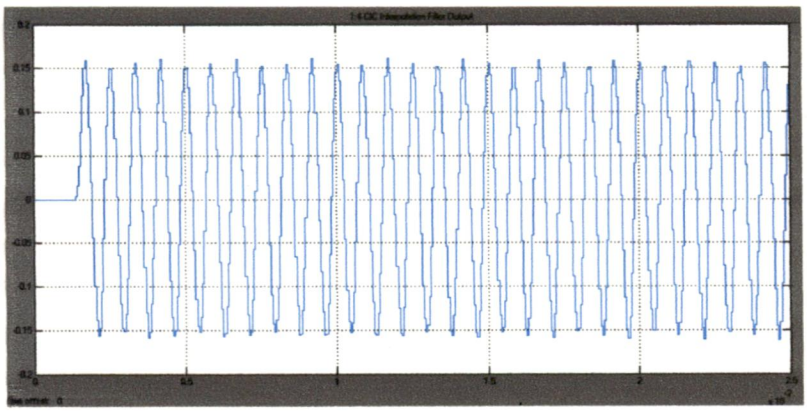
System Generator also extends Simulink through standard APIs to interface directly to HDL simulation tools (enabling import of HDL modules) and directly to hardware platforms (hardware co-simulation). System Generator provides hardware co-simulation, making it possible to incorporate a design running in an FPGA directly into a Simulink simulation. "Hardware Co-Simulation" compilation targets automatically create a bit-stream and associate it to a block [33]. When the design is simulated in Simulink, results for the compiled portion are calculated in hardware. This allows the compiled portion to be tested in actual hardware and can speed up simulation dramatically. This allows us to

exploit FPGAs to the extent that we can significantly accelerate imulation, and validate our design working in hardware, all without necessarily having to invoke a traditional FPGA tool explicitly.

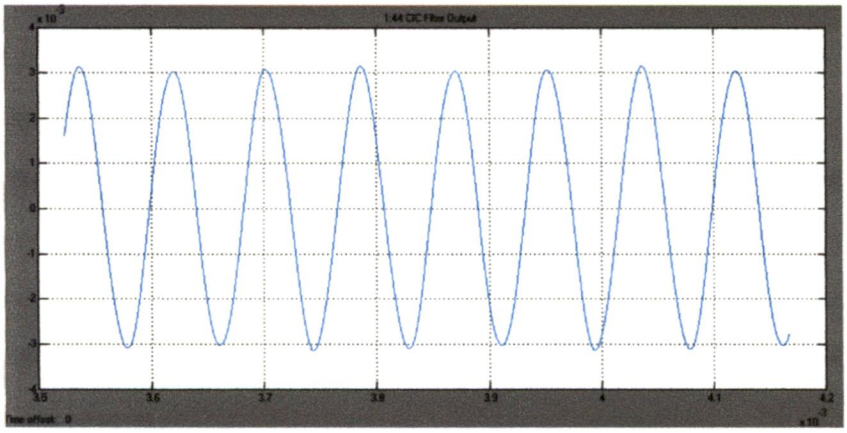
The stage wise outputs of the hardware co-simulated DAC are shown in Fig. 5.4. A Sinusoidal signal of frequency 12 KHz at 44.1 KHz with sampling frequency and 16-bit data width is applied FPGA configured as Delta-Sigma DAC. It is initially 4 or times oversampled by the first stage CIC filter whose output is shown in 5.4(b) and then again 44 times oversampled by second stage CIC filter whose output is shown in 5.4(c). This signal now applied to the Delta-Sigma Modulator this modulator truncates data width of the input signal to 1 bit. The 1 bit FPGA output is given to a 3rd order Butterworth analog Low pass filter which acts as a reconstruction filter whose output is shown in Fig. 5.4(e)



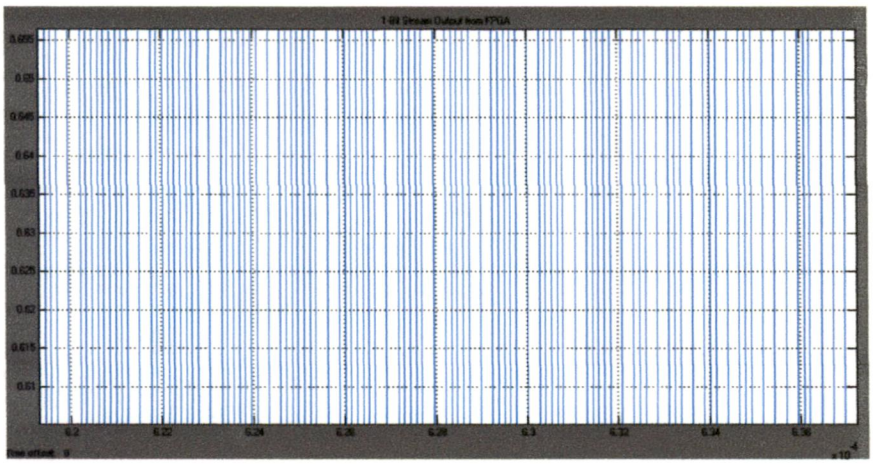
(a)



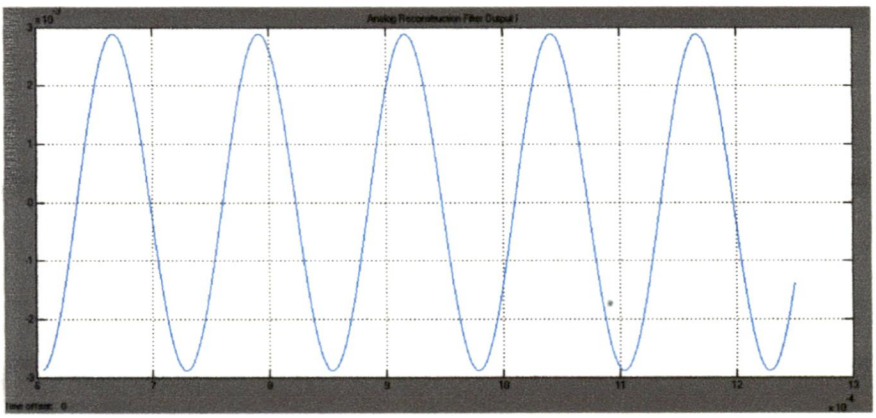
(b)



(c)



(d)



(e)

Figure 5.4: Stage wise outputs of the designed Delta-Sigma DAC, (a) Input Signal (b) CIC 1:4 Interpolation Filter Output Signal (c) CIC 1:44 Interpolation Filter Output Signal (d) 1-Bit Output Signal from FPGA (e) Analog Reconstruction Filter Output

5.5.1 Synthesis Results

Table I shows the hardware resources occupied in the Xilinx Spartan 3E (xc3s500e-4fg320) circuit. The whole system occupies 16% of the whole FPGA hardware resources. the The maximum operating frequency of the system is 7.661MHz which is 176 times of the input sampling frequency. The generated floor plan of the designed Delta-Sigma DAC is shown in Fig.5.5 and its enlarged version is shown in Fig.5.7.

Table I: Xilinx Spartan 3E (xc3s500e-4fg320) Device Utilization

Unit Name	Number used	Total Number	Percent
Number of Slice Filp Flops	1,176	9,312	12%
Number of 4 input LUTs	1,177	9,312	12%
Number of slices	788	4656	16%
Number of bonded IOBs	1	232	1%
Number of RAMB16s	2	20	10%
BUFGMUXs	4	24	16%
BSCANs	1	1	100%

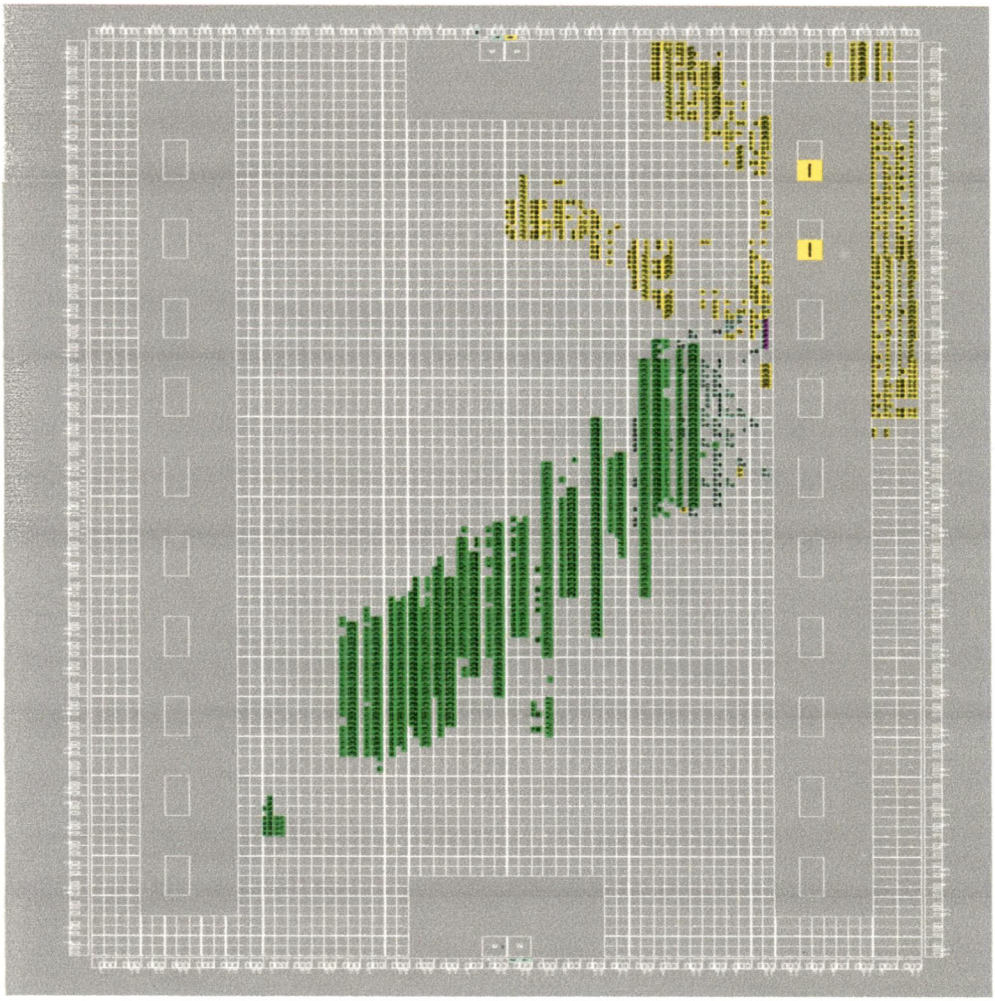


Figure 5.5: System Components FPGA Configuration

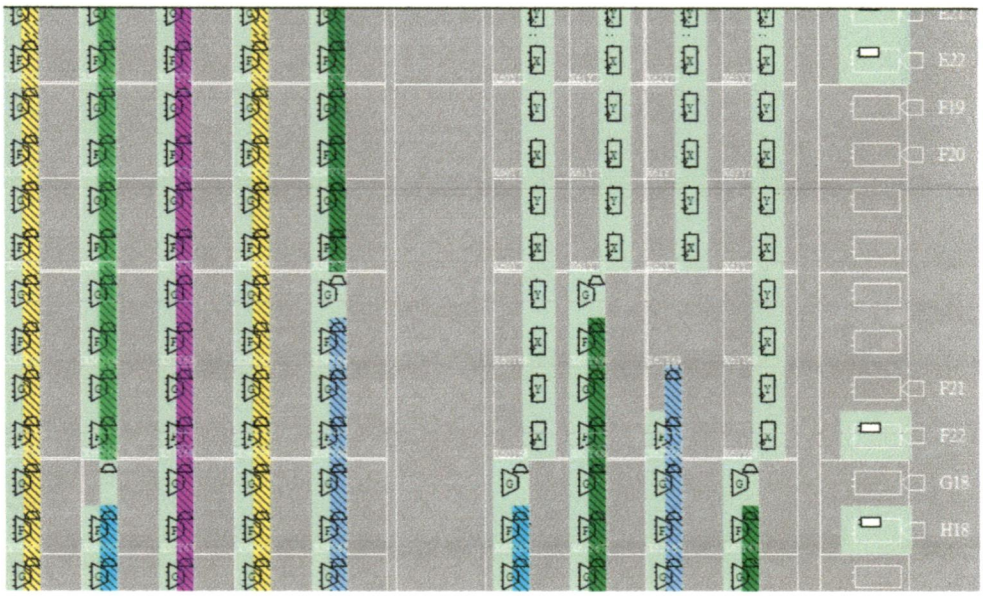


Figure 5.6: System Components FPGA Configuration Enlarged

Power Analysis

The estimated power consumption of the DAC is found to be 81.85mW using the Xilinx XPower Analyzer. This tool gives the power value at worst case conditions i.e without heat sink and at high junction temperatures. Fig.5.7 shows the individual power taken by the each element and also the total power consumed by the entire DAC.

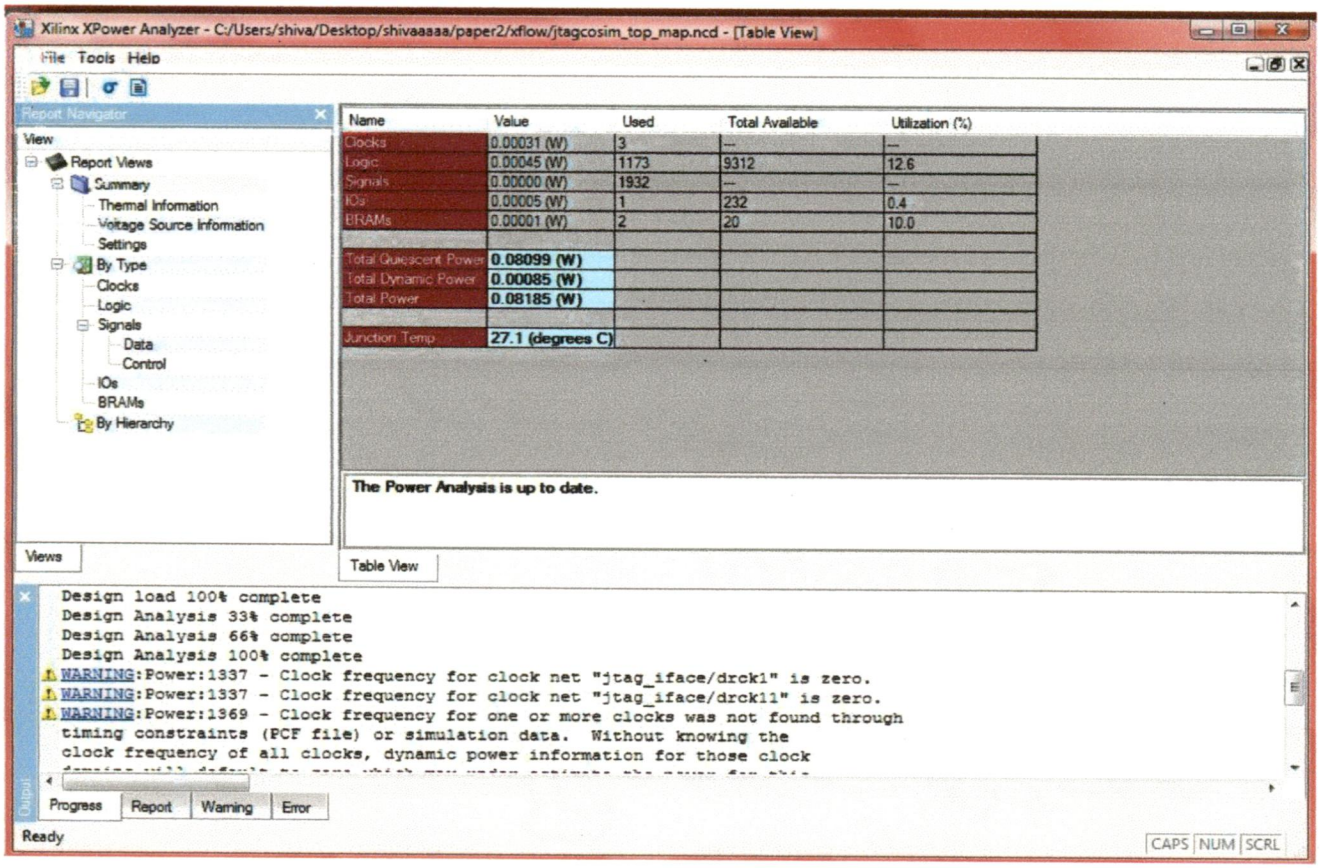


Figure 5.7.: Screen Shot of the Power Analyzer Tool

Chapter 6

CONCLUSIONS

The ΔΣ modulator presented in this design employed a second order noise shaper with error-feedback. The choice of this type of noise shaper helped in eliminating the use of multipliers thus reducing the area required. The main characteristics of the designed DAC are summarized in Table II which is shown below:

Table II. Main Characteristics of Designed DAC

Component/ Parameter	**Type / Value**
Interpolator	Two Stage CIC Filter
Modulator	Second order with Error Feedback
Input Data Width	16 bits
Input Sample Rate	44.1 KHz
Oversampling Ratio	176
Dynamic Range	90dB
Output SNR	92dB
Output Sample Rate	7.761 MHz
Output Data Width	1 bit
Application Area	Portable Audio Systems

This thesis presented an interpolator architecture for a ΔΣ DAC which used CIC filters instead of Half Band Filters (HBFs) resulting in a coefficient-free design. Also, the employment of CIC filters led to a design which required no multipliers and reduced amount of storage requirements. This led to 10-20 times decrease in area and 10-15 times decrease in power requirements. The technique of pipelining used in CIC filters resulted in an increase in throughput.

The use of Xilinx based System generator tool for implementing the above design on FPGA resulted in a drastic decrease of design time required to generate the bit-stream file. As targeted by the architecture, the reduction in area can be observed in terms of the number of slices used in FPGA implementation. This count was found to be equal to 788 out of 4656 [see Table I] utilizing only 16% of the resources. Hence, our aim of developing a Delta-Sigma Digital to Analog Converter (DAC) has been achieved.

REFERENCES

[1] Udo Zulzer, "Digital Audio Signal Processing", A John Wiley & Sons Ltd. Publication, 2008.

[2] I. Fujimori, A. Nogi, and T. Sugimoto, "A Multibit Delta-Sigma Audio Dac With 120db Dynamic Range". IEEE Journal of Solid-State Circuits, vol. 35, no.8, pp.1066-1073, August 2000.

[3] I. Fujimori, and T. Sugimoto, "A 1.5 V, 41 mw Dual-Channel Audio Delta-Sigma D/A Converter", IEEE Journal of Solid-State Circuits, vol. 33, no.12, pp. 1863-1998, December 1998.

[4] Philips Intellectual Prorerty and Standards, "IEC-908: Compact Disc Audio-The Red Book", International Electrotechnical Commission Stadards Document, no. 28/10/04-3122 783 0027 2, June 1980.

[5] S. R. Norsworthy, R. Schreier and G. C. Temes, "Delta-Sigma Data Converters: Theory, Design, and Simulation", IEEE Press, NY, USA, Nov. 1996.

[6] R. Schreier, and G.C. Temes, "Understanding Delta-Sigma Data Converters", IEEE Press, 2005.

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

[8] Yasuyuki Matsuya, Kuniharlu Uchimura, Atsushi Iwata, and Takao Kaneko, "A 17-bit Oversampling D-to-A Conversion Technology Using Multi-Stage Noise Shaping," IEEE Journal of Solid State Circuits, vol. 24, pp. 969-975, August 1989.

[9] Hans J. Schouwenaars, D. Wouter J.Groeneveld,Carneils A. .Bastiaansen, Hank A. H. Termeer, "An Oversampled Multibit CMOS D/A Converter for Digital Audio with 115-dB Dynamic Range," IEEE Journal of Solid-State Circuits, vol.26, no.12, pp.1775-1780, Dec 1991.

[10] Peter J. A. Naus, Eise C. Dijkmans, Eduard F. Stikvoort, Andrew J. McKnight, David J. Holland, and Werner Bradinal, "A CMOS Stereo 16-bit D/A Converter for Digital Audio," IEEE Journal of Solid-State Circuits , vol. sc-26, pp.390-394, June 1987.

[11] Peicheng Hu, Ken Suyama, Paul F. Ferguson, Jr., and Wai Lee, "A 22-KHz Multi-bit Switched-Capacitor Sigma-Delta D/A Converter with 92dB Dynamic Range," IEEE Journal of Solid-State Circuits, vol.30, pp.1316-1325, December 1995.

[12] Feng Chen and Bosco Leung, "A High-Resolution Multibit Sigma-Delta Modulator with Individual Level Averaging," IEEE Journal of Solid-State Circuits, vol. 30, pp.453-460, April 1995.

[13] David K. Su and Bruce A. Wooley, "A CMOS Oversampling D/A Converter with a Current-Mode Semidigital Reconstrucion Filter," IEEE Journal of Solid-State Circuits, vol. 28, pp.1224-1233, December 1993.

[14] K. Nguyen, R. Adams and K. Sweetland, " A 113dB SNR Oversmpling Sigma-Delta DAC for CD/DVD Applications", IEEE Transactions on Consumer Electronics, vol. 44, no. 3, pp. 1019-1023, August 1998.

[15] N. K. Kim, W. S. Yeum, and J. W. Kim, "A 2.4V, 12 mW Stereo Audio D/A Converter With Double Sampling Switching", ISCAS 2000-IEEE International Symposium on Circuits and System, Geneva, Switzerland, May 28-31, 2000.

[16] M. Annovazzi, V. Colonna, G. Gandolfi, F. Stefani, and A. Baschirotto, "A Low-Power 98-dB Multibit Audio DAC in a Standard 3.3-v 0.35-um CMOS Technology", IEEE Journal of Solid-State Circuits, vol. 37, no.7 pp. 825-834, July 2002.

[17] Behzad Razavi, "Principles of Data Conversion System Design", IEEE Press, 1995.

[18] Schreier, R., "An Empirical Study Of High-Order Single-Bit Delta-Sigma Modulators", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, , vol.40, no.8, pp.461-466, Aug 1993.

[19] K. A. Shehat, M. A. Aboul-Dahab, S. H .El Ramly, K. A. Hamouda, "An FPGA Based 1-bit All Digital Transmitter Employing Delta-Sigma Modulation with RF output for SDR," 2nd International Conference on Signals, Circuits and Systems, 2008. SCS 2008., vol., no., pp.1-6, 7-9 November 2008.

[20] S. K. Mitra, Digital signal processing-a computer based approach, second edition, McGraw-Hill Companies, Inc., 2001

[21] R. E. Crochiere and L. R Rabiner, " Interpolation and Decimation of Digital Signals- a tutorial review", Proc. of the IEEE, vol. 69, no.3, pp. 300-331, March 1981.

[22] Ben Ameur, N.; Loulou, M., "Design of efficient digital interpolation filters and sigma-delta modulator for audio DAC," 3rd International Conference on Design and Technology of Integrated Systems in Nanoscale Era, 2008. DTIS 2008., vol., no., pp.1-7, 25-27 March 2008.

[23] Y.Djadi, T.A. Kwasniewski, C. Chan and V.Szwarc, "A High Throughput Programmale Decimation and Intepolation Filter", Proc. of the International Conference on Signal Processing Applications and Technology 1994, pp. 1743-1748,1994.

[24] Uwe Meyer-Baese, "Digital Signal Processing with FPGAs", Springer Publications, 2003.

[25] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-29, no. 2, pp. 155-162, April1981.

[26] E. Terhardt: Calculating Virtual Pitch, Hearing Res., Vol. 1, pp. 155–182, 1979.

[27] ISO 389-7:2005: Acoustics – Reference Zero for the Calibration of Audiometric Equipment – Part 7: Reference Threshold of Hearing Under Free-Field and Diffuse-Field Listening Conditions, Geneva, 2005.

[28] Yonghong Gao; Tenhunen, H., "Implementation aspects of an oversampling D/A converter for DMT-ADSL systems," The 6th IEEE International Conference on Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99, vol.3, pp.1397-1400, 1999.

[29] Cheung, R.C.C.; Pun, K.P.; Yuen, S.C.L.; Tsoi, K.H.; Leong, P.H.W., "An FPGA-based re-configurable 24-bit 96kHz sigma-delta audio DAC," Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on , vol., no., pp. 110-117, 15-17 Dec. 2003.

[30] Xilinx. "Spartan-3E FPGA Family: Complete Data Sheet".

[31] Mathworks URL: http://www.mathworks.com

[32] Xilinx Inc., "System Generator User Guide", URL: http://www.xilinx.com/products/software/sysgen/app_docs/user_guide.htm

[33] G.E. Martinez-Torres, J.M. Luna-Rivera and R.E. Balderas-Navarro, "FPGA-Based Educational Platform for Wireless Transmission Using System Generator," Proc. of IEEE International Conference on Reconfigurable Computing and FPGA's, ReConFig 2006, vol., no., pp.1-9, Sept. 2006.

LIST OF PUBILICATIONS

[1] P. Shivakumar, Sudeb. Dasgupta, "A Novel Area Optimized Architecture for Δ-Σ Digital to Analog Converter for Audio Applications", IEEE 18th Annual Symposium on Emerging Needs in Computing, Communication, Signals and Power, Bangalore , August 2009.(Communicated)