NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

INÊS FERREIRA CASALEIRO NOGUEIRA DE BRITO

Bachelor in Electrical and Computer Engineering

STUDY OF A CALIBRATION SCHEME FOR A SIGMA-DELTA MODULATOR USING PASSIVE INTEGRATORS

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
ELECTRICAL AND COMPUTER
ENGINEERING

STUDY OF A CALIBRATION SCHEME FOR A SIGMA-DELTA MODULATOR USING PASSIVE INTEGRATORS

INÊS FERREIRA CASALEIRO NOGUEIRA DE BRITO

Bachelor in Electrical and Computer Engineering

Adviser: Prof. Nuno Filipe Silva Veríssimo Paulino
Full Professor, NOVA University Lisbon

Co-adviser: Dr. Blazej Nowacki
Senior Analog Designer, Renesas Electronics Corporation

Examination Committee:

Chair: Prof. José Manuel Fonseca
Full Professor, NOVA University Lisbon

Rapporteur: Prof. João Carlos da Palma Goes
Full Professor, FCT-NOVA

Adviser: Prof. Nuno Filipe Silva Veríssimo Paulino
Full Professor, NOVA University Lisbon

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2023

Study of a Calibration Scheme for a Sigma-Delta Modulator Using Passive Integrators

Acknowledgements

Firstly and foremost, I acknowledge and would like to thank dearly to my Adviser Prof. Nuno Paulino and Co-adviser Dr. Blazej Nowacki, for their shown support, availability, commitment and most of all, patience.

I am sincerely and deeply grateful to Dr. Blazej Nowacki, for his encouragement, motivation and guidance. More then a supervisor, he has also been a friend and has helped me join and integrate the group of Renesas Electronics Corporation.

I would like to state my gratitude to Prof. Nuno Paulino for his support and help, for always having an open door, and for all the useful remarks on the development of this project.

I would also like to acknowledge my gratitude to Prof. João Goes who has truly inspired and lead me into the pursue of studding electronics. I would like to acknowledge the Department of Electrical Engineering of the School of Science and Technology of Universidade NOVA de Lisboa, for the nurturing and cultivation of the consistent pursue of knowledge. On this remark I would also like to thank Prof. João Goes, Prof. Nuno Paulino, Prof. João Oliveira and Prof. Luís Oliveira, who have guided me in their unique ways and integrated me in the internship program that has lead me to Renesas.

I would like to aknowledge the Renesas Electronics Corporation for the opportunity to do research in state of the art technology, in the area of electronics and in particularly I would also like to thank Tommy Mulane and Tong Lu for all the support and technical help in this project. I would like to thank dearly to my fellow colleagues and friends from the Lisbon Office in Renesas, for all the support.

Finally, I would like give the most heartfelt thanks to my family for guiding and supporting me through out my life. To Francisco Soares Branco for his unconditional support, motivation, patience and above all for being by my side through difficult times, new adventures and hard decisions. A warm and special appreciation to my friends and people who have been by my side and have helped me in one way or another, during this and other journeys. Specifically to Alexandre Moreira, Ana Carlota Fonseca, António Mendes Pinto, Guilherme Santos, Henrique Cetra, Lourenço Rodrigues, Leonor Nogueira de Brito, William Lake, Madalena Menéres Pimentel and Maria Ana Gamito.

Abstract

In modern electronics systems there has been a great push to have a constant size reduction of Complementary Metal-Oxide-Semiconductor (CMOS) technologies which can have different consequences, such as a reduction of the intrinsic gain of CMOS transistors and an increase of their transition frequency. This has resulted in increasing the difficulty of designing high gain, high bandwidth low power amplifiers. This can be a problem when designing high performance Sigma-Delta Modulators (ΣΔMs). Recently, it was proposed to use active-passive ΣΔMs architectures, where the loop gain is mainly obtained from the comparators processing gain, thus reducing the gain required for the amplifiers. Multi-stAge Noise SHaping (MASH) structures present a stable alternative to high-order single-loop typologies. Nevertheless, typically, these are very sensitive to the analog circuit imperfections, since their performance relies on the accuracy of the values of the analog circuit's transfer functions matching with the coefficients in the Digital Cancellation Logic (DCL) in order to avoid noise leakage and achieve their desired performance. This is the case with the passive 2-1 MASH ΣΔM developed based on [1], which has it's performance limited by process variations of the values of it's constituted components. In the used CMOS technology, process variations can cause the $R \cdot C$ value to change up to 37%, which causes mismatch between the ΣΔM analog transfer functions and the DCL transfer functions and also causes a variation in the overloading voltage of the modulator. Both these issues can cause degradation in Signal-to-Noise-and-Distortion Ratio (SNDR) of the modulator. In order to solve these problems, this thesis proposes a circuit capable of measuring the value of $\frac{RC}{T_S}$ and then using this value to calibrate the coefficients in the DCL to match the analog transfer functions and adjust the reference voltage (V_{ref}) in order to reduce the variation of the ΣΔM overload voltage. In order to validate the calibration concept, the performance of 2-1 MASH ΣΔM implemented with and without calibration was analyzed by performing process, voltage and temperature (PVT) corners and Monte Carlo simulations. The results show that by implementing the calibration it was possible to reduce the variability of the SNDR results, as well as increase worst corners results.

Keywords: Analog to Digital Converter (ADC), Sigma-Delta Modulators (ΣΔMs), 2-1 MASH ΣΔM, Passive Integrators, RC Time Constant Calibration, RC Time Constant.

Resumo

Ao longo do tempo, a tecnologia *CMOS* tem evoluindo de forma a permitir a progressiva redução das dimensões dos transístores. Esta redução permitiu o aumento da frequência máxima de operação (frequência de transição) dos transístores, mas resultou também na redução do seu ganho intrínseco. Assim, desenvolver amplificadores de elevada largura de banda e ganho, com baixos níveis de potência dissipada tornou-se um grande desafio. O que pode representar um problema no desenvolvimento de Moduladores Sigma-Delta ($\Sigma\Delta$Ms) de alto desempenho. Recentemente, tem-se proposto a utilização de arquiteturas $\Sigma\Delta$ passivas ou híbridas activas-passivas, em que o ganho de malha do modulador depende, em grande parte, do ganho de processamento do comparador, evitando a necessidade de amplificadores de alto desempenho. O que é uma vantagem na implementação das estruturas em cascata (*MASH*), que, tipicamente, são mais sensíveis às imperfeições da parte analógica do circuito. Isto deve-se ao facto do seu desempenho estar altamente dependente da correspondência entre as funções de transferência do circuito analógico e os coeficientes registados na lógica de cancelamento digital (*DCL*). Este é o caso do modulador $\Sigma\Delta$ desenvolvido com base na tese [1], que tem o seu desempenho limitado pelas variações de processo que afetam as suas componentes. Estas variações podem causar uma variação $R \cdot C$ máxima de 37%, provocando alterações no ganho do modulador e uma divergência entre os coeficientes analógicos (provenientes das funções de transferência do circuito) e digitais (*DCL*), o que resulta na degradação no desempenho do circuito. Assim, nesta tese propõe-se um método de calibração baseado num circuito capaz de medir a variação de $\frac{RC}{T_S}$, de forma a converter esta variação num coeficiente digital capaz de calibrar os coeficientes da *DCL* de forma a coincidirem com as funções de transferência analógicas e calibrar a tensão de referência do *ADC*, de modo a diminuir a variabilidade da sua realimentação. De forma a validar o proposto método de calibração, o modulador $\Sigma\Delta$ foi simulado com e sem calibração, tendo em conta as variações de processo, temperatura e diferença de potencial máximas e também de Monte Carlo. Estas simulações demonstraram a capacidade do circuito de calibração em reduzir a variabilidade dos resultados SNDR, tal como aumentar o desempenho do ADC, nas piores condições.

Palavras-chave: Conversores Analógico para Digital (*ADC*), Moduladores Sigma-Delta, Integradores Passivos, Calibração da Constante de Tempo RC, Constante de tempo RC.

Contents

List of Figures

List of Tables

List of Listings

Acronyms

A/D Analog to Digital 1, 5
ADC Analog to Digital Converter 1

CMOS Complementary Metal-Oxide-Semiconductor vii, 2

DAC Digital to Analog Converter 1
DCL Digital Cancellation Logic vii, 2
DNL Differential Non-Linearity 12
DR Dynamic Range 13

ENOB Effective Number of Bits 14

FIR Finite Impulse Response 40
FoM Figure of Merit 14

INL Integral Non-Linearity 12

LUT Look-Up Table 40

MASH Multi-stAge Noise SHaping vii

NMOS N-type Metal-Oxide-Semiconductor 66
NTF Noise Transfer Function 16

OSR Oversampling Ratio 8

PMOS P-type Metal-Oxide-Semiconductor 3
PSD Power Spectral Density 11

PVT Process, Voltage and Temperature 3

S/H Sample and Hold 9
SC Switched-Capacitor 20
SFDR Spurious Free Dynamic Range 13
SNDR Signal-to-Noise-and-Distortion Ratio vii, 14
SNR Signal-to-Noise Ratio 13
STF Signal Transfer Function 16

THD Total Harmonic Distortion 13

UIS Ultra Incomplete Settling 25

1

Introduction

1.1 Background and Motivation

All the information that can be obtained from the physical world, such as light and sound, can be reduced to analog signals. These can be defined as signals that are continuous in time and can take, at any moment, an infinite amount of values.

Signals can be transmitted from one point to another, enabling the use, analysis and transmission of the information contained in them. Digital circuitry provides the computational and signal processing power necessary to analyse these signals, however, it can not interact directly with the physical world. In the computer age, every electronic device and digital core can only process a discrete signal that is represented as bits of information - digital signal. Thus the need for data converters, there is the Analog to Digital Converter (ADC) and the Digital to Analog Converter (DAC).

In the present day, ADC's are some of the most significant building blocks utilized in different types of electronic circuits, that can have medical, instrumentation, and communication applications. There are many different ways to implement an Analog to Digital (A/D) data converter. For this reason there are different types of ADC structures that can be selected to accommodate different desired requirements, such as bandwidth and resolution, for example. The main objective of an ADC is to convert analog information into a digital signal, and it's performance can have a huge impact on the speed and efficiency at which it does so [1].

Sigma-Delta ($\Sigma\Delta$) modulation has it's origins in the early 1960s and has evolved immensely since then. In fact, in the present day stands as one of the most common and popular methods to develop a a high-performance, high-resolution, A/D converters, and this is due to the fact that it exploits oversampling and digital signal filtering to generate a high resolution digitized output [2] [3].

However, there are other important requirements. As modern electronics systems need an energy efficient operation, there has been a great influence to decrease the levels of energy consumption in all different types of building blocks, ADC's included. Furthermore, there has also been a great push to have a constant size reduction of Complementary

Metal-Oxide-Semiconductor (CMOS) technologies which resulted in a reduction of the intrinsic gain of CMOS transistors and an increase of their transition frequency. It has become a great challenge to design high bandwidth and gain amplifiers with low levels of power dissipation. And, as a repercussion, designing the ADCs that depend upon such amplifiers, as Sigma-Delta Modulators ($\Sigma\Delta$Ms) do, is no exception to this challenge [1].

One approach that can be highly effective into dealing with this problem is to take advantage of the mixed active-passive or passive Sigma-Delta architectures. In the case of deep nanometer (nm) CMOS technologies, this can be very tempting as it allows the comparator to achieve a large gain through positive feedback, which improves with faster transitions. One downgrade that this approach can have is that the passive integrators have a higher sensitivity to the impacts of noise coupling, than the active ones. In spite of this, a passive $\Sigma\Delta$Ms architecture allows for the design of high performance $\Sigma\Delta$Ms without the need for high performance amplifiers and allowing for significant power reduction [4] [5] [6].

Therefore, this project starts by focusing on the study of a continuous-time 2-1 Multi-stAge Noise SHaping (MASH) $\Sigma\Delta$M, developed in [1], using passive RC integrators and simple differential pairs as low-gain blocks, to better understand how it takes advantage of the passive $\Sigma\Delta$ architectures to achieve a competitive power efficiency amongst other relevant works.

After careful analysis it was realized that this $\Sigma\Delta$M has it's performance limited by process variations of the values of its constituted components. Since this modulator relies on matching the values of the analog circuit with the coefficients in the Digital Cancellation Logic (DCL) to achieve its desired performance, any deviation from the ideal components values causes a degradation.

Therefore, the purpose of this project is to study the implementation of extra circuits that can measure the value of the relevant analog circuit's parameters and convert these values into a digital coefficient that can be used in the digital cancellation logic to compensate the variations of the analog parameters, in the performance of a 2-1 MASH$\Sigma\Delta$M based of [1]. Additionally, it was realised that it is also possible to correct the ADCs feedback variation by adjusting its reference voltage using the measured digital coefficient. Thus, these two calibrations will be studied in order to see what kind of improvement can be obtained in the performance of the MASH$\Sigma\Delta$M, by implementing these two solutions.

1.2 Contributions

This thesis presents a calibration method for a two stage 2-1 MASH $\Sigma\Delta$M, with 14-bit resolution, 12 MHz bandwidth, and 1.6 GHz clock frequency. The architecture of this ADC originated from [1]. Since the publication of [1], the 2-1 MASH $\Sigma\Delta$M has been redesigned with new specifications and requirements, and therefore it was implemented in a UMC CMOS 40 nm Low-Power (LP) process technology, clocked at a frequency of $F_S = 1.6GHz$, a bandwidth of 12 MHz and a 1.1 V supply voltage.

After careful analysis and study of the behaviour of the ΣΔ modulator, a calibration method was proposed in order to compensate the RC time constant variation in the passive integrators, and consider it as an important factor that affects the performance of the modulator in different circumstances.

In order to validate this calibration methodology, a High-Level Model of the system was developed. This model determines a correction factor from the RC time constant value (including deviation) and then uses this factor to ajust the ADC's reference voltage and calibrate the DCL coefficients. The calibration of the DCL is done by recalculating coefficients of the modulator's transfer functions, considering the effects of the correction factor previously determined by calibration. As the results from high-level simulations were encouraging, a calibration circuit was developed in the same technology as the modulator.

This modulator was designed based on genetic algorithms used to minimize the impact of analog parameters variations on it's performance [7]. This optimization takes into consideration the variation of all the analog parameters in the circuit and finds the solution with the lowest sensitivity that maximizes the SNDR for a desired maximum differential input amplitude voltage ($V_{in,diff}$). An objective of the optimization process was to ensure proper functionality of the modulator in all Process, Voltage and Temperature (PVT) corners, considering a maximum $V_{in,diff}$ of 0.75 V. This caused a selection of a design point (circuit sizing) to be conservative. And as the modulator was originally optimized for a reference voltage of 1.1 V, calibrating the reference voltage (V_{ref}) through corners variation allows to increase the dynamic range of the ADC. Nevertheless, adjusting the V_{ref} causes problems in the reference switches, which are implemented as P-type Metal-Oxide-Semiconductor (PMOS) transistors.

Therefore in order to perform relevant electric simulations of the proposed calibration method, it is necessary to do some modifications to the 2-1 MASH ΣΔM. These modifications consist of redesigning the passive integrators, as significant alterations are made in the switches and feedback capacitors in order to calibrate the V_{ref}.

After implementing these modifications, it was possible to improve the worst corner performance of the 2-1 MASH ΣΔM, by implementing this calibration, and obtain a smaller variation of the ΣΔM across corners, resulting in a more robust system.

1.3 Thesis Layout

This thesis is composed of six chapters, including the present introduction. The chapters are structured as follows:

- **Chapter 2 - State of the Art**
  This chapter presents an introduction to the fundamentals of data conversion, some basic definitions of ΣΔ modulation and common methods to improve high accuracy

and performance of ADC ΣΔ modulators while maintaining a low power consumption. Then an overview of different calibration techniques is presented, with state of the art examples of recently published works.

- **Chapter 3 - Analysis of High-Level Simulations**
  An analysis of the 2-1 MASH ΣΔM, based of [1], is presented for a general context and to better understand the architecture chosen to perform the necessary calibration. The proposed High-Level Model of the calibration architecture is presented, with the respective *MATLAB* ® simulations results of its theoretical variation and impact on the performance of the 2-1 MASH ΣΔM.

- **Chapter 4 - Electric Circuit and Simulations**
  This chapter presents the developed electric circuits schematics to achieve the purpose of this thesis, and their respective performance.

- **Chapter 5 - Validation of the Calibration Concept**
  In this chapter, a comparison between the modulator's performance metrics, before and after implementing the calibration, is presented, in order to validate the proposed calibration concept. Additionally, the modifications done to the 2-1 MASH ΣΔM are also presented.

- **Chapter 6 - Conclusions and Future Work**
  In the final chapter, conclusion are drawn regarding the obtained results and some suggestions for future work are proposed.

2

State of The Art

In this chapter, an overview of the theoretical aspects, important for the better understanding of the content stated in this thesis and it's general context, are presented.

Firstly an introduction to the fundamentals of data conversion is presented, including a description of an ADC structure and important performance metrics. Moreover, some basic definitions of ΣΔ modulation are presented including the description of different architectures and methods to improve the modulator's high accuracy and performance while maintaining a low power consumption. Some examples of the described architectures are also presented.

Finally an overview of different calibration techniques is presented alongside some examples of recently published works in this area.

2.1 Data Conversion Fundamentals

This section presents the basic definitions of A/D conversion.

As mentioned before, an analog signal can take, at any moment, an infinite amount of values, and because it is impractical to process infinite samples, it is essential to obtain a finite number of values, or bits of information, that can translate the information contained in the analog signal, to the digital domain. This is accomplished by the use of ADC blocks (Fig.2.1).

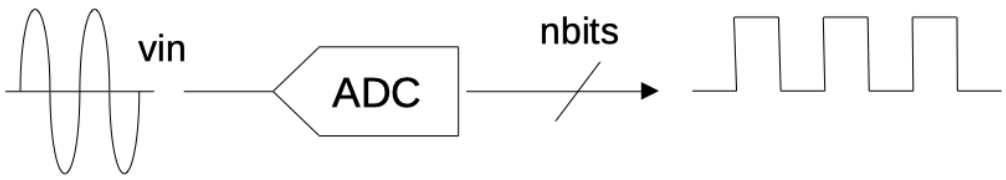


Figure 2.1: Basic diagram block of an ADC.

As it can be seen, one of the main tasks of this building block is to sample the signal

periodically, with a specific time interval, T_s, thus converting a continuous signal into a discrete one. This can also be described as the quantization process.

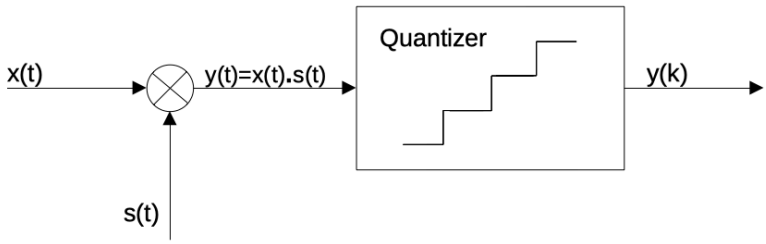


Figure 2.2: Sampling and quantization process.

The process of sampling illustrated by this diagram block, can be explained in the time domain, where the input analog signal x(t) is multiplied by the following sampling function, s(t) (expression 2.1) which allows to obtain the sampled and quantized output signal, y(k) [8].

$$s(t) = \sum_{n=-\infty}^{\infty} \delta \cdot (t - n \cdot T_s) \tag{2.1}$$

As described in Equation 2.1, this function is composed of periodic Diracs delta pulses where the samples are separated by the constant period, T_s. The sampling process in the time domain is demonstrated in Figure 2.3. Although it should be mentioned that this does not illustrate the process of discretization of the amplitude of the signal, as it depends on the quantizer's resolution.

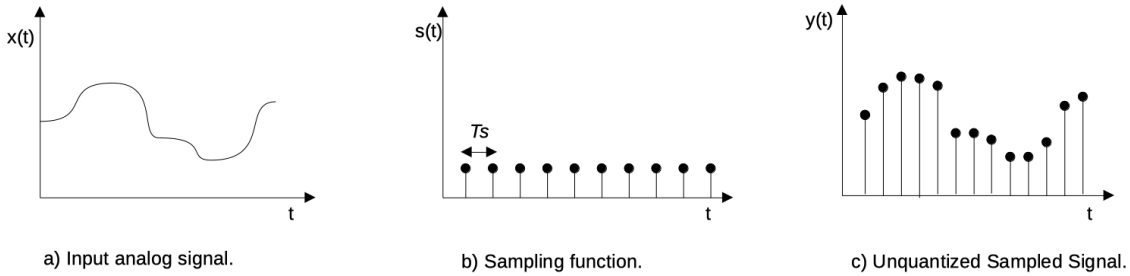


Figure 2.3: Time domain sampling process.

Another very important aspect that needs to be analyzed is the required sampling frequency value, in order not to lose signal information.

That is why the Fourier transform is used, because it decomposes a signal into it's frequency's content. In the following equations, a representation of the continuous time and discrete time definitions can be seen, respectively.

$$f(\xi) = \int_{-\infty}^{\infty} f(x) \cdot e^{-i2\pi x\xi}, dx \tag{2.2}$$

$$X[n] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i2\pi kn/N} \tag{2.3}$$

One can see the effects of applying the Fourier Transform, directly when analysing Fig. 2.3 (b) and seeing that the former Dirac pulses, previously described in the time domain, are converted into Dirac pulses in the frequency domain, in Fig. 2.4 (b), with the same period, as $f_s = 1/T_s$ [8].

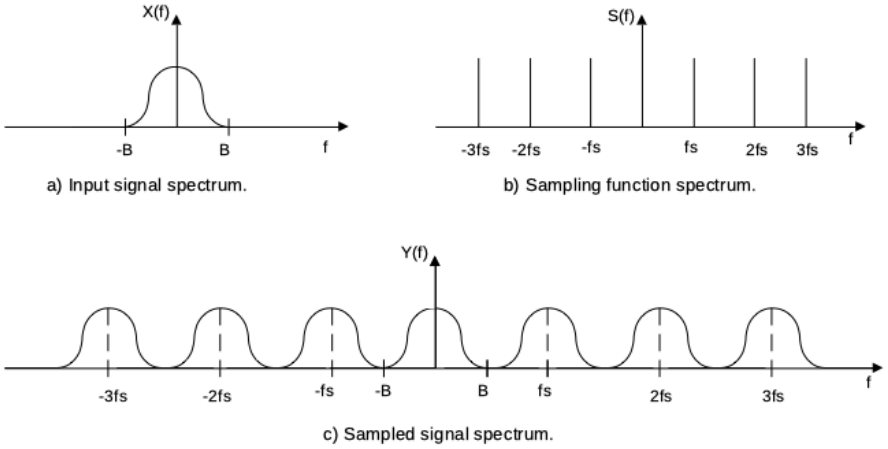


Figure 2.4: Frequency domain sampling process.

This is due to the convolution operation in the frequency domain. In fact, when two signals, represented in the time domain, are multiplied, as it happens in Fig. 2.2, it originates the Equation 2.5, which results in spectral signal replicas centered on integral multiples of the sampling frequency, f_s, as illustrated in Figure 2.4 (c) [8].

$$S(jw) = \frac{2\pi}{T_s} \sum_{n=-\infty}^{+\infty} \delta \cdot (w - k\frac{2\pi}{T_s}) \tag{2.4}$$

$$Y(jw) = \frac{1}{2\pi} X(jw) \otimes S(jw) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(jw - \frac{j2\pi k}{T_s}) \tag{2.5}$$

It should be noticed, by observing Figure 2.5, that the maximum signal frequency, B, defines the bandwidth of the input signal described in the frequency domain, and if the sampling frequency, f_s, is too low, it originates the aliasing effect which is depicted in Fig. 2.5 a). This effect is characterized by the overlap of the spectral repetitions.

To avoid this situation (as seen Fig. 2.5 (b)) the Nyquist (or Shannon) Theorem articulates that a signal can be sampled with no loss of information only if the maximum signal frequency (B) is smaller than half of the sampling frequency values ($F_s/2$), also known as the Nyquist frequency (Equation 2.6).

$$F_s = 2 \cdot B \tag{2.6}$$

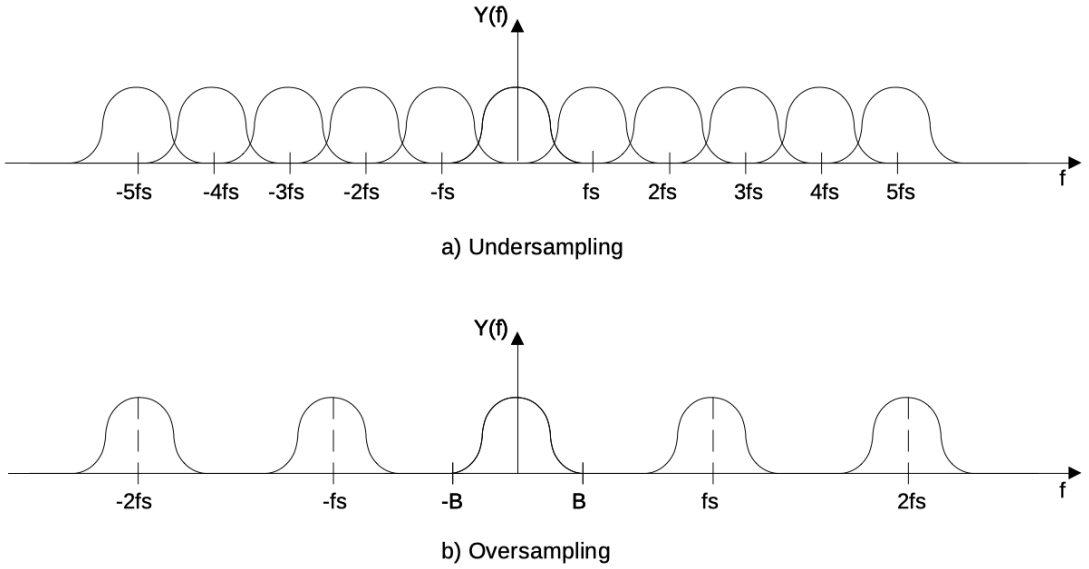


Figure 2.5: Over and under sampling in the frequency domain sampling.

If this condition is valid, then by using an ideal low pass filter with a cut off frequency equal to B, it's possible to recover the original signal. Usually the sampling frequency is higher than the minimum required value ($F_s/2$) in order to simplify the design of the low pass filters in the system.

2.1.1 Nyquist-rate and Oversampling Converters

There are many ways to characterize different types of data converters. One of the most common, is according to their Oversampling Ratio (OSR). Taking this into account, there can be the Nyquist-rate and the Oversampling converters.

The first kind of A/D converters sample the input signal at the Nyquist frequency, Equation 2.6, and there is a one-to-one correspondence between their input and output signals. The samples are processed separately and the converter has no memory. Each processed digital output word, i.e binary code (of given length), corresponds to the ADC's sequence (of the same length) of the input samples. In this case, the sampling and quantization operations limit the performance of this type of converters due to the fact that it's accuracy relies mostly on the matching of it's analog components, making it very challenging to achieve a high resolution. Furthermore, the conversion resolution of the Nyquist-rate converters depends on the quantizer's resolution [1].

On the other hand, the Oversampling converters have a higher sampling rate, $F_s >> 2 \cdot B$, typically by a factor of 8 to 512. The OSR is described as:

$$OSR = \frac{F_s}{2 \cdot B} \tag{2.7}$$

A ΣΔM, is a very common example of an Oversampling A/D data converter. In this case, opposite to the Nyquist-rate converters, a one-to-one correspondence between the

input and output signals, isn't necessary as these kind of converters require memory elements in order to use all the previous input samples to produce a valid output. Moreover, there are huge benefits in using a high oversampling ratio. As the transition band increases extremely for OSR»1, there is no need to implement an anti-aliasing filter with a sharp cut-off. This is due to the fact that no frequency components between $[B, F_s - B]$ can alias into the band of interest $[0, B]$, as illustrated in 2.7, which allows to avoid the introduction of phase distortion [9] [1].

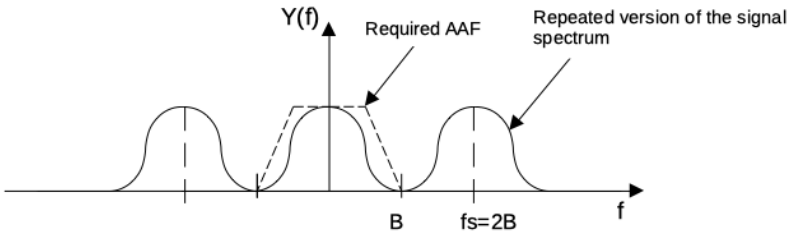


Figure 2.6: Anti-aliasing filter for Nyquist-rate ADC.

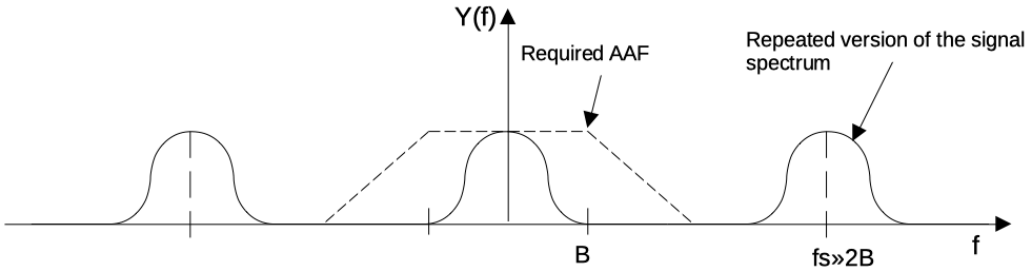


Figure 2.7: Anti-aliasing filter for Oversampling ADC.

2.1.2 ADC Structure

By now, some of the main components of a general ADC have been mentioned, such as the quantizer and the anti-aliasing filter. However, in order to have a successful and realistic diagram block of an ADC, there a few blocks missing, such as the Sample and Hold (S/H) block, and a binary encoder, that are illustrated in Figure 2.8.

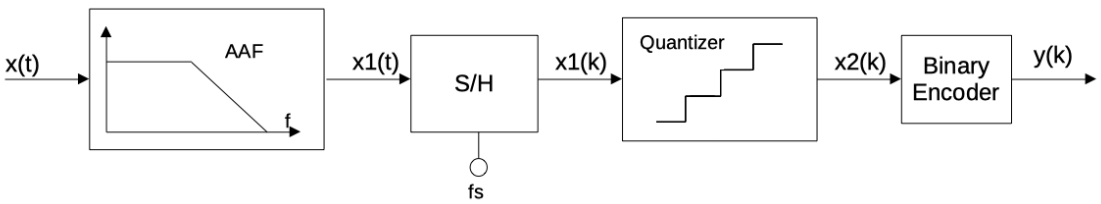


Figure 2.8: A generic block diagram for an ADC.

The AAF is used to prevent folding and the aliasing effect (previously mentioned). This block is then followed by a S/H block that samples the CT signal, x1(t), at a sampling

frequency, F_s, generating the DT signal x1(k), by performing the time discretization process, previously analysed. Then, the Quantizer provides an amplitude discretization of the this signal, x1(k), by mapping a continuous range of amplitudes of its input signal into a finite number of discrete levels. And, finally, each discrete level is converted into a certain unique binary code, by the binary encoder [1].

2.1.3 Quantization

One can define quantization as the amplitude discretization of an input signal into a set of discrete levels, and this process is accomplished by the quantizer. This is only possible due to it's staircase characteristic. An example of a quantizer's ideal staircase characteristic can bee seen in Figure 2.9.

It is important to understand that the resolution of the quantizer defines the number discrete levels presented on the staircase characteristic. For this reason, an N bit quantizer will have a digital output with 2^N discrete levels. Moreover, the difference between two consecutive quantized values, is called the quantization step, Δ, and can be calculated as $\Delta = Y_{MAX}/2^N - 1$ [8].

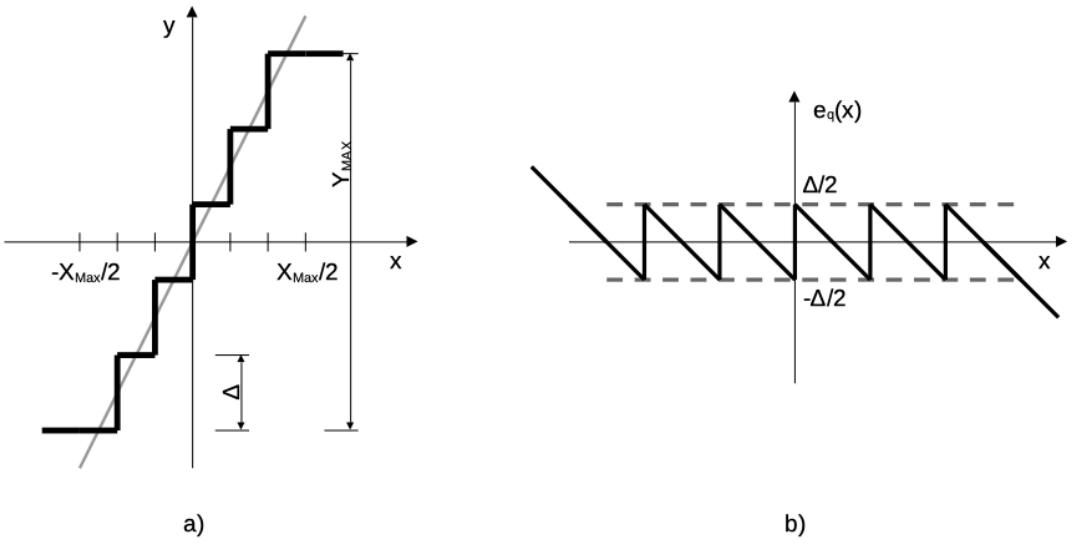


Figure 2.9: Representation of (a) the ideal staircase characteristic of the quantizer and (b) the quantization error.

The quantization error, e_q, is defined as the difference between the original value of the input signal and the quantized value. For all the inputs, in the range of X_{MAX}, as illustrated in Fig. 2.9 b), the e_q is limited to $\pm\Delta/2$, and it has the average value of zero. In addition, it is possible for the quantizer to become overloaded and its e_q grow beyond its limits for inputs exceeding X_{MAX} [1].

Assuming a rapidly variation of the input signal, e_q becomes uniformly distributed between $\pm\Delta/2$ (i.e constant over frequency) [10]. This allows for the error e_q to be modeled as white quantization noise and also the linearization of the quantizer, which is a

nonlinear element [1]. The Power Spectral Density (PSD) of the quantization noise power is defined as:

$$S_Q(f) = \frac{1}{F_s}\left[\frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e_q^2\, de_q\right] = \frac{\Delta^2}{12F_s} \tag{2.8}$$

And the in-band quantization noise power is given by:

$$P_Q(f) = \int_{-B}^{B} S_Q(f)\, df = \frac{\Delta^2}{12OSR} \tag{2.9}$$

Therefore, the PSD of the quantization noise power decreases, if the sampling frequency, F_s, increases, which means that the quantization noise will be spread out over a wider frequency band reducing the in-band noise, as depicted in Fig. 2.10 [8]. This can bee seen as an advantage of using oversampling converters because in that case, as seen previously, the sampling frequency range, F_s, is larger [1].

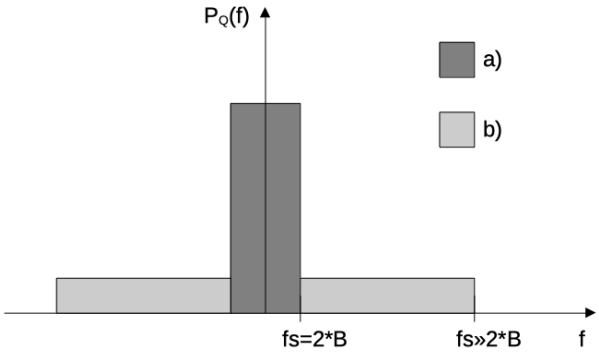


Figure 2.10: Spectral density of the quantization noise for: Nyquist-rate (a) and oversampling converter (b).

One can also notice that doubling the OSR (i.e. sampling at twice the rate) decreases the quantization noise power by 3 dB (one-half) [10].

2.1.4 Performance Metrics

In order to fully understand the different types of ADC architectures and important techniques that will be discussed further, some methods to characterized them need to be defined. Thus, the need for performance metrics, these enable the characterization and comparison of data converters. These performance metrics can be separated into two categories: static or dynamic [8].

Nonetheless, before going any further there's one basic concept that needs to be addressed due to its up most importance, the *lsb* voltage. The resolution of a quantizer is defined by the least significant bit voltage (V_{lsb}). This voltage defines the minimum conversion step of the quantizer. Specifically in an ADC, the V_{lsb} corresponds to the

minimum change to the input voltage that results in a change of the least significant bit of the output code, as represented by the following Equation.

$$V_{lsb} = \frac{V_{REF}}{2^{nbits}} \tag{2.10}$$

In Equation 2.10, the reference voltage, V_{REF}, defines the conversion range of the quantizer and the number of bits, *nbits*, defines the resolution of the quantizer.

Moving on, the static performance metrics that will be used through out this work are the following:

- **Offset error** - This error is measured in LSB and it can be defined as the difference between the real first transition level and the ideal one [8].

- **Differential Non-Linearity (DNL)** - The DNL error of a transfer function allows step by step comparison. It is the difference between the actual step size obtained and the ideal step size (1 LSB). This error is measured in LSB and it is often referred to as a local error (Fig. 2.11).

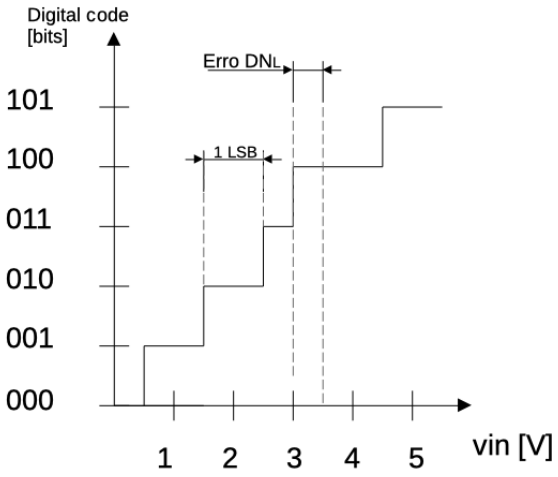


Figure 2.11: Example of a Differential Non-Linearity error.

- **Integral Non-Linearity (INL)** - The INL error is a global error, as it provides information about the converter's distortion (Fig. 2.12). This error is defined as the transfer function deviation from it's ideal straight line and is measured in LSB. It can also be known as the cumulative sum of the DNL values [8].

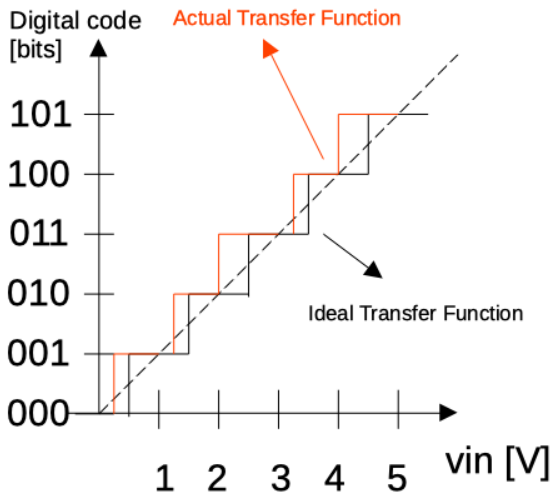


Figure 2.12: Example of a Integral Non-Linearity error.

Now follows the dynamic performance metrics:

- **Signal-to-Noise Ratio (SNR)** - This performance metric can be defined as the ratio between the signal power, P_S, and the noise power, P_N, at the output of the converter, for a certain input amplitude, A [9].

$$P_S = \frac{A^2}{2} \qquad [W] \tag{2.11}$$

$$SNR = 10 \cdot \log\left[\frac{P_S}{P_N}\right] \qquad [dB] \tag{2.12}$$

- **Dynamic Range (DR)** - The DR can be described as a ratio of the rms value of the smallest detectable input sinusoidal signal (SNR=0dB), to the rms value of the maximum amplitude (A_{MAX}) input sinusoidal signal, for which the structure still operates correctly [9].

$$DR = \frac{A_{MAX}^2}{2 \cdot P_N} \qquad [dB] \tag{2.13}$$

- **Total Harmonic Distortion (THD)** - The THD, as the name implies, is a performance metric that measures the harmonic distortion of a signal and is defined as the ratio between the fundamental frequency power and the sum of the power from all the harmonic frequencies above the fundamental one [8].

- **Spurious Free Dynamic Range (SFDR)** - This performance metric is defined as the ratio of the signal power to the power of the strongest spectral tone [9]. This can be seen as the difference between the signal power and the maximum spurious power.

- **Signal-to-Noise-Distortion-Ratio (SNDR)** - The SNDR is the ratio between the input signal power and the power of all the distortion and noise elements affecting the output signal [1] [8].

$$SNDR = 10 \cdot \log\left[\frac{P_S}{P_D + P_N}\right] \qquad [dB] \tag{2.14}$$

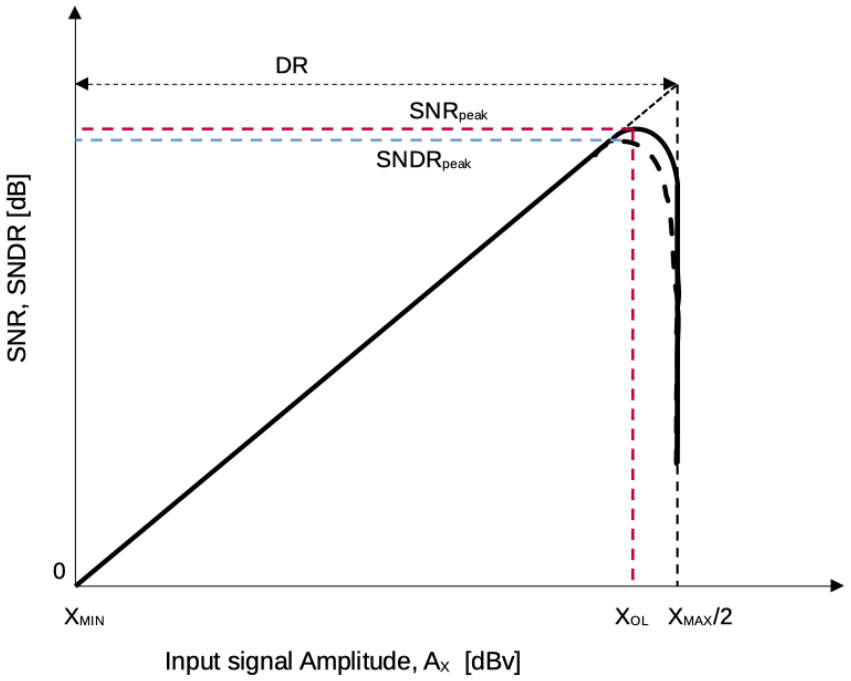


Figure 2.13: Typical Graphic of: SNR, SNDR and DR.

This Figure 2.13 presents a graphic that illustrates the typical results of SNDR, SNR and DR. The maximum values of SNR and SNDR are refered as SNR_{PEAK} and $SNDR_{PEAK}$, respectively. It is important to outline that the SNR grows with the input signal amplitude, A_X, until it gets to X_{OL}, the overload value. As expected, the quantizer often starts to overload beyond this level, and consequently the SNR curve drops suddenly. Due to the distortion, the $SNDR_{PEAK}$ value is lower than SNR_{PEAK} [1].

- **Effective Number of Bits (ENOB)** - The ENOB is the resolution of the A/D converter, in bits [8].

$$ENOB = \frac{SNDR_{peak} - 1.76}{6.02} \qquad [bit] \tag{2.15}$$

- **Figure of Merit (FoM)** - This performance metric is often taken into consideration because it is able to establish a comparison between different implementations of modulators [9]. Therefore, several FoM's can be described.

The Walden FoM (FoM_w) - this figure of merit has to consider different specification metrics like signal bandwidth, ENOB, and power consumption. In this case, the smaller the FoM_w, the better is the circuit's energy efficiency [1].

$$FoM_w = \frac{P_C}{2 \cdot B \cdot 2^{ENOB}} \qquad \left[\frac{J}{conv - step}\right] \tag{2.16}$$

The Schreier FoM (FoM_s) - this FoM is very well-known, and widely used to rank circuits. In this case, opposite to the FoM_w, the better circuits (i.e. the more efficient ones) achieve larger values for the FoM_s [1].

$$FoM_s = DR + 10 \cdot \log\left[\frac{B}{P_C}\right] \qquad [dB] \tag{2.17}$$

2.2 Sigma-Delta Modulation

The correct operation of ΣΔ Modulators relies on oversampling and noise shaping which are two techniques related to fundamental processes involved in an ADC [11].

Therefore, the basic principles of A/D conversion, previously analyzed in the last section, can be applied directly to a sigma-delta modulator. To this extent, an introduction to the basic definitions of ΣΔM is presented. Further, this section also presents important descriptions of different types of architectures and methods to improve the modulator's high accuracy and performance while also maintaining a low power consumption.

There are two main types of oversampling converters: the Δ modulators and, as previously mentioned, the ΣΔ modulators. Both contain a DAC, an ADC and a loop filter, represented as an integrator, as depicted in Figs. 2.14 and 2.15.

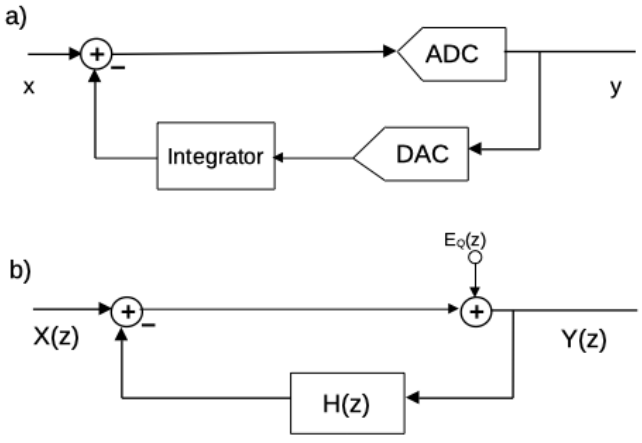


Figure 2.14: Diagram block of a Delta Modulator (a) and its linear Z-domain model (b)

A Δ modulator, as illustrated in Fig.2.14, has a loop filter in the feedback path, which is a great disadvantage because it amplifies the non-linearity DAC distortion. Thus, instead

of having it in a feedback path, the ΣΔ modulator has a loop filter in the forward path, as illustrated in Fig. 2.15, specifically to avoid this shortcoming [2].

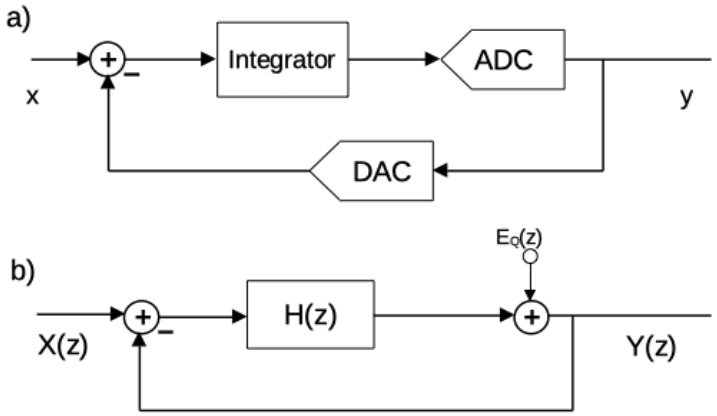


Figure 2.15: Diagram block of a first order Sigma-Delta Modulator (a) and its linear Z-domain model (b)

Figure 2.15 b) presents the ideal frequency domain model of the modulator, and for it to be analysed correctly, the Z-transform is necessary. An ideal transfer function of an integrator, in the Z-domain, is $H(z) = z^{-1}/1 - z^{-1}$. X and e_Q can be represent by their following Z-transforms *X(z)* and $E_Q(z)$, respectively. Hence, the output of the ΣΔM is given by the Equation 2.18 [1] [8].

$$Y(z) = z^{-1}X(z) + (1 - z^{-1})E_Q(z) \tag{2.18}$$

The Equation 2.18 demonstrates that the output of the modulator depends on the input signal and the quantization noise, nevertheless, it's preferable to have it depending on the Signal Transfer Function (STF) and the Noise Transfer Function (NTF). Thus, if this expression is multiplied by the input signal, the STF can be obtained, and if it is multiplied by the quantization noise, the same will happen for the NTF, as it can be seen by the Equations 2.19 and 2.20.

$$Y(z) = STF(z)X(z) + NTF(z)E_Q(z) \tag{2.19}$$

$$STF(z) = \frac{H(z)}{1 + H(z)} = z^{-1} \qquad NTF(z) = \frac{1}{1 + H(z)} = 1 - z^{-1} \tag{2.20}$$

It should be noticed that the zeros of the *NTF(z)* are equal to the poles of *H(z)*, and in general, if, *H(z)* has a high gain, in signal band, it results in STF(z) being close to one and to the *NTF(z)* being very small. Which shows that a high gain (in this conditions) has a greater impact in the quantization noise (that is strongly attenuated) in opposite to the input signal, that is largely unaffected by it. Ideally, for a DC signal, the quantization

noise would be completely cancelled [1]. Moreover, in the ideal case, the NTF(z), can be described as:

$$NTF(z) = (1 - z^{-1})^L \tag{2.21}$$

Where the filter order is represented by L, and (assuming that the quantization error can be modeled as an additive white noise source, that OSR=1 and $z = e^{j2\pi/F_S}$) P_Q can be redefined as:

$$P_Q \equiv \int_{-B}^{B} \frac{\Delta^2}{12 * F_S} |NTF(f)|^2, df \simeq \frac{\Delta^2}{12} * \frac{\pi^{2L}}{(2L+1) * OSR^{2L+1}} \tag{2.22}$$

It is very important to understand that by doubling the OSR, the P_Q decreases the additional $6L$ dB (and $L = 1$ because it is a first order modulator), which means that using a 1st order ΣΔM has a greater impact in the SNR than using just oversampling (P_Q only decreases 3 dB, as seen by the Equation 2.9) [1].

Overall, one can conclude that there are many advantages of using ΣΔ modulation and increasing it's oversampling ratio, such as: relaxed requirements of complexity of the analog circuitry, the required front-end anti-aliasing filter can be easier implemented and, compared to the case of Nyquist-rate converters, a lower quantizer's resolution can be used.

Although, in order to be able to have these improvements, the ΣΔMs require the incorporation of digital circuits for the output filter, a faster system performance for the same signal bandwidth and the noise filtering is essential because it allows for the quantization noise to occur at higher frequencies than the signal's bandwidth [1].

Therefore, in order to attenuate the noise power a digital low-pass filter must be added after the modulator. The signal is usually downsampled, after digital filtering, to the Nyquist rate without affecting the SNR. This operation is known as decimation [1]. The components required for these operations can be illustrated by the diagram block depicted in Figure 2.16.

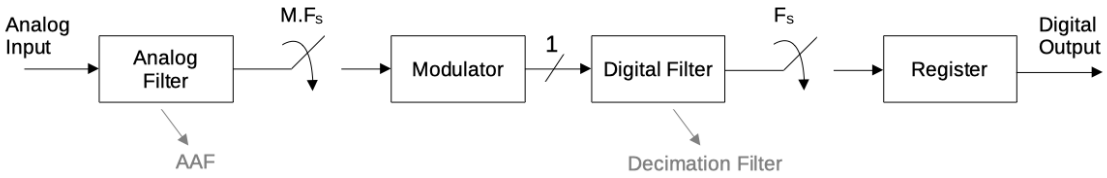


Figure 2.16: Diagram block of a general architecture of a ΣΔ ADC.

2.2.1 Basic Architectures of ΣΔ Modulators

As the general architecture of a ΣΔM ADC was shown, it is important to know that there can be different types of modulators available to fulfill different types of requirements

such as resolution, bandwidth, SNR and others. Therefore, it is important to classify ΣΔ architectures into different categories, according to different specifications, such as the type of loop filter that is implemented, the number of bits of the internal quantizer, or even the number of quantizers used in the complete ΣΔM [1].

2.2.1.1 Single-Loop Low-Pass ΣΔM

The first architecture to be presented is the previously analyzed first order ΣΔM, but now a discrete-time model is depicted, in Figure 2.17.

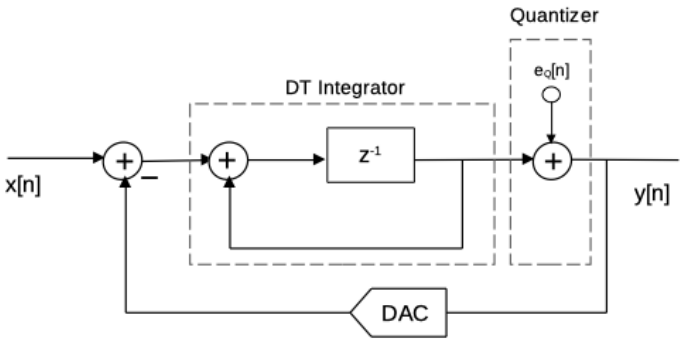


Figure 2.17: Example of a First-Order DTΣΔM.

Here the loop filter is a discrete-time integrator, whose z transfer function is the previously analized, *H(z)*, resulting in the *STF(z)* and *NTF(z)* from Equation 2.20. And the first-order ΣΔM shapes the quantization noise with 20 dB/decade slope, as illustrated in Fig.2.18.

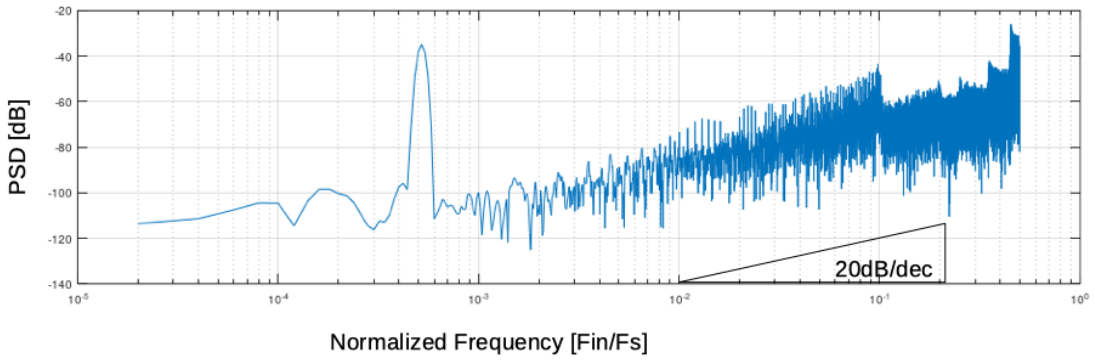


Figure 2.18: Example of the output PSD of a First-Order ΣΔM.

One can easily point out that the quantization error power contains strong tone distortion components and it is not, at all, uniformly distributed over the entire spectrum, which are the main disadvantages of this kind of modulator. This happens due to the fact that the quantization error can be strongly correlated with the input signal and therefore might not be considered white noise. It should also be noted that by doubling the OSR

of a first order ΣΔM it only improves 9 dB (1.5 bit), and, in fact, this is not very precise due to the previously mentioned correlation [1]. This allows to reach the conclusion that these modulators require a large OSR for accurate analog to digital data conversion. Furthermore, because of the significant tone structure present at the output of a first order ΣΔ A/D converter, this modulator is rarely used in audio applications.

On the other hand, by adding one integrator and an extra feedback path (as depicted in Fig. 2.19) to the first-order system, the problems described previously can be mostly avoided. This is why the standard second-order ΣΔM is often preferred and widely more used [1].

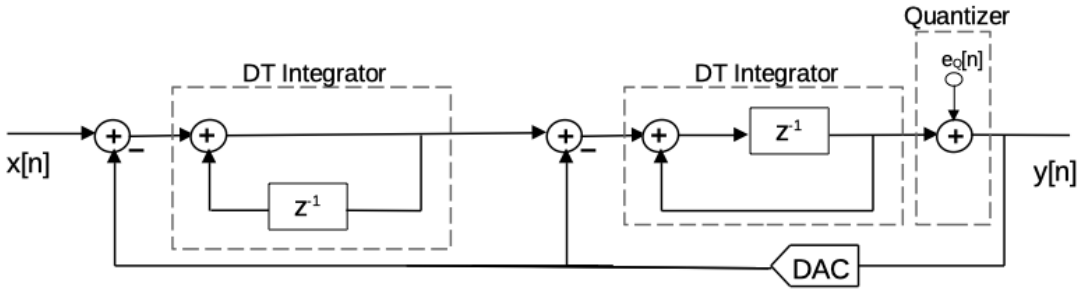


Figure 2.19: Example of a Second-Order DTΣΔM.

In this case, the second-order ΣΔM has a 40dB/decade slope of the quantization noise and the signal and noise transfer functions can be seen by the Equations 2.23.

$$STF(z) = \frac{H(z)}{1+H(z)} = z^{-1} \qquad NTF(z) = \frac{1}{1+H(z)} = (1-z^{-1})^2 \tag{2.23}$$

These changes, allow to decorrelate the quantization error from the input signal (which means less idle tones in output spectrum), to amplify the noise outside the signal bandwidth in a more efficient way and even provide additional suppression of the in-band quantization noise. Moreover, each time the OSR is doubled there is an improvement of 15 dB (2.5 bits) of the SNR of a second-order ΣΔM [1].

In theory, by adding more feedback branches and integrators to the loop, even higher-order NTFs can be obtained [2]. For an L-order loop filter resulting in $NTF(z) = (1-z^{-1})^L$, the quantization noise is shaped with $20 * L\ dB/decade$. And the SNR can improve $6L + 3dB$ $(L + 0.5bit)$ by doubling the OSR [1].

Nevertheless, higher order ΣΔMs have stability issues as these structures use feedback loops, so it is very complicated to achieve the theoretical performance, in reality, due to these limitations. The main struggle is that high order ΣΔMs are only conditionally stable as a single bit quantizer is used. The stability can rely on the limited range input signal amplitudes or precise circuit coefficient matching [1].

To overcome these issues of higher order ΣΔMs, MASH structures can be used.

2.2.1.2 Continuous-Time and Discrete-Time ΣΔM

The presented low-pass single-loop architectures are discrete-time ΣΔM and to better understand what this truly means a DT ΣΔ modulator is illustrated in Fig. 2.20 a). As previously analyzed and seen in Fig.2.8 and Fig.2.16 it is necessary to have an anti-aliasing filter before the modulator. Then the filtered input goes through the S/H block to be sampled and from there the signal is ready for the following part of the completely DT modulator, as illustrated in Fig.2.20 a) [1].

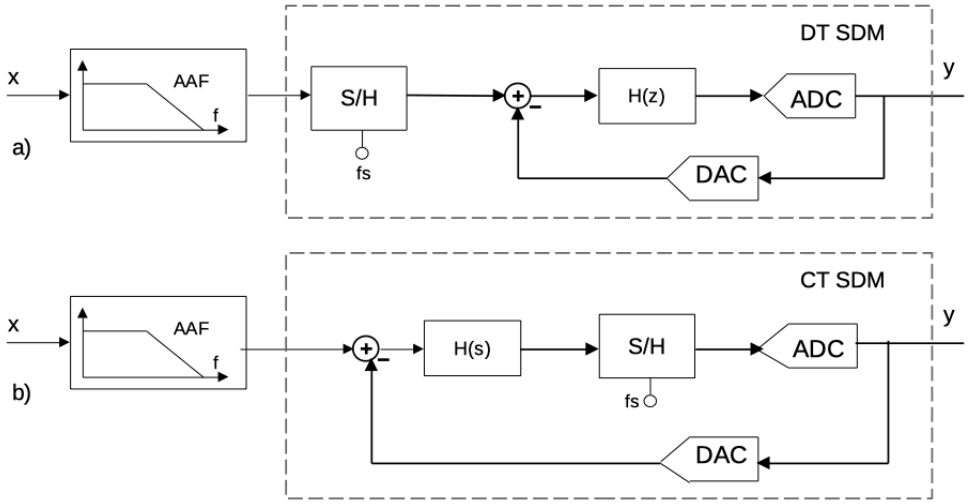


Figure 2.20: Example of (a) a Discrete-Time and (b) a Continuous-Time modulators.

However, this is not the case in Fig.2.20 b). Here it is important to notice that a continuous-time integrator is being used, while the rest of the circuit operates in DT, which allows for the sampling operation, performed by the S/H block, to happen at the output of the loop filter. Thus, the output signal of CTΣΔM is still discrete-time [1].

It is also worth mentioning that the S/H flaws and the wideband noise folding, in this case, do not influence the circuit as much as in the DT ΣΔMs [1]. Moreover, the CT modulators possess inherent anti-aliasing, which means that the AAF, usually, is not required, because the degree of alias suppression is equivalent to the degree of quantization-noise suppression [1]. Further, these modulators can operate with a clock frequency (and thus achieve a signal bandwidth) which is 2-4 times greater than what Switched-Capacitor (SC) discrete-time techniques can achieve [2].

Although, despite these advantages, it is important to know that the CT signals need to be processed with high linearity, and that this system is very sensitive to the effects of DAC memory and to the clock jitter noise. Another important issue is the possibility of a delay in the CT ΣΔM loop filter response, that is related with the time delay in the response of the DAC and quantizer. This is called excess-loop-delay and it is likely that it's equivalent relative error signal would be correlated with the input signal, so it could manifest itself as a distortion product. Usually this delay can introduce additional poles

and is a fraction of the sampling period, which, leads to the increase of the STF and NTF orders [1] [2].

2.2.1.3 Multi-bit and Single-bit ΣΔM

The distinction between multi-bit and single-bit modulators architectures, depends on the number of bits required for the quantizer, in the ΣΔM, to perform properly. When the quantizer offers $2^N - 1$ levels of quantization usually it can be designated with 0.5 additional bits. The quantization characteristic of single-bit and multi-bit quantizers is depicted in Fig. 2.21 a) and b) respectively.

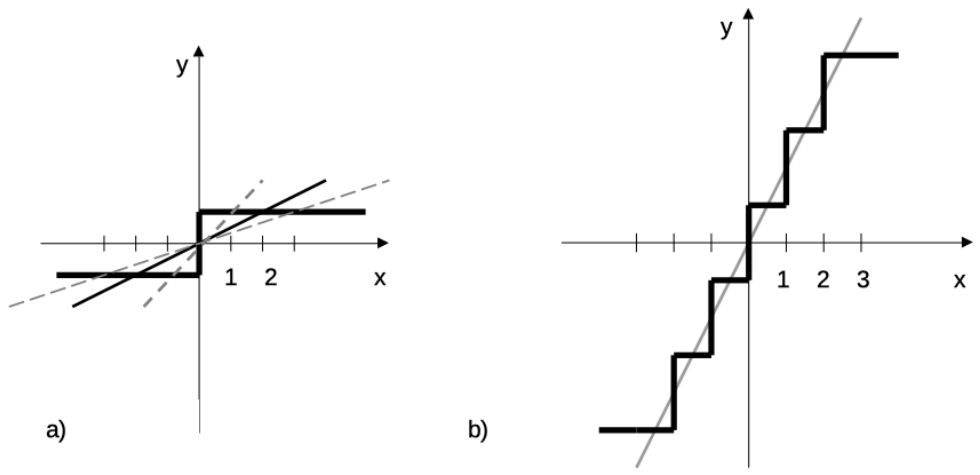


Figure 2.21: Quantization staircase characteristics of (a) a single-bit and (b) a multi-bit quantizers.

In the feedback loop of the modulator, the DAC has the same number of bits as the ADC. And, as seen in previous figures, usually the the input of the ΣΔM receives the output signal of the internal DAC because it is directly fed back into it, so it's very likely that the DAC's non-linearity can strongly influence the ΣΔM's linearity. Hence, a very common solution is to use a 1-bit DAC which is inherently linear and a single-bit quantizer. By resorting to a single-bit ΣΔM it is important to have a large enough OSR, in order to achieve a high resolution [1].

Further, as the single-bit quantizer's gain depends on it's input signal amplitude, it cannot be easily determined. When the the gain increases, the input signal amplitude decreases and the same happens the other way around [1].

On the other hand the multi-bit architecture has the benefit of each additional bit used in the quantizer will yield a 6 dB improvement in the SNR, and is also able to make higher order modulators more stable, unfortunately this can only be achieved at a cost of increased circuit complexity and the SNDR improvement is limited due to mismatch errors in the feedback DACs [1] [12].

2.2.1.4 Multistage Noise Shaping (MASH) ΣΔM

All the different types of modulators, previously described, were single-loop architectures, as all required only one quantizer in each complete modulator presented. Therefore, a new alternative, is the cascaded modulator architecture also known as MASH (or multi-loop) ΣΔM.

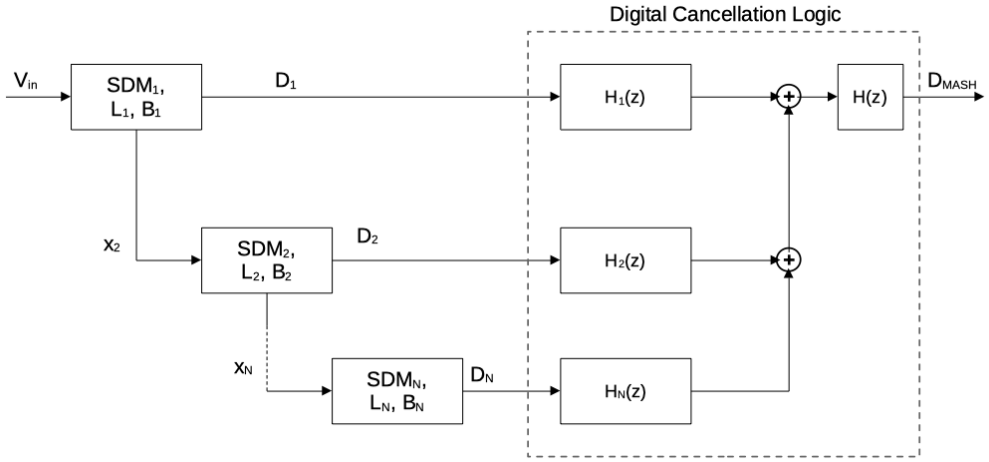


Figure 2.22: A general MASH ΣΔM structure.

This architecture (Fig.2.22) uses a combination of first and second order modulators in order to obtain a higher order noise transfer function.The principle behind this idea is simple, by using a second modulator to modulate the quantization error of the first modulator, one can combine the two outputs to cancel the quantization error of the first modulator. This approach can be implemented recursively and use more than just two modulators in cascade. Thus the resulting modulator is inherently stable because it uses only first and second order modulators that are inherently stable.

In single-loop ΣΔM structures, the coefficients of the NTF and of the STF can suffer alterations due to the component's mismatch errors, however, this changes will not significantly affect the SNR performance of the modulator. This is because of the large OSR, even if the loop transfer function changes due to component variations, the quantization error suppression inside the signal band still remains large. Thus, these single-loop architectures can be, to some extent, nonreactive to the mismatch of analog components. However, MASH structures, on the contrary, demand a very high matching precision between the digital (digital cancellation logic) and the analog (modulator's circuit) transfer functions. Ideally, with an appropriate recombination of all digital outputs, the quantization noise of all the stages can be canceled, except from the last one. Although, in reality, the accuracy of the cancellation can decrease due to the mismatch between the values in the DCL and the values in the analog modulator's circuit, which, as a consequence, culminates in the introduction of poorly shaped noise from previous stages in the output spectrum (it is important to point out that the CT ΣΔM are more susceptible to the imperfect matching of the modulator's components) [1] [9]. Thus, the highest number of stages,

of a given MASH modulator, is limited by the circuit non-linearities. When a certain limit is reached, there is no need to try and add more stages because no improvement in the performance can be achieved [1].

2.2.2 Low Power ΣΔ Modulators

In order to improve the performance and optimize the low power dissipation of ΣΔMs, there are a few techniques and improvements that can be applied to the previously presented architectures.

2.2.2.1 Multi Feedback and Feed-Forward ΣΔM

Starting with the example of the second-order ΣΔM, a number of alternative implementations exist which still allow to perform a second-order modulation, and also give a unity-gain STF, as well as the same $NTF(z) = (1 - z^{-1})^2$ other than the one previously demonstrated [2]. As so, a general structure of a second-order modulator, with arbitrary feedback and feed-forward coefficients is presented in Fig.2.23.

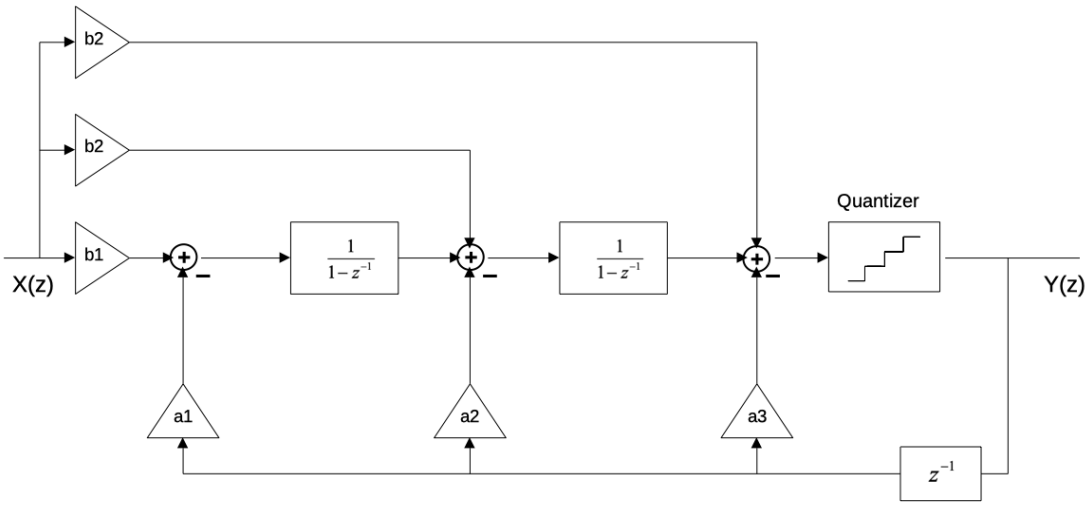


Figure 2.23: A second-order ΣΔM with arbitrary feedback and feed-forward coefficients.

By adding both multiple feed-forward and feedback features into the second-order structure, it allows to enhance stability and improve the Dynamic Range which in turn provides more flexibility [2]. The same happens to higher order ΣΔMs when applying this concept. Although the matter of stability is notably important for orders $L > 2$ of ΣΔM [1]. The presented architectures in [13] are a great example of how the use of feedback and feed-forward structures can improve the modulator's performance and allow for higher noise shaping order than a simpler architecture.

2.2.2.2 Passive ΣΔM

An important way to distinguish different types of ΣΔ Modulators is according to the distribution of the loop gain.

In an active ΣΔM, the loop gain is located in the integrators, while in a passive ΣΔM, the gain is mainly concentrated in the quantizer. Hence, the signal's amplitude is significantly reduced at the input of the quantizer, due to the fact that passive integrators can only attenuate. Thus, is very common to resort to a single-bit quantizer (comparator) in passive modulators, instead of a multi-bit quantizer [1].

Passive ΣΔMs can be implemented in Discrete-Time or Continuous-Time by using low pass filter instead of an active comparator. An example of a CT Passive ΣΔ Modulator is given in [14] where the integrators are implemented with first order resistor-capacitor (RC) circuits with a pole and a pole and a zero, as shown in Fig. 2.24 a) and b), respectively [8].

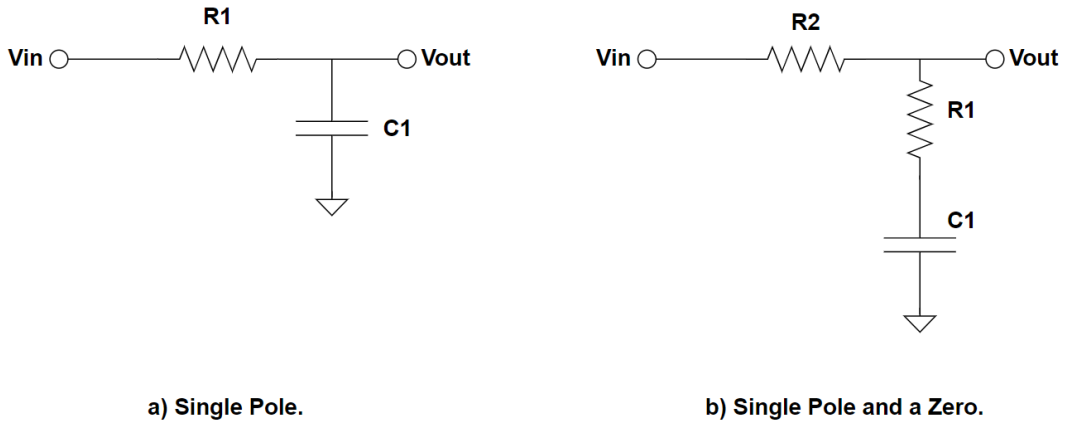


Figure 2.24: CT Passive RC integrators.

In this case, by adding a zero with the second CT passive RC integrator to the loop gain it is possible to have a stable closed loop system with only one feedback path. The CT passive RC integrators have the following transfer functions described in Equation 2.24 .

$$H(s) = \frac{1}{1 + sR_1C_1} \qquad H(s) = \frac{1 + sR_1C_1}{1 + s(R_1 + R_2)C_1} \tag{2.24}$$

DT Passive integrators can be implemented with SC circuits, as illustrated in Fig. 2.25, which, typically have the transfer function described in Equation 2.25.

$$H(z) = \frac{\frac{C_1}{C_1+C_2} * z^{-1}}{1 - \frac{C_1}{C_1+C_2} * z^{-1}} \tag{2.25}$$

Moreover, in [15], an example can be seen where the effectiveness of the DT passive filters in ΣΔ modulators is studied, with focus on very low-speed and low-power applications. The achieved 430 nW and 0.296 pJ/step passive ADC has a very low power consumption level and FoM [15].

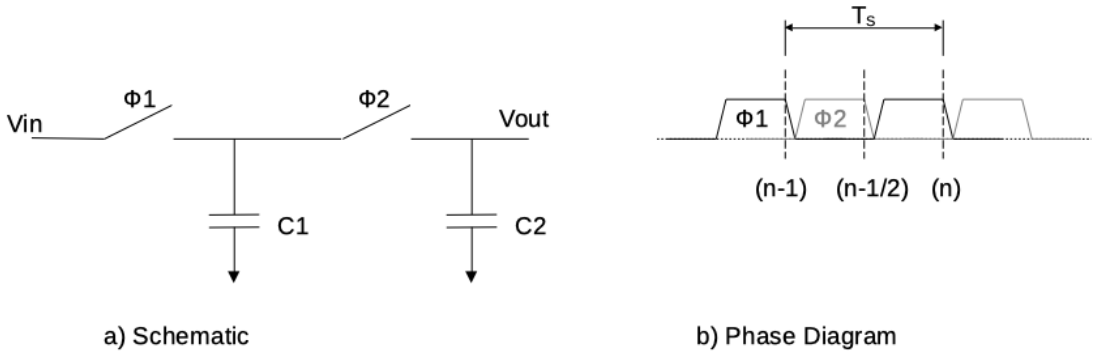


Figure 2.25: DT Passive SC integrator.

2.2.2.3 Ultra Incomplete Settling (UIS)

The best way to describe the concept of Ultra Incomplete Settling (UIS) is as a circumstance where the capacitor of an RC circuit is only charged to a small degree (Figs.2.26 and 2.27) [8].

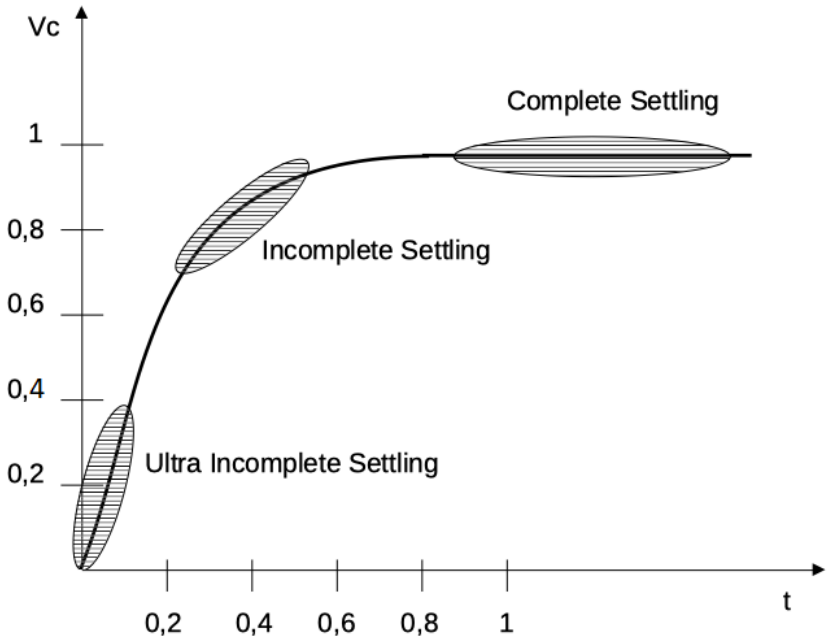


Figure 2.26: Capacitor voltage for a step input (normalized time).

As presented in [1], after a step input with amplitude *Vin*, the capacitor's voltage in an RC circuit is given by the Equation 2.26.

$$V_c(t) = V_{in} \cdot (1 - e^{\frac{-t}{R_{on}*C}}) + V_{C0} \cdot e^{\frac{-t}{R_{on}*C}} \tag{2.26}$$

Where *Vc0* is the initial voltage in the capacitor, just before the input step. Fig. 2.26 illustrates a plot of Equation 2.26 as a function of time, for generic values of C and Ron.

In a switched-capacitor circuit operating normally, there are two most common scenarios that can be expected: either the capacitor is (almost) completely charged (complete settling area in the graphic of Fig.2.26) or discharged at the end of the clock phase. However, if the clock's phase period is much smaller than the time constant of the circuit, then the RC circuit operates under the ultra-incomplete settling (UIS) mode, $T_S << RC$. This can easily happen by adding a resistance, R, with an explicit calculated value in series with the switch from the first phase, just as illustrated in Fig. 2.27 [1].

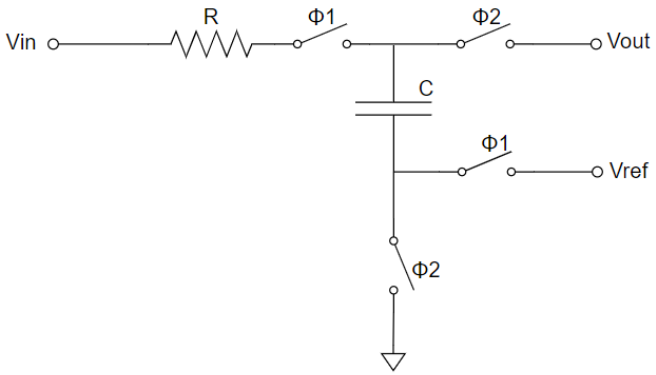


Figure 2.27: Schematic of an RC passive integrator in UIS region.

Although it needs to be outlined that the presence of parasitic capacitances and finite on-resistances may affect the performance of the ΣΔM and it is extremely important to consider these effects when designing a modulator to perform at higher clock frequencies. Hence one can resort to the implementation of the single-ended branch of the passive RC integrator, illustrated in Fig. 2.27 which allows to connect the parasitic capacitances in parallel with C (that is much bigger, and as so, the parasitic capacitances can be neglected). It should also be pointed out that the RON of the first switch of phase $\phi 1$ limits the transfer speed, and thus, the value of this resistance should be kept small [1].

2.2.3 Literature Review

In order to compliment the research done in this section, it is considered important to present significant works published with different and relevant examples of the architectures previously described. Therefore Tables 2.1 and 2.2 present a summary of the performance parameters of the single-loop DT and CT ΣΔM. Table 2.3 presents the cascade ΣΔMs also distinguished between DT and CT structures. It should be known that the main criteria for selecting these ΣΔMs is their low FoM_W presented amid recently published authors. These tables are based on the ones presented in Chapter 3 of [1] with some additions from the most recent relevant works published in the last six years.

Table 2.1: Examples of single-loop low-pass DT ΣΔMs.

Single-bit Quantizer									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[16]	0.18	0.715	81,0	85.0	0.036	4	0.02	98.1	172.4
[17]	0.18	0.44	78.2	79.3	0.036	4	0.02	98.1	164.8
[18]	0.13	0.602	75.0	88,0	7.4	64	0.5	1610	166.3
[19]	0.13	0.338	61.0	64.0	0.0075	1.4	0.01	409.0	155.2
[20]	0.045	0.04	56.3	59.6	9	1500	15	562.1	151.8
Multi-bit Quantizer									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[21]	0.18	0.492	84.0	88.0	0.14	3.2	0.1	54.0	176.5
[22]	0.18	2.32	78.0	86.0	5.1	144	2.2	178.5	172.3
[23]	0.09	0.26	63.0	66.0	1.2	38.4	1.94	267.9	158.1
[24]	0.18	3.67	81,0	83.0	15	60	2.5	327.1	165.2
[25]	0.18	2.16	95.0	100.0	0.87	5	0.025	378.5	174.6

Table 2.2: Examples of single-loop low-pass CT ΣΔMs.

Single-bit Quantizer									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[26]	0.065	0.013	69.1	76.2	0.256	320	2	27.5	175.1
[27]	0.016	0.0194	66.9	67.8	5.25	2400	40	36.3	166.6
[27]	0.016	0.0194	67.7	68.2	4.3	1800	30	36.1	166.6
[27]	0.016	0.0194	69.4	70	3.24	1200	20	33.6	167.9
[28]	0.065	0.039	68.6	71.2	1820	650	10	41.4	138.6
Multi-bit Quantizer									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[29]	0.13	0.17	80.4	82.9	11.4	64	15	44.4	174.1
[30]	0.028	0.25	79.8	82.2	64.3	20	50	80.5	168.7
[31]	0.04	0.051	70.0	70.6	2.57	300	10	50	165.9
[32]	0.028	0.08	73.6	78.1	3.9	640	18	27.7	174.7
[33]	0.18	0.72	90.8	93.5	0.09	3.072	0.024	66.2	177.8

Table 2.3: Examples of cascade low-pass ΣΔMs.

Discrete Time Implementation									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[34]	0.09	1	70.0	72.0	27.9	420	20	269.9	160.6
[35]	0.09	0.076	65.0	66.0	6.83	320	2	1175	150.7
[36]	0.032	0.13	63.0	66.0	28	400	20	606.4	154.5
[37]	0.13	0.2	64.0	70.0	4.3	38.4	1.92	864.6	156.5
[38]	0.065	0.28	67.0	67.0	46	240	15	838.1	152.1
Continuous Time Implementation									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[39]	0.065	0.17	61.0	70.0	10.5	208	10	572.6	159.8
[39]	0.065	0.17	55.0	61.0	10.5	208	14.9	769.1	152.5
[40]	0.028	0.34	74.6	85.0	78	1800	50	177.7	173.1
[41]	0.028	0.9	72.6	90.0	235	3200	45	749.0	172.8
[41]	0.028	0.9	71.4	88.0	235	3200	53.3	726.0	171.6

2.3 Calibration Techniques

Finally, after reviewing the main theoretical aspects of ΣΔ Modulators, it is important to keep in mind that the purpose of this project is to study the implementation of a calibration circuit for the CT 2-1 MASH ΣΔM, based on [1]. This circuit must be able to measure the value of the relevant analog circuit's parameters and convert these values into a digital coefficient that can be used in the circuit's digital cancellation logic. Therefore, this section focuses on the analysis of different calibration techniques that can be applied to MASH ΣΔ modulators in order to improve their performance. Moreover, a comparison of the published calibration circuits with similar characteristics to the requirements of this work, will be presented in Table. 2.4.

As previously mentioned, the operation of ΣΔ Modulators relies on the combination of oversampling and noise shapping, which make a better trade-off between the sampling speed and signal-to-noise ratio compared to the Nyquist-rate ADC. And to achieve higher resolution, cascaded structures (MASH) offer a robust stable alternative instead of using high-order single-loop ΣΔ typologies [42].

Nevertheless, it is important to remember that these structures are sensitive to analog circuit imperfection due to the requirements of matching between the analog and the digital transfer functions, in order to avoid the leakage of prior stage's quantization noise. Furthermore, designing wide-band high resolution ΣΔ Modulators can be a very challenging task, even when taking advantage of hybrid active-passive or passive structures in order to achieve power efficiency, the ΣΔM can still have it's performance limited by process variations of the values of it's constituted components.

To overcome this problem, calibration techniques are usually applied for the compensation, although in MASH ΣΔMs there isn't a general approach to compensate for the circuit nonidealities as there are so many architectures with different characteristics, and,

therefore, different ways to calibrate in order to achieve the desired performance.

Most of the known works published in reference to calibration techniques are focused on the multibit DAC linearization, as this is one of the most common problems in Multi-bit ΣΔ architectures. Usually some kind of linearization scheme is needed if linearity beyond the intrinsic device matching is required . An example can be seen in [43] where the calibration technique allows to correct the DAC errors by storing a digital estimation of the DAC errors in a lookup table (LUT), which is used during the normal conversion [43].

Some other works, address the problem of limited DC gain and recur to the solution of adding analog elements, such as a tunable capacitor [44] or an extra unity gain buffer [45], to the conventional integrator. Although the accuracy of these analog calibration methods is confined to the accuracy of the added analog element and, thus, might not be able to omit the limited gain effects entirely [42].

It is also important to know that some circuit imperfections, including the limited DC gain, in MASH structures can lead to quantization noise leakage. In order to decrease the noise leakage, a common calibration technique that can be used is to change the parameters of digital filters to match the imperfections of the analog circuit in order to correct them. As this calibration technique is performed in the digital domain, it has the advantage of the added area and power consumption being lower than that of the analog calibration techniques. For this reason this approach will be used and further analysed in this thesis.

There are many different types of calibration methods that can be applied in the digital domain. In [46], through adaptive off-line or on-line calibration, the coefficients needed for the digital correction are determined by linear finite-impulse response (FIR) filters. This paper is divided into two parts, whereas in the first part an on-line "blind" calibration technique is applied, that doesn't use any reference and operates directly on the digital output, during conversion, with the only requirement, on the unknown input signal, is that its spectrum be bandlimited [46].

On the other hand, the second part of this paper [47] describes a different approach. The quantization noise is added at the input of the first-stage quantizer, where there is an injection of a pseudo-random two-level test signal. Then the test signal leaks into the output signal, where it can be detected and used to control the digital noise-cancellation filter [47].

In [48], an adaptive fifth-order polynominal digital filter is applied to a MASH 2-2 ΣΔM to remove the first stage settling errors. The main disadvantage of this technique is not operating in the background and mainly ignoring the second stage settling errors. However, for example in [42], an accumulative error model is proposed to study the effects of both limited DC gain and GBW in DT integrators, and a digital calibration technique is used to compensate for the errors induced by these nonidealities in a MASH 2-2 ΣΔM. As so, this calibration scheme operates in the background and corrects both stages of the MASH modulator [42].

A particular unique approach in digital calibration of MASH structures that can be used in some cases is described in [49] where the authors elaborate a new filter design technique applied to a model-matching problem with polytopic uncertainties in parameters, that is based on linear matrix inequalities. This technique can lead to an optimal, less conservative, solution to the robust integrator transfer functions filtering problem [49].

It should be mentioned that in this case in particular, the calibration in the digital domain will be towards the components of the passive CT RC integrators, based on [1]. Some interesting examples of RC time constant calibration circuits can be seen in [50] and in [51].

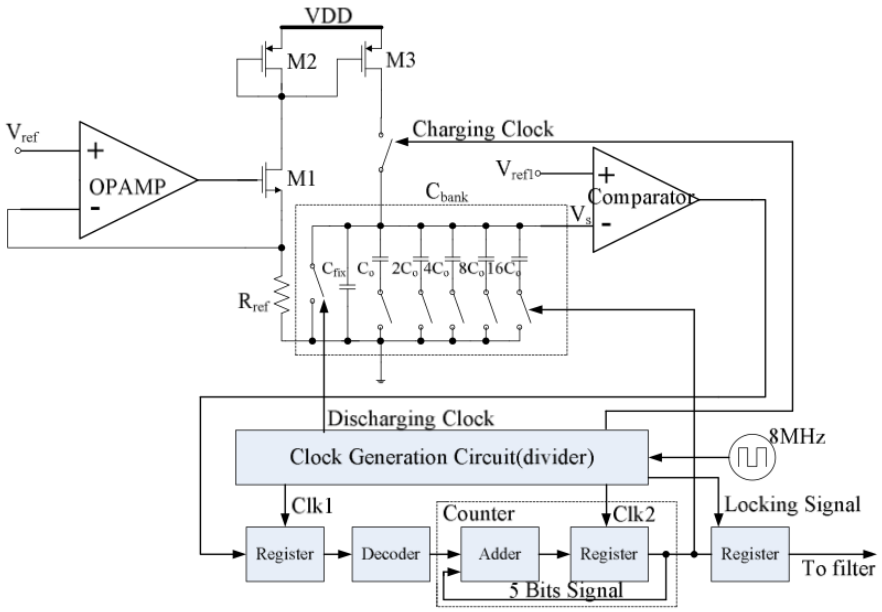


Figure 2.28: RC time constant auto-calibrating circuit from article [50].

In [50], a combination of an active RC structure and a gm-C structure, also referred to as an active-gm-RC structure, is adopted with the intend to design a Low Pass Filter (LPF), due to the better linearity and less current consumption that it provides. Nonetheless, the filter coefficients of the active-gm-RC structure vary due to the process variation. In order to solve this, a frequency auto-tuning circuit was added to the structure in order to calibrate the frequency variation.

Therefore, Fig. 2.28 presents the RC time constant auto-calibrating circuit, from [50]. This circuit is in charge of detecting the variation of it's RC time constant in order to control it in the LPF core. Therefore, in order to detect effectively the necessary values and do the calibration, the resistance and capacitance of resistor, Rref, and capacitor array, Cbank, must be identical with the ones in the filter core. If there is deviation between the pre-settled value and the RC time constant in the calibration circuit, then the same is happening to the RC time constant of the filter core. The calibration is then performed

by changing the control codes of the respective capacitor arrays in order to achieve the desired capacitance for the RC time constant of the filter to approach it's ideal value [50].

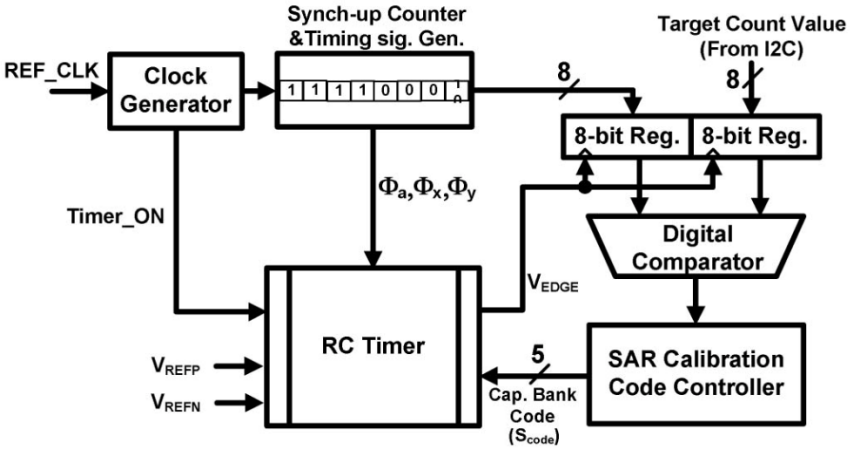


Figure 2.29: Block diagram of the proposed RC calibrator from article [51].

In [51] a very different RC calibrator is presented, as shown in Fig. 2.29, and it has the following components: an RC timer circuit, a clock generator, a synchronous counter, a digital comparator, and a SAR calibration code controller block. In this case, the calibration is performed by the RC timer, by adjusting the crossing time when two differential signals VDO1 and VDO2, which are integrated with reference signals VREFP and VREFN, cross each other to be the same as the target time point. This is possible due to use of the capacitor bank which changes the capacitance according to the RC time constant calibration code, Scode, provided from the SAR calibration code controller [51].

However, this project differs from the presented examples as it will consist in calculating the variation of the analog components, due to it's process deviations, from it's optimal point and adjust this variation with a digital coefficient that can be used in the digital cancellation logic. The following table (Tab. 2.4) presents published MASH calibration circuits (in the digital domain) with similar characteristics to the requirements of this work.

Table 2.4: Published calibration circuits (in the digital domain) with similar characteristics to the requirements of this work.

Digital RC Calibration Circuits									
Ref	**Tech** μ **m**	**Area** **mm2**	**SNDR** **dB**	**DR** **dB**	P_C **mW**	F_S **MHz**	**B** **MHz**	FoM_W **fJ/conv-step**	FoM_S **dB**
[52]	0.04	0.265	75.8	76.8	43	1000	50.3	-	165.1
[53]	0.04	0.05	90	102	0.4	6.5	0.024	-	179
[48]	0.25	0.39	75	77	5	62	1	0.56	-
[54]	0.18	1.7	63	67	68	160	10	-	-

3

Analysis of High-Level Simulations

Firstly, an overview of the implementation of the 2-1 MASH ΣΔM, based of [1], is presented, for general context and to better understand the approach used on the implementation of the calibration. This is followed by a study of the theoretical variation of the calibrator, and, a presentation of the proposed RC time constant calibration architecture and it's correspondent high-level model, implemented in *MATLAB* ®, with it's respective results.

3.1 Continuous-Time Passive Integrator with SC Feedback

This section presents an analysis of the integrators used in the implementation ΣΔM, based of [1], as it is fundamental to the better understanding of the full modulator and how it needs to be calibrated.

In order to take advantage of the UIS concept without the downside of applying bootstrapped switches in the output and input signal paths, as seen in Fig. 2.27, a new passive RC integrator is considered, in the 4th chapter of [1] (Fig. 3.1). This architecture allows to avoid the possible increase of the power dissipation that happens during the turning off and on of the switches, and ease the control of the non-ideal effects (due to the parasitic capacitances).

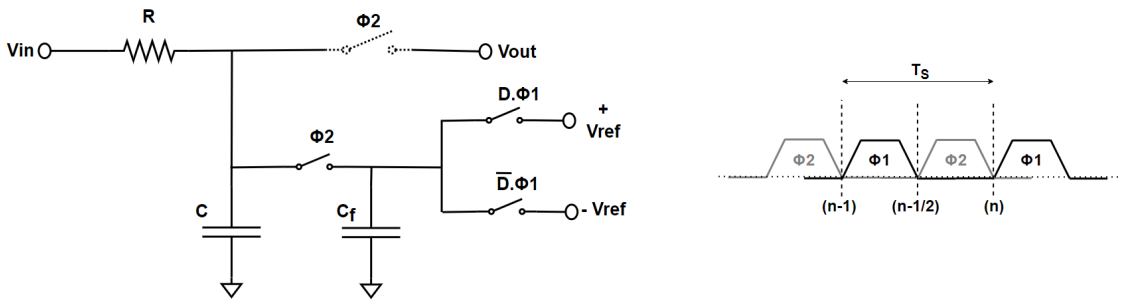


Figure 3.1: Schematic of a single-ended RC integrator (proposed in [1]) and corresponding phase diagram.

This integrator is composed by a resistor (R), a capacitor (C) and the feedback capacitor (C_f). The signal D represents the $\Sigma\Delta$M's output bit-stream voltage and the output switch defined by the grey dashed line, in Fig. 3.1, denotes the sampling operation of the quantizer.

Although this is a continuous-time integrator, it's output signal can be determined by the DT instants controlled by the sampling clock signal, seen in Fig. 3.1, which in turn results in the DT transfer function of this circuit. Thus, the charge conservation principles result in the following equations (neglecting the RON of the swithces).

$$Q_C^{(\phi_2)} + Q_{C_f}^{(\phi_2)} = Q_C^{(\phi_1)} + Q_{C_f}^{(\phi_1)} \tag{3.1}$$

$$V_C^*(n \cdot T_s) \cdot (C + C_f) = V_C[(n - \frac{1}{2}) \cdot T_S] \cdot C + D[(n-1) \cdot T_S] \cdot V_{ref} \cdot C_f \tag{3.2}$$

$$V_C^*(n \cdot T_s) = V_C[(n - \frac{1}{2}) \cdot T_S] \cdot \frac{C}{(C + C_f)} + D[(n-1) \cdot T_S] \cdot V_{ref} \cdot \frac{C_f}{(C + C_f)} \tag{3.3}$$

In the first equation Q_{C,C_f} represent the charge in C_f and C at phases $\phi 1$ and $\phi 2$, and in the following equations V_C and V_C^* are, respectively, the voltages sampled at the capacitor C before and after applying the feedback.

Firstly by analysing the schematic of the RC integrator during phase $\phi 1$ one can conclude that the output voltage is given by:

$$V_c(t) = V_{in} \cdot (1 - e^{\frac{-t}{\tau}}) + V_{C0} \cdot e^{\frac{-t}{\tau}} \tag{3.4}$$

Here V_{C0} is the voltage (at the beginning of phase $\phi 1$) across the capacitor C , and t is $T_S/2$ due to the consideration of one phase duration being $\tau = R \cdot C$. The output voltage of the circuit, at the end of phase $\phi 1$ ($(n - 1/2) \cdot T_S$) is given by:

$$V_C[(n - \frac{1}{2}) \cdot T_S] = V_{in} \cdot (1 - e^{\frac{-T_S}{2 \cdot R \cdot C}}) + V_C[(n-1) \cdot T_S] \cdot e^{\frac{-T_S}{2 \cdot R \cdot C}} \tag{3.5}$$

And the output voltage of the circuit, at the end of phase $\phi 2$ ($n \cdot T_S$) is given by:

$$V_C[n \cdot T_S] = V_{in} \cdot (1 - e^{\frac{-T_S}{2 \cdot R \cdot C_{eq}}}) + V_C^*[(n) \cdot T_S] \cdot e^{\frac{-T_S}{2 \cdot R \cdot C_{eq}}} \tag{3.6}$$

Here $C_{eq} = C + C_f$, and the voltage in the capacitor C after applying the feedback, V_C^*, can be attained by the previously seen Equation 3.3, producing:

$$V_C[n \cdot T_S] = V_{in} \cdot (1 - e^{\frac{-T_S}{2 \cdot R \cdot C_{eq}}}) + \left[V_C[(n - \frac{1}{2}) \cdot T_S] \cdot \frac{C}{C_{eq}} + D[(n-1) \cdot T_S] \cdot V_{ref} \cdot \frac{C_f}{C_{eq}} \right] \cdot e^{\frac{-T_S}{2 \cdot R \cdot C_{eq}}} \tag{3.7}$$

Therefore, by combining the expression obtain in this Equation, with the one from Equation 3.5, and assuming that the exponential terms can be approximated in accord with the following assumption:

$$if: \; x << 1 \rightarrow e^x = 1 + x \tag{3.8}$$

Then, the value of the output voltage of the integrator at the end of phase $\phi 2$ is given by:

$$V_C(n \cdot T_S) \approx V_{in} \cdot \frac{T_S}{2 \cdot R \cdot C_{eq}} + V_C[(n-1) \cdot T_S] \cdot \frac{C}{C_{eq}} \cdot \left(1 - \frac{T_S}{2 \cdot R \cdot C_{eq}} - \frac{T_S}{2 \cdot R \cdot C}\right) + \\ + D[(n-1) \cdot T_S] \cdot V_{ref} \cdot \frac{C_f}{C_{eq}} \cdot (1 - \frac{T_S}{2 \cdot R \cdot C_{eq}}) \tag{3.9}$$

This equation can also be described as:

$$V_c(n \cdot t) = V_{in} \cdot \alpha + V_C[(n-1) \cdot T_S] \cdot \beta + D[(n-1) \cdot T_S] \cdot \gamma \tag{3.10}$$

Where the coefficients α, β and γ denote adequate factors from the Equation 3.9. For analyses purpose, β can be simplified by considering that $C \approx C_{eq} = C + C_f$ (due to the fact that $C_f << C$), as seen in [1].

$$\alpha = \frac{T_S}{R \cdot (C + C_f)} \tag{3.11}$$

$$\beta = \frac{C}{C_{eq}} \cdot \left(1 - \frac{T_S}{2 \cdot R \cdot C_{eq}} - \frac{T_S}{2 \cdot R \cdot C}\right) \approx \frac{C}{C_{eq}} \cdot (1 - \alpha) \approx (1 - \alpha) \tag{3.12}$$

In order to improve the circuit's performance, the integrator was implemented as a differential circuit, as depicted in the Fig. 3.2.

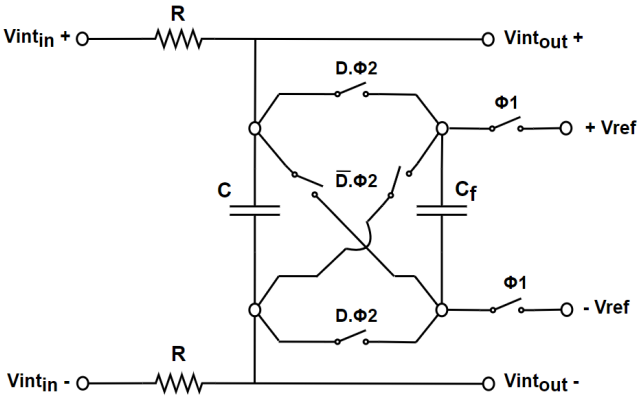


Figure 3.2: Schematic of the differential RC integrator, proposed in [1].

As illustrated by Fig. 3.2, this integrator now incorporates two resistors, therefore, the value of α changes according to this modification (and β is still described depending on alpha).

$$\alpha = \frac{T_S}{2 \cdot R \cdot (C + C_f)} \tag{3.13}$$

$$\beta \approx \frac{C}{C_{eq}} \cdot (1 - \alpha) \approx (1 - \alpha) \tag{3.14}$$

And the integrator's transfer function can be obtained by using the Z transform, and assuming the reference input voltage is equal to zero, resulting in:

$$H(z) = \frac{V_C(z)}{V_{in}(z)} = \frac{\alpha}{1 - \beta \cdot z^{-1}} \tag{3.15}$$

The advantage of implementing this circuit, is that under the right conditions, it can be understood as non-delaying passive DT 1st order integrator. The minimum and maximum gain values of the transfer function are, respectively:

$$H_{min} = |H(z = -1)| = \left|\frac{\alpha}{1 + \beta}\right| \approx \frac{\alpha}{2 - \alpha} \tag{3.16}$$

$$H_{max} = |H(z = 1)| = \left|\frac{\alpha}{1 - \beta}\right| = 1 \tag{3.17}$$

Thus, it is essential to have a value of β as close to one as possible so that the integrator's frequency response can be closer to the ideal one. This can be achieved by resorting to the use of very small values of α which, in turn, results in a highly limited amplitude for the output signal of the integrator.

3.2 Continuous-Time 2-1 MASH ΣΔM

After a careful analyses of the behaviour of the integrators using UIS, this section presents a complete schematic of the differential 2-1 MASH ΣΔM, based on [1], and the analysis of it's behaviour and performance under different conditions.

3.2.1 Block Diagram and Transfer Functions

The block diagram of the modulator implemented in [1] is presented in Fig. 3.3.

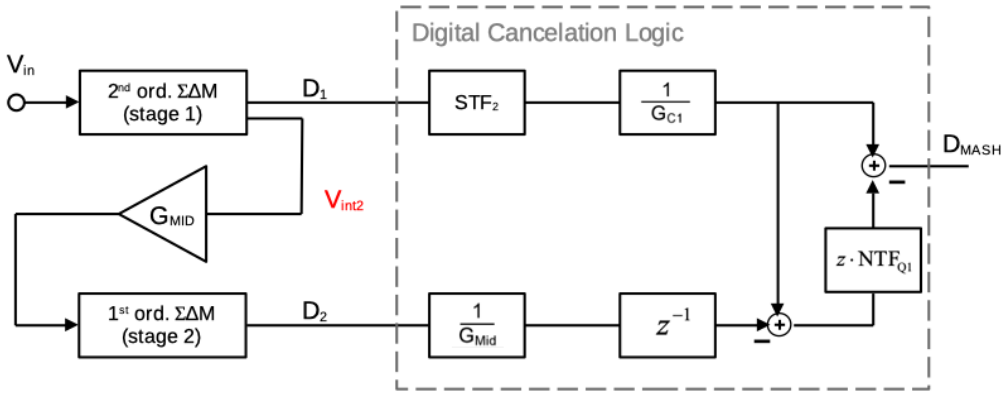


Figure 3.3: Block diagram of the 3^{rd} MASH ΣΔM, proposed in [1].

As one can see, this MASH modulator is composed by two stages, the first stage being a second-order ΣΔM and the second stage, a first-order ΣΔM. The block diagrams of these

stages are depicted in Figs. 3.4 and 3.5, respectively. The integrators are represented by their *H(z)* transfer functions (previously derived in the last section - Equation 3.15).

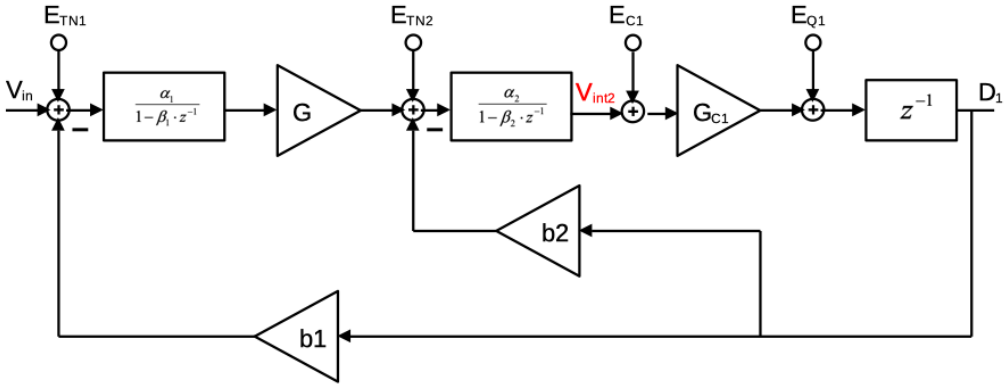


Figure 3.4: Block diagram of the 1^{st} stage, second-order ΣΔM, proposed in [1].

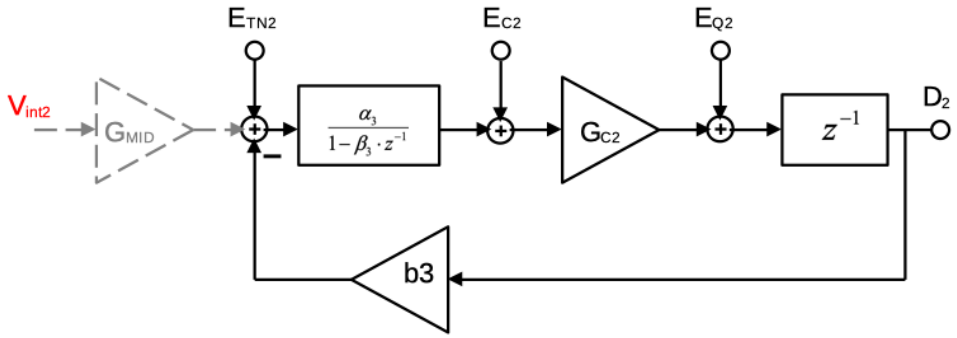


Figure 3.5: Block diagram of the 2^{nd} stage, first-order ΣΔM, proposed in [1].

The outputs of the two stages (D_1 and D_2) are applied to the digital cancellation (DCL) block, that combines them in order to cancel E_{Q1} and shape E_{Q2} by NTF_{Q1}, as illustrated in Fig. 3.3.

Here it is also possible to see that the input signal of the first-order ΣΔM, that is amplified by G_{MID}, V_{int2}, is also the input signal of the comparator of the second-order ΣΔM. The blocks G_{MID} and G are necessary, in order to separate the adjacent integrators to prevent loading effects and also provide gain (approximately of 20dB).

It is also worth mentioning that the coefficients b_1, b_2 and b_3, depicted in Figs. 3.4 and 3.5, represent the feedback coefficients, which can be approximated to the following equations.

$$b_1 \approx \frac{V_{ref1,2}}{\alpha_1} \cdot \frac{C_{f1}}{C_1 + C_{f1}} \qquad b_2 \approx \frac{V_{ref1,2}}{\alpha_2} \cdot \frac{C_{f2}}{C_2 + C_{f2}} \qquad b_3 \approx \frac{V_{ref3}}{\alpha_3} \cdot \frac{C_{f3}}{C_3 + C_{f3}} \tag{3.18}$$

Here $V_{ref1,2}$ is the reference voltage applied to the integrators from the first stage, and V_{ref3} is the reference voltage applied to the integrator from the second stage. The signal processing gains of the comparators, G_{C1} and G_{C2} can be written as the following equations.

$$G_{C1} \approx \frac{1}{\alpha_2 \cdot b_2} \qquad G_{C2} \approx \frac{1}{\alpha_3 \cdot b_3} \tag{3.19}$$

In order to increase these gains and be able to have more quantization noise attenuation, it is necessary to have $\alpha_{2,3} << 1$ and $b_{2,3} << 1$.

Finally, the signal and noise transfer functions of the second-order modulator (1^{st} MASH stage), are given by:

$$STF_1(z) = \frac{D_1}{V_{in}} = \frac{G \cdot G_{C1} \cdot z \cdot \alpha_1 \cdot \alpha_1}{z^2 + z \cdot (G_{C1} \cdot \alpha_2 \cdot (b_2 + G \cdot b_1 \cdot \alpha_1) - \beta_1 - \beta_2) + \beta_1 \cdot (-G_{C1} \cdot b_2 \cdot \alpha_2 + \beta_2)} \tag{3.20}$$

$$NTF_{Q1}(z) = \frac{D_1}{E_{Q1}} = \frac{(z - \beta_1) \cdot (z - \beta_2)}{z^2 + z \cdot (G_{C1} \cdot \alpha_2 \cdot (b_2 + G \cdot b_1 \cdot \alpha_1) - \beta_1 - \beta_2) + \beta_1 \cdot (\beta_2 - G_{C1} \cdot b_2 \cdot \alpha_2)} \tag{3.21}$$

One should know that the amplitude of V_{in} should not exceed b_1 and the maximum theoretical value of V_{in}/b_1 is 1, meaning that, at DC STF(z=1), is approximately $\frac{1}{b1}$, corresponding to the analog digital conversion factor (with units of $[1/V]$).

$$STF_1(z = 1) = \frac{G \cdot G_{C1}}{1 + G \cdot G_{C1} \cdot \left(b_1 + \frac{b_2}{G}\right)} \approx \frac{G \cdot G_{C1}}{G \cdot G_{C1} \cdot b_1} \approx \frac{1}{b_1} \tag{3.22}$$

$$NTF_{Q1}(z = 1) = \frac{1}{1 + G \cdot G_{C1} \cdot \left(b_1 + \frac{b_2}{G}\right)} \approx \frac{1}{G \cdot G_{C1} \cdot b_1} \tag{3.23}$$

$$NTF_{C1}(z = 1) = \frac{G_{C1}}{1 + G \cdot G_{C1} \cdot \left(b_1 + \frac{b_2}{G}\right)} \approx \frac{G_{C1}}{G \cdot G_{C1} \cdot b_1} \approx \frac{G_{C1}}{G \cdot b_1} \tag{3.24}$$

The equations of the signal and noise transfer functions of the first-order modulator (2^{nd} MASH stage), can be written as:

$$STF_2(z) = \frac{G_{C2} \cdot \alpha_3}{z + G_{C2} \cdot b_3 \cdot \alpha 3 - \beta_3} \tag{3.25}$$

$$NTF_{Q2}(z) = \frac{1 - \beta_3 \cdot z^{-1}}{z + G_{C2} \cdot b_3 \cdot \alpha 3 - \beta_3} \tag{3.26}$$

3.2.2 Circuit Implementation

Following the the description of the diagram block, seen in Fig. 3.3, the schematic of the differential circuit of the 2-1 MASH ΣΔM is now presented in Fig. 3.6. As one can see, the modulator is based on the passive integrators previously analyzed, in section 3.1.

Each integrator can be described by the coefficients, from equations 3.27 and 3.28.

$$\alpha_i = \frac{T_S}{2 \cdot R_i \cdot (C_i + C_{fi})} \tag{3.27}$$

$$\beta_i \approx \frac{C_i}{C_i + C_{fi}} \cdot (1 - \alpha_i) \tag{3.28}$$

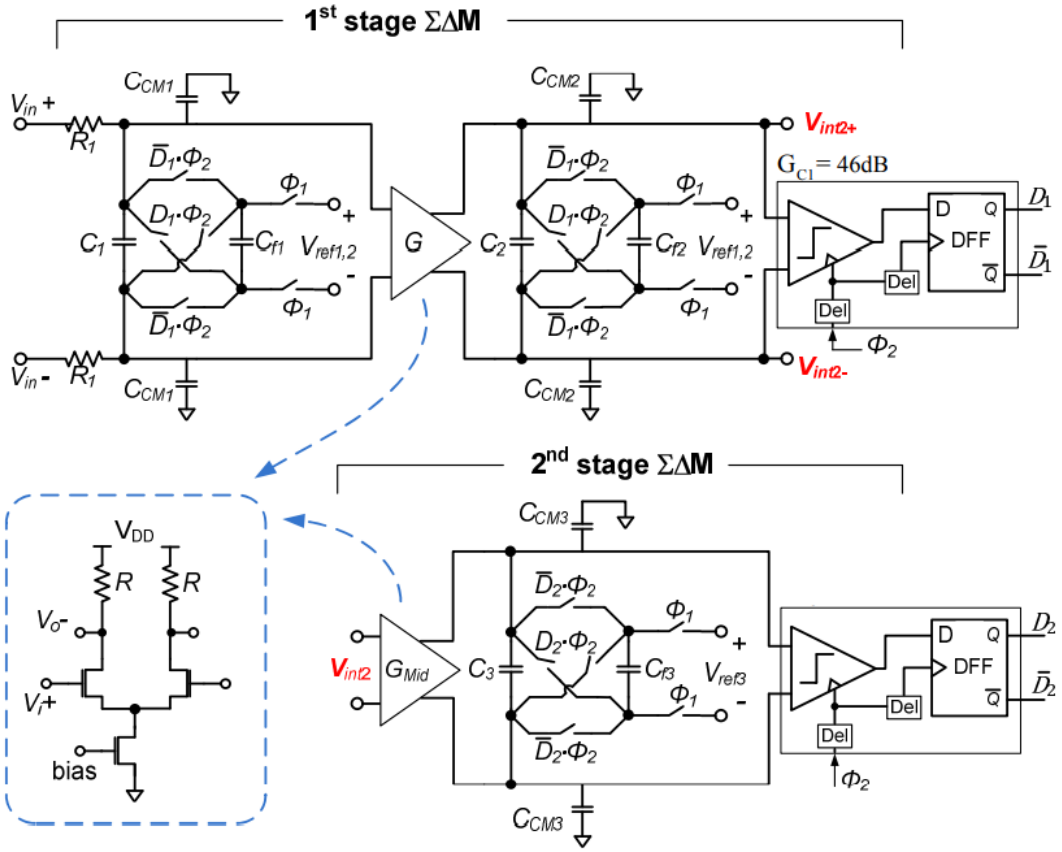


Figure 3.6: Schematic of the 2-1 MASH ΣΔM, from [1].

However, by analysing the schematic from Fig. 3.6 it is important to outline, that the capacitance of the two common-mode capacitors C_{CM} and the differential capacitor C (present in each integrator) amount to the value of the capacitance C_i, from the Equations 3.27 and 3.28. In fact, the common-mode capacitors are added to the circuit in order to reduce the integrator's commom-mode voltage output swing.

The clock signal used in this circuit is generated by a NAND based clock generator (Fig. 3.7). The internal inverters from this generator define the interval of time between two phases in order for it to be non-overlapping.

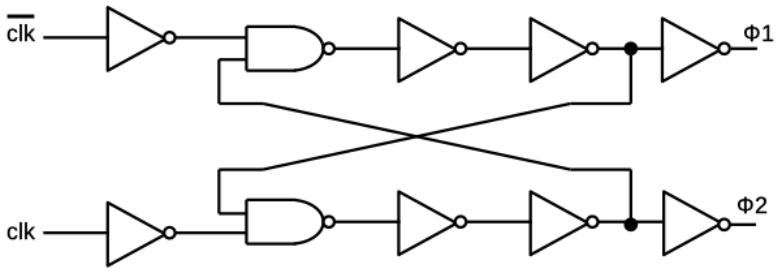


Figure 3.7: NAND based two-phases clock generator, used in [1].

Finally, the gain blocks are implemented as differential pairs loaded by resistors which are the load of the gain stages, their resistances provide a well determined gain ($G = gm \cdot R$) and, more importantly, these are also a part of the RC time constants of the second and third integrators.

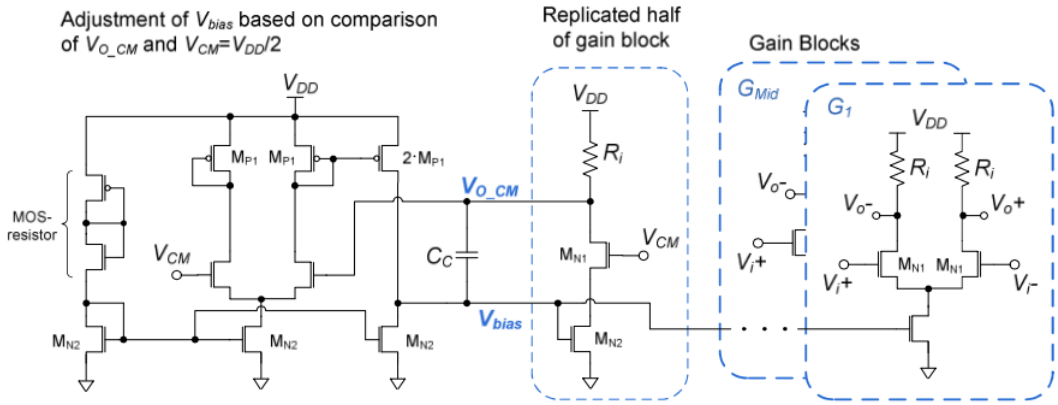


Figure 3.8: Gain blocks with the replica bias circuit, from [1].

As illustrated by Fig. 3.8, the gain blocks require a replica bias circuit to adjust the buffer's output common-mode voltages to the desired values, as it can deviate from it, due to temperature, process and supply voltage variations.

3.2.3 DCL Implementation

Considering that the objetcive of the DCL block is to cancel the quantization noise of the first modulator inside the signal's bandwidth, it is possible to simplify it's implementation, due to the fact that the poles from STF_2 and NTF_{Q1} are located at frequencies higher than the signal bandwidth.

Thus, it is only necessary to evaluate the denominators of STF_2 and NTF_{Q1} at z=1, and use the values obtained to scale the output signals of each stage. The modified DCL block that was implemented in the 2-1 MASH ΣΔM, is illustrated in Fig. 3.9. For the sake of avoiding the use of digital multiplier circuits, one can resort to the use of multiplexers controlled by two bit-streams of the two modulators to select between a positive and negative value for the coefficients, which in practical terms allows to have more power and hardware efficiency, then the typical implementation of the DCL block, seen previously in Fig. 3.3.

The coefficients seen in Fig. 3.9, can be derived from the STF_2 and NTF_{Q1}, as seen in the following equations:

$$DCL_{NTF_{Q1}} = \frac{Numerator(z \cdot NTF_{Q1}(z))}{Denominator(NTF_{Q1}(z=1))} = \frac{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}{1 + b_1 + b_2} \tag{3.29}$$

$$K = \frac{Denominator(STF_2(z=1))}{G_{C2} \cdot \frac{\alpha_3}{G_{C1}} \cdot G_{MID}} \tag{3.30}$$

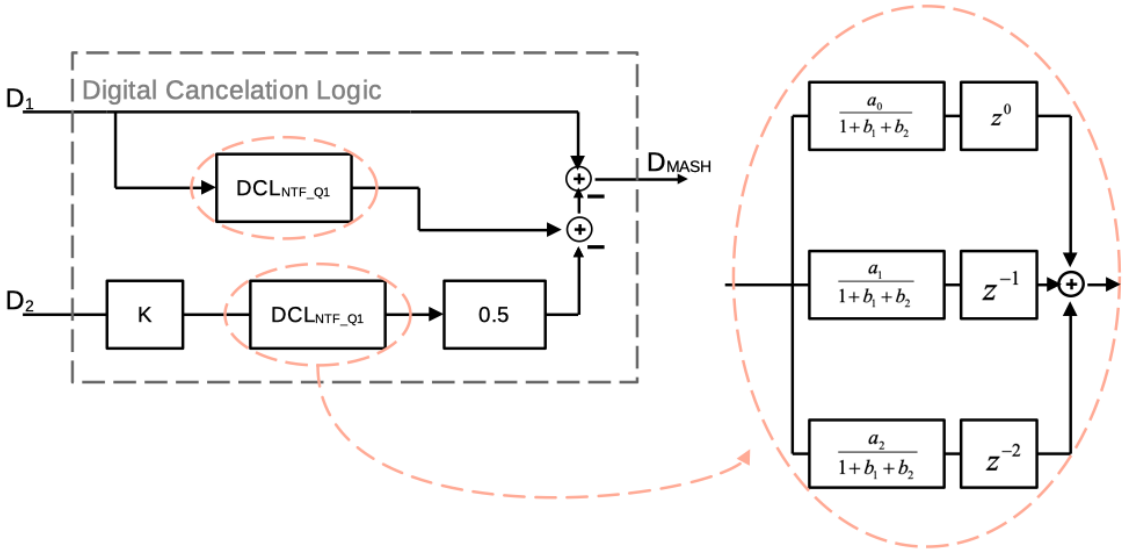


Figure 3.9: First modification of DCL block, proposed in [1].

However, it was possible to further simplify the DCL, by realising that this first modification, can be translated as a Finite Impulse Response (FIR) filter with the following inputs: $D_1[n-2]$, $D_1[n-1]$, $D_1[n]$, $D_2[n-2]$, $D_2[n-1]$, $D_2[n]$. Which can be implemented as a 6-bit Look-Up Table (LUT) (as it can be seen in the 5^{th} chapter from [1]), avoiding the use of area and power hungry multipliers.

3.2.4 Modulators Performance

For MASH modulators it is important to have a good matching between the digital and analog transfer functions. For this reason, an optimization procedure based on genetic algorithms, that takes into account thermal noise contributions as well as the effects of mismatch errors, is implemented in the design methodology of the MASH ΣΔM 2-1, developed in [1], in order to maximize the SNDR. Thus, this optimization aims at finding a solution with low sensitivity that maximizes the SNDR for a desired maximum input differential amplitude.

After the optimization, to determine the average value of the SNDR, a Monte-Carlo analysis is run with the respective process and mismatch variations of G, G_{MID}, C_i, C_{fi} and R_i. This analysis had 1000 cases and resulted in a peak value SNDR of 72.9 dB with a 1.3 % of standard deviation.

However, as stability and low sensitivity to the component's variation, is an important factor taken into consideration by the optimization process, a new simulation of the ΣΔM was run, varying each design parameter around the optimal design point, as seen in Fig. 3.10. As one can see, that the solution chosen by the optimization procedure, has, in fact, a low sensitivity to the component's variations, while achieving the intended SNDR value.

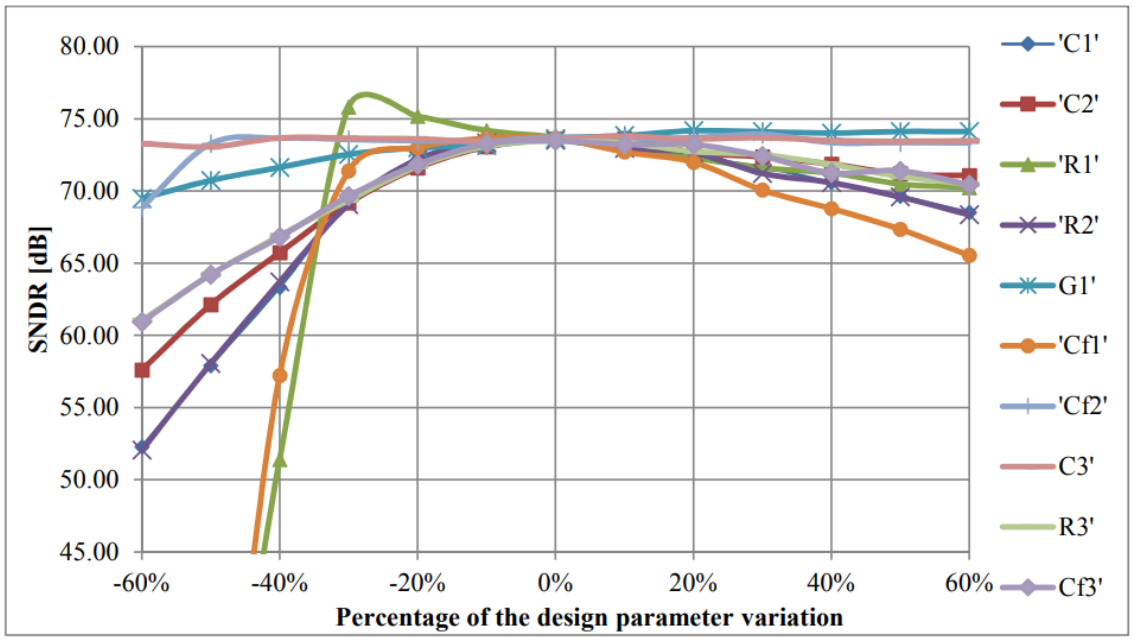


Figure 3.10: Graphic of SNDR of the MASH modulator vs. variation of the analog components around the optimal design point, selected by the genetic algorithm based optimizer, from [1].

Because of this, the coefficients in the DCL used to implement the 2-1 MASH ΣΔM correspond to the nominal values of the same coefficients in the analog circuits (α_1, β_1, b_1, α_2, β_2, b_2, α_3, β_3 and b_3). Nevertheless, it is possible to improve the circuit's performance, by decreasing the R_1 value, at the cost of increasing the modulator's sensitivity to the component's variations and possibly lead to it's overload. This conclusion encourages the study of the theoretical variation of the ΣΔM's performance when being submitted to different DCL calibrations. Ideally, calibrating the DCL's coefficients will allow to maintain the modulator's stability through process variations of RC components, while increasing the worst corner's performance. If this can be achieved, then, in the future, it would be possible to select a new design point that will allow for a less conservative solution, and possibly achieve, overall, a better performance.

3.3 Calibration Circuit

By studying the performance of the High-Level model of the 2-1 MASH ΣΔM, based of [1], in different conditions in a *MATLAB* ® environment, and studying different calibration techniques, it is possible to come to a conclusion regarding what should be the approach to calibrate this modulator. Thus, in this section different high-level simulations that corroborate the pursuit of the calibration of this circuit are presented and a calibration scheme is proposed. Moreover, the respective High-level Model of the calibration circuit, developed in *MATLAB* ®, will also be presented.

When analysing the RC calibration circuits, the article [50] is an example of how a

tuning circuit can be used in order to detect the RC time constant variation. Nevertheless, the calibration done in [50] is in the analog domain, as the value of the capacitance used in the circuit is altered, by resorting to the implementation of a digitally programmable capacitor array, C_{bank}, in parallel with C. This array is adjusted using a digital code, until reaching the ideal RC value. Analog calibration such as [50] that requires varying the value of the capacitors, can add extra thermal noise to the passive integrators and require switches that also can cause problems that introduce extra parasitic capacitances. Which is why, in this project, the calibration circuit, like in [50] must also be able to measure the value of the relevant analog circuit's parameters, nevertheless, instead of tuning them, the objective is to convert these values into a digital coefficient. This coefficient can be used in the circuit's digital cancellation logic, in order to correct the effects of the components variations and be able to maintain the performance of the modulator through process, voltage and temperature (PVT) corners variations, while increasing the worst corner's performance.

Therefore, ideally, the proposed calibration circuit must be able to read the values of R_i and C_i of the integrators, used in the 2-1 MASH ΣΔM, and calculate their deviation from the ideal RC time constant value. This deviation causes a variation of α (Eq. 3.27), resulting in a mismatch between the circuit's analog coefficients and the DCL digital coefficients, resulting in a noise leakage in the DCL, thus decreasing the SNDR. Assuming that the value of the $R \cdot C$ variation is known, it is possible to calculate a correction factor in order to update all the coefficients in DCL to reflect this variation, because ultimately all the coefficients depend on the ratio $\frac{T_s}{RC}$ which determines α, as seen in Equation 3.27.

In order to verify that this is, in fact, a valid approach, high-level simulations were run in *MATLAB* ® to study the circuit's components variation and better understand the outcome of the modulator with the implementation of an ideal calibration.

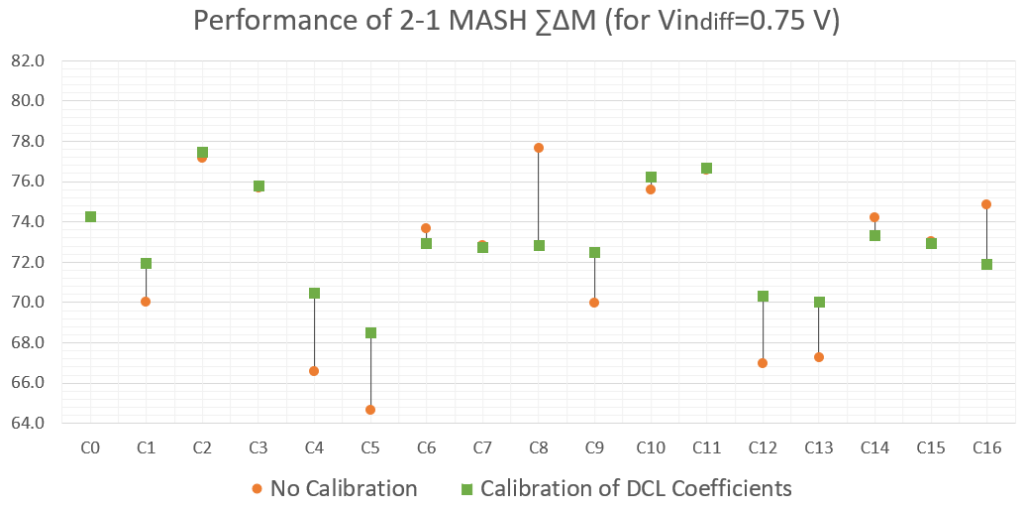
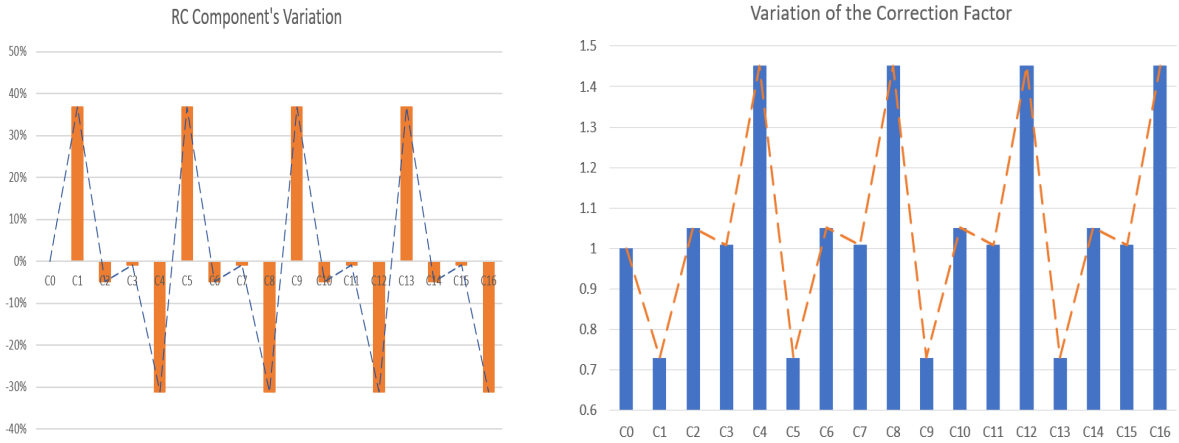


Figure 3.11: Graphic of the SNDR variation for different corners, for an amplitude of Vin of 0.75V.

In the graphic depicted in Fig. 3.11 it is possible to see the results of modulator's performance for different parameters variations, presented in Table 3.1, for an amplitude of a differential input voltage, $V_{in,diff}$, of 0.75 V, in two different scenarios. In the first scenario, the 2-1 MASH ΣΔM performs as explained through out the last section, without any kind of calibration. In the second scenario, the calibration of α's is presented, considering that the DCL coefficients are updated to reflect the exact $R \cdot C$ variation (ideal calibration). In this case, it is possible to calculate the ideal correction factor, considering the components variations, and adjust the values of alpha with the known DCL transfer functions. Consequently, the variation of the correction factor reflects the variation of the of the RC components, as illustrated in Figs. 3.12a and 3.12b.

Table 3.1: Variations of the analog design parameters of the 2-1 MASH ΣΔM, in the High-Level simulations, considered in *MATLAB* ®.

Variation	**R**	**C**	G_1, G_{MID}	**Vref**
Maximum	+18%	+16%	+13%	+5%
Minimum	-18%	-16%	-13%	-5%



(a) $R \cdot C$ Component's Variation, in percentage.

(b) Absolute variation of the correction factor.

Figure 3.12: Variation of (a) $R \cdot C$ Component's and (b) the correction factor, with outlined trend-lines characteristics, when simulating with variation of Design Parameters from Table 3.1.

As it can be seen in both Figures 3.12a and 3.12b, the first corner (C0), illustrates the nominal conditions of the analog circuit, which means that there is no $R \cdot C$ variation in this case. It is important to point out that the design parameters variation considered in *MATLAB* ®, illustrated in Fig. 3.12a and depicted Table 3.1, is chosen based on the results from previous electric simulations which were analysed to study the effects of process variation in the relevant analog parameters. This allows to define the maximum and minimum values for the design parameters variations, considered in *MATLAB* ®

simulations. One can notice, by looking at Fig. 3.11, that the results of the ideal alpha calibration are not always better then when no calibration is applied. However, what is important to retain from this graphic is that in the worst cases (with lower SNDR values), the performance of the circuit is always improved with the ideal alpha calibration. Nonetheless the calibration still does not present significant improvement in the overall performance of the modulator.

A possible reason for this, is the influence of the component's variations in the feedback coefficients. As previously seen, in Equation 3.18 the feedback coefficients (b_1, b_2 and b_3) are defined depending on the voltage V_{ref_i}, the capacitors C_i and C_{f_i} and the values of α_i. Thus, the values of the RC components and their respective variations have a direct impact on the feedback coefficients (Fig. 3.13), which in turn affect the amplitude of the input signal of each stage of the modulator (as illustrated by Figs. 3.4 and 3.5).

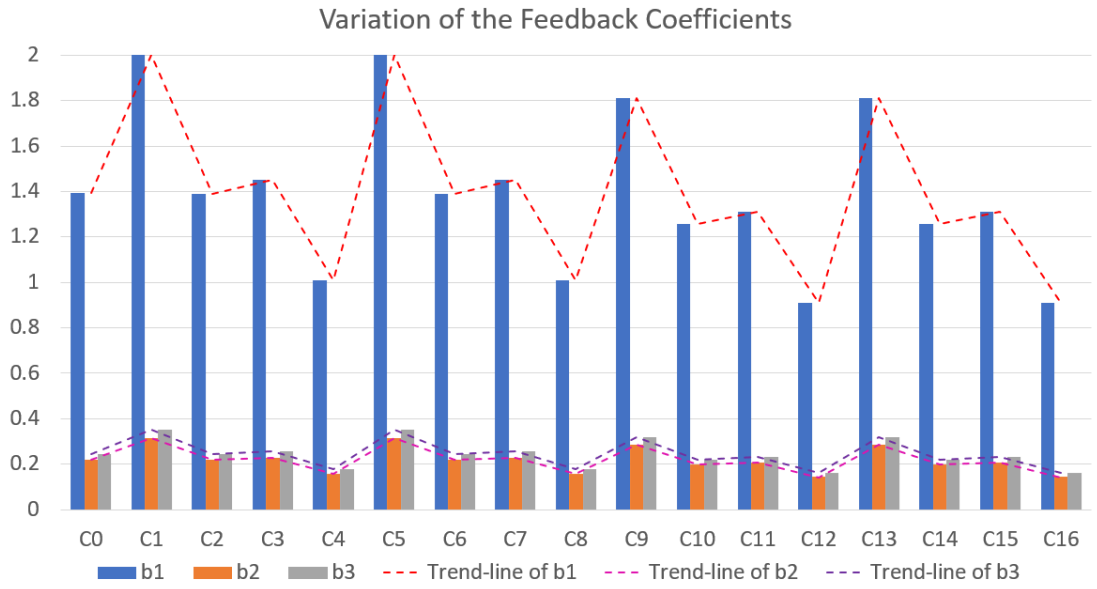


Figure 3.13: Absolute variation of the feedback coefficients, for corners simulation, with their respective trend-lines.

To that end, the high-level model of the ΣΔM was used to obtain the SNDR value as function of the input amplitude for the different parameter variations, presented in Table 3.1. The simulations results, depicted in Fig. 3.14, reveal that the range between the maximum and minimum values of the SNDR is [65.24; 77.30] dB, for a $V_{in,diff}$ of 0.75V.

Then, a high level model of the MASH ΣΔM, considering that the DCL coefficients were updated to reflect the $R \cdot C$ variation (ideal calibration), allowed to obtain the SNDR results as a function of the $V_{in,diff}$. The results of this model, for different corners (shown in Table 3.1), are presented in Fig. 3.15 and it is possible to see that the range between the maximum and minimum values of the SNDR is of [66.41; 78.05] dB, for a differential input voltage of 0.75V.

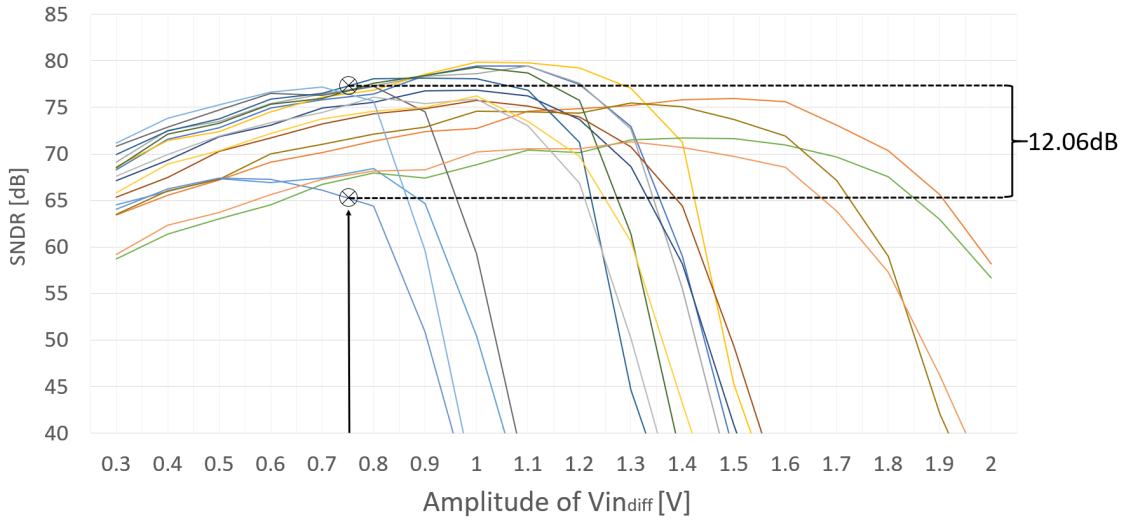


Figure 3.14: Performance of the high level model of the 2-1 MASH ΣΔM (no calibration) , for $V_{in,diff}$, when simulating with variation of Design Parameters from Table 3.1.

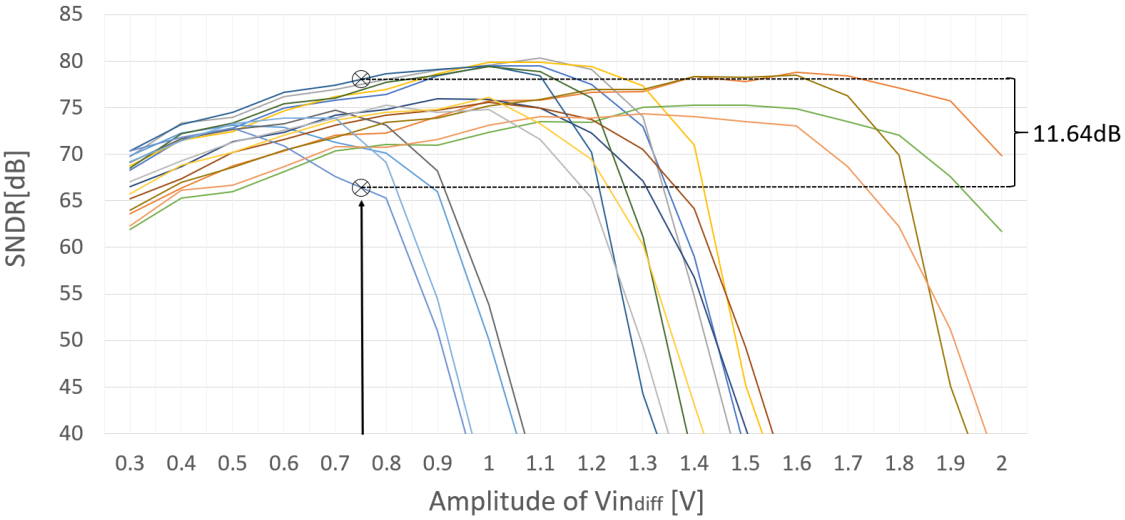


Figure 3.15: Performance of the high level model of the 2-1 MASH ΣΔM with calibration of the DCL coefficients , for $V_{in,diff}$, when simulating with variation of Design Parameters from Table 3.1.

The SNDR graphs of Figs. 3.14 and 3.15 show that the $V_{in,diff}$ that causes the modulator to overload changes depending on the corner. As the variation of the reference voltage (V_{ref}) is smaller than the variation of R and C, the overload amplitudes values are grouped around 3 values, corresponding the variations of $R \cdot C$. Since the design specification used in the optimization process called for a maximum $V_{in,diff}$ of 0.75V, this forced the selection of a design point that was very conservative.

The overload amplitude is directly dependent on the the gain of the ΣΔM, since this gain depends on V_{ref} (eq. 3.18), it is possible to adjust V_{ref} according to the RC variation, thus reducing the total feedback variations. The results can be seen in Figures 3.16 and

3.17. By analysing these graphics it is clear that not only, it is possible to reduce the total feedback variations (Fig 3.16) but also, Fig. 3.17 clearly shows that calibrating both the DCL coefficients and the V_{ref} to compensate for $R \cdot C$ variations allows to increase the SNDR to [71.73; 79.30] dB while also increasing $V_{in,diff}$ to 1.1V.

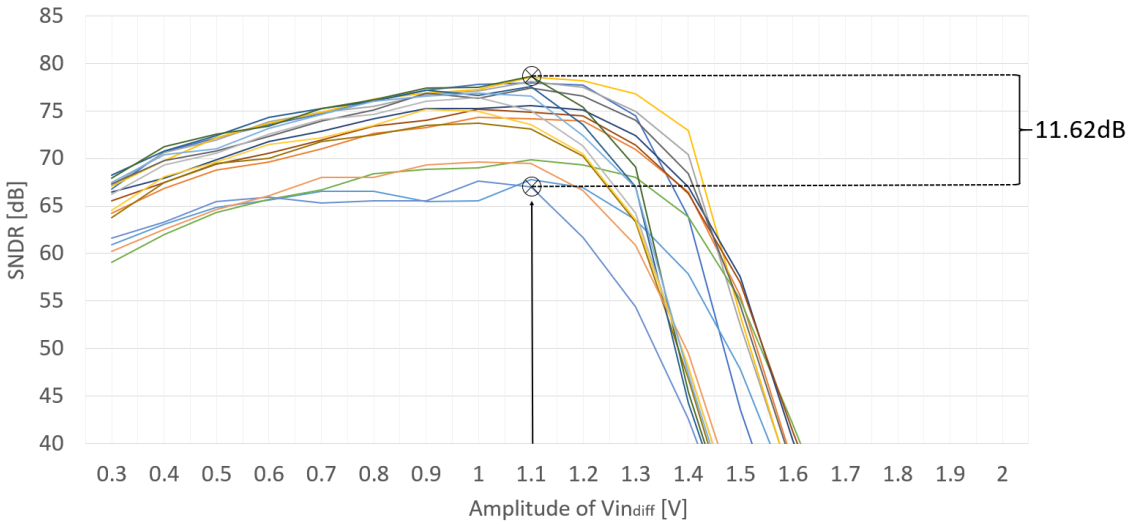


Figure 3.16: Performance of the high level model of the MASH ΣΔM with calibration of Vref, for different differential input amplitudes, when simulating for the corners illustrated in Table 3.1.

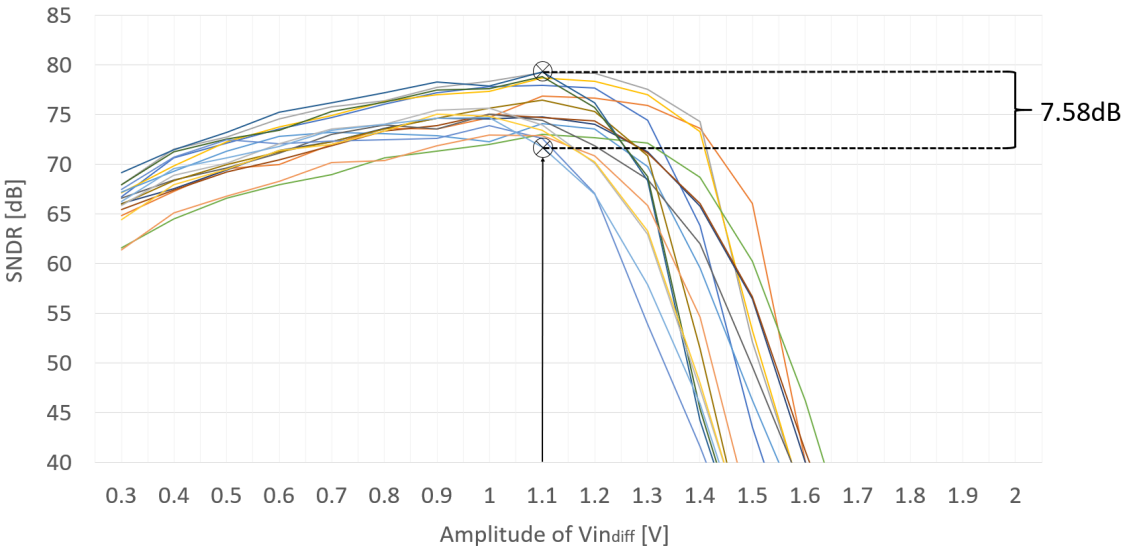


Figure 3.17: Performance of the high level model of the MASH ΣΔM with calibration of Vref and DCL coefficients, for different differential input amplitudes, when simulating for the corners illustrated in Table 3.1.

However, since the required range for adjusting V_{ref} is ($\approx [0.7; 1.7]$ V) and the maximum available voltage is 1.1V, it is necessary to adjust the values of the feedback capacitors, C_{fi}, to be able to obtain a smaller range of variation for V_{ref}, as illustrated by the following equation 3.31.

$$b_i \approx \frac{V_{ref_i} \cdot 0.635}{\alpha_i} \cdot \frac{C_{fi}/0.635}{C_{eq_i}} \tag{3.31}$$

With this adjustment it is possible to maintain the same feedback coefficients while using a much more appropriate range of values for Vref ($\approx [0.497; 1.1]$ V), as this tuning factor, of 0.635, allows to take advantage of the maximum available voltage. In Figure 3.18 it's possible to see the results of the modulator's performance in the same conditions, as previously seen in Fig. 3.11, versus the results of the modulator's performance with the new alterations, for a $V_{in,diff}$ of 1.1 V.

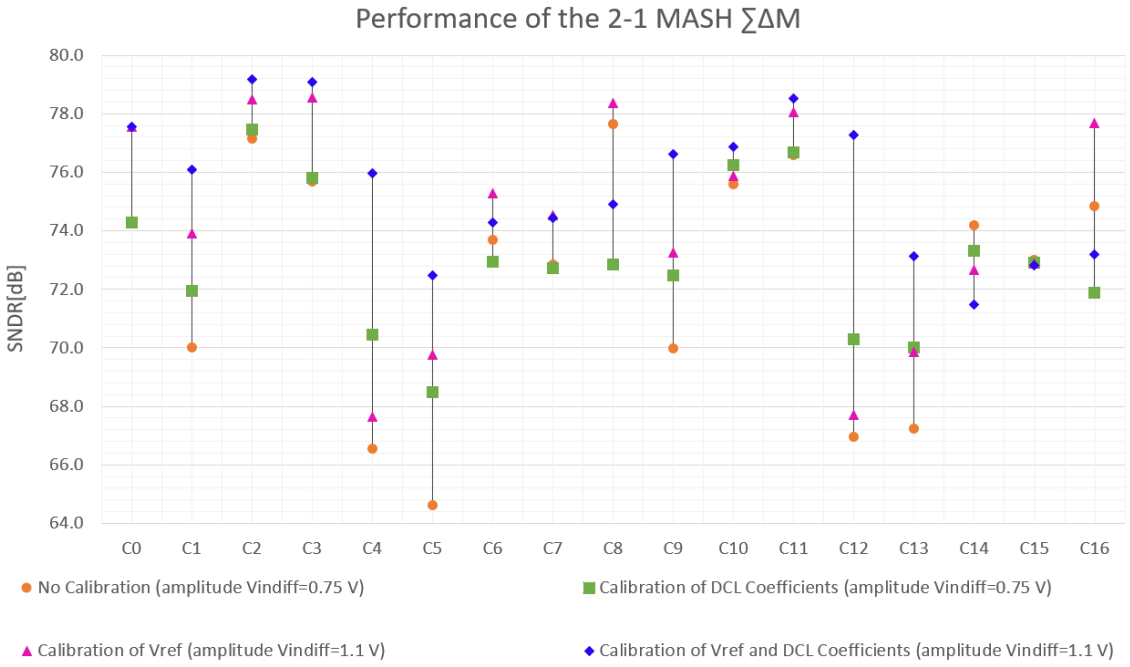


Figure 3.18: Graphic of the SNDR variation for different corners, testing the performance of different DCL's in MASH modulator for an amplitude of $V_{in,diff}$ of 0.75V Vs. an amplitude of $V_{in,diff}$ of 1.1 V with implementation of the calibration of α's in the DCL and reference voltage.

It is possible conclude that, in general, a better performance of the modulator can be obtained with the ideal calibration of the α in the DCL and reference voltage, since it is possible to decrease the varibility of the results for different corners, while also increasing worst corner performance.

Therefore, in order to calibrate the 2-1 MASH ΣΔM for RC components process variations, it's essential to develop a calibration circuit that it is capable of measuring the value of $\frac{T_s}{RC}$ in order to generate a correction factor which will calibrate α in the DCL transfer functions, improving quantization noise cancellation. Additionally it will also adjust the V_{ref} of the ADC in order to decrease the variability of the feedback signals.

3.3.1 Proposed α Calibration Architecture

The proposed architecture to implement the calibration concept presented in this thesis, is illustrated in Fig. 3.19, which depicts a block diagram of 2-1 MASH ΣΔM with the calibration circuit and calibrated DCL block and V_{ref} voltage. A theoretical analysis of the calibration architecture is presented in this section, to have a general idea of how the 2-1 MASH ΣΔM can operate with the calibration.

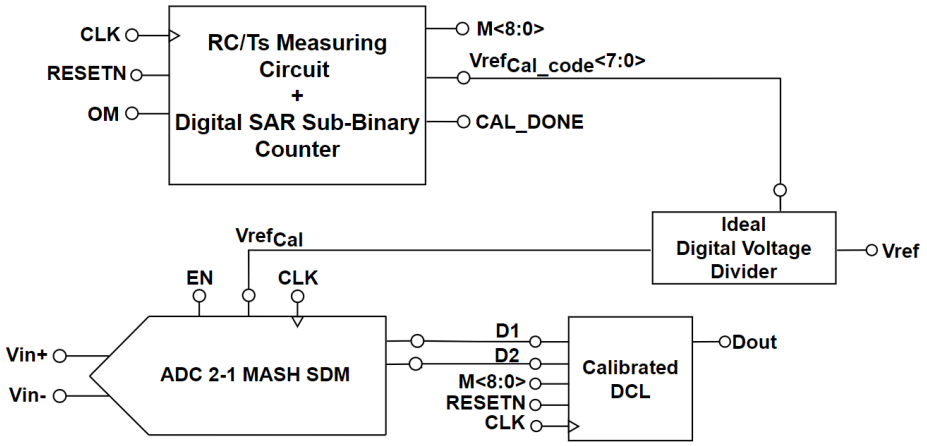


Figure 3.19: Block Diagram of 2-1 MASH ΣΔM with the calibration circuit and calibrated DCL block and V_{ref} voltage.

In order to properly measure the values of the relevant analog components it was necessary to develop an analog $\frac{RC}{T_S}$ measuring circuit (Fig. 3.21) and a digital counter (Fig. 3.23). The resistors and capacitors, used in the $\frac{RC}{T_S}$ measuring circuit have the same values as the ones used on the passive integrators in the MASH 2-1 ΣΔ Modulator. In this way is possible to analyse them and their variation to produce a factor M that indicates the correction factor needed in the DCL and enables the adjustment of the reference voltage. This factor is determined by the number of T_s clock cycles (=M) that take to charge capacitor C (which has the same capacitance as $C_{i=2}$), to a certain voltage value, using current I_1' (Fig. 3.21). In this way, it will take different periods of time to charge C, according to the process variations being simulated, allowing to measure the RC time constant through different conditions. To perform this operation a digital counter is essential, as it will register the M values being measured and produce two digital outputs, which correspond to a certain correction factor. The DCL will be calibrated through the digital output signal M with a precision of 9 bits (M[8:0]). To adjust the reference voltage it was decided to resort to an ideal digital voltage divider triggered by $V_{Ref_{Cal\_Code}}$ signal of 8 bits resolution, that will generate the new reference voltage $V_{Ref_{Cal}}$, as illustrated by Fig. 3.19.

For the correct operation of the calibration circuit to be possible, it was necessary to define an Operating Mode (OM) signal. This is an active-high signal of two bits, as illustrated in Fig.3.20. This signal, has three states defined: off, stand-by and active. Thus,

it is possible to perform multiple calibrations that can happen one at-a-time.

The OM signal was designed specifically to accommodate the needs of the calibration circuit, as it is not possible to start the calibration right away due to the necessity of settling different voltages, this is why there is a stand-by state. Once the signal goes from the off state to the stand-by state all the analog blocks start to operate, since it's necessary for some signals to settle before the the actual calibration process starts (in the active state), in order not to cause any kind of offset to the M value. Therefore, the all calibration is completely dependent on this signal.

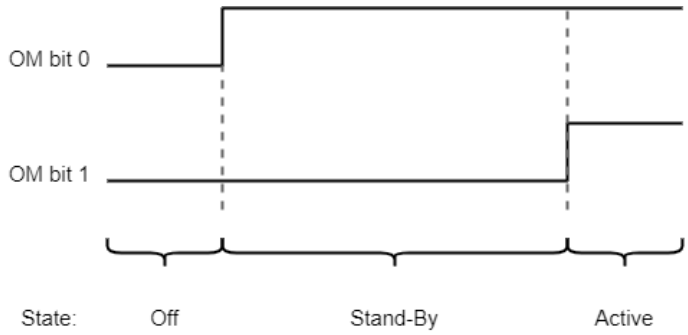


Figure 3.20: Operating Mode signal.

Additionally, to have a better understanding of this circuit's implementation, it should be known that the RESETN and EN signals (Fig. 3.19) enable the circuits to operate, the signal CAL_DONE indicates when the calibration is over, CLK defines the clock signal used for the operation of the different blocks, and D1 and D2, define the digital output signals of the 1^{st} and 2^{nd} order $\Sigma\Delta$Ms of the MASH. A more detailed description of the operation and implementation of these circuits will be presented further.

3.3.1.1 $\frac{RC}{T_S}$ Measuring Circuit

To be able to measure the correct $R \cdot C$ variation, the circuit presented in Fig. 3.21 is used.

In this circuit, current I_1 is defined by V_{refK} and R_x (through a feedback loop using an opamp) and then is mirrored from M1 to M4 (cascode current sources) and is given by:

$$I_1 = I_1' = V_{refK}/R_X \tag{3.32}$$

I_1' is applied to the RC circuit. The voltage $Vout_{RC}$ as function of time (after resetting C), is given by:

$$Vout_{RC}(t) = I_1' \cdot R \cdot (1 - e^{-t/R \cdot C}) \tag{3.33}$$

As illustrated in Fig. 3.22, a digital block counts M clock cycles, at this time instant the RC voltage is:

$$Vout_{RC}(M \cdot T_s) = V_{ref} \cdot (1 - e^{-M \cdot T_s/R \cdot C}) \tag{3.34}$$

By defining an RC time constant, Tx, as:

$$T_x = R \cdot C \qquad\qquad T_x = M \cdot T_S \tag{3.35}$$

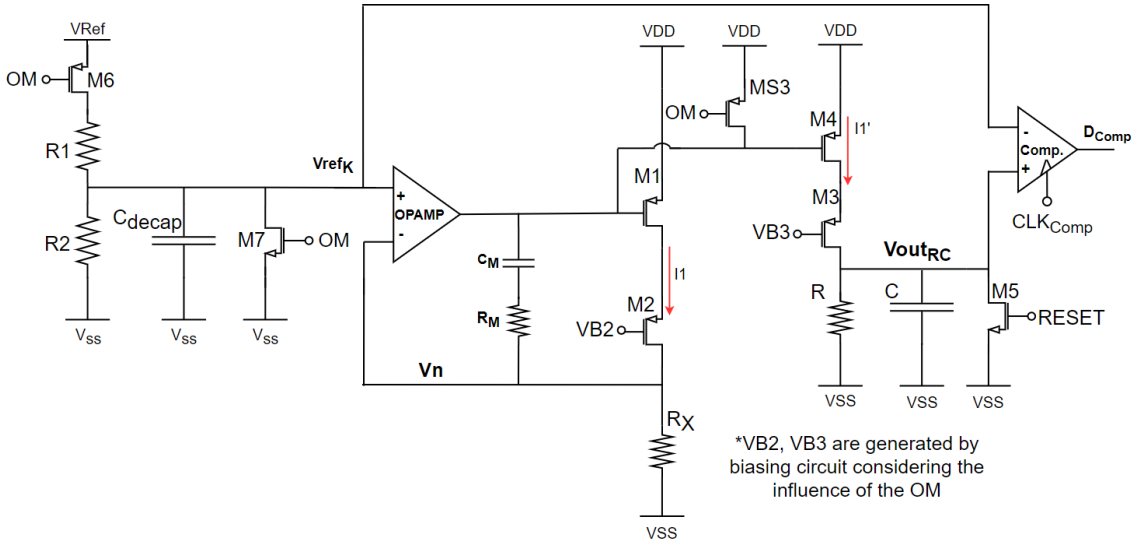


Figure 3.21: Block diagram of the $\frac{RC}{T_S}$ measuring circuit.

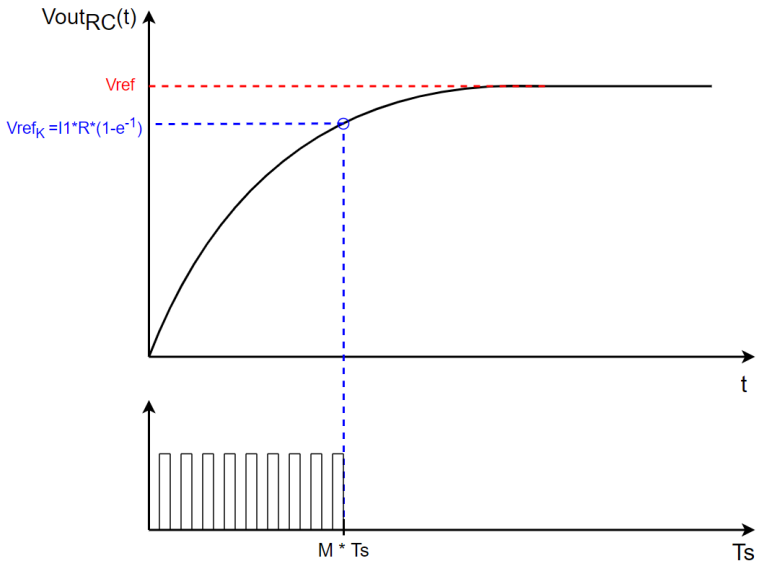


Figure 3.22: Graphic of the development of $Vout_{RC}$ through time and the reaching of the M value.

Then it's possible to compare the voltage $Vout_{RC}$ to V_{refK}, resulting in the digital output signal of the comparator, D_{Comp}. After this output, the RESET signal is activated and the cycle is repeated. If D_{Comp} is 0, the value of M is decreased, otherwise it is increased for the next measurement cycle. The value of M is adjusted until, on average, $Vout_{RC}$ is equal to:

$$Vout_{RC}(T_x) = V_{ref} \cdot (1 - e^{-M \cdot T_s / R \cdot C}) = V_{refK} \tag{3.36}$$

Considering that:

$$V_{ref} = I_1' \cdot R; \quad R_X = K \cdot R\,; \quad K = 1 - e^{-1} \tag{3.37}$$

$$Vout_{RC}(T_x) = Vref \cdot (1 - e^{-1}) \qquad V_{refK} = I_1 \cdot R_X \tag{3.38}$$

Since:

$$V_{refK} = \frac{V_{ref} \cdot R_x}{R} = V_{ref} \cdot K \tag{3.39}$$

$$Vout_{RC}(t) = I_1' \cdot R \cdot (1 - e^{-t/R \cdot C}) \tag{3.40}$$

$$t = Tx \tag{3.41}$$

This results in:

$$(1 - e^{-M \cdot T_s/R \cdot C}) = K \tag{3.42}$$

$$e^{-1} = e^{-M \cdot T_S/R \cdot C} \qquad \Leftrightarrow \qquad M = \frac{R \cdot C}{T_S} \tag{3.43}$$

Therefore, it's also possible to establish a relationship between the M value, and the α coefficient.

$$\alpha = \frac{T_S}{2 \cdot R_2 \cdot (C_2 + C_{f_2})} \qquad C = C_2 \qquad R = 2 \cdot R_2 \tag{3.44}$$

$$M = \frac{R \cdot C}{T_S} \qquad M \approx \frac{1}{\alpha} \tag{3.45}$$

Since $C_{f_2} << C_2$ the variation of the feedback capacitors is not considered in the measurement for being so small in comparison with $R \cdot C$ variation, which causes the main effect to the α coefficients. Since the $R \cdot C$ variation will affect all the integrators in the same way, it is only necessary to measure the RC variation from one integrator, and apply it in all α_i coefficients, in the DCL's transfer functions. For the nominal values of R and C, M is equal to M_{Nom}, and the correction factor is given by:

$$CorrectionFactor = \frac{M_{Meas}}{M_{Nom}} \tag{3.46}$$

Where M_{Meas} is the value measured by the calibration circuit. In the used technology the $R \cdot C$ variation has a range of [-31%; +37%], as seen in Fig. 3.12a, which define a maximum and minimum values of M as $M_{Max} = 263$ and $M_{Min} = 131$, with a nominal value of $M_{Nom} = 191$ (typical conditions).

3.3.1.2 Digital SAR Sub-Binary Counter

In order to be able to obtain different values of M inside this range, it is necessary to use the digital counter block (Fig. 3.23) to count the number of clock cycles that equals the RC time constant, that corresponds to value of M, according to the $R \cdot C$ variation being measured, as illustrated in Fig. 3.22.

It is important to know that when calculating the ideal value of M according to the $R \cdot C$ variation, the result can be a fractional number, however, this number is always rounded to an integer in order to be possible to obtain a result by the counter.

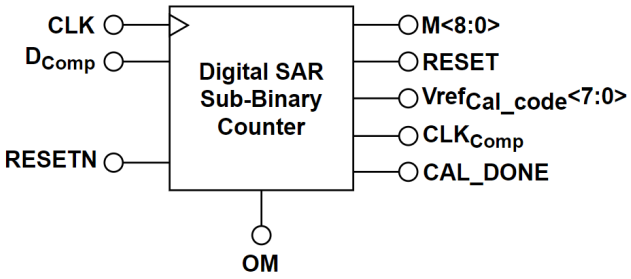


Figure 3.23: Block diagram of the digital counter of the calibration circuit.

Figure 3.24 illustrates the implementation of a simple counter, in *MATLAB* ®, in order to calculate the necessary M for a variation of the $R \cdot C$ components of +20% , which translates to a necessary M of approximately 229.

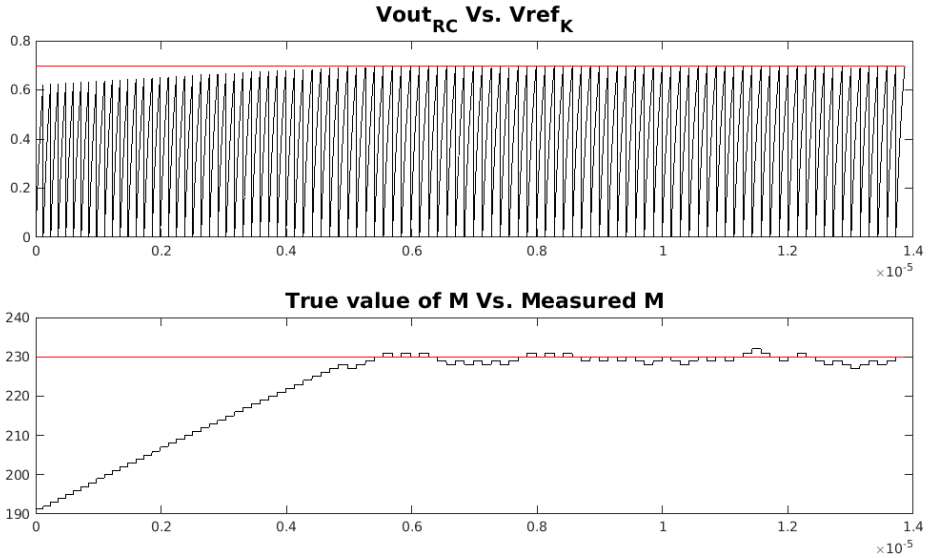


Figure 3.24: Intended output voltage of the calibration circuit being achieved while the circuit measures the value of M, using a simple counter. In the first graphic, the y - axis represents the voltage, while in the second graphic, it represents the values of M. In both graphics, the x-axis represents time.

In this figure, there are two graphics depicted. In the first graphic, the red line shows the value of $Vref_K$ and the black one indicates the variation of $Vout_{RC}$. In the graphic below, the red line indicates the desired M value, for 20% $R \cdot C$ variation, and the black one indicates the value of M being measured by the counter, in each clock cycle.

One should notice that by using this simple search algorithm, every time a clock cycle is counted, and the M hasn't reached the desired value, the capacitor from the calibration

circuit needs to be recharged (done by RESET signal - Figs. 3.21 and 3.23), in order for $Vout_{RC}$ to be compared with $Vref_K$ again and the M value to be incremented. Hence, implementing a simple counter is time consuming. Therefore, in order to speed up this process, the hypothesis of implementing a Successive Approximation Register (SAR) binary search algorithm was also studied in *MATLAB* ®. To implement this algorithm an initial search range ([-31%; +37%]) must be defined and in each decision step, the range is divided by two. So in this case, instead of having a decrement/increment of 1 in each decision step, when comparing $Vout_{RC}$ to $Vref_K$, it is possible to define an interval of values in which the M factor can vary, for every step. As the RC variation is known, this is a very simple algorithm to implement. Nevertheless, as the search range is divided by 2 in each decision step, with no overlap of the search ranges in the next decision, this means that if an error occurs, within any of the first 3 steps, even if all the subsequent bit decisions are correct, then the final decision threshold will have a significant error associated to it.

For this reason, an alternative algorithm is used, the SAR sub-binary search algorithm. In this case, a new technique is implemented which consists of introducing redundant decision steps in the SAR binary search algorithm, such that dynamic errors that occurred in the first bit decisions can be later corrected. It was also decided to increase the maximum search range to 40%, as another precaution, in order to make sure that the algorithm is capable of reaching every possible M value, according to the $R \cdot C$ variation. As a result, in the same conditions as the graphic presented in Fig. 3.24 the graphic illustrated in Fig. 3.25 was obtained in *MATLAB* ®, and it's possible to see a much faster implementation of the counter which facilitates the search for the M value.

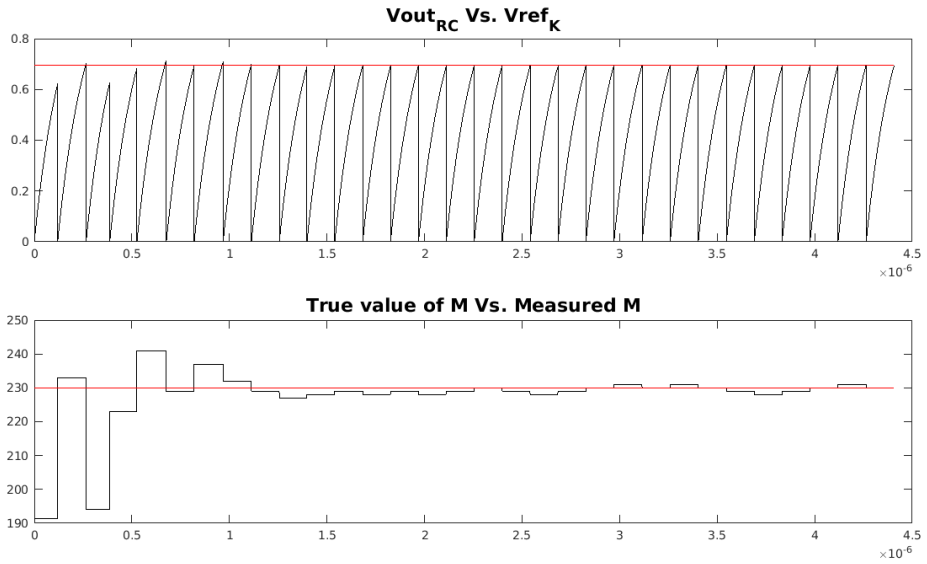


Figure 3.25: Intended output voltage of the calibration circuit being achieved while the circuit measures the value of M, using the sub-binary SAR search algorithm. In the first graphic, the y - axis represents the voltage, while in the second graphic, it represents the values of M. In both graphics, the x-axis represents time.

Thus, it was decided to implement a SAR sub-binary search algorithm, as faster solution to achieve the M value. Finally, the counter then returns this value as a digital output with 9 bits to the new proposed DCL block. The M value needs this accuracy of 9 bits, as the maximum possible M value exceeds the 8 bits limit ($M_{Max} = 263 > 2^8 = 256$).

3.3.1.3 Ideal Digital Voltage Divider

Besides M[8:0], the digital counter has another important digital output signal which is $Vref_{Cal\_code}$. This signal represents the correction factor that calibrates the feedback coefficients, more precisely calibrates the reference voltage. $Vref_{Cal\_code}$ determines the output tap of a digital resistive voltage divider that generates the adjusted reference voltage, V_{refCal} (Fig. 3.19).

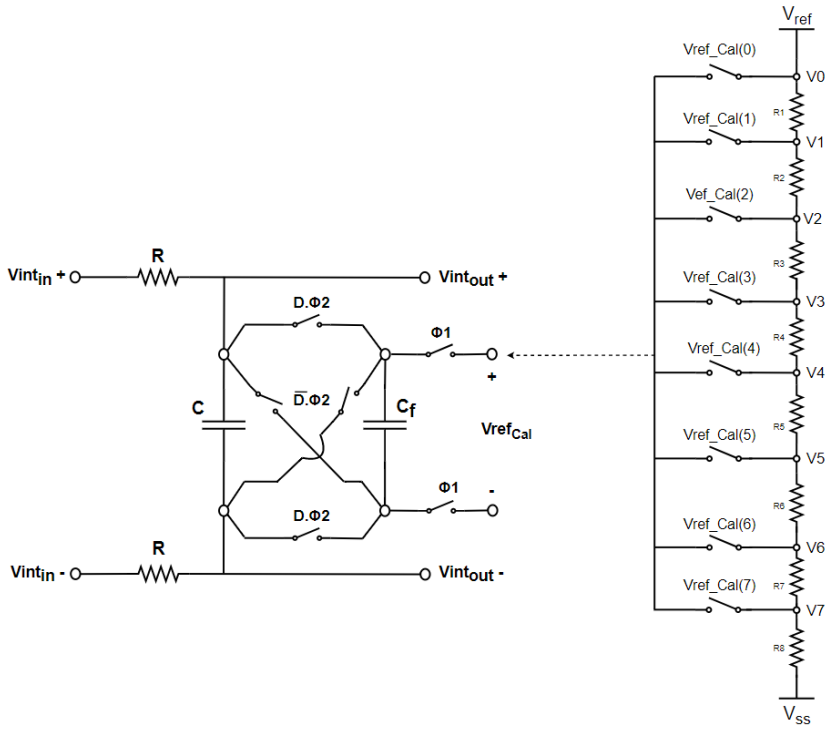


Figure 3.26: Ideal digital voltage divider used for the implementation of the Vref calibration.

Ideally, this factor should be the same as the one calibrating the RC components variations, as seen previously, in Section 3.3. However, the calibration of the feedback coefficients will be implemented with a digital resistive voltage divider, and in order to avoid added circuitry complexity it was decided to study the variation of the performance of the MASH 2-1 ΣΔ Modulator with a lower accuracy of the correction factor value. This study was conducted in *MATLAB* ® and the results showed that by reading the correction with a precision of only 3 bits, the maximum error associated to this change is of 3%, which allows for an easier implementation of the calibration of the feedback coefficients

by selecting one of 8 possible values for V_{ref}. In Fig. 3.26 it's possible to see a simple schematic of how this calibration will be done.

As the calibration studied with the high-level model, developed in *MATLAB* ®, presented good results with a precision of 3 bits for the correction factor used to adjust the value of V_{refCal}, it was decided to study the range of the obtained V_{refCal} values, in order to implement a digital resistive voltage divider to calibrate the reference voltage. By knowing the maximum range of V_{refCal}, [0.497; 1.1] V, it's possible to divide this range into 8 different common values that will be implemented in the digital resistive voltage divider, and access to, by the the result of the digital signal $Vref_{Cal\_code}$ of 3 bits ($2^3 = 8$), which is determined by the M value. To implement this, a one-hot logic was used in the digital sub-binary counter, as it can be seen in Table 3.2.

Table 3.2: Ideal range of values of Vref for every code, in a 3-Bit Logic and a One-Hot Logic.

Range of M	V	[V]	3-Bit Logic	One-Hot Logic
[131,147]	V_0	1.1	000	0000 0001
[148,164]	V_1	1.01395	001	0000 0010
[165;181]	V_2	0.92798	010	0000 0100
[182;198]	V_3	0.84192	011	0000 1000
[199;215]	V_4	0.75586	100	0001 0000
[216;232]	V_5	0.6698	101	0010 0000
[233;249]	V_6	0.58374	110	0100 0000
[250;266]	V_7	0.49768	111	1000 0000

3.3.1.4 Calibrated Digital Cancellation Logic

Finally, a block diagram of the calibrated DCL can be seen in Fig. 3.27. As it is necessary to update all the coefficients in the DCL to reflect the $R \cdot C$ variation and each M is unique and corresponds to a certain correction factor, and it's range of values is known, it's possible to previously calculate the impact of each M (i.e correction factor) in the α coefficients from the transfer functions in order to implement different finite impulse response filters (implemented as 6-bit LUTs) accessible through the measured M value, as an index.

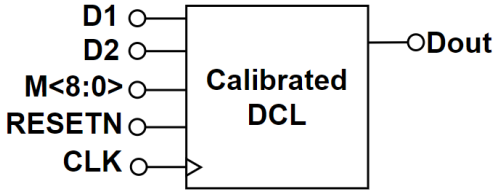


Figure 3.27: Diagram block of the calibrated DCL.

However, as there is a range of 133 M's this would require 133 different DCL LUT's. As an attempt to lower the required area for all these different tables, several simulations were run in order to understand if this number could be reduced, and it was realized that it is possible to use only 20 tables, with each table covering an interval of 7 M's ($7 \cdot 19 = 133$) plus an extra default table with nominal DCL values (for typical conditions).

3.3.2 High-Level Model

In this sub-section a high-level implementation of the previously described architecture will be presented. The high-level model was developed in a *MATLAB* ® environment.

It was decided to implement the calibration circuit as function, in *MATLAB* ® (Listing 3.1), in order to facilitate it's incorporation in the developed high-level model of the 2-1 MASH ΣΔM.

```
%%%%%%%%%%%%%% PARAMETERS Description:%%%%%%%%%%%%%%%%%%%
%INPUTS:
% Rm - nominal resistance
% Cm - nominal capacitance
% R - resistance being measured
% C - capicitance being measured
% Vref - reference voltage
% Ts - 1/Fs (Fs - sampling frequency) from the MASH
%OUTPUTS:
% last_m - value of the last measured M
% correctionF - value of the correction factor calculated
% scalar - integer value of the calculated scalar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [last_m, correctionF, scalar] = measureM(Rm,Cm,R,C,Vref,Ts,v_offset)
```

Listing 3.1: Matlab code of the definition of the function MeasureM implemented in order to simulate the behaviour of the calibration circuit with the respective input and output parameters.

The main objective of this model is to simulate the measuring of the RC components, in order to calculate the value of M. To this end an ideal M must be previously defined in order to determine the maximum error associated to every measurement.

Thus, an ideal M (M_i) of 200 was chosen in order to have a measurement error equivalent to less then 1%. By having an ideal M of 200, it allows to have a significant possible range of M variation, that allows to obtain results for $R \cdot C$ process variation. However, to achieve an M of this magnitude a respective scalar needs to be calculated in order to be able to achieve higher desired values of M. It is important to acknowledge that in the future implementation of this circuit this scalar represents the amount of resistors that will be in parallel in order to obtain the necessary resistance. To determine the value of the scalar, the following deduction was implemented, based on the known nominal values of the resistance (R_m) and the capacitance (C_m) used in the second integrator.

$$M_i = \frac{R_i \cdot C_i}{T_S} \Leftrightarrow M_i = 200 \tag{3.47}$$

$$R_i = scalar \cdot R_m \qquad C_i = C_m \tag{3.48}$$

$$R_i = \frac{M_i \cdot T_S}{C_m} \tag{3.49}$$

$$scalar = \frac{R_i}{R_m} \tag{3.50}$$

Here, R_i and C_i represent the ideal values of the RC components. Then, the starting point is defined by the nominal case. As the nominal values of the resistance and capacitance of the second integrator are known, it is possible to calculate the nominal M for this case, as illustrated by the following equations.

$$scalar = 8 \qquad R_m = 2994\Omega \qquad C_m = 4.9952970pF \qquad T_S = 6.25e-10s \tag{3.51}$$

$$M_{Nom} = scalar \cdot \frac{R_m \cdot C_m}{T_S} \approx 191 \tag{3.52}$$

After knowing the result of the nominal case, the SAR sub-binary search algorithm can be implemented with the the increased maximum range of 40%, as illustrated in Fig. 3.28, in order to calculate the value of M.

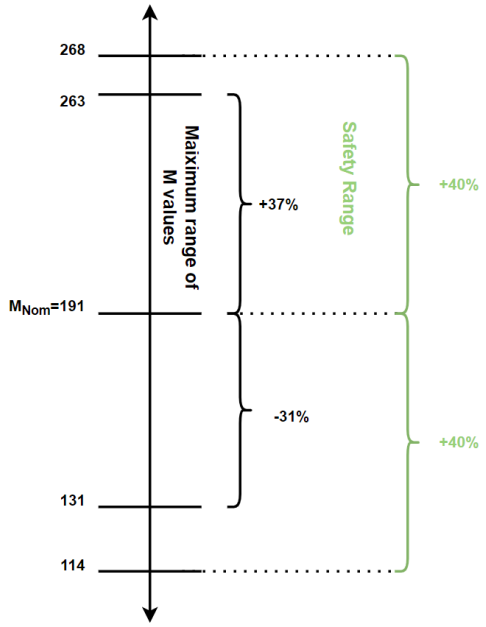


Figure 3.28: Maximum Search range of the sub-binary SAR search algorithm.

The definition of the SAR sub-binary steps can bee seen in Listing 3.2. It is important to point out that the range of the first step is not equivalent to the 40% variation represented in Fig. 3.28, in fact it is approximately half, due to the necessity of introducing

redundant decision steps. This is necessary, in order to be able to cover every possible scenario inside the range of the M values, including the maximum and minimum values of this range.

```
% defining the sub-binary ranges for every predefined step
RC_var=[42 39 29 18 12 8 5 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];

%The number of steps will be equal to the number of cycles in the
% main loop of the calibration cycle
[~, var_size]=size(RC_var);
```

Listing 3.2: Definition of the sub-binary steps.

On the other hand, in order to implement SAR sub-binary search algorithm it is necessary to simulate the comparison between $Vout_{RC}$ with $Vref_K$. For this reason the RC circuit can be analysed as a 1^{st} order RC filter and it's respective transfer function can be obtained (considering it as continuous-time signal) after applying the Laplace transform, resulting in Equation 3.53.

$$H_{RC}(s) = \frac{R \cdot \frac{1}{s \cdot C}}{R + \frac{1}{s \cdot C}} = \frac{R}{R \cdot C \cdot s + 1} = \frac{num(s)}{den(s)} \tag{3.53}$$

This equation is converted into the z-domain, by the application of the bilinear transformation. The implementation of the bilinear function can be seen in Listing 3.3.

```
%define the systems dynamic for the bilinear function
den=[C*R 1];
num=[R];
% defining Z filter coefficients with the bilinear function
%INPUTS:
%   num: numerator of the RC filter Laplace transfer function
%   den: denominator of the RC filter Laplace transfer function
%   Tss: MATLAB 's computation period (1/fss)
%OUTPUTS:
%   znum: numerator in z-domain
%   zden: denominator in z-domain
[znum,zden]=bilinear(num,den,1/Tss);
```

Listing 3.3: Implementation of the bilinear function in matlab.

Then, in order to actually simulate the behaviour of the circuit, the filter function, in *MATLAB* ®, is used. This function computes the values along a given period (T_{ss}) that represents the first clock phase (ϕ_1) and returns a vector, and in order to simulate the sampling of the second clock phase, only the last value of the resulting vector is taken into account. The implementation of the filter function can be seen in Listing 3.4.

```
%INPUTS:
%   znum: numerator in z-domain
%   zden: denominator in z-domain
%   InR+Iref: the current signal at the input of the RC circuit.
```

```
%   Traditionally the filter function always has a 4th parameter which
%   defines initial condition of the capacitor, however in this case
%   the initial condition is always zero as the charge in the capacitor
%   is always reset in each comparison
%OUTPUTS:
%   [vout,iC2]: the computed filtered output of the calibration circuit
%   note: only the last value of vout is 'sampled' (vout(end))
InR=randn(1,np)*sqrt(4*Kb*Tamb/(2*Tss*R));
Iref=ones(1,np)*Vref/R;
[vout,iC2]=filter(znum,zden,InR+Iref);
```

Listing 3.4: Implementation of the filter function in matlab.

Finally, the main loop cycle in which the iterative comparison of $Vout_{RC}$ with $Vref_K$ is made, the RC time constant is calculated and the value of M is obtained, can be seen in Listing 3.5.

```
%calibration cycle parameters:
ncycles=var_size;
meas=zeros(1,ncycles);        %defining the number of measurements
meas(1)=M_Nom;                %starting point is always the nominal case
%defining the vectors needed to save the results:
vout_total=zeros(1,round(ncycles*Tclk/Tss*110));
measure=zeros(1,round(ncycles*Tclk/Tss*110));
%defining the index of the for loop and the offset
index=1;
Voffset=v_offset;
%calibration cycle:
for i=1:(ncycles)
    %number of points:
    np=round((meas(i))*Tclk/Tss);
    %defining the signal at the input of the RC circuit;
    InR=randn(1,np)*sqrt(4*Kb*Tamb/(2*Tss*R));
    Iref=ones(1,np)*Vref/R;
    [vout,iC2]=filter(znum,zden,InR+Iref);
    %comparision between Vout_RC and Vref_K with aditional random noise
    %and offset for realism
    if (vout(end)>(Vref*(1-exp(-1))+randn*Sigma_VnComp+Voffset))
         meas(i+1)=round(meas(i)-RC_var(i));
    else
         meas(i+1)=round(meas(i)+RC_var(i));
    end
    %saving the results of each measurement and incrementing the index
    vout_total(index:index+np-1)=vout;
    measure(index:index+np-1)=meas(i)*ones(1,np);
    index=index+np;
end
```

Listing 3.5: Implementation of the main for loop in charge of simulating the calibration cycle in matlab.

When the algorithm is capable of detecting the last value of M, after the settling period (the 32 steps previously defined, in Listing 3.2), it is possible to calculate the correction factor (Cor_F), as it represents the deviation from the nominal case, which can be translated into the following expression.

$$Cor_F = \frac{MeasuredM}{M_{Nom}} = \frac{R \cdot C}{R_{Nom} \cdot C_{Nom}} \tag{3.54}$$

This final operation is defined in the code, as presented in Listing 3.6.

```
% result of M and calculation of correction factor for alphas:

last_m=meas(end);
alpha_nom= Tclk/(Rideal*Cideal);
alpha_cal=1/last_m;
correctionF=alpha_cal/alpha_nom; %R*C/(R_nom*C_nom)
```

Listing 3.6: Calculating the correction factor in matlab.

In this Listing 3.6 it is possible to see how the correction factor of the α's is calculated and that last measured value of M is saved to later be used. To initially validate the calibration concept and test the calibrator's performance variation , on an ideal level, the correction factor was necessary in order to correct the α and feedback coefficients. This is how the simulation results presented in Section 3.3 were obtained. Later, different M value precisions were tested to ensure the validity of the calibration architecture proposed previously, in section 3.3.1.

However, for final simulations, this function is later incorporated in the High-Level model of the 2-1 MASH $\Sigma\Delta$ Modulator where the value of M is used in the selection of the correct DCL 6-bit LUT and the selection of the necessary reference voltage according to the RC variation, that was previously studied. These selections are both implemented with a series of simple *if* conditions in *MATLAB* ®.

3.3.3 High-Level Simulations

As the implementation of the High-Level calibration model has been presented and analyzed, in this section, different simulations results that support the addition of the calibration circuit to the the 2-1 MASH $\Sigma\Delta$ Modulator will be presented, including the theoretical variation of the calibrator.

Firstly, it is of the upmost importance to verify if, in fact, the algorithm designed to implement the High-Level model of the calibration circuit is capable of calculating the correct correction factor. It is also important to analyse, how much offset can the comparator sustain while performing the calibration. Therefore, a simulation was run, in order to calculate the deviation of the M factor being measured by the model, from the ideal M. As it can be seen in Fig. 3.29, it's possible to analyse the different absolute errors of M measured by the model, for different offsets of the comparator, of the calibration circuit.

Figure 3.29: Variation of the absolute error of M (in percentage), for each respective value of offset in the comparator, when simulating the parameters variation depicted in Table 3.1.

Additionally, it was also decided to analyse the error associated to the calibration of the α parameters, when varying the offset of the comparator, and it's results can be seen in Fig. 3.30. By looking at these figures one can conclude that this function is capable of achieving an absolute error of approximately 5% while measuring M with an offset error of 10 mV and relative error of α of also approximately 5% in the same conditions.

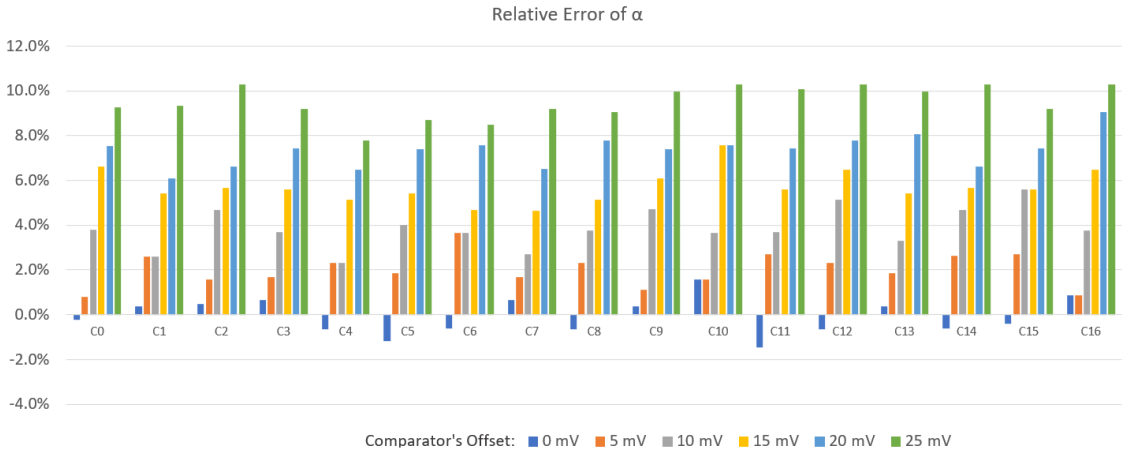


Figure 3.30: Variation of the relative error of α (in percentage), for each respective value of offset in the comparator, when simulating the parameters variation depicted in Table 3.1.

It is important to point out that part of these errors are due to the necessity of having integer values of M. A way to prevent this error would be to calculate the average of the M values obtained in a certain interval of the settling period. This would allow to obtain a more accurate correction factor. However, it's implementation would lead to added circuit complexity, which is preferable to avoid. Then, another important analysis

is the theoretical variation of the performance of the modulator, when different types of calibrations are implemented, as presented in Fig. 3.31.

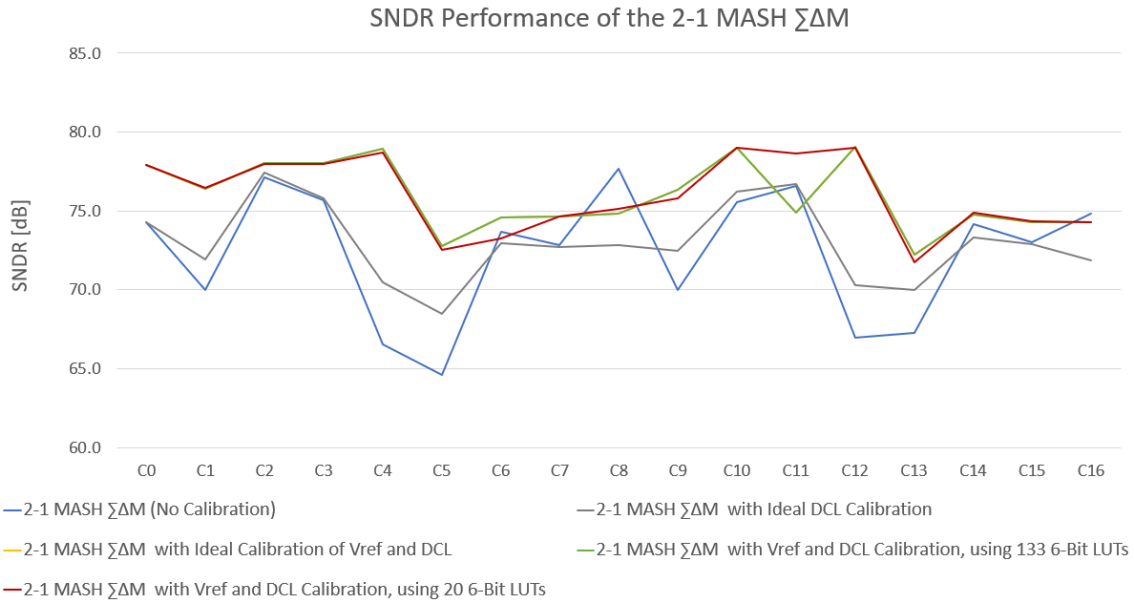


Figure 3.31: Performance of the 2-1 MASH ΣΔM's different implementations, when simulating the parameters variation depicted in Table 3.1, in *MATLAB* ®.

Fig. 3.31 shows the values of the SNDR of the 2-1 MASH ΣΔM with and without DCL and V_{ref} calibrations. By comparing these values it is possible to conclude that the best results are obtained using the previously explained calibration method to adjust both Vref and the DCL coefficients and that limiting the LUTs number to 20 results in acceptable improvements, as stated in Section 3.3.1.

Firstly, there is the scenario which illustrates how the 2-1 MASH ΣΔM operates without calibration for a $V_{in,diff}$ of 0.75 V, in different corners, obtaining a range of SNDR results of [64.6;77.6] dB. One can see that it is the one scenario that has the lowest SNDR values.

Then there is the second scenario which illustrates the ideal calibration of the α's coefficients used in the DCL to reflect the RC variation, still for a $V_{in,diff}$ of 0.75 V. As seen previously in Section 3.3 it enables the calibration to improve the performance of the modulator, in the worst corners with a lower SNDR result, as one can see in Fig. 3.31.

This is followed by two identical scenarios. The first being when the previously described High-Level calibration model, defining α correction factor for the DCL and adjust the reference voltage is used with the MASH 2-1 ΣΔM, for a $V_{in,diff}$ of 1.1 V. Then, in the same exact conditions, instead of using the transfer functions of the digital cancellation logic in order to produce the necessary coefficients to cancel the quantization noise, these are obtained by an equivalent method, that is the selection of the suitable DCL 6-bit LUT selected by the calibration model through out 133 tables. In Fig. 3.31, both scenarios have the exact same SNDR results, which indicates that this operation is

being performed correctly by the High-Level model, for different RC variations.

Lastly, it is possible to see the calibration being performed by the High-Level Model, in the exact same conditions as before, however, now it only requires the 20 DCL 6-bit LUTs instead of the 133, and the range of SNDR results is [71.76; 79.01] dB. In Fig. 3.31, it is now clear to see that, as stated before, the modulator doesn't loose significantly in performance, compared to when resorting to the 133 tables. In order to additionally verify this conclusion, and not base the decision of using just 20 DCL 6-bit LUTs in the results of only 16 process corners, a Monte Carlo simulation of 1000 cases was run in order to certify that within the range of the RC variation this approach was not decreasing the performance of the 2-1 MASH ΣΔM. The results showed that using the 20 DCL LUTs resulted in the maximum SNDR loss of 4.461 dB, however there were also some cases where, by using this approach, the SNDR results would improve 6.023 dB, when comparing to the results of the simulation when resorting to 133 DCL 6-bit LUTs. Thus, it was decided to implement the 20 DCL 6-bit LUTs.

Additionally, a final Monte Carlo Simulation of 10000 cases was run in order to compare the different implementations of calibration. Initially, a baseline is defined with the performance of the original modulator for an amplitude of $V_{in,diff}$ of 0.75 V. The results obtained are depicted in Fig. 3.32. After this a new MC simulation was run still for the amplitude of V_{in} of 0.75 V, only with ideal calibration of α's, using the transfer functions of the DCL. The results obtained are illustrated in Fig. 3.33.

Finally the last MC simulation was run now for the amplitude of $V_{in,diff}$ of 1.1 V, where the calibration is performed by the High-Level model, with the respective necessary alterations of the 2-1 MASH ΣΔM, resorting to the 20 DCL 6-bit LUTs, and the results can be seen in Fig. 3.34.

After analysing all these Monte Carlo simulation results presented in this section one can conclude that overall it's possible to decrease the variability of the results of the 2-1 MASH ΣΔM's performance while also increasing it's performance in the worst corner, when taking advantage of the calibration. Therefore, after finishing this study of theoretical variation of the performance of the ΣΔM and the calibrator, under different conditions, simulated in a *MATLAB* ® environment, it was decided to implement it's schematic.

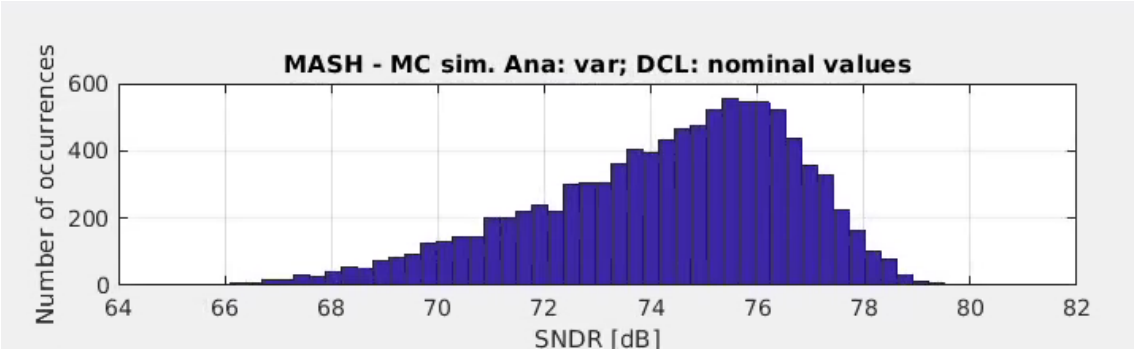


Figure 3.32: Histogram of the variation of the 2-1 MASH ΣΔM performance in the different 10000 cases, for an amplitude of V_{in} of 0.75 V, without calibration, using one nonimal DCL 6-bit LUT.

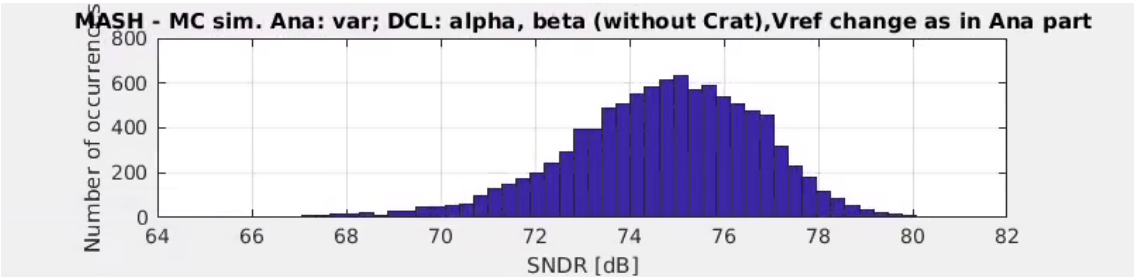


Figure 3.33: Histogram of the variation of the 2-1 MASH ΣΔM performance in the different 10000 cases, for an amplitude of V_{in} of 0.75 V, only with ideal calibration of α's, no alteration to the circuit and using the transfer functions of the DCL.

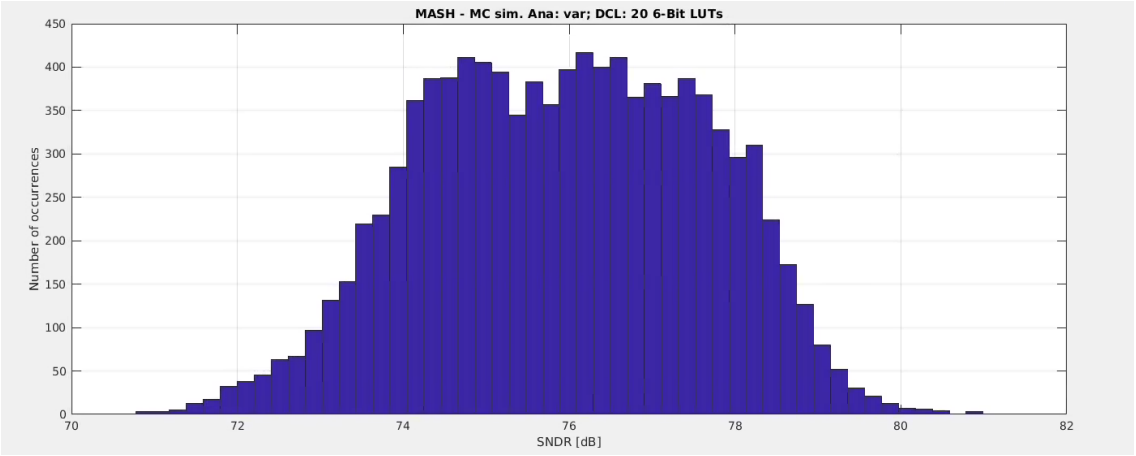


Figure 3.34: Histogram of the variation of the 2-1 MASH ΣΔM performance in the different 10000 cases, for an amplitude of V_{in} of 1.1 V, with the calibration performed by the High-Level model, with the respective necessary alterations of the 2-1 MASH ΣΔM resorting to the 20 DCL 6-bit LUTs and calibration of V_{ref}.

4

Electric Circuits and Simulations

In this chapter, all the electric circuits used for the implementation of the proposed calibration architecture will be presented, and their performance analyzed. As mentioned in Chapter 1, since the publication of the thesis [1], a ΣΔM using the same architecture (2-1 MASH) was designed in UMC CMOS 40 nm Low Power (LP) process technology, using the optimization process based on genetic algorithms [7]. This new design has a 1.1 V power supply voltage, is clocked at a frequency of $F_S = 1.6GHz$ and has a bandwidth of 12 MHz. Therefore, the different blocks that constitute the calibration circuit have been designed in the same technology as the modulator and meet the same design specifications. An ideal digital voltage divider was used to illustrate how V_{ref} could be calibrated. The digital blocks incorporated in this circuit were implemented in *VerilogA*. The schematics presented in this chapter were designed in the *CADENCE* ® Electronic Design Automation (EDA) Explorer software and simulated in *Viruoso*.

4.1 Electrical Circuits

In order to implement the proposed calibration architecture it was necessary to develop both analog and digital circuits. The analog circuits that constitute the calibration architecture are the voltage divider that generates the voltage identified as $Vref_K$, the comparator, the amplifier, the $\frac{RC}{T_S}$ measuring circuit and the ideal digital voltage divider controlled by a digital signal. The digital circuits that constitute the calibration architecture are the SAR sub-binary digital counter and the calibrated digital cancellation logic block, both designed in *VerilogA* to later be converted into *Verilog RTL* and synthesized. All the circuits are sensitive to fabrication process variations, device mismatch and temperature changes that have an impact in the behaviour and operation of these circuits. Thus, all circuits were tested in a temperature range of −40°C to 125°C, a supply and reference voltage range of 1.045 V to 1.155 V and all combinations of slow and fast transistors.

4.1.1 Voltage Divider

The voltage divider is composed of two resistors and a capacitor in parallel, as it can be seen in Fig. 4.1. This circuit is used to produce the necessary $Vref_K$ voltage to later be

compared to $Vout_{RC}$, as previously explained in Section 3.3.1.

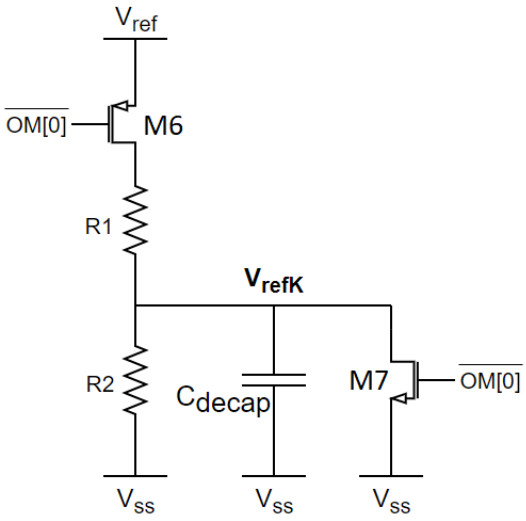


Figure 4.1: Electric schematic of the voltage divider circuit that generates $Vref_K$.

Thus, the following deduction, from Expression 4.1 is used to determine the size of this circuit's resistors components.

$$Vref_K = V_{ref} \cdot \frac{R_2}{R_1 + R_2} = V_{ref} \cdot K = V_{ref} \cdot (1 - e^{-1}) = 0.6953V \tag{4.1}$$

Nevertheless, while testing the implementation of the circuit it was realised that when the comparator analysed both input voltages $Vref_K$ and $V_{out_{RC}}$ it produced kickback noise, and in order to attenuate it, it was necessary to add the capacitor, Cdecap, seen in Fig. 4.1. Moreover, the transistors incorporated in this circuit work as switches. In Fig, 4.1 it's possible to see a PMOS switch and a N-type Metal-Oxide-Semiconductor (NMOS) switch.

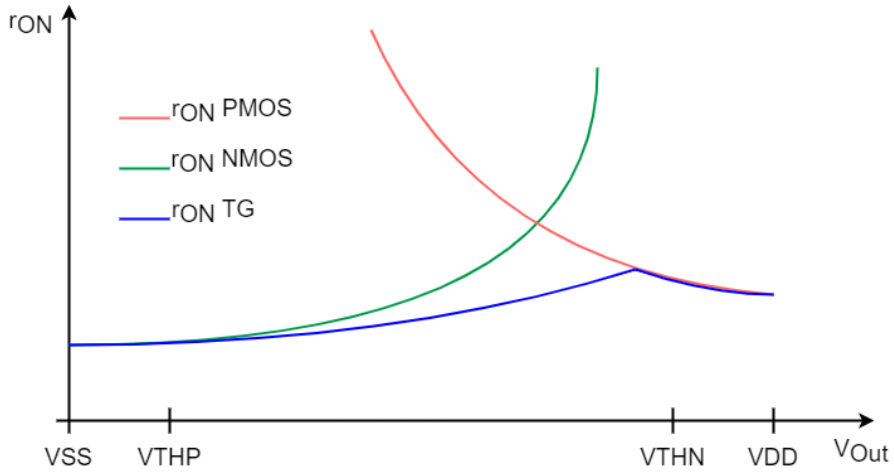


Figure 4.2: Typical resistance characteristic of a PMOS switch, a NMOS switch and a transmission gate switch.

In Fig. 4.2 one can see the resistance characteristic of a PMOS switch, a NMOS switch and a transmission gate switch. Thus, due to the intrinsic resistance of the MOS switches it's possible to conclude that the NMOS switch operates as an active-high switch, and the PMOS switch operates as an active-low switch.

All the switches implemented in the analog circuits are fundamental for the good operation of the calibration circuit. In the voltage divider, the switches are activated by the first bit of the operating mode signal (OM[0]), in order for the voltage $Vref_K$ to settle during the stan-by state, before initiating the calibration (active state).

The size of the transistors and other components of the voltage divider can be seen in Table 4.1.

It was decided to use unitary resistances, as one can see in Table 4.1, in order to facilitate future layout of circuit, and decrease mismatch errors. Voltage $Vref_K$ produced by the voltage divider circuit, once it settles, is of, approximately, 0.6983 V, as Fig. 4.3 shows.

Table 4.1: Size of the transistors and components of the voltage divider circuit.

Devices	**M6**	**M7**		**R2**	**R1**		**Cdecap**
(W/L) [μm/nm]	4/40	4/100	**[[kΩ]**	32.8828	18.4144	**[pF]**	5.031
Nr. Fingers	2	2					
Parallel	1	1	**Series**	50	28	**Parallel**	1

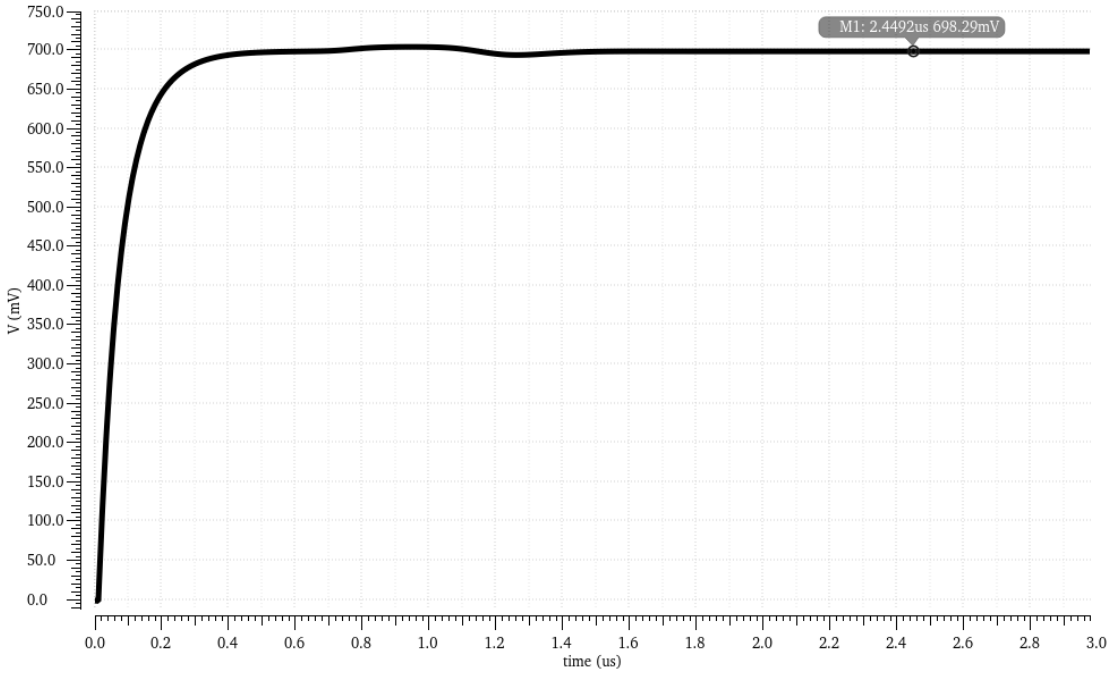


Figure 4.3: Settling of the voltage $Vref_K$, due to the decoupling capacitor Cdecap to the voltage divider circuit, in typical conditions.

4.1.2 Comparator

In order to compare the two voltages $Vref_K$ and $Vout_{RC}$, it was decided to resort to the comparator (Fig. 4.5) used in the implementation of the 2-1 MASH ΣΔ Modulator.

Since the modulator was designed in a 40 nm LP CMOS process, clocked at a frequency of $F_S = 1.6GHz$, it is known that this comparator is capable of comparing signals with a very small amplitude difference, clocked at this high frequency, with a maximum offset of 12 mV. However in order to perform correctly at such high frequencies, it is necessary to implement three different stages of pre-amplifiers, as seen in Fig. 4.4.

These stages are necessary in order to pre-amplify the input signal for the comparator to produce a clear logic output value (V_{DD} or 0 V), at the indicated sampling frequency.

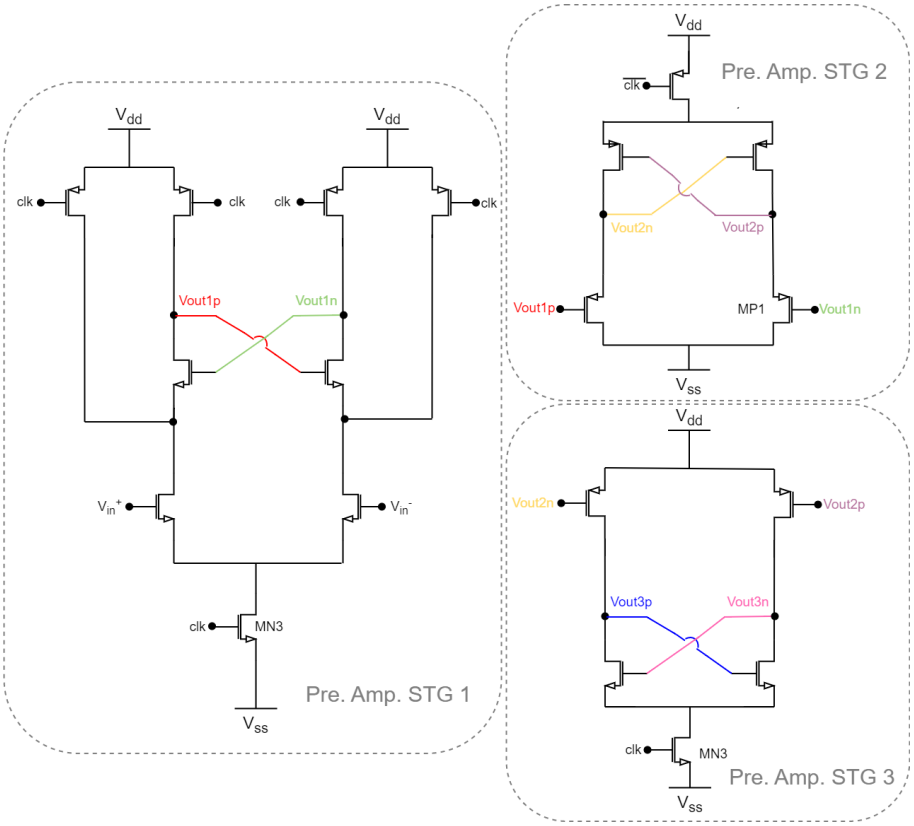


Figure 4.4: Electric schematic of the first three pre-amplifying stages of the comparator.

In Fig. 4.6 it is illustrated how the comparator compares the two voltages in order to produce the output logical signal D_{Comp}.

It should be known that the comparison between the two voltages $Vref_K$ and $Vout_{RC}$ only occurs once the signal $Vref_K$ has settled, during the active state of the OM signal, and that the comparator is triggered by the CLK_{Comp} signal which is generated by the digital counter.

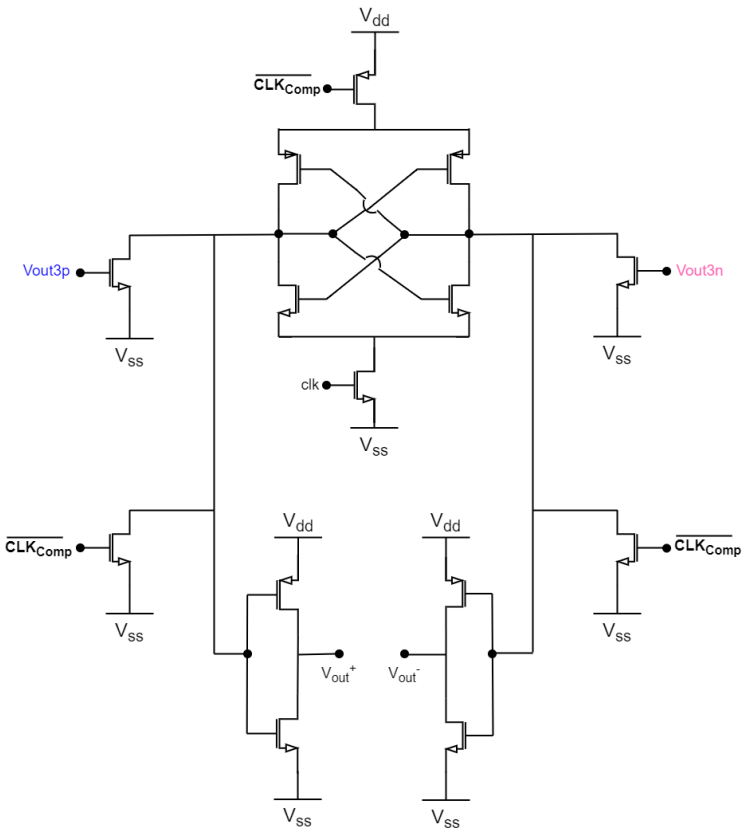


Figure 4.5: Electric schematic of the comparator's final stage.

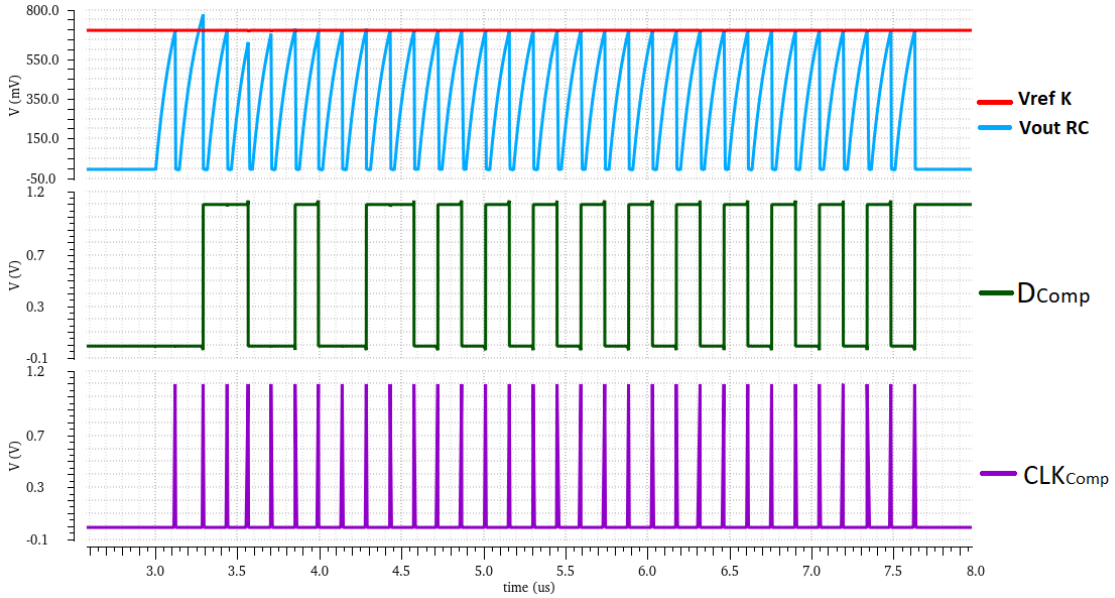


Figure 4.6: Results of simulating the comparison between the two voltages $Vref_K$ and $Vout_{RC}$ and the respective output signal of the comparator, in typical conditions, during active state of OM.

4.1.3 Amplifier

The amplifier is implemented in order to provide a negative feedback function through the added branch of the $\frac{RC}{T_S}$ measuring circuit, which allows for the positive and negative terminals of the OPAMP to be at the same potential, $Vn = Vref_K$, so that the current I_1 through the resistor R_X is:

$$I_1 = Vref_K / R_X \tag{4.2}$$

This allows for the current I_1, to be mirrored to the second branch of the $\frac{RC}{T_S}$ measuring circuit, as it can be seen in Fig. 4.7. The schematic of the amplifier can be seen in Fig. 4.8, with its respective biasing circuit.

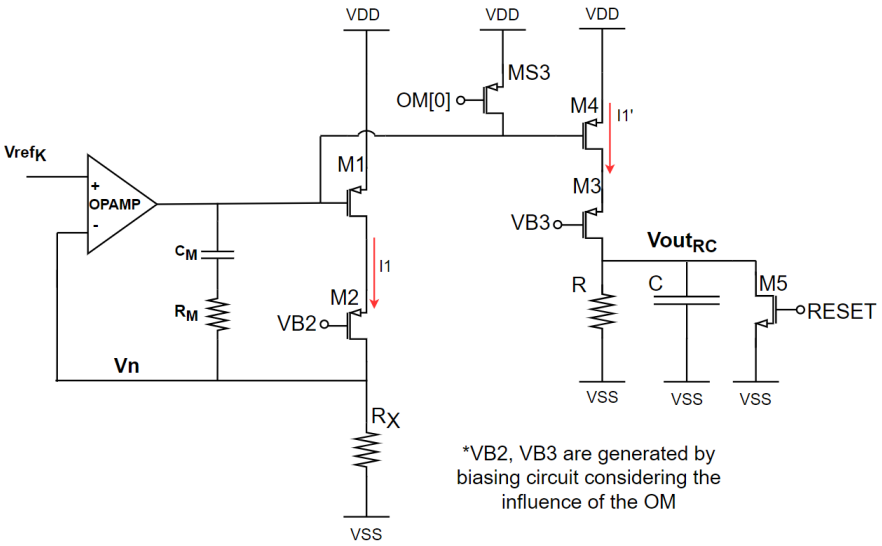


Figure 4.7: Schematic of the amplifier and the two branches of the $\frac{RC}{T_S}$ measuring circuit.

As these are the main requirements for the desired behavior of the amplifier it's only necessary to ensure a 40 dB DC gain and a stable phase margin, thus a single-ended one stage differential cascode pair amplifier was implemented, as it can be seen in Fig. 4.8. In order to analyze this amplifier, its small signal model was derived and analyzed, resulting in the following gain equation:

$$A_v = \frac{vout}{vin} \approx -gm_{N1} \cdot (rds_{P1} \| rds_{N1}) \tag{4.3}$$

The MOS transistor's transconductance, gm, is defined as:

$$gm = \frac{2 \cdot I_D}{V_{GS} - V_{th}} \tag{4.4}$$

Where the V_{th} is the threshold voltage of the transistor and V_{GS} is the equivalent to the voltage drop from the Gate terminal to the Source terminal of a transistor. Additionally, the transistor MN2 operates as a current source for the differential cascode pair, setting double the current of each side at Direct Current (DC). The current of a MOS transistor in DC is given by the following equation:

$$I_D = \frac{1}{2} \cdot \frac{W}{L} \cdot \mu_{n,p} C_{ox} \cdot (V_{GS} - V_{th})^2 \tag{4.5}$$

Here, the parameters $\mu_{n,p}$ and C_{ox} are technology parameters. The current of MN2 is provided through the current mirror circuit, which is implemented with the ratio 2:1, as the current from the Bias circuit is generated by the voltage supply and the resistor R1. Moreover, the transistors MS1, MS2 and MS3 function as MOS switches, in order to guarantee that this circuit only operates when the calibration is enabled. The size for this amplifier and its respective biasing circuit can be seen in Tables 4.2 and 4.3.

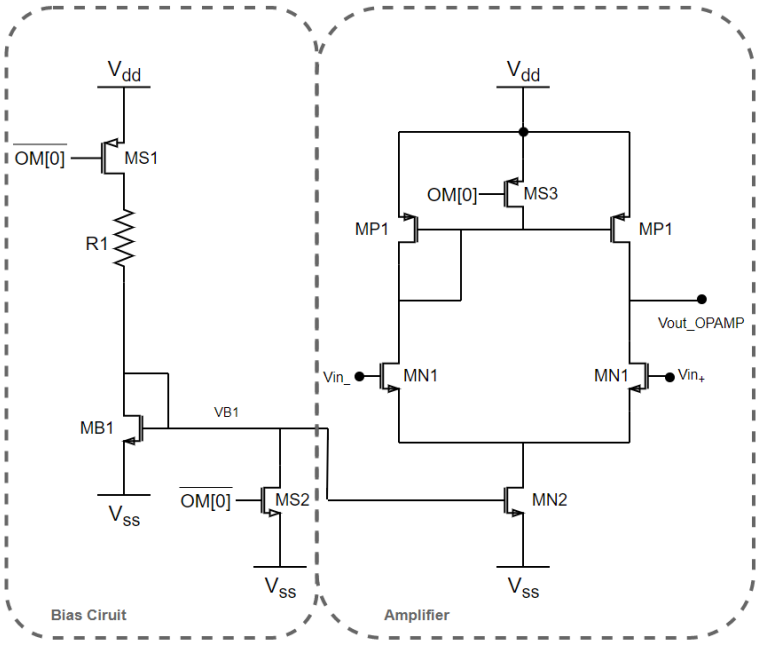


Figure 4.8: Electric schematic of the amplifier and it's respective biasing circuit.

Table 4.2: Size of the transistors of the amplifier circuit.

Devices	**MP1**	**MN1**	**MN2**	**MS3**
(W/L) [μm/μm]	10/8	13/15	8/4	4/0.04
Nr. Fingers	2	2	4	2
Parallel	1	1	1	1

Table 4.3: Size of the transistors and components of the Bias circuit.

Devices	**MS1**	**MS2**	**MB1**		**R1**
(W/L) [μm/μm]	4/0.04	8/0.1	8/4	**[[kΩ]**	153.6
Nr. Fingers	2	2	4		
Parallel	1	1	2	**Series**	32

It is important to point out that the sizes of the transistors were increased in order to decrease the possibility of mismatch contributions errors. A possible alternative for the implementation of this amplifier could be a current-mirror operational transconductance

amplifier (OTA) which could lead to less added area and power consumption. In Fig. 4.9 and 4.10 is possible to see the resulting bode diagrams of the amplifier and the initial requirements being satisfied in the nominal conditions.

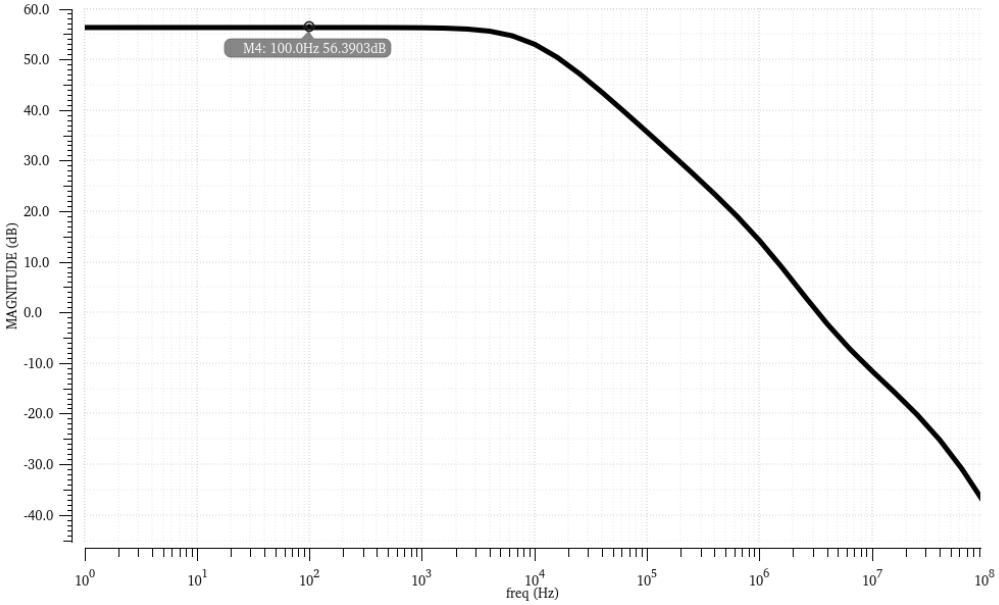


Figure 4.9: Bode diagram of the gain of the amplifier at typical device speed, a supply voltage of 1.1 V and 27°C.

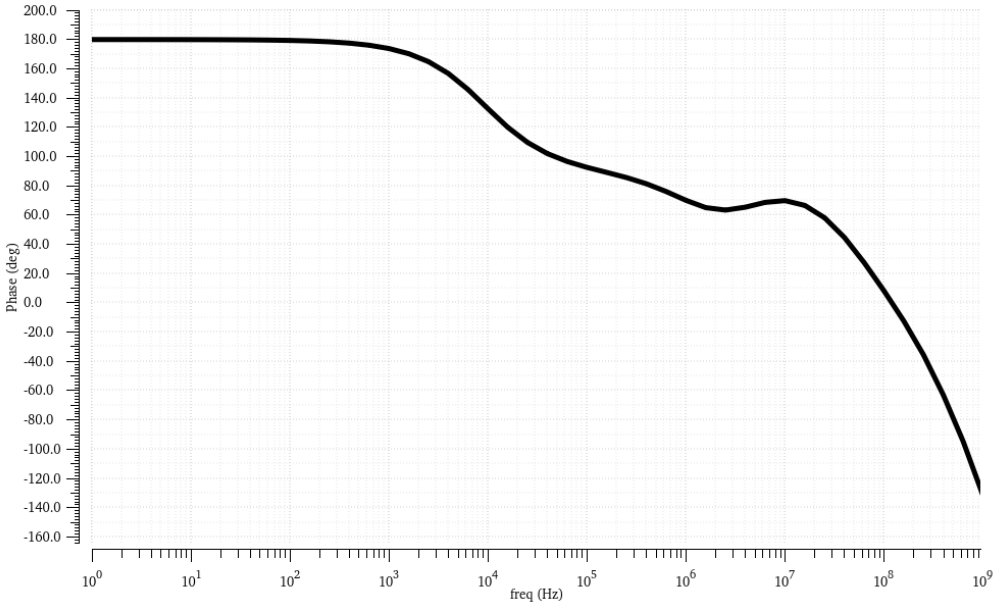


Figure 4.10: Bode diagram of the phase margin of the amplifier at typical device speed, a supply voltage of 1.1 V and 27°C.

However, in order to guarantee a well working amplifier and secure the correct performance of the calibration circuit, this circuit is simulated under different corners, corresponding to all the different variations that the calibration circuit will undergo.

Therefore the variation being tested for the amplifier is of the corresponding high,

medium and low temperatures, different PMOS and NMOS device speed and the corresponding high, medium and low supply voltages. The results of this corners simulation can be seen in Table 4.4, and show that the amplifier has a DC gain > 52 dB and is stable in process, voltage and temperature (PVT) corners. Additionally, it's possible to see the bode diagrams of the amplifier, showing the gain and phase variation through the different PVT corners, in Figs. 4.11 and 4.12.

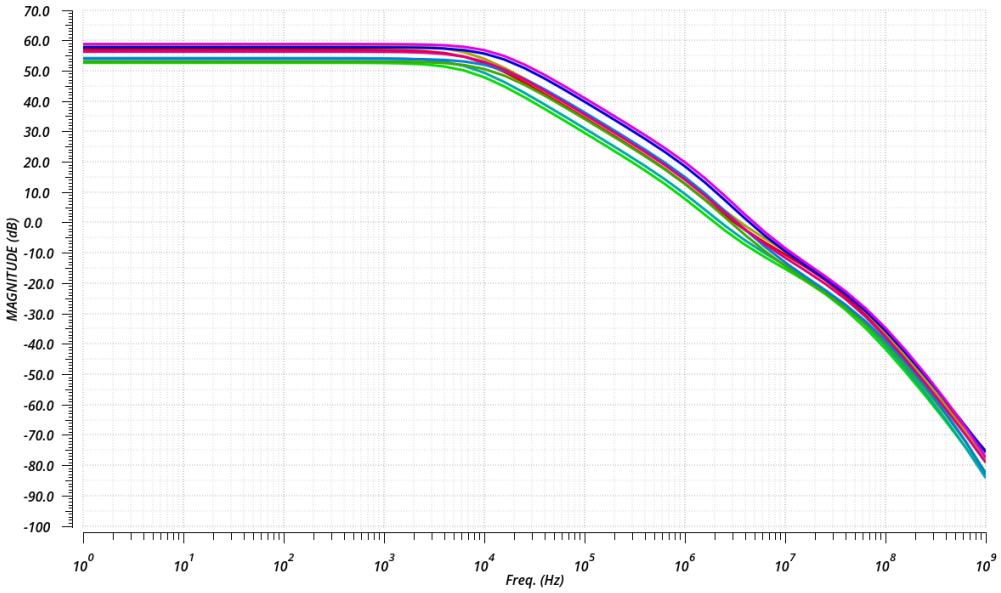


Figure 4.11: Bode diagram of the gain variation of the amplifier, for different PVT conditions, from Table 4.4.

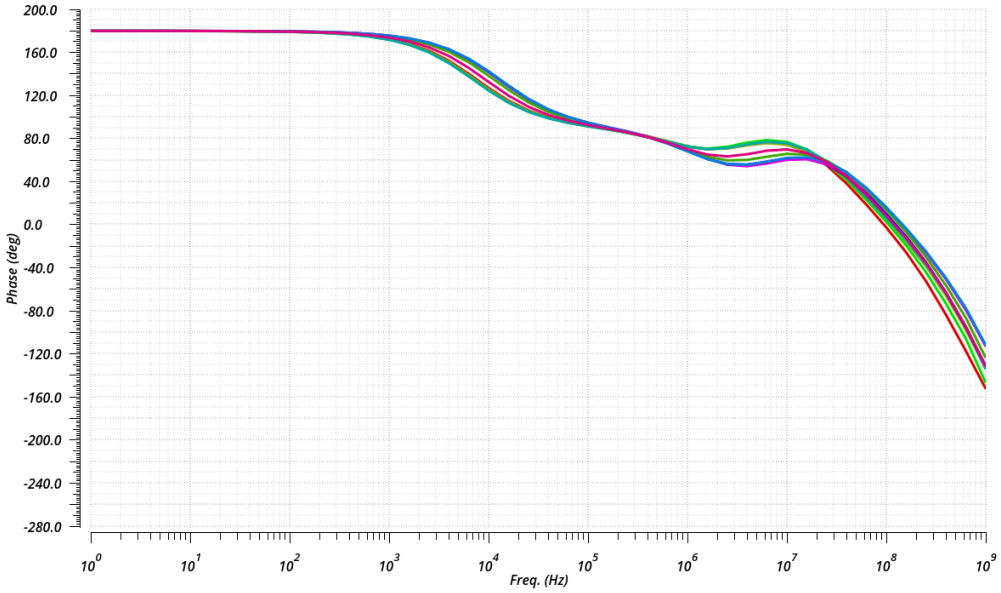


Figure 4.12: Bode diagram of the phase variation of the amplifier, for different PVT conditions, from Table 4.4.

It is also interesting to test if the input terminals of the OPAMP stay at the same potential, and thus, the input signal of the terminal Vin_+ is a fixed value, while the input

signal of the terminal $Vin\_$ ($Vin\_ = V_n$) is provided by the feedback function though the added branch of the RC measuring circuit, and the results from Table 4.4 show that this is achieved.

Table 4.4: Results for supply voltage, temperature and device speed corners simulation of the amplifier's performance.

Transistors Speed	**tt**	**ff**	**ff**	**ff**	**ff**	**ss**	**ss**	**ss**	**ss**
Temperature [°C]	**27**	**-40**	**125**	**-40**	**125**	**-40**	**125**	**-40**	**125**
Supply Voltage [V]	**1.1**	**1.045**	**1.045**	**1.155**	**1.155**	**1.045**	**1.045**	**1-155**	**1.155**
Gain [dB]	56.39	57.83	53.16	58.91	54.06	57	52.74	58.63	54.3
PM [°]	64.53	56.47	60.11	55.41	56.15	73.39	71.32	73.33	70.97
Vin_+ **[mV]**	700.0	700.0	700.0	700.0	700.0	700.0	700.0	700.0	700.0
$Vin\_$ **[mV]**	699.9	700.1	700.5	700.2	700.7	699.6	699.5	699.8	699.8

4.1.4 $\frac{RC}{T_S}$ Measuring Circuit

The $\frac{RC}{T_S}$ measuring circuit is the main analog block of the calibration circuit, as it allows to measure the $R \cdot C$ variation. In order to facilitate the visualization of the operation of this circuit, both electric schematic and the block diagram of this circuit can be seen, respectively, in Figs. 4.14 and 4.13.

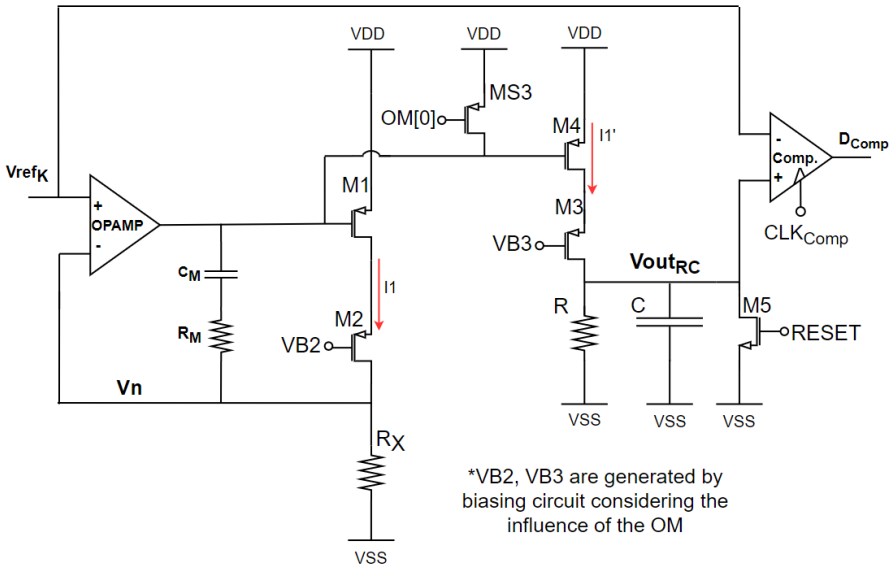


Figure 4.13: Block diagram of the simplified $\frac{RC}{T_S}$ measuring circuit with the comparator and amplifier.

In order to implement this circuit, it is essential to use the amplifier previously described in Section 4.1.3. Therefore, in Fig. 4.14 it's possible to see the amplifier with it's respective biasing circuit, the first branch in the RC measuring circuit with a compensation bias for the cascode transistor, M_2, and the *Miller Compensation* in order to stabilize the output signal of the amplifier.

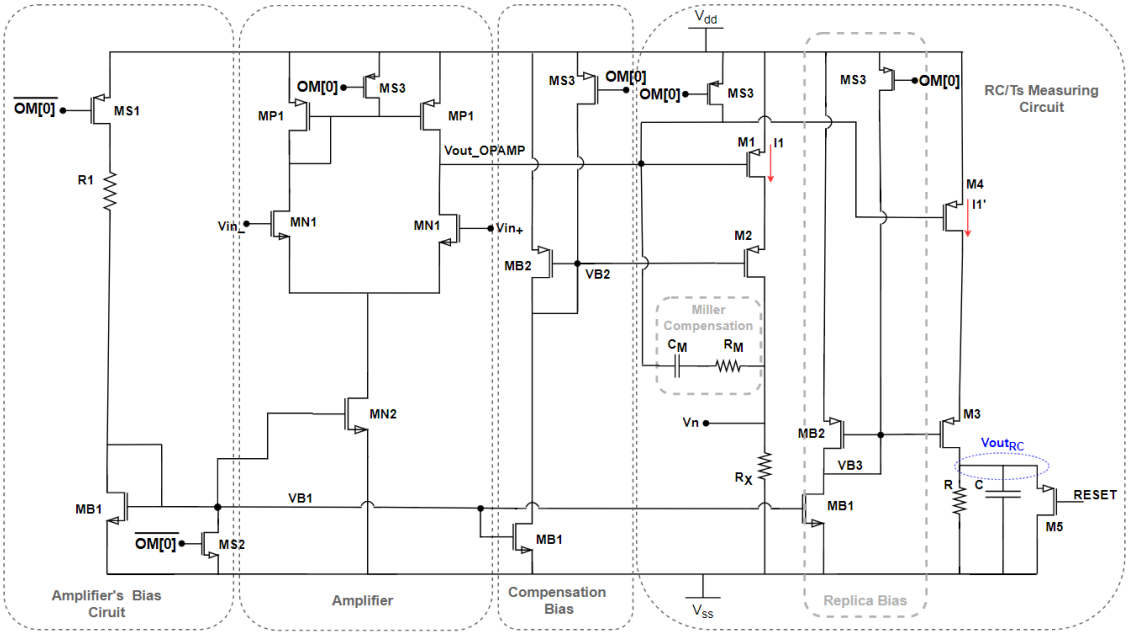


Figure 4.14: Electric schematic of the $\frac{RC}{T_S}$ measuring circuit, with the amplifier and their respective biasing circuits.

Then there's the second branch of the $\frac{RC}{T_S}$ measuring circuit, with a replica of the compensation bias for the tansistor M_3 (as the transistors from both branches have the same size). The cascoded transistor, M_2 and M_3, allow to use signals through the MOS switches (transistors MS), to better control the behaviour of the circuit, through the OM signal, and allow for a better flow of the current through both branches of this circuit.

As previously stated, in Section 4.1.3, the negative feedback causes the positive and negative inputs of the amplifier to have almost the same voltage. This makes V_n equal to $Vref_K$, which allows for the current I1 to be defined as $I_1 = Vref_K/R_X$, and this current is then mirrored by the current mirror, which is implemented with M_4 with a factor of 1:1, to the second branch of the circuit. Since $Vref_K$ was defined previously as:

$$Vref_K = I_1 \cdot R \cdot K \tag{4.6}$$

The resistor R_X, has to be defined as:

$$R_X = K \cdot R \qquad K = 1 - e^{-1} \tag{4.7}$$

In order for the behaviour of this circuit to be uniform with the theoretical analyses presented in Section 3.3.1, the resistor R needs to have 16 times the size of the resistors R_2 from the second integrator, since M_{Meas} was defined previously in the theoretical analysis of this circuit, in section 3.3.1, as:

$$M_{Meas} = scalar \cdot \frac{R \cdot C}{Tclk} \qquad scalar = 8 \qquad R = 2 \cdot R_2 \tag{4.8}$$

$$M_{Meas} = \frac{16 \cdot R_2 \cdot C_2}{T_S} \tag{4.9}$$

Therefore, R and R_X, depicted in Fig. 4.14, are built of the same unitary resistors as R_2 from the second integrator of the ΣΔM, and C is built as a real capacitor with the size of the capacitor used in the second integrator, C_2. However, simulations showed that the parasitics added to the $\frac{RC}{T_S}$ measuring circuit delayed the response of the $Vout_{RC}$ signal in 5% (which, as a consequence, added an offset to the final value of the measured M). To compensate for this, the capacitor used in the circuit is 5% smaller then C_2. Moreover, the transistor M_5 is necessary in order to reset the charge in the capacitor, C, and, as a consequence, reset $Vout_{RC}$ signal, when necessary. The size of the components of this circuit can be seen in Tables. 4.5 and 4.6.

Table 4.5: Size of the lumped elements of the $\frac{RC}{T_S}$ measuring circuit.

Devices	**RM**	**RX**	**R**		**CM**	**C**
[kΩ]	8.982	14.97	23.9521	**[pF]**	0.461	4.7452
Nr. of Elements in Series	6	10	16	**Nr. of Elements in Parallel**	1	258

Table 4.6: Size of the transistors of the $\frac{RC}{T_S}$ measuring circuit and the respective compensation biasing circuit.

Devices	**MB1**	**MB2**	**MS3**	**M1=M4**	**M2=M3**	**M5**
(W/L) [μm/μm]	8/4	6/6	4/0.04	64/2	60/0.2	8/0.04
Nr. Fingers	4	2	2	16	20	2
Parallel	2	1	1	2	1	5

One should know that the size of the devices that compose the amplifier and its respective biasing circuit, remain the same, as described in section 4.1.3.

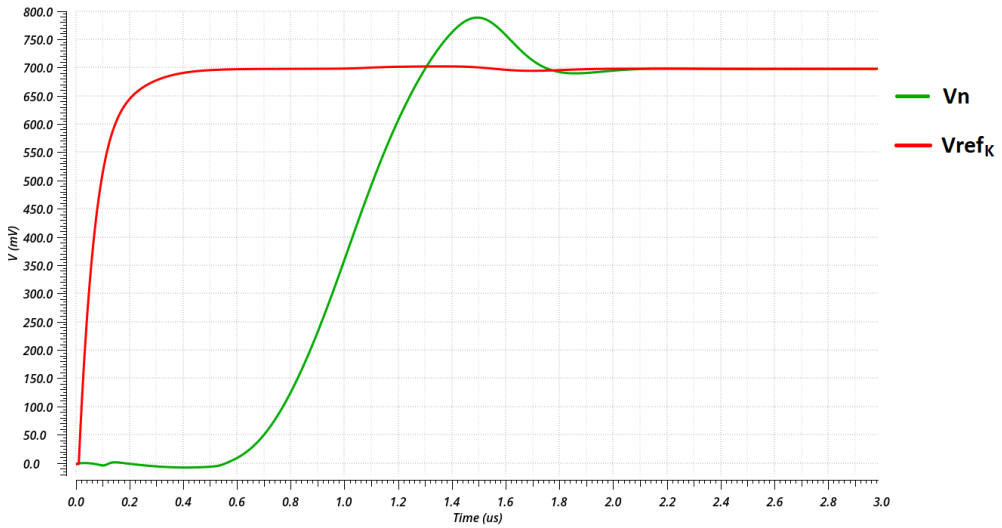


Figure 4.15: Simulation of the two voltages Vn and $Vref_K$, at typical device speed and 27°C, with supply voltage of 1.1 V (typical conditions).

In Fig. 4.15 it's possible to see, that once both signals, Vn and $Vref_K$, have settled both have the same voltage, verifying that indeed the positive and negative terminals are at the same voltage. Moreover, in Fig. 4.16, it is illustrated how the circuit performs in nominal conditions, once $Vref_K$ has settled. In this Figure, it's clear to see the effect of the reset switch, and how the circuit's performance is completely dependent on the operations defined in the SAR digital counter block.

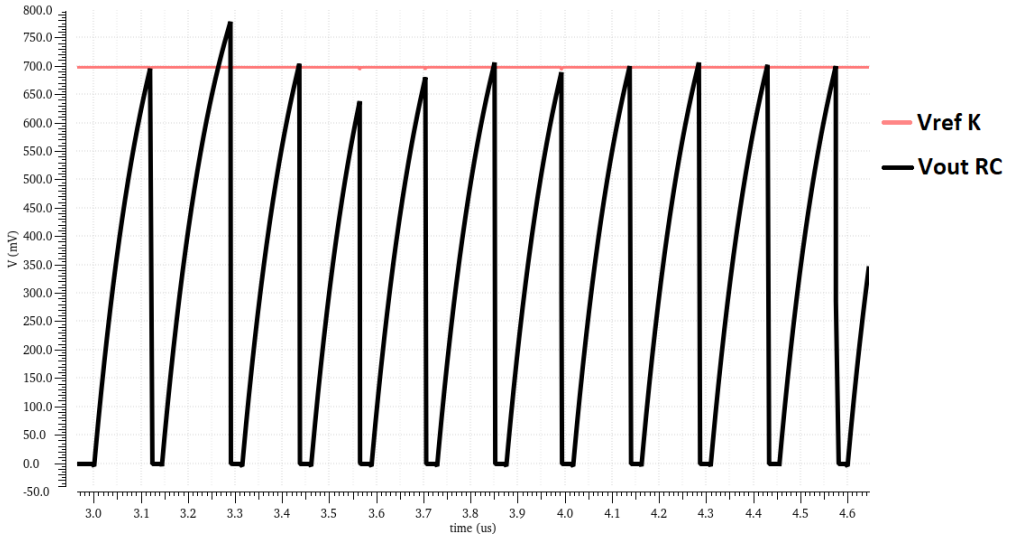


Figure 4.16: Simulation of the two voltages $Vout_{RC}$ and $Vref_K$, in typical conditions.

4.1.5 SAR Sub-Binary Digital Counter

Now that all the analog blocks that are necessary for the measurement of M, have been described, it becomes easier to understand exactly how these are controlled by the OM signal and the outputs produced by the digital SAR Sub-Binary Counter. This is a digital block implemented in VerilogA with the purpose of measuring the $R \cdot C$ variation and counting the number of clock cycles equivalent to the RC time constant. The VerilogA code of this digital block is presented in annex II.1.

As previously mentioned in Section 3.3.1 the OM signal was designed in order enable all the signals, necessary for the correct measurement of M, to settle before the active state begins. Thus, the need for stand-by state in order for the voltages $Vref_K$, Vn, $Vout_{OPAMP}$, V_{B_2} and V_{B_3} to settle before the $\frac{RC}{T_S}$ measuring circuit starts to operate, so not to cause any kind of offset to the M value. Therefore, the algorithm designed for the operation of the digital SAR sub-binary counter is completely dependent on this signal. Moreover, as the first bit of the OM signal triggers all the analog blocks, the second bit is only necessary to alert the digital counter that enough time has passed, and these signals have already settled, which means that the process of measuring the RC time constant can begin, and the $\frac{RC}{T_S}$ measuring circuit can operate.

As it can be seen in Fig. 4.17, this block has as inputs, the CLK signal, the ouptut signal of the comparator, D_{Comp}, the enable signal RESETN and the OM signal. And produces

the final M[8:0] signal of 9 bits, the RESET signal which discharges the capacitor in order to reset $Vout_{RC}$, the $Vref_{CAL\_code}$ signal of 8 bits which will calibrate the reference voltage through the digital resistive voltage divider circuit, the CLK_{Comp} signal which triggers the comparator , and finally, the CAL_DONE signal which indicates when the calibration is finished.

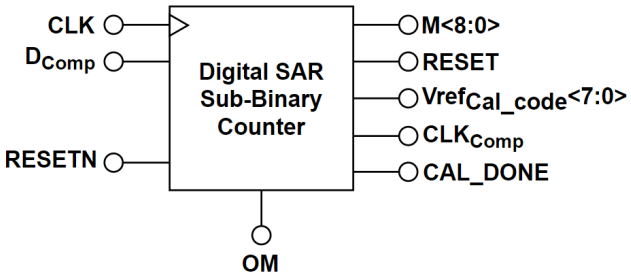


Figure 4.17: Digital block of the SAR sub-binary counter.

A flow chart presenting the algorithm implemented in VerilogA can be seen in Fig.4.18, as it facilitates the better comprehension of how the calibration was implemented through the digital sub-binary counter. As illustrated by Fig. 4.18, besides the I/O signals of the digital block, previously described, the algorithm also contains internal signals.

One should know that there are initial conditions set for the variables/internal signals of this block, for when the circuit is not operating or it is reset, these are defined as the initial step (Fig. 4.18). The conditions are the following:

- **Counter=0** - This variable is in charge of counting the number of clock cycles, when it is necessary.
- **step_count=0** - This variable is necessary to count the number of steps taken in order to measure M.
- **Num_steps=32** - Contains the number of predefined steps that are necessary to measure M (the variation of M for every step is the same as in Listing. 3.2).
- **state=0** - Variable that is used to know which is the current state.
- **state_next=0** - Variable that is used to define which is the following state.
- **latch_en=0** - Internal signal used to trigger the CLK_{Comp} signal.
- **latch_en_next=0** - Internal signal used to define the latch_en signal.
- **M=M_nom** - M is the variable used to calculate the value of M in every step, and it can either be incremented or decremented according to the signal of the comparator. M_Nom is the value of the nominal M (191).

- **last_m=M_nom** - Variable used to save the last measured M, after the settling period, and used to generate the M[8:0] digital signal. When the calibration circuit is not operating, it will still provide the nominal M code, in order for the new DCL block to still operate under nominal DCL conditions.

- **cal_busy=0** - Signal that indicates whether or not the circuit is performing a calibration.

- **rst=1** - Internal signal that triggers the RESET signal to discharge the capacitor of the RC measuring circuit. The switch in parallel with the capacitor is active-high and therefore, when the circuit is starting to operate, rst=1 and $Vout_{RC} = 0V$.

Other important variables are:

- **done** - Internal signal that triggers the signal CAL_DONE.

- **active** -Internal signal which indicates that the state Active has been reached and the second bit of the OM signal equals 1.

- **RC_Var[]** - Vector that contains the 32 steps with the previously studied $R \cdot C$ variation converted into M step units.

- **Meas[]** - Vector that is used to save the different 32 Ms measured in one calibration.

- **Reseting_cap** - Variable that defines the duration (number of clock cycles) of reset of the capacitor, in order for $Vout_{RC}$ to reach 0 V.

Now that all variables and different signals are defined, it is easier to understand how the algorithm, illustrated in Fig. 4.18, works.

It is important to point out that when the calibration circuit is enabled, the digital counter is working under unknown conditions. This is why the OM signal always starts at the off state (OM[0]=OM[1]=0) and has an initial step with predefined conditions and internal signals. This state can also be referred to as a "waking-up"state. During this state, the counter is stuck in this initial loop in order to define an initial M[8:0] output, to adjust the reference voltage and DCL configuration for the operation of the MASH 2-1 $\Sigma\Delta$M. Since, in this description, a calibration cycle hasn't pass, the only known conditions are the ones defined on the initial step, and therefore the V_{ref} and DCL will be adjusted (initially) to nominal conditions (since cal_busy=0 - defined in initial step).

Then if OM[0] is triggered, the **step_count** variable is checked in order to see if the current number of steps taken is 32 or not. If it is 32, then this means that a calibration has been performed, the circuit has finished it's operation, however it's still in active mode. In order to start a new calibration, first the second bit of OM signal has to be 0 (and as a consequence the first one too - because the signal is active-high) in order for the **step_count** to be reset and the circuit to be uniform with the OM and both be in the Off

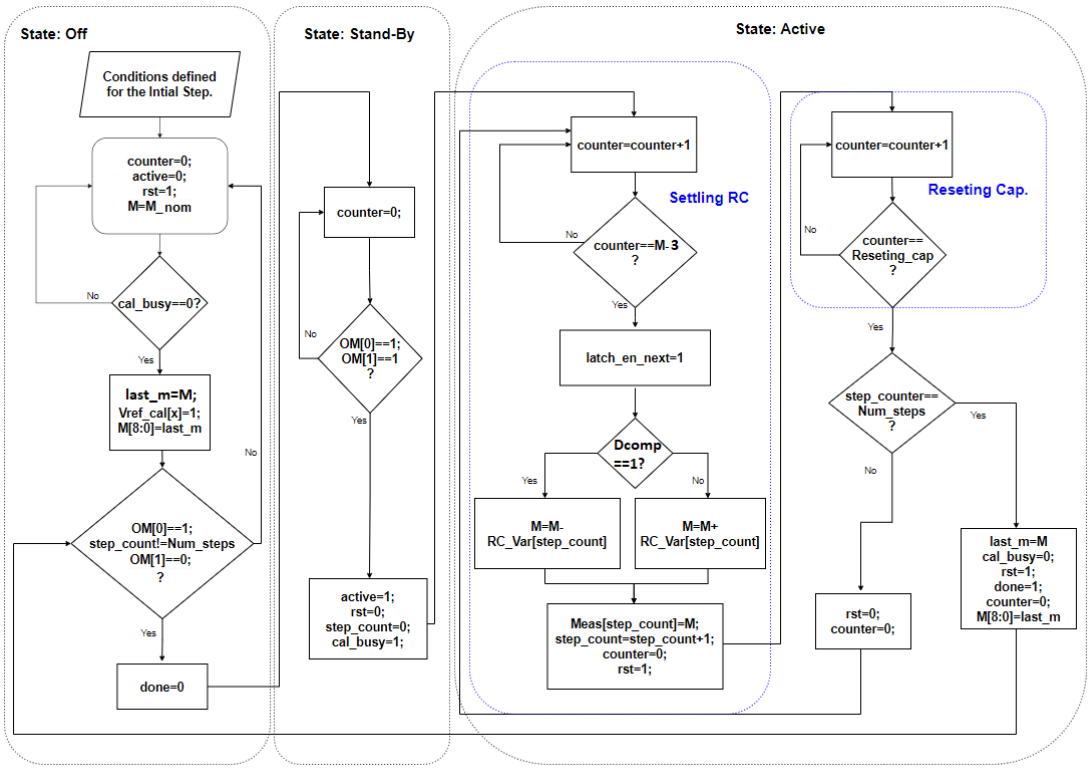


Figure 4.18: Flow chart of the algorithm implemented in VerilogA for the operation of the SAR sub-binary counter.

state. If it isn't 32 then this means that the circuit is ready to start operating and therefore the internal signal done has to be equal to zero.

It should be known that while the circuit is operating, a cross event is set where the variable **count** is being incremented, the variable **state** is being replaced by **state_next** and the variable **latch_en** is also being replaced by **latch_en_next**, at the falling edge of the clock signal.

Moreover, when OM[0] signal is triggered, the state stand-by has been reached (meanwhile, all the switches, from the analog block previously analyzed, have been turned on in order to allow all the relevant signals to settle) and the **count** variable is reset to zero, in order not to cause an offset to the M value. And it stays in this state the OM[1] signal is triggered. Once this event happens the variable **active**=1 in order to indicate that the state active has been reached, **cal_busy** equals 1 and the **step_count** and variable is reset in order to start the calibration, and **rst**=0, in order to open the NMOS switch (transistor M_5 of the $\frac{RC}{T_S}$ measuring circuit) to let $Vout_{RC}$ grow while the clock cycles are being counted.

Then the cycle of settling the $R \cdot C$ variation has been entered, where the clock cycles keep being counted until $N_{clk\_cycles} = M - 3$, which indicates that it's time to trigger the comparator, to account with the kickback noise from the comparator, which causes an offset in $Vout_{RC}$. Since in the next two clock cycles the number of clock cycles counted

will be equal to the last measured M (when this happens for the first time, $N_{clk\_cycles} = M = 191$), the signal **latch_en_next** is triggered, so that in the next two clock cycles, once the last measured value of M is reached, the comparator already compared $Vref_K$ with $Vout_{RC}$. According to the decision of the comparator, the new M value is either incremented or decremented and the it's saved in the vector **Meas[]**. Now, once the M value has been saved, the **step_count** variable is incremented and the **count** variable is reset and the **rst** signal is triggered in order to close the NMOS switch to reset the $Vout_{RC}$ signal.

The MOS switch will remain closed for a certain number of clock cycles, and once it reaches **Reseting_cap** clock cycles, it checks the number of steps (**step_count**) that have been counted. If **step_count=Num_steps=32**, then all the steps have been taken and the **last_m** will be equal to the last measured M value, the circuit has finished the calibration (**Cal_ready**=1 and **done**=1), the $Vout_{RC}$ signal is reset through **rst**=1, the variable **count** is also reset, and the circuit will go to the off state. If **step_count** is not equal to 32, then the calibration hasn't finish and the circuit goes back to the cycle of settling the RC variation.

4.1.6 Digital Resistive Voltage Divider

The digital resistive voltage divider circuit is implemented in order to provide a way to calibrate the reference voltage of the 2-1 MASH ΣΔM and validate the calibration concept being presented in this thesis. Therefore this circuit is implemented with ideal components and its respective schematic can be seen in Fig. 4.19.

As one can see, this circuit is implemented with ideal switches in order to produce a well defined reference voltage with no added distortion or delay.

Furthermore, to guarantee that the output signal (V_{ref_Cal}) has a driving strength, an ideal buffer is used in order to provide additional current without compromising the signal, as illustrated in Fig. 4.19.

Table 4.7: Size of the components that compose the digital resistive voltage divider circuit.

Devices	**R**	**R8**
[kΩ]	6.5765	37.4864
Series	10	57

The resistors that compose this circuit are unitary resistors and the size of the components of this circuit can be seen in Tab. 4.7. It is important to point out that the resistors R1, R2, R3, R4, R5, R6 and R7 all have the same size ($R = R1 = R2 = R3 = R4 = R5 = R6 = R7$).

With this implementation, it was possible to obtain the results presented on Tab. 4.8 for each voltage node. Additionally, in Fig. 4.20, one can see the circuit's performance when testing for the different necessary voltages of Vref.

Table 4.8: Results of the simulations for the range of values of Vref for every code, in a 3-Bit Logic and a One-Hot Logic.

Range of M	V	[V]	3-Bit Logic	One-Hot Logic	One-Hot bit
[131,147]	V_0	1.1	000	0000 0001	$Vref_{cal}(0)$
[148,164]	V_1	1.013	001	0000 0010	$Vref_{cal}(1)$
[165;181]	V_2	0.9267	010	0000 0100	$Vref_{cal}(2)$
[182;198]	V_3	0.8401	011	0000 1000	$Vref_{cal}(3)$
[199;215]	V_4	0.7535	100	0001 0000	$Vref_{cal}(4)$
[216;232]	V_5	0.6669	101	0010 0000	$Vref_{cal}(5)$
[233;249]	V_6	0.5803	110	0100 0000	$Vref_{cal}(6)$
[250;266]	V_7	0.4937	111	1000 0000	$Vref_{cal}(7)$

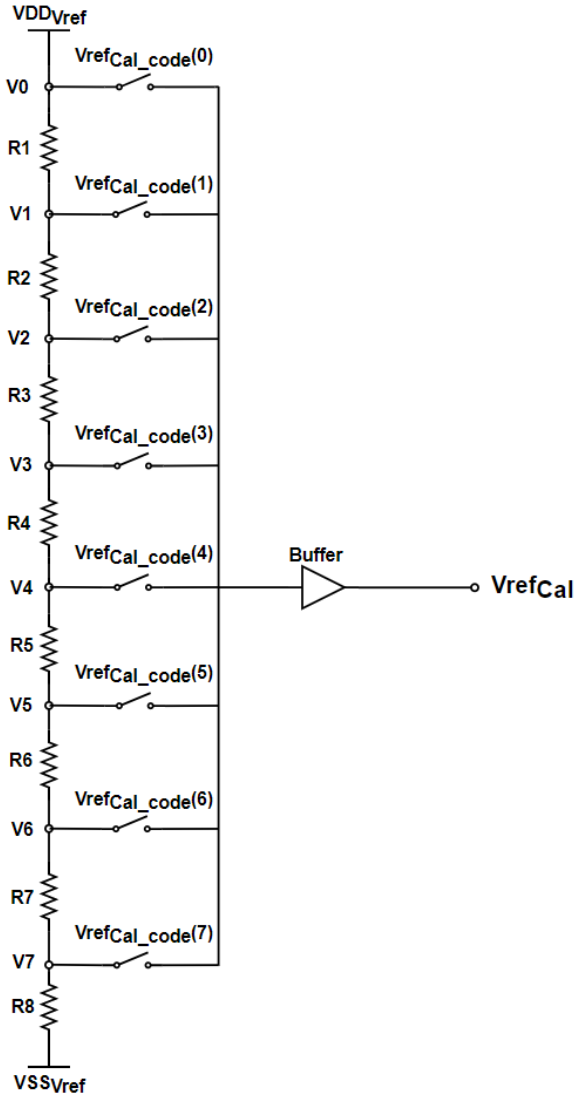


Figure 4.19: Electric schematic of the ideal digital resistive voltage divider circuit.

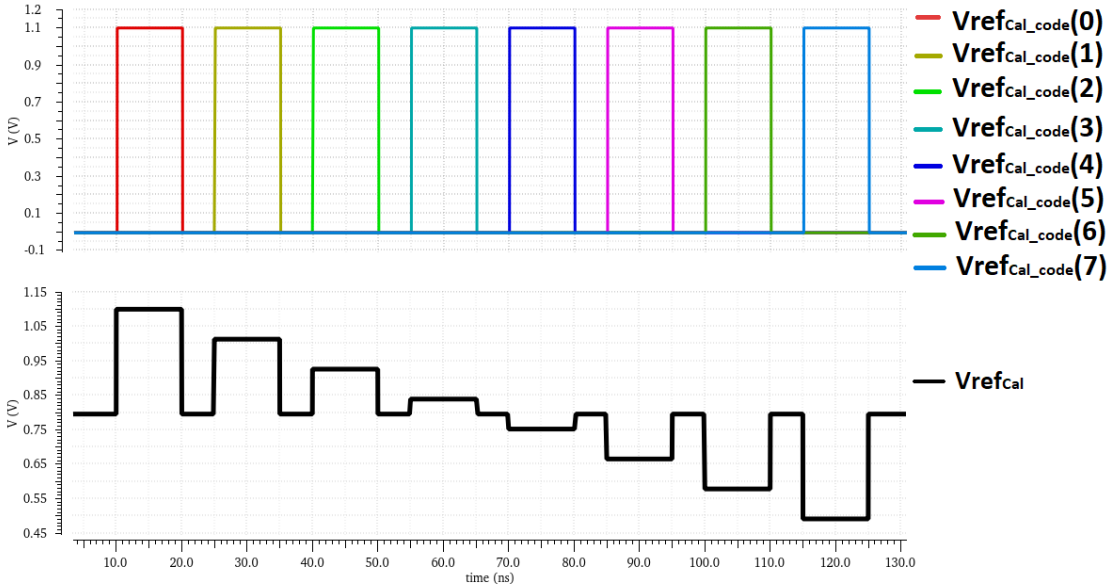


Figure 4.20: Output of the resistive voltage divider when testing the control signal $V_{refCal_{Code}}$.

4.1.7 Digital Cancellation Logic

The new calibrated Digital Cancellation Logic block (Fig. 4.21) was implemented in VerilogA, having the different 20 DCL 6-bit LUT registered in order to facilitate it's access. The VerilogA code implemented for this digital block is presented in annex II.2.

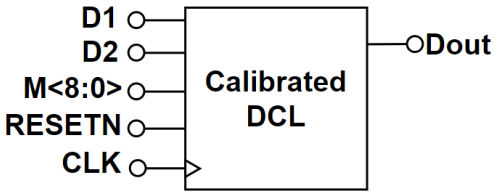


Figure 4.21: Diagram block of the Digital Cancellation Logic.

In order to correctly perform the DCL calibration, it's necessary to recalculate the α coefficients, which are multiplied by the correction factor previously calculated that reflects the $R \cdot C$ variation of the components of the modulator's integrators. As each correction factor is unique it's possible to associate it to it's respective M value, measured by the calibration circuit. Thus, it's also possible to previously calculate the impact of each M in STF_2 and NTF_{Q1} in order to implement different finite impulse response filters, implemented as 6-bit LUTs with the following inputs: $D_1[n-2]$, $D_1[n-1]$, $D_1[n]$, $D_2[n-2]$, $D_2[n-1]$, $D_2[n]$.

Therefore, as each correction factor is unique, it's possible to have several different 6-bit LUTs, that can be accessed to by using the respective M as an index value for the

different tables. As previously mentioned, in Chapter 3, there is a maximum range of 133 different M values and it is possible to use only 20 tables as each table covers an interval of 7 M's ($7 \cdot 19 = 133$) including an extra default table, for typical conditions.

The implementation of this block in VerilogA is quite simple. By having all the tables previously defined, the DCL input M[8:0] is used to select (through a series of *if* conditions) the correct DCL 6-bit LUT that suits the $R \cdot C$ variation of the 2-1 MASH ΣΔM.

Then, by receiving the output signals D1 and D2 of the 2^{nd} and 1^{st} order ΣΔM that compose the MASH, its possible to have clear access to the ouptut code Dout. This block also has the possibility of being reset, with RESETN input signal, which means that DCL uses default LUT. The 20 calculated DCL 6-bit LUT can be seen in the annex I.

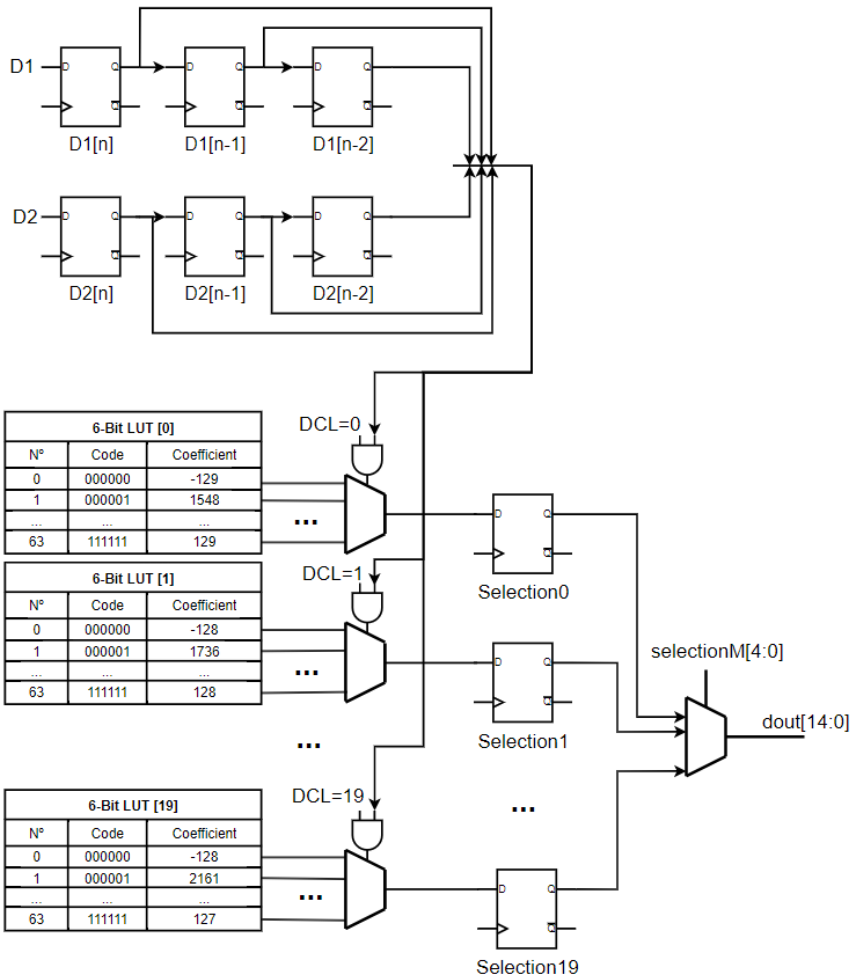


Figure 4.22: Block diagram of the implementation Digital Cancellation Logic.

Fig. 4.22 presents a block diagram of possible implementation of the calibrated DCL. Adding calibration of the DCL causes almost no penalty in power dissipation (besides small leakage current) as the DCL uses only one LUT while the rest remain inactive, as illustrated in Fig. 4.22. It was also verified that the 20 DCL LUTs would occupy and area of 2193 μm^2 which is only six times bigger then the previously used DCL (one LUT), of

345 μm^2. In reality, this amounts to an area increase of the digital part of the ΣΔM of only 5.28%, since most of this area is occupied by the decimation filter.

4.2 α Calibration Circuit's Performance

As each analog and digital block of the α calibration circuit has been carefully analyzed and their performance tested, individually, it is also important to see how the circuit performs as a whole and if it is capable of measuring the RC time constant. To this end, it was decided to test the calibration circuit, separately from the 2-1 MASH ΣΔM and the new implemented DCL, as illustrated in Fig. 4.23, in order to see if it would be able to measure the correct M value and select the necessary reference voltage.

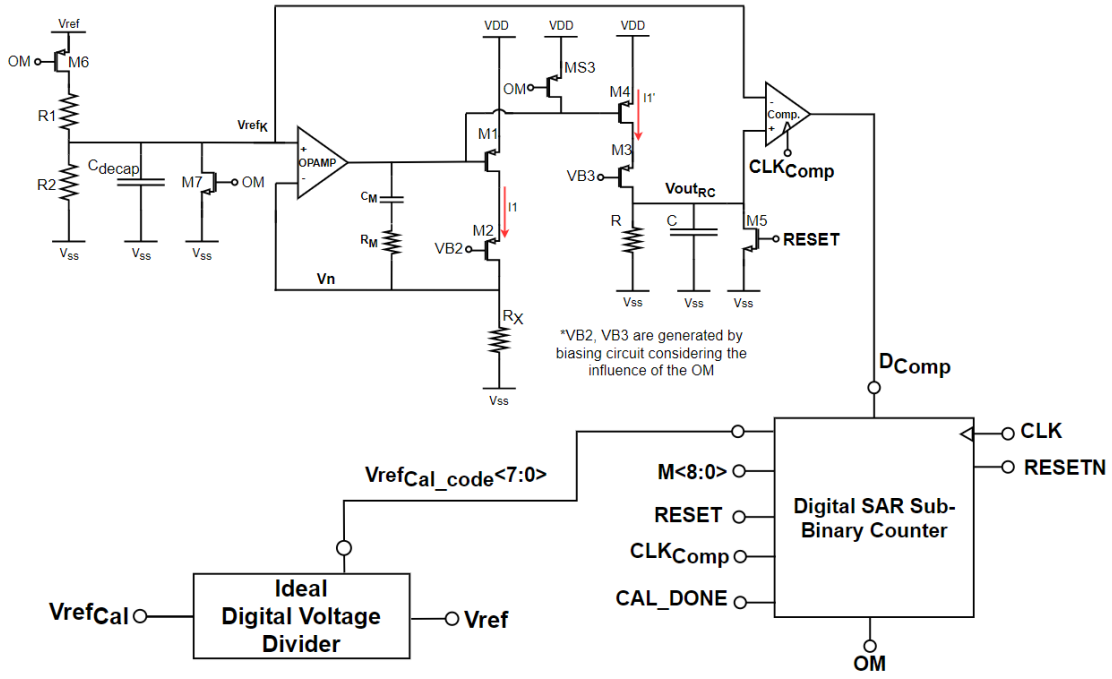


Figure 4.23: Block diagram of the calibration circuit.

Therefore, in this section, the results of the simulations testing the $\frac{RC}{T_S}$ measuring circuit, the ideal digital voltage divider circuit and the digital SAR sub-binary counter are presented in order to see how they operate altogether.

In Fig. 4.24, one can see how the circuit performs one calibration, in the time period of 10 μs by analysing the behaviour of the different signals affected by it's performance.

As illustrated by Fig. 4.24, it's possible to see the OM signal, indicating the different states of the calibration circuit operation. The off state occurs during 10 ns, the stand-by state during 3 μs and the active state during 6 μs, returning then to the off state. As demonstrated by the the variations of the $Vout_{RC}$ signal, the calibration starts when OM[1] signal turns to 1. The signal **Cal_busy** depicts when the calibration is actually occurring during the active state.

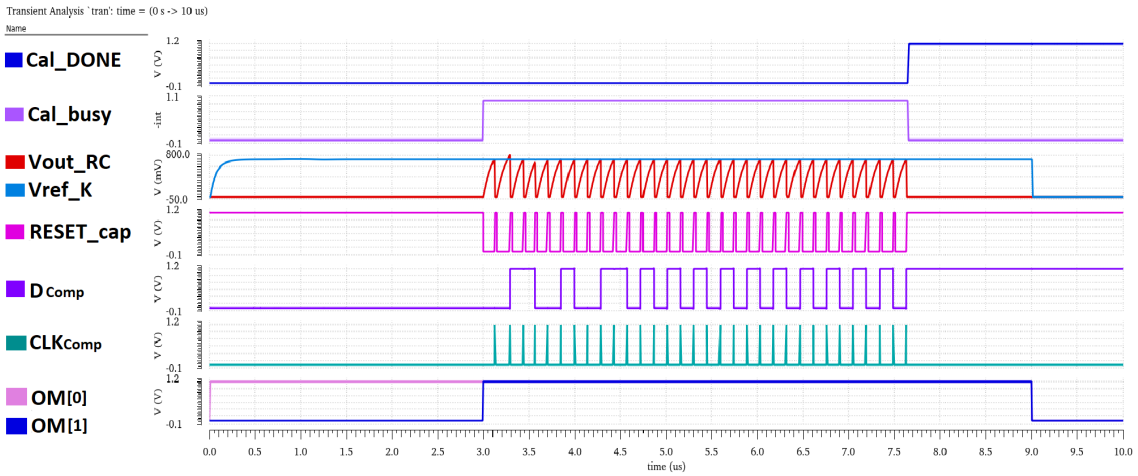


Figure 4.24: Simulation of the behaviour of the different external signals and the internal signal **Cal_busy** during the performance of the calibration circuit.

It is also possible to see that before each comparison between $Vout_{RC}$ and $Vref_K$ the signal CLK_{Comp} triggers the comparator which informs the digital SAR sub-binary counter whether or not the $Vout_{RC}$ signal is bigger then $Vref_K$, and indicates if the M value should be increased or decreased in order to get closer to the equivalent ideal M for the current $R \cdot C$ variation. In this case in particular, in Fig. 4.24, the nominal scenario is demonstrated, and as it can be seen in Fig. 4.25, there is no RC variation in the nominal scenario, which means that the last value of M being measured should be equal to the value of M for nominal conditions (191). The variable **last_m** always saves the last measured value of M, which, in this case is 192. Once a calibration is finished, the value of the M variable is always reset to the default value. In this way, if the calibration is not desired, the circuit will still provide the nominal value of M for the necessary reference voltage and DCL table to be selected. It should also be known that the **counter** variable only counts the clock cycles necessary to do the comparison between $Vout_{RC}$ and $Vref_K$ and reset the capacitor (which is performed by the signal **RESET_cap** from Fig. 4.24), otherwise this variable is reset to zero.

In order to test the performance of all the analog circuits, previously described, together with the SAR sub-binary digital counter, in different conditions, it was decided run a corner's simulation, corresponding to high, medium and low temperatures, different PMOS and NMOS device speed and the corresponding high, medium and low supply voltages. The $R \cdot C$ variation is known for the fast corner as [R-16%; C-15%] and for the slow corner as [R+17%, C+15%], therefore, the expected values of M in these corners are, respectively, $M_{RC_{ff}} = 137$ and $M_{RC_{ss}} = 258$. The results of corners simulations can be seen in Tab. 4.9. It's possible to observe the results of the new reference voltage $V_{ref_{Cal}}$ generated by the ideal digital voltage divider, and compare the voltages $Vref_K$ and Vn. It is important to point out that since the reference voltage varies in the same way as supply voltage, in the performed simulations, $V_{ref_{Cal}}$ is also influenced by this variation.

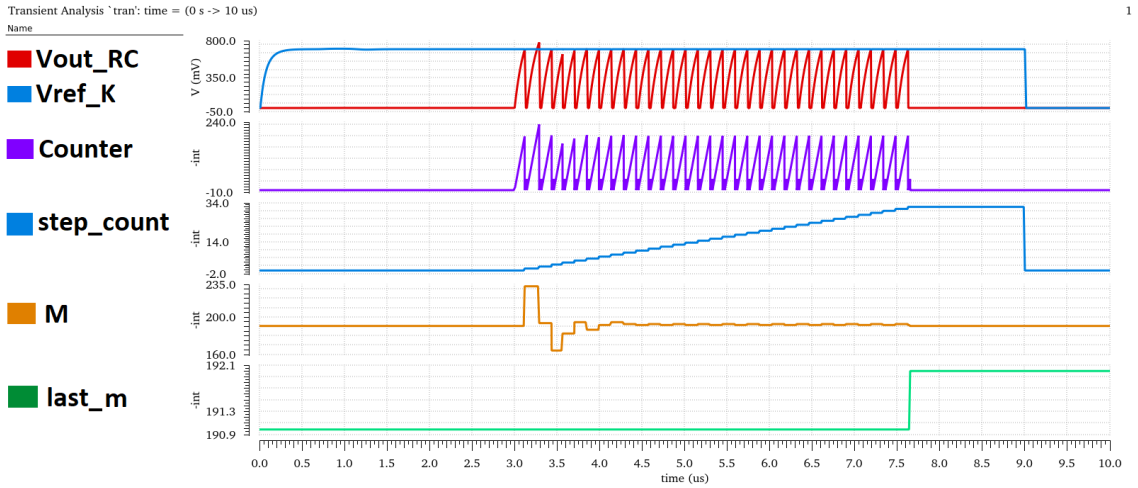


Figure 4.25: Simulation of the behaviour of the different internal signals reacting to the performance of $Vout_{RC}$ during the performance of the calibration circuit.

Table 4.9: Results for supply voltage, temperature and device speed corners simulation of the calibration circuit performance.

Transistors Speed	**tt**	**ff**	**ff**	**ff**	**ff**	**ss**	**ss**	**ss**	**ss**
Temperature [°C]	**27**	**-40**	**125**	**-40**	**125**	**-40**	**125**	**-40**	**125**
Supply Voltage [V]	**1.1**	**1.045**	**1.045**	**1.155**	**1.155**	**1.045**	**1.045**	**1.155**	**1.155**
Expected M	191	137	137	137	137	258	258	258	258
Measured M	192	140	142	140	142	252	258	252	258
$Vref_K$ **[mV]**	698.4	663.2	662.8	733.9	733.5	660.3	662.4	734.2	733.7
Vn **[mV]**	698.1	662.9	662.6	733.7	733.3	684.9	657.8	733.7	733.8
$Vref_{Cal}$ **[mV]**	840.1	1045.0	1045.0	1155.0	1155.0	469.0	469.0	518.4	518.4

5

Validation of the Calibration Concept

The ΣΔM developed, based in [1], relies on matching the values of the analog circuit with the coefficients in the Digital Cancellation Logic to achieve it's desired performance. As any deviation from the nominal component's values causes a degradation in it's performance, this modulator is clearly limited by process variations, as previously analysed in Chapter 3, Section 3.2.4, of this thesis.

Originally, the purpose of this thesis was to study the design and implementation of extra circuits that can measure the values of $R \cdot C$ components of each integrator and calculate their deviation from the nominal RC time constant value. This deviation can be defined as a correction factor that would affect the coefficients obtained by the modulators transfer functions and compensate for the variations of the analog parameters, in the 2-1 MASH ΣΔM's performance. This correction factor would be applied directly to the α coefficients in the DCL's transfer functions, nonetheless, it was realized that it could also be used to adjust the V_{ref} of the ADC in order to decrease the variability of the feedback signals. This approach allows to increase a full scale voltage of the 2-1 MASH ΣΔM, as seen in Chapter 3, Section 3.3.

Thus, this new approach translates into developing a new DCL and a calibration circuit capable of measuring the values of RC components, in order to produce a correction factor that will allow to perform these two calibrations, as thoroughly explained previously, in Chap 3, Section 3.3.1.

However this approach requires some modifications in the 2-1 MASH ΣΔM's circuit. Therefore, in this chapter a block diagram that illustrates how the V_{ref} and DCL calibration is implemented with the respective necessary circuits and the new developed DCL added to the 2-1 MASH ΣΔM's circuit is presented. Further, the modifications made to the ΣΔM that enable the calibration will also be presented. Finally, as the performance of the calibration circuit was analyzed individually, in Chapter 4, Section 4.2, now, the results of a comparison between the modulator's performance metrics, before and after implementing the calibration, will be presented. Thus, in order to validate the proposed

calibration concept, the 2-1 MASH ΣΔM and calibration circuit have been designed in a UMC CMOS 40 nm LP process technology, clocked at a frequency of $F_S = 1.6GHz$ with an input signal frequency of $F_{in} \approx 2MHz$, a bandwidth of 12 MHz and a 1.1 V supply and reference voltage. An ideal digital voltage divider with buffer were used to illustrate how V_{ref} could be calibrated. The digital blocks incorporated in this circuit were implemented in VerilogA.

In Fig. 5.1, it's possible to have a global view of how the different analog and digital circuits, previously described and analyzed in Chapter 4, Section 4.1, are implemented and perform together with the calibrated DCL in order to calibrate the 1^{st} and 2^{nd} Order ΣΔM's that compose the analog part of the 2-1 MASH ΣΔM.

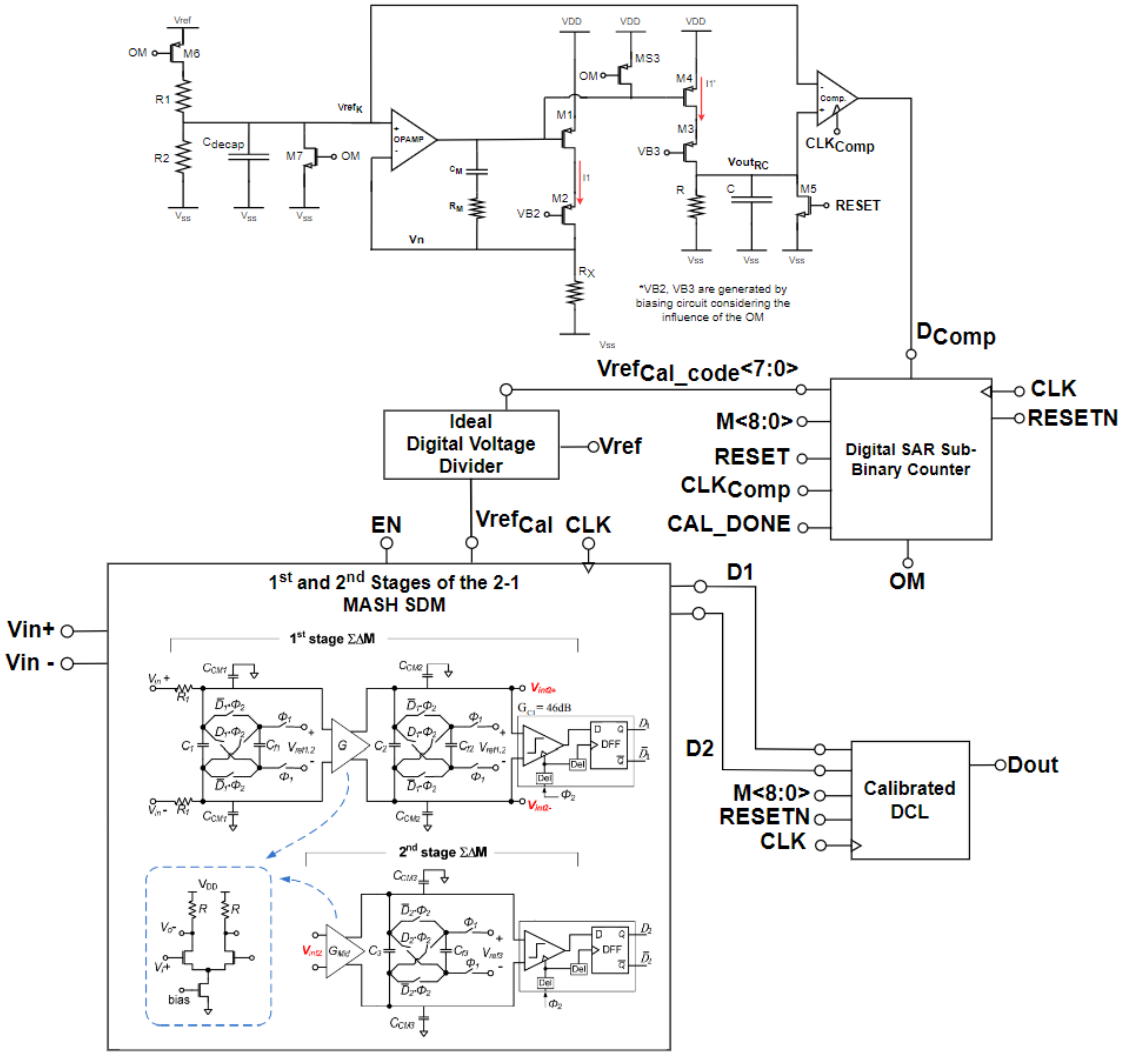


Figure 5.1: Diagram block of the implementations of the added RC calibration circuit and the calibrated DCL to the 2-1 MASH ΣΔM's analog circuit.

The different blocks illustrated in Fig. 5.1, that compose the calibration circuit are the voltage divider that generates the voltage identified as $Vref_K$, the comparator, the amplifier, the $\frac{RC}{T_S}$ measuring circuit, the digital SAR sub-binary counter and the digital

resistive voltage divider which is implemented with ideal components in order to validate the calibration concept.

The two stages of the ΣΔM provide their respective output signals D1 and D2 that select the correct output code Dout of one of the 6-Bit LUTS included in the new developed DCL. As seen, in Chapter 3, Section 3.2, the previously used DCL was implement as a 6-Bit LUT, that contained all the necessary coefficients. Nevertheless, as each correction factor is unique, and has an effect in all the coefficients calculated by transfer functions of the DCL, the new calibrated DCL contains multiple 6-Bit LUTS and the correct one is selected through the use of the correction factor, M[8:0] (from calibration circuit) as an index value. As the correction factor is a reflection of the RC component's process variation, it allows for the selection of the better fitted reference voltage and DCL's 6-Bit LUT.

5.1 Baseline Performance of the 2-1 MASH ΣΔM

In this section, the simulation results of the 2-1 MASH ΣΔM and also the 2^{nd} - order modulator are presented. These results are presented in order to establish a baseline for a comparison between the performance of different implementations of the modulator.

The 2-1 MASH ΣΔM was designed in a UMC CMOS 40 nm LP process technology, clocked at a frequency of $F_S = 1.6GHz$ with an input signal frequency of $F_{in} \approx 2MHz$, a bandwidth of 12 MHz and a 1.1 V supply and reference voltage, for a maximum input differential amplitude ($V_{in,diff}$), of 0.75 [V]. In order to test the performance of the ΣΔM through out different scenarios, it was decided to do a simulation with process, voltage and temperature (PVT) variations depicted in Table 5.1. These variations amount to 32 different corners, and in Figs. 5.2 and 5.3, it's possible to see the results of transient and transient noise simulations of the 2^{nd} - order ΣΔM and the 2-1 MASH ΣΔM, for these corners, respectively.

Table 5.1: External conditions considered for the simulated corners.

Corners Variation		**Typical**	**Maximum**	**Minimum**
Process device speed	**Transistors MOS**	**tt**	**ff**	**ss**
	Resistors	**tt**	**ff**	**ss**
	Capacitors	**tt**	**ff**	**ss**
Voltage	**Supply and Reference [V]**	**1.1**	**1.155**	**1.045**
Temperature [°C]		**27**	**125**	**-40**
Bandwith (B) [MHz]		**12**		
Clock Sampling Frequency (F_S) [GHz]		**1.6**		
Tech. [nm]		**UMC CMOS 40 LP process**		

As one can see in Fig. 5.3, the 2-1 MASH ΣΔM tends to loose performance on the last corners [Cffss 24; Cffss 31], which are the corners in which both resistors and capacitors are being simulated with slow device speed. As stated previously in Section 4.2, the $R \cdot C$ variation is known for the fast corner as [R-16%; C-15%] and for the slow corners as [R+17%, C+15%].

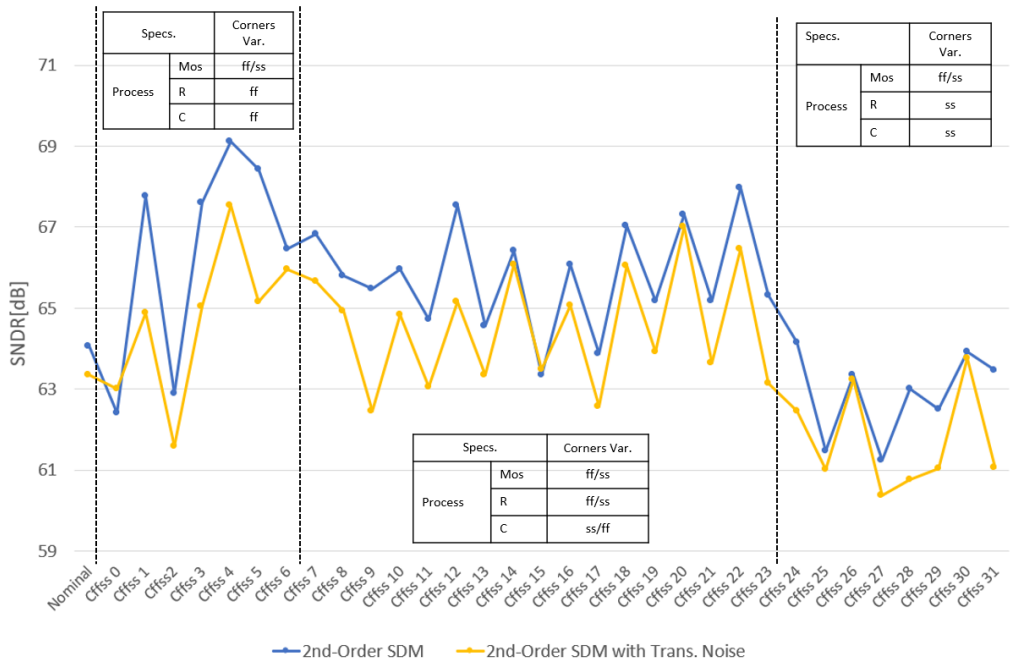


Figure 5.2: SNDR results of the 2^{nd} - order ΣΔM, for the PVT variations, simulated with and without transient noise.

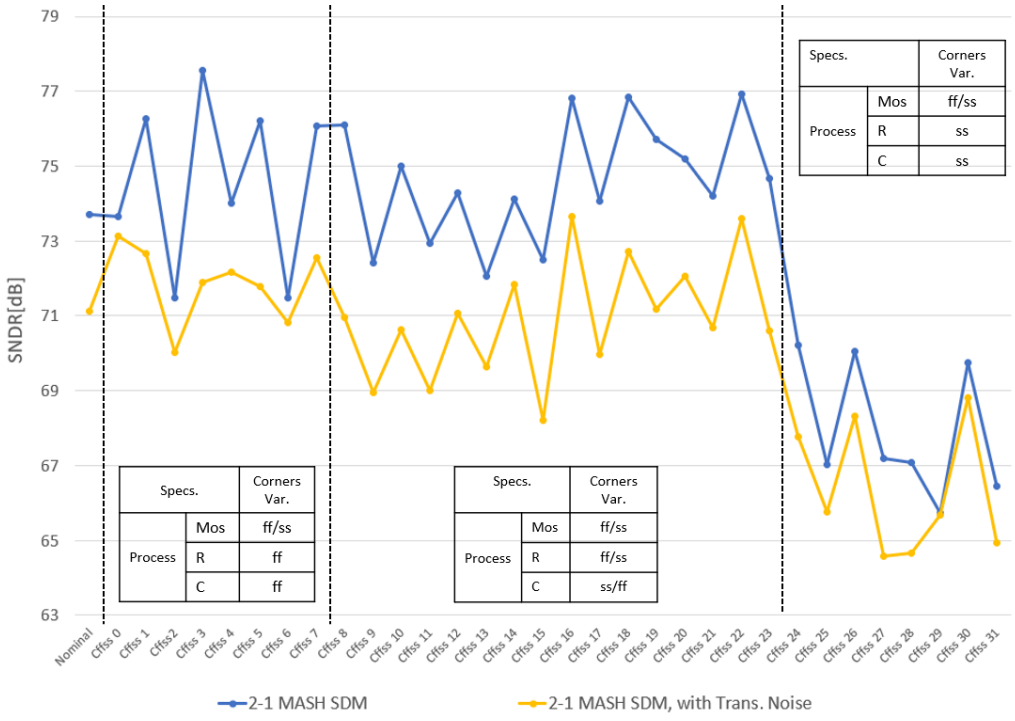


Figure 5.3: SNDR results of the 2-1 MASH ΣΔM, for the PVT variations, simulated with and without transient noise.

Figures 5.4 and 5.5 illustrate the fast Fourier Transform (FFT) of the 2^{nd} - order ΣΔM and the 2-1 MASH ΣΔM output signals for the typical corner, respectively. These FFTs were obtained from the results of simulations without transient noise with 2^{14} points with Hanning window of the output bit-stream, for a 750 mV, 2 MHz input signal, for the typical corner.

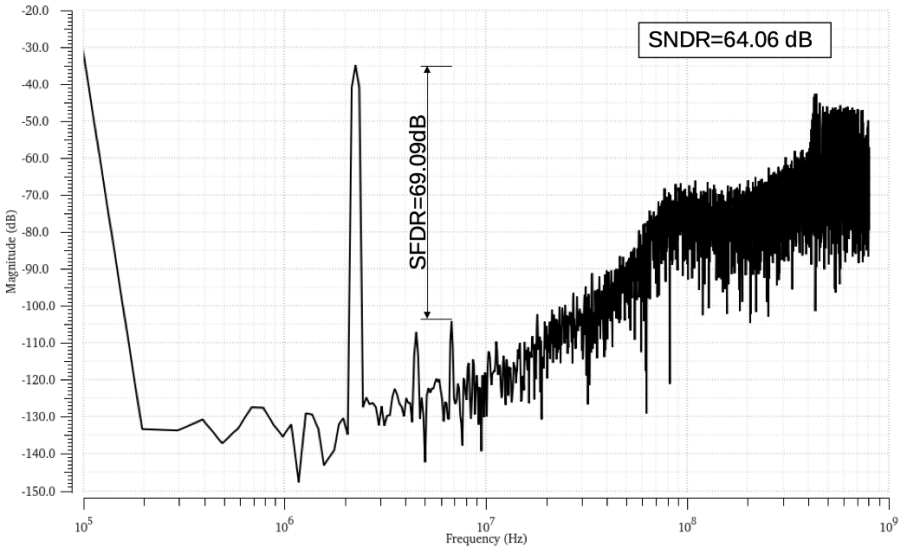


Figure 5.4: Graphic of the Power Spectral Density (PSD) of the 2^{nd}- Order Modulator being simulated in typical conditions.

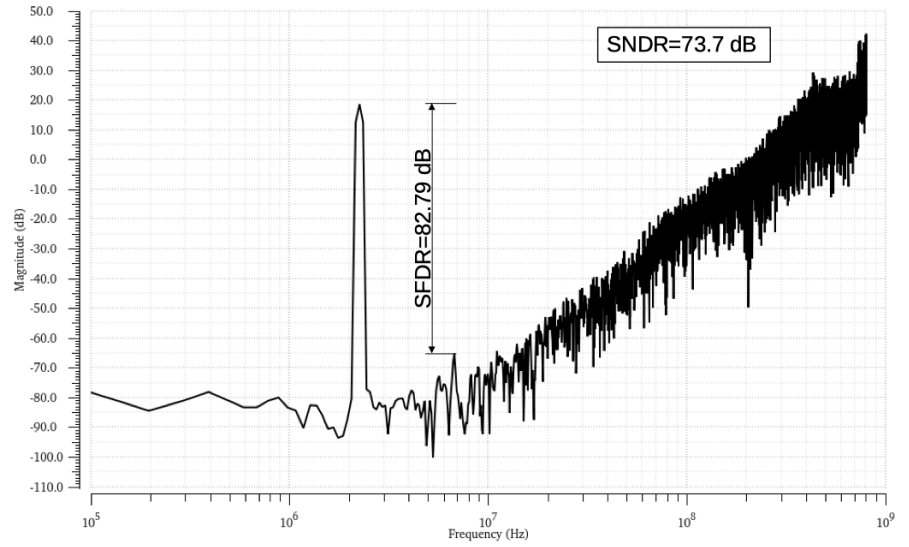


Figure 5.5: Graphic of the Power Spectral Density (PSD) of the 2-1 MASH ΣΔM being simulated in typical conditions.

5.2 Alterations to the Passive Integrators of the 2-1 MASH ΣΔM

In order to perform the calibration of the 2-1 MASH ΣΔM, it was realized that this modulator required some necessary alterations in order to obtain relevant results. These modifications focused on the passive integrators, since the calibration of V_{ref} forces the modulator to be subjected to lower V_{ref} voltages. The ΣΔM was originally optimized for V_{ref}=1.1V, therefore in order to calibrate V_{ref} it's necessary to compensate this variation by adjusting the feedback paths of the modulator, as previously mentioned in Section 3.3.

The first modifications done to all the passive integrators (outlined on Fig. 5.6) focuses on decreasing nominal V_{ref} voltage value by a factor of 0.635 and consequently increase the feedback capacitors, C_{fi}, by the same factor, in order to keep the same value of the feedback signal, as illustrated by the equation 5.1, seen previously in Section 3.3.

$$b_i \approx \frac{V_{ref_i} \cdot 0.635}{\alpha_i} \cdot \frac{C_{fi}/0.635}{C_{eq_i}} \tag{5.1}$$

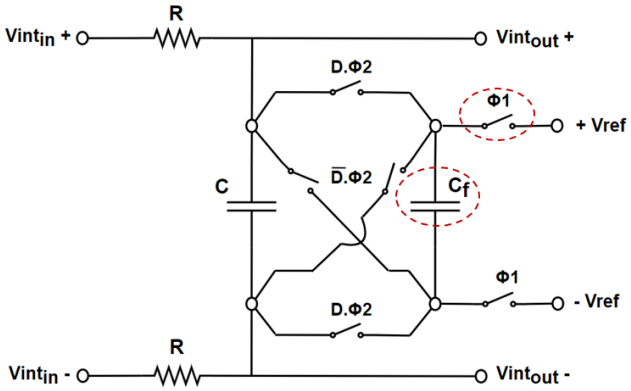


Figure 5.6: Schematic of the differential RC integrator, with outlined component's which suffered the necessary modifications, for all the redesigned integrators.

Now, it's possible to define a $Vref_{Cal}$ range of $\approx [0.497; 1.1]$ V. However, in order to use these voltages, it's necessary to change the switch of phase ϕ_1, connected to the terminal V_{ref}+, from a PMOS switch to a transmission gate switch.

The transmission gate switch is composed by an NMOS transistor and a PMOS transistor, as illustrated in Fig. 5.7. As seen previously, in Fig. 4.2, the resistance characteristic of this type of switch is equivalent to the parallel of each transistor resistance, therefore, this switch has less voltage drop across it's input and output during it's closed state, which means one can pass a full range signal. However, this implementation requires two control signals for each switch, as depicted in Fig 5.7.

Nevertheless, through different simulations it was realized that these modifications weren't enough. Since the transistors that compose all the switches from the integrators are already working in weak inversion when performing in the slow corners, results showed that the Output Swing (OS) of all the integrators increased immensely in these

conditions, causing a significant performance drop of the modulator. This is due to the fact that the settling of the switches, of the first integrator, just isn't enough to provide the necessary feedback, as C_{f1} (which already has the greatest capacitance) is increased to accommodate the new range of $Vref_{Cal}$.

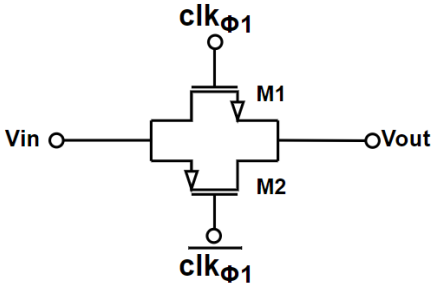


Figure 5.7: Typical implementation of a transmission gate switch.

In order to amend this situation, it was necessary to do some additional alterations to the switches of the 1^{st} integrator. Specifically, doubling the size (W) of the transistors of the switches working in phase ϕ_2, and implement these resized switches with a Bulk-Switching technique.

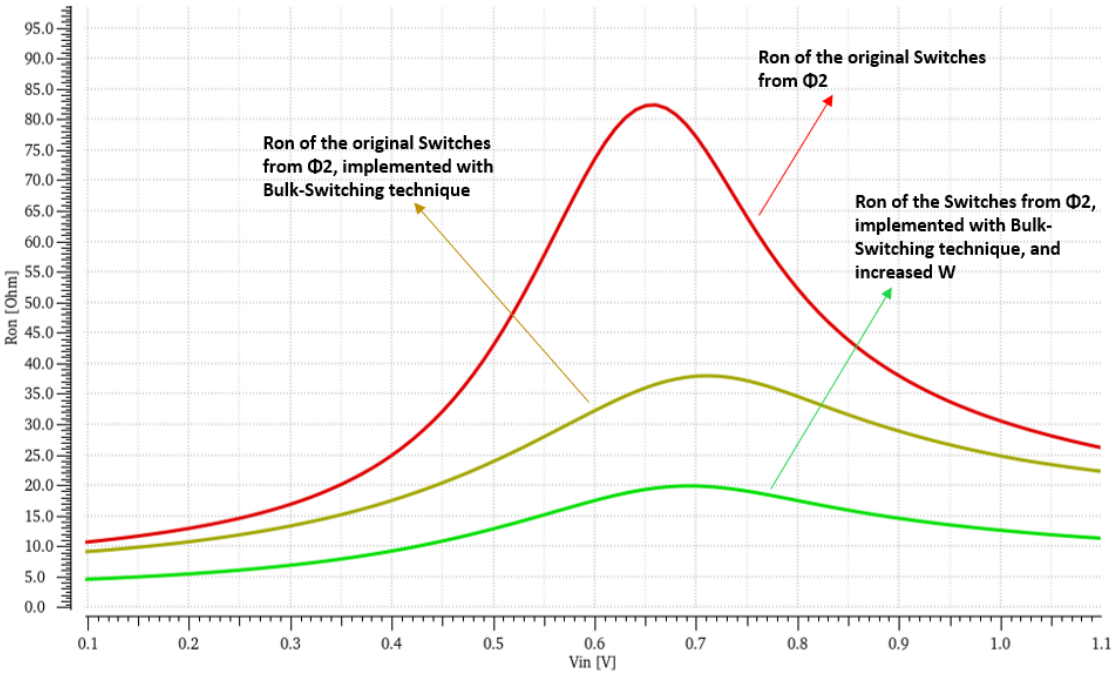


Figure 5.8: Variation of impedance across the voltage swing for the transmission gate switch of phase ϕ_2, through out the modifications made, simulated in typical conditions.

By increasing the size of the transistors (M1 and M2, from Fig. 5.10) that compose the switches working in phase ϕ_2 it was possible to decrease the internal impedance of these transmission gate switches, however, as a consequence there was also an increase

of the parasitic capacitances of these switches, and therefore the capacitance of C_{f1} was decreased to account with this.

As presented in [55], the simple transmission gate switch has the advantage of allowing for a full signal-swing at the cost of when operating with a low supply voltage, the equivalent switch conductance has a large variation, which leads to a large harmonic distortion in the sampled signal. Nonetheless, by implementing the transmission-gate switches with a Bulk-Switching technique, as illustrated in Fig. 5.10, this approach allows to produce a lower threshold voltage value independent of the input signal level, since the body effect is removed. Overall, these modifications allow to reduce the impedance curve of the switches working in phase ϕ_2, as seen from the results of the simulations of the transmission gate switch with different implementations, in typical conditions, presented in Fig. 5.8.

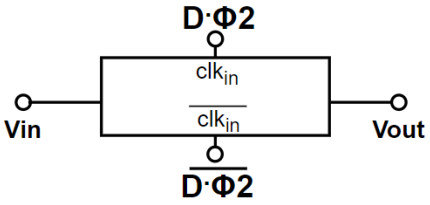


Figure 5.9: Block diagram of the redesigned switches for phase ϕ_2, with the implementation of the Bulk-Switching technique, and increased size of the transistors.

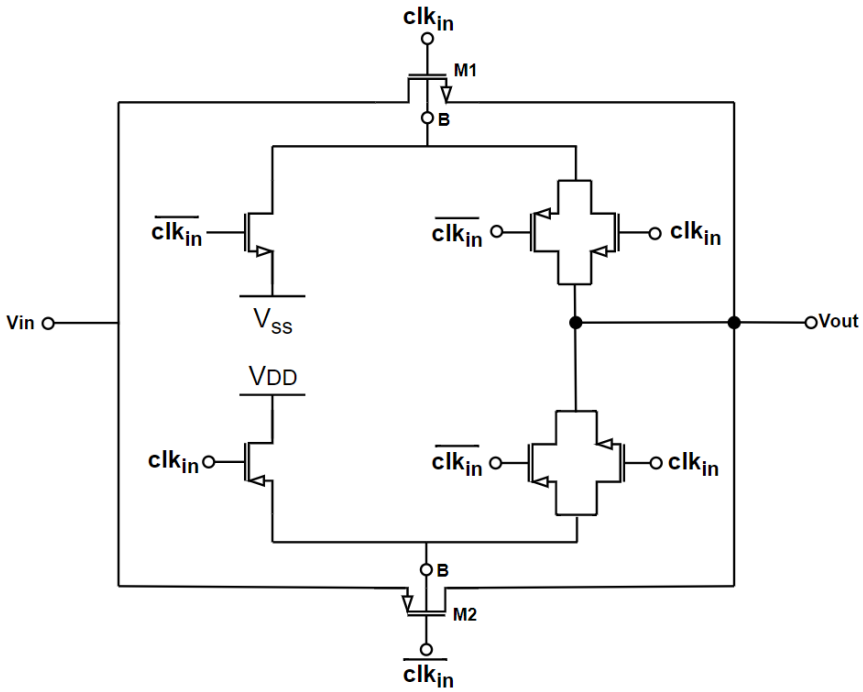


Figure 5.10: Schematic of the electric circuit of the new switches for phase ϕ_2, with the implementation of Bulk-Switching technique.

In Fig. 5.11, one can see the new schematic for the first passive integrator, with the

implementation of a Bulk-Switching technique for the switches. Additionally, it was also decided to use transistors with an Ultra Low Threshold Voltage (ULVT) configuration, in order to lower the ON resistance of the switches and help compensate the feedback. The transistor used in the implementation of the bulk-switching circuit have a smaller size then M1 and M2, and were implement with minimum length size (*Lmin*).

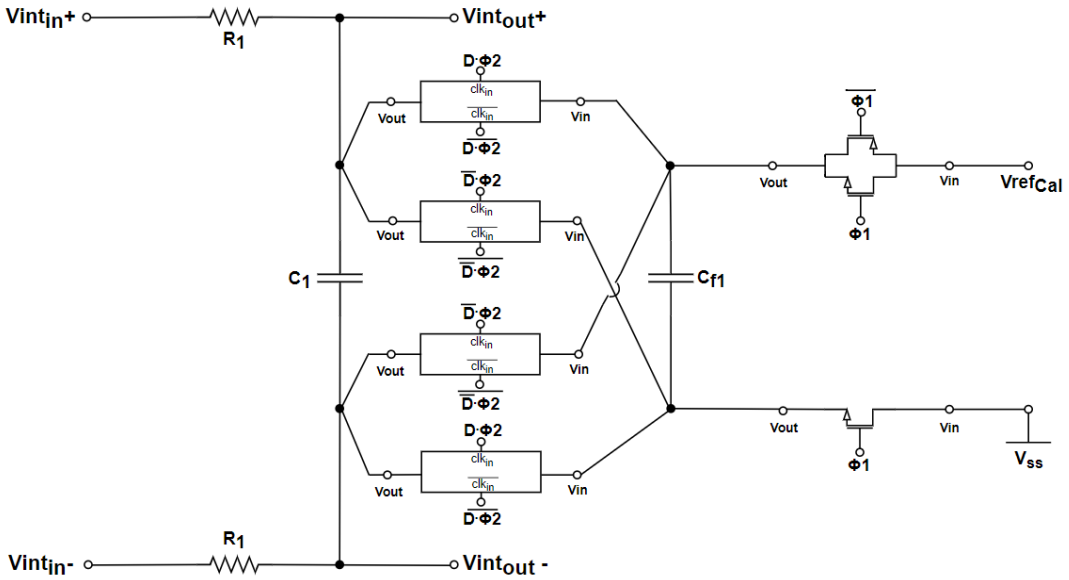


Figure 5.11: Schematic of the electric circuit of the redesigned differential RC integrator, for the first passive integrator, of the 2^{nd} - order ΣΔM.

5.3 Redesigned 2-1 MASH ΣΔM's Performance with the Added Calibration Circuit and Calibrated DCL

In this section, the final simulation results of the calibrated 2-1 MASH ΣΔM and 2^{nd} - order modulator (1^{st} stage), are presented. As mentioned previously, through out this thesis, the calibration has two purposes. Firstly, the V_{ref} of the ADC is adjusted in order to decrease the variability of the feedback signals. Secondly, it allows to calibrate the DCL, improving quantization noise cancellation.

To this end different simulation results will be presented in order to compare the different implementations of the 2-1 MASH ΣΔM. This ΣΔM is implemented with the necessary modifications, previously described in Section 5.2, when simulating the modulator's circuit with calibration. These modifications allow to correctly perform the calibration of V_{ref}, which in turn decreases the variability of the feedback signals, and therefore it's possible to simulate the calibration of the 2-1 MASH ΣΔM with a $V_{in,diff}$ of 1.1 V, as deducted in Chapter 3.3.

As previously mentioned in this chapter, the calibration circuit is designed in the same technology as the modulator (UMC CMOS 40 nm LP process), clocked at a frequency of

$F_S = 1.6GHz$ with an input signal frequency of $F_{in} \approx 2MHz$, a bandwidth of 12 MHz and a 1.1 V supply voltage. An ideal voltage divider with buffer is used to illustrate how V_{ref} could be calibrated. The digital block incorporated in this circuit is implemented in VerilogA, for these simulations. The calibration is tested for the same PVT variations as the original modulator, previously seen in Section 5.1 (Tables 5.1), thus, in Figs. 5.12 and 5.13, it's possible to see the results of transient and transient noise simulations of the redesigned 2^{nd} - order ΣΔM and the 2-1 MASH ΣΔM, for these corners, when being calibrated.

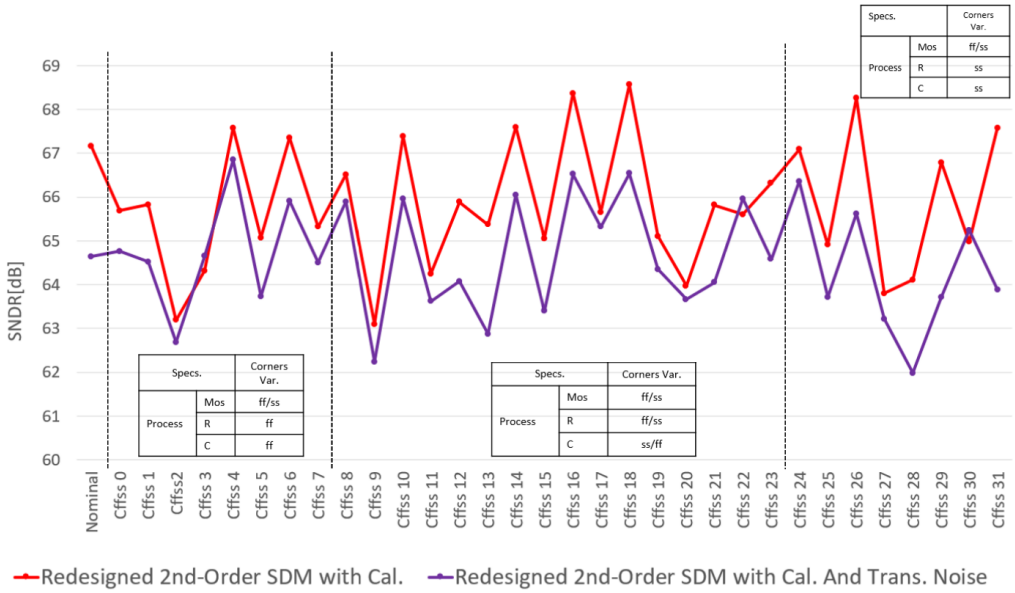


Figure 5.12: SNDR results of the redesigned 2^{nd} - order ΣΔM with calibration, for the PVT variations, simulated with and without transient noise.

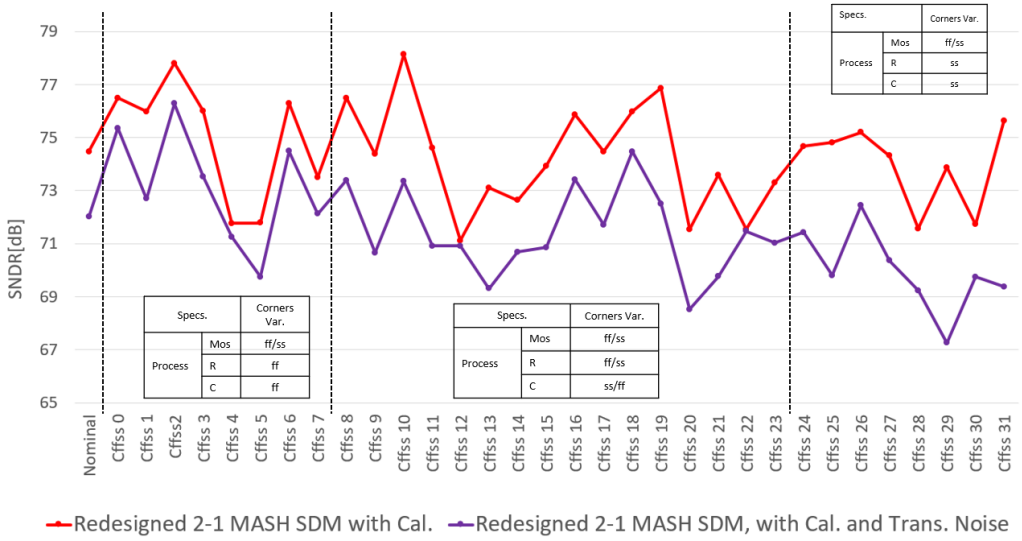


Figure 5.13: SNDR results of the redesigned 2-1 MASH ΣΔM with calibration, for the PVT variations, simulated with and without transient noise.

Figures 5.14 and 5.15 illustrate the fast Fourier Transform (FFT) of the redesigned 2^{nd} - order ΣΔM and the 2-1 MASH ΣΔM output signals, respectively, after calibration

is performed. These FFTs were obtained from the results of simulations of typical corner without transient noise, with 2^{14} points with Hanning window of the output bit-stream, for a 1.1 V, 2 MHz input signal. Figs. 5.16 and 5.17 present a clear comparison between the results of baseline performance and the improvement presented with the implementation of the calibration circuit, when simulated without added transient noise.

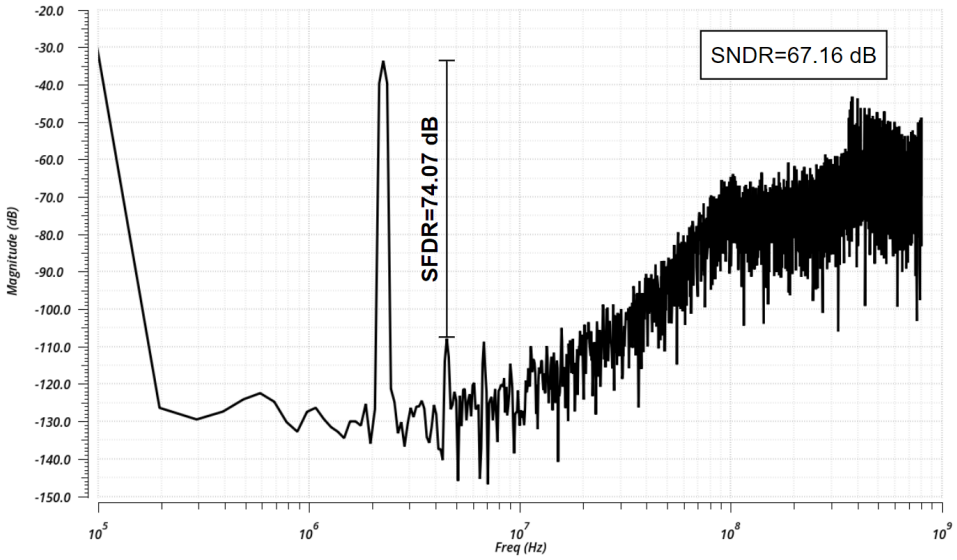


Figure 5.14: Graphic of the Power Spectral Density (PSD) of the redesigned 2^{nd}- Order Modulator being simulated with calibration, in typical conditions, without transient noise.

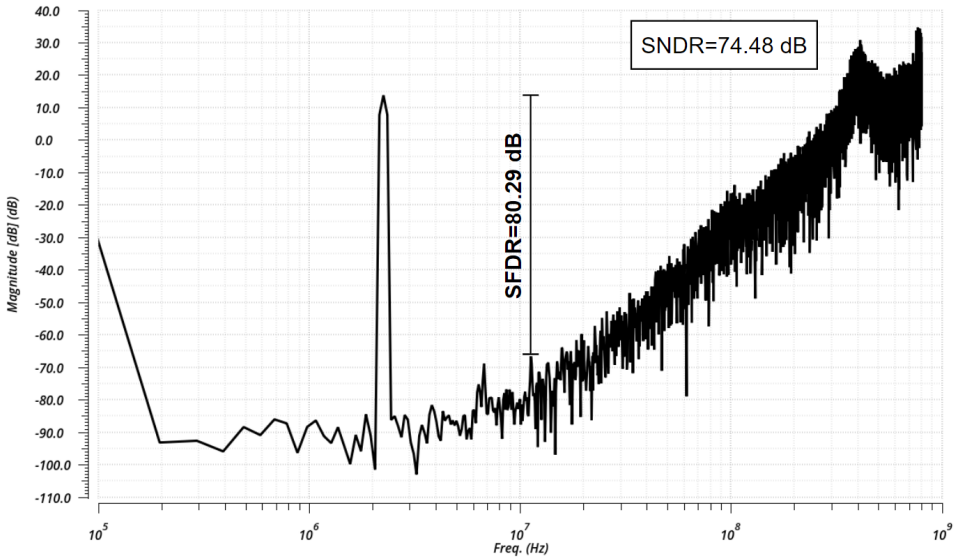


Figure 5.15: Graphic of the Power Spectral Density (PSD) of the redesigned 2-1 MASH ΣΔM being simulated with calibration, in typical conditions, without transient noise.

By analysing these results, presented in Figs. 5.16 and 5.17, one can take three clear conclusions:

- Overall, it's possible to decrease the variability of the 2-1 MASH ΣΔMs performance

with the calibration as the range of results, from simulations without transient noise, goes from [65.74; 77.57], to [71.12; 78.13], increasing the worst corners results to approximately 1 dB ENOB resolution.

- In the slow corners, which is where the modulator tends to loose performance, there is a clear impact in it's performance, caused by calibration, which allows to avoid the drop of the SNDR results.
- There is a drop in the performance of the redesigned 2-1 MASH ΣΔM, when being calibrated, in corners Cffss 4, 12, 20 and 28.

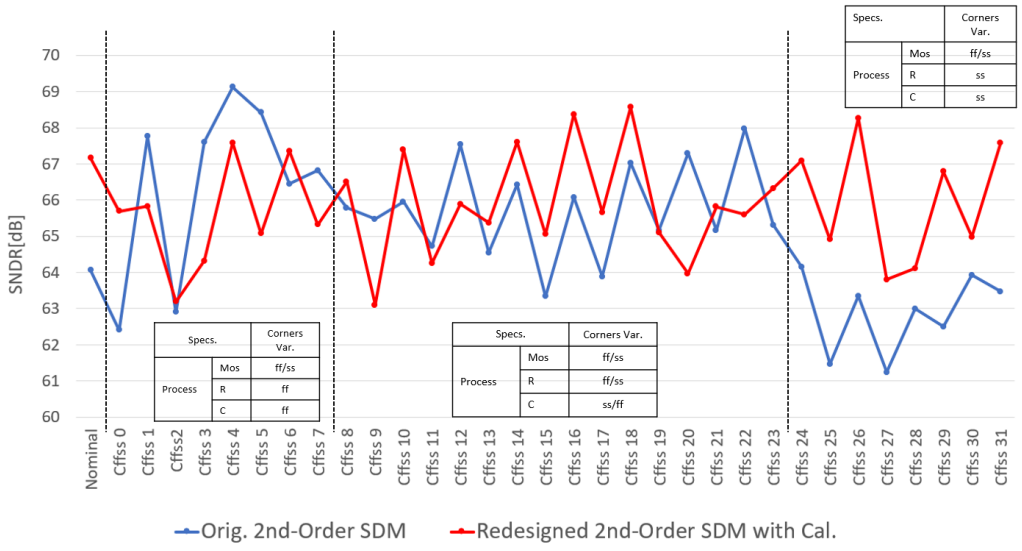


Figure 5.16: SNDR results of the 2^{nd} - order ΣΔM, with and without calibration, for the PVT variations, simulated without transient noise.

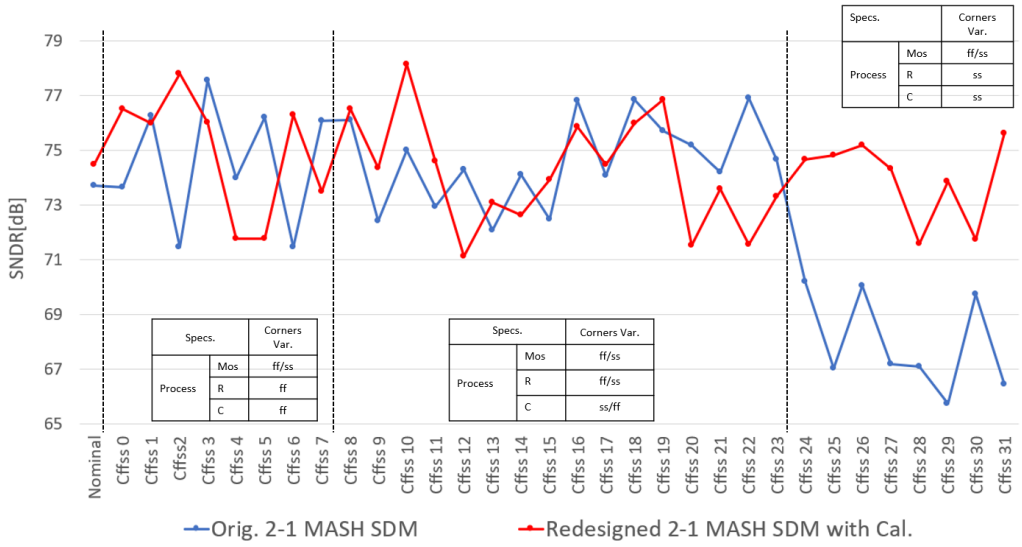


Figure 5.17: SNDR results of the 2-1 MASH ΣΔM with and without calibration, for the PVT variations, simulated without transient noise.

A more detailed analysis of these corners can be done by studying the results of the redesigned 2-1 MASH ΣΔM, presented in Table 5.2. One can see that a common factor between these corners, is the slow transistors MOS configuration, the $-40°C$ temperature and a supply and reference voltage of 1.045 V.

Table 5.2: Results of the redesigned 2-1 MASH ΣΔM circuit performance metrics simulated for PVT corners Cffss 4, 12, 20 and 28.

		Corners Variation			
Process	**MOS**	**ss**	**ss**	**ss**	**ss**
	R	**ff**	**ss**	**ff**	**ss**
	C	**ff**	**ff**	**ss**	**ss**
Voltage	**Supply and Reference [V]**	**1.045**	**1.045**	**1.045**	**1.045**
Temperature [°C]		**-40**	**-40**	**-40**	**-40**
Performance Metrics		**Cffss 4**	**Cffss 12**	**Cffss 20**	**Cffss 28**
ENOB [bit]		11.63	11.52	11.59	11.6
SNDR [dB]		71.78	71.12	71.54	71.58
SNR [dB]		76.86	74.28	74.34	73.75
SFDR [dBFS]		73.91	74.8	76.16	77.01
THD [dB]		-73.2	-73.61	-74.32	-75.01
Selected M		140	188	186	252
Vref Cal [mV]		1.045	798.1	798.1	469

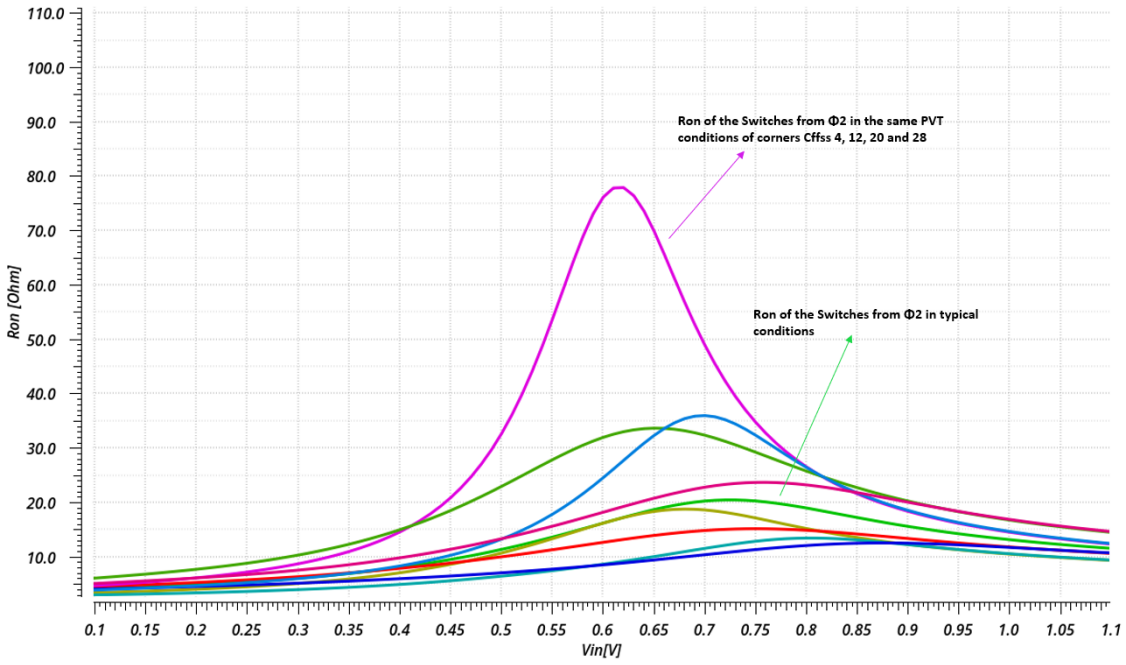


Figure 5.18: Impedance curve of the transmission gate switch working in phase ϕ_2, for different PVT conditions.

To understand if this drop of performance could be related with the switches working in phase ϕ_2, it was decided to simulate these switches separately, for the different PVT variations (without accounting for the proccess variations of the R and C devices as there are none, when testing the switch separately). The results of these simulations can be seen in Fig. 5.18, and as suspected, the impedance curve of these switches increases when being submitted to the PVT conditions of corners Cffss 4, 12, 20 and 28. These results

lead to the conclusion that, indeed, the switches are causing this drop on the performance for these specific corners, and for the input signal from [0.4; 0.8] V, the ON resistance variation is bigger which indicates that the NMOS transistor will be off and the PMOS transistor will be struggling, as it can be seen in Fig. 5.18. A solution for this problem could be to implement these switches with a bootstrapping circuit as it would allow using smaller sizes for the transistors and improve ON resistance variation at the cost of higher power dissipation.

Finally, Figs. 5.19 and 5.20 present a clear comparison between the results of baseline performance and the improvement presented with the implementation of the calibration circuit, when simulated with added transient noise.

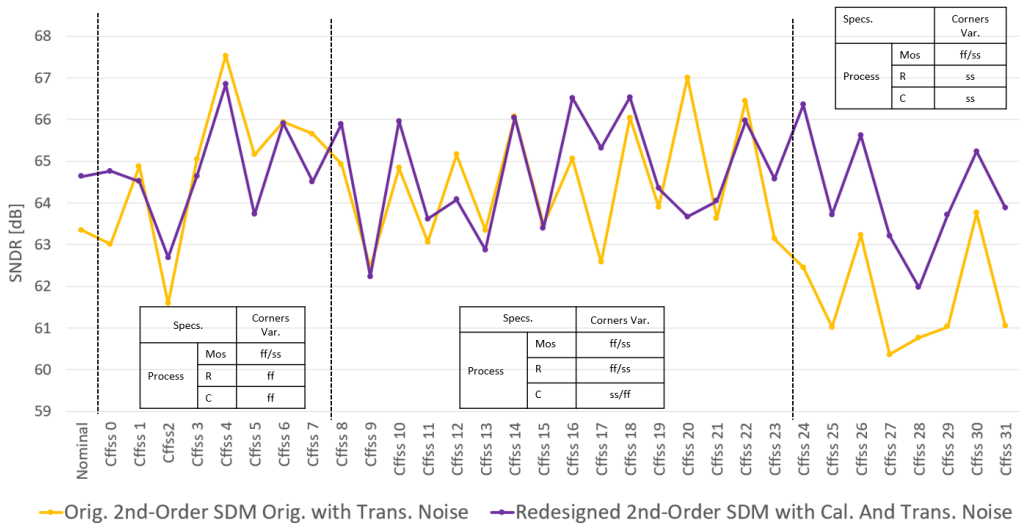


Figure 5.19: SNDR results of the 2^{nd} - order ΣΔM, for the PVT variations, simulated with and without Calibration, with transient noise in both cases.

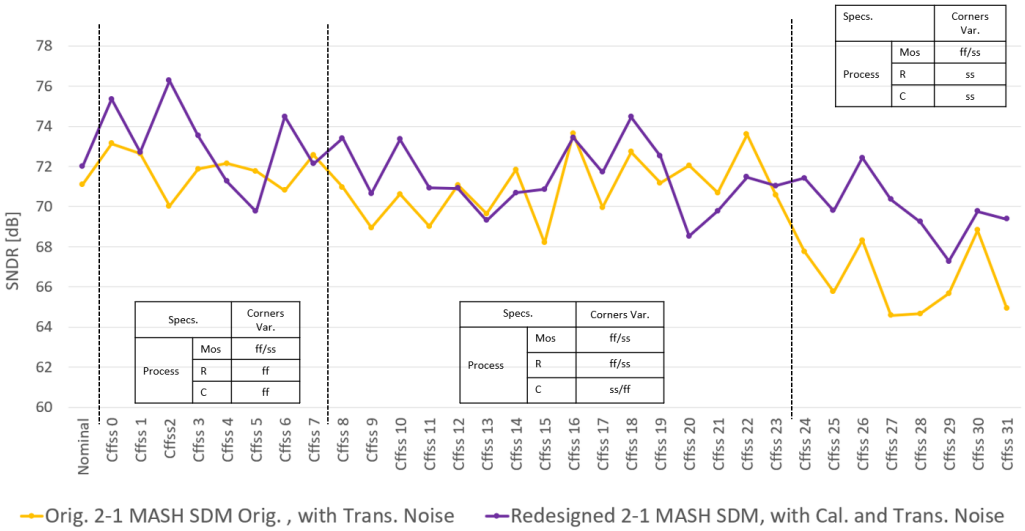


Figure 5.20: SNDR results of the 2-1 MASH ΣΔM with and without calibration, for the PVT variations, simulated without calibration, with transient noise in both cases.

These results were obtained considering no noise contributions on the ideal digital voltage divider block with buffer, as this is an ideal block, only implemented as a way to validate the V_{ref} calibration.

As seen in Figs. 5.19 and 5.20, even with the added transient noise, overall there is a clear advantage in implementing the proposed calibration, since it is possible to increase worst corners results to approximately 0.5 dB ENOB resolution.

To summarize the measured key performance parameters measured between the different implementations of the 2-1 MASH ΣΔM, Tables 5.3 and 5.4 are presented. Table 5.1, previously presented in Section 5.1, shows the external conditions to which both implementations were submitted during simulations, and Table 5.3 and 5.4, present the results of the measured key performance parameters, simulated with and without transient noise, respectively.

It is important to point out that this thesis focuses on validating the calibration concept previously presented, and as some blocks were implemented on an ideal level, only to demonstrate the 2-1 MASH ΣΔM performance when being calibrated, it is not possible to present accurate area and power consumption of the ADC.

Table 5.3: Measured key performance paremeters for corners simulation of the 2-1 MASH ΣΔM's circuit without Transient Noise.

		Corners		
MASH Impl.	**Performance Metrics**	**Typ.**	**Max.**	**Min.**
Original 2-1 MASH ΣΔM, without Cal.	**ENOB [bit]**	11.95	12.59	10.63
	SNDR [dB]	73.7	77.57	65.74
	SNR [dB]	74.69	77.82	67.58
	SFDR [dBFS]	82.79	85.17	71.19
	THD [dB]	-79.2	-85.85	-69.62
	$V_{in,diff}$ **[mV]**	750.0		
Redesigned 2-1 MASH ΣΔM with Cal. of Vref and DCL Coeffiecients	**ENOB [bit]**	12.08	12.68	11.52
	SNDR [dB]	74.48	78.13	71.12
	SNR [dB]	77.04	78.29	74.28
	SFDR [dBFS]	80.29	86.96	74.8
	THD [dB]	-77.49	-86.91	-73.61
	$V_{in,diff}$ **[V]**	1.1		

Table 5.4: Measured key performance paremeters for corners simulation of the 2-1 MASH ΣΔM's circuit with Transient Noise.

		Corners		
MASH Impl.	**Performance Metrics**	**Typ.**	**Max.**	**Min.**
Original 2-1 MASH ΣΔM, without Cal.	**ENOB [bit]**	11.52	11.94	10.43
	SNDR [dB]	71.11	73.64	64.58
	SNR [dB]	71.4	74.19	65.05
	SFDR [dBFS]	82.11	84.11	73.5
	THD [dB]	-79.14	-80.19	-71.76
	$V_{in,diff}$ **[mV]**	750.0		
Redesigned 2-1 MASH ΣΔM with Cal. of Vref and DCL Coeffiecients	**ENOB [bit]**	11.67	12.38	10.88
	SNDR [dB]	72.01	76.28	67.27
	SNR [dB]	73.56	77.74	68.65
	SFDR [dBFS]	80.57	82.69	74.47
	THD [dB]	-76.35	-80.77	-71.91
	$V_{in,diff}$ **[V]**	1.1		

Additionally, in order to verify the sensitivity of the proposed calibration circuit to mismatch errors, a 200 case Monte Carlo analysis of schematic of ADC with calibration was performed. This is due to the fact that, in the used technology, a reasonable number of Monte Carlo iterations is 30. The statistical significance of 30 iterations is quite high. If the circuit operates correctly for all 30 iterations, there is 99% probability that over 80% of all possible component values operate correctly. Thus it was decided to run 50 cases for 4 different corner specifications (considering supply, V_{ref} and temperature corners), resulting in the 200 cases. Results showed a mean value of 71.5678 dB with a standard deviation 1.58895 dB, which shows the robustness of the calibration circuit (Fig. 5.21).

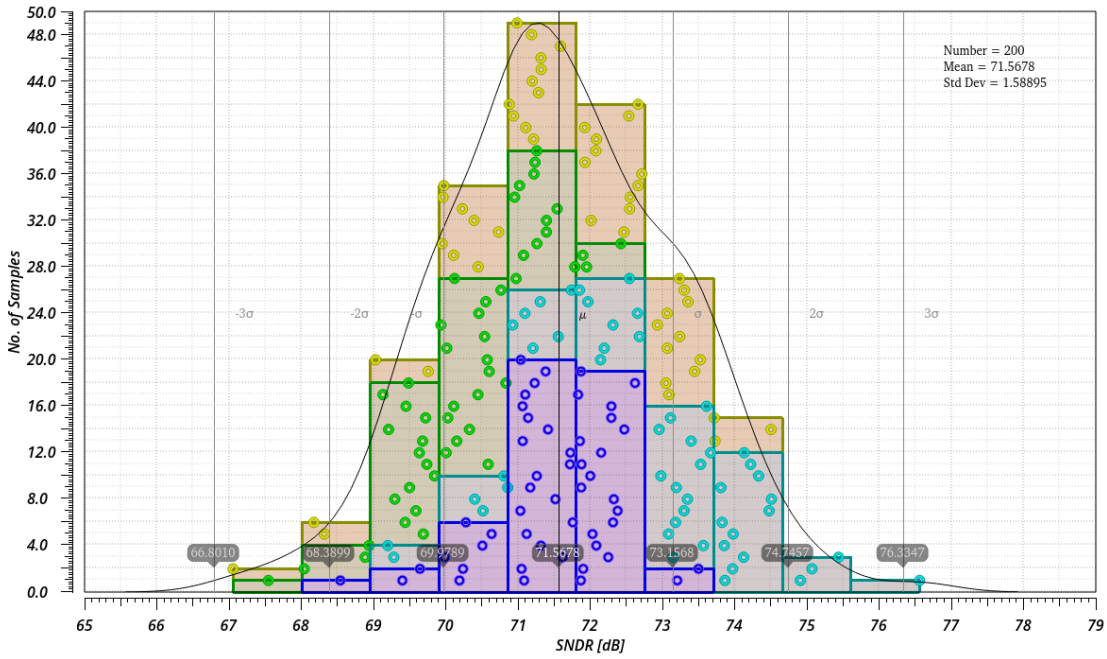


Figure 5.21: Histogram of SNDR results of the redesigned 2-1 MASH ΣΔM with calibration, for Monte Carlo Simulation of 200 cases with electrical transient noise.

Finally, it is important to point out that results presented in Figs. 5.19 and 5.20, in reality, are comparing the performance of completely different implementations of the 2-1 MASH ΣΔM. And in the interest of seeing the improvement that the calibration causes to the redesigned 2-1 MASH ΣΔM, a new simulation was run. In Fig. 5.22, it's possible to see the performance of the redesigned 2-1 MASH ΣΔM, implemented with the modifications presented in Section 5.2 and simulated with a $V_{in,diff}$ of 0.75 V and a fixed V_{ref} value of 0.6985 V ($= 1.1 \cdot 0.635$), versus, the performance of the same redesigned ΣΔM being simulated with calibration of the reference voltage and DCL coefficients and a $V_{in,diff}$ of 1.1 V. Fig. 5.22, shows that using calibration allows to obtain a smaller variation of the redesigned 2-1 MASH ΣΔMs performance across corners resulting in a more robust system, with overall better results. Table 5.5 summarizes the measured key performance parameters measured between the different implementations of the redesigned 2-1 MASH ΣΔM.

Table 5.5: Measured key performance paremeters for corners simulation of the 2-1 MASH ΣΔM's circuit with Transient Noise.

		Corners		
MASH Impl.	**Performance Metrics**	**Typ.**	**Max.**	**Min.**
Redesigned 2-1 MASH ΣΔM, without Cal.	**ENOB [bit]**	11.1	11.46	9.754
	SNDR [dB]	68.61	70.74	60.49
	SNR [dB]	68.89	71.45	61.68
	SFDR [dBFS]	79.57	81.31	69.13
	THD [dB]	-76.68	-77.04	-65.52
	$V_{in,diff}$ **[mV]**	750.0		
Redesigned 2-1 MASH ΣΔM with Cal. of Vref and DCL Coeffiecients	**ENOB [bit]**	11.67	12.38	10.88
	SNDR [dB]	72.01	76.28	67.27
	SNR [dB]	73.56	77.74	68.65
	SFDR [dBFS]	80.57	82.69	74.47
	THD [dB]	-76.35	-80.77	-71.91
	$V_{in,diff}$ **[V]**	1.1		

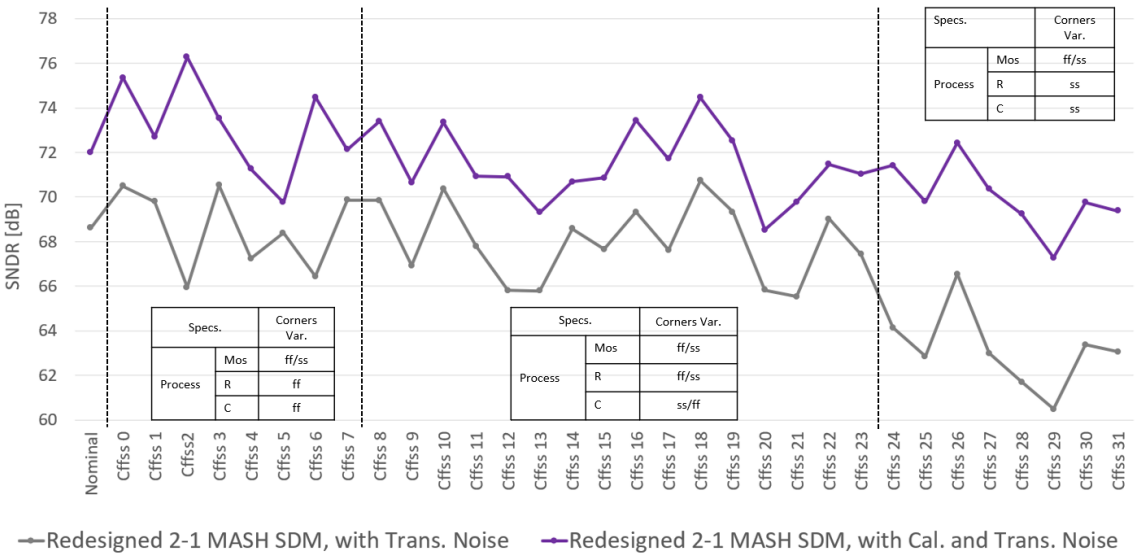


Figure 5.22: SNDR results of the redesigned 2-1 MASH ΣΔM, for the PVT variations, simulated with and without Calibration, with transient noise.

6

Conclusions and Future Work

This thesis is focused on the improvement of a 2-1 MASH ΣΔM using passive RC integrators and low-gain blocks. MASH structures, typically, are sensitive to the analog circuit imperfections, as their performance relies on the accuracy of the transfer functions matching the digital and analog parts of the circuit in order to avoid noise leakage. This illustrates a problem for the 2-1 MASH ΣΔM based on [1], which has it's performance limited by process variations of the values of it's constituted components.

Thus, this document, illustrates how a calibration scheme for the 2-1 MASH ΣΔM was implemented.

Firstly, a calibration architecture, based on the measurement of the relevant analog circuit's parameters variation, is proposed. This variation is converted into a digital coefficient that is used to decrease the variability of the feedback signals and update the DCL in order to reduce quantization noise of the modulator.

Then, a high-level model of the proposed architecture is presented and it's possible to see that the obtained results corroborate the pursuit of the proposed calibration concept, as seen in Chapter 3. This is then followed by electrical and circuit level simulations of the calibration circuit, presented in Chapter 4, where each of the building blocks of this circuit is tested.

Lastly, in order to validate the calibration concept, a comparison between the modulator's performance metrics, before and after implementing the calibration, is presented in Chapter 5. As seen in Chapter 3, since an optimization methodology based on genetic algorithms is used to design the MASH ΣΔM to minimize the impact of analog variations on it's performance, this optimization takes into consideration the variation of all the analog parameters in the circuit and finds the solution with the lowest sensitivity that maximizes the SNDR for a desired maximum input differential amplitude [7].

As the original variation of the reference voltage is smaller than the variation of R and C components, the overload amplitudes values are grouped around 3 values, corresponding the variations of $R \cdot C$, as seen in Section 3.3. An objective of the optimization process was to ensure proper functionality of the modulator in all PVT corners, considering a maximum $V_{in,diff}$ of 0.75 V. This caused a selection of a design point (circuit sizing) to

be conservative. This affects the performance of 2-1 MASH ΣΔM when being calibrated, since it was originally optimized for a V_{ref} of 1.1 V, and now is being subjected to a new V_{ref} range of [0.469;1.115] V ($Vref_{Cal}$). Thus, the switches of the passive integrators were redesigned for these conditions, as seen in Chapter 5. These modifications allowed to validated the calibration concept presented in this thesis, by seeing the overall improvement in the performance of the 2-1 MASH ΣΔM. Measurement results showed that the 2-1 MASH ΣΔM implemented with calibration achieves peak SNR/SNDR/SFDR of 77.74 dB/ 76 dB/82.69 dBFS, for an input signal bandwidth of 12 MHz, simulated with added transient noise for different process, voltage and temperature (PVT) corners. Furthermore, due to calibration is possible to guarantee a minimum SNR/SNDR/SFDR of 68.65 dB/ 67.27 dB/74.47 dBFS, in the same conditions. Due to Monte Carlo simulations it was possible to validate the reliability of the calibration implemented. Therefore, the main future steps following this thesis would be:

- Develop a new optimization methodology based on the previously studied genetic algorithm, considering the implemented High-Level model of the calibration circuit, proposed in this thesis. Due to the calibration, it should allow for a selection of a less conservative design point of the 2-1 MASH ΣΔM with better performance.
- When implementing the new 2-1 MASH ΣΔM, it would be important to further improve the redesigned the switches in the first passive integrator, considering a bootstraping technique, which would allow to implement the switches with smaller size transistors and and decrease the ON resistance.
- Since an ideal block was used to demonstrate a possible future implementation of the calibration of the reference voltage, it is necessary to design a bandgap circuit and a buffer circuit to substitute the ideal digital voltage divider, taking into consideration the effects of the $Vref_{Cal}$ signal and it's variation in the overall performance of the 2-1 MASH ΣΔM.
- After performing these alterations it would be desired to do the layout of the final 2-1 MASH ΣΔM with the mentioned alterations and added calibration circuit, in order to have on-chip results, in the future.

Bibliography

[1] B. Nowacki. In: *Design of Sigma-Delta Modulators for Analog-to-Digital Conversion Intensively Using Passive Circuits*. Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa., 2016. URL: http://hdl.handle.net/10362/18507 (cit. on pp. vii, ix, 1, 2, 4, 8–11, 14–26, 28, 30, 32, 34–36, 38–41, 65, 88, 105).

[2] R. Schreier and G. C. Temes. In: *Understanding Delta-Sigma Data Converters*. John Wiley and Sons, Ltd, 2005. ISBN: 9781119258308. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119258308.fmatter. URL: https://ieeexplore.ieee.org/servlet/opac?bknumber=5264508 (cit. on pp. 1, 16, 19–21, 23).

[3] V. Kumar and D. Chen. "An overview and behavioral modeling of higher order multi-bit (ΣΔ) A/D converters". In: *2008 IEEE International Conference on Electro/Information Technology*. 2008, pp. 128–133. DOI: 10.1109/EIT.2008.4554280 (cit. on p. 1).

[4] A. Hussain et al. "Active–Passive ΔΣ Modulator for High-Resolution and Low-Power Applications". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.1 (2017), pp. 364–374. DOI: 10.1109/TVLSI.2016.2580712 (cit. on p. 2).

[5] H. Wang, F. Schembari, and R. B. Staszewski. "Passive SC ΔΣ Modulator Based on Pipelined Charge-Sharing Rotation in 28-nm CMOS". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.2 (2020), pp. 578–589. DOI: 10.1109/TCSI.2019.2944467 (cit. on p. 2).

[6] B. Nowacki, N. Paulino, and J. Goes. "A Third-Order MASH ΣΔ Modulator Using Passive Integrators". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.11 (2017), pp. 2871–2883. DOI: 10.1109/TCSI.2017.2704164 (cit. on p. 2).

[7] J. L. A. de Melo et al. "A Systematic Design Methodology for Optimization of Sigma-Delta Modulators Based on an Evolutionary Algorithm". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.9 (2019), pp. 3544–3556. DOI: 10.1109/TCSI.2019.2925292 (cit. on pp. 3, 65, 105).

[8] D. A. Naydenov. In: *An Ultra Low Power Amplifier-less Sigma-Delta Modulator for Audio Applications*. Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa., 2019. URL: http://hdl.handle.net/10362/76572 (cit. on pp. 6, 7, 10–14, 16, 24, 25).

[9] M. Ortmanns and F. Gerfers. *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*. 1st ed. Springer Series in Advanced Microelectronics. Springer, 2005. ISBN: 3540284060; 9783540284062. URL: https://link.springer.com/book/10.1007%2F3-540-28473-7 (cit. on pp. 9, 13, 14, 22).

[10] T. C. Carusone, D. Johns, and K. Martin. *Analog Integrated Circuit Design*. 2nd ed. John Wiley & Sons, Inc, 2011. ISBN: 978-0-470-77010-8 (pbk.) URL: https://www.wiley.com/en-us/Analog+Integrated+Circuit+Design,+2nd+Edition-p-9780470770108 (cit. on pp. 10, 11).

[11] J. M. de la Rosa. "Sigma-Delta Modulators: Tutorial Overview, Design Guide, and State-of-the-Art Survey". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.1 (2011), pp. 1–21. DOI: 10.1109/TCSI.2010.2097652 (cit. on p. 15).

[12] P. Aziz, H. Sorensen, and J. vn der Spiegel. "An Overview of Sigma-Delta Converters". In: *IEEE Signal Processing Magazine* 13.1 (1996), pp. 61–84. DOI: 10.1109/79.482138 (cit. on p. 21).

[13] H. Li et al. "Two effective single-loop high-performance sigma-delta modulators based on 0.13μm CMOS". In: *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*. 2010, pp. 342–344. DOI: 10.1109/ICSICT.2010.5667728 (cit. on p. 23).

[14] J. L. A. de Melo et al. "A simple class-D audio power amplifier using a passive CT ΣΔ modulator for medium quality sound systems". In: *2015 22nd International Conference Mixed Design of Integrated Circuits Systems (MIXDES)*. 2015, pp. 543–546. DOI: 10.1109/MIXDES.2015.7208582 (cit. on p. 24).

[15] A. F. Yeknami, F. Qazi, and A. Alvandpour. "Low-Power DT ΣΔ Modulators Using SC Passive Filters in 65 nm CMOS". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61.2 (2014), pp. 358–370. DOI: 10.1109/TCSI.2013.2278346 (cit. on p. 24).

[16] Y. Chae, I. Lee, and G. Han. "A 0.7V 36W 85dB-DR Audio ΣΔ Modulator Using Class-C Inverter". In: *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. 2008, pp. 490–630. DOI: 10.1109/ISSCC.2008.4523271 (cit. on p. 27).

[17] N. Maghari and U.-K. Moon. "A third-order DT ΣΔ modulator using noise-shaped bidirectional single-slope quantizer". In: *2011 IEEE International Solid-State Circuits Conference*. 2011, pp. 474–476. DOI: 10.1109/ISSCC.2011.5746403 (cit. on p. 27).

[18] L. Yao, M. Steyaert, and W. Sansen. "A 1-V, 1-MS/s, 88-dB sigma-delta modulator in 0.13-m digital CMOS technology". In: July 2005, pp. 180–183. ISBN: 4-900784-01-X. DOI: 10.1109/VLSIC.2005.1469362 (cit. on p. 27).

[19] F. Michel and M. Steyaert. "A 250 mV 7.5 W 61 dB SNDR SC ΣΔ modulator using near-threshold-voltage-biased inverter amplifiers in 130 nm CMOS". In: *Solid-State Circuits, IEEE Journal of* 47 (Mar. 2012), pp. 709–721. DOI: 10.1109/JSSC.2011.2179732 (cit. on p. 27).

[20] R. van Veldhoven, N. Nizza, and L. J. Breems. "Technology portable, 0.04mm2, Ghz-rate ΣΔ modulators in 65nm and 45nm CMOS". In: *2009 Symposium on VLSI Circuits* (2009), pp. 72–73 (cit. on p. 27).

[21] A. Pena-Perez, E. Bonizzoni, and F. Maloberti. "A 84dB SNDR 100kHz bandwidth low-power single op-amp third-order ΣΔ modulator consuming 140μW". In: *2011 IEEE International Solid-State Circuits Conference* (2011), pp. 478–480 (cit. on p. 27).

[22] S. Kwon and F. Maloberti. "A 14mW Multi-bit ΣΔ Modulator with 82dB SNR and 86dB DR for ADSL2+". In: *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*. 2006, pp. 161–170. DOI: 10.1109/ISSCC.2006.1696045 (cit. on p. 27).

[23] J. Koh, Y. Choi, and G. Gomez. "A 66dB DR 1.2V 1.2mW single-amplifier double-sampling 2nd-order ΣΔ ADC for WCDMA in 90nm CMOS". In: *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.* 2005, 170–591 Vol. 1. DOI: 10.1109/ISSCC.2005.1493923 (cit. on p. 27).

[24] K. Lee et al. "A Noise-Coupled Time-Interleaved Delta-Sigma ADC With 4.2 MHz Bandwidth, -98 dB THD, and 79 dB SNDR". In: *IEEE Journal of Solid-State Circuits* 43.12 (2008), pp. 2601–2612. DOI: 10.1109/JSSC.2008.2006311 (cit. on p. 27).

[25] H. Park et al. "A 0.7-V 100-dB 870 μW digital audio ΣΔ modulator". In: *2008 IEEE Symposium on VLSI Circuits* (2008), pp. 178–179 (cit. on p. 27).

[26] J. de Melo, J. Goes, and N. Paulino. "A 0.7 V 256 μW ΣΔ modulator with passive RC integrators achieving 76 dB DR in 2 MHz BW". In: June 2015. DOI: 10.1109/VLSIC.2015.7231294 (cit. on p. 27).

[27] S. Loeda et al. "A 10/20/30/40 MHz Feedforward FIR DAC Continuous-Time ΣΔ ADC With Robust Blocker Performance for Radio Receivers". In: *IEEE Journal of Solid-State Circuits* 51.4 (2016), pp. 860–870. DOI: 10.1109/JSSC.2016.2519395 (cit. on p. 27).

[28] S. Zeller et al. "A 0.039 mm2 inverter-based 1.82 mW 68.6 dB-SNDR 10 MHz-BW CT-ΣΔ -ADC in 65 nm CMOS using power- and area-efficient design techniques". In: *IEEE Journal of Solid-State Circuits* 49 (July 2014), pp. 1548–1560. DOI: 10.1109/JSSC.2014.2321063 (cit. on p. 27).

[29] T. Kim, C. Han, and N. Maghari. "28.2 An 11.4mW 80.4dB-SNDR 15MHz-BW CT delta-sigma modulator using 6b double-noise-shaped quantizer". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 468–469. DOI: 10.1109/ISSCC.2017.7870464 (cit. on p. 27).

[30] T. He et al. "A 50MHZ-BW continuous-time ΣΔ ADC with dynamic error correction achieving 79.8dB SNDR and 95.2dB SFDR". In: *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*. 2018, pp. 230–232. DOI: 10.1109/ISSCC.2018.8310268 (cit. on p. 27).

[31] K. Matsukawa et al. "A 10 MHz BW 50 fJ/conv. continuous time ΣΔ modulator with high-order single opamp integrator using optimization-based design method". In: June 2012, pp. 160–161. ISBN: 978-1-4673-0848-9. DOI: 10.1109/VLSIC.2012.6243839 (cit. on p. 27).

[32] Y.-S. Shu et al. "A 28fJ/conv-step CT ΣΔ modulator with 78dB DR and 18MHz BW in 28nm CMOS using a highly digital multibit quantizer". In: *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*. 2013, pp. 268–269. DOI: 10.1109/ISSCC.2013.6487729 (cit. on p. 27).

[33] S. Pavan et al. "A Power Optimized Continuous-Time ΣΔ ADC for Audio Applications". In: *IEEE Journal of Solid-State Circuits* 43.2 (2008), pp. 351–360. DOI: 10.1109/JSSC.2007.914263 (cit. on p. 27).

[34] P. Malla et al. "A 28mW Spectrum-Sensing Reconfigurable 20MHz 72dB-SNR 70dB-SNDR DT ΣΔ ADC for 802.11n/WiMAX Receivers". In: *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. 2008, pp. 496–631. DOI: 10.1109/ISSCC.2008.4523274 (cit. on p. 28).

[35] L. Bos et al. "A multirate 3.4-to-6.8mW 85-to-66dB DR GSM/bluetooth/UMTS cascade DT ΣΔM in 90nm digital CMOS". In: *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. 2009, 176–177, 177a. DOI: 10.1109/ISSCC.2009.4977365 (cit. on p. 28).

[36] B. R. Carlton et al. "A 32nm, 1.05V, BIST enabled, 10–40MHz, 11-9 bit, 0.13mm2 digitized integrator MASH ΣΔ ADC". In: *2011 Symposium on VLSI Circuits - Digest of Technical Papers*. 2011, pp. 36–37 (cit. on p. 28).

[37] A. Dezzani and E. Andre. "A 1.2-V dual-mode WCDMA/GPRS Sigma Delta modulator". In: *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.* 2003, 58–59 vol.1. DOI: 10.1109/ISSCC.2003.1234206 (cit. on p. 28).

[38] S.-C. Lee and Y. Chiu. "A 15-MHz bandwidth 1-0 MASH ΣΔ ADC with nonlinear memory error calibration achieving 85-dBc SFDR". In: *Solid-State Circuits, IEEE Journal of* 49 (Mar. 2014), pp. 695–707. DOI: 10.1109/JSSC.2014.2304364 (cit. on p. 28).

[39] J. Sauerbrey et al. "A configurable cascaded continuous-time ΣΔ modulator with up to 15MHz bandwidth". In: *2010 Proceedings of ESSCIRC*. 2010, pp. 426–429. DOI: 10.1109/ESSCIRC.2010.5619734 (cit. on p. 28).

[40] D.-Y. Yoon, S. Ho, and H.-S. Lee. "An 85dB-DR 74.6dB-SNDR 50MHZ-BW CT MASH ΣΔ modulator in 28nm CMOS". In: *Digest of Technical Papers - IEEE International Solid-State Circuits Conference* 58 (Mar. 2015), pp. 272–273. DOI: 10.1109/ISSCC.2015.7063031 (cit. on p. 28).

[41] Y. Dong et al. "A Continuous-Time 0–3 MASH ADC Achieving 88 dB DR With 53 MHz BW in 28 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 49.12 (2014), pp. 2868–2877. DOI: 10.1109/JSSC.2014.2364829 (cit. on p. 28).

[42] A. Bafandeh and M. Yavari. "Digital Calibration of Amplifier Finite DC Gain and Gain Bandwidth in MASH ΣΔ Modulators". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 63.4 (2016), pp. 321–325. DOI: 10.1109/TCSII.2015.2504027 (cit. on pp. 28, 29).

[43] M. De Bock et al. "Calibration of DAC Mismatch Errors in ΣΔ ADCs Based on a Sine-Wave Measurement". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 60.9 (2013), pp. 567–571. DOI: 10.1109/TCSII.2013.2268413 (cit. on p. 29).

[44] S.-H. Wu and J.-T. Wu. "A 81-dB Dynamic Range 16-MHz Bandwidth ΣΔ Modulator Using Background Calibration". In: *IEEE Journal of Solid-State Circuits* 48.9 (2013), pp. 2170–2179. DOI: 10.1109/JSSC.2013.2264137 (cit. on p. 29).

[45] A. Pena-Perez, V. Gonzalez-Diaz, and F. Maloberti. "Analog sigma-delta modulation with op-amp gain compensation for nanometer technologies". In: *Analog Integr Circ Sig Process* 76.3 (2013), pp. 297–305. DOI: https://doi.org/10.1007/s10470-013-0058-2 (cit. on p. 29).

[46] G. Cauwenberghs and G. Temes. "Adaptive digital correction of analog errors in MASH ADCs. I. Off-line and blind on-line calibration". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.7 (2000), pp. 621–628. DOI: 10.1109/82.850421 (cit. on p. 29).

[47] P. Kiss et al. "Adaptive digital correction of analog errors in MASH ADCs. II. Correction using test-signal injection". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.7 (2000), pp. 629–638. DOI: 10.1109/82.850422 (cit. on p. 29).

[48] K. A. O'Donoghue, P. J. Hurst, and S. H. Lewis. "A Digitally Corrected 5-mW 2-MS/s SC ΣΔ ADC in 0.25μm CMOS With 94-dB SFDR". In: *IEEE Journal of Solid-State Circuits* 46.11 (2011), pp. 2673–2684. DOI: 10.1109/JSSC.2011.2167369 (cit. on pp. 29, 31).

[49] F. Yang and M. Gani. "An H∞ Approach for Robust Calibration of Cascaded Sigma–Delta Modulators". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 55.2 (2008), pp. 625–634. DOI: 10.1109/TCSI.2007.913704 (cit. on p. 30).

[50] X. Fan, L. Zhang, and C. Zhu. "An Active-gm-RC Structured CMOS Analog Filter with Time Constant Auto-Tuning". In: *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*. 2010, pp. 1–4. DOI: 10.1109/WICOM.2010.5601380 (cit. on pp. 30, 31, 41, 42).

[51] I. Nam et al. "A Fully Differential RC Calibrator for Accurate Cut-off Frequency of a Programmable Channel Selection Filter". In: *JSTS:Journal of Semiconductor Technology and Science* 16 (Oct. 2016), pp. 682–686. URL: https://www.researchgate.net/publication/310506660_A_Fully_Differential_RC_Calibrator_for_Accurate_Cut-off_Frequency_of_a_Programmable_Channel_Selection_Filter (cit. on pp. 30, 31).

[52] A. Edward et al. "A 43-mW MASH 2-2 CT ΣΔ Modulator Attaining 74.4/75.8/76.8 dB of SNDR/SNR/DR and 50 MHz of BW in 40-nm CMOS". In: *IEEE Journal of Solid-State Circuits* 52.2 (2017), pp. 448–459. DOI: 10.1109/JSSC.2016.2616361 (cit. on p. 31).

[53] T.-Y. Lo. "A 102dB dynamic range audio sigma-delta modulator in 40nm CMOS". In: *IEEE Asian Solid-State Circuits Conference 2011*. 2011, pp. 257–260. DOI: 10.1109/ASSCC.2011.6123559 (cit. on p. 31).

[54] L. Breems, R. Rutten, and G. Wetzker. "A cascaded continuous-time ΣΔ Modulator with 67-dB dynamic range in 10-MHz bandwidth". In: *IEEE Journal of Solid-State Circuits* 39.12 (2004), pp. 2152–2160. DOI: 10.1109/JSSC.2004.836245 (cit. on p. 31).

[55] A. Galhardo, J. Goes, and N. Paulino. "Novel linearization technique for low-distortion high-swing CMOS switches with improved reliability". In: June 2006, 4 pp. –2004. DOI: 10.1109/ISCAS.2006.1693006 (cit. on p. 95).

I

DCL Twenty 6-Bit Look-Up Tables

In this first annex, the 6-Bit Look-Up Tables that compose the calibrated DCL are presented. The DCL coefficients are calculated during design phase, considering the previously studied implementation method, which consists of developing 19 LUTs plus a default one. In Tables I.1-I.8, each column corresponds to a finite impulse response filter with inputs $D_1[n-2]$, $D_1[n-1]$, $D_1[n]$, $D_2[n-2]$, $D_2[n-1]$, $D_2[n]$ and illustrates the necessary LUT corresponding to it's respective M range.

Table I.1: Calibrated DCL described as 6-Bit Look-Up Tables, for M=[131;165].

		DCL Output				
Nº	Inputs	M < 137	M=[138;144]	M=[145;151]	M=[152;158]	M=[159;165]
0	000 000	-129	-128	-129	-129	-129
1	000 001	1548	1736	1932	2138	2354
2	000 010	-3589	-3970	-4372	-4793	-5234
3	000 011	-1912	-2106	-2311	-2526	-2751
4	000 100	1656	1852	2056	2270	2495
5	000 101	3333	3716	4117	4537	4978
6	000 110	-1804	-1990	-2187	-2394	-2610
7	000 111	-127	-126	-126	-127	-127
8	001 000	-505	-524	-545	-565	-585
9	001 001	1172	1340	1516	1702	1898
10	001 010	-3965	-4366	-4788	-5229	-5690
11	001 011	-2288	-2502	-2727	-2962	-3207
12	001 100	1280	1456	1640	1834	2039
13	001 101	2957	3320	3701	4101	4522
14	001 110	-2180	-2386	-2603	-2830	-3066
15	001 111	-503	-522	-542	-563	-583
16	010 000	647	688	727	769	809
17	010 001	2324	2552	2788	3036	3292
18	010 010	-2813	-3154	-3516	-3895	-4296
19	010 011	-1136	-1290	-1455	-1628	-1813
20	010 100	2432	2668	2912	3168	3433
21	010 101	4109	4532	4973	5435	5916
22	010 110	-1028	-1174	-1331	-1496	-1672
23	010 111	649	690	730	771	811
24	011 000	271	292	311	333	353
25	011 001	1948	2156	2372	2600	2836

Table I.2: Continued description of the calibrated DCL as 6-Bit Look-Up Tables, for M=[131;165].

		DCL Output				
Nº	Inputs	M < 137	M=[138;144]	M=[145;151]	M=[152;158]	M=[159;165]
26	011 010	-3189	-3550	-3932	-4331	-4752
27	011 011	-1512	-1686	-1871	-2064	-2269
28	011 100	2056	2272	2496	2732	2977
29	011 101	3733	4136	4557	4999	5460
30	011 110	-1404	-1570	-1747	-1932	-2128
31	011 111	273	294	314	335	355
32	100 000	-273	-294	-315	-335	-355
33	100 001	1404	1570	1746	1932	2128
34	100 010	-3733	-4136	-4558	-4999	-5460
35	100 011	-2056	-2272	-2497	-2732	-2977
36	100 100	1512	1686	1870	2064	2269
37	100 101	3189	3550	3931	4331	4752
38	100 110	-1948	-2156	-2373	-2600	-2836
39	100 111	-271	-292	-312	-333	-353
40	101 000	-649	-690	-731	-771	-811
41	101 001	1028	1174	1330	1496	1672
42	101 010	-4109	-4532	-4974	-5435	-5916
43	101 011	-2432	-2668	-2913	-3168	-3433
44	101 100	1136	1290	1454	1628	1813
45	101 101	2813	3154	3515	3895	4296
46	101 110	-2324	-2552	-2789	-3036	-3292
47	101 111	-647	-688	-728	-769	-809
48	110 000	503	522	541	563	583
49	110 001	2180	2386	2602	2830	3066
50	110 010	-2957	-3320	-3702	-4101	-4522
51	110 011	-1280	-1456	-1641	-1834	-2039
52	110 100	2288	2502	2726	2962	3207
53	110 101	3965	4366	4787	5229	5690
54	110 110	-1172	-1340	-1517	-1702	-1898
55	110 111	505	524	544	565	585
56	111 000	127	126	125	127	127
57	111 001	1804	1990	2186	2394	2610
58	111 010	-3333	-3716	-4118	-4537	-4978
59	111 011	-1656	-1852	-2057	-2270	-2495
60	111 100	1912	2106	2310	2526	2751
61	111 101	3589	3970	4371	4793	5234
62	111 110	-1548	-1736	-1933	-2138	-2354
63	111 111	129	128	128	129	129

Table I.3: Calibrated DCL described as 6-Bit Look-Up Tables, for M=[166;200].

		DCL Output				
Nº	Inputs	M=[166;172]	M=[173;179]	M=[180;186]	M=[187;193]	M=[194;200]
0	000 000	-130	-129	-128	-129	-129
1	000 001	2579	2815	3061	3315	3579
2	000 010	-5694	-6173	-6670	-7189	-7726
3	000 011	-2985	-3229	-3481	-3745	-4018
4	000 100	2728	2973	3227	3490	3763
5	000 101	5437	5917	6416	6934	7471
6	000 110	-2836	-3071	-3315	-3570	-3834
7	000 111	-127	-127	-126	-126	-126
8	001 000	-606	-625	-644	-665	-685

Table I.4: Continued description of the calibrated DCL as 6-Bit Look-Up Tables, for M=[166;200].

		DCL Output				
Nº	Inputs	M=[166;172]	M=[173;179]	M=[180;186]	M=[187;193]	M=[194;200]
9	001 001	2103	2319	2545	2779	3023
10	001 010	-6170	-6669	-7186	-7725	-8282
11	001 011	-3461	-3725	-3997	-4281	-4574
12	001 100	2252	2477	2711	2954	3207
13	001 101	4961	5421	5900	6398	6915
14	001 110	-3312	-3567	-3831	-4106	-4390
15	001 111	-603	-623	-642	-662	-682
16	010 000	848	889	930	969	1009
17	010 001	3557	3833	4119	4413	4717
18	010 010	-4716	-5155	-5612	-6091	-6588
19	010 011	-2007	-2211	-2423	-2647	-2880
20	010 100	3706	3991	4285	4588	4901
21	010 101	6415	6935	7474	8032	8609
22	010 110	-1858	-2053	-2257	-2472	-2696
23	010 111	851	891	932	972	1012
24	011 000	372	393	414	433	453
25	011 001	3081	3337	3603	3877	4161
26	011 010	-5192	-5651	-6128	-6627	-7144
27	011 011	-2483	-2707	-2939	-3183	-3436
28	011 100	3230	3495	3769	4052	4345
29	011 101	5939	6439	6958	7496	8053
30	011 110	-2334	-2549	-2773	-3008	-3252
31	011 111	375	395	416	436	456
32	100 000	-376	-395	-416	-437	-457
33	100 001	2333	2549	2773	3007	3251
34	100 010	-5940	-6439	-6958	-7497	-8054
35	100 011	-3231	-3495	-3769	-4053	-4346
36	100 100	2482	2707	2939	3182	3435
37	100 101	5191	5651	6128	6626	7143
38	100 110	-3082	-3337	-3603	-3878	-4162
39	100 111	-373	-393	-414	-434	-454
40	101 000	-852	-891	-932	-973	-1013
41	101 001	1857	2053	2257	2471	2695
42	101 010	-6416	-6935	-7474	-8033	-8610
43	101 011	-3707	-3991	-4285	-4589	-4902
44	101 100	2006	2211	2423	2646	2879
45	101 101	4715	5155	5612	6090	6587
46	101 110	-3558	-3833	-4119	-4414	-4718
47	101 111	-849	-889	-930	-970	-1010
48	110 000	602	623	642	661	681
49	110 001	3311	3567	3831	4105	4389
50	110 010	-4962	-5421	-5900	-6399	-6916
51	110 011	-2253	-2477	-2711	-2955	-3208
52	110 100	3460	3725	3997	4280	4573
53	110 101	6169	6669	7186	7724	8281
54	110 110	-2104	-2319	-2545	-2780	-3024
55	110 111	605	625	644	664	684
56	111 000	126	127	126	125	125
57	111 001	2835	3071	3315	3569	3833
58	111 010	-5438	-5917	-6416	-6935	-7472
59	111 011	-2729	-2973	-3227	-3491	-3764
60	111 100	2984	3229	3481	3744	4017
61	111 101	5693	6173	6670	7188	7725
62	111 110	-2580	-2815	-3061	-3316	-3580
63	111 111	129	129	128	128	128

Table I.5: Calibrated DCL described as 6-Bit Look-Up Tables, for M=[201;235].

		DCL Output				
Nº	Inputs	M=[201;207]	M=[208;214]	M=[215;221]	M=[222;228]	M=[229;235]
0	000 000	-129	-129	-128	-130	-130
1	000 001	3852	4135	4428	4728	5039
2	000 010	-8282	-8857	-9451	-10066	-10698
3	000 011	-4301	-4593	-4895	-5208	-5529
4	000 100	4046	4338	4641	4951	5272
5	000 101	8027	8602	9197	9809	10441
6	000 110	-4107	-4390	-4682	-4985	-5296
7	000 111	-126	-126	-126	-127	-127
8	001 000	-705	-725	-744	-764	-784
9	001 001	3276	3539	3812	4094	4385
10	001 010	-8858	-9453	-10067	-10700	-11352
11	001 011	-4877	-5189	-5511	-5842	-6183
12	001 100	3470	3742	4025	4317	4618
13	001 101	7451	8006	8581	9175	9787
14	001 110	-4683	-4986	-5298	-5619	-5950
15	001 111	-702	-722	-742	-761	-781
16	010 000	1049	1089	1130	1168	1208
17	010 001	5030	5353	5686	6026	6377
18	010 010	-7104	-7639	-8193	-8768	-9360
19	010 011	-3123	-3375	-3637	-3910	-4191
20	010 100	5224	5556	5899	6249	6610
21	010 101	9205	9820	10455	11107	11779
22	010 110	-2929	-3172	-3424	-3687	-3958
23	010 111	1052	1092	1132	1171	1211
24	011 000	473	493	514	534	554
25	011 001	4454	4757	5070	5392	5723
26	011 010	-7680	-8235	-8809	-9402	-10014
27	011 011	-3699	-3971	-4253	-4544	-4845
28	011 100	4648	4960	5283	5615	5956
29	011 101	8629	9224	9839	10473	11125
30	011 110	-3505	-3768	-4040	-4321	-4612
31	011 111	476	496	516	537	557
32	100 000	-477	-497	-516	-538	-558
33	100 001	3504	3767	4040	4320	4611
34	100 010	-8630	-9225	-9839	-10474	-11126
35	100 011	-4649	-4961	-5283	-5616	-5957
36	100 100	3698	3970	4253	4543	4844
37	100 101	7679	8234	8809	9401	10013
38	100 110	-4455	-4758	-5070	-5393	-5724
39	100 111	-474	-494	-514	-535	-555
40	101 000	-1053	-1093	-1132	-1172	-1212
41	101 001	2928	3171	3424	3686	3957
42	101 010	-9206	-9821	-10455	-11108	-11780
43	101 011	-5225	-5557	-5899	-6250	-6611
44	101 100	3122	3374	3637	3909	4190
45	101 101	7103	7638	8193	8767	9359
46	101 110	-5031	-5354	-5686	-6027	-6378
47	101 111	-1050	-1090	-1130	-1169	-1209
48	110 000	701	721	742	760	780
49	110 001	4682	4985	5298	5618	5949
50	110 010	-7452	-8007	-8581	-9176	-9788
51	110 011	-3471	-3743	-4025	-4318	-4619
52	110 100	4876	5188	5511	5841	6182

Table I.6: Continued description of the calibrated DCL as 6-Bit Look-Up Tables, for M=[201;235].

		DCL Output				
Nº	Inputs	M=[201;207]	M=[208;214]	M=[215;221]	M=[222;228]	M=[229;235]
53	110 101	8857	9452	10067	10699	11351
54	110 110	-3277	-3540	-3812	-4095	-4386
55	110 111	704	724	744	763	783
56	111 000	125	125	126	126	126
57	111 001	4106	4389	4682	4984	5295
58	111 010	-8028	-8603	-9197	-9810	-10442
59	111 011	-4047	-4339	-4641	-4952	-5273
60	111 100	4300	4592	4895	5207	5528
61	111 101	8281	8856	9451	10065	10697
62	111 110	-3853	-4136	-4428	-4729	-5040
63	111 111	128	128	128	129	129

Table I.7: Calibrated DCL described as 6-Bit Look-Up Tables, for M=[236;263], plus the default tables.

		DCL Output				
Nº	Inputs	M=[236;242]	M=[243;249]	M=[250;256]	M=[257;263]	default table
0	000 000	-129	-129	-129	-129	-128
1	000 001	5360	5690	6029	6378	2161
2	000 010	-11348	-12018	-12707	-13415	-4821
3	000 011	-5859	-6199	-6549	-6908	-2532
4	000 100	5604	5944	6294	6653	2277
5	000 101	11093	11763	12452	13160	4566
6	000 110	-5615	-5945	-6284	-6633	-2416
7	000 111	-126	-126	-126	-126	-127
8	001 000	-803	-823	-843	-861	-560
9	001 001	4686	4996	5315	5646	1729
10	001 010	-12022	-12712	-13421	-14147	-5253
11	001 011	-6533	-6893	-7263	-7640	-2964
12	001 100	4930	5250	5580	5921	1845
13	001 101	10419	11069	11738	12428	4134
14	001 110	-6289	-6639	-6998	-7365	-2848
15	001 111	-800	-820	-840	-858	-559
16	010 000	1247	1287	1327	1365	756
17	010 001	6736	7106	7485	7872	3045
18	010 010	-9972	-10602	-11251	-11921	-3937
19	010 011	-4483	-4783	-5093	-5414	-1648
20	010 100	6980	7360	7750	8147	3161
21	010 101	12469	13179	13908	14654	5450
22	010 110	-4239	-4529	-4828	-5139	-1532
23	010 111	1250	1290	1330	1368	757
24	011 000	573	593	613	633	324
25	011 001	6062	6412	6771	7140	2613

Table I.8: Continued description of the calibrated DCL as 6-Bit Look-Up Tables, for M=[236;263], plus the default tables.

		DCL Output				
Nº	Inputs	M=[236;242]	M=[243;249]	M=[250;256]	M=[257;263]	default table
26	011 010	-10646	-11296	-11965	-12653	-4369
27	011 011	-5157	-5477	-5807	-6146	-2080
28	011 100	6306	6666	7036	7415	2729
29	011 101	11795	12485	13194	13922	5018
30	011 110	-4913	-5223	-5542	-5871	-1964
31	011 111	576	596	616	636	325
32	100 000	-577	-597	-617	-637	-326
33	100 001	4912	5222	5541	5870	1963
34	100 010	-11796	-12486	-13195	-13923	-5019
35	100 011	-6307	-6667	-7037	-7416	-2730
36	100 100	5156	5476	5806	6145	2079
37	100 101	10645	11295	11964	12652	4368
38	100 110	-6063	-6413	-6772	-7141	-2614
39	100 111	-574	-594	-614	-634	-325
40	101 000	-1251	-1291	-1331	-1369	-758
41	101 001	4238	4528	4827	5138	1531
42	101 010	-12470	-13180	-13909	-14655	-5451
43	101 011	-6981	-7361	-7751	-8148	-3162
44	101 100	4482	4782	5092	5413	1647
45	101 101	9971	10601	11250	11920	3936
46	101 110	-6737	-7107	-7486	-7873	-3046
47	101 111	-1248	-1288	-1328	-1366	-757
48	110 000	799	819	839	857	558
49	110 001	6288	6638	6997	7364	2847
50	110 010	-10420	-11070	-11739	-12429	-4135
51	110 011	-4931	-5251	-5581	-5922	-1846
52	110 100	6532	6892	7262	7639	2963
53	110 101	12021	12711	13420	14146	5252
54	110 110	-4687	-4997	-5316	-5647	-1730
55	110 111	802	822	842	860	559
56	111 000	125	125	125	125	126
57	111 001	5614	5944	6283	6632	2415
58	111 010	-11094	-11764	-12453	-13161	-4567
59	111 011	-5605	-5945	-6295	-6654	-2278
60	111 100	5858	6198	6548	6907	2531
61	111 101	11347	12017	12706	13414	4820
62	111 110	-5361	-5691	-6030	-6379	-2162
63	111 111	128	128	128	128	127

II

VerilogA

II.1 VerilogA Code for SAR Sub-Binary Digital Counter

```
//================================
// Description of the block
//================================
// This block receives the signal of a comparator, the clock and an enable
//(RESETN)
//
// It's is to be able to calculate the ideal value of alpha,
// knowing M=1/alpha, and the values of R, C and Ts. The comparator sends a
// signal and if comp=1 the value of M decreases, if not, it increases, the
// values of M change according to an RC_Var, previously studeid and the last
// value of M obtained is the closest to the ideal 1/alpha, this values should
// be sent to the DCL.
//
// Therefore, this block has as output the CLKcomp that triggers the
// comparator. The RESET_C to reset the capacitor of the circuit, the
// M_CODE, that sends the value of M in binary. (9 bits); VREF_CAL
// code (3 bits in binary - 8 bits in one-hot logic )that will calibrate the
// reference voltage and CAL_DONE that indicates when the calibration is over.
//================================

`include "constants.vams"
`include "disciplines.vams"

// Defining constants:
`define NUM_STEPS 32
`define RESET_COUNTER 40
`define NUM_BITS 9
`define ONE_HOT_BITS 8

// initialization of the block:

module ALPHA_CAL_COUNTER ( AVDD, AVSS, COMP, CLK, OM_B1, OM_B2, RESET_C,
    RESETN, CLKcomp, M_CODE, CAL_DONE, VREF_CAL);

```

```
34 //================================
35 // I/O
36 //================================
37
38 input       CLK;
39 electrical  CLK;
40
41 input         OM_B1;
42 electrical  OM_B1;
43
44 input       OM_B2;
45 electrical  OM_B2;
46
47 input       AVDD;
48 electrical  AVDD;
49
50 input       AVSS;
51 electrical  AVSS;
52
53 input       COMP;
54 electrical  COMP;
55
56 input       RESETN;
57 electrical  RESETN;
58
59 output      CLKcomp;
60 electrical  CLKcomp;
61
62 output      CAL_DONE;
63 electrical  CAL_DONE;
64
65 output      RESET_C;
66 electrical  RESET_C;
67
68 output      [`NUM_BITS-1:0] M_CODE;
69 electrical  [`NUM_BITS-1:0] M_CODE;
70
71 output      [`ONE_HOT_BITS-1:0] VREF_CAL;
72 electrical  [`ONE_HOT_BITS-1:0] VREF_CAL;
73
74 //================================
75 // Block parameters
76 //================================
77
78 parameter integer   Rm    = 23952;  //2994*8
79 parameter integer   R_factor  = 8;
80 parameter real      Cm    = 4.9952970e-12;
81 parameter real      Ts    = 6.2500e-10;
82 parameter real      vth   = 0.4;
83 parameter real      tdelay  = 100p;
```

```
84 parameter real      trise = 20p;
85 parameter real      tfall = 20p;
86 parameter real      vcm   = 500m;
87 parameter real      vtol    = 0.1;
88 parameter integer   M_ideal = 191;
89
90 //================================
91 // Internal signals declaration
92 //================================
93
94 integer   M;
95 integer   enable, stop;
96 integer   latch;
97 integer   latch_en, latch_en_next;
98 integer   rst_c;
99 integer   reset_values;
100 real      ttol;
101 integer   on, stand_by, active;
102 integer   state, state_next;
103 integer   comp;
104 integer   value_of_counter;
105 integer   last_m;
106 integer   cal_busy;
107 integer   cal_done;
108 integer   counter;     // counts clock cycles
109 integer   step_count;   // counts steps
110
111 // Defining sub-binary steps:
112 integer RC_var [`NUM_STEPS - 1 :0]='{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
     1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 5, 8, 12, 18, 29, 39, 42};
113 integer     meas [`NUM_STEPS-1:0];
114 integer     M_code[`NUM_BITS-1:0];
115 integer     vref_code[`ONE_HOT_BITS-1:0]='{0, 0, 0, 0, 0, 0, 0, 0};
116
117 analog begin
118
119   @ (initial_step) begin
120
121     ttol=trise/10;
122     counter=0;
123     step_count=0;
124     stop=0;
125     state=0;
126     state_next=0;
127     latch_en = 0;
128     latch_en_next=0;
129     M=M_ideal;
130     last_m=M_ideal;
131     rst_c=1;
132     cal_busy=0; // 0
```

```
133     cal_done=0;
134     reset_values=0;
135   end
136 
137 // Defining events for falling edge of clock signal
138   @ (cross ( V(CLK) - vth,  -1,ttol,vtol)) begin
139     counter=counter+1;
140     latch_en = latch_en_next;
141     state = state_next;
142   end
143 
144 // convert voltage to logic value (0/1)
145   on = V(OM_B1) > vth ? 1:0;
146   stand_by = V(OM_B2) > vth ? 1:0;
147   reset_values= V(RESETN) > vth ? 1:0;
148 // Defining events for falling edge of OM signal
149   @ ( cross( V(OM_B1) - vth, -1,ttol,vtol )) begin
150     state=0;
151   end
152 
153   @ ( cross( V(OM_B2) - vth, -1,ttol,vtol )) begin
154     step_count=0;
155   end
156 
157 // ============ DEFINING DEFAULT VALUES ========================
158 
159   if(!reset_values) begin
160     ttol=trise/10;
161     counter=0;
162     step_count=0;
163     stop=0;
164     state=0;
165     state_next=0;
166     latch_en = 0;
167     latch_en_next=0;
168     M=M_ideal;
169     last_m=M_ideal;
170     rst_c=1;
171     cal_busy=0;
172     cal_done=0;
173     vref_code[0]=0; vref_code[1]=0; vref_code[2]=0; vref_code[3]=1; vref_code
      [4]=0; vref_code[5]=0; vref_code[6]=0; vref_code[7]=0; // Default VREF code
174   end
175 
176 // ============ DEFINING DIFFERENT STATES ========================
177 
178   case(state)
179 
180   0: begin        // state OFF
181 
```

```
182     latch_en_next=0;
183     state_next=0;
184     stop=1;
185     counter=0;
186     active=0;
187     M=M_ideal;
188     rst_c=1;
189 
190       if(!cal_busy) begin
191 
192           if(last_m<148) begin vref_code[0]=1; vref_code[1]=0; vref_code[2]=0;
     vref_code[3]=0; vref_code[4]=0; vref_code[5]=0; vref_code[6]=0; vref_code
    [7]=0; end
193         else if(last_m>= 148 && last_m<165) begin vref_code[0]=0; vref_code
    [1]=1; vref_code[2]=0; vref_code[3]=0; vref_code[4]=0; vref_code[5]=0;
    vref_code[6]=0; vref_code[7]=0; end
194         else if(last_m>= 165 && last_m<182) begin vref_code[0]=0; vref_code
    [1]=0; vref_code[2]=1; vref_code[3]=0; vref_code[4]=0; vref_code[5]=0;
    vref_code[6]=0; vref_code[7]=0; end
195         else if(last_m>= 182 && last_m<199) begin vref_code[0]=0; vref_code
    [1]=0; vref_code[2]=0; vref_code[3]=1; vref_code[4]=0; vref_code[5]=0;
    vref_code[6]=0; vref_code[7]=0; end
196         else if(last_m>= 199 && last_m<216) begin vref_code[0]=0; vref_code
    [1]=0; vref_code[2]=0; vref_code[3]=0; vref_code[4]=1; vref_code[5]=0;
    vref_code[6]=0; vref_code[7]=0; end
197         else if(last_m>= 216 && last_m<233) begin vref_code[0]=0; vref_code
    [1]=0; vref_code[2]=0; vref_code[3]=0; vref_code[4]=0; vref_code[5]=1;
    vref_code[6]=0; vref_code[7]=0; end
198         else if(last_m>= 233 && last_m<250) begin vref_code[0]=0; vref_code
    [1]=0; vref_code[2]=0; vref_code[3]=0; vref_code[4]=0; vref_code[5]=0;
    vref_code[6]=1; vref_code[7]=0; end
199         else if(last_m>= 250) begin vref_code[0]=0; vref_code[1]=0; vref_code
    [2]=0; vref_code[3]=0; vref_code[4]=0; vref_code[5]=0; vref_code[6]=0;
    vref_code[7]=1; end
200       end
201 
202     if((on==1 && step_count!=`NUM_STEPS) && stand_by==0 )begin
203       cal_done=0;
204       state_next=1;
205     end
206   end
207   1: begin      // state stand_by
208 
209     counter=0;
210     if(on==1 && stand_by==1) active=1;
211     if(active==1) begin
212       rst_c=0;
213       stop=0;
214       step_count=0;
215       cal_busy=1;
```

```
216       state_next=2;
217     end
218   end
219   2: begin      // state ON (settling_RC)
220
221     if(counter==(M-3))  latch_en_next=1;
222     if(counter==(M-2))  latch_en_next=0;
223     if((counter==M) && (!stop)) begin
224
225       value_of_counter=counter;
226       comp = V(COMP) > vth ? 1:0;
227       if (comp==1) begin
228         M=M-RC_var[step_count];
229       end
230       if (comp!=1)begin
231         M=M+RC_var[step_count];
232       end
233       meas[step_count]=M;
234       step_count=step_count+1;
235       counter=0;
236       rst_c=1;
237       state_next=3;
238     end
239   end
240   3: begin      // state ON (RESET_Capacitor)
241
242     latch_en_next=0;
243
244     if ((counter)==`RESET_COUNTER) begin
245
246       value_of_counter=counter;
247       if(step_count ==(`NUM_STEPS)) begin // calibration is over -> goes to
    state OFF
248         stop=1;
249         last_m=M;
250         cal_busy=0;
251         rst_c=1;
252         counter=0;
253         cal_done=1;
254         state_next=0;
255       end
256       if(step_count !=(`NUM_STEPS)) begin // calibration is not over ->
    settling
257         rst_c=0;
258         state_next=2;
259         counter=0;
260       end
261     end
262   end
263   endcase
```

```
264 //Converting internal signals to output signals
265   latch = (V(CLK) > vth) & latch_en;
266   V(CLKcomp) <+ V(AVDD)*transition (latch, tdelay, trise, tfall);
267   V(RESET_C)  <+ V(AVDD)*transition (rst_c, tdelay, trise, tfall);
268   V(CAL_DONE)   <+ V(AVDD)*transition (cal_done, tdelay, trise, tfall);
269
270 //Converting M value (integer) to a binary number with precision of 9 bits
271   generate i (`NUM_BITS-1, 0)
272     M_code[i] = (last_m & (1 << i)) ? 1:0;
273 // Generating digital output signal with precision of 9 bits
274   generate i (`NUM_BITS - 1, 0)
275    V(M_CODE[i])   <+ V(AVDD)*transition(M_code[i], tdelay, trise, tfall);
276
277 //Generating a digital output signal with precision of 8 bits to calibrate
278 //Vref through the digital resistive divider
279   generate i (`ONE_HOT_BITS - 1, 0)
280    V(VREF_CAL[i])   <+ V(AVDD)*transition(vref_code[i], tdelay, trise, tfall);
281 end
282 endmodule
```

Listing II.1: Implementation of the SAR sub-binary digital counter in VerilogA.

II.2 VerilogA Code for Calibrated Digital Cancellation Logic

```
1  //================================
2  // Description of the block
3  //================================
4  // This block receives the clock signal, an enable (RESETN),
5  // the value of the last M measured by the RC measuring ciruit
6  // (M_CODE with precision of 9 bits) the outputs of the 1st and 2nd
7  // order Modulators of the MASH (D1 and D2), and has the output DOUT.
8  //
9  // The DCL has 20 6-Bit LUT registered which contain
10 // the necessary coefficients previously studied in design phase.
11 // The purpose of this DCL is to receive the M value which is used as
12 // an index in order to select the correct LUT according to the RC variation
13 // that the MASH is being submitted to, which reflects
14 // process variation. When the correct LUT is select it's respective
15 // coefficients are saved in var DoutLogicrMux [63:0] which will be
16 // decoded in order to obtain DOUT.
17 //================================
18
19 `include "constants.vams"
20 `include "disciplines.vams"
21
22 // defining constant:
23 `define NUM_BITS_M    9
24
25 module DCL_CAL (CLK, RESETN, M_CODE, D1, D2, DOUT);
26
```

```
27 //================================
28 // I/O
29 //================================
30
31 input         CLK;
32 electrical    CLK;
33
34 input         RESETN;
35 electrical    RESETN;
36
37 input         [`NUM_BITS_M -1:0] M_CODE;
38 electrical    [`NUM_BITS_M -1:0] M_CODE;
39
40 input         D1;
41 electrical    D1;
42
43 input         D2;
44 electrical    D2;
45
46 output        DOUT;
47 electrical    DOUT;
48
49 //================================
50 // PARAMETERS
51 //================================
52
53 parameter real    tdelay      = 1e-9 ;    //delay [s] on output signals
54 parameter real    trise     = 6e-11 ;     //rise time [s] on output signals
55 parameter real    tfall     = 6e-11 ;     //fall time [s] on output signals
56 parameter real    vth       = 0.45 ;    //threshold voltage [V] when
       converting input voltage to bit value
57
58 // defining a vector that will save the coefficients
59 // from the selected table
60 integer DoutLogicrMux [63:0];
61
62 //================================
63 // DEFINING THE 20 6-BIT LUTs
64 //================================
65
66 integer DoutLogicrMux1 [63:0] =
       '{129,-1548,3589,1912,-1656,-3333,1804,127,505,-1172,3965,2288,
67 -1280,-2957,2180,503,-647,-2324,2813,1136,-2432,-4109,1028,-649,-271,-1948,
68 3189,1512,-2056,-3733,1404,-273,273,-1404,3733,2056,-1512,-3189,1948,271,649,
69 -1028,4109,2432,-1136,-2813,2324,647,-503,-2180,2957,1280,-2288,-3965,1172,
70 -505,-127,-1804,3333,1656,-1912,-3589,1548,-129};
71 integer DoutLogicrMux2 [63:0] =
       '{128,-1736,3970,2106,-1852,-3716,1990,126,524,-1340,4366,2502
72 ,-1456,-3320,2386,522,-688,-2552,3154,1290,-2668,-4532,1174,-690,-292,-2156,
73 3550,1686,-2272,-4136,1570,-294,294,-1570,4136,2272,-1686,-3550,2156,292,690,
```

```
-1174,4532,2668,-1290,-3154,2552,688,-522,-2386,3320,1456,-2502,-4366,1340,
-524,-126,-1990,3716,1852,-2106,-3970,1736,-128};
integer DoutLogicrMux3 [63:0] =
    '{128,-1933,4371,2310,-2057,-4118,2186,125,544,-1517,4787,2726
,-1641,-3702,2602,541,-728,-2789,3515,1454,-2913,-4974,1330,-731,-312,-2373,
3931,1870,-2497,-4558,1746,-315,314,-1747,4557,2496,-1871,-3932,2372,311,
730,-1331,4973,2912,-1455,-3516,2788,727,-542,-2603,3701,1640,-2727,-4788,
1516,-545,-126,-2187,4117,2056,-2311,-4372,1932,-129};
integer DoutLogicrMux4 [63:0] =
    '{129,-2138,4793,2526,-2270,-4537,2394,127,565,-1702,5229,2962
,-1834,-4101,2830,563,-769,-3036,3895,1628,-3168,-5435,1496,-771,-333,-2600,
4331,2064,-2732,-4999,1932,-335,335,-1932,4999,2732,-2064,-4331,2600,333,771,
-1496,5435,3168,-1628,-3895,3036,769,-563,-2830,4101,1834,-2962,-5229,1702,
-565,-127,-2394,4537,2270,-2526,-4793,2138,-129};
integer DoutLogicrMux5 [63:0] =
    '{129,-2354,5234,2751,-2495,-4978,2610,127,585,-1898,5690,3207,
-2039,-4522,3066,583,-809,-3292,4296,1813,-3433,-5916,1672,-811,-353,-2836,
4752,2269,-2977,-5460,2128,-355,355,-2128,5460,2977,-2269,-4752,2836,353,811,
-1672,5916,3433,-1813,-4296,3292,809,-583,-3066,4522,2039,-3207,-5690,1898,
-585,-127,-2610,4978,2495,-2751,-5234,2354,-129};
integer DoutLogicrMux6 [63:0] =
    '{129,-2580,5693,2984,-2729,-5438,2835,126,605,-2104,6169,3460,
-2253,-4962,3311,602,-849,-3558,4715,2006,-3707,-6416,1857,-852,-373,-3082,
5191,2482,-3231,-5940,2333,-376,375,-2334,5939,3230,-2483,-5192,3081,372,
851,-1858,6415,3706,-2007,-4716,3557,848,-603,-3312,4961,2252,-3461,-6170,
2103,-606,-127,-2836,5437,2728,-2985,-5694,2579,-130};
integer DoutLogicrMux7 [63:0] =
    '{129,-2815,6173,3229,-2973,-5917,3071,127,625,-2319,6669,3725,
-2477,-5421,3567,623,-889,-3833,5155,2211,-3991,-6935,2053,-891,-393,-3337,
5651,2707,-3495,-6439,2549,-395,395,-2549,6439,3495,-2707,-5651,3337,393,891,
-2053,6935,3991,-2211,-5155,3833,889,-623,-3567,5421,2477,-3725,-6669,2319,
-625,-127,-3071,5917,2973,-3229,-6173,2815,-129};
integer DoutLogicrMux8 [63:0] =
    '{128,-3061,6670,3481,-3227,-6416,3315,126,644,-2545,7186,3997,
-2711,-5900,3831,642,-930,-4119,5612,2423,-4285,-7474,2257,-932,-414,-3603,
6128,2939,-3769,-6958,2773,-416,416,-2773,6958,3769,-2939,-6128,3603,414,932,
-2257,7474,4285,-2423,-5612,4119,930,-642,-3831,5900,2711,-3997,-7186,2545,
-644,-126,-3315,6416,3227,-3481,-6670,3061,-128};
integer DoutLogicrMux9 [63:0] =
    '{128,-3316,7188,3744,-3491,-6935,3569,125,664,-2780,7724,4280,
-2955,-6399,4105,661,-970,-4414,6090,2646,-4589,-8033,2471,-973,-434,-3878,
6626,3182,-4053,-7497,3007,-437,436,-3008,7496,4052,-3183,-6627,3877,433,972,
-2472,8032,4588,-2647,-6091,4413,969,-662,-4106,6398,2954,-4281,-7725,2779,
-665,-126,-3570,6934,3490,-3745,-7189,3315,-129};
integer DoutLogicrMux10 [63:0] =
    '{128,-3580,7725,4017,-3764,-7472,3833,125,684,-3024,8281,4573,
-3208,-6916,4389,681,-1010,-4718,6587,2879,-4902,-8610,2695,-1013,-454,-4162,
7143,3435,-4346,-8054,3251,-457,456,-3252,8053,4345,-3436,-7144,4161,453,1012,
-2696,8609,4901,-2880,-6588,4717,1009,-682,-4390,6915,3207,-4574,-8282,3023,
-685,-126,-3834,7471,3763,-4018,-7726,3579,-129};
```

```
integer DoutLogicrMux11 [63:0] =
    '{128,-3853,8281,4300,-4047,-8028,4106,125,704,-3277,8857,4876,
-3471,-7452,4682,701,-1050,-5031,7103,3122,-5225,-9206,2928,-1053,-474,-4455,
7679,3698,-4649,-8630,3504,-477,476,-3505,8629,4648,-3699,-7680,4454,473,1052,
-2929,9205,5224,-3123,-7104,5030,1049,-702,-4683,7451,3470,-4877,-8858,3276,
-705,-126,-4107,8027,4046,-4301,-8282,3852,-129};
integer DoutLogicrMux12 [63:0] =
    '{128,-4136,8856,4592,-4339,-8603,4389,125,724,-3540,9452,5188,
-3743,-8007,4985,721,-1090,-5354,7638,3374,-5557,-9821,3171,-1093,-494,-4758,
8234,3970,-4961,-9225,3767,-497,496,-3768,9224,4960,-3971,-8235,4757,493,1092,
-3172,9820,5556,-3375,-7639,5353,1089,-722,-4986,8006,3742,-5189,-9453,3539,
-725,-126,-4390,8602,4338,-4593,-8857,4135,-129};
integer DoutLogicrMux13 [63:0] =
    '{128,-4428,9451,4895,-4641,-9197,4682,126,744,-3812,10067,5511,
-4025,-8581,5298,742,-1130,-5686,8193,3637,-5899,-10455,3424,-1132,-514,-5070,
8809,4253,-5283,-9839,4040,-516,516,-4040,9839,5283,-4253,-8809,5070,514,1132,
-3424,10455,5899,-3637,-8193,5686,1130,-742,-5298,8581,4025,-5511,-10067,3812,
-744,-126,-4682,9197,4641,-4895,-9451,4428,-128};
integer DoutLogicrMux14 [63:0] =
    '{129,-4729,10065,5207,-4952,-9810,4984,126,763,-4095,10699,5841,
-4318,-9176,5618,760,-1169,-6027,8767,3909,-6250,-11108,3686,-1172,-535,-5393,
9401,4543,-5616,-10474,4320,-538,537,-4321,10473,5615,-4544,-9402,5392,534,
1171,-3687,11107,6249,-3910,-8768,6026,1168,-761,-5619,9175,4317,-5842,
-10700,4094,-764,-127,-4985,9809,4951,-5208,-10066,4728,-130};
integer DoutLogicrMux15 [63:0] =
    '{129,-5040,10697,5528,-5273,-10442,5295,126,783,-4386,11351,6182,
-4619,-9788,5949,780,-1209,-6378,9359,4190,-6611,-11780,3957,-1212,-555,-5724,
10013,4844,-5957,-11126,4611,-558,557,-4612,11125,5956,-4845,-10014,5723,554,
1211,-3958,11779,6610,-4191,-9360,6377,1208,-781,-5950,9787,4618,-6183,-11352,
4385,-784,-127,-5296,10441,5272,-5529,-10698,5039,-130};
integer DoutLogicrMux16 [63:0] =
    '{128,-5361,11347,5858,-5605,-11094,5614,125,802,-4687,12021,6532,
-4931,-10420,6288,799,-1248,-6737,9971,4482,-6981,-12470,4238,-1251,-574,
-6063,10645,5156,-6307,-11796,4912,-577,576,-4913,11795,6306,-5157,-10646,
6062,573,1250,-4239,12469,6980,-4483,-9972,6736,1247,-800,-6289,10419,4930,
-6533,-12022,4686,-803,-126,-5615,11093,5604,-5859,-11348,5360,-129};
integer DoutLogicrMux17 [63:0] =
    '{128,-5691,12017,6198,-5945,-11764,5944,125,822,-4997,12711,6892,
-5251,-11070,6638,819,-1288,-7107,10601,4782,-7361,-13180,4528,-1291,-594,
-6413,11295,5476,-6667,-12486,5222,-597,596,-5223,12485,6666,-5477,-11296,
6412,593,1290,-4529,13179,7360,-4783,-10602,7106,1287,-820,-6639,11069,
5250,-6893,-12712,4996,-823,-126,-5945,11763,5944,-6199,-12018,5690,-129};
integer DoutLogicrMux18 [63:0] =
    '{128,-6030,12706,6548,-6295,-12453,6283,125,842,-5316,13420,7262,
-5581,-11739,6997,839,-1328,-7486,11250,5092,-7751,-13909,4827,-1331,-614,
-6772,11964,5806,-7037,-13195,5541,-617,616,-5542,13194,7036,-5807,-11965,
6771,613,1330,-4828,13908,7750,-5093,-11251,7485,1327,-840,-6998,11738,
5580,-7263,-13421,5315,-843,-126,-6284,12452,6294,-6549,-12707,6029,-129};
integer DoutLogicrMux19 [63:0] =
    '{128,-6379,13414,6907,-6654,-13161,6632,125,860,-5647,14146,7639,
```

```
-5922,-12429,7364,857,-1366,-7873,11920,5413,-8148,-14655,5138,-1369,-634,
-7141,12652,6145,-7416,-13923,5870,-637,636,-5871,13922,7415,-6146,-12653,
7140,633,1368,-5139,14654,8147,-5414,-11921,7872,1365,-858,-7365,12428,5921,
-7640,-14147,5646,-861,-126,-6633,13160,6653,-6908,-13415,6378,-129};
integer DoutLogicrMux20 [63:0] =
    '{127,-2162,4820,2531,-2278,-4567,2415,126,559,-1730,5252,2963,-1846,
-4135,2847,558,-757,-3046,3936,1647,-3162,-5451,1531,-758,-325,-2614,4368,
2079,-2730,-5019,1963,-326,325,-1964,5018,2729,-2080,-4369,2613,324,757,
-1532,5450,3161,-1648,-3937,3045,756,-559,-2848,4134,1845,-2964,-5253,1729,
-560,-127,-2416,4566,2277,-2532,-4821,2161,-128};

//================================

integer j;
integer d1d2 [0:5] ;
integer d1;
integer d1_r;
integer d1_rr;
integer d2;
integer d2_r;
integer d2_rr;
integer out;
integer M=0;
integer M_code[`NUM_BITS_M-1:0];

analog begin

   @ (initial_step) begin
      d1_rr = 0;
      d1_r = 0;
      d1 = 0;
  end

   @(cross(V(RESETN)-vth, -1)) begin

   end

   // convert voltage signal to logic value
  generate i (`NUM_BITS_M - 1, 0) begin
    M_code[i] = (( V(M_CODE[i]) > vth ) ? 1 : 0 );
  end

  // convert binary to decimal
  generate i (`NUM_BITS_M - 1, 0) begin
    M = M + (M_code[i] ? 1 << i : 0 );
    end

//========CHOOSING THE NECESSARY TABLE========

  if(M <= 137) begin
```

```
206     for(j=0; j<64; j=j+1) begin
207             DoutLogicrMux[j] = DoutLogicrMux1[j];
208     end
209     end
210     if(M>= 138 && M<= 144) begin
211        for(j=0; j<64; j=j+1) begin
212             DoutLogicrMux[j] = DoutLogicrMux2[j];
213     end
214     end
215     if(M>= 145 && M<= 151) begin
216         for(j=0; j<64; j=j+1) begin
217             DoutLogicrMux[j] = DoutLogicrMux3[j];
218     end
219     end
220     if(M>= 152 && M<= 158) begin
221         for(j=0; j<64; j=j+1) begin
222             DoutLogicrMux[j] = DoutLogicrMux4[j];
223     end
224     end
225     if(M>= 159 && M<= 165) begin
226         for(j=0; j<64; j=j+1) begin
227             DoutLogicrMux[j] = DoutLogicrMux5[j];
228     end
229     end
230     if(M>= 166 && M<= 172) begin
231         for(j=0; j<64; j=j+1) begin
232             DoutLogicrMux[j] = DoutLogicrMux6[j];
233     end
234     end
235     if(M>= 173 && M<= 179) begin
236         for(j=0; j<64; j=j+1) begin
237             DoutLogicrMux[j] = DoutLogicrMux7[j];
238     end
239     end
240     if(M>= 180 && M<= 186) begin
241         for(j=0; j<64; j=j+1) begin
242             DoutLogicrMux[j] = DoutLogicrMux8[j];
243     end
244     end
245     if(M>= 187 && M<= 193) begin
246 
247         if(M==191) begin
248             for(j=0; j<64; j=j+1) begin
249               DoutLogicrMux[j] = DoutLogicrMux20[j];
250         //$display("%d", DoutLogicrMux[j]);
251       end
252           end else begin
253             for(j=0; j<64; j=j+1) begin
254               DoutLogicrMux[j] = DoutLogicrMux9[j];
255       end
```

```
256         end
257     end
258     if(M>= 194 && M<= 200) begin
259         for(j=0; j<64; j=j+1) begin
260             DoutLogicrMux[j] = DoutLogicrMux10[j];
261     end
262     end
263     if(M>= 201 && M<= 207) begin
264         for(j=0; j<64; j=j+1) begin
265             DoutLogicrMux[j] = DoutLogicrMux11[j];
266     end
267     end
268     if(M>= 208 && M<= 214) begin
269         for(j=0; j<64; j=j+1) begin
270             DoutLogicrMux[j] = DoutLogicrMux12[j];
271     end
272     end
273     if(M>= 215 && M<= 221) begin
274         for(j=0; j<64; j=j+1) begin
275             DoutLogicrMux[j] = DoutLogicrMux13[j];
276     end
277     end
278     if(M>= 222 && M<= 228) begin
279         for(j=0; j<64; j=j+1) begin
280             DoutLogicrMux[j] = DoutLogicrMux14[j];
281     end
282     end
283     if(M>= 229 && M<= 235) begin
284         for(j=0; j<64; j=j+1) begin
285             DoutLogicrMux[j] = DoutLogicrMux15[j];
286     end
287     end
288     if(M>= 236 && M<= 242) begin
289         for(j=0; j<64; j=j+1) begin
290             DoutLogicrMux[j] = DoutLogicrMux16[j];
291     end
292     end
293     if(M>= 243 && M<= 249) begin
294         for(j=0; j<64; j=j+1) begin
295             DoutLogicrMux[j] = DoutLogicrMux17[j];
296     end
297     end
298     if(M>= 250 && M<= 256) begin
299         for(j=0; j<64; j=j+1) begin
300             DoutLogicrMux[j] = DoutLogicrMux18[j];
301     end
302     end
303     if(M>= 257) begin
304         for(j=0; j<64; j=j+1) begin
305             DoutLogicrMux[j] = DoutLogicrMux19[j];
```

```
306     end
307     end
308 //=========================================
309 //converting input signals D1 and D2 into a binary code
310    @(cross(V(CLK) - vth, 1)) begin
311     d1_rr = d1_r;              //D1[n-2]
312     d1_r = d1;                 //D1[n-1]
313     d1    = V(D1) > vth ? 1:0; //D1[n]
314     d2_rr = d2_r;              //D2[n-2]
315     d2_r = d2;                 //D2[n-1]
316     d2    = V(D2) > vth ? 1:0; //D2[n]
317 //Decoding input binary code to select the correct coefficient
318 //from the selected LUT
319     out = DoutLogicrMux[d1*32+d1_r*16+d1_rr*8+d2*4+d2_r*2+d2_rr];
320   end
321
322 // converting internal signal into output signal
323     V(DOUT)   <+ transition(out , tdelay, trise, tfall);
324 end
325 endmodule
```

Listing II.2: Implementation of the Digital Cancellation Logic (DCL) in VerilogA.