ΔΣ Digital-To-Analog Converter with Reduced Hardware Requirements using Segmentation

By
Denis Romanov B. Eng. 2018

Department of Electrical Engineering
McGill University, Montreal

June 2021

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

Abstract

Presented in this work is a thorough examination of the use of segmentation in ΔΣ Digital-to-Analog converters (DAC). ΔΣ DACs use the principle of negative feedback to recreate an incoming signal and convert it into a pulse modulated density (PDM) signal. This effectively recreates the input signal, with some added quantization noise. Segmentation is a process in which a digital code is split into disparate codes that can then act as separate signals.

In this work, segmentation has been used to relax the requirements on the digital hardware while sacrificing little in the way of performance in terms of Signal-to-Noise ratio (SNR). This was accomplished by having the ΔΣ DAC split into multiple modulators, each of which dealt with a segment of the original input data. The ΔΣ modulator dealing with the most significant bits provides the bulk of the response. The remaining ΔΣ modulators dealing with the least significant bits provide some reprieve from the quantization noise, while also requiring less hardware to implement than the high-performance converter dealing with the most significant bits.

A rigorous mathematical analysis was first performed to demonstrate the effects of segmentation on a segmented ΔΣ modulator's performance. These established equations gave rise to several design principles, which were then used to create several examples of segmented ΔΣ modulator designs. This was then followed by several simulations in MATLAB's Simulink environment to back up those findings. Lastly, the simulation designs were mapped on to a field-programmable gate array (FPGA) and tested in a laboratory to confirm the simulations.

Résumé

Cette thèse présente un examen approfondi de l'utilisation de la segmentation dans les convertisseurs numérique-analogique (DAC) ΔΣ. Les DAC ΔΣ utilisent le principe de rétroaction négative pour recréer un signal entrant et le convertir en un signal de densité modulée par impulsions (PDM). Cela recrée le signal d'entrée, avec un bruit de quantification supplémentaire. La segmentation est un processus dans lequel un code numérique est divisé en plusieurs codes qui peuvent ensuite agir comme des signaux distincts.

Dans cette thèse, la segmentation a été utilisée afin d'assouplir les exigences sur le matériel numérique tout en sacrifiant peu de performance en termes de rapport signal/bruit (SNR). Ceci a été accompli en divisant le DAC ΔΣ en plusieurs modulateurs, chacun d'entre eux traitant un segment des données d'entrée d'origine. Le modulateur ΔΣ traitant des bits les plus significatifs fournit la majeure partie du comportement du modulateur segmenté. Les modulateurs ΔΣ restants traitant des bits les moins significatifs offrent un certain sursis au bruit de quantification, tout en nécessitant moins de matériel à mettre en œuvre que le convertisseur haute performance traitant des bits les plus significatifs.

Une analyse mathématique rigoureuse a d'abord été réalisée pour démontrer les effets de la segmentation sur les performances d'un modulateur ΔΣ segmenté. Ces équations ont donné lieu à plusieurs principes de conception, qui ont ensuite été utilisés pour créer plusieurs exemples de modulateurs ΔΣ segmentés. Cela a ensuite été suivi de plusieurs simulations dans l'environnement Simulink de MATLAB pour étayer ces résultats. Enfin, les plans de simulation ont été mis en correspondance avec un réseau de portes programmable *in situ* (FPGA) et testés en laboratoire pour confirmer les simulations.

Acknowledgements

Firstly, I would like to give my heartfelt that to Professor Gordon Roberts. Had it not been for his guidance, I would not have considered pursuing this degree to begin with. His insights, enthusiasm, and encouragement played a key role in this whole endeavor. He also helped greatly in writing for a research paper for the IEEE Transactions on Circuits and Systems (TCAS) and gave me many opportunities to work at McGill as a teaching assistant. Professor Roberts also gave me the opportunity to work with the wonderful people at Ciena, and the funding I needed to complete my studies.

I would also like to thank Dr. Ahmed S. Emara, a friend and my senior at McGill University. It is thanks to his own experience that I got the grasp of the principles that I needed to start work on this thesis at all. His encouragements as we both sought to complete our theses were not unappreciated. It was thanks to him that the testing board was developed and used in this thesis. Many thanks to all the members of the Microelectronics and Computer Systems (MACS) Laboratory, especially my cohorts Muhtady Muhsain and Rafid Khan. Special thanks to Sadok Aouini and Soheyl Ziabakhsh (whose seat I now occupy at the laboratory) of Ciena for their insights.

I also acknowledge the contributions of McGill University itself, and the partners I interacted with. I have received funding for my studies through the Mathematics of Information Technology and Complex Systems (MITACS). In addition, I acknowledge that the Canadian Microelectronics Corporation (CMC) has also contributed to my success by granting me access to many of their design tools.

And lastly, I would like to thank my family for all their support and encouragement. There are no words to properly convey the amount of help they have provided me, nor are there words to express my gratitude for that help.

In memory of my grandmother, Nina.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ADC	Analog-to-Digital Converter
CBS	Coarse Bitstream
DAC	Digital-to-Analog Converter
DAQ	Data Acquisition System
DDC	Digital-to-Digital Converter
DEM	Dynamic Element Matching
DNL	Differential Nonlinearity
DSP	Digital Signal Processing
FBS	Fine Bitstream
FPGA	Field Programmable Gate Array
INL	Integral Nonlinearity
LPF	Low-Pass Filter
LSB	Least Significant Bit
MSB	Most Significant Bit
NTF	Noise Transfer Function
Op-Amp	Operational Amplifier
PDM	Pulse Density Modulation
PWM	Pulse Width Modulation
SNR	Signal-to-Noise Ratio
STF	Signal Transfer Function
TLE	Total Logic Elements

Chapter 1: Introduction and Background

1.0. Introduction

This chapter will serve as an introduction to the core ideas explored in this thesis, namely the basics of mixed-signal circuit, Digital-to-Analog conversion (DAC), the history and operation of ΔΣ modulation, and the principle of segmentation in data converters. Lastly, the motivation behind this thesis is explained by introducing the main benefits and shortcomings of ΔΣ modulation, and how segmentation could be used to remedy the latter.

1.1. Mixed-Signal Systems

Firstly, this section will serve as an introduction to mixed-signal systems, as the entire thesis pertains to them.

1.1.1. Defining Mixed-Signal Systems

In electronics, one can categorize circuits and the signals associated with them as being either digital or analog. In practice, the signals encountered on a day-to-day basis are analog, such as soundwaves, while general purpose computers and dedicated integrated circuits deal with digital information in the form of binary codes.

A mixed-signal circuit is defined as a circuit which is comprised of both analog and digital components. This definition, while sound, does include certain circuits that many engineers would be hesitant to call mixed-signal circuits, such as the comparator. A comparator has two analog inputs, and one output that indicates which of the two input signals is higher (i.e. a binary output), as shown in Fig. 1.1. For the purpose of this thesis, the mixed signal circuits in question are ones that strictly convert analog electronic signals into digital information or vice-versa, which are widely agreed to be true mixed-signal circuits. These are the Analog-to-Digital converter (ADC or A/D) and the Digital-to-Analog converter (DAC or D/A).

1.1.2. Motivations to use Mixed-Signal Systems

Since the rise of home computers and especially after the smartphone's meteoric rise in prominence, the importance of mixed-signal circuits and systems

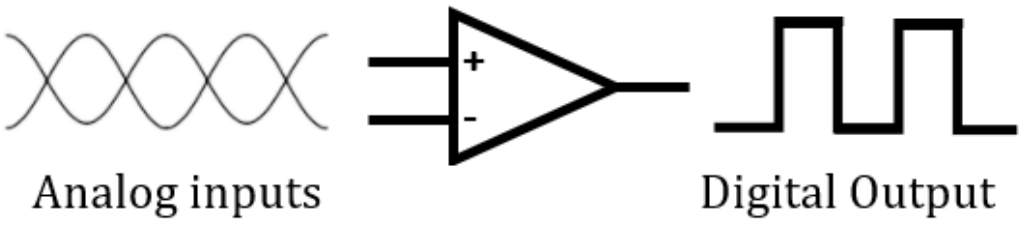

Fig.1.1. A comparator, using analog inputs and producing a digital output, may be considered a mixed-signal circuit

has only continued to grow over the last few decades. While analog signals are what most people deal with on a regular basis, one would be hard pressed to emulate the convenience of digital storage and data manipulation in the realm of analog electronics.

For a great number of applications, electronic systems are designed to accept analog inputs and converted into digital information through the use of sensors such as light detectors and microphones and transferring this information to a computer system.

This information can then be stored, distributed, edited, analysed, or processed with all the convenience and staggering options that digital systems offer. The high levels of integration that can be achieved with digital circuits, combined with the inherent imperviousness to noise that digital signals have as compared to analog signals as shown in Fig. 1.2. The subfigures (a) and (c) show a typical analog and digital signal respectively, while (b) and (d) show those same signals after being exposed to an identical level of noise. It is much easier to reconstruct the intended signal in the digital case, as in the analog case the distinction between noise and signal is not so easily made. The additional insensitivity to variations between parts in digital circuits as apposed to analog components makes the use of digital systems all the more desirable.

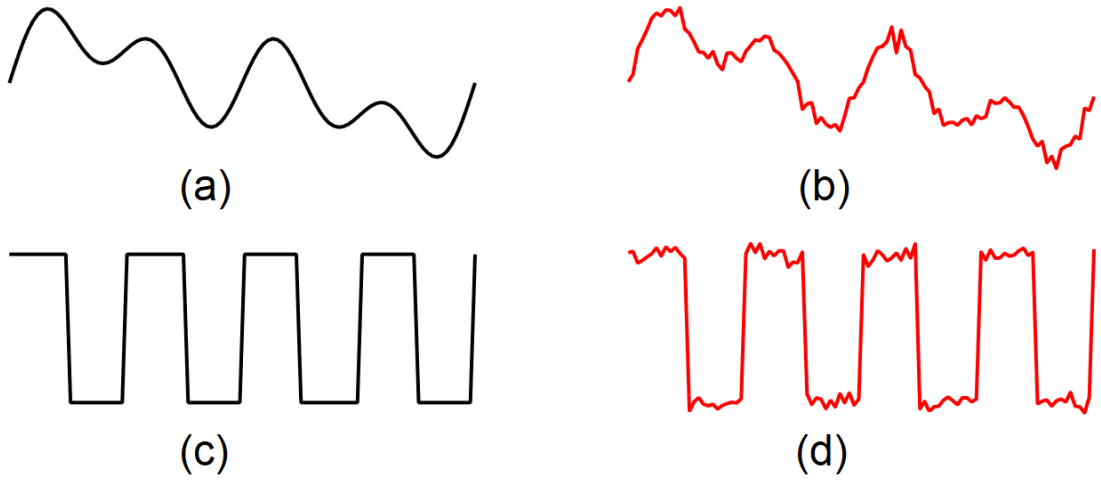

Fig.1.2. A comparison of an analog signal before and after being subjected to noise as (a) and (b) respectively, versus a digital signal before (c) and after noise injection (d).

In a mixed-signal system, this information is the converted into an analog signal using actuators such as displays and speakers, to be interpreted by people once again.

The sensors and actuators as mentioned earlier are in fact the ADCs and DACs from section 1.1.1. Both of these circuits have their own purposes, and seldom can an engineer find a complex system that does not employ both.

1.1.3. Relevance of DACs and ADCs to this Thesis

This thesis pertains to a specific kind of converter, known as Delta-Sigma (ΔΣ). While it can be implemented to use either as a DAC or ADC, this work focuses on the DAC implementation of a ΔΣ modulator. More specifically, this thesis explores some modifications that can be made to a ΔΣ DAC and how these modifications affect the converter's performance versus the costs incurred by these changes, as opposed to a more typical converter. Section 1.2. will cover the essentials of ΔΣ, and data converters in general.

1.2. Background Information

This section offers a quick overview of the principles of operation of DACs and ΔΣ modulators, as well as a brief history of the latter to understand the motivation behind using ΔΣ modulation. This section will also provide a brief introduction to the principle of segmentation, to better understand the motivation behind this thesis, as covered in section 1.3. A more in-depth look at the operating principles will be presented in chapter 2.

1.2.1. Basic Principles of Digital-to-Analog Conversion

The DAC acts as a decoder; it uses a digital input code representing an integer which is then converted into an analog voltage level. The operation can be roughly thought of as a multiplication by some factor to convert to and from the

analog and digital equivalents. Thinking of the output as V_O, and the input as D_{IN}, one can write

$$V_O = G_{DAC} * D_{IN} \tag{1.1}$$

where the factor G_{DAC} is some real-valued constant. Because D_{IN} pertains to data from a digital system, it is typically an N-bit wide base-2 unsigned integer. It would then be expressed as

$$D_{IN} = D_0 + D_1 * 2^1 + D_2 * 2^2 + \cdots + D_{N-1} * 2^{N-1}. \tag{1.2}$$

Given that the factors D_0 to D_{N-1} are bits, they will take on values of either 1 or 0, and D_{IN} will therefore be an integer ranging from 0 to 2^N-1. D_0 is called the least significant bit (LSB) as it has the least impact on D_{IN}, while D_{N-1} is considered the most significant bit (MSB) as it has the most impact.

A diagram demonstrating a typical DAC is shown in Fig.1.3. along with an example transfer curve for a 4-bit DAC with G_{DAC} of 0.1V. It is plain to see that this results in one-to-one mapping, with the LSB step size (least significant change in voltage) or V_{LSB} is equal to 0.1V.

1.2.2. History and Operation of ΔΣ Modulation

While some variation exists between various implementations of ΔΣ converters, the principle of negative feedback lies at the core of any and all of them. Fig. 1.4. demonstrates an example of a ΔΣ DAC structure.

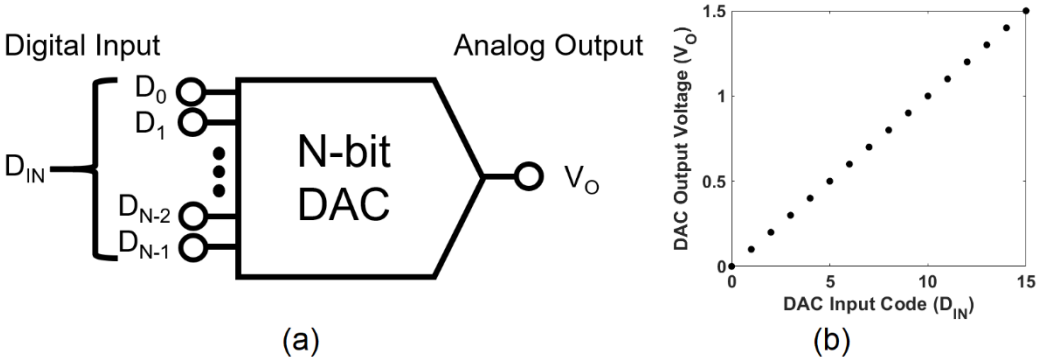

Fig.1.3. A diagram of an N-bit DAC (a) and a transfer curve for a 4-bit DAC

A digital code serves as the input to two difference amplifiers, one into the feedforward path and the other in the feedback path. The feedforward path leads into a comparator, whose output leads into the feedback path via a digital-to-digital converter for scaling purposes. Two of the signal paths are padded with 1 bit, as some operations may cause overflow otherwise. Here a multiplexer with two possible output states of *0* and 2^N*-1* is used to realize this digital-to-digital converter operation. These two integers represent the smallest and largest signal associated with an N-bit data path. This signal then passes into the difference amplifier that leads into the loop filter denoted by *H(z)*.

Over time, a bitstream appears at the output which contains a replica of the original incoming signal *x*(t) buried in quantization noise. The signal *x*(t) can be extracted from the bitstream using a lowpass reconstruction analog filter with a corner frequency set to the signal bandwidth of *x*(t). An example of this is demonstrated in Fig. 1.5. If one were to think of the modulator output in terms of a density of different code levels (i.e. integers), when scaled by the D/A block, the density can now be expressed in terms of a voltage-level pulse modulated density (PDM) signal.

The concept of ΔΣ modulation dates to 1962, as written by Inose [1]. The motivation for writing this at the time was a drawback in Delta modulation (ΔM) [2], introduced in 1948, which ΔΣ is itself derived from. The drawback stems from the fact that the pulses produced as the output to the delta modulator carry the information that corresponds to the differentiation of the amplitude of the input

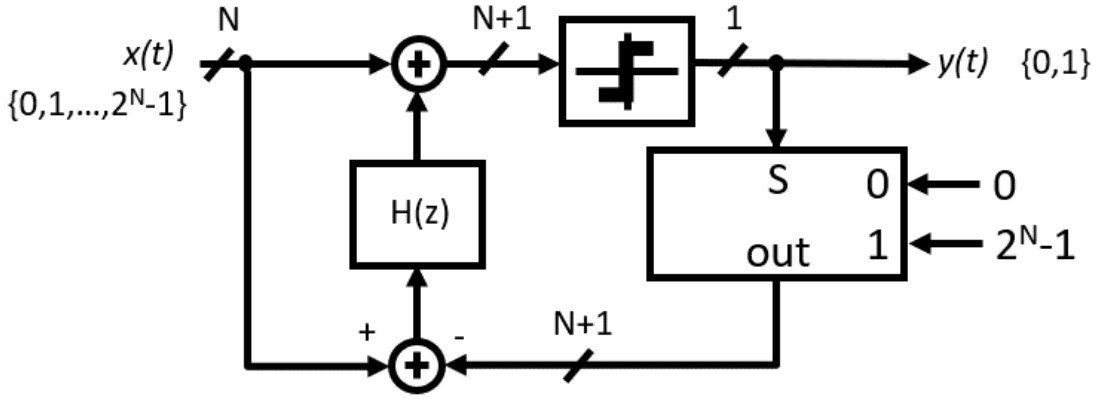

Fig.1.4. Structure of a ΔΣ modulator. The number of bits used for each signal path is identified.

signal. To put it simply; whether amplitude rises or falls from one sampling pulse to another. To recover the information contained in the input waveform, these pulses must be integrated. As a result, any transmission disturbances (namely noise) will cause an accumulated error in the output signal. This is in contrast to ΔΣ modulation, which uses what is effectively two opposing operations to achieve a not too dissimilar result.

1.2.3. The Principle of Segmentation

The process of segmentation in DACs typically involves the partitioning of input digital data into segments, typically into a segment comprised of the MSBs and another segment comprised of LSBs [3] The outputs of the individuals DACs are then recombined by a process specific to the segmented DAC in question.

There are multiple reasons for which a designer may use a segmented DAC. It may be that the performance requirements a designer faces may dictate specific performance metrics, for which no single architecture may be adequate. It may be that a linear increase in the number of bits leads to an exponential increase of hardware required to create the DAC. As such, segmenting a DAC may be worth it to save on hardware.

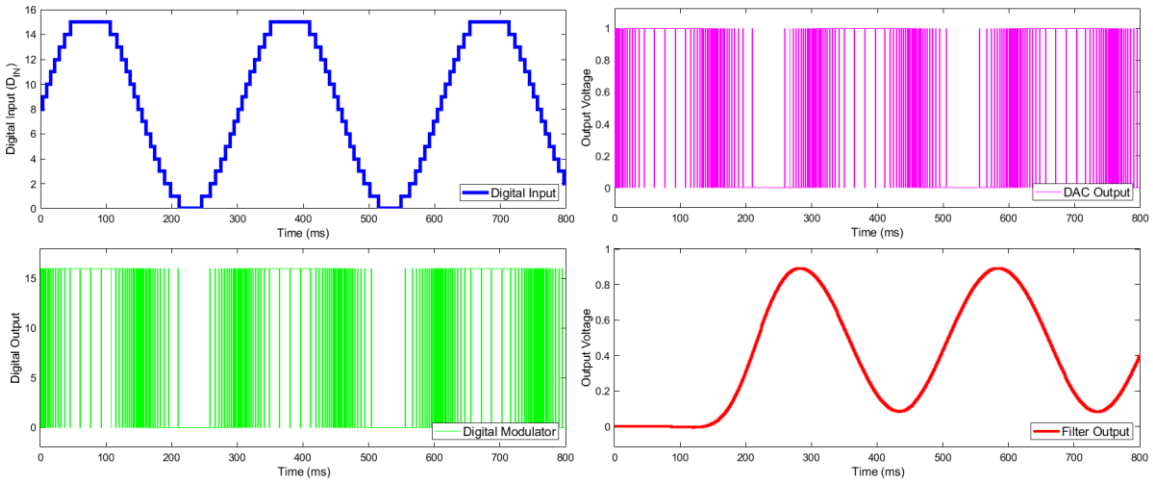

Fig.1.5. Various Stages of a 4-bit ΔΣ Modulator for a sinusoidal input. Note the distinction between the Modulator output in green and the DAC output in magenta in which the logic high and low are mapped to real voltages.

It is worth noting that segmentation need not send the data segments to separate converters. Indeed, some architectures may allow for data to be treated differently within the same modulator. Examples such as this will be elaborated on in the next chapter.

1.3. Motivation

This section covers the motivation behind the use of ΔΣ modulation, and the motivation behind the research performed for this thesis. Firstly, the benefits of ΔΣ modulation will be covered. Following this will be a subsection covering the drawbacks and limitations of ΔΣ modulation and the improvements sought to ΔΣ modulation through the use of segmentation.

1.3.1. Benefits of ΔΣ Modulation

One major advantage of using ΔΣ DACs is that they are nearly all-digital, as the only true analog component required is the low-pass filter (LPF). This makes it quite compact when implemented on a chip.

In addition, ΔΣ modulation possesses highly desirable properties in terms of noise-shaping. ΔΣ modulators, much like other sampling systems, run on a clock which use a frequency (known as the Sampling Frequency) that is at least double the maximum signal frequency, though this is typically much larger. This has a desirable consequence on imaging effects; the undesirable frequencies are

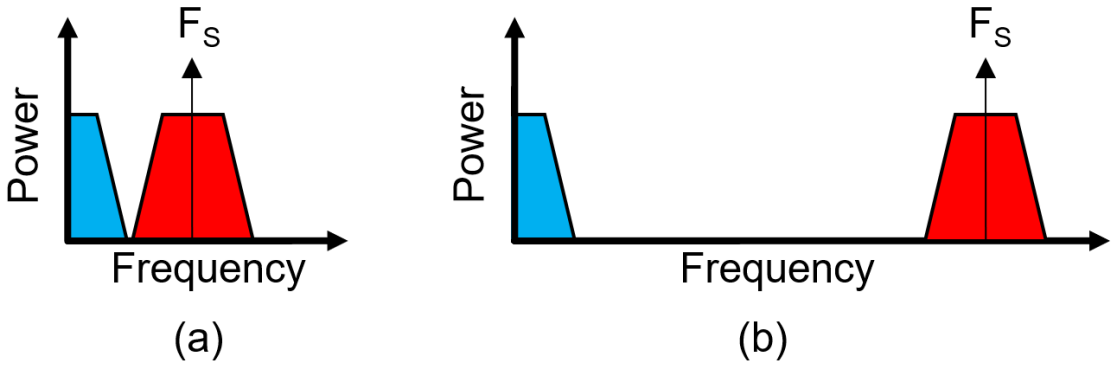

Fig.1.6. A comparison of the distance between the frequency band in blue and the imaging frequencies in red for a low sampling frequency (a) and high sampling frequency (b).

“pushed” away from the signal band, as shown in Fig. 1.6. This means that not only will there be less interference from aliasing within the input spectrum, but the filter used to reconstruct the signal can also be much simpler as it no longer needs a sharp roll-off.

ΔΣ modulation also holds other advantages where noise is not evenly distributed, but rather pushed to higher frequencies. This is covered in greater detail in sub-section 2.2.7. in comparison to other DACs.

1.3.2. Improvements to ΔΣ Modulation using Segmentation

In general, to increase the SNR (and main performance metric) of a ΔΣ modulator, the hardware and area costs must also increase significantly. A higher order modulator will require far more hardware than one with identical data path width but lower loop filter order. The same is true of an increase in the width of the data path.

Seeking to remedy this, the work in this thesis aims to use segmentation to create segmented ΔΣ DACs that uses separate loop filters for the LSBs and MSBs. The idea is that the width of the data path cannot be reduced to meet a certain SNR, but the filters themselves will proportionately require less area to create than a single filter if they are chosen judiciously.

In addition, an increase in resolution for ΔΣ modulation affects more than just the increase in hardware costs. Consider a ΔΣ modulator has N bits in its signal path. Any signal created from a N-bit single word requires a full pass of 2^N samples to complete. This means that as the resolution increases, the length of the output bitstream increase exponentially. This presents problems of latency and bandwidth/filter requirements.

In a 2-segment ΔΣ DAC, however, each segment would have N_C and N_F bits, such that the two add up to the original N. Once the outputs are added together after processing, we can expect a significantly shorter output settling time on account of partitioning of the overall resolution of the ΔΣ modulator.

1.4. Summary

This chapter introduced the core ideas of the thesis, offering a quick introduction to mixed-signal systems, Digital-to-Analog conversion, the history and working principles of ΔΣ modulation, and the principle of segmentation in data converters. This chapter also covered the advantages and limitations of ΔΣ modulation and introduced the motivation behind applying segmentation of ΔΣ DACs.

Chapter 2: Literature Review

2.0. Introduction

This chapter serves as a review of literature and topics relevant to fully understanding this thesis. Firstly, the metrics relevant to gauging the performance of DACs will be covered. Secondly, various DAC architectures are explored so as to make a point of comparison to ΔΣ DACs which will be featured prominently in the rest of the thesis. Thirdly, literature concerning various topologies for modulator realization will be covered. Lastly, relevant literature concerning the use of segmentation in ΔΣ modulation is presented, so as to better understand the purpose and scope of the work in this thesis.

2.1. DAC Characteristics

This section covers the essential metrics used to gauge the performance of a DAC. To start, the metrics as they pertain to the DC characteristics are established. Following this, the analog channel performance metrics will be developed on. Most of the information covered here is taken from [4].

2.1.1. DC Performance Metrics

While the thesis primarily concerns analog channel measurements, a working DAC must meet certain minimal DC requirements to be considered viable. The description of DACs provided in subsection 1.2.1. makes some assumptions that are rarely true in practice. Often, DACs have several limitations on linearity, as explained here.

2.1.1.1. Lines of Comparison

To properly understand the non-idealities characterized by the measurements that will be elaborated on in this sub-section, one must have proper points of comparison. Once data from a DAC is collected, it can be compared against the ideal line to observe how it deviates. Alternatively, one may use different comparison lines; The best-fit line (approximates the data such that total deviation is minimized) and the endpoint-to-endpoint line (approximating the data by interpolating the endpoints) may be used instead, as shown in Fig. 2.1.

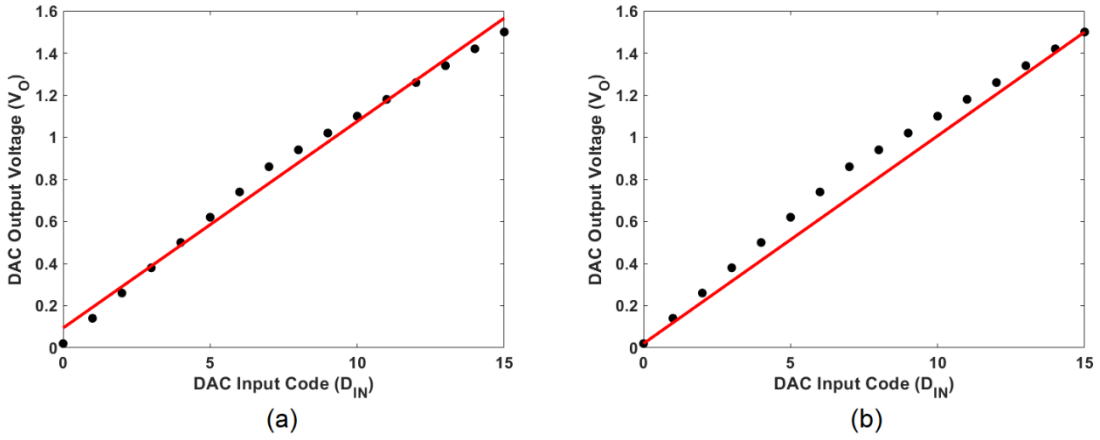

Fig.2.1. A transfer curve of a non-ideal 4-bit DAC with a best-fit line (a) and endpoint-endpoint line (b) both in red.

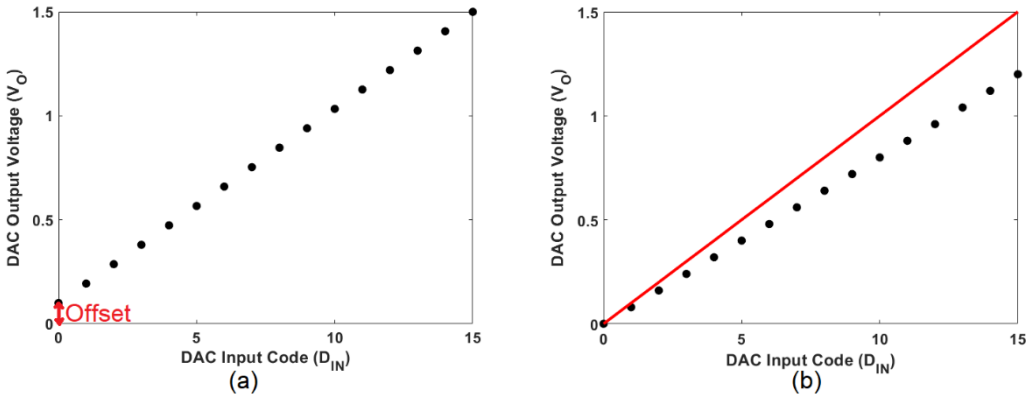

Fig.2.2. An example of offset (a) and gain error (b) as compared to the ideal line in red

2.1.1.2. Offset and Gain Error

Recall the transfer curve for an ideal 4-bit DAC, as shown in Fig. 1.3. Because the initial code (corresponding to zero) corresponds to an output voltage value of 0, and so an ideal converter is said to have an offset error of 0. In a non-ideal DAC, this output voltage's value is typically non-zero and is defined as the offset, as shown in Fig. 2.2. Offset can be defined as the difference between the measured voltage and some baseline; if not the ideal line, then either the best-fit line or the endpoint-to-endpoint line.

Similarly, gain error compares the gain G_{DAC} from the ideal DAC transfer characteristic against that same gain value derived from either the endpoint-to-endpoint line or the best-fit line. The gain derived this way, rather than at each code transition, is rather insensitive to any one code's location and deviation from the ideal. Because of this, the gain from one of the previously mentioned lines creates the truest representation of the DAC's gain. Gain error is typically expressed as a percentage defined as

$$Gain\ Error\% = \frac{G_{DAC} * G_{IDEAL}}{G_{IDEAL}} \times 100\%. \qquad (2.1)$$

2.1.1.3. Monotonicity

An ideal DAC is monotonic, as the output voltage derived from any one code is greater than the output voltage produced by the input code immediately preceding it. Denoting the output voltage of a code *i* as *S(i)*, and the next code's

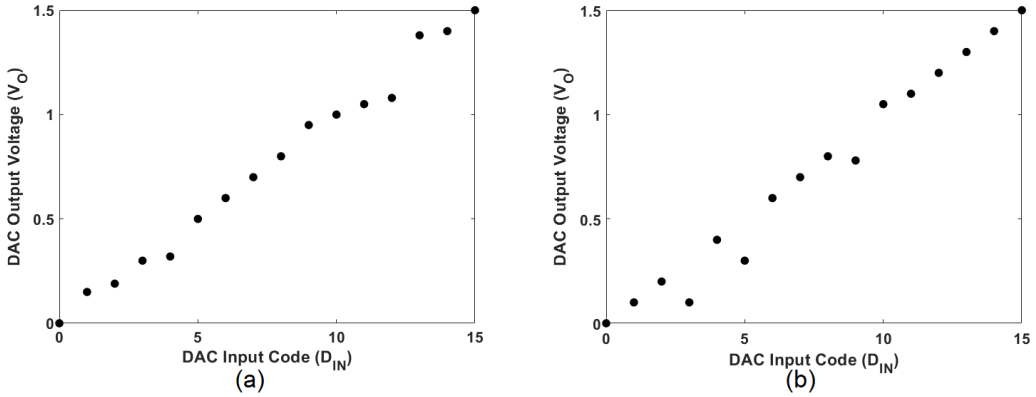

Fig.2.3. Example transfer curves of a monotonic DAC (a) and non-monotonic DAC (b)

output as *S(i+1)*, one can take the discrete first derivative of the transfer curve, denoted as *S'(i)* from the formula

$$S'(i) = S(i+1) - S(i). \tag{2.2}$$

For a DAC to be monotonic, the sign of S'(i) must be the same for any given code *i* (i.e. all positive for a rising ramp input or all negative for a falling ramp input). A transfer curve for a monotonic and non-monotonic DAC can be found in Fig.2.3.

2.1.1.4. Differential Nonlinearity

One limitation of the previously mentioned metrics is that they characterize a DAC with a single value taken from an average. Notice that in the previous examples concerning monotonicity, the step sizes are not uniform. Two DACs may end up with an identical gain value as derived from their best-fit lines, but different transfer curves, as shown in Fig. 2.4. Note that in (a) the values fall within the ideal range, unlike (b) where the endpoints fall outside of it.

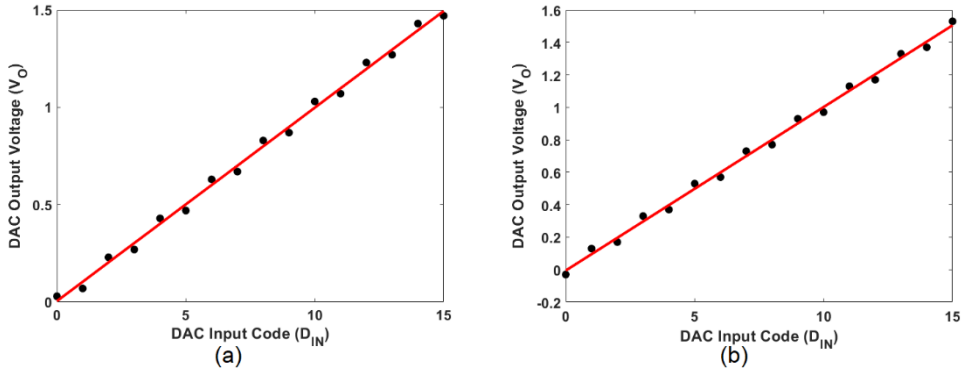

Fig.2.4. An example of two different transfer curves with identical best-fit lines.

Differential non-linearity (DNL) is a figure of merit specifically used to address this issue, as it is used to describe the uniformity of the LSB step sizes between adjacent DAC input codes. DNL represents the error in each step size, expressed in fractions of LSBs. It is computed in a manner not dissimilar from the discrete first derivative, normalizing the result as follows

$$DNL(i) = \frac{S(i+1) - V_{LSB}}{V_{LSB}} LSB. \quad (2.2)$$

As mentioned previously, it is important to keep the V_{LSB} value consistent, choosing either the ideal value, or the one derived from one of the linear approximations (endpoints or best-fit).

2.1.1.5. Integral Nonlinearity

Integral nonlinearity (INL), much like DNL, is a figure of merit that results in a curve that provides a fuller understanding of a DAC's transfer characteristic. INL is a comparison of the actual DAC transfer curve and one of the three previously mentioned lines; ideal, endpoint-to-endpoint, or best-fit. More specifically, INL is the subtraction of the reference line from the DAC's transfer curve, normalized to be expressed as a fraction of the average LSB step size, or

$$INL(i) = \frac{S(i) - S_{ref}(i)}{V_{LSB}} LSB. \quad (2.3)$$

It is worth noting that while the above formula describes the absolute error if using the ideal line, the INL curve is the integral of the DNL curve, thus its name. DNL is a measurement of the error in step sizes from one code to the next, while INL is the accumulated error for the step sizes.

2.1.2. Analog Channel Performance Metrics

The bulk of the thesis is focused on the noise-shaping capabilities of ΔΣ modulation, and as such this subsection introduces some of the essential performance metrics and sources of non-idealities relevant to ΔΣ modulation.

2.1.2.1. Noise

All practical circuits generate some amount of noise. The source of this noise may be thermal, in the case of resistors, 1/f noise for CMOS transistors, or this noise may be generated from quantization for ADCs and DACs. For the purposes of this thesis, noise will be defined as any signal component other than the primary input test signal.

2.1.2.2. Signal-to-Noise Ratio

The primary figure of merit that will be used for characterizing the modulators in this thesis is the Signal-to-Noise ratio (SNR). It is the ratio of the input signal (typically in volts RMS) and the total combined RMS voltage of all the noise sources (frequencies that are do not correspond to the input). Typically, SNR is described in dB as

$$SNR = 20log_{10}\left(\frac{S}{\sqrt{N_1 + N_2 + \cdots + N_X}}\right), \quad (2.4)$$

where S corresponds to the signal, and all N values are the various noise sources.

2.1.2.3. Quantization Noise

In ΔΣ converters, one inherent source of noise is quantization. Whenever a signal passes from the digital to analog domains and vice versa, it is not perfectly recreated. If one were to look at the mapping between digital codes and the real-world voltages they represent, there is a loss of information within the realm of LSBs, as shown in Fig. 2.5.

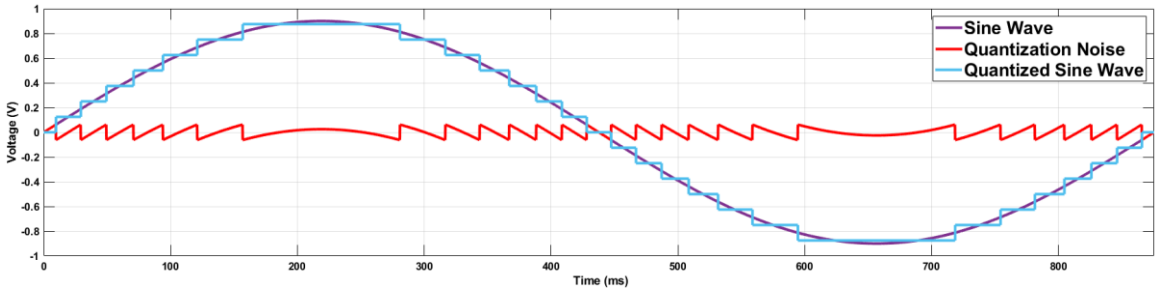

Fig.2.5. An example of quantization and the effective noise it creates.

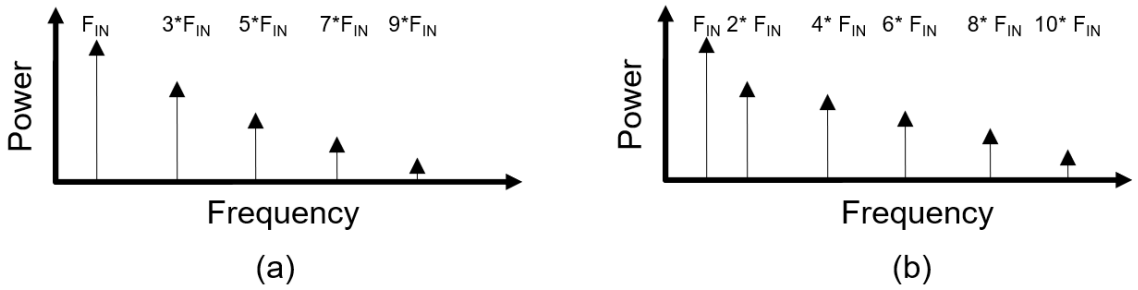

Fig.2.6. An example of two output frequency spectra displaying symmetric harmonic distortion (a) and asymmetric harmonic distortion (b).

2.1.2.4. Harmonic Distortion

Whenever a signal passes through a non-linear circuit, harmonic distortion arises. The output spectrum of a nonlinear circuit does include the frequency components injected at the input, but also frequencies that are integer multiples (harmonics) of those same input frequencies, as shown in Fig. 2.6. Harmonic distortion, if tested on its own, is typically tested with one test frequency at a time. The input test tone's frequency is often referred to as the fundamental tone. Symmetric distortion is called so as it is symmetrical about the x-axis, generating odd harmonics. Conversely, asymmetrical distortion (typically clipping on only one portion of the waveform) adds in even harmonics.

2.1.2.5. Intermodulation Distortion

Intermodulation distortion, much like harmonic distortion, involves the injection of output tones that were not present at the input. The main difference between the two is that intermodulation is caused when multiple tones are present at the input. The frequencies may be any sum or difference of any integer multiples of the input tones, as shown in Fig. 2.7.

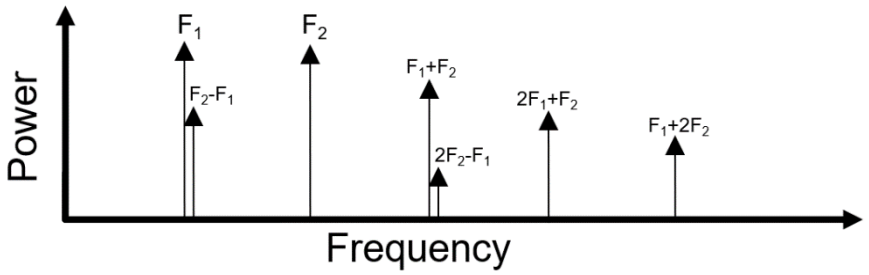

Fig.2.7. An example of Intermodulation Distortion.

For any two input tones F_1 and F_2 from a multi-tone signal, distortion components of any frequency satisfying

$$|q \times F_1 \pm p \times F_2| \tag{2.5}$$

may appear. The variables *q* and *p* are both integers. For intermodulation defined as nth order, the sum of *q* and *p* must be equal to *n*, or

$$q + p = n \tag{2.6}$$

2.1.2.6. Imaging

Imaging is a type of distortion that can occur in sampled systems, which makes this quite relevant to the thesis. As a DAC uses a sampling frequency F_S, an input signal with frequency F_t is reconstructed at the output, several image tones are introduced. These tones, denoted as F_{image} will appear the frequencies satisfying

$$F_{image} = n \times F_s \pm F_t \tag{2.6}$$

An example of imaging is shown in Fig. 2.8. As one can surmise, aliasing can be quite an issue if the sampling frequency is quite low or close to the input signal frequency. Oversampling can aid in overcoming this.

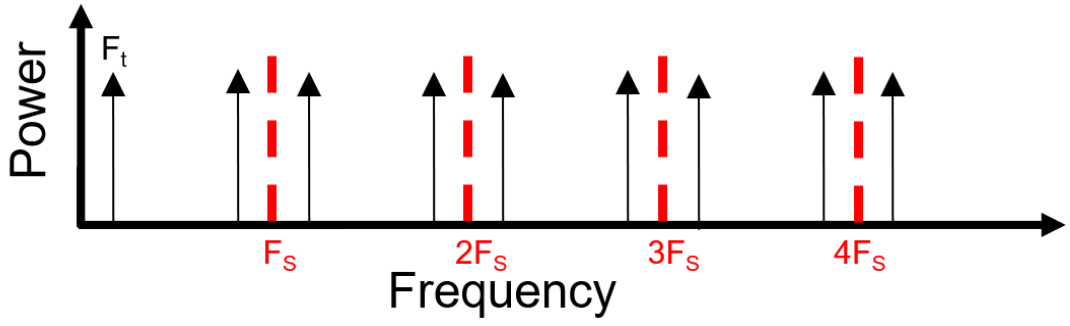

Fig.2.8. An example of imaging

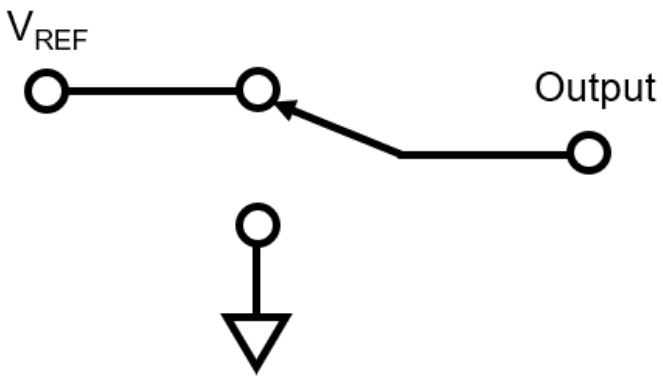

Fig.2.9. The Changeover Switch 1-Bit DAC: (Single-Pole, Double Throw, SPDT)

2.2. DAC Architectures

This section provides an expeditious summary of several common DAC architectures as reported in the relevant literature; namely the benefits and limitations of each architecture are presented and compared. It should be noted that much of the information presented here is also presented in [3].

2.2.1. Changeover Switch

The changeover switch is also known as a single-pole, double throw (SPDT) switch. It can be thought of as the simplest DAC as it switches the output between two set voltage levels: typically, either between a reference voltage and ground or between a positive reference voltage and a negative voltage with the same magnitude. Fig. 2.9. shows this switch.

The main advantage of this architecture is that it is quite simple to implement, but as a downside it can't effectively be scaled up to accept a multi-bit input. While quite simple and not needing more discussion in this subsection, the changeover switch is used as a building block in many other DACs.

2.2.2. Kelvin Divider DACs

The Kelvin Divider DAC, named after Lord Kelvin, is also called the string DAC. Aside from the changeover switch discussed previously, this is the simplest DAC structure (or simplest altogether if considering DACs that accept digital words). A string DAC is composed of 2^N resistors of equal resistance, and an additional 2^N switches. These are allocated such that there is one for each node

of the chain and output. An example of a string DAC is shown in Fig. 2.10. As one can see, the DAC also includes some digital circuitry called a decoder, which takes the N bit input and converts it into 2^N separate signals of which only one is active. This corresponds to the value of the N-bit input decoded, and as such returns the output voltage.

The architecture itself is quite simple, and the digital circuitry involved in the decoder is quite cheap to implement. In addition, the voltage output is inherently monotonic, as even a short-circuit in one of the resistors cannot cause any particular output code's voltage to exceed the output of the following code.

The major downside lies in the heavy use of analog components. For a linear increase in resolution, one must incur an exponential cost in components. Analog components such as resistors can take up a considerable amount of space and generate a non-negligible amount of heat if they become numerous. Being so dependent on a chain of resistors means that any efforts at trimming said resistors is costly and impractical. This is further exacerbated by their small size.

2.2.3. Thermometer DACs

Thermometer DACs are also called Fully-Decoded DACs, and are not too dissimilar to string DACs in principle. Once more, an N-bit code serves as the

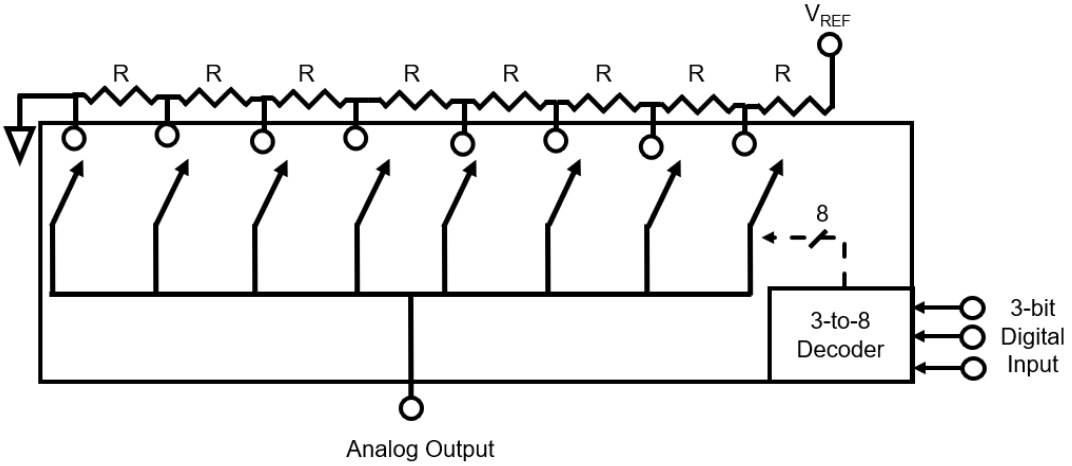

Fig.2.10. Kelvin Divider/String DAC

input which is passed through a decoder to provide the output. However, the main difference lies in the fact that for whatever input code is used, a number of switches corresponding to that input will be activated. Fig. 2.11. (a) shows a thermometer DAC. Notice here that the number of elements in the chain in 2^N-1 rather than 2^N.

The primary advantage of such an architecture is similar to that of a string DAC, as the thermometer DAC is inherently monotonic. In addition, since the current sources (or equivalents) are to be of the same value, nonlinearities are quite small. A small modification to the DAC, as shown in Fig. 2.11. (b) in which the switches steer the current sources either to the output node, or another one which acts as its complement. This is quite useful for high-speed applications, as steering a current from one path to another one causes far less glitch than simply turning it on or off.

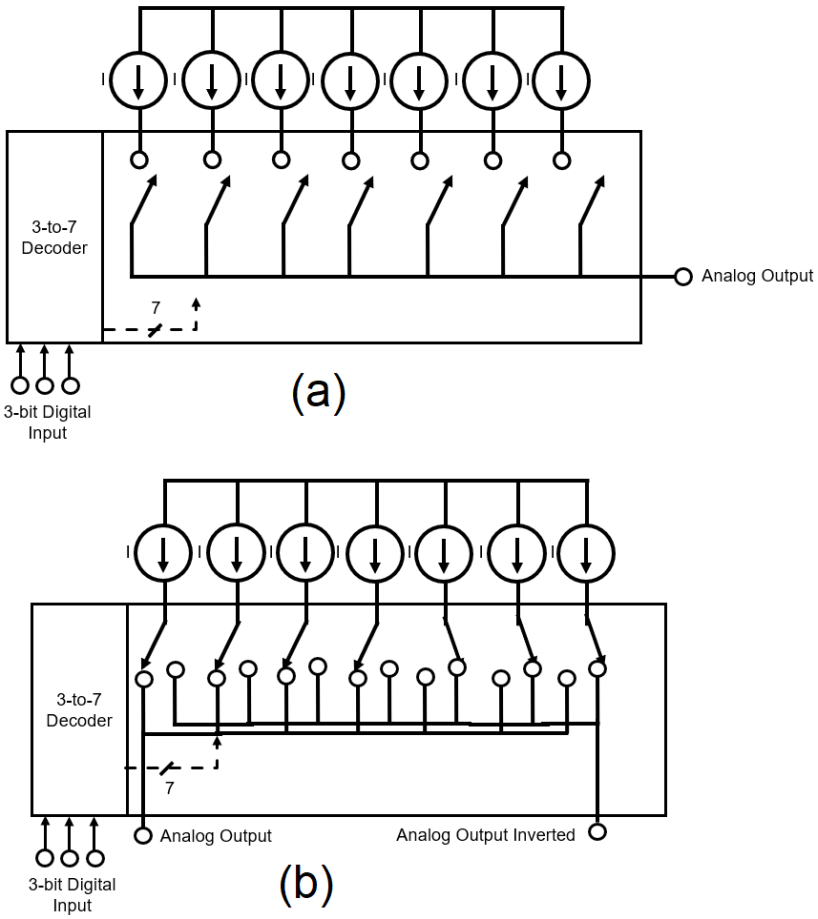

Fig.2.11. Diagram of a thermometer DAC (a) and thermometer DAC with complementary current outputs

The greatest drawback here, however, is the over-reliance on a large number of analog components; A linear increase in resolution N begets an exponential increase in the number of analog components used, and subtracting one as compared to a string DAC does little to alleviate this.

2.2.4. R-2R DACs

The R-2R DAC is among the most common building blocks used for DACs. This specific architecture can be used in either a voltage or current mods, as exemplified in Fig. 2.12. It is evident that the converter is named after the ladder network made up of two different resistor values, in a ratio of 2:1. This architecture was first proposed by Smith in 1953 [5]

The main advantage of the R-2R DAC is that a linear increase in resolution only requires a linear increase in the number of resistors used. More specifically, an N-bit R-2R DAC requires 2N resistors. In addition, their relatively low numbers make trimming more appealing. The R-2R ladder also does not need any extra digital circuitry, such as a decoder. Furthermore, the voltage-mode variation presents a constant output impedance, which eases the process of stabilizing any amplifier connected to the output node.

One major disadvantage of the R-2R ladder DAC lies at the input. The switches are required to operate over the voltage range covering VREF to ground, which can be difficult from a manufacturing standpoint. In addition, the input impedance, unlike output impedance, varies drastically over the span of all the input codes that can be used, and so the input must be driven by a low impedance.

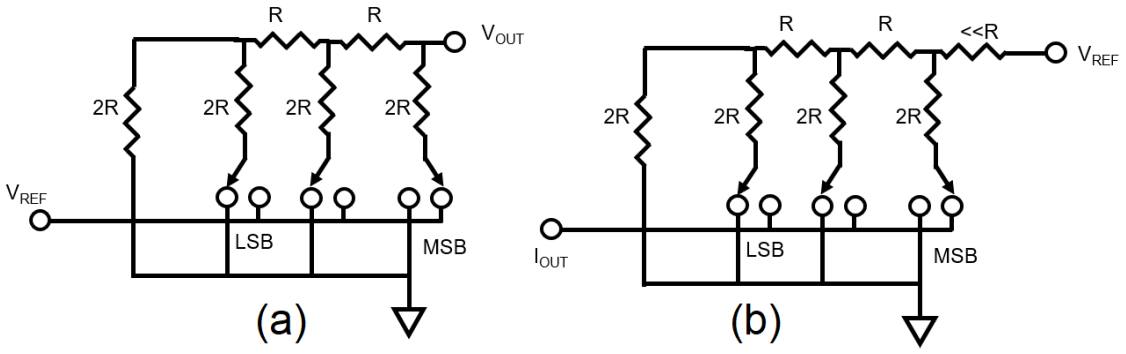

Fig.2.12. R-2R DAC structure for voltage output (a) and current output (b)

To add to this, the value G_{DAC} cannot be readily changed by means of a resistor in series with the voltage reference terminal.

2.2.5. Binary-Weighted DACs

All of the previous architectures use very uniform analog components. As the name suggests, binary-weighted DACs use binary-weighted resistors or sources, which are then used to produce the desired analog value from the digital input code switching those sources on or off, as shown in Fig. 2.13. One of the earliest references to binary-weighted DACs can be found in a patent by Paul M. Rainey, dating back to 1921.

The main advantage of the binary-weighted DAC is how few components are needed for a high-resolution DAC. For an N-bit DAC producing 2^N possible outputs, one needs N sources to create such a DAC.

The greatest downside of binary weighted DACs is the precision required off the components themselves. For example, if one were to take a 7-bit DAC using 7 binary-weighted current sources, the sources responsible for LSB and MSB contributions would need to be matched such that non-linearities are minimized (i.e., a ratio of 64:1). This can quickly become a problem at resolutions of 10 bits or above, as now the ratio between the current sources responsible for

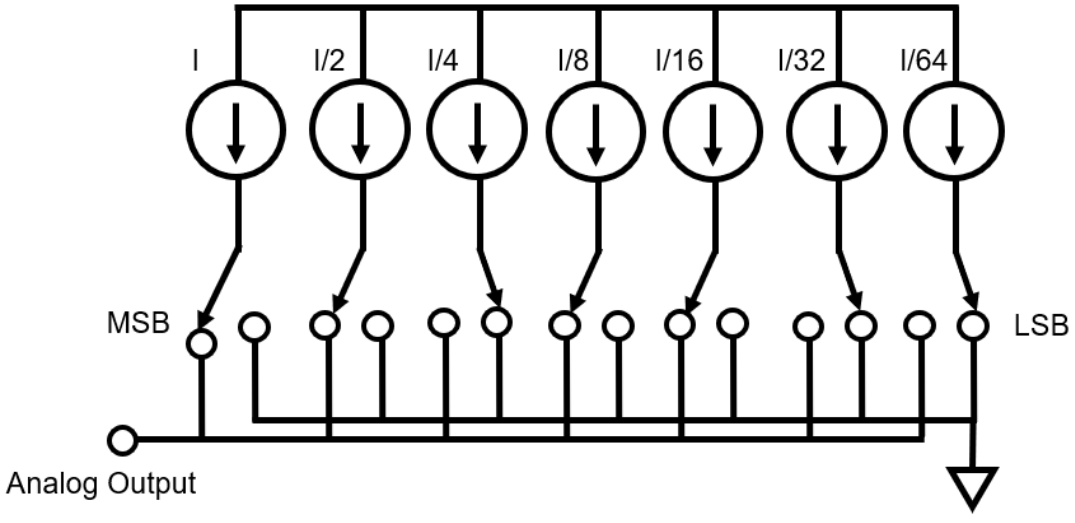

Fig.2.13. Current-Mode Binary-Weighted DAC

the LSB and MSB contributions now result in the former being a fraction of a percent of the latter. This makes components mismatch errors a serious issue.

2.2.6. Pulse-Width Modulation DACs

A pulse-width modulated (PWM) DAC is a type of converter is typically constructed using a counter to generate an output voltage proportional to the digital input word. The entire circuit operates off of a sampling clock, which causes an N-bit counter to cycle through all its possible values, compared to the digital input. If the input is greater than the counter's value, a high voltage is produced at the output. Otherwise, a low voltage is produced. This effectively creates a rectangular wave with 2^N sampling periods each lasting T_S and a duty cycle proportional to the input. This output bitstream must then be passed through a low-pass filter (LPF) to reconstruct the intended. Fig. 2.14 shows a diagram of a PWM DAC, as well as the output bitstream of a 3-bit PWM DAC of an input code of 5.

The primary advantage that PWM DACs provide is the high level of linearity, requiring no calibration. In addition, PWM requires digital components primarily, which are quite economical in their area usage.

The primary disadvantage faced when using PWM is the analog LPF, which consumes a significant silicon area when creating a high-resolution DAC. Since the output is always at a period of 2^N sampling periods, the filter must be able to remove frequencies that are twice as close to the input frequency range with each additional bit of resolution.

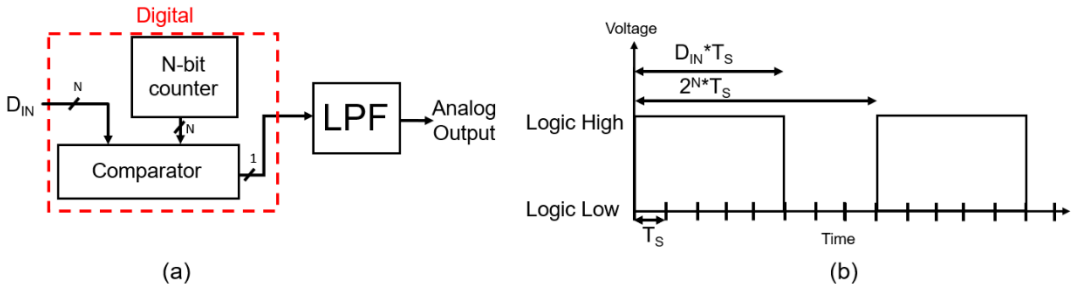

Fig.2. 14 A block diagram of a PWM DAC (a) and an example output bitstream (b) for a 3-bit DAC and input of 5

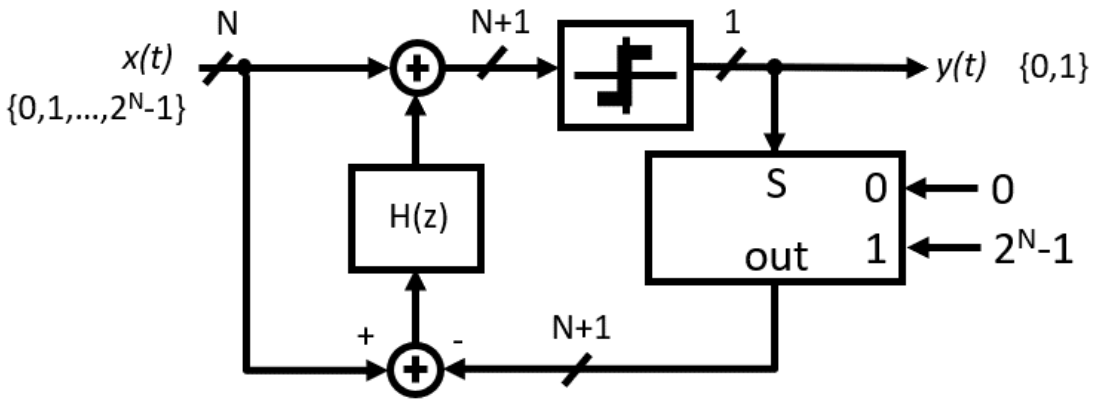

Fig.2.15. Structure of a ΔΣ Modulator

2.2.7 ΔΣ Modulation DACs

Much like PWM DACs, ΔΣ DACs produce a bitstream at the output of a digital circuit which is then passed through a LPF to produce the intended signal. The structure of a typical ΔΣ DAC is shown in Fig. 2.15. The key difference between them is that the bitstream produced by a ΔΣ DAC is the result of pulse density modulation (PDM), or a much more evenly distributed array of pulses. As an example, consider Fig. 2.16. which shows what output is produced by both a PWM DAC and the simplest ΔΣ DAC (with the filter replaced with a delay rather than a filter) for a digital input word equal to half of the input range.

The nature of this output bitstream has several desirable consequences for frequency response. As can be seen in Fig. 2.17. where the red area is quantization noise that remains, with blue being the noise removed by the digital filter. In a typical DAC with sampling frequency F_s, the quantization noise is concentrated at the lower frequencies and so cannot be adequately dealt with using a low pass filter. With oversampling, that same quantization noise is now spread out all the way to $KF_s/2$, drastically reducing the noise observed at the input signal band. Combining this with the noise-shaping properties of the ΔΣ

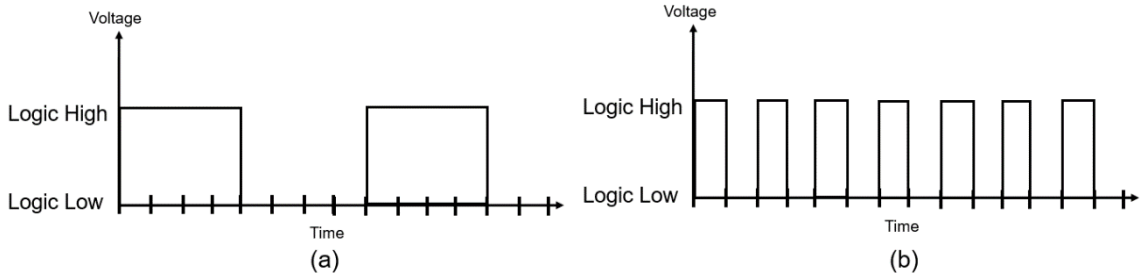

Fig.2.16. A comparison of DAC output bitstreams using N=3 and input of 4 for a PWM DAC (a) and ΔΣ DAC (b)

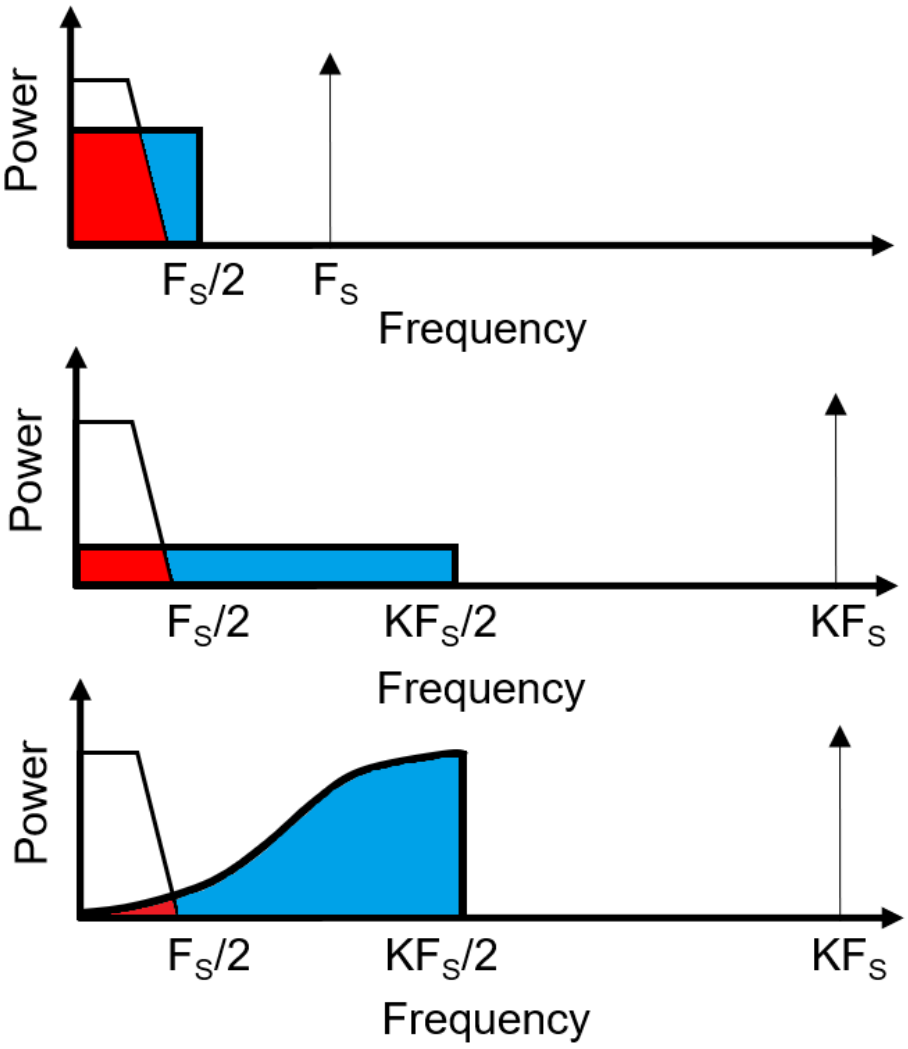

Fig.2.18. Frequency distribution of noise, for an ordinary DAC (a), an oversampling DAC (b), and a noise-shaping DAC (c)

modulator (thanks to the filter in its feedback loop), the noise is further pushed out to the higher frequencies, out of the input signal band. These properties make the ΔΣ modulator a fantastic low-cost, high-resolution option (24 bits). Its low power and low bandwidth make it a suitable for applications in voiceband and general audio signal processing. It also reduces the need for a costly low-pass filter, thanks to the PDM nature of the output.

There still exist limitations to ΔΣ modulation, namely the loop filter's component cost rising exponentially with a linear increase in resolution, on top of the worst-case PDM output being on par with the worst case PWM output. These are the issues that this thesis aims to address.

2.3 Segmented ΔΣ Modulation

Where the two previous sections covered more general information concerning DACs, this section of the second chapter seeks to address topics more specific to this thesis. In the first subsection, a review of the literature concerning segmentation in DACs will be covered, followed by a review of the use of segmentation in ΔΣ converters in the second subsection.

2.3.1. Segmented DACs

A segmented DAC is a Digital-to-Analog converter whose architecture treats various “segments” of the digital input word differently. In some cases, this may entail using one of the DAC architectures for one of the segments, and a different architecture of another to better exploit their various strengths and shore up their weaknesses.

A common use for segmentation uses a thermometer DAC architecture to handle the MSBs (coarse DAC) while employing a binary-weighted DAC to handle the LSBs (fine DAC) as seen in [6, 7]or an R-2R DAC as seen in [6, 7, 8]. This type of architecture provides excellent linearity while mitigating the exponential increase in area consumption. In addition, the second case provides the advantage of only needing one resistor value, making manufacturing much simpler.

It is worth noting that one of the earliest cases of a segmented DAC can be found in [9], back in 1979. The motivation in that case was to make a monotonic DAC with no trimming. By dividing what would have been an ordinary resistor-ladder converter into eight segments, resistors with lower tolerances could be used for each segment for comparable non-linearities to an ordinary R-2R DAC (0.05% versus 0.4%), resulting in a decrease in resistors used by approximately 35%. A diagram of this can be found in Fig. 2.18. The leftmost segment consists of a generator and decoder which create the voltage reference used for the right segment, which uses a binary-weighted segmented 9-bit DAC to create 512 separate voltage levels from the reference fed into it. The “coarse” DAC effectively selects one of 8 current references, and then split using the “fine” DAC, creating

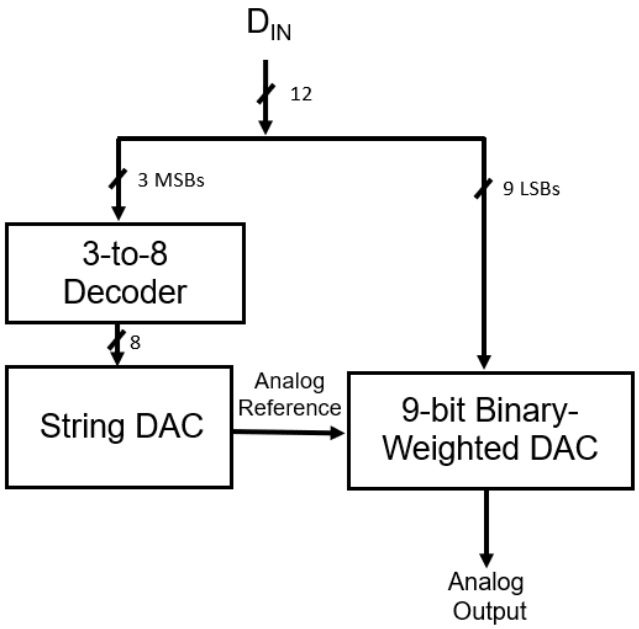

Fig.2.20. A Block diagram of an early segmented DAC, as described in [9]

4096 output voltage levels. The 9-bit DAC is in fact a 4-bit and 5-bit DAC used together, making this example segmented in two ways.

2.3.2. Segmentation in ΔΣ DACs

This subsection presents several examples of segmentation being used in ΔΣ converters and contrasting that with the aim of the structure used later in this thesis. This is done in order to better understand what the thesis is about, as it is vital to understand what this work is not about first.

The use of segmentation in ΔΣ modulators has previously been adopted for the sake of reducing a ΔΣ ADC quantizer to reduce the exponential complexity increase of the Dynamic matching element (DEM) and the digital-to-analog converter needed in its construction [10]. While this holds some similarity, this work uses segmentation for the Delta-Sigma converter as a whole, including H(z). Fig. 2.19. includes a simplified diagram of the converter. Notice here that the segmented DACs are being used within a ΔΣ Analog-to-Digital converter's feedback path, to be fed back into the modulator input. This is in sharp contrast to this thesis, which aims to use segmentation of the modulator itself.

ΔΣ modulator segmentation has also been used to in techniques for segment-mismatch shaping in multi-bit ΔΣ DACs [11] and ADCs [12] significantly

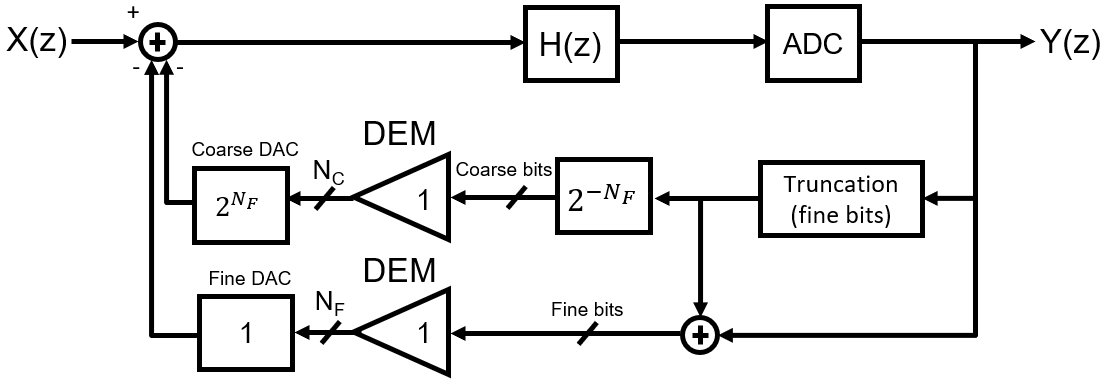

Fig.2.21. Block Diagram of ΔΣ ADC using segmented DAC, as seen in [10]

improving the noise-shaping response of the modulator in question. Once again, Fig. 2.20. and Fig. 2.21. include diagrams of these examples, showing here once again that the segmentation takes place within on the component level rather than a system level. In the case of the DAC, segmentation is used in conjunction with element selection logic (ESL) to transform a multibit ΔΣ modulator output into a vector to control the DACs.

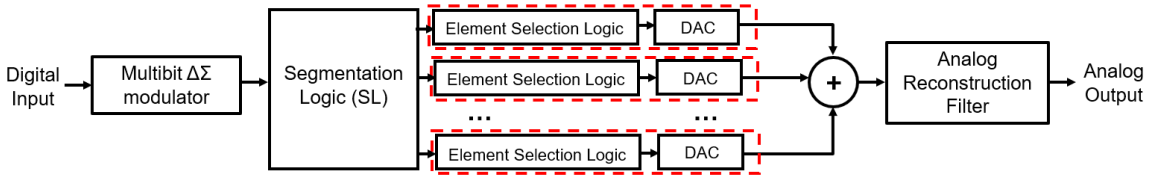

Fig.2.23. A mismatch-shaped segmented multibit ΔΣ DAC, as shown in [11]

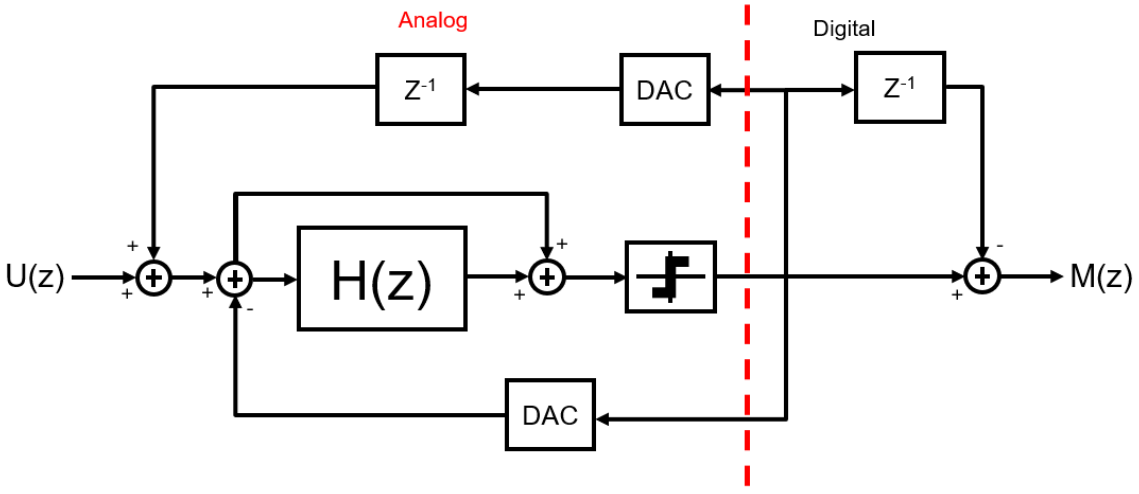

Fig.2.22. Block diagram of the architecture proposed in [12], with mismatch error feedback

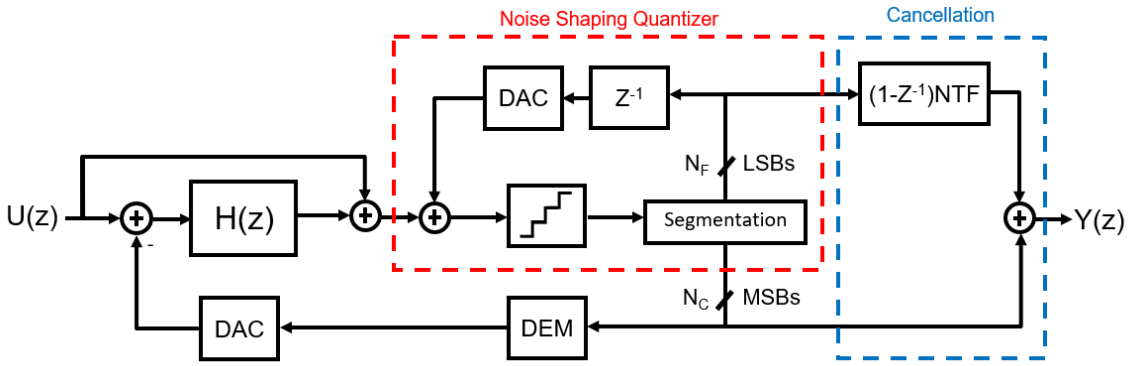

Fig.2.24. Block diagram of the digital noise-coupling technique from [13]

Furthermore, a digital noise-coupling technique for ΔΣ modulators also made use of segmentation [13], but rather than applying this to the input and modulator proper, the quantizer inputs were segmented instead. The MSBs were led into the modulator output directly, while the LSBs were used for the noise coupling as they approximate the quantization noise introduced by the system. It also helped reduce the area of the auxiliary DAC (treating the LSBs) in the digital noise coupling case, which is not too dissimilar from one of the goals of this thesis. Fig. 2.22 demonstrates this structure. A similar process was used in [14], in this case relaxing the bandwidth and settling error requirements, thanks to the uniform linear settling behavior.

It is also worth noting [15] as it pertains to segmentation with after ΔΣ modulation has been performed, which stands in contrast to this work where the ΔΣ modulator is itself segmented. Once again, the structure is shown in Fig. 2.23.

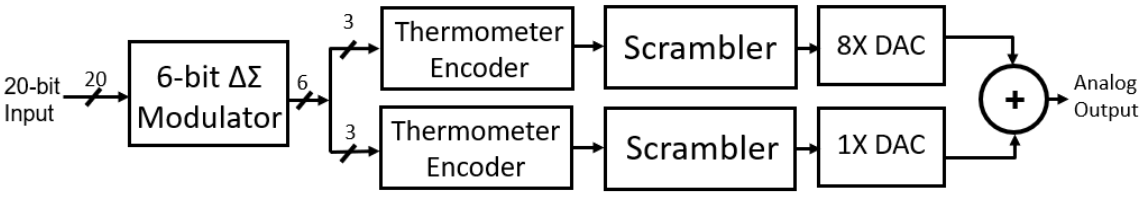

Fig.2.25. Segmentation combined with scrambling, as seen in [15]

2.4 Summary

This chapter reviewed the literature relevant to fully understanding the material, context, and goals of the thesis. The essentials of DAC testing and the metrics to gauge performance were introduced in the first subsection, the second subsection compared various DAC architectures to better understand why ΔΣ was chosen, and in the last subsection literature more specific to segmented ΔΣ modulation was reviewed to fully understand where this thesis stands and what gap in knowledge it seeks to fill.

Chapter 3: Theory and Design of Segmented ΔΣ Modulators

3.0. Introduction

This chapter will introduce the theory and mathematical analysis needed to describe segmented ΔΣ digital-to-analog conversion. In the first section, the theory of ΔΣ is established. In the second section, this theory is used in a mathematical analysis of the 2-segment ΔΣ converter. In the last section, this analysis is extrapolated to a generalized K-segment ΔΣ converter.

Fig.3.1 Typical structure of a ΔΣ-based DAC

3.1. Theory of ΔΣ Modulation

This first section covers the principles behind the operation of DS modulation, in order to lay the foundation for the rest of the thesis. First the basic principles of ΔΣ conversion will be covered, followed by a mathematical analysis of a typical ΔΣ structure, which leads into the subsection concerning the modulator structure that will be used in this thesis. In the last subsection, the performance properties of this structure will be covered.

3.1.1. ΔΣ Converter Principles

The typical structure of an oversampling converter involves three stages as seen in Fig. 3.1. In the case of D/A, the first stage is the ΔΣ modulator, which provides a sequence of pulses (bitstream) as an output. This is the followed by a 1-bit DAC which converts the two logic values into analog quantities, e.g., voltage level which are then fed into an analog lowpass filter (LPF) to reconstruct the original input.

3.1.2. General ΔΣ Modulator Structures

Figure 3.2 shows the signal-flow graph of a typical delta-sigma modulator and the linear model. The main blocks are a pair of linear filters H_1 and H_2, and the one-bit quantizer. The input *x* is filtered by the first linear filter, and then negative feedback is applied to the result by way of the second filter, which produces the input to the quantizer, *e*. The output of the quantizer, *y*, acts as the output, while being filtered by the second filter block, and sent back to the “sigma” block. This creates a negative feedback loop in an ultimately non-linear system. The quantizer can be modelled as a noise source, as is made apparent in Fig. 3.2. (b). With this

new change, the quantizer output *y* is given via a sum of the original quantizer input e and the associated quantization noise q from the new adder. This noise model allows one to perform a Z-transform on the system, giving the equation

$$Y(z) = \frac{H_1(z)}{1+H_2(z)} \cdot X(z) + \frac{1}{1+H_2(z)} \cdot Q(z). \tag{3.1}$$

Thanks to this, there is now a relatively simple way to relate the input, noise, and output in the frequency domain. One can think of the factor being multiplied by X(z) as the signal transfer function (STF)

$$STF(z) = \frac{H_1(z)}{1 + H_2(z)} \tag{3.2}$$

which determines how the original signal is interpreted at the output Y(z), and the factor multiplying Q(z) as the noise transfer function (NTF)

$$NTF(z) = \frac{1}{1 + H_2(z)} \tag{3.3}$$

which similarly determines how the noise is interpreted at the output. From here,

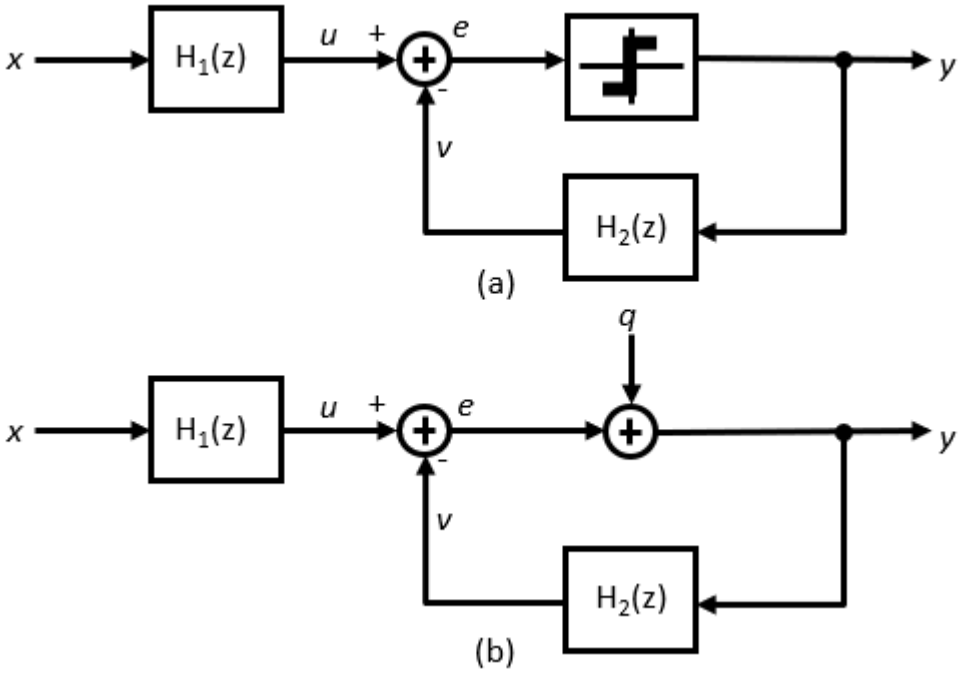

Fig.3.2. Signal flow graph (a) and linear model (b) of a ΔΣ modulator

one can rewrite our original equation for the output as the more general

$$Y(z) = STF(z) \cdot X(z) + NTF(z) \cdot Q(z). \tag{3.4}$$

In the frequency domain, the noise source can be thought of as white noise. This is an approximation, but for any modulator above first order, this is fairly accurate.

From the noise transfer function representation that we've derived in equation (3.4), we can determine that the value of $H_2(z)$ should be large within the relevant bandwidth, in order to mitigate noise at the output. However, if we recall our representation of the signal transfer function in equation (3.3), here $H_1(z)$ must also be large to offset the magnitude of $H_2(z)$ and thus transfer the input signal over to the output. From here, STF(z) should be unity, and NTF(z) should be zero in the ideal case.

For the purposes of this thesis, we will be using a modified model from the one seen in Fig. 3.2., as to have STF at unity regardless of the components used, and with an overall simpler topology. To start, take the definitions of STF as given by equation (3.2). For a unity STF, it is implied that

$$H_1(z) = 1 + H_2(z). \tag{3.5}$$

And while it is possible to simply implement two filter blocks this way, one could instead rearrange the modulator as shown in Fig.3.3 to use a single filter H for both operations. Simply having the input signal lead directly to the output would include quantization noise into the STF, but here it is expressed as

$$STF = \frac{1}{(1 + H) - H} = 1 \tag{3.6}$$

as the input signal both feeds straight to the output and forward into the filter input with opposite sign, effectively cancelling the noise being "injected". This also does not affect the flow of data in the feedback loop for the NTF, as such NTF is expressed as

$$NTF(z) = \frac{1}{1 + H(z)}. \tag{3.7}$$

It has been shown that the structure in Fig. 3.3. is not only much simpler as it only uses one filter, but it can also guarantee unity STF with no dependence on the filter

block. More importantly, because of the position of the filter block in the feedback loop, any poles it has will be zeros of the noise transfer function. We will make good use of this property later to streamline design of modulators.

3.1.3. General Operation Principles

At the core of all ΔΣ modulators lies the principle of negative feedback. This is no different for the structure used in this thesis. The ΔΣ modulator structure seen in Fig. 3.3 will be used unless stated otherwise.

A digital code serves as the input to two difference amplifiers, one into the feedforward path and the other in the feedback path. The feedforward path leads into a comparator, whose output leads into the feedback path via a digital-to-digital converter for scaling purposes. Here a multiplexer with two possible output states of *0* and 2^N*-1* is used to realize this digital-to-digital converter operation. These two integers represent the smallest and largest signal associated with an N-bit data path. This signal then passes into the difference amplifier that leads into the loop filter denoted by *H(z)*. Over time, a bitstream appears at the output which contains a replica of the original incoming signal *x*(t) buried in quantization noise. The signal *x*(t) can be extracted from the bitstream using a lowpass reconstruction analog filter with a corner frequency set to the signal bandwidth of *x*(t). If we think of the modulator output in terms of a density of different code levels (i.e., integers), when scaled by the D/A block, the density can now be expressed in terms of a voltage-level pulse modulated density (PDM) signal. Subsequently, the output of the LPF is

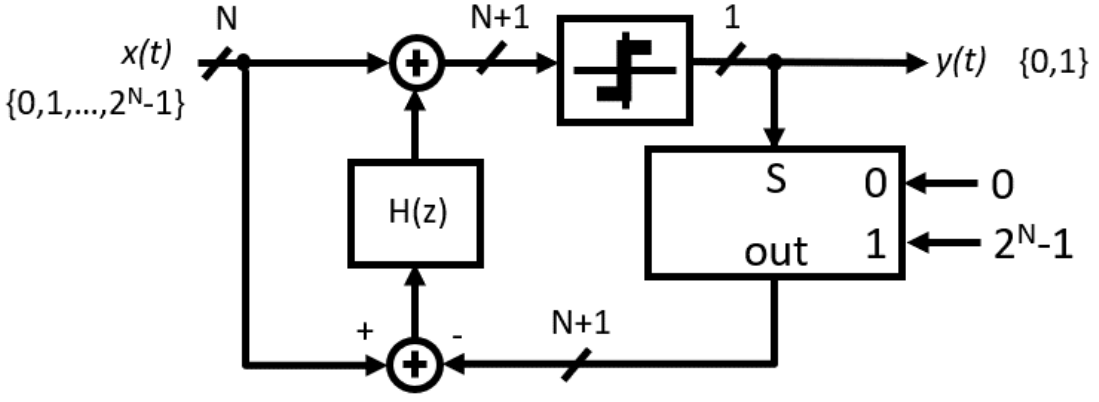

Fig.3.3. Structure of a ΔΣ Modulator with a unity signal transfer function (STF). The number of bits used for each signal path is identified.

the convolution of the input voltage density defined by the term *w*(t) and the impulse response of the LPF denoted by *h*(t). Thus, the output voltage $v_o(t)$ can be expressed as

$$v_o(t) = \int_{-\infty}^{\infty} h(t-\tau)w(\tau)d\tau. \quad (3.8)$$

Generally, the output signal $v_o(t)$ would approximate the average behavior of the ΔΣ output bit stream as a real number function.

ΔΣ modulated output signals operate with a sampling rate f_s significantly higher than the incoming bandwidth of the signal *x(t)*, denoted *BW*. One commonly refers to the ratio of the Nyquist frequency of the sampling process to input signal bandwidth as the "oversampling ratio" denoted as *OSR* and expressed as

$$OSR = \frac{F_S/2}{BW}. \quad (3.9)$$

In this thesis, OSR will be set to 100 unless stated otherwise.

3.1.4. Performance Properties

The performance properties of a ΔΣ modulator are strongly dependent on its order, the amount of quantization noise produced by the quantizer, and the bandwidth and the order of the LPF. As a result of the quantization operation introduced by the quantizer, a source of error is introduced, which will impact the quality of the signal that appears at the output of the LPF. For digital ΔΣ DACs, the quantizer divides the signal range of the incoming digital signal into two halves and identifies which half the instantaneous value belongs in. A two-value symbol set, such as 0 and 1, is used to identify the appropriate region.

For an N-bit path modulator who integer values range from *0* to 2^N*-1*, a *0* or 2^N*-1* valued signal is fed back to combine with the input digital signal to provide corrective action. Consequently, the average power P_Q of the error that is fed back by the quantizer can be stated as

$$P_Q = \frac{\Delta^2}{12}, \tag{3.10}$$

where

$$\Delta = 2^N - 1. \tag{3.11}$$

Adopting a linear perspective of the ΔΣ modulator of Fig. 3.3, where the quantizer is modelled as unity gain structure with an additive noise source *q(t)* having an average power of P_Q. Subsequently, the output behavior of the ΔΣ modulator *y(t)* can be expressed in terms of the input signal *x(t)* and the quantization error signal *q(t)* using the corresponding z-transform of its linear behavior given by equation 3.5, where *Y(z), X(z)* and *Q(z)* are the z-transform of *y(t), x(t)* and *q(t),* respectively. *STF(z)* is the transfer function from the input *X*(z) to the output *Y(z)* when *Q(z)* is set equal to zero, and *NTF(z)* is the transfer function of the quantization noise *Q(z)* to the output *Y(z)* when *X(z)* is set equal to zero.

For the ΔΣ modulator displayed in Fig. 3.3, this modulator guarantees a unity STF, i.e.,

$$STF(z) = 1 \tag{3.12}$$

with no dependence on the loop filter block *H(z)*. The filter block, however, does affect the NTF as follows

$$NTF(z) = \frac{1}{1 + H(z)}. \tag{3.13}$$

Consequently, we will use a loop filter *H(z)* having a low-pass response with a high gain at low frequencies, but a low gain at high frequencies. Ideally, a DC gain of infinite magnitude and a zero gain at infinity.

Adopting a power spectral density (PSD) point view, the output PSD can be described in term of the PSD of the input signal and the quantization noise. Here we

shall use the symbol capital *S* to represent a PSD and a subscript letter to indicate the signal source. Hence, one can write

$$S_Y(f) = |STF(f)|^2 \cdot S_X(f) + |NTF(f)|^2 \cdot S_Q(f). \tag{3.14}$$

However, as the magnitude of STF equals unity and the PSD for the quantizer is equal to the average power P_Q spread over the sampling frequency f_s, one can reduce the previous equation to

$$S_Y(f) = S_X(f) + |NTF(f)|^2 \cdot \frac{1}{f_s}\frac{\Delta^2}{12} \tag{3.15}$$

As the 1-bit D/A operation is simply a one-to-one mapping of the input to its output, one can write the output PSD of the D/A block as

$$S_W(f) = \alpha^2 S_Y(f) \tag{3.16}$$

where α represents the gain of the D/A operation. Consequently, the PSD before the reconstruction filter is described as follows

$$S_W(f) = \alpha^2 S_X(f) + \alpha^2 |NTF(f)|^2 \cdot \frac{1}{f_s}\frac{\Delta^2}{12}. \tag{3.17}$$

As the final output of the digital-to-analog converter, *w(t)* would be convolved with the impulse response of the LPF according to Eqn. (3.3). As the output consists of signal and noise components, the output signal-to-noise ratio (SNR) defined over the bandwidth of the LPF can be described as

$$SNR_o = \frac{\int_0^{BW} S_X(f)df}{\frac{\Delta^2}{12f_s}\int_0^{BW}|NTF(f)|^2 df} \tag{3.18}$$

For an *N*-bit ΔΣ modulator operating under maximum input sinusoidal conditions, the peak-to-peak value would be $2^N - 1$, which is equivalent to an RMS value of $(2^N - 1)/2\sqrt{2}$, or quite simply as $\Delta/2\sqrt{2}$. Thus, one can state the maximum output SNR as

$$SNR_{o.max} = \frac{\left(\Delta/2\sqrt{2}\right)^2}{\frac{\Delta^2}{12f_s}\int_0^{BW}|NTF(f)|^2 df} = \frac{\frac{3}{2}f_s}{\int_0^{BW}|NTF(f)|^2 df} \tag{3.19}$$

Here the maximum output SNR is inversely proportional to the in-band area under the squared-magnitude of the NTF. The smaller this area, the higher the available SNR. It is also interesting to note that the gain α of the D/A does not impact the output SNR.

3.2. Two-Segment ΔΣ Modulator Topology

This section explains the basic principles of segmentation and how this is incorporated into ΔΣ modulation. The principles of operation of a two-segment ΔΣ architecture are then elaborated on. In the last subsection, the equation for maximum SNR is derived.

3.2.1. Segmentation Principles

In many cases, an engineer may be tasked with designing a DAC that satisfies performance metrics so specific that those metrics may not be satisfied with any one architecture. In such cases, one possible solution is to combine multiple DACs into a single DAC with a higher resolution overall to meet those requirements. These DACs need not be of the same types, nor do they need to have matching resolution. This means that any properties affecting the MSB (coarse) and LSB (fine) sections need not be identical, or even influence each other. This has some interesting implications for sigma-delta, as will be discussed in the next sub-section.

3.2.2. Incorporating Segmentation into ΔΣ Modulation

Because of the nature of ΔΣ it must necessarily have a certain number of bits in the signal path, N. It then follows that any signal created from a single digital code requires a full pass of 2^N samples to complete. This means that as the resolution increases, the length of the output string increase exponentially. This is what presents the problems of latency and bandwidth/filter requirements.

In a two-segment system, however, each segment would have N_C and N_F bits, such that the two add up to the original N. Assuming that the outputs are added together after processing, we can significantly shorten the amount of time every "period" takes up, relaxing the frequency constraints. There are added effects to this, including changes to the characteristic "noise shaping" of ΔΣ modulation, discussed later in this work.

3.2.3. General Operation of a 2-Segment ΔΣ DAC

Fig. 3.4. shows a 2-segmented ΔΣ modulator model. It consists of two ΔΣ modulators, denoted as coarse and fine, followed by two 1-bit DAC operations, subsequently, one path is scaled by a factor of 2^{N_F}, while the other is left unscaled. The results are then added to together to form a PDM output signal $w_s(t)$. This signal is then passed through a low-pass analog filter for reconstruction purposes producing an output voltage signal $v_o(t)$.

Critical to the operation of a segment ΔΣ modulator is the front-end bit-partitioning function that separate the incoming N-bit wide digital word x(t) into two parts of word length of N_C and N_F, where

$$N = N_C + N_F. \tag{3.20}$$

The most significant bits (MSBs) of the input digital word are allocated to a signal which is denoted as $x_c(t)$. Likewise, the least-significant bits (LSBs) are allocated to

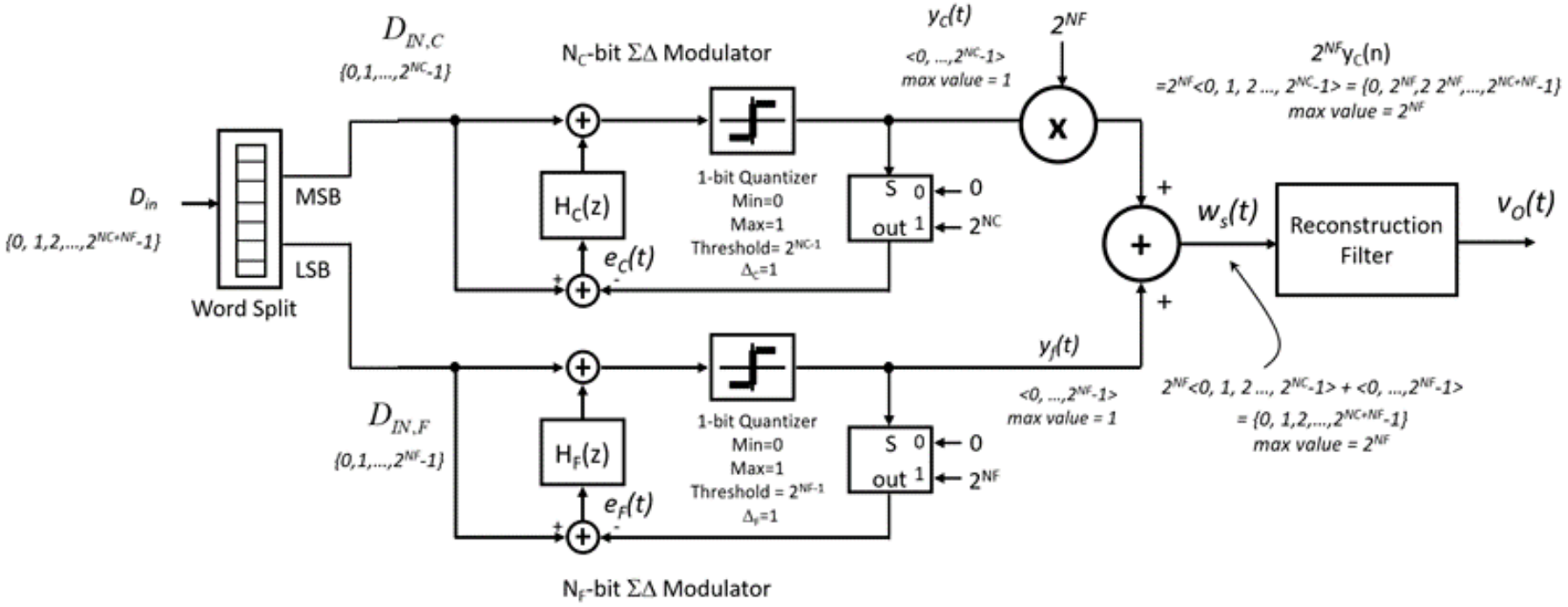

Fig.3.4. An example of a segmented ΔΣ modulator

a signal $x_f(t)$. Fig. 3.5 illustrates the bit partitioning of the input signal $x(t)$ into $x_c(t)$ and $x_f(t)$. Consequently, the input signal range varies from 0 to $2^{N_C+N_F}$-1, the coarse signal ranges from 0 to 2^{N_C}-1 and the fine signal ranges from 0 to 2^{N_F}-1.

Mathematically, the process of separating the input digital word x(t) into the coarse

$$x_C = floor\left\{\frac{x}{2^{N_F}}\right\} \tag{3.21}$$

word, $x_c(t)$ can be stated as

where the operation *floor*{} represents the integer portion of number representing x divided by 2^{N_F}. Likewise, the process of separating the input digital word into its corresponding fine word $x_f(t)$ can be stated as

$$x_F = x - 2^{N_F} floor\left\{\frac{x}{2^{N_F}}\right\}. \tag{3.22}$$

Collectively, the contribution of the two ΔΣ modulation paths results in an output equation whose pulse density contains the input information as follows

$$W_s(z) = 2^{N_F} W_C(z) + W_F(z) \tag{3.23}$$

where

$$W_C(z) = \alpha X_C(z) + \alpha NTF_C(z) Q_C(z) \tag{3.24}$$

and

$$W_F(z) = \alpha X_F(z) + \alpha NTF_F(z) Q_F(z). \tag{3.25}$$

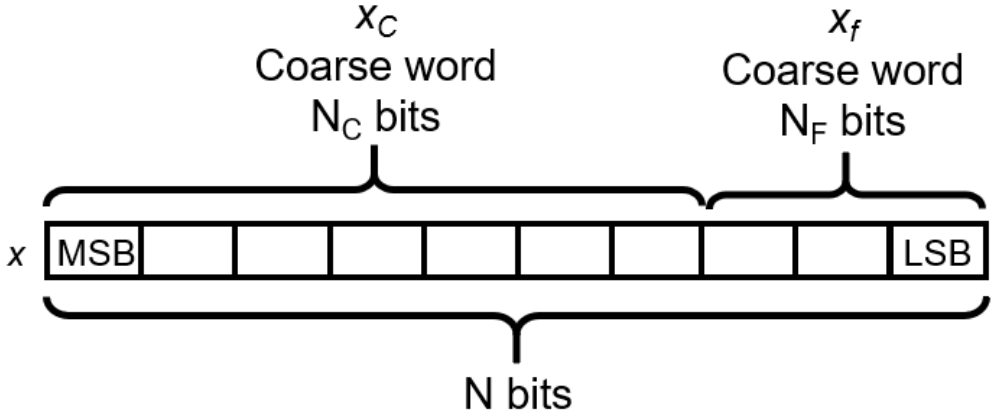

Fig.3.5. Partitioning of the digital word into two parts: Coarse and Fine

With the truncation operation included, the equations (3.24) and (3.25) can be rewritten using z-transforms as

$$W_C(z) = \alpha floor\left\{\frac{X(z)}{2^{N_F}}\right\} + \alpha NTF_C(z)Q_C(z) \tag{3.26}$$

and

$$W_F(z) = \alpha\left[X(z) - 2^{N_F} floor\left\{\frac{X(z)}{2^{N_F}}\right\}\right] + \alpha NTF_F(z)Q_F(z) \tag{3.27}$$

respectively. When substituted back into equation (3.23), one can write

$$W_s(z) = \alpha X(z) + \alpha 2^{N_F} NTF_C(z)Q_C(z) + \alpha NTF_F(z)Q_F(z) \tag{3.28}$$

Here the nonlinear operation of truncation using the flooring operation cancels between the coarse and fine terms. This is true only because the STF of the modulator of Fig. 3.3. is equal to unity across all frequencies. The PSD at the output of the summer can then be written as

$$S_W(f) = \alpha^2 S_X(f) + \alpha^2 2^{2N_F}|NTF_C(f)|^2 \cdot \frac{1}{f_s}\frac{\Delta_C^2}{12} + \alpha^2|NTF_F(f)|^2 \cdot \frac{1}{f_s}\frac{\Delta_F^2}{12} \tag{3.29}$$

where Δ_C and Δ_F are quantization steps, and NTF_C and NTF_F are the noise transfer functions of the ΔΣ modulator in the coarse and fine signal paths, respectively. Here $\Delta_C = 2^{N_C} - 1 \approx 2^{N_C}$ and $\Delta_F = 2^{N_F} - 1 \approx 2^{N_F}$, to correspond to the front-end bit-partitioning procedure. Consequently, equation (3.29) can be written more concisely as

$$S_W(f) \approx \alpha^2 S_X(f) + \alpha^2 2^{2N_F}|NTF_C(f)|^2 \cdot \frac{1}{f_s}\frac{2^{2N_C}}{12} + \alpha^2|NTF_F(f)|^2 \cdot \frac{1}{f_s}\frac{2^{2N_F}}{12}, \tag{3.30}$$

which further simplifies to

$$S_W(f) \approx \alpha^2 S_X(f) + \frac{\alpha^2 2^{2N}}{12 f_s}|NTF_C(f)|^2 + \frac{\alpha^2 2^{2N_F}}{12 f_s}|NTF_F(f)|^2 \tag{3.31}$$

It is readily apparent from this last equation that the coarse NTF makes a larger noise contribution than the fine NTF if one assumes they have similar gains over the passband region of the modulator.

3.2.4. Maximum Output SNR

The output signal-to-noise ratio (SNR) defined over the bandwidth of the LPF can be described as

$$SNR_{o,max,segmented} = \frac{\int_0^{BW} S_X(f)df}{\frac{1}{12f_s}\left[2^{2N}\int_0^{BW}|NTF_C(f)|^2df + 2^{2N_F}\int_0^{BW}|NTF_F(f)|^2df\right]}. \quad (3.32)$$

For an N-bit input word, the maximum sinusoidal condition would have an RMS value of $(2^N - 1)/2\sqrt{2}$ which can be bounded by $2^N/2\sqrt{2}$. Thus, one can state the maximum output SNR as

$$SNR_{o,max,segmented} = \frac{\int_0^{BW} S_X(f)df}{\frac{1}{12f_s}\left[2^{2N}\int_0^{BW}|NTF_C(f)|^2df + 2^{2N_F}\int_0^{BW}|NTF_F(f)|^2df\right]}. \quad (3.33)$$

which further reduces to

$$SNR_{o,max,segmented} \approx \frac{\frac{3}{2}f_s}{\int_0^{BW}|NTF_C(f)|^2df + 2^{-2N_C}\int_0^{BW}|NTF_F(f)|^2df} \quad (3.34)$$

Unlike an unsegmented modulator, the 2-segment modulator has two noise sources, one from the coarse signal modulator and the other from the fine signal modulator. Using this knowledge, one can simplify the equation even further. We will assume that the coarse term is dominant as it as it contributes 2^{2N_F} times the power of its counterpart. For dominance, assuming a ten-times magnitude change, then

$$\frac{2^{2N_C}}{10}\int_0^{BW}|NTF_C(f)|^2df \geq \int_0^{BW}|NTF_F(f)|^2df. \quad (3.35)$$

If this condition is met, for coarse and fine ΔΣ modulators of similar gain and shape, the noise contribution of the coarse ΔΣ modulator will dominate on account of the small weighting factor 2^{-2N_C}, i.e.,

$$\int_0^{BW} |NTF_C(f)|^2 df \gg 2^{-2N_C} \int_0^{BW} |NTF_F(f)|^2 df \tag{3.36}$$

Thus, the output SNR can be further approximated as

$$SNR_{o,max,segmented} \approx \frac{\frac{3}{2} f_s}{\int_0^{BW} |NTF_C(f)|^2 df} \tag{3.37}$$

This equation is identical to equation (3.19), its unsegmented counterpart. If the coarse segment and unsegmented modulators use the same loop filter, then

$$SNR_{o,max,segmented} \approx SNR_{o,max,unsegmented}. \tag{3.38}$$

What this means for a segmented design is that the maximum output SNR can stay relatively unchanged, albeit worsened somewhat by the noise power added by the fine segment.

3.3. Extrapolating Beyond Two Segments

In this section, a process similar to the one from the previous section will be repeated to extend the previous findings to a more general K-segment ΔΣ modulator topology. First the general principles are explained, then the maximum SNR equation is established.

3.3.1. General Operation Principles

The general principles of a K-segment ΔΣ modulator are not dissimilar from the 2-segment case discussed in the previous section. The structure is shown in Fig.3.6. For a modulator with K segments, the first segment, has the N_1 most significant bits from the overall input word of N bit length *x(t)* allocated to it, resulting in the input signal $x_1(t)$. The second segment has the input $x_2(t)$, made up of the N_2 MSBs following the ones from $x_1(t)$. This pattern continues for all K segments. N can be expressed as

$$N = N_1 + N_2 + \cdots + N_{K-1} + N_K. \tag{3.39}$$

Each of the input words pass through their respective ΔΣ modulator and create an output bitstream. As with the two segmented case these then pass through the D/A and get scaled by a weighting factor. The factor Γ^L by which a D/A output of the L^{th} segment is scaled according to

$$\Gamma_L = 2^{N_{L-1}+N_{L-2}+\cdots+N_{K-1}+N_K}. \tag{3.40}$$

Notice that setting K=2 leads to repeating the findings from the previous section. Using the same process from the 2-segment case here, one can write the scaled sum of the segment outputs as

$$W_s(z) = \alpha X(z) + \sum_{L=1}^{K} \alpha \Gamma_L NTF_L(z) Q_L(z), \tag{3.41}$$

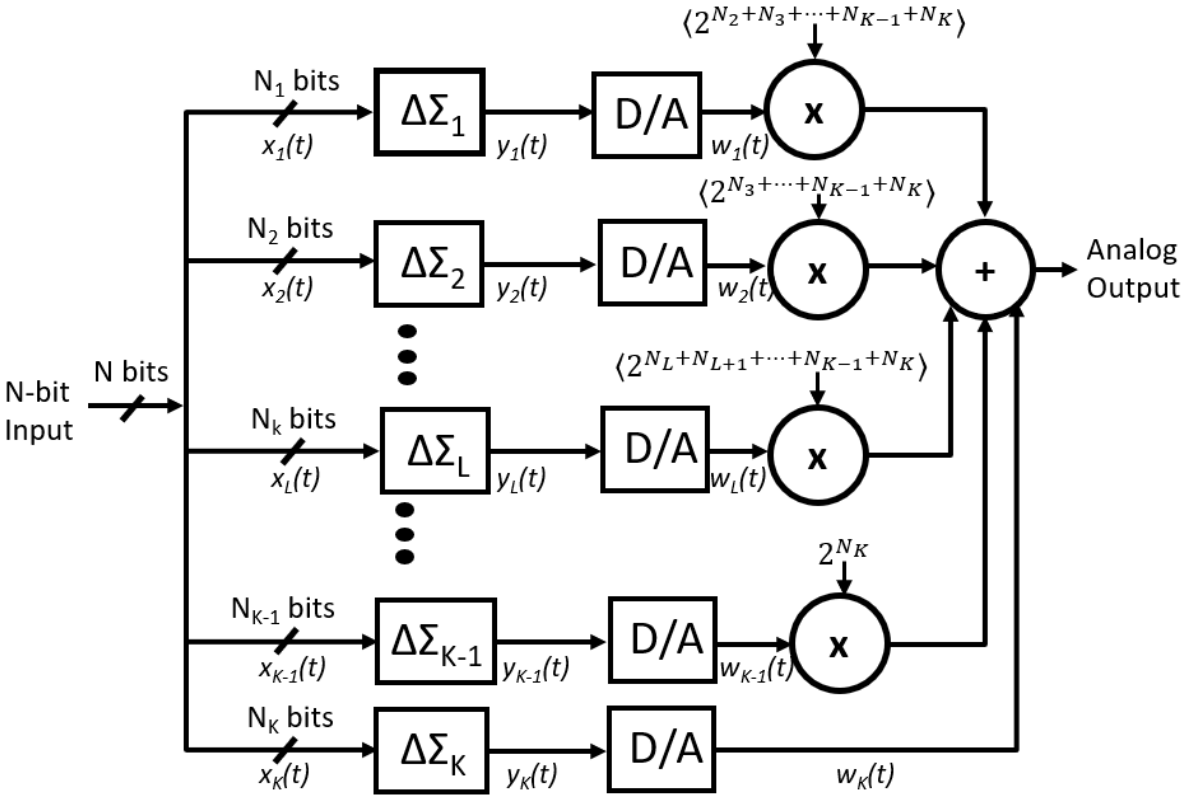

Fig.3.6. A block diagram of a K-segment ΔΣ Modulator

which can now be used to find a K-segment equivalent to the PSD equation (3.31) as

$$S_W(f) \approx \alpha^2 S_X(f) + \sum_{L=1}^{K} \frac{\alpha^2 \Gamma_L^2 2^{2N}}{12 f_s} |NTF_L(f)|^2. \tag{3.42}$$

3.3.2. Maximum Output SNR

As with the 2-segment case, here we will calculate the maximum SNR of the K-segment modulator by repeating the steps which lead to equation (3.33) resulting in

$$SNR_{o,max,segmented,K} \approx \frac{\frac{3}{2} f_s}{\sum_{L=1}^{K} \int_0^{BW} (\Gamma_L - \Gamma_{K-1})^2 |NTF_L(f)|^2 df} \tag{3.43}$$

With this, we can surmise a relationship between the noise contributions of the different segments, similar to equation (3.36). We can express this relationship as

$$\int_0^{BW} |NTF_{K-1}(f)|^2 df \gg 2^{-2N_K} \int_0^{BW} |NTF_K(f)|^2 df. \tag{3.44}$$

To put it another way, for any given pair of “adjacent” segments the coarser one must be dominant. Extrapolating this all the way from segment K to 1, the coarsest segment (segment 1) will dominate the noise contribution of the whole converter. As such, the system’s SNR can be approximated as

$$SNR_{o,max,segmented,K} \approx \frac{\frac{3}{2} f_s}{\int_0^{BW} |NTF_1(f)|^2 df}. \tag{3.45}$$

And so, if the segment “coarse” segment uses the same filter as its unsegmented counterpart, we can generalize with

$$SNR_{o,max,segmented,K} \approx SNR_{o,max,unsegmented}. \tag{3.46}$$

Therefore, for an appropriate allocation of input data among the segments, a segmented design can approximate the SNR of an ordinary ΔΣ converter.

3.4. Summary

This chapter provided a mathematical analysis of the segmented ΔΣ architecture, first for a simple 2-segment converter, and then a general K-segment converter. The maximum SNR achievable by these converters was found, along with the design constraints that must be followed to reach this maximum SNR.

Chapter 4: Synthesis and Design of Segmented ΔΣ Modulators

4.0. Introduction

The previous chapter covered the mathematical basis for segmented ΔΣ systems. But the equations alone do not provide a consistent method to make a segmented ΔΣ DAC.

This chapter will build upon those equations to create a method for designing segmented ΔΣ modulators, by defining the constraints that any segmented design must abide by as well as the constraints on the transfer functions making up the core of any given segment. The first section covers the design method for a 2-segment DAC, the second section covers the method for a generalized K-segment DAC, and the final section covers two running examples which will appear later in the thesis.

4.1. Design method for 2-Segment ΔΣ Modulators

In the previous chapter, we've seen how a segmented ΔΣ DAC's maximum SNR can be found from the responses of its constituent modulators, and that this figure is almost entirely determined by the response of the coarse segment. This, however, is not particularly helpful for an engineer that wishes to create a segmented ΔΣ DAC from specifications that they are given. This section in particular covers the steps needed to properly design a 2-segment ΔΣ DAC from a given set of requirements.

4.1.1. Noise Constraints for a 2-segment ΔΣ DAC

To begin designing a 2-segment ΔΣ DAC from a set of constraints, one must first begin by designing its unsegmented equivalent. Recall equation (3.33), which states the

$$SNR_{o,max,segmented} = \frac{\int_0^{BW} S_X(f)df}{\frac{1}{12f_s}\left[2^{2N}\int_0^{BW}|NTF_C(f)|^2df + 2^{2N_F}\int_0^{BW}|NTF_F(f)|^2df\right]}. \quad (4.1)$$

One can calculate the maximum SNR from the result of the coarse and fine modulator NTFs. But we found that the fine segment's NTF could be ignored altogether if the coarse segment is found to be dominant, as defined in (3.35). From this, we can write the equation for maximum SNR as

$$SNR_{o,max,segmented} \approx \frac{2^{2N_F-3}}{\frac{1}{12f_S}\int_0^{BW}[2^{2N_F}|NTF_C(f)|^2\Delta^2]df}. \quad (4.2)$$

If we were to rearrange the terms such that SNR is on the RHS and the integral of the NTF ends up on the LHS, we get

$$\int_0^{BW}[2^{2N_F}|NTF_C(f)|^2\Delta^2]df \approx \frac{2^{2N_F-3}}{\frac{1}{12f_S}SNR_{o,max,segmented}}, \quad (4.3)$$

and if we were to take the f_s term and transfer it to the numerator of the RHS, carry the 2^{2N_F} factor from the RHS over, and simplify again, we obtain

$$\int_0^{BW} [|NTF_C(f)|^2 \Delta^2] df \approx \frac{12 f_S 2^{-3}}{SNR_{o,max,segmented}}. \tag{4.4}$$

This now gives us a constraint on the NTF that we can choose for the coarse section of our design, limited by our target SNR and the sampling frequency to be used. This can be simplified even further, if we recall that Δ in this system would be the change in code to another, or one. In such a case,

$$\int_0^{BW} |NTF_C(f)|^2 df \approx \frac{12 f_S 2^{-3}}{SNR_{o,max,segmented}}, \tag{4.5}$$

or as

$$\int_0^{BW} |NTF_C(f)|^2 df \approx \frac{3}{2} \frac{f_S}{SNR_{o,max,segmented}}. \tag{4.6}$$

For the fine segment, a similar procedure to the previous one can be followed to find a constraint on its NTF, given the sampling frequency and target $SNR_{O,max,segmented}$. Recall equation (3.35), which dictated the condition for dominance of the coarse term, and substitute the integral of the coarse NTF with our condition to obtain

$$\int_0^{BW} |NTF_F(f)|^2 df \leq \frac{1}{\Delta^2} \frac{12}{10} \frac{f_S 2^{2N_F - 3}}{SNR_{o,max,segmented}} \tag{4.7}$$

which can be re-arranged into a simpler form, if we assume that Δ is one and we rearrange terms, we get

$$\int_0^{BW} |NTF_F(f)|^2 df \leq \frac{3}{20} \frac{f_S 2^{2N_F}}{SNR_{o,max,segmented}}. \tag{4.8}$$

There are multiple other ways of writing this, as we could have ignored the 12/10 term altogether, but in this form, we have something more comparable to our coarse segment's expression.

This concludes the first step, which determines the noise constraints for the component modulators of the 2-segment ΔΣ DAC. The next three are outlined in the next sub-section.

4.1.2. Selecting Filters for a 2-segment ΔΣ DAC

The second step needed to design a 2-segment ΔΣ DAC is to find the appropriate NTF for each of the two segment modulators. There are several methods that may be employed, though all of them rely on the placement of poles and zeros to create the NTF. Some methods are outlined in Some methods are outlined in [16], and specialized filter-design software can be found in [17] and [18], which can be used to design Butterworth, Chebyshev, or Elliptic pole configurations for the NTF. Once this step is complete, we can find the filter equation H(z) from the relationship in equation (3.13), reiterated here as

$$\int_0^{BW} |NTF_F(f)|^2 df \leq \frac{3}{20} \frac{f_S 2^{2N_F}}{SNR_{o,max,segmented}}. \tag{4.9}$$

For the third step, one must map this filter function to a lower resolution N_C, to act as the coarse loop filter $H_C(z)$. For the final step, we take into consideration the noise transfer function constraints from equation (3.36) and use them to find an appropriate NTF for the fine section according to

$$\int_0^{BW} |NTF_F(f)|^2 df \leq \frac{2^{2N_C}}{10} \int_0^{BW} |NTF_C(f)|^2 df. \tag{4.10}$$

Here it is assumed that the noise power contributed by the fine ΔΣ modulator will be at least 10 times smaller than that contributed by the coarse modulator after weighting by 2^{2N_C}, so that the final noise profile of the segmented converter will not be too dissimilar from that of the coarse noise profile.

4.1.3. General Form of the Transfer Functions

When designing a response for a ΔΣ DAC, the noise transfer function is usually decided upon first, and the filter block is then derived from that target. In the case of a segmented design, the process follows closely. Consider NTF to be of the general form

$$NTF(z) = \frac{c_N z^N + c_{N-1} z^{N-1} + \cdots + c_2 z^2 + c_1 z + c_0}{z^N + d_{N-1} z^{N-1} + \cdots + d_2 z^2 + d_1 z + d_0}. \tag{4.11}$$

Plugging this into equation (3.13), we can establish that the filter block must be of the form

$$H(z) = \frac{\frac{1 - c_N}{c_N} z^N + \frac{d_{N-1} - c_{N-1}}{c_N} z^{N-1} + \cdots + \frac{d_2 - c_2}{c_N} z^2 + \frac{d_1 - c_1}{c_N} z + \frac{d_0 - c_0}{c_N}}{z^N + \frac{c_{N-1}}{c_N} z^{N-1} + \cdots + \frac{c_2}{c_N} z^2 + \frac{c_1}{c_N} z + \frac{c_0}{c_N}}. \tag{4.12}$$

But this can be narrowed down further. Recall that any ΔΣ modulator that will act as a component for a segmented DAC is time-sampled, and as such will always produce an output after at least a clock cycle's worth of delay. Next, consider that the filter cannot have a direct path from input to output. Seeing as a delay-feedback loop exists here, *H(z)* is thus restricted to being proper; its finite zeros are fewer than its poles. From here, our conditions imply that the NTF's first denominator constant is

$$c_N = 1, \tag{4.13}$$

which simplifies our previous expressions further to

$$H(z) = \frac{(d_{N-1} - c_{N-1}) z^{N-1} + \cdots + (d_2 - c_2) z^2 + (d_1 - c_1) z + (d_0 - c_0)}{z^N + c_{N-1} z^{N-1} + \cdots + c_2 z^2 + c_1 z + c_0} \tag{4.14}$$

and

$$NTF(z) = \frac{z^N + c_{N-1}z^{N-1} + \cdots + c_2 z^2 + c_1 z + c_0}{z^N + d_{N-1}z^{N-1} + \cdots + d_2 z^2 + d_1 z + d_0}, \tag{4.15}$$

which means that as frequency approaches infinity, NTF(z) approaches unity, and *H(z)* approaches zero.

4.1.4. Conditions for Realizability.

For a segmented DSMOD to be realizable, each individual segment must be realizable. Firstly, as with all discrete-time linear systems, the poles must be bound within the z-plane's unit circle. While this may seem like a significant hinderance at first, but because of the freedom we have when designing coefficients, we can readily place the poles such that we can match this criterion.

Secondly, the quantizer which leads into the output is a non-linear element, and as such must be accounted for using non-linear methods [19]. In [16], we find that we can establish absolute stability for the modulator, by looking at the response within the frequency domain; the magnitude of the noise transfer function should not exceed 2.0 over the Nyquist band or

$$\left|NTF(e^{j2\pi f/f_s})\right|_{0<f<f_s/2} < 2.0 \tag{4.16}$$

to put it another way.

A designer must keep in mind, however, that at that boundary, the modulator will only be stable for a limited input amplitude range, and so they should aim to keep the maximum magnitude of the NTF at a safe margin lower than this. Typical values are around 1.6.

Thirdly, the modulator must, of course, be realizable. For the feedback configuration we have chosen, there must necessarily be a unit delay at a minimum. The transfer function chosen must be strictly causal. This means that H(z) must have a denominator of higher order than its numerator, which leads

into the NTF must have matching orders for its numerator and denominator, as well as matching leading coefficients.

And lastly, while not exactly a constraint, it is possible to maximize SNR by manipulating the placement of zeros of the NTF. The zeros of the NTF (also poles of H(z)) should be placed on the unit circle within the relevant frequency band. There is documentation pertaining to this [19] and many tools such as DSMOD, and details will be left to the reader's discretion.

It is crucial to note that it is possible to create an unrealizable segmented modulator from segments that each meet these requirements listed in this section. The reasons for this have more to do with the digital implementation of the modulator, which will be covered in detail in the next chapter.

4.2. Design method for K-Segment ΔΣ Modulators

Designing a K-segment modulator requires a process that is nigh-identical to the two-segment case. This will be re-iterated here.

4.2.1. Noise Constraints for a K-segment ΔΣ DAC

To establish what constraints are imposed for each component modulator in the segmented DAC, one must again use the formulae developed in the previous chapter and extrapolate the results from the previous section to a more general K-segment architecture.

Recall Fig.3.4, referred to as Fig.4.2. here. Note that the coarse segment's contribution is weighted by a factor of two to the power of the number of bits in the finest segment, hence its appearance in the formulae pertaining to noise contribution. This is mirrored in the K-segment case as shown in Fig.4.1, with the one change being that scaling factor must use only the number of bits in all segments finer than the segment in question.

The basis for finding the design constraints for a K-segment modulator is not too dissimilar from how we derived our results for the two-segment case. Recall

$$SNR_{o,max,segmented,K} \approx \frac{\frac{3}{2}f_s}{\sum_{L=1}^{K}\int_0^{BW}(\Gamma_L - \Gamma_{K-1})^2|NTF_L(f)|^2 df}. \tag{4.17}$$

From here, we can build on this and what was found in the previous subsection to derive the equivalent formulas. We can think of segment 1 as the coarse segment, and the agglomeration of segments 2 to K can be thought of as the equivalent of the fine segment. We can use this thought process to “recursively” solve for each NTF, starting from this coarse segment. For this first point, we can write

$$\int_0^{BW}|NTF_1(f)|^2 df \approx \frac{3}{2}\frac{f_S}{SNR_{o,max,segmented,K}}. \tag{4.18}$$

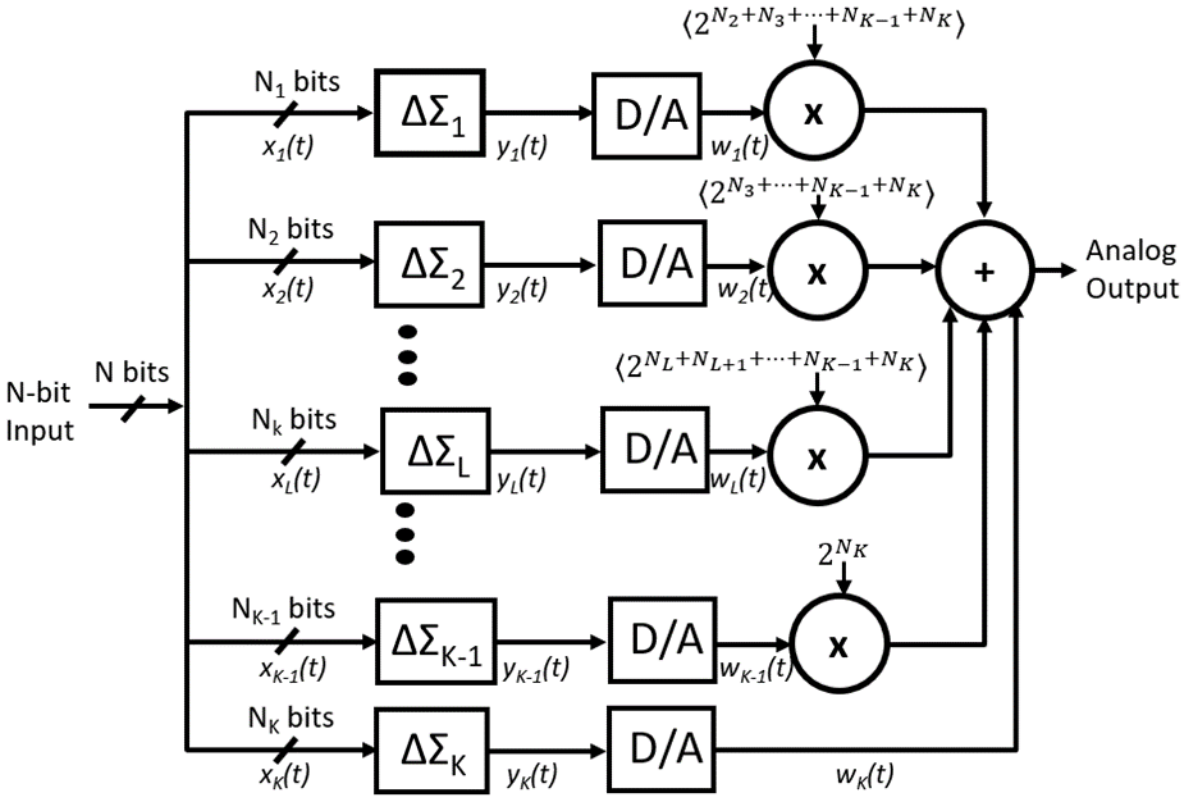

Fig.4.1. A block diagram of a K-segment ΔΣ Modulator

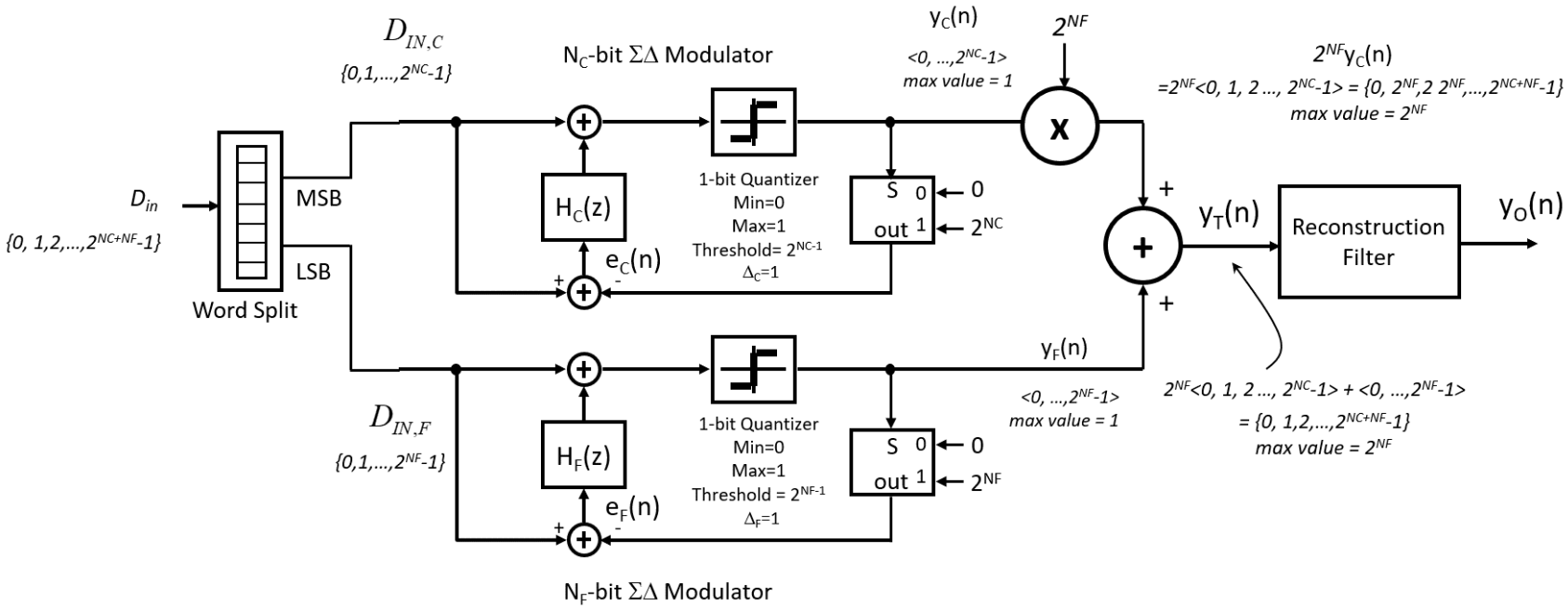

Fig.4.2. An example of a segmented ΔΣ modulator

which leads us to the same equation found in the two-segment case, in other words

$$\int_0^{BW} |NTF_C(f)|^2 df = \int_0^{BW} |NTF_1(f)|^2 df. \tag{4.19}$$

As such, the coarsest segment can have its NTF approximated as that of the unsegmented equivalent. For the “fine segment” (segments 2 through K), we can think of the total BW noise power of the modulator as

$$\int_0^{BW} |\Gamma_F NTF_F(f)|^2 df = \sum_{i=2}^{K} \int_0^{BW} |\Gamma_i NTF_i(f)|^2 df\,. \tag{4.20}$$

While this may be enough for some designers, if we were to follow the same logic of each segment dominating the ones finer than it, we can express the L_2 norm for segment 2 as

$$\int_0^{BW} |NTF_2(f)|^2 df \leq \frac{1}{10}\frac{3}{2}\frac{f_S 2^{2N_1}}{SNR_{o,max,segmented,K}}. \tag{4.21}$$

From here we can find the more general case for any segment k,

$$\int_0^{BW} |NTF_k(f)|^2 df \leq \frac{1}{10^{k-1}}\frac{3}{2}\frac{f_S 2^{2(N_1+\cdots+N_{k-1})}}{SNR_{o,max,segmented,K}} \tag{4.22}$$

It should be noted that the *k-1* term applies to the one tenth term as each segment must dominate the noise response, as compared to the segments “finer” than it. Another way to express this is with the equation

$$\int_0^{BW} |NTF_k(f)|^2 df \leq \frac{1}{10} 2^{2N_{k-1}} \int_0^{BW} |NTF_{K-1}(f)|^2 df \tag{4.23}$$

4.2.2. Selecting Filters for a K-segment ΔΣ DAC

All the information provided in subsection 4.2.1. concerning the selection of filters for a 2-segment DAC applies in the generalized case as well, with a new complication. To reduce hardware costs with segmentation, typically lower-order

(less costly) modulators are used in the “finer” segments, as contribute a lesser portion of the noise thanks to the weighting operation. In addition, a modulator of a higher order at a “finer” segment would not improve the noise response, again due to the arrangement of the segments and the data that they handle.

However, this can be taken too far. Fine modulators with significantly poorer noise responses than the coarse modulators would effectively be injecting quantization noise into the system with little improvement. Therefore, it is considered more sound design-wise to use a “ladder” of modulators, in which each modulator has a lower order than the neighboring one handling the more significant bits. But it must still not be too far off, typically only one or two orders lower.

4.3. Running Examples of Segmented ΔΣ DACs

This section includes two examples of segmented ΔΣ DACs. The first is a 2-segment DAC and a 3-segment DAC. These will be featured again in the next two chapters, where these will be simulated in MATLAB and then mapped on to an FPGA.

4.3.1. 2-Segment Running Example

For this example, we want to create a 2-segment ΔΣ modulator with a signal path that has 16 bits split evenly between the two segments, as illustrated in Fig.4.3. The sampling frequency is 6.28MHz, and the OSR is set to 100. The engineer has a target SNR of 99, or approximately 16.5 effective number of bits (ENOB), as shown in the following equation for ΔΣ converters [3]

$$ENOB = \frac{SNR - 1.76dB}{6.02}. \tag{4.24}$$

From equation (4.6) the given values can be plugged in to find the following equation for the noise level expected out of the converter in the signal band,

$$\int_0^{BW} |NTF_C(f)|^2 df \approx \frac{3}{2} * \frac{f_S}{SNR}. \quad (4.25)$$

Which results in an RMS noise total level of $3.45*10^{-6}$ V. From here the aim is to find a noise-shaping curve that fits the design target as closely as possible. Using MATLAB to run through all the previous equations in conjunction with existing functions pertaining to NTF, a 3rd order inverse-Chebyshev NTF was chosen. More specifically, it can be characterized as

$$NTF_C(z) = \frac{z^3 - 2.9993z^2 + 2.9993z^1 - 1}{z^3 - 2.1658z^2 + 1.6460z^1 - 0.4293}. \quad (4.26)$$

This results in a loop filter described as

$$H_C(z) = \frac{0.8334z^2 - 1.3533z^1 + 0.5705}{z^3 - 2.9993z^2 + 2.999z^1 - 1}. \quad (4.27)$$

The noise transfer function is shown in Fig.4.4. This NTF results in an RMS noise level of $2.10*10^{-6}$ V in the signal band. Now, with equation (4.10), the requirements for the noise transfer function of the fine segment can be calculated using

$$\int_0^{BW} |NTF_F(f)|^2 df \leq \frac{2^{2N_C}}{10} \int_0^{BW} |NTF_C(f)|^2 df. \quad (4.28)$$

This means the total RMS noise from the NTF should be about $7.01*10^{-4}$. It was the decided that the noise transfer function characterized by

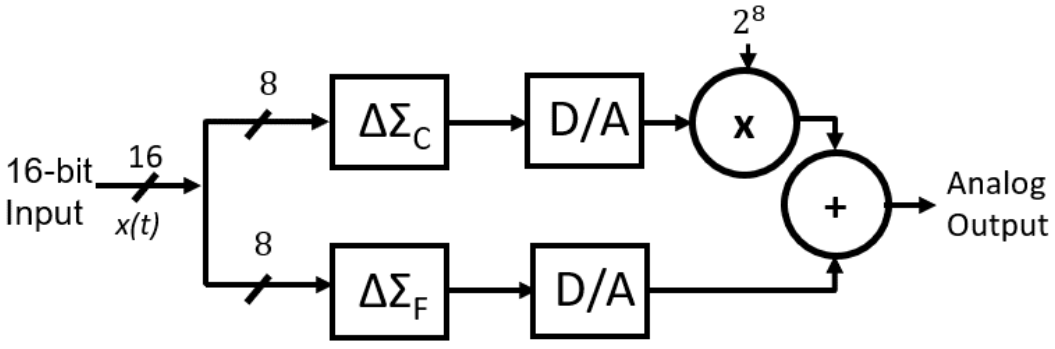

Fig.4.3. Structure of the 2-segment running example

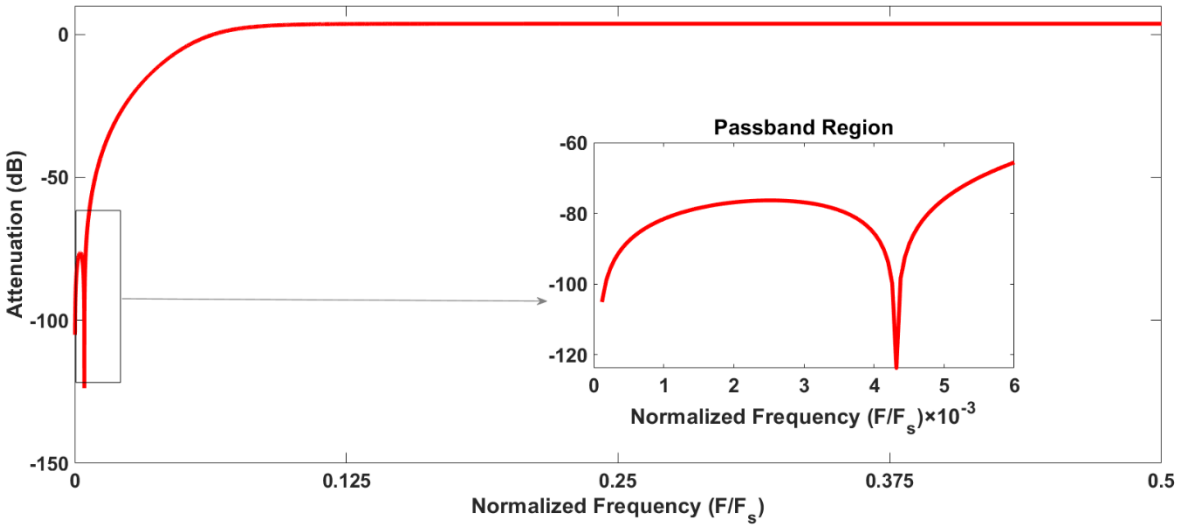

Fig.4.4. The Noise Transfer Function of the Coarse Segment, shown in the Nyquist and signal band.

$$NTF_F(z) = \frac{z^1 - 1}{z^1 - 0.5095}, \tag{4.29}$$

which is a 1st order Butterworth response, would be chosen as the NTF for the fine segment. The filter, therefore, can be characterized by the equation

$$H_F(z) = \frac{0.4905}{z^1 - 1}. \tag{4.30}$$

The noise transfer function for this segment can be visualized in Fig.4.5. The total RMS noise is found to be $3.45*10^{-6}$ V, resulting in an SNR of 99.89.

The results from this example are summarized in Table 4.1, found below. The entirety of the system is summarized in Fig.4.7.

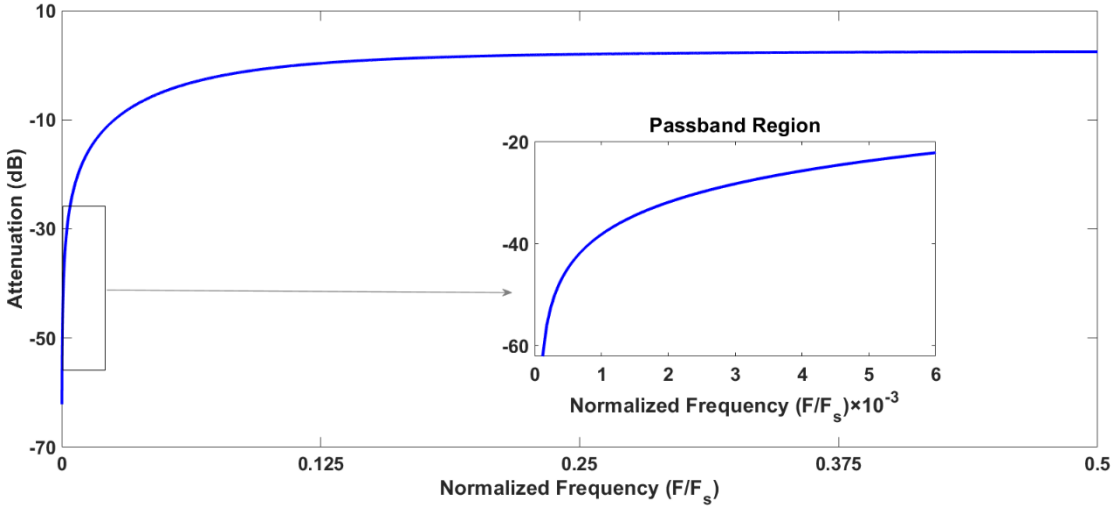

Fig.4.5. The Noise Transfer Function of the Fine Segment, shown in the Nyquist and signal band.

Table 4.1.

A Summary of the Two-Segment ΔΣ DAC

Modulator	Data Allocated	Scale Factor	SNR found	ENOB found
Coarse	8 Bits	2^8	104.371	17.395
Fine	8 Bits	1	53.645	8.941
Overall	16 Bits	N/A	99.894	16.649

4.3.2. 3-Segment Running Example

In this example, an engineer has the task of designing a 3-segment ΔΣ modulator that has 20 bits along the signal path, which will be split in a ratio of 12:4:4, going from coarse to fine. Refer to Fig.4.6 for the structure. Once more, the sampling frequency will be set to 6.28MHz and the OSR will be 100. The target SNR is 120dB.

Starting off with the coarse response, the in-band total RMS noise level should be no higher than $2.23*10^{-7}$. The loop filter was designed MATLAB's functions to find a 4^{th} order inverse Chebyshev noise response which fits the requirements. The noise transfer function for the coarse segment is described with the equation

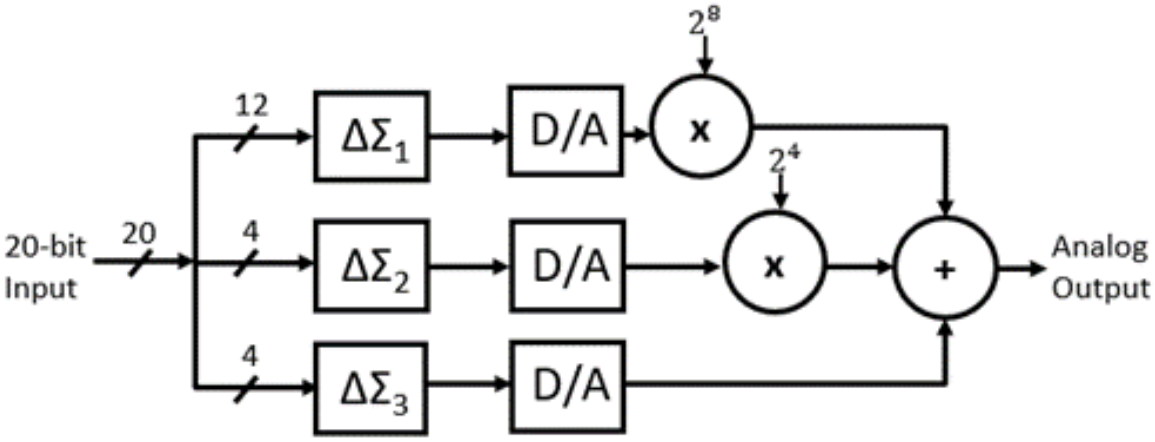

Fig.4.6. Structure of the 3-segment Example

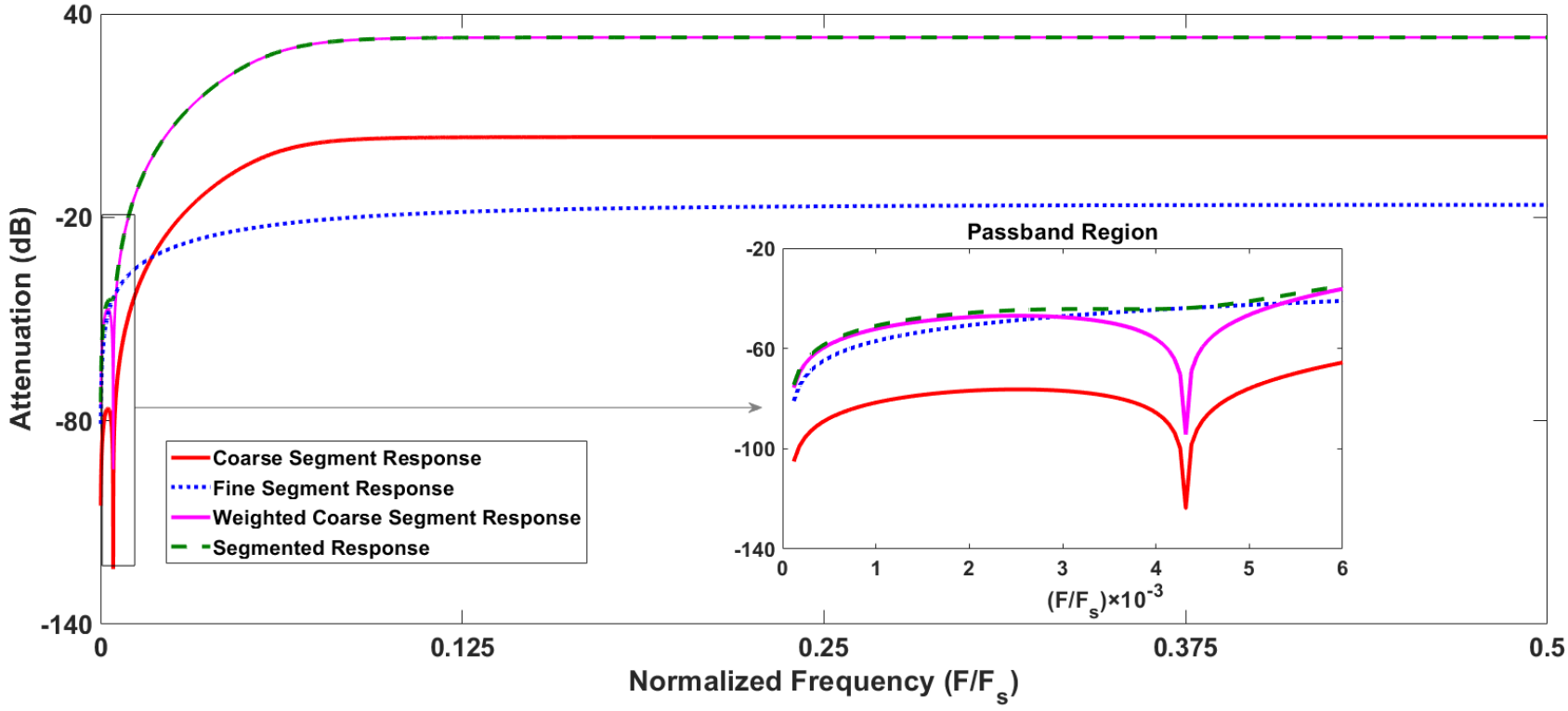

Fig.4.7. The overall input-output magnitude response of the 2-segment example with a view of the passband region together with the coarse, weighted-coarse and fine segment behaviors for comparison purposes.

$$NTF_1(z) = \frac{z^4 - 3.9993z^3 + 5.998z^2 - 3.999z + 1}{z^4 - 3.1427z^3 + 3.7774z^2 - 2.0487z + 1}, \tag{4.31}$$

Which is due to a loop described as

$$H_1(z) = \frac{0.8563z^3 - 2.2207z^2 + 1.9503z - 0.5780}{z^4 - 3.9993z^3 + 5.998z^2 - 3.999z + 1}, \tag{4.32}$$

The response of this modulator is visualized in Fig.4.8.

Using this loop filter produces an RMS noise level of $2.130*10^{-7}$V. This is just below the target, but thanks to the scaling, the remaining $5.0865*10^{-10}$ V_{RMS} is still a significant remainder in the “noise budget” for the DAC.

The second/middle NTF was found using the previous noise that the modualtor could afford to produce, and scaling it as indicated by equation (4.21). This means that the RMS noise produced by this middle modulator can be no greater than $3.334*10^{-5}$ V before scaling. It was determined that a 2nd order Butterworth NTF would be used, with characteristic equation

$$NTF_2(z) = \frac{1z^2 - 2z^1 + 1}{1z^2 - 1.143z^1 + 0.4128}, \tag{4.33}$$

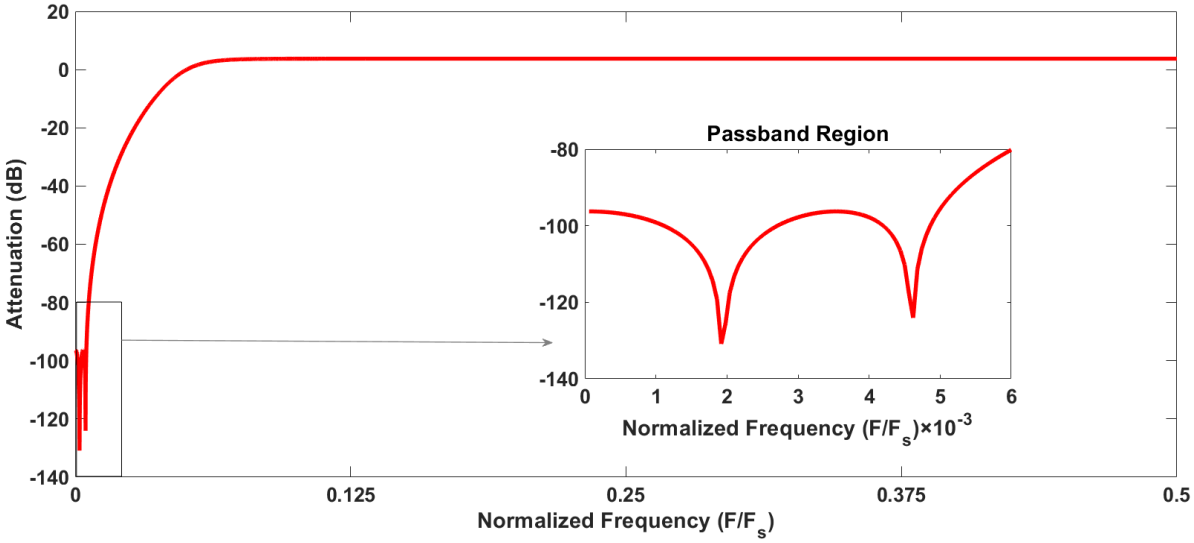

Fig.4.8. The Noise Transfer Function of the Coarse Segment of the 3-segment example

which results in a loop filter described as

$$H_2(z) = \frac{0.8570z^1 - 0.5872}{1z^2 - 2z^1 + 1}. \tag{4.34}$$

The RMS noise level created by this modulator is equal to $3.0497*10^{-5}V_{RMS}$. The response of this modulator can be see in Fig.4.9.

This leaves only the third and finest modulator. Here, it must produce no more than $4.033*10^{-10}V_{RMS}$ of noise. Given the quite substantial scaling differences between the segments, the choice of filter here is hardly consequantial. As such, a 1st order Butterworth noise transfer function was chosen, characterized by

$$NTF_3(z) = \frac{z - 1}{z - 0.5095}, \tag{4.35}$$

which is produced by the loop filter

$$H_3(z) = \frac{0.4905}{z - 1}. \tag{4.36}$$

This modulator produces a noise output of $7.1446*10^{-4}V_{RMS}$. Its response is visualized in Fig.4.10

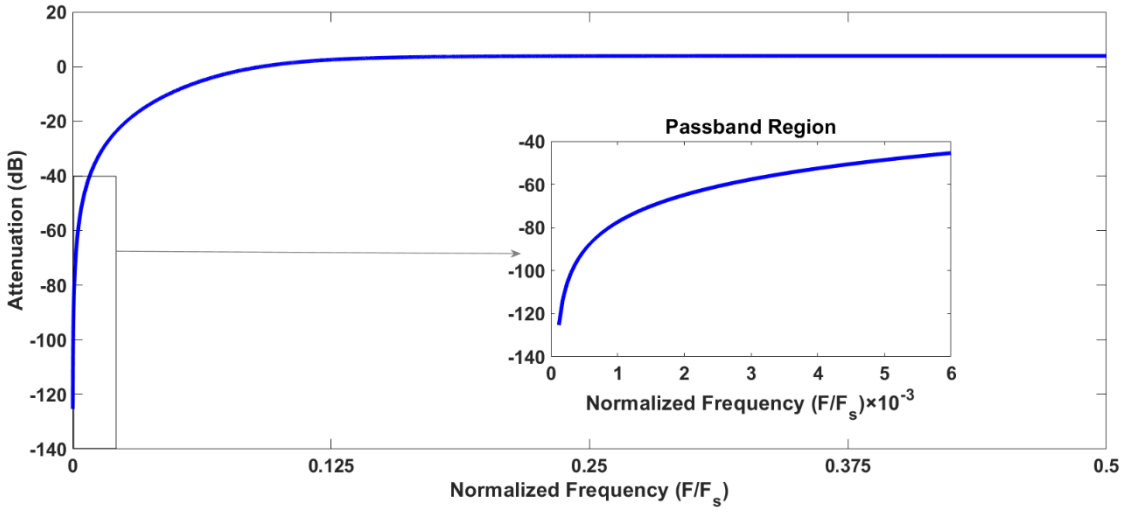

Fig.4.9. The Noise Transfer Function of the Middle Segment of the 3-segment example

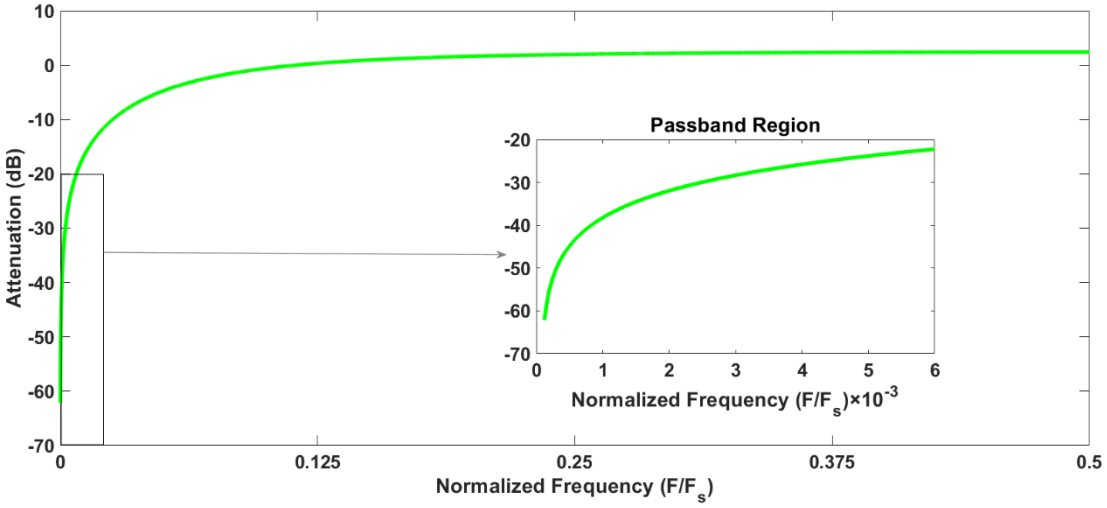

Fig.4.10. The Noise Transfer Function of the Middle Segment of the 3-segment example

The segmented DAC has a noise total of $2.1345*10^{-7}$ V_{RMS}. Its SNR is found to be 124.38dB, satisfying the design requirements and providing 20.73 as its ENOB. The DAC is summarized in Table 4.2. and Fig.4. 11.

Table 4.2.
A Summary of the Three-Segment ΔΣ DAC

Modulator	Data Allocated	Scale Factor	SNR found	ENOB found
Coarse	12 Bits	2^8	124.399	20.733
Middle	4 Bits	2^4	81.284	13.547
Fine	4 Bits	1	53.889	8.982
Overall	20 Bits	N/A	124.383	20.731

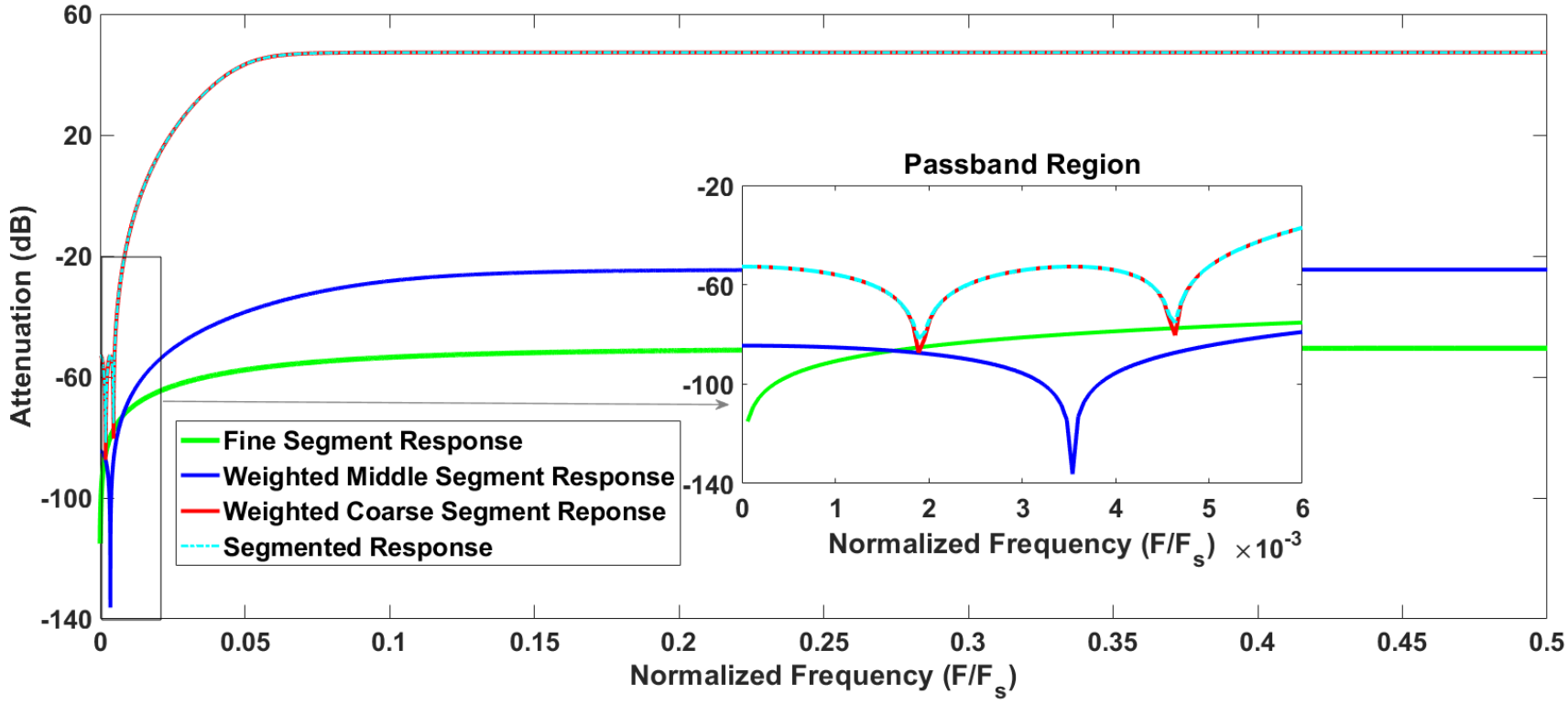

Fig.4. 11 The overall input-output magnitude response of the 3-segment example with a view of the passband region together with the coarse, weighted-coarse, weighted middle, and fine segment behaviors for comparison purposes.

4.4. Summary

In this chapter, a general design method was demonstrated for both a two-segment ΔΣ DAC and a generalized K-segment ΔΣ DAC. This method made use of the properties of segmented ΔΣ DACs found in the previous chapter, which dictate that the coarse segment's response typically dominates the response overall.

It has been shown that there is a systematic and automatable way of producing segmented designs in the last subsection. Two designs were created all will be used again in the following chapters. This is to add on more non-idealities and prove the effectiveness and accuracy of the design method.

Chapter 5: Mapping Segmented ΔΣ Modulators to Digital Realization

5.0. Introduction

The two previous chapters have shown how a segmented ΔΣ modulator can have its noise shaping response changed with different weight being applied to each constituent modulator's output. This is done by partitioning the digital input data among the ΔΣ modulators which make up the overall DAC. But this presents new design challenges, as it presents an extra degree of freedom, and further possibilities for the design to become unstable, or for the design lack critical characteristics in its noise-shaping.

In this chapter, we will establish sound design principles for segmented ΔΣ modulators, regarding both the issue of partitioning and modulator topologies.

5.1. Partitioning Data

For any ΔΣ modulator, there is a limit on the input range in which it can be considered stable. Partitioning presents new constraints on a DAC design, which will be covered in this section.

5.1.1. Considerations on Data Partitioning

Consider a 1.5 Hz, 6554-amplitude sine wave centered around a DC level of 2^{15} is applied to a 2-segmented ΔΣ DAC, with N = 16, and $N_C = N_F = 8$, as shown in Fig.5.1. This can be thought of as a sine wave centered at the 50% of the maximum input range, and an amplitude of 20% of that same range (for a 40% swing).

The signals appearing at the input before segmentation, input to the coarse modulator, and the input to the fine segment modulator are displayed in Fig.5.2. The input signal to the coarse section (middle curve) swings by about 40% of its maximum full-scale range of 2^8.

Further increasing the input amplitude to the overall DAC will result in a larger portion of the coarse segment input range being used. This contrasts with that which occurs with the input to the fine segment modulator as shown in the bottom graph. Here the signal occupies nearly 100% of its full-scale input range. This condition will always occur unless the signal amplitude is low enough to not make use of the coarse modulator.

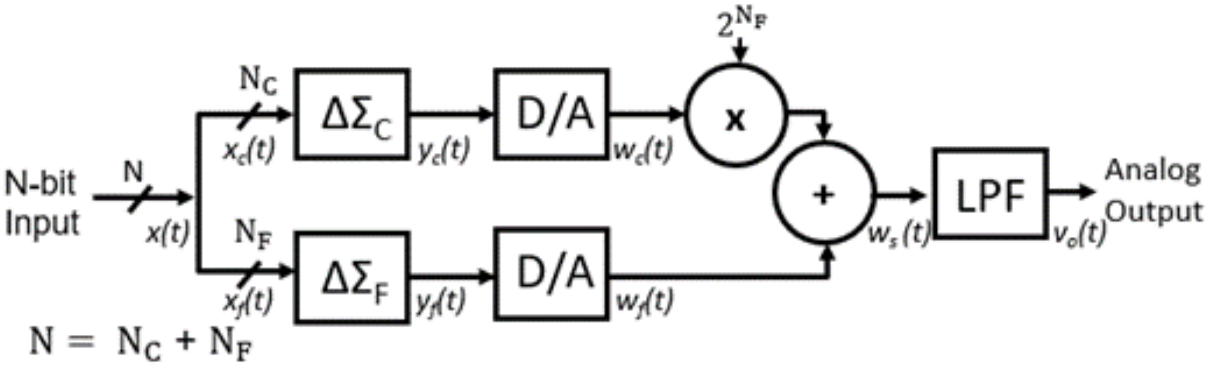

Fig.5.1. Structure of the segmented ΔΣ DAC.

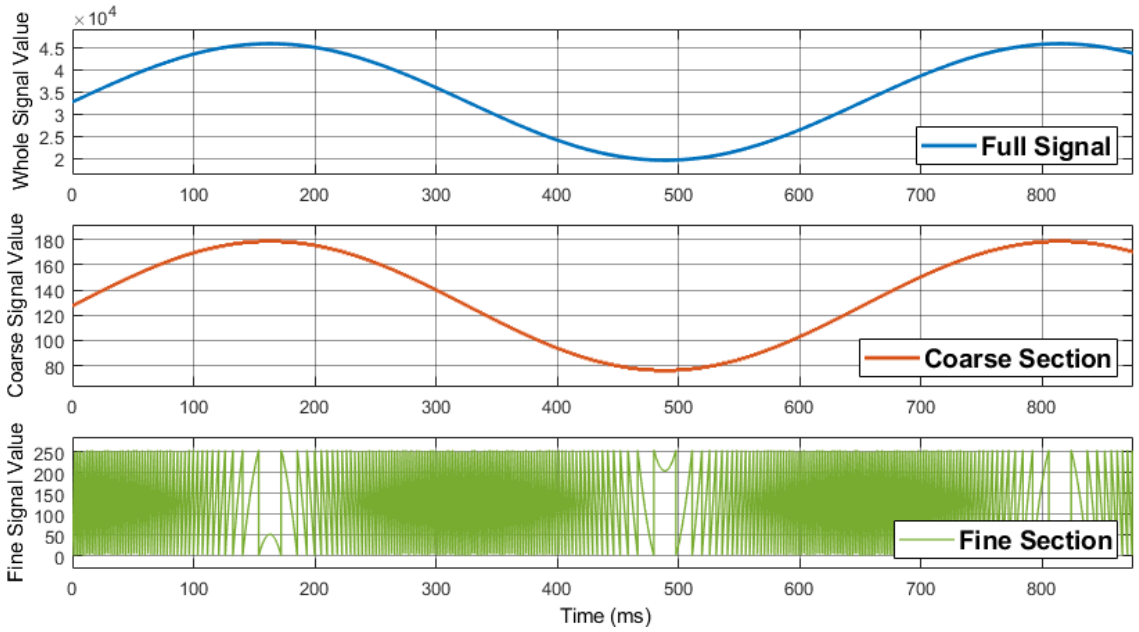

Fig.5.2. An example case of the input range problem for segmented designs, where N_C=N_F=8

Thus, the fine modulator must be designed for stable operation over a wide input range [19]. This generally implies the need for a low-order modulator (less than 3rd).

Because of segmentation, all non-coarse segments will end up with a near full input range should the input signal ever affect the coarse segment, which it will in nearly all practical uses. Due to excess noise from the output feeding back into the input, this may cause instability in some designs [19].

5.1.2. Limits on Data Partitioning

This subsection is dedicated to the examination of the practical limits of partitioning data in a segmented ΔΣ DAC, so as to be better informed when making a design based on the equations and principles established in the previous chapters.

It would be a mistake to assume that a segmented design derived from theory is fully realizable in a practical sense (such as on an FPGA). To demonstrate, a practical example will be used. Recall section 4.3.1. In it, an example segmented DAC consisting of two ΔΣ modulators was created: the coarse

segment, using a 3rd order inverse-Chebyshev response, and the fine modulator using a 1st order Butterworth response.

This design and several variants were replicated using MATLAB's Simulink environment, and compared the SNR found in the simulations against the expected SNR, as derived from the equations in previous chapters. The variants were created by changing the partitioning of the input data between the two segments, while keeping the number of input bits N equal to 16.

We can see on Fig.5.3. how the SNR of the system changes as we partition the data path differently ($N_C + N_F = N = 16$). The green line shows the expected results, the blue line shows the results found in the simulation, and the magenta line shows the SNR of a single ΔΣ DAC, effectively showing how the coarse DAC alone performs, with no data partitioning (i.e., N_C=N).

There is a clear range over which the expected results found from theory match up closely with the simulations, but performance falls drastically at the points where the coarse segment receives 4 bits or fewer. The performance degradation for low N_C is self-explanatory; for a data path as narrow as this, DACs will generally experience stability issues. As such, even the noise-shaping provided by the fine DAC will be of no use, as the most significant bits effectively produce noise.

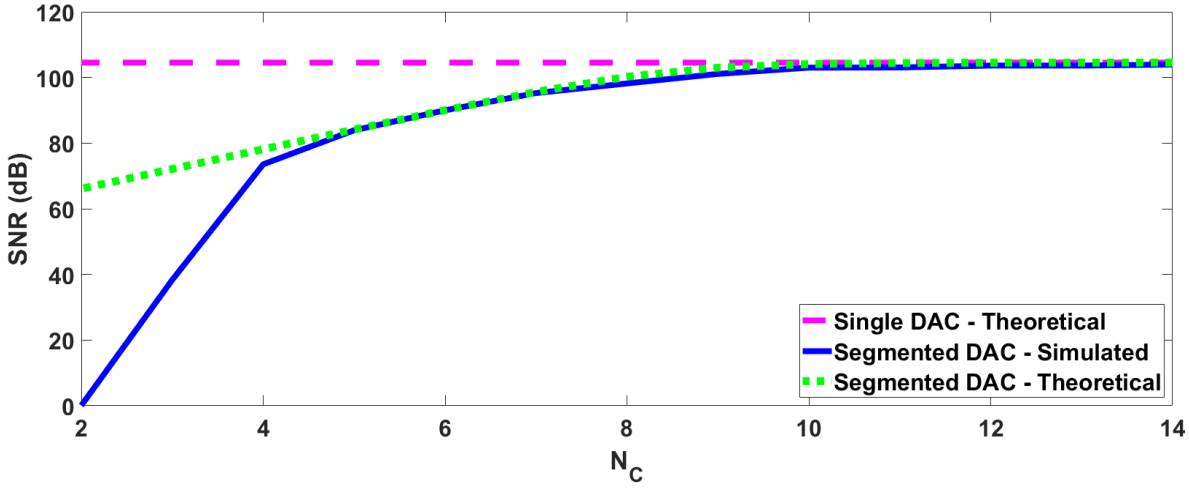

Fig.5.3. A comparison of various partitioning of data for a segmented ΔΣ DAC using 3rd order coarse 1st order fine modulators N_C+N_F =16 (the 16-Bits Partitioning Example)

To cement this further, a second example is presented. The segmented ΔΣ DAC sports a data path consists of 20 bits, with a 4th order coarse modulator with Inverse-Chebyshev noise shaping, and a 2nd order fine modulator with a Butterworth shaping response.

Just as before, the partition of the input data bits is varied, and comparing this to a single DAC using the same noise response as the coarse DAC, with no partitioning of the input data (i.e., $N_C = N = 20$). The results are displayed in Fig.5.4.

As can be seen, the data found here matches up quite well with what was found in the first example. The biggest difference is that the performance drop for high N_C is slightly lower (around 1dB, rather than 0.5dB). It was also found that it is generally unsound to leave the coarse DAC with a minority of the input data, with 5 bits being a critical number.

More information can be extracted from these simulations, seeing as they have effectively simulated four different orders of modulators in the two examples.

Firstly, it is generally unwise to use a higher order modulator for the fine segment, as performance issues have already been observed when placing them as coarse DACs with a narrow data path. Adding on further constraints would only

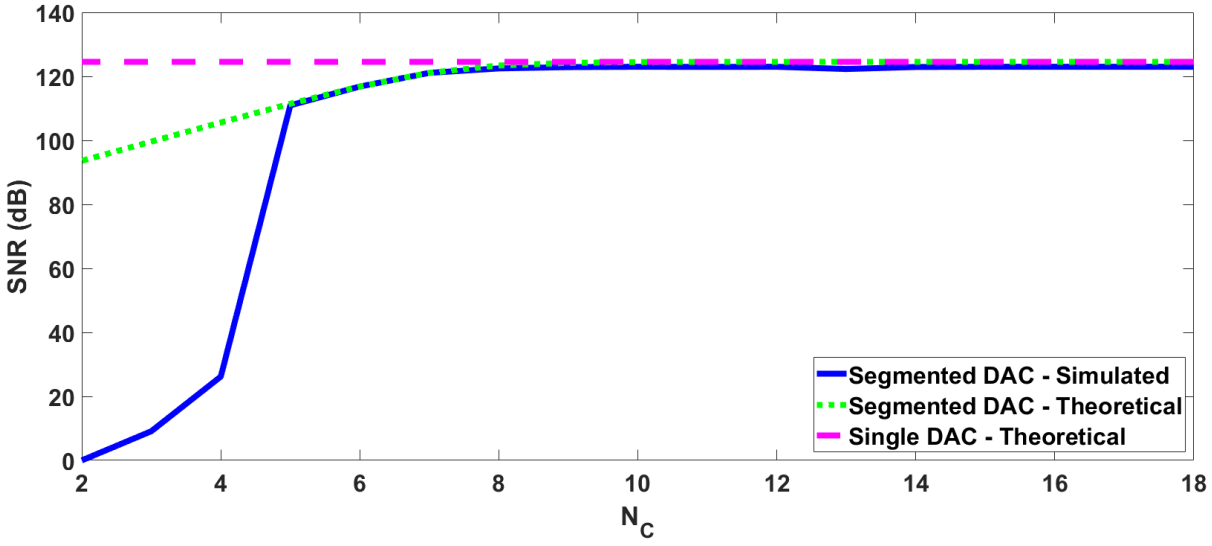

Fig.5.4. A comparison of various partitioning of data for a segmented ΔΣ DAC using 4th order coarse 2nd order fine modulators (N_C+N_F =20) (the 20-bits Partitioning Example)

worsen this. Second order modulators work as intended, third order is passable, albeit risky.

Secondly, it is better to allow the coarse modulator to maintain a sizeable portion of the input data, as its performance dictates the bulk of the segmented DAC's performance. At a bare minimum, 4 bits are necessary for proper operation, but most of the performance can be retained if the coarse modulator maintains at least half of the input data bits.

5.1.3. Remedies to Partitioning Problems

As mentioned in the previous sub-section, some issues may arise when mapping a segmented ΔΣ DAC to a digital realization. However, some of these issues can be mitigated with the methods outlined here.

Firstly, using low-order modulators can mitigate the instability issues. For many designs which would only use the fine segment to "relax" the hardware requirements, this may be sufficient.

Secondly, if a low-order modulator cannot satisfy the design specifications, a designer can always use a pre-amplifier or bit-shifting to effectively reduce the range of the modulator's input signal as seen in Fig.5.5The pre-amplifier, in practice, would simply be a summation of several bit-shift operations. Often, this would mean scaling the difference between the incoming signal and the intended "center" value of the signal (mid-range in most cases), as shown inFig.5.6. To correct for the effectively lowered input, the coefficient used when adding the

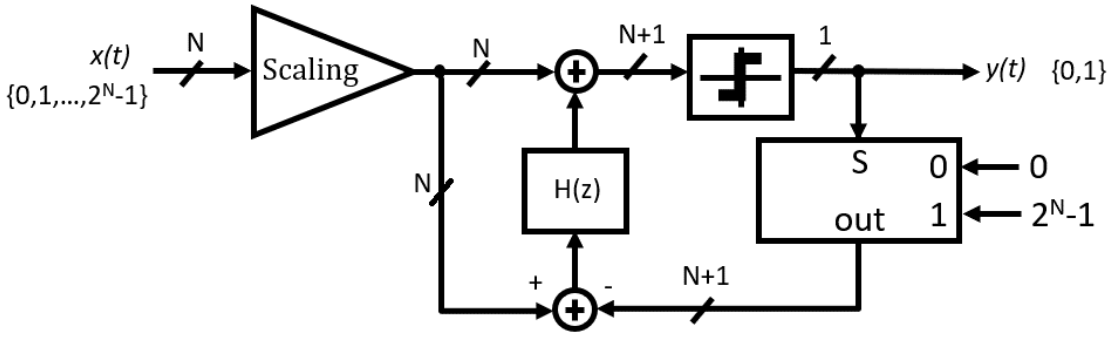

Fig.5.5. An example of a pre-amplifier used to scale down an input

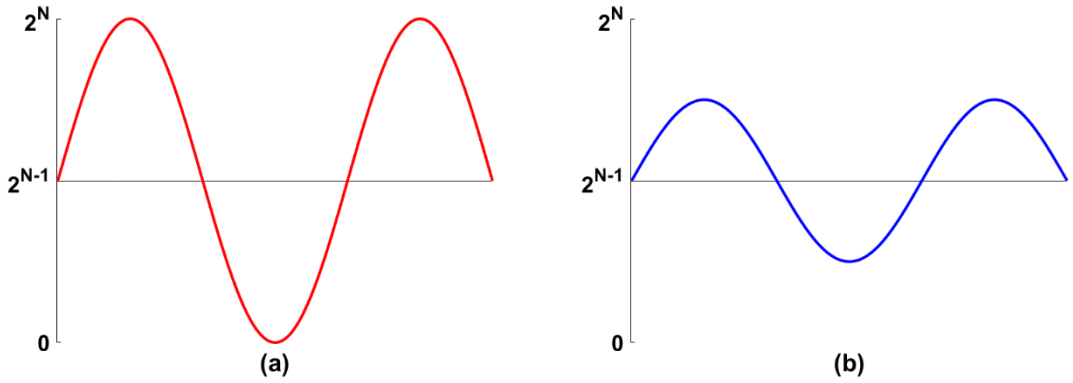

Fig.5.6. A Signal before scaling (a) and after scaling (b). Note that while swing has changed, the range has not.

individual segment outputs can be increased to compensate. As an example, one could imagine a fine segment that would normally be weighted such that its 1-bit output would only produce 1/16th of the voltage produced by coarse output signal. But in the case where the input has already been halved due to scaling. This could be resolved so that the 1-bit output would now have a weight of 1/8th of the coarse output signal.

Thirdly, a designer could pad the signal paths of the feedback loop to effectively have the system treat the input as if it were reduced, as seen in Fig.5.7. The advantage of this method over the previous one is that no information is lost, and thus no quantization noise is added. Fig.5.8 shows an example of a signal being padded with a single bit. Note that while padding causes the signal to cover only 50% of the full input range, the values from before and after padding are

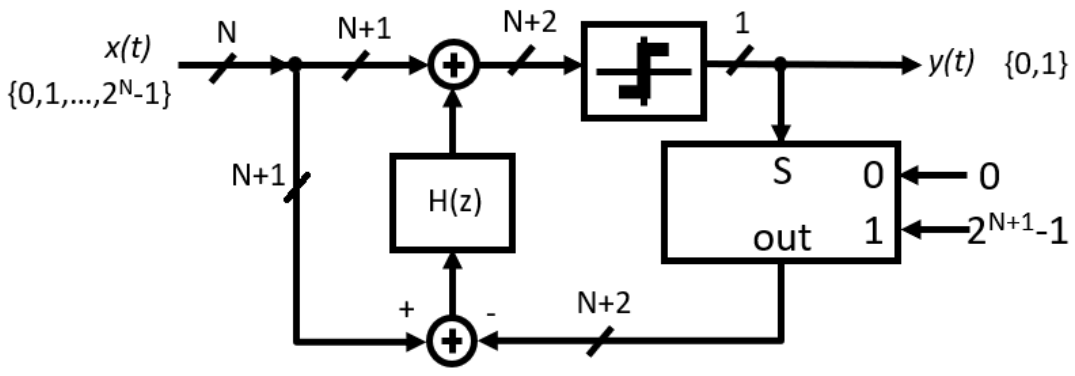

Fig.5.7. An example of a modulator using padding in its operation.

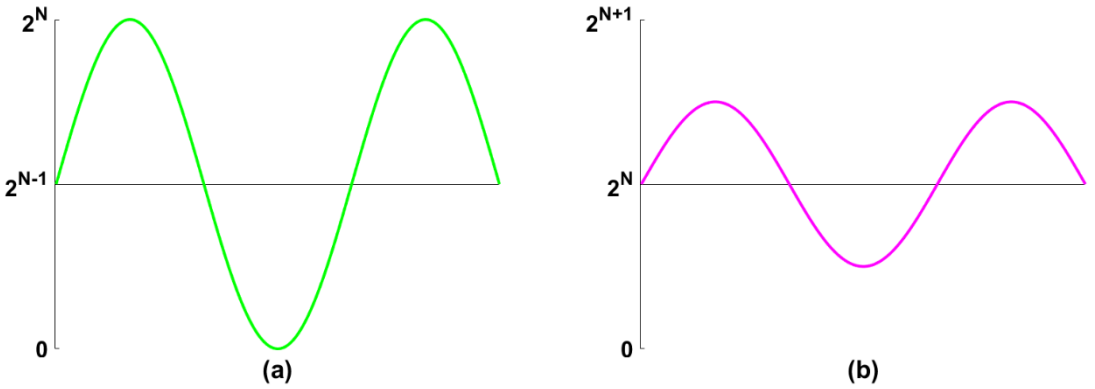

Fig.5.8. A signal before padding (a) and after padding (b). Note that the signal itself is unchanged, only the width of the data path.

identical. The downside is that this would increase the hardware required to realize the design, but as will be seen in this chapter and the next this may still outperform singular designs.

As with the pre-amplifier/shift method, we would need to adjust the reconstruction coefficient to reflect the changes made. Here too we would need to increase the factor, multiplying by two if one bit is added, by four if two are added, and so on. Note that since adding a single padding bit effectively doubles the input range, one bit is typically all that is required to solve instability issues. Rarely would a designer need two or more bits, at which point the design itself likely sports other, more pressing problems.

5.2. Transfer Function Mapping

In the previous section, several segmented ΔΣ designs were simulated, and the data from those simulations was compiled to get a better understanding of where the limits of segmentation lie.

There is, however, another layer of complexity to the designs that the previous performance equations do not account for. For most simulation tools, the data types used for the coefficients, accumulators and outputs of the loop filter will use a floating-point format. However, when mapped to an FPGA, fixed-point representation is used. Thus, mapping the realization from simulation to hardware

leads to quantization issues. In particular, the loss in precision for the coefficients of the loop filter can cause results to stray from the desired operation.

5.2.1. Direct-Form Realization Topology

There exist several topologies to map modulators onto fixed-point hardware. The most straight-forward of these is the Direct Realization, which simply uses a series of delays and scaling units. This is shown in Fig.5.9, as part of a modulator, using the same unity STF configuration as the rest of the thesis.

This topology, however, has its limitations. The rather “naive” approach used to create the modulator leaves many of its qualities lacking, as will be shown in the next sub-section.

5.2.2. Limitations of direct mapping

For most simulation tools, the data types used for the coefficients, accumulators, and outputs of both numerator and denominator will use a floating-point format, unless the designer explicitly asks otherwise.

The biggest problem out of the previously mentioned factors is that of the data type used for filter block coefficients, as the data types of the accumulators and output are trivial to derive once the input format and transfer function coefficients have their data types identified.

This can be a problem for a designer, especially if they are trying to establish a workflow that can create a design that can be described in a hardware description language (HDL), as floating point cannot be used on FPGA, or be readily mapped on to a digital circuit without using more expensive general-purpose computers.

To demonstrate, a single 16-bit ΔΣ DAC will be used in a simulation, using the direct-form realization. The ΔΣ modulator structure used is identical to that shown in Fig.5.9. The OSR was set to 100 and the sampling frequency was set to 1 Hz. The loop filter was designed using the method of optimized pole and zero placement for a Butterworth filter as described by Schreier [16]. The loop filter used can be characterized by

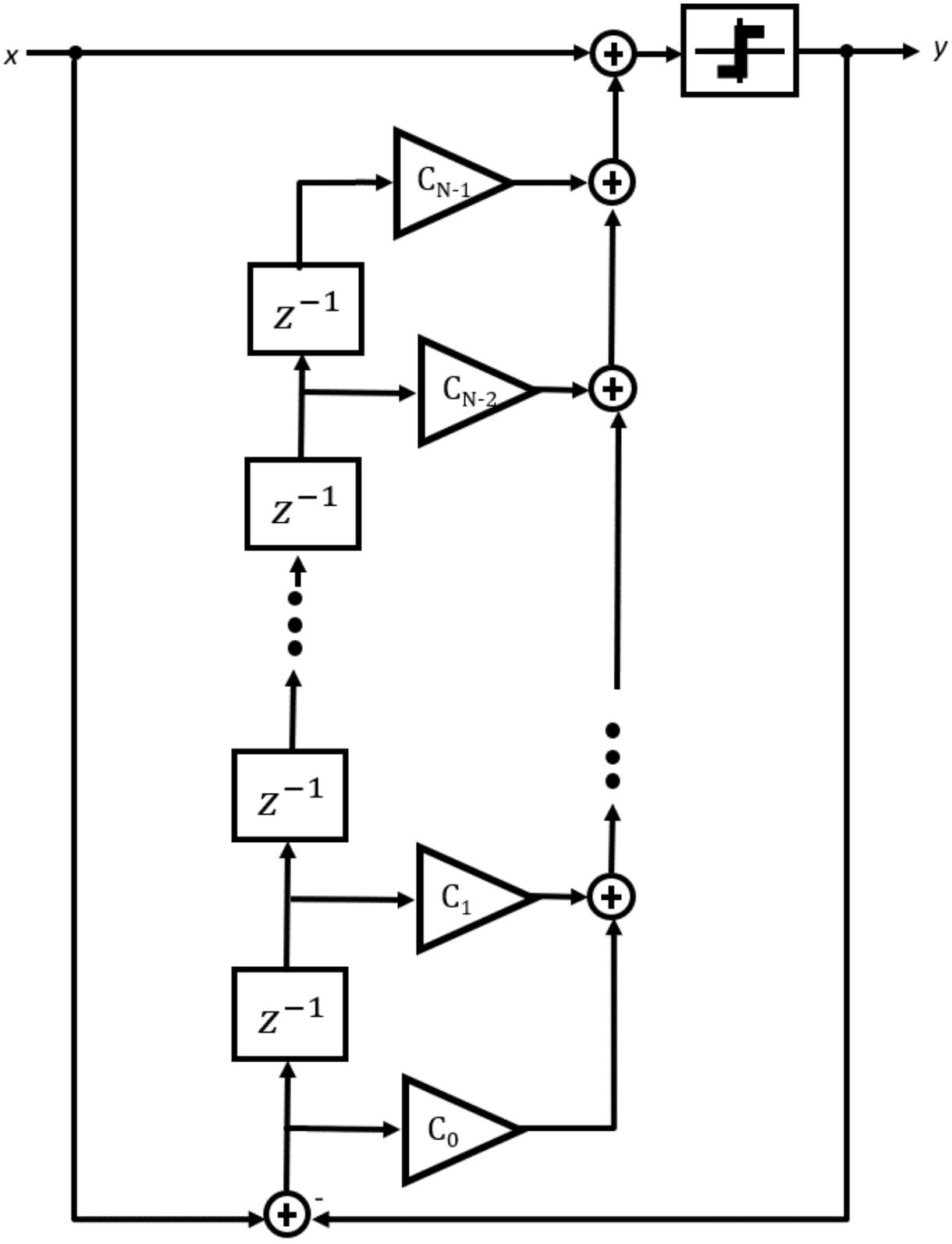

Fig.5.9. A ΔΣ Modulator with unity STF, using direct form realization.

$$H(z) = \frac{0.9164z^3 - 2.3523z^2 + 2.0495z - 0.60350}{z^4 - 3.9992z^3 + 5.9983z^2 - 3.9992z + 1}. \quad (5.1)$$

A 3 mHz, 2^{14} amplitude digital sine wave centered around 2^{15} was applied to the ΔΣ modulator input and the output was observed for three separate conditions involving the numerical precision of the coefficients: double-precision (64 bits) values, 3 bits for the integer value and 4 bits for the fractional part, and 3 bits for the integer value and 3 bits for the fractional part. The periodogram of the output power spectrum densities (PSDs) are displayed in Fig.5.10.

It is apparent that the 4-bit fractional-coefficient implementation follows the behavior predicted by theory. However, the 3-bit fractional-coefficient realization sufferred serious error, rendering it useless.

Complicating this issue further is the fact that there is no sure way for a designer to tell if the quantization effects will render the design useless, as in some cases a loss in coefficient precision doesn't always change the overall performance. This can be a problem for a designer, especially if they are trying to establish a workflow in a hardware description language (HDL), as floating point number representation cannot be used on an FPGA or be readily mapped onto a digital circuit without using expensive general-purpose microprocessors or DSPs.

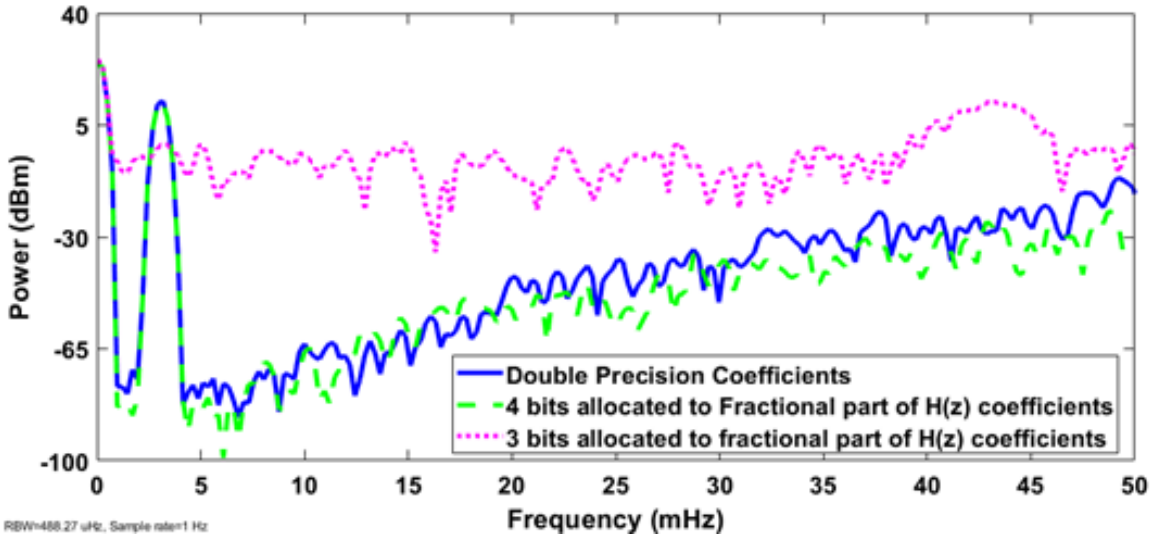

Fig.5.10. The PSD for a 2-segment ΔΣ modulator subject to different numerical precision in the loop filter coefficients.

5.2.3. Resonator-Cascade Topology

From the previous section, it has been made apparent that using the direct form of the NTF introduces problems that would make the design process of a segmented ΔΣ DAC quite tedious, and likely un-automatable with software.

Thankfully, this is a problem that has been solved multiple times over with various topologies that accomplish the same task as the direct form, while also being far more useful on hardware that will use fixed-point data types. For this thesis, the resonator-cascade will be the choice.

Modulator realization, as a design challenge, can be reduced to the task of finding a structure that treats its input data such that it produces the required NTF. While it is possible multipliers to accomplish the same effect with the direct method, this is only the readily apparent solution. The linear filter H(z) can have each term in its numerator and denominator as a function made up of structure coefficients. In making a structure like this, an engineer minimizes the number of multipliers used. Fig.5.12. demonstrates the general structure of a resonator-cascade N^{th} order modulator, as seen in previous work [18]. The structure is made up of repeating cells that cascade into the next one. Each cell is made up of a pair of discrete-time integrators, one forward integrator characterized as $\frac{1}{z-1}$, leading into a backward integrator characterized as $\frac{z}{z-1}$. Note here that only the latter type has direct feedthrough. Within these cells, one can note a feedback gain block labelled as A_i, and two B_i gain blocks which feed into the output path. What makes these gain blocks more advantageous to use that the typical multipliers are the fact that, in most cases, the gain be approximated to a single bit value, or effectively a shift register. Note that here the STF remains at unity, which was already stated as a desirable property when designing a ΔΣ DAC.

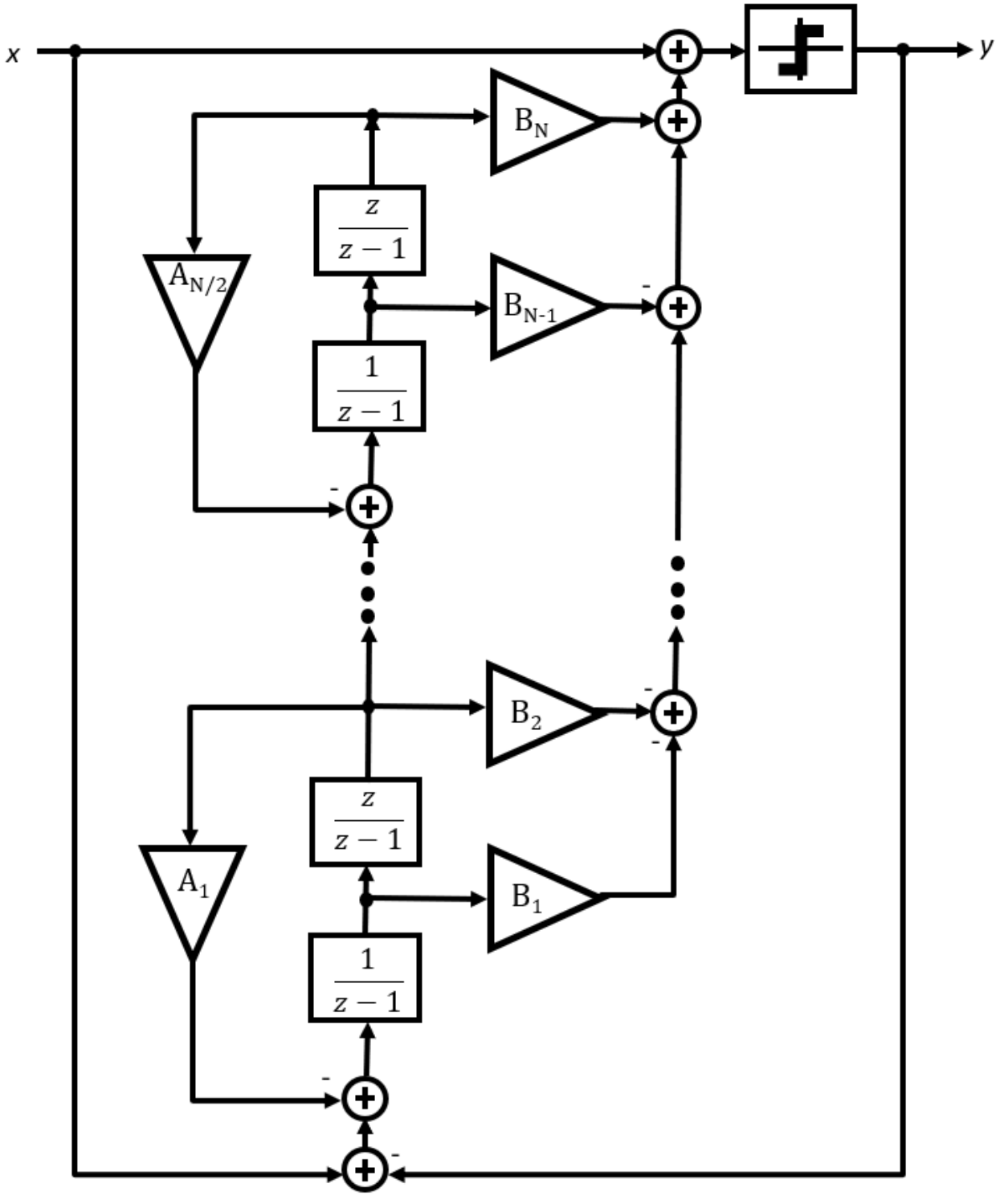

Fig.5.11. A ΔΣ Modulator with unity STF, using resonator-cascade realization.

5.2.4. A comparison of Direct Mapping and Resonator-Cascade Topology

The core difference between a direct-form and a resonator-cascade structure is the means to represent a multiplication operation. In the direct-form topology, a multiplication is implemented using multipliers, whereas in resonator-cascade these would be implemented using arithmetic shifts (which are less expensive to implement). For comparison, the two realizations were simulated with several full-scale input tones centered around $3*10^{-3}$ F_S as described previously. The tones are not set to be equal to enable them to be seen on the same frequency plot. The periodogram of the output PSDs over the passband region of the two ΔΣ modulators are shown in Fig.5.12.

The blue dotted curve represents the resonator-cascade topology and the orange-dotted curve represents the direct form one. In addition, the expected passband magnitude behavior is illustrated by the solid red line. As can be seen, the ideal passband behavior demonstrates two attenuation zeros at approximately 1.8 mHz and 4.3 mHz.

One can also note that the PSDs of the two curves agree very closely in the low frequency region of the passband region but differ significantly in the upper

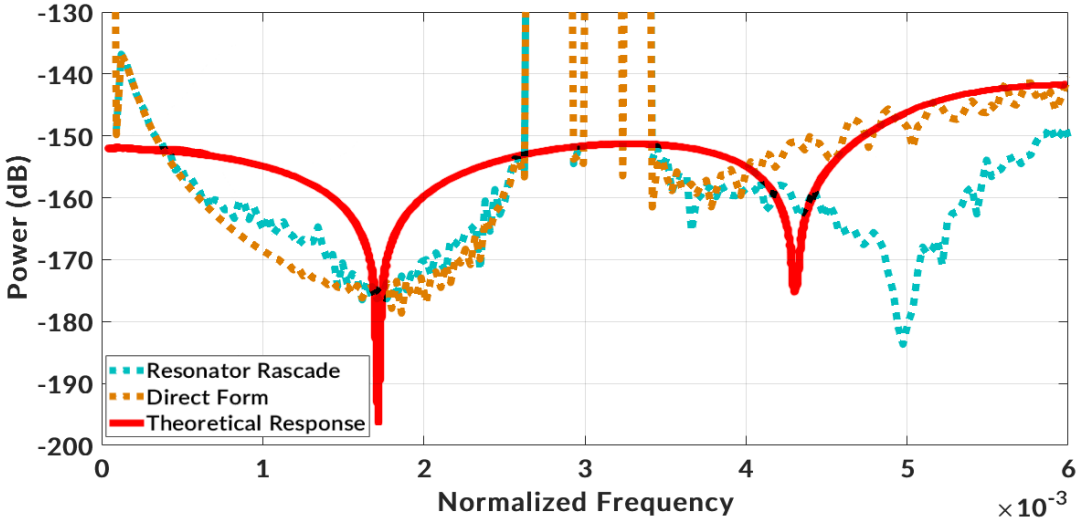

Fig.5.12. Passband response comparison found through simulation and theory.

frequency region. This can be attributed to the fact that the direct form realization seems to have been lost the one of its attenuation zeros. This is not the case with the resonantor-cascade realization, although its attenuation zeros are shifted with respect to the ideal behavior.

5.3. General method/Flowchart for mapping on to FPGA.

Over the course of the previous chapters, the understanding of segmented ΔΣ modulators has been expanded. From the theory, to design, to practical considerations when mapping our design to an FPGA. The process can be summarized as shown in Fig.5.13.

5.4. Summary

In this chapter, the theory and methods developed in the two previous chapters were refined into a set of design principles. With these principles, it was shown how a segmented modulator can be created using data partitions, and how they may result in significant departure from theory. In addition, the use of different filter topologies was explored, and the results showed in radically different frequency behavior.

In the next chapter, the principles and simulation results found up until this point will be validated in various experiments.

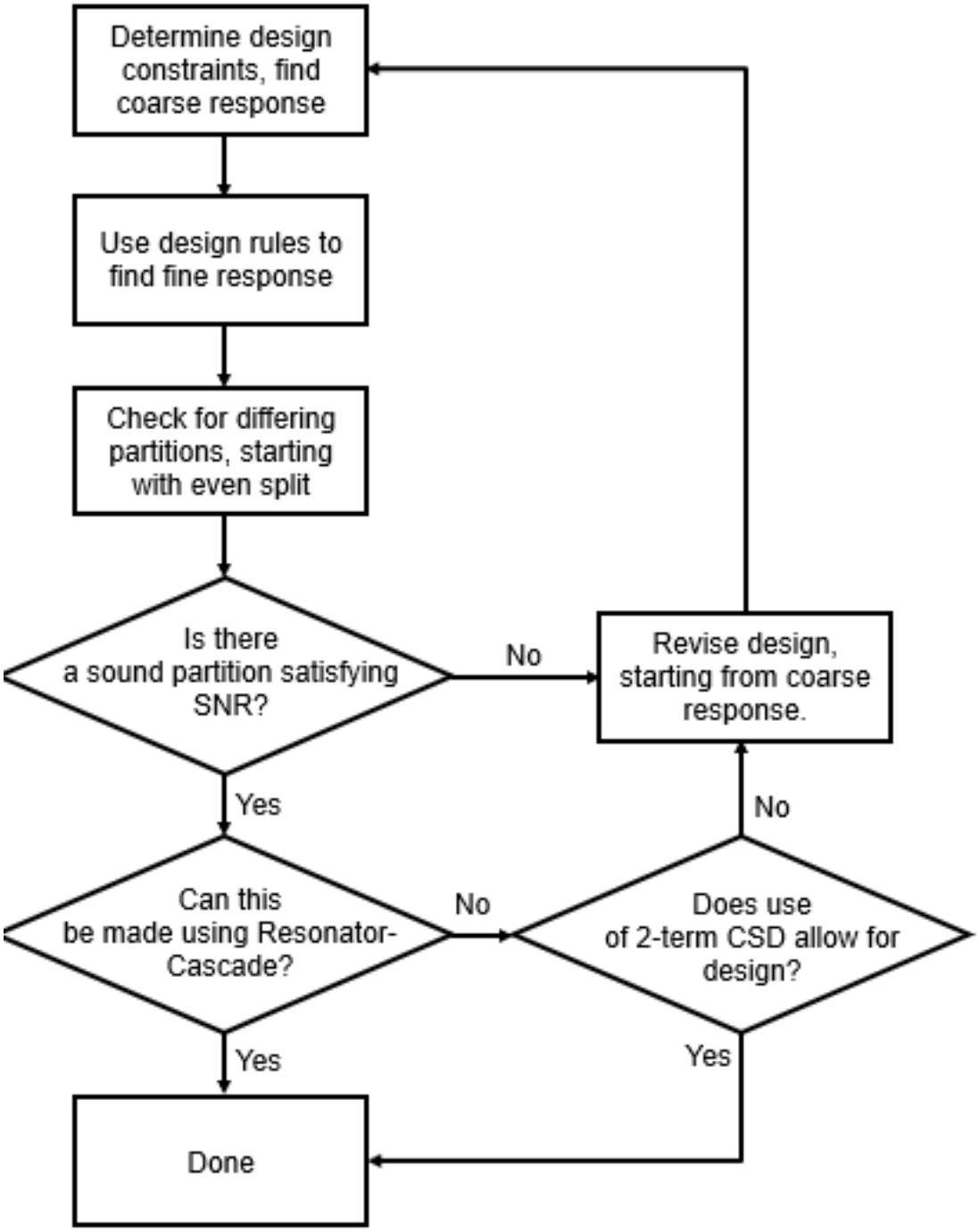

Fig.5.13. Flowchart for the general process for designing a segmented ΔΣ DAC.

Chapter 6: Experimental Validation

6.0. Introduction

The previous chapter served to build the theory behind segmented ΔΣ DACs, as well as the design principles needed to create them.

In this chapter, several experiments will be performed to validate those previous findings, and further refine the principles of segmented ΔΣ modulator design.

To be validated in this chapter are the running examples introduced in chapter 4, the findings on data partitioning as shown in chapter 5, the simulations concerning topology comparison, and finally the hardware costs of various partitions and realizations.

In all experiments in this chapter, the sampling frequency will be set to 3.125 MHz

6.1. Validating Segmented ΔΣ DAC Noise Responses

In chapter 4, we presented two examples of segmented ΔΣ modulators, but only in theory. For those two cases, however the specifics of the design were not considered. In this section, these two cases will be mapped on to an FPGA to gauge how well the theory and simulations predict on-board behavior.

6.1.1. Hardware Set-up

To gather data, one of two methods was used. For the two-segment case, a digital sinusoidal signal is synthesized within the FPGA and applied as the input to the ΔΣ modulator in question. The segmented outputs of the ΔΣ modulator are weighted and added together by means of a current-steering circuit, then filtered using the discrete component reconstruction circuit shown in Figures 6.2 and 6.3 [20]. The output was then sampled using the Agilent DSA80000B Digital Signal Analyzer and processed using MATLAB. However, due to the varying topologies not matching an even split of 16 bits across two DACs, all other experiments will instead use the readings from the FPGA directly and all analog values will be extrapolated from the digital output, rather than using a reconstruction circuit.

6.1.2. 2-Segment Running Example

The first circuit to be put to the test was the 2-segment running example from section 4.3.1. Its implementation is shown in Fig.6.1. As can be seen, the

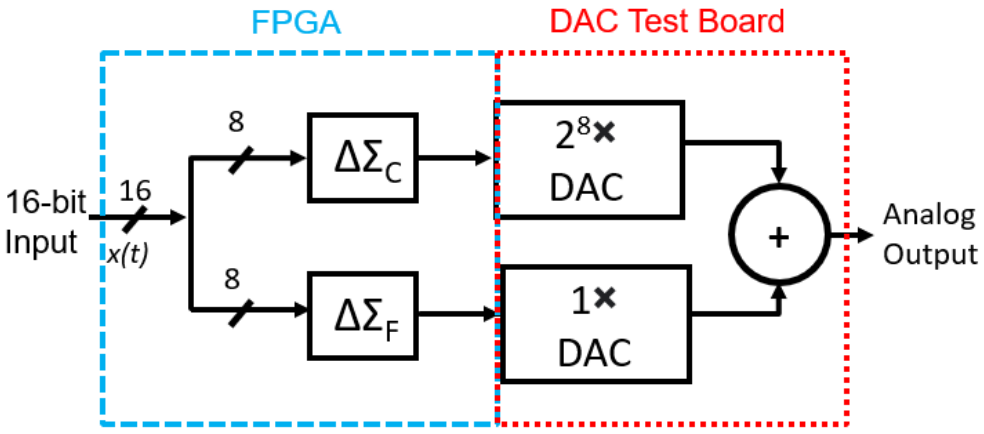

Fig.6.1. A block diagram of the test set-up.

FPGA produces the two bitstreams, while the DAC test board was used to create the analog voltage. Table 6.1. presents details on the components.

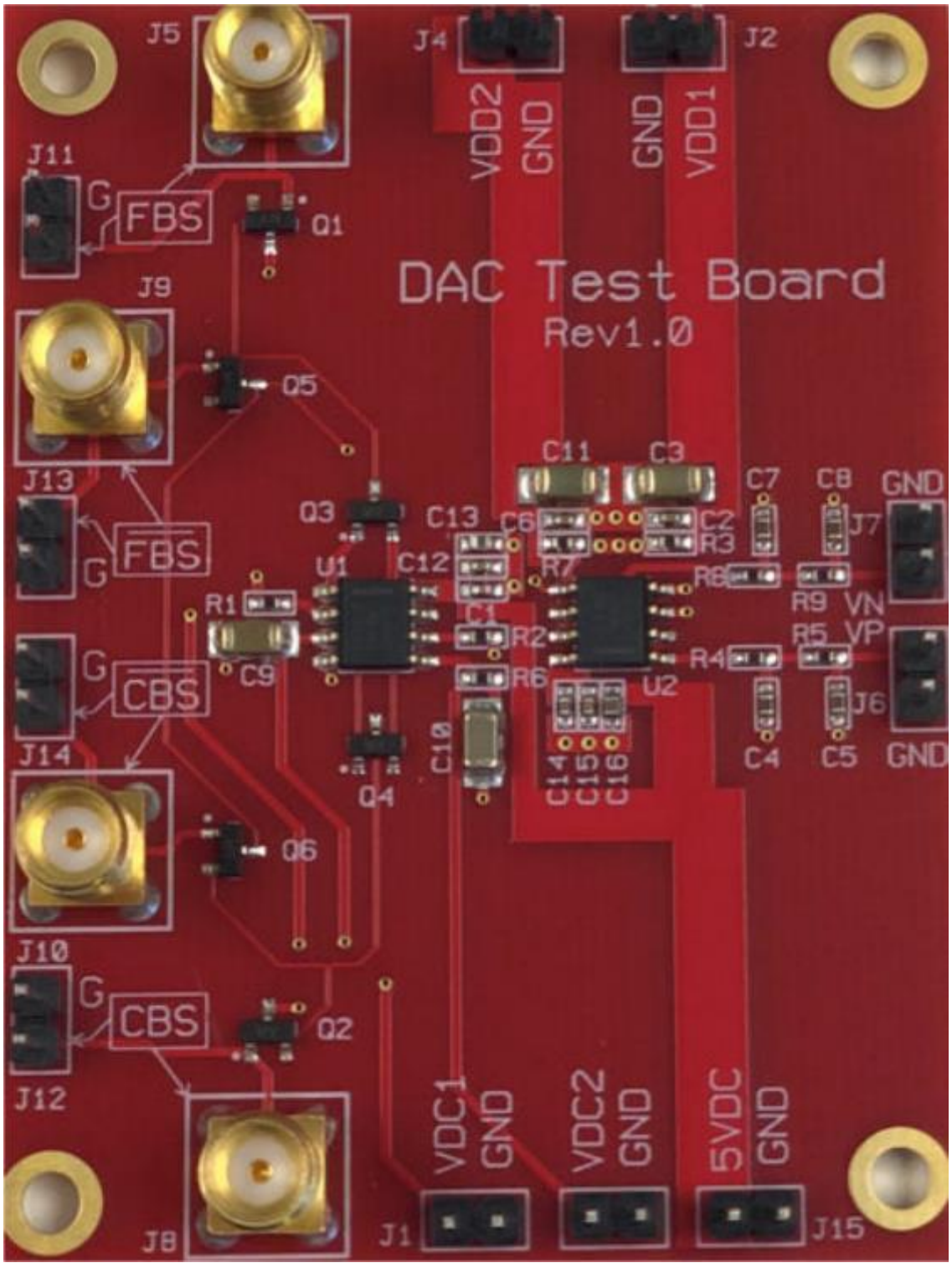

Fig.6.2. Photograph of the discrete-component reconstruction circuit.

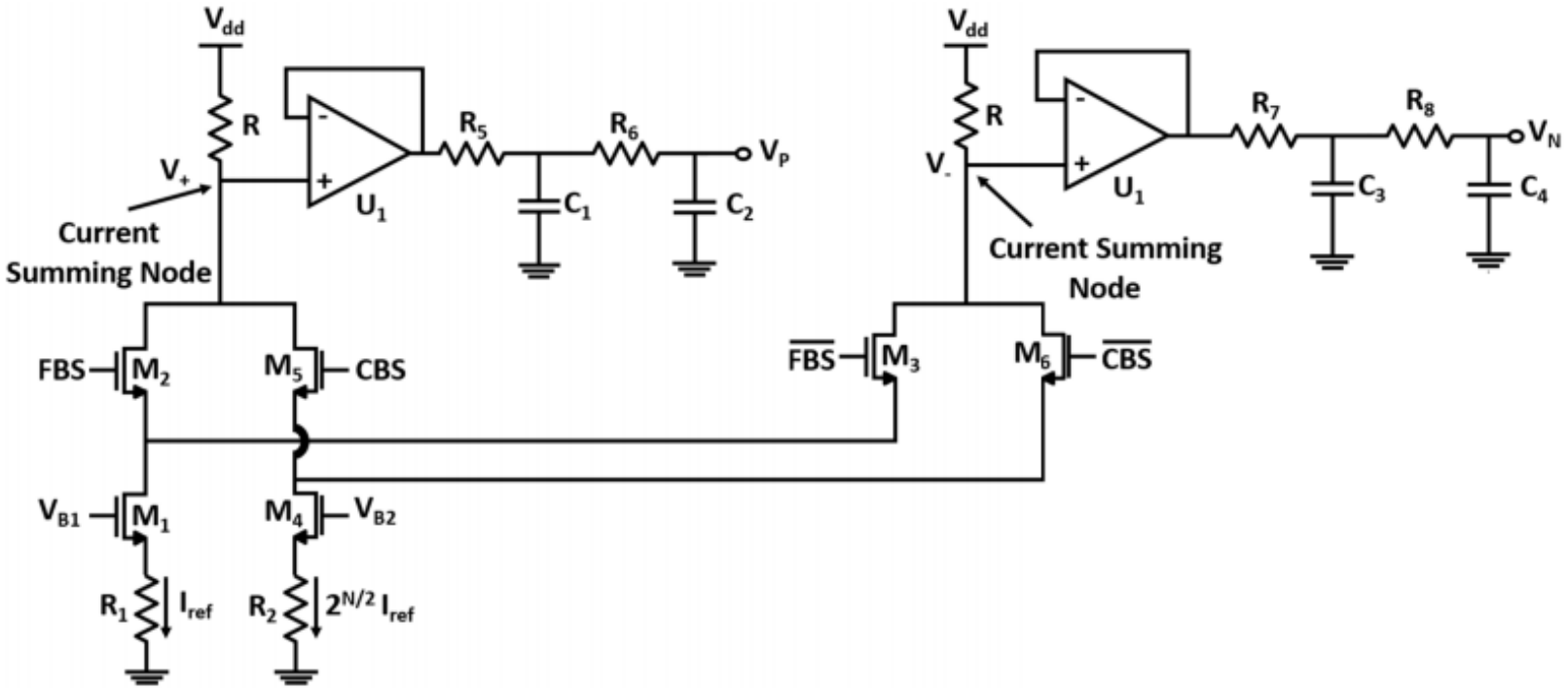

Fig.6.3. Schematic of the discrete implementation of the reconstruction circuit.

Table 6.1.

Component Values of the DAC Test Board

Component	Value/Type
R_1	100 Ω
R_2	10 Ω
R	64.9Ω
R_5, R_6, R_7, R_8	3.3 kΩ
C_1, C_2, C_3, C_4	2.2 nF
Op-Amps	TLV9052IDR
Transistors	BS170

It is also important to note from Fig. 6.3. that a complement to the coarse bitstream (CBS) and fine bitstream (FBS) were produced, with the outputs read differentially. The direct form ΔΣ modulator design was tested by way of several input sinusoidal tones and the results captured and analyzed by digital sampling techniques using periodogram of the output power spectral densities (PSDs). Fig.6.4. compares the passband response behavior between theory and the experimental results.

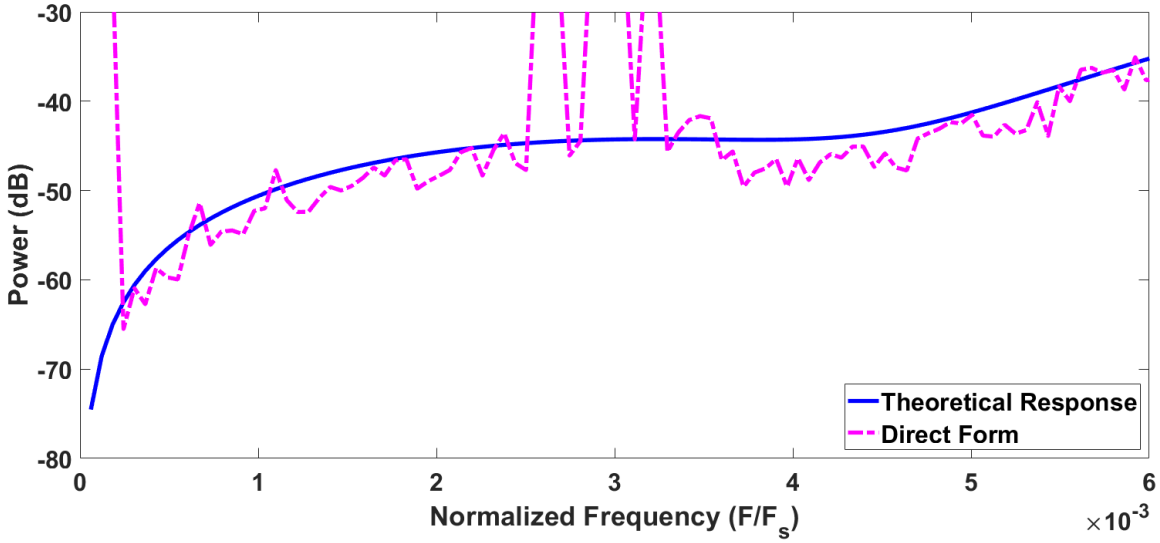

Fig.6.4. PSD comparison of theoretical and experimental passband responses of 2-segment ΔΣ modulator implemented using a direct form topology

Aside from the DC power and input tones centered at bin $3*10^{-3}F_S$ (here using 3.125MHz), the 2-segment ΔΣ modulator behavior in practice lines up very well with theory. Thus, confirming the principles described earlier.

6.1.3. 3-Segment Running Example

The 3-segment ΔΣ modulator example previously described in 4.3.2 was synthesized in an FPGA using resonator-cascade and tested on the bench. The circuit was then stimulated with input tones centered around $3*10^{-3}F_S$ (F_S being 3.125MHz), and the output signal sampled using the Agilent DSA80000B Digital Signal Analyzer. The data set was subsequently analyzed using MATLAB. The periodogram of the output PSD is plotted in Fig.6.5.

As is evident from the plot, the actual 3-segment ΔΣ modulator generated results that are in excellent agreement as expected from the work performed in Section IV. It is interesting to note that this particular example has the attenuation zeros coinciding with those predicted by theory. This is simply a fact that the desired loop filter coefficients experienced very little quantization effects.

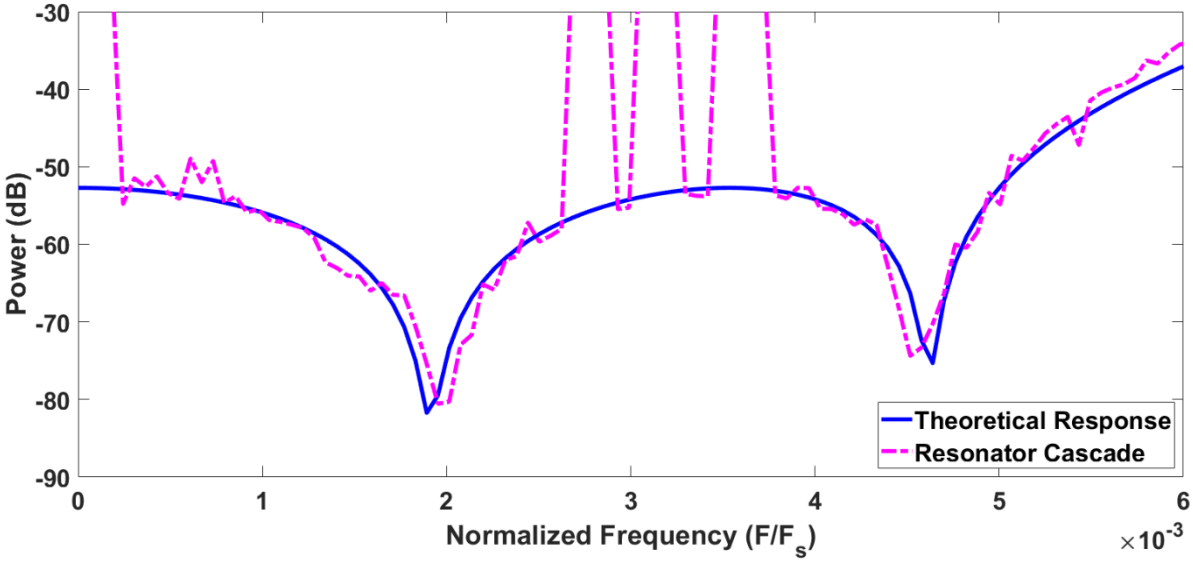

Fig.6.5. PSD comparison of theoretical and experimental passband responses of 3-segment ΔΣ modulator implemented using a resonator-cascade topology.

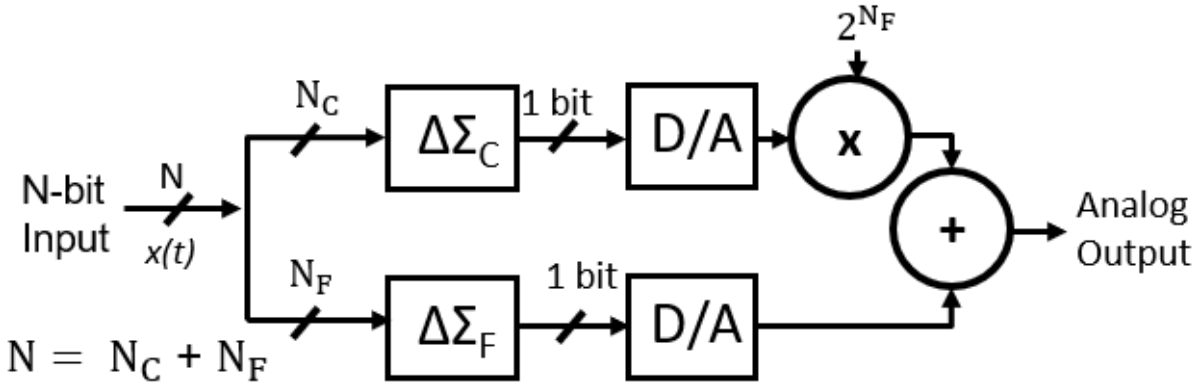

Fig.6. 6 The structure to be tested in section 6.2.

6.2. Validating Findings on Partitioning

A significant portion of chapter 5 covered the data partitioning aspect of ΔΣ modulator design. This section will serve to verify the validity of those findings. The process used here will be nigh identical to that of the previous section, but the data will be read from the FPGA directly, as almost none of the structures make use of the 8-8 bit split.

6.2.1. 16-Bits Partitioning Example

To validate the effects of bit partitioning between the coarse and fine modulator segments on the peak-SNR performance, numerous designs with the same loop filters was synthesized and placed on an FPGA. Each design was excited with a full-scale amplitude digital sinewave and the SNR behavior was then extracted. The peak-SNR versus N_C results were plotted in Fig.6.7. and compared with the simulation results of section 5.1.2.

As is evident, the results match quite well. On closer investigation, one finds that the two curves differ by no more than 0.4 dB for N_C above 4 bits.

The findings bode well for the theory and design principles developed so far and reinforce the idea that having the coarse DAC have a minority of the data allocated to it is generally unsound.

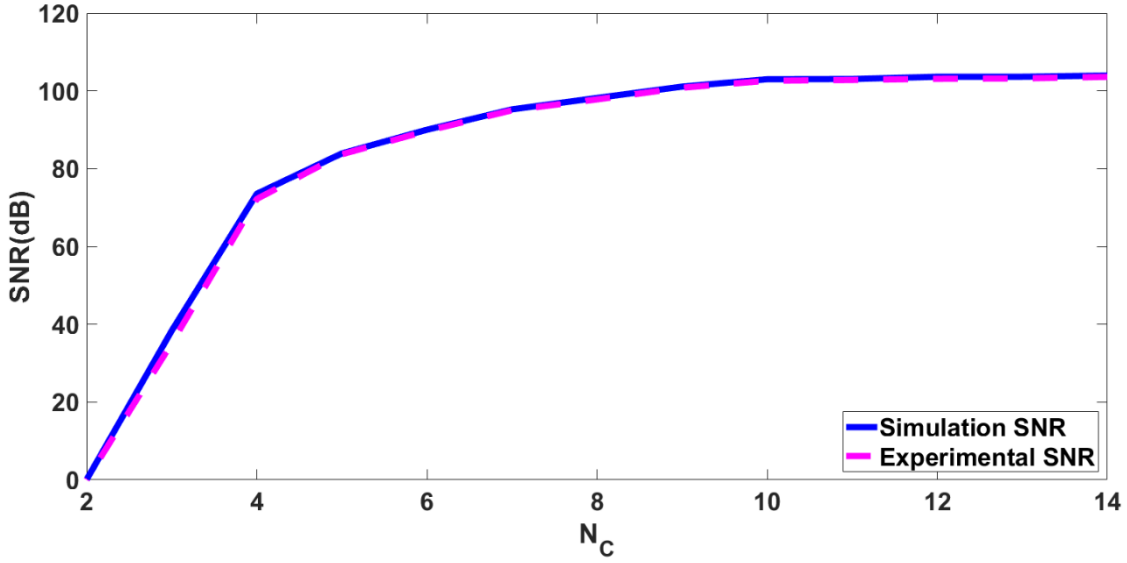

Fig.6.7. A comparison of the peak SNR vs bit partitioning N_C between simulations and experiment for the 16-bit example in sub-section 5.1.2.

6.2.2. 20-Bits Partitioning Example

The same process as that of the previous sub-section was repeated, albeit for the 20-bit example from section 5.1.2.

Once again, the simulations regarding the effects of partitioning on modulator performance are quite accurate, only further validating the design principles discussed earlier.

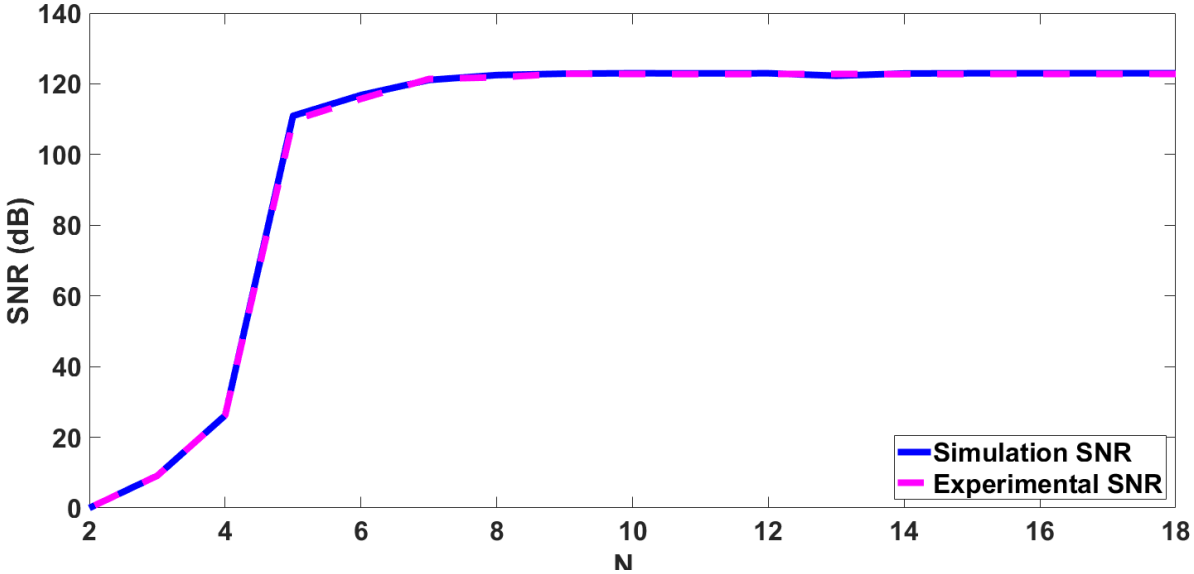

Fig.6.8. A comparison of the peak SNR vs bit partitioning N_C between simulations and experiment for the 20-bit example in sub-section 5.1.2.

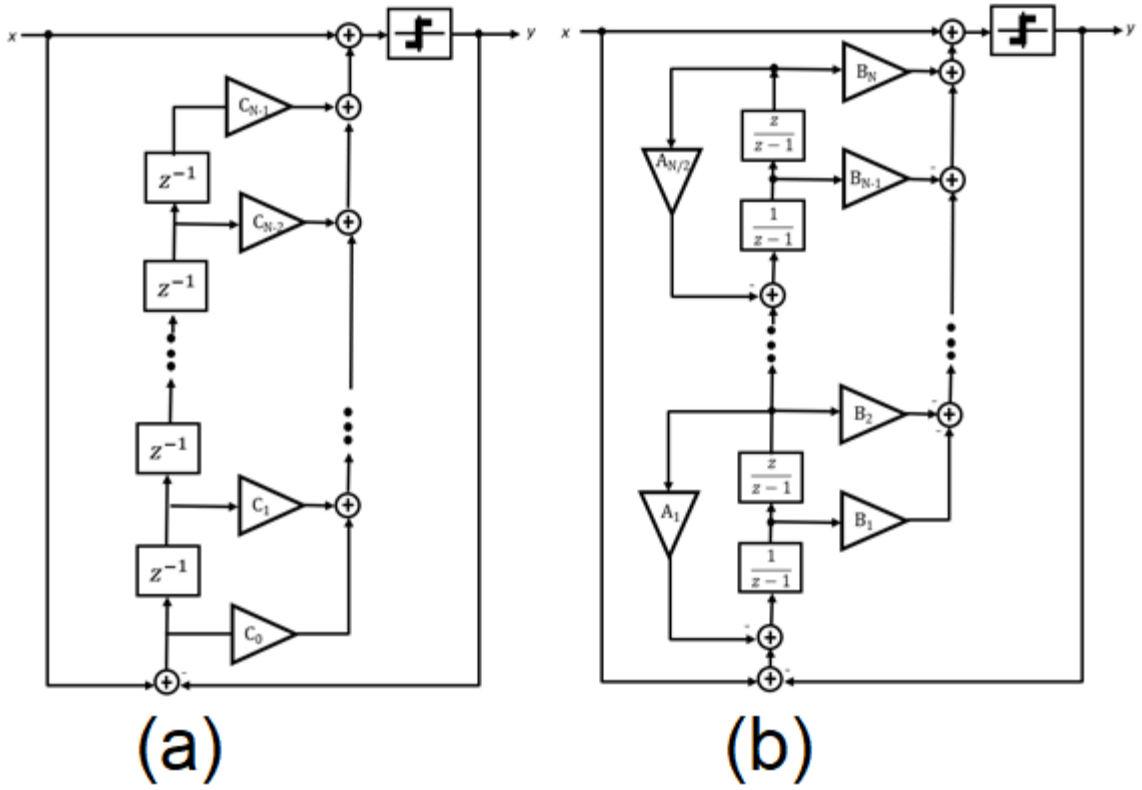

Fig.6.9. Direct-Form (a) and Resonator-Cascade Form (b) of a ΔΣ Modulator with unity STF

6.3. Validating Findings on Topology Comparison

For this section, the designs from section 5.2.4. were mapped onto an FPGA to validate the simulation findings. This entails a comparison of the structures shown in Fig.6.9.

6.3.1. Set-up

The circuit was excited with the same input sine waves used to create the periodogram of the PSD results found in Fig. 5.11. The FPGA will produce two simultaneous bitstreams for the two separate designs. This is to ensure that both are using the same input data for fair comparison. The output signals were once again sampled using the Agilent DSA80000B Digital Signal Analyzer. The data set was subsequently analyzed using MATLAB.

6.3.2. SNR comparison

The results of this experiment are shown in Fig.6.10. As with the simulations, the resonator-cascade topology produces a more desirable passband noise-shaping response, i.e., notches are well defined, albeit shifted slightly in frequency. The results are consistent with what was found with simulations.

It is curious to note that the Resonator-Cascade modulator structure deviates noticeably in the second “trough” at what was once the $5*10^{-3}F_s$ mark in simulations. It is now located closer to $4.9*10^{-3}F_s$, which may be due to input data not quite matching the simulations.

6.4. Hardware Costs/ Mapping

This section is dedicated to the comparison of the costs to implement the various topologies and partitions shown in previous sections.

6.4.1. Hardware Costs for Different Partitions

It has been observed previously in this thesis that the performance of a segmented ΔΣ DAC matches theory well, until a critical breakdown when the coarse segment receives a minority of the data bits.

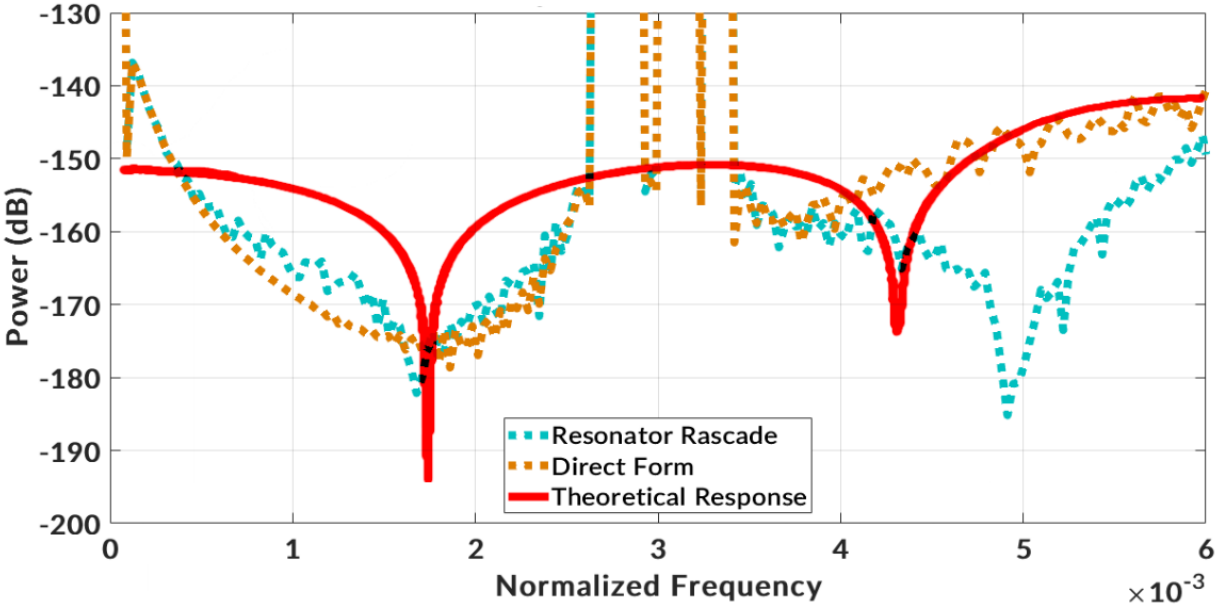

Fig.6.10. A comparison of PSD for direct form and Resonator-Cascade realizations, as compared to theory.

This alone, however, does not give a designer full insight into choosing how to partition data. The main draw of segmentation is relaxing the hardware costs for a relatively small decrease in SNR. The hardware requirement for a ΔΣ modulator increases exponentially as its order increases linearly. It may be tempting to think that the hardware cost of a segmented ΔΣ modulator is direct in-between of the costs of either modulator on its own. Figures 6.11. and 6.12 demonstrate the total number of logic elements (TLE) used in a given modulator partition, using the same modulators as those of sub-sections 6.2.1 and 6.2.2 respectively, as compared to an unsegmented ΔΣ modulator.

For the 16-bits partitioning example, the curve is comprised of two mostly linear sections, with a break at N_C of 8. The slope corresponding to N_C from 2 to 8 bits is less than the slope from 9 to 14 bits. This can be explained by the fact that on an FPGA the logic elements are grouped in 8-, 16-, or 32-bit cells. More efficient designs result when a lower resolution cell is used to its full capacity. For instance, at N_C=8, the course and fine modulators can be implemented with two 8-bit cells. However, at N_C=9, an 8-bit cell and a 16-bit cell is necessary. However, the 16-bit cell has many unused logic elements, decreasing its hardware efficiency.

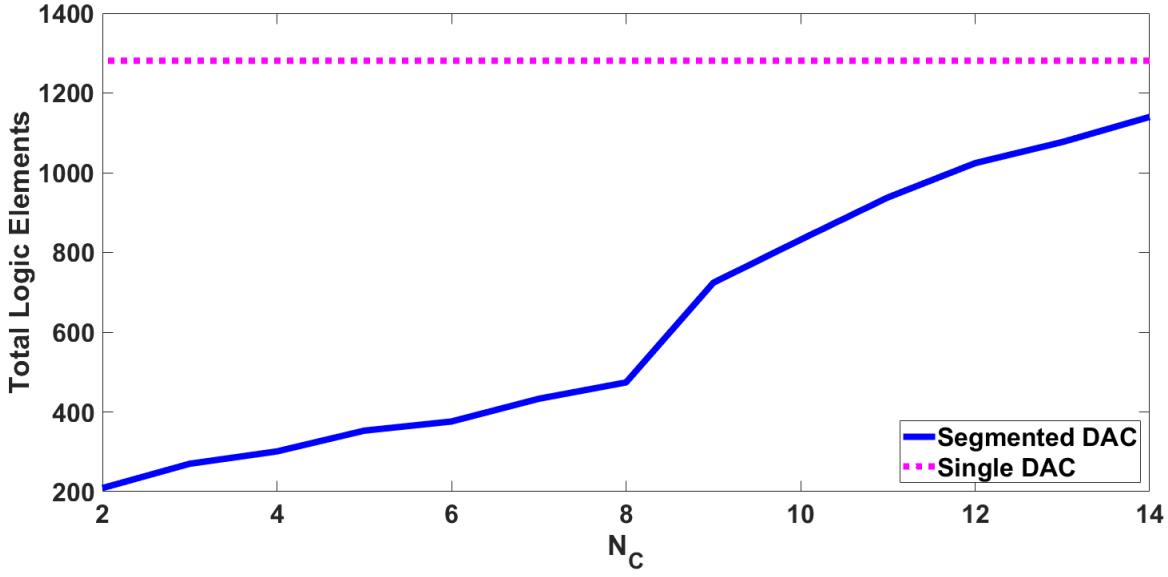

Fig.6.11. A comparison of total logic elements for different partitioning of data, using the 16-Bits Partitioning Example

The 20-bits partitioning example displays behavior that is not too dissimilar, albeit with two highly linear sections where N_C goes from 2 to 8, and once more from 12 to 18. This second segment can also be thought of the segment where N_F goes from 8 down to 2. The same logic from the 16-bit case can be applied here as well. The rather sharp increase in TLE can be explained by the fact that a larger cell must now be used for the coarse segment, while the fine segment maintains its use of one 16-bit cell.

6.4.2. Hardware costs for different topologies

As was discussed the latter half of section 5.2., the resonator-cascade topology can replicate the same type of performance as the direct form topology, while mostly using arithmetic shifts rather than multipliers. As can be surmised, this ends up being much less taxing in terms of the number of logic elements used to realize the linear filter. Table 6.2 presents a breakdown of the logic elements used by the two topologies for the two previous examples. Here, the difference in the total logic elements is quite significant, being an order of magnitude smaller in the case of resonator-cascade topology. The greatest savings achieved by using resonator-cascade topology are found in the combinational logic (the bulk of logic elements) and registers.

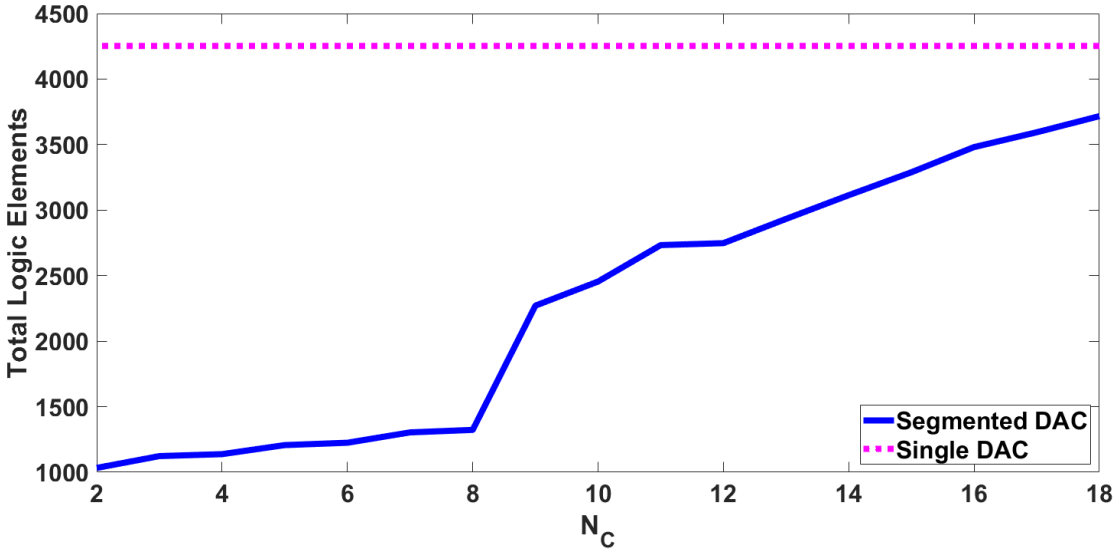

Fig.6.12. A comparison of total logic elements for different partitioning of data, using the 20-Bits Partitioning Example

Table 6.2.

A Breakdown of Logic Element usage, depending on Topology of Modulator

Modulator	Total Logic Elements	Combinational Only	Register Only	Combinational with Register
Direct Form	3304	3104	49	151
Resonator-Cascade	487	359	0	128

Here, the difference in the total logic elements is quite significant, being an order of magnitude smaller in the case of resonator-cascade. The greatest savings achieved by using resonator-cascade are found in combinational logic (the bulk of logic elements) and the registers, as it uses none on their own. The breakdown is even more revealing when looking at table 6.3, as it shows a breakdown of the logic elements by how many LUT inputs they have.

Table 6.3.

A Breakdown of Logic Element usage by number of LUT inputs, depending on Topology of Modulator

Modulator	Total Logic Elements	4-Input Functions	3-Input Functions	2-Input Functions
Direct Form	3304	8	1476	771
Resonator-Cascade	487	8	333	146

The 4 input functions are equal in number, with the 3 and 2 input functions sporting similar disparities to those of the number of logic elements. This can largely be attributed to the lack of multipliers used in the case of resonator-cascade, as seen in the schematic representation found in FIGURE.

It should also be noted that [18] found some cases in which a single arithmetic shift unit was not enough to provide a stable modulator that met the necessary frequency response. For these cases, approximating the feed-forward factors with a sum of two shifts (two-term CSD) is found to be enough.

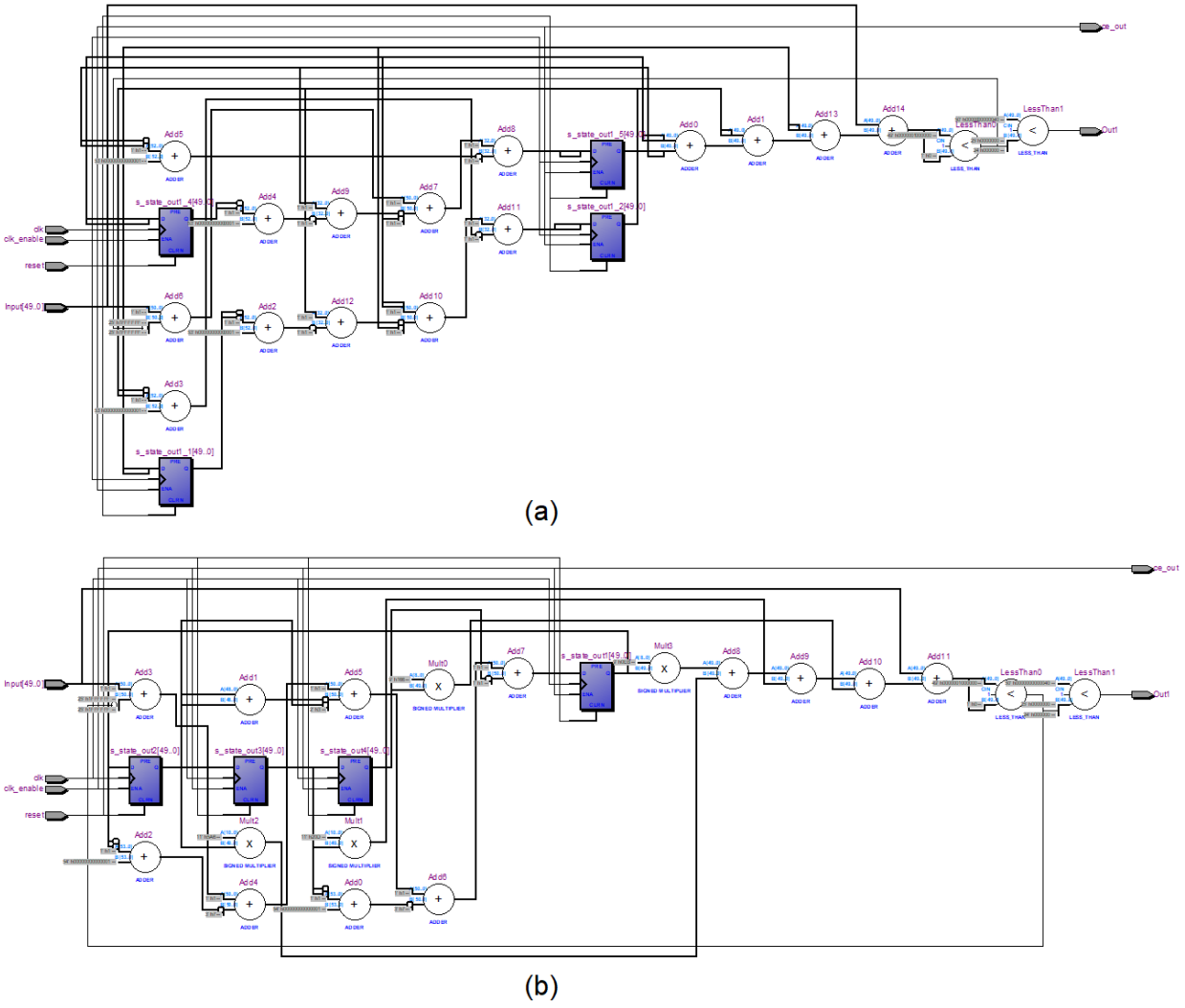

Fig.6.13. Schematic view of resonator-cascade (a) and direct form (b) realizations on FPGA, as produced by the Quartus RTL viewer

6.5. Conclusion

This chapter cemented the theory and simulations from the preceding chapters in this thesis, showing that segmentation in ΔΣ DACs can be used to great effect to reduce hardware costs while maintaining reasonable performance. It was also confirmed that Resonator-Cascade is a desirable option for modulator realization, as it maintains the expected response while also costing far less to implement than direct-form.

Chapter 7: Conclusion

7.1. Discussion of Results

The goal of the research performed and presented in this work has been to explore the effects of segmentation in ΔΣ DACs. In the beginning stages of the research, several types of improvements that were sought in performing this research. These were an improvement to DAC performance as measured by its SNR, the relaxation of the reconstruction filter by creating fewer samples per pass, and hardware relaxation.

While those last two were met, the first of these was quickly dropped upon realizing the additive noise effects of the particular segmented ΔΣ DAC structure that was being used. This thesis presents only some of the work that was performed.

More than a mere collection of data taken from mathematical analysis, simulations, and experiments, this thesis is meant to create a method by which segmented ΔΣ DACs can be created, using a given design target in mind.

The first two chapters acted as an introduction to the subject of the thesis. The third chapter used mathematics to determine the effects of segmentation on a

ΔΣ DAC, and found that in general, a design would end up with a slight decrease in performance, even with a significant “downgrade” in the fine segment.

The fourth chapter served as a formalization of the results of the third chapter, to create a method for the design of a segmented ΔΣ DAC, allow for an engineer to extract a design from a target SNR, rather than deriving the performance from the specifications of a DAC.

The fifth chapter solidified the conclusions of its preceding chapter as well. Using simulations, additional non-idealities that are part-and-parcel with practical circuits were taken into account. This led to the understanding that the segment leading the noise response should still be given most of the input data for it to perform correctly, in addition to the practical limitations on which modulators could be used for the non-leading segments. This chapter also served to demonstrate the varying performance of different modulator realizations. Using tools to export designs from simulations (such as Simulink) was shown to be sub-optimal. It was shown that, instead, a designer should consider a different realization as it results in a noise response that is closer to theory.

The sixth chapter served to validate its preceding chapter. Through various on-bench tests of the various DACs from previous chapters, it was found that the simulations are quite accurate in terms of performance. It was also found that there are indeed significant savings to be made in using segmentation in ΔΣ DACs, as a large portion of elements can be saved (typically around 50% to 10% for the functional DACs shown here). The validity of the Resonator-Cascade topology was also further cemented here, with its response being quite close to what was expected from theory.

7.2. Direction for Future Work

While much was accomplished in writing this thesis, there was a substantial reduction in the scope of the work.

Initially, the aim of the work was to create an automated design tool, likely in the form of a MATLAB package, to facilitate design. It was meant to build upon a similar tool which served as inspiration for this work; Xavier Haurie's DSMOD. This tool would have included the option to select among a different number of segments, allowed for a designer to choose from a range of filter realizations (not just resonator cascade), and recommended different partitioning schemes based on their logic element usage versus likelihood of instability.

Unfortunately, this was never developed, as time constraints proved to be too great. That being said, all of the elements mentioned above could all be potential improvements on the work. Even if such a tool is not developed, one could explore the possibility of using multi-bit segmented modulators, different loop filter realizations, etc.

Bibliography

[1] H. Inose, Y. Yasuda and J. Murakami, "A Telemetering System by Code Modulation - Δ- Σ Modulation," in IRE Transactions on Space Electronics and Telemetry, vol. SET-8, no. 3, pp. 204-209, Sept. 1962, doi: 10.1109/IRET-SET.1962.5008839.

[2] R. Steele, "Delta Modulation systems", London: Pentech Press Ltd., 1975.

[3] J. Bryant and W. Kester, “Data converter architectures” in The Data Conversion Handbook, W. Kester, 3rd Edition. Oxford, U.K.: Elsevier/Newness, 2005, pp. 147-174.

[4] G. Roberts and M. Burns, "An Introduction to Mixed-Signal IC Test and Measurement", Oxford University Press, 2005.

[5] B.D. Smith, "Coding by Feedback Methods," Proceedings of the I. R. E., Vol. 41, August 1953, pp. 1053-1058.

[6] W. Guo, T. Abraham, S. Chiang, C. Trehan, M. Yoshioka and N. Sun, “An Area- and Power-Efficient I_{ref} Compensation Technique for Voltage-Mode R-2R DACs,” in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 7, pp. 656-660, July 2015.

[7] Y. Li, T. Zeng, and D. Chen, "A high resolution and high accuracy R-2R DAC based on ordered element matching," in IEEE International Symposium on Circuits and Systems, Beijing, China, May 2013, pp. 1974-1977

[8] D. Seo, “A Heterogeneous 16-bit DAC Using a Replica Compensation,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, no. 6, pp.1455-1463, July 2008.

[9] J. A. Schoeff, "An Inherently Monotonic 12 Bit DAC," IEEE Journal of Solid State Circuits, Vol. SC-14, No. 6, December 1979, pp. 904-911.

[10] Y. Cheng, C. Petrie, B. Nordick, D. Comer, D. Comer, “Multibit Delta-Sigma Modulator With Two-Step Quantization and Segmented DAC”, IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 53, no. 9, September 2006

[11] N. Sun, “High-Order Mismatch-Shaped Segmented Multibit ΔΣ DACs With Arbitrary Unit Weights”. IEEE Transactions on Circuits and Systems—I: Regular Papers, Vol. 59, no. 2, February 2012

[12] Y. Wang, T. He, P. Silva, Y. Zhang and G.C. Temes “Wide-band high-accuracy ΔΣ ADC using segmented DAC with DWA and mismatch shaping”, Electronics Letters 25th May 2017 Vol. 53 No. 11 pp. 713–714

[13] L. He, Y. Zhang, F. Long, F. Mei, M. Yu, F. Lin, L. Yao, X. Jiang, “Digital Noise-Coupling Technique for Delta–Sigma Modulators With Segmented Quantization”, IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 61, no. 6, June 2014

[14] L. He, G. Zhu, F. Long, Y. Zhang, L. Wang, F. Lin, L. Yao and X. Jiang, “A Multibit Delta–Sigma Modulator With Double Noise-Shaped Segmentation”, IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 62, no. 3, March 2015

[15] R. Adams, K. Nguyen and K. Sweetland, “A 113-dB SNR Oversampling DAC with Segmented Noise-Shaped Scrambling”, IEEE Journal of Solid-State Circuits, Vol. 33, no. 12, December, 1998.

[16] R. Schreier, 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.

[17] R. Schreier, “Noise-Shaped Coding", Ph.D. Thesis, University of Toronto, Toronto, 1991.

[18] X. Haurie, "Signal Generation using High-Order Delta-Sigma Modulation," Master’s Thesis, McGill University, November, 1996.

[19] W. Lee, A Novel Higher Order Interpolative Modulator Topology for High Resolution Oversampling A/D Converters”, Master’s Thesis, Massachusetts Institute of Technology, June, 1987.

[20] A. Emara, "Designing Area-Efficient Programmable DC Voltage Generators Using Sigma-Delta Bitstreams For Testing Applications", Ph.D. Thesis Montreal, 2021.